RELUCTANCE MACHINES WITH FLUX ASSISTANCE

Thesis submitted for degree of

Doctor of Philosophy

at the University of Leicester

by

Ewan Roderick Tearlach Goodier BEng (Birmingham)

Department of Engineering

University of Leicester

April 2003

UMI Number: U170556

All rights reserved

INFORMATION TO ALL USERS
The quality of this reproduction is dependent upon the quality of the copy submitted.

In the unlikely event that the author did not send a complete manuscript
and there are missing pages, these will be noted. Also, if material had to be removed,
a note will indicate the deletion.

UMI U170556
Published by ProQuest LLC 2013. Copyright in the Dissertation held by the Author.
Microform Edition © ProQuest LLC.
All rights reserved. This work is protected against
unauthorized copying under Title 17, United States Code.

ProQuest LLC
789 East Eisenhower Parkway
P.O. Box 1346
Ann Arbor, MI 48106-1346

ACKNOWLEDGEMENT

The author would like to thank Professor Charles Pollock and all the members of the Centre for Advanced Electronically Controlled Drives, University of Leicester, for their help and advice. I would like to thank my family for their support and I would especially like to thank my wife Rebecca for her encouragement, support and motivation for which I am truly grateful.

ABSTRACT

Ewan Roderick Tearlach Goodier BEng (Birmingham)

RELUCTANCE MACHINES WITH FLUX ASSISTANCE

This thesis presents three reluctance machines with flux assistance. These machines provide alternative novel geometries that provide high efficiencies with a reduction in the ampere turns in the armature windings for torque production, lowering armature winding switching losses and reducing the power electronic rating. The Dual Stack Variable Reluctance Machine is a switched reluctance variant of the homopolar inductor alternator topology. The Single Stack Variable Reluctance Machine is a simplification of the Dual Stack machine. Both machines use a toroidal field winding to provide additional flux. The methods of connecting armature coils on each stator pole to utilise the armature flux and the choice of power electronic circuitry are important. Testing shows that such machines favour unipolar excitation with single coil per pole for the armature windings. Use of the field winding in series with the armature windings improves torque production. The Dual Stack Variable Reluctance Machine can have the mechanical angular displacement between the two stator stacks varied to provide an improved back emf waveshape for smoother torque production. The Single Stack Variable Reluctance Machine has parasitic and axial air gaps that pose interesting design issues (e.g. end thrust). Magnets can be placed in steel sections where flux is unidirectional. An ideal candidate for magnet insertion is the Flux Switching Motor. A Permanent Magnet Flux Switching Motor has been built that replaces the field windings with ferrite magnets. The Permanent Magnet Flux Switching Motor achieves efficiencies of over 80%. It adds no additional cost to the fan application as cost savings in lower temperature rated thermoplastics offsets the cost of magnets. A prototyping circuit incorporating a novel micro-processor program to alter the commutation timings as the machine operates has been designed to allow fast optimisation of each machine for minimum input power.

CONTENTS

SECTION TITLE **PAGE NUMBER**

CHAPTER 1

INTRODUCTION

This thesis presents three motor designs with novel geometry that are based on the concept of reluctance machines with flux assistance.

Five factors comprise the motivation behind this thesis:-

Increase in energy efficiency
Reduction in material content
Increase in power density
Reduction in kVA and complexity of power electronics
Lower cost

As with any machine design, any improvement in energy efficiency compared to existing equivalent machines is desirable. With greater awareness of the limitations of the planet's resources, there is an increasing global need to have greener products. This is becoming increasingly more important, especially in consumer electrical goods. All designs should ideally use less energy, be made from recyclable materials and give fewer emissions (e.g. noise, heat). However, such improvements must not unreasonably increase the price of the product.

By improving energy efficiency, smaller machines can be built which give the same output power as larger machines. The smaller machine uses less material quantities in its design.

The power density of machines can also be improved by having machines that produce more output power per unit volume or weight. Improvements in power density can allow either same sized machines to produce greater output power or smaller machines to produce the same level of output power.

The power electronics used with each machine has a kVA rating. Higher input power machines tend to have large, expensive power electronic circuitry to provide the high kVA rating. Reductions in kVA rating through efficiency gains allow smaller and cheaper components to be used. More significantly, reluctance machines are known to have a high kVA/kW penalty due to the need to deliver the field energy and the electromechanical energy through the power electronic converter. It is one objective of this thesis to design machines which minimise this penalty.

Cost is always an issue in the commercial environment. Machines built at lower expense but with at least the same level of quality allow the manufacturers to develop their sales markets to their advantage. Cost savings can be obtained, for example, through smaller machines needing less materials, with cheaper components and materials being used.

It has been noted [1] that motor efficiency is not always of concern to some customers of motor manufacturers such as equipment manufacturers since it is the buyers of the equipment that pay the kWh bills. The apparent lack of concern has been recognised in the USA and a Federally-sponsored study recommended the introduction of measures to encourage the inclusion of explicit efficiency data on nameplates and to even introduce mandatory minimum efficiency levels.

Greener products are becoming more marketable, notwithstanding any associated higher costs involved, as consumer awareness grows and new government legislation comes into effect. The net result has been that consumers are more likely to buy the greener product that would lower their electricity bill (through using less kWh). Examples of commercially successful 'green' electrical products that have been more expensive than similar non-'green' products are the Lo-Watt range of fans by Vent-Axia (http://www.vent-axia.co.uk) and, on a larger domestic scale, energy-saving light bulbs produced by e.g. Philips (7W c.f. 40W).

These five factors, when suitably assessed and applied to each machine application, can be used towards the design of lower cost, higher efficiency machines. Into this can be incorporated the use of higher cost materials, such as permanent magnets, that have properties known to benefit machine design.

Three motor designs based on the concept of reluctance machines with flux assistance have been designed, built and tested. Each design incorporates a novel geometry. The appropriate use of field windings and magnets in such designs was required. This thesis provides a complete account of the design procedures, testing and result analysis for the building of dual stack and single stack variable reluctance machines with toroidal field windings and for a Permanent Magnet Flux Switching Motor.

Chapter 2 covers background information regarding reluctance machines with flux assistance, introducing the relevant types of machine. The uses of field windings and permanent magnets in machine designs are discussed. The placement and flux orientation of the magnets are discussed. The basis of the motor designs within this thesis is discussed.

Chapter 3 discusses the power electronic circuitry used for reluctance based machines and shows that the choice and complexity of circuit is dependent on the type of armature windings used. Also discussed is the development of microprocessor programs and experimental control electronics circuitry for prototyping the optimum electronic commutation timings for two phase variable reluctance and flux switching machines.

Chapter 4 describes the design, build and testing of a dual stack variable reluctance drive. The use of a dc current toroidal field winding for additional flux is shown. The design is based upon the design structure of the homopolar inductor alternator. Optimum winding types and methods of improving torque production are shown. The effect of changing the angular displacement between the two stacks is discussed.

Chapter 5 shows a simplification of the dual stack design. The design, build and testing of a single stack variable reluctance motor with a dc current toroidal field winding is described. Methods for improving torque production and the effects of magnetic forces across the air gaps not between the stator and rotor poles are discussed.

Chapter 6 discusses the replacement of a field winding in the Flux Switching Motor with permanent magnets. The Permanent Magnet Flux Switching Motor was designed to replace an induction motor in a Domus S1R 100mm axial wall fan. The lamination design is presented through the use of Finite Element Analysis and spreadsheet calculations. The test results and conclusions are presented with factors such as dB noise levels, air flow performance, and rotor skewing included.

Chapter 7 compares all three reluctance machines with flux assistance. The use of field windings and magnets are compared, as are the types of armature windings and power electronic circuits. The geometries and flux paths of the machines are also compared.

Chapter 8 presents the conclusions from this thesis and offers suggestions for future work.

The thesis ends with a bibliography, a list of publications and the appendices.

CHAPTER 2

BACKGROUND TO RELUCTANCE MACHINES WITH FLUX ASSISTANCE

2.1 INTRODUCTION TO TYPES OF MACHINE RELEVANT TO THIS THESIS

This thesis shows the use of field windings and permanent magnets in novel geometries as novel alternative solutions to the problems of energisation and de-energisation in the armature windings.

A permanently energised dc current coil retains dc flux as a field. Sets of diametrically opposing coils can be energised in turn to produce electromagnetic energy conversion as armatures, to generate torque to turn the rotor.

It is appropriate to highlight some uses of field windings and magnets in both generator and motor applications. They are usually applied either as a generator or as a motor, but some designs are both. The use of field windings and permanent magnets in generator and machine topologies is shown to enlighten the reader for explaining the machines developed in this thesis.

The theory behind acoustic noise cancellation techniques, cogging torque reduction and the use of powdered iron as a replacement of laminated steel sections is shown. The airflow regulatory requirements for fan applications are presented. The theory is offered as background information for the reader to better understand the work presented.

Permanent magnets can be applied to both machines and generators. The advantages of their use has to be placed in context with the disadvantages. Field windings offer a source of dc flux that is varied by the level of field current. Magnets can replace field windings meaning that no field current is needed and there will be no heating effects due to I^2R copper losses. The cost of magnets can be cheaper if designed correctly but their B-H curves and properties may restrict the design of machines with magnets. Permanent magnet excitation can be beneficial in smaller machines but tends to be costly and offer smaller efficiency gains in larger machines [1]. With some permanent magnet materials the field flux changes little with increasing temperature. There is little or no control of the magnet field making their use unsuitable in some applications. One of the main disadvantages of permanent magnets is demagnetisation which can occur through high armature mmfs, large mechanical disturbances (e.g. knocking, disassembly), excessive temperatures, etc.

The efficiency improvements that can be obtained with the use of magnets is increasingly being recognised in terms of environmental benefits. The lack of rotor I^2R loss and magnetising current stator loss in permanent magnet machines is mainly responsible for greater efficiency when compared to induction machines.

2.2 THEORY OF TORQUE PRODUCTION

Linear analysis assumes that there is no magnetic saturation, that is to say that inductance is unaffected by the current [2]. Fringing flux is ignored and all the flux is assumed to cross the air gap in a radial direction (between the stator and rotor pole faces). Mutual coupling between phases is small and is ignored. The voltage equation for one phase is (2.1):-

$$V = IR + \frac{d\psi}{dt} = IR + \omega\frac{d\psi}{d\theta} = IR + \omega\frac{d(LI)}{d\theta} = IR + L\frac{dI}{dt} + \omega\, I\frac{dL}{d\theta} \tag{2.1}$$

where V is the terminal voltage, I is the current, ψ is the flux-linkage, R is the phase resistance, L is the phase inductance, θ is the rotor position and ω is the angular velocity. The last term of (2.1) is also known as the back emf, E (2.2). The supply voltage is dropped across a resistance volt drop, a L(dI/dt) term and the back emf. A graph of inductance versus rotor position can be used to give dL/dθ.

$$E = \omega\, I\frac{dL}{d\theta} \tag{2.2}$$

The instantaneous electrical power is given by (2.3):-

$$VI = I^2R + LI\frac{dI}{dt} + \omega\, I^2\frac{dL}{d\theta} \tag{2.3}$$

The rate of change of magnetic stored energy at any instant is given by (2.4):-

$$\frac{d}{dt}\left(\frac{1}{2}LI^2\right) = \frac{1}{2}I^2\frac{dL}{dt} + LI\frac{dI}{dt} = \frac{1}{2}I^2\omega\frac{dL}{d\theta} + LI\frac{dI}{dt} \tag{2.4}$$

From the Law of Conservation of Energy, the mechanical power conversion P=ω T is what is left after the resistive loss (I^2R) and the rate of change of magnetic stored energy are subtracted from the input power (VI), where T is the instantaneous electromagnetic torque (2.5).

$$T = \frac{1}{2}I^2\frac{dL}{d\theta} \tag{2.5}$$

A non-linear analysis [2] uses magnetisation curves (curve of flux linkage versus current at a particular rotor position). The stored magnetic energy, W_f, and co-energy, W_c, are defined in (2.6):-

$$W_f = \int I d\psi \qquad W_c = \int \psi dI \tag{2.6}$$

In a magnetically linear device with no saturation, $W_f = W_c$. When saturation is taken into account, $W_f < W_c$ (*Figure 2.1*). In a rotor displacement $\Delta\theta$ at constant current, the energy exchanged with the supply is (2.7):-

$$\Delta W_e = \int EIdt = \int I\frac{d\psi}{dt}dt = \int Id\psi \qquad (2.7)$$

and the change in magnetic stored energy is (2.8) (*Figure 2.1*):-

$$W_f = OBC - OAD \qquad (2.8)$$

The mechanical work done is (2.9) (*Figure 2.1*):-

$$\begin{aligned} \Delta W_m &= \Delta W_e - \Delta W_f \\ \Delta W_m &= ABCD - (OBC - OAD) \\ \Delta W_m &= OABCD - OBC \\ \Delta W_m &= OAB \end{aligned} \qquad (2.9)$$

and this is equated to $T\Delta\theta$, so that, in the limit, when $\Delta\theta \to 0$,

$$T = \left[\frac{dW_c}{d\theta}\right]_{I=\text{constant}} \qquad (2.10)$$

The average torque is (2.11):-

$$T_{average} = \frac{SW}{2\pi} \qquad (2.11)$$

where S is the number of strokes per revolution (S=number of phases x number of rotor poles), and W is the area enclosed between the unaligned magnetisation curve U, the aligned magnetisation curve A and the line UA at current I (*Figure 2.2*).

The area W is described further in [2, 3] in which the use of freewheel diodes in power electronic circuits such as the asymmetric half bridge converter (Chapter 3) can aid in providing additional positive torque, as shown, by example, in *Figure 2.3*.

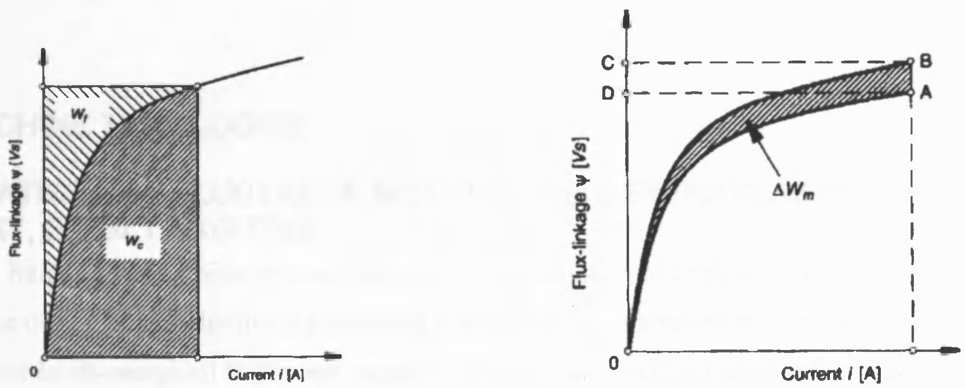


Figure 2.1 :- Coenergy, W_c, stored field energy, W_f, and calculation of instantaneous torque from the rate of change of coenergy at constant current [2].

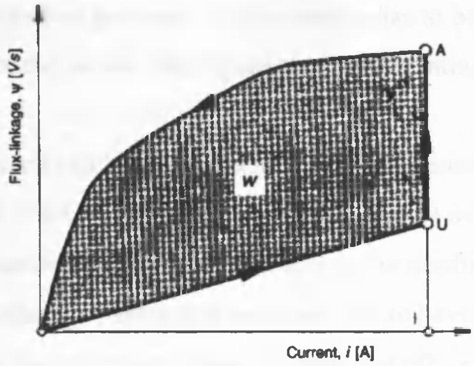


Figure 2.2 :- Calculation of average torque [2].

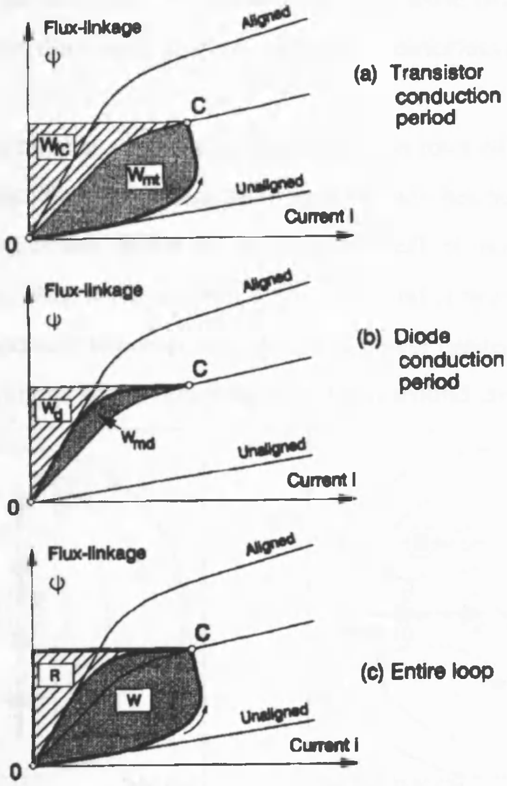


Figure 2.3 :- Example of energy conversion loop showing stages comprising average torque [2].

2.3 MACHINE TOPOLOGIES

2.3.1 SWITCHED RELUCTANCE MOTORS AND GENERATORS (DOUBLY SALIENT, SINGLY EXCITED)

Switched Reluctance Machines produce torque by the excitation of a phase winding while the inductance of that phase winding is increasing due to rotation of a salient rotor pole. The magnetic circuit must be de-energised to prevent negative torque being produced when the salient rotor pole reaches a position of minimum reluctance. To continue torque production a subsequent phase winding has to be energised. A substantial amount of energy is therefore delivered and recovered due to the continual energisation/de-energisation process. To this energy has to be included energy used in electromechanical work. Hence the power electronics have a high rating.

The Switched Reluctance Machine can be used as both motor and generator and is ideally suited as an Integral Starter/Generator, ISG, (ISA – Integrated Starter/Alternator) in applications such as aircraft engines [4, 5]. Switched reluctance generators are the dual of the machine as a motor [4, 6]. They do not require magnets or field windings. Permanent magnets, due to having a constant flux source, give problems when faults occur, as the excitation cannot be switched off. Field coils can also fault due to short or open circuits within the windings. By not having either these faults are avoided. Such starter/generator systems can be done with position sensors or sensorless as in [7].

A connecting device such as a turbine or aircraft engine turns the rotor of the generator. The generator needs a bus voltage to generate. For each phase there are two switches and two diodes. Initially, both switches are on, building up a current. When the switches turn off, current decreases due to conduction through the diodes. But when $dL/d\theta < 0$, the back emf is negative and it tends to increase the current and convert the mechanical power into electrical power (generating) [4]. The generated current waveform is a mirror image of the motoring waveform around the aligned rotor position.

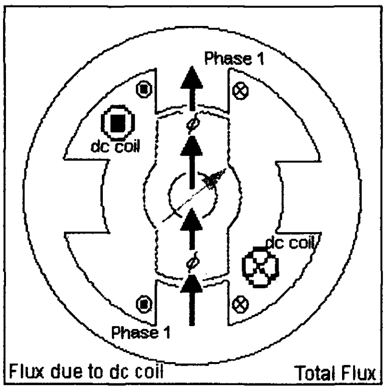


Figure 2.4a :- Switched reluctance version.

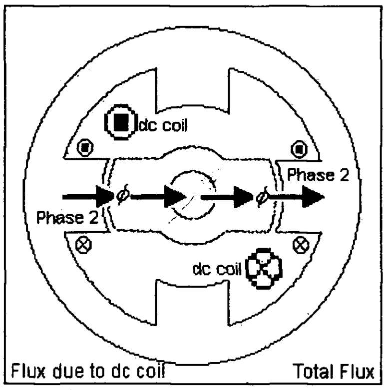
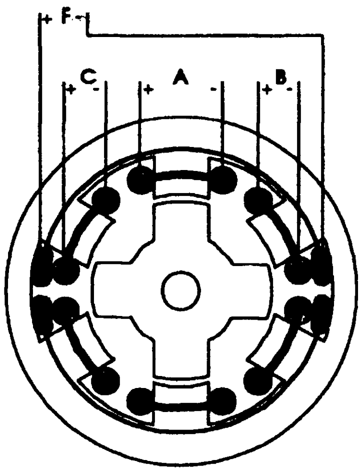


Figure 2.4b :- Switched reluctance version.



Structure of a 6/4 pole SRDC motor

Figure 2.5 :- Switched reluctance drive with auxiliary dc winding [10].

2.3.2 SWITCHED RELUCTANCE WITH AN ADDITIONAL FULLY PITCHED COIL

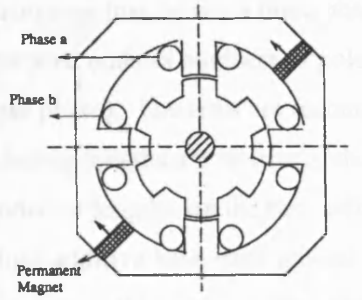
The addition of a fully pitched coil to switched reluctance machines has been addressed by many researchers [8-13] and the use of a fully pitched coil or a permanent magnet that is mutually coupled to all the machine phase windings has been proposed. This additional winding is used as a commutation winding [11] to retain the flux within the motor while one winding current decreases and another winding increases. It is shown that the kVA rating of the power electronic converter was reduced. Other researchers [10, 12] have demonstrated improved motor performance from continuous energisation of the fully pitched coil. The simplest example of these systems is illustrated by *Figure 2.4a* and *Figure 2.4b*. A switched reluctance machine with four stator poles and two rotor poles has a coil wound around each stator pole with diagonally opposite coils connected together to form a phase winding. A dc current permanently energised fully pitched coil retains dc flux as a field. Two sets of diametrically opposing coils are energised in turn to produce electromagnetic energy conversion as armatures. Subsequent energisations of each armature set of coils (labelled as Phases 1 and 2 in *Figures 2.4a* and *2.4b*) give resultant flux that aligns the rotor with the energised stator poles.

A Switched Reluctance Drive with auxiliary DC windings achieves built-in field excitation (*Figure 2.5*)[10]. A larger percentage of armature current is used in production of torque rather than for magnetising the motor (building the field). Up to 50% increased torque density than a conventional Switched Reluctance Drive is achieved without losing performance. To accommodate the extra copper area in the 6/4 pole combination, two stator slots are deepened for the fully pitched field windings. It allows better coupling with the phase windings as it acts as a secondary winding of a transformer (absorbing and releasing transitional energy). Further information can be found in United States Patent 5,866,964 [12].

2.3.3 SWITCHED RELUCTANCE MACHINES WITH PERMANENT MAGNET DC FLUX EXCITATION IN THE STATOR

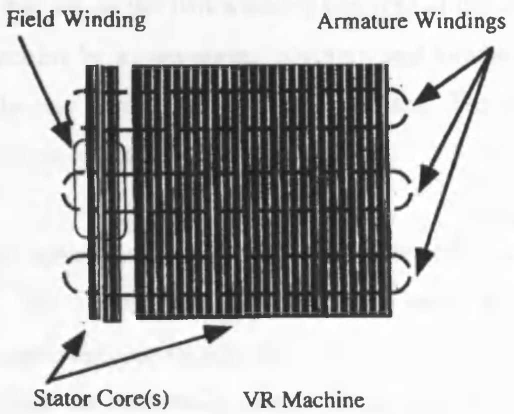
Field excitation from permanent magnets in the stator yoke has led to a doubly salient single phase permanent magnet generator with improved torque production and efficiency compared to induction and variable reluctance machines (*Figure 2.6*)[14, 15]. It has two permanent magnets in the 2-phase 4/6 structure. The rotor has no windings on it. The structure of the stator is simple, even with the inclusion of the magnets. It is the reverse arrangement of a 6/4 motor described in [16].

The phases of a Variable Reluctance Machine are decoupled from each other, making its use as a generator more appealing since the faulted condition (usually shorted) will not conduct current should the remaining phases continue to operate [17]. A problem for the Variable Reluctance Generator is that of a lack of self-excitation (c.f. Permanent Magnet and Lundell generators). A separate stack and small field winding could be installed adjacent to the main stack to provide the excitation requirements during starting (*Figures 2.7, 2.8*)[17]. The armature windings of the main stack are extended to enclose the poles of the second stack. The alternative would be to place magnets in the core of the second stack to replace the field windings. However the magnets cannot be turned off when the generator has started but the flux from the magnets could be used for additional torque.

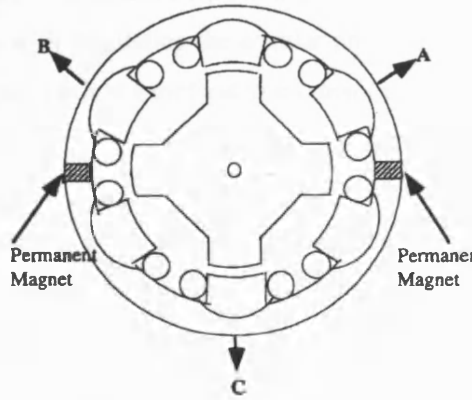


Magnetic Structure of Doubly Salient Single Phase Permanent Magnet Generator.

Figure 2.6 :- Doubly Salient Single Phase Stator Permanent Magnet Generator [14, 15].

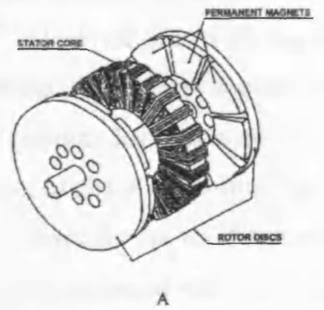


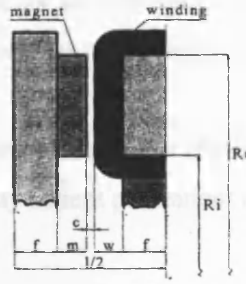
Doubly salient doubly excited VR machine.



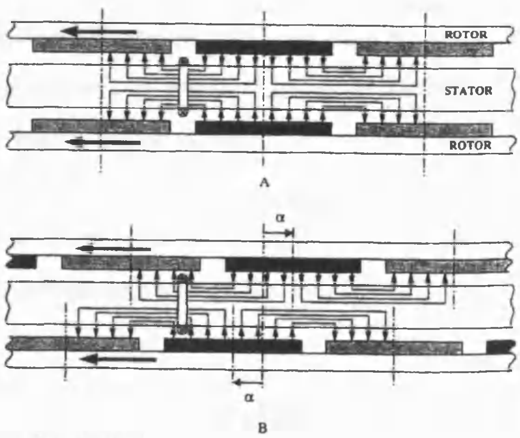
Showing side view of excitation when employing permanent magnets.

Figures 2.7, 2.8 :- Variable Reluctance Generator with stator permanent magnets and field winding adjacent to stator stack [17]





Slotless Axial-Flux PM Machine: A, basic layout; B, cross-sectional view of one-half of the machine electromagnetic structure.



Mechanical weakening of the flux linkage via the regulation of the angular phase displacement 2α between the two PM rotor discs.

Figures 2.9, 2.10 :- Axial Flux Permanent Magnet Machine layout and rotor disc displacement [18].

An Axial Flux Permanent Magnet Machine can be used for starter/alternator concepts [18]. The magnets give axial flux that is used for torque production or for generation (*Figures 2.9, 2.10*). It consists of a single toroidal stator between two steel rotor discs with surface mounted permanent magnets. The magnets give axially directed magnetic field in the air gaps of the machine. The stator is a slotless toroidally wound strip-iron core that carries a three phase winding wound in a toroidal fashion by means of as many concentrated coils as numbers of poles to form each phase of the winding (i.e. one coil per pole and per phase). The coils are rectangular according to the core cross section. The axially directed end winding lengths are relatively short resulting in low resistance and lower copper losses. The active conductor lengths are the two radial portions facing the magnets, whose polarities are arranged to induce additive back emfs around a stator coil. A tangential force acts on the active conductor lengths when a current is in the stator coils to interact with the flux driven by the magnets across the two annular air gaps into the stator core. Machine torque is the contribution of all forces that act on the two working surfaces of the core. Weakened flux linkage is possible in this type of machine by a cam-spring governor and treadle lever regulating the angular phase displacement between the two permanent magnet rotor discs. The net result is a method to maintain constant back emf magnitude at higher rotor speeds.

The doubly salient permanent magnet machine also exists in a two stator stack form (*Figures 2.10, 2.11*)[19]. The 4 stator, 6 rotor single phase machine has high torque capability, symmetric current and inductance but cannot start from a fully aligned rotor position as there is neither permanent magnet torque nor reluctance torque. It has been solved by joining two single phase machines together by installing two rotors on the same shaft and two stators in one machine frame. The stator or rotor of one machine can be shifted with respect to the other machine by 45 degrees , producing a two phase motor. When one side of the machine is in the fully aligned position, the other side is fully unaligned, giving 90 electrical degrees out of phase. The pulsating reluctance torque (proportional to the square of current) does not contribute to energy conversion as, between each side of the motor, it is 180 degrees out of phase, cancelling itself out. There is no electromagnetic coupling between the two stacks hence a space for the windings must exist between the two stacks. The additional copper requirements and increased size are compensated by higher torque capability, smoother torque and good stating torque when compared to a 6/4 Doubly Salient Permanent Magnet structure. Improved control methods for Doubly Salient Permanent Magnet Motors can be found in [20], highlighting a method for improving the unequal voltage distribution from a split capacitor power converter.

A similar magnet arrangement can be found in (*Figure 2.13* and *Figure 2.14*)[21 and 22] where three phase 6/4 doubly salient permanent magnet machines are shown.

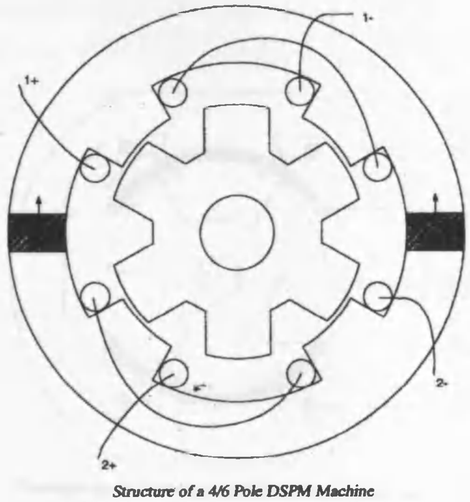


Figure 2.11 :- 6/4 Doubly Salient Permanent Magnet Machine with magnets on stator [19].

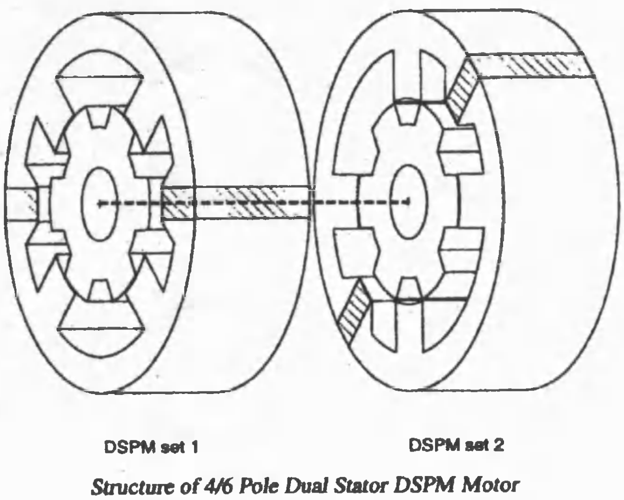
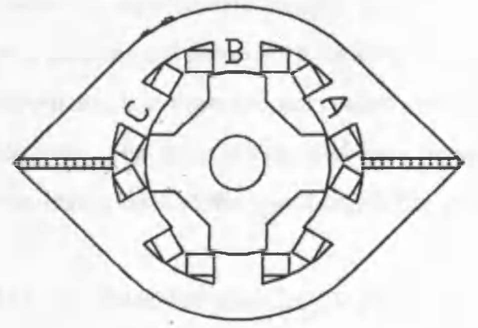
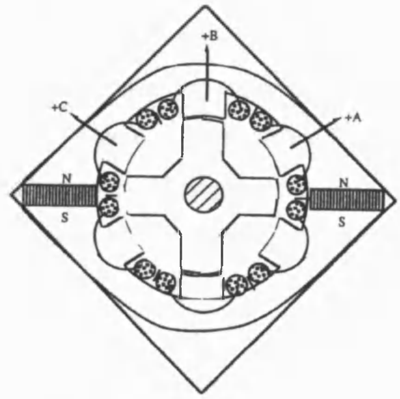


Figure 2.12 :- Dual stator version of 6/4 Doubly Salient Permanent Magnet Machine with magnets on stator [19].



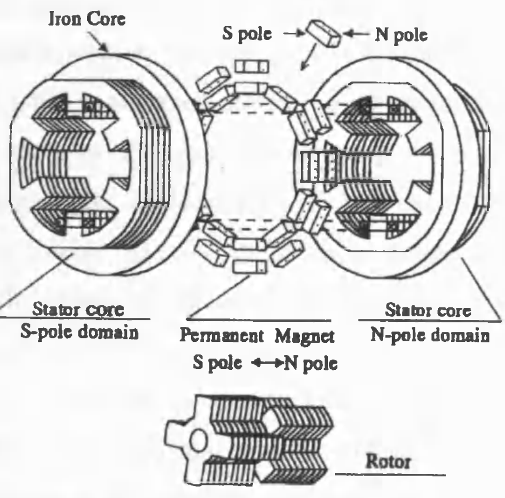
Cross section of DSPM motor.

Figure 2.13 :- An alternative 4/6 Doubly Salient Permanent Magnet Machine [21].



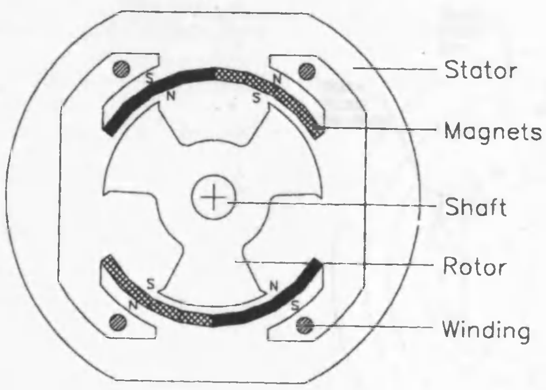
Cross Section of Doubly Salient Permanent Magnet Motor with Stator Magnet Excitation.

Figure 2.14 :- Another version of the 4/6 Doubly Salient Permanent Magnet Machine [22].



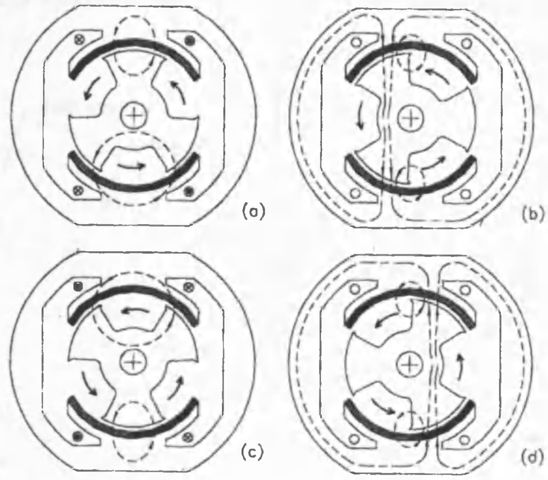
Structure of a doubly salient-pole homopolar machine.

Figure 2.15 :- Doubly salient homopolar machine (dual stator) with permanent magnets [23].



The single-phase, 2/3 configuration flux-reversal generator prototype.

Figure 2.16 :- Flux Reversal Machine structure [24].



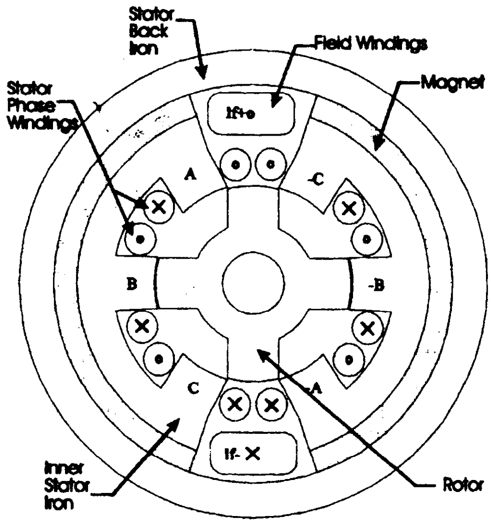
The principle of operation of the flux-reversal machine.

Figure 2.17 :- Flux Reversal Machine operation [25].

A doubly salient homopolar machine exists with axial flux permanent magnets between the stator cores (it is a permanent magnet synchronous machine). The magnets make one stator core a North-pole domain and the other a South-pole domain (*Figure 2.15*)[23]. The windings for one phase are shown and it is drawn to the reader's attention that the rotor is skewed by 45 degrees into two adjacent sections. The laminations used were those for a Switched Reluctance Motor. It was noted that the iron losses limited the speed capability of such a machine.

The Flux-Reversal Machine [24] is another type of brushless doubly-salient permanent magnet machine (*Figures 2.16, 2.17*). It has a three pole rotor and a two pole stator. Two magnets of opposite polarity are placed on each stator pole face. *Figure 2.17a* is an equilibrium position where the flux is set up by the magnets circulates entirely within each stator pole and no flux is within the stator back-iron. No flux links the coils in this position. In *Figure 2.17b* the rotor is displaced by 30 degrees anticlockwise, so that the rotor poles overlap one or other of the magnets. Flux now passes through the coils and the back-iron and the phase flux is at a maximum at this position. In *Figure 2.17c* the rotor is at a second equilibrium position, displaced from the first one by 60 degrees, hence there is no flux in the back iron and no flux linking the coils. A further 30 degree anticlockwise displacement gives *Figure 2.17d* and phase flux is at a maximum but in the opposite direction to that in *Figure 2.17b*. The back emf is zero at *Figure 2.17b* and *Figure 2.14d* maximum positive at *Figure 2.17a* and maximum negative at *Figure 2.17c*. The motor operates because of the variable flux-linkage induces an emf that interacts with the alternating armature current. Although field excitation is due to permanent magnets, the flux-linkage of the armature windings is modulated by the variation of the magnetic circuit reluctance as the rotor rotates, in such a way that a bipolar emf is induced without rotating magnets.

2.3.4 SWITCHED RELUCTANCE MACHINES WITH PERMANENT MAGNETS AND FULLY PITCHED COILS ON THE STATOR



Doubly Salient Permanent Magnet Motor (DSPM)

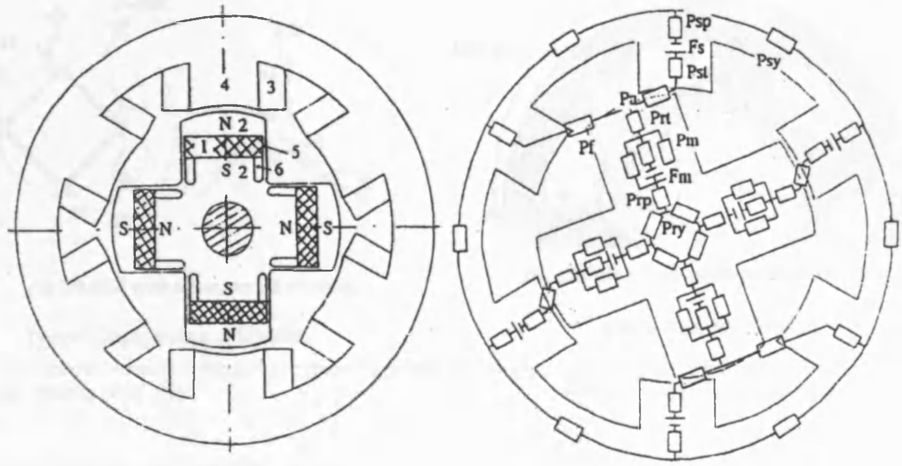
Figure 2.18 :- Doubly Salient Permanent Magnet Motor with Flux Control [8, 25].

Lower cost designs can be achieved whilst maintaining the higher efficiencies by using ferrite magnets. Such a design is a doubly salient permanent magnet motor with flux control (*Figure 2.18*)[8, 25]. It places the magnets between the stator teeth and the stator yoke, allowing concentration of permanent magnet flux through one tooth at a time. The area of the magnets is wide allowing the use of cheaper ferrites. The space between the magnets accommodates a dc field winding that boosts or weakens the permanent magnet flux (for low and high speeds respectively). This motor still retains some design similarities with the switched reluctance drive but is operated as a brushless dc machine as electromechanical power conversion results from interaction of permanent magnet and/or field flux with stator current. It differs from a brushless dc machine family as the flux in the air gap changes with permeance variation instead of magnet rotation.

2.3.5 SWITCHED RELUCTANCE MACHINES WITH PERMANENT MAGNETS IN THE ROTOR

The Switched Reluctance Machine has some disadvantages as part of its operation. It is known to have poor utilisation of the active copper and iron due to its variable reluctance action. The stator phases carry a current component for torque production and a current component for magnetisation, having a current commutation associated with a large turn-off inductance that reduces torque production capability, and a substantially reduced air gap is used to push the motor into the highly saturated region [26]. To reduce these 'deficiencies' permanent magnets are used to provide rotor excitation (*Figure 2.19*)[27]. This doubly salient permanent magnet (DSPM) motor is similar to a square-waveform PM brushless DC motor, since the torque it developed is dominated by current-magnet interaction (mutual torque), instead of the current-iron (reluctance torque) of Switched

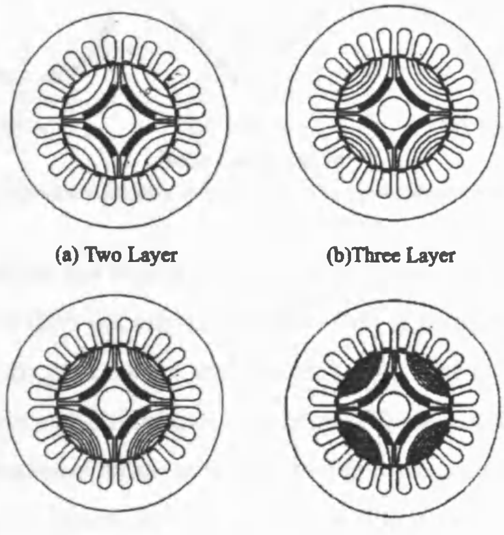
Reluctance motors. An electronically commutated Doubly-Salient Permanent Magnet (DSPM) motor has been developed such that it has the basic structure of a three-phase 6/4 Switched Reluctance motor, except for the presence of four high-energy (NdFeB) permanent magnets inside the rotor poles [26]. The torque is produced by the same method as for permanent magnet brushless DC motors but the cogging torque is used to keep the rotor stationary when there is no supply. The rotor poles act as flux guides for the permanent magnets. Such a motor has been designed with the aid of a magnetic equivalent circuit (*Figure 2.19*).



Cross-sectional configuration of the new electronically-commutated DSPM small motor

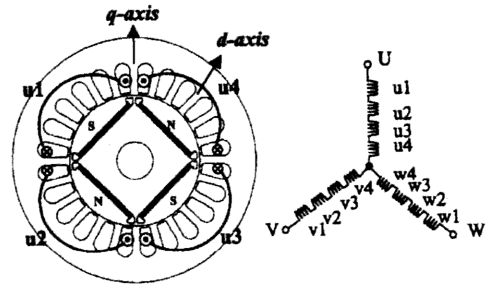
Equivalent circuit for the new DSPM motor.

Figure 2.19 :- Doubly-salient permanent magnet motor with rotor magnets and its magnetic equivalent circuit [26].

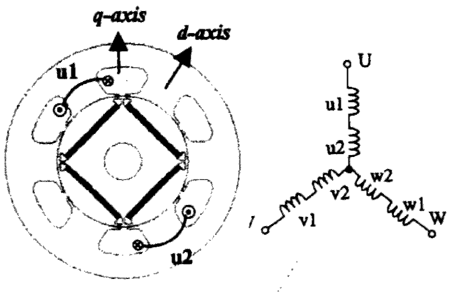


The rotor configurations of PMA SynRM.

Figure 2.20 :- Increasing number of flux barriers for deepest slit inserted permanent magnet synchronous motor [28].



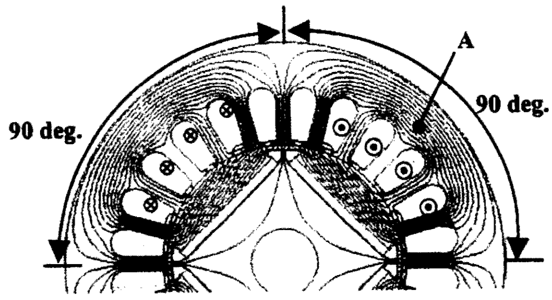
(a) IPMSM with distributed winding.



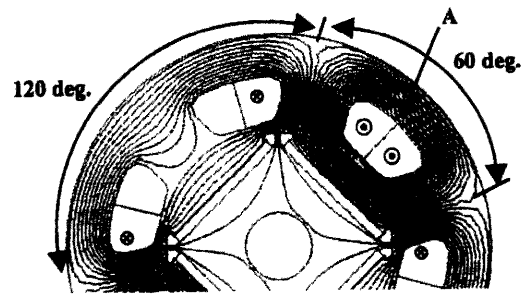
(b) IPMSM with concentrated winding.

Typical Configuration of IPMSM.

Figure 2.21 :- Interior Permanent Magnet Synchronous Motor with different winding types [28].



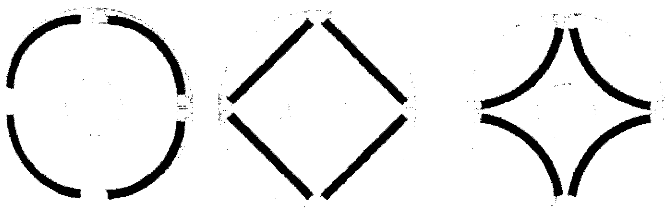
(a) IPMSM with distributed winding.



(b) IPMSM with concentrated winding.

q-axis flux distribution.

Figure 2.22 :-Flux paths within each winding type Interior Permanent Magnet Synchronous Motor [28].



(a)Type A (b) Type B (c) Type C

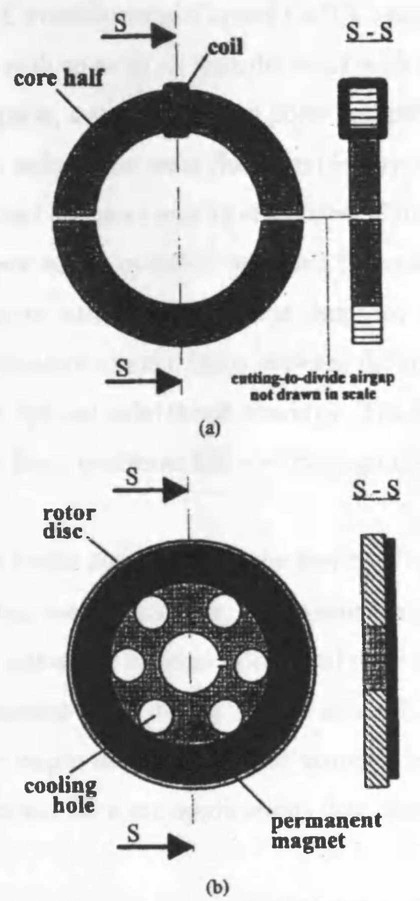
Rotor configuration.

Figure 2.23 :- Magnet placement strategies within Interior Permanent Magnet Synchronous Motor [28].

The synchronous reluctance motor has higher efficiencies as it only utilises the reluctance torque generated by rotor saliency and does not use the magnetic flux of permanent magnets. The efficiency can be improved by reducing the peak current and copper losses. The use of magnets can be used to produce highly efficient magnet assisted reluctance motors (*Figure 2.20*)[28]. Permanent magnets can be added to a synchronous reluctance motor to reduce copper losses and increase the efficiency (to 94.4% for a 750W motor, a 3.6% increase). They are embedded in the deepest slit of a flux barrier to take into consideration irreversible demagnetisation under a load current. The magnets produce 24% of the total torque.

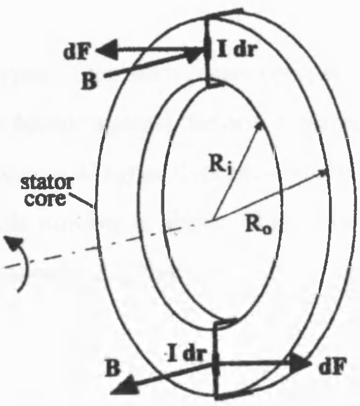
A range of brushless motors named Interior Permanent Magnet Synchronous Motors (IPMSM) also exist. It has been shown that analysis of the placement of the permanent magnets within the rotor of an IPMSM can minimise iron losses and that a concentrated winding (rather than a distributed winding) using rectangular wire rather than circular wire minimises copper losses (*Figures 2.21, 2.22*)[29]. If the magnets are treated as air, the flux density is greater from the concentrated winding, even though there is an asymmetry in the winding distribution due to the nature of the machine. Comparison of rotors with differing saliency ratios but with same thickness and length (hence volume) of magnets, shows that the flux density in the stator increases as the magnets are embedded deeper into the rotor (this increases the saliency ratio). The flux density affects iron losses and, as the saliency and flux density increases, so too will the iron losses. Thus a Type A [29] rotor is best for minimising the iron losses (*Figure 2.23*). Rectangular windings improve the packing factor and provide a reduction in the mean length per turn allowing improvements in copper losses and a reduction in overall machine size and weight.

The slotless axial flux permanent magnet machine (AFPM) is used in both generator and motor applications (*Figures 2.24, 2.25*)[30]. It differs from radial-flux machines due to the flux in the air gap being along the mechanical axis of the machine. The machine stator is a slotless strip-iron core that carries winding coils (the core may be in two halves joined together for production purposes). The toroidal stator is placed between two rotor discs which carry axially polarised permanent magnets. The magnets drive flux across the annular air gaps into the stator core. The flux travels circumferentially along the toroidal core, back across the air gaps then back through the back iron of the permanent magnet rotor discs. The current in the conductors interacts with the flux from the magnets to give a tangential force. A single rotor, twin stator version is discussed in [31] (*Figure 2.26*). Multi-stage AFPM designs (including water cooled form) also exist. The toroidal core was replaced with a water duct in a two-stage version [32] (*Figure 2.27*).



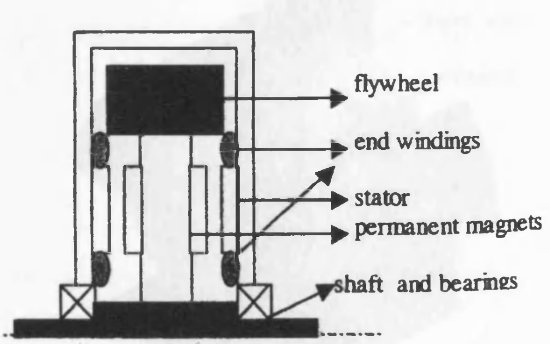
Schematic view of the main parts of an AFPM. (a) Slotless stator core with concentrated coils. (b) Surface-mounted PM rotor disc.

Figure 2.24 :- Slotless Axial Flux Permanent Magnet Machine [30].



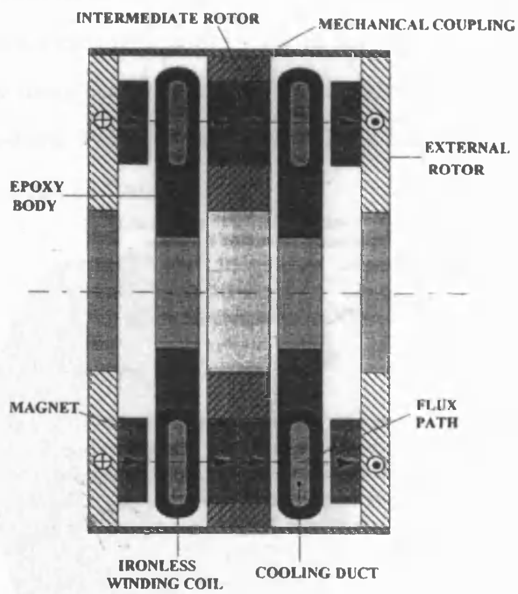
Schematic representation of the torque production mechanism in AFPM's.

Figure 2.25 :- Torque production within Axial Flux Permanent Magnet Machine [30].



AFPM machine flywheel arrangement.

Figure 2.26 :- Single rotor, twin stator AFPM machine [31].



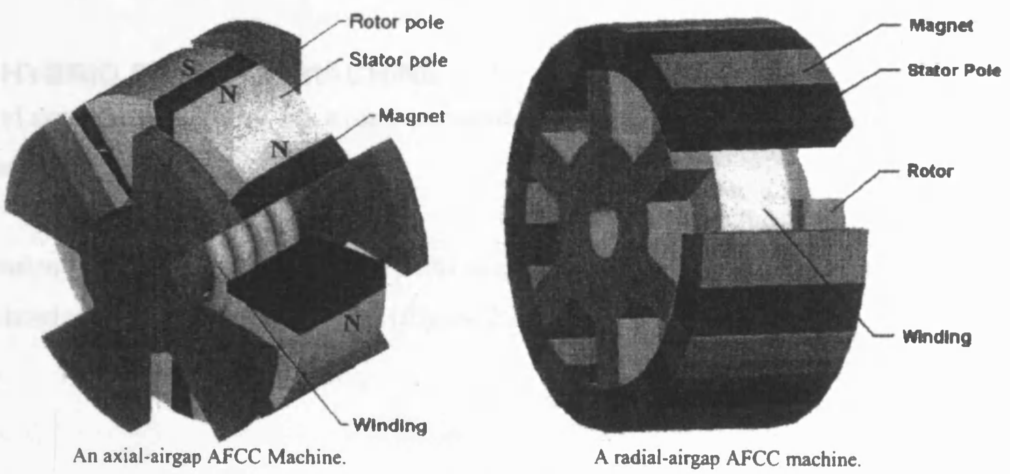
Cross-section of two-stage axial-flux permanent magnet machine with ironless water-cooled stator winding

Figure 2.27 :- Multi-stage, water cooled AFPM machine [32].

Axial Flux Circumferential Current (AFCC) machines also use permanent magnets and exist in two forms; one with an axial air gap, the other with a radial air gap. (*Figure 2.28*)[9, 33]. Such machines have three parts, a stator with iron poles and permanent magnets, a circumferential winding and a rotor with salient poles. The main flux provided by two nearby permanent magnets is concentrated in the stator pole and becomes axially orientated. This flux then passes across the air gap, through the rotor pole and rotor centre cylinder, returning to the adjacent stator pole. The winding links all the main flux. The rotor poles on the two end plates are shifted by one pole pitch. The axial air gaps could introduce unbalanced axial force between the rotor and stator (due to mechanical tolerances) and thus may require special axial thrust bearings. The radial air gap version of the AFCC has similar unbalanced force problems but conventional bearings can still be used.

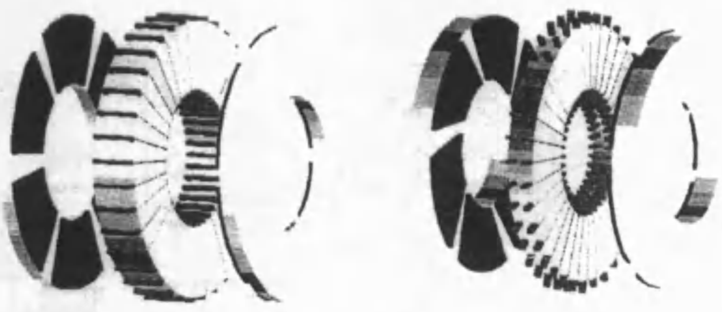
Much work exists that compares the merits of similar machine types. One such paper compares the two stator/one rotor Axial Flux Permanent Magnet Synchronous Motor against the one external cylindrical stator/one internal cylindrical rotor Radial Flux Permanent Magnet Synchronous Motor [34]. In simplest terms the Axial Flux motor is best when the pole number is above 10 and the ratio of axial motor length to external motor diameter is below 0.3. It demonstrates that some motor designs are better suited for some applications than others.

TORUS concept machines are internal stator, external rotor designs (*Figures 2.29, 2.30*)[35]. The strip wound steel stator is slotted such that polyphase windings can be placed in the slots. The rotors are disc shaped and carry axially magnetised NdFeB permanent magnets on the inner surfaces. Two versions exist, the names of which show the flux direction from the magnets. The TORUS NN motor has magnet driven flux entering the stator that then travels circumferentially along the stator core. The TORUS NS motor has magnet driven flux entering the stator that then travels axially along the machine axis of rotation. The NN version has back-to-back wrapped windings and the NS version has short-pitched lap windings. Further design information can be found in [35].



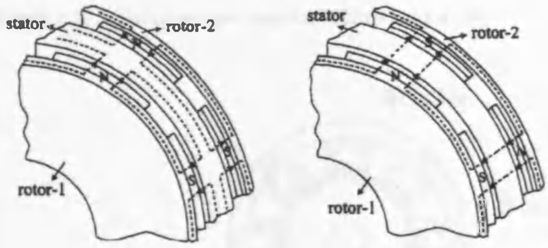
An axial-airgap AFCC Machine. A radial-airgap AFCC machine.

Figure 2.28 :- Axial Flux Circumferential Current machine in axial and radial air gap forms [33].


(a) (b)

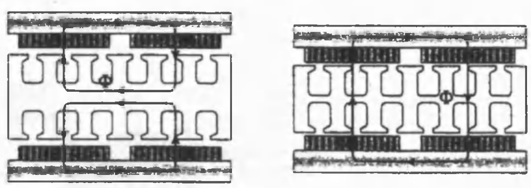
Slotted TORUS concept machine models (a) NN type, (b) NS type

Figure 2.29 :- TORUS concept machine layouts [35].



(a) NN type (b) NS type

3D Flux paths of TORUS NN type and TORUS NS type structures


(a) NN type (b) NS type

One pole pair of the slotted TORUS machine at the average diameter and flux directions for both NN type and NS type structures

Figure 2.30 :- TORUS concept machine flux paths for each machine type [35].

2.3.6 STEPPER (STEPPING) MACHINE

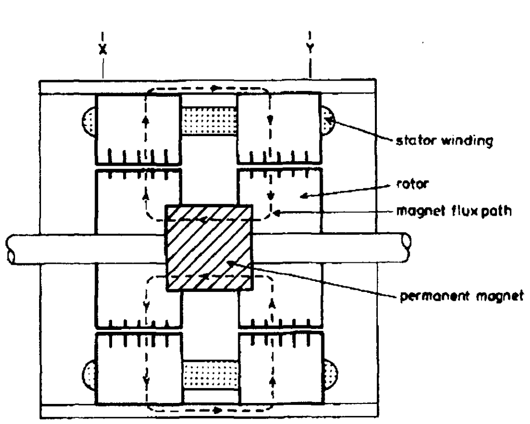
A stepper machine [36] uses a magnetic field to move a rotor. Stepping can be done in full step, half step or other fractional step increments. Voltage is applied to poles around the rotor. The voltage changes the polarity of each pole, and the resulting magnetic interaction between the poles and the rotor causes the rotor to move. Stepper machines provide precise positioning and ease of use, especially in low acceleration or static load applications.

There are three types of stepper machine: Multi-stack variable reluctance stepper machine, Single stack variable reluctance stepper machine, and hybrid stepper machine.

2.3.6.1 HYBRID STEPPING MACHINE

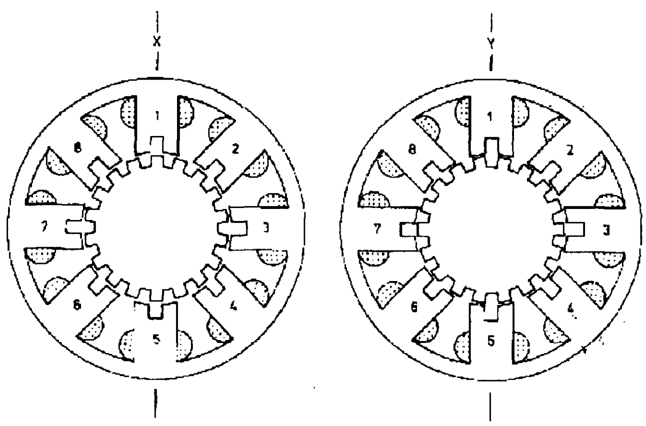
The hybrid stepper machine [36] has a rotor-mounted permanent magnet that has axially orientated flux as shown in *Figure 2.31*.

The excitation of each phase gives the flux paths as shown in *Figure 2.31*. The length of each step is simply related to the number of rotor teeth (*Figure 2.32*).



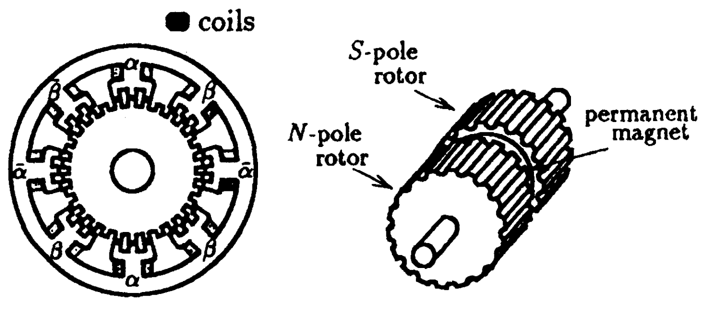
Cross-section of a hybrid motor parallel to the shaft

Figure 2.31 :- Hybrid stepper machine in cross section [36].



Cross-section of a hybrid motor perpendicular to the shaft

Figure 2.32 :- Ends of stepper machine [36].



Construction of two-phase hybrid stepping motor.

Figure 2.33 :- Hybrid stepping machine with axially orientated flux from the rotor magnets [37].

A two phase hybrid stepping machine has axially orientated flux permanent magnets on the rotor. Such a machine has many teeth on the rotor and stator (*Figure 2.33*)[37]. It was originally designed as an ac two phase synchronous machine for low speed applications so may theoretically be considered as a multi pole synchronous machine. One such machine has 8 stator poles, each with 5 teeth and has a rotor with 50 teeth. The rotor is magnetised axially by the permanent magnet such that one end is magnetically North and the other end is South. The teeth in the North pole rotor is one tooth pitch out from the teeth of the South pole rotor. By energising each of the two phases in turn, the rotor turns one step (defined increment of rotor position) at a time [36].

2.3.7 TRANSVERSE FLUX MACHINES

The transverse flux machine has permanent magnets and is capable of high torques per unit volume. The pole number can be raised without any increase in conductor volume. More information on transverse flux machines may be found in patents by Weh, Rolls Royce and Mitcham, such as, for example, EP0763880, EP0762618 (*Figure 2.34*), DE19507233, DE4400443, US5633551 and US5051641 (more examples may be found through patent searches). SMC's (Soft Magnetic Composites) allow high operating frequencies and enable the back iron to be made continuous [38].

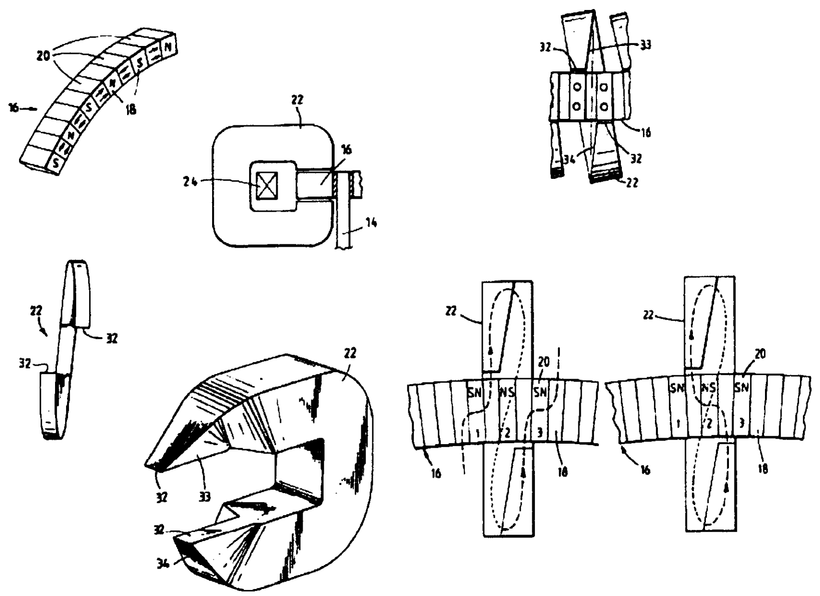
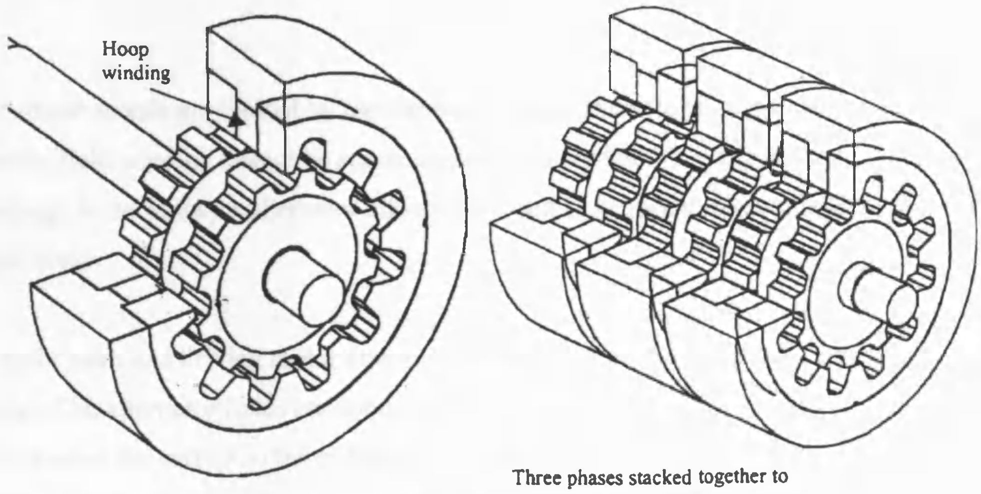


Figure 2.34 :- Examples of geometry of transverse flux machines based on EP0763880 and EP0762618.

Having many poles the machine is better suited to low-speed applications but direct drive is possible due to very high specific torques.

2.3.8 SWITCHED RELUCTANCE MACHINES WITH TOROIDAL PHASE WINDINGS (SINGLY EXCITED)

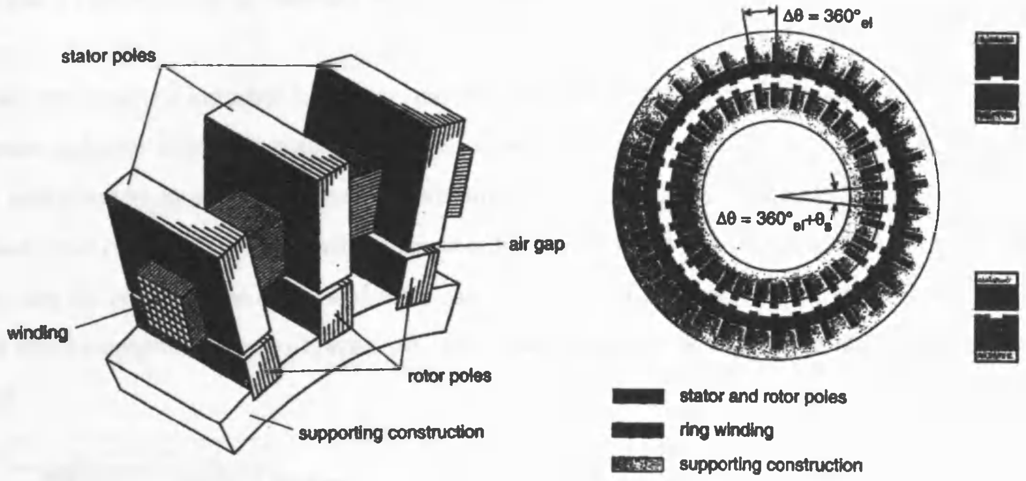
There has also been a recent development for Switched Reluctance Drives. An as yet unpublished paper reports the use of simple hoop windings to solve a problem of a reduction in mmf per armature coil as the pole number increases (which means that torque does not necessarily increase with an increase in pole number (*Figure 2.35*)[39]. The new topology allows an increase in torque with increase in pole number. It has separate isolated phases, each consisting of a simple hoop winding enclosed by a magnetic circuit. The rotor and stator have identical tooth numbers, so the permeance of the magnetic circuit linking the hoop coil varied with rotor position. Circumferential current in the hoop drives magnetic flux up one set of rotor teeth, across the air gap, radially up the set of stator teeth, axially across the stator core, then back through the other set of rotor and stator teeth to complete the magnetic circuit through an axial return path in the rotor. Since the hoop coil does not compete for space with the teeth, the pole number can be varied without affecting the coil mmf. Solid steel is suggested for the machine, as the lower speeds of operation would not cause significant eddy currents. A larger machine would require laminations but this would detrimentally affect the three dimensional flux paths of such a machine. It is suggested that powdered iron could be used as an alternative construction technique but the magnetic properties are not always favourable. A final design proposes use of laminations where the flux paths were laminar and powdered iron where the paths were three-dimensional. The laminations cannot be constructed as a complete ring. Instead a small gap is placed at one circumferential location in each phase. The gap prevents formation of a shorting ring that allows eddy currents to flow around circumferentially to oppose the mmf of each winding. The eddy currents would have impaired performance and caused major losses. The inclusion of the air gap prevents this. The multi phase machine comprises the phases being stacked axially but with the poles for each phase being offset from the next so that energisation of each phase in sequence causes the rotor to turn. This hybrid design could be regarded as a form of transverse flux switched reluctance machine. This design is very similar to that described in (*Figure 2.36*)[40]. The design of [40] (*Figure 2.36*) appears to be well suited for magnet insertion in the stator poles as the stator poles will have uni-directional flux present when the winding within the poles is energised.



One Phase of a hoop wound SRM.

Three phases stacked together to produce a multi-phase machine.

Figure 2.35 :- Hoop winding switched reluctance machine [39].



Segment of one motor phase

Structure of one phase of the TFR

Figure 2.36 :- Possibly the same hoop winding switched reluctance machine [40].

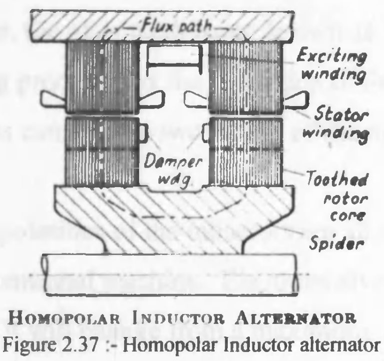
2.3.9 INDUCTOR ALTERNATOR

The inductor alternator was first developed to produce high-voltage AC at frequencies of 250Hz to 200kHz. The advent of high power radiotelegraphy meant that a source of high-frequency AC was needed. A later use of the inductor alternator was for supplying induction furnaces at 2000Hz for industrial high-frequency-heating loads [41, 42, 43]. Generating frequencies above 1000Hz (referred to in older books as cycles per second, c/s) requires both high speeds and large pole numbers [44]. The absence of conductors on the rotating part of the machine greatly simplified the problem of insulation. This type of alternator advantageously allows high pole numbers with high operating speeds.

Since peripheral speeds are limited by mechanical stresses, the pole pitch becomes uneconomically small and the field winding cannot be successfully accommodated in other designs. The absence of rotor windings in the inductor alternator allows the speed to be raised to the safe limit of rotor centrifugal stress.

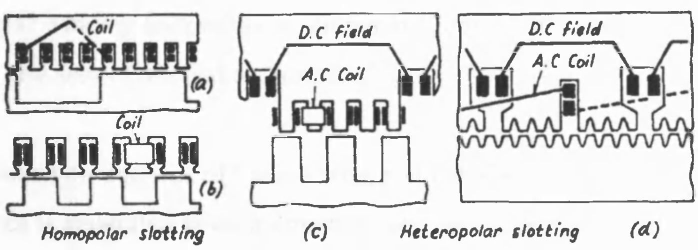
The alternator rotor and driving motor armature are mounted on the same shaft and mounted on the same frame. The alternator fields are connected in series with the motor armature, so that as the machine is loaded the current to the armature is increased, and so increases the excitation, compensating for the drop in voltage caused by armature reaction. The motor field has to be differentially compounded to keep the speed constant.

There are two types of inductor alternator, namely the homopolar and heteropolar variants. The homopolar inductor alternator machine has two stator and two rotor cores forming a single magnetic circuit, energised by an annular dc excited winding, *Figure 2.37*. The heteropolar type has a single stator and rotor core. The short shaft makes possible a very short gap, an important advantage in determining the output. The stator and rotor carry varieties of slotting, and the stator periphery is divided into heteropolar zones, *Figure 2.38*. Only the homopolar inductor alternator will be discussed in detail.



HOMOPOLAR INDUCTOR ALTERNATOR

Figure 2.37 :- Homopolar Inductor alternator [42].



SLOT ARRANGEMENTS FOR INDUCTOR ALTERNATORS

Figure 2.38 :- Slot arrangements for inductor alternator [42].

The homopolar and heteropolar inductor alternators have the same basic principle namely the pulsation or variation of the magnetic flux linking the stator coil system as a result of the cyclic changes of gap permeance round the stator bore. The permeance is most effectively varied by use of rectangular rotor teeth and a short air gap. One complete cycle of gap-permeance variation is produced through the movement of the rotor through one tooth pitch. The generated frequency, f_g, is independent of the stator tooth arrangement and is shown by (2.12)[42] :-

$$f_g = v.S_r \qquad \text{where} \quad v = \text{rotor speed (revolutions/second)} \qquad (2.12)$$
$$S_r = \text{number of rotor slots (or teeth)}$$

The stator coils should have a pitch of one-half the rotor pitch in order to fully utilise the flux variation. *Figures 2.38 (a)* and *(b)* show two arrangements for homopolar inductor alternators. One has a conventional winding: the other has coils round individual stator teeth. For comparison, a heteropolar winding arrangement is shown in *Figure 2.38 (c)*. Here the total working flux remains substantially constant so that no appreciable alternating EMF is induced in the dc exciting coils.

The rotor slot-pitch in an inductor alternator of rotor diameter D_r is (2.13) [42]

$$y_r = \pi D/S_r = \pi Dv/vS_r = u/f \qquad (2.13)$$

so that with the peripheral speed, u, limited mechanically by centrifugal stress, the slot-pitch must be very small for high frequencies. It then becomes necessary to use stator and rotor teeth of equal pitch (*Figure 2.38 (d)*), producing a pulsating flux; with the stator coils housed in larger slots of span equal to any suitable number of the smaller teeth.

The homopolar inductor alternator is usually symmetrical about the central field coil winding (*Figure 2.39*) [44, 45], the stator on either side being slotted and the armature winding consisting of coils wound around each tooth (labelled as 'a-c output winding'). The rotor has half as many slots as the stator, the rotor teeth being known as 'inductors' (*Figure 2.39*) [46]. The magnetic flux paths flux, ϕ, being produced by the excitation of the field winding follows those shown in *Figure 2.40*. Cast steel yokes connect the two halves of the rotor and stator (labelled as 'housing').

The polarities of the inductors are all the same on one side of the machine and not alternate as in the conventional machine. Electromotive force is generated in each armature turn since the flux linking with it will change from a maximum, as an inductor faces the tooth to a minimum, as the pole passes by and a slot takes its place. The gap permeance fluctuates as the stator and rotor teeth move in and out of alignment.

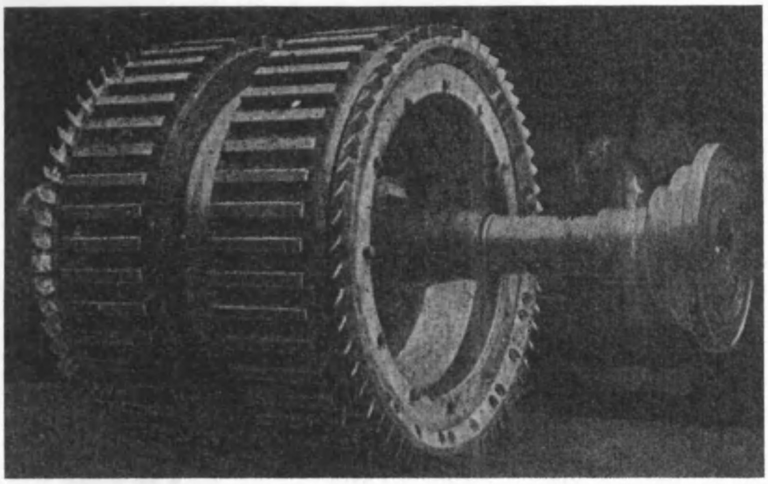
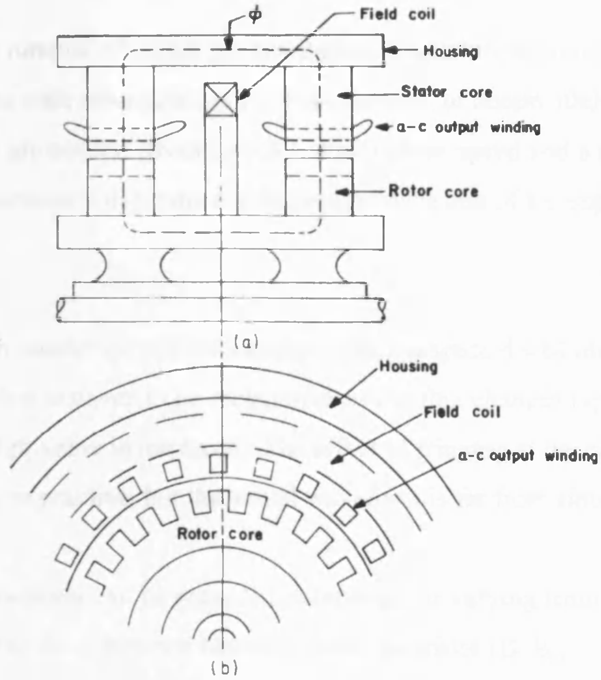
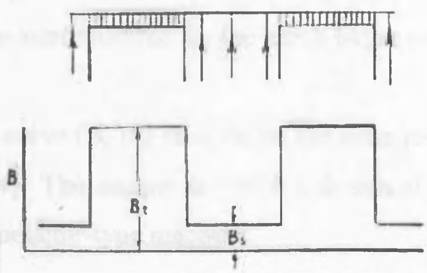


Figure 2.39 :- Inductor alternator rotor [46].



Homopolar type inductor alternator

Figure 2.40 :- Winding arrangement for inductor alternator [44, 45, 46].



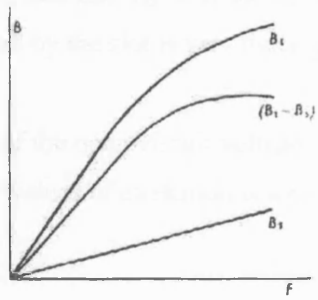


Figure 2.41 :- Flux waveform for inductor alternator [44].

Figure 2.42 :- Difference curve for inductor alternator [44].

	Alternator 1	Alternator 2	Alternator 3	Alternator 4
Kva	587	520	95	30
Power factor, %	95	80	80	50
Frequency, Hz	3,200	3,200	3,200	3,200
Speed, rpm	24,000	39,000	39,000	6,000
Number of poles	16	10	10	64
Subtransient reactance per unit		0.50		
Synchronous reactance per unit	0.87	1.20	1.67	0.24
Leakage reactance per unit	0.35	0.32	0.28	
Efficiency, %	90	90	87	
Harmonic content, %				
Line-to-neutral, no load	10.3	3.5	5.5	18.0
rated load		5.5	8.3	21.0
Line-to-line, no load	1.6	4.6	2.0	0.5
rated load		6.8	1.1	0.5
Length of stator stack, in.	4.0	3.06	2.4	2.5
Number of stator stacks	4	4	2	2
Stator punching O.D., in.	11.7	10.27	7.96	18.5
I.D., in.	9.5	7.65	6.0	15.0
Single air gap, in.	0.125	0.12	0.08	0.06
Number of stator slots	72	60	60	108
Housing length, in.	36.7	29.5	14.0	
Diameter, in.	18.3	16.5	10.0	
Weight, lb	975	630	115	

Courtesy of Department of the Army, U.S. Army Mobility Equipment Research and Development Center, Fort Belvoir, Va.

Figure 2.43 :- Homopolar inductor alternator data [46].

It should be noted that the number of cycles per revolution is equal to the number of rotor projections and not half this number as with other generators. For example, to obtain 10kHz at 3000rpm (= 50rev/sec), 200 rotor slots are needed (from f_g=v.S_r). For a given speed and a number of rotor 'poles' (or teeth), the inductor alternator will produce a frequency twice that of a comparable synchronous machine.

The emf waveform of each conductor will be identical with the space distribution of flux density B. This would appear in the first instance to be rectangular as the flux changes rapidly from the low value in the slot portion to the high value in the tooth. The effect of fringing at the edges of the rotor teeth modifies this considerably in practice, but the actual waveform is far from sinusoidal.

For simplicity, the flux waveform can be regarded as rectangular varying from B_t to B_s (*Figure 2.41* [44]). The coil emf is due to the difference between these quantities (B_t-B_s).

As the excitation F is increased, B_t will follow a normal type of magnetisation curve to be expected from a small gap with an associated iron circuit, but B_s will increase linearly over the same excitation range, since the mmf required for the much larger gap presented by the slot is very much greater.

The difference curve (B_t-B_s) thus shows the anticipated shape of the open-circuit voltage characteristic *Figure 2.42* [44]. The manner in which it droops at the higher values of excitation is a particular feature of the inductor-type machine.

Some data for military application homopolar inductor alternators is given in *Figure 2.43* [47]. It should be noted that the homopolar inductor alternators are highly efficient and operate at very high speeds. It should also be noted that they tend to be large in diameter and have high pole numbers.

The homopolar inductor alternator is said to be more expensive to construct as it differs considerably from other standard types of electrical machine. The heavy yokes and separate cores add to their cost. The central field coil magnetises the shaft and bearing troubles are prevalent. The rotor has an abnormally large moment of inertia that imposes a heavy starting duty on the driving motor or turbine. There is a large time constant of 5 to 20 seconds of the field system since all flux is linked with a single coil. It may be difficulty to maintain constant voltage under fluctuating load conditions. The design of the heteropolar inductor alternator was to overcome these disadvantages. In one type of heteropolar inductor alternator, the field coils are in slots in the stator and are no longer concentric with the rotor shaft. The Lundell alternator (*Figure 2.44*)[48, 49], a synchronous claw pole alternator, has discs with protruding claws which are mounted on a hub so that the claws intermesh. A field coil wound (inside the claws) around the hub, producing North poles on the claws of one disc and South poles on the other disc, resulting in alternate magnetic poles on alternate claws. *Figure 2.44* shows an example of the Lundell (claw pole) alternator and its principle of operation.

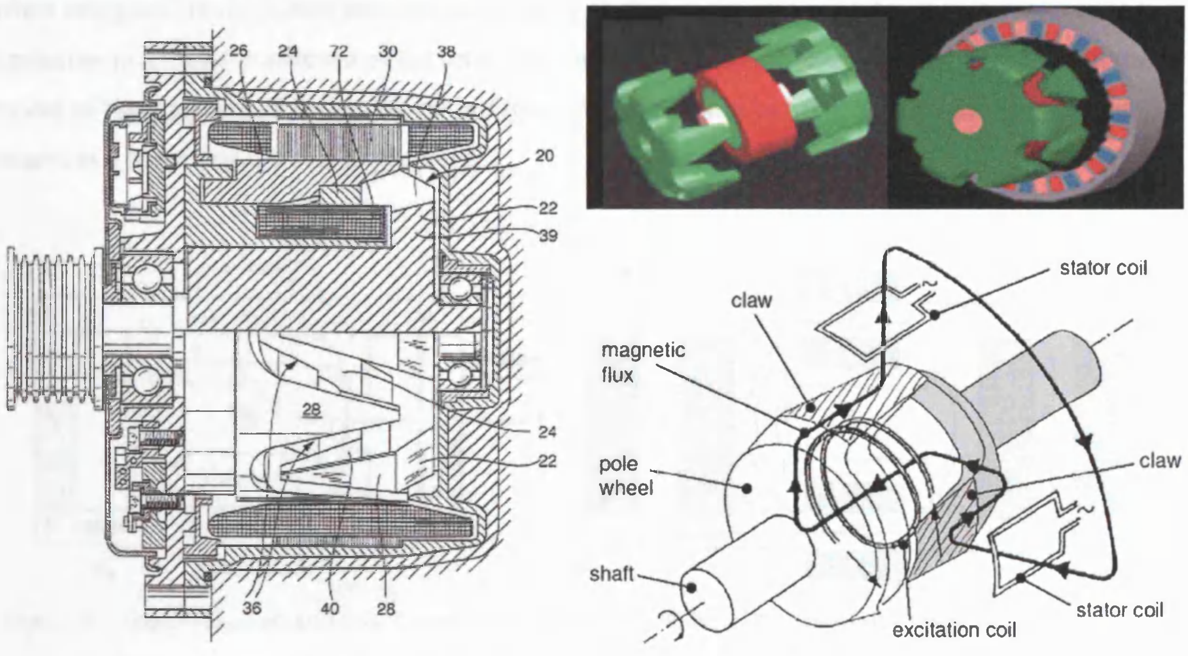


Figure 2.44 :- Lundell (claw pole) alternator and its principle of operation [48, 49].

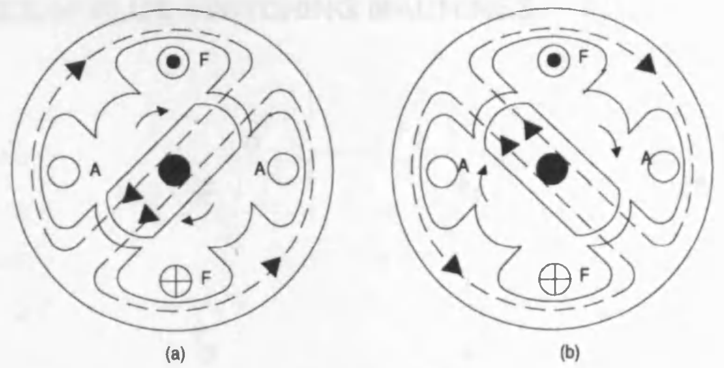


Figure 2.45 :- Inductor Alternator based on a Switched Reluctance lamination [50].

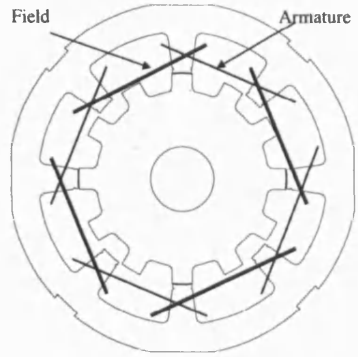


Figure 2.46 :- 8/12 Inductor Alternator [50].

Research exists demonstrating the operation of a Switched Reluctance lamination generator using an inductor alternator based topology (*Figures 2.45, 2.46*)[50].

2.3.10 LAW'S RELAY LIMITED MOTION ACTUATOR

The reluctance principles as found in inductor alternators and hybrid stepper machines is also present in the Law's Relay Limited Motion Actuator [51]. It has dc field excitation in permanent magnet or field winding forms as shown in *Figure 2.47* and *Figure 2.48* respectively. The armature winding, when energised, (a dc current that changes polarity to change direction of rotor movement) results in a limitation in angular movement of the rotor. Such a machine may be modelled by aid of a resistive model of the magnetic circuit as shown in *Figure 2.49*. It should be noted that the use of the magnets results in a stator that is not a single piece.

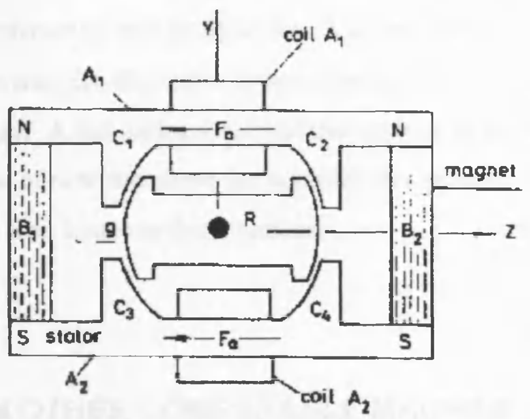


Figure 2.47 :- Magnet version of Law's Relay Limited Motion Actuator [51].

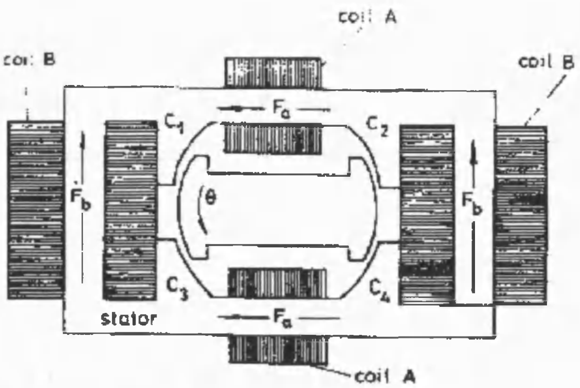


Figure 2.48 :- Field winding version of Law's Relay Limited Motion Actuator [51].

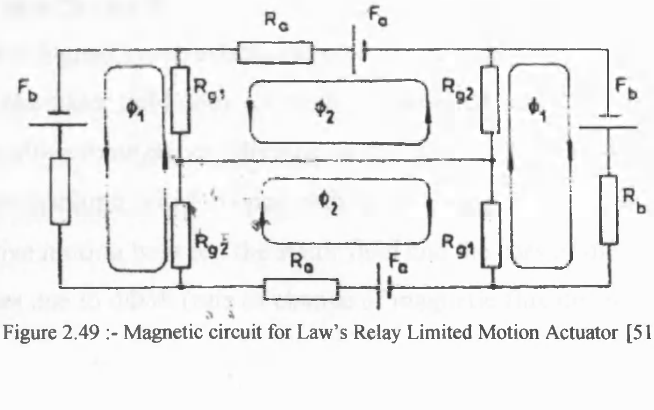


Figure 2.49 :- Magnetic circuit for Law's Relay Limited Motion Actuator [51].

2.3.11 FLUX SWITCHING MACHINES

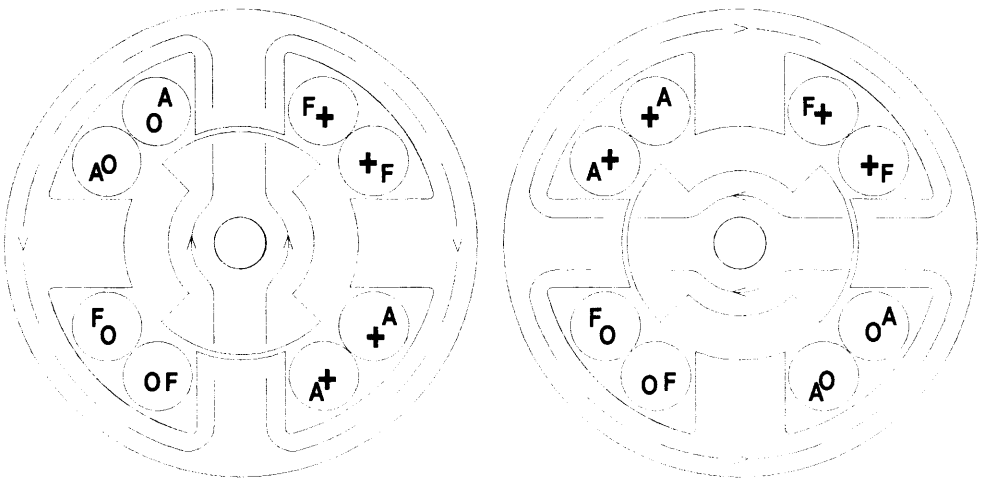


Figure 2.50 :- 4/2 flux switching machine - the two aligned positions achieved by switching armature current polarity (left: field positive, armature positive, right: field positive, armature negative).

The Flux Switching Machine (FSM) combines the switched reluctance machine and the inductor alternator. It exhibits machine characteristics similar to those of a D.C. machine but it uses the reluctance principle for rotation. The machine windings are fully pitched. The field and armature windings shown in *Figure 2.50* have a conventional dot and cross notation to denote positive and negative currents. The field winding carries dc current at all times. The armature current carries bipolar current (the polarity changes with each phase for the two-phase 4/2 machine shown). When the armature current is positive the top and bottom stator pole pair is energised which pulls the rotor into vertical alignment as shown. When the armature current is negative the stator poles on the left and right are energised which pull the rotor into the horizontal position. The field winding is permanently energised so flux is always present. The armature can be regarded as orientating the flux between the flux paths shown causing the rotor to attract and align with one stator pole pair or the other. A full explanation and background to the new flux switching technology is given in [52]. It is also of note that there are unipolar flux sections in the stator behind the field winding which reduces the iron losses in those sections.

2.4 OTHER COMPARABLE MACHINE CLASSES OF RELEVANCE

2.4.1 INDUCTION MACHINES

Induction machines have rugged construction, maintenance-free operation, and low cost [53]. In a single-phase machine, the stator field does not rotate, but alternates poles as the single sinewave voltage swings from positive to negative. Because of this, the stator field remains lined up in one direction with the poles changing position once each cycle. Since the stator magnetic field does not rotate, there is no relative motion between the stator field and the bars of the rotor. A voltage is induced in the rotor bars due to $d\Phi/dt$ (rate of change of magnetic flux due to sinusoidal current).

Since the bars on the rotor are short-circuited, current flows in the rotor, but the magnetic field this current produces is lined up with the stator field so no net torque is produced on the rotor. A single-phase induction machine has no means for starting itself but will run once it has been started. The double revolving-field theory of single-phase induction machines states that a stationary pulsating magnetic field can be resolved into two rotating magnetic fields, each rotating in opposite directions and each of equal magnitude. The induction machine responds to each magnetic field separately and the net torque of the machine is the sum of the torques due to each of the two magnetic fields [53]. If power is applied to a three-phase machine while it is forced to turn backwards, its rotor currents will be very high. However, the rotor frequency is also very high, making the rotor's reactance much larger than its resistance. Since the rotor's reactance is very high, the rotor current lags the rotor voltage by nearly 90 degrees, producing a magnetic field that is nearly 180 degrees from the stator magnetic field. The induced torque in the machine is proportional to the sine of the sine of the angle between the two fields, and the sine of an angle near 180 degrees is a very small number. The machine's torque would be very small, except that the extremely high rotor currents partially offset the effect of the magnetic field angles. In the single-phase machine, the forward and reverse magnetic fields are produced by the same current, each contributing a component to the total voltage in the stator. The forward rotating magnetic field, having a high effective rotor resistance, limits stator current flow in the machine. The current supplying the reverse stator magnetic field is limited to a small value and, since the reverse rotor magnetic field is at a very large angle with respect to the reverse stator magnetic field, the torque due to the reverse magnetic fields is very small near synchronous speed.

However, most of these single-phase induction machines use some sort of automatic starting and are often referred to by the starting method used. There are three basic types, namely, shaded-pole, split-phase induction, and split-phase capacitor start machines [53].

A three-phase induction machine requires a revolving stator field derived from the three-phase power source. The induction machine uses no slip rings or commutator assembly. Instead, current is induced in the rotor by the cutting of electromagnetic flux lines (from the rotating stator field) across the rotor windings. This voltage induced in the rotor causes a current to flow which sets up an electromagnetic field in the rotor. The induction machine generates its own rotor current.

Induction machines can never operate at synchronous speed. If the rotor were to turn at the same speed as the rotating field, no lines of force would be cut by the rotor conductors, and no rotor field would be produced. An induction machine rotates at something less than synchronous speed. The difference between the synchronous speed and the actual rotor rpm is called slip. The full load slip in most induction machines varies between 4 and 6 percent. Should an induction machine become heavily overloaded or stalled (the slip would be 100%), the rotor would be damaged. In general, slip in an induction machine should not exceed 10%.

2.4.1.1 SHADED-POLE MACHINES

A shaded-pole machine uses a small shorted shading coil wound in a small notch in the stator pole piece [53]. In some machines this coil consists of a single copper ring or copper band. When the electromagnetic field builds up around the main coil, the flux cuts across the conductors of the shading coil. Since the shading coil is shorted, current flows which produces a field opposite that of main field. The main field is then strongest on the side away from the shading coil. However, the field through the shading coil reaches maximum intensity much later - at a time when the main field is already decreasing. The electromagnetic field in the pole piece then appears to be stronger on the side nearest (or through) the shading coil. This produces a sweeping motion from side to side in the stator pole piece. The small motion of the field back and forth between shaded and unshaded portions produces a weak torque to start the machine. Because of the weak starting torque, shaded-pole machines are built only in small sizes.

An alternative machine topology has stepped poles as well to provide a reluctance augmented start.

2.4.2 BRUSHLESS DC MACHINE

DC machines [53] operate because of the interaction between magnetic and electromagnetic fields. The stationary field is not always a permanent magnetic field. It may be supplied by an electromagnetic field, as with a dc generator. The field windings are excited from a dc power supply, and provide a steady electromagnetic field. The polarity of the field is determined by the direction of the current flow through the windings. However, a machine also requires dc excitation to the armature. This could be supplied by a separate dc power supply, but often the field windings (stator) and the armature assembly (rotor) are supplied from the same source.

There are three types of dc machines: series, shunt, and compound. This refers to the method in which the field coils and the armature coils are connected. The torque produced by a dc machine is proportional to the armature and field currents.

In a direct current (DC) machine [53], a device known as a split ring commutator switches the direction of the current each half rotation to maintain the same direction of motion of the shaft. The speed of direct-current machines is controlled by varying the field or armature voltage. Brushed DC machines have built-in commutation, meaning that as the machine rotates, mechanical brushes automatically commutate coils on the rotor. Brushless DC machines use an external power drive to allow commutation of the coils on the stator. Brush-type machines are used when cost is a priority, while brushless machines are selected for specific requirements, such as maintenance-free operation, high speeds, and hazardous environments where sparking could be dangerous. The rotor contains a permanent magnet and the stator has the conducting coil of wire. Brushless machines offer reduced maintenance, no spark hazard, and better speed control. They are widely used in computer disk drives, tape recorders, CD drives, and other electronic devices.

A series dc machine is one in which the field coil and the armature coil are in series. Since the starting current through the armature also flows through the field coils, the series machine develops a high starting torque, but has poor speed regulation. As the speed decreases, the amount of torque delivered to the load increases because, as the armature slows down, the emf developed in the armature decreases, increasing the current through the armature and field, increasing the torque. Increasing the current through the field coils and the armature increases their fields which increases the torque. A series machine runs fast with a light load, and runs slower as the load is increased. If a series machine is allowed to run without a load, it may run so fast that the armature will fly apart. It is for this reason that series machines should not be used with belt coupling to the load. If the belt were to break, the machine would be without a load, and could destroy itself.

Shunt dc machines are connected with the field windings and the armature windings in parallel. The field windings carry only part of the excitation current, and are made of smaller-gauge wire than those of a series machine. A shunt machine has good speed regulation, but low starting torque. An increase in load torque to a shunt machine slows the machine down, causing a reduction in generated back emf, which results in an increase in armature current that causes an increase in torque to match the load torque at a lower speed. By adding a variable resistance to the field coil circuit, machine speed can be controlled above its nominal value. Increasing the field resistance decreases the field current (field weakening) and the emf induced in the armature. This causes greater current to flow through the armature, increasing the torque and the machine will speed up. Decreasing the field resistance causes the field current to increase, increasing the emf developed in the armature, thus reducing armature current. The reduced current produces less torque and the machine slows down. Thus, the speed of a shunt dc machine can be relatively independent of the load.

A separately excited dc machine has a field circuit supplied from a separate constant voltage power supply. When the supply voltage to each machine is assumed constant, there is no practical difference between the separately excited and shunt dc machines.

A compound machine uses both series and shunt field windings. The advantage of this arrangement is that the compound machine can be made to produce a variety of operating characteristics. By adjusting the polarity and placement of the field windings with respect to each other, a number of machine types can be designed. For example, when the series and shunt fields are in series-aiding, the machine is a cumulative compound machine. When the fields are series-opposing, the machine is a differential compound machine. Another factor in the design of a compound machine is the placement of the series winding. When the shunt winding is connected directly across the armature, the machine is a short shunt compound machine. When the shunt winding is connected in parallel with the armature and the series winding, the machine is a long shunt compound machine. By controlling the field strengths around the coils using cumulative/differential, or short/long shunt methods, a compound machine can produce any characteristic that can be produced by a pure series or shunt machine. A compound machine can be safely operated without a load and can have the speed characteristics of a shunt machine and the starting torque characteristics of the series machine.

Because the dc resistance of most machine armatures is low (0.05 to 0.5 ohm), and because the back emf does not exist until the armature begins to turn, it is necessary to use an external starting resistance in series with the armature of a dc machine to keep the initial armature current to a safe value. As the armature begins to turn, the back emf increases and, since the back emf opposes the applied voltage, the armature current is reduced.

2.4.3 FLUX SWITCH GENERATOR

A NASA report [54] shows different types of homopolar and heteropolar generator. Page 68 [54] shows a flux-switch generator that is a simple version of the heteropolar inductor alternator (*Figure 2.51*). It has two alnico magnets as the source of flux and two steel stator sections. On page 69 [54] is shown an electromagnetic flux-switch alternator, but it is only used where low outputs and waveform shape is unimportant. As is clearly shown in *Figure 2.52*, there are sections in the stator back-iron where flux is uni-directional and sections where the flux is bi-directional.

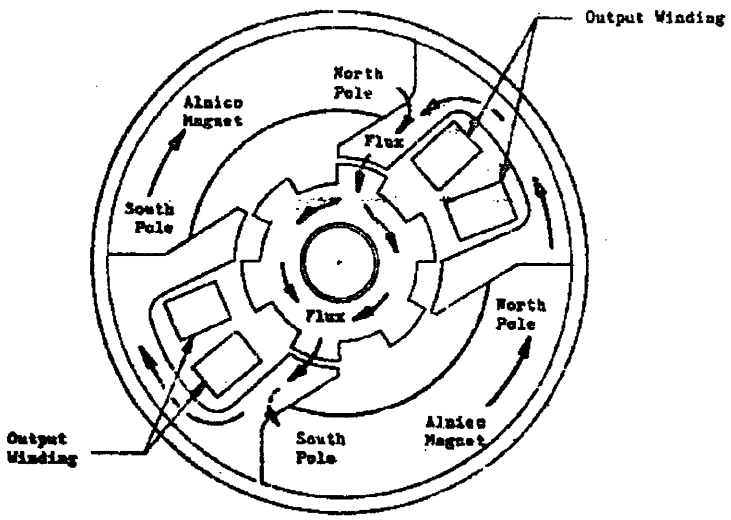


Figure 2.51 :- Flux switch generator [54].

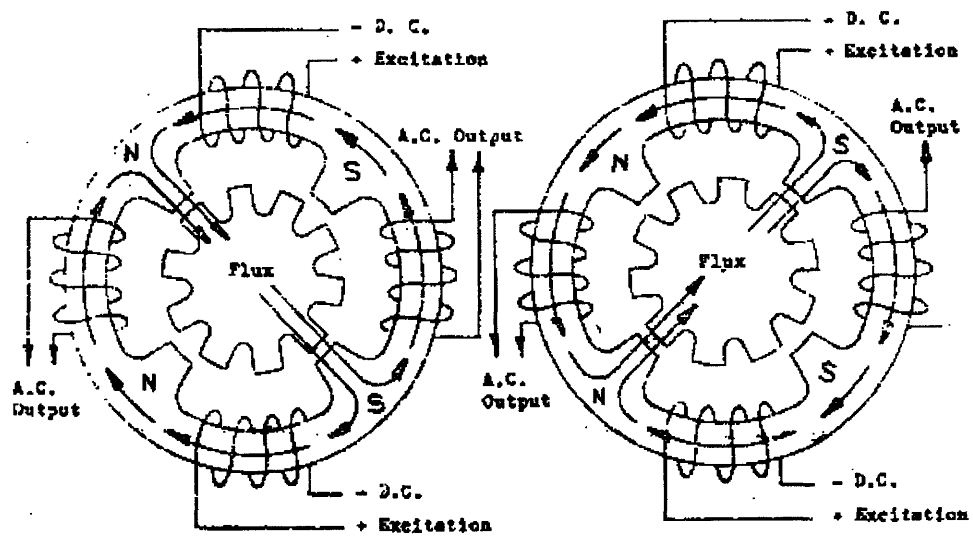


Figure 2.52 :- Flux paths in electromagnetic flux-switch alternator [54].

2.5 RELUCTANCE MACHINE DESIGN ISSUES

Issues such as cogging torque, acoustic noise cancellation techniques, powdered iron technology and fan industry requirements are discussed in detail in Appendix A.

2.6 SUMMARY

There are a wide variety of generator and motor topologies that utilize field windings and permanent magnets to give desirable benefits in machine design. Some machines actually use both permanent magnets and field windings together for improved performance capability (e.g. Doubly Salient Permanent Magnet Motor with Flux Control (Figure 2.18)[8, 25]).

The homopolar inductor alternator has a common rotor and a field winding placed between two stator stacks. A back iron links the stator stacks. The homopolar inductor alternator has high pole numbers, allowing generation of high frequency ac voltages in the stator windings. The homopolar inductor alternator geometry can be used in novel variable reluctance motors with flux assistance from toroidal field windings. This is the basis of two machine topologies being used in this thesis (the Dual Stack Variable Reluctance Machine and the Single Stack Variable Reluctance Machine as described in Chapters 4 and 5 respectively).

Although the use of field windings allows production of constant flux, this requires input power. Where flux in a machine is known to be unidirectional, there is the possibility of replacing the steel in those sections with permanent magnets. This would be at the expense of increased reluctance of the magnetic circuit, but the magnets, having replaced the field windings would allow a reduction in total input power. The flux switching motor is an ideal candidate for magnet insertion as there are sections in the back-iron where magnetic flux is uni-directional. A Flux Switching Motor (FSM) is reviewed for the third machine topology (the Permanent Magnet Flux Switching Motor as described in Chapter 6).

When designing motors with magnets, the choice of magnet material, its size and placement all affect machine performance. The same may be said for field windings. The additional flux from either can be utilised to reduce switching losses and to improve efficiencies. It is shown that magnets may be inserted on or even inside the rotor or be inserted in the stator. The orientation of flux from the magnets can be radial or axial. This does cause some complications regarding the flux paths within the machines. This has been overcome in newer designs by using improving magnetic capabilities of the modern generations of powdered iron and its alloys.

Axial flux from the toroidal field windings and flux from magnets can be implemented in novel geometries to be shown in this thesis in Chapters 4, 5 and 6. All the geometries will be shown to have potential for improving torque production and lowering power electronic rating requirements for the armature windings. Low input power, high efficiency designs may therefore be attainable.

The chosen lamination shapes of the motors affects the acoustic characteristics of the motors during operation. Cogging torque may also be affected by the motor geometry and is of particular interest when constant flux sources are used. It is shown that altering the motor geometry, e.g. by skewing the rotor or stator stack, and improvements in the control algorithms may reduce the effects of cogging torque. Improving the commutation of the armature currents may result in quieter operation of these motors.

CHAPTER 3

EXPERIMENTAL POWER AND CONTROL ELECTRONICS

3.1 INTRODUCTION

The machines developed during this thesis (discussed in Chapters 4, 5 and 6) require appropriate selection of power and control electronics to operate them. The selection is dependent on how the windings within each machine are connected and energised.

Figure 3.1 illustrates the field and armature winding topologies with associated power electronic circuitry that exist for reluctance machines with dc magnetic flux assistance. A discussion into their merits is discussed in [3, 92-94].

A field coil energised with dc current requires a simple dc power source. Where the field is due to a permanent magnet, a cause for concern is that of back emf in the phase (armature) windings as the rotor turns but with no armature excitation due to this permanent source of flux.

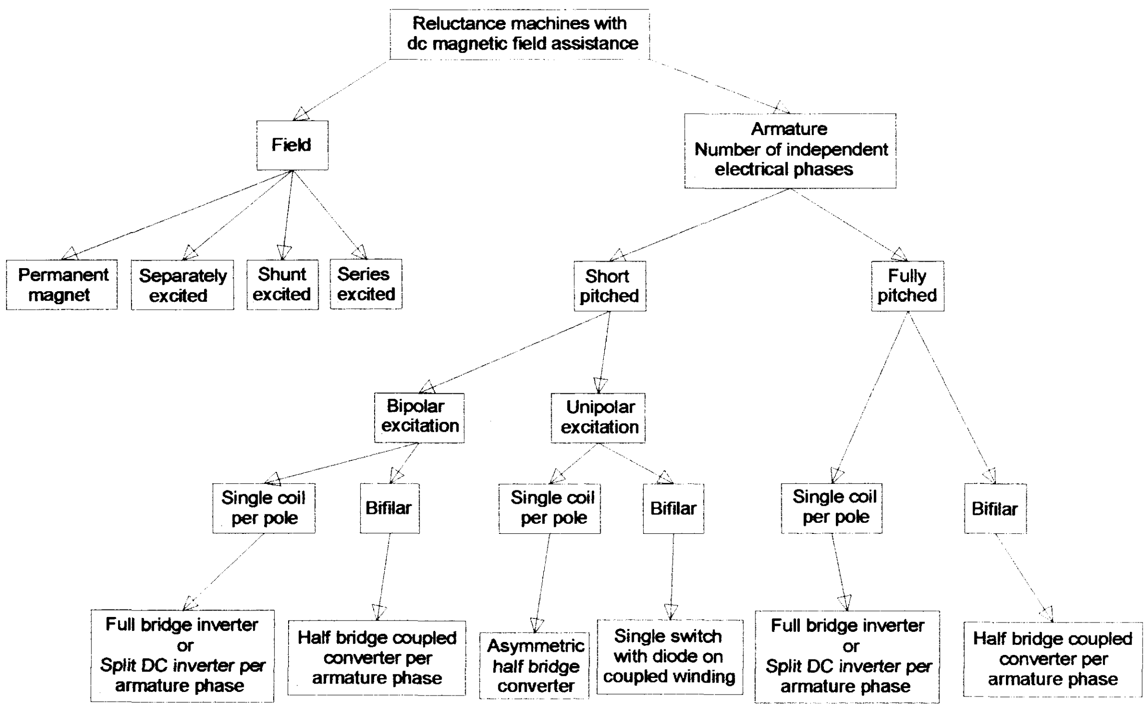


Figure 3.1 :- Field and armature winding topologies with associated power electronic circuitry for reluctance machines.

The phase windings (armature) and field windings are energised separately or together in series. The field may be from either a permanent magnet or from an energised field winding. The armature windings may be short pitched or fully pitched (*Figures 3.2* and *3.3*). The armature windings and field windings are chosen to be separately energised as this allows greater flexibility in testing by being able to vary the supply to the winding types.

Switched reluctance and stepping motors have developed around the concept of short pitching each phase winding, generally around a single stator tooth (*Figure 3.2a*). By employing such a winding torque is derived due to the rate of change of self inductance of the excited phases. Utilisation of the electric circuit is poor, due to the fact that each phase winding is restricted to periods when the self inductance is rising, so that any one winding can only contribute to positive torque for a maximum of one half of the time. Alternatively, if the machine is wound with a fully pitched winding (*Figure 3.2b*) then it can derive virtually all its torque from a changing mutual inductance between phases. The machine therefore becomes a dual of the conventionally wound machine, which operates entirely upon rate of change of self inductance. Because of the fundamental change in operating mode, each phase of the machine can contribute to positive torque production for considerably greater than one half of the cycle of rotation, leading to a more efficient utilisation of the winding.

Short pitched armature windings (*Figure 3.2a*) are energised in either a bipolar or unipolar manner due to the choice of power electronics, which is illustrated in *Figure 3.1*. Short pitched coils are simply wound around each stator pole.

Bipolar and unipolar excitation (the names referring to the current flow within the copper windings) is achieved through the armature windings being wound either bifilar or single coil per pole.

Fully pitched armature windings are achieved through the windings being wound either bifilar or single coil per pole. The copper spans across more than one stator pole in fully pitched windings.

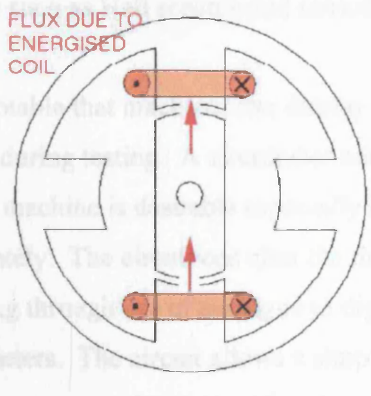


Figure 3.2a :- Short pitched winding for one phase for a Switched Reluctance Machine.

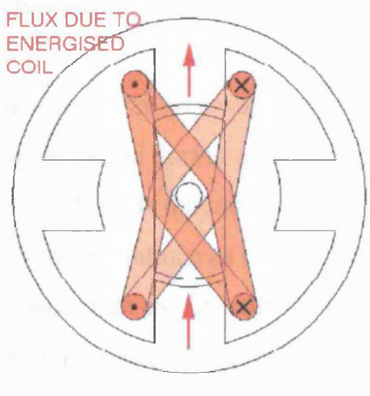


Figure 3.2b :- Fully pitched winding for one phase for a Switched Reluctance Machine.

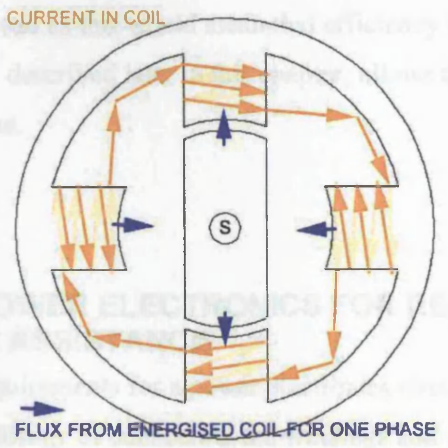


Figure 3.3a :- Short pitched windings with bipolar excitation which are bifilar (only one phase shown).

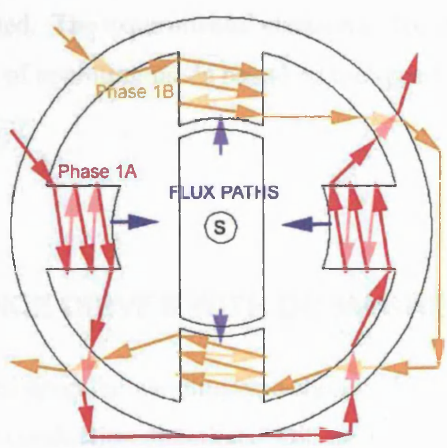


Figure 3.3a :- Short pitched windings with unipolar excitation which are single coil per pole.

The choice of fully or short pitched windings, the method of current flow within the windings and the winding types determine which power electronic circuits may be used. The appropriate selection of power electronics is needed for best performance from the machine windings. An inappropriate selection could reduce the overall performance or even, at worst, result in circuitry and/or machine failure. *Figures 3.3a* and *3.3b* show how the coils on the stator poles may be connected for short pitched windings with bipolar excitation which are bifilar and short pitched windings with unipolar excitation which are single coil per pole. The figures show current flow and flux paths when the coils are energised.

The control electronics manages the timings of switching the armature windings on and off by feeding control signals to the power electronics. This chapter introduces a novel experimental prototyping circuit for developing the timing characteristics for a Switched Reluctance Machine, Flux Switching Machine or a variant of either. It will control a two-phase machine with either one or two switches per phase [3, 92-94]. The position is determined via an opto sensor that changes state when the rotor is aligned with a stator pole pair (the point in back-emf testing when the back-emf changes polarity). Opto sensors are used for simplicity of rotor position detection, although other rotor position detection means such as Hall sensors and sensorless control methods are alternatives.

It is notable that machines can display different operational characteristics as they warm up and cool down during testing. A circuit that allows the timing sequences to be varied during continued testing of the machine is desirable especially if the final timing sequence needs to be determined quickly and accurately. The circuit can alter the timing sequences of the armature excitations as the machine is running through use of analogue to digital converters to provide values that can be used as time delay parameters. The circuit allows a simple, fast and reliable optimisation process for meeting machine performance requirements. The aim of the circuit is to achieve minimisation of input power for a given load as this would mean that efficiency is maximised. The experimental electronics for this circuit, described later in this chapter, allows three types of operating mode based on the speed of the machine.

3.2 POWER ELECTRONICS FOR RELUCTANCE DRIVES WITH DC MAGNETIC FLUX ASSISTANCE

The requirements for a power electronics circuit are based upon the winding type chosen, the methodology of connecting the windings and the current conduction directions. Bifilar windings use the transformer effect to transfer current quickly from one phase to the next since the windings of each phase are wound round each other. Inverter circuits use a dc supply to give ac supply to the windings. Converter circuits use a dc supply to give a pulsed dc supply to the windings. [3, 94-96].

The machines described in Chapters 4, 5 and 6 require suitable power electronic circuits depending on the selection of winding type and armature energisation required for desired flux production for each machine; the types of power electronic circuits appropriate for each selection being detailed below.

3.2.1 SHORT PITCHED WINDINGS WITH BIPOLAR EXCITATION

The requirement for bipolar current in short pitched phase windings requires the use of an inverter. A single-phase full (H) bridge inverter requires four power switches two of which have floating gate drives (*Figure 3.4*). Alternatively a split dc inverter circuit can be used which requires two switches one of which would have a floating gate drive (*Figure 3.5*).

3.2.1.1 SHORT PITCHED WINDINGS WITH BIPOLAR EXCITATION WHICH ARE SINGLE COIL PER POLE

One circuit used is the split dc inverter (*Figure 3.5*). The short pitched winding armature coils can be in series producing a single coil per pole configuration such that both ‘phases’ are energised simultaneously as single phase with bipolar current (*Figure 3.6*).

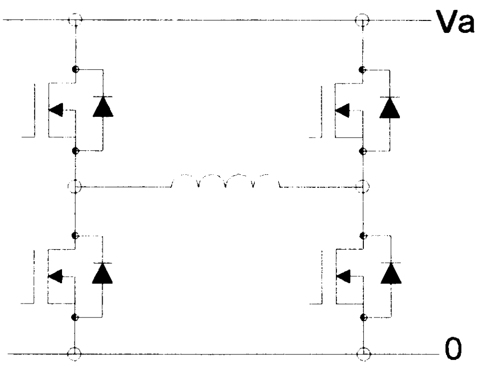


Figure 3.4 :- Single phase full bridge inverter circuit.

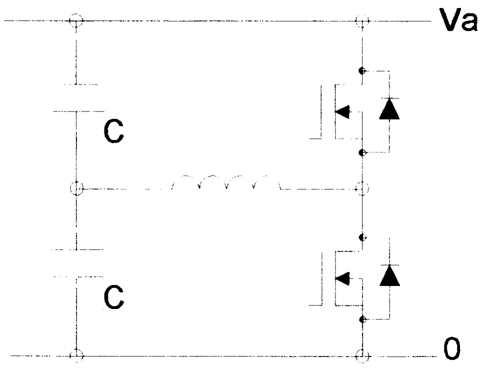


Figure 3.5 :- Split dc inverter circuit.

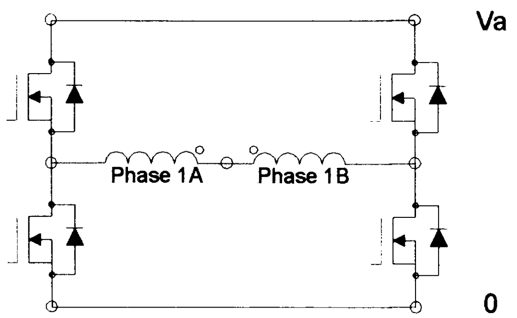


Figure 3.6 :- H-bridge inverter circuit to use all bifilar windings as single coil per pole (at the same time) with bipolar current.

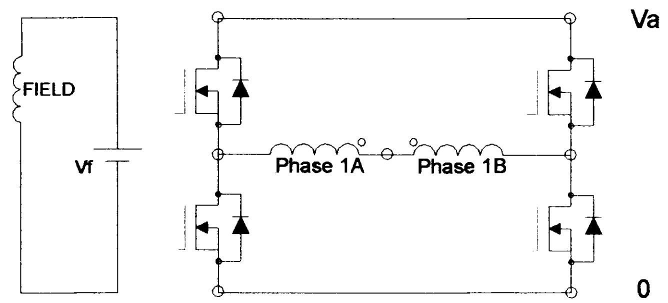


Figure 3.7 :- H-bridge inverter circuit to use all bifilar windings as single coil per pole (at the same time) with bipolar current.

The other possible circuit is a full or H-bridge circuit that requires two floating switches and two ground referenced switches.

For the same ampere turns from the armature windings, half the current, I, would be needed (as the number of turns has doubled with this method). The resistance, R, has doubled due to twice the turns being used. The resultant mechanical output power would theoretically be the same as the current flows for twice the time every cycle (3.1). Using this circuit allows the ampere turns to be potentially doubled if the windings allow the increased copper losses. If the windings and power electronics allow for increased supply then it may be possible for more ampere turns to be utilised up to double the ampere turns, hence more torque production is possible. This circuit uses all available armature copper.

$$2.(0.5I).(0.5I).(2R) = I^2R \tag{3.1}$$

The full bridge inverter circuit has twice the number of switches, two of which are floating gate drives, increasing the overall circuit complexity. True bipolar current is driven through the armature windings. Due to concerns regarding the thermal loading (the maximum ampere turns in the armature at any one time has doubled with this circuit) and difficulty of building such a circuit, this circuit was not investigated any further as the bifilar winding topology with half bridge converter circuit would give similar results with a much simpler circuit. The full bridge inverter circuit also demonstrates that a machine with bifilar windings is essentially single phase. It is single phase because, if only one coil set (1A or 1B) is energised, the machine operates as per normal. When operating, half the time the current is positive in sense and the remaining time the current is negative. All the windings share the same flux linkage at all times and it is possible that one set of phase windings energised with bipolar current will operate the machine. Thus it is correct to label each set of phase windings as Phase 1A and Phase 1B (*Figures 3.6, 3.7*).

3.2.1.2 SHORT PITCHED WINDINGS WITH BIPOLAR EXCITATION WHICH ARE BIFILAR

Bifilar phase windings allow the use of the half-bridge converter circuit. This is shown in *Figure 3.8* in which the toroidal field coil is separately excited from a dc power source. Phase 1A and Phase 1B are formed by series connection of coils positioned on all of the four stator poles (such that the back emf from each coil summate). This drive can produce torque from all rotor positions through simultaneous energisation of all stator poles of the switched reluctance stator. Only two ground referenced switches are required for this circuit.

An alternative circuit is shown in *Figure 3.9*. This circuit is based on a current source half bridge converter. The current source inductor is the toroidal field winding. Testing the machine with the circuit in *Figure 3.8* allows the field ampere turns to be varied independently but a series connection of *Figure 3.9* could be used if the desired field ampere turns were achievable with the armature currents.

The labels Phase 1A and Phase 1B refer to the two bifilar windings used for positive and negative armature current. They are each made up of coils on each stator pole. The circuits in *Figures 3.8, 3.9* are for bipolar flux in the stator poles. The bifilar windings offer a transformer effect for faster transfer of current from one phase to the next during switching. When the supply to the armature is switched off (i.e. MOSFET, IGBT or transistor is turned off), the current transfers from that armature winding to the secondary winding (as would happen in a transformer, due to the windings being linked magnetically by mutual inductance) due to the windings being bifilar wound. The current then freewheels through the diode of the secondary windings.

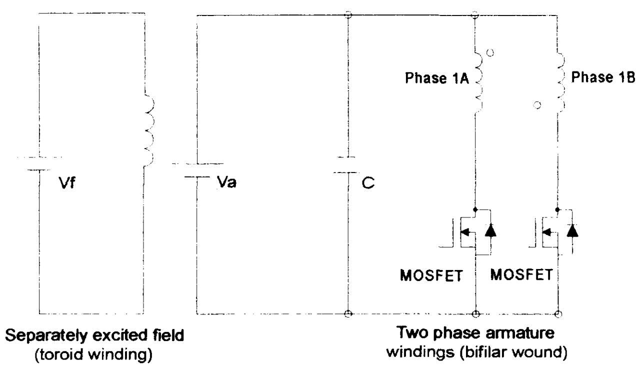


Figure 3.8 :- Circuit used for bifilar windings for coupled windings, bipolar excitation (half bridge converter circuit).

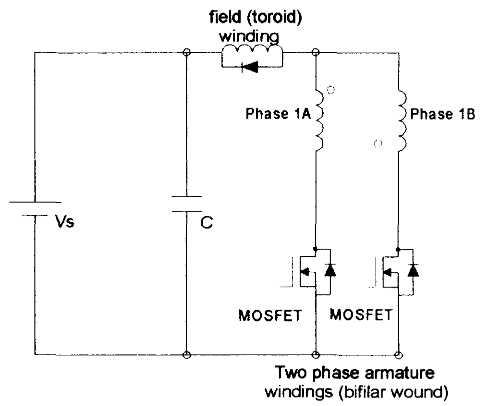


Figure 3.9 :- Field alternative circuit for bifilar windings (current source half bridge converter circuit).

3.2.2 SHORT PITCHED WINDINGS WITH UNIPOLAR EXCITATION

3.2.2.1 SHORT PITCHED WINDINGS WITH UNIPOLAR EXCITATION WHICH ARE SINGLE COIL PER POLE

When the machine is operated with short pitched windings with unipolar current the asymmetric half bridge converter circuit is used (*Figure 3.10*). The flux in the stator poles is unipolar. Single coil per pole short pitched windings cannot use the transformer theory to transfer current from one phase to the next making the use of the half bridge converter and full bridge inverter circuits inappropriate. Flux is not linked between the two sets of windings so, in this form of winding type, the machine is actually two phase. Diodes give a current return path when a phase is switched off making the asymmetric half bridge circuit a good choice (*Figure 3.10*). It has two ground referenced gate drives and two floating gate drives making the circuit more complicated. It can have a small value power resistor (sense resistor, R_{sense}) to be used to limit the size of the current in the armature via a closed loop feedback to the gate driver circuitry.

3.2.2.2 SHORT PITCHED WINDINGS WITH UNIPOLAR EXCITATION WHICH ARE BIFILAR

This circuit has a single switch with the diode on the coupled winding, as shown in *Figure 3.11*.

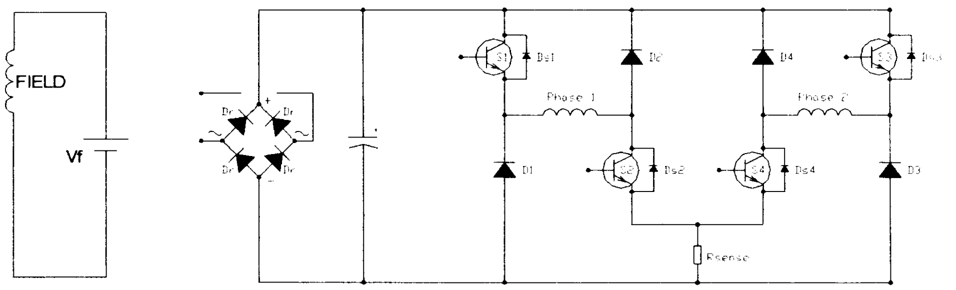


Figure 3.10 :- Asymmetric half bridge converter for short pitched, unipolar excitation two phase armature circuit.

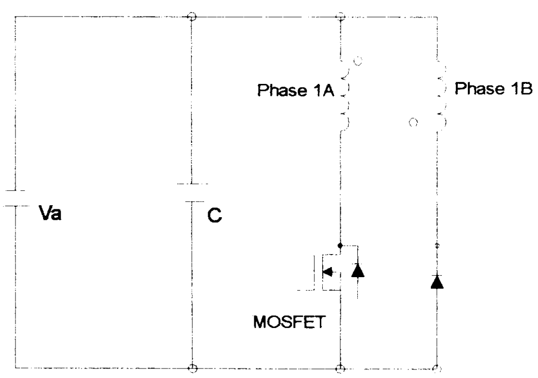


Figure 3.11 :- Circuit used for short pitched windings which are bifilar with unipolar excitation.

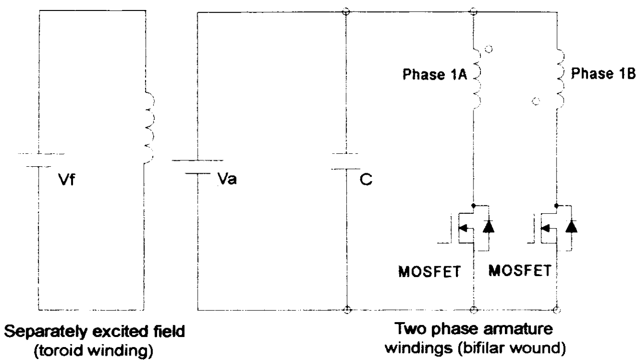


Figure 3.12 :- Circuit used for bifilar windings (half bridge converter circuit).

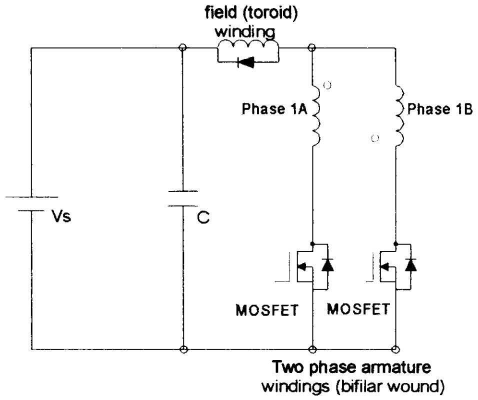


Figure 3.13 :- Alternative circuit for bifilar windings (current source half bridge converter circuit).

3.2.3 FULLY PITCHED WINDINGS

3.2.3.1 FULLY PITCHED WINDINGS WHICH ARE SINGLE COIL PER POLE

The choice of circuits for this winding selection is either the full bridge inverter or the split dc inverter per armature phase, as shown in *Figures 3.4* and *3.5*.

3.2.3.2 FULLY PITCHED WINDINGS WHICH ARE BIFILAR

Bifilar fully pitched phase windings allow use of the half bridge converter circuit. This is shown in *Figure 3.12* in which the toroidal field coil is separately excited from a dc power source. Phase 1A and Phase 1B are formed by series connection of coils positioned on all four stator poles. This drive can produce torque from all rotor positions through simultaneous energisation of all stator poles. Only two switches are required for this circuit.

An alternative circuit is shown in *Figure 3.13*. This circuit is based on a current source half bridge converter. The current source inductor is the toroidal field winding. Testing the machine with the circuit in *Figure 3.12* allows the field ampere turns to be varied independently but, once an optimum is found, the series connection of *Figure 3.13* could be used.

The labels Phase 1A and Phase 1B refer to the two bifilar windings used for positive and negative armature current. They are each made up of coils on every stator pole. The bifilar windings offer a transformer effect for faster transfer of current from one phase to the next during switching.

3.2.4 CIRCUIT CHOICE FOR MACHINE TYPES

The Switched Reluctance Machine has a power electronics circuit that is dependent on the choice of winding type and how each coil of the windings is to be energised. This is true also for the Variable Reluctance Machine with dc assisted excitation.

The flux switching machine uses bifilar windings in the armature coils. It also has a field winding that can be excited in series with the armature coils. The Permanent Magnet Flux Switching Machine (PMFSM) has no field winding. Thus its can use the half bridge converter as the simplest circuit (bifilar windings with bipolar excitation). The H-bridge inverter may be used as an alternative if all the copper is used at the same time (bipolar excitation, single coil per pole) but this increases the circuit complexity.

3.3 EXPERIMENTAL ELECTRONICS FOR RELUCTANCE MACHINES WITH DC MAGNETIC FLUX ASSISTANCE

3.3.1 BACKGROUND

Switched Reluctance Machines and Flux Switching Machines cannot operate without some form of control circuitry acting on the power electronics. They require a form of rotor position sensor so the correct phase winding or correct current polarity can be energized to turn the rotor in a pre-determined direction (optical or Hall Effect sensors are commonplace but work exists for 'sensorless' detection of rotor position). The control of the phase firing timings is through the use of a microprocessor. The increase in complexity of controlling such machines does allow advantageous control. Pulse width modulation and pulsed timings can be performed to give better operational characteristics. Efficiency can be improved through greater control of the timings and it is possible to use control methods to reduce acoustic noise.

3.3.2 THE BACK EMF WAVESHAPE

The shape of the back emf curve is very important. Ideally a square wave is the optimum shape for maximum torque production. A simple power supply has only a constant power supply that may be varied manually in terms of magnitude only (pulse width modulation, PWM, is associated with more complicated power electronics). Using a simple dc machine analogy, the power supply V, the back emf E, armature IR copper loss volt drop and the inductance, L are linked by (3.2).

$$V = E + IR + L(dI/dt) \tag{3.2}$$

Where the back emf and current waveshapes are of the same magnitude polarity, the product of back emf and current gives positive torque. When the power supply voltage is fixed, where the back emf is small the rate of rise in current is high and low amounts of torque is produced. Where the back emf is large and the current is large and increasing, a large and increasing amount of positive torque is produced, which would speed the machine up. Where the back emf falls and the current is the same the amount of torque produced is still positive but reducing in magnitude. If the back emf is negative and the current positive, negative torque is produced, which would have a tendency to slow the machine down.

It is therefore apparent that the timings of energisation of armature phase winding is energised is dependent upon the waveshape and magnitudes of the back emf curve. If the armature is energised during negative back emf, the torque is negative. There are parts of the positive sections of the back emf curve where energisation will produce greater amounts of positive torque. It may be possible to not energise sections where the back emf is positive but small in magnitude as the amount of torque produced is minimal compared to the armature I^2R copper losses imposed. Thus it is not always appropriate to energise an armature phase winding for all the period that the back emf is positive in magnitude. Viewing the back emf waveshape, the armature current waveshape and the input and output powers will help in predicting the optimum timings for energizing each armature phase winding.

This is complicated by the speed of the machine. The back emf is proportional to the rotor speed *(Figure 3.14)*. The faster the rotor speed, the greater the magnitude in back emf. A faster rotor means that current build up for producing more torque is limited through there being less time for current rise. Positive torque production is limited due to higher speeds. There is a method for resolving this which involves starting the energisation earlier while the back emf is still negative. Although the torque produced during the negative back emf period is negative, when the back emf is positive, the current has already built up allowing a much greater positive torque production that will overcome and greatly exceed the negative torque such that the machine produces more torque than it would have done if the energisation had not been brought forward (advanced). An optimum timing setting also exists for this type of operational mode, in a similar manner to that of the slower speeds operation.

For the purposes of this thesis, pulse width modulation, PWM, is not used for regulating the voltage supply to the armature windings, although it is appreciated that PWM has its advantages. PWM can complicate the power electronics circuitry and the control electronics. Voltage regulation is not required, per se, since the machines have been designed to operate at various manually set voltages.

Instead, the control electronics is designed with three operational modes, one for each of the specified speed ranges – start up/slow speeds, mid range speeds and higher speeds.

Conventional control is via a small and preferably inexpensive microcontroller such as those found in the PIC range by Microchip Technology [97]. Such microcontrollers e.g. PIC16F84 are EEPROM devices which are relatively simple to program and reprogram when corrections are required. The calculations of timing sequences are relatively easy to achieve.

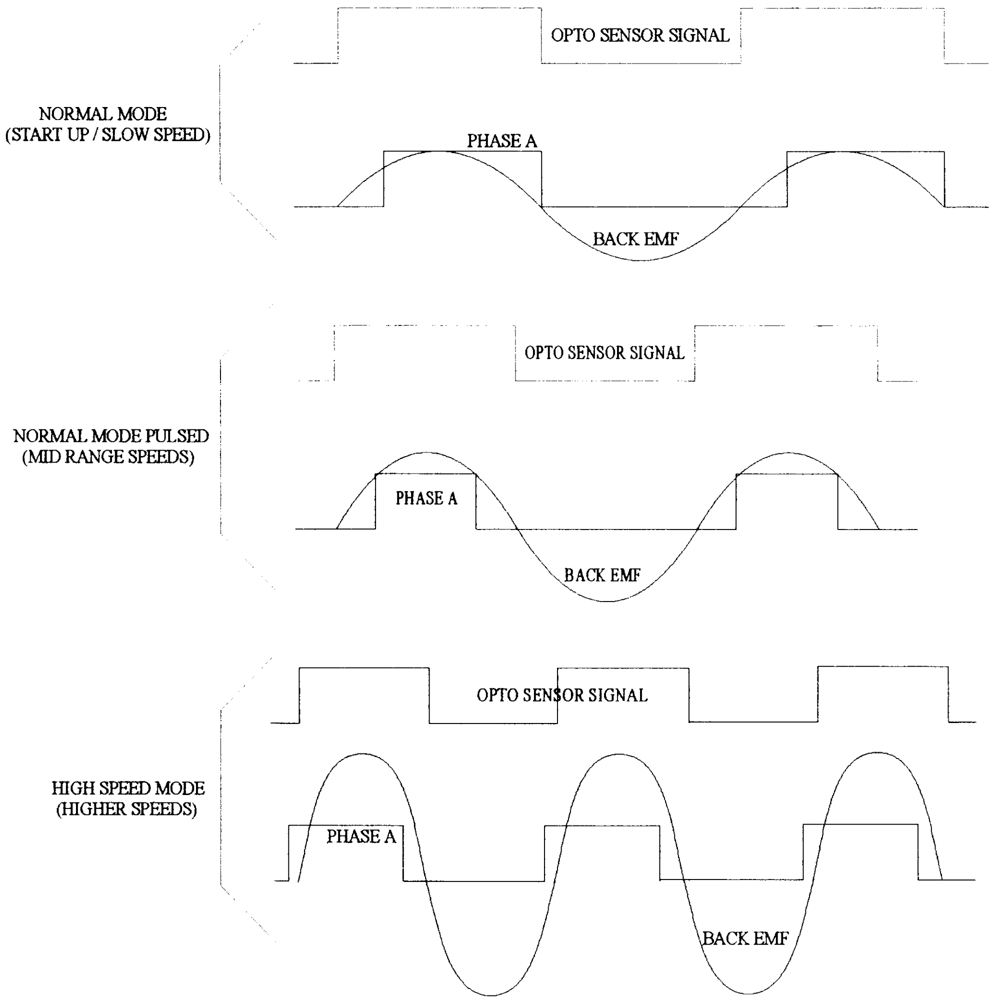


Figure 3.14 :- Speed Modes in relation to an example sinusoidal back emf trace and an opto sensor signal.

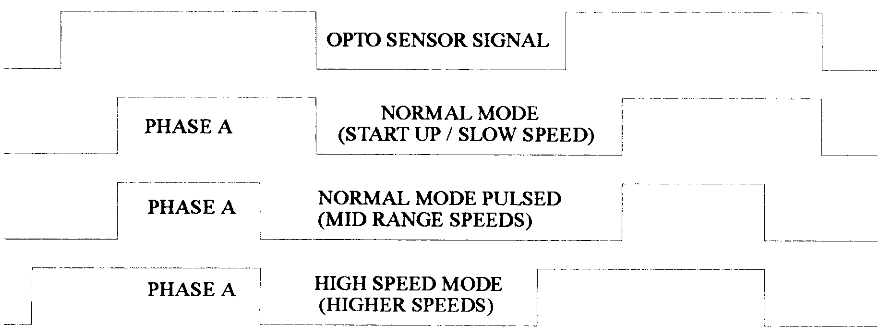


Figure 3.15 :- Speed Modes relative to opto sensor signal.

3.3.3 EXPERIMENTAL SPEED MODES

There were three speed modes that will be referred to as Normal Mode (without pulsing), Normal Mode (with pulsing) and High Speed Mode (*Figures 3.14, 3.15, 3.16, 3.17*). Each Mode is designed for operating a machine depending on both the back emf waveshape and speed requirement, depending on which mode would give the maximum efficiency.

3.3.3.1 NORMAL MODE (WITHOUT PULSING)

At start up and lower speeds the Normal Mode without pulsing is used and this starts armature phase firing at the change or a delay after the change of the opto sensor state (*Figures 3.14, 3.15, 3.20, 3.21*). The use of an opto sensor is one way that such machines determined which phase was to be energised.

From a stationary position, the torque produced has to turn the rotor and, at low speeds, much of the available cycle is used for positive torque production to build up and maintain rotor speed.

This method was acceptable when the machine speeds are slow as the time required for current transfer between successive energised phase windings is long and allows for current build-up for positive torque production. As the machine speeds up the time for current build up is not as long as required, giving less time for positive torque. The timing sequence is referred to as Normal Mode without pulsing.

This is only possible due to use of a variable voltage supply. Pulse Width Modulation, PWM, could have been used to control the current with PWM increasing with speed until the motor is fast enough for a single pulse mode (same as Normal Mode with pulsing). PWM was not used in order to simplify the control algorithm.

3.3.3.2 NORMAL MODE (WITH PULSING)

At mid range speeds it is desirable to pulse the supply to the armature windings to give better machine performance (Normal Mode with pulsing) (*Figures 3.15, 3.20*). The back emf magnitude is greater at higher speeds. A pulsed energisation during a period where the back emf is of a larger magnitude gives efficient higher torque production (where the back emf is low, the torque produced is at the expense of higher copper loss in the energised armature windings due to raised current levels).

The armature phase firing started at or a delay after the opto sensor has changed state but is then switched off after a second delay but before the opto sensor has changed state again. The next phase is energised in the same way upon the next opto sensor state change.

3.3.3.3 HIGH SPEED MODE

At higher speeds the increasing back-emf inhibits the fast rise in positive torque-producing current for a given supply voltage. As the machine speeds up, the time delay between the switching phases becomes a larger fraction of the time for one cycle, resulting in the next phase having less time to build up enough current to produce a significantly useful positive torque on the rotor to keep it turning (*Figures 3.15, 3.20, 3.22*).

The phase firing is brought forward (advanced) to a point before the opto sensor changed state such that the back-emf is small allowing a faster current rise (*Figures 3.15, 3.22*). The initial negative torque produced was offset by the boost in positive torque production as the opto sensor changed state and that phase remains energised. The phase is turned off such that a small delay existed between energising each phase to allow the current in each phase to minimise before the next phase is turned on. This is called the High Speed Mode.

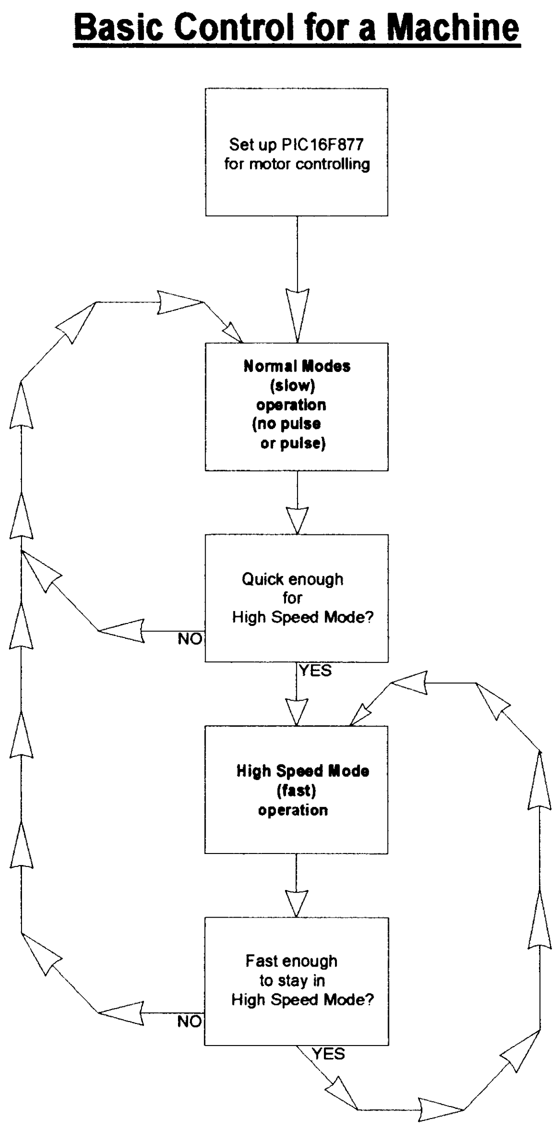


Figure 3.16 :- Flow diagram for Normal Mode (no pulsing) and High Speed Mode.

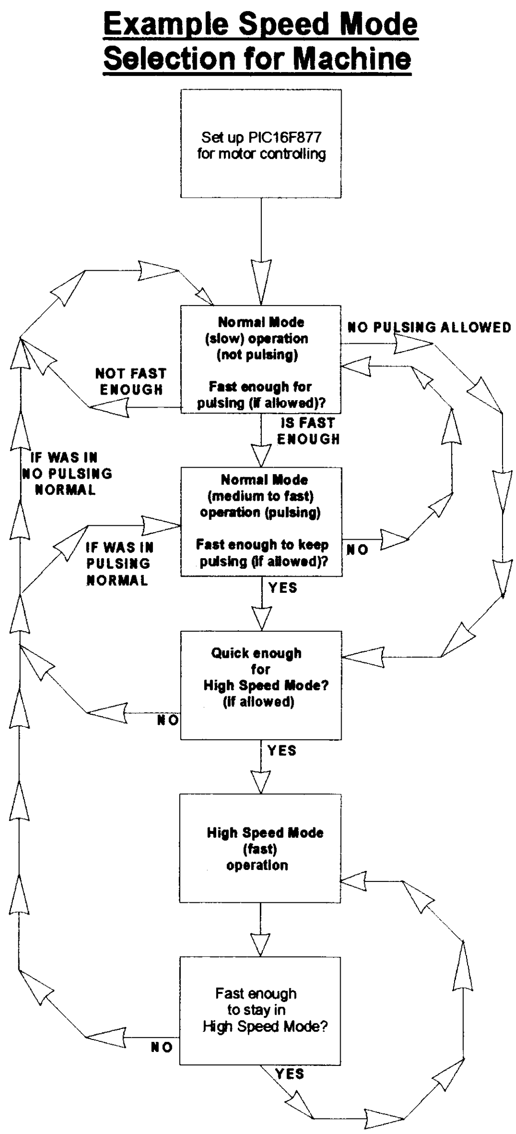


Figure 3.17 :- Complete flow diagram.

3.3.3.4 CHANGING BETWEEN SPEED MODES

The crossover between Modes has to occur at a definable speed (*Figures 3.16-3.19*). The method allows a machine speeding up to enter a faster Speed Mode but allows a slowing machine to return to a slower Speed Mode, without there being an error situation where a machine is stuck switching erratically between two Speed Modes.

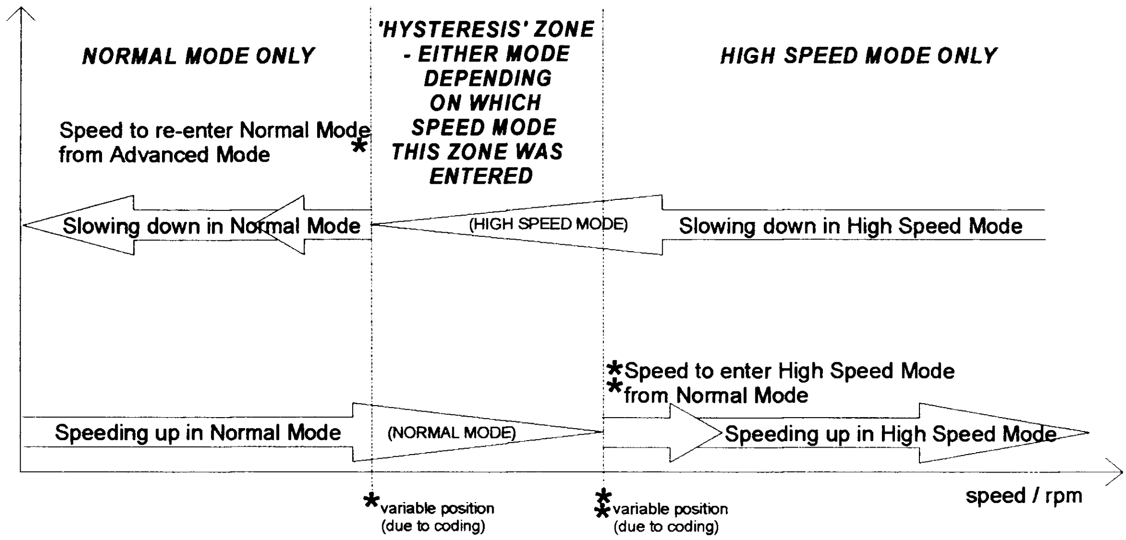


Figure 3.18 :- Hysteresis between Speed Modes.

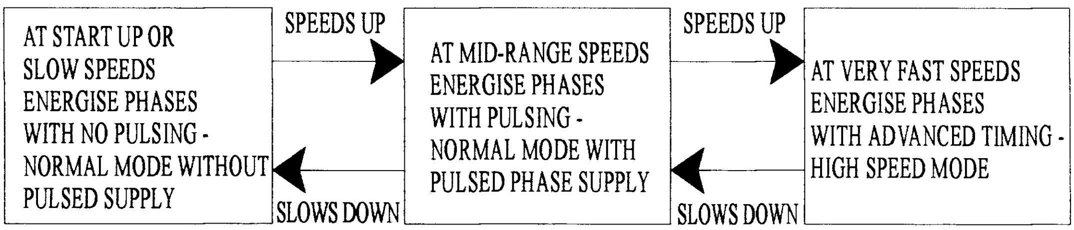


Figure 3.19 :- Speed Modes transition chart.

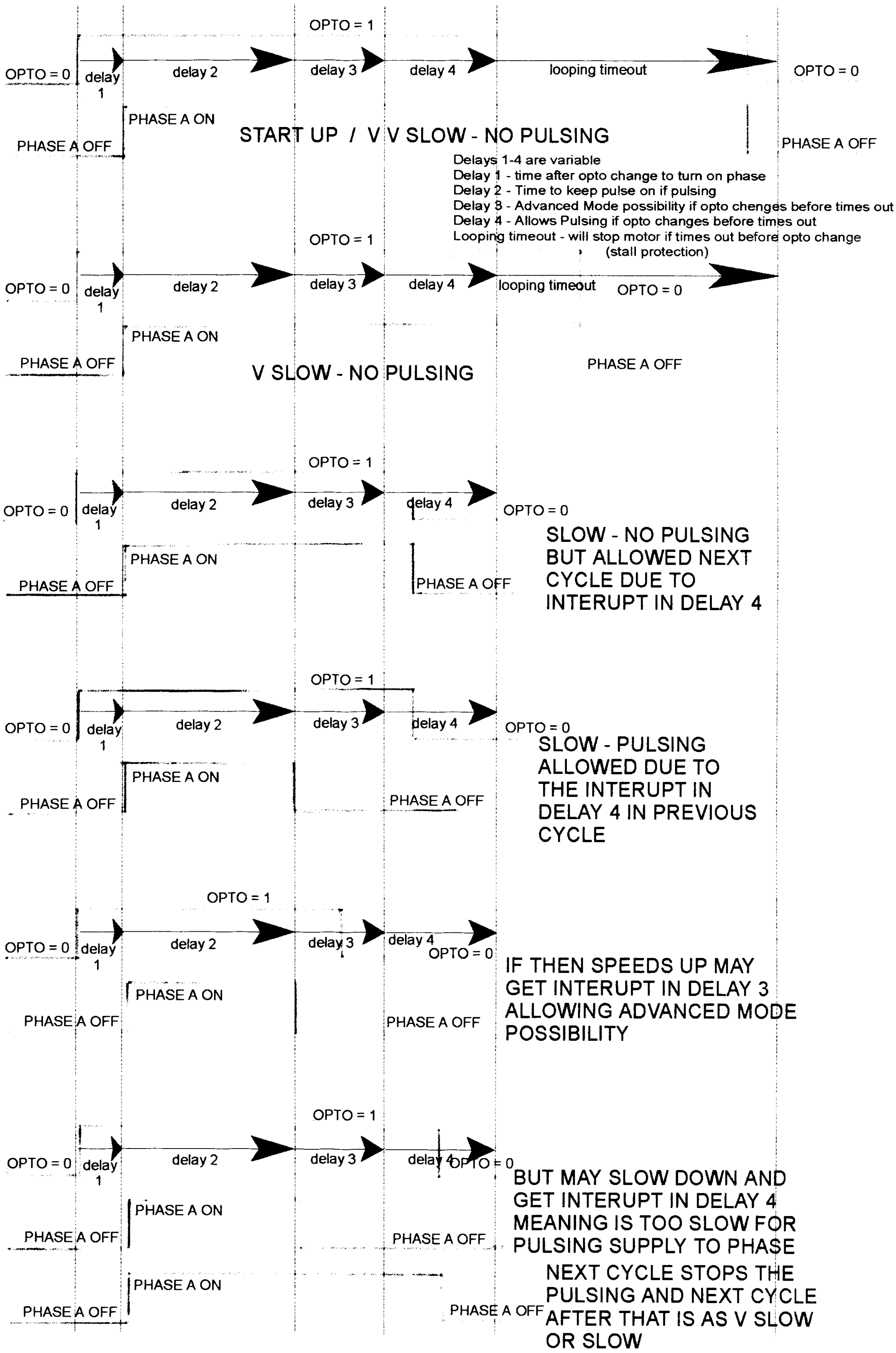


Figure 3.20 :- Transitions and timings behind Normal Modes.

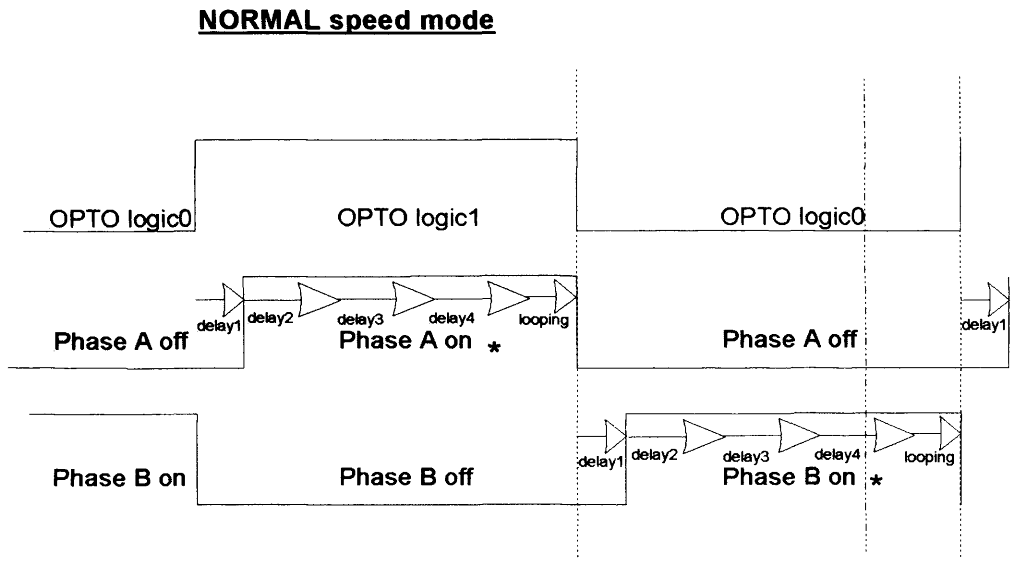


Figure 3.21 :- Normal Speed Modes variable delays.

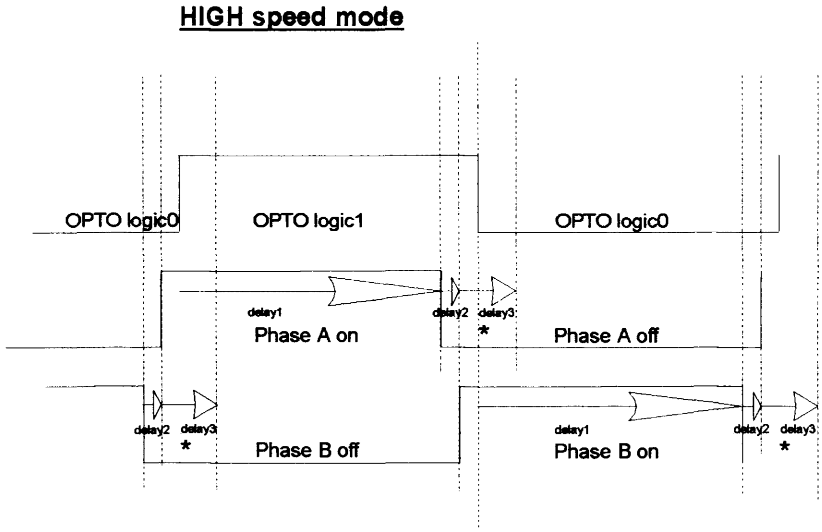


Figure 3.22 :- High Speed Mode variable delays.

The speeds at which the machine change speed modes is variable. A variable hysteresis loop exists to prevent the machine from changing in between modes indefinitely (*Figure 3.18*). The machine then operates within any one of three possible Speed Modes depending upon the machine speed (*Figures 3.16, 3.17, 3.19*).

Some control algorithms may not want the Normal Mode with pulsing and this is shown by the flow diagram in *Figure 3.16.* The basic outline of the full program code is seen in *Figure 3.17.*

The frequency of the opto sensor is directly related to the rotor speed. For example, for a 4/2 two-phase machine, the frequency of each phase winding, if multiplied by 30, represents the speed (rpm) of the machine. The changing points are definable by rpm values.

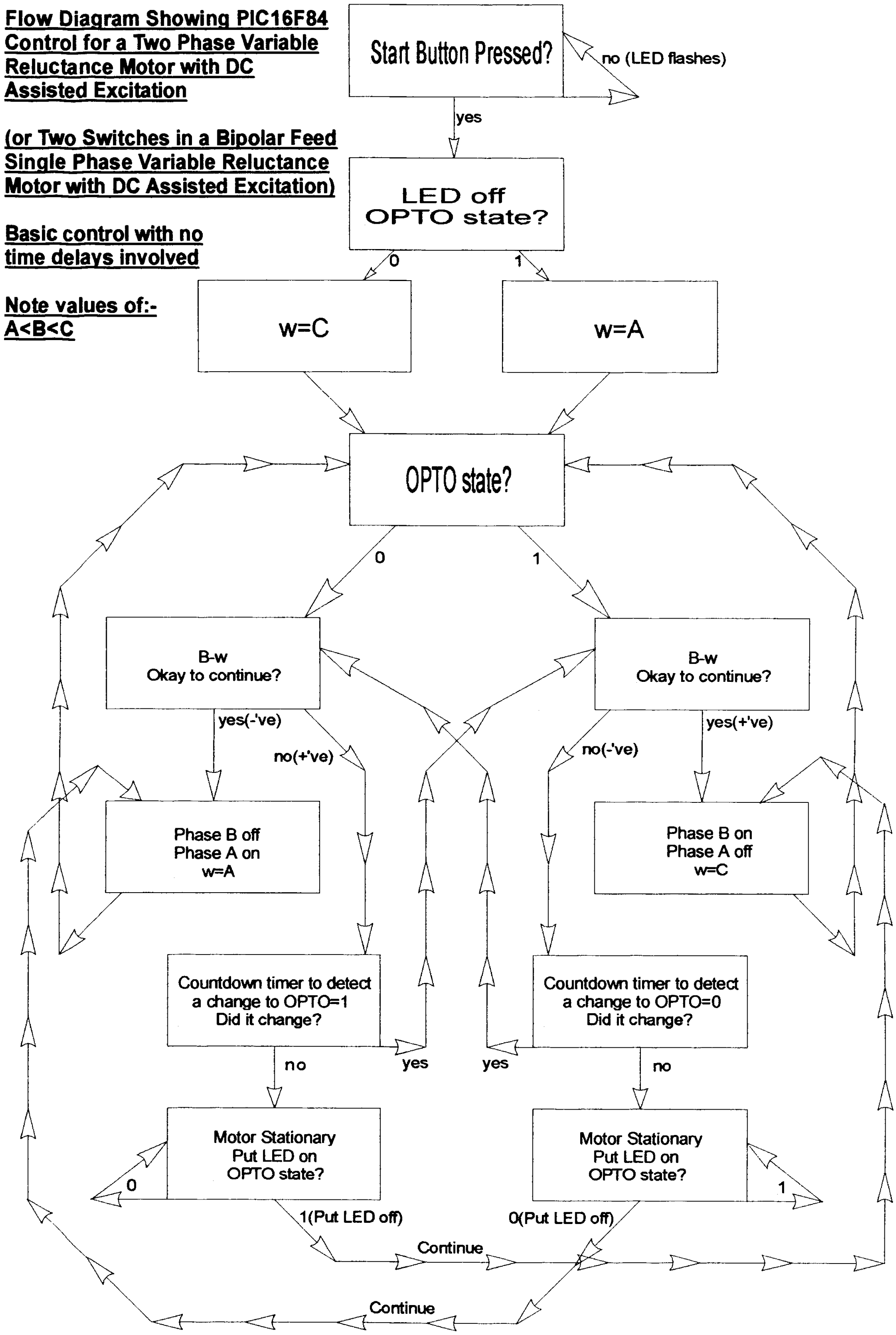


Figure 3.23 :- Simplest control flow diagram.

3.3.4 IMPLEMENTATION OF CONTROL ALGORITHMS

Figure 3.23 is an example of a simple control flow diagram. A simple low cost microcontroller only uses time delays programmed into the coding and there is no easy method of altering this externally. At slow speeds the change of optical (opto) sensor signal from low to high or vice versa would signal one phase to switch off, followed by a time delay to allow the current to reduce out of that phase before the next phase was turned on (*Figure 3.23*). The opto edge could equally well trigger the turn on event with a timed pulse length before the switch is turned off again.

Simple microprocessor software can implement any of the Speed Modes and can easily change between them through the speed range. Such software cannot modify the armature phase firing angles in response to a user input as the angles are fixed in the software.

It is possible to calculate, for a given speed, the appropriate armature phase firing angles but it is at the expense of overly complex software program code.

Direct time delays are used instead for the phase angle timings in experimental timing applications. This control circuit is specifically designed for experimentally working out the best time delays for specific machine applications. As such, when the time delays are found which best matched the machine application, the phase angle is merely a calculation for that speed. This control circuit allowed the machine to have various time delays for different speeds i.e. a time delay profile can be designed for that machine to get best performance at various speeds.

Any requirement to alter the time delays within the program of a simple microcontroller would usually involve reprogramming the microcontroller. In simple programs used in microcontrollers such as the PIC microcontroller (for example, *Figure 3.23*), the time delays are fixed values put into the program code and the only way to alter it is to take the PIC out of the circuit and to reprogram it again. The machine would have to be turned off during this time. It is desirable to alter the timing sequences as the machine is running as it is easier to optimise control parameters during testing.

A problem is that, for any new machine design, it is not known how exactly the timing between phases in both speed Modes should be optimised. Likewise, it is not known which speeds the machine should have to go between the Modes or even if all the Modes are required, such as not having the High Speed Mode for slow speed machine designs.

The PIC16F8X range by Microchip [98] has 8 internal analogue to digital (A/D) converters. The voltage inputs to these converters (eight in total) are used to give variable time delays. Simple potential dividers are used to provide the variable voltage to give the variable time delays for the armature phase firing timings. It is possible to have a program that could give all possible time delays for the firing sequences as variables that could be altered at any time without the need for stopping the machine and reprogramming the PIC. It is possible to have variable time delays with an accuracy potentially to 1μs (for a conventional 4MHz PIC, or even 0.2μs for the new 20MHz generation of PIC).

This allows the machine to be controlled and re-timed as the machine is in fact turning. Hence optimization of the machine control for specific applications is possible. The speed at which changing between Speed Modes is controllable as well as overriding the Advanced Mode to prevent it from being accessed by the machine.

The coding was such that the 20MHz PIC could have been altered to give several 'steps' similar to a speed response curve, allowing greater optimization of the machine over a wider speed range.

The program code was such that it was transferable to smaller and cheaper microcontrollers when optimisation of the timing sequences is completed. The control coding was written so that it was easily altered for programming the much smaller PIC16F84 microcontroller using the delay times found by this development circuit (the values being fixed in such a microcontroller).

A flow diagram representing this coding is shown in Appendix B to give a brief idea of how this coding worked. The PIC16F84 version of the flow chart is shown in Appendix B demonstrating the simplicity of the conversion of the coding into the PIC16F84. By being able to do this the final circuit size is kept small. Appendix C demonstrates the coding used for the PIC16F877 for a two speed mode application (Normal Mode without pulsing and High Speed Mode). Optimization of the machine control is shown in Appendix B. The PIC16F877 has built-in analogue to digital converters that allow the timing delays to be variable. The PIC16F84 has no analogue to digital converters and, as such, requires the delays to be made fixed vales based upon optimised values obtained from the PIC16F877. The PIC16F84 is a less complex, smaller and cheaper microcontroller than the upwardly compatible PIC16F877 [97, 98].

3.3.5 OTHER 16F877 PROGRAM FEATURES

Features of this controller include starting the machine from powering up the controller (a switch exists to override this). The machine also does not have to enter the High Speed Mode as an override button exists (it is not regarded as essential to start the Machine in the High Speed Mode). If the machine is stationary for a short time period (around two seconds) it turns off all the phases automatically. User intervention is then required to restart the machine (a master reset option standard to all such PIC microcontrollers is utilized by use of a switch). The value of the potential divider does not have to be accepted by the microcontroller as a switch is used to allow the value to be used or ignored. If timing delays are such that the machine attempts to fire both phases at the same time, it is prevented from doing so by switching off the already energized phase and then switching on the next phase following a built-in safety delay - the error being reported with a diagnostic LED. An LCD display with its own dedicated PIC microcontroller has been developed to give user timing information and error information (Appendix D). Phase windings with two switches per phase are accommodated via two outputs per phase (both have identical outputs) and one switch per phase is merely achieved by using one output only. The coding required is adaptable to smaller microprocessors so optimised values are entered at programming and the finalized control is still achievable at the smallest cost.

Changes to the timing of the machine are immediate. The machine does not require being switched off for altering the timing. The circuit only needs to be connected to the gate drivers of the armature switches. As such it is an ideal prototyping circuit.

3.4 ADDITIONAL CONTROL ALGORITHMS

There are several possible ways of energising Reluctance machines. A discussion into the merits of each type is discussed in [3, 92-94]. Some of the control circuits required are discussed in papers such as [94, 95, 96]. The use of microprocessors for control is a requirement in such machines, one such example being that of the PIC microprocessor range [97, 98].

Some machines may require more complex control algorithms for specific applications. More complex control algorithms for any of these machines may include features such as sensorless rotor position estimation [99-103] and high performance speed control [104, 105]. Such references have been included for further reading purposes only to illustrate some of the recent research publications. A simpler reference for machine control may be found in [3]. It is known that improved control algorithms may reduce the effects of cogging torque and acoustic noise. It is also a fact that some machine problems may be alleviated through redesigning the machines with the aid of 2/3D Finite Element Analysis and other design aids such as simpler two dimensional reluctance calculations [106].

3.5 SUMMARY

The choice of circuit is dependent on armature winding type and power electronic circuit complexity. The decision to have the field winding in series with the armature windings also affects circuit choice. The circuit cost is proportional to the number of components used (switches and diodes). The bipolar excitation with bifilar winding armature is ideally suited to the low cost half bridge converter circuit. The unipolar excitation with single coil per pole short pitched windings can be used with an asymmetric half bridge converter. The bipolar excitation single coil per pole configuration allows full utilisation of armature copper but is not investigated further as it has the most complex circuitry (full bridge inverter circuit) and offers the highest armature copper losses.

The gate drive input signal is generally derived from a position sensor signal, such as from an opto sensor, as current commutation in reluctance machines cannot occur without some form of rotor position detection. Microcontrollers are often required for better manipulation of the gate drive signals for improved machine performance.

A new controller is demonstrated to operate at a speed of 19020 rpm in both main speed modes for Variable Reluctance based two-phase machines. It is demonstrated to operate at up to 126,000rpm using signal generators. Full timing control is shown for the two speed mode possibilities. Control is 'on-the-fly' as it is immediate upon the machine in question. The PIC16F877 microprocessor does not need to be reprogrammed in order to control such machines.

The optimisation potential is demonstrated with the Variable Reluctance Drives and the PMFSM, as will be shown later in this thesis. The timing values have been successfully transferred into the smaller PIC16F84 microprocessor where the same optimised control timings were still maintained.

The control circuitry and the program coding can be used in assisting optimisation of control timings for 2-phase machines or single phase bipolar machines.

It is possible to improve this type of coding further to have more than three speed modes and that the control could then become similar to a speed response curve. This feasible concept was not investigated any further.

CHAPTER 4

DUAL STACK VARIABLE RELUCTANCE MACHINE WITH AXIALLY ORIENTATED FLUX ASSISTANCE

4.1 INTRODUCTION

An alternative way of providing additional dc flux to the systems explained in Chapter 2 is through the use of a toroidal, axially oriented coil. A Dual Stack Variable Reluctance Machine was constructed and tested in which the stator comprises two stacks separated by a coil to give axial magnetic flux. This construction was similar to a homopolar inductor alternator [38-41, 44]. This chapter describes and develops the complex magnetic structure of the machine and describes the test results. The possible power electronic and control circuits for the motor are described. The chapter describes the theory of selection of the angular orientation of the two stacks to give an improved waveform. This chapter extends this theory to discuss the relationship between crest factor and current waveform shape for armature windings. This chapter shows that optimisation of crest factor is not necessarily the best for torque production. An optimum angular position between the stacks may be obtained.

4.2 MACHINE TOPOLOGY

4.2.1 DESIGN FOR A DUAL STACK VARIABLE RELUCTANCE MACHINE

The easiest method to building a Dual Stack Variable Reluctance machine was to start with a pre-existing Switched Reluctance machine. The machine chosen was a 70mm stack length (laminated stator and rotor) two-phase 4/2 Switched Reluctance lawnmower motor. The outer diameter was 92mm. The lamination dimensions were to be used in the modelling of the motor.

A homopolar inductor alternator concept was applied to the motor. In simplest form the homopolar inductor alternator has its stator in three sections. The two end sections are stator stacks and the middle section incorporates the field winding. The two end sections of the Dual Stack Variable Reluctance Machine were derived from the stator stack laminations of the Switched Reluctance machine (by splitting up the laminations). The middle section was formed with a machined steel section that incorporates the toroidal field winding, linking the two back irons of the stator stacks. The rotor was to remain in its original state, namely a laminated two pole graded design.

The winding topology was reviewed during the design stages. The original machine had only one stator stack requiring four sets of coils (one set per stator pole). The new design had two stacks so the number of armature coils doubled. Excitation of the coils was such that one pair of diametrically opposite stator poles was energised as one phase and the other pair as the other phase. The inclusion of a field winding also increased the number of winding terminations.

The flux paths within the machine differ from the Switched Reluctance concept. The flux does not travel from one pole, across the rotor, to the other diametrically opposite pole when the stator coils for those poles are energized. Instead, the flux travels along the axis of the rotor, through the stator poles and back axially through the stator (i.e. the flux travels in an unconventional direction compared to a normal switched reluctance motor). It is an axial unipolar flux. Depending on the chosen direction of the flux through the rotor axis, the current direction in the stator coil windings is chosen according as to whether they are required to 'push' or 'pull' flux into or out of the rotor.

Short pitched coil windings were wound on the stator poles to provide flux to allow control of rotor movement (*Figures 4.1* and *4.2*). A toroidal field winding placed internally within the machine meant that flux is produced with little or no end winding leakage (due to the nature of the proposed flux paths). Also, less flux was to be produced by the armature coils on the stator poles. It is known that end winding leakage is present in machines where parts of the coils are external to the machine. By less flux being produced by the armature coils the amount of end winding leakage from these coils is reduced. End winding flux leakage cannot be eliminated fully as part of the coil is on the outer edge of the machine.

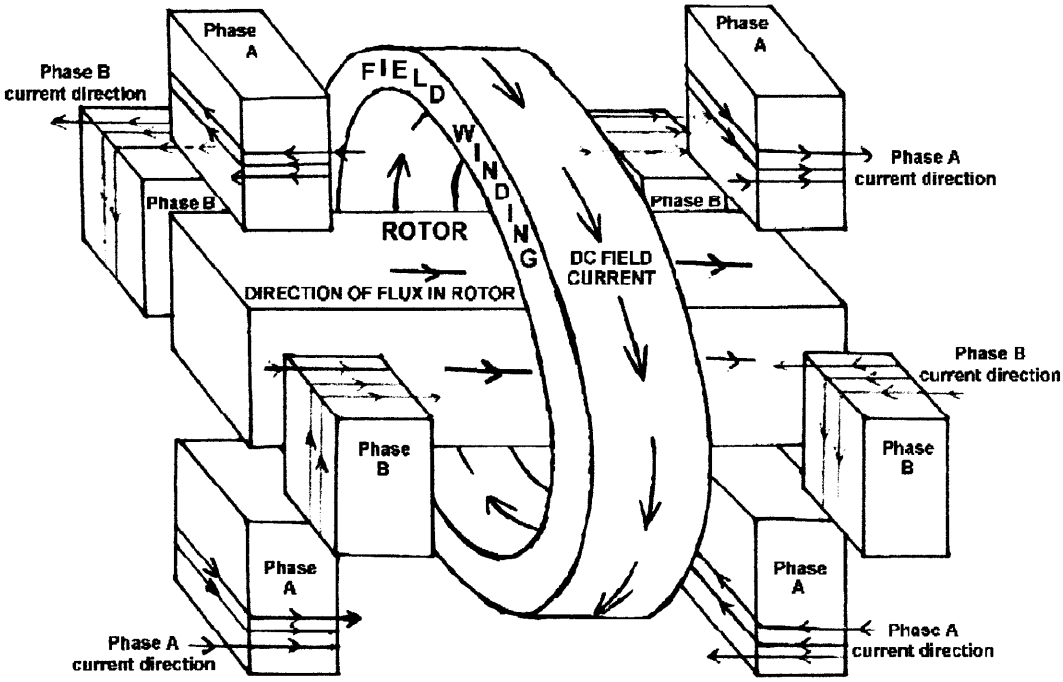


Figure 4.1 :- Simplified model showing a possible winding layout within the machine.

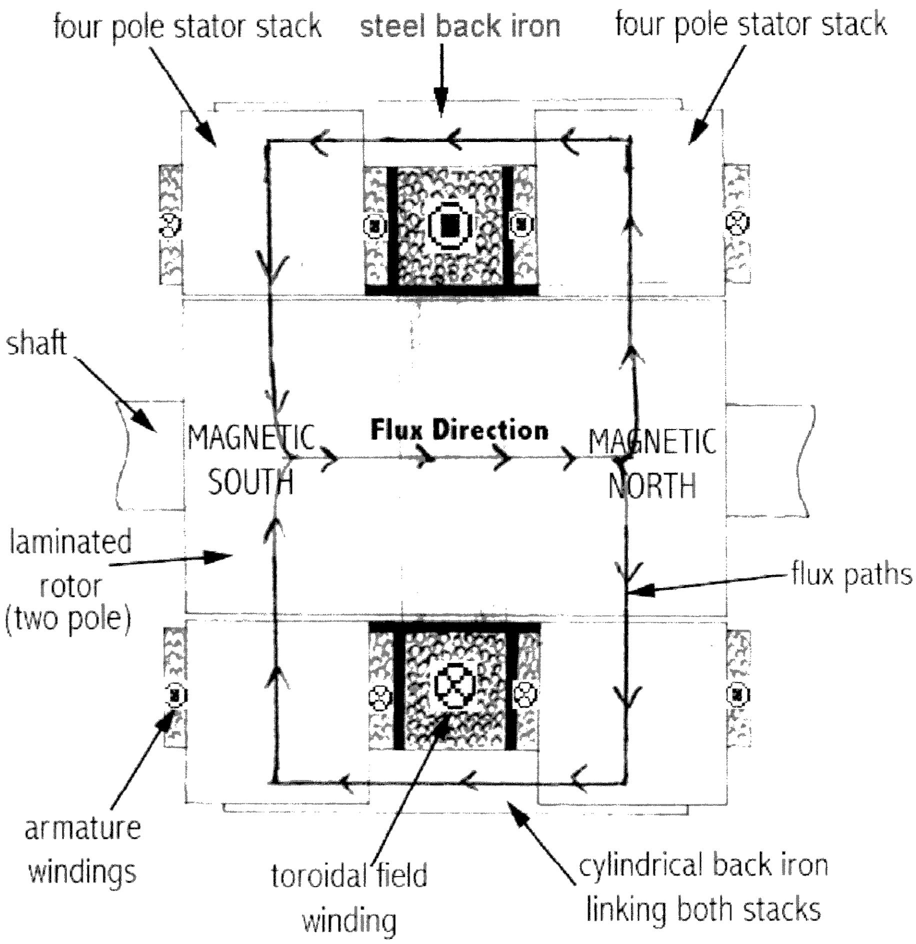


Figure 4.2 :- Magnetic flux path within machine.

The axially orientated dc coil is toroidal in shape (*Figure 4.1*). It provides maximum packing factor (0.76) and hence excellent utilisation of winding area. There are no end winding losses associated with the toroidal field. Permanent excitation of the field winding with dc current efficiently produces axial flux within the rotor. The toroidal field winding, depending on the sense of current within it, polarises the rotor so one end was a North Pole and the other a South Pole (*Figure 4.2*). The current direction in the field is not important as the machine operates the same way whichever way the rotor is magnetised.

This toroidal field winding produces a significant proportion of the total magneto motive force of the machine. All the copper of the field is used as the coil is permanently energised with dc current.

4.3 COMPARISON WITH HOMOPOLAR INDUCTOR ALTERNATOR AND SWITCHED RELUCTANCE MACHINES WITH DC FLUX

It is possible to have a homopolar inductor alternator operating as a motor. Switched Reluctance laminations were used to build a Variable Reluctance machine based upon a homopolar inductor alternator concept. A constant dc supply to the centrally placed field winding gives constant dc flux with the armature windings energised to produce torque to turn the motor. The flux from the field windings was used to aid torque production, improve efficiency of the motor, and reduce the required rating of power electronic components.

Figure 4.3 demonstrates a simple design comprising two stacks, each with four stator poles, and a two pole rotor and a centrally placed toroidal field winding between the two stacks. A fast method of building such a motor involved the redevelopment of a set of pre-existing Switched Reluctance laminations. This was the starting point for building a dual stack Variable Reluctance machine.

4.4 WINDING CONFIGURATIONS

The magnetic flux paths within the machine are three-dimensional. The toroidal field winding is permanently energised with dc current. The flux it produces is axial along the length of the rotor production making the rotor magnetically polarised at its ends. The flux at the ends is guided into or out of the stator poles by the stator pole windings in order to turn the rotor. At one end flux leaves the rotor whilst at the other it enters. There are two sets of bifilar wound windings on each stator pole, the poles labelled as A, A', B, B', C, C', D and D' (*Figures 4.3* and *4.4*).

There are different possible winding layouts. The choice of layout affects the choice of power electronics circuit for energising the windings. The flux paths required in the machine determined the sense of current in the armature coils. One stator stack will be seen to push flux into the rotor, the other stack pulls flux out.

When a phase is energised, the energised stator poles become an effective South Pole or North Pole (*Figures 4.1, 4.2* and *4.3*), depending on the sense of current. The field winding magnetises the rotor axially such that the rotor pole faces were North Poles at one end and South Poles at the other.

Where there are like Poles between aligned rotor and stator pole faces, repulsion between the fluxes forces the rotor away from that stator pole pair giving the effect of reduced leakage flux from that stator pole pair. Where there are unlike Poles between unaligned rotor and stator pole faces, attraction pulls the rotor into alignment. The rotor turns until it reaches an aligned position where the pole faces of the rotor and stator are of unlike magnetic polarity (flux crosses the air gap between them).

The effect is the same if the field current direction changes. Each phase has its own aligned rotor position. The forces of repulsion and attraction turn the rotor.

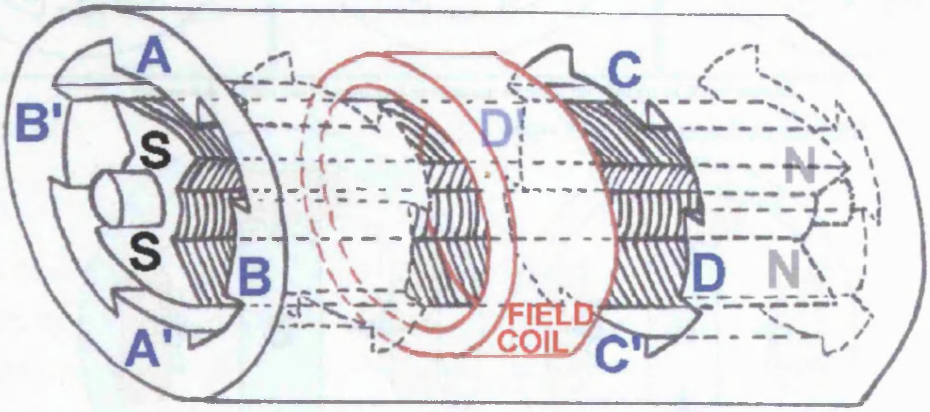


Figure 4.3 :- Coil labels for both stacks in three dimensions.

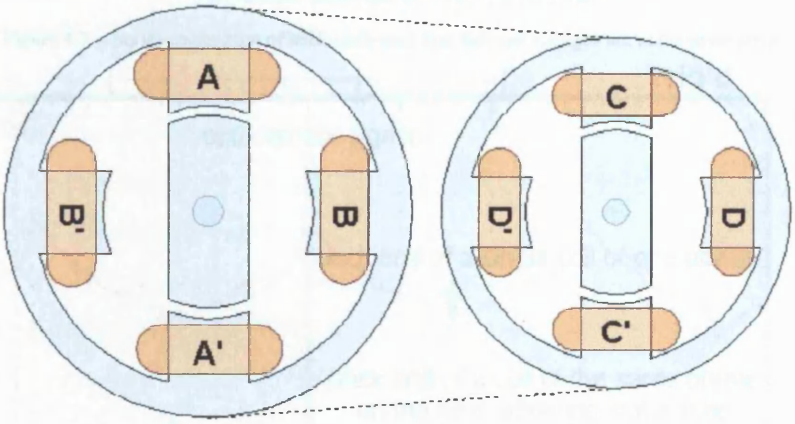


Figure 4.4 :- Coil labels for both stacks (left stack at front, right stack at rear).

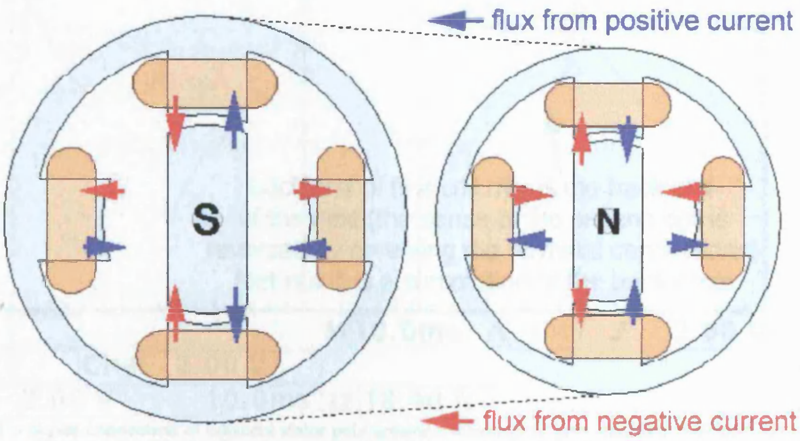


Figure 4.5 :- Single phase bipolar windings (only one direction of current at a time).

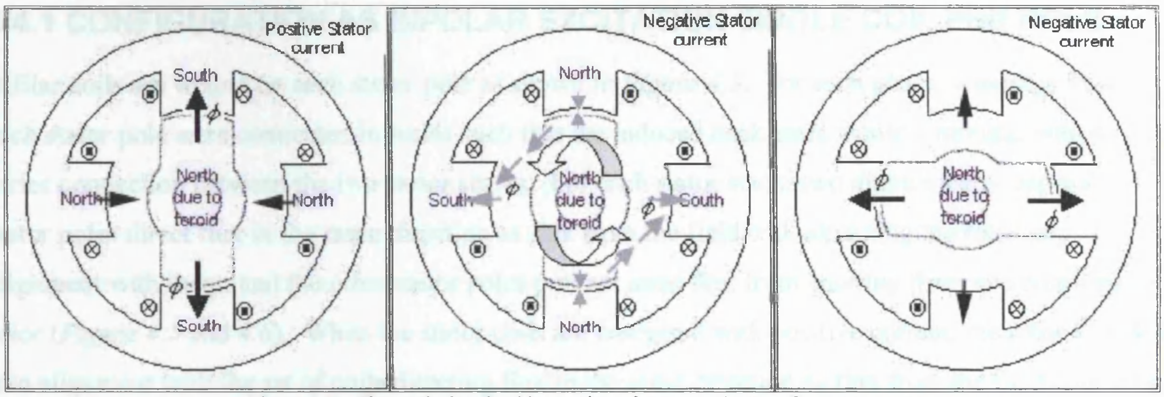


Figure 4.6 :- Flux paths involved in rotation of rotor on change of stator current.

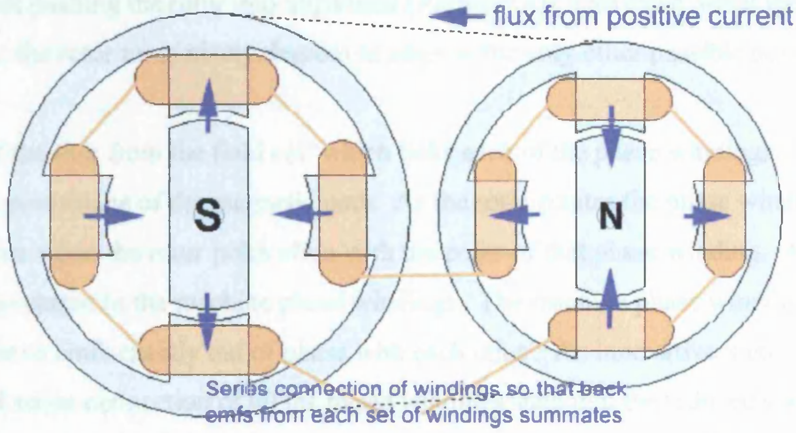


Figure 4.7 :- Series connection of back emfs such that induced voltages are in the same sense.

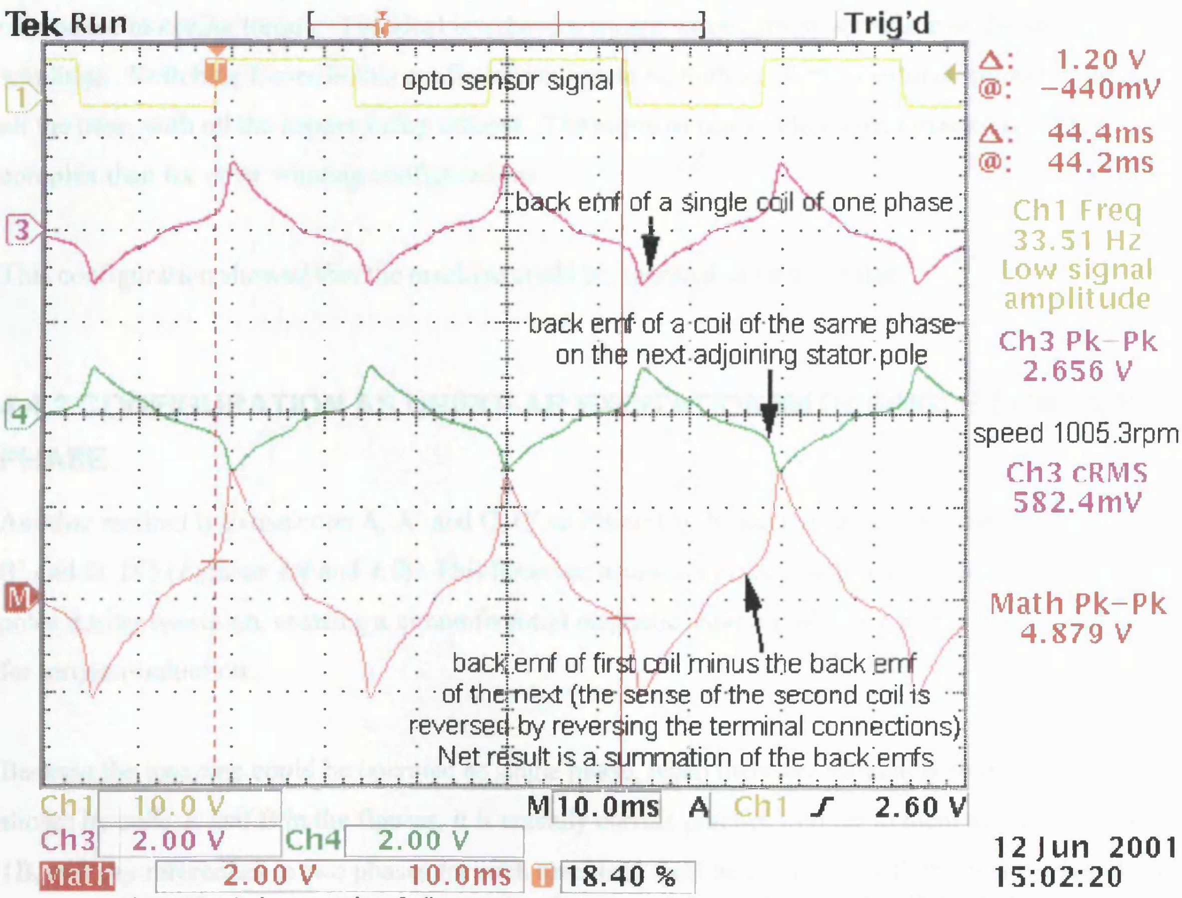


Figure 4.8 :- Series connection of adjacent stator pole armature windings to give summation of induced voltages.

4.4.1 CONFIGURATION AS BIPOLAR EXCITATION SINGLE COIL PER POLE

Bifilar coils are wound on each stator pole as shown in *Figure 4.5*. For each phase, windings from each stator pole were connected in series such that the induced back emfs would summate, with a series connection between the two stator stacks. For each stator stack, two diametrically opposite stator poles direct flux in the same direction as flux from the field coil, attracting the rotor into alignment with them, and the other stator poles prevent axial flux from entering them and repelling the rotor (*Figure 4.5* and *4.6*). When the stator coils are energised with positive current, the rotor is pulled into alignment with the set of coils directing flux in the same direction as flux from the field coil, with the unaligned poles pushing the rotor into alignment (*Figure 4.6*). Therefore, when the current direction changes, the rotor turns ninety degrees to align in the only other possible position.

The magnitude of the flux from the field coil which links each of the phase windings of the stator is modulated by the permeance of the magnetic path. As the rotor rotates the phase winding flux linkage reaches a maximum when the rotor poles align with the poles of that phase winding. An alternating emf is therefore produced in the machine phase windings. The machine phase windings on adjacent reluctance poles have emfs exactly out of phase with each other. An innovative method of driving this machine involved series connection of all the phase windings such that the induced voltages are all in the same sense (*Figure 4.7*). A bipolar current is then forced to flow in opposition to this induced emf to produce motoring torque. The ideal is to have a square wave current waveform in the armature windings. Switching losses in this configuration would be high as all the windings would be energised all the time, with all the copper being utilised. The required power electronic circuitry is more complex than for other winding configurations.

This configuration showed that the machine could be operated as single phase.

4.4.2 CONFIGURATION AS UNIPOLAR EXCITATION WITH SINGLE COIL PER PHASE

Another method is to use coils A, A' and C, C' as Phase 1 only and the remaining coils for Phase 2 (B, B' and D, D') (*Figures 4.4* and *4.7*). This however means that flux could escape through unexcited poles during operation, creating a circumferential magnetic short circuit. Not all the flux is available for torque production.

Because the machine could be operated as single phase, when there are references to two phases, as shown by coils A and B in the figures, it is actually correct practice to refer to them as phases 1A and 1B, and any references to two phases for such machines shall be construed with the literal meaning in mind.

4.4.3 CONFIGURATION AS BIPOLAR EXCITATION WITH BIFILAR WINDINGS

To prevent flux leakage, two diametrically opposite stator poles are energised to bring the rotor into alignment and the other two poles are energised to push the rotor away for each phase, as shown in *Figure 4.8*. This is so that one stator pole pair provides flux in the direction of the field flux and the other pair opposes any flux leakage and pushes the rotor away which was thought to aid maximum flux usage for torque production. Only one phase is energised at a time.

When the phase changes, the transformer effect (the armature windings of one phase are magnetically linked to the armature windings of the other phase in the same manner as a transformer as both phases have parallel wound windings) provided by the bifilar windings of the second configuration reduced the switching losses and speeds up switching. The copper is thus used more effectively.

For both configurations, shown in *Figures 4.9* and *4.*10, half the copper is energised at any one time. For both configurations, one phase has one stator pole pair as its aligned position and the other phase has the other stator pole pair as its aligned position.

This second configuration (*Figure 4.10*) was to be used initially for the design of the machine. It was assumed that the transformer effect of bifilar windings would allow fast current transfer between windings that could help the motor during operation. Future diagrams may refer to the windings in sections 4.4.2 and 4.4.3 as 'conv' and 'bif' for unipolar excitation with single coil per phase and bipolar excitation with bifilar windings respectively only for the purpose of simplifying the text.

Appendix F shows an alternative diagram of one variant for the flux paths in such a machine.

Chapter 3 covers the possible power electronic circuits used for each winding configuration.

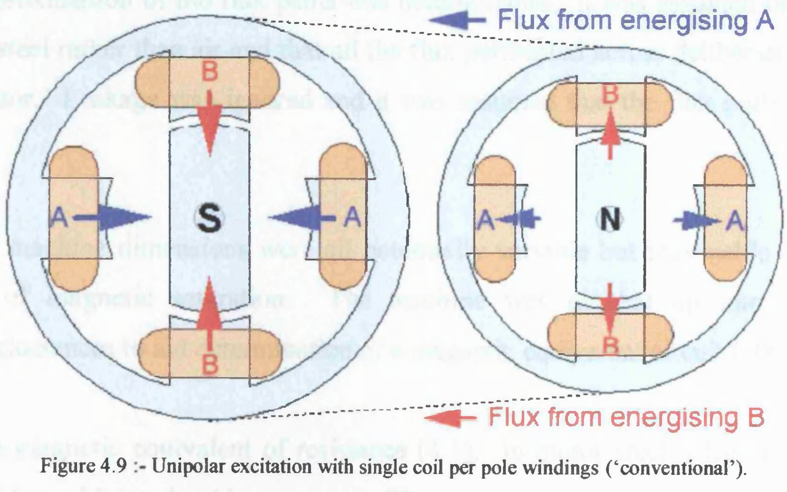


Figure 4.9 :- Unipolar excitation with single coil per pole windings ('conventional').

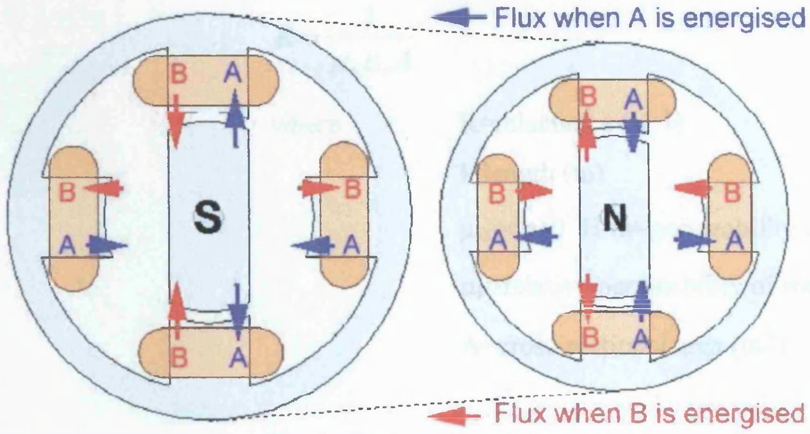


Figure 4.10 :- Bipolar excitation with bifilar windings (only one armature energised at a time).

4.5 STATIC MODELLING OF 3D MAGNETIC CIRCUIT

The windings, when energised, produce a magneto motive force (mmf). The flux produced by such windings, in an ideal situation, remains solely within the steel of the machine and across any air gaps between the rotor and stator poles (based on the assumption of no flux leakage). Saturation was ignored due to an assumption of linearity for ease of calculations. To evaluate a new concept, magnetic equivalent circuit techniques were more appropriate than finite element analysis as it allowed a quick method to evaluate different geometries.

The dual stack variable reluctance machine was modelled from first principles, namely by calculating the magnetic equivalent circuit by reluctance calculations [106]. The machine shape was determined in the form of a sketch of the machine and a sketch of the magnetic paths that might be found within the machine when energised in an aligned rotor position. By using the principles of magnetism an intuitive first approximation of the flux paths was determinable. It was assumed that flux permeates preferentially in steel rather than air and that all the flux permeated across deliberate air gaps between the rotor and stator. Leakage was ignored and it was assumed that the flux paths were as short as possible.

At this stage the machine dimensions were all potentially variable but reasonable care was taken to minimise areas of magnetic saturation. The machine was divided up into radial, axial and circumferential reluctances to aid determination of a magnetic equivalent circuit [106].

Reluctance is the magnetic equivalent of resistance (4.1). In motor steel it has a low value (due to high μ_r). In air it has a high value (due to μ_r=1). The amount of air gaps in the design should be kept minimal so as to minimise reluctance of the motor flux paths.

$$R = \frac{l}{\mu_o \mu_r A} \qquad (4.1)$$

where R=reluctance (H^{-1})

l=length (m)

$\mu_o = 4\pi 10^{-7}$ H/m=permeability of free space

μ_r=relative permeability of material

A=cross sectional area (m^2)

As with any design, constraints were defined which included those due to use of existing laminations. The length of the two stator stacks, the space for armature and field windings, dimensions of the linking back iron between the two stacks were all definable. As the design was an adaptation of an existing motor, the laminations gave fixed values for some dimensions but the axial length of the individual stack laminations (length b) was still variable to an upper limit (the total length of the original stack).

The flux paths were split into sections where the flux was substantially uni-directional within that section. Each section was for one material only. Where two materials were next to each other, then two sections were made even though the flux was in the same direction. This was because it aided calculation of the reluctance within the sections. Where saturation was thought to occur, extra sections were included to give extra modelling security. The adjacent sections have to be in parallel or in series with each other (*Figures 4.11*). By having the sections in this manner and by calculating the reluctance of each section, a magnetic reluctance equivalence was calculated for the machine (in the same manner as resistances in series or parallel) (*Figure 4.12*).

By having as many variables as possible defined within the reluctance calculations, as the dimensions of the machine were varied, the equivalent reluctance was calculated. The design had to incorporate space for field and armature coils (slot areas) and any former used to hold the field and stacks in position relative to each other.

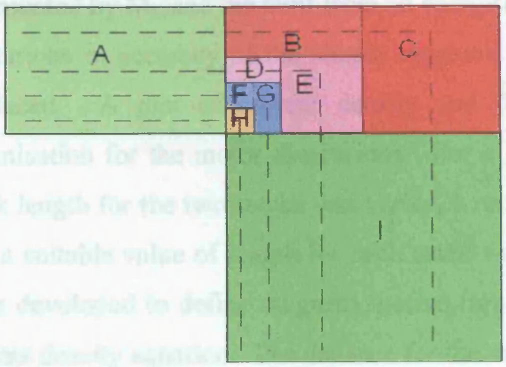


Figure 4.11 :- Labelled sections for obtaining equivalent reluctance

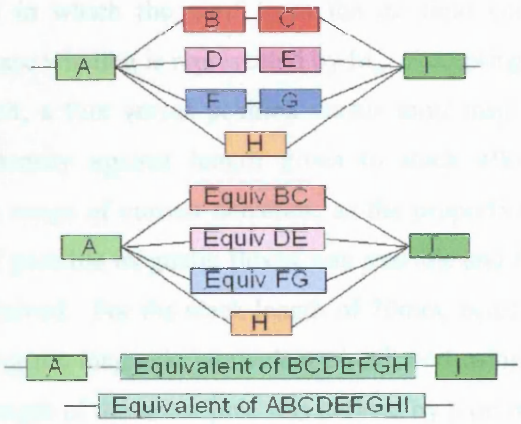


Figure 4.12 :- Determining equivalent reluctance of Figure 4.9

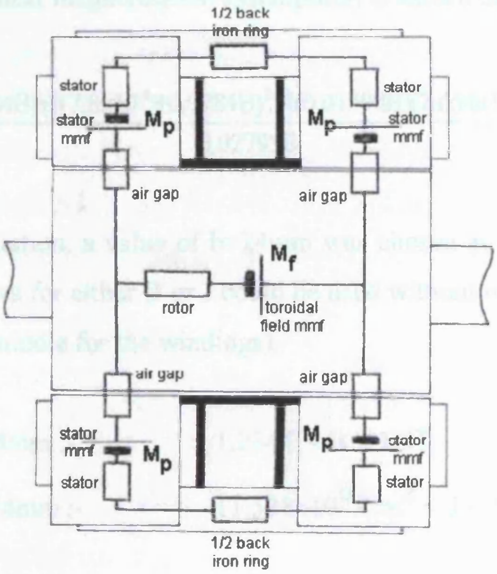


Figure 4.13 :- Magnetic equivalent circuit.

The greater the number of sections used the greater the potential accuracy of any calculations. This was a crude yet effective version of Finite Element Analysis. A magnetic equivalent circuit was thus derived (*Figure 4.13*). For each section the flux within each was affected by the cross sectional area that the flux passed though and this was a limiting factor on how great the flux could be. The total reluctance was simplified to a function of the dimensions of the motor. Dimensional parameterisation of the reluctances allowed any dimension and variation in shape to be accommodated. The machine was then modelled with a 2-dimensional model even though the magnetic paths were three-dimensional.

The product of magnetic flux and reluctance was equal to the total ampere turns of the motor. The amount of space available for windings (copper area) was the packing factor times the slot area. Dividing this by the cross sectional area of one strand of wire gave the number of turns available. The current density times the copper area gave the expected current in the windings. *Figure 4.13* shows an example of the equivalent circuit for one position in which the mmf from the dc field coil is represented by M_f, and the mmf from an energised phase winding is represented by M_p. Accepting the limitations in accuracy of the simple magnetic circuit, a flux versus position versus mmf map was produced. A plot of current density and flux density against length given to stack allowed optimisation for the motor dimensions. For a given range of current densities, as the proportion of stack length for the two stacks was varied, a range of possible magnetic fluxes was possible and from this a suitable value of length for each stack was obtained. For the stack length of 70mm, equations were developed to define magneto motive force using the magnetic equivalent circuit and using the current density equation. The distance for the axial length of the stator pole was defined by b (in mm). For a given flux density, B, and current density, J, and the motor dimensions, the equation for an optimum value of b (using a linear magnetization assumption) is shown in (4.2).

$$b=\frac{(7.8x10^{-6}J+0.9784B)\pm[(7.8x10^{-6}+0.9784B)^2-4(0.01398B)(2.654x10^{-4}J-805.85B)]^{1/2}}{0.02795B} \qquad (4.2)$$

From manipulation of this equation, a value of b=24mm was chosen as an optimum value. This was because a large choice of values for either B or J could be used without making the length of b too big (as space was required in the middle for the windings).

For various values of J at b=24mm :- $1.234T < B < 1.8T$

For various values of B at b=24mm :- $11.538x10^6 A/m^2 < J < 18.270x10^6 A/m^2$

A middle value of B and its corresponding value J were chosen i.e.:- B=1.5T, $J=15.794x10^6 A/m^2$. Due to the machine using laminations it was assumed that the flux within the machine would not be as great as predicted due to cross lamination fluxes, iron losses and eddy currents. It was assumed to be a good value though. Using these values, a magnetomotive force (mmf) of almost 1232A-t was calculated. This gave an idea of the machine capability.

4.6 DYNAMIC SIMULATION

The field coil was made using self bonding wire for dimensional stability (see Appendix E).

It was assumed that there was no flux leakage and that, for a two phase 4/2 motor design, a 90° revolution of the rotor gave a change in magnetic flux of dB=B. It was assumed that all the flux was used in torque production. The back emf expected for a given speed is shown in (4.3).

$$E_a = N\frac{d\Phi}{dt} = NA\frac{dB}{dt}$$

A = stator pole area, N = turn number per phase,

B = flux density in stator pole

$$x \text{ rpm} = \frac{x}{60} \text{ rps} \tag{4.3}$$

$$1 \text{ revolution takes } \frac{60}{x} \text{ seconds}$$

$$\tfrac{1}{4} \text{ revolution takes } \frac{15}{x} \text{ seconds}$$

$$\tfrac{1}{4} \text{ revolution} \quad dB{=}B$$

$$\tfrac{1}{4} \text{ revolution} \quad E_a = NA\frac{Bx}{15}$$

The model predicted the total magnetic equivalent reluctance for the machine. This value multiplied by magnetic flux was the same as the total ampere turns of the machine. The number of available turns for the field and stator (armature) windings was determined by the slot areas, packing factors and wire diameters selected during the modelling process. A dc motor equivalence was used to predict the motor performance (4.4). L(dI/dt) was ignored as the current, I, was assumed to be constant, for ease of calculation.

$$\begin{aligned} V_a &= E_a + I_a R_a \\ V_a I_a &= E_a I_a + (I_a)^2 R_a \end{aligned} \tag{4.4}$$

Such a theory assumed that the field winding produced all the flux and that all this flux interacted with current in the stator coils (armature) to produce torque. As the possible number of armature turns was known then the required current was calculated and used in the dc motor equivalence equations. The supply to the armature windings equalled the sum of the back emf and the volt drop across the armature windings (current times resistance). The back emf was predicted as was the armature winding resistance as shown by the previous equations. The power calculation was a simple expansion of the dc machine voltage equation (4.4). Torque was calculated through electromechanical energy conversion in the phase winding circuit model (4.5). The interaction of the current and back emf gave rise to torque production. It was possible to predict the performance of the motor for a given speed in radians per second.

$$\begin{aligned} E_a I_a = T\omega \quad \text{where } & E_a = \text{back emf (V)} \\ & I_a = \text{armature coil current (A)} \\ & T = \text{torque (Nm)} \\ & \omega = \text{speed in rads}^{-1} \end{aligned} \tag{4.5}$$

The magnetic flux was limited by the cross sectional area of sections of the motor. The smallest area was regarded as the constraining value and, for a required magnetic flux density, the mmf of the motor was calculated. The mmf produced by the stator coils and the field coil was to match this value and from the dc motor equations the performance of the motor was dynamically modelled for a given rotor speed and supply voltage. The choice of wire size and packing factor were varied and the model included error detection to prevent selection of impossible dimensions and parameters.

The model ignored laminations and assumed that the steel helped flux permeation. The laminations existed in the stator stacks in the radial plane. It was suggested that the mmf needed to drive flux perpendicular to the laminations would be greater than expected. The flux paths were three dimensional but the laminations were designed for two dimensional fluxes in the same plane as the laminations. Also, effects such as iron losses and eddy currents were assumed to not exist. There was a risk that the model would over predict the flux within the machine.

The choice and dimensions of plastic formers used to hold the stator stacks parallel to each other can be found in Appendix E.

4.6.1 TEMPERATURE RISE CALCULATIONS

Work was then conducted on how much heat was produced by the motor at various magnetic flux densities. The flux density was directly linked to the current density and mmf as shown in (4.6) (the equations are specific to stator axial-based lengths of b=24mm).

$$\begin{aligned} mmf &= 821.285B = 7.8x10^{-5}J \\ J &= 10.529x10^{6}B \end{aligned} \tag{4.6}$$

Stefan's Law was used to calculate the radiated heat as a temperature rise above ambient temperature when the wire carried a current (Appendix E). The modelling used on Microsoft Excel was such that, for any calculated slot area, mmf proportions from the windings or plastic former thickness, at any specified flux density (B), the currents in the coils, the voltages required, the I^2R resistive losses and the associated temperature rises above ambient were instantaneously calculated. The design was optimised numerically to give the shape with minimum reluctance (*Figure 4.14*) and optimum thermal characteristics. This allowed the maximization of the magnetic flux for a given supply. Table 4.1 gives a list of the main dimensions and turns numbers within the machine. *Figure 4.14* shows how the machine dimensions.

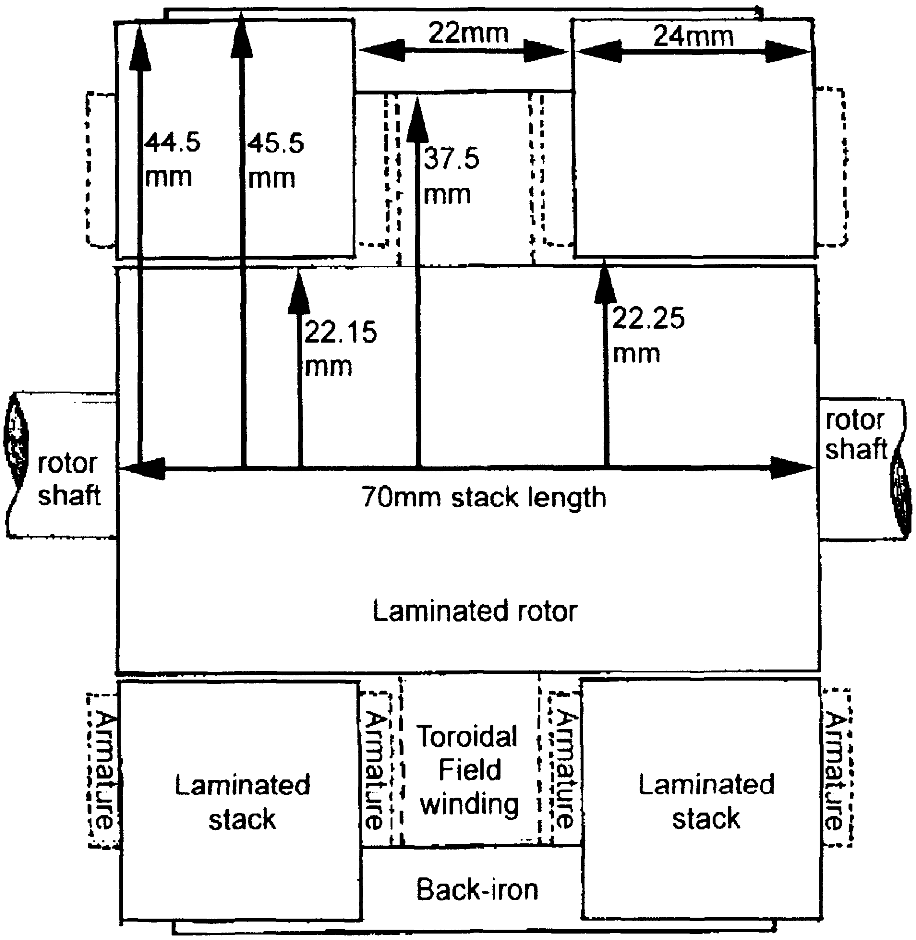


Figure 4.14 :- Surface area calculations for an armature coil.

TABLE 4.1 - DIMENSIONS WITHIN MACHINE

	Toroid	Per Phase	Total
Turns Wire Diameter	550 0.475mm	600 0.315mm	- -
Resistance	9.82ohm	2.615ohm	-
mmf	estimated	maximum	1478
Packing factor	0.711	0.3	-
Power wanted Speed wanted	- -	- -	300W 8000

4.7 INITIAL TEST RESULTS

4.7.1 WIRING AND BACK EMF TEST RESULTS

The stator stacks were initially in perfect alignment. For one stack, when one diagonally opposite set of coils for one phase was energised, they were connected so they gave a North Pole at the stator pole face. The equivalent pair on the other stack was connected so that they gave a South Pole at the stator pole face. This meant that, at an aligned rotor position with only these coils excited, the flux produced was axial along the rotor (i.e. the fluxes from each stack complemented each other, not cancelling each other out). The other poles were excited so they would give opposing pole faces to that on the same stack. The net result was that, when one phase was energised, one diagonally opposite pole pair was equivalent North Pole and the other equivalent South Pole whilst the other stack was the same except that the equivalent Poles were 90° shifted. When the field was energised the rotor had one end as a South Pole and the other as a North Pole. When one phase was energised and the flux from each diagonally opposite pole pair had two possible effects. Where the rotor pole and stator pole were of the same magnetic polarity, repulsion occurred and the rotor was forced away and, where the magnetic polarities were different, then attraction occurred and the force generated brought the rotor towards that stator pole pair. Thus each stator pole was used to generate torque.

In each stack for each phase the two pairs of diagonally opposite coils had back emfs which were inverted forms of the other. By connecting the coils such that the sense of one is the same as the sense of the coil on the adjacent stator pole it was possible to add the back emfs. This was done for each phase.

The design was connected to a dc motor and the back emf was measured from the stator windings of each phase (i.e. the motor was tested as a generator).

The back emf waveform was not square wave, as ideal, but was in fact very peaky and was probably due to the rotor having a graded design. The magnitude of the back emf was half that which was expected. The possible causes of reduction were:-

1 The value of reluctance for the magnetic circuit was approximated by using mean paths. The true flux paths are unknown and the μ_r value of steel was approximated. The use of laminations had not been accounted for.
2 Flux leakage was assumed to be zero.

4.7.2 MOTOR TESTING

Tests were carried out on the machine at no load (configuration as bipolar excitation with bifilar windings). The machine was not suitable for torque testing. It was very prone to overheating as the load increased (*Figure 4.15*). The winding temperature reached 80°C at no load.

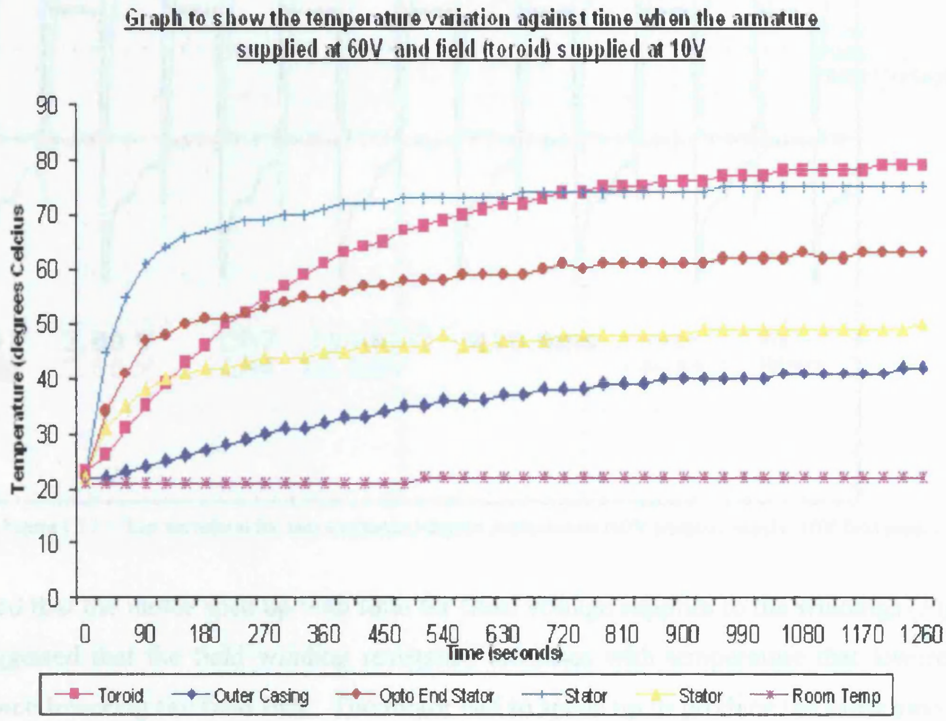


Figure 4.15 :- Temperature measurements during tests.

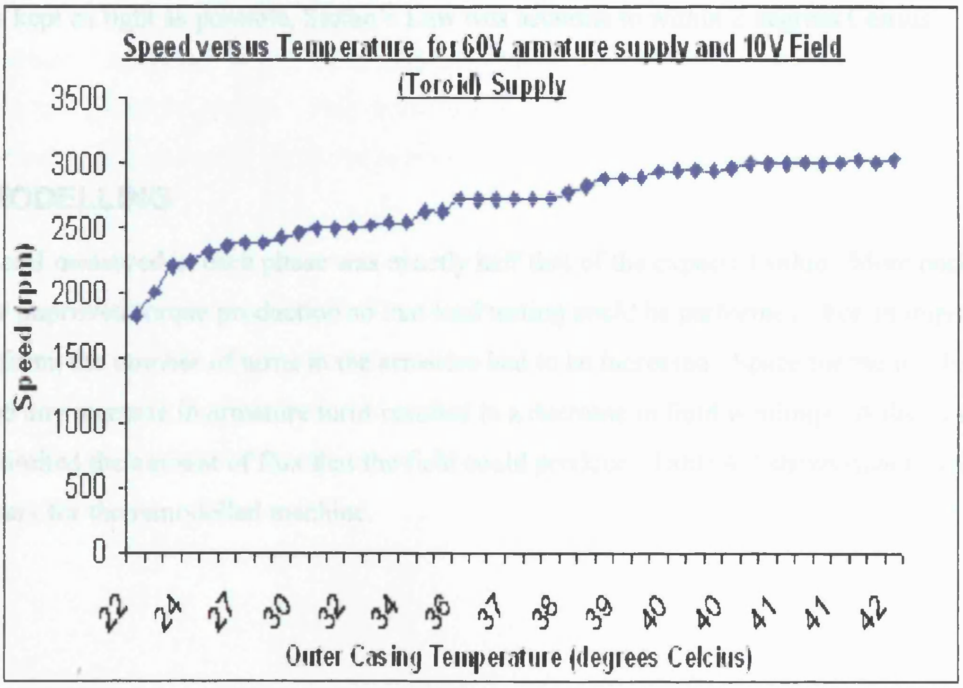


Figure 4.16 :- Speed of machine increases with temperature for fixed supply.

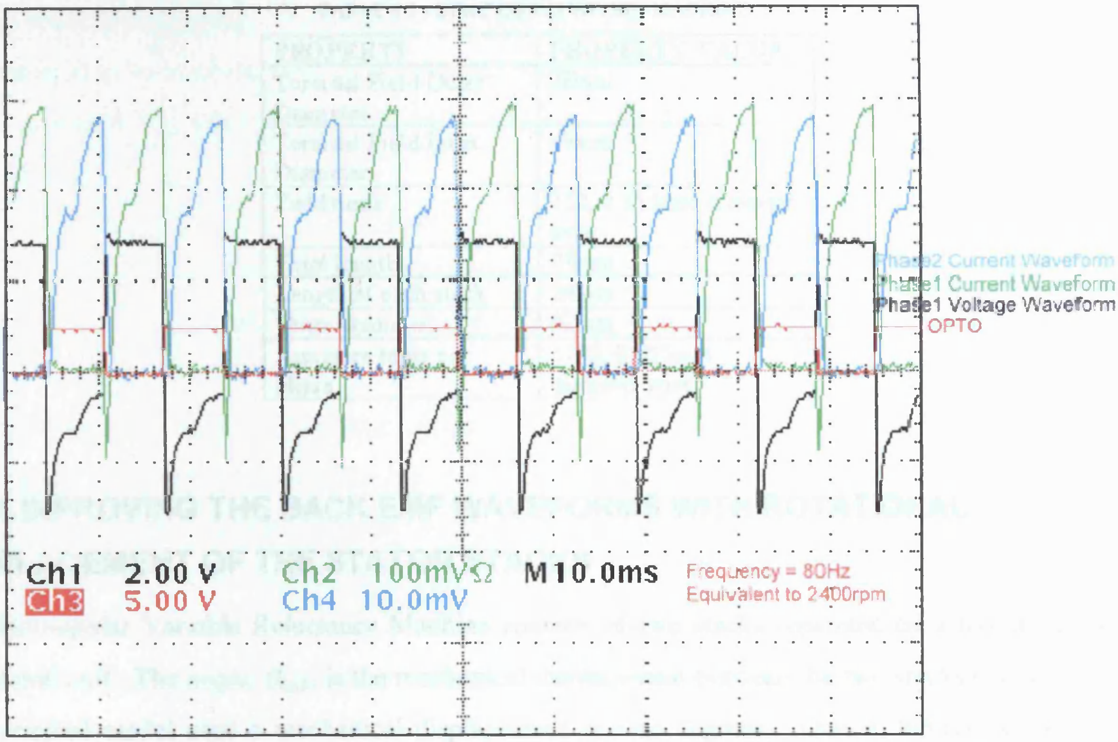


Figure 4.17 :- Test waveform for zero mechanical degree displacement (60V armature supply, 10V field supply).

It was noted that the motor sped up with time for fixed voltage supplies to the windings (*Figure 4.16*). It was suggested that the field winding resistance increases with temperature that lowered the field current hence lowering the field flux. The motor had to speed up to produce the same amount of back emf. *Figure 4.17* shows an experimental waveform for a no load test. It was found that, if the packing factor was kept as tight as possible, Stefan's Law was accurate to within 2 degrees Celsius.

4.8 REMODELLING

The back emf measured in each phase was exactly half that of the expected value. More back emf was needed for improved torque production so that load testing could be performed. For an improved back emf waveform, the number of turns in the armature had to be increased. Space for the windings was limited and any increase in armature turns resulted in a decrease in field windings. A decrease in field windings limited the amount of flux that the field could produce. Table 4.2 shows dimensions and turn numbers for the remodelled machine.

TABLE 4.2 - DIMENSIONS WITHIN MACHINE

PROPERTY	PROPERTY VALUE
Toroidal Field Outer Diameter	78mm
Toroidal Field Inner Diameter	47mm
Field turns	750, 0.315mm diameter wire
Total length	73mm
Length of each stack	24mm
Outer diameter	92mm
Armature turns per phase	1192, 0.315mm diameter wire

4.8.1 INPROVING THE BACK EMF WAVEFORMS WITH ROTATIONAL DISPLACEMENT OF THE STATOR STACKS

The Homopolar Variable Reluctance Machine consists of two stacks separated by a toroidal axially orientated coil. The angle, θ_{shift}, is the mechanical displacement between the two stacks (*Figure 4.18*). The original model used a mechanical displacement of zero degrees. That is, the two stacks were perfectly aligned with each other.

The back emf in each coil stator pole coil is produced by passing current through the field winding and turning the rotor (as in the case of a generator). This machine has an electrical angle of twice the mechanical angle.

Zero shift degrees is defined as the initial angular condition. Each stack has windings associated with each of the two armature phases. This initial condition is such that the back emf generated in the armature windings of one stack would be in phase with the same phase armature windings in the other stack (*Figure 4.19*).

To expand this concept further, consider the coils in one stack as being labelled as A, A', B, B' and in the other stack as C, C', D, D' for each phase (*Figure 4.18*). In each stack the diametrically opposing coils are connected such that any back emf produced in each add (not cancel each other out). The sum is termed as V_{xn} where x is the coil pair (AA', BB', CC', DD') and n was the phase number (1 or 2) – as shown in (4.7).

$$\begin{aligned} A + A' &= V_{An} \\ B + B' &= V_{Bn} \\ C + C' &= V_{Cn} \\ D + D' &= V_{Dn} \qquad \text{where } n = 1, 2 \text{ (the phase number)} \end{aligned} \tag{4.7}$$

In one stack, for each phase, the back emf V_{An} adds to V_{Bn} due to the way the coils are connected. V_{An} is connected to be in phase with V_{Bn} at zero shift degrees. In the other stack, for each phase, the back emf V_{Cn} adds to V_{Dn} due to the way the coils are connected. V_{Cn} is also connected to be in phase with V_{Dn} (4.8).

$$\text{Back emf per phase armature in Stack}_{AB} = V_{An} - V_{Bn} \tag{4.8}$$
$$\text{Back emf per phase armature in Stack}_{CD} = V_{Cn} - V_{Dn}$$

For each phase, the back emf generated in one stack and the back emf generated in the other stack are represented by vectors. For zero mechanical degrees the back emfs from each stack are in phase and so are zero electrical degrees different (4.9). The connections between each stack are such that for each phase there is summation of the vectors at this angle.

$$\text{Zero degrees shift total back emf per phase} = (V_{An} - V_{Bn}) + (V_{Cn} - V_{Dn}) \tag{4.9}$$

$$\text{Zero degrees shift total back emf in Phase 1} = (V_{A1} - V_{B1}) + (V_{C1} - V_{D1})$$

$$\text{Zero degrees shift total back emf in Phase 2} = (V_{A2} - V_{B2}) + (V_{C2} - V_{D2})$$

The back emf per phase is thus $V_{phase\ n} = V_{stack1} + V_{stack2}$

where $V_{stack1} = (V_{An} - V_{Bn})$

$V_{stack2} = (V_{Cn} - V_{Dn})$

and n is the phase number

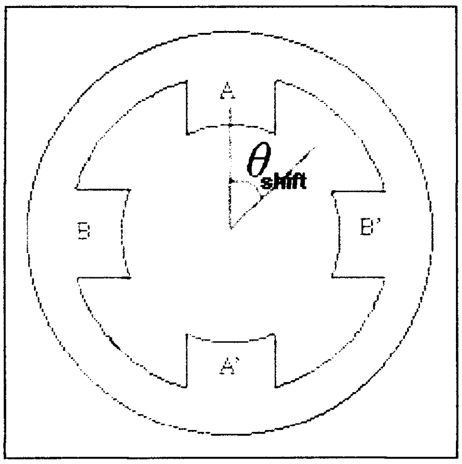


Figure 4.18 :- Shift angle displacement between stacks, θ_{shift}.

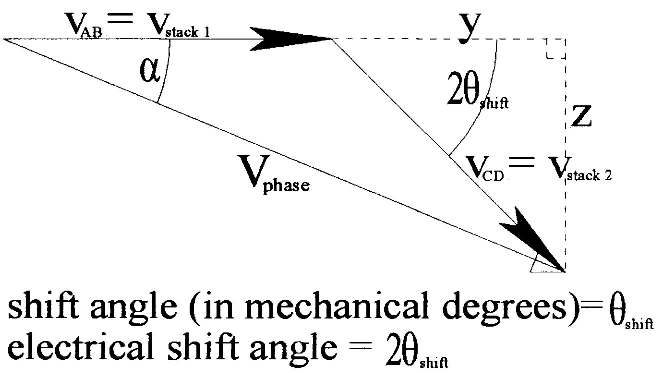


Figure 4.19 :- Vector theory in relation to displacement angle between stacks.

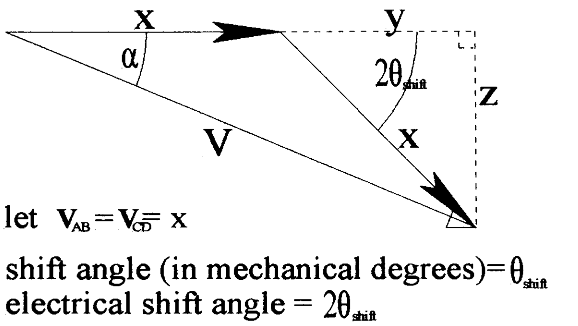


Figure 4.20 :- Vector theory simplified.

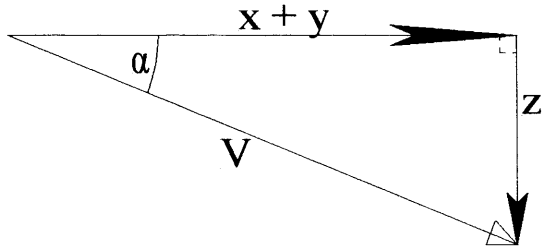


Figure 4.21 :- Vector theory defining electrical angle shift in back emf waveform.

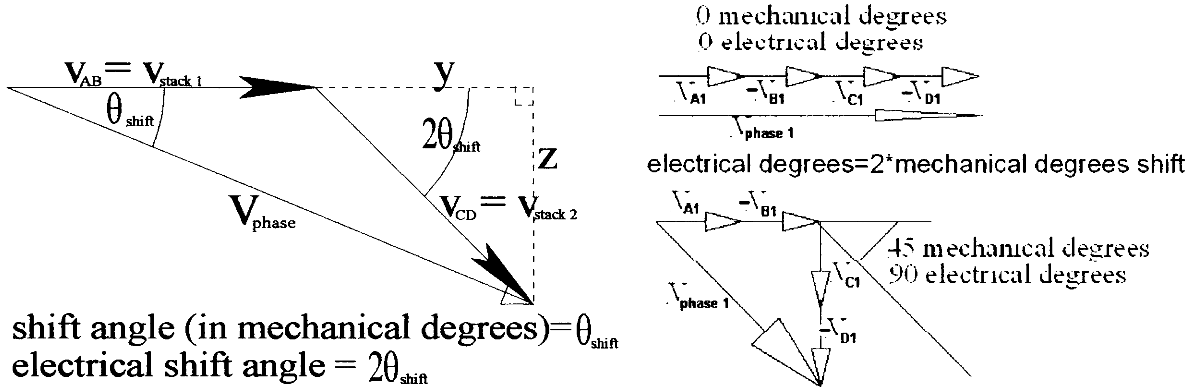


Figure 4.22 :- Electrical displacement in relation to mechanical displacement and two examples.

This is shown in vector form below whereby the mechanical shift angle θ_{shift} is shown as an electrical angle of displacement $2\theta_{\text{shift}}$ (*Figure 4.19*). This is due to the machine having two rotor poles and four stator poles per stack. As shown, the total voltage for the phase, V_{phase}, is shown as being displaced by an electrical angle α.

The first stack is regarded, in this case, as the reference stack, so the second stack is displaced by the electrical angle $2\theta_{\text{shift}}$.

If $V_{stack1} = V_{stack2} = x$ (in magnitude value) and $V_{phase} = V$ (for simplicity) for a given phase, the vector diagram simplifies as below (4.10). V_{stack2} is split into y and z components so that V_{phase} can be calculated by geometric calculations. The relationship between α and θ_{shift} is then found (*Figures 4.20, 4.21*).

Calculations :- $y = x\cos(2\theta_{\text{shift}})$

$z = x\sin(2\theta_{\text{shift}})$

$$\tan\alpha = \left(\frac{z}{(x+y)}\right)$$

$$\tan\alpha = \left(\frac{x\sin(2\theta_{shift})}{x + x\cos(2\theta_{shift})}\right) = \left(\frac{x\sin(2\theta_{shift})}{x(1+\cos(2\theta_{shift}))}\right) = \left(\frac{\sin(2\theta_{shift})}{1+\cos(2\theta_{shift})}\right)$$

$$\tan\alpha = \left(\frac{2\sin\theta_{shift}\cos\theta_{shift}}{(1+2\cos^2\theta_{shift}-1)}\right) = \left(\frac{2\sin\theta_{shift}\cos\theta_{shift}}{2\cos^2\theta_{shift}}\right) = \left(\frac{2\sin\theta_{shift}\cos\theta_{shift}}{2\cos\theta_{shift}\cos\theta_{shift}}\right)$$

$$\tan\alpha = \left(\frac{\sin\theta_{shift}}{\cos\theta_{shift}}\right) = \tan\theta_{shift}$$

hence $\alpha = \theta_{shift}$

Also $\sin\alpha = \dfrac{z}{V}$

$V = \dfrac{x\sin(2\theta_{shift})}{\sin\alpha}$ but $\alpha = \theta_{shift}$

therefore

$$V_{phase} = V = x\left(\frac{\sin(2\theta_{shift})}{\sin(\theta_{shift})}\right) \tag{4.10}$$

The overall electrical angle of displacement for each phase equals the mechanical angle of displacement even though the electrical angle between each stack is twice that of the mechanical angle (*Figure 4.22*). The back emf in each phase is shown by vector diagrams to be a function of the mechanical angle in terms of overall electrical displacement. As the mechanical angle varies so too does the effective surface area of the stator pole faces over which the flux passes through. The back-emf traces are then variable in magnitude and shape.

The theory of vectors is supplemented by utilising the back emf waveform shape itself.

The back emf trace for one stack for one armature phase was recorded for one electrical cycle. This waveform is the same shape as that of the same phase on the other stack (disregarding any phase difference).

For one armature phase data points were taken and a plot of back emf in each stack against rotor position was replicated. The sum of the two waveforms (superimposed) gave the total back emf waveform for that phase winding. The Stack 1 plot was taken as the reference waveform. The Stack 2 plot was phase shifted against the Stack 1 plot according to the mechanical angle of difference between the two stacks, θ_{shift}. Dependent on the shape of the waveform for each stack, the overall waveform shape for each phase was changed as the mechanical angle, θ_{shift}, was altered.

4.8.2 SELECTION OF THEORETICAL IDEAL ANGLES

For smooth torque production to be ensured, the back emf waveform shape should be as close to a square wave as possible (the ideal condition). The rate of change of permeance affects the rate of change of flux as the rotor turns and this itself is affected by the rotor and stator pole shapes. But, with this type of motor, the angular position between the two stacks was used to manipulate the back emf trace to obtain a squarer shaped trace. To aid the determination of the best value of θ_{shift} the term crest factor is introduced.

Crest factor is used to compare the ideal back emf waveforms. It is the ratio of the peak to rms of the back emf curve. A square wave is the ideal and has a value of one. Hence a value closer to one is more idealistic. A square wave back emf trace would also represent smooth torque production. The crest factor is only applicable to waveforms with a repetitive cycle.

For the ideal square wave the crest factor is 1, for a sinusoidal wave the crest factor is $\sqrt{2}$ (1.4142). It is regarded that the nearer the crest factor is to 1 the better the waveform.

The other remaining item of interest within the back emf curve is the average of the modulus of the half cycle mean for the total back emf waveform for the armature phase.

It was then possible for a plot of peak, rms, averaged modulus half cycle mean back emfs and crest factor against stack displacement angle, θ_{shift} (*Figures 4.23, 4.24, 4.25, 4.26*).

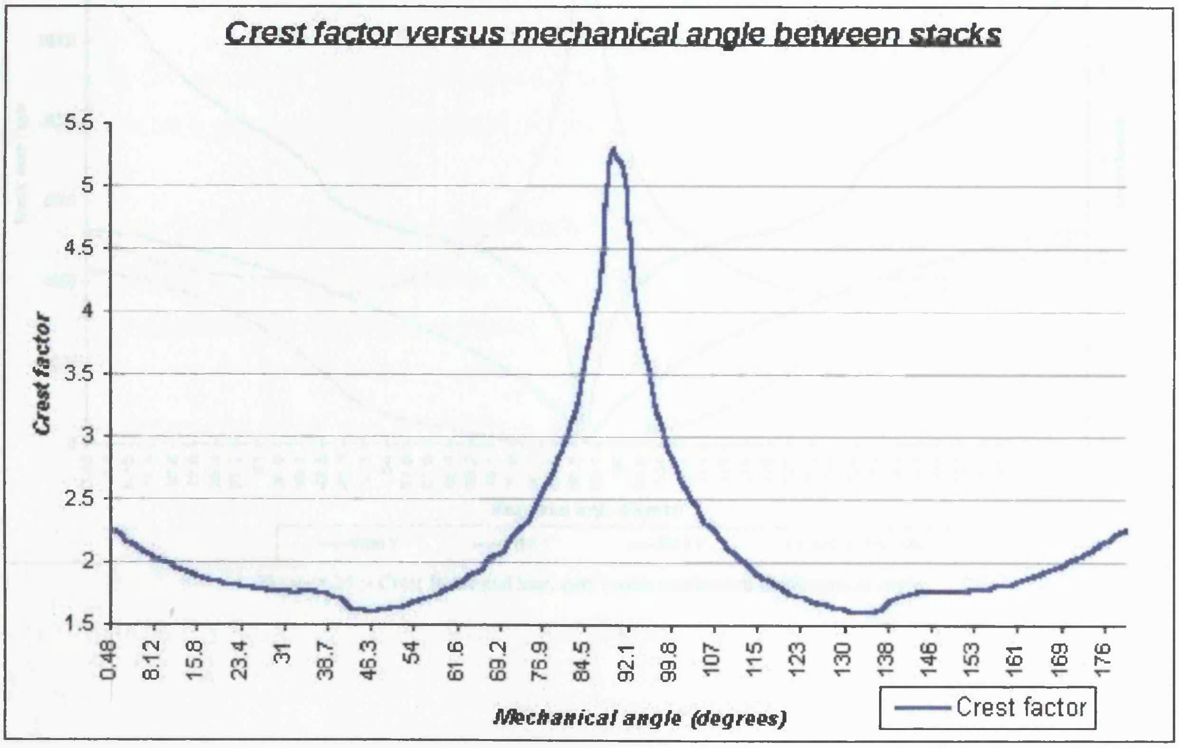


Figure 4.23 :- Crest factor against mechanical displacement angle.

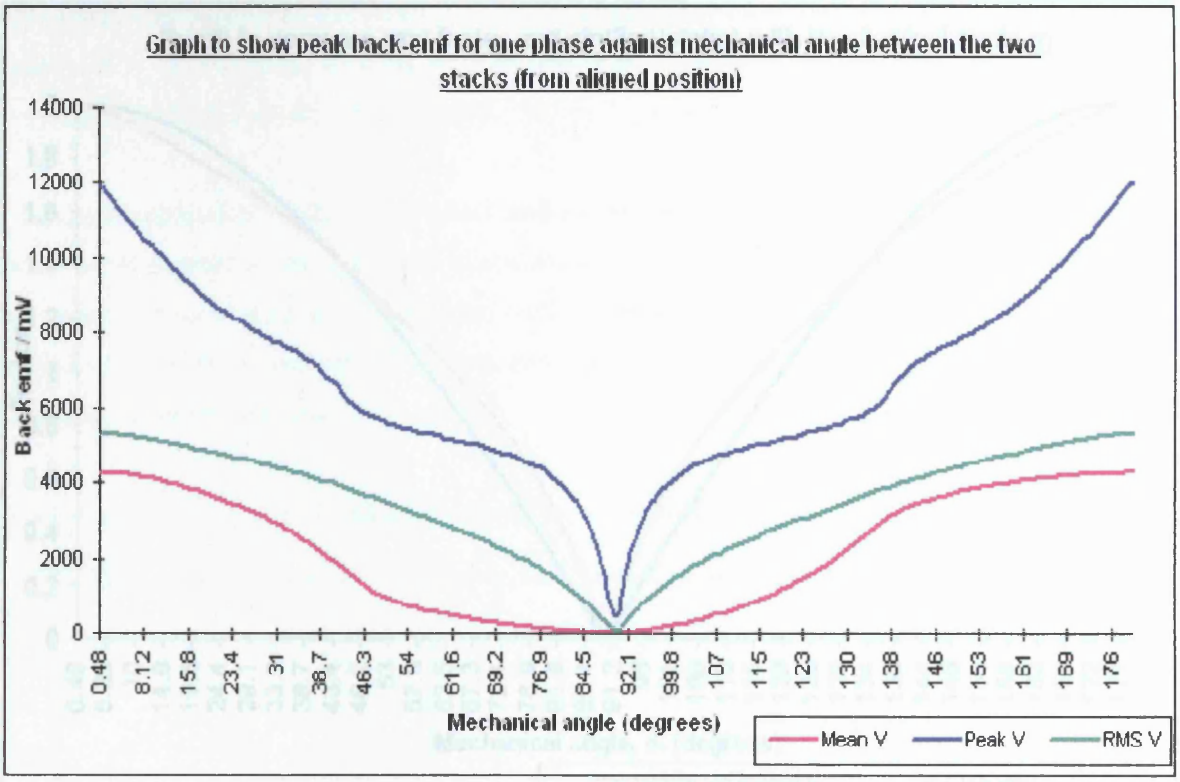


Figure 4.24 :- Back emf magnitudes against mechanical displacement angle.

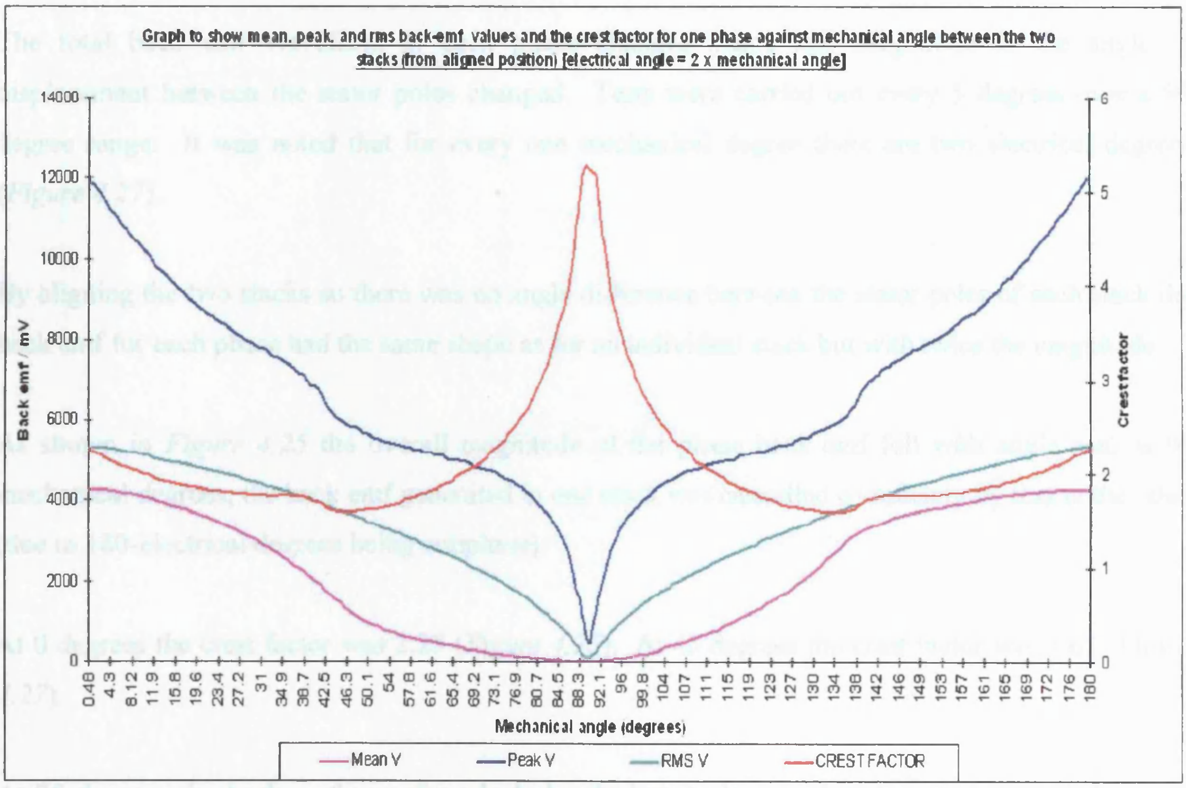


Figure 4.25 :- Crest factor and back emf versus mechanical displacement angle.

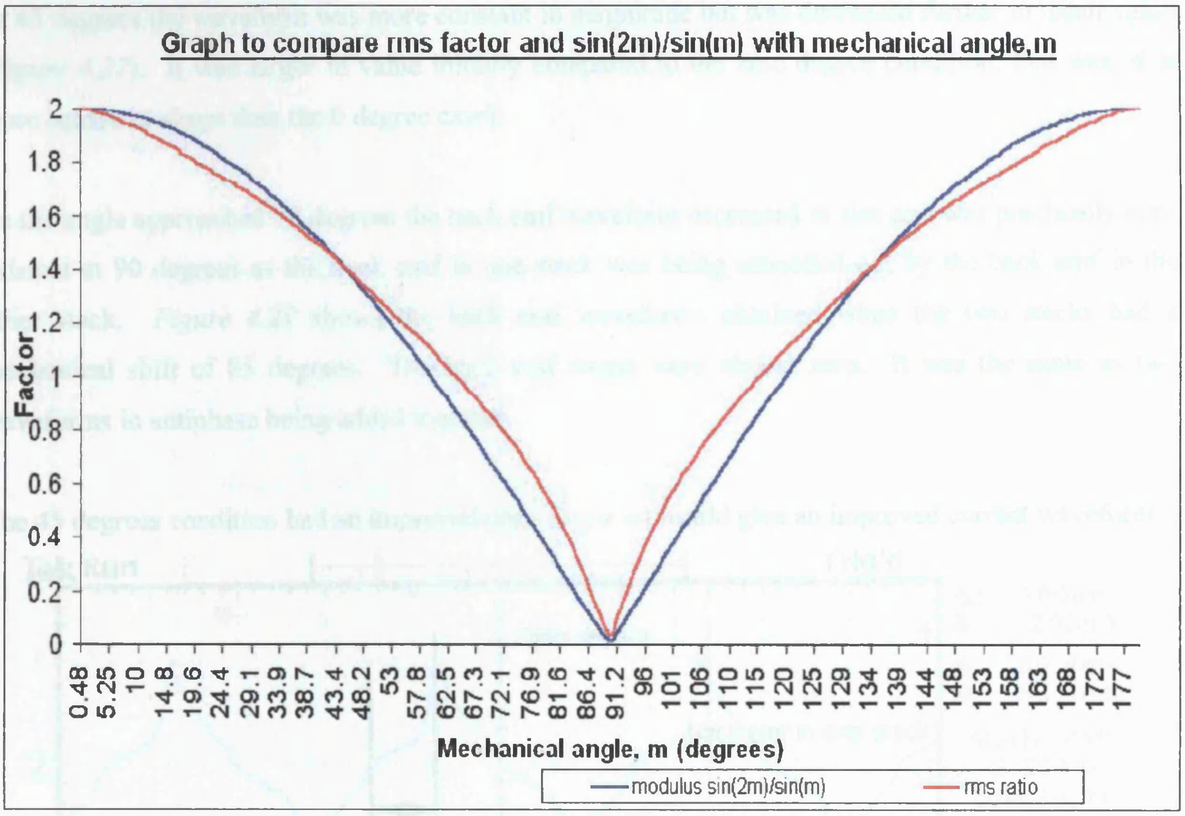


Figure 4.26 :- Predicted back emf versus rms back emf with mechanical angle.

The total back emf waveform in each phase changed shape and magnitude as the angle of displacement between the stator poles changed. Tests were carried out every 5 degrees over a 90 degree range. It was noted that for every one mechanical degree there are two electrical degrees (*Figure 4.27*).

By aligning the two stacks so there was no angle difference between the stator poles of each stack the back emf for each phase had the same shape as for an individual stack but with twice the magnitude.

As shown in *Figure 4.25* the overall magnitude of the phase back emf fell with angle and, at 90 mechanical degrees, the back emf generated in one stack was cancelled completely by that in the other (due to 180-electrical degrees being antiphase).

At 0 degrees the crest factor was 2.25 (*Figure 4.25*). At 45 degrees the crest factor was 1.62 (*Figure 4.27*).

At 25 degrees the back emf waveform had already lost its large spike characteristic and the mean value over a half cycle had lowered (*Figure 4.28*). It had a crest factor of 1.85.

At 45 degrees the waveform was more constant in magnitude but was decreased further in mean value (*Figure 4.27*). It was larger in value initially compared to the zero degree condition, that was, it is more square in shape than the 0 degree case).

As the angle approached 90 degrees the back emf waveform decreased in size and was practically non-existent at 90 degrees as the back emf in one stack was being cancelled out by the back emf in the other stack. *Figure 4.29* shows the back emf waveforms obtained when the two stacks had a mechanical shift of 85 degrees. The back emf traces were almost zero. It was the same as two waveforms in antiphase being added together.

The 45 degrees condition had an improved crest factor so would give an improved current waveform.

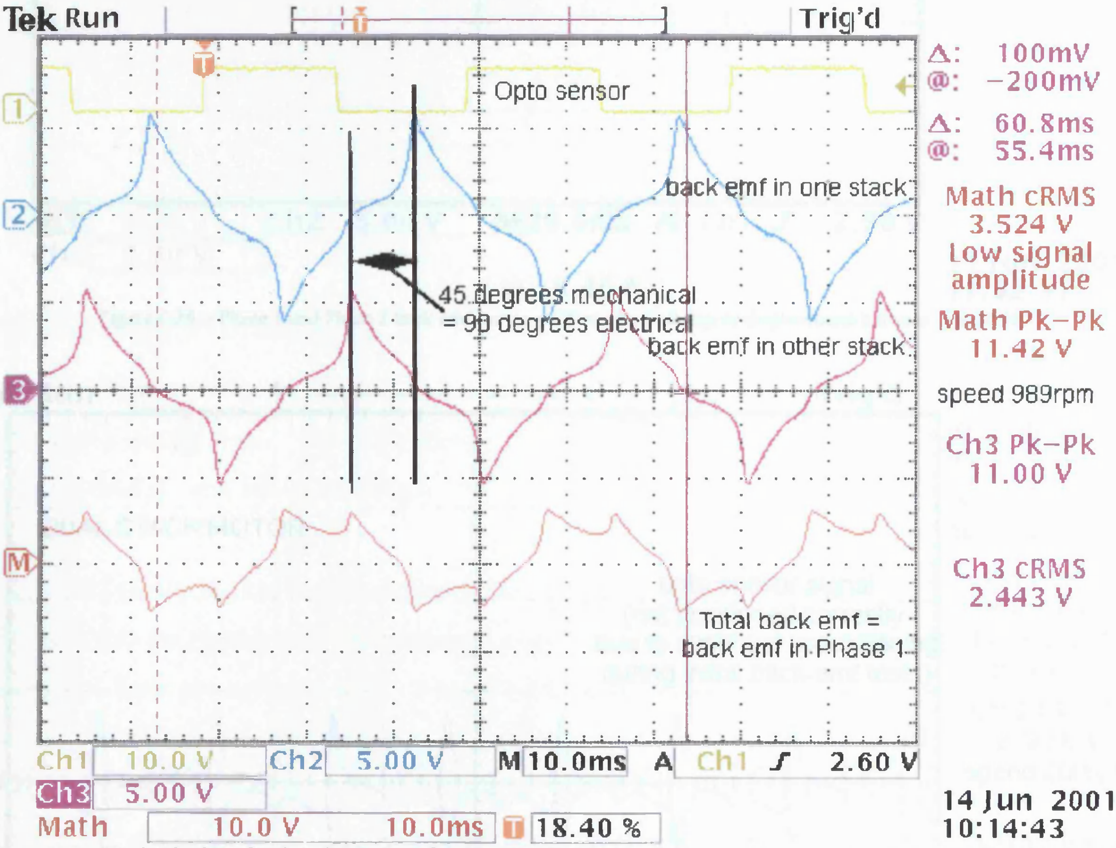


Figure 4.27 :- Stack 1 back emf and stack 2 back emf at 45 mechanical degree displacement between each stack and the resultant sum for Phase 1.

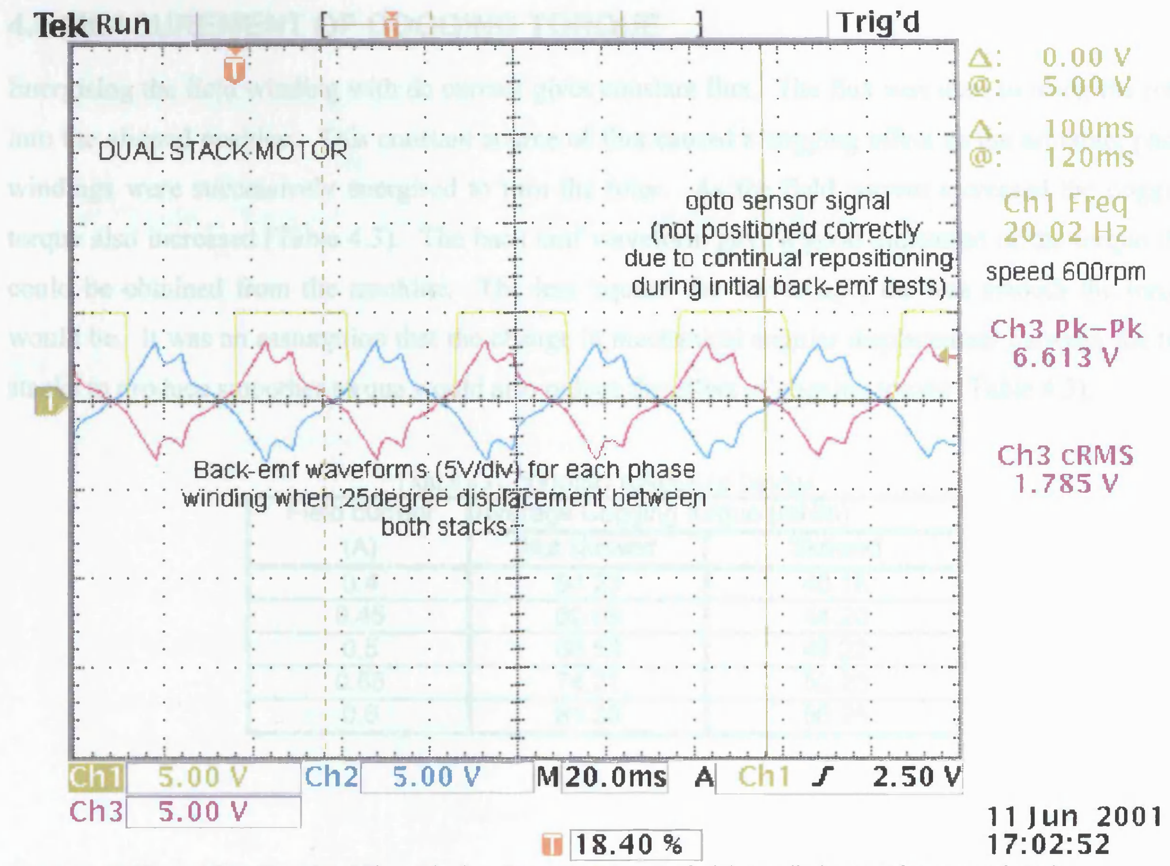


Figure 4.28 :- Phase 1 and Phase 2 back emf curves at 25 mechanical degree displacement between each stack.

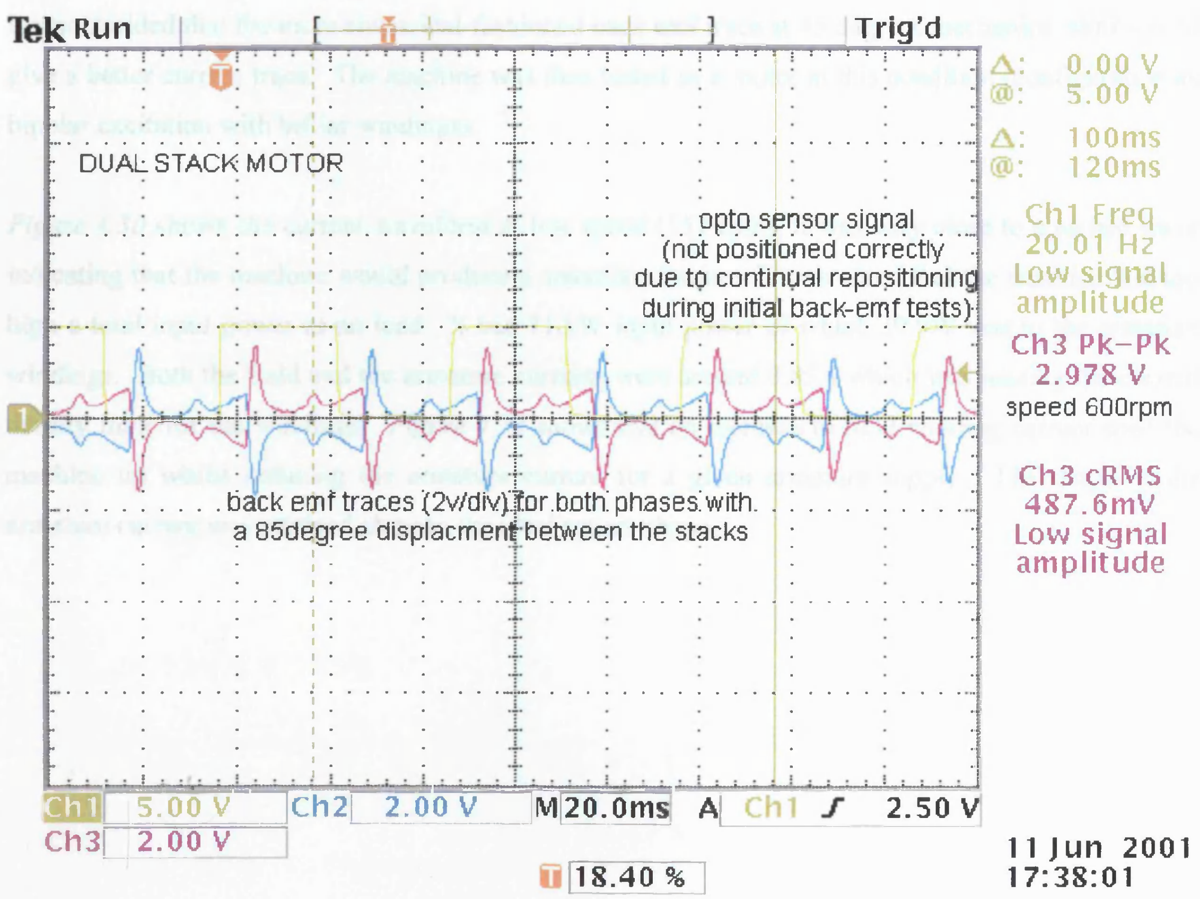


Figure 4.29 :- Phase 1 and Phase 2 back emf curves at 85 mechanical degree displacement between each stack.

4.9 MEASUREMENT OF COGGING TORQUE

Energising the field winding with dc current gives constant flux. The flux was used to bring the rotor into the aligned position. This constant source of flux caused a cogging effect as the armature phase windings were successively energised to turn the rotor. As the field current increased the cogging torque also increased (Table 4.3). The back emf waveform gave a good indication of the torque that could be obtained from the machine. The less 'square' the waveshape, the less smooth the torque would be. It was an assumption that the change in mechanical angular displacement between the two stacks to produce smoother torque would also reduce the effect of cogging torque (Table 4.3).

TABLE 4.3 – COGGING TORQUE OF DSVRM

Field current (A)	Average Cogging torque (mNm)	
	Not skewed	Skewed
0.4	54.23	40.18
0.45	60.58	44.20
0.5	66.53	48.22
0.55	74.21	52.23
0.6	81.30	56.25

4.10 LOAD TEST RESULTS

It was decided that the more sinusoidal-fashioned back emf trace at 45 degrees mechanical shift would give a better current trace. The machine was then tested as a motor at this condition (configuration as bipolar excitation with bifilar windings).

Figure 4.30 shows the current waveform at low speed (551 rpm). It was very close to a square wave indicating that the machine would produce a smoother torque. It was noted that the machine had too high a total input power at no load. It had 71.5W input power of which 39.9W was to the armature windings. Both the field and the armature currents were around 0.85A which was nearing the current density limit for the windings. *Figure 4.31* shows that an increase in field winding current sped the machine up whilst reducing the armature current for a given armature supply. The shape of the armature current was retained close to the ideal square shape.

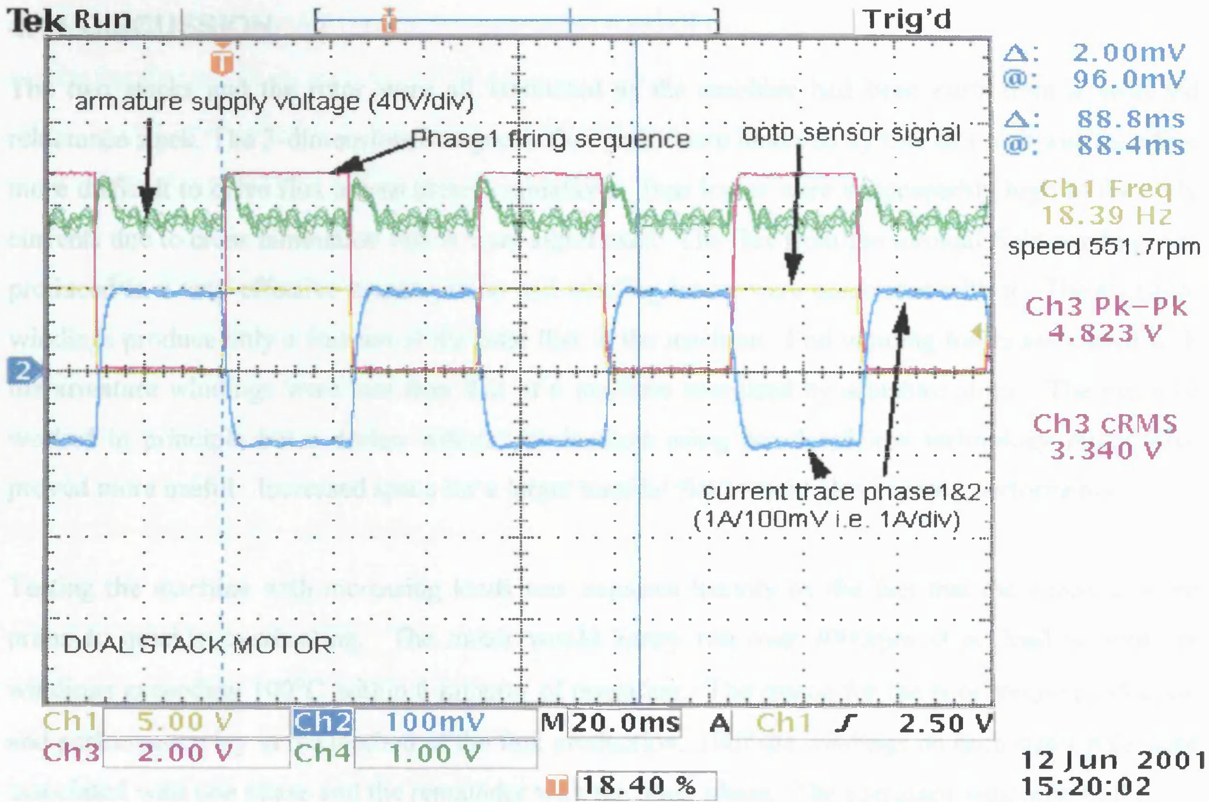


Figure 4.30 :- Current trace at 551rpm (45 mechanical degrees).

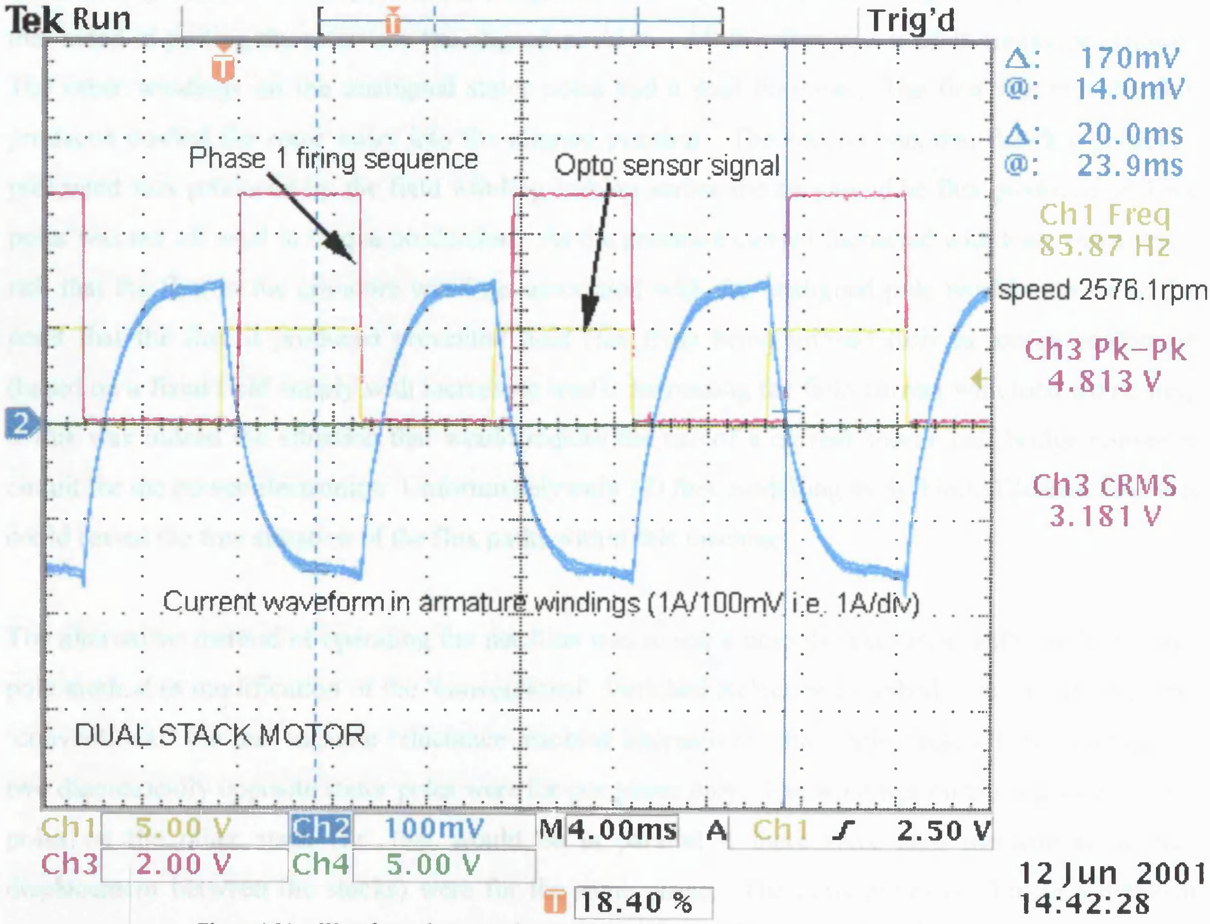


Figure 4.31 :- Waveforms demonstrating machine running at 2576rpm (45 mechanical degrees).

4.11 DISCUSSION

The two stacks and the rotor were all laminated as the machine had been built from a switched reluctance stack. The 3-dimensional magnetic flux paths were hindered by this fact as it was therefore more difficult to drive flux across these laminations. Iron losses were unacceptably high as the eddy currents due to cross lamination fluxes were significant. The flux from the toroidal field winding was produced in a very effective manner as no end winding losses were associated with it. The armature windings produce only a fraction of the total flux in the machine. End winding losses associated with the armature windings were less than that of a machine energised by armature alone. The machine worked in principle but a design without laminations using powdered iron technology might have proved more useful. Increased space for a larger toroidal field would also improve performance.

Testing the machine with increasing loads was impaired heavily by the fact that the windings were prone to quickly overheating. The motor would barely run over 3000rpm at no load without the windings exceeding 100°C within 6 minutes of operation. The reason for the poor torque production and performance lay in the method of the flux production. Half the windings on each stator pole were associated with one phase and the remainder with the other phase. The energised windings on the two diametrically opposite stator poles for the aligned position (for that phase energisation) generated flux that aided in pulling the rotor into the aligned position. All this flux was used in torque production. The other windings on the unaligned stator poles had a dual function. The first was that the flux produced pushed the rotor away into the aligned position. The second was that the flux produced prevented flux produced by the field winding leaking across the air gap. The flux produced in these poles was not all used in torque production. As the armature current increased with load, there was a risk that the flux in the armature windings associated with the unaligned pole would increase to the point that the flux it produced prevented field flux from being utilised fully in torque production (based on a fixed field supply with increasing load). Increasing the field current with load would help if this was indeed the situation that would require the use of a current source half bridge converter circuit for the power electronics. Unfortunately only 3D flux modelling using Finite Element Analysis could reveal the true situation of the flux paths within this machine.

The alternative method of operating the machine was to use a unipolar excitation with single coil per pole method (a modification of the 'conventional' Switched Reluctance method – explaining the term 'conventional' for this variable reluctance machine alternative). For each stack, all the windings in two diametrically opposite stator poles were for one phase only. The windings on the equivalent stator poles on the other stack (i.e. that would be in parallel if there were zero mechanical degrees displacement between the stacks) were for the same phase. The coils produced flux in a different manner than a conventional Switched Reluctance machine in that the flux from each stator pole set of windings aided the flux produced by the field winding (push flux into or out of the rotor, depending on the magnetic polarity of the rotor due to the field). All the armature flux was used in flux production (excepting that which escaped as flux leakage in the unenergised stator poles). This method provided better torque production and hence gave a different performance characteristic. The machine had unipolar current with unipolar flux in the stator poles rather than the bipolar flux in the stator poles.

The motor operated with the new winding configuration but with a newly optimised mechanical displacement between the two stacks of 29°. The new angle allowed, for a marginal increase in crest factor, a larger back emf waveform that still maintained a shape indicative of reasonably smooth torque production (*Figure 4.32*). The amplitude and shape of the back emf in each phase was identical to that for the bifilar version. The machine was tested at various speeds with various loads and field currents to optimise the phase firing timings and supply to the field and armature for best performance.

The initial tests involved optimising the phase timing sequence so that the input power for a given speed was minimised. There were three possible speed modes that were used (Chapter 3). Initially the back emf, after the opto sensor changed state at zero volt crossing, was small and the torque produced would be small (a fast rise in armature current was seen as well). After this, the back emf quickly rose and had a period of relative large value. This section was excellent for smoother torque production (shown by a levelling off of the current waveform to a constant value with little ripple). In the last section the back emf fell towards zero. Here the torque produced was reducing (and the current would be rising again). The middle section of relatively constant back emf was the best for torque production, and this was best achieved using the Normal Mode with pulsing (*Figure 4.33*). By optimising when each phase was energised during each opto sensor period, the maximum efficiency was obtained through optimised torque production. Ideally the current waveform would be a square wave for smooth torque production. As was shown, the optimised firing sequence at 4000rpm gave a current waveform that produced very smooth torque and left the machine operating audibly quieter than at zero degrees mechanical displacement between the stacks (no acoustic tests were performed). The machine operated at a much lower temperature, allowing load testing to be performed (*Figure 4.34*).

The effects of changing the field current for a fixed speed and load were investigated. A plot of efficiency against field supply showed that the machine increased its efficiency as the field supply increased (*Figure 4.35*). This suggested that having the field in series with the armature windings would improve the machines performance at higher loads.

This was confirmed in a plot of efficiency against load at 4000rpm where the field current was increased from 0A to 0.6A. Increasing the field current increased the efficiency (*Figure 4.36*). It was hard to ascertain whether there was a marked difference in improvement in efficiency with load for increasing field current as the graph seemed to imply towards the higher end of the loads. The initial remark was definitely true however.

Investigations into the effects of speed on efficiency versus load for a fixed field supply showed that the machine seemed to be less efficient at lower speeds. There was a marked difference in efficiency when at 3000rpm but only a small difference between 5000rpm and 4000rpm (*Figure 4.37*). The machine did not operate successfully at 6000rpm (its speed varied at the optimised timings so the tests were abandoned).

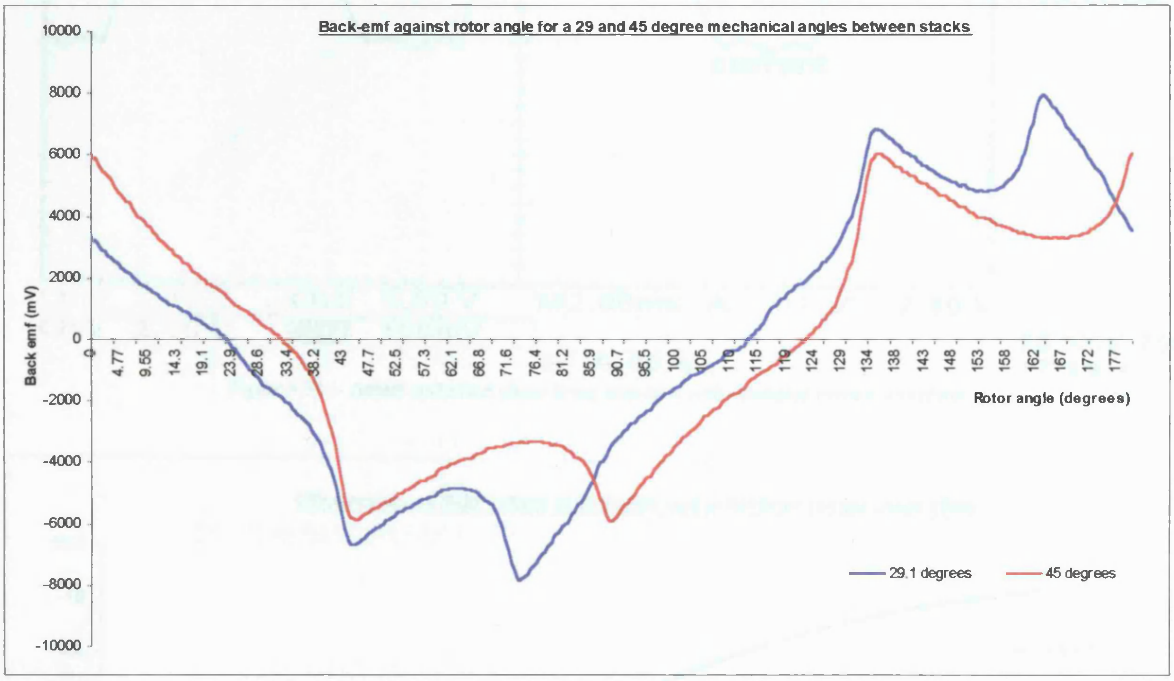


Figure 4.32 :- 29 and 45 degree mechanical displacement back emf waveforms for each phase.

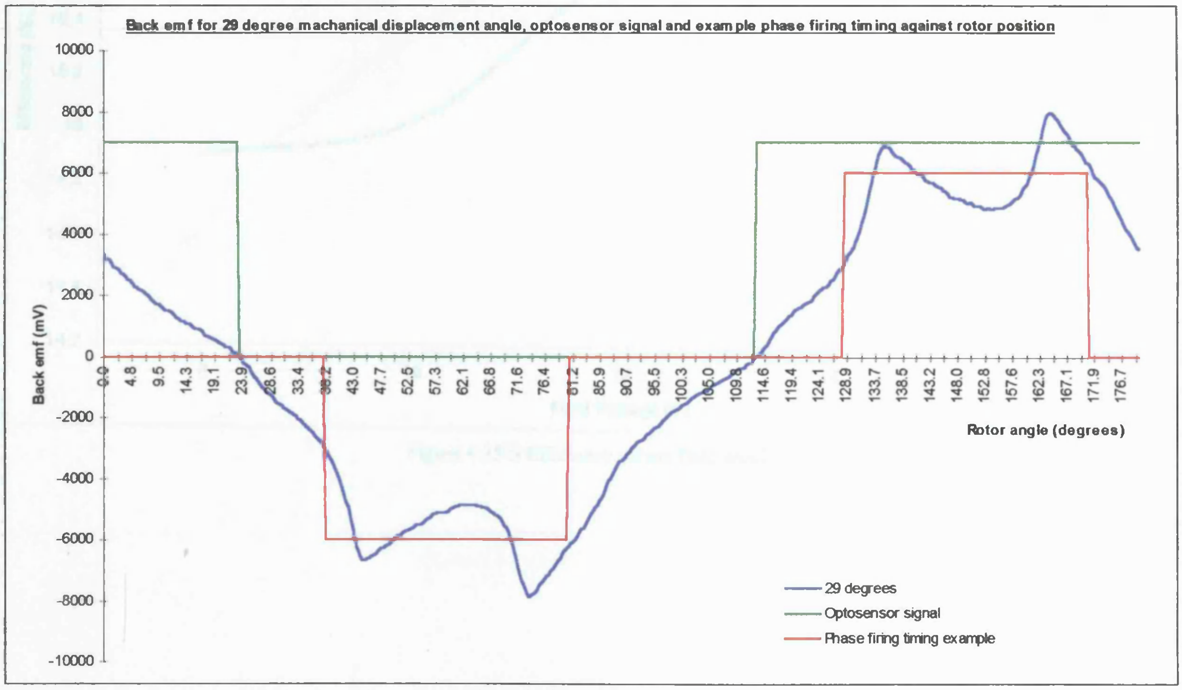


Figure 4.33 :-Possible armature firing timings for a 29 degree mechanical displacement angle.

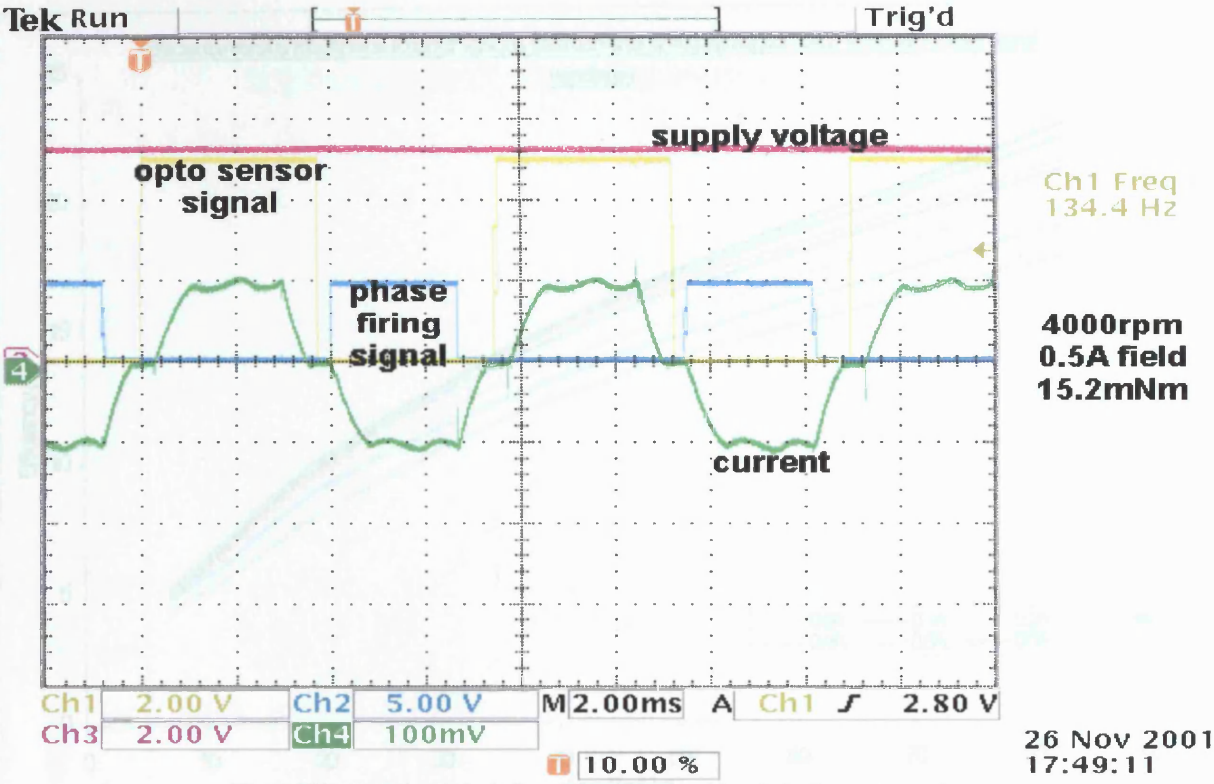


Figure 4.34 :- Actual optimised phase firing sequence with optimised current waveform.

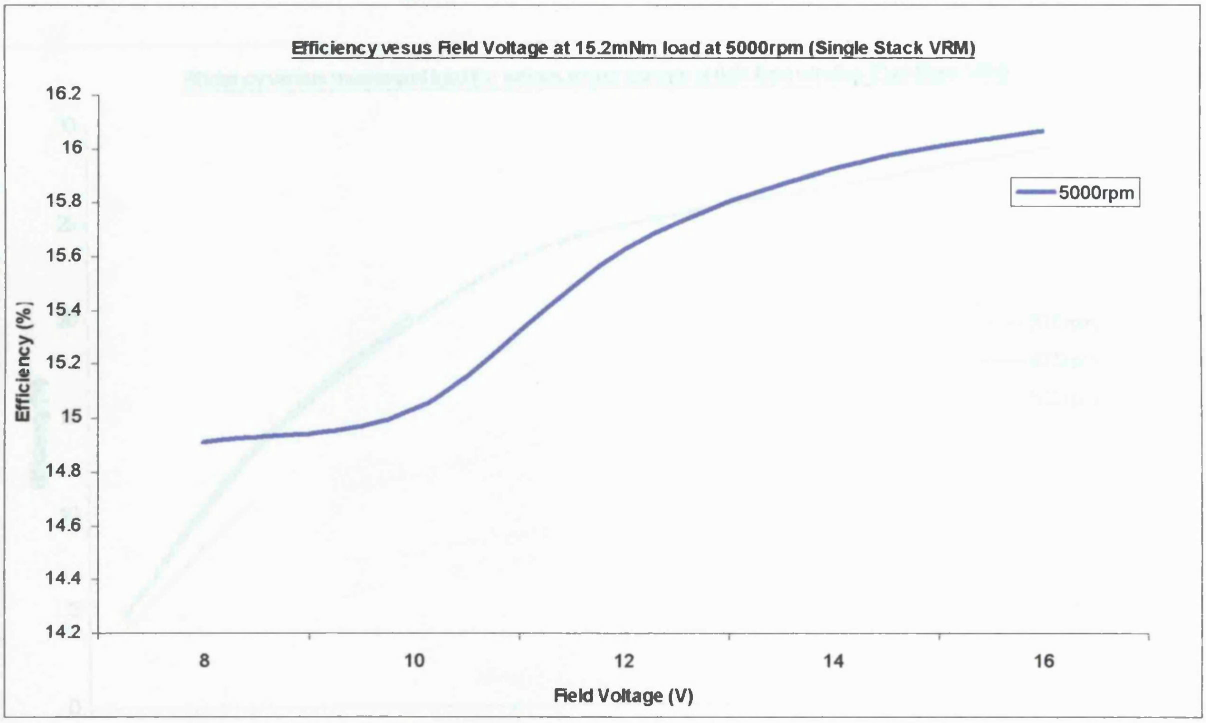


Figure 4.35 :- Efficiency versus field supply.

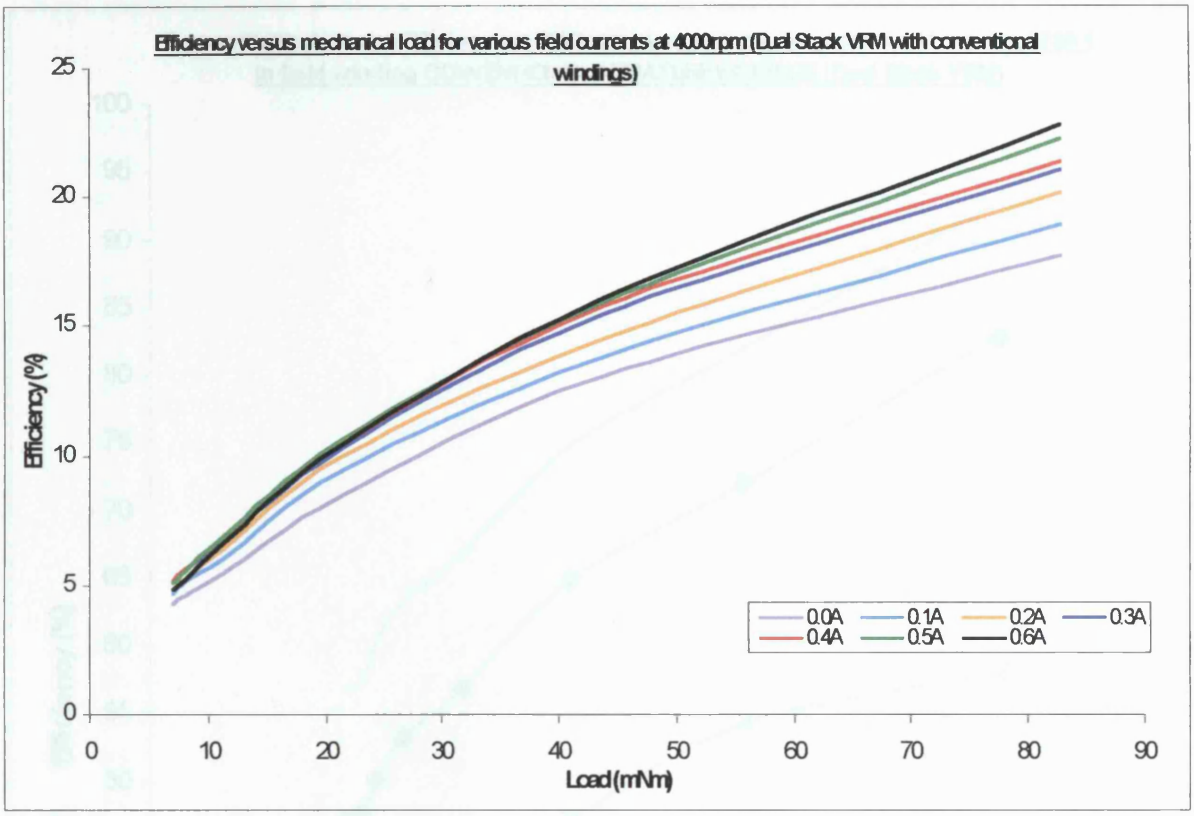


Figure 4.36 :- Field current effect on efficiency.

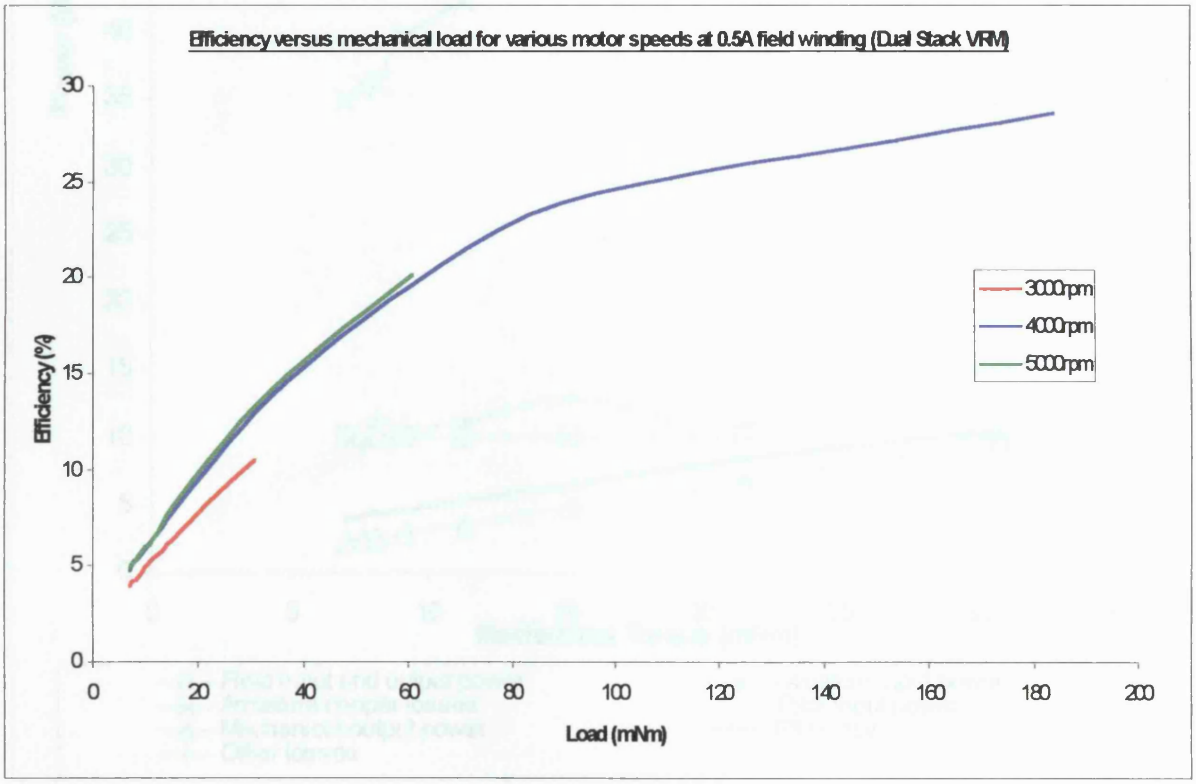


Figure 4.37 :- Speed effect on efficiency.

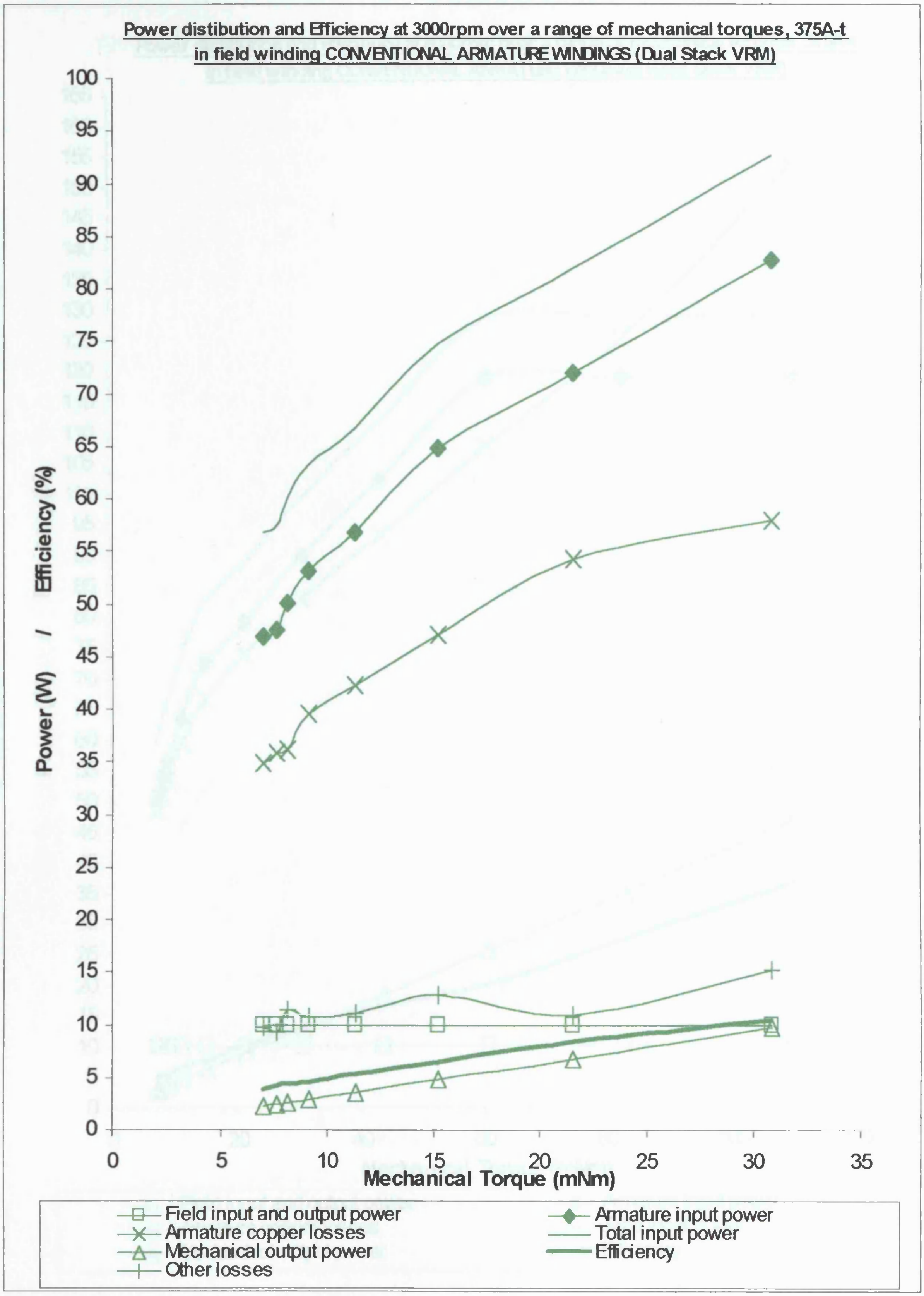


Figure 4.38 :- Input and output powers and efficiency against load for 3000rpm.

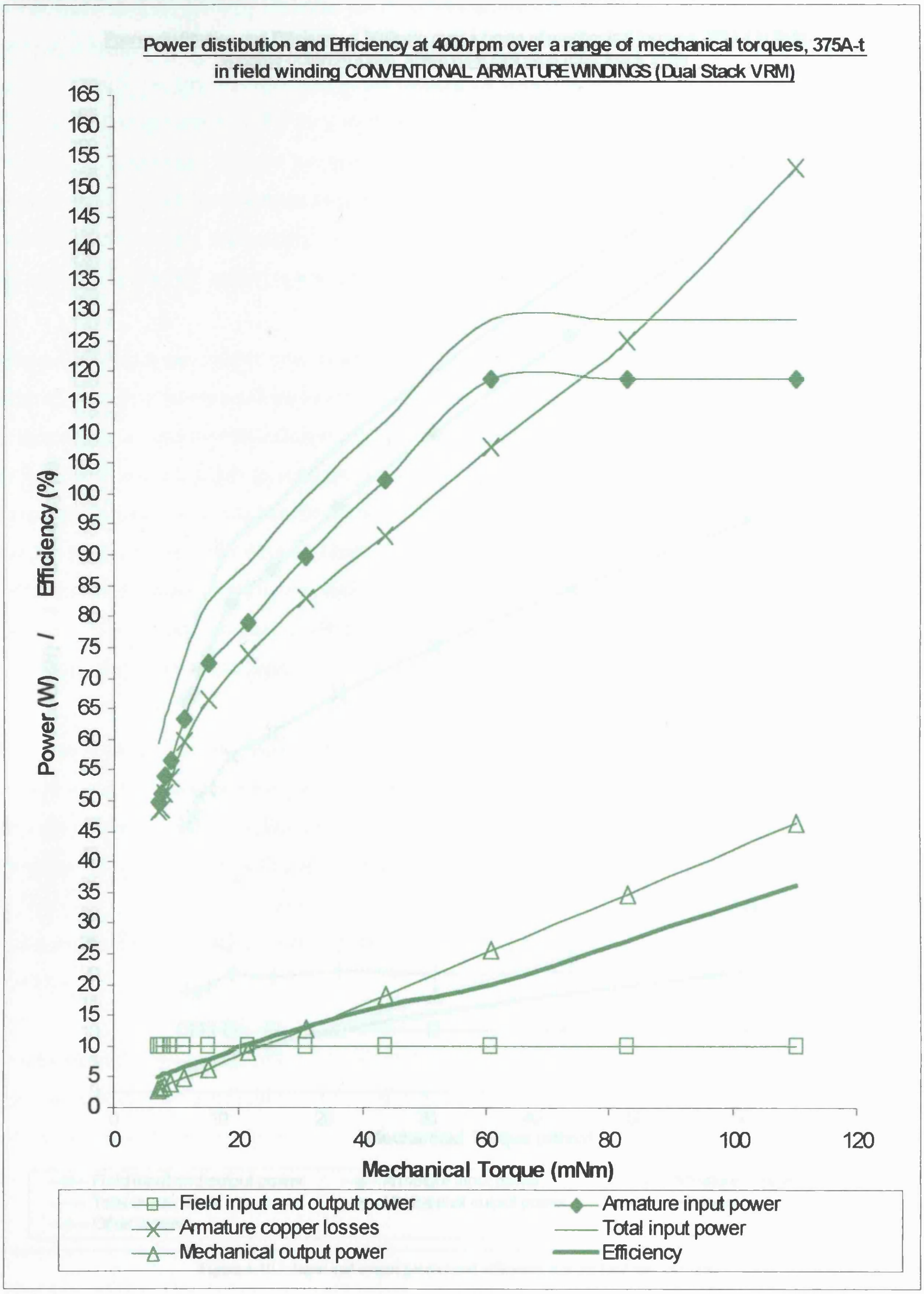


Figure 4.39 :- Input and output powers and efficiency against load for 4000rpm.

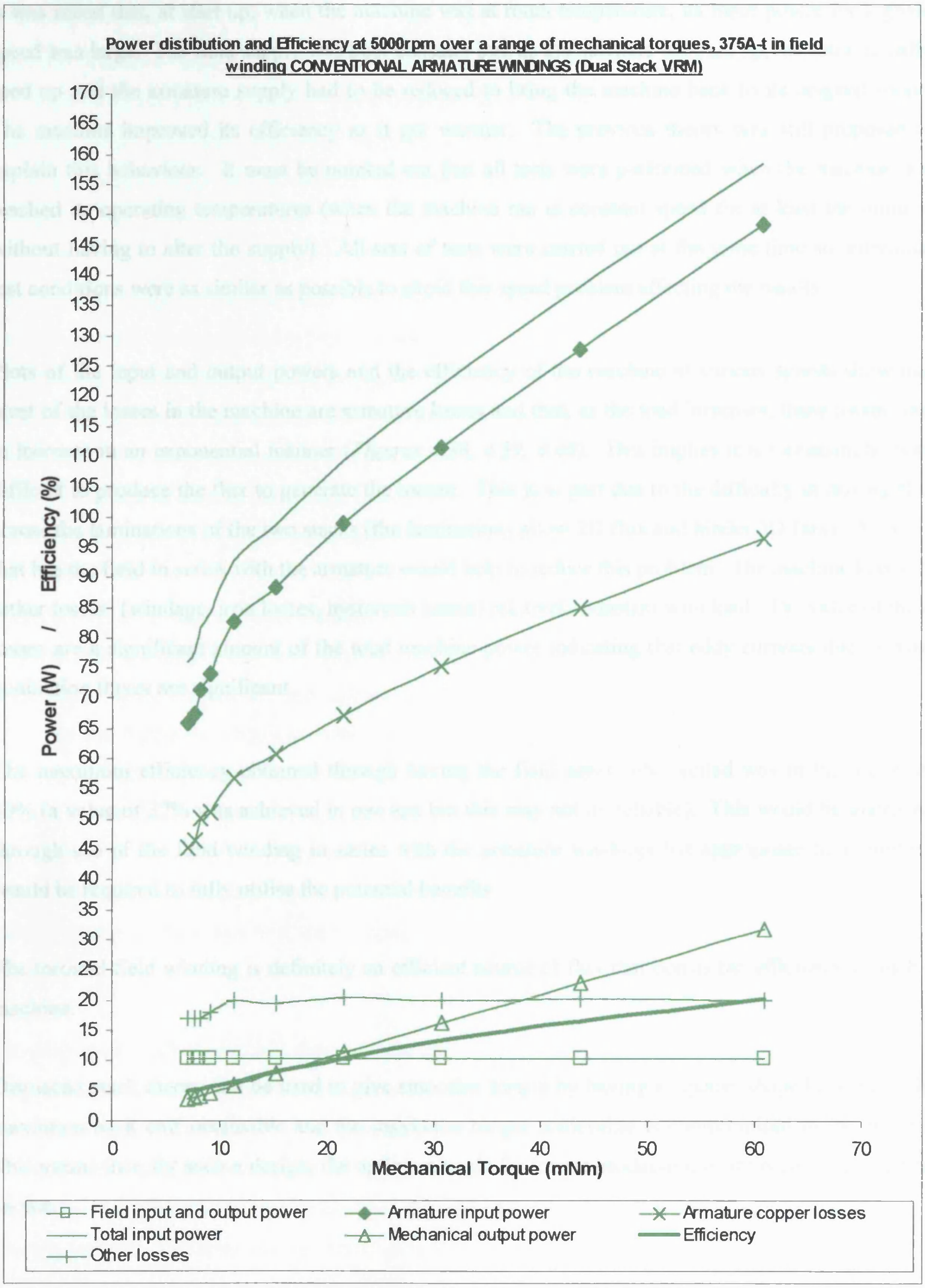


Figure 4.40 :- Input and output powers and efficiency against load for 5000rpm.

It was noted that, at start up, when the machine was at room temperature, its input power for a given speed was large. The field supply was kept constant but, as the machine warmed up, the rotor actually sped up and the armature supply had to be reduced to bring the machine back to its original speed. The machine improved its efficiency as it got warmer. The previous theory was still proposed to explain this behaviour. It must be pointed out that all tests were performed when the machine had reached its operating temperatures (when the machine ran at constant speed for at least ten minutes without having to alter the supply). All sets of tests were carried out at the same time so individual test conditions were as similar as possible to avoid this speed problem affecting the results.

Plots of the input and output powers and the efficiency of the machine at various speeds show that most of the losses in the machine are armature losses and that, as the load increases, these losses start to increase in an exponential manner (*Figures 4.38, 4.39, 4.40*). This implies it is increasingly more difficult to produce the flux to generate the torque. This is in part due to the difficulty in driving flux across the laminations of the two stacks (the laminations allow 2D flux and hinder 3D flux). A circuit that has the field in series with the armature would help to reduce this problem. The machine keeps its 'other losses' (windage, iron losses, hysteresis losses) relatively constant with load. The value of these losses are a significant amount of the total machine power indicating that eddy currents due to cross lamination fluxes are significant.

The maximum efficiency obtained through having the field separately excited was in the region of 30% (a value of 37% was achieved in one test but this may not be reliable). This would be improved through use of the field winding in series with the armature windings but appropriate turn numbers would be required to fully utilise the potential benefits.

The toroidal field winding is definitely an efficient source of flux that boosts the efficiency of such a machine.

Displaced stack theory can be used to give smoother torque by having a squarer shaped current. The maximum back emf obtainable and the maximum torque achievable is compromised in the process. This means that, for such a design, the optimum angle for torque production is not necessarily the best choice.

4.12 SUMMARY

The new machine construction has provided a valuable insight into alternative reluctance machine geometries and excitation schemes. The test results have demonstrated that the concept (of a switched reluctance variant of the homopolar inductor alternator topology) works and that control of the machine characteristics can be achieved via either the field current or the phase winding (armature) current.

The complex magnetic geometry has been reduced to a simple model for design purposes. However, test results have shown the model to overestimate the magnetic fluxes, indicating that eddy currents due to cross lamination fluxes may be significant. It is therefore concluded that this geometry would be very appropriate to be evaluated with powdered iron core construction. Powdered iron would allow the possibility of three dimensional flux paths with reduced iron and eddy current losses and the ability to shape a motor to benefit the complex flux paths. This machine has a topology that is ideally suited for an investigation using powdered iron technology.

One factor that partly explains efficiency limitations with this design is caused by the use of the toroidal field winding itself. The dc current in the field winding will give dc flux axially within the rotor, when considered as the only source of flux. This would not cause significant circumferential eddy currents in the steel sections with axial flux - as there is no significant change in flux magnitude to cause the eddy currents to oppose the change. But a sense coil around the field winding does show a ripple in the flux, and spikes of back emf where switching occurs in the armature coils of each phase. Thus there is a degree of eddy current losses in these motors. It is possible to reduce the circumferential eddy currents by use of powdered iron or by breaking the paths by insertion of e.g. plastic spacer so flux can still travel axially as desired but electrically insulates preventing eddy currents.

The displaced stack theory has shown that a current waveform can be improved to give a smoother torque due to the current having a squarer shape. However, this has also been shown to limit the torque capability of the motor. It is necessary to select a displacement angle that gives a higher back emf and hence torque for a slightly less square current. It was shown that optimum smooth torque was at 45 degrees mechanical displacement but 29 degrees gave a better torque output for a slightly less smooth torque. Different angles would give different machine characteristics. For this machine, smooth torque production was desired and was obtained.

The use of unipolar excitation with single coil per pole windings produces unipolar flux per stator pole. It gives a higher torque output than the first version with bipolar excitation with bifilar windings. This is due to all of the flux produced by the unipolar excitation with single coil per pole windings being available for torque production. In the case of bipolar excitation with bifilar windings model, some of the flux is used to prevent flux leaking into the unaligned stator poles and to push the rotor into the aligned position. It is speculated that this bipolar excitation with bifilar windings model was detrimental to axial flux from the field winding. A three dimensional finite element analysis would be best suited for accurately modelling this style of machine. It has been modelled with two dimensional analysis with a satisfactory level of success. Only the 3D model would reveal the truth as to what happens with the flux in the bifilar windings. The 3D model would also possibly provide enough data to model L(dI/dt) in the dc machine model of (4.4).

The bipolar excitation with bifilar windings version of the machine is better suited to a current source half bridge converter circuit to energise a series connection of armature and field windings. The best torque production is when unipolar excitation with single coil per pole windings is used but this requires an asymmetric half bridge converter circuit.

CHAPTER 5

SINGLE STACK VARIABLE RELUCTANCE MACHINE WITH AXIALLY ORIENTATED FLUX ASSISTANCE

5.1 INTRODUCTION

The dual stack variable reluctance machine is a complex design incorporating a three dimensional flux path. It incorporated two stacks separated by a toroidal field winding. It was realized that a simplification of the machine design would allow a smaller machine to be built. This simplification would produce a machine based upon the same homopolar inductor alternator principles [14-17, 20, 21] as the dual stack machine but with a reduction in the overall machine complexity by halving the number of stator stacks. There was still the novel method of providing additional dc flux through the use of a toroidal, axially oriented coil. A two phase 4/2 variable reluctance machine was constructed and tested in which the stator comprised a single short stack adjacent to an axially orientated coil arranged to give axial magnetic flux. This construction is the machining equivalent of half a homopolar inductor alternator.

The complexity is reduced by using one stack and replacing the other stack with a steel block to complete the magnetic circuit. An example sketch of the design is shown in *Figure 5.1*. It is possible to house the toroidal field winding within this steel block. The rotor is altered to accommodate this housing block for the toroidal field winding. It has a conventional two pole rotor (for a 4/2 design) at the stator stack end with the remainder of the rotor cylindrical at the field housing end (a boss design). This allows a novel addition of a magnetic slip ring to the machine. The magnetic slip ring forms part of the rotor (by the toroidal field winding end of the machine), giving a constant air gap around the rotor. By doing so there is a constant cylindrical air gap as part of the complete magnetic circuit but it allows minimization of overall machine axial length. The design is especially useful as the section housing the field forms part of the back-iron and the inner part of it surrounds the magnetic slip ring. The toroidal field winding is simply dropped into this housing during production and is kept in place by the stator windings on the only stack. This allows manufacturing benefits. To give extra space for the field winding, a different type of bearing is used in preference to the usual ball bearing. The needle roller bearing is just as effective in principle as the ball bearing but occupies a smaller radius, allowing more design flexibility in modelling the machine.

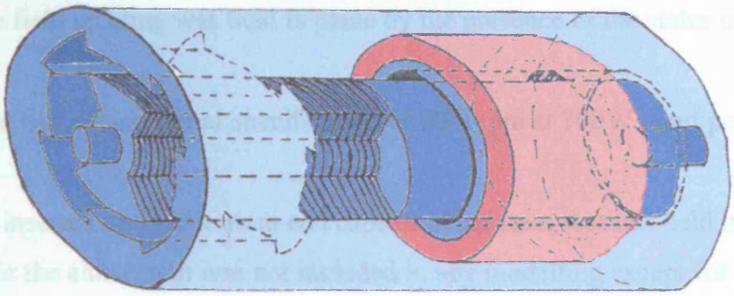


Figure 5.1 :- Sketch of single stack variable reluctance machine with interior view.

As with the previous design, an existing 4/2 switched reluctance design of 54mm stack length was used to make this new machine (the laminations were used in a lawnmower machine). The lamination dimensions were thus known and this allowed limitations to be imposed on the modelling process in terms of dimensional parameterization. As with the dual stack motor, the field winding was used to produce a unipolar axial magnetic flux and bifilar windings were used for the stator coils to direct the flux into or out of (as well as providing additional flux to) diametrically opposing stator poles pairs.

The machine design included cooling ducts due to the temperature problems encountered with the dual stack machine. The aluminium end caps that existed for the 4/2 laminations had enough axial length to allow safe inclusion of an internal fan allowing improved cooling potential. It was proposed that the fan would suck air from the motor rather than blowing air into it as this had proven in the past to be the wiser method of motor cooling.

5.2 MACHINE TOPOLOGY - DESIGN FOR A SINGLE STACK VARIABLE RELUCTANCE MACHINE

Self-bonding wire was used in this machine (Appendix E) to keep the wire coils in their wound shapes. The wire self-bonded at $120A/mm^2$, although heating in an oven, using hot air streams and painting on methylated spirits were possible. The use of varnish and glues to bind the windings together were unsuccessful due to the high packing factor in the toroidal field winding (0.7).

Self-bonding wire limited the packing factor obtainable because this type of wire had an additional layer outside the insulation which bonded together with similar wire when activated by heat. The packing factor obtained in the motor for the toroidal field winding was 0.52 compared to 0.7. This was still a high value but it imposed limitations on expectable packing factors. The self-bonding process did not appear to affect the wire resistance based on values before and after treatment.

Nylon66 was also used to hold the stator coils in place so they did not interfere with the rotor (Appendix E). The field winding was held in place by the presence of the stator coils.

The machine design had a basic initial specification of 8000rpm at 300W input power.

The bearings were inserted into aluminium end caps or aluminium sections held in place by end caps. Being non-magnetic the aluminium was not included in any modelling except for allowing space for inclusion of any parts.

5.3 WINDING CONFIGURATIONS

There are different possible winding configurations that could be used for this type of machine, which are based upon and explained further in terms of magnetic flux paths in Section 4.4 of Chapter 4. The winding possibilities are shown in *Figures 5.2* to *5.5.* Future diagrams may refer to the windings in sections 4.4.2 and 4.4.3 as 'conv' and 'bif' for unipolar excitation with single coil per phase and bipolar excitation with bifilar windings respectively only for the purpose of simplifying the text.

The toroidal field winding provides maximum packing factor and copper utilisation due to its shape. There are no end winding losses associated with the field winding. The energisation of the field winding was arbitrarily chosen to make the rotor stack a South Pole. The magnitude of the flux from the field coil linking each of the phase windings of the stator is modulated by the permeance of the magnetic path. As the rotor rotates, the coil flux linkage reaches a maximum when the rotor poles are aligned with the stator poles of that coil. An alternating emf is therefore produced in each of the stator coils. The stator coils on adjacent stator teeth have emfs exactly out of phase with each other. This is why the term armature can be used for these coils.

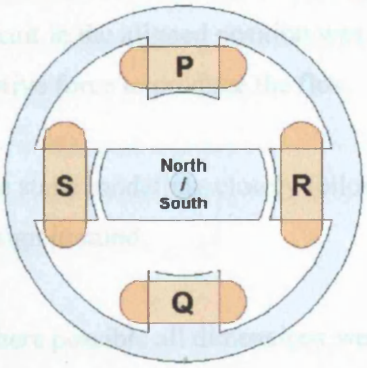


Figure 5.2 :- Coil labels PQRS. for single stack.

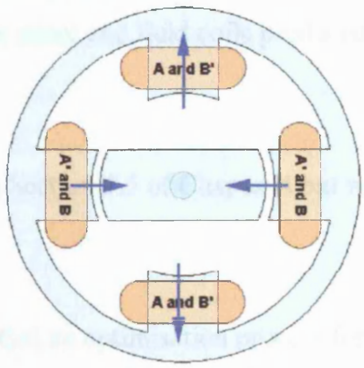


Figure 5.3 :- Bipolar excitation with bifilar windings (phase B energised shown).

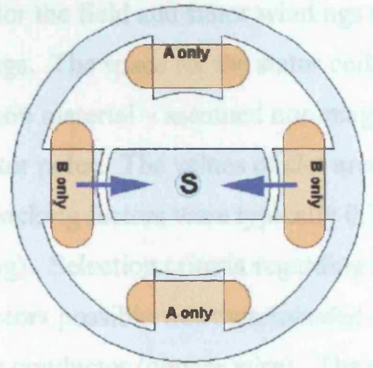


Figure 5.4 :- Unipolar excitation with single coil per pole ('conventional').

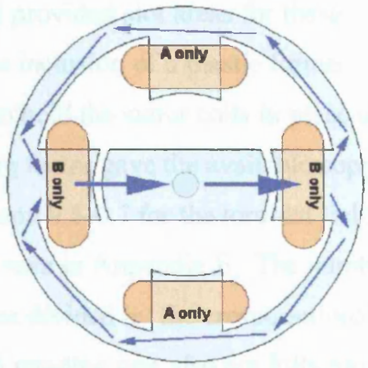


Figure 5.5 :- Switched Reluctance energisation – the field winding is unenergised – a comparison model only.

An innovative method of driving this machine involves series connection of all the phase windings such that the induced voltages are all in the same sense, as shown in *Figure 5.3*. Forcing a bipolar current to flow in opposition to this induced emf produces motoring torque.

5.4 STATIC MODELLING OF 3D MAGNETIC CIRCUIT

As with the dual stack motor a sketch of an initial design was made, incorporating the two pole rotor with cylindrical section forming part of it and the four pole stator lamination section. The field winding was set within the steel block that surrounds the cylindrical rotor section and which formed the completion of the magnetic circuit. The potential flux paths were sketched onto this design to illustrate how flux might travel within the motor for an aligned rotor position. It was assumed that, for ease of calculation, the magnetic fields were those that were intuitively drawn for the model and that no flux leakage would occur. The use of a 54mm stack length, 88.5mm diameter 4/2 lamination stack gave certain limitations of the size of the final machine. It was desirable to keep the final overall length of the machine the same.

The motor was modelled as a magnetic equivalent circuit such that a total reluctance for the magnetic circuit in the aligned position was calculated and the stator and field coils produced the magneto motive force to produce the flux.

The static modelling closely followed that shown in Section 4.5 of Chapter 4 but with a single stack design in mind.

Where possible all dimensions were variables such that an optimisation process for minimising machine total reluctance was possible. It was decided that, as the overall axial length of the machine was not to exceed 54mm to maintain overall size, the axial length of the 4/2 laminations could be a value less than 54mm which would be included into the model as a variable. The model also allowed space for the field and stator windings to be inserted. The model provided slot areas for these windings. The space for the stator coils was compromised by the inclusion of a plastic former (Nylon66 material – assumed non-magnetic – see Appendix E) to hold the stator coils in place around the stator poles. The values of slot area multiplied by the packing factor gave the available copper area (packing factors were typically 0.3-0.35 for the stator coils and 0.5-0.7 for the toroidal field winding). Selection criteria regarding the packing factor can be seen in Appendix E. The number of conductors possible was then rounded down value for copper area divided by the cross-sectional area for one conductor (copper wire). The space for the toroidal field winding was also not fully toroidal. The section of the steel housing (near the bearing) provided a magnetic link between the rotor and back-iron, carrying all the magnetic flux. It is known that flux does not travel well around a 90° corner and heavy saturation at that point is commonplace so it is replaced by two 45° corners to round-off this point. The toroidal winding is wound on a specially made former to allow for this design feature.

In addition to the air gap between the stator and rotor lamination there are two air gaps within the flux return path that pose interesting design issues (*Figures 5.6, 5.7*). The first is the radial air gap whose function is to provide a continuation of the flux paths between the cylindrical section of the rotor and the housing of the field. Its purpose is as a return path and not for torque production and it is therefore regarded as a parasitic air gap (*Figures 5.6, 5.7*). This air gap has to be very small (0.2mm, ideally less) to minimise its reluctance but mechanical clearance has to be retained.

There is also a gap between the end face of the rotor and the field housing. If this is too small a large axial force (end thrust) would be produced. This axial gap was chosen to be 10 times greater than the radial gap (i.e. 2mm) to minimise this problem.

There is a possible use of this end thrust effect. Ordinarily this effect is detrimental to the machine as bearings are not designed for axial forces. Some loads have an effect of pulling or pushing the rotor in an axial direction e.g. fan blades moving air. If the machine are connected correctly the end thrust produced by the machine could be used to reduce or cancel the axial forces caused by the load. It is noted that such a cancellation technique has to allow for the fact that, at start up, the axial load forces would be minimal. Any machine making use of the effect (not covered in this thesis) would have to be specially designed.

The resistance of the coils were predicted by knowing the average length per turn of wire (found from the model) and knowing the cross sectional area of one turn of wire (5.1). The coil resistance was proportional to the square of the number of turns (5.2).

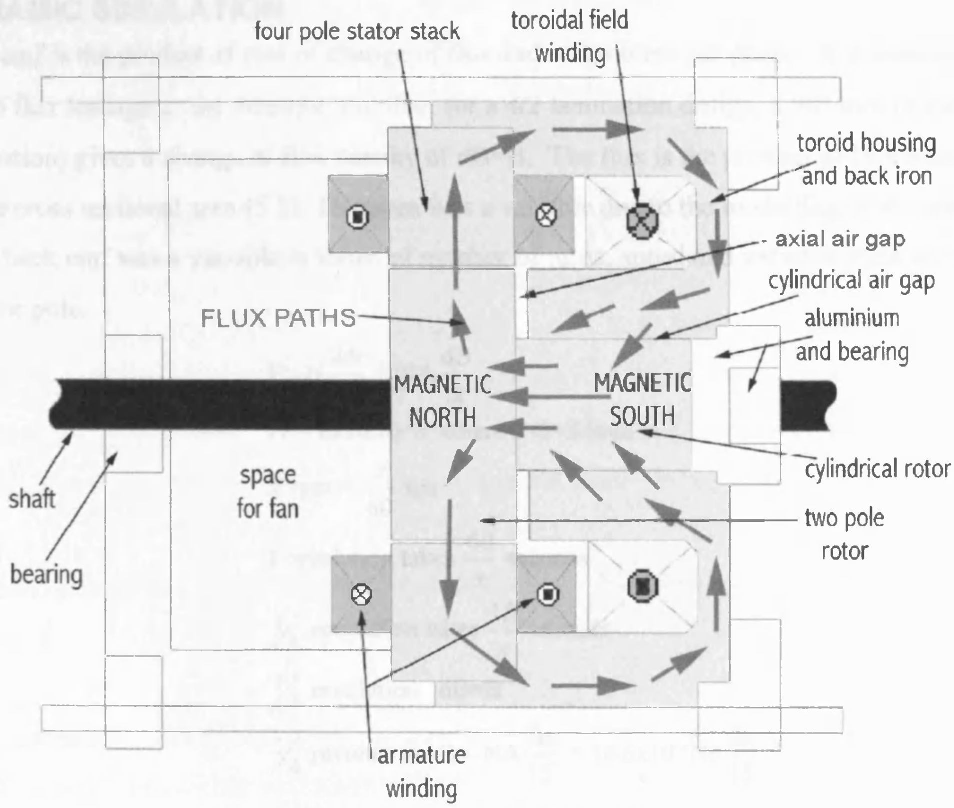


Figure 5.6 :- Flux paths in the proposed machine.

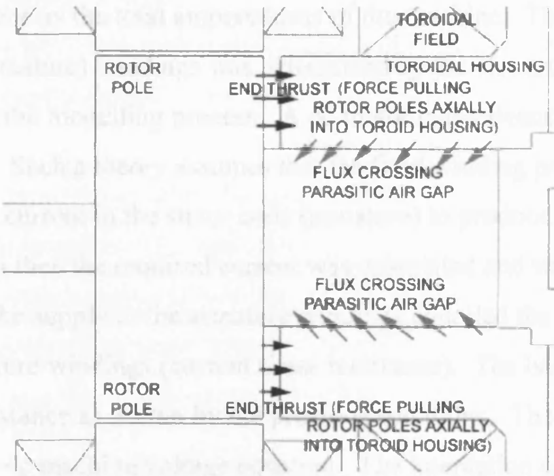


Figure 5.7 :- The parasitic air gap and the production of end thrust.

$$\text{Resistance, } R = \frac{\rho l}{A} \quad \text{where} \quad \begin{aligned} &\rho = 1.7\text{x}10^{-8}\,\Omega\text{m} \\ &l = N(\text{average length per turn}) \\ &A = \frac{\text{copper area}}{N} \\ &N = \text{number of turns} \end{aligned} \tag{5.1}$$

$$\text{hence} \quad R = kN^2 \quad \text{where} \quad k = \frac{\rho(\text{average length per turn})}{\text{copper area}} \tag{5.2}$$

5.5 DYNAMIC SIMULATION

The back emf is the product of rate of change of flux and stator turns per phase. It is assumed that there is no flux leakage in the machine and that, for a 4/2 lamination design, a 90° turn of the rotor (1/4 revolution) gives a change of flux density of dB=B. The flux is the product of flux density and stator pole cross sectional area (5.3). This area was a variable due to the modelling of the machine and hence the back emf was a variable in terms of number of turns, speed and variable cross sectional area of the stator pole.

$$
\begin{aligned}
&E = N\frac{d\Phi}{dt} = NA\frac{dB}{dt}\\
&A = 16.6\text{x}10^{-3}b \quad \text{where } 0<b<54\text{mm}\\
&x \text{ rpm} = \frac{x}{60} \text{ rps}\\
&1 \text{ revolution takes } \frac{60}{x} \text{ seconds}\\
&\tfrac{1}{4} \text{ revolution takes } \frac{15}{x} \text{ seconds}\\
&\tfrac{1}{4} \text{ revolution } \quad dB=B\\
&\tfrac{1}{4} \text{ revolution } \quad E = NA\frac{Bx}{15} = 16.6\text{x}10^{-3}Nb\frac{Bx}{15}
\end{aligned}
\qquad (5.3)
$$

The model predicted the total magnetic equivalent reluctance for the machine. This value multiplied by magnetic flux is the same as the total ampere turns of the machine. The number of available turns for the field and stator (armature) windings was determined by the slot areas, packing factors and wire diameters selected during the modelling process. A dc motor equivalence was used to predict the motor performance (5.4). Such a theory assumes that the field winding produced all the flux and that all this flux interacts with current in the stator coils (armature) to produce torque. If the number of armature turns was known then the required current was calculated and was then used in the dc motor equivalence equations. The supply to the armature windings equalled the sum of the back emf and the volt drop across the armature windings (current times resistance). The back emf was predicted as was the armature winding resistance as shown by the previous equations. The power calculation (5.4) was a simple expansion of the dc machine voltage equation. The interaction of the current and back emf gave rise to torque production (5.5). It was then possible to predict the performance of the motor for a given speed in radians per second. L(dI/dt) was ignored as the current, I, was assumed to be constant, for ease of calculation.

$$
\begin{aligned}
&V_a = E_a + I_a R_a\\
&V_a I_a = E_a I_a + (I_a)^2 R_a
\end{aligned}
\qquad (5.4)
$$

$$E_a I_a = T\,\omega \quad \text{where} \quad \begin{array}{l} E_a = \text{back emf (V)} \\ I_a = \text{armature coil current (A)} \\ T = \text{torque (Nm)} \\ \omega = \text{speed in rads}^{-1} \end{array} \tag{5.5}$$

The magnetic flux was limited by the cross sectional area of sections of the motor. The smallest area was regarded as the constraining value and, for a required magnetic flux density, the mmf of the motor was calculated. The mmf produced by the stator coils and the field coil was to match this value and, from the dc motor equations, the performance of the motor was dynamically modelled for a given rotor speed and supply voltage. The choice of wire size and packing factor was varied and the model included error detection to prevent selection of impossible dimensions.

Temperature calculations for the windings using Stefan's Law is calculated using the same theory as in Appendix E.

It should be stressed that this model did not take into consideration that the 4/2 switched reluctance section of the machine had laminations and that these laminations were highly likely to affect the modelling of the machine. It was suggested that the mmf needed to drive flux across the laminations would be greater than expected. The flux paths were three dimensional but the laminations were designed for two dimensional fluxes in the same plane as the laminations. Also, effects such as iron losses and eddy currents were assumed to not exist.

The field winding was held in place within the steel housing unit but there was some concern that the winding, being toroidal in shape, could rotate within the housing and have its insulation scraped off by the cooling ducts. The solution was to reduce the number of turns in the field and place insulation taping around the toroidal windings preventing any possible rough parts from breaking the insulation. The disadvantage of this was lost copper space hence lost ampere-turns for the field winding.

Table 5.1 gives the choice of turns for the armature and field based on the design model used. Pictures of the final design and its dimensions can be seen in Appendix G.

TABLE 5.1 - DIMENSIONS WITHIN MACHINE

	Field	per Phase	Total
Turns	1600	556	-
Wire Diameter	0.315mm	0.315mm	-
Resistance	36.25ohm	12.6ohm	-
mmf	-	-	1502
Packing Factor, PF	0.517	0.3	-
Power Wanted	-	-	300W
Speed Wanted	-	-	8000rpm

5.6 BACK EMF TEST RESULTS

A variable speed dc motor was used to drive the rotor of the Single Stack Variable Reluctance Machine (SSVRM). A discussion of the results and the effects of a bearing imbalance are discussed in Appendix G.

5.7 MOTOR TESTING

The machine required optimisation in order to be as efficient as possible for any required specification, usually for a given load or speed. This involved the optimisation of the phase firing timing sequence to minimise the total input power. A prototyping circuit was developed to allow optimisation of the sequences. This is described in Chapter 3 and Appendices B and C.

At no load the machine achieved over 19000rpm but no test data was recorded as it was only done to ascertain basic speed limitations. This was actually faster than the speed rating of the ball bearings (rated to 18,000rpm). It was noted that at such speeds the motor was exceptionally noisy.

The optimisation of the timing sequence for phase firing at a fixed speed allowed testing the machine at varying loads (see Chapter 3 and Appendix G). The use of a hysteresis brake allowed small loads to be applied to the machine (by applying a dc current to the brake the load was a function of rotor speed and dc current value) (Appendix B).

5.7.1 BIPOLAR EXCITATION WITH BIFILAR WINDINGS TEST RESULTS

This winding type was assumed to provide the best method of transferring current between the phase windings as one phase turns off and the next on with the simplest power electronic circuit (half-bridge converter). The turn numbers are summarised in Table 5.2.

TABLE 5.2 – TURN NUMBERS WITHIN MACHINE

Field	Armature (2 phases)
1400 turns	4x139 = 556 turns per phase
60.7 ohm	14 ohm per phase

Load tests were carried out at speeds of 5000rpm and 10,000rpm. The machine was initially found, for the initial given windings, to be most efficient at no load when in the High Speed Mode.

5.7.2 NO LOAD TESTING AT 5000RPM IN HIGH SPEED MODE

5.7.2.1 OPTIMISATION OF THE TOROIDAL FIELD WINDING

As can be seen in *Figure 5.8* there was a minimum input power. The value chosen as the optimum was 6.6 volts for the field. The current equivalent was used as the optimised value as the ampere-turn value it gave was the magnetic circuit value required for constant flux production.

5.7.2.2 OPTIMISE VAL7 (VALUE DETERMINING HOW ADVANCED THE PHASE FIRING STARTS AT)

This value represented the delay after opto state change when the phase was energised and was such that it altered how far advanced the start of phase firing before the next opto state change was. The optimum value for minimum input power was selected to be 90 (*Figure 5.9*). This variable clearly had a great effect on minimising input power for this machine.

5.7.2.3 OPTIMISE VAL6 (VALUE DETERMINING DELAY BETWEEN FIRING EACH PHASE)

This variable determined the size of time delay between energising each phase. It was apparent that, for this machine, the value selected for minimum input power was near 68 (*Figure 5.10*).

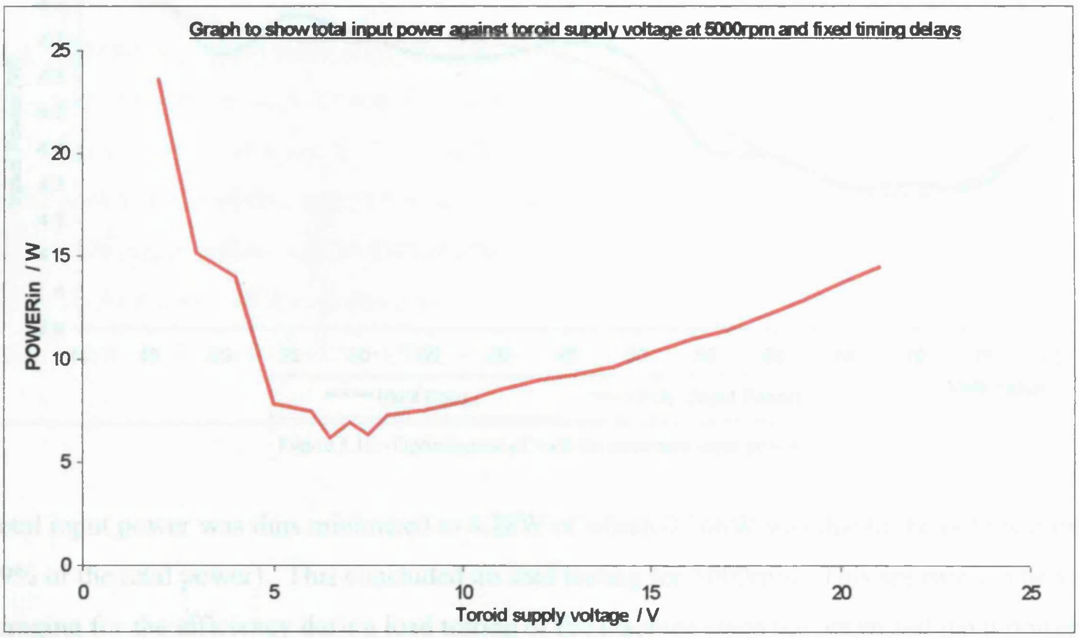


Figure 5.8 :- Optimisation of field supply for minimum input power

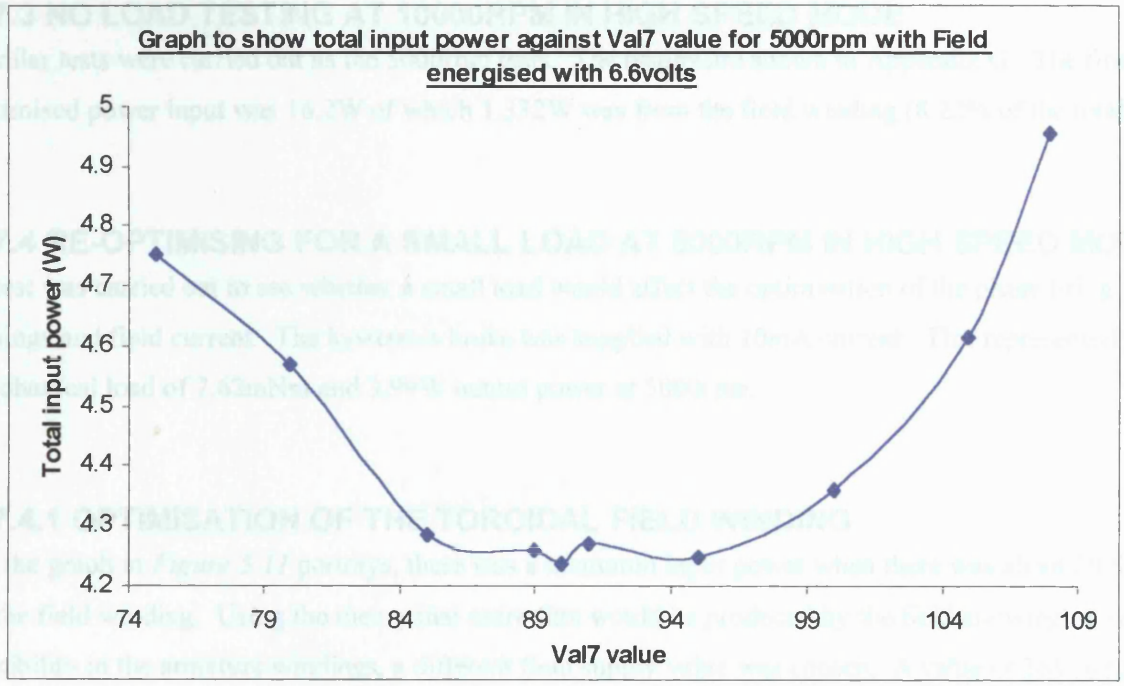


Figure 5.9 :- Optimisation of Val7 for minimum input power.

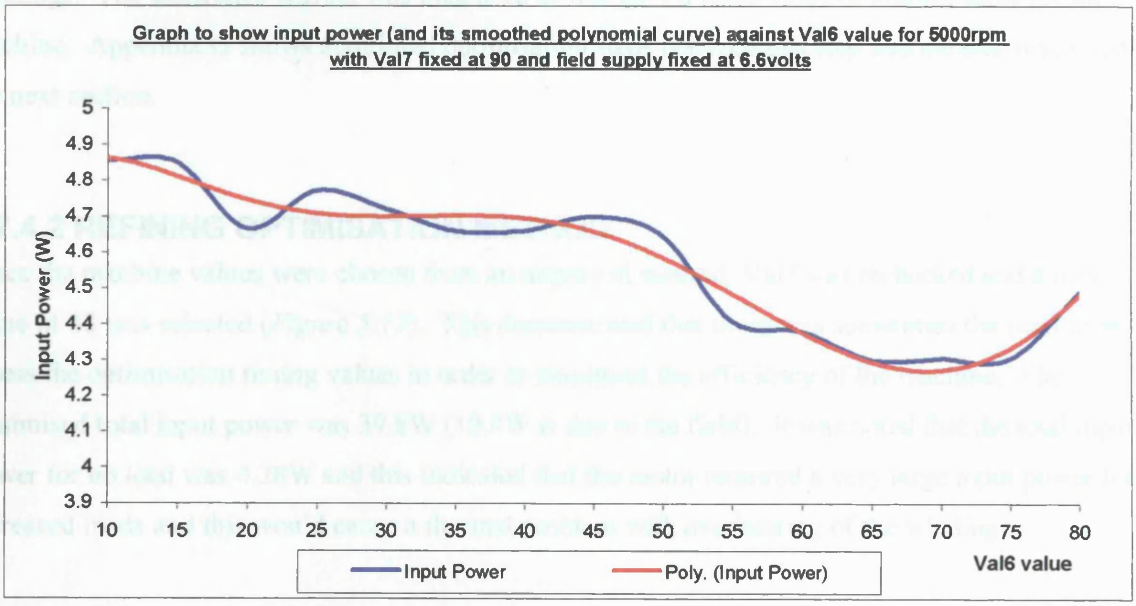


Figure 5.10:- Optimisation of Val6 for minimum input power.

The total input power was thus minimised to 4.28W of which 0.766W was due to the field winding (17.89% of the total power). This concluded no load testing for 5000rpm. This appeared to be very encouraging for the efficiency during load testing of the machine since the optimised input power was very small in value.

5.7.3 NO LOAD TESTING AT 10000RPM IN HIGH SPEED MODE

Similar tests were carried out as the 5000rpm tests. The results are shown in Appendix G. The final optimised power input was 16.2W of which 1.332W was from the field winding (8.22% of the total).

5.7.4 RE-OPTIMISING FOR A SMALL LOAD AT 5000RPM IN HIGH SPEED MODE

A test was carried out to see whether a small load would affect the optimisation of the phase firing timings and field current. The hysteresis brake was supplied with 10mA current. This represented a mechanical load of 7.62mNm and 3.99W output power at 5000rpm.

5.7.4.1 OPTIMISATION OF THE TOROIDAL FIELD WINDING

As the graph in *Figure 5.11* portrays, there was a minimum input power when there was about 20.5V in the field winding. Using the theory that more flux would be produced by the field allowing more flexibility in the armature windings, a different field supply value was chosen. A value of 26V across the field was selected as it allowed a small reduction in the input power required for the armature windings. The associated current was then used as this gave a fixed value of ampere turns for the machine. Appendix G shows additional optimisation steps between this step and the step discussed in the next section.

5.7.4.2 REFINING OPTIMISATION METHOD

Since the machine values were chosen from an unproved method, Val7 was rechecked and a new value of 88 was selected (*Figure 5.12*). This demonstrated that there was sometimes the need to re-assess the optimisation timing values in order to maximise the efficiency of the machine. The minimised total input power was 39.8W (10.4W is due to the field). It was noted that the total input power for no load was 4.28W and this indicated that the motor required a very large input power for increased loads and this would cause a thermal problem with overheating of the windings.

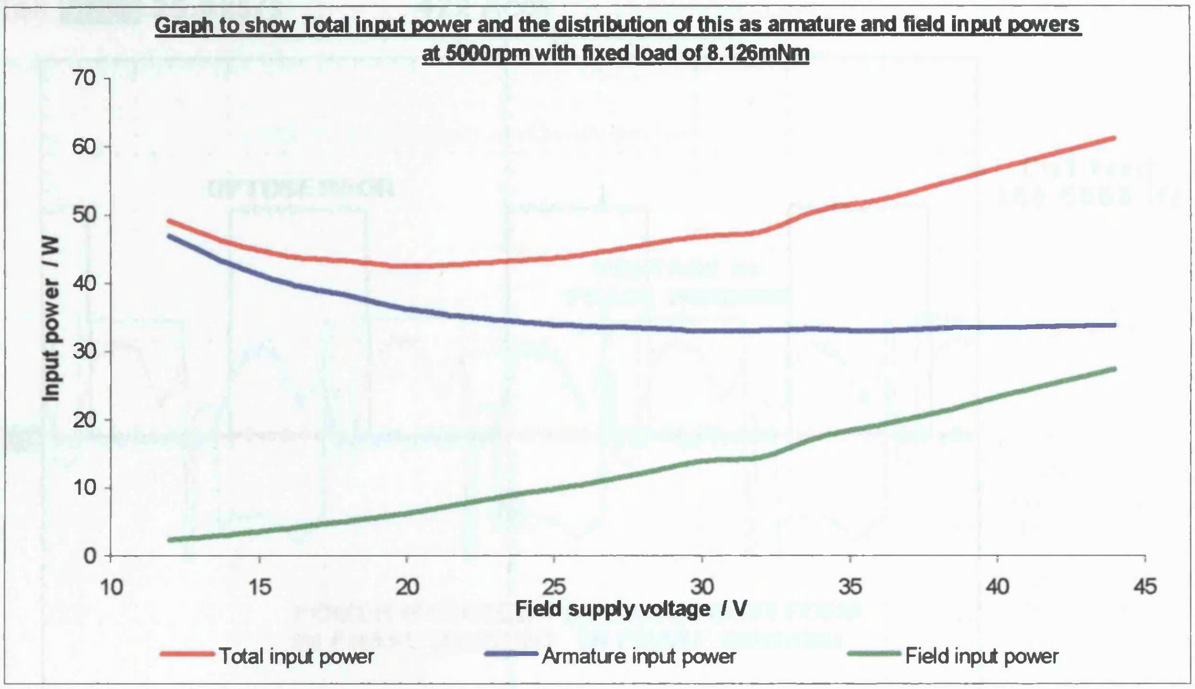


Figure 5.11 :- Optimisation of field supply for minimum input power.

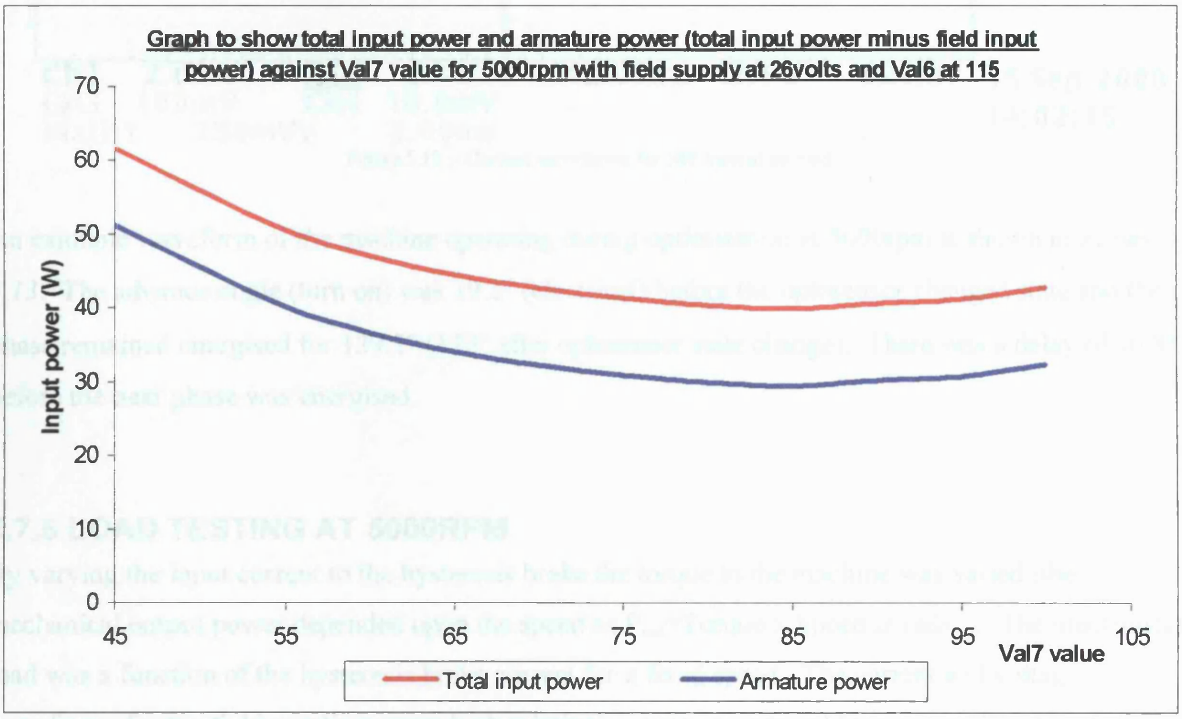


Figure 5.12 :- Re-optimisation of Val7 for minimum input power.

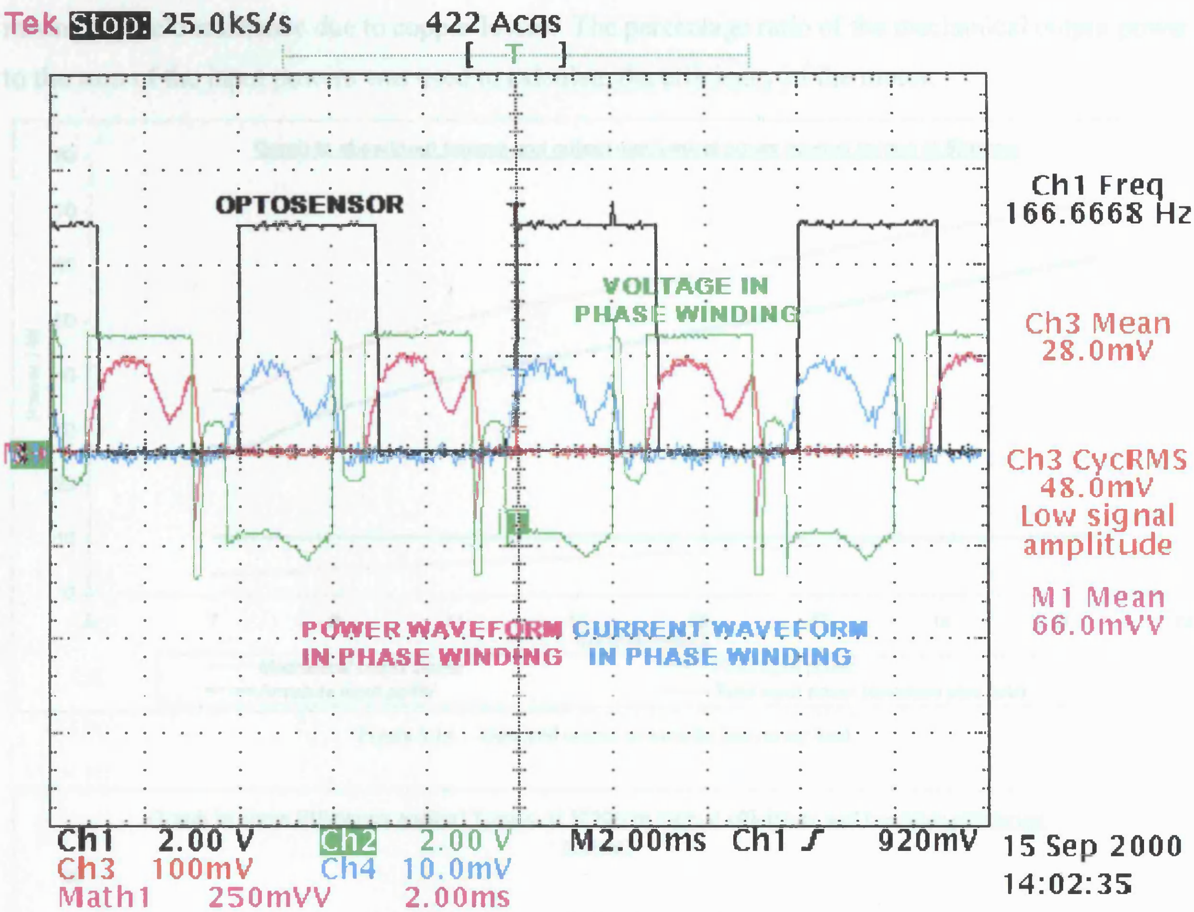


Figure 5.13 :- Current waveforms for 5000rpm at no load.

An example waveform of the machine operating during optimisation at 5000rpm is shown in *Figure 5.13*. The advance angle (turn on) was 19.2° (electrical) before the optosensor changed state and the phase remained energised for 139.2° (120° after optosensor state change). There was a delay of 40.8° before the next phase was energised.

5.7.5 LOAD TESTING AT 5000RPM

By varying the input current to the hysteresis brake the torque to the machine was varied (the mechanical output power depended upon the speed as P_{out}=Torque x Speed in rads^{-1}). The mechanical load was a function of the hysteresis brake current for a fixed speed. The current and voltage waveforms for the field windings were both relatively constant and could be assumed as dc values.

As the load was increased, for a fixed current supply to the field winding and optimised armature phase firing timings, the armature input power supply was adjusted until the speed was at 5000rpm. The current and voltage values for the field and armature were noted as well as the input power to the armature and associated power electronic components (via a power meter). The voltage supply to the field was adjusted to keep the field current constant to give fixed ampere turns due to temperature drift raising the field resistance due to copper losses. The percentage ratio of the mechanical output power to the sum of the input powers was used to calculate the efficiency of the motor.

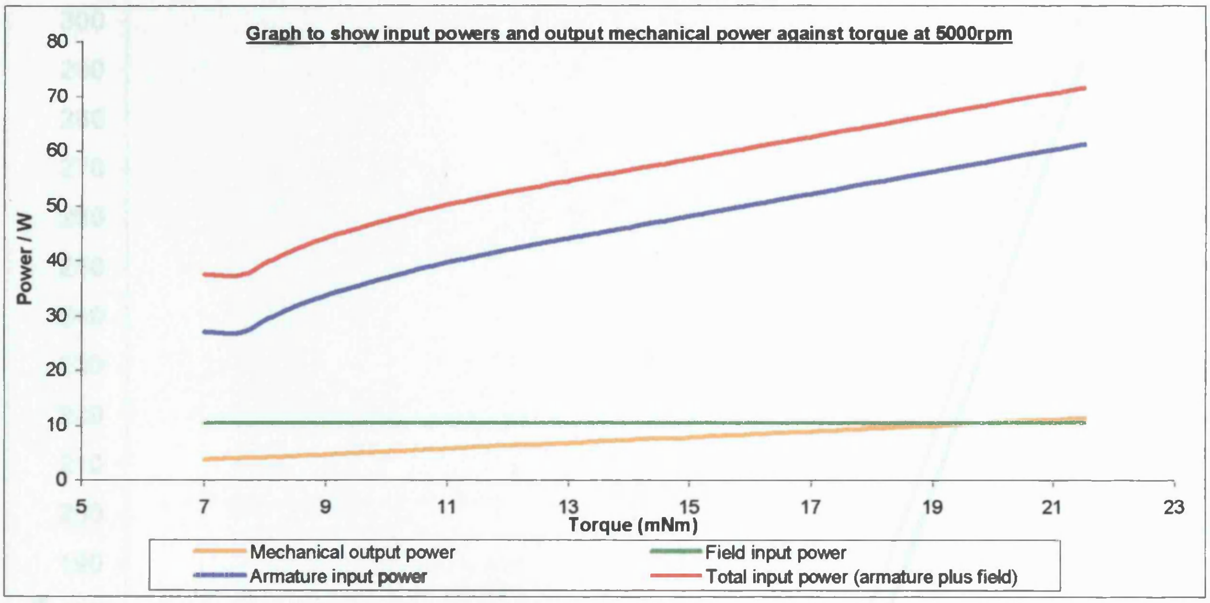


Figure 5.14 :- Input and output powers for increasing load.

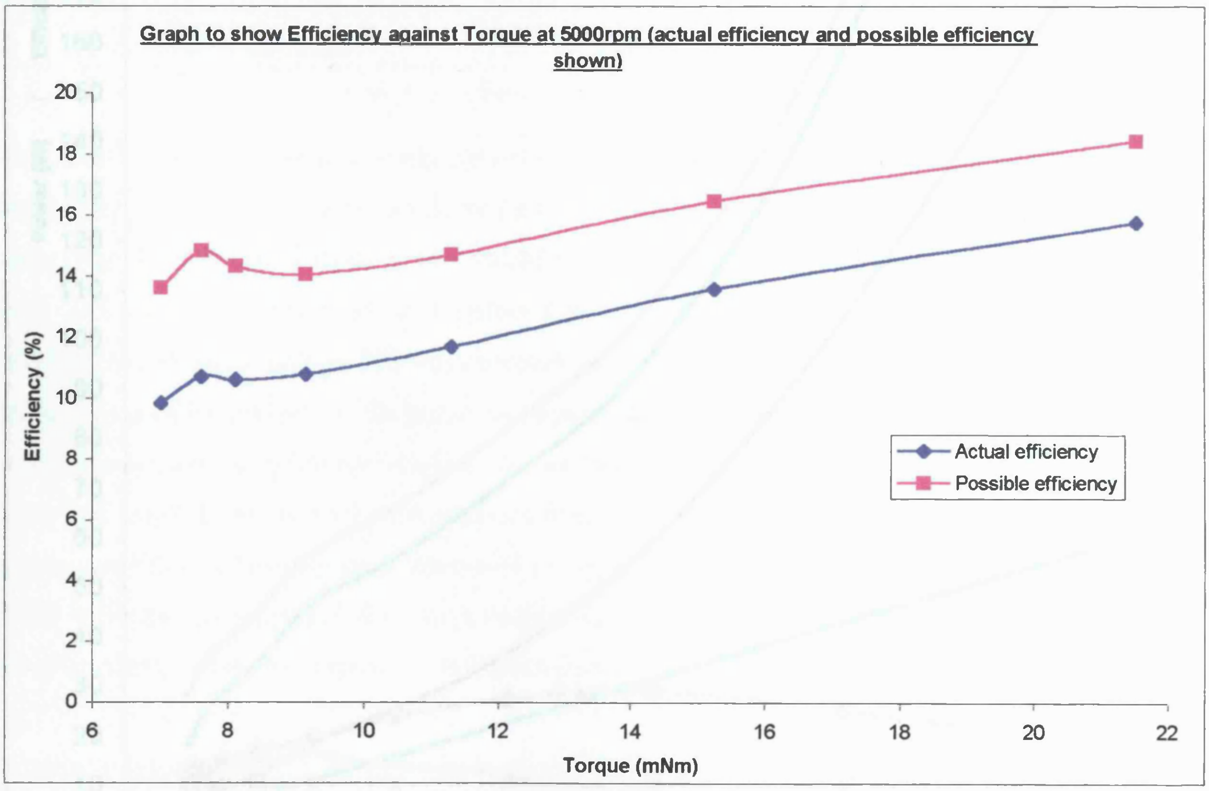


Figure 5.15 :- Efficiency with increasing load.

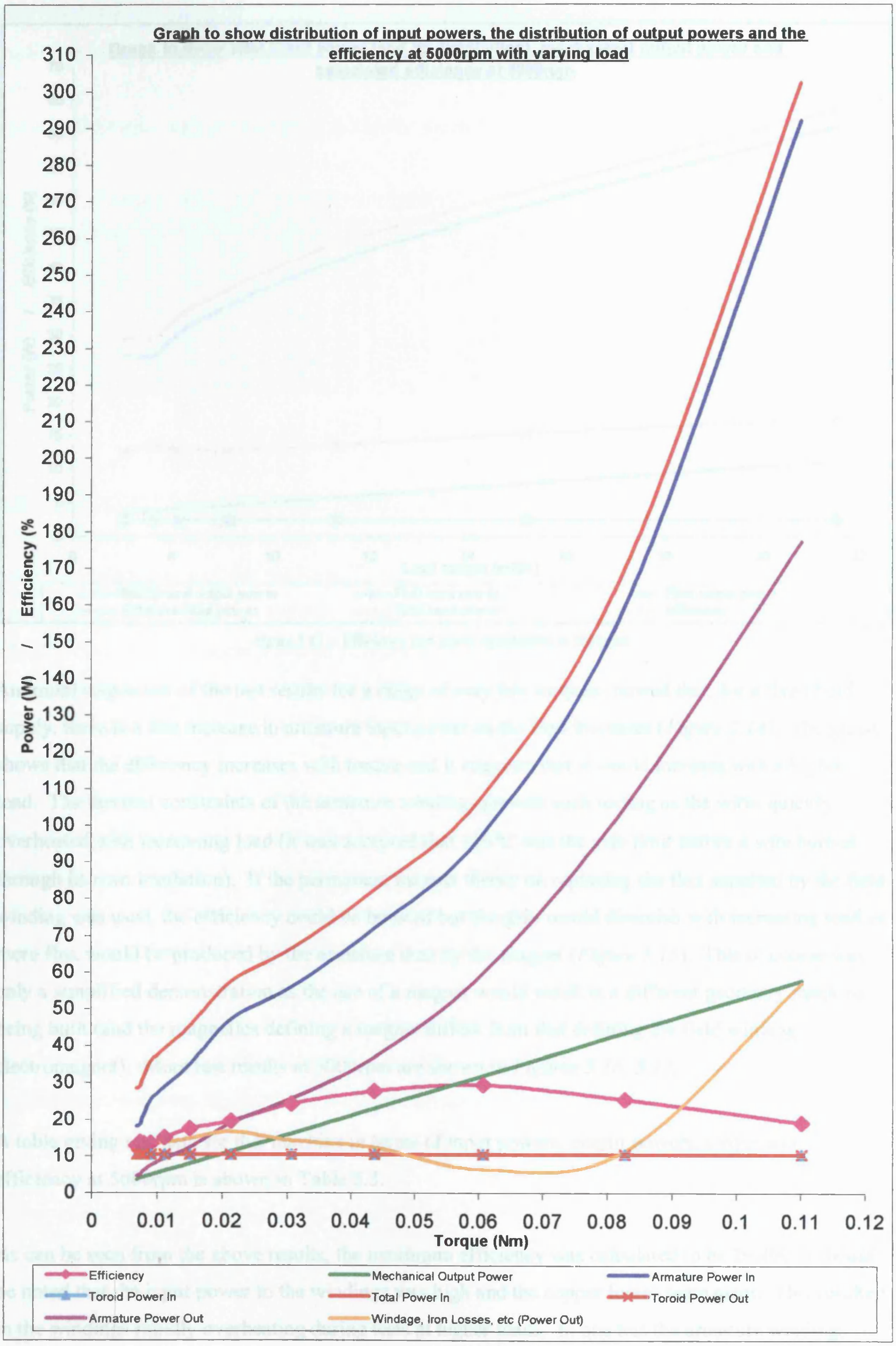


Figure 5.16 :- Efficiency and power distribution at 5000rpm.

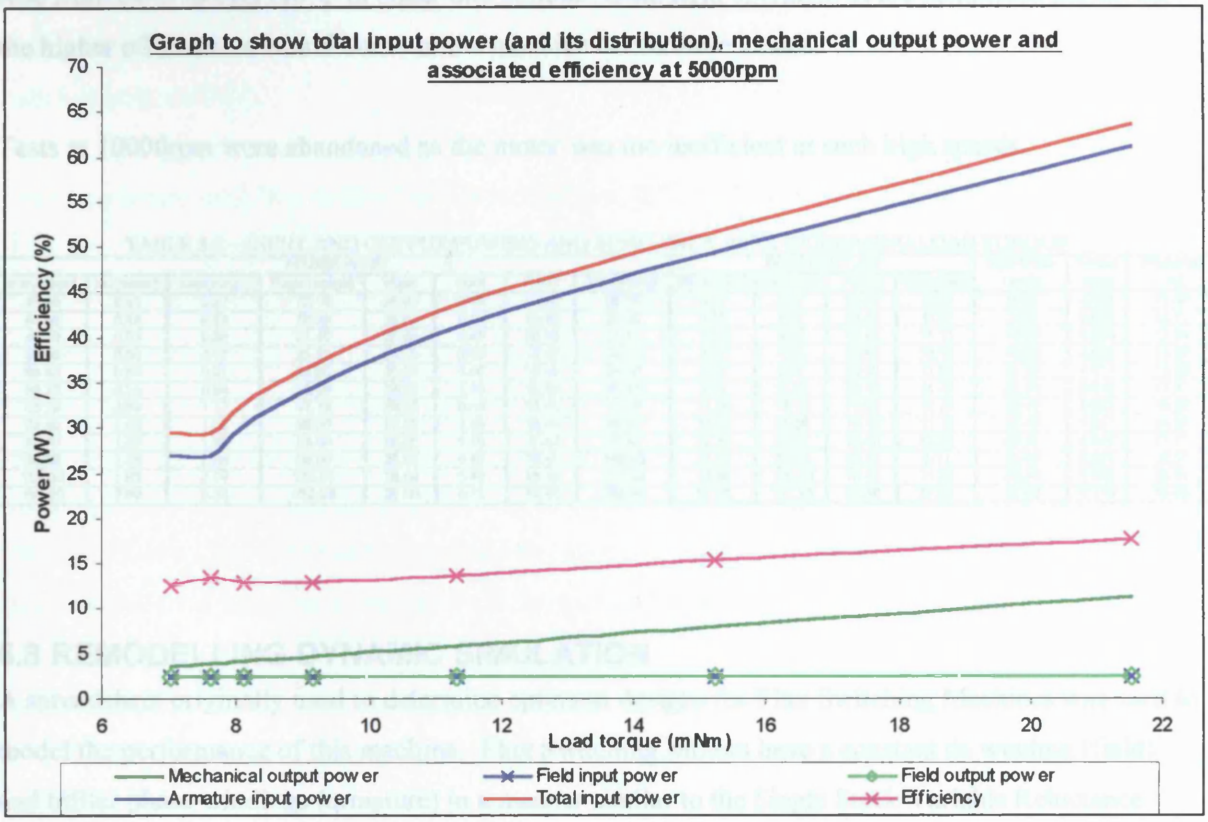


Figure 5.17 :- Efficiency and power distribution at 5000rpm.

An initial inspection of the test results for a range of very low torques showed that, for a fixed field supply, there is a fast increase in armature input power as the load increases (*Figure 5.14*). The graph shows that the efficiency increases with torque and it suggests that it would increase with a higher load. The thermal constraints of the armature windings prevent such testing as the wires quickly overheated with increasing load (it was accepted that 110°C was the safe limit before a wire burned through its own insulation). If the permanent magnet theory on replacing the flux supplied by the field winding was used, the efficiency could be boosted but the gain would diminish with increasing load as more flux would be produced by the armature than by the magnet (*Figure 5.15*). This of course was only a simplified demonstration as the use of a magnet would result in a different geometry machine being built (and the magnetics defining a magnet differs from that defining the field winding electromagnet). More test results at 5000rpm are shown in *Figures 5.16, 5.17*.

A table giving test data for this machine in terms of input powers, output powers, torque and efficiency at 5000rpm is shown in Table 5.3.

As can be seen from the above results, the maximum efficiency was calculated to be 29.4%. It should be noted that the input power to the windings was high and the copper losses were great. This resulted in the windings rapidly overheating during tests at higher loads. In one test the armature windings rose from 24°C to over110°C in under one minute. Long term operation at the conditions producing the higher efficiencies would burn out the machine hence these values.

Tests at 10000rpm were abandoned as the motor was too inefficient at such high speeds.

TABLE 5.3 – INPUT AND OUTPUT POWERS AND EFFICIENCY WITH INCREASING LOAD TORQUE.

POWER IN (W)								POWER OUT (W)				Hyst brake	Torque	Efficiency
Vdc(meter)	Idc(meter)	Irms(scope)	Parm (scope)	Vfield	Ifield	Pfield	Power total	Mechanical	Armature	Toroid	Remainder	I (mA)	(Nm)	(%)
46.40	0.46	0.57	18.10	26.00	0.40	10.40	28.50	3.67	4.60	10.40	9.84	0.00	0.007	12.86
46.60	0.48	0.59	18.60	26.00	0.40	10.40	29.00	3.98	4.92	10.40	9.70	5.00	0.008	13.72
48.50	0.53	0.66	21.40	26.00	0.40	10.40	31.80	4.24	6.05	10.40	11.11	10.00	0.008	13.34
52.00	0.58	0.72	25.30	26.00	0.40	10.40	35.70	4.82	7.36	10.40	13.12	15.00	0.009	13.49
55.40	0.65	0.81	28.90	26.00	0.40	10.40	39.30	5.92	9.31	10.40	13.67	20.00	0.011	15.06
58.20	0.75	0.94	35.00	26.00	0.40	10.40	45.40	7.96	12.31	10.40	14.73	25.00	0.015	17.53
66.70	0.88	1.17	47.10	26.00	0.40	10.40	57.50	11.26	19.10	10.40	16.74	30.00	0.022	19.58
72.40	1.02	1.35	56.00	26.00	0.40	10.40	66.40	16.13	25.68	10.40	14.19	35.00	0.031	24.29
82.70	1.20	1.62	72.00	26.00	0.40	10.40	82.40	22.88	36.87	10.40	12.25	40.00	0.044	27.77
96.60	1.48	2.06	98.00	26.00	0.40	10.40	108.40	31.83	59.85	10.40	6.31	45.00	0.061	29.37
122.30	1.93	2.74	160.00	26.00	0.40	10.40	170.40	43.35	105.33	10.40	11.32	50.00	0.083	25.44
161.20	2.60	3.55	**292.00**	26.00	0.40	10.40	302.40	57.70	177.26	10.40	57.03	55.00	0.110	19.08

5.8 REMODELLING DYNAMIC SIMULATION

A spreadsheet originally used to determine optimum designs for Flux Switching Machines was used to model the performance of this machine. Flux Switching Motors have a constant dc winding (field) and bifilar phase windings (armature) in a manner similar to the Single Stack Variable Reluctance Machine (except the physical placement and geometries differed). The spreadsheet was adapted so that test results for the Variable Reluctance Machine were replicated within the spreadsheet with a level of acceptable accuracy. By doing so the spreadsheet was used to remodel the machine for an improved performance.

Data required were current values in the field, back-emf values and magnetic flux values all with respect to mechanical angle of the rotor from aligned position. These were calculated from back-emf testing at a known field current which was not large enough to cause magnetic saturation within the machine. The current chosen was 400mA in the field.

The model had to predict how hot the windings would become during operation due to an overheating problem in the original winding selection. The field and armature windings would heat up as the current flowing through them increased. This was proportional to the power in the coils. The power per degree Celsius rise above room temperature (25°C), Υ ,was calculated from test results (where T was the measured temperature in °C) (5.6).

$$\Upsilon = \frac{W}{T-25}$$
$$T = \frac{W}{\Upsilon} + 25 \qquad (5.6)$$

The test result copper loss powers from the field and armature windings were used for calculating the Υ values for each (at 0.0152Nm torque the temperatures were 58.3°C and 49.95°C for the field and armature respectively).

The spreadsheet used data from a test run at 5000rpm, 0.0152Nm torque and from back-emf testing at 1071rpm. The spreadsheet originally predicted the performance of the machine assuming that there was no delay between switching between phases. The optimised machine incorporated a delay between energising each phase which had to be included into the model. The spreadsheet also had to be altered so that the dc field supply can be kept at a constant supply current (user defined). The temperature of the windings had to be calculated using the Υ values from actual test results. The model was also altered such that it calculated how many turns can physically be inserted within the machine. The spreadsheet had to determine the maximum number of turns possible for both the armature and field based upon available slot area and wire diameters.

This model calculated current density, resistance, power dissipation, torque, efficiency. It showed if there were fundamental errors such as overheating wires, too high a current density, and too many turns to fit in the machine.

The spreadsheet model was changed so that it predicted the current waveform for the same test conditions of armature and field supply as that of the actual test waveform for those conditions. It was then assumed that the model would give a more accurate prediction of the machine performance for a variety of speeds, loads, supply to the windings, turn off and turn on angles, and winding turn diameters and numbers. The spreadsheet was such that calculated values are fed back into the equation so that the answers were optimised (an iteration process).

Remodelling to give a better design involved altering the turns ratios and wire diameter and then altering the amount of current in the field winding. Altering the field current had the effect of having the back-emf values automatically recalculated and fed back into the spreadsheet. The supply voltage to the armature windings was kept at the same value as in the original test result that was being replicated. The above variables were changed until the original torque was obtained again. The net result was identical speed, supply voltage and torque but the efficiency varied. The aim was to maximise the efficiency then to note the effect on efficiency as the torque was increased (by increasing the supply voltage only – nothing else was altered).

A table of possible values is shown in Table 5.4 whereby the original torque was reproduced with different turn ratios. The possible torque at a higher armature supply was calculated.

By using the table, a prediction of the correct number of turns for the field and armature windings to give the best efficiency possible at 5000rpm was made. The machine was then rewound and retested. The optimised windings are included in Table 5.5 containing data for the machine.

TABLE 5.4 – PREDICTION OF MACHINE PERFORMANCE WITH NEW TURN NUMBERS.

				Field						Armature						Iron loss	Field	Armature	Iron loss factor		
Effic	Ind factor	Angle	Angle	turns (/2)	boost current	current	A-t	temp	J	turns(/2)	current	A-t	temp	J	total A-t		losses	losses	calculated	supply	torque
17.5	2.07	-22	60	700	1	0.3893	545.02	58.3	4.954	278	0.8629	479.8	49.95	11.06	1024.82	15.9	10.23	11.49	66.64	58.2	0.0152
8.7	2.07	-29	60	700	1.45	0.5644	790.16	106.713	7.1845	278	1.2	840.2	108.135	18.39	1630.36	39.7	25.11	38.25	80.25	100	0.0187
20.9	2.07	-22	60	675	0.875	0.3532	476.82	50.75	4.496	312	0.7308	456	44.7	9.373	932.82	13.2	7.914	9.076	62.345	58.2	0.0152
9.1	2.07	-30	60	675	1.43	0.5772	779.22	108.5	7.34	312	1	848	107.6	17.39	1625.22	41.7	25.39	38.46	79.7	118	0.0203
24.4	2.07	-22	60	650	0.76	0.3186	414.18	44.744	4.05636	346	0.6366	440.6	41.39	8.166	864.78	11	6.068	7.547	58.1	58.2	0.0152
9.6	2.07	-30	60	650	1.41	0.5911	768.43	108.7	7.52	346	0.9	887.6	107.6	16.45	1656.03	43.1	25.7	38.08	79.14	132	0.0217
28.1	2.07	22	60	625	0.66	0.2834	364.26	39.74	3.607	381	0.6674	432.4	38.22	7.278	786.66	9.2	4.632	6.6496	63.73	68.2	0.0162
10	2.07	-30	60	625	1.39	0.606	757.5	109.89	7.714	380	0.8	937.7	109.48	15.82	1695.2	44.8	26.09	38.91	78.57	150	0.0233
31.8	2.07	-22	60	600	0.55	0.2498	299.76	35.84	3.179	415	0.52007	431.7	37.95	6.6708	731.46	7.8	3.33	5.97	49.42	58.2	0.0152
10.6	2.07	-30	60	600	1.36	0.6176	741.12	109.57	7.8616	414	0.7	979	109.59	15.16	1720.12	45.7	25.99	38.96	77.7	167	0.025
35.2	2.07	-22	60	575	0.465	0.2203	253.345	33	2.804	449	0.483	434.2	37.07	6.202	687.545	6.6	2.45	5.56	45.449	58.2	0.0152
11.1	2.07	-30	60	575	1.33	0.6302	724.73	109.34	8.022	448	0.7	1017.4	109.313	14.56	1742.13	46.2	25.92	38.8	76.86	184	0.0266
38.5	2.07	-22	60	560	0.385	0.1907	209.77	30.6	2.428	484	0.458	443.4	36.664	5.875	653.17	5.6	1.75	5.37	41.365	58.2	0.0152
11.7	2.07	-30	60	550	1.3	0.644	708.4	109.213	8.198	484	0.6	1060.2	109.885	14.04	1768.6	46.9	25.88	39.09	75.99	203	0.0283
41.2	2.07	-22	60	525	0.32	0.1661	174.405	29.08	2.114	518	0.439	455.5	36.49	5.63	629.905	4.8	1.25	5.292	37.7	58.2	0.0152
12.3	2.07	-30	60	525	1.27	0.6581	692.065	109.194	8.39	518	0.6	1093.4	109.2	13.53	1785.456	47.1	25.875	38.777	75.11	220	0.0298
43.4	2.07	-22	60	500	0.26	0.1417	141.7	27.82	1.804	552	0.43116	476	36.79	5.53	617.7	4.1	0.866	5.43	33.98	58.2	0.0152
12.8	2.07	-30	60	500	1.24	0.6757	675.7	109.303	8.602	552	0.5	1130.7	109.59	13.137	1806.4	47.4	25.908	38.95	74.218	238	0.0314
46	2.07	-22	60	475	0.21	0.1206	114.475	26.92	1.530	587	0.4262	500.4	37.27	5.466	614.875	3.5	0.593	5.663	30.54	58.2	0.0152
13.3	2.07	-30	60	475	1.21	0.6941	659.395	109.896	8.835	586	0.5	1166.6	109.896	12.767	1825.995	47.5	25.988	39.097	73.315	256	0.033
45.7	2.07	24	60	450	0.185	0.112	100.8	26.57	1.42	620	0.2987	518.6	37.49	5.36	610.4	3.2	0.486	5.75	28.867	58.2	0.0152
13.9	2.07	-30	60	450	1.18	0.7145	643.05	109.998	9.095	620	0.477	1186.4	109.23	12.376	1839.45	47.4	26.12	38.79	72.4	273	0.0345
45.9	2.07	-24	60	425	0.145	0.093	79.05	26.02	1.18	654	0.2977	558	38.78	5.47	637.05	2.7	0.314918	6.345	25.37	58.2	0.0152
14.5	2.07	-30	60	425	1.145	0.7341	623.985	109.65	9.344	654	0.4509	1228.9	109.25	12.05	1862.885	47	26.0159	38.797	71.318	290	0.0361
46	2.07	-24	60	400	0.11	0.0749	59.92	25.625	0.9538	690	0.30305	610.3	40.73	5.67	670.22	2.3	0.1922	7.247	22.105	58.2	0.0152
15.1	2.07	-30	60	400	1.11	0.7561	604.88	109.49	9.6247	690	0.426	1262.1	109.225	11.73	1866.98	46.7	25.965	38.787	70.22	308	0.0378
															0						
41.7	2.07	-24	60	375	0.076	0.0552	41.4	25.318	0.7029	724	0.329	701.8	45.15	6.2166	743.2	1.8	0.09778	9.2795	18.374	58.2	0.0152
15.7	2.07	-30	60	375	1.075	0.7811	586.825	109.54	9.9427	724	0.4057	1295.1	109.616	11.472	1880.925	46.2	25.98	38.958	69.104	325	0.0396

TABLE 5.5 – REVIEWED PROPERTIES WITHIN MACHINE

PROPERTY	PROPERTY VALUE
Toroidal Field Outer Diameter	72.9mm
Toroidal Field Inner Diameter	43.2mm
Field turns	950, 0.315mm diameter wire
Total length	54mm (motor alone), 108mm inclusive
Length of stack	20mm
Outer diameter	92mm
Armature turns per phase	2348, 0.315mm diameter wire

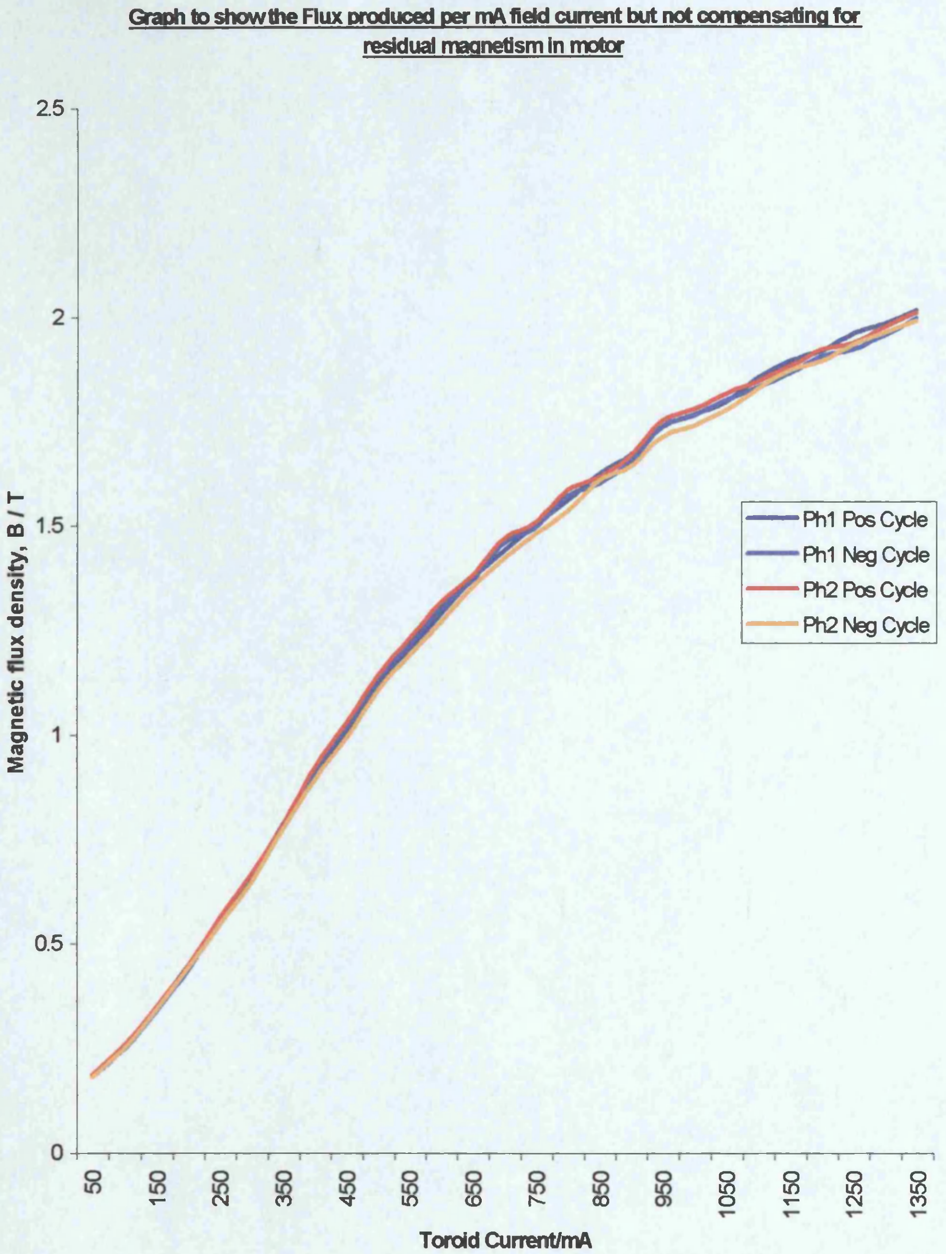


Figure 5.18 :- Flux density versus field current in toroidal field winding.

5.9 TEST RESULTS

From the graph of flux density against field current in the field winding (determined using the equations detailed in Section G.2 in the Appendices), saturation was seen to occur when the plot started to lose its linearity trend as the current increased (*Figure 5.18*). Saturation of the machine was seen to occur at about 1.22Tesla. This method was not totally accurate as the graph was clearly non-linear initially. An investigation to the efficiency of flux production was carried out by plotting a curve of magnetic flux per mA toroidal field current (*Figure 5.19*). At low field current there was the most flux per mA and this was due to the fact that back emf testing with the energisation of the field had magnetised the machine. The level of residual magnetisation (when 0mA field current was used) is found to be 0.191T. If this was taken into account and the graph of flux per mA field current was corrected then it was clear that the field had more efficient flux production at around 378mA but this was due to magnetic saturation occurring in the machine (*Figure 5.20*). This is another method to aid prediction of magnetic flux saturation.

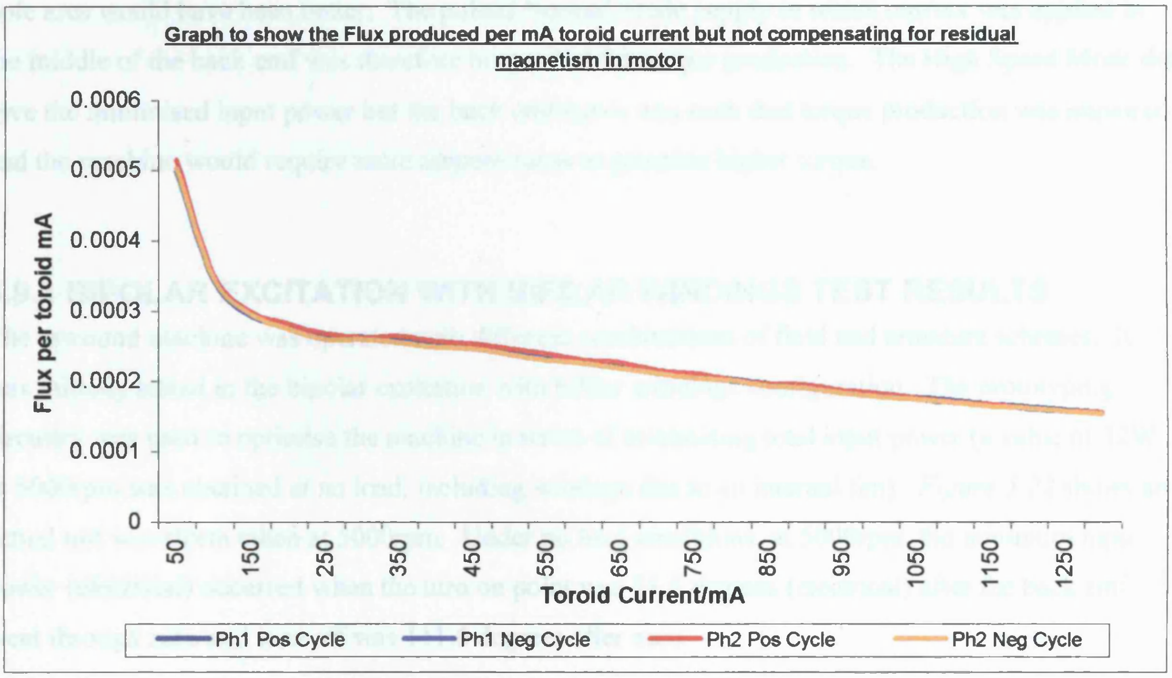


Figure 5.19 :- Flux produced per mA of field current.

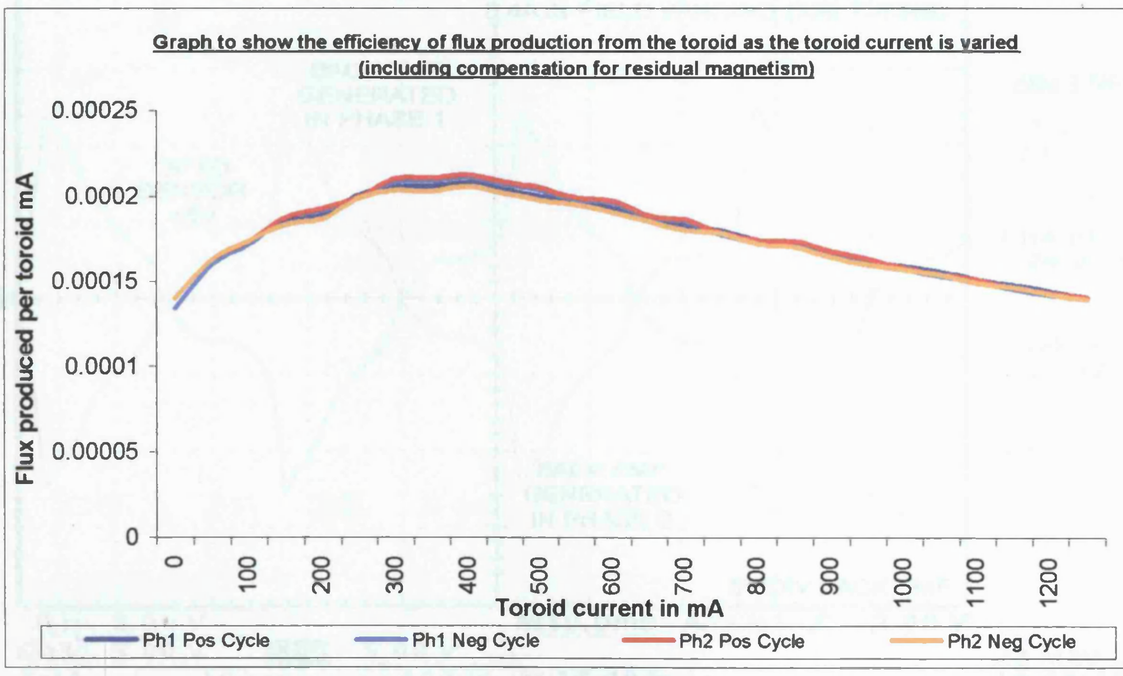


Figure 5.20 :- Corrected flux produced per mA of field current.

Back emf testing showed that the back emf waveform was almost triangular in shape (*Figure 5.21*). It was not a good shape for torque production as, for a given voltage supply to the armature windings, the initial low back emf allowed fast current rise without significant torque production and the peak in the back emf could exceed the supply voltage at higher speeds, reducing torque production. This back emf waveshape was caused by the pole arcs of the stator and rotor laminations. The pole arcs were optimised for the use of these laminations in a switched reluctance machine. In this machine wider pole arcs would have been better. The pulsed Normal Mode supply in which current was applied in the middle of the back emf was therefore best suited for torque production. The High Speed Mode did give the minimised input power but the back emf curve was such that torque production was impaired and the machine would require more ampere-turns to generate higher torque.

5.9.1 BIPOLAR EXCITATION WITH BIFILAR WINDINGS TEST RESULTS

The rewound machine was operated with different combinations of field and armature schemes. It was initially tested in the bipolar excitation with bifilar windings configuration. The prototyping circuitry was used to optimise the machine in terms of minimising total input power (a value of 32W at 5000rpm was obtained at no load, including windage due to an internal fan). *Figure 5.22* shows an actual test waveform taken at 5000rpm. Under no load conditions, at 5000rpm, the minimum input power (electrical) occurred when the turn on point was 21.6 degrees (electrical) after the back emf went through zero and turn off was 111.6 degrees after zero.

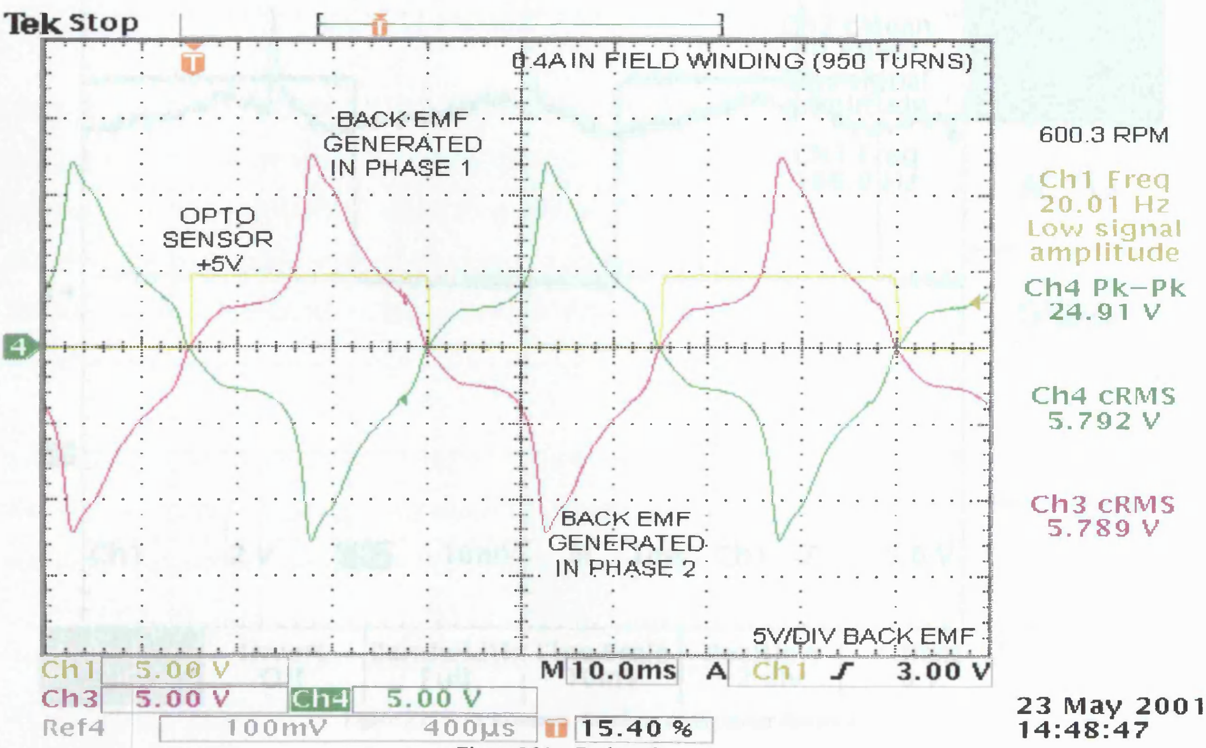


Figure 5.21 :- Back emf waveform.

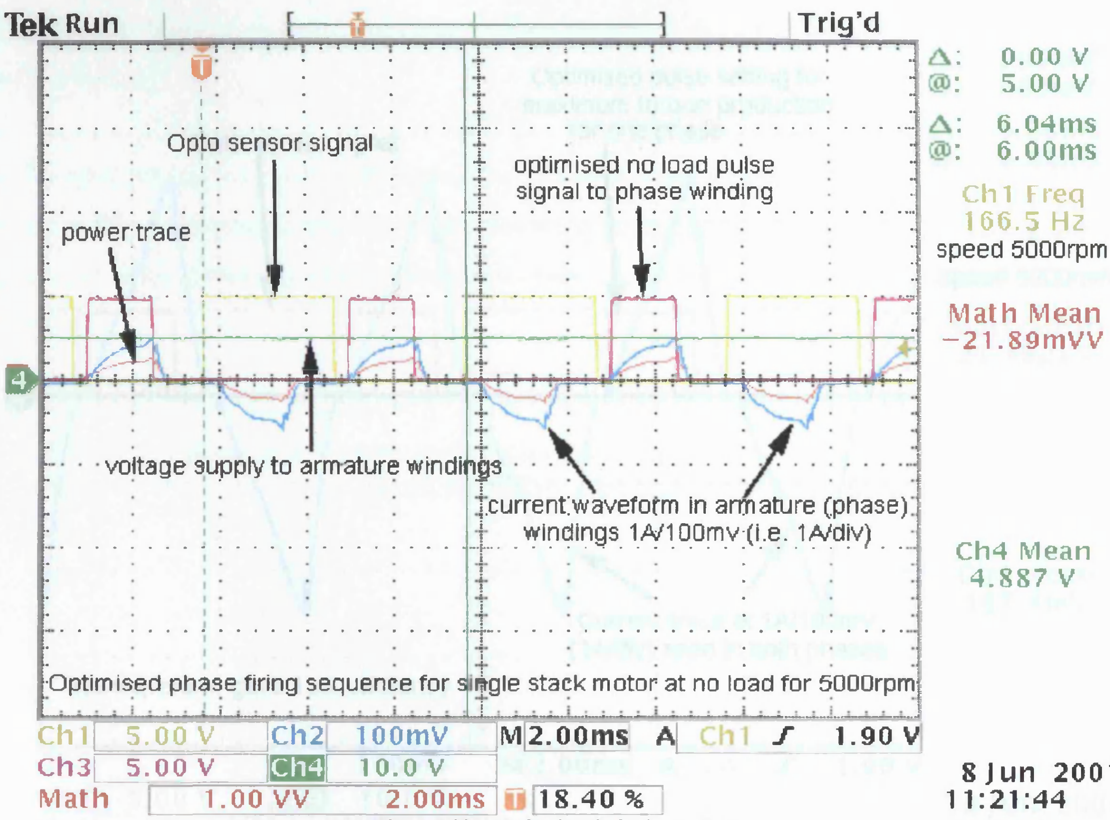


Figure 5.22 :- No load optimised waveform.

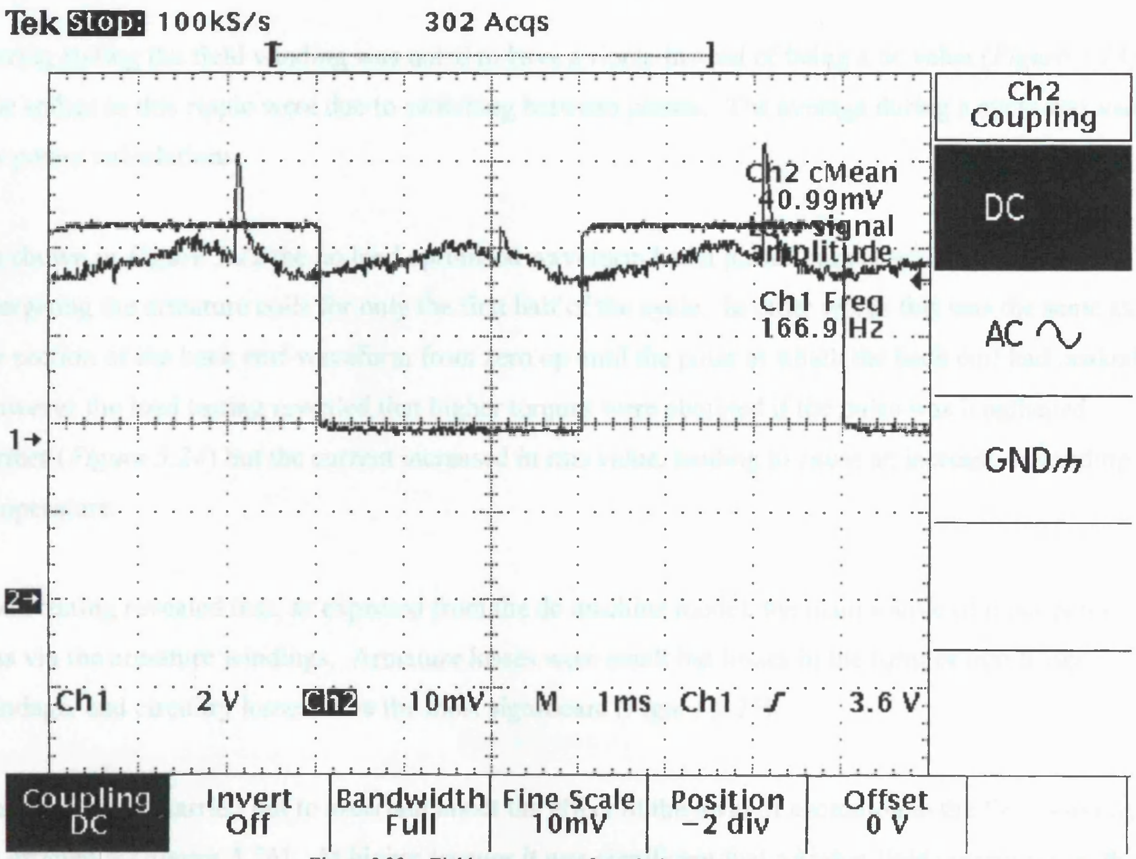


Figure 5.23 :- Field current waveform during motor operation.

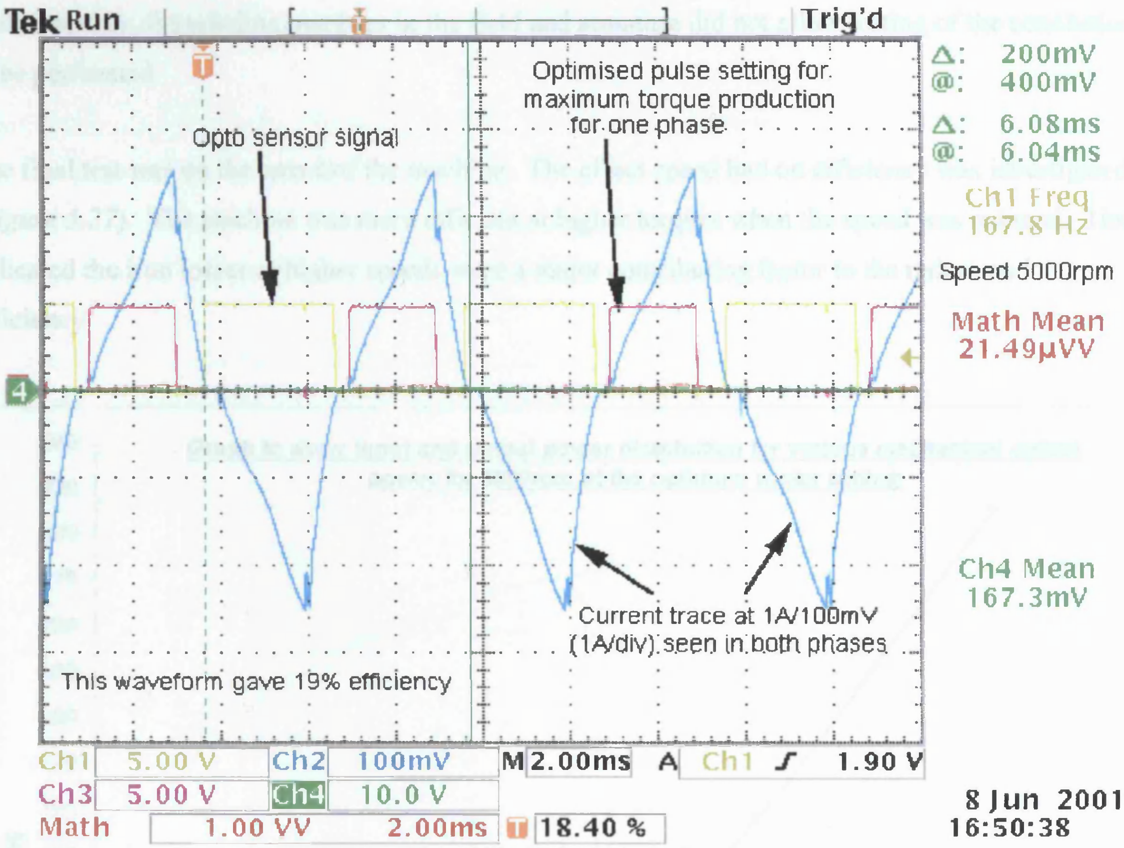


Figure 5.24 :- Higher load waveform.

During testing the field winding was noted to have a ripple instead of being a dc value (*Figure 5.23*). The spikes in this ripple were due to switching between phases. The average during a cycle was used for power calculations.

As shown in *Figure 5.22* the no load optimised waveform had a pulsed supply which involved energising the armature coils for only the first half of the cycle. In other words this was the same as the section of the back emf waveform from zero up until the point at which the back emf had peaked. However the load testing revealed that higher torques were obtained if the pulse was lengthened further (*Figure 5.24*) but the current increased in rms value, tending to cause an increase in winding temperature.

Load testing revealed that, as expected from the dc machine model, the main source of input power was via the armature windings. Armature losses were small but losses in the form of iron losses, windage, and circuitry losses were the most significant (*Figure 5.25*).

Tests were then carried out to ascertain about the effect of the level of excitation in the field winding on efficiency (*Figure 5.26*). At higher torques it was significant that a higher field current raised the efficiency. This suggested that it is appropriate to have the field in series with the armature.

Unfortunately, the winding numbers in the field and armature did not allow testing of the conclusion to be performed.

The final test was on the speed of the machine. The effect speed had on efficiency was investigated (*Figure 5.27*). The machine was more efficient at higher torques when the speed was reduced. This indicated the iron losses at higher speeds were a major contributing factor to the reduction in efficiency.

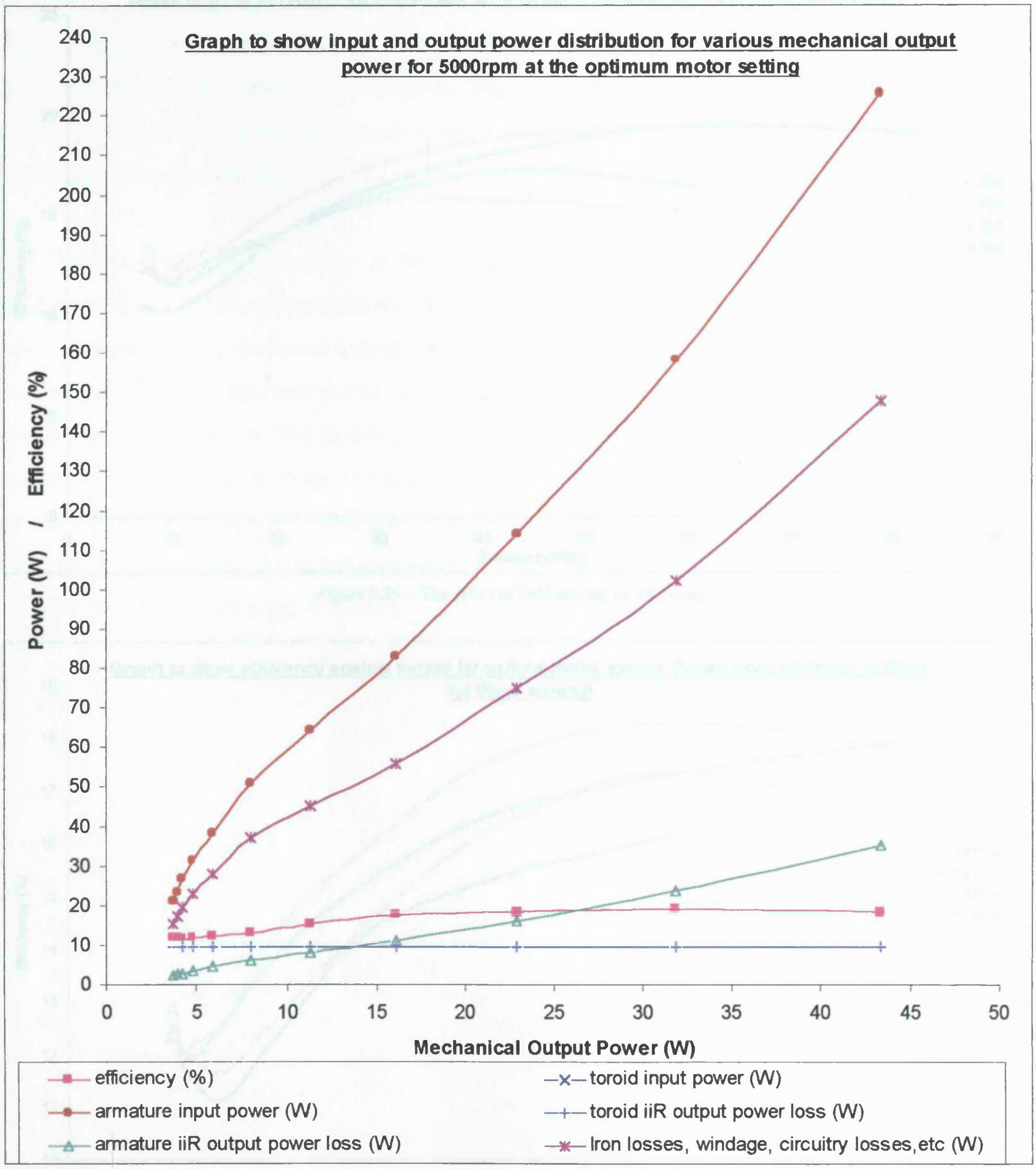


Figure 5.25 :- Input and Output Power distribution and efficiency over a range of mechanical output powers.

The machine was noted to self-start throughout all the tests.

Load testing yielded a peak efficiency of 19%. However, the stack and the rotor poles were laminated and this did not assist 3-dimensional flux paths. It was harder to drive flux across the laminations. Eddy currents due to cross lamination fluxes were significant. The model overestimated magnetic fluxes because of this. It was practically impossible to correct the model for such a situation.

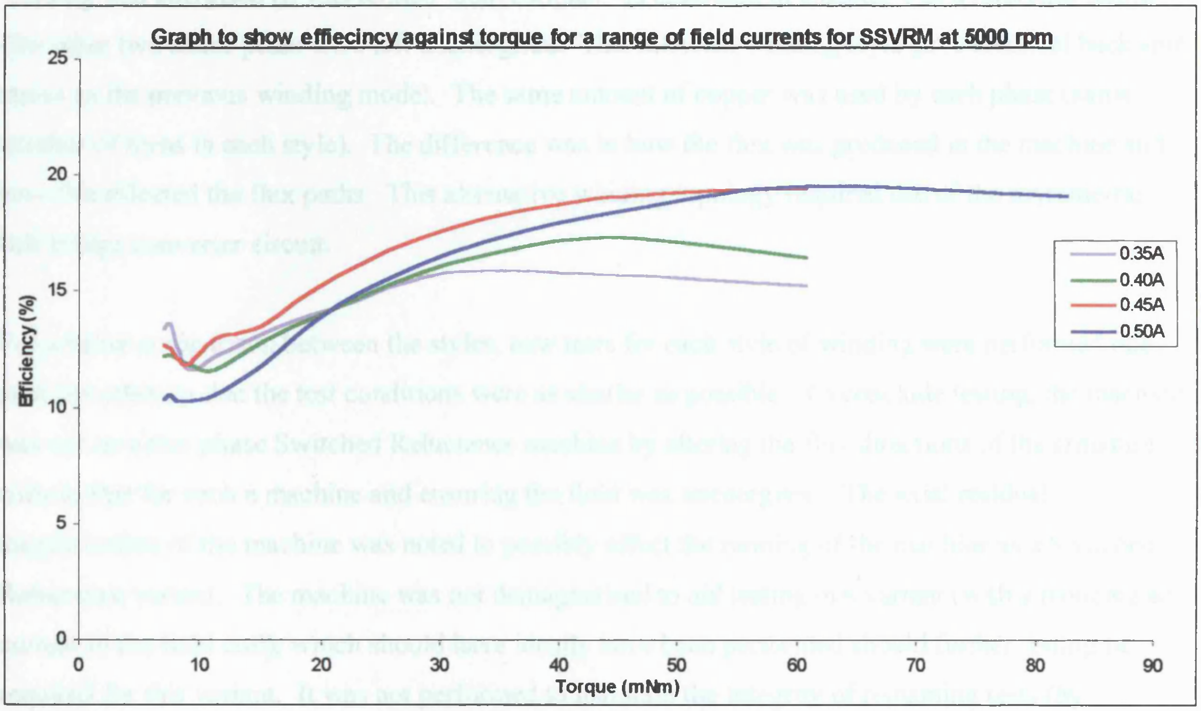


Figure 5.26 :- The effect of field current on efficiency.

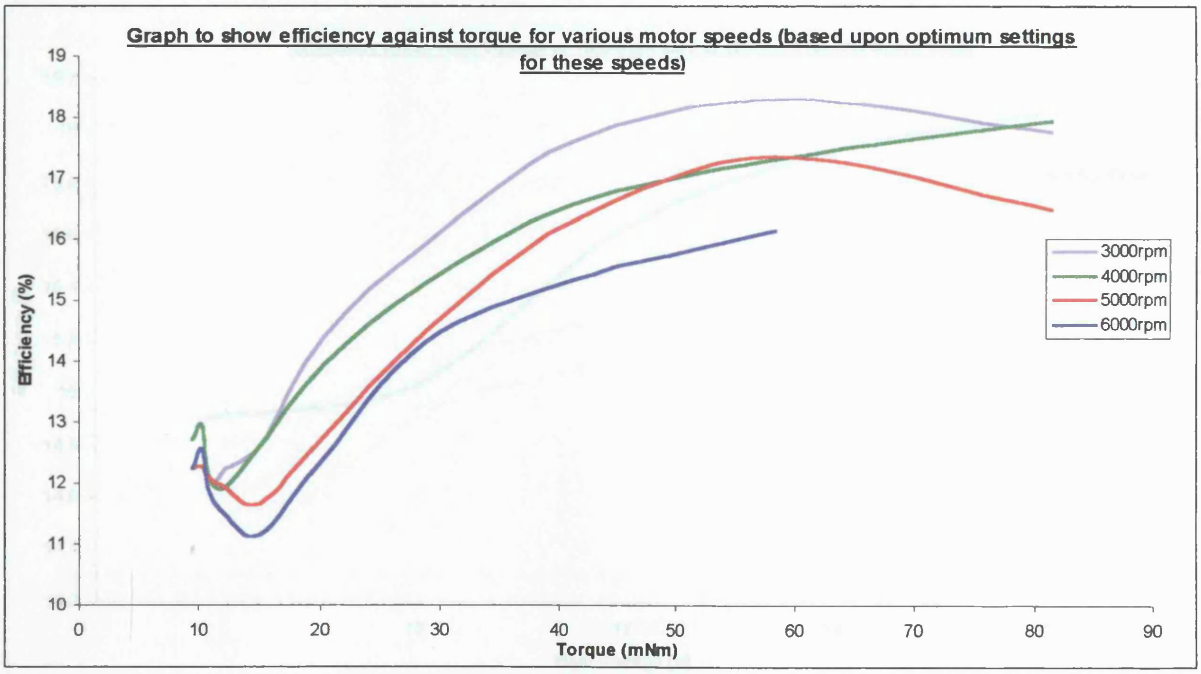


Figure 5.27 :- The effect of speed on efficiency.

5.9.2 UNIPOLAR EXCITATION WITH SINGLE COIL PER POLE WINDING ('CONVENTIONAL') TEST RESULTS

Previous tests were carried out using bipolar excitation with bifilar windings where flux is produced in all the stator poles when each phase was energised but only half the copper was used in each pole. The unipolar excitation with single coil per pole windings method for this style of machine was to have all the copper for two diametrically opposite stator poles used for one phase only and that the flux produced by each phase aided the direction of flux into or out of the poles due to how the field winding was energised (it was termed 'conventional' as described in Chapter 4 as a reference term). The other two stator poles were left unenergised. This different winding style gave identical back emf traces as the previous winding model. The same amount of copper was used by each phase (same number of turns in each style). The difference was in how the flux was produced in the machine and how this affected the flux paths. This alternative winding topology required use of the asymmetric half bridge converter circuit.

For a better comparison between the styles, new tests for each style of winding were performed one after the other so that the test conditions were as similar as possible. To conclude testing, the machine was run as a two phase Switched Reluctance machine by altering the flux directions of the armature coils to that for such a machine and ensuring the field was unenergised. The axial residual magnetisation of the machine was noted to possibly affect the running of the machine as a Switched Reluctance variant. The machine was not demagnetised to aid testing this variant (with a reducing ac current in the field coil), which should have ideally have been performed should further testing be required for this variant. It was not performed to maintain the integrity of remaining tests (by maintaining the residual magnetisation).

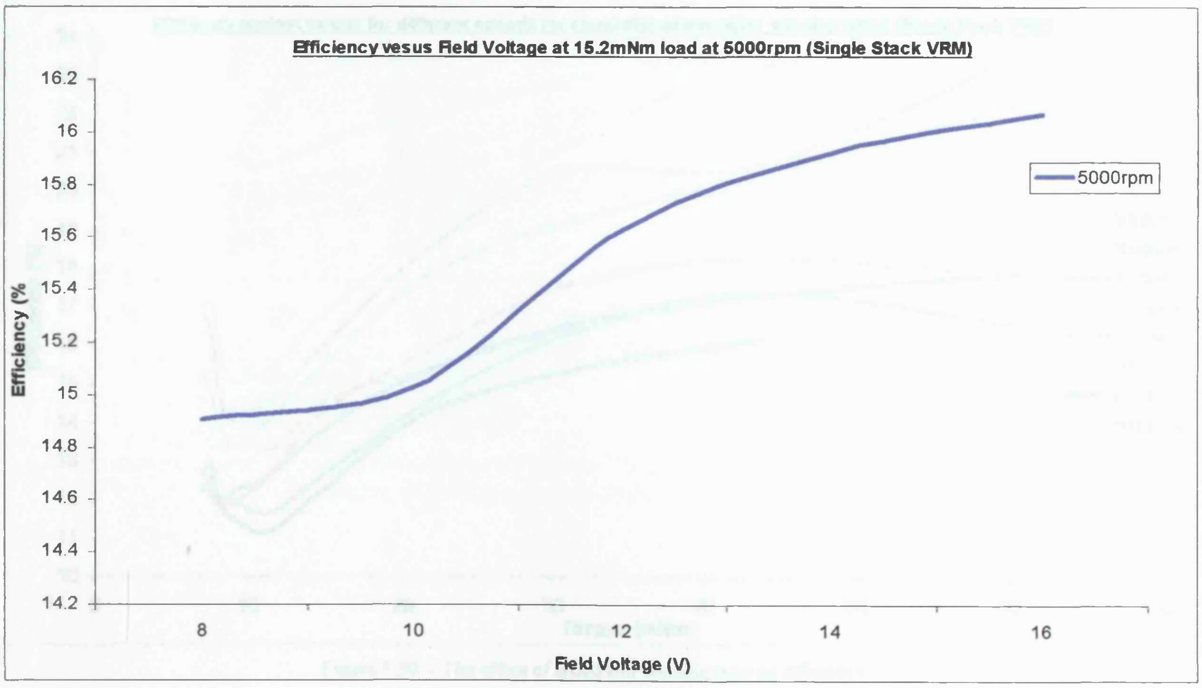


Figure 5.28 :- The effect of field supply on efficiency.

As with the bipolar excitation with bifilar windings machine, the efficiency of the unipolar excitation with single coil per pole windings machine was improved as the field current increased (for a fixed speed and fixed phase timings) (*Figure 5.28*). The machine was also noted to improve efficiency as the speed was reduced (*Figure 5.29*). This followed the same trend as noted previously with the bipolar variant. When comparing the efficiencies of the bipolar and unipolar styles there were noticeable differences. For any given speed the bipolar excitation with bifilar windings current machine (first version) was shown to be less efficient (*Figure 5.30*).

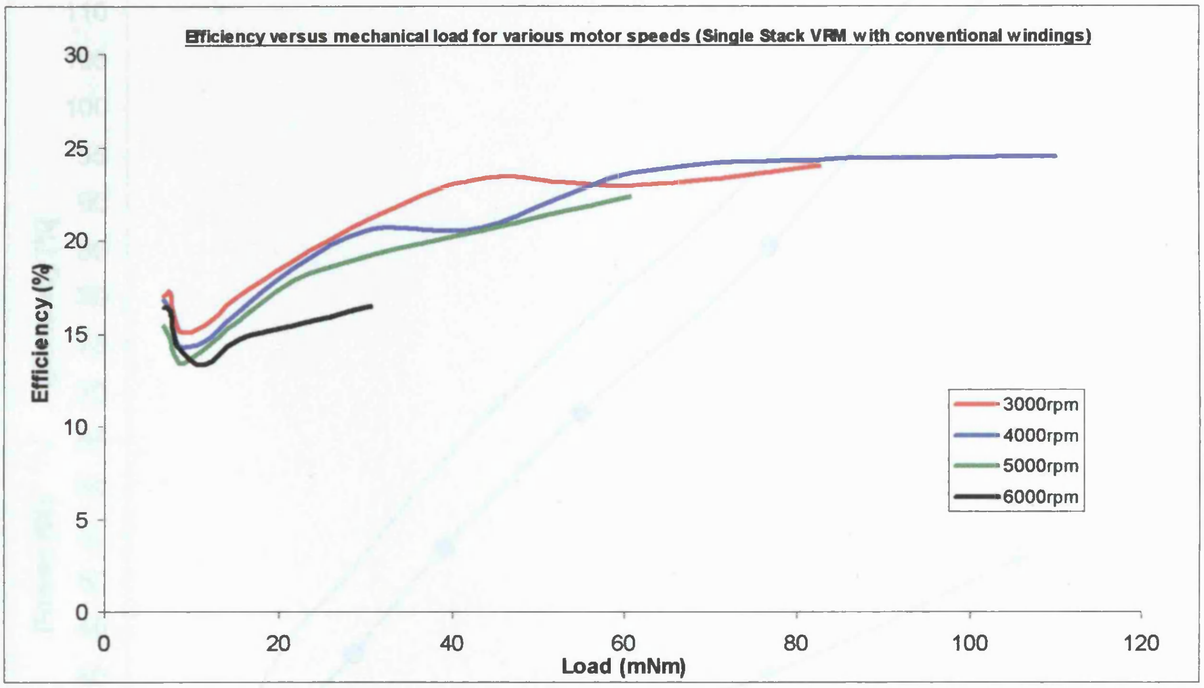


Figure 5.29 :- The effect of speed on efficiency.

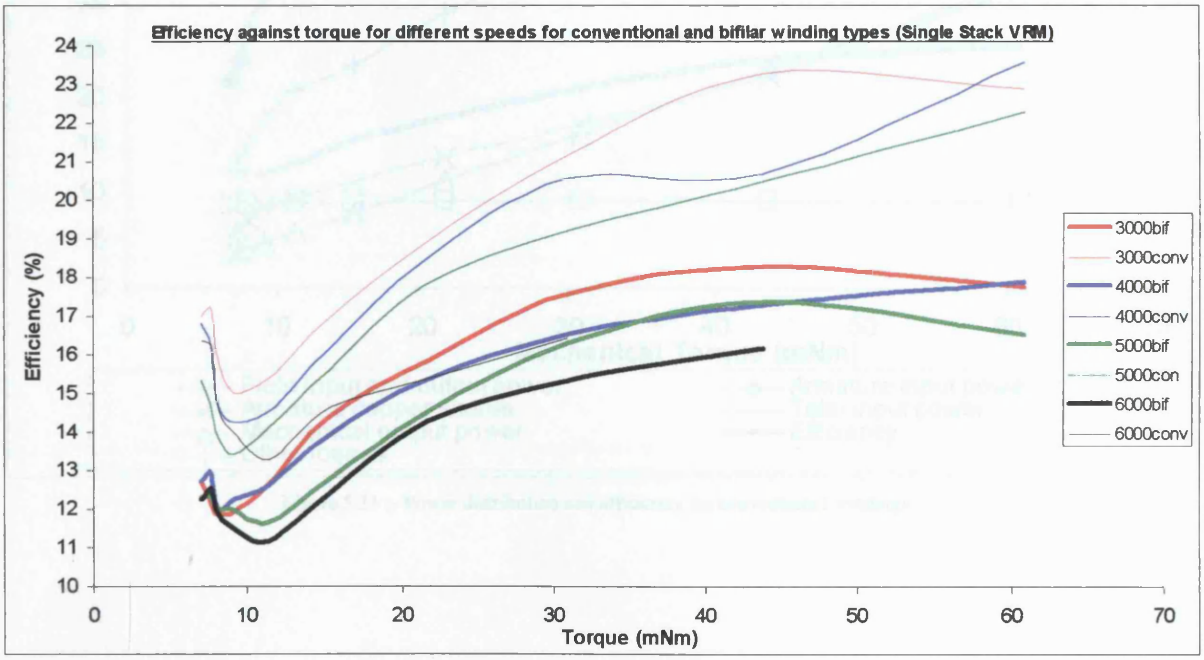


Figure 5.30 :- The effect of speed and winding type on efficiency.

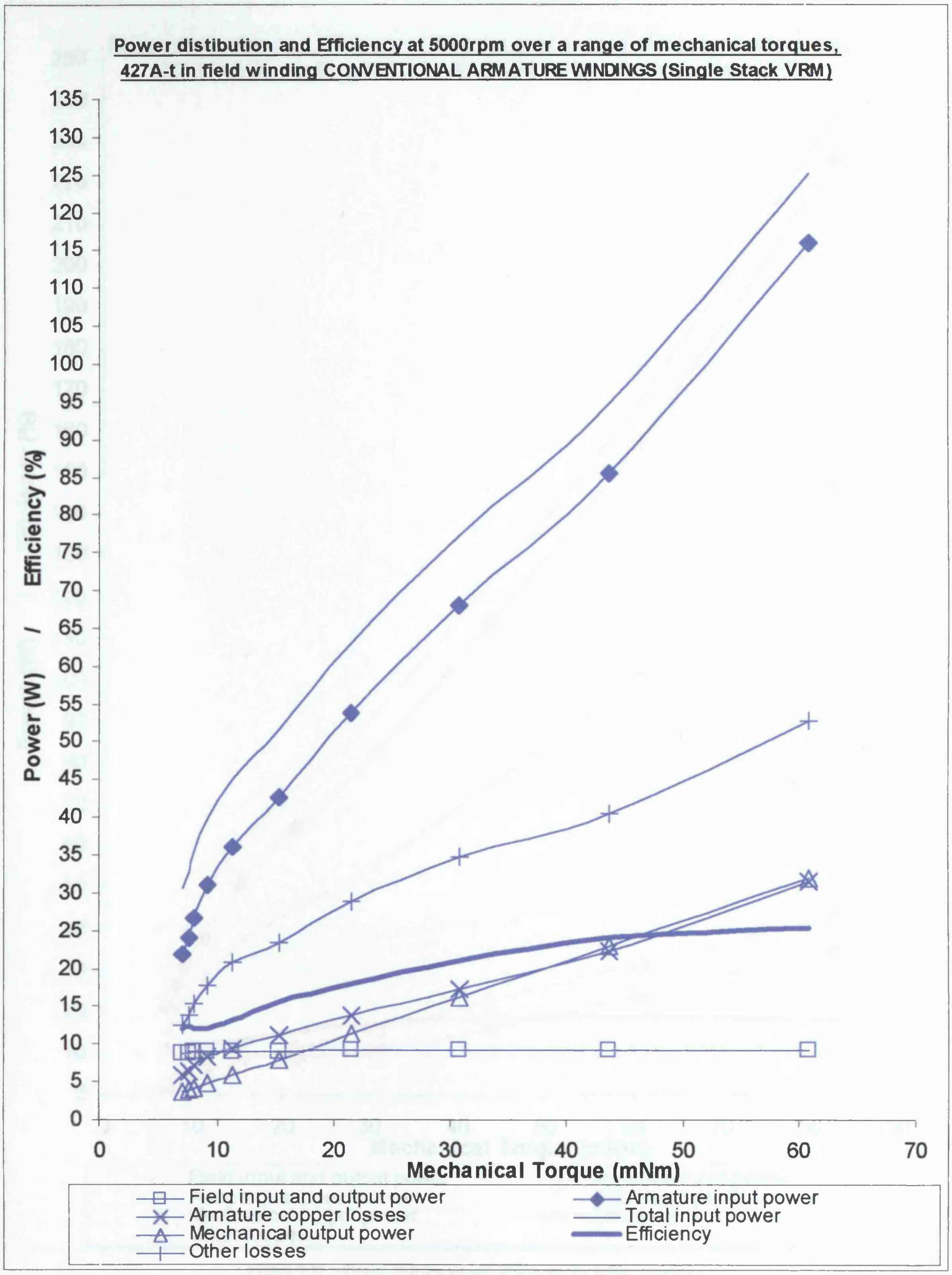


Figure 5.31 :- Power distribution and efficiency for conventional windings.

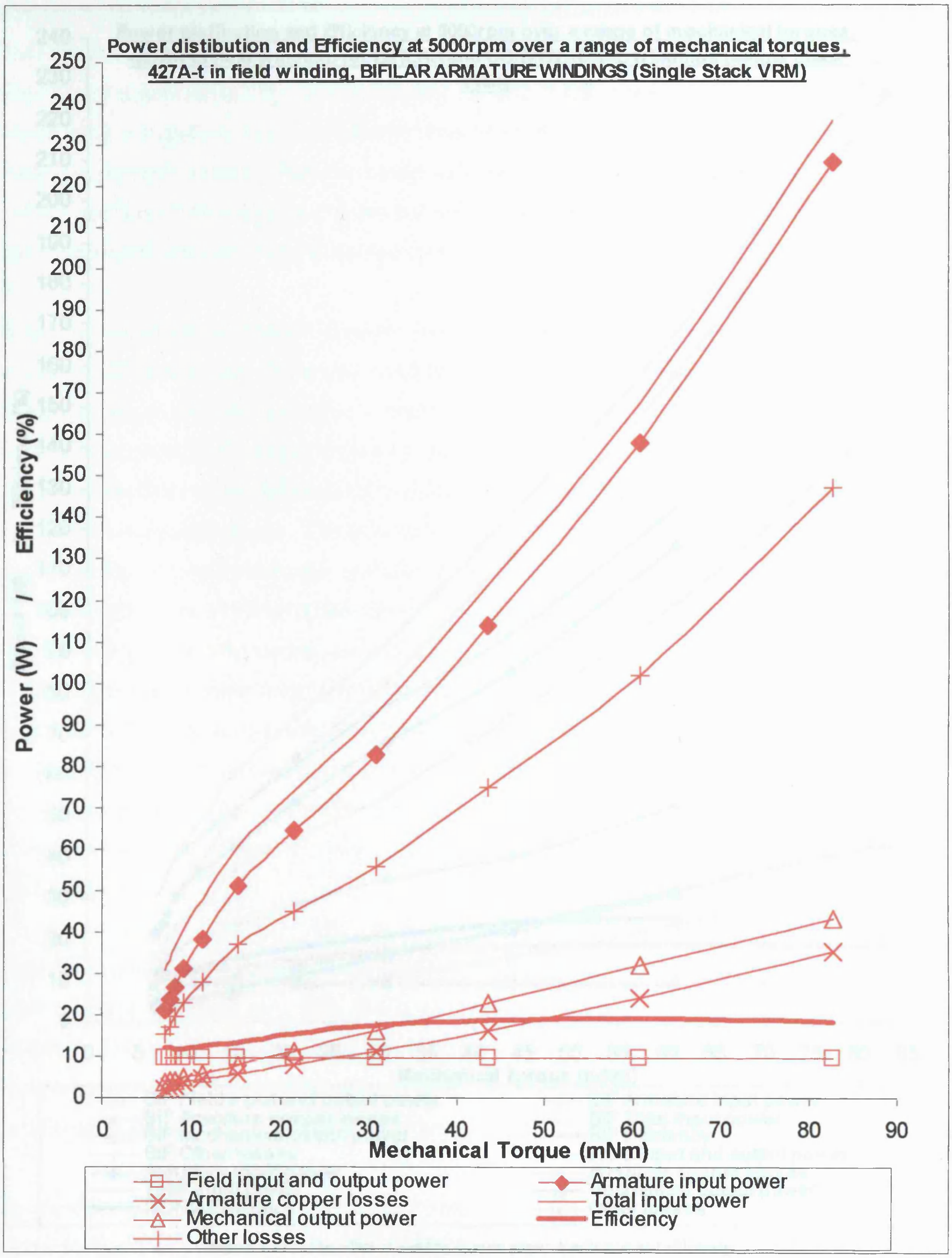


Figure 5.32 :- Power distribution and efficiency for bifilar windings

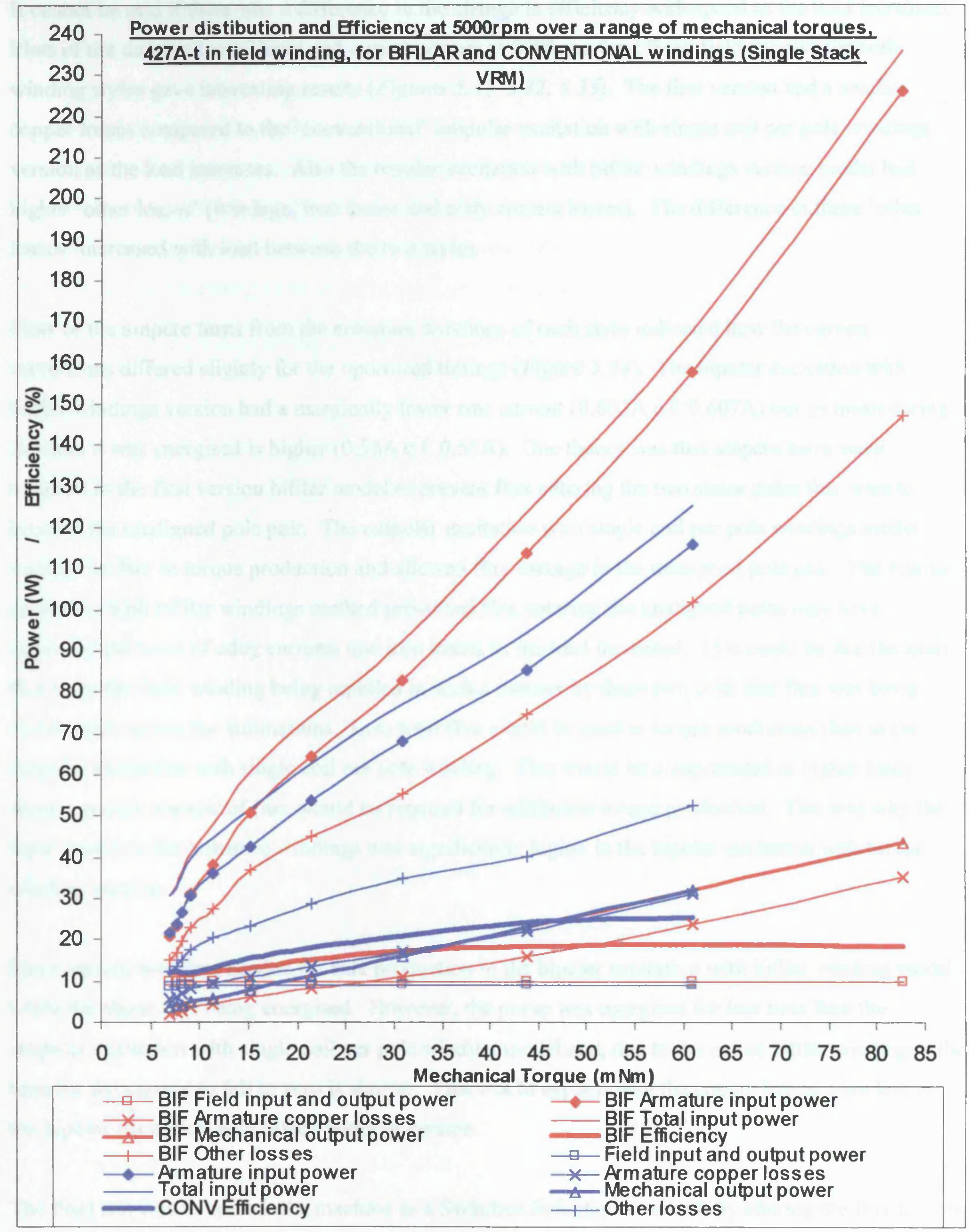


Figure 5.33 :- The effect of winding type on power distribution and efficiency.

It cannot be said if there was a difference in the change in efficiency with speed as the load increased. Plots of the distribution of input and output powers at 5000rpm (and fixed field supply) for both winding styles gave interesting results (*Figures 5.31, 5.32, 5.33*). The first version had a smaller copper losses compared to the 'conventional' unipolar excitation with single coil per pole windings version as the load increases. Also the bipolar excitation with bifilar windings version model had higher 'other losses' (windage, iron losses and eddy current losses). The difference in these 'other losses' increased with load between the two styles.

Plots of the ampere turns from the armature windings of each style indicated how the current waveforms differed slightly for the optimised timings (*Figure 5.34*). The bipolar excitation with bifilar windings version had a marginally lower rms current (0.602A c.f. 0.607A) but its mean during the time it was energised is higher (0.58A c.f. 0.55A). One theory was that ampere turns were required in the first version bifilar model to prevent flux entering the two stator poles that were to become the unaligned pole pair. The unipolar excitation with single coil per pole windings model used all its flux in torque production and allowed flux leakage in the unaligned pole pair. The bipolar excitation with bifilar windings method prevented flux entering the unaligned poles may have increased the level of eddy currents and iron losses as the load increased. This could be due the axial flux from the field winding being repelled in such a manner by these two coils that flux was being driven back across the laminations. Less total flux would be used in torque production than in the unipolar excitation with single coil per pole winding. This would be compounded at higher loads since a greater amount of flux would be required for additional torque production. This was why the input power to the armature windings was significantly higher in the bipolar excitation with bifilar winding version.

More current was thus needed for flux production in the bipolar excitation with bifilar winding model while the phase was being energised. However, the phase was energised for less time than the unipolar excitation with single coil per pole winding model and, due to the use of bifilar windings - the time for the current to fall to zero is shorter. This would explain why the copper losses were less in the bipolar excitation with bifilar winding version.

The final test was to operate the machine as a Switched Reluctance variant, by altering the flux to flow within the laminated sections and to not have the field winding energized (*Figure 5.34*). The machine was heavily magnetised to the point of parking itself in an aligned position when not operated so it was not an ideal test for a switched reluctance operation. The current waveform required for 5000rpm was optimised such that when the each phase was switched off there was sufficient time for the current waveform to fall towards zero. The machine did self start during testing but its efficiency was found to be very poor in comparison to the other winding types. The high current required to run the machine gave it an overheating problem which meant only two test results were taken (*Figure 5.35*).

It was conclusive that this machine operated more efficiently when a unipolar excitation with single coil per pole winding arrangement was used. The high iron losses due to the use of laminated material to carry flux perpendicular to the plane of the laminations meant that slower rotor speeds aided efficiency. Having the field in series with the armature as in a current source half bridge converter circuit was an ideal arrangement as an increase in field current boosted efficiency.

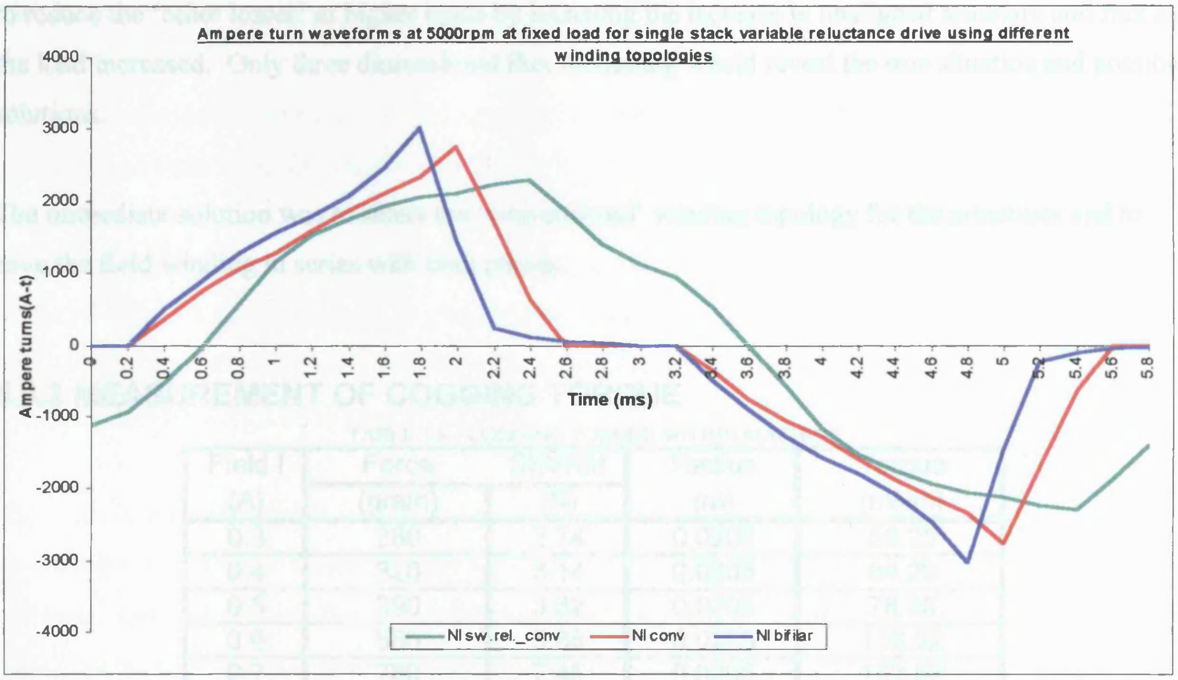


Figure 5.34 :- Current waveforms (as A-t version) for various winding types

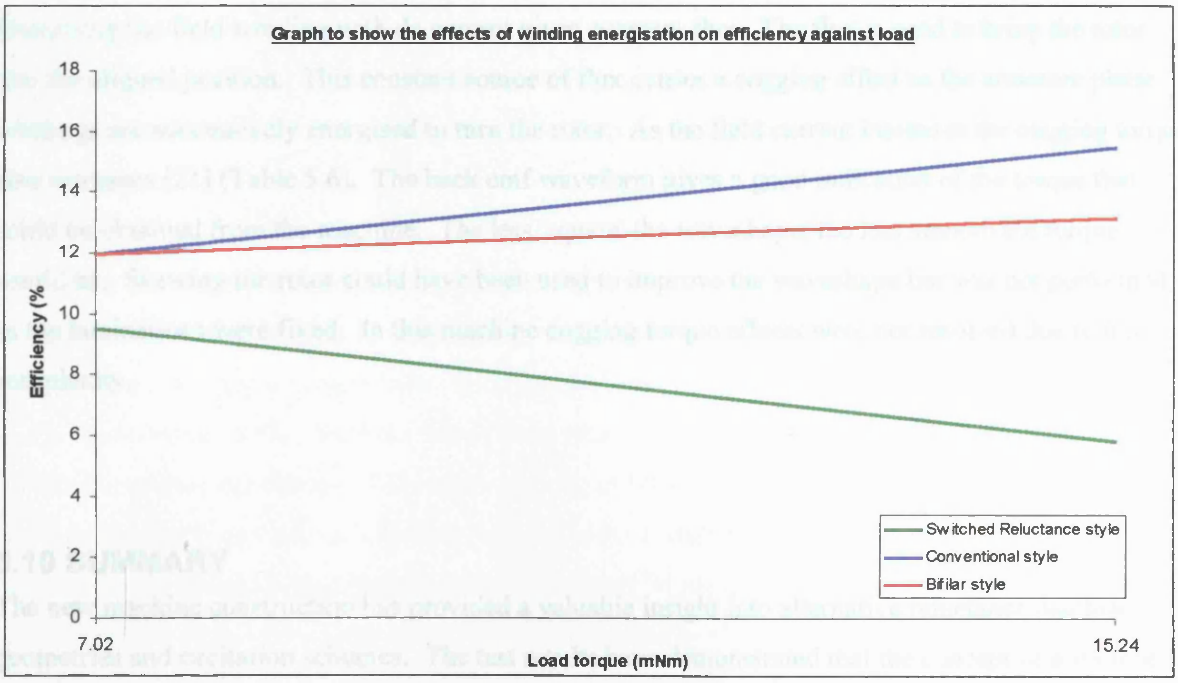


Figure 5.35 :- Efficiency for various winding types with load.

The bipolar excitation with bifilar winding version did not benefit torque production as first thought in this machine. Bipolar excitation may possibly detrimentally reduce the amount of axial flux entering the stator poles. Energy was wasted in preventing flux entering the unaligned stator poles. It was proposed that, as the armature current increases, flux produced in these unaligned poles detrimentally affected the flux paths for the aligned poles causing additional losses. More total flux was required to generate the torque required for a given load. Having the field in series with the armature allowed more field flux to be present as the load increased. It was speculatively suggested that this arrangement could benefit the bifilar winding by having more axial flux from the field at higher loads to reduce the 'other losses' at higher loads by lessening the increase in unaligned armature coil flux as the load increased. Only three dimensional flux modelling would reveal the true situation and possible solutions.

The immediate solution was to select the 'conventional' winding topology for the armatures and to have the field winding in series with both phases.

5.9.3 MEASUREMENT OF COGGING TORQUE

TABLE 5.6 – COGGING TORQUE WITHIN MACHINE

Field I (A)	Force SSVRM (gram)	(N)	Radius (m)	Torque (mNm)
0.3	280	2.74	0.0205	56.25
0.4	320	3.14	0.0205	64.29
0.5	390	3.82	0.0205	78.35
0.6	580	5.68	0.0205	116.52
0.7	760	7.45	0.0205	152.68

Energising the field winding with dc current gives constant flux. The flux is used to bring the rotor into the aligned position. This constant source of flux causes a cogging effect as the armature phase windings are successively energised to turn the rotor. As the field current increases the cogging torque also increases [21] (Table 5.6). The back emf waveform gives a good indication of the torque that could be obtained from the machine. The less 'square' the waveshape, the less smooth the torque would be. Skewing the rotor could have been used to improve the waveshape but was not performed as the laminations were fixed. In this machine cogging torque effects were not resolved due to this complexity.

5.10 SUMMARY

The new machine construction has provided a valuable insight into alternative reluctance machine geometries and excitation schemes. The test results have demonstrated that the concept of a switched reluctance variant of the homopolar inductor alternator works and that control of the machine characteristics can be achieved via either the field current or the phase winding (armature) current. The power electronics only requires two power switches and can produce torque at all rotor angles.

The field winding is a very efficient source of magnetic flux and could be well suited to be placed in series with the armature windings.

The effects of end thrust could be used in a beneficial manner if the load produces an axial force opposing it as the forces could cancel each other out.

The machine appears to be more suited to a current source half bridge converter circuit to energise the armature and field windings when using bipolar excitation with bifilar windings but the asymmetric half bridge converter is the circuit for unipolar excitation with single coil per pole windings which is the winding type for best torque production. This is due to the machine being more efficient when the field coil carries a higher current. As the load increases so too does the current required to produce the increased torque. A new model with improved turns ratios between the armature and phase windings would allow this circuit to be used effectively.

The main disadvantage of the machine, as constructed, has been its low efficiency. The test results have shown that the iron losses are very significant. This was to be expected from the use of laminated material to carry flux perpendicular to the plane of the laminations. It is proposed that a significant improvement in performance could be obtained through the use of powdered iron material in parts of the machine structure. It is suggested that this machine is more suited to powdered iron than a conventional switched reluctance machine. The use of unipolar excitation with single coil per pole windings with each phase on one diametrically opposite pair of stator poles gives a greater efficiency than each phase having windings on all stator poles as more flux is used in torque production.

One factor that partly explains efficiency limitations with this design is caused by the use of the field winding itself. The dc current in the winding would give dc flux axially within the rotor, when considered as the only source of flux. This would not cause significant circumferential eddy currents in the steel sections within the field coil as there was small change in flux magnitude to cause the eddy currents to oppose the change. But a sense coil around the field winding did show a ripple in the flux, and spikes of back emf where switching occured in the armature coils of each phase. Thus there was a degree of eddy current losses in these motors. It would be possible to reduce the circumferential eddy currents by use of powdered iron or by breaking the paths by insertion of e.g. a plastic spacer so flux can still travel axially as desired but electrically insulating from eddy currents.

CHAPTER 6

THE PERMANENT MAGNET FLUX SWITCHING MOTOR (PMFSM)

6.1 INTRODUCTION

Magnets are used in many machines, from dc motors to hysteresis brakes. They provide dc flux without the need for any input of electrical power. They could be used to replace field coils that provide permanent constant dc flux. If done successfully, the machine would be able to provide the same torque but at reduced total input power (i.e. higher efficiency).

To replace a field winding with a permanent magnet, the nature by which flux is produced needs investigating. A field winding is a coil of wire (usually many turns) that passes current to produce flux in the middle of the coil in an axial direction. The coil is often surrounding a steel section that is used to direct the flux in preference to air. The coil forms no part of the magnetic circuit except that the steel sections of the magnetic circuit have to be designed such that space exists for the field coil.

This chapter introduces the use of permanent magnets in Flux Switching Motors. The design of such a motor for applications up to 100W will be discussed.

Small motors are prone to high field magnetising losses, resulting in low efficiencies. The use of permanent magnets will be shown to improve the efficiency through replacing the field windings of Flux Switching Motor (FSM) designs.

It is difficult to make small motors that are efficient. Except for designs such as the permanent magnet brushless dc motor, the smaller the motor the greater the proportion of losses associated with the windings (high field magnetising losses). It is common for induction motors with an outer diameter in the region of 2 to 3 inches to have efficiencies in the region of 10-30%.

One of the potential applications of such a design is for low power fans (100W or less). This chapter shows the design of a PMFSM having the same external dimensions as an induction motor that drives a 100mm fan (the Domus S1R 100mm Axial Wall Fan). The PMFSM would match the induction motor in terms of its load condition (same rpm and torque hence same net output power) but better the overall efficiency by as much as possible using the cheapest and fewest magnets possible.

Motor designs are introduced as a precursor to a direct replacement to the induction motor. The replacement motor will be shown to have a reduced copper loss and reduced input power for identical output power compared to the similarly sized induction motor. The chapter demonstrates that it is possible to increase the efficiency of the Permanent Magnet Flux Switching Motor (PMFSM) at potentially no extra cost.

Small machines are well suited for permanent magnet usage as the magnetizing losses are higher in smaller machines. By using magnets these losses can be reduced. It is possible to replace a field winding with permanent magnets in the FSM. In magnets, flux is produced from the magnet material. This means that the magnet has to be a part of the magnetic circuit. This poses interesting design issues. As the field winding is no longer required, the space available for it is no longer needed and is thus available for machine steel and/or magnet. The magnet also has a higher reluctance to an equivalent-sized steel section. The total reluctance of the machine would be different when magnets are used.

Any Switched Reluctance Machines with two or more phases is a potential candidate for magnet insertion. Take, for example, the two phase 8/12 design (*Figure 6.1*). By looking at the flux paths when each phase is energized, there are four sections in the back iron where flux is always in the same direction (uni directional) and four sections where the flux is bidirectional. It is theoretically possible to place magnets in the sections of back iron where flux was unidirectional. This has been shown by Lipo in [12, 13, 15, 17, 20]. A NASA report [50] also shows, as discussed in Section 2.3.3 of Chapter 2, uni- and bi-directional flux paths within the stator of a Flux Switch Generator.

The flux from the inserted magnets is in the same direction as the unidirectional flux. If the magnet properties and dimensions are selected correctly then the magnet flux aids flux produced by the armature coils. With the use of magnets, less armature ampere-turns are required for the same total level of flux within the machine, resulting in a possible efficiency boost.

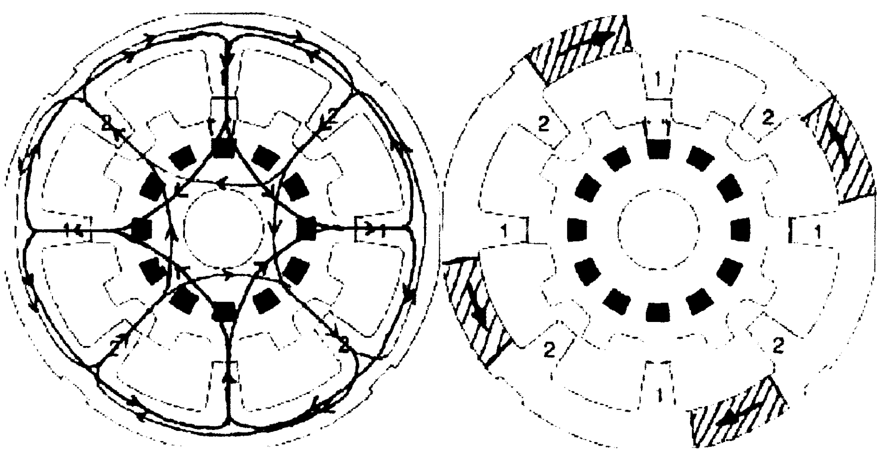


Figure 6.1 :- 8/12 Switched Reluctance Machine with back iron sections that could have magnets included.

In a brushless dc machine, the magnets are on the surface of the rotor and the air gap flux density is the same as the magnet flux density. Buried or interior permanent magnet rotors have the magnet at right angles to the air gap. The air gap flux density is greater than the magnet flux density because the area at the air gap is reduced, focussing the flux.

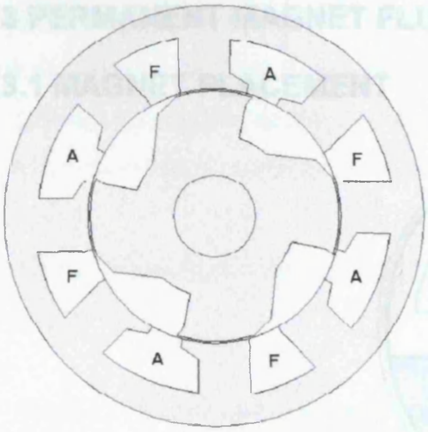
Stator based magnets, where the stator teeth are thinner than the magnet surface area, allow the use of ferrite magnets with a flux density in the region of 0.2T to give in the region of 1T in the air gap due to the reduction in area. The use of thinner teeth could produce a back emf trace that has an undesirable high peak waveshape. Widening the teeth would improve the back emf waveshape and reduce the reluctance in the air gap but this would also reduce flux focussing in the air gap. A trade off is necessary. There is less difficulty in constructing a suitable stator for a PMFSM when compared to the placement of magnets buried in, for example, a four segment rotor.

6.2 PERMANENT MAGNETS AND THE FLUX SWITCHING MOTOR

6.2.1 FLUX SWITCHING MOTOR FLUX PATHS

An example of a lamination that has been used in the Flux Switching Motor is shown in *Figure 6.2.* The armature and field windings found within the motor are shown in *Figure 6.3.* The field winding slot carries dc current at all times and the armature slots are energised with bi-directional mmf, alternating every 45°. Further details of such motors can be found in Chapter 2 (Section 2.2.11 and *Figure 2.45*) and [48].

Magnetic flux paths within the Flux Switching Motor are shown in *Figures 6.4, 6.5* and *6.6* for each polarity of armature current (green and blue) for a two phase armature winding (bifilar wound) (current directions shown by arrows) (*Figure 6.6*). As is shown, the back-iron has sections where the flux is bi-directional (behind the armature slots) and sections where the flux is uni-directional (behind the field slots). Four orange arrows are where flux is uni-directional. The direction of the arrows are in the sense of the flux paths. It is therefore possible to remove the field winding altogether and replace the back iron sections of unidirectional flux with permanent magnets. The space that held the field windings could be used to allow more space for the permanent magnet. It is then possible to have a smaller machine using permanent magnets that operated at reasonably high efficiencies, due to removal of field losses and a reduction in magnetizing losses. This machine is termed the Permanent Magnet Flux Switching Motor.



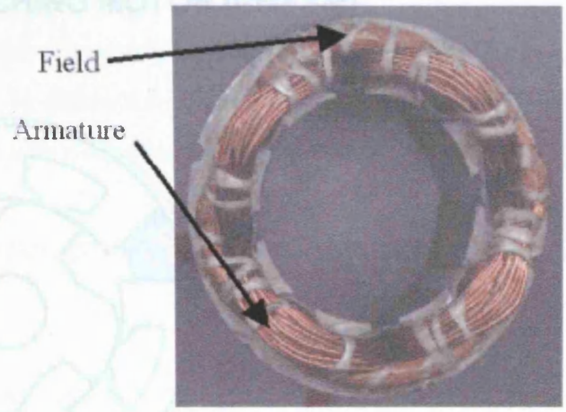


Figure 6.2 :- Example of lamination used in Flux Switching Motor. Figure 6.3 :- Field and armature windings of Flux Switching Motor.

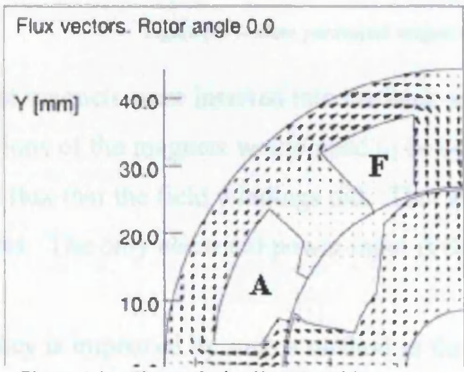


Figure 6.4 :- Flux paths in aligned position.

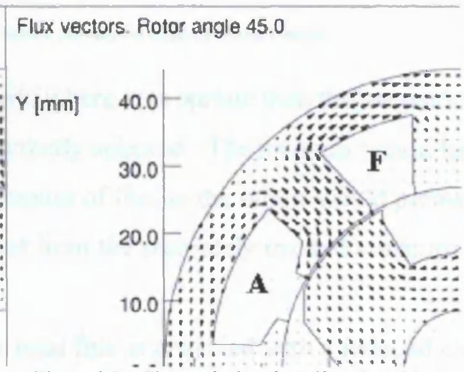


Figure 6.5 :- Flux paths in other aligned position.

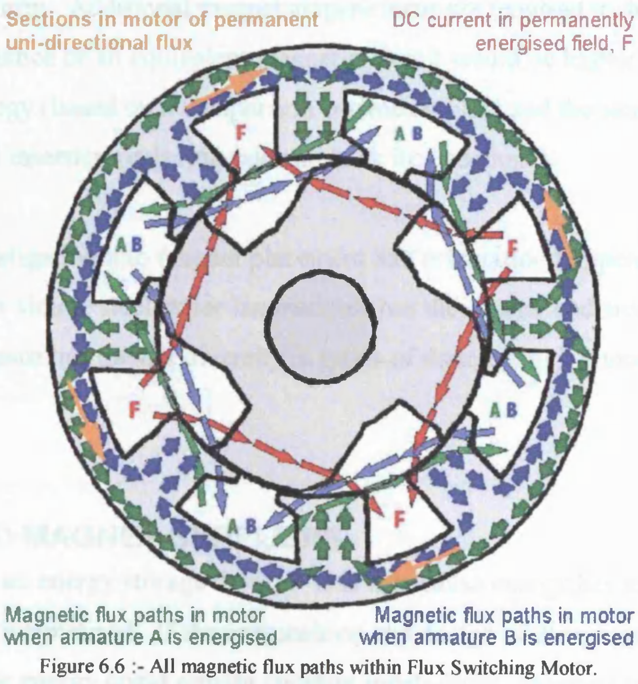


Figure 6.6 :- All magnetic flux paths within Flux Switching Motor.

6.3 PERMANENT MAGNET FLUX SWITCHING MOTOR (PMFSM)

6.3.1 MAGNET PLACEMENT

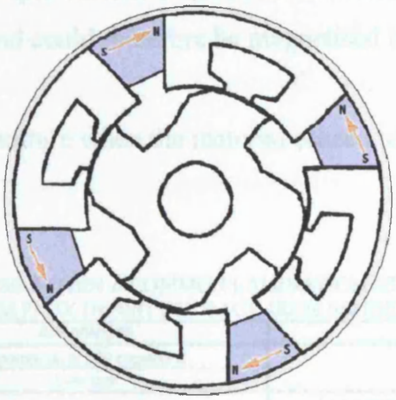


Figure 6.7 :- More permanent magnet space possible within an initial design.

If permanent magnets were inserted into the back-iron where appropriate then the magnetic properties and dimensions of the magnets would need to be correctly selected. The magnets would have to provide the flux that the field windings did. That amount of flux in the motor would permanently exist with magnets. The only electrical power input is that from the separately excited armature windings.

The efficiency is improved by such a method as the total flux is provided with a reduced electrical input power. It is not a simple replacement theory as the magnet sections have a permeance different to the steel of the back-iron. Additional magnet ampere turns are required to drive flux through the motor as the total reluctance of an equivalent magnetic circuit would be higher than that of a Flux Switching Motor topology (based on a comparison between a FSM and the same lamination shape but with permanent magnet insertion replacing relevant back iron sections).

It is clear from the investigation into magnet placement and orientation (Appendix H) that the design shown in *Figure 6.7* has simple steel stator laminations, has the shorter and simpler magnetic flux paths and the magnet space has greater diversity in terms of dimensions for modelling.

6.4 MAGNETS AND MAGNET SUPPLIERS

A permanent magnet is an energy storage device. This is because energy has to be put into the material when it is first magnetised. If the composition and design of the magnet is correct then the magnetisation and hence energy could remain constant indefinitely. Magnets vary from magnetic steel alloys to ceramic ferrites to rare-earth alloys. To avoid irreversible demagnetisation during use the permanent magnets require a high coercivity. The coercive force is the negative field required to reduce the magnetic flux to zero after saturation. A high coercivity is always preferable.

Anisotropic materials have a preferred direction of magnetisation that cannot normally be changed after the material has been manufactured. Anisotropic magnets have a considerably higher energy product when compared to the isotropic version of the same material. Isotropic materials have no preferred direction of orientation and could therefore be magnetised in any direction.

The Curie temperature is the temperature when the material ceased to have any ferromagnetic properties.

TABLE 6.1 – MAGNET EFFECTIVENESS WITHIN A COMMON LAMINATION AT FIXED ALIGNED ROTOR POSITION (SHOWING MAXIMUM FLUX DENSITY IN BACK-IRON BEHIND ARMATURE SLOT.

Magnet type	Advantages	Disadvantages
Ferrite (Fer1, Fer2, Fer3)	Operate up to 250 degrees C	Brittle
	Low cost	
Cast Alcomax		Low coercive force, needs to be magnetised after assembly as poor handling and exposure to magnetic fields reduces performance,
		Length must be 5 times diameter
NeodiniumIronBoron [NdFeB] (N33, N35, N38, N40, N44, N48)	Highest energy per unit volume of any material	Poor temperature coefficients (lose 0.1% of remenance per degree C when open circuit)
	Offers 30-40% higher energy levels than SmCo	
	Lower cost than SmCo	Best to operate at below 80 degrees C
	Expensive	Brittle (high attractive forces require handling safety procedures)
Plastic Bonded NeodiniumIronBoron [NdFeB]	340 degree C Curie temperature	Maximum operating temperature 120 degrees C
	Expensive	Rather poor temperature coefficient
		Isotropic
Samarium Cobalt (SmCo1:5, SmCo2:17)	Almost impossible to demagnetise	Extremely brittle (take great care when handling)
	Operate with high field over 100 degrees C (which is near copper insulation temperature limit)	

TABLE 6.2 – MAGNET EFFECTIVENESS WITHIN A COMMON LAMINATION AT FIXED ALIGNED ROTOR POSITION (SHOWING MAXIMUM FLUX DENSITY IN BACK-IRON BEHIND ARMATURE SLOT.

Magnet Type	Maximum Magnetic Flux Density,B (T)
NdFeB - N38	0.653
NdFeB - N35	0.6525
NdFeB - N40	0.652
NdFeB - N44	0.651
NdFeB - N33	0.65
NdFeB - N48	0.645
SmCo - 2:17	0.64
SmCo - 1:5	0.565
Alcomax	0.432
Plastic Bonded NdFeB	0.376
Fer2	0.162
Fer3	0.149
Fer1	0.0406

The magnet supplier approached was Magnet Sales [107]. Others were contacted so as to get a range of magnet types but many refused to reply. Magnet data for the range in [107] is shown in Table 6.1.

An IEE presentation into permanent magnets [108] suggests that the cost of rare earth magnets may fall soon as one of the major producers, China, is soon to lose its royalties on their production, leading to an increase in competition.

6.4.1 MAGNET SELECTION AND PRICING

B-H curves were replicated for each magnet type, example B-H curves being found in Appendix H. For a given lamination shape at a fixed rotor position (*Figure 6.8*), each material was used in turn as the magnet type and the peak magnetic flux density was measured in a fixed cross section of back-iron. The 'effectiveness' of each was then tabulated as shown in Table 6.2. The table shows, as would be predicted, that rare-earth magnets are the most effective with the ferrites the least. Ferrites could still be used by altering the lamination geometry to offer sufficient peak air gap flux.

The decision was made to attempt to utilise Fer3 (ferrite) as the magnet for the motor, particularly on the grounds of lower cost. Fer2 could not be used as first hoped due to the fact that it was only sold in preformed blocks whereas Fer3 could be cut to size (to 0.1mm accuracy).

For a quote of 400,000 blocks of Fer3 magnet a year, each block would cost 15 pence each (60p per motor) (120p per motor N38 magnet). One off cost was 2500p per motor for both Fer3 and N38 magnets. From Table 6.2 it was noted that N38 gave over 4 times the flux as Fer3 but the cost was twice that of Fer3 per motor. The cost per Tesla is better with N38 but Fer3 was chosen because the need to keep magnet cost low prevails in designing a lower cost machine.

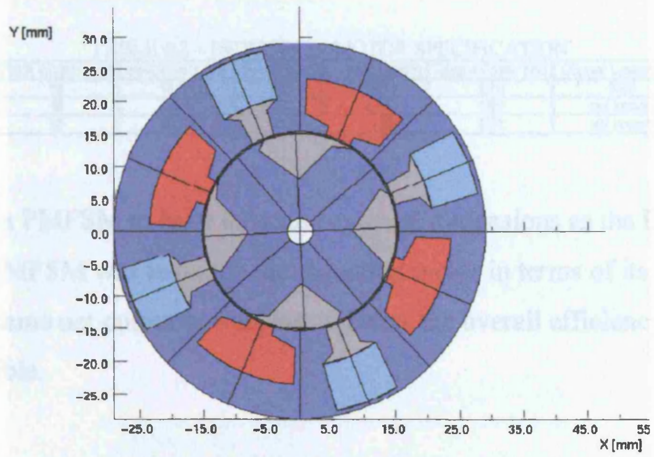


Figure 6.8 :- Lamination used for determining magnet effectiveness.

6.5 SPECIFICATION

6.5.1 INDUCTION MOTOR SPECIFICATION

Small induction motors have low efficiencies due to high field magnetising losses. Iron losses are considerable and extra ampere turns are required to generate the required flux in the motor. The associated copper losses are a significant factor in motor design. Poor efficiency motors such as the shaded pole induction motor are prone to high temperatures due to these factors. The shaded poles (copper rings) actually had marks where heat has affected the steel (*Figure 6.9*).

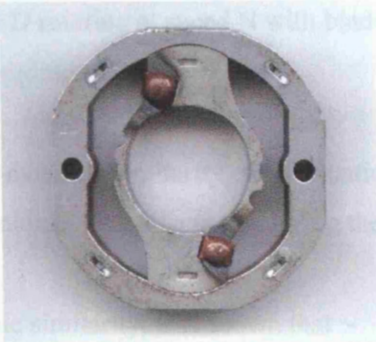


Figure 6.9 :- Shaded pole motor steel laminations.

Domus Ventilation uses induction motors to drive their fans. The Domus S1R 100mm Axial Wall Fan uses a shaded pole induction motor by GI-EM. The motor has 16mm stack length and has a diameter of 58mm. The induction motor was not capable of driving the hysteresis brake, even at no load. This implied that the torque from the fan is very low since the brake was meant for low-torque applications (hence the mechanical output power was small). From a data sheet from a similar motor, the torque was predicted to be 6.41mNm at the load speed of 2520rpm. The data from that sheet matched very closely the actual data obtained through testing. Data for the Domus motor and other 16mm stack length induction motors is shown in Table 6.3.

TABLE 6.3 – INDUCTION MOTOR SPECIFICATION

	Stack length (mm)	Input Power (W)	Speed (rpm)	Torque (mNm)	Output Power (W)	Copper Losses (W)	Other losses (W)	Efficiency (%)
Domus S1R Induction Motor (GI-EM)	16	15.98	2520	6.41	1.70	5.89	8.37	10.6
CCL induction motor	16	16.1	2400	6.41	1.61	not known	not known	10.0
SR16-HO2A	16	20.46	1950	10.85	2.21	not known	not known	10.8

The aim was to design a PMFSM to have the same external dimensions as the Domus S1R 100mm Axial Wall Fan. The PMFSM was to match the induction motor in terms of its load condition (same rpm and torque hence same net output power) but to better the overall efficiency by using as few cheap magnets as possible.

Standard stack lengths used in small motor designs are 10, 12, 16, 20, 25 and 30 mm but any value between 10 and 30 is possible. Many of the motors used in small fans are 16mm in stack length. The Domus fan motor has two screws holding the stack in place and, between the two screws, the stacks actually splay out. The largest measurement for stack length was 17.5mm for one such motor and, due to this the reason, a 17.5mm stack length was used in the PMFSM.

6.5.2 AIR FLOW RATES

The fan impellar design and the speed of the rotation of the fan determin the air flow for the Domus S1R fan (given constant duct resistance). Equations exist defining the fan power, pressure generated and throughflow (6.1, 6.2a, 6.2b) [88, 89]. P is the shaft power, Q is the throughflow of air of density

ϕ, viscosity μ, for a fan of diameter D rotating at speed N with blades having representative characteristic dimensions, a, and roughness, k.

Under conditions of geometric and dynamic similarity, such equations were used to define a range of fan characteristics for the same machine. The effects of varying the fan speed were thus highlighted.

For identical geometric and dynamic similarity, it is shown that :-

$$(\Delta p / N^2 D^2 \rho)_1 = (\Delta p / N^2 D^2 \rho)_2, \quad (Q / ND^3)_1 = (Q / ND^3)_2 \tag{6.1}$$

For a constant fan diameter and constant density it is shown that :-

$$(\Delta p / N^2)_1 = (\Delta p / N^2)_2, \quad (Q / N)_1 = (Q / N)_2 \tag{6.2a}$$

$$(Q_1 / Q_2) = (N_1 / N_2) \tag{6.2b}$$

Thus it was assumed that as the fan speed increased, the flow of air increased proportionally (6.2b) (the power required would increase in ratio with the square of speeds).

6.5.3 MINIMUM PMFSM SPECIFICATION

A PMFSM was to be built to directly replace the induction motor that is found in the Domus S1R 100mm Axial Wall Fan. The minimum required specification was to match the output power at the rated load speed of the Domus S1R induction motor with an input power below that of the Domus motor. The minimum specification is shown in Table 6.4. It did not matter whether the PMFSM also operated at a range of speeds above and/or below the required rated speed. The primary requirement was to meet the criteria specified in Table 6.4. The motor was to be modelled using static and dynamic analysis.

TABLE 6.4 – MINIMUM PMFSM SPECIFICATION

	Basic specification
Stack length (mm)	17.5
Outer Diameter (mm)	58
Shaft diameter (mm)	4
Load speed (rpm)	2500
Load torque (mNm)	6.41
Output Power (W)	1.7

6.6 ELECTROMAGNETIC DESIGN

The Permanent Magnet Flux Switching Motor was designed using a combination of both FEA and Excel spreadsheet. The FEA gave data for the magnetic flux with respect to rotor angle. The spreadsheet used data obtained from FEA to aid in prediction of motor performance. Such data included flux at different rotor angles that was used to calculate armature reluctance. For an 8/4 lamination it was possible to use the rotor angle range of 0 to 45 degrees (in 4.5 degree intervals) then extrapolate the remaining 45 degrees from the data obtained. This worked only if the flux data obtained from FEA was accurate.

6.6.1 STATIC MODEL - FINITE ELEMENT ANALYSIS

A Finite Element Analysis (FEA) package called Opera (by Vector Fields) was used to predict the magnetic flux paths that existed within a PMFSM design (due to permanent magnets and due to armature currents in energised phase windings). Finite Element Analysis allowed non-linear electromagnetic equations for various excitation states and lamination geometries to be solved. The machine was parameterised for creating and solving the electromagnetics of differing lamination geometries. Finite Element Analysis allowed the size, shape and material components of the motor to be defined; the B-H curves of each material used within the motor was definable. The number of copper conductors and the current carried by them or a current density for the copper area was defined. To model the magnetics of the motor, the laminations including air gaps were split into small triangular sections ('elements') of one material type only (air, a defined laminated steel, a defined permanent magnet type, etc). All the 'elements' were connected together and electromagnetic equations were solved for each 'element' that were then combined in such a manner by the FEA package to give a complete solution for the motor. The more elements that were used, especially in zones of higher magnetic saturation, the greater the potential accuracy of the model, but at the cost of longer time to compute the results. The FEA was used to predict the effectiveness of magnets, winding numbers, lamination shapes, variations in size, etc, provided all necessary definitions were given to correctly define the motor. Data from the FEA was used in a spreadsheet for further calculations, such as inductance calculations. The effects of varying the size and types of magnets within the PMFSM was ascertained by the FEA plotting the magnitudes and directions of the flux paths produced by the magnets. A Permanent Magnet Flux Switching Motor was designed with the help of the Opera2D FEA package. The FEA process could be computationally intensive but Genetic Algorithm development could aid this problem.

The magnets had flux path orientations that had to be defined as having internal flux paths which were parallel to the tangent of the outer back-iron. The orientation of the plane of flux in each magnet had to be defined in such a manner as to replicate the flux paths that would have been produced by the field winding in a Flux Switching Motor. Opera2D did have some B-H curves for permanent magnets but other B-H curves from Magnet Sales [107] were used.

One disadvantage of this analysis was that it did not account for the effects of eddy currents and hysterisis losses. Iron losses are the most difficult calculations to make. To clarify the range of iron losses possible, iron loss data was obtained from similarly sized motors. The Domus S1R induction motor has iron and windage losses of 8.37W. It is a suitable hypothesis for the PMFSM to say that the iron and windage losses would be in the region of 2-4W as a good first approximation based on an assumption that the PMFSM would be a better design than the Domus S1R induction motor.

6.6.2 STATIC MODEL - ARMATURE RELUCTANCE CALCULATIONS

The variation of armature inductance with rotor position was calculated for a dynamic simulation of the motor. This was found in FEA by demagnetising the magnet while retaining its permeance. FEA was performed over a 90° rotor range with a known number of armature turns at 0.1A and 0.2A (the choice of currents was such as to avoid saturation in the motor). The rate of change of flux with current at each angle position gave the inductance value per turn (6.3). A plot of reluctance against rotor angle was then achievable for the given lamination (*Figure 6.10*).

$$\text{1) Inductance, } L = N\left(\frac{d\phi}{di}\right) \qquad \textit{and also}$$

$$\text{2) Reluctance, } R = \frac{mmf\_of\_one\_coil}{flux\_of\_that\_coil} \qquad \textit{or} \qquad (6.3)$$

$$\text{Reluctance, } R = \frac{N^2}{L}$$

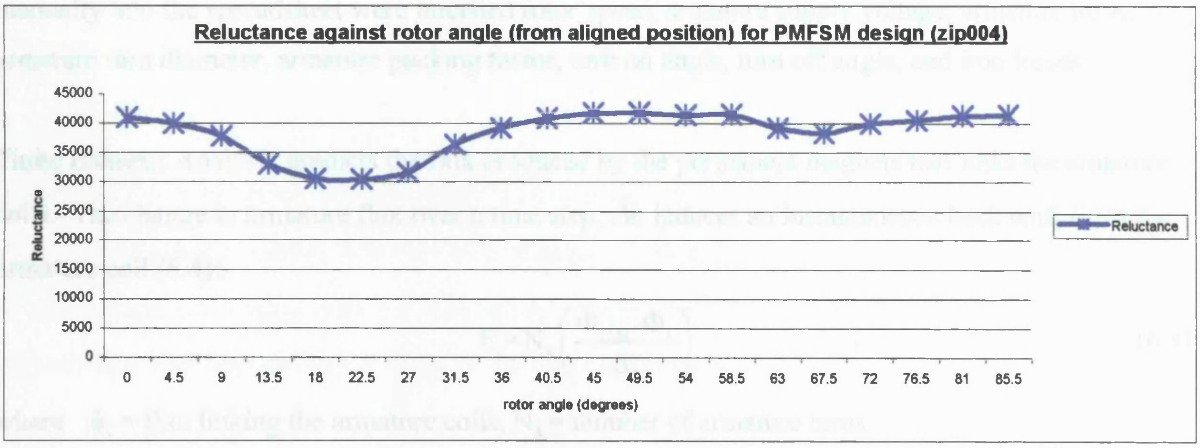


Figure 6.10 :- Reluctance against rotor angle for a given lamination.

6.6.3 DYNAMIC MODELLING OF THE PMFSM

The FEA provided static data (data at fixed rotor position) that was used to calculate the dynamic performance of the motor. Such data included rotor position (at which the calculations were performed to give the data), armature flux, armature mmf and output torque and armature slot area. The spreadsheet that utilised the data was a modified version of one that was used to calculate the dynamic performance of Flux Switching Motors. The Microsoft Excel spreadsheet took the static data and predicted running performance of the motor.

The spreadsheet allowed factors such as stack length, number of turns, wire diameter, packing factor, speed, supply voltage, armature turn on/off angles, armature windings temperature, mean length per armature turn, etc to be variable. By using these variables the performance of a motor design was predicted with a degree of known certainty (limited by the number of assumptions made). The only disadvantage was that the eddy current and iron losses could not be predicted so an estimate had to be included (2-4W when combined although 10% of the total input power was assumable at higher input powers). The workings of the spreadsheet are discussed further in Appendix H.

The performance data was extensive. Among information obtained was –

Armature current	Armature Copper losses	Supply Current
RMS armature mmf	Torque	Net Mechanical Output Power
Total Power Input	Efficiency	

The spreadsheet predicted the back-emf, current and power waveforms. The spreadsheet was used to give the same mechanical output power as the induction motor but at a greater efficiency. The spreadsheet manipulated the data to provide information about the motor. Parameters that were fed manually into the spreadsheet were intended rotor speed, armature supply voltage, armature turns, armature turn diameter, armature packing factor, turn on angle, turn off angle, and iron losses.

Finite Element Analysis predicts the flux produced by the permanent magnets that links the armature coils. The change in armature flux over a time step, Δt, induces an instantaneous back emf, E_t, in the armature coil (6.4):-

$$E_t = N_a \left(\frac{\Phi_{t+\Delta t} - \Phi_t}{\Delta t} \right) \tag{6.4}$$

where ϕ_t = flux linking the armature coils, N_a = number of armature turns

The polarity of the voltage applied to an armature winding switches between positive and negative as the two armatures are energised alternately (6.5):-

$$V_a = V_s \tag{6.5}$$

where V_s = Supply voltage (can be positive or negative)

$$V_a = V_s$$

where V_s = Supply voltage (can be positive or negative)e current, $I_{(t+\Delta t)}$, is given by (6.6):-

$$I_{(t+\Delta t)}=I_t+[(V_a-I_tR_a)-E_t]\left(\frac{\Delta t(Rel)}{N_a^{\ 2}}\right) \tag{6.6}$$

where I_t previous instantaneous value of armature current

R_a Armature winding resistance

Rel Reluctance of magnetic circuit

The instantaneous output power, P_t , is shown in (6.7):-

$$P_t=I_tE_t \tag{6.7}$$

The mechanical power is the average of these instantaneous values over a commutation cycle.

The effects of eddy currents and hysteresis are added to the spreadsheet as an estimated value. It is added to the input power, P_{in}, as the term iron loss, P_{iron}. The input power is calculated from the available output power, P_{out}, plus the copper losses from the armature windings, and the iron loss estimation (6.8):-

$$P_{in}=P_{out}+(I_a^{\ 2}R_a)+P_{iron} \tag{6.8}$$

where I_a RMS armature current

R_a Armature winding resistance

P_{iron} Iron losses

The average output torque of the machine ,T_{av}, is calculated using (6.9):-

$$T_{av}=\frac{P_{out}}{\omega} \tag{6.9}$$

where ω is the speed of the machine (rads^{-1}) from which Δt is derived. The machine is dynamically simulated at constant speed. The time step of length Δt seconds, is a function of the angular velocity, ω, and the number of time steps per revolution. The number of time steps depend on the displacement angle of the rotor between calculations in the Finite Element Analysis (for static calculations).

6.6.4 PMFSM LAMINATION OPTIMISATION

The starting point of the FEA design is shown in *Figure 6.11.* It had the same rotor as used in a FSM design and had thin magnet section. There was a slot by the magnet between the stator pole teeth to allow the flux paths to enter the pole teeth and into the rotor. This slot had a tapered shape (tapering towards the magnet to prevent flux crossing the slot instead of entering the rotor poles via the stator/rotor air gap. There was no back iron surrounding the outer edge of the magnets as previous work has shown that such sections are both too thin for structural integrity and would provide paths for undesirable flux leakage.

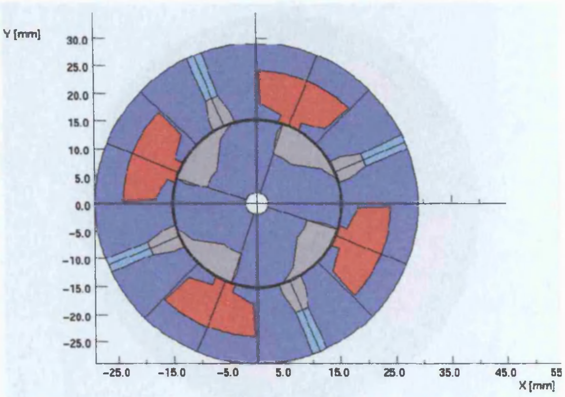


Figure 6.11 :- Lamination used for initial test work.

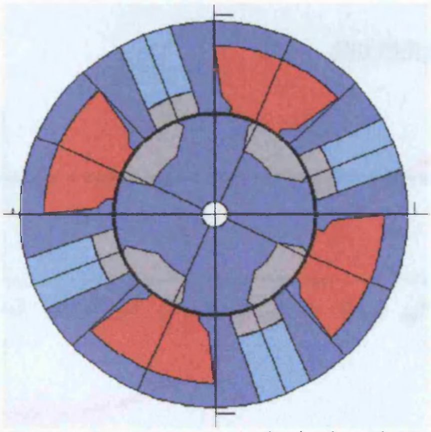


Figure 6.12 :- Improved rotor lamination, Zip000.

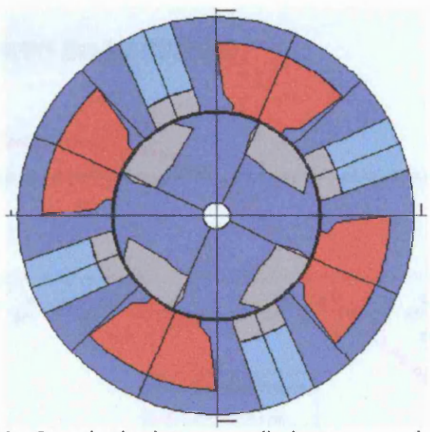


Figure 6.13 :- Investigation into more radical rotor geometries, Zip006.

The rotor was then changed to a style that was known to give a squarer back-emf (*Figure 6.12*). The magnet was also made larger to enable more flux from the magnet. The armature slot was increased as well to allow an increase in wire diameter for a given number of turns. Work was undertaken varying the rotor shapes, stator pole shapes and magnet geometries (*Figure 6.13*). All geometries were considered but with the main aim of getting the best efficiency possible out of the motor through all possible means.

The FEA allowed the flux paths for each lamination selection to be shown in terms of both direction and magnitude. Minimisation of lost flux through improving the lamination design and use of the spreadsheet had suggested that the design shown below in *Figure 6.14* was an optimum (a design that during prototyping was referred to as Zip004 – the fourth design considered). This was based upon the fact that factors such as iron losses were guessed values and, as such, until the motor was built and tested this would remain as an assumption. This was based upon the use of Fer3 as the magnet choice. More expensive rare earth magnets were not used as the gains in efficiency were not justified by the additional costs.

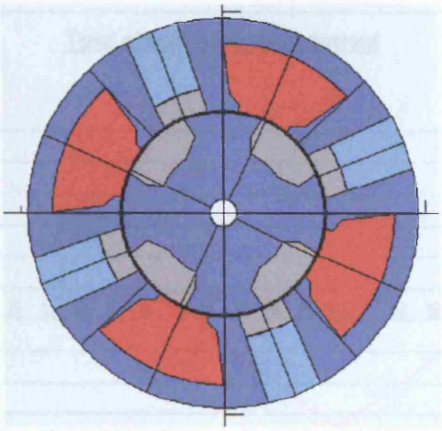


Figure 6.14 :- Optimum lamination, Zip004.

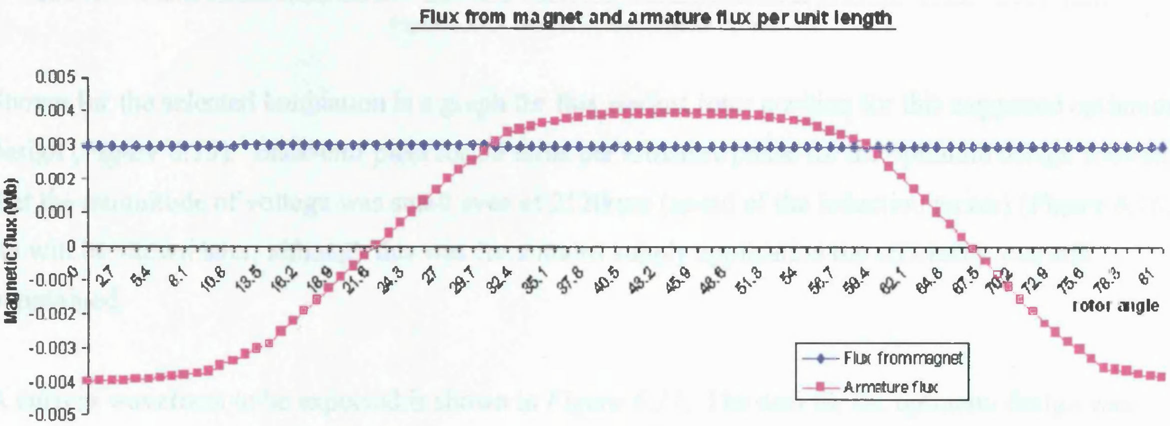


Figure 6.15 :- Field and armature flux with rotor angle.

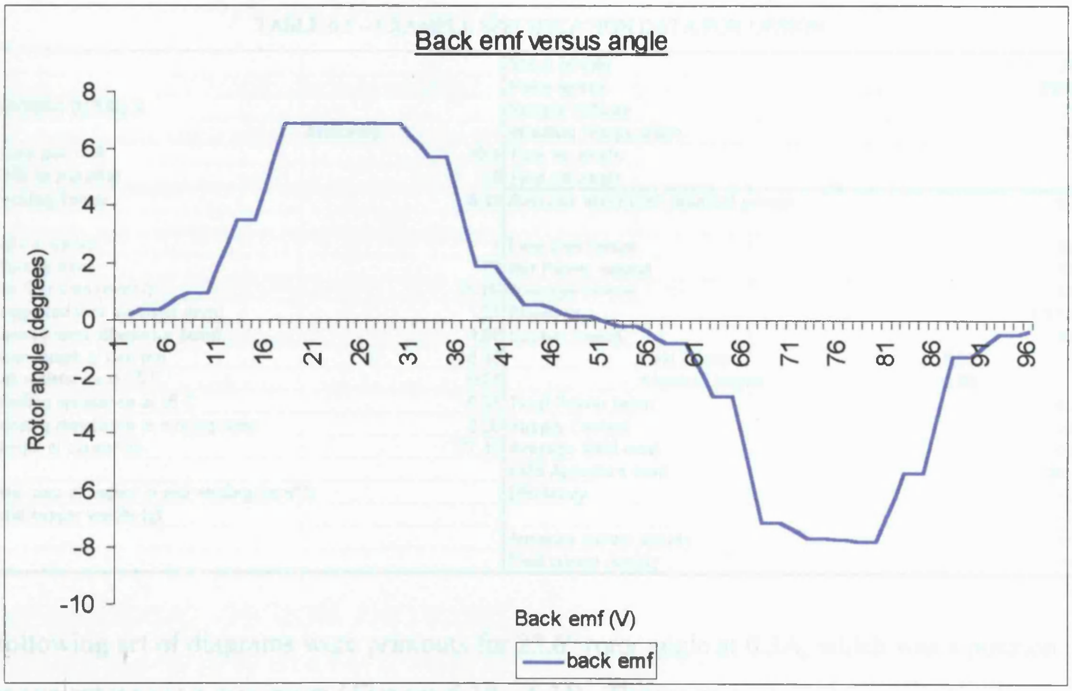


Figure 6.16 :- Back-emf against rotor angle.

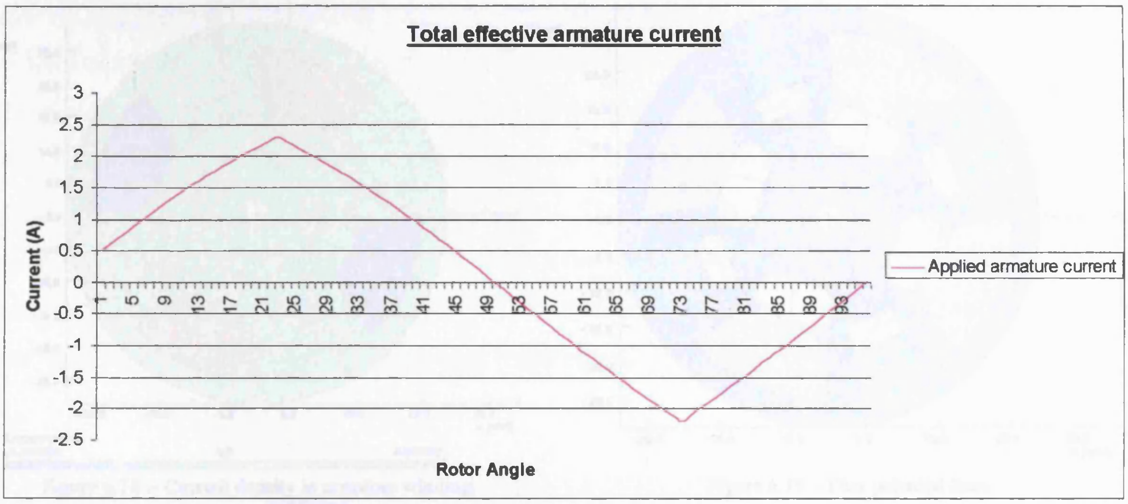


Figure 6.17 :- Total effective armature current.

Shown for the selected lamination is a graph for flux against rotor position for this suggested optimum design (*Figure 6.15*). Back-emf plots for 36 turns per armature phase for the optimum design showed that the magnitude of voltage was small even at 2520rpm (speed of the induction motor) (*Figure 6.16*). As will be shown later, although this was for a mains supply application the efficiency was still maintained.

A current waveform to be expected is shown in *Figure 6.17*. The data for the optimum design was summarised in Table 6.5. The lamination dimensions are shown in Appendix H.

TABLE 6.5 – EXAMPLE SPECIFICATION DATA FOR DESIGN.

		Stack length		**17.5**
		Rotor speed		**2520.0**
WINDING DETAILS		**Supply voltage**		**240**
	Armature	**Winding temperature**		**110**
Turns per coil	**10.0**	**Turn on angle**		**-24**
Coils in parallel	**1.0**	**Turn off angle**		**21**
Packing factor	**0.19**	**Average electromechanical power**		**5.80**
Coils in series	4	**Less iron losses**		**4.00**
Effective turns	72	**Net Power output**		**1.80**
Half Slot area (mm^2)	75.68	**Average torque**		**6.80**
Suggested wire diameter (mm)	1.01	**Pressure**		1701.04
Chosen wire diameter (mm)	**1.00**	**Copper losses**		**0.68**
mean length of turn (m)	0.18	Field losses	0.00	
Coil resistance at 25 C	0.07	Armature losses	0.68	
Winding resistance at 25 C	0.27	**Total Power input**		**6.47**
Winding resistance at running temp	0.36	**Supply Current**		**0.03**
Weight of copper (g)	177.49	**Average field mmf**		**0.00**
		RMS Armature mmf		**98.12**
Total area of copper in end winding (mm^2)		**Efficiency**		**27.75**
Total copper weight (g)				
		Armature current density		1.73
		Field current density		0.00

The following set of diagrams were printouts for 22.6° rotor angle at 0.3A, which was a position where current was at a maximum (*Figures 6.18 – 6.21*). The magnet was not demagnetised at any angle.

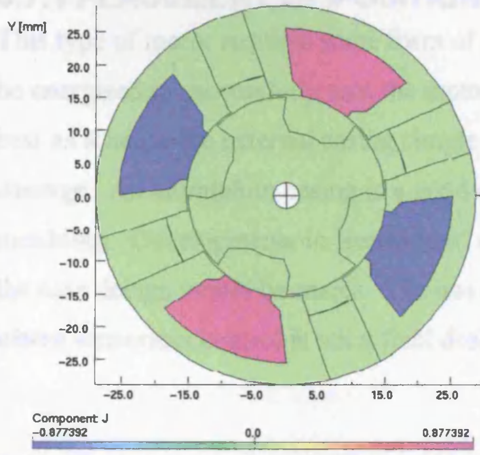


Figure 6.18 :- Current density in armature winding.

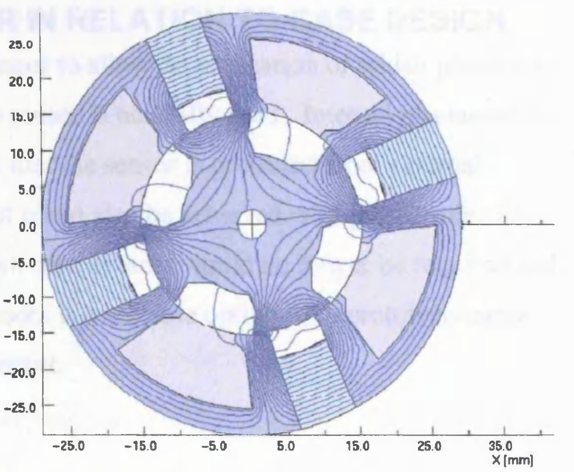


Figure 6.19 :- Flux potential lines.

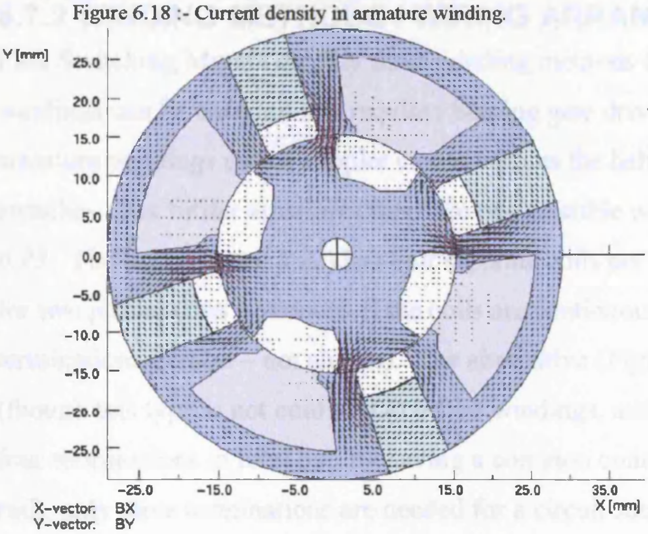


Figure 6.20 :- Flux paths shown with directional arrows.

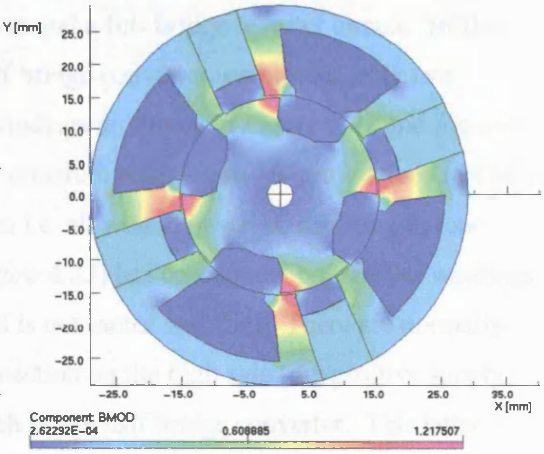


Figure 6.21 :- Flux density plot.

As was shown, the PMFSM, when operating at exactly the same condition as the induction motor and having exactly the same external dimensions gave 27.75% efficiency based on an assumed iron loss of 4W. It only required 6.47W input power compared to 15.96W required to drive the induction motor. Only 9 turns per phase were inserted into the motor due to the compactness of the winding design limiting the packing factor obtainable. It was thus expected that the motor would be more effective with a lower supply voltage and that a pulsed supply would be needed for higher supply voltages as the back emf would not be as large as the supply voltage. This motor was to be referred to as Initial Design.

6.7 MECHANICAL DESIGN AND BUILD

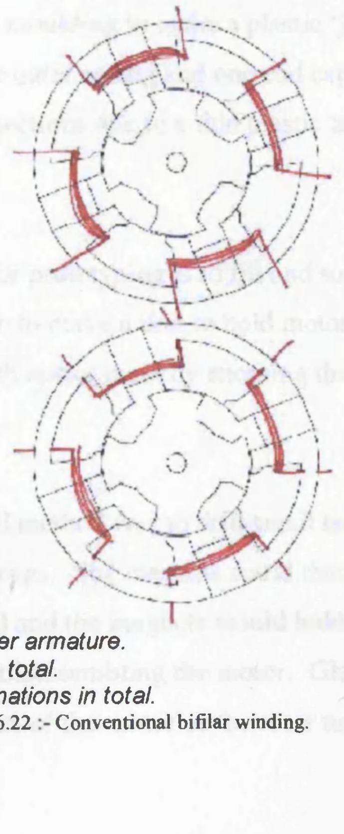
The PMFSM was designed such that it would fit within the fan casing of the Domus S1R 100mm Axial Wall Fan. This posed build issues relating to fitting the motor within the fan housing.

6.7.1 PLACEMENT OF POSITION SENSOR IN RELATION TO CASE DESIGN

This type of motor requires some form of position sensor to allow determination of which phase is to be energised to successfully turn the motor. An opto sensor is normally used. Internal placement is best as it keeps the external casing simple and means that the sensor is protected from external damage. An aluminium casing is a good choice but it could also be achieved with plastic injection moulding. Developments in 'sensorless' control mean opto sensors would no longer be required and the case design would be easier. The use of opto sensors is a sensible option for a prototype motor where sensorless control is not a final design requirement.

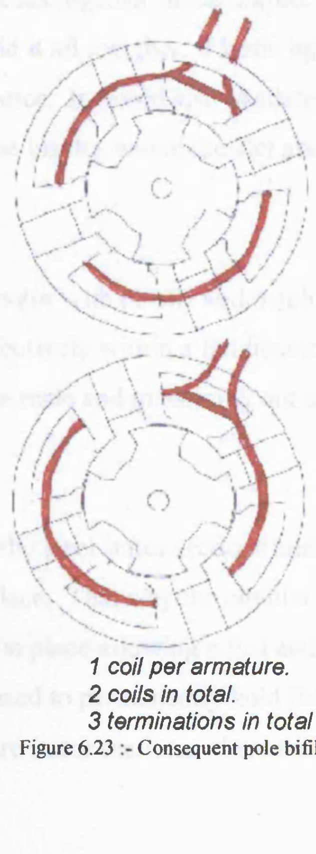
6.7.2 WINDING METHODS / WIRING ARRANGEMENT

Flux Switching Motors employ three winding methods for armature windings. Non-bifilar armature windings can be used but this requires floating gate drives and a full-bridge inverter circuit. Bifilar armature windings uses a simpler circuit such as the half bridge converter, employing only two switches. For bifilar armatures there are two possible windings as shown in *Figure 6.22* and *Figure 6.23*. The first (*Figure 6.22*) has four separate coils per armature resulting in 16 terminations in total for two phases (this is reduced if the coils are continuous i.e. all wound in series, resulting in four terminations in total – not shown). The alternative (*Figure 6.23*) has consequent pole bifilar windings (though this type is not confined to bifilar windings, and is not motor specific). There are normally four terminations in total but, by having a common connection on the high side (the positive supply rail), only three terminations are needed for a circuit such as the half bridge converter. This latter winding type is the preferred option as it also allows a simple end cap shape to be utilised.



4 coils per armature.
8 coils in total.
16 terminations in total.

Figure 6.22 :- Conventional bifilar winding.



1 coil per armature.
2 coils in total.
3 terminations in total (one common).

Figure 6.23 :- Consequent pole bifilar winding topology.

6.7.3 HOLDING THE MACHINE TOGETHER

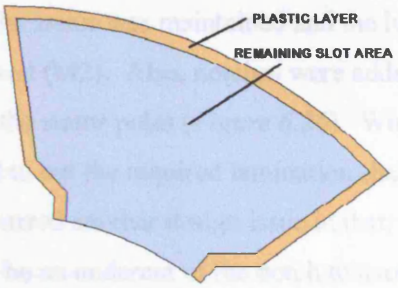


Figure 6.24 :- Packing factor affected by plastic mould injection.

Since the motor is made from four separate steel stator sections with the magnets in between, there has to be a method of holding the entire stator together.

By convention, for motors used in household fans, one end cap forms part of or is attached to the fan housing. There are several possible methods of holding the motor together for such an application.

The simplest is to glue the steel and magnet pieces together with the aid of a former to give the circular guide. If a circular steel block is used to put the pieces around then the flux from the magnets passes through this block and aids holding the pieces together as the glue sets. However, the method would be messy and there is no strict guarantee that the pieces would set to form a perfect circular shape and that there would be a risk of the pieces being glued to the former, temporarily holding it all in place.

Another similar method is to have a former holding the pieces together in the correct place then using plastic injection moulding to make a plastic 'jacket' to hold it all together. Plastic injection moulding would create the outer casing and one end cap for a fan motor. It would also insulate the windings from the stator sections due to a thin plastic layer lining the insides where the slot area is (*Figure 6.24*).

An alternative for prototyping is to fill and surround the motor with plastic and machine out any unwanted plastic to make a unit to hold motor in place effectively within a fan housing. This could also be done with epoxy resin by encasing the motor in the resin and machining out any unwanted resin.

A more practical method was to drill small holes through the steel stator sections and bolt them onto aluminium end caps. The magnets could then be slid in place. That way the circular nature of the motor is ensured and the magnets would hold themselves in place allowing a fast and effective method of building and disassembling the motor. Glue could be used to permanently hold the magnets in place. From FEA of the motor as the rotor turned, there are stator sections where the magnetic flux is small in value which would be best suited for holes to be drilled (*Figure 6.25*). The holes were drilled in a section where the strength of the stator was maintained and the hole was just large enough for a threaded stainless steel rod to be used (M2). Also, notches were added to the stator sections to allow the magnets to be held away from the stator poles (*Figure 6.26*). Wire erosion of steel lamination plates (0.3mm thickness) was used to cut the required lamination shapes (a .DXF file was drawn for the machine to cut from). This incurred another design issue in that, for a flush fit between the magnet and the steel sections, there had to be an undercut of the notch to avoid a bad fit (the wire used in cutting the steel laminations was round so an undercut was needed to prevent a rounded corner point where a 90 degree point was needed).

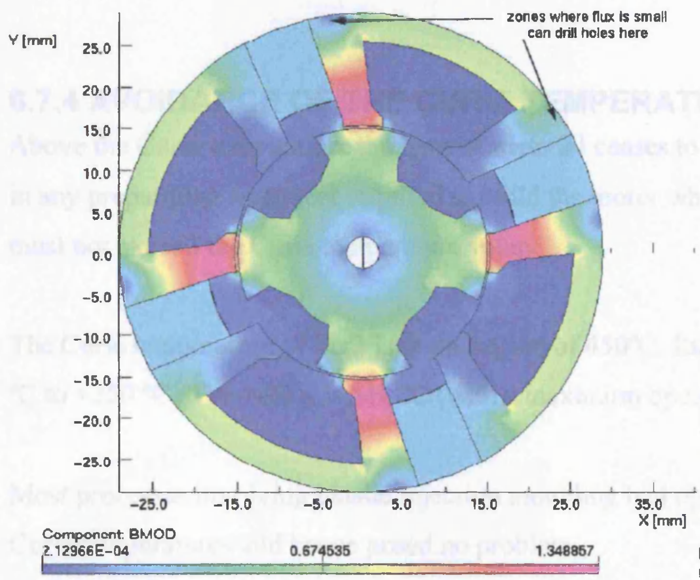


Figure 6.25 :- Low flux stator areas where drilling may be allowed.

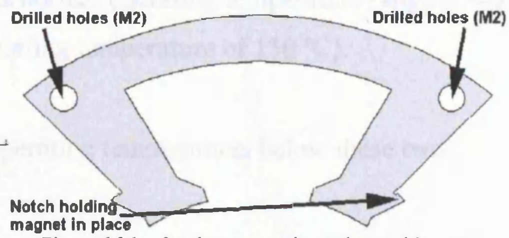


Figure 6.26 :- Steel stator section to be used in motor.

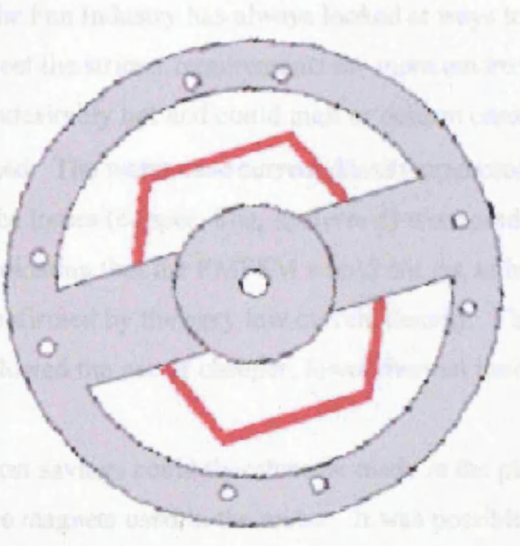


Figure 6.27 :- Possible end cap configuration with winding shown in Figure 6.49

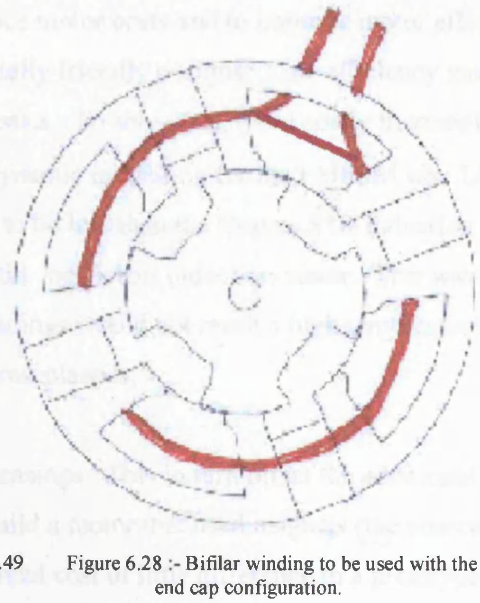


Figure 6.28 :- Bifilar winding to be used with the end cap configuration.

With this approach, one end cap held all the motor in place while the winding was inserted by a method as in *Figure 6.23*. The other end cap was a very simple end plate to ensure the motor is held tightly with the housing of the fan.

Any such method would require the placement of the opto sensor to be taken into account. Aluminium end caps were easily machined to allow space for the opto sensor. Internal placement of the opto sensor allowed a simple external shape (cylindrical) and protected the sensor from external damage. The aluminium end caps (*Figure 6.27*) were designed such that the windings in *Figure 6.28* were used (the end caps at each end were rotated 90 degrees to each other to allow the windings to be inserted without being wrapped around the end caps as well).

6.7.4 AVOIDANCE OF THE CURIE TEMPERATURE

Above the Curie temperature the magnet material ceases to have any ferromagnetic properties. Hence, in any preparation treatment required to build the motor while magnets are present, the temperature must not exceed the Curie temperature value.

The Curie temperature of Fer3 is in the region of 450°C. Its normal operating temperature range is -40 °C to +250 °C. For N38 it is 310 °C (with a maximum operating temperature of 150 °C).

Most processes involving plastic injection moulding had operating temperatures below these two Curie temperatures and hence posed no problem.

6.7.5 THERMOPLASTIC PROPERTIES OF MOTOR CASINGS

The Fan Industry has always looked at ways to reduce motor costs and to improve motor efficiency to meet the stricter requirements for more environmentally friendly designs. Low efficiency motors get undesirably hot and could melt or deform cheap plastics. To solve this, more costly thermoplastics are used. The worst-case current density predicted in dynamic modelling for the PMFSM was $1 A/mm^2$. The losses (copper, iron, hysteresis) were predicted to be less than the Domus S1R Induction Motor, indicating that the PMFSM would not get as hot as the inefficient induction motor. This was confirmed by the very low current density. The windings would not reach a high temperature which allowed the use of cheaper, lower thermal limit, thermoplastics.

Cost savings could therefore be made in the plastic casings. This in turn offset the additional cost of the magnets used in the motor. It was possible to build a motor that used magnets (the source of the additional cost) but have an end product that had a final cost of little difference to a prototype induction motor.

The lamination designs are shown in *Figures 6.29* and *6.30* from which 0.3mm thick laminations were cut by wire erosion (*Figure 6.31*). The magnets were cut to size and the complete stator is shown in *Figure 6.32*.

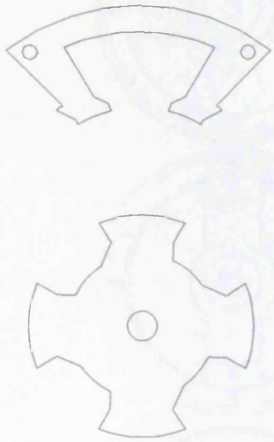


Figure 6.29 :- Proposed laminations to be used.

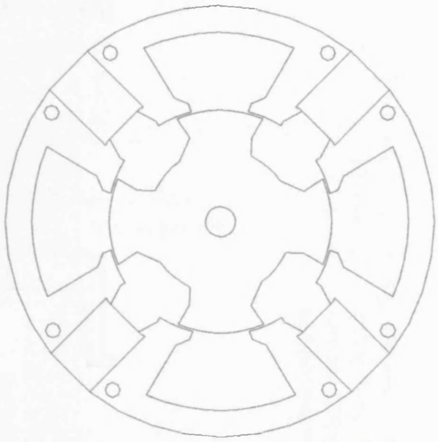


Figure 6.30 :- Proposed final shape of laminations for motor.

Figure 6.31 :- Actual laminations made by wire erosion method.

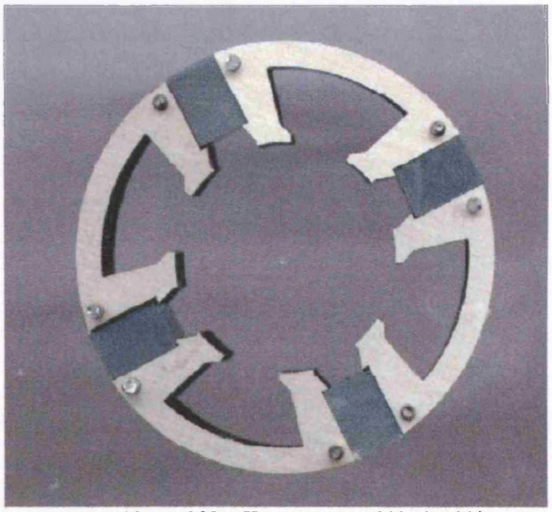


Figure 6.32 :- How stator would look within motor.

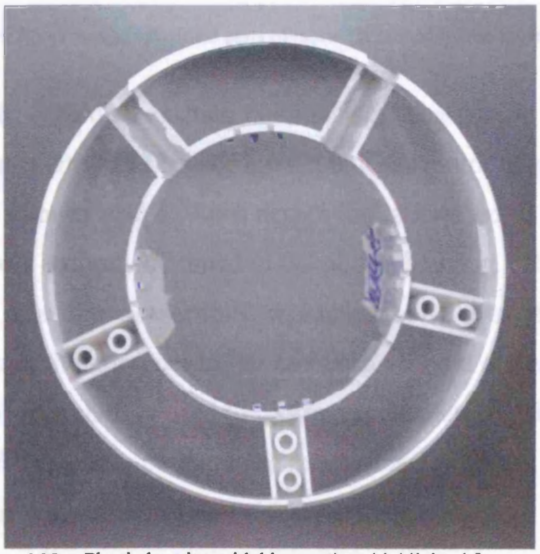


Figure 6.33 :- Plastic housing with blue sections highlighted for removal.

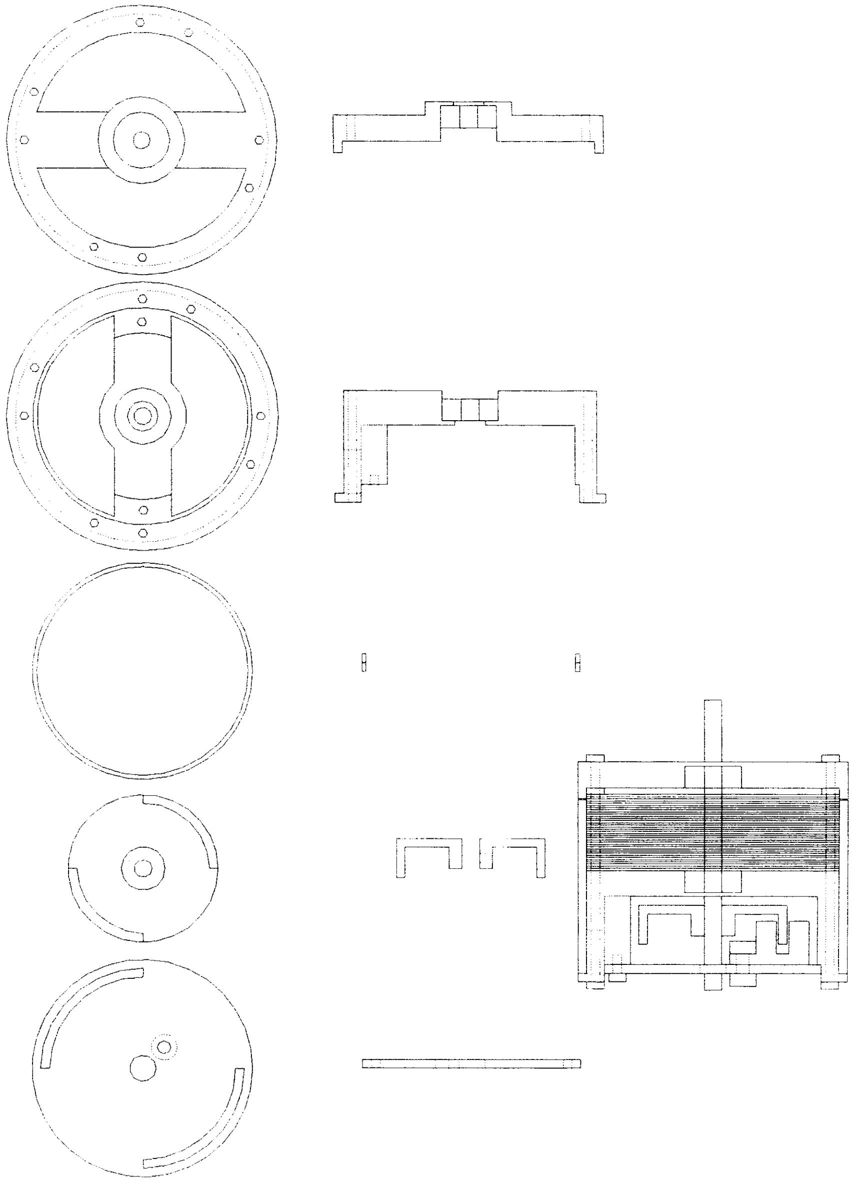


Figure 6.34 :- Plans of individual parts of motor for assembly into fan housing.

The motor was designed to fit within an existing fan housing. This plastic housing required some machining to remove sections that would impair insertion of the motor (*Figure 6.33*) as the original motor was not fully circular in outer shape. Such design limitations led to two end caps being designed so that the motor was securely clamped to the plastic housing. The opto sensor was kept internally within the motor. The opto sensor itself was housed on a printed circuit board disc and this would ideally eventually house the microcontroller and even the gate driver and switches circuitry for the armature windings. The rotor shaft needed to be 4mm in diameter to allow the fan to be attachable to the shaft. The drawings for the extra components required are shown in *Figure 6.34* and the proposed layout of the motor is shown in *Figure 6.35*. Appendix H shows photos of the built motor.

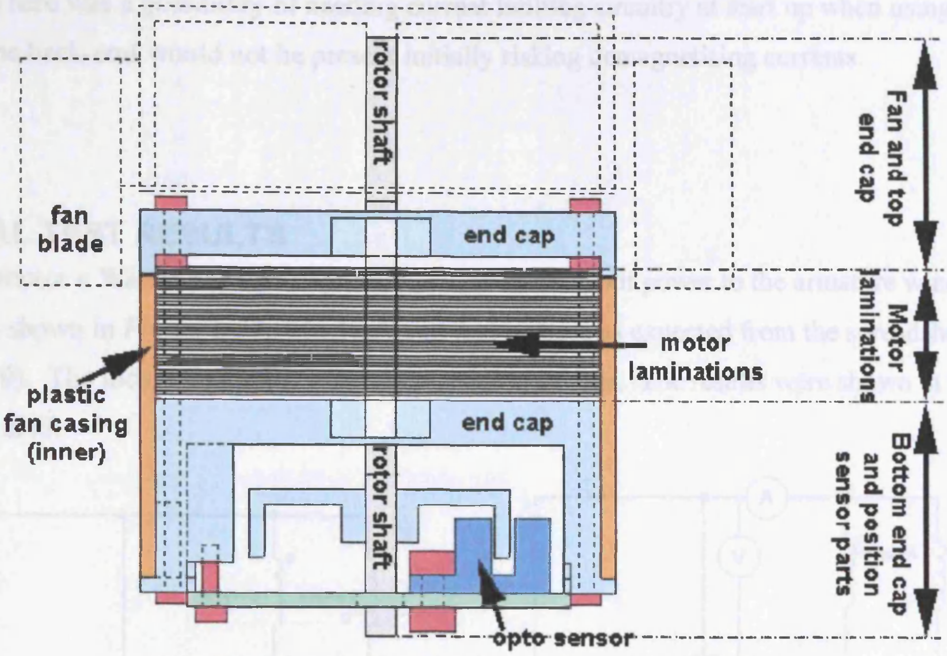


Figure 6.35 :- Plan of motor within plastic housing and with fan attached.

Figure 6.36 :- New bifilar windings from fan end.

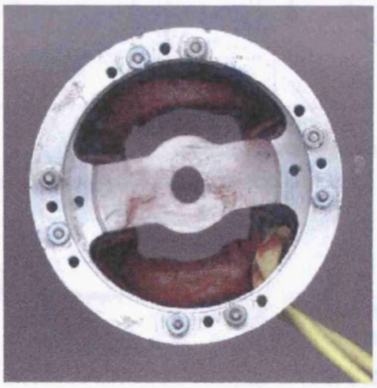


Figure 6.37 :- New bifilar windings from opto sensor end.

Figure 6.36 and *Figure 6.37* show how the bifilar winding arrangement actually appeared. 9 turns of 1mm diameter wire for each phase were used in the first winding of the motor.

The same fan impellar was used in the PMFSM as the load, allowing the comparison of performance with that of the existing induction motor.

6.8 POWER ELECTRONICS CIRCUITRY

The use of a consequent pole bifilar winding topology as in *Figure 6.23* meant that the main power electronic circuitry should be the half bridge converter circuit as shown in *Figure 6.38* (Chapter 3). The bifilar windings acted as transformers during switching between phases, allowing faster transfer of current from the switch that has just turned off. This required the freewheel diodes of the MOSFET switches. There was a possibility of needing current limiting circuitry at start up when using 240V supply as the back-emf would not be present initially risking demagnetising currents.

6.9 INITIAL TEST RESULTS

To test the motor a Watt meter was required to measure the input power to the armature windings and switches as shown in *Figure 6.39*. The back emf had a shape as expected from the spreadsheet (*Figure 6.40*). The motor was tested over a range of fan speeds. The results were shown in Table 6.6 and *Figure 6.41*.

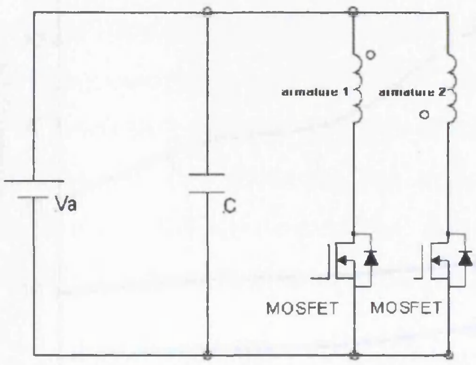


Figure 6.38 :- Half bridge converter circuit with two ground referenced switches.

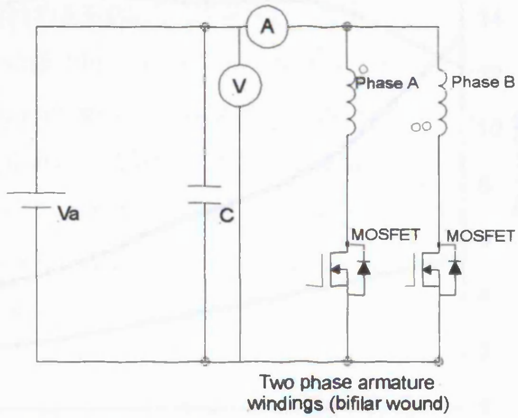


Figure 6.39 :- Watt meter placement in circuit

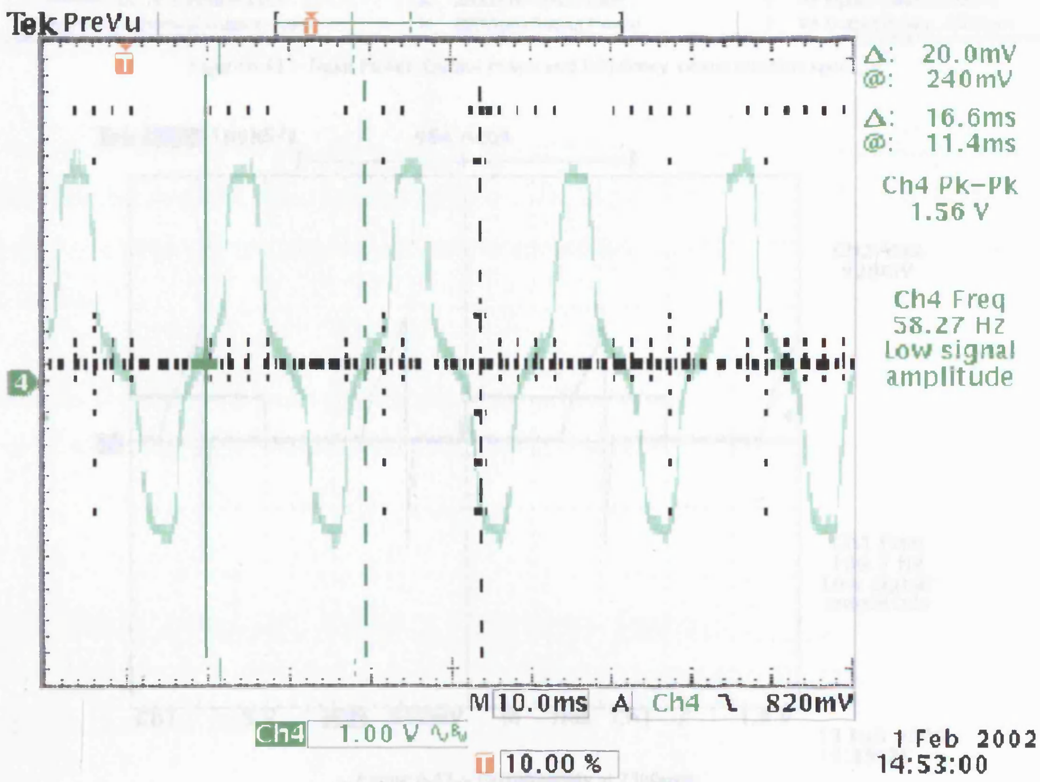


Figure 6.40 :- Back emf waveform at 874rpm.

TABLE 6.6 – TEST RESULTS FROM PMFSM AND INDUCTION MOTOR COMPARISON

Opto speed (Hz)	Rotor speed (rpm)	V rms (V)	I rms (A)	P input (W)	P mech (W)	IIR (W)	other losses (W)	Efficiency (%)
100	**1500**	5.02	2.643	**2.91**	**0.612**	0.68457	1.613425998	**21.03093**
125	**1875**	5.035	2.77	**3.43**	**0.95625**	0.75194	1.7218058	**27.87901**
133.3	**2000**	4.997	2.6	**3.54**	**1.088**	0.66248	1.78952	**30.73446**
166.66	**2500**	5.009	2.59	**4.04**	**1.7**	0.65739	1.6826062	**42.07921**
200	**3000**	5.029	2.4	**4.89**	**2.448**	0.56448	1.87752	**50.06135**
233.3	**3500**	5.06	3.073	**6.83**	**3.332**	0.92545	2.572553758	**48.78477**
266.66	**4000**	5.003	3.341	**8.95**	**4.352**	1.0939	3.504096462	**48.6257**
300	**4500**	5.008	4.511	**13.43**	**5.508**	1.99421	5.927786142	**41.01266**
SP IM	***2520***	-	-	***15.96***	***1.7***	***5.89***	***8.37***	***10.65163***

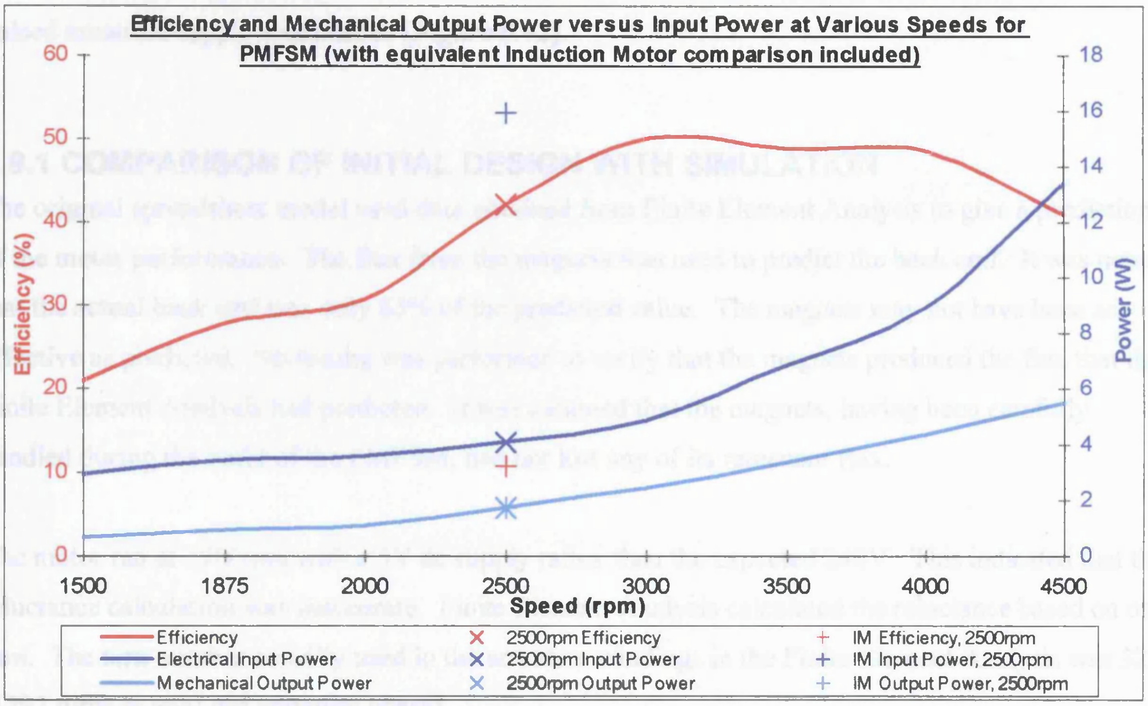


Figure 6.41 :- Input Power, Output Power and Efficiency versus machine speed.

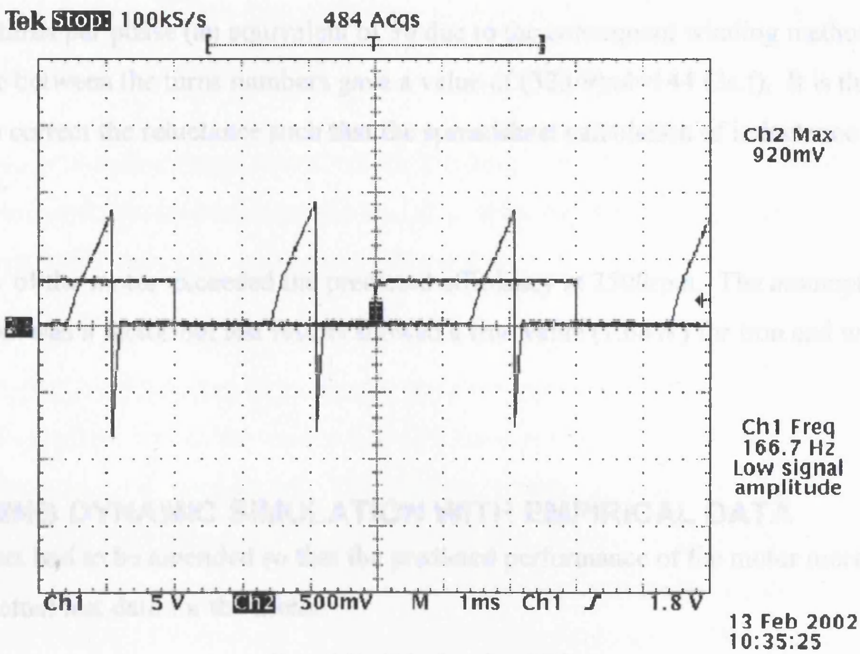


Figure 6.42 :- Pulsed supply at 2500rpm.

The results were very encouraging. The input power was very low across the speed range. The copper losses (IIR in *Figure 6.41*) were very small. The iron and windage losses were also low. This resulted in a very efficient design. 42% efficiency and 4.04W input power was obtained at 2500rpm. It was 3.94 times more efficient than the Domus induction motor as it takes only a quarter of the input power. The motor was more efficient over a wide speed range.

As noted from the Table 6.6, the dc power supply for these speeds was not as high as expected and a pulsed armature supply was needed (*Figure 6.42*).

6.9.1 COMPARISON OF INITIAL DESIGN WITH SIMULATION

The original spreadsheet model used data obtained from Finite Element Analysis to give a prediction of the motor performance. The flux from the magnets was used to predict the back emf. It was noted that the actual back emf was only 65% of the predicted value. The magnets may not have been as effective as predicted. No testing was performed to verify that the magnets produced the flux that the Finite Element Analysis had predicted. It was assumed that the magnets, having been carefully handled during the build of the PMFSM, had not lost any of its remenant flux.

The motor ran at 2500rpm with a 5V dc supply rather than the expected 240V. This indicated that the reluctance calculation was inaccurate. Finite Element Analysis calculated the reluctance based on one turn. The turn number actually used in the armature windings in the Finite Element Analysis was 323 (1292 turns in total per armature phase).

There were 9 turns per phase (an equivalent of 36 due to the consequent winding method chosen). The difference between the turns numbers gave a value of (323/9)x4=144 (3s.f). It is this value that was needed to correct the reluctance such that the spreadsheet calculation of inductance gave the correct results.

The efficiency of the motor exceeded the predicted efficiency at 2500rpm. The assumption of iron losses to be 4W was a factor but test results showed a true value (1.68W) for iron and windage losses.

6.10 REFINING DYNAMIC SIMULATION WITH EMPIRICAL DATA

The spreadsheet had to be amended so that the predicted performance of the motor more closely matched the actual test data for the motor.

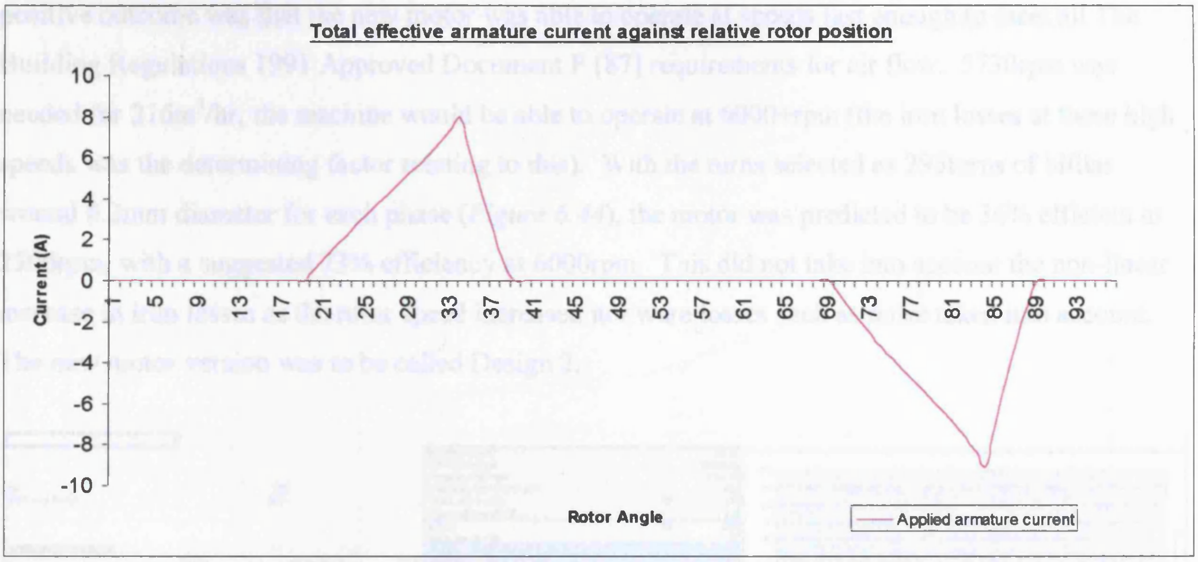


Figure 6.43 :- Example of spreadsheet current waveform to closely match actual test current waveforms.

Pulsed and non-pulsed armature timings were included in the spreadsheet. The inclusion of a pulsed armature supply had not been introduced to this spreadsheet before. The timings of the pulse were definable within the refined model.

Iron and windage losses at 2500rpm had been evaluated and a correction for this as the rotor speed varied was included.

The magnitude of the back emf waveform was amended to be 65% of the original calculation.

The reluctance was corrected by a factor of 144 to allow a more accurate prediction of the current waveforms and hence motor performance.

With the above corrections the test waveform for 2500rpm and a non-pulsed (conventional Flux Switching Motor timing sequence) were successfully replicated with accurate predictions for input power, losses and efficiency (*Figure 6.43* c.f. *Figure 6.2*).

6.10.1 IMPROVED DESIGN FOR HIGHER VOLTAGE

The corrected spreadsheet was then used to remodel the motor for 240V supply. Two coils of 295 turn, 0.2mm diameter wire was selected. This was not as many turns as was really needed for a high supply voltage but a higher turn number reduced the wire diameter to a value that could not be wound for the motor (0.16mm diameter was the lowest size allowable for this size of motor). One major note was that the back emf at 2500rpm was far less than the supply voltage. 240V may not be the best supply voltage for this type of machine at 2500rpm as a pulsed supply would have to be used. The positive outcome was that the new motor was able to operate at speeds fast enough to meet all The Building Regulations 1991 Approved Document F [87] requirements for air flow. 5730rpm was needed for 216m^3/hr, the machine would be able to operate at 6000+rpm (the iron losses at these high speeds was the determining factor relating to this). With the turns selected as 295turns of bifilar wound 0.2mm diameter for each phase (*Figure 6.44*), the motor was predicted to be 36% efficient at 2500rpm, with a suggested 73% efficiency at 6000rpm. This did not take into account the non-linear increase in iron losses as the rotor speed increased nor were losses such as noise taken into account. The new motor version was to be called Design 2.

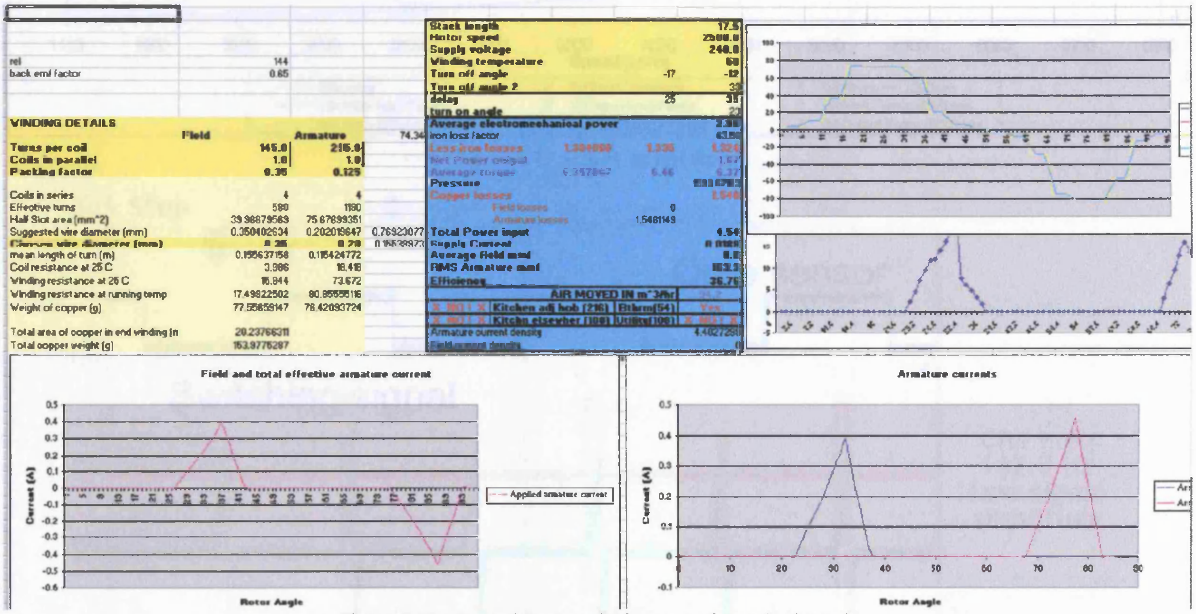


Figure 6.44 :- Spreadsheet model incorporating pulsed supply.

6.11 TESTING OF REVISED DESIGN

6.11.1 REVISED DESIGN TEST RESULTS

Test results at 240V dc supply are shown in Table 6.7, *Figure 6.45*. The test waveform for 2500rpm is shown in *Figure 6.46*.

TABLE 6.7 – 240V DC TEST RESULTS.

Opto speed (Hz)	Rotor speed (rpm)	V0 value	V1 value	IIR (W)	others (W)	V rms (V)	I rms (A)	P input (W)	P mech (W)	Efficiency (%)	mech pow out factor	Air moved m^3/hr	Air moved l/sec	Temp (C) after25mins
74	1110	255	2	0.04	3.39	241.5	0.0352	3.773	0.34	[illegible]	0.20	41.74	11.59	28.9
100	1500	181	2	0.07	4.14	241.6	0.0434	4.815	0.61	12.71	0.36	56.40	15.67	
125	1875	136	2	0.09	4.70	241.4	0.051	5.747	0.96	16.64	0.56	70.50	19.58	
133.3	2000	125	2	0.10	4.91	240.9	0.0528	6.099	1.09	17.83	0.64	75.18	20.88	
166.66	2500	91	2	0.14	5.09	240	0.0635	6.931	1.70	24.53	1.00	94.00	26.11	28.9
200	3000	68	2	0.19	5.80	241.6	0.0735	8.438	2.45	29.01	1.44	112.80	31.33	
233.3	3500	50	2	0.25	6.18	241.2	0.0848	9.765	3.33	34.11	1.96	131.58	36.55	
266.66	4000	33	2	0.32	6.59	241.3	0.0953	11.253	4.35	38.67	2.56	150.40	41.78	
300	4500	14	2	0.50	10.10	241.3	0.12	16.11	5.51	34.19	3.24	169.20	47.00	
333.34	5000	3	2	0.78	12.06	240.7	0.1494	19.64	6.80	34.62	4.00	188.00	52.22	
340	5100	1	2	1.01	13.35	240	0.17	21.43	7.07	33.01	4.16	191.76	53.27	
400	6000	2	3	2.22	21.66	240	0.2524	33.67	9.79	29.08	5.76	225.60	62.67	29.5
414	6210	1	3	2.69	23.88	240	0.2783	37.06	10.49	28.30	6.17	233.50	64.86	
422	6330	1	4	5.54	36.81	240	0.3991	53.25	10.90	20.47	6.41	238.01	66.11	
SP [illegible]	2520	-	-	-	-	-	-	15.92	1.7	10.67839196	Improvement at 2500rpm			2.30

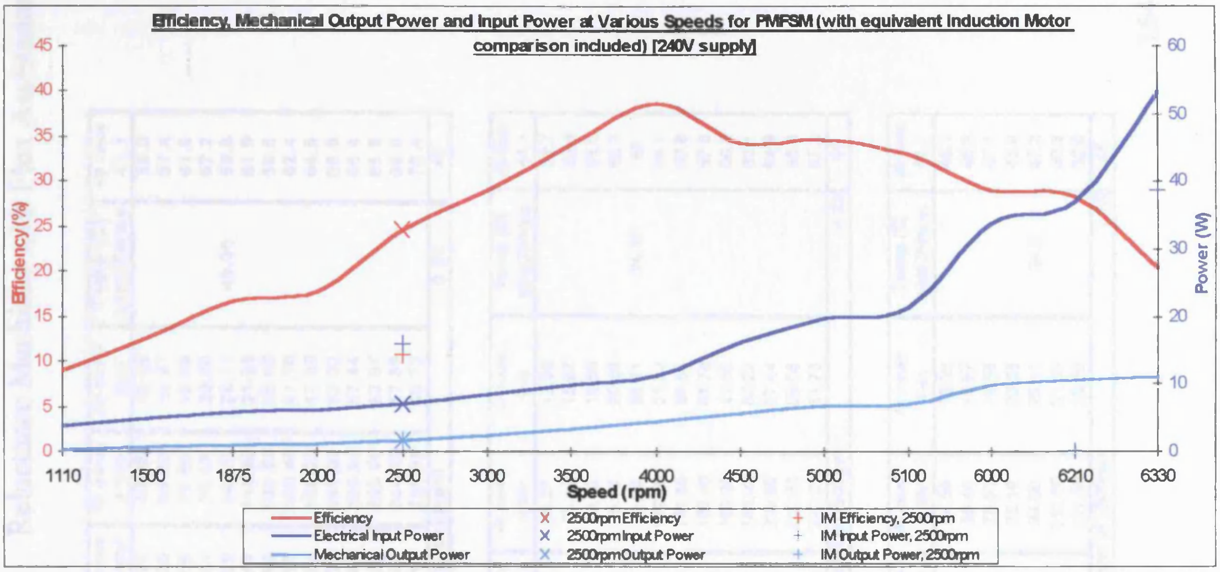


Figure 6.45 :- 240V dc test results.

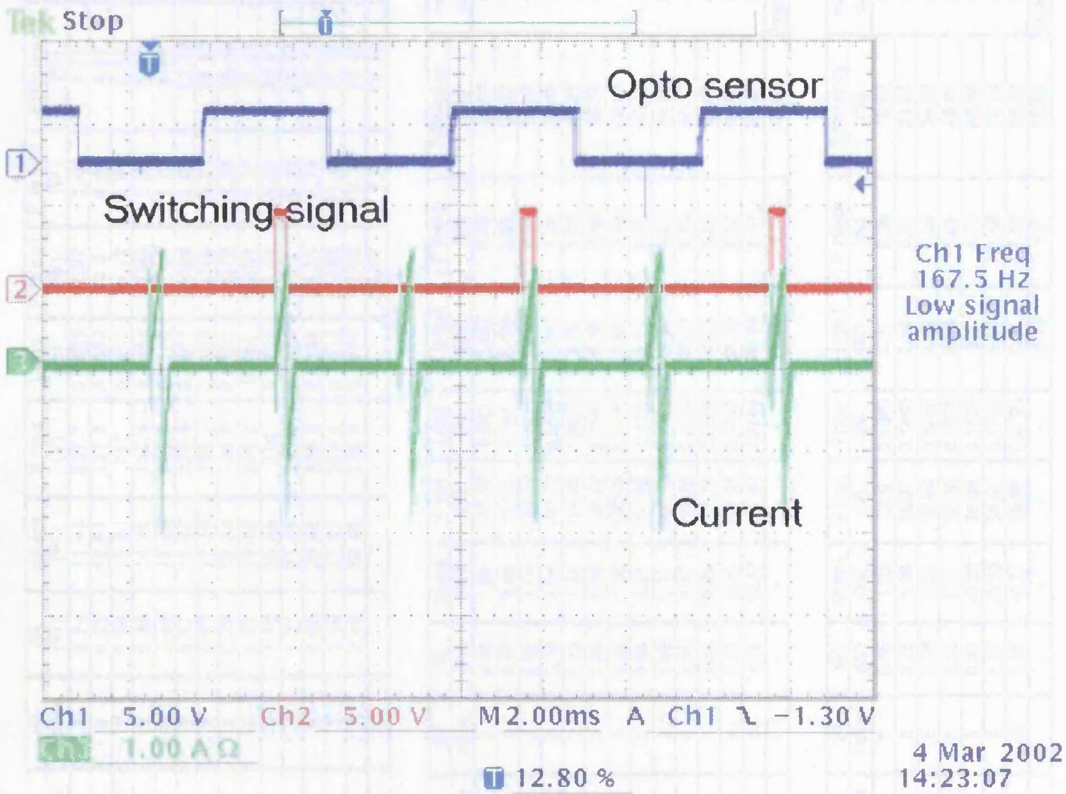


Figure 6.46 :- 2500 rpm waveforms.

Tests were then carried out at 240V ac rms. The results are shown in Table 6.8, *Figure 6.47*. The tests were repeated for 110V ac rms (Table 6.8, *Figure 6.48*) and 40V ac rms (Table 6.8, *Figure 6.49*). 110V was the supply used in the US. The final test was to ascertain the supply voltage required for maximum efficiency at 2500rpm. This was found to be 40V ac rms

TABLE 6.8 – 240V, 110V AND 40V AC TEST RESULTS

Opto speed (Hz)	Rotor speed (rpm)	V0 value	V1 value	IIR (W)	others (W)	V rms (V)	I rms (A)	P input (W)	P mech (W)	Efficiency (%)	mech pow out factor	Air moved m^3/hr	Air moved l/sec	Temp (C) after25mins	dB noise 41.1
81	**1215**	255	2	0.24	11.43	237.7	0.0826	**12.07**	**0.40**	**3.33**	0.24	45.68	12.69		56.9
100	**1500**	200	2	0.28	12.61	237.6	0.0899	**13.5**	**0.61**	**4.53**	0.36	56.40	15.67		57.4
125	**1875**	152	2	0.36	13.81	237	0.1015	**15.12**	**0.96**	**6.32**	0.56	70.50	19.58		61.8
133.3	**2000**	141	2	0.39	14.36	236.6	0.1054	**15.83**	**1.09**	**6.87**	0.64	75.18	20.88		57.2
166.66	**2500**	104	2	0.46	15.49	236.5	0.1155	**17.65**	**1.70**	**9.63**	1.00	94.00	26.11	45.30	59.6
200	**3000**	80	2	0.57	16.67	236.5	0.1277	**19.69**	**2.45**	**12.43**	1.44	112.80	31.33		61.9
233.3	**3500**	62	2	0.67	17.49	236.6	0.1386	**21.49**	**3.33**	**15.50**	1.96	131.58	36.55		59.8
266.66	**4000**	48	2	0.79	18.64	236.6	0.1503	**23.78**	**4.35**	**18.30**	2.56	150.40	41.78		62.4
300	**4500**	37	2	0.92	19.48	236.2	0.163	**25.91**	**5.51**	**21.26**	3.24	169.20	47.00		64.6
333.34	**5000**	27	2	1.10	20.66	236	0.1777	**28.56**	**6.80**	**23.81**	4.00	188.00	52.22		66.6
366.66	**5500**	18	2	1.31	21.66	235.6	0.1942	**31.2**	**8.23**	**26.37**	4.84	206.80	57.44		66.4
400	**6000**	9	2	1.66	23.86	235.9	0.2184	**35.31**	**9.79**	**27.73**	5.76	225.60	62.67		66.8
433.33	**6500**	3	2	2.19	27.97	235.9	0.2512	**41.66**	**11.49**	**27.58**	6.76	244.40	67.89		69.8
445	**6675**	1	2	2.45	29.45	235.6	0.2654	**44.02**	**12.12**	**27.53**	7.13	250.98	69.72		70.4
SP IM	***2520***	-	-	-	-	-	-	***15.92***	***1.7***	***10.68***	% improvement			0.90	45

Opto speed (Hz)	Rotor speed (rpm)	V0 value	V1 value	IIR (W)	others (W)	V rms (V)	I rms (A)	P input (W)	P mech (W)	Efficiency (%)	mech pow out factor	Air moved m^3/hr	Air moved l/sec	Temp (C) after25mins	dB noise 41.1
78.5	**1178**	255	2	0.06	1.99	107.99	0.042	**2.427**	**0.38**	**15.54**	0.22	44.27	12.30		52.7
100	**1500**	190	2	0.07	2.05	107.9	0.045	**2.736**	**0.61**	**22.37**	0.36	56.40	15.67		53.3
125	**1875**	140	2	0.09	2.03	107.72	0.0511	**3.078**	**0.96**	**31.07**	0.56	70.50	19.58		54.5
133.3	**2000**	126	2	0.10	2.06	107.72	0.0528	**3.247**	**1.09**	**33.49**	0.64	75.18	20.88		52.3
166.66	**2500**	55	2	0.13	1.93	107.52	0.0617	**3.76**	**1.70**	**45.21**	1.00	94.00	26.11	24.30	52
200	**3000**	72	3	0.39	4.08	108.03	0.1064	**6.92**	**2.45**	**35.38**	1.44	112.80	31.33		56.1
233.3	**3500**	50	3	0.50	4.10	108.05	0.1194	**7.93**	**3.33**	**42.01**	1.96	131.58	36.55		57.8
266.66	**4000**	23	3	0.64	4.21	107.95	0.136	**9.21**	**4.35**	**47.25**	2.56	150.40	41.78		57.5
300	**4500**	9	3	1.08	5.77	107.93	0.1766	**12.36**	**5.51**	**44.56**	3.24	169.20	47.00		60.2
333.34	**5000**	12	4	1.80	7.47	107.53	0.2274	**16.07**	**6.80**	**42.32**	4.00	188.00	52.22		63.1
366.66	**5500**	3	4	2.84	9.29	107.26	0.2856	**20.36**	**8.23**	**40.41**	4.84	206.80	57.44		64.9
377	**5655**	1	4	3.23	9.80	107.24	0.3046	**21.73**	**8.70**	**40.03**	5.12	212.63	59.06		65.6
394	**5910**	1	5	4.88	12.57	107.44	0.3744	**26.96**	**9.50**	**35.25**	5.59	222.22	61.73		67.2
SP IM	***2520***	-	-	-	-	-	-	***15.92***	***1.7***	***10.68***	Improvement at 2500rpm			4.23	45

Opto speed (Hz)	Rotor speed (rpm)	V0 value	V1 value	IIR (W)	others (W)	V rms (V)	I rms (A)	P input (W)	P mech (W)	Efficiency (%)	mech pow out factor	Air moved m^3/hr	Air moved l/sec	Temp (C) after25mins	dB noise 41.1
79	**1185**	255	8	0.65	1.28	38.9	0.1368	**2.31**	**0.38**	**16.53**	0.22	44.56	12.38		45.1
100	**1500**	185	8	0.75	1.08	38.97	0.1469	**2.44**	**0.61**	**25.08**	0.36	56.40	15.67		46.8
125	**1875**	128	8	0.58	0.92	38.98	0.1286	**2.45**	**0.96**	**39.03**	0.56	70.50	19.58		47.1
133.3	**2000**	113	8	0.52	0.77	38.95	0.1225	**2.38**	**1.09**	**45.69**	0.64	75.18	20.88		45.9
166.66	**2500**	45	8	0.43	-0.03	38.98	0.1109	**2.1**	**1.70**	**80.95**	1.00	94.00	26.11	24.2	47.2
200	**3000**	20	9	1.07	-0.33	38.91	0.1755	**3.19**	**2.45**	**76.74**	1.44	112.80	31.33		50.3
233.3	**3500**	1	10	2.55	-0.77	38.95	0.2706	**5.11**	**3.33**	**65.19**	1.96	131.58	36.55		52.6
SP IM	***2520***	-	-	-	-	-	-	***15.92***	***1.7***	***10.68***	Improvement at 2500rpm			7.58	45

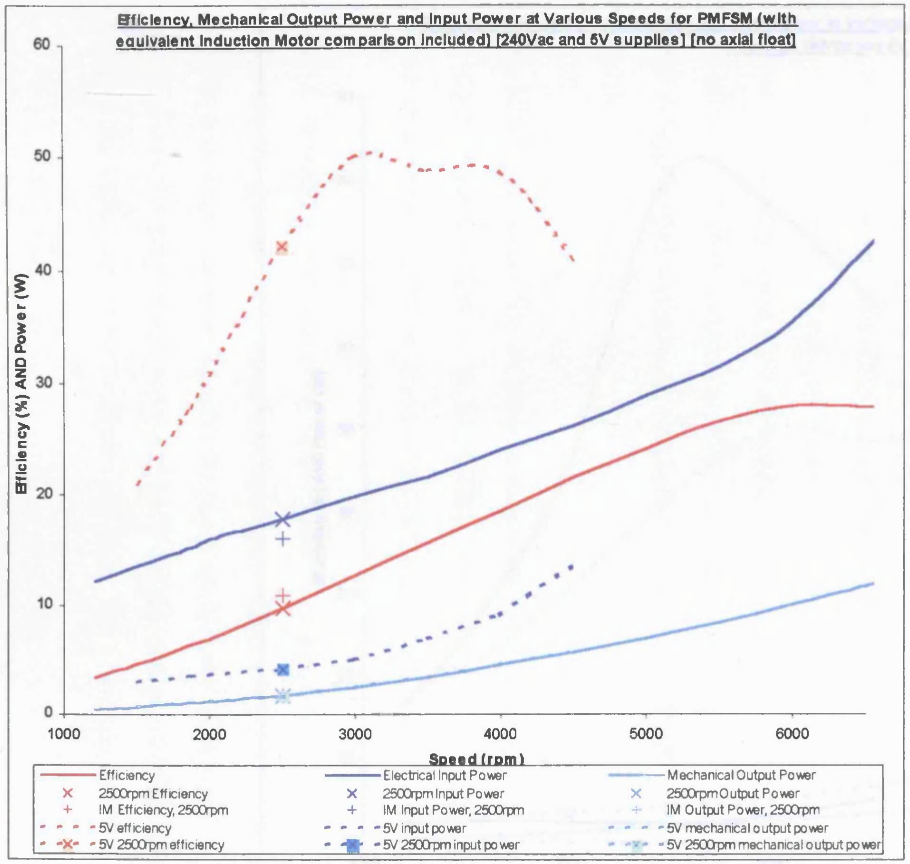


Figure 6.47 :- 240V ac test results.

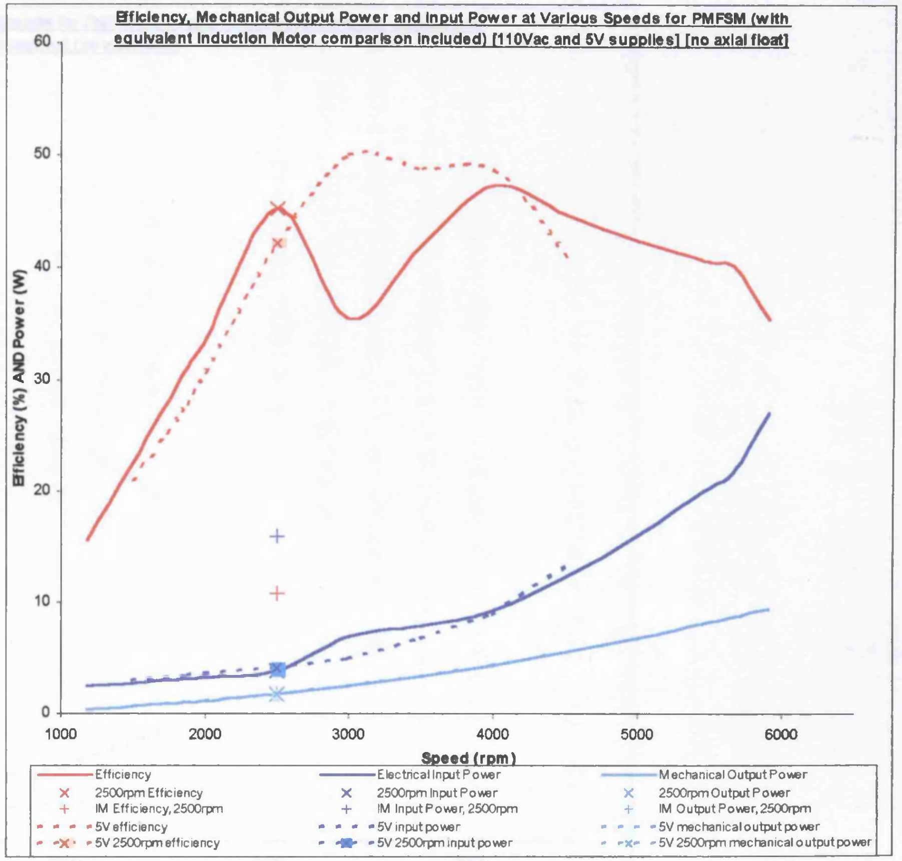


Figure 6.48 :- 110V ac test results.

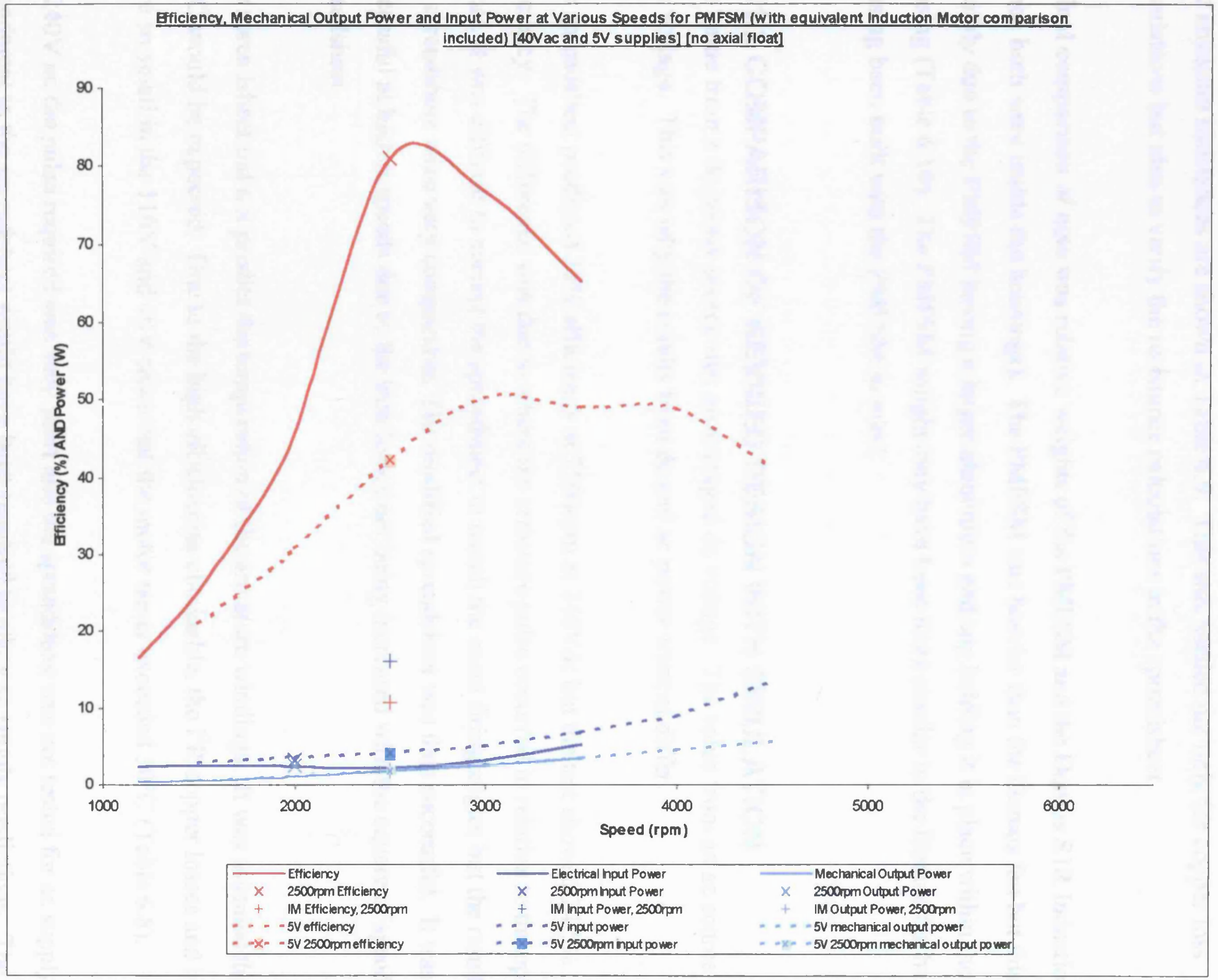


Figure 6.49 :- 40V ac test results.

TABLE 6.9 – ARMATURE RESISTANCE.

Resistance (ohms)			
Phase A	Phase B	Total	Avge
34.88	34.69	69.57	34.79

TABLE 6.10 – COMPARISON OF WEIGHTS BETWEEN PMFSM AND ORIGINAL INDUCTION MOTOR.

Motor (including casings)	Weight (g)
Domus S1R 100mm Axial Wall Fan	560
PMFSM 100mm Axial Wall Fan	630

The armature resistances are shown in Table 6.9. This was needed not only for copper loss calculations but also to verify the resistance calculations in the spreadsheet.

A final comparison of note was relative weights of the PMFSM and the Domus S1R Induction motor (when both were inside fan housings). The PMFSM was heavier than the Domus fan but this was possibly due to the PMFSM having a larger aluminium end cap holding it in place within the original housing (Table 6.10). The PMFSM weight may have been more similar to the Domus fan had the fan housing been built with the PMFSM in mind.

6.11.2 COMPARISON OF REVISED DESIGN WITH SIMULATION

The value from a dc power source was an averaged dc voltage. The value from an ac source was an rms voltage. This was why the results from dc and ac power sources differ.

The spreadsheet predicted 36% efficiency at 2500rpm at 240Vdc but the test showed 24.5% efficiency. The difference was due to where the armature pulse occurred in relation to the opto sensor signal. It was difficult to correct the spreadsheet to match the exact firing angles but the results from the spreadsheet were very comparable. The modified spreadsheet was thus successful. It was less successful at higher speeds due to the iron losses not being increased with the square of speed in the spreadsheet.

The spreadsheet did not predict the temperature of the armature windings. It was assumed that 40-60°C would be expected. Due to the high efficiencies obtainable, the I^2R copper losses and iron losses were so small in the 110V and 40V cases that the motor never exceeded 30°C (Table 6.8).

At 240V ac the pulse required was very short and the spreadsheet was not tested for ac supply. An amendment to the spreadsheet would have been required to allow ac supply predictions. The small turn number meant that, at lower speeds, the pulse would have to be short, as the spreadsheet confirmed. The efficiency was lower at the 240V supply compared to the 110V supply (it was less efficient than the induction motor at 240V – 9.63%). The 110V supply offered 45% efficiency with

3.76W input power. Both the 240V and 110V versions operated at very high speeds (6675rpm and 5910 rpm respectively) allowing wider potential applications.

81.0% (3s.f.) efficiency was obtained with a 40V rms ac supply (2.1W input power) (Table 6.8, *Figure 6.49*). This was a 7.58 times improvement on the induction motor currently used (10.7% efficient, 15.92W input power [9.36 times more power for the same air flow rate]). The motor only had a temperature of 24.2°C after 25 minutes of continual use. This highlighted the very low copper losses (0.43W at 2500rpm). The use of permanent magnets had substantially improved machine performance.

6.11.3 AIR FLOW COMPARISON WITH OTHER MOTORS

To move air at the same rate as the Vent-Axia LoWatt range of machines, the speed of the motor was increased in proportion to the increase in air flow rate (Domus quoted 94m^3/hr for its induction motor model, Vent-Axia quoted 110m^3/hr (with wall kit fitted) for its permanent magnet brushless dc motor). A comparison of input power for same air movement rate showed the relative efficiencies of the designs (Table 6.11). The Vent-Axia LoWatt range of fans had been merchandised as a green product so these fans were benchmarks for a 'green' design. The PMFSM had to operate at 2500/94x110 rpm=2925rpm for 110m^3/hr (6.2b). The PMFSM was not as efficient as the LoWatt models at higher supply voltages (the LoWatt existed in a 12V dc form format in the comparison). At 110V rms ac, the PMFSM was similar in performance. The 40V rms ac version took far less input power for the same air flow rate (41% less power) thus demonstrating the merit of the motor as a 'green' solution. It was shown that the PMFSM was very competitive even as a first prototype.

TABLE 6.11 – AIR FLOW COMPARISONS.

Motor	Input Power (W)	Efficiency (%)	Input Power at 110m^3/hr air flow (W)
Vent-Axia LoWatt LP (100mm fan)	5.20	?	5.20
Vent-Axia LoWatt UP (150mm fan)	6.90	?	6.90
Domus S1R 100mm Axial Wall Fan	15.92	10.68	N/A
PMFSM 5Vrms (dc supply)	4.04	42.00	4.89
PMFSM 240Vrms (dc supply)	6.93	24.50	6.28
PMFSM 240Vrms (ac supply)	17.65	9.63	19.20
PMFSM 110Vrms (ac supply)	3.76	45.20	6.31
PMFSM 40Vrms (ac supply)	2.10	81.00	3.07
PMFSM 40Vrms (ac supply) [one phase]	2.45	69.40	-
PMFSM 40Vrms (ac supply) [other phase]	2.52	67.50	-
PMFSM 240Vrms (ac supply) [skewed rotor]	17.95	9.50	19.50
PMFSM 110Vrms (ac supply) [skewed rotor]	6.50	26.20	7.25
PMFSM 40Vrms (ac supply) [skewed rotor]	2.73	62.30	3.94

6.12 ACOUSTIC NOISE TESTS

The Fan Industry is increasingly becoming aware of the need for greener products. Low power consumption is required and there is a possibility of EU directives enforcing strict requirements regarding this issue. Also, the noise from the overall package is of concern. Domus had always produced machines that were quiet, a fact that it was keen to retain. Strictly speaking, the motor designed for this application had not been built to be a quiet motor – it was been built to provide a low power, high efficiency alternative to the current product range for the Fan Industry as a whole. But a comparison was made between noise emissions from the Domus induction motor package and the first prototype PMFSM package. The tests were performed in the quietest room available (an anechoic chamber would have been ideal), and the readings (measured in dBA – A weighted range) were taken 3 metres away from the motors, perpendicular to the central axis of each (Tables 6.12a, b, c, *Figures 6.50* and *6.51*). At low speeds, the main noises were from the motors themselves, such as bearing and shaft noises, motor vibration, and switching noises (the induction motor did not have the latter problem). At higher speeds the noise from the flow of air tended to be more prominent. It was not within the scope of this research to cover where noise was produced by the PMFSM. Short pulses in the armature windings did cause switching noise, and the size of pulse did affect the tone. This was made worse at higher supply voltages where the pulses were shorter. Longer pulses with the 40V rms ac source provided a quieter motor. The efficiency was adversely affected by motor noise so any future spreadsheet model ought to include this to give a better indication of motor performance. The 40V version was only 2.2dBA louder than the Induction Motor (47.2dBA c.f. 45.0dBA, background level 41.1dBA) (*Figure 6.51*). For a first prototype that was not built to as high a specification as a final product, it was a very encouraging sign. It would possibly be acceptable for applications such as toilets and bathrooms where the background noise is low. In applications such as the kitchen, it would not be noticed in terms of noise, as cooking and using washing machines etc will create far greater dBA levels.

TABLES 6.12a, b, c – dB NOISE LEVELS.

240V ac rms		110V ac rms		40V ac rms		IM at 240V ac rms		background noise	
speed	dB	speed	dB	speed	dB	speed	dB	speed	dB
1215	56.9	1178	52.7	1185	45.1			1000	41.1
1500	57.4	1500	53.3	1500	46.8			1500	41.1
1875	61.8	1875	54.5	1875	47.1			1875	41.1
2000	57.2	2000	52.3	2000	45.9			2000	41.1
2500	59.6	2500	52	2500	47.2	2500	45	2500	41.1
3000	61.9	3000	56.1	3000	50.3			3000	41.1
3500	59.8	3500	57.8	3500	52.6			3500	41.1
4000	62.4							4000	41.1
4500	64.6							4500	41.1
5000	66.6							5000	41.1
5500	66.4							5500	41.1
6000	66.8							6000	41.1
6500	69.8							6500	41.1
6675	70.4							7000	41.1

dB levels at 3m distance		at speed (rpm)= 2500		
Background	IM	40Vacrms	110Vacrms	240Vacrms
41.1	45	47.2	51.2	59.6

Running with One Phase Only

Opto speed (Hz)	**Rotor speed** (rpm)	**V0** value	**V1** value	**IIR** (W)	**others** (W)	**V rms** (V)	**I rms** (A)	**P input** (W)	**P mech** (W)	**Efficiency** (%)	**mech pow out factor**	**Air moved** m^3/hr	**Air moved** l/sec	**dB noise** 41.1
166.66	**2500**	28	11	0.60	0.15	39.38	0.131	**2.45**	**1.70**	**69.39**	1.00	94.00	26.11	46.9
166.66	**2500**	31	11	0.63	0.19	39.38	0.1348	**2.52**	**1.70**	**67.46**	1.00	94.00	26.11	47.8

It was noted that the motor could be operated with only one phase being used (the other phase was not given a gate signal to energise it). The motor could start from stationary but only when the rotor poles were in the positions such that energising that phase would cause the rotor to turn. Only half the aligned rotor positions were used for torque production. The use of one phase only allowed a lengthening of the pulse for torque production and a halving of the number of pulses required. It was noted that the machine operated at lower efficiencies than the two phase version for each phase but the noise level from one phase was slightly less than the two phase version (the other was slightly more).

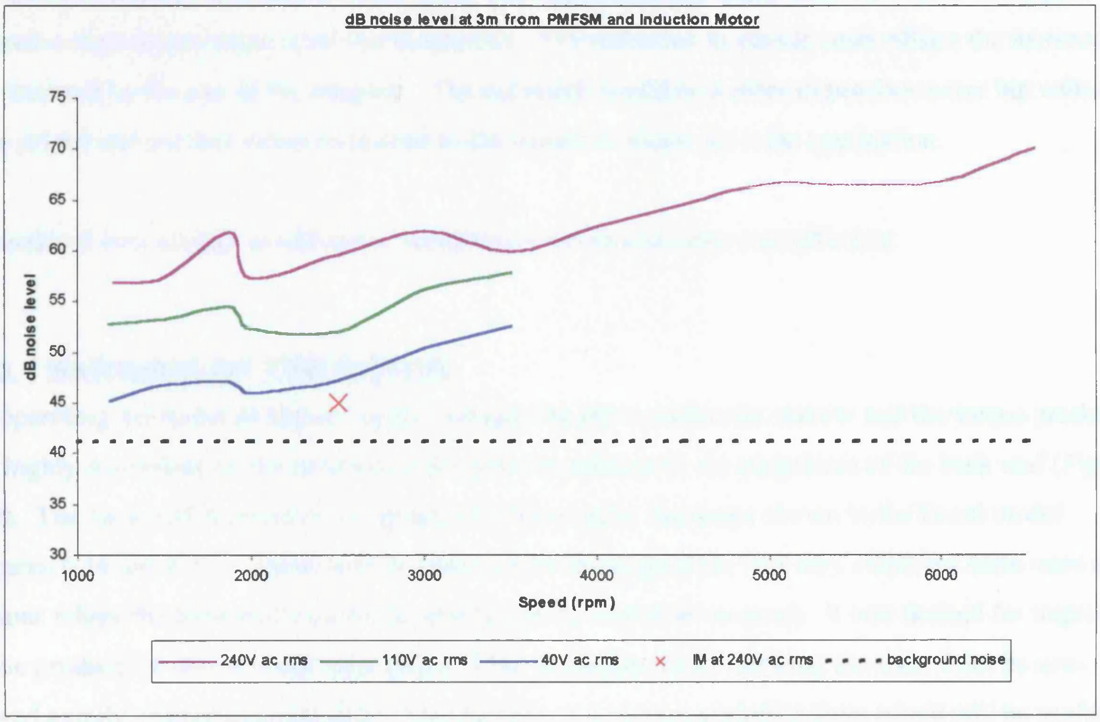


Figure 6.50 :- dB noise levels with rotor speed for various supply voltages.

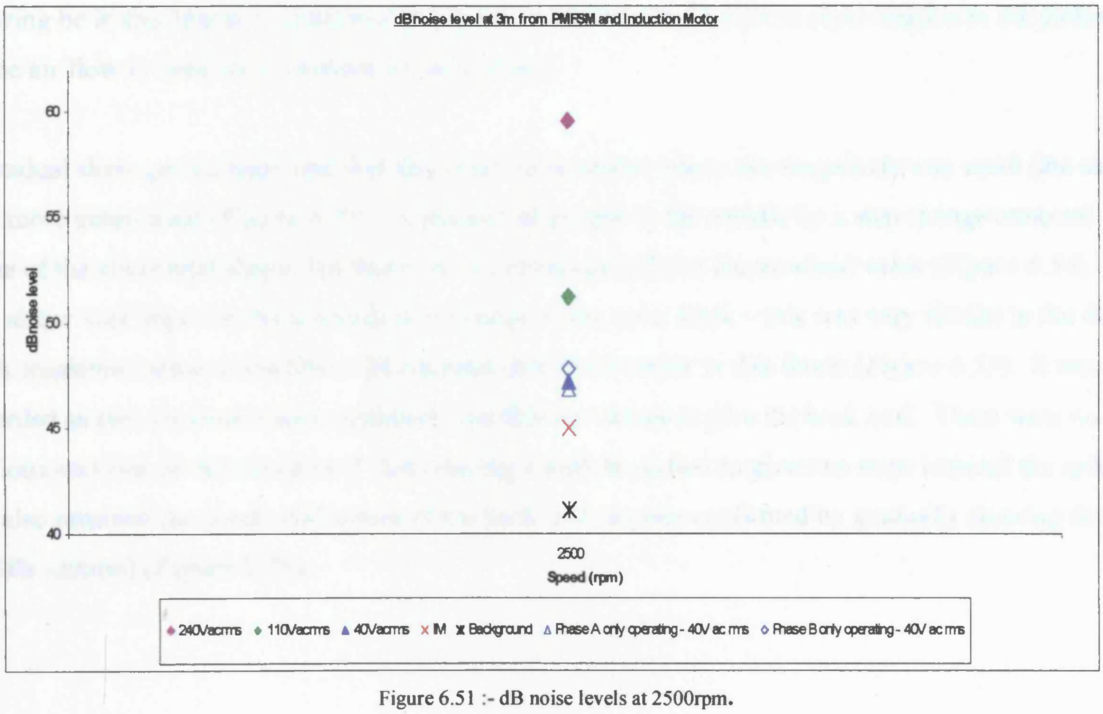


Figure 6.51 :- dB noise levels at 2500rpm.

6.13 FURTHER DESIGN IMPROVEMENTS

The motor indicated the potential for much improved efficiency and a wide range of practical uses at no extra final product cost. It would replace a wide range of motors.

It operated at a lower temperature than many existing motors and hence showed it does not require expensive high temperature rated thermoplastics. The reduction in plastic costs offsets the increased cost incurred by the use of the magnets. The net result would be a more expensive motor but with a same priced end product when compared to the induction motor for a fan application.

A powdered iron variant would make manufacture easier and more cost effective.

6.13.1 SKEWING OF THE ROTOR

For operating the motor at higher supply voltages the pulse width was narrow and the torque produced was highly dependent on the position of the pulse in relation to the magnitude of the back emf (*Figure 6.40*). The back emf waveform was practically identical to the shape shown in the Excel model (*Figures 6.16* and *6.52*). There were sections where the magnitude was very small but there were also sections where the back emf was flat topped (good for smoother torques). It was desired for improved torque production over a wider rotor angle. This was achieved by skewing the rotor from its non-skewed axially aligned original state. The skewing was chosen such that there would still be preferred aligned positions for the rotor in relation to the stator poles and that any air movement caused by the skewing be in the direction of air movement due to the fan blades (left to right rotation in the pictures as the air flow is towards the bottom of each photo).

A gradual skew gave a back emf that had small rotor angles where the magnitude was small (the shape was more sinusoidal) (*Figure 6.53*). A gradual skew split in the middle by a step change removed some of the sinusoidal shape, but there was a prominent spike at the maximal value (*Figure 6.54*). The shape was improved by a simple step change in the rotor stack – this was very similar to the dual stack implementation in the DSVRM machine described earlier in this thesis (*Figure 6.55*). It was regarded as two separate rotors combining the flux variations to give the back emf. There were no obvious sections of little back emf. Introducing a middle section to give two steps reduced the spike but also returned the sinusoidal nature of the back emf (as was confirmed by gradually skewing this middle section) (*Figure 6.56*).

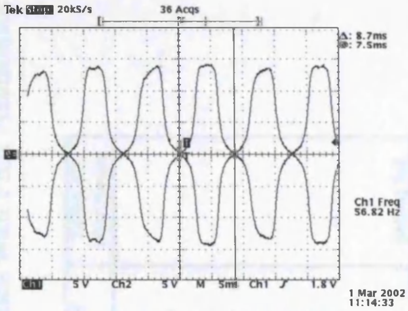


Figure 6.52 :- Non-skewed back emf.

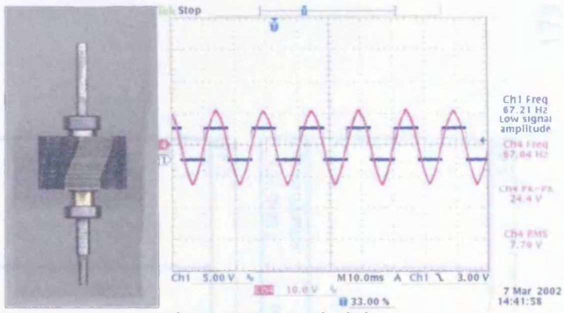


Figure 6.53 :- Gradual skew.

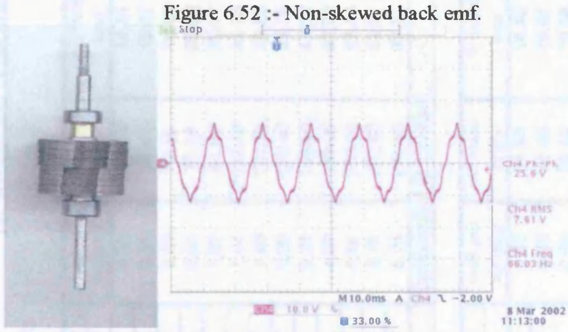


Figure 6.54 :- Stepped skew with graduation.

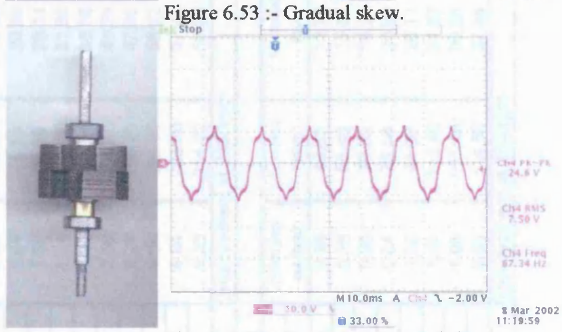


Figure 6.55 :- Dual stack stepped skew.

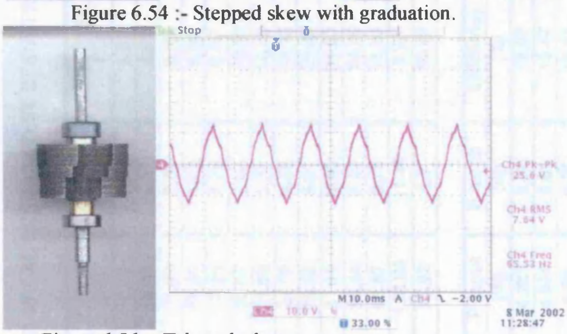


Figure 6.56 :- Tri-stack skew.

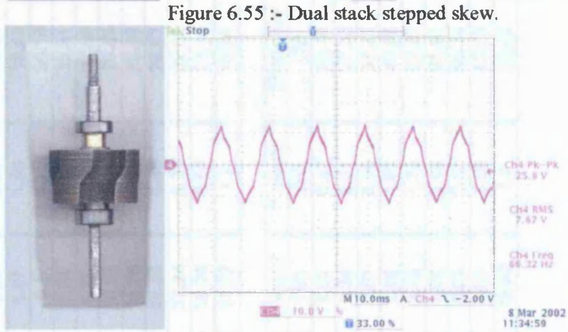


Figure 6.57 :- 40V Dual stack graduated skew.

Gradual skewing appeared to give a more sinusoidal back emf but a stepped skew maintained a section of larger back emf but there was a high value spike where the step occurred (as this step passed the stator pole) (*Figure 6.57*).

The preferred back emf was from that of the rotor having a single stepped skew (*Figure 6.55*). This demonstrated the usefulness of a dual stack theory for manipulating the back emf waveform. The resultant waveform had a back emf that was larger over a wider rotational angle of the rotor but with a slight reduction in the maximum value in back emf. This meant that the motor would be less efficient but with improved torque production at higher supply voltages. The motor took 2.73W input power and was 62.3% efficient at 250rpm at 40V rms ac (c.f. 2.1W, 81% for non-skewed version).

Tests were then performed to show how a skewed rotor affected the motor performance (Table 6.13, *Figures 6.58 – 6.60*).

TABLE 6.13 – 240V, 110V AND 40V AC SKEWED TEST RESULTS.

SKEWED (DUAL STACK VERSION)

Opto speed (Hz)	**Rotor speed (rpm)**	V0 value	V1 value	IIR (W)	others (W)	V rms (V)	I rms (A)	**P input (W)**	**P mech (W)**	**Efficiency (%)**	mech pow out factor	Air moved m^3/hr	Air moved l/sec	**Temp (C)** after25mins
92	**1380**	255	2	0.27	12.24	237.01	0.0874	**13.02**	**0.52**	**3.98**	0.30	51.89	14.41	
100	**1500**	231	2	0.28	12.66	237.4	0.0903	**13.56**	**0.61**	**4.51**	0.36	56.40	15.67	
125	**1875**	176	2	0.35	13.92	237.4	0.1007	**15.23**	**0.96**	**6.28**	0.56	70.50	19.58	
133.3	**2000**	162	2	0.38	14.19	236.9	0.105	**15.66**	**1.09**	**6.94**	0.64	75.18	20.88	
166.66	**2500**	117	2	0.47	15.78	237.4	0.1165	**17.95**	**1.70**	**9.47**	1.00	94.00	26.11	
200	**3000**	83	2	0.58	17.04	237.9	0.1291	**20.07**	**2.45**	**12.20**	1.44	112.80	31.33	
233.3	**3500**	57	2	0.68	17.98	237.5	0.1398	**21.99**	**3.33**	**15.15**	1.96	131.58	36.55	
266.66	**4000**	38	2	0.81	18.85	237.4	0.1529	**24.02**	**4.35**	**18.12**	2.56	150.40	41.78	
300	**4500**	24	2	1.00	20.47	237.4	0.1697	**26.98**	**5.51**	**20.42**	3.24	169.20	47.00	
333.34	**5000**	10	2	1.27	22.94	235.9	0.1909	**31.01**	**6.80**	**21.93**	4.00	188.00	52.22	
366.66	**5500**	20	3	4.10	46.22	235.3	0.3432	**58.54**	**8.23**	**14.05**	4.84	206.80	57.44	
400	**6000**	11	3	4.82	49.25	235	0.3724	**63.87**	**9.79**	**15.33**	5.76	225.60	62.67	
433.33	**6500**	7	3	15.20	90.20	237.5	0.6611	**116.89**	**11.49**	**9.83**	6.76	244.40	67.89	getting hot
SP IM	***2520***	-	-	-	-	-	-	***15.92***	***1.7***	***10.68***	Improvement at 2500rpm			0.89

SKEWED (DUAL STACK VERSION)

Opto speed (Hz)	**Rotor speed (rpm)**	V0 value	V1 value	IIR (W)	others (W)	V rms (V)	I rms (A)	**P input (W)**	**P mech (W)**	**Efficiency (%)**	mech pow out factor	Air moved m^3/hr	Air moved l/sec	**Temp (C)** after25mins
91.9	**1379**	255	3	0.16	3.60	108.52	0.0681	**4.281**	**0.52**	**12.07**	0.30	51.83	14.40	
100	**1500**	226	3	0.17	3.76	108.24	0.0707	**4.544**	**0.61**	**13.47**	0.36	56.40	15.67	
125	**1875**	168	3	0.26	4.07	108.12	0.0857	**5.28**	**0.96**	**18.11**	0.56	70.50	19.58	
133.3	**2000**	154	3	0.27	4.22	108.24	0.0877	**5.57**	**1.09**	**19.52**	0.64	75.18	20.88	
166.66	**2500**	99	3	0.34	4.46	108.21	0.0993	**6.5**	**1.70**	**26.15**	1.00	94.00	26.11	
200	**3000**	53	3	0.40	4.39	108.15	0.1074	**7.24**	**2.45**	**33.81**	1.44	112.80	31.33	
233.3	**3500**	25	3	0.59	4.87	108.06	0.1306	**8.79**	**3.33**	**37.90**	1.96	131.58	36.55	
266.66	**4000**	33	4	1.50	8.72	108.08	0.2079	**14.58**	**4.35**	**29.85**	2.56	150.40	41.78	
300	**4500**	14	4	1.93	9.32	108.02	0.2356	**16.76**	**5.51**	**32.86**	3.24	169.20	47.00	
333.34	**5000**	12	5	4.33	14.22	107.8	0.3528	**25.35**	**6.80**	**26.83**	4.00	188.00	52.22	
366.66	**5500**	6	6	9.36	19.66	107.88	0.5186	**37.24**	**8.23**	**22.09**	4.84	206.80	57.44	
380	**5700**	1	6	9.84	19.59	107.47	0.5318	**38.26**	**8.84**	**23.10**	5.20	214.32	59.53	
SP IM	***2520***	-	-	-	-	-	-	***15.92***	***1.7***	***10.68***	Improvement at 2500rpm			2.45

SKEWED (DUAL STACK VERSION)

Opto speed (Hz)	**Rotor speed (rpm)**	V0 value	V1 value	IIR (W)	others (W)	V rms (V)	I rms (A)	**P input (W)**	**P mech (W)**	**Efficiency (%)**	mech pow out factor	Air moved m^3/hr	Air moved l/sec	**Temp (C)** after25mins
88	**1320**	255	7	0.44	1.11	39.24	0.1123	**2.02**	**0.47**	**23.46**	0.28	49.63	13.79	
100	**1500**	214	7	0.52	1.10	39.28	0.1228	**2.24**	**0.61**	**27.32**	0.36	56.40	15.67	
125	**1875**	153	7	0.65	0.96	29.24	0.1368	**2.57**	**0.96**	**37.21**	0.56	70.50	19.58	
133.3	**2000**	136	7	0.68	0.90	39.23	0.1395	**2.66**	**1.09**	**40.88**	0.64	75.18	20.88	
166.66	**2500**	42	7	0.66	0.37	39.1	0.1378	**2.73**	**1.70**	**62.27**	1.00	94.00	26.11	
200	**3000**	6	8	1.68	-0.15	39.22	0.22	**3.98**	**2.45**	**61.51**	1.44	112.80	31.33	
233.3	**3500**	1	11	3.21	-0.52	39.2	0.304	**6.03**	**3.33**	**55.24**	1.96	131.58	36.55	
239	**3585**	1	12	3.71	-0.57	39.19	0.3264	**6.63**	**3.50**	**52.73**	2.06	134.80	37.44	
SP IM	***2520***	-	-	-	-	-	-	***15.92***	***1.7***	***10.68***	Improvement at 2500rpm			5.83

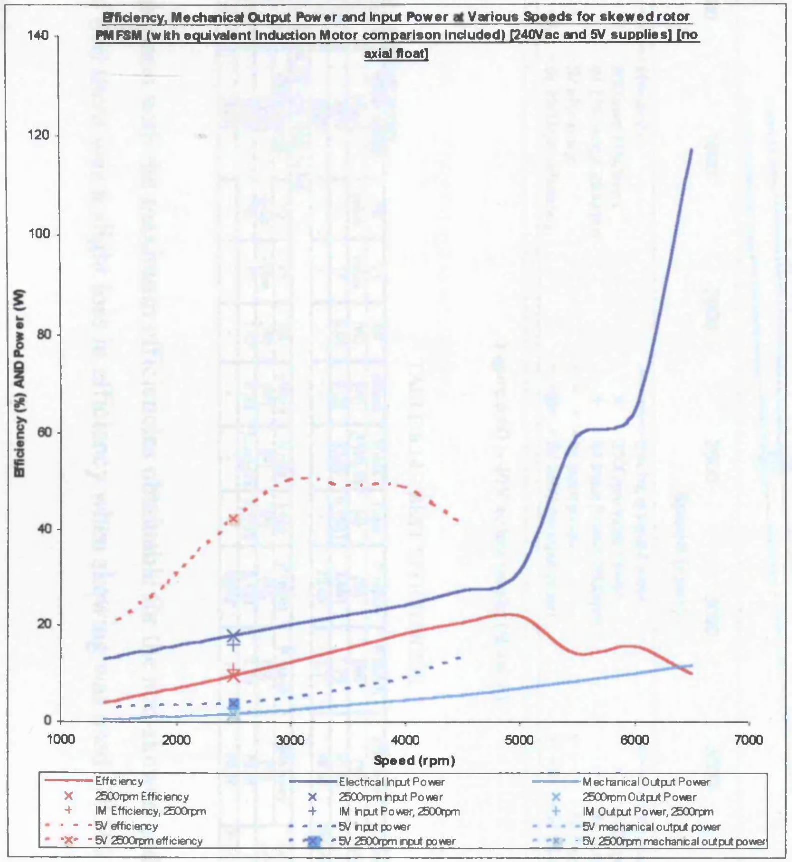


Figure 6.58 :- 240V ac test results (skewed).

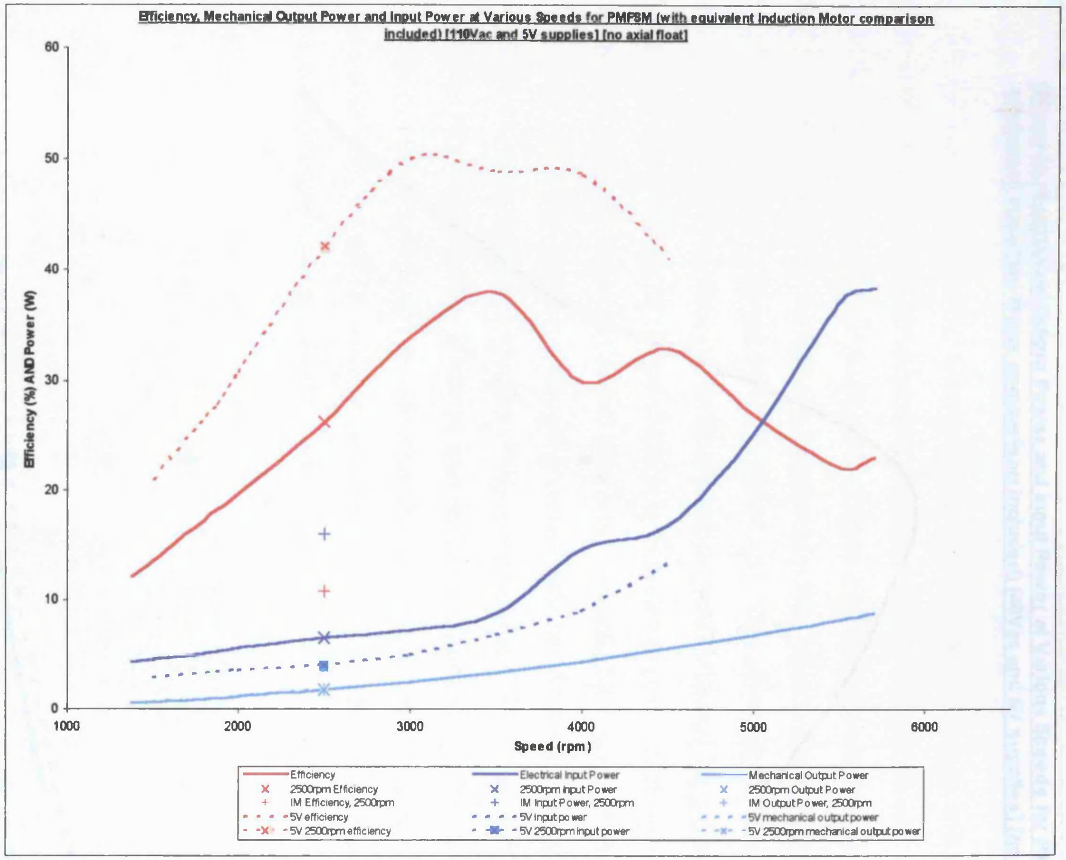


Figure 6.59 :- 110V ac test results (skewed).

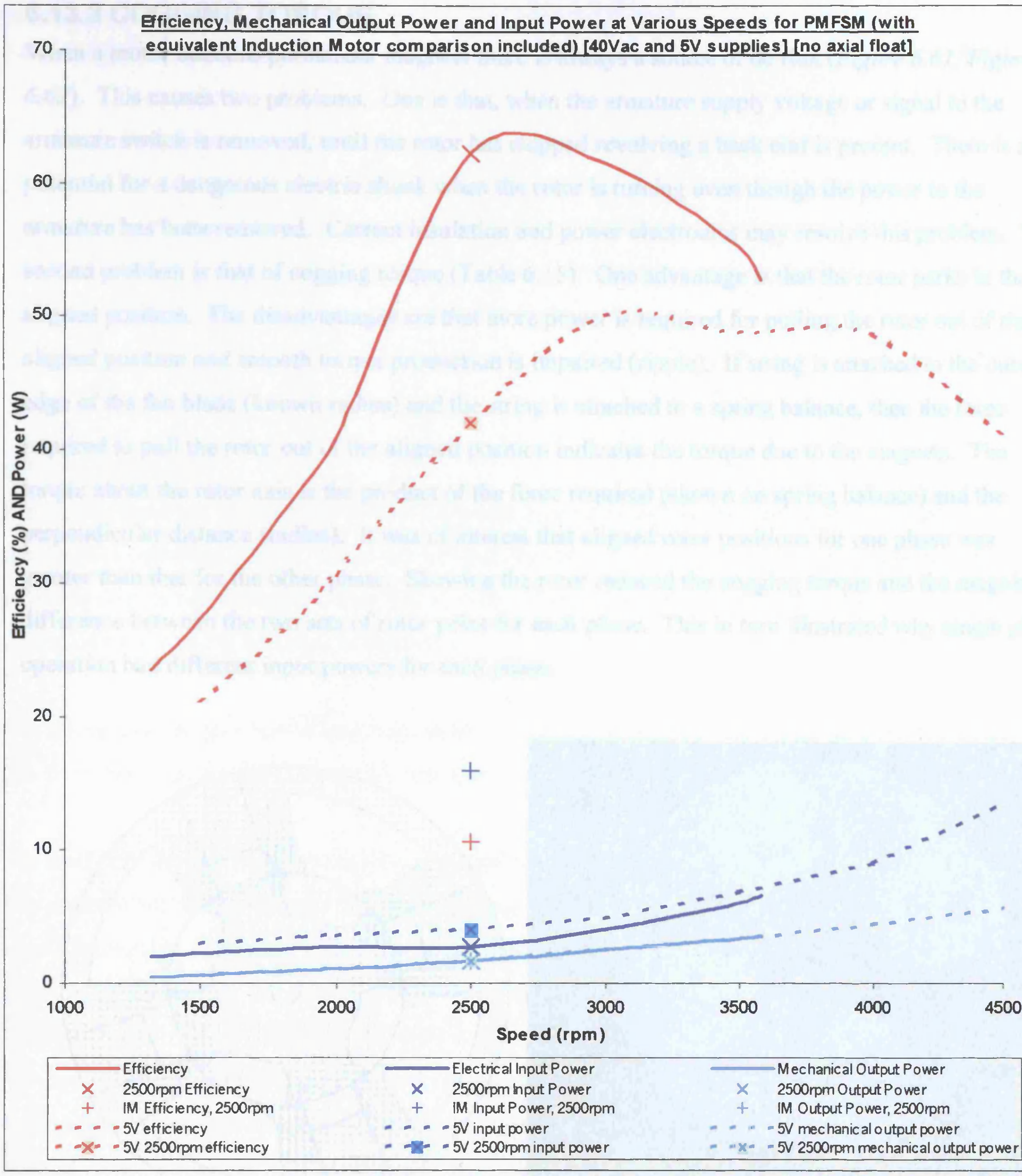


Figure 6.60 :- 40V ac test results (skewed).

TABLE 6.14 – BEST EFFICIENCIES.

Best non-skewed both phases

Opto speed (Hz)	Rotor speed (rpm)	V0 value	V1 value	IIR (W)	others (W)	V rms (V)23.56	I rms (A)	P input (W)	P mech (W)	Efficiency (%)	mech pow out factor	Air moved m^3/hr	Air moved l/sec
166.66	**2500**	1	17	0.98	-0.59	22.89	0.1681	**2.094**	**1.70**	**81.18**	1.00	94.00	26.11
SP IM	***2520***	-	-	-	-	-	-	***15.92***	***1.7***	***10.68***	Improvement at 2500rpn		7.60

SKEWED (DUAL STACK VERSION)

Opto speed (Hz)	Rotor speed (rpm)	V0 value	V1 value	IIR (W)	others (W)	V rms (V)	I rms (A)	P input (W)	P mech (W)	Efficiency (%)	mech pow out factor	Air moved m^3/hr	Air moved l/sec
166.66	**2500**	12	17	1.07	-0.63	23.49(24.21)	0.1752	**2.138**	**1.70**	**79.51**	1.00	94.00	26.11
SP IM	***2520***	-	-	-	-	-	-	***15.92***	***1.7***	***10.68***	Improvement at 2500rpn		7.45

A comparison with the maximum efficiencies obtainable for the non-skewed and skewed motors showed that there was a slight loss in efficiency when skewing was used (79.5% c.f. 81.2%) (Table 6.14).

6.13.2 COGGING TORQUE

When a motor contains permanent magnets there is always a source of dc flux (*Figure 6.61*, *Figure 6.62*). This causes two problems. One is that, when the armature supply voltage or signal to the armature switch is removed, until the rotor has stopped revolving a back emf is present. There is a potential for a dangerous electric shock when the rotor is turning even though the power to the armature has been removed. Correct insulation and power electronics may resolve this problem. The second problem is that of cogging torque (Table 6.15). One advantage is that the rotor parks in the aligned position. The disadvantages are that more power is required for pulling the rotor out of the aligned position and smooth torque production is impaired (ripple). If string is attached to the outer edge of the fan blade (known radius) and the string is attached to a spring balance, then the force required to pull the rotor out of the aligned position indicates the torque due to the magnets. The torque about the rotor axis is the product of the force required (shown on spring balance) and the perpendicular distance (radius). It was of interest that aligned rotor positions for one phase was greater than that for the other phase. Skewing the rotor reduced the cogging torque and the magnitude difference between the two sets of rotor poles for each phase. This in turn illustrated why single phase operation had different input powers for each phase.

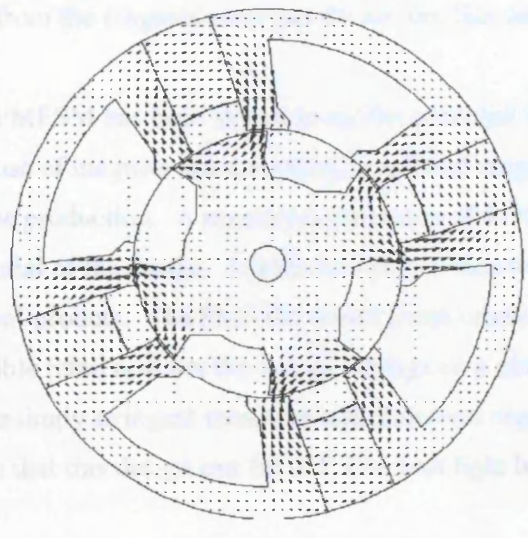


Figure 6.61 :- Flux paths due to magnets only.

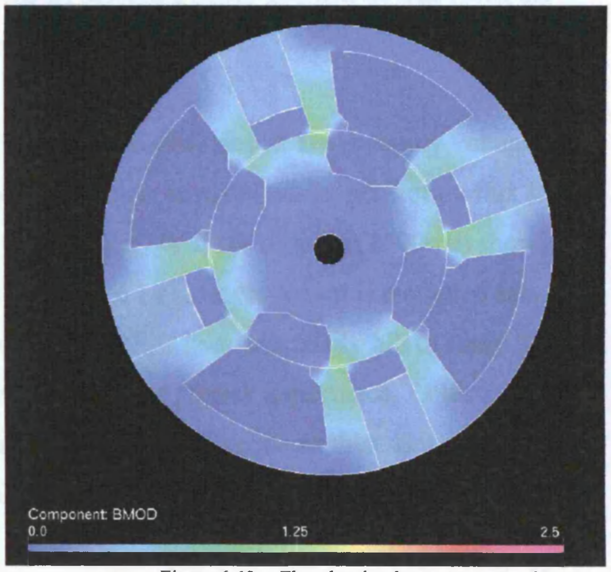


Figure 6.62 :- Flux density due to magnet only.

TABLE 6.15 – COGGING TORQUE.

Stator pole number (in rotational order)	Cogging torque (mNm) Not skewed	Skewed
1	4.17	1.83
2	34.99	8.66
3	4.00	2.00
4	39.98	9.16
5	4.33	2.67
6	36.65	7.33
7	5.66	3.00
8	33.32	8.00

6.13.3 POWDERED IRON

An investigation into the use of powdered iron to replace steel laminations is shown in Appendix H.

Powdered iron has real potential for this type of motor. It is known that magnets can be made from powder materials. Replacing the laminations with powdered iron allows more complex three-dimensional shapes and three dimensional flux paths. It is therefore suggested that this type of motor could be built in a mould injection system – involving fewer stages in the manufacturing process, especially if the plastic outer casing was included in the process.

6.14 SUMMARY

A PMFSM has been successfully built to fit inside a pre-existing fan housing to directly replace a shaded pole induction motor. It has one weaved coil (consequent winding) incorporating two parallel bifilar strands of wire instead of four sets of coils. The motor is similar in stack length to the Induction Motor and is identical in diameter.

It is shown that cheaper ferrite magnets can be utilised by selecting stator geometries that focus the flux from the magnets such that the air gap flux density is increased.

The PMFSM has been shown to exhibit potential for very high efficiency at lower supply voltages. The use of magnets has dramatically reduced magnetising losses incurred whilst generating flux for torque production. A maximum efficiency of 81% with only 2.1W input power at 2500rpm shows the potential of the design. It takes less power than the Vent Axia LoWatt range that is marketed as a 'green' product. The PMFSM would never create enough savings in an electricity bill to repay the possible retail cost but the energy savings on a global scale are of greater importance. With increasingly stringent measures and directives regarding efficiency of products there is a commercial niche that this design can fill (c.f. low watt light bulbs).

At fixed speeds increasing the supply voltage reduces the efficiency primarily due to a low number of turns in the armature coils (the back emf is not big enough). More turns would be preferable but the wire diameter becomes too thin to successfully wind on a commercial basis.

One solution would be to split the distribution of turns. One phase could have only a small number of turns to aid initial start up of the motor before the other phase runs the motor on its own – this other phase would have far more turns allowing better performance at 240V rms ac. A second alternative would be to alter the fan design such that the motor operated at higher speeds such that the back emf would be greater but higher speeds incurs undesirable noise issues (air noise, increased frequency of switching noise). A third possibility would be to use NeFeB rare earth magnets to replace the Fer3 magnets currently used but the increase in back emf and efficiency boost at the expense of increased magnet cost is not necessarily a viable option. The last possible solution is to put the armature windings from each phase in series and then use an H-bridge inverter circuit to employ bipolar current. The number of turns is doubled so this will allow improved efficiency at higher supply voltages but the circuit complexity has doubled (four switches instead of two).

The 110V rms ac version has increased efficiency compared to the Domus induction motor and operates over a wider speed range. The potential air flow rate is such that the motor can be used for all applications. The higher air flow rates are such that this design utilising a 4 inch fan could replace larger fan units. This has implications for the Fan Industry in that a single size fan unit may replace a wide range of fans, not only in terms of applications but also in terms of larger units being phased out (e.g. 6 inch and 9 inch fans). With the PMFSM having its armature phase pulses determining the speed of the fan, it is possible for one PMFSM being made such that a set of dip switches can determine how the fan is to operate (since a PIC microcontroller is easily programmable for such possibilities). The 110V ac version also competes in terms of performance with the Vent-Axia LoWatt range.

The 40V rms ac version is the preferred option given the current turns selection. The prototype fan is only 2dB noisier than the Domus fan. The efficiency is 81% at the same speed as the Domus fan and the input power is 2.1W. This is a 7.58 times the efficiency of the induction motor currently being used (10.7% efficient, 15.92W input power [9.36 times more power for the same air flow rate]). Also, by having such a low input power, the motor only operates at 24.2°C, allowing the use of cheaper thermoplastics. Integration with low temperature rated thermoplastic moulded products provides scope for cost savings that offsets the extra cost of the magnets so the end product is at no additional expense. The 40V motor convincingly outperforms the LoWatt range at identical air flow rate (41% less input power required).

The model does not predict very accurately the performance of the machine at higher speeds. Losses such as noise, vibration, bearing losses cannot be predicted. Iron losses are not predicted accurately and this is a major source of potential losses at higher speeds (non-linear). A newer model would have to address these issues. Finite Element Analysis and the current spreadsheet give a clear indication of the motor performance at lower speeds and have allowed a motor to be built that is 81% efficient.

Skewing allows manipulation of the back emf waveform to allow torque production over a larger rotor angle. This is at the expense of a small amount of efficiency but it does allow a reduction in cogging torque. The use of the dual stack theory of the DSVRM design has proved to be best for this motor.

The motor is best suited for lower supply voltage to the armature coils. Higher voltages require pulsed timings and perhaps PWM control algorithms (not covered in this work). Otherwise the use of transformers or autotransformers may be needed for efficient motor operation at higher supply voltages.

Noise from the motor is an issue which has not been designed for. Longer armature pulses or operation of one phase only may improve noise from the motor. Better manufacture of the motor may reduce vibrational noise. Also, much work exists on altering the control algorithms for acoustic noise cancellation.

It may also be advisable to consider an external rotor version of the motor. The fan can be incorporated directly into the design. Less copper will be used as the mean length per turn can be reduced (reducing copper losses).

A powdered iron version of PMFSM has scope for further investigation. It is possible for magnets and stator sections to be made from powdered material allowing a simple process to incorporate the fan housing into the final design (compaction press method, for example).

With the high efficiencies, low running temperatures and wide speed ranges of the motor, it could be argued that the PMFSM is too powerful for the 2500rpm application. It may be possible to reduce the motor diameter and/or stack length such that the motor will no longer have the ability to produce so much output power. A smaller design would be cheaper and allows an improved rotor design for a 4 inch fan blade. The fan could also have blades to move air over the windings for extra cooling and total airflow. The original specification was to design a motor that replaced the Domus induction motor unit. This has been achieved with a great amount of success to the point that further work may be considered to bring this design to a production level.

The Permanent Magnet Flux Switching Motor has shown significant commercial potential and could be used for a wider range of applications than the existing induction motor plus some of the larger fan applications (e.g. 6 inch fans). The very low power ability and its inherent electronic control has attracted interest from Domus Ventilation Ltd. and Vent Axia Ltd. who are looking at investing in furthering its development.

CHAPTER 7

COMPARISON OF RELUCTANCE MACHINES WITH FLUX ASSISTANCE

7.1 INTRODUCTION

Three machines have been designed, built and tested. The Dual and Single Stack Variable Reluctance Machines (DSVRM and SSVRM) have been built using the same design methods, using the same underlying theory regarding the use of the toroidal field winding. They differ in the shape of each machine. The Permanent Magnet Flux Switching Motor replaced the field winding with permanent magnets. This thesis investigated the design and performance of these machines. This chapter investigates similarities and differences between all three machines in terms of effectiveness of flux production. This includes use of one or two stacks for the variable reluctance machines, permanent magnets replacing the field coils, axial and radial fluxes, and bipolar versus unipolar armature currents.

7.2 COMPARISONS BETWEEN SINGLE AND DUAL STACK VARIABLE RELUCTANCE MACHINES

The dual stack variable reluctance machine is a unique machine in that its back emf waveform is in fact variable depending on the mechanical displacement angle between the two stacks. The back emf in each phase is of two components, a back emf in one stack and an additional second back emf which is phase shifted due to the angle of the displacement. The back emf at zero degrees displacement is triangular due to small stator pole arcs. There is a similar back emf shape in the single stack variable reluctance machine. Wider pole arcs would reduce this problem (or a skewing of the rotor and/or stator perhaps). The problem was overcome in the dual stack machine by being able to phase shift the back emfs within each phase by altering the displacement angle. By doing so the shape of the back emf is made less triangular and more square which allows smoother torque production over a larger rotational angle of the rotor. The disadvantage is that this reduces the magnitude of the back emf. A trade off is required that allows smoother torque production but a larger back emf (which means a larger magnitude of torque produced as well). The single stack machine does not have this capability.

The single stack machine has a smaller size. Only one stack is used instead of two. The field winding was housed in a steel section linking the back iron to the rotor. This housing allows the field winding to be simply dropped in during manufacture. The armature windings prevent the field winding from falling out. The dual stack machine needs the field winding clamped inside the cylindrical back iron linking the two stacks. The armature windings of each stack hold the field in place. The single stack machine uses less steel (it is smaller and lighter) and is easier to build.

Both machines produce axial flux from the toroidal field winding. Both were found to be most efficient when having unipolar excitation with single coil per pole windings energised such that all the windings on two diametrically opposite stator poles are associated with one phase only. Bipolar excitation with bifilar windings hindered torque production as flux is used to prevent flux leaking out of the unaligned stator poles. The unipolar excitation with single coil per pole method does not attempt to prevent flux leakage so all its produced flux is available for torque production. Also, more flux is produced in each energized pole aiding pulling the rotor into the aligned position (the bipolar excitation with bifilar windings version has two sets of coils pulling the rotor into alignment and two more preventing flux leakage and also pushing the rotor into alignment).

The diagrams may refer to the windings as 'conv' and 'bif' for unipolar excitation with single coil per phase and bipolar excitation with bifilar windings respectively only for the purpose of simplifying the text.

Both machines are well suited for having the field winding in series with the armature windings as increasing the field current improves efficiency and torque production.

The tests carried out and the manner of the tests were identical for each machine. The dual stack machine had a tendency to improve in efficiency as it heated up, which was not noted in the single stack machine. Both machines were capable of self-starting (this is true for the angles of displacement chosen for load testing). The use of identical tests allowed a direct comparison of the performance of each machine.

The dual stack machine required more total input power at lower loads but, as the load increased, it only required a gradual increase in input power to maintain machine speed (*Figure 7.1*). The single stack machine started with much less input power at low load but showed a definite trend for far higher input powers at higher loads (a marked increase).

This was shown in the efficiency curves whereby the efficiency of the single stack machine was far greater than the dual stack version at low loads but the difference was far less at higher loads. The trend implied that the dual stack machine would be more efficient at higher loads if testing was continued.

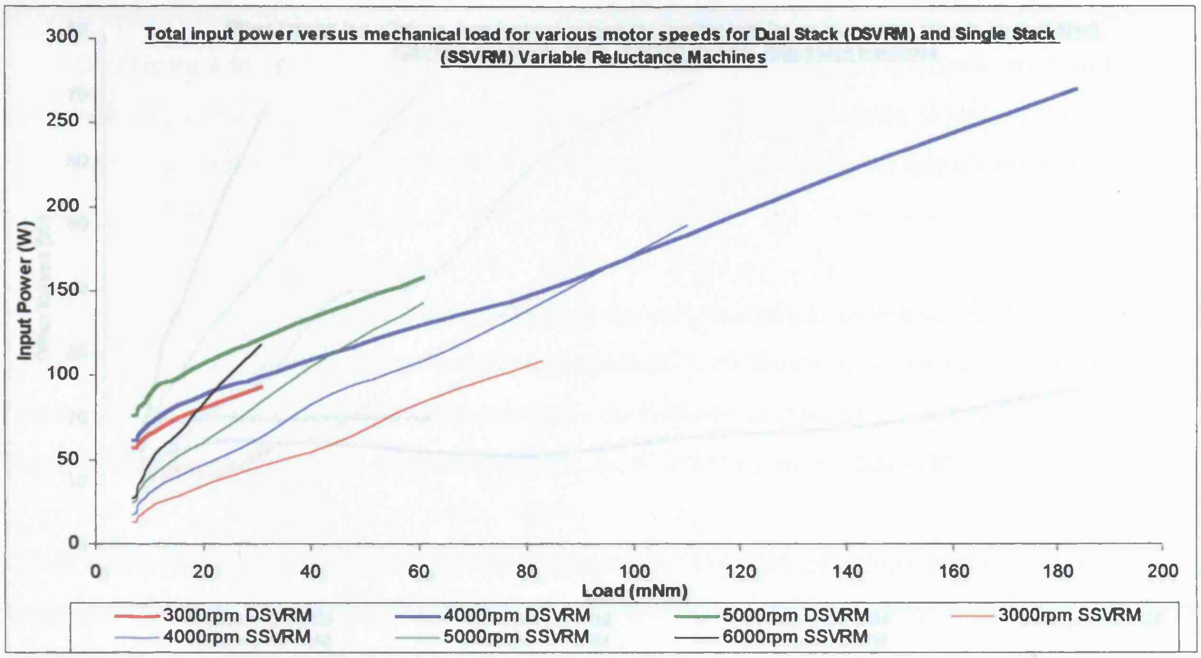


Figure 7.1 :- Input powers for VRM machines at various speeds.

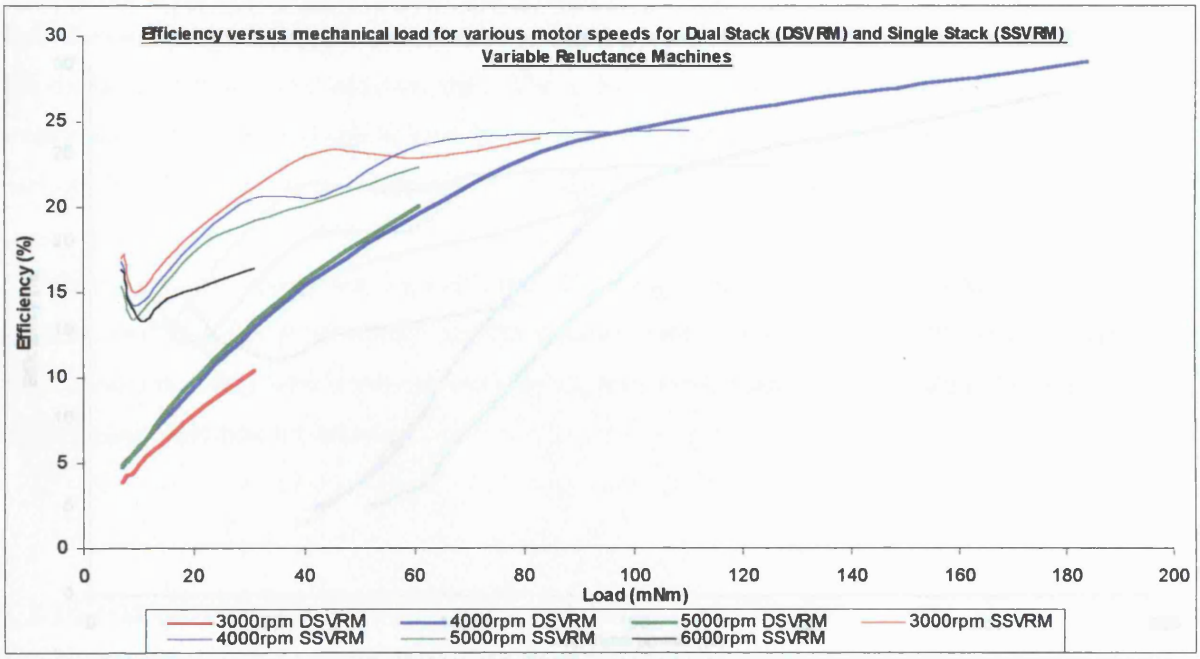


Figure 7.2 :- Efficiencies for VRM machines at various speeds.

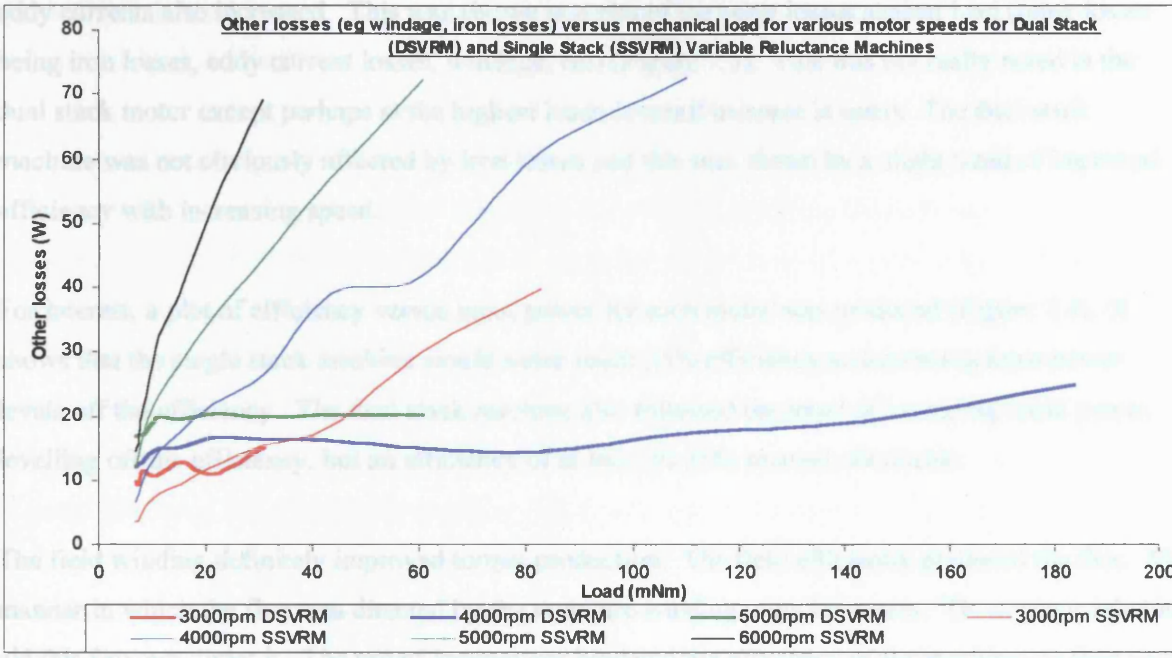


Figure 7.3 :- Iron losses for VRM machines at various speeds.

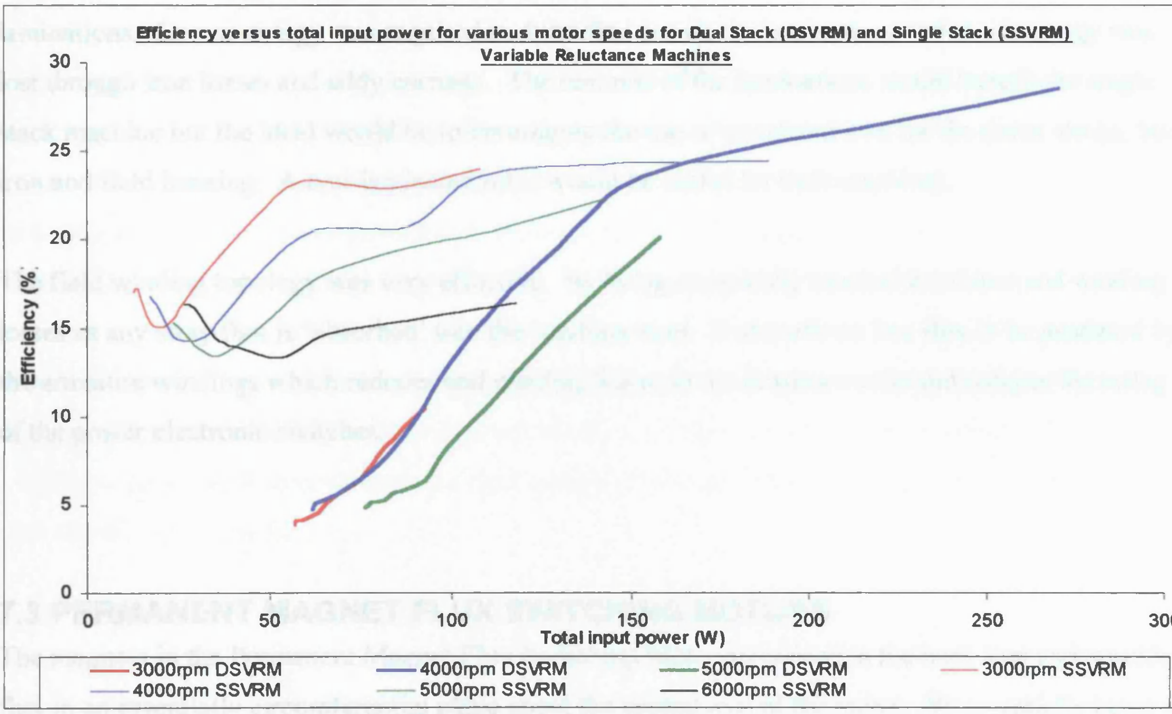


Figure 7.4 :- Efficiency versus input power for VRM machines at various speeds.

The interesting note was the effect of speed on efficiency (*Figure 7.2*). The single stack machine was more efficient at lower speeds but the trend was reversed in the dual stack machine where lower speeds reduced the efficiency. The single stack machine was noted to require significant increases in input power for small increases in load and that the efficiency improved with a reduction in rotor speed. This implied that iron losses due to cross lamination fluxes were significant. As the load increased more input power was required to drive the flux across the laminations. Iron losses and eddy currents also increased. This was shown in a plot of the other losses against load (other losses being iron losses, eddy current losses, windage, etc) (*Figure 7.3*). This was not really noted in the dual stack motor except perhaps at the highest loads (a small increase is seen). The dual stack machine was not obviously affected by iron losses and this was shown by a slight trend of improved efficiency with increasing speed.

For interest, a plot of efficiency versus input power for each motor was produced (*Figure 7.4*). It shows that the single stack machine would never reach 25% efficiency as increasing input power levels off the efficiency. The dual stack machine also followed the trend of increasing input power, levelling off the efficiency, but an efficiency of at least 30-35% seemed obtainable.

The field winding definitely improved torque production. The field efficiently produced the flux. The manner in which the flux was directed by the armature windings was important. The armature should aid this flux not direct it. The use of laminations hindered the efficiency of the machines as they were deigned for flux along the plane of the laminations and not three dimensional flux across the laminations. Excess energy was required to drive flux across the laminations and much energy was lost through iron losses and eddy currents. The removal of the laminations would benefit the single stack machine but the ideal would be to investigate the use of powdered iron for the stator stacks, back iron and field housing. A non-laminated rotor would be useful for both machines.

The field winding topology was very effective. By being completely internal it reduces end winding losses as any stray flux is 'absorbed' into the machine steel. It also allows less flux to be produced by the armature windings which reduces end winding losses in the armature coils and reduces the rating of the power electronic switches.

7.3 PERMANENT MAGNET FLUX SWITCHING MOTORS

The magnets in the Permanent Magnet Flux Switching Motor are placed in the back iron and provide flux in an essentially circumferential plane about the central axis of the motor. By essentially being a constant flux source the iron losses in the steel sections either side of the magnets are reduced. The iron losses are also reduced by there being no requirement to energise and completely de-energise the flux in the motor for each phase excitation.

The use of the magnets dramatically reduce the magnetizing losses during the flux provision for torque production in small motors. This allowed a low running temperature for such a motor. With losses being low in such a motor the obtainable efficiency was 81% since the required input power was low for a given small motor application. This was compared to a same sized induction motor with an efficiency of just under 11%.

The laminations used gave back emf waveforms that had sections where the magnitudes were near to zero for too great a rotational angle of the rotor. A method of spreading the back emf over a larger angle was needed. The DSVRM motor used a displaced stack theory to improve the shape of the back emf waveform. By altering the angular displacement between the two stacks, the back emf was made more 'square' which in turn improved smoothness of torque production and allowed quieter operation to be possible. The back emf was spread over a wider rotational angle so its magnitude was larger over that angle. The back emf in the PMFSM was manipulated in a similar manner as the rotor was skewed. Skewing, like the angular displacement theory, spread the back emf over a larger angle. Both methods reduced the peak value of back emf. The PMFSM appeared to have a better back emf waveshape when the rotor was skewed by having the rotor as two half sections altered by the skewed angle. It was analogous to the displaced stator stack of the DSVRM.

Skewing the rotor and displaced stack theory both were used to reduce the effects of cogging torque. The dc energised field windings and permanent magnets gave continuous flux. This flux pulled the rotor into aligned positions. Flux from the armature had to overcome this to turn the rotor. Cogging was notable as the rotor turned. It gave a torque that was not smooth. Thus cogging was an undesirable feature of machines with a dc flux source. Skewing and stack displacement theory were used for this purpose.

The PMFSM could be made with skewed magnet sections to aid as a in smooth torque production. The magnets could also be placed in an internal stator (external rotor) design. An external rotor design could also be possible with the toroidal field winding but would require a much different motor shape than shown in this thesis.

7.4 PERMANENT MAGNETS AND EXCITATION COIL WINDINGS

All three machines required a pulsed signal to the armature windings to obtain maximal efficiencies. Flux production was best in the DSVRM and SSVRM when unipolar excitation with single coil per pole windings were used as opposed to bipolar excitation with bifilar windings. The PMFSM used bifilar windings in a consequent winding formation as they allowed a simpler winding configuration to be inserted.

The field windings (as with any coil) concentrated flux axially along the middle of the coil – the more ampere turns there was, the greater the flux produced. The motor shape had to accommodate the winding coils of the field winding and the armature coils. End windings accounted for lost flux which was not a problem with magnet use. The field winding acts as an inductor to store magnetic energy, especially when used in a current source half bridge inverter circuit.

The dc current field coils in the single and dual stack variable reluctance machines produce a constant source of additional flux in the axial plane. This flux production requires an input energy that in turn has I^2R copper losses. The use of magnets provides additional flux but without any input power being required (and hence no I^2R copper losses).

The use of permanent magnets means that there is always a source of flux whereas the field windings could be switched off. This imposes safety issues when the motor excitation is switched off but the rotor is still turning (a back emf would still be present). The magnetic flux from the magnets comes from the magnet material itself – the dimensions and B-H curves determine the ampere-turn equivalence and hence flux magnitude from the magnet. The magnets are part of the complete magnetic circuit but they increase the reluctance of the magnetic circuit (magnets are often regarded as virtual air gaps). This allows the magnet shape to be altered to aid the flux paths. Powdered magnets are used for three dimensional shapes. When magnets are used there are two problems to be reviewed during the design process. The first is that of demagnetization, in either a fault condition in the armature windings or at startup, where the back emf does not limit the current from the supply voltage. This was checked with Finite Element Analysis. The second is that the magnets limited the maximum flux within the motor – it was preferable to have lower flux densities within the motor. It coincided with a fact that lower flux densities allow smaller magnets to be used. Magnets cost more than copper hence less magnet material made the overall motor cost less (rare earth magnets invariably cost far more than ferrites but there are suggestions that the cost per tonne and the difference in costs per tonne is falling [109]).

It was noted that the slot area for the bifilar armature windings was not large enough for enough turns for highly efficient higher voltage applications. It was possible that the ferrite magnets could be replaced by rare earth magnets such as Neodymium Iron Boron (NdFeB) as the flux from the stronger magnets would give more back emf.

7.5 AXIAL AND RADIAL FLUX PATHS

In addition to the air gap between the stator and rotor lamination there are two air gaps within the flux return path that posed interesting design issues within the Single Stack Variable Reluctance Machine (*Figure 7.5*). The first is the radial airgap whose function is to provide a continuation of the flux paths between the cylindrical section of the rotor and the housing of the field. Its purpose is for a return path and not for torque production. It is therefore regarded as a parasitic air gap.

This air gap is very small to minimise its reluctance but a mechanical clearance has to be retained. Its presence increases the equivalent magnetic reluctance for the machine and hence more ampere turns would be required for flux production. A gap of 0.2mm was chosen for the machine.

There is also a gap between the end face of the rotor and the field housing (*Figure 7.5*). If this was too small a large axial force (end thrust) would be produced. This axial gap was chosen to be 10 times greater than the radial gap (i.e. 2mm) to minimise this problem.

This end thrust effect could be used to advantage however. Ordinarily this effect is detrimental to the machine as bearings are not designed for axial forces. Some loads have an effect of pulling or pushing the rotor in an axial direction e.g. fan blades moving air. If the load were connected correctly, the end thrust produced by the machine could be used to reduce or cancel the axial forces caused by the load. This cancellation technique has to allow for the fact that, at start up, the axial load forces may be minimal and so the machine would have to be designed especially for such a use.

The problem of having an axial flux path that may not be of constant flux was that of eddy currents. Eddy currents flowed circumferentially to produce a flux in opposition to the change in axial flux. It was therefore of note that electrical insulator spacers may be inserted in the axial plane to prevent circumferential eddy currents.

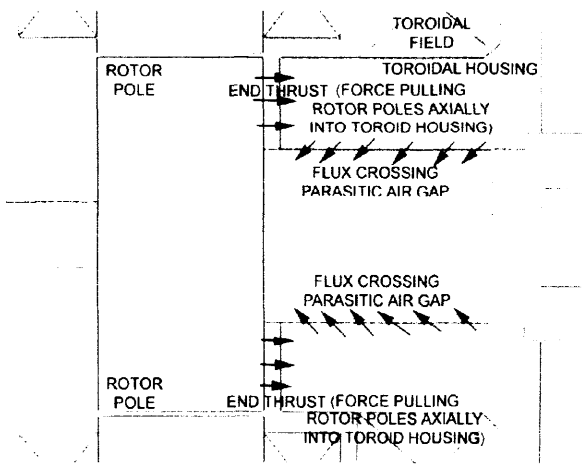


Figure 7.5 :- The parasitic air gap and the production of end thrust.

7.6 BIPOLAR AND UNIPOLAR EXCITATION

From the test results, bipolar excitation with bifilar windings did not give the required efficiency when operating the machine. Flux produced in the stator poles in the unaligned pole pair for that phase energisation was not utilized fully for torque production. Instead it was partially used in preventing of flux leakage through the poles. The bipolar excitation with single coil per pole windings, from the manner in which the coils are connected, could produce more flux but would require the more complex H-bridge power electronics circuit.

The unipolar excitation with single coil per pole windings version, by having only two of the stator poles in each stack energised at any one time, had more flux available for torque production. Such a circuit required a floating gate drive (asymmetric half bridge converter) which complicated the circuitry required.

7.7 SUMMARY

Provided that the reluctance machine is designed properly, ideally with three dimensional Finite Element Analysis, and that the comments in this chapter are taken into account, it should be possible to produce variable reluctance machines with dc assisted toroidal field windings that are of noteworthy efficiency.

There are design considerations that should be noted for improved designs of Permanent Magnet Flux Switching Motors. The main improvement required would be that of higher efficiency at higher supply voltages. The use of magnets can give efficiency boosts to the design of small motors.

Both field windings and magnets reduce the switching losses in the machines as a continuous source of flux is provided.

There is potential for a hybrid motor that has field windings and permanent magnets. Such a motor uses the field winding to provide additional flux for torque production but it could be used as a flux weakening device such that the motor could operate at higher speeds. Perhaps hybrid versions of the Single and Dual Stack Variable Reluctance Machines and the Permanent Magnet Flux Switching Motor would be of worthy investigation for future research.

CHAPTER 8

CONCLUSIONS AND FUTURE WORK

8.1 CONCLUSIONS

Reluctance drives with flux assistance have been shown to offer advantages in terms of motor performance and power electronic ratings. The use of field windings and permanent magnets may be used beneficially and without any notable increase in overall product cost.

Toroidal field windings in single and dual stack variable reluctance drives are very effective sources of axial flux. The machines were shown to prove the concept of additional flux to aid in the production of reluctance torque. The use of switched reluctance laminations were shown to give increased iron losses that did not allow the potential efficiency benefits of such a machine to be fully evaluated. The three dimensional flux paths of such machines are better suited to powdered iron technology. Iron losses would be lowered, the three dimensional flux paths would be improved as there would no longer be laminated sheets used (a particular problem was that of the laminated rotor) and eddy currents would be practically eliminated. As the machine laminations were originally designed for a conventional switched reluctance motor without a requirement for a dc flux path, both machines had peaky, almost triangular back emf waveforms that were due to small pole arcs being used in the laminations. An improved lamination shape would have aided torque production.

For both the variable reluctance machines, it was shown that bipolar exciation with bifilar windings did not give as great an efficiency as the unipolar excitation with single coil per pole. The method by which flux is produced by the armature windings has a noticeable effect of torque capability. It was also very apparent that the field windings should be placed in series with the armature windings. As the load increased an increase in field mmf improved the efficiency.

The single stack variable reluctance machine has parasitic air gaps due to a cylindrical rotor section and also has an effect called end thrust due to axial magnetic forces across an axial air gap. It was proposed that the latter effect could have been used to compensate for axial forces from rotor impellars used in fan applications.

The dual stack variable reluctance machine was successfully shown to have improved torque production by changing the mechanical angular displacement between the two stacks such that the back emfs components from each stack for each phase could be phase shifted. The net result was an ability to alter the displacement angle to give a squarer back emf (by considering the waveshape plus the crest factor). This gave a squarer current waveform with a small current ripple and so torque production was smoother with a noticeably quieter level of operation.

The permanent magnet flux switching machine has a novel design. It is an innovative use of permanent magnets in the stator of a rotating machine. The permanent magnet flux switching motor has different flux paths compared to machines with magnets inserted in the stator, such as, for example, those by Lipo [14, 15, 17, 19, 22] (for example, in the aligned position of [14, 15], flux enters the rotor and travels radially across to the diametrically opposite rotor pole and exits out of that pole, which is different to that of the PMFSM).

The field windings and back iron behind the their slots have been replaced with cheap ferrite magnets in the Permanent Magnet Flux Switching Motor. The source of flux without any need for input power allowed a method for magnetising the motor to a level such that more flux produced by the armature coils could be used in torque production. Magnetising losses were reduced. This was the basis for an efficiency gain for a small motor for a fan application.

The Permanent Magnet Flux Switching Motor operated at very low temperatures that has allowed the possibility of using cheaper lower temperature thermoplastics for the fan housing. It is proposed that the cost savings in plastics could offset the cost addition of the magnets hence a more expensive Permanent Magnet Flux Switching Motor would not necessarily affect the overall cost of the final product.

The motor was compared with similarly priced fans commercially available from Domus and Vent-Axia. The price was not a condition for the comparison but it has been an added bonus that the improvements have been achieved without excessively expensive magnets and electronics etc. The Permanent Magnet Flux Switching Motor was shown to not only be capable of moving a much higher air volume per hour than its competitors but it could also operate at a fraction of the input power of induction motors and less than permanent magnet brushless dc motors. It operated at a speed of 2500rpm with only 2.1W of input power, at 81% efficiency and well below 30°C operating temperature. It was only a fraction louder (2.2dBA louder) than the reputedly quiet commercially manufactured induction motor fan (45.0dBA) (47.2dBA c.f. 45.0dBA). An improved construction would have made this difference even less noticeable.

The Permanent Magnet Flux Switching Motor was shown to be able to comfortably outperform its competitors when the armature commutation timings were optimised. This was possible by using the prototyping circuit board and PIC program developed for this thesis. For all the machines designed, built and tested, the optimum armature timings were effectively found for any test condition by changing the timings as the motors were still operating. Fast and effective prototyping optimisation has been demonstrated for two phase reluctance machines. The only problem may have been that of no ability to PWM each phase. PWM may be required for additional current control to allow the machine to be drive a range of loads, increasing the potential of the machine in commercial products. PWM could be introduced to future improvements.

A low supply voltage design was highly efficient at 81%. The higher supply voltages to the armature did lessen the efficiency as there were not enough armature turns.

The torque production in the Permanent Magnet Flux Switching Motor was improved by increasing the rotational angle for which the back emf was a large value. This was achieved through a skewing of the rotor in a similar manner to the dual stack angular displacement.

The Permanent Magnet Flux Switching Motor has shown significant commercial potential and could be used for a wider range of applications than the existing induction motor plus some of the larger fan applications (e.g. 6 and 9 inch fans). The very low power ability and its inherent electronic control has attracted interest from Domus Ventilation Ltd. and Vent Axia Ltd. who are looking at investing in furthering its development.

It has been shown that the axial flux from the toroidal field windings and the circumferential flux from the ferrite magnets have been implemented in three novel geometries, all of which have shown much potential for improving torque production and lowering power electronic rating requirements for the armature windings. Low input power, high efficiency designs may be attainable if correct design procedures are followed. Three dimensional Finite Element Analysis is a preferred option for some design topologies but, as shown, is not essential.

The reluctance machines with flux assistance that have been built and tested were all successful proof-of-concept machines. The variable reluctance machines would be better if redesigned with optimum design shapes and improved material selection rather than using existing switched reluctance steel laminations. The author believes the Permanent Magnet Flux Switching Motor has the greatest commercial potential and only requires minor alterations for a commercial package.

8.2 RECOMMENDED WORK FOR FUTURE

It was evident that the iron losses in the two variable reluctance machines inhibited the potential efficiencies of the machines and that this was attributed to the use of laminated steel affecting the three-dimensional flux paths. Powdered iron is suggested as an alternative for such machines as complex flux paths are allowed and eddy currents can be minimized. The use of powdered iron technology such as Somaloy500 is recommended for future designs. The advantages it offers for the variable reluctance machines outweigh the reduction in maximum flux available through a lower permeability. The ability to design with complex three dimensional shapes could also be used to advantage with three dimensional flux paths.

Permanent magnets used in the variable reluctance machines to replace the toroidal field winding may be a worthy investigation. It does cause problems regarding the reluctance of the circuit and the flux paths altering due to magnet use so a direct comparison would be difficult.

An external rotor version of the permanent magnet flux switching motor is definitely warranted. By being an internal stator, copper usage is lessened and the motor can be made more compactly. Efficiencies above 81% could be realized. More work needs to be carried out on the effects of current in the armature winding sections adjacent to the magnets to verify the effects of flux from the armature on the magnets. In a manufacturing sense, powdered iron would be useful in the permanent magnet flux switching motor as simpler manufacturing may be possible.

Perhaps hybrid versions of the single and dual stack variable reluctance machines and the Permanent Magnet Flux Switching Motor with field and permanent magnets together may be of worthy investigation for future research to give controllable additional dc flux.

The use of three-dimensional Finite Element Analysis is a must where three dimensional flux paths are present. It allows a clearer insight into the paths within complex designs and may reveal where any problems exist before such a motor is built.

8.3 CONTRIBUTION TO KNOWLEDGE

C. E. Niehoff & Co. Ltd [110, 111] design and manufacture heavy duty brushless alternators. One of the range of products is the Dual Stator Brushless Alternator. The field and stator windings are stationary and no magnets are used. It is alleged that the design is similar to that of the Dual Stack Variable Reluctance Machine (DSVRM). It has been claimed that a similar design exists from this company for a military specification. There has been no contact from the company regarding pictorial evidence of this claim as only a commercial catalogue without diagrams has been forwarded. A search for patents by the company did not provide any further information of relevance. It would be a fair assumption that the alternator has many more pole numbers and that it does not have any Switched Reluctance laminations. The alternator is not being used as a motor either. Any similarities are only regarded as that of a dual stator concept. Unless proven otherwise the DSVRM will be regarded as a novel design concept with no known similarity except as that of being based upon the design concept of the homopolar inductor alternator. The internet address [111] for C. E. Niehoff & Co. Ltd does not offer any additional information to the product catalogue [110] supplied by the company.

8.4 SUMMARY OF CONTRIBUTION BY AUTHOR

The author has presented the design, build and test of two reluctance machines that use field windings and one reluctance machine that uses permanent magnets to give additional dc flux via novel geometries.

The additional flux has been shown to help reduce switching losses and to reduce the ampere turns needed in the armature coils for torque producing flux. The toroidal field windings in dual and single stack variable reluctance machines are presented as switched reluctance variants of the homopolar inductor alternator topology. The field winding of the flux switching motor has been replaced with permanent magnets. The use of magnets or separately energised dc field windings has removed the field energy through the power electronic converter, lowering the kVA/KW rating of the power electronics.

The Dual Stack and Single Stack Variable Reluctance Machines can be designed using a dc motor equivalence and by calculating the magnetic reluctance of the motor by two dimensional analysis. The appropriate choice of winding type for armature flux production in these machines is very important.

The Permanent Magnet Flux Switching Motor has been built as a low power, highly efficient motor for a fan application. The motor is more efficient than existing similarly sized machines used by the Fan Industry, such as, for example, the single phase shaded pole induction motor. It has been shown that the use of permanent magnets does not necessarily increase the overall cost of the application for the motor, as efficiency gains through lowered armature copper losses allows cheaper thermoplastics to be used in the build of the complete application containing the motor (as the PMFSM ran at a low operating temperature). Reluctance drives with flux assistance are shown to achieve efficiencies of over 80%, demonstrating improved performance over similarly sized commercially available small motors.

This thesis shows a novel microprocessor program and its circuitry for optimizing the control of the three prototype machines.

This thesis shows methods for improving torque production and reducing power electronic converter ratings in reluctance machines with flux assistance through appropriate choice of winding type, mechanical angular displacement and by skewing.

Reluctance machines with flux assistance are ideal candidates for experimental designs using powdered iron to allow three dimensional flux paths whilst reducing eddy currents.

BIBLIOGRAPHY

[1] "Permanent Magnets in Alternators and Special Drives", H. R. Bolton, First U.K. Conference on the Manufacture and Application of Permanent Magnets, London, 1980, pp101-110.

[2] "Electronic Control of Switched Reluctance Motors", T. J. E. Miller, Newnes, 2001.

[3] "Switched Reluctance Motors and Their Control", T. J. E. Miller, Oxford University Press, 1993.

[4] "Generating with the Switched Reluctance Motor", A. Radun, APEC IEEE,Volume 1, 1994, pp41-47.

[5] "A Novel High Speed Switched Reluctance Generator", W. F. Ray, M. T. Ebrahim, EPE, 1995, pp3.811-3.815.

[6] "Control of a Switched Reluctance Aircraft Engine Starter-Generator Over a Wide Speed Range", S. R. MacMinn, J. W. Sember, Proceedings IECEC 89 (Washington DC), Aug6-11, 1989.

[7] "Performance of a High-Speed Switched Reluctance Starter/Generator System Using Electronic Position Sensing", S. R. Jones, B. T. Drager, IEEE IAS 1995, Volume 1, pp249-253.

[8] "Design Considerations and Test Results for a Doubly Salient PM Motor with Flux Control", F. Leonardi, T. Matsuo, Y. Li, T. A. Lipo, P. McCleer, IEEE IAS 1996, pp458-463.

[9] "Axial flux circumferential current permanent magnet (AFCC) Machine", J. Luo, D. Qin, T. Lipo, S. Li, S. Huang, 0-7803-4943-1/98, IEEE, 1998.

[10] "Switched reluctance motor with dc assisted excitation", Y Li, J.D. Lloyd, G.E. Horst, IEEE IAS Annual Meeting Conference 1996, pp. 801-807.

[11] "Variable reluctance drive system", T Lipo, F. Liang, Electric Power Research Institute, United States Patent Number 5,459,385, Oct. 17, 1995.

[12] "Reluctance Machine with Auxiliary Field Excitations", Y. Li, United States Patent 5,866,964, 2 February, 1999.

[13] "New winding configurations for doubly salient reluctance machines", B. C. Mecrow, IEEE Transactions on Industry Applications, Vol. 32, No.6, November/December 1996 pp 1348-1356.

[14] "A Novel Doubly Salient Single Phase Permanent Magnet Generator", B. Sarlioglu, Y. Zhao, T. A. Lipo, IEEE IAS Annual Meeting 1994, pp9-15.

[15] "Nonlinear Modeling and Simulation of Single Phase Doubly Salient Permanent Magnet Generator", B. Sarlioglu, T. A. Lipo, IEEE IAS Annual Meeting 1998, pp18-26.

[16] "A Novel Permanent Magnet Motor with Doubly Salient Structure", Y. Liao, F. Liang, T. A. Lipo, IEEE IAS Annual Meeting 1992, pp308-314.

[17] "Self Excited Variable Reluctance Generator", T. Matsuo, J. Luo, E. Hoffman, T. A. Lipo, IEEE IAS Annual Meeting 1997, pp653-660.

[18] "Permanent-Magnet, Direct Drive, Starter/Alternator Machine with Weakened Flux Linkage for Constant-Power Operation Over Extremely Wide Speed Range", F. Caricchi, F. Crescimbini, F. G. Capponi, L. Solero, IEEE IAS Annual Meeting 2001, pp1626-1633.

[19] "Design Considerations and Test Results for a Doubly Salient PM Motor with Flux Control", F. Leonardi, T. Matsuo, Y. Li, T. A. Lipo, P. McCleer, IEEE IAS 1996, pp458-463.

[20] "New Advanced Control Methods for Doubly Salient Permanent Magnet Motor", F. Blaabjerg, L. Christensen, P. O. Rasmussen, L. Oestergaard, P. Pederson, IEEE IAS 1995, pp222-230.

[21] "Static Characteristics of a New Doubly Salient Permanent Magnet Motor", M. Cheng, K. T. Chau, C. C. Chan, IEEE IAS 1999, pp222-230.

[22] "A Permanent Magnet AC Machine Structure with True Field Weakening Capability", A. Shakal, Y. Liao, T. A. Lipo, Electric Machines and Power Systems, 24:497-509, 1996.

[23] "Characteristics of a Doubly Salient-Pole Homopolar Machine in a Constant-Power Speed Range", N. Sugitani, A. Chiba, T. Fukao, IEEE,0-7803-4943-1/98, 1998, pp 663-670.

[24] "The Flux-Reversal Machine : A New Brushless Doubly-Salient Permanent-Magnet Machine", R. P. Deodhar, S. Andersson, I. Boldea, T. J. E. Miller, IEEE IAS 1996, pp786-793.

[25] "Two New Competitors Advanced motors targeted for battery-powered applications", P. J. McCleer, Appliance Manufacturer, July 1998, pp39-41.

[26] "A new doubly salient permanent magnet motor for adjustable speed drives", Y. Liao, T. A. Lipo, Proc. Symp. On Power Electronics, Electrical Drives and Advanced Electrical Motors, 1992, pp415-420.

[27] "A New Electronically-Commutated Doubly-Salient Permanent-Magnet Small Motor", M. M. Radulescu, C. Martis, K. Biro, IEE Electric Machines and Drives, Conference Publication No. 412, 1995, pp213-216.

[28] "Optimum Design of Highly Efficient Magnet Assisted Reluctance Motor", H. Murakami, Y. Honda, Y. Sadanaga, Y. Ikkai, S. Morimoto, Y. Takeda, IEEE IAS 2001, pp2296-2301.

[29] "Highly Efficient Brushless Motor Design for an Air-conditioner of the Next Generation 42V Vehicle", H. Murakami, H. Kataoka, Y. Honda, S. Morimoto, Y. Takeda, IEEE IAS Annual Conference 2001, pp 461-466.

[30] "Low-Cost Compact Permanent Magnet Machine for Adjustable-Speed Pump Application", F. Caricchi, F. Crescembini, O. Honorati, IEEE IAS 1998, pp109-116.

[31] "Design, development and testing of a high-speed axial-flux permanent-magnet machine", F. Sahin, A. M. Tuckey, A. J. A. Vandenput, IEEE IAS 2001, pp1640-1647.

[32] "Multi-Stage Axial-Flux PM Machine for Wheel Direct Drive", F. Caricchi, F. Crescembini, F. Mezzetti, E. Santini, IEEE IAS 1995, pp679-684.

[33] "Design and Experiments of A Novel Axial Flux Circumferential Current Permanent Magnet (AFCC) Machine with Radial Airgap", J. Luo, S. Huang, S. Chen, T. A. Lipo, IEEE IAS 2001, pp 1989-1996.

[34] "A Comparison Between the Axial Flux and the Radial Flux Structures for PM Synchronous Motors", A. Cavagnino, M. Lazzari, F. Profumo, A. Tenconi, IEEE IAS Annual Conference 2001, pp 1611-1618.

[35] "TORUS Concept Machines: Pre-Prototyping Design Assessment for Two Major Topologies", S. Huang, M. Aydin, T. A. Lipo, IEEE IAS Annual Conference 2001, pp 1619-1625.

[36] "STEPPING MOTORS: a guide to modern theory and practice", P. P. Acarnley, IEE Control Engineering Series 19, Second Edition, 1984, Peter Peregrinus Ltd.

[37] "Instantaneous Torque Analysis of Hybrid Stepping Motor", N. Matsui, M. Nakamura, T. Kosaka, IEEE IAS 1995, Volume 1, pp 701-706.

[38] "An overview of the powder processing of soft magnetic composites", G. Roberts, A. Davidson, S. Gair, J. Hajto, Engineering Science and Education Journal, December 2001, IEE, pp237-240.

[39] "Switched Reluctance Machines with Simple Hoop Windings", H. M. Amreiz, B. C. Mecrow, submitted for PEMD, 2002.

[40] "Transverse Flux Reluctance Motor for Direct Servodrive Applications", R. Kruse, G. Pfaff, C. Pfeiffer, IEEE IAS 1998, pp 655-661.

[41] "The Electrical Encyclopedia A handbook of modern electrical practice for the working electrician", Volume II DEC-IRON, S. G. Blaxland Stubbs, 1934, The Waverley Book Company, pp 670-671.

[42] "The Performance and Design of Alternating Current Machines transformers, three-phase inductions motors and synchronous machines", Third Edition, M. G. Say, 1958,.Sir Isaac Pitman & Sons, pp514-515.

[43] "Alternating Current Machines", Fourth Edition, M. G. Say, 1976, Pitman Publishing, Chapter 12.

[44] "Electrical Machines", Second Edition, A. Draper, Longman, 1967, pp 205-208.

[45] "The Theory of the Inductor Alternator", J. H. Walker, IEE Volume 89, Part 2, 1942, pp 227-228.

[46] "The New Electrical Encyclopedia A practical handbook of modern electrical engineering for working engineers", Volume II DA-IS, S. G. Blaxland Stubbs, 1953, The Waverley Book Company, pp 834-836.

[47] "Electromagnetic and Electromechanical Machines", L. W. Matsch, Intext, 1972, pp 449-455.

[48] "Reluctance Models for Mechatronic System Simulation", Dr.-Ing. Karsten Schonherr, Robert Bosch, 06/09/2002.

[49] http://www.annauniv.edu/prodetails1.htm 09/04/2003

[50] "Low Cost Brushless Generators", L. McGrow, C. Pollock, IEEE IAS Annual Conference 1999, pp 1229-1236.

[51] "Performance Prediction of Law's Relay Actuator", H. R. Bolton and Y. Shakweh, IEE Proceedings, Vol 137, Part B: Electric Power Applications, pp 1-13, Jan 1999.

[52] "The Flux Switching Motor, a DC Motor with Magnets or Brushes", C.Pollock and M. Wallace, Proceedings of the IEEE Industrial Application Society 34th Annual Meeting", Vol. 3, pp 1980-1987, 1999.

[53] "Electric Machinery Fundamentals", Second Edition, Stephen Chapman, M^cGraw-Hill, 1991.

[54] "NASA Final Report - Brushless Rotating Electrical Generators for Space Auxiliary Power Systems", J. N. Ellis, F. A. Collins, National Aeronautics and Space Administration, 26 April, 1965, Contract No. NAS 3-2783.

[55] "Reduction of Cogging Torque in Spindle Motors for CD-ROM Drive", C. C. Hwang, S. B. John, S. S. Wu, IEEE Transactions on Magnetics, Volume 34, No. 2, March 1998, pp468-470.

[56] "Reduction of Cogging Torque in Permanent Magnet Motors", T. Li, G. Slemon, IEEE Transactions on Magnetics, Volume 24, No. 6, March 1998, pp2901-2903.

[57] "Torque and Cogging Torque in Sandwich Type CD-ROM Spindle Motor", C. M. Chao, S. J. Wang, C. P. Liao, D. R. Huang, T. F. Ying, IEEE Transactions on Magnetics, Volume 34, No. 2, March 1998, pp471-473.

[58] "Cogging Torque Reduction of a Single-Phase Brushless DC Motor", D. Huang, T. Ying, S. Wang, C. Zhou, IEEE Transactions on Magnetics, Volume 34, No. 4, March 1998, 2075-2077.

[59] "Acoustic noise and vibration of rotating electric machines", A. J. Ellison, C. J. Moore, Proceedings of the IEE, Volume 115, 1968, pp1633-1640.

[60] "Electronic Cancellation of Acoustic Noise in a Fan System Driven By a Switched Reluctance Motor", M. Brackley, C. Pollock, EPE '99.

[61] "Acoustics for Engineers", J. D. Turner, A. J. Pretlove, Macmillan Education Ltd, 1991.

[62] "The effect of skewing and pole spacing on magnetic noise in electric machinery", S. J. Makina, Transaction ASME, 1934, 5, pp711-720.

[63] "The Origin and Reduction of Acoustic Noise in Doubly Salient Variable-Reluctance Motors", D. E. Cameron, J. H. Lang, S. D. Umans, IEEE Transactions on Industry Applications, Volume 28, No. 6, November/December 1992, pp1250-1255.

[64] "The Origin of Permanent Magnet Induced Vibration in Electric Machines", B. S. Rahman, D. K. Lieu, Transaction of the ASME, Journal of Vibration and Acoustics, Volume 113, October 1991, pp476-481.

[65] "Analysis and Reduction of Acoustic Noise from a Brushless D.C. Drive", M. Brackley, C. Pollock, IEEE IAS 1999, pp888-893.

[66] "Time Domain Analysis of Vibration and Acoustic Noise in the Switched Reluctance Drive", C. Y. Wu, C. Pollock, Conference Record, 6^{th} International Conference on Electrical Machines and Drives, IEE, 1993, pp558-563.

[67] "Analysis and Reduction of Vibration and Acoustic Noise in the Switched Reluctance Drive", C. Y. Wu, C. Pollock, IEEE Transactions on Industry Applications, Volume 31, No. 1, January/February 1995, pp91-98.

[68] "Acoustic Noise Cancellation Techniques for Switched Reluctance Drives", C. Y. Wu, C. Pollock, IEEE IAS 1995, Volume 2, pp448-455.

[69] "Reduction of Noise and Vibration in Switched Reluctance Motors : New Aspects", A. Michaelides, C. Pollock, IEEE IAS 1996, Volume 2, pp771-778.

[70] "Comparison of the Acoustic Noise of A Flux Switching and a Switched Reluctance Drive", M. Brackley, C. Pollock, IEEE IAS 2001, pp2089-2096.

[71] http://www.hoganas.com.

[72] "SMC update No1 1997", Hoganas, http://www.hoganas.com, 1997.

[73] "Permanent Magnet Machines with Powdered Iron Cores and Pre-Pressed Windings", A. G. Jack, B. C. Mecrow, P. G. Dickinson, D. Stephenson, J. S. Burdess, J. N. Fawcett, T. Evans, IEEE IAS Annual Meeting, 1999, pp97-103.

[74] "SMC update No1 1999", Hoganas, http://www.hoganas.com, 1999.

[75] "SMC update No1 2000", Hoganas, http://www.hoganas.com, 2000.

[76] "Development of a transverse flus traction motor in a direct drive system", R. Blissenbach, U. Schafer, W. Hackmann, G. Henneberger, http://www.hoganas.com.

[77] http://www.pasci.com/theproduct/release.html, 02/07/2002.

[78] "An Iron Composite Based Switched Reluctance Machine", M. Alakula, T. Cedell, M. Persson, L. Sjoberg, Stockholm Power Tech Conference, IEEE/KTH, 1995, pp251-255.

[79] "Iron Powder in Electrical Machines, Possibilities and Limitations", O. Andersson, Hoganas AB, http://www.hoganas.com, 2001.

[80] "Soft Magnetic Composite Materials – Use for Electrical Machines", A. G. Jack, B.C. Mecrow, A. G. Jack, IEEE, 1999.

[81] "Fundamental of Manufacturing for Engineers", T. F. Waters, Chapter 7, pp173-183, 1996, UCL Press.

[82] "Route from Powder to an Electrical Machine Application", O. Krogen, P. Jansson, Powder Metallurgy World Congress & Exhibition, 1998, Proceedings Vol5, 520-525.

[83] "Iron loss in Machines with Powdered Iron Stators", M. Persson, P. Jansson, A. G. Jack, A. G. Jack, B.C. Mecrow, Electric Machines and Drives 7^{th} International Conference, IEE, 1995, pp242-246.

[84] "Iron-Resin Composite Materials for AC Magnetic Applications", C. Gelinas, J. M. Battison, Powder Metallurgy World Congress & Exhibition, 1998, Proceedings Vol5, pp508-513.

[85] "Some Considerations about Soft Magnetic Materials for AC Applications", E. Enescu, E. Soptea, O. Filip, S. Gavriliu, E. Cristescu, Powder Metallurgy World Congress & Exhibition, 1998, Proceedings Vol5, pp600-605.

[86] "Recyclability and Life Cycle Environmental Loading of PM Products", K. Halada, K. Minagawa, K. Ijima, Powder Metallurgy World Congress & Exhibition, 1998, Proceedings Vol3, pp331-336.

[87] "The Building Regulations 1991 Approved Document F", The Department of the Environment and The Welsh Office, HMSO, 1995 Edition.

[88] "Domus Modular Ventilation Systems Extractor Fan Catalogue 2001...the complete solution", Domus Ventilation Limited, http://www.domusventilation.com, 2001.

[89] "Fluid Mechanics", J. F. Douglas, J. M. Gasiorek, J. A. Swaffield, Fourth Edition, Chapters 22 and 25, 2001, Prentice Hall.

[90] "Air Flow Vs. Pressure Characteristics", http://www.comairrotron.com/Engineering/airflow.htm, 02/15/2002.

[91] "Vent-Axia System Calculator", D. J. Clark, Vent-Axia, http://www.vent-axia.co.uk, 1997.

[92] "Electric power system components–transformers and rotating machines", R. Stein, W. Hunt, 1979,VNR.

[93] "Design and Control of a Variable Reluctance Drive with axially orientated dc flux assistance", E. R. T. Goodier, C. Pollock, IEEE IAS Annual Conference 2001, pp 2065-2072.

[94] "Homopolar Variable Reluctance Machine incorporating an Axial Field Coil", E. R. T. Goodier, C. Pollock, IEEE IAS Annual Conference 2001, pp 1997-2007.

[95] Switched Reluctance Drives: a comparative evaluation, C.Pollock and A. Michaelides, Power Engineering Journal December 1995.

[96] Variable-speed switched reluctance motors, Lawrenson Stephenson Blenkinsop Corda Fulton, IEE proceedings, Volume 127, Part B, Number 4, July 1980.

[97] MICROCHIP DATA BOOK 1994, Microchip Technology Inc.

[98] PIC16F87X DATA SHEET, Microchip Technology Inc.

[99] "Saliency-Based Position Estimation in PM Synchronous Motors", V. Petrovic, A. M. Stankovic, IEEE IAS Annual Conference 2001, pp 801-806.

[100] "Implementation of Sensorless Vector Control for Super-High Speed PMSM of Turbo-Compressor", B. Bae, S. Sul, J. Kwon, J. Shin, IEEE IAS Annual Conference 2001, pp 1203-1209.

[101] "High Performance Current Based Sensorless Speed Servo System of PM Motor Based on Current Estimation", K. Ohishi, Y. Nakamura, IEEE IAS Annual Conference 2001, pp 1240-1246.

[102] "A Sliding Mode Observer for Sensorless Control of Permanent Magnet Synchronous Motors", C. Li, M. Elbuluk, IEEE IAS Annual Conference 2001, pp 1273-1278.

[103] "New Rotor Position Detection for Permanent Magnet Synchronous Machines Using the "Inform"-Method", M. Schrodl, T. Stefan, ETEP Volume 1 No.1, January/February, 1991, pp 47-53.

[104] "High Performance Speed Control of Permanent Magnet Synchronous Motor with Eccentric Load", J. Park, D. Koo, J. Kim, G. Kang, J. Park, IEEE IAS Annual Conference 2001, pp 815-820.

[105] "Speed control of a PMSM motor based on the new disturbance observer", Y. S. Suh, T. W. Chun, IEEE IAS Annual Conference 2001, pp 1319-1323.

[106] "Two-dimensional fields in electrical engineering", L.V. Bewley, 1963, Dover Publications.

[107] "Magnets The Complete Field", Magnet Sales & Services Limited, Kembrey Park, Swindon, Wiltshire, SN2 6UN, 1999, http://www.magnetsales.co.uk.

[108] "Permanent magnet materials – Fundamentals, design and application", IEE Power Division Seminar at the Institute of Civil Engineers, London, 4 July 2000.

[109] "Cost and Performance Trends of Permanent Magnets in Automotive Applications", J. G. W. West, IEE Seminar Permanent Magnet Materials – fundamentals, design and application, Tuesday 4th July, 2000, pp2/1-2/26.

[110] "C. E. Niehoff & Co. Ltd. Brushless Alternators Product Catalogue 2001", C. E. Niehoff & Co. Ltd, 2001.

[111] www.ceniehoff.com, December 2002.

PUBLICATIONS

PUBLISHED WORK

"Design and Control of a Variable Reluctance Drive with axially orientated dc flux assistance", E. R. T. Goodier, C. Pollock, IEEE IAS Annual Conference 2001, pp 2065-2072.

"Homopolar Variable Reluctance Machine incorporating an Axial Field Coil", E. R. T. Goodier, C. Pollock, IEEE IAS Annual Conference 2001, pp 1997-2007.

"Homopolar Variable Reluctance Machine incorporating an Axial Field Coil", E. R. T. Goodier, C. Pollock, Volume 38, Number 6, November/December 2002, IEEE Transactions on Industry Applications, pp 1534-1541.

UNPUBLISHED OR AWAITING FOR SUBMISSION FOR PUBLICATION

Mechanical angular displacement in Dual Stack Variable Reluctance Machines with axially orientated dc flux assistance.

Comparison of excitation techniques in a Single Stack Variable Reluctance Machine with dc assisted axial flux excitation.

Design comparisons of Variable Reluctance Machines with axially orientated dc flux assistance.

SOFTWARE USED

Below is a list of the main software used in the preparation of work for this thesis.

Vector Fields Opera2D Finite Element Analysis
Microsoft Excel 2000
Microsoft Word 2000
Design Science MathType 4
Notetab Lite
Microchip MPLAB
Microchip MPASM
Autodesk AutoSketch 5
JASC PaintShop Pro 4
Adobe Photoshop 4
ACD Systems ACDSee32

APPENDICES

These Appendix sections relate to text in previous chapters. The information contained within each Appendix is aimed to complement the understanding of the reader.

Appendix A introduces cogging torque, acoustic noise cancellation techniques, powdered iron technology and fan industry requirements.

Appendix B shows the flow diagram for the timing control in the PIC16F877 microcontroller. The 16F84 version of this is shown with fixed delays instead of being able to vary them as in thePIC16F877 version. Included is the test procedure for using the prototyping circuitry and some test results demonstrating the optimisation obtained from using the experimental contol electronics to optimise the armature timings.

The PIC16F877 coding for a two speed variable delay length control program is in Appendix C.

The LCD circuit design and LCD display program coding (for a PIC16F877 microcontroller) to give the information obtained from the prototyping circuit is shown in Appendix D.

Data sheet information for Self Bonding Wire (Solderable Self-Bonding Enamel (SALDAVEX AUTOVEX F)), packing factor considerations, the use of plastic formers and the data sheet information for Nylon66 is given in Appendix E.

A Dual Stack Variable Reluctance Machine concept diagram is shown in Appendix F.

Photographs of the Single Stack Variable Reluctance Machine, information relating to a noted bearing imbalance during back emf testing, and optimisation during load testing can be found in Appendix G.

Appendix H discusses magnetic short circuits obtained when investigating maintaining stator integrity in the Permanent Magnet Flux Switching Motor. Also shown are example B-H curves for some of the magnets considered, lamination dimensions, the dynamic modelling spreadsheet, powdered iron discussion, and photographs of the Permanent Magnet Flux Switching Motor.

The author has kept the program coding for the PIC microcontrollers as small text. This was done to minimise the number of pages used in the printing of this thesis.

APPENDIX A

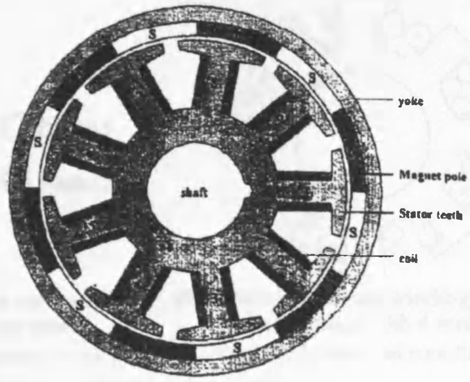
A.1 COGGING TORQUE

Cogging torque arises from interaction of flux from permanent magnets (or from energised field coils) between the rotor and stator poles. Cogging torque can increase the load of the system, increasing power dissipation. Cogging torque produces vibration and noise that may be detrimental to the performance of position and speed control systems [55]. Cogging torque can be reduced by skewing [56] but this can add complexity to the construction (the stator is skewed if the magnets are on the rotor and vice versa).

Skewing can also reduce the mechanical output torque which may be disadvantageous. Surface mounted rotor magnets can have their angular width changed in relation to the rotor tooth plus slot dimension such that the cogging torque may be reduced to 1% of rated torque [56]. A rotational shift of one rotor pole pair relative to a second rotor pole pair for a four-pole rotor design can also reduce the cogging torque [56]. Such techniques do require, for optimum results, the use of finite element analysis.

In the Sandwich Type CD-ROM spindle machine (three phase, 12 poles, 9 slots, radial winding, radial air gap type brushless dc machine) [56] used in optical drives, the large cogging torque increases the load of the servo system hence increases the power dissipation. This is an undesirable effect that can also give dead points where the machine cannot start. It is possible, in this type of machine, to design the cogging torque waveform [57] to prevent dead points. Correct design of the machine can effectively eliminate the dead points as well as reduce cogging torque (*Figure A.1*)[58] by having multiple radii of curvature of the salient poles.

Skewing of the rotor magnets is another suggested method of cogging torque reduction. Variation of the ratio of armature teeth to magnet pole arc can lead to a reduction in cogging torque. This has been implemented in a three-phase, Y-connected permanent magnet brushless DC machine with an external rotor (for use in a spindle machine in a CD-ROM drive).



Motor construction.

Figure A.1 :- Sandwich Type CD-ROM spindle machine [58].

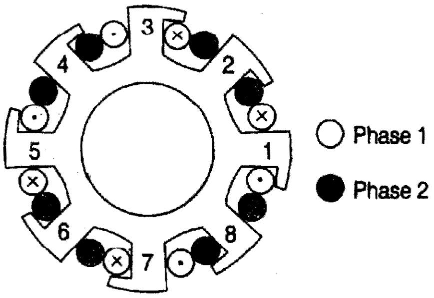
A.2 ACOUSTIC NOISE CANCELLATION TECHNIQUES

The reduction of acoustic noise produced by electrical machines has been reported in many papers. It is known that noise is produced by mechanical, magnetic and aerodynamic sources [59] and methods exist to reduce such noise. It is becoming increasingly more important for motor manufacturers to produce machines that are quieter than their competitors, especially as British and European regulations are becoming stricter with regards to noise emissions. The sound radiated from the machine and the vibration emitted to its supporting structure have to be studied together.

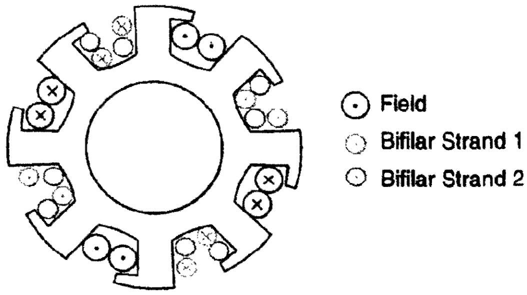
The sound pressure level is only part of the concern for domestic applications. The frequencies at which noise is produced is also of particular interest since the human ear is not uniformly sensitive across the frequency range and that certain frequencies (e.g. pure tones in the human speech frequency range) can create interference and annoyance to individuals [60, 61]. Parts of the driving system can also have techniques applied for a reduction in acoustic noise, such as to reduce the amplitude of resonant frequency of the fan housing [60].

Although there is little work being carried out on acoustic noise cancellation techniques as part of this thesis, the dBA noise levels for fan motors is needed due to low noise levels being a selling point for fan applications. There are many sources of noise and vibration in machines [59]. Vibrational mechanical noises may result from unbalanced rotors, the type of bearing used, shaft curvature, and resonances due to manufacturing asymmetries unbalancing magnetic and mechanical forces.

Magnetic forces in the air gaps account for a degree of mechanical vibration (magnetostriction and flux between laminations give less vibrations). Variation in armature current in each slot and air gap lengths cause the machine to vibrate. Torque ripple may also excite stator vibrations that could emit acoustic noise. Skewing of the rotor and/or stator can reduce vibrational noise [62]. Aerodynamic noise, especially for fan motors, is due to pressure variations and turbulence in the air flow.



Two phase switched reluctance winding on stator with eight salient teeth. (External rotor with 4 teeth not shown)
Figure A.2 :- External rotor switched reluctance motor [70].



Flux switching motor winding on stator with eight salient teeth. (External rotor with 4 teeth not shown)
Figure A.3 :- External rotor flux switching motor [70].

In doubly salient variable reluctance motors the dominant noise source is the radial deformation of the stator due to its radial magnetic attraction to the rotor [63]. It is also suggested that the stator current could interact with the local magnetic field to produce a force on the windings. The force could excite winding vibrations that would emit acoustic noise.

In brushless dc motors where the magnets are on the rotor, a principle source of vibration is the induced travelling forces from the rotating permanent magnets acting on the stator [64]. It is a more serious problem when the forcing frequencies matches one or more of the structural resonant frequencies. The stress amplitude varies with B^2 hence rare earth magnets are more problematic than ferrites.

The acoustic noise from brushless dc motors is similar to that from switched reluctance drives [65]. As such switched reluctance motor acoustic noise analysis and noise cancellation techniques can be found in [65-69]. Radial vibrations of the stator tends to be the dominant source of mechanical vibration and acoustic noise [63]. From time domain analysis, each step change in armature winding applied voltage creates a step change in the gradient of armature current that produces a dampened vibration [65]. Acoustic noise will propagate at the natural mechanical resonance of the motor housing.

It has been demonstrated that the Flux Switching Motor is a quieter machine than the Switched Reluctance Motor (*Figures A.2, A.3*)[70]. Identical lamination, two phase, external rotor machines (except for the winding types required for each type of machine) were compared. Flux control methods between each motor differ considerably (the armature currents orientate the flux paths in the Flux Switching Motor, but generate all the flux in the Switched Reluctance Motor as the Flux Switching Motor has a dc current conducting field winding that is always energised). The Flux Switching Motor provides a smoother transition of flux from one stator pole set to the next as the dc field winding prevents flux being removed with every working stroke. This reduces radial forces that results in a reduction in acoustic noise. It is therefore of importance to consider flux production and flux paths within electrical machines in order to ascertain methods of acoustic noise reduction.

To reduce vibration and acoustic noise, there is a range of different techniques. Techniques include two stage commutation [65, 67, 69], three stage commutation [68], and voltage smoothing [68]. Such controlling techniques can be implemented to accommodate for the design and construction of the motor [65]. Two stage commutation techniques can be realised for the Switched Reluctance Drive with an asymmetric half bridge inverter circuit [68]. Three stage commutation techniques are required for circuits such as the split dc power converter and the capacitor dump power converter. Vibration cancellation can be effective in the switched reluctance motor at any speed [69].

A.3 POWDERED IRON TECHNOLOGY

Metal powders are becoming increasingly common in electric machines. Conventional machines use laminated steel to minimise magnetic field losses but this restricts the magnetic forces to two dimensions (those in the plane of the laminations). Resultant machines therefore risk being overly large, heavy and not as effective as desired. Hoganas [71, 72] are the one of the world leaders in the production of iron and steel powders (they developed the Soft Magnetic Composite, SMC, range). SMC's can be used to develop electrical machines with three-dimensional magnetic fields due to it having an isotropic nature. Such powders hence improve design flexibility. Powder metallurgical methods give minimal material wastes and can be up to 50% more cost effective than conventional lamination production methods. Complex shaped parts with good dimensional accuracy, smooth surfaces and tight tolerances are easily manufactured with powder metallurgy. Eddy current losses are low at higher frequencies due to the powder nature of the material allowing reduction in motor size and weight. Alternating magnetic fields in any direction is allowable. Also, the use of powdered materials allows improvements in recyclability compared to laminated steels (the crushing process allows quick separation of motor components) [69].

An overview of the powder metallurgy process and technological advances can be found in [70]. The paper discusses the merits of the process, the methods of pressing powders, post-pressing operations and overviews some recent developments.

Powder metallurgy is already being used to build machines. Claw pole motors, permanent magnet machines capable of high torque per unit volume, are made with SMC's.

Radial/axial machines employ permanent magnets coupled with electronic commutation. Radial and axial air gaps are utilised to achieve direct drive motor designs with minimal weight. The SMC core carries three-dimensional fields. High torques per unit volume and high efficiencies are achieved with such designs.

The University of Newcastle upon Tyne uses Hoganas Somaloy500 to build permanent magnet servomotors capable of increased torque per unit volume [71, 72]. Lyng Electronics (Norway) produce a compact electric valve actuator (Lyng Eltorque) utilising a transverse flux machine employing SMC materials in the stator core [75]. The Induction Motor represented 80% of total machine production in 2000 [76]. Its design is simple yet sturdy, making it suitable for production using SMC materials.

Somaloy500 [74] is best used at frequencies below 10kHz and is characterised by high induction and lowest core losses below 100Hz. It is best used in electrical machines and ignition systems. Somaloy550 [74] is optimised for frequencies below 400Hz and has a higher maximum permeability (higher torque potential) but is less suitable for thin sections and is not as strong as Somaloy500.

It is claimed [74] that modern rare earth magnets can be better utilised with SMC materials to produce electrical machines with output levels exceeding non-SMC machines of the same size. SMC's have been used to allow improvements in transverse flux machines as three dimensional flux paths are better realised compared to laminations in the radial and axial planes. Somaloy500 gives substantially reduced low-frequency iron losses and an unsaturated relative permeability of over 5000 with a saturation flux density of approximately 2T.

Segway Company produces a self-balancing, electric-powered transportation device which contains a brushless servo motor (the product is named Segway Human Transporter, HT). Danaher Motion, through its Pacific Scientific business, supplies these motors. Amongst its design features, the motor's construction uses a proprietary injection-moulding process to mould key components of the motor and encapsulate the windings in one step [77]. The process simplifies manufacturing and brings the added benefits of higher quality and improved motor performance. The motor, with its new design, produces 40% more torque per unit of volume than comparably sized motors.

Powdered iron technology has been implemented in a 1.5kW 6/4 Switched Reluctance Drives [78]. For the same output power a laminated machine has 85% efficiency compared to 82% for the high-density iron composite version (a comparable performance). Hysteresis losses were shown to be slightly higher than the laminated version but eddy current losses were negligible up to a frequency of 100kHz.

Hoganas have published some of their papers on their internet site at http://www.hoganas.com. One paper describes the use of iron powder in electrical machines [79]. It is shown that, for medium frequency applications, iron powder components can compete with laminates. The material properties of the iron powders can be exploited in particular when considering the methods of compaction. The paper also discusses limitations of powder metallurgy. Eddy current losses increase by a power of two as the frequency of the alternating magnetic field increases (a dynamic loss). Hysteresis losses increase linearly with frequency [79, 80]. Eddy currents can be reduced by use of smaller particle size powders, thicker coating of insulation around the particles with larger amounts of binder/lubricant. This results in components with good high frequency properties but has lower density, lower induction, lower permeability and higher hysteresis losses. High permeability is needed for minimising the magnetising currents required. High magnetic induction is needed for high torque production. Iron or Silicone-Iron laminations have been previous choices for soft magnetic properties. The magnetic properties are good in the plane of the laminates but poor in the direction normal to them. The anisotropic properties limit laminations to a two dimensional magnetic flux pattern. Coated powdered iron gives isotropic properties. Improvements in manufacture of powdered irons have removed the restriction of machine design to laminates with two dimensional flux paths. The exact properties depend on the powder used, the mix, the compaction method, and temperature [81]. Sintering cannot be used on SMC's limiting the strength compared to laminates. Smaller motors are better for SMC's as less magnetic forces tend to be prevalent. Moderate centrifugal forces are the limit for a rotor made with SMC. The isotropic nature of SMC's allows complex three dimensional shapes to be utilised fully for optimum machine performance as flux paths are no longer restricted to two planes. Radial and axial fluxes are accommodated which is very difficult to implement successfully with laminates. The machines can be designed to allow rounded corners of copper windings utilising pole faces and slot area in a manner not realistically possible before. Savings to copper used and copper losses can be implemented. The compaction press limits the size of components manufactured. Close tolerances are achievable and segmenting the machine can allow simpler winding processes to be used. The maximum relative permeability in the plane of the laminate is above 10 times that of a SMC such as Somaloy500 [82] (due to non-magnetic materials constantly present such as coatings, and pores). But, in the third plane, the laminate has poorer permeability then the powdered iron. SMC's are best used where maximum permeability is of less importance. This is of particular note in permanent magnet machines where the magnet sections can be viewed as effective air gaps. Machines with large air gaps may also be potential candidates for SMC's, depending on the application. Laminates tend to have lower hysteresis losses but higher eddy current losses. The frequency of the application determines whether the SMC gives lower losses. In the frequency range of 50 to a few hundred Hz the eddy current loss is higher in SMC components than laminates but the difference is smaller than suggested by material data [80, 83]. The magnetic induction is dependent on the manufacture of the SMC. The permeability affects the saturation at lower magnetic fields. High density SMC have saturation levels comparable to laminates but, below saturation, SMC's are worse due to lower permeability and hence lower magnetic induction. The use of more iron in critical regions can compensate for this. The total cost of SMC tends to be more than that of laminates but the benefits of design and reduced losses make SMC motors a realistic viable alternative.

SMC's can be tailored for individual applications and it is important that the correct material is used for each application [84, 85]. But to make full use of SMC it is important that design packages such as three dimensional Finite Element Analysis (such as by Opera) are utilised to incorporate more complex flux paths within the machine designs. It may be that, with increasing concerns regarding the global environment, the recyclability of SMC's and laminates become a greater issue in the choice of material to manufacture machines with [86].

A.4 FAN INDUSTRY REQUIREMENTS

The Fan Industry has to produce fans that meet The Building Regulations 1991 Approved Document F [87]. For the purposes of this thesis, the regulations of interest are those regarding ventilation of rooms containing openable windows (i.e. located on an external wall) for domestic and non-domestic buildings. For kitchens the extract ventilation rate is 30 litres/second (108m^3/hour) adjacent to a hob or 60 litres/second (216m^3/hour) elsewhere. For utility rooms the extract ventilation rate is 30 litres/second (108m^3/hour). For bathrooms with or without a toilet the extract ventilation rate is 15 litres/second (54m^3/hour). The Domus S1R 100mm axial wall fan [88] extracts at 26.1 litres/second (94m^3/hour) at about 2520rpm. For the same fan blade a new extraction rate can be obtained by altering the rotor speed (in proportion to the change in extraction rate) [89, 90]. Thus 15 litres/second (54m^3/hour) is obtained at about 1448rpm, 30 litres/second (108m^3/hour) at about 2895rpm and 60 litres/second (216m^3/hour) at about 5790rpm. It is however common practice to build fans that exceed these extraction rates since these values are minimum values (the Domus S1R fan extracts at 26.1litres/second rather than 15 litres/second). Issues such as ducting requirements and applications, etc. are not of concern for this thesis, but there are sources of relevant information from fan suppliers such as Vent-Axia [91].

APPENDIX B

B.1 PIC16F877 PROTOTYPING CIRCUIT FLOW CHART

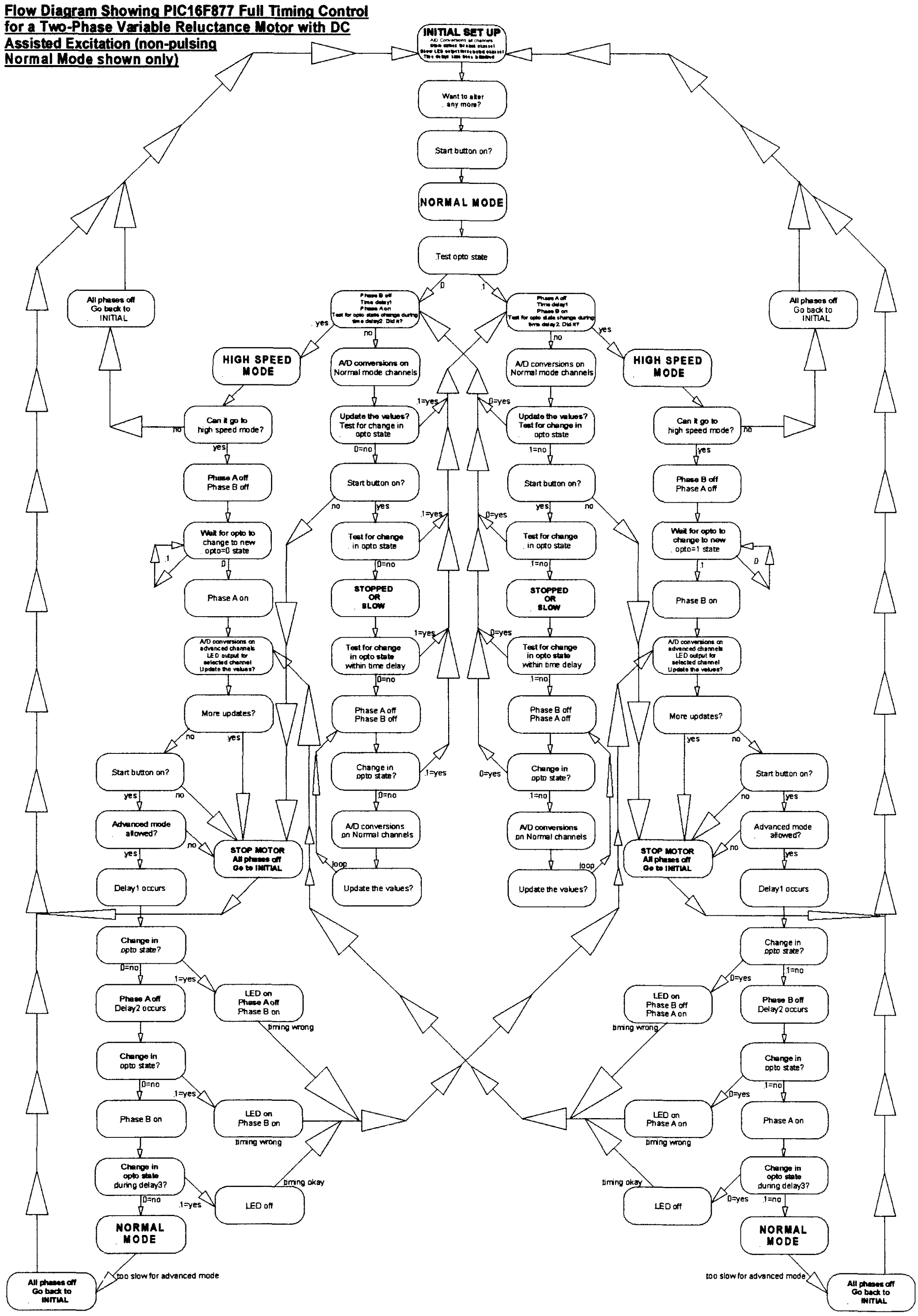


Figure B.1:- PIC program flow chart with A/D conversion allowed.

B.2 PIC16F84 PROTOTYPING CIRCUIT EQUIVALENT FLOW CHART

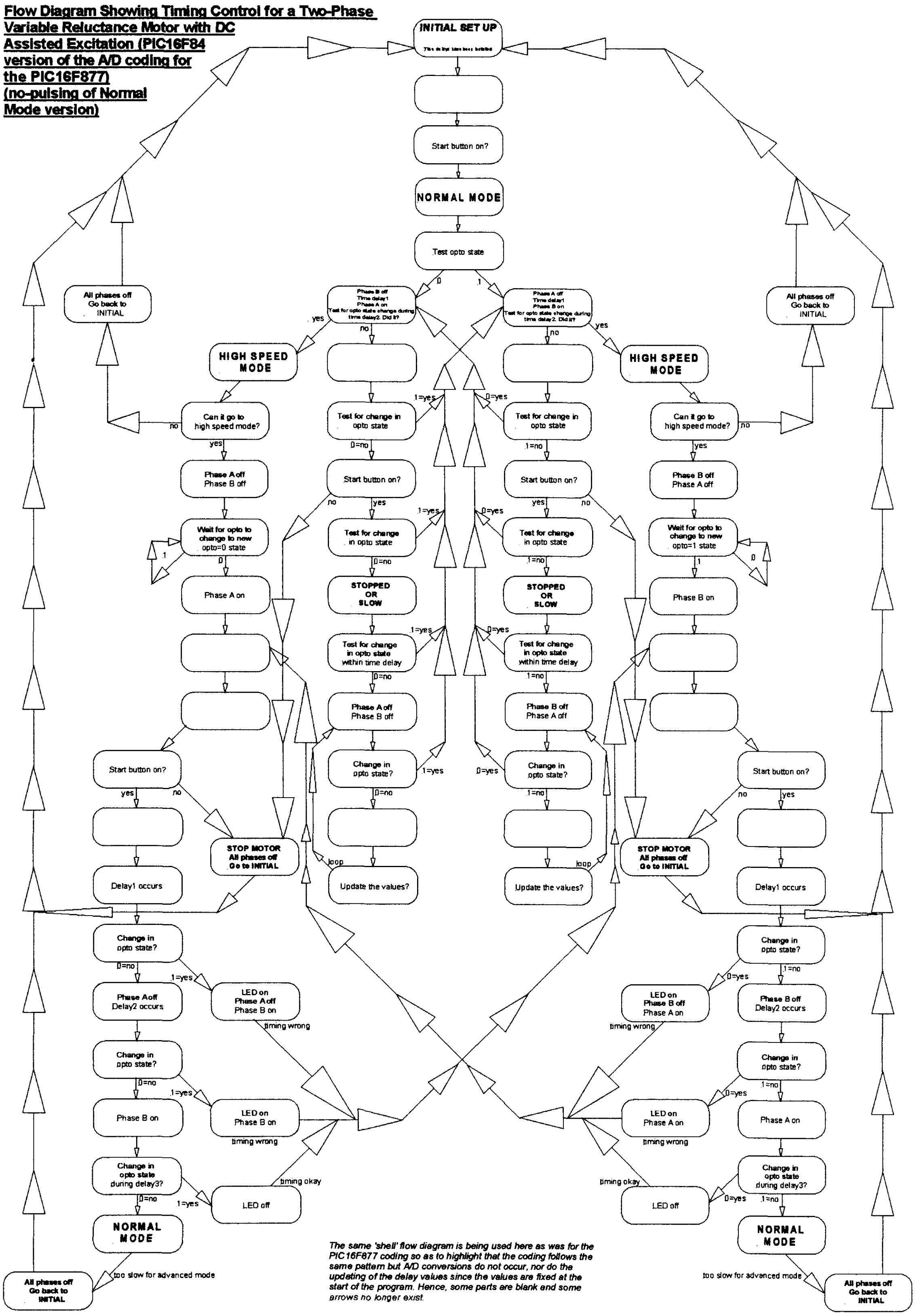


Figure B.2 :- PIC program flow chart with no A/D conversion allowed.

B.3 TEST PROCEDURE USING PROTOTYPING CIRCUITRY

The preferred method for optimisation is as follows : -

- 1. Get the machine running at no load in a chosen Speed Mode at a fixed arbitrarily chosen speed (say 5000rpm), selecting timing delays in that Mode to obtain, by inspection, a timing sequence giving a low input power. Note down the values for the timing delays as shown by the prototyping circuit. The field supply voltage (and hence current) is at a fixed value. Vary the supply to the armature phase windings until 5000rpm (or other chosen speed) is reached again.
- 2. Calculate the field supply power and measure (using a power meter) or calculate the armature input power supplying the phase windings (which includes power electronic circuitry power drain). The sum of these is the total input power.
- 3. For each Speed Mode there are variables that alter the timings of the phase firings. For each variable of the timings, measure the input powers as the value of that variable is altered (the other values are noted but left untouched). A value for that variable can be selected where minimum total input power is obtainable. Do the same for the other variables that affect the timings for that Speed Mode. Repeat as necessary.
- 4. Repeat stages 1-3 for a range of field winding supply. This will allow a plot of input powers (total and components) against field supply to be plotted and a minimised input power based upon field supply to be calculated. Calculate the field input supply voltage (or current) that minimises the total input power.
- 5. Repeat stages 3 and 4 until a minimised input power supply is obtained for that Speed Mode. Steps 3 and 4 can be carried out in the reverse order as optimisation will be guaranteed if the stages are repeated.
- 6. Repeat stages 1 to 5 for all Speed Modes. Do the same for a fixed load if necessary as the timings may be optimised for a given load rather than at no load.
- 7. Repeat stages 1 to 6 for a range of speeds.

The end result is a machine optimised for minimum input power for a given speed and load.

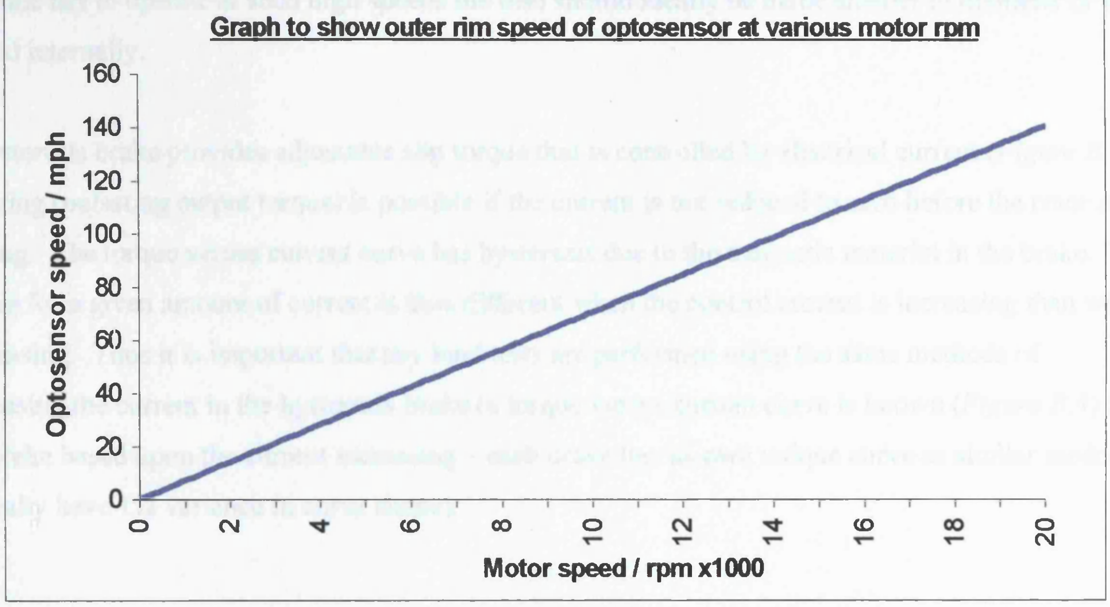


Figure B.3 :- Opto sensor peripheral speed with rpm of rotor.

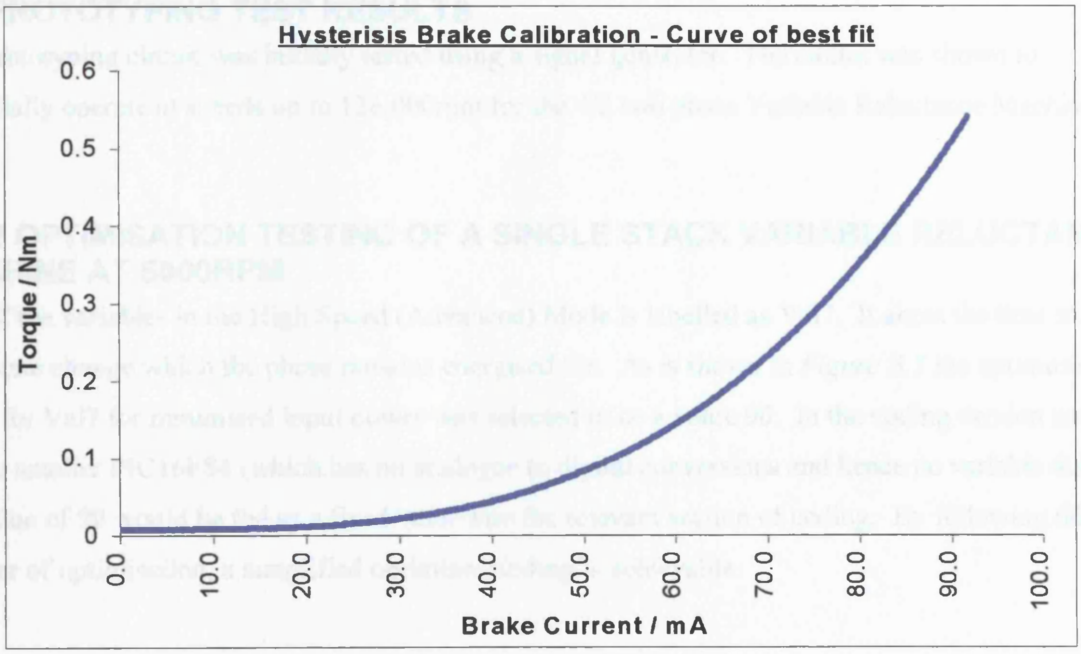


Figure B.4 :- Hysteresis brake torque versus current curve.

From these results a machine could be optimised for changing between Speed Modes (since the timing delays involved in changing between the Speed Modes does not affect the timings within each Mode).

The prototyping control circuit being is capable of running a two phase machine of this type at speeds up to 120,000rpm, based on tests using a signal generator as an optosensor signal.

The single stack variable reluctance machine utilises an external opto sensor disc for determining which phase should be energised at any one time. The disc in this machine has a 6cm diameter and the outer edge speed of the disc rotates at its edge at near 135mph at 19000rpm (*Figure B.3*). If the machine has to operate at such high speeds the disc should ideally be made smaller in diameter or be placed internally.

A hysteresis brake provides adjustable slip torque that is controlled by electrical current (*Figure B.4*). Cogging (pulsating output torque) is possible if the current is not reduced to zero before the rotor stops turning. The torque versus current curve has hysteresis due to the magnetic material in the brake. The torque for a given amount of current is thus different when the control current is increasing than when decreasing. Thus it is important that any load tests are performed using the same methods of increasing the current in the hysteresis brake (a torque versus current curve is known (*Figure B.*4) for the brake based upon the current increasing – each brake has its own unique curve as similar models typically have 1% variance in curve shape).

B.4 PROTOTYPING TEST RESULTS

The prototyping circuit was initially tested using a signal generator. The circuit was shown to potentially operate at speeds up to 126,000rpm for the 4/2 two phase Variable Reluctance Machines.

B.4.1 OPTIMISATION TESTING OF A SINGLE STACK VARIABLE RELUCTANCE MACHINE AT 5000RPM

One of the variables in the High Speed (Advanced) Mode is labelled as Val7. It alters the time from opto state change which the phase remains energised for. As is shown in *Figure B.5* the optimum value for Val7 for minimised input power was selected to be a value 90. In the coding version used for the smaller PIC16F84 (which has no analogue to digital conversions and hence no variable delays), the value of 90 would be fed as a fixed value into the relevant section of coding. By following this manner of optimisation, a simplified optimised coding is achievable.

The same was done with the other variable time delays such as Val6 which represents the delay between energising each phase in the Advanced Mode (*Figure B.6*). As shown, the choice of Val6 is less clear as the trend is uncertain but optimisation is still possible so Val6 was chosen to be 67. The numerical value was used as a multiplier for the individual time delays.

The machine can be optimised for almost any criteria, for example minimum input power at no load. The total input power at no load was thus minimised to 4.28W at 5000rpm. The waveform achieved with optimised timing control is shown in *Figure B.7*.

The main feature is that the coding is such that the numbers obtained for each channel is programmed into the smaller PIC16F84 microcontroller. The PIC16F84 microcontroller works the machine with exactly the same timing delays as determined by the prototyping circuit, but the delays are fixed values for optimised operation. The board allows optimization for any specification or requirement for example, for a given speed, torque or efficiency. Control is shown to be 'on-the-fly' as it is immediate upon the machine being optimised.

Full timing control has been demonstrated on the Single Stack Variable Reluctance Machine at speeds in High Speed Mode of up to 19020rpm.

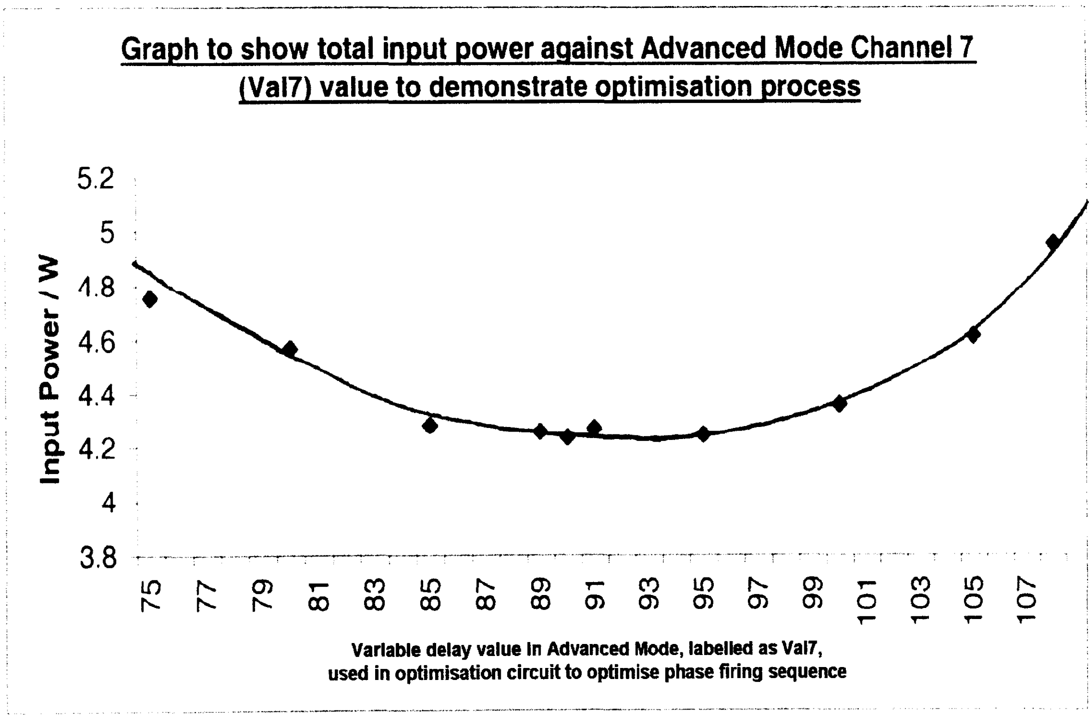


Figure B.5 :- Optimisation of input power by varying timing control.

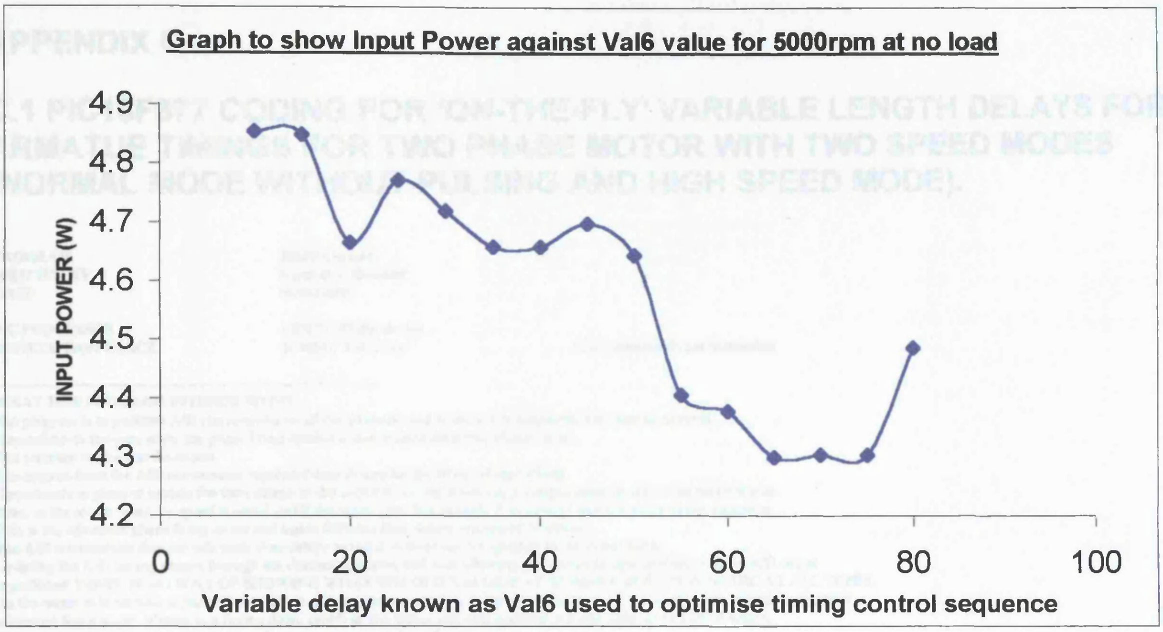


Figure B.6 :- Optimisation of input power by varying timing control.

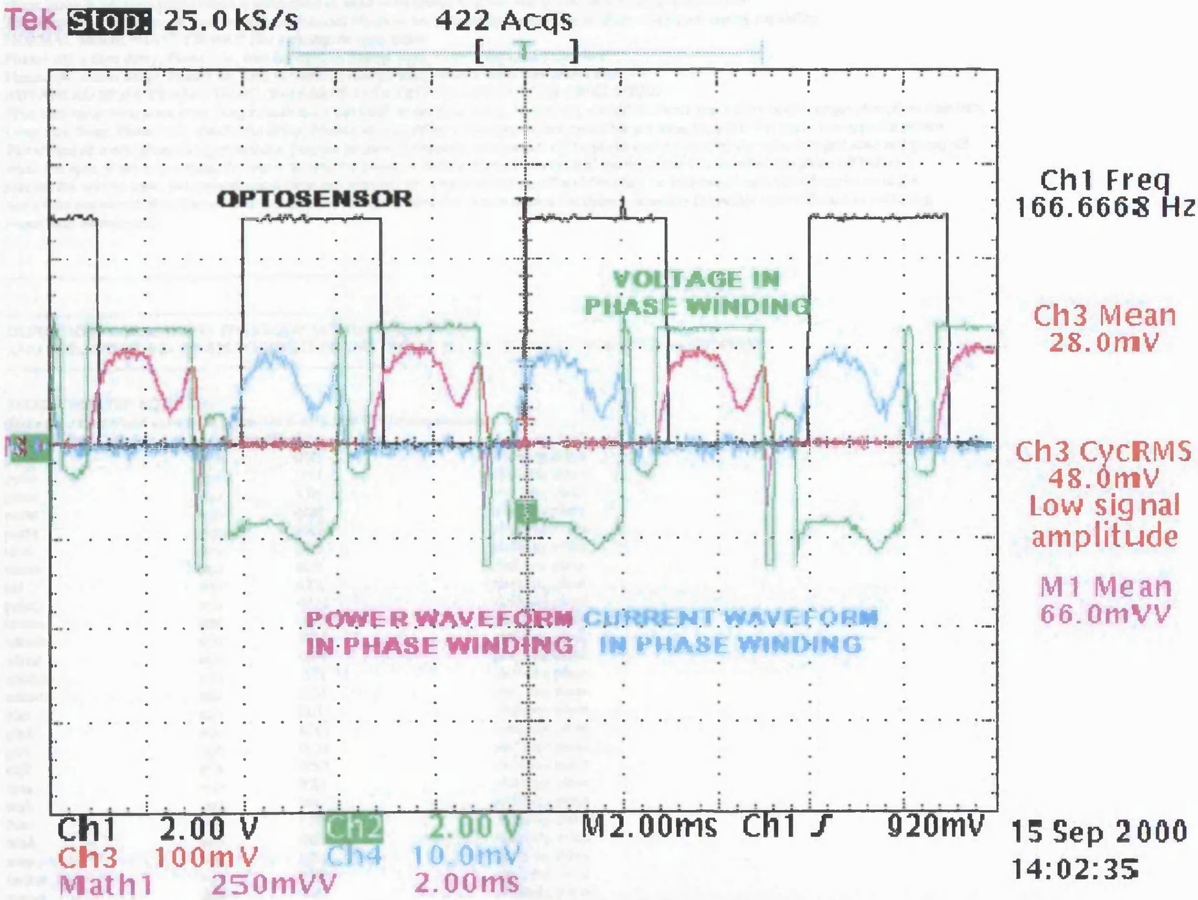


Figure B.7 :- Example of optimised phase firing sequence.

APPENDIX C

C.1 PIC16F877 CODING FOR 'ON-THE-FLY' VARIABLE LENGTH DELAYS FOR ARMATUE TIMINGS FOR TWO PHASE MOTOR WITH TWO SPEED MODES (NORMAL MODE WITHOUT PULSING AND HIGH SPEED MODE).

```
;PROGRAM                 20MHz3a.asm
;WRITTEN BY              Ewan R.T. Goodier
;DATE                    06/09/2000

;PIC PROCESSOR           16F877, 40 pin device
;INSTRUCTION CLOCK       20 MHz, T=0.05us          0.2microseconds per instruction

;-------------------------------------------------------
;WHAT THIS PROGRAM INTENDS TO DO
;The program is to perform A/D conversions on all the channels and to show the output for the selected channel.
;Depending on the opto state, the phase firing routine is determined for a two-phase motor.
;The program then starts the motor.
;The outputs from the A/D conversions represent time delays for the firing of each phase.
;Opportunity is given to update the time delays as the motor is turning allowing the opportunity to maximise performance.
;Also, as the motor turns, its speed is noted and if the motor gets fast enough, it is allowed to enter a new firing sequence.
;This is the advanced phase firing mode and needs different time delays compared to before.
;The A/D conversions also provide such time delays to suit and these can be updated as the motor turns
;Updating the A/D timing occurs through the channel selection and then allowing the chance to update if the channel A/D ouput
;is preferred.THERE IS NO WAY OF SHOWING WHAT THE OLD VALUE WAS SO MAKE SURE IT IS NOTED AT ALL TIMES.
;As the motor is in advanced phase firing, it is monitored by a crude time delay sequence to see if it is always fast enough to maintain
;advanced firing mode. If there is a timing delay problem, the motor will still continue but will light up an LED if wrong.
;If the motor is too slow then the motor will return to normal phase firing mode.
;The aim is to allow full control on an experimental basis of a two-phase switched reluctance motor where the timing of the
;normal firing of the phases is unknown, the point at which an advance phase firing may be beneficial (plus the timing of the
;phase firing in advance phase mode is unknown) all need to be found to allow maximum performance of the motor.
;The motor can be prevented from entering Advanced Mode or brought out of it manually to allow additional testing capability.
;NORMAL MODE PHASE FIRING? (for a change in opto state)
;Phase1 off, a time delay, Phase2 on, wait for opto to change state, when it does then next line
;Phase2 off, a time delay, Phase1 on, wait for opto to change state, when it does then above line
;ADVANCED MODE PHASE FIRING (REQUIRES THOUGHT!!!) - ADVANCED=HIGH SPEED
;Wait until other opto state is on, long time delay, Phase2off, small time delay, Phase1 on, should be a time gap before opto changes state, then next line
;Long time delay, Phase1 off, small time delay, Phase2 on, should be a time gap before opto changes state, then this line again but opposite phases
;The advanced mode phase firing requires a diagram to show it properly, but instead of the phase coming on after the opto changed state and going off
;when the opto enters a new state, the aim is to turn the phase on before the opto has entered the first state and switches the phase off before it
;reaches the second state, but maintaining a time gap between one phase switching off and the other on (the two phases should not be on at the
;same time nor should there be an instantaneous switching between the phases (hence the delays - it makes the motor more efficient as switching
;losses may be reduced).
;
;
;-------------------------------------------------------
;
;-------------------------------------------------------
;DEFINITIONS FOR MAKING THE PROGRAM WORK ARE BELOW
;AND THE ACTUAL PROGRAM CODING IS FOUND BELOW IT - NOTE THAT ADVANCED=HIGH SPEED
;-------------------------------------------------------
;
;FIXED REGISTER EQUATES
;these must be defined exactly as below for the PIC16F877 microprocessor
;see data sheet for what they are responsible for
porta           equu    05H     ;defining place
portb           equ     06H     ;defining place
portc           equ     07H     ;defining place
portd           equ     08H     ;defining place
porte           equ     09H     ;defining place
tmr0            equ     01H     ;defining place
status          equ     03H     ;defining place
pcl             equ     02H     ;defining place
pclath          equ     0AH     ;defining place
intcon          equ     0BH     ;defining place
adresh          equ     1EH     ;defining place
adresl          equ     9EH     ;defining place
adcon0          equ     1FH     ;defining place
adcon1          equ     9FH     ;defining place
pie1            equ     8CH     ;defining place
pie2            equ     8DH     ;defining place
pir1            equ     0CH     ;defining place
pir2            equ     0DH     ;defining place
trisa           equ     85H     ;defining place
trisb           equ     86H     ;defining place
trisc           equ     87H     ;defining place
trisd           equ     88H     ;defining place
trise           equ     89H     ;defining place
option_reg      equ     81H     ;defining place
sspcon          equ     14H     ;defining place
rcsta           equ     18H     ;defining place
;
;VARIABLE REGISTER EQUATES
;these can have different names for each but you cannot have the same name twice and the position of the equate must fall within the
;designated zones for the microcontroller in question - see the data sheets for guidance
;are used for storing 8bit long values
count           equ     20H     ;defining place - general purpose count register
del1            equ     21H     ;defining place - time value
del2            equ     22H     ;defining place - time value
adres           equ     23H     ;defining place - A/D conversion value
temp            equ     24H     ;defining place - time value
tamp            equ     25H     ;defining place - time value
del             equ     26H     ;defining place - time value
```

```
dell1       equ     28H     ;defining place - time value
dell11      equ     29H     ;defining place - time value
val0        equ     2AH     ;defining place - ch0 A/D value
val1        equ     2BH     ;defining place - ch1 A/D value
val2        equ     2CH     ;defining place - ch2 A/D value
val3        equ     2DH     ;defining place - ch3 A/D value
val4        equ     2EH     ;defining place - ch4 A/D value
val5        equ     2FH     ;defining place - ch5 A/D value
val6        equ     30H     ;defining place - ch6 A/D value
val7        equ     31H     ;defining place - ch7 A/D value
redel1      equ     32H     ;defining place - time value
redel2      equ     33H     ;defining place - time value
cheq        equ     34H     ;defining place - A/D channel selection shown
chec        equ     35H     ;defining place - A/D channel selection shown
zor         equ     36H     ;defining place - exclusive or value
val0a       equ     37H     ;defining place - ch0 A/D value
val1a       equ     38H     ;defining place - ch1 A/D value
val2a       equ     39H     ;defining place - ch2 A/D value
val3a       equ     3AH     ;defining place - ch3 A/D value
val4a       equ     3BH     ;defining place - ch4 A/D value
val5a       equ     3CH     ;defining place - ch5 A/D value
val6a       equ     3DH     ;defining place - ch6 A/D value
val7a       equ     3EH     ;defining place - ch7 A/D value
motval      equ     3FH     ;defining place - value for phase firing selection
time        equ     40H     ;defining place -  time value - advanced mode
time1       equ     41H     ;defining place - time value - advanced mode
time2       equ     42H     ;defining place - time value - advanced mode
time3       equ     43H     ;defining place - time value - advanced mode
extrachk    equ     44H     ;defining place - used for showing a timing problem
time5       equ     45H     ;defining place - time value - advanced mode          unused
d1star      equ     46H     ;defining place - time value - advanced mode
nothere     equ     47H     ;defining place - time value - advanced mode
d2star      equ     48H     ;defining place - time value - advanced mode
d2plus      equ     49H     ;defining place - time value - advanced mode
extra1      equ     4AH     ;defining place - time value - advanced mode
extra2      equ     4BH     ;defining place - time value - advanced mode
cownt       equ     4CH     ;defining place - time value
val0b       equ     4DH     ;defining place - ch0 A/D value
val1b       equ     4EH     ;defining place - ch1 A/D value
val2b       equ     4FH     ;defining place - ch2 A/D value
val3b       equ     50H     ;defining place - ch3 A/D value
val4b       equ     51H     ;defining place - ch4 A/D value
val5b       equ     52H     ;defining place - ch5 A/D value
val6b       equ     52H     ;defining place - ch6 A/D value
val7b       equ     54H     ;defining place - ch7 A/D value
heur        equ     55H     ;defining place - time value - advanced mode
heur1       equ     56H     ;defining place - time value - advanced mode
heur2       equ     57H     ;defining place - time value - advanced mode
heur3       equ     58H     ;defining place - time value - advanced mode
heur4       equ     59H     ;defining place - time value - advanced mode
;
;BIT DEFINITIONS
;registers sometimes have names to bit numbers but instead of referring to the bit number itself, they can be given names instead to
;aid in clarity of what the program is doing/affecting. Below is giving names to some of the bit numbers.
;There is no repetition as the nmemonic coding commands are such that the program automatically works out which bit to aim for and does not
;mind is the name or bit number is used
strtbut     equ     1       ;start button bit
fast        equ     5       ;direction bit
refresh     equ     5       ;refresh the system bit
phase       equ     7       ;adv/norm phase bit
opto        equ     6       ;optosensor bit
z           equ     2       ;zero bit flag
carry       equ     0       ;carry bit
toif        equ     2       ;TMR0 overflow interrupt flag bit
rbif        equ     0       ;RB<7:4> int. flag bit
rp0         equ     5       ;used in bank selector
rp1         equ     6       ;used in bank selector
peie        equ     6       ;peripheral interupt enable bit
gie         equ     7       ;global interupt enable bit
adif        equ     6       ;A/D converter interrupt flag bit
adie        equ     6       ;A/D converter interrupt enable bit
go          equ     2       ;A/D conversion status bit
f           equ     1       ;field register
w           equ     0       ;working register
;
;OUTPUTS
;Uses names to define bit numbers as explained in the section above
led3        equ     4       ;LED bit - for stopped or timing problem in advanced mode
lhslsw1     equ     0       ;Phase A bottom switch
lhshsw1     equ     1       ;Phase A top switch
rhslsw2     equ     2       ;Phase B bottom switch
rhshsw2     equ     3       ;Phase B top switch
;
;*******************************************************************
;                       CODING STARTS BELOW
;*******************************************************************

org         00h             ;tells assembler where to start loading the code 'goto  start' is the first bit of REAL program code

;*******************************************************************
;                       START SEQUENCE
;*******************************************************************

goto        start           ;goto start

init        clrf    porta       ;(from start) clear porta for defining inputs/outputs
            clrf    portb       ;clear portb for defining inputs/outputs
            clrf    portc       ;clear portc for defining inputs/outputs
            clrf    portd       ;clear portd for defining inputs/outputs
            clrf    porte       ;clear porte for defining inputs/outputs
            bcf     status,6    ;preparing to select bank 1
```

```
            bsf      status,5           ;bank1 selected
            bcf      option_reg,7       ;portb pull-ups are enabled by individual port latches
            bcf      sspcon,5           ;ra5,rc3,rc4,rc5 configured as I/O pins, serial port disabled
            bcf      rcsta,7            ;disable serial port
            bcf      trise,4            ;allows general purpose I/O mode for portd, disable parallel slave port mode portd
            bcf      intcon,7           ;disable all interupts
            clrf     pie1               ;disables peripheral
            return                      ;go back to line below (start)

init1       movlw    b'01101111'        ;(from main) set up portb        opto inputs,start input, A/D select, LED for stop or bad timing,
            movwf    trisb              ;set up portb
            movlw    B'00111111'        ;set up porta                    A/D inputs, A/D select, master clear
            movwf    trisa              ;set up porta
            movlw    B'00000000'        ;set up portc                    A/D output LED lighting
            movwf    trisc              ;set up portc
            movlw    B'00000111'        ;set up porte                    A2D inputs and setting up portd
            movwf    trise              ;set up porte
            movlw    B'00010000'        ;set up portd                    phase firing, A/D output LED lighting, direction input
            movwf    trisd              ;set up portd
            clrf     adcon1             ;all A/D inputs analogue, Vref+=supply=5V, Vref-=ground=0V, no reference voltages
            bcf      status,5           ;bank0 selected
            movlw    b'01000001'        ;Fosc/8, channel0 selected,A/Dconversion not in progress, A/D converter module operating
            movwf    adcon0             ;set it up as above
            bcf      portb,phase        ;normal mode
            return                      ;got back to line below (main)

conv        movlw    0x0f               ;(from the A/D request for the channel selected) set up time delay (15dec)
            movwf    tamp               ;put 15 into tamp
lupe        decfsz   tamp,f             ;decrement by 1, skip next line if zero
            goto     lupe               ;not zero so decrement again - delay required to allow internal capacitors to charge up to value of input voltage
            bsf      adcon0,2           ;set A/D conversion into progress
            movlw    0x3c               ;set up time delay (60dec)
            movwf    temp               ;put 60 into temp
dog         decfsz   temp,f             ;decrement by 1, skip next line if zero
            goto     dog                ;not zero so decrement again - allows time for A/D conversions to occur
conv1       btfsc    adcon0,2           ;has A/D conversion ended?
            goto     conv1              ;no - recheck until it has
            return                      ;A/D ended - value stored in adresl - got back to relevant point in coding

start       call     init               ;goto init then return here for line below

main        call     init1              ;goto init1 then return here for line below
            bcf      portb,led3         ;LED off

chzer       bcf      portd,lhslsw1      ;turn off phaseA, bottom switch                   stop motor as timing is wrong
            bcf      portd,lhshsw1      ;turn off phaseA, top switch
            bcf      portd,rhslsw2      ;turn on phaseB, bottom switch
            bcf      portd,rhshsw2      ;turn on phaseB, top switch    all phases off so set up for normal operation again - motor speed not important
            bcf      portb,led3         ;LED off
            bcf      adcon0,3           ;select channel zero for A/D conversion          (after init1 has been called)
            bcf      adcon0,4           ;select channel zero for A/D conversion
            bcf      adcon0,5           ;select channel zero for A/D conversion
            call     conv               ;goto conv for A/D conversion of channel selected then return to line below
            movf     adresl,w           ;A/D value is in adresl - move it to working register
            movwf    val0               ;move the value in the working register to val0 - A/D for that channel now in the file register for that channel
            goto     chone              ;A/D the next channel
chone       bsf      adcon0,3           ;select channel one for A/D conversion
            bcf      adcon0,4           ;select channel one for A/D conversion
            bcf      adcon0,5           ;select channel one for A/D conversion
            call     conv               ;goto conv for A/D conversion of channel selected then return to line below
            movf     adresl,w           ;A/D value is in adresl - move it to working register
            movwf    val1               ;move the value in the working register to val1 - A/D for that channel now in the file register for that channel
            goto     chtwo              ;A/D the next channel
chtwo       bcf      adcon0,3           ;select channel two for A/D conversion
            bsf      adcon0,4           ;select channel two for A/D conversion
            bcf      adcon0,5           ;select channel two for A/D conversion
            call     conv               ;goto conv for A/D conversion of channel selected then return to line below
            movf     adresl,w           ;A/D value is in adresl - move it to working register
            movwf    val2               ;move the value in the working register to val2 - A/D for that channel now in the file register for that channel
            goto     chthr              ;A/D the next channel
chthr       bsf      adcon0,3           ;select channel three for A/D conversion
            bsf      adcon0,4           ;select channel three for A/D conversion
            bcf      adcon0,5           ;select channel three for A/D conversion
            call     conv               ;goto conv for A/D conversion of channel selected then return to line below
            movf     adresl,w           ;A/D value is in adresl - move it to working register
            movwf    val3               ;move the value in the working register to val3 - A/D for that channel now in the file register for that channel
            goto     chfou              ;A/D the next channel
chfou       bcf      adcon0,3           ;select channel four for A/D conversion
            bcf      adcon0,4           ;select channel four for A/D conversion
            bsf      adcon0,5           ;select channel four for A/D conversion
            call     conv               ;goto conv for A/D conversion of channel selected then return to line below
            movf     adresl,w           ;A/D value is in adresl - move it to working register
            movwf    val4               ;move the value in the working register to val4 - A/D for that channel now in the file register for that channel
            goto     chfiv              ;A/D the next channel
chfiv       bsf      adcon0,3           ;select channel five for A/D conversion
            bcf      adcon0,4           ;select channel five for A/D conversion
            bsf      adcon0,5           ;select channel five for A/D conversion
            call     conv               ;goto conv for A/D conversion of channel selected then return to line below
            movf     adresl,w           ;A/D value is in adresl - move it to working register
            movwf    val5               ;move the value in the working register to val5 - A/D for that channel now in the file register for that channel
            goto     chsix              ;A/D the next channel
chsix       bcf      adcon0,3           ;select channel six for A/D conversion
            bsf      adcon0,4           ;select channel six for A/D conversion
            bsf      adcon0,5           ;select channel six for A/D conversion
            call     conv               ;goto conv for A/D conversion of channel selected then return to line below
            movf     adresl,w           ;A/D value is in adresl - move it to working register
            movwf    val6               ;move the value in the working register to val6 - A/D for that channel now in the file register for that channel
            goto     chsev              ;A/D the next channel
chsev       bsf      adcon0,3           ;select channel seven for A/D conversion
            bsf      adcon0,4           ;select channel seven for A/D conversion
            bsf      adcon0,5           ;select channel seven for A/D conversion
```

```
        call    conv            ;goto conv for A/D conversion of channel selected then return to line below
        movf    adresl,w        ;A/D value is in adresl - move it to working register
        movwf   val7            ;move the value in the working register to val7 - A/D for that channel now in the file register for that channel
        goto    display         ;display the A/D output for whichever channel is selected using the 3 pink wires - go to display
display btfsc   portb,3         ;3 pink wires are used to select which A/D channel is to be shown, which is detemined below
        goto    disa            ;1/3/5/7    channels applicable
        btfsc   portb,2         ;0/2/4/6    channels applicable
        goto    disb            ;2/6        channels applicable
        btfsc   porta,4         ;0/4        channels applicable
        goto    four            ;4out       channel 4 chosen
        goto    zero            ;0out       channel 0 chosen
disa    btfsc   portb,2         ;1/3/5/7    channels applicable
        goto    disc            ;3/7        channels applicable
        btfsc   porta,4         ;1/5        channels applicable
        goto    five            ;5out       channel 5 chosen
        goto    one             ;1out       channel 1 chosen
disb    btfsc   porta,4         ;2/6        channels applicable
        goto    six             ;6out       channel 6 chosen
        goto    two             ;2out       channel 2 chosen
disc    btfsc   porta,4         ;3/7        channels applicable
        goto    seven           ;7out       channel 7 chosen
        goto    three           ;3out       channel 3 chosen
zero    bcf     portd,7         ;show channel number 0 using 3 leds            channel 0 chosen
        bcf     portd,6         ;show channel number 0 using 3 leds
        bcf     portd,5         ;show channel number 0 using 3 leds
        movf    val0,w          ;move the value in val0 into the working register
        movwf   portc           ;move the contents of the working register to the LED sequence and so the A/D result for this channel is shown
        sublw   b'00000000'     ;subtract w from zero - if val? is above zero, the result will be negative
        btfsc   status,0        ;test the carry bit, if set, then val5 is zero and makes the time delay wrong, if val? is above zero, then skip
                                ; the next line and continue
        goto    chzer           ;val5 is zero so go back to the beginning and fix the value
        goto    wrongtimer      ;A/D conversions complete, channel selection output given and time delays can be made from the val? contents –
                                ;go start the motor (testdir1)
one     bsf     portd,7         ;show channel number 1 using 3 leds            channel 1 chosen
        bcf     portd,6         ;show channel number 1 using 3 leds
        bcf     portd,5         ;show channel number 1 using 3 leds
        movf    val1,w          ;move the value in val1 into the working register
        movwf   portc           ;move the contents of the working register to the LED sequence and so the A/D result for this channel is shown
        sublw   b'00000000'     ;subtract w from zero - if val? is above zero, the result will be negative
        btfsc   status,0        ;test the carry bit, if set, then val5 is zero and makes the time delay wrong, if val? is above zero, then
                                ;skip the next line and continue
        goto    chzer           ;val5 is zero so go back to the beginning and fix the value
        goto    wrongtimer      ;A/D conversions complete, channel selection output given and time delays can be made from the val? Contents
                                ; - go start the motor (testdir1)
two     bcf     portd,7         ;show channel number 2 using 3 leds            channel 2 chosen
        bsf     portd,6         ;show channel number 2 using 3 leds
        bcf     portd,5         ;show channel number 2 using 3 leds
        movf    val2,w          ;move the value in val2 into the working register
        movwf   portc           ;move the contents of the working register to the LED sequence and so the A/D result for this channel is shown
        sublw   b'00000000'     ;subtract w from zero - if val? is above zero, the result will be negative
        btfsc   status,0        ;test the carry bit, if set, then val5 is zero and makes the time delay wrong, if val? is above zero, then skip the
                                ; next line andcontinue
        goto    chzer           ;val5 is zero so go back to the beginning and fix the value
        goto    wrongtimer      ;A/D conversions complete, channel selection output given and time delays can be made from the val? Contents
                                ; - go start the motor (testdir1)
three   bsf     portd,7         ;show channel number 3 using 3 leds            channel 3 chosen
        bsf     portd,6         ;show channel number 3 using 3 leds
        bcf     portd,5         ;show channel number 3 using 3 leds
        movf    val3,w          ;move the value in val3 into the working register
        movwf   portc           ;move the contents of the working register to the LED sequence and so the A/D result for this channel is shown
        sublw   b'00000000'     ;subtract w from zero - if val? is above zero, the result will be negative
        btfsc   status,0        ;test the carry bit, if set, then val5 is zero and makes the time delay wrong, if val? is above zero, then skip the
                                ; next line and continue
        goto    chzer           ;val5 is zero so go back to the beginning and fix the value
        goto    wrongtimer      ;A/D conversions complete, channel selection output given and time delays can be made from the val? Contents
                                ; - go start the motor (testdir1)
four    bcf     portd,7         ;show channel number 4 using 3 leds            channel 4 chosen
        bcf     portd,6         ;show channel number 4 using 3 leds
        bsf     portd,5         ;show channel number 4 using 3 leds
        movf    val4,w          ;move the value in val4 into the working register
        movwf   portc           ;move the contents of the working register to the LED sequence and so the A/D result for this channel is shown
        sublw   b'00000000'     ;subtract w from zero - if val? is above zero, the result will be negative
        btfsc   status,0        ;test the carry bit, if set, then val5 is zero and makes the time delay wrong, if val? is above zero, then skip
                                ; the next line and continue
        goto    chzer           ;val5 is zero so go back to the beginning and fix the value
        goto    wrongtimer      ;A/D conversions complete, channel selection output given and time delays can be made from the val? Contents
                                ; - go start the motor (testdir1)
five    bsf     portd,7         ;show channel number 5 using 3 leds            channel 5 chosen
        bcf     portd,6         ;show channel number 5 using 3 leds
        bsf     portd,5         ;show channel number 5 using 3 leds
        movf    val5,w          ;move the value in val5 into the working register
        movwf   portc           ;move the contents of the working register to the LED sequence and so the A/D result for this channel is shown
        sublw   b'00000000'     ;subtract w from zero - if val? is above zero, the result will be negative
        btfsc   status,0        ;test the carry bit, if set, then val5 is zero and makes the time delay wrong, if val? is above zero, then skip
                                ; the next line and continue
        goto    chzer           ;val5 is zero so go back to the beginning and fix the value
        goto    wrongtimer      ;A/D conversions complete, channel selection output given and time delays can be made from the val? Contents
                                ; - go start the motor (testdir1)
six     bcf     portd,7         ;show channel number 6 using 3 leds            channel 6 chosen
        bsf     portd,6         ;show channel number 6 using 3 leds
        bsf     portd,5         ;show channel number 6 using 3 leds
        movf    val6,w          ;move the value in val6 into the working register
        movwf   portc           ;move the contents of the working register to the LED sequence and so the A/D result for this channel is shown
        sublw   b'00000000'     ;subtract w from zero - if val? is above zero, the result will be negative
        btfsc   status,0        ;test the carry bit, if set, then val5 is zero and makes the time delay wrong, if val? is above zero, then skip the
                                ;next line and continue
        goto    chzer           ;val5 is zero so go back to the beginning and fix the value
        goto    wrongtimer      ;A/D conversions complete, channel selection output given and time delays can be made from the val? Contents
                                ; - go start the motor (testdir1)
seven   bsf     portd,7         ;show channel number 7 using 3 leds            channel 7 chosen
        bsf     portd,6         ;show channel number 7 using 3 leds
```

```
                bsf         portd,5         ;show channel number 7 using 3 leds
                movf        val7,w          ;move the value in val7 into the working register
                movwf       portc           ;move the contents of the working register to the LED sequence and so the A/D result for this channel is shown
                sublw       b'00000000'     ;subtract w from zero - if val? is above zero, the result will be negative
                btfsc       status,0        ;test the carry bit, if set, then val5 is zero and makes the time delay wrong, if val? is above zero, then skip
                                            ; the next line and continue
                goto        chzer           ;val5 is zero so go back to the beginning and fix the value
                goto        wrongtimer      ;A/D conversions complete, channel selection output given and time delays can be made from the val? Contents
                                            ; - go start the motor (testdir1)

wrongtimer      btfss       portb,0         ;master timing control on?
                goto        mastflash       ;yes
                goto        nstart          ;no, so see if can start motor
nstart          btfsc       portb,strtbut   ;test for start button being on
                goto        strtflash       ;no, so keep on A/D converting
                goto        ninit           ;yes, so start initialisation for normal mode

strtflash       goto        strtfla         ;continue A/D conversions whilst the start button detection is being carried out
strtfla         bsf         portb,led3      ;led flashes quickly - prompt for start switch to be put on - LED on
                call        strtdelay       ;delay timing sequence to allow visible flashing sequence - return to line below after strtdelay coding has finished
                bcf         portb,led3      ;LED off - flash occurring
                call        strtdelay       ;delay timing sequence to allow visible flashing sequence - return to line below after strtdelay coding has finished
                goto        chzer           ;go to chzer to allow retest of start button being pressed and A/D value alteration
strtdelay       movlw       0x0a            ;load 2dec into working register - this coding allows a time delay so flashing of LED is visible
                movwf       del111          ;move 2dec from working register into del111 file register
strdelay        movlw       0x58            ;load 88dec into working register
                movwf       del11           ;move 88dec from working register into del11 file register
stdelay         movlw       0xee            ;load 238dec into working register
                movwf       del1            ;move 238dec from working register into del1 file register
sdlay           decfsz      del1,1          ;decrement del1 and keep new value, skip next line if zero
                goto        sdlay           ;del1 not zero so decrement again
                decfsz      del11,1         ;del1 is zero, decrement del11 and keep new value, skip next line if zero
                goto        stdelay         ;del11 not zero, reload del1 again so can decrement again and maintain delay timing properly
                decfsz      del111,1        ;del11 is zero, decrement del111 and keep new value, skip next line if zero
                goto        strdelay        ;del111 not zero, reload del1 and del11 so can both decrement again and maintain delay timing properly
                return                      ;time delay complete - go back to line below the call strtdelay - total
                                            ;delay=1micro*2*88*238*6executions=0.251sec

mastflash       goto        mastfla         ;continue A/D conversions whilst the start button detection is being carried out
mastfla         bsf         portb,led3      ;led flashes quickly - prompt for start switch to be put on - LED on
                call        mastdelay       ;delay timing sequence to allow visible flashing sequence - return to line below after strtdelay coding has finished
                bcf         portb,led3      ;LED off - flash occurring
                call        mastdelay       ;delay timing sequence to allow visible flashing sequence - return to line below after strtdelay coding has finished
                goto        chzer           ;go to chzer to allow retest of start button being pressed and A/D value alteration
mastdelay       movlw       0x0a            ;load 2dec into working register - this coding allows a time delay so flashing of LED is visible
                movwf       del111          ;move 2dec from working register into del111 file register
masdelay        movlw       0x2c            ;load 44dec into working register
                movwf       del11           ;move 44dec from working register into del11 file register
madelay         movlw       0xee            ;load 238dec into working register
                movwf       del1            ;move 238dec from working register into del1 file register
mdlay           decfsz      del1,1          ;decrement del1 and keep new value, skip next line if zero
                goto        mdlay           ;del1 not zero so decrement again
                decfsz      del11,1         ;del1 is zero, decrement del11 and keep new value, skip next line if zero
                goto        madelay         ;del11 not zero, reload del1 again so can decrement again and maintain delay timing properly
                decfsz      del111,1        ;del11 is zero, decrement del111 and keep new value, skip next line if zero
                goto        masdelay        ;del111 not zero, reload del1 and del11 so can both decrement again and maintain delay timing properly
                return                      ;time delay complete - go back to line below the call strtdelay - total
                                            ;delay=1micro*2*44*238*6executions=0.1257sec

ninit           btfsc       portb,6         ;test for opto state
                goto        ngo1            ;opto clear ie opto=0
                goto        ngo2            ;opto set ie opto=1
ngo1            movlw       0x5c            ;opto=0  put 5c(hex) into working register
                movwf       motval          ;store 5c in motval file register - motval used to detect how phase firing and opto position are linked
                goto        ntests          ;initialisation complete so motor works properly - go to testopto section
ngo2            movlw       0x5a            ;opto=1  put 5a(hex) into working register
                movwf       motval          ;store 5a in motval file register - motval used to detect how phase firing and opto position are linked
                goto        ntests          ;initialisation complete so motor works properly - go to testoptos section
ntests          btfss       portb,6
                goto        nsgoons1
                goto        nsgoons2
nsgoons1        bsf         portd,lhslsw1
                bsf         portd,lhshsw1
                goto        ntestopto
nsgoons2        bsf         portd,rhslsw2
                bsf         portd,rhshsw2
                goto        ntestopto
ntestopto       btfsc       portb,strtbut   ;test for start button still being on
                goto        chzer           ;button off so stop motor
                goto        ntestopt        ;button still on
ntestopt        bcf         portb,phase
                btfsc       portb,0         ;if the user wants to set more than one A/D time value before updating, then use this
                goto        ntestopt11      ;not required, so user happy with timing at the moment
                goto        chzer           ;timing can be altered for all channels as long as the master switch is on - motor stops and is essentially reset
ntestopt11      btfss       portb,6         ;test the state of the opto to decide which phase is to be fired first in normal timing
                goto        ngoon1          ;opto=0         so phaseA on, phaseB off      goto goon1
                goto        ngoon2          ;opto=1         so phaseA off, phaseB on      goto goon2

ngoon1          movf        motval,w        ;opto=0         bring back value of motval to working rgister,w to determine how firing is to occur
                sublw       0x5b            ;0x5b-w                    just a calculation to determine if can continue or if the motor has stopped

                btfsc       status,0        ;check value of carry      the carry bit shows the result to be +'ve or -'ve
                goto        nstill1         ;+'ve(carry bit set)                       0x5b-0x5a    same state    goto still1 coding
                goto        nfire1          ;-'ve(carry bit clear)         0x5b-0x5c   new state     goto fire1 coding
nfire1          bcf         portd,rhslsw2   ;turn off phaseB, bottom switch                           OPTO=0
                bcf         portd,rhshsw2   ;turn off phaseB, top switch
                call        nxdelay1        ;time delay (using A/D conversion to give variable delay)
                bsf         portd,lhslsw1   ;turn on phaseA, bottom switch
                bsf         portd,lhshsw1   ;turn on phaseA, top switch
                movlw       0x5a            ;5a given to motval
                movwf       motval          ;see above
```

```
            btfsc   portb,strtbut
            goto    chzer
ndisplay1n  btfsc   portb,3        ;3 pink wires are used to select which A/D channel is to be shown - go to n1del1 is wrong channel
            goto    ndisaa1n       ;1/3/5/7    channels applicable
            btfsc   portb,2        ;0/2/4/6    channels applicable
            goto    ndisbb1n       ;2/6        channels applicable
            btfsc   porta,4        ;0/4        channels applicable
            goto    n1del1         ;4out       channel 4 chosen
            goto    nzer1          ;0out       channel 0 chosen
ndisaa1n    btfsc   portb,2        ;1/3/5/7    channels applicable
            goto    ndiscc1n       ;3/7        channels applicable
            btfsc   porta,4        ;1/5        channels applicable
            goto    n1del1         ;5out       channel 5 chosen
            goto    none1          ;1out       channel 1 chosen
ndisbb1n    btfsc   porta,4        ;2/6        channels applicable
            goto    n1del1         ;6out       channel 6 chosen
            goto    ntwo1          ;2out       channel 2 chosen
ndiscc1n    btfsc   porta,4        ;3/7        channels applicable
            goto    n1del1         ;7out       channel 7 chosen
            goto    nthree1        ;3out       channel 3 chosen

nzer1       bcf     portd,7        ;show channel number 2 using 3 leds          channel 0 chosen
            bcf     portd,6        ;show channel number 2 using 3 leds
            bcf     portd,5        ;show channel number 2 using 3 leds
            goto    nchzer1        ;A/D conversion for that channel and store the value in different file register to normal (val2a instead of val2)
nzer11      btfsc   portb,refresh  ;test the bit for the requirement  to replace the contents of val2 with val2a - allows alterartion of timing while
                                   ; program is running
            goto    n1del1         ;no requirement to refresh so ignore value of val1a and keep val2 as it is - go to testopt
            movwf   val0           ;requirement to refresh, so val2 is given the value of val2a which is shown on the LED sequence
            goto    n1del1         ;timing has been carried out and so continue with which motor phase firing sequence for normal operation
                                   ;- go to testopt
none1       bsf     portd,7        ;show channel number 3 using 3 leds          channel 1 chosen
            bcf     portd,6        ;show channel number 3 using 3 leds
            bcf     portd,5        ;show channel number 3 using 3 leds
            goto    nchone1        ;A/D conversion for that channel and store the value in different file register to normal (val3a instead of val3)
none11      btfsc   portb,refresh  ;test the bit for the requirement  to replace the contents of val3 with val3a - allows alterartion of timing while
                                   ; program is running
            goto    n1del1         ;no requirement to refresh so ignore value of val3a and keep val3 as it is -
            movwf   val1           ;requirement to refresh, so val3 is given the value of val3a which is shown on the LED sequence
            goto    n1del1         ;timing has been carried out and so continue with which motor phase firing sequence for normal operation
                                   ; - go to testopt
nchzer1     bcf     adcon0,3       ;select channel two for A/D conversion
            bcf     adcon0,4       ;select channel two for A/D conversion
            bcf     adcon0,5       ;select channel two for A/D conversion
            call    conv           ;goto conv forA/D conversion oft channel selected then return to line below
            movf    adresl,w       ;A/D value is in adresl - move it to working register
            movwf   val0a          ;move the value in the working register to val2a - A/D for that channel now in the different file register for that
                                   ;channel
            movwf   portc          ;show the output from this A/D conversion in the LED sequence
            sublw   b'00000000'    ;subtract w from zero - if val? is above zero, the result will be negative
            btfsc   status,0       ;test the carry bit, if set, then val5 is zero and makes the time delay wrong, if val? is above zero, then skip the
                                   ;next line and continue
            goto    chzer          ;val5 is zero so go back to the beginning and fix the value
            movf    adresl,w       ;A/D value is in adresl - move it to working register
            goto    nzer11         ;go back to the A/D channel selection and see whether this value is to replace the original value
nchone1     bsf     adcon0,3       ;select channel three for A/D conversion
            bcf     adcon0,4       ;select channel three for A/D conversion
            bcf     adcon0,5       ;select channel three for A/D conversion
            call    conv           ;goto conv forA/D conversion oft channel selected then return to line below
            movf    adresl,w       ;A/D value is in adresl - move it to working register
            movwf   val1a          ;move the value in the working register to val3a - A/D for that channel now in the different file register
                                   ; for that channel
            movwf   portc          ;show the output from this A/D conversion in the LED sequence
            sublw   b'00000000'    ;subtract w from zero - if val? is above zero, the result will be negative
            btfsc   status,0       ;test the carry bit, if set, then val5 is zero and makes the time delay wrong, if val? is above zero, then
                                   ; skip the next line and continue
            goto    chzer          ;val5 is zero so go back to the beginning and fix the value
            movf    adresl,w       ;A/D value is in adresl - move it to working register
            goto    none11         ;go back to the A/D channel selection and see whether this value is to replace the original value


ntwo1       bcf     portd,7        ;show channel number 2 using 3 leds          channel 2 chosen
            bsf     portd,6        ;show channel number 2 using 3 leds
            bcf     portd,5        ;show channel number 2 using 3 leds
            goto    nchtwo1        ;A/D conversion for that channel and store the value in different file register to normal (val2a instead of val2)
ntwo11      btfsc   portb,refresh  ;test the bit for the requirement  to replace the contents of val2 with val2a - allows alterartion of
                                   ; timing while program is running
            goto    n1del1         ;no requirement to refresh so ignore value of val1a and keep val2 as it is - go to testopt
            movwf   val2           ;requirement to refresh, so val2 is given the value of val2a which is shown on the LED sequence
            goto    n1del1         ;timing has been carried out and so continue with which motor phase firing sequence for normal operation
                                   ;- go to testopt
nthree1     bsf     portd,7        ;show channel number 3 using 3 leds          channel 3 chosen
            bsf     portd,6        ;show channel number 3 using 3 leds
            bcf     portd,5        ;show channel number 3 using 3 leds
            goto    nchthr1        ;A/D conversion for that channel and store the value in different file register to normal (val3a instead of val3)
nthree11    btfsc   portb,refresh  ;test the bit for the requirement  to replace the contents of val3 with val3a - allows alterartion of timing
                                   ; while program is running
            goto    n1del1         ;no requirement to refresh so ignore value of val3a and keep val3 as it is -
            movwf   val3           ;requirement to refresh, so val3 is given the value of val3a which is shown on the LED sequence
            goto    n1del1         ;timing has been carried out and so continue with which motor phase firing sequence for normal operation
                                   ; - go to testopt
nchtwo1     bcf     adcon0,3       ;select channel two for A/D conversion
            bsf     adcon0,4       ;select channel two for A/D conversion
            bcf     adcon0,5       ;select channel two for A/D conversion
            call    conv           ;goto conv forA/D conversion oft channel selected then return to line below
            movf    adresl,w       ;A/D value is in adresl - move it to working register
            movwf   val2a          ;move the value in the working register to val2a - A/D for that channel now in the different file register
                                   ; for that channel
            movwf   portc          ;show the output from this A/D conversion in the LED sequence
            sublw   b'00000000'    ;subtract w from zero - if val? is above zero, the result will be negative
            btfsc   status,0       ;test the carry bit, if set, then val5 is zero and makes the time delay wrong, if val? is above zero, then
```

```
                                        ; skip the next line and continue
            goto        chzer           ;val5 is zero so go back to the beginning and fix the value
            movf        adresl,w        ;A/D value is in adresl - move it to working register
            goto        ntwo11          ;go back to the A/D channel selection and see whether this value is to replace the original value
nchthr1     bsf         adcon0,3        ;select channel three for A/D conversion
            bsf         adcon0,4        ;select channel three for A/D conversion
            bcf         adcon0,5        ;select channel three for A/D conversion
            call        conv            ;goto conv forA/D conversion oft channel selected then return to line below
            movf        adresl,w        ;A/D value is in adresl - move it to working register
            movwf       val3a           ;move the value in the working register to val3a - A/D for that channel now in the different file register
                                        ;for that channel
            movwf       portc           ;show the output from this A/D conversion in the LED sequence
            sublw       b'00000000'     ;subtract w from zero - if val? is above zero, the result will be negative
            btfsc       status,0        ;test the carry bit, if set, then val5 is zero and makes the time delay wrong, if val? is above zero, then skip
                                        ; the next line and continue
            goto        chzer           ;val5 is zero so go back to the beginning and fix the value
            movf        adresl,w        ;A/D value is in adresl - move it to working register
            goto        nthree11        ;go back to the A/D channel selection and see whether this value is to replace the original value
n1del1      btfss       portb,0         ;master button for more?
            goto        chzer           ;yes, so do more
            movf        val2,w          ;time to see if fast enough for advanced mode
            movwf       heur3           ;val2 to heur3
ns11a       movf        val3,w          ;
            movwf       heur2           ;val3 to heur2
ns11b       movlw       0x0a            ;
            movwf       heur1           ;10 to heur1
n11         decfsz      heur1,f         ;start time delay
            goto        n11a            ;not zero
            goto        n11b            ;zero
n11a        btfss       portb,6         ;opto is 0, has it gone to 1 too early - ie is timing wrong
            goto        n11             ;opto=0 - still too slow
            goto        nadv1           ;opto=1 - too fast so advanced mode
n11b        decfsz      heur2,f         ;continue time delay
            goto        ns11b           ;not zero
            goto        n11c            ;zero
n11c        decfsz      heur3,f         ;continue time delay
            goto        ns11a           ;not zero
            goto        nfin11          ;zero
nadv1       bsf         portb,led3      ;LED on                              ADVANCED MODE WANTED
            goto        advanced        ;go to next firing change      as now opto=1
nfin11      bcf         portb,led3      ;LED off                             NORMAL MODE RETAINED
            goto        ntestopto       ;want to get opto=1 - stay in normal mode - go to ntestopto to wait for state change
nstill1     bcf         portb,led3      ;stopped or very slow turning - opto=0
            movlw       0x82            ;load working register with value 26dec        state still opto=0 - total
                                        ;delay=1micro*26*255*75*6executions=2.9835sec
            movwf       del111          ;load this value into del111
n1xdeelaya  movlw       0xff            ;load working register with value 255dec
            movwf       del11           ;load this value into del11
n1xdeeelaya movlw       0x4b            ;load working register with value75dec
            movwf       del1            ;load this value into del1
n1xdlaya    decfsz      del1,1          ;decrement del1 and keep new value, skip next line if zero
            goto        n1xdlaya1       ;go and test opto for a change in state
            goto        n1xdlayaa       ;continue countdown
n1xdlayaa   decfsz      del11,1         ;del1 zero - decrement del11 and keep new value, skip next line if zero
            goto        n1xdeeelaya     ;del11 not zero, reload and decrement del1 for correct time delay sequence
            goto        n1doesitsta     ;continue countdown
n1doesitsta decfsz      del111,1        ;del11 zero -  decrement del111 and keep new value, skip next line if zero
            goto        n1xdeelaya      ;del111 not zero - reload and decrement del1 and del11 as before for correct time delay sequence
            goto        n1refire1a      ;opto state not changed within time limit so motor has stopped - go to refire1a coding
n1xdlaya1   btfss       portb,6         ;test the opto for a change in state
            goto        n1xdlaya        ;opto=0 still thus continue countdown  - go to xdlaya coding
            goto        n1goon3         ;opto=1 thus opto state still changing and hence motor still turning - go to goon3 coding to allow motor
                                        ; to turn still
n1goon3     movlw       0x5a            ;give w new value of 5a(hex)so can change to next firing routine with no problems - motor is still turning
            movwf       motval          ;put 5a(hex) into motval so circuit can switch phases on and off correctly for each opto state
n1goon3a    btfsc       portb,strtbut   ;test for start button still being on
            goto        chzer           ;button off so stop motor
            btfsc       portb,0         ;if the user wants to set more than one A/D time value before updating, then use this
            goto        nfire2          ;go to coding labelled goon2
            goto        chzer           ;timing can be altered for all channels as long as the master switch is on
n1refire1a  bsf         portb,led3      ;motor has stopped turning - opto=0 - turn on LED as a warning light - now switch off all phases to avoid
                                        ; burnout of windings
            bcf         portd,lhslsw1   ;turn off phaseA, bottom switch
            bcf         portd,lhshsw1   ;turn off phaseA, top switch
            bcf         portd,rhslsw2   ;turn off phaseB, bottom switch
            bcf         portd,rhshsw2   ;turn off phaseB, top switch
n1refire1   btfss       portb,6         ;test opto for being set i.e. output logic 1 - motor must turn to do this
            goto        n1refire11      ;opto=0 - no state change - check again - but A/D conversion check first
            goto        n1fire2more     ;opto=1 - state has changed - so try to turn motor again - go to line below
n1fire2more bcf         portb,led3      ;switch LED off
            movlw       0x5a            ;
            movwf       motval          ;5a into motval
            goto        nfire2          ;go to coding labelled fire2 to start motor again
n1refire11  goto        refdisplay      ;although motor has stopped, the A/D timing alterations can still continue
n1refir11   btfsc       portb,strtbut   ;start button on?
            goto        chzer           ;no so stop trying to turn motor
            goto        n1refire1a      ;go back to checking for the change of opto state - go to coding labelled n1refire1a


ngoon2      movf        motval,w        ;opto=1         bring back value of motval to working rgister,w to determine how firing is to occur
            sublw       0x5b            ;0x5b-w                        just a calculation to determine if can continue or if the motor has stopped
            btfsc       status,0        ;check value of carry          the carry bit shows the result to be +'ve or -'ve
            goto        nfire2          ;+'ve(carry bit set)                   0x5b-0x5a     new state      goto fire2 coding
            goto        nstill2         ;-'ve(carry bit clear)         0x5b-0x5c     same state     goto still2 coding
nfire2      bcf         portd,lhslsw1   ;turn off phaseA, bottom switch                             OPTO=1
            bcf         portd,lhshsw1   ;turn off phaseA, top switch
            call        nxdelay1        ;time delay (using A/D conversion to give variable delay)
            bsf         portd,rhslsw2   ;turn on phaseB, bottom switch
            bsf         portd,rhshsw2   ;turn on phaseB, top switch
            movlw       0x5c            ;
            movwf       motval          ;5c into motval
```

```
		btfsc	portb,strtbut
		goto	chzer
ndisplay2n	btfsc	portb,3		;3 pink wires are used to select which A/D channel is to be shown, which is detemined below
		goto	ndisaa2n	;1/3/5/7	channels applicable
		btfsc	portb,2		;0/2/4/6	channels applicable
		goto	ndisbb2n	;2/6		channels applicable
		btfsc	porta,4		;0/4		channels applicable
		goto	n2del2		;4out		channel 4 chosen
		goto	nzer2		;0out		channel 0 chosen
ndisaa2n	btfsc	portb,2		;1/3/5/7	channels applicable
		goto	ndiscc2n	;3/7		channels applicable
		btfsc	porta,4		;1/5		channels applicable
		goto	n2del2		;5out		channel 5 chosen
		goto	none2		;1out		channel 1 chosen
ndisbb2n	btfsc	porta,4		;2/6		channels applicable
		goto	n2del2		;6out		channel 6 chosen
		goto	ntwo2		;2out		channel 2 chosen
ndiscc2n	btfsc	porta,4		;3/7		channels applicable
		goto	n2del2		;7out		channel 7 chosen
		goto	nthree2		;3out		channel 3 chosen


nzer2		bcf	portd,7		;show channel number 2 using 3 leds			channel 0 chosen
		bcf	portd,6		;show channel number 2 using 3 leds
		bcf	portd,5		;show channel number 2 using 3 leds
		goto	nchzer2		;A/D conversion for that channel and store the value in different file register to normal (val2a instead of val2)
nzer22		btfsc	portb,refresh	;test the bit for the requirement  to replace the contents of val2 with val2a - allows alterartion of timing
					; while program is running
		goto	n2del2		;no requirement to refresh so ignore value of val1a and keep val2 as it is - go to testopt
		movwf	val0		;requirement to refresh, so val2 is given the value of val2a which is shown on the LED sequence
		goto	n2del2		;timing has been carried out and so continue with which motor phase firing sequence for normal operation
					;- go to testopt
none2		bsf	portd,7		;show channel number 3 using 3 leds			channel 1 chosen
		bcf	portd,6		;show channel number 3 using 3 leds
		bcf	portd,5		;show channel number 3 using 3 leds
		goto	nchone2		;A/D conversion for that channel and store the value in different file register to normal (val3a instead of val3)
none22		btfsc	portb,refresh	;test the bit for the requirement  to replace the contents of val3 with val3a - allows alterartion of timing
					;while program is running
		goto	n2del2		;no requirement to refresh so ignore value of val3a and keep val3 as it is -
		movwf	val1		;requirement to refresh, so val3 is given the value of val3a which is shown on the LED sequence
		goto	n2del2		;timing has been carried out and so continue with which motor phase firing sequence for normal operation
					; - go to testopt
nchzer2		bcf	adcon0,3	;select channel two for A/D conversion
		bcf	adcon0,4	;select channel two for A/D conversion
		bcf	adcon0,5	;select channel two for A/D conversion
		call	conv		;goto conv forA/D conversion oft channel selected then return to line below
		movf	adresl,w	;A/D value is in adresl - move it to working register
		movwf	val0a		;move the value in the working register to val0a - A/D for that channel now in the different file register for
					; that channel
		movwf	portc		;show the output from this A/D conversion in the LED sequence
		sublw	b'00000000'	;subtract w from zero - if val? is above zero, the result will be negative
		btfsc	status,0	;test the carry bit, if set, then val5 is zero and makes the time delay wrong, if val? is above zero, then skip the
					;next line and continue
		goto	chzer		;val5 is zero so go back to the beginning and fix the value
		movf	adresl,w	;A/D value is in adresl - move it to working register
		goto	nzer22		;go back to the A/D channel selection and see whether this value is to replace the original value
nchone2		bsf	adcon0,3	;select channel three for A/D conversion
		bcf	adcon0,4	;select channel three for A/D conversion
		bcf	adcon0,5	;select channel three for A/D conversion
		call	conv		;goto conv forA/D conversion oft channel selected then return to line below
		movf	adresl,w	;A/D value is in adresl - move it to working register
		movwf	val1a		;move the value in the working register to val1a - A/D for that channel now in the different file register for
					; that channel
		movwf	portc		;show the output from this A/D conversion in the LED sequence
		sublw	b'00000000'	;subtract w from zero - if val? is above zero, the result will be negative
		btfsc	status,0	;test the carry bit, if set, then val5 is zero and makes the time delay wrong, if val? is above zero, then
					;skip the next line and continue
		goto	chzer		;val5 is zero so go back to the beginning and fix the value
		movf	adresl,w	;A/D value is in adresl - move it to working register
		goto	none22		;go back to the A/D channel selection and see whether this value is to replace the original value


ntwo2		bcf	portd,7		;show channel number 2 using 3 leds			channel 2 chosen
		bsf	portd,6		;show channel number 2 using 3 leds
		bcf	portd,5		;show channel number 2 using 3 leds
		goto	nchtwo2		;A/D conversion for that channel and store the value in different file register to normal (val2a instead of val2)
ntwo22		btfsc	portb,refresh	;test the bit for the requirement  to replace the contents of val2 with val2a - allows alterartion of timing
					; while program is running
		goto	n2del2		;no requirement to refresh so ignore value of val1a and keep val2 as it is - go to testopt
		movwf	val2		;requirement to refresh, so val2 is given the value of val2a which is shown on the LED sequence
		goto	n2del2		;timing has been carried out and so continue with which motor phase firing sequence for normal operation
					; - go to testopt
nthree2		bsf	portd,7		;show channel number 3 using 3 leds			channel 3 chosen
		bsf	portd,6		;show channel number 3 using 3 leds
		bcf	portd,5		;show channel number 3 using 3 leds
		goto	nchthr2		;A/D conversion for that channel and store the value in different file register to normal (val3a instead of val3)
nthree22	btfsc	portb,refresh	;test the bit for the requirement  to replace the contents of val3 with val3a - allows alterartion of timing
					; while program is running
		goto	n2del2		;no requirement to refresh so ignore value of val3a and keep val3 as it is -
		movwf	val3		;requirement to refresh, so val3 is given the value of val3a which is shown on the LED sequence
		goto	n2del2		;timing has been carried out and so continue with which motor phase firing sequence for normal operation
					; - go to testopt
nchtwo2		bcf	adcon0,3	;select channel two for A/D conversion
		bsf	adcon0,4	;select channel two for A/D conversion
		bcf	adcon0,5	;select channel two for A/D conversion
		call	conv		;goto conv forA/D conversion oft channel selected then return to line below
		movf	adresl,w	;A/D value is in adresl - move it to working register
		movwf	val2a		;move the value in the working register to val2a - A/D for that channel now in the different file register for
					; that channel
		movwf	portc		;show the output from this A/D conversion in the LED sequence
		sublw	b'00000000'	;subtract w from zero - if val? is above zero, the result will be negative
```

```
            btfsc   status,0        ;test the carry bit, if set, then val5 is zero and makes the time delay wrong, if val? is above zero, then skip
                                    ; the next line and continue
            goto    chzer           ;val5 is zero so go back to the beginning and fix the value
            movf    adresl,w        ;A/D value is in adresl - move it to working register
            goto    ntwo22          ;go back to the A/D channel selection and see whether this value is to replace the original value
nchthr2     bsf     adcon0,3        ;select channel three for A/D conversion
            bsf     adcon0,4        ;select channel three for A/D conversion
            bcf     adcon0,5        ;select channel three for A/D conversion
            call    conv            ;goto conv forA/D conversion oft channel selected then return to line below
            movf    adresl,w        ;A/D value is in adresl - move it to working register
            movwf   val3a           ;move the value in the working register to val3a - A/D for that channel now in the different file register
                                    ;for that channel
            movwf   portc           ;show the output from this A/D conversion in the LED sequence
            sublw   b'00000000'     ;subtract w from zero - if val? is above zero, the result will be negative
            btfsc   status,0        ;test the carry bit, if set, then val5 is zero and makes the time delay wrong, if val? is above zero, then skip the
                                    ;next line and continue
            goto    chzer           ;val5 is zero so go back to the beginning and fix the value
            movf    adresl,w        ;A/D value is in adresl - move it to working register
            goto    nthree22        ;go back to the A/D channel selection and see whether this value is to replace the original value
n2del2      btfss   portb,0         ;master button for more?
            goto    chzer           ;yes, so do more
            movf    val2,w          ;time to see if fast enough for advanced mode
            movwf   heur3           ;val2 to heur3
ns22a       movf    val3,w          ;
            movwf   heur2           ;val3 to heur2
ns22b       movlw   0x0a            ;
            movwf   heur1           ;10 to heur1
n22         decfsz  heur1,f         ;start countdown
            goto    n22a            ;not zero
            goto    n22b            ;zero
n22a        btfsc   portb,6         ;opto is 1, has it gone to 0 too early - ie is timing wrong
            goto    n22             ;opto=1 - still too slow
            goto    nadv2           ;opto=0 - too fast so advanced mode
n22b        decfsz  heur2,f         ;continue countdown
            goto    ns22b           ;not zero
            goto    n22c            ;zero
n22c        decfsz  heur3,f         ;continue countdown
            goto    ns22a           ;not zero
            goto    nfin22          ;zero
nadv2       bsf     portb,led3      ;LED on                                    ADVANCED MODE WANTED
            goto    advanced        ;go to next firing change       as now opto=0
nfin22      bcf     portb,led3      ;LED off                                   NORMAL MODE RETAINED
            goto    ntestopto       ;want to get opto=0 - stay in normal mode - go to ntestopto to wait for state change
nstill2     bcf     portb,led3      ;stopped or very slow turning - opto=1
            movlw   0x82            ;load working register with value 26dec        state still opto=1 - total
                                    ;delay=1micro*26*255*75*6executions=2.9835sec
            movwf   del111          ;load this value into del111
n2xdeelayb  movlw   0xff            ;load working register with value 255dec
            movwf   del11           ;load this value into del11
n2xdeeelayb movlw   0x4b            ;load working register with value 75dec
            movwf   del1            ;load this value into del1
n2xdlayb    decfsz  del1,1          ;decrement del1 and keep new value, skip next line if zero
            goto    n2xdlayb2       ;go and test opto for a change in state
            goto    n2xdlaybb       ;
n2xdlaybb   decfsz  del11,1         ;del1 zero - decrement del11 and keep new value, skip next line if zero
            goto    n2xdeeelayb     ;del11 not zero, reload and decrement del1 for correct time delay sequence
            goto    n2doesitstb     ;
n2doesitstb decfsz  del111,1        ;del11 zero -  decrement del111 and keep new value, skip next line if zero
            goto    n2xdeelayb      ;del111 not zero - reload and decrement del1 and del11 as before for correct time delay sequence
            goto    n2refire2a      ;opto state not changed within time limit so motor has stopped - go to nrefire2a coding     change firing routine
n2xdlayb2   btfsc   portb,6         ;test the opto for a change in state
            goto    n2xdlayb        ;opto=1 still thus continue countdown  - go to xdlayb coding
            goto    n2goon4         ;opto=0 thus opto state still changing and hence motor still turning - go to goon4 coding to allow motor
                                    ; to turn still
n2goon4     movlw   0x5c            ;give w new value of 5c(hex)so can change to next firing routine with no problems - motor is still turning
            movwf   motval          ;put 5c(hex) into motval so circuit can switch phases on and off correctly for each opto state
n2goon4a    btfsc   portb,strtbut   ;test for start button still being on
            goto    chzer           ;button off so stop motor
            btfsc   portb,0         ;if the user wants to set more than one A/D time value before updating, then use this
            goto    nfire1          ;go to coding labelled nfire1
            goto    chzer           ;timing can be altered for all channels as long as the master switch is on
n2refire2a  bsf     portb,led3      ;motor has stopped turning - opto=0 - turn on LED as a warning light - now switch off all phases to avoid
                                    ; burnout of windings
            bcf     portd,lhslsw1   ;turn off phaseA, bottom switch
            bcf     portd,lhshsw1   ;turn off phaseA, top switch
            bcf     portd,rhslsw2   ;turn off phaseB, bottom switch
            bcf     portd,rhshsw2   ;turn off phaseB, top switch
n2refire2   btfsc   portb,6         ;test opto for being set i.e. output logic 0 - motor must turn to do this
            goto    n2refire22      ;opto=1 - no state change - check again - but A/D conversion check first
            goto    n2fire1more     ;opto=0 - state has changed - so try to turn motor again - go to line below
n2fire1more bcf     portb,led3      ;switch LED off
            movlw   0x5c            ;
            movwf   motval          ;5c to motval
            goto    nfire1          ;go to coding labelled fire1 to start motor again
n2refire22  goto    refdisplay      ;although motor has stopped, the A/D timing alterations can still continue
n2refir22   btfsc   portb,strtbut   ;start button on?
            goto    chzer           ;no so stop trying to turn motor
            goto    n2refire2a      ;go back to checking for the change of opto state - go to coding labelled refire2


nxdelay1    btfsc   portb,3         ;3 pink wires are used to select which A/D channel is to be shown, which is detemined below
            goto    n1disaa         ;1/3/5/7      channels applicable
            btfsc   portb,2         ;0/2/4/6      channels applicable
            goto    n1disbb         ;2/6          channels applicable
            btfsc   porta,4         ;0/4          channels applicable
            goto    nxd1            ;4out         channel 4 chosen
            goto    nzeroa          ;0out         channel 0 chosen
n1disaa     btfsc   portb,2         ;1/3/5/7      channels applicable
            goto    n1discc         ;3/7          channels applicable
            btfsc   porta,4         ;1/5          channels applicable
            goto    nxd1            ;5out         channel 5 chosen
```

```
        goto    nonea           ;1out   channel 1 chosen
n1disbb btfsc   porta,4         ;2/6    channels applicable
        goto    nxd1            ;6out   channel 6 chosen
        goto    ntwoa           ;2out   channel 2 chosen
n1discc btfsc   porta,4         ;3/7    channels applicable
        goto    nxd1            ;7out   channel 7 chosen
        goto    nthreea         ;3out   channel 3 chosen
nzeroa  bcf     portd,7         ;show channel number 0 using 3 leds        channel 0 chosen
        bcf     portd,6         ;show channel number 0 using 3 leds
        bcf     portd,5         ;show channel number 0 using 3 leds
        goto    nchzera         ;A/D conversion for that channel and store the value in different file register to normal (val0a instead of val0)
nzeroaa btfsc   portb,refresh   ;test the bit for the requirement  to replace the contents of val0 with val0a - allows alterartion of timing
                                ; while program is running
        goto    nxd1            ;no requirement to refresh so ignore value of val0a and keep val0 as it is  - go to testopt
        movwf   val0            ;requirement to refresh, so val0 is given the value of val0a which is shown on the LED sequence
        goto    nxd1            ;timing has been carried out and so continue with which motor phase firing sequence for normal operation
                                ; - go to testopt
nonea   bsf     portd,7         ;show channel number 1 using 3 leds        channel 1 chosen
        bcf     portd,6         ;show channel number 1 using 3 leds
        bcf     portd,5         ;show channel number 1 using 3 leds
        goto    nchonea         ;A/D conversion for that channel and store the value in different file register to normal (val1a instead of val1)
noneaa  btfsc   portb,refresh   ;test the bit for the requirement  to replace the contents of val1 with val1a - allows alterartion of timing
                                ;while program is running
        goto    nxd1            ;no requirement to refresh so ignore value of val1a and keep val1 as it is - go to testopt
        movwf   val1            ;requirement to refresh, so val1 is given the value of val1a which is shown on the LED sequence
        goto    nxd1            ;timing has been carried out and so continue with which motor phase firing sequence for normal operation
                                ; - go to testopt
ntwoa   bcf     portd,7         ;show channel number 2 using 3 leds        channel 2 chosen
        bsf     portd,6         ;show channel number 2 using 3 leds
        bcf     portd,5         ;show channel number 2 using 3 leds
        goto    nchtwoa         ;A/D conversion for that channel and store the value in different file register to normal (val2a instead of val2)
ntwoaa  btfsc   portb,refresh   ;test the bit for the requirement  to replace the contents of val2 with val2a - allows alterartion
                                ;of timing while program is running
        goto    nxd1            ;no requirement to refresh so ignore value of val1a and keep val2 as it is - go to testopt
        movwf   val2            ;requirement to refresh, so val2 is given the value of val2a which is shown on the LED sequence
        goto    nxd1            ;timing has been carried out and so continue with which motor phase firing sequence for normal operation
                                ; - go to testopt
nthreea bsf     portd,7         ;show channel number 2 using 3 leds        channel 2 chosen
        bsf     portd,6         ;show channel number 2 using 3 leds
        bcf     portd,5         ;show channel number 2 using 3 leds
        goto    nchthreea       ;A/D conversion for that channel and store the value in different file register to normal (val2a instead of val2)
nthreeaa btfsc  portb,refresh   ;test the bit for the requirement  to replace the contents of val2 with val2a - allows alterartion of
                                ;timing while program is running
        goto    nxd1            ;no requirement to refresh so ignore value of val1a and keep val2 as it is - go to testopt
        movwf   val3            ;requirement to refresh, so val2 is given the value of val2a which is shown on the LED sequence
        goto    nxd1            ;timing has been carried out and so continue with which motor phase firing sequence for normal operation
                                ; - go to testopt
nchzera bcf     adcon0,3        ;select channel zero for A/D conversion
        bcf     adcon0,4        ;select channel zero for A/D conversion
        bcf     adcon0,5        ;select channel zero for A/D conversion
        call    conv            ;goto conv forA/D conversion oft channel selected then return to line below
        movf    adresl,w        ;A/D value is in adresl - move it to working register
        movwf   val0a           ;move the value in the working register to val0a - A/D for that channel now in the different file register
                                ; for that channel
        sublw   b'00000000'     ;subtract w from zero - if val? is above zero, the result will be negative
        btfsc   status,0        ;test the carry bit, if set, then val5 is zero and makes the time delay wrong, if val? is above zero, then skip
                                ;the next line and continue
        goto    chzer           ;val5 is zero so go back to the beginning and fix the value
        movf    adresl,w        ;A/D value is in adresl - move it to working register
        movwf   portc           ;show the output from this A/D conversion in the LED sequence
        goto    nzeroaa         ;go back to the A/D channel selection and see whether this value is to replace the original value

nchonea bsf     adcon0,3        ;select channel one for A/D conversion
        bcf     adcon0,4        ;select channel one for A/D conversion
        bcf     adcon0,5        ;select channel one for A/D conversion
        call    conv            ;goto conv forA/D conversion oft channel selected then return to line below
        movf    adresl,w        ;A/D value is in adresl - move it to working register
        movwf   val1a           ;move the value in the working register to val1a - A/D for that channel now in the different file register
                                ;for that channel
        sublw   b'00000000'     ;subtract w from zero - if val? is above zero, the result will be negative
        btfsc   status,0        ;test the carry bit, if set, then val5 is zero and makes the time delay wrong, if val? is above zero, then skip
                                ;the next line and continue
        goto    chzer           ;val5 is zero so go back to the beginning and fix the value
        movf    adresl,w        ;A/D value is in adresl - move it to working register
        movwf   portc           ;show the output from this A/D conversion in the LED sequence
        goto    noneaa          ;go back to the A/D channel selection and see whether this value is to replace the original value
nchtwoa bcf     adcon0,3        ;select channel two for A/D conversion
        bsf     adcon0,4        ;select channel two for A/D conversion
        bcf     adcon0,5        ;select channel two for A/D conversion
        call    conv            ;goto conv forA/D conversion oft channel selected then return to line below
        movf    adresl,w        ;A/D value is in adresl - move it to working register
        movwf   val2a           ;move the value in the working register to val2a - A/D for that channel now in the different file register
                                ;for that channel
        movwf   portc           ;show the output from this A/D conversion in the LED sequence
        sublw   b'00000000'     ;subtract w from zero - if val? is above zero, the result will be negative
        btfsc   status,0        ;test the carry bit, if set, then val5 is zero and makes the time delay wrong, if val? is above zero, then skip
                                ; the next line and continue
        goto    chzer           ;val5 is zero so go back to the beginning and fix the value
        movf    adresl,w        ;A/D value is in adresl - move it to working register
        goto    ntwoaa          ;go back to the A/D channel selection and see whether this value is to replace the original value
nchthreea bsf   adcon0,3        ;select channel two for A/D conversion
        bsf     adcon0,4        ;select channel two for A/D conversion
        bcf     adcon0,5        ;select channel two for A/D conversion
        call    conv            ;goto conv forA/D conversion oft channel selected then return to line below
        movf    adresl,w        ;A/D value is in adresl - move it to working register
        movwf   val3a           ;move the value in the working register to val2a - A/D for that channel now in the different file register
                                ; for that channel
        movwf   portc           ;show the output from this A/D conversion in the LED sequence
        sublw   b'00000000'     ;subtract w from zero - if val? is above zero, the result will be negative
        btfsc   status,0        ;test the carry bit, if set, then val5 is zero and makes the time delay wrong, if val? is above zero, then skip
                                ; the next line and continue
```

```
                goto     chzer          ;val5 is zero so go back to the beginning and fix the value
                movf     adresl,w       ;A/D value is in adresl - move it to working register
                goto     nthreeaa       ;go back to the A/D channel selection and see whether this value is to replace the original value
nxd1            btfss    portb,0        ;master on?
                goto     chzer          ;yes, so restart it all as new values wanted
                movf     val0,w         ;move the value of the channel0 A/D conversion into the working register
                movwf    del111         ;put this value in the file register del111
nxdeelay1       movf     val1,w         ;move the value of the channel1 A/D conversion into the working register
                movwf    del11          ;move this value into the del11 register
;nxdeeelay1     movlw    0x02           ;move the value of the 10 into the working register
;               movwf    del1           ;move this value into the del1 register
nxdlay1         decfsz   del11,1        ;decrement the value in del1 by one and keep the new value, skip the next line if zero
                goto     nxdlay1        ;del1 not zero - go and decrement again
                decfsz   del111,1       ;del1 is zero - decrement the value in del11 by one and keep the new value, skip the next line if zero
                goto     nxdeelay1      ;del11 not zero - reload and decrement del1 again
;               decfsz   del111,1       ;del11 is zero - decrement the value in del111 by one and keep the new value, skip the next line if zero
;               goto     nxdeelay1      ;del111 not zero -  reload and decrement del1 and del11 again
                return                  ;del111 is zero - end of time delay sequence          total delay=1micro*5**3executions=sec?


refdisplay      btfsc    portb,3        ;3 pink wires are used to select which A/D channel is to be shown, which is detemined below
                goto     rdisaa         ;1/3/5/7      channels applicable
                btfsc    portb,2        ;0/2/4/6      channels applicable
                goto     rdisbb         ;2/6          channels applicable
                btfsc    porta,4        ;0/4          channels applicable
                goto     rfoura         ;4out         channel 4 chosen
                goto     rzeroa         ;0out         channel 0 chosen
rdisaa          btfsc    portb,2        ;1/3/5/7      channels applicable
                goto     rdiscc         ;3/7          channels applicable
                btfsc    porta,4        ;1/5          channels applicable
                goto     rfivea         ;5out         channel 5 chosen
                goto     ronea          ;1out         channel 1 chosen
rdisbb          btfsc    porta,4        ;2/6          channels applicable
                goto     rsixa          ;6out         channel 6 chosen
                goto     rtwoa          ;2out         channel 2 chosen
rdiscc          btfsc    porta,4        ;3/7          channels applicable
                goto     rsevena        ;7out         channel 7 chosen
                goto     rthreea        ;3out         channel 3 chosen
rzeroa          bcf      portd,7        ;show channel number 0 using 3 leds                    channel 0 chosen
                bcf      portd,6        ;show channel number 0 using 3 leds
                bcf      portd,5        ;show channel number 0 using 3 leds
                goto     rchzera        ;A/D conversion for that channel and store the value in different file register to normal (val0a instead of val0)
rzeroaa         btfsc    portb,refresh  ;test the bit for the requirement  to replace the contents of val0 with val0a - allows alterartion of timing
                                        ; while program is running
                goto     rstill         ;no requirement to refresh so ignore value of val0a and keep val0 as it is  - go to testopt
                movwf    val0           ;requirement to refresh, so val0 is given the value of val0a which is shown on the LED sequence
                goto     rstill         ;timing has been carried out and so continue with which motor phase firing sequence for normal operation
                                        ; - go to testopt
ronea           bsf      portd,7        ;show channel number 1 using 3 leds                    channel 1 chosen
                bcf      portd,6        ;show channel number 1 using 3 leds
                bcf      portd,5        ;show channel number 1 using 3 leds
                goto     rchonea        ;A/D conversion for that channel and store the value in different file register to normal (val1a instead of val1)
roneaa          btfsc    portb,refresh  ;test the bit for the requirement  to replace the contents of val1 with val1a - allows alterartion of timing
                                        ;while program is running
                goto     rstill         ;no requirement to refresh so ignore value of val1a and keep val1 as it is - go to testopt
                movwf    val1           ;requirement to refresh, so val1 is given the value of val1a which is shown on the LED sequence
                goto     rstill         ;timing has been carried out and so continue with which motor phase firing sequence for normal operation
                                        ; - go to testopt
rtwoa           bcf      portd,7        ;show channel number 2 using 3 leds                    channel 2 chosen
                bsf      portd,6        ;show channel number 2 using 3 leds
                bcf      portd,5        ;show channel number 2 using 3 leds
                goto     rchtwoa        ;A/D conversion for that channel and store the value in different file register to normal (val2a instead of val2)
rtwoaa          btfsc    portb,refresh  ;test the bit for the requirement  to replace the contents of val2 with val2a - allows alterartion of timing
                                        ; while program is running
                goto     rstill         ;no requirement to refresh so ignore value of val1a and keep val2 as it is - go to testopt
                movwf    val2           ;requirement to refresh, so val2 is given the value of val2a which is shown on the LED sequence
                goto     rstill         ;timing has been carried out and so continue with which motor phase firing sequence for normal operation
                                        ;- go to testopt
rthreea         bsf      portd,7        ;show channel number 3 using 3 leds                    channel 3 chosen
                bsf      portd,6        ;show channel number 3 using 3 leds
                bcf      portd,5        ;show channel number 3 using 3 leds
                goto     rchthra        ;A/D conversion for that channel and store the value in different file register to normal (val3a instead of val3)
rthreeaa        btfsc    portb,refresh  ;test the bit for the requirement  to replace the contents of val3 with val3a - allows alterartion of timing
                                        ;while program is ;running
                goto     rstill         ;no requirement to refresh so ignore value of val3a and keep val3 as it is - go to testopt
                movwf    val3           ;requirement to refresh, so val3 is given the value of val3a which is shown on the LED sequence
                goto     rstill         ;timing has been carried out and so continue with which motor phase firing sequence for normal operation
                                        ; - go to testopt
rfoura          bcf      portd,7        ;show channel number 4 using 3 leds                    channel 4 chosen
                bcf      portd,6        ;show channel number 4 using 3 leds
                bsf      portd,5        ;show channel number 4 using 3 leds
                goto     rchfoua        ;A/D conversion for that channel and store the value in different file register to normal (val4a instead of val4)
rfouraa         btfsc    portb,refresh  ;test the bit for the requirement  to replace the contents of val4 with val4a - allows alterartion of timing
                                        ; while program is running
                goto     rstill         ;no requirement to refresh so ignore value of val4a and keep val4 as it is - go to testopt
                movwf    val4           ;requirement to refresh, so val4 is given the value of val4a which is shown on the LED sequence
                goto     rstill         ;timing has been carried out and so continue with which motor phase firing sequence for normal operation
                                        ; - go to testopt
rfivea          bsf      portd,7        ;show channel number 5 using 3 leds                    channel 5 chosen
                bcf      portd,6        ;show channel number 5 using 3 leds
                bsf      portd,5        ;show channel number 5 using 3 leds
                goto     rchfiva        ;A/D conversion for that channel and store the value in different file register to normal (val4a instead of val4)
rfiveaa         btfsc    portb,refresh  ;test the bit for the requirement  to replace the contents of val4 with val4a - allows alterartion of timing
                                        ;while program is running
                goto     rstill         ;no requirement to refresh so ignore value of val5a and keep val5 as it is - go to testopt
                movwf    val5           ;requirement to refresh, so val5 is given the value of val5a which is shown on the LED sequence
                goto     rstill         ;timing has been carried out and so continue with which motor phase firing sequence for normal operation
                                        ;- go to testopt
rsixa           bcf      portd,7        ;show channel number 6 using 3 leds                    channel 6 chosen
                bsf      portd,6        ;show channel number 6 using 3 leds
                bsf      portd,5        ;show channel number 6 using 3 leds
```

```
        goto    rchsixa      ;A/D conversion for that channel and store the value in different file register to normal (val6a instead of val6)
rsixaa  btfsc   portb,refresh ;test the bit for the requirement to replace the contents of val6 with val6a - allows alterartion of timing
                             ;while program is running
        goto    rstill       ;no requirement to refresh so ignore value of val6a and keep val6 as it is - go to testopt
        movwf   val6         ;requirement to refresh, so val6 is given the value of val6a which is shown on the LED sequence
        goto    rstill       ;timing has been carried out and so continue with which motor phase firing sequence for normal operation
                             ; - go to testopt
rsevena bsf     portd,7      ;show channel number 7 using 3 leds                    channel 7 chosen
        bsf     portd,6      ;show channel number 7 using 3 leds
        bsf     portd,5      ;show channel number 7 using 3 leds
        goto    rchseva      ;A/D conversion for that channel and store the value in different file register to normal (val7a instead of val7)
rsevenaa btfsc  portb,refresh ;test the bit for the requirement to replace the contents of val7 with val7a - allows alterartion of timing
                             ;while program is running
        goto    rstill       ;no requirement to refresh so ignore value of val7a and keep val7 as it is - go to testopt
        movwf   val7         ;requirement to refresh, so val7 is given the value of val7a which is shown on the LED sequence
        goto    rstill       ;timing has been carried out and so continue with which motor phase firing sequence for normal operation
                             ; - go to testopt
rchzera bcf     adcon0,3     ;select channel zero for A/D conversion
        bcf     adcon0,4     ;select channel zero for A/D conversion
        bcf     adcon0,5     ;select channel zero for A/D conversion
        call    conv         ;goto conv forA/D conversion oft channel selected then return to line below
        movf    adresl,w     ;A/D value is in adresl - move it to working register
        movwf   val0a        ;move the value in the working register to val0a - A/D for that channel now in the different file register
                             ;for that channel
        movwf   portc        ;show the output from this A/D conversion in the LED sequence
        sublw   b'00000000'  ;subtract w from zero - if val? is above zero, the result will be negative
        btfsc   status,0     ;test the carry bit, if set, then val5 is zero and makes the time delay wrong, if val? is above zero, then
                             ;skip the next line and continue
        goto    chzer        ;val? is zero so go back to the beginning and fix the value
        movf    adresl,w     ;A/D value is in adresl - move it to working register
        goto    rzeroaa      ;go back to the A/D channel selection and see whether this value is to replace the original value

rchonea bsf     adcon0,3     ;select channel one for A/D conversion
        bcf     adcon0,4     ;select channel one for A/D conversion
        bcf     adcon0,5     ;select channel one for A/D conversion
        call    conv         ;goto conv forA/D conversion oft channel selected then return to line below
        movf    adresl,w     ;A/D value is in adresl - move it to working register
        movwf   val1a        ;move the value in the working register to val1a - A/D for that channel now in the different file register
                             ;for that channel
        movwf   portc        ;show the output from this A/D conversion in the LED sequence
        sublw   b'00000000'  ;subtract w from zero - if val? is above zero, the result will be negative
        btfsc   status,0     ;test the carry bit, if set, then val5 is zero and makes the time delay wrong, if val? is above zero, then skip
                             ;the next line and continue
        goto    chzer        ;val? is zero so go back to the beginning and fix the value
        movf    adresl,w     ;A/D value is in adresl - move it to working register
        goto    roneaa       ;go back to the A/D channel selection and see whether this value is to replace the original value
rchtwoa bcf     adcon0,3     ;select channel two for A/D conversion
        bsf     adcon0,4     ;select channel two for A/D conversion
        bcf     adcon0,5     ;select channel two for A/D conversion
        call    conv         ;goto conv forA/D conversion oft channel selected then return to line below
        movf    adresl,w     ;A/D value is in adresl - move it to working register
        movwf   val2a        ;move the value in the working register to val2a - A/D for that channel now in the different file register
                             ; for that channel
        movwf   portc        ;show the output from this A/D conversion in the LED sequence
        sublw   b'00000000'  ;subtract w from zero - if val? is above zero, the result will be negative
        btfsc   status,0     ;test the carry bit, if set, then val5 is zero and makes the time delay wrong, if val? is above zero, then skip
                             ;the next line and continue
        goto    chzer        ;val? is zero so go back to the beginning and fix the value
        goto    rtwoaa       ;go back to the A/D channel selection and see whether this value is to replace the original value
rchthra bsf     adcon0,3     ;select channel three for A/D conversion
        bsf     adcon0,4     ;select channel three for A/D conversion
        bcf     adcon0,5     ;select channel three for A/D conversion
        call    conv         ;goto conv forA/D conversion oft channel selected then return to line below
        movf    adresl,w     ;A/D value is in adresl - move it to working register
        movwf   val3a        ;move the value in the working register to val3a - A/D for that channel now in the different file
                             ; register for that channel
        movwf   portc        ;show the output from this A/D conversion in the LED sequence
        sublw   b'00000000'  ;subtract w from zero - if val? is above zero, the result will be negative
        btfsc   status,0     ;test the carry bit, if set, then val5 is zero and makes the time delay wrong, if val? is above zero, then
                             ;skip the next line and continue
        goto    chzer        ;val? is zero so go back to the beginning and fix the value
        movf    adresl,w     ;A/D value is in adresl - move it to working register
        goto    rthreeaa     ;go back to the A/D channel selection and see whether this value is to replace the original value
rchfoua bcf     adcon0,3     ;select channel four for A/D conversion
        bcf     adcon0,4     ;select channel four for A/D conversion
        bsf     adcon0,5     ;select channel four for A/D conversion
        call    conv         ;goto conv forA/D conversion oft channel selected then return to line below
        movf    adresl,w     ;A/D value is in adresl - move it to working register
        movwf   val4a        ;move the value in the working register to val4a - A/D for that channel now in the different file register
                             ;for that channel
        movwf   portc        ;show the output from this A/D conversion in the LED sequence
        sublw   b'00000000'  ;subtract w from zero - if val? is above zero, the result will be negative
        btfsc   status,0     ;test the carry bit, if set, then val5 is zero and makes the time delay wrong, if val? is above zero, then
                             ;skip the next line and continue
        goto    chzer        ;val? is zero so go back to the beginning and fix the value
        movf    adresl,w     ;A/D value is in adresl - move it to working register
        goto    rfouraa      ;go back to the A/D channel selection and see whether this value is to replace the original value
rchfiva bsf     adcon0,3     ;select channel five for A/D conversion
        bcf     adcon0,4     ;select channel five for A/D conversion
        bsf     adcon0,5     ;select channel five for A/D conversion
        call    conv         ;goto conv forA/D conversion oft channel selected then return to line below
        movf    adresl,w     ;A/D value is in adresl - move it to working register
        movwf   val5a        ;move the value in the working register to val5a - A/D for that channel now in the different file
                             ; register for that channel
        movwf   portc        ;show the output from this A/D conversion in the LED sequence
        sublw   b'00000000'  ;subtract w from zero - if val? is above zero, the result will be negative
        btfsc   status,0     ;test the carry bit, if set, then val5 is zero and makes the time delay wrong, if val? is above zero, then skip
                             ;the next line and continue
        goto    chzer        ;val? is zero so go back to the beginning and fix the value
        movf    adresl,w     ;A/D value is in adresl - move it to working register
        goto    rfiveaa      ;go back to the A/D channel selection and see whether this value is to replace the original value
```

```
rchsixa     bcf     adcon0,3        ;select channel six for A/D conversion
            bsf     adcon0,4        ;select channel six for A/D conversion
            bsf     adcon0,5        ;select channel six for A/D conversion
            call    conv            ;goto conv forA/D conversion oft channel selected then return to line below
            movf    adresl,w        ;A/D value is in adresl - move it to working register
            movwf   val6a           ;move the value in the working register to val6a - A/D for that channel now in the different file register
                                    ; for that channel
            movwf   portc           ;show the output from this A/D conversion in the LED sequence
            sublw   b'00000000'     ;subtract w from zero - if val? is above zero, the result will be negative
            btfsc   status,0        ;test the carry bit, if set, then val5 is zero and makes the time delay wrong, if val? is above zero, then skip
                                    ;the next line and continue
            goto    chzer           ;val? is zero so go back to the beginning and fix the value
            movf    adresl,w        ;A/D value is in adresl - move it to working register
            goto    rsixaa          ;go back to the A/D channel selection and see whether this value is to replace the original value
rchseva     bsf     adcon0,3        ;select channel seven for A/D conversion
            bsf     adcon0,4        ;select channel seven for A/D conversion
            bsf     adcon0,5        ;select channel seven for A/D conversion
            call    conv            ;goto conv forA/D conversion oft channel selected then return to line below
            movf    adresl,w        ;A/D value is in adresl - move it to working register
            movwf   val7a           ;move the value in the working register to val7a - A/D for that channel now in the different file register
                                    ; for that channel
            movwf   portc           ;show the output from this A/D conversion in the LED sequence
            sublw   b'00000000'     ;subtract w from zero - if val? is above zero, the result will be negative
            btfsc   status,0        ;test the carry bit, if set, then val5 is zero and makes the time delay wrong, if val? is above zero, then skip
                                    ;the next line and continue
            goto    chzer           ;val? is zero so go back to the beginning and fix the value
            movf    adresl,w        ;A/D value is in adresl - move it to working register
            goto    rsevenaa        ;go back to the A/D channel selection and see whether this value is to replace the original value
rstill      movf    motval,w        ;opto=1         bring back value of motval to working rgister,w to determine how firing is to occur
            sublw   0x5b            ;0x5b-w                 just a calculation to determine if can continue or if the motor has stopped
            btfsc   status,0        ;check value of carry   the carry bit shows the result to be +'ve or -'ve
            goto    n1refir11       ;+'ve(carry bit set)            0x5b-0x5a       go back to still coding
            goto    n2refir22       ;-'ve(carry bit clear)  0x5b-0x5c       go back to still coding


advanced    btfsc   portb,fast      ;can it go to adv mode?
            goto    norm            ;no, so go to norm
            goto    accel           ;yes, so adv mode timing
norm        bsf     portb,led3      ;LED on to show not allowed to be in this mode
            goto    ntestopto       ;can't go to advance mode as not allowed to, so go back to start
accel       bcf     portd,lhslsw1   ;turn off phaseA, bottom switch
            bcf     portd,lhshsw1   ;turn off phaseA, top switch
            bcf     portd,rhslsw2   ;turn off phaseB, bottom switch
            bcf     portd,rhshsw2   ;turn off phaseB, top switch
            bsf     portb,phase     ;advance phase
            btfss   portb,0         ;master button on?
            goto    chzer           ;yes so continue
            btfsc   portb,fast      ;can it go to adv mode?
            goto    norm            ;no, so go to norm
            bcf     portb,led3      ;LED off to show allowed to be in this mode
            btfsc   portb,strtbut   ;test for start button being on
            goto    chzer           ;off
            goto    fopto           ;on
fopto       btfss   portb,6         ;test current state of opto. Can only begin firing at new change of opto state
            goto    fastclr         ;opto clear - now wait til set
            goto    fastset         ;opto set - now wait til clear
fastclr     btfss   portb,6         ;test for set
            goto    fastclr         ;still clear
            goto    setfast         ;set so start firing in adv mode        opto=1
fastset     btfsc   portb,6         ;test for clear
            goto    fastset         ;still set
            goto    clrfast         ;clear so start firing in adv mode      opto=0

setfast     bsf     portd,rhslsw2   ;turn on phaseB, bottom switch                  opto=1
            bsf     portd,rhshsw2   ;turn on phaseB, top switch
setfast1    movlw   0x5a            ;opto=0  put 5a(hex) into working register
            movwf   motval          ;store 5a in motval file register - motval used to detect how phase firing and opto position are linked
            btfsc   portb,refresh   ;update the value?
            goto    f1refre
            bsf     portb,phase     ;advance phase
            bcf     adcon0,3        ;select channel 4 for A/D conversion
            bcf     adcon0,4        ;select channel 4 for A/D conversion
            bsf     adcon0,5        ;select channel 4 for A/D conversion
            call    conv            ;goto conv for A/D conversion of channel selected then return to line below
            movf    adresl,w        ;A/D value is in adresl - move it to working register
            movwf   val4a           ;move the value in the working register to val4a - A/D for that channel now in the file register for that channel
            sublw   b'00000000'     ;subtract w from zero - if val? is above zero, the result will be negative
            btfsc   status,0        ;test the carry bit, if set, then val5 is zero and makes the time delay wrong, if val? is above zero, then
                                    ;skip the next line and continue
            goto    chzer           ;val? is zero so go back to the beginning and fix the value
            bsf     adcon0,3        ;select channel 5 for A/D conversion
            bcf     adcon0,4        ;select channel 5 for A/D conversion
            bsf     adcon0,5        ;select channel 5 for A/D conversion
            call    conv            ;goto conv for A/D conversion of channel selected then return to line below
            movf    adresl,w        ;A/D value is in adresl - move it to working register
            movwf   val5a           ;move the value in the working register to val5a - A/D for that channel now in the file register for that channel
            sublw   b'00000000'     ;subtract w from zero - if val? is above zero, the result will be negative
            btfsc   status,0        ;test the carry bit, if set, then val5 is zero and makes the time delay wrong, if val? is above zero, then skip
                                    ;the next line and continue
            goto    chzer           ;val? is zero so go back to the beginning and fix the value
            bcf     adcon0,3        ;select channel 6 for A/D conversion
            bsf     adcon0,4        ;select channel 6 for A/D conversion
            bsf     adcon0,5        ;select channel 6 for A/D conversion
            call    conv            ;goto conv for A/D conversion of channel selected then return to line below
            movf    adresl,w        ;A/D value is in adresl - move it to working register
            movwf   val6a           ;move the value in the working register to val6a - A/D for that channel now in the file register for that channel
            sublw   b'00000000'     ;subtract w from zero - if val? is above zero, the result will be negative
            btfsc   status,0        ;test the carry bit, if set, then val5 is zero and makes the time delay wrong, if val? is above zero, then
                                    ;skip the next line and continue
            goto    chzer           ;val? is zero so go back to the beginning and fix the value
            bsf     adcon0,3        ;select channel 7 for A/D conversion
```

```
                bsf     adcon0,4        ;select channel 7 for A/D conversion
                bsf     adcon0,5        ;select channel 7 for A/D conversion
                call    conv            ;goto conv for A/D conversion of channel selected then return to line below
                movf    adresl,w        ;A/D value is in adresl - move it to working register
                movwf   val7a           ;move the value in the working register to val7a - A/D for that channel now in the file register for that channel
                sublw   b'00000000'     ;subtract w from zero - if val? is above zero, the result will be negative
                btfsc   status,0        ;test the carry bit, if set, then val5 is zero and makes the time delay wrong, if val? is above zero, then skip
                                        ;the next line and continue
                goto    chzer           ;val? is zero so go back to the beginning and fix the value
fastdisp1       btfsc   portb,3         ;3 pink wires are used to select which A/D channel is to be shown, which is detemined below
                goto    f1disa          ;1/3/5/7        channels applicable
                btfsc   portb,2         ;0/2/4/6        channels applicable
                goto    f1disb          ;2/6            channels applicable
                btfsc   porta,4         ;0/4            channels applicable
                goto    f1four          ;4out           channel 4 chosen
                goto    f1refre         ;0out           channel 0 chosen
f1disa          btfsc   portb,2         ;1/3/5/7        channels applicable
                goto    f1disc          ;3/7            channels applicable
                btfsc   porta,4         ;1/5            channels applicable
                goto    f1five          ;5out           channel 5 chosen
                goto    f1refre         ;1out           channel 1 chosen
f1disb          btfsc   porta,4         ;2/6            channels applicable
                goto    f1six           ;6out           channel 6 chosen
                goto    f1refre         ;2out           channel 2 chosen
f1disc          btfsc   porta,4         ;3/7            channels applicable
                goto    f1seven         ;7out           channel 7 chosen
                goto    f1refre         ;3out           channel 3 chosen
f1four          bcf     portd,7         ;show channel number 4 using 3 leds                     channel 4 chosen
                bcf     portd,6         ;show channel number 4 using 3 leds
                bsf     portd,5         ;show channel number 4 using 3 leds
                movf    val4a,w         ;move the value in val4a into the working register
                movwf   portc           ;move the contents of the working register to the LED sequence and so the A/D result for this channel is shown
                btfsc   portb,refresh   ;update the value?
                goto    f1fou           ;no, so keep original value
                movwf   val4            ;yes, so make val?a become val?
                goto    f1refre         ;master update?
f1fou           goto    f1refre         ;master update?
f1five          bsf     portd,7         ;show channel number 5 using 3 leds                     channel 5 chosen
                bcf     portd,6         ;show channel number 5 using 3 leds
                bsf     portd,5         ;show channel number 5 using 3 leds
                movf    val5a,w         ;move the value in val5a into the working register
                movwf   portc           ;move the contents of the working register to the LED sequence and so the A/D result for this channel is shown
                btfsc   portb,refresh   ;update the value?
                goto    f1fiv           ;no, so keep original value
                movwf   val5            ;yes, so make val?a become val?
                goto    f1refre         ;master update?
f1fiv           goto    f1refre         ;master update?
f1six           bcf     portd,7         ;show channel number 6 using 3 leds                     channel 6 chosen
                bsf     portd,6         ;show channel number 6 using 3 leds
                bsf     portd,5         ;show channel number 6 using 3 leds
                movf    val6a,w         ;move the value in val6a into the working register
                movwf   portc           ;move the contents of the working register to the LED sequence and so the A/D result for this channel is shown
                btfsc   portb,refresh   ;update the value?
                goto    f1sixe          ;no, so keep original value
                movwf   val6            ;yes, so make val?a become val?
                goto    f1refre         ;master update?
f1sixe          goto    f1refre         ;master update?
f1seven         bsf     portd,7         ;show channel number 7 using 3 leds                     channel 7 chosen
                bsf     portd,6         ;show channel number 7 using 3 leds
                bsf     portd,5         ;show channel number 7 using 3 leds
                movf    val7a,w         ;move the value in val7a into the working register
                movwf   portc           ;move the contents of the working register to the LED sequence and so the A/D result for this channel is shown
                btfsc   portb,refresh   ;update the value?
                goto    f1sev           ;no, so keep original value
                movwf   val7            ;yes, so make val?a become val?
                goto    f1refre         ;master update?
f1sev           goto    f1refre         ;master update?

f1refre         btfss   portb,0         ;is the master update button on?
                goto    chzer           ;yes, so restart everything
                goto    faststrt1       ;no, so continue
faststrt1       btfsc   portb,strtbut   ;is the start button still on?
                goto    chzer           ;no, so restart everything
                goto    fastadv1        ;yes, so check if the motor can go into advance mode
fastadv1        btfsc   portb,fast      ;can it go to adv mode?
                goto    norm            ;no, so go to norm
                goto    f1del           ;yes, so adv mode timing

f1del           movlw   0x12
                movwf   heur3
ts1a            movf    val7,w
                movwf   heur2
t1              decfsz  heur2,f
                goto    t1a
                goto    t1b
t1a             btfsc   portb,6         ;opto is 1, has it gone to 0 too early - ie is timing wrong
                goto    t1              ;opto=1 - still okay
                goto    eror1           ;opto=0 - timing wrong
t1b             decfsz  heur3,f
                goto    ts1a
                goto    fin1
eror1           bsf     portb,led3      ;LED on
                bcf     portd,rhslsw2   ;turn off phaseB, bottom switch
                bcf     portd,rhshsw2   ;turn off phaseB, top switch
                call    wdelay          ;v smalldelay
                bsf     portd,lhslsw1   ;turn on phaseA, bottom switch
                bsf     portd,lhshsw1   ;turn on phaseA, top switch
                goto    clrfast0        ;goto next section                                      CORRECT???????
fin1            bcf     portd,rhslsw2   ;turn off phaseB, bottom switch
                bcf     portd,rhshsw2   ;turn off phaseB, top switch
                goto    f1del1
```

```
fldel1      movlw    0x05
            movwf    heur3
ts11a       movf     val6,w
            movwf    heur2
t11         decfsz   heur2,f
            goto     t11a
            goto     t11b
t11a        btfsc    portb,6        ;opto is 1, has it gone to 0 too early - ie is timing wrong
            goto     t11            ;opto=1 - still okay
            goto     eror11         ;opto=0 - timing wrong
t11b        decfsz   heur3,f
            goto     ts11a
            goto     fin11
eror11      bsf      portb,led3     ;LED on                    TIMING WRONG
            bsf      portd,lhslsw1  ;turn on phaseA, bottom switch
            bsf      portd,lhshsw1  ;turn on phaseA, top switch
            goto     clrfast0       ;goto next section                              CORRECT???????
fin11       bsf      portd,lhslsw1  ;turn on phaseA, bottom switch
            bsf      portd,lhshsw1  ;turn on phaseA, top switch
            goto     quik1          ;still fast enough for advanced mode?
quik1       movf     val5,w
            movwf    heur3
ts111a      movf     val4,w
            movwf    heur2
ts111b      movlw    0x0a
            movwf    heur1
t111        decfsz   heur1,f
            goto     t111a
            goto     t111b
t111a       btfsc    portb,6        ;opto is 1, want it to go to 0 now
            goto     t111           ;opto=1 so continue countdown
            goto     clrfast11      ;opto=0 - changed within the time limit thus stay in advanced
t111b       decfsz   heur2,f
            goto     ts111b
            goto     t111c
t111c       decfsz   heur3,f
            goto     ts111a
            goto     slow1          ;too slow - go to normal mode
clrfast11   bcf      portb,led3     ;LED off                   TIMING OKAY
            goto     clrfast0


wdelay      movlw    0x0a           ;load 2dec into working register - this coding allows a time delay so flashing of LED is visible
            movwf    del111         ;move 2dec from working register into del111 file register
wtrdelay    movlw    0x02           ;load 2dec into working register
            movwf    del11          ;move 2dec from working register into del11 file register
wtdelay     movlw    0x02           ;load 2dec into working register
            movwf    del1           ;move 2dec from working register into del1 file register
wdlay       decfsz   del1,1         ;decrement del1 and keep new value, skip next line if zero
            goto     wdlay          ;del1 not zero so decrement again
            decfsz   del11,1        ;del1 is zero, decrement del11 and keep new value, skip next line if zero
            goto     wtdelay        ;del11 not zero, reload del1 again so can decrement again and maintain delay timing properly
            decfsz   del111,1       ;del11 is zero, decrement del111 and keep new value, skip next line if zero
            goto     wtrdelay       ;del111 not zero, reload del1 and del11 so can both decrement again and maintain delay timing properly
            return                  ;time delay complete - go back to line below the call strtdelay – total
                                    ;delay=1micro*2*2*2*6executions=48microsec


clrfast     bsf      portd,lhslsw1  ;turn on phaseA, bottom switch                  opto=0
            bsf      portd,lhshsw1  ;turn on phaseA, top switch
clrfast0    movlw    0x5c           ;opto=0  put 5c(hex) into working register
            movwf    motval         ;store 5c in motval file register - motval used to detect how phase firing and opto position are linked
            btfsc    portb,refresh  ;update the value?
            goto     f0refre
            bsf      portb,phase    ;advance phase
            bcf      adcon0,3       ;select channel 4 for A/D conversion
            bcf      adcon0,4       ;select channel 4 for A/D conversion
            bsf      adcon0,5       ;select channel 4 for A/D conversion
            call     conv           ;goto conv for A/D conversion of channel selected then return to line below
            movf     adresl,w       ;A/D value is in adresl - move it to working register
            movwf    val4a          ;move the value in the working register to val4a - A/D for that channel now in the file register for that channel
            sublw    b'00000000'    ;subtract w from zero - if val? is above zero, the result will be negative
            btfsc    status,0       ;test the carry bit, if set, then val5 is zero and makes the time delay wrong, if val? is above zero, then
                                    ;skip the next line and continue
            goto     chzer          ;val? is zero so go back to the beginning and fix the value
            bsf      adcon0,3       ;select channel 5 for A/D conversion
            bcf      adcon0,4       ;select channel 5 for A/D conversion
            bsf      adcon0,5       ;select channel 5 for A/D conversion
            call     conv           ;goto conv for A/D conversion of channel selected then return to line below
            movf     adresl,w       ;A/D value is in adresl - move it to working register
            movwf    val5a          ;move the value in the working register to val5a - A/D for that channel now in the file register for that channel
            sublw    b'00000000'    ;subtract w from zero - if val? is above zero, the result will be negative
            btfsc    status,0       ;test the carry bit, if set, then val5 is zero and makes the time delay wrong, if val? is above zero, then
                                    ;skip the next line and continue
            goto     chzer          ;val? is zero so go back to the beginning and fix the value
            bcf      adcon0,3       ;select channel 6 for A/D conversion
            bsf      adcon0,4       ;select channel 6 for A/D conversion
            bsf      adcon0,5       ;select channel 6 for A/D conversion
            call     conv           ;goto conv for A/D conversion of channel selected then return to line below
            movf     adresl,w       ;A/D value is in adresl - move it to working register
            movwf    val6a          ;move the value in the working register to val6a - A/D for that channel now in the file register for that channel
            sublw    b'00000000'    ;subtract w from zero - if val? is above zero, the result will be negative
            btfsc    status,0       ;test the carry bit, if set, then val5 is zero and makes the time delay wrong, if val? is above zero, then
                                    ;skip the next line and continue
            goto     chzer          ;val? is zero so go back to the beginning and fix the value
            bsf      adcon0,3       ;select channel 7 for A/D conversion
            bsf      adcon0,4       ;select channel 7 for A/D conversion
            bsf      adcon0,5       ;select channel 7 for A/D conversion
            call     conv           ;goto conv for A/D conversion of channel selected then return to line below
            movf     adresl,w       ;A/D value is in adresl - move it to working register
            movwf    val7a          ;move the value in the working register to val7a - A/D for that channel now in the file register for that channel
```

```
                sublw   b'00000000'     ;subtract w from zero - if val? is above zero, the result will be negative
                btfsc   status,0        ;test the carry bit, if set, then val5 is zero and makes the time delay wrong, if val? is above zero, then
                                        ;skip the next line and continue
                goto    chzer           ;val? is zero so go back to the beginning and fix the value
fastdisp0       btfsc   portb,3         ;3 pink wires are used to select which A/D channel is to be shown, which is detemined below
                goto    f0disa          ;1/3/5/7    channels applicable
                btfsc   portb,2         ;0/2/4/6    channels applicable
                goto    f0disb          ;2/6        channels applicable
                btfsc   porta,4         ;0/4        channels applicable
                goto    f0four          ;4out       channel 4 chosen
                goto    f0refre         ;0out       channel 0 chosen
f0disa          btfsc   portb,2         ;1/3/5/7    channels applicable
                goto    f0disc          ;3/7        channels applicable
                btfsc   porta,4         ;1/5        channels applicable
                goto    f0five          ;5out       channel 5 chosen
                goto    f0refre         ;1out       channel 1 chosen
f0disb          btfsc   porta,4         ;2/6        channels applicable
                goto    f0six           ;6out       channel 6 chosen
                goto    f0refre         ;2out       channel 2 chosen
f0disc          btfsc   porta,4         ;3/7        channels applicable
                goto    f0seven         ;7out       channel 7 chosen
                goto    f0refre         ;3out       channel 3 chosen
f0four          bcf     portd,7         ;show channel number 4 using 3 leds                channel 4 chosen
                bcf     portd,6         ;show channel number 4 using 3 leds
                bsf     portd,5         ;show channel number 4 using 3 leds
                movf    val4a,w         ;move the value in val4a into the working register
                movwf   portc           ;move the contents of the working register to the LED sequence and so the A/D result for this channel is shown
                btfsc   portb,refresh   ;update the value?
                goto    f0fou           ;no, so keep original value
                movwf   val4            ;yes, so make val?a become val?
                goto    f0refre         ;master update?
f0fou           goto    f0refre         ;master update?
f0five          bsf     portd,7         ;show channel number 5 using 3 leds                channel 5 chosen
                bcf     portd,6         ;show channel number 5 using 3 leds
                bsf     portd,5         ;show channel number 5 using 3 leds
                movf    val5a,w         ;move the value in val5a into the working register
                movwf   portc           ;move the contents of the working register to the LED sequence and so the A/D result for this channel is shown
                btfsc   portb,refresh   ;update the value?
                goto    f0fiv           ;no, so keep original value
                movwf   val5            ;yes, so make val?a become val?
                goto    f0refre         ;master update?
f0fiv           goto    f0refre         ;master update?
f0six           bcf     portd,7         ;show channel number 6 using 3 leds                channel 6 chosen
                bsf     portd,6         ;show channel number 6 using 3 leds
                bsf     portd,5         ;show channel number 6 using 3 leds
                movf    val6a,w         ;move the value in val6a into the working register
                movwf   portc           ;move the contents of the working register to the LED sequence and so the A/D result for this channel is shown
                btfsc   portb,refresh   ;update the value?
                goto    f0sixe          ;no, so keep original value
                movwf   val6            ;yes, so make val?a become val?
                goto    f0refre         ;master update?
f0sixe          goto    f0refre         ;master update?
f0seven         bsf     portd,7         ;show channel number 7 using 3 leds                channel 7 chosen
                bsf     portd,6         ;show channel number 7 using 3 leds
                bsf     portd,5         ;show channel number 7 using 3 leds
                movf    val7a,w         ;move the value in val7a into the working register
                movwf   portc           ;move the contents of the working register to the LED sequence and so the A/D result for this channel is shown
                btfsc   portb,refresh   ;update the value?
                goto    f0sev           ;no, so keep original value
                movwf   val7            ;yes, so make val?a become val?
                goto    f0refre         ;master update?
f0sev           goto    f0refre         ;master update?

f0refre         btfss   portb,0         ;is the master update button on?
                goto    chzer           ;yes, so restart everything
                goto    faststrt0       ;no, so continue
faststrt0       btfsc   portb,strtbut   ;is the start button still on?
                goto    chzer           ;no, so restart everything
                goto    fastadv0        ;yes, so check if the motor can go into advance mode
fastadv0        btfsc   portb,fast      ;can it go to adv mode?
                goto    norm            ;no, so go to norm
                goto    f0del           ;yes, so adv mode timing

f0del           movlw   0x12
                movwf   heur3
ts0a            movf    val7,w
                movwf   heur2
t0              decfsz  heur2,f
                goto    t0a
                goto    t0b
t0a             btfss   portb,6         ;opto is 0 has it gone to 1 too early - ie is timing wrong
                goto    t0              ;opto=0 - still okay
                goto    eror0           ;opto=1 - timing wrong
t0b             decfsz  heur3,f
                goto    ts0a
                goto    fin0
eror0           bsf     portb,led3      ;LED on
                bcf     portd,lhslsw1   ;turn off phaseA, bottom switch
                bcf     portd,lhshsw1   ;turn off phaseA, top switch
                call    wdelay          ;v smalldelay
                bsf     portd,rhslsw2   ;turn on phaseB, bottom switch
                bsf     portd,rhshsw2   ;turn on phaseB, top switch
                goto    setfast1        ;goto next section                                 CORRECT???????
fin0            bcf     portd,lhslsw1   ;turn off phaseA, bottom switch
                bcf     portd,lhshsw1   ;turn off phaseA, top switch
                goto    f0del0
f0del0          movlw   0x05
                movwf   heur3
ts00a           movf    val6,w
                movwf   heur2
t00             decfsz  heur2,f
```

```
            goto        t00a
            goto        t00b
t00a        btfss       portb,6         ;opto is 0, has it gone to 1 too early - ie is timing wrong
            goto        t00             ;opto=0 - still okay
            goto        eror00          ;opto=1 - timing wrong
t00b        decfsz      heur3,f
            goto        ts00a
            goto        fin00
eror00      bsf         portb,led3      ;LED on                 TIMING WRONG
            bsf         portd,rhslsw2   ;turn on phaseB, bottom switch
            bsf         portd,rhshsw2   ;turn on phaseB, top switch
            goto        setfast1        ;goto next section                                  CORRECT???????
fin00       bsf         portd,rhslsw2   ;turn on phaseB, bottom switch
            bsf         portd,rhshsw2   ;turn on phaseB, top switch
            goto        quik0           ;still fast enough for advanced mode?
quik0       movf        val5,w
            movwf       heur3
ts000a      movf        val4,w
            movwf       heur2
ts000b      movlw       0x0a
            movwf       heur1
t000        decfsz      heur1,f
            goto        t000a
            goto        t000b
t000a       btfss       portb,6         ;opto is 0, want it to go to 1 now
            goto        t000            ;opto=0 so continue countdown
            goto        setfast00       ;opto=1 - changed within the time limit thus stay in advanced
t000b       decfsz      heur2,f
            goto        ts000b
            goto        t000c
t000c       decfsz      heur3,f
            goto        ts000a
            goto        slow0           ;too slow - go to normal firng mode
setfast00   bcf         portb,led3      ;LED off                TIMING OKAY
            goto        setfast1

slow1       bcf         portd,lhslsw1   ;turn off phaseA, bottom switch                     stop motor as timing is wrong
            bcf         portd,lhshsw1   ;turn off phaseA, top switch
            bcf         portd,rhslsw2   ;turn off phaseB, bottom switch
            bcf         portd,rhshsw2   ;turn off phaseB, top switch
slow11      bsf         portb,led3      ;LED on
            call        strtdelay
            bcf         portb,led3      ;LED off
            call        strtdelay
            goto        chzer           ;start again

slow0       bcf         portd,lhslsw1   ;turn off phaseA, bottom switch                     stop motor as timing is wrong
            bcf         portd,lhshsw1   ;turn off phaseA, top switch
            bcf         portd,rhslsw2   ;turn off phaseB, bottom switch
            bcf         portd,rhshsw2   ;turn off phaseB, top switch
slow00      bsf         portb,led3      ;LED on
            call        sstrtdelay
            bcf         portb,led3      ;LED off
            bcf         portb,phase     ;normal phase
            goto        chzer           ;start again

sstrtdelay  movlw       0x0a            ;load 2dec into working register - this coding allows a time delay so flashing of LED is visible
            movwf       del111          ;move 2dec from working register into del111 file register
sstrdelay   movlw       0x11            ;load 17dec into working register
            movwf       del11           ;move 17dec from working register into del11 file register
sstdelay    movlw       0xe7            ;load 231dec into working register
            movwf       del1            ;move 231dec from working register into del1 file register
ssdlay      decfsz      del1,1          ;decrement del1 and keep new value, skip next line if zero
            goto        ssdlay          ;del1 not zero so decrement again
            decfsz      del11,1         ;del1 is zero, decrement del11 and keep new value, skip next line if zero
            goto        sstdelay        ;del11 not zero, reload del1 again so can decrement again and maintain delay timing properly
            decfsz      del111,1        ;del11 is zero, decrement del111 and keep new value, skip next line if zero
            goto        sstrdelay       ;del111 not zero, reload del1 and del11 so can both decrement again and maintain delay timing properly
            return                      ;time delay complete - go back to line below the call strtdelay – total
                                        ;delay=1micro*2*17*231*6executions=0.050112sec


end
```

APPENDIX D

D.1 LCD INFORMATION DISPLAY

A second microcontroller powers a LCD display to give information about the speed mode, A/D values for selected channels, update options, etc (*Figure D.1*). The microcontroller improves the display of the timing delay values. The main microprocessor shows each selected channel delay by 8 LED's representing the 8-bit binary number of the A/D conversion for that channel. The second PIC16F877 drove a 16x2 LCD alphanumeric display. The PIC converts the input from the selected channel into a decimal number in terms of hundreds, tens, and ones (in a similar manner to an abacus) and then converts each one into the correct 8-bit number for the LCD display.

The second PIC also uses information from the first PIC16F877 to give the following LCD display features : -

The time delay channel chosen
The value of that channel
The speed Mode that channel affects
The speed Mode the machine is in
The fact whether the value is being updated into the time delay or not

The working A/D circuit and accompanying LCD display circuit is shown below in *Figure D.1*. The basic flow diagram for the LCD display coding is shown in *Figure D.2* but this did not show the fact that the LCD program was very coding-intensive (Appendix D.2).

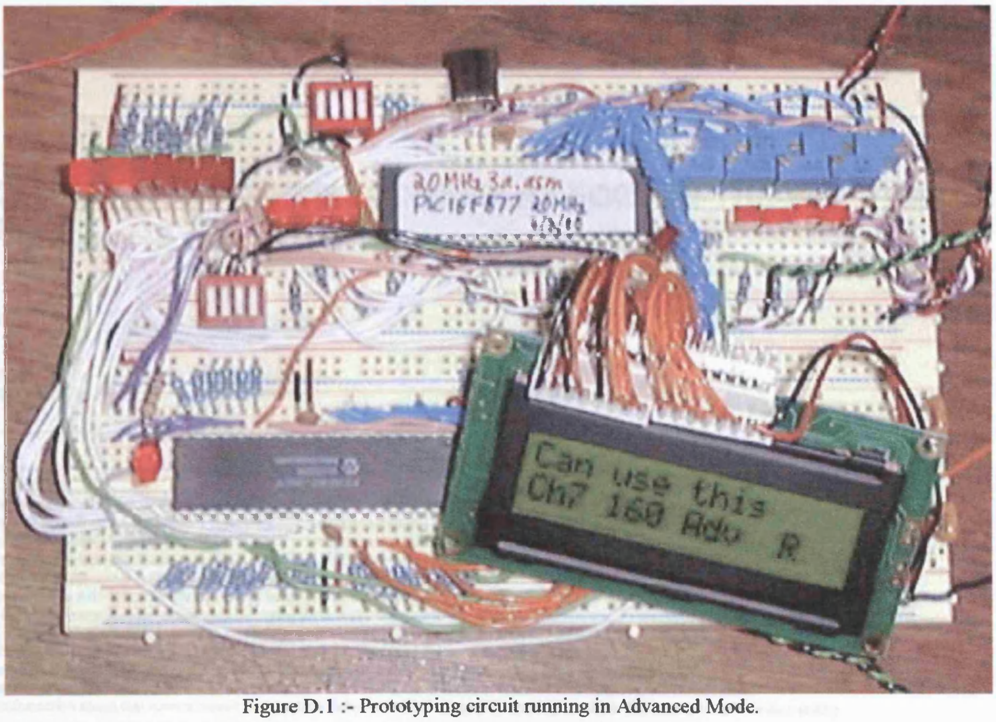


Figure D.1 :- Prototyping circuit running in Advanced Mode.

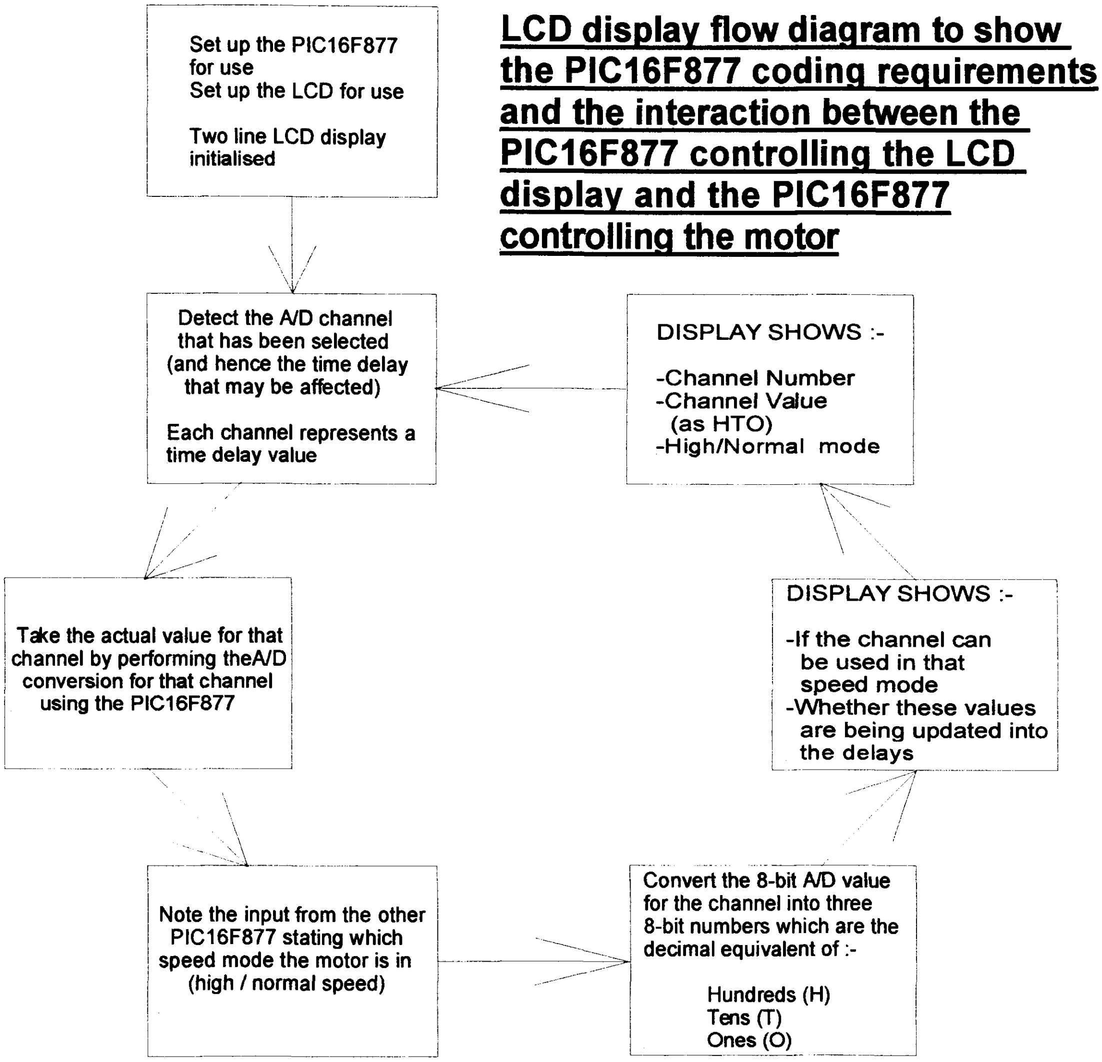


Figure D.2 :- LCD display flow chart.

D.2 PIC16F877 CODING FOR LCD DISPLAY FOR 'ON-THE-FLY' TIMINGS CONTROL

```
;PROGRAM                  tesplcd.asm
;WRITTEN BY               Ewan R.T. Goodier
;DATE                     21/09/2000

;PIC PROCESSOR            16F877, 40 pin device
;INSTRUCTION CLOCK        20 MHz, T=0.05us               0.2microsecond per instruction

;------------------------------------------
;
;------------------------------------------
;WHAT DOES THIS DO
;This program is an assist to the user of 20MHz3a.asm              Note that ADVANCED=HIGH SPEED
;It takes the output data such as the A/D output from the LEDs and converts it to
;the coding required to control a Hitachi LM016L 16x2 LCD alphanumeric display.
;It also notes the channel selected and compares the LED output for that channel from
;the first PIC16F877 with the actual A/D value it obtained from A/D converting that channel
;itself. This allows the true A/D value for the channel to be shown on the display as a decimal
;number rather than as a binary number which may be wrong for that channel due to the channel
;not being used in the speed mode of the motor (a result of maintaining high performance rate for the circuit).
;The display will state the channel selected, the value for that channel, the speed mode of the motor (normal or
;advanced), whether that channel has any effect on that speed mode, and whether the update values button is set or not.
;The coding does not affect performance of the motor as it is not linked to the control of the motor. It uses data from the control circuit
;to tell the user information about that control circuit and the time delay values being used. It is merely an aid for the user and could be time saving.
;------------------------------------------
;
```

```
;FIXED REGISTER EQUATES
;these must be defined exactly as below for the PIC16F877 microprocessor
;see data sheet for what they are responsible for
porta       equu    05H     ;defining place
portb       equ     06H     ;defining place
portc       equ     07H     ;defining place
portd       equ     08H     ;defining place
porte       equ     09H     ;defining place
tmr0        equ     01H     ;defining place
status      equ     03H     ;defining place
pcl         equ     02H     ;defining place
pclath      equ     0AH     ;defining place
intcon      equ     0BH     ;defining place
adresh      equ     1EH     ;defining place
adresl      equ     9EH     ;defining place
adcon0      equ     1FH     ;defining place
adcon1      equ     9FH     ;defining place
pie1        equ     8CH     ;defining place
pie2        equ     8DH     ;defining place
pir1        equ     0CH     ;defining place
pir2        equ     0DH     ;defining place
trisa       equ     85H     ;defining place
trisb       equ     86H     ;defining place
trisc       equ     87H     ;defining place
trisd       equ     88H     ;defining place
trise       equ     89H     ;defining place
option_reg  equ     81H     ;defining place
sspcon      equ     14H     ;defining place
rcsta       equ     18H     ;defining place
;
;VARIABLE REGISTER EQUATES
;these can have different names for each but you cannot have the same name twice and the position of the equate must fall within the
;designated zones for the microcontroller in question - see the data sheets for guidance
;are used for storing 8bit long values
acount      equ     20H     ;defining place - general purpose count register
del1        equ     21H     ;defining place  - time value
del2        equ     22H     ;defining place - time value
adres       equ     23H     ;defining place - A/D conversion value
temp        equ     24H     ;defining place - time value
tamp        equ     25H     ;defining place - time value
del         equ     26H     ;defining place - time value
del11       equ     28H     ;defining place - time value
del111      equ     29H     ;defining place - time value
val0        equ     2AH     ;defining place - ch0 A/D value
val1        equ     2BH     ;defining place - ch1 A/D value
val2        equ     2CH     ;defining place - ch2 A/D value
val3        equ     2DH     ;defining place - ch3 A/D value
val4        equ     2EH     ;defining place - ch4 A/D value
val5        equ     2FH     ;defining place - ch5 A/D value
val6        equ     30H     ;defining place - ch6 A/D value
val7        equ     31H     ;defining place - ch7 A/D value
redel1      equ     32H     ;defining place - time value
redel2      equ     33H     ;defining place - time value
lcdcha      equ     34H     ;defining place - A/D channel selection shown
count       equ     35H     ;defining place - A/D channel selection shown
zor         equ     36H     ;defining place - exclusive or value
val0a       equ     37H     ;defining place - ch0 A/D value
val1a       equ     38H     ;defining place - ch1 A/D value
val2a       equ     39H     ;defining place - ch2 A/D value
val3a       equ     3AH     ;defining place - ch3 A/D value
val4a       equ     3BH     ;defining place - ch4 A/D value
val5a       equ     3CH     ;defining place - ch5 A/D value
val6a       equ     3DH     ;defining place - ch6 A/D value
val7a       equ     3EH     ;defining place - ch7 A/D value
motval      equ     3FH     ;defining place - value for phase firing selection
time        equ     40H     ;defining place -  time value - advanced mode
time1       equ     41H     ;defining place - time value - advanced mode
time2       equ     42H     ;defining place - time value - advanced mode
time3       equ     43H     ;defining place - time value - advanced mode
extrachk    equ     44H     ;defining place - used for showing a timing problem
ledcnt      equ     45H     ;defining place - time value - advanced mode        unused
deccnt1     equ     46H     ;defining place - time value - advanced mode
deccnt2     equ     47H     ;defining place - time value - advanced mode
deccnt3     equ     48H     ;defining place - time value - advanced mode
deccnt4     equ     49H     ;defining place - time value - advanced mode
deccnt5     equ     4AH     ;defining place - time value - advanced mode
deccnt6     equ     4BH     ;defining place - time value - advanced mode
ADcnt       equ     4CH     ;defining place - time value
pha1        equ     4DH     ;defining place - ch0 A/D value
pha2        equ     4EH     ;defining place - ch1 A/D value
pha3        equ     4FH     ;defining place - ch2 A/D value
pha4        equ     50H     ;defining place - ch3 A/D value
pha5        equ     51H     ;defining place - ch4 A/D value
pha6        equ     52H     ;defining place - ch5 A/D value
pha7        equ     52H     ;defining place - ch6 A/D value
alpha       equ     54H     ;defining place - ch7 A/D value
nope        equ     55H     ;defining place - time value - advanced mode
heur1       equ     56H     ;defining place - time value - advanced mode
heur2       equ     57H     ;defining place - time value - advanced mode
heur3       equ     58H     ;defining place - time value - advanced mode
heur4       equ     59H     ;defining place - time value - advanced mode
;
;BIT DEFINITIONS
;registers sometimes have names to bit numbers but instead of referring to the bit number itself, they can be given names instead to
;aid in clarity of what the program is doing/affecting. Below is giving names to some of the bit numbers.
;There is no repetition as the nmemonic coding commands are such that the program automatically works out which bit to aim for and does not
;mind is the name or bit number is used
strtbut     equ     1       ;start button bit
fast        equ     4       ;direction bit
refresh     equ     5       ;refresh the system bit
phase       equ     4       ;adv/norm phase bit
```

```
opto            equ     6               ;optosensor bit
z               equ     2               ;zero bit flag
carry           equ     0               ;carry bit
toif            equ     2               ;TMR0 overflow interrupt flag bit
rbif            equ     0               ;RB<7:4> int. flag bit
rp0             equ     5               ;used in bank selector
rp1             equ     6               ;used in bank selector
peie            equ     6               ;peripheral interupt enable bit
gie             equ     7               ;global interupt enable bit
adif            equ     6               ;A/D converter interrupt flag bit
adie            equ     6               ;A/D converter interrupt enable bit
go              equ     2               ;A/D conversion status bit
rs              equ     6
enable          equ     7
f               equ     1               ;field register
w               equ     0               ;working register
;
;OUTPUTS
;Uses names to define bit numbers as explained in the section above
led3            equ     4               ;LED bit - for stopped or timing problem in advanced mode
lhslsw1         equ     0               ;Phase A bottom switch
lhshsw1         equ     1               ;Phase A top switch
rhslsw2         equ     2               ;Phase B bottom switch
rhshsw2         equ     3               ;Phase B top switch
;
;*******************************************************************
;                       CODING STARTS BELOW
;*******************************************************************

org     0x000                           ;tells assembler where to start loading the code 'goto  start' is the first bit of REAL program code

;*******************************************************************
;                       START SEQUENCE
;*******************************************************************
                bcf     pclath,3        ;page0
                bcf     pclath,4

goto            start                   ;goto start

init            clrf    porta           ;(from start) clear porta for defining inputs/outputs
                clrf    portb           ;clear portb for defining inputs/outputs
                clrf    portc           ;clear portc for defining inputs/outputs
                clrf    portd           ;clear portd for defining inputs/outputs
                clrf    porte           ;clear porte for defining inputs/outputs
                bcf     status,6        ;preparing to select bank 1
                bsf     status,5        ;bank1 selected
                bcf     option_reg,7    ;portb pull-ups are enabled by individual port latches
                bcf     sspcon,5        ;ra5,rc3,rc4,rc5 configured as I/O pins, serial port disabled
                bcf     rcsta,7         ;disable serial port
                bcf     trise,4         ;allows general purpose I/O mode for portd, disable parallel slave port mode portd
                bcf     intcon,7        ;disable all interupts
                clrf    pie1            ;disables peripheral
                return                  ;go back to line below (start)

init1           movlw   b'00000000'     ;(from main) set up portb       0out,1in        ouputs to LCD display
                movwf   trisb           ;set up portb
                movlw   B'00111111'     ;set up porta                                   A/D inputs, A/D select, master clear
                movwf   trisa           ;set up porta
                movlw   B'00111111'     ;set up portc                   LCD outputs and switch inputs, phase firing state bit input
                movwf   trisc           ;set up portc
                movlw   B'00000111'     ;set up porte                                   A2D inputs and setting up portd
                movwf   trise           ;set up porte
                movlw   B'11111111'     ;set up portd                                   inputs from LED display
                movwf   trisd           ;set up portd
                clrf    adcon1          ;all A/D inputs analogue, Vref+=supply=5V, Vref-=ground=0V, no reference voltages
                bcf     status,5        ;bank0 selected
                movlw   b'01000001'     ;Fosc/8, channel0 selected,A/Dconversion not in progress, A/D converter module operating
                movwf   adcon0          ;set it up as above
                return                  ;got back to line below (main)


start           call    delay
                call    delay
                call    init            ;goto init then return here for line below

main            call    delay
                call    delay
                call    init1           ;goto init1 then return here for line below

strtlcd2        call    delay
                call    delay
                call    delay
                bcf     portc,rs
                movlw   b'00111000'     ;8bit operation, 2line display, 5x7 dot character font
                movwf   portb
                bsf     portc,enable
                call    latch
                call    delay
                movlw   0xff
                movwf   time1
dectime2        decfsz  time1,f
                goto    dectime2
                bcf     portc,enable
                call    latch
                call    delay
delaylcd2       movlw   0xff
                movwf   time2
getn2           movlw   0xff
                movwf   time3
decn2           decfsz  time3,f
                goto    decn2
```

```
                decfsz  time2,f
                goto    getn2
                goto    lcdon

delay           movlw   0xc3            ;c8
                movwf   time2
getn1           movlw   0xc3            ;c3
                movwf   time3
decn1           decfsz  time3,f
                goto    decn1
                decfsz  time2,f
                goto    getn1
                return

latch           bsf     portc,enable
                movlw   0xaa            ;aa
                movwf   time1
dectime1        decfsz  time1,f
                goto    dectime1
                bcf     portc,enable
                return


lcdon           bcf     portc,rs
                movlw   b'00001110'     ;display&cursor on
                movwf   portb
                call    latch
                call    delay

                movlw   b'00000110'     ;set entry mode, When address incremented, move cursor to right and vice versa. display not shifted
                movwf   portb
                call    latch
                call    delay

                movlw   b'00000001'     ;clear display
                movwf   portb
                call    latch
                call    delay

lcdon1          bcf     portc,rs
                movlw   b'00000011'     ;reset and home
                movwf   portb
                call    latch
                call    delay
                bsf     portc,rs

dfine2          movlw   b'11111110'  ;
                movwf   deccnt4         ;decimal value in
                movlw   b'11111110'  ;
                movwf   deccnt5         ;decimal value in
                movlw   b'11111110'  ;
                movwf   deccnt6         ;decimal value in
                movlw   b'11111110'  ;
                movwf   deccnt3         ;decimal value in
                movlw   b'11111110'  ;
                movwf   deccnt2         ;decimal value in
                movlw   b'11111110'  ;
                movwf   deccnt1         ;decimal value in

dfine           bcf     portc,rs
                movlw   b'00000011'     ;reset and home
                movwf   portb
                call    latch
                call    delay
                bsf     portc,rs        ;set the data register
                goto    dfine1


dfine3          bcf     portc,rs
                movlw   b'00000011'     ;reset and home
                movwf   portb


dfine1          bsf     portc,rs        ;set the data register
                movlw   b'11111110'  ;
                movwf   portb
                call    latch
                call    delay
                movlw   b'11111110'  ;
                movwf   portb
                call    latch
                call    delay
                movlw   b'11111110'  ;
                movwf   portb
                call    latch
                call    delay
                movlw   b'11111110'  ;
                movwf   portb
                call    latch
                call    delay
                movlw   b'11111110'  ;
                movwf   portb
                call    latch
                call    delay
                movlw   b'11111110'  ;
                movwf   portb
                call    latch
                call    delay
                movlw   b'11111110'  ;
                movwf   portb
```

```
		call	latch
		call	delay
		movlw	b'11111110'	;
		movwf	portb
		call	latch
		call	delay
		movlw	b'11111110'	;
		movwf	portb
		call	latch
		call	delay
		movlw	b'11111110'	;
		movwf	portb
		call	latch
		call	delay
		movlw	b'11111110'	;
		movwf	portb
		call	latch
		call	delay
		movlw	b'11111110'	;
		movwf	portb
		call	latch
		call	delay
		movlw	b'11111110'	;
		movwf	portb
		call	latch
		call	delay
		movlw	b'11111110'	;
		movwf	portb
		call	latch
		call	delay
		bcf	portc,rs
		movlw	b'11000000'		;move to second line
		movwf	portb
		call	latch
		call	delay
		bsf	portc,rs
		movf	deccnt1,w	;
		movwf	portb
		call	latch
		call	delay
		movf	deccnt2,w	;
		movwf	portb
		call	latch
		call	delay
		movf	deccnt3,w	;
		movwf	portb
		call	latch
		call	delay
		movlw	b'11111110'	;
		movwf	portb
		call	latch
		call	delay
		movf	deccnt4,w	;
		movwf	portb
		call	latch
		call	delay
		movf	deccnt5,w	;
		movwf	portb
		call	latch
		call	delay
		movf	deccnt1,w	;
		movwf	portb
		call	latch
		call	delay
		movlw	b'11111110'	;
		movwf	portb
		call	latch
		call	delay
		movlw	b'11111110'		;
		movwf	portb
		call	latch
		call	delay
		movlw	b'11111110'		;
		movwf	portb
		call	latch
		call	delay
		movlw	b'11111110'		;
		movwf	portb
		call	latch
		call	delay
		movlw	b'11111110'		;
		movwf	portb
		call	latch
		call	delay
		movlw	b'11111110'		;
		movwf	portb
		call	latch
		call	delay
		movlw	b'11111110'	;
		movwf	portb
		call	latch
		call	delay
		movlw	b'11111110'		;
		movwf	portb
		call	latch
		call	delay

afine		bcf	portc,rs
		movlw	b'00000011'		;reset and home
		movwf	portb
		call	latch
```

```
call      delay
bsf       portc,rs          ;set the data register
movlw     b'01010111'       ;W
movwf     portb
call      latch
call      delay
movlw     b'01100101'       ;We
movwf     portb
call      latch
call      delay
movlw     b'01101100'       ;Wel
movwf     portb
call      latch
call      delay
movlw     b'01100011'       ;Welc
movwf     portb
call      latch
call      delay
movlw     b'01101111'       ;Welco
movwf     portb
call      latch
call      delay
movlw     b'01101101'       ;Welcom
movwf     portb
call      latch
call      delay
movlw     b'01100101'       ;Welcome
movwf     portb
call      latch
call      delay
movlw     b'11111110'       ;Welcome (space)
movwf     portb
call      latch
call      delay
movlw     b'01110100'       ;Welcome (space) t
movwf     portb
call      latch
call      delay
movlw     b'01101111'       ;Welcome (space) to
movwf     portb
call      latch
call      delay
movlw     b'11111110'       ;Welcome (space) to (space)
movwf     portb
call      latch
call      delay
movlw     b'01110100'       ;Welcome (space) to (space) t
movwf     portb
call      latch
call      delay
movlw     b'01101000'       ;Welcome (space) to (space) th
movwf     portb
call      latch
call      delay
movlw     b'01100101'       ;Welcome (space) to (space) the
movwf     portb
call      latch
call      delay
bcf       portc,rs
movlw     b'11000000'       ;move to second line
movwf     portb
call      latch
call      delay
bsf       portc,rs
movlw     b'01000001'       ;A
movwf     portb
call      latch
call      delay
movlw     b'00110010'       ;A2
movwf     portb
call      latch
call      delay
movlw     b'01000100'       ;A2D
movwf     portb
call      latch
call      delay
movlw     b'11111110'       ;A2D (space)
movwf     portb
call      latch
call      delay
movlw     b'01100011'       ;A2D (space) c
movwf     portb
call      latch
call      delay
movlw     b'01101111'       ;A2D (space) co
movwf     portb
call      latch
call      delay
movlw     b'01101110'       ;A2D (space) con
movwf     portb
call      latch
call      delay
movlw     b'01110100'       ;A2D (space) cont
movwf     portb
call      latch
call      delay
movlw     b'01110010'       ;A2D (space) contr
movwf     portb
call      latch
call      delay
movlw     b'01101111'       ;A2D (space) contro
```

```
            movwf   portb
            call    latch
            call    delay
            movlw   b'01101100'         ;A2D (space) control
            movwf   portb
            call    latch
            call    delay
            movlw   b'11111110'         ;A2D (space) control (space)
            movwf   portb
            call    latch
            call    delay
            movlw   b'01100011'         ;A2D (space) control (space) c
            movwf   portb
            call    latch
            call    delay
            movlw   b'01100011'         ;A2D (space) control (space) cc
            movwf   portb
            call    latch
            call    delay
            movlw   b'01110100'         ;A2D (space) control (space) cct
            movwf   portb
            call    latch
            call    delay
            btfss   portc,5
            goto    chazer              ;button on so stop motor
            goto    lcdon1              ;button still off

chazer      goto    chzer

conv        movlw   0x0f                ;(from the A/D request for the channel selected) set up time delay (15dec)
            movwf   tamp                ;put 15 into tamp
lupe        decfsz  tamp,f              ;decrement by 1, skip next line if zero
            goto    lupe                ;not zero so decrement again - delay required to allow internal capacitors to charge up to value of input voltage
            bsf     adcon0,2            ;set A/D conversion into progress
            movlw   0x3c                ;set up time delay (60dec)
            movwf   temp                ;put 60 into temp
dog         decfsz  temp,f              ;decrement by 1, skip next line if zero
            goto    dog                 ;not zero so decrement again - allows time for A/D conversions to occur
conv1       btfsc   adcon0,2            ;has A/D conversion ended?
            goto    conv1               ;no - recheck until it has
            return                      ;A/D ended - value stored in adresl - got back to relevant point in coding


chzer       bcf     portc,rs
            movlw   b'00000011'         ;reset and home
            movwf   portb
            call    latch
            call    delay
            bcf     adcon0,3            ;select channel zero for A/D conversion          (after init1 has been called)
            bcf     adcon0,4            ;select channel zero for A/D conversion
            bcf     adcon0,5            ;select channel zero for A/D conversion
            call    conv                ;goto conv for A/D conversion of channel selected then return to line below
            movf    adresl,w            ;A/D value is in adresl - move it to working register
            movwf   val0                ;move the value in the working register to val0 - A/D for that channel now in the file register for that channel
            goto    chone               ;A/D the next channel
chone       bsf     adcon0,3            ;select channel one for A/D conversion
            bcf     adcon0,4            ;select channel one for A/D conversion
            bcf     adcon0,5            ;select channel one for A/D conversion
            call    conv                ;goto conv for A/D conversion of channel selected then return to line below
            movf    adresl,w            ;A/D value is in adresl - move it to working register
            movwf   val1                ;move the value in the working register to val1 - A/D for that channel now in the file register for that channel
            goto    chtwo               ;A/D the next channel
chtwo       bcf     adcon0,3            ;select channel two for A/D conversion
            bsf     adcon0,4            ;select channel two for A/D conversion
            bcf     adcon0,5            ;select channel two for A/D conversion
            call    conv                ;goto conv for A/D conversion of channel selected then return to line below
            movf    adresl,w            ;A/D value is in adresl - move it to working register
            movwf   val2                ;move the value in the working register to val2 - A/D for that channel now in the file register for that channel
            goto    chthr               ;A/D the next channel
chthr       bsf     adcon0,3            ;select channel three for A/D conversion
            bsf     adcon0,4            ;select channel three for A/D conversion
            bcf     adcon0,5            ;select channel three for A/D conversion
            call    conv                ;goto conv for A/D conversion of channel selected then return to line below
            movf    adresl,w            ;A/D value is in adresl - move it to working register
            movwf   val3                ;move the value in the working register to val3 - A/D for that channel now in the file register for that channel
            goto    chfou               ;A/D the next channel
chfou       bcf     adcon0,3            ;select channel four for A/D conversion
            bcf     adcon0,4            ;select channel four for A/D conversion
            bsf     adcon0,5            ;select channel four for A/D conversion
            call    conv                ;goto conv for A/D conversion of channel selected then return to line below
            movf    adresl,w            ;A/D value is in adresl - move it to working register
            movwf   val4                ;move the value in the working register to val4 - A/D for that channel now in the file register for that channel
            goto    chfiv               ;A/D the next channel
chfiv       bsf     adcon0,3            ;select channel five for A/D conversion
            bcf     adcon0,4            ;select channel five for A/D conversion
            bsf     adcon0,5            ;select channel five for A/D conversion
            call    conv                ;goto conv for A/D conversion of channel selected then return to line below
            movf    adresl,w            ;A/D value is in adresl - move it to working register
            movwf   val5                ;move the value in the working register to val5 - A/D for that channel now in the file register for that channel
            goto    chsix               ;A/D the next channel
chsix       bcf     adcon0,3            ;select channel six for A/D conversion
            bsf     adcon0,4            ;select channel six for A/D conversion
            bsf     adcon0,5            ;select channel six for A/D conversion
            call    conv                ;goto conv for A/D conversion of channel selected then return to line below
            movf    adresl,w            ;A/D value is in adresl - move it to working register
            movwf   val6                ;move the value in the working register to val6 - A/D for that channel now in the file register for that channel
            goto    chsev               ;A/D the next channel
chsev       bsf     adcon0,3            ;select channel seven for A/D conversion
            bsf     adcon0,4            ;select channel seven for A/D conversion
            bsf     adcon0,5            ;select channel seven for A/D conversion
            call    conv                ;goto conv for A/D conversion of channel selected then return to line below
```

```
            movf    adresl,w        ;A/D value is in adresl - move it to working register
            movwf   val7            ;move the value in the working register to val7 - A/D for that channel now in the file register for that channel
            goto    chandetect      ;display the A/D output for whichever channel is selected using the 3 pink wires - go to display

chandetect  btfsc   portc,1         ;3 pink wires are used to select which A/D channel is to be shown, which is detemined below
            goto    disa1           ;1/3/5/7    channels applicable
            btfsc   portc,2         ;0/2/4/6    channels applicable
            goto    disb1           ;2/6        channels applicable
            btfsc   portc,3         ;0/4        channels applicable
            goto    four1           ;4out       channel 4 chosen
            goto    zero1           ;0out       channel 0 chosen
disa1       btfsc   portc,2         ;1/3/5/7    channels applicable
            goto    disc1           ;3/7        channels applicable
            btfsc   portc,3         ;1/5        channels applicable
            goto    five1           ;5out       channel 5 chosen
            goto    one1            ;1out       channel 1 chosen
disb1       btfsc   portc,3         ;2/6        channels applicable
            goto    six1            ;6out       channel 6 chosen
            goto    two1            ;2out       channel 2 chosen
disc1       btfsc   portc,3         ;3/7        channels applicable
            goto    seven1          ;7out       channel 7 chosen
            goto    three1          ;3out       channel 3 chosen
zero1       movlw   b'00110000'
            movwf   lcdcha
            movf    val0,w
            movwf   ADcnt
            goto    numer
one1        movlw   b'00110001'
            movwf   lcdcha
            movf    val1,w
            movwf   ADcnt
            goto    numer
two1        movlw   b'00110010'
            movwf   lcdcha
            movf    val2,w
            movwf   ADcnt
            goto    numer
three1      movlw   b'00110011'
            movwf   lcdcha
            movf    val3,w
            movwf   ADcnt
            goto    numer
four1       movlw   b'00110100'
            movwf   lcdcha
            movf    val4,w
            movwf   ADcnt
            goto    numer
five1       movlw   b'00110101'
            movwf   lcdcha
            movf    val5,w
            movwf   ADcnt
            goto    numer
six1        movlw   b'00110110'
            movwf   lcdcha
            movf    val6,w
            movwf   ADcnt
            goto    numer
seven1      movlw   b'00110111'
            movwf   lcdcha
            movf    val7,w
            movwf   ADcnt
            goto    numer           ;LCD chan values done, val? put into ADcnt

showit1     goto    letsgo


numer       movf    ADcnt,w
            movwf   alpha
leddetect   btfsc   portd,0         ;find the LED value - stated A/D value - not necessarily the correct one
            goto    bit01
            goto    bit00
bit01       bsf     ledcnt,0
            goto    det1
bit00       bcf     ledcnt,0
            goto    det1
det1        btfsc   portd,1
            goto    bit11
            goto    bit10
bit11       bsf     ledcnt,1
            goto    det2
bit10       bcf     ledcnt,1
            goto    det2
det2        btfsc   portd,2
            goto    bit21
            goto    bit20
bit21       bsf     ledcnt,2
            goto    det3
bit20       bcf     ledcnt,2
            goto    det3
det3        btfsc   portd,3
            goto    bit31
            goto    bit30
bit31       bsf     ledcnt,3
            goto    det4
bit30       bcf     ledcnt,3
            goto    det4
det4        btfsc   portd,4
            goto    bit41
            goto    bit40
```

```
bit41     bsf      ledcnt,4
          goto     det5
bit40     bcf      ledcnt,4
          goto     det5
det5      btfsc    portd,5
          goto     bit51
          goto     bit50
bit51     bsf      ledcnt,5
          goto     det6
bit50     bcf      ledcnt,5
          goto     det6
det6      btfsc    portd,6
          goto     bit61
          goto     bit60
bit61     bsf      ledcnt,6
          goto     det7
bit60     bcf      ledcnt,6
          goto     det7
det7      btfsc    portd,7
          goto     bit71
          goto     bit70
bit71     bsf      ledcnt,7
          goto     detend
bit70     bcf      ledcnt,7
          goto     detend

detend    movf     ledcnt,w
          xorwf    ADcnt,w
          movwf    extrachk
          btfsc    extrachk,0
          goto     differ
          btfsc    extrachk,1
          goto     differ
          btfsc    extrachk,2
          goto     differ
          btfsc    extrachk,3
          goto     differ
          btfsc    extrachk,4
          goto     differ
          btfsc    extrachk,5
          goto     differ
          btfsc    extrachk,6
          goto     differ
          btfsc    extrachk,7
          goto     differ
          goto     same

differ    movlw    b'00000001'
          movwf    nope
          goto     contin
same      movlw    b'00000000'
          movwf    nope
          goto     contin

contin    bsf      pclath,3
          bcf      pclath,4
          goto     numem


letsgo    btfss    portc,phase
          goto     itsnorm
          goto     itsadv
itsnorm   movlw    b'11111110'     ;(space)              NORMAL MODE WITHOUT PULSING
          movwf    pha1            ;decimal value in
          movlw    b'01001110'     ;N
          movwf    pha2            ;decimal value in
          movlw    b'01101111'     ;o
          movwf    pha3            ;decimal value in
          movlw    b'01110010'     ;r
          movwf    pha4            ;decimal value in
          movlw    b'01101101'     ;m
          movwf    pha5            ;decimal value in
          goto     charefre
itsadv    movlw    b'11111110'     ;(space)              HIGH SPEED MODE     advanced timing
          movwf    pha1            ;decimal value in
          movlw    b'01000001'     ;A
          movwf    pha2            ;decimal value in
          movlw    b'01100100'     ;d
          movwf    pha3            ;decimal value in
          movlw    b'01110110'     ;v
          movwf    pha4            ;decimal value in
          movlw    b'11111110'     ;space
          movwf    pha5            ;decimal value in
          goto     charefre
charefre  btfss    portc,0         ;test if refreshed is on - clr if on
          goto     isrefre
          goto     notrefre
isrefre   movlw    b'11111110'     ;(space)
          movwf    pha6            ;decimal value in
          movlw    b'01010010'     ;R                    refresh timing number used
          movwf    pha7            ;decimal value in
          goto     chacha
notrefre  movlw    b'11111110'     ;(space)
          movwf    pha6            ;decimal value in
          movlw    b'01010101'     ;U                    don't update timing number used
          movwf    pha7            ;decimal value in
          goto     chacha
chacha    movlw    b'01000011'     ;C
          movwf    deccnt1         ;
          movlw    b'01101000'     ;h
          movwf    deccnt2         ;
```

```
            movf    lcdcha,w        ;whatever the channel number is
            movwf   deccnt3         ;
            btfss   portc,phase     ;retest phase state
            goto    itsnorm1        ;normal mode
            goto    itsadv1         ;advanced mode

itsnorm1    movf    lcdcha,w        ;chan no                norm
            sublw   b'00110011'     ;3-chan no (0-3 is 0 or +ve=1, rest -ve=0)
            btfsc   status,0        ;
            goto    fine            ; 0-3 norm = wrong
            goto    noway           ; 4-7 adv = right

itsadv1     movf    lcdcha,w        ;chan no                adv
            sublw   b'00110011'     ;3-chan no(0-3 is 0 or +ve, rest -ve)
            btfsc   status,0        ;
            goto    noway           ;0-3 norm = wrong
            goto    fine            ;4-7 adv = right


fine        bcf     portc,rs
            movlw   b'00000011'     ;reset and home
            movwf   portb
            call    latch
            call    delay
            bsf     portc,rs        ;set the data register
            movlw   b'01000011'     ;C
            movwf   portb
            call    latch
            call    delay
            movlw   b'01100001'     ;Ca
            movwf   portb
            call    latch
            call    delay
            movlw   b'01101110'     ;Can
            movwf   portb
            call    latch
            call    delay
            movlw   b'11111110'     ;Can(space)
            movwf   portb
            call    latch
            call    delay
            movlw   b'01110101'     ;Can u
            movwf   portb
            call    latch
            call    delay
            movlw   b'01110011'     ;Can us
            movwf   portb
            call    latch
            call    delay
            movlw   b'01100101'     ;Can use
            movwf   portb
            call    latch
            call    delay
            movlw   b'11111110'     ;Can use(space)
            movwf   portb
            call    latch
            call    delay
            movlw   b'01110100'     ;Can use t
            movwf   portb
            call    latch
            call    delay
            movlw   b'01101000'     ;Can use th
            movwf   portb
            call    latch
            call    delay
            movlw   b'01101001'     ;Can use thi
            movwf   portb
            call    latch
            call    delay
            movlw   b'01110011'     ;Can use this
            movwf   portb
            call    latch
            call    delay
            movlw   b'11111110'     ;Can use this(space)
            movwf   portb
            call    latch
            call    delay
            movlw   b'11111110'     ;Can use this (space) (space)
            movwf   portb
            call    latch
            call    delay
            movlw   b'11111110'     ;Can use this (space)(space)(space)
            movwf   portb
            call    latch
            call    delay
            bcf     portc,rs
            movlw   b'11000000'     ;move to second line
            movwf   portb
            call    latch
            call    delay
            bsf     portc,rs
            movf    deccnt1,w       ;C
            movwf   portb
            call    latch
            call    delay
            movf    deccnt2,w       ;h
            movwf   portb
            call    latch
            call    delay
            movf    deccnt3,w       ;actual channel number
            movwf   portb
```

```
                call    latch
                call    delay
                movlw   b'11111110'     ;space
                movwf   portb
                call    latch
                call    delay
                movf    deccnt4,w       ;1st number value
                movwf   portb
                call    latch
                call    delay
                movf    deccnt5,w       ;2nd number value
                movwf   portb
                call    latch
                call    delay
                movf    deccnt6,w       ;3rd number value
                movwf   portb
                call    latch
                call    delay
                movf    pha1,w          ;space
                movwf   portb
                call    latch
                call    delay
                movf    pha2,w          ;N/A
                movwf   portb
                call    latch
                call    delay
                movf    pha3,w          ;o/d
                movwf   portb
                call    latch
                call    delay
                movf    pha4,w          ;r/v
                movwf   portb
                call    latch
                call    delay
                movf    pha5,w          ;m/space
                movwf   portb
                call    latch
                call    delay
                movlw   b'11111110'     ;space
                movwf   portb
                call    latch
                call    delay
                movf    pha7,w          ;R/U
                movwf   portb
                call    latch
                call    delay
                movlw   b'11111110'     ;space
                movwf   portb
                call    latch
                call    delay
                btfss   portc,5
                goto    chzer           ;button on so stop motor
                goto    lcdon1          ;button still off


noway           bcf     portc,rs
                movlw   b'00000011'     ;reset and home
                movwf   portb
                call    latch
                call    delay
                bsf     portc,rs        ;set the data register
                movlw   b'01000011'     ;C
                movwf   portb
                call    latch
                call    delay
                movlw   b'01100001'     ;Ca
                movwf   portb
                call    latch
                call    delay
                movlw   b'01101110'     ;Can
                movwf   portb
                call    latch
                call    delay
                movlw   b'00100111'     ;Can'
                movwf   portb
                call    latch
                call    delay
                movlw   b'01110100'     ;Can't
                movwf   portb
                call    latch
                call    delay
                movlw   b'11111110'     ;Can't(space)
                movwf   portb
                call    latch
                call    delay
                movlw   b'01110101'     ;Can't u
                movwf   portb
                call    latch
                call    delay
                movlw   b'01110011'     ;Can't us
                movwf   portb
                call    latch
                call    delay
                movlw   b'01100101'     ;Can't use
                movwf   portb
                call    latch
                call    delay
                movlw   b'11111110'     ;Can't use(space)
                movwf   portb
                call    latch
```

```
		call	delay
		movlw	b'01110100'	;Can't use t
		movwf	portb
		call	latch
		call	delay
		movlw	b'01101000'	;Can't use th
		movwf	portb
		call	latch
		call	delay
		movlw	b'01101001'	;Can't use thi
		movwf	portb
		call	latch
		call	delay
		movlw	b'01110011'	;Can't use this
		movwf	portb
		call	latch
		call	delay
		movlw	b'11111110'	;(space)
		movwf	portb
		call	latch
		call	delay
		bcf	portc,rs
		movlw	b'11000000'	;move to second line
		movwf	portb
		call	latch
		call	delay
		bsf	portc,rs
		movf	deccnt1,w	;C
		movwf	portb
		call	latch
		call	delay
		movf	deccnt2,w	;h
		movwf	portb
		call	latch
		call	delay
		movf	deccnt3,w	;actual channel number
		movwf	portb
		call	latch
		call	delay
		movlw	b'11111110'	;space
		movwf	portb
		call	latch
		call	delay
		movf	deccnt4,w	;1st number value
		movwf	portb
		call	latch
		call	delay
		movf	deccnt5,w	;2nd number value
		movwf	portb
		call	latch
		call	delay
		movf	deccnt6,w	;3rd number value
		movwf	portb
		call	latch
		call	delay
		movf	pha1,w	;space
		movwf	portb
		call	latch
		call	delay
		movf	pha2,w	;N/A
		movwf	portb
		call	latch
		call	delay
		movf	pha3,w	;o/d
		movwf	portb
		call	latch
		call	delay
		movf	pha4,w	;r/v
		movwf	portb
		call	latch
		call	delay
		movf	pha5,w	;m/space
		movwf	portb
		call	latch
		call	delay
		movlw	b'11111110'	;space
		movwf	portb
		call	latch
		call	delay
		movf	pha7,w	;R/U
		movwf	portb
		call	latch
		call	delay
		movlw	b'11111110'	;space
		movwf	portb
		call	latch
		call	delay
		btfss	portc,5
		goto	chzer	;button on so stop motor
		goto	lcdon1	;button still off


org	0x800

numern		btfss	alpha,0	;xxxxxxxx
		goto	abit0is0	;
		goto	abit0is1	;
abit0is0	btfss	alpha,1	;xxxxxxx0
		goto	abit1is0	;
		goto	abit1is1	;
```

```
abit0is1    btfss   alpha,1       ;xxxxxxx1
            goto    aabit1is0     ;
            goto    aabit1is1     ;
abit1is0    btfss   alpha,2       ;xxxxxx00
            goto    abit2is0      ;
            goto    abit2is1      ;
abit1is1    btfss   alpha,2       ;xxxxxx10
            goto    aabit2is0
            goto    aabit2is1
aabit1is0   btfss   alpha,2       ;xxxxxx01
            goto    aaabit2is0    ;
            goto    aaabit2is1    ;
aabit1is1   btfss   alpha,2       ;xxxxxx11
            goto    aaaabit2is0   ;
            goto    aaaabit2is1   ;
abit2is0    btfss   alpha,3       ;xxxxx000
            goto    abit3is0      ;
            goto    abit3is1      ;
abit2is1    btfss   alpha,3       ;xxxxx100
            goto    aabit3is0     ;
            goto    aabit3is1     ;
aabit2is0   btfss   alpha,3       ;xxxxx010
            goto    bbit3is0      ;
            goto    bbit3is1      ;
aabit2is1   btfss   alpha,3       ;xxxxx110
            goto    bbbit3is0     ;
            goto    bbbit3is1     ;
aaabit2is0  btfss   alpha,3       ;xxxxx001
            goto    cbit3is0      ;
            goto    cbit3is1      ;
aaabit2is1  btfss   alpha,3       ;xxxxx101
            goto    ccbit3is0     ;
            goto    ccbit3is1     ;
aaaabit2is0 btfss   alpha,3       ;xxxxx011
            goto    dbit3is0      ;
            goto    dbit3is1      ;
aaaabit2is1 btfss   alpha,3       ;xxxxx111
            goto    ddbit3is0     ;
            goto    ddbit3is1     ;
abit3is0    btfss   alpha,4       ;xxxx0000
            goto    abit4is0
            goto    abit4is1
abit3is1    btfss   alpha,4       ;xxxx1000
            goto    aabit4is0
            goto    aabit4is1
aabit3is0   btfss   alpha,4       ;xxxx0100
            goto    bbit4is0
            goto    bbit4is1
aabit3is1   btfss   alpha,4       ;xxxx1100
            goto    bbbit4is0
            goto    bbbit4is1
bbit3is0    btfss   alpha,4       ;xxxx0010
            goto    cbit4is0
            goto    cbit4is1
bbit3is1    btfss   alpha,4       ;xxxx1010
            goto    ccbit4is0
            goto    ccbit4is1
bbbit3is0   btfss   alpha,4       ;xxxx0110
            goto    dbit4is0
            goto    dbit4is1
bbbit3is1   btfss   alpha,4       ;xxxx1110
            goto    ddbit4is0
            goto    ddbit4is1
cbit3is0    btfss   alpha,4       ;xxxx0001
            goto    ebit4is0
            goto    ebit4is1
cbit3is1    btfss   alpha,4       ;xxxx1001
            goto    eebit4is0
            goto    eebit4is1
ccbit3is0   btfss   alpha,4       ;xxxx0101
            goto    fbit4is0
            goto    fbit4is1
ccbit3is1   btfss   alpha,4       ;xxxx1101
            goto    ffbit4is0
            goto    ffbit4is1
dbit3is0    btfss   alpha,4       ;xxxx0011
            goto    gbit4is0
            goto    gbit4is1
dbit3is1    btfss   alpha,4       ;xxxx1011
            goto    ggbit4is0
            goto    ggbit4is1
ddbit3is0   btfss   alpha,4       ;xxxx0111
            goto    hbit4is0
            goto    hbit4is1
ddbit3is1   btfss   alpha,4       ;xxxx1111
            goto    hhbit4is0
            goto    hhbit4is1


abit4is0    btfss   alpha,5       ;xxx00000
            goto    abit5is0
            goto    abit5is1
abit4is1    btfss   alpha,5       ;xxx10000
            goto    aabit5is0
            goto    aabit5is1
aabit4is0   btfss   alpha,5       ;xxx01000
            goto    bbit5is0
            goto    bbit5is1
aabit4is1   btfss   alpha,5       ;xxx11000
            goto    bbbit5is0
            goto    bbbit5is1
```

```
bbit4is0        btfss   alpha,5         ;xxx00100
                goto    cbit5is0
                goto    cbit5is1
bbit4is1        btfss   alpha,5         ;xxx10100
                goto    ccbit5is0
                goto    ccbit5is1
bbbit4is0       btfss   alpha,5         ;xxx01100
                goto    dbit5is0
                goto    dbit5is1
bbbit4is1       btfss   alpha,5         ;xxx11100
                goto    ddbit5is0
                goto    ddbit5is1
cbit4is0        btfss   alpha,5         ;xxx00010
                goto    ebit5is0
                goto    ebit5is1
cbit4is1        btfss   alpha,5         ;xxx10010
                goto    eebit5is0
                goto    eebit5is1
ccbit4is0       btfss   alpha,5         ;xxx01010
                goto    fbit5is0
                goto    fbit5is1
ccbit4is1       btfss   alpha,5         ;xxx11010
                goto    ffbit5is0
                goto    ffbit5is1
dbit4is0        btfss   alpha,5         ;xxx00110
                goto    gbit5is0
                goto    gbit5is1
dbit4is1        btfss   alpha,5         ;xxx10110
                goto    ggbit5is0
                goto    ggbit5is1
ddbit4is0       btfss   alpha,5         ;xxx01110
                goto    hbit5is0
                goto    hbit5is1
ddbit4is1       btfss   alpha,5         ;xxx11110
                goto    hhbit5is0
                goto    hhbit5is1
ebit4is0        btfss   alpha,5         ;xxx00001
                goto    ibit5is0
                goto    ibit5is1
ebit4is1        btfss   alpha,5         ;xxx10001
                goto    iibit5is0
                goto    iibit5is1
eebit4is0       btfss   alpha,5         ;xxx01001
                goto    jbit5is0
                goto    jbit5is1
eebit4is1       btfss   alpha,5         ;xxx11001
                goto    jjbit5is0
                goto    jjbit5is1
fbit4is0        btfss   alpha,5         ;xxx00101
                goto    kbit5is0
                goto    kbit5is1
fbit4is1        btfss   alpha,5         ;xxx10101
                goto    kkbit5is0
                goto    kkbit5is1
ffbit4is0       btfss   alpha,5         ;xxx01101
                goto    lbit5is0
                goto    lbit5is1
ffbit4is1       btfss   alpha,5         ;xxx11101
                goto    llbit5is0
                goto    llbit5is1
gbit4is0        btfss   alpha,5         ;xxx00011
                goto    mbit5is0
                goto    mbit5is1
gbit4is1        btfss   alpha,5         ;xxx10011
                goto    mmbit5is0
                goto    mmbit5is1
ggbit4is0       btfss   alpha,5         ;xxx01011
                goto    nbit5is0
                goto    nbit5is1
ggbit4is1       btfss   alpha,5         ;xxx11011
                goto    nnbit5is0
                goto    nnbit5is1
hbit4is0        btfss   alpha,5         ;xxx00111
                goto    obit5is0
                goto    obit5is1
hbit4is1        btfss   alpha,5         ;xxx10111
                goto    oobit5is0
                goto    oobit5is1
hhbit4is0       btfss   alpha,5         ;xxx01111
                goto    pbit5is0
                goto    pbit5is1
hhbit4is1       btfss   alpha,5         ;xxx11111
                goto    ppbit5is0
                goto    ppbit5is1

abit5is0        bcf     pclath,3
                bsf     pclath,4
                btfss   alpha,6         ;xx000000
                goto    abit6is0
                goto    abit6is1
abit5is1        bcf     pclath,3
                bsf     pclath,4
                btfss   alpha,6         ;xx100000
                goto    aabit6is0
                goto    aabit6is1
aabit5is0       bcf     pclath,3
                bsf     pclath,4
                btfss   alpha,6         ;xx010000
                goto    bbit6is0
                goto    bbit6is1
```

```
aabit5is1   bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx110000
            goto    bbbit6is0
            goto    bbbit6is1
bbit5is0    bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx001000
            goto    cbit6is0
            goto    cbit6is1
bbit5is1    bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx101000
            goto    ccbit6is0
            goto    ccbit6is1
bbbit5is0   bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx011000
            goto    dbit6is0
            goto    dbit6is1
bbbit5is1   bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx111000
            goto    ddbit6is0
            goto    ddbit6is1
cbit5is0    bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx000100
            goto    ebit6is0
            goto    ebit6is1
cbit5is1    bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx100100
            goto    eebit6is0
            goto    eebit6is1
ccbit5is0   bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx010100
            goto    fbit6is0
            goto    fbit6is1
ccbit5is1   bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx110100
            goto    ffbit6is0
            goto    ffbit6is1
dbit5is0    bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx001100
            goto    gbit6is0
            goto    gbit6is1
dbit5is1    bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx101100
            goto    ggbit6is0
            goto    ggbit6is1
ddbit5is0   bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx011100
            goto    hbit6is0
            goto    hbit6is1
ddbit5is1   bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx111100
            goto    hhbit6is0
            goto    hhbit6is1
ebit5is0    bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx000010
            goto    ibit6is0
            goto    ibit6is1
ebit5is1    bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx100010
            goto    iibit6is0
            goto    iibit6is1
eebit5is0   bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx010010
            goto    jbit6is0
            goto    jbit6is1
eebit5is1   bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx110010
            goto    jjbit6is0
            goto    jjbit6is1
fbit5is0    bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx001010
            goto    kbit6is0
            goto    kbit6is1
fbit5is1    bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx101010
            goto    kkbit6is0
            goto    kkbit6is1
ffbit5is0   bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx011010
            goto    lbit6is0
            goto    lbit6is1
ffbit5is1   bcf     pclath,3
```

```
            bsf     pclath,4
            btfss   alpha,6         ;xx111010
            goto    llbit6is0
            goto    llbit6is1
gbit5is0    bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx000110
            goto    mbit6is0
            goto    mbit6is1
gbit5is1    bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx100110
            goto    mmbit6is0
            goto    mmbit6is1
ggbit5is0   bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx010110
            goto    nbit6is0
            goto    nbit6is1
ggbit5is1   bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx110110
            goto    nnbit6is0
            goto    nnbit6is1
hbit5is0    bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx001110
            goto    obit6is0
            goto    obit6is1
hbit5is1    bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx101110
            goto    oobit6is0
            goto    oobit6is1
hhbit5is0   bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx011110
            goto    pbit6is0
            goto    pbit6is1
hhbit5is1   bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx111110
            goto    ppbit6is0
            goto    ppbit6is1
ibit5is0    bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx000001
            goto    qbit6is0
            goto    qbit6is1
ibit5is1    bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx100001
            goto    qqbit6is0
            goto    qqbit6is1
iibit5is0   bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx010001
            goto    rbit6is0
            goto    rbit6is1
iibit5is1   bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx110001
            goto    rrbit6is0
            goto    rrbit6is1
jbit5is0    bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx001001
            goto    sbit6is0
            goto    sbit6is1
jbit5is1    bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx101001
            goto    ssbit6is0
            goto    ssbit6is1
jjbit5is0   bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx011001
            goto    tbit6is0
            goto    tbit6is1
jjbit5is1   bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx111001
            goto    ttbit6is0
            goto    ttbit6is1
kbit5is0    bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx000101
            goto    ubit6is0
            goto    ubit6is1
kbit5is1    bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx100101
            goto    uubit6is0
            goto    uubit6is1
kkbit5is0   bcf     pclath,3
            bsf     pclath,4
            btfss   alpha,6         ;xx010101
            goto    vbit6is0
            goto    vbit6is1
kkbit5is1   bcf     pclath,3
            bsf     pclath,4
```

```
        btfss   alpha,6         ;xx110101
        goto    vvbit6is0
        goto    vvbit6is1
lbit5is0 bcf    pclath,3
        bsf     pclath,4
        btfss   alpha,6         ;xx001101
        goto    wbit6is0
        goto    wbit6is1
lbit5is1 bcf    pclath,3
        bsf     pclath,4
        btfss   alpha,6         ;xx101101
        goto    wwbit6is0
        goto    wwbit6is1
llbit5is0 bcf   pclath,3
        bsf     pclath,4
        btfss   alpha,6         ;xx011101
        goto    xbit6is0
        goto    xbit6is1
llbit5is1 bcf   pclath,3
        bsf     pclath,4
        btfss   alpha,6         ;xx111101
        goto    xxbit6is0
        goto    xxbit6is1
mbit5is0 bcf    pclath,3
        bsf     pclath,4
        btfss   alpha,6         ;xx000011
        goto    ybit6is0
        goto    ybit6is1
mbit5is1 bcf    pclath,3
        bsf     pclath,4
        btfss   alpha,6         ;xx100011
        goto    yybit6is0
        goto    yybit6is1
mmbit5is0 bcf   pclath,3
        bsf     pclath,4
        btfss   alpha,6         ;xx010011
        goto    zbit6is0
        goto    zbit6is1
mmbit5is1 bcf   pclath,3
        bsf     pclath,4
        btfss   alpha,6         ;xx110011
        goto    zzbit6is0
        goto    zzbit6is1
nbit5is0 bcf    pclath,3
        bsf     pclath,4
        btfss   alpha,6         ;xx001011
        goto    daabit6is0
        goto    daabit6is1
nbit5is1 bcf    pclath,3
        bsf     pclath,4
        btfss   alpha,6         ;xx101011
        goto    daaabit6is0
        goto    daaabit6is1
nnbit5is0 bcf   pclath,3
        bsf     pclath,4
        btfss   alpha,6         ;xx011011
        goto    dabbit6is0
        goto    dabbit6is1
nnbit5is1 bcf   pclath,3
        bsf     pclath,4
        btfss   alpha,6         ;xx111011
        goto    dabbbit6is0
        goto    dabbbit6is1
obit5is0 bcf    pclath,3
        bsf     pclath,4
        btfss   alpha,6         ;xx000111
        goto    dacbit6is0
        goto    dacbit6is1
obit5is1 bcf    pclath,3
        bsf     pclath,4
        btfss   alpha,6         ;xx100111
        goto    daccbit6is0
        goto    daccbit6is1
oobit5is0 bcf   pclath,3
        bsf     pclath,4
        btfss   alpha,6         ;xx010111
        goto    dadbit6is0
        goto    dadbit6is1
oobit5is1 bcf   pclath,3
        bsf     pclath,4
        btfss   alpha,6         ;xx110111
        goto    daddbit6is0
        goto    daddbit6is1
pbit5is0 bcf    pclath,3
        bsf     pclath,4
        btfss   alpha,6         ;xx001111
        goto    daebit6is0
        goto    daebit6is1
pbit5is1 bcf    pclath,3
        bsf     pclath,4
        btfss   alpha,6         ;xx101111
        goto    daeebit6is0
        goto    daeebit6is1
ppbit5is0 bcf   pclath,3
        bsf     pclath,4
        btfss   alpha,6         ;xx011111
        goto    dafbit6is0
        goto    dafbit6is1
ppbit5is1 bcf   pclath,3
        bsf     pclath,4
        btfss   alpha,6         ;xx111111
```

```
                goto    daffbit6is0
                goto    daffbit6is1

org     0x1000                          ;PAGE1
abit6is0        btfss   alpha,7         ;x00000000
                goto    abit7is0
                goto    abit7is1
abit6is1        btfss   alpha,7         ;x10000000
                goto    aabit7is0
                goto    aabit7is1
aabit6is0       btfss   alpha,7         ;x0100000
                goto    bbit7is0
                goto    bbit7is1
aabit6is1       btfss   alpha,7         ;x1100000
                goto    bbbit7is0
                goto    bbbit7is1
bbit6is0        btfss   alpha,7         ;x0010000
                goto    cbit7is0
                goto    cbit7is1
bbit6is1        btfss   alpha,7         ;x1010000
                goto    ccbit7is0
                goto    ccbit7is1
bbbit6is0       btfss   alpha,7         ;x0110000
                goto    dbit7is0
                goto    dbit7is1
bbbit6is1       btfss   alpha,7         ;x1110000
                goto    ddbit7is0
                goto    ddbit7is1
cbit6is0        btfss   alpha,7         ;x0001000
                goto    ebit7is0
                goto    ebit7is1
cbit6is1        btfss   alpha,7         ;x1001000
                goto    eebit7is0
                goto    eebit7is1
ccbit6is0       btfss   alpha,7         ;x0101000
                goto    fbit7is0
                goto    fbit7is1
ccbit6is1       btfss   alpha,7         ;x1101000
                goto    ffbit7is0
                goto    ffbit7is1
dbit6is0        btfss   alpha,7         ;x0011000
                goto    gbit7is0
                goto    gbit7is1
dbit6is1        btfss   alpha,7         ;x1011000
                goto    ggbit7is0
                goto    ggbit7is1
ddbit6is0       btfss   alpha,7         ;x0111000
                goto    hbit7is0
                goto    hbit7is1
ddbit6is1       btfss   alpha,7         ;x1111000
                goto    hhbit7is0
                goto    hhbit7is1
ebit6is0        btfss   alpha,7         ;x0000100
                goto    ibit7is0
                goto    ibit7is1
ebit6is1        btfss   alpha,7         ;x1000100
                goto    iibit7is0
                goto    iibit7is1
eebit6is0       btfss   alpha,7         ;x0100100
                goto    jbit7is0
                goto    jbit7is1
eebit6is1       btfss   alpha,7         ;x1100100
                goto    jjbit7is0
                goto    jjbit7is1
fbit6is0        btfss   alpha,7         ;x0010100
                goto    kbit7is0
                goto    kbit7is1
fbit6is1        btfss   alpha,7         ;x1010100
                goto    kkbit7is0
                goto    kkbit7is1
ffbit6is0       btfss   alpha,7         ;x0110100
                goto    lbit7is0
                goto    lbit7is1
ffbit6is1       btfss   alpha,7         ;x1110100
                goto    llbit7is0
                goto    llbit7is1
gbit6is0        btfss   alpha,7         ;x0001100
                goto    mbit7is0
                goto    mbit7is1
gbit6is1        btfss   alpha,7         ;x1001100
                goto    mmbit7is0
                goto    mmbit7is1
ggbit6is0       bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0101100
                goto    nbit7is0
                goto    nbit7is1
ggbit6is1       bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1101100
                goto    nnbit7is0
                goto    nnbit7is1
hbit6is0        bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0011100
                goto    obit7is0
                goto    obit7is1
hbit6is1        bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1011100
                goto    oobit7is0
```

```
                goto    oobit7is1
hhbit6is0       bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0111100
                goto    pbit7is0
                goto    pbit7is1
hhbit6is1       bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1111100
                goto    ppbit7is0
                goto    ppbit7is1
ibit6is0        bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0000010
                goto    qbit7is0
                goto    qbit7is1
ibit6is1        bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1000010
                goto    qqbit7is0
                goto    qqbit7is1
iibit6is0       bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0100010
                goto    rbit7is0
                goto    rbit7is1
iibit6is1       bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1100010
                goto    rrbit7is0
                goto    rrbit7is1
jbit6is0        bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0010010
                goto    sbit7is0
                goto    sbit7is1
jbit6is1        bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1010010
                goto    ssbit7is0
                goto    ssbit7is1
jjbit6is0       bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0110010
                goto    tbit7is0
                goto    tbit7is1
jjbit6is1       bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1110010
                goto    ttbit7is0
                goto    ttbit7is1
kbit6is0        bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0001010
                goto    ubit7is0
                goto    ubit7is1
kbit6is1        bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1001010
                goto    uubit7is0
                goto    uubit7is1
kkbit6is0       bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0101010
                goto    vbit7is0
                goto    vbit7is1
kkbit6is1       bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1101010
                goto    vvbit7is0
                goto    vvbit7is1
lbit6is0        bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0011010
                goto    wbit7is0
                goto    wbit7is1
lbit6is1        bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1011010
                goto    wwbit7is0
                goto    wwbit7is1
llbit6is0       bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0111010
                goto    xbit7is0
                goto    xbit7is1
llbit6is1       bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1111010
                goto    xxbit7is0
                goto    xxbit7is1
mbit6is0        bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0000110
                goto    ybit7is0
                goto    ybit7is1
mbit6is1        bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1000110
                goto    yybit7is0
                goto    yybit7is1
```

```
mmbit6is0     bsf     pclath,3
              bsf     pclath,4
              btfss   alpha,7         ;x0100110
              goto    zbit7is0
              goto    zbit7is1
mmbit6is1     bsf     pclath,3
              bsf     pclath,4
              btfss   alpha,7         ;x1100110
              goto    zzbit7is0
              goto    zzbit7is1
nbit6is0      bsf     pclath,3
              bsf     pclath,4
              btfss   alpha,7         ;x0010110
              goto    fdabit7is0
              goto    fdabit7is1
nbit6is1      bsf     pclath,3
              bsf     pclath,4
              btfss   alpha,7         ;x1010110
              goto    fdaabit7is0
              goto    fdaabit7is1
nnbit6is0     bsf     pclath,3
              bsf     pclath,4
              btfss   alpha,7         ;x0110110
              goto    fdbbit7is0
              goto    fdbbit7is1
nnbit6is1     bsf     pclath,3
              bsf     pclath,4
              btfss   alpha,7         ;x1110110
              goto    fdbbbit7is0
              goto    fdbbbit7is1
obit6is0      bsf     pclath,3
              bsf     pclath,4
              btfss   alpha,7         ;x0001110
              goto    fdcbit7is0
              goto    fdcbit7is1
obit6is1      bsf     pclath,3
              bsf     pclath,4
              btfss   alpha,7         ;x1001110
              goto    fdccbit7is0
              goto    fdccbit7is1
oobit6is0     bsf     pclath,3
              bsf     pclath,4
              btfss   alpha,7         ;x0101110
              goto    fddbit7is0
              goto    fddbit7is1
oobit6is1     bsf     pclath,3
              bsf     pclath,4
              btfss   alpha,7         ;x1101110
              goto    fdddbit7is0
              goto    fdddbit7is1
pbit6is0      bsf     pclath,3
              bsf     pclath,4
              btfss   alpha,7         ;x0011110
              goto    fdebit7is0
              goto    fdebit7is1
pbit6is1      bsf     pclath,3
              bsf     pclath,4
              btfss   alpha,7         ;x1011110
              goto    fdeebit7is0
              goto    fdeebit7is1
ppbit6is0     bsf     pclath,3
              bsf     pclath,4
              btfss   alpha,7         ;x0111110
              goto    fdfbit7is0
              goto    fdfbit7is1
ppbit6is1     bsf     pclath,3
              bsf     pclath,4
              btfss   alpha,7         ;x1111110
              goto    fdffbit7is0
              goto    fdffbit7is1
qbit6is0      bsf     pclath,3
              bsf     pclath,4
              btfss   alpha,7         ;x0000001
              goto    fdgbit7is0
              goto    fdgbit7is1
qbit6is1      bsf     pclath,3
              bsf     pclath,4
              btfss   alpha,7         ;x1000001
              goto    fdggbit7is0
              goto    fdggbit7is1
qqbit6is0     bsf     pclath,3
              bsf     pclath,4
              btfss   alpha,7         ;x0100001
              goto    fdhbit7is0
              goto    fdhbit7is1
qqbit6is1     bsf     pclath,3
              bsf     pclath,4
              btfss   alpha,7         ;x1100001
              goto    fdhhbit7is0
              goto    fdhhbit7is1
rbit6is0      bsf     pclath,3
              bsf     pclath,4
              btfss   alpha,7         ;x0010001
              goto    fdibit7is0
              goto    fdibit7is1
rbit6is1      bsf     pclath,3
              bsf     pclath,4
              btfss   alpha,7         ;x1010001
              goto    fdiibit7is0
              goto    fdiibit7is1
rrbit6is0     bsf     pclath,3
```

```
                bsf     pclath,4
                btfss   alpha,7         ;x0110001
                goto    fdjbit7is0
                goto    fdjbit7is1
rrbit6is1       bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1110001
                goto    fdjjbit7is0
                goto    fdjjbit7is1
sbit6is0        bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0001001
                goto    fdkbit7is0
                goto    fdkbit7is1
sbit6is1        bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1001001
                goto    fdkkbit7is0
                goto    fdkkbit7is1
ssbit6is0       bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0101001
                goto    fdlbit7is0
                goto    fdlbit7is1
ssbit6is1       bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1101001
                goto    fdllbit7is0
                goto    fdllbit7is1
tbit6is0        bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0011001
                goto    fdmbit7is0
                goto    fdmbit7is1
tbit6is1        bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1011001
                goto    fdmmbit7is0
                goto    fdmmbit7is1
ttbit6is0       bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0111001
                goto    fdnbit7is0
                goto    fdnbit7is1
ttbit6is1       bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1111001
                goto    fdnnbit7is0
                goto    fdnnbit7is1
ubit6is0        bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0000101
                goto    fdobit7is0
                goto    fdobit7is1
ubit6is1        bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1000101
                goto    fdoobit7is0
                goto    fdoobit7is1
uubit6is0       bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0100101
                goto    fdpbit7is0
                goto    fdpbit7is1
uubit6is1       bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1100101
                goto    fdppbit7is0
                goto    fdppbit7is1
vbit6is0        bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0010101
                goto    fdqbit7is0
                goto    fdqbit7is1
vbit6is1        bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1010101
                goto    fdqqbit7is0
                goto    fdqqbit7is1
vvbit6is0       bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0110101
                goto    fdrbit7is0
                goto    fdrbit7is1
vvbit6is1       bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1110101
                goto    fdrrbit7is0
                goto    fdrrbit7is1
wbit6is0        bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0001101
                goto    fdsbit7is0
                goto    fdsbit7is1
wbit6is1        bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1001101
                goto    fdssbit7is0
                goto    fdssbit7is1
wwbit6is0       bsf     pclath,3
                bsf     pclath,4
```

```
                btfss   alpha,7         ;x0101101
                goto    fdtbit7is0
                goto    fdtbit7is1
wwbit6is1       bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1101101
                goto    fdttbit7is0
                goto    fdttbit7is1
xbit6is0        bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0011101
                goto    fdubit7is0
                goto    fdubit7is1
xbit6is1        bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1011101
                goto    fduubit7is0
                goto    fduubit7is1
xxbit6is0       bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0111101
                goto    fdvbit7is0
                goto    fdvbit7is1
xxbit6is1       bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1111101
                goto    fdvvbit7is0
                goto    fdvvbit7is1
ybit6is0        bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0000011
                goto    fdwbit7is0
                goto    fdwbit7is1
ybit6is1        bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1000011
                goto    fdwwbit7is0
                goto    fdwwbit7is1
yybit6is0       bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0100011
                goto    fdxbit7is0
                goto    fdxbit7is1
yybit6is1       bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1100011
                goto    fdxxbit7is0
                goto    fdxxbit7is1
zbit6is0        bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0010011
                goto    fdybit7is0
                goto    fdybit7is1
zbit6is1        bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1010011
                goto    fdyybit7is0
                goto    fdyybit7is1
zzbit6is0       bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0110011
                goto    fdzbit7is0
                goto    fdzbit7is1
zzbit6is1       bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1110011
                goto    fdzzbit7is0
                goto    fdzzbit7is1
daabit6is0      bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0001011
                goto    gfabit7is0
                goto    gfabit7is1
daabit6is1      bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1001011
                goto    gfaabit7is0
                goto    gfaabit7is1
daaabit6is0     bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0101011
                goto    gfbbit7is0
                goto    gfbbit7is1
daaabit6is1     bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1101011
                goto    gfbbbit7is0
                goto    gfbbbit7is1
dabbit6is0      bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0011011
                goto    gfcbit7is0
                goto    gfcbit7is1
dabbit6is1      bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1011011
                goto    gfccbit7is0
                goto    gfccbit7is1
dabbbit6is0     bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0111011
```

```
                goto    gfdbit7is0
                goto    gfdbit7is1
dabbbit6is1     bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1111011
                goto    gfddbit7is0
                goto    gfddbit7is1
dacbit6is0      bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0000111
                goto    gfebit7is0
                goto    gfebit7is1
dacbit6is1      bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1000111
                goto    gfeebit7is0
                goto    gfeebit7is1
daccbit6is0     bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0100111
                goto    gffbit7is0
                goto    gffbit7is1
daccbit6is1     bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1100111
                goto    gfffbit7is0
                goto    gfffbit7is1
dadbit6is0      bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0010111
                goto    gfgbit7is0
                goto    gfgbit7is1
dadbit6is1      bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1010111
                goto    gfggbit7is0
                goto    gfggbit7is1
daddbit6is0     bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0110111
                goto    gfhbit7is0
                goto    gfhbit7is1
daddbit6is1     bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1110111
                goto    gfhhbit7is0
                goto    gfhhbit7is1
daebit6is0      bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0001111
                goto    gfibit7is0
                goto    gfibit7is1
daebit6is1      bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1001111
                goto    gfiibit7is0
                goto    gfiibit7is1
daeebit6is0     bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0101111
                goto    gfjbit7is0
                goto    gfjbit7is1
daeebit6is1     bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1101111
                goto    gfjjbit7is0
                goto    gfjjbit7is1
dafbit6is0      bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0011111
                goto    gfkbit7is0
                goto    gfkbit7is1
dafbit6is1      bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1011111
                goto    gfkkbit7is0
                goto    gfkkbit7is1
daffbit6is0     bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x0111111
                goto    gflbit7is0
                goto    gflbit7is1
daffbit6is1     bsf     pclath,3
                bsf     pclath,4
                btfss   alpha,7         ;x1111111
                goto    gfllbit7is0
                goto    gfllbit7is1

abit7is0        movlw   b'00110000'     ;00000000   000     1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00111101'     ;                   2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00100001'     ;                   3rd
                movwf   deccnt6         ;decimal value in
                goto    showit3
abit7is1        movlw   b'00110001'     ;10000000   128     1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110010'     ;                   2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00111000'     ;                   3rd
```

```
            movwf   deccnt6         ;decimal value in
            goto    showit3
aabit7is0   movlw   b'00110000'     ;01000000   064     1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110110'     ;                   2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110100'     ;                   3rd
            movwf   deccnt6         ;decimal value in
            goto    showit3
aabit7is1   movlw   b'00110001'     ;11000000   192     1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00111001'     ;                   2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110010'     ;                   3rd
            movwf   deccnt6         ;decimal value in
            goto    showit3
bbit7is0    movlw   b'00110000'     ;00100000   032     1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110011'     ;                   2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110010'     ;                   3rd
            movwf   deccnt6         ;decimal value in
            goto    showit3
bbit7is1    movlw   b'00110001'     ;10100000   160     1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110110'     ;                   2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110000'     ;                   3rd
            movwf   deccnt6         ;decimal value in
            goto    showit3
bbbit7is0   movlw   b'00110000'     ;01100000   96      1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00111001'     ;                   2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110110'     ;                   3rd
            movwf   deccnt6         ;decimal value in
            goto    showit3
bbbit7is1   movlw   b'00110010'     ;11100000   224     1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110010'     ;                   2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110100'     ;                   3rd
            movwf   deccnt6         ;decimal value in
            goto    showit3
cbit7is0    movlw   b'00110000'     ;00010000   16      1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110001'     ;                   2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110110'     ;                   3rd
            movwf   deccnt6         ;decimal value in
            goto    showit3
cbit7is1    movlw   b'00110001'     ;10010000   144     1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110100'     ;                   2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110100'     ;                   3rd
            movwf   deccnt6         ;decimal value in
            goto    showit3
ccbit7is0   movlw   b'00110000'     ;01010000   80      1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00111000'     ;                   2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110000'     ;                   3rd
            movwf   deccnt6         ;decimal value in
            goto    showit3
ccbit7is1   movlw   b'00110010'     ;11010000   208     1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110000'     ;                   2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00111000'     ;                   3rd
            movwf   deccnt6         ;decimal value in
            goto    showit3
dbit7is0    movlw   b'00110000'     ;00110000   48      1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110100'     ;                   2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00111000'     ;                   3rd
            movwf   deccnt6         ;decimal value in
            goto    showit3
dbit7is1    movlw   b'00110001'     ;10110000   176     1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110111'     ;                   2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110110'     ;                   3rd
            movwf   deccnt6         ;decimal value in
            goto    showit3
ddbit7is0   movlw   b'00110001'     ;01110000   112     1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110001'     ;                   2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110010'     ;                   3rd
            movwf   deccnt6         ;decimal value in
            goto    showit3
ddbit7is1   movlw   b'00110010'     ;11110000   240     1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110100'     ;                   2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110000'     ;                   3rd
            movwf   deccnt6         ;decimal value in
            goto    showit3
ebit7is0    movlw   b'00110000'     ;00001000   8       1st
```

```
            movwf   deccnt4         ;decimal value in
            movlw   b'00110000'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00111000'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit3
ebit7is1    movlw   b'00110001'     ;10001000    136        1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110011'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110110'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit3
eebit7is0   movlw   b'00110000'     ;01001000    72         1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110111'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110010'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit3
eebit7is1   movlw   b'00110010'     ;11001000    200        1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110000'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110000'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit3
fbit7is0    movlw   b'00110000'     ;00101000    40         1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110100'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110000'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit3
fbit7is1    movlw   b'00110001'     ;10101000    168        1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110110'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00111000'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit3
ffbit7is0   movlw   b'00110001'     ;01101000    104        1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110000'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110100'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit3
ffbit7is1   movlw   b'00110010'     ;11101000    232        1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110011'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110010'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit3
gbit7is0    movlw   b'00110000'     ;00011000    24         1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110010'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110100'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit3
gbit7is1    movlw   b'00110001'     ;10011000    152        1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110101'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110010'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit3
ggbit7is0   movlw   b'00110000'     ;01011000    88         1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00111000'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00111000'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit3
ggbit7is1   movlw   b'00110010'     ;11011000    216        1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110001'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110110'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit3
hbit7is0    movlw   b'00110000'     ;00111000    56         1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110101'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110110'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit3
hbit7is1    movlw   b'00110001'     ;10111000    184        1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00111000'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110100'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit3
hhbit7is0   movlw   b'00110001'     ;01111000    120        1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110010'     ;                       2nd
            movwf   deccnt5         ;decimal value in
```

```
                movlw   b'00110000'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit3
hhbit7is1       bcf     pclath,3
                bsf     pclath,4
                movlw   b'00110010'     ;11111000       248     1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110100'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00111000'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit3
ibit7is0        movlw   b'00110000'     ;00000100       4       1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110000'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110100'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit3
ibit7is1        movlw   b'00110001'     ;10000100       132     1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110011'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110010'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit3
iibit7is0       movlw   b'00110000'     ;01000100       68      1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110110'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00111000'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit3
iibit7is1       movlw   b'00110001'     ;11000100       196     1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00111001'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110110'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit3
jbit7is0        movlw   b'00110000'     ;00100100       36      1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110011'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110110'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit3
jbit7is1        movlw   b'00110001'     ;10100100       164     1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110110'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110100'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit3
jjbit7is0       movlw   b'00110001'     ;01100100       100     1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110000'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110000'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit3
jjbit7is1       movlw   b'00110010'     ;11100100       228     1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110010'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00111000'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit3
kbit7is0        movlw   b'00110000'     ;00010100       20      1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110010'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110000'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit3
kbit7is1        movlw   b'00110001'     ;10010100       148     1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110100'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00111000'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit3
kkbit7is0       bcf     pclath,3
                bsf     pclath,4
                movlw   b'00110000'     ;01010100       84      1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00111000'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110100'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit3
kkbit7is1       movlw   b'00110010'     ;11010100       212     1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110001'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110010'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit3
lbit7is0        movlw   b'00110000'     ;00110100       52      1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110101'     ;                       2nd
```

```
                movwf   decnt5          ;decimal value in
                movlw   b'00110010'     ;                       3rd
                movwf   decnt6          ;decimal value in
                goto    showit3
lbit7is1        movlw   b'00110001'     ;10110100    180        1st
                movwf   decnt4          ;decimal value in
                movlw   b'00111000'     ;                       2nd
                movwf   decnt5          ;decimal value in
                movlw   b'00110000'     ;                       3rd
                movwf   decnt6          ;decimal value in
                goto    showit3
llbit7is0       movlw   b'00110001'     ;01110100    116        1st
                movwf   decnt4          ;decimal value in
                movlw   b'00110001'     ;                       2nd
                movwf   decnt5          ;decimal value in
                movlw   b'00110110'     ;                       3rd
                movwf   decnt6          ;decimal value in
                goto    showit3
llbit7is1       movlw   b'00110010'     ;11110100    244        1st
                movwf   decnt4          ;decimal value in
                movlw   b'00110100'     ;                       2nd
                movwf   decnt5          ;decimal value in
                movlw   b'00110100'     ;                       3rd
                movwf   decnt6          ;decimal value in
                goto    showit3
mbit7is0        movlw   b'00110000'     ;00001100    12         1st
                movwf   decnt4          ;decimal value in
                movlw   b'00110001'     ;                       2nd
                movwf   decnt5          ;decimal value in
                movlw   b'00110010'     ;                       3rd
                movwf   decnt6          ;decimal value in
                goto    showit3
mbit7is1        movlw   b'00110001'     ;10001100    140        1st
                movwf   decnt4          ;decimal value in
                movlw   b'00110100'     ;                       2nd
                movwf   decnt5          ;decimal value in
                movlw   b'00110000'     ;                       3rd
                movwf   decnt6          ;decimal value in
                goto    showit3
mmbit7is0       movlw   b'00110000'     ;01001100    76         1st
                movwf   decnt4          ;decimal value in
                movlw   b'00110111'     ;                       2nd
                movwf   decnt5          ;decimal value in
                movlw   b'00110110'     ;                       3rd
                movwf   decnt6          ;decimal value in
                goto    showit3
mmbit7is1       movlw   b'00110010'     ;11001100    204        1st
                movwf   decnt4          ;decimal value in
                movlw   b'00110000'     ;                       2nd
                movwf   decnt5          ;decimal value in
                movlw   b'00110100'     ;                       3rd
                movwf   decnt6          ;decimal value in
                goto    showit3

showit3         bcf     pclath,3
                bcf     pclath,4
                goto    showit1         ;PAGE0

org     0x1800                          ;PAGE2
nbit7is0        movlw   b'00110000'     ;00101100    44         1st
                movwf   decnt4          ;decimal value in
                movlw   b'00110100'     ;                       2nd
                movwf   decnt5          ;decimal value in
                movlw   b'00110100'     ;                       3rd
                movwf   decnt6          ;decimal value in
                goto    showit
nbit7is1        movlw   b'00110001'     ;10101100    172        1st
                movwf   decnt4          ;decimal value in
                movlw   b'00110111'     ;                       2nd
                movwf   decnt5          ;decimal value in
                movlw   b'00110010'     ;                       3rd
                movwf   decnt6          ;decimal value in
                goto    showit
nnbit7is0       movlw   b'00110001'     ;01101100    108        1st
                movwf   decnt4          ;decimal value in
                movlw   b'00110000'     ;                       2nd
                movwf   decnt5          ;decimal value in
                movlw   b'00111000'     ;                       3rd
                movwf   decnt6          ;decimal value in
                goto    showit
nnbit7is1       movlw   b'00110010'     ;11101100    236        1st
                movwf   decnt4          ;decimal value in
                movlw   b'00110011'     ;                       2nd
                movwf   decnt5          ;decimal value in
                movlw   b'00110110'     ;                       3rd
                movwf   decnt6          ;decimal value in
                goto    showit
obit7is0        movlw   b'00110000'     ;00011100    28         1st
                movwf   decnt4          ;decimal value in
                movlw   b'00110010'     ;                       2nd
                movwf   decnt5          ;decimal value in
                movlw   b'00111000'     ;                       3rd
                movwf   decnt6          ;decimal value in
                goto    showit
obit7is1        movlw   b'00110001'     ;10011100    156        1st
                movwf   decnt4          ;decimal value in
                movlw   b'00110101'     ;                       2nd
                movwf   decnt5          ;decimal value in
                movlw   b'00110110'     ;                       3rd
                movwf   decnt6          ;decimal value in
                goto    showit
```

```
oobit7is0   movlw   b'00110000'   ;01011100    92     1st
            movwf   deccnt4       ;decimal value in
            movlw   b'00111001'   ;                   2nd
            movwf   deccnt5       ;decimal value in
            movlw   b'00110010'   ;                   3rd
            movwf   deccnt6       ;decimal value in
            goto    showit
oobit7is1   movlw   b'00110010'   ;11011100    220    1st
            movwf   deccnt4       ;decimal value in
            movlw   b'00110010'   ;                   2nd
            movwf   deccnt5       ;decimal value in
            movlw   b'00110000'   ;                   3rd
            movwf   deccnt6       ;decimal value in
            goto    showit
pbit7is0    movlw   b'00110000'   ;00111100    60     1st
            movwf   deccnt4       ;decimal value in
            movlw   b'00110110'   ;                   2nd
            movwf   deccnt5       ;decimal value in
            movlw   b'00110000'   ;                   3rd
            movwf   deccnt6       ;decimal value in
            goto    showit
pbit7is1    movlw   b'00110001'   ;10111100    188    1st
            movwf   deccnt4       ;decimal value in
            movlw   b'00111000'   ;                   2nd
            movwf   deccnt5       ;decimal value in
            movlw   b'00111000'   ;                   3rd
            movwf   deccnt6       ;decimal value in
            goto    showit
ppbit7is0   movlw   b'00110001'   ;01111100    124    1st
            movwf   deccnt4       ;decimal value in
            movlw   b'00110010'   ;                   2nd
            movwf   deccnt5       ;decimal value in
            movlw   b'00110100'   ;                   3rd
            movwf   deccnt6       ;decimal value in
            goto    showit
ppbit7is1   movlw   b'00110010'   ;11111100    252    1st
            movwf   deccnt4       ;decimal value in
            movlw   b'00110101'   ;                   2nd
            movwf   deccnt5       ;decimal value in
            movlw   b'00110010'   ;                   3rd
            movwf   deccnt6       ;decimal value in
            goto    showit
qbit7is0    movlw   b'00110000'   ;00000010    2      1st
            movwf   deccnt4       ;decimal value in
            movlw   b'00110000'   ;                   2nd
            movwf   deccnt5       ;decimal value in
            movlw   b'00110010'   ;                   3rd
            movwf   deccnt6       ;decimal value in
            goto    showit
qbit7is1    movlw   b'00110001'   ;10000010    130    1st
            movwf   deccnt4       ;decimal value in
            movlw   b'00110011'   ;                   2nd
            movwf   deccnt5       ;decimal value in
            movlw   b'00110000'   ;                   3rd
            movwf   deccnt6       ;decimal value in
            goto    showit
qqbit7is0   movlw   b'00110000'   ;01000010    66     1st
            movwf   deccnt4       ;decimal value in
            movlw   b'00110110'   ;                   2nd
            movwf   deccnt5       ;decimal value in
            movlw   b'00110110'   ;                   3rd
            movwf   deccnt6       ;decimal value in
            goto    showit
qqbit7is1   movlw   b'00110001'   ;11000010    194    1st
            movwf   deccnt4       ;decimal value in
            movlw   b'00111001'   ;                   2nd
            movwf   deccnt5       ;decimal value in
            movlw   b'00110100'   ;                   3rd
            movwf   deccnt6       ;decimal value in
            goto    showit
rbit7is0    movlw   b'00110000'   ;00100010    34     1st
            movwf   deccnt4       ;decimal value in
            movlw   b'00110011'   ;                   2nd
            movwf   deccnt5       ;decimal value in
            movlw   b'00110100'   ;                   3rd
            movwf   deccnt6       ;decimal value in
            goto    showit
rbit7is1    movlw   b'00110001'   ;10100010    162    1st
            movwf   deccnt4       ;decimal value in
            movlw   b'00110110'   ;                   2nd
            movwf   deccnt5       ;decimal value in
            movlw   b'00110010'   ;                   3rd
            movwf   deccnt6       ;decimal value in
            goto    showit
rrbit7is0   movlw   b'00110000'   ;01100010    98     1st
            movwf   deccnt4       ;decimal value in
            movlw   b'00111001'   ;                   2nd
            movwf   deccnt5       ;decimal value in
            movlw   b'00111000'   ;                   3rd
            movwf   deccnt6       ;decimal value in
            goto    showit
rrbit7is1   movlw   b'00110010'   ;11100010    226    1st
            movwf   deccnt4       ;decimal value in
            movlw   b'00110010'   ;                   2nd
            movwf   deccnt5       ;decimal value in
            movlw   b'00110110'   ;                   3rd
            movwf   deccnt6       ;decimal value in
            goto    showit
sbit7is0    movlw   b'00110000'   ;00010010    18     1st
            movwf   deccnt4       ;decimal value in
            movlw   b'00110001'   ;                   2nd
```

```
            movwf   deccnt5         ;decimal value in
            movlw   b'00111000'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
sbit7is1    movlw   b'00110001'     ;10010010   146         1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110100'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110110'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
ssbit7is0   movlw   b'00110000'     ;01010010   82          1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00111000'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110010'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
ssbit7is1   movlw   b'00110010'     ;11010010   210         1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110001'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110000'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
tbit7is0    movlw   b'00110000'     ;00110010   50          1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110101'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110000'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
tbit7is1    movlw   b'00110001'     ;10110010   178         1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110111'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00111000'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
ttbit7is0   movlw   b'00110001'     ;01110010   114         1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110001'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110100'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
ttbit7is1   movlw   b'00110010'     ;11110010   242         1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110100'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110010'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
ubit7is0    movlw   b'00110000'     ;00001010   10          1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110001'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110000'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
ubit7is1    movlw   b'00110001'     ;10001010   138         1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110011'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00111000'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
uubit7is0   movlw   b'00110000'     ;01001010   74          1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110111'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110100'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
uubit7is1   movlw   b'00110010'     ;11001010   202         1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110000'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110010'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
vbit7is0    movlw   b'00110000'     ;00101010   42          1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110100'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110010'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
vbit7is1    movlw   b'00110001'     ;10101010   170         1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110111'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110000'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
vvbit7is0   movlw   b'00110001'     ;01101010   106         1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110000'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110110'     ;                       3rd
            movwf   deccnt6         ;decimal value in
```

```
                goto    showit
vvbit7is1       movlw   b'00110010'     ;11101010   234     1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110011'     ;                   2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110100'     ;                   3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
wbit7is0        movlw   b'00110000'     ;00011010   26      1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110010'     ;                   2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110110'     ;                   3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
wbit7is1        movlw   b'00110001'     ;10011010   154     1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110101'     ;                   2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110100'     ;                   3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
wwbit7is0       movlw   b'00110000'     ;01011010   90      1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00111001'     ;                   2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110000'     ;                   3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
wwbit7is1       movlw   b'00110010'     ;11011010   218     1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110001'     ;                   2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00111000'     ;                   3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
xbit7is0        movlw   b'00110000'     ;00111010   58      1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110101'     ;                   2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00111000'     ;                   3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
xbit7is1        movlw   b'00110001'     ;10111010   186     1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00111000'     ;                   2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110110'     ;                   3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
xxbit7is0       movlw   b'00110001'     ;01111010   122     1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110010'     ;                   2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110010'     ;                   3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
xxbit7is1       movlw   b'00110010'     ;11111010   250     1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110101'     ;                   2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110000'     ;                   3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
ybit7is0        movlw   b'00110000'     ;00000110   6       1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110000'     ;                   2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110110'     ;                   3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
ybit7is1        movlw   b'00110001'     ;10000110   134     1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110011'     ;                   2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110100'     ;                   3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
yybit7is0       movlw   b'00110000'     ;01000110   70      1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110111'     ;                   2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110000'     ;                   3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
yybit7is1       movlw   b'00110001'     ;11000110   198     1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00111001'     ;                   2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00111000'     ;                   3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
zbit7is0        movlw   b'00110000'     ;00100110   38      1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110011'     ;                   2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00111000'     ;                   3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
zbit7is1        movlw   b'00110001'     ;10100110   166     1st
                movwf   deccnt4         ;decimal value in
```

```
            movlw   b'00110110'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110110'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
zzbit7is0   movlw   b'00110001'     ;01100110    102        1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110000'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110010'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
zzbit7is1   movlw   b'00110010'     ;11100110    230        1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110011'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110000'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdabit7is0  movlw   b'00110000'     ;00010110    22         1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110010'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110010'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdabit7is1  movlw   b'00110001'     ;10010110    150        1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110101'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110000'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdaabit7is0 movlw   b'00110000'     ;01010110    86         1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00111000'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110110'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdaabit7is1 movlw   b'00110010'     ;11010110    214        1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110001'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110100'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdbbit7is0  movlw   b'00110000'     ;00110110    54         1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110101'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110100'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdbbit7is1  movlw   b'00110001'     ;10110110    182        1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00111000'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110010'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdbbbit7is0 movlw   b'00110001'     ;01110110    118        1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110001'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00111000'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdbbbit7is1 movlw   b'00110010'     ;11110110    246        1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110100'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110110'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdcbit7is0  movlw   b'00110000'     ;00001110    14         1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110001'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110100'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdcbit7is1  movlw   b'00110001'     ;10001110    142        1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110100'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110010'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdccbit7is0 movlw   b'00110000'     ;01001110    78         1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110111'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00111000'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdccbit7is1 movlw   b'00110010'     ;11001110    206        1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110000'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110110'     ;                       3rd
```

```
            movwf   deccnt6         ;decimal value in
            goto    showit
fddbit7is0  movlw   b'00110000'     ;00101110     46     1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110100'     ;                    2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110110'     ;                    3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fddbit7is1  movlw   b'00110001'     ;10101110     174    1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110111'     ;                    2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110100'     ;                    3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdddbit7is0 movlw   b'00110001'     ;01101110     110    1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110001'     ;                    2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110000'     ;                    3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdddbit7is1 movlw   b'00110010'     ;11101110     238    1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110011'     ;                    2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00111000'     ;                    3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdebit7is0  movlw   b'00110000'     ;00011110     30     1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110011'     ;                    2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110000'     ;                    3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdebit7is1  movlw   b'00110001'     ;10011110     158    1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110101'     ;                    2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00111000'     ;                    3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdeebit7is0 movlw   b'00110000'     ;01011110     94     1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00111001'     ;                    2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110100'     ;                    3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdeebit7is1 movlw   b'00110010'     ;11011110     222    1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110010'     ;                    2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110010'     ;                    3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdfbit7is0  movlw   b'00110000'     ;00111110     62     1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110110'     ;                    2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110010'     ;                    3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdfbit7is1  movlw   b'00110001'     ;10111110     190    1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00111001'     ;                    2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110000'     ;                    3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdffbit7is0 movlw   b'00110001'     ;01111110     126    1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110010'     ;                    2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110110'     ;                    3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdffbit7is1 movlw   b'00110010'     ;00111100     254    1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110101'     ;                    2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110100'     ;                    3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdgbit7is0  movlw   b'00110000'     ;00000001     1      1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110000'     ;                    2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110001'     ;                    3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdgbit7is1  movlw   b'00110001'     ;10000001     129    1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110010'     ;                    2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00111001'     ;                    3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdggbit7is0 movlw   b'00110000'     ;01000001     65     1st
```

```
                movwf   deccnt4         ;decimal value in
                movlw   b'00110110'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110101'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
fdggbit7is1     movlw   b'00110001'     ;11000001    193        1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00111001'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110011'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
fdhbit7is0      movlw   b'00110000'     ;00100001    33         1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110011'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110011'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
fdhbit7is1      movlw   b'00110001'     ;10100001    161        1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110110'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110001'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
fdhhbit7is0     movlw   b'00110000'     ;01100001    97         1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00111001'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110111'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
fdhhbit7is1     movlw   b'00110010'     ;11100001    225        1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110010'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110101'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
fdibit7is0      movlw   b'00110000'     ;00010001    17         1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110001'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110111'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
fdibit7is1      movlw   b'00110001'     ;10010001    145        1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110100'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110101'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
fdiibit7is0     movlw   b'00110000'     ;01010001    81         1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00111000'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110001'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
fdiibit7is1     movlw   b'00110010'     ;11010001    209        1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110000'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00111001'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
fdjbit7is0      movlw   b'00110000'     ;00110001    49         1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110100'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00111001'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
fdjbit7is1      movlw   b'00110001'     ;10110001    177        1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110111'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110111'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
fdjjbit7is0     movlw   b'00110001'     ;01110001    113        1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110001'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110011'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
fdjjbit7is1     movlw   b'00110010'     ;11110001    241        1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110100'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110001'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
fdkbit7is0      movlw   b'00110000'     ;00001001    9          1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110000'     ;                       2nd
                movwf   deccnt5         ;decimal value in
```

```
            movlw   b'00111001'     ;                   3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdkbit7is1  movlw   b'00110001'     ;10001001   137     1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110011'     ;                   2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110111'     ;                   3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdkkbit7is0 movlw   b'00110000'     ;01001001   73      1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110111'     ;                   2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110011'     ;                   3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdkkbit7is1 movlw   b'00110010'     ;11001001   201     1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110000'     ;                   2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110001'     ;                   3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdlbit7is0  movlw   b'00110000'     ;00101001   41      1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110100'     ;                   2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110001'     ;                   3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdlbit7is1  movlw   b'00110001'     ;10101001   169     1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110110'     ;                   2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00111001'     ;                   3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdllbit7is0 movlw   b'00110001'     ;01101001   105     1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110000'     ;                   2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110101'     ;                   3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdllbit7is1 movlw   b'00110010'     ;11101001   233     1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110011'     ;                   2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110011'     ;                   3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdmbit7is0  movlw   b'00110000'     ;00011001   25      1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110010'     ;                   2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110101'     ;                   3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdmbit7is1  movlw   b'00110001'     ;10011001   153     1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110101'     ;                   2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110011'     ;                   3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdmmbit7is0 movlw   b'00110000'     ;01011001   89      1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00111000'     ;                   2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00111001'     ;                   3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdmmbit7is1 movlw   b'00110010'     ;11011001   217     1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110001'     ;                   2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110111'     ;                   3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdnbit7is0  movlw   b'00110000'     ;00111001   57      1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110101'     ;                   2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110111'     ;                   3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdnbit7is1  movlw   b'00110001'     ;10111001   185     1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00111000'     ;                   2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110101'     ;                   3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
fdnnbit7is0 movlw   b'00110001'     ;01111001   121     1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110010'     ;                   2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110001'     ;                   3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
```

```
fdnnbit7is1    movlw   b'00110010'   ;11111001    249    1st
               movwf   deccnt4       ;decimal value in
               movlw   b'00110100'   ;                   2nd
               movwf   deccnt5       ;decimal value in
               movlw   b'00111001'   ;                   3rd
               movwf   deccnt6       ;decimal value in
               goto    showit
fdobit7is0     movlw   b'00110000'   ;00000101    5      1st
               movwf   deccnt4       ;decimal value in
               movlw   b'00110000'   ;                   2nd
               movwf   deccnt5       ;decimal value in
               movlw   b'00110101'   ;                   3rd
               movwf   deccnt6       ;decimal value in
               goto    showit
fdobit7is1     movlw   b'00110001'   ;10000101    133    1st
               movwf   deccnt4       ;decimal value in
               movlw   b'00110011'   ;                   2nd
               movwf   deccnt5       ;decimal value in
               movlw   b'00110011'   ;                   3rd
               movwf   deccnt6       ;decimal value in
               goto    showit
fdoobit7is0    movlw   b'00110000'   ;01000101    69     1st
               movwf   deccnt4       ;decimal value in
               movlw   b'00110110'   ;                   2nd
               movwf   deccnt5       ;decimal value in
               movlw   b'00111001'   ;                   3rd
               movwf   deccnt6       ;decimal value in
               goto    showit
fdoobit7is1    movlw   b'00110001'   ;11000101    197    1st
               movwf   deccnt4       ;decimal value in
               movlw   b'00111001'   ;                   2nd
               movwf   deccnt5       ;decimal value in
               movlw   b'00110111'   ;                   3rd
               movwf   deccnt6       ;decimal value in
               goto    showit
fdpbit7is0     movlw   b'00110000'   ;00100101    37     1st
               movwf   deccnt4       ;decimal value in
               movlw   b'00110011'   ;                   2nd
               movwf   deccnt5       ;decimal value in
               movlw   b'00110111'   ;                   3rd
               movwf   deccnt6       ;decimal value in
               goto    showit
fdpbit7is1     movlw   b'00110001'   ;10100101    165    1st
               movwf   deccnt4       ;decimal value in
               movlw   b'00110110'   ;                   2nd
               movwf   deccnt5       ;decimal value in
               movlw   b'00110101'   ;                   3rd
               movwf   deccnt6       ;decimal value in
               goto    showit
fdppbit7is0    movlw   b'00110001'   ;01100101    101    1st
               movwf   deccnt4       ;decimal value in
               movlw   b'00110000'   ;                   2nd
               movwf   deccnt5       ;decimal value in
               movlw   b'00110001'   ;                   3rd
               movwf   deccnt6       ;decimal value in
               goto    showit
fdppbit7is1    movlw   b'00110010'   ;11100101    229    1st
               movwf   deccnt4       ;decimal value in
               movlw   b'00110010'   ;                   2nd
               movwf   deccnt5       ;decimal value in
               movlw   b'00111001'   ;                   3rd
               movwf   deccnt6       ;decimal value in
               goto    showit
fdqbit7is0     movlw   b'00110000'   ;00010101    21     1st
               movwf   deccnt4       ;decimal value in
               movlw   b'00110010'   ;                   2nd
               movwf   deccnt5       ;decimal value in
               movlw   b'00110001'   ;                   3rd
               movwf   deccnt6       ;decimal value in
               goto    showit
fdqbit7is1     movlw   b'00110001'   ;10010101    149    1st
               movwf   deccnt4       ;decimal value in
               movlw   b'00110100'   ;                   2nd
               movwf   deccnt5       ;decimal value in
               movlw   b'00111001'   ;                   3rd
               movwf   deccnt6       ;decimal value in
               goto    showit
fdqqbit7is0    movlw   b'00110000'   ;01010101    85     1st
               movwf   deccnt4       ;decimal value in
               movlw   b'00111000'   ;                   2nd
               movwf   deccnt5       ;decimal value in
               movlw   b'00110101'   ;                   3rd
               movwf   deccnt6       ;decimal value in
               goto    showit
fdqqbit7is1    movlw   b'00110010'   ;11010101    213    1st
               movwf   deccnt4       ;decimal value in
               movlw   b'00110001'   ;                   2nd
               movwf   deccnt5       ;decimal value in
               movlw   b'00110011'   ;                   3rd
               movwf   deccnt6       ;decimal value in
               goto    showit
fdrbit7is0     movlw   b'00110000'   ;00110101    53     1st
               movwf   deccnt4       ;decimal value in
               movlw   b'00110101'   ;                   2nd
               movwf   deccnt5       ;decimal value in
               movlw   b'00110011'   ;                   3rd
               movwf   deccnt6       ;decimal value in
               goto    showit
fdrbit7is1     movlw   b'00110001'   ;10110101    181    1st
               movwf   deccnt4       ;decimal value in
               movlw   b'00111000'   ;                   2nd
```

```
		movwf	deccnt5		;decimal value in
		movlw	b'00110001'	;		3rd
		movwf	deccnt6		;decimal value in
		goto	showit
fdrrbit7is0	movlw	b'00110001'	;01110101	117	1st
		movwf	deccnt4		;decimal value in
		movlw	b'00110001'	;		2nd
		movwf	deccnt5		;decimal value in
		movlw	b'00110111'	;		3rd
		movwf	deccnt6		;decimal value in
		goto	showit
fdrrbit7is1	movlw	b'00110010'	;11110101	245	1st
		movwf	deccnt4		;decimal value in
		movlw	b'00110100'	;		2nd
		movwf	deccnt5		;decimal value in
		movlw	b'00110101'	;		3rd
		movwf	deccnt6		;decimal value in
		goto	showit
fdsbit7is0	movlw	b'00110000'	;00001101	13	1st
		movwf	deccnt4		;decimal value in
		movlw	b'00110001'	;		2nd
		movwf	deccnt5		;decimal value in
		movlw	b'00110011'	;		3rd
		movwf	deccnt6		;decimal value in
		goto	showit
fdsbit7is1	movlw	b'00110001'	;10001101	141	1st
		movwf	deccnt4		;decimal value in
		movlw	b'00110100'	;		2nd
		movwf	deccnt5		;decimal value in
		movlw	b'00110001'	;		3rd
		movwf	deccnt6		;decimal value in
		goto	showit
fdssbit7is0	movlw	b'00110000'	;01001101	77	1st
		movwf	deccnt4		;decimal value in
		movlw	b'00110111'	;		2nd
		movwf	deccnt5		;decimal value in
		movlw	b'00110111'	;		3rd
		movwf	deccnt6		;decimal value in
		goto	showit
fdssbit7is1	movlw	b'00110010'	;11001101	205	1st
		movwf	deccnt4		;decimal value in
		movlw	b'00110000'	;		2nd
		movwf	deccnt5		;decimal value in
		movlw	b'00110101'	;		3rd
		movwf	deccnt6		;decimal value in
		goto	showit
fdtbit7is0	movlw	b'00110000'	;00101101	45	1st
		movwf	deccnt4		;decimal value in
		movlw	b'00110100'	;		2nd
		movwf	deccnt5		;decimal value in
		movlw	b'00110101'	;		3rd
		movwf	deccnt6		;decimal value in
		goto	showit
fdtbit7is1	movlw	b'00110001'	;10101101	173	1st
		movwf	deccnt4		;decimal value in
		movlw	b'00110111'	;		2nd
		movwf	deccnt5		;decimal value in
		movlw	b'00110011'	;		3rd
		movwf	deccnt6		;decimal value in
		goto	showit
fdttbit7is0	movlw	b'00110001'	;01101101	109	1st
		movwf	deccnt4		;decimal value in
		movlw	b'00110000'	;		2nd
		movwf	deccnt5		;decimal value in
		movlw	b'00111001'	;		3rd
		movwf	deccnt6		;decimal value in
		goto	showit
fdttbit7is1	movlw	b'00110010'	;11101101	237	1st
		movwf	deccnt4		;decimal value in
		movlw	b'00110011'	;		2nd
		movwf	deccnt5		;decimal value in
		movlw	b'00110111'	;		3rd
		movwf	deccnt6		;decimal value in
		goto	showit
fdubit7is0	movlw	b'00110000'	;00011101	29	1st
		movwf	deccnt4		;decimal value in
		movlw	b'00110010'	;		2nd
		movwf	deccnt5		;decimal value in
		movlw	b'00111001'	;		3rd
		movwf	deccnt6		;decimal value in
		goto	showit
fdubit7is1	movlw	b'00110001'	;10011101	157	1st
		movwf	deccnt4		;decimal value in
		movlw	b'00110101'	;		2nd
		movwf	deccnt5		;decimal value in
		movlw	b'00110111'	;		3rd
		movwf	deccnt6		;decimal value in
		goto	showit
fduubit7is0	movlw	b'00110000'	;01011101	93	1st
		movwf	deccnt4		;decimal value in
		movlw	b'00111001'	;		2nd
		movwf	deccnt5		;decimal value in
		movlw	b'00110011'	;		3rd
		movwf	deccnt6		;decimal value in
		goto	showit
fduubit7is1	movlw	b'00110010'	;11011101	221	1st
		movwf	deccnt4		;decimal value in
		movlw	b'00110010'	;		2nd
		movwf	deccnt5		;decimal value in
		movlw	b'00110001'	;		3rd
		movwf	deccnt6		;decimal value in
```

```
                goto    showit
fdvbit7is0      movlw   b'00110000'     ;00111101   61      1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110110'     ;                   2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110001'     ;                   3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
fdvbit7is1      movlw   b'00110001'     ;10111101   189     1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00111000'     ;                   2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00111001'     ;                   3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
fdvvbit7is0     movlw   b'00110001'     ;01111101   125     1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110010'     ;                   2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110101'     ;                   3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
fdvvbit7is1     movlw   b'00110010'     ;11111101   253     1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110101'     ;                   2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110011'     ;                   3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
fdwbit7is0      movlw   b'00110000'     ;00000011   3       1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110000'     ;                   2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110011'     ;                   3rd
                movwf                   ;decimal value in
                goto    showit
fdwbit7is1      movlw   b'00110001'     ;10000011   131     1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110011'     ;                   2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110001'     ;                   3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
fdwwbit7is0     movlw   b'00110000'     ;01000011   67      1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110110'     ;                   2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110111'     ;                   3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
fdwwbit7is1     movlw   b'00110001'     ;11000011   195     1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00111001'     ;                   2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110101'     ;                   3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
fdxbit7is0      movlw   b'00110000'     ;00100011   35      1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110011'     ;                   2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110101'     ;                   3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
fdxbit7is1      movlw   b'00110001'     ;10100011   163     1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110110'     ;                   2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110011'     ;                   3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
fdxxbit7is0     movlw   b'00110000'     ;01100011   99      1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00111001'     ;                   2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00111001'     ;                   3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
fdxxbit7is1     movlw   b'00110010'     ;11100011   227     1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110010'     ;                   2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110111'     ;                   3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
fdybit7is0      movlw   b'00110000'     ;00010011   19      1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110001'     ;                   2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00111001'     ;                   3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
fdybit7is1      movlw   b'00110001'     ;10010011   147     1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110100'     ;                   2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110111'     ;                   3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
fdyybit7is0     movlw   b'00110000'     ;01010011   83      1st
                movwf   deccnt4         ;decimal value in
```

```
                movlw   b'00111000'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110011'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
fdyybit7is1     movlw   b'00110010'     ;11010011     211       1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110001'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110001'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
fdzbit7is0      movlw   b'00110000'     ;00110011     51        1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110101'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110001'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
fdzbit7is1      movlw   b'00110001'     ;10110011     179       1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110111'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00111001'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
fdzzbit7is0     movlw   b'00110001'     ;01110011     115       1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110001'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110101'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
fdzzbit7is1     movlw   b'00110010'     ;11110011     243       1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110100'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110011'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
gfabit7is0      movlw   b'00110000'     ;00001011     11        1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110001'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110001'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
gfabit7is1      movlw   b'00110001'     ;10001011     139       1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110011'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00111001'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
gfaabit7is0     movlw   b'00110000'     ;01001011     75        1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110111'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110101'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
gfaabit7is1     movlw   b'00110010'     ;11001011     203       1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110000'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110011'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
gfbbit7is0      movlw   b'00110000'     ;00101011     43        1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110100'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110011'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
gfbbit7is1      movlw   b'00110001'     ;10101011     171       1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110111'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110001'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
gfbbbit7is0     movlw   b'00110001'     ;01101011     107       1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110000'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110111'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
gfbbbit7is1     movlw   b'00110010'     ;11101011     235       1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110011'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110101'     ;                       3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
gfcbit7is0      movlw   b'00110000'     ;00011011     27        1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110010'     ;                       2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110111'     ;                       3rd
```

```
                movwf   deccnt6         ;decimal value in
                goto    showit
gfcbit7is1      movlw   b'00110001'     ;10011011     155      1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110101'     ;                      2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110101'     ;                      3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
gfccbit7is0     movlw   b'00110000'     ;01011011     91       1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00111001'     ;                      2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110001'     ;                      3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
gfccbit7is1     movlw   b'00110010'     ;11011011     219      1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110001'     ;                      2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00111001'     ;                      3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
gfdbit7is0      movlw   b'00110000'     ;00111011     59       1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110101'     ;                      2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00111001'     ;                      3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
gfdbit7is1      movlw   b'00110001'     ;10111011     187      1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00111000'     ;                      2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110111'     ;                      3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
gfddbit7is0     movlw   b'00110001'     ;01111011     123      1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110010'     ;                      2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110011'     ;                      3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
gfddbit7is1     movlw   b'00110010'     ;11111011     251      1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110101'     ;                      2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110001'     ;                      3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
gfebit7is0      movlw   b'00110000'     ;00000111     7        1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110000'     ;                      2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110111'     ;                      3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
gfebit7is1      movlw   b'00110001'     ;10000111     135      1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110011'     ;                      2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110101'     ;                      3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
gfeebit7is0     movlw   b'00110000'     ;01000111     71       1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110111'     ;                      2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110001'     ;                      3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
gfeebit7is1     movlw   b'00110001'     ;11000111     199      1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00111001'     ;                      2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00111001'     ;                      3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
gffbit7is0      movlw   b'00110000'     ;00100111     39       1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110011'     ;                      2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00111001'     ;                      3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
gffbit7is1      movlw   b'00110001'     ;10100111     167      1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110110'     ;                      2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110111'     ;                      3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
gfffbit7is0     movlw   b'00110001'     ;01100111     103      1st
                movwf   deccnt4         ;decimal value in
                movlw   b'00110000'     ;                      2nd
                movwf   deccnt5         ;decimal value in
                movlw   b'00110011'     ;                      3rd
                movwf   deccnt6         ;decimal value in
                goto    showit
gfffbit7is1     movlw   b'00110010'     ;11100111     231      1st
```

```
            movwf   deccnt4         ;decimal value in
            movlw   b'00110011'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110001'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
gfgbit7is0  movlw   b'00110000'     ;00010111    23         1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110010'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110011'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
gfgbit7is1  movlw   b'00110001'     ;10010111    151        1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110101'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110001'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
gfggbit7is0 movlw   b'00110000'     ;01010111    87         1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00111000'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110111'  ;                          3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
gfggbit7is1 movlw   b'00110010'     ;11010111    215        1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110001'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110101'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
gfhbit7is0  movlw   b'00110000'     ;00110111    55         1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110101'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110101'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
gfhbit7is1  movlw   b'00110001'     ;10110111    183        1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00111000'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110011'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
gfhhbit7is0 movlw   b'00110001'     ;01110111    119        1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110001'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00111001'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
gfhhbit7is1 movlw   b'00110010'     ;11110111    247        1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110100'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110111'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
gfibit7is0  movlw   b'00110000'     ;00001111    15         1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110001'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110101'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
gfibit7is1  movlw   b'00110001'     ;10001111    143        1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110100'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110011'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
gfiibit7is0 movlw   b'00110000'     ;01001111    79         1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110111'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00111001'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
gfiibit7is1 movlw   b'00110010'     ;11001111    207        1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110000'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110111'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
gfjbit7is0  movlw   b'00110000'     ;00101111    47         1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110100'     ;                       2nd
            movwf   deccnt5         ;decimal value in
            movlw   b'00110111'     ;                       3rd
            movwf   deccnt6         ;decimal value in
            goto    showit
gfjbit7is1  movlw   b'00110001'     ;10101111    175        1st
            movwf   deccnt4         ;decimal value in
            movlw   b'00110111'     ;                       2nd
            movwf   deccnt5         ;decimal value in
```

```
		movlw	b'00110101'	;		3rd
		movwf	deccnt6		;decimal value in
		goto	showit
gfjjbit7is0	movlw	b'00110001'	;01101111	111	1st
		movwf	deccnt4		;decimal value in
		movlw	b'00110001'	;		2nd
		movwf	deccnt5		;decimal value in
		movlw	b'00110001'	;		3rd
		movwf	deccnt6		;decimal value in
		goto	showit
gfjjbit7is1	movlw	b'00110010'	;11101111	239	1st
		movwf	deccnt4		;decimal value in
		movlw	b'00110011'	;		2nd
		movwf	deccnt5		;decimal value in
		movlw	b'00111001'	;		3rd
		movwf	deccnt6		;decimal value in
		goto	showit
gfkbit7is0	movlw	b'00110000'	;00011111	31	1st
		movwf	deccnt4		;decimal value in
		movlw	b'00110011'	;		2nd
		movwf	deccnt5		;decimal value in
		movlw	b'00110001'	;		3rd
		movwf	deccnt6		;decimal value in
		goto	showit
gfkbit7is1	movlw	b'00110001'	;10011111	159	1st
		movwf	deccnt4		;decimal value in
		movlw	b'00110101'	;		2nd
		movwf	deccnt5		;decimal value in
		movlw	b'00111001'	;		3rd
		movwf	deccnt6		;decimal value in
		goto	showit
gfkkbit7is0	movlw	b'00110000'	;01011111	95	1st
		movwf	deccnt4		;decimal value in
		movlw	b'00111001'	;		2nd
		movwf	deccnt5		;decimal value in
		movlw	b'00110101'	;		3rd
		movwf	deccnt6		;decimal value in
		goto	showit
gfkkbit7is1	movlw	b'00110010'	;11011111	223	1st
		movwf	deccnt4		;decimal value in
		movlw	b'00110010'	;		2nd
		movwf	deccnt5		;decimal value in
		movlw	b'00110011'	;		3rd
		movwf	deccnt6		;decimal value in
		goto	showit
gflbit7is0	movlw	b'00110000'	;00111111	63	1st
		movwf	deccnt4		;decimal value in
		movlw	b'00110110'	;		2nd
		movwf	deccnt5		;decimal value in
		movlw	b'00110011'	;		3rd
		movwf	deccnt6		;decimal value in
		goto	showit
gflbit7is1	movlw	b'00110001'	;10111111	191	1st
		movwf	deccnt4		;decimal value in
		movlw	b'00111001'	;		2nd
		movwf	deccnt5		;decimal value in
		movlw	b'00110001'	;		3rd
		movwf	deccnt6		;decimal value in
		goto	showit
gfllbit7is0	movlw	b'00110001'	;01111111	127	1st
		movwf	deccnt4		;decimal value in
		movlw	b'00110010'	;		2nd
		movwf	deccnt5		;decimal value in
		movlw	b'00110111'	;		3rd
		movwf	deccnt6		;decimal value in
		goto	showit
gfllbit7is1	movlw	b'00110010'	;11111111	255	1st
		movwf	deccnt4		;decimal value in
		movlw	b'00110101'	;		2nd
		movwf	deccnt5		;decimal value in
		movlw	b'00110101'	;		3rd
		movwf	deccnt6		;decimal value in
		goto	showit

showit		bcf	pclath,3
		bcf	pclath,4
		goto	showit1		;PAGE0


end
```

APPENDIX E

E.1 SOLDERABLE SELF-BONDING ENAMEL (SALDAVEX AUTOVEX F)

(as provided by Scientific Wire Company, London)

Applications

Round copper wires used in wirings of electrical machines and electrical instruments.

Chemical and Physical Properties

-Composition of resin modified polyurethane, polyamide, polyvinilbutiral (PVB), with self-bonding properties.

-Thermal class (IEC172) 155.

-Softening temperature of bonding layer 160-190°C.

Storage and Handling

-Special transport precautions – none.

-Special storage precautions – none.

-Personal protection measures – none.

Inflammability and Explosion Danger

-Not inflammable without primer.

-With primer it is inflammable at temperatures over 400°C.

-It is immediately auto-extinguishing.

-It is not explosive.

Toxicological Data

Hazardous fumes or dust may be generated when soldering, welding , hot staking, burning, wire brushing, melting, or processing magnet wires. Excessive exposure to dust may cause irritation to the eyes, skin, and/or respiratory systems. Wear proper protective eye, skin and breathing equipment. Avoid breathing fumes.

During the curing of AUTOVEX wire, some fumes are produced. These are made by the wire surface lubrication (wax and paraffin) and by the residue of AUTOVEX resin solvents (aromatic hydrocarbon, etilglycole).

These fumes are less than 0.01% of the mass. It is better to evacuate these solvents fumes by a cowl on the bonding place.

In case of total destruction for fire (for example by a persisting flame at approximately 700°C), before self-extinguishing, the product will exale CO, CO_2, H_2O steam and will leave a carbon residue.

Round wires are lubricated using a minimum quantity of wax materials of max 75mg/m^2 of wire surface, which for diameter ranging around 0.50mm corresponds to a max of 75mg per kg of wire, or following Customer Specifications.

Testing with such wires showed that when a high enough current was in the coil to start the self-bonding process the coil became more malleable and the temperature within the coil of over 100°C was usually sufficient to enable the process (the size of coil affected the temperature required due to the heating effects of copper losses).

Self-bonding wire is a wire which, when subject to a short duration of a higher current than normal operation rating current, will bond together and maintain its overall shape. It will not have short circuits if done correctly.

E.2 PACKING FACTOR CONSIDERATIONS

The range of values of packing factor for wire in the field and stator coils needs to be calculated. Since the wire is essentially circular in cross-section, then there must be a proportion of the allowed space for the coils which cannot be filled, the remaining fraction (the packing factor) is for copper (the coils) (E.1).

e.g. number of turns = N
radius of one copper turn = r
area of one turn = πr^2=a
total area = A

Packing factor = $PF = \frac{Na}{A} = \frac{N\pi r^2}{A}$ (E.1)

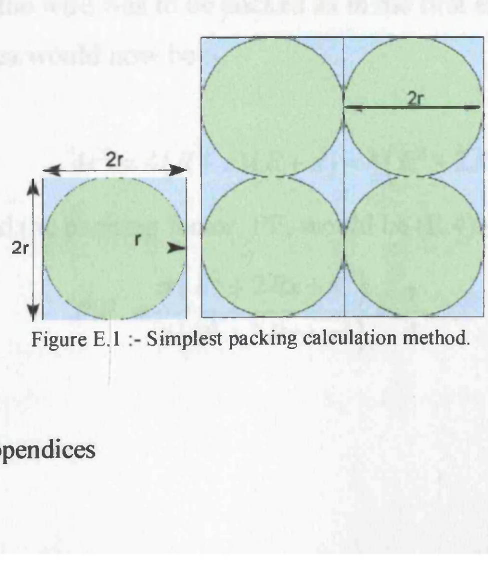


Figure E.1 :- Simplest packing calculation method.

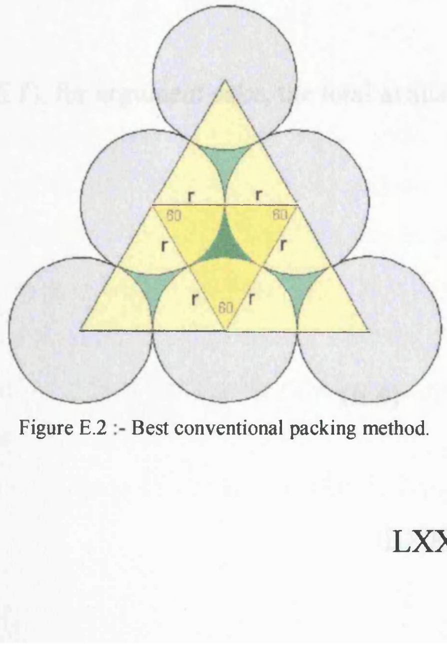


Figure E.2 :- Best conventional packing method.

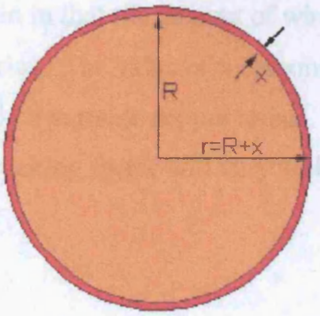


Figure E.3 :- Insulation layer.

Assume that the wire is packed as in *Figure E.1* (a simple winding style, viewable as a pessimistic method of winding the coils)

area of square = $4r^2$

area of circle = πr^2

ratio of circle area to square area = $\pi/4$ = 0.7854 = Packing Factor

A more efficient packing is as shown in *Figure E.2*. As shown, more wires occupy the squares (same length of 2r as the first example). The centres of each circle are linked by isosceles triangles. The area of each triangle can be shown to be $\sqrt{3}r^2$ where r is the radius of the wire. As shown the middle of each triangle is air space. The area which is copper is 3 individual 60° segments, each of area $\frac{1}{12}r^2$.

The packing factor is then $\dfrac{\sqrt{3}r^2 - 3\left(\frac{1}{12}r^2\right)}{\sqrt{3}r^2} = \dfrac{\left(\sqrt{3}-\frac{1}{4}\right)r^2}{\sqrt{3}r^2} = \dfrac{\left(\sqrt{3}-\frac{1}{4}\right)}{\sqrt{3}} = 0.855662$ (E.2)

The next problem to investigate is that of insulation layers, since it is common form to give diameter of the copper only, not of the diameter of the copper plus its insulation layer. A simple diagram highlighting the situation is shown in *Figure E.3*. Suppose the radius r is made up of the actual radius of copper, R, plus the insulation layer, x. Thus the total area is (E.3):-

$$\pi r^2 = \pi\left(R+x\right)\left(R+x\right) = \pi\left(R^2+2Rx+x^2\right) \quad \text{(E.3)}$$

If the wire was to be packed as in the first example (*Figure E.1*), for argument sake, the total available area would now be :-

$$4r^2 = 4\left(R+x\right)\left(R+x\right) = 4\left(R^2+2Rx+x^2\right)$$

and the packing factor, PF, would be (E.4):-

$$P.F. = \frac{\pi\left(R^2+2Rx+x^2\right)}{4\left(R^2+2Rx+x^2\right)} = \frac{\pi}{4} \quad \text{(E.4)}$$

This gives a packing factor of $\pi/4$ again in that the amount of wire that can be fitted in the space is a fixed proportion as the total radius varies. The value of x determines how much of this is actually for copper, since the total radius is actually a variable copper radius, R, plus an assumed fixed value of insulation thickness. Hence the true packing factor will vary with total wire radius (E.5):-.

$$P.F. = \frac{\pi R^2}{4\left(R^2 + 2Rx + x^2\right)} < \frac{\pi}{4} \tag{E.5}$$

For small changes in total wire radius where the insulation layer is small compared to the total wire thickness it can be acceptable to assume a calculated packing factor based on actual test data, especially if little effort was put into compacting the wire into the space. This is shown below where R is very much bigger than x (E.6):-.

$$R^2 + 2Rx + x^2 \rightarrow R^2 \qquad if \quad R >> x$$

thus $$P.F. = \frac{\pi R^2}{4\left(R^2 + 2Rx + x^2\right)} \qquad R >> x$$

hence $$P.F. = \frac{\pi R^2}{4\left(R^2\right)} = \frac{\pi}{4} \tag{E.6}$$

The equation shows that, in this situation, x has little or no real effect on the packing factor and hence the packing factor and true packing factor have almost identical values.

The more pessimistic a value of packing factor, the greater is the chances of the turns fitting within the given slot area e.g. 80% of the calculated value could be used for all wire types.

Self bonding wire has a third layer on its outside that is used to bond wires together during bonding treatment (*Figure E.4*). This third layer has caused a reduction in the packing factor for the wire. A conventional 0.315mm diameter wire in a manually wound toroidal coil can have a packing factor of 0.7. The self bonding 0.315mm equivalent in a similar sized toroidal coil only offers 0.52 as a packing factor. This would mean that an initial assumption of the packing factor of self bonding wire is 74% of that for similar sized conventional wire. This is due to the thickness, y, of the additional layer that performs self bonding. Let the wire thickness of 0.315mm be the size of the copper plus insulation (R+x) and let the self-bonding layer be y. Calculating the areas of a conventional wire and a self bonding wire and incorporating the packing factors allows the thickness of y to be calculated (E.7).

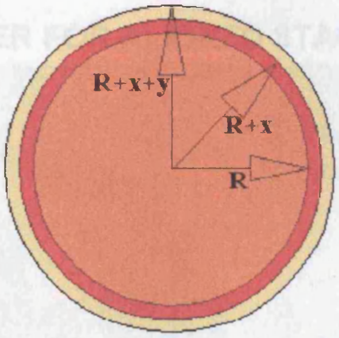


Figure E.4 :- Copper layer, insulation layer and self-bonding layer.

$$
\begin{aligned}
&\pi(0.315+y)^2 = \pi(0.315)^2\left(\frac{0.7}{0.52}\right) \\
&(0.315+y)^2 = (0.315)^2(1.346) = 0.134 \qquad \text{(E.7)} \\
&0.315+y = 0.365 \\
&y = 0.05\text{mm} = 50\mu\text{m}
\end{aligned}
$$

The self bonding wire is therefore 0.415mm in diameter based on this calculation, a 31% increase in size. The self bonding layer is 50µm thick. This explains the reduction in packing factor. The packing factor will improve with thicker wire choice to a value more similar to that of conventional copper wire.

E.3 USE OF PLASTIC FORMER FOR ALIGNED STACKS

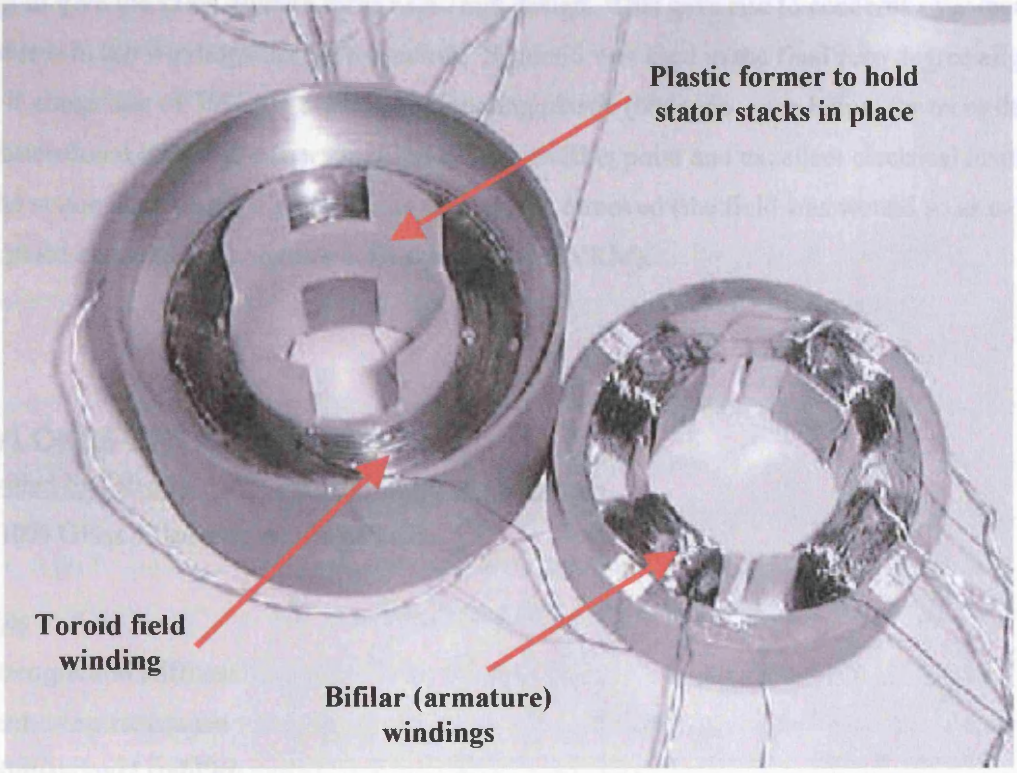


Figure E.5 :- Dual Stack Machine in section displaying field (toroid) and bifilar armature windings [test windings included].

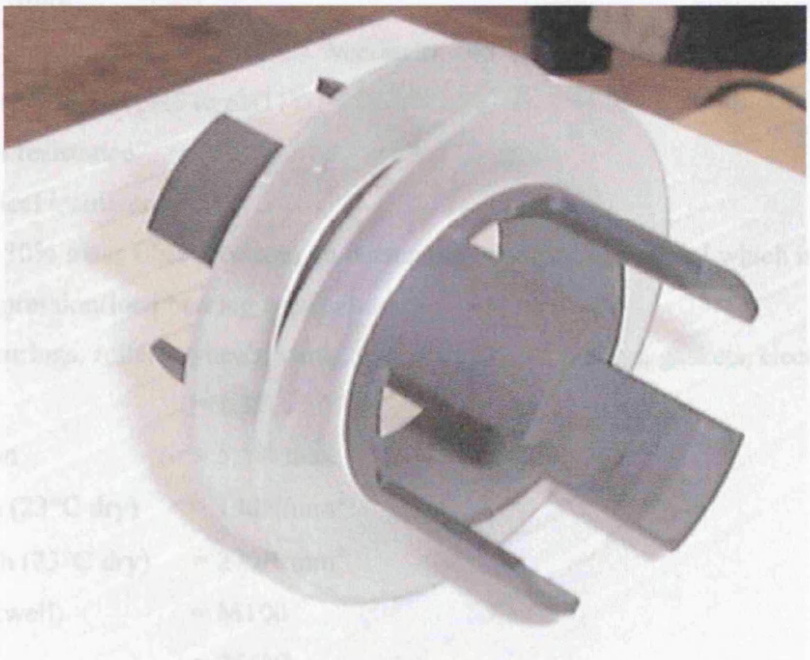


Figure E.6 :- Nylon66 former to hold stacks in place with two perspex discs to hold toroidal windings in place.

A plastic former is required to guarantee that the toroidal field winding was held in place and that the two stacks are held in perfect alignment (*Figures E.5* and *E.6*). This has to be incorporated into the design of the dual stack variable reluctance machine. Perspex was originally used but was found to be thermally unsuitable. 1mm thickness was also wanted as it allowed more copper area. However the former could only be made from 2mm thickness. This means that more current is required in the windings to give the same ampere turns as a 1mm design. This gave rise to concerns over increased temperatures in the windings during operation. Nylon66 was used in the final zero degree alignment model. It comprises of 30% glass-filled engineering plastic (Nylon66 – see below for more data). It offers dimensional stability, easier machining, high melting point and excellent electrical insulation. When the stator stacks were displaced, the former was removed (the field was wound so as to hold itself in place between the armature coils within the DSVRM).

E.4 NYLON66-ERTALON®66GF-30

(as provided by October 1999 Farnell Industrial catalogue)

This is 30% Glass-filled Engineering Plastic.

Properties

-High strength and stiffness
-Excellent creep resistance
-Good dimensional stability
-Black colour or white
-Continuous working temperature (120°C). Maximum 145°C
-Good chemical resistance (pH5 to pH11)
-Good hydrolysis resistance
-Excellent electrical insulator
-The addition of 30% glass fibre produces an outstanding composite material which is ideal for demanding compression/load bearing applications :-

gears, bearings, rollers, wheels, cams, nuts, valve seats, pulleys, gaskets, electrical insulators

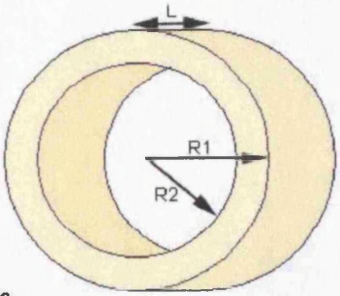
-Specific gravity = 1.35
-Water absorption = 5.5% max
-Tensile strength (23°C dry) = 190N/mm^2
-Flexural strength (23°C dry) = 270N/mm^2
-Hardness (Rockwell) = M100
-Melt point = 255°C
-Thermal conductivity = 0.24W/k.m
-Flammability = UL94-HB
-Volume resistivity = 1013Ωcm
-Dielectric strength = 45kV/mm
-Surface resistivity = 1020Ω

E.5 STEFAN'S LAW

Stefan's Law is used to calculate the radiated heat as a temperature rise above ambient temperature when the wire carried a current (E.8). The modelling used on Microsoft Excel was such that, for any calculated slot area, mmf proportions from the windings or plastic former thickness, at any specified flux density (B), the currents in the coils, the voltages required, the I^2R resistive losses and the associated temperature rises above ambient were instantaneously calculated. The design was optimised numerically to give the shape with minimum reluctance and optimum thermal characteristics. This allowed the maximization of the magnetic flux for a given supply.

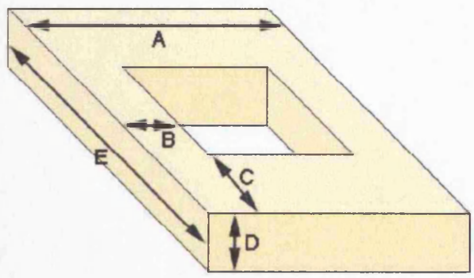
The temperature rise, T_{risen} (in Kelvin), had to have 293 Kelvin subtracted to get temperature rise above an ambient temperature of 20°C. The total surface areas of the windings were calculated (*Figures E.7, E.8*). For ease of calculation, each area was regarded as the surface area if each set of windings were just a single large wire of cross-sectional area equalling the proportion of the slot area available. Stefan's Law was only provides an indication of the radiated heat from the wires by linking the I^2R resistive losses to radiated heat into the air. Conduction and radiation by the outer surface of the motor were not taken into account. Work was carried out to model how heat was conducted into the motor's metal body and, from this, how heat was lost from the whole motor by radiation. This was halted as it would have become a research study in its own right. Work concentrated on maximizing the radiated heat loss from the stators as a whole and the field on its own. Equilibrium temperature rises from the motor at various flux densities (and hence currents) had to be measured experimentally when the motor was built, to verify the Stefan's Law calculations.

$$\begin{aligned} &P = \sigma\ Ae\left[\left(T_{risen}\right)^4 - \left(T_{original}\right)^4\right] \\ &\sigma = 6.67x10^{-8} \quad W/m^2K^4 \\ &A = surface \quad area/m^2 \\ &e = emissivity(0.7\ for\ copper) \\ &T = temperature/Kelvin \\ &P = i^2R \quad power \quad loss \end{aligned} \tag{E.8}$$



Total Surface Area = $2\pi\left[(R1)^2 - (R2)^2 + (R1)L + (R2)L\right]$

Figure E.7 :- Surface area calculations for field coil.



Total Surface Area = 4(AD+AC+BE+DE-2BC-BD-CD)

Figure E.8 :- Surface area calculations for an armature coil.

It is possible to calculate the thermal model for all the windings if the I^2R power losses and the dimensions are known for the windings in question. It gave an insight into the possible temperature performance of the machine. This theory assumed no fan was present.

The static and dynamic modelling process gives information about the machine. The power loss from the windings was known, the total surface area was calculated and the ambient temperature was used as $T_{original}$ leaving T_{risen} as the unknown. The temperature rise above ambient was found from subtracting $T_{original}$ from $T_{unknown}$.

APPENDIX F

F.1 DUAL STACK VARIABLE RELUCTANCE MACHINE CONCEPT DRAWING

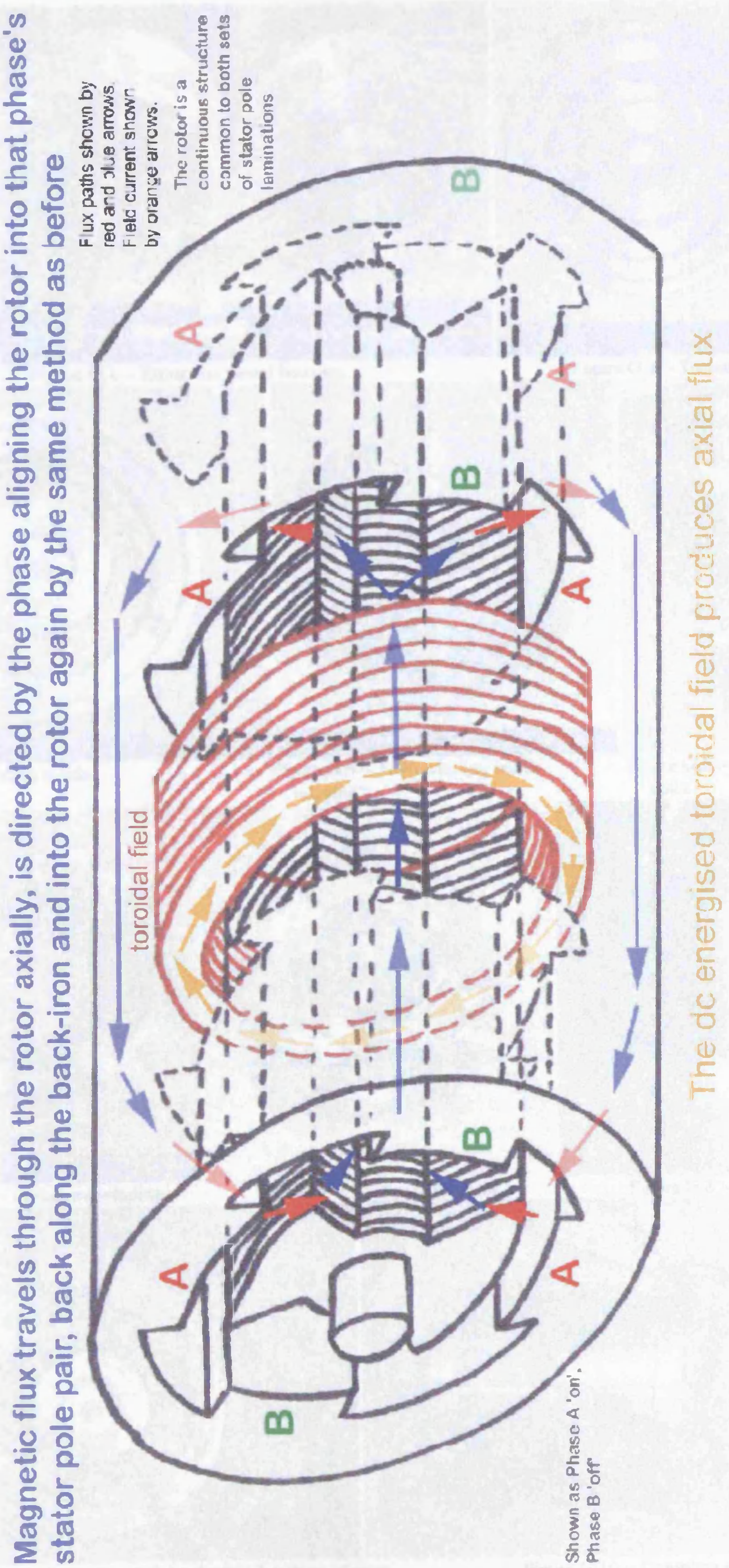


Figure F.1 :- Flux paths when using second method for winding type.

APPENDIX G

G.1 PHOTOGRAPHS OF SINGLE STACK VARIABLE RELUCTANCE MOTOR

Figure G.1 :- Rotor and toroid housing.

Figure G.2 :- Toroid housing.

Figure G.3 :- Field winding inside housing.

Figure G.4 :- Field winding inside housing.

Figure G.5 :- Field adjacent to armature coils.

Figure G.6:- Armature windings.

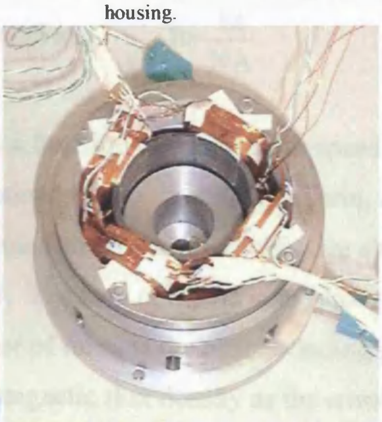


Figure G.7 :- Space for rotor.

Figure G.8 :- Cooling ducts.

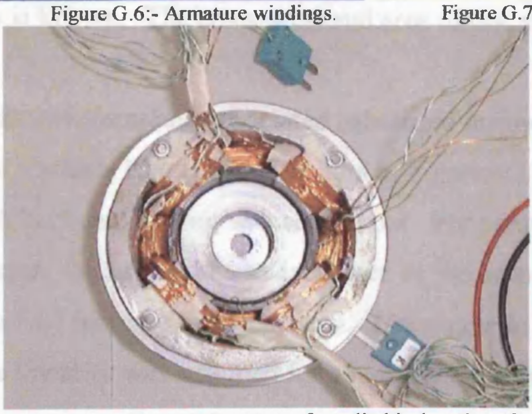


Figure G.9 :- Space for cylindrical section of rotor.

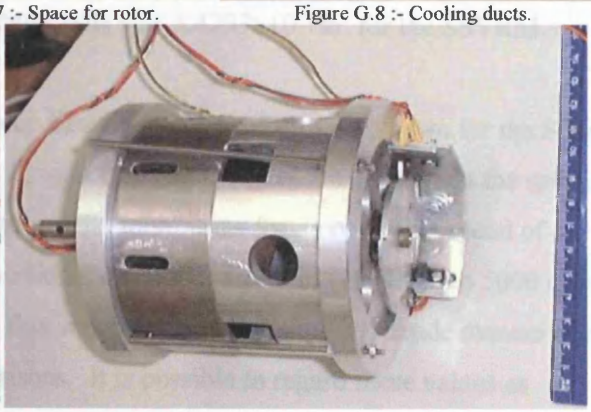


Figure G.10 :- Complete machine.

G.2 BACK EMF TEST RESULTS

The maximum speed obtained from the dc motor was slightly above 1100rpm. The two sets of stator coils (labelled as Phase1 and Phase2) were connected to the oscilloscope so the generated voltage (back-emf) could be seen and measured. The source of the magnetic flux for back-emf production was the field winding. The current in the field winding was varied for different speeds so an indication of the back emf waveform at higher speeds (e.g. 8000rpm) was obtainable through interpolation of the data (since the dc motor would not operate over 1105rpm).

The initial test was to ensure that the back emfs from each coil for each phase summed in the correct manner for each winding topology used to drive the machine. The total back emf waveform in one armature phase winding would be the same size but in antiphase to that in the other armature phase winding.

The magnetic flux produced can be calculated from the back emf waveform. From Faraday's Law (G.1):-

$$E=N\frac{d\Phi}{dt} \tag{G.1}$$

Integration of both sides will yield $\int Edt=\int Nd\Phi$

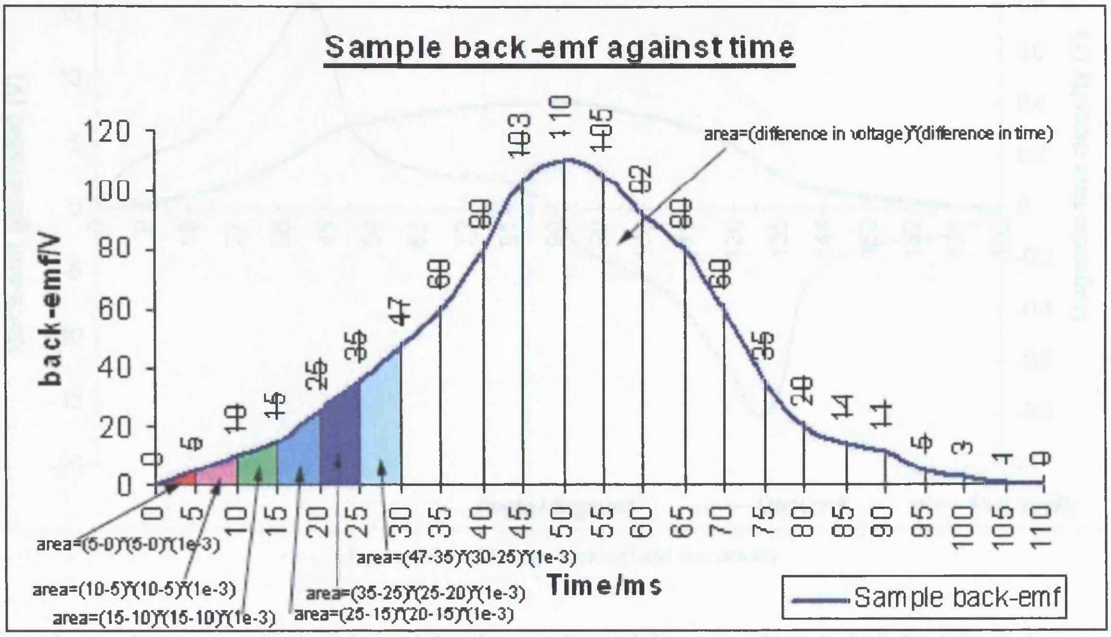
Further simplification gives the flux linkage $Et=N\Phi$

The magnetic flux is then $\Phi={Et}/{N}$ where $\Phi=BA$

The magnetic flux density is calculated as $B=\frac{Et}{NA}$

The magnetic flux linkage was thus a function of time (motor speed) and back-emf value (G.1). Instead of trying to integrate a non-sinusoidal back-emf waveform, calculating areas under the waveform using trapezoid area calculations is a fast and effective alternative, as shown in the example back-emf waveform of *Figure G.11*. The sum of all the individual trapezoids gives the magnetic flux linkage. Dividing this by the number of turns in the phase winding gives the magnetic flux. The magnetic flux can be converted to magnetic flux density as the cross-sectional area, A, of the stator poles is known. The cross-sectional area of one stator pole was $3.4793\text{x}10^{-4}$ m^2 for the SSVRM.

Actual waveform and associated calculated magnetic flux linkage waveforms are shown for the Single Stack Variable Reluctance Machine in *Figures G.12* to *G.15*. These values are accurate at the speed at which back-emf testing was carried out. For predicting the waveforms for an operating speed of 5000rpm, the back-emf was increased by the proportional difference in speed (1071rpm to 5000 rpm was a 4.67 increase in proportion). The magnetic flux was increased in a similarly crude manner and hence linearity was assumed for simplistic comparisons. It is possible to regard these values as acceptable since the actual calculated maximum magnetic flux density was in the region of about 0.8T (which is close to the point at which saturation would start).



Sum of areas as time increments gives the total magnetic flux linkage up to that point in time. If an area is negative (i.e. back-emk if negative) then the sum value decreases. This value, if divided by the number of turns in the armature winding, will give the magnetic flux value at that point in time Depending on the speed the time can be converted into electrical angle or mechanical angle.

Figure G.11 :-Summation of area under curve yields flux linkage.

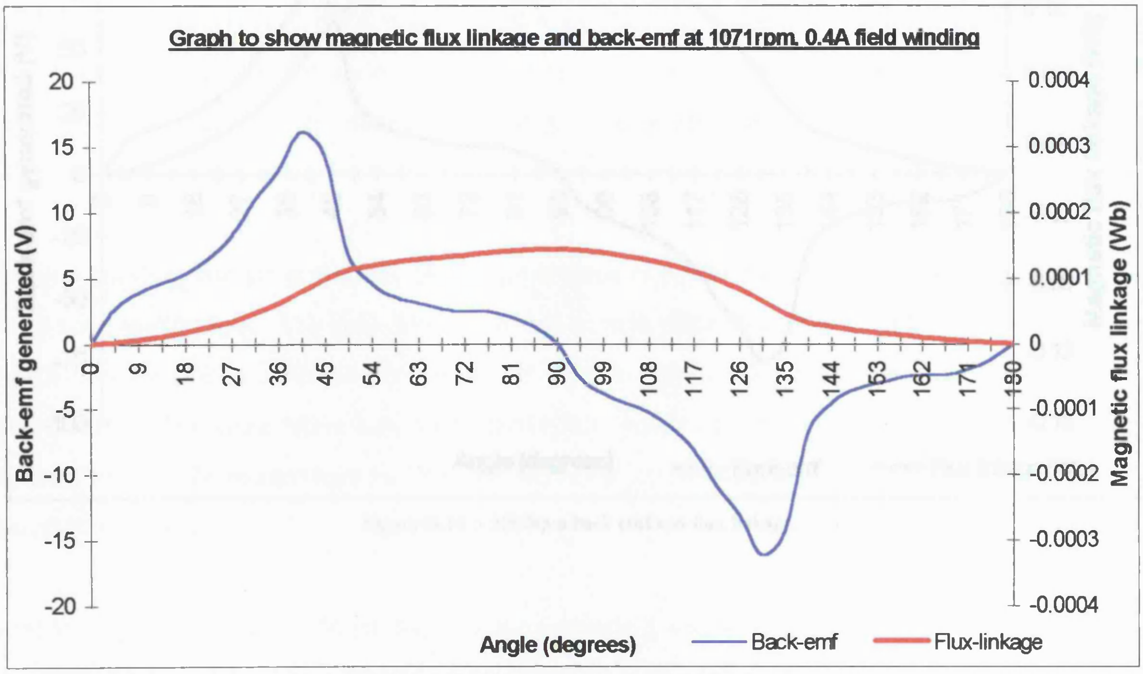


Figure G.12 :- 1071rpm back emf and flux linkage.

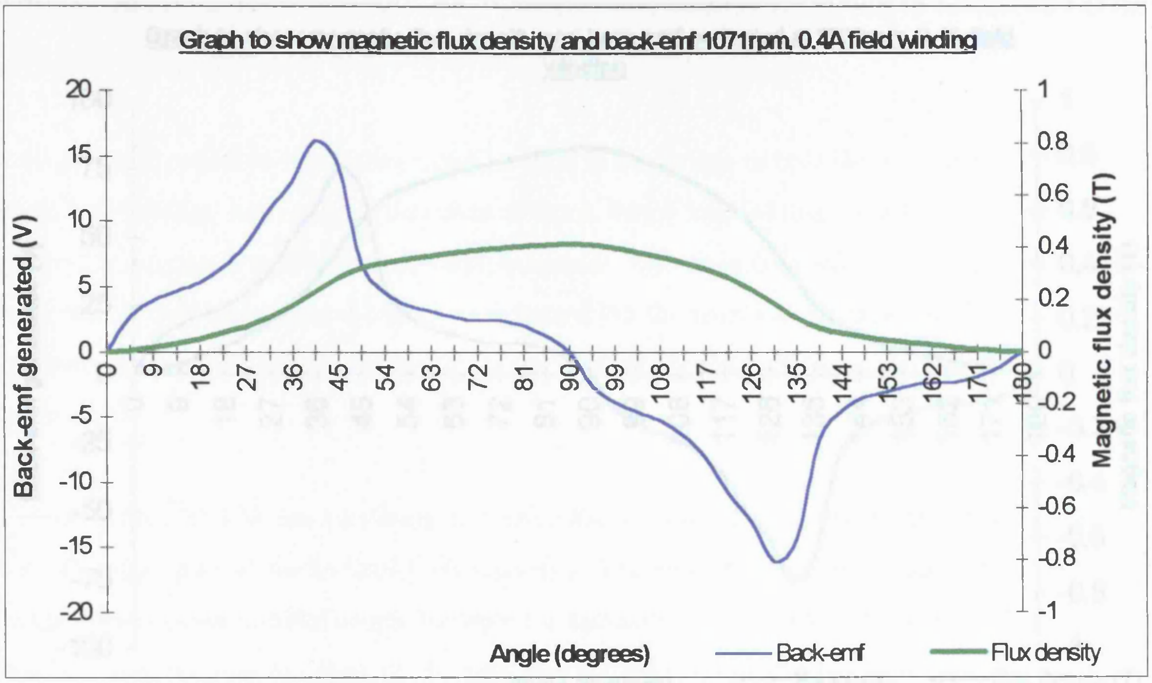


Figure G.13 :- 1071rpm back emf and flux density.

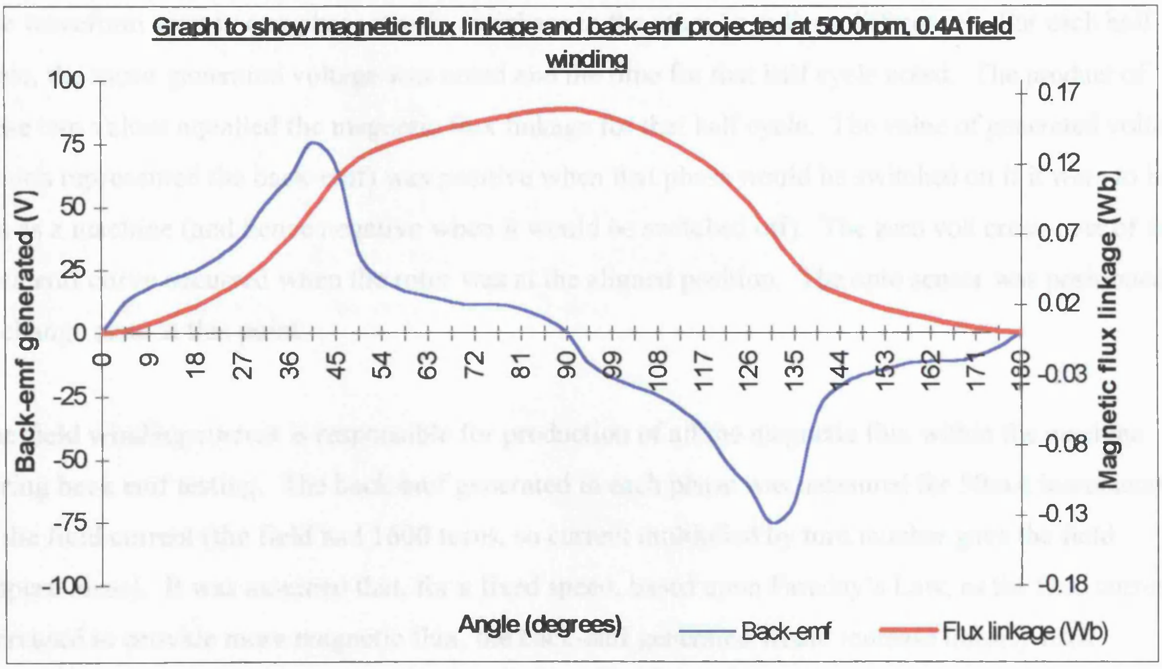


Figure G.14 :- 5000rpm back emf and flux linkage.

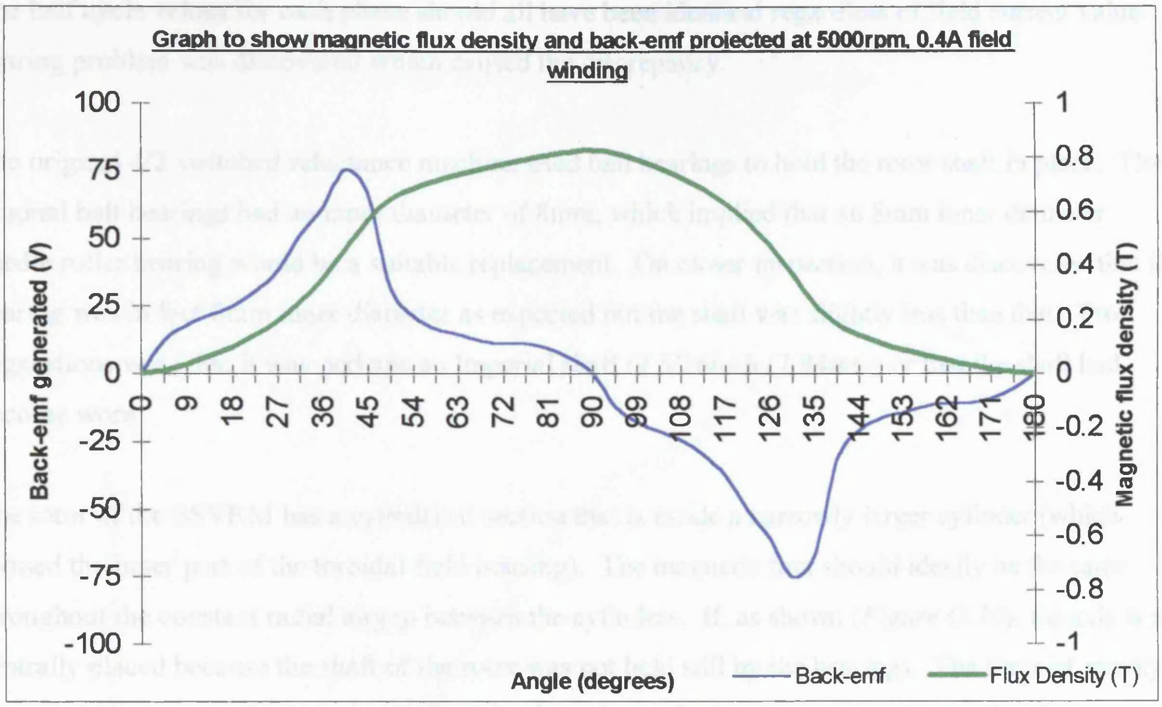


Figure G.15 :- 5000rpm back emf and flux density.

The waveform seen in one phase was in antiphase to the other (π radians difference). For each half cycle, the mean generated voltage was noted and the time for that half cycle noted. The product of these two values equalled the magnetic flux linkage for that half cycle. The value of generated voltage (which represented the back-emf) was positive when that phase would be switched on if it were to be run as a machine (and hence negative when it would be switched off). The zero volt cross-over of the back emf curve occurred when the rotor was at the aligned position. The opto sensor was positioned to change state at this point.

The field winding current is responsible for production of all the magnetic flux within the machine during back emf testing. The back-emf generated in each phase was measured for 50mA increments in the field current (the field had 1600 turns, so current multiplied by turn number gave the field ampere-turns). It was assumed that, for a fixed speed, based upon Faraday's Law, as the field current increased to provide more magnetic flux, the back-emf generated would increase linearly until saturation occurred.

Initial testing with the SSVRM produced some interesting results most significantly at higher field currents (*Figure G.17*). The modulus of back-emf for each half cycle of each phase was plotted as graphs of peak and mean generated back emfs against field current and as magnetic flux linkage against field current. It was noted that, as the field current increased from zero, the general trend between phases was similar. But as the current increased further the half cycle values differed within each phase but the sums of the half cycles for each phase matched very closely that of the other phase.

The half cycle values for each phase should all have been identical regardless of field current value. A bearing problem was discovered which caused the discrepancy.

The original 4/2 switched reluctance machine used ball bearings to hold the rotor shaft in place. The original ball bearings had an inner diameter of 8mm, which implied that an 8mm inner diameter needle roller bearing would be a suitable replacement. On closer inspection, it was discovered that the bearing was in fact 8mm inner diameter as expected but the shaft was slightly less than that. Two suggestions were that it was perhaps an Imperial shaft of 5/16inch (7.94mm) or that the shaft had become worn.

The rotor of the SSVRM has a cylindrical section that is inside a narrowly larger cylinder (which formed the inner part of the toroidal field housing). The magnetic flux should ideally be the same throughout the constant radial airgap between the cylinders. If, as shown (*Figure G.16*), the axis is not centrally placed because the shaft of the rotor was not held still by the bearings. The force of gravity acts upon the rotor as the rotor is not fixed in place due to the bearing mismatch. As the phase armature windings are energised in sequence, the magnetic fluxes from each phase attempt to align the rotor with the relevant stator poles. The magnetic forces are affected by gravity and the rotor does not remain in a fixed axial plane. The planes of the stator poles affect the positions of the rotor axis as the rotor turns. The unwanted axial plane movement of the rotor axis causes additional unwanted acoustic noise that increases with speed. If the cylindrical part of the rotor were loose enough to be in contact with the field housing then friction would provide additional noise and heating effects. This was possible as the radial airgap was very small (0.2mm).

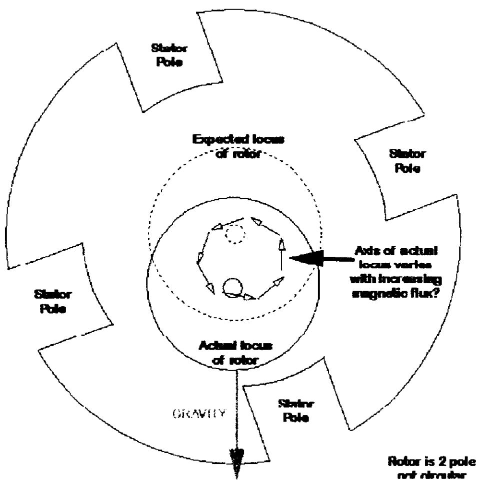


Figure G.16 :- Explanation of mismatch in back emf curves.

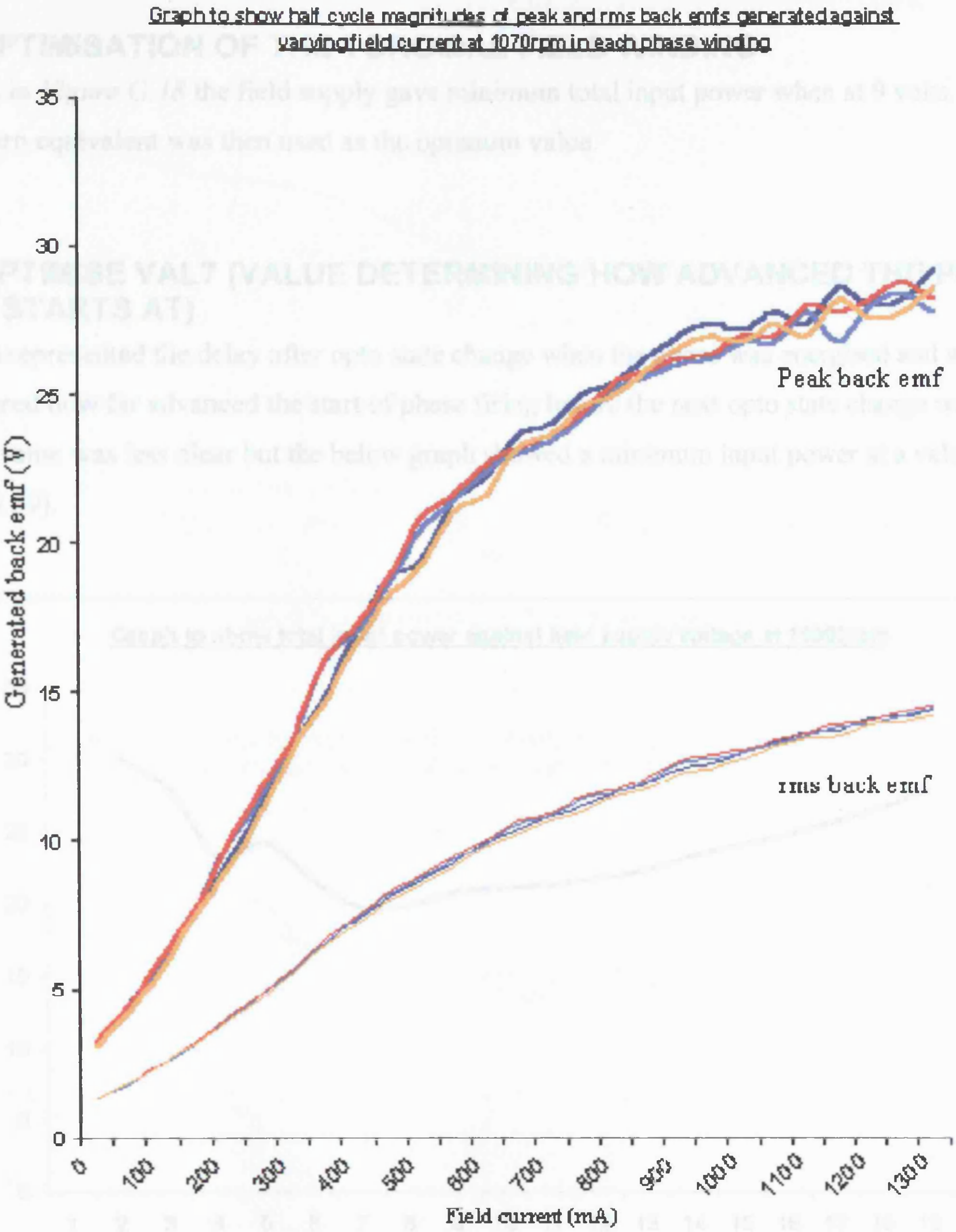


Figure G.17 :- Back emf test results at 1070rpm.

The small (suggested to be 0.06mm) shaft discrepancy was enough to cause the strange results (*Figure G.17*), where the magnitudes of negative and positve half cycle peaks of the back emf traces varied more with increasing flux (increasing field current) but the average magnitudes for each phase were similar. As the positive cycle increased in magnitude, the negative cycle decreased and vice versa, for each phase. This was due to the bearing problem. The solution was to re-machine the shaft down to 7mm which meant that a ball bearing replaced the needle roller bearing to hold the rotor in a fixed, centred, axial plane.

G.3 NO LOAD TESTING AT 10000RPM IN HIGH SPEED MODE

G.3.1 OPTIMISATION OF THE TOROIDAL FIELD WINDING

As shown in *Figure G.18* the field supply gave minimum total input power when at 9 volts. The ampere-turn equivalent was then used as the optimum value.

G.3.2 OPTIMISE VAL7 (VALUE DETERMINING HOW ADVANCED THE PHASE FIRING STARTS AT)

The value represented the delay after opto state change when the phase was energised and was such that it altered how far advanced the start of phase firing before the next opto state change was. The optimum value was less clear but the below graph showed a minimum input power at a value of 38 (*Figure G.19*).

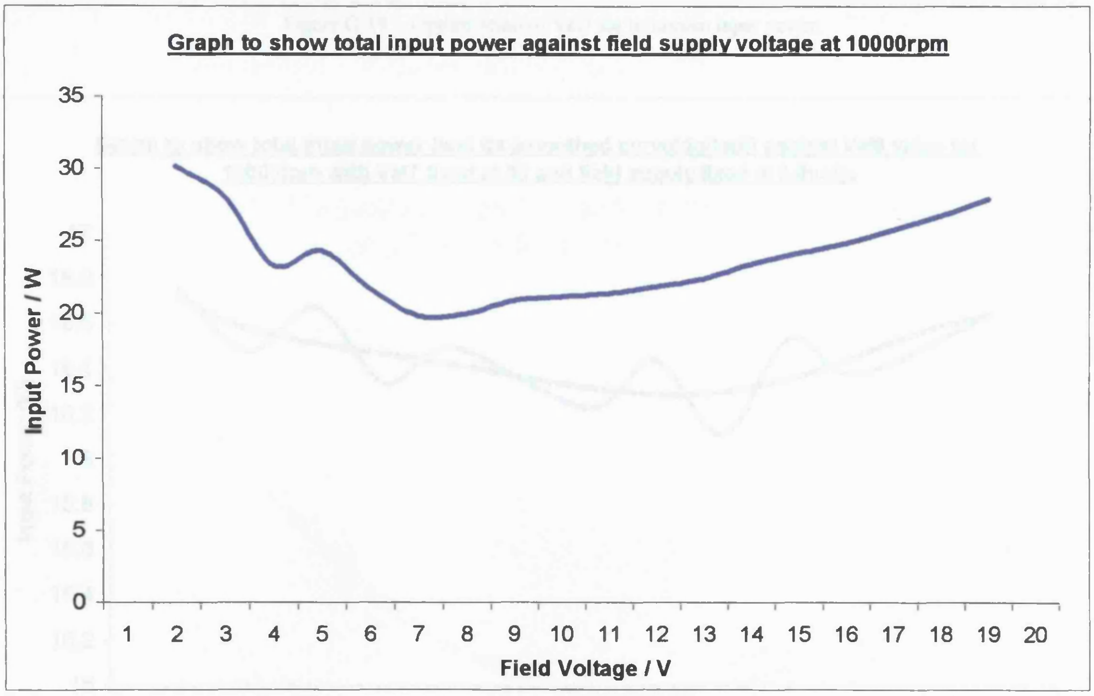


Figure G.18 :- Optimisation of field supply for minimum input power.

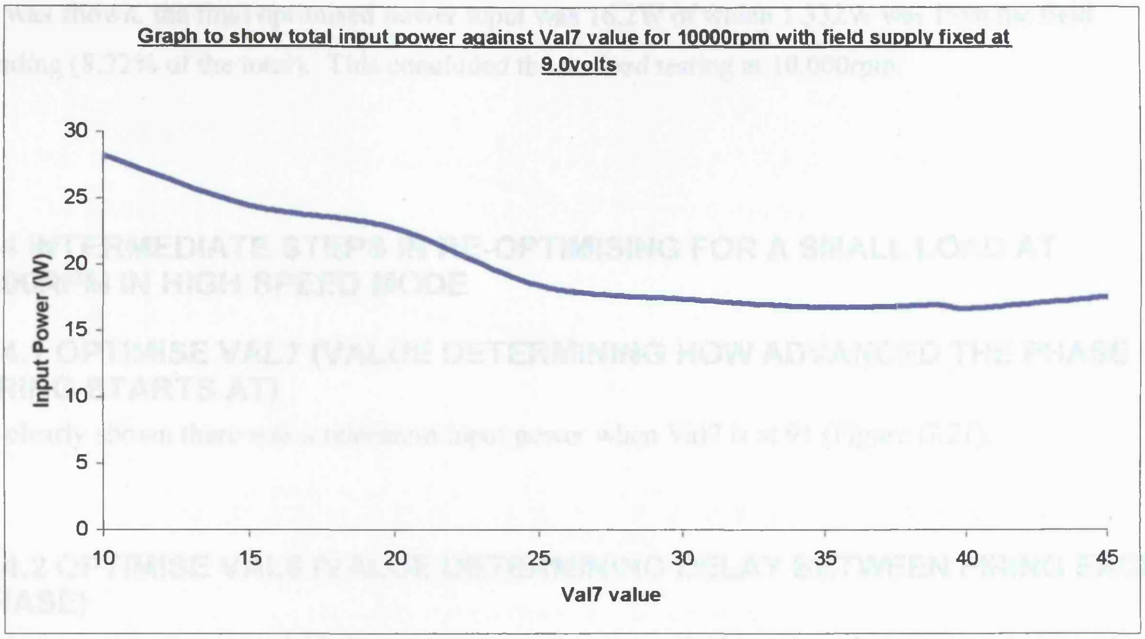


Figure G.19 :- Optimisation of Val7 for minimum input power.

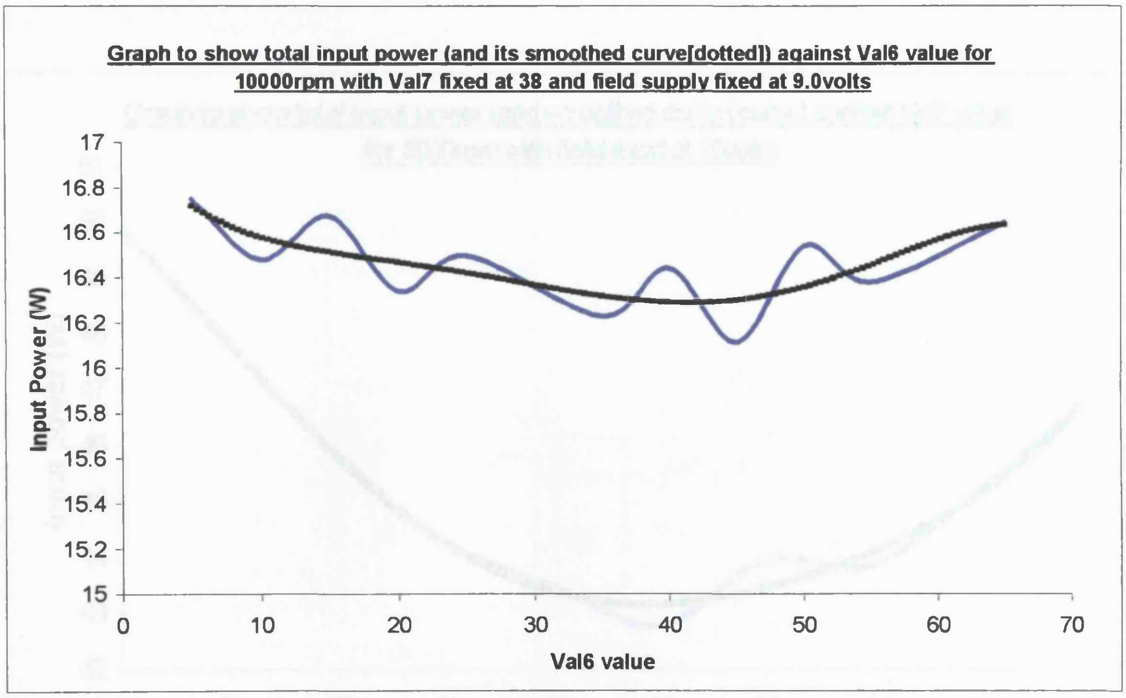


Figure G.20 :- Optimisation of Val6 for minimum input power.

G.3.3 OPTIMISE VAL6 (VALUE DETERMINING DELAY BETWEEN FIRING EACH PHASE)

This variable determines the size of time delay between energising each phase. The optimum value was not a clear choice from the results but a smoothed curve parabola for the best fit indicated that minimum power occurs at a value of 40 (*Figure G.20*).

As was shown, the final optimised power input was 16.2W of which 1.332W was from the field winding (8.22% of the total). This concluded the no-load testing at 10,000rpm.

G.4 INTERMEDIATE STEPS IN RE-OPTIMISING FOR A SMALL LOAD AT 5000RPM IN HIGH SPEED MODE

G.4.1 OPTIMISE VAL7 (VALUE DETERMINING HOW ADVANCED THE PHASE FIRING STARTS AT)

As clearly shown there was a minimum input power when Val7 is at 91 (*Figure G.21*).

G.4.2 OPTIMISE VAL6 (VALUE DETERMINING DELAY BETWEEN FIRING EACH PHASE)

As happened in the no load 5000rpm optimisation process, Val6 was less obvious to optimise and a value of 115 was regarded to be a preferred value (*Figure G.22*).

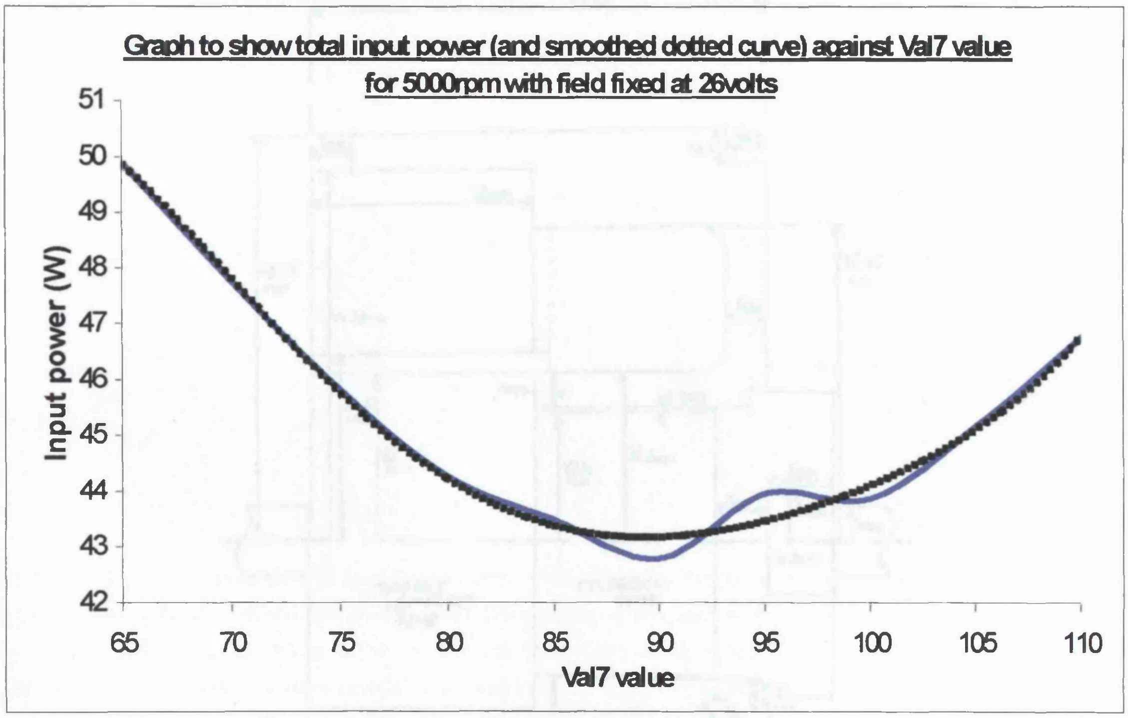


Figure G.21 :- Optimisation of Val7 for minimum input power.

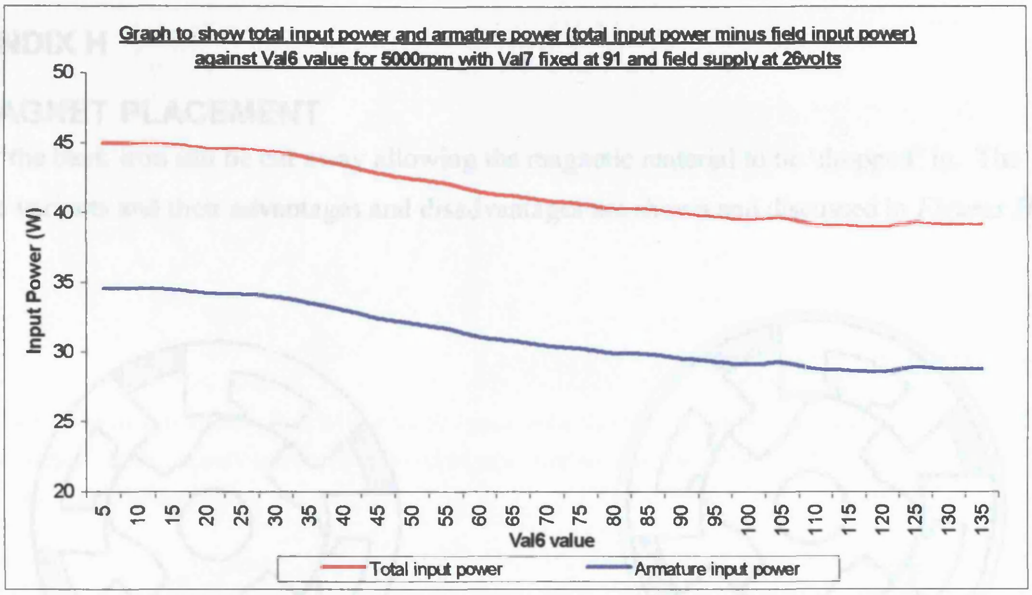


Figure G.22 :- Optimisation of Val6 for minimum input power.

G.5 SSVRM DIMENSIONS

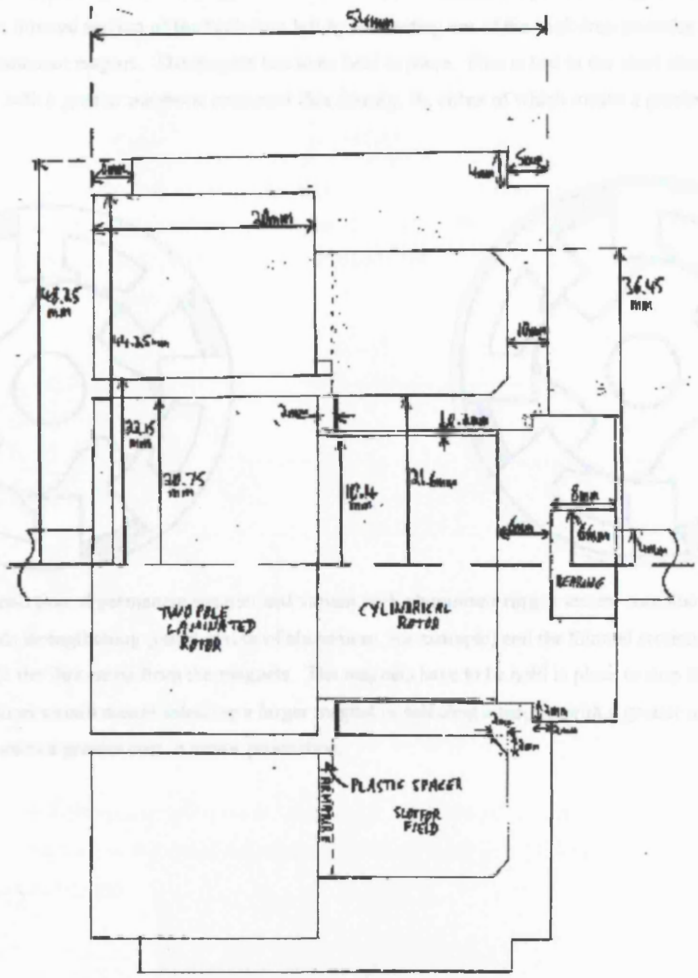
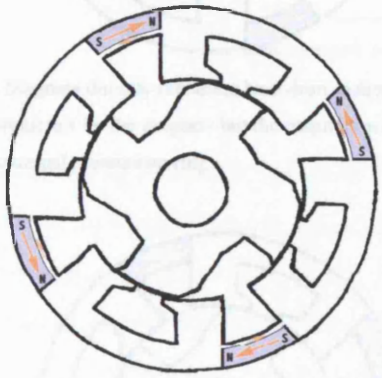


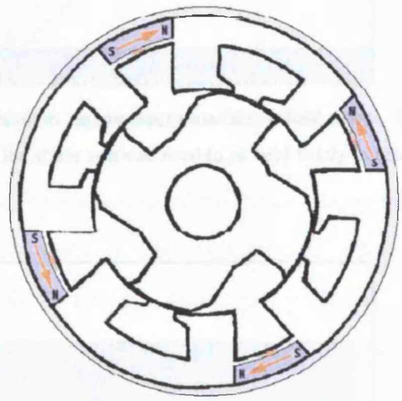
Figure G.23 :- Optimisation of Val6 for minimum input power.

APPENDIX H

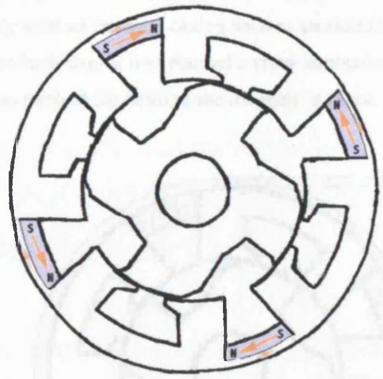
H.1 MAGNET PLACEMENT

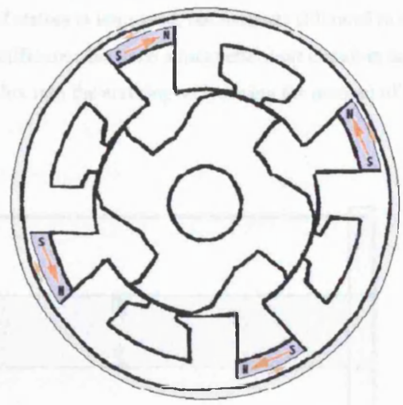
Parts of the back-iron can be cut away allowing the magnetic material to be 'dropped' in. The possible variants and their advantages and disadvantages are shown and discussed in *Figures H.1* to *H.6.*





Figures H.1 and H.2:- Outer insertion of permanent magnet and variant with aluminium ring to hold magnets in place. This potentially weakens the back-iron and the thinned section of the back-iron left by the cutting out of the back-iron provides a magnetic short-circuit path for the flux paths from the permanent magnet. The magnet has to be held in place. Flux is lost in the short circuit meaning selecting a larger magnet or selecting a magnet with a greater magnetic remanent flux density, B_r, either of which means a greater cost in motor production.





Figures H.3 and H.4:- Inner insertion of permanent magnet and variant with aluminium ring to ensure machine held stably. The potentially weak back-iron probably needs strengthening (outer casing of aluminium, for example) and the thinned sections by the magnets provides a magnetic short-circuit path for the flux paths from the magnets. The magnets have to be held in place to stop falling into the middle of the machine. Flux is lost in the short circuit means selecting a larger magnet or selecting a magnet with a greater magnetic remanent flux density, B_r, either of which means a greater cost in motor production.

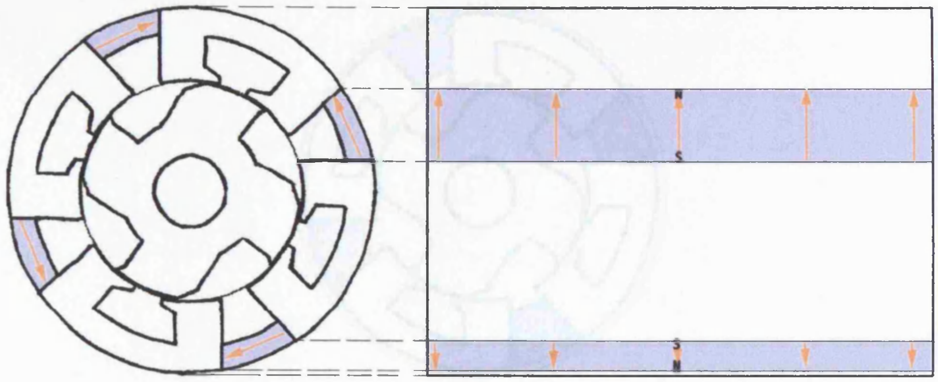


Figure H.5 :- Magnets directly replacing back-iron sections and variant with aluminium ring to ensure machine is held stably. This has no magnetic short circuit by the magnets but the magnets need holding in place and the stator sections need to be held firmly in place with, for example, an external aluminium ring.

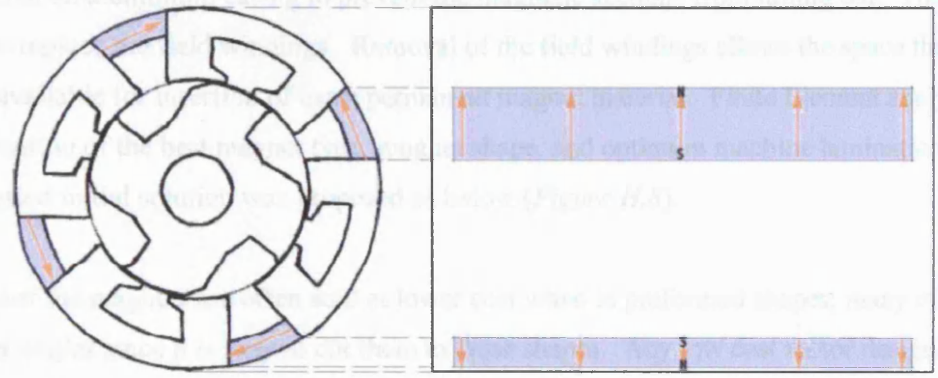


Figure H.6 :- Magnetic short circuit at axial ends of motor. Structural integrity of stators is improved, but magnets still need to be held in place, possibly with an external casing such as an aluminium casing. The main difference between a magnetic short circuit at the end relative to being in the back-iron is that magnetic cross lamination flux is needed to get flux into the end ring so reducing the amount of short circuit and there is no method for holding the magnets in place.

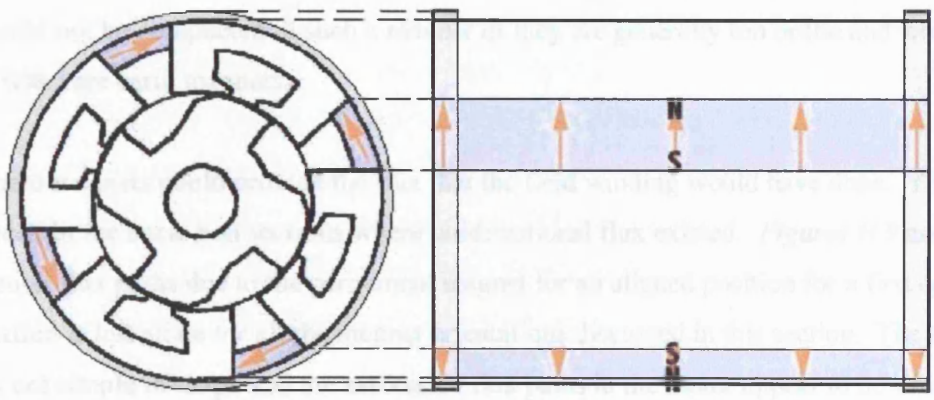


Figure H.7 :- Aluminium end caps holding magnets in place. A reduction in the amount of aluminium casing used to hold the motor together may be advantageous. This can be done by the use of conventional motor end cap (a gain from this would be reduced costs in the motor since less metal is needed in motor design).

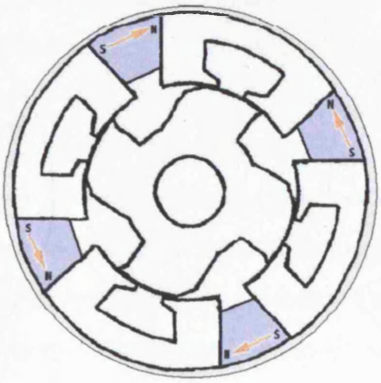


Figure H.8 :- More permanent magnet space within an initial design.

The preferable design concept is that of no short-circuit sections associated with the magnets and the possible use of an aluminium casing to prevent the magnetic sections from falling out. The permanent magnet is to replace the field windings. Removal of the field windings allows the space they occupied to become available for insertion of extra permanent magnet material. Finite Element analysis aids in the determination of the best magnet type, magnet shape, and optimum machine lamination shape. A simple untested initial solution was proposed as below (*Figure H.8*).

It is noted that the magnets are often sold at lower cost when in preformed shapes, many of which having right angles since it is easy to cut them to these shapes. Any low cost motor design should therefore follow this geometry (square or rectangular).

The stator sections could be compacted together to maximise the packing factor of the armature windings within the sections while leaving the back-iron mechanically strong. The permanent magnet materials could not be compacted in such a manner as they are generally too brittle and would shatter (especially with rare earth magnets).

The permanent magnets could provide the flux that the field winding would have done. The magnets must be placed in the back-iron sections where unidirectional flux existed. *Figures H.9* and *H.10* show the actual flux paths due to the permanent magnet for an aligned position for a first concept. The rotor lamination is left alone for all the magnet orientations discussed in this section. The steel stator laminations are simple in shape and the associated flux paths in the motor appear to be short and simple. Note the dimensions of the magnets and the fact that the magnitude of the magnet end face area is a guide to the magnitude of magnetic flux produced and that the magnet length (in the direction of flux inside the magnet) represents magnetomotive force (mmf) from the magnet. This orientation was used as a reference against the other possible magnet orientations. It was called 'original concept orientation'.

Figures H.9, H.10 :- Original design (called zip004) with flux directions shown. Easiest design. Shortest flux paths.

Figures H.11, H.12 :- Magnets at diagonal angle. More complex steel laminations, risk of unwanted areas of saturation (* in diagram), but longer meaning more flux due to larger end face area. Thinner magnet giving less mmf. Longer flux paths

Another possible orientation is shown by *Figures H.11, H.12*. The magnets are orientated at a diagonal angle. The end face area of each magnet is larger than those in *Figures H.9, H.10* which meant that there is more magnetic flux from the magnets but the magnet is thinner (when constraining the inner and outer radii of the stator sections to be constant in all cases) so less magnetomotive force is produced by the magnets. The positioning of the magnets presented the problems of magnetic saturation occurring in sections other than the rotor and stator poles (which are the only places where saturation is potentially wanted). These saturation problem areas are shown by the symbol * in *Figure H.11*. The magnetic flux paths are longer than those shown in *Figure H.10*. The steel stator laminations are more complex in shape than the original concept orientation. This design is not as desirable as that of *Figure H.10*.

The next possible orientation is that of magnets inserted at 90 degrees to that of the original concept orientation (*Figures H.13, H.14*). To prevent magnetic short circuits (a loss in useful magnetic flux) there are sections of air or aluminium inserted to guide the magnetic flux in the paths shown in *Figure H.13*. The steel stator laminations are complex in shape and there is great potential for unwanted magnetic saturation in the sections shown by the symbol * in *Figure H.13*. The magnetic flux paths are longer and more complicated and the magnet end face area is reduced with a potential reduction in magnet length. The effectiveness of the magnets is compromised so this orientation is not good.

The final orientation type is a new approach to this problem. *Figure H.15* shows the main shape of *Figure H.9* but with a few alterations. There are spacers (plastic or aluminium) where the magnets used to be. The magnets are now four sections making a rubber ring magnet which surround the original lamination. This is enclosed by a steel ring to provide the completion of the magnetic flux path circuit (*Figure H.16*). The sections of the rubber permanent magnet provide radial flux. The steel ring outside this provides completion of the flux paths for the magnets. The bi-directional path sections in the inner back-iron aid directing the flux from the magnets into the stator poles (bi-directional due to armature winding energisation). The spacers have to be non-magnetic to avoid magnetic short circuits and, as such, would be plastic or aluminium. Solid spacers hold the laminations in place. Air spacers make holding the machine innards in place more difficult as the laminations have to be glued in place. Generally, this orientation type would make the design more complex as well as increasing the overall size. Very long and complex flux paths exist and it should be noted that the motor is the largest of all the concepts introduced.

Figure H.13, H.14 :- Magnets flux rotated 90 degrees. Shaded areas to be air or aluminium. More complex steel laminations, shaded areas of air or aluminium to reduce flux leakage paths, risk of saturation in sections other than stator poles (surrounding the magnets) (* in diagram), magnet size is reduced. Very long flux paths.

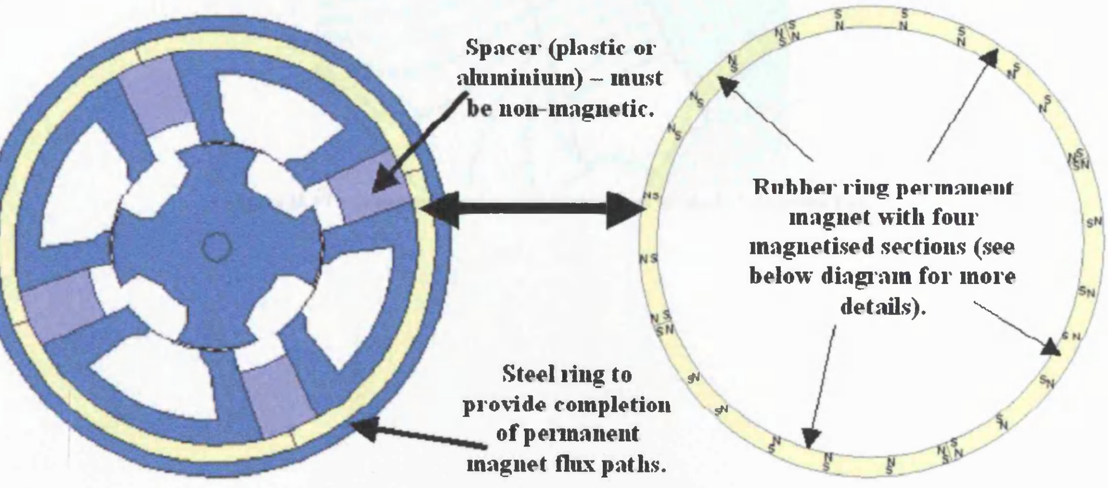


Figure H.15 :- Layout of a rubber ring permanent magnet.

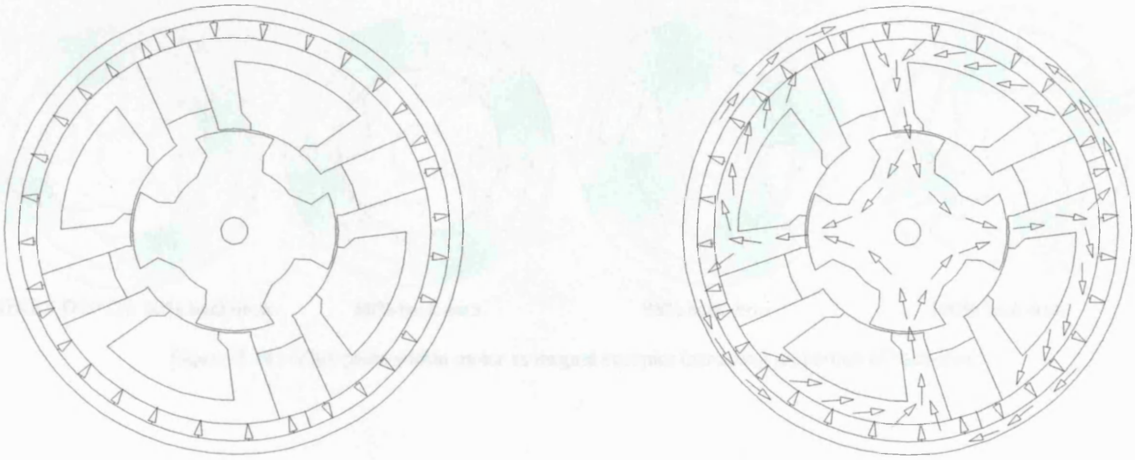


Figure H.16 :- Magnetic flux from a rubber ring permanent magnet.

Chapter 6 summarises this investigation into magnet placement topologies.

H.2 SHORT-CIRCUIT FLUX PATHS WHEN MAINTIANING STATOR STRUCTURAL INTEGRITY

As shown in *Figures H.17* and *H.18*, there is lost torque producing flux when the back-iron behind the magnets is maintained as flux is lost in the short circuit.. For small machines, a narrower short-circuit loses stator structural integrity due to being too thin and hence it is better to have no short circuit and then to address methods for holding the magnets and stator sections together.

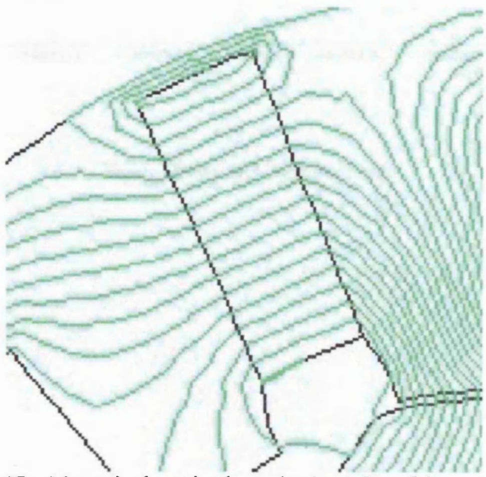
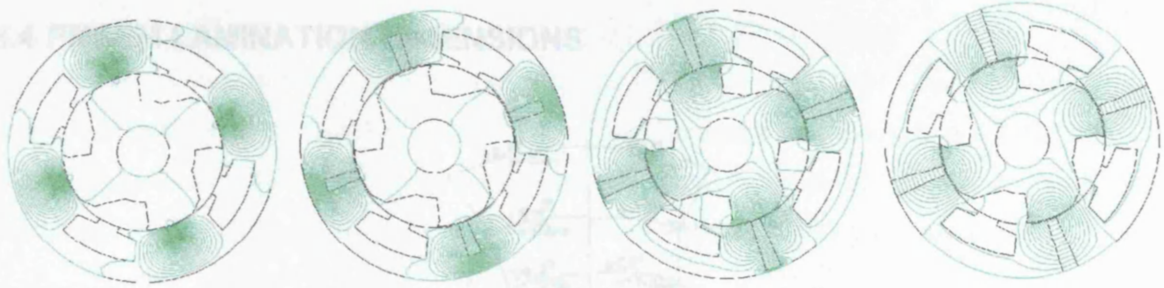


Figure H.17 :- Magnetic short circuit causing loss of 'useful' magnetic flux.



MAGNET DEPTH: 30% back-iron 60% back-iron 95% back-iron 100% back-iron

Figure H.18 :- Flux paths within motor as magnet occupies increasing proportion of back-iron.

H.3 EXAMPLE B-H CURVES

B-H curves were for Fer2 (*Figure H.19*) and N48 (*Figure H.20*).

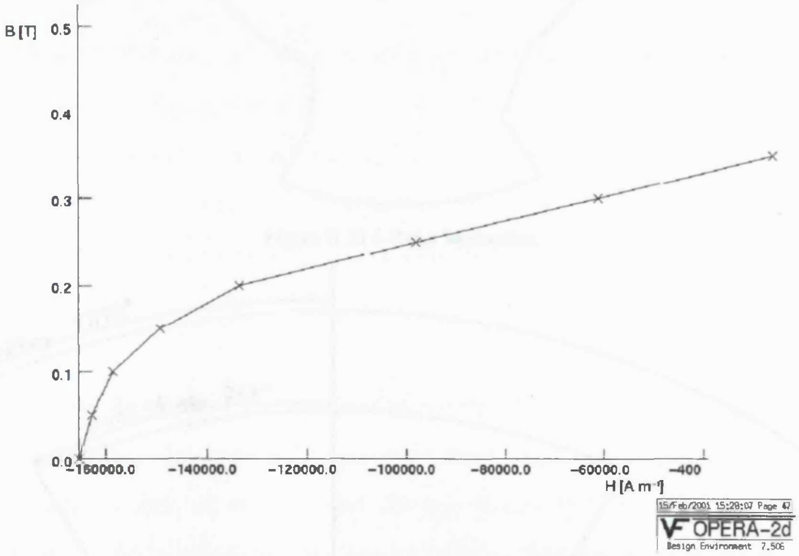


Figure H.19 :- BH curve for Fer2.

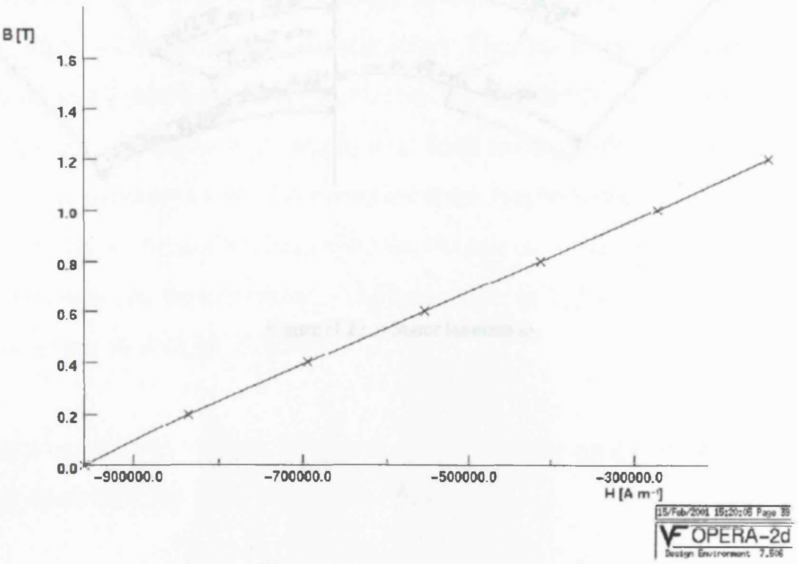


Figure H.20 :- BH curve for N48.

H.4 PMFSM LAMINATION DIMENSIONS

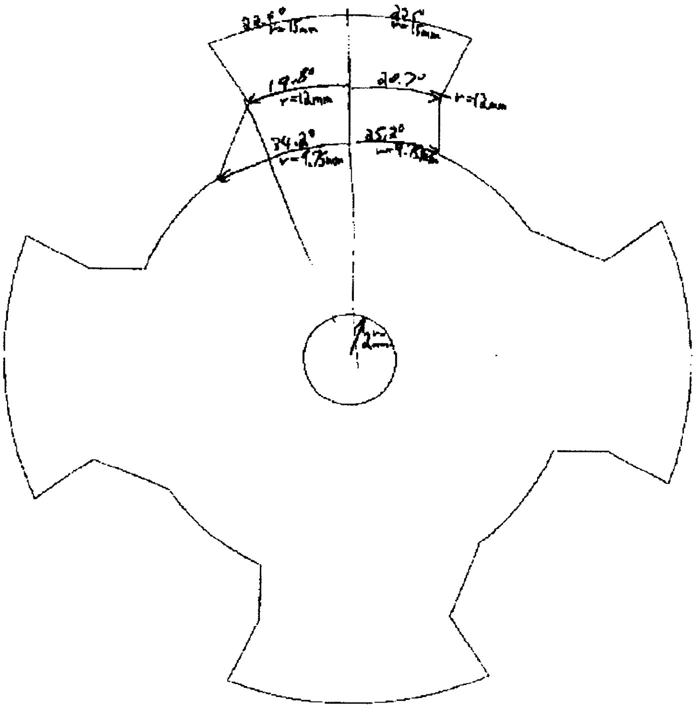


Figure H.21 :- Rotor lamination.

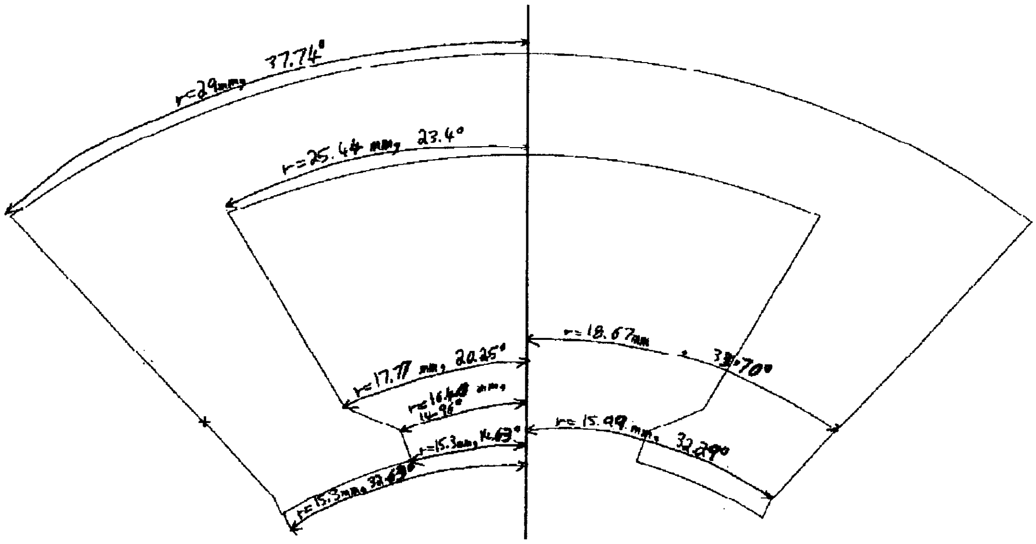


Figure H.22 :- Stator lamination.

H.5 SPREADSHEET FOR DYNAMIC MODELLING OF PMFSM

The spreadsheet used to model the Permanent Magnet Flux Switching Motor is based upon the spreadsheet developed by Professor Charles Pollock and his staff and students at the Centre for Advanced Electronically Controlled Drives, now based at the University of Leicester, to model Flux Switching Motors. This section of the Appendices compliments Chapter 6 of this thesis.

The spreadsheet has been written using Microsoft Excel. The spreadsheet used to model the Flux Switching Motor assumes that each phase, energised in sequence, will have permanent supply of armature voltage to each phase i.e. the model does not support a pulsed armature supply voltage. The model does allow the start of energisation of each phase to be varied in relation to rotor position. The spreadsheet uses iterative mathematic equations to allow a cycling of data to produce an optimum solution based upon the given data supplied (dynamic modelling).

Finite Element Analysis performs static analysis of the machine at different rotor positions – the rotor angles being selected to provide suitable data points for the spreadsheet. The spreadsheet is given data from the Finite Element Analysis – the data being based on rotor position defined by the number of rotor and stator poles. The Finite Element Analysis usually provides data in relation to flux from field windings, but it was able to measure field flux from the magnets used due to the way it calculated the flux.

Magnetic reluctance against rotor position is needed for the spreadsheet. As magnets are used in the PMFSM, the Finite Element Analysis model had to be modified. This requires the magnet material used in the Finite Element Analysis model to be 'demagnetised' whilst retaining its permeance (same shape of its B-H curve). The magnet sections can be redefined as having mathematically translated B-H curves such that the B-H curve is the same shape as that for the magnet but the magnet produces no flux, giving a B-H curve akin to a steel or similar alloy. The flux from small currents (e.g. 0.1A and 0.15A, with a known small number of armature turns, Na e.g. Na=10) is calculated using Finite Element Analysis for the 'demagnetised' machine at fixed rotor positions. The change in flux due to the change in current is measured from the Finite Element Analysis data. The change in flux multiplied by the number of turns and divided by the change in current gives a value for inductance from which the reluctance can be calculated. The reluctance at a given rotor position is suitably termed in the spreadsheet as RELUCTANCE.

The spreadsheet had been given its own Microsoft Excel cell reference definitions, based upon it being originally used for modelling the Flux Switching Motor.

The maximum angle over which a phase should be energised for is defined as (*Figure H.23*) HALF_ELEC_ANGLE=360*((1/Nr)-(1/Ns)), where Nr is defined as the number of rotor poles and Ns is defined as the number of stator poles.

In the PMFSM spreadsheet model, 'ON' defines a negative angle value. This angle allows the angular position of determining when energisation of the phase may be started to be selected (as the Finite Element Analysis does not necessarily have zero degrees rotor position at the point where a back emf trace would be changing polarity at the zero crossing (this angle allows the Finite Element data and spreadsheet model to be matched). Adding the angular value of 'DIFF' to 'ON' defines the angle when to turn on the armature winding for that phase (*Figure H.23*). Adding the angular value of 'OFF' to 'ON' defines when to turn the phase off (*Figure H.23*). This is the basis for allowing the spreadsheet to model a pulsed supply.

Vs is defined as the armature supply voltage in the spreadsheet. When a phase is energised, it is given Vs as the supply and the armature current in that phase increases in magnitude. When the phase turns off, the voltage is –Vs due to freewheel diodes of the half bridge converter circuit and the current will start to fall. The spreadsheet calculates the current for the rotor positions and stops the current becoming negative in the following rotor position by forcing the current to zero for that following position if it is attempting to become negative, and also setting the supply voltage is to Vs=0 until the next phase is energised (the current is forced to zero until the next phase is energised. This requires a degree of complexity in the Microsoft Excel equations used.

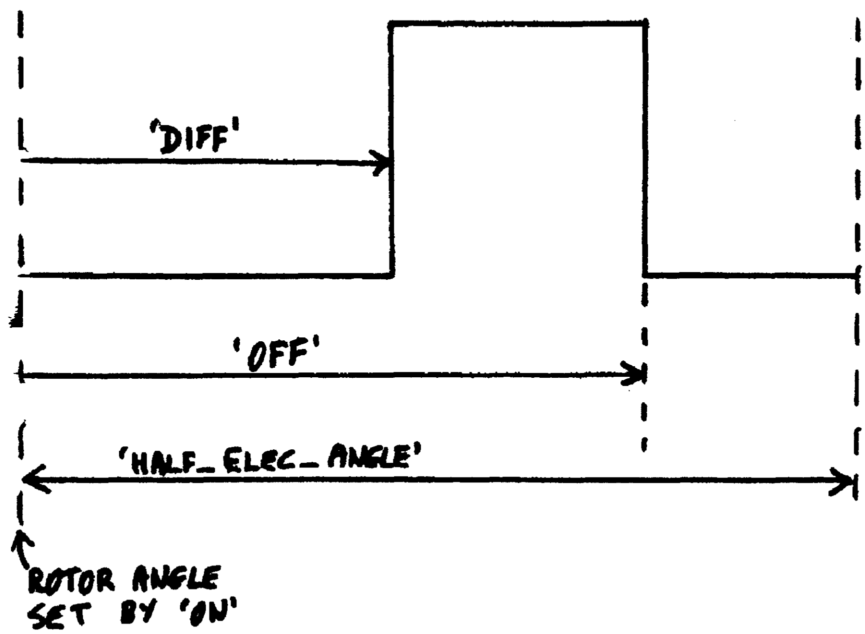


Figure H.23 :- defining how spreadsheet model determines turning on one phase using angles relative to rotor angle.

Figure 6.44 in Chapter 6 shows the main user interface of the spreadsheet used to model the PMFSM. The spreadsheet was adapted to show the air moved per minute such that the potential applications of the designed motor are instantly shown. The spreadsheet allows the rotor speed to be changed, the armature supply voltage to be varied, and the armature turn numbers and turn diameter to be changed.

Shown below is a list of main calculations added and amended in the spreadsheet for the first phase energisation (the second phase energisation requires minor alterations to the coding, such as changing '<' to '>', etc, so as to ensure the spreadsheet worked correctly. Also shown are some of the basic equations required to facilitate understanding the calculations. This list is incomplete due to the complexity of the spreadsheet. The spreadsheet uses iterative calculations to determine an optimum solution for the given parameters and supplied data, by feeding back answers back into the equations.

BASIC EQUATIONS:-

INSTANTANEOUS_POWER =MOTIONAL_EMF*ARMATURE_CURRENT

TOTAL_ARMATURE_MMF=Na*ARMATURE_CURRENT

TIME_FOR_CHANGE_IN_ANGLE=(ROTOR_ANGLE-PREVIOUS_ROTOR_ANGLE)/(SPEED*6)

MOTIONAL_EMF=INDUCED_ARMATURE_VOLTS_PER_TURN*Na*BACK_EMF_CORRECTION

AVERAGED_MOTIONAL_EMF_CD=(MOTIONAL_EMF_AT_PREVIOUS_ANGLE_STEP+MOTIONAL_EMF+MOTIONAL_EMF_AT_NEXT_ANGLE_STEP)/3

ROTOR_ANGLE_POSITION = *determined from rotor and stator pole numbers and from choice of number of rotor positions used in Finite Element Analysis*

ADAPTED EQUATIONS:-

ARMATURE_1_SWITCH_LOGIC=IF(AND(ROTOR_ANGLE_POSITION>=(ON+DIFF+(360/2/Nr)),ROTOR_ANGLE_POSITION<(OFF+(360/2/Nr))),1,0)

ARMATURE_2_SWITCH_LOGIC=IF(OR(AND(ROTOR_ANGLE_POSITION>=ON+DIFF,ROTOR_ANGLE_POSITION<OFF),AND(ROTOR_ANGLE_POSITION>=ON+DIFF+(360/Nr),ROTOR_ANGLE_POSITION<OFF+(360/Nr))),1,0)

VOLTAGE_POLARITY=ARMATURE_2_SWITCH_LOGIC-ARMATURE_1_SWITCH_LOGIC

APPLIED_VOLTAGE=IF(ABS(VOLTAGE_POLARITY)=0,IF(PREVIOUS_ROTOR_POSITION_CURRENT>0,-1*Vs,0),(Vs)*VOLTAGE_POLARITY)

ARMATURE_CURRENT=IF((PREVIOUS_ROTOR_POSITION_CURRENT+((APPLIED_VOLTAGE-(PREVIOUS_ROTOR_POSITION_CURRENT*Ra)-AVERAGED_MOTIONAL_EMF_CD)*TIME_FOR_CHANGE_IN_ANGLE/Na/Na*RELUCTANCE)<0),0,PREVIOUS_ROTOR_POSITION_CURRENT+((APPLIED_VOLTAGE-(PREVIOUS_ROTOR_POSITION_CURRENT*Ra)-AVERAGED_MOTIONAL_EMF_CD)*TIME_FOR_CHANGE_IN_ANGLE/Na/Na*RELUCTANCE))

ARMATURE_CURRENT_SQUARED =ARMATURE_CURRENT*ARMATURE_CURRENT

ARMATURE_1_CURRENT =IF((IF(((IF(AND(ROTOR_ANGLE_POSITION>=(ON+(360/2/Nr)), ROTOR_ANGLE_POSITION<(OFF+(360/2/Nr))),1,0))+ARMATURE_1_SWITCH_LOGIC+ARMATURE_2_SWITCH_LOGIC)=1,1,0)) =1,ARMATURE_CURRENT,0)

ARMATURE_2_CURRENT =IF((IF(((IF(AND(ROTOR_ANGLE_POSITION>=(ON+(360/2/Nr)), ROTOR_ANGLE_POSITION<(OFF+(360/2/Nr))),1,0))+ARMATURE_1_SWITCH_LOGIC+ARMATURE_2_SWITCH_LOGIC)=1,1,0)) =0,-1*ARMATURE_CURRENT,0)

H.6 POWDERED IRON

To possibly lower the iron losses in the motor the steel laminations could be replaced with powdered iron (*Figures H.24, H.25*). The rotor may not always be manufactured with powdered iron due to the possibility of the material falling apart at higher speeds due to rotational forces (there would have to be an upper rotor speed limit depending on the rotor shape and size). There are three grades based upon the densities of a powdered iron called Somaloy500 (6.97g/cm^3 , 7.23g/cm^3, 7.31g/cm^3) (made by Hoganas in Sweden). The densities vary due to differences in the compaction process used. The B-H curves for Somaloy500 (a powdered iron) (*Figures H.26, H.27, H.28*) confirm that any motor using this material requires more armature ampere turns to derive the same total flux within the motor when compared to steel. The reduction in permeability in flux is due to the iron powder particles in Somaloy500 being coated in insulating material and also containing compounds used for bonding resulting in less flux for a given magnetic field. This reduction in permeability requires an increase in armature current to provide the same flux level which in turn increases the copper losses. There is a requirement to find the trade off point where the reduction in iron losses is more than the increase in copper losses. The difference between a steel lamination design and same lamination shape designs incorporating Somaloy500 in the stator only or stator plus rotor are shown by graphs of armature flux versus rotor position (*Figures H.28, H.29, H.30*). These graphs compare (using all three grades of Somaloy500) an all steel lamination design (that has been built i.e. the PMFSM), a laminated steel rotor plus Somaloy500 stators design, and a Somaloy500 rotor plus stator design.

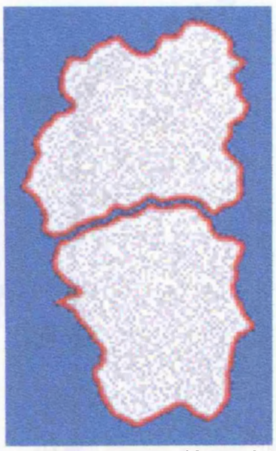


Figure H.24 :- powdered iron microscopically

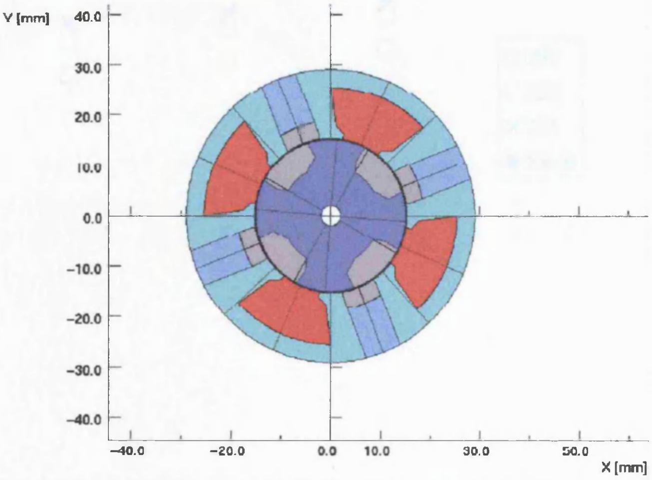


Figure H.25 :- Somaloy500 (stator) and steel (rotor) FE model of Zip004 design

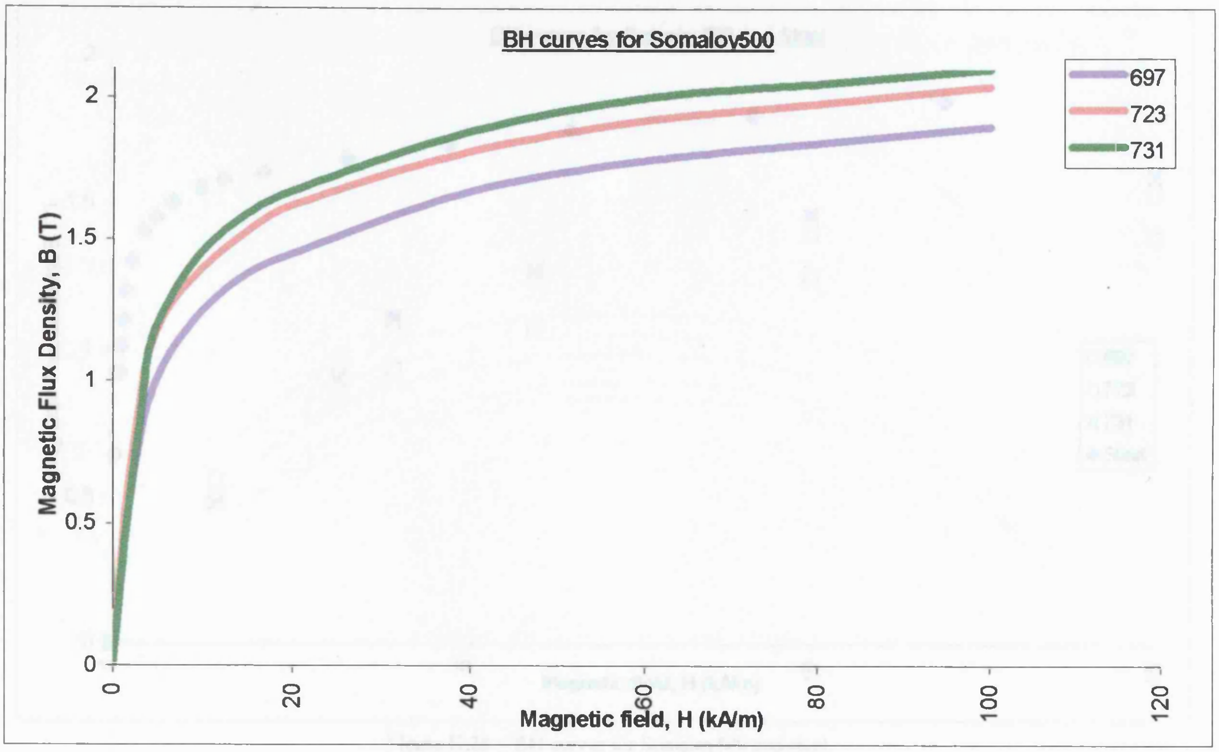


Figure H.26 :- BH curves for Somaloy500.

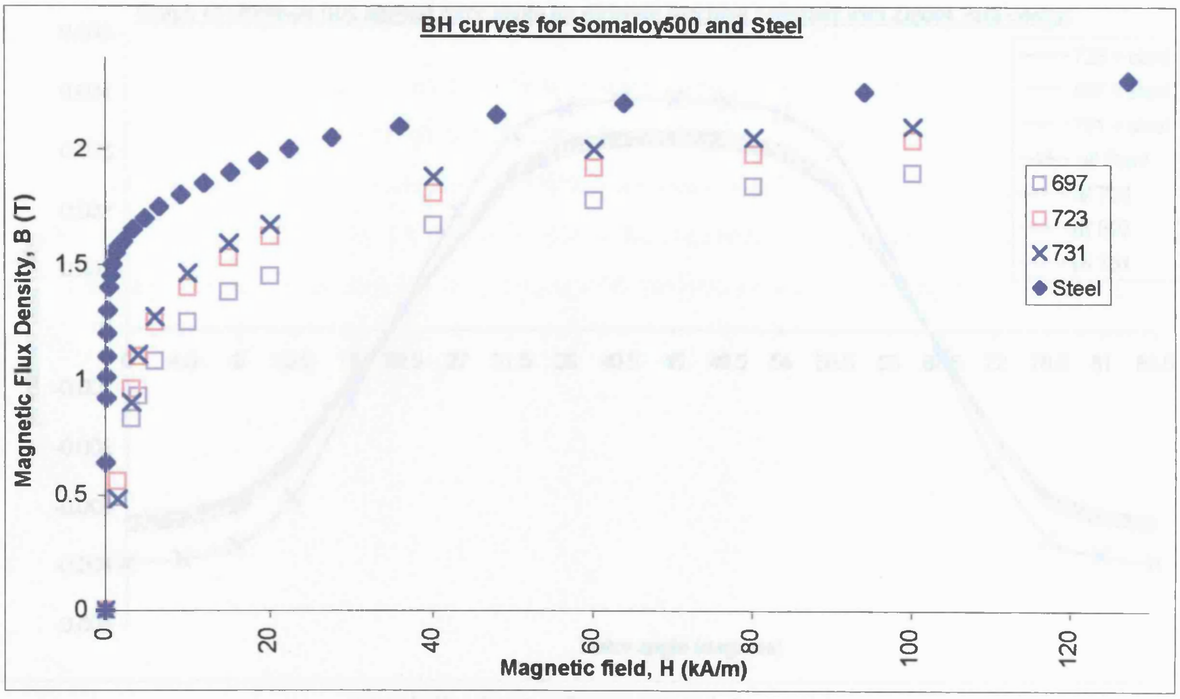


Figure H.27 :- BH curves for Somaloy500 and steel.

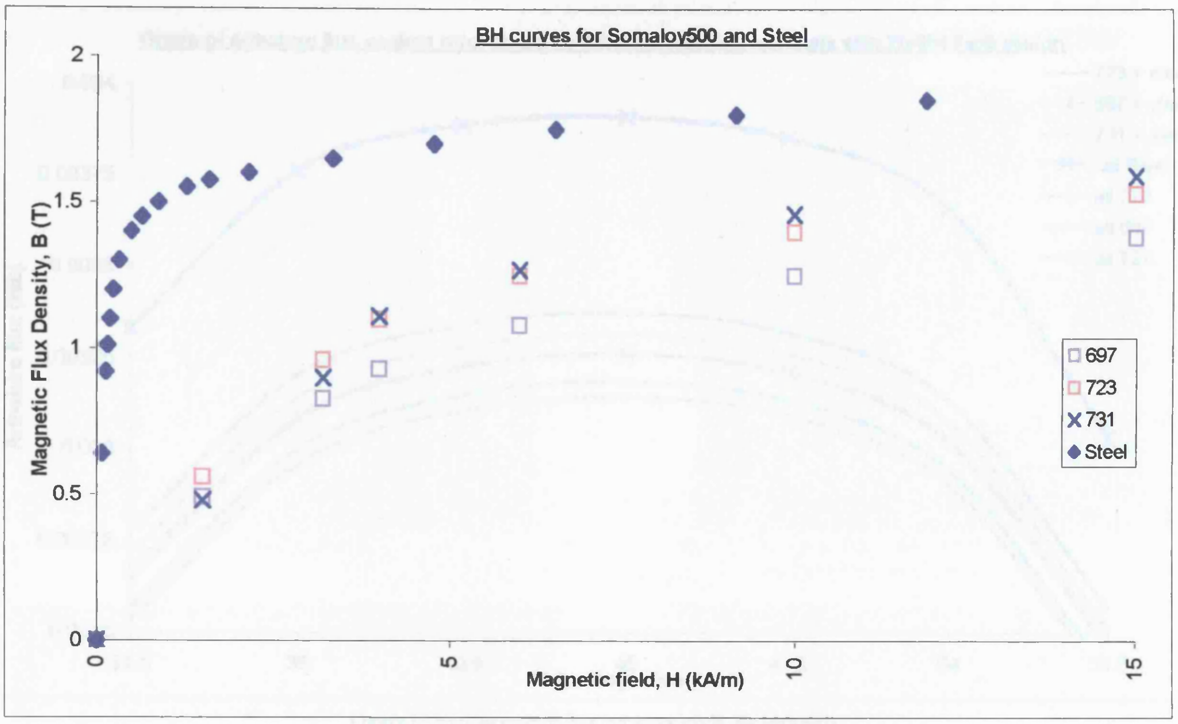


Figure H.28 :- BH curves for Somaloy500 and steel.

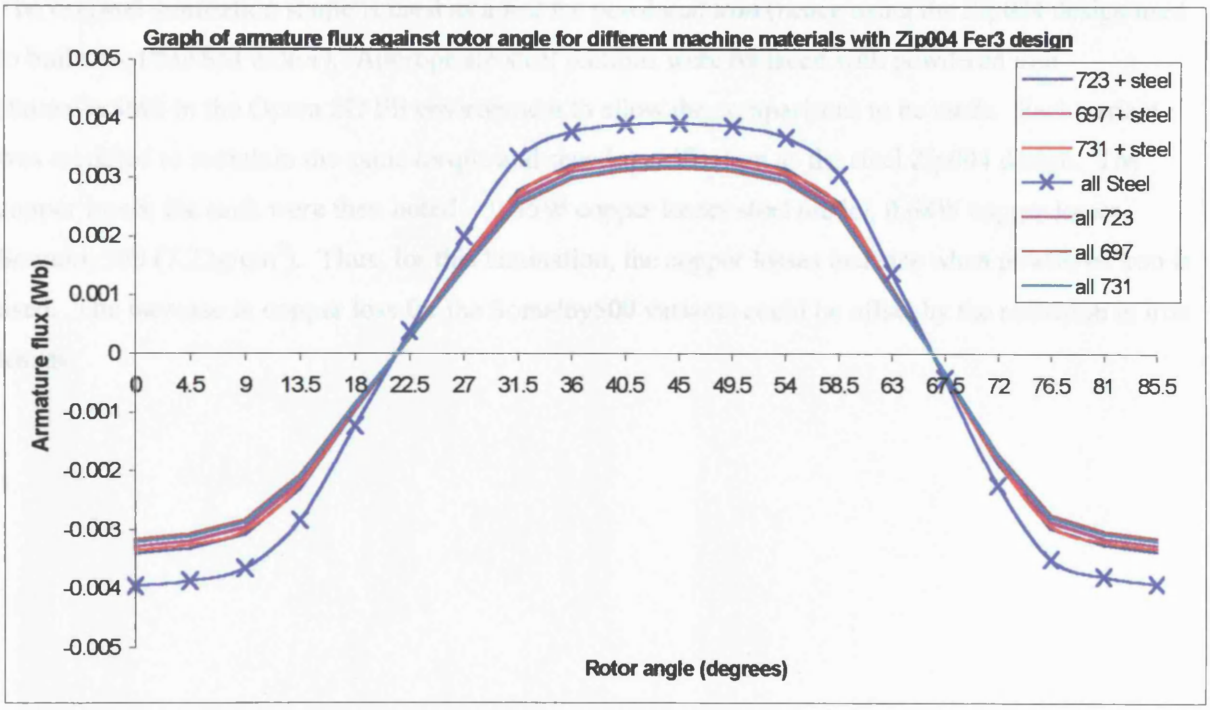


Figure H.29 :- Armature flux per rotor cycle for materials.

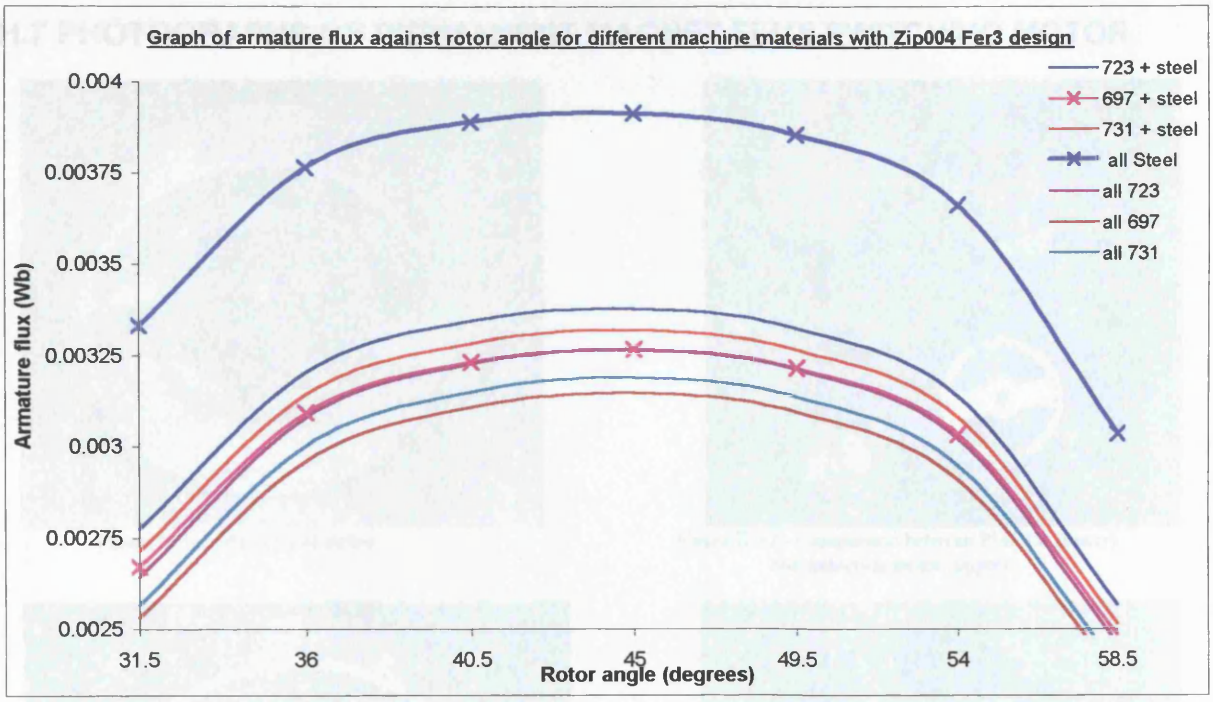


Figure H.30 :- Armature flux per rotor cycle for materials.

The original lamination shape is used as a test for powdered iron (hence using the Zip004 design used to build the PMFSM motor). Appropriate steel sections were replaced with powdered iron (Somaloy500) in the Opera 2D FE environment to allow the comparisons to be made. Each variant was modeled to maintain the same torque and speed specification as the steel Zip004 design. The copper losses for each were then noted - 0.45W copper losses steel model, 0.68W copper losses Somaloy500 (7.23g/cm^3). Thus, for this lamination, the copper losses increase when powdered iron is used. The increase in copper loss for the Somaloy500 variants could be offset by the reduction in iron losses.

H.7 PHOTOGRAPHS OF PERMANENT MAGNET FLUX SWITCHING MOTOR

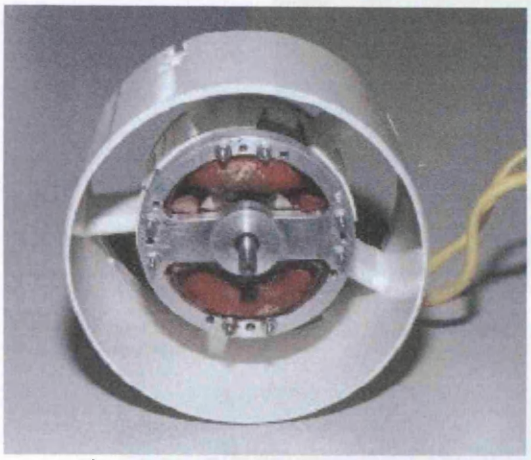


Figure H.31 :- Fan end of motor.

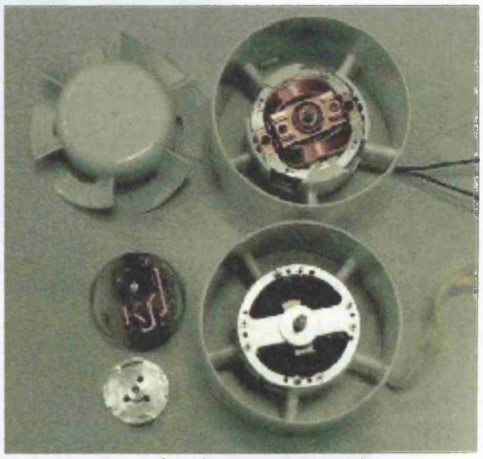


Figure H.32:- Comparison between PMFSM (lower) and induction motor (upper).

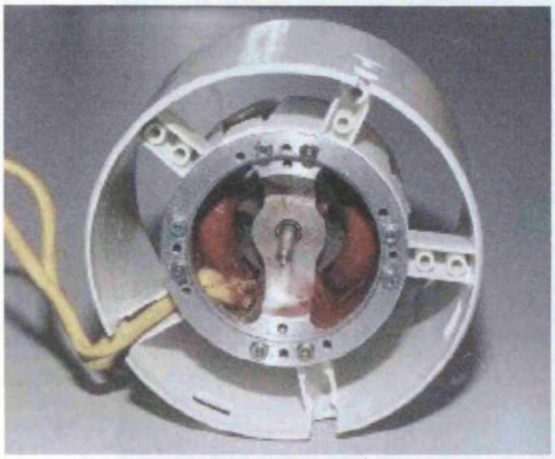


Figure H.33 :- Optosensor end of motor.

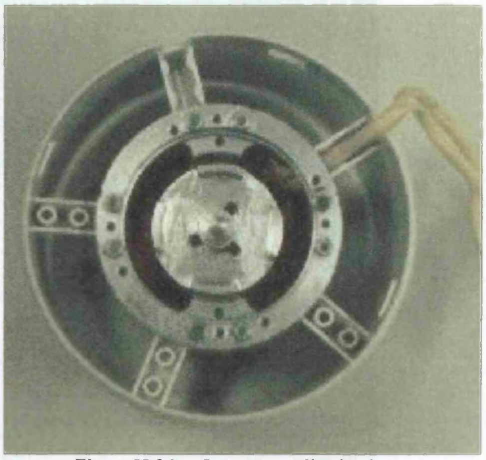


Figure H.34 :- Optosensor disc in place.

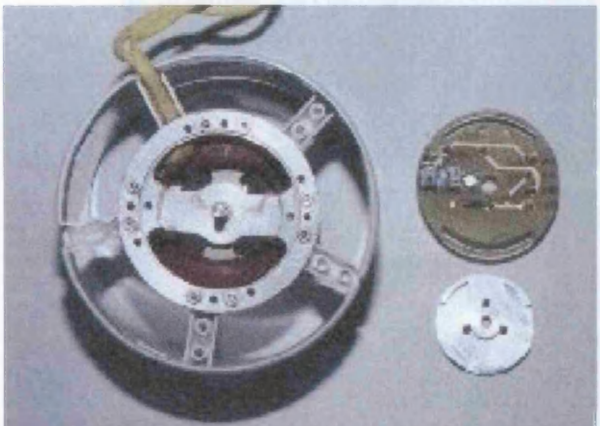


Figure H.35 :- Optosensor end of motor in parts.

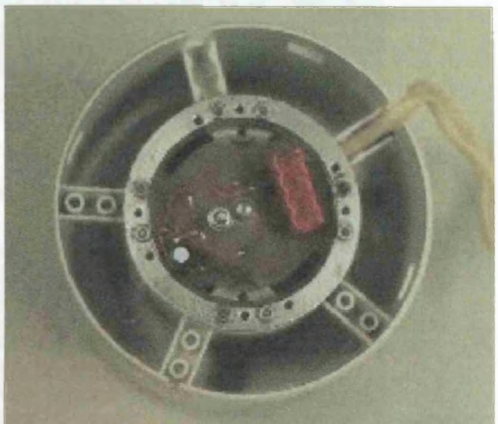


Figure H.36 :- Optosensor end of motor assembled.

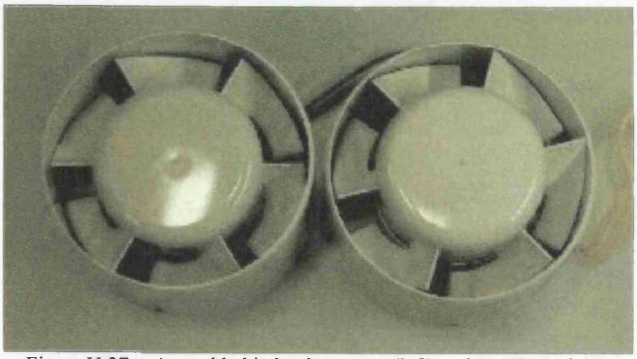


Figure H.37 :- Assembled induction motor (left) and PMFSM (right).

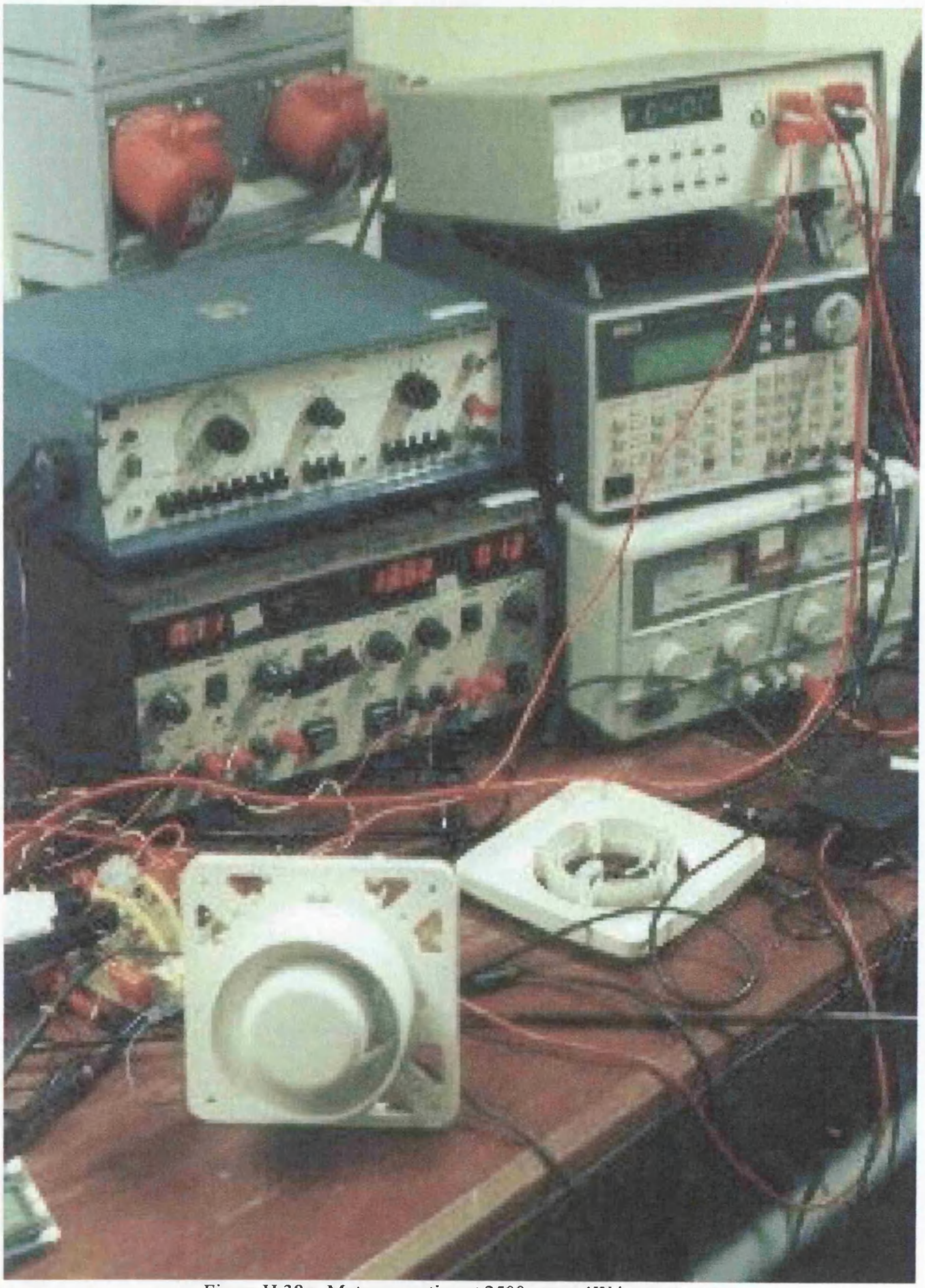


Figure H.38 :- Motor operating at 2500rpm at 4W input power.