Guidance and Control Using Model Predictive Control for Low Altitude Real-Time Terrain Following Flight

by

Tiffany Rae Lapp

B.S. Aeronautical and Astronautical Engineering
University of Washington, 2001

SUBMITTED TO THE DEPARTMENT OF AERONAUTICS AND ASTRONAUTICS
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE IN AERONAUTICS AND ASTRONAUTICS

AT THE
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

JUNE 2004

Signature of Author: _____
Department of Aeronautics and Astronautics
May 14, 2004

Certified by: _____
Dr. Leena Singh
Charles Stark Draper Laboratory, Inc.
Thesis Supervisor

Certified by: _____
Dr. Brent Appleby
Lecturer, Department of Aeronautics and Astronautics
Thesis Advisor

Accepted by: _____
Edward M. Greitzer
H.N. Slater Professor of Aeronautics and Astronautics
Chairman, Committee on Graduate Students

[This Page Intentionally Left Blank]

Guidance and Control Using Model Predictive Control for Low Altitude Real-Time Terrain Following Flight

by

Tiffany Rae Lapp

Submitted to the Department of Aeronautics and Astronautics on May 14, 2004 in
Partial Fulfillment of the Requirements for the Degree of
Master of Science in Aeronautics and Astronautics

ABSTRACT

This thesis presents the design and implementation of a model predictive control based trajectory optimization method for Nap-of-the-Earth (NOE) flight. A NOE trajectory reference is generated over a subspace of the terrain. It is then inserted into the cost function and the resulting trajectory tracking error term is weighted for more precise longitudinal tracking than lateral tracking through the introduction of the TF/TA ratio. The TF/TA ratio, control effort penalties and MPC prediction horizon are tuned for this application via simulation and eigenvalue analysis for stability and performance. Steps are taken to reduce complexity in the optimization problem including perturbational linearization in the prediction model generation and the use of control basis functions which are analyzed for their trade-off between approximation of the optimal cost/solution and reduction of the optimization complexity. Obstacle avoidance including preclusion of ground collision is accomplished through the establishment of hard state constraints. These state constraints create a 'safe envelope' within which the optimal trajectory can be found. Results over a variety of sample terrains are provided to investigate the sensitivity of tracking performance to nominal velocities. The mission objective of low altitude and high speed was met satisfactorily without terrain or obstacle collision, however, methods to preclude or deal with infeasibility must be investigated as terrain severity (measured by commanded flight path angle) is increased past 30 degrees or speed is increased to and past 30 knots.

Thesis Supervisor:	Dr. Leena Singh
Title:	Senior Member of the Technical Staff, The Charles Stark Draper Laboratory, Inc.
Thesis Supervisor:	Dr. Brent Appleby
Title:	Lecturer, Department of Aeronautics and Astronautics Massachusetts Institute of Technology Principle Member of the Technical Staff, The Charles Stark Draper Laboratory, Inc.

[This Page Intentionally Left Blank]

ACKNOWLEDGEMENTS

I would like to express my appreciation to several people who have been key to my success here at Draper and MIT.

Firstly I would like to thank Dr. Leena Singh. I am truly grateful for her guidance, insight, attention and encouragement despite her hectic schedule.

I would like to thank Dr. Brent Appleby for making the time between his busy schedule and travel to do whatever he could to push this project forward, enable my research, and to offer helpful suggestions for this thesis.

Additionally, I would like to thank Pierro Miotto and Fred Boelitz for their willingness to take a moment to answer my questions and provide additional points of view throughout this process.

I would like to thank my friends and family without whom I would have never even gotten into this place, let alone survived it. Thank you to my family who has encouraged me to constantly push beyond my own limits and believed in me every step of the way. Your prayers have gotten me through.

Thank you to all of my friends in the Graduate Christian Fellowship, you are truly my brothers and sisters and I am proud to be a part of your family.

Thank you to all of my fellow Draper students in the Hill building, you have made the endless hours of studying, research and thesis writing bearable. May none of us have to work weekends at our future jobs.

A special thank you to my fiancé, Ryan, for believing in me even through the hard times, helping me when I refused help, keeping me sane and just plain putting up with me these past two years. You are my best friend.

Finally, I would like to thank Draper Laboratory for the opportunity to research such an interesting topic and continue my education. It has been a pleasure to work here.

This thesis was prepared at The Charles Stark Draper Laboratory, Inc., under Internal Company Sponsored Research Project #15254, Model Predictive Control, Contract Number IRD03-2-5038 and Internal Company Sponsored Research Project #13202, Enhanced Air Vehicle Flight Code, Contract Number IRD04-0-6048-1.

Publication of this thesis does not constitute approval by Draper or the sponsoring agency of the findings or conclusions contained therein. It is published for the exchange and stimulation of ideas.

Tiffany R. Lapp Date

[This Page Intentionally Left Blank]

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

[This Page Intentionally Left Blank]

Chapter 1

INTRODUCTION

Unmanned aerial vehicles (UAVs) have played an important role throughout history dating back to the flying bombs of WWI and WWII. Today their role has expanded from the early payload delivery missions to include remote sensing, providing valuable geographic, meteorological and ecological data for scientists and military alike [1]. Recent examples of deployment include reconnaissance and battle damage assessment which enhances battlefield awareness of commanders by delivering a combination of camera and sensor data without risking pilots' lives [2]. Current research in the area of UAV design and control is focused on exploiting the lack of human casualty inherent in UAV deployment to operate in environments that would otherwise be too dangerous for human piloted vehicles.

One such application is the use of UAV's for operation within high threat, hostile military environments. A hostile operational environment is defined by the Department of Defense as "an operational environment in which hostile forces have control as well as the intent and capability to effectively oppose or react to the operations a unit intends to conduct," [3]. It is inherently obvious that in these environments, as threat exposure increases, so does the probability of vehicle attrition. Because of this, a primary research objective for unmanned aerial vehicles is enhancement of vehicle survivability. Reducing threat line of sight reduces the probability of detection, enhancing vehicle survivability. To effectively lessen threat line of sight, suitable guidance and control algorithms can

take advantage of terrain masking through Nap-of-the-Earth (NOE) flight while simultaneously flying as fast as possible to enhance vehicle survivability if detected.

NOE flight or terrain following flight is defined as "flight close to the Earth's surface during which airspeed, height, and/or altitude are adapted to the contours and cover of the ground in order to avoid enemy detection and fire," [3]. In seeking cover of the ground to evade detection, cluttered and dangerous terrain coupled with the high-speed NOE flight objective translate to a highly constrained control problem. It is assumed that at such low altitudes, the vehicle will be operating with an incomplete obstacle map. Even the highest resolution map may not be able to warn of unexpected obstacles such as buildings, trees and wires or threats such as snipers. This type of information can only be obtained through real-time sensor sweeps during flight. This motivates the need for an efficient algorithm for introducing new obstacle knowledge obtained from the sensors into the trajectory optimization real-time to allow dynamic path replanning. In addition, due to the vehicle's proximity to many constantly changing constraints, we require agile maneuvering to move quickly between different ranges of its flight envelope.

1.1 LITERATURE REVIEW

The terrain following guidance and control problem has been an area of research interest for the past 20 years, however due to both technological and political reasons, this interest has been especially amplified in the past 5 years. This section surveys some of the previous work that has been done in the field of trajectory generation and obstacle avoidance as applied to the problem of terrain following/nap-of-the-earth flight.

Low-altitude, terrain following flight requires a high degree of continuous situational awareness equating to high workload for pilots. The reduction of pilot workload under these conditions through the automation of terrain following flight and obstacle avoidance was the premise for several investigations into nap-of-the-earth flight at NASA Ames Research Center [4 – 6]. The terrain following problem was treated successfully in these applications using heuristics-based flight-path planning. Obstacle avoidance was accomplished through reference trajectory modification by placing a temporary waypoint to divert the current nominal trajectory around the obstacle. This research emphasized that at the low altitudes required by NOE flight, explicit consideration of vehicle

capability is essential in the autopilot design to assure safe flight in such close proximity to the ground regardless of the level of autonomy being explored.

Another body of literature focuses on the fully autonomous trajectory generation and tracking realm. To obtain the high accuracy tracking performance crucial to survival at the low altitudes defining NOE flight, optimal control is a prime candidate for trajectory generation. Another important attribute of optimization based control is that vehicle capability limits can be input directly into the problem by adding the vehicle dynamic constraints. However the high computational burden presented by optimal control is a known drawback to using this control strategy. To reduce computational burden on the optimizer and incorporate the "look ahead" that a human pilot is able to provide, predictive control was first proposed as a solution to the terrain following control problem in 1989 by Hess and Jung [7]. They applied generalized predictive control (GPC), which generates control outputs by minimizing a quadratic cost function consisting of a weighted sum of errors between desired and predicted system output and predicted controls within a finite horizon. The cost function was weighted to track a simulated elevation reference comprised of a sum of sines while simultaneously tracking a longitudinal velocity reference and rejecting unknown turbulence injections. This algorithm, integrated with an online estimator of the internal model's parameters, accomplished the desired tracking with maximum errors totaling 10% of the maximum errors seen with the "classical" multi-loop design presented in their work.

The success of Hess and Jung has motivated many subsequent applications of various forms of predictive control reaching beyond the terrain tracking problem to issues of trajectory optimization and obstacle avoidance [8 - 12]. Li et. al. applied an optimal preview control scheme to the terrain tracking of a cruise missile [8]. The method presented considered tracking error, the current command signal and an additional augmented error system. This augmented error allowed future command signals to be included in the optimization through a linear quadratic index, allowing the problem to then be solved as an optimal regulator problem. This implementation effectively drove the cruise missile to track complicated terrain accurately. However, it presented a distinct tradeoff between system performance and controller complexity.

In [9], a GPC algorithm was proposed with an innovation to the standard prediction model implementation. In standard GPC algorithms, these parameters must be calculated from the discrete model, linearized at every time-step, or a nonlinear optimization must be computed. Alternatively, a neural network model was proposed as a predictor to calculate the GPC parameters.

A different application of neural networks to predictive control of trajectory tracking was proposed in [10]. The method presented, dubbed model predictive neural control (MPNC), is a combination of a neural network feedback controller and a state-dependent Riccati equation (SDRE) controller. For this method, the initial stabilizing control is provided by the SDRE controller, incorporating robustness improvement over traditional linearization based Linear Quadratic (LQ) controllers. The SDRE control is then augmented by the neural network (NN) controller, optimized within a receding horizon model predictive control framework. Initially the SDRE controller dominates, providing stable tracking and good conditions for the NN training. Then gradually, as training of the NN progresses, it takes over more of the control authority, yielding superior performance over traditional SDRE and LQ control.

Non-linear model predictive control (NMPC) was applied by Kim et al. to combine the trajectory generation and tracking problems [11]. A quadratic cost function is defined with an output trajectory tracking error term and a control term. Additionally, a term can also be introduced to bound the state variables that do not directly appear in the output. The resulting cost is minimized subject to input constraints, guaranteeing physically realizable trajectories, and solving both the trajectory generation and tracking problems in a single step.

In a 2-D urban terrain guidance and control problem, Singh and Fuller utilized NMPC with hard output state constraints defined in the earth-frame to accomplish obstacle avoidance [12]. The use of control basis function inputs incorporated into perturbational linearization of the plant is also proposed, preserving the plant's non-linear characteristics while providing a model simplification which significantly cuts down on computational complexity.

1.2 PROBLEM STATEMENT

In traditional autonomous NOE flight, with or without obstacle avoidance, velocity is typically sacrificed to attain fine altitude tracking. This research will propose a control method which will strive to attain the altitude tracking performance of previous research while maintaining the high speeds necessary for survival in case of vehicle detection.

It is evident from the literature review that Model Predictive Control (MPC) is a prime candidate algorithm for this terrain following guidance and control problem. MPC is one of the few methods available for multi-variable systems that can incorporate hard constraints on controls and states [11]. Its finite horizon especially lends MPC to changing environments and objective functions. This research seeks to take advantage of this capability for obstacle avoidance by imposing hard constraints on desired output states. This idea is extended from the lateral obstacle avoidance presented in [12] to include ground collision avoidance for NOE flight. This research also seeks to combine the trajectory generation and tracking problems using MPC in a fashion that lends itself to real-time implementation and dynamic replanning based on sensor updates.

The proposed algorithm begins with a coarsely sampled terrain database for initial NOE reference trajectory generation. A cost function is then defined which introduces a Terrain Following/Terrain Avoidance (TF/TA) parameter similar to that suggested in [8] allowing the user to adjust to what degree the terrain is tracked at the expense of the lateral tracking performance. This cost function is minimized by Model Predictive Control (MPC), a finite horizon, optimal control scheme, subject to dynamic vehicle constraints, control input constraints, and output constraints which fold in detailed terrain and obstacle information provided real-time by a Light Detection and Ranging (LIDAR) sensor. Thus, the new refined optimal trajectory inherently includes both terrain following and obstacle avoidance of terrain features such as trees to go around or wires to climb over at whatever nominal speed is input. In order to reduce the computational complexity of the optimization, control basis function inputs incorporated into perturbational linearization of the plant are utilized. This research will extend beyond the basic algorithm design and tuning to simulated vehicle application with an LQR gain-scheduled inner loop stability augmentation system/attitude controller. A generic algorithm is proposed to deal with unexpected terrain features as they apply to the NOE

path (obstacles requiring a go-over response) along with two methods for incorporating the lateral obstacle information (obstacles requiring a go-around response) into the optimization constraints. In addition, sensitivity analysis of terrain tracking to various terrain cross sections is explored with respect to a range of nominal velocities varying from 10 knots to 60 knots. Two samples of these results are plotted below in Figure 1-1 and Figure 1-2.

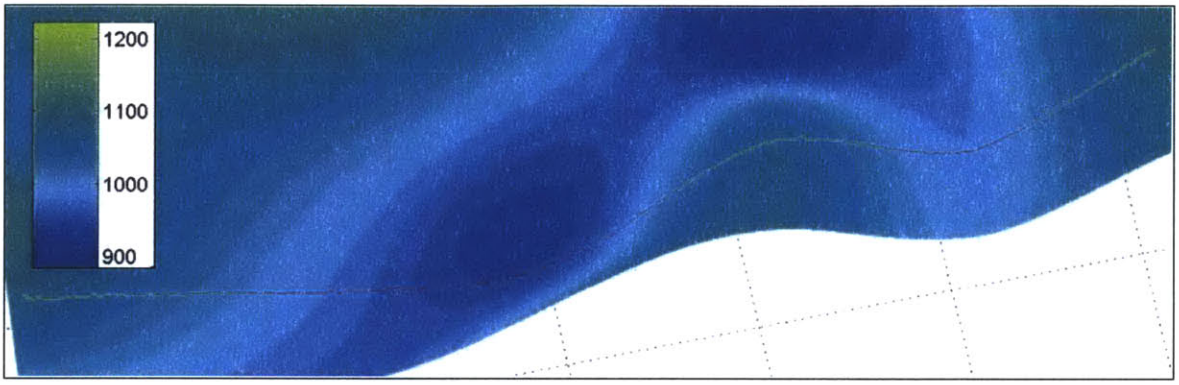

Figure 1-1: Sample results over a docile terrain section with maximum climb angle of 8 degrees plotted with its maximum attainable nominal velocity of 40 knots.

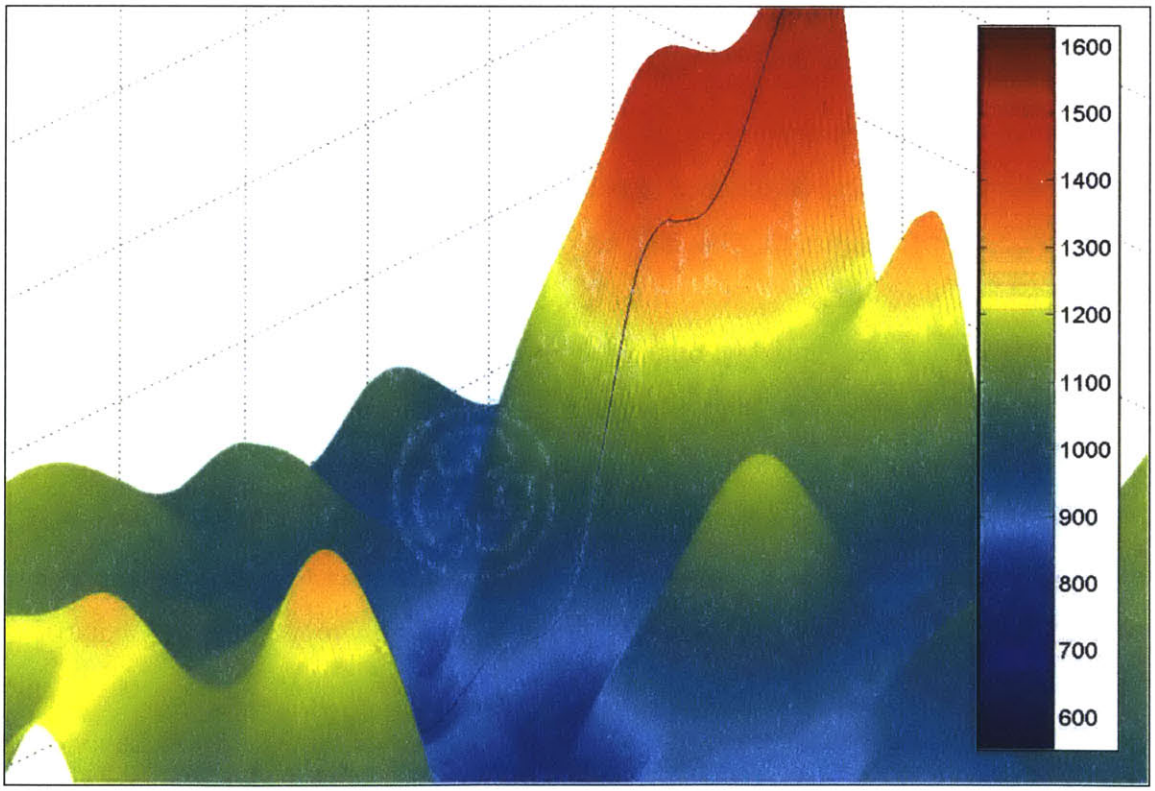

Figure 1-2: Sample results over a mountainous terrain section with maximum climb angle of 60 degrees plotted with its maximum attainable nominal velocity of 15 knots.

1.3 THESIS ORGANIZATION

Following the introduction, this thesis is organized as follows:

Chapter 2 Vehicle and Terrain Models: This chapter introduces and discusses the frames for and terrain over which the reference trajectory will be set. The non-linear plant model used in this research is also introduced in this chapter.

Chapter 3 Nap of the Earth Flight: This chapter describes the process used to generate and update the reference trajectory traversing the terrain from the specified initial waypoint to the final waypoint. It also explains the form of the cost function selected for this research and how the mission objectives are included into the optimization by way of the reference trajectory and cost function parameters.

Chapter 4 The Tracking Problem: This chapter introduces a build up of the model predictive control (MPC) algorithm from its infinite horizon roots in Linear Quadratic Tracking, the restriction to a finite horizon in the traditional MPC implementation, and finally introduction of an alternative implementation which drastically decreases algorithm complexity and run-time.

Chapter 5 MPC Parameter Tuning for Terrain Following and Terrain Following Results: This chapter treats the problem of following the terrain at 6 meters above ground level (AGL), enforcing ground collision avoidance through the application of a lower bound on altitude which is set at a safe distance above the ground. Parameter sensitivities for cost function penalties and prediction horizon length are investigated with respect to stability and tracking performance. With MPC parameters set, the algorithm sensitivities to terrain severity at varying nominal velocities are explored through application to several terrain cross sections.

Chapter 6 Constraint Based Obstacle Avoidance: This chapter describes the heuristics used to set the constraints for longitudinal (go-over) obstacle avoidance response and lateral (go-around) obstacle response, in addition to the terrain following altitude constraints described in Chapter 5. The heuristic descriptions are followed by the results obtained from each obstacle avoidance method.

Chapter 7 Conclusions and Recommendations: This chapter summarizes the conclusions reached in the course of this research and includes recommendations for future work related to these topics.

[This Page Intentionally Left Blank]

Chapter 2

VEHICLE AND TERRAIN MODELS

Before the Model Predictive Control (MPC) algorithm can be fully developed, there are two key pieces which must be introduced. The reference trajectory, folding the mission objectives into the cost function, defined discretely in time, and the prediction model describing the system dynamics. Both will be used together to enforce attainment of mission objectives through precise tracking of current and future references. This chapter will set up the frames for and discuss the terrain over which the reference trajectory (to be introduced in Chapter 3) will be set. The MPC internal model of the plant will be introduced as well as the high fidelity helicopter model to which the proposed guidance and control algorithm will be applied.

2.1 FRAME OF REFERENCE

There are two frames of reference that must be introduced before the vehicle and terrain models can be presented. Firstly, in order to describe trajectories (reference and actual) in an intuitive manner relative to the terrain, an inertial coordinate frame must be introduced. Figure 2-1 illustrates the earth frame (N E D reference frame) as defined for this research.

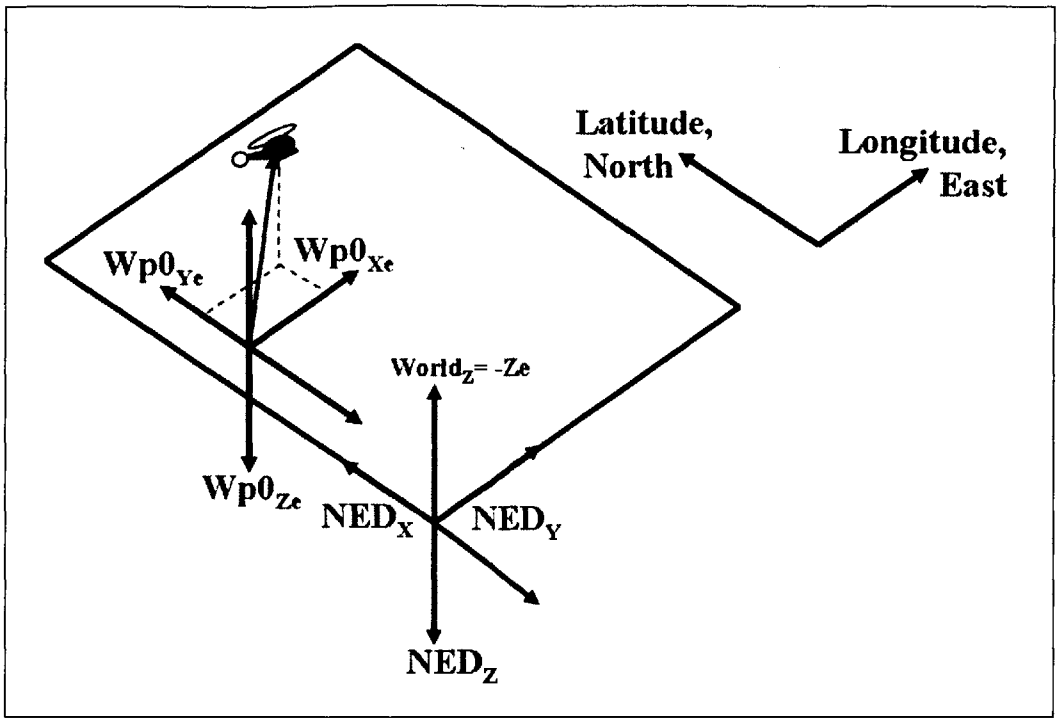

Figure 2-1: Earth Frame definition showing traditional N E D coordinates contrasted with the earth frame Xe, Ye and Ze defined for this research. North and East are also indicated for reference.

The earth frame is an inertial reference frame, with axes X, Y and Z, in which the origin is arbitrarily placed at the vehicle's initial position on the earth's surface at sea-level (see Section 2.3 for assumptions about earth curvature). The X-axis is aligned with lines of longitude, the Y-axis is aligned with lines of latitude and the Z-axis is orthogonal to those two, with positive Z indicating meters below sea-level. N E D refers to the defined positive directions in this frame, north, east and down, however it should be noted that for this research, the traditional NEDx axis was assigned to the earth frame Y-axis and the traditional NEDy axis was conversely assigned to the earth frame X-axis. This is due to the arbitrarily set earth frame coordinates on the terrain and appropriate angle modifications were made to account for the fact that the earth frame is not defined as a purely right handed frame.

In order to present the standard helicopter equations of motion, the vehicle body-fixed frame must be introduced. Figure 2-2, 2-3 and 2-4 illustrate this coordinate frame from the front, side and top of the vehicle, respectively [13, 14]. Body attitude is described in terms of Euler angles; ϕ, describing rotation around the I-axis (radians), θ, describing

rotation around the J axis, and ψ, describing rotation about the J-axis. These angles are utilized in the coordinate transformation between the body and earth frames.

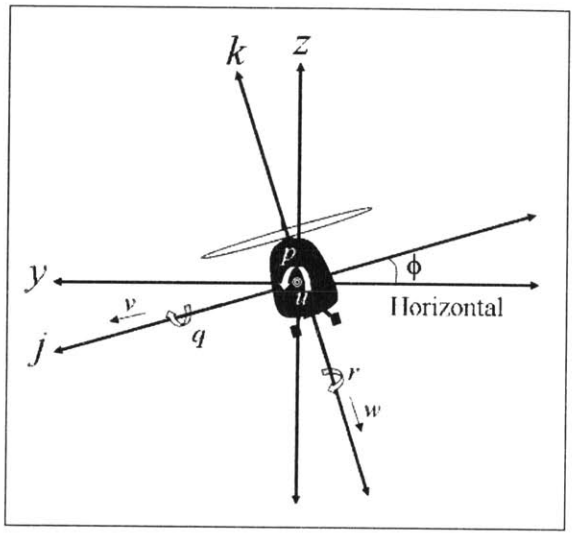

Figure 2-2: Body-fixed Frame; front view.

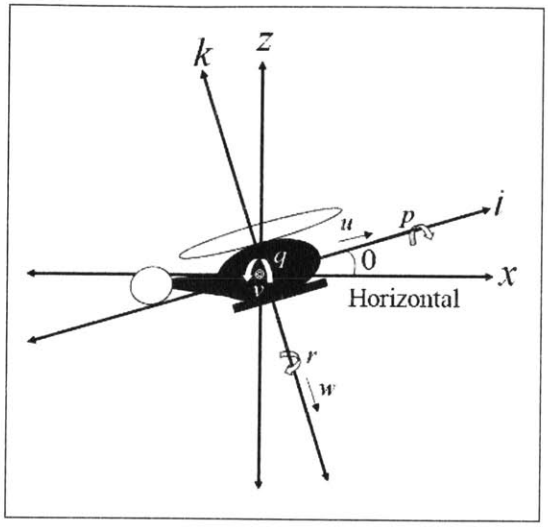

Figure 2-3: Body-fixed Frame; side view.

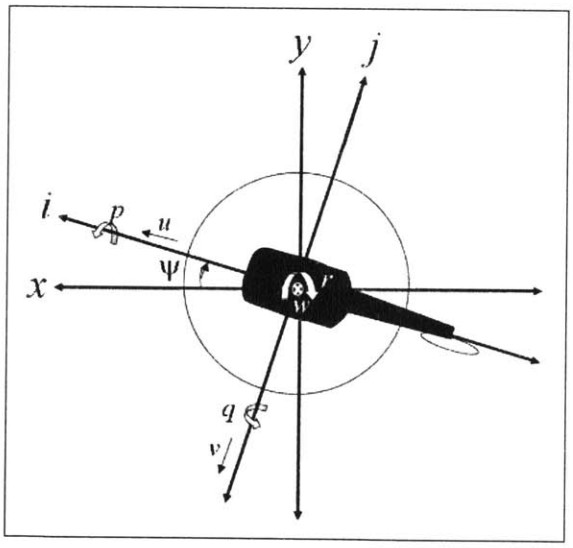

Figure 2-4: Body-fixed Frame; top view.

As can be clearly seen in the figures, the lateral plane axes, I and J, point through the nose of the vehicle and perpendicularly out the right side of the vehicle, respectively. The K-axis, orthogonal to I and J, points straight out the bottom of the vehicle. The vehicle states will be defined in terms of this frame, propagated with the calculated controls at each iteration and then transformed to the earth frame. The transformed state is utilized in the control optimization to minimize the error between the trajectory reference and actual output trajectory, both defined in the earth frame.

2.2 VEHICLE MODEL

To accomplish true low-altitude NOE flight, an agile maneuvering vehicle is required. Currently, NOE flight is a common mode of operation for manned helicopters which can provide such maneuvering. This research seeks to automate this capability, therefore a helicopter was selected as the test vehicle for this algorithm. There are two levels of the vehicle model required for this research, an internal predictive model for the controller and a high fidelity plant model for the control trajectory to be applied to in simulation.

2.2.1 Prediction Model

As mentioned in the chapter introduction, the prediction model is an internal representation of the vehicle dynamics with which MPC can predict and minimize future tracking errors. The base for the prediction model used in this research consists simply of generic equations of motion for a helicopter with the following state vector:

(1) $$x = \begin{bmatrix} V \\ \omega \\ \phi \\ \theta \\ \psi \\ X_e \end{bmatrix},$$

where:

(2) $V = \begin{bmatrix} u \\ v \\ w \end{bmatrix}$, the vehicle velocities defined in the body frame,

(3) $\omega = \begin{bmatrix} p \\ q \\ r \end{bmatrix}$, the vehicle angular velocities defined in the body frame,

ϕ, θ and ψ are the Euler angles defined previously in Section 2.1, and

(4) $X_e = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$, the earth frame positions, as defined in Section 2.1, calculated by rotating and integrating the body frame velocities accordingly (for

more information on the implementation of the prediction model see Section 4.3). This generic helicopter has four control inputs: the main rotor thrust, T_{MR}, roll cyclic command setting the roll angle of the main rotor plane, θ_R, pitch cyclic command setting the pitch angle of the main rotor plane, θ_P, and the tail rotor thrust, T_{TR}. The sign conventions for these controls are defined in Figure 2-5 and Figure 2-6. The two force inputs, T_{MR} and T_{TR} are translated using the cyclic rotor plane angle commands into the components with respect to the body frame axes so they can be input into the equations of motion. The resulting forces are:

$$F_{Main} = \begin{bmatrix} T_{MR}\cos(\theta_R)\sin(\theta_P) \\ -T_{MR}\sin(\theta_R) \\ T_{MR}\cos(\theta_R)\cos(\theta_P) \end{bmatrix} \quad (5)$$

and

$$F_{Tail} = \begin{bmatrix} 0 \\ T_{TR} \\ 0 \end{bmatrix}. \quad (6)$$

The control constraints that were placed on the vehicle (and converted to constraints on the scaling factors, α, as described in Section 4.3.2.1) are the following:

$0G$	≤	Main Rotor Thrust	≤	$3½G$
-20°	≤	Pitch Angle of Main Rotor Plane	≤	20°
-20°	≤	Roll Angle of Main Rotor Plane	≤	20°
$-½G$	≤	Tail Rotor Thrust	≤	$½G$

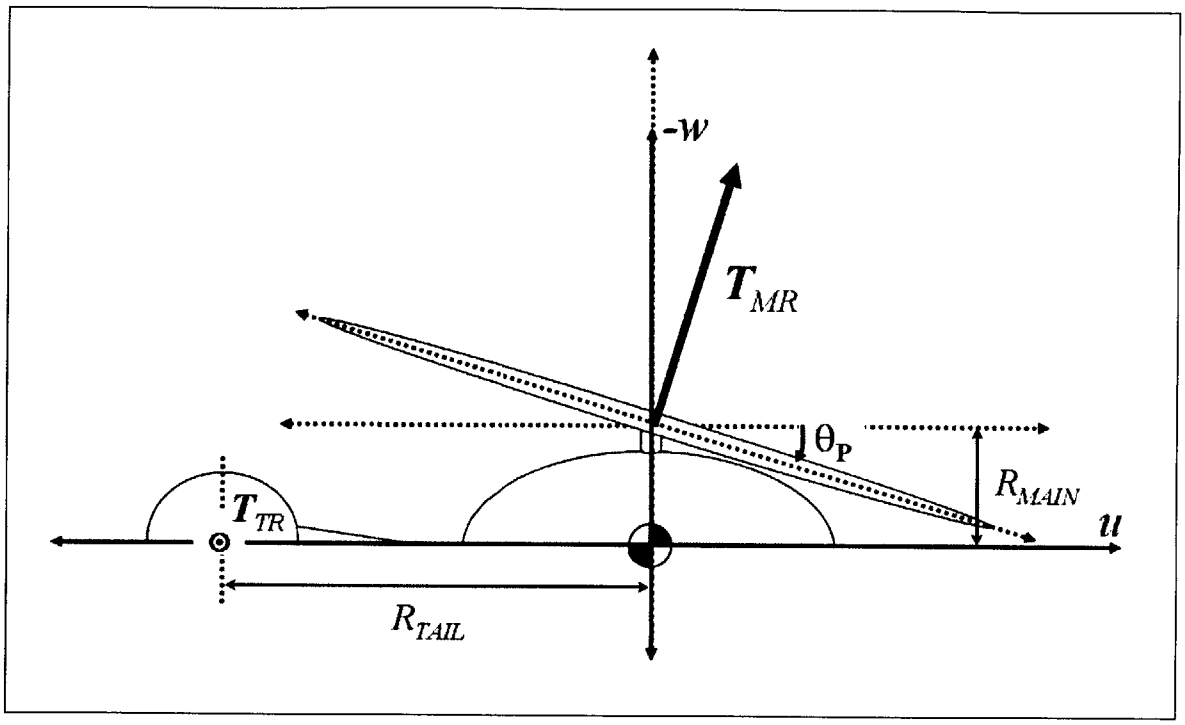

Figure 2-5: Side view of helicopter defining the positive direction for θ_P and defining model constants R_{MAIN} and R_{TAIL} with respect to the control inputs T_{MR} and T_{TR}.

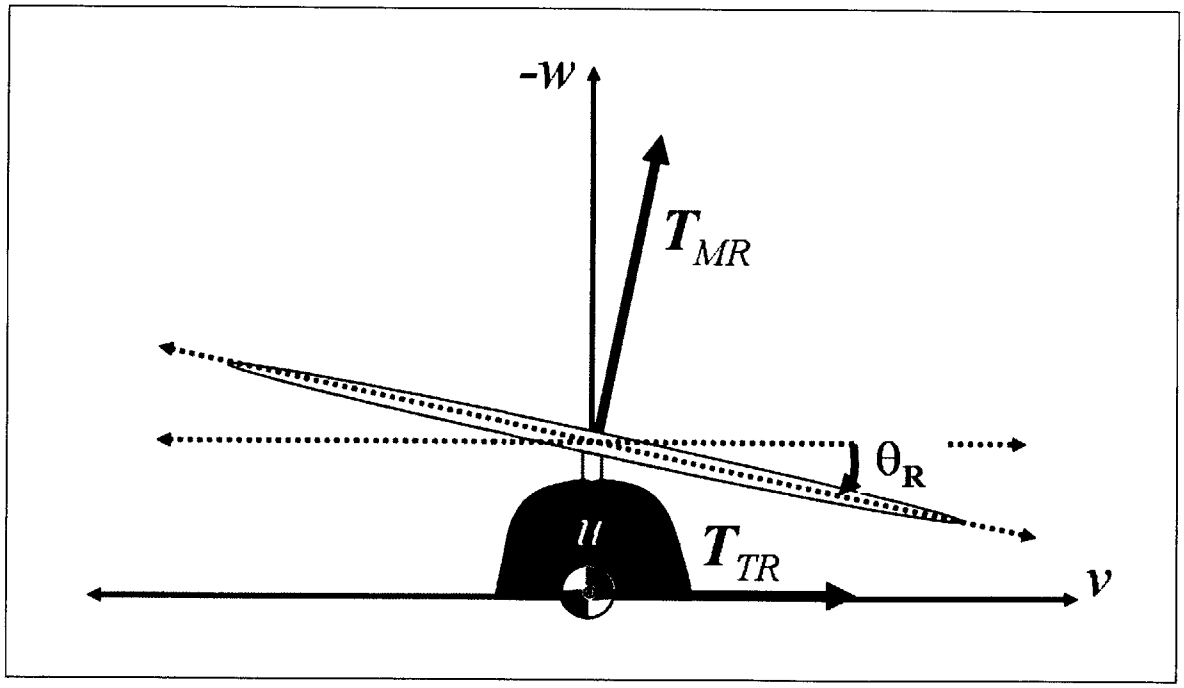

Figure 2-6: Front view of helicopter defining the positive direction for θ_R and T_{TR}.

The equations of motion used in the prediction model are as follows:

(7) $$\dot{V} = -\omega \times V + R_B^i(G) - k_D \|V\|_2 V + (F_{Main} + F_{Tail})$$

(8) $$\dot{\omega} = J^{-1}(-\omega \times J\omega) + T$$

(9) $$\dot{\phi} = p + q\sin\phi\tan\theta + r\cos\phi\tan\theta$$

(10) $$\dot{\theta} = q\cos\phi - r\sin\phi$$

(11) $$\dot{\psi} = \frac{q\sin\phi + r\cos\phi}{\cos\theta}$$

(12) $$\dot{X}_e = R_i^B(V)$$

where:

k_D = generic drag term approximating $\frac{1}{2}C_D\rho S$, (C_D = drag coefficient, ρ = air density, S = vehicle reference area), multiplied by velocity squared in (7) yielding the force of drag on the vehicle

R_i^B = rotation matrix from body frame to inertial frame

R_B^i = rotation matrix from inertial frame to body frame

J = vehicle moment of inertia matrix

G = force of gravity specified in the inertial frame

T = sum of the torques generated in the body frame by F_{Main} and F_{Tail}

For the MPC loop simulations the calculated controls were applied to these equations of motion limiting the model variation between internal and truth model to errors from the perturbational linearization contained in the controller solution. Future work involving the application of these algorithms to a high fidelity truth model and investigation of MPC robustness to variation in the plant is described in Section 7.2.

2.3 TERRAIN MODEL

Almost all current and proposed low-altitude flight systems integrate onboard obstacle sensing with a terrain/obstacle database. A terrain/obstacle database that is often used for such applications is Digital Terrain Elevation Data (DTED) which is maintained by The National Imagery and Mapping Agency (recently restructured to form the National Geospatial-Intelligence Agency). Much of the low-altitude flight systems to date

use DTED Level 1 (100 m spacing) or Level 2 (30 m spacing), however new technologies such as the Shuttle Radar Topography Mission (SRTM) which flew in February 2002 [15] and the Geographic Synthetic Aperture Radar (GeoSAR) System, a new, dual-frequency airborne radar mapping system designed to map the earth beneath foliage and other vegetation, released in June, 2001 [16], have increased the availability of database with resolution to Level 5 (1 m spacing). Figure 2-7 shows the increase in feature accuracy that is obtained with increasing levels of DTED data as shown for the example urban region of Baltimore, Maryland.

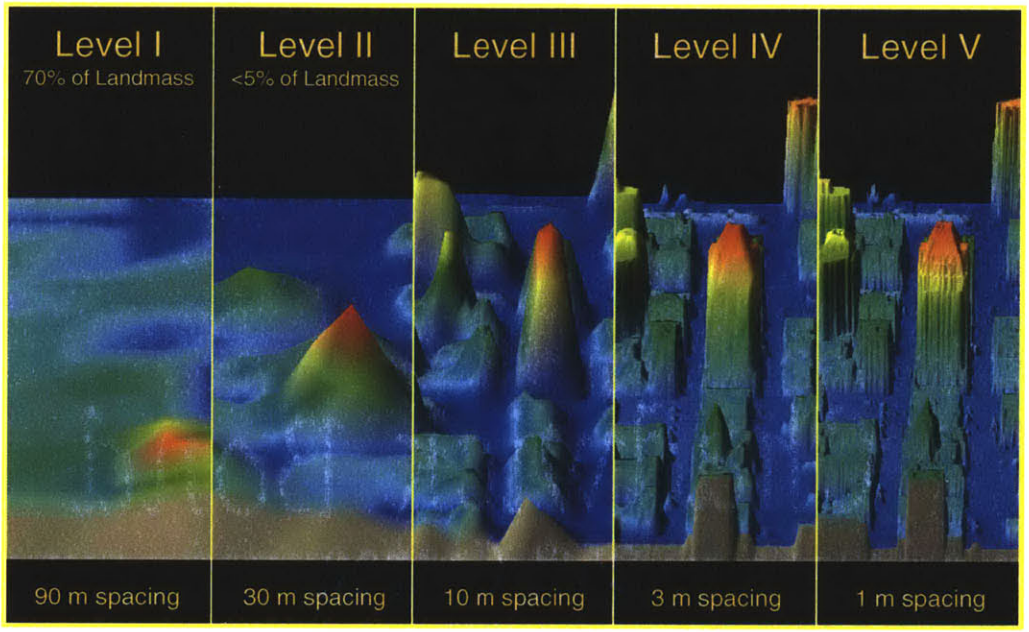

Figure 2-7: DTED Level 1 through 5 Representation of Baltimore, MD [17].

The terrain model used in this research begins with Digital Terrain Elevation Data (DTED) Level 0 of area around Mt. Adams in Washington State. Not shown in Figure 2-7, DTED Level 0 has 1 km spacing and is the only DTED data available without security clearance. DTED Level 0 data was derived from Level 1 and various other sources initially to provide the FAA with some planning data. This data is now available for public release through the NGA geospatial engine [18]. Detailed information about the elevation data formal for all levels of DTED can be found in the Military Performance Specification MIL-D-89020A, [19]. Since higher resolution data is attainable for actual implementation to low-altitude flight systems of the algorithm being designed in this research, the 1 km resolution data was interpolated to 25 m resolution

(approximately DTED Level 2) and that resulting terrain database was used in simulation and algorithm development.

MPC requires a discrete reference trajectory with which to input the control objectives. In our case, this trajectory must consist of the desired x, y and z location of the vehicle for each point in time along the trajectory. To be able to construct such a reference, a terrain map is needed from which the appropriate elevation can be extracted for each (x,y) coordinate. Therefore, the DTED data given in terms of latitude and longitude was reformatted to the earth frame using the Matlab script *dted.m* which extracts the DTED data and formats it into a matrix of elevations referenced by latitudes and longitudes. A custom script was then applied called *mtadams4mpc.m* which sets the origin of the earth frame (as specified in Section 2.1) and sets up the x and y axes to the correct distances in meters from the origin corresponding to the latitudes and longitudes previously assigned. For all simulation, the curvature of the earth is assumed to have a negligible affect upon our terrain model. The size of each pre-interpolation DTED array element was assumed to be 1 km by 1 km and the x and y axes were set based on that assumption.

[This Page Intentionally Left Blank]

Chapter 3

NAP OF THE EARTH (NOE) FLIGHT

Nap of the Earth (NOE) flight or terrain following flight is defined by the Department of Defense as "flight close to the Earth's surface during which airspeed, height, and/or altitude are adapted to the contours and cover of the ground in order to avoid enemy detection and fire," [3]. This indicates altitudes of less than 10 meters above the ground. In order to fully take advantage of ground cover, for this research the reference altitude of 6 meters above ground level (AGL) was selected. The following chapter describes the process used to generate and update the reference trajectory traversing the terrain from the specified initial waypoint to the final waypoint. It also explains the form of the cost function selected for this research and how the mission objectives are included into the cost function by way of the reference trajectory and cost function parameters.

3.1 STRAIGHT LINE REFERENCE TRAJECTORY GENERATION

To establish the reference trajectory from which the actual trajectory will be subtracted in the cost function, we follow what is typical to most terrain following systems and start with an initial waypoint and a final waypoint, using a straight line path between the two as our first guess at a nominal 2D trajectory. The z-dimension is extracted from elevation data corresponding to each (x,y) reference coordinate and the terrain following reference path is set at the desired distance, 6 meters, above the ground level along our 2D path. The complete reference trajectory is synthesized by sampling the

resulting 3D reference path at the control rate (20 Hz) assuming the nominal velocity selected for the simulation (varying from 10-30 knots). This reference trajectory generation function is rerun at every MPC recalculation therefore, rather than having a predetermined invariant reference path from which the new trajectory is resampled, a new reference trajectory is redrawn using the new position after the N_U controls have been applied as the new initial waypoint. This is particularly useful in the replanning sense when applied to obstacle avoidance scenarios. Figure 3-1 below demonstrates this over three MPC loop iterations (t = 0, t = H_U and t = $2H_U$), showing the initial reference trajectory and the actual trajectory at each time.

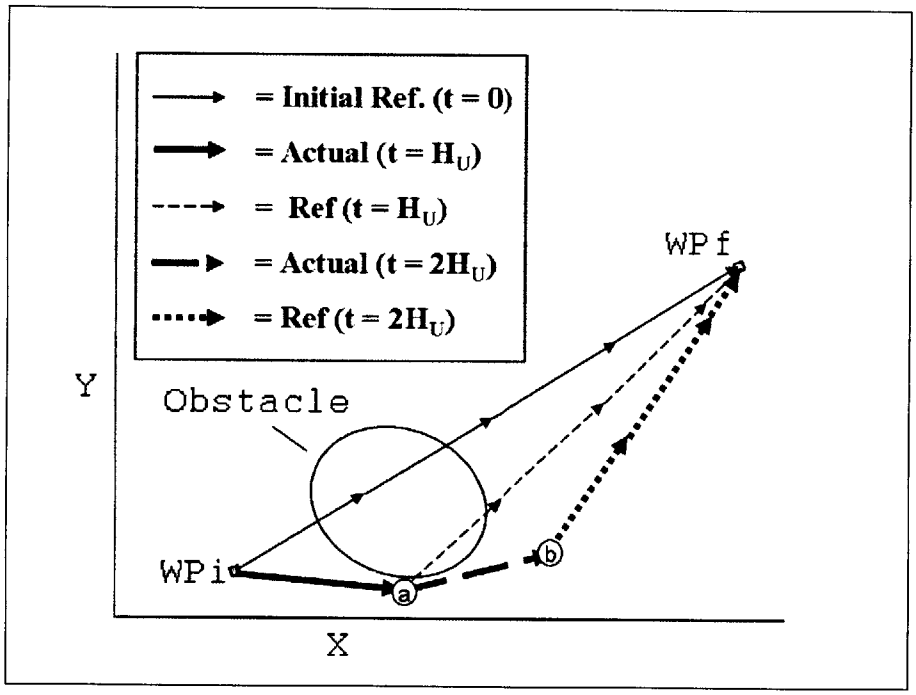

Figure 3-1: Diagram illustrating the reference trajectory update functionality with an obstacle for three iterations of the MPC loop.

As you can see, at t = 0, the straight line reference takes the vehicle through an obstacle, therefore with appropriately defined constraints, the resultant position at t = H_U when MPC is recalculated (after N_U controls have been applied), the new position is actually at point 'a'. From here, the straight line reference is redrawn to the final waypoint, but again, the reference leads through the obstacle. After appropriately defined constraints are applied and the next N_U controls have been calculated and applied, the new resultant position at t = $2H_U$ is at point 'b'. Finally, from this new initial position, an uninhibited straight line reference can be drawn to the endpoint and tracked. However,

even if there are no obstacles causing the actual trajectory to deviate from the reference trajectory due to the placement of constraints, performance degradation due to disturbance such as position drift due to wind will also be accounted for with each reference regeneration. The benefit of this in application is shown in the figures below which compare an actual obstacle avoidance scenario with and without the reference update feature.

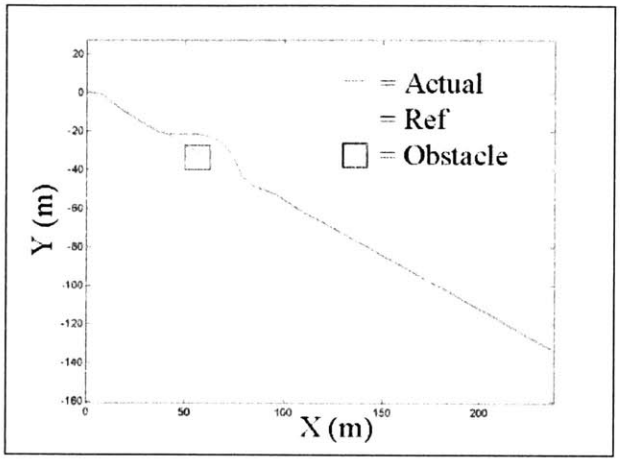

Figure 3-2: Obstacle avoidance with no reference update.

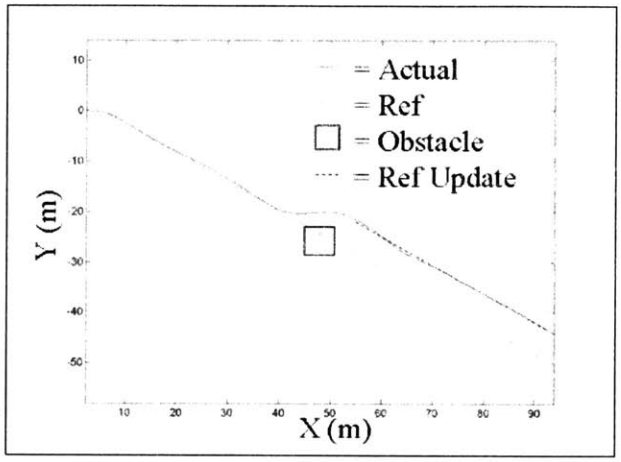

Figure 3-3: Obstacle avoidance with reference update.

Figure 3-2, shows the vehicle response without path replanning. As you can see, the vehicle is drawn back to the initial reference trajectory possibly wasting time and energy. In contrast, with the path replanning incorporated into the reference trajectory generation, Figure 3-3 shows the vehicle is able to pass the obstacle then continue straight on to the final waypoint. (The final waypoint in Figures 3-2 and 3-3 is placed several kilometers in the future, hence the redrawn post-obstacle reference trajectory looks parallel to the original reference trajectory.)

3.2 SPLINE SEARCH REFERENCE TRAJECTORY GENERATION

It is well known that in a high threat environment where nap-of-the-earth flight is desirable, a straight line path from the initial waypoint to the final waypoint is not always the best solution, even at 6 meters above ground. For example, if the initial waypoint and the final waypoint lie on either side of a mountain, it would be far less desirable to fly straight over the mountain than to use the mountain as a shield and fly around the bottom of the mountain. With this in mind, a spline reference trajectory search algorithm was

designed and implemented in this research as an addition to the MPC outer loop guidance and control. The spline search reference trajectory generator starts with the straight line reference and tests all of the points along the horizon for the max deviation in altitude. It then extends splines of varying radii to the right and to the left of this original reference, testing for max deviation as well. The path which results in the least overall altitude deviation (spline or straight line) is chosen as the best reference trajectory and then sampled appropriately for the desired horizon length according to the procedure defined in the previous section. Figure 3-4 below shows one iteration of this trajectory test. A straight line reference is plotted on the sample terrain from initial to final waypoints along with potential spline references.

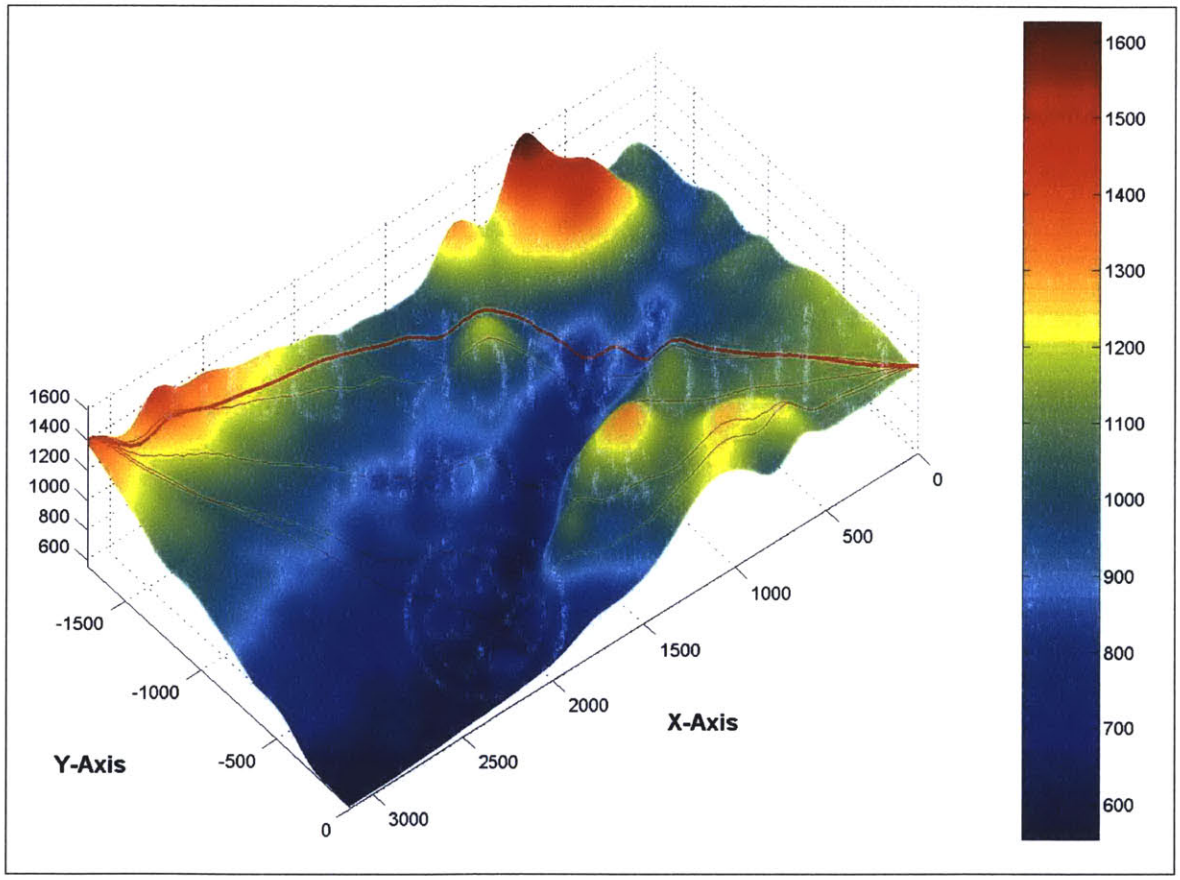

Figure 3-4: Sample terrain with straight line and spline potential trajectory options plotted and selected trajectory in bold.

The selected reference trajectory appears in bold. As you can see, using the criteria of least overall altitude deviation, the spline closest to the X-axis is selected as the reference trajectory because it involves flying around the deep ravine and skirting along the edge of the mountain in the closest corner of the plot, rather than flying down and through the

ravine. This spline reference test can be applied at each MPC loop recalculation, thus updating the trajectory in a manner similarly to that described in the reference re-draw feature in Section 3.1.

Perceptibly, the spline reference trajectory search algorithm could be easily modified to incorporate a number of different optimal criteria in addition to or instead of maximum altitude deviation but such investigation and discussion exceeds the scope of this thesis.

3.3 ATTITUDE ASSERTION THROUGH Ψ-REFERENCE

As currently defined, the trajectory to be tracked over time t_i, where i is each time-step along the prediction horizon length, includes solely position references:

$$\mathbf{x}_{REF_k} = \begin{bmatrix} x_{REF_k} \\ y_{REF_k} \\ z_{REF_k} \end{bmatrix}.$$

In addition, to assert vehicle attitude as it follows the reference trajectory, a Ψ-trajectory must also be included in the reference. The state Ψ is defined as the yaw angle (rotation about the z-axis) between the vehicle's body axis forward velocity component, u, and the earth frame axis x. For this research, the Ψ-reference points the vehicle in the direction of the desired path. To accomplish this, given the established (x,y,z) references over a given horizon length, the following recursion is followed to set the Ψ-reference:

$$\Psi_{REF_{i+1}} = atan2\left\{\left(y_{REF_{i+1}} - y_{REF_i}\right),\left(x_{REF_{i+1}} - x_{REF_i}\right)\right\} \tag{13}$$

where *atan2* is a 4-quadrant arctangent solver available in most math function libraries.

In general, with this stipulation in the cost function, one can specify a certain side-slip to point the vehicle at a target as it flies past. Thus the Ψ-reference yields an opportunity to take a longer look at the target as you are passing it and potentially (depending on actual vehicle dynamic limits) even as you are retreating to a protected region. The same recursion listed above can be used to generate the reference only the target location should be inserted as follows:

$$\Psi_{REF_{i+1}} = atan2\left\{\left(y_{TARGET} - y_{REF_i}\right),\left(x_{TARGET} - x_{REF_i}\right)\right\} \tag{14}$$

3.4 COST FUNCTION DEFINITION TO SUPPORT NOE FLIGHT

The performance objectives of low altitude and high speed are imposed on the optimization problem through the definition of the cost function. These objectives are inserted into the problem through the low altitude, high speed reference trajectory. The cost function must be constructed to promote the conformity of vehicle performance to this trajectory with minimum error. To accomplish this, a quadratic form cost is set up:

$$(15)\quad J_k = \sum_{i=k}^{k+N_p} \begin{bmatrix} x_i - x_{REF_i} \\ y_i - y_{REF_i} \\ z_i - z_{REF_i} \\ \Psi_i - \psi_{REF_i} \end{bmatrix}^T \begin{bmatrix} Q_{X_i} & 0 & 0 & 0 \\ 0 & Q_{Y_i} & 0 & 0 \\ 0 & 0 & Q_{Z_i} & 0 \\ 0 & 0 & 0 & Q_{\psi_i} \end{bmatrix} \begin{bmatrix} x_i - x_{REF_i} \\ y_i - y_{REF_i} \\ z_i - z_{REF_i} \\ \Psi_i - \psi_{REF_i} \end{bmatrix} + \begin{bmatrix} T_{MR_i} \\ \theta_{R_i} \\ \theta_{P_i} \\ T_{TR_i} \end{bmatrix}^T \begin{bmatrix} R_{T_{MRi}} & 0 & 0 & 0 \\ 0 & R_{\theta_{Ri}} & 0 & 0 \\ 0 & 0 & R_{\theta_{Pi}} & 0 \\ 0 & 0 & 0 & R_{T_{TRi}} \end{bmatrix} \begin{bmatrix} T_{MR_i} \\ \theta_{R_i} \\ \theta_{P_i} \\ T_{TR_i} \end{bmatrix}$$

$$(16)\quad = (\boldsymbol{x}_i - \boldsymbol{x}_{REFi})^T Q_i (\boldsymbol{x}_i - \boldsymbol{x}_{REFi}) + \boldsymbol{u}_i{}^T R_i \boldsymbol{u}_i$$

$$(17)\quad = \boldsymbol{e}^T Q_i \boldsymbol{e} + \boldsymbol{u}_i{}^T R_i \boldsymbol{u}_i$$

This will penalize the tracking error ($\boldsymbol{e}$), defined to be the output state minus the reference trajectory state ($\boldsymbol{x}-\boldsymbol{x}_{REF}$) at each time t_i along the prediction horizon length, while simultaneously penalizing high control input ($\boldsymbol{u}$).

In addition to mission objective imposition on the problem through the reference trajectory inclusion in the cost function, the objectives are also enforced through the selection of the penalty matrices, Q_i and R_i. In the state weighting matrix, Q_i, introduce a Terrain Following/Terrain Avoidance (TF/TA) parameter similar to that suggested in [20]. This allows the user to adjust to what degree the reference trajectory is tracked longitudinally at the expense of the lateral tracking performance. The TF/TA ratio, τ, is implemented in Q_i through the assignment of Q_{Xi} and Q_{Yi} to 1 and the assignment of Q_{Zi} to τ. Setting a high value for τ here will allow little deviation from the set altitude about ground while tolerating a meandering xy-track. A smaller value will emphasize lateral tracking while not attending as highly to maintaining the precise distance above ground, (a safety margin will be enforced by constraints in either case).

The Ψ and cyclic control input penalties, $Q_{\Psi I}$, $Q_{\theta Ri}$ and $Q_{\theta Pi}$ are not as easily comparable to the other penalties Q_{Xi}, Q_{Yi} and Q_{Zi}, and R_{TMRi} and R_{TTRi}, because of the difference in units between them. A combination of two commonly used rules-of-thumb for cost function weight selection in optimal control will be employed to normalize the cost function penalties for initial testing. The order of magnitude rule normalizes the

magnitude of cost contribution between the two units in question (meters and radians) in the setting of the cost penalties. The percentage of range rule normalizes the penalties so that the cost contribution for a deviation is associated with the percentage of range spanned by that deviation, rather than by the magnitude of the number itself.

With respect to the terrain following mission objectives, tracking a heading reference is no more important than the lateral tracking therefore to account for or normalize the magnitude difference in the cost contribution from radians relative to the cost contribution from meters, $Q_{\Psi i}$ was set to one order of magnitude higher than Q_{Xi} and Q_{Yi}. Naturally, if the Ψ-reference is being used to impose an attitude trajectory in support of reconnaissance operations, $Q_{\Psi i}$ should be set at a higher value to reflect its importance to the overall mission objective, keeping in mind the order of magnitude normalization to x, y and z penalties.

The same order of magnitude normalization idea holds for the control penalty matrix, R_i. Of the four control inputs, two can be reasonably compared to meters and two must be compared to radians. T_{MR} and T_{TR} are both measured in m/s^2 and their associated penalties, R_{TMRi} and R_{TTRi}, can therefore be reasonably compared to the position tracking states. Control input being less important to penalize than the tracking error in the enforcement of our mission objectives, if $Q_{Zi} = \tau = 10$, and $Q_{Xi} = Q_{Yi} = 1$, R_{TMRi} and R_{TTRi} should be set to 0.1. Normalizing between the order of magnitude difference between meters and radians, $R_{\theta Ri}$ and $R_{\theta Pi}$ should be set an order of magnitude higher, 1, to more accurately penalize their control effort with respect to the rest of the problem. However, unlike the states which are essentially unconstrained other than for obstacle and terrain avoidance, the control inputs are constrained, therefore their penalties should also reflect the relative magnitude of an input with respect to their range. For main rotor thrust, the constraint is set as a maximum load factor of 3.5 Gs. This results in an acceleration range of 34.3 m/s^2 as compared to the cyclic commands which have a range ~1.04 radians (from -0.5 to +0.5 radians) $\cong$ 60 degrees (from -30 to +30 degrees). Therefore a 0.1 unit of cyclic command is actually 9.4% of the total range as opposed to one full unit of thrust which is 2.9% of the total range. To try to balance this, for all tests leading up to Section 5.2.2 where the weighting matrices will be determined through performance

analysis, the cyclic gains will be reduced by an order of magnitude, yielding the final initial control weighting matrix values: $R_{\theta Ri} = R_{\theta Pi} = R_{TMRi} = R_{TTRi} = 0.1$.

Many resources are available specifically targeting the selection of weights as tuning parameters in optimization problems. See [21 - 23] for alternative suggestions or methods of cost function penalty selection.

Chapter 4

THE TRACKING PROBLEM

The terrain following control problem boils down to a straightforward reference tracking problem once the constraints of unexpected threats or obstacles are removed. Since terrain following flight is defined by its very low proximity to the ground, excellent reference tracking is crucial to the vehicle's survival. This leads us to optimal control as a prime candidate method for this research. Optimal control is defined as the control which minimizes a cost functional, within which the controller design requirements are defined by appropriately weighting the required states and controls. This framework will enable the precision altitude tracking necessary at such close proximities to the ground with the freedom to penalize poor lateral tracking and control effort.

4.1 LINEAR QUADRATIC REGULATOR (LQR) TRACKING

This section introduces linear quadratic tracking, the readily computed solution to the generic optimal tracking problem, as a potential solution to the terrain following problem. An example of its implementation to a generic terrain following problem such as that which this research is attempting to solve is also included and analyzed for real-time application.

4.1.1 Algorithm Description

As described in Section 3.4, mission objectives are folded into the optimization largely through the construction of a cost functional to penalize the error between the actual and reference trajectories and to penalize high control effort. Described in the continuous time domain for this LQR tracking formulation [22, 24], (15) becomes:

$$(18) \qquad J(x(t),t) = \int_0^{T_F} \left[\{x(t) - x_{REF}(t)\}^T Q \{x(t) - x_{REF}(t)\} + u(t)^T Ru(t) \right] dt$$

subject to the linearized state equation:

$$(19) \qquad \dot{x}(t) = Ax(t) + Bu(t)$$

Two common methods available for derivation of a closed form solution to the linear quadratic tracking problem are dynamic programming and calculus of variations. For complete derivations of both methods see [22] and [24]. This thesis will approach the solution from a dynamic programming standpoint. The optimal cost-to-go function is defined, without a penalty on the terminal state, as:

$$(20) \qquad J^*(x(t),t) = \min_{u(\bullet)} \int_t^{T_F} \left[\{x(t) - x_{REF}(t)\}^T Q \{x(t) - x_{REF}(t)\} + u(t)^T Ru(t) \right] dt$$

Applying Bellman's principle of optimality [22], the optimal cost can be written as the optimal cost to the next time step plus the minimum cost-to-go from the next step to the end, yielding the continuous time recurrence equation:

$$(21) \qquad J^*(x(t),t) = \min_{u(t)} \left[\int_t^{t+dt} J(x(t),t)dt + J^*(x(t+dt),t+dt) \right]$$

which can be approximated as the following:

$$(22) \qquad \approx \min_{u(t)} \left[\{x(t) - x_{REF}(t)\}^T Q \{x(t) - x_{REF}(t)\} + u^T(t) Ru(t) \right] dt + J^*(x(t+dt),t+dt).$$

The approximation sign indicates that the result is correct to the first order in dt. As $dt \to 0$, the result is an equality. Applying a first order approximation to the optimal cost to go, $J^*(x(t+dt),t+dt)$ in the second half of the equation yields:

$$(23) \approx \min_{u(t)} \left[\{x(t) - x_{REF}(t)\}^T Q \{x(t) - x_{REF}(t)\} + u^T(t) Ru(t) \right] dt + J^*(x(t),t) + \left[\dot{J}^*(x(t),t) \right] dt$$

One way this equation can be solved is to guess a solution form and see if it can be made to satisfy the differential equation and the boundary conditions [22]. For the convex cost function of this research, the optimal cost form is asserted to be a quadratic in the state:

$$J^*(x(t),t) = x^T(t)P(t)x(t) + 2q^T(t)x(t) + r(t) \tag{24}$$

where $P(t)$ is an n × n matrix, $q(t)$ is an n × 1 vector and $r(t)$ is a scalar. This is an appropriate guess because a convex cost function minimized subject to convex constraints yields a convex solution space [25], such as the space described by the quadratic in (24). Taking the derivative of (24), $J^*(x(t),t)$ and $\dot{J}^*(x(t),t)$ can be inserted into (23), yielding:

$$\begin{aligned} &\approx \min_{u(t)}\Big[\{x(t)-x_{REF}(t)\}^T Q\{x(t)-x_{REF}(t)\}+u^T(t)Ru(t)\Big]dt + x(t)^T P(t)x(t)+2q(t)^T x(t)+ \\ &\quad r(t)+\Big[\dot{x}^T(t)P(t)x(t)+x^T(t)P(t)\dot{x}(t)+x^T(t)\dot{P}(t)x(t)+2\dot{q}(t)x(t)+q^T(t)\dot{x}(t)+\dot{r}(t)\Big]dt \end{aligned} \tag{25}$$

Substituting in the linearized state equation constraint for $\dot{x}(t)$ yields:

$$\begin{aligned} &= \min_{u(t)}\Big[\{x(t)-x_{REF}(t)\}^T Q\{x(t)-x_{REF}(t)\}+u^T(t)Ru(t)\Big]dt + x(t)^T P(t)x(t)+2q(t)^T x(t)+ \\ &\quad r(t)+\Big[\{Ax(t)+Bu(t)\}^T P(t)x(t)+x^T(t)P(t)\{Ax(t)+Bu(t)\}+x^T(t)\dot{P}(t)x(t)+ \\ &\quad 2\dot{q}(t)x(t)+q^T(t)\{Ax(t)+Bu(t)\}+\dot{r}(t)\Big]dt \end{aligned} \tag{26}$$

Finally, combining (22), (24), and (26), and subtracting $x^T(t)P(t)x(t)+2q^T(t)x(t)+r(t)$ from both sides yields:

$$0 = \min_{u(t)}\begin{bmatrix} \{x(t)-x_{REF}(t)\}^T Q\{x(t)-x_{REF}(t)\}+u^T(t)Ru(t)+\{Ax(t)+Bu(t)\}^T P(t)x(t)+ \\ x^T(t)P(t)\{Ax(t)+Bu(t)\}+x^T(t)\dot{P}(t)x(t)+2\dot{q}(t)x(t)+2q^T(t)\{Ax(t)+Bu(t)\}+\dot{r}(t) \end{bmatrix} \tag{27}$$

To find the optimal control, $u(t)$, the derivative of the quantity within the brackets is taken with respect to u and set equal to zero:

$$\frac{d}{du}[\bullet] = 0 \tag{28}$$

$$= 2Ru(t) + 2B^T P(t)x(t) + 2B^T q(t) \tag{29}$$

Therefore, solving for u yields the optimal control:

$$u(t) = -R^{-1}B^T\left[P(t)x(t)+q(t)\right]. \tag{30}$$

The optimality conditions for $P(t)$ and $q(t)$ can be found by inserting $u(t)$ from (30) back into (26):

$$J^*(x(t),t) = \{x(t) - x_{REF}(t)\}^T Q\{x(t) - x_{REF}(t)\} + \{-R^{-1}B^T[P(t)x(t) + q(t)]\}^T R\{-R^{-1}B^T[P(t)x(t) + q(t)]\} +$$
$$x(t)^T P(t)x(t) + 2q(t)^T x(t) + r(t) + \{Ax(t) + B\{-R^{-1}B^T[P(t)x(t) + q(t)]\}\}^T P(t)x(t) +$$
(31)
$$x^T(t)P(t)\{Ax(t) + B\{-R^{-1}B^T[P(t)x(t) + q(t)]\}\} + x^T(t)\dot{P}(t)x(t) +$$
$$2\dot{q}(t)x(t) + 2q^T(t)\{Ax(t) + B\{-R^{-1}B^T[P(t)x(t) + q(t)]\}\} + \dot{r}(t)\}dt$$

Subtracting $J^*(x(t),t) = x^T(t)P(t)x(t) + 2q^T(t)x(t) + r(t)$ from both sides of the equation and expanding and collecting like terms, the equality in (31) becomes:

$$0 = \left\{ \begin{array}{l} x^T\left(Q - PBR^{-1}B^TP + A^TP + \dot{P} + PA\right)x + \\ \left(q^T\left(-4BR^{-1}B^TP + 2A\right) + 2\dot{q} - 2x_{REF}^TQ\right)x + \dot{r} + x_{REF}^TQx_{REF} \end{array} \right\} \quad (32)$$

The equality in (32) yields the optimality conditions for $P(t)$, $q(t)$ and $r(t)$. For $\{\ \}$ to equal 0 for all values of x, the following equations must hold:

$$-\dot{P}(t) = A^TP(t) + P(t)A + Q - P(t)BR^{-1}B^TP \quad (33)$$

$$-\dot{q}(t) = \left[A^T - 2PBR^{-1}B^T\right]q(t) - Qx_{REF}(t) \quad (34)$$

Equation (33) is the Ricatti equation which can be solved for $P(t)$ and (34) is the associated co-state equation which can be solved for $q(t)$, with their respective initial conditions as specified in (35).

$$\begin{array}{l} \dot{P}(T_F) = 0 \\ \dot{q}(T_F) = 0 \end{array} \quad (35)$$

As seen in (34), the complete reference trajectory for the time period T_I *to* T_F, $x_{REF}(t)$, must be available *a priori*. If the system is stabilizable, the steady state value of P can be found by solving the algebraic Ricatti equation, setting $\dot{P} = 0$:

$$0 = A^TP(t) + P(t)A + Q - P(t)BR^{-1}B^TP \quad (36)$$

The algebraic Ricatti equation can be solved explicitly for P by forming the system Hamiltonian, H, as follows [23]. Firstly, define $\lambda = Px$ and $\dot{\lambda} = \dot{P}x + P\dot{x}$. Plugging these equalities and equation (30) into (36), yields the differential equations (37) from which the Hamiltonian matrix of this system, H, can be formed:

$$\begin{array}{l} \dot{\lambda} = -Qx(t) - A^T\lambda \\ \dot{x} = Ax - BR^{-1}B^T\lambda \end{array} \quad (37)$$

Equivalently:

$$\begin{bmatrix}\dot{x}\\ \dot{\lambda}\end{bmatrix}=\begin{bmatrix}A & -BR^{-1}B^T\\ -Q & -A^T\end{bmatrix}\begin{bmatrix}x\\ \lambda\end{bmatrix}$$
$$=H\begin{bmatrix}x\\ \lambda\end{bmatrix} \tag{38}$$

Assuming the eigenvalues of H are distinct, H can be diagonalized as demonstrated:

$$\begin{bmatrix}\dot{z}_1(t)\\ \dot{z}_2(t)\end{bmatrix}=\begin{bmatrix}-\Lambda & 0\\ 0 & \Lambda\end{bmatrix}\begin{bmatrix}z_1\\ z_2\end{bmatrix} \tag{39}$$

See [26] for a method to find the steady-state optimal control when the eigenvalues of H are not distinct.

Defining the similarity transform which diagonalizes H as:

$$\begin{bmatrix}x(t)\\ \lambda(t)\end{bmatrix}=\begin{bmatrix}X_{11} & X_{12}\\ Y_{21} & Y_{22}\end{bmatrix}\begin{bmatrix}z_1(t)\\ z_2(t)\end{bmatrix} \tag{40}$$

where the columns $\begin{bmatrix}X_{11}\\ Y_{21}\end{bmatrix}$ and $\begin{bmatrix}X_{12}\\ Y_{22}\end{bmatrix}$ are the eigenvectors of H corresponding to the eigenvalues of the Hamiltonian system, $-\Lambda$ and Λ, respectively. Since the Hamiltonian system eigenvalues are the closed loop system eigenvalues and we have already asserted that the system is stabilizable, x and λ are assumed to be in the stable space of H. Since x and λ are assumed to lie in the stable subspace, they can be represented as a linear combination of the stable eigenvectors of H:

$$\begin{bmatrix}x\\ \lambda\end{bmatrix}=\begin{bmatrix}X_{11}\\ Y_{21}\end{bmatrix}z_1 \tag{41}$$

This matrix equation can be split into two equations separating x and λ; $x = X_{11}z_1$ and $\lambda = Y_{21}z_1$, then solving the x-equation for z_1 yields:

$$z_1 = X_{11}^{-1}x \tag{42}$$

Exploiting the linear relation between λ and Y, z_1 can be inserted as defined in (42), which can be plugged back into the original definition of λ:

$$\lambda = Px = Y_{21}z_1 = Y_{21}X_{11}^{-1}x, \tag{43}$$

and solved explicitly for P:

$$P = Y_{21}X_{11}^{-1} \tag{44}$$

Given the steady state value of P, the established reference, $x_{REF}(t)$, can be input to complete (34) enabling it to be integrated backwards in time from t = T_F to 0 to calculate the costate, $q(t)$. See [22] and [23] for a complete derivation.

4.1.2 **Example: Mt. Adams**

Calculation of discrete values of P and q followed by simulation through discrete integration of the system equation with u inserted as defined in (30), has been carried out for a sampled 2D trajectory over the terrain specified in Section 2.3. A very simplified helicopter model, with longitudinal states velocity, pitch angle and vehicle altitude, as linearized about the nominal velocity and pitch angle, to obtain the A and B matrices necessary for this implementation. Figure 4-1 below shows the 2D altitude reference tracking results along with the corresponding error values.

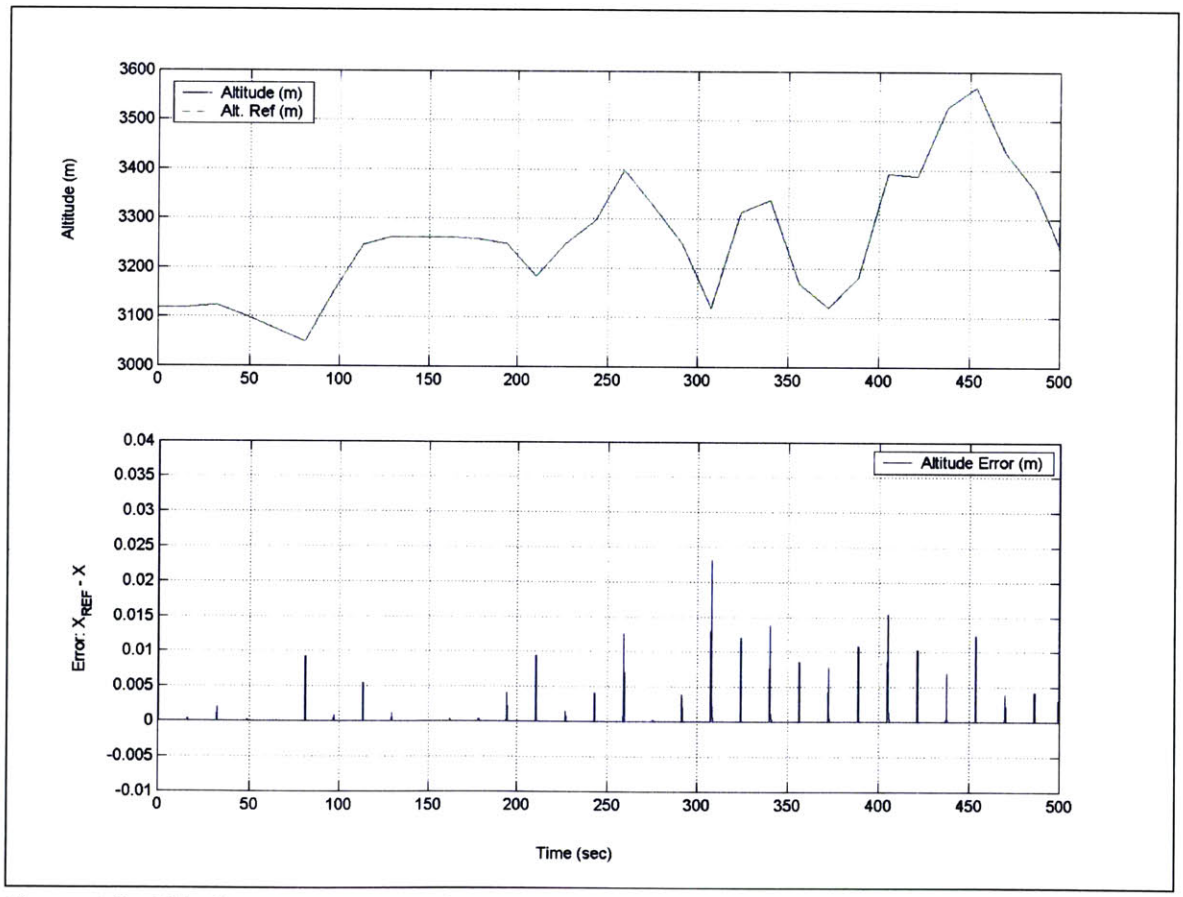

Figure 4-1: Altitude tracking data and corresponding error data from LQR tracking implementation on sample terrain around Mt. Adams in Washington State.

With this very simplified case there is hardly any tracking error. It is, however, important to note that although this controller can provide the very best performance to all of the specified requirements, it may be simultaneously suboptimal with respect to other unspecified design parameters or distinct test conditions. The results appear noteworthy in this application partially due to the fact that the plant to which the resulting LQR gains were applied was the linear model from which the gains were created. If the non-linear model or actual flight vehicle dynamics deviate significantly from the nominal states, the margins inherent to LQR may not be enough to guarantee the quality of performance.

Though Linear Quadratic Tracking produces a fine answer in theory, there are many issues presented in actual implementation which cause this method to be impractical for our application. Firstly, LQR tracking requires a single linearization of the plant model about its nominal states for the entire trajectory that is to be flown. This is obviously not going to be valid for all flight regimes, particularly for the mission objective requiring agile maneuvering between extremes of the vehicle flight envelope. Secondly, as noted in the problem definition, LQR tracking requires the complete reference trajectory to be input *a priori* for optimization of the LQR gains. Not only does this preclude the possibly of mission or trajectory replanning throughout the flight, it also assumes perfect accuracy of the reference. Especially at the very low altitudes above ground required by NOE flight, even the highest resolution map may not be able to warn of unexpected obstacles such as buildings, vegetation, and rocks or other geographic formations. This type of information can only be obtained through real-time sensor sweeps during flight. Finally, LQR yields no method for inputting state or control constraints to the problem. Thus, the resulting control trajectories of LQR are not necessarily realizable and may lead to control saturation, significantly affecting performance.

4.2 Model Predictive Control (MPC)

These problems posed by LQR are resolved by alternatively using Model Predictive Control (MPC). MPC is a repeating, finite horizon optimal control scheme which uses an internal model of the plant dynamics (prediction model) to predict and feed forward future errors, thus minimizing both the current error and the predicted future error for the

finite horizon length [21]. As noted in Section 1.2 the problem of terrain following guidance and control is highly constrained for a variety of reasons. The internal model can use the vehicle's predicted future positions to control the vehicle very close to the constraints, taking fuller advantage of the feasible flight envelope without constraint violation. In addition, through its recursive nature, MPC allows for the real-time incorporation of updated terrain and obstacle information, re-linearization of the plant around the current flight condition as well as dynamic path re-planning. The following section will present a systematic build up of basic model predictive control theory.

4.2.1 Unconstrained Solution

MPC, constrained or unconstrained, is set up much like the LQR tracking problem. It produces a set of control inputs to optimize the system states utilizing a quadratic form cost functional identical to standard linear quadratic tracking. However, specific to MPC (and finite horizon control in general), the cost is summed over the finite horizon interval length rather than over an infinite horizon. The time spanned by the number of time-steps ahead of the present, (N_P), whose penalty terms are included in the cost minimization is called the prediction horizon, ($H_P = N_P \cdot T_S$, where T_S is the sample rate of the control loop). Therefore, in this formulation, the following cost:

$$J_i = \sum_{k=i}^{i+N_P-1} (y_{k+1} - r_{k+1})^T Q_k (y_{k+1} - r_{k+1}) + u_k^T R_k u_k \,, \tag{45}$$

where Q_k is assumed to be a positive semi-definite matrix of error penalties for each time-step k and R_k is assumed to be a positive definite matrix of control penalties at each step as well, is minimized to determine the optimal sequence of commands, u_k, for the prediction horizon length, H_P. The choice of this horizon length is very important as it is well known that choosing an insufficient length H_P leads to instability. This must be balanced with the negative performance affects of an excessively long H_P due to the added computational effort required. Therefore, this subject will be dealt with in greater detail in Chapter 5 detailing the implementation used in this research. Once the control sequence has been determined, the first N_U controls, inputs u_i through u_{i+Nu}, are applied, and the calculation is repeated. The time spanning the number of controls applied before each MPC recalculation defines the MPC rate.

It is worth noting additionally that any performance objectives could be folded in here, making MPC online optimization a very versatile and effective way to deal with real-time reconfiguration and control needs. When the cost function described in (45) is minimized subject to the constraints defining obstacle free subspace, the resulting controls synthesize a trajectory with minimal deviation from the reference trajectory while avoiding encountered obstacles and terrain features.

Taking advantage of the MPC update rate, a discrete-time, non-linear plant model that is assumed and is linearized for each MPC update as follows:

(46) $$x(k+1) = Ax(k) + Bu(k)$$

For the purposes of this research, full state feedback is also assumed, therefore: $\hat{x}(k \mid k) = x(k) = y(k)$. See [21] for a detailed implementation of MPC using an observer. Without disturbances or measurement noise, the state can be predicted at time k for time $k+1$ to time $k + N_P$ by iterating the model in (46) yielding:

(47) $$\begin{bmatrix} \hat{x}(k+1|k) \\ \hat{x}(k+2|k) \\ \vdots \\ \hat{x}(k+N_P|k) \end{bmatrix} = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^{N_P} \end{bmatrix} x(k) + \begin{bmatrix} B & 0 & \cdots & 0 \\ AB+B & B & & 0 \\ \vdots & \vdots & & \vdots \\ \sum_{i=0}^{N_P-1} A^i B & \sum_{i=0}^{N_P-2} A^i B & \cdots & B \end{bmatrix} \begin{bmatrix} \hat{u}(k|k) \\ \hat{u}(k+1|k) \\ \vdots \\ \hat{u}(k+N_P|k) \end{bmatrix}$$

where $\hat{x}(k+i \mid k)$ and $\hat{u}(k+i \mid k)$ are the predicted state estimates at time $k+i$ of the state x and control u, respectively, each estimated at time k. For ease of manipulation, define each of the matrices in (47) as follows:

$$Z(k) = \begin{bmatrix} \hat{x}(k+1|k) \\ \hat{x}(k+2|k) \\ \vdots \\ \hat{x}(k+N_P|k) \end{bmatrix}, \quad \Psi = \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^{N_P} \end{bmatrix}, \quad \Theta = \begin{bmatrix} B & 0 & \cdots & 0 \\ AB+B & B & & 0 \\ \vdots & \vdots & & \vdots \\ \sum_{i=0}^{N_P-1} A^i B & \sum_{i=0}^{N_P-2} A^i B & \cdots & B \end{bmatrix}, \quad U(k) = \begin{bmatrix} \hat{u}(k|k) \\ \hat{u}(k+1|k) \\ \vdots \\ \hat{u}(k+N_P|k) \end{bmatrix}$$

thus (47) can be rewritten as $Z(k) = \Psi x(k) + \Theta U(k)$. With these definitions of the predicted states, $\hat{x}$, we can now rewrite (45), the cost function to be minimized, as:

(48) $$J_k = \|Z(k) - T(k)\|_{\Omega}^2 + \|U(k)\|_{\Gamma}^2$$

where Ω is the matrix of the state weighting matrices associated with each time-step in the horizon at time k:

$$\Omega = \begin{bmatrix} Q_k & 0 & \cdots & 0 \\ 0 & Q_{k+1} & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & Q_{k+N_P} \end{bmatrix},$$

Γ is the matrix of the control weighting matrices associated with each time-step in the horizon at time k:

$$\Gamma = \begin{bmatrix} R_k & 0 & \cdots & 0 \\ 0 & R_{k+1} & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ 0 & \cdots & 0 & R_{k+N_P} \end{bmatrix}, \text{ and}$$

$T(k)$ is defined as the vector of reference states spanning the prediction horizon at time k:

$$T(k) = \begin{bmatrix} x_{REF}(k+1|k) \\ \vdots \\ x_{REF}(k+1|k+N_P) \end{bmatrix}.$$

Then, defining an error vector, $\mathrm{E}(k)$, by the known quantities $T(k)$, Ψ and $x(k)$:

(49) $$\mathrm{E}(k) = T(k) - \Psi x(k)$$

the cost becomes:

(50) $$J_k = \left\|\Theta U(k) - \mathrm{E}(k)\right\|_{\Omega}^2 + \left\|U(k)\right\|_{\Gamma}^2$$

which when expanded equals:

(51) $$J_k = \mathrm{E}(k)^T \Omega \mathrm{E}(k) - 2U(k)^T \Theta^T \Omega \mathrm{E}(k) + U(k)^T \left[\Theta^T \Omega \Theta + \Gamma\right] U(k)$$

Since $\mathrm{E}(k)$, Θ, Q and R are known, we are actually minimizing the quadratic form:

$$J_k = const - U(k)^T G + U(k)^T H U(k)$$

where:

$$G = 2\Theta^T \Omega \mathrm{E}(k)$$

$$H = \Theta^T \Omega \Theta + \Gamma$$

Taking advantage of this form, the minimum can be found analytically by taking the gradient of J_k and setting it equal to zero. This yields:

(52) $$\nabla_{U(k)} J_k = -G + 2HU(k)$$

which can be solved for $U^*(k)$:

$$U^*(k) = \frac{1}{2} H^{-1} G \tag{53}$$

However, this alone is only enough to guarantee a stationary point. A necessary condition for minima of a convex function is a positive definite matrix of second derivatives or Hessian. To ensure $U^*(k)$ is actually the minimum, $\nabla_{U(k)} J_k$ is differentiated again with respect to $U(k)$, yielding the Hessian for this additional test:

$$\frac{\partial^2 J_k}{\partial U(k)^2} = 2H = 2(\Theta^T \Omega \Theta + \Gamma) \tag{54}$$

Since the weighting matrices Q_k and R_k, which make up Ω and Γ have been defined as positive semi-definite and positive definite, respectively, the Hessian for this cost function, (54), is positive definite and thus the $U^*(k)$ derived in (53) is indeed the optimal control trajectory. See [21] for a complete derivation.

4.2.2 **Constrained MPC**

The cost function described in the above can also be minimized subject to linear inequality constraints on the inputs. These constraints, on the control input and state, take the following form, borrowing from the notation established in Section 4.2.1:

$$u_{i,\min} \le u(k+i \mid k) \le u_{i,\max} \tag{55}$$

$$x_{i,\min} \le x(k+i \mid k) \le x_{i,\max} \tag{56}$$

The constraint equations listed in (55) and (56) both represent hard constraints on the optimization, therefore can never be violated. This can present a problem in the optimization when the constraints are set too tightly, leaving no feasible solution space. Great care has been taken in this research to assure that state constraints are set in such a way as to provide a non-empty convex feasible solution space. This will be discussed in further detail in Chapter 6. For cases when a convex feasible solution space is not as forthcoming, soft constraints which may be violated if required to maintain feasibility of the optimization, may be added in the following form:

$$\begin{aligned} & x_{i,\min} - s \le x(k+i \mid k) \le x_{i,\max} + s \\ & s \ge 0 \end{aligned} \tag{57}$$

where s is a slack variable that is introduced to the optimization carrying a significant penalty in the cost function. This penalty causes s to equal zero enforcing this constraint, except in cases where there is no other feasible solution outside of constraint violation. As previously stated, however, only hard constraints will be applied in this research, thus revising the problem statement to the following:

$$\min_{u_k} \quad J_i = \sum (x_{k+1} - r_{k+1})^T Q_k (x_{k+1} - r_{k+1}) + u_k^T R_k u_k$$

$$\text{subject to} \quad u_{i,\min} \le u(k+i \mid k) \le u_{i,\max} \tag{58}$$

$$x_{i,\min} \le x(k+i \mid k) \le x_{i,\max}$$

With the addition of constraints, MPC can no longer be solved analytically as in the unconstrained case. This necessitates a numerical solver. For this research, the quadratic program solver SQOPT, commercially available from Stanford Business Software, Inc., was selected. SQOPT is a set of Fortran subroutines, with a C-code wrapper, which use a two-phase, active set-type method [27]. These methods are most efficient when many constraints or bounds are active at the solution, therefore making them a good fit for the highly constrained problem presented in this research. For a discussion of the computational speed merits of SQOPT in relation to the Matlab quadratic program solver, *quadprog*, see [28]. For an in-depth description of the SQOPT algorithm, see [15] and for information on how to obtain SQOPT, see [29].

4.2.3 **Example: Mt. Adams**

A sample implementation of standard constrained MPC was carried out over a section of Mt. Adams to compare to the tracking errors of LQR tracking. This implementation also serves as a performance baseline not only in tracking but in computational run-time for comparison to a simplified MPC formulation which will be described in the following section.

Figure 4-2 shows the 3-dimensional plot of the sample terrain area surrounding Mt. Adams in Washington State and the resultant trajectory (plotted in white) following the reference at 6 meters above the terrain at a 10 knot nominal velocity. In this plot, it is

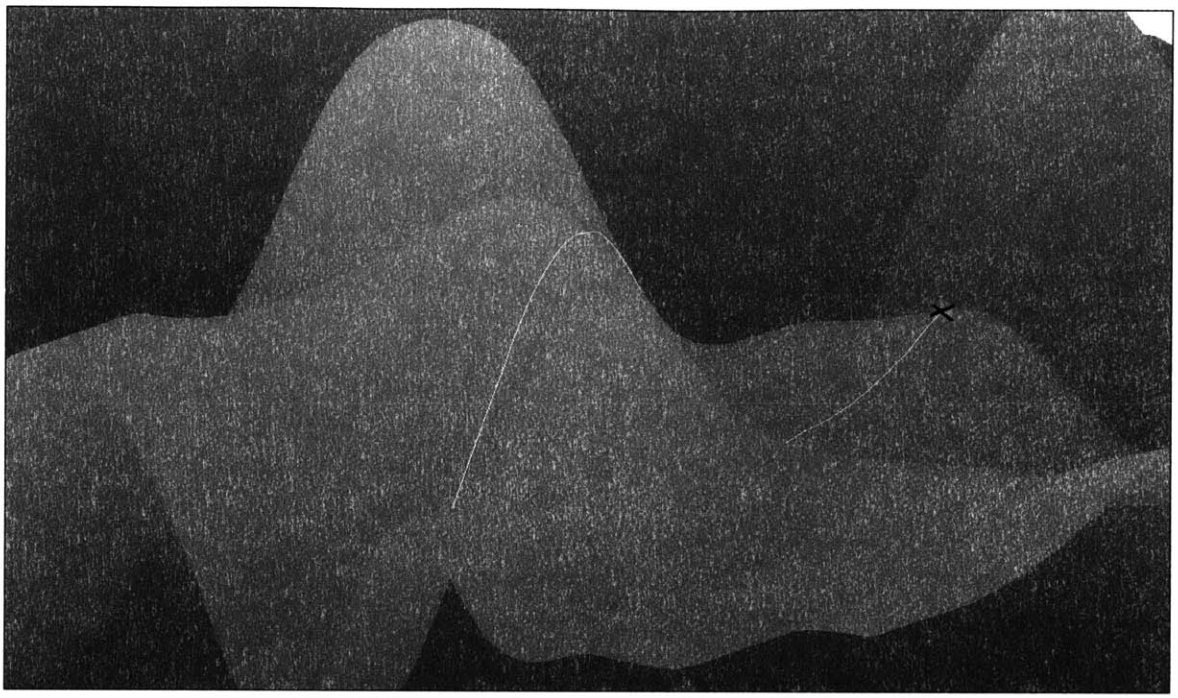

Figure 4-2: 3-Dimensional plot of the sample terrain area plotted with the resultant trajectory (in white) from standard MPC terrain following.

apparent that the 6 meter above ground reference is being obeyed even as the trajectory crests the first hill and makes its way towards the final waypoint marked by the black 'x'. The significant tracking error values for each axis are organized in Table 4-1 followed by the average optimizer computation time per iteration gathered using the Matlab scripts *tic* and *toc* which start and stop a stop-watch timer and print the number of seconds required for the operation conducted between the two.

Table 4-1: Maximum, Minimum and Average Error Values for Standard MPC Terrain Following

	Minimum	Maximum	Average (abs)
X-Tracking Error	-0.2892 m	2.1005 m	0.04 m
Y-Tracking Error	-0.2471 m	2.4047 m	0.077 m
Z-Tracking Error	-0.3426 m	0.6331 m	0.0817 m
Average Optimizer Computational Time: 3.438 seconds/MPC control iteration*			

* Runs for run-time analysis were conducted with SQOPT, a quadratic programming optimizer coded in Fortran with a C-wrapper and mexed for use in a Matlab Simulink simulation environment. The simulation was conducted on a computer with an Intel Pentium 4 processor, a 2.2 GHz processing speed and 1.04 GB of RAM.

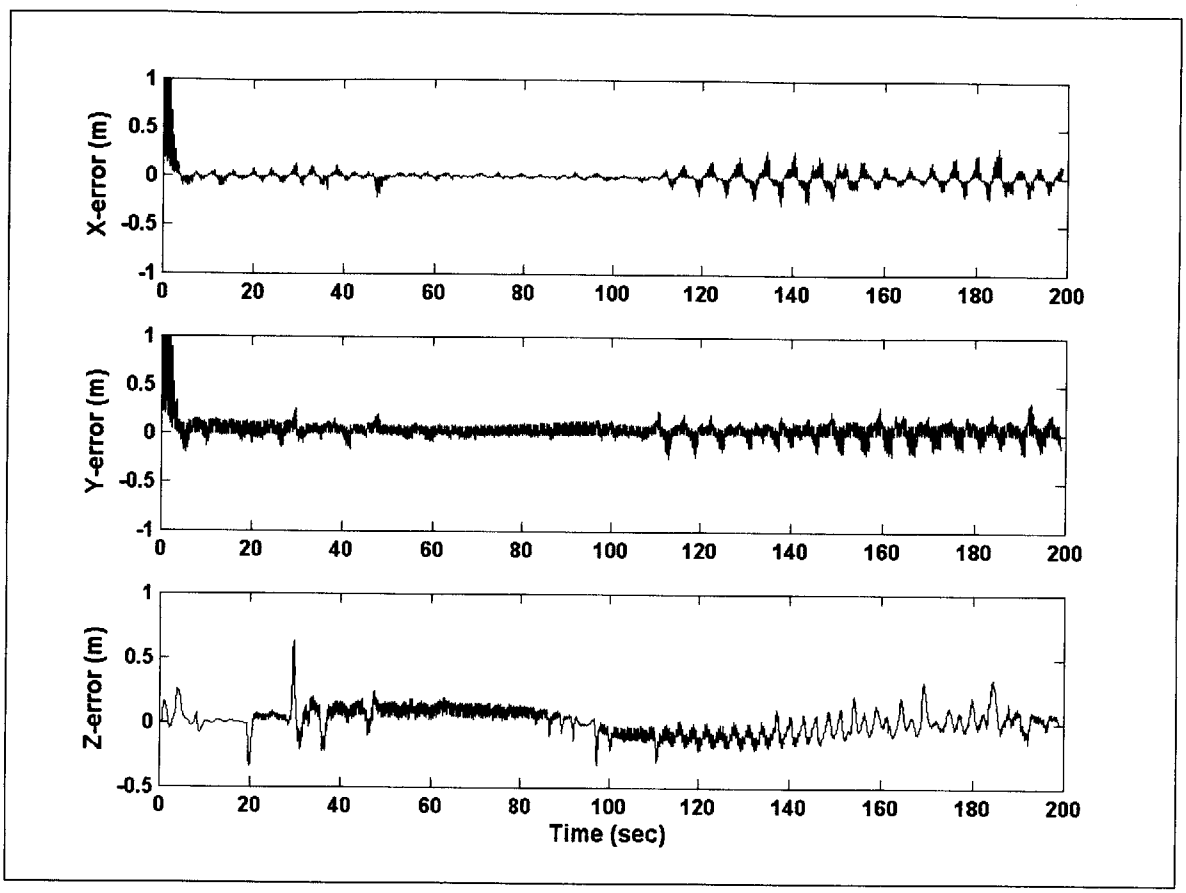

Figure 4-3: Tracking errors from standard MPC applied to terrain following over sample terrain.

The tracking errors, $\boldsymbol{e} = (\boldsymbol{x}-\boldsymbol{x}_{REF})$, are plotted versus time for each axis X, Y and Z, in Figure 4-3. Though the error tracking values are very good for this MPC implementation, MPC must calculate (N_P)*(Number of Control Inputs) separate controls to span the entire prediction horizon. For this research that amounts to 201*4 = 804 controls along a 10 second prediction horizon at the control rate of 20 Hz with four helicopter control inputs. Such computational requirements lead to an optimizer computational time on the order of several seconds on a PC[†]. This would make an MPC-based formulation impractical in real-time implementation. This motivates the need for a formulation which cuts down on computational complexity, allowing the optimizer to solve for the desired trajectory at the specified real-time rate.

[†] Runs for run-time analysis were conducted with SQOPT, a quadratic programming optimizer coded in Fortran with a C-wrapper and mexed for use in a Matlab Simulink simulation environment. The simulation was conducted on a computer with an Intel Pentium 4 processor, a 2.2 GHz processing speed and 1.04 GB of RAM.

4.3 MPC FOR REAL-TIME IMPLEMENTATION

Even incorporating the computational speed advantage of SQOPT over Matlab's quadratic program solver *quadprog*, it is evident that for MPC to be applied real-time, a less complex formulation which reduces the problem size must be proposed. The following section proposes such a formulation. In this real-time formulation two simplifications have been incorporated into the problem definition to alleviate the burden on the optimizer.

4.3.1 Perturbational Linearization

First, the prediction model is generated for MPC by perturbational linearization of the 6-degree-of-freedom non-linear model (described in Section 2.2) about nominal output and control trajectories. Perturbational linearization assumes that the input and output states of a non-linear model can be expressed as a sum of a nominal trajectory and a perturbation about that nominal. The nominal trajectory captures the key non-linearities of the system in its operating regime and the perturbational component is assumed to have a linear mapping between the input perturbation about the nominal controls and the perturbational output about the nominal output. Therefore, the non-linearity of the system response can be preserved while the linearity of the perturbational response can be exploited in the control strategy. With an established nominal control trajectory, $\mathbf{u_0}$, and nominal output state trajectory, $\mathbf{y_0}$, perturbational analysis can be used to linearize the model around these trajectories as follows. Given a non-linear model, $\dot{x} = F_1(x,u)$, the nominal state and control trajectories are defined by:

(59) $$\dot{x}_0 = F_1(x_0, u_0)$$

By perturbing the nominal trajectory, the perturbational linearization of the non-linear model, $\dot{x} = F_1(x,u)$, is obtained:

(60) $$\dot{x} = \dot{x}_0 + \delta\dot{x} = F_1(x_0,u_0) + \left.\frac{\partial F_1}{\partial x}\right|_{x_0,u_0} \delta x + \left.\frac{\partial F_1}{\partial u}\right|_{x_0,u_0} \delta u$$

(61) $$y = y_0 + \delta y = F_2(x_0,u_0) + \delta y$$

Thus the perturbational state and outputs, δx and δy, are defined by the linearization of F_1 and F_2 around the nonlinear trajectories (x_0, u_0) yielding:

(62) $$\delta\dot{x} = A(x_0, u_0)\delta x + B(x_0, u_0)\delta u$$

(63) $$\delta y = C(x_0, u_0)\delta x + D(x_0, u_0)\delta u$$

This linearization can then be exploited by charging the optimizer to solve a convex quadratic optimization problem to find the optimal linearized perturbational control, δu. From this, the control trajectory to be applied to the vehicle, $u = u_0 + \delta u$, can be found without requiring non-linear programming to find the optimal u.

In this implementation of perturbational linearization, the nominal control trajectory ($\mathbf{u_0}$) is set as the trim controls for the initial iteration, and for every step thereafter it is updated as the prediction output of the previous optimization cycle. The nominal output trajectory ($\mathbf{y_0}$) is calculated using the nonlinear input/output relationship defined in the helicopter model by integrating the helicopter equations of motion for each step in the horizon length, with initial conditions of the current position and the nominal control input.

4.3.2 Control Basis Functions

To further decrease complexity, a parameterization of the perturbational control input trajectory along the prediction horizon is proposed using *control basis functions*:

(64) $$u(k+i \mid k) = u_0(k+i) + \delta u = u_0(k+i) + \sum_{n=1}^{N_{BASES}} \beta_n(i)\alpha_n(k)$$

where $u(k+i \mid k)$ is the value of control input u predicted at time k for the time $k+i$, and i is within the prediction horizon, H_P. The β_n are predefined vector sequences of control spanning the length of the prediction horizon to be optimally scaled and summed to approximate the optimal control trajectory for the horizon at time t_k. The basis functions, β_n, are subject to the condition that the basis matrix:

(65) $$\mathrm{B} = \begin{bmatrix} | & | & & | \\ \beta_1 & \beta_2 & \cdots & \beta_{N_{BASES}} \\ | & | & & | \end{bmatrix},$$

is full rank and invertible. Given these predefined sequences, β_n, the complexity of the optimization problem can be greatly reduced with comparable performance by charging the optimizer to find optimal scale factors ($\boldsymbol{\alpha}$) for each basis function perturbational

control trajectory along the length of the horizon (optimizer computes N_{BASES} scale-factors), as opposed to finding an optimal trajectory of perturbational controls for each step along the horizon (optimizer computes N_P controls). The following section will describe the method of applying control basis functions in the MPC framework and will formally discuss different families of candidate basis functions, the results attainable with each family, and ultimately the basis functions selected for this research.

4.3.2.1 *Control Basis Function Implementation*

Taking advantage of this input parameterization, rather than explicitly calculating $A(x_0, u_0)$, $B(x_0, u_0)$, $C(x_0, u_0)$ and $D(x_0, u_0)$, an input-output mapping is constructed using the linearity between input and output. Combining the basis functions, β_n, into a matrix, B, the vector $\delta \boldsymbol{u}$ can be written as a linear combination of the basis functions as follows:

$$\begin{bmatrix} \delta u_k \\ \delta u_{k+1} \\ \vdots \\ \delta u_{k+H-1} \end{bmatrix} = \begin{bmatrix} | & | & & | \\ \beta_1 & \beta_2 & \cdots & \beta_{N_{BASES}} \\ | & | & & | \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{N_{BASES}} \end{bmatrix} \tag{66}$$

$$\delta \boldsymbol{u}_k \quad = \quad \mathrm{B} \quad \boldsymbol{\alpha}$$

Conveniently, based on the linearized model, the vector $\delta \boldsymbol{y}$ can be described as the same linear combination of a response matrix, S, made up of response vectors, $\boldsymbol{s_n}$, associated with each basis function, β_n. To derive this response matrix, we revisit the continuous perturbational linear model from (63):

$$\delta \dot{y} = C(x_0, u_0)\delta x + D(x_0, u_0)\delta u$$

Discretizing this model into the output space yields the following:

$$\delta y_{k+1} = G(x_0, u_0)\delta x_k + H(x_0, u_0)\delta u_k \tag{67}$$

By substituting a single basis function (β_n) for the control vectors spanning the prediction horizon at a given time t_k, U(k), a response vector also spanning the prediction horizon associated with that particular basis function ($\boldsymbol{s_n}$) can be obtained through the following recursion:

$$s_{n,i} = G(x_0, u_0)\delta x_i + H(x_0, u_0)\beta_{n,i} \tag{68}$$

where n indicates the associated basis function (column of B) and i indicates the associated step along the prediction horizon. k indicates the current time-step indicating the initial condition for the recursion. For the first time-step in the horizon associated with time t_k, δx_k should be used and s_{n-1} should be used as δx_i thereafter. In other words, as described in [10], each control sequence, β_n, is input to the linear system yielding a corresponding output trajectory vector, $\boldsymbol{s_n}$, for the prediction horizon. The response vectors associated with each basis function can then be combined to form a response corresponding to the basis matrix, B:

$$S = \begin{bmatrix} | & | & & | \\ s_1 & s_2 & \cdots & s_{N_{BASES}} \\ | & | & & | \end{bmatrix} \tag{69}$$

Note that unlike the basis matrix (B) which must be calculated once *a priori*, the current state, δx_k, is necessary in the generation of the response matrix, S, therefore must be regenerated, updating the response for the current linear regime each time MPC is run. Thus, based on the linear model, the output $\delta\mathbf{y}$ can be written as the same linear combination of each basis function output, described by the output matrix, S:

$$\begin{bmatrix} \delta y_k \\ \delta y_{k+1} \\ \vdots \\ \delta y_{k+H-1} \end{bmatrix} = \begin{bmatrix} | & | & & | \\ s_1 & s_2 & \cdots & s_{N_{BASES}} \\ | & | & & | \end{bmatrix} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{N_{BASES}} \end{bmatrix} \tag{70}$$

$$\boldsymbol{\delta y_k} \quad = \quad \boldsymbol{S} \quad \boldsymbol{\alpha}$$

The cost function can then be described in terms of the nominal and perturbed states:

$$\mathbf{y} = \mathbf{S}\boldsymbol{\alpha} + \mathbf{y_0} \tag{71}$$

$$\mathbf{u} = \mathrm{B}\boldsymbol{\alpha} + \mathbf{u_0} \tag{72}$$

Thus, the problem can be redefined in its final form assuming full state feedback:

$$\min_{u_k} J_i = \sum_{k=i}^{i+H-1} \left[\left(S\alpha + x_0 - x_{REF_{k+1}} \right)^T Q_k \left(S\alpha + x_0 - x_{REF_{k+1}} \right) + \left(\mathrm{B}\alpha + u_0 \right)^T R \left(\mathrm{B}\alpha + u_0 \right) \right] \tag{73}$$

subject to $\quad u_{i,\min} \le u(k+i \mid k) \le u_{i,\max}$

$$x_{i,\min} \le x(k+i \mid k) \le x_{i,\max}$$

4.3.2.2 *Candidate Basis Functions*

As suggested in Section 4.3.2.1, computational simplification can be attained through the use of control basis functions to create the input/output system map or prediction model within MPC. Notice that no requirements are placed on the shape or magnitude of the functions β_n other than the stipulation that the basis matrix, B, is full rank. With that in mind, the following section will propose 6 families of orthogonal functions as potential control basis functions. They will each be systematically applied to the terrain following guidance and control problem using the implementation proposed in Section 4.3.2.1. The results will be analyzed for tracking performance and computational simplification and a final basis function will be selected for the duration of this thesis work. Note that all of the basis functions proposed are normalized and span the prediction horizon, N_P time-steps long. For all basis function comparison simulations, a 10 second prediction horizon ($H_P = 10$), 20 Hz control rate ($N_P = 201$) and 2 Hz MPC update rate are set.

The first three basis function families introduced are spaced out in time as the ramps are in [10]. The box functions are the first and simplest basis functions introduced to this study. They simply consist of a pulse signal lasting n_1 time-steps, where n_1, the length of each box, equals the number of time-steps in the prediction horizon divided by the number of basis functions being applied. This forms the following basis matrix:

$$B = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 1 & 0 & 0 & & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 1 & 0 & \cdots & 0 \\ 0 & 1 & 0 & & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 1 & & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix}, \text{ spanning the time horizon, } \begin{bmatrix} t_k \\ t_{k+1} \\ \vdots \\ t_{k+n_1+1} \\ t_{k+n_1+2} \\ \vdots \\ t_{k+2n_1+1} \\ \vdots \\ t_{k+N_P-1} \end{bmatrix}. \tag{74}$$

Figure 4-4 shows a graphic illustration of 5 basis functions spanning the prediction horizon.

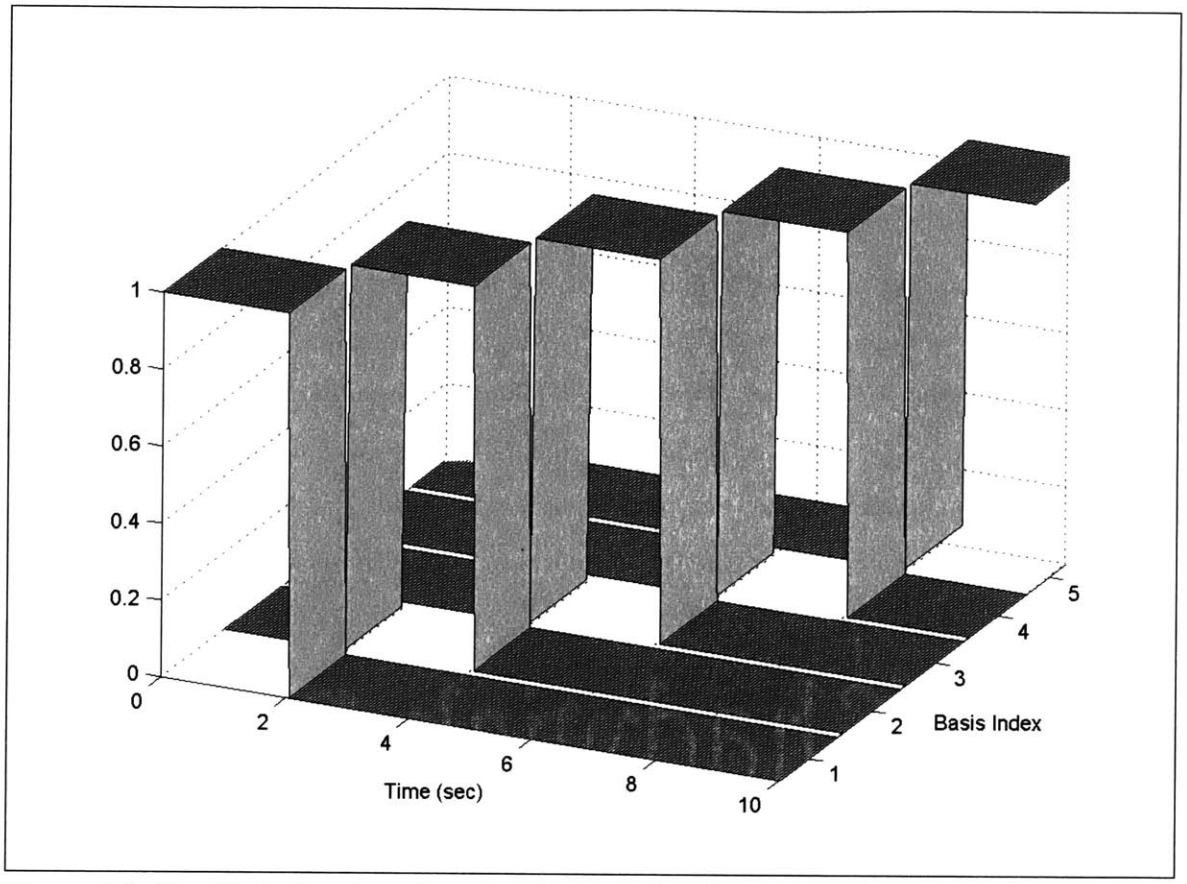

Figure 4-4: Box Basis functions formulated across the 10 second prediction horizon, (5 bases applied).

The second family of basis functions introduced simply overlap the box functions. The overlapping boxes are formed similarly to the traditional boxes, however every box spans $2n_1$ steps, leaving the last box to only span n_1 steps. This forms the following basis matrix and is more clearly illustrated in Figure 4-5:

$$B = \begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 1 & 1 & 0 & & 0 \\ \vdots & \vdots & \vdots & \cdots & \vdots \\ 0 & 1 & 1 & & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 1 & & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \end{bmatrix} \text{ spanning the time horizon, } \begin{bmatrix} t_k \\ \vdots \\ t_{k+n_1+1} \\ \vdots \\ t_{k+2n_1+1} \\ \vdots \\ t_{k+3n_1+1} \\ \vdots \\ t_{k+N_P-1} \end{bmatrix}. \tag{75}$$

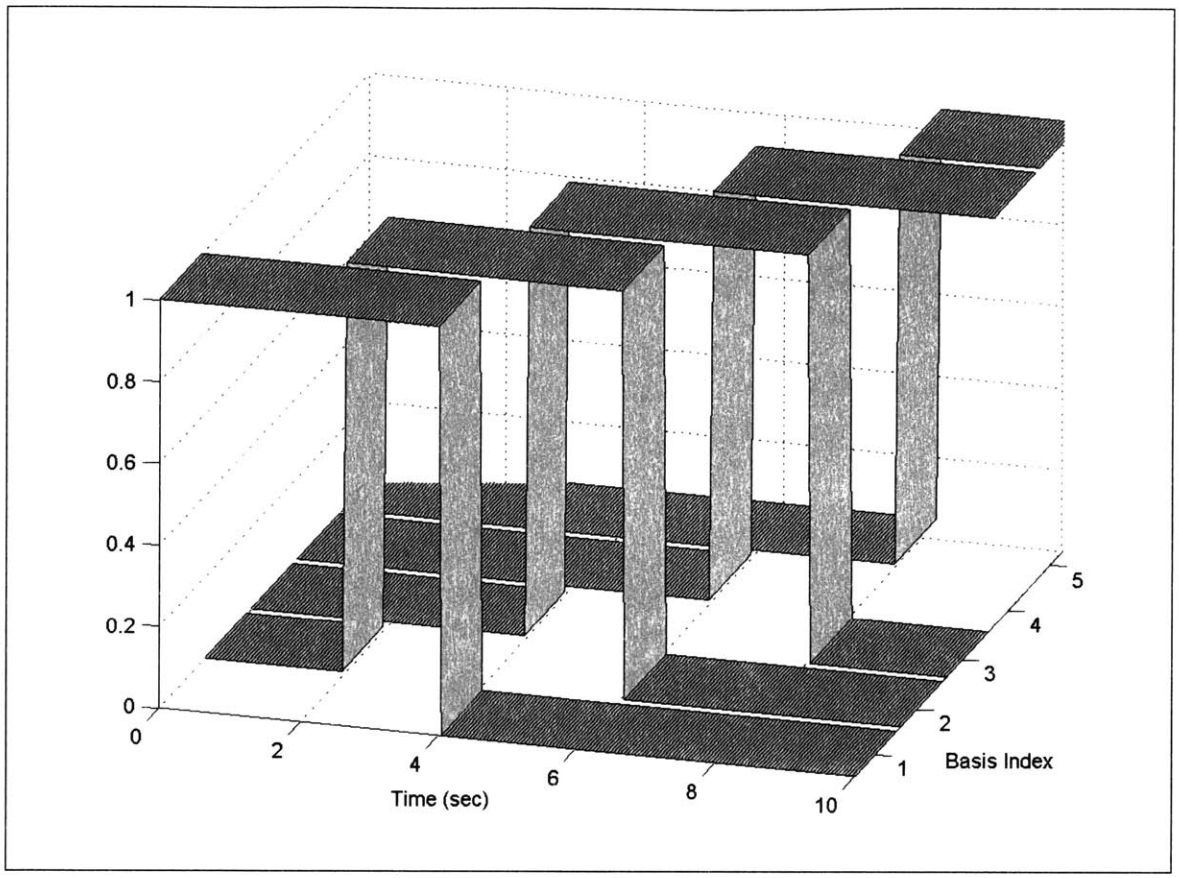

Figure 4-5: Overlapping Box Basis functions formulated across the 10 second prediction horizon, (5 bases applied).

The tent functions investigated in this research are defined to ramp up to a peak value of 1 in the first n_1 time-steps. Then they ramp back down to zero in the next n_1 time-steps, defining each tent to span $2n_1$ steps total. As with the two box function families, the span of each individual basis function depends on the number of basis functions applied. The corresponding tent basis matrix B is defined in (76).

$$\text{(76)} \qquad \mathrm{B} = \begin{bmatrix} du & 0 & 0 & \cdots & 0 \\ 2du & 0 & 0 & & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ p & du & 0 & \cdots & 0 \\ p-du & 2du & 0 & & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & p & du & & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & p \end{bmatrix} \text{ spanning the time horizon, } \begin{bmatrix} t_k \\ t_{k+1} \\ \vdots \\ t_{k+n_1} \\ t_{k+n_1+1} \\ \vdots \\ t_{k+2n_1} \\ \vdots \\ t_{k+N_P-1} \end{bmatrix}.$$

For example, a ramp duration of n_1= 1 second (as applied in [10]) results from applying 10 basis functions. See Figure 4-6 for graphic illustration.

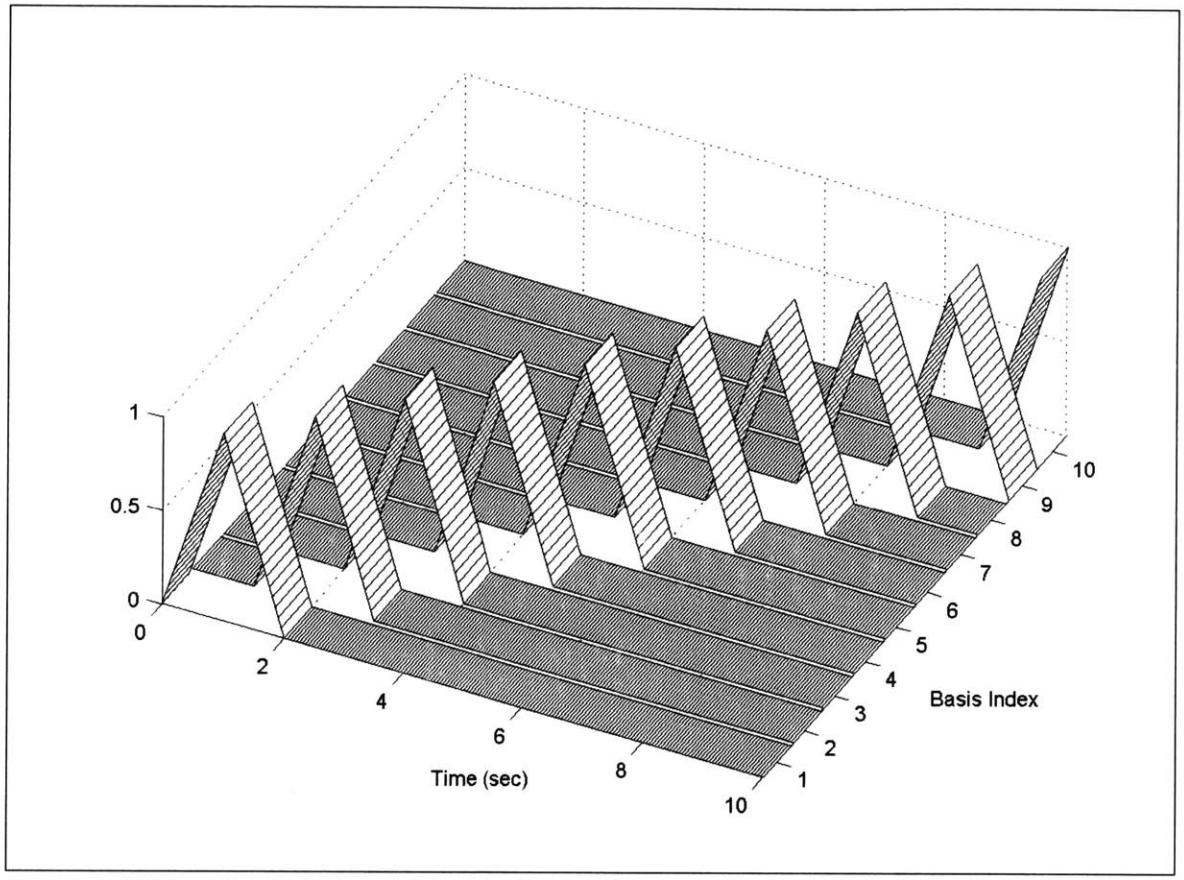

Figure 4-6: Tent Basis functions formulated across the 10 second prediction horizon, (10 bases applied).

Beyond the repeated functions spaced out in time, we will explore three families of orthogonal polynomial functions, the first two of which are related under the broad category of Gegenbauer polynomials [30]. Gegenbauer polynomials are defined by the generating function:

$$(77) \qquad \left(1-2xt+t^2\right)^{-\lambda} = \sum_{n=0}^{\infty} C_n^{\lambda}(x)t^n, \qquad |t|<1, \qquad |x|\le 1,$$

where $\lambda > -\frac{1}{2}$. The orthogonality property for these polynomials is given by:

$$(78) \qquad \int_{-1}^{1}(1-x^2)^{\lambda-\frac{1}{2}} C_n^{\lambda}(x)C_m^{\lambda}(x)dx = 0, \qquad m \neq n$$

and the governing differential equation is:

$$(79) \qquad \left(1-x^2\right)\ddot{y} - (2\lambda+1)x\dot{y} + n(n+2\lambda)y = 0\,.$$

The first basis function polynomial family investigated is the family of Chebyshev polynomials. Chebyshev polynomials are the solution to Chebyshev's Differential Equation:

$$(80) \qquad (1-x^2)\ddot{y} - x\dot{y} + n^2 y = 0$$

where n is a real number. Defined using the Gegenbauer recursion relation, Chebyshev polynomials are defined as:

$$(81) \qquad \mathrm{B}_{Cheb_0}(x) = 1, \qquad \mathrm{B}_{Cheb_0}(x) = \frac{n}{2}\lim_{\lambda\to 0}\frac{C_n^{\lambda}(x)}{\lambda}, \qquad n = 1,2,3,\ldots$$

This can alternatively be expressed by Rodrigues' Formula as follows:

$$(82) \qquad \mathrm{B}_{Cheb_N}(x) = \frac{2}{\pi}\frac{\sqrt{1-x^2}}{(-1)^n(2n-1)(2n-3)\ldots 1}\frac{d^n}{dx^n}\left(1-x^2\right)^{n-\frac{1}{2}}.$$

Chebyshev Polynomials, B_{Cheb_N}, form a complete orthogonal set on the interval $-1 \le x \le 1$ with respect to a weighting function, $\frac{1}{\sqrt{1-x^2}}$. It can be shown that:

$$(83) \qquad \int_{-1}^{1}\frac{1}{\sqrt{1-x^2}}\mathrm{B}_{Cheb_N}(x)\mathrm{B}_{Cheb_M}(x)dx = 0 \qquad M \neq N$$

By using this orthogonality, a piecewise continuous function $f(x)$ in $-1 \le x \le 1$ can be expressed in terms of Chebyshev polynomials, resulting in the orthogonal series expansion known as a Fourier-Chebyshev Series expansion [30]:

$$(84) \qquad \sum_{N=0}^{\infty} C_N \mathrm{B}_{Cheb_N}(x) = \begin{cases} f(x) & \textit{where } f(x) \textit{ is continuous} \\ \dfrac{f(x^-)+f(x^+)}{2} & \textit{at discontinuities} \end{cases},$$

where

$$(85) \qquad C_N = \begin{cases} \dfrac{1}{\pi}\displaystyle\int_{-1}^{1}\frac{1}{\sqrt{1-x^2}} f(x)\mathrm{B}_{Cheb_N}(x)dx & n = 0 \\ \dfrac{2}{\pi}\displaystyle\int_{-1}^{1}\frac{1}{\sqrt{1-x^2}} f(x)\mathrm{B}_{Cheb_N}(x)dx & n = 1,2,\ldots \end{cases}.$$

The first 10 Chebyshev polynomials are plotted in Figure 4-7.

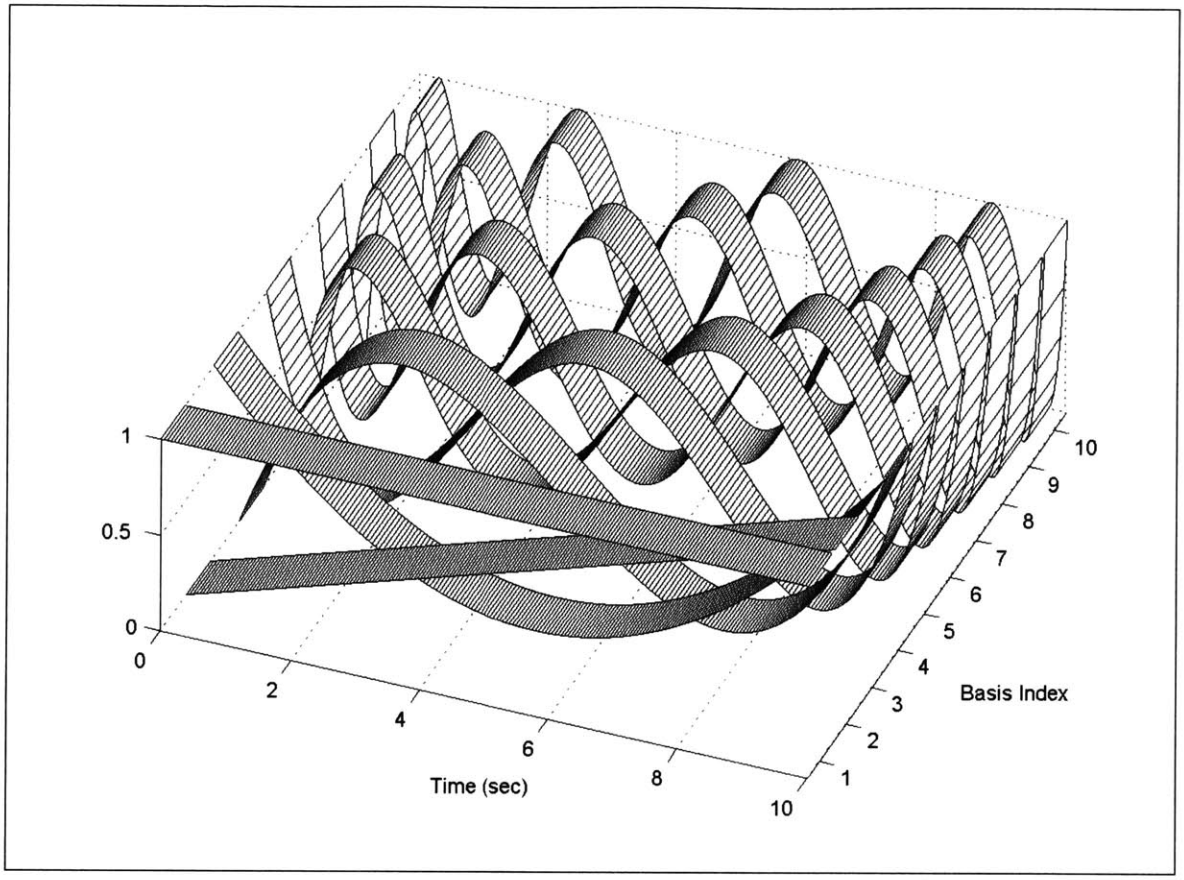

Figure 4-7: Chebyshev polynomials plotted along the prediction horizon from 0^{th} order to 9^{th} order.

The second polynomial family investigated is Legendre polynomials. Legendre polynomials are the solution to Legendre's Differential Equation:

(86) $$(1-x^2)\ddot{y} - 2x\dot{y} + n(n+1)y = 0$$

where n is a real number. Defined using the Gegenbauer recursion relation, with $\lambda = \frac{1}{2}$, Legendre polynomials are defined as:

(87) $$\mathrm{B}_{Legend_N}(x) = \sum_{k=0}^{[n/2]} \frac{(-1)^k (2n-2k)! x^{n-2k}}{2^n k!(n-k)!(n-2k)!}, \qquad n = 0,1,2,\ldots$$

This can alternatively be expressed by Rodrigues' Formula as follows:

(88) $$\mathrm{B}_{Legend_N}(x) = \frac{1}{2^n n!} \frac{d^n}{dx^n}\left(x^2-1\right)^n, \qquad n = 0,1,2,\ldots$$

Legendre Polynomials, B_{Legend_N}, form a complete orthogonal set on the interval $-1 \le x \le 1$, thus:

$$(89) \qquad \int_{-1}^{1} \mathrm{B}_{Legend_N}(x)\mathrm{B}_{Legend_M}(x)dx = 0 \qquad M \neq N$$

By using this orthogonality, a piecewise continuous function $f(x)$ in $-1 \leq x \leq 1$ can be expressed in terms of Legendre polynomials, resulting in the orthogonal series expansion known as a Fourier-Legendre Series expansion [30]:

$$\sum_{N=0}^{\infty} C_N \mathrm{B}_{Legend_N}(x) = \begin{cases} f(x) & where \;\; f(x) \;\; is \;\; continuous \\ \dfrac{f(x^-) + f(x^+)}{2} & at \;\; discontinuities \end{cases},$$

where

$$C_N = \frac{2n+1}{2} \int_{-1}^{1} f(x)\mathrm{B}_{Legend_N}(x)dx.$$

The first 10 Legendre polynomials are plotted in Figure 4-8.

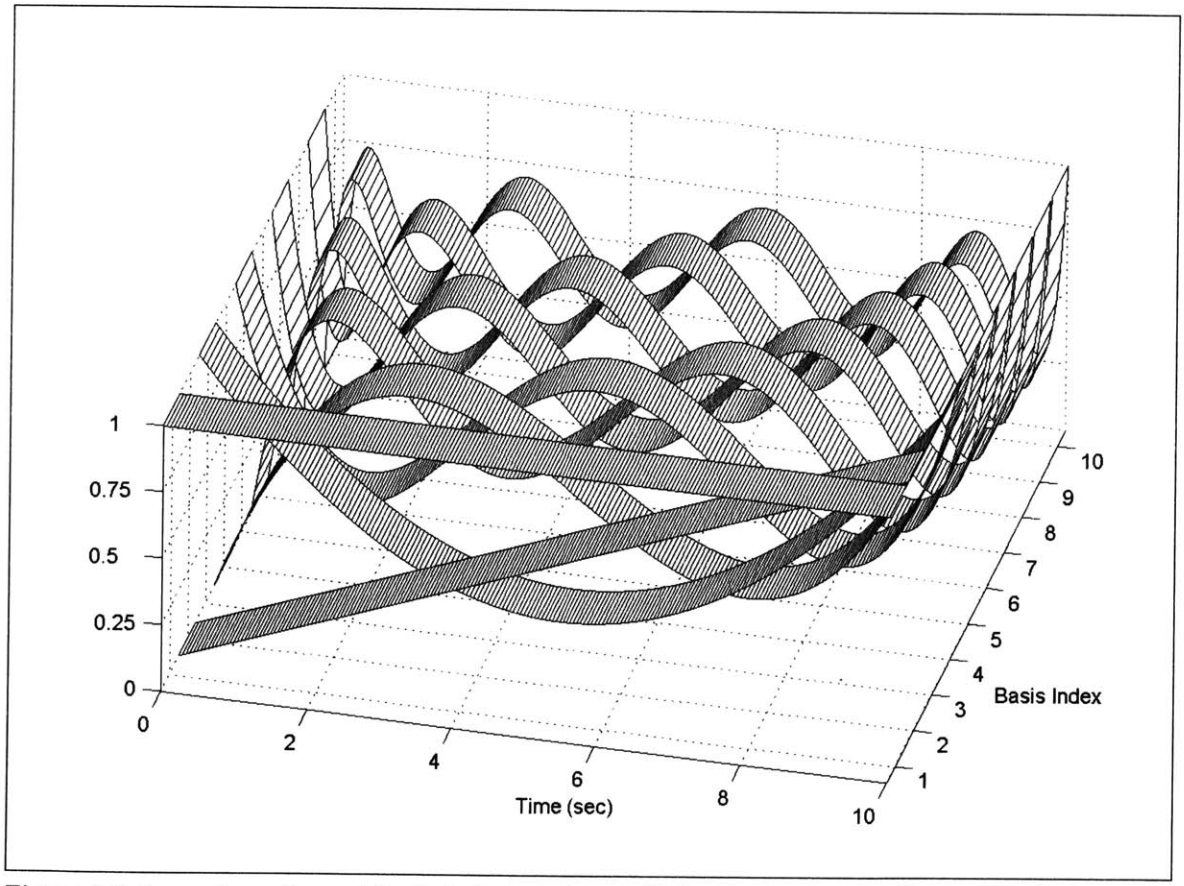

Figure 4-8: Legendre polynomials plotted along the prediction horizon from 0^{th} order to 9^{th} order.

The final family of basis functions being investigated is Laguerre polynomials. Laguerre polynomials are defined by the generating function:

$$(90) \qquad (1-t)^{-1} e^{\left[\frac{-xt}{1-t}\right]} = \sum_{n=0}^{\infty} \mathrm{B}_{Laguerre}(x) t^n, \qquad |t| < 1, \qquad 0 \le x < \infty,$$

and are the solutions to Laguerre's Differential Equation:

$$(91) \qquad x\ddot{y} - (1-x)\dot{y} + ny = 0$$

where n is a real number. Expressing the exponential term as a series, $\mathrm{B}_{Laguerre}$ becomes:

$$(92) \qquad \mathrm{B}_{Laguerre_n}(x) = \sum_{k=0}^{n} \frac{(-1)^k n! x^k}{(k!)^2 !(n-k)!}, \qquad n = 0,1,2,\ldots$$

This can alternatively be expressed by Rodrigues' Formula as follows:

$$(93) \qquad \mathrm{B}_{Laguerre_n}(x) = \frac{e^x}{n!} \frac{d^n}{dx^n}\left(x^n e^{-x}\right), \qquad n = 0,1,2,\ldots$$

Laguerre Polynomials, $\mathrm{B}_{Laguerre_n}$, form a complete orthogonal set on the interval $0 < x < \infty$, with respect to the weighting function e^{-x}, thus

$$(94) \qquad \int_0^{\infty} e^{-x} \mathrm{B}_{Laguerre_n}(x) \mathrm{B}_{Laguerre_m}(x) dx = 0 \qquad m \neq n$$

By using this orthogonality, a piecewise continuous function $f(x)$ can be expressed in terms of Laguerre polynomials, resulting in the orthogonal series expansion known as a Fourier-Laguerre Series expansion [30]:

$$(95) \qquad \sum_{n=0}^{\infty} C_n \mathrm{B}_{Laguerre_n}(x) = \begin{cases} f(x) & \text{where } f(x) \text{ is continuous} \\ \dfrac{f(x^-) + f(x^+)}{2} & \text{at discontinuities} \end{cases},$$

where:

$$(96) \qquad C_n = \int_0^{\infty} e^{-x} f(x) \mathrm{B}_{Legend_n}(x) dx.$$

The first 10 Legendre polynomials are plotted in Figure 4-9.

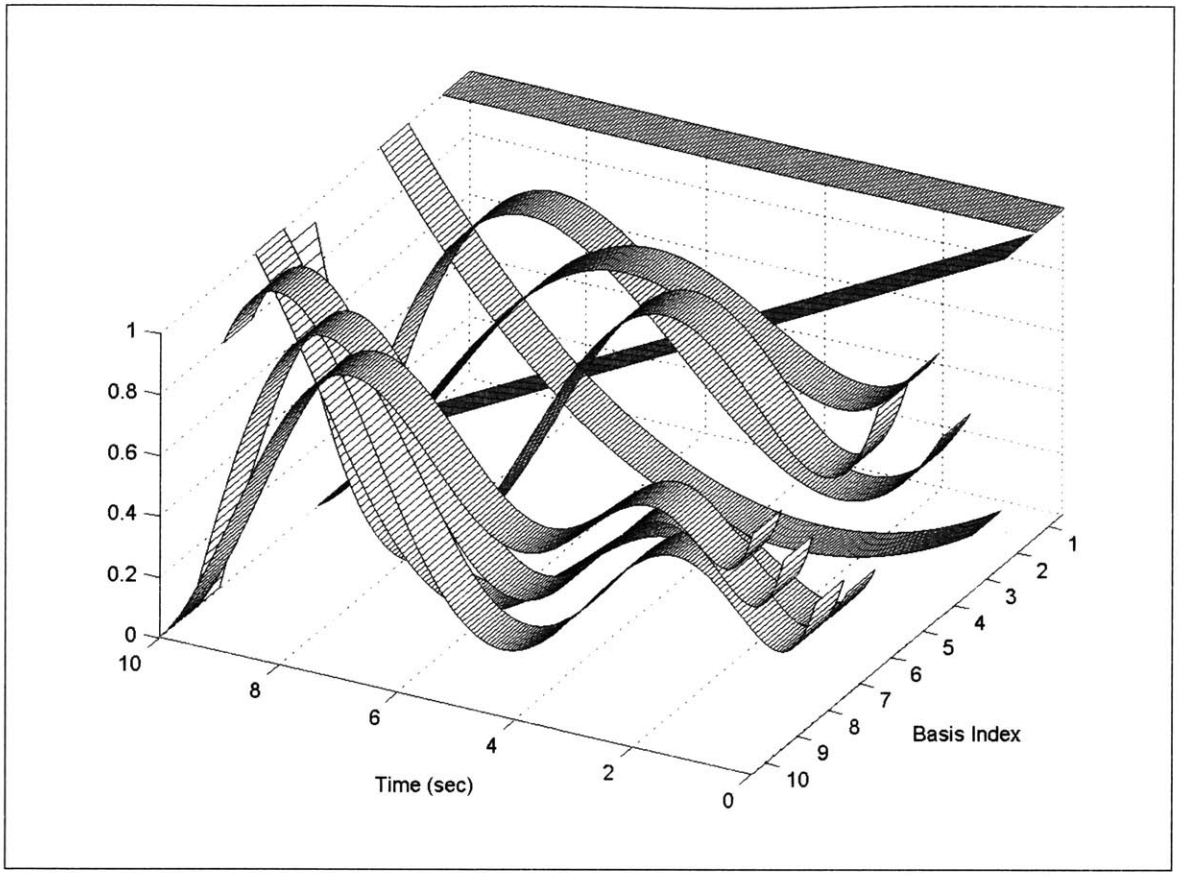

Figure 4-9: Laguerre polynomials plotted along the prediction horizon from 0^{th} order to 9^{th} order.

4.3.2.3 *Basis Function Comparison and Selection*

In this study, the motivation is to take advantage of the series expansions from each of the families of basis functions to approximate the optimal control of our system. The approximating basis function must balance a satisfactory approximation of the optimal cost J^* while adequately decreasing the complexity of the optimization problem. This trade-off is especially apparent when considering the two families of basis functions presented in this research. Polynomial functions may be exploited to further simplify the problem depending on how well the functions themselves can approximate the control input. If the function family happens to be a good fit generally for the optimal control sequence to be generated, it may be able to approximate the control with fewer basis functions than the repeating function families. However, the repeating function cost approximations with n basis functions, $J^*_{Apx_n}$, approaches the optimal cost using the

standard MPC formulation, J^*, as n approaches the number of time-steps in the horizon, N_P, $\lim_{n \to N_P} J^*_{Apx_n} = J^*$. The polynomial functions carry no such guarantee.

To investigate this trade-off for all of the candidate basis function families, the cost function was simplified to penalize only terrain tracking error with a commanded step in altitude. The MPC loop was then run for one iteration to calculate the optimal unconstrained control trajectory and the associated optimal cost, $J^*_{Apx_n}$. This was carried out, incrementing the number of approximating basis functions spanning the prediction horizon from 1 to 50, for each family of basis functions. The resultant optimal costs are plotted in Figure 4-10.

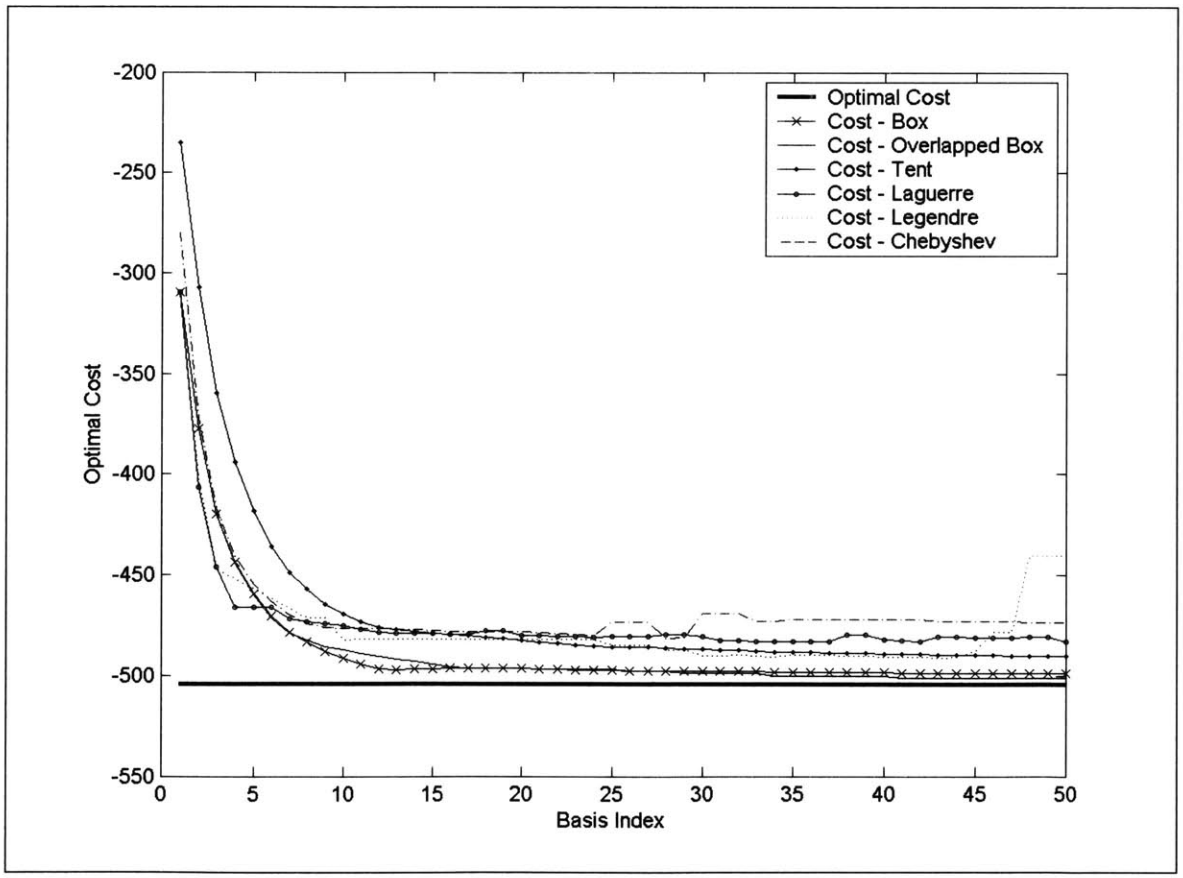

Figure 4-10: Optimal cost approximations varying with the number of applied basis functions plotted for each basis function family along with the true optimal cost yielded by standard MPC formulation.

As you can see in the figure, the basis functions which most nearly approach the true optimal cost J^* are the repeating functions; boxes, overlapped-boxes and the tents, all of

which approach the optimal cost as the number of applied basis functions increases. In contrast, the polynomial functions approach a cost limit as the order is increased that is higher than the repeating functions. However, it is interesting to note that the rate at which the polynomial functions approach the optimal cost limit is initially greater than that of the repeated functions. Specifically, depending upon the computational runtime difference between 5 and 15 basis functions and the tracking performance degradation as the cost declines from optimal, the trade-off between computational complexity and performance could lead to either the Laguerre polynomials or the repeating tents or boxes. Figure 4-11 shows a close up of original plot honing in on the 5 to 15 basis function range. (Note: The Chebyshev and Legendre polynomials were dropped out of consideration after preliminary testing showed they did not lend themselves to this application causing infeasibility during MPC initialization.)

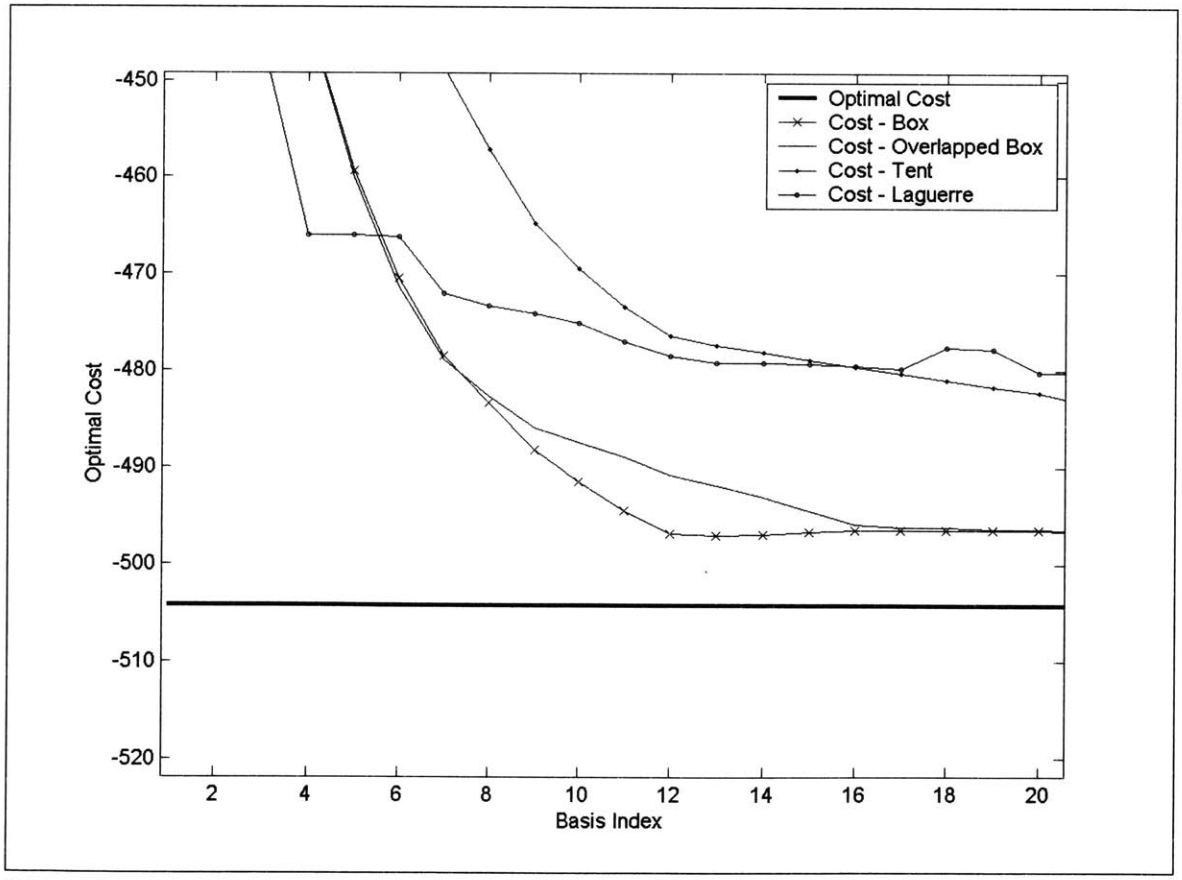

Figure 4-11: Close-up of Optimal Cost Approximation with Increasing Number of Basis Functions.

To weigh the computational benefit of selecting 5 applied basis functions versus 15, the runtimes[‡] must be investigated. The optimizer computational time per iteration information was gathered using the Matlab scripts *tic* and *toc*. These commands start and stop a stop-watch timer, outputting the number of seconds required for the operations conducted between the two. These times are plotted in Figure 4-12 with respect to the number of bases applied.

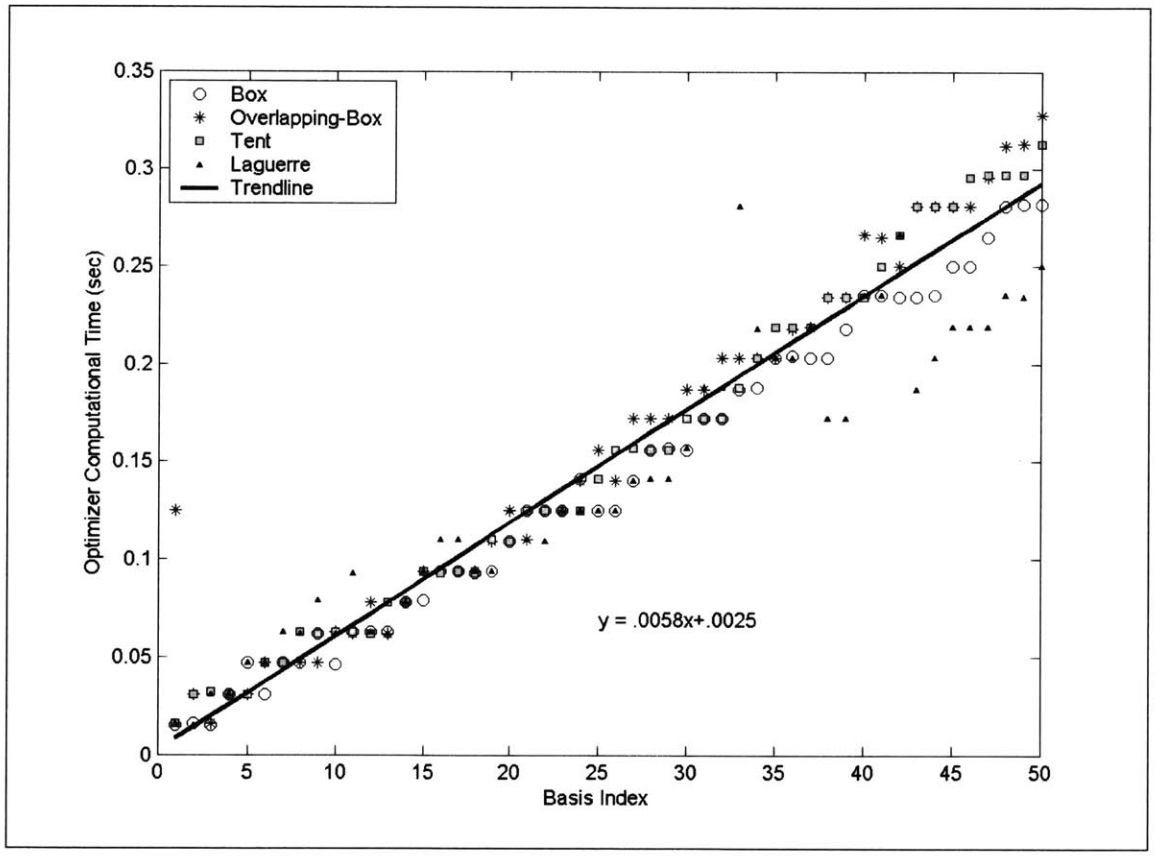

Figure 4-12: Runtime per iteration plotted versus the number of bases applied.

As you can see, the computational time required by the optimizer per iteration increases linearly with number of basis functions applied.

The second half of the trade-off is the performance degradation as the cost less accurately approximates the optimal. The comparison of the optimal trajectory and the basis function approximations after a single MPC iteration is plotted in Figure 4-13.

[‡] Runs for runtime analysis were conducted with SQOPT, a quadratic programming optimizer coded in Fortran, converted to C and .mex-ed for use in a Matlab Simulink simulation environment. The simulation was conducted on a computer with an Intel Pentium 4 processor, a 2.2 GHz processing speed and 1 GB of RAM.

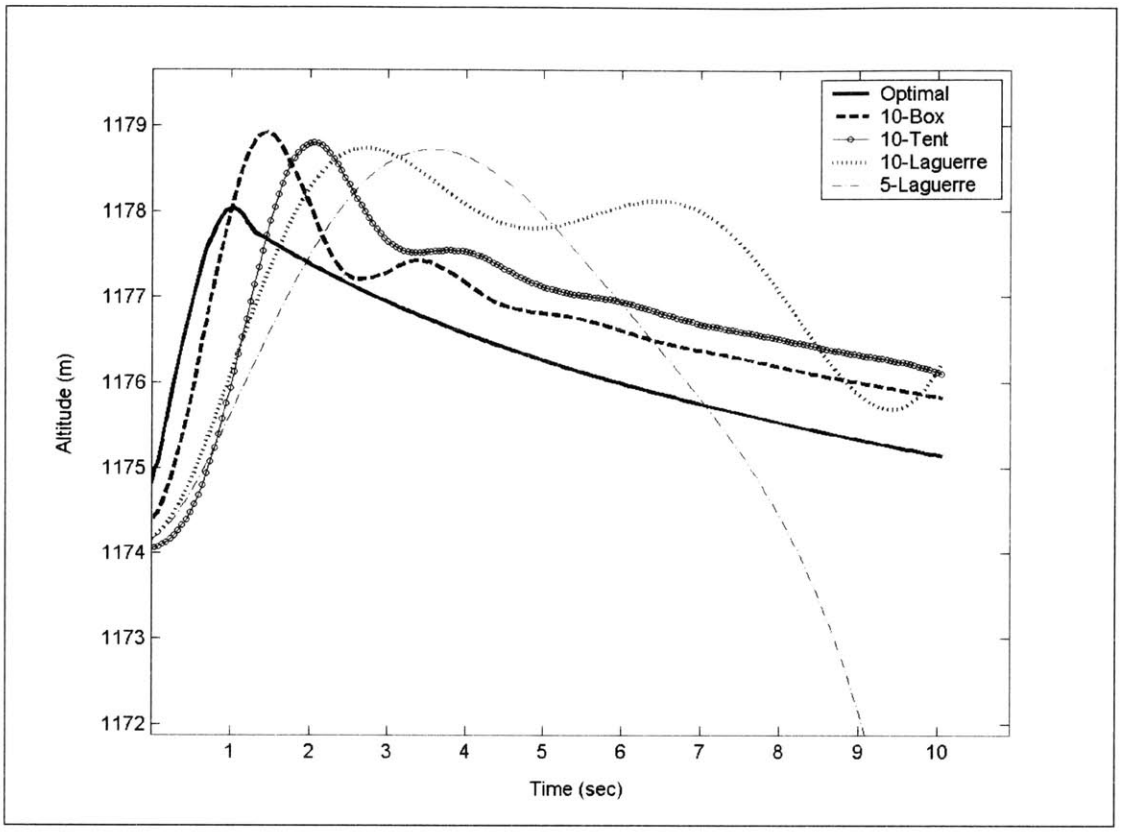

Figure 4-13: Optimal trajectory plotted with the basis function approximations at k=1.

However, due to the repeating nature of MPC, the performance tradeoff can not be measured simply by looking at the optimizer's first guess at the controls over the entire trajectory length. The resulting state trajectory found by applying the optimal controls for the entire prediction horizon does not accurately demonstrate MPC performance in a single iteration because once MPC is initialized; the first 0.5 second trajectory of controls within the horizon has been iterated upon 20 times before it is actually applied. In attempt to capture this iterative performance and increase the applicability of this exercise, the step in altitude reference from Figure 4-13 was inserted into the cost described in (15) with constant X, Y and Ψ references, and followed by MPC for 15 seconds. The resulting responses are plotted in Figure 4-14 for each of the remaining prospective basis function candidate combination; 10 boxes, 10 tents and 5 Laguerre polynomials, each spanning the 10 second prediction horizon.

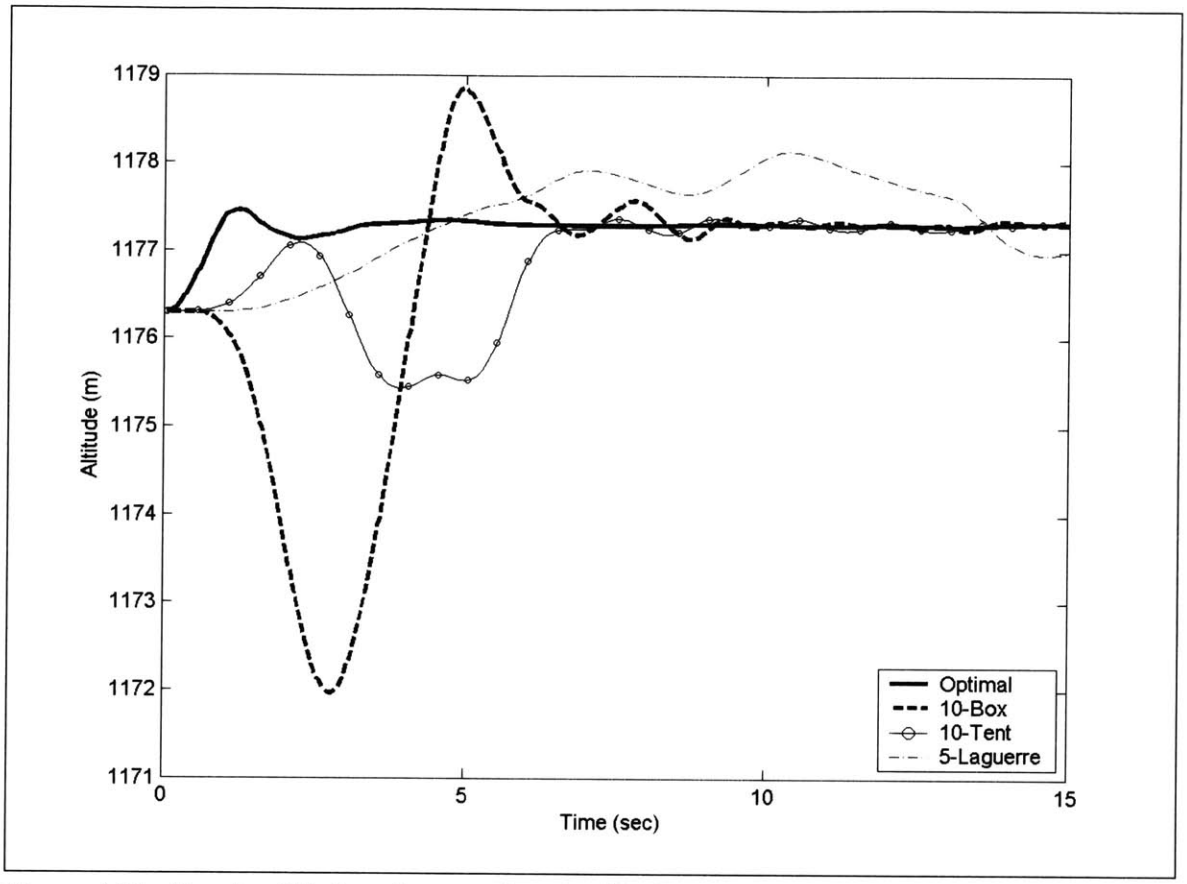

Figure 4-14: Step in altitude reference plotted with the responses of 10 Boxes, 10 Tents and 5 Laguerre polynomials applied in MPC with the 4-state error cost from (15) including constant X, Y and Ψ references, over 15 seconds.

In contrast to Figure 4-13 which shows the Laguerre functions to be a poor approximation of the optimal, it is clear from Figure 4-14 that the Laguerre polynomials satisfactorily approximate the optimal control when applied in the repeating MPC algorithm, while decreasing the computational complexity. Note that 15 seconds of response is plotted exhibiting the initialization of the MPC algorithm within the first 10 seconds, and within the final 0.5 seconds of the response each 0.5 second trajectory of control applied has seen 20 iterations of improvement before their respective application. From this plot we can surmise that the tents and boxes each converge to the optimal solution but with potentially large transient error. The Laguerre polynomials appear to have the characteristic of lower transient errors as the algorithm is initializing but a slightly higher average error upon initialization.

4.3.3 **Example: Mt. Adams with Basis Function Application**

Given the results from the previous section, the next step is to apply these basis functions to our problem of terrain following guidance and control. We start by reviving the cost from (15) and inserting this into (73), the MPC cost formulation summed over the prediction horizon using basis functions to approximate the optimal solution becomes:

(97)

$$J_i = \sum_{k=i}^{i+H-1} \left[\left(S\alpha + \begin{bmatrix} x_{0_k} - x_{REF_k} \\ y_{0_k} - y_{REF_k} \\ z_{0_k} - z_{REF_k} \\ \Psi_{0_k} - \psi_{REF_k} \end{bmatrix} \right)^T Q_k \left(S\alpha + \begin{bmatrix} x_{0_k} - x_{REF_k} \\ y_{0_k} - y_{REF_k} \\ z_{0_k} - z_{REF_k} \\ \Psi_{0_k} - \psi_{REF_k} \end{bmatrix} \right) + \left(\mathrm{B}\alpha + \begin{bmatrix} T_{MR_k} \\ \theta_{R_k} \\ \theta_{P_k} \\ T_{TR_k} \end{bmatrix} \right)^T R_k \left(\mathrm{B}\alpha + \begin{bmatrix} T_{MR_k} \\ \theta_{R_k} \\ \theta_{P_k} \\ T_{TR_k} \end{bmatrix} \right) \right]$$

where:

$$Q_k = \begin{bmatrix} Q_{x_k} & 0 & 0 & 0 \\ 0 & Q_{y_k} & 0 & 0 \\ 0 & 0 & Q_{z_k} & 0 \\ 0 & 0 & 0 & Q_{\psi_k} \end{bmatrix} \quad \textit{and} \quad R_k = \begin{bmatrix} R_{T_{MRk}} & 0 & 0 & 0 \\ 0 & R_{\theta_{Rk}} & 0 & 0 \\ 0 & 0 & R_{\theta_{Rk}} & 0 \\ 0 & 0 & 0 & R_{T_{TRk}} \end{bmatrix}.$$

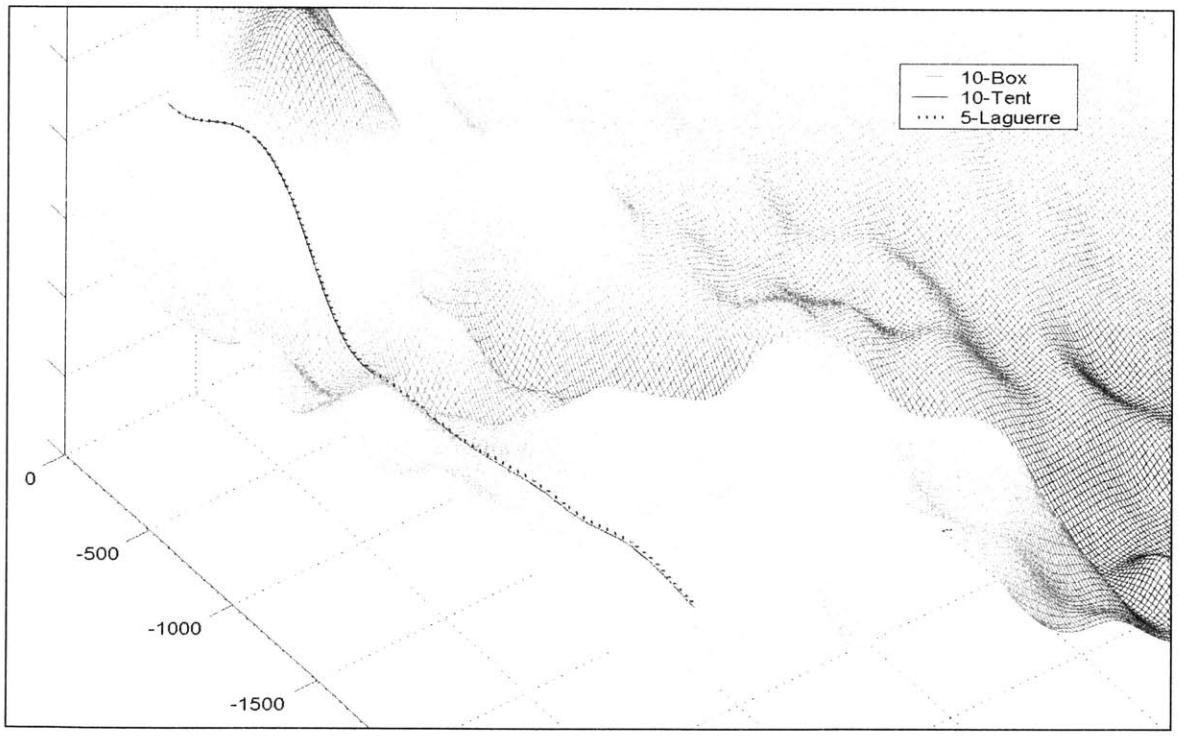

Figure 4-15: Resultant trajectories plotted in 3D on the sample terrain for each of the candidate basis functions: 10 Boxes, 10 Tents and 5 Laguerre polynomials.

Figure 4-15 plots the response for each candidate basis function as applied to the sample section of Mt. Adams terrain seen in the LQR tracking test. The tracking errors corresponding to each candidate basis function are shown in Figure 4-16.

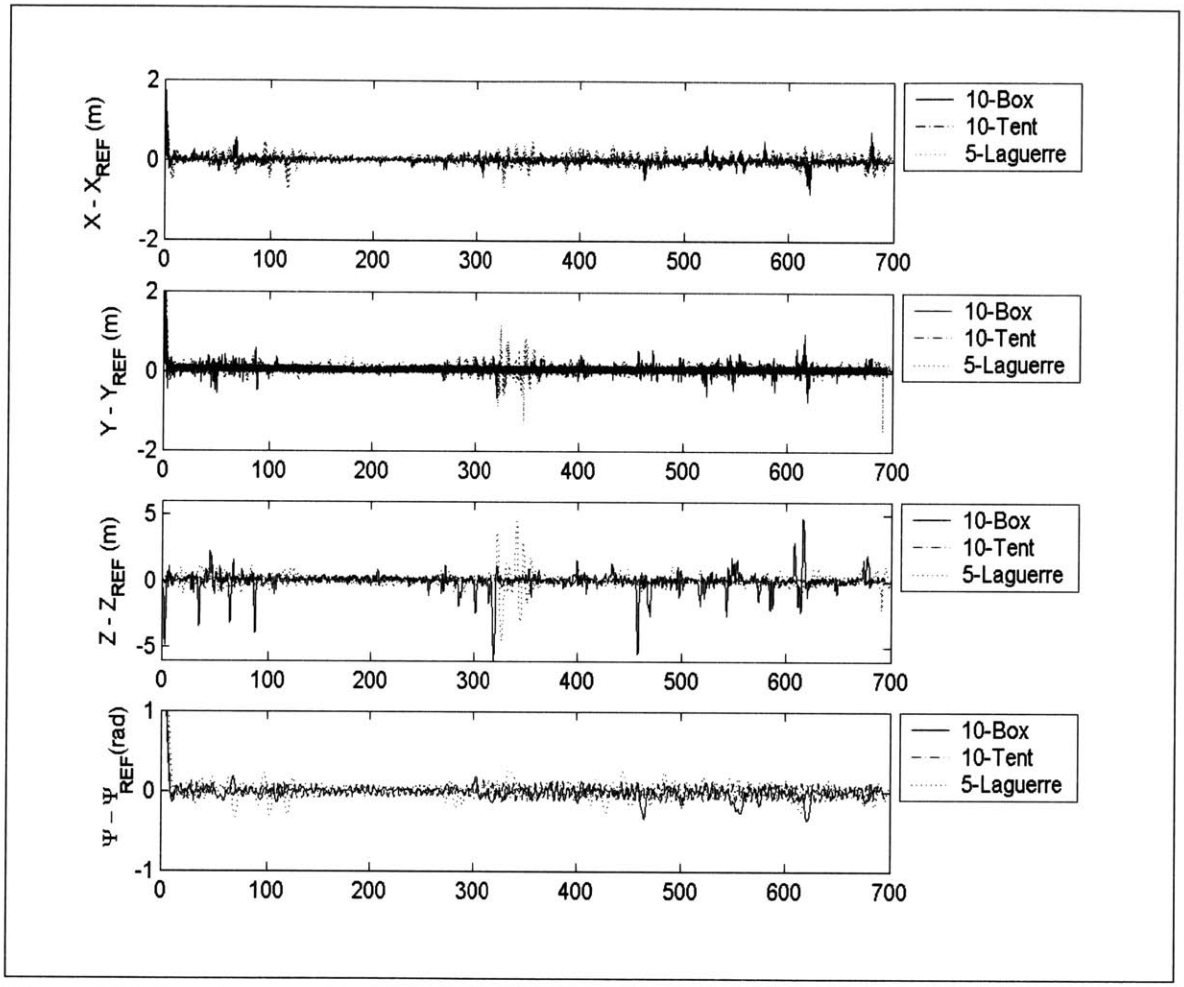

Figure 4-16: Resultant tracking error plotted for each of the candidate basis functions: 10 Boxes, 10 Tents and 5 Laguerre polynomials.

Through investigation of both Figures 4-15 and 4-16, it is clear that all three basis functions approximate the optimal control trajectory comparably, with respect to tracking error magnitudes, in actual guidance and control application with the complex 4-state terrain following cost. The optimizer computational runtimes for each candidate basis function implementation are plotted in Figure 4-17.

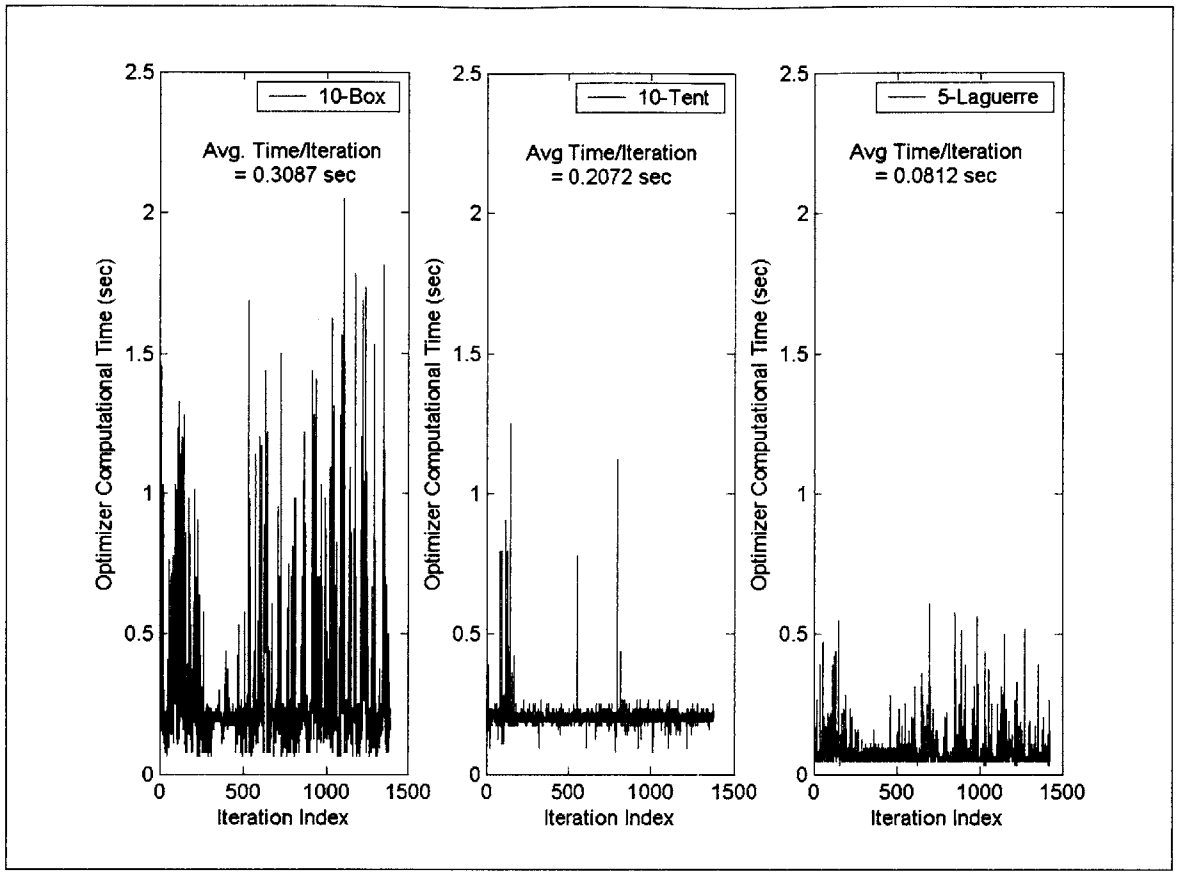

Figure 4-17: Optimizer computational runtimes plotted for each of the candidate basis functions when applied to actual terrain, minimizing the 4-state terrain following cost function.

Figure 4-17 confirms that all three basis function implementations take considerably less time per iteration than the average 3.4 seconds per iteration that regular MPC requires. Since the tracking errors have already been shown to be comparable between the three basis function candidates, the basis function selected for the duration of this thesis can be chosen based on the reduction of computational complexity and runtime. Since the choice of 5 Laguerre basis polynomials requires half as many functions to produce tracking performance equivalent to that of 10 boxes or tents, cutting the required time for optimization in half, they have been selected for the duration of this research.

Having severely cut down the computational complexity of the problem, the tracking errors resulting from basis function implementation are quite comparable to the tracking errors found in regular MPC implementation, especially when the average error values are considered. This is demonstrated in Table 4-2 which compares the tracking error values in X, Y and Z between the standard MPC formulation as in Section 4.2.3 and the MPC implementation with Laguerre polynomial basis functions.

Table 4-2: Maximum, Minimum and Average Error Values for Standard MPC Terrain Following compared with those of MPC with Laguerre polynomial basis function implementation

	Regular MPC			MPC w/Laguerre basis functions		
	Min. (m)	Max. (m)	Avg (abs)	Min. (m)	Max. (m)	Avg (abs)
X-Tracking Error	-0.2892	2.1005	0.0420	-0.4758	2.1298	0.0848
Y-Tracking Error	-0.2471	2.4047	0.0778	-0.5061	2.4201	0.1354
Z-Tracking Error	-0.3426	0.6331	0.0817	-0.7841	1.2244	0.2481
Average Optimizer Computational Time	3.438 sec			0.0812 sec		

Though the maximum error values are higher with the basis function implementation, the average error values are still very small for both the basis function and standard MPC formulations. This proves basis function implementation to be an effective way of reducing computational complexity and runtime without sacrificing performance.

Chapter 5

MPC PARAMETER TUNING AND RESULTS FOR TERRAIN FOLLOWING

This research seeks to take advantage of the obstacle avoidance inherent with the application of hard state constraints and extend it to ground collision avoidance for NOE flight. The following chapter will treat the problem of following the terrain at 6 meters AGL, enforcing ground collision avoidance through the application of a lower bound on altitude which is set at a safe distance above the ground. Parameter sensitivities for cost function penalty and prediction horizon length will be investigated with respect to stability and tracking performance. Once all parameters have been set, the algorithm sensitivities to terrain severity at varying nominal velocities will also be explored through application to several different terrain cross sections.

5.1 State Constraint Generation and Implementation

The altitude constraint to be applied at each time-step along the reference trajectory is found in a similar manner to the setting of the reference trajectory. For each (x,y) coordinate along the final sampled reference trajectory, the constraint is set at the actual terrain elevation at that point. This establishes a state constraint defined in the earth frame for each time step, i, along the prediction horizon:

(98) $$z_{i,\min} \leq z(k+i \mid k)$$

Since the optimizer is being charged to find scale factors for each control basis function, it is necessary to convert the output spatial constraints into constraints on the control scale factors, α. To accomplish this, the equalities (71) and (72), derived in Section 4.3.2.1, are used to form inequalities for the state and control constraints in terms of α:

(99) $$y_{\lim} \leq S\alpha + y_0$$

(100) $$u_{\lim} \leq \mathrm{B}\alpha + u_0$$

These can be rearranged to yield the traditional linear constraint inequality:

(101) $$\begin{bmatrix} S \\ B \end{bmatrix} \alpha \leq \begin{bmatrix} y_{\lim} - y_0 \\ u_{\lim} - u_0 \end{bmatrix}$$

Finally with all constraints in terms of α, the cost function from (73) can now be solved and the terrain following portion of the algorithm is complete.

5.2 VARIATION OF MPC PARAMETERS

Now that the terrain following problem has been transposed as a convex optimization problem via perturbational linearization of the model and its dimension reduced by introducing low order basis functions, it can be posed and solved in real-time. With the application and algorithm established, MPC parameter sensitivities can be investigated and the appropriate parameters selected for this application. There currently exists no formal procedures for MPC parameter selection, thus successfully applying MPC can be somewhat of an art form. For the terrain following application, parameter selection is an especially challenging task because there are multiple states and control inputs for which weighting matrices must be populated. With this in mind a balance must be found between keeping the problem size to that which can be computed in real-time and how much look ahead is required to allow the controller adequate anticipation of changes in the terrain.

5.2.1 Variation of Prediction Horizon Length

MPC literature has indicated that choosing an insufficient look-ahead leads to instability [21, 31], therefore the selection of the prediction horizon length, H_P, is crucial

to the successful implementation of MPC. The anticipatory qualities of MPC are directly related to the prediction horizon length. Therefore, if not chosen wisely, a stable prediction horizon may still result in poor tracking performance as MPC is not obtaining adequate knowledge of the system. Depending on the non-linearities present in the plant model and the severity of the maneuvers required, as the prediction horizon length increases past a given threshold the performance begins to degrade as the vehicle creeps beyond the flight regime about which the prediction model was linearized. The computational burden involved with an excessively long prediction horizon may also contribute to poor performance. If after searching for an excessively long trajectory of controls, the maximum allowable iterations may be reached, forcing the optimizer to abort the optimization without returning a feasible solution. The prediction horizon should be selected as one of the shortest horizon lengths which yield good tracking performance while also maintaining system stability.

To establish the appropriate length of prediction horizon for this thesis, the constrained MPC terrain following algorithm was tested against the worst case terrain for prediction horizon lengths varying from 3 seconds to 18 seconds.

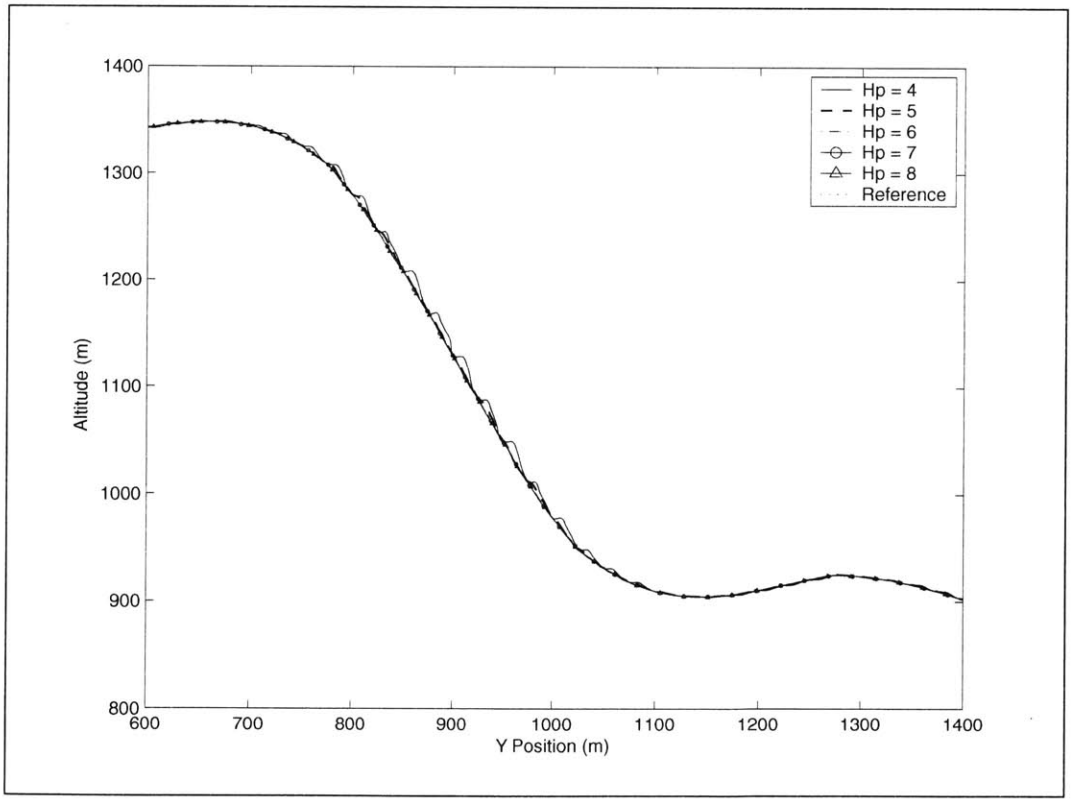

Figure 5-1: Tracking performance plotted for prediction horizons varying from 4 to 8 seconds.

From this test, it was determined that for this application, a 3 second prediction horizon is not sufficient for stability. A 4 second horizon was found to be marginally stable. Increasing the horizon length from 4 seconds to 8 seconds, the tracking performance improved with each additional second and then leveled off at approximately 6 second prediction horizons. This is more noticeable in Figure 5-2, a close-up of Figure 5-1.

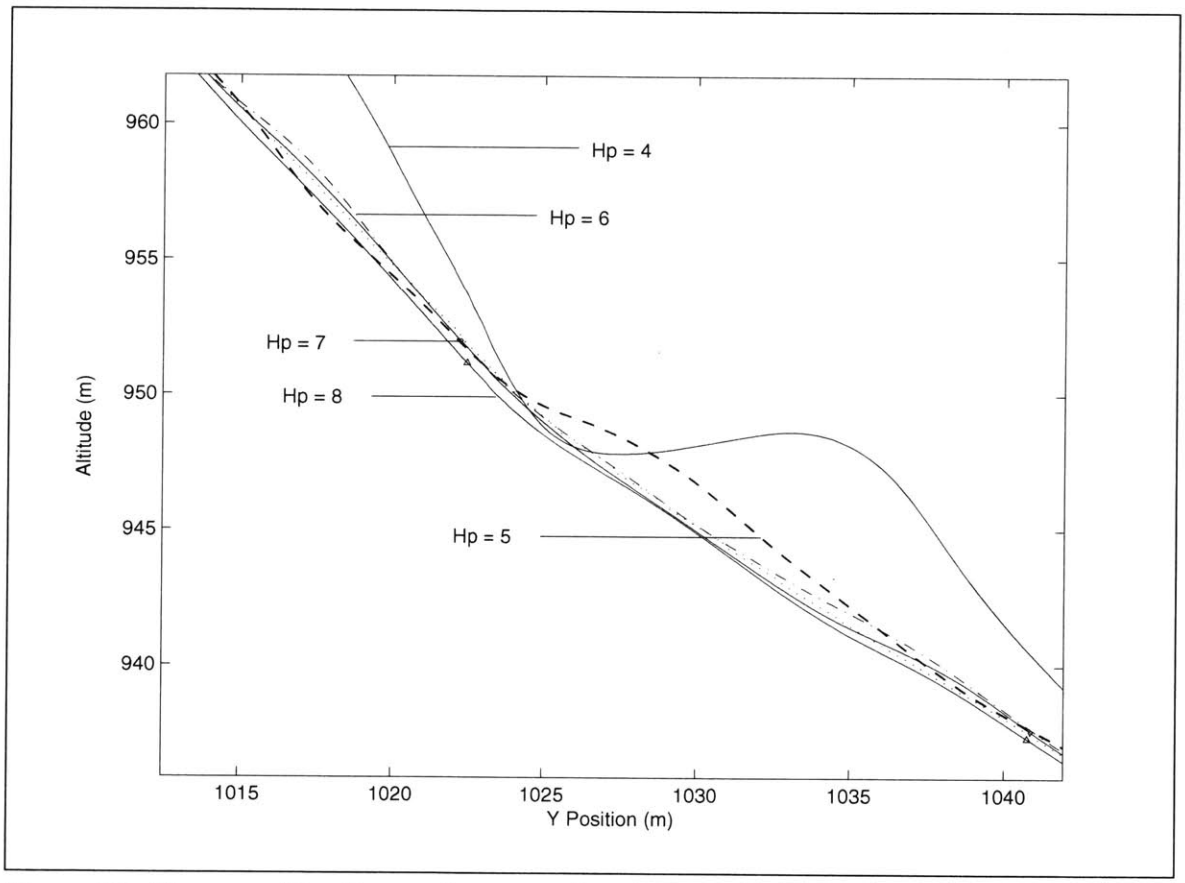

Figure 5-2: Close up of tracking performance plotted for prediction horizons varying from 4 to 8 seconds.

Figure 5-3 shows that a prediction horizon of 6 seconds sufficiently accommodates our tracking problem. Prediction horizon lengths around 8 seconds are sufficient to prevent instability and yield good tracking performance. We selected a conservative prediction horizon length of 10 seconds for the duration of this research in attempt to accommodate a wider range of terrain beyond that which was tested here. A 10 second horizon was selected to enable obstacle avoidance as well, because the 10 second look-ahead accommodates the obstacle sensor range at typical NOE flight speeds.

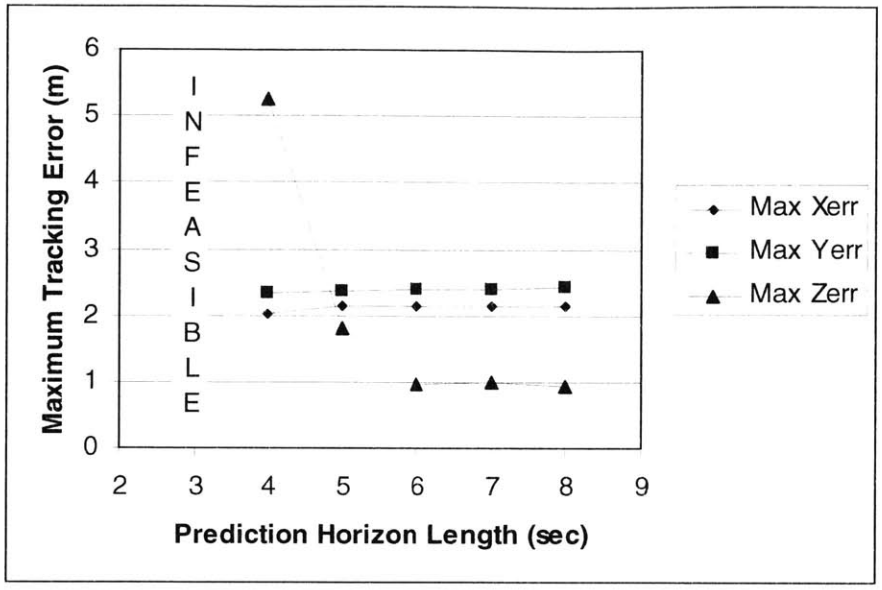

Figure 5-3: Maximum tracking errors in each of the earth frame axes plotted for varying prediction horizon length.

5.2.2 Variation of Terrain Following/Terrain Avoidance (TF/TA) Ratio

The values of Q_X, Q_Y, Q_Z and Q_Ψ have been predetermined as described in Section 3.4. To implement the TF/TA ratio, τ, we assigned $Q_X = Q_Y = a$ and $Q_Z = a\tau$. Similarly, normalizing between meters and radians and the relative importance of Ψ-tracking to the terrain following problem determined the Ψ-weight, $Q_\Psi = 10\tau$. Setting a high value for τ will allow little deviation from the set altitude above ground while allowing the xy-track to meander. A smaller value will emphasize lateral tracking while not attending as highly to maintaining the precise distance above ground (a safety margin will be enforced by constraints in either case).

Before the analysis was conducted comparing different combinations of the control matrices, R_k, performance tradeoffs were conducted to determine the appropriate TF/TA ratio. In testing various TF/TA ratios to determine Q_k, the R_k weights for all controls were defined as $R_k = 0.1$. These control penalty combinations will be tested and compared once the appropriate TF/TA has been established. The TF/TA ratios tested are listed in Table 5-1, followed by their resultant tracking error performance plotted in Figure 5-4.

Table 5-1: Weight variations conducted for determination of Q_k values

TF/TA, τ	$Q_{X/Y}$	Q_{Ψ}	$R_{TMR/TTR}$	R_{CYC}
1	1	10	0.1	0.1
10	1	10	0.1	0.1
100	1	10	0.1	0.1

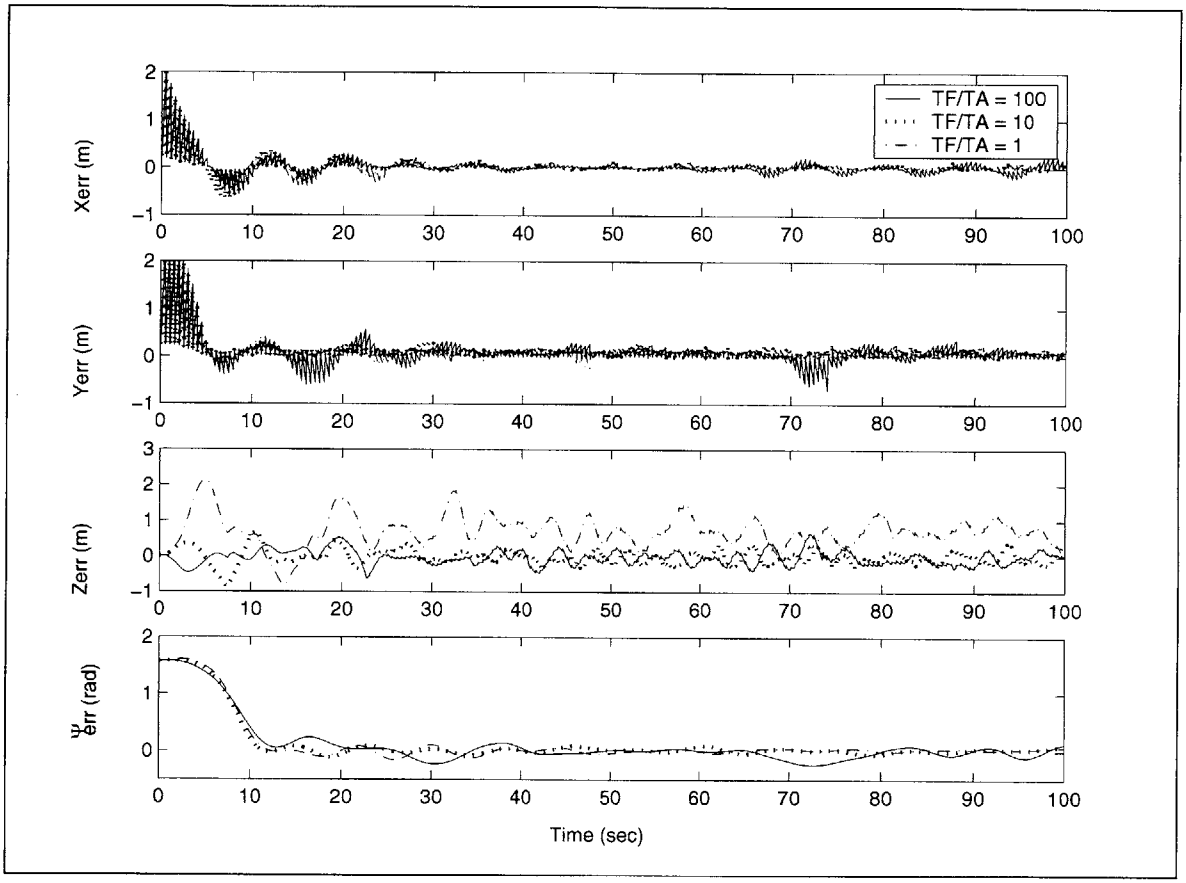

Figure 5-4: Tracking error performance plotted for the four references (X, Y, Z and Ψ) towards the determination of the TF/TA ratio.

It is clear from Figure 5-4 that the X and Y error values are acceptable for all TF/TA ratios, therefore we are free to make the TF/TA selection based on the desired altitude and heading (Ψ) tracking. At a glance, it is obvious that the TF/TA ratio necessary to obtain the altitude tracking accuracy imperative to terrain following flight must be greater than 1, as this ratio leads to maximum altitude errors of 2 meters even at the nominal speed of 10 knots (nominal speeds will be varied up to 30 knots later in this section). The maximum altitude tracking error of TF/TA = 1 is halved by increasing the ratio to 10 or 100, as shown in Figure 5-4. The average Z-error highlighted in Table 5-2, is reduced even further when the TF/TA ratio is increased, eliminating TF/TA = 1 from consideration.

Table 5-2: Average Error Values for TF/TA = 1, TF/TA = 10 and TF/TA = 100

	TF/TA = 1	TF/TA = 10	TF/TA = 100
	Avg (abs)	Avg (abs)	Avg (abs)
X-Tracking Error (m)	0.0391	0.0790	0.0857
Y-Tracking Error (m)	0.1031	0.1494	0.1644
Z-Tracking Error (m)	0.7104	0.1611	0.1584
Ψ-Tracking Error (m)	0.1358	0.1558	0.1962

Though there is a significant difference in altitude tracking performance between TF/TA = 1 and TF/TA = 10 or 100, there is no apparent tracking benefit associated with applying TF/TA = 100 as opposed to TF/TA = 10. Therefore, since relatively equivalent performance is attainable with both TF/TA ratios, we select the TF/TA ratio of 10 for the duration of this research to keep the magnitude of the cost smaller.

5.2.3 **Variation of Control Weighting Matrices**

The investigation of the control weighting matrices was conducted by both linear closed loop eigenvalue analysis of the perturbational component of key system states and tracking performance analysis. The linear closed loop eigenvalue analysis was conducted on the two key system states, altitude (δz) and yaw rate (δr), with respect to varying R_k. In applying perturbational linearization, a linear input-output mapping is assumed between the perturbational control and the perturbational state as follows:

(102) $$\delta u^* = K^* \delta x^*$$

Using (102), K^* can be backed out based on (71) and (72) and corresponding closed loop eigenvalues can be obtained for stability analysis.

These closed loop eigenvalues were found for these inputs by using the Matlab script *linmod* to obtain the linear state-space matrices *A*, *B*, *C* and *D*, linearizing the system about the input nominal states. From the state space matrices, the input/output mapping K^*, was applied to the system as illustrated in Figure 5-5, and eigenvalues of the [δz, δr] subspace to the controls, [δT_{MR}, δT_{TR}].

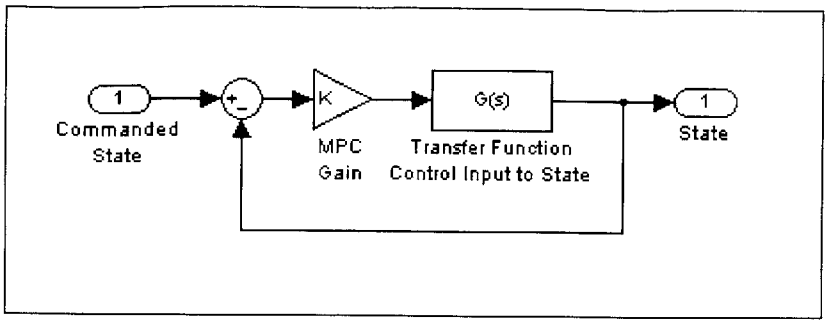

Figure 5-5: Block diagram showing loop closure with MPC gain

To obtain the MPC input/output mapping gains, K^*, with which to close the feedback loop, firstly MPC was run for 15 seconds tracking a constant reference to initialize the system. Then, to obtain K^* from (102), the equalities derived in (71) and (72) to equate $\boldsymbol{u}$ and $\boldsymbol{x}$ in terms of $\boldsymbol{\alpha}$, were inserted as follows:

(103) $$B\alpha^* = K^* S\alpha^*.$$

Then, using the pseudo-inverse of $S\alpha$, (103) can be solved for K^*:

(104) $$K^* = \left(B\alpha^*\right)\left(S\alpha^*\right)^{-1}.$$

To test out the sensitivities of the transfer functions $\delta T_{MR}/\delta z$ and $\delta T_{TR}/\delta r$ to various combinations of control weighting matrices R, the above procedure was conducted and K^* calculated for the following weighting combinations listed in Table 5-3.

Table 5-3: Weight variation conducted for stability analysis of closed loop with MPC gain

Test Run	Q_Z	$Q_{X/Y}$	Q_Ψ	$R_{TMR/TTR}$	R_{CYC}
1	10	1	10	0.1	1
2	10	1	10	0.1	0.01
3	10	1	10	0.1	0.001
4	10	1	10	1	0.1
5	10	1	10	0.01	0.1
6	10	1	10	0.001	0.1

The resulting closed loop poles are plotted in Figure 5-6 for the transfer function $\delta T_{MR}/\delta z$ and in Figure 5-7 for the transfer function $\delta T_{TR}/\delta r$. For the transfer function $\delta T_{MR}/\delta z$, the controller moves the open loop pole at the origin to the left to obtain varying degrees of faster response depending on the control weighting matrix combination. Note that pole movement at the origin due to varying control weights is very small (on the order

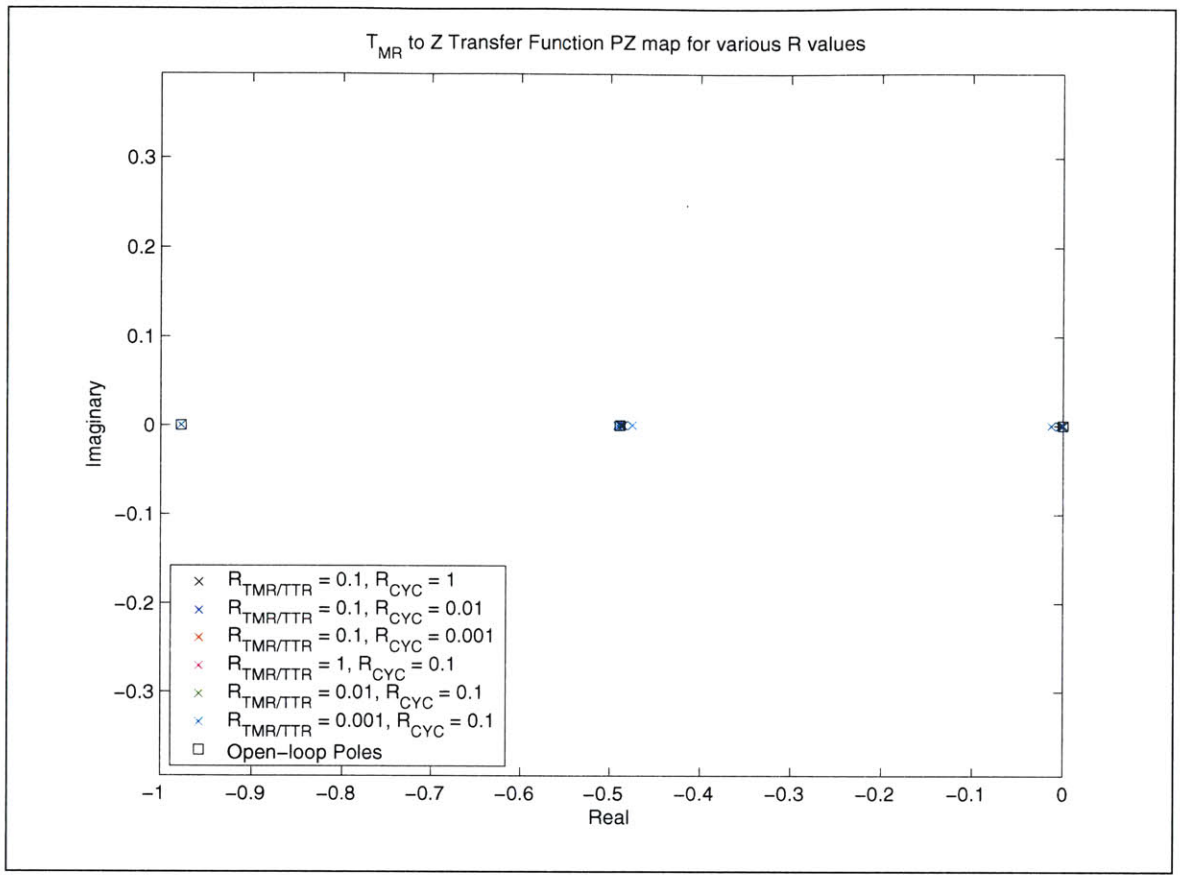

Figure 5-6: Closed loop eigenvalues plotted for the transfer function $\delta T_{MR}/\delta z$ with K^* calculated for various control weight combinations.

of 10^{-3}). Given that these characteristics are consistent for all control weighting matrix combinations, it can be surmised that there is very little system sensitivity to the control weighting matrices, R, with respect to the primary longitudinal control input to state transfer function, $\delta T_{MR}/\delta z$.

The primary lateral control input to state transfer function investigated, $\delta T_{TR}/\delta r$, yielded more interesting sensitivity results to variations in the control weighting matrix. These eigenvalues are plotted in Figure 5-7 on the following page. The movement of the pole at the origin is the most interesting feature to note on this plot as some control matrix combinations move the pole into the right half plane. This may indicate instability of some control weighting matrix combinations and will be kept in mind as tracking performance tests are evaluated.

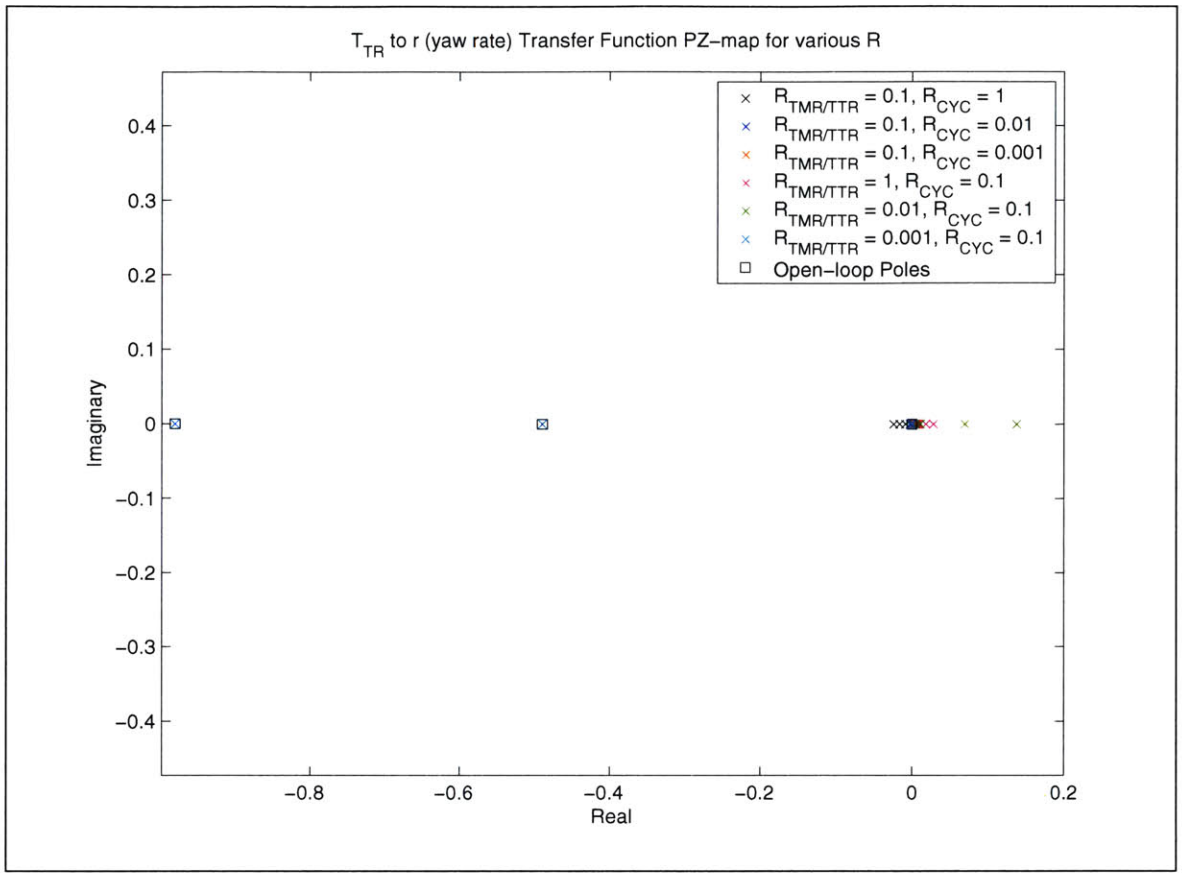

Figure 5-7: Closed loop eigenvalues plotted for the transfer function $\delta T_{TR}/\delta r$ with K^* calculated for various control weight combinations.

While the eigenvalue sensitivity analysis yields a glimpse into the effects of the MPC controller, the relatively small variations in pole placement with variation in control weighting matrices, R_k, make it difficult to specify the best values. Because of this, tracking performance with varying R_k is also included in this investigation to give additional insight. The initial conditions of this test were set to excite both longitudinal and lateral dynamics. A 1 meter step in altitude was required of the vehicle to provide a picture of the longitudinal response speed and damping for each control weight combination. An initial 90° turn was also initiated to obtain lateral damping performance with each of the control weightings. The lateral damping is especially important to investigate per the eigenvalue analysis to assure that movement of the pole at the origin slightly into the right half plane for some control weight combinations is transient and does not affect the steady state stability of the system. The tracking performance for each of the weighting matrix combinations are plotted for each reference axis (X, Y, Z and Ψ) in Figure 5-8.

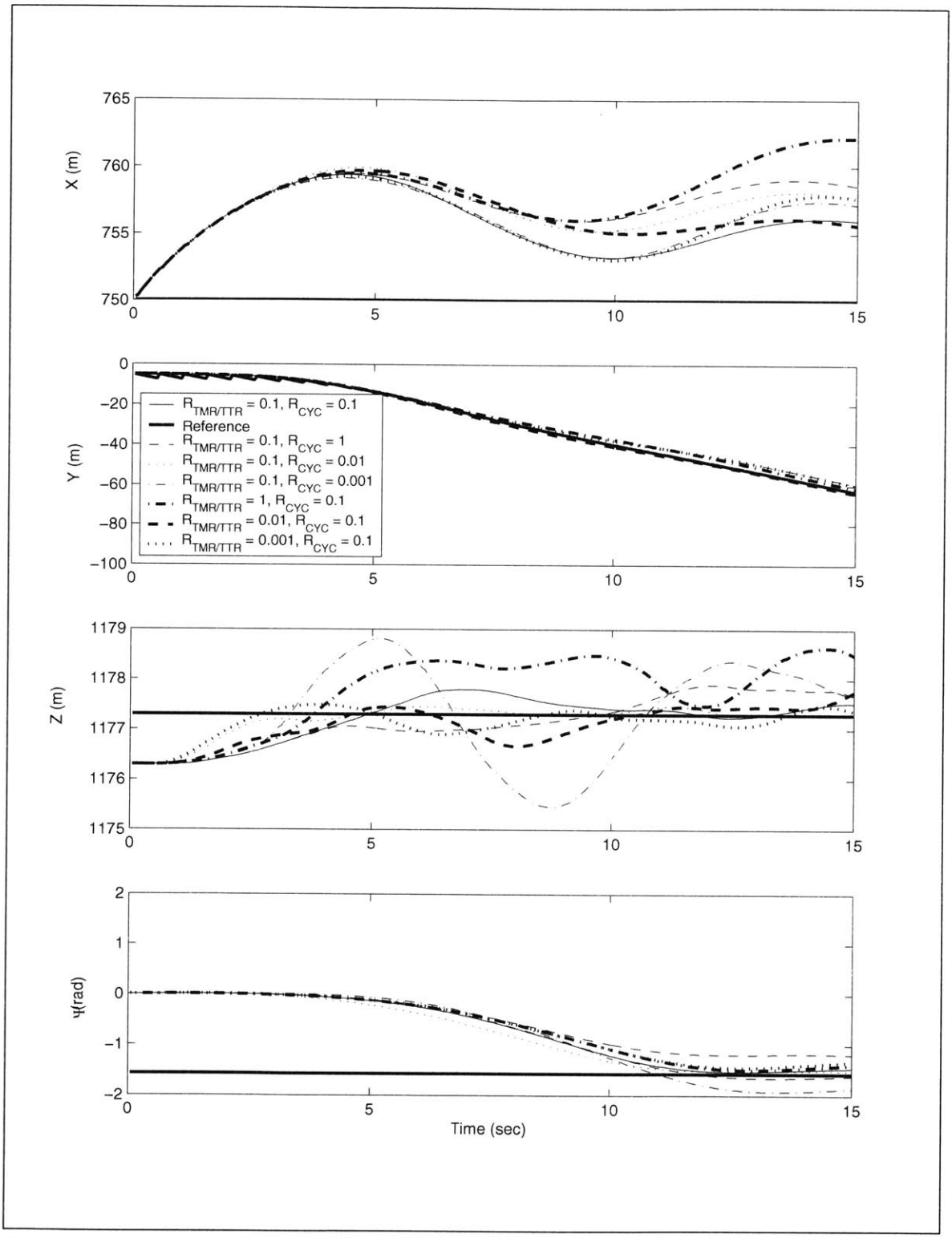

Figure 5-8: Tracking performance plotted for various control weight combinations for each tracked state, X, Y, Z and Ψ.

As we suspected per the eigenvalue analysis which shows the pole at the origin being moved into the right half plane by the MPC gain associated with the control weight combination of $R_{TMR/TTR} = 1$, $R_{CYC} = 0.1$, the oscillation in the lateral response appears to be growing, indicating instability and ruling out that weight combination. Longitudinally, also consistent with the pole-zero maps, there is no such instability, however, a large amplitude oscillation is produced by the combination $R_{TMR/TTR} = 0.1$, $R_{CYC} = 0.001$. Since this particular weighting matrix combination also includes the slowest longitudinal damping response, it is removed from consideration as well.

Overall, the performance data indicates that the test combinations $R_{TMR/TTR} = 0.1$, $R_{CYC} = 0.1$ and $R_{TMR/TTR} = 0.1$, $R_{CYC} = 0.01$ yield the best performance. Both of these combinations exhibit fast damping of the X oscillation caused by the initial 90° turn to traverse the Y-axis. The two weighting combinations also exhibit the best altitude tracking with fast response and very little overshoot or oscillation. Finally, unlike many of the gain combinations, they zero-in on the Ψ reference without overshoot. This is especially the case for the second of the two specified combinations which had the fastest response of all of the gain combinations. Therefore for the duration of tests, the optimal weightings for the cost function describing terrain following guidance and control Q_k and R_k will be set as follows:

$$Q_{Xi} = Q_{Yi} = 1,\ Q_{Zi} = 10,\ Q_{\Psi i} = 10,\ R_{\theta Ri} = R_{\theta Pi} = 0.01 \text{ and } R_{TMRi} = R_{TTRi} = 0.1.$$

5.3 MPC Results: Variation of Terrain Severity and Nominal Velocity

Given the established MPC parameters, sensitivity investigation can be conducted on the nominal velocity and terrain steepness. To obtain comparison data for this part of the study, varying terrain cross-sections were tested at nominal velocities ranging from 10 to 30 knots. These four terrain sections, shown in Figure 5-9, were selected to provide a variety in levels of tracking difficulty. The relative difficulty of each section was determined based on the changes in flight path angle required for tracking the terrain. All simulation runs were conducted with the initial waypoint set to the appropriate X-coordinate (selecting which cross section is being traversed) and Y = 0 (the top edge of the map). Aiming to traverse the Y-axis at a constant X-position to obtain each desired cross-section, the final waypoint was set to the lower edge of the map at the initial X-coordinate and Y = -3600.

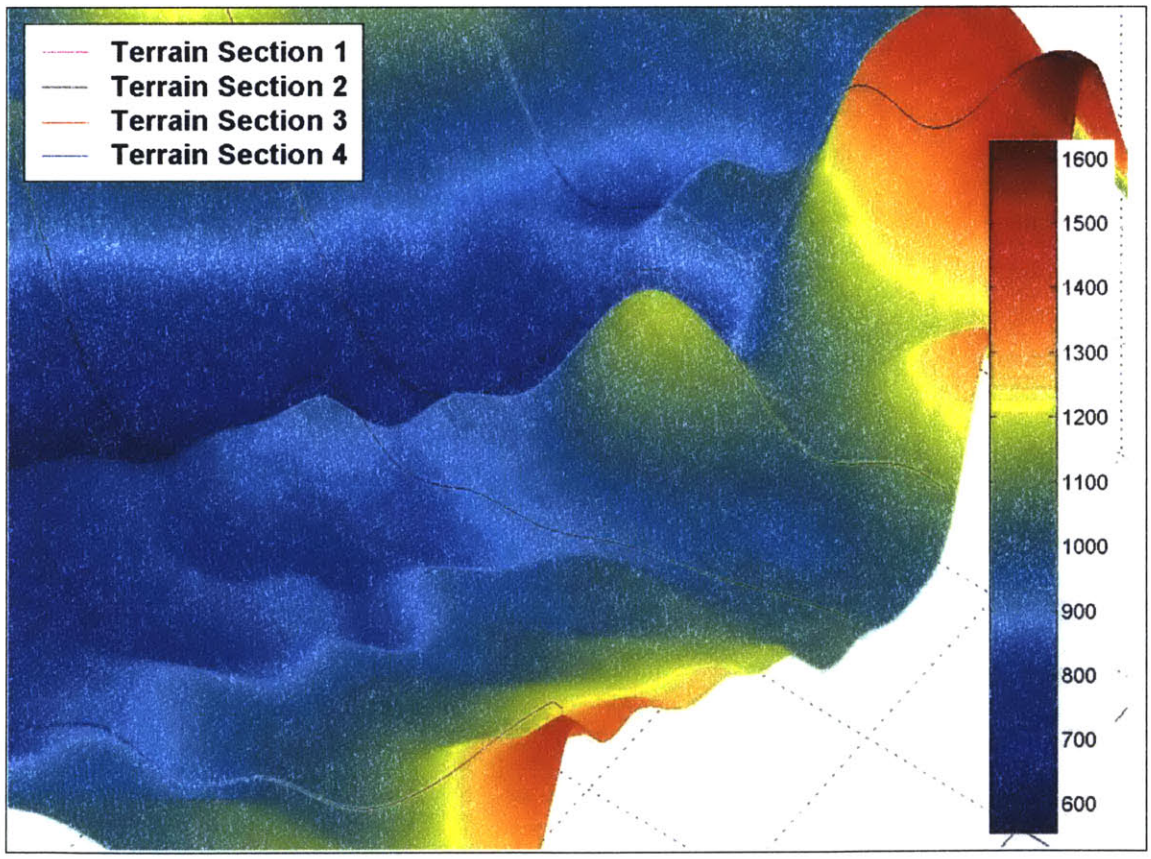

Figure 5-9: 3-Dimensional plot of the sample terrain area with the four terrain section reference trajectories plotted and labeled 1 to 4 from left to right.

5.3.1 **Terrain Section 1**

The first terrain section selected for investigation is marked by relatively low maximum flight path angle magnitudes (maximum magnitude of 40 degrees declining right at the start and gradually decreasing for the duration of the traverse).

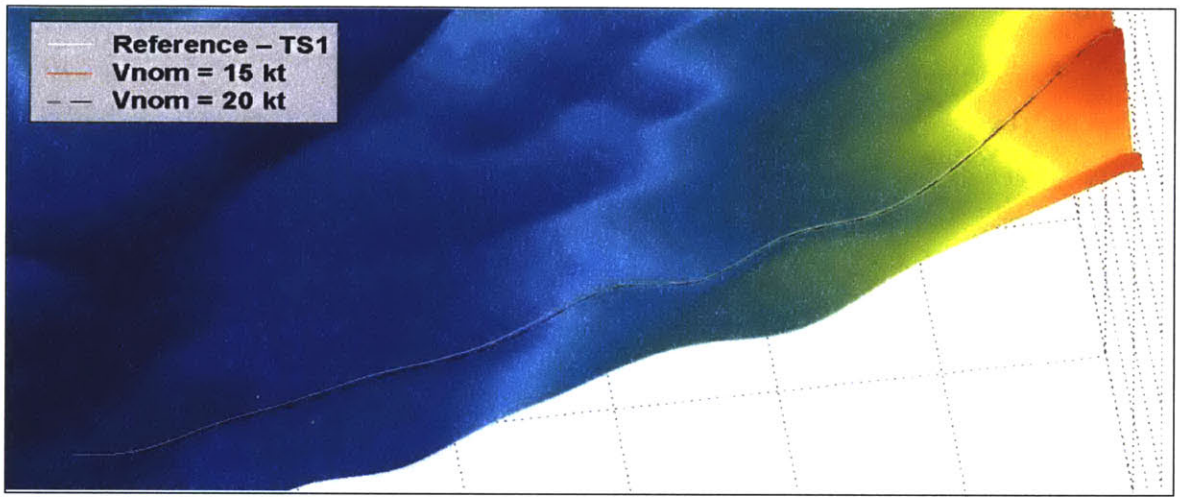

Figure 5-10: Terrain Section 1 results for varying nominal velocities plotted with the initial reference line connecting the initial and final waypoints.

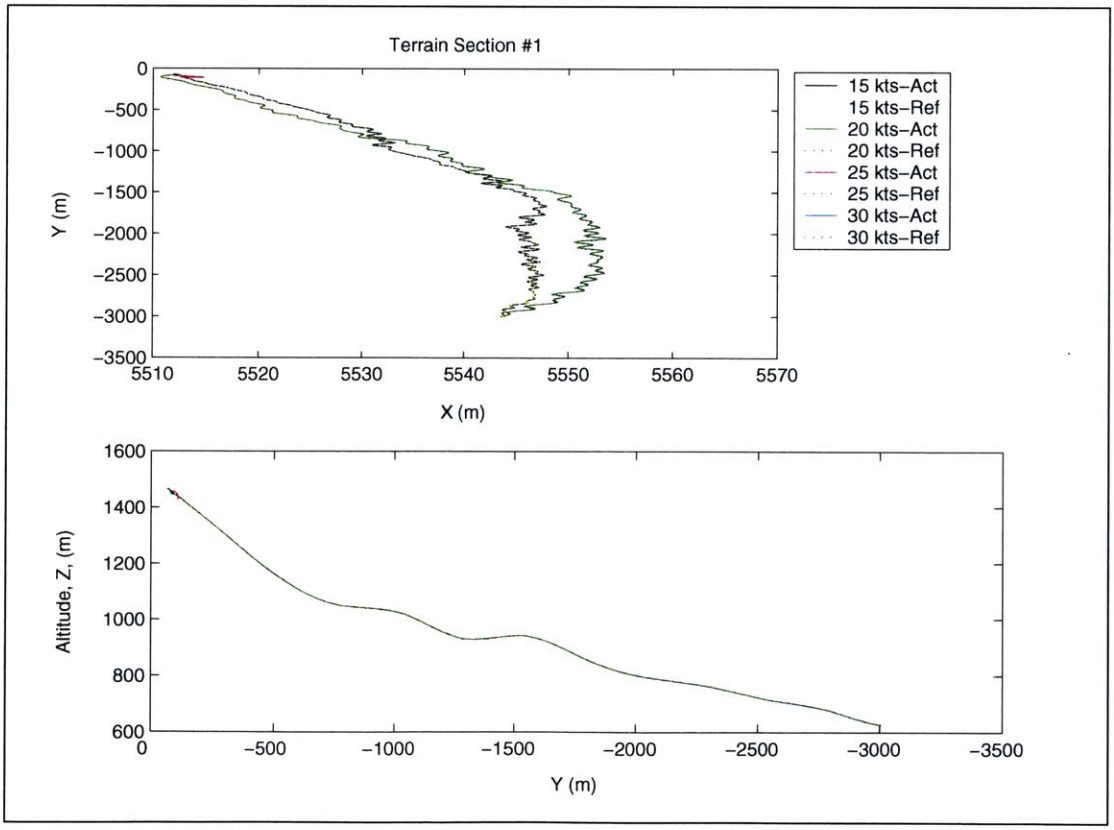

Figure 5-11: Terrain Section 1 plotted with simulation data at nominal velocities varying from 15 to 30 knots. Upper figure is a bird's eye view of the performance. Lower figure is a cross-sectional view.

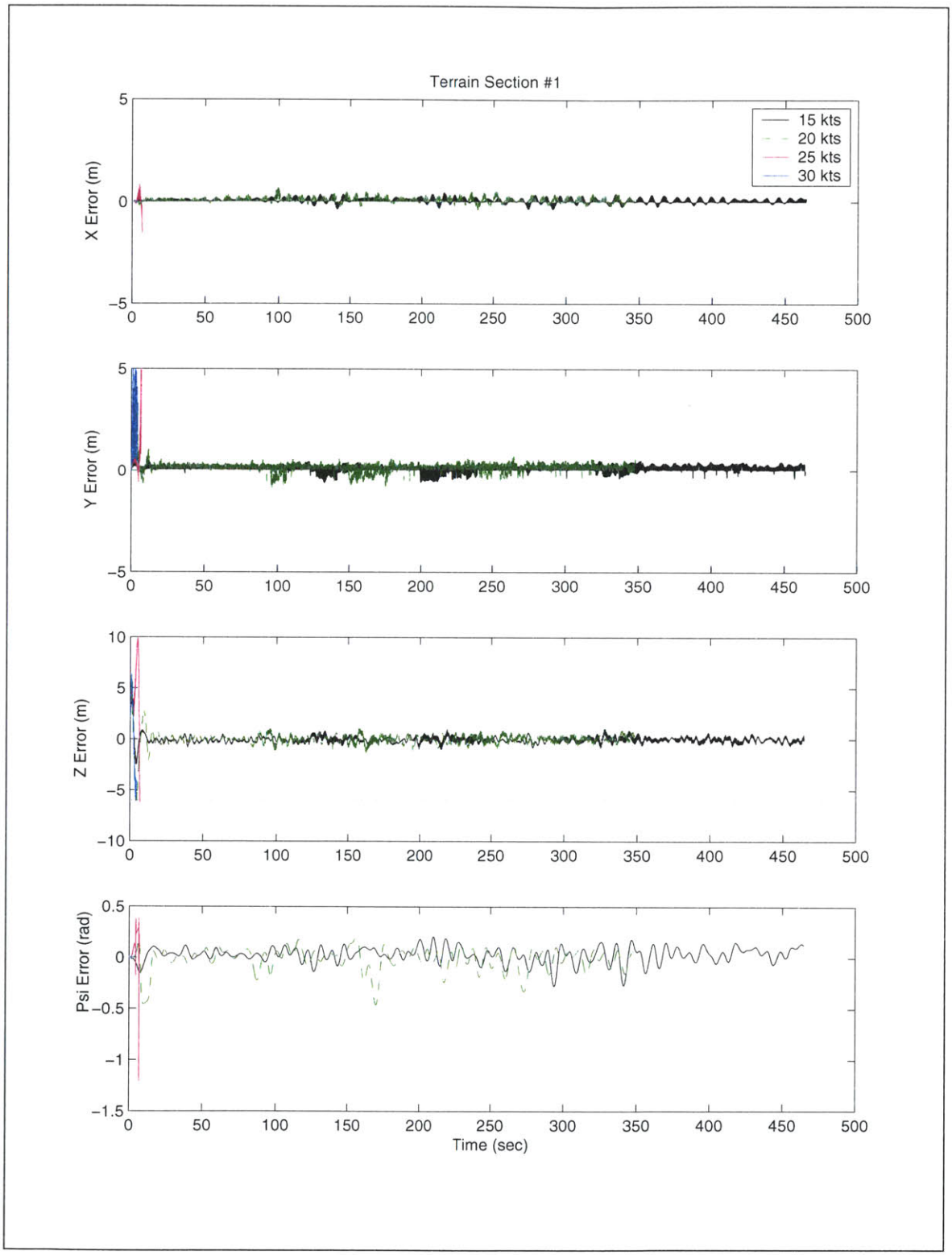

Figure 5-12: Terrain Section 1 reference tracking errors in X, Y, Z and Ψ for nominal velocities varying from 15 to 30 knots.

5.3.1.1 *Lateral Tracking Variation – Terrain Section 1*

The performance data in Figure 5-11 shows a drift in the lateral tracking from the initial reference of a straight line down the Y-Axis at X = 5512, to where the run was stopped at X=5545. Though the lateral reference is redrawn each time as the line connecting the final waypoint and the current position, the lateral error contribution in the cost, per its assigned weight, is not enough to drive a significant change in lateral direction, leading to the lateral offset as the final waypoint placed at (5512, -3600) is approached. It is clear that at about halfway through the run (Y= -1500) the controller begins to direct the helicopter back towards the final waypoint. A 3D plot of the resultant trajectory for each feasible nominal velocity is shown in Figure 5-10, which illustrates the small magnitude of the drift with respect to the total distance traveled.

The Ψ-reference was tracked with an average error of less than +/- 0.1 radians (~5 degrees). The maximum error magnitude was 0.5 radians (~30 degrees) occurring while the controller was initializing. Given the greater emphasis on longitudinal tracking in this research, the lateral offset and maximum heading error are well within acceptable limits.

5.3.1.2 *Longitudinal Tracking Variation – Terrain Section 1*

The altitude tracking error plotted in Figure 5-12 shows clearly that longitudinal (Z) reference tracking is very accurate even at the nominal velocity of 20 knots, with an average tracking error of approximately 1 meter with a maximum error of less than 3 meters even at 20 knots. Both are well within the applied limit, indicated in Figure 5-12 on the Z-tracking error plot by the dotted line. For this terrain section, the higher altitude tracking errors occurred within the first 10 seconds of the run, as the algorithm was initializing. At the nominal velocities of 25 and 30 knots the optimizations aborted due to infeasibility when the helicopter encountered the initial trajectory requiring ~25 degrees negative slope. We hypothesize that at higher flight path angles, the limits of helicopter maneuverability at high speeds are approached increasing the probability of infeasibility in the optimization. However, the nominal velocities of 25 and 30 knots were likely not

attainable for this terrain section due to the lack of algorithm initialization before encountering the slope, rather than the steepness of the slope itself.

5.3.2 **Terrain Section 2**

The second terrain section to be investigated was selected for its tame first half to promote algorithm initialization followed by ramping up of required flight path angles for the second half of the run. Unlike Terrain Section 1, Terrain Section 2 starts out with a much more gentle ~10 degree downward slope but then instead of getting shallower as the terrain progresses, there are two small hills encountered on either side of a valley creating a sequence of gradually increasing reference flight path angle magnitudes. The sequence of flight path angles starts with an initial flight path angle of -10 degrees, followed by +20 degrees as the first hill is encountered. Then as the hill is passed, the reference flight path angle of -25 degrees is commanded into the valley followed by +30 degrees ascending out of the valley at the end of the run. This combination of terrain features was selected in the hopes of initializing the algorithm well before the more severe flight path angles are encountered, enabling the maximum feasible flight path angle associated with the remaining feasible nominal velocities of 10, 15 and 20 knots to be found.

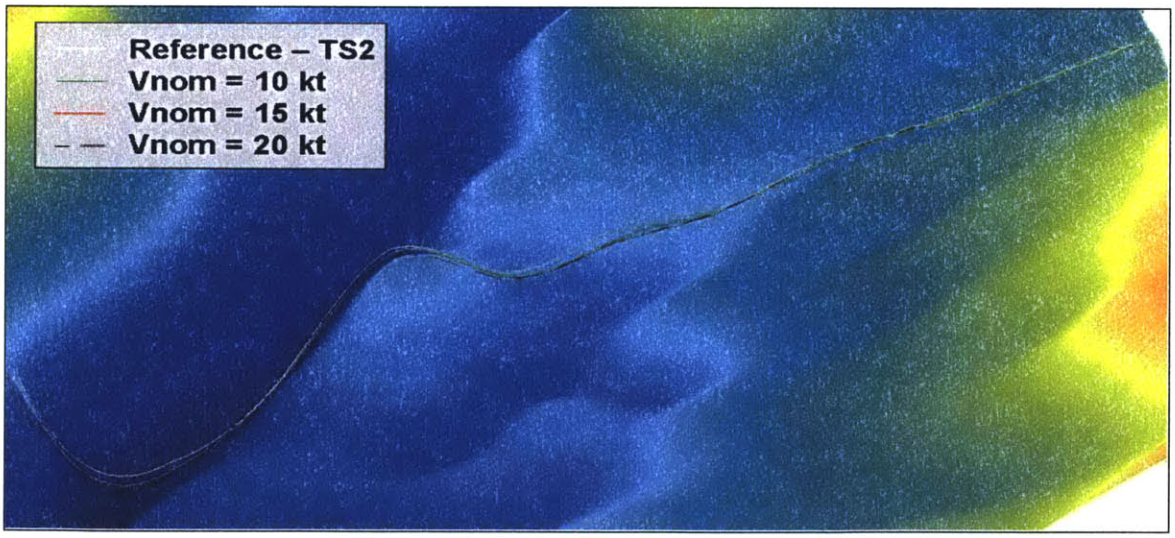

Figure 5-13: Terrain Section 2 results for varying nominal velocities plotted with the initial nominal reference line connecting the initial and final waypoints.

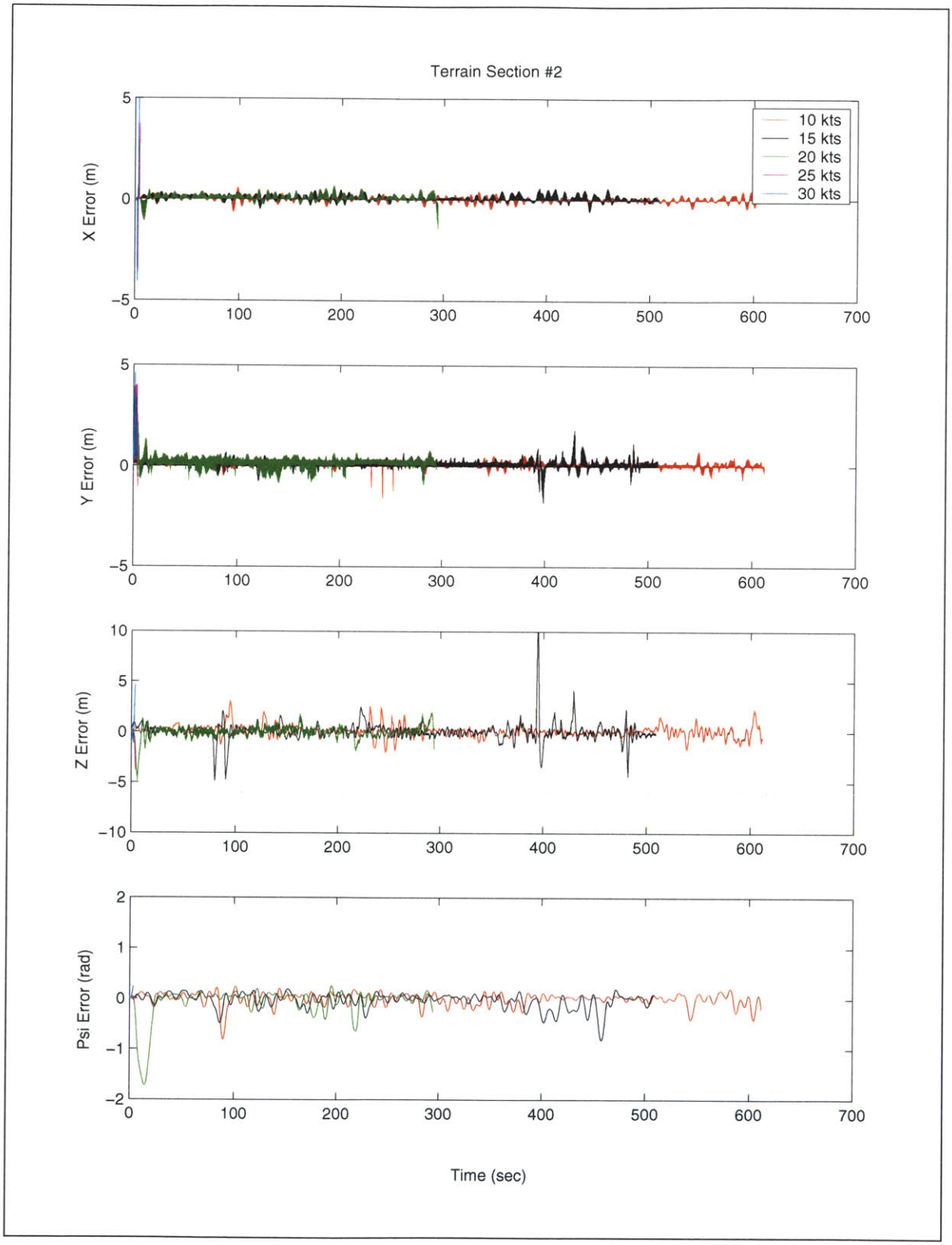

Figure 5-14: Terrain section 2 reference tracking errors in X, Y, Z and Ψ for nominal velocities varying from 10 to 30 knots.

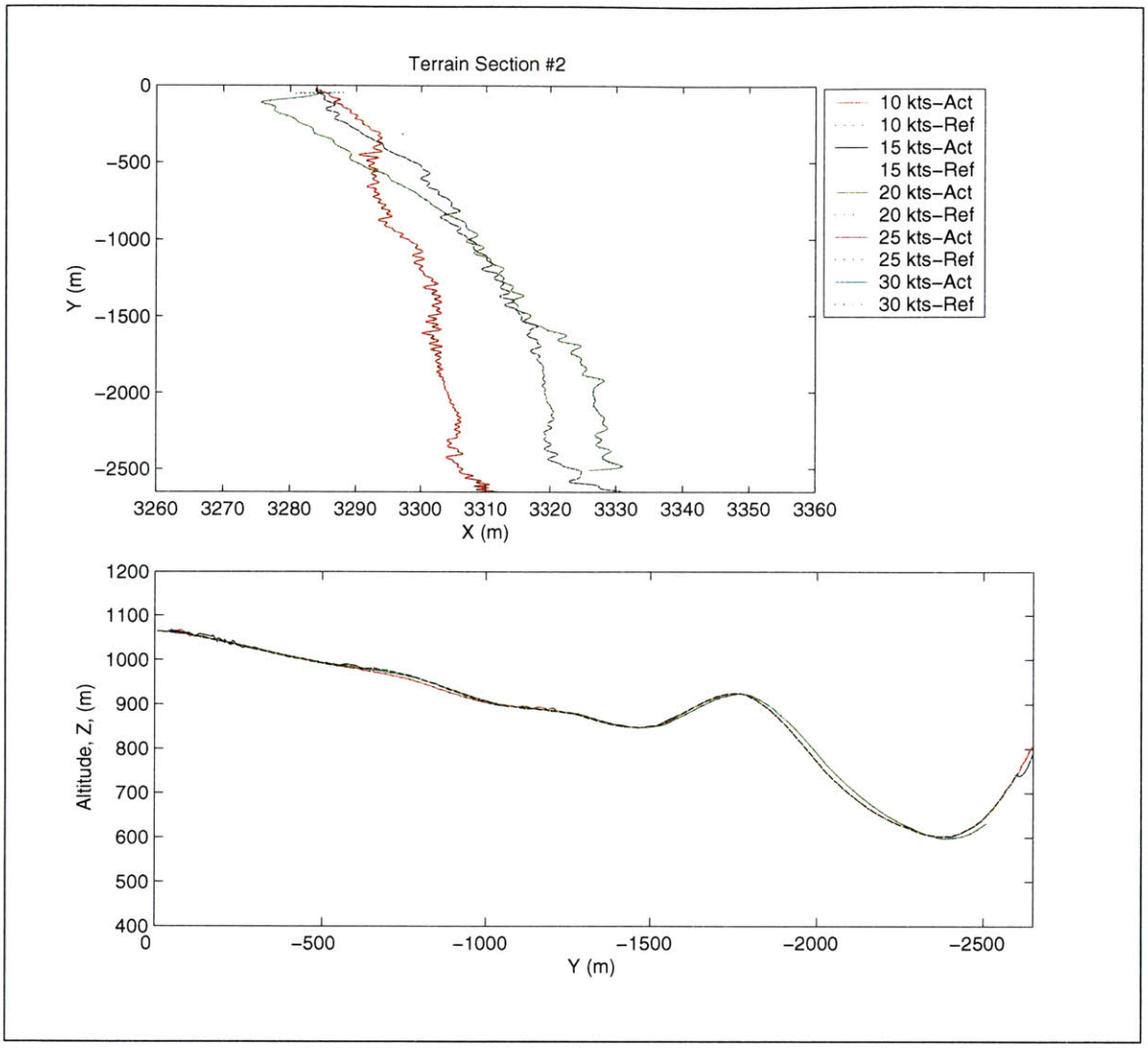

Figure 5-15: Terrain section 2 plotted with simulation data at nominal velocities varying from 10 to 30 knots. Upper figure is a bird's eye view of the performance. Lower figure is a cross-sectional view.

5.3.2.1 *Lateral Tracking Variation – Terrain Section 2*

The lateral tracking trends for Terrain Section 2 are very similar to that of Terrain Section 1. The same lateral drift appears in the first half of the run, showing the same variation with nominal velocity (lower nominal velocities of 10 and 15 knots yielding smaller lateral offsets from the nominal reference connecting the initial and final waypoints.) As with Terrain Section 1, at around Y = -1500 the controller begins to curb the drift, however this point happens to coincide with the first slope increase in Terrain Section 2. This appears to have a negative effect only on the 20 kt nominal velocity as a further lateral drift is instigated at this point. Notably different from the Terrain Section 1

data, though the controller appears to arrest the drift midway through the Terrain Section 2 runs, there is no definitive sign of the helicopter moving back toward the final waypoint. This may be due to the fact that the most severe flight path angles required in these runs occur between Y = -1500 and Y = -2500 preventing the controller from removing the lateral drift due to the higher demands for terrain tracking response.

The heading tracking error also reflects the redirection of control effort to track altitude as the terrain gradually becomes more difficult. The maximum heading tracking errors (algorithm initialization transients excluded) occur for all three feasible speeds during the second half of the run when the required flight path angles begin to ramp up. It is expected that lateral tracking (including heading) will suffer as more is required from longitudinal tracking due to the lower penalty on lateral tracking as compared to longitudinal tracking in the cost function definition. Figure 5-13, a 3D plot of the resultant trajectory for each nominal velocity illustrates the relative magnitude of the lateral drift with respect to the total distance traveled.

5.3.2.2 *Longitudinal Tracking Variation – Terrain Section 2*

The increase in altitude tracking error with increased nominal velocity becomes more pronounced as the terrain of the traversed cross section becomes more severe. It was hypothesized in Section 5.3.1.2 that at the higher flight path angles the limits of the helicopter maneuverability at high speeds are approached. This is further supported by the tracking error data as the required flight path angles increase through Terrain Section 2. The maximum altitude tracking errors for this terrain section are -3 meters for Vnom = 10 kt, 10 meters above and -5 meters below the reference for Vnom = 15 kt and -5 meters for Vnom = 20 kt. The maximum error magnitude for nominal velocity of 15 kt is mentioned above and below the reference because even though the error above the reference is greater in magnitude, it still has no effect on the feasibility of the optimization. The error magnitude below the reference still lies well within the 6 meter limit imposed by the ground (indicated on the Z-tracking error plot in Figure 5-14 by the dotted line). It is also worth noting that the optimization aborts at Y = -2500 for the 20 knot nominal velocity, just as the reference trajectory is demanding the highest angle of attack for this terrain section, which is 30 degrees. The published helicopter limits show

that helicopter maneuverability crests at a certain operational speed then rapidly decays due to quicker vehicle control saturation associated with increase in speed. This combination of speed and flight path angle appears to have broached the vehicle's dynamic limits, leading to the observed infeasibility of the optimization in this simulation.

5.3.3 **Terrain Section 3**

The third terrain section was selected because it is slightly more difficult than Terrain Section 2, introducing a 30 degree upslope a quarter of the way through the run. The objective for testing this terrain section is to confirm the maximum acceptable reference flight path angle for the 20 knot nominal velocity at 30 degrees and hopefully move towards finding a similar limit for the nominal velocities 10 and 15 knots.

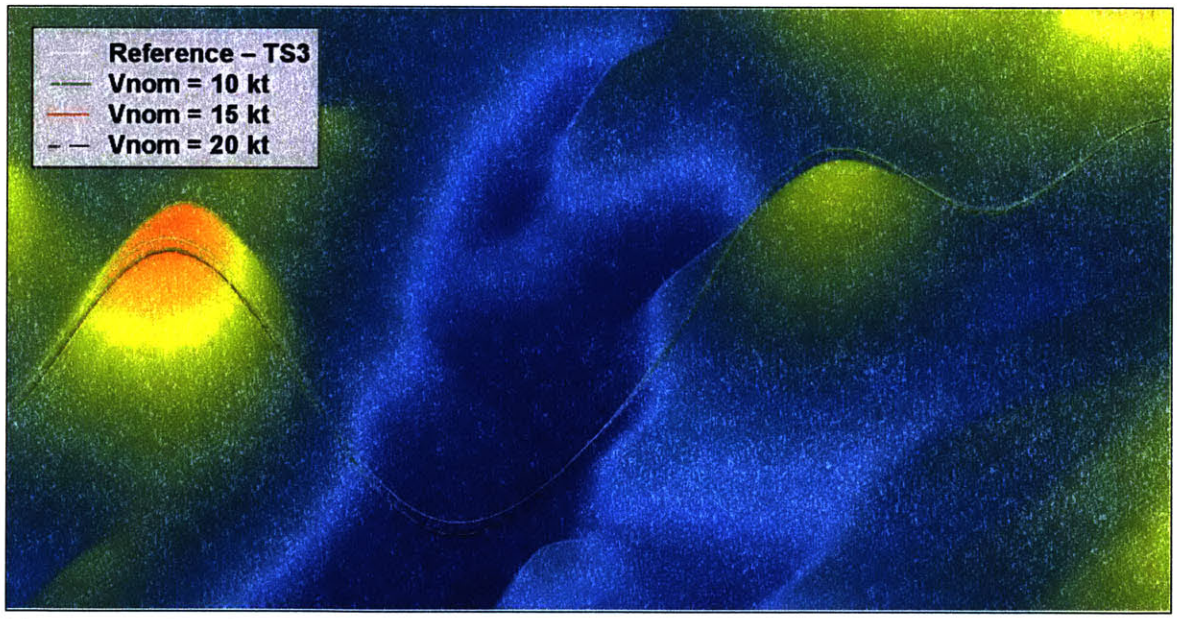

Figure 5-16: Terrain section 3 results for varying nominal velocities plotted with the initial nominal reference line connecting the initial and final waypoints.

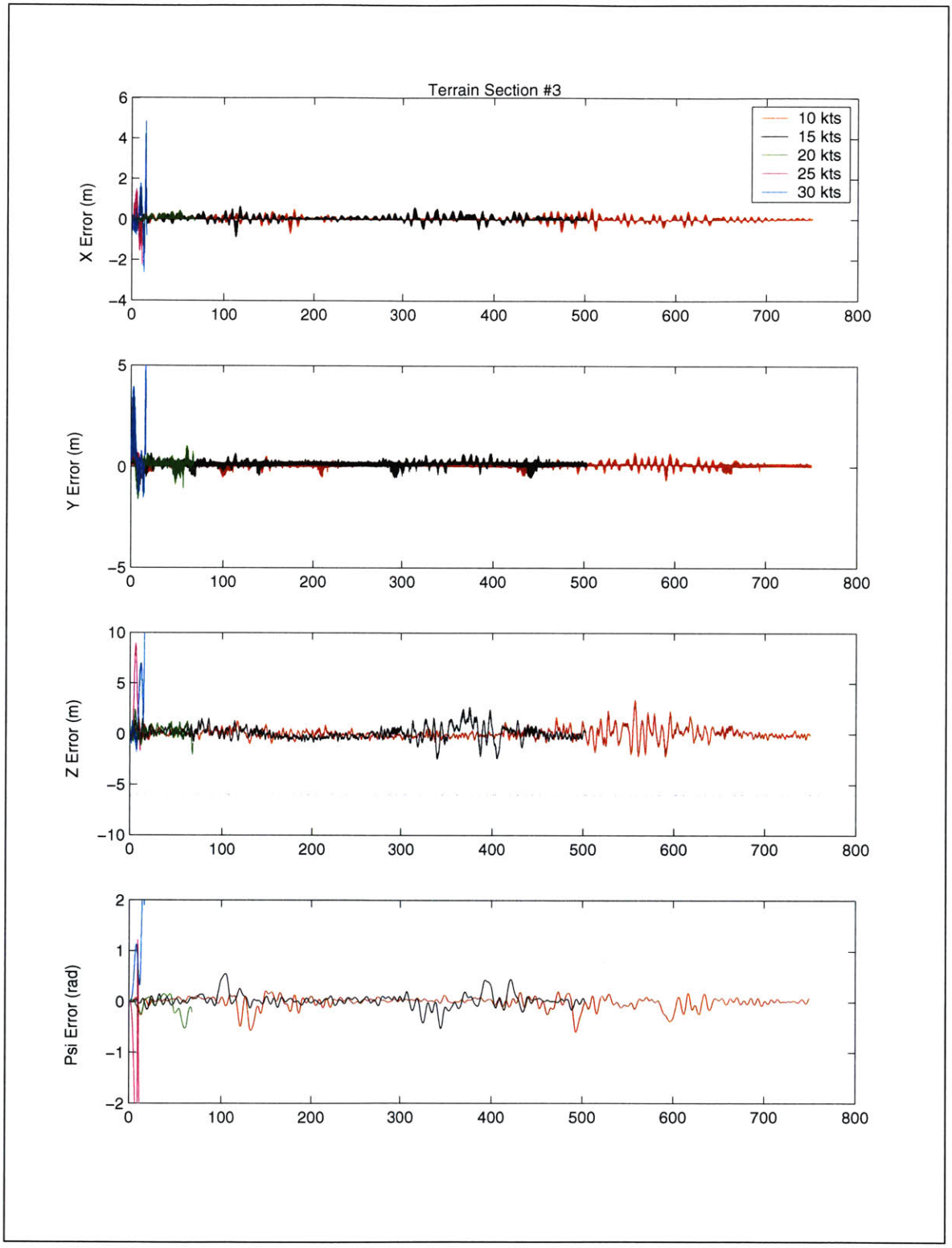

Figure 5-17: Terrain section 3 reference tracking errors in X, Y, Z and Ψ for nominal velocities varying from 10 to 30 knots.

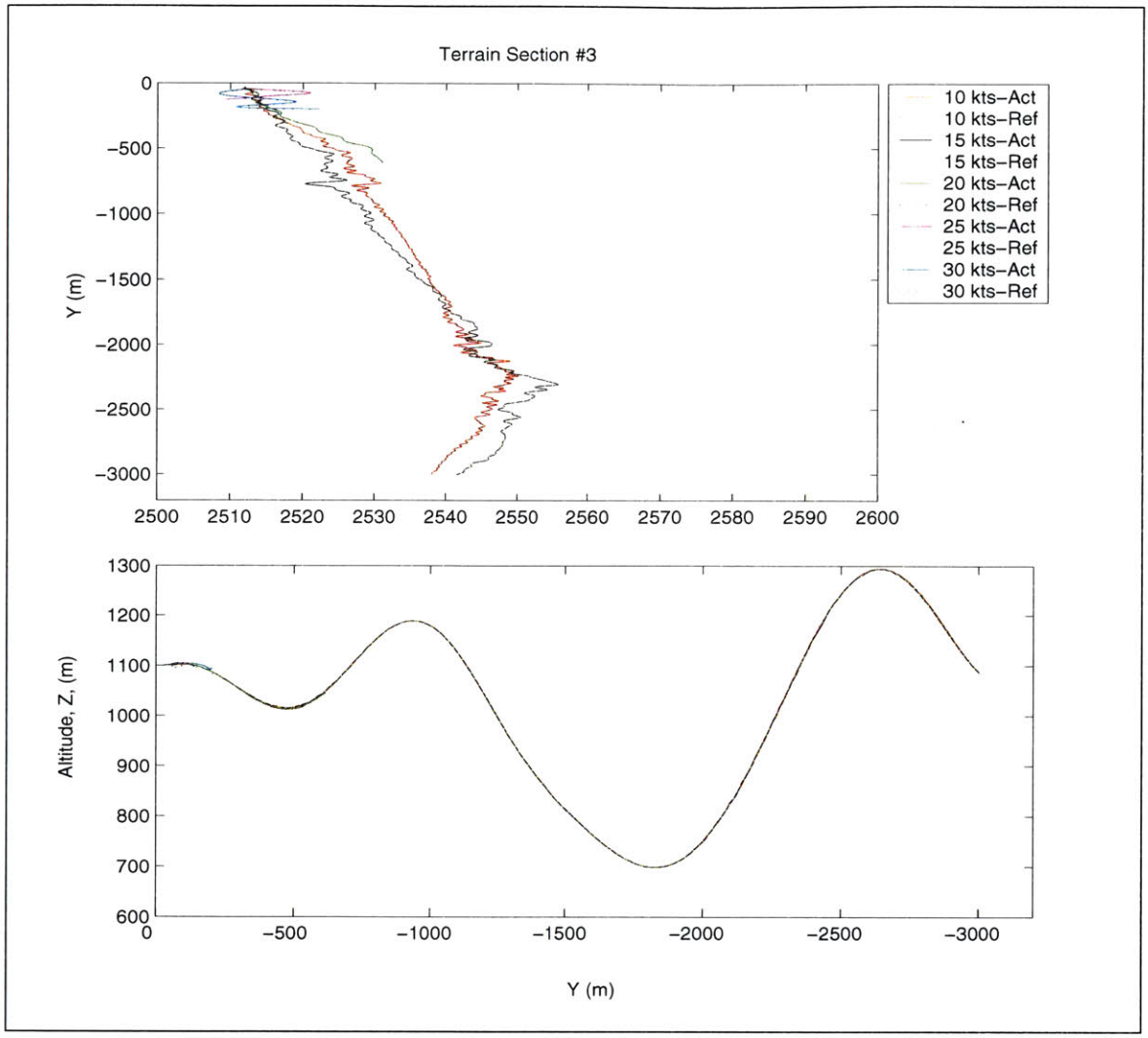

Figure 5-18: Terrain section 3 plotted with simulation data at nominal velocities varying from 10 to 30 knots. Upper figure is a bird's eye view of the performance. Lower figure is a cross-sectional view.

5.3.3.1 *Lateral Tracking Variation – Terrain Section 3*

The same lateral drift characteristics seen in Terrain Sections 1 and 2 are apparent in Terrain Section 3, only as observed in the investigation of the Terrain Section 2 data, the lateral drift is worsened by the extreme terrain features. This is apparent in Figure 5-18 which shows that the control effort used to either redirect the helicopter towards the final waypoint, or at least arrest the lateral drift, is being diverted by the controller to aid in longitudinal tracking until the helicopter has passed the most difficult sections. This is a positive result even though the outcome is a higher magnitude of lateral drift because it shows that the applied TF/TA ratio is doing its job, sacrificing lateral tracking to aid in longitudinal tracking when the terrain becomes more difficult.

5.3.3.2 *Longitudinal Tracking Variation – Terrain Section 3*

As with Terrain Sections 1 and 2, nominal velocities of 25 and 30 knots aborted their optimizations due to infeasibility as they met the more severe flight path angle requirements. Also consistent with Terrain Section 2, the 20 kt nominal velocity run became infeasible when the 30 degree maximum flight path angle was encountered. Though the maximum flight path angle reference for feasibility of nominal velocities above 20 kts has not yet been clearly defined through these simulations, it appears that ~30 degrees is the maximum flight path angle for a 20 kt nominal velocity. Again, none of the feasible runs violated the altitude constraint (set 6 meters below the reference trajectory to preclude terrain collision, indicated on the Z-tracking error plot in Figure 5-17 by the dotted line) at any point in the simulation.

5.3.4 **Terrain Section 4**

The fourth and final terrain section was selected due to its extreme (increasing steadily in magnitude to ~60°) flight path angle requirements.

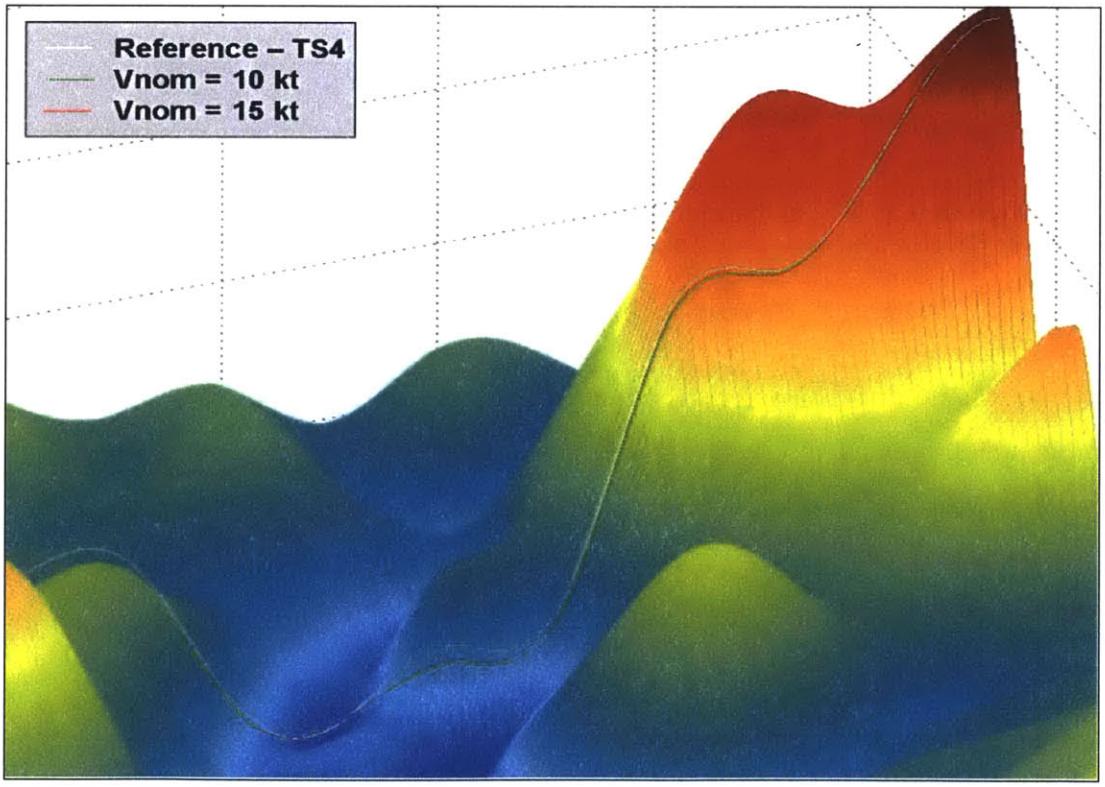

Figure 5-19: Terrain section 4 results for varying nominal velocities plotted with the initial reference line connecting the initial and final waypoints.

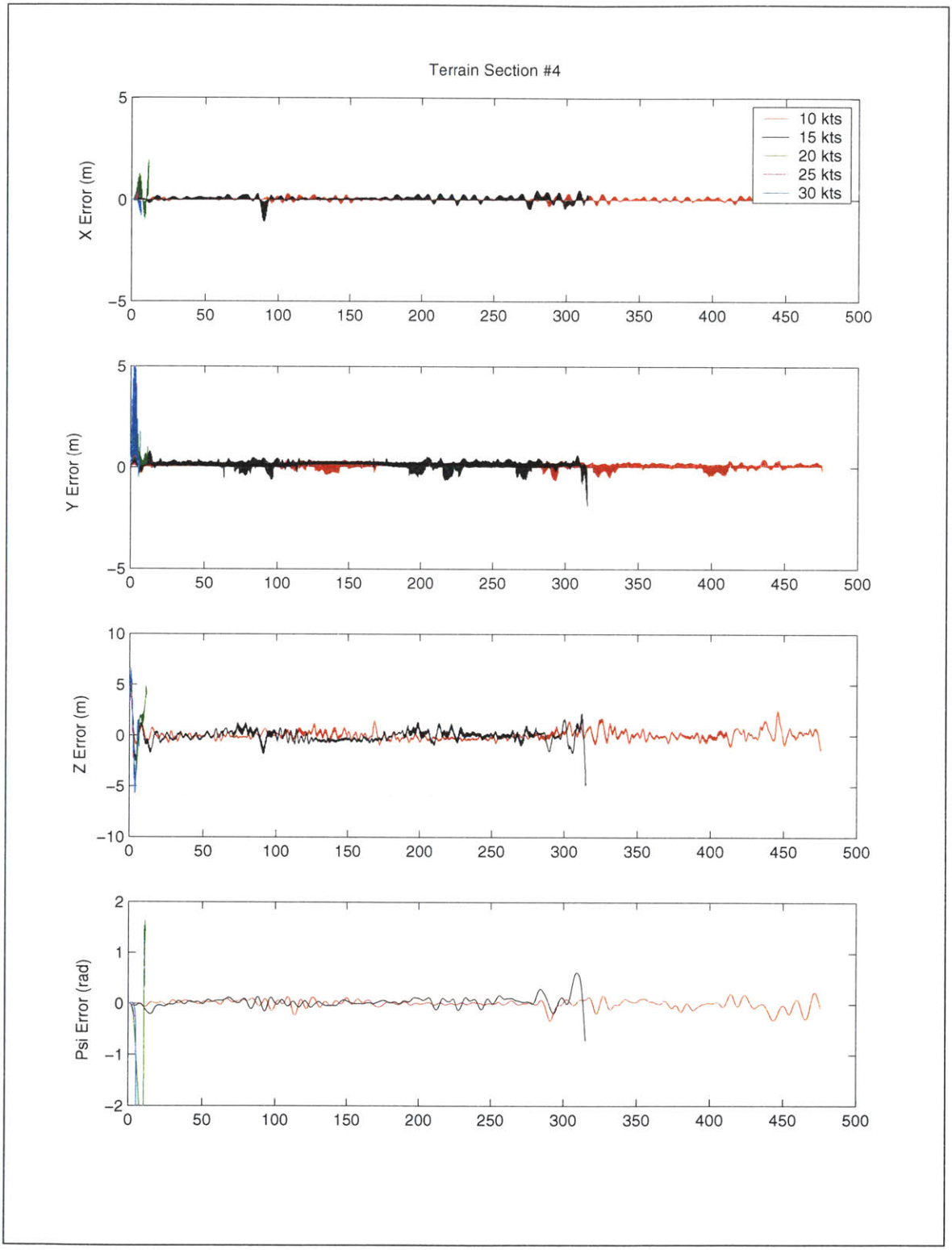

Figure 5-20: Terrain section 4 reference tracking errors in X, Y, Z and Ψ for nominal velocities varying from 10 to 30 knots.

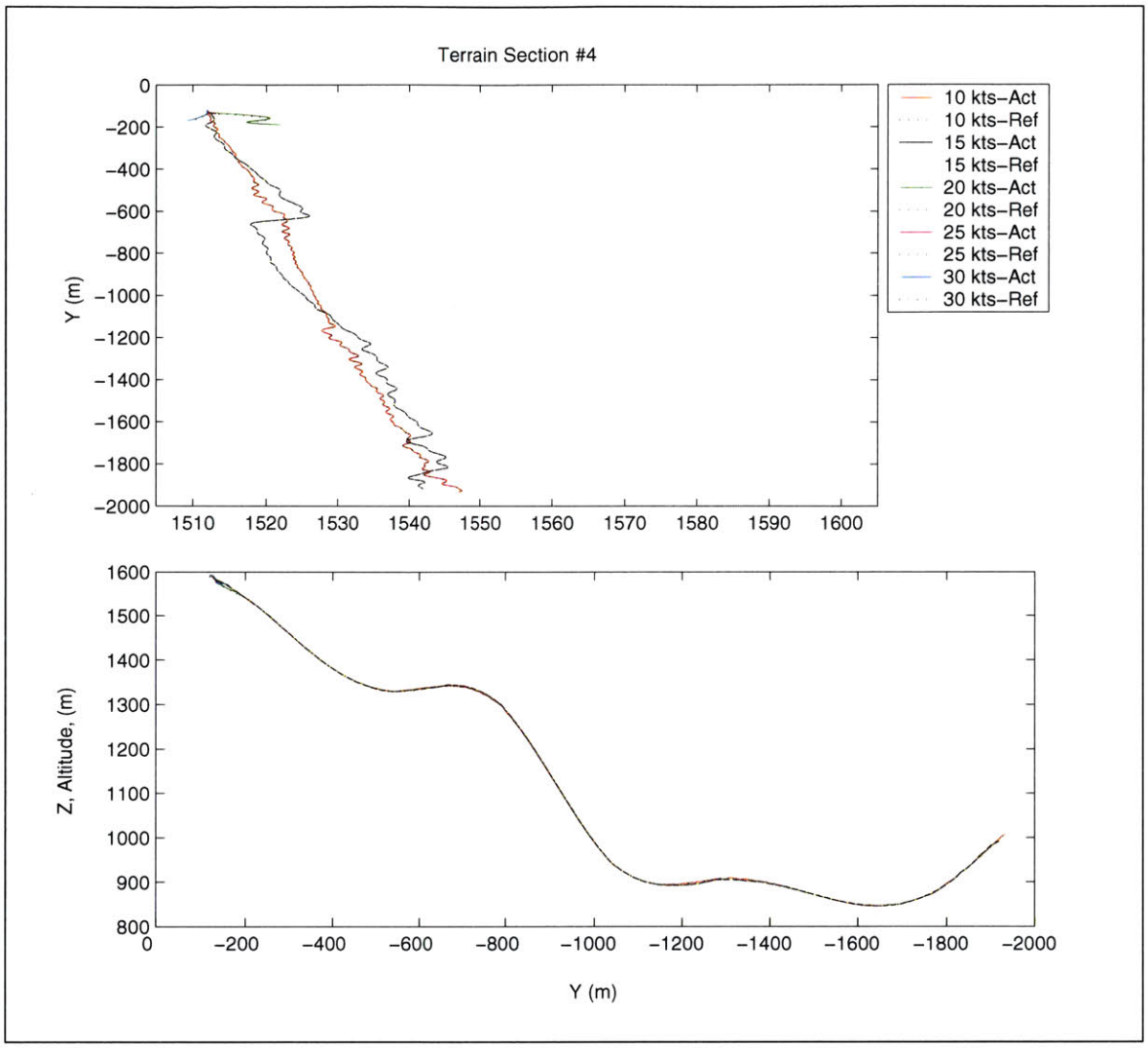

Figure 5-21: Terrain section 4 plotted with simulation data at nominal velocities varying from 10 to 30 knots. Upper figure is a bird's eye view of the performance. Lower figure is a cross-section view.

5.3.4.1 *Lateral and Longitudinal Tracking Variation – Terrain Section 4*

The tracking for all four states is basically the same for this terrain section as that of the previous, less severe, terrain cross sections. The major difference to note is where the infeasibility occurs as the nominal velocity is varied. Consistent with all previously tested terrain sections, the 25 and 30 knot runs were again unable to initialize due to the extreme initial flight path angle and the optimizations were aborted in the first 10 seconds of each run. Also consistent with its previous performance, the 20 knot nominal velocity run also aborted shortly thereafter, unable to initialize with the 40 degree decline. Both 10 and 15 knot nominal velocity runs were able to successfully initialize and track the extreme

terrain slopes, including the ~60 degree decline between Y = -800 and 1000. A limit was discovered, however, when they reached the upslope of ~42 degrees. Since the vehicle cannot respond fast enough to meet the 40 degree slope with a nominal velocity of 10 or 15 knots, it falls short, violating the constraints resulting ground collision, thus optimizations in both runs were forced to abort due to infeasibility. Suggestions for how this algorithm can be made more robust to handle all types of terrain features are included in Section 7.2.

5.3.5 **Final Notes on Terrain Severity vs. Velocity Sensitivity**

Having determined the angle of attack limits for 10, 15 and 20 knots on challenging terrain sections, an additional, less severe, terrain section was tested to obtain an upper limit on nominal velocity for this vehicle given the current horizon length. The first significant feature of this terrain section is its initial 2500 meter long 4 degree decline, conducive to algorithm initialization at most nominal velocities. This initialization is followed by the maximum slope of the terrain section, an 8 degree incline, for the next 1500 meters. Finally, the remaining 2000 meters are split between a 6 degree down-slope and a 7 degree incline.

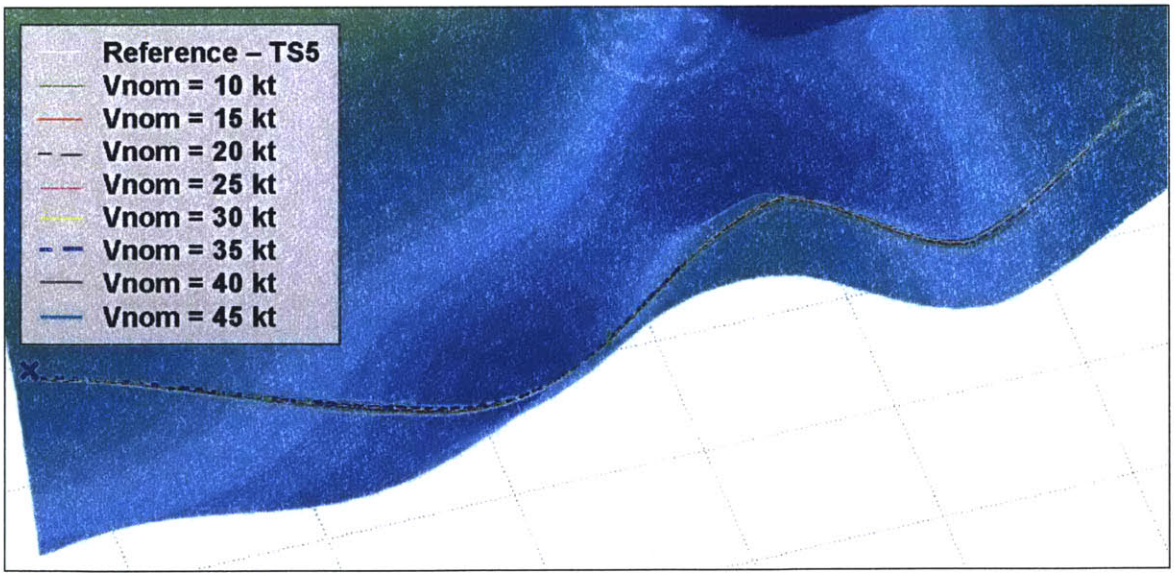

Figure 5-22: Terrain section 5 results for varying nominal velocities plotted with the initial reference line connecting the initial and final waypoints.

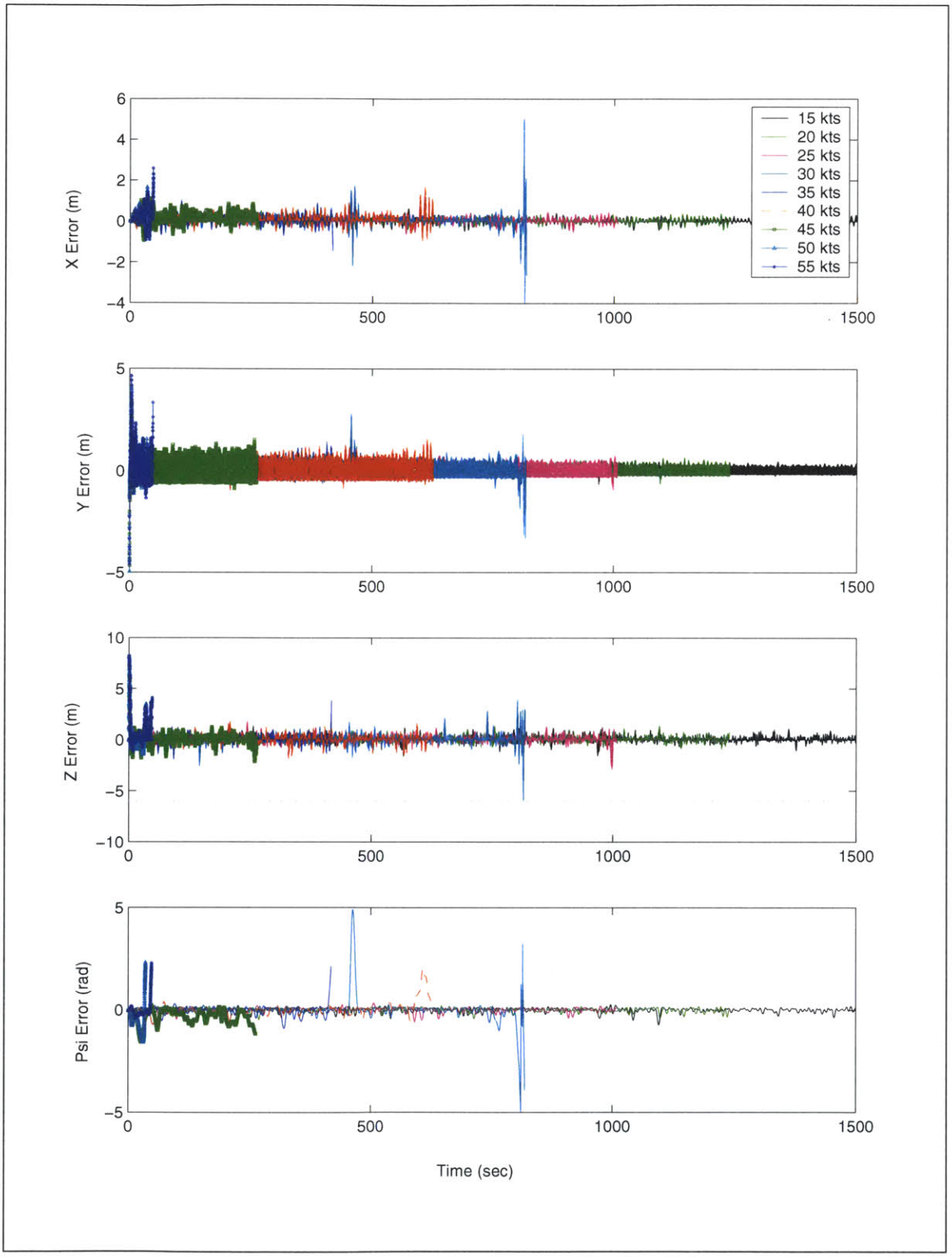

Figure 5-23: Terrain section 5 reference tracking errors in X, Y, Z and Ψ for nominal velocities varying from 15 to 55 knots.

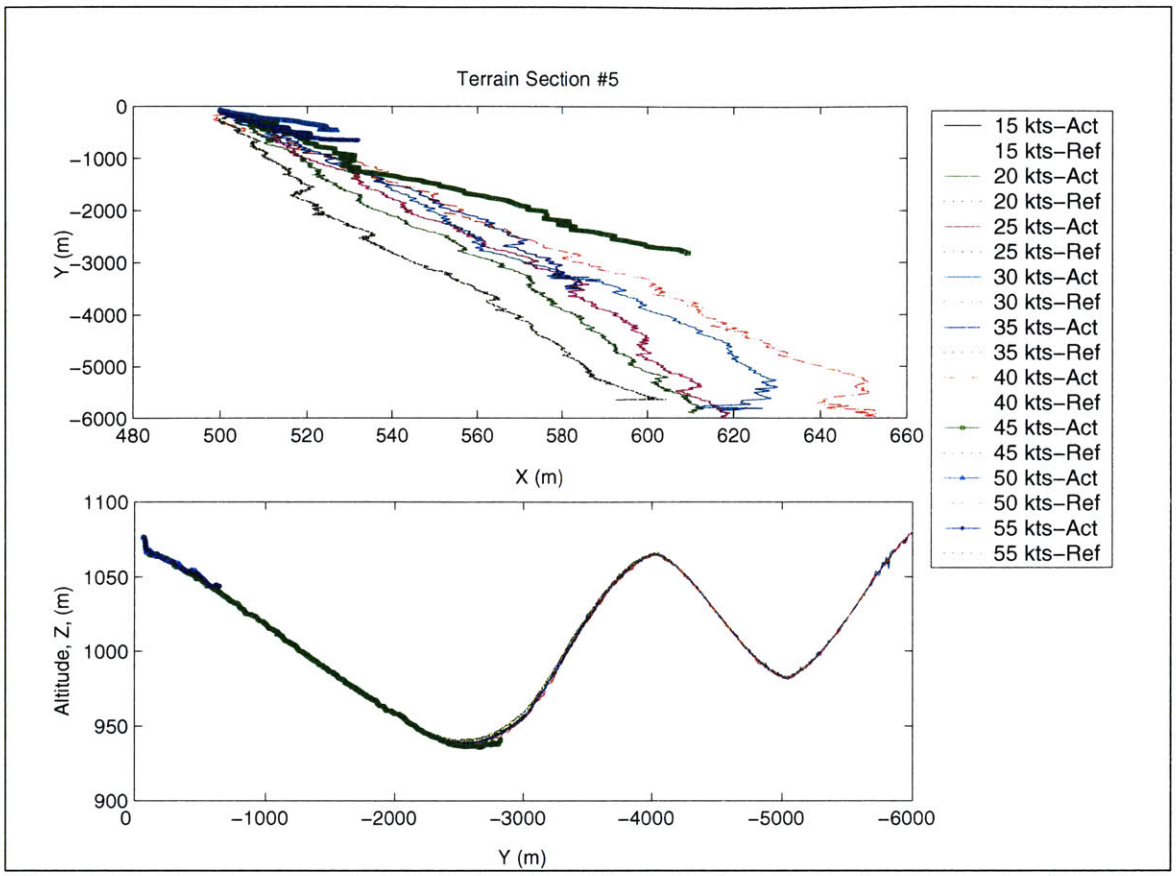

Figure 5-24: Terrain section 5 plotted with simulation data at nominal velocities varying from 15 to 55 knots. Upper figure is a bird's eye view of the performance. Lower figure is a cross-section view.

5.3.5.1 *Lateral and Longitudinal Tracking Variation – Terrain Section 5*

As hoped, this terrain section was realizable by all tested nominal velocities from 10 to 40 knots with tracking errors comparable to those seen for the 10 to 20 knot nominal velocities previously tested, and error magnitudes increasing slightly with nominal velocity as expected. This flat terrain section would be more characteristic of the type of terrain encountered in nap-of-the-earth flight missions flown "under canopy," or below the canopy of the trees in a forest, causing obstacle encounters to be very frequent as opposed to the sporadically placed obstacles dealt with in this research. It is worth mentioning the result over this terrain, however, because it underlines the conclusion that this velocity and terrain severity sensitivity study leads to: terrain severity, namely the maximum changes in flight path angle required by the reference trajectory over a given terrain, are the limiting factors on the maximum speeds attainable in terrain following

flight. A summary of the flight path angle limits defined for this helicopter model with the MPC parameters as defined in Section 5.2 is listed in Table 5-4.

Table 5-4: Summary of Nominal Velocity sensitivity to commanded angle of attack

Nominal Velocity	Flight Path Angle Limit
10 kts	~ 40°
15 kts	~ 40°
20 kts	~ 30°
45 kts	~ 8°

5.3.5.2 *Tracking Sensitivity to Velocity and Terrain with Alternative Bases*

As a precursor to this thesis, an abbreviated version of this analysis was performed and documented comparing the terrain sections 2 and 4 with the identical MPC algorithm implemented alternatively with 10 tent basis functions in [32]. The error values shown in [32] with the 10 tent basis function implementation concur with what the preliminary analysis exhibited in Section 4.3.2.3, yielding comparable error values between the 10 tent basis function implementation and the 5 Laguerre polynomial implementation. The sensitivity of tracking performance to nominal velocity and terrain severity (measured by commanded flight path angle) also exhibited similar results; however the initial and final waypoints were swapped between the two analyses. Therefore though the error data can be reasonably compared between the two sources, the commanded angle of attack limits are not a one-to-one comparison. Section 7.2 contains some detailed suggestions as to how this can be expanded upon in future research.

Chapter 6

CONSTRAINT BASED OBSTACLE AVOIDANCE

Recent obstacle avoidance strategies make use of two main ideas; reference trajectory modification to produce or enhance obstacle avoidance response [4, 5, 10], and definition of 'safe envelope' (either by hard or soft constraints) within which the new trajectory is to be planned [6, 10, 20]. The method applied in this research utilizes the 'safe envelope' idea; to handle obstacles, a constraint free, convex feasible space is defined that excludes the obstacle and terrain spaces and constrains the optimizer to produce the sequence of controls (N_P-interval long) that produce spatial trajectories within this feasible solution space. This assures obstacle and terrain collision avoidance.

In addition to the terrain itself, this chapter deals with two types of obstacles; trees, towers or poles which require the helicopter to fly around them, and utility wires which require overflight, (for this study, flight under wires is not considered a feasible solution). The interpolated terrain was populated randomly with the two different types of obstacles; lateral obstacles (trees, poles, etc.), modeled as cylinders which require 'go-around' obstacle avoidance response, and longitudinal obstacles (utility wires, etc.), modeled as walls which require 'go-over' obstacle avoidance response. Figure 6-1 shows the obstacle-populated terrain utilized in this chapter. For this study, it is assumed that sensor information is preprocessed to provide MPC with obstacle location and size information (including the discernment between lateral or longitudinal required obstacle

responses) within a 150 meter radius of our vehicle. The following sections describe the heuristics used to set the constraints for longitudinal (go-over) obstacle avoidance response and lateral (go-around) obstacle response, followed by the results obtained for each method. In both cases, constraints defined in the earth frame were created to form the obstacle free subspaces within which the optimizer will search for the optimal trajectories. The constraints were defined in this way because given the assumption of obstacle location knowledge in the earth frame from a higher level obstacle detection algorithm, it was very straightforward to directly apply them to the earth-frame position states (x,y,z) without having to transform to the body frame. See Section 7.2 for a discussion of recommendations for future work related to the actual sensor configuration and obstacle detection algorithms as they relate to the setting of constraints to enforce obstacle avoidance.

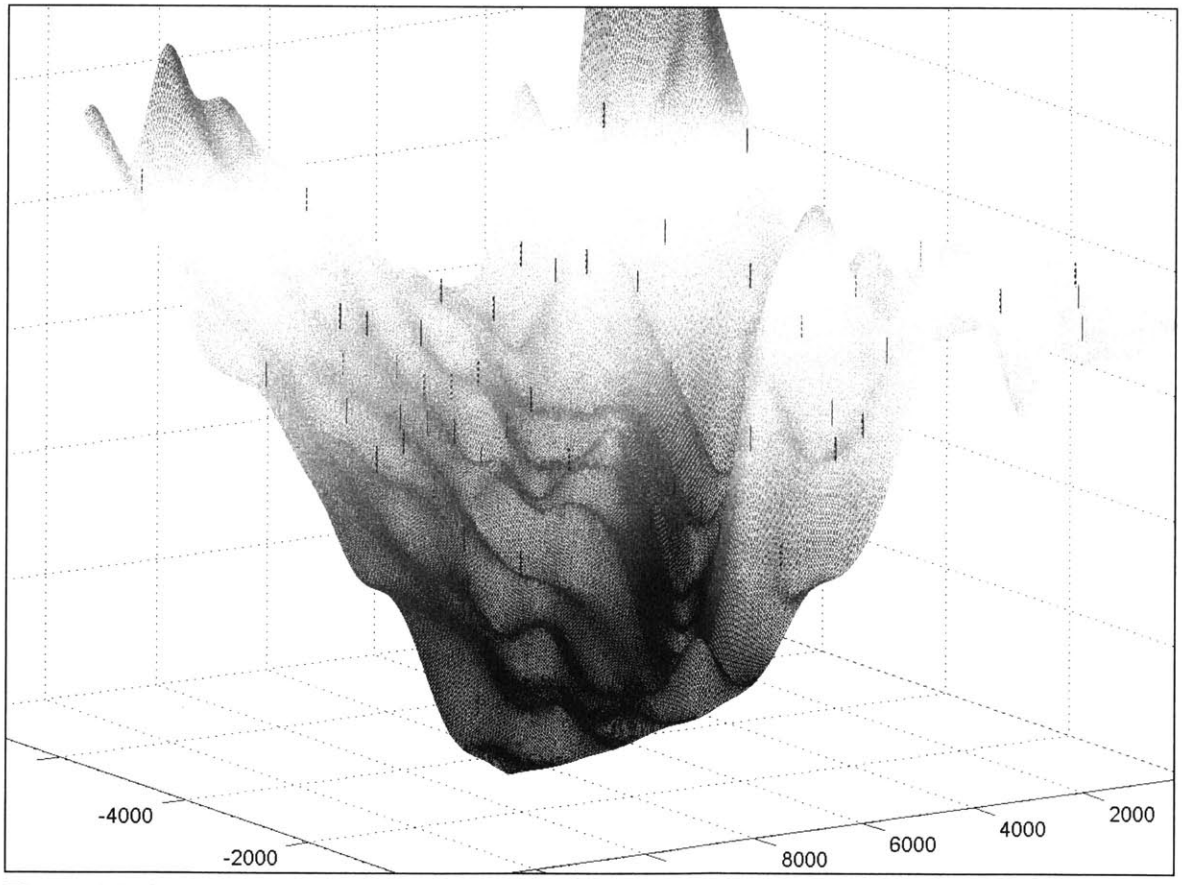

Figure 6-1: Obstacle populated sample terrain used for all obstacle avoidance work.

6.1 LONGITUDINAL OBSTACLE AVOIDANCE

6.1.1 Constraint Generation

When obstacles requiring a longitudinal response are detected within the prediction horizon radius (T_H*V_{NOM} meters), the convex feasible set is defined to constrain the optimizer to search for a path in an obstacle free subspace by setting a lower bound on the altitude for the trajectory portion in the vicinity of the obstacle. That lower bound, which is set to the minimum allowable altitude above the ground when no obstacles are detected, is increased to the minimum allowable altitude above the obstacle when the obstacle is encountered. This is illustrated for a prediction horizon length in Figure 6-2. For this application, a hard upper limit for altitude is never set although a maximum altitude limit could easily be imposed.

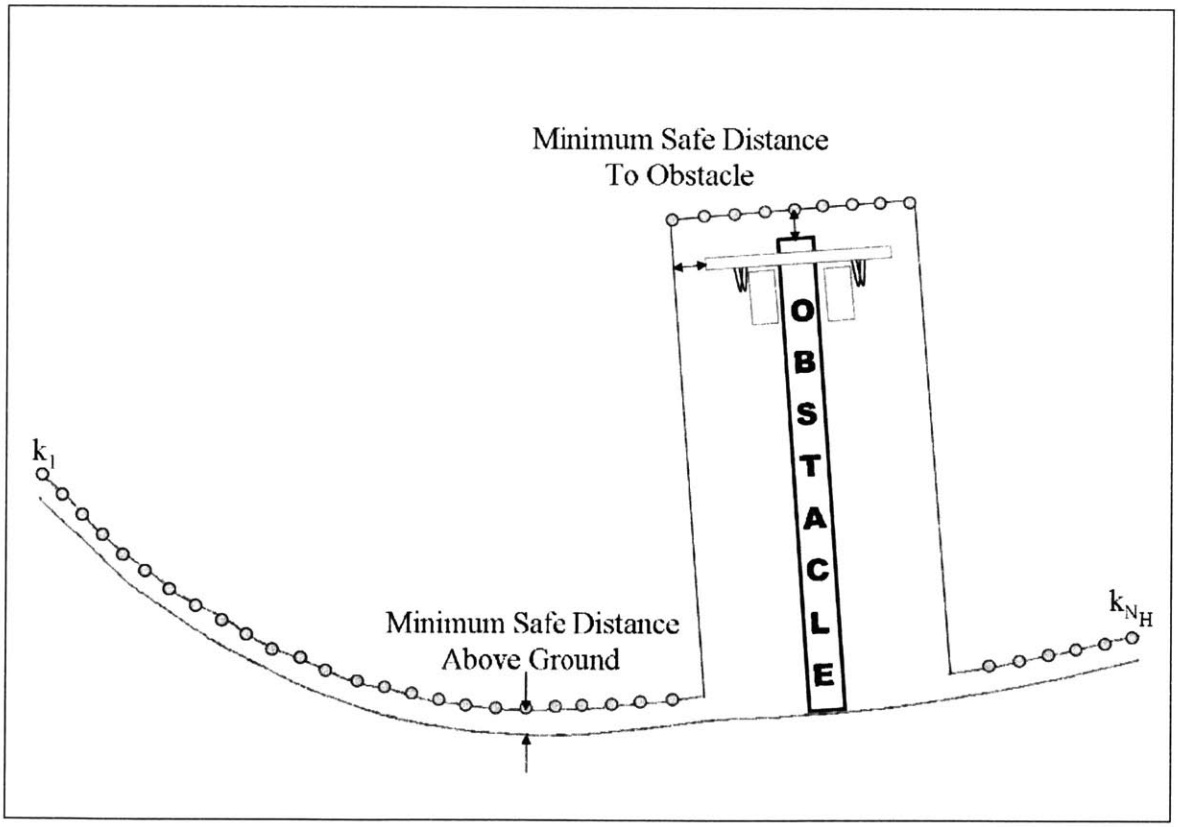

Figure 6-2: Lower bound altitude constraint on Z at each point along the prediction horizon, demonstrating the minimum safe distance above the ground and the change in constraint due to an obstacle requiring longitudinal response.

The Z-constraint cannot be implemented without a modification to the reference trajectory due to the discontinuous jump in the constraint upon encountering an obstacle. This is because even though the look-ahead provided by MPC begins to anticipate the change in constraints as the helicopter draws closer to the obstacle, the anticipation is not fast enough to command the helicopter above the new obstacle-updated altitude constraint before the new constraint is enforced. Therefore, strictly changing the constraint to account for the obstacle without trajectory modification causes the problem to become infeasible when the longitudinal obstacle is encountered even at low nominal velocities.

The reference trajectory modification that was implemented to remedy this infeasibility problem involved updating the reference trajectory upon obstacle detection to reflect the presence of the obstacle well before the obstacle-updated altitude constraint is enforced. This is illustrated in Figure 6-3.

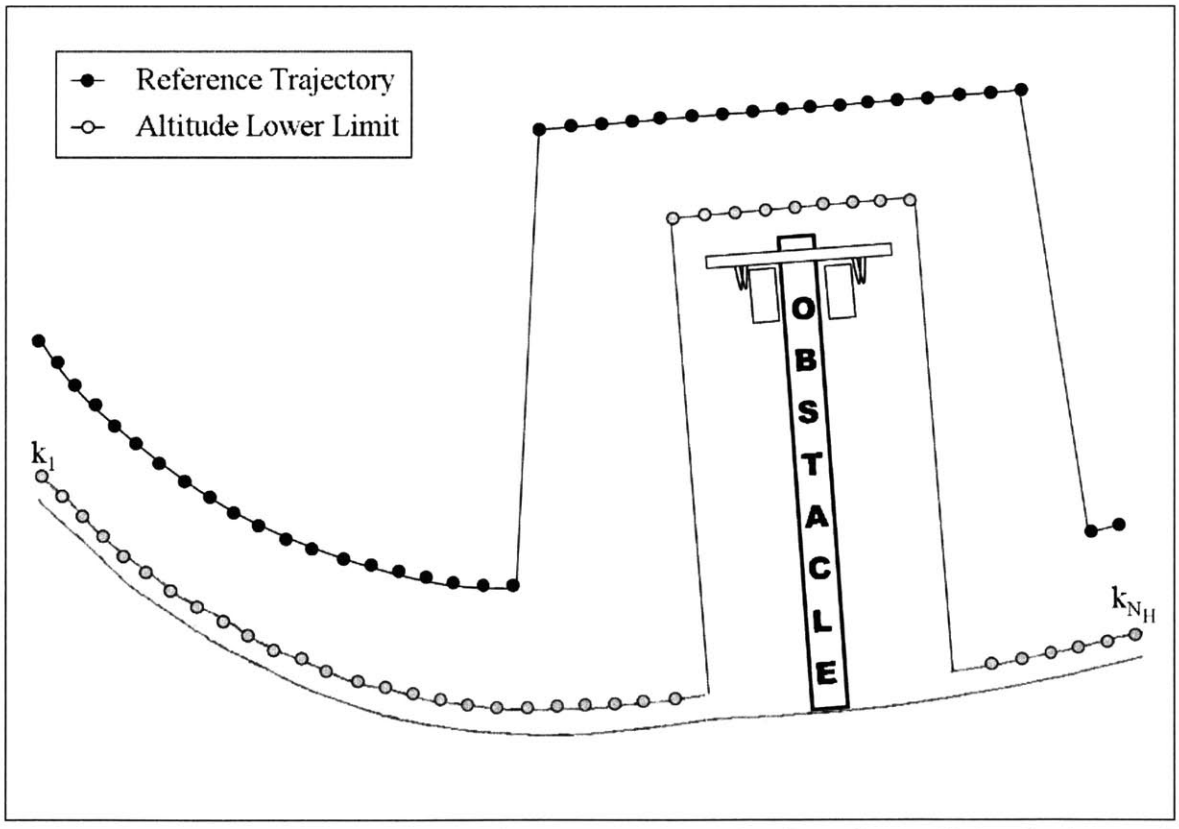

Figure 6-3: Lower bound altitude constraint on Z at each point along the prediction horizon, and altitude reference trajectory for the same horizon reflecting modification due to the presence of an obstacle requiring longitudinal response.

As the figure illustrates, when the helicopter enters the vicinity of the obstacle there is the same discontinuous jump in the reference trajectory as in the constraint. This ensures that the helicopter altitude will be higher than the obstacle height constraint by the time it reaches the time-step when the altitude constraint is increased from ground to obstacle height. However, since the reference trajectory is only a soft constraint in the cost function, its violation does not result in infeasibility. This allows lag and overshoot in the initial response to be tolerated within the distance on either side of the obstacle at which the trajectory is changed to the obstacle height. This distance must be set to enable the vehicle to obtain the specified altitude rise before reaching the obstacle. The distance required is governed by the vehicle's nominal velocity and the required altitude gain from the current reference altitude to the obstacle height, however for all nominal velocities tested in the following results this distance was set to 25 meters in order to accommodate the maximum nominal velocity tested.

6.1.2 **Results**

Figure 6-4 shows longitudinal obstacle avoidance with varying nominal velocities.

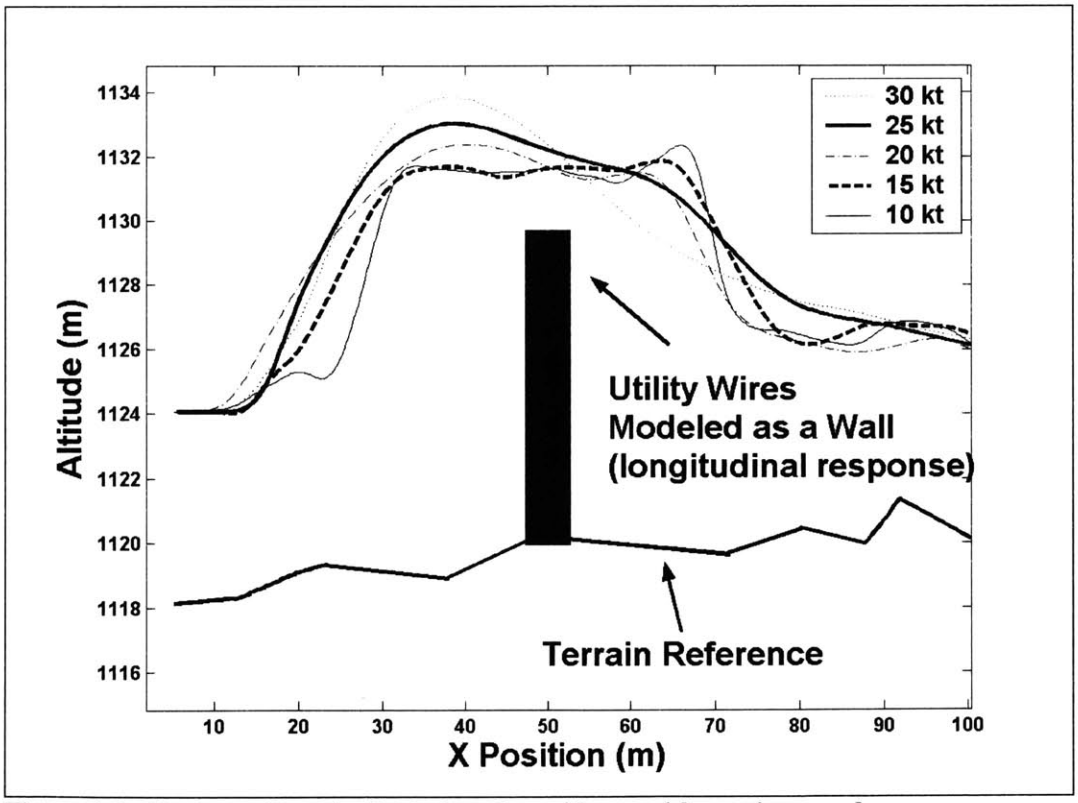

Figure 6-4: Close-up of longitudinal obstacle avoidance with varying speeds.

This constraint imposition with associated reference trajectory modification proved to be an effective method of longitudinal obstacle avoidance at all tested speeds. The safety margin was maintained above and around the obstacle and the reference was recaptured once the longitudinal obstacle was passed. MPCs anticipation of the upcoming jump in reference is clear. The higher nominal velocities start to climb over the obstacle earlier than the lower nominal velocities because they predict that they will not be able to handle the discontinuous jump in altitude reference (as previously stated, the reference modification is identical for all nominal velocities tested). In addition, there is a degredation of precise obstacle avoidance response as the nominal speed increases. The overshoot up to the new obstacle height reference increases as the speed increases. Furthermore, the step down in the reference once the obstacle is passed is cut more as speed increases. For nominal speeds above 30 knots, a greater reference lead distance on each side of the obstacle is necessary for safe obstacle avoidance.

6.2 LATERAL OBSTACLE AVOIDANCE: STRAIGHT LINE CONSTRAINTS

6.2.1 Constraint Generation

The convex feasible set for lateral obstacle avoidance is defined by setting an upper and/or lower bound determined from the location of the closest obstacles in range for the prediction horizon as defined in the earth frame. Figure 6-5 shows the feasible ranges for a variety of obstacle placements over the distance covered in three prediction horizons worth of time. If multiple obstacles are detected within the radius of the prediction horizon for a given point, the closest upper obstacle is set as the upper limit for that particular point and the closest lower obstacle is set as the lower limit. As Figure 6-5 shows, at $t = t1$, two obstacles are detected within the prediction horizon, therefore an upper bound is set on y at the lower edge of the upper obstacle and a lower bound is set on y at the upper edge of the lower obstacle. As with the longitudinal obstacles, since a different constraint is associated with each time-step along the prediction horizon, the upper or lower bounds can be removed once the vehicle has passed the obstacle, leaving the remainder of the horizon laterally unconstrained if there are no other obstacles within

that horizon. This is exhibited at t = t2 in Figure 6-5 after one complete prediction horizon has passed. There are no obstacles detected, therefore no lateral constraints are set until the next obstacle is detected. For example, at t = t3, the square obstacle is detected hence a lower bound on Y-position in the earth frame must be set as illustrated.

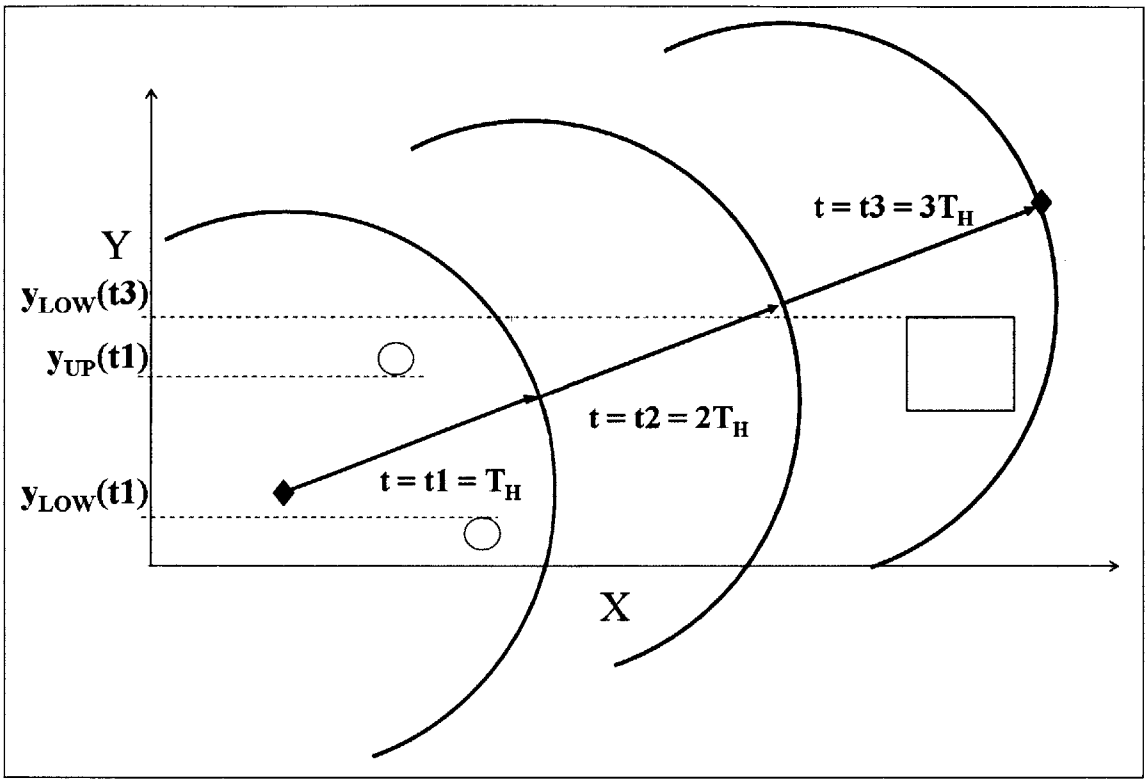

Figure 6-5: Lateral Constraint application method for obstacle avoidance where $y_{LOW}(t)$ and $y_{UP}(t)$ are the lower and upper bounds set on the helicopter earth frame position coordinate, y, in the MPC loop at time t.

6.2.2 Results

The helicopter successfully avoided all obstacles without violation of the set safety margins at the nominal velocities tested. The reference trajectory was redrawn successfully in the lateral obstacle case so that the new straight line path between the endpoint and the current position became the updated reference track. Figure 6-6 shows a bird's eye view of the lateral obstacle avoidance and reference regeneration.

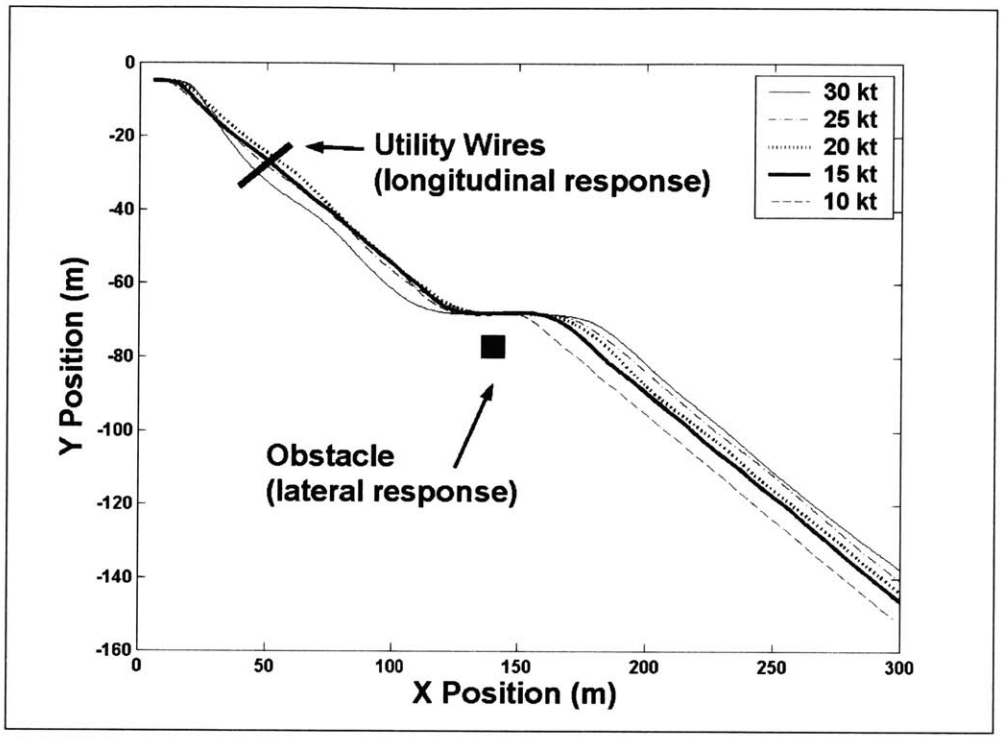

Figure 6-6: Y Position vs. X Position Close-up of Lateral Obstacle Avoidance and Reference Regeneration with varying speeds.

Notice that no reference trajectory modification is required for this lateral obstacle avoidance. This is because the algorithm's choice between setting an upper or lower bound depends purely on whether the vehicle is currently above or below the obstacle in question with respect to the axis to be constrained. Since the upper bounds are guaranteed by their creation to be set above the vehicle's current position, rather than the position at which the obstacle is directly encountered, and the lower bounds are similarly guaranteed to be below, the potential for infeasibility as seen in the longitudinal obstacle avoidance is eradicated.

6.3 Lateral (Go-Around) Obstacle Avoidance: Intermediate Waypoint Driven Constraints

A second method to define the convex feasible set for lateral obstacle avoidance involves the implementation of intermediate waypoint driven constraints as suggested in [10]. With this method, an intermediate waypoint is placed on one side or the other of the obstacle, orthogonally to the path of the vehicle, at a safe radius from the obstacle. The new constraint is defined by the line connecting the vehicle's current position and the

intermediate waypoint, appropriately excluding the side of the line on which the obstacle resides. Thus instead of setting a constant constraint on X or Y over the horizon, the less traversed axis is constrained as a function of the more highly traversed axis according to the aforementioned constraint line. Additionally, to aid the obstacle avoidance maneuver, a lateral reference trajectory modification is made. This modification directs the vehicle around the obstacle and ensures the optimization problem is feasible at the initial horizon steps when the obstacle has first been detected. The vehicle is directed around the obstacle by way of a second intermediate waypoint placed orthogonal to the vehicle's present path at double the defined safe radius from the obstacle. The lateral reference trajectory is diverted to this waypoint until the vehicle has safely passed the obstacle. Then, as with the method suggested in Section 6.2, the obstacle constraint is removed and the reference is redrawn to continue on to the final waypoint laterally unconstrained. The waypoint driven constraint and reference trajectory path modification are illustrated in Figure 6-7, along with the original reference trajectory path and the constraint that would have been set with the method described in Section 6.2 for comparison.

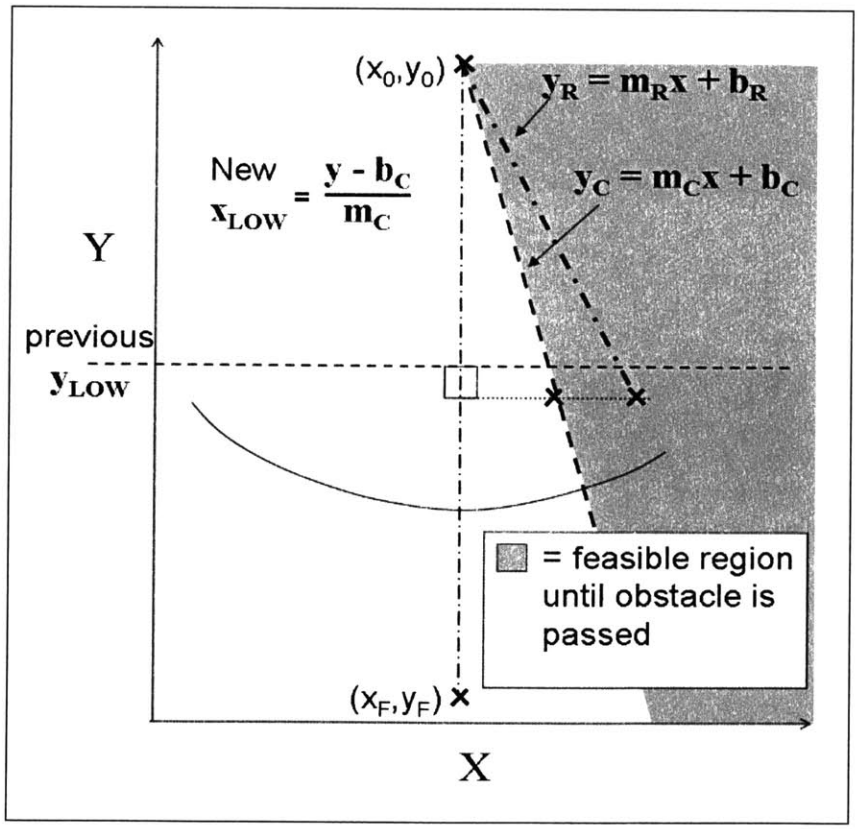

Figure 6-7: Lateral Constraint application method utilizing the placement of intermediate waypoints for the constraint generation (Y_C) and the reference trajectory modification (Y_R).

6.3.1 **Results**

This obstacle avoidance method successfully avoided all lateral obstacles tested. The reference trajectory was successfully redrawn from the intermediate waypoint to the final waypoint again once the obstacle was passed and the lateral constraints were lifted. Figure 6-8 shows a bird's eye view of the resultant trajectory utilizing this constraint generation strategy for lateral obstacle avoidance.

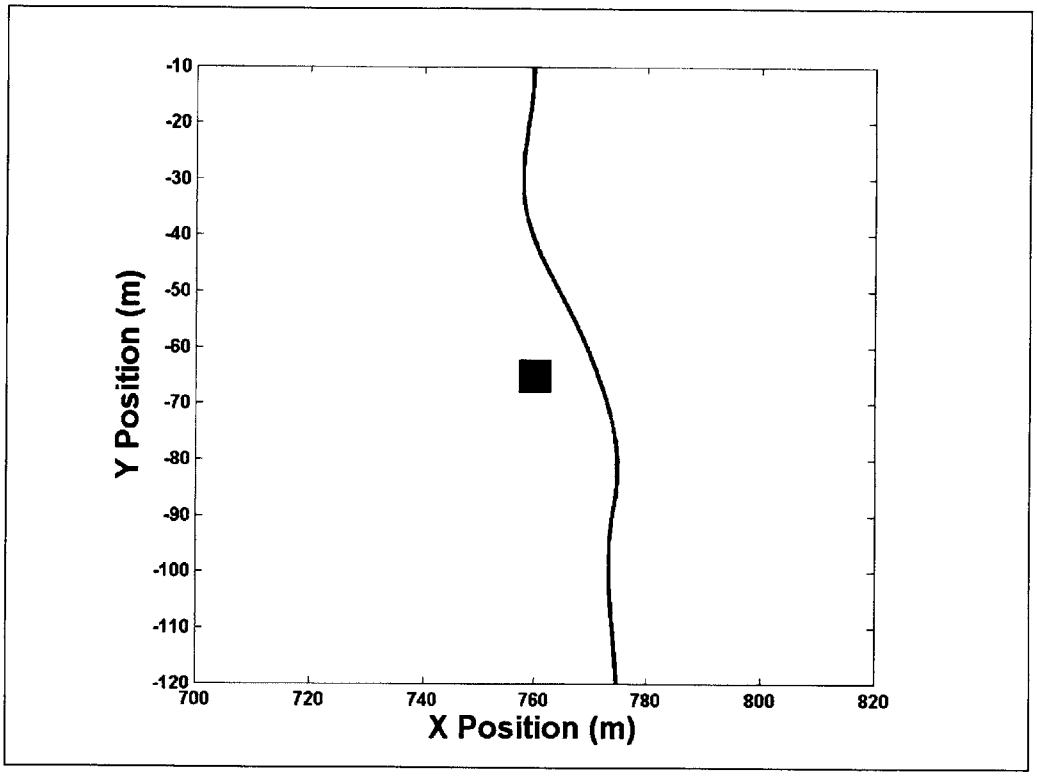

Figure 6-8: Earth frame Y-position vs. X-position close-up of lateral obstacle avoidance and reference regeneration with intermediate waypoint based constraint definition.

Chapter 7

CONCLUSIONS AND RECOMMENDATIONS

The objective of this thesis was the design and implementation of a Model Predictive Control based trajectory optimization method for Nap-of-the-Earth (NOE) flight including obstacle avoidance, emphasizing the mission objective of low altitude, high speed flight. This chapter provides a summary of conclusions reached throughout this research followed by recommendations for future work related to these topics.

7.1 CONCLUSIONS

This research sought to take advantage of MPC's repeating nature lending it to changing environments and objective functions. The trajectory generation and tracking problems were combined in a fashion facilitating real-time implementation and dynamic replanning based on sensor updates. MPC's construct, allowing the application of hard state and control constraints to multivariable problems, was also exploited to enforce obstacle and terrain collision avoidance by the imposition of hard constraints on desired output states.

Steps taken to reduce complexity in the optimization problem, including perturbational linearization in the prediction model generation and the use of control basis functions, proved to be a very effective method for reducing the problem size and optimizer computational time. Pre-implementation, candidate basis functions were

analyzed and evaluated for their trade-off between accurate approximation of the optimal cost/solution and reduction of the optimization complexity. Results showed an average decrease in the optimizer computational time of ~93% with the repeating-tent basis functions compared to the standard MPC formulation, and an average decrease of ~96% with the Leguerre polynomial basis functions. The Leguerre polynomial basis functions were selected and implemented for the duration of testing.

Combining the trajectory generation and tracking problems, the mission objectives of low altitude, high speed terrain following flight were met through precise tracking of the reference trajectory input into the cost function. The implementation of a terrain following to terrain avoidance (TF/TA) ratio to adjust the degree to which the terrain is tracked (longitudinally) at the expense of lateral tracking proved to be very effective. With the ratio of 10 determined through simulation, the terrain cross-section variation results showed that lateral tracking exhibited higher errors overall, specifically in the sections when altitudes approached the hard constraints set to enforce terrain collision avoidance. This shows that the choice of TF/TA ratio was successful in enforcing the mission objectives by sacrificing lateral tracking for longitudinal tracking when necessary.

Terrain and obstacle collision avoidance was accomplished through the establishment of hard state constraints. These state constraints were set by heuristics fashioned to create a 'safe envelope' in the earth frame bounding the subspace within which the optimizer is searching for the optimal trajectory. Firstly, the longitudinal constraints were simply set as the terrain elevation over which the helicopter is flying at any given time, with the exception of the go-over obstacle avoidance response which is addressed in the following paragraph. Results over a variety of sample terrain showed the altitude tracking errors approach, but never exceed, the hard ground constraint. Therefore the application of hard minimum altitude constraint has proven to be an effective method of preventing ground collision. However, a known problem associated with application of hard state constraints is the potential for an empty solution space, or infeasibility, leaving the optimizer without a control to output. When the different terrain cross sections (selected for their varying levels of difficulty, measured by commanded flight path angle) were tested at a range of nominal velocities from 10 to 30 knots, this problem was encountered. Vehicle dynamic

limits caused infeasibility such that as the terrain difficulty increased, the maximum nominal velocity attainable decreased. The flight path angle limits obtained based on the tests conducted in this research are displayed in Table 7-1. Future work regarding this issue of infeasibility in the application of this algorithm to difficult terrain is detailed in Section 7.2.

Table 7-1: Summary of Nominal Velocity sensitivity to commanded angle of attack

Nominal Velocity	Flight Path Angle Limit
10 kts	~ 40°
15 kts	~ 40°
20 kts	~ 30°
45 kts	~ 8°

Longitudinal obstacle avoidance (a fly-over obstacle response) was accomplished through hard state constraints differing from those set for terrain collision avoidance by constraint placement at a safe distance above the obstacle instead of the ground. Since this heuristic results in constraint discontinuity when such an obstacle is encountered, a reference trajectory modification from ~6 meters above the terrain to a safe distance above the obstacle is included well before the obstacle associated altitude constraint is enforced. This allows obstacle avoidance at higher nominal velocities because even though the trajectory tracking lags due to vehicle dynamic constraints, the hard constraints within the obstacle's vicinity are maintained.

Finally, two heuristics were presented in this thesis for the placement of constraints to carry out lateral obstacle avoidance (a go-around obstacle response). The first involves setting a constraint in the earth frame at a constant X or Y position while the helicopter is in the obstacle's vicinity, restricting the helicopter to one side of the obstacle until it is safely past the obstacle and the constraint is removed. This method resulted in successful obstacle avoidance at all velocities tested.

The second heuristic was designed by placing an intermediate waypoint at a safe distance on either side of the obstacle, orthogonal to the vehicle's trajectory. The constraint is implemented on X as a function of Y, defined by the line connecting the vehicle's XY-position when the obstacle was first detected and the intermediate waypoint. A reference modification is also made by directing the reference which

originally passes through the obstacle's space to a second intermediate waypoint placed outside of the constraint waypoint. This method also produced successful obstacle avoidance at all nominal velocities and configurations of initial and final waypoints tested.

7.2 RECOMMENDATIONS

The NOE flight problem is very complex, entailing many issues. These issues can be split into two main categories with respect to this thesis. The first group has been explored through this course of this thesis research, a subset of which merit further investigation. The second group was intentionally set aside by assumptions made in the problem definition. The following section details future research possibilities on these topics.

The first group of issues can be addressed to investigate and/or improve the robustness of this algorithm. Firstly, to simplify the problem definition, full state feedback was assumed throughout this research. Further research could include the addition of an estimator and investigation of the effects of state estimation on the robustness of the algorithm. A derivation of MPC with state estimation is available in [21].

Secondly, there are a two ways in which the simulation used in this thesis can be made to more closely represent actual implementation and conditions. As noted in Section 2.2.1, the non-linear equations of motion used to generate the prediction model were used as the 'truth' model in all of the simulations performed throughout this research. Future research could investigate the robustness of MPC with the selected tuning parameters by applying the algorithm to a more realistic plant model. Additionally, no outside disturbances were applied to the simulations presented within this thesis. One of the benefits of MPC application is the ability to input known or estimated disturbances such as winds directly into the prediction model to improve disturbance rejection. Such studies would be a logical next step in improving the overall fidelity of the simulation.

Thirdly, as suggested in Section 2.3, in actual implementation of this algorithm, the interpolation of the DTED data would be replaced by higher fidelity DTED data

supplemented by real-time sensor updates. An accurate sensor model and a method of real-time map update could be investigated in future work. Since this research also assumes that sensor information is pre-processed to provide the obstacle position and size information (including the discernment between lateral or longitudinal required obstacle responses), an obstacle detection algorithm could be designed and included in this investigation. Possible ideas for the collection and storage of this data for both static and dynamic obstacles are available as applied to a sonar-based underwater mapping system in [33]. Depending on actual sensor configurations and/or obstacle detection algorithms, future work might also investigate the definition of obstacle avoidance constraints by azimuth and elevation or by some other means in the body frame as opposed to their definition in the earth frame.

Finally, in actual implementation, though MPC has been shown to operate very effectively as an inner loop controller through this research, increased performance may be possible through the addition of an inner loop stability augmentation system (SAS) to track the state output by MPC. See [34] for a sample implementation of MPC both with and without an inner loop SAS as applied to ascent load management of a reusable launch vehicle.

Beyond the implementation concerns for future research investigation, an issue which the terrain following flight poses to a potential optimal control algorithm is the sensitivity of optimization feasibility and performance to nominal velocity and terrain severity. The vehicle dynamic limits have been identified as contributing factors through this thesis research, but a more rigorous investigation could be conducted into the mechanism which causes the infeasibility when the nominal velocity and terrain difficulty are increased. Such an investigation might include examination of optimization feasibility sensitivities to terrain severity/nominal velocity with an increased prediction horizon and/or sensor range. We hypothesize that an increased prediction horizon might yield the look-ahead necessary to allow for higher attainable nominal velocities over difficult terrain. Additionally, by using some guidance from a terrain analysis feature, the MPC cost function may be appended with a terminal cost-to-go beyond its prediction horizon to guarantee system stability. For this application, the new MPC cost would become:

$$J_i = \sum_{k=i}^{i+N_p-1} (y_{k+1} - r_{k+1})^T Q_k (y_{k+1} - r_{k+1}) + u_k^T R_k u_k + J_T(y_{i+N_p-1}, u_{i+N_p-1}) \quad (105)$$

where $J_T(y_{i+N_p-1}, u_{i+N_p-1})$ is the terminal cost-to-go specified as a function of the state and control at the final step in the horizon, y_{i+N_p-1} and u_{i+N_p-1} respectively. Investigation of methods for providing stability guarantees for a standard MPC implementation can be found in [35 – 38].

The last potential area for improvement in this algorithm is in the area of path planning. The spline path optimization and reference trajectory generator are essentially playing the role of path planning in the algorithm presented herein. However, the spline and reference trajectory generator could easily be integrated with a higher level path planner to include more than just terrain information. For example, a path planner could incorporate high level threat information into the reference trajectory that it provides, in addition to the terrain data, directing the vehicle away from an enemy territory or an unexpected enemy vehicle detected in the area. An example of a path planner that could be integrated to feed a reference trajectory including both threat response and terrain following to the algorithm presented within this thesis is described and motivated in [39]. Additionally, given MPC's repeating nature, variable weighting matrices are possible. Therefore another avenue for future work could be the discussed reference trajectory input from a higher level path planner along with a corresponding TF/TA ratio to indicate the level of lateral tracking importance with respect to nearby threats, as compared to altitude tracking. This could also be the avenue by which a Ψ-reference could be appropriately weighted to focus the vehicle on a target as the target is passed or to reduce visibility by orienting exhaust away from an encountered threat. Overall, it should be kept in mind that survivability and mission success is directly dependent upon how well high level mission goals are reflected in the reference trajectory and cost function definitions.

REFERENCES

[1] Schroer, R., "UAVs: the future. [A century of powered flight: 1903:2003]," *IEEE Aerospace and Electronic Systems Magazine*, vol. 28, issue: 7, July, 2003, pp 61-63.

[2] French, M., "UAVs advance since Desert Storm," *Federal Computer Week*, Feb. 19, 2003, http://www.fcw.com/fcw/articles/2003/0217/web-uav-02-19-03.asp Accessed: Feb. 25, 2004.

[3] Joint Doctrine Division, J-7, Joint Staff, "Department of Defense Dictionary of Military Terms", as amended through December 2003, http://www.dtic.mil/doctrine/jel/doddict/, Accessed: Feb. 28, 2004.

[4] V. H. L. Cheng, T. Lam, "Automatic Guidance and Control Laws for Helicopter Obstacle Avoidance," *Proceedings of the 1992 International Conference on IEEE Robotics and Automation*, vol. 1, 12-14 May 1992, pp 252 – 260.

[5] R. A. Coppenbarger, V. H. L. Cheng, "Simulation Evaluation of a Pilot Interface with an Automated Rotorcraft Obstacle Avoidance System," *Second IEEE Conference on Control Applications*, vol. 2, 13-16 Sept. 1993, pp649-657.

[6] R. A. Coppenbarger, "Sensor-Based Automated Obstacle-Avoidance System for Nap-of-the-Earth Rotorcraft Missions," *Proceedings of SPIE Head-Mounted Displays*, vol. 2735, June 1996, pp 221 – 232.

[7] R. A. Hess, Y. C. Jung, "An Application of Generalized Predictive Control to Rotorcraft Terrain-Following Flight," *IEEE Trans. Systems, Man and Cybernetics*, vol. 19, no. 5, Sept.-Oct 1989, pp 955-962.

[8] D. Li, D. Zhou, Z. Hu, H. Hu, "Optimal Preview Control Applied to Terrain Following Flight," *Proceedings of the 40th IEEE Conference on Decision and Control*, vol. 1, 4-7 Dec. 2001, pp 211 -216.

[9] A. Bogdanov, E. Wan, M. Carlsson, Y. Zhang, R. Kieburtz, A. Baptista, "Model Predictive Neural Control of a High Fidelity Helicopter Model," AIAA Paper AIAA-2001-4164, Aug. 2001.

[10] L. Singh, J. Fuller, "Trajectory Generation for a UAV in Urban Terrain, using Nonlinear MPC," *Proceedings of the 2001 American Control Conference*, vol. 3, 25-27 June 2001, pp 2301 – 2308H.

[11] J. Kim, D.H. Shim, S. Sastry, "Nonlinear Model Predictive Tracking Control for Rotorcraft-based Unmanned Aerial Vehicles," *Proceedings of the 2002 American Control Conference*, vol. 5, 8-10 May 2002, pp 3576 – 3581.

[12] A. Bayoumy, J. Bordeneuve-Guibé, "A Neural Predictive Control Scheme for Nonlinear Plants," AIAA Paper AIAA 2002-1541, April 2002.

[13] V. Gavrilets, B. Mettler, E. Feron, "Nonlinear Model for a Small-Size Acrobatic Helicopter," AIAA Paper AIAA-2001-4333, Aug. 2001.

[14] C. Flood, *Real-time Trajectory Optimization for Terrain Following Based on Non-linear Model Predictive Control*, Linköping University document number LiTH-ISY-EX-3208, Jan 11, 2001.

[15] National Geospatial-Intellegence Agency (NGA), "Shuttle Radar Topography Mission (SRTM) Historic Mission," updated: Nov 13, 2002, http://www.nga.mil/portal/site/nga01/index.jsp?epi-content=GENERIC&itemID=ad586591e1b3af00VgnVCMServer23727a95RCRD&beanID=1629630080&viewID=Article, accessed March 31, 2004.

[16] National Geospatial-Intellegence Agency (NGA), "NIMA, Partners Unveil GeoSAR: A New Radar Mapping System to Model the Earth Beneath the Trees," updated June 8, 2001, http://www.nga.mil/portal/site/nga01/index.jsp?epi-content=GENERIC&itemID=a7586591e1b3af00VgnVCMServer23727a95RCRD&beanID=1629630080&viewID=Article, accessed March 31, 2004.

[17] J. Turner, C. Moscoso, "21st Century Terrain – Entering the Urban World," https://peoiewswebinfo.monmouth.army.mil/JPSD/RTV/Terrain.pdf, accessed May 04, 2004.

[18] National Geospatial-Intellegence Agency (NGA), "Geospatial Engine," http://geoengine.nima.mil/, accessed March 31, 2004.

[19] Defense Mapping Agency, *Military Specification for Digital Terrain Elevation Data (DTED)*, Department of Defense document number MIL-D-89020, May 5, 1993.

[20] R. Denton, J. Jones, P. Froeberg, "Demonstration of an Innovative Technique for Terrain Follow/Terrain Avoidance – The Dynapath Algorithm," *Proceedings of the IEEE 1985 National Aerospace and Electronics Conference*, 20-24 May, 1985, pp 522-529.

[21] J. Maciejowski, *Predictive Control with Constraints*, Prentice Hall, NY; 2002.

[22] D. Kirk, *Optimal Control Theory: An Introduction*, Prentice Hall, 1970.

[23] J. Burl, *Linear Optimal Control: H_2 and H_∞ Methods*, Addison-Wesley, 1998.

[24] H. Kwakernaak, R. Sivan, *Linear Optimal Control Systems*, Wiley Interscience, 1972.

[25] D. P. Bertsekas, *Nonlinear Programming*, Athena Scientific, 1995.

[26] W. F. Arnold III, A. J. Laub, "Generalized eigenproblem algorithms and software for algebraic Riccati equations," *Proceedings of IEEE*, 72, no. 12, 1984, pp 1746-1754.

[27] P. Gill, W. Murray and M. Saunders, *User's Guide for SQOPT 5.3: A Fortran Package for Large-Scale Linear and Quadratic Programming*, Technical Report SOL 86-1. Revised July 30, 1998.

[28] A. Alaniz, *Model Predictive Control with Application to Real-Time Hardware and a Guided Parafoil*, Massachusetts Institute of Technology; 2004.

[29] Stanford Business Software Inc., "Distribution of SOL/UCSD Optimization Software", updated: Feb 22, 2003, http://www.sbsi-sol-optimize.com, Accessed: Feb. 28, 2004.

[30] L. Andrews, *Special Functions for Engineers and Applied Mathematicians*, Macmillan Publishing Company, NY, 1985.

[31] D. Mayne, J. Rawlings, C. Rao, and P. Scokaert, "Constrained model predictive control: Stability and optimality," *Automatica*, vol. 36, 2000, pp. 789 – 814.

[31] W. F. Arnold III, A. J. Laub, "Generalized eigenproblem algorithms and software for algebraic Riccati equations," *Proceedings of IEEE*, 72, no. 12, 1984, pp 1746-1754.

[32] T. Lapp, L. Singh, "Model Predictive Control Based Trajectory Optimization for Nap-of-the-Earth (NOE) Flight Including Obstacle Avoidance," *Proceedings of the 2004 American Control Conference*, 30 June – 2 July 2004.

[33] M. Nervegna, *A Sonar-Based Mapping System For An Unmanned Undersea Vehicle*, Massachusetts Institute of Technology; 2002.

[34] A. Martin, *Model Predictive Control For Ascent Load Management of a Reusable Launch Vehicle*, Massachusetts Institute of Technology; 2002.

[35] A. Jadbabaie, J. Hauser, "On the Stability of Unconstrained Receding Horizon Control with a General Terminal Cost," *Proceedings of the 40th IEEE Conference on Decision and Control*, December 2001, pp 4826 – 4831.

[36] A. Jadbabaie, J. Yu, J. Hauser, "Unconstrained Receding-Horizon Control of Nonlinear Systems," *IEEE Transactions on Automatic Control*, Vol. 46, No. 5, May 2001, pp 776 – 783.

[37] D. Mayne, J. De Doná, G. Goodwin, "Improved Stabilizing Conditions for Model Predictive Control," *Proceedings of the 39th IEEE Conference on Decision and Control*, Sydney, Australia, December 2000, pp 172 – 177.

[38] M. McConley, B. Appleby, M. Dahleh, E. Feron, "A Computationally Efficient Lyapunov-Based Scheduling Procedure for Control of Nonlinear Systems with Stability Guarantees," *IEEE Transactions on Automatic Control*, Vol. 45, No. 1, January 2000, pp 33 – 49.

[39] R. Pettit, *Low-Altitude Threat Evasive Trajectory Generation for Autonomous Aerial Vehicles*, Massachusetts Institute of Technology; 2004.