University of Leicester

QinetiQ

The Systems Engineering Doctorate Centre

Continuous Local Motion Planning & Control for Unmanned Vehicle Operation within Complex Obstacle-Rich Environments

Thesis submitted to the University of Leicester

for the degree of

Doctor of Engineering

By Andrew J Berry[1]

QinetiQ Research Scientist & EngD Student at the University of Leicester & the Systems Engineering Doctorate Centre[2]

August 2010

Supervisors: Ian Postlethwaite[3], Da-Wei Gu[4] & Jeremy Howitt[5]

[1] QinetiQ, Cody Technology Park, Ively Rd, Farnborough, Hampshire, GU14 0LX, UK, ajberry@qinetiq.com

[2] Department of Electronic & Electrical Engineering, Loughborough University, Leicestershire, LE11 3TU

[3] Professor and Deputy Vice-Chancellor, Northumbria University

[4] Professor, Control & Instrumentation Research Group, Department of Engineering, University of Leicester

[5] Assistant Technical Director, Aerospace, QinetiQ

Abstract

Continuous Local Motion Planning & Control for Unmanned Vehicle Operation within Complex Obstacle-Rich Environments

Andrew J. Berry[6]

This thesis considers the guidance and control of unmanned vehicles within complex environments. A systems engineering approach was adopted where significant effort was directed towards defining a high level capability requirement and subsequent problem exploration, decomposition and definition, prior to addressing the technical focus. The goal of this approach was to ensure that technical work was directed towards realistic end user requirements and operational scenarios.

As the complexity of an operational environment increases, so does the requirement to consider the local obstacle space continually, and this is aided by splitting the motion planning functionality into distinct global and local layers. The technical focus of this thesis is on the development and simulation-based testing of a new local motion planning and control framework, where knowledge of i) feasible vehicle manoeuvre constraints ii) local obstacle map iii) current environment conditions are all combined into a continuous receding horizon approach. This framework separates the output and control space elements of the problem, reducing the complexity of the local motion trajectory optimisation and therefore enabling faster design and increased horizon length. Bezier polynomial functions are used to describe local motion trajectories which are constrained to vehicle performance limits and optimised to achieve a specified goal. The primary problem addressed is 'situation-aware' trajectory tracking, but other local motion planning modes are also considered.

Development and testing of the new framework is undertaken within simulation (Matlab), based on a nonlinear 6 degree of freedom model of a quadrotor unmanned air vehicle. Situation-aware trajectory tracking is demonstrated in the presence of static and dynamic obstacles, as well as the presence of realistic turbulence and gusts. The *immediate-term* deconfliction of multiple unmanned vehicles, and multiple formations of unmanned vehicles, is also demonstrated, including the provision of rules-of-the-air type behaviour.

[6] QinetiQ Research Scientist and EngD student at the University of Leicester and the Systems Engineering Doctorate Centre, ajberry@qinetiq.com

Acknowledgements

Firstly, I would like to thank QinetiQ for sponsoring this research, as well as providing a range of support from other researchers in the same and related fields. Thanks of course to my various supervisors, both academic, Professor Ian Postlethwaite and Professor Dawei Gu, and industrial, John Goddard and Jeremy Howitt. A particular thanks to Ian for maintaining interest and supervision even after moving to a new university, to Dawei for his continued support, and to John and Jeremy for helping to obtain QinetiQ support in the early stages of the programme.

Thanks also to the various staff at the SEDC, in particular Roger Goodall, Roger Dixon, Sharon Henson and Karen Holmes, who helped to coordinate the formal training elements and various conferences and other events at Loughborough University. The other EngD students in my cohort, Samantha, Sunny, Chi & Mandeep, should also be thanked for always making the Loughborough sessions entertaining (even during exams). Thanks also to Dr James Whidborne of Cranfield University for providing the Matlab / SIMULINK model of the quadrotor vehicle.

Finally, my biggest thanks must go to my wife Rachel for her relentless support and encouragement, and to my young daughter Emily for not making me feel too guilty for working in evenings and weekends. It’s amazing how much things can change in four years.

Contents

List of Figures

List of Tables

Symbols

$\mathbf{B_i}(\tau)$	ith order basis function
c_i	coefficient for each of the ith order basis functions
J	objective function (or individual term)
$\mathbf{p}(\tau)$	polynomial curve description as a function of τ
r_n	revolutions per minute (rpm) of quadrotor rotor number *n*
$\mathbf{t_{horizon}}$	receding horizon length (seconds)
u	forward speed
v	lateral speed
w	vertical speed
θ	pitch angle
ϕ	roll angle
τ	polynomial curve parameter
λ	scaling factor for individual cost terms
φ	heading angle
γ	flight path angle
∇	gradient operator

Acronyms

ADS-B	automatic dependent surveillance - broadcast
ATC	air traffic control
BLOS	beyond line of sight
COM	computer on module
CONOPS	concept of operation
COTS	commercial off the shelf
DoD	department of defence
DODAF	department of defence architecture framework
EngD	engineering doctorate
FMV	full motion video
GMTI	ground moving track indicator
HMI	human machine interface
INCOSE	international council for systems engineering
IED	improvised explosive device
INS	inertial navigation system
IPT	integrated project team
ISR	intelligence surveillance & reconnaissance
LIDAR	light detection & ranging
LMP	local motion planning
LP	linear programming
LOS	line of sight
MAV	micro air vehicle
MILP	mixed integer linear programming
MoD	ministry of defence
MODAF	ministry of defence architecture framework
NEC	network enabled capability
NMPC	nonlinear model predictive control
PD	proportional differential
QP	quadratic programming
RHC	receding horizon control
ROVER	remotely operated video enhanced receiver
RRT	rapidly exploring random tree
SA	situation awareness
S&A	sense & avoid
SAR	synthetic aperture radar

SDG	statistical discrete gust
SE	systems engineering
SEDC	systems engineering doctorate centre
SLP	sequential linear programming
SQP	sequential quadratic programming
SysML	systems modelling language
SUAV	small unmanned air vehicle
TCAS	traffic collision avoidance system
UAV	unmanned air vehicle
UAS	unmanned air [vehicle] system
UGV	unmanned ground vehicle
UML	unified modelling language
US	United States
USV	unmanned surface vehicle

1 Introduction

1.1 Context of the Research

The research presented within this thesis was conducted under an Engineering Doctorate (EngD) programme, based at the Systems Engineering Doctorate Centre[7] (SEDC). The SEDC is a collaboration between five universities[8] and is based within the Electrical Engineering Department at Loughborough University. The EngD programme is an alternative to the traditional PhD, providing a more vocationally oriented doctorate in engineering. The research engineers (RE) are supervised academically by one of the contributing universities (in this case the University of Leicester) and also industrially by a sponsoring company (in this case QinetiQ) where the RE is based during the research. The role of the academic supervisor matches that of a traditional PhD supervisor, and the role of the industrial supervisor is to ensure that the research matches the technical interests of the sponsoring company.

EngD research at the SEDC must contain elements of systems engineering, with the primary options being:

1. A pure systems engineering research programme
2. Novel application of systems engineering processes to a known technical problem
3. Novelty in a technical area, but undertaken using an overarching systems engineering approach

The approach taken with this programme was the third, with the research predominantly interested in developing a novel technical capability. The aim of the systems engineering framework (discussed in detail in Section 2.2) is primarily to aid definition of a realistic system and operation context within which technical work can take place. The overriding aim of the process was to ensure that technical work occurs with a view to required military needs and realistic operational scenarios.

1.2 Background

1.2.1 Growth in Military Use of Unmanned Vehicles

Over the past ten to fifteen years there has been a spectacular growth in the military use of unmanned aerial systems (UAS). During this time the annual global market for UAS has risen from $100Million to almost $5Billion, and is projected to reach $11.9Billion by 2019 [73]. This growth has been led primarily by the US, during operations in Iraq and Afghanistan, and Israel during recent neighbouring conflicts. However, the popularity of such UAS has also resulted in the development and

[7] http://www.lboro.ac.uk/departments/el/sedc/

[8] Loughbough University, University of Bath, Queen’s University Belfast, University of Leicester and the University of Strathclyde

procurement of many new systems by many other military forces, e.g. the UK's Watchkeeper Tactical UAV System.

There are many key drivers for this growth, including:

- *Reduced risk to human life* - Particularly important in dirty or dangerous missions. Additionally, loss of a pilot also results in significant additional training costs for a new pilot.
- *Increased air vehicle efficiency* - Removing the pilot and all associated systems significantly reduces the weight of the air vehicle, therefore reducing the cost per flight hour.
- *Enabling long endurance missions* - Without a pilot on-board traditional limits on flight length are removed, allowing routine 24hr – 36hr flights. This capability enables new operational procedures, where mission may move away from launch-execute-recover towards maintaining over-watch and having a system on station 24hrs / day to provide rapid response to evolving situations.
- *Reduced theatre footprint* - Satellite based mission control of the vehicles enables the operators to be located anywhere in the world, therefore reducing the number of troops required in the theatre of operations.

However, it is interesting to note that the technology required to provide such capability has been available for some time, therefore this explosive growth has also been influenced by other significant factors, such as:

- *Military need* - Rapid development of a new capability often accompanies an ongoing conflict.
- *Political need* - Reduced tolerance to loss of life leads to greater expenditure on research and development.
- *Suitably benign operational environment* - The vast majority of current UAS would be vulnerable in a hostile air environment, and therefore growth would be significantly curtailed.

It is the combination of these factors, alongside the growing acceptance of unmanned vehicles as well as a desire to move towards net-centric operations that has led to the revolutionary shift in military operations towards unmanned vehicles.

Initially, UAS missions were limited to short duration intelligence, surveillance and reconnaissance (ISR), artillery monitoring and target acquisition type roles. However, since then common operational usage has expanded to include hunter-killer, convoy escort, base protection, 'over the hill / round the corner' reconnaissance and many more (see Section 3.1 for further details). Increasingly sophisticated sensor / mission packs are able to provide many capabilities, including signals intelligence, detailed full motion video and thermal imagery, SAR / GMTI, target designation, as well as on board stores for force application. A spectrum of UAS categories and platforms has now emerged, ranging from high altitude, long endurance (HALE) platforms such as the Northrop Grumman Global Hawk (Figure 1 left), through medium altitude long endurance (MALE) platforms such as the General Atomics Predator (Figure 1 right), to tactical UAVs such as the Boeing ScanEagle and AAI Shadow (Figure 2), and finally to the small and micro UAVs such as the Honeywell gMAV and Aerovironment Raven and Wasp (Figure 3).

The magnitude of the military operational shift that has occurred is succinctly highlighted by the fact that a single Global Hawk UAS, just one of many information streaming platforms in constant operation, uses five times the bandwidth of the entire US Desert Storm operation (see section B.3).

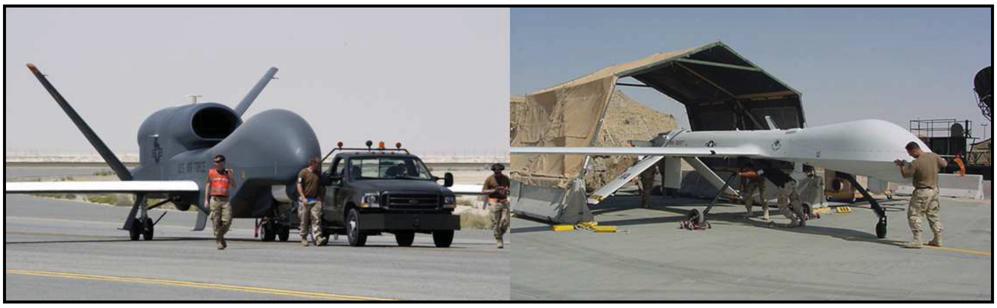


Figure 1 - Global Hawk HALE UAV (left) and Predator MALE UAV

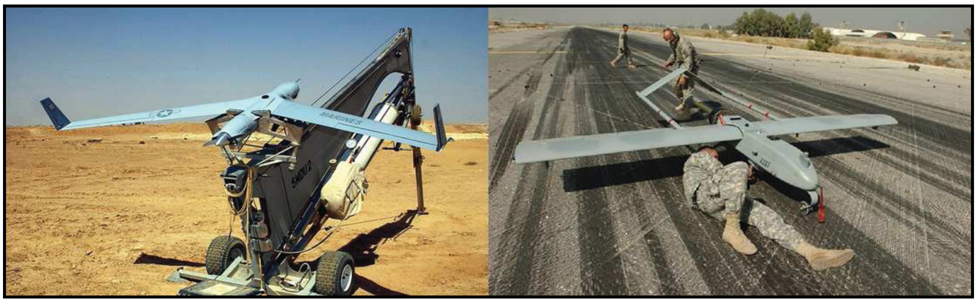


Figure 2 - ScanEagle (left) and Shadow Tactical UAVs

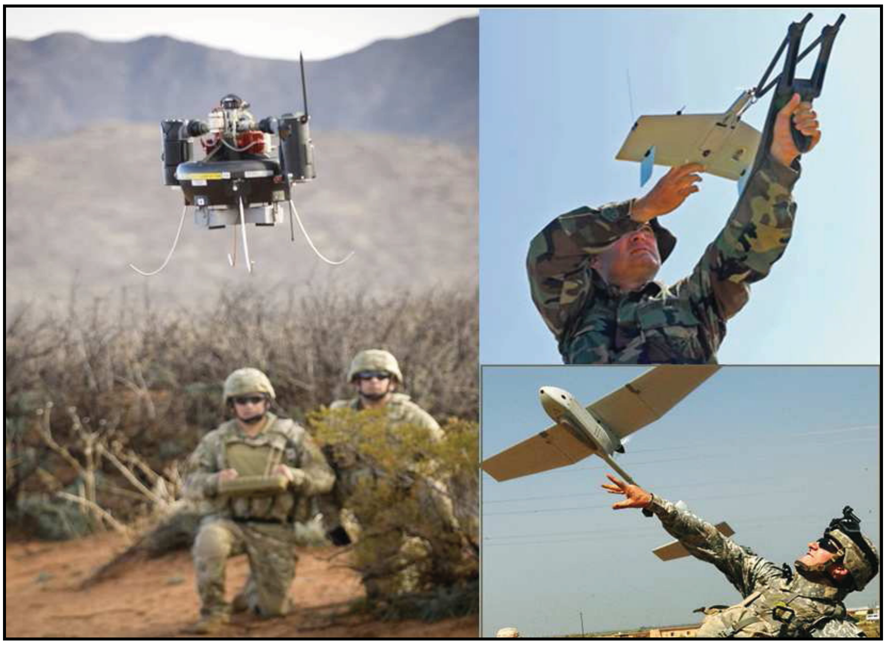


Figure 3 - Class-1 gMAV (left), Wasp (above right) & Raven (below right)

The growth in popularity in unmanned vehicles is not limited to the air domain. Benefits realised by UAV systems, as well as the increased acceptance of the technology, has also led to increasing interest in unmanned (or optionally manned) ground, surface and undersea vehicles. With the exception of bomb disposal robots, these systems tend to be less mature / more developmental than UAV systems, although many desirable missions have already been defined, including:

- *Optionally driven road vehicles* - Unmanned supply convoys reduces human exposure to mines or improvised explosive devices (IEDs). Also allows a driver to take cover in the event of incoming fire.
- *Reconnaissance & surveillance* - Small, man-portable UGVs such as the QinetiQ Dragon Runner (Figure 4 - right) enable ground based scouting of hostile areas, buildings, caves, tunnels etc. prior to entry by ground troops.
- *Assisted carriage* - Helping to reduce the burden carried by ground troops in the field, e.g. Lockheed Martin's Squad Mission Support System (SMSS) (Figure 4 - left)

Ground vehicles have an advantage over air vehicles in that they are less constrained by weight and power, and are obviously less susceptible to wind effects. However, it must be noted that the ground environment is typically more complex than the air, with a higher likelihood of obstructions (such as buildings, rubble, barriers etc.) as well as complex terrain such as forests, mountains, water etc. Additionally, ground vehicles are likely to be more vulnerable to hostile forces and are more limited for ISR type roles (e.g. unable to view roof-top locations).

In the US, this military shift towards unmanned vehicles is being enforced by a Department of Defence (DoD) preference for unmanned vehicles in new system acquisition [7]. In addition to this, the US Congress has set high level goals for the DoD that i) by 2010 a third of operational deep strike aircraft should be unmanned and ii) by 2015 a third of the US Army's FCS operational combat ground vehicles will be unmanned [7].

Many non-military opportunities for unmanned vehicles also exist, including infrastructure monitoring, surveying, coast guard, border patrol, police and fire service etc. Current applications are severely restricted by regulations, however work is underway to address this and many applications are expected to follow.

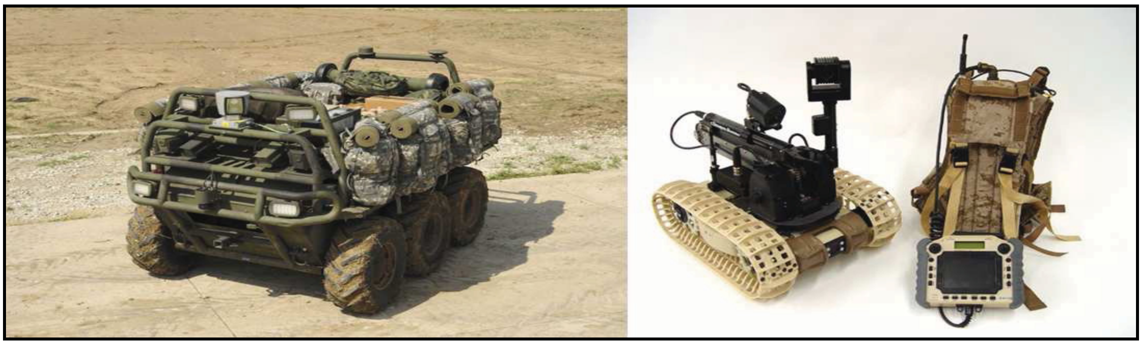


Figure 4 - Squad mission support system (left) and Dragon Runner

1.2.2 Demand for Small & Micro UAVs in Complex Environments

Even though the high numbers of ISR platforms that are now in operation have increased the availability of (near) real-time full motion (aerial) video (FMV) to unprecedented levels, the demand is seemingly 'insatiable' [7]. This is particularly true for ground troops operating within hostile environments where the additional situation awareness that is provided can be life-saving. There are two main options for meeting this requirement:

1. Direct access to video feeds from nearby high value UAVs or manned aircraft
2. Small low cost UAVs operated directly by the source of the demand

The first option is limited in that the supply of such high value assets is unlikely to be able to keep up with demand, therefore restricting availability. The second option is preferred as it has the potential to keep pace with demand and provide the time critical response required. This is a revolutionary capability, allowing small units of ground troops to perform their own aerial reconnaissance without relying on external support.

This growing requirement for small, man-portable, UAVs is complicated by an increasing need for operation within built-up urban environments. Additionally, it is predicted that by 2025 60% of the world's population will live in urban areas [75], therefore this need is likely to grow, with expected roles including humanitarian, peacekeeping and traditional shooting wars.

In addition to military use there is growing interest from the civil sector for small and micro UAVs, with examples including security (police) and safety of life (fire service). Civil operations are currently hindered by more stringent regulatory requirements[9] although work is underway to address this[10]. The focus of this thesis however is military operations.

The primary complicating factors for small and micro UAVs within urban environments include:

- Flight within an urban canyon
- Urban wind effects
- Line of sight (LOS) communications links

[9] CAA CAP-722 - Without a sense and avoid system UAS are limited to operate below 400ft, within line-of-sight of, and up to a maximum distance of 500m from, the operator. Operation of UAS is also allowed within temporary restricted airspace, but this should not be used for routine operations and requires a minimum of 90days notrice.

[10] ASTRAEA - Autonomous Systems Technology Related Airborne Evaluation and Assessment. The programme seeks to research, develop and validate the necessary technologies, systems, facilities and procedures to promote and enable safe, routine and unrestricted use of UASs (www.projectastraea.co.uk).

Each of these issues are discussed further below, and are subsequently addressed to some extent by the research focus of this thesis.

Flight Within Urban Canyon

Firstly, launch and recovery may need to be performed from the current operator location. For example, if a unit of ground troops is under attack then moving to a suitably spacious area for launch and recovery may not be an option. The priority will be to take cover, then launch the UAV in order to detect the enemy location, strength and increase situation awareness.

Secondly, urban environments are often characterised by obstructions to line of sight, therefore complicating the surveillance task. Examples of this are shown in Figure 5, where it is clear that high level ISR assets would never be capable of viewing the obstructed areas. Many other examples of such obstructions exist, including tunnels, bridges, trees, canopies etc.

Additionally, there are many cities around the world with high rise developments as shown in Figure 6. Within such environments sensor limitations of small low cost vehicles may result in flight above the urban canyon being impractical for target identification or designation.

Finally, there may be mission specific drivers for urban canyon operations. Examples include:

- *Kamikaze Micro Air Vehicle (MAV)* - A small or micro UAV with an explosive payload may be used to provide a small diameter non-line-of-sight attack capability (discussed in more detail in Section 4.2.2).
- *Stealth* - Launching a UAV from the current unit location may, in certain circumstances, be undesirable as it could inform enemy troops of the unit's location. In this situation it would be preferable for the UAV to fly within the urban canyon until well clear of the operating unit.
- *Payload delivery* - It may be necessary to line up the UAV next to a building or specific window in order to delivery certain types of payloads

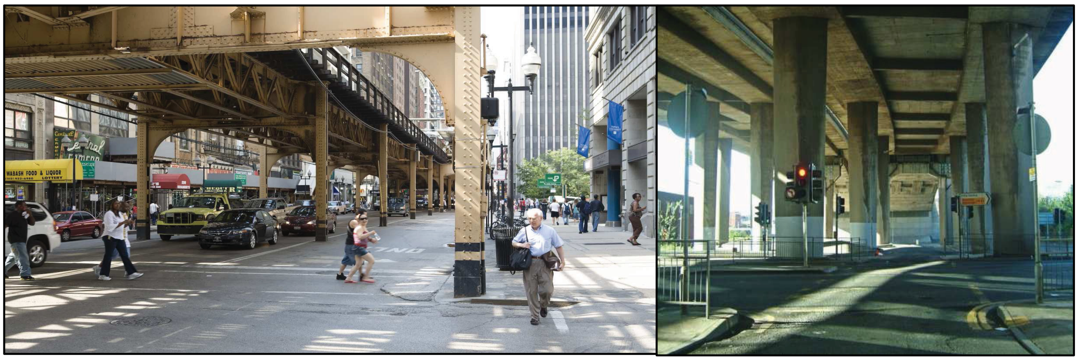


Figure 5 - Example line-of-sight obstructions

Figure 6 - Modern high-rise urban developments

Urban Wind Effects

The impact of urban wind, gusts and turbulence on small and micro UAVs is a significant concern and is discussed in detail in sections 3.2. The issue is fundamentally due to the scale of wind effects in relation to the achievable airspeeds of the vehicle. Steady wind speeds can only be countered by sustainable vehicle airspeed, therefore this will always be a limiting factor for small vehicles. Turbulence and gusts are short term unsteady effects but they will significantly impact the ability of the vehicle to stick to a predefined obstacle free trajectory. This has implications on the chosen planning, guidance and control architecture as discussed in Section 4.3.

It may be feasible to desensitise the vehicle to gusts somewhat [56], or design a vehicle capable of short term busts of control that are able to counter unsteady gust effects in the presence of obstacles. However, given levels of vehicle gust sensitivity and control power the role of the guidance and control architecture will always be to minimise operating restrictions due to current wind conditions for a particular obstacle density.

Although these issues appear prohibitive, nature provides many examples of insects and small birds successfully flying in highly turbulent and gusting conditions, therefore there is no reason to assume that man made vehicles cannot do the same.

LOS Communications

It is likely that small and micro UAVs would be controlled via LOS communication links, therefore causing a significant complication in low level urban environments. Two feasible options for addressing this issue are increasing vehicle autonomy, or using additional vehicles as communications relays. Increased vehicle autonomy is also well suited to operation by dismounted troops in hostile environments where the loss of mobility and immediate location situation awareness that would result from head-down vehicle control may be counter-productive. Operator to vehicle communication would revert to shorts bursts of high level command and control information whenever communication is restored.

1.3 Research Focus

1.3.1 Overview & Current Approaches

The focus of this thesis is the guidance and control of unmanned vehicles within complex, obstacle-rich environments. Algorithm development and simulation based testing is applied to a quadrotor UAV (Figure 7), although the defined framework and algorithms are generic to vehicle type (e.g. fixed wing, rotary wing, ground vehicle etc.).

The complexity of the operational environment may be characterised by i) Proximity of obstacles known *a priori*, ii) Likelihood of encountering disturbances that are significant with respect to the local obstacle space & iii) Likelihood of encountering new obstacles *en route*. UAVs that are currently in-service tend to operate in relatively simple environments, both well clear of obstacles and in airspace that is de-conflicted from other traffic in advance of the mission. Air vehicle guidance and control is typically based around either manual operator control or waypoint based automatic navigation where the vehicle will fly directly towards the next waypoint until a new target waypoint is selected. Air vehicle situation awareness[11] is typically limited to either operator control based on real-time visual cues, or an *event triggered* sense and avoid (S&A) system. With the event triggered approach the vehicle is controlled to the desired trajectory in a *blind* fashion (with no consideration of the local obstacle space), while separate collision avoidance functionality scans the local space, triggering an evasive manoeuvre if required.

This type of guidance and control architecture, with its clearly separated guidance-control and situation-awareness, becomes less appropriate as the complexity of the operational environment increases. Firstly, a degree of vehicle autonomy is highly desirable, therefore precluding a reliance on manual control. Additionally, waypoint based navigation suffers from a lack of

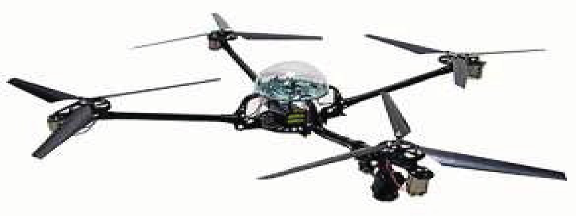


Figure 7 - Quadrotor UAV

[11] Within this thesis in the context of unmanned vehicle guidance the term ‘situation awareness’ is used to refer to the vehicle guidance and control algorithms being aware of the vehicle’s local obstacle space, including static and dynamic obstacle known both *a priori* and detected *en route* via on board sensors.

continuous ground track specification, therefore leaving the vehicle position incompletely specified during heading changes. This would not be appropriate during operation within complex (and sometimes dynamic) environments, suggesting that the trajectory description should be continuous. Finally, if the vehicle is expected to operate in close proximity to obstacles, and significant disturbances are expected, then the motion planning and control architecture needs to continually consider the local obstacle space, rather than treat it in an event triggered fashion. A small or micro UAV will be highly susceptible to turbulence and gusts, therefore making it unlikely (or overly restrictive in terms of operating conditions) that it will be able to accurately hold a pre-defined obstacle free trajectory. As the frequency of obstacle proximity events increases, it is likely to become more efficient to endow the trajectory tracking algorithms with situation awareness, rather than rely on regular switching between trajectory tracking and collision avoidance modes.

An additional complication to tracking a global trajectory within a complex environment is the detection of unexpected obstacles. These can be broadly be classified into one of two types:

- Those that necessitate a change to the global plan (e.g. a large obstacle that requires significant deviation from the original trajectory, indicating a mismatch between the obstacle map used to generate the global trajectory and that detected *en route*)
- Those that do not affect the global plan (e.g. overhead cables, other vehicles, birds, sign posts etc.)

For the first type of unexpected obstacle there is little alternative but to attempt to maintain a safe distance from the obstacle while a large scale trajectory re-plan is performed. However, with the second type of obstacle a human pilot would intuitively manoeuvre around it without significantly deviating from the global plan. For example, if another MAV is encountered while following a trajectory then it should be relatively straightforward to manoeuvre around it and re-join the original trajectory. Similarly, if while following a trajectory an overhead cable is detected then it only requires a small height adjustment to avoid a collision, which can again be achieved without regenerating the global plan. A *blind* trajectory tracking layer would be unable to perform such manoeuvres as it has no knowledge of the surrounding obstacle space, therefore again suggesting utility in a situation aware trajectory tracking layer.

1.3.2 Global / Local Guidance & Control Architecture

In order to address the issues discussed above, the approach taken within this thesis is to divide the vehicle motion planning architecture into distinct global and local layers, as described below:

- *Global Motion Planning* - Mission / goal focused, possibly coordinating multiple vehicles within the same system. Global trajectory updated in response to new information / mission goals, but this is expected to occur on a timescale that allows the lower layers to consider it as static.

- *Local Motion Planning* - High rate planning, with horizon limited to a maximum of the sensor horizon, allowing greater emphasis on accurate vehicle dynamics and rapid reaction to disturbances and newly detected obstacles. Responsible for tracking the global trajectory, given detailed knowledge of the local situation (e.g. disturbances, obstacles detected en-route, local wind conditions etc.)

This architecture is analogous to a human driving a car while being directed by a GPS route planner. The route planner provides the global goal directed plan which the driver is then responsible for following, while performing lane following / changing, traffic de-conflictions and adjusting for other local environment issues, e.g. traffic lights, obstacles on the road, speed restrictions etc.

Within the motion planning literature many viable approaches may be found for application to the global problem of planning a dynamically feasible path through an obstacle field, e.g. rapidly exploring random trees (RRT) [41], Voronoi diagram [28], visibility lines, graph based searches such as A* etc. The local motion planning and control work in the literature is less mature at present[12], hence this is the primary technical focus of this thesis.

1.3.3 Local Motion Planning & Control

A suitable framework for the solution of the local motion planning and control problem may be found in receding horizon control (RHC), also known as model predictive control. Fundamentally, this is a control design approach, where a vehicle model is used to predict resulting output space trajectories over a specified design horizon. The problem may be posed in such a way that, given a formulation of the output space goals a feasible control sequence that optimises the resulting vehicle output space trajectory may be calculated. Only the immediate control inputs are implemented, and the problem then re-solved at the next design step.

While this framework may be directly applied to the local motion planning problem, there are some practical difficulties that must be considered. Firstly, the accuracy of the output space design depends on the vehicle model. A complex 6 degree of freedom (DOF) nonlinear model will provide highly accurate predictions, but will require high computational effort. The computational effort may be reduced by simplifying the vehicle model, but only at the expense of accuracy, of both the required control signals and the resulting output space trajectory. Additionally, the required computational effort is strongly affected by the design horizon, with increasing horizon length also increasing the dimension, and hence the optimisation effort, of the problem. This results in a three way trade off between design horizon, vehicle model accuracy and re-design rate, which for a vehicle with fast dynamics (such as a MAV) significantly compromises the ability to provide key design goals of horizon length and design rate.

[12] This is discussed further in Section 3.3 which discusses a literature search of this field.

A further complication is that as the accuracy of the vehicle model decreases, so does the accuracy of the designed control signals, therefore requiring an additional tracking layer to accurately follow the designed output space trajectories. Even with a full 6DOF nonlinear vehicle model the designed control signals are unlikely to be perfect, and given the presence of disturbances this suggests that an additional tracking layer will still be necessary.

Given this requirement for a tracking layer, and the likelihood that the vehicle model will need to be significantly simplified to provide the desired horizon length and planning rate, this thesis considers the division of the receding horizon problem into two distinct components:

- *Receding horizon trajectory design* - output space component
- *Receding horizon trajectory tracking* - control space component

The aim of this division is to simplify the optimisation process, allowing both high rate design and a horizon length that encompasses the vehicle manoeuvre dynamics. It is also noted that this division is suited to the dynamics of the vehicle, with the control space component needing to run at a higher rate than the output space component.

1.3.4 Deconfliction of Multiple Unmanned Vehicles

An additional issue of concern to this thesis is the deconfliction of multiple unmanned vehicles. As UAVs become smaller, lighter, cheaper and more capable, the likelihood of multiple vehicles, or multiple systems of multiple vehicles, operating in a single complex environment will increase. Many forms of deconfliction exist, each operating over various time horizons and placing different demands on each vehicle, e.g. centralised airspace segregation, centralised routing of multiple vehicles, decentralised cooperative collision avoidance, rule based decentralised uncooperative collision avoidance etc. In order to provide the desired rapid reaction and operational flexibility it is likely that decentralised deconfliction would play a key role.

This thesis also considers the use of the local motion planning framework to enable immediate term decentralised deconfliction of multiple vehicles, investigating cooperative, uncooperative and rule based approaches.

1.4 Contribution of this Thesis

The primary contributions of this thesis are:

- Definition of a new generic framework for continuous local motion planning (LMP) and control. This framework focuses on the division of the traditional receding horizon control problem into distinct output and control space problems. The aim of this division is to simplify the resulting optimisation problem, allowing faster design and increased horizon length.
- Application of the new LMP framework to a quadrotor UAV, including simulation based demonstration of:

    - Operation in close proximity to static obstacles
    - Avoidance of dynamic obstacles
    - Complex environment operation in the presence of significant wind / turbulence disturbances.
- Further development of the new LMP framework for application towards immediate term deconfliction of multiple vehicles, including:
    - Provision of pre-agreed multi-vehicle deconfliction rules, and other rules-of-the-air type behaviour.
    - Demonstration of efficient handling of complex multi-vehicle scenarios.
    - Demonstration of MAV formation safe scatter / reform in the presence of unexpected obstacles and disturbances.
    - Demonstration of deconfliction of multiple formations of multiple vehicles.
- Demonstration that a continuous output space receding horizon trajectory design can be accurately tracked by a simple control space layer, therefore providing an alternative approach for the application of receding horizon control.
- Identification, implementation and simulation based demonstration of other LMP modes of operation, including:
    - Leader offset formation flight
    - Neighbour offset formation flight
    - Point to point guidance
    - Tracking a moving target
- Generation of realistic & desirable potential mission profiles for military MAV operations in urban environments.

1.5 Publications & Intellectual Property

Certain technical developments presented in this thesis are subject to the following patent applications:

- Europe: Application Number EP 10251329.8
- US: Application Number US 12/846,220

Elements of this thesis have appeared in the following publications:

- Berry, A., Howitt, J., Postlethwaite, I., Gu, D, 2009 “Situation Aware Trajectory Tracking for Micro Air Vehicles in Obstacle-Rich Environments” AIAA Guidance, Navigation & Control Conference
- Berry, A., Howitt, J., Postlethwaite, I., Gu, D, 2010 “Enabling the Operation of Multiple Micro-Air-Vehicles in Increasingly Complex Obstacle-Rich Environments” AIAA Infotech@Aerospace Conference

- Berry, A., Howitt, J., Postlethwaite, I., Gu, D, 2010 “Local Motion Planning & Control for Micro-Air-Vehicles in Complex Environments” AIAA Guidance, Navigation & Control Conference
- Berry, A., Howitt, J., Postlethwaite, I., Gu, D, “A Continuous Local Motion Plannign Framework for Unmanned Vehicle Operation Within Complex Environments” To be submitted December 2010.

1.6 Thesis Structure

The remainder of this thesis is as follows:

- Chapter two discusses the subject of Systems Engineering (SE) and defines the overarching research framework that was followed.
- Chapter Three presents the results of four literature reviews: i) Military experiences with UAS ii) Guidance & control approaches for unmanned vehicles, iii) Primary characteristics of wind flow within urban environments & iv) Optimisation approaches.
- Chapter four is the repository for the majority of the systems engineering effort, detailing high level mission requirements, realistic mission profiles and additional contextual information that helps to inform the technical development. This chapter finishes with a definition of the technical problem to be addressed by the remainder of the thesis.
- Chapter five presents the new LMP framework, focussing on the output space component of the defined situation aware trajectory tracking problem.
- Chapter six presents simulation results for the LMP framework for a single quadrotor UAV, including obstacle free trajectory tracking and the avoidance of both static & dynamic obstacles.
- Chapter seven presents the development of the single vehicle LMP framework to provide decentralised immediate term deconfliction of multiple vehicles and multiple formations of vehicles. The provision of search biases including rules-of-the-air type behaviour is also presented.
- Chapter eight presents a proof of concept of the control space component of the defined problem, demonstrating situation aware trajectory tracking with a full six DOF nonlinear model of the quadrotor UAV.
- Chapter nine demonstrates the use of the situation aware LMP framework to provide behaviour other than trajectory tracking. Simulation results are presented for point to point based guidance and tracking of a dynamic target.
- Chapter ten considers some final system level issues, including realistic environments, gust / turbulence disturbances and computational effort.
- A discussion and conclusions are then presented in Chapters eleven.

2 Research Framework - *A Systems Engineering Approach*

2.1 Introduction to Systems Engineering

"Systems engineering is the core competency that will underpin profitability and growth in tomorrow's industry and commerce"[13]

2.1.1 Complexity of Modern Systems

Systems engineering (SE) is a relatively new[14] interdisciplinary field of engineering that is aimed at managing complexity in the design and development of systems. The degree of complexity to be found in many modern systems is unprecedented, with well known examples including telecommunication satellites, information technology systems, the international space station, the large hadron collider, space & planetary exploration etc. Military fighter and civil transport aircraft have been around for many years, but the current generation such as Eurofighter Typhoon, Lockheed Martin Raptor, Airbus A380 and Boeing 787 Dreamliner contain levels of sophistication previously unseen.

The primary enabler of this rise in complexity is the increasing availability of powerful low cost and lightweight processing power, allowing either new functional capabilities or efficiencies to provide either a military or commercial advantage. Within the aerospace domain software enabled control has provided many benefits. Perhaps the most influential of these include:

- *Fly-by-wire* - Allowing stabilisation of highly manoeuvrable or stealthy platforms, standardisation of flight response characteristics for reduced pilot training, novel task oriented control modes or care-free handling to reduce human errors.
- *Glass-cockpits* - Providing simplified and standardised pilot displays to enable reduced training and type conversion costs.
- *Increased flight automation* - Enabling either remote / autonomous operation or a more efficient task / mission oriented role for the pilot (particularly important due to the increasing number of on-board systems).
- *Engine management* - Providing increased reliability and efficiency.
- *Health & usage monitoring* - Enabling maintenance schedules to be optimised to suite the particular vehicle usage. A highly successful example if this is modern Rolls Royce aero engines, which are able to continually monitor themselves, providing real-time alerts to both the pilots and ground based service centres in the

[13] Systems Engineering Innovation Centre www.seic-loughborough.com

[14] International Council for Systems Engineering (INCOSE) was founded in 1991.

event of a detected fault. In many cases the systems are sophisticated enough to diagnose the failure, schedule the required maintenance, also ensuring that the required parts are available on landing.

These and other newly enabled capabilities typically result in a greater level of integration between a growing number of system components. This increased integration between many system elements is a key driver of complexity, making it harder to predict or manage the impact of changing system requirements.

2.1.2 Development Issues with Modern Complex Systems

While the existence of this range of highly complex systems displays our ability to solve the related technical or scientific problems, the development of such products is typically characterised by delays and cost overruns. Several highly publicised examples are shown in Table 1.

Astute class submarines	2009 National Audit Office estimate was a cost overrun of £1.4Billion (approx. 34% increase since major decision) and the in-service date delay of 57 months [46].
Airbus A400M Military Transport Aircraft	2009 National Audit Office estimate was a cost overrun of £657Million (approx. 25% increase since major decision) and the in-service date delay of 82 months [46].
Queen Elizabeth class aircraft carriers	2009 National Audit Office estimate was a cost overrun of £1Billion (approx. 25% increase since major decision) and the in-service date delay of 10 months [46].
Eurofighter Typhoon (tranches 1 & 2)	2009 National Audit Office estimate was a cost overrun of £847Million (to UK alone) (approx. 5% increase since major decision) and the in-service date delay of 54 months [46].
Airbus A380	In service date delayed by 2 years, unit costs increased by 25%. Significant delays due to issues with configuration management, wiring arrangements and transition between initial and full rate production.
Boeing 787 Dreamliner	First flight delayed by 2 years due to both supplier and technical issues including wing box failure due to incorrect modelling of material strength. Orders for 71 planes cancelled.

Table 1 - Examples of significant large scale project over-runs

The impact of these issues affect both the manufacturer (via a combination of unexpected costs, scheduling and customer relationship issues) and the customers (via increased costs and schedule difficulties). Indeed, the scale of these issues is such that commercial differentiation between manufacturers in no longer solely based on the ability to solve the technical issues, the ability to efficiently manage the development process is also key. While a significant component of the overruns may simply be an optimistic bias towards planning and costing, it is clear that significant inefficiencies with managing the system complexity also exist. The existence of such inefficiencies suggest that value may be gained by improved management of system complexity. SE attempts to realise this value by engineering (or designing / optimising) the organisational thinking, processes, tools and methodology used to develop a complex product.

2.1.3 Systems Approach, Concepts & Toolset

In order to address the issues that arise from this increase in system complexity and integration, the field of SE offers several overarching principles, as well as a toolset of design standards, architecture frameworks and design tools. These are all discussed further below.

The primary overarching SE principles include:

- *Importance of the early stages of development* - It is typically found that the cost of removing bugs or design errors increases significantly with time through the development process. This is an intuitive result as it is easy to see why altering a design during early concept development will cost less than during manufacturing. The scale of the cost escalation can be very significant, therefore any work that can be done up front to prevent the introduction of a bug or design misunderstanding therefore represents a lower cost option overall than fixing it at a later stage. SE thinking encourages a *proactive* approach to searching for potential errors, bugs or requirement-gaps, as opposed to the traditional *reactive* (or fire-fighting) approach where the design team will attempt to fix issues as they arise during system design and development.
- *Emphasis on team based development* - The importance of team work throughout the project is also emphasised. The domain knowledge of a team is always greater than that of an individual, and team working can be useful in helping to stimulate thought and ideas. Additionally, people often interpret statements / requirements in different ways, therefore a team is used to force discussion and come to an agreement. Multi-disciplinary teams are encouraged to ensure that different viewpoints, e.g. training, logistics, disposal etc., are considered at an early stage in the project.
- *Structured understanding & response to customer needs* - Key here is to ensure that customer needs & the associated problem space are fully understood, including the demands that this places on customer time. It is recognised that customers are unlikely to provide a complete set of requirements. This is not a failing of a particular customer, but rather a realistic outlook given the complexities that are often found in modern engineering projects. Recognising this difficulty up front allows the design and development process to be engineered to deal with this difficulty, investing significant *early-stage* effort into requirements generation and analysis. Translation of the problem into measurable requirements is a must, which then allows structured analysis of requirements for gaps, conflicts etc.
- *Consideration of the entire system life-cycle & wider environment* - This is intended to encourage early stage consideration of any linkages between the system and the greater external environment. An attempt should be made to uncover all the project

stakeholders and understand the various interactions between them. This again focuses on early stage uncovering of hidden requirements.

- *Abstraction of system functionality from implementation* solutions - This is intended to aid innovative thinking by avoiding preconceived notions of how a solution will be provided. It avoids thinking in terms of what has been done in the past, instead focusing the initial design process on generic system functionality, e.g. what the solution needs to do, not how it will be done.
- *Top down functional development* - Eases the handling of complexity by allowing design complexity to flow down into deeper levels of functionality.
- *Systematic approach towards handling the above principles* - The above principles should be structured into a formal / systematic development process that ensures appropriate effort is directed at certain key tasks.

Two examples of SE based design standards are the systems modelling language[15] (SysML), and model driven architecture[16] (MDA). SysML is a systems based adaptation of UML that allows definition of system requirements, structure and behaviour. MDA is a standard which promotes the concept of a move from document-centric to model-centric design.

An example of an architecture framework that is commonly used in the defence industry is the ministry of defence architecture framework[17] (MODAF). This framework defines several different viewpoints of a system architecture, each providing a range of artefacts that provide different categories of information, e.g. structural, behavioural, tabular, pictorial etc. Underlying the framework is the fact that information gained from a particular viewpoint may also feed into a different viewpoint. For example, the definition of a MODAF OV-2 (Operation Node Relationship Diagram) will partly define the information that is required for a MODAF OV-3 (Operational Information Exchange Matrix). The framework can then ultimately be seen as defining a central system data repository (or core architectural model) from which different viewpoints can be automatically generated as required.

These standards and architecture frameworks have been implemented in several computational design packages such as IMB's Rhapsody, Sparx Systems Enterprise Architect, Kennedy Carter xUML, ARTiSAN Studio. These implementations allow the interactions between different systems elements and viewpoints to be modelled and understood at an early stage in the design. There are two critical uses for these architecture tools, both of which are discussed briefly below.

Firstly, a precise and comprehensible system architecture can be used as an aid to generating a shared understanding of the system. This can be used to enable an

[15] www.sysml.org

[16] www.omg.org/mda

[17] www.modaf.org.uk

effective dialogue between the customer, the user and the manufacturer regarding the proposed system, it's intended operational use and expected behaviour etc. Correctly aligning this understanding at the initial stages of a project is critical to it's success. Additionally, it is also essential that the project development team share the same interpretation / understanding of the system, and the architecture again helps here. Even after project development is complete the system architecture may be used to allow a wider understanding of the system, which may help in the formation of system of system or family of system capabilities.

Secondly, system architecture concepts and tools allow a high level design to take shape, helping to manage system complexity and allowing clear specification of both intended system components, arrangements, interfaces, functionality, behaviour etc. The designer can explore the system design space, allowing a greater insight into the problem to be gained. Architectural design utilises abstraction to allow a single or small group of people to maintain an overview of the entire system, something that is critical for maintaining conceptual integrity.

These systems principles are tools are typically incorporated into the well known system development V-Model, as shown in Figure 8. The aim of this development process is to emphasise the need for:

- Clear definition of user and system requirements.
- Clear linkages between user and system requirements and subsequent verification and validation.
- Problem definition and decomposition.

While the field of SE originally developed out of a need to better manage complex development projects, there is increasing interest in the benefits that a systems approach may also bring to research projects. An example of this is the Systems Engineering for Autonomous Systems (SEAS) Defence Technology Centre[18] (DTC), where an overall SE framework was used to manage and aid exploitation of the individual research programmes. The SE approach to this research programme is discussed within the next chapter.

[18] www.seasdtc.com

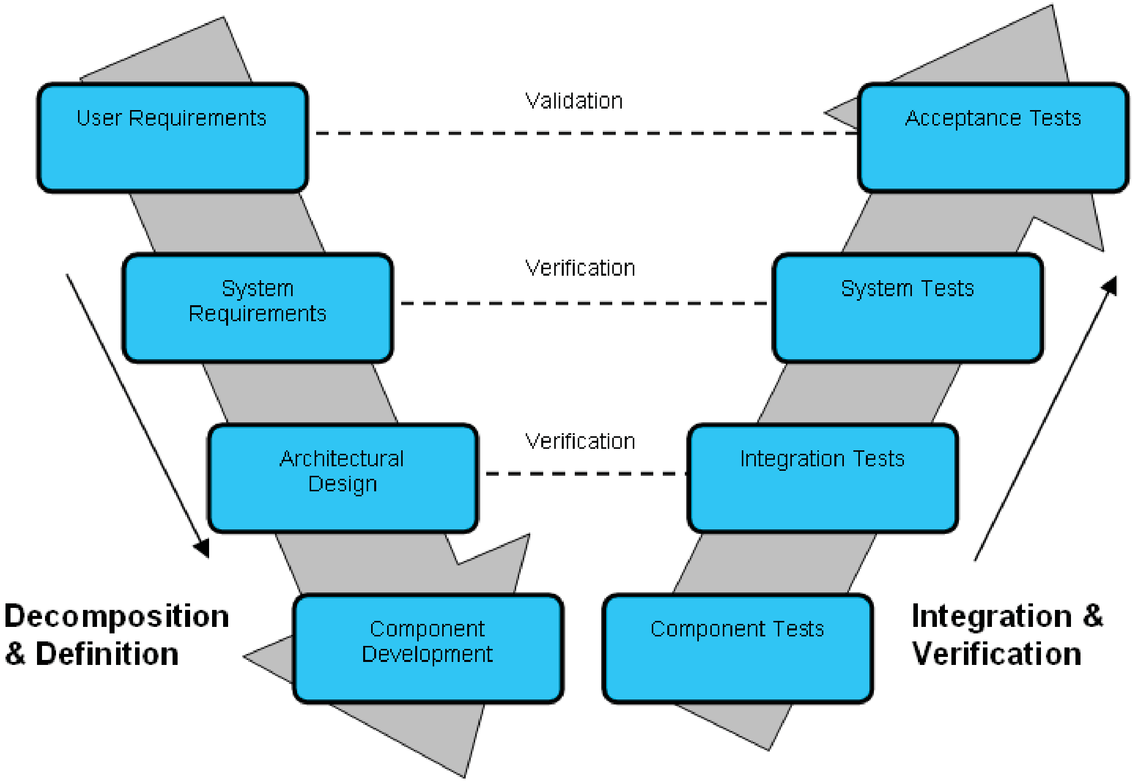


Figure 8 - Idealised systems engineering V-Model

2.2 Research Programme Work-Plan

The work plan used within this research programme was developed from an SE perspective, with the overriding aim of ensuring that the technical work was conducted with a view towards realistic and desirable military applications. An overview of the work plan is shown in Figure 9, where it can be seen that it was based on the idealised SE V-Model process, following an overall top-down, bottom-up approach. The top-down effort is shown on the left side of the 'V', focussing on problem scoping, definition and decomposition. This side contains the primary SE input to the research, adding additional early stage effort towards helping to ensure that there was sufficient understanding of military need, operational requirements and context before beginning to address detailed technical issues that were highlighted. The bottom-up effort is shown on the right side of the V-Model, detailing various stages of the technical work. This technical effort was targeted at specific requirements / problems defined during the previous scoping and decomposition effort. Further detail on each of the stages of work is given below.

Top-down scoping, decomposition and definition

- *Initial literature reviews (sections 3.1 & 3.3)* - There were two focuses to this stage. Firstly, effort was directed at learning from the current military experience with unmanned vehicles, including the types of vehicles in service, current operational uses, successes, failures, lessons learned etc. Secondly, a review of unmanned vehicle guidance and control architectures and approaches was conducted. This was a general cross domain review, aiming to understand current capabilities.

- *Define high level capability requirement (section 4.1*) - The aim of this stage was to use the information from the previous review of military experience with unmanned vehicles to define a realistic and desirable military requirement that the subsequent research would be directed towards.

- *Capability context exploration & definition (section 4.2)* - This stage was directed towards providing greater contextual information regarding the defined high level capability requirement that could be used to inform subsequent technical development and system level test and demonstration. Effort was directed towards greater understanding of desirable mission profiles, the operational environment (e.g. urban wind effects), capability visions and the wider system context (e.g. sensor requirements, operator interface etc.).

- *Logical model of guidance & control architecture (section 4.3)* - The defined high level capability requirement and contextual information was then used to create a proposed logical model of a guidance and control architecture suited towards unmanned vehicle operation within complex obstacle rich environments.

- *Identify technology needs of the logical architecture (section 4.4)* - The aim here was to map current capabilities on to the defined logical architecture, therefore identifying capability gaps that may be used for detailed technical development.

- *Detailed problem definition (section 4.5)* - The identified capability gap from the logical architecture was then used to create a detailed definition of a technical problem to be address by the remainder of the research.

Bottom-up detailed technical development, test and integration

- *Targeted literature reviews (section 3.3)* - This literature review was targeted directly at the previously defined technical problem, focussing of detailed guidance and control algorithms and optimisation approaches that may be suitable for real-time operation.

- *Development of local motion planning framework (chapter 5)* - Technical development of output space algorithms suited to the defined technical problem, and informed by the operational and system contextual information.

- *Single vehicle output space test & development (chapter 6)* - Multiple scenario test and development of the local motion planning framework, target at single vehicle operations.

- *Multiple vehicle output space test & development (chapter 7)* - Multiple scenario test and development of the local motion planning framework, target at multiple vehicle operations.

- *Control space test & development (chapter 8)* - Proof of concept effort targeted at the control space component of the defined technical problem.

- *Systems level test & demonstration (chapter 10)* - The goal here was to test and demonstrate the developed technical algorithms applied to the proposed mission profiles within realistic operational environments (e.g. including turbulence & gust effects).

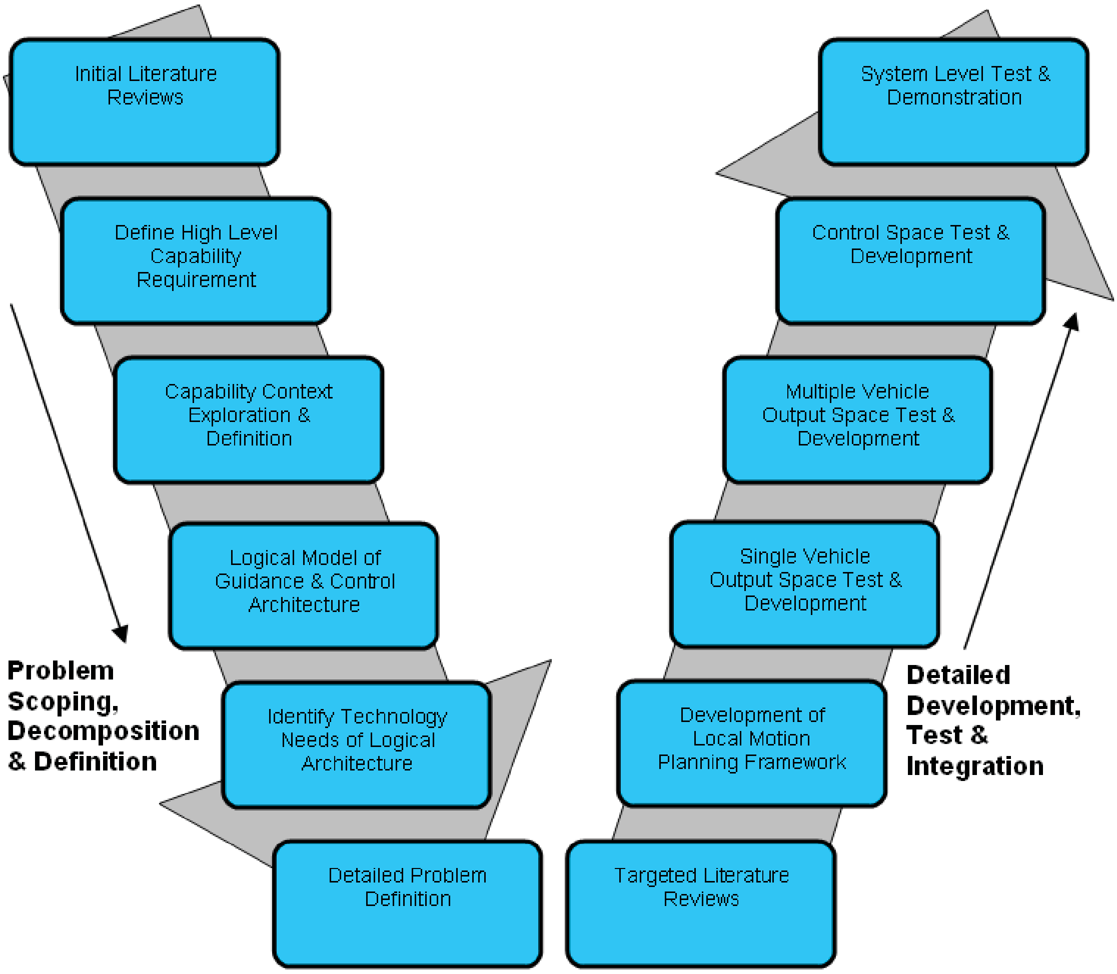

Initial Literature Reviews
Define High Level Capability Requirement
Capability Context Exploration & Definition
Logical Model of Guidance & Control Architecture
Identify Technology Needs of Logical Architecture
Detailed Problem Definition
Problem Scoping, Decomposition & Definition
Targeted Literature Reviews
Development of Local Motion Planning Framework
Single Vehicle Output Space Test & Development
Multiple Vehicle Output Space Test & Development
Control Space Test & Development
System Level Test & Demonstration
Detailed Development, Test & Integration

Figure 9 - Systems engineering research plan

3 Literature Reviews

3.1 Military Experience with Unmanned Vehicles

> *"The demand for game-changing, UAV-borne capabilities is insatiable"*[19]

The starting point for this research was a review of military operational experience with UAVs. The aim of this was to ensure that the research would be informed by lessons learned, current operational roles, successes and challenges etc. Information was obtained from a variety of sources, including aviation and defence press, armed forces briefings, US congressional testimony etc. Key information that emerged from this review is presented below, with article and testimony extracts provided in Appendix A, along with specific examples of operational uses.

3.1.1 UAV Operational Roles

Examples of typical operational roles of UAVs that have emerged out of military operations in Iraq and Afghanistan include:

- *Persistent surveillance & reconnaissance* - A key benefit of many UAVs is the enhanced endurance when compared with manned assets. For example the Predator and Global Hawk platforms can stay in the air for over twenty-four hours, something that is not feasible with manned platforms. This capability enables a greater aerial presence, allowing typical missions to move away from traditional short-endurance missions where aircraft are launched to perform a specific task, towards enabling persistent over-watch and providing rapid response to evolving situations. For example, a UAV may be launched to perform routine reconnaissance or improvised explosive device (IED) detection but it would also be on call to be reassigned to provide coverage of an unexpected event such as an IED explosion. During major situations (near) real-time FMV may also be relayed to tactical operations centres allowing senior commanders to view events as they happen. Such rapid reaction and operational flexibility also allows extended tracking of targets of opportunity.
- *IED detection [A.1]* - This is a common role for surveillance and reconnaissance UAVs using aerial imagery to detect suspicious activity, e.g. digging near a road at night. Common operational routes, busy intersection etc. may be monitored day and night limiting the ability of insurgents to plant road-side devices. In addition to this persistent surveillance, aerial imagery may also be compared with the same imagery at a previous time, allowing change detection algorithms to highlight risk spots for further investigation.

[19] US Air Force General Norty Swartz, September 25, 2009

- *Base protection / perimeter defence [A.2]* - Surveillance platforms are often used to monitor activity around bases, helping to detect the source of incoming fire, coordinating response and tracking fleeing insurgents.
- *Route reconnaissance [A.3]* - Prior to moving troops or supplies, it is now common to use UAVs to perform a reconnaissance role, looking for IEDs or ambush sites.
- *Convoy escort* - While troops and supplies are in motion it is now common for them to be accompanied by a UAV, which will monitor the surrounding area and scout the upcoming route.
- *Over the hill, round the corner reconnaissance & surveillance [A.4]* - The key aim here is using the UAV platforms to provide aerial imagery that can increase situation awareness. This capability is typically provided by small UAVs such as Desert Hawk (Figure 10), Dragon Eye (Figure 10) and Raven (Figure 3) that can be operated by small units of ground troops.
- *Strike / Hunter killer* - In order to reduce ‘sensor to shooter’ times and address frustration felt by UAV operators over monitoring insurgent activity but not being able to react, certain UAV platforms now carry weapons (e.g. Predator, Reaper, Warrior). This capability allows quick reaction strikes on targets of opportunity and enables long-duration hunter-killer missions. In fact it has been reported that this capability has almost completely replaced the use of Tomahawk Cruise missiles [73].
- *Launch detection* - Israeli forces have used a persistent network of Hermes 450 UAVs to locate the source of Katyusha rocket launchers, allowing target coordinates to be forwarded to fighter aircraft.
- *Precision target designation* - Certain platforms (Predator, Reaper, Shadow) carry laser designators that are able to illuminate targets for either their own weapon systems or that of other platforms.
- *Signals intelligence (SIGINT)* - Persistent UAV platforms such as Global Hawk are also commonly used for SIGINT roles (or electronic surveillance), where on-board sensors are able to intercept enemy communications or radar, surface to air missile systems etc.

Other roles that UAV platforms are commonly used for include battle damage assessment, communications relay, artillery guidance etc. However, a complete discussion is beyond the scope of this thesis.

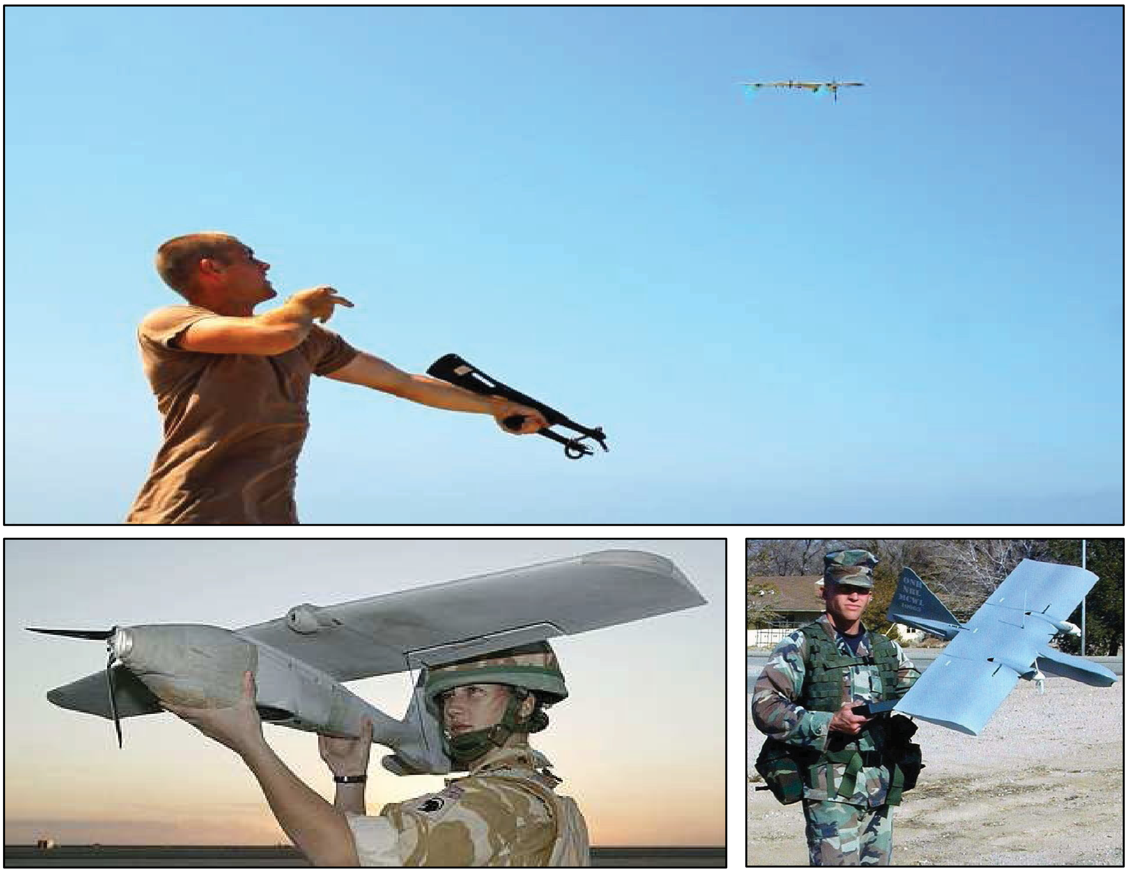


Figure 10 - Wasp (above), Desert Hawk (left) & Dragon Eye (right) small UAVs

3.1.2 Key Operational Issues

"Not a single commander in Iraq or Afghanistan will tell you that he or she is happy with the amount of available UAV support"[20]

Key issues that have emerged during recent UAV operations include:

- *Lack of system availability [B.1 & B.5]* - A common theme that emerged was that the increasing availability of near real-time aerial imagery has further fuelled the demand. The popularity of small UAVs at the company or small unit level has been driven by the inability of high value assets to provide the time-critical availability that is often required. This problem is exacerbated by airspace congestion issues, where flight plan requirements and coordination with air traffic control can significantly increase the response time of platforms that are available.
- *Airspace congestion [B.2]* - Increasing numbers of small, tactical and strategic UAV assets have significantly complicated the air traffic situation. The growing number of UAVs share the airspace with many other assets such as fast jets, helicopters, missiles, rockets, mortars, artillery etc. and this has resulted in complex coordination requirements that reduce the flexibility of the platforms.

[20] Statement by US Brigadier General Jeffery Schloesser to House Armed Service Committee, 2005

- *Lack of access to aerial imagery from available UAVs* - Systems such as the V-Rambo (video receiver and monitor for battlefield operations - Figure 11) were developed in response to frustration that even when UAV assets were available, the imagery was only seen by the operator and commanders. V-Rambo allows ground troops direct access to aerial imagery from local assets.

- *Frequency congestion [B.3]* - There is a general shortage of available bandwidth, which may result in a reduction of operational capability [5]. Additionally, some vehicle types use the same frequencies, requiring the additional ground coordination prior to major operations. It has also been reported that communication links may be jammed (potentially by the link to another vehicle), resulting in either a lost vehicle or terminated mission. Intermittent control provides a strong driver for increase vehicle independence.

- *Frustration over inability to engage detected targets [B.4]* - Smaller platforms such as the Raven are unable to provide accurate target locations, therefore requiring additional verification before engaging detected targets. Additionally, operators of larger platforms with suitable target designation but no on board weapons have also reported frustration over an inability to act. This is particularly true when a suitable weapon carrying platform or ground based unit is not immediately available, resulting in the tracked target evading capture. This is a particular concern in urban environments, where continuous target tracking may not be possible.

- *Noise levels [B.6]* - Engine noise can be both an advantage and a disadvantage. The obvious downside is lack of stealth. Raven operators have reported this as a concern, and often use gliding manoeuvres to attempt to avoid detection. However, the plus side is that the noise alone may be enough to discourage enemy operation in a given area, and this has been reported to be successful.

- *Line of sight communication links [B.3]* - Most small and tactical platforms use line of sight links, which may reduce the operational radius. This is a particular concern for small UAVs that operate at low altitudes. Additionally, in urban environments communication links can be lost when attempting to look behind buildings. The Raven communication link is also reported as being heavily directional, therefore requiring the operator to point the control link towards the vehicle. A complicating factor is that low level flight provides better quality imagery, but also restricts line of sight range.

- *High vehicle losses [B.7]* - Reliability issues have been reported when compared to manned aircraft, due to mechanical failures, operator errors and rapid fielding of systems without fully established reliability and maintenance infrastructures [5]. For example, Canadian forces deployed a single Crecerelle UAV in Afghanistan in 2004 which was ‘unserviceable after only a few months due to crashes and airframe damage’ [73]. Additionally, in Iraq in 2003 British operations experienced a Phoenix UAV ‘casualty rate of over 25%’ [73]. In some cases ‘Shadow and Raven operators have posted reward notices on vehicles to encourage return in the event of one being downed by hostile fire or mechanical issues' [*B.7*].

- *Reliance on benign operational environment* - A key issue for the current generation of UAVs operating within Iraq and Afghanistan is their vulnerability to air defences. These platforms tend to be slow, visible to radar, un-manoeuvrable and defenceless. For example, in Kosovo 'a number of UAVs were lost to Serbian helicopters simply flying alongside & shooting at them with side machine guns' [73]. The Kosovo campaign saw UAVs 'suffering losses of ten times that of manned aircraft' [73]. Small low cost drones may be harder to shoot down with small arms fire, but are susceptible to fragmentation type devices (although these may have significant operational constraints, particularly at low levels where there may be a risk of injuring friendly troops or non-combatants on the ground).
- *Small UAV dependence on weather* - This restriction prevents these platforms from performing base protection tasks, as they cannot be relied on to always be useable [73].
- *Network systems integration issues* - Issues have been reported over the lack of integration of various systems resulting in targeting coordinates being manually transferred from one asset to another. There is a continual drive towards common standards to ease this issue.

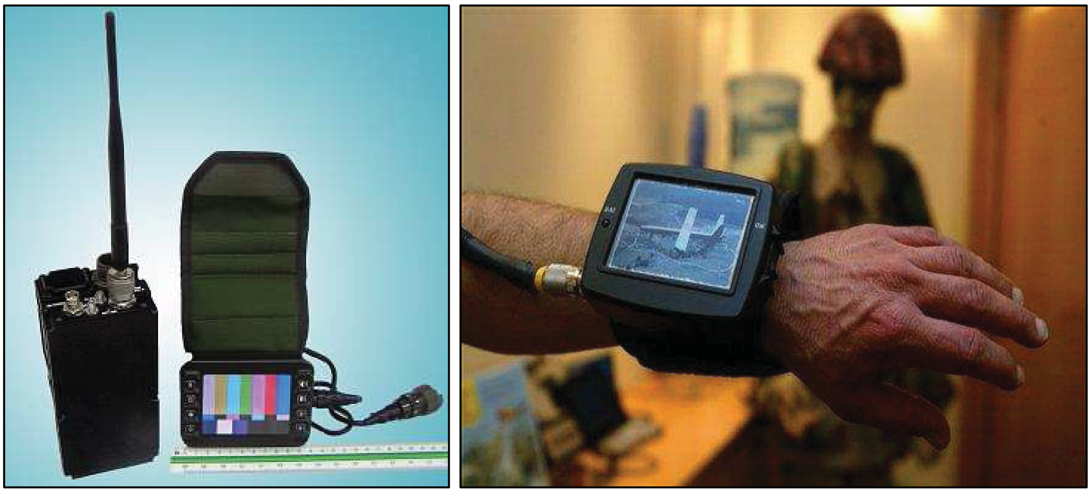


Figure 11 - Dismounted soldier access to real-time aerial imagery

3.2 Urban Wind Characteristics

The aim of this section is to briefly review the primary wind field characteristics that may be expected within urban environments. For small and micro air vehicles this flow field is particularly important as the effects are likely to be significant with regard to the maximum vehicle airspeed. In addition to informing system design and simulation based testing, an understanding of wind effects within this environment may also be used to develop CONOPS and enhance operator awareness of effects that may be encountered.

3.2.1 *Characteristic Flow about a Single Building / Structure*

The primary features of steady wind flow about a single building is reported in various papers [5, 50], with the main features illustrated in Figure 12. As the flow encounters the building a stagnation point is found approximately at 70% of building height where the flow divides to travel either above, to the side or down the front face of the building. It can also be seen that the wind flow separates over the top and sides of the building, resulting in a re-circulating flow region that causes a suction effect immediately behind the building. Depending on building depth (flow-wise) the detached flow on the roof may re-attach before separating again over the rear edge [50]. The downward flow on the windward face forms a vortex at ground level which moves to either side of the building and joins with the flow accelerating around the sides. A sheltered region is created downwind of the building which may extend a considerable distance [5]. It is also reported [50] that within the building wake a significant increase (greater than 20%) in turbulence intensity can be found. It is possible for the local flow around the sides and top of the building may be faster than the free-stream flow, with [5] reporting a flow at ground level of up to 2 or 3 times the free-stream.

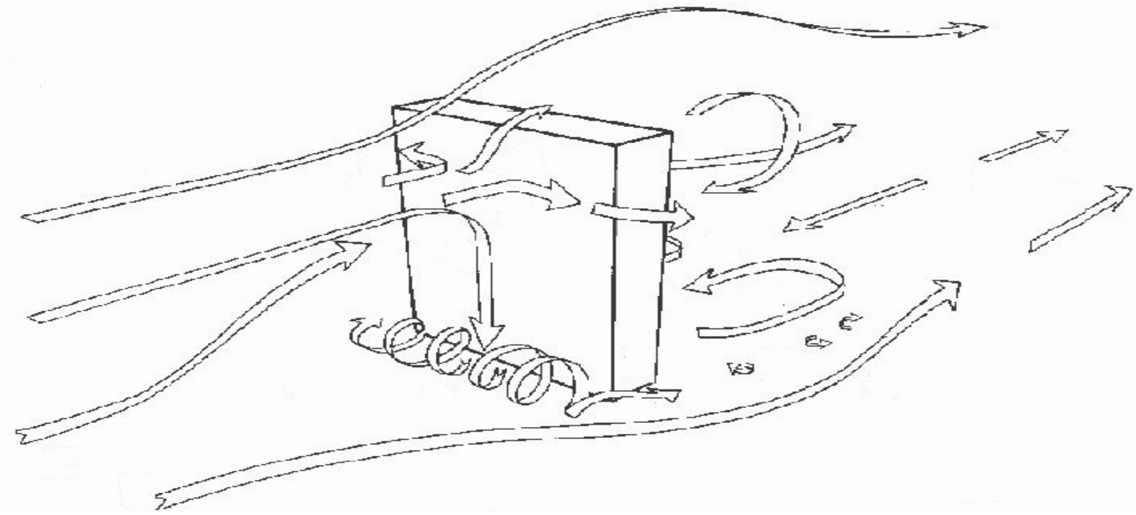


Figure 12 - Wind flow about a single building

Reproduced with permission from [5] Blocken & Carmeliet, "Pedestrian Wind Environment around Buildings: Literature Review and Practical Examples", Journal of Thermal Envelope and Building Science.

3.2.2 *Flow Interaction Between Buildings (2D Effects)*

The interaction of flow between two buildings is also reported by many authors [5, 48, 50, 35] with the primary interaction being the downward flow at the face of the second building reinforcing the re-circulatory flow behind the first building. This results in a standing vortex between the buildings, with the characteristic flow and wind speeds (relative to the free-stream) displayed in Figure 13. This image also displays both a small standing vortex at the base of the first building and the re-circulatory / suction region behind the second building.

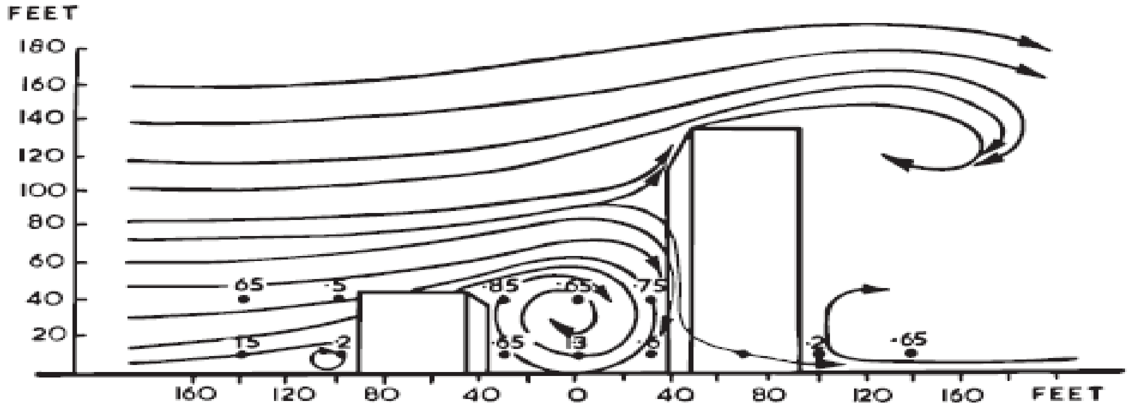


Figure 13 - Wake interference between two buildings

Reproduced with permission from [5] Blocken & Carmeliet, "Pedestrian Wind Environment around Buildings: Literature Review and Practical Examples", Journal of Thermal Envelope and Building Science.

The impact of the urban canyon aspect ratio (building height / canyon width) was investigated by Li et al [42] using a large eddy numerical simulation model. The main flow features are visualised via stream-functions in Figure 14 where it can be seen that the number of re-circulating flow regions increases with canyon depth. Each vortex is driven by shear forces from the flow above, therefore the strength of each vortex diminishes towards the ground. The mean horizontal and vertical wind speeds at peak levels within these vortices reduce from approximately 20% of free-stream towards the top of the canyon down to 0.1% towards ground level. These figures indicate the sheltering effect that can occur within deep canyons. Turbulence levels within the canyon, measured by dimensionless velocity fluctuations ($u''u''/U^2$ & $w''w''/W^2$) are reported [42] to reduce from approximately 0.01 at the top of the canyon to negligible values towards the bottom.

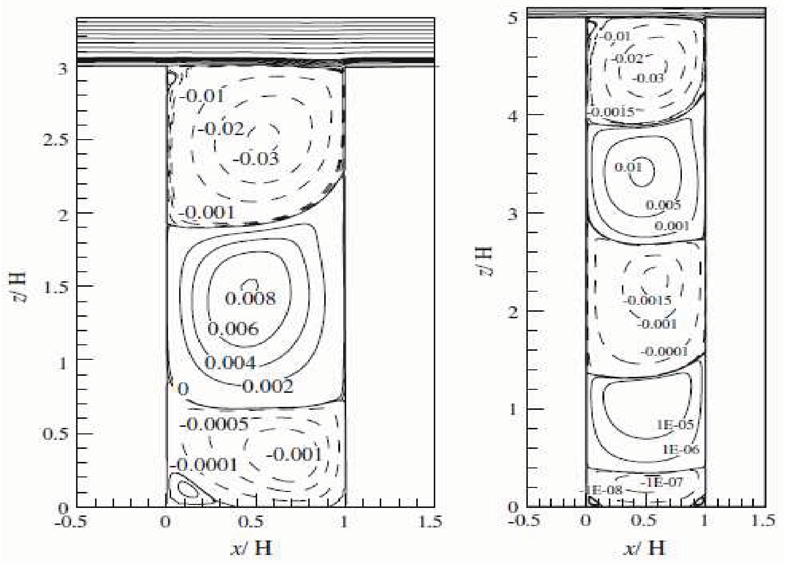


Figure 14 - Impact of canyon depth on flow field

Reproduced with permission from [42] Li, Liu & Leung, "Large-Eddy Simulation of Flow and Pollutant Dispersion in High-Aspect-Ratio Urban Canyons with Walls", Boundary-Layer Meteorology.

For a two dimensional wind flow perpendicular to several buildings the type of effect that can be expected is dependent on the geometry of the buildings, with the critical dimensions being building height (H) and canyon width (W). Three distinct flow regimes can be found, as reported by Oke [48], with each illustrated in Figure 15:

- *Isolated roughness flow (Approximately H / W < 0.3)* - Well spaced structures allowing the free-stream flow to recover before encountering the next structure.
- *Wake interference flow (Approximately 0.3 > H / W < 0.7)* - As the building spacing reduces the flow from one begins to interfere with the next, resulting in complex secondary flows.
- *Skimming flow (Approximately H / W > 0.8)* - Typically a stable vortex forms, with the majority of the flow not entering the canyon. As the canyon aspect ratio increases so does the likelihood of several vortices forming (see Figure 14).

Within dense medium to high rise city environments it is skimming flow that dominates, with the vertical and horizontal winds speeds highly variable within the canyon. Karatasou et al [35] report that downdraft velocities can peak at 95% of free-stream wind speed at heights of approx 3H / 4, while at H / 2 the downward speed can be close to zero. Updraft velocity is relatively independent of height, peaking at approximately 55% of free-stream wind speed [35]. Horizontal wind speeds within the canyon typically range up to 55% of the free-stream wind speed.

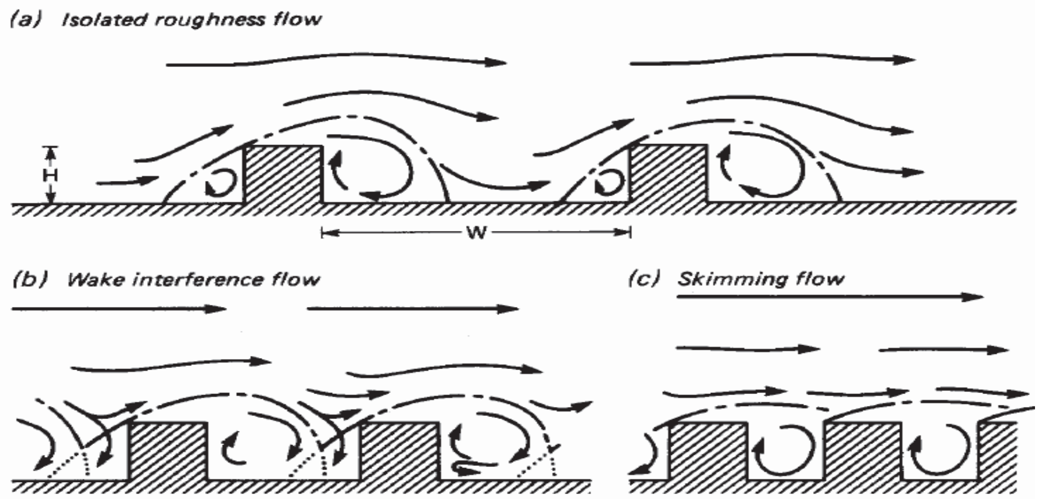


Figure 15 - Urban canyon primary 2D flow regimes

Reproducedwith permission from Oke [48] “Street Design and Urban Canopy Layer Climate”, Energy and Buildings

3.2.3 Three-Dimensional Effects

If the free-stream wind is not perpendicular to the buildings then a secondary type of flow occurs, known as channelling, where the wind is channelled down the urban canyons between rows of buildings. Woods et al [78] discuss oblique flow as a combination of channelling and the vortex creation discussed above, with the vortex driven down the canyon in a corkscrew motion by the channelling component [35]. This is illustrated in Figure 16 where a) displays perpendicular flow, b) channelling flow and

c) oblique flow. An example of wind speeds for oblique flow is provided by Karatasou et al [35] who report that for a 5m/s flow at 45deg to a canyon of H / W = 1 the maximum along and across canyon wind speeds were 0.6m/s and 0.8m/s respectively. The maximum vertical wind speed was 1m/s, with the flow stronger on the downward face (0.8 - 1m/s) than on the upward face (0.6m/s)

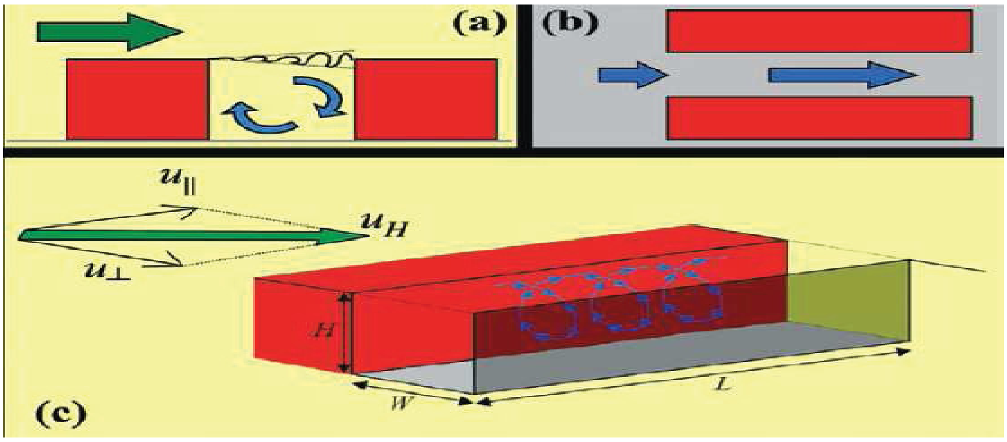


Figure 16 - Different 3D urban canyon flow regimes

Reproduced with permission from Wood & Co-authors [78] "Dispersion experiments in central London: the 2007 DAPPLE project". Bulletin of the American Meteorological Society

3.2.4 City Level Effects

The high level impact of an urban environment on the large scale wind flow is an effective increase in surface friction which extends the boundary layer higher than would be expected in a rural environment, as illustrated in Figure 17. For dense building arrangements skimming flow results, and the mean building height (h_{mean}) can be used to divide the lower urban boundary layer into three levels [57]:

- *Canopy layer (CL)* - Flow field below h_{mean} where the flow field is dependent on the geometry of the buildings as discussed in Section 3.2.1 to Section 3.2.3.
- *Roughness sub-layer (RS)* - Region immediately above the rooftops where the flow field is still highly dependent on the geometry of the buildings below. Extends from h_{mean} to between $2h_{mean}$ to $5h_{mean}$
- *Inertial sub-layer (IS)* - Region from the top of the RS to approximately 10% of the total boundary layer. Within this layer the flow field is horizontally homogeneous.

The impact of building geometry on turbulence intensity within the RS is discussed within [61] where at altitudes below $3.1h_{mean}$ significant fluctuations can be found in the horizontal plane. At the height $1.8h_{mean}$ the turbulence intensity is reported to vary between 0.3 – 0.5 depending on position. It is also reported that little is known about the importance of roof-tops or canyon region in the production of turbulence within the CL and RS [61].

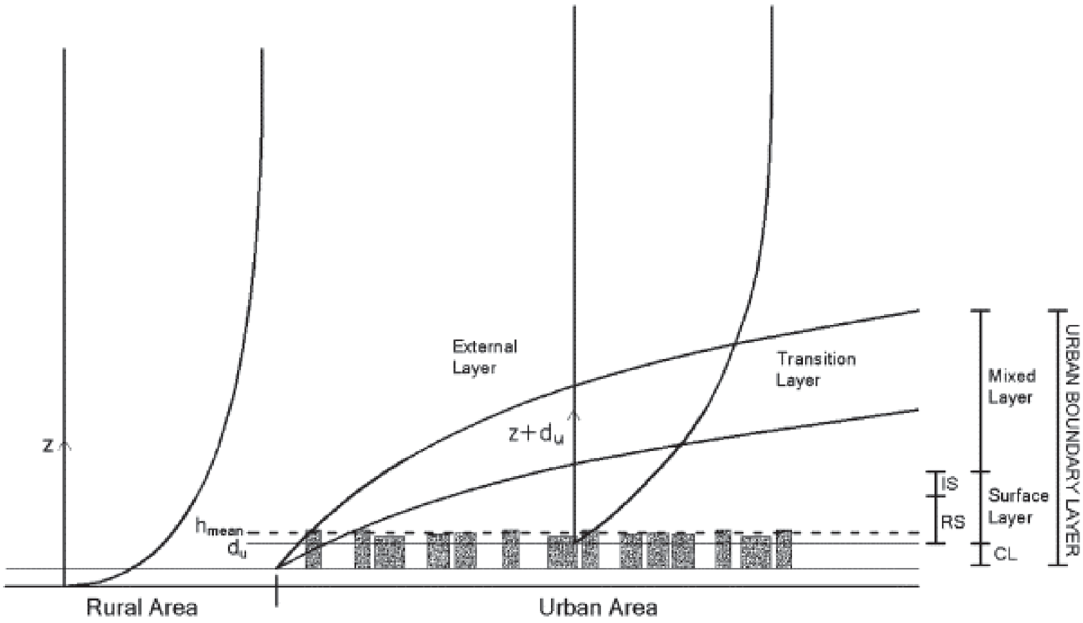


Figure 17 - Influence of the urban area on the large scale wind flow

Reproduced with permission from Ricciardelli & Polimeno [57] "Some characteristics of the wind flow in the lower urban boundary layer" Journal of Wind Engineering and Industrial Aerodynamics.

3.3 Motion Planning & Control of Unmanned Vehicles

A wide range of unmanned vehicle motion planning and control research can be found in the literature, much of it directed towards increasingly autonomous operation in complex environments. The range of architectural approaches proposed includes:

- *Static global planning + 'blind' trajectory tracking + event triggered sense-and-avoid capability*
- *Reactive global planning + 'blind' trajectory tracking*
- *Static global planning + 'situation-aware' reactive trajectory tracking*
- *Static global planning + 'situation-aware' receding horizon control (RHC) based trajectory tracking*
- *Goal based RHC using a cost to go map*
- *Direct feed of obstacle proximity sensors to control actuators*

Many different solution approaches have been used to implement these architectures, with results demonstrated in simulation and actual test on a wide range of unmanned vehicles. This work appears to be increasingly domain independent, with common approaches applied to both air and ground vehicles. A large amount of research can also be found on multiple vehicle deconfliction, for both manned and unmanned vehicles. Key issues include deconfliction horizon, intent prediction and sharing, centralised Vs decentralised, cooperative Vs uncooperative approaches.

The key research found in the literature as applicable to the work of this thesis is discussed further in the sections below.

3.3.1 Global Planning Algorithms

Voronoi graph based approaches have been used by [21] for UAV path planning while avoiding threat locations (e.g. surface to air missile defences) and by [28] for rotorcraft UAV flight through urban canyons. A Voronoi graph allows paths that are equidistant from known obstacles to be calculated, and therefore is well suited to urban canyon planning. However, as it is a purely geometric approach vehicle performance limits are not directly accounted for and the resulting paths may be far from either feasible of desirable in practice.

A library of feasible manoeuvres (all generated off line) is used by McConley et al [43], Singh et al [70] and Richards [60] in a framework known as an 'automaton' that can be used for rapid dynamically feasible path planning. The library is formed from a set of feasible trajectory primitives, which can be calculated off-line using a complex nonlinear vehicle model, therefore avoiding the need to simplify vehicle dynamics for on-line trajectory design. The on-line component is reduced to a graph based search (e.g. using an A* algorithm) allowing the most appropriate combination of manoeuvre to be chosen given the current situation. This work is appealing as it is aimed at enabling aggressive maximum performance manoeuvres, as well as transferring the problem of ensuring dynamic feasibility of the manoeuvres into the off-line calculation. However, difficulties may be encountered as the set of feasible manoeuvres to store, evaluate and choose from is effectively infinite.

Rapidly exploring random trees (RRT) have been successfully applied to path planning in complex environments by many researchers, including [77 & 81] for global planning and [74] for local motion planning. This method was first proposed by LaValle [40, 41] and progresses by applying feasible control steps to grow a trajectory tree from a current state towards randomly generated points. Complex performance constraints can be handled, with only feasible control options applied to the chosen point in the tree. Key properties of this approach include it's simplicity, it's ability to rapidly react to new information (by maintaining regions of the trajectory tree that are still feasible) and the bias of the search towards unexplored regions. The search can be either from the current vehicle state only, or it may be bi-directional, also beginning from the goal position. The performance of this approach is seen to be good, however, the random and somewhat jagged nature of the trajectory tree is not guaranteed to be optimal, and requires smoothing before use by a tracking algorithm.

Another reactive global planning approach is presented by Hrabar [29], this time based on a probabilistic roadmap and a D* Lite[21] graph search algorithm. The probabilistic roadmap generates the initial global trajectory by randomly sampling points in the environment, then joins them to neighbouring points if a visibility line exists between them. Knowledge of the obstacle space is formulated into a grid based occupancy map which can be updated on line by the on-board sensors. The resulting probabilistic

[21] Focussed dynamic lightweight adaptation of A*

network is then searched using the D* Lite algorithm, to select the preferred path. If on-board sensors detect an unexpected obstacle then the roadmap is updated, and importantly the D* Lite algorithm allows only the local path to be altered, therefore ensuring rapid operation.

A wide range of other approaches can be found for application to the large-scale or global trajectory planning problem, with commonly used methods including the graph based A* and branch & bound methods, visibility graphs, genetic algorithms, Dubin's curves. Each of these approaches has advantages and disadvantages. For example, Dubin's curves suffer from the fixed turn rate approximation, which is not realistic in the presence of wind and typically results in all manoeuvres being at maximum aggression. Additionally, the resulting path is discontinuous, making it infeasible and fundamentally unsuited to accurate vehicle placement in complex environments.

3.3.2 Receding Horizon Control

Shim and Sastry et al [65 - 69] have applied nonlinear model predictive control (NMPC) to situation aware trajectory tracking for rotorcraft UAVs. Their approach uses the traditional control space design, with vehicle performance constraints imposed using Lagrange multipliers and the resulting optimisation solved via a gradient based approach. Static and dynamic obstacles were modelled using the potential function approach, and avoidance was demonstrated in 2D and 3D simulations, as well as flight tests using a model rotorcraft UAV. The flight test results presented in [69] employ a complex model and a high design rate (up to 50Hz), but the horizon length is limited to 0.7secs, significantly impairing the benefit of the receding horizon approach. Additionally, although their design procedure produces both control signals and resulting output space trajectories, the control signals are not used, instead relying on a separate trajectory tracking layer [69].

Cowling et al [9 - 11] generate feasible and obstacle free receding horizon trajectories for a simplified model of a quadrotor UAV. A constrained optimization process (Matlab function *fmincon*) is used, based on the goal of moving from the current state to arrive at a goal state at a desired time. The continually designed trajectories, which therefore reduce in length over the course of the mission, are described using a set of 7th order polynomials, allowing a reduction in the search space as well as the enforcement of desired initial and goal states. As the design horizon decreases throughout the mission a separate time profile polynomial is required as well as the spatial description polynomials. A key element of this work is that the quadrotor model that is used exhibits differential flatness, therefore allowing an output space trajectory to be converted analytically into the required control signals. This simplifies the optimisation process as the goal position and obstacles all live in the output space, and also removes the need to design in the control space.

Another approach that has been commonly used for receding horizon control is mixed integer linear programming (MILP). Bellingham, Richards & How [3] employ MILP to enforce both dynamic manoeuvre constraints (speed / turn rate) as well as obstacle constraints in a 2D receding horizon framework. In this work the global plan is incorporated within the receding horizon framework by producing a grid of cost-to-go values across the operational environment. This allows various candidate terminal points of the receding horizon trajectories to be evaluated with consideration of the global situation, and helps to avoid the entrapment problem. This approach allows flexibility with the receding horizon trajectory design, but generation of the cost-to-go grid is likely to be computationally intensive and therefore potentially slow to respond to changes in the environment.

The above work was extended into 3D by Kuwata and How [39], where a visibility line problem is used to connect the end of the receding horizon trajectories to the cost-to-go map. Similar work can also be found in [64 & 58]. The complexity of MILP problems depends heavily on the number of binary variables, which in turn depends on the number of obstacles and constraints to be met. Typically the work found in the literature uses a commercially available solver (e.g. CPLEX) to perform the optimisation. MILP formulations result in complex optimisation problems, with the binary variables requiring a series of linear programming problems to be solved, therefore real-time implementation is likely to be challenging.

3.3.3 Global / Local Motion Planning Architectures

The Tartan Racing[22] entry into DARPA's Urban Challenge uses a continuous twin horizon planning architecture, with a strategic layer planning the road plan and a local motion planning layer responsible for tracking the road centre-line while avoiding static and dynamic obstacles. The local motion planner is based on an approach discussed in [26 & 27] where selection is made between a candidate set of resulting feasible trajectories. These trajectories are generated using a complex vehicle model, where feasible control sequences described by polynomials are shaped via a gradient search to arrive at a set of target postures[23]. Because a complex nonlinear vehicle model is used the calculated control sequence can be directly executed without requiring a tracking controller.

The Team MIT entry into DARPA's Urban Challenge [74] also uses a global / local motion planning division. The global, or strategic, layer plans road network routes using an A* algorithm that minimises time to destination. A modified RRT approach is then used to track the desired mission plan target waypoints, while accounting for local static and dynamic obstacles. This implementation demonstrates that an RRT can also be used for local motion planning, growing a tree of feasible trajectories at a rate of 10Hz using constraints from a complex nonlinear vehicle model.

Yang & Sukkarieh [81] employ a global / local planning architecture for fixed wing UAV operation in the presence of obstacles. A RRT approach is used to design the global trajectory which is then followed using a RHC formulation that is also able to account

[22] Tartan Racing, led by Carnegie Mellon University were the winning team in the 2007 DARPA Urban Challenge.

[23] In this context posture refers to the combination of position, orientation and rate.

for local obstacles. This work employs a high RHC design rate, but the RHC planning is limited to a short design horizon and only lateral dynamics.

The problem of an unmanned ground vehicle (UGV) tracking a path that is nominally clear of obstacles, but not guaranteed to be so was tackled by Hamner et al [23]. The global trajectory is created using an A* search, and the local tracking algorithm that is also responsible for avoiding unexpected obstacles is based on a variant of the 'pure-pursuit' approach, with the addition of repulsion forces for obstacle avoidance and attraction forces for returning to the trajectory. This single point tracking approach (as opposed to a receding horizon approach) is computationally simple, but unpredictable in complex scenarios, where the resulting command becomes a superposition of commands due to individual obstacles. This approach was also extended into 3D and applied to a rotorcraft UAV by Scherer et al [63] with the aim of allowing fast (3-10m/s) low level flight, using an on board laser scanner to detect obstacles. Many successful obstacle avoidance test flight tests were performed, demonstrating the utility of a single point situation aware tracking approach for simple scenarios.

An argument regarding a global / local motion planning algorithm for UGV operation in complex environments is also presented by Dippold et al [12]. The presented approach uses visual obstacle information and a repellent force to iteratively deform the global path around detected obstacles.

Yoon et al [82] use a variation of the approach discussed in [65 - 69] used to control a UGV in a complex environment. Potential functions are again used for obstacle avoidance, but this time based on parallax information, rather than proximity. Importantly, comparisons are made between the receding horizon approach and a reactive obstacle avoidance method, and it is demonstrated that the reactive approach is unable to cope with complex scenarios.

3.3.4 Visual Based Guidance

The use of reactive vision based guidance is also discussed in many papers, aiming to exploit the small, lightweight and passive nature of these sensors. Hrabar and Sukhatme [30] use optic flow to provide a centring response for urban canyon operation, and stereo vision to drive obstacle avoidance. Beyeler, Zufferey and Floreano [4] use optic flow to reduce the need for GPS and inertial sensors, as well as providing centring response, altitude hold and obstacle avoidance. This approach feeds visual sensor information directly to control actuators, providing a lightweight reactive architecture fundamentally suited to small vehicles. Key issues with optic flow include it's inability to detect obstacles directly ahead, and the reliance on contrast between obstacles. Additionally, the absence of global positioning information such as GPS or inertial systems suggest that this implementation is only suited towards safe-wandering behaviour, rather than achieving specific goals.

The relative nature of obstacle avoidance, in contrast to the global nature of path tracking, is emphasised by Andert et al [1], including the limitations of GPS and Inertial sensors for close proximity flight to obstacles known in advance. Instead, onboard visual sensors are used to identify known obstacles (in this case gates), then create new waypoints that allow the rotorcraft UAV to navigate relative to the gate, rather than using the global trajectory plan.

3.3.5 Multiple Vehicle Deconfliction

The 'free flight' concept of air traffic management and control is described by Hoekstra, van Gent and Ruigrok in [25]. In this approach, responsibility for aircraft separation is transferred from ground based air traffic controllers, to the pilots. This approach is enabled by each aircraft broadcasting ID, state and intent data, (e.g. using an ADS-B[24] type system) allowing receiving equipment on each aircraft to build up a picture of the local traffic situation. The primary benefit of this approach is that it enables direct routing, which can be significantly more efficient that the current approaches. Various conflict detection and avoidance strategies are discussed, including extended visual flight rules, vector product of velocity vectors, potential function methods, TCAS like manoeuvres, negotiation and priority.

Prandini, Lygeros and Sastry [54] discuss three different horizons of aircraft conflict detection as *Long-range (*Several hrs - Involves pre-flight approval at the level of the entire aerospace system), *Mid-range (*Tens of minutes - Involves in flight modifications of flight plans by air traffic control) & *Short-range (*Seconds to minutes - On board flight management system based conflict detection and avoidance, e.g. TCAS[25]).

Richards and How [59] demonstrated the use of MILP to perform centralised deconfliction of three and four 2D aircraft with constant speed and constant turn rate restrictions. Another centralised deconfliction strategy is proposed by Frazzoli et al [18], this time allowing individual aircraft to state their preferred speeds and headings. The issue of decentralised cooperative receding horizon trajectory sharing, and the impact of communication delays and failures is discussed by Izadi, Gordan and Zhang [32]. In the event of communication failures, knowledge of the performance limits of cooperative vehicles and the size of the communication delay is used to extend safety 'tubes' about the predicted trajectories of the other vehicles. Simulations are presented using simple 2D vehicle dynamics and a receding horizon on 3s.

Rather than using a proximity based approach, Gates [20] proposes basing deconfliction on the expected miss distance (EMD). The EMD is the distance between aircraft at the predicted closest point of approach. At this point a miss vector can be calculated, which is then used to formulate a simple decentralised deconfliction law. It is suggested that this approach 'greatly outperforms' instantaneous repulsion methods, primarily because it allows crossing paths and milder turns therefore being less likely to

[24] Automatic Dependent Surveillance - Broadcast

[25] Traffic Collision Avoidance System [15]. TCAS-II is mandated in Europe for all commercial transport aircraft with over 19 passenger seats, and in the USA for all commercial transport aircraft with over 30 passenger seats.

exceed vehicle performance limits. The EMD based deconfliction law has been successfully applied to short term and long term deconfliction, of two or more vehicles.

3.4 Optimisation Approaches

This section presents a brief review of some of the fundamental concepts and techniques of the field of optimisation. A more complete description may be found in [55], or many other texts. The aim of this review was to inform the development of a real-time constrained optimisation algorithm that is suitable for application to the defined local motion planning problem (Section 4.5).

3.4.1 Introduction & Overview

Optimisation is a mathematical field that is interested in finding the minimum or maximum values of a function of one or more variables. The function to be optimised, $C = f(X)$, is known as the objective (or cost) function, and the variables $X = (x_1, x_2, ..., x_m)^T$ are known as the design variables. Optimisation problems are commonly classified according to:

- *Constraints*: Linear, nonlinear, equality, inequality, etc.
- *Design variables*: Single variable, multivariable, real, integer, binary, etc.
- *Objective functions*: Linear, quadratic, general nonlinear, convex[26], non-convex, etc.

Although some very general optimisation techniques do exist (e.g. sequential quadratic programming) it is generally more efficient to apply an algorithm that is designed for a particular problem type. It is therefore often worth directing effort towards transforming the problem into a standard form, allowing reliable, efficient and commonly available algorithms (e.g. within Matlab or other commercial mathematical packages) to be used. Examples of standard problems include:

- *Linear Programming (LP)* - Both the objective function and all constraints are linear. The standard form of a LP problem is:

  Minimise

  $$f(X) = c^T X \tag{3-1}$$

  Subject to the following constraints: $AX \le b$, $X \ge 0$

- *Quadratic Programming (QP)* - The objective function is quadratic and the constraints are linear. The standard form of a QP problem is:

  Minimise

[26] A set is convex if for every pair of points in the set the line joining them is always inside the set.

$$f(X) = c^T X + \frac{1}{2} X^T D X \qquad (3\text{-}2)$$

Subject to the following constraints: $AX \leq b$, $X \geq 0$

- *Mixed Integer Linear Programming (MILP)* **-** With this type of problem the objective function and constraints are all linear combinations of the design variables, but the design variables are a combination of real-valued and integer-valued. A common source of integer variables in linear problems is constraints that take an *either / or* form, rather than the more usual *and* form (where all constraints must be satisfied simultaneously).

For the more general case of non-standard problems there are two primary approaches for optimising a function:

- *Classical* - With this approach a set of necessary and sufficient conditions for the optimal design point are first derived. These conditions are then solved (numerically if necessary) to calculate the optimal design. The necessary and sufficient conditions are derived using differential calculus, and therefore the objective function must be continuous and differentiable.
- *Numerical* - Numerical methods are based on using repeated evaluations of the objective function to direct variations in a current design point towards the optimal values.

A key issue for optimisation problems and solutions is that of complexity and tractability. The tractability of a problem, or more accurately the tractability of the chosen solution approach to a problem, refers to the growth in solution time (t_s) with growth in design space (n). If the relationship between t_s and n is polynomial in n, then the problem can be considered tractable. However, if the relationship is exponential (e.g. if t_s grows exponentially with n) or worse, then the problem can be considered intractable. Tractability is therefore not a measure of solvability, it is a measure of complexity growth with problem dimension.

For many complex problems the solution time may be reduced by the use of heuristics. A heuristic is a *rule-of-thumb* or an *educated guess* that helps to reduce the overall search time by potentially sacrificing a degree of optimality. Heuristics can be highly problem specific, based on the designer's understanding of the search space. For example, with a UAV guidance problem the optimisation algorithm may initially hold speed or altitude constant, allowing a more complete exploration of the remaining design space.

Some of the key issues mentioned above are discussed further in the following sections.

3.4.2 *Standard Constraint Forms*

In most real-world applications the design variables are subject to certain constraints, for example vehicle performance limits or finite resources, and these must be explicitly accounted for in the optimisation techniques. The two most common forms of constraints are:

- Equality constraints: $g_i(X) = 0$ for i=1,2,…,n
- Inequality constraints: $g_i(X) \leq 0$ for i=1,2,…,n

Certain standard problems and techniques require constraints in equality form, and inequality constraints can be converted into equality constraints. For example:

$g(X) \leq 0$ can be represented as $g(X) + y^2 = 0$ by the introduction of a new nonnegative *slack*[27] design variable (y^2). This does, however, introduce an additional design variable for each transformed constraint.

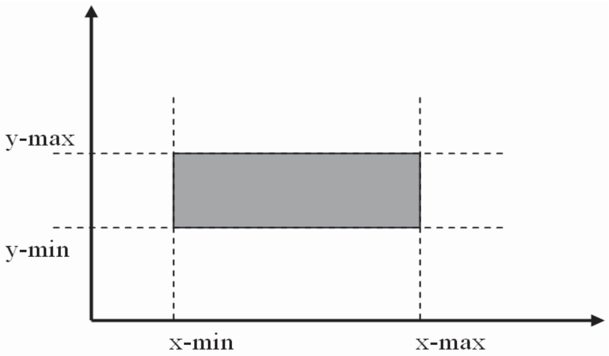


Figure 18 - Example 'Either / Or' Constraints

Typically, a set of constraints will need to be simultaneously enforced, known as *and* constraints. However, in certain circumstances constraints may take an *either / or* form where at least one of a set of two or more constraints must be enforced at all times. An example of *either / or* constraints is the obstacle avoidance problem shown in Figure 18, where a 2D obstacle constraint is shown in grey. For path planning purposes, this obstacle avoidance constraint may be written as shown in equations (3-3) to (3-6), where only one of the four constraints needs to be met to avoid the collision.

$$y < y_{Min} \tag{3-3}$$

$$y > y_{Max} \tag{3-4}$$

$$x < x_{Min} \tag{3-5}$$

$$x < x_{Max} \tag{3-6}$$

[27] The term s*lack variable* refers to the use of the variable to 'take up the slack' when required, ensuring that equality constraints are met when necessary.

These constraints can be converted into *and* constraints (easier to apply) by the introduction of a very large constant (*M*), new binary design variables ($k_1,...,k_4$) and a new constraint. The original constraints can be re-written:

$$y < y_{Min} + k_1 \times M \qquad (3\text{-}7)$$

$$y > y_{Max} - k_2 \times M \qquad (3\text{-}8)$$

$$x < x_{Min} + k_3 \times M \qquad (3\text{-}9)$$

$$x > x_{Max} - k_4 \times M \qquad (3\text{-}10)$$

It can then be ensured that at least one constraint is active at all times by imposing the following new constraint on the values of the $k_1,...,k_4$:

$$k_1 + k_2 + k_3 + k_4 \leq 3 \qquad (3\text{-}11)$$

Constraints are called linear if they can be written as a linear combination of the design variables. For example, a linear inequality constraint may be written:

$\sum_{i=1}^{m} k_i x_i \leq 0$, $where \;\; k_i, \;\; i = 1,2,...,m$ are constants.

Nonlinear constraints involve a nonlinear combination of the design variables. Methods of handling constraints are addressed in the following section which discusses classical and numerical optimisation techniques.

3.4.3 Classical Methods - Unconstrained

Classical optimisation approaches use differential calculus to derive a set of necessary and sufficient conditions for the existence of minimum or maximum points. Once these conditions are known, then they may be solved directly to obtain the optimal design variables. The objective function is therefore only evaluated once, at the end of the process, after the optimal design has already been calculated. This has obvious benefits for problems where the objective function is complex to evaluate.

Necessary Conditions

The necessary condition for an optimal value of an unconstrained objective function is that it is a stationary point, i.e. the first derivative of the objective function (with respect to all design variables) is zero. Stationary points may be either i) Local & global minimum / maximum, ii) Point of inflection or iii) Saddle point (2 or more design variables).

Sufficient Conditions

For 1D problems (single design variable) it is possible to differentiate between min, max & points of inflection by investigating the higher order derivatives of the objective function (with respect to the design variable):

If $f'(x) = f''(x) = f^{n-1} = 0, \ \ but \ \ f^n \neq 0$ then:

(i) $f(x)$ is a relative *minimum* if $f^n(x)$ > 0 & n is even

(ii) $f(x)$ is a relative *maximum* If $f^n(x)$ < 0 & n is even

(iii) neither a *minimum* or *maximum* if n is odd

For multivariable problems the nature of a stationary point can be determined by examining the second derivatives of the objective function with respect to the design variables. This will take the form of an *n*x*n* matrix (for n design variables) known at the Hessian matrix.

(i) If H is positive-definite then the stationary point is a *minimum*.

(ii) If H is negative-definite then the stationary point is a *maximum.*

3.4.4 Classical Methods - Constrained

For constrained problems the previous necessary and sufficient conditions may not be applicable as it is possible that a design point will exist on a constraint boundary, and not at a stationary point.

Direct Substitution

For simple objective functions and constraints it may be possible to rearrange the constraint equations into a form that allows direct substitution into the objective function. This results in a new unconstrained objective function, allowing solution by the previous necessary and sufficient conditions.

Constrained Variation

The aim of this approach is to develop an analytical expression for the necessary conditions that accounts for the constraints. Again inequality constraints must first be converted into equality form. A necessary condition can be developed as follows, where $x_1 \& x_2$ are the design variables, the objective function is $f(x)$, and the constraint is $g(x) = 0$.

At a stationary point: $$df = \frac{df}{dx_1}dx_1 + \frac{df}{dx_2}dx_2 = 0 \qquad (3\text{-}12)$$

The equality constraint can be enforced by constraining variations in the design variables to maintain $g(x) = 0$. A Taylor series expansion of the constraint function about the optimal point (x^*) can be used to determine changes required in one variable due to variation of the other:

$$g(x) \approx g(x^*) + \frac{dg}{dx_1}dx_1 + \frac{dg}{dx_2}dx_2 = 0$$

Now, at the optimal point $g(x^*) = 0$ (to meet the constraint), resulting in:

$$\frac{dg}{dx_1}dx_1 + \frac{dg}{dx_2}dx_2 = 0 \tag{3-13}$$

Given a defined change in x_1, equation (13) defines the required change in x_2 to maintain the constraint. Substituting (13) into (12) and rearranging results in the following necessary condition for x^* to be a constrained stationary point.

$$\frac{df}{dx_1} - \frac{dg/dx_1}{dg/dx_2} \times \frac{df}{dx_2} = 0 \tag{3-14}$$

Necessary conditions for a general problem can be developed in a similar way.

Lagrange Multipliers

This method may be used to convert a problem of *m* variables and *n* constraints into an unconstrained problem of *m+n* variables. Again, the method applies only to equality constraints, therefore other forms will first need to be converted as previously discussed. The unconstrained problem is defined by the Lagrange equation:

$$L(X,\lambda) = f(X) + \sum_{i=1}^{n} \lambda_i g_i(X) \tag{3-15}$$

Where the new variables λ_i are known as Lagrange multipliers. The Lagrange equation can then be solved by applying the usual unconstrained necessary conditions:

$$\frac{\partial L}{\partial x_i} = 0 \quad for \quad i = 1,2,...,m$$
$$\frac{\partial L}{\partial \lambda_j} = 0 \quad for \quad j = 1,2,...,n$$

Resulting in the following set of necessary conditions for the constrained optimal point:

$$\begin{aligned} &\frac{\partial f(X)}{\partial x_i} + \sum_{i=1}^{n} \lambda_i \frac{\partial g_i(X)}{\partial x_i} = 0 \quad for \quad i = 1,2,...,m \\ &\frac{\partial L(X,\lambda)}{\partial \lambda} = g_j(X) = 0 \quad for \quad j = 1,2,...,n \end{aligned} \tag{3-16}$$

Kuhn-Tucker (KT) Conditions

A more general set of necessary conditions for optimal design points is given by the Kuhn-Tucker conditions, which are also derived using Lagrange multipliers. The KT conditions form the basis of the highly general and successful sequential quadratic programming (SQP) approach. An example implementation of this method is the Matlab function *fmincon*.

3.4.5 General Numerical Approach

Numerical optimisation methods are based on using repeated evaluation of the objective function to direct a starting design towards the optimal point. They are therefore iterative in nature, with an example algorithm as follows:

1. Select a starting design X_i
2. Find a suitable search direction S_i
3. Find a suitable step size k_i in the chosen direction
4. Calculate the next design point using: $X_{i+1} = X_i + k_i S_i$
5. Evaluate the objective function at the new current design and test for optimality (e.g. is the change in value of the objective function greater than a minimum termination tolerance). If yes, then terminate algorithm, otherwise return to step-2.

This approach has two main phases for each iteration i) the choice of search direction and ii) the choice of step size. For 1D problems the only direction choice is between a positive or negative step. For multivariable problems once a search direction has been defined, the problem has effectively been reduced to a 1D problem, with the remaining variable being the step-size in the chosen direction.

If there is any overhead associated with the calculation of S_i, then rather than taking a single step then re-calculating the preferred direction, a line search may be preferred. A line search is an iterative 1D search using the specified direction. The line search can be terminated either after a set number of steps, or when it has converged. The search direction is then recalculated at the new design point for a new iteration.

Various approaches to these problems are presented in the following sections.

3.4.6 Numerical Methods - 1D Problems

The two main categories of 1D numerical strategies are elimination and interpolation techniques, both of which are discussed briefly below.

Elimination methods

With these approaches the search progresses by eliminating individual design points or areas of the design space one at a time. For unconstrained problems, options include fixed and accelerated step sizes. Selection of a fixed step size is a balance between accuracy and efficiency, with small steps resulting is a large number of iterations before the optimum design is found. Using an accelerated step size helps to overcome this issue, with the size of the step increasing with each iteration until the objective function changes in the wrong direction. If the search area is constrained then more general approaches may be used, such as the dichotomous search, interval halving or the Fibonacci method. Care must be taken when applying these methods as they are all susceptible to local minimums.

Interpolation methods

Interpolation methods are based on using a curve-fitted approximation to the objective function about the current design state. This local approximation can then be used along with classical optimisation techniques to find the optimal step size from the current design. The optimal step size can then be used to generate a new design state as discussed in section 3.4.5. An example of this is shown below using a quadratic curve:

Use $h(\lambda) = a + b\lambda + c\lambda^2$ as an approximation of the objective function around the current design point. The constants a, b & c can then be calculated by evaluating the objective function at 3 values of λ. The optimum step size (λ^*) can then be estimated by using the following necessary condition:

$$\frac{dh}{d\lambda} = b + 2c\lambda = 0, \Rightarrow \lambda^* = -b/2c \qquad (3\text{-}17)$$

Another popular interpolation method is Newton's method. This approach uses a Taylor expansion of the objective function around the current position, then applies necessary condition ($\partial f / \partial \lambda = 0$) to get the optimum point. This can be applied iteratively using:

$$\lambda_{i+1} = \lambda_i - \frac{f'(\lambda_i)}{f''(\lambda_i)} \qquad (3\text{-}18)$$

This method uses second order derivative information to improve the performance of a gradient search, and can result in very fast convergence. However, care must be taken as if the initial position is not close to the solution then it can diverge. A disadvantage of this approach is that the second order derivative information may either be unavailable of hard to obtain. If the derivative terms are hard to obtain then a quasi-Newton method may be used, where they are estimated using a finite difference approach.

3.4.7 Numerical Methods - Unconstrained Multivariable Problems

There are two distinct approaches to solving unconstrained nonlinear multivariable optimisation problems, non-gradient based & gradient based. Typically gradient based methods are more efficient, however, if the objective function is not differentiable, or the effort required to generate the gradient information (1^{st} & 2^{nd} order) is prohibitive then non-gradient methods may be appropriate.

Examples of non-gradient based methods include:

- **Grid search** - Only possible for unconstrained problems if bounds are available on the design variables. Will become prohibitively large if there are a large number of design variables. However, a coarse grid may be used to provide a good starting point for a local search technique, therefore increasing the chance of finding a global minimum.

- **Univariate search** - Perform a series of 1D searches varying only a single design variable at a time. This is a simple approach, but results in slow convergence.
- **Pattern directions** - This approach aims to improve on the univariate search but allowing more efficient search directions that are not parallel to the design axes. An examples of a pattern search algorithm is Powell's methods.
- **Conjugate directions** - The aim of a conjugate direction search is to prevent a 1D search in one iteration from spoiling the work done by the 1D search of the previous iteration. Conjugate direction methods can be very efficient, and are able to minimise a quadratic function in a finite number of steps. General nonlinear functions can be approximated by quadratic functions near the solution, therefore using conjugate directions is expected to speed up convergence of any function. Conjugate directions may be determined using a variation of the pattern search method described above. An example of this is Powell's method.

Examples of gradient based methods include:

- **Steepest Descent** - Probably the most commonly used method of optimisation. Care must be taken as the direction of steepest descent is a local property, therefore this approach can sometimes be slow to converge. The direction of steepest descent is given by:

$$-\nabla f, \qquad \text{where } \nabla f = \left(\frac{\partial f}{dx_1}, \frac{\partial f}{dx_2}, \ldots, \frac{\partial f}{dx_n} \right) \tag{3-19}$$

- **Fletcher-Reeves Method** - This is a conjugate gradient method where the search direction at iteration *i* is found from:

$$S_i = -\nabla f_i + \frac{\nabla f_i^{\,T} \nabla f_i}{\nabla f_{i-1}^{\,T} \nabla f_{i-1}} S_{i-1} \tag{3-20}$$

3.4.8 Numerical Methods - Constrained Multivariable Problems

There are two broad types of constrained optimisation approaches, *direct methods* (those that handle the constraints explicitly as part of the algorithm) and *indirect methods* (the constraints are handled as part of a series of unconstrained problems). Examples of both are briefly discussed below.

Direct Methods

- **Sequential Linear Programming** - This method uses a series of LP problems to optimise a nonlinear problem. The LP problem is calculated by generating a first order Taylor expansion of the nonlinear objective function and constraints at the current point in the design space. The solution of the LP problem can then be tested for convergence. If it has not converged to a suitable level then the process is repeated at the new design point.

- **Method of Feasible Directions** - This approach uses the standard nonlinear iterative search technique ($X_{i+1} = X_i + k_i S_i$, see section 3.4.5) but also performs checks to ensure that the search direction is both feasible (doesn't violate any constraints) and useable (moves the value of the objective function in the desired direction). A search direction *S* is feasible if it satisfies the following equation for all constraints:

  $$S^T \nabla g_i(X_i) \leq 0 \tag{3-21}$$

  A search direction is useful if at least one component moves in the same direction as the direction of steepest descent, therefore:

  $$S^T \nabla f(X_i) < 0 \tag{3-22}$$

  At each iteration equations (3-21) & (3-22) can be tested to ensure that the chosen direction is both feasible & useful.

  There are two critical steps in this process:

  1. *Direction finding problem* - If all the constraints are inactive then any suitable direction can be used. However, if at least one constraint is active then we must test (3-21) to ensure that the chosen direction will not violate the constraint. If it does, then it must be updated to a feasible direction. Ideally, the solution is to project the desired direction onto each of the active constraint surfaces, which should lose the infeasible directions and keep the feasible ones. However, this process appears not to be simple. Additionally, there is likely to be several useable & feasible directions to choose from, resulting in a further optimisation problem. The method of feasible directions formulates this as a LP problem & uses the simplex method to find the direction of steepest descent that is feasible given the constraints.
  2. *Step size problem* - Any suitable 1D optimisation algorithm can be used, however, care must be taken not to violate any of the constraints.

Indirect Methods - Penalty Functions

The basic approach here is to replace constraints by additional terms in the objective function which can then be optimised as an unconstrained problem. The key to the popularity of these approaches is that it allows constrained problems to be solved using any one of the many unconstrained algorithms, e.g. conjugate gradients. The primary downside however, is that it significantly complicates the objective function, increasing the chance of getting stuck in local minimums. The applicability of these approaches is heavily dependent on the problem and in particular the constraints being imposed.

There are two classes of penalty function methods, interior & exterior methods:

- **Interior Penalty Function (barrier) methods** - A penalty term is designed such that it ‘blows-up’ as the constraint boundary is approached. All solutions generated at each iteration will lie in the *interior* of the original constrained design space.
- **Exterior Penalty Function Methods** - Allow the augmented objective function to violate the original constraints (e.g. *exterior* to the constrained search space). However, at each iteration the constraints should be violated less and less until the solution converges on the constrained optimum.

4 Problem Scoping, Exploration & Definition

4.1 High Level Military Requirement

One of the key high-level military requirements that emerged from the review of UAV operational experience discussed in Section 3.1 is the need for an aerial reconnaissance and surveillance capability operated directly by small units of dismounted soldiers. The primary driver for this requirement is a potentially time-critical need for additional situation awareness that is not satisfied by currently available systems. Although real-time aerial video can now be provided to ground troops via systems like ROVER (Figure 11), this causes a reliance on high value assets (such as manned aircraft, Predator, Global hawk etc.) that are operated from higher levels of command and therefore may be required for other duties. This lack of availability is compounded by airspace and communication frequency congestion issues (B.1, B.2 & B.3) which result in advance deconfliction requirements and consequently slower reaction to the time-critical need.

Small and micro UAVs that are light enough to be man portable and cheap enough to be widely distributed offer a viable alternative. In addition to removing the reliance on external support, small vehicle size and low altitude operation potentially ease airspace coordination issues helping to increase operational flexibility and therefore availability.

The overarching military requirement that drives the technical work of this thesis is therefore:

‘The use of man-portable small or micro UAV systems to allow small units of dismounted ground troops to perform their own aerial reconnaissance and surveillance without relying on external support’

Although some such systems are currently both in service and highly popular (e.g. Raven, Wasp, gMav, Desert Hawk - Figure 3 & Figure 10), several limitations exist, including:

- *Restricted to relatively simple operational environments* - Current systems are designed for operation both well clear of obstacles and in airspace that is de-conflicted from other traffic in advance of the mission.
- *Manual control* - Continual operator supervision and control is required. This is particularly undesirable for dismounted troops operating in hostile environments where the loss of mobility and immediate location situation awareness may be counter-productive.
- *LOS communication links* - This is another limiting factor for operation within complex environments where LOS is significantly restricted.
- *Centralised advance deconfliction* - Airspace and command frequency congestion results in less operational flexibility and slower reaction time to evolving situations.

- *System size* - Man transportable systems such as Raven still prove to be a significant burden for a dismounted soldier.
- *Single vehicle operation* - Current systems are operated as a single platform per operator, therefore losing potential benefits of multi-vehicle cooperation.

4.2 Context Exploration & Definition

The goal of this stage was to generate additional contextual information regarding the defined high level military requirement. This information was subsequently used to aid definition of a specific technical problem to be addressed (Section 4.5), and inform test and demonstration scenarios. This work is a key part of the systems engineering effort, focussing on early stage problem definition and exploration in order to uncover requirements and hidden issues.

Key information generated includes:

- *Characteristics of typical complex urban environments*
- *Definition & exploration of potential mission profiles*
- *Definition of capability visions*
- *High level system description*

Each of these areas is discussed further in the next section.

4.2.1 Urban Environment Characteristics

Effort has been directed within this research towards a greater understanding of potential operating environments, including typical building spacing, realistic obstacles fields and urban wind characteristics. Key information that was obtained includes:

- *Examples of additional urban clutter* - In addition to the obvious range of buildings, urban environments are complicated by the presence of many other obstacles, most of which are unlikely to be present in an *a priori* 3D obstacle map. Common examples include trees, overhead cables (domestic power, telecom, tram / rail), roof-top aerials, flag poles and flags / banners, signs / billboards (both on top of buildings and on stand-alone structures), building cranes, transmitter towers (e.g. extensions on roof-tops or larger ground based towers), mobile phone masts, chimneys, lampposts, traffic lights, etc. A range of examples of this type of urban clutter are shown in Figure 19 and Figure 20. It is clear that a comprehensive *a priori* map of the environment is not feasible, therefore enforcing the requirement for on-board obstacle detection, mapping and avoidance. An additional benefit of being able to handle obstacle detection and avoidance in real-time is that it significantly reduces the level of detail required in the *a priori* obstacle map, both simplifying generation of the map an reducing memory requirements.
- *Urban wind characteristics* - A review of the primary characteristics of steady and unsteady wind flows around urban structures was presented in Section 3.2.

- *Complicating factors of military operation in urban environment* - The primary complication results from the complex layout of the urban structure limiting both field-of-view and the field-of-fire. This reduces both situation awareness and the ability to react to hostile actions. The rules of engagement are also likely to be restrictive due to the presence of civilians, as well as a desired to limit collateral damage. The complex environment also restricts the availability / reliability of the GPS system, as well as causing line-of-sight communications problems, e.g. multi-path effects. The likely presence of civilians also adds a critical challenge in identifying combatants from non-combatants. Additionally, the urban environment provides many opportunities for concealment (of both explosives and hostile forces), including buildings, rooftops, basements, subway stations and tunnels, sewage pipes, maintenance corridors etc.

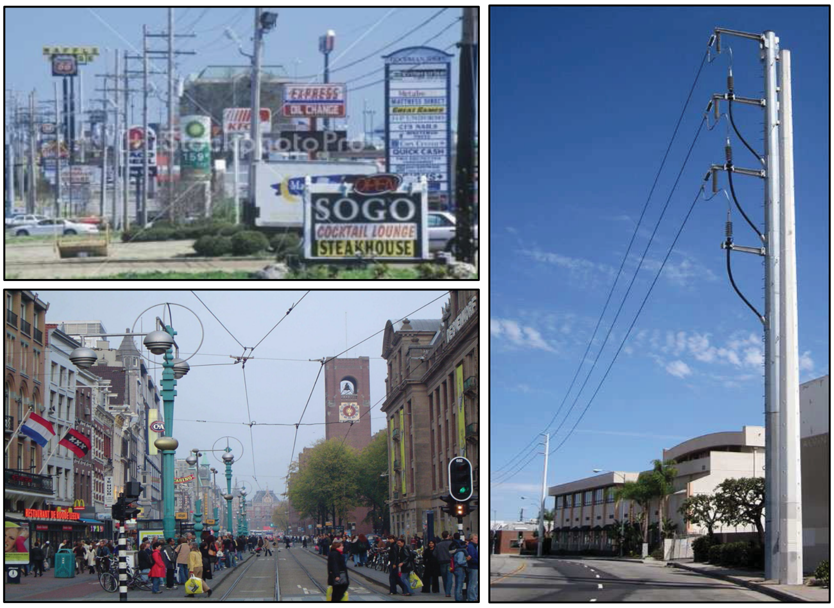


Figure 19 - Examples of overhead cables commonly found in urban environments

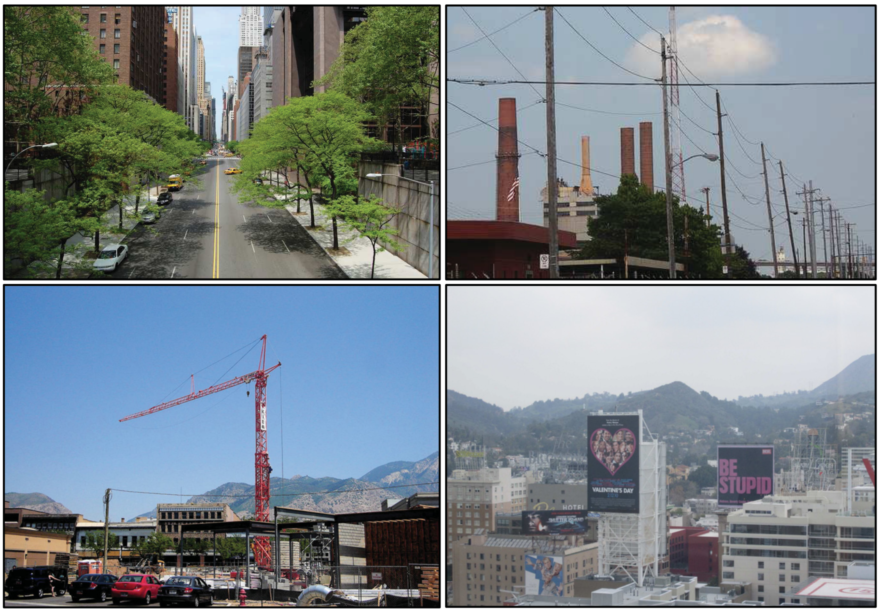


Figure 20 - Range of non-building urban clutter

4.2.2 Mission Profiles

The aim of this stage was to define and further explore several realistic and desirable mission profiles for the use of a system of one or more small / micro UAVs within complex urban environments. These proposed missions were derived both from the limitations of current systems, and consideration of additional capabilities and operational uses that may be enabled by the use of developing technology (i.e. miniaturised sensors, electronics, image processing, real-time mapping, vehicle localisation, control & design of small / micro air-vehicles etc.). Detailed descriptions are provided in appendix C for the following missions:

- *Time critical aerial reconnaissance & surveillance (C.1)*
- *Kamikaze small / micro UAV with explosive payload used to engage a beyond line of sight (BLOS) target (C.2)*
- *Urban area reconnaissance & surveillance (C.3)*
- *Patrol Escort (C.4)*
- *Kamikaze small / micro UAV with explosive payload used to destroy an enemy UAV (C.5)*

4.2.3 Capability Visions

Capability visions are used to summarise system capabilities / technologies that are implied by the military requirement and proposed mission profiles.

- **CV-1:** Reduced UAV system size to the point where a single dismounted soldier is able to routinely carry two or more small / micro UAVs.
- **CV-2:** Immediate unplanned small / micro UAV response to evolving or unexpected situations enabled by a greater reliance on in–flight decentralised vehicle deconfliction, rather than the current pre-flight centralised approach.
- **CV-3:** Sustained autonomous small / micro UAV flight within urban canyon to enable launch and recovery from current operator location, enhanced vehicle stealth, desired sensor line-of-sight, payload delivery etc.
- **CV-4:** Increased vehicle autonomy with the aim of reducing operator workload to levels suitable for dismounted soldier operating within hostile environments. Mission / task based operator interface, delegating motion planning, control, coordination functionality to the air-vehicle and system. Long term goal of increasing small and micro UAV autonomy towards *associate* level, where the system is able to provide continual utility without requiring continual control or supervision.
- **CV-5:** Coordinated use of multiple small / micro UAVs to maintain line-of-sight communication links within complex environments. This is again critical to avoid relying on external support (e.g. HALE UAV stationed above area of operation, or

satellite link). One or more air vehicles may be used to perform mission duties while the others are used as communication relays.

- **CV-6:** High density small / micro UAV operation within the same complex environment enabled again by a greater reliance on in-flight decentralised deconfliction, rather than the current pre-flight centralised approach. Reduced vehicle size and increased in-flight flexibility result in an increased chance of multiple vehicles, or multiple systems of multiple vehicles, operating within the same complex environment (e.g. several units, each operating their own system). As before speed of response and operational flexibility is critical.
- **CV-7:** Use of a small or micro UAV with an explosive payload optimised to perform a kamikaze type attack on enemy target. UAV effectively becomes a low-cost man-portable guided weapon, enabling individual ground troops to target enemy forces that are either obscured from sight (e.g. behind neighbouring buildings or on nearby rooftops) or difficult to hit (e.g. enemy small / micro UAV).

4.2.4 High Level System Overview

The previously discussed military requirement, proposed mission profiles and capability visions were used to create a single high level system overview. The intention of this was to both rapidly convey the primary system uses and to encourage consideration of both sub-system and super-system issues than may impact on either the technical problem or test scenarios.

The primary user based capabilities that is offered by a small / micro UAV based system are:

- *Enhanced soldier survivability* - Provided via improved situation awareness due to the provision of a high availability BLOS reconnaissance and surveillance capability.
- *Enhanced soldier effectiveness* - Provided via the ability to both engage and track BLOS targets at both ground and roof-top levels.

Key system use-cases are displayed in an operation concept graphic (MODAF OV-1) shown in Figure 21. This graphic attempts to convey in a simple format the basic elements of the system, e.g. one or more small / micro UAVs providing an enhanced BLOS capability to a dismounted operator within an urban environment. A preference for a head-up interface is also displayed by the image of the soldier with a helmet mounted display.

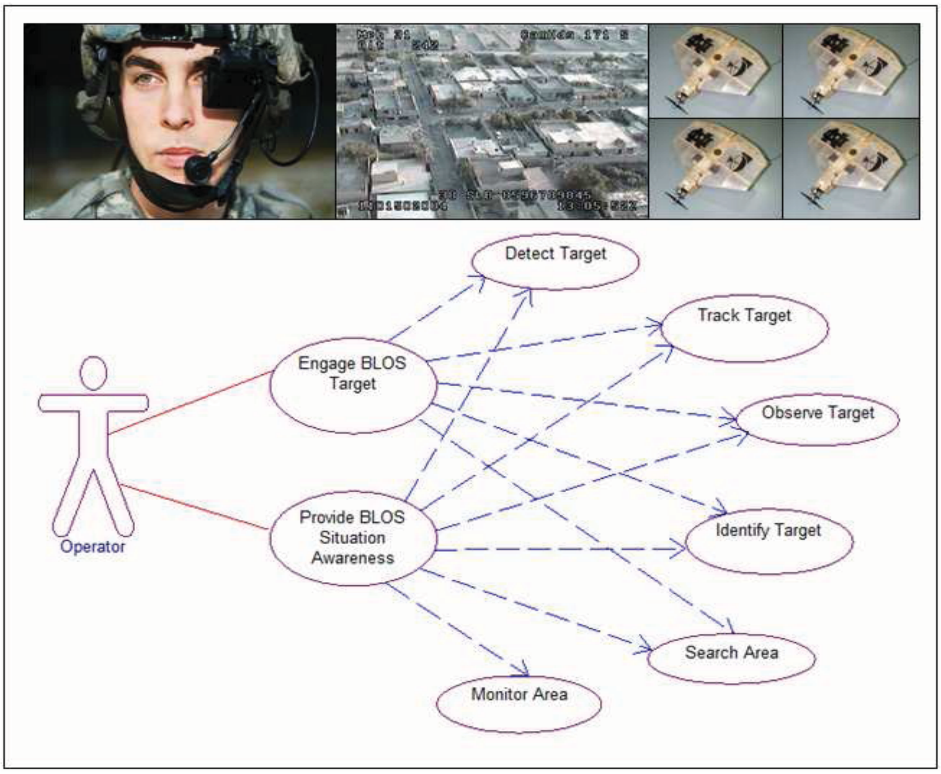


Figure 21 - Operational Concept Graphic for small / micro UAV system

4.3 Logical Model of Motion Planning and Control Architecture

The aim of a logical system architecture is to define required functionality in an implementation independent form. This is intended as an aid to innovation by avoiding 'solutioneering[28]'. The motion planning and control architecture (including related components) adopted for the technical work of this thesis is shown in Figure 22. This architecture is designed specifically for unmanned vehicle operation within complex obstacle-rich environments and comprises four primary components, each of which is discussed below:

- *Mission / goal based reasoning* - Responsible for specifying the requirements for large scale motion plans. This may involve coordinating several vehicles or prioritising current high level goals. For low autonomy systems this subsystem may be replaced by an operator interface defining, for example, start / end positions or an area to be searched. For multiple vehicle systems this layer may be part of the operator interface rather than on each individual vehicle.
- *On board sensing* - Responsible for all real-time sensing, including localisation, dynamic state, wind vector and obstacle sensing.
- *Obstacle / environment modelling* - Responsible for generation of a real-time 4D environment model that can be used by both goal based reasoning and motion

[28] Beginning a design process with preconceptions about the final solution is sometimes known as *solutioneering.*

planning elements. This model represents the system's current beliefs about its operational environment, and contains both static and dynamic obstacles, including predictions of future obstacle positions if required.

- *Motion planning and control* - Responsible for all motion planning and control, and comprised of the following four levels:
    - *Level-1: Global planning* - Large scale, or global, motion planning, e.g. calculate optimal obstacle free trajectory from point A to point B while also visiting points C,D,E etc. This global plan must include at least first order vehicle dynamics, and is likely to form part of a larger plan, for example coordinating the use of several vehicles. The designed global plan is assumed to be *pseudo static*, e.g. subject to change with new information / mission goals but this is assumed to occur at a timescale that allows the layers below it to treat it as static.
    - *Level-2: Local planning* - Continuous local motion planning, responsible for execution of the global trajectory given detailed knowledge of the local situation (e.g. disturbances, obstacles detected en-route, local wind conditions). Planning horizon Limited to a maximum of the sensor horizon, allowing increased emphasis on accurate vehicle dynamics and rapid reaction to disturbances and unexpected obstacles.
    - *Level-3: Outer-loop control* - Traditional autopilot or outer-loop control functionality such as provision of altitude, speed, heading demand. Responsible for handling any complex vehicle response nonlinearities or control coupling and proving a defined level of performance that can be used by the local motion planning.
    - *Level-4: Inner-loop control* - Traditional inner-loop control for vehicle stabilisation.

This architecture resulted from the following primary architectural decisions:

- *Division of motion planning into distinct global and local layers* - This issue is also discussed in section 1.3.1 and section 1.3.2, with the primary driver being removing the burden of global trajectory re-design each time a new obstacle is detected, or a dynamic obstacle changes path. Many obstacles that may be detected en route can be treated as local issues, and handled without impact on the global trajectory. For example, if while monitoring a street canyon an overhead cable, or bird, is detected then an appropriate response would be a temporary deviation from the global trajectory until it can be safely rejoined. This delegation of responsibility can only be enabled by making the local trajectory tracking layer 'situation aware', e.g. giving it access to the real-time 4D obstacle map. The combination of operation in close proximity to obstacles known *a priori* and significant wind disturbances also suggests that a situation aware tracking layer is appropriate, allowing the global trajectory to be safely rejoined after a gust with an awareness of the surrounding obstacle space. A receding horizon planned response is preferred to a reactive[29] approach as it allows the future consequences of current control actions to be considered, increasing the ability to deal with complex dynamic scenarios.

- *Vehicle performance interface between outer-loop control (level 3) and local motion planning (level 2)* - In order to reduce the complexity of the local motion planning task this architecture uses an output space design for the local motion plan rather than the traditional control space design used in RHC. As discussed in section 4.5 this is intended to allow both an increased design rate and horizon, in order to suit complex environment operations. This approach is enabled by the use of a vehicle performance specification interface provided by the lower level control algorithms, ensuring that the output space designs are dynamically feasible.
- *Separation of obstacle sensing & environment modelling from motion planning & control* - The approach taken here is to abstract the detection of the local obstacle space and subsequent creation of a local obstacle map away from all motion planning and control tasks. These are three highly complex, related, but fundamentally separate tasks, and maintaining this clear separation avoids additional design complexity[30].
- *Availability of a priori obstacle map* - In order to reduce the demands on real-time obstacle detection and mapping it is assumed that a basic obstacle map (e.g. main buildings layout) will be available pre-flight. This information would routinely be available to troops, therefore this is not an unrealistic assumption. Additionally, this *a priori* map may also be used to aid the localisation tasks based on relative position from globally known obstacles.

[29] The term *reactive* is used here to mean a reflexive, or instantaneous / non-planned approach that does not consider the future impact of current control signals.

[30] Note that some architectures reviewed in the literature survey of section 3.3 do not maintain this separation, e.g. the optic flow methods of [30] & [4].

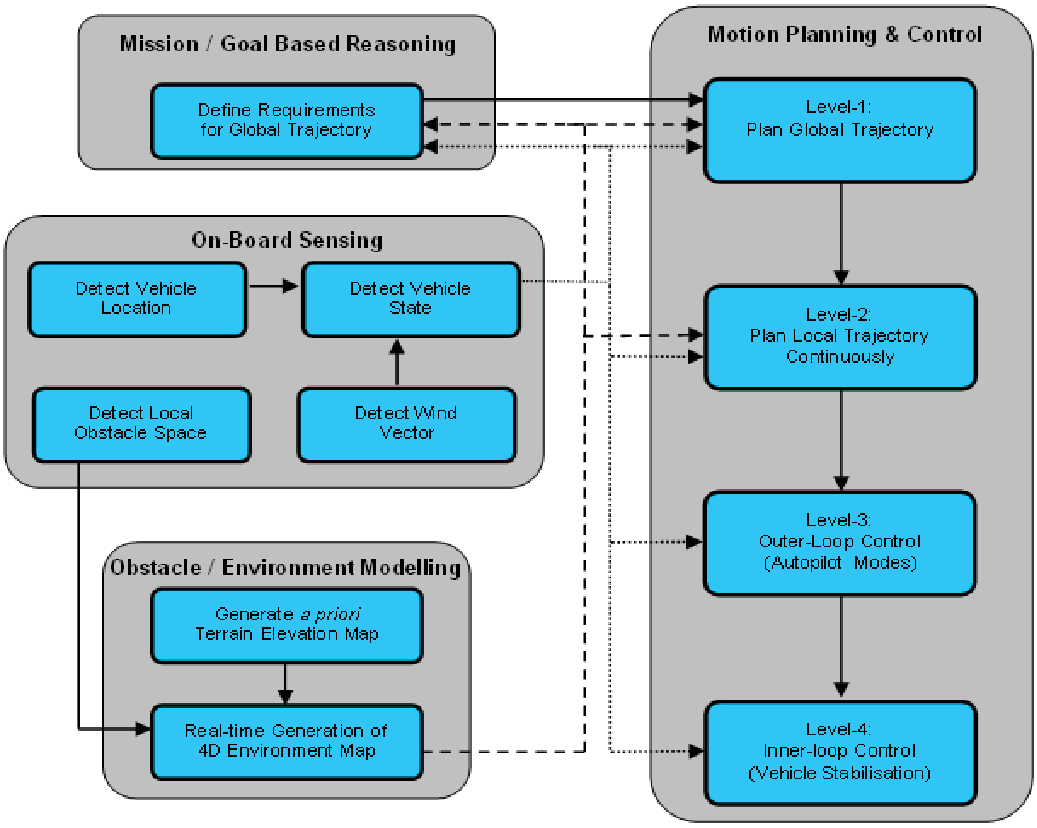

Mission / Goal Based Reasoning
Define Requirements for Global Trajectory
Motion Planning & Control
Level-1: Plan Global Trajectory
On-Board Sensing
Detect Vehicle Location
Detect Vehicle State
Detect Local Obstacle Space
Detect Wind Vector
Level-2: Plan Local Trajectory Continuously
Level-3: Outer-Loop Control (Autopilot Modes)
Obstacle / Environment Modelling
Generate *a priori* Terrain Elevation Map
Real-time Generation of 4D Environment Map
Level-4: Inner-loop Control (Vehicle Stabilisation)

Figure 22 - Logical Architecture of Vehicle Guidance & Control Functionality

4.4 Technology Needs of the Logical Architecture

This section considers specific technologies, or solutions, that may be applied to the previously defined logical architecture. The focus of this thesis is the guidance and control algorithms, but the other primary components are also reviewed in order to investigate the feasibility of the defined architecture, helping to identify critical related technologies. Each of the core components and internal functionality of the architecture shown in Figure 22 is discussed further in the next section.

4.4.1 Mission / Goal Based Reasoning

For systems with a low level of autonomy this component would revert to a simple operator interface, allowing the user direct control of each vehicle's current task. For systems with higher levels of autonomy a mission-task based operator interface would be preferred, delegating the creation of sub-tasks with associated vehicle positioning requirements to this component. Implementation of this functionality requires a machine intelligence / reasoning based approach that allows the overall vehicle usage to be optimised to meet high level operator goals.

Related research can be found in the literature, but operator acceptance and regulatory issues limits practical applications. An example multi-agent based implementation approach is discussed by Baxter, Horn and Leivers [2] where an intelligent agent framework is used to allow a single operator to task / goal level control over a package of UAVs. Another example implementation approach is the SOAR[31] cognitive architecture for developing intelligent systems.

4.4.2 On-board Sensing

A detailed discussion of sensor capabilities for small / micro UAVs is beyond the scope of this thesis. However, some high level consideration is given to the necessary sensors in order to consider the realism of the proposed system. The primary sensing requirements of the architecture along with potential solution options are each discussed below.

- *Detect vehicle location* - The default localisation solution for most modern systems is the global positioning system (GPS). GPS receivers are small and light and therefore are well suited to small / micro UAVs. However, for the proposed complex environment application several issues arise. Firstly, a GPS solution is unlikely to be accurate enough for the precise placement required in complex environments (standard military accuracy is approximately 3-5m, which may be further degraded in urban environments due to satellite loss and multi-path issues). Secondly, within complex urban environments a GPS localisation fix may be worsened by either multi-path effects or loss of satellite coverage, therefore further reducing accuracy. Finally, GPS signals are susceptible to jamming by hostile forces, therefore again reducing reliability. The accuracy of the GPS localisation fix may be enhanced by also employing an inertial navigation system (INS) where a set of gyros and accelerometers are used along with navigation equations to provide positions, speeds and attitudes via a dead-reckoning approach. Suitably small / light sensors are now commonly available based on microeletromechanical (MEM) technology. However, for the situation of sustained GPS unavailability INS based position errors are likely to accumulate to levels unacceptable for absolute navigation, therefore an additional source of localisation is required. Given the assumed availability of a 3D *a priori* environment map, it should be feasible to use environment feature recognition (using on-board obstacle sensors) to calculate the current position relative to known features. This approach is similar to that used in the SLAM (simultaneous localisation and mapping) problem and matches the human approach well, where we navigate relative to known features rather than in an absolute frame. The final solution as a blend of these three sources is shown in Figure 23, and is assumed to be available for the remainder of this thesis.

[31] http://sitemaker.umich.edu/soar/home

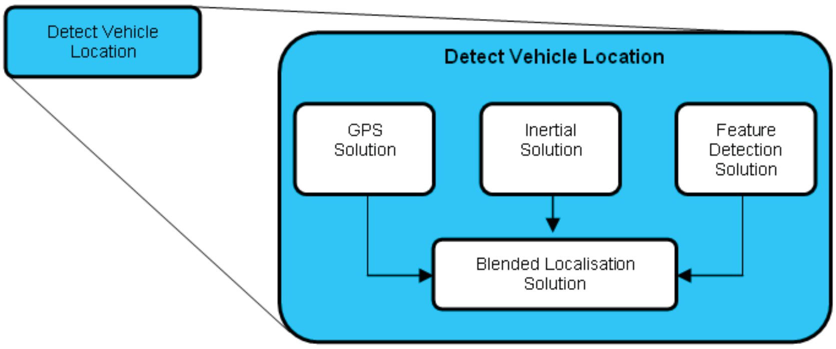


Figure 23 - Solution Architecture for Vehicle Localisation Functionality

- *Detect local obstacle space* - Due to the rise in popularity of unmanned vehicles many types of obstacles sensors are now available, a full discussion of which is again beyond the scope of this thesis. Examples include passives systems such as stereo vision and optic flow and active systems such as ultrasonic, radar, LIDAR, etc. Active systems such as LIDAR / Radar typically offer the greatest accuracy, but at the expense of size and weight. However, the technology is getting smaller as more products as developed specifically for small / micro UAVs. An example of this is the IMSAR NanoSAR[32] (157x190x114mm) which weighs only 0.9kg but has a range of 1km with an accuracy of 1m. Additionally, very small, light and low power short pulse ultra-wideband (UWB) radars have also been used for obstacle avoidance and detection of overhead cables etc. Fontana [15] discussed an ultra-wide-band radar that was designed for micro UAV applications that was only 65x70mm with a weight of 15g. For small, low-power, lightweight applications visual sensors are perhaps the most promising, with commercial off the shelf (COTS) technology rapidly developing. Visual systems are often reliant on computationally intensive image processing algorithms, however, processing power is becoming smaller, lighter and cheaper therefore this may not be prohibitive.

- *Detect vehicle dynamic state* - Detection of a vehicle's dynamic state (e.g. airspeed, attitudes, accelerations etc.) is a relatively mature technology and may be achieved by a variety of standard approaches including pressure sensors and the previously discussed INS approach.

- *Detect wind vector* - For small / micro UAVs where the wind vector may be a significant component of the maximum available airspeeds it is particularly important to measure current wind conditions. This data combined with an understanding of urban wind flow characteristics may be used to help anticipate significant flow changes (e.g. at canyon intersections and suction effects behind buildings) that would increase control accuracy. Calculation of the current wind vector may be achieved by comparing airspeeds with ground speeds (via localisation solution).

[32] synthetic aperture radar (SAR) , see www.imsar.com for further details.

4.4.3 Obstacle / Environment Modelling

A detailed discussion of real-time obstacle modelling is again beyond the scope of this thesis. The generation of such an obstacle map, and in particular the assurance of it's quality, is a significant challenge and is the focus of much research, with examples including the Tartan Racing and Team MIT entries into the DARPA Urban Challenge [74 & 77]. Inputs to such a process would be both the *a priori* static obstacle map and the real-time sensor returns, and the output would be a 4D (predictive) obstacle occupancy map allowing the motion planning component to be situation aware. A potential low level architecture is shown in Figure 24, with key problems including association of sensor returns with known features or current tracks, creation of new tracks / features and predictions of states based on current state and past behaviour. This is a form (or subcomponent) of the SLAM problem, but is aided by the assumption that *a priori* data is available. Generation of the *a priori* obstacle map may be achieved by a variety of sources, e.g. using standard terrain elevation maps, image processing data from reconnaissance flights.

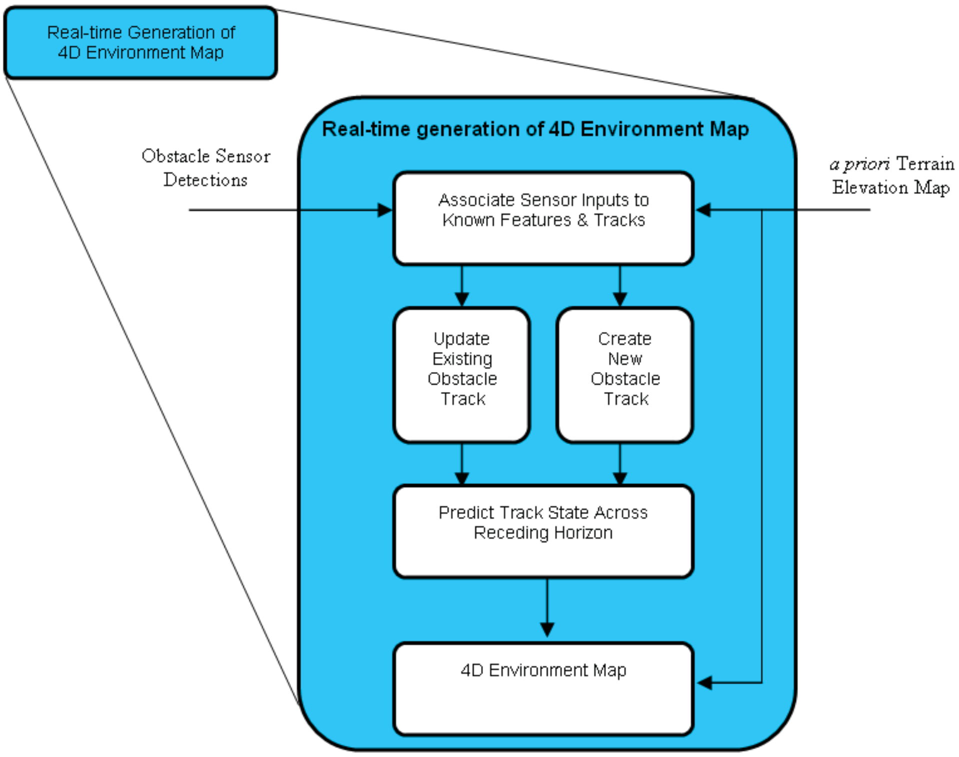

Real-time Generation of 4D Environment Map
Real-time generation of 4D Environment Map
Obstacle Sensor Detections
a priori Terrain Elevation Map
Associate Sensor Inputs to Known Features & Tracks
Update Existing Obstacle Track
Create New Obstacle Track
Predict Track State Across Receding Horizon
4D Environment Map

Figure 24 - Solution Architecture for Generation of 4D Environment Map

4.4.4 Motion Planning & Control

The previously defined motion planning and control architecture is comprised of four levels, each of which are discussed below.

- *Level-1: Global planning* - A wide range of techniques and algorithms are available to apply to this problem including Voronoi graph, visibility lines, rapidly exploring random tree, probabilistic roadmaps, Dubin's paths, differential geometry, genetic algorithms etc. (section 3.3.1). For planning paths through or over urban canyons, standard graph based approaches can be used (e.g. A* search) allowing optimal routing decisions to be made. Such approaches are commonly used in commercial systems such as GPS navigation devices.

- *Level-2: Local planning* - A framework for this problem that may be found within the literature is receding horizon control (RHC). The primary difference is that the standard RHC problem is posed in the control space, therefore overlapping into the level-3 / level-4 functionality.
- *Level-3 & Level-4: Inner & outer-loop control* - This functionality can be considered as reasonably mature technology, and is typically implemented via multi-loop PID feedback controllers. For more complex multivariable problems modern robust control approaches such as H^{∞} are also available. Although such techniques are less commonly implemented in practice, they offer better handling of coupling between different control axes, as well as guarantees of defined levels of robustness.

Overall, it is considered that the local motion planning layer is the least mature at present, and therefore this is the focus of the technical work of this thesis.

4.5 Definition of Technical Focus - *Continuous Local Motion Planning & Control*

4.5.1 Problem Overview

Based on the defined military requirement, subsequent problem context exploration and the development of a logical system architecture and associated technology needs, the technical focus chosen for this thesis is the development of local motion planning and control algorithms for the operation of small / micro UAVs within complex obstacle-rich environments. As the complexity of the environment increases so does the need to consider the local obstacle space continually, and this is enabled by continuous situation aware local motion planning. The primary problem that is addressed is that of *situation aware* trajectory tracking, where the goal is to track a continuous global trajectory with an awareness of the local obstacle space, e.g. making the tracking algorithms *situation aware*. This is illustrated in Figure 25 for the trajectory tracking problem where a UAV is shown tracking a global trajectory in the presence of both static & dynamic obstacles. The performance limits of the vehicle (for the current state) are shown as a feasible manoeuvre envelope emanating from the current vehicle state, and the role of the local motion planning layer is then to calculate (continually) this set of feasible trajectories, then select the optimal one for execution. Fundamentally this is therefore a constrained optimisation problem, with only the initial portion of the trajectory executed before repeating the design using the latest information.

As the vehicle performance limits of the vehicle are known in advance (as fixed by the vehicle design process), the effort required is to transform the results of these limits into the current scenario and understand their impact. The problem therefore involves repeatedly exploring the feasible manoeuvre envelope (given the current dynamic state) and selecting option that best fits the current mission priorities. Other local motion planning and control problems are also considered, including single point (rather than trajectory) acquire and dynamic obstacle tracking. As well as developing an approach suitable for operation of a single vehicle consideration must also be given to multiple vehicle operation, aiding the provision of decentralised multiple vehicle deconfliction.

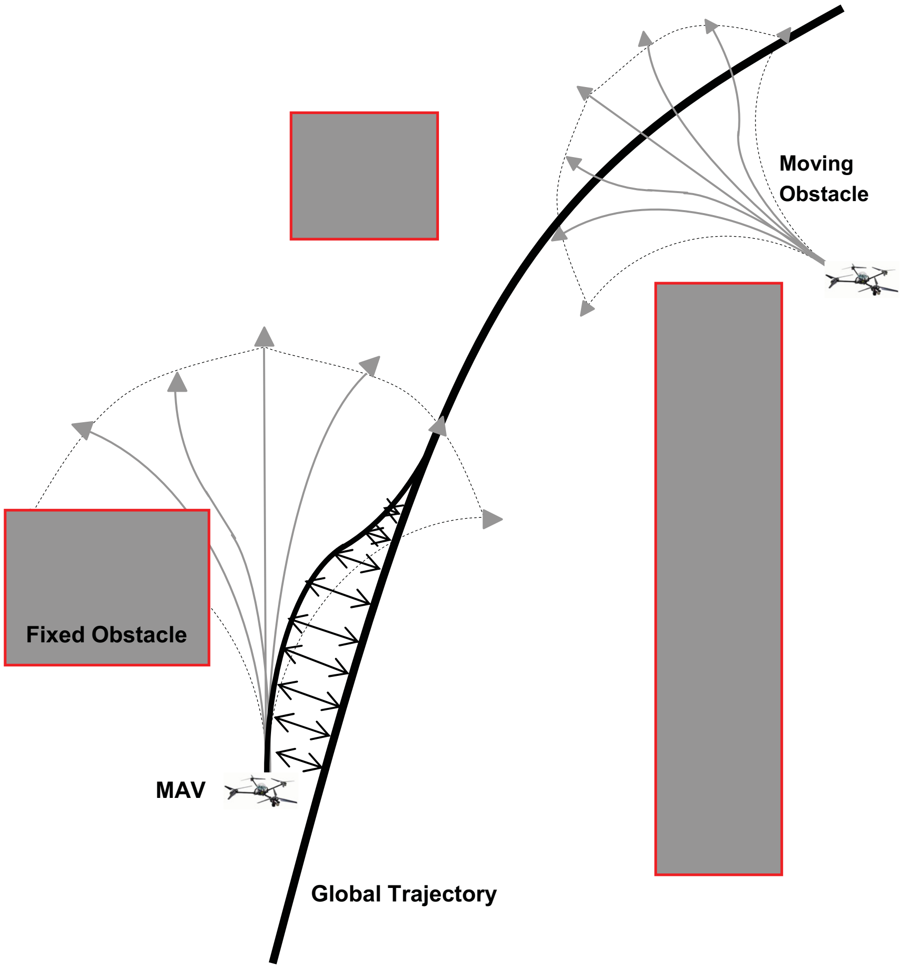


Figure 25 - Situation Aware Trajectory Tracker

The primary design goals for the local motion planning and control problem are:

- *High trajectory re-design rate* - A 'continuous' / high rate design process (e.g. 10Hz or faster) provides the necessary rapid response to disturbances & new information. If this high design rate response cannot be provided then an additional event triggered sense and avoid layer may be required, therefore further complicating the overall system.
- *Receding horizon length related to sensor capabilities and vehicle dynamics* - Planning horizon must allow for collision avoidance given vehicle performance limits. For example, if it takes *n* seconds to transition from maximum forward speed to a hover then the horizon length must be at least *n* seconds. Requirements for

the range of the on board collision avoidance sensors would also be related to vehicle dynamics (e.g. must exceed manoeuvre limits), therefore limiting continuous local motion planning to sensor confirmed free space.

- *Full 4D design* - For UAV operation within complex environments including the deconfliction of several vehicles it will be necessary to manoeuvre in all 4 dimensions.
- *Incorporate known wind vector information* - Given the importance of wind behaviour for small / micro UAVs known / estimated wind data should be used where possible.
- *Simulation based on a nonlinear 6dof vehicle model* - Interested in the feasibility of application to a real vehicle, therefore a full nonlinear 6dof vehicle simulation model should be used.

It can be argued that this type of approach has the potential to improve on the performance of a human pilot for the following reasons:

- *Faster response* - Automatic control systems are able to provide response times far superior to human reactions.
- *Sensor improvements on human eye* - A human pilot is limited to the visual spectrum, whereas on board sensors have the potential to see through fog, cloud etc. as well as seamlessly incorporating non line-of-sight information when available.
- *Better handling of complex multi-obstacle tracking & de-confliction* - The ability of human pilots to manage complex multiple obstacle deconfliction scenarios degrades rapidly with the number of obstacles.
- *Replace pilot judgment with knowledge of vehicle manoeuvre capabilities* - Manual human control (aircraft, car, bicycle etc.) is based largely on learned performance expectations, e.g. braking distance or cornering performance in a car, or turn rate performance in an aircraft. Humans all have a different understanding of the true available performance capabilities that would be available to an automatic trajectory optimisation system.

4.5.2 Design Space

There are two primary options for solution of this problem, control space or output space design (See Section 1.3.3). The option taken within this thesis is to perform an output space design. The aim of this is to meet the design rate and horizon length goals discussed previously, helping to handle the complexities of both operating within obstacle-rich environments and decentralised multi-vehicle deconfliction.

5 Local Motion Planning Framework - *Output Space Component*

5.1 Overview

This chapter describes the development of a new generic framework for the previously defined situation aware trajectory tracking problem. Alternative implementation options for elements of this framework are also discussed, including pros and cons of the various design decisions. Before presenting the framework, the quadrotor vehicle that it is applied to is introduced. It should be noted though, that the overall framework is generic to vehicle type, and therefore is equally applicable to unmanned (or optionally manned) vehicles in other domains, e.g. ground, sea etc.

5.2 Quadrotor Unmanned Air Vehicle

Algorithm development and test within this thesis is based on a nonlinear 6DOF Matlab / SIMULINK implementation of a quadrotor UAV (Figure 7). Quadrotor vehicles can be found in a variety of sizes, with the simulation model used here based on the Draganflyer XPro UAV (length / width <1m, weight 2.36kg, payload 0.5kg, endurance approx 18mins). Although this vehicle is larger than desired for the previously discussed missions, it is still considered suitable for operation within complex obstacle-rich environments due to it's high manoeuvrability, including vertical take-off and landing, hovering, backwards, sideways flight etc. This ability to rapidly manoeuvre in any direction is particularly advantageous for handling disturbances due to gusts and turbulence while in close proximity to obstacles. Additionally, this model provides a realistic challenge for algorithm development and test, and it's behaviour is also representative of smaller quadrotor vehicles.

One of the primary advantages of the quadrotor configuration is it's simplicity. Each rotor is powered by an individual electric motor, with control provided purely by varying individual rotor speeds. Each rotor produces both lift and torque, therefore resulting in a high degree of coupling between the different axes and control inputs. This coupling increases the manoeuvrability of the platform at the expense of an increase in control complexity.

Although individual control of each rotor is possible, in order to manage the coupling they are typically [43] used in a coordinated fashion. An example control strategy is outlined below:

- Individual control inputs (rotor rpm) referred to as $r_1,\ r_2,\ r_3,\ r_4$.
- Vehicle aligned with rotor-1 & rotor-3 in x-axis (body), rotor-2 & rotor-4 in y-axis (body).
- Rotor-1 & rotor-3 turn clockwise & rotor-2 & rotor-4 turn anti-clockwise, therefore torque can be cancelled by keeping $r_1 + r_3 = r_2 + r_4$.

- Height control provided by collectively varying the speed of all rotors by the same magnitude (therefore balancing torque).
- Forward speed (u) control provided by increasing r_1 and decreasing r_3 by the same amount. This maintains the torque balance, while causing a pitching moment to increase the pitch angle (θ). Varying θ moves total thrust away from vertical resulting in forward / backwards motion.
- Lateral speed (v) control uses the same approach as forward speed, but now varying r_2 & r_4 in order to vary the roll angle ϕ.
- Note that as changes in θ and ϕ move the thrust vector away from the horizontal, total thrust also needs to be increased to decouple vertical from horizontal motion.
- Torque control can be provided by varying the balance between all four rotor speeds.

5.3 Trajectory Description

5.3.1 Polynomial Functions

In order to reduce the overall dimension of the design problem, local motion trajectories are described by polynomial functions. Various types of polynomials are available, and they are typically defined as a linear combination of coefficients and basis functions as shown in equation (5-1).

$$p(\tau) = \sum_{i=0}^{n} c_i B_i(\tau) \tag{5-1}$$

where:

$\mathbf{p}(\tau)$	=	curve as a function of τ
n	=	order of the polynomial
τ	=	curve parameter. min / max define the range of the basis function
c_i	=	coefficient for each of the ith order basis functions
$\mathbf{B_i}(\tau)$	=	ith order basis function

Various options exist for the choice of basis function including, Hermite, Chebyshev, Laguerre, Bezier etc. (see Figure 26 to Figure 30) but all can be reduced via simple algebraic manipulation to the basic polynomial form of:

$$p(\tau) = c_0 \times \tau^0 + c_1 \times \tau^1 + c_2 \times \tau^2 + c_3 \times \tau^3 + c_4 \times \tau^4 + c_5 \times \tau^5 + etc.. \tag{5-2}$$

Note that equation (5-2) simply represents the most basic form of polynomial, that of equation (5-1) with basis functions as increasing powers of the curve parameter τ (as shown in Figure 30).

Plots of different sets of basis functions are shown in Figure 26 to Figure 30 where it can be seen that the functions define the influence of each of the coefficients over the length of the curve. For example, if a basis function has zero value at a point in a curve then the value of the coefficient of that basis function has no effect on the curve at that point. Additionally, the range of the curve parameter (τ) can define the shape of the basis functions, with the obvious examples being the range of the basis functions in Figure 28 and Figure 30, where the higher order terms will swamp the lower order terms as τ increases beyond 1. For the LMP framework described here Bezier polynomials are used, as discussed further in the next section.

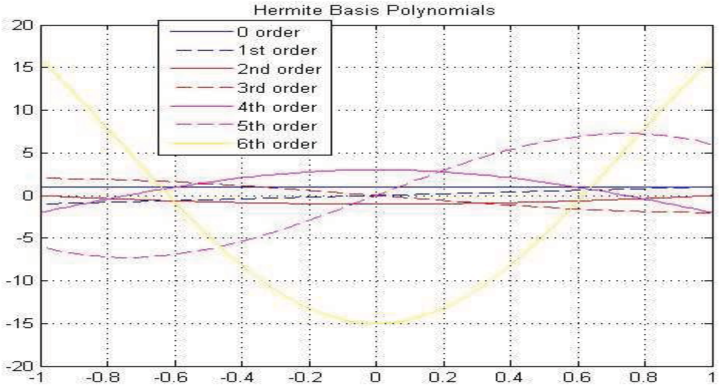


Figure 26 - Hermite Basis Functions

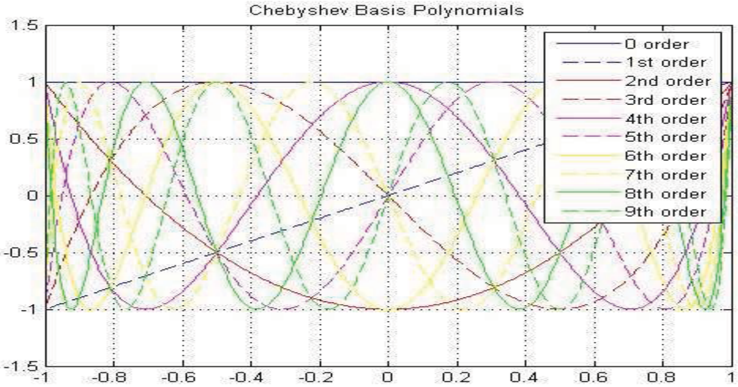


Figure 27 - Chebyshev Basis Functions

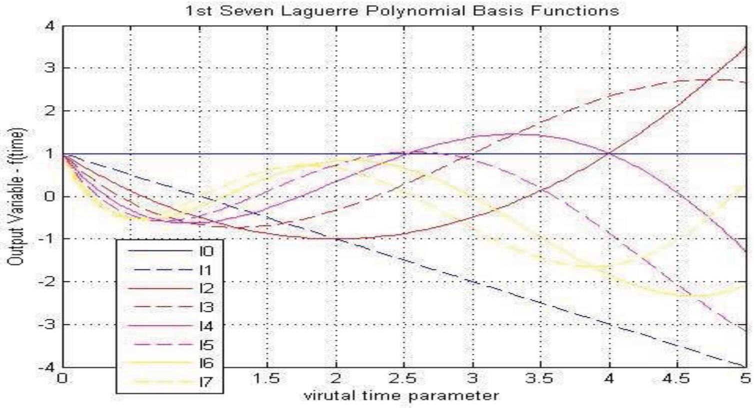


Figure 28 - Laguerre Basis Functions

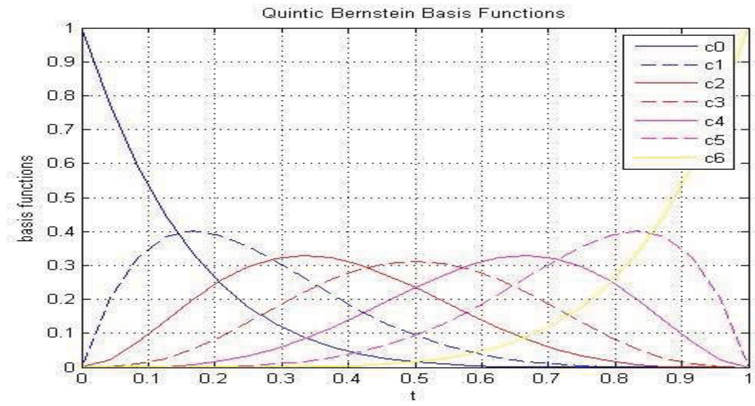


Figure 29 - Bezier Basis Functions

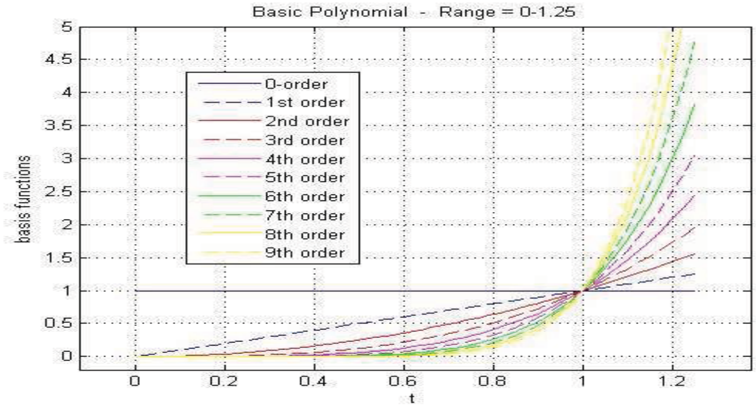


Figure 30 - Basic Polynomial Basis Functions

The order of a curve defines both the numner of design variables and the number of points of inflection, therefore increasing curve-order allows more complex trajectories. The order of the curve should be related to both the dynamics of the vehicle and the design horizon. For example, a short design horizon for a vehicle with slow dynamics does not require a high order curve, as the dynamic constraints of the vehicle dictate that the available trajectory flexibility will never be used, therefore resulting in an unnecessarily large design space.

5.3.2 Bezier Curves

There are two primary advantages of the Bezier basis functions over the others shown above. Firstly, it can be seen from Figure 29 that the basis functions help to decouple the influence of each of the coefficients. For example, the coefficient for the 6th order basis function can be seen to have increasing effect towards the end of the curve, whereas the coefficient for the 0th order basis function has increasing effect towards the start of the curve. Similarly, the range of influence of each of the coefficients depends on which basis function it operates on. It is expected that this decoupling effect will help to reduce computational effort during trajectory optimisation.

The second primary advantage of Bezier curves is that there is an intuitive relationship between the coefficients and the resulting curve. Bezier curves were originally developed as a design tool for the automotive industry, with the shape of a curve being easily manipulated by varying the position of different control points (the curve coefficients). To aid the design process the control points are placed in intuitive locations to influence the end curve. This intuitive relationship helps both to define upper & lower bounds on the optimisation process as well as aiding development & debugging.

The basis functions used in Bezier curves are defined below, and are known as Bernstein polynomials:

$$\mathbf{B}_{\mathbf{v},\mathbf{n}} = \begin{pmatrix} \mathbf{n} \\ \mathbf{v} \end{pmatrix}^{\mathbf{v}} \tau^{\mathbf{v}} (1-\tau)^{\mathbf{n}-\mathbf{v}}, \quad \mathbf{v} = 0,1,2,...,\mathbf{n} \tag{5-3}$$

Where:

$\begin{pmatrix} \mathbf{n} \\ \mathbf{v} \end{pmatrix}$ *is a binomial coefficient*

$\mathbf{n}$ *is the highest order of the curve that the coefficients are to be calculated for*

$\mathbf{v}$ *is the order of the basis function*

For example, for a 6th order Bezier curve the basis functions can be calculated to be:

$$\begin{aligned}
\mathbf{B}_0 &= (1-\tau)^6 \\
\mathbf{B}_1 &= 6\tau(1-\tau)^5 \\
\mathbf{B}_2 &= 15\tau^2(1-\tau)^4 \\
\mathbf{B}_3 &= 20\tau^3(1-\tau)^3 \\
\mathbf{B}_4 &= 15\tau^4(1-\tau)^3 \\
\mathbf{B}_5 &= 6\tau^5(1-\tau)^5 \\
\mathbf{B}_6 &= \tau^6
\end{aligned} \tag{5-4}$$

It can be seen from equation (5-4) that the Bernstein basis functions differ from the others in that each individual function depends on the order of the polynomial. For example, the six basis functions used for a 5th order curve are all different from the seven basis functions used for a 6th order curve, although they always display the same arrangement of varying influence across the range of τ. With the other types of basis function increasing the order of the curve simply add an extra basis function to the existing set. Also, with Bezier curves the range of the curve parameter τ is always limited to between zero and one.

5.3.3 Polynomial Shaping

Using polynomial functions to describe trajectories simplifies the design to a curve shaping process, where the design variables are the set of coefficients (c_i) of the chosen basis functions. Using this process a curve can be varied, or shaped, by altering one or more of the design variables. This is illustrated in Figure 31, where the impact of varying c_2 in a 5th order Bezier curve is illustrated.

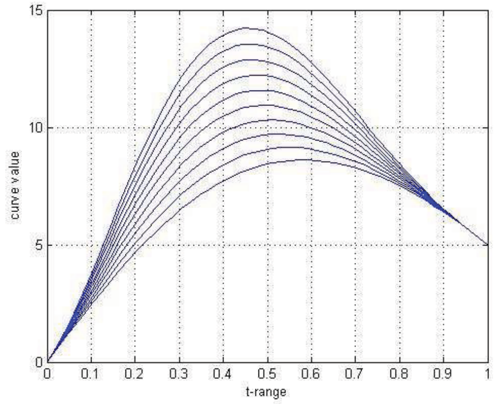


Figure 31 - Polynomial Trajectory Shaping

5.3.4 Quadrotor Receding Horizon Speed Profiles

Four dimensional trajectories for the quadrotor vehicle are described using three sixth order Bezier curves. Sixth order curves are used as they are considered to provide a good compromise between dynamic flexibility over the design horizon and number of design variables. The trajectory description was chosen to match the format of the vehicle performance constraints (section 5.4.2), with the three curves describing receding horizon speed profiles in the forward, lateral and vertical axis:

$$\begin{aligned}
\mathbf{u}(\tau) &= \mathbf{c}_0^{\mathbf{u}}\mathbf{B}_0(\tau) + \mathbf{c}_1^{\mathbf{u}}\mathbf{B}_1(\tau) + \mathbf{c}_2^{\mathbf{u}}\mathbf{B}_2(\tau) + \mathbf{c}_3^{\mathbf{u}}\mathbf{B}_3(\tau) + \mathbf{c}_4^{\mathbf{u}}\mathbf{B}_4(\tau) + \mathbf{c}_5^{\mathbf{u}}\mathbf{B}_5(\tau) + \mathbf{c}_6^{\mathbf{u}}\mathbf{B}_6(\tau) \\
\mathbf{v}(\tau) &= \mathbf{c}_0^{\mathbf{v}}\mathbf{B}_0(\tau) + \mathbf{c}_1^{\mathbf{v}}\mathbf{B}_1(\tau) + \mathbf{c}_2^{\mathbf{v}}\mathbf{B}_2(\tau) + \mathbf{c}_3^{\mathbf{v}}\mathbf{B}_3(\tau) + \mathbf{c}_4^{\mathbf{v}}\mathbf{B}_4(\tau) + \mathbf{c}_5^{\mathbf{v}}\mathbf{B}_5(\tau) + \mathbf{c}_6^{\mathbf{v}}\mathbf{B}_6(\tau) \\
\mathbf{w}(\tau) &= \mathbf{c}_0^{\mathbf{w}}\mathbf{B}_0(\tau) + \mathbf{c}_1^{\mathbf{w}}\mathbf{B}_1(\tau) + \mathbf{c}_2^{\mathbf{w}}\mathbf{B}_2(\tau) + \mathbf{c}_3^{\mathbf{w}}\mathbf{B}_3(\tau) + \mathbf{c}_4^{\mathbf{w}}\mathbf{B}_4(\tau) + \mathbf{c}_5^{\mathbf{w}}\mathbf{B}_5(\tau) + \mathbf{c}_6^{\mathbf{w}}\mathbf{B}_6(\tau)
\end{aligned} \tag{5-5}$$

where:

$\mathbf{B_i}(\tau)$ = *ith Bernstein basis function for a 6th order curve as defined in section 5.3.2.*

c_i^u = *ith coefficient in the u-axis*

c_i^v = *ith coefficient in the v-axis*

c_i^w = *ith coefficient in the w-axis*

τ = *curve parameter (*$0 \le \tau \le 1$*)*

This choice of trajectory description helps keep the relationship between performance constraints and the trajectory description as simple as possible, therefore simplifying the enforcement of constraints during the optimisation process. The design variables are the coefficients of the polynomials, therefore resulting in a total of twenty-one variables, seven for each axis, as shown in equations (5-6) & (5-7):

$$\begin{aligned}
\mathbf{C^u} &= \begin{bmatrix}\mathbf{c}_0^{\mathbf{u}} & \mathbf{c}_1^{\mathbf{u}} & \dots & \mathbf{c}_6^{\mathbf{u}}\end{bmatrix}^{\mathbf{T}} \\
\mathbf{C^v} &= \begin{bmatrix}\mathbf{c}_0^{\mathbf{v}} & \mathbf{c}_1^{\mathbf{v}} & \dots & \mathbf{c}_6^{\mathbf{v}}\end{bmatrix}^{\mathbf{T}} \\
\mathbf{C^w} &= \begin{bmatrix}\mathbf{c}_0^{\mathbf{w}} & \mathbf{c}_1^{\mathbf{w}} & \dots & \mathbf{c}_6^{\mathbf{w}}\end{bmatrix}^{\mathbf{T}}
\end{aligned} \tag{5-6}$$

therefore the full design vector is given by:

$$\mathbf{C} = \left[\mathbf{C}^{\mathbf{u}} \quad \mathbf{C}^{\mathbf{v}} \quad \mathbf{C}^{\mathbf{w}}\right]^{\mathbf{T}} \tag{5-7}$$

5.3.5 Calculation of Acceleration, Rate of Acceleration & Position Profiles

A fixed receding horizon time ($\mathbf{t}_{\mathbf{horizon}}$) is used in order to simplify the computation of the curve derivatives, as shown below.

For Bezier curves we always have $0 \le \tau \le 1$ which allows us to define the relationship between τ and time (t) as:

$$\mathbf{t} = \mathbf{t}_{\mathbf{horizon}} \times \tau \tag{5-8}$$

which then allows the acceleration profiles & rates of acceleration profiles over the receding horizon to be calculated in each of the three axis as follows:

$$\frac{\mathbf{du}}{\mathbf{dt}} = \frac{\mathbf{du}}{\mathbf{d}\tau} \cdot \frac{\mathbf{d}\tau}{\mathbf{dt}} = \frac{1}{\mathbf{t}_{\mathbf{horizon}}} \cdot \frac{\mathbf{du}}{\mathbf{d}\tau} \tag{5-9}$$

and

$$\frac{\mathbf{d}^2\mathbf{u}}{\mathbf{dt}^2} = \frac{\mathbf{d}}{\mathbf{dt}}\left(\frac{\mathbf{du}}{\mathbf{dt}}\right) = \frac{\mathbf{d}}{\mathbf{dt}}\left(\frac{1}{\mathbf{t}_{\mathbf{horizon}}} \cdot \frac{\mathbf{du}}{\mathbf{d}\tau}\right) = \frac{\mathbf{d}}{\mathbf{d}\tau}\left(\frac{1}{\mathbf{t}_{\mathbf{horizon}}} \cdot \frac{\mathbf{du}}{\mathbf{d}\tau}\right)\frac{\mathbf{d}\tau}{\mathbf{dt}} = \frac{1}{\mathbf{t}^2_{\mathbf{horizon}}} \cdot \frac{\mathbf{d}^2\mathbf{u}}{\mathbf{d}\tau^2} \tag{5-10}$$

Differentiating equations (5-5) with respect to time, and substituting equation (5-9) and (5-10) gives the following acceleration and rate of acceleration profiles:

$$\begin{gathered}
\dot{\mathbf{u}}(\tau) = \frac{1}{\mathbf{t}_{\mathbf{horizon}}}\left(\mathbf{c}_0^{\mathbf{u}}\frac{\mathbf{dB}_0(\tau)}{\mathbf{d}\tau} + \mathbf{c}_1^{\mathbf{u}}\frac{\mathbf{dB}_1(\tau)}{\mathbf{d}\tau} + \mathbf{c}_2^{\mathbf{u}}\frac{\mathbf{dB}_2(\tau)}{\mathbf{d}\tau} + \mathbf{c}_3^{\mathbf{u}}\frac{\mathbf{dB}_3(\tau)}{\mathbf{d}\tau} + \mathbf{c}_4^{\mathbf{u}}\frac{\mathbf{dB}_4(\tau)}{\mathbf{d}\tau} + \mathbf{c}_5^{\mathbf{u}}\frac{\mathbf{dB}_5(\tau)}{\mathbf{d}\tau} + \mathbf{c}_6^{\mathbf{u}}\frac{\mathbf{dB}_6(\tau)}{\mathbf{d}\tau}\right) \\
\ddot{\mathbf{u}}(\tau) = \frac{1}{\mathbf{t}^2_{\mathbf{horizon}}}\left(\mathbf{c}_0^{\mathbf{u}}\frac{\mathbf{d}^2\mathbf{B}_0(\tau)}{\mathbf{d}\tau^2} + \mathbf{c}_1^{\mathbf{u}}\frac{\mathbf{d}^2\mathbf{B}_1(\tau)}{\mathbf{d}\tau^2} + \mathbf{c}_2^{\mathbf{u}}\mathbf{d}\frac{\mathbf{d}^2\mathbf{B}_2(\tau)}{\mathbf{d}\tau^2} + \mathbf{c}_3^{\mathbf{u}}\frac{\mathbf{d}^2\mathbf{B}_3(\tau)}{\mathbf{d}\tau^2} + \mathbf{c}_4^{\mathbf{u}}\frac{\mathbf{d}^2\mathbf{B}_4(\tau)}{\mathbf{d}\tau^2} + \mathbf{c}_5^{\mathbf{u}}\frac{\mathbf{d}^2\mathbf{B}_5(\tau)}{\mathbf{d}\tau^2} + \mathbf{c}_6^{\mathbf{u}}\frac{\mathbf{d}^2\mathbf{B}_6(\tau)}{\mathbf{d}\tau^2}\right)
\end{gathered} \tag{5-11}$$

Similarly for the *v* and *w* axes:

$$\begin{gathered}
\dot{\mathbf{v}}(\tau) = \frac{1}{\mathbf{t}_{\mathbf{horizon}}}\left(\mathbf{c}_0^{\mathbf{v}}\frac{\mathbf{dB}_0(\tau)}{\mathbf{d}\tau} + \mathbf{c}_1^{\mathbf{v}}\frac{\mathbf{dB}_1(\tau)}{\mathbf{d}\tau} + \mathbf{c}_2^{\mathbf{v}}\frac{\mathbf{dB}_2(\tau)}{\mathbf{d}\tau} + \mathbf{c}_3^{\mathbf{v}}\frac{\mathbf{dB}_3(\tau)}{\mathbf{d}\tau} + \mathbf{c}_4^{\mathbf{v}}\frac{\mathbf{dB}_4(\tau)}{\mathbf{d}\tau} + \mathbf{c}_5^{\mathbf{v}}\frac{\mathbf{dB}_5(\tau)}{\mathbf{d}\tau} + \mathbf{c}_6^{\mathbf{v}}\frac{\mathbf{dB}_6(\tau)}{\mathbf{d}\tau}\right) \\
\ddot{\mathbf{v}}(\tau) = \frac{1}{\mathbf{t}^2_{\mathbf{horizon}}}\left(\mathbf{c}_0^{\mathbf{v}}\frac{\mathbf{d}^2\mathbf{B}_0(\tau)}{\mathbf{d}\tau^2} + \mathbf{c}_1^{\mathbf{v}}\frac{\mathbf{d}^2\mathbf{B}_1(\tau)}{\mathbf{d}\tau^2} + \mathbf{c}_2^{\mathbf{v}}\frac{\mathbf{d}^2\mathbf{B}_2(\tau)}{\mathbf{d}\tau^2} + \mathbf{c}_3^{\mathbf{v}}\frac{\mathbf{d}^2\mathbf{B}_3(\tau)}{\mathbf{d}\tau^2} + \mathbf{c}_4^{\mathbf{v}}\frac{\mathbf{d}^2\mathbf{B}_4(\tau)}{\mathbf{d}\tau^2} + \mathbf{c}_5^{\mathbf{v}}\frac{\mathbf{d}^2\mathbf{B}_5(\tau)}{\mathbf{d}\tau^2} + \mathbf{c}_6^{\mathbf{v}}\frac{\mathbf{d}^2\mathbf{B}_6(\tau)}{\mathbf{d}\tau^2}\right)
\end{gathered} \tag{5-12}$$

$$\begin{gathered}
\dot{\mathbf{w}}(\tau) = \frac{1}{\mathbf{t}_{\mathbf{horizon}}}\left(\mathbf{c}_0^{\mathbf{w}}\frac{\mathbf{dB}_0(\tau)}{\mathbf{d}\tau} + \mathbf{c}_1^{\mathbf{w}}\frac{\mathbf{dB}_1(\tau)}{\mathbf{d}\tau} + \mathbf{c}_2^{\mathbf{w}}\frac{\mathbf{dB}_2(\tau)}{\mathbf{d}\tau} + \mathbf{c}_3^{\mathbf{w}}\frac{\mathbf{dB}_3(\tau)}{\mathbf{d}\tau} + \mathbf{c}_4^{\mathbf{w}}\frac{\mathbf{dB}_4(\tau)}{\mathbf{d}\tau} + \mathbf{c}_5^{\mathbf{w}}\frac{\mathbf{dB}_5(\tau)}{\mathbf{d}\tau} + \mathbf{c}_6^{\mathbf{w}}\frac{\mathbf{dB}_6(\tau)}{\mathbf{d}\tau}\right) \\
\ddot{\mathbf{w}}(\tau) = \frac{1}{\mathbf{t}^2_{\mathbf{horizon}}}\left(\mathbf{c}_0^{\mathbf{w}}\frac{\mathbf{d}^2\mathbf{B}_0(\tau)}{\mathbf{d}\tau^2} + \mathbf{c}_1^{\mathbf{w}}\frac{\mathbf{d}^2\mathbf{B}_1(\tau)}{\mathbf{d}\tau^2} + \mathbf{c}_2^{\mathbf{w}}\frac{\mathbf{d}^2\mathbf{B}_2(\tau)}{\mathbf{d}\tau^2} + \mathbf{c}_3^{\mathbf{w}}\frac{\mathbf{d}^2\mathbf{B}_3(\tau)}{\mathbf{d}\tau^2} + \mathbf{c}_4^{\mathbf{w}}\frac{\mathbf{d}^2\mathbf{B}_4(\tau)}{\mathbf{d}\tau^2} + \mathbf{c}_5^{\mathbf{w}}\frac{\mathbf{d}^2\mathbf{B}_5(\tau)}{\mathbf{d}\tau^2} + \mathbf{c}_6^{\mathbf{w}}\frac{\mathbf{d}^2\mathbf{B}_6(\tau)}{\mathbf{d}\tau^2}\right)
\end{gathered} \tag{5-13}$$

In order to evaluate equations (5-11), (5-12) and (5-13), the Bernstein basis functions defined in equation (5-4) must be differentiated with respect to τ, which after some manipulation, results in:

$$\begin{aligned}
\frac{\mathbf{dB}_0(\tau)}{\mathbf{d}\tau} &= -6 + 30\tau - 60\tau^2 + 60\tau^3 - 30\tau^4 + 6\tau^5 \\
\frac{\mathbf{dB}_1(\tau)}{\mathbf{d}\tau} &= 6 - 60\tau + 180\tau^2 - 240\tau^3 + 150\tau^4 - 36\tau^5 \\
\frac{\mathbf{dB}_2(\tau)}{\mathbf{d}\tau} &= 30\tau - 180\tau^2 + 360\tau^3 - 300\tau^4 + 90\tau^5 \\
\frac{\mathbf{dB}_3(\tau)}{\mathbf{d}\tau} &= 60\tau^2 - 240\tau^3 + 300\tau^4 - 120\tau^5 \\
\frac{\mathbf{dB}_4(\tau)}{\mathbf{d}\tau} &= 60\tau^3 - 150\tau^4 + 90\tau^5 \\
\frac{\mathbf{dB}_5(\tau)}{\mathbf{d}\tau} &= 30\tau^4 - 36\tau^5 \\
\frac{\mathbf{dB}_6(\tau)}{\mathbf{d}\tau} &= 6\tau^5
\end{aligned} \tag{5-14}$$

and

$$\begin{aligned}
\frac{\mathbf{d}^2\mathbf{B}_0(\tau)}{\mathbf{d}\tau^2} &= 30 - 120\tau + 180\tau^2 - 120\tau^3 + 30\tau^4 \\
\frac{\mathbf{d}^2\mathbf{B}_1(\tau)}{\mathbf{d}\tau^2} &= 60 + 360\tau - 720\tau^2 + 600\tau^3 - 180\tau^4 \\
\frac{\mathbf{d}^2\mathbf{B}_2(\tau)}{\mathbf{d}\tau^2} &= 30 - 360\tau + 1080\tau^2 - 1200\tau^3 + 450\tau^4 \\
\frac{\mathbf{d}^2\mathbf{B}_3(\tau)}{\mathbf{d}\tau^2} &= 120\tau - 720\tau^2 + 1200\tau^3 - 600\tau^4 \\
\frac{\mathbf{d}^2\mathbf{B}_4(\tau)}{\mathbf{d}\tau^2} &= 180\tau^2 - 600\tau^3 + 450\tau^4 \\
\frac{\mathbf{d}^2\mathbf{B}_5(\tau)}{\mathbf{d}\tau^2} &= 120\tau^3 - 180\tau^4 \\
\frac{\mathbf{d}^2\mathbf{B}_6(\tau)}{\mathbf{d}\tau^2} &= 30\tau^4
\end{aligned} \tag{5-15}$$

In order to evaluate the objective function (see section 5.6.1) receding horizon trajectories are discretised into *n* steps across the range of $0 \le \tau \le 1$. Each of these *n* points on the trajectory can be calculated by first discretising the basis functions into *n* steps, as shown in equation 5-16. The discretised basis function matrices for speed, acceleration and rate of acceleration can then all be calculated during the off-line phase for the range $0 \le \tau \le 1$. Therefore, given a set of design variables, the on-line calculation of the entire 4D speed, acceleration & rate of acceleration profiles requires only the multiplication of design coefficients against known constant matrices.

$$B = \begin{bmatrix} B_0(\tau_1) & B_0(\tau_2) & \dots & B_0(\tau_\eta) \\ B_1(\tau_1) & B_1(\tau_2) & \dots & B_1(\tau_\eta) \\ \vdots & \vdots & & \vdots \\ B_6(\tau_1) & B_6(\tau_2) & \dots & B_6(\tau_\eta) \end{bmatrix}$$

$$B' = \frac{dB}{d\tau} = \begin{bmatrix} \frac{dB_0(\tau_1)}{d\tau} & \frac{dB_0(\tau_2)}{d\tau} & \dots & \frac{dB_0(\tau_\eta)}{d\tau} \\ \frac{dB_1(\tau_1)}{d\tau} & \frac{dB_1(\tau_2)}{d\tau} & \dots & \frac{dB_1(\tau_\eta)}{d\tau} \\ \vdots & \vdots & & \vdots \\ \frac{dB_6(\tau_1)}{d\tau} & \frac{dB_6(\tau_2)}{d\tau} & \dots & \frac{dB_6(\tau_\eta)}{d\tau} \end{bmatrix}$$

$$B'' = \frac{d^2B}{d\tau^2} = \begin{bmatrix} \frac{d^2B_0(\tau_1)}{d\tau^2} & \frac{d^2B_0(\tau_2)}{d\tau^2} & \dots & \frac{d^2B_0(\tau_\eta)}{d\tau^2} \\ \frac{d^2B_1(\tau_1)}{d\tau^2} & \frac{d^2B_1(\tau_2)}{d\tau^2} & \dots & \frac{d^2B_1(\tau_\eta)}{d\tau^2} \\ \vdots & \vdots & & \vdots \\ \frac{d^2B_6(\tau_1)}{d\tau^2} & \frac{d^2B_6(\tau_2)}{d\tau^2} & \dots & \frac{d^2B_6(\tau_\eta)}{d\tau^2} \end{bmatrix}$$

(5-16)

The output space receding horizon trajectory profiles are therefore calculated from:

$$\begin{aligned} u &= C^{u^T} B \\ \dot{u} &= C^{u^T} B' \\ \ddot{u} &= C^{u^T} B'' \\ v &= C^{v^T} B \\ \dot{v} &= C^{v^T} B' \\ \ddot{v} &= C^{v^T} B'' \\ w &= C^{w^T} B \\ \dot{w} &= C^{w^T} B' \\ \ddot{w} &= C^{w^T} B'' \end{aligned} \tag{5-17}$$

Similarly, vehicle position profiles in all three axes may be calculated by integrating the basis functions with respect to time[33], to give:

[33] Note that in order to integrate the basis functions $\tau = t / t_{horizon}$ was substituted into equations (5-4) allowing integration with respect to time, rather than via the curve parameter (τ).

$$\begin{aligned}
\int_0^{t_{horizon}} B_0(t) &= t + \frac{-3}{t_{horizon}} t^2 + \frac{5}{t_{horizon}^2} t^3 + \frac{-5}{t_{horizon}^3} t^4 + \frac{3}{t_{horizon}^4} t^5 + \frac{-1}{t_{horizon}^5} t^6 + \frac{1}{7 \times t_{horizon}^6} t^7 \\
\int_0^{t_{horizon}} B_1(t) &= \frac{3}{t_{horizon}} t^2 + \frac{-10}{t_{horizon}^2} t^3 + \frac{15}{t_{horizon}^3} t^4 + \frac{-12}{t_{horizon}^4} t^5 + \frac{5}{t_{horizon}^5} t^6 + \frac{-6}{7 \times t_{horizon}^6} t^7 \\
\int_0^{t_{horizon}} B_2(t) &= \frac{5}{t_{horizon}^2} t^3 + \frac{-15}{t_{horizon}^3} t^4 + \frac{18}{t_{horizon}^4} t^5 + \frac{-10}{t_{horizon}^5} t^6 + \frac{15}{7 \times t_{horizon}^6} t^7 \\
\int_0^{t_{horizon}} B_3(t) &= \frac{5}{t_{horizon}^3} t^4 + \frac{-12}{t_{horizon}^4} t^5 + \frac{10}{t_{horizon}^5} t^6 + \frac{-20}{7 \times t_{horizon}^6} t^7 \\
\int_0^{t_{horizon}} B_4(t) &= \frac{3}{t_{horizon}^4} t^5 + \frac{-5}{t_{horizon}^5} t^6 + \frac{15}{7 \times t_{horizon}^6} t^7 \\
\int_0^{t_{horizon}} B_5(t) &= \frac{1}{t_{horizon}^5} t^6 + \frac{-6}{7 \times t_{horizon}^6} t^7 \\
\int_0^{t_{horizon}} B_6(t) &= \frac{1}{7 \times t_{horizon}^6} t^7
\end{aligned} \tag{5-18}$$

Where t = time.

As with the speed, acceleration and rate of acceleration profiles, each of these basis functions can be calculated, discretised and stored in advance:

$$B^{\mathrm{int}} = \begin{bmatrix} B_0^{\mathrm{int}}(\tau_1) & B_0^{\mathrm{int}}(\tau_2) & \dots & B_0^{\mathrm{int}}(\tau_\eta) \\ B_1^{\mathrm{int}}(\tau_1) & B_1^{\mathrm{int}}(\tau_2) & \dots & B_1^{\mathrm{int}}(\tau_\eta) \\ \vdots & \vdots & & \vdots \\ B_6^{\mathrm{int}}(\tau_1) & B_6^{\mathrm{int}}(\tau_2) & \dots & B_6^{\mathrm{int}}(\tau_\eta) \end{bmatrix} \tag{5-19}$$

Where:

$B_i^{\mathrm{int}}(\tau_1)$ = *i^{th} integrated Bezier basis function at the 1^{st} of n positions along the design horizon*

On-line calculation of forward, lateral and vertical position profiles is then reduced to the following matrix multiplications:

$$\begin{aligned} x &= x_0 + C^{u^T} B^{\mathrm{int}} \\ y &= y_0 + C^{v^T} B^{\mathrm{int}} \\ z &= z_0 + C^{w^T} B^{\mathrm{int}} \end{aligned} \tag{5-20}$$

5.3.6 Boundary Conditions

In order to ensure a smooth transition between the vehicle’s current state and the receding horizon, boundary conditions can be enforced based on the current sensed vehicle state. The following initial conditions are enforced for the quadrotor vehicle:

$$
\begin{aligned}
&u_0,\ v_0 \,\&\, w_0 \\
&\frac{\partial u_0}{\partial t},\ \frac{\partial v_0}{\partial t} \,\&\, \frac{\partial w_0}{\partial t} \\
&\frac{\partial^2 u_0}{\partial t^2},\ \frac{\partial^2 v_0}{\partial t^2} \,\&\, \frac{\partial^2 w_0}{\partial t^2}
\end{aligned}
\tag{5-21}
$$

These initial conditions can be enforced algebraically by setting $\tau = 0$ (for the initial condition) and manipulating the speed, acceleration and rate of acceleration equations (5-5), (5-11), (5-12) & (5-13) to give:

$$
\begin{aligned}
&\mathbf{c}_0^{\mathbf{u}} = \mathbf{u}_0 \\
&\mathbf{c}_1^{\mathbf{u}} = \frac{\dot{\mathbf{u}}_0 \times \mathbf{t}_{\mathbf{horizon}}}{6} + \mathbf{c}_0^{\mathbf{u}} \\
&\mathbf{c}_2^{\mathbf{u}} = \frac{\ddot{\mathbf{u}}_0 \times \mathbf{t}_{\mathbf{horizon}}^2}{30} - \mathbf{c}_0^{\mathbf{u}} + 2\mathbf{c}_2^{\mathbf{u}} \\
&\mathbf{c}_0^{\mathbf{v}} = \mathbf{v}_0 \\
&\mathbf{c}_1^{\mathbf{v}} = \frac{\dot{\mathbf{v}}_0 \times \mathbf{t}_{\mathbf{horizon}}}{6} + \mathbf{c}_0^{\mathbf{v}} \\
&\mathbf{c}_2^{\mathbf{v}} = \frac{\ddot{\mathbf{u}}_0 \times \mathbf{t}_{\mathbf{horizon}}^2}{30} - \mathbf{c}_0^{\mathbf{v}} + 2\mathbf{c}_2^{\mathbf{v}} \\
&\mathbf{c}_0^{\mathbf{w}} = \mathbf{w}_0 \\
&\mathbf{c}_1^{\mathbf{w}} = \frac{\dot{\mathbf{w}}_0 \times \mathbf{t}_{\mathbf{horizon}}}{6} + \mathbf{c}_0^{\mathbf{w}} \\
&\mathbf{c}_2^{\mathbf{w}} = \frac{\ddot{\mathbf{w}}_0 \times \mathbf{t}_{\mathbf{horizon}}^2}{30} - \mathbf{c}_0^{\mathbf{w}} + 2\mathbf{c}_2^{\mathbf{w}}
\end{aligned}
\tag{5-22}
$$

These nine equations reduce the design space from twenty-one variables, to twelve, significantly reducing the complexity of the optimisation problem. Note that BCs are only enforced at the beginning of the receding horizon trajectory. This is a design choice, that allows the vehicle state at the end of the design horizon to vary as necessary in order to provide the optimal (as defined by the cost function) trajectory.

5.3.7 Advantages of Polynomial Trajectory Description

The primary advantages of employing a polynomial trajectory description include:

- *Smoothness* - Fundamentally, vehicle trajectories are smooth, rather than jagged or discontinuous, and polynomials provide this characteristic. Random or probabilistic approaches will typically result in discontinuous trajectories, therefore requiring an additional smoothing process via some form of polynomial curve fitting.
- *Varying degrees of aggression* - When imposing performance limits via polynomials all manoeuvres up to the defined limits will be allowed, therefore allowing gentle manoeuvring to occur as easily as maximum performance manoeuvres. This a strong advantage over methods that propagate trajectories

using the maximum manoeuvres limits (e.g. Dubin's turns). These approaches result in maximum performance manoeuvres being expected when in typically low-gain flight regimes.

- *Variable resolution* - The continuous polynomial curve can be discretized into any resolution that suits the system performance and computational requirements without altering the smoothness of the curve. Additionally, different axes can employ different curve resolutions or order to suit the dynamics of the vehicle.
- *Analytical enforcement of boundary conditions (BC)* - Enforcing BCs means that the smoothness of the designed trajectory from the current vehicle state can be ensured. Additionally, end conditions can be enforced if required. Each imposed BC reduces the number of design variables by one, reducing the dimension of the design problem.
- *Efficient description of a trajectory* - If the basis functions of the polynomial are known then it is fully described by the design variables only. For example, a 6th order curve is described by only seven constant values. This allows efficient storage and transmission of trajectory information, for example for a cooperative collision avoidance scheme where the MAVs transmit their trajectory intentions to each other.
- *Analytical calculation of curve differentials* - It is possible to calculate time differentials of polynomials analytically, therefore allowing performance limits and BCs to be imposed on rates, acceleration, rates of accelerations etc. with minimal additional calculation.
- *Rapid on-line calculation* - As the discretised basis functions for position, speed, acceleration and rate of acceleration across the design horizon can all be calculated and stored off-line, the on-line calculation of the full receding horizon description reduces to a single 7xn matrix multiplication for each profile.

5.4 Vehicle Performance Modelling and Constraint Handling

5.4.1 Vehicle Performance Map

Care must be taken with an output space design in order to ensure that the resulting trajectories are achievable given the performance constraints of the vehicle. One of the goals of this work is to reduce on-line computation effort, therefore the approach taken is to translate the vehicle manoeuvre capabilities into a performance map that can be rapidly accessed on-line. This performance map can be generated off-line from a complex nonlinear vehicle model, allowing it to be scheduled around the flight envelope to the required level of fidelity. Even for a highly detailed performance map the data can be stored in a look-up-table, allowing it to be rapidly accessed during the trajectory optimisation stage.

The performance map must account for the limitations imposed on the vehicle by both the available control effectors and the lower level autopilot and inner-loop control modes. For example, certain vehicle types may exhibit aerodynamic instabilities, aeroelastic effects, or complex control nonlinearities / coupling that may require the low level control loops to restrict control power to less that that available from a static aerodynamic analysis of the vehicle. The performance map must account for any such effects, and therefore becomes an interface between the local motion planning layer and the lower level control loops, defining the agreed and achievable level of control power around the flight envelope.

5.4.2 Quadrotor Implementation

In order to simplify the local trajectory optimisation process, it is desirable to chose the format of the performance map such that any coupling between performance parameters is minimised. Simulation tests with the quadrotor have shown that during typical operation the forward and lateral performance limits are largely decoupled due to the low control effort required. However, as the maximum speed is approached in one axis the available control power for the other axis is reduced, therefore some coupling does exist. A greater degree of coupling exists between the vertical axis and both forward and lateral axes. This is due to the higher percentage of available thrust that is required for vertical control than horizontal control. However, it should be feasible for this degree of coupling to be minimised or removed altogether by ensuring suitable control power margins are always available in each axis. A suitable format for specification of the vehicle performance map would therefore be:

- *Minimum / maximum forward speed, acceleration & rate of change of acceleration.*
- *Minimum / maximum lateral speed, acceleration & rate of change of acceleration.*
- *Minimum / maximum vertical speed, acceleration & rate of change of acceleration.*

Generation of the performance map data can be achieved using the vehicle simulation model to perform a series of performance tests at different spot points in the flight envelope. For example:

```
→ For m values of altitude between zero and maximum altitude:
    → For n values of forward speed between min / max values:
        → For p values of lateral speed between min / max values:
            → Perform maximum u-axis acceleration & deceleration tests
            → Perform maximum v-axis acceleration & deceleration tests
            → Perform maximum w-axis acceleration & deceleration tests (including
              both climb and descent)
```

The above tests provide maximum response data from a set of trim points. It is also feasible to extend these tests to also include response characteristics from accelerating states, e.g. stepping up and down current values of acceleration. Data from set point tests can then be entered into a look-up table, allowing interpolation between data from individual spot tests. It is clear that by performing a number of such tests it is possible to generate a detailed map of the vehicle performance constraints.

Simulation tests performed within this thesis are limited to a small portion of the quadrotor's flight envelope, therefore a detailed performance map of the quadrotor was not produced. However, a limited set of tests were conducted, and used to generate a basic set of quadrotor performance constraints for use within this thesis. Examples of the quadrotor's performance shown in Figure 32 and Figure 33, and the derived performance constraints are shown in Table 2. Enforcement of these performance constraints during the optimisation process is discussed in the next section.

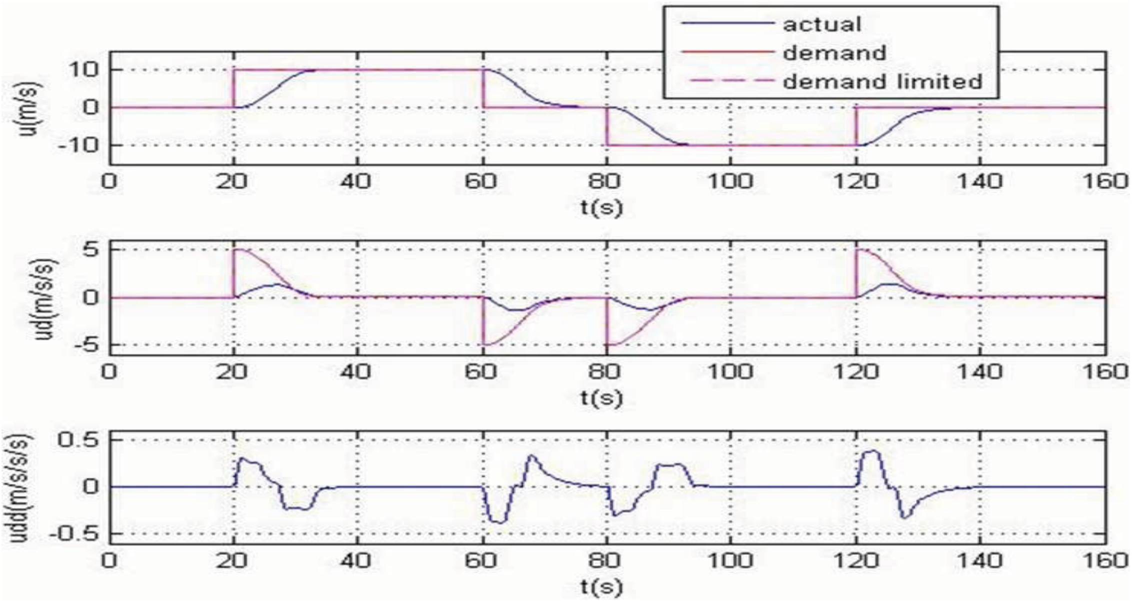


Figure 32 - Performance Test, U (or V) Axis

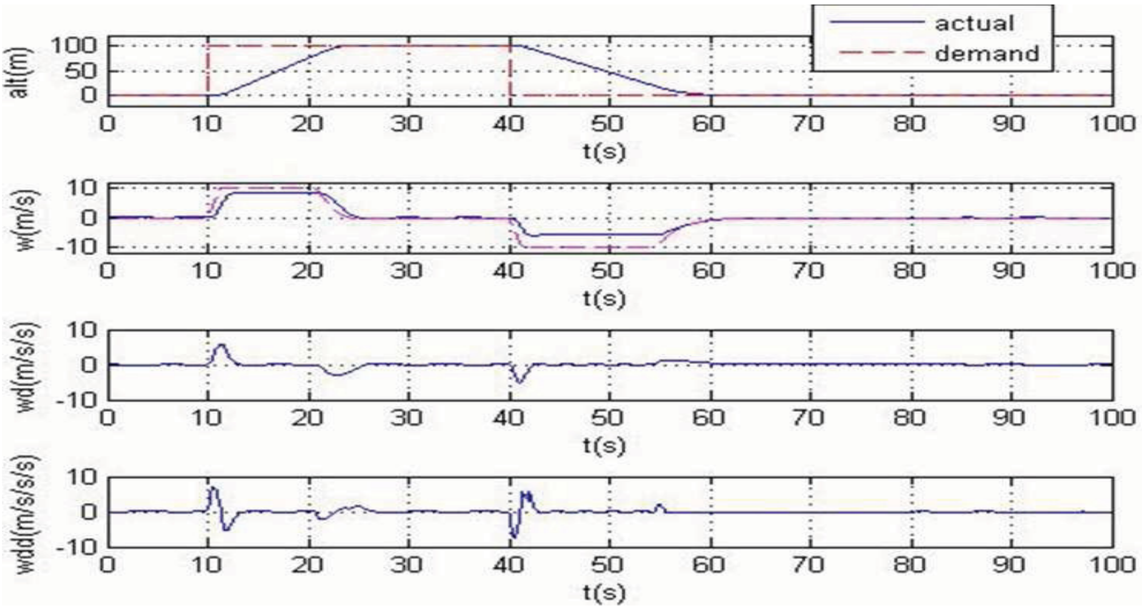


Figure 33 - Performance Test, W axis

	u (&v) (m/s)	$\dot{u}$ (&$\dot{v}$) (m/s^2)	$\ddot{u}$ (&$\ddot{v}$) (m/s^3)	w (m/s)	$\dot{w}$ (m/s^2)	$\ddot{w}$ (m/s^3)
Min	-10	-2	-0.4	-6	-7	-6
Max	10	2	0.4	9	7	6

Table 2 - Example Quadrotor Vehicle Performance Constraints

5.4.3 Enforcing vehicle performance constraints

Many different methods exist for enforcement of performance constraints, e.g. Lagrange multipliers, constrained variations, feasible directions etc. (section 3.4). The approach used here is a penalty function method, where the cost function is augmented to include additional terms to punish performance values that approach or exceed the defined limits. The penalty function used is the Yukawa potential function shown in equation (5-23).

$$C_{performance} = A * \frac{e^{(-\alpha d)}}{d} \tag{5-23}$$

where:

$C_{performance}$ *= performance constraint penalty term to be added to the cost function*

A *= potential function scaling factor*

α *= potential function decay rate*

d *= performance margin (%)*

The performance margin (*d*) is defined as the percentage difference from the current vehicle state to the defined limit, and can be calculated as follows:

$$\begin{aligned} &d = 100 - \left(\frac{state\_current}{state\_\max} \right) \times 100 \\ &if \\ &d \le d_{\min} \\ &then \\ &d = d_{\min} \end{aligned} \tag{5-24}$$

where:

$d_{\min}$ *= a defined minimum performance margin to maintain* $d > 0$*.*

The behaviour of the potential function can be designed to suit the required performance by choice of the scaling factor (A) and decay rate (α). The impact of these design parameters is shown in Figure 34, where it can be seen that as the performance margin reduces towards zero, the cost term rises towards infinity. This results in a performance barrier that maintains the design in the feasible search space, and is therefore an interior point method.

In order to evaluate the cost function, candidate trajectories are discretised across the design horizon into *n* steps, with each performance constraint enforced at each step. The performance constraints to be enforced are minimum and maximum values of $u, \dot{u}, \ddot{u}, v, \dot{v}, \ddot{v}, w, \dot{w}, \ddot{w}$ (as shown in Table 2), therefore three separate *u, v & w* trajectory profiles result in 2 x *n* x 9 proximity function calculations for each candidate trajectory, with the total performance cost equal to the summation of all individual performance constraint costs. The choice of the potential function method for handling these constraints is a key component of the framework, and has a significant impact on the computational effort of the approach.

The performance constraint penalty function design parameters used for the simulation tests shown within this thesis are as follows:

A = 500,000, α = 0.6

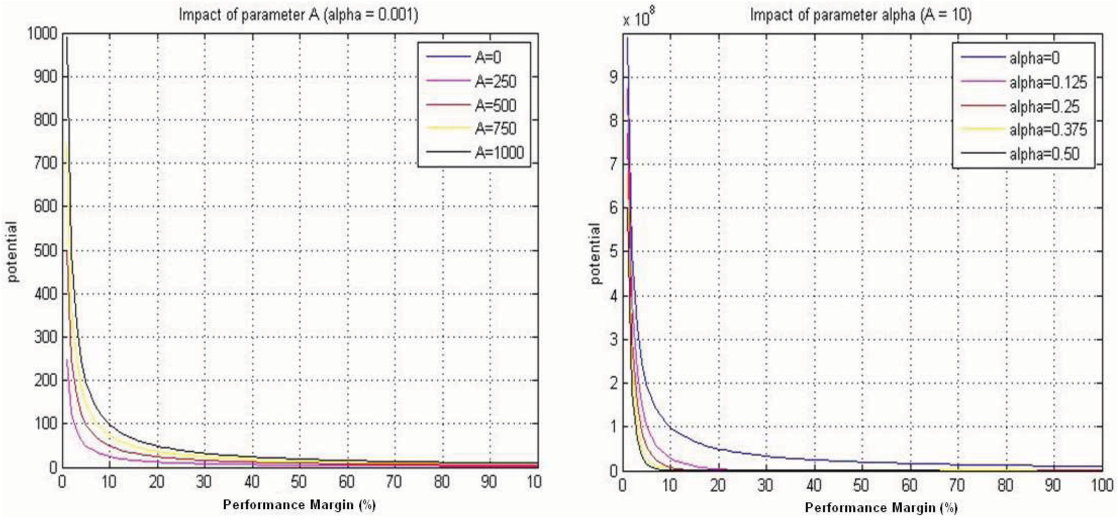


Figure 34 - Yukawa potential function design parameters

5.4.4 Impact of Performance Limits on the Feasible Design Space

It is possible to visualise the impact of these performance constraints on a single fifth order Bezier curve. A fifth order curve has six design variables, therefore after imposing three boundary conditions there are only three variables left, allowing the design space to be visualised in three dimensions. A fifth order curve describing speed in the u-axis is given by:

$$u(\tau) = c_0^u B_0(\tau) + c_1^u B_1(\tau) + c_2^u B_2(\tau) + c_3^u B_3(\tau) + c_4^u B_4(\tau) + c_5^u B_5(\tau)$$

Imposing initial conditions for speed, acceleration and rate of acceleration defines the variables $c_0^u, c_1^u \; \& \; c_2^u$, therefore simplifying the speed description to:

$$u(\tau) = k_1 B_0(\tau) + k_2 B_1(\tau) + k_3 B_2(\tau) + c_3^u B_3(\tau) + c_4^u B_4(\tau) + c_5^u B_5(\tau)$$

An example constraints (maximum speed) is given by:

$$u(\tau) = k_1 B_0(\tau) + k_2 B_1(\tau) + k_3 B_2(\tau) + c_3^u B_3(\tau) + c_4^u B_4(\tau) + c_5^u B_5(\tau) \le u_{\max}$$

the boundary of which may be visualised by setting:

$$k_1 B_0(\tau) + k_2 B_1(\tau) + k_3 B_2(\tau) + c_3^u B_3(\tau) + c_4^u B_4(\tau) + c_5^u B_5(\tau) = u_{\max}$$

which defines a surface in 3D space for each point on the speed profile. The visualisation shown in Figure 35 results from repeating the above for fifty points over the curve for the following constraints:

- speed limits = +/- 45 m/s
- acceleration +/- 15 m/s^2
- rate of acceleration +/- 5m/s^3

Also shown in Figure 35 is the resulting feasible design space which can be seen to have been reduced to a single bounded region. It is important to note that the performance constraints do not annex this feasible design space into several disconnected regions, therefore reducing the likelihood of the performance constraints creating local minimums in the cost function.

It can therefore be argued that the nature of vehicle performance limits are suited to the penalty function method of constraint. The minimum and maximum limits create an outer boundary to the feasible design space, with the majority of vehicle motion occurring well away from the limits. This suggests that during routine non-performance limited manoeuvres the constraints will play no part in the optimisation algorithm, therefore ensuring that the additional penalty terms do not adversely affect cost function behaviour.

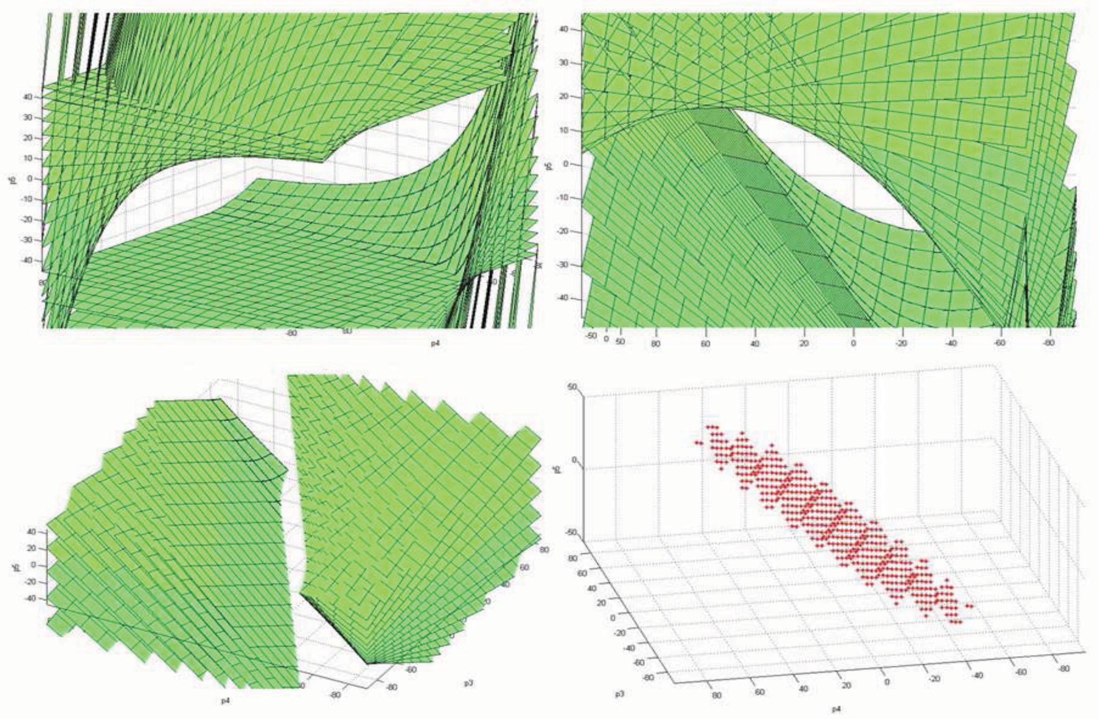


Figure 35 - Impact of speed (above left), acceleration (above right) and rate of acceleration (below left) constraints on the resulting feasible search space (below right).

5.5 Obstacle Modelling & Constraint Handling

5.5.1 Obstacle Modelling

Two types of obstacles are considered, those known *a priori* (e.g. building layout of the operational environment) and those detected *en route* (e.g. overhead cables, trees and other forms of urban clutter). For modelling purposes the difference between these types is that 3D or depth information is likely to be available only for *a priori* obstacles. This is illustrated in Figure 36 where it is clear that an obstacle face may be detected, but the region shaded by it will remain unknown.

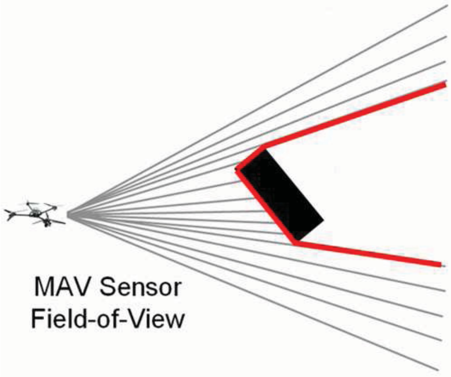


Figure 36 - *En route* obstacle detection

All obstacles are modelled using either cuboids, spheres or cylinders, as shown in the example obstacle set in Figure 37. If necessary, complex obstacles may be formed from several of these ‘primitives’. There are two primary advantages of using simple geometric descriptions. Firstly, both collision checks and the calculation of proximity information is simplified, therefore reducing on-line calculation. This process can also be aided by transforming the obstacle space to align it with the environment space (e.g. x, y, z axis all aligned). Secondly, standard geometric shapes acts as an efficient parameterisation of the obstacle, therefore reducing memory requirements. During the optimisation process many candidate trajectories are considered, each requiring consideration of the obstacle space, therefore obstacle representation is a key contributor to overall computational effort. If desired, ellipsoids may also be used to provide a better fit to certain obstacles, while also allowing simple calculations[34], but this was not performed.

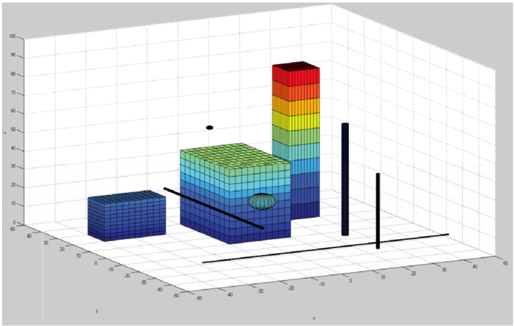


Figure 37 - Example obstacle models

Real-time generation of a 4D obstacle map is not the focus of this work, therefore for the remainder of this thesis it is assumed that real-time obstacle information is available. Enforcement of obstacle constraints during optimisation is discussed in the next section.

[34] For calculation of proximity / intersection data ellipsoids can be converted into spheres via an axis transformation.

5.5.2 Enforcing Obstacle Constraints

As with the performance constraints, many methods also exist for handling obstacle constraints, with the two primary options being:

- *Constraints to the optimisation process*
- *Penalty function terms in the cost function*

The presence of obstacles introduces additional complexity into the problem, therefore the choice presented by the above two options is whether this complexity is best handled in the search algorithm or the cost function. With the first approach the feasible search space no longer includes the known obstacle positions, therefore significantly complicating the optimisation process. Additionally, 3D obstacle information is required, and as mentioned previously this may not be available for obstacles detected by on-board sensors. An additional disadvantage is that although the obstacles themselves are infeasible there is no punishment for a near miss, and for the UAV local motion planning problem this is not desirable. Although collisions do occur as discrete events (e.g. collision or no collision), it is preferable to employ an approach that treats obstacles in a manner closer to human behaviour, where the safe clearance distance may vary depending on many factors (e.g. own speed, obstacle speed, own manoeuvrability, obstacle manoeuvrability, etc.).

Although penalty function methods complicate the objective function, they benefit from the major advantage that the design space is unhindered by constraints, therefore simplifying the search process. Additionally, they offer the potential to provide the human type behaviour discussed above rather than the binary collision / no-collision approach. The approach taken within this work for handling obstacle constraints is therefore the penalty function method, again using the Yukawa potential function, this time punishing proximity to each obstacle, as shown below:

$$C_{obstacle} = A * \frac{e^{(-\alpha d)}}{d} \tag{5-25}$$

where:

$C_{obstacle}$ = *obstacle constraint penalty term to be added to the cost function*

A = *potential function scaling factor*

α = *potential function decay rate*

d = *distance of the nearest point on the obstacle to the point of interest*

An example proximity field generated from equation 5-25 about a point obstacle is shown in Figure 38, where the repelling effect of the cost term is clear. The proximity field tends to infinity as the distance tends to zero, therefore for finite obstacles a minimum proximity distance must be enforced, which results in the cost term saturating for collision points. In order to drive the search algorithm away from the collision a simple linear obstacle intersection model is also used. This is illustrated in Figure 39, which shows a horizontal ‘slice’ of the proximity cost around a sphere both with and without the linear intersection term. Note that the intersection cost term can be based on proximity of the nearest obstacle face only, therefore it does not require full 3D obstacle information. It should also be noted that this combined proximity / intersection cost model is discontinuous at the point where the two models meet, therefore gradient information may be inaccurate. This was not found to be problematic for the search algorithm, but could be remedied if necessary by use of a nonlinear intersection algorithm.

For evaluation of the objective function (section 5.6.1) the receding horizon trajectories are discretised into *n* points, therefore the obstacle proximity calculation is repeated for each point along each candidate trajectory, with the total cost for each obstacle equalling the summation of all cost terms at each of the *n* horizon points. The desired safe clearance distance may be enforced by specification of the design parameters, with the impact of each shown previously in Figure 34. Multiple obstacles are handled by simply summing the total cost terms from each individual obstacle.

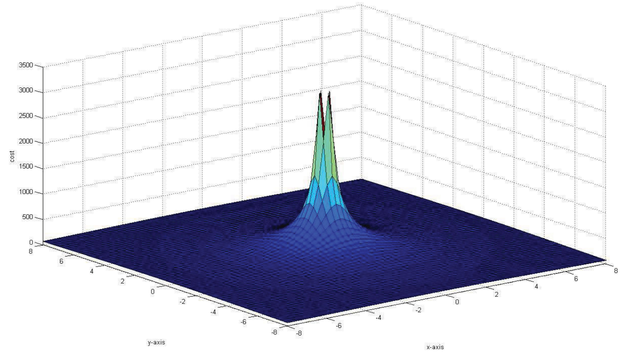


Figure 38 - Example proximity field around a point obstacle

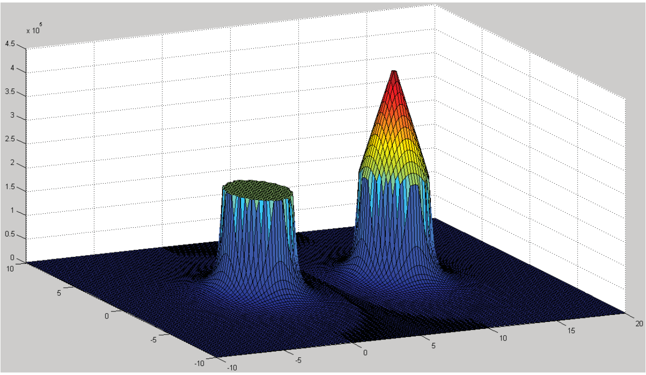


Figure 39 - Proximity and proximity + intersection cost models for finite obstacles

5.5.3 Dynamic obstacles

Dynamic obstacles can be handled by providing predictions regarding the position of each known obstacle across the receding design horizon. The proximity calculation at each point over the design horizon can then be based on the predicted obstacle location at that time. For the simulations shown in this thesis obstacle prediction was based on a simple propagation of the current sensed obstacle state over the design horizon. A more complex prediction method (e.g. Kalman filter) which also considers the history of obstacle state rather than just instantaneous state would be preferred for a real application, but was beyond the scope of this thesis.

Another issue to be considered for dynamic obstacles is the uncertainty of predicted obstacle positions over the receding horizon. Fundamentally, this uncertainty increases over the design horizon, with longer horizons resulting in greater uncertainty. Examples of this are shown in Figure 40 for horizon lengths of 4, 6, 8 & 10 seconds, based on the assumption that the manoeuvre limits of the obstacle match those of the quadrotor UAV. The growth in uncertainty with design horizon is clear. The human response to this uncertainty is to use judgement regarding the manoeuvre limits and likely behaviour of the obstacle to provide an estimate of its achievable positions over the horizon. This allows a defensive strategy to be implemented, effectively staying out of reach of the obstacle. However, without the human in the loop estimating such performance limits and the resulting likely behaviour would be difficult, requiring a combination of obstacle recognition plus a database of associated performance limits.

Within this thesis a pragmatic approach was taken to this issue, basing predictions on current obstacle state data, but increasing required clearance distances across the design horizon. An example of this is shown in Figure 41, which for comparison also displays the maximum manoeuvre envelope limits of the quadrotor vehicle. As well as partially accounting for uncertainty, this approach also helps to provide strong intent signalling that aids the efficiency of decentralised uncooperative deconfliction as discussed further in Chapter 7.

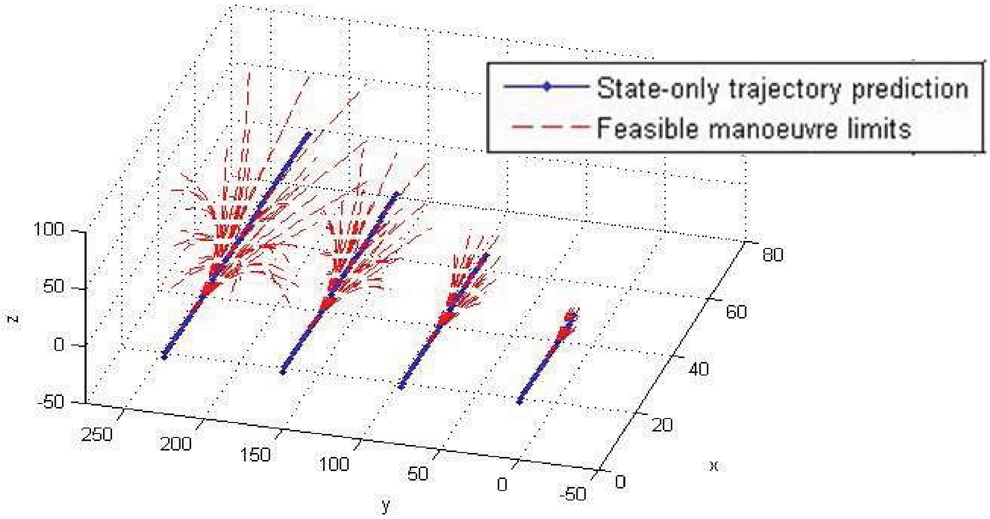


Figure 40 - Impact of design horizon on prediction certainty

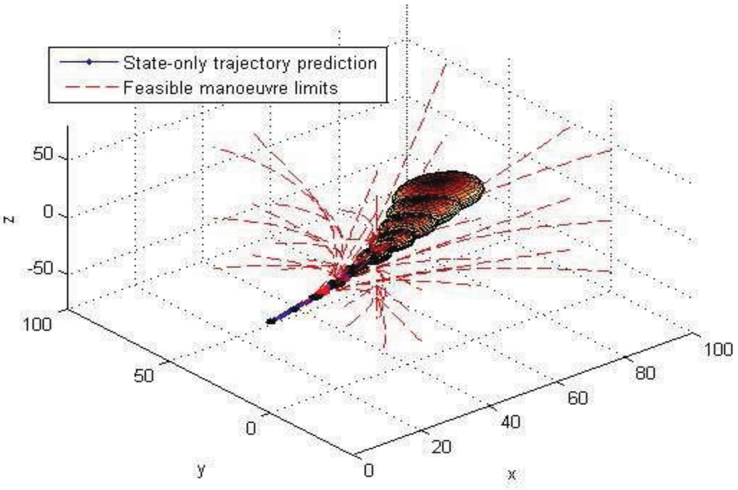


Figure 41 - Increased Obstacle Clearance Distance over the Prediction Horizon

5.5.4 *Relative speed bias to obstacle cost*

The obstacle cost function described in section 5.5.2 is based purely on proximity to the obstacle. Intuitively, this doesn't match human behaviour where the relative speed between the vehicle and obstacle is likely to impact on the desired clearance distance. Examples of this include:

- *Risk / consequence* - A high speed collision is likely to result in greater damage.
- *Sensitivity to unexpected manoeuvres* - A high relative speed also reduces the time available to deal with unexpected obstacle motion or disturbances.

The obstacle cost function can be enhanced to account for this behaviour by varying either of the obstacle cost potential function (equation 5-25) design parameters (*A* & α) with relative vehicle / obstacle speed. The implementation method chosen here was increasing the obstacle cost scaling factor (*A*) linearly with relative speed squared as shown in Figure 42. The total relative speed and resulting scaling factor were limited to maximum values as shown in Figure 42, leading to a variation in obstacle cost with relative speed shown in Figure 43.

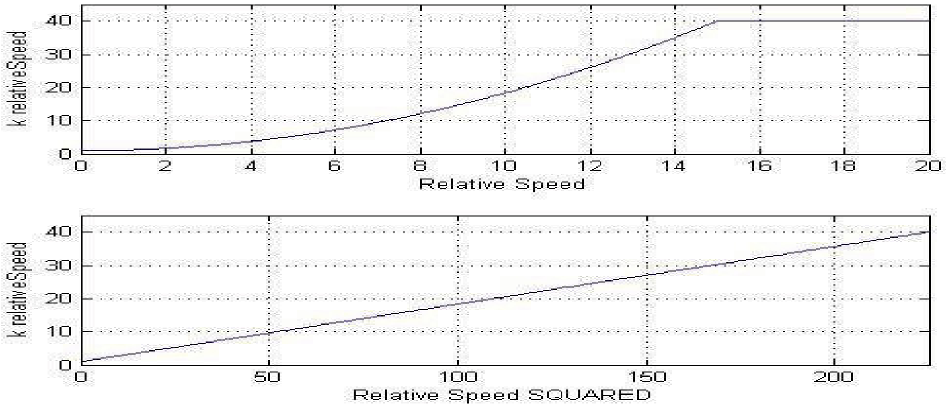


Figure 42 - Calculation of Relative Speed Scaling Factor

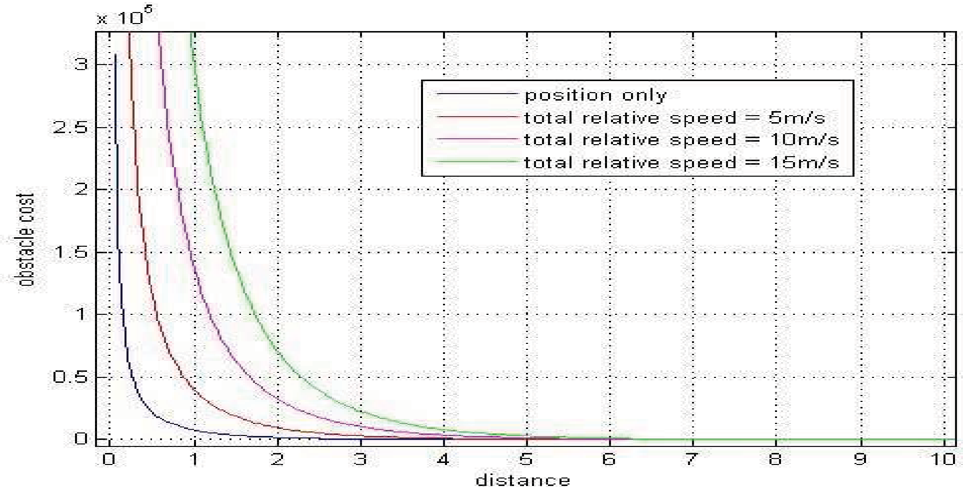


Figure 43 - Impact of Relative Speed on Obstacle Cost

5.6 Optimisation

The role of the optimisation algorithm is to find the set of design variables that minimise a defined objective function, given a set of constraints (e.g. initial state, performance limits, obstacles). A general discussion of optimisation approaches was provided in section 3.4, and the aim of this section is to review the implementation used within this local motion planning and control framework. The primary features of the developed optimisation approach include:

- *Objective function*
- *Search algorithm*
- *Avoiding local minimums*

Each of these topics is discussed in detail below.

5.6.1 Objective Function

The previously defined situation aware trajectory tracking problem (Section 4.5) can be solved by minimising the following objective function:

$$J^{total} = \sum_{i=1}^{n} \left\{ \lambda_{position} J_i^{position} + \lambda_{speed} J_i^{speed} + \lambda_{performance} J_i^{performance} + \lambda_{obstacle} J_i^{obstacle} \right\} + \lambda_{terminal} J^{terminal}$$

(5-26)

where:

$$J_i^{position} = \left(x_i^{demand} - x_i^{actual}\right)^2 + \left(y_i^{demand} - y_i^{actual}\right)^2 + \left(z_i^{demand} - z_i^{actual}\right)^2 \tag{5-27}$$

$$J_i^{speed} = \left(u_i^{demand} - u_i^{actual}\right)^2 + \left(v_i^{demand} - v_i^{actual}\right)^2 + \left(w_i^{demand} - w_i^{actual}\right)^2 \tag{5-28}$$

$$J_i^{performance} = \sum_{j=1}^{p} A_{performance} \frac{e^{-\alpha_{performance} d_{performance}}}{d_{performance}} \tag{5-29}$$

$$J_i^{obstacle} = \sum_{j=1}^{m} \left(A_{obstacle} \frac{e^{-\alpha_{obstacle} d_{obstacle}}}{d_{obstacle}} + J_{interception} \right) \qquad (5\text{-}30)$$

$$J^{terminal} = \lambda_{hdg} \left(\varphi_n^{demand} - \varphi_n^{actual} \right)^2 + \lambda_{fpa} \left(\gamma_n^{demand} - \gamma_n^{actual} \right)^2 \qquad (5\text{-}31)$$

λ = scaling factor for individual cost terms

n = number of points in the design horizon to be costed (e.g. trajectory resolution)

φ = heading angle

γ = flight path angle

p = number of performance constraints to be implemented

m = number of known obstacles

$J_{interception}$ = linear intersection cost model as discussed in section 0

This objective function defined in equations 5-25 to 5-31 represents the mathematical formulation of the performance priorities of the system. Each final trajectory results from a balance of cost terms as defined by the scaling parameters (λ)*.* For example, the obstacle clearance distance is defined by the cost balance between the trajectory tracking terms (position and speed) against obstacle proximity. It is feasible that these performance priorities may naturally vary during the course of a mission which introduces the potential of a situation dependent cost function, although this was not explored within this thesis.

5.6.2 Gradient Based Search

The overriding advantage of handling both obstacle and vehicle performance constraints via penalty function methods is that it allows the search algorithm to proceed unconstrained. This is a significant simplification, and helps to enable the goal of rapid optimisation for high rate trajectory design.

A steepest descent based search is employed, with a finite line search used for each calculated search direction. The primary advantage of the steepest descent approach is it's simplicity and speed, which again is aimed towards allowing rapid convergence for real-time implementation. The direction of steepest descent is generally a local property, therefore a conjugate gradient approach (Fletcher-Reeves) was also tested, although this was not found to improve convergence rates. The main disadvantage of a gradient based approach is that it can become trapped in local minimums of the cost function, the likelihood of which depends on the particular problem. This issue is addressed in section 5.6.3, which discusses the use of a coarse grid of the search space to escape local minimums.

The steepest descent search direction is given by:

$$S = -\nabla J(C) \qquad (5\text{-}32)$$

Where:

$J(C)$ is the value of the cost function (equation 5-26) for a the current design C, and

$$\nabla J(C)=\begin{bmatrix}\frac{\partial J}{\partial C_1} & \frac{\partial J}{\partial C_2} & \cdots & \frac{\partial J}{\partial C_{12}}\end{bmatrix} \tag{5-33}$$

The individual cost function gradient terms in equation 5-33 are approximated using a central difference formula as shown in equation 5-34.

$$\frac{\partial J}{\partial C_i}=\frac{J(C_i+\Delta C)-J(C_i-\Delta C)}{2\times\Delta C} \tag{5-34}$$

As discussed in section 5.3.4 and section 5.3.6, the design variables (C) are the set of twelve coefficients of the polynomial curves describing the speed profiles of the vehicle in the u, v & w axis. When evaluating equation (5-34) for each design variable care must be taken to check whether a detected stationary point is a maximum, minimum or a saddle point. If a maximum is detected then the most beneficial direction is used. If a minimum is detected then the gradient is left at zero, and if a saddle point is detected then the gradient is updated to suit the appropriate direction.

Once the desired search direction (S) has been calculated, a new design point can be calculated from:

$$C_{new}=C_{current}+k\cdot S \tag{5-35}$$

The step size (k) is calculated via a line search that attempts to minimise the cost function while only travelling in the direction S. A simple fixed step size search is implemented, although various methods are available to improve performance if required (e.g. accelerated step, golden section, quadratic / cubic interpolation methods, Newton / quasi Newton etc). This line search terminates when either the cost of the current design increases, or a maximum number of steps are taken. The direction of steepest descent is then re-calculated before performing another line search.

The overall algorithm is therefore comprised of a series of major iterations and minor iterations. For each major iteration the direction of steepest descent is calculated and then used to provide the search direction for the minor iterations where many different step sizes in this direction are explored. The overall search time can be limited by setting a maximum number of major and minor iterations.

The main computational effort during the search is evaluation of the cost function, which is driven primarily by three factors i) resolution of the trajectories ii) number of known obstacles and iii) number of performance constraints. For set values of these parameters the total computational effort required by the algorithm can be estimated as follows:

- *q = maximum number of major iterations*
- *r = maximum number of minor iterations*

- *s = number of design variables*
- *Calculation of the direction of steepest descent requires two evaluations of the cost function for each design variable, giving a total of 2s evaluations for each major iteration.*
- *Each minor iteration requires a single evaluation of the cost function, therefore resulting in a maximum number of r evaluations for each major iteration.*
- *The total number of cost function evaluations per search is therefore given by: q x (2s + r).*
- *The default set-up for the simulations shown in this thesis was as follows: q = 20, r = 100, s = 12, therefore the maximum number of cost function evaluations = 2,480.*

Total optimisation effort is typically reduced by terminating both the line search and overall search if the cost decrease is below a set tolerance. This convergence criteria may also be set to match current mission performance requirements, e.g. during critical mission phases only accept position errors of less than a defined tolerance. Alternatively, this output performance driven optimization is that the search may be allowed to 'relax' during operation in less complex environments such as during a transit to the area of interest.

5.6.3 Avoiding Local Minimums

The main disadvantage of a gradient based search is that it is susceptible to getting trapped in local minimums in the cost function. For the defined local motion planning problem the primary source of local minimums are obstacles. As discussed previously, the performance constraints tend to act as an enclosing boundary around the entire search space, and therefore are less likely to result in local minimums.

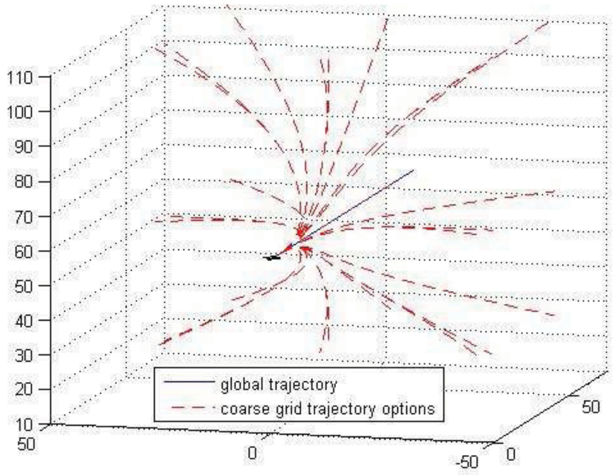


Figure 44 - Coarse grid of feasible design space

The approach taken here to escape local minimums and ensure sufficient coverage of the overall design space is to create a coarse grid of the feasible design space that allows a series of set design points to be compared to the solution from the gradient search. An example of a candidate trajectory set that results from this type of grid for the quadrotor vehicle is shown in Figure 44. This set of trajectories can be selected to provide a range of manoeuvres in each axis given the current vehicle state and performance limits, therefore ensuring that maximum performance manoeuvres in each axis are always available if required. The resolution of the grid may be set to match performance or memory requirements also employing non-steady manoeuvres (e.g. s-shape) in different axis if desired.

This set of feasible manoeuvres provides fundamentally useful information for any vehicle, manned or unmanned. For example, for a ground vehicle such as a manned road car, it can be used for visual indication of braking distances, turning circles, cornering performance, acceleration performance etc. This information can usefully be used either as indications / warning to a driver or as a design aid for unmanned navigation.

A number of options for the creation of this trajectory set were considered, including:

- *Library of pre-calculated manoeuvres* - This library may be calculated in advance from either actual flight data or the full nonlinear vehicle model, therefore ensuring dynamic feasibility. However, implementation issues include ensuring that the trajectories are feasible given the current vehicle state and storage of a high number of trajectories.
- *Direct grid of current feasible search space* - Use the current set of performance constraints to calculate / estimate a regular grid of the feasible design space (see section 5.4.4). This approach is able to impose boundary conditions directly, but calculation / estimation of this grid may be complex.
- *Optimise for manoeuvres in a defined set of directions* - Use a simple search algorithm to calculate a set of feasible trajectory options beginning from the current state. This search would be simpler than the final search as each option would simply aim to maximise performance in a different axis, e.g. max climb at constant speed, max turn while accelerating to max speed. This approach is likely to be computationally expensive.
- *On-line vehicle control model (e.g. nonlinear 6dof)* - Use an on-board vehicle model to propagate the current vehicle state forward in time while applying a set of control inputs, e.g. max. stick right, max. stick left, max thrust, etc. Feasibility is guaranteed by the complex vehicle model and boundary conditions can be handled. However, relying on a vehicle model is potentially computationally intensive.
- *On-line vehicle performance model* - Rather than applying control inputs, propagate the vehicle state entirely in the output space restraining trajectories by the vehicle performance data stored in the on-board performance map (see section 5.4.1).

The implementation approach used here is the on-line vehicle performance model. The primary advantages of which are that boundary conditions can be easily implemented, the vehicle performance model matches that used to provide constraints to the gradient search algorithm, and that as the set of candidate trajectory can be generated rapidly on-line there is no need to store a large library of candidate manoeuvres.

Generation of this set of trajectories is performed using a set of desired speeds in each axis to propagate the current vehicle state towards. The speed demands used to create the coarse grid shown in Figure 44 are shown below:

- *Forward speed options* = $\begin{bmatrix} u_{\min} & 0 & u_{\max} \end{bmatrix}$
- *Lateral speed options* = $\begin{bmatrix} v_{\min} & 0 & v_{\max} \end{bmatrix}$
- *Vertical speed options* = $\begin{bmatrix} w_{\min} & 0 & w_{\max} \end{bmatrix}$

This coarse grid is therefore comprised of three options in each of three axis, giving a total of $3^3 = 27$ candidate trajectories. Ideally, the feasible manoeuvre grid would be finely spaced, but there is obviously a computation cost associated with each option that is tested. The final grid size is a compromise between computational effort and reward in terms of avoidance of local minimums.

The coarse grid trajectory options were calculated as follows:

- *For each of a set of desired speed values in each axis*
  - *Initial all states (* $\mathbf{u}, \mathbf{v}, \mathbf{w}, \dot{\mathbf{u}}, \dot{\mathbf{v}}, \dot{\mathbf{w}}, \ddot{\mathbf{u}}, \ddot{\mathbf{v}}, \ddot{\mathbf{w}}$ *) to the specified initial conditions*
  - *Propagate u, v & w from their initial values towards the desired values using max. performance limits define in the vehicle performance map (* $\dot{\mathbf{u}}_{\max}, \dot{\mathbf{v}}_{\max}, \dot{\mathbf{w}}_{\max}, \ddot{\mathbf{u}}_{\max}, \ddot{\mathbf{v}}_{\max}, \ddot{\mathbf{w}}_{\max}$ *)*
  - *Step from 0 to* $\mathbf{t}_{\mathbf{horizon}}$ *at a predefined resolution*

A trajectory designed by this approach may be passed directly to the control space layer for tracking, or alternatively it may be used as the starting point for the gradient based optimisation. However, before any optimisation can occur the chosen trajectory must be converted into the polynomial form used by the gradient search algorithm. This can be done by employing a least squares curve fitting technique, where a sixth order Bezier polynomial is matched to the desired trajectory. Details regarding this curve-fit are provided in Appendix D, where it is shown that using a Bezier polynomial a closed form solution exists. As the Bezier basis function matrices are all calculated off-line, so can the curve-fit matrix ($B_{least\_squares}$), therefore reducing the on-line curve-fitting process to a single matrix multiplication for each of the three speed profiles, as shown below:

$$\begin{aligned} C^u &= B_{least\_squares} Y^u \\ C^v &= B_{least\_squares} Y^v \\ C^w &= B_{least\_squares} Y^w \end{aligned} \qquad (5\text{-}36)$$

where:

$$B_{least\_squares} = (B^T B)^{-1} B^T$$

B *= matrix of Bezier basis functions for the defined trajectory resolution*

C *= polynomial coefficients for least squares curve-fit to the desired axis speed profile*

Y *= desired axis speed profile for the preferred coarse grid trajectory*

Combining the gradient based optimisation with the coarse grid of feasible manoeuvres allows a two stage trajectory design approach, where a candidate trajectory from the coarse grid may be used as the starting point for further optimisation using the gradient search. The coarse grid provides the breadth of manoeuvre options, ensuring that the edges of the manoeuvre envelope can be reached when necessary. The gradient based optimisation then performs the final trajectory shaping to get a desired level of performance. The true benefit of the optimisation step is therefore to provide accurate performance without requiring excessively large quantities of discrete manoeuvres to be either stored or generated on line.

An animation illustrating this two stage design process is provided in the file '*animation1 - two stage trajectory design*' where the quadrotor vehicle is shown tracking a straight line global trajectory when a single static sphere obstacle is detected. The set of coarse grid trajectories are generated then compared using the defined cost function. The curve-fit process then converts the lowest cost trajectory into the appropriate formation for optimisation, which can then be seen to shape the trajectory back towards the global trajectory. Note that this entire animated trajectory design process can be implemented within the a receding horizon framework, therefore a new trajectory is subsequently calculated at the next design time (e.g. 0.1secs later).

5.7 Tracking a General 4D Trajectory

5.7.1 Target Trajectory

In order to evaluate the objective function the demanded positions and speeds across the design horizon are required. These values are likely to vary along the global trajectory, so for each candidate point on the receding horizon:

$$\left(x_i^{actual}, y_i^{actual}, z_i^{actual}, u_i^{actual}, v_i^{actual}, w_i^{actual}\right)$$

it is necessary to find the nearest demanded point on the global trajectory:

$$\left(x_i^{demand}, y_i^{demand}, z_i^{demand}, u_i^{demand}, v_i^{demand}, w_i^{demand}\right)$$

Ideally, calculation of the appropriate demanded position and speed data for each point on each candidate trajectory can be performed via a nearest point calculation on the global trajectory. The nearest point calculation for a continuous global trajectory is a reasonably simple optimisation problem. However, for the high number of required calculations this approach is likely to be too computationally expensive. For example, evaluation of a single candidate trajectory would require *n* (trajectory resolution) nearest-point calculations for each of the three speed profiles, giving 3*n* problems per trajectory. During optimisation the cost function may be evaluated up to 2,480 times (section 5.6.2) resulting in up to 372,000 nearest point calculations (using *n* = 50).

Although each nearest point problem may be seeded with the result from the previous, the computational effort would still be significant.

This issue can be avoided as follows. Rather than performing a nearest point calculation for each point in each candidate receding horizon trajectory, a single nearest point calculation[35] is performed only for the current vehicle position. Using this calculated nearest position on the global trajectory as the starting point, a suitable portion of the global trajectory can be isolated based on the length of the design horizon. This section of the global trajectory can then be curve-fitted via a least-squares approach to create a target trajectory for use in the cost function. The least squares curve-fit can be rapidly performed on-line via a single matrix multiplication for each speed axis (section 5.6.3 and appendix D), therefore calculating the target trajectory with little computational effort. The resolution of the target trajectory can be set to match that of the cost function removing the need for the repeated nearest point calculation, e.g. point *i* on the candidate trajectory can be compared with point *i* on the target trajectory. An additional advantage of this target trajectory approach is that it also allows the global trajectory to be specified at a lower resolution than the receding horizon trajectory, with the least squares curve-fit providing interpolation between the given points.

The procedure for tracking a general 4D trajectory then becomes:

- *Calculate the nearest point on the global trajectory to the current UAV position*
- *Isolate a section of the global trajectory from the nearest point up to $t_{horizon}$ from the nearest point.*
- *Perform the least-squares curve-fit on the isolated global trajectory section to get target u, v & w trajectories.*
- *Optimise the starting UAV trajectory to match the current target trajectory*

This process is illustrated in Figure 45 for the case of a UAV acquiring a general 4D trajectory, which shows the global trajectory, current UAV position, target trajectory and the initial and final receding horizon UAV trajectories.

5.7.2 Time Control

If time control is also required then an alteration to the above procedure is needed. If a time error has accumulated while tracking the trajectory then this requires a change in the speed specified in the global trajectory. However, the target trajectory specifies the global trajectory speed profile both explicitly, via the speed cost term, and implicitly, via the position cost term where each position has an associated time therefore defining

[35] Note that the issue of multiple nearest points can be avoided by only making limited sections of the global trajectory available to the tracking layer, therefore preventing a section from turning back on itself. Additionally, if a conflict still arises then the time profile may be used to resolve it.

speed also. Therefore simply adding a time cost term to the objective function results in a conflict between this and the position and speed profiles and does not provide the desired behaviour.

It is feasible to fit the target trajectory to the current desired time profile of the global trajectory rather than the nearest position, for example by generating the target trajectory from the current preferred time position rather than the nearest position. However, in the event of large time errors building up this approach may result in large deviations from the global trajectory as it effectively prioritises time over position, which may not be desirable for operation within complex environments.

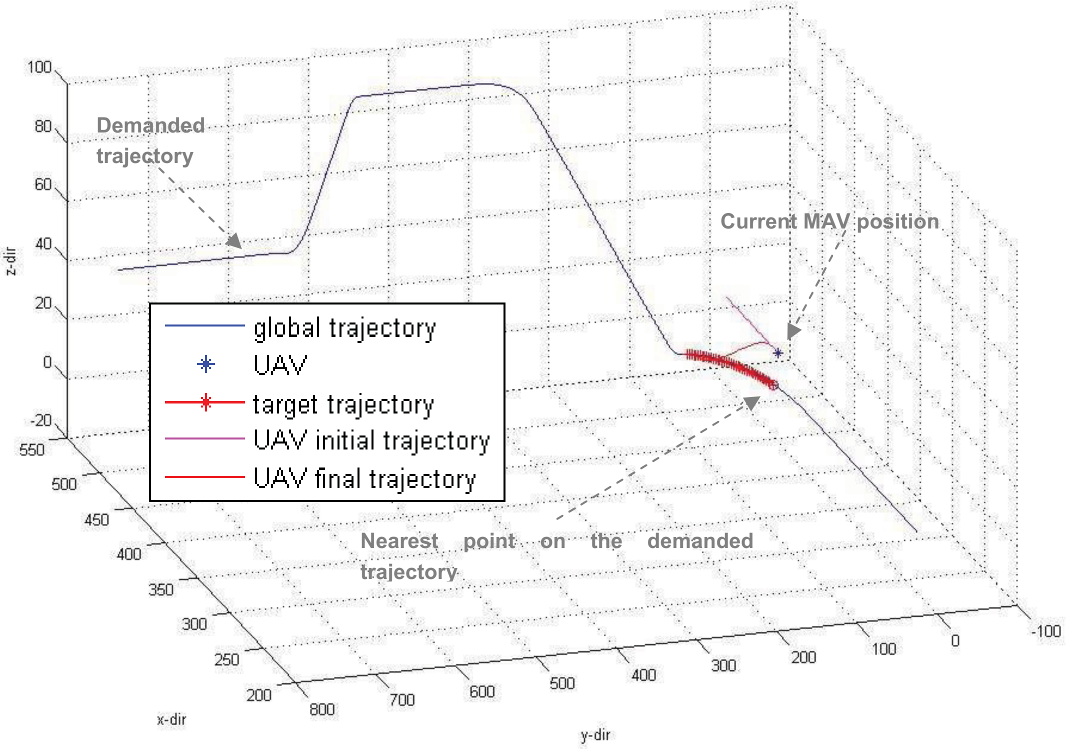


Figure 45 - General 4D trajectory tracking via calculation of a target trajectory

As the target trajectory defines the desired speed profile, time control can be provided without sacrificing the priority of position tracking by varying the length of the target trajectory. For example, a speed increase can be demanded by stretching the target trajectory over a longer section of the global trajectory. Conversely, if a speed reduction is required then the target trajectory can be compressed over a shorter section of the global trajectory. This process is illustrated in Figure 46, and as long as the cost function assumes that the target trajectory still lasts for the standard fixed time horizon, then minimising the cost function without an explicit time error term will provide the desired speed up / slow down.

The magnitude of the stretch / compression of the target trajectory was defined by a gain on the time error at the nearest position on the global trajectory to the vehicle's current position. The stretch / compression was then limited to be within a defined percentage of the fixed time-horizon.

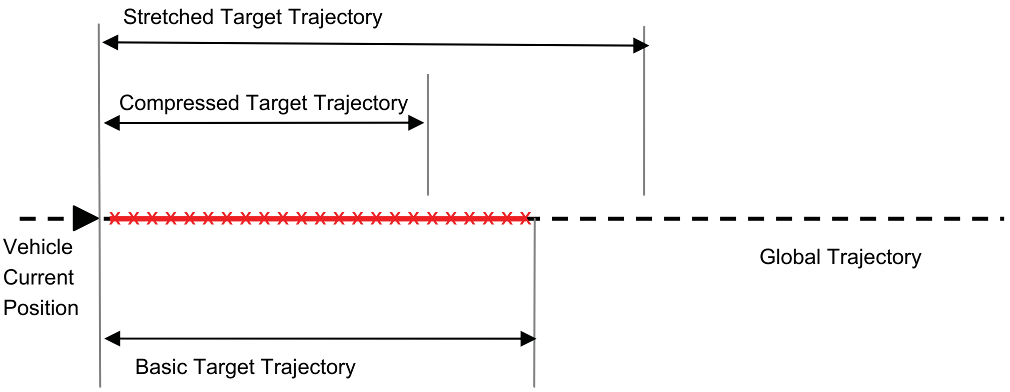


Figure 46 - Stretch / Compression of the Target Trajectory for Time Control

A problem that can be encountered with the time control algorithm described and demonstrated above, is that if the target trajectory is significantly stretched around a corner in the global trajectory, then the actual vehicle speed around this corner may exceed the trajectory design speed, therefore leading to a reduction in position tracking performance. In order to avoid this problem, it is desirable for the trajectory tracking algorithm not to attempt to reduce positive time errors (which require speed increases) while turning a corner. This can be implemented by checking the change in heading & flight path angle at the start and end of the target trajectory. If a significant difference is present (e.g. greater than 10deg) then the size of a target stretch can be reduced towards the default target time horizon.

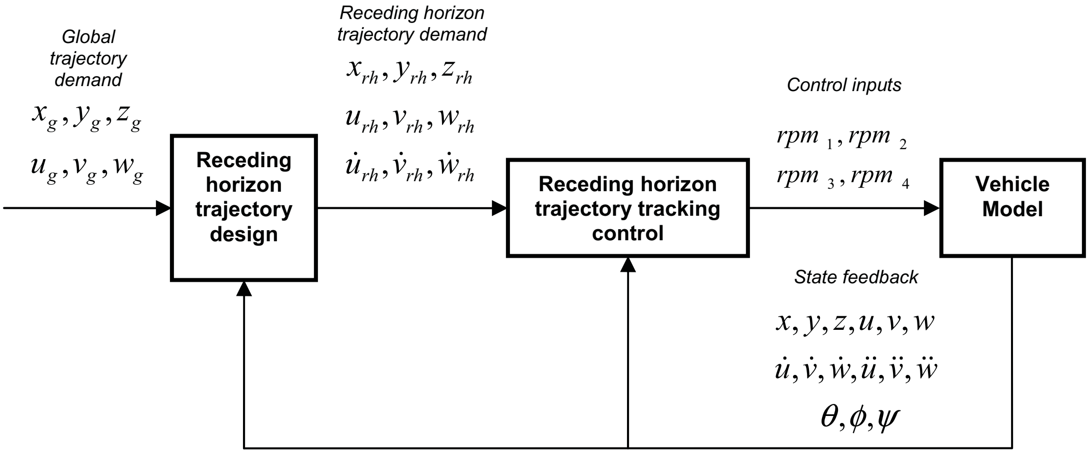
5.8 Summary

The primary components and key design decisions of the new local motion planning framework are as follows:

- *Output space design with vehicle performance limits described via off-line generated vehicle performance map.*
- *Receding horizon trajectories described using three 6th order Bezier polynomials, one for each of the u, v & w axis.*
- *Receding horizon length fixed at* $t_{horizon} = 10s$.
- *Initial vehicle state boundary conditions imposed are* $u_0, \dot{u}_0, \ddot{u}_0, v_o, \dot{v}_0, \ddot{v}_0, w_o, \dot{w}_0, \ddot{w}_0$. *No boundary conditions imposed at end of receding horizon trajectory.*
- *Boundary conditions enforced analytically, reducing number of design variables from 21 to 12.*
- *Obstacles described using standard geometric shapes (cuboids, spheres and cylinders) to provide efficient description and to simplify proximity and collision calculations.*
- *Optimisation via steepest-descent based gradient approach.*

- *Obstacle and vehicle performance constraints enforced via penalty function approach.*
- *Coarse grid of feasible trajectories allows escape from local minimums in the cost function. Optimisation of coarse grid trajectories possible after least-squares curve-fit to convert into suitable form.*
- *Predictions regarding future obstacle positions based on simple state propagation, with uncertainty handling by increasing desired clearance distance across the design horizon.*
- *Obstacle proximity function enhanced to also scale with relative vehicle-obstacle speed.*
- *Cost function position and speed errors calculated via target trajectory calculated via a least squares curve-fit to the global trajectory.*
- *Time control based on stretching and compressing the target trajectory to alter the demanded speed.*

A block diagram illustrating the implemented motion planning & control framework is provided in Figure 47, which also shows the inputs / outputs for each block.



Global trajectory demand
x_g, y_g, z_g
u_g, v_g, w_g
Receding horizon trajectory design
Receding horizon trajectory demand
x_{rh}, y_{rh}, z_{rh}
u_{rh}, v_{rh}, w_{rh}
$\dot{u}_{rh}, \dot{v}_{rh}, \dot{w}_{rh}$
Receding horizon trajectory tracking control
Control inputs
rpm_1, rpm_2
rpm_3, rpm_4
Vehicle Model
State feedback
x, y, z, u, v, w
$\dot{u}, \dot{v}, \dot{w}, \ddot{u}, \ddot{v}, \ddot{w}$
θ, ϕ, ψ

Figure 47 - Local Motion Planning & Control Framework

The overall framework is divided into distinct off-line and on-line components, the primary calculations for each shown in Figure 48 and Figure 49. The ability to calculate the various matrices off-line provides a significant computational advantage during the on-line stage, allowing the full position, speed and acceleration profiles to be calculated rapidly for each candidate design. This, combined with the penalty function handling of obstacle and performance constraints allows rapid trajectory optimisation.

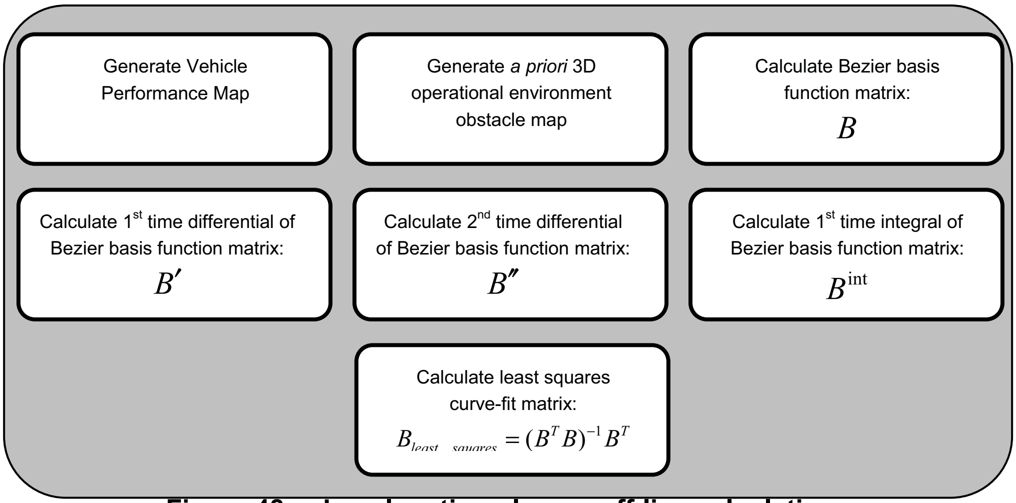

Generate Vehicle Performance Map
Generate *a priori* 3D operational environment obstacle map
Calculate Bezier basis function matrix: B
Calculate 1st time differential of Bezier basis function matrix: B'
Calculate 2nd time differential of Bezier basis function matrix: B''
Calculate 1st time integral of Bezier basis function matrix: B^{int}
Calculate least squares curve-fit matrix: $B_{least\quad squares} = (B^T B)^{-1} B^T$

Figure 48 - Local motion planner off-line calculations

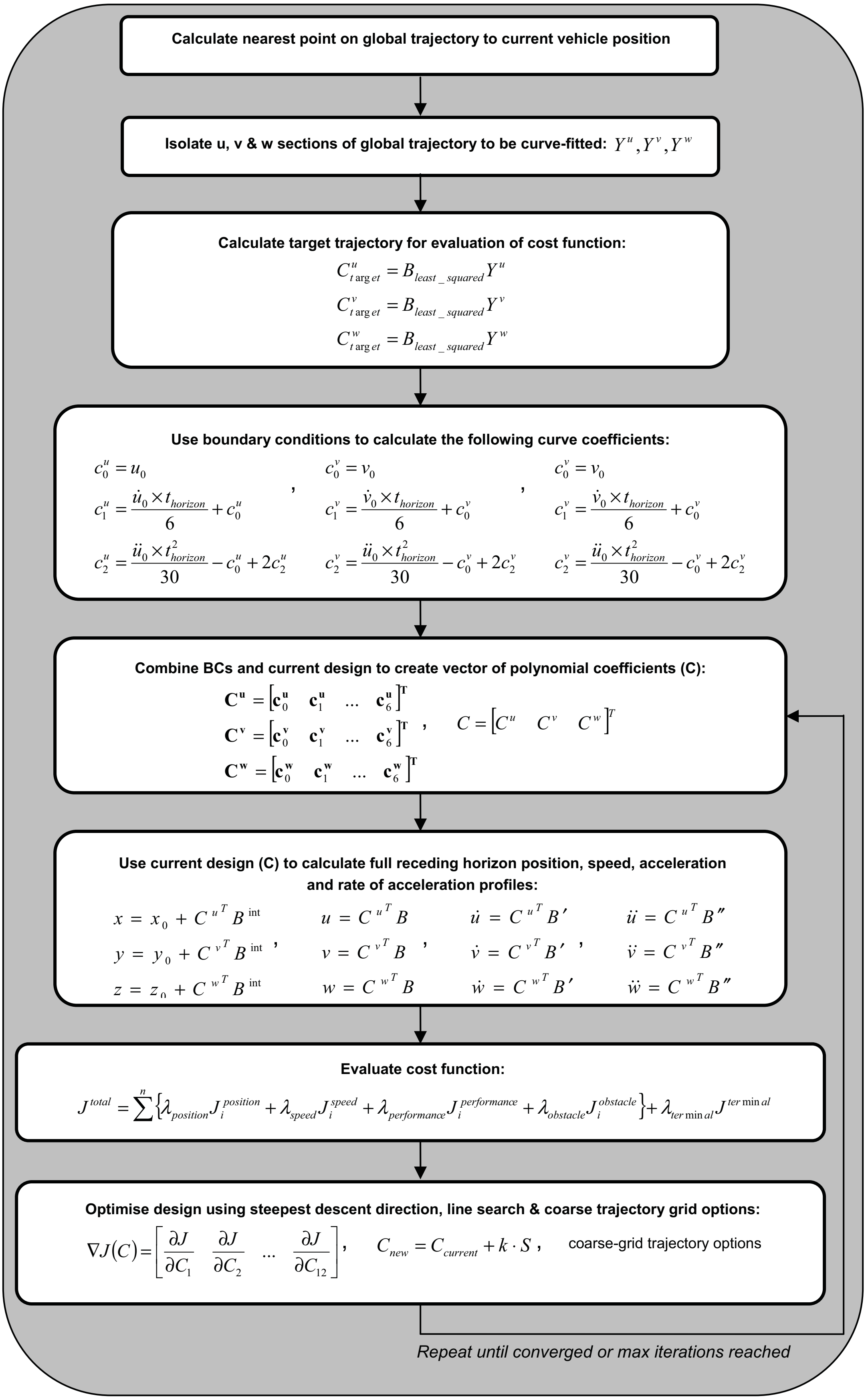

Calculate nearest point on global trajectory to current vehicle position
Isolate u, v & w sections of global trajectory to be curve-fitted: Y^u, Y^v, Y^w
Calculate target trajectory for evaluation of cost function:
$C^u_{t\arg et} = B_{least\_squared} Y^u$
$C^v_{t\arg et} = B_{least\_squared} Y^v$
$C^w_{t\arg et} = B_{least\_squared} Y^w$
Use boundary conditions to calculate the following curve coefficients:
$c^u_0 = u_0$, $c^u_1 = \frac{\dot{u}_0 \times t_{horizon}}{6} + c^u_0$, $c^u_2 = \frac{\ddot{u}_0 \times t^2_{horizon}}{30} - c^u_0 + 2c^u_2$,
$c^v_0 = v_0$, $c^v_1 = \frac{\dot{v}_0 \times t_{horizon}}{6} + c^v_0$, $c^v_2 = \frac{\ddot{u}_0 \times t^2_{horizon}}{30} - c^v_0 + 2c^v_2$,
$c^v_0 = v_0$, $c^v_1 = \frac{\dot{v}_0 \times t_{horizon}}{6} + c^v_0$, $c^v_2 = \frac{\ddot{u}_0 \times t^2_{horizon}}{30} - c^v_0 + 2c^v_2$
Combine BCs and current design to create vector of polynomial coefficients (C):
$\mathbf{C^u} = [\mathbf{c^u_0} \quad \mathbf{c^u_1} \quad \dots \quad \mathbf{c^u_6}]^\mathbf{T}$
$\mathbf{C^v} = [\mathbf{c^v_0} \quad \mathbf{c^v_1} \quad \dots \quad \mathbf{c^v_6}]^\mathbf{T}$,
$\mathbf{C^w} = [\mathbf{c^w_0} \quad \mathbf{c^w_1} \quad \dots \quad \mathbf{c^w_6}]^\mathbf{T}$
$C = [C^u \quad C^v \quad C^w]^T$
Use current design (C) to calculate full receding horizon position, speed, acceleration and rate of acceleration profiles:
$x = x_0 + C^{u^T} B^{\text{int}}$, $y = y_0 + C^{v^T} B^{\text{int}}$, $z = z_0 + C^{w^T} B^{\text{int}}$,
$u = C^{u^T} B$, $v = C^{v^T} B$, $w = C^{w^T} B$,
$\dot{u} = C^{u^T} B'$, $\dot{v} = C^{v^T} B'$, $\dot{w} = C^{w^T} B'$,
$\ddot{u} = C^{u^T} B''$, $\ddot{v} = C^{v^T} B''$, $\ddot{w} = C^{w^T} B''$
Evaluate cost function:
$J^{total} = \sum^n \{\lambda_{position} J^{position}_i + \lambda_{speed} J^{speed}_i + \lambda_{performance} J^{performance}_i + \lambda_{obstacle} J^{obstacle}_i\} + \lambda_{ter\min al} J^{ter\min al}$
Optimise design using steepest descent direction, line search & coarse trajectory grid options:
$\nabla J(C) = \left[\frac{\partial J}{\partial C_1} \quad \frac{\partial J}{\partial C_2} \quad \dots \quad \frac{\partial J}{\partial C_{12}}\right]$, $C_{new} = C_{current} + k \cdot S$, coarse-grid trajectory options
Repeat until converged or max iterations reached

Figure 49 - Local motion planner primary on-line calculations

6 Simulation Results - *Single Vehicle Scenarios*

This chapter provides simulation results demonstrating the LMP framework presented in chapter 5 applied to the quadrotor UAV. Unless otherwise stated, all presented results use the following default settings:

- *Design horizon:* $t_{horizon} = 10s$
- *Local motion trajectory resolution: n = 50*
- *Design rate of either 2.5Hz or 5Hz*[36]
- *Cost function scaling parameters:* $\lambda_{position} = 1000$, $\lambda_{speed} = 5000$, $\lambda_{performance} = 1$, $\lambda_{obstacle} = 1$, $\lambda_{ter\min al} = 500$

All simulation results shown in this section assume that the output space trajectories are perfectly tracked by the quadrotor vehicle. This is implemented by setting the vehicle state at the start of each iteration to the state either 0.2s or 0.4s (for 5Hz and 2.5Hz design rates) from the start of the previous receding horizon trajectory. It is also assumed that perfect obstacle data is available, with the sensor range limited to 100m. The global trajectories to be tracked are specified ‘continuously’ via a series of waypoints spaced at 50Hz, and are constrained to be within quadrotor speed and acceleration performance limits. All development and simulation was performed in the Matlab environment, using scripted m-code.

6.1 Obstacle Free Trajectory Tracking

Before addressing the presence of static and dynamic obstacles it is necessary to demonstrate that the new LMP framework is capable of producing suitable trajectory tracking performance. As well as minimising position and time errors, it is also critical that the framework is able to handle regular trajectory acquire / re-acquire manoeuvres due to gust / turbulence disturbances, unexpected obstacles and global trajectory re-designs.

6.1.1 Accuracy of the target trajectory

The cost function minimises errors between candidate receding horizon trajectories and a target trajectory that was curve-fitted to the global trajectory. Before considering trajectory tracking errors, it is necessary to verify the accuracy of the least-squares curve-fit process. A series of tests were performed, progressing along several global trajectories performing the curve-fit process at each step then comparing the output to the actual data. Typical position and speed errors that result from the curve-fit process

[36] Although a faster design rate is preferable for real application, these values are suitable for development and demonstration.

are shown in Figure 50 where it can be seen that position errors peak at approx. 0.08m and speed errors peak at less that 0.25m/s.

This level of accuracy results from the default global trajectory performance and 6th order Bezier polynomials. If required it is possible to improve significantly on this performance (appendix E), although this was not considered necessary for the simulation tests shown within this thesis.

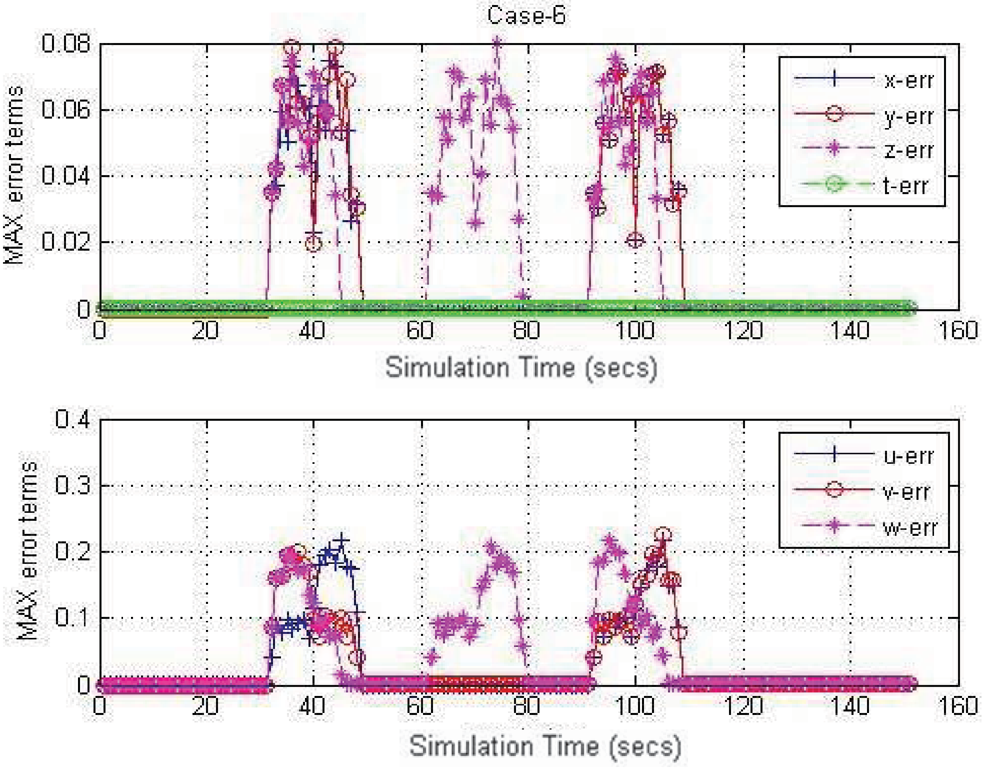


Figure 50 - Typical position and speed errors due in the target trajectory

6.1.2 3D trajectory tracking

An example of typical performance found for 3D trajectory tracking is shown in Figure 51 to Figure 55. It can be seen that the overall performance is good, with position errors peaking at approximately 0.35m around trajectory corners. It can also be seen that vehicle performance limits have not been exceeded, although given that the global trajectory is performance limited performance constraints should not have been significantly challenged.

If required, position errors may be reduced by several methods, including:

- *Optimisation effort* - By either increasing the maximum number of iterations or reducing the convergence limit (cost change per iteration).

- *Order of receding horizon polynomials* - Increasing the order of the polynomials (e.g. to 7th or 8th order) allows the receding horizon trajectories greater flexibility to match the demanded trajectory. This increased flexibility comes at the expense of a higher dimensional optimisation problem, although the gradient search mitigates the impact of this.

- *Receding horizon resolution* - Increasing the resolution of the trajectories allows the cost function to better discern small errors.
- *Performance limits used in global trajectory* - Less aggressive corners can be tracked with smaller errors. It is unlikely that the design of the global trajectory needs to use maximum vehicle performance limits (this example uses approx. 50% acceleration).
- *Target trajectory curve-fit calculation* - Reducing errors here directly impacts on the total position error.

However, it should be noted that even while operating within complex environments, the architectural approach of making the trajectory tracking layer situation aware means that deviations from the global trajectory are performed with an awareness of the surrounding obstacle space, therefore making precise trajectory tracking less critical than with a 'blind' tracking layer.

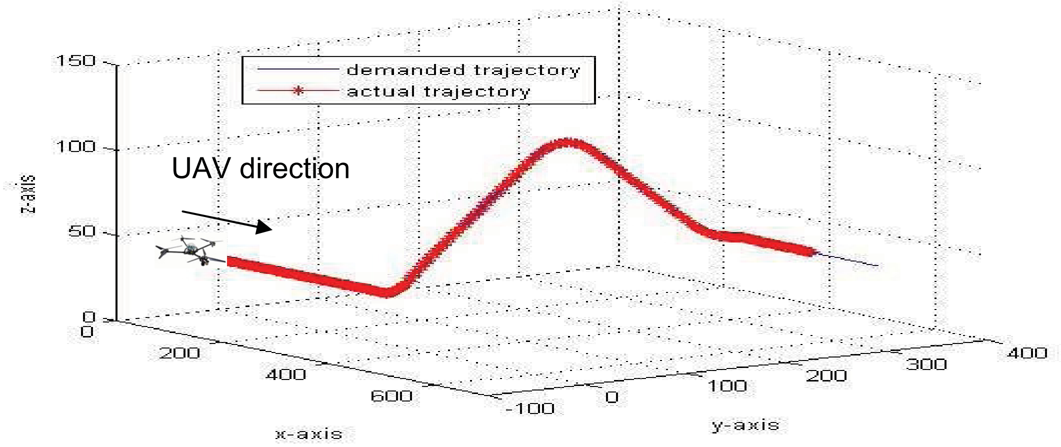


Figure 51 - Obstacle free trajectory tracking performance (3D view of trajectory)

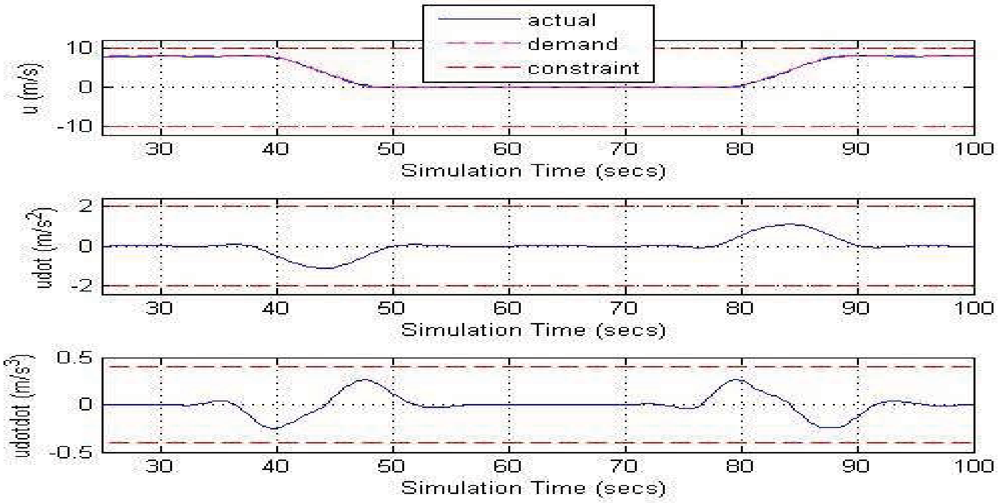


Figure 52 - Obstacle free trajectory tracking performance (u-axis time history)

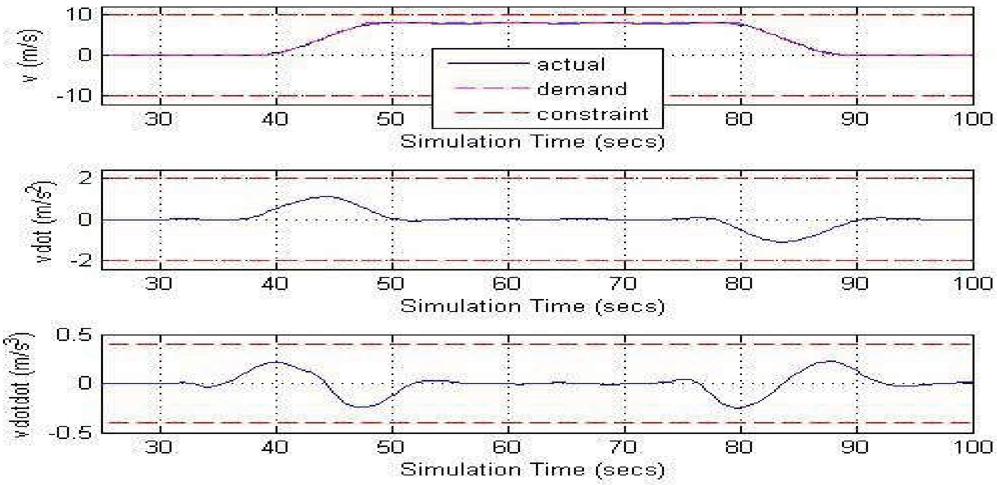


Figure 53 - Obstacle free trajectory tracking performance (v-axis time history)

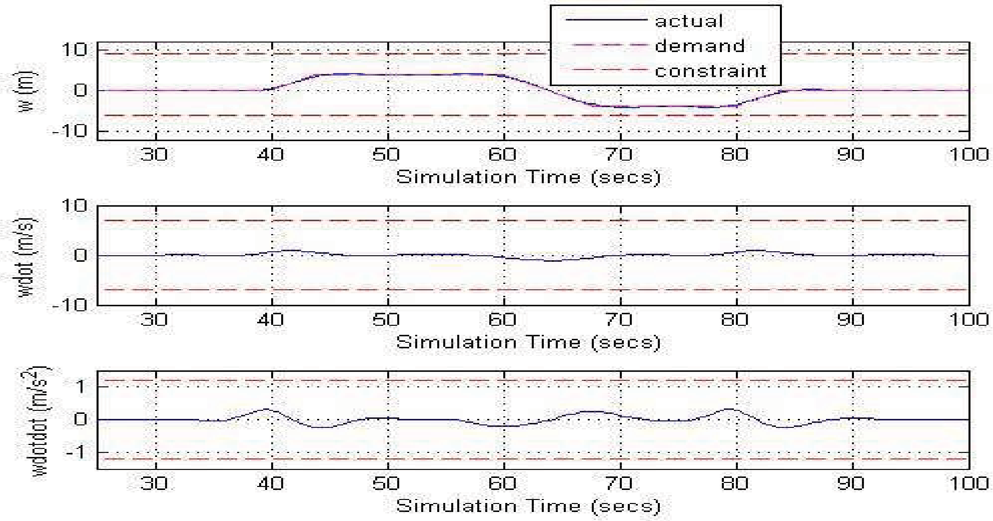


Figure 54 - Obstacle free trajectory tracking performance (w-axis time history)

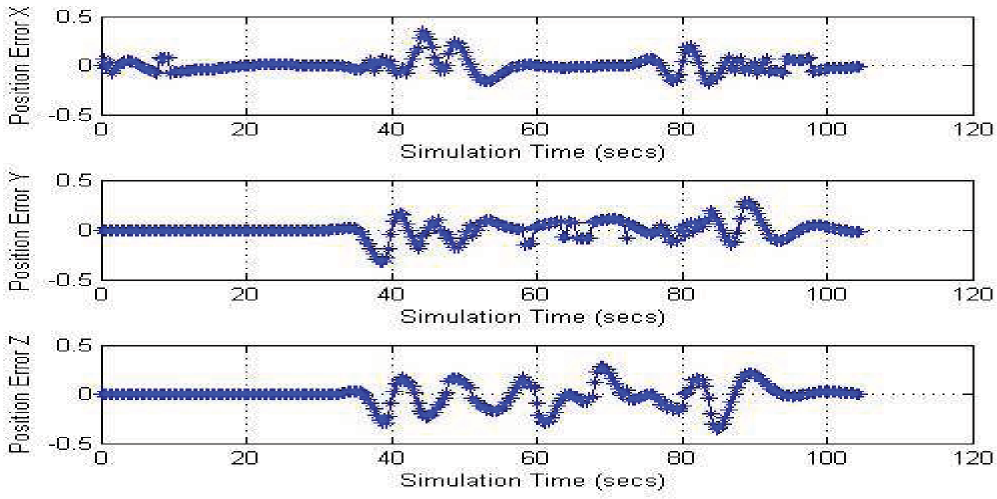


Figure 55 - Obstacle free trajectory tracking performance (position errors)

6.1.3 4D trajectory tracking

An example of 4D control is provided in Figure 56. In this example the initial vehicle state has significant position and speed errors, leading to the build up of a time profile error (upper right plot). It can be seen that the trajectory is smoothly acquired, with the time error gradually driven to zero. Importantly the maximum performance of the vehicle is used to reduce the position and time errors, without violating the performance constraints. It can also be seen that the reduction in time error flattens out during the first corner, in order to prevent the additional speed from causing additional position errors.

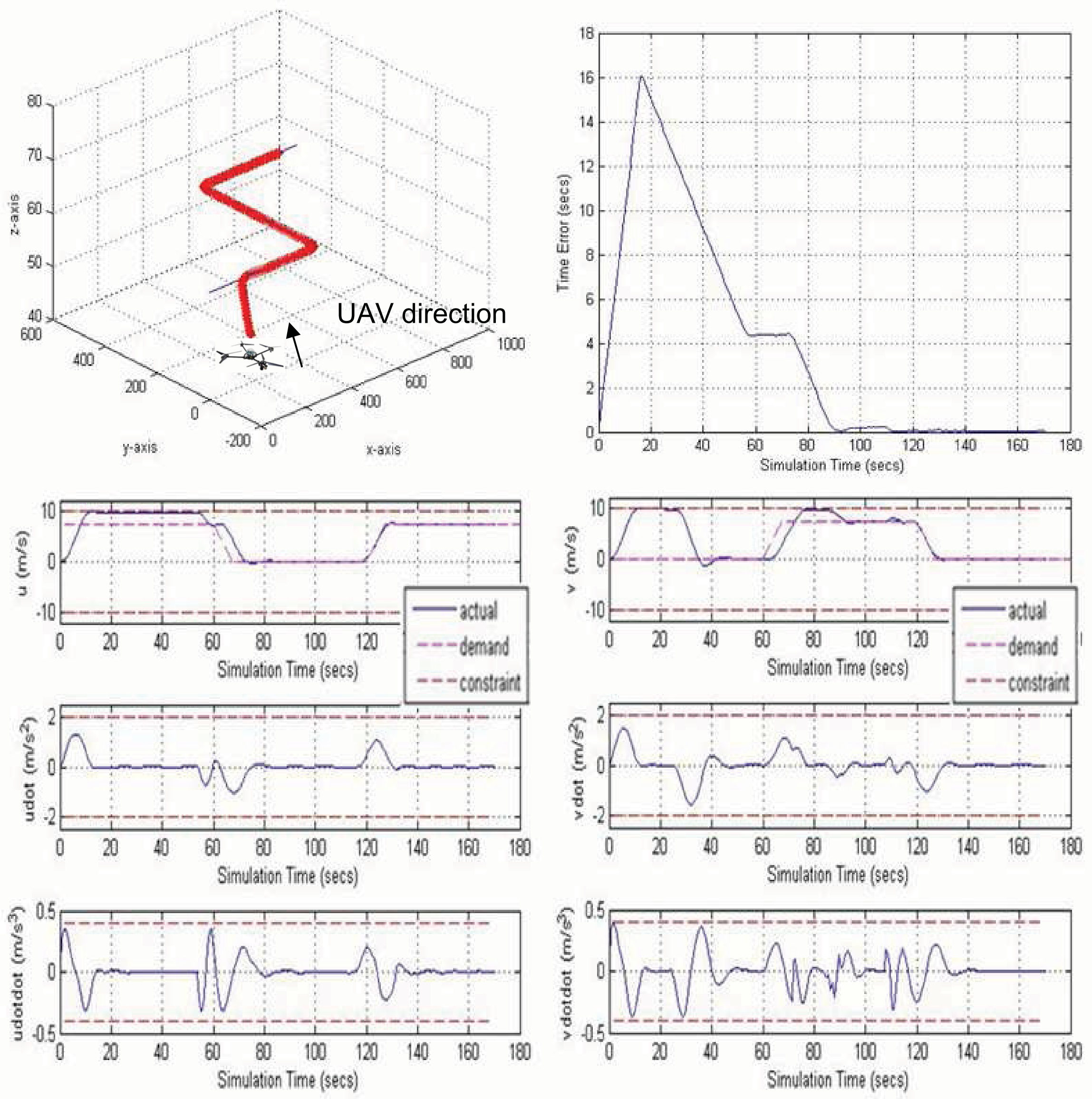


Figure 56 - 4D tracking example

6.1.4 Trajectory acquire performance

Due to their susceptibility to gusts and turbulence it is particularly important for small / micro UAVs to be able to smoothly acquire / re-acquire a given trajectory. Additionally, the situation aware trajectory tracking problem is based on an assumption that certain obstacles can be handled at the local level, therefore requiring departure and re-acquire manoeuvres when unexpected obstacles are detected *en route*. An example of the ability of the LMP framework to provide good trajectory acquire performance is provided in Figure 57 to Figure 61. In this scenario position and speed errors are present in all three axes at the start of the simulation, and it can be seen that all errors are rapidly reduced to zero. This behaviour is typical of that found, with maximum performance limits used and little overshoot. The use of maximum rate of acceleration performance limits during the manoeuvre can clearly be seen in Figure 59.

The cost function term that provides this desirable performance is J_{speed} (equation 5-28). Without this term the acquire behaviour is typically oscillatory, and heavily dependent on the terminal cost as shown in Figure 62.

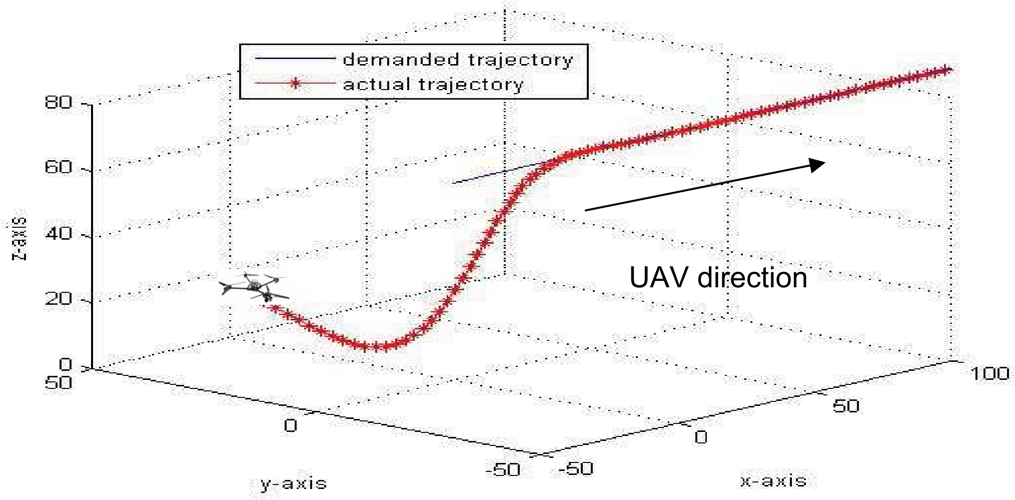


Figure 57 - Trajectory acquire example (3D overview)

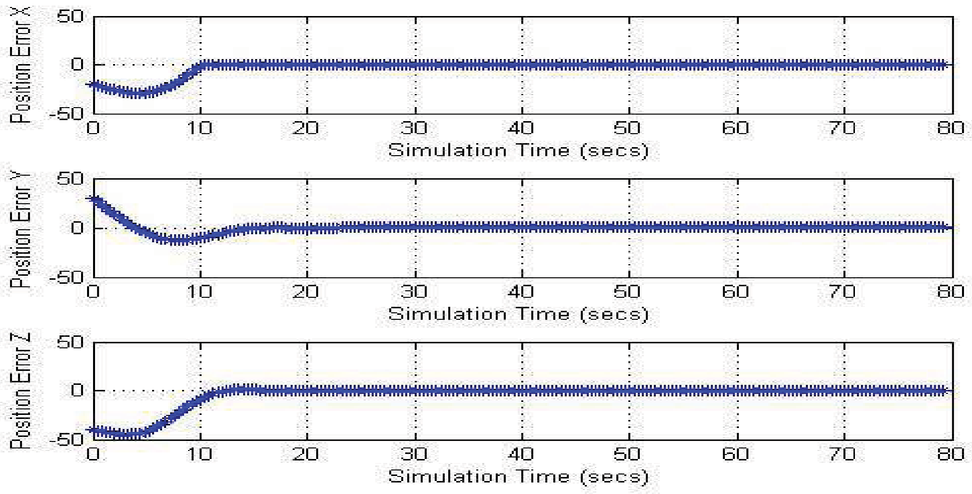


Figure 58 - Trajectory acquire example (position errors)

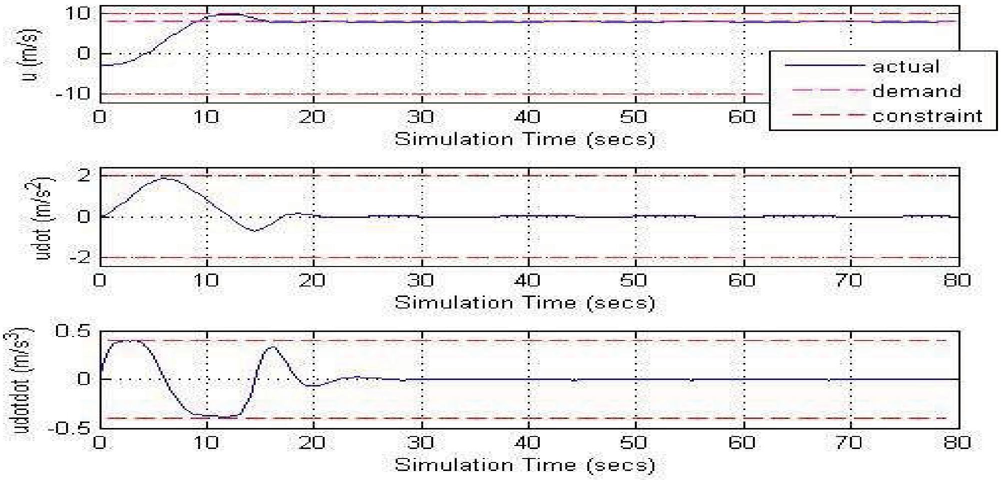


Figure 59 - Trajectory acquire example (u-axis time history)

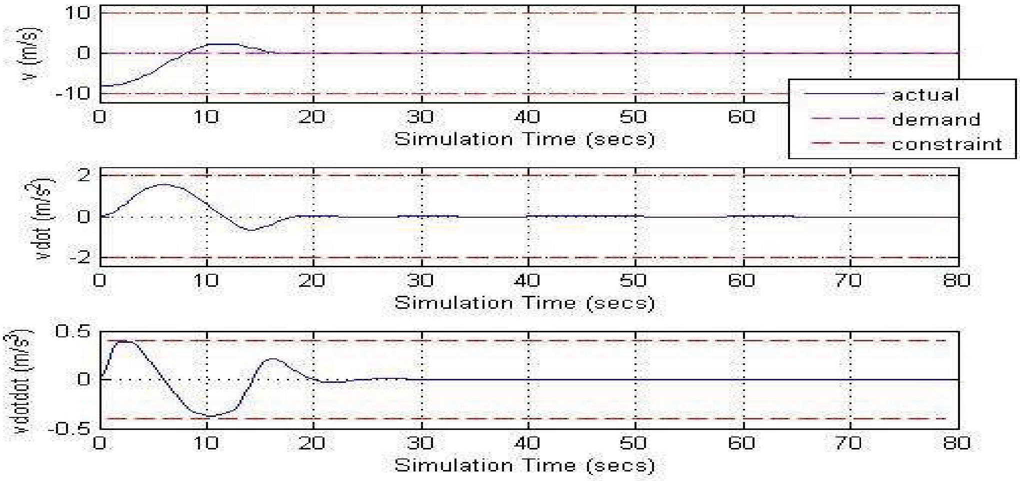


Figure 60 - Trajectory acquire example (v-axis time history)

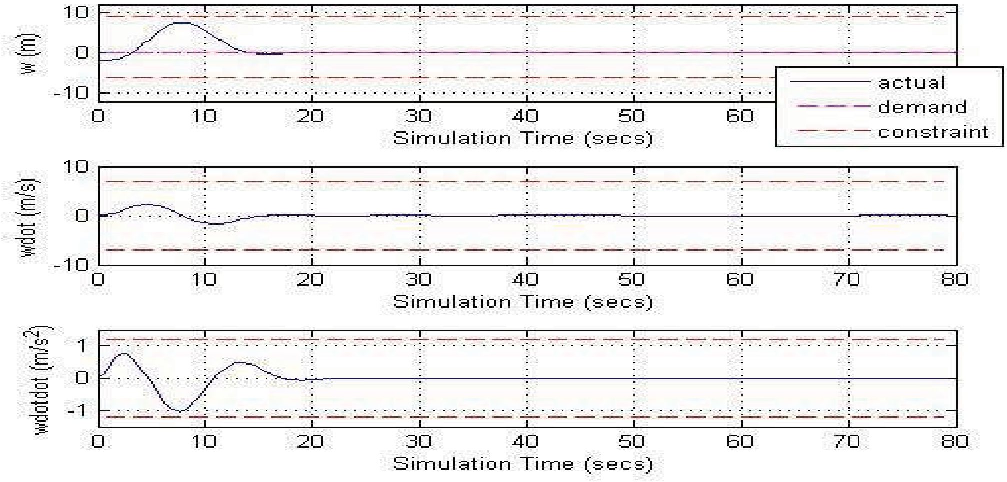


Figure 61 - Trajectory acquire example (w-axis time history)

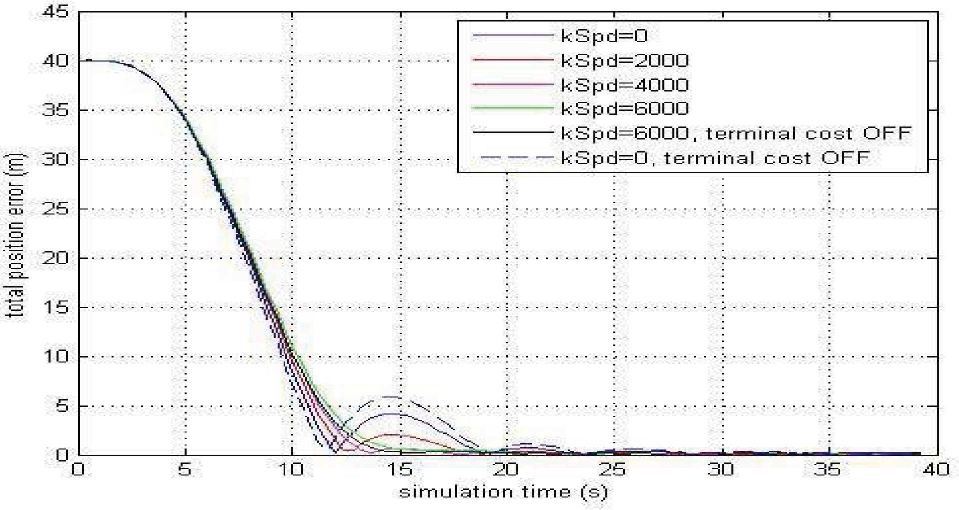


Figure 62 - Importance of speed cost term in providing acquire performance

6.2 Static Obstacle Scenarios

For static obstacles it is necessary to differentiate between obstacles that directly intersect the global trajectory and those that do not. The reason for this is that the optimisation approach requires a cost function gradient to direct the search prior to collision occurring. For example, if a cuboid obstacle is both perpendicular to and intersects the global trajectory the proximity based penalty function will not provide a gradient, therefore the vehicle will come to a stop in front of the obstacle. Within this thesis this issue is avoided by modelling obstacles that intersect the global trajectory as either spheres or cylinders, which always provide a proximity based gradient to direct the search. If a cuboid model was preferred then gradient would need to be provided by a different method, e.g. adding an artificial surface on the intersecting face (although this was not performed in this work).

6.2.1 Static obstacles that do not intersect the global trajectory

An example of the behaviour of the LMP framework in the presence of static obstacles that are close to, but do not intersect, the demanded trajectory is shown in Figure 63. It can be seen that the impact of the obstacle proximity field is to repel the vehicle from the obstacles, while it continues to follow the trajectory. The final position errors result from a balance of costs between the pull towards the trajectory and the push away from the obstacle, and can clearly be set to a desired value by designing the weights in the objective function. The resulting behaviour is similar to what would be expected from a human pilot, who would naturally maintain a safe clearance from the obstacles. It should also be emphasized that after each obstacle is passed the demanded trajectory is rejoined in a smooth and efficient manner.

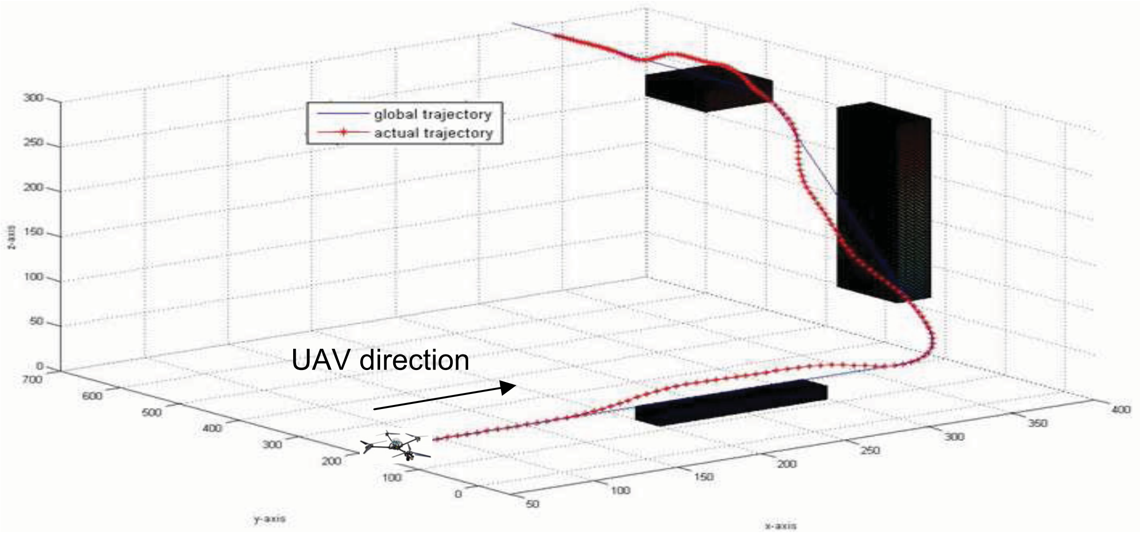


Figure 63 - Static obstacle example-1

6.2.2 *Static obstacles that intersect the global trajectory*

An example of the behaviour of the LMP framework when static obstacles that directly intersect the demanded trajectory are encountered is shown in Figure 64. It can be seen that the performance is again good, with the vehicle continuing to track the demanded trajectory, while making reasonable avoidance manoeuvres around each obstacle. The choice of avoidance direction is dependent only on the gradient of the cost function, therefore the vehicle will manoeuvre in a direction that reduces the distance from the obstacle while also attempting to follow the demanded trajectory. Time histories of the vehicle state in all three axes are also shown for this scenario in Figure 65, Figure 66 & Figure 67. It can be seen from these figures that the vehicle performance limits were successfully enforced by the penalty function approach. It can also be seen that the manoeuvres approach the performance limits of the vehicle, suggesting that the defined framework is capable of extracting the maximum performance from the vehicle when required.

The speed of the required reaction of the vehicle is defined by the decay rate of the obstacle proximity function (Section 5.4.3), e.g. a slow decay rate provides more advance warning that an obstacle was being approached. Conversely, if the decay rate is very rapid (confined to close proximity to the obstacle) then the reaction time is reduced, requiring faster vehicle acceleration to provide the same safe clearance distance. Note that this occurs within the defined design horizon ($T_{horizon} = 10s$), although this should be set to approximately match sensor capabilities.

Finally, as with the previous example the trajectory re-acquire performance is seen to be good. In both of the shown static obstacle scenarios the optimisation was based purely on the gradient search, therefore it is clear that global optimality, while desirable, is not necessary for acceptable system performance.

An animation of static obstacle scenario-2 is provided in the file 'animation-2 static obstacles'. The obstacles in this example are all detected *en route*, and it is clear that re-designing the global trajectory is not necessary. Additionally, the smooth nature of the successive re-acquire manoeuvres can also be seen.

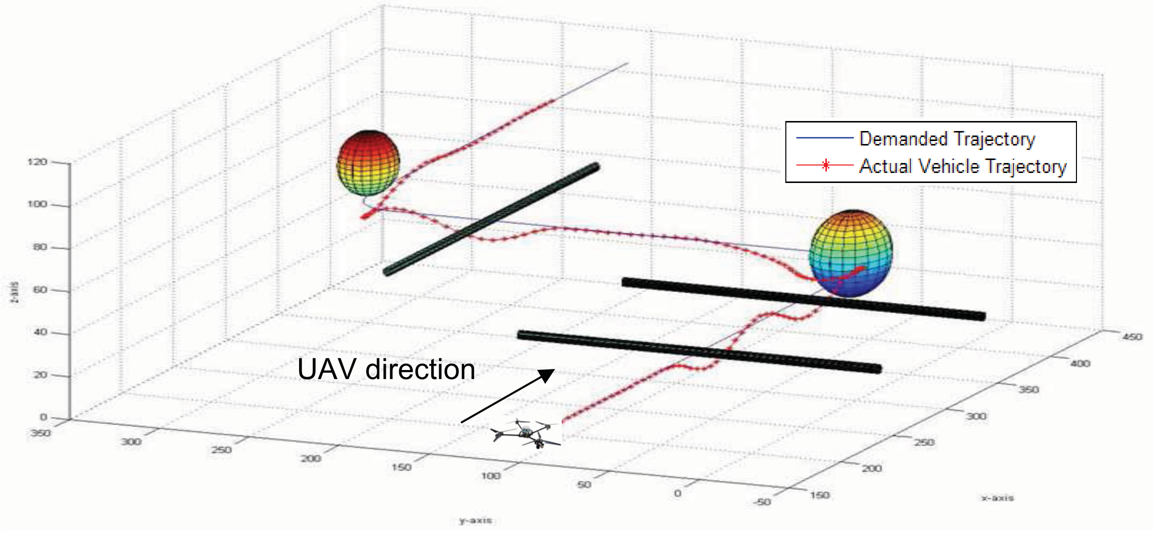


Figure 64 - Static obstacles example-2

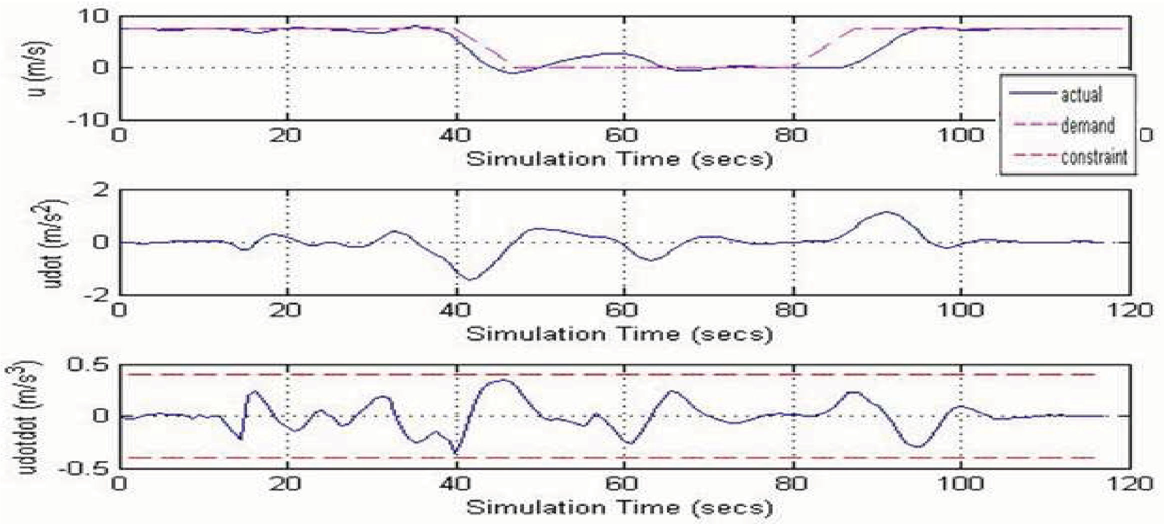


Figure 65 - Static obstacles example-2 (u-axis time history)

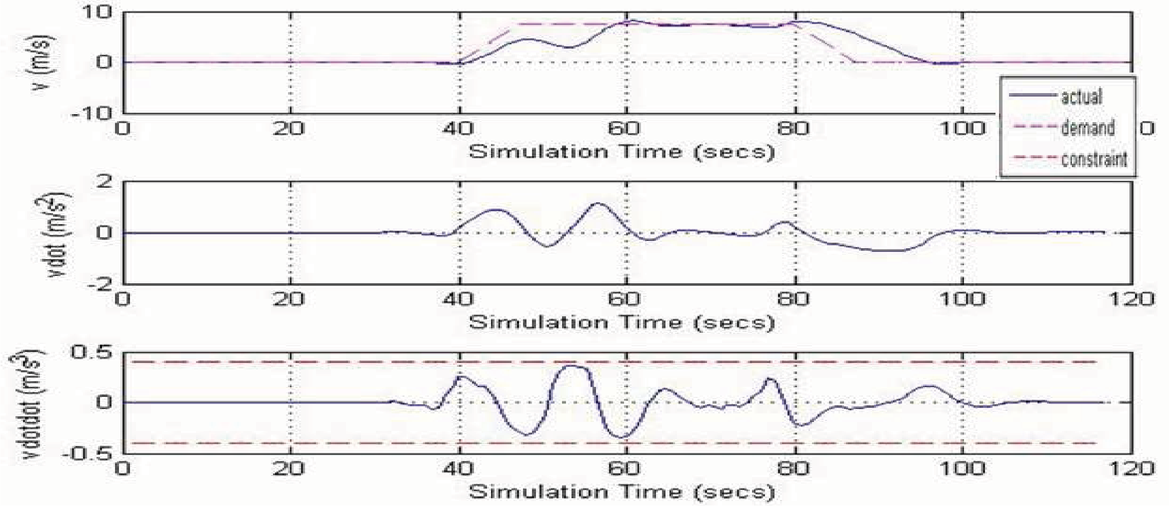


Figure 66 - Static obstacles example-2 (v-axis time history)

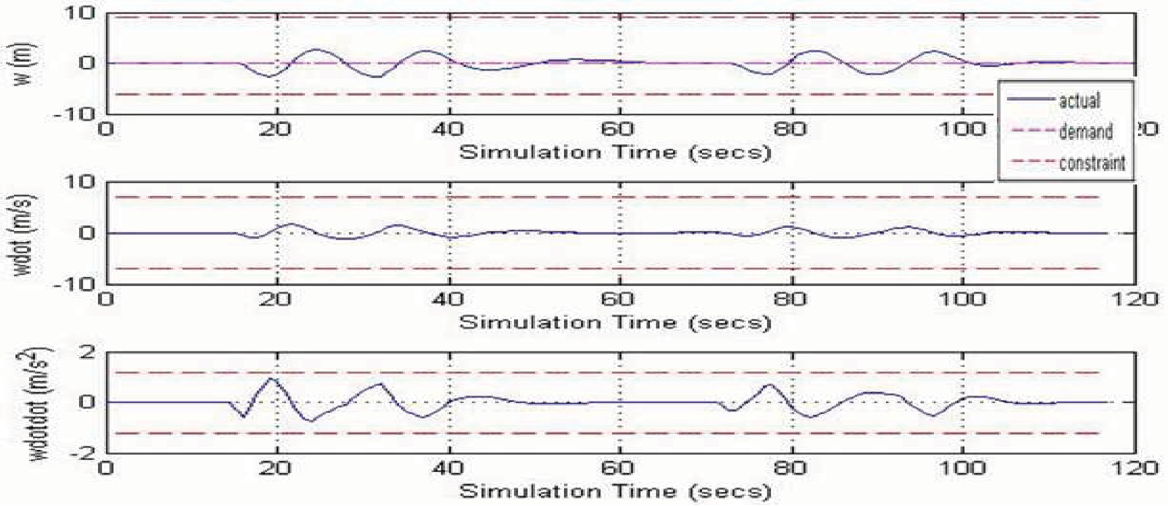


Figure 67 - Static obstacles example-2 (w-axis time history)

6.2.3 Static obstacles and disturbances

In order to test the robustness of the algorithms to disturbances several scenarios were tested with random disturbances added to the vehicle state vector every few seconds. The impact of each of these disturbances is to add random position, speed and acceleration errors, therefore requiring regular trajectory re-acquire manoeuvres. A typical example of the performance of the algorithms is shown in Figure 68, where it can be seen that the demanded trajectory is followed in a stable and predictable manner, with each of the disturbances followed by a smooth rejoin manoeuvre hence displaying the inherent stability of the approach.

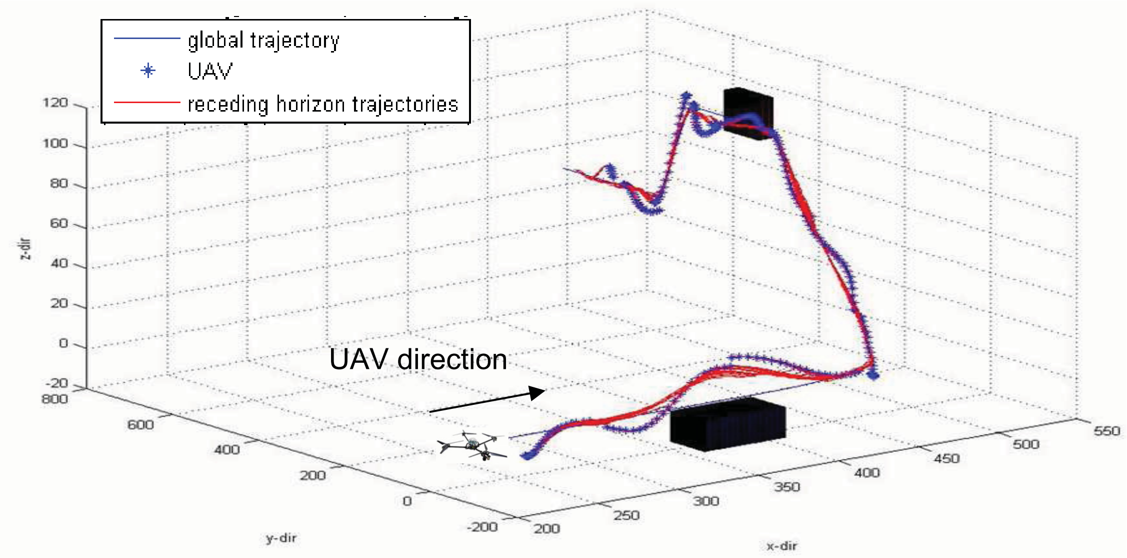


Figure 68 - Static obstacles example-3

6.3 Dynamic Obstacle Scenarios

For dynamic obstacle scenarios the motion of the obstacles is indicated in the 3D plots by displaying their positions every few seconds. Additionally, to aid clarity animations are provided for certain scenarios.

6.3.1 *Single dynamic obstacle scenarios*

An example of the behaviour of the LMP framework when a single dynamic obstacle approaching on a side on collision course is encountered is provided in Figure 69 to Figure 72. In this scenario the vehicle is travelling at 8m/s while the obstacle, modelled as a sphere of radius 2m, is travelling at 4m/s. It can be seen that the obstacle is successfully avoided primarily by a v-axis manoeuvre, although a u-axis slow-down is also used to increase the clearance distance. It can also be seen that vehicle performance limits are successfully enforced. The choice of avoidance direction is driven purely by the gradient of the obstacle cost function, therefore it is not known in advance. It is worth noting again that the subsequent trajectory acquire manoeuvre is smooth. As with static obstacles, the provided clearance distance may be tuned via the weighting terms in the cost function.

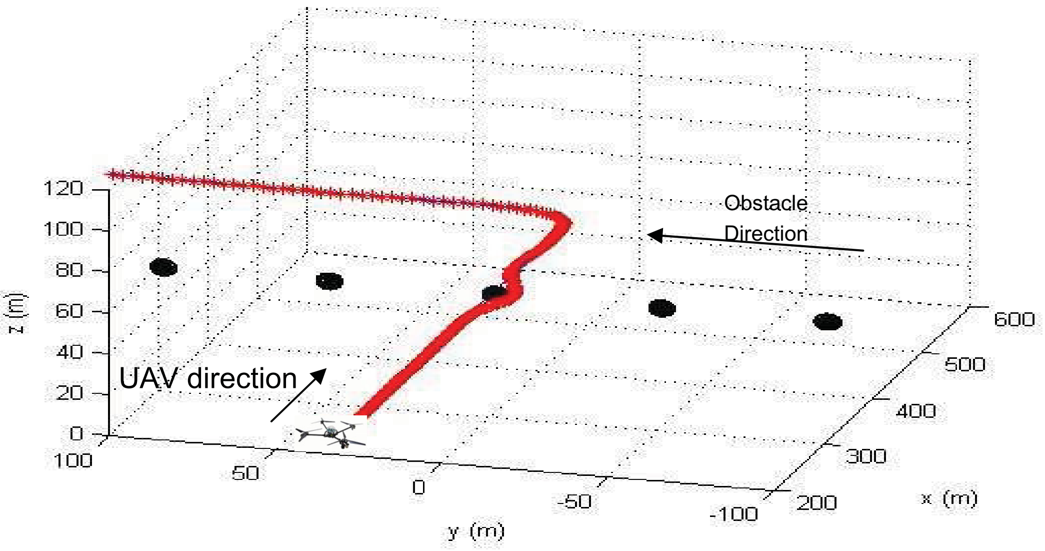


Figure 69 - Single dynamic obstacle example-1

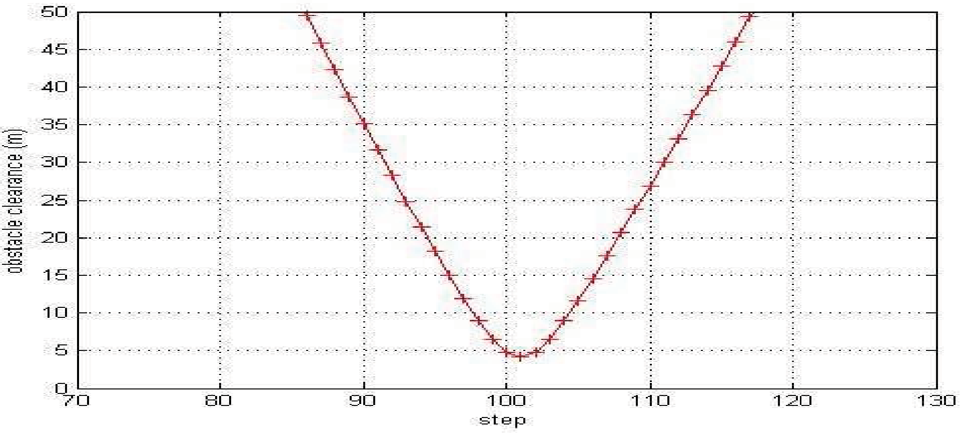


Figure 70 - Single dynamic obstacle example-1 (obstacle clearance)

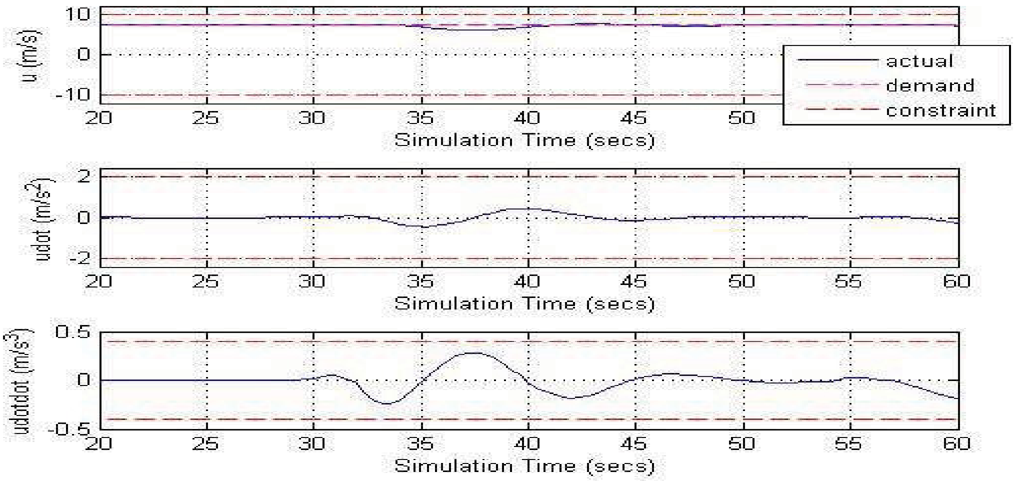


Figure 71 - Single dynamic obstacle example-1 (u-axis time history)

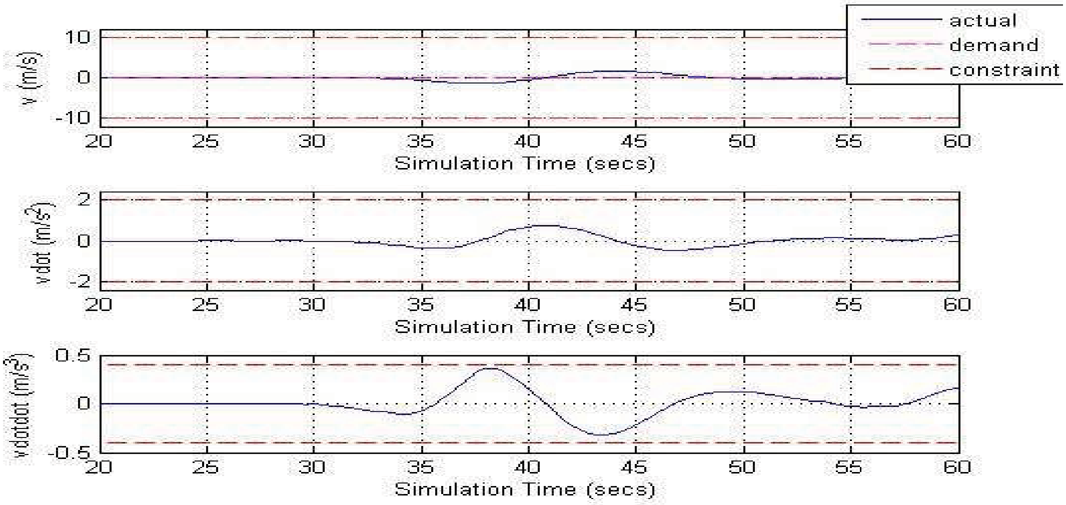


Figure 72 - Single dynamic obstacle example-1 (v-axis time history)

A second single dynamic obstacle is shown in Figure 73 to Figure 76, this time with the obstacle approaching on a collision course from below and behind at 5.6m/s. It can be seen that again the obstacle is successfully avoided, this time with the manoeuvre contained within the u and w axes. The primary avoidance direction is in the w-axis, although it can be seen that in order to provide the required obstacle clearance the vehicle accelerates in the u-axis also. This acceleration takes the vehicle up to the maximum speed in the u-axis, but as before all performance limits are successfully enforced. As before, the chosen avoidance direction depends only on the proximity gradient of the obstacle penalty function.

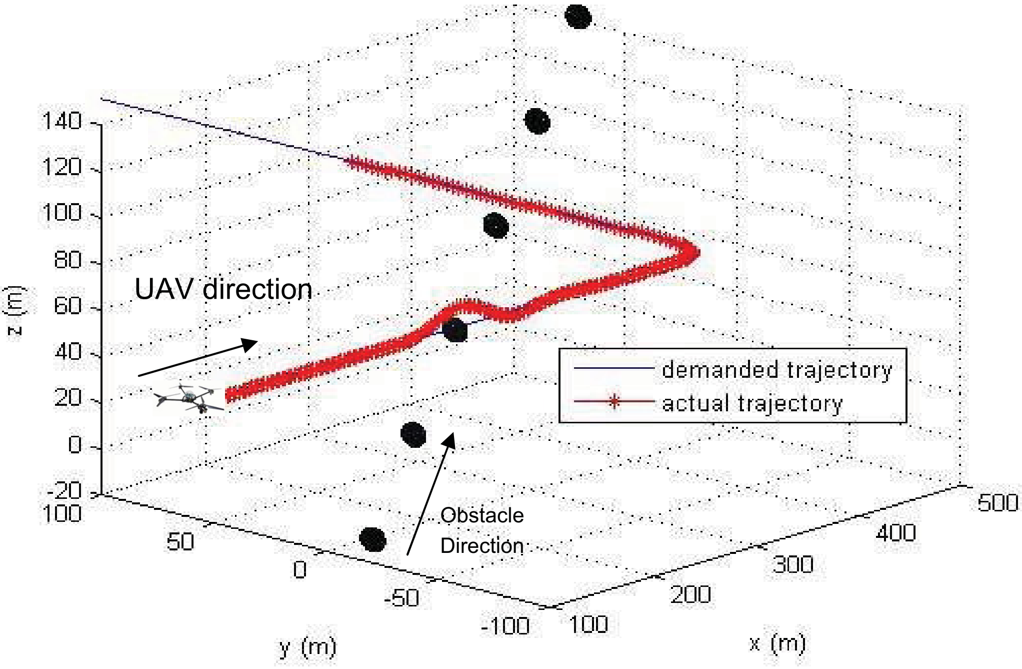


Figure 73 - Single dynamic obstacle example-2

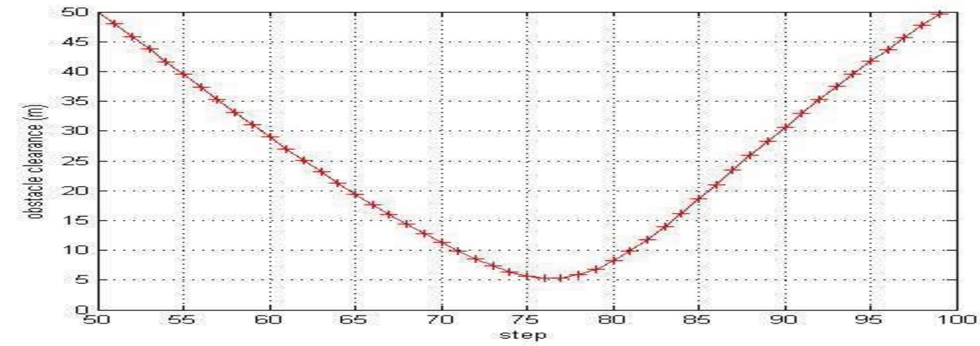


Figure 74 - Single dynamic obstacle example-2 (obstacle clearance)

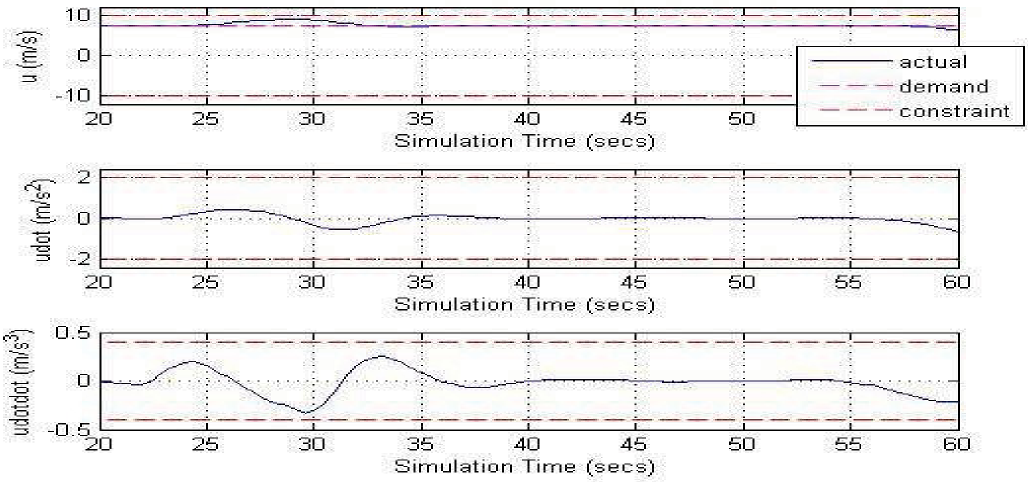


Figure 75 - Single dynamic obstacle example-1 (u-axis time history)

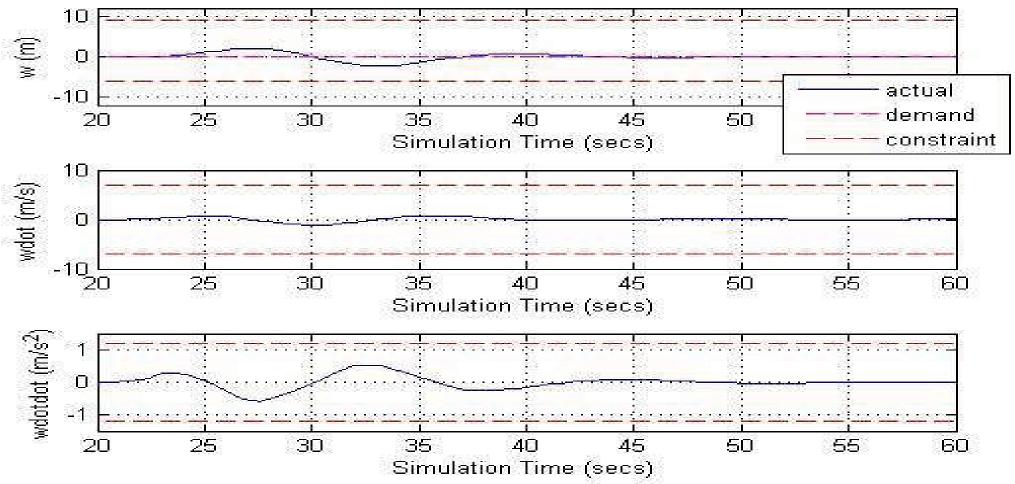


Figure 76 - Single dynamic obstacle example-1 (w-axis time history)

A range of other single obstacle scenarios were also tested with similar results. An animation for example-1 is provided in ‘animation3 - dynamic obstacle1’, and a final scenario is shown in ‘animation4 - dynamic obstacle2‘, where the obstacle approaches the vehicle on a collision course from behind. It is assumed in this scenario that the vehicle is able to detect an obstacle behind it, and the behaviour helps to emphasise that the receding horizon extends forward in time, and not position, therefore predicting the collision and planning an avoidance manoeuvre.

6.3.2 Multiple dynamic obstacle scenarios

The ability of the LMP framework to cope with multiple dynamic obstacle scenarios is demonstrated in Figure 77 to Figure 80, where three unexpected dynamic obstacles all on collision courses are detected while the vehicle is tracking a global trajectory. It can be seen that all three obstacles are successfully avoided, primarily by manoeuvring in the v-axis. A small u-axis slowdown is again used to increase obstacle clearance, but the w-axis is not used at all. A more efficient solution is likely to involve manoeuvring in the w-axis, but this scenario does not provide a gradient to direct the search in this axis. It can be seen in Figure 78 that a larger clearance distance is provide for the first obstacle than the next two. This primarily results from the collision course of this obstacle requiring less manoeuvring than the other two obstacles, but also emphasises that provided clearance distances are not fixed.

For the previous single obstacle scenarios a human pilot should easily be able to cope with the obstacle deconfliction. However, as the number of obstacles increases, the ability of a human pilot to simultaneously monitor all obstacles while planning a safe evasive manoeuvre diminishes. The LMP framework in theory should be able to cope with highly complex scenarios, as long as the vehicle performance limits allow a solution. Another animation example is provided in the file ‘animation5 - dynamic obstacle3’ in which the three obstacle scenario of Figure 77 is made more complex by the additional of a fourth obstacle. In this animation it appears that the receding horizon trajectory passes through the first obstacle, indicating that the current trajectory plan expects a collision. However, it must be noted that the receding horizon spans ten seconds while the obstacle is only plotted at the current time therefore the intersection occurs at different points in time and therefore is misleading.

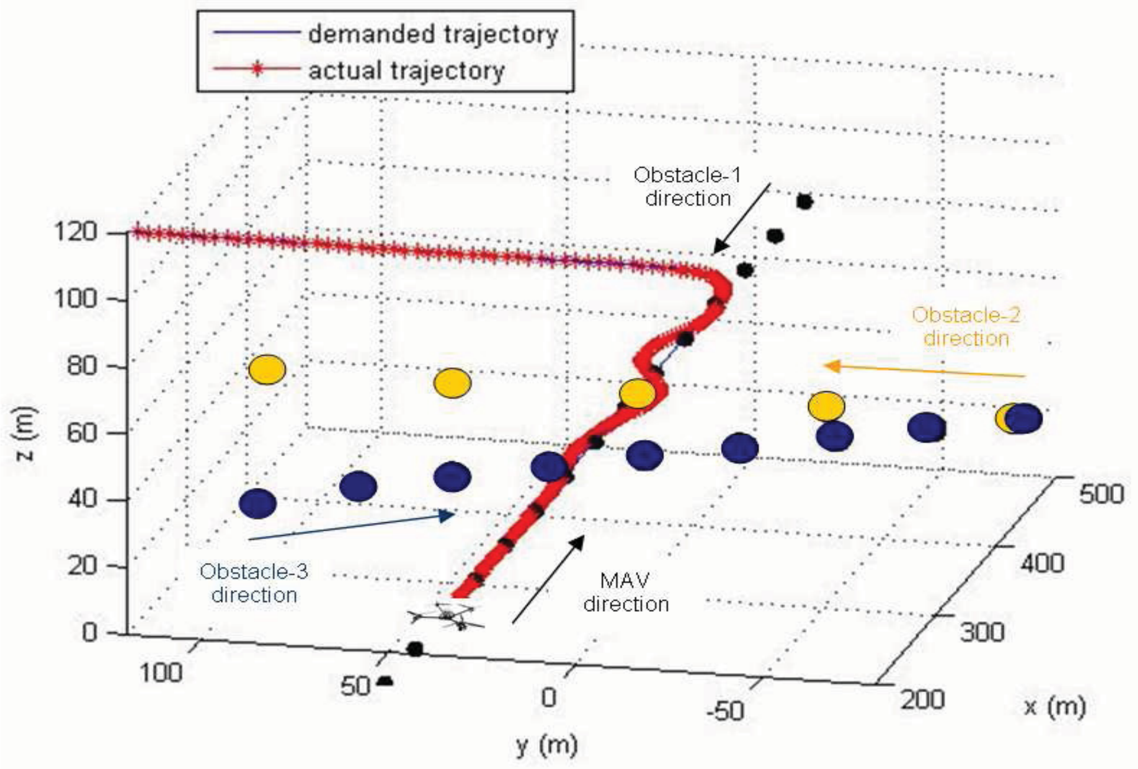


Figure 77 - Multiple dynamic obstacle scenario

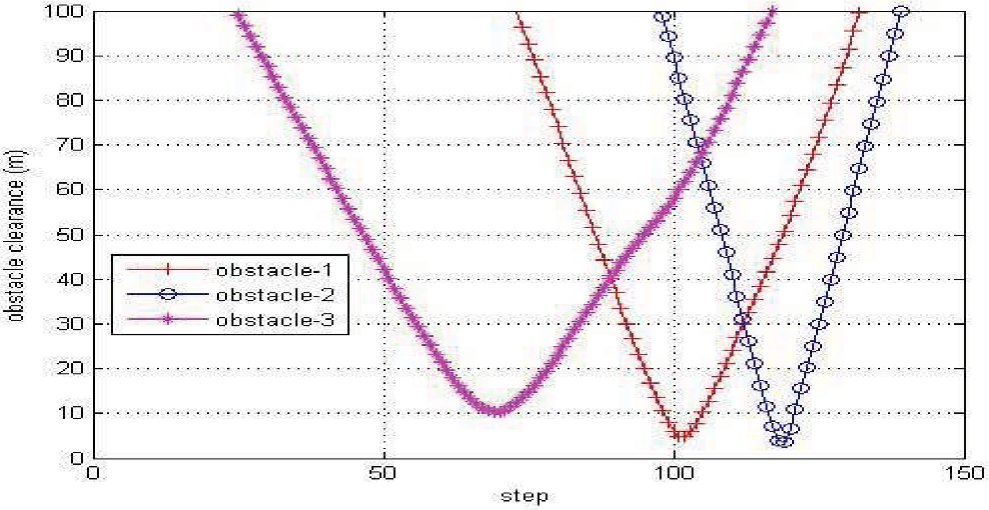


Figure 78 - Multiple dynamic obstacle scenario (obstacle clearance)

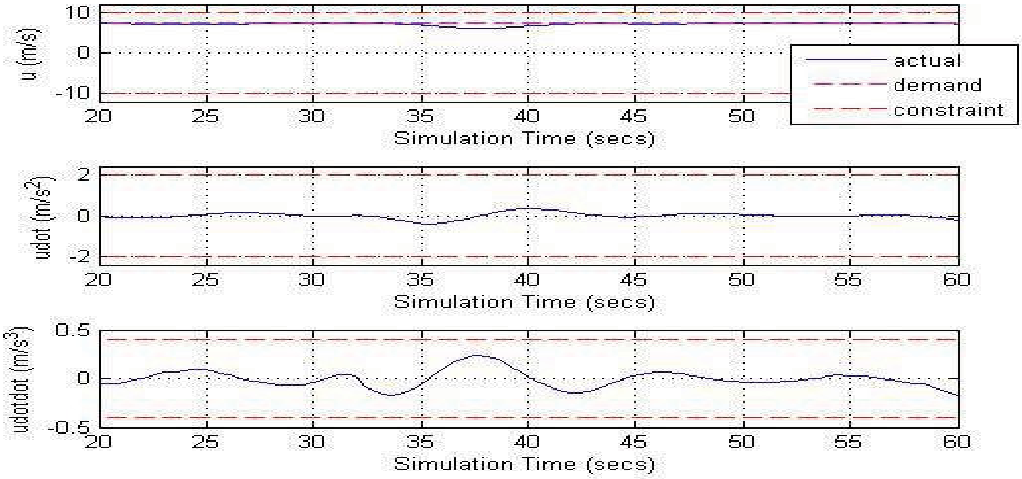


Figure 79 - Multiple dynamic obstacle scenario (u-axis time history)

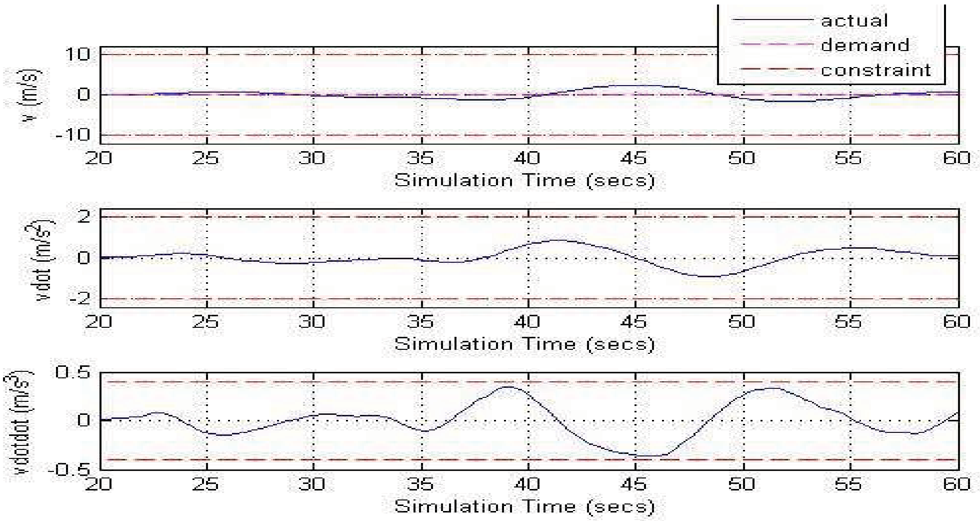


Figure 80 - Multiple dynamic obstacle scenario (u-axis time history)

6.3.3 Challenging scenarios

In order to further investigate the behaviour of the LMP framework a series of more challenging simulation scenarios were executed. These tests were focussed on the following themes:

- *Increasing the size of a single obstacle* - In these test cases the UAV follows a constant altitude global trajectory at 7.5m/s while an obstacle travelling at 2m/s approaches on a collision course at 90deg from the right. The obstacle is a sphere, with the radius increased after each successful test. Tests were performed with the following radii: 1m, 5m, 10m, 15m, 20m, 25m, 30m, 35m, 50m & 75m, and example results are provided in Figure 81 to Figure 83. The collision was successfully avoided in all cases, while also enforcing all vehicle performance constraints. It can be seen that the resulting manoeuvres use approximately maximum vehicle performance (particularly in the $\ddot{u}$ and $\ddot{v}$ profiles), suggesting again that the trajectory description and optimisation is able to extract maximum vehicle performance.

- *Increasing the speed of a single obstacle* - In this series of test cases a single obstacle again approaching from the right (at 90deg) was used, this time with a fixed radius of 2m & a range of approach speeds. The impact of obstacle speed on clearance distance can be seen in Figure 84 where it can be seen that a collision first occurs for an obstacle speed of 22m/s. Avoidance of faster obstacles can be achieved by either increasing the trajectory design rate (to provide faster reaction to newly detected obstacles), or the sensor range.
- *Very large obstacles* - These tests were intended to investigate behaviour when obstacle that cannot easily be manoeuvred around were encountered. The role of the local motion planner here is to prevent a collision while waiting for the global planner to calculate a new trajectory that accounts for the newly detected large obstacle. Animations are provided for two scenarios, firstly in 'animation6 - rising surface' a rising surface forces the vehicle away from the trajectory until the obstacle has been cleared. In the second scenario, shown in 'animation7 - very large sphere', the vehicle encounters a large sphere approaching on a head-on collision course. The sphere is large enough to reduce the avoidance gradient to zero, therefore the vehicle simply maintains a safe clearance from the obstacle while holding the trajectory.
- *Restricted direction avoidance manoeuvres* - These tests were designed such that the proximity gradient from an incoming dynamic obstacle directed the vehicle towards either a ground or overhead surface obstacle. An example is shown in Figure 85 where vehicle trajectories are compared for the incoming obstacle with i) no surfaces, ii) a ground surface only and iii) both a ground and overhead surface. It can be seen that local motion planning framework deals well with the surface obstacles, manoeuvring into the available space rather than getting trapped by the incoming obstacle.
- *Erratic obstacle motion* - In all previous dynamic obstacle examples the obstacle speed was constant, therefore allowing accurate prediction of future positions. In order to test the uncertainty handling approach (section 5.5.3) a series of test were conducted where the obstacle motion was harder to predict. An example of this is shown in Figure 86 where the unsteady obstacle path can be seen. The resulting obstacle clearance distances using several different uncertainty growth models is shown in Figure 87 where it can be seen that without the uncertainty growth model a collision occurs at 20s.

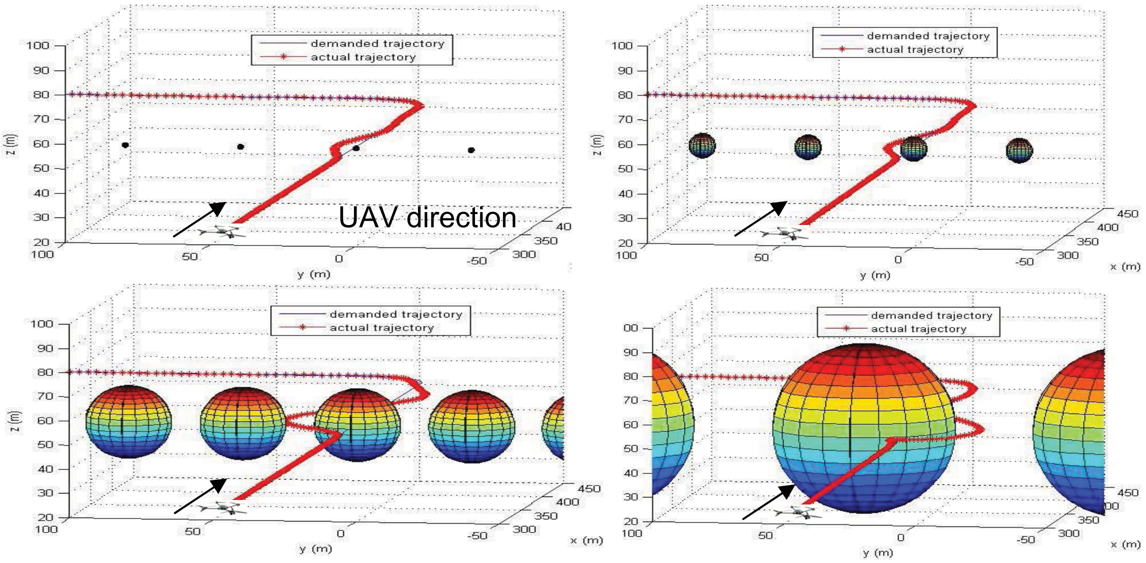


Figure 81 - Increasing obstacle size

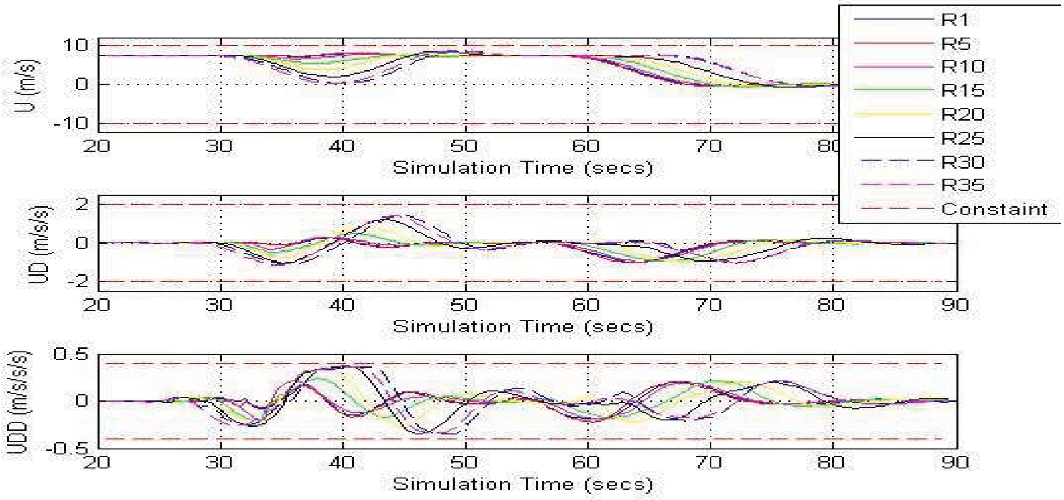


Figure 82 - Increasing obstacle size (u-axis time history)

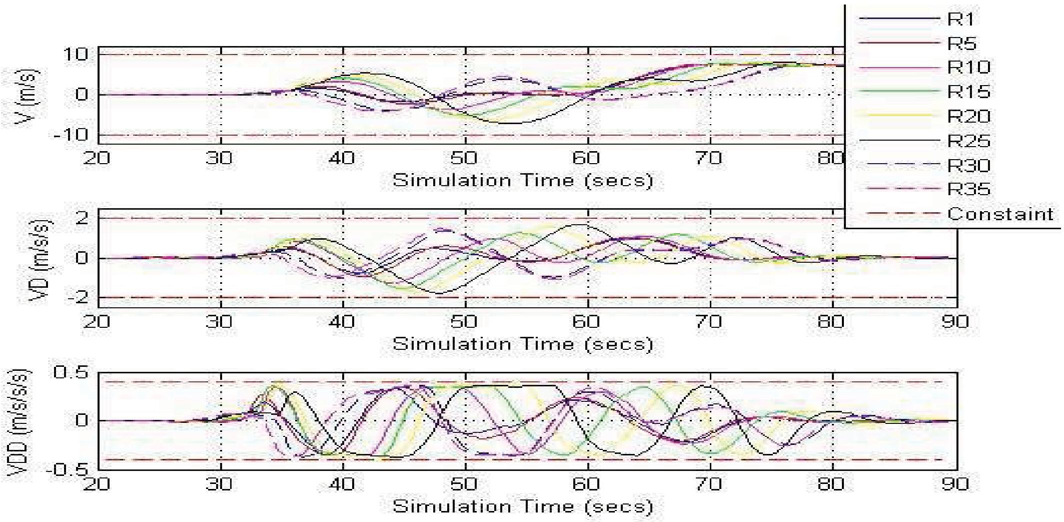


Figure 83 - Increasing obstacle size (v-axis time history)

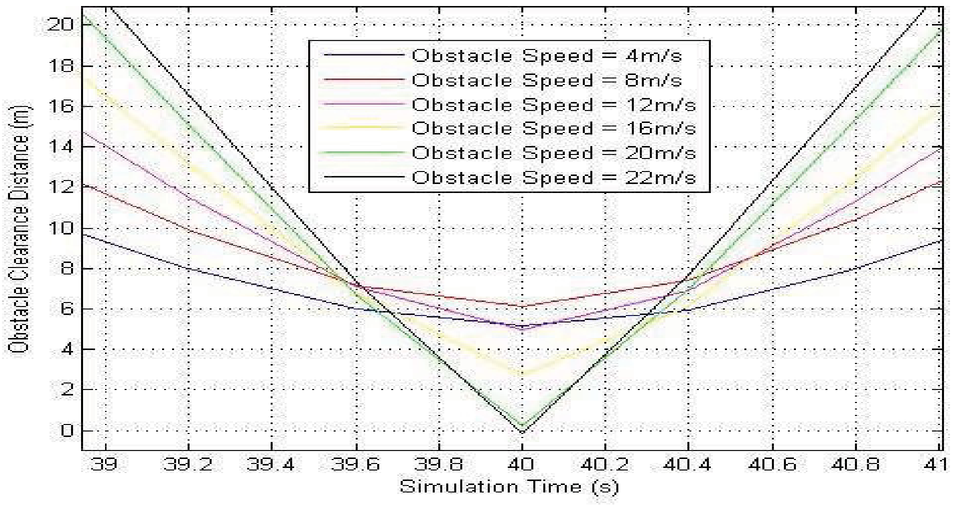


Figure 84 - Reduction in clearance distance with obstacle speed

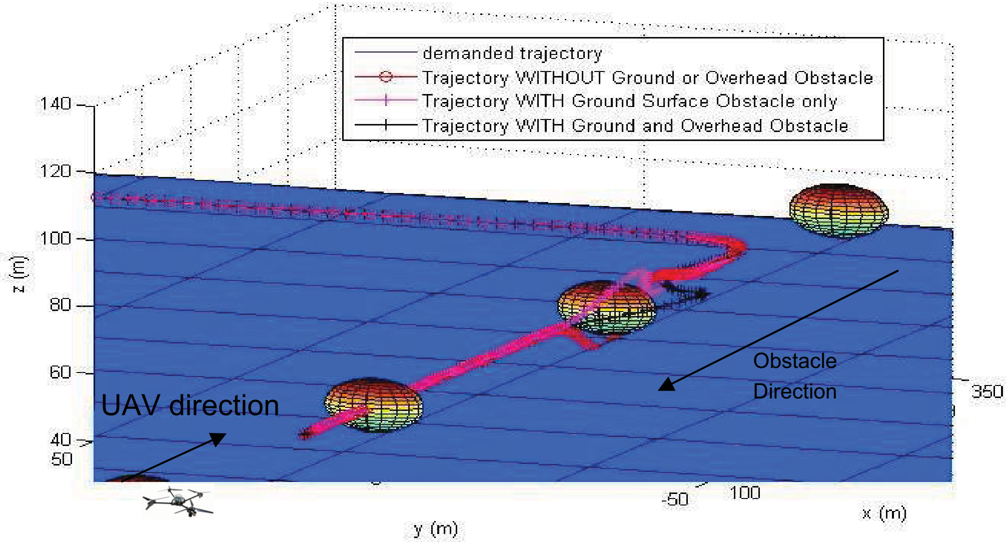


Figure 85 - Impact of Overhead & Ground Surface Obstacles

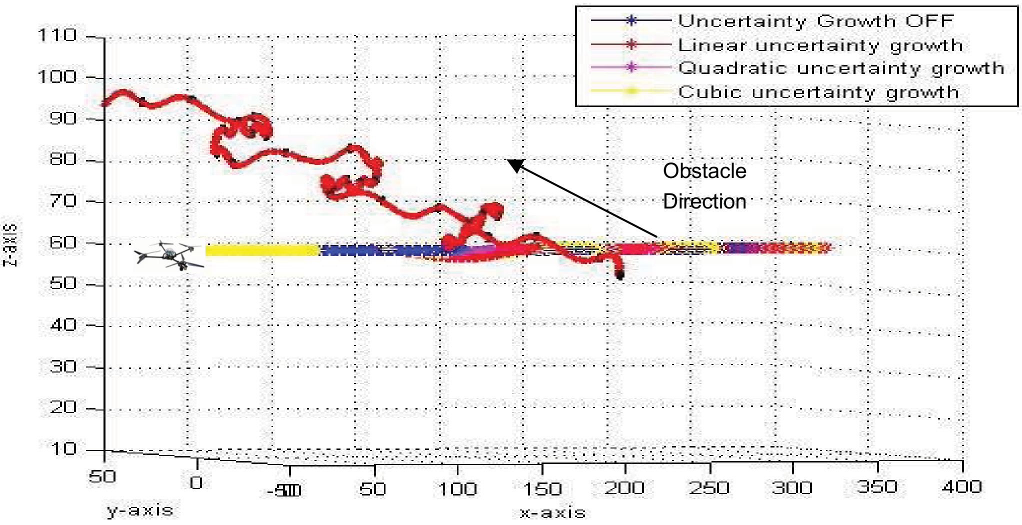


Figure 86 - Erratic obstacle motion

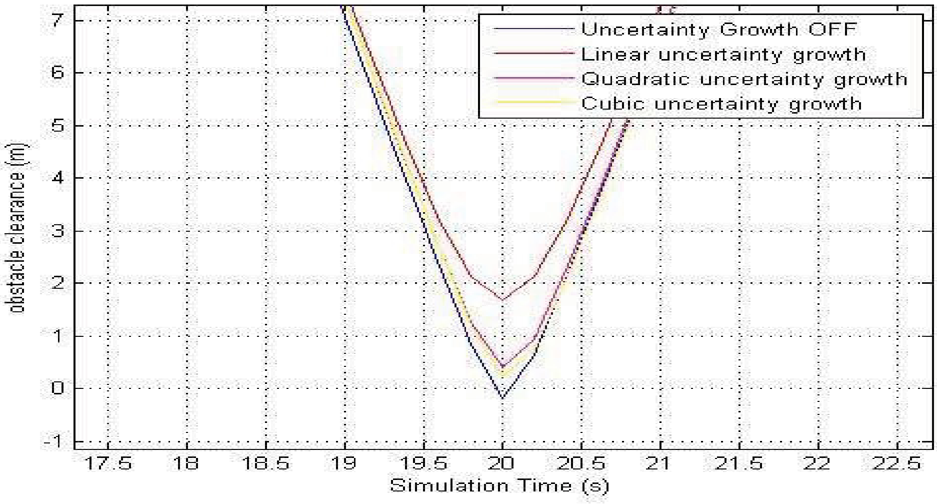


Figure 87 - Erratic obstacle motion (obstacle clearance)

6.4 Avoiding Local Minimums

An example of the ability of the coarse grid of feasible trajectories to avoid local minimums (section 5.6.3) is presented in Figure 88 to Figure 92. The UAV is following a straight & level global trajectory at 7.5m/s when a moving obstacle is detected on a collision course approaching from the right. The obstacle is travelling at 20m/s, the sensor range of the MAV is limited to 60m and the trajectory re-design rate is 0.4secs. It can be seen that with the pure gradient search a collision results at approx. t=40secs, which the event triggered collision avoidance manoeuvre is able to avoid. It is interesting to note that the gradient search attempts to avoid the collision by *accelerating* in the u-axis, whereas the chosen coarse grid manoeuvre at this point *decelerates* in the u-axis. This suggests that the steepest descent search at this time directs the search towards a local minimum, which is unable to avoid the collision. In the pure gradient search the u performance limit is quickly reached, resulting in the avoidable collision.

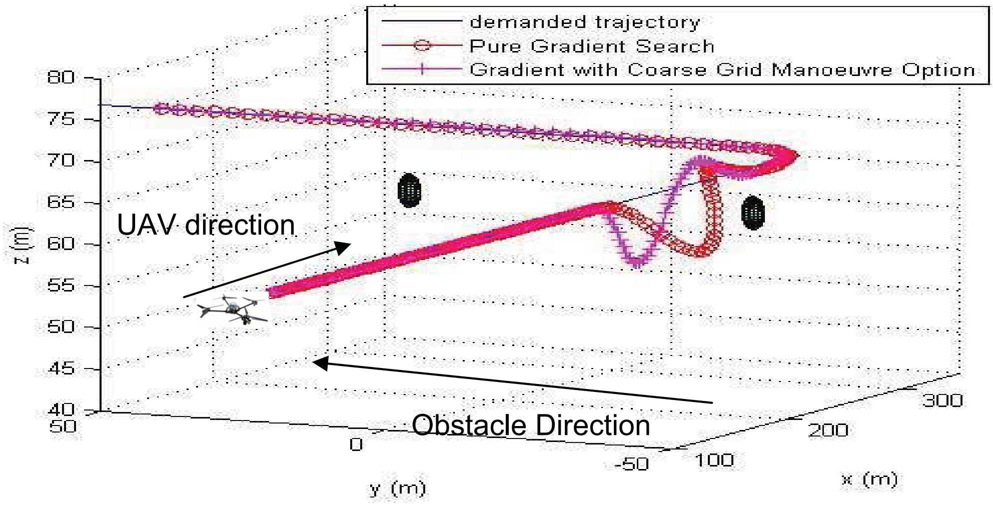


Figure 88 - Escape from local minimum (3D View)

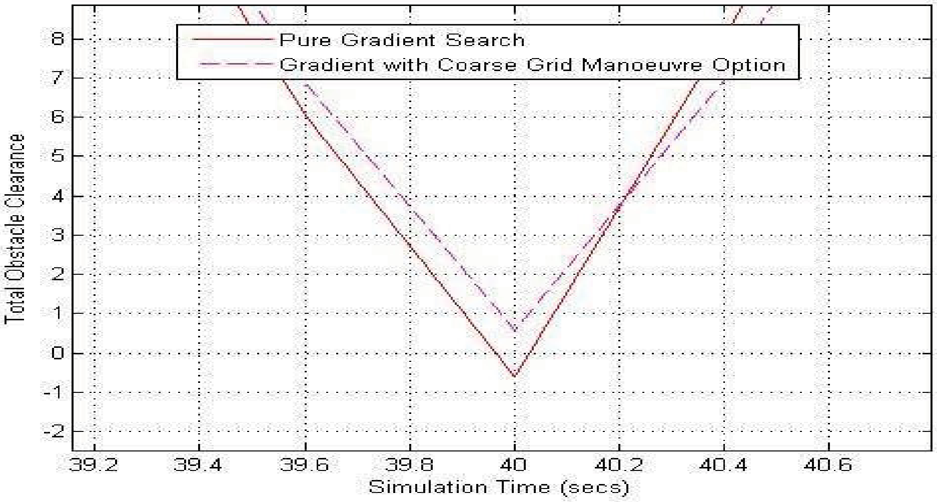


Figure 89 - Escape from local minimum (Obstacle Clearance)

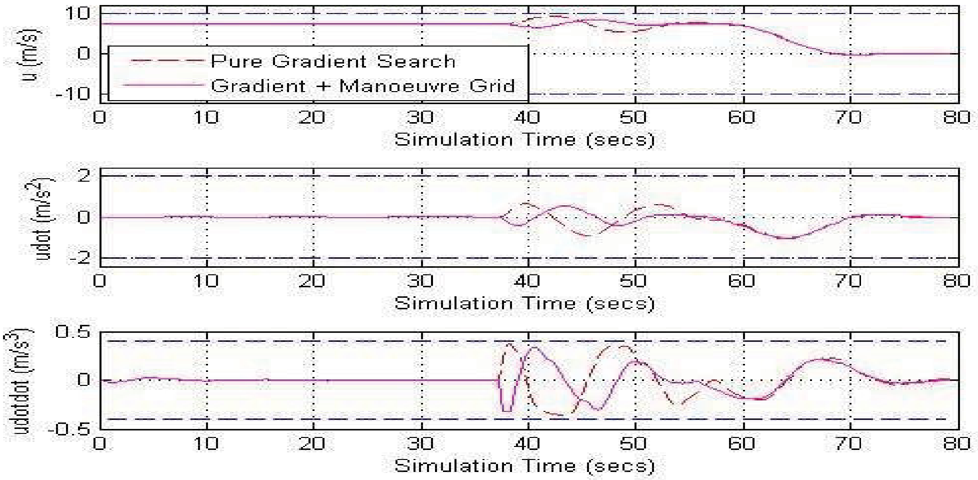


Figure 90 - Escape from local minimum (u-axis time history)

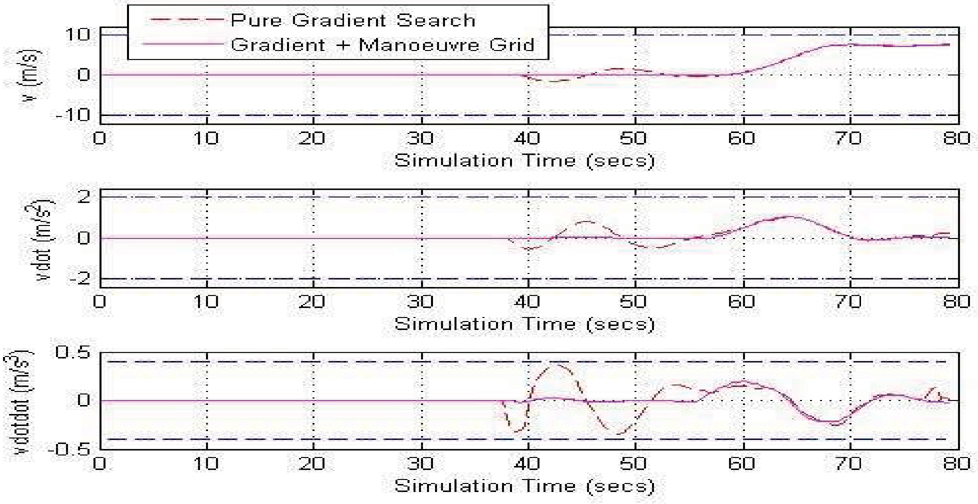


Figure 91 - Escape from local minimum (v-axis time history)

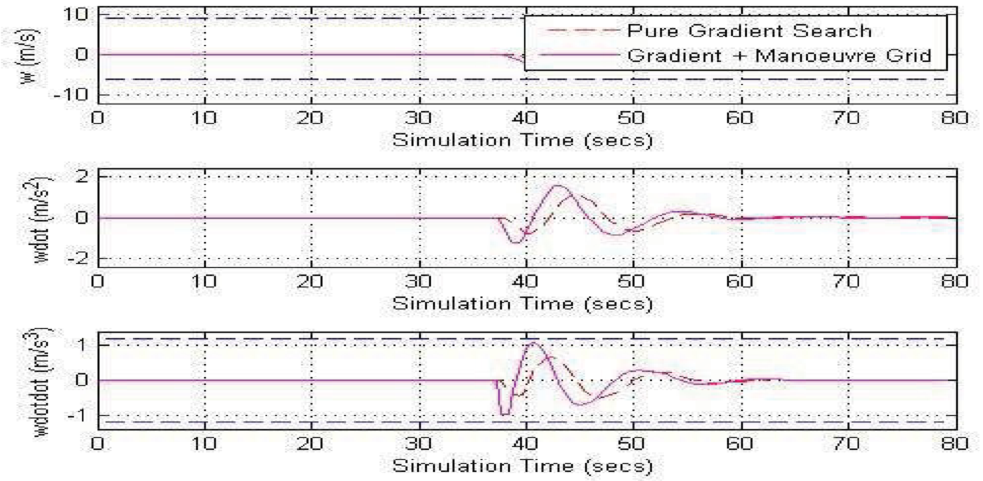


Figure 92 - Escape from local minimum (w-axis time history)

An example of the coarse grid trajectories that are tested at the time when the manoeuvre is first triggered is shown in Figure 93. The lowest cost trajectory in the coarse grid is highlighted with red crosses and the obstacle is shown in two positions, firstly (the lower one) is at the same time at the quad-rotor is plotted, and secondly (the one directly in front of the vehicle) at the time when the collision is 1st projected to occur.

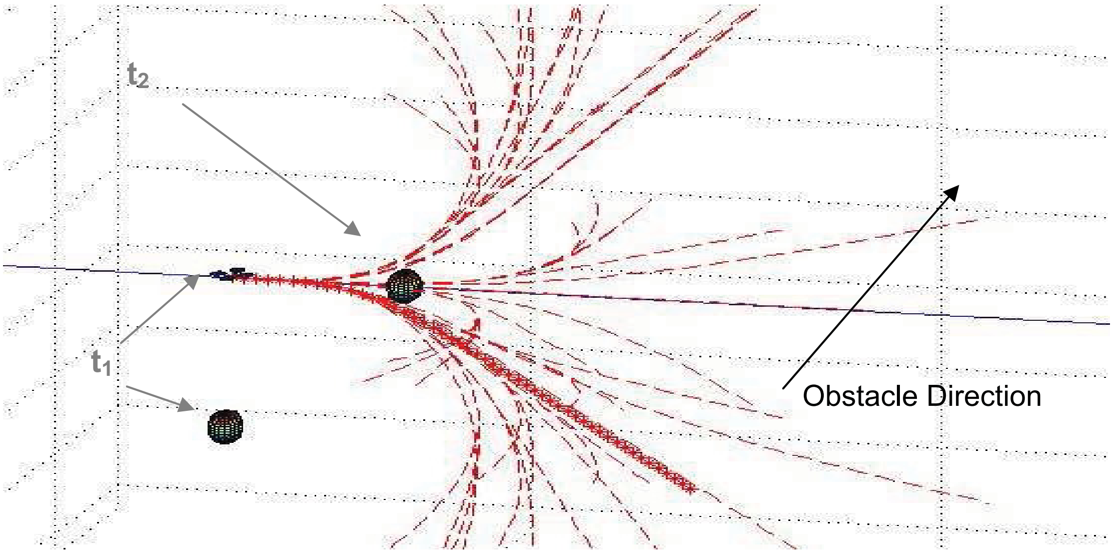


Figure 93 - Coarse Grid Trajectories at Obstacle Detection Time

6.5 Relative Speed Bias

As discussed in section 5.5.4 the preferred obstacle clearance distance may be impacted by the relative speed between the vehicle and the obstacle, and this behaviour can be factored into the proximity based obstacle penalty function by scaling the penalty term. An example of this behaviour for the UAV passing a static obstacle is shown in Figure 94 & Figure 95. It can be seen that as the vehicle speed increases, so does the provided clearance distance, therefore providing the required behaviour. This response is entirely tuneable, therefore desired behaviour (once specified) can be implemented.

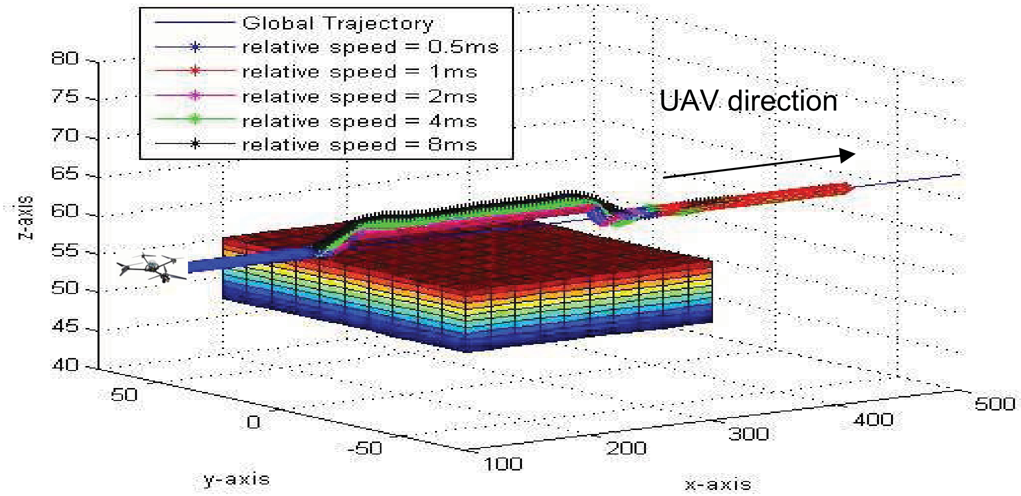


Figure 94 - Relative Speed Test Scenario (static obstacle)

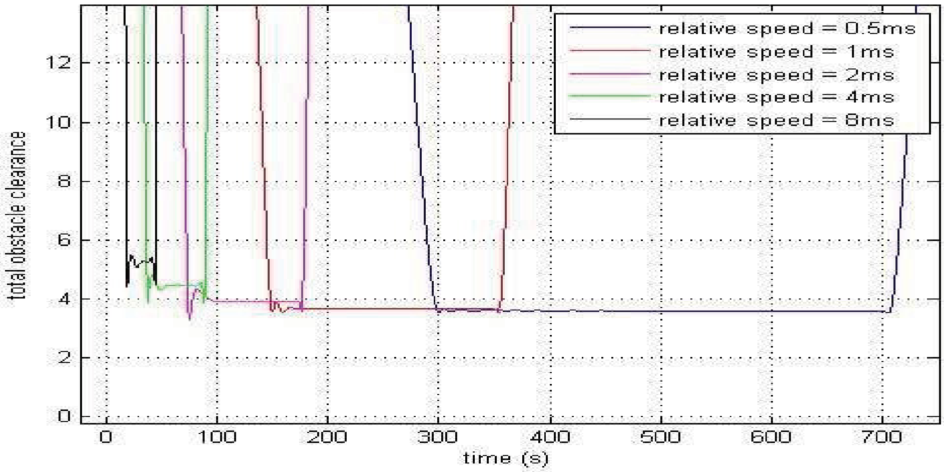


Figure 95 - Impact of Relative Speed on Obstacle Clearance

6.6 Computation Effort

All development and simulation testing discussed within this thesis was performed using Matlab / SIMULINK on the following machine:

- Lenovo T400 laptop, Windows XP Professional, processor = 2.53Ghz, memory = 160GB hard drive & 2GB RAM

Examples of the computational effort required by the local motion planning framework are provided in Figure 96 - Figure 98. It can be seen that peak computational effort for the optimisation algorithm, even in the presence of three dynamic obstacles, is typically below one second. The required effort typically peaks when either a trajectory corner is encountered or a trajectory departure or re-acquire manoeuvre is required, e.g. due to a detected obstacle. The computation effort is also impacted by the number of obstacles to be considered.

Although these examples are slower than required for real-time implementation this simulation was executed using the Matlab m-code language, therefore significant improvements could be realised by using a compiled language such as c-code. Additionally, no effort has been made to optimise the search algorithm, therefore further significant benefits are likely to be achievable. For example, the current implementation of the line search uses a simple fixed step size, therefore use of an accelerating step size or one of the other approaches discussed in section 3.4.6 is likely to be beneficial.

Two other key factors that impact the computational effort are the resolution of the receding horizon trajectories used in the cost function (currently *n*=50) and the order of the polynomial functions (currently 6th order). However, care must be taken with both of these parameters to ensure that performance is not affected.

Finally, it should be noted that the nature of the successive trajectory re-design results in an overall framework that is similar to dynamic programming optimisation, where a single problem is divided into a series of smaller problems. Each receding horizon trajectory design can be sub-optimal while the overall resulting trajectory may still approach optimality. This is aided by the fact that the boundary conditions mean that it is the latter part of the trajectory design that varies most during optimisation, therefore allowing each successive re-design to improve the trajectory before the vehicle reaches it.

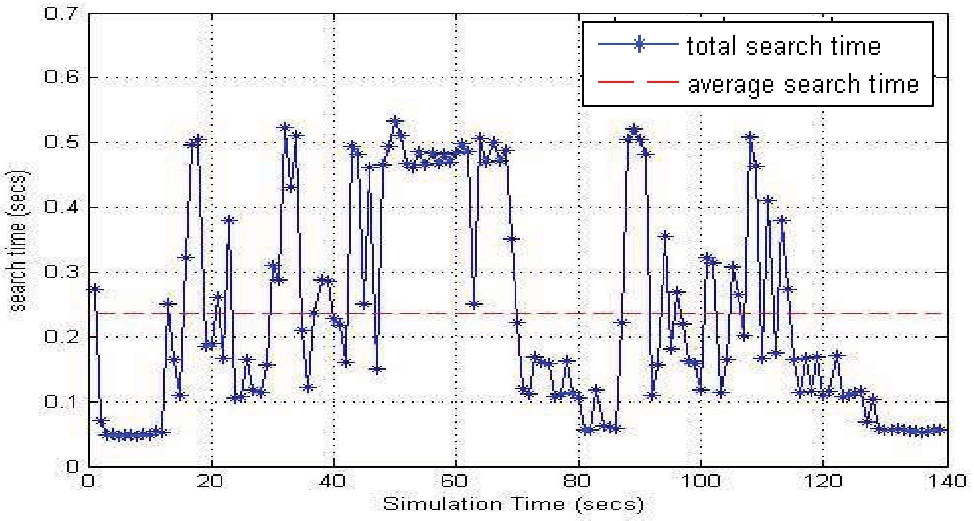


Figure 96 - Computation effort for static obstacle example-2

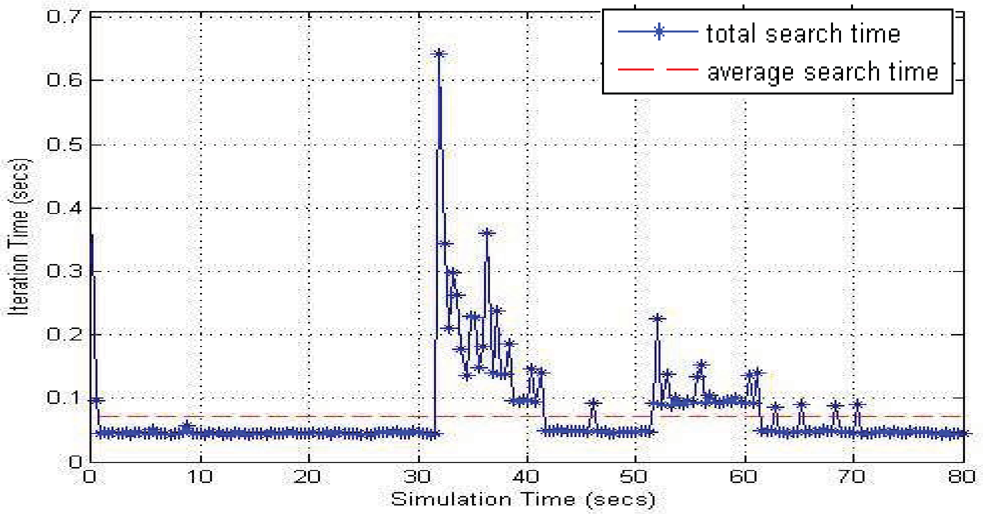


Figure 97 - Computation effort for dynamic obstacle example-1

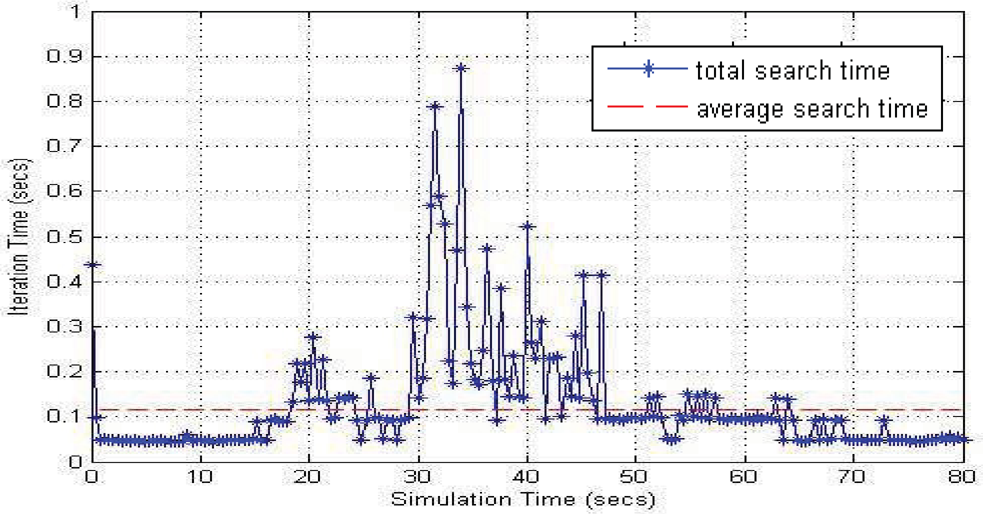


Figure 98 - Computation effort for multiple dynamic obstacle scenario

7 Simulation Results - *Multiple Vehicle Scenarios*

7.1 Discussion of Vehicle Deconfliction Strategies

The aim of this chapter is to demonstrate the use of the local motion planning framework to enable decentralised multi-vehicle deconfliction, the need for which was highlighted in the capability visions presented in section 4.2.3. Before presenting implementation decisions and simulation results, a range of deconfliction strategies are discussed, including rule-based behaviour and deconfliction horizons.

7.1.1 Deconfliction Strategies

A hierarchy of deconfliction strategies is presented below, starting from the most efficient (in terms of traffic flow not computational effort):

- ***Centralised*** - Single planner calculating trajectories for all vehicles. This allows a single planning algorithm to consider all vehicles simultaneously. Calculated trajectories are transmitted to each vehicle for execution. Rare due to complexity of computation & communication. Given that all planning is performed centrally, there is no need for individual vehicles to be in communication or to follow pre-defined sets of rules. Instead, all vehicles must only be in communication with the centralised planner.

  Example - Commercial aircraft flying in controlled airspace under instrument flight rules are de-conflicted by an air traffic control (ATC) centre responsible for the aircraft's current sector. Deconfliction may be via either segregated airspace or specific trajectory routing.

- ***Decentralised, active, rule-following*** - Decentralised, therefore each vehicle plans only it's own trajectory. Active, therefore the vehicles are in communication with each other, and rule-following means that pre-agreed rules or behaviours will be followed.

  Example-1 - Road cars both follow pre-agreed rules & actively communicate via indicators, brake lights, hand signals, horn, flashing headlights etc. Problems occur when expected behaviour is not followed, e.g. turning without indicating. Note that certain signals have different meaning in different scenarios, e.g. flashing light may indicate either i) OK for lane change, or ii) disapproval of lane change, therefore communication can be highly context dependent.

  Example-2 - TCAS-I (like TCAS-II but no advisory manoeuvres – see [15]) uses communication to generate 3D picture of local traffic & alert the pilot to potential collisions. However, each pilot determines their own avoidance manoeuvre based on current scenario and rules-of-the-air.

  Example-3 - 'Free Flight' concept of future air traffic control [25], where responsibility for de-confliction is passed from ATC to individual aircraft.

Communication is via ADS-B, and various rules have been proposed to try to allow efficient avoidance manoeuvres without requiring negotiation between vehicles.

- ***Decentralised, active*** - As above but no pre-agreed rules.

  Example - Several unmanned vehicles all independently planning receding horizon trajectories which are transmitted to all neighbouring vehicles. May be effective for groups of vehicles of the same type, but integration with manned vehicles would require visualisation of the intentions of each vehicle, e.g. head-up projection of planned trajectories.

- ***Decentralised, passive, rule-following*** - Each vehicle plans only it's own trajectory, pre-agreed rules are followed but there is no communication between vehicles.

  Example - General aviation & non-TCAS traffic fits into this category as there is no manoeuvre related communication between individual vehicles.

- ***Decentralised, passive*** - As above, but there are no pre-agreed rules.

  Example - Pedestrian traffic generally doesn't have agreed rules to follow. This works OK for slow traffic until the environment becomes very crowded, e.g. a busy station, then it all gets a bit ad hoc, with progress often determined by indicating to others your intent to keep going.

Consideration was also given to subdividing the above hierarchy further by differentiating between active approaches that provide inter-vehicle negotiation and those that are simple transmit / receive systems. For example, TCAS-II uses inter-vehicle communication to negotiate preferred manoeuvres for each vehicles. However, negotiation can be considered to create a virtually centralised approach, therefore this was not considered necessary.

The choice of deconfliction approach is driven largely by practical issues such as the constraints of individual applications. For example, segregated airspace provides the greatest level of safety, but at the expense of rapid reaction, flexibility and traffic density. At the other end of the spectrum a rule-based decentralised approach avoids the need for formal inter-vehicle communication / negotiation, and allows greater traffic density, although a degree of optimality is sacrificed by decentralising the problem[37]. For small / micro UAVs the required low probability of collision would be provided in practice by using a combination of methods, e.g. altitude ceiling or exclusion zone to be clear of manned vehicles, then coordinated routing within each system, with decentralised deconfliction as a final safety measure.

[37] For example, traffic flow on the road network can be significantly improved via increasing centralised control, e.g. dynamic speed limits can slow traffic, reduce vehicle separation and increase overall throughput..

It is therefore likely that for the current small / micro UAV application decentralised deconfliction would play a key role, reducing reliance on complex coordination between multiple operators, enabling rapid deployment and an increased traffic density. Additionally, even when some form of centralised routing is present certain mission segments would still require a decentralised response, e.g. if a disturbance or obstacle is detected during a formation ingress / egress then each vehicle would be required to separate then reform the formation without conflicting with each other.

7.1.2 Rule Based Deconfliction

Rule based deconfliction is used in various domains (e.g. the rules of the road, or rules-of-the-sea, rules of the air etc.), primarily as an aid to passive decentralised deconfliction. The existence of these rules allows the likely actions (or intent) of other vehicles to be predicted, reducing the chance of a conflict deteriorating, or even leading to a collision. Additionally, rules may be used to increase the efficiency of a traffic network, for example by helping to organise flows in different directions. Pre-agreed rules are often based on establishing priority, which may be achieved by either relative position (e.g. give way to right) or vehicle classification (e.g. powered aircraft give way to gliders, gliders give way to balloons, etc.), although this is likely to be problematic for unmanned vehicles due to the need for automatic vehicle recognition and classification. An additional issue that arises for unmanned vehicles is knowing whether or not the other vehicle will follow the rules. This again becomes a classification problem which is further complicated by the need to know if the other vehicle is aware of the potential conflict or not.

Rule based behaviour often results in a preference to pass on one side of a vehicle, e.g. in a head on scenario both vehicles turn right. However, this type of rule becomes ambiguous as the miss distance reduces, for example at what distance is it no longer acceptable to pass directly ahead of another vehicle to get to the desired side? Finally, if using pre-agreed rules to anticipate actions of another vehicle care must be taken to also account for the local obstacle space or the presence of other vehicles, e.g. cannot expect the other vehicle to manoeuvre in a certain direction if this may result in undesired proximity to an obstacle. This adds a new obstacle dimension to the prediction of the other vehicle’s behaviour which again complicates implementation for unmanned vehicles.

7.1.3 Deconfliction Horizons

The previously discussed deconfliction options operate over a range of time horizons. This was also discussed in Section 3.3.5 where the following classification was presented:

- *Long-term* - Several hrs, e.g. pre-flight approval at the level of the entire aerospace system.
- *Mid-term* - Tens of minutes, e.g. in-flight modifications of flight plans by air traffic control.

- *Short-term* - Seconds to minutes, e.g. on- board flight management system based conflict detection and avoidance (TCAS).

For the local motion planning framework presented in this thesis a fourth horizon can be added to this list:

- *Immediate-term* - Up to the receding planning horizon, e.g. less than 10s.

The focus of this horizon is within the region where pre-agreed rules may become ambiguous, and guidance decision priority switches towards increasing miss-distance. Biases to the pure-proximity based behaviour and certain rules may still be beneficial, although these may take a different form to, or be a limited set of, the traditional rules of the air / ground / sea etc. as these tend to be aimed more towards the short or medium term horizon.

7.2 Planning Framework Updates for Multiple Vehicle Deconfliction

Based on the above discussion of deconfliction strategies, the LMP framework was updated to aid the provision of *immediate-term* decentralised deconfliction of multiple unmanned vehicles. The following key behaviours addressed are:

- *Biasing the pure-proximity based cost function to aid provision of rule-based behaviours.*
- *Vector product based immediate term evasive manoeuvres.*
- *Communication of intent via local broadcast of receding horizon trajectories.*

Implementation of each of these behaviours is discussed further below.

7.2.1 Cost Function Bias-1: Future Cost of Current Position

Using a pure-proximity based obstacle cost function results in the chosen avoidance direction being based only on the geometry of the scenario. This results in manoeuvres that pass directly ahead of an obstacle having the same cost as those passing directly behind. Although being directly ahead of the obstacle doesn’t reduce the distance separation, a human pilot would intuitively prefer to pass behind a moving obstacle as being directly ahead of it places the vehicle only moments from a collision. This is an example of a rule-type behaviour that may be desirable to implement.

Calculation of the pure-proximity based cost function is illustrated in Figure 99, where it can be seen that the proximity used at each step along the horizon is based on matching the time for both vehicle and obstacle positions. One method of providing the desired behaviour is to update this calculation to also consider the cost of future obstacle positions. This is illustrated in Figure 100, where at each time-step the cost is calculated based on vehicle position and the predicted obstacles positions, not only at the current time-step, but also over the remaining time-steps in the receding horizon. Simulation results demonstrating the impact of this calculation are provided in Section 7.3.1.

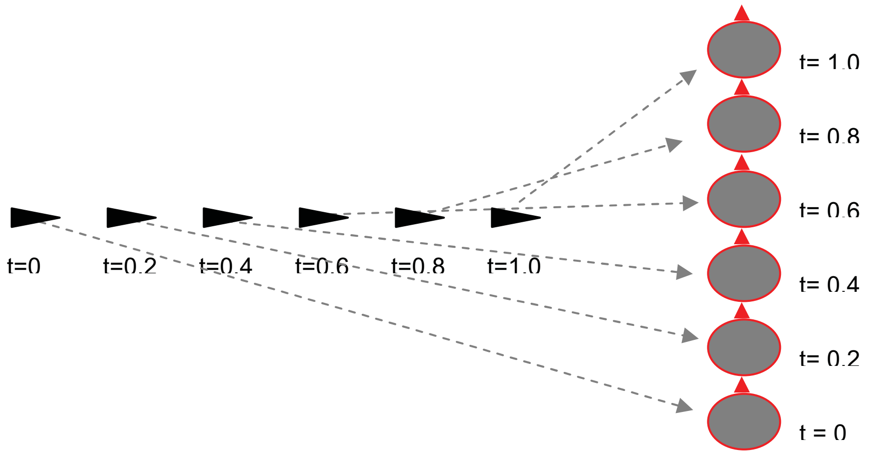


Figure 99 - Proximity Cost Comparing Matching Time Positions Only

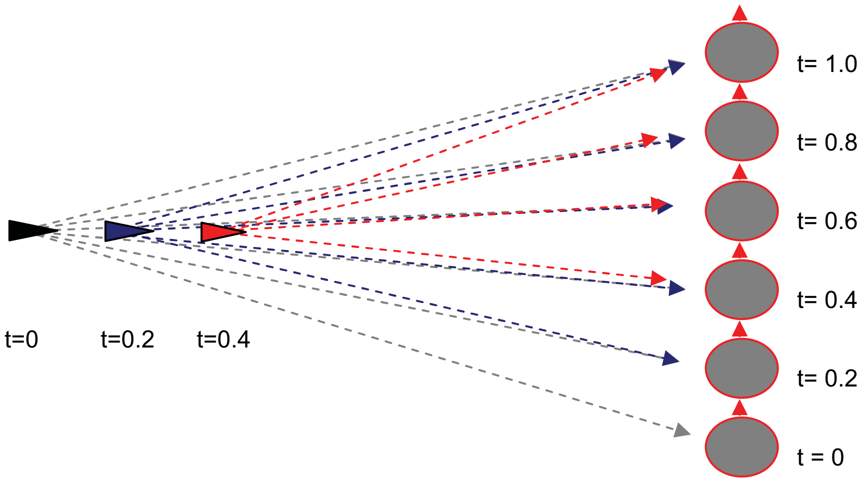


Figure 100 - Proximity Cost Comparing Future Cost of Current Positions

7.2.2 Cost Function Bias-2: Velocity Vector Avoidance

Another way of implementing rule-type behaviour is to bias the obstacle proximity cost function to give higher costs in certain undesired directions. For example, if it is preferred to pass to the right of a vehicle, then the cost function may punish positions on the other side more, therefore encouraging greater separation on the less desired side. Alternatively, positions facing the velocity vector of the other vehicle may be punished more than those behind it. This section presents a method for implementing this type of behaviour.

The proximity based obstacle cost function (section 5.5.2) is given by: $C_{obstacle} = A * \frac{e^{(-\alpha d)}}{d}$

This cost function may be shaped towards giving a higher cost in certain locations by altering either the scaling factor (A) or decay rate (α). The relative speed bias discussed in section 5.5.4 uses the scaling factor, therefore the decay rate is used here, the impact of which can be seen in Figure 34.

The formulation below shows how to bias the proximity function away from the obstacle velocity vector, although other behaviour may also be provided by basing the calculation on a different vector. A cone around the obstacle velocity vector can be considered as shown in Figure 101. In this figure the obstacle is at the origin of the velocity vector.

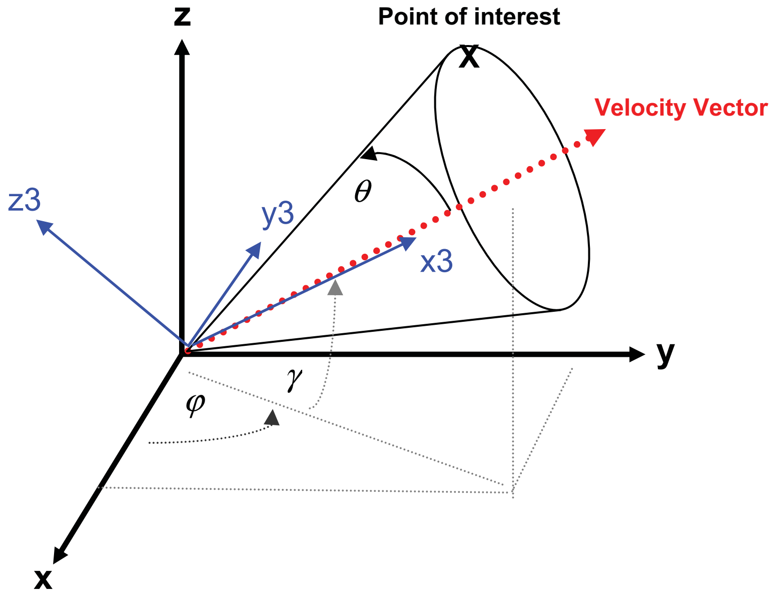


Figure 101 - Biasing a proximity field towards a velocity vector

A point of interest can be related to the cost function by calculating the cone-angle (θ) that this point makes around the velocity vector (note that the base of the cone is perpendicular to the velocity vector). To calculate this angle, transform the coordinate frame into one that has it's x-axis aligned with the velocity vector. This can be done as follows:

- Calculate φ, γ from the x-y-z coordinates to the velocity vector of interest
- Rotate x-y-z by φ around the z-axis to get x2-y2-z2 frame
- Rotate x2-y2-z2 by γ around the y2-axis to get x3-y3-z3 frame
- The transformation can be performed via a matrix calculation:

$$A1 = \begin{bmatrix} \cos\varphi & \sin\varphi & 0 \\ -\sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}, \quad A2 = \begin{bmatrix} \cos\gamma & 0 & \sin\gamma \\ 0 & 1 & 0 \\ -\sin\gamma & 0 & \cos\gamma \end{bmatrix}, \quad P_{x3,y3,z3} = A2 \times A1 \times P_{x,y,z} \tag{7-1}$$

Once the point of interest is known in the new frame the cone angle (θ) can be calculated as follows:

- The coordinated of the point of interest in the new frame are given by: P_{x3}, P_{y3}, P_{z3}

- If $P_{x3} <= 0$ then the point of interest is behind the velocity vector, therefore no change to the cost function
- Else, $radius_{cone} = \sqrt{P_{y3}{}^2 + P_{z3}{}^2}$, therefore $\theta = \tan^{-1}(radius_{cone} / P_{x3})$ (7-2)

Now, it is also desirable to smoothly blend the increase in the cost function from maximum to normal as θ varies from +/- 90deg. This can be achieved by using a scaled cosine function as shown in Figure 102, where a default cosine function is compared with several shaped versions.

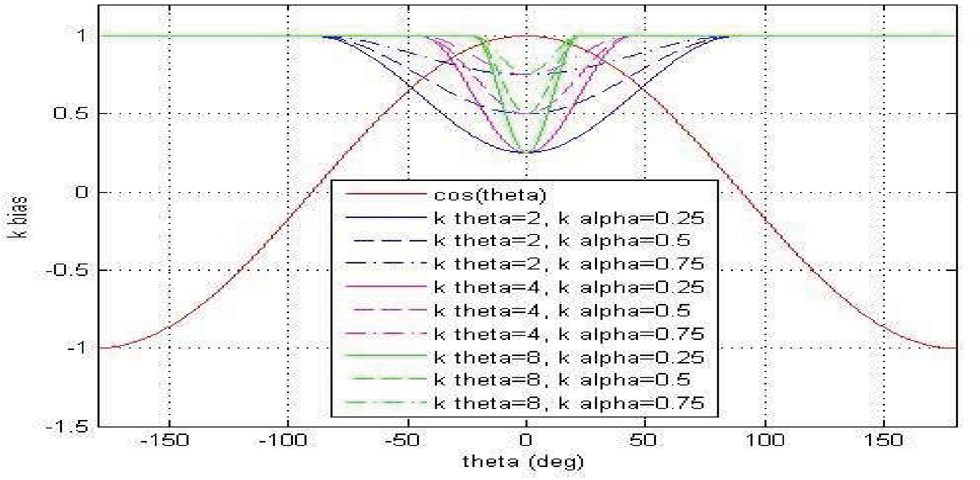


Figure 102 - Shaping the drop-off rate via the cone angle (θ)

These functions can be used to calculate a scaling factor k_α that can be used to scale the value of α used in the proximity function. The calculations are as follows:

- First limit theta to between a defined range of interest:

$$\frac{-\pi}{k_\theta} \le \theta \le \frac{\pi}{k_\theta}$$

- Then scale theta back up to get the full range of the cosine function:

$$\theta = \theta \times k_\theta$$

This has the effect of sharpening the cosine wave around $\theta = 0$ as shown in Figure 102. The wave then has to be scaled appropriately and set to be between a minimum value (for maximum effect - as reducing α results in a slower drop-off rate) and 1 for no effect. This is done as follows:

$$k_{bias} = \frac{1}{2}\{1 - \cos\theta + k_\alpha \times (1 + \cos\theta)\} \qquad (7\text{-}3)$$

therefore giving:

$$\alpha_{bias} = \alpha \times k_{bias} \qquad (7\text{-}4)$$

The key characteristics that can be varied are:

- Sharpness of the drop-off around the velocity vector (via k_θ)
- Minimum value to scale α by (e.g. do not want to let α get down to zero, as this will still provide a large cost far away from the obstacle (via k_α)

An example of the impact of this proximity field bias is provided in Figure 103, which shows the cost function resulting from a slice of x-y positions (with z=constant). It can be seen that the cost drops off much slower towards the right side of the plot (where the velocity vector of the obstacle is pointing) that it does in any other direction, therefore providing the desired shape. Simulation examples of this impact of this cost function shaping are provided in Section 7.3.2.

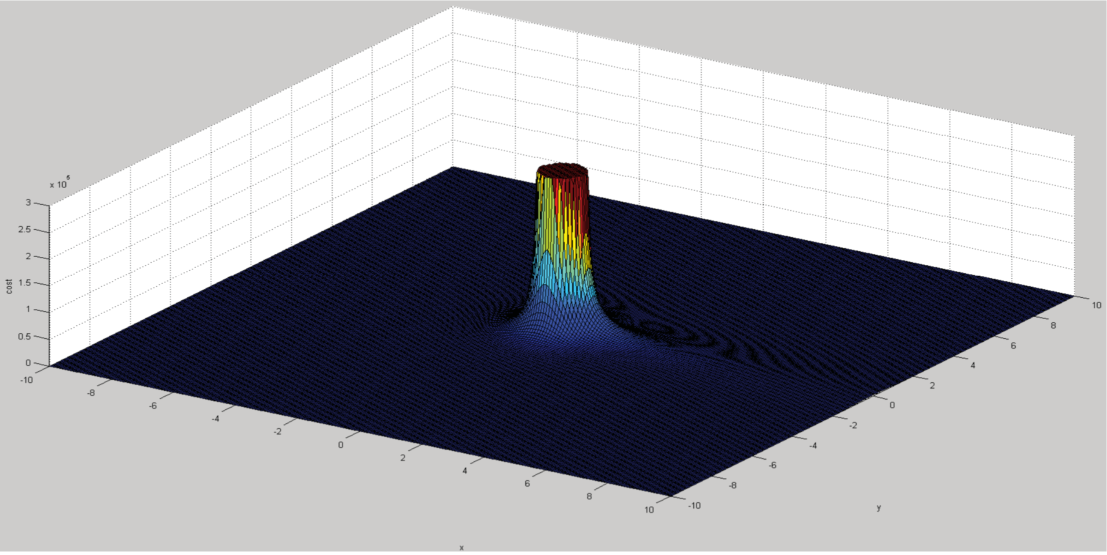


Figure 103 - Example Biased Proximity Field

7.2.3 *Cost Function Bias-3: Cross Product Avoidance Direction*

It is possible to use the cross product of the velocity vectors of two vehicles to calculate evasive manoeuvre directions that guarantee that the vehicles are directed away from each other. Each vehicle's velocity vector defines a line in 3D space, therefore both velocity vectors can be used to define a plane in 3D space. The cross product of these two vectors defines a vector that is at right angles from this plane. Critically, the cross product is non-commutative ($V_1 \times V_2 = -(V_2 \times V_1)$) which guarantees that if each vehicle manoeuvres towards cross product vector then they will be directed to different sides of the plane, and therefore 180deg away from each other.

This rule based behaviour was discussed in [25] as a candidate approach for providing short or medium term conflict resolution manoeuvres within the free-flight concept, although it was not taken forward as the manoeuvres were sometimes seen as sub-optimal. However, for the *immediate-term* deconfliction of interest here it can be used to provide an additional bias on top of the proximity based deconfliction that ensures vehicles manoeuvre away from each other. This approach does not require inter-vehicle communication, it only assumes that each vehicle is able to measure the velocity vector of the other vehicle.

The desired avoidance direction can be encouraged by adding a bias to the predicted position of the other vehicle or obstacle using the preferred avoidance vector. The scale of bias is increased from zero at the start of the design horizon to a maximum value at the end, as illustrated below:

- Velocity vector of vehicle-1 $V_1 = (u_1, v_1, w_1)$
- Velocity vector of vehicle-2 $V_2 = (u_2, v_2, w_2)$
- The avoidance direction used by vehicle-1 is calculated as follows:

$$V_1 \times V_2 = \det \begin{bmatrix} i & j & k \\ u_1 & v_1 & w_1 \\ u_2 & v_2 & w_2 \end{bmatrix} = \left(u_{cp}^1, v_{cp}^1, w_{cp}^1\right) \qquad (7\text{-}5)$$

- This avoidance vector can then be factored into the predicted positions of vehicle-2 across the design horizon as follows:

$$\begin{aligned} x_{bias}^i &= x_{actual}^i + \frac{u_{cp}^1 \times i \times k_{\max}}{n} \\ y_{bias}^i &= y_{actual}^i + \frac{v_{cp}^1 \times i \times k_{\max}}{n} \\ z_{bias}^i &= z_{actual}^i + \frac{w_{cp}^1 \times i \times k_{\max}}{n} \end{aligned} \qquad (7\text{-}6)$$

Where:

x_{bias}^i *= updated predicted position of vehicle-2 at point i in the receding horizon*

x_{actual}^i *= predicted position of vehicle-2 at point i in the receding horizon*

i *= current position in the receding horizon*

$k_{\max}$ *= maximum shift in the predicted position due to the cross product avoidance direction (=1m in shown examples)*

n *= number of steps in the receding horizon (=50 in all shown tests)*

Note that prior to evaluating equation 7-6 the avoidance direction vector must be normalised to between 0-1.

This approach doesn't prevent the usual proximity function from driving the vehicle apart, it simply adds a bias that will help also to drive the vehicles in opposite directions. This pre-agreed rule allows the intent of the other vehicle to be factored into it's predicted position across the design horizon, then the chosen avoidance manoeuvre is finally provided by the usual proximity based approach which also considers the remaining obstacle space.

The maximum shift ($k_{\max}$) can be set to suit the application. A small value (e.g. $k_{\max} = 0.1\,\mathrm{m}$) will only affect behaviour for conflicts where two vehicles are heading directly for each other, with proximity the dominating effect. A larger value (e.g. $k_{\max} = 5\,\mathrm{m}$) is able to force the manoeuvre direction, although this may be problematic as the obstacle position prediction would no longer be accurate.

As this approach is implemented by biasing the predicted positions rather than forcing chosen manoeuvre directions it can easily be implemented for multiple vehicles. Each vehicle considers conflict pairs of itself and any other vehicle, therefore updating it's own believe about the position of the other vehicles over the design horizon.

Note that the cross product of two vectors has a singularity if the vectors are parallel. This can be handled as follows:

- If both vectors are the same the both vehicles are parallel, therefore no avoidance action required.
- If the vehicles are approaching head-on then direct both to perform a right-turn in axis defined by their velocity vectors.

Simulation results demonstrating the impact of this approach are provided in Section 7.3.3, Section 7.4, Section 7.5 and Section 7.6.

7.2.4 Communication of Receding Horizon Trajectory Intent

Another option for improving intent information is inter-vehicle cooperation. An example of this is the TCAS system, where each vehicle transmits and receives current state data, allowing a traffic picture to be built up and potential conflicts detected. While TCAS is predominantly a short / medium term solution (14nm range & 15-35s advisory thresholds), the local motion planning framework provides an opportunity for immediate-term cooperation (e.g. less than 10s), where transmission of receding horizon trajectories may be used to provide accurate intent information. As the receding horizon trajectories are fundamentally local information, the transmission range of current trajectory intent can be very short, e.g. <200m, therefore reducing power consumption. Such sharing of receding horizon trajectory intent is likely to be capable of providing significant improvements over an uncooperative approach, where intent must be inferred purely from behaviour.

It should be noted that this approach requires only a simple transmit / listen system, and therefore doesn't require any inter-vehicle negotiation. Simulation examples demonstrating the impact of sharing current receding horizon trajectories are provided in Section 7.4, Section 7.5 and Section 7.6, where perfect communication is assumed.

7.3 Impact of Framework Updates on Obstacle Deconfliction

Prior to presenting multiple vehicle simulation results, the basic behaviour that is provided by the framework updates discussed in Section 7.2 is demonstrated for single obstacle deconfliction.

7.3.1 Future Cost of Current Position

This section presents a simulation example of the impact of the future cost of current obstacle positions (Section 7.2.1). In this simulation the future cost was added for only 10 time steps from the current time. Using a time horizon of $t_{horizon} = 10s$ and trajectory resolution of $n = 50$ this results in additional cost terms being added for two seconds after the current time. This period can be considered a safety time margin, similar to the safe clearance distance, within which additional future risk is considered, i.e. if the collision is more than this margin in the future then it is considered not to a sufficient risk as to further alter the vehicle trajectory.

The impact of this addition obstacle cost calculation can be seen in Figure 104 to Figure 106, where the results with and without the future cost update are compared for a scenario where an obstacle approaches on a collision course from the right at 10m/s. It can be seen that the effect of the future cost of current position term is to discourage positions directly ahead of an obstacle, therefore the new vehicle trajectory passes behind rather than in front of the obstacle. In order to provide the desired safe clearance distance the path in font of the obstacle requires a forward acceleration, whereas it can be seen that the path behind the obstacle requires a deceleration.

Adding the future cost of current positions also has a computational overhead which can be minimised by reducing the safe time margin that is considered. With the 2 second margin used in this example the impact on the iteration time is approx 0.2secs, as shown in Figure 107.

Several other scenarios were also tested, each providing the same behaviour. This result demonstrates that the desired rule-type behaviour of a preference to pass behind rather than ahead of an obstacle can be implemented via a simple addition to the proximity cost function.

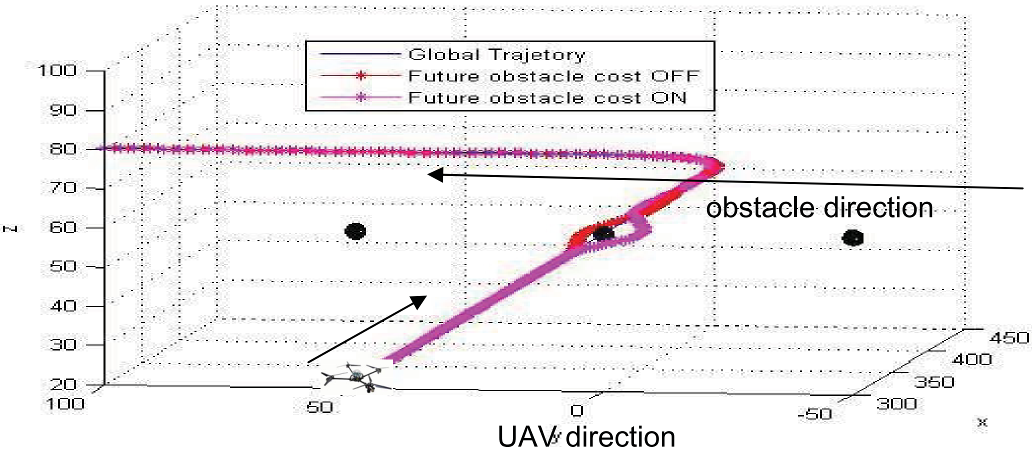


Figure 104 - Future obstacle cost bias

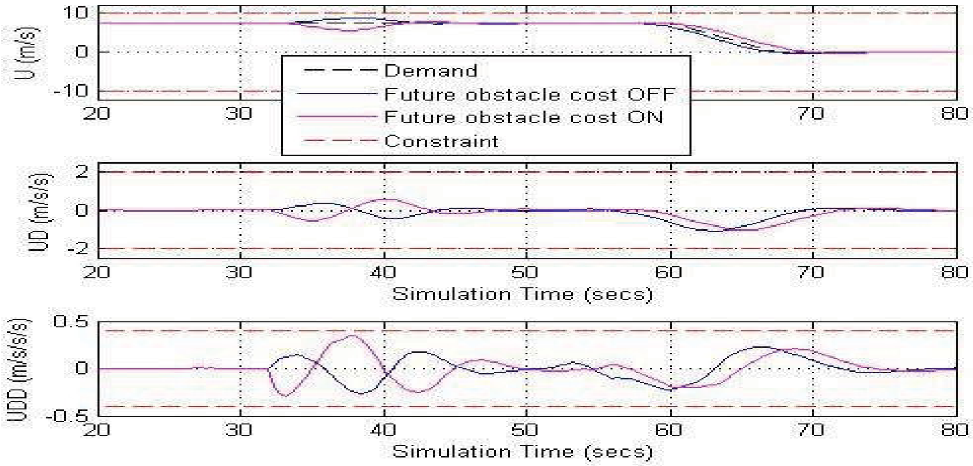


Figure 105 - Future obstacle cost bias (u-axis time history)

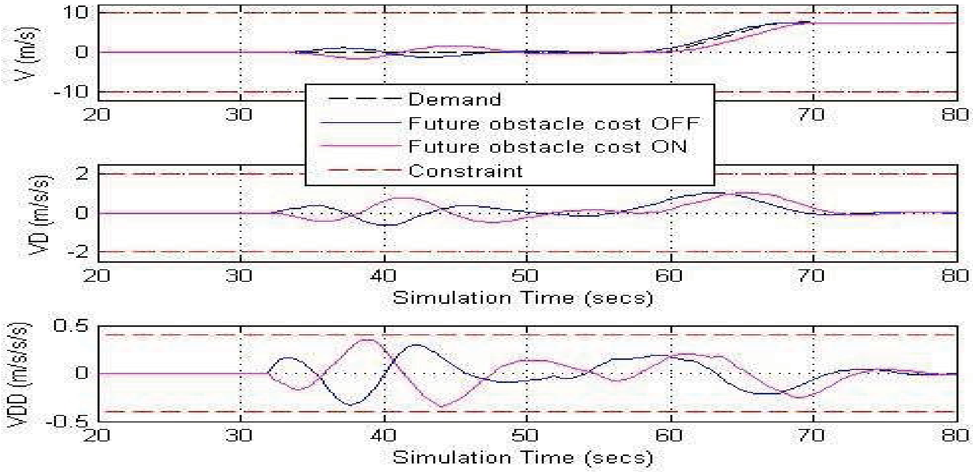


Figure 106 - Future obstacle cost bias (v-axis time history)

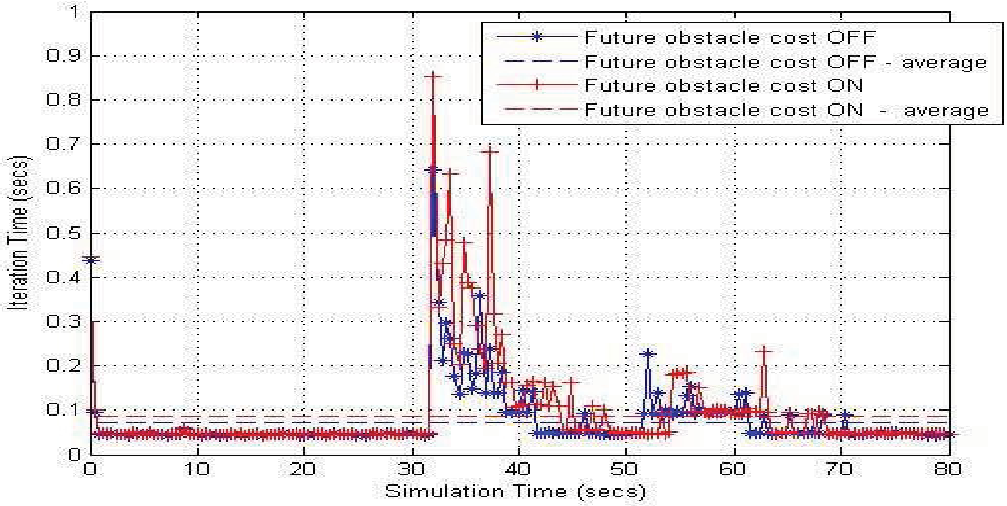


Figure 107 - Future obstacle cost (increased computational effort)

7.3.2 Velocity Vector Avoidance

An example of the impact of the velocity vector bias in the proximity cost function is shown in Figure 108 and Figure 109, where it can be seen that the vehicle is now directed away from the velocity vector of the obstacle. In this case the primary avoidance direction is changed from a left turn passing directly ahead of the obstacle to a descent passing underneath the obstacle. It is important to note that this behaviour contrasts with that provided by the 'future cost of current position' calculation presented in the previous section, where the cost function gradient strongly prefers to pass directly behind the obstacle, rather than above or below it. It can also be seen in Figure 110 that the this additional cost function bias again tends to increase the computation effort.

Several further scenarios were also tested, again resulting in similar behaviour. Overall it appears that the velocity vector bias term is a useful way of providing desirable behaviour, i.e. avoid the direct axis of motion. Additionally, this type of bias may also be used to increase clearance distances in different regions of the obstacle (i.e. greater clearance required on right than on left), although this has not yet been investigated.

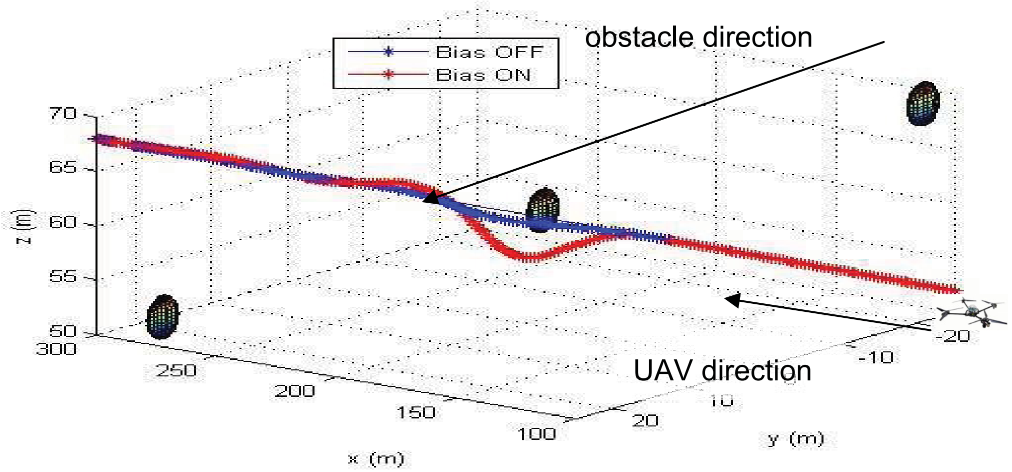


Figure 108 - Cross product proximity cost bias

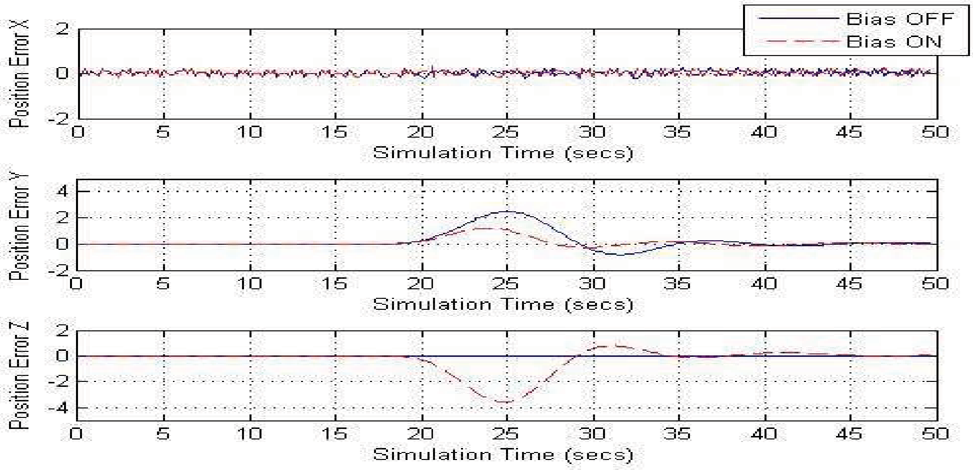


Figure 109 - Velocity vector proximity cost bias (global trajectory position errors)

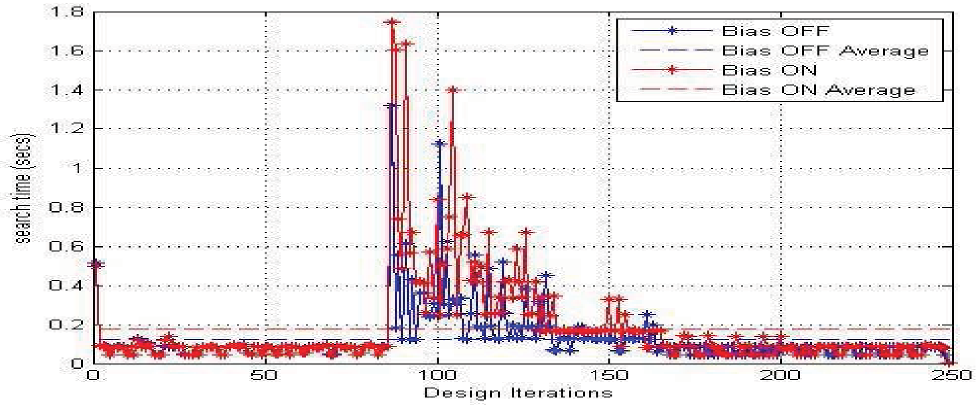


Figure 110 - Velocity vector proximity cost bias (increased computational effort)

7.3.3 *Cross Product Avoidance Manoeuvre*

The impact of the cross product avoidance bias is shown in two examples below, firstly for a single obstacle approaching from the right (Figure 111) and secondly for a multiple obstacle scenario (Figure 112). Both figures compare the results with and without the cross product bias. It can be seen in Figure 111 for the single obstacle scenario that the impact of the bias is to drive the UAV out of the horizontal plane, rather than passing ahead or behind the obstacle. This in itself can be a useful property, as it can be seen in most of the previously presented scenarios (even with multiple obstacles) that the UAV tended to remain in the horizontal plane if the vehicle and obstacles shared the same altitude. The multi-obstacle scenario shown in Figure 112 demonstrates that for complex scenarios the behaviour reverts back to a basic proximity driven cost, therefore demonstrating that the cross product bias can be implemented without degrading multi-obstacle performance.

Importantly, these examples demonstrate that the cross product bias does not need to be implemented in both vehicles to be successful, therefore helping to account for scenarios where the other vehicle is either not aware of the potential collision, or is aware but will not follow the cross product avoidance rule.

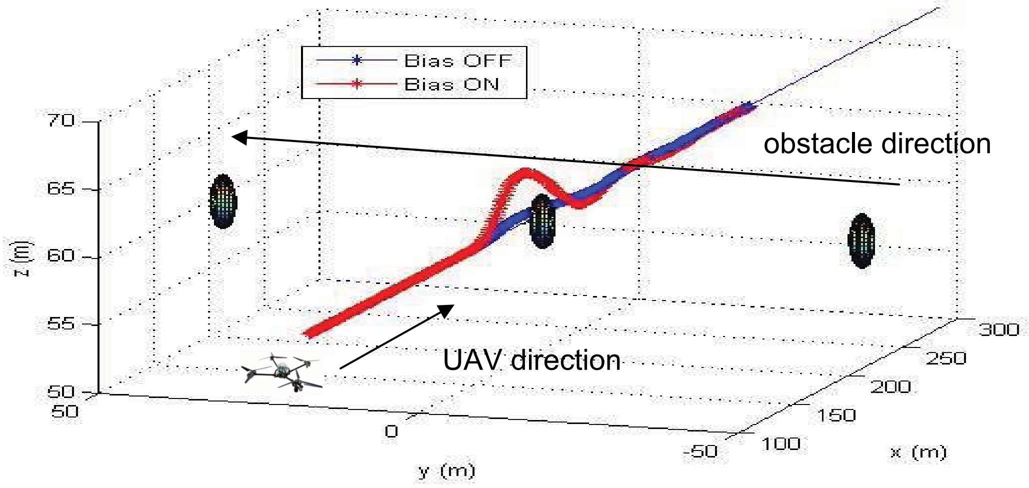


Figure 111 - Cross product avoidance direction example-1

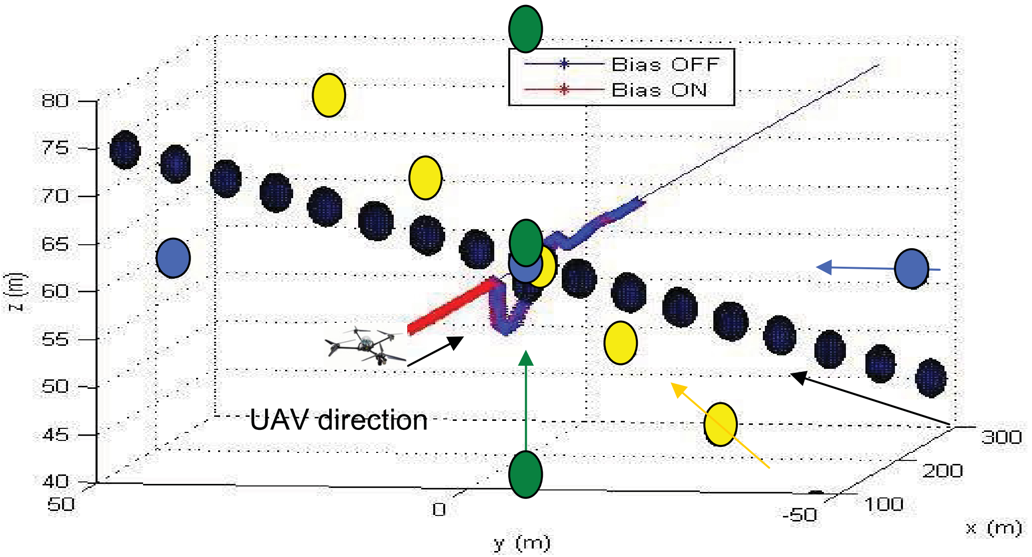


Figure 112 - Cross product avoidance direction example-2

7.4 Multiple Vehicle Deconfliction

Examples of multiple vehicle decentralised deconfliction are shown in Figure 113 to Figure 116, where each vehicle is shown following a straight line global trajectory that conflicts in the centre of the plot. These scenarios were designed to challenge the local motion planning framework, with the complexity of the confliction increasing from two vehicles up to eight vehicles all colliding at a single position. All vehicles are initially travelling at 6m/s, with a sensor horizon of 100m. In these test cases the only addition to the basic local motion planning framework used for each vehicle was the cross product avoidance rules.

It can be seen that even for the eight vehicle convergence case all collisions are avoided with each vehicle smoothly returning back to their global trajectory after the conflict has been resolved. The behaviour can be seen to be well ordered, with deviations from the global trajectory kept suitably small, and is similar to what could be expected from a single centralised planner. The avoidance manoeuvres utilise motion in all three axis as well as accelerating or decelerating in order to provide the desired balance between global position errors and proximity to other vehicles. An animation illustrating the four vehicle convergence scenario is provided in the file ‘animation8 - four vehicle convergence’.

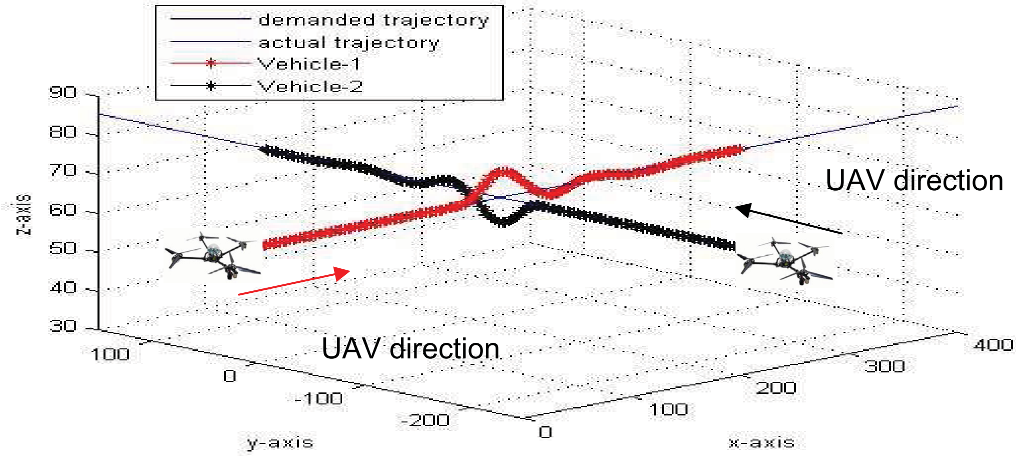


Figure 113 - Two vehicle convergence (cross product ON)

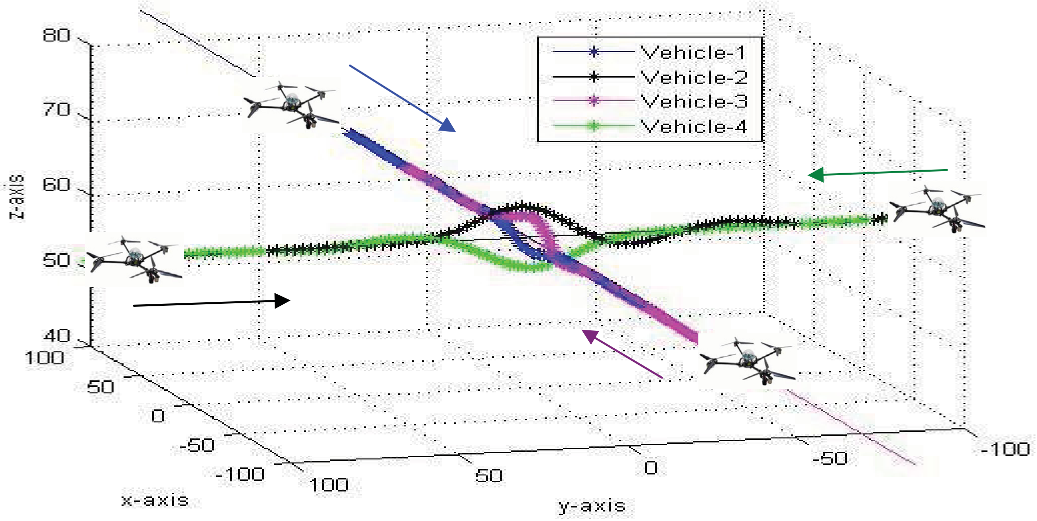


Figure 114 - Four vehicle convergence (cross product ON)

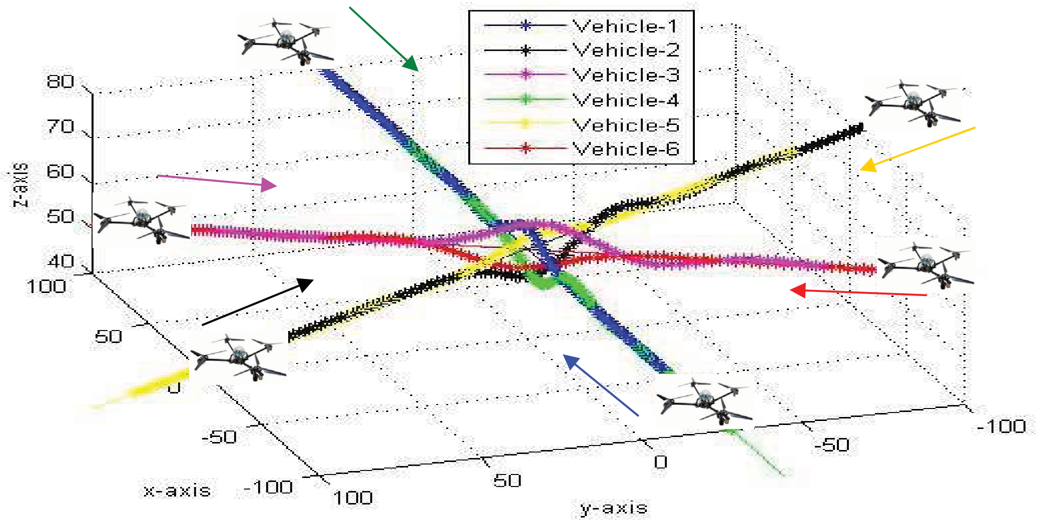


Figure 115 - Six vehicle convergence (cross product ON)

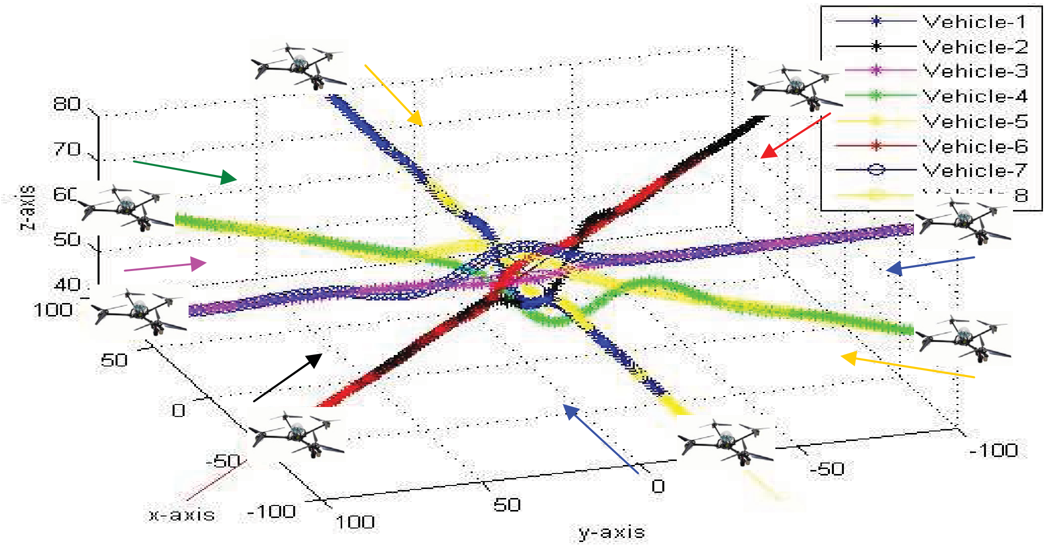


Figure 116 - Eight vehicle convergence (cross product ON)

The impact of turning off the cross product avoidance rules can be seen in Figure 117 and Figure 118 which shown the results of the four and six vehicle converge tests. It can be seen that without the cross product rules all vehicles now choose to manoeuvre in the same direction (DOWN for four vehicle scenario and UP for six vehicle scenario), which extends the conflict and reduces the time available for it to be resolved. In the four vehicle scenario all collisions are still avoided, although the clearance distance is significantly reduced, and the behaviour can be seen to be less efficient with much larger positions errors from the global trajectory. An animation illustrating this degraded performance is provided in the file 'animation9 - four vehicle convergence CP OFF' where the poorer performance is clear. For the six vehicle scenario disabling the cross product avoidance manoeuvres results in a collision between vehicle-1 & vehicle-3.

In the above example all vehicles are at the same altitude, therefore resulting in positional symmetry in the situation faced by individual vehicles. It is this symmetry that the cross product avoidance rules break, ensuring that vehicles always manoeuvre away from each other. For more realistic scenarios where this positional symmetry is less likely (i.e. due to small differences in altitude) the cross product rules are likely to be overwhelmed by the proximity cost. As previously discussed the focus of the deconfliction provided here is in the immediate term, therefore defaulting to a proximity based avoidance direction is likely to be acceptable.

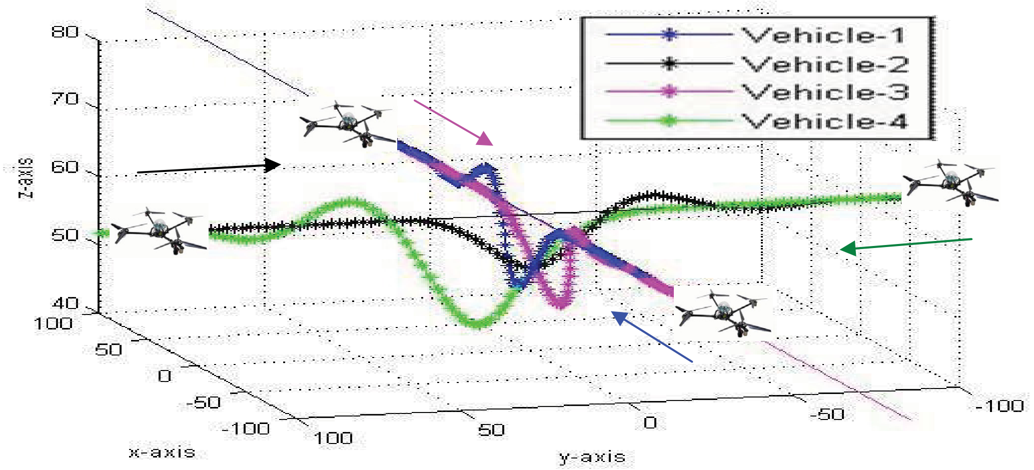


Figure 117 - Four vehicle convergence (cross product OFF)

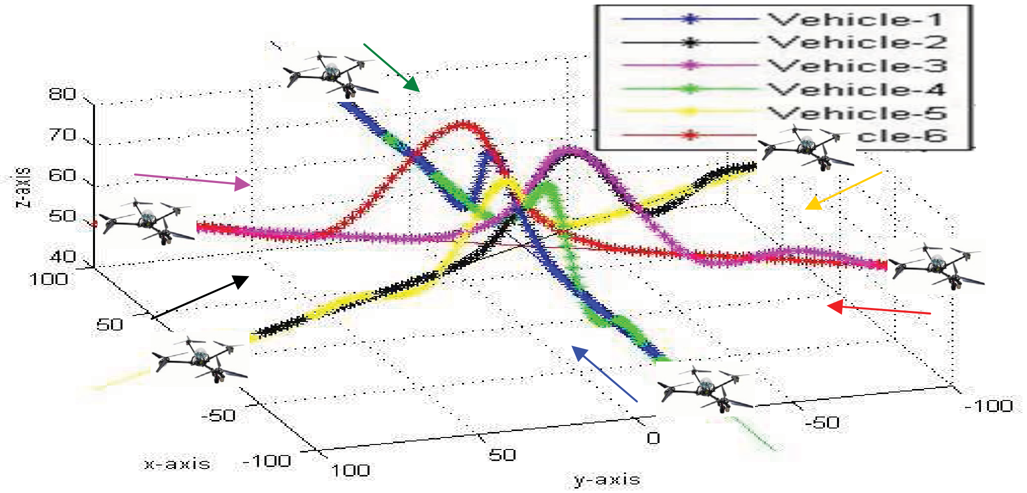


Figure 118 - Six vehicle convergence (cross product OFF)

The primary issue that arises during multiple vehicle deconfliction scenarios is errors in the predicted positions of other vehicles across the design horizon. Basing these predictions purely on current vehicle state can be highly inaccurate as shown in **Figure 119** which shows both current receding horizon trajectory intentions and the predictions made by the other vehicle for a two vehicle a head-on conflict. It can be seen that each vehicle designs a trajectory that is well clear of the predicted path of the other vehicle, but given the prediction errors this does not necessarily resolve the conflict.

For the majority of scenarios this error can be handled by the cross product avoidance rules directing vehicles away from each other, i.e. in Figure 119 one vehicle would be directed upwards and the other downwards. However, for complex situations the overall performance may be improved by removing the error in prediction of intent. This can be done by transmitting receding horizon trajectory intentions. An example of the impact of this is shown in Figure 120 and Figure 121 which show the results of an eight vehicle convergence test with the manoeuvre direction limited by an obstacle two metres below the global trajectory. It can be seen that as expected the cooperative approach results in overall vehicle trajectories that are more organised and efficient.

The inter-vehicle and vehicle to obstacle clearance distances are shown in Figure 122 and Figure 123 for cooperative case, where it can be seen that all collisions are avoided, and the evasive manoeuvres do not significantly reduce the clearance distance to the static obstacle.

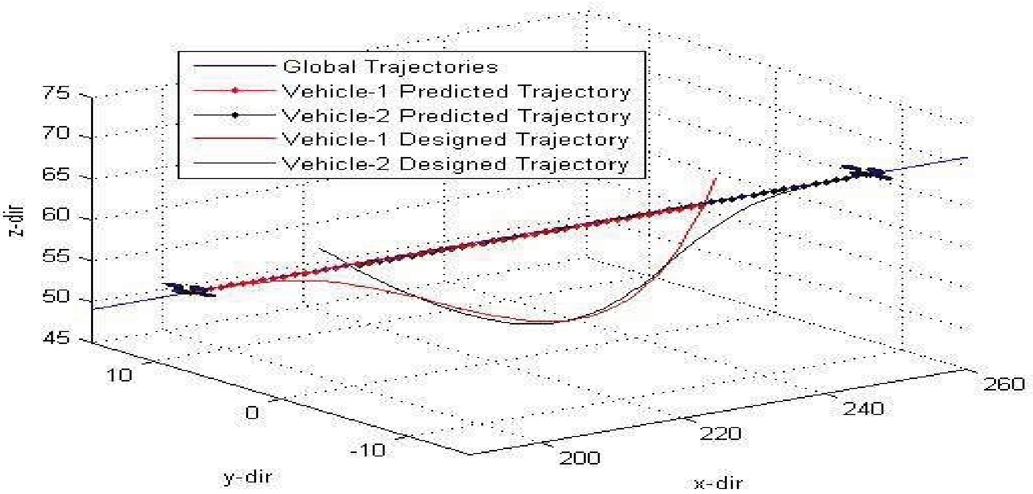


Figure 119 - Errors in predicted vehicle positions, example

In the absence of explicit negotiation between each vehicle the receding horizon acts as a buffer, allowing a 'virtual negotiation' to occur. This virtual negotiation is based on either predicted vehicle intent for the non-cooperative case, or transmitted intent for the cooperative case, and continues until all vehicles are satisfied that all predicted conflicts are removed. For the uncooperative case the period of negotiation is likely to be greater, due to the potential for errors in predicting the actual intent of the other vehicles. However, for almost all tested scenarios the ten second receding horizon length was sufficient to avoid a collision. It will typically be the case that explicit negotiation will resolve a conflict faster than this virtual negotiation. However, explicit negotiation requires a more complex system, which has not been investigated within this thesis.

This period of virtual negotiation can clearly be seen in the file 'animation 10 - eight vehicle convergence' which shows the cooperative eight vehicle convergence scenario. This simulation is unrealistic in that the implementation results in all vehicles planning trajectories at exactly the same time, i.e. they are perfectly synchronised. This results in each vehicle planning for the required clearance distance, which due to both vehicles aiming for the same clearance results in twice the necessary distance provided. Both vehicles then re-plan to reduce the clearance resulting in the opposite effect, i.e. less than the desired clearance. This effect continues for a few design cycles before a compromise is obtained. This effect can be removed by de-synchronising the planning of each vehicle, as shown in the file 'animation 11 - eight vehicle convergence desynchronised' where the initial period of virtual negotiation is almost completely removed. In both cases all collisions are successfully avoided while also respecting each vehicle's performance limits. An increased horizon length can therefore be seen to provide grater time for this negotiation, therefore helping to compensate for poor position prediction or initial manoeuvring in the same direction.

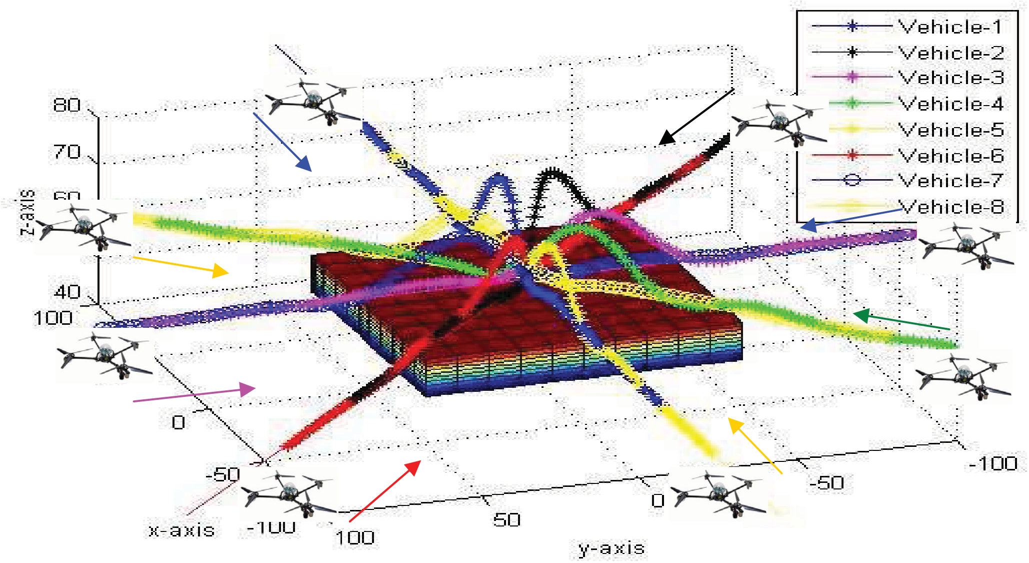


Figure 120 - Eight vehicle convergence with obstacle (communication OFF)

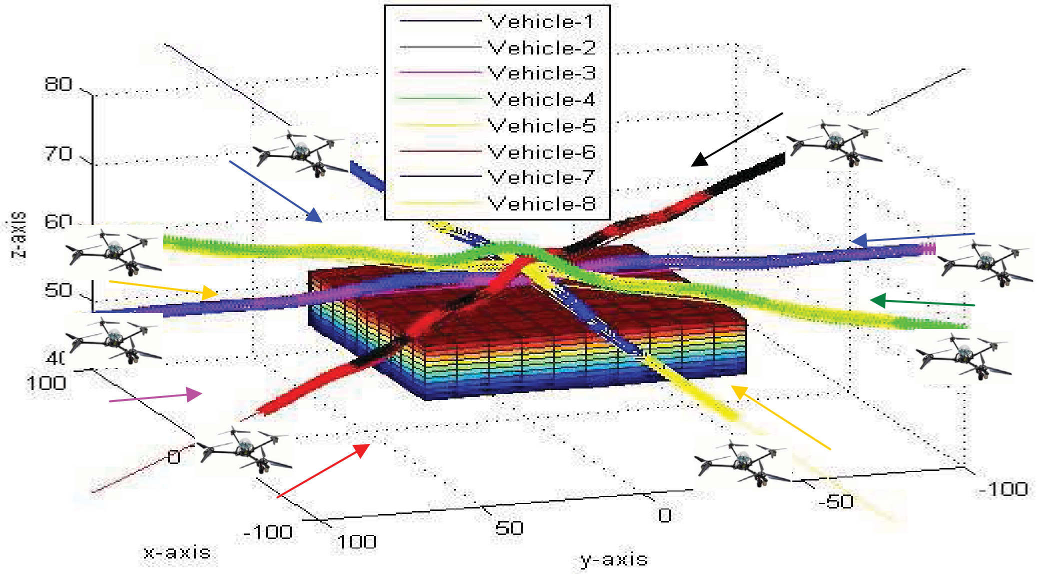


Figure 121 - Eight vehicle convergence with obstacle (communication ON)

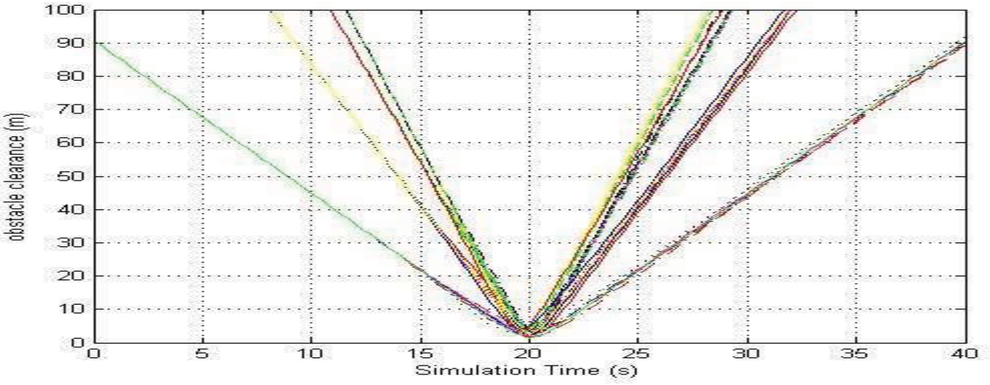


Figure 122 - Clearance distance between all eight vehicles

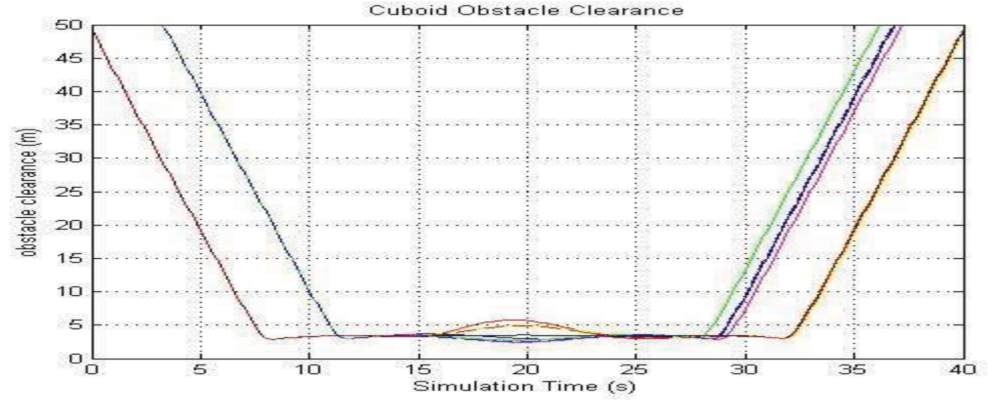


Figure 123 - Clearance distance between each vehicle and the cuboid obstacle

7.5 Formations of Unmanned Vehicles

In scenarios where multiple vehicles are operated by the same system, certain mission phases may require formation flight behaviour, i.e. ingress, egress, etc. In these scenarios it is important to ensure that each vehicle is able to take evasive action from an incoming object without resulting in a collision with another vehicle in the same formation. The ability of the local motion planning framework to provide this behaviour is demonstrated in this section.

A typical example of behaviour that can be provided is shown in Figure 124 to Figure 126 in a scenario where a four vehicle formation encounters a dynamic obstacle on a head-on collision course. Each vehicle is travelling at 4m/s, the obstacle is travelling at 5m/s and the formation is defined with each vehicle offset 3m from it's neighbour in the x and y axis. It can be seen that all four vehicles successfully avoid the obstacle, while also choosing avoidance directions that do not conflict with other vehicles in the formation. Several other scenarios were also tested, including larger formations and multiple vehicles, resulting in similar behaviour.

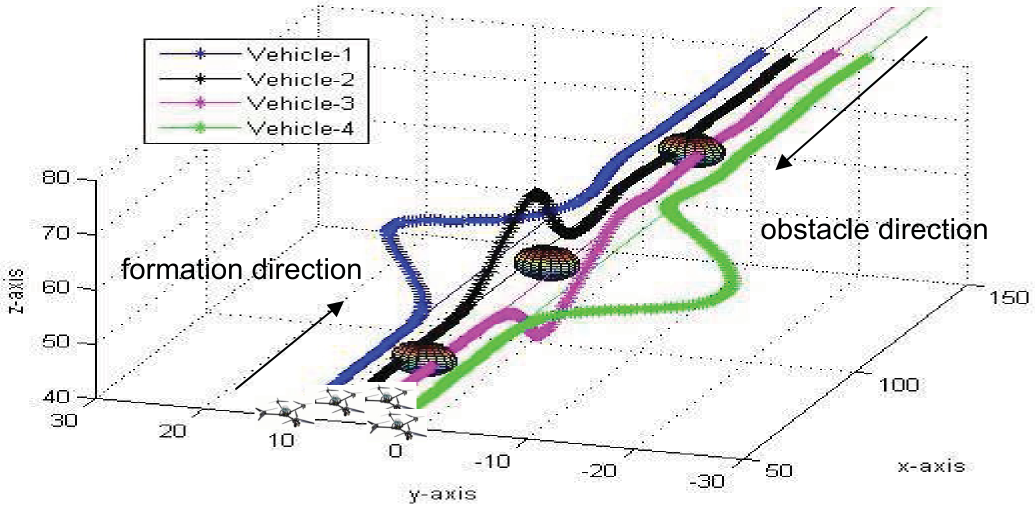


Figure 124 - Four vehicle formation (absolute formation)

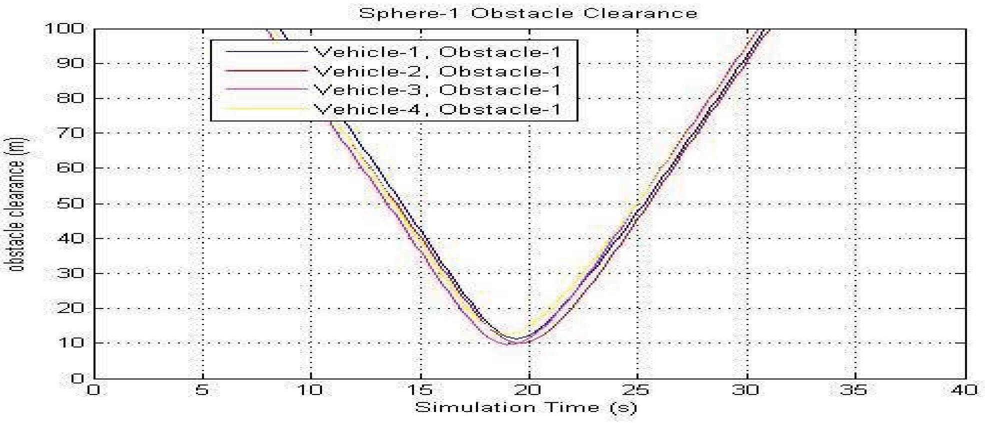


Figure 125 - Four vehicle formation, obstacle clearance

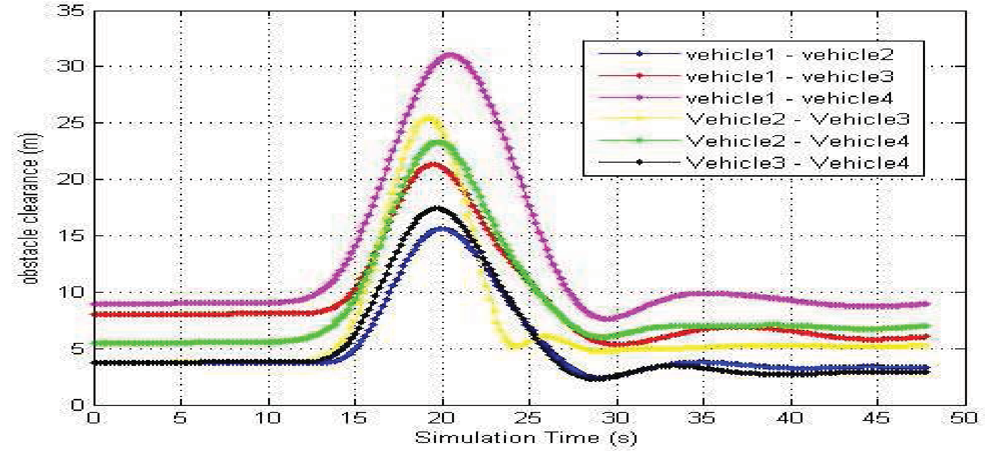


Figure 126 - Four vehicle formation, vehicle to vehicle clearance

It is also possible to vary the type of behaviour provided by altering the definition of the formation. For example, in Figure 124 the formation is defined by aligning the global trajectories for all four vehicles relative to each other. Each vehicle therefore independently follows it's own global trajectory, while considering the presence of the other vehicles in the same manner to any other obstacles. While this approach typically provides good performance, it has a potential drawback in that there is no coordination between the vehicles during formation split and formation re-acquire manoeuvres. This lack of coordination may be problematic in tight formations if any overshoot is present in the trajectory acquire behaviour, e.g. when two vehicles reacquire trajectories from either side if they both overshoot inter-vehicle clearance may be significantly reduced. Although the situation awareness of the trajectory tracking approach is likely to handle this issue, improved performance may be obtained by increasing the coordination between vehicles in the same formation.

This issue can be addressed by employing a cooperative approach, where the formation is defined relative to a leader vehicle (or neighbouring vehicles), rather than via individual trajectories. This cooperative assumption is not unreasonable for vehicles that are part of the same system, and the Bezier polynomial trajectory specification provides a highly compact description of the receding horizon trajectory[38], therefore allowing efficient transmission of intent. Using this approach only the leader vehicle follows a global trajectory, and the other vehicles calculate current target trajectories by apply a formation specific offset either to the receding horizon trajectory of the leader, or to that of their neighbouring vehicles.

An example of the impact of this relative formation definition is shown in Figure 127 where vehicle-3 is defined as the leader. It can be seen that the vehicles now tend to behave in a more coordinated manner, all choosing the same formation split and formation re-acquire manoeuvres. This clearly contrasts with the absolute formation behaviour shown in Figure 124 where each vehicle chooses a different avoidance direction.

Animations of both absolute and relative formation flight behaviour are provided in the files 'animation 12 - absolute formation' and 'animation 13 - relative formation'.

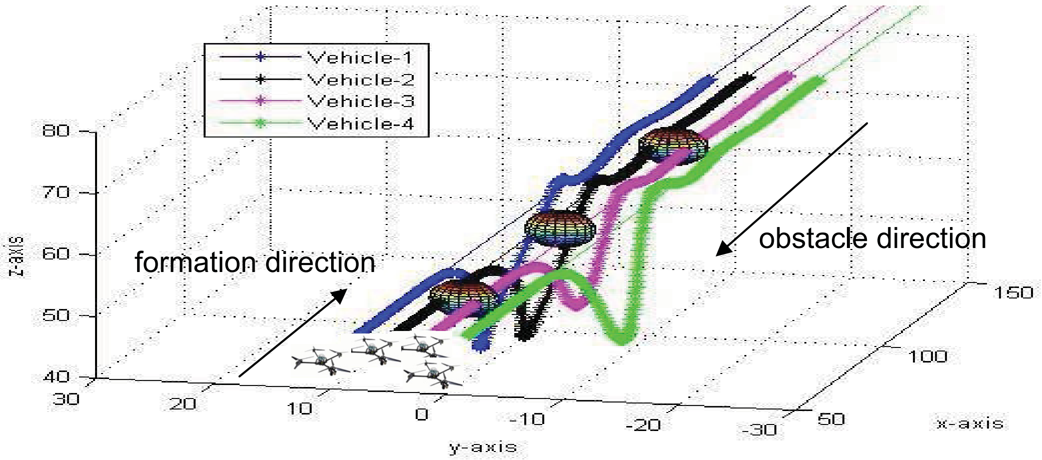


Figure 127 - Four vehicle formation (relative formation)

[38] See Section 5.3 for full definition of the receding horizon trajectories.

7.6 Multiple Formations of Unmanned Vehicles

In order to further challenge the ability of the local motion planning framework tests were also conducted with multiple formations of multiple unmanned vehicles. Again the deconfliction performance was found to be good, with the cross product avoidance rules again leading to well ordered efficient manoeuvres. This can be seen in Figure 128 where two formations of unmanned vehicles approach each other side on at the same altitude. It can be seen the resulting behaviour is similar to that shown in Figure 111, with both formations following the appropriate rule based manoeuvre direction. The conflict therefore results in minimal disruption to the paths of both sets of unmanned vehicles. With the cross product manoeuvres disabled the local motion planners still avoid any collisions, although the overall behaviour is now less well ordered as shown in Figure 129. As with the single vehicle convergence tests the cross product rules only dominate if there is no clear proximity based direction, helping to resolve symmetry in the scenario faced by each vehicle.

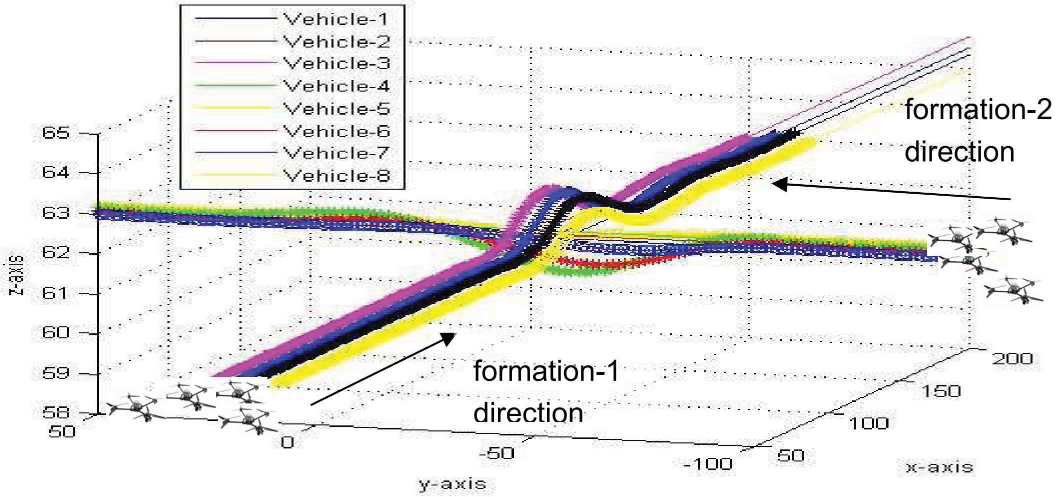


Figure 128 - Two formation conflict, cross product manoeuvre ON

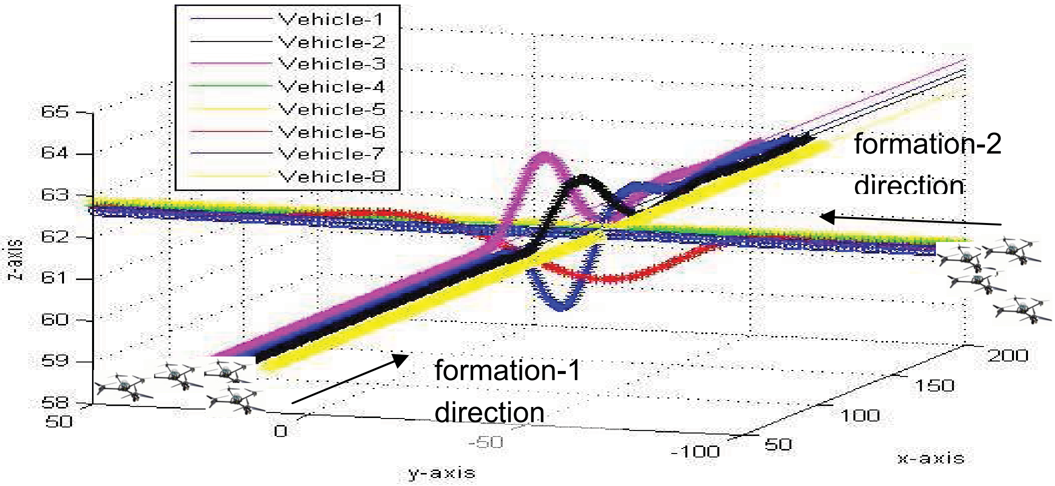


Figure 129 - Two formation conflict, cross product manoeuvre OFF

An animation of the two formation conflict shown in Figure 128 is provided in the file 'animation 14 - two formation conflict'.

Several more complex scenarios were also tested, including the one shown in Figure 130 where a conflict arises between three formations of unmanned vehicles in the presence of a large static surface obstacle three metres below the global trajectories and two dynamic obstacles, both at the same altitude as all the unmanned vehicles. The speed of the unmanned vehicles and obstacles all ranged between 4m/s and 8m/s. This scenario was designed to be challenging, with all conflicts occurring at approximately the same location and time. The behaviour displayed in Figure 130 is representative of typical system performance, with all collisions avoided and each vehicle smoothly re-joining it's formation after the required evasive manoeuvres. This scenario provides a good example of a situation where a human pilot would struggle with the simultaneous deconfliction of multiple obstacles and vehicles, but the receding horizon planning approach can be seen to cope.

An animation displaying this multi-vehicle scenario is provided in 'animation 15 - multiple formations and obstacles' where the complex nature of the scenario is clear. The unsteady path of obstacle-2 can also be seen, which further complicates the scenario by ensuring that predictions regarding it's position over the design horizon are rarely accurate. This example also emphasises the need for the continual receding horizon design as a way of handling uncertainty over the future positions of other vehicles and obstacles, and it can be seen that the receding horizon trajectory from each vehicle varies significantly during the conflict.

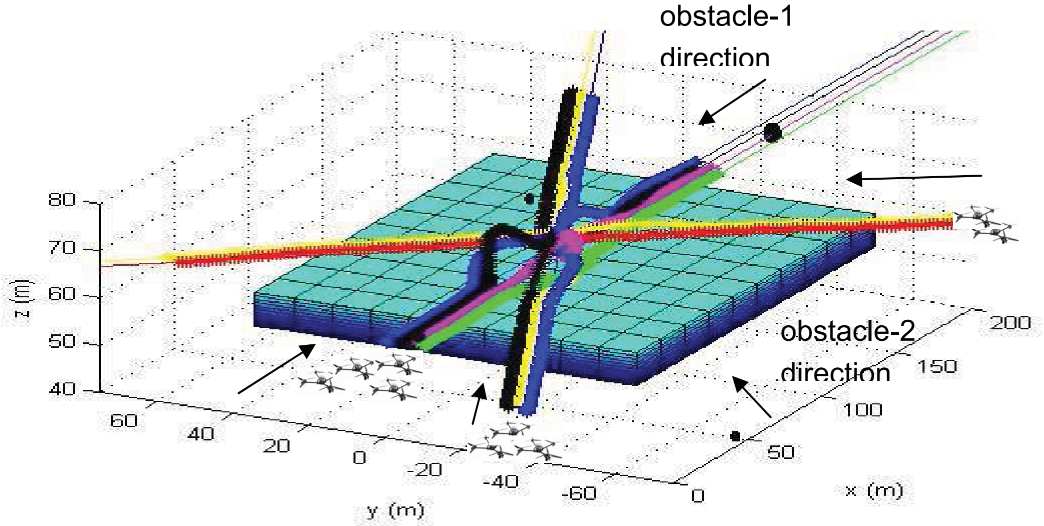


Figure 130 - Three formation scenario, with static & dynamic obstacles

During system testing it was found that collisions did occasionally occur, but that they tended to be due to either i) poor prediction of future behaviour of vehicles, or ii) required manoeuvres exceeding the performance limits of the vehicle (i.e. due to very fast obstacles). It should be noted that the behaviour presented within this chapter assumes perfect tracking of the output space receding horizon trajectories (as with Chapter 6), as well as perfect obstacle detection and tracking data. The results shown are considered to be a good starting position from which to increase the system realism.

8 Tracking Continuously Designed Local Motion Trajectories - *Control Space Component*

8.1 Overview

A key element of the research presented within this thesis is the division of the defined local motion planning and control problem into distinct output and control space elements. The output space component of this problem was considered to be the key element of this research, hence it was the focus of Chapters 5 to 7, where perfect tracking of the continually designed receding horizon trajectories was assumed. The aim of this chapter is to address the control space element of the overall problem, removing the assumption of perfect tracking.

The simulation model used for this work was the nonlinear 6 DOF quadrotor model introduced in Section 5.2. The goal of the control space component is to calculate control signals that allow the quadrotor vehicle to track the receding horizon trajectories designed by the output space component. It is important to note that this trajectory tracking problem differs from the usual one in that the receding horizon trajectory being tracked is updated at a high rate (i.e. 5Hz), with each update zeroing existing tracking errors. This creates a dynamic interaction between the trajectory design and trajectory tracking layers, which must be correctly handled in order to make the output / control space division feasible. Although this interaction complicates tracking problem it does result in a key advantage that the vehicle is always on trajectory therefore the RH trajectory tracking layer isn't concerned with trajectory acquire performance. With traditional trajectory tracking approaches such as 'pure pursuit', performance is typically a compromise between tracking and acquire performance, potentially requiring switching between different modes. Focusing only on trajectory tracking simplifies the problem, allowing higher gain control without degrading acquire performance.

The intention of this chapter is not to present a detailed control space implementation for the quadrotor vehicle. Rather, the goal is proof of concept, demonstrating that the continually designed receding horizon trajectories can be accurately tracked, therefore not significantly impacting on the simulation performance presented previously. Control space implementation is therefore restricted to the horizontal plane, with the assumption that if the x-y axes can be successfully controlled then the same will be true of the vertical axis.

8.2 Control Space Implementation

In order to track the receding horizon trajectories the following control modes were implemented on the SIMULINK model of the quadrotor vehicle.

- *Forward speed control*
- *Lateral speed control*

- *Altitude hold*
- *Heading hold*

Each of the implemented control modes are based on the overall control strategy outlined in Section 5.2, and is discussed further below.

8.2.1 Forward Speed Control

Forward speed control was provided via a multi-loop proportional plus differential (PD) controller with a feed-forward target for acceleration. Target values of both forward speed and acceleration were derived from the current receding horizon trajectories using a *pursuit* type approach with a one second anticipation time to match the vehicle dynamic response. The resulting control loops are shown in equations 8-1 to 8-4:

$$\dot{u}_{dem}(t) = \dot{u}_{target}(t) + k_p^u \times u_{err}(t) + k_d^u \times \dot{u}_{err}(t) \tag{8-1}$$

$$\theta_{dem}(t) = k_p^{\dot{u}} \times (\dot{u}_{dem}(t) - \dot{u}_{actual}(t)) \tag{8-2}$$

$$q_{dem}(t) = k_p^{\theta} \times (\theta_{dem}(t) - \theta_{actual}(t)) \tag{8-3}$$

$$R_1^u = R_3^u = k_p^q \times (q_{dem}(t) - q_{actual}(t)) \tag{8-4}$$

where:

$u_{err}(t) = u_{target}(t) - u_{actual}(t)$

$u_{t\arg et}(t) = u_{RH}(t_{current} + t_{anticipation})$

u_{RH} *= forward speed receding horizon trajectory*

$\dot{u}_{err}(t) = \frac{d}{dt}(u_{err}(t))$

$\dot{u}_{target}(t) = \dot{u}_{RH}(t_{current} + t_{anticipation})$

$\dot{u}_{RH}$ *is the forward acceleration receding horizon trajectory*

θ *= vehicle pitch angle*

q *= vehicle pitch rate*

R_n *= demanded rpm of the nth rotor*

8.2.2 Lateral Speed Control

Similarly, v-axis control was provided as follows:

$$\dot{v}_{dem}(t) = \dot{v}_{target}(t) + k_p^v \times v_{err}(t) + k_d^v \times \dot{v}_{err}(t) \tag{8-5}$$

$$\phi_{dem}(t) = k_p^{\dot{v}} \times (\dot{v}_{dem}(t) - \dot{v}_{actual}(t)) \tag{8-6}$$

$$p_{dem}(t) = k_p^{\phi} \times (\phi_{dem}(t) - \phi_{actual}(t)) \tag{8-7}$$

$$R_2^v = R_4^v = k_p^p \times (p_{dem}(t) - p_{actual}(t)) \tag{8-8}$$

where:

$v_{err}(t) = v_{target}(t) - v_{actual}(t)$

$v_{t\arg et}(t) = v_{RH}(t_{current} + t_{anticipation})$

v_{RH} *= forward speed receding horizon trajectory*

$\dot{v}_{err}(t) = \frac{d}{dt}(v_{err}(t))$

$\dot{v}_{target}(t) = \dot{v}_{RH}(t_{current} + t_{anticipation})$

$\dot{v}_{RH}$ *= forward acceleration receding horizon trajectory*

ϕ *= vehicle roll angle*

p *= vehicle roll rate*

8.2.3 Altitude Hold

In order to provide basic height hold behaviour the following z-axis control loops were used:

$$w_{dem}(t) = k_p^z \times (z_{dem}(t) - z_{actual}(t)) \tag{8-9}$$

$$-R_1^z = R_2^z = -R_3^z = R_4^z = k_p^w \times (w_{dem}(t) - w_{actual}(t)) \tag{8-10}$$

Additionally, a feed-forward attitude / height compensation loop was added to decouple u-v control from altitude:

$$R_1^{attcomp} = -R_2^{attcomp} = R_3^{attcomp} = -R_4^{attcomp} = \frac{R_{hover}}{\cos\phi \times \cos\theta} \tag{8-11}$$

Where

R_{hover} *is the rpm required to maintain altitude at the current value.*

8.2.4 Heading Hold

In order to maintain the vehicle body axis with the earth axis (to allow the u-v control loop) a heading angle hold loop was also added as follows:

$$r_{dem}(t) = k_p^r \times \varphi_{err}(t) + k_d^r \times \dot{\varphi}_{err}(t) \tag{8-12}$$

$$R_1^{\varphi} = R_2^{\varphi} = R_3^{\varphi} = R_4^{\varphi} = k_p^r \times (r_{dem}(t) - r_{actual}(t)) \tag{8-13}$$

where:

$$\varphi_{err}(t) = \varphi_{target}(t) - \varphi_{actual}(t)$$

$$\varphi_{target}(t) = 0$$

8.2.5 Combined Implementation

The final control signals were then given by the summation of the individual terms:

$$\begin{aligned}
R_1^{total} &= R_1^u + R_1^z + R_1^{attcomp} + R_1^{\varphi} \\
R_2^{total} &= R_2^u + R_2^z + R_2^{attcomp} + R_2^{\varphi} \\
R_3^{total} &= R_3^u + R_3^z + R_3^{attcomp} + R_3^{\varphi} \\
R_4^{total} &= R_4^u + R_4^z + R_4^{attcomp} + R_4^{\varphi}
\end{aligned} \qquad (8\text{-}14)$$

In addition to the above control laws, the saturation limits shown in Table 3 were also imposed in order to prevent large errors within the controllers from resulting in poor behaviour. After some manual tuning the control law gains were set as shown in Table 4 and Table 5.

	Speed (m/s)	*Acceleration (m/s/s)*	*Rate (deg/s)*	*Attitude (deg)*
u-axis control	$-15 \le u_{dem} \le 15$	$-15 \le \dot{u}_{dem} \le 15$	$-2 \le q_{dem} \le 2$	$-10 \le \theta_{dem} \le 10$
v-axis control	$-15 \le v_{dem} \le 15$	$-15 \le \dot{v}_{dem} \le 15$	$-2 \le p_{dem} \le 2$	$-10 \le \phi_{dem} \le 10$
altitude hold	$-10 \le w_{dem} \le 10$			
heading hold			$-90 \le r_{dem} \le 90$	

Table 3 - Saturation limits in control space implementation

	u_{dem} (v_{dem})	$\dot{u}_{dem}$ $(\dot{v}_{dem})$	θ_{dem} (ϕ_{dem})	r_{dem} (p_{dem})
k_p	1	0.1	0.5	10
k_d	0.4	0	0	0

Table 4 - u-axis (&v-axis) control gains

	z_{dem}	w_{dem}	φ_{dem}	r_{dem}
k_p	1	0.1	2	30
k_d	0.4	0	6	0

Table 5 - Altitude & Heading hold control gains

8.3 Simulation Results - *Trajectory Tracking*

A typical example of the performance of the coupled output and control space layers is provided in Figure 131 to Figure 134, where the quadrotor vehicle is tracking the global trajectory shown in Figure 131. The resulting total position errors between the quadrotor and the *global* trajectory are compared in Figure 132, where it can be seen that total position errors are maintained at less than 0.5m. Tracking errors between the quadrotor vehicle and the receding horizon trajectories are shown in Figure 133 and Figure 134, where it can be seen that position errors are maintained at less than 0.005m. This is the critical result, as it demonstrates that the vehicle is always controlled to the receding horizon trajectories which are both aware of the local obstacle space and *continually* designed. It can also be seen in these figures that speed and acceleration errors are maintained at negligibly small values. The high frequency content seen in Figure 133 and Figure 134 matches the frequency of the receding horizon design rate, therefore is due to the zeroing of errors while imposing boundary conditions. This is illustrated by the close up image shown in **Figure 135**.

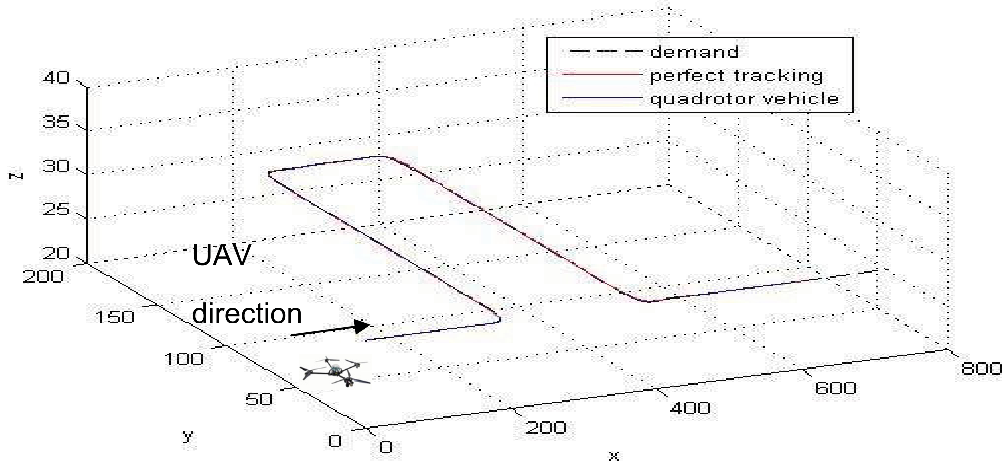


Figure 131 - Control space example-1, comparison of quadrotor vehicle result with perfect tracking

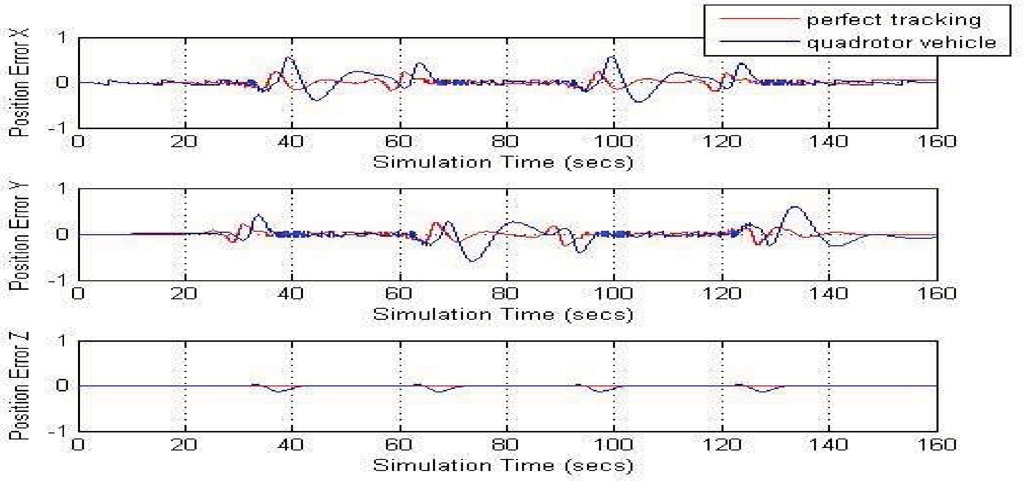


Figure 132 - Example-1, position errors between the vehicle and global trajectory

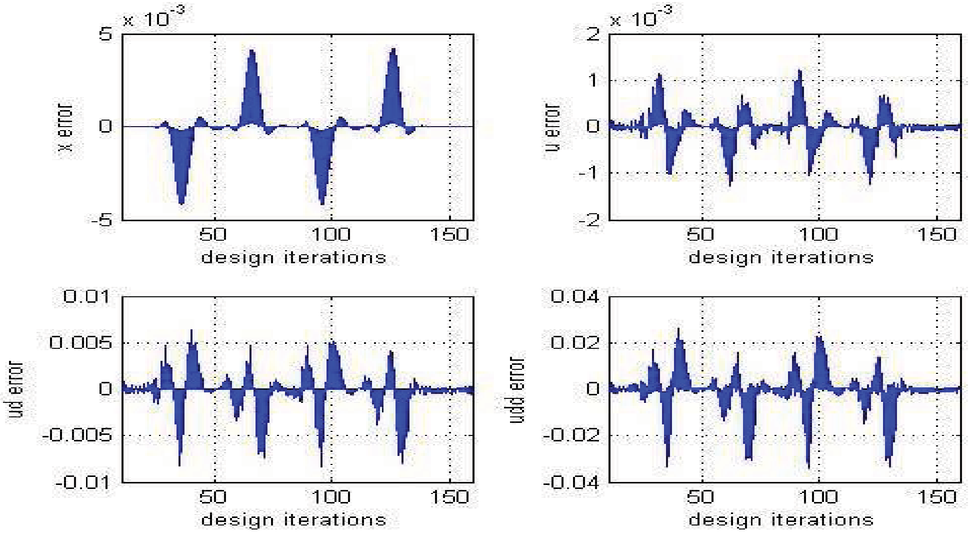


Figure 133 - Example-1, errors between the vehicle and RH trajectory (x-axis)

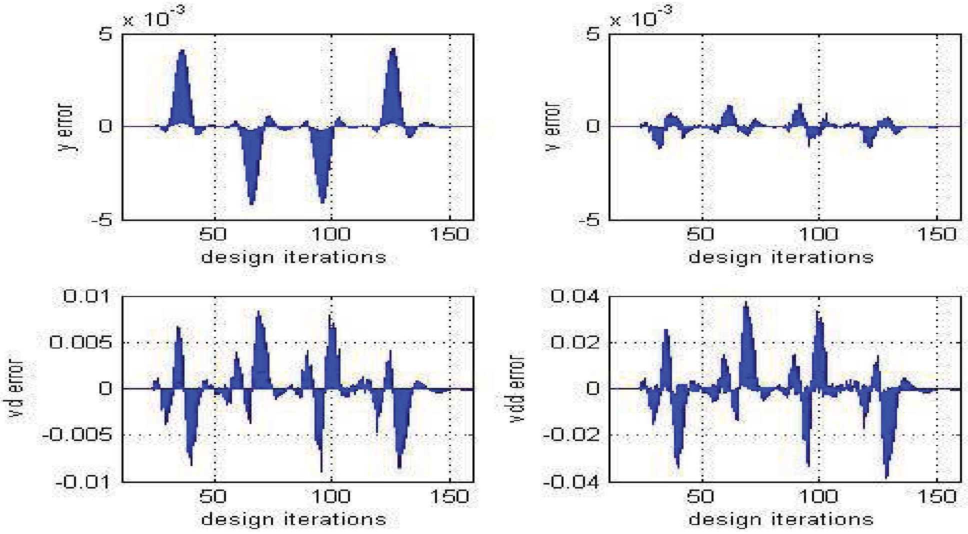


Figure 134 - Example-1, errors between the vehicle and RH trajectory (y-axis)

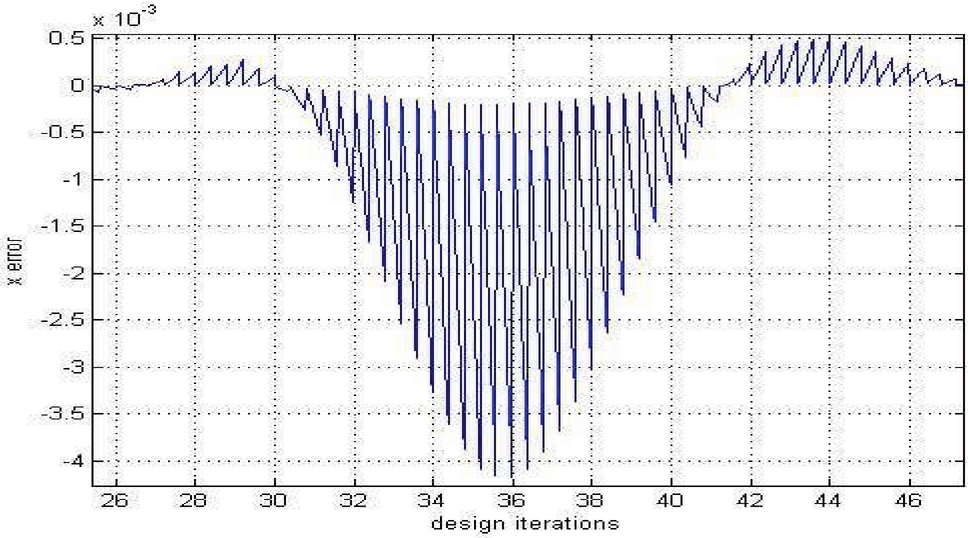


Figure 135 - Example-1, close up of figure 132 (top-left)

8.4 Simulation Results - *Dynamic Obstacle Avoidance*

A second example of the performance of the coupled output and control space layers, this time in the presence of a dynamic obstacle, is shown in Figure 136 to Figure 139. In this scenario the quadrotor is following a straight line global trajectory at 6m/s when a dynamic obstacle modelled as a sphere of radius 5m is detected on a head-on collision course at 10m/s. It can be see that the avoidance manoeuvre with the full nonlinear 6DOF quadrotor vehicle model is similar to the manoeuvre that results from perfect tracking of the receding horizon trajectories. The output space receding horizon design framework has no preference for avoidance direction, therefore the right turn (rather than left) is due to minor position differences when the obstacle is first detected. It can also be seen in Figure 137 that the obstacle clearance distance for the quadrotor vehicle is reduced from approx. 6m down to approx. 3.5m. This reduction appears to be due to minor tracking errors leading to differences in initial design states, which subsequently grows over the course of the simulation. If necessary, this safe clearance distance may be increased by either increasing the obstacle cost function, or by improved tracking performance. It can also be seen in Figure 138 and Figure 139 that tracking errors between the full quadrotor model and the receding horizon trajectories are again negligibly small, therefore providing the continuous situation aware vehicle guidance that is necessary for unmanned vehicle operation within complex obstacle rich-environments.

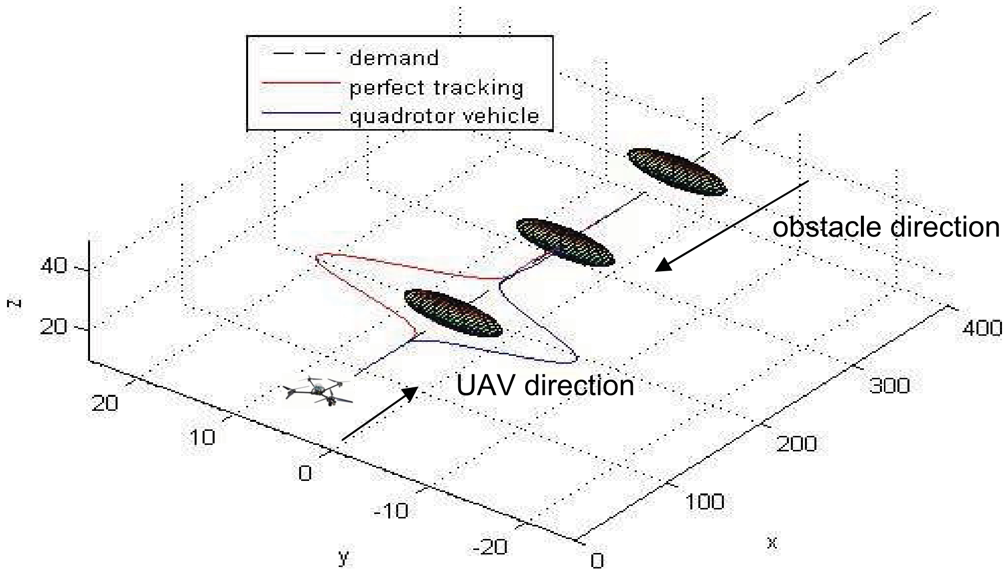


Figure 136 - Control space example-2, comparison of quadrotor vehicle result with perfect tracking

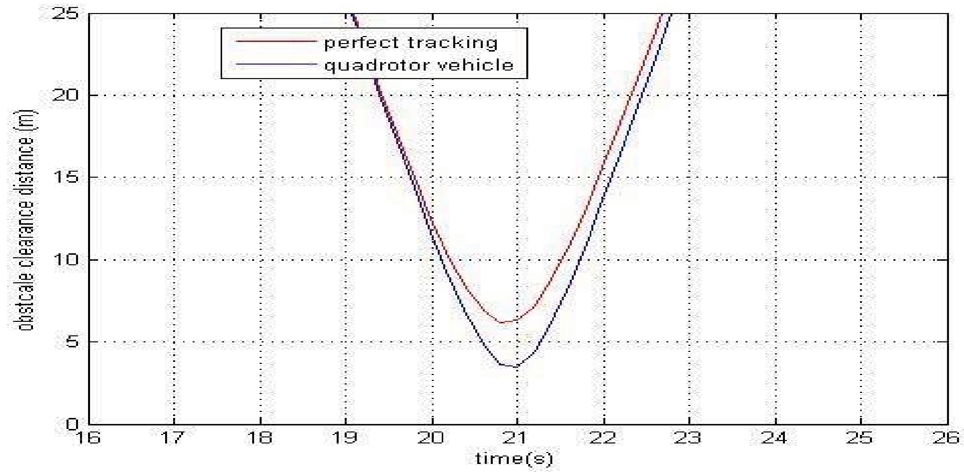


Figure 137 - Example-2, comparison of obstacle clearance distance

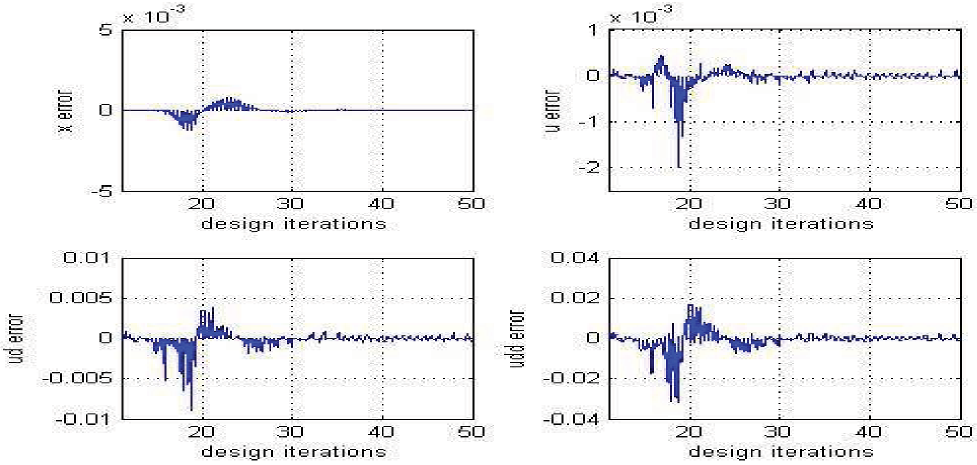


Figure 138 - Example-2, x-axis tracking errors between the vehicle and the RH trajectory

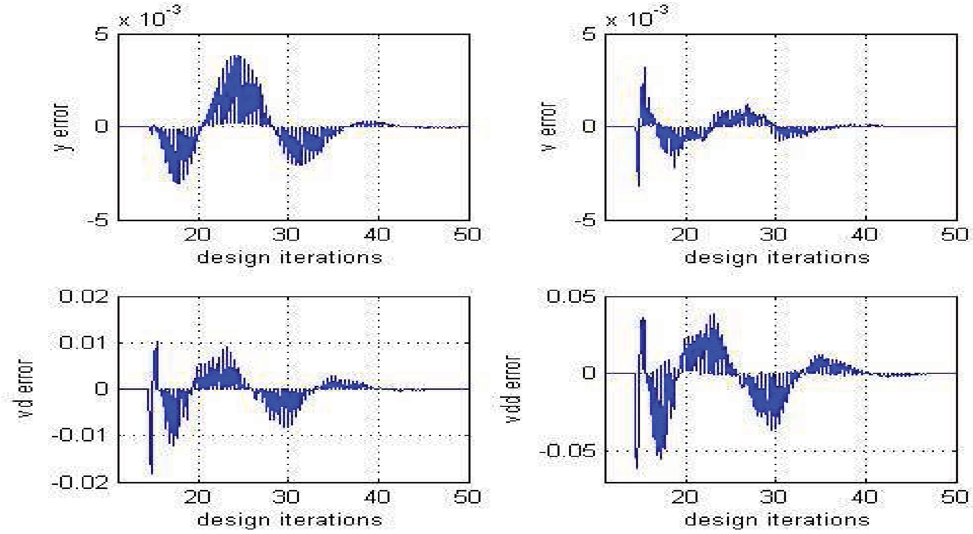


Figure 139 - Example-2, y-axis tracking errors between the vehicle and the RH trajectory

8.5 Simulation Results - *Impact of the Receding Horizon Design Rate*

As mentioned previously, in order to provide rapid reaction to newly detected obstacles and disturbances it is desirable for the output space receding horizon trajectory design to occur at a high rate. The aim of the simulation tests presented in this section were to ensure that a high rate trajectory design does not interfere with the performance of the control space receding horizon trajectory tracking layer. Tests were performed for the global trajectory shown in Figure 131 with three different design rates, 1.25Hz, 2.5Hz and 5Hz, and the resulting position errors (global trajectory and receding horizon trajectory) are shown in Figure 140 to Figure 142. It can be seen from Figure 140 that the design rate has little impact on position errors from the global trajectory. These errors are defined predominantly by the performance of the output space receding horizon trajectory design, and may be reduced by a variety of methods (i.e. increased order of polynomials, increase optimisation effort, reduced performance limits in the global trajectory, etc.). It can be seen in Figure 141 and Figure 142 that as the design rate is increased, the tracking errors from the receding horizon trajectories decrease, with the 5Hz design reducing position errors to approx. 0.002m. It can also be seen that speed, acceleration and rate of acceleration errors are reduced by increasing the receding horizon design rate. This is a key result, demonstrating that the desired high rate output space design can be successfully tracked by a separate control space component without it significantly degrading overall performance. Without this result the division of the problem into distinct output and control space layers would either not be feasible, or would require a more complex tracking controller, therefore further complicating implementation on a real vehicle.

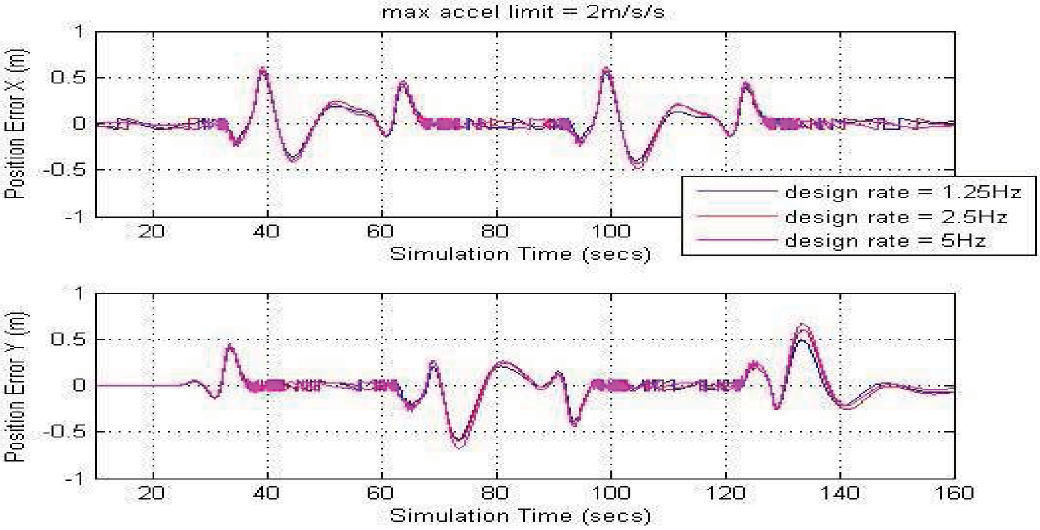


Figure 140 - Impact of design rate on total position errors

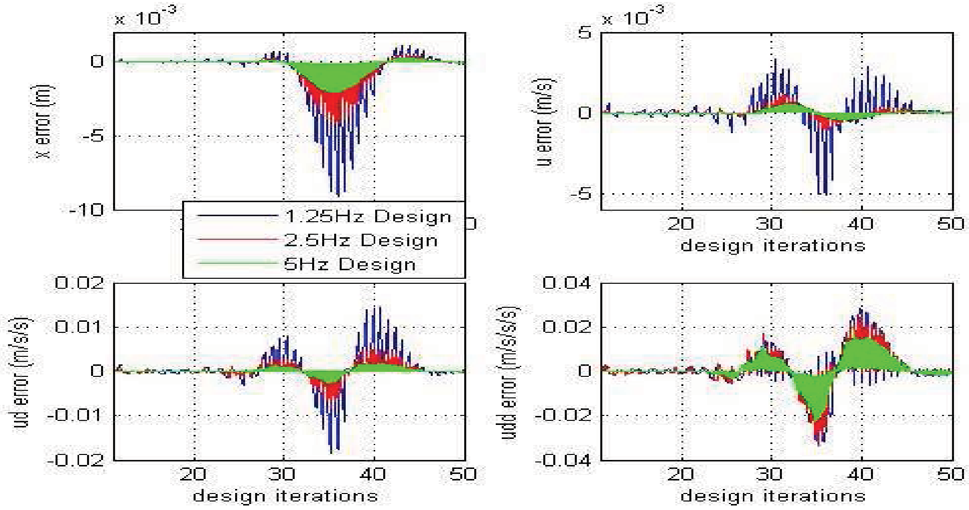


Figure 141 - Impact of design rate on receding horizon tracking errors (x-axis)

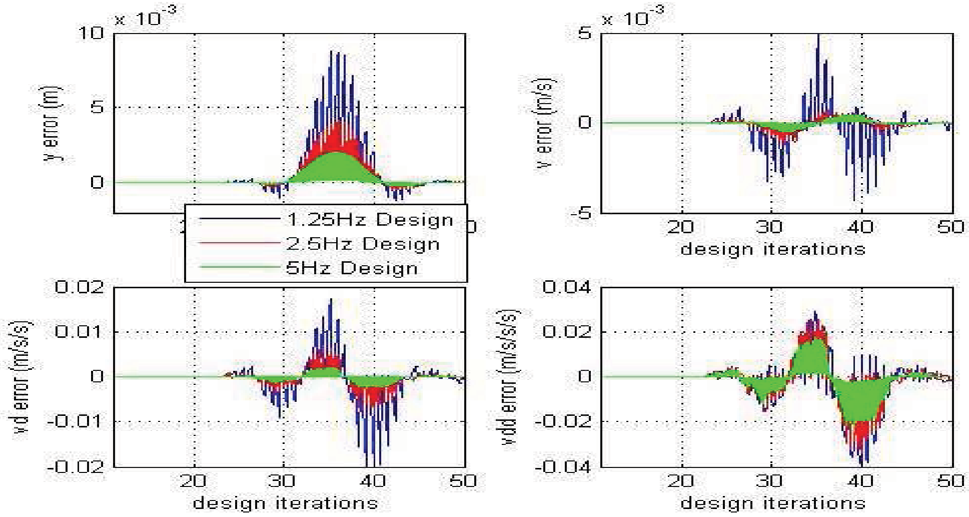


Figure 142 - Impact of design rate on receding horizon tracking errors (y-axis)

8.6 Simulation Results - *Characteristic Response of Poor Trajectory Tracking*

At the start of each receding horizon trajectory design process the current vehicle state is imposed via boundary conditions to the polynomial trajectory. This ensures that the newly designed trajectory will progress smoothly from the current vehicle position. However, if the receding horizon tracking layer does not provide the expected vehicle state (i.e. due to tracking errors) by the start of the next design iteration then these tracking errors will be imposed on the subsequent trajectory design, resulting in the previously expected manoeuvre no longer being feasible. The characteristic behaviour that results from this process is illustrated in Figure 143, where it can be seen that due to the tracking errors being imposed via boundary conditions, the trajectory designed at one iteration is no longer feasible at the next iteration, which results in repeated infeasible design and poor overall behaviour.

This is the key behaviour that the tracking controller needs to avoid, with the primary trigger being a lagging response between the receding horizon trajectories and the vehicle. A lead response is preferred as this allows the previous trajectory to be continued without exceeding the vehicle performance limits.

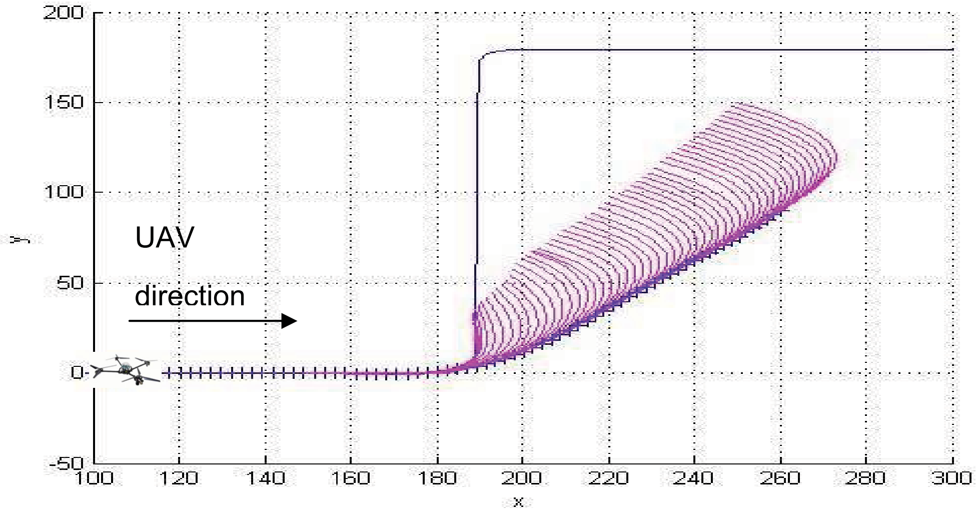


Figure 143 - Impact of poor RH trajectory tracking

9 Alternative Local Motion Planning Modes

9.1 Introduction

The work discussed so far within this thesis has focussed on the use of the situation aware LMP framework to track a large scale, or global, trajectory, a problem referred to as situation aware trajectory tracking. However, consideration of desirable missions for small / micro UAVs within complex environments suggests that other modes of operation may also be desirable. For example, if an operator is using an unmanned vehicle to perform surveillance / reconnaissance within a complex environment then in order to obtain a specific line-of-sight (i.e. through a window, or under a bridge) he / she may want to guide the vehicle by making small / local adjustments to its position. If there are no known obstacles between the current and desired position, then given that the LMP is situation aware the design of a global trajectory to track may be unnecessary, therefore suggesting utility in a point to point vehicle positioning approach. Note that this guidance mode differs from coarse waypoint tracking in that the vehicle would only have a single target position and would be expected to hold that position once obtained.

Additionally, if the small / micro UAV is being used to track a moving target then the design of a global trajectory is complicated by not knowing where the target will go in a global sense. In such circumstances the vehicle could be controlled relative to the target, rather than held to a pre-designed obstacle free global trajectory, and for operation within a complex environment this would benefit from situation aware local motion planning.

From an implementation perspective, the key difference between the situation aware trajectory tracking mode and the point to point or target tracking modes is the design of the objective function. For the trajectory tracking problem the objective function is as shown in equations 5-26 to 5-31, with the desired positions and speeds across the design horizon provided by the least-squares curve-fit of the global trajectory (see section 5.7.1). Provision of different modes of operation would require a change to the calculation of this target trajectory, therefore potentially altering the behaviour of the approach.

The aim of this chapter is to demonstrate the use of the situation aware local motion planning framework to provide the modes discussed above. All simulation results assume perfect tracking of the designed receding horizon trajectories.

9.2 Simulation Results - *Point to Point Navigation*

In order to provide point to point guidance the objective function requires a target position, rather than a target trajectory. This can be implemented by reducing the target trajectory to a single point in 3D space, therefore the optimisation will attempt to position the vehicle at the desired position at all points in the design horizon, therefore providing position acquire and hold behaviour, rather than simply position acquire. The simulation results presented in this section demonstrate that this doesn't adversely affect the behaviour of the local motion planning framework, either in terms of optimisation or vehicle performance.

The ability of the LMP framework to acquire a stationary position is shown in Figure 144, where the total position error is shown for a range of initial errors. It can be seen that for each case the position error is driven to zero with little or no position overshoot. A time history of vehicle state in the x-axis for the case of an initial error of 40m is shown in Figure 145, where it can be seen that vehicle performance limits are successfully enforced, while also approaching maximum vehicle performance.

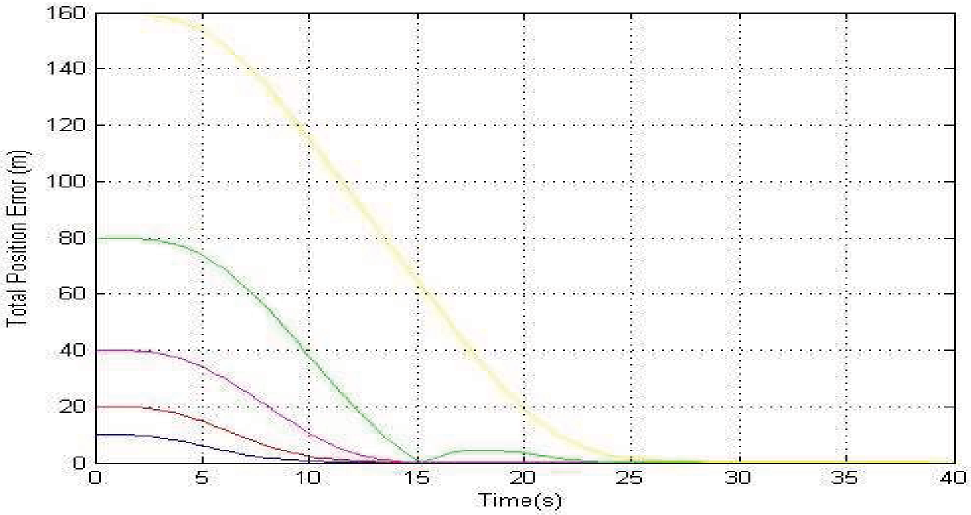


Figure 144 - Position acquire performance with a range of initial errors

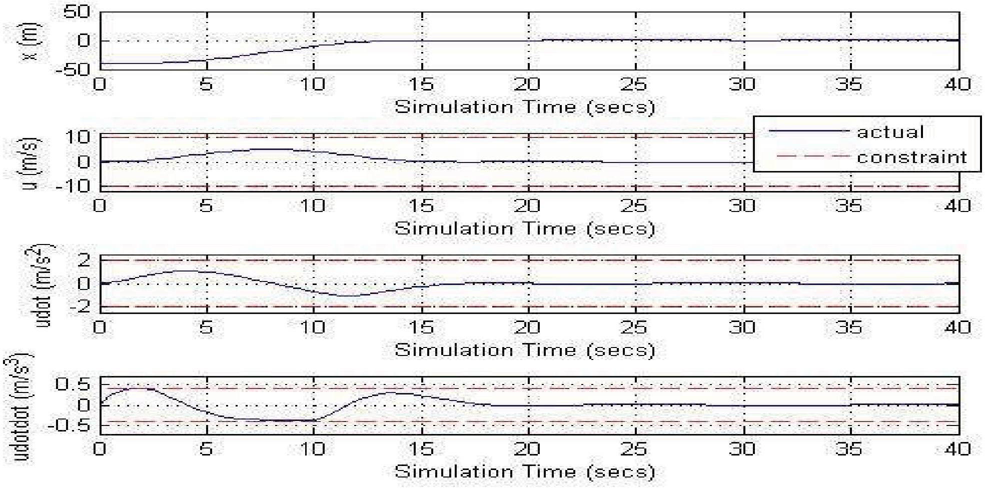


Figure 145 - Vehicle u-axis performance during position acquire from initial error of 40m

A second example is provided in Figure 146 to Figure 148, where the vehicle is tasked to acquire then hold a series of positions in close proximity to a set of static obstacles. A series of random disturbances to the position, speed and acceleration of the vehicle are also added each time knocking the vehicle off course or position. It can be seen that the position errors are rapidly reduced to zero after either a disturbance or new target position, therefore again confirming the ability of the framework to provide the desired point to point behaviour. The computation effort during this test is shown in Figure 148 where it can be seen that the peak effort is approx. 1.5s. These peaks occurred either after a disturbance and when a new target position was activated and result from the increased optimisation effort required due the previous design being far from optimal. Although this level of effort is greater than real-time, it still suggests that real-time performance may be feasible after implementation in a compiled language such as ‘c’.

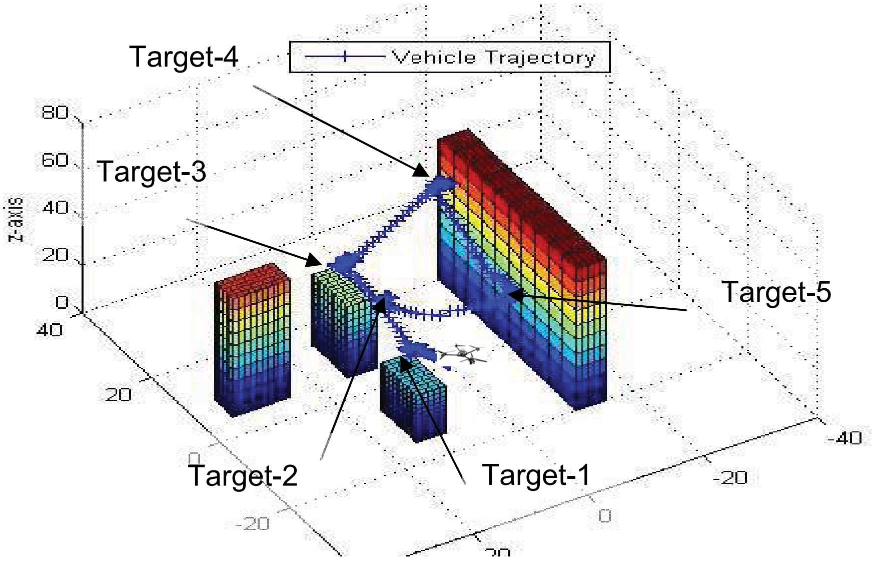


Figure 146 - Situation aware point to point guidance near static obstacles

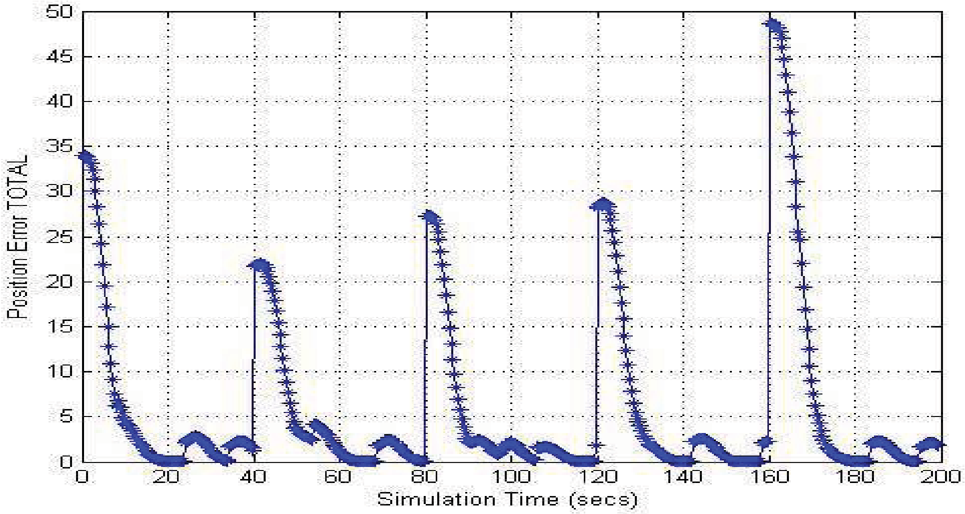


Figure 147 - Total position errors in presence of disturbances

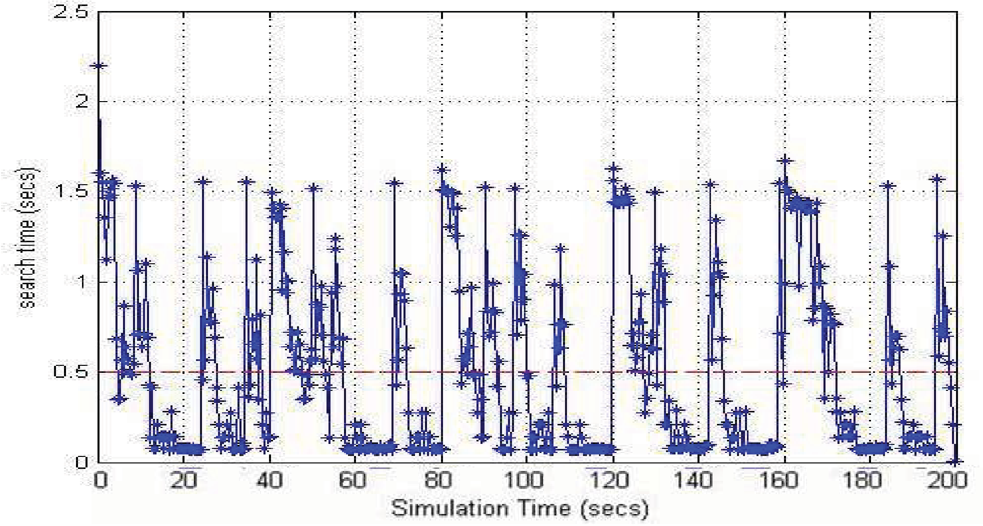


Figure 148 - Computation effort

A final example is provided in Figure 149 and Figure 150 where the quadrotor is attempting to hold a fixed position while avoiding three incoming dynamic obstacles. It can be seen that the quadrotor successfully avoids all three obstacles, each time returning to the target position.

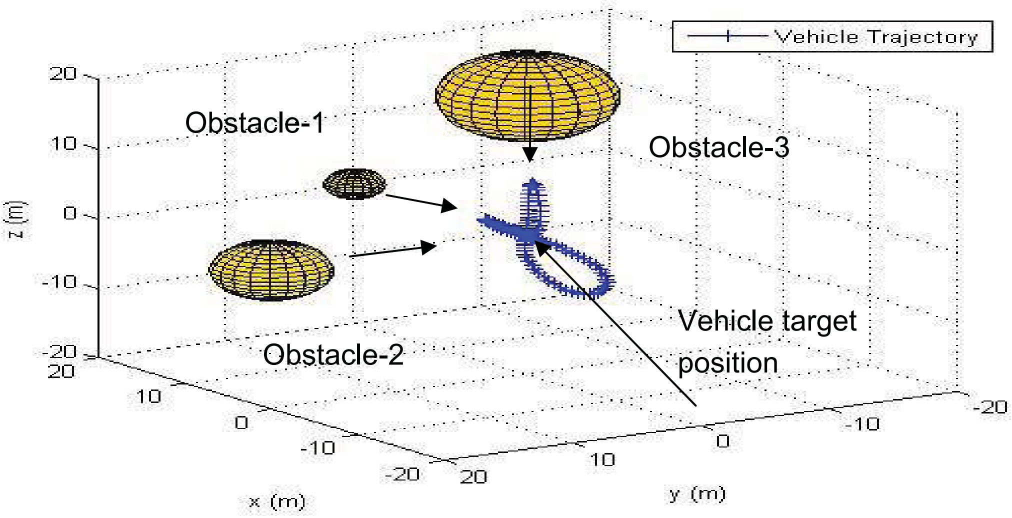


Figure 149 - Position Hold with Incoming Obstacles (overview)

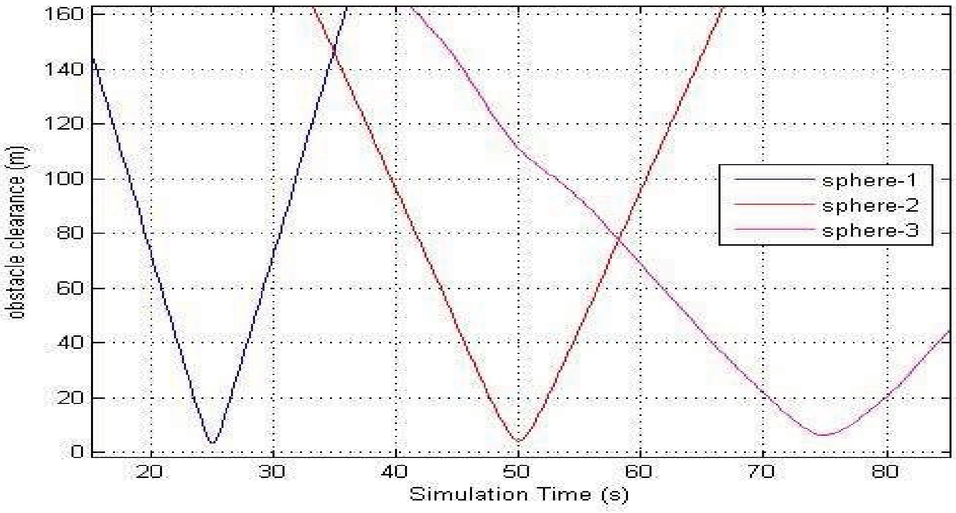


Figure 150 - Position hold obstacle clearance distances

An animation illustrating the examples discussed in this section is provided in the file 'animation 16 - point to point mode' where the quadrotor vehicle is tasked with several target positions in the presence of disturbances, static and dynamic obstacles. The current target position is shown as a red star, and the disturbances can clearly be seen as the vehicle being displaced every few seconds. The ability of the situation aware receding horizon planner to deal with such disturbances is clear, provided the vehicle has sufficient control power to avoid collisions.

9.3 Simulation Results - *Tracking a Moving Target*

The provision of target tracking capability is similar to point to point guidance, with the added complexity that the target position may be moving. This then introduces the prediction issue encountered with dynamic obstacles, and again the approach used here is to propagate the current target state across the design horizon. The predicted target position and speed across the design horizon can then be used directly as the target trajectory in the objective function. A moving target tracking example is shown in Figure 151 to Figure 153, where the quadrotor has been tasked to hold position above the defined target. It can be seen in Figure 152 that the target speed in the u-axis exceeds the quadrotor performance limit between 115s and 140s, which results in a position error build up. During this time the quadrotor holds maximum speed, then drives the error back to zero once the target speed reduces. The presence of static or dynamic obstacles would be treated in a similar fashion, with the vehicle prioritising obstacle avoidance over target tracking in the same manner demonstrated for the situation aware trajectory tracking problem.

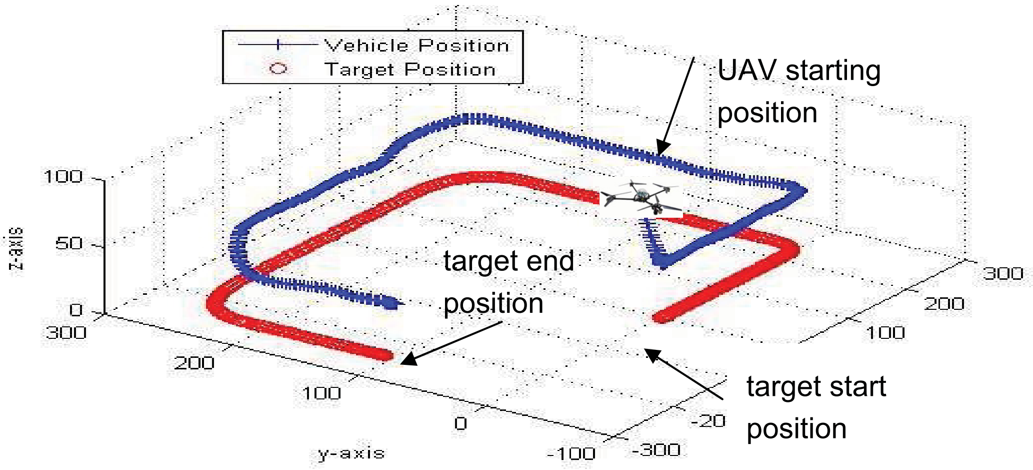


Figure 151 - Target tracking example

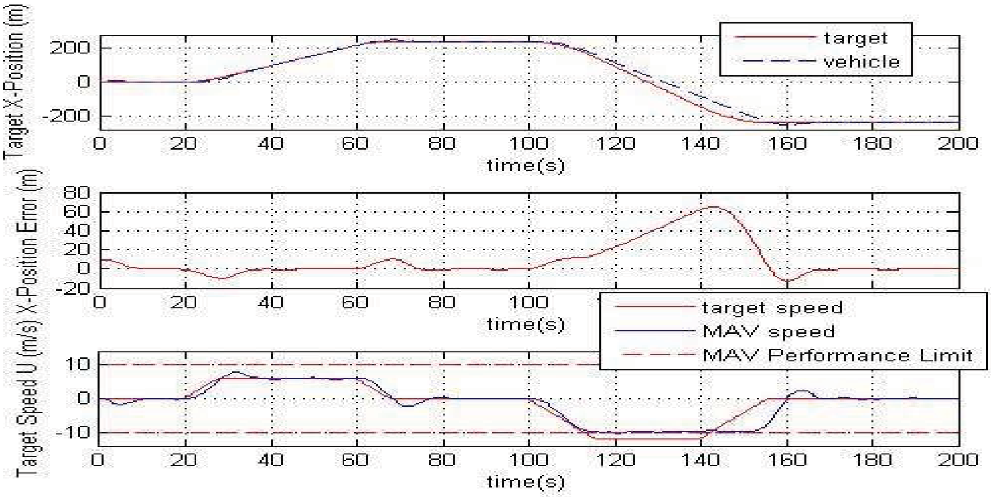


Figure 152 - Target tracking, u-axis time history

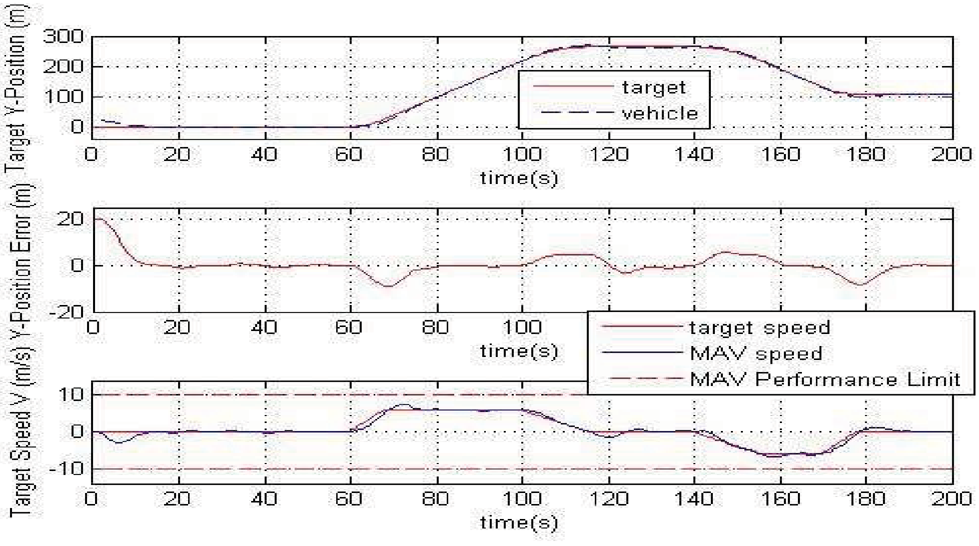


Figure 153 - Target tracking, y-axis time history

10 Additional System Level Issues

The aim of this chapter is to consider some final system level issues that would arise with implementation of the local motion planning framework on a real vehicle. The issues considered are:

- *Handling / impact of wind effects*
- *Realistic environment scenarios*
- *Computation effort*

This list is not exhaustive, with additional key issues including communications, obstacle sensors, navigation / state sensors, integration with mission planning algorithms, etc. Further consideration of these issues is desirable but beyond the scope of this thesis.

10.1 Handling Wind Effects

From the literature review presented in Section 3.2 it is clear that the wind field within urban environments can be highly complex, resulting from a combination of steady, unsteady and turbulent effects. The precise flow field encountered is dependent on the wind strength, direction and turbulence content, as well as the detailed geometry of the environment. Given the complex and unsteady nature of the flow-field it is unlikely that a detailed and accurate map of the wind field will be available, therefore reinforcing the need for the vehicle to be able to handle significant and unexpected disturbances. The presence of such disturbances is part of the argument for the need for the continuous local motion planning layer.

However, it should be feasible to predict some of the primary steady flow features, therefore allowing the global and local planning layers to anticipate significant changes in the wind field due to the geometry of the environment. For example, the 'suction' effect behind a building can be predicted then factored into the LMP layer to reduce the chance of the vehicle becoming drawn closer to a building than desired. Anticipation of this effect would drive either a wider clearance distance or a planned airspeed change to compensate for the effect, effectively reducing the size of the disturbance. Similarly, flow channelling and sheltering effects should be fairly steady therefore allowing them to be accounted for. Channelling flow in particular is likely to be a significant component of the vehicle's maximum airspeed, therefore this effect would need to be factored into both global and local planning layers. This would require the creation of a 3D flow field map of the environment that could be accessed by the planning layers as required. This map would be subject to change with significant changes to the wind direction or strength but could otherwise be considered as static, and generated before the start of the mission.

Unsteady flow effects and the differences between predictions and reality would need to be dealt with by the control and output space components of the LMP framework. As the control space layer is tracking a trajectory through space, it should be possible for it to reduce the scale of such disturbances seen by the output space layer. This would be aided by reducing the performance limits used in the LMP layer, therefore providing a greater performance margin for the control space layer to use. This would be akin to a human driving a car carefully or slowly in difficult conditions and would be simple to implement within the local motion planning framework. By comparing airspeed with ground speed a vehicle is able to measure the movement of the air mass that it is in, therefore allowing such differences to be factored into current receding horizon trajectories. The key advantage of having a 3D map of predicted major effects (rather than only instantaneous measurements) is that it allows the receding horizon trajectory optimisation to be based on the wind speed expected at each location, rather than the wind speed currently experienced by the vehicle.

Simulation examples demonstrating the impact turbulence and gusts are provided within a realistic environment in the following section. In these tests it is assumed that the steady flow is zero, therefore the turbulence and gust disturbances need to be measured in real-time and factored into to the current receding horizon trajectory.

10.2 Realistic Environment Test & Demonstration

All previous simulation tests used random or challenging arrangements of vehicles or obstacles in order to test specific elements of the LMP framework. The aim of this section is to place the local motion planning framework within a more realistic environment, also including representative turbulence and gust effects.

The environment used for these tests is representative of a generic medium rise city development, with a building arrangement that results in canyon widths of 30m (parallel with x-axis) and 20m (parallel with y-axis). The thinner canyon is the one used for the challenging parts of the scenarios. The global trajectory to be followed is at a fixed height of 30m, with the various building heights ranging from 20m to 50m. This environment is illustrated in Figure 156 and primarily acts to restrict the manoeuvring of the vehicles is a way that could be expected in reality. These tests are also intended to confirm that the various scaling / design parameters used within the framework are suited to a realistic environment.

The turbulence and gust model used in these simulations is shown in Figure 154 and Figure 155, and are based on a statistical discrete gust (SDG) model developed by Brindley and Bradley [6]. This model is comprised of a standard gust shape which is then combined with a statistical representation of the likely frequency content of the disturbances (i.e. occasional large gusts with more regular smaller gusts). The frequency and scale of the turbulence is designed to represent low level flow in the presence of large structures, and therefore is appropriate for this work. In the first model the disturbances peak at over 3m/s in each axis, compared with the quadrotor's maximum airspeed of 10m/s. In the second model the scale of the disturbances is reduced by approximately half. The magnitude of the turbulence content is dependent on several factors including the free-stream wind speed and turbulence, channelling and sheltering effects, altitude etc. and the values used here were chosen to be close to the upper level of what the vehicle's performance limits are able to cope with.

All simulation results shown in this section are for the output space element of the local motion planning framework only, therefore it is assumed that the designed trajectories are perfectly tracked. The disturbances that are imposed therefore represent the impact of gusts and turbulence after the control space layer has attempted to minimise the impact on the output space component. It is also assumed that the vehicle is able to measure the wind speed currently experienced, therefore allowing compensation for sustained gusts.

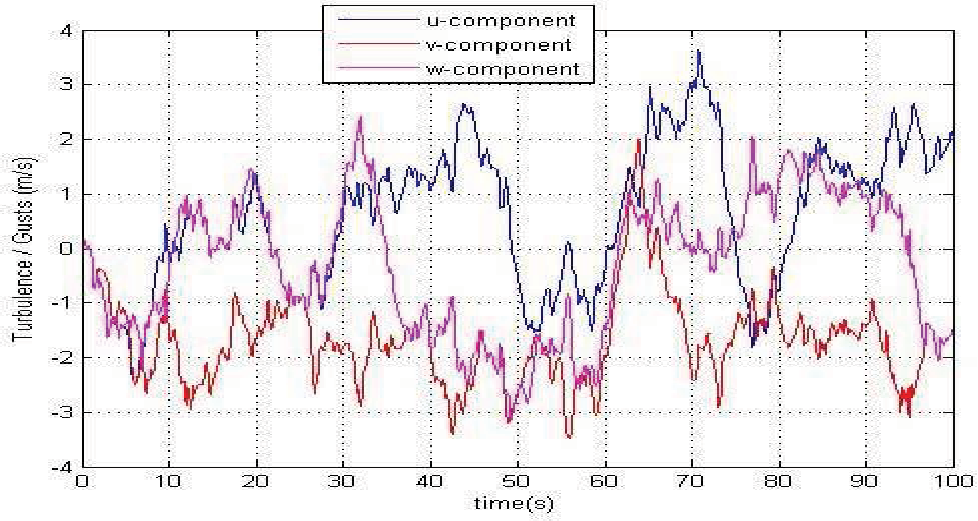


Figure 154 - Gust & turbulence model-1

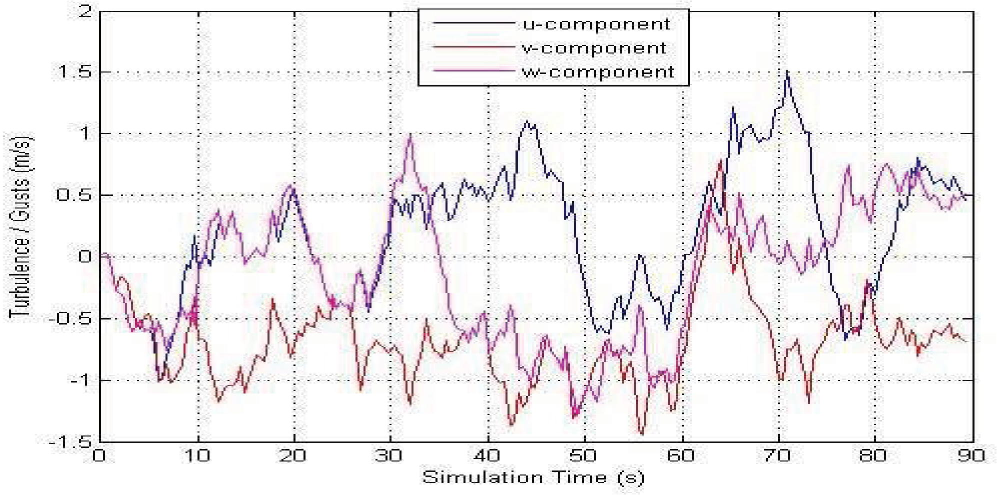


Figure 155 - Gust & turbulence model-2

10.2.1 Simulation Example-1: Single Vehicle with Turbulence Model-1

In the first scenario a single quadrotor vehicle tracking a global trajectory through an urban canyon 20m wide encounters a single dynamic obstacle on a head-on collision course travelling at 10m/s. The obstacle is modelled as a sphere of radius 4m, and therefore blocks almost half of the canyon width. Additionally, the vehicle also experiences the gust / turbulence model shown in Figure 154. The results of this simulation are shown in Figure 156 to Figure 161, where the impact of the disturbances can clearly be seen. The vehicle continually manoeuvres up to the rate of acceleration performance limits in order to minimise deviations from the global trajectory (shown in Figure 161), although this is not sufficient to avoid position errors of approximately 5m (max) accumulating. All performance limits are successfully enforced, and It can be seen in Figure 160 that all obstacles are avoided, including the dynamic sphere.

An animation illustrating the impact of the disturbances in the first leg of this scenario is provided in the file 'animation 17 – turbulence'. It can be seen that although the vehicle experiences significant buffeting, the receding horizon trajectory always leads back to the global trajectory, providing both a stabilising effect as well as ensuring that the path back to the trajectory is obstacle free.

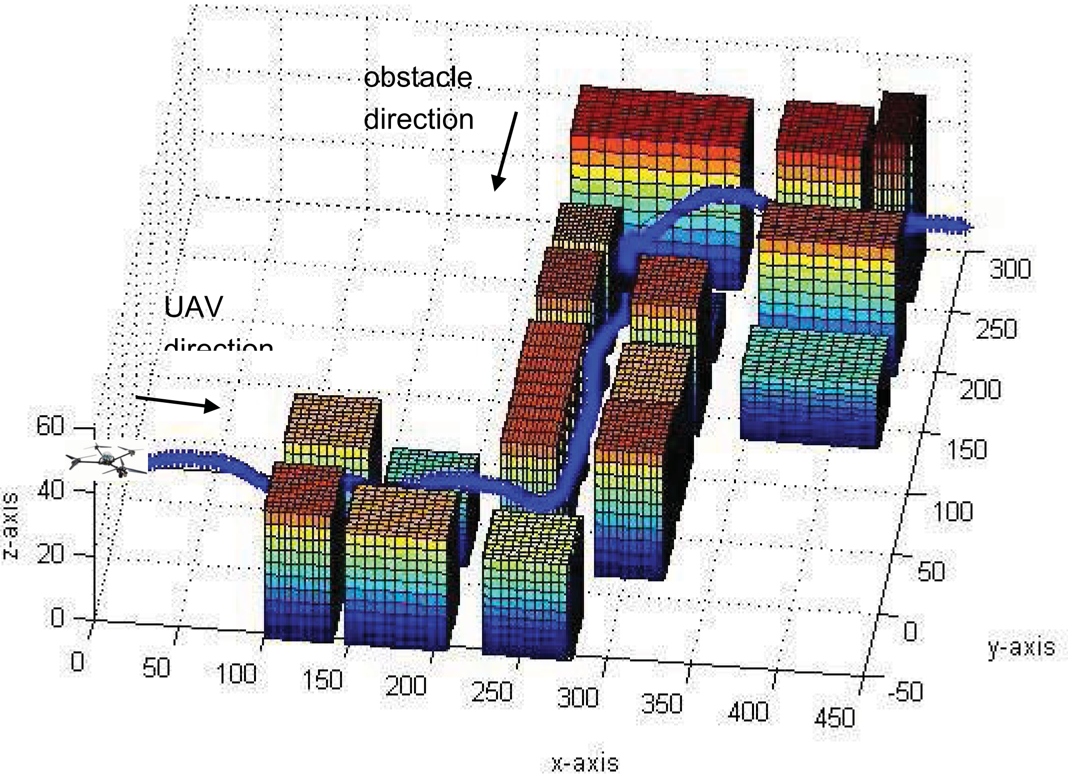


Figure 156 - Realistic environment example-1, overview

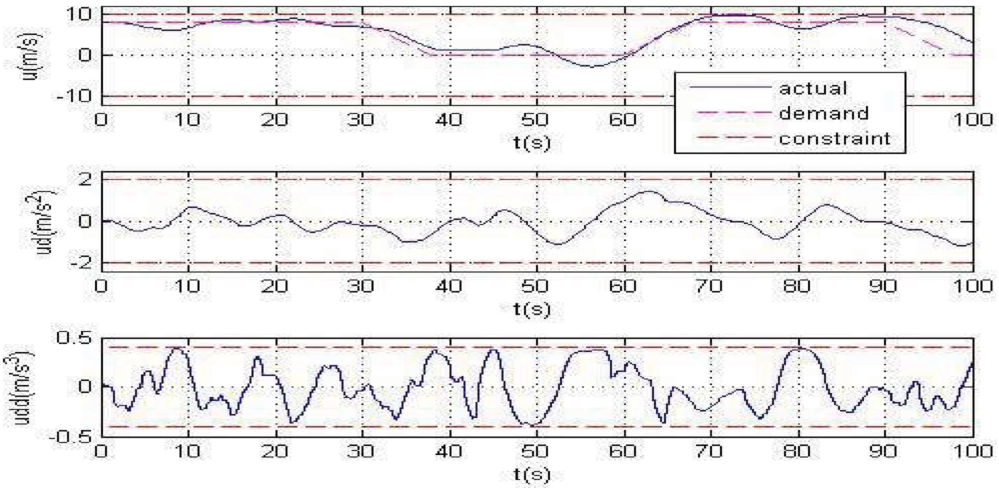


Figure 157 - Realistic environment example-1, u-axis time history

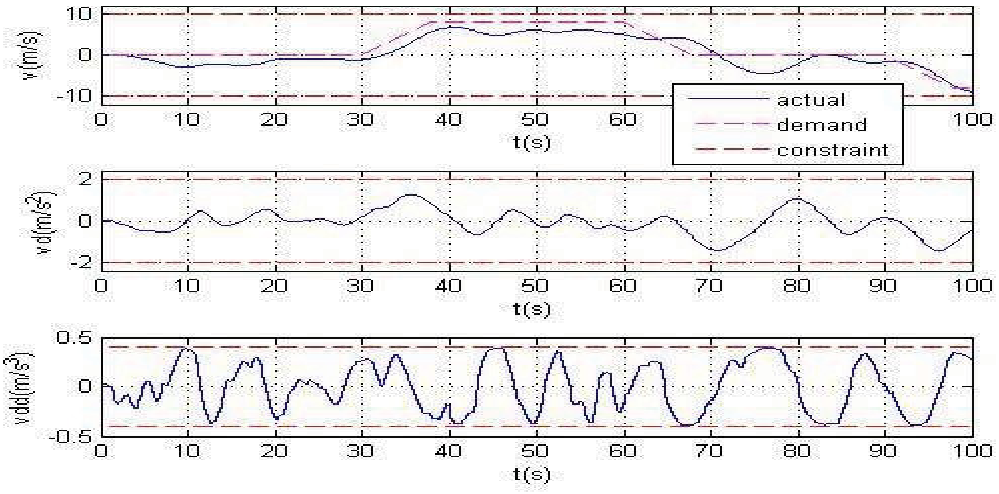


Figure 158 - Realistic environment example-1, v-axis time history

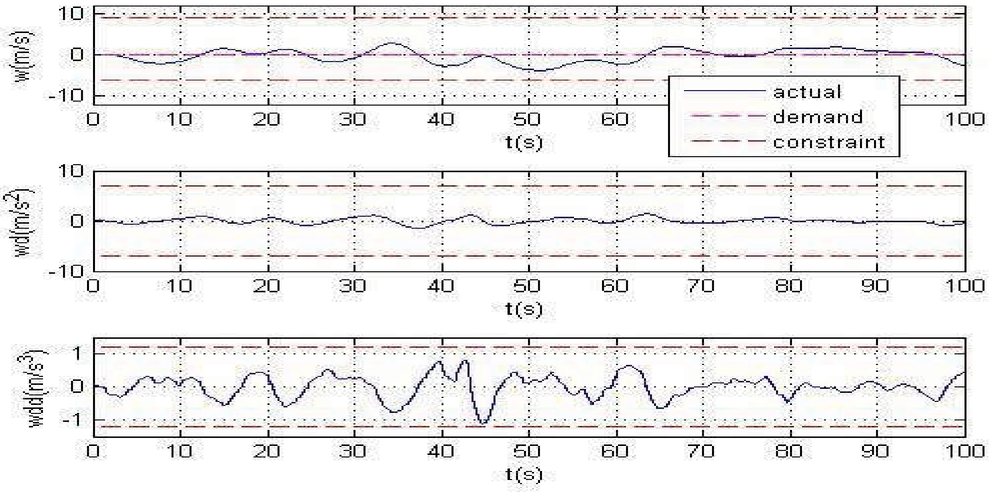


Figure 159 - Realistic environment example-1, w-axis time history

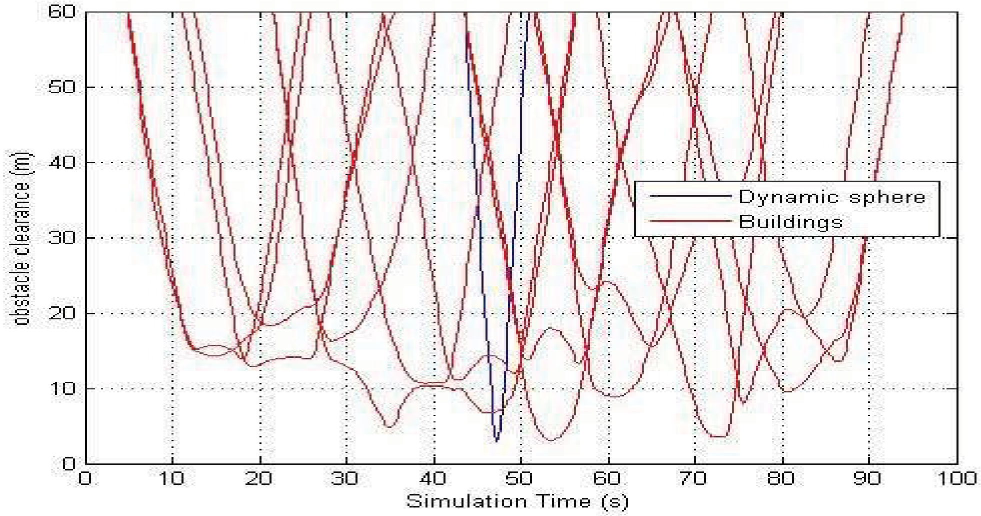


Figure 160 - Realistic environment example-1, obstacle clearance distances

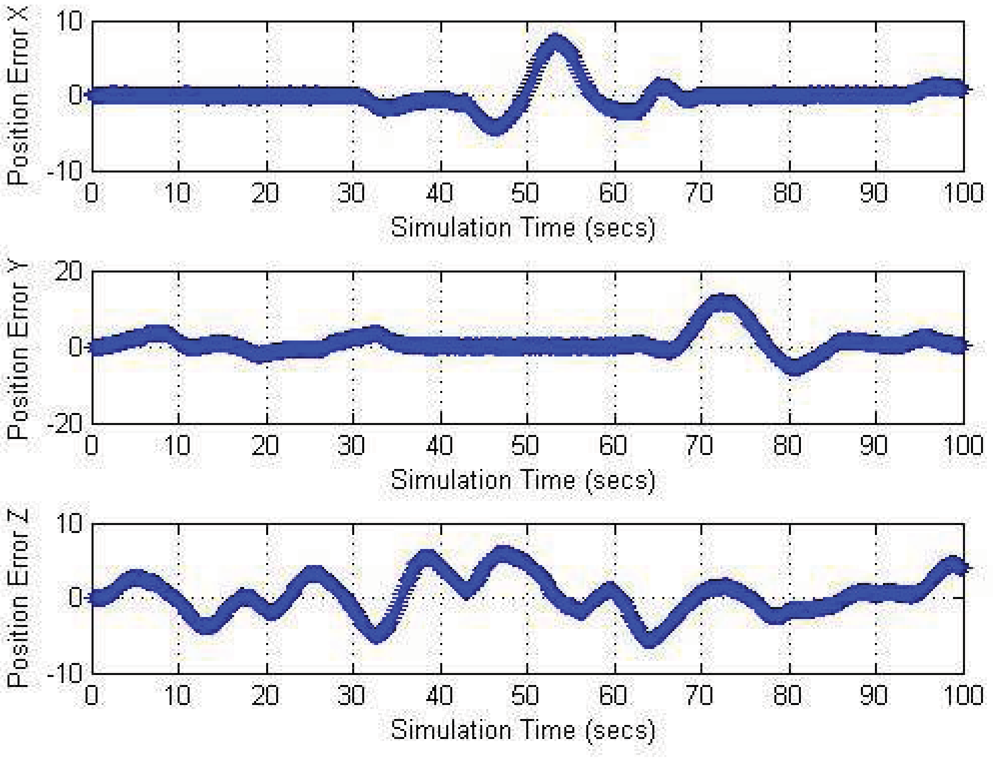


Figure 161 - Realistic environment example-1, position error time history

10.2.2 Simulation Example-2: Multiple Vehicles with Turbulence Model-2

The second scenario was designed to be more challenging, using the same environment, but this time with a formation of three quadrotors from one system encountering a fourth quadrotor moving in the opposite direction in the 20m canyon. All vehicles experience gusts and turbulence with disturbances in each axis peaking at approximately 1.5m/s as shown in Figure 155. No communication was employed between any vehicle, and the cross product avoidance rules were enabled.

The results of this scenario are shown in Figure 162 to Figure 168, where the time histories for all four vehicles are shown in each plot. The impact of the disturbances is again clear, with each vehicle again manoeuvring up to it's rate of acceleration performance limits to minimise deviations from the global trajectory. All other performance limits are also enforced and all collisions are successfully avoided (as shown in Figure 166 and Figure 167). As with the previous example, the performance limits of the vehicle do not allow the disturbances to be removed immediately, therefore position errors from the global trajectory can be seen across the simulation, although they are typically smaller than those seen with the larger gust / turbulence model.

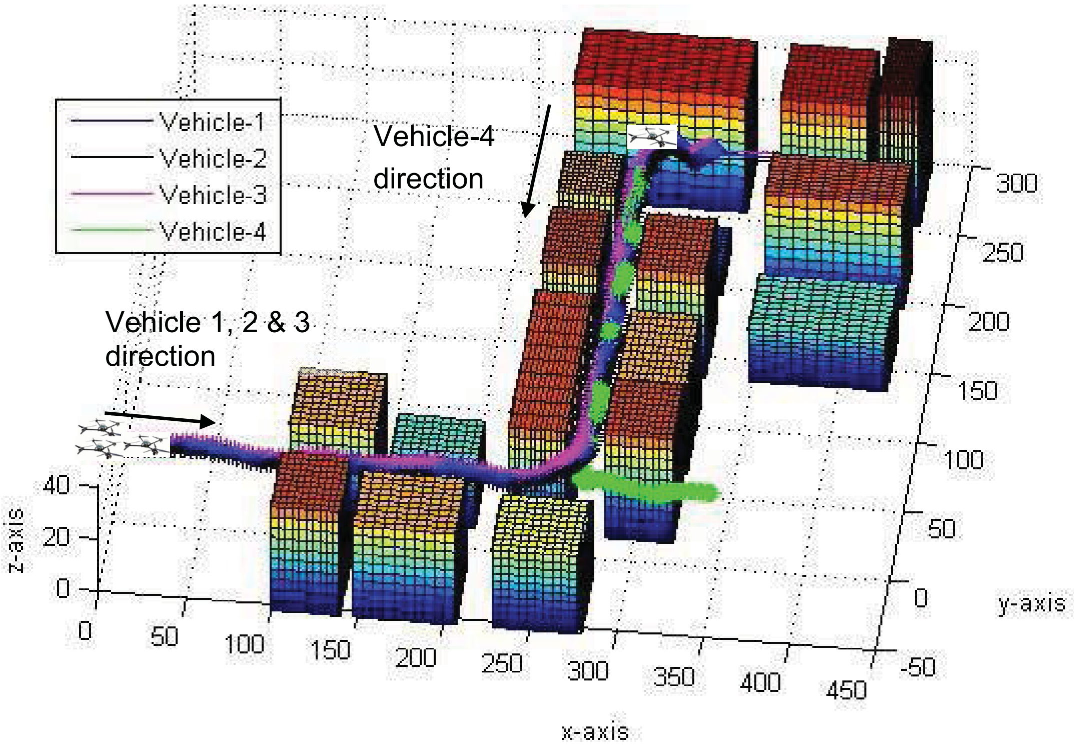


Figure 162 - Realistic environment example-2, overview

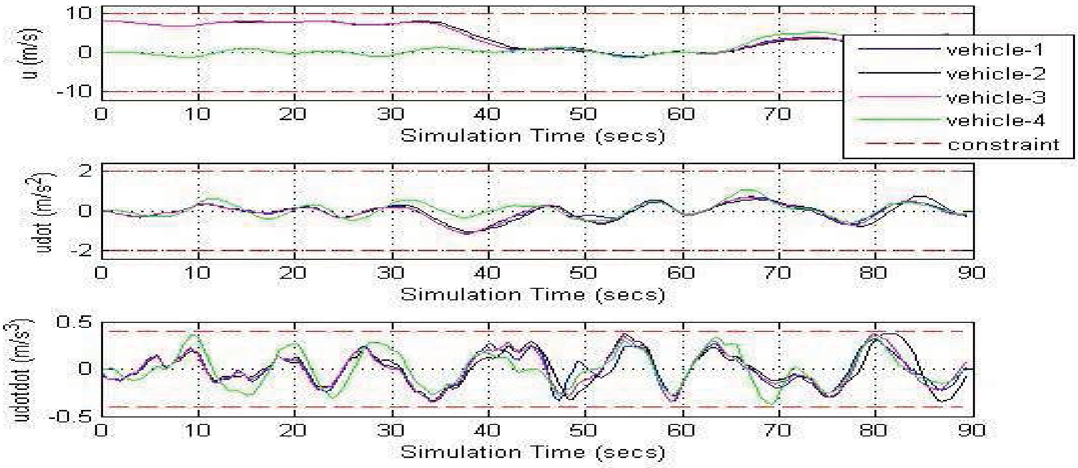


Figure 163 - Realistic environment example-2, u-axis time history

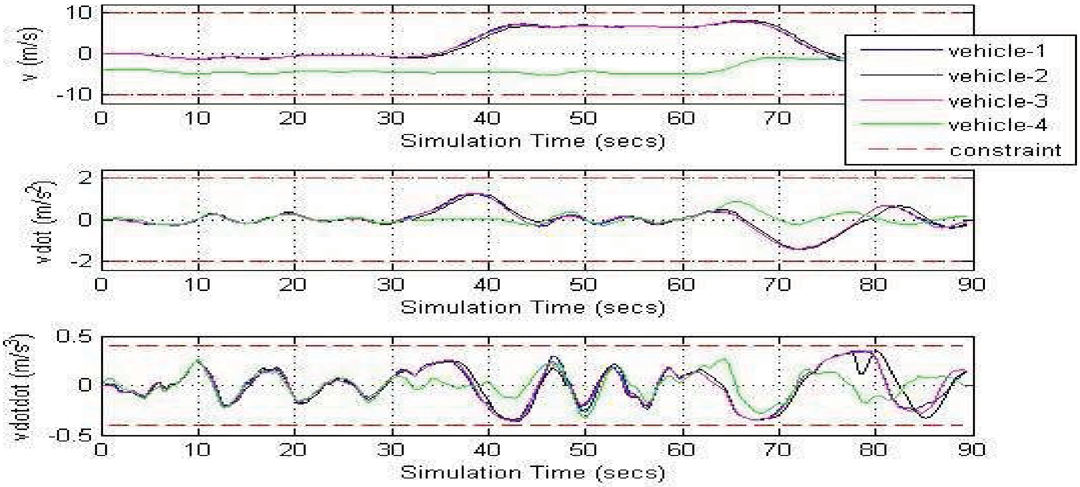


Figure 164 - Realistic environment example-2, v-axis time history

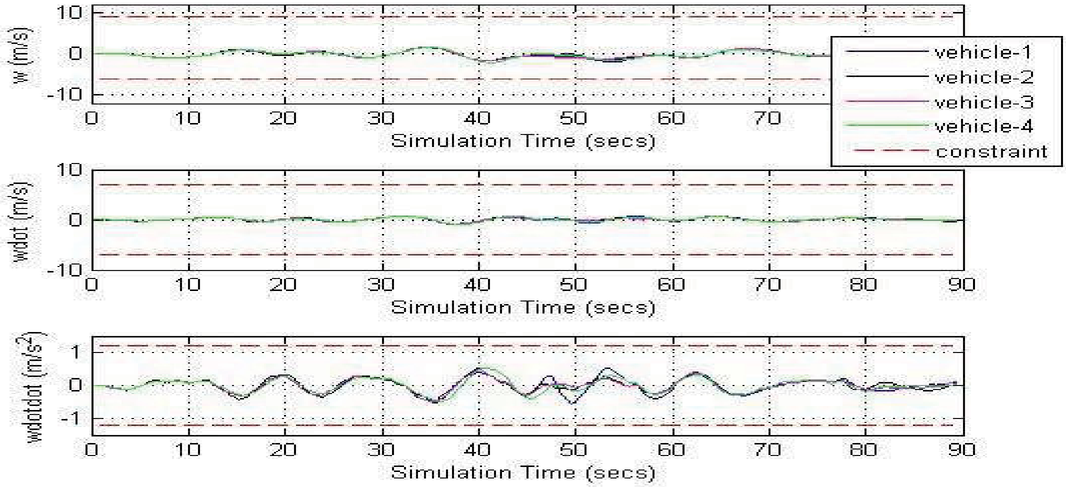


Figure 165 - Realistic environment example-2, w-axis time history

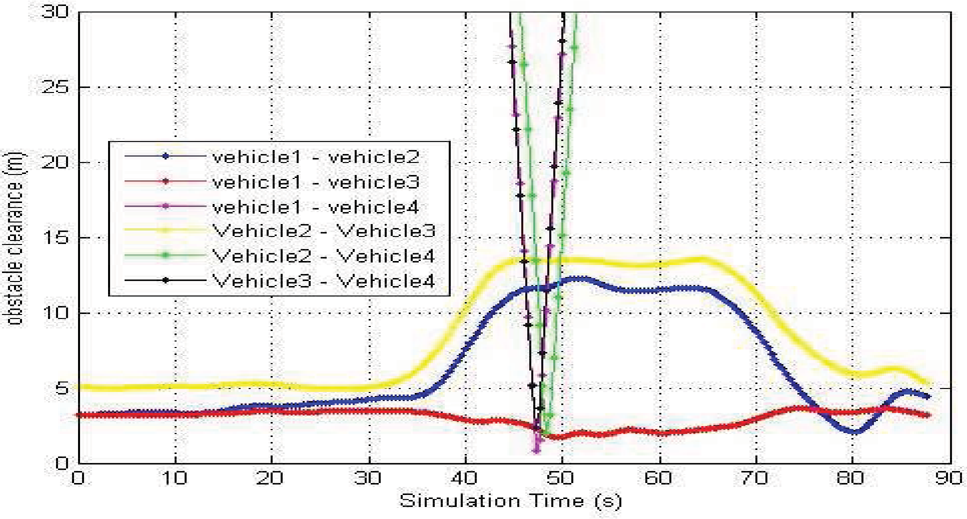


Figure 166 - Realistic environment example-2, vehicle clearance distances

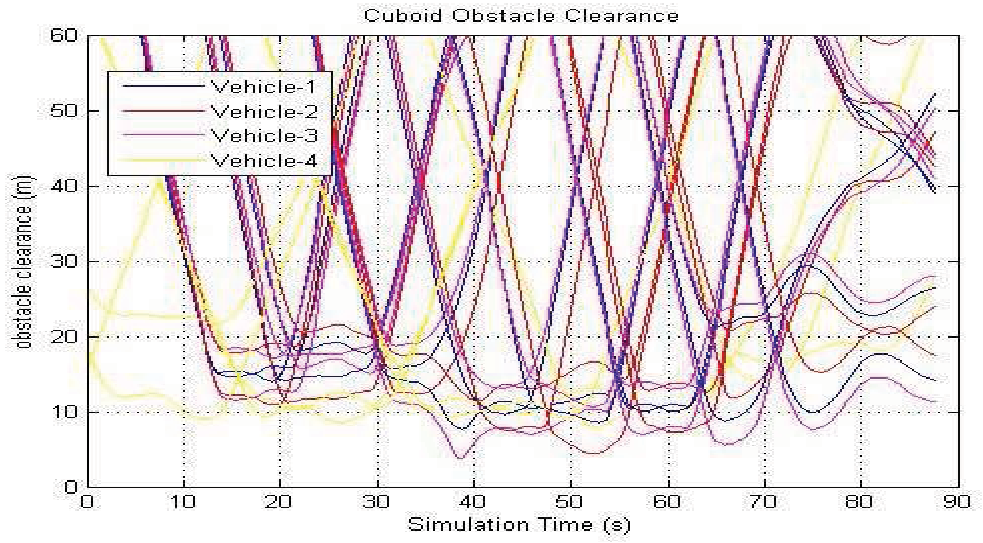


Figure 167 - Realistic environment example-2, building clearance distances

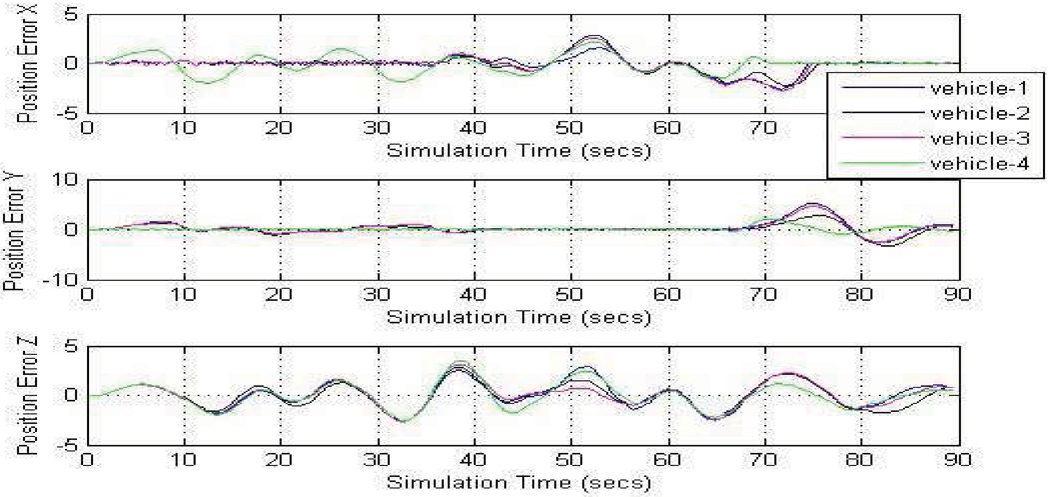


Figure 168 - Realistic environment example-2, position error time history

10.3 Computational Effort

Although implementation of the LMP framework on a real vehicle was beyond the scope of this thesis, it is interesting to consider currently available processing power for such vehicles. A brief and non-comprehensive survey of such capabilities was conducted, with the results outlined in Appendix F. Examples of the computational effort required for the LMP framework were provided in Section 6.6, where the peak optimisation time was under 1s. Implementation of the coarse grid of feasible trajectory options results in a further increase in effort, particularly if the optimisation begins from one of the coarse grid options. This is due to the fact that the starting design is likely to be far from the final design, therefore suggesting that improvements could be realised by suitable optimisation of the search algorithm. Similarly, for more complex scenarios involving multiple obstacles and vehicles the computation effort may increase further, with occasional peaks twice that previously reported. These results were based on scripted m-files rather than a compiled language, and no significant effort was directed towards optimising the search algorithm, therefore significant effort reduction should be feasible.

Consideration of a) the optimisation effort discussed above, b) the processing power of the development machine (see Section 6.6) and c) the COM / Autopilot units presented in Appendix F suggests that real-time implementation of the LMP framework on a real vehicle may be possible. Optimisation of both the search algorithm and the entire framework implementation as well as implementation in a compiled language such as ‘c’ is likely to reduce the required effort significantly. Additionally, the search algorithms may be enclosed within an ‘anytime’ framework, where the best result available after the time limit has expired is output. This type of approach benefits from the fact that good system performance does not require perfect optimisation. Additionally, the initial part of the trajectory is fixed by the boundary conditions and each successive optimisation improves the previous design, therefore the true optimal trajectory may emerge from a series of sub-optimal trajectories.

However, it must be noted that the LMP framework is just one of many computationally expensive components that would be required on a real-vehicle. Although some of these components may require their own processing boards, integration of all the necessary system elements using current technology is likely to be challenging. It must also be noted that available processing power is still becoming available in smaller and lighter units, therefore any implementation issues encountered at present are likely to be temporary.

11 Discussion, Conclusions & Recommendations for Future Work

The increasing demand for unmanned vehicles within complex obstacle rich environments has been discussed. The work presented within this thesis is based largely on an argument that the operation of unmanned vehicles within complex environments benefits from the use of a continuous LMP layer, rather than the more traditional global planning plus blind trajectory tracking and event triggered collision avoidance. It was also argued that placing the reactive element within the global motion planning layer is inefficient, as many types of obstacles and disturbances can be handled at the local level. The adoption of such a LMP layer results in a motion planning and control architecture with distinct global and local layers, and it is the local planning layer that this thesis is primarily concerned with.

11.1 Comments on the Local Motion Planning & Control Framework

11.1.1 Output Space Framework – Single Vehicle

The technical work presented within this thesis has been focussed primarily on proof of concept, rather than detailed implementation for a particular vehicle. The quadrotor simulation model provided a realistic target vehicle, but the developed LMP and control framework is generic to vehicle type or domain, i.e. air, land, surface, underwater, space. Additionally, the aim was to focus on long-term desirable general behaviour rather than developing something for a specific vehicle.

Overall, the new LMP framework was found to be very successful. Confidence in the approach was gradually increased during an extensive range of simulation based tests, with adjustments made as necessary and subsequently incorporated within the standard framework. This resulted in a high degree of confidence being gained in the framework. It was found that in complex scenarios the vehicle could be made to collide with another vehicle or obstacle, but only rarely and primarily in the following circumstances:

- *Obstacle motion that is hard to predict* - For example erratic motion which results in each prediction across the design horizon being highly misleading.
- *Late detection of conflict* - For example due to obstacle occlusion, speed, sensor direction etc.
- *Vehicle performance limits* - For example if a disturbance results in the feasible manoeuvre envelope not avoiding the collision.

The use of a proximity function to provide obstacle avoidance benefits from not requiring obstacle depth information. However, it also suffers from not allowing a specific clearance distance to be specified. The overall behaviour results from a balance of costs between trajectory tracking and obstacle proximity, therefore as the trajectory error increases, the safe clearance distance reduces. In most circumstances this behaviour is likely to be acceptable, although provision may need to be made if large trajectory errors are considered likely.

Imposing vehicle performance limits via the penalty function approach was found to be very successful, and in none of the tests conducted was it found that performance limits were exceeded. The set of performance limits was shown to create an enclosing boundary to the feasible design space, within which manoeuvring can occur unaffected by the performance limits. This allows the majority of manoeuvres to be designed with the optimisation effectively unconstrained, therefore reducing computation effort. An additional advantage is that the framework allows the imposed performance limits to vary over the course of the design horizon, something that becomes more important as the horizon length increases.

When obstacles are detected this can have the impact of dividing the feasible design space into unconnected regions, therefore reducing the effectiveness of the gradient search. In these circumstances the coarse grid of feasible manoeuvres (Section 5.6.3) provides a mechanism for the search 'jumping' to a new region of the design space. This approach ensures that maximum performance manoeuvres are always available if necessary. It was also shown that a simple and efficient least squares curve-fit process could be used to allow a coarse-grid trajectory to be converted into the required format for optimisation, although some issues remain over guaranteeing performance limits are not exceeded when subsequently enforcing boundary conditions. It is again emphasised that global optimality is not necessary for good system performance.

A key element of the framework is the use of a target trajectory (Section 5.7) to allow rapid calculation of position and speed errors in the objective function. The least-squares curve-fit of the global trajectory can be performed very efficiently, removing the need for multiple nearest-point calculations. Additionally, this process also provides a mechanism of interpolating within a coarse specification of the global trajectory.

A potential downside of the approach is that the vehicle control has been removed from the global trajectory by several layers. For example, the final control space trajectory tracking layer no longer tracks the global trajectory. Instead, it tracks the situation aware receding horizon trajectory, which is optimised to match the target trajectory which has been calculated via a curve-fit on the global trajectory. However, each of these elements has been demonstrated to introduce only small errors, so the additional separation and complexity is considered worthwhile in order to make the tracking layer situation aware.

No significant effort was directed towards verifying that the sixth order Bezier polynomials matched the dynamic capabilities of the vehicle. However, it is clear that the use of a higher order polynomial would allow a better match to be provided if required. As the optimisation is gradient based the increase in computational effort would not scale linearly with the number of design variables. This is due to the fact that the majority of the search effort occurs within the line-search where the number of design variables has no impact. Additionally, it is also possible that the required dynamic feasibility can be provided with a lower order polynomial therefore reducing computational effort.

All of the simulation tests provided within this thesis were targeted at the quadrotor vehicle, with the local motion trajectories described by three polynomials describing the speed profiles in the forward, lateral and vertical axis. This formulation was used for the quadrotor vehicle as it provided a suitable match with the chosen performance limits of the vehicle. When applying the framework to a different vehicle it is likely that a different trajectory specification would be required, for example the performance limits of a fixed wing vehicle may suit the following description:

- Airspeed profile
- Heading angle profile
- Flight path angle profile

The aim when implementing the framework on a different vehicle would always be to try to match the description to the performance limits, simplifying the coupling between each axis and therefore simplifying the optimisation process. Depending on the vehicle dynamics it is possible that each profile would require a different order of polynomial, for example a fixed wing vehicle may have a slower dynamic response in airspeed than in heading or flight path angle.

11.1.2 Output Space Framework – Multiple Vehicles

The extensions to the single vehicle framework to allow decentralised deconfliction of multiple vehicles were also found to be very successful. The continuous ten second planning horizon provides a natural home for *immediate-term* deconfliction, where the priority is increasing separation (Section 7.1.3). Additionally, it was also shown that the basic proximity based objective function may be enhanced to provide simple rule based behaviour that either provides more acceptable behaviour or increases the efficiency of the decentralised approach. If the desired rapid response and operational flexibility (Section 7.1.1) is to be enabled then a greater reliance on this type of decentralised approach is considered necessary.

It was shown that for many highly complex scenarios vehicle cooperation was not required to provide good behaviour. However, the continuous local motion trajectory planning provides a significant opportunity for sharing trajectory intent, which was demonstrated to improve overall performance in the most complex scenarios. The polynomial based trajectory description is highly efficient, reducing the amount of information that would need to be transmitted. A simple transmit / receive approach avoids the need for explicit inter-vehicle negotiation, and was demonstrated to allow a form of virtual negotiation (Section 7.4) to occur. The key impact of intent sharing is that it removes a key error source, that of prediction of future positions of other vehicles.

A key argument that emerges from this work is that of delegation, and the required increase in vehicle autonomy. The concept of the situation aware trajectory tracking layer is driven by a delegation of authority from the global planning layer to the local planning layer. Endowing the trajectory tracking layer with situation awareness allows the vehicle to safely deviate from the desired global trajectory when necessary, therefore reducing the requirement for regular global trajectory re-design. Similarly, an increased emphasis on decentralised deconfliction of multiple vehicles relies on a delegation from the central planning authority to the individual vehicles. Such delegation is a key driver of increased vehicle autonomy, enabling a reduction in demands on the operator.

11.1.3 Output / Control Space Division

It was demonstrated in Chapter 8 that even a simple control scheme is able to track the continuously changing output space trajectory. This validates the concept of dividing the local motion planning and control problem into separate output and control space layers. Although a control space design may be capable of incorporating more complex vehicle performance limits, it is expected that for certain applications the additional performance fidelity will not compensate for the loss of horizon length or design rate. A high design rate and extended local horizon length are considered to be key requirements in the provision of good behaviour in complex multi-obstacle / vehicle scenarios. It is also expected that a more complex detailed control space implementation will be capable of improving the performance demonstrated in Chapter 8.

The performance limits to be enforced by the output space component become the interface between the output and control space layers. The control space component is delegated with providing the defined performance, while also handling any complexities such as control coupling or nonlinearities. This helps to isolate these effects from the optimisation based output space layer.

11.1.4 Absolute or Relative Navigation?

It has been assumed that a combination of GPS, inertial & feature recognition based positioning is able to calculate the vehicle’s current global position. However, as on board obstacle sensors will detect the location of obstacles relative to the vehicle, the critical collision avoidance functionality can be performed in a relative frame, therefore not requiring a global position fix. Errors in the global navigation solution will result in position errors from the global trajectory, which although important from a mission point of view are not safety critical. As the accuracy of the global position fix varies during a mission, so will the global position tracking error, resulting in the relative obstacle map varying about the *a priori* global obstacle map.

11.2 Comments on the Systems Engineering Framework

An unusual element of this research programme was the emphasis on the systems engineering framework. One of the primary aims of this approach was to ensure that the work was conducted with a view towards a realistic military need, and it is considered that the high level military requirement defined in Section 4.1 achieves this aim. This high level requirement was derived from a substantial review of operational experience and issues, and subsequently aided the definition of mission profiles (Section 4.2.2), capability visions (Section 4.2.3) and the detailed problem specification (Section 4.5). Creating this level of background or contextual information provides both justification and realism in the work. Additionally, creating this clear specification of the technical problem to be addressed without concern for solution techniques aided both the targeted literature search and the creation of a new solution framework to address the problem.

Consideration of the wider logical motion planning and control architecture (Section 4.3) and associated technology needs (Section 4.4) also helped to ensure that the impact of the approach on other system components was considered. The primary issues here were related to vehicle positioning and on board obstacle sensing and mapping which subsequently aids the definition of other problems to be addressed.

11.3 Recommendations for Future Research

- *Coupling of LMP framework with a 3D wind field* - As discussed in Section 10.1.
- *Optimisation of the search algorithm* - No significant effort was directed towards ensuring that the search algorithm was efficient.
- *Investigation of the impact of different design horizons* - All simulation tests presented within this thesis used a fixed time horizon of ten seconds.
- *Investigation of the impact of different curve resolutions* - All simulation tests presented within this thesis used a fixed curve resolution of 50. This value has a direct impact on computational effort, therefore it may be possible to reduce it without impacting performance.
- *Comparison of quadrotor dynamics with the 6th order Bezier curve* - No tests were conducted to verify the match between the chosen polynomial order and vehicle dynamics. It may be possible to reduce the order or the curves without sacrificing feasibility.
- *Implementation of the framework within a compiled language* - All simulation tests presented within this thesis were executed via Matlab m-scripts. Implementation within a compiled language will allow a better assessment of the likelihood of achieving real-time operation on specific hardware.
- *Implementation on a real vehicle* - It is likely that many issues may be uncovered and resolved via targeting and testing the framework on a real vehicle. An obvious candidate is a quadrotor, but other types or domains may also be suitable, i.e. UGVs. Note that implementation on a real vehicle is likely to require development of a more detailed performance map and control space component than presented within this thesis.
- *Integration with other system elements* - For example sensor / obstacle modelling, global planning, navigation solution etc.

- *Manoeuvre expectation models* - Collision avoidance in manned operation is often based on an assumed model of the feasible or likely manoeuvre limitations of the obstacle. For example, collision avoidance behaviour in road cars is heavily influenced by the driver's expectation of how the other vehicle is likely to and able to behave. A defensive strategy ensures that the reachable area is avoided completely.

- *Rapid generation of 3D steady wind field within the operational environment* - As the steady wind in an area may significantly change during the course of a mission it will be necessary to be able to estimate the 3D flow field in the operational environment reasonable rapidly, i.e. a solution generated in a matter of seconds, rather than minutes.

- *Generation of navigation solution (position fix) via blended GPS, inertial and feature recognition* - As discussed in section 4.4.2, bullet point "Detect Vehicle Location".

- *Generation of 3D obstacle / environment map* - As discussed in Section 4.4.3 "Obstacle / Environment Modelling".

- *Development of the 'associate' concept, where an unmanned vehicle is able to provide continual utility without requiring continual oversight or control.* - This was discussed briefly in section C.4 and requires a degree of autonomy that is not present in current systems.

- *Development of anti small / micro UAV capabilities* - As discussed in section C.5 'Destroy Enemy UAV'.

Appendices

A Military UAV Experience - Operational Missions

"The lessons learned from that fight [operation Iraqi Freedom] and the daily challenges our troops face indicate that UAV's are critical to both force protection and enhancing situational awareness and intelligence gathering"[39]

A.1 IED Detection

"These system enhancements have allowed the Pioneer [UAV] to have great success against the Improvised Explosive Device (IED) threat. In this mission, the primary focus is to locate IED emplacers in the act. A secondary goal is to look for suspicious objects, disturbed earth, and roadside hot or cold spots. In the last six months, Pioneer has flown 1,106 hours (approximately 35% of total hours flown) focusing on the IED threat. During that period, Pioneer located 1,140 possible IEDs, and provided over 3,400 images of these areas to supported units for investigation. The Pioneer also has supported over 50 post-IED attacks. The Electro Optical/ Infrared sensor acquires and tracks trigger men or other enemy moving to or from the IED site, detects secondary IED locations, and provides over-watch for supported units during counter-indirect fire missions.**"** - *Statement of Lieutenant General John G Castellaw, Deputy Commandant for Aviation before the Tactical Air and Land Forces Subcommittee of the House Armed Services Committee on FY 2007 Naval UAS and UCAS programs.*

A.2 Base Protection

"Our strength lies in the fact we're virtually undetected, can provide real-time video, and have the ability to strike if needed. What the insurgents don't know is we're watching things as they happen," he said. In the event of mortar attacks, the Predator can locate the origin and pass on intelligence to ground forces that can then apprehend suspects. In the past three months, approximately 25 insurgents were apprehended through Predator base defence missions, said Maj. Craig Babbitt the 46th ERS commander.**"** **-** *Extract from article published on www.af.mil, 6th March 2007, "Silent eye in the sky: Predators keep constant vigil**"** by Senior Airman Candace Romano, 332nd Air Expeditionary Wing Public Affairs*

"Today, Shadow units tied into Q-36 counter-battery radars quickly respond to inbound mortars and rockets by flying over the impact point on an outbound azimuth in order to locate the insurgents. Their integration into ground operations and flexible response capability makes them the reconnaissance, surveillance and targeting acquisition (or RSTA) system of choice for addressing areas that aren't covered by other assets at the brigade level.**"** - *Statement by US Brigadier General Jeffrey Schloesser, Director, Army Aviation Task Force Office of the Deputy Chief of Staff, Army G-3/5/7 Before the House Armed Service Committee Tactical Air & land Forces Subcommittee, United States House on US Army UAV Programmes, 9th March 2005.*

A.3 Route Reconnaissance

"During the period of the Fallujah uprising, the Hunter company placed a ground control set with the 2nd armoured cavalry regiment in Najaf. The direct-support hunter

[39] Testimony of US Major General James D Thurman, Director, Army Aviation Task Force, Office of the deputy Chief of Staff, before the House Armed Services Committee US House of Representatives Subcommittee on Tactical Air & land Forces Regarding US Army UAV programmes, March 17th, 2004

crew, working with the regimental fire-support cell, discovered insurgents preparing an ambush along the planned route of ingress. The regimental commander adjusted the ground operation accordingly. The patrol feinted a drive into the city, rather than entering Najaf as previously planned. The insurgents reacted to the feint, occupying the ambush positions. The Hunter crew guided in an AC-130 gunship, as well as mortar and Apache fires, and then confirmed the resulting destruction of the insurgent ambush and two 23mm anti-aircraft pieces, with no losses to coalition forces or equipment." - *Statement by US Brigadier General Jeffrey Schloesser, Director, Army Aviation Task Force Office of the Deputy Chief of Staff, Army G-3/5/7 Before the House Armed Service Committee Tactical Air & land Forces Subcommittee, United States House on US Army UAV Programmes, 9th March 2005.*

A.4 Over the hill, round the corner surveillance

"In response to attacks on unit patrols in the form of ambushes, mortars, rockets, and to attacks on the infrastructure, the [US] Army fielded the Raven small UAV. The Raven is a beyond line of sight collection capability employed at the patrol, platoon and company level….. For the Army, as well as the soldiers and the commanders on the ground, the Raven is a solid success story…. [for example] In Afghanistan, soldiers deployed the Raven when they learned of a possible ambush on a particularly dangerous stretch of road. Without endangering a single soldier, they were able to observe insurgents emplacing IEDs for command detonation on a part of the road that traversed a narrow pass. Based on the Raven's real-time video, the unit leaders determined the enemy position was too strong for a ground assault and ordered an air strike on the ambush site. In Iraq as well, soldiers frequently fly raven's in advance of their patrols through areas which are known for IED emplacement and ambushes. The Frequent Raven over-flights seems to deter IED placement and anti-coalition activity." - *Statement by US Brigadier General Jeffrey Schloesser, Director, Army Aviation Task Force Office of the Deputy Chief of Staff, Army G-3/5/7 Before the House Armed Service Committee Tactical Air & land Forces Subcommittee, United States House on US Army UAV Programmes, 9th March 2005.*

"This technology is changing the way we fight and we will not go without it," says one anonymous US Army commander in the Middle East. Another US Army Captain in 2nd Brigade, 1st Infantry Division, adds, "I can't stress enough the fact that those casualty numbers would have been much higher if we didn't have Ravens. One SF operator said Raven was now an "invaluable asset to Special Operations Forces." - *Extract from article in Jane's International Defence Review, 1st July 2006, "Upping the stakes: demand rises for new-generation tactical UAVs" by White, A.*

B Military UAV Experience - Operational Issues

B.1 Lack of system availability

"Not a single commander in Iraq or Afghanistan will tell you that he or she is satisfied with the amount of available UAV support. Commanders at all levels tell us that they need more access to UAV capability, from the tactical to operational level" - *Statement by US Brigadier General Jeffrey Schloesser, Director, Army Aviation Task Force Office of the Deputy Chief of Staff, Army G-3/5/7 Before the House Armed Service Committee Tactical Air & land Forces Subcommittee, United States House on US Army UAV Programmes, 9th March 2005.*

"Because of the need to reserve the airspace, the Shadow is not a quick-reaction-force (QRF) asset. Conceding that the Predator and other UAV assets will be tasked for theater-wide targets at a higher echelon of command, the real benefit, according the Shadow platoon, is the ability the UAV gives the brigade commander to get his own "eyes on target" without having to fight for airtime on other platforms like the Predator. The inability to quickly adjust the flight path in a fluid battlefield environment is compounded by the fact that UAVs are still under Air Force flight-plan constraints and requirements. Coordination with the Air Force is a time-consuming process, thus negating the potential benefit of having a tactical UAV at the brigade level. To overcome this obstacle, CPT Gourley hopes – in a best-case scenario, at least – "to have the Shadow more or less tasked to be in the air as much as possible in support of ongoing operations. The Shadow can then be re-tasked in the air to cover any contingency that might be necessary." He envisions at least one mission to "track vehicles to do area searches and road searches looking for IEDs [improvised explosive devices] and things of that sort. We're really good at route recons and smaller-level things like that." - *Extract from article published in the Journal of Electronic Defence (JED) , "Soldier Pilots" January 2006, Billingsley, D.*

"Just as with the Shadow, the biggest obstacle to using the Raven in tactical operations will potentially be its priority position in a crowded skies environment. SGT Singleton and other pilots anticipate incorporating a third soldier as a radio operator to monitor the airspace issue, because, SGT Singleton said, "every time you fly it, there is going to be restriction with the air-traffic control.

Standard operating procedure mandates that Raven operators submit a flight plan 24 hours prior to desired departure, which is two days less than the Shadow requirement. Still, the necessary protocol robs the Raven of any quick-launch capability. According to one Delta Company operator: "It is kind of the same thing we faced as forward observers. We've got to sit and wait before we can actually do something.

SGT Singleton said he thinks that "the chain of command is going to get pretty fed up, because if something happens, they're going to want [to] get the bird over there, and I think they're going to get frustrated by just how long it takes for us to get airspace coordination unless there's a way that we find out we can get airspace coordination really quick by the end. That's going to be the biggest drawback for our commanders who want something quick, and we have to actually go through all these steps to get it accomplished." - *Extract from article published in the Journal of Electronic Defence (JED) , "Soldier Pilots" January 2006, Billingsley, D.*

B.2 Airspace congestion

"As the number of UAVs continue to grow, airspace deconfliction becomes increasingly important, particularly for army helicopters that routinely operate in the same environment" - *Testimony of Major General James D. Thurman (Director, Army Aviation Task Force Office of the Deputy Chief of Staff, Army G-3) Before the House Armed Services Committee, Tactical Air and Land Forces Subcommittee, 17th March 2004.*

"The army is making improvements in UAV airspace command and control. UAVs during OIF-1 [operation Iraqi Freedom] were not consistently integrated into the airspace control measures, creating significant challenges. We are now participating in the 72hr air tasking order cycle and launching in coordination with clearance facilities that de-conflict aircraft via radar and multi-user internet relay chat rooms." - *Statement by US Brigadier General Jeffrey Schloesser, Director, Army Aviation Task Force Office of the Deputy Chief of Staff, Army G-3/5/7 Before the House Armed Service Committee Tactical Air & land Forces Subcommittee, United States House on US Army UAV Programmes, 9th March 2005.*

"Flying in a crowded skies environment is perhaps the greatest challenge to the Shadow. Without any form of aircraft-avoidance system, word among the platoon is that there has been at least one case where a UAV struck the tail of a Blackhawk helicopter in Iraq, nearly causing the helo to crash. Standard operating procedure for the Shadow is to schedule a flight 72 hours in advance, reserve a slot, and then push out. Traditionally, an operations officer at the brigade level will work out the air-tasking order" - *Extract from article published in the Journal of Electronic Defence (JED) , "Soldier Pilots" January 2006, Billingsley, D.*

B.3 Line of sight communication & frequency congestion

"Commanders were also challenged in frequency management and bandwidth. Hunter & Pioneer used the same frequencies for both uplink – which provided platform control - and downlink - which provided video. Commanders were forced to de-conflict the frequencies prior to combat operations." - *Testimony of Major General James D. Thurman (Director, Army Aviation Task Force Office of the Deputy Chief of Staff, Army G-3) Before the House Armed Services Committee, Tactical Air and Land Forces Subcommittee, 17th March 2004.*

"Like the Shadow, it [Raven] is controlled by a line-of-sight signal. It isn't possible to fly it around a corner, view what is on the other side, and then act accordingly. "If you got a building in the way, if you can't see it, then it's not going to get signal, and you might lose link," SGT Singleton warned. The Raven is equipped with an onboard internal navigation system or GPS and can be programmed to go back to a designated grid once the link is lost until line of sight is restored and signal can be regained.

The Raven also operates with a directional antenna, "so you have to actually point the antenna in the direction of the aircraft & or it is going to lose signal," according to SGT Singleton. The versions Charlie and Delta companies will field are limited to only four frequencies, "so they can be jammed if a certain frequency is used, [and the lack of frequencies] can interfere with you so you can lose link even when you're not out of range." Other pilots acknowledge that, for whatever reason, during their training, flying over water caused them to lose link with the bird." **-** *Extract from article published in the Journal of Electronic Defence (JED) , "Soldier Pilots" January 2006, Billingsley, D.*

*"*The number of UAVs flying in the Central Command theatre has exploded to at least 500, according to the latest USAF estimates, resulting in bandwidth and frequency becoming increasingly scarce commodities in Iraq and Afghanistan. UAVs consume huge amounts of bandwidth and spectrum because they are operated remotely and transmit large amounts of real-time intelligence data.

Bigger, long-endurance drones in particular exact a demanding toll on bandwidth, using their satellite uplinks to bounce a large number of signals off communication satellites.

Today, just one Global Hawk UAV consumes about 500 Mbits/s of satellite-provided bandwidth, which is more than five times the total bandwidth consumed by the entire US military during Operation 'Desert Storm'." - *Extract from article published in Jane's Defence Weekly, "USAF Eyes UAVS with Reduced Demands on Bandwidth, Spectrum", Harrington, C. 15 August 2007.*

B.4 Frustration at inability to engage targets

"Glenn tells Jane's that numerous individuals have expressed frustration at their inability to engage a target immediately after identifying it. The threat of losing a target in built-up areas, according to Glenn, is very large when he or she takes refuge in a structure, unless supporting troops have over-watch of the specific building complex. However, with multiple access into buildings, forces cannot assume that the target's entry point will be the same as his or her exit point. In addition, Glenn reports that soldiers have been frustrated by the amount of time needed to positively identify a target and engage it due to a prolonged decision-making process. "The gap between the identification of a target and the ability to engage a threat in Iraq has led to high levels of frustrations for individuals interviewed" - *Extract from Jane's Information Group article "Straight from the forces MOUT: UAVs enter the urban environment", June 2007*

"The Raven's camera generates an eight-digit grid, yet the accuracy of its imagery has been questioned, and as a result, a secondary grid has to be generated from another observation platform. According to a Delta Company operator: "It's not cleared to fire off the grid you get from a Raven, so they send somebody else out there just to verify that was actually the grid to the target, and then they could fire. They said sometimes it was accurate, sometimes it was off, but I think that will be the next generation – being able to actually get target location on something from the camera itself." - *Extract from article published in the Journal of Electronic Defence (JED) , "Soldier Pilots" January 2006, Billingsley, D.*

B.5 Desire for small UAVs

"Most of the living and dying is going on in squad, platoon and company level in this fight. So you have to give those Soldiers what they need, when they need it. And they need it all the time.

Approval chains for unmanned aerial vehicle support can be lengthy, taking time that tactical units on the ground and in the fight cannot afford. "They don't have time, when they need UAS support, to carry it up to the Joint Force Air Component Commander, ask for a Predator, and then have it go through that decision loop and then have it repositioned," Rizzi said. "They need it there, and they need it there 24/7." What Soldiers need, Rizzi said, is UAS support that is built into their combat units - unmanned aerial systems owned by the Army, flown by the Army, to provide support to the Army's ground units -- who are actually in the fight -- when they need it." - *Extracts from ASD News article, 13th Jan 2010, "Unmanned Aircraft Changing Soldiers' Battlefield Perspective" Interview with Rizzi, Glenn, A. Deputy director and senior technical advisor of the United States Army Unmanned Aircraft Systems Centre of Excellence at Fort Rucker, Ala.*

"UAVs that are wed to airfields are very tough for the tactical commander… one aircraft requires security, launch and recovery, engineers, lo support, etc. The more we can shed the overhead and make them airfield independent the more practical they will be to use" - *Testimony of Major General James D. Thurman (Director, Army Aviation Task Force Office of the Deputy Chief of Staff, Army G-3) Before the House Armed Services Committee, Tactical Air and Land Forces Subcommittee, 17th March 2004*

B.6 Engine noise

"The sound of the Shadow's Motto Guzzi engine is another concern. According to the Army, more than 20 UAVs were shot down in Kosovo in 1999 and more, including Shadows, have been downed in Iraq and Afghanistan by alert enemy ground forces. Despite these considerations, CPT Gourley is not overly concerned: "In open terrain, in the countryside, people below would hear it, but in the cities, the urban landscape, with lots of city traffic, it is unlikely that people would notice it overhead…….. However, some units within the brigade are benefiting from the noise factor. According to Shadow platoon members, psychological-operations (PSYOPS) units have recorded the sound of the Shadow and broadcast it in an effort to make the enemy think one is overhead, in an effort to deter insurgent strikes.

The Raven isn't as noisy as the Shadow, but according to Charlie and Delta Company operators, they intend to fly it between 2,000 and a few hundred feet off the deck. To do so, without detection, they have practiced turning the engine off and letting the bird glide over the target, take the image, and then turn power back on. As SGT Singleton explained: "Let's say, from a distance, you see somebody out there. Then you can just let it glide through the air. Then once it gets past whatever you're trying to look at, slowly turn the power back up for it to gain altitude again.

But one operator from Delta Company is cautious. "It just takes a lot of altitude," he said. "You've got to get real high, and then you turn off your engine, but you've got to be really close to get really good video. The lower you get, the better picture you get, the more you risk. It can only go so low before you have to turn the engines back on." - *Extract from article published in the Journal of Electronic Defence (JED) , "Soldier Pilots" January 2006, Billingsley, D.*

B.7 High number of vehicle losses

"CPT Gourley and his pilots expect some growing pains operating a new system in a hostile environment. The platoon has had limited ability to integrate the Shadow system into their brigade training operations. The UAVs were sent directly from Redstone Arsenal to the brigade's staging area in Kuwait, instead of returning to Ft. Campbell, KY, with the Shadow platoon, so they did not train a single day with the brigade prior to deployment…. labels posting a reward for the return of the UAVs to coalition forces are plastered on the sides of the Shadow and the Raven in an effort to minimize aircraft loss in the event one does go down due to hostile fire or mechanical issues." - *Extract from article published in the Journal of Electronic Defence (JED) , "Soldier Pilots" January 2006, Billingsley, D.*

C Mission Profile Descriptions

C.1 Time critical aerial reconnaissance & surveillance

A small unit of dismounted soldiers on routine patrol in an urban environment come under enemy fire. Immediate response is to take cover then attempt to assess the location and strength of the enemy force. Additional support is called for but may not be available for some time. Without breaking cover, one or more small / micro UAVs are hand launched and used to search the surrounding area. Visual or thermal imagery may be used to detect enemy troop positions, maintaining surveillance in order to monitor behaviour and help coordinate a suitable response. If the enemy troops subsequently attempt to escape then the air vehicles may be used to track individual targets, helping to coordinate the use of additional forces to intercept the targets. The key advantages brought by the small / micro UAV system are:

- Rapid response to unexpected situation providing additional situation awareness as an aid to decision making.
- Reduced likelihood of taking casualties.
- Reduced likelihood of enemy forces escaping.

A degree of system autonomy would allow the operator to retain focus on immediate situation tasks (e.g. returning fire, communicating with colleagues, improving cover, scanning for enemy movement etc.) rather than being completely head-down occupied. Automatic area search planning could also be enhanced by the use of acoustic shot location detection sensors[40], combined with knowledge of own troop positions to avoid monitoring own personnel. Automatic target detection and operator notification for classification would help to remove the continual video scanning task, with candidate targets automatically placed under surveillance while waiting operator classification. For operation within complex environments the systems would require *a priori* knowledge of building / street layout, aiding flight path planning to avoid line-of-sight restrictions.

C.2 Engage beyond line of sight (BLOS) target

[40] For example www.shotspotter.com. Also, SWATS (Soldier Worn Acoustic Targeting System) is an acoustic gunshot localisation devise worn by a dismounted soldier. Similar version also exist for vehicle deployment.

The aim of this mission is to use a small / micro UAV carrying an explosive payload to perform a kamikaze attack on an enemy target. This lethal UAV would effectively become a low cost man-portable guided weapon, enabling individual ground troops to target enemy forces that are obscured from sight, e.g. behind neighbouring buildings or on nearby rooftops. This capability is a natural extension of the UAV surveillance roles discussed in the previous sections, allowing detected targets to be attacked without line-of-sight, and therefore reducing exposure to enemy fire.

This is a significant capability increase for a ground soldier, providing a real advantage over enemy personnel limited to line-of-sight attack only. It also changes the requirements for an enemy soldier attempting to take cover, knowing that an attack may come from any direction irrespective of the location of the attacker.

Two potential modes of operation are outlined below:

- *Single UAV operation* - The operator guides the lethal UAV to the area of interest at a surveillance altitude then uses an onboard camera to detect the current target location. The operator then selects the target which the UAV begins to visually track while planning an attack approach path. The Operator authorises the attack and uses the onboard camera to watch the UAV fly the approach and attack trajectory. In this scenario the target may be moving, with continual tracking maintained by the UAV during the approach. An alternative use case is where the target location is known in advance (e.g. perhaps defined via another system) and is static then the attack trajectory may be defined in advance.
- *Two UAV operation* - The operator uses video imagery from a reconnaissance small / micro UAV (as described in appendix C.1) to designate a target. Once a target is selected the reconnaissance UAV visually tracks it, providing real-time target position updates if required. This targeting data is then used by the system to generate an attack flight path by a lethal UAV with an explosive payload. The system then maintains the currency of this plan, requests approval from the operator and waits for availability of a lethal UAV to perform the attack. The operator then enables the explosive payload on a lethal UAV and hand launches it. Once in the air the system guides the lethal UAV along the pre-approved attack trajectory.

It is likely that this application would require sustained UAV operation within an urban canyon, although this would depend on the designed attack trajectory. Even if the target is in the open then a vertical attack path at UAV airspeeds (typically low) may lead to too much advance warning, hence an increased likelihood of target evasion of the attack. A horizontal approach utilising the urban canyon for cover may be more effective. Additionally, during the launch of a UAV it may not be desirable to broadcast the position of the operator (e.g. similar effect to launching a flare). In this case the UAV trajectory may either remain below roof level for the entire mission, or at least until it is outside the immediate vicinity of the operator.

Once a trajectory has been designed for the lethal UAV the operator must be able to view & authorise it, or else request alternative plan. It may be desirable for the operator to specify an attack direction, or sections of the flight path. In this case the system will allow an operator to insert mandatory waypoints then produce a trajectory that uses them.

C.3 Urban area reconnaissance & surveillance

From a static user location, coordinate the use of one or more autonomous small / micro UAVs to perform a reconnaissance / surveillance mission for a pre-defined area prior to entry by ground forces. This mission would be used as a final intelligence gathering exercise to identify the current ground situation, for example:

- Location & movement of enemy forces.
- Location & movement of civilians.
- Distinguish between combatants & non-combatants.
- Location of roadblocks / potential ambush sites.
- Location of unidentified & potentially hazardous objects.

Each small / micro UAV would essentially be a flying video camera, allowing the system / operator to view areas of interest. A combination of different flight altitudes (e.g. 1,000ft down to street level - *platform & sensor pack dependent*) could be used to provide different capabilities as required. The higher flight regime would allow both a wider area coverage and detected targets of interest to be tracked from a safe / efficient altitude. Low level flight (e.g. close to roof top level or below, depending on the nature of the urban environment) may be required to provide the close up images or viewing angles required to help identify obscured targets, e.g. enemy forces deliberately evading detection by hiding under canopies, bridges, tunnels etc. Additionally, low level flight may help to prevent detection of the MAV by enemy forces, reducing platform vulnerability to radar detection & ground fire.

The flight paths of the air vehicles would be designed automatically (using an *a priori* 3D map of the operational environment) to allow unobstructed views along the road network, between buildings, houses, alleys, etc. as well as on roof tops. An initial higher altitude sweep could be used to rapidly identify obvious targets, areas of interest or potential targets that require closer inspection from low flying UAVs.

Coordinating the use of multiple vehicles to achieve the mission would provide several advantages. Firstly, it would allow an individual vehicle to track an identified target while other vehicles continue with the reconnaissance sweep. Having multiple assets at the disposal of the system would allow multiple separated targets to be tracked simultaneously. Alternatively, one vehicle could be used to continuously track a target, helping to direct a second vehicle in for close-up imagery. Secondly, the use of several vehicles enables a parallel search, both reducing the overall mission time and the likelihood of a target evading detection. Thirdly, each vehicle could be used as a communication relay, ensuring that low flying platforms remain in contact with the system. Finally, the robustness of the overall system would be enhanced by the ability to continue with the mission after the loss of a platform.

The mission would begin with a static plan to be followed, but is likely to become dynamic in response to the information obtained. Targets may appear / move / disappear and areas of interest will be identified. This will result in a dynamic list of tracks that need to be monitored as well as a list of ground locations that need to be examined via different viewing angles or close range imagery. This will require the system to make efficient use of the group of air vehicles at it's disposal, automatically assigning assets to individual or cooperative tasks (e.g. one UAV monitors a potentially hostile individual, providing precise location guidance to another UAV that attempts to get close up imagery to determine if they are armed). Additionally, the number of small / micro UAVs at the system's disposal may vary during a mission as endurance limits are reached or platforms are lost or re-directed to other missions. The variable nature of this situation requires a system that can regularly re-plan in response to changes in both environment and the mission state. This implies a required level of autonomy that is not present in currently fielded systems.

Ideally, the system should be useable by a single operator requiring minimal supervision. Flight path planning & control would be performed automatically, using a pre-defined area of interest, and appropriate urban mapping data. Continual monitoring of video output from all available vehicles would not be possible, therefore the system would need to be able to perform some automatic target tracking and / or classification, directing an operator towards imagery of interest.

Note that the system described above would also be useful in an non-urban environment, e.g. open countryside, where there is also a requirement to see beyond the current line of sight, e.g. over the hill, behind obstacles etc.

C.4 Patrol escort

The aim of this mission is to use multiple micro UAVs (MAVs) to perform an early warning / protection role, escorting a small unit of ground troops through a highly hostile urban area. The system could be used to perform two primary roles:

- *Line-of-Sight Enhancement* - Using the locations of patrol members and a 3D map of the area, the system will be able to calculate the current line-of-sight of the patrol. This information can then be used to direct MAVs to provide surveillance of blind spots or likely ambush locations, e.g. behind neighbouring buildings or on nearby rooftops. As the patrol moves though the urban environment the MAVs would continually re-calculate blind spots and maintain cover where required.
- *Scouting of the Designated Patrol Route* - The system could use a 3D map of the environment and the designated patrol route to perform a scouting role, inspecting the patrol route for enemy forces, groups of potentially hostile people, road blocks, traffic congestion, etc. This role is similar to the line-of-sight enhancement but acts on a longer timescale, providing early warning of evolving situations than may endanger the patrol. This scouting role could also apply to the wider area, providing early warning of approaching hostile forces from any direction.

Both roles are intended to increase the situation awareness of the ground troops, by allowing the wider environment within which the patrol is occurring to be monitored. The primary aim of the system would be to reduce the risk of ambush or encountering an unexpected situation. However, another significant benefit would be allowing the patrol to maintain surveillance of targets that emerge during a mission. Both of these aims are highly desirable, and can be provided in a limited sense by external air support if available. An intelligent MAV based system offers the potential to simultaneously house this capability at the small unit level, and extend it's utility, removing external dependence and ensuring availability when required.

The utility of such a system can be envisioned with reference to Figure 169, which shows an example of a typical urban environment that ground forces may need to patrol. It can be seen that there are many obstructions to a ground unit's line of sight, significantly limiting situational awareness and providing numerous ambush opportunities.

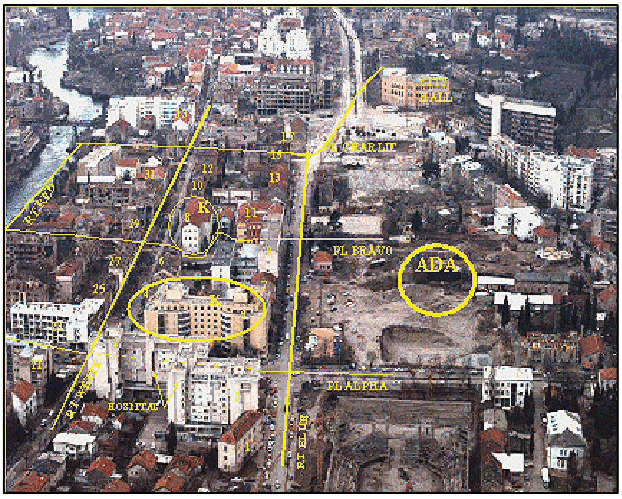


Figure 169 - Example Urban Environment

Within this patrol escort role a spectrum of utility could be provided by a MAV based system with different levels of capability. This could range from manual control and monitoring of video of a single MAV, to several autonomous UAVs acting as associates of the patrol by understanding goals and anticipating emerging needs / uses.

A system that could behave as an *associate* of the operator would be particularly useful. Associate capability implies that the system provides continual utility without requiring continual oversight, e.g. if the MAVs are commanded to act as scouting escorts of the patrol then they can then be left to operate without regular interaction. If the system subsequently detects a potential target then the operator would be notified and directed towards the imagery of interest. If no targets are detected then the operator would not be burdened by control of the system. The key advantage of this level of autonomy is that it allows the operator to remain focussed on his / her current environment, ensuring that situation awareness isn't hindered by being absorbed by video or system data.

Examples of *associate* level of capability within this context include:

- Autonomously scouting the surrounding area, detecting potential targets then bringing them to the attention of the operator.
- Detecting that the patrol is under attack and adjusting system goals accordingly, e.g. rather than carry on with scouting mission the system should attempt to locate & provide surveillance of the source of the attack (e.g. acoustic localisation[40], muzzle flash, direction of patrol fire etc.) then provide surveillance & tracking.
- Detect intended target under fire by friendly force and provide surveillance imagery, e.g. after being ambushed a patrol concentrates fire at two distinct locations. These locations are detected by the system and MAVs are assigned to provide imagery.
- Detection of new patrol route (e.g. patrol deviates from planned route due to un-expected circumstances). System uses the current patrol location & expected final destination to predict new route then provide scouting capability as before. As the patrol continues the system route prediction can be confirmed / denied / re-planned as required.
- Informing the operator if there is an attack option for a target using weapons known to be available (e.g. in range of a mortar or lethal MAV)

This associate level of capability would require the system to have an understanding of the goals and evolving needs of the patrol, allowing the system's goals to also depend on the evolving situation, e.g. anticipating the requirements of the operator. This capability implies a level of autonomy not available in current systems.

C.5 Destroy enemy UAV

The aim of this mission is to use a small / micro UAV carrying an explosive payload to use a kamikaze type attack to destroy an enemy UAV. The increasing popularity of such systems suggests that in future conflicts they are likely to also be used by hostile forces, therefore detection and destruction or neutralisation (e.g. via jamming) of such assets is likely to be a critical technology growth area.

For example, if during an operation a hostile UAV is detected maintaining surveillance over a friendly unit then it is critical that it is dealt with. It may be difficult destroy a small UAV with standard side arms, and surface-to-air or air-to-air missiles are likely to be both expensive and unavailable. A fragmentation type device may be successful for predictable flight paths, but if the UAV is moving erratically to evade damage then it may be hard to get close enough to cause significant damage. A kamikaze type small / micro UAV with an explosive payload has the critical advantage of being able to track and approach the target in real-time, ensuring that the explosive charge is not detonated until within a critical proximity of the target. If the enemy UAV detects the attack and moves away then this is also successful as the unwanted surveillance asset has been removed. Additionally, the kamikaze UAV could then be returned undamaged for use in another mission.

D Least Squares Bezier Curve-fit

A least squares curve-fit minimises the square of the difference between the data to be fitted ($\mathbf{y}$) and a model of that data (p) as shown below:

$$S = \sum_{i=1}^{m} \left(y_i - p_i(C)\right)^2$$

where

$\mathbf{S}$ = *sum of the residual error*

m = *number of data points to be fitted*

$\mathbf{y_i}$ = *data to perform curve-fit to at position i*

$p_i(C) = \sum_{j=1}^{n} C_j B_{j,i}$ = *estimate of data at position i using a polynomial data model with design* C

$B_{j,i}$ = *value of the j^th^ Bezier basis function at position i in the curve (as defined in section 5.3.2)*

C = *design coefficients of the Bezier curve =* $\begin{bmatrix} C_1 & C_2 & \dots & C_n \end{bmatrix}$

If the model being used to describe the data is linear, as with a Bezier curve, then a closed form solution is available. This solution is calculated as follows:

S is a function of the design vector C and can be minimised by setting it's partial derivatives with respect to C_i to zero.

$$\frac{\partial S}{\partial C_j} = 0 \qquad \text{for j = 1,2,...,n}$$

now,

$$\frac{\partial S}{\partial C_j} = \frac{\partial}{\partial C_j}\left(\sum_{i=1}^{m}\left(y_i^{actual} - \sum_{j=1}^{n} C_j B_{j,i}\right)^2\right) = 2\sum_{i=1}^{m}\left(y_i^{actual} - \sum_{j=1}^{n} C_j B_{j,i}\right)\left(-B_{j,i}\right)$$

Therefore the conditions for **S** being minimised can be written as:

$$\frac{\partial S}{\partial C_j} = -2B_{j,i}\sum_{i=1}^{n}\left(y_i^{actual} - \sum_{j=1}^{n} C_j B_{j,i}\right) = 0$$

This can be re-arranged to give: $(B^T B)C = B^T Y$

Therefore the design vector that minimises the least squares difference between a Bezier curve and a given data set (**y**) can be calculated from: $C = (B^T B)^{-1} B^T Y$.

Note that the term $(B^T B)^{-1} B^T = B_{least\_squared}$ is fixed by the Bezier basis functions and the chosen resolution of the curve, therefore this term only needs to be calculated once, at the start of a simulation. The Bezier curve that best fits a given data set (Y) in a least squares sense can therefore be calculated on-line by a single matrix-vector multiplication: $C = B_{least\_squared} Y$.

E Improving the Accuracy of the Target Trajectory Calculation

The primary factors that affect the accuracy of the least-squares curve-fit process include:

- *Aggressiveness of the trajectory*
- *Length of the receding horizon*
- *Order of the polynomial used for the curve-fit*

Each of these factors is discussed further below.

E.1 Aggressiveness of the trajectory

The impact of the aggressiveness of the demanded trajectory on the accuracy of the target trajectory is shown in Figure 170 & Figure 171 where the accuracy of the process is compared for a range performance limits. As the demanded trajectory becomes more aggressive the position and speed error in the target trajectory increases.

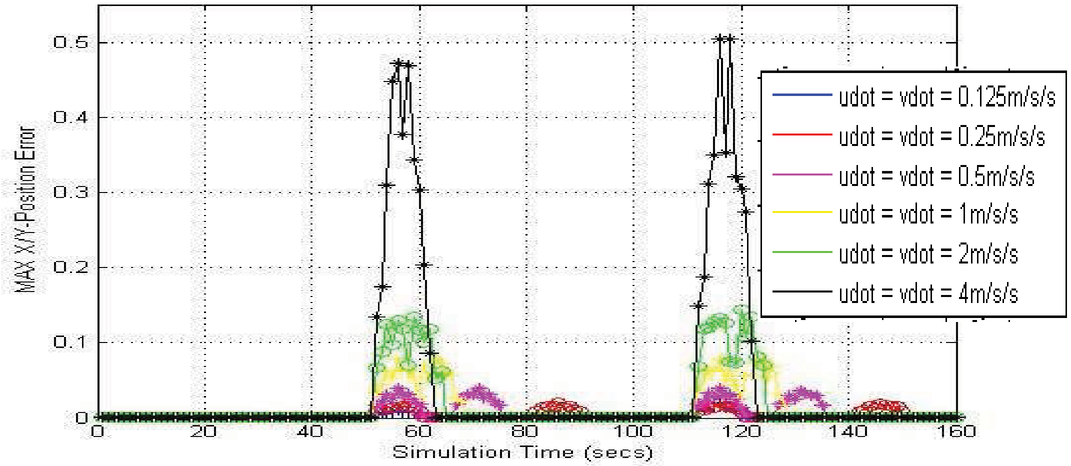


Figure 170 - Target trajectory maximum position errors

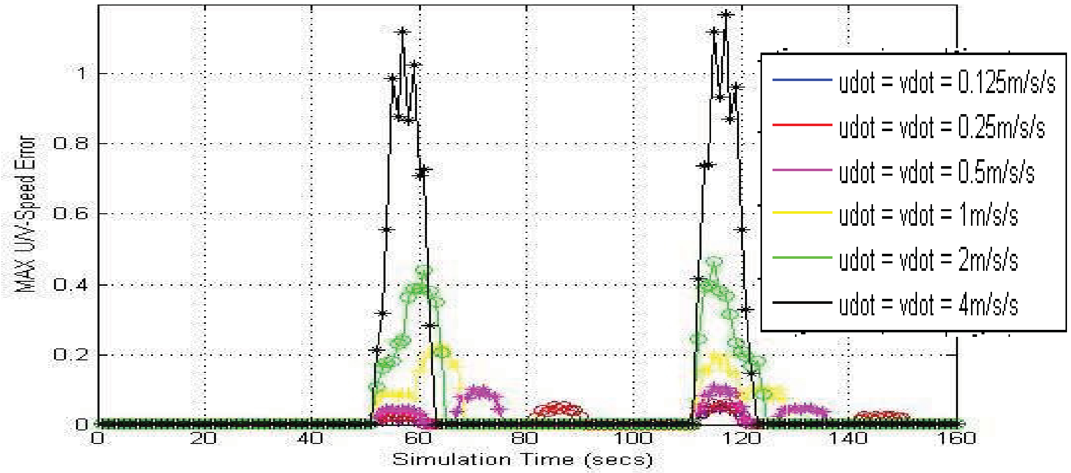


Figure 171 - Target trajectory maximum speed errors

E.2 Length of the receding horizon

Another factor which will clearly affect the accuracy of the target trajectory is receding horizon time. Increasing this time will stretch the 6th order B-curve over a longer section of the demanded trajectory, therefore reducing the accuracy of the curve-fit. Conversely, reducing the receding horizon time will impose the 6th order B-curve on a smaller section of the demanded trajectory, therefore increasing the accuracy of the curve-fit. This is validated by the results shown in Figure 172 & Figure 173 where the target trajectory errors are shown to dramatically grow with the receding horizon time.

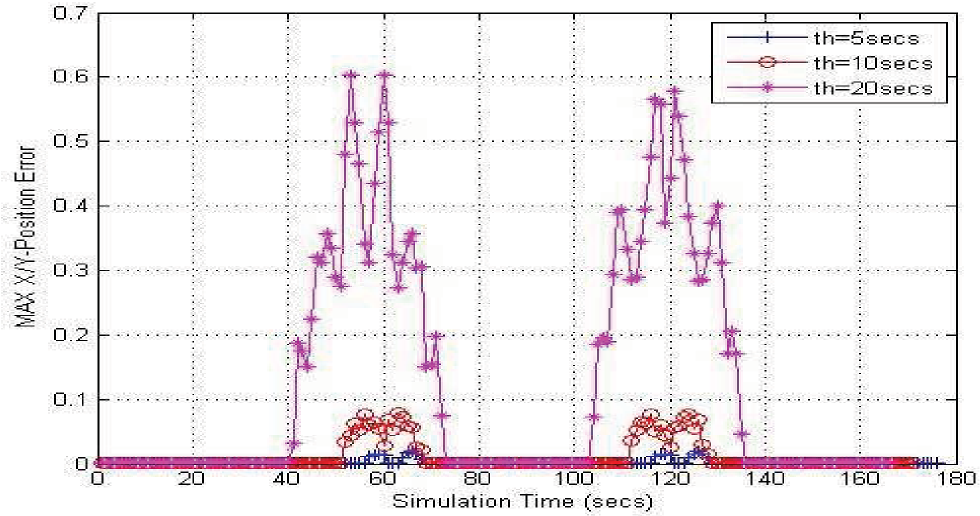


Figure 172 - Impact of horizon length on target trajectory position errors

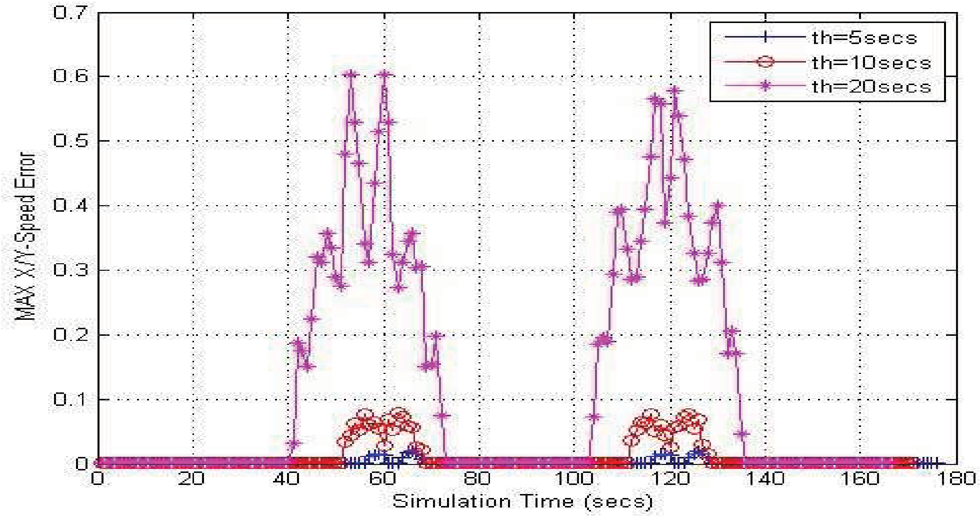


Figure 173 - Impact of horizon length on target trajectory speed errors

E.3 Order of the Bezier Polynomial

Another factor which impacts the accuracy of the target trajectory is the order of the B-curve used in the curve-fit process. This can be seen in Figure 174 & Figure 175 where the max position error doubles from approx. 0.06m for a 7th order curve to 0.12 for a 5th order curve. The max speed errors show a similar pattern, but are not as strongly affected.

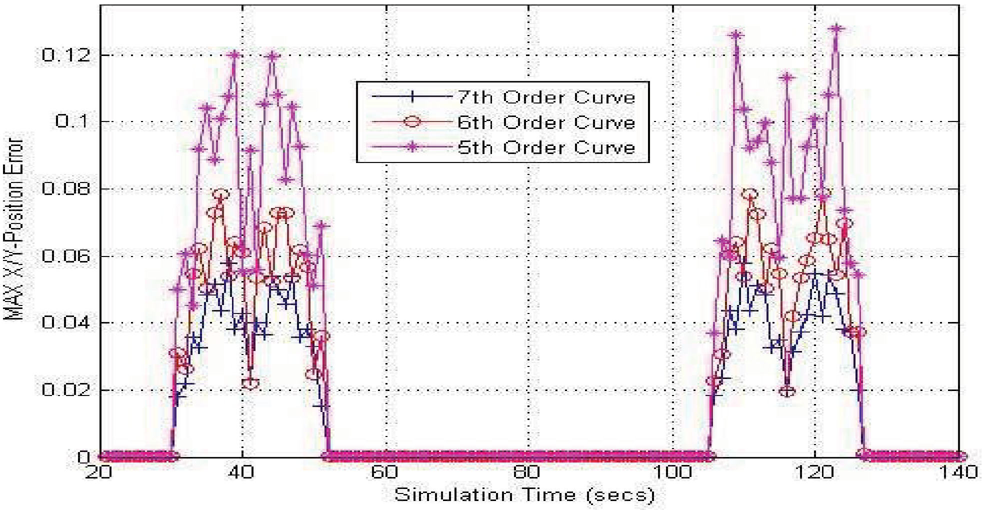


Figure 174 - Impact of curve order on target trajectory position errors

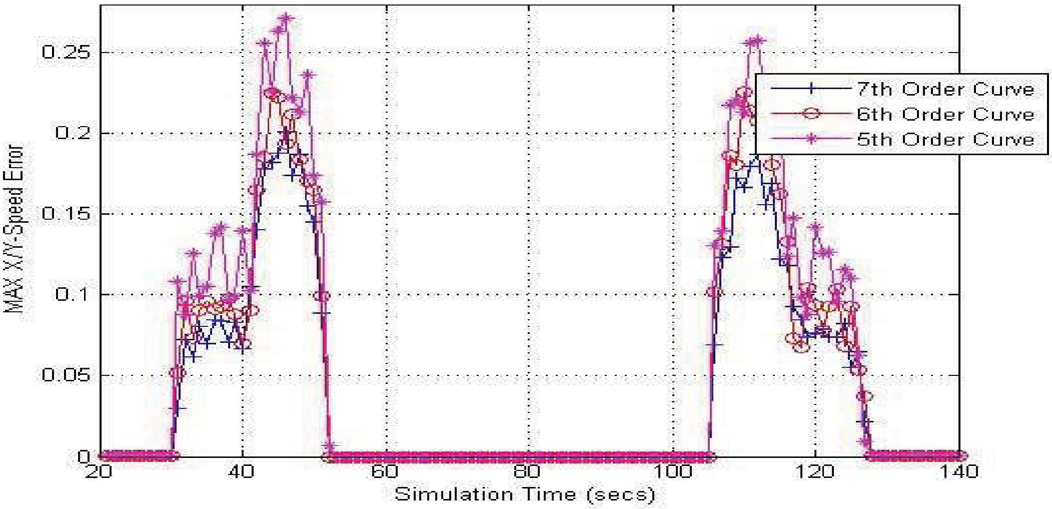


Figure 175 - Impact of curve order on target trajectory speed errors

F Examples of Current Small Scale Processing Power

F.1 Computer on Module (COM) Units

COM units are self contained single-board computers, often optimised for low power / cost embedded system applications. A range of processing speeds and memory options are available, along with standard operating system support such as Linux, VxWorks, Windows etc. Examples of COM units include:

- *Gumstix Overo series (Linux support)* - size = 58x17x4.2mm, processor = 600MHz, memory = 256MB flash & 256MB RAM
- *Gumstix Verdex pro (Linux support)* - size = 80x20x5.3mm, processor = 600MHz, memory = 32MB flash & 128MB RAM
- *Kontron nanoETXexpress-SP* - size = 84x55mm, processor = 1.6GHz (Intel Atom), memory = 8GB flash & 2GB RAM
- *Kontron microETXexpress-PC* - size = 95x95mm, processor = 1.86GHz (Intel Core 2 Duo), memory = 4GB RAM (connection to external hard drive available)

- *Kontron EB405 (optimised for cost sensitive applications)* - size = 115x75mm, processor = 266MHz (PowerPC), memory = 32MB flash & 256MB RAM
- *Kontron EB8540 (high performance, VxWorks, Linux support)* - size = 115x75mm, processor = 800MHz (PowerPC), memory = 32MB flash & 256MB RAM
- *Xbow Stargate* - size = 89x64mm, processor = 400Mhz, memory = 32MB flash & 64MB RAM
- *Congatec conga-QA (designed for low power consumption)* - size = 70x70mm, processor = 1.6GHz (Intel Atom)
- *ADLINK nanoX-ML* - size = 84x55mm, processor = 1.6GHz (Intel Atom), memory = 1-8GB disk & 512MB RAM
- *ADLINK express-ATC* - size = 95x95mm, processor = 1.6GHz (Intel Atom), memory = 1-8GB disk & 2GB RAM

F.2 COTS Small / Micro UAV Autopilots

In addition to the processing capability that can be obtained via COM units, a range of COTS UAV autopilots are also available. These units tend to be custom designed processor modules that provide a set of standard flight control modes including vehicle stabilisation, altitude hold, heading demand, auto-land, waypoint following etc. These control loops may be tuned to suit the particular vehicle, and all necessary sensors (e.g. MEMS gyros / accelerometers, pressure altitude / airspeed, ultrasonic altitude etc.) and communications components are typically also integrated. The vehicle can then be controlled using standard ground based software. Example products include:

- *MicroPilot MP2028^{xp} (low end product for disposable UAVs)* - weight = 28g, size = 100x40x15mm, on-board sensors = GPS receiver (1Hz), 3-axis gyros / accelerometers, pressure altimeter and airspeed, telemetry & data-logging up to 1.5MB
- *MicroPilot MP2128^{Heli} (high end product for fixed wing and VTOL UAVs)* - weight = 28g, size = 100x40x15mm, on-board sensors = GPS receiver (4Hz), 3-axis gyro / accelerometers, ultrasonic altitude, pressure altimeter and airspeed, 12state Kalman filter, telemetry & data-logging
- *Procerus Kestrel* - size = 51x35x12mm, weight = 16.7g, processor = 29MHz, memory = 512k flash & 512kRAM, sensors = 3-axis rate gyros and accelerometers, pressure altitude and airspeed
- *Cloud Cap Piccolo LT* - size = 130x59x19mm, weight = 110g (including 900MHz radio), processor = 40MHz, memory = 448k
- *Blue Bear Systems Research SNAP* - Processor = 400MHz, 36g, on-board data-logging, Matlab / SIMULINK interface, communicates direct to other SNAP autopilots allowing creation of ad-hoc mobile networks, i.e. coordinated / cooperative control of multiple unmanned vehicles. 3 Gyros, 3 accelerometers, external pressure & temperature.

References

1. Andert, F., Adolf, F., Goormann, L. & Dittrich, J.S. (2010). Autonomous Vision-Based Helicopter Flights through Obstacle Gates. *Journal of Intelligent & Robotic Systems*. Volume 57. Numbers 1-4. P. 259-280.
2. Baxter, J.W., Horn, G.S., Leivers, D.P. (2008). Fly-by-Agent: Controlling a Pool of UAVs via a Multi-Agent System. *Applications and Innovations in Intelligent Systems XV*. Volume 6. P. 219-230. DOI: 10.1007/978-1-84800-086-5_16.
3. Bellingham, J.S., Richards, A. & How, J.P. (2002). Receding Horizon Control of Autonomous Vehicles. *Proceedings of the American Control Conference*. P. 3741-3746.
4. Beyeler, A., Zufferey, J.C. & Floreano, D. (2009). Vision-based control of near-obstacle flight. *Autonomous Robots*. Vol. 27, pp. 201-219.
5. Blocken, B. & Carmeliet, J. (2004). Pedestrian Wind Environment around Buildings: Literature Review and Practical Examples. *Journal of Thermal Envelope and Building Science*. vol. 28, no. 2, pp. 107-159.
6. Brindley, G., Bradley, R. (1996). Turbulence Modeling in the Presence of Large Structures for Application to Flight Simulation. *Technical report by Glasgow Caledonian University on behalf of the UK Defense Research Agency*.
7. US Department of Defence. (2009). *FY2009-2034 Unmanned Systems Integrated Roadmap*.
8. Conte, G., Duranti, S. & Merz, T. (2004). Dynamic 3D Path Following for an Autonomous Helicopter. *In Proc. of the 5th IFAC Symposium on Intelligent Autonomous Vehicles*.
9. Cowling, I.D., Whidborne, J.F. & Cooke, A.K. (2006). MBPC for autonomous operation of a quadrotor air vehicle. *21st International Conference on Unmanned Air Vehicle Systems*. pp 35.1-35.9, Bristol, UK.
10. Cowling, I.D., Whidborne, J.F. & Cooke, A.K. (2006). Optimal Trajectory Planning and LQR Control for a Quadrotor UAV. *UKACC International Conference on Control*.
11. Cowling, I.D., Yakimenko, O.A., Whidborne, J.F. & Cooke, A.K. (2007). A Prototype of an Autonomous Controller for a Quadrotor UAV. *Proceedings of the European Control Conference*. P. 4001-4008.
12. Dippold, A., Ma, L. & Hovakimyan, N. (2009). Vision-Based Obstacle Avoidance of Wheeled Robots Using Fast Estimation. *Journal of Guidance, Control & Dynamics,* vol. 32, no. 6, pp. 1931-1937.
13. Eele, A. & Richards, A. (2009). Path-Planning with Avoidance Using Nonlinear Branch-and-Bound Optimization. *Journal of Guidance, Control & Dynamics,* vol. 32, no. 2, pp. 384-385-394.
14. Eklunk, J.M., Sprinkle, J. & Sastry, S. (2005). Implementing and Testing a Nonlinear Model Predictive Tracking Controller for Aerial Pursuit Evasion Games on a Fixed Wing Aircraft. *American Control Conference*.
15. Eurocontrol. (2010). Airborne Collision Avoidance System (ACAS)". Retrieved 20th December 2010 from Eurocontrol website: http://www.eurocontrol.int/msa/public/standard_page/ACAS_Startpage.html.
16. Fontana, R. J. (2004). Recent System Applications of Short Pulse Ultra-Wideband (UWB) Technology. *IEEE Transactions on Microwave Theory and Techniques.* Vol. 52, NO. 9p2087-p2104.
17. Frazzoli, E., Dahleh, M. & Feron, E. (1999). A Hybrid Control Architecture for Agressive Manoeuvering of Autonomous Helicopters. *38th IEEE Conference on Decision & Control*.
18. Frazzoli, E., Mao, Z., H., Oh, J., H. & Feron, E. (2001). Resolution of Conflicts Involving Many Aircraft via Semidefinite Programming. *Journal of Guidance, Control & Dynamics,* vol. 24, no. 1, pp. 79-80-86.
19. Frew, E.W. (2005). Receding Horizon Control Using Random Search for UAV Navigation with Passive, Non-Cooperative Sensing. *AIAA Guidance, Navigation, and Control Conference and Exhibit*.
20. Gates, D., J. (2009). Properties of a Real-Time Guidance Method for Preventing a Collision. *Journal of Guidance, Control & Dynamics.* vol. 32, no. 3, pp. 705-716.
21. Gu, D., Kamal, W.A. & Postlethwaite, I. (2004). A uav waypoint generator", *AIAA 1st Intelligent Systems Technical Conference*.
22. Gu, D., Postlethwaite, I. & Kim, Y. (2005). A Comprehensive Study on Flight Path Selection Algorithms. *Appears in Target Tracking: Algorithms and Applications, 2006. The IEE Seminar on (Ref. No. 2006/11359). P 77-90.*

23. Hamner, B., Singh, S. & Roth, S. (2008). An efficient system for combined route traversal and collision avoidance. *Autonomous Robots.* Vol. 24, no. 4, pp. 365-385.
24. Harbick, K., Montgomery, J.F. & Sukhatme, G.S. (2001). Planar Spline Trajectory Following for an Autonomous Helicopter. *IEEE International Symposium on Computational Intelligence in Robotics and Automation*.
25. Hoekstra, J.M., Gent, R.N.H.W. & Ruigrok, R.C.J. (2002). Designing for Sefety: the 'free flight' air traffic management concept. *Reliability Engineering and System Safety,*pp. 215-232.
26. Howard, T. & Kelly, A. (2005). Trajectory Generation on Rough Terrain Considering Actuator Dynamics. *5th International Conference on Field and Service Robotics*.
27. Howard, T., Knepper, R.A. & Kelly, A. (2006). Constrained Optimization Path Following of Wheeled Robots in Natural Terrain. *10th International Symposium on Experimental Robotics*.
28. Howlett, J.K., Whalley, M., Tsenkov, P., Schulein, G. & Takahashi, M. (2007). Flight Evaluation of a System for Unmanned Rotorcraft Reactive Navigation in Uncertain Urban Environments. *Annual forum proceedings – American Helicopter Society*.
29. Hrabar, S. (2008). 3D Path Planning and Stereo-based Obstacle Avoidance for Rotorcraft UAVs. *IEEE/RSJ International Conference on Intelligent Robots and Systems*.
30. Hrabar, S. & Sukhatme, G. (2009). Vision-based Navigation through Urban Canyons. *Journal of Field Robotics*. Vol. 26, no. 5, pp. 431-452.
31. Hu, J., Prandini, M. & Sastry, S. (2002). Optimal Coordinated Manoevers for Three-Dimensional Aircraft Conflict Resolution. *Journal of Guidance, Control & Dynamics*. Vol. 25, no. 5, pp. 888-889-900.
32. Izadi, H., A., Gordon, B., W. & Zhang, Y. (2009). Decentralized Receding Horizon Control for Cooperative Multiple Vehicles Subject to Communication Delays. *Journal of Guidance, Control & Dynamics*. Vol. 32, no. 6, pp. 1959-1965.
33. Kamal, W.A., Gu, D. & Postlethwaite, I. (2005). MILP and its Application in Flight Path Planning", *16th IFAC World Congress*.
34. Kamal, W.A., Gu, D. & Postlethwaite, I. (2005). A Decentralised Probabilistic Framework for the Path Planning of Autonomous Vehicles. *IFAC World Congress, Prague.*
35. Karatasou, S., Santamouris, M., Geros, V. (2006). Environmental Design of Urban Buildings – An Integrated Approach. *Earthscan*. ISBN-10: 1-902916-42-5 (hardback)
36. Kendoul, F., Fantoni, I. & Nonami, K. (2009). Optic flow-based vision system for autonomous 3D localization and control of small aerial vehicles. *Autonomous Robots*. Vol. 57, no. 6-7, pp. 591-602.
37. Keviczky, T., Vanek, B., Borrelli, F. & Balas, G.J. (2006). Hybrid decentralized receding horizon control of vehicle formations.
38. Kim, H.J., Shim, D. & Sastry, S. (2002). Nonlinear model predictive tracking control for rotorcraft-based unmanned aerial vehicles", *American Control Conference*.
39. Kuwata, Y. & How, J.P. (2004). Three Dimensional Receding Horizon Control for UAVs, *AIAA Guidance, Navigation, and Control*.
40. LaValle, S.M. (2002). From Dynamic Programming to RRTs: Algorithmic Design of Feasible Trajectories. *In Control Problems in Robotics.* P19-37. Springer-Verlag, Berlin .
41. LaValle, S.M. (1998). Rapidly-exploring random trees: A new tool for path planning. TR 98-11, Computing Science Dept., Iowa State University, October 1998.
42. Li, X., Liu, C. & Leung, D. (2008). Large-eddy simulation of flow and pollutant dispersion in high aspect ratio urban street canyons with wall model. *Boundary Layer Meteorology*, vol. 129, pp. 249-268.
43. Martinez, V., (2007). Modelling of the Flight Dynamics of a Quadrotor Helicopter. MSc Thesis, Cranfield University.
44. McConley, M., Piedmonte, M.D., Appleby, B., Frazzoli, E., Feron, E. & Dahleh, M. (2000). Hybrid Control for Aggressive Manoeuvering of Aurtonomous Aerial Vehicles. *19th Digital Avionics Systems Conference*.
45. Nagy, B. & Kelly, A. (2001). Trajectory Generation for Car-Like Robots using Cubic Curvature Polynomials. *Field and Service Robots*.
46. National Audit Office. (2009). The Major Projects Report 2009.
47. Neto, A.A., Macharet, D.G. & Campos, M.F.M. (2010). On the Generation of Trajectories for Multiple UAVs in Environments with Obstacles. *Journal of Intelligent & Robotic Systems,* vol. 57, no. 1-4, pp. 123-141.
48. Oke, T., R., (1988). Street Design and Urban Canopy Climate", *Energy & Buildings*, 11 pp. 103-113.
49. Patron, P., Evans, J. & Lane, D. (2006). Fault Tolerant Autonomous Decision Making for Unmanned Vehicles. *1st SEAS DTC Technical Conference*.

50. Peterka, J., A., Meroney, R., N. & Kothari, K., M. (1985). Wind Flow Patterns About Buildings. *Journal of Wind Engineering and Industrial Aerodynamics*. Vol. 21, pp. 21-38.
51. Petit, N., Milam, M. & Murray, R. (2001). Inversion Based Constrained Trajectory Optimization. *5th IFAC symposium on nonlinear control systems*.
52. Pettersson, P.O. & Doherty, P. (2006). Probabilistic roadmap based path planning for an autonomous unmanned helicopter. *Journal of Intelligent and Fuzzy Systems*. Vol. 17, Number 4/2006, pp. 395-10.
53. Pettit, R.L. & Homer, M.L. (2004). An Autonomous Threat Evasion Response Algorithm for Unmanned Air Vehicles During Low Altitude Flight. *AIAA 1st Intelligent Systems Technical Conference*.
54. Prandini, M., Hu, j., Lygeros, J. & Sastry, S. (2000). A Probabilistic Approach to Aircraft Conflict Detection. *IEEE Transactions on Intelligent Transportation Systems,* vol. 1, no. 4, pp. 199-220.
55. Rao, S.S. (1996). Engineering Optimization, Theory and Practice. John Wiley & Sons.
56. Rezgui, D., Lowenberg, M. & Bunniss, P. (2006). Design Concept for Stabilizing and Controlling Rotary Wing Micro Air Vehicles in Gusts. *UAV Systems - 21st International Conference*.
57. Ricciardelli, F. & Polimeno, S. (2006). Some characteristics of the wind flow layer in the lower urban boundary layer. *Journal of Wind Engineering and Industrial Aerodynamics*. pp. 815-832.
58. Richards, A. (2008). Adaptive Navigation in an Uncertain Urban Environment. *SEAS DTC Technical Conference*.
59. Richards, A. & How, J.P. (2002). Aircraft Trajectory Planning with Collision Avoidance using Mixed Integer Linear Programming. *American Control Conference*.
60. Richards, N., Sharma, M. & Ward, D. (2004). A Hybrid A*/Automaton Approach to On-Line Path Planning with Obstacle Avoidance. *AIAA 1st Intelligent Systems Technical Conference*.
61. Rotach, M., W. & et al (2005). BUBBLE - an Urban Boundary Layer Meteorology Project. *Theoretical and Applied Climatology*. Vol. 81, pp. 231-261.
62. Rubio, J.C., Vagners, J. & Rysdykz, R. (2004). Adaptive Path Planning for Autonomous UAV Oceanic Search Missions. *AIAA 1st. Intelligent. Systems. Technical Conference*.
63. Scherer, S., Singh, S., Chamberlain, L., J. & Saripalli, S. (2007). Flying Fast and Low Among Obstacles. *Proceedings of the International Conference on Robotics and Automation*.
64. Schouwenaars, T., Valenti, M., Feron, E. & How, J.P. (2005). Implementation and Flight Test Results of MILP-based UAV Guidance. *IEEE Aerospace Conference*.
65. Shim, D., Chung, H., Kim, H.J. & Sastry, S. (2005). Autonomous Exploration in Unknown Urban Environments for Unmanned Aerial Vehicles. *AIAA GN&C Conference*.
66. Shim, D., Chung, H. & Sastry, S. (2006). Conflict-free navigation in unknown urban environments. *IEEE Robotics and Automation Magazine*. Vol. 13, no. 3, pp. 27-33.
67. Shim, D., Kim, H.J. & Sastry, S. (2003). Decentralized nonlinear model predictive control of multiple flying robots. *IEEE Conference on Decision and Control*.
68. Shim, D. & Sastry, S. (2007). An Evasive Maneuvering Algorithm for UAVs in See-and-Avoid Situations. *American Control Conference*.
69. Shim, D. & Sastry, S. (2006). A Situation-aware Flight Control System Design using Real-time Model Predictive Control for Unmanned Autonomous Helicopters. *AIAA Guidance Navigation & Control Conference*.
70. Singh, L., Yang, L., McConley, M. & Appleby, B. (2005). Autonomous Guidance and Control for Agile UAV Maneuvering. *Annual Formum Proceedings - American Helicopter Society*.
71. Strens, M., Baxter, J. & Gardner, A. (2007). A Generic Architecture for Hybrid Autonomous Decision Systems. *2nd SEAS DTC Technical Conference*.
72. Strens, M., Kapetanakis, S. & Baxter, J. (2006). Towards a Generic Architecture for Multi-Vehicle Autonomy. *1st SEAS DTC Technical Conference*.
73. Teal Group Corporation (2010). *World Umanned Aerial Vehicle Systems - 2010 Market Profile and Forecasts*.
74. Team MIT 2007, 11/20/07-last update*, Team MIT - Technical Report — DARPA Urban Challenge*. Available: http://www.darpa.mil/GRANDCHALLENGE/resources.asp.
75. Tether, T. (Director US Defence Advanced Projects Agency), Statement to the Subcommittee on Terrorism, Unconventional Threats and Capabilities House Armed Services Committee, United States House of Representatives, March 21st 2007

76. Thompson, S. & Kagami, S. (2005). Continuous Curvature Trajectory Generation with Obstacle Avoidance for Car-Like Robots. *Computational Intelligence for Modelling, Control and Automation*.
77. Urmson, C. & et al (2007). 11/20/07-last update*, Tartan Racing: A Multi-Modal Approach to the DARPA Urban Challenge*. Available: http://www.darpa.mil/GRANDCHALLENGE/resources.asp.
78. Wood, C.,R., Arnold, SJ, Balogun, A.,A., Barlow, JF, Belcher, S.,E., Britter, R.,E., Cheng, H., Dobre, A., Lingard, J.,J.,N., Martin, D., Neophytou, M., Petersson, F.,K., Robins, A.,G., Shallcross, D.,E., Smalley, R.,J., Tate, J.,E., Tomlin, A.,S. & White, I.,R. (2009). Dispersion experiments in central London: the 2007 DAPPLE project. *Bulletin of the American Meteorological Society*, 90:955–969
79. Wzorek, M., Conte, G., Rudol, P., Merz, T., Duranti, S. & Doherty, P. (2006). From Motion Planning to Control - A Navigation Framework for an Autonomous Unmanned Aerial Vehicle. *21th Bristol UAV Systems Conference*.
80. Yadav, V., Wang, X. & Balakrishnan, S.N. (2006). Neural network approach for obstacle avoidance in 3-D environments for UAVs. Minneapolis, MN edn.
81. Yang, K., Gan, S.,K. & Sukkarieh, S. (2010). An Efficient Path Planning and Control Algorithm for RUAVs in Unknown and Cluttered Environments. *Journal of Intelligent & Robotic Systems*. Vol. 57, no. 1-4, pp. 101-102-122.
82. Yoon, Y., Shin, J., Kim, H.J., Park, Y. & Sastry, S. (2009). Model-predictive active steering and obstacle avoidance for autonomous ground vehicles. *Control Engineering Practice*. Vol. 17, no. 7, pp. 741-742-750.