Optimum Flight Trajectories for Terrain Collision Avoidance

by

Tapan Sharma
Bachelors of Aeronautical Engineering (Honours)

Submitted in fulfilment of the requirements
For a degree of Masters of Engineering

at

The Sir Lawrence Wackett Centre for Aerospace Design Technology
School of Aerospace, Mechanical and Manufacturing Engineering
Science, Engineering and Technology Portfolio

Royal Melbourne Institute of Technology

March 2006

Declaration

- I certify that except where due acknowledgement has been made; the work is that of the author alone.

- The work has not been submitted previously, in whole or part, to qualify for any other academic award.

- The content of the thesis is the result of work which has been carried out since the official commencement date of the approved research programme.

- Any editorial work, paid or unpaid, carried out by a third party is acknowledged.

Tapan Sharma

March 2006

Acknowledgments

- I would like to thank Associate Professor Cees Bil for his supervision and guidance for this thesis.

- I would like to thank my family members for their constant support throughout my master's candidature.

- I would also like to thank Dr Paul William Research Fellow RMIT University for his guidance.

- I would like to express my sincere gratitude to the staff and students at the Sir Lawrence Wackett Centre.

- I would also like to express my gratitude to Dipesh Parekh for his contribution in proof reading this thesis.

- Last but not least, I would like to thank Associate Professor Andrew Eberhard for his commitment in this thesis.

Table of Contents

List of Figures

List of Tables

Nomenclature

ρ	Air density
ψ	Heading angle
t_f	Minimum time
ν	Non dimensional speed
$\dot{W}_z$	Wind speed in z direction in body axis
ϕ	Bank angle
γ	Climb angle
$\dot{z}$	States in altitude
$\dot{x}$	States in x direction
α	Angle of attack
g	Gravity acceleration
$\dot{W}_y$	Wind speed in y direction in body axis
$n_{z(turn)}$	Load factor for turn
$u_k = u(t_k)$	Value of the control vector at a node point
$y_k = y(t_k)$	Value of the state vector at a node point
p	Additional parameters required for ***Direct***
hc	Clearance height
u_U	Controls upper limit
Z	Coordinate of terrain in z direction
M	Mayer, Mach number
$\bar{x}$	Non dimensional x distance
$\bar{y}$	Non dimensional y distance
$\bar{z}$	Non dimensional z distance
V_s	Speed of sound
x_L	States lower limit
$h_T[x(t), y(t)]$	Height of the terrain at the aircraft position
C_L	Lift coefficient

u_L	Controls lower limit
V	Speed
$\dot{y}$	States in y direction
D	Aerodynamic Drag
L	Aerodynamic Lift
X	Coordinate of terrain in x direction
Y	Coordinate of terrain in y direction
C_D	Drag coefficient
L	Integral
m	Mass
τ	Non-dimensional time
x_U	States upper limit
T	Thrust
ε	Thrust angle with respect to fixed body
η	Thrust settings
$\dot{W}_x$	Wind speed in x direction in body axis
S	Wing Area
$n_{z(C\lim b/descent)}$	Load factor for a climb
j	Cost function
T_{max}	Maximum thrust
u	Control vectors
W	Weight
x	State vectors

List of refereed papers

1. Sharma T., Bil C. and Eberhard A. “Optimal Flight Trajectories for Terrain Collision Avoidance” AIAC, Eleventh Australian International Aerospace Congress Melbourne Australia, March 2005

2. Sharma T., Bil C. and Eberhard A. “Control System for Optimal Flight Trajectories for Terrain Collision Avoidance” KES, Ninth International Conference on Knowledge-Base Intelligence Information & Engineering Systems, 14 – 16 September, 2005 Hilton, Melbourne, Australia

3. Sharma T., Williams P., Bil. C. and Eberhard A. “Optimal Three-Dimensional Aircraft Terrain Following and Collision Avoidance” The 7th Biennial Engineering Mathematics and Applications Conference, Melbourne Victoria, 25 – 28 September 2005

4. Sharma T., Bil C. and Eberhard A. “Optimized Three Dimensional Collision Avoidance Trajectories into Terrain” The 1st Malaysian Software Engineering Conference, Copthorne Orchid Hotel Penang, 12th to 13th December 2005

5. Sharma T., Bil C. and Eberhard A. “Optimum Flight Trajectories and Sensitivity Analysis for Terrain Collision Avoidance Systems” 25th International Congress of the Aeronautical Sciences, Hamburg Germany, September 2006

Summary

Ground Proximity Warning Systems (GPWS), Enhanced Ground Proximity Warning Sensors (EGPWS) and Terrain Awareness Systems (TAWS) have been developed to aid in the reduction of aircraft ground collisions. They are devices which provide pilots with an aural warning signal of proximity of terrain. These systems make use of a downward looking sensor which senses the proximity of oncoming terrain. Certainly these warning devices are beneficial if the pilot reacts to them but they do not assist in improving the situation awareness of the flight crew or what action to take to avoid a collision

The implementation of such systems has reduced aircraft accidents caused by Controlled Flight into Terrain (CFIT) however it has not been eliminated. Thus it is necessary for a new system to be developed, that would not only act like a warning, but would also be capable of assisting the pilot by providing him with safe escape trajectories in a situation which could eventuate into a CFIT accident. Pilots usually conduct a pull- up manoeuvre when in ground proximity to increase altitude. This is a logical response but in high mountainous terrain, this manoeuvre may still result in a collision. Furthermore, the sudden pull up manoeuvre could cause the aircraft to exceed its aerodynamics, structural and propulsion limitations. For example load factor.

Hence, the primary aim of this research is to develop a methodology utilizing the availability of a three dimensional digital terrain topology database and aircraft position to compute safe escape trajectories in both vertical and lateral directions. The aircraft model used was a Phantom F4. The objective of this thesis is to prove that flying around a terrain can provide the pilot a better chance of survival than by conducting the regular pull up manoeuvre in case there is not adequate time. To add more value to this study, two more objective functions have been added, minimum time and minimum clearance from the terrain. In the former case, the aircraft has to clear the terrain in the least possible time whilst in the latter case; the aircraft has to clear the terrain by flying close to the terrain at a specified clearance. The two scenarios have been selected as military aircraft are most often involved in Terrain Avoidance (TA) and Terrain Following (TF) operations to prevent them from being exposed to enemy fire. However emphasis is given more on avoiding collision rather than planning a collision avoidance strategy.

The second part of this investigation involves a sensitivity analysis of instrument errors on the ability to fly an optimal escape trajectory. Instrumental errors are always present and should be considered in any flight simulation to determine how practical the methodology is. To investigate the extent of influence of instrumental errors, there is a need to conduct a sensitivity analysis which is presented in chapter 6. The sensitivity study involves consideration of various scenarios in which the aircraft is required to fly an optimal flight trajectory out of collision. The principal reason for such analysis is to determine the sensitivity of the optimal escape trajectory solution subjected to instrument errors.

Snopt [1] and ***Direct*** [2] software were used extensively in ***Matlab*** [3] environment version 7 for all the analytical work conducted for this thesis. The three dimensional terrains were initially generated via using functions such as cylinders, cones, etc. Subsequently the more complicated shape of the terrains were generated in ***Terrain Generator*** [4] which was exported and converted to ***Direct*** format file using B-Splines function in Matlab. Further details pertaining to generation of results are provided in the Chapter 3.

The results obtained in this thesis show that generation of safe aircraft trajectories in a three dimensional digital terrain topology are possible. Although the equations of motion were based on three degrees of freedom, there were limitations added on the dynamics of the aircraft to make it realistic. The ability to use different terrains for modelling also proves that the method is versatile. Finally investigation of the sensitivity analysis shows the ability to counter act the errors in navigational instruments of the aircraft.

1 Introduction

According to recent National Transportation Safety Bureau (NTSB) findings, Controlled Flight into Terrain (CFIT) has contributed to more than one third of aircraft accidents into terrains [5]. CFIT is a phenomenon in which an aircraft crashes into a terrain despite the effort by the pilot to salvage the situation as shown in Figure 1.1.

Figure 1.1: Example of a CFIT situation [5]

CFIT is classified as an accident into terrains with the knowledge of the pilot of the impending disaster. [5] This type of accident occurs because of three major reasons, but is derived from numerous contributory factors. These are as follows:

i. Situation awareness

- Lack of visibility (fog, darkness, clouds, etc)
- Pilot fatigue or inexperience

ii. Erroneous data

- Inaccurate maps
- Instrument errors
- Miscommunication with Air Traffic Controllers (ATC)

iii. Physical disabilities

- Impairment due to illness
- Vertigo

It is evident from the aforementioned that other than inaccurate maps and instrument errors, the cause of CFIT is not associated with aircraft performance limitations but are more related with human factors such as pilot's perception, attitude and performance.

Ground Proximity Warning System (GPWS) [8] is a warning system which can aid pilots to avoid a CFIT accident. A GWS alerts the pilot. A GPWS system alerts the pilot when one of the thresholds is exceeded between 50 and 2450 feet radio altitude. A brief summary of the capabilities of GPWS is to detect:

i. Excessive Decent Rate
ii. Excessive Terrain Closure Rate
iii. Altitude Loss after Take Off or Go Around
iv. Unsafe Terrain Clearance During High Speed Flight or While not in Landing Configuration
v. Below Glideslope Deviation Alert

The GPWS technology utilises a "look down approach", that is, to say it makes use of the airplane's radio altimeter to provide aural warnings of impending collision. Its usage however is limited to moderately steep terrains. In the case of a steep slope or vertical cliff, the system would not be able to detect the danger in time. Another shortcoming of the GPWS was its susceptivity false warnings [7].

On the other hand, Enhanced Ground Warning System (EGPWS) [9] is another type of warning system implemented in aircraft. The primary features of EGPWS are similar to that of GPWS; however one of the underlying differences was the introduction of a forward looking sensor and its ability to look at the aircraft's position in a three-dimensional space to terrain database stored in an onboard computer This sensor is capable of detecting both low lying and steep terrains. It is this capability of EGPWS which resulted in the reduction of false aural warnings. Figure 1.2 illustrates the decrease in the number of CFIT risk for large commercial jet aircraft which uses EGPWS.

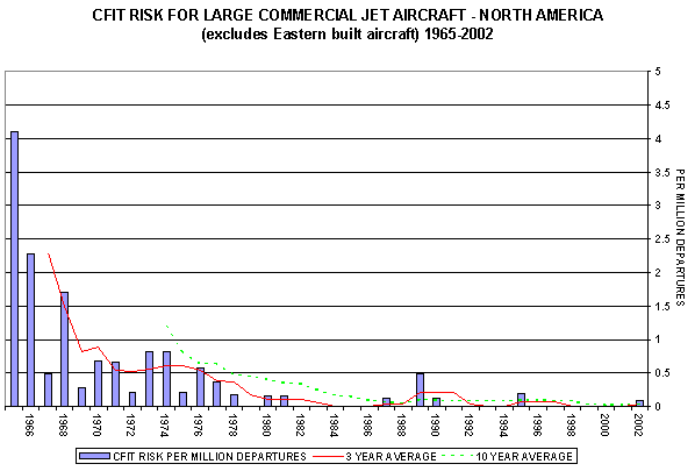


Figure 1.2: CFIT risk for large commercial jet aircrafts [6]

Although technology has advanced, numerous accidents recently require that such incidents are prevented in the first instance. Figure 1.3 depicts the number of accidents caused by CFIT for the period of 1992 – 2002.

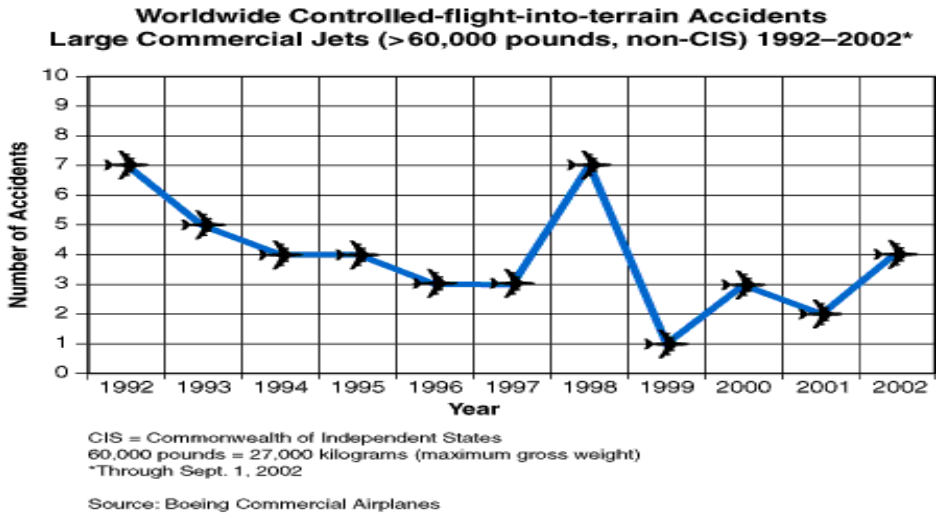


Figure 1.3: World wide CFIT accidents for the period of 1992 – 2002 [6]

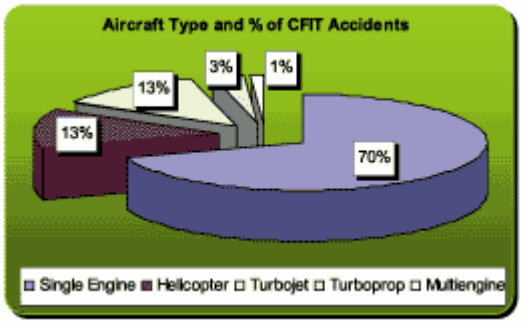


Figure 1.4: Aircraft Type and % of CFIT accidents [5]

Figure 1.4 illustrates the percentage of the types of aircraft that have been involved in CFIT accidents. Modern aircraft implements Global Positioning Systems (GPS), which are capable of determining exact location of the aircraft and updating location data instantaneously. Figure 1.5 depicts the phases of flight for which CFIT occurs. Most CFIT accidents occur during the approach and low altitude manoeuvre and cruise phases. This is when the pilots' skills are really put to the test.

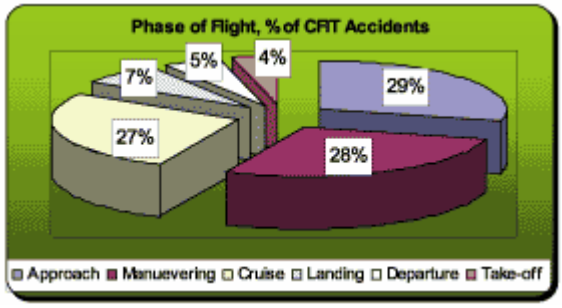


Figure 1.5: Phase of Flight and % of CFIT Accidents [5]

Military aircraft are often required to fly close to terrain to avoid radar detection however; this is not easily achieved due to the requirement of quick reflexes in control from the pilot and the need for the aircraft to function close to its operational constraints. Not all pilots have the sheer confidence to fly as close to the terrain as possible. It has been noticed that, pilots adopt a similar response when confronted by mountainous terrain. That is, they pitch up in order to clear the oncoming terrain. Unfortunately, there have been several accidents where pilots detected danger too late and were unable to pitch up in time and crashed into the oncoming terrain. The reasons for the crashes are mainly due to the structural, propulsive and aerodynamic limitations of the aircraft.

To prevent such accidents from happening, an alternative system for collision avoidance is required. The alternative solution will require the pilot to fly around the terrain rather than pulling up on the control stick to climb over the terrain. Of course, the pull up manoeuvre is best suited when the terrain level is low or if the terrain is wide but if it is otherwise, performing a lateral manoeuvre to flying around the terrain may give the pilot an increased chance of survival.

1.1 Problem Formulation

A good understanding of terrain avoidance systems needs to be attained. Most aircraft have terrain avoidance systems installed so that they can be alerted of the oncoming dangers and take the appropriate measures. Unfortunately, there have been instances where detection of danger was delayed resulting in a collision into terrain. There have been incidents where the pilot is alerted of the impending disaster, however due to shock, loss of consciousness or the lack of situational awareness, the aircraft was flown right into the oncoming terrain. This research undertaken is to determine a feasible solution for these types of accidents.

A safe and evasive method needs to be developed whereby the pilot can be presented with safe escape trajectories. This can be taken further by implementing an automated control system that could take over if the pilot is does not have the confidence to fly the optimised trajectory and once the aircraft has successfully avoided collision, the controls can be handed back to the pilot. The methodology can be also implemented on board Unmanned Aerial Vehicles (UAV) which often has to perform collision avoidance whilst on a mission. This project can also be expanded further to solve for a combat battle situation when an enemy aircraft is on pursuit. The escaping aircraft will be able to perform Terrain Following (TF) by keeping as close to the terrain as possible so as to avoid being detected whilst preventing an imminent collision. Additionally it will avoid being shot down by the enemy whilst performing a Terrain Avoidance (TA). Figure 1.6 depicts the usual type of escape trajectory conducted by the pilot whilst Figure 1.7 shows the new methodology proposed in this thesis to be implemented.

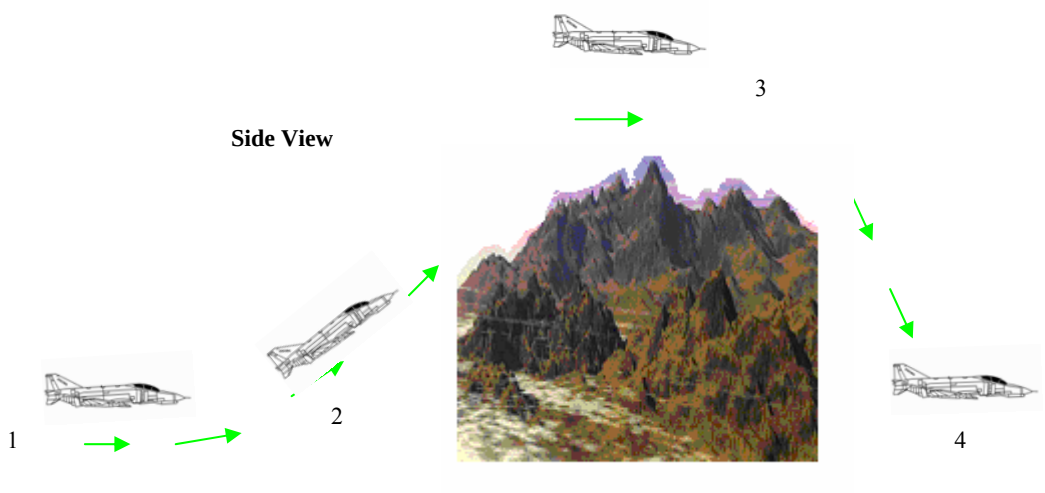


Figure 1.6: Example of a pull up manoeuvre

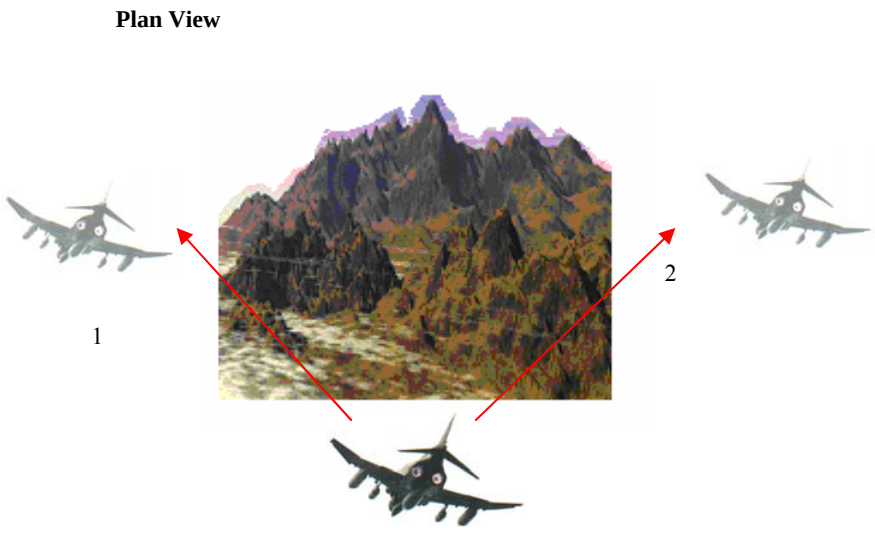


Figure 1.7: “Over or around? That’s the question”

1.2 Research objectives

The objective of this research is to introduce a new methodology that computes optimal escape trajectories within a three dimensional digital terrain. It has been established that conducting a pull up manoeuvre may not always be the best option for a pilot to salvage the aircraft in the case of an imminent collision. To prove this further, several simulations were conducted which shows that performing a lateral turn can produce safer trajectories in comparison to a vertical pull up. For the purpose of this investigation, several assumptions were made.

- The study is based on the availability of a three dimensional terrain database describing the terrain topology.

- The aircraft dynamics are based on point mass equations of motion. However, constraints are put on angle of attack, bank angle as well as rate of change of angle of attack and roll rate to account for physical limitation of the control devices and aircraft performance.

- The optimiser only looks at the down distance when generating a safe clearance from the terrain. This theory will be explained clearer in the collision avoidance chapter.

1.3 Outline

This thesis demonstrates that by considering equations of motion for a point mass aircraft model and terrain database knowledge, one can solve trajectory optimization problems. Various researchers have made significant progress in two dimensional collision avoidance systems. However the real challenge is to deal with three dimensional terrain scenarios. The reason for this is when an aircraft is subjected to CFIT; having knowledge of the terrain in a three dimensional domain would aid the aircraft to acquire more escape trajectories in the new lateral direction.

Chapter 2 of this thesis provides an elaborative discussion of the various collision avoidance systems that will be developed. Following this, an introduction on trajectory generation will be conducted. This chapter concludes with a discussion about terrain modelling. Optimal

control problems have dealt with several different methods. The various methods will be discussed in chapter 2.1

Chapter 3 explains the methodology utilised for the aircraft which includes terrain building, explanation of ***Snopt*** and ***Direct*** software tools, collision avoidance technique, discretization method and sensitivity analysis which involves introduction of errors in the aircraft instruments.

Chapter 4 presents the rationalization of using the six different terrain models and explains how the initial and final positions of the aircraft are set.

Chapter 5 exhibits the plots for collision avoidance and entails a discussion of all the results that have been obtained for a scenario of pull over or go around, minimum time and minimum clearance scenario via simulations

Chapter 6 entails results and discussion for sensitivity analysis with respect to each plot.

Lastly chapter 6 provides a holistic conclusion and discusses the future work which can be conducted in relation to this thesis. The control plots for all the simulations conducted are entailed in the appendix chapter.

2 Literature Review

2.1 Introduction

In this chapter, existing collision avoidance systems and methods are discussed. Furthermore a brief discussion on how trajectories are generated is provided. This core of this thesis is not interested in generating escape trajectories but rather utilising the already available methods. This chapter will conclude with a short section on different types of terrain modelling.

2.2 Existing Collision Avoidance Systems

In the past, a lot of work has been done in relation to trajectory optimisation using optimal control theory. Using cubic spline interpolation in Curved path approaches and dynamical interpolation [10] was one of the few of the methods that were employed to calculate and design set of intercept points that a trajectory was required to pass through. The system was computed in terms of spline functions in order to meet these requirements for dynamic interpolation. Adding constraints and ensuring that the aircraft followed the required path completed the optimisation method. To further enhance the procedure, a cubic spline was applied to the pseudo controls. This aforementioned method can be implemented to solve problems pertaining to practical dynamical interpolations in the future.

The GEC Avionics Ground and Obstacles Collision Avoidance Technique (GOCAT) can be used to elucidate what previous technologies had been used to reduce CFIT accidents and why a new system was required. The main advantage was that terrain database knowledge was utilised. The improvement this system provided was that it reduced the number of false warnings because of its realistic knowledge of the terrain and the obstacles around the aircraft [11]. The disadvantage is that the method specified was only practical for aircraft motion in a two dimensional scenario. Furthermore it was not so practical because the aircraft was only capable of holding terrain elevations up to 1300 by 1300 mile area.

Using digital database, Ground collision Avoidance System (GCAS) [12] in conjunction with Terrain Following (TF) method, pilots were able to execute high rate turns and evasive manoeuvres in the day as well as night conditions in fighter jets such as F-16S. This technology was heavily used by fighter jets because of (TF). The Advanced Fighter Technology Integration (AFTI) programme objective was to demonstrate the benefits for tests in All Terrain Ground Collision Avoidance System. The flowchart summarizes the events that take place in GCAS is shown in figure 2.1

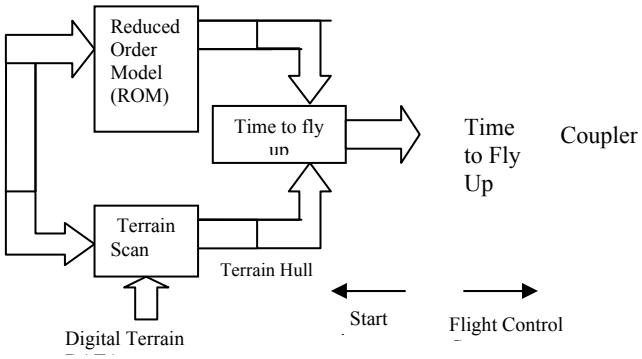


Figure 2.1: All Terrain GCAS Functional Diagram [5]

Traffic Alert and Collision Avoidance System II (TCAS II) was an avionic system that was required to be fitted on all aircrafts that had a payload of more than thirty passengers. The aim of TCAS II was to alert pilots of the traffic around them and to provide escape manoeuvres in a vertical direction [13]. Figure 2.2 and 2.3 exhibits examples of TCAS software where the aircraft is about to collide into a terrain and into an intruder aircraft respectively.

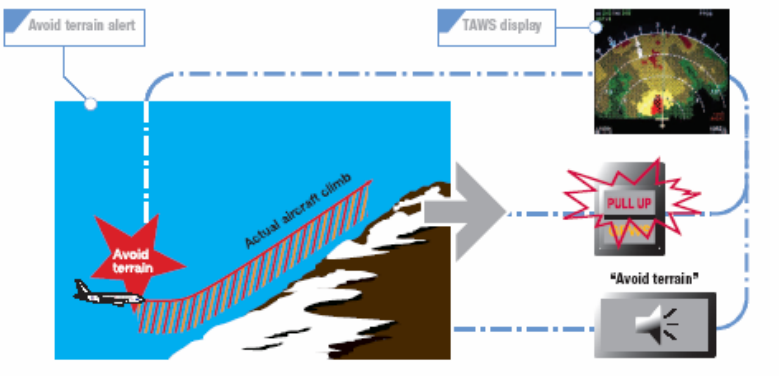


Figure 2.2: Terrain Collision Avoidance [13]

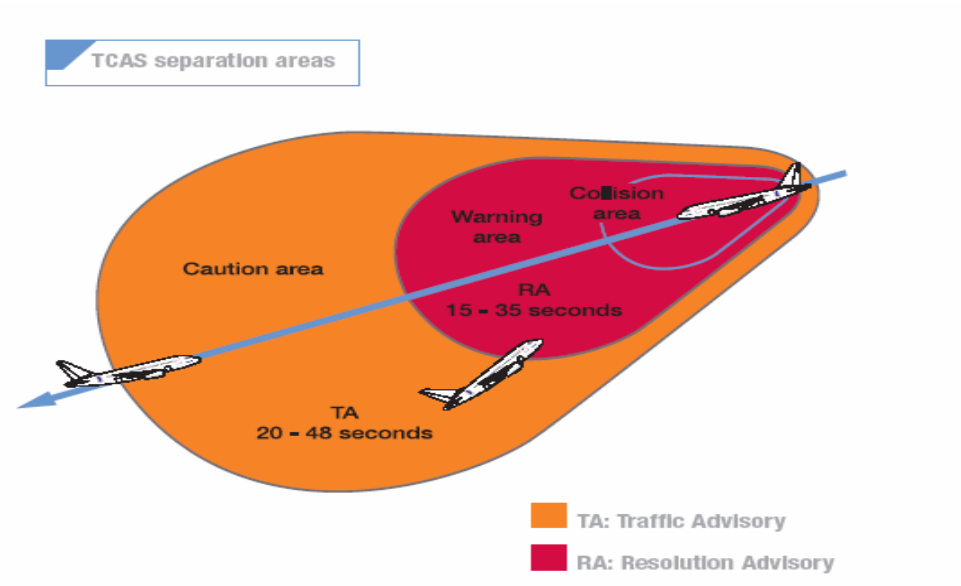


Figure 2.3: Intruder aircraft Avoidance [13]

To prevent collision during landing, two different paths were evaluated for use with a collision alerting system for independent closely spaced parallel approaches in instrument conditions. They were climb only and climb followed by turn manoeuvre from parallel traffic. Upon completing the project, it was discovered that the climb case, resulted in approximately 38 times more collisions than a climbing turn manoeuvre. The climbing turn case was less sensitive to pilots' reactions compared to the straight climb case. Finally looking at the graphs that were plotted in [14], the climb only manoeuvre resulted in a system that was more likely to have a collision.

Another example of modelling free flight has been provided in [15]. The behaviour of the aircraft was modelled as an autonomous hybrid automata and the reachability analysis was provided by the tool Hytech to search for optimal trajectories. The theory involved each aircraft being surrounded by two imaginary cylindrical spaces in the shape of hockey pucks. No aircraft were permitted to enter the protected zone. The large circle shape formed the alert zone. This is illustrated in figure 2.4. In this paper, a decentralised approach was implemented where each aircraft was allowed to optimize its own trajectory.

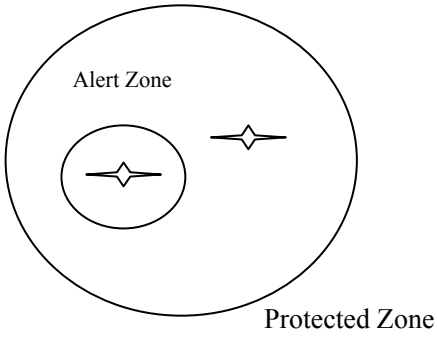


Figure 2.4: Alert zone and protected zone surrounding an aircraft. [15]

To reduce the complexity of the problem, the movement of the aircraft was limited to two dimensional planes. The cost function was to minimise time. Therefore we the two aircrafts were considered to start at the same time, the cost that had to be minimised was the sum of the time that each aircraft was needed to reach its destination shown in equation (1).

$$J = \int_0^{T_1} dt + \int_0^{T_2} dt = T_1 + T_2 \tag{1}$$

The scenario involved two aircrafts. It was discovered that the best option was the roundabout strategy whereby the aircrafts avoided collision by the example of cars at a round about.

Another project, which utilised Auto Ground Collision Avoidance System (Auto GCAS), was tested for fighter planes such as F-16 and JAS39. It used a similar technology to GCAS [16], the only difference being that it was automated. The aircraft's recovery was automatically performed whenever the trajectory had penetrated a distance from the terrain profile. The advantages of such system were that false warnings were minimised to almost zero. This is illustrated in Figure 2.5.

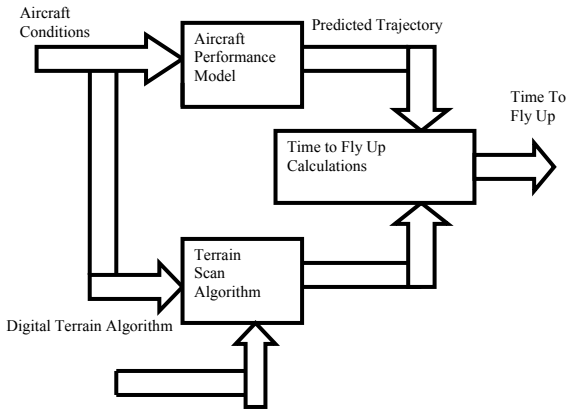


Figure 2.5: Automatic Ground Collision Avoidance Algorithm Architecture [16]

To enable the system to operate in a free flight environment, emphasis was placed on a flight path which permitted the aircraft to fly under Instrument Flight Rules (IFR). To prevent CFIT accidents, the terrain agent processed the terrain elevation and flight path data. The methodology consisted of a combination of Knowledge Based Expert System (KBES) and optimal control algorithms [17]. The KBES was responsible for decision making and determining appropriate actions that were required to detect and avoid conflicts. The optimisation method used was fixed size 'Breadth First Search'. The optimal control utilised the data provided by KBES and generated optimum avoidance trajectories. The results of the project revealed that, although the simulation results were not fully optimized, the combined method provided a feasible technique to obtain good avoidance trajectories. Another positive of this method was that the combination theory of the two modules provided a rapid prototyping methodology for developing traffic avoidance agents.

James K.Kuchar [18] developed a statistical model of terrain that estimated the probability of CFIT accidents after being alerted by a GPWS. The method used was by deriving a terrain model from an actual terrain database which was used to create a Markov chain simulation. After running the simulations, the probability of terrain collision was computed as a function of terrain and aircraft trajectory profile. Contours of collision avoidance probabilities were then generated and plotted against current alerting thresholds. The results of the simulations showed that the probability of terrain collision was less that 10^{-8} for smooth terrains and was approximately 0.01 for steep terrains. Additionally the probability of a false alarm occurring deceased from 0.08 to 10^{-4} for steep terrains

Arthur Richards and Johnathan P How [19] introduced a method for finding optimal collision free trajectories for multiple aircraft. The methodology utilised a programme called Mixed Integer Linear Programming (MILP). In this paper, only linear constraints were developed, enabling MILP approach to be applied to aircraft collision avoidance. The results obtained were based on a two dimensional analysis. Future use of this methodology involved the extension of the formula to include multiple waypoints and path-planning whereby each vehicle was required to adhere to a set of points in an order chosen within the optimisation scheme.

Another collision avoidance technique which was introduced [21] related to T-CAS II. It provided the pilots with the necessary advisory aid when confronted by danger. The principal of this method was to design an autopilot function for the realisation of escape manoeuvres for civil transport aircrafts via the introduction of a supervision layer to employ efficient predictive control techniques in fast paced situations.

Radar Assisted Collision Avoidance and Guidance Strategy RACAGS) [22] is another method to avoid collision. The system utilises active sensors to provide a range of information about obstacles ahead of the vehicle during low altitude planar flight. The onboard guidance subsystem generates both guidance and avoidance commands based on the information received from the radar. Thus, the information obtained aided in collision avoidance. The avoidance command is derived from figure 2.6. The position of the vehicle is shown as p(x,y). To avoid collision, the vehicle can take either Circular path 1 or 2.

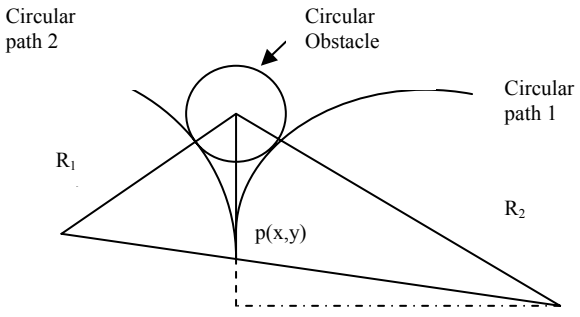


Figure 2.6: Two possible collision-free paths

A unified framework of collision detection and avoidance is discussed in [23]. This is done by using computer graphics and animations. The basis of this collision analysis framework was the vector fields in which data from natural or physically based phenomena were gathered into sets and generated from mathematical model, was collectively represented from mathematical models. The framework used Egbert's technique [24] for collision avoidance and collision detection.

2.3 Trajectory Generation

In order to combat terrain collisions, it is imperative that a survey on trajectory generation as well as trajectory optimisation is conducted. The Energy State Approximation method presented in [25] is one of the various methods which used a point mass system to generate an optimised trajectory for a supersonic aircraft. In this paper, the energy state approximation was extended to minimum time to climb and maximum-range problems. Point mass approximation had been chosen because of the subsonic aircraft. The state variables used were V, h, m and γ, and the control variable was α. The energy state equation used is shown in equation 2.

$$E = 1/2V^2 + gh \tag{2}$$

An attempt to solve a TF optimisation problem for a point mass system is shown in [26], which was solved by inverse dynamic approach. In this paper, point mass equations 3 to 6 were used.

$$\dot{y} = v\sin\gamma \tag{3}$$

$$\dot{x} = v\cos\gamma \tag{4}$$

$$\dot{v} = \frac{T\cos\alpha - D}{m} - g\sin\gamma \tag{5}$$

$$\dot{\gamma} = \frac{T\sin\alpha + L}{mv} - \frac{g}{v}\cos\gamma \tag{6}$$

Where the state variables are the position coordinates x and y, the velocity v and the flight path angle γ. L and D are the aerodynamic lift and drag respectively:

$$L = qSC_L \tag{7}$$

$$D = qSC_L \tag{8}$$

Where S is the reference area of the aircraft, and $q = pv^2/2$ is the dynamic pressure with the atmosphere density denoted by p which is assumed to be an exponential function of the altitude y with the scale height equal to 23,800 ft. The lift and drag coefficients, C_L and C_D are functions of angle of attack α and Mach number M. The thrust is defined by:

$$T = T_{\max}(m, y)\eta \tag{9}$$

Where

$$0 \le \eta \le 1. \tag{10}$$

The maximum thrust $T_{\max}$ is a function of Mach number M and altitude y in general. The value of η determines the required thrust level, and thus is called η the throttle setting. The aerodynamic and propulsion controls are represented by α and η. The objective functions was a combination of minimum time and closeness of the terrain in a two dimensional terrain. In this method, the inverse dynamic was well suited for the trajectory optimisation purpose because had advantages of robustness and good conditioning.

A method for real time trajectory generation is presented in [27] where Non-Linear Model Predictive Control (MPC) is utilised for generating escape trajectories. For this research, an effort to solve a collision avoidance problem is undertaken on both point mass and six degrees freedoms for aircraft models is made for a two dimensional terrain profile. In [28], Hermite Simpson discretization method is used to obtain an optimal trajectory for the Saab J35 Draken aircraft. The equations of motion used are of point mass system. The equations of motion for the performance model was summarised as:

$$\dot{x} = f(x, u) \tag{11}$$

Where the vector of state variables is

$$x = (V, h, X_E, m_f, \gamma)^T \tag{12}$$

And the control vector is

$$u = (\alpha, \delta_T)^T \tag{13}$$

Additional requirements are implemented as algebraic constraints in the form

$$\underline{g} \le g(x,u) \le \bar{g} \tag{14}$$

Where $\underline{g}$ and $\bar{g}$ are the upper and lower bounds on the algebraic constraints.

The cost function for this investigation was

$$X_E(t_F) = \int_{t=0}^{t_F} V(t)\cos\gamma(t)dt \tag{15}$$

Another research paper is shown in [29], whereby an optimal trajectory is generated for a minimum fuel turn for a three dimensional case scenario using Hermite Simpson discretization method. In this study, the two dimension point mass equations are converted to a three dimensional equations of motion and the cost functions used were:

$$X_E(t_F) = \int_{t=0}^{t_F} V(t)\cos\gamma(t)\cos\psi(t)dt \tag{16}$$

$$Y_E(t_F) = \int_{t=0}^{t_F} V(t)\cos\gamma(t)\sin\psi(t)dt \tag{17}$$

A new way of generating a trajectory in a least amount of time is presented in [30] utilising a simplified nonlinear longitudinal helicopter model which used minimum time as the objective function. Another method is presented in [31] which utilises a new method based on a new data structure, “framed octree”, by computing distance transformation utilising a spherical path planning wave. The proposed method uses the combination of accuracy of three

dimensional grid based path planing with the efficiency of octree based techniques. A receding horizon optimal control was used for autonomous trajectory generation and flight control for an unmanned aerial vehicle in urban terrain for non real time scenario utilising MPC is presented in [32]. A modified simple shooting method for generating of trajectory is presented in [33] whereby combination of theoretical result of Pontryagin's Minimum Principal. The methodology is described in Equation (18) to (21). The method used is presented in [34]

$$J = \int_{t_0}^{t_1} l(x,u)dt \tag{18}$$

Subject to

$$\dot{x} = f(x,u), \tag{19}$$

$$x(t_0) = x_0, x(t_f) = x_f \text{ fixed}, \tag{20}$$

$$u(.) \in \Omega(t). \tag{21}$$

Numerical solution by direct collocation is described in [35] to [36]. This method demonstrates that using an appropriate discretisation of control and state variables, a constrained optimal control problem can be solved by using Sequential quadratic methods.

Terrain modelling

Terrain modelling has become a very important factor for training purposes or real combat scenarios where knowledge of the opponent's territory is of importance. An example of a terrain model can be found in [37], uses a Digital Terrain System (DTS) consisting of a stored digital map of the terrain elevation data with the measurements of the aircraft's dynamics and height above the ground to provide navigation and terrain referenced cues. The functions of DTS are:

- Terrain Referenced Navigation (TRN)
- Terrain Following (TF)
- Predictive Ground Collision Avoidance System (PGCAS)
- Obstructive Warning and Cueing
- Ranging

Another method for generating terrain model is presented in [4]. The terrains are generated via software ***Terrain Generator***. The features of the software are:

- Full 3D control over the appearance of the terrain.
- Fractal terrain generation via eleven unique configurable algorithms.
- Height map terrain generation.
- Multi level Undo, Redo and Revert.
- Four render modes: Textured, Solid, Wire frame, and Points.
- Several different styles of light maps.
- Lots of additional display options.
- Save your terrain as a .tgm file for latter editing.
- Exports to .vmf, .map, .rmf, .t3d, .dxf, .txt, .bmp and .jpg file formats.
- Ability to export hint brushes along with your terrain for polygon reduction in Quake engine based games.
- Texture browser.
- Simple vertex locking.
- Recent .tgm file menu.
- Quick and easy setup.

More explanation on ***Terrain Generator*** is presented in Chapter 3.

2.4 Summary

In this chapter, different collision avoidance methodologies were discussed. From this review of literature it appears that little has been conducted in avoiding terrain collisions in a three dimensional environment. Therefore this thesis will primarily investigate the terrain collision problem in a three dimensional environment. Generation of different trajectories have been discussed and after evaluation, the Direct Transcription method has been chosen for the investigation purpose pertaining to reasons for the selection of this method is given in chapter 3.3. For the terrain simulation, ***Terrain Generator*** was chosen because of the simple usage and outcome of good results.

3 Methodology

3.1 Introduction

This chapter discusses the main difference between a rigid body and point mass systems and provides the reasons for the implementation of the latter system for this research. The equations of motion are derived for the McDonnell Douglas F4 Phantom aircraft. Although the equations of motion are derived for a specific aircraft, this optimisation method is applicable to any aircraft model. Following that, ***Snopt*** and ***Direct*** software are introduced. A discussion on the avoidance technology is also provided. This is followed by a discussion of the various types of terrain model representations. Finally a discussion pertaining to sensitivity analysis follows.

3.2 Difference in rigid body and point mass

A point mass system has zero volume. It is often easier to treat bodies as a point masses especially when the dimensions of the bodies are much less than the distances under consideration. A point mass has a body with no extent, which implies that there are no moments acting on the body. A point mass also possesses negligible moments of inertia. Finally a point mass can be represented by three degrees of freedom and equation governing its motion can be achieved by translation only

On the other hand, a rigid body has weight and it has moment of inertia. The main difference between the two systems is that a rigid body furthermore is represented by six degree of freedom. A rigid body also has moments acting on it. Finally the equations governing the motion of the body requires by translation and moments.

For this thesis, a point mass system has been adopted. The rationale for using a simple point mass representation is that the required body rotations can take place in a much smaller time than required for the whole “escape trajectory”. One other reason is that McDonnell Douglas F4 Phantom aircraft, has to fly at high speeds and is highly agile with respect to attitude change, therefore point mass assumptions is necessary.

3.3 Aircraft model

For the purpose of flight-path control design, it is sufficient to treat only the translational motion of the aircraft. The aircraft chosen for simulation purposes is a F4 Phantom shown in Figure 3.1.

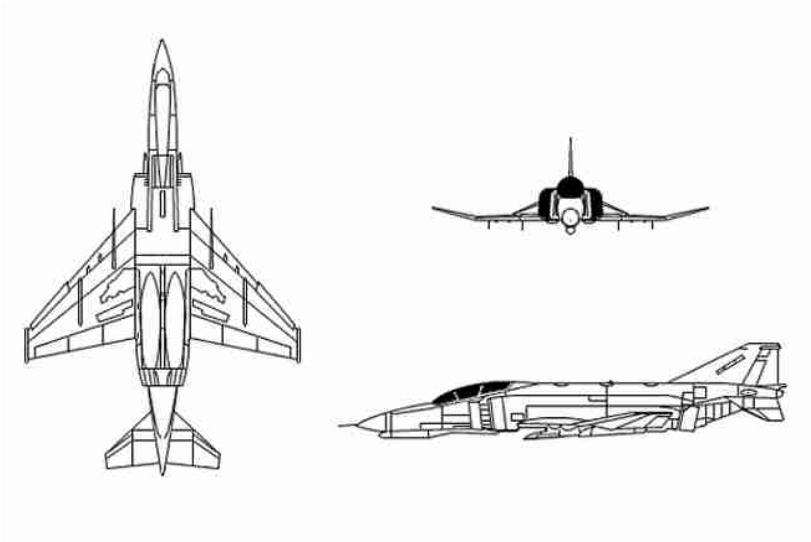


Figure 3.1: F4 Phantom Aircraft [25]

Table 3.1: Specifications of F4 Phantom [25]

MacDonnell – Douglas F4 Phantom Aircraft	**Imperial units**	**SI units**
Length:	62' 10”	18.90 m
Height:	16' 6"	5.03 m
Wing Span:	38' 5"	11.71 m
Wing Area:	530.00 sq ft	49.23 sq m
Empty Weight:	28276.0 lbs	12823.0 kg
Gross Weight:	50341.0 lbs	22830.0 kg
Max Weight:	58000.0lbs	26303.0 kg
Thrust (without afterburner)(each):	10900 lbs	4944 kg
Range:	1375 miles	2214.00 km
Cruise Speed:	587.00 mph	945.00 km/h
Max Speed:	1459.00 mph	2349.00 km/h
Climb:	48300.0 ft/min	14721.1 m/min
Ceiling:	59400.0 ft	18104.0 m

Firstly, the equations of motion are expressed in a velocity coordinate frame attached to the aircraft. The kinematic equations of motion of the aircraft are given by:

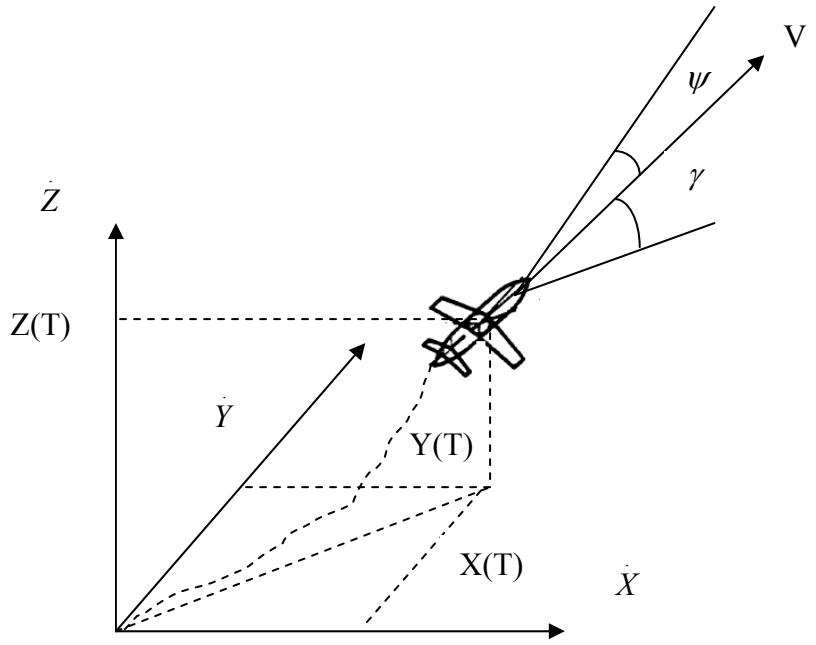


Figure 3.2: A typical flight path trajectory for an aircraft

$$\dot{X} = V\cos\gamma\cos\psi + \dot{W}_x \tag{22}$$

$$\dot{Y} = V\cos\gamma\sin\psi + \dot{W}_y \tag{23}$$

$$\dot{Z} = V\sin\gamma + \dot{W}_z \tag{24}$$

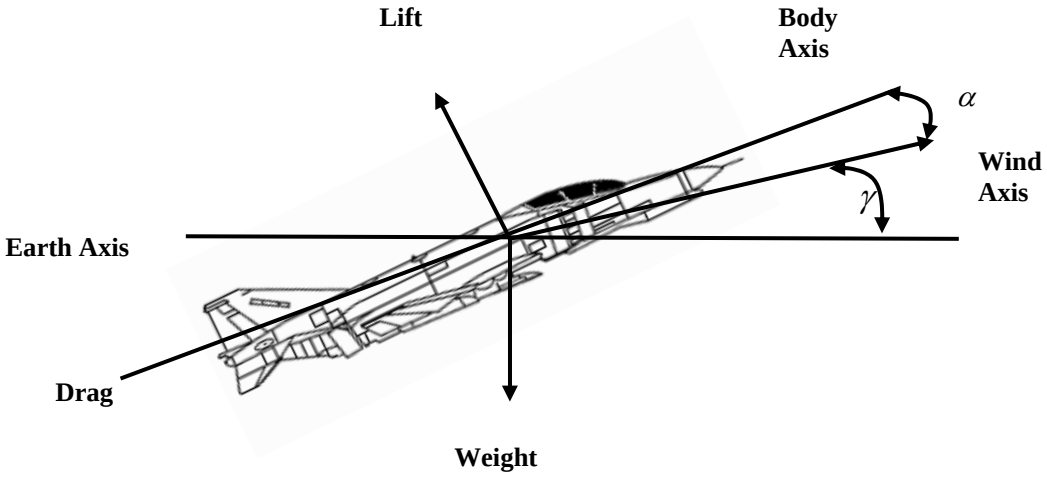


Figure 3.3: Point mass model of aircraft for longitudinal equations of motion [43]

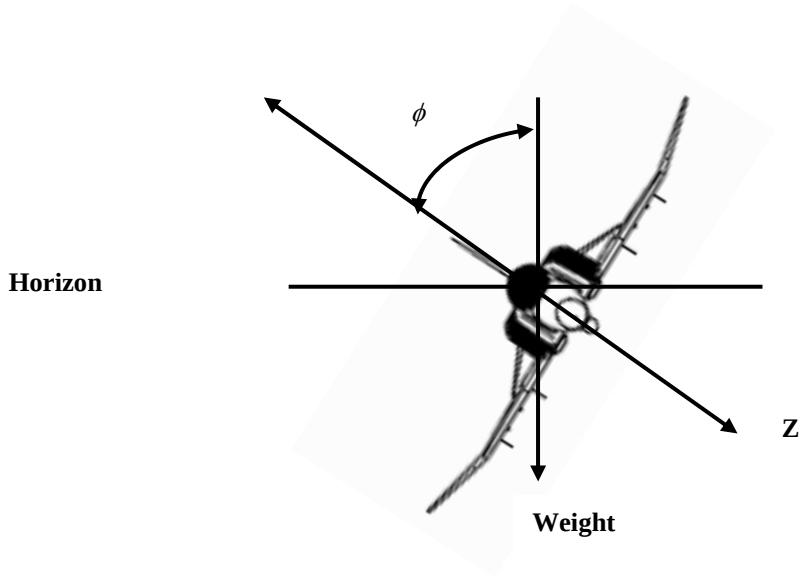


Figure 3.4: Point mass model lateral equations of motion [43]

Longitudinal Forces

$$\rho = 1.225*(1-Z/44.441/1000)^{4.256} \quad (25)$$

$$\text{Lift} = 1/2\rho V^2 S C_L \quad (26)$$

$$\text{Drag} = 1/2\rho V^2 S C_D \quad (27)$$

$$\dot{V} = \frac{T\cos(\alpha+\varepsilon)-D}{m} - g\sin\gamma - \dot{W}_x\cos\gamma\cos\psi - \dot{W}_y\cos\gamma\sin\psi - \dot{W}_z\sin\gamma \quad (28)$$

$$\dot{\gamma} = \frac{[L+T\sin(\alpha+\varepsilon)]\cos\phi}{mV} - \frac{g}{V}\cos\gamma + \dot{W}_x\frac{\sin\gamma\cos\psi}{V} + \dot{W}\frac{\sin\gamma\sin\psi}{V} - \dot{W}_z\frac{\cos\gamma}{V} \quad (29)$$

$$\dot{\psi} = \frac{[L+T\sin(\alpha+\varepsilon)]\sin\phi}{mV\cos\gamma} + \dot{W}_x\frac{\sin\psi}{V\cos\gamma} - \dot{W}_y\frac{\cos\psi}{V\cos\gamma} \quad (30)$$

To reduce the time taken for simulations and to avoid round off errors, non-dimensional equations were introduced. This was important because all the states possessed non uniform values. Therefore via introducing the non-dimensional equations, all the states are transformed to uniform values. The non-dimensional equations of motion were obtained by utilising Equations 31 to 35.

$$M = V / V_s \tag{31}$$

$$\tau = \frac{g}{V_s} t \tag{32}$$

$$\bar{x} = \frac{g}{V_s^2} x \tag{33}$$

$$\bar{y} = \frac{g}{V_s^2} y \tag{34}$$

$$\bar{z} = \frac{g}{V_s^2} z \tag{35}$$

The time derivatives are transformed by the following relationship:

$$\frac{d}{dt} = \frac{g}{V_s} \frac{d}{d\tau} \tag{36}$$

Where V_s is the speed of sound. Thus the left hand equations of motion are:

$$\dot{x} = \frac{g}{V_s} \frac{V_{s^2}}{g} = V_s \bar{\dot{x}} \tag{37}$$

$$\dot{y} = \frac{g}{V_s} \frac{V_{s^2}}{g} = V_s \bar{\dot{y}} \tag{38}$$

$$\dot{z} = \frac{g}{V_s} \frac{V_{s^2}}{g} = V_s \bar{\dot{z}} \tag{39}$$

$$\dot{v} = \frac{g}{V_s} V_s v' = g \bar{v}' \tag{40}$$

$$\dot{\gamma} = \frac{g}{V_s} \gamma' \tag{41}$$

$$\dot{\varphi} = \frac{g}{V_s}\psi' \tag{42}$$

This allows the equations of motion to be expressed in the following form:

$$\dot{X} = v\cos\gamma\cos\psi + W_x / V_s \tag{43}$$

$$\dot{Y} = v\cos\gamma\sin\psi + W_y / V_s \tag{44}$$

$$\dot{Z} = v\sin\gamma + W_z / V_s \tag{45}$$

$$v' = \frac{T\cos(\alpha+\varepsilon) - D}{mg} - \sin\gamma - \frac{\dot{W}_x}{g}\cos\gamma\cos\psi - \frac{\dot{W}_y}{g}\cos\gamma\sin\psi - \frac{\dot{W}_z}{g}\sin\gamma \tag{46}$$

$$\gamma' = \frac{\left[L + T\sin(\alpha+\varepsilon)\right]\cos\phi}{mgv} - \sin\gamma + \dot{W}_x\frac{\sin\gamma\cos\psi}{vg} + \dot{W}_y\frac{\sin\gamma\sin\psi}{vg} - \dot{W}_z\frac{\cos\gamma}{vg} \tag{47}$$

$$\psi' = \frac{\left[L + T\sin(\alpha+\varepsilon)\right]\sin\phi}{mgv\cos\gamma} + \dot{W}_x\frac{\sin\psi}{vg\cos\gamma} - \dot{W}_y\frac{\cos\psi}{vg\cos\gamma} \tag{48}$$

The lift and drag coefficients for $M <$ 1.15 are given by [25]. The reason for using the equations below is because the aerodynamic data is accurate up to Mach number less than 1.15 and in the simulation the speed of the aircraft did not exceed Mach 1.15. In this study, post stall conditions are not taken into consideration because the study is at its initial stage and the main concern is to find out if performing a lateral manoeuvre does make a difference. The lift and drag coefficients are given as:

$$C_L = C_{L_\alpha}\alpha \tag{49}$$

$$C_{L_\alpha} = 3.44 + 1/\cosh^2\left(\frac{M-1}{0.06}\right) \tag{50}$$

$$C_D = 0.013 + 0.0144\left[1 + \tanh\left(\frac{M-0.98}{0.06}\right)\right] + \left(0.54 + 0.15 + \left[1 + \tanh\left(\frac{M-0.9}{0.06}\right)\right]\right)C_{L_\alpha}\alpha^2 \tag{51}$$

Where $C_{L\alpha}$ and C_D are given in tabulated data as functions of Mach number M and are fitted by least square polynomials.

Controls for aircraft

The controls for the aircraft are:

(a) The maximum thrust T_{max} is expressed in units of 1000 lbs and is a function of Mach number M and altitude h in (units of 10 000 feet): Afterburners for this aircraft are also not taken into calculations to make calculations are simple as possible at this stage.

$$T_{max} = \begin{bmatrix} 1 \\ M \\ M^2 \\ M^3 \\ M^4 \end{bmatrix}^T \begin{bmatrix} 30.21 & -0.668 & -6.877 & 1.951 & -0.1521 \\ -33.80 & 3.347 & 18.13 & -5.865 & 0.4757 \\ 100.80 & -77.56 & 5.441 & 2.864 & -0.3355 \\ -78.99 & 101.40 & -30.28 & 3.236 & -0.1089 \\ 18.74 & -31.60 & 12.04 & -1.785 & 0.09417 \end{bmatrix} \begin{bmatrix} 1 \\ h \\ h^2 \\ h^3 \\ h^4 \end{bmatrix}$$

$$\text{Thrust} = \eta * T_{max} \qquad \text{with } 0 \le \eta \le 1 \tag{52}$$

The maximum thrust T_{max} of the aircraft is given in tabular form as a function of altitude and Mach number in [25]. Linear interpolations are used here for the table look-up evaluations of T_{max}. The reason for the choice for the above thrust model was to simulate as realistically as possible.

The use of the angle of attack and bank angle as pseudo controls can in theory lead to instant and rapid changes in the controls, which would otherwise be impossible to achieve in practice. Rapid changes would cause an aircraft to deviate from the trajectory and may prove to be fatal if it occurs in the case of limited clearance from the terrain. To account for the limited rigid-body reorientation of the aircraft, the rate of angle of attack and bank angle are considered as following in Equation 53 and 54.

$$-8 \le \dot{\alpha} \le 8 \qquad \text{degrees/second} \tag{53}$$

$$-15 \le \dot{\phi} \le 15 \qquad \text{degrees/second} \tag{54}$$

For control design, the rate of angle of attack and rate of bank are treated as new control variables, whilst the constraints on the actual angle of attack and bank angle become the first order state constraints in the following equations:.

$$-20 \le \alpha \le 20 \qquad \text{degrees} \tag{55}$$

$-70 \leq \phi \leq 70$ degrees (56)

The values in Equation 55 to 56 are estimated values and have been chosen due to the lack of realistic aerodynamic data for F4 Phantom.

Load factor limitations

The final constraint which was implemented was the load factor. This was implemented to ensure that the aircraft would operate within its structural limitations. It was important to consider and implement this factor so that the simulations would represent a realistic scenario. Otherwise the aircraft would perform a 90 degrees turn for example in one second which would not be possible in a realistic scenario. The load factor for both the vertical and lateral manoeuvre is defined in Equation 57 and 58. The load factors have been described as $n_{z(climb)}$ and $n_{z(turn)}$ individually but they have both been used together to simulate a climbing turn and descending turn scenario.

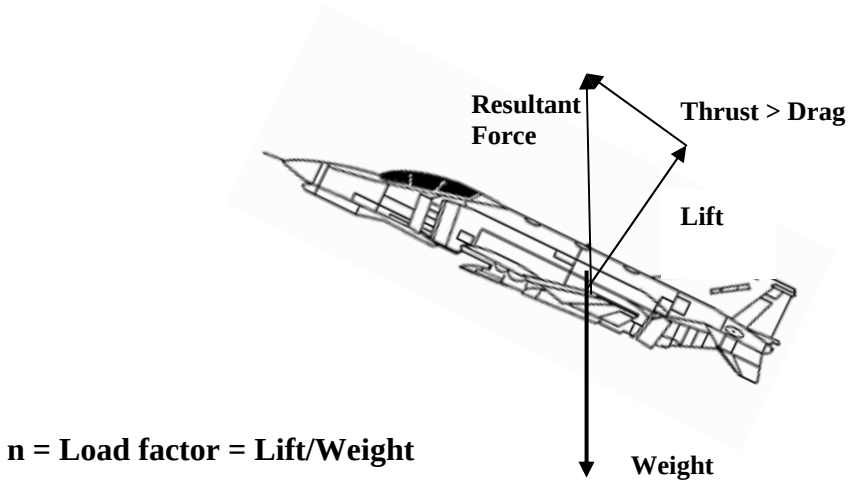


Figure 3.5: Load factor for longitudinal manoeuvre [43]

$0 \leq n_z \text{ (climb)} \leq 9$ (57)

n = Load Factor = 1/cos (ϕ)

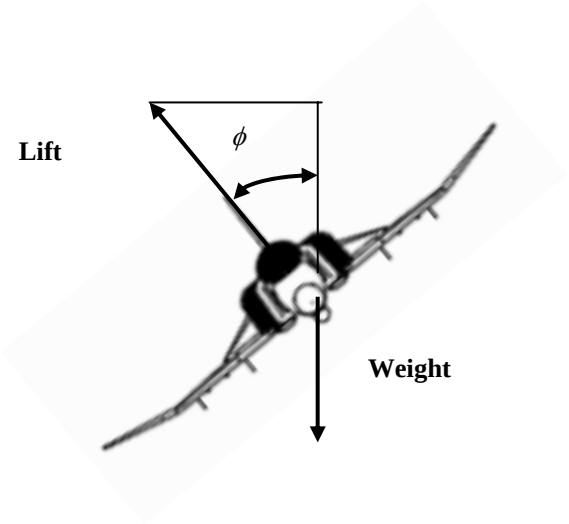


Figure 3.6: Load factor for lateral manoeuvre [44]

$$0 \leq n_{z(turn)} \leq 9 \tag{58}$$

Assumptions stated for simulation purposes

- Constant velocity will be maintained for simulation purposes
- Change in the mass fuel rate has been disregarded
- Point mass equations will be utilised
- Non rotating earth has been assumed
- A safety distance of 50 metres is implemented to take into account obstacles such as buildings and trees.

After the assumptions have been stated, next in line was to introduce the various softwares that were used for the research.

3.4 *Snopt* software tool

Snopt is a general-purpose system for solving optimisation problems involving many variables and constraints. It minimises a linear or non-linear function to bind on the variables and sparse linear or non-linear constraints. It is suitable for large-scale linear and Quadratic Programming (QP), for linear constrained optimisation and for general non-linear programs.

Snopt finds solution that are locally optimal and ideally the non-linear functions should be smooth and users should define the gradients.

Snopt uses a Sequential Quadratic Programming (SQP) algorithm that obtains a search direction from a sequence of quadratic programming sub problems. Each QP sub problem minimises a quadratic model of a certain Lagrangian function subject to linearised constraints. An augmented Lagrangian merit function is reduced along each search direction to ensure convergence from any starting point.

Snopt is most effective when some of the variables are entered nonlinearly or if the number of active constraints (including simple bounds) is nearly as large as the number of variables. ***Snopt*** requires relatively few evaluations of the problem functions.

The wrapper ***Snopt*** accepts a format that allows the constraints and variables to be defined in any order, irrespective of whether or not they occur nonlinearly in the objective or constraints. ***Snopt*** is designed to solve an optimization problem in the form

$$\text{(NPA)} \qquad \underset{x}{\text{minimize}} \qquad F_{obj}(x)$$

$$\text{Subject to } l_x \le x \le u_x, \qquad l_F \le F(x) \le u_F$$

Where l and u are constant lower and upper bounds, F(x) is a vector of smooth linear and nonlinear constraint functions $\{F_i(x)\}$ and $\{F_{obj}(x)\}$ is the component of F to be minimised. (The optional parameter maximize may be used to specify a problem whereby $F_{obj}(x)$ is maximised instead of minimised.) The ***Snopt*** interface records the variables and constraints so that the problem is in the form (sparseNP).

Ideally the first derivatives (gradients) of F_i should be known and coded by the user. If few gradients are known, ***Snopt*** will estimate the other missing gradients utilising finite differences.

Note that the upper and lower bounds are specified for all the variables and functions. This form allows full generality in specifying various types of constraint. Special values are used to indicate the absent bounds ($l_j = -\infty$ or $u_j = +\infty$ for appropriate j). Free variables and free constraints ("free rows") possess infinite bounds. Fixed variable and equality constraints have $l_j = u_j$.

In general, the components of F are structured in the sense that they are formed from sums of linear and non-linear functions.

Inform reports the results of the call to ***Snopt***

0 Optimal Solution found, i.e. the primal and dual infeasibility are negligible.
1 The problem is infeasible.
2 The problem is unbounded (or badly scaled).
3 Excessive iteration.
4 Feasible solution but the requested accuracy for the dual infeasibility could not be achieved.
5 The Super basics limit is too small.
6 The user has requested termination.
9 The current point cannot be improved.

3.5 *Direct* software tool

The fundamental premise of ***Direct*** is that the original continuous problem can be transcribed into a finite dimensional nonlinear programming (NLP) problem.
Direct requires the following files to be specified.

a) Cost function
b) Boundary Conditions
c) Path constraints
d) States
e) Calling file

These inputs are discussed in the following sections,

3.6 Solution via Direct Transcription

Direct is a ***Matlab*** based application for solving single-phased optimal control dynamic optimisation and parameter estimation problems. It is based on direct transcription

formulations of optimal control problems and it incorporates a range of different discretisation methods. The solution to the optimal control problem defined in the preceding section would typically involve the application of an indirect method utilising Pontryagin's Maximum Principal, or similar for minimax, optimal control problems. Instead of performing this laborious task of deriving the necessary conditions and solving the resulting problem, direct transcription methods combines the latest in numerical methods with optimisation algorithms to automate the procedure. In this view, one can discretise the continuous problem by appropriate scheme to convert it into a large-scale parameter optimisation problem. This process has been automated in the software ***Direct***.

The idea behind direct transcription involves discretising the state and control representation of a continuous trajectory. Utilising this, the optimal control problem can be transcribed into a NLP problem. The optimal control problem can be thought of as a NLP with an infinite number of controls and constraints.

Each phase of the trajectory can be separated into segments such that they abide by the following:

$$t_1 = t_1 < t_2 < \ldots < t_n = t_F \tag{59}$$

where t_1 is the initial time and t_F is the final time. The individual time plots are set as nodes. The value of the state vector at a node point is $y_k = y(t_k)$ and the control vector is $u_k = u(t_k)$

In the direct transcription method, the values of the states and controls at the nodes are treated as a set of non linear variables. The differential equation of the problem is represented by a system of defect constraints that are enforced at each of the discretisation nodes. In the Hermite Simpson [41] discretization method, these constraints and bounds are imposed at the mid point of each the trajectory segment.

The geometry and control discretisation can be shown with a simple diagram as in Figure 3.8. The example below has the trajectory divided into 7 segments which can be represented by 8 discrete nodes with their corresponding time denoted by t_i, states denoted by y_i and controls denoted by u_i. The midpoint segment is also displayed.

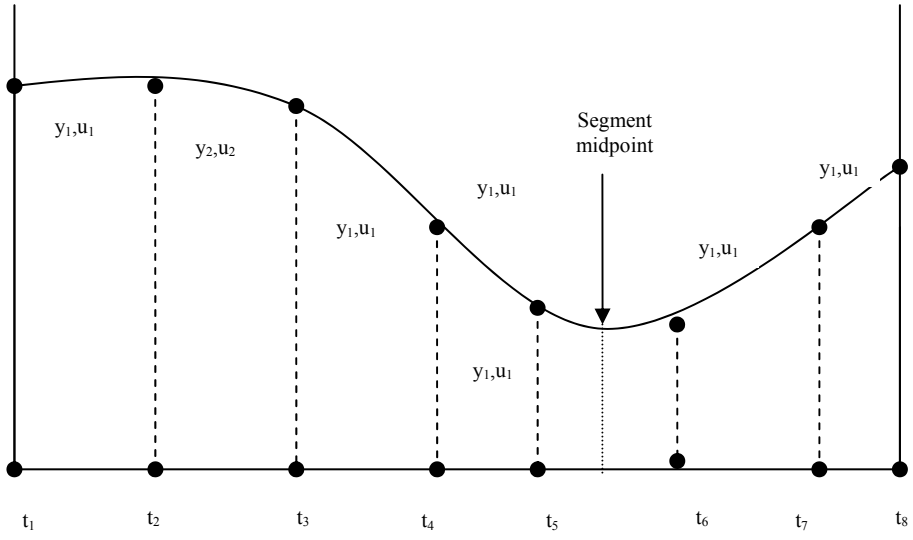


Figure 3.7: Trajectory and control discretisation [41]

For the compressed Hermite Simpson discretisation or collocation method, the non linear programming variables are given by:

$$x = \left[(y_1, u_1, u_m)_i, (y, u, u_m)_{i+1}, ..., (y, u)_f, t_0, t_f \right]^T \tag{60}$$

where u_m, u_{m2} are values of the controls at the mid point of the discretisation segments.

The defects for the Hermite Simpson discretisation method are given by:

$$\zeta_k = y_{k+1} - y_k - \frac{h_k}{6}[f(y_{k+1}, u_{k+1}) + 4f(y_{m_k}, u_{m_k}) + f(y_k, u_k)] \tag{61}$$

where *f(x,u)* represents the equations of motion evaluated at the nodes and mid points. The state vector and equations of motion at the mid point of the segment are given by

$$y_{m_k} = \frac{1}{2}[y_k + y_{k+1}] + \frac{h_k}{8}[f(y_k, u_k) - f(y_{k+1}, u_{k+1})] \tag{62}$$

and

$$f_{m_k} = f[y_{m_k}, u_{m_k}, p, t_k + \frac{h_k}{2}] \quad (63)$$

for k = 1,….n_N – 1

In the aforementioned equations, h_k is the time interval between segments. For equal duration, h_k is equal to $(t_f - t_i)/n_N - 1)$ where n_N is the number of nodes and $n_N - 1$ is the number of segments.

Cost function

In this file, the user defines the Mayer (terminal cost) and Integral parts to the cost function, which is defined as:

$$M[x(t_f), t_f] + \int_{t_0}^{t_f} L[x(t), u(t), t]dt \quad (64)$$

where M is the Mayer which contains the terminal cost.
L is the Lagrangian which contains the integral cost
t_f is the final time
$x(t)$ is the value of the states at the specified time t.
$u(t)$ is the value of the controls at the specified time t.

i. Minimum time

Minimum time is defined for a scenario, whereby the aircraft is required to clear the terrain in the shortest time possible without violating any path or additional constraints such as load factor or controls. Therefore:

Mayer = t_f (65)

Integral = 0 (66)

ii. Minimum clearance from terrain

Minimum clearance from terrain is defined for a scenario whereby the aircraft is required to conduct TF as well as TA without violating any path constraints or additional constraints such as load factor or controls. Therefore:

$$\text{Mayer} = 0 \tag{67}$$

$$\text{Integral} = (\text{Altitude of aircraft} + 50\text{m tolerance} - \text{Height of Terrain at the flying position of aircraft}) \tag{68}$$

iii. Sensitivity analysis

The final cost function is used to test the sensitivity of flight instruments such as altitude, speed and heading indicator. For example, if an aircraft is flying at a specified altitude, but the aircraft's indicated altitude meter is not matching with actual altitude indicator, then which one does the pilot follow and how much room for error is allowable? Therefore an investigation is required for these altitude, speed and heading indicators. Therefore:

$$\text{Mayer} = 0 \tag{69}$$

For the integral part, an interpolated method is adopted. In simple terms, a safe optimised trajectory is obtained. Following that some errors are introduced in the new trajectory in which the aircraft is flown. The objective is to find out, if the aircraft will be able to fly back onto the safe optimised trajectory. The full explanation of this methodology will be explained in chapter 6.

The mathematical equations for these cost functions used for this research are entailed in the Appendix A.3.1 Cost Functions.

Boundary conditions

Function BC = Boundary Conditions (y,u,t);
Global data;

Where ‘’data’’ is a global structure used by ***Direct*** for passing various parameters between functions and ***Snopt.*** In this case it is used to store values for the desired initial and final states. Any global variables can be used to pass values herein or they can simply be written in this file. The boundary constraints that were added on the aircraft model were on the X distance, Y distance, altitude, velocity, flight path angle, heading angle rate of change in angle of attack and rate of change in bank angle. The boundary conditions file is entailed in the Appendix A.3.1 Boundary Conditions

Path constraint

This function defines any nonlinear (or possible linear) path constraints. Setting the upper and lower bounds can enforce box path constraints on the states/controls. For this research, the path constraint was done by developing a terrain model using ***Terrain Generator*** as stated in section 2.3. Equations 57 and 58 were added to this file for the limitations on the load factor. An example of the path constraints and load factors are entailed in Appendix A.3.1 Path Constraints.

States equations

This file defines the dynamical equations for the problem. In this file, the state equations were functions of the states, controls, vector of parameters and time. Equations 6 to 43 were added to this file for the full aerodynamics and propulsion of the F 4 Phantom aircraft which are shown in Appendix A.3.1 States. The equations 43 to 48 are converted into a Quadratic Problem by first order differentiation that is done in ***Maple***. Additional requirements are implemented as purely algebraic constraint files in the form of:

$$x_L \le x \le x_U \tag{70}$$

$$u_L \le u \le u_U \tag{71}$$

Calling file

This file defines the main parameters of the problems such as variables, bounds, etc and calls the solver. The number of nodes that is required for the optimiser is set in this file.
Discretisation scheme, Discretisation.

This is an important parameter which determines which of the direct transcription formulations to be implement. Each different discretisation method has an influence on accuracy and solution speed. To set the Discretization, assign the following:

Discretization = [‘parameter’]

Where parameter is one of the Discretisation options. See ***Direct*** for all available 16 methods. For this research Hermite Simpson discretization method was used because it gave the best results for simulation purposes.
The calling file is attached in Appendix A.3.1 Calling file.

The structure of the optimization process is as follows

$$M[x(t_f), t_f] + \int_{t_0}^{t_f} L[x(t), u(t), t]dt \tag{72}$$

subject to dynamical constraints

$$\dot{x}(t) = f(x(t), u(t), t) \tag{73}$$

The box constraints are given by

$$x_L \leq x \leq x_U \tag{74}$$

$$u_L \leq u \leq u_U \tag{75}$$

After the structure of ***Direct*** has been explained, the next step is linking the software with ***Snopt*** and ***Terrain Generator***.

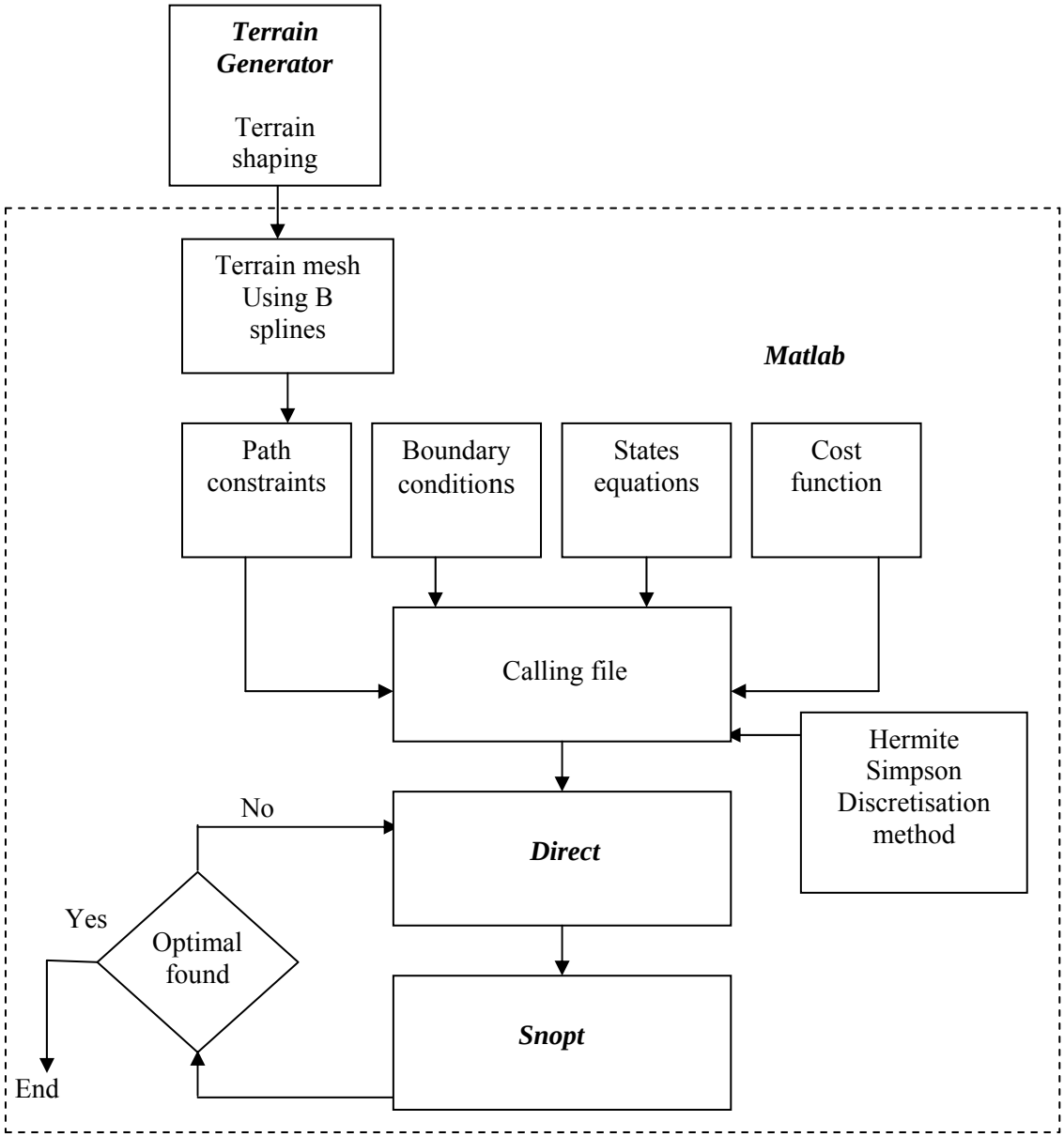


Figure 3.8: Procedure for optimisation

3.7 Collision Avoidance Problem

As required by ***Direct***, for an optimised collision avoidance problem, a path constraint is a requisite. Therefore the addition of obstacles is done in the path constraint file. It is assumed that terrain that is being avoided is available from a known terrain database. The principal

idea is to keep the flight trajectory free of obstacles whilst keeping the aircraft as close to the terrain as possible. To take into account obstacles such as buildings and other low lying obstacles, a clearance height h_c is utilised so that the aircraft maintains a safe tolerance above the terrain.

A variety of different techniques have been utilised to plan such trajectories. In this research, a weighted minimum time and minimum aircraft clearance from the terrain was utilised as the performance index.

$$h(t) \stackrel{\Delta}{=} z(t) - hc - h_T[x(t), y(t)] \tag{76}$$

where h_T [x(t),y(t)] is the height of the terrain at the aircraft position. For a scalar-valued function, $f : R \supseteq \Omega \rightarrow R,$ the –norm of f is defined by equation (76)

$$\| f \|_{l_p} \stackrel{\Delta}{=} \left(\int_{\Omega} |f(t)|^p \, dt \right)^{1/p} \tag{77}$$

$$J_1 = t_f \tag{78}$$

$$J_2 = \int_{t_0}^{t_f} h(t)dt = \| h(t) \|_{l_1} \tag{79}$$

The control problem requires the aircraft to be taken from a given state:

$$[\bar{x}(0), \bar{y}(0), \bar{z}(0), \bar{V}(0), \gamma(0), \psi(0)] \quad = \quad [\bar{x}(0), \bar{y}(0), \bar{z}(0), \bar{V}(0), \gamma(0), \psi(0)] \tag{80}$$

to a final state:

$$[\bar{x}(t_f), \bar{y}(t_f), \bar{z}(t_f), \bar{V}(t_f), \gamma(t_f), \psi(t_f)] \quad = \quad [\bar{x}f, free, free, free, \gamma_f, \psi_f] \tag{81}$$

In order to minimise the cost functions in Equation 78 and 79 subjected to the dynamical constraints defined by Equations 43 to 48 and the path constraint:

$$z(t) \geq h \; + h_T[x(t), y(t)] \tag{82}$$

The method by which the aircraft avoids the terrain is via interpolating the height $z(t)$ of the terrain with respect to the $x(t)$ and $y(t)$ coordinates of the terrain as given by in equation (82). The aircraft is always a safe distance above the terrain profile. The optimiser only calculates the clearance above the terrain, not the lateral distance to terrain, for example flying in a valley therefore this type of collision avoidance problem can only be applied to a terrain that does not have very steep slopes as shown in Figure 3.9.

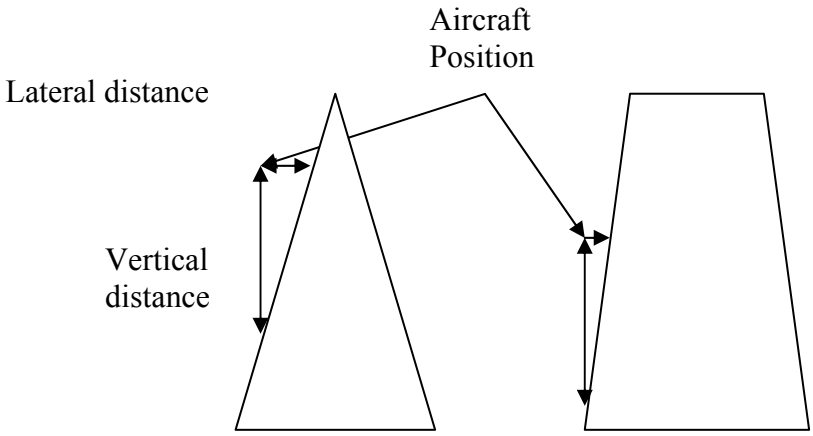


Figure 3.9: Determination of lateral distance for collision avoidance

3.8 Terrain Modelling

This part of the thesis investigates the different types of terrain models that were generated. The terrain models were generated in both two and three dimensions.

The utilisation of a high definition complex terrain is relatively difficult. To test the aircraft's manoeuvrability, a two-dimensional path constraint was employed initially. The terrain profile for these tests was created utilising equation 83 which was a polynomial curve fitting function. The disadvantage of polynomial fitting is that it can only provide results in two dimensional arrays only. Therefore another methodology was sought for the implementation of arrays.

$$f(x) = a_0 x^N + a_1 x^{N-1} + a_2 x^{N-2} + \ldots + a_{N-1} x + a_N \tag{83}$$

For the three dimensional simple terrain models, the function plots for cylinders and cones were utilised. The equation utilised for generating these plots is given by Equation 84. The terrain profile was created via organizing several cylinders and cones by increasing and decreasing the heights to represent terrain models. The principal idea was to generate a realistic representation of the mountainous terrain.

$$S(u,v) = r(v)\cos u, r(v)\sin u, v) \tag{84}$$

Validation of the methodology was done by using the optimiser to trial run on two dimensional and three dimensional simple terrain models. This was a stepping stone towards achieving the ultimate goal that was, to develop safe escape trajectories for collision avoidance for realistic three dimensional terrain models. Results and discussions for two dimensional and three dimensional simple terrain models are entailed in the list of publications section [46] and [47].

Three Dimensional Complex Terrain Model

A three dimensional complex terrain model for investigation was developed. The terrain utilised for the optimization problem was modelled utilising a matrix of elevation data provided by the terrain generation program. Terrain models were constructed which were representative of complex terrains. The terrain profiles were created via utilising third party software ***Terrain Generator***. After the terrain profiles have been created, the files are exported into Matlab in the text file format. Utilising B-splines [42], the terrain data is provided as a set of x and y coordinates, and a matrix of z coordinates representing the elevation. To obtain the solution, interpolated values of the elevation data are required. In addition, gradients of the constraints are calculated via utilisation of finite differences. This implies that the smooth derivatives of the terrain data are required for the solution algorithm to be effective. It is possible to provide C continuity by approximating the data with a tensor product cubic B spline of the form shown in equation 85. It is critical to provide a relatively accurate guess of the controls when utilising B splines. Hence, a good guess would produce results which are relatively accurate.

$$h_T(x,y) = \sum_{i=1}^{n_1}\sum_{j=1}^{n_2} c_{i,j} B_i(x) B_j(y) \tag{85}$$

Where $c_{i,j}$ are a set of coefficients and $B_i(x)$ and $B_j(y)$ form the basis for cubic B splines [42]

The required shape of the terrain is generated by ***Terrain Generator*** shown in Figure 3.10. However the coordinates that is obtained from this are not suitable to be used by ***Direct*** as the shape has non-smooth derivatives. As mentioned above, ***Direct*** requires smooth derivatives of the terrain data for the solution algorithm to be effective. Therefore B splines are used to create a surface mesh that creates smooth derivatives and which can be imported into ***Direct's*** path constraint file. Figure 3.11 and 3.12 illustrates plots of terrain with and without using B splines.

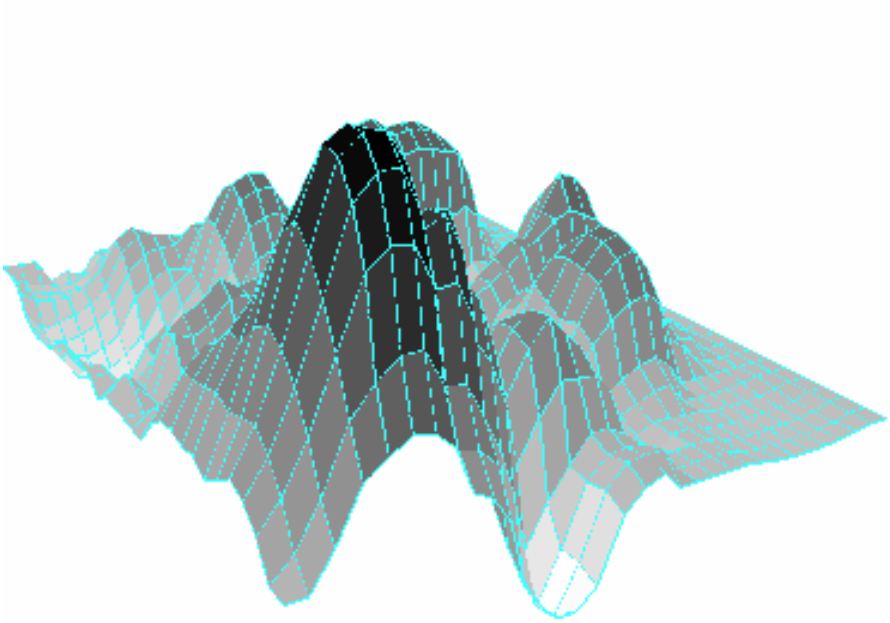


Figure 3.10: Three dimensional plots from ***Terrain generator***

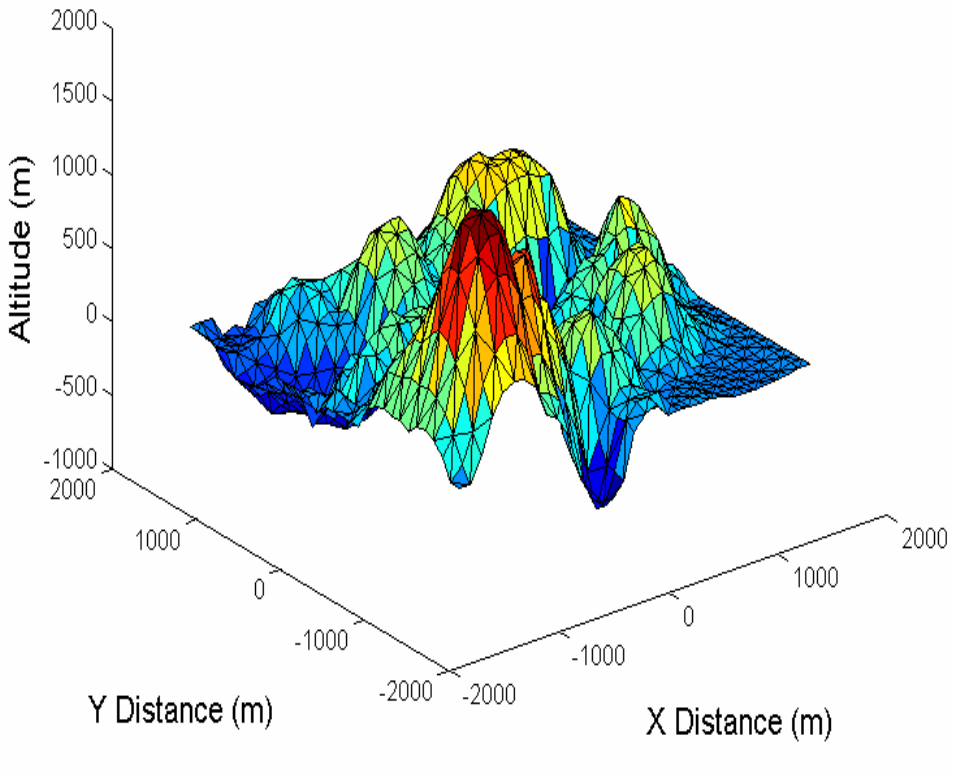


Figure 3.11: Three dimensional terrain profile without using B splines

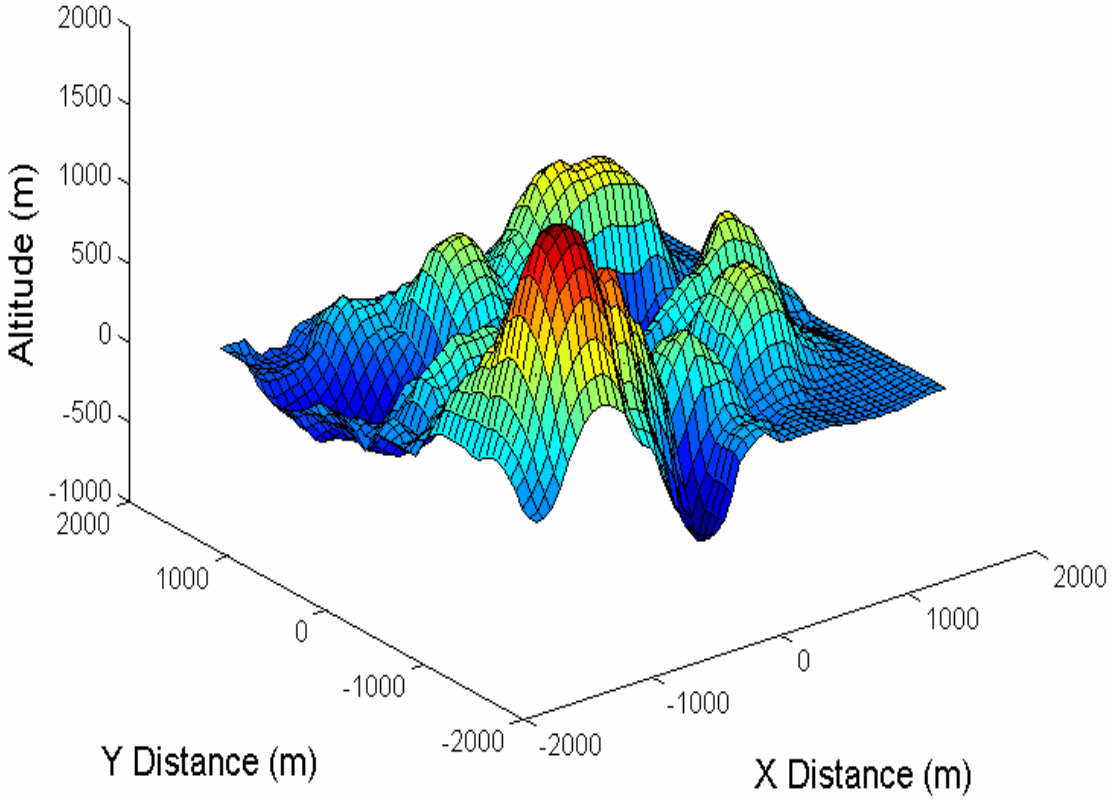


Figure 3.12: Three dimensional plot using B splines

3.9 Description of optimisation process.

The simulation procedure is as follows:

1) State initial position for aircraft: X distance, Y distance, Altitude, Flight path, Velocity and Heading angle
2) A limit was set on only the X distance end point.
3) The simulation was started with an initial number of nodes
4) Boundary conditions were added on the equations of motion of the aircraft.
5) The terrain models generated in the Terrain Generating software were set as path constraints:
6) For all the simulations, a minimum time and minimum clearance cost function was utilised.
7) Hermite Simpson discretization method was used for all the simulations conducted.
8) Initial guess for the control inputs for the aircraft were added.
9) The simulation process was started.
10) If no optimal solution is found, the initial values for the controls were changed.
11) Back to step 8 until optimal solution found.

3.10 Computational processing requirements

All the simulations were conducted on an Intel Centrino 1.7 Ghz computer with 1 gigabyte of ram with Windows XP, Matlab 7.0 and Terrain generator.

3.11 Summary

In this chapter, the components required for the collision avoidance problem were discussed. Although a brief introduction was given on the creation of two dimensional and three dimensional cones and cylinder obstacles, this thesis does not deal with solving such problems. For examples on the two and three dimensional simple terrain models, the reader can consult the published papers as per list of refereed papers [46] and [47].

4 Introduction to Simulation

4.1 Introduction

In this chapter, optimised trajectories for six different terrain profiles will be presented. The initial position was randomly varied. The final position was defined as a target point. The graphs obtained show one example where a maximum pull up manoeuvre fails and the go around manoeuvre saves the day. The rest of the plots three dimensional views for minimum time and minimum clearance conditions are provided in chapter 5.1. The plots for sensitivity analysis are provided in chapter 6.2. For all the control plots the individual result are entailed in the appendix A.2.1. The control plots for thrust and the rate of thrust for minimum time case scenario are not included as the aircraft's thrust was set to 100% and in a minimum clearance scenario, when the aircraft is conducting manoeuvre at high speed, the main controls that are used to control the aircraft are angle of attack and bank angle only. Figure 4.1 illustrates the thesis investigation methodology which summarizes the different simulations that were conducted. For the different terrain models, the aircraft had to adhere to the minimum time and minimum clearance cost functions, which was followed by sensitivity cost functions involving lateral position, speed indicator and altitude indicator errors. For all test scenarios involving minimum time, minimum clearance and sensitivity analysis, the initial and final positions of the aircraft are summarised as per Table 4.1. There are six different terrains models used for the simulation purposes. The terrains have been generated in different shapes and sizes to show that the optimiser can find a safe trajectory for the aircraft.

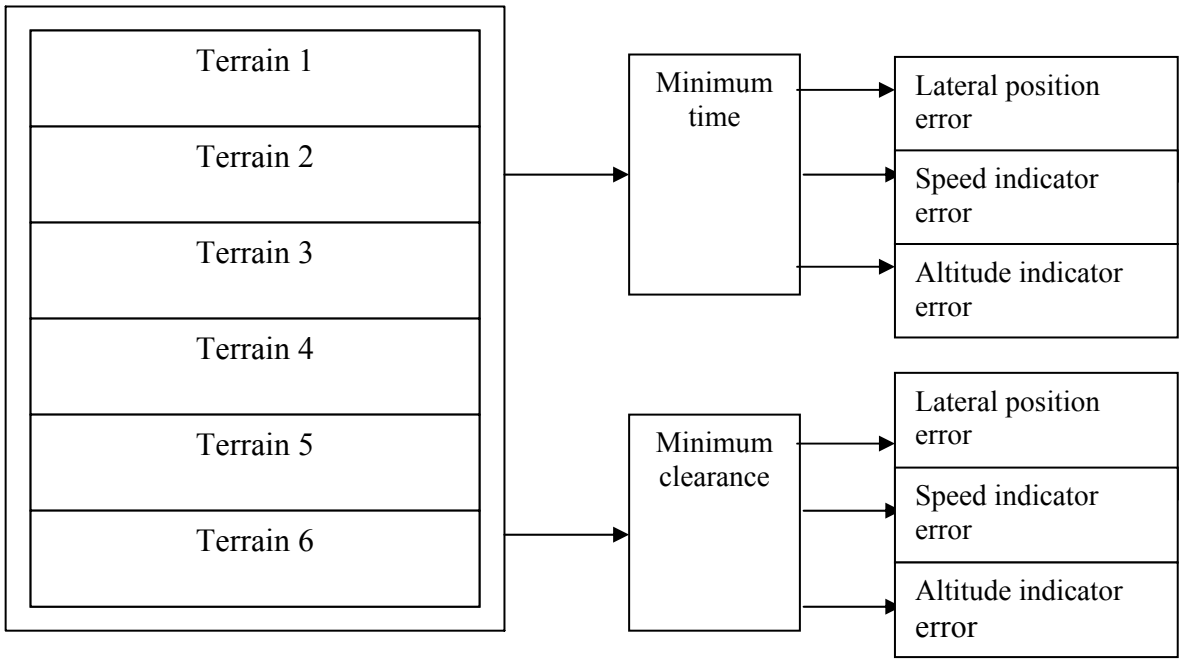


Figure 4.1: Thesis Investigation methodology

Table 4.1: Initial and final position of aircraft for minimum time and minimum clearance

	X Distance Start (m)	X Distance End (m)	Y Distance Start (m)	Z Distance Start (m)	Mach no. Start	Flight Path Angle Start (Deg)	Flight Path Angle End (Deg)	Heading Angle Start (Deg)	Heading Angle End (Deg)
Terrain 1	-1000	8000	-2100	300	0.5	0	0	0	0
Terrain 2	-2000	1500	-700	1000	0.45	0	0	0	0
Terrain 3	-9000	8000	0	1500	0.5	0	0	0	0
Terrain 4	-8000	8000	0	2100	0.5	0	0	0	0
Terrain 5	-8000	8000	0	2100	0.5	0	0	0	0
Terrain 6	-8000	8000	-800	1500	0.5	0	0	0	0

X Distance (Start) is the start position of the aircraft in the longitudinal axis.
X Distance (End) is the target point of the aircraft in the longitudinal axis
Y Distance (Start) is the start position of the aircraft in the lateral axis.
Z Distance (Start) is the start position of the aircraft in the altitude.
Mach number (Start) is the initial velocity of the aircraft.
Flight path angle (Start) is the position the aircraft is facing in the longitudinal axis.
Flight path angle (End) is the position the aircraft will face in the longitudinal axis.
Heading Angle (Start) is the position the aircraft is facing in the lateral axis.
Heading Angle (End) is the position the aircraft will face in the lateral axis.

For the simulations, the end target points for Y distance, Z distance and Mach number has been left free for the optimiser to work out the best possible collision avoidance trajectory.

5 Simulation Results and Discussion

5.1 Three Dimensional Scenario

For all the simulations conducted, the initial position in the altitude, lateral and longitudinal position was determined via setting the aircraft on a head on collision course into the terrain model.

Figure 5.1 shows that the maximum pull up manoeuvre did not prevent the aircraft from crashing into the terrain, but performing a go around manoeuvre prevented the crash. The aircraft crashed in the maximum pull up because it exceeded the constraints added on the rate of changes as shown in the appendix A.2.1

The minimum time plot Figure 5.2 for terrain data 1 depicts that the optimiser chose a path that involved the aircraft to perform a maximum pull up manoeuvre and fly to the end point by using momentum. The aircraft took 75.6 seconds to fly from the initial position to the final position for terrain model 1. The plots for rate of change in the appendix section 9.1 exhibit that the aircraft did not undergo abrupt changes in relation to the rate of angle of attack and bank angle. Figure 5.3 shows that the aircraft performed a combination of pull up and lateral manoeuvre to avoid the terrain and achieved a minimum clearance of 1.18m in addition tolerance of 50 metres. The control plots entailed in the appendix A.2.1 exhibit a significant change in the rate of change in angle of attack and bank angle.

Terrain data 2 was modelled as a terrain with a number of nonlinear peaks. From Figure 5.4, it shows that the optimiser optimised a trajectory which required little manoeuvring. There was no need to perform a go around manoeuvre as the terrain peaks were rather low. The control plots detailed in the appendix were smooth indicating that there were no significant rates of changes. The aircraft took 42.7 seconds to fly from the initial position to the final position for terrain model 2. From Figure 5.5, TF was conducted by performing a lateral manoeuvre and the aircraft maintained minimum clearance of 31.1 meters from the safety clearance of 50 meters. The reason the aircraft had a minimum clearance of 31.1 meters was because the aircraft struggled to fly close to the terrain as shown by the control plots in the appendix A.2.1 which exhibits that that there was significant deflection in the rates of changes in angle of attack and exceeded the limit set on the rate change in bank angle.

From Figure 5.6, it can be seen that the aircraft perform TA by conducting a lateral manoeuvre through a valley of the terrain model 3. The aircraft took 68 seconds to fly from the initial position to the final position for terrain model 3. Figure 5.7 shows that the aircraft performed TA and TF to keep as close to the terrain. The aircraft achieved a minimum clearance of 0.68 metres above the safety clearance of 50 meters.

The plots for terrain data 4 as shown in Figure 5.8 and 5.9 had similar results for minimum time and minimum clearance to terrain case. It is evident that the aircraft performed a lateral manoeuvre for both cases via avoiding the high peak of terrain. The aircraft took 70 seconds to fly from the initial position to the final position for terrain model 4. The difference however can be noted by observing the rate of change and control plots. There was a significant deflection in the rate of change for angle of attack and bank angle for the minimum clearance to terrain than minimum time case. The aircraft achieved a clearance of 0.18 metres above the safety clearance of 50 meters.

For the terrain model 5, the aircraft demonstrated a unique feature whilst generating an escape trajectory. The aircraft flew between the peaks of the terrain model by performing a well coordinated lateral manoeuvre as shown in Figure 5.10. The aircraft took 80 seconds to fly from the initial position to the final position for terrain model 5. The control plots as detailed in the appendix A.2.1 exhibits little variation for angle of attack, bank angle and rates of change in comparison to the minimum clearance scenario. Figure 5.11 shows that the aircraft flew around the terrain profile whilst achieving a clearance of 1.65 metres above the safety clearance of 50 meters.

The two plots for minimum time and minimum clearance to terrain exhibits that the escape trajectories generated by the optimiser. Figure 5.12 shows that the aircraft flew through the terrain valley without any collision. The aircraft took 89 seconds to fly from the initial position to the final position for terrain model 6. The control plots exhibit that the aircraft controls did not deflect significantly in order to manoeuvre around the terrain. The results shown in Figure 5.13 were similar to those in 5.11 which reveal that the aircraft conducted TF. The aircraft achieved a minimum clearance of 0.85 metres above the safety clearance of 50 meters. .

X start position (m)	Y start position (m)	Altitude start position (m)	Initial velocity (Mach)	Flight path (deg)	Heading angle (deg)
-5600	0	100	1.0	0	0
X finish position (m)	Safety clearance tolerance (m)				
8000	50				

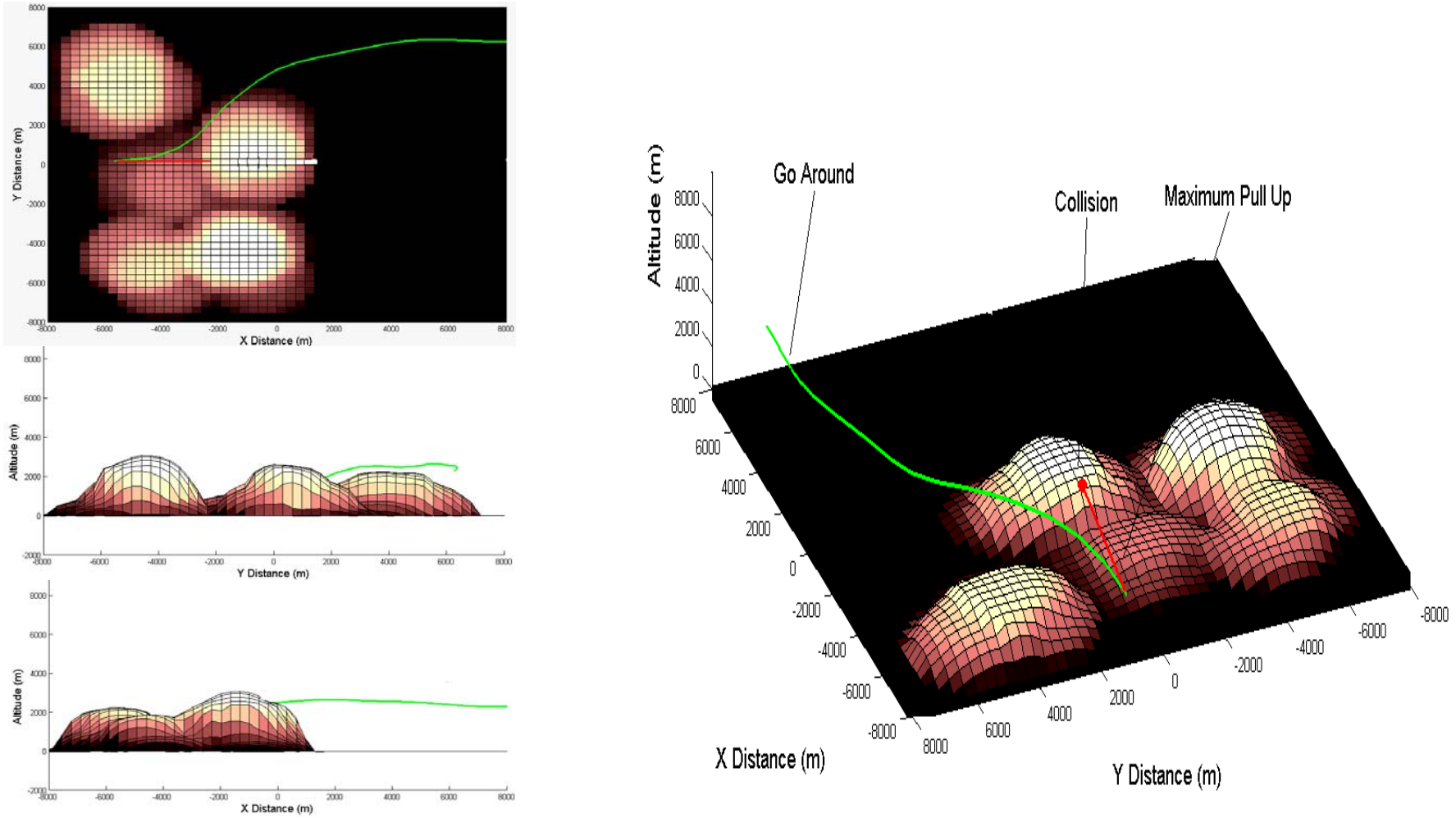


Figure 5.1: Three dimensional plot for a maximum pull up and go around

X start position (m)	Y start position (m)	Altitude start position (m)	Initial velocity (Mach)	Flight path (deg)	Heading angle (deg)
0	-2100	300	0.5	0	0

X finish position (m)	Safety clearance tolerance (m)	Time taken to clear terrain (sec)
8000	50	75.6

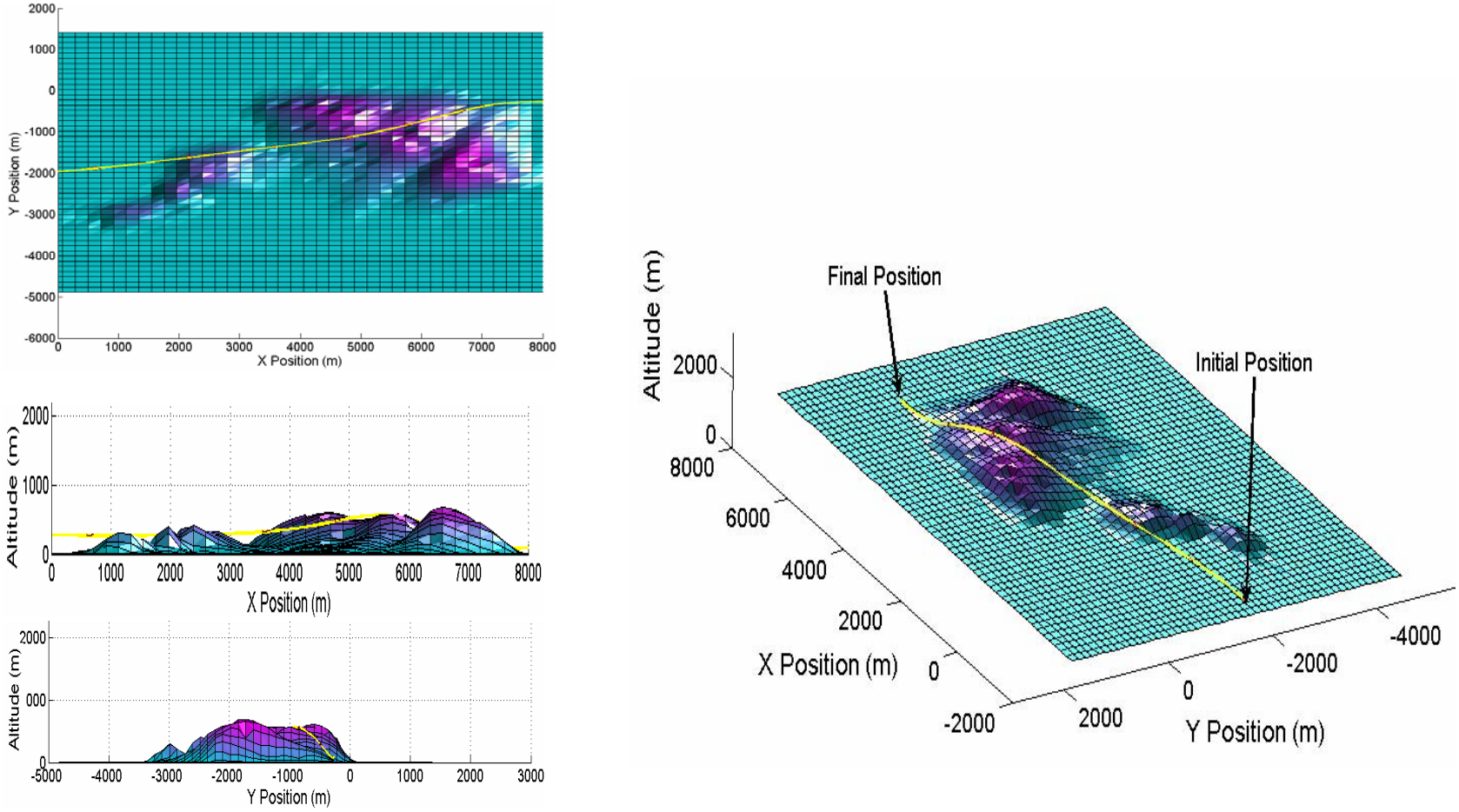


Figure 5.2: Three dimensional plot for terrain 1 minimum time scenario

X start position (m)	Y start position (m)	Altitude start position (m)	Initial velocity (Mach)	Flight path (deg)	Heading angle (deg)
0	-2100	300	0.5	0	0

X finish position (m)	Safety clearance height (m)	Proximity to terrain (m)
8000	50	1.18m

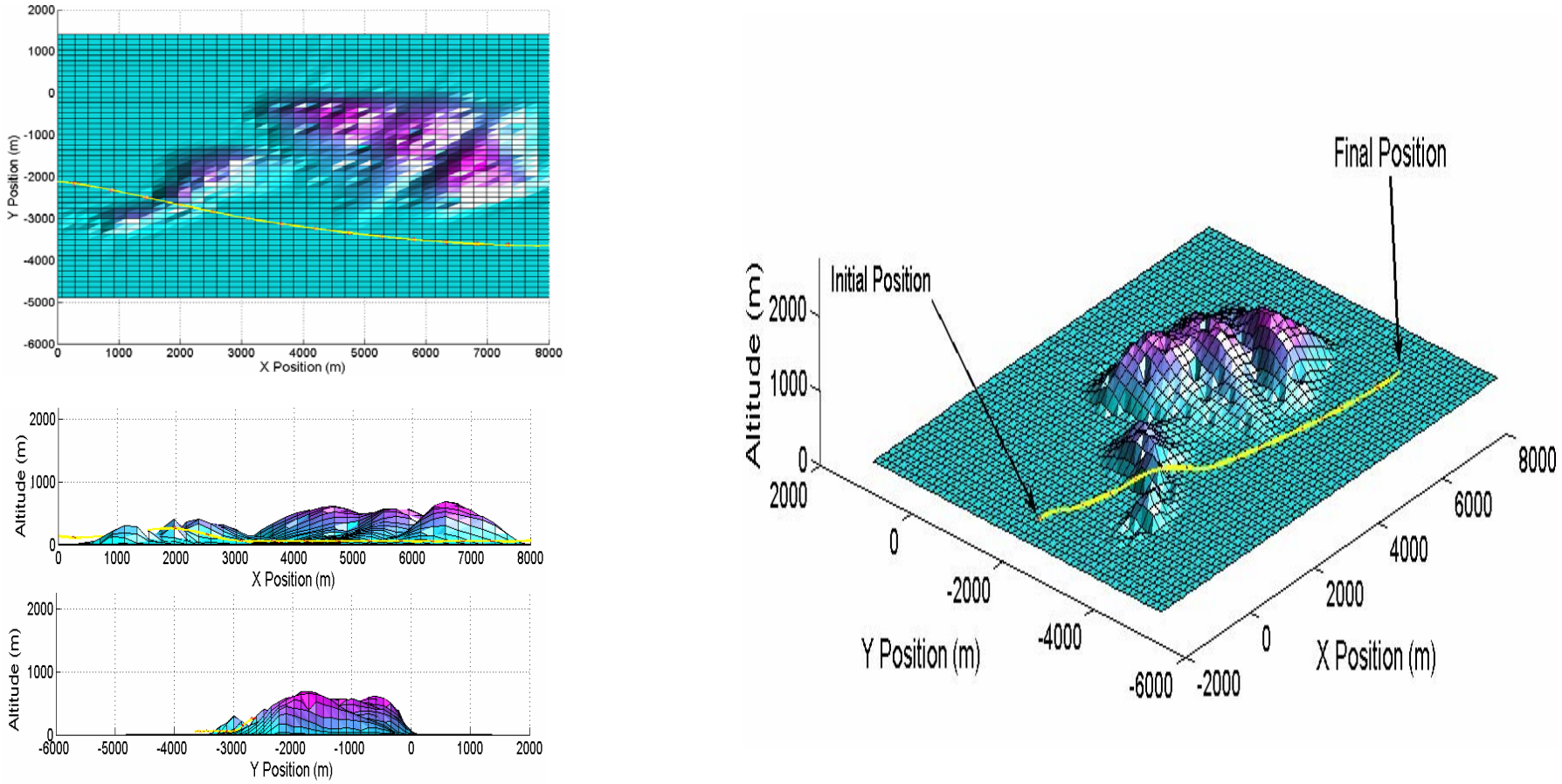


Figure 5.3: Three dimensional plot for terrain 1 minimum clearance scenario

X start position (m)	Y start position (m)	Altitude start position (m)	Initial velocity (Mach)	Flight path (deg)	Heading angle (deg)
-2000	-700	1000	0.45	0	0

X finish position (m)	Safety clearance height (m)	Time taken to clear terrain (sec)
1500	50	42.7

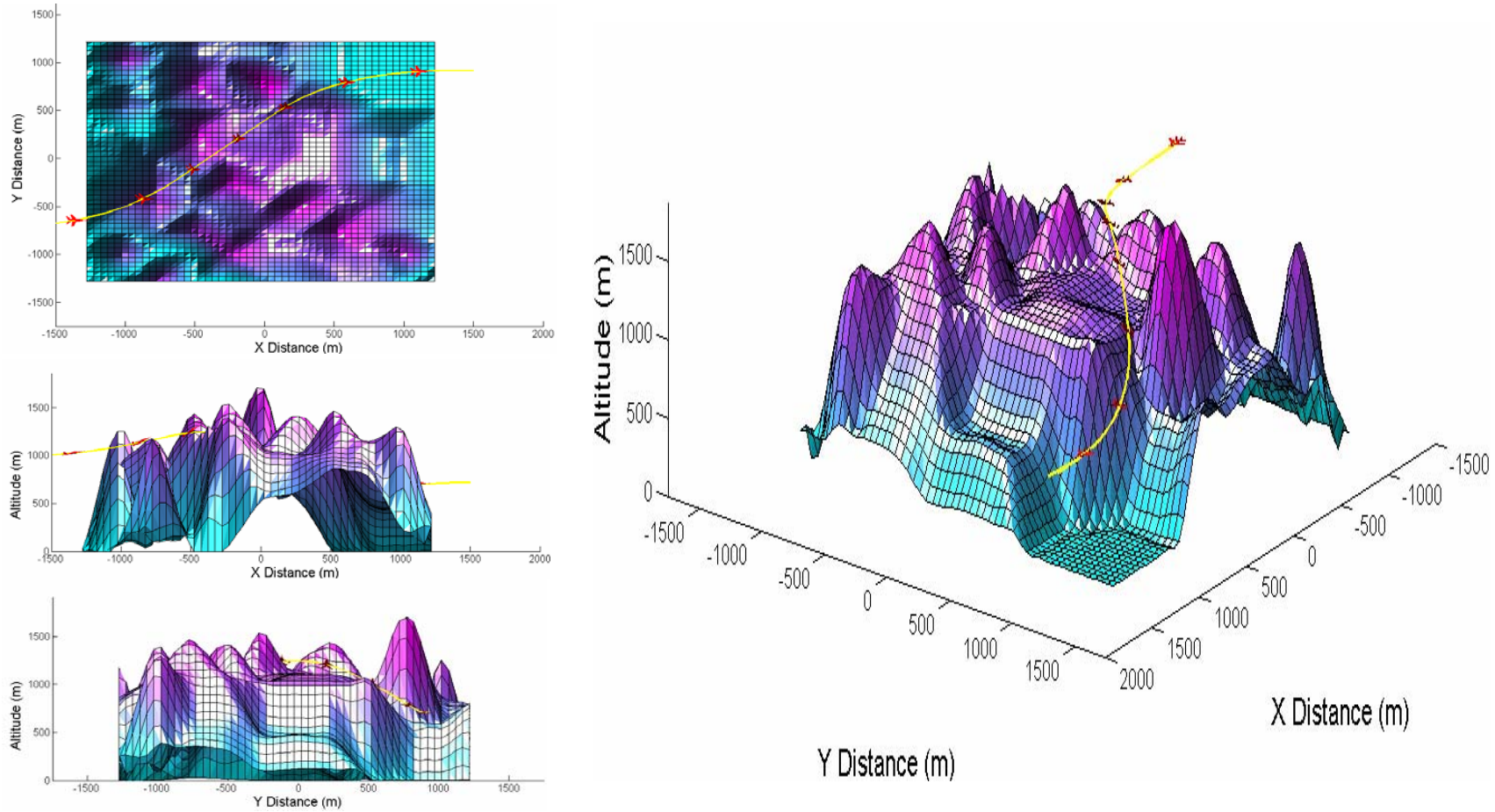


Figure 5.4: Three dimensional plot for terrain 2 minimum time scenario

X start position (m)	Y start position (m)	Altitude start position (m)	Initial velocity (Mach)	Flight path (deg)	Heading angle (deg)
-2000	-700	1000	0.45	0	0

X finish position (m)	Safety clearance tolerance (m)	Proximity to terrain (m)
1500	50	31.06

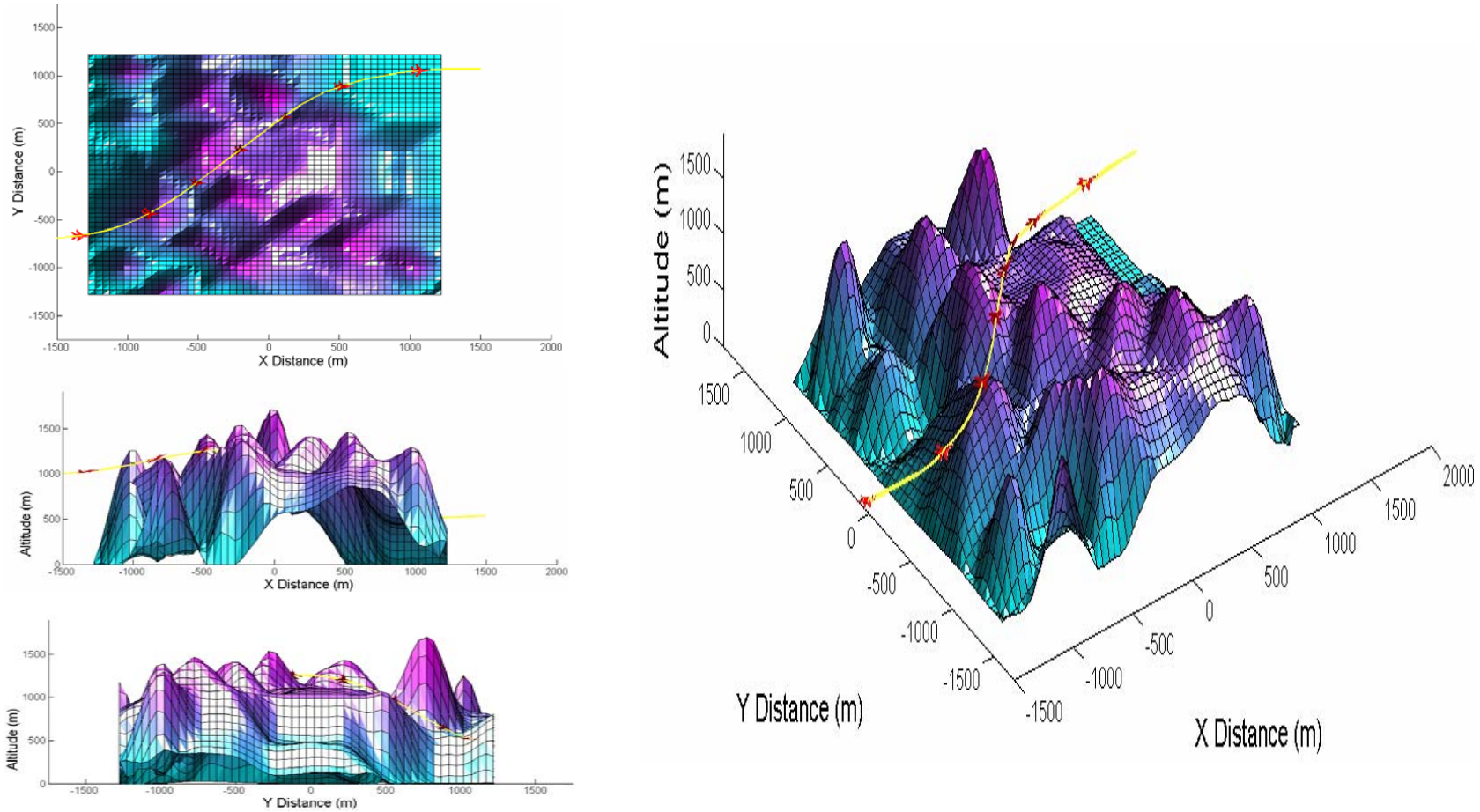


Figure 5.5: Three dimensional plot for terrain 2 minimum clearance scenario

X start position (m)	Y start position (m)	Altitude start position (m)	Initial velocity (Mach)	Flight path (deg)	Heading angle (deg)
-9000	0	1500	0.5	0	0

X finish position (m)	Safety clearance tolerance (m)	Time taken to clear terrain (sec)
8000	50	68

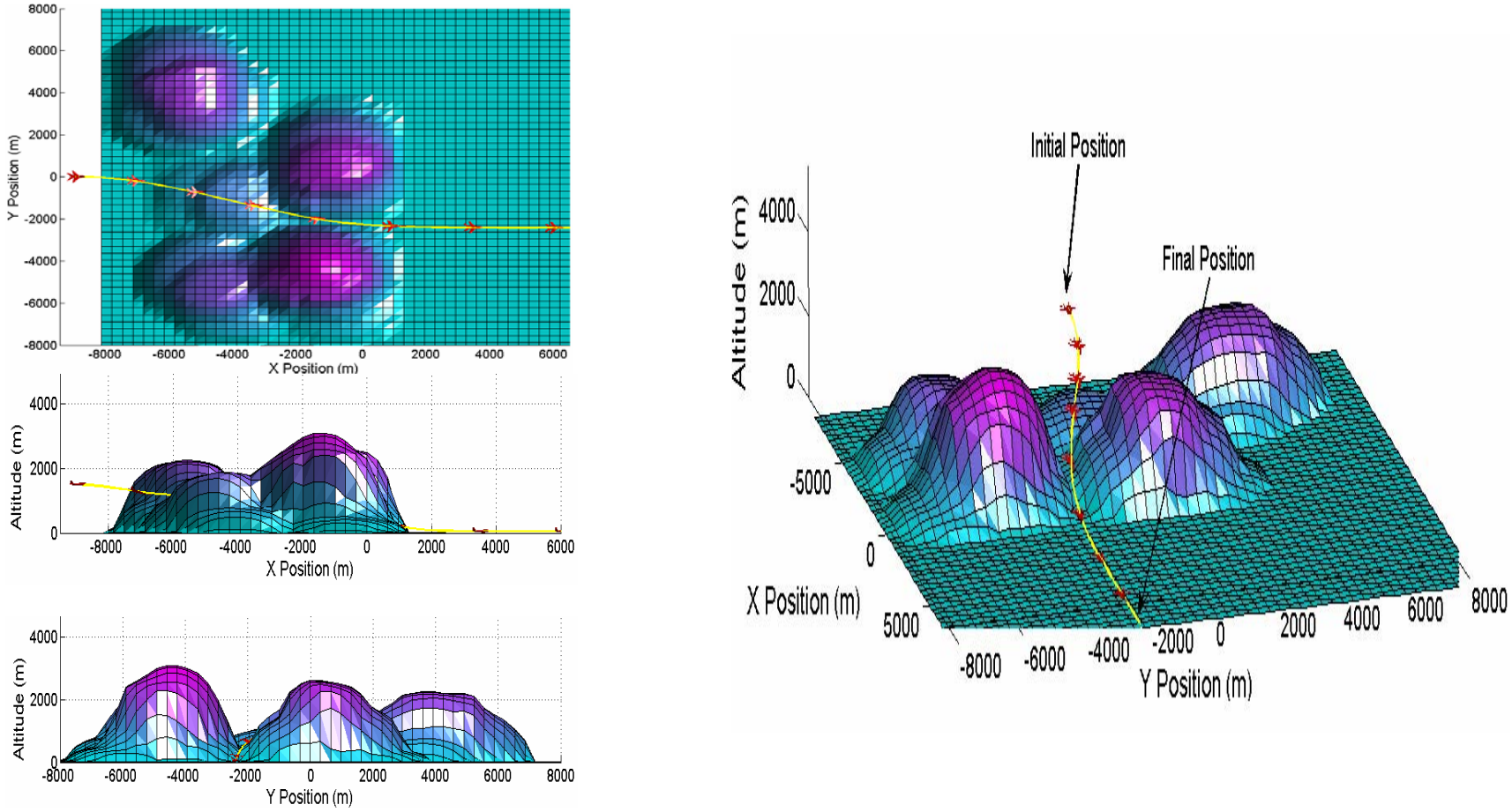


Figure 5.6: Three dimensional plot for terrain 3 minimum time scenario

X start position (m)	Y start position (m)	Altitude start Position (m)	Initial velocity (Mach)	Flight path (deg)	Heading angle (deg)
-9000	0	1500	0.5	0	0

X finish position (m)	Safety clearance tolerance (m)	Proximity to terrain (m)
8000	50	0.68

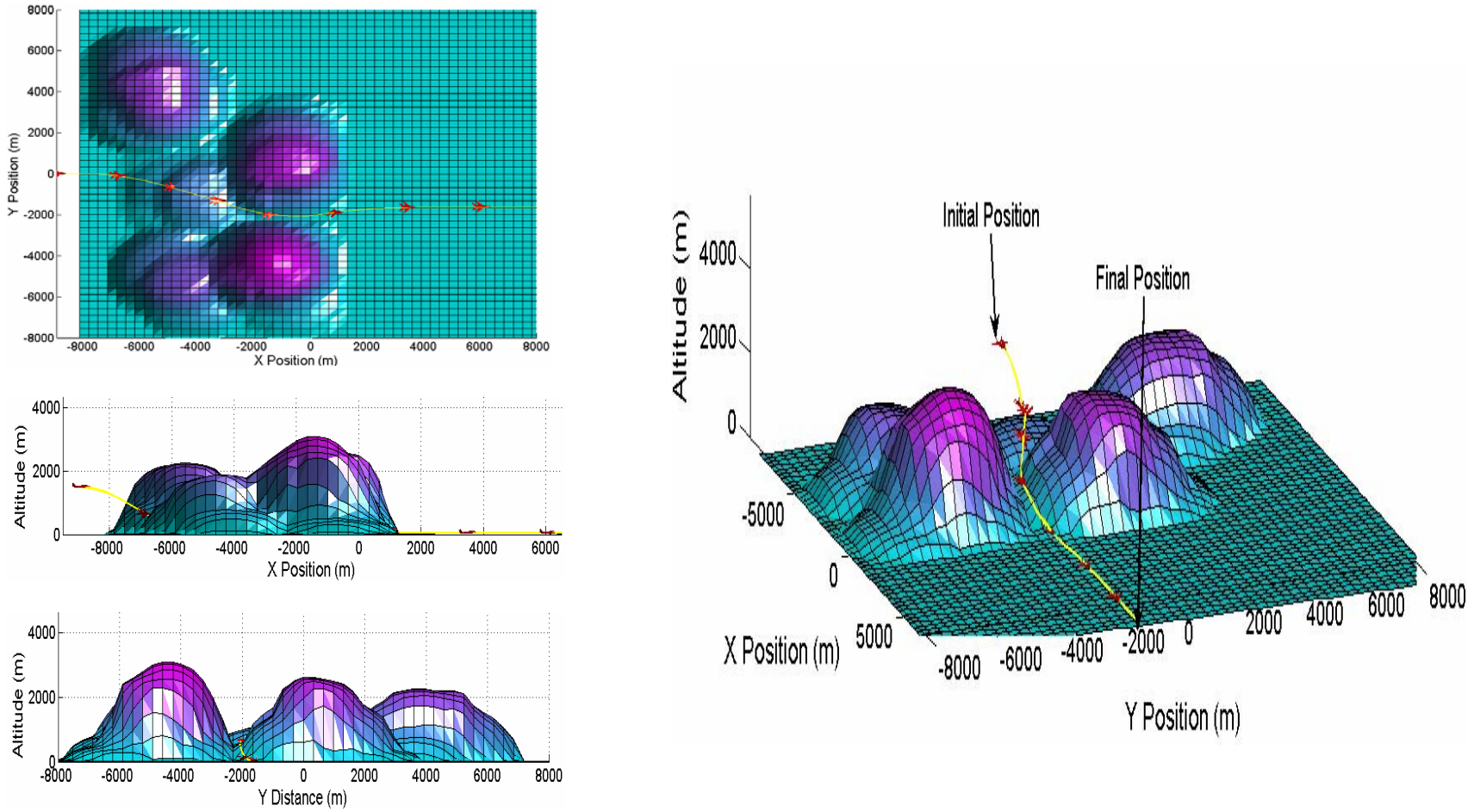


Figure 5.7: Three dimensional plot for terrain 3 minimum clearance scenario.

X start position (m)	Y start position (m)	Altitude start position (m)	Initial velocity (Mach)	Flight path (deg)	Heading angle (deg)
-8000	0	2100	0.5	0	0

X finish position (m)	Safety clearance tolerance (m)	Time taken to clear terrain (sec)
8000	50	70

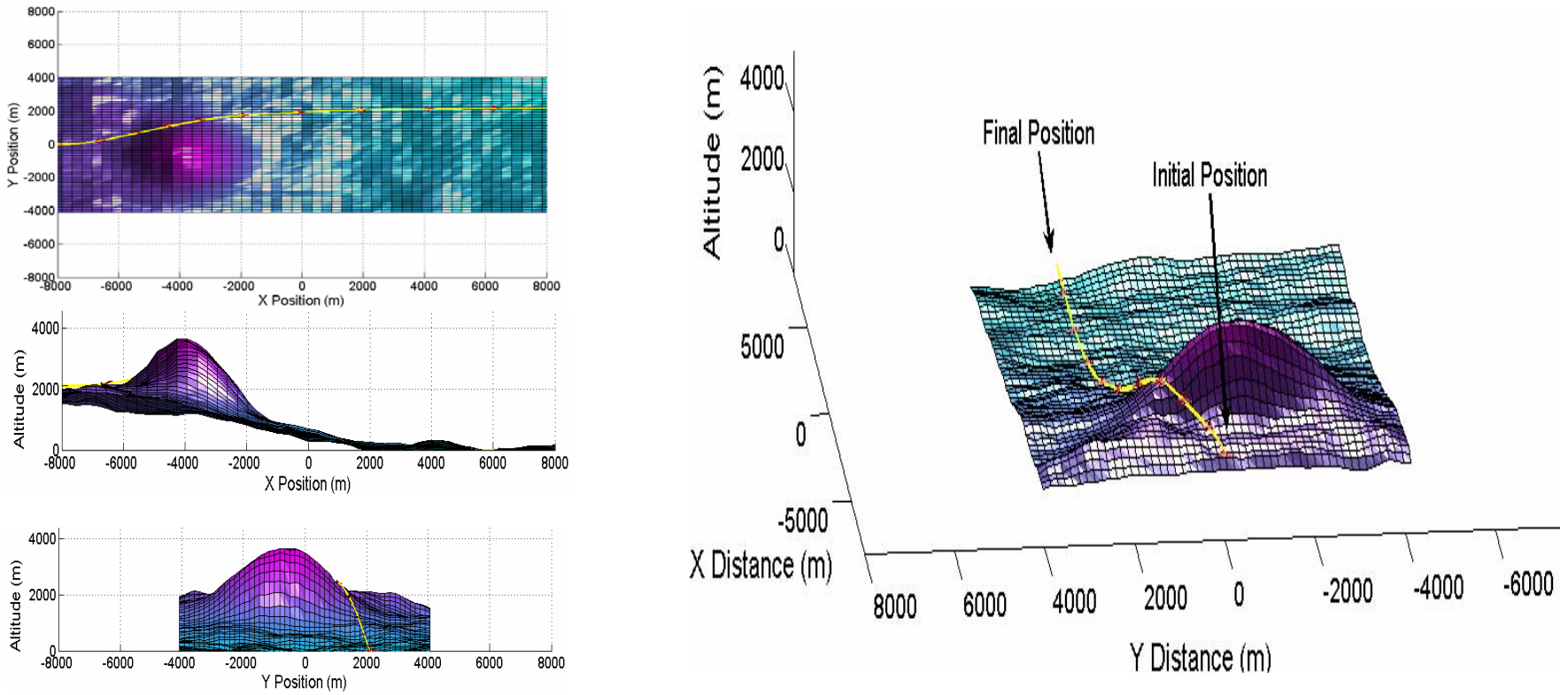


Figure 5.8: Three dimensional plot for terrain 4 minimum time scenario

X start position (m)	Y start position (m)	Altitude start position (m)	Initial velocity (Mach)	Flight path (deg)	Heading angle (deg)
0	0	2100	0.5	0	0

X finish position (m)	Safety clearance tolerance (m)	Proximity to terrain (m)
8000	50	0.18

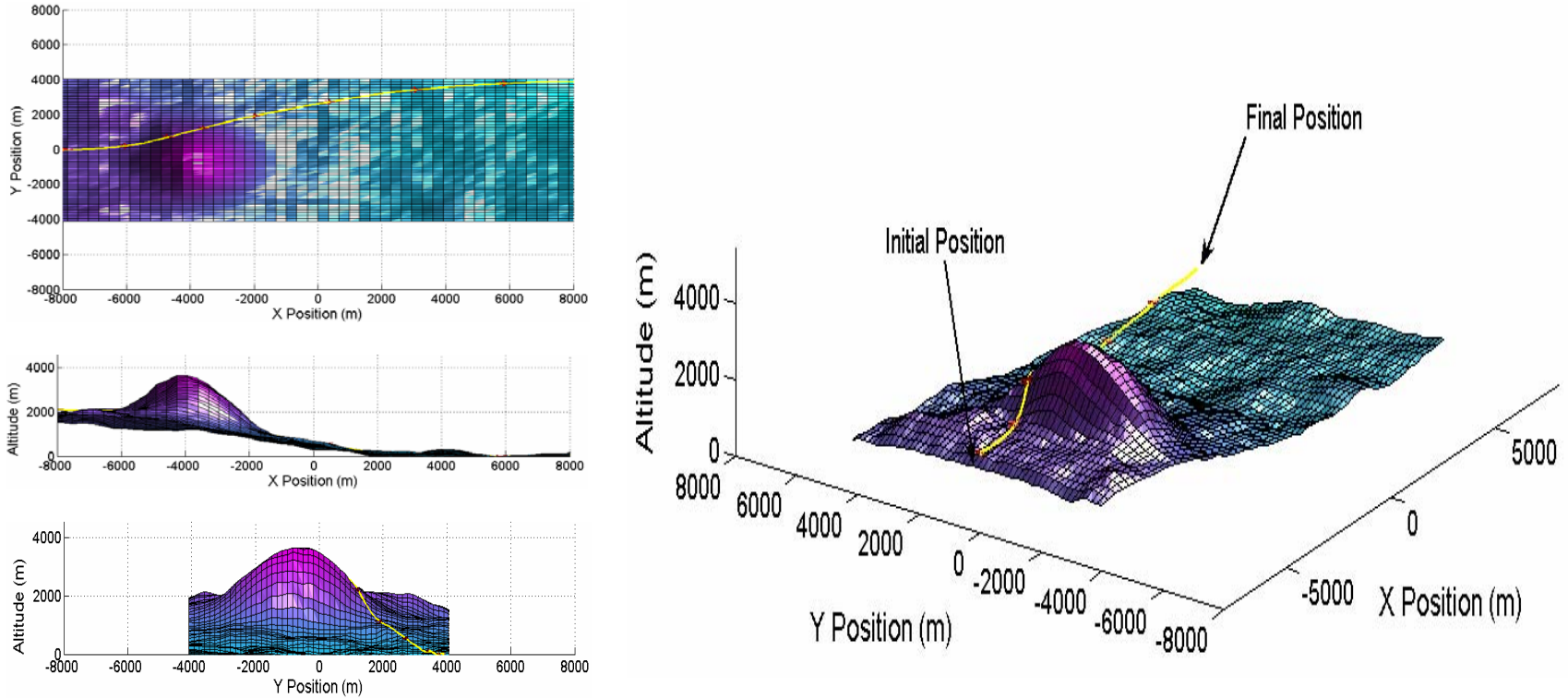


Figure 5.9: Three dimensional plot for terrain 4 minimum clearance scenario

X start position (m)	Y start position (m)	Altitude start position (m)	Initial velocity (Mach)	Flight path (deg)	Heading angle (deg)
-8000	0	2100	0.5	0	0

X finish position (m)	Safety clearance tolerance (m)	Time taken to clear terrain (sec)
8000	50	80

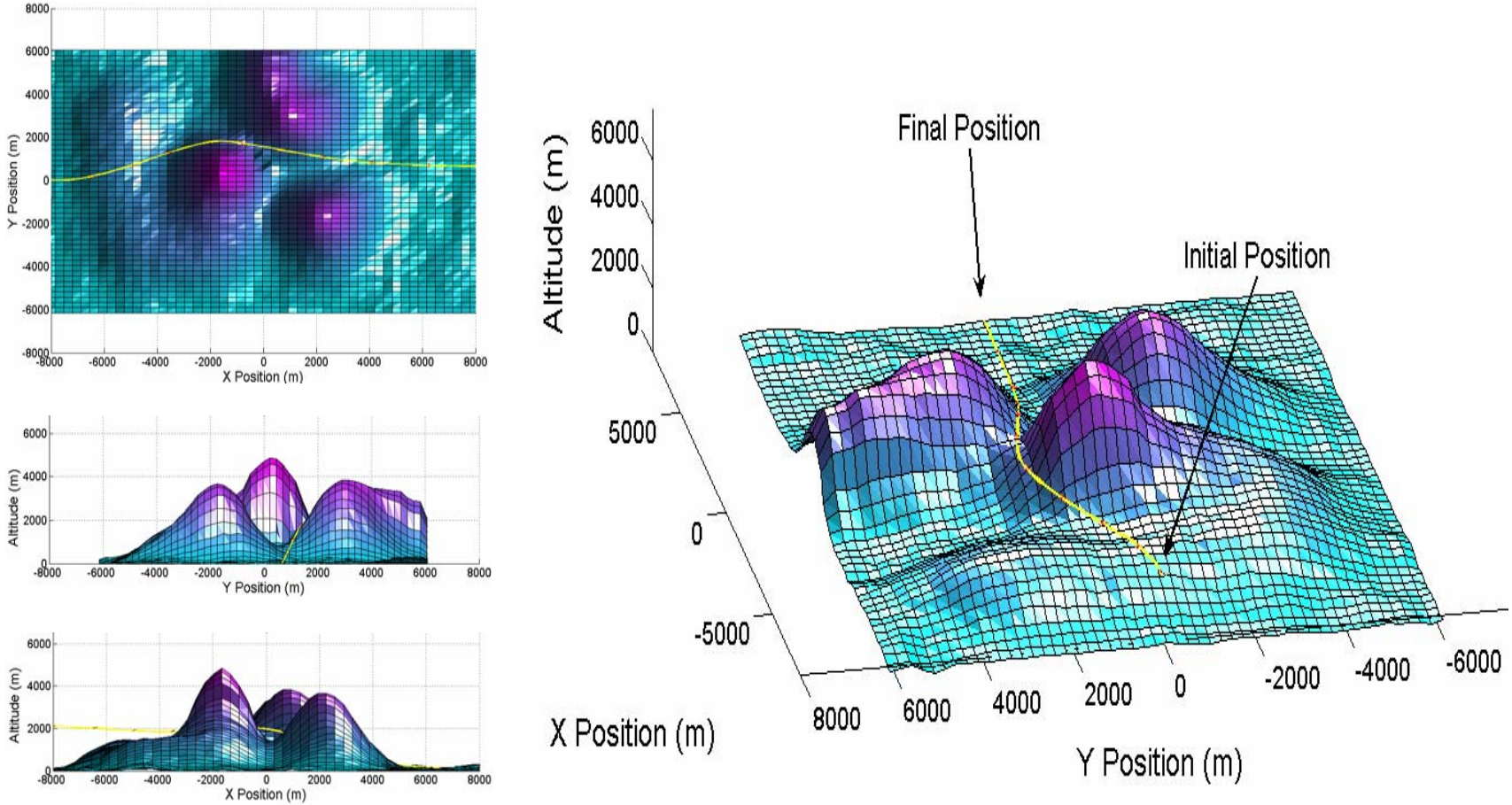


Figure 5.10: Three dimensional plot for terrain 5 minimum time scenario

X start position (m)	Y start position (m)	Altitude start position (m)	Initial velocity (Mach)	Flight path (deg)	Heading angle (deg)
-8000	0	2100	0.5	0	0

X finish position (m)	Safety clearance tolerance (m)	Proximity to terrain (m)
8000	50	1.65

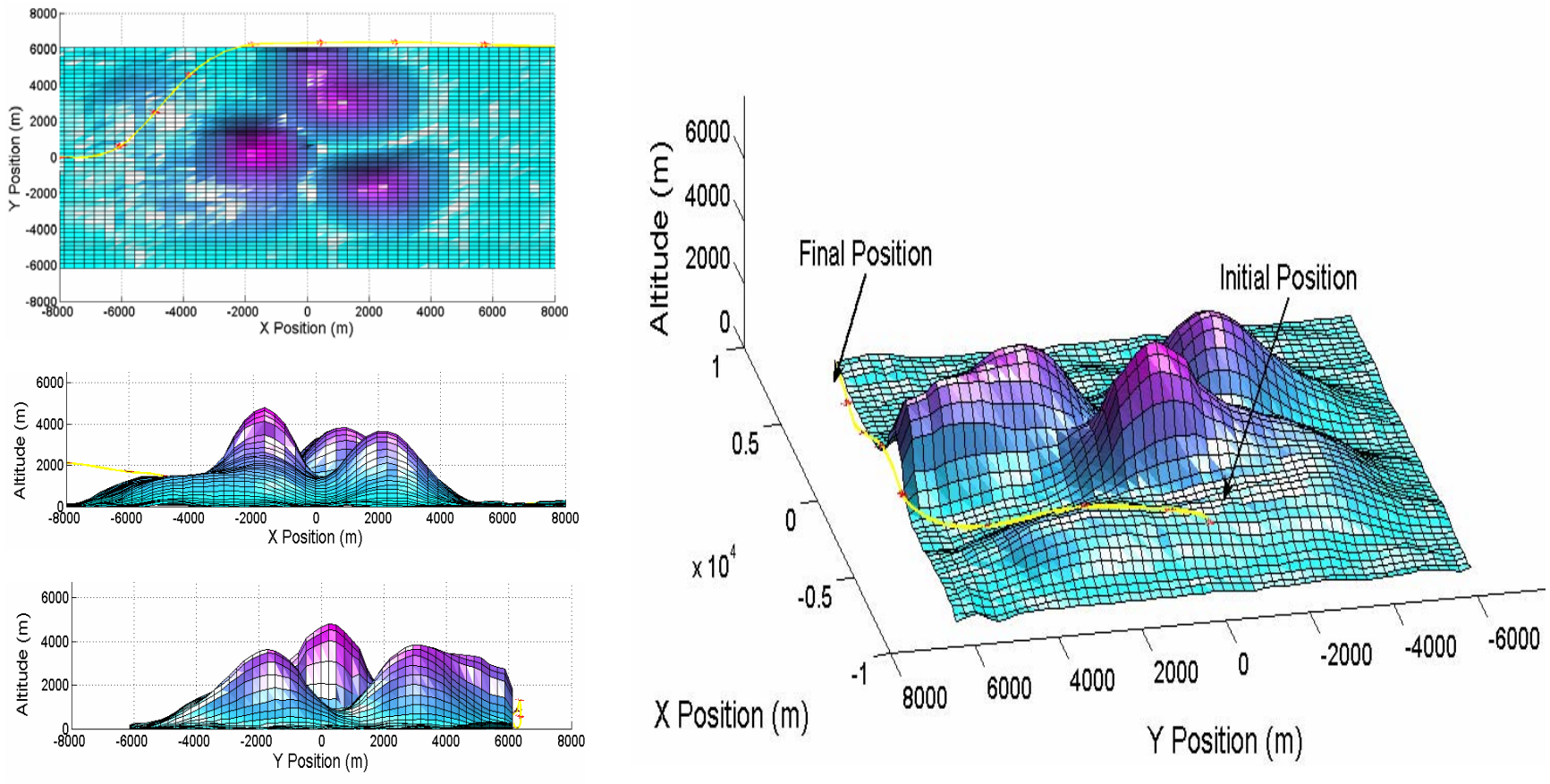


Figure 5.11: Three dimensional plot for terrain 5 minimum clearance scenario

X start position (m)	Y start position (m)	Altitude start position (m)	Initial velocity (Mach)	Flight path (deg)	Heading angle (deg)
-8000	0	1500	0.5	0	0

X finish position (m)	Safety clearance tolerance (m)	Time taken to clear terrain (sec)
8000	50	89

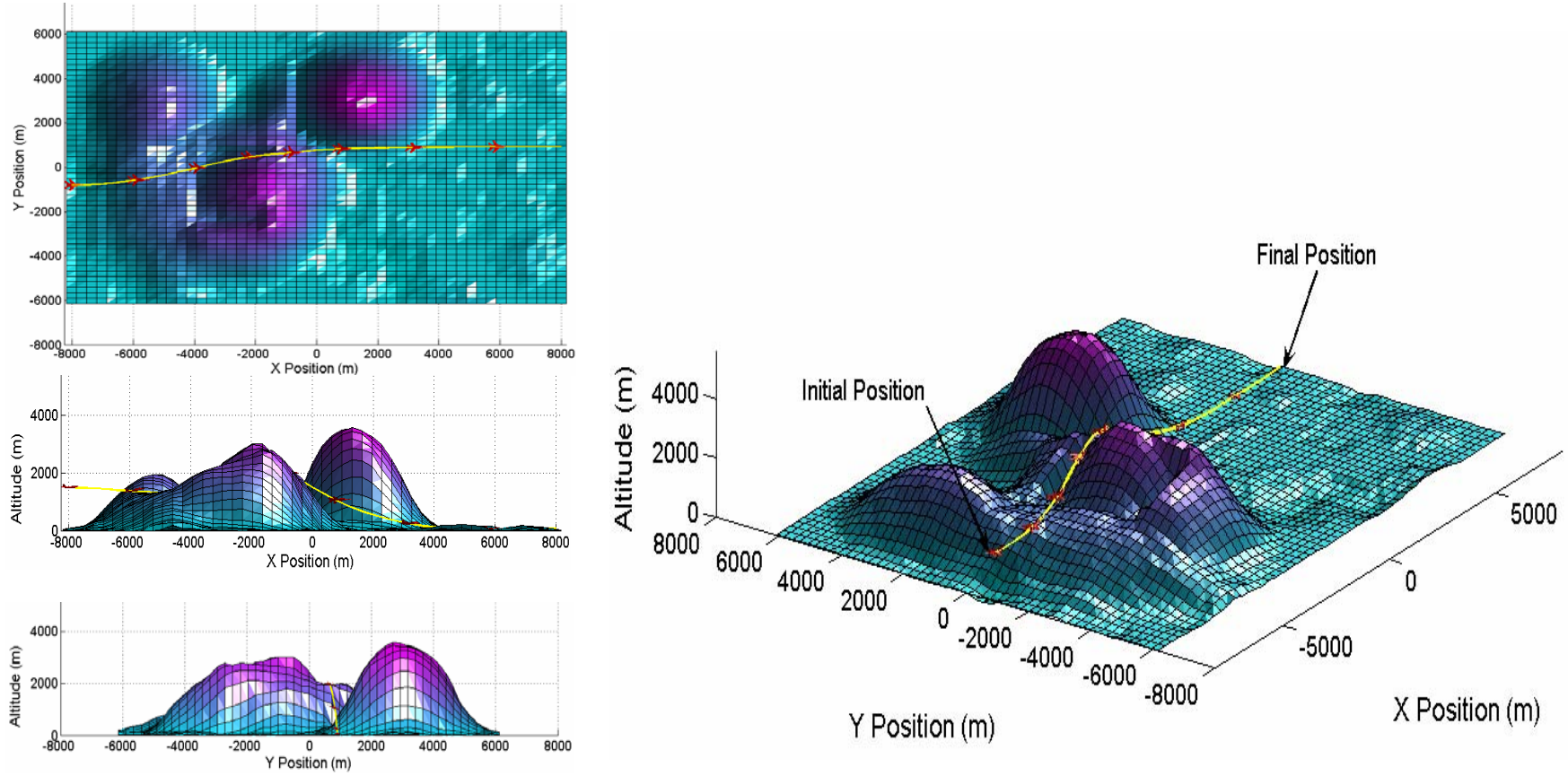


Figure 5.12: Three dimensional plot for terrain 6 minimum time scenario

X start position (m)	Y start position (m)	Altitude start position (m)	Initial velocity (Mach)	Flight path (deg)	Heading angle (deg)
0	-800	1500	0.5	0	0

X finish position (m)	Safety clearance tolerance (m)	Proximity to terrain (m)
8000	50	0.85

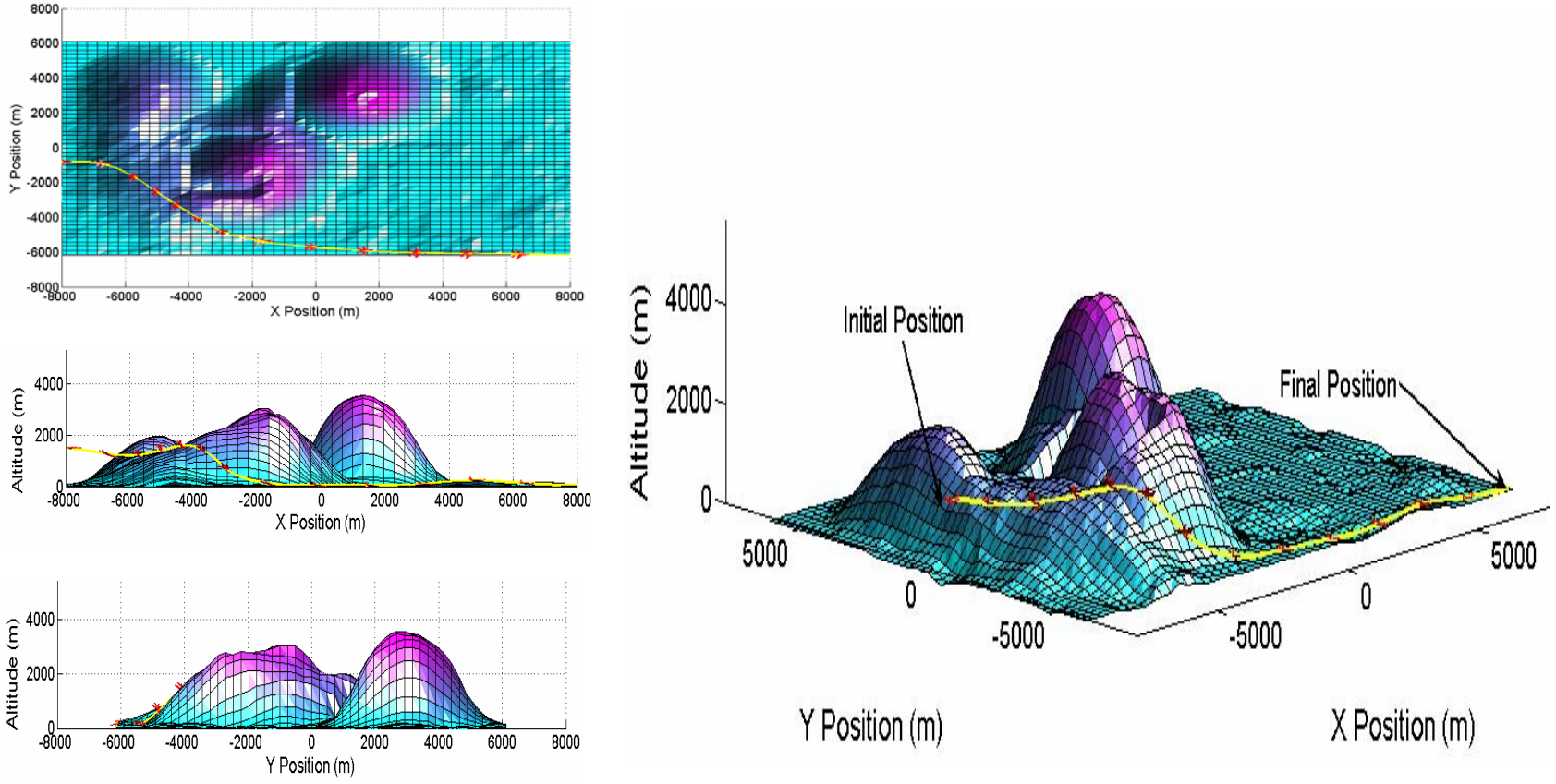


Figure 5.13: Three dimensional plot for terrain 6 minimum clearance scenario

6 Sensitivity Analysis Problem

6.1 Introduction

In this section, the methodology and the equations required in relation to the sensitivity studies are discussed. It is assumed that the aircraft is flying an optimised trajectory. The scenario is set as; the aircraft is flying at the optimised trajectory. Generally the indicated speed, altitude or lateral position will guide the pilot, however if he chooses to do otherwise then this could lead to a problematic situation.

The level of confidence and trust the pilots have in flight instruments plays a critical role. In attempt to combat this situation, certain tolerances in the states errors have been allowed. Equations 86 to 88 shows the tolerance parameters utilised for this purpose.

$$M_{actual} = M_{instrument} \pm 0.01 \text{ Mach} \tag{86}$$

$$Z_{actual} = Z_{instrument} \pm 100 \text{ Meters} \tag{87}$$

$$Y_{actual} = Y_{instrument} \pm 100 \text{ Meters} \tag{88}$$

Therefore the introduction of errors in the navigational instruments acts as the upper and lower boundaries for simulation purposes. The solution is obtained by the method of interpolation. The steps are summarised below.

- First an optimal trajectory is generated. Optimal trajectory definition is that the trajectory is free of obstacles and is within the structural and aerodynamics limit of the aircraft.
- The optimised trajectory is entered into the software and is used as a reference trajectory.
- The software is run again, with the same conditions as the optimal trajectory, but the only differences added are the equations 86 to 88.
- The new trajectory is generated by interpolating the state values of the new trajectory with respect to time step of the reference trajectory as shown in equation 89.

$$J = \left(y_{new} - y_{ref} \right)^2 \quad (89)$$

where

y_{new} = trajectory generated by interpolating the values of y_{ref} and time step of the reference trajectory

y_{ref} = optimal reference trajectory

6.2 Sensitivity Analysis discussion

Figure 6.1, 6.2 and 6.3 exhibits the minimum time plot for terrain data 1 for in a scenario where the aircraft was subjected to errors in terms of the lateral position, speed and altitude.

- The aircraft did not deviate from its optimised trajectory whilst having errors in lateral position up to $\pm$ 500 metres.
- The aircraft started to deviate from its optimised trajectory when the actual speed became 6% less and 12% more than the reference speed.
- In this case, the aircraft's initial position was set at 300 meters from the ground, therefore, even having altitude errors of up to $\pm$ 300 meters from the reference altitude, the aircraft did not deviate from its optimised trajectory

Figure 6.4, 6.5 and 6.6 exhibits the minimum clearance plot for terrain data 1 for in a scenario where the aircraft was subjected to errors in terms of the lateral position, speed and altitude.

- The aircraft did not deviate from its optimised trajectory even though having lateral position errors up to $\pm$ 500 meters
- The aircraft stated to deviate from its optimised trajectory when the actual speed was 8% higher than reference speed.
- Same results as shown in figure 6.3

Figure 6.7, 6.8 and 6.9 exhibits the minimum time plot for terrain data 2 for in a scenario where the aircraft was subjected to errors in terms of the lateral position, speed and altitude.

- The aircraft did not deviate from its optimised trajectory even though having lateral position errors up to ± 500 meters
- The aircraft was bound for collision when the actual speed fell in the range of 3% to 15% and increased more than 15% of the reference speed
- The aircraft crashed when the actual altitude was 400 meters above the reference altitude

.

Figure 6.10, 6.11 and 6.12 exhibits the minimum clearance plot for terrain data 2 for in a scenario where the aircraft was subjected to errors in terms of the lateral position, speed and altitude.

- The aircraft did not deviate from its optimised trajectory even though having lateral position errors up to ± 500 meters
- The aircraft crashed when the actual speed was 3% more than the reference speed.
- The aircraft crashed into the terrain when the actual altitude was in the range of 100 to 500 metres above the reference altitude.

Figure 6.13, 6.14 and 6.15 exhibits the minimum time plot for terrain data 3 for in a scenario whereby the aircraft was subjected to errors in terms of the lateral position, speed and altitude.

- The aircraft did not deviate from its optimised trajectory even though having lateral position errors up to ± 500 meters
- The aircraft started to crash when the actual speed was 30% less than the reference speed
- Altitude errors up to ± 500 meters did not cause the aircraft to crash.

Figure 6.16, 6.17 and 6.18 exhibits the minimum clearance plot for terrain data 3 for in a scenario whereby the aircraft was subjected to errors in terms of the lateral position, speed and altitude.

- The aircraft did not deviate from its optimised trajectory even though having lateral position errors up to ± 500 meters
- The aircraft started to crash when the actual speed was 30% less than the reference speed

- Altitude errors up to ± 500 meters did not cause the aircraft to crash.

Figure 6.19, 6.20 and 6.21 exhibits the minimum time plot for terrain data 4 for in a scenario where the aircraft was subjected to errors in terms of the lateral position, speed and altitude.

- The aircraft did not deviate from its optimised trajectory even though having lateral position errors up to ± 500 meters
- The aircraft crashed when the actual speed was 20% less and 30% more than the reference speed
- The aircraft did not crash when the actual speed was up to 500 meters above the reference altitude but crashed when the actual speed was 200 meters below the reference speed.

Figure 6.22, 6.23 and 6.24 exhibits the minimum clearance plot for terrain data 4 for in a scenario where the aircraft was subjected to errors in terms of the lateral position, speed and altitude.

- The aircraft did not deviate from its optimised trajectory even though having lateral position errors up to ± 500 meters
- The aircraft crashed when the actual speed was 30% less than the reference speed.
- The aircraft crashed into the terrain when the actual altitude was 100 meters less than the reference altitude.

Figure 6.25, 6.26 and 6.27 exhibits the minimum time plot for terrain data 5 for in a scenario where the aircraft was subjected to errors in terms of the lateral position, speed and altitude.

- The aircraft did not deviate from its optimised trajectory even though having lateral position errors up to ± 500 meters
- The aircraft deviated from the optimised trajectory when the actual speed was less than 10% of the reference speed.
- The aircraft deviated from the optimised trajectory when the actual altitude was 400 and 500 meters below the reference altitude.

Figure 6.28, 6.29 and 6.30 exhibits the minimum clearance plot for terrain data 5 for in a scenario where the aircraft was subjected to errors in terms of the lateral position, speed and altitude.

- The aircraft deviated from the optimised trajectory when the aircraft was flown 400 meters left of the reference position.
- The aircraft deviated from the optimised trajectory when the actual speed was 10% and 20% less than the reference speed.
- The aircraft deviated from the optimised trajectory when the actual altitude was 200 and 300 meters less than the reference altitude.

Figure 6.31, 6.32 and 6.33 exhibits the minimum time plot for terrain data 6 for in a scenario where the aircraft was subjected to errors in terms of the lateral position, speed and altitude.

- The aircraft did not deviate from its optimised trajectory even though having lateral position errors up to $\pm$ 500 meters
- The aircraft was bound for collision when the actual speed was 4% and 6% lesser than the reference speed.
- The aircraft crashed when the actual altitude was 300 meters and 500 meters less than the actual altitude.

Figure 6.34, 6.35 and 6.36 exhibits the minimum clearance plot for terrain data 6 for in a scenario where the aircraft was subjected to errors in terms of the lateral position, speed and altitude.

- The aircraft did not deviate from its optimised trajectory even though having lateral position errors up to $\pm$ 500 meters
- The aircraft crashed when the actual speed was 8% and 10% less than the reference speed.
- Altitude errors up to $\pm$ 500 meters did not cause the aircraft to crash.

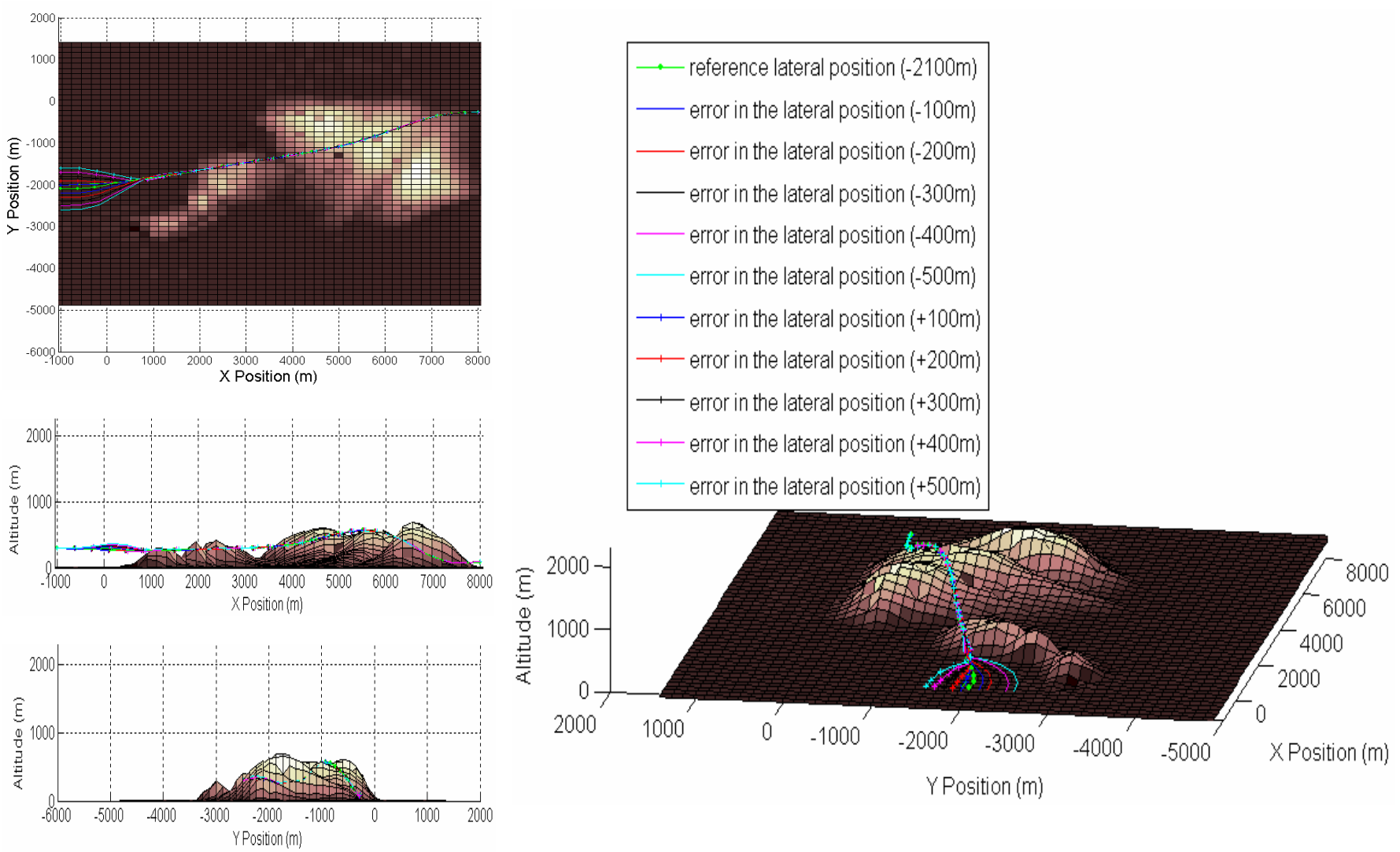


Figure 6.1: Error in lateral position for terrain 1 minimum time scenario

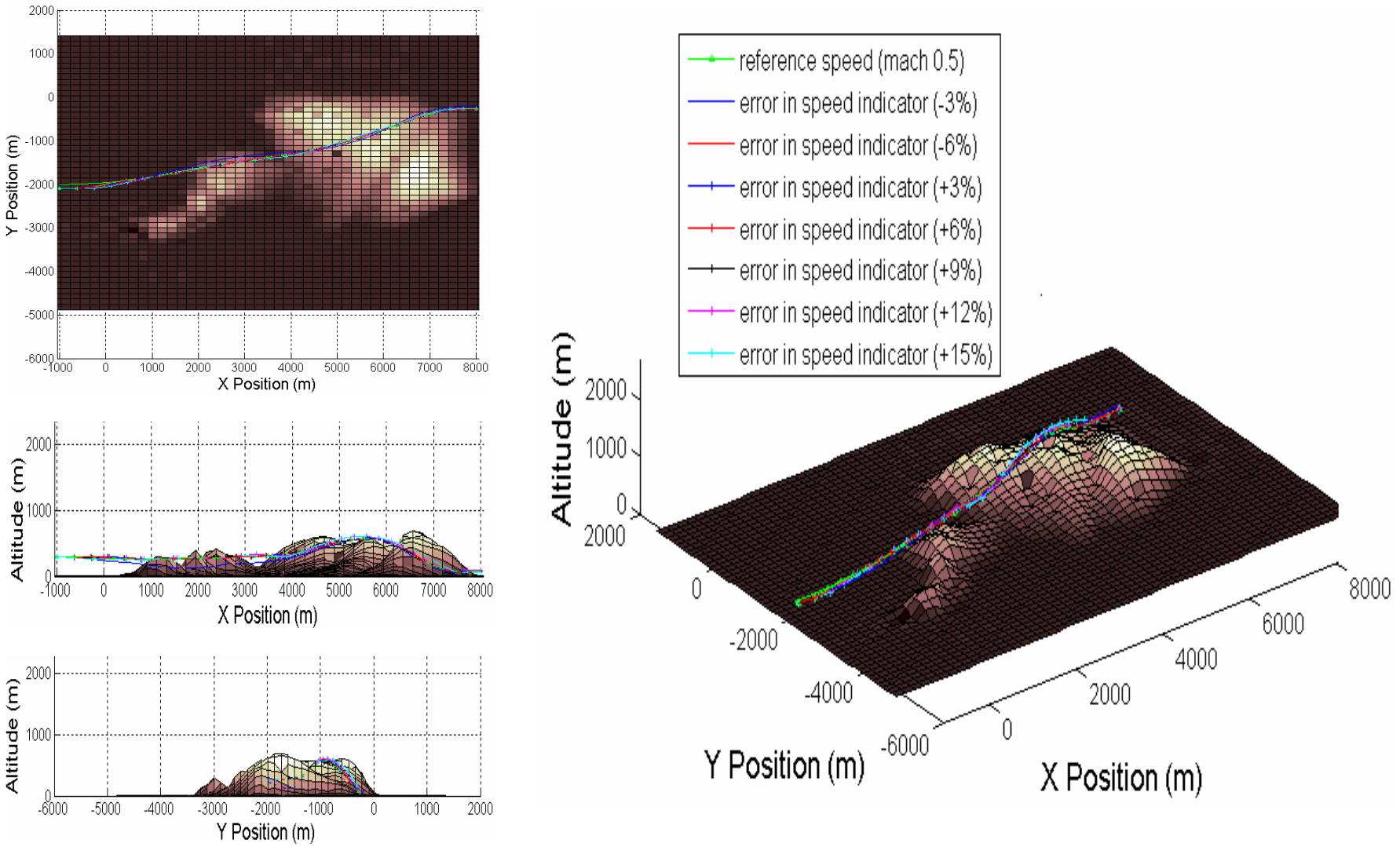


Figure 6.2: Error in speed indicator for terrain 1 minimum time scenario

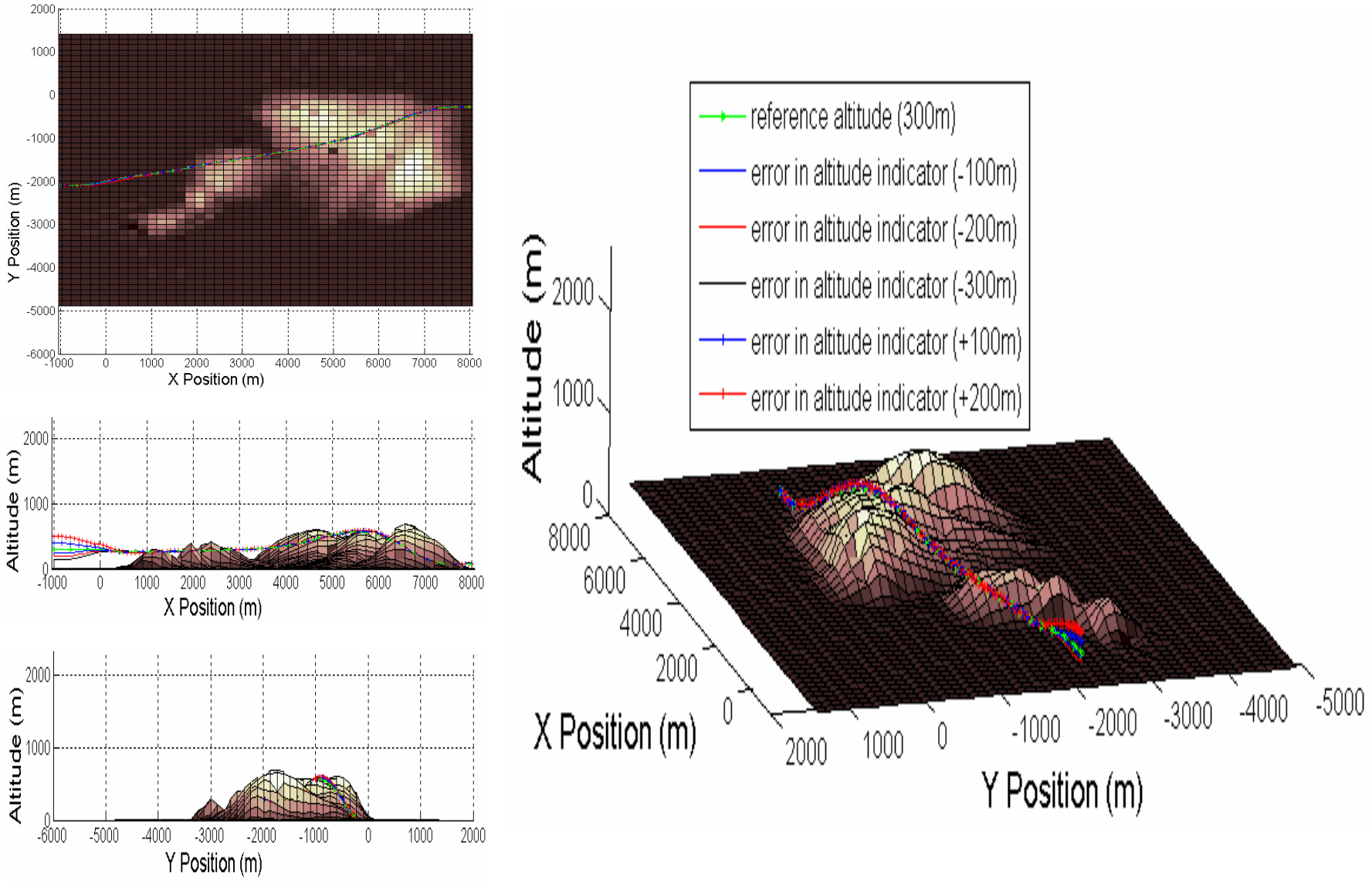


Figure 6.3: Error in altitude indicator for terrain 1 minimum time scenario

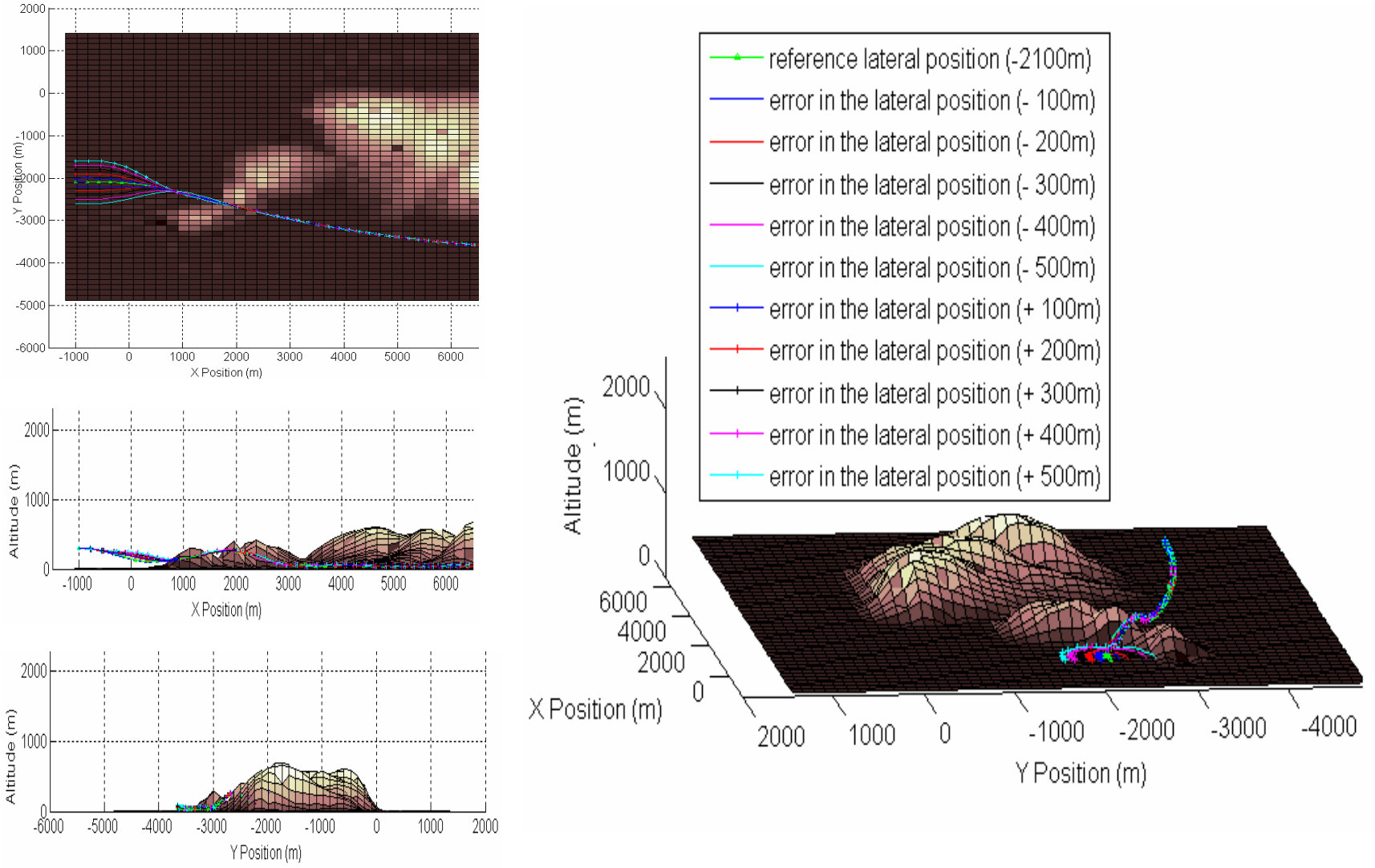


Figure 6.4: Error in lateral position for terrain 1 minimum clearance scenario

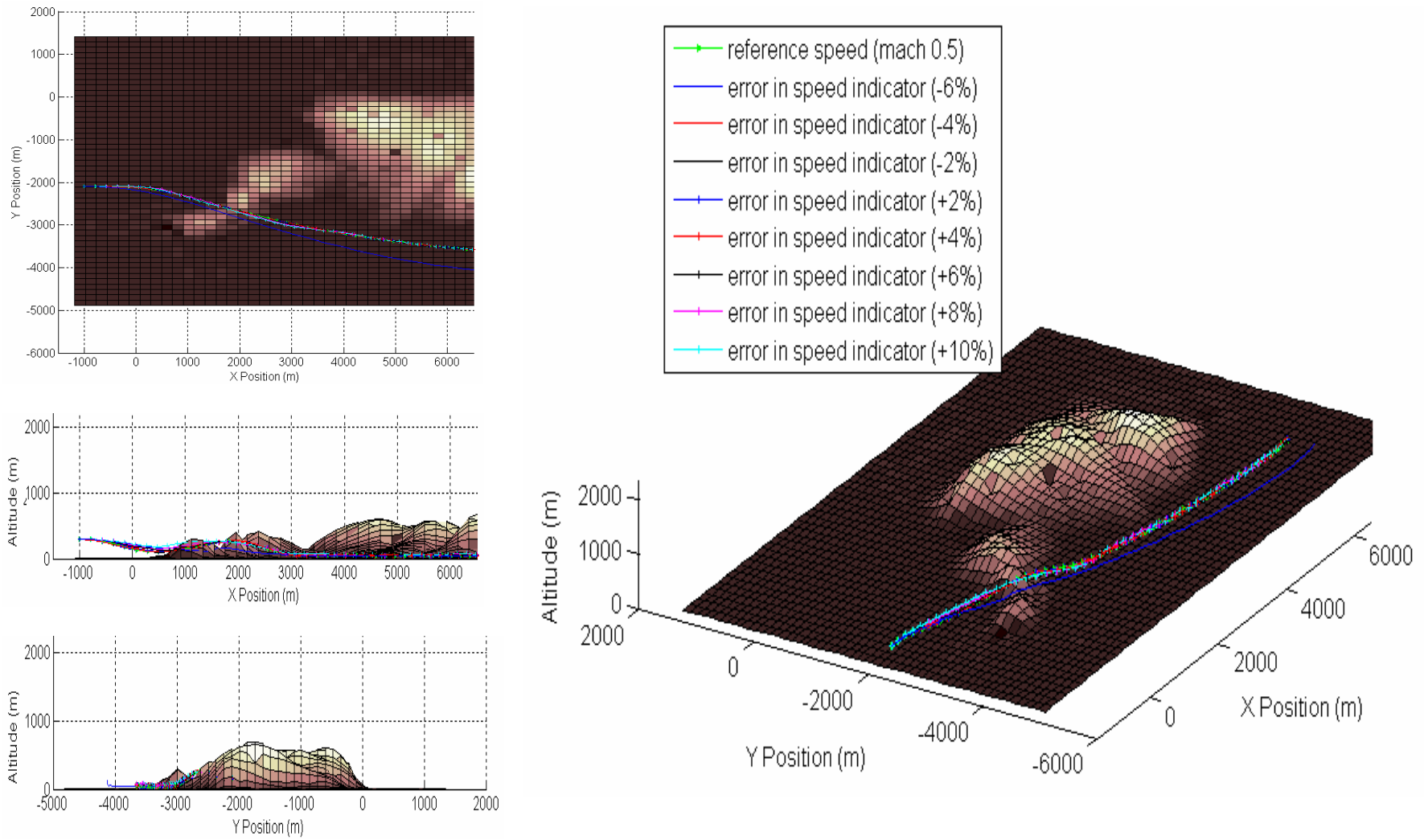


Figure 6.5: Error in speed indicator for terrain 1 minimum clearance scenario

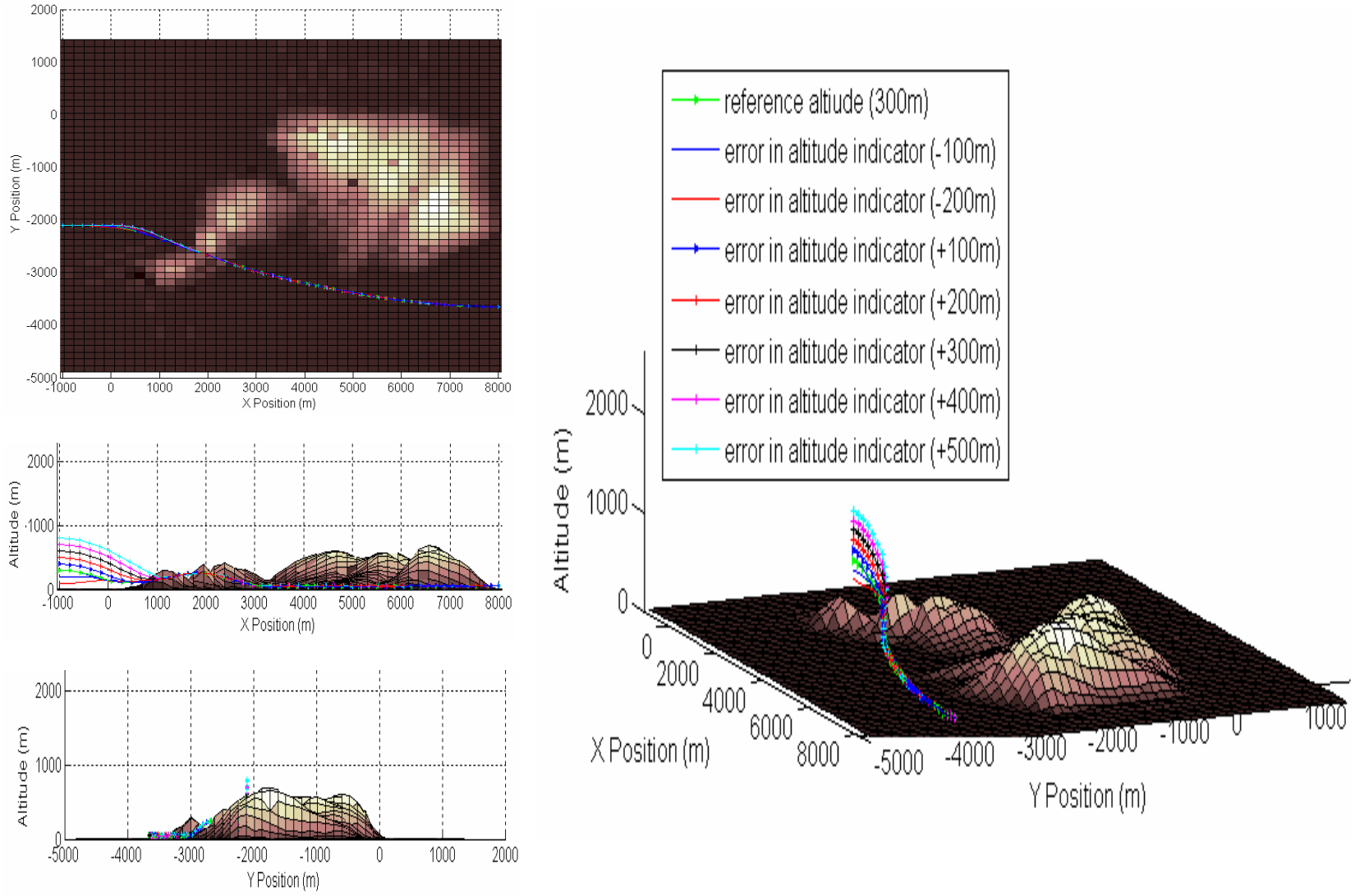


Figure 6.6: Error in altitude indicator for terrain 1 minimum clearance scenario

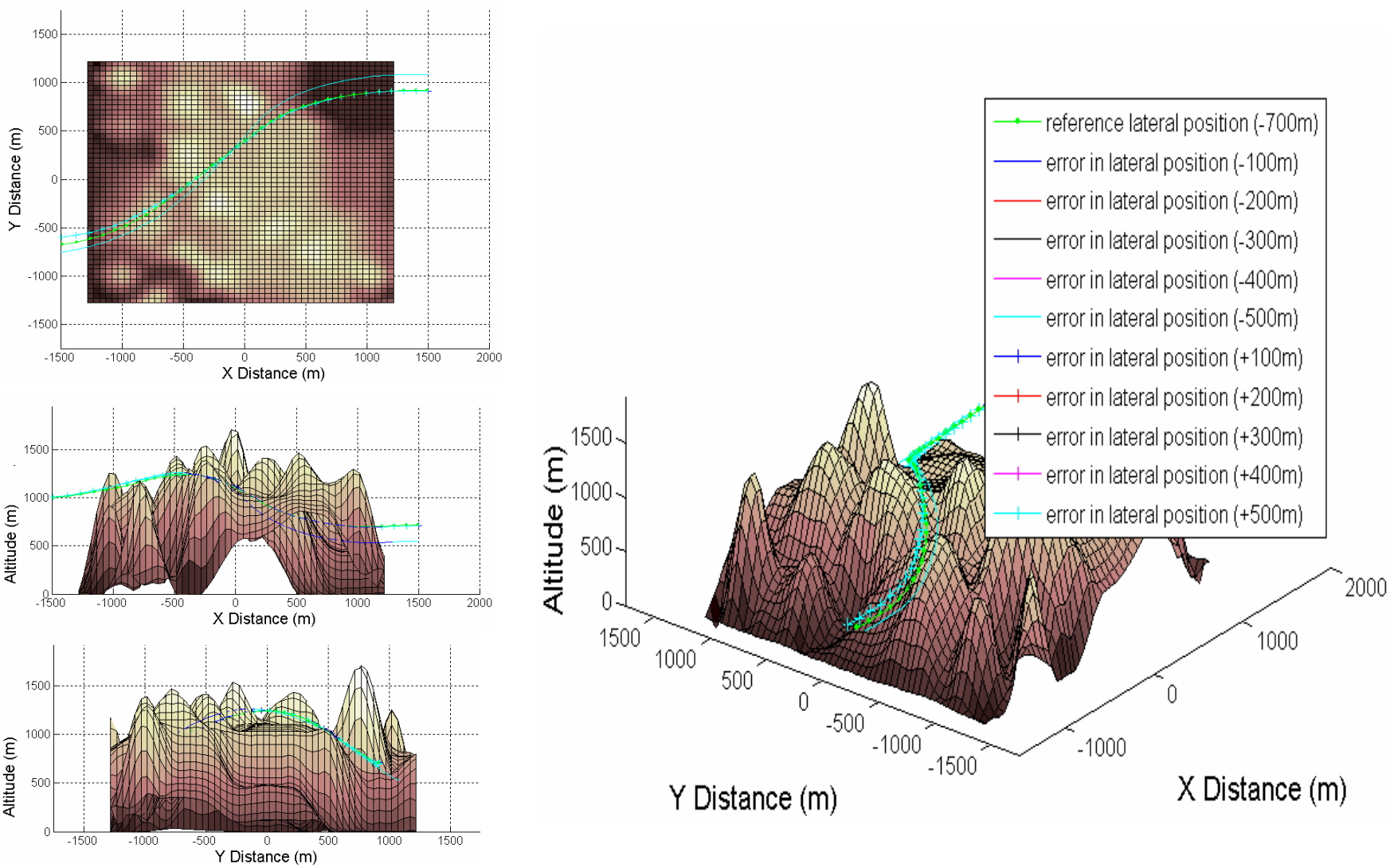


Figure 6.7: Error in lateral position for terrain2 minimum time scenario

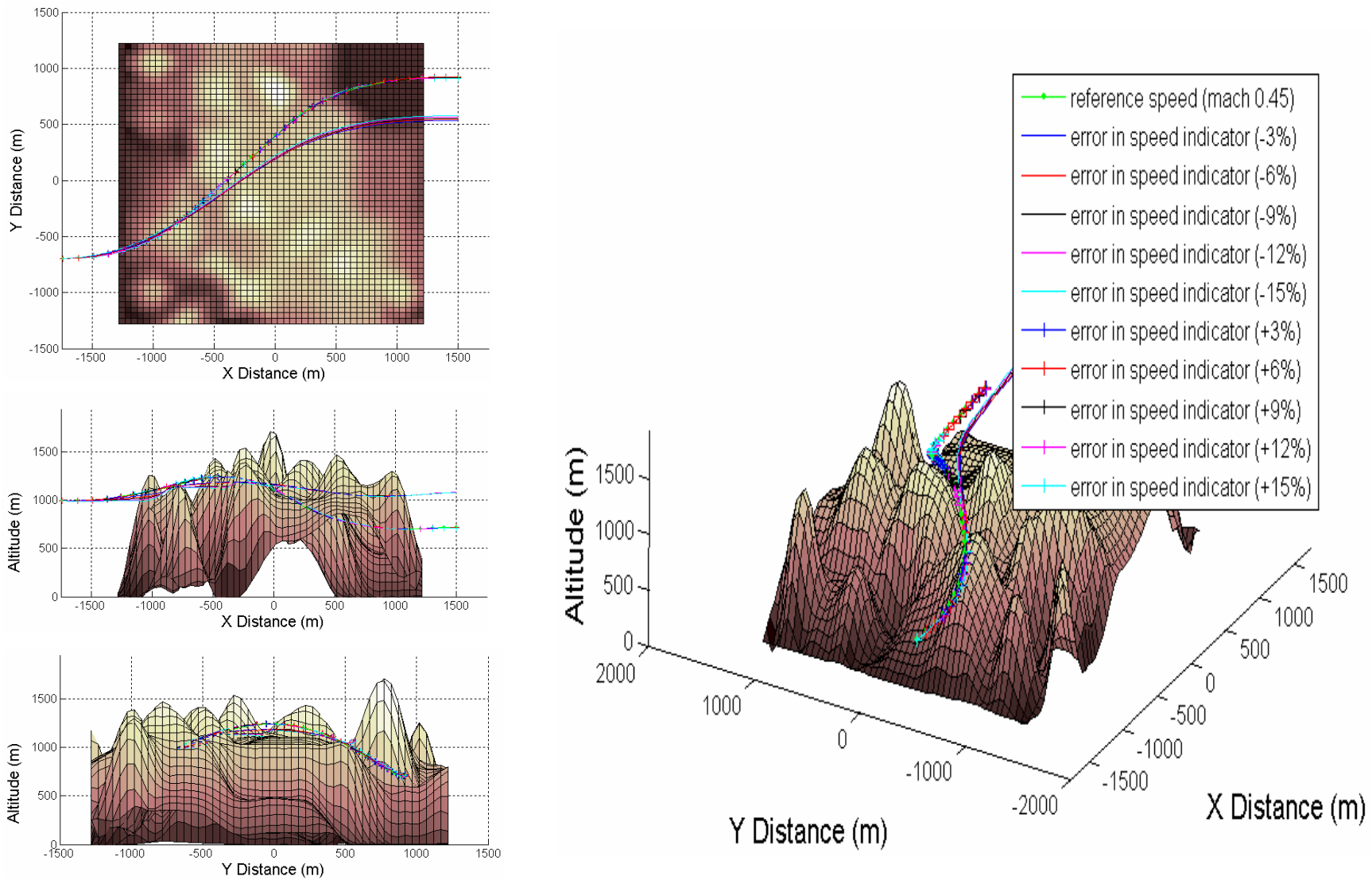


Figure 6.8: Error in speed indicator for terrain 2 minimum time scenario

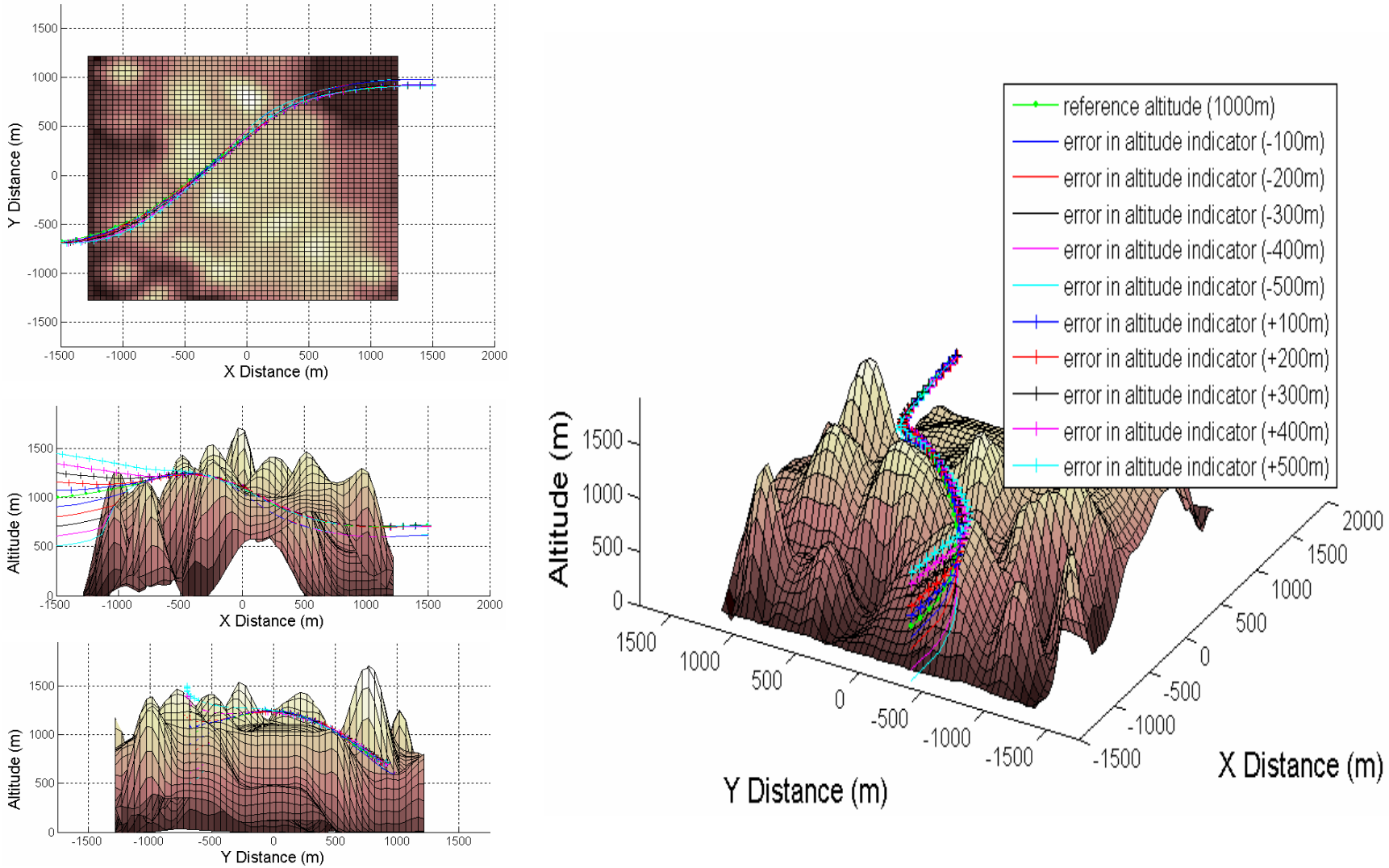


Figure 6.9: Error in altitude indicator for terrain 2 minimum time scenario

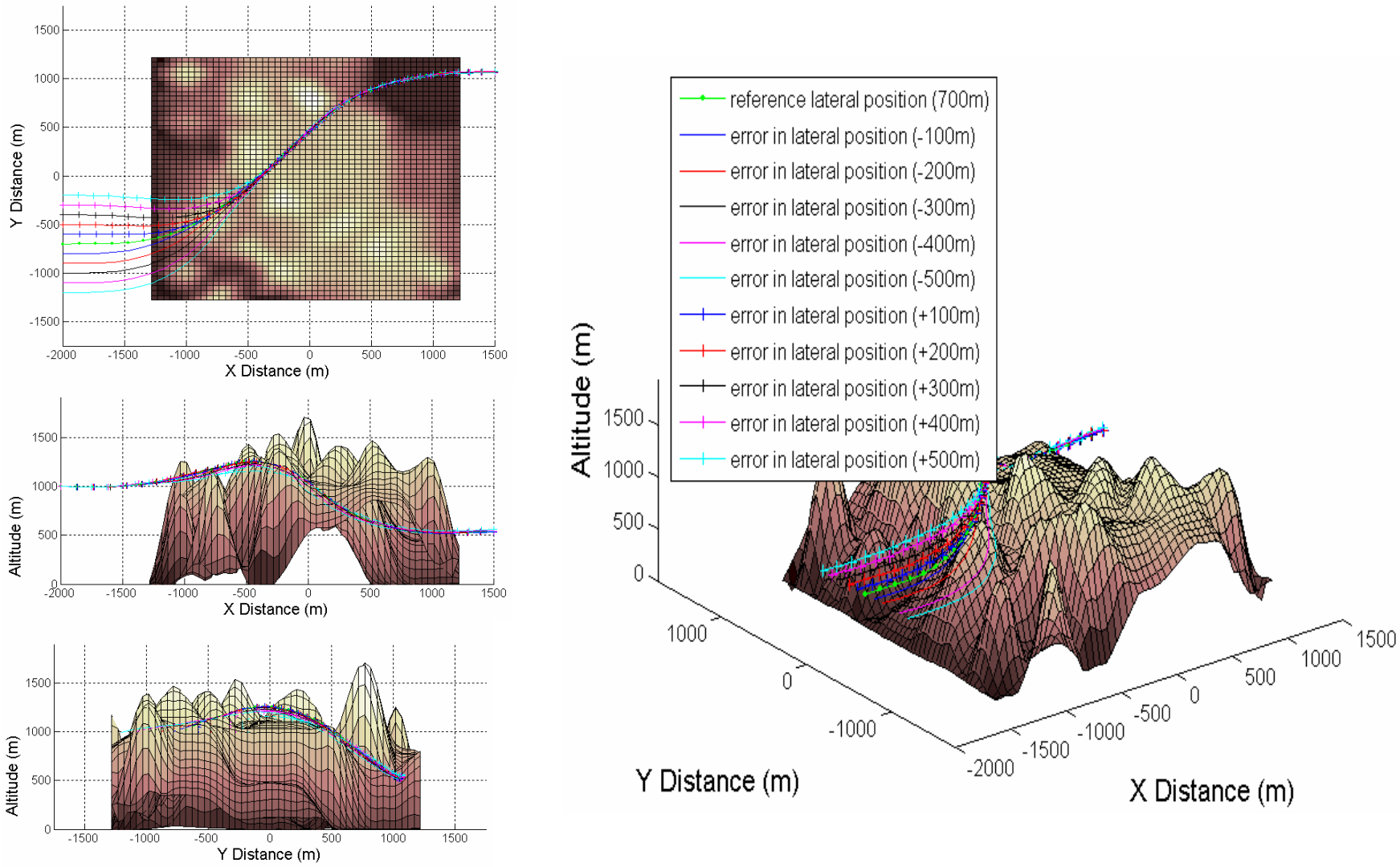


Figure 6.10: Error in lateral position for terrain 2 minimum time scenario

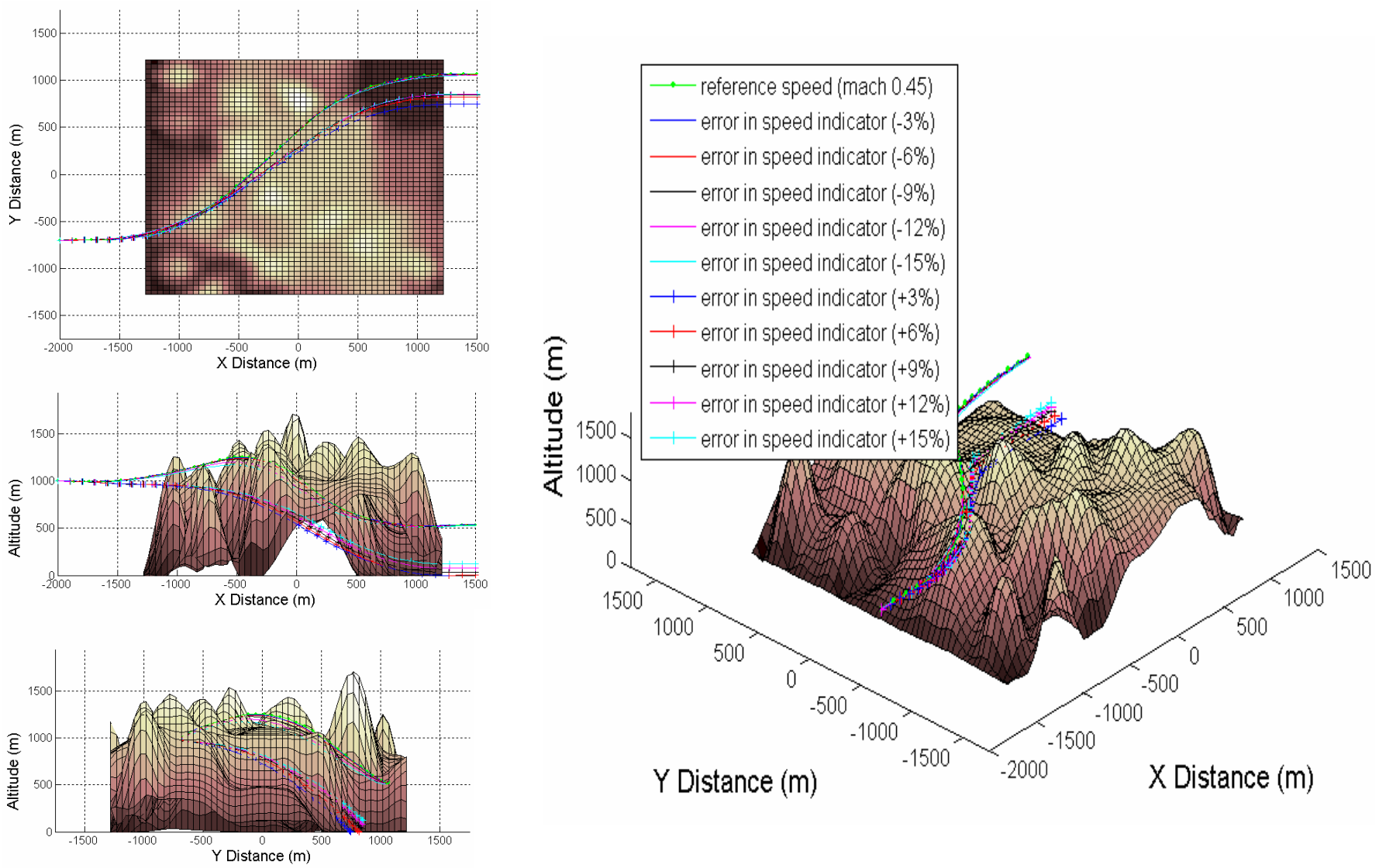


Figure 6.11 Error in speed indicator for terrain 2 minimum time scenario

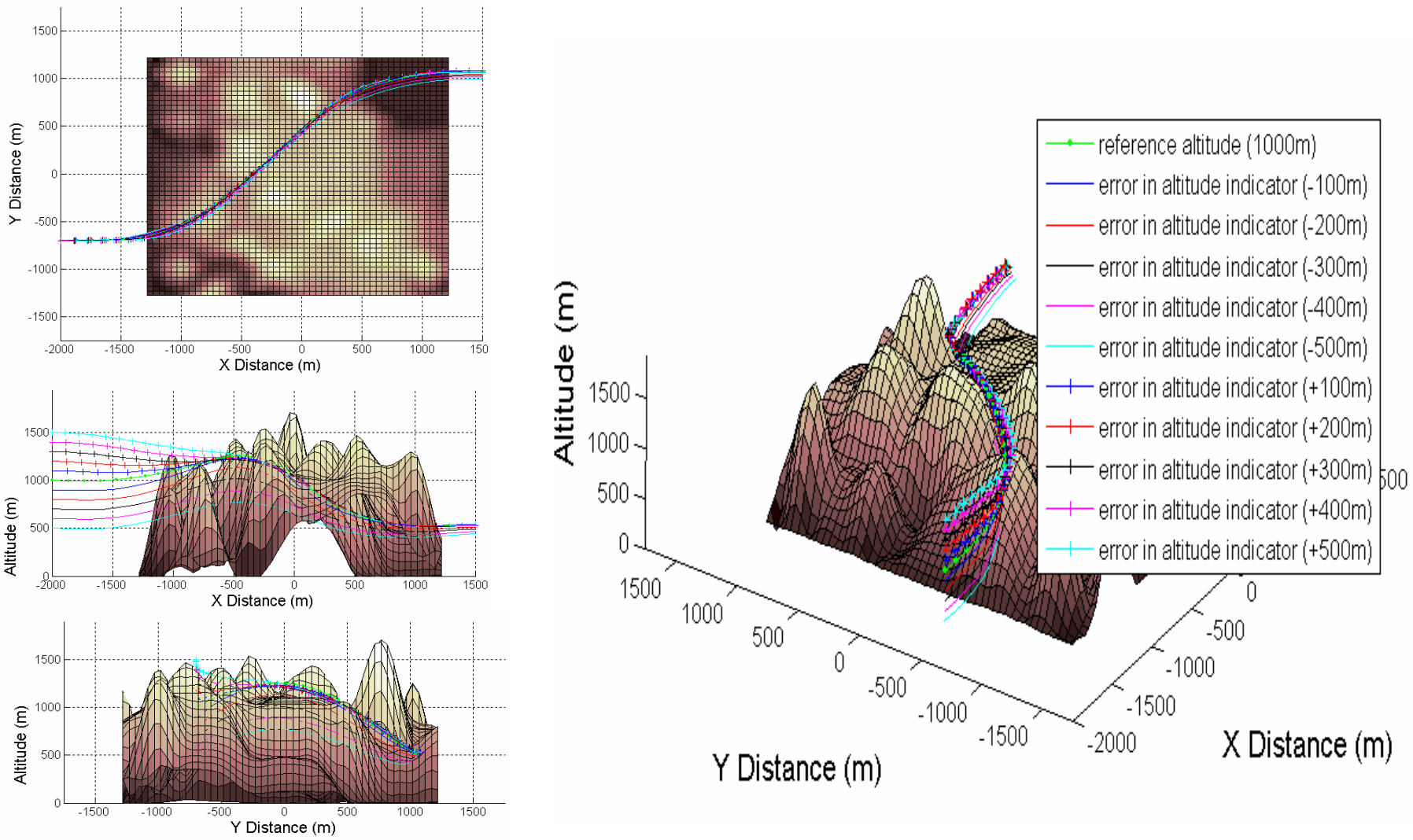


Figure 6.12: Error in altitude indicator for terrain 2 minimum clearance scenario

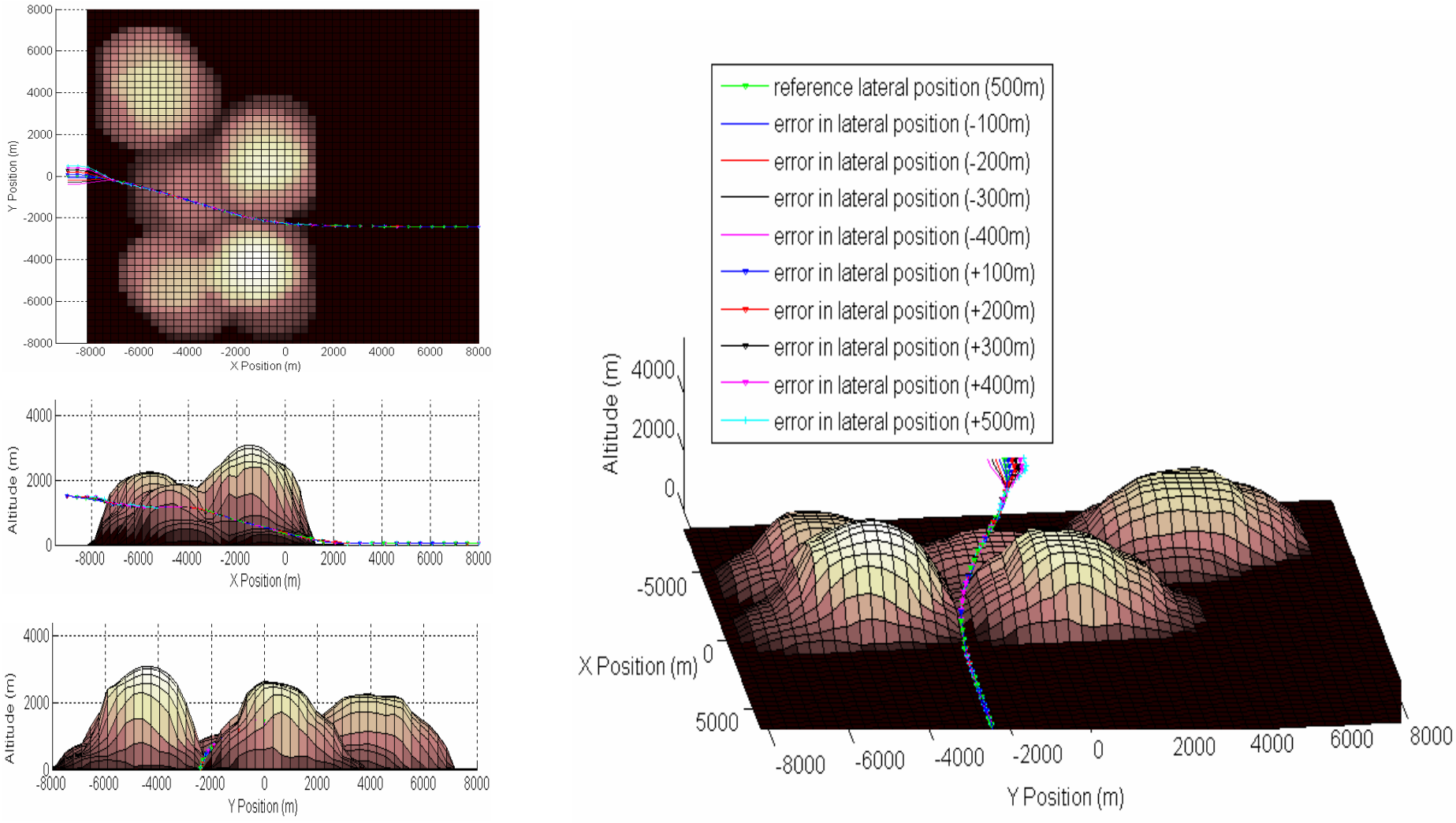


Figure 6.13: Error in lateral position for terrain 3 minimum time scenario

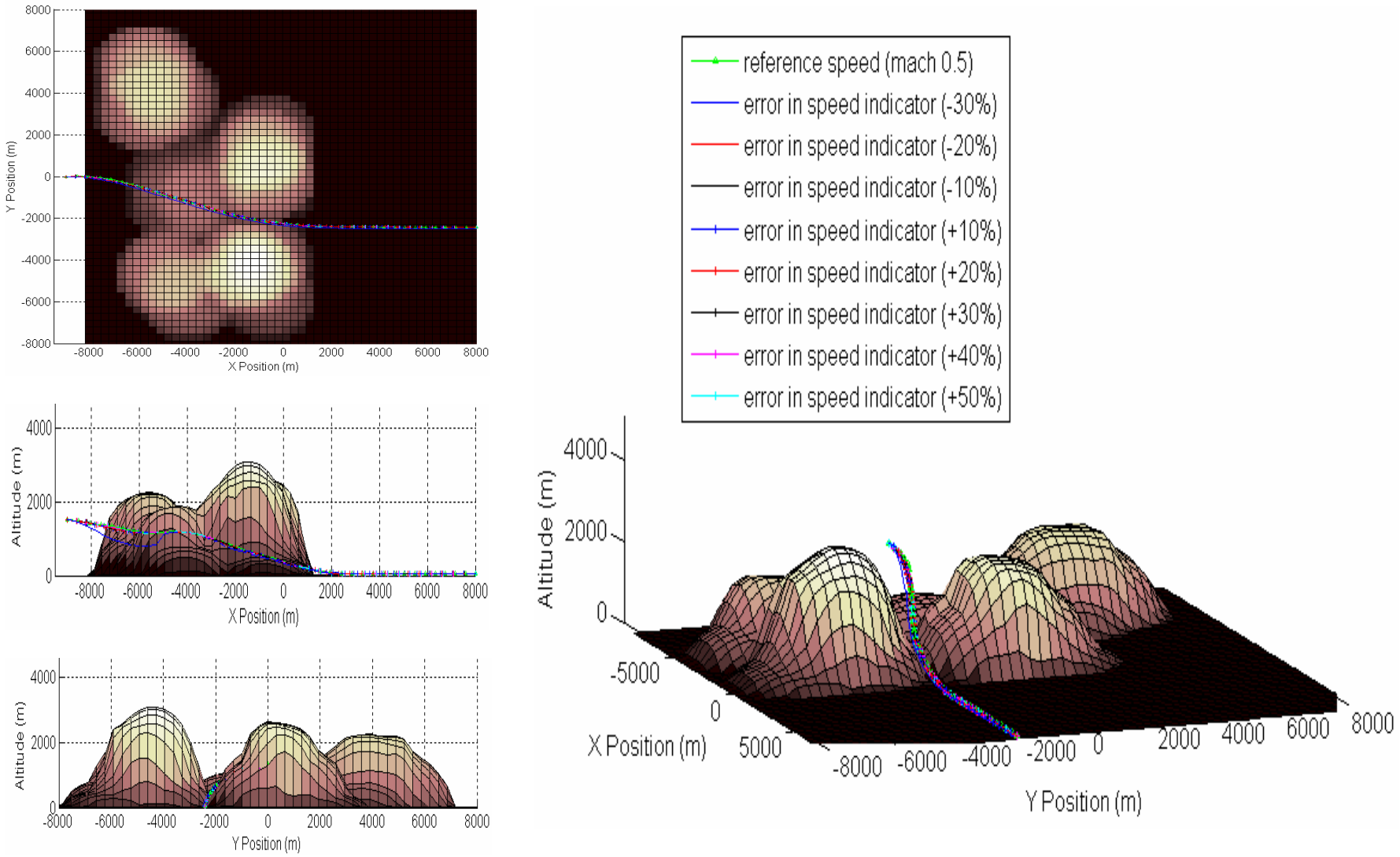


Figure 6.14: Error in speed indicator for terrain 3 minimum time scenario

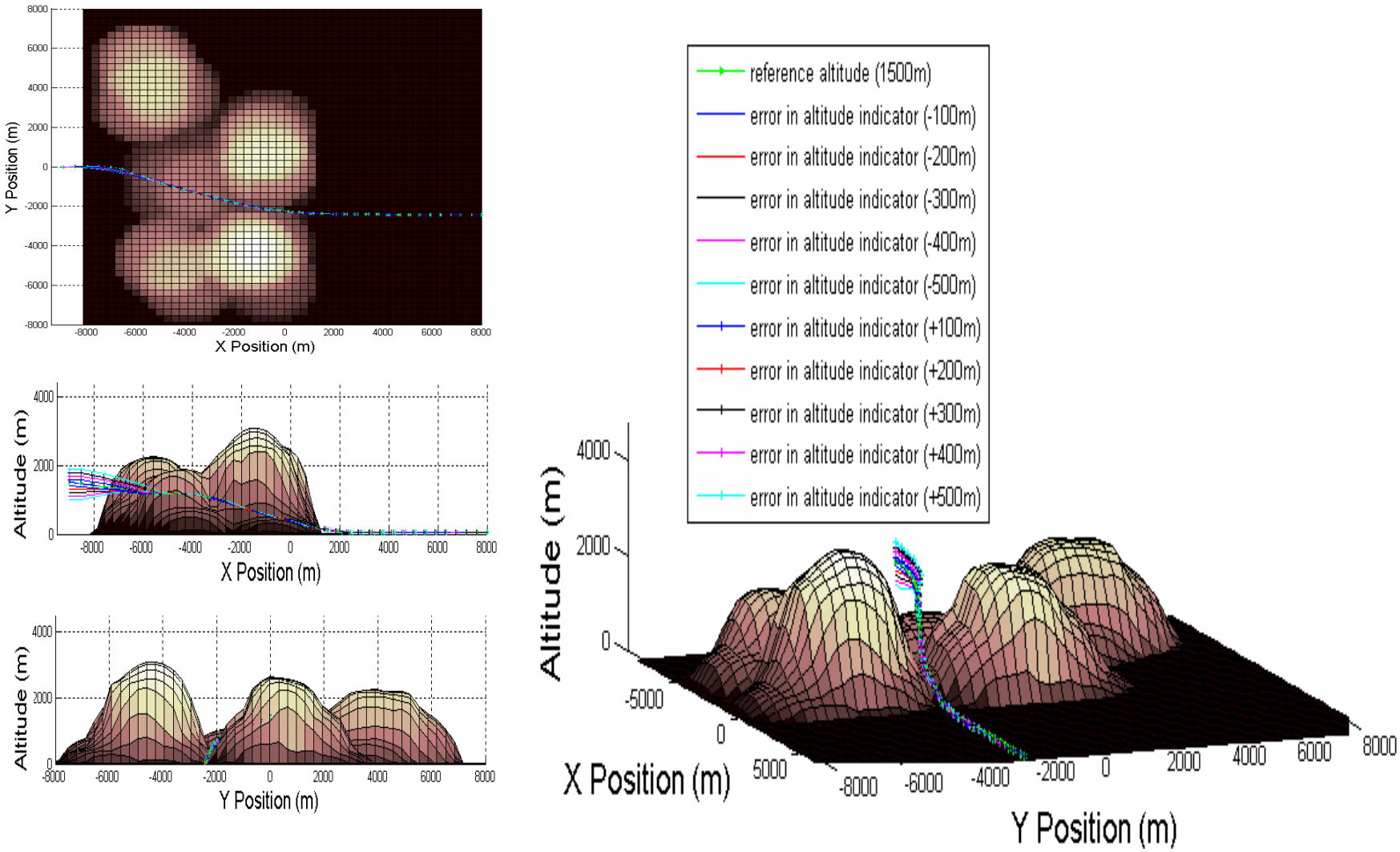


Figure 6.15: Error in altitude indicator for terrain 3 minimum clearance scenario

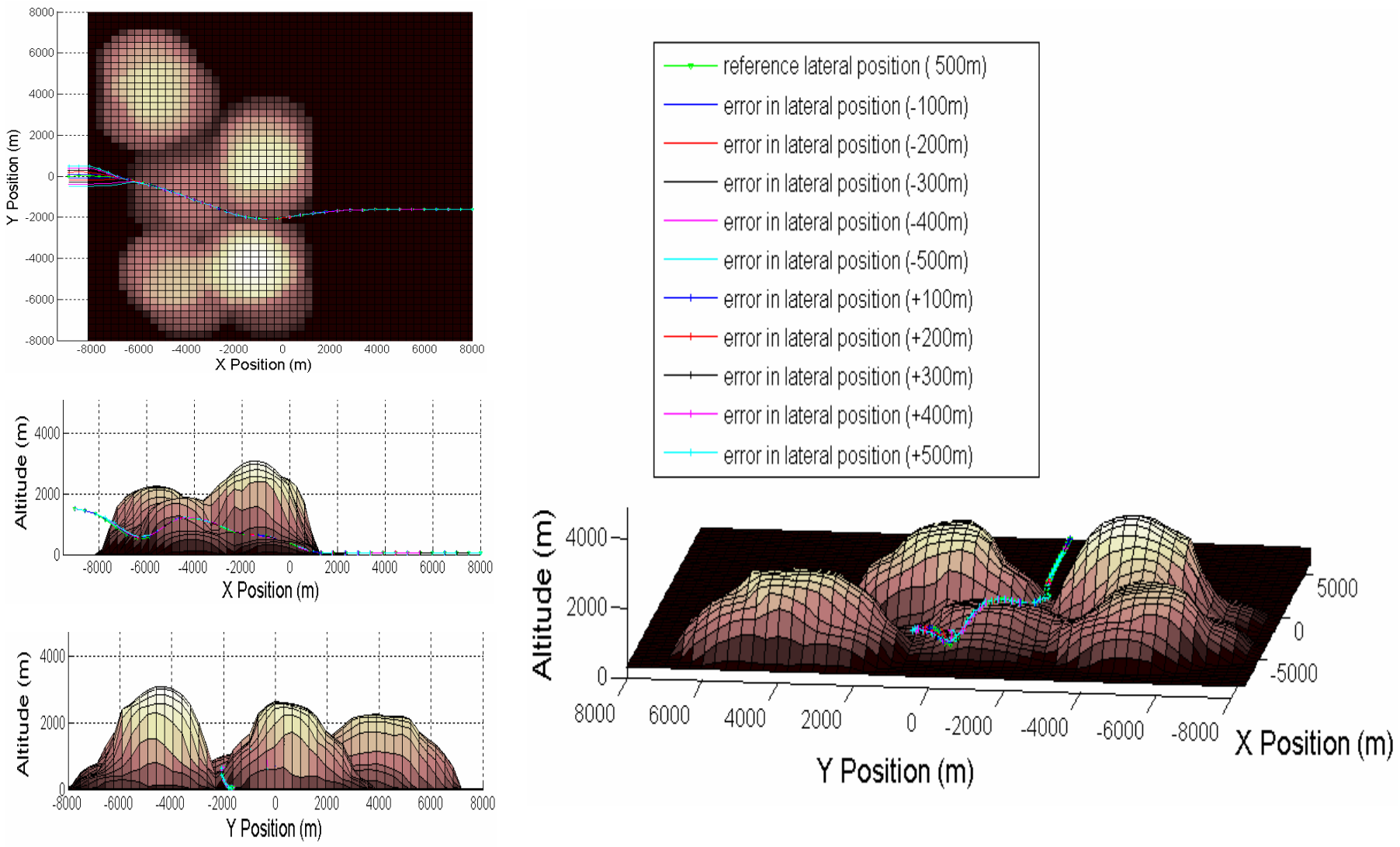


Figure 6.16: Error in lateral position for terrain 3 minimum clearance scenario

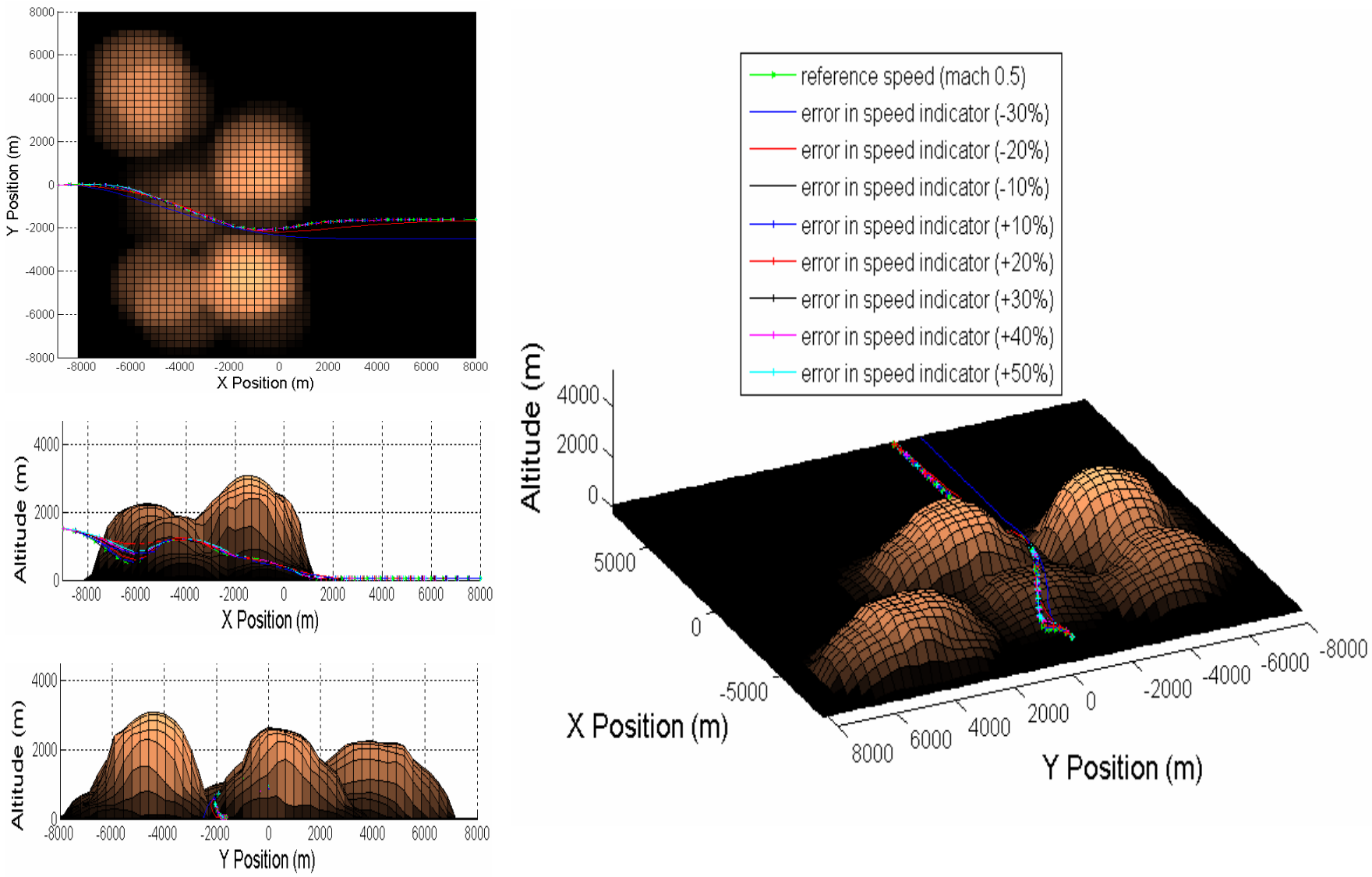


Figure 6.17: Error in speed indicator for terrain 3 minimum clearance scenario

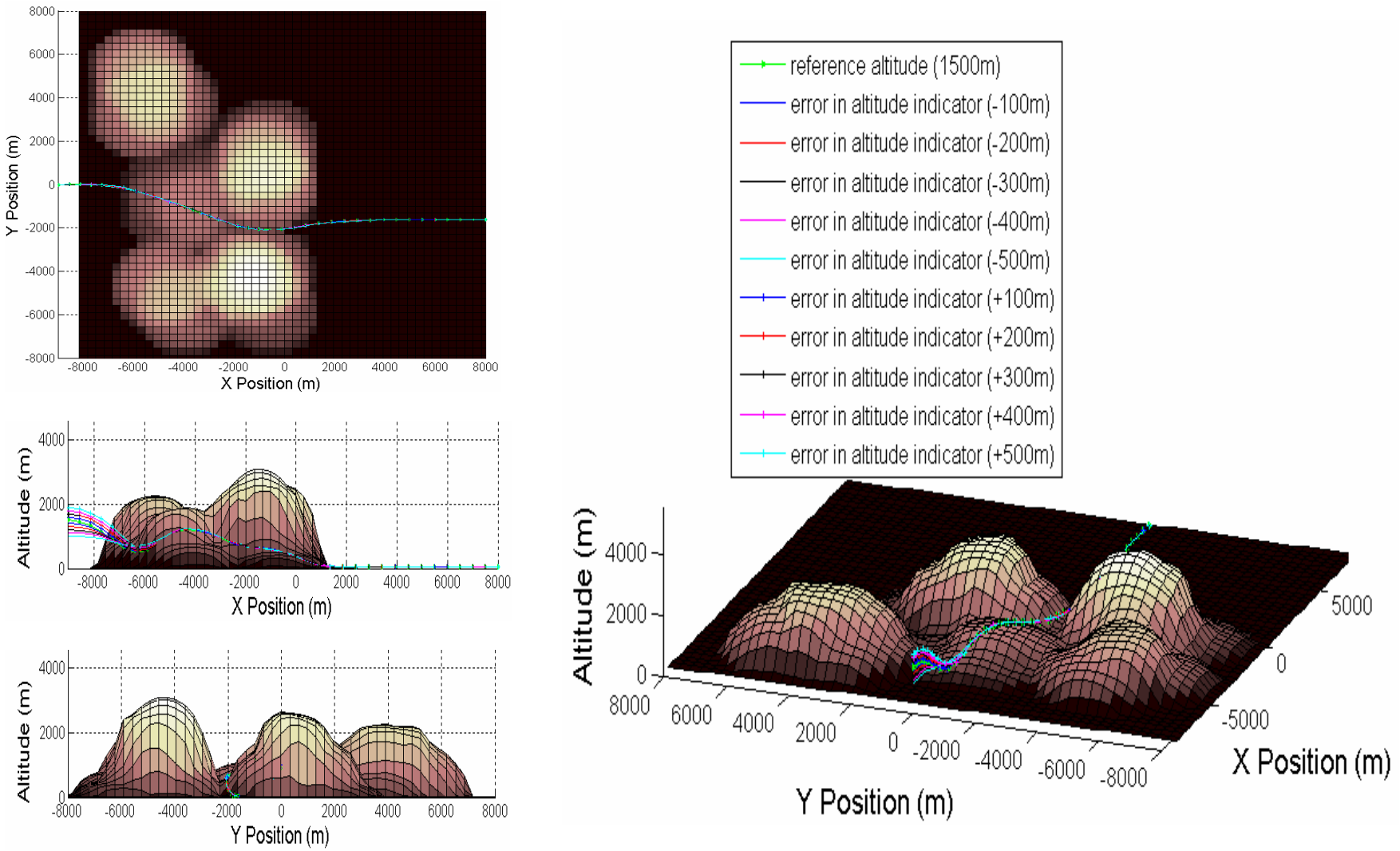


Figure 6.18: Error in altitude indicator for terrain 3 minimum clearance scenario

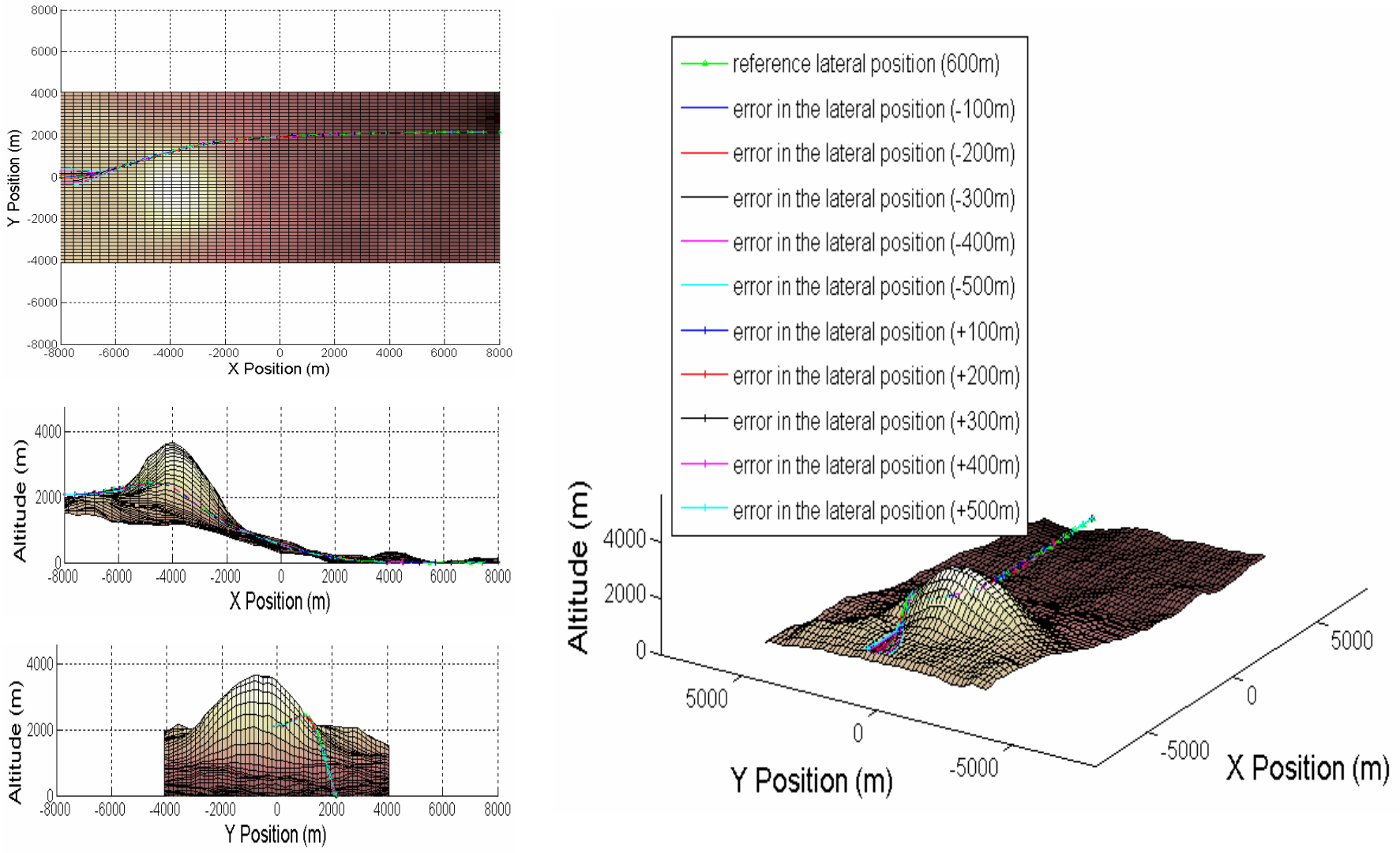


Figure 6.19: Error in lateral position for terrain 4 minimum time scenario

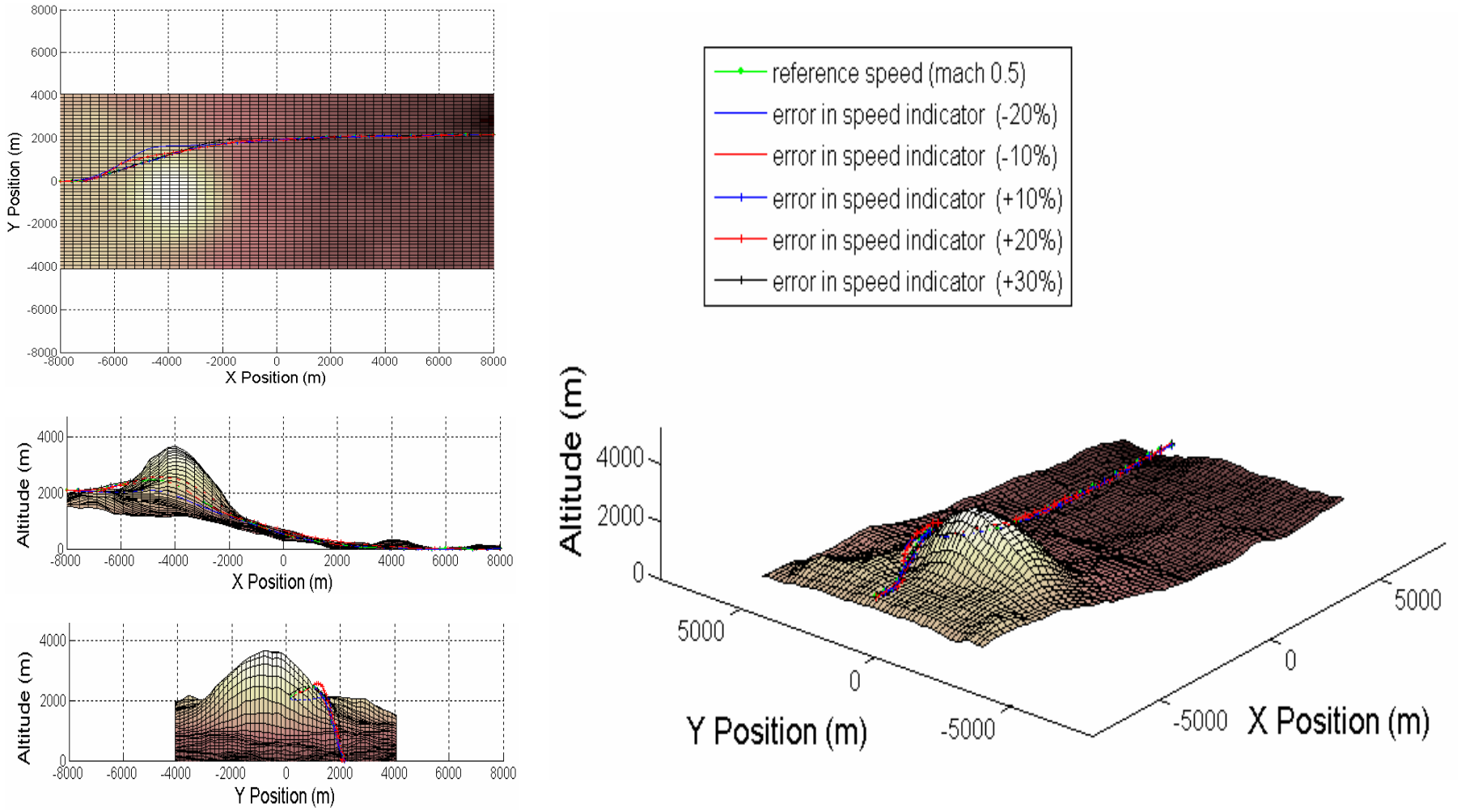


Figure 6.20: Error in speed indicator for terrain 4 minimum time scenario

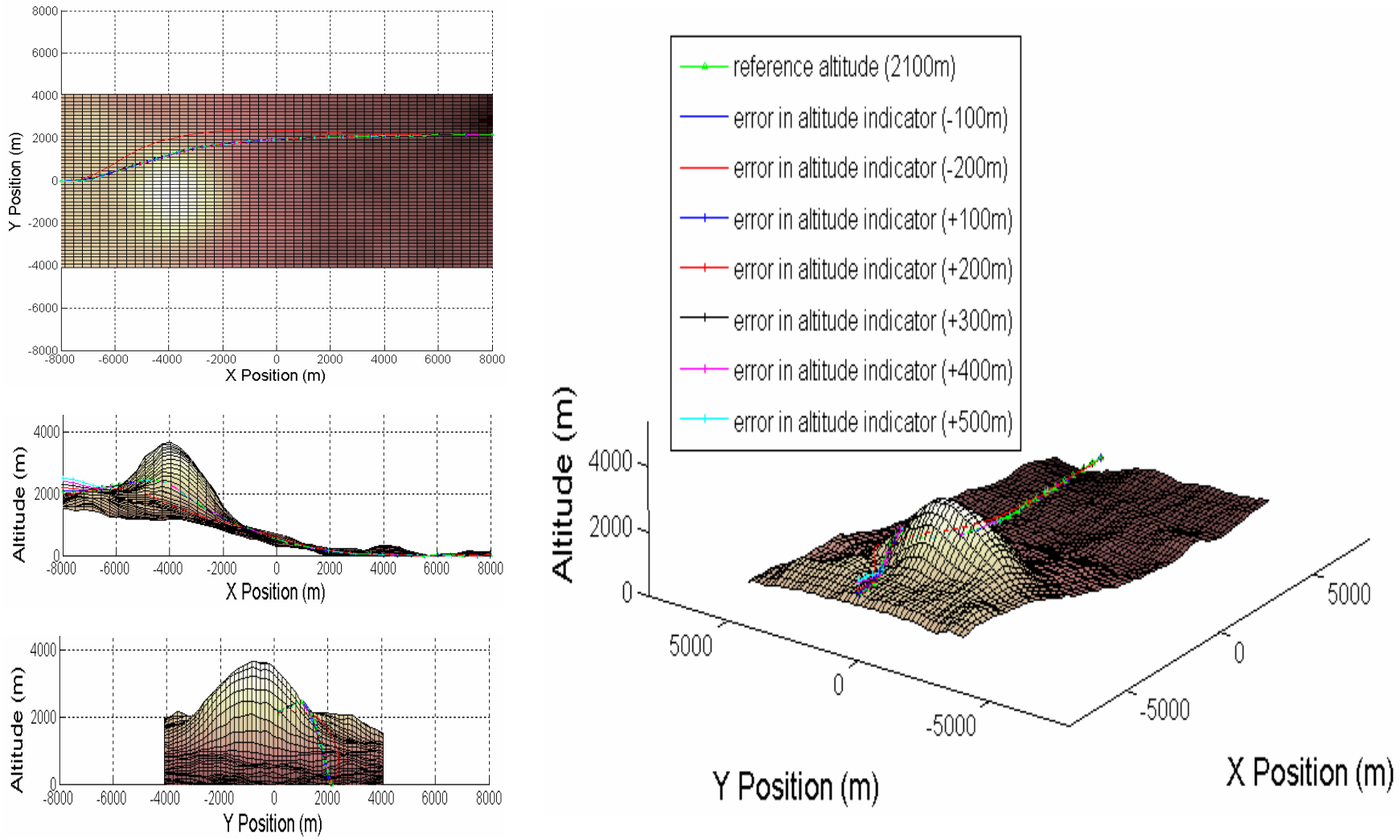


Figure 6.21: Error in altitude indicator for terrain4 minimum time scenario

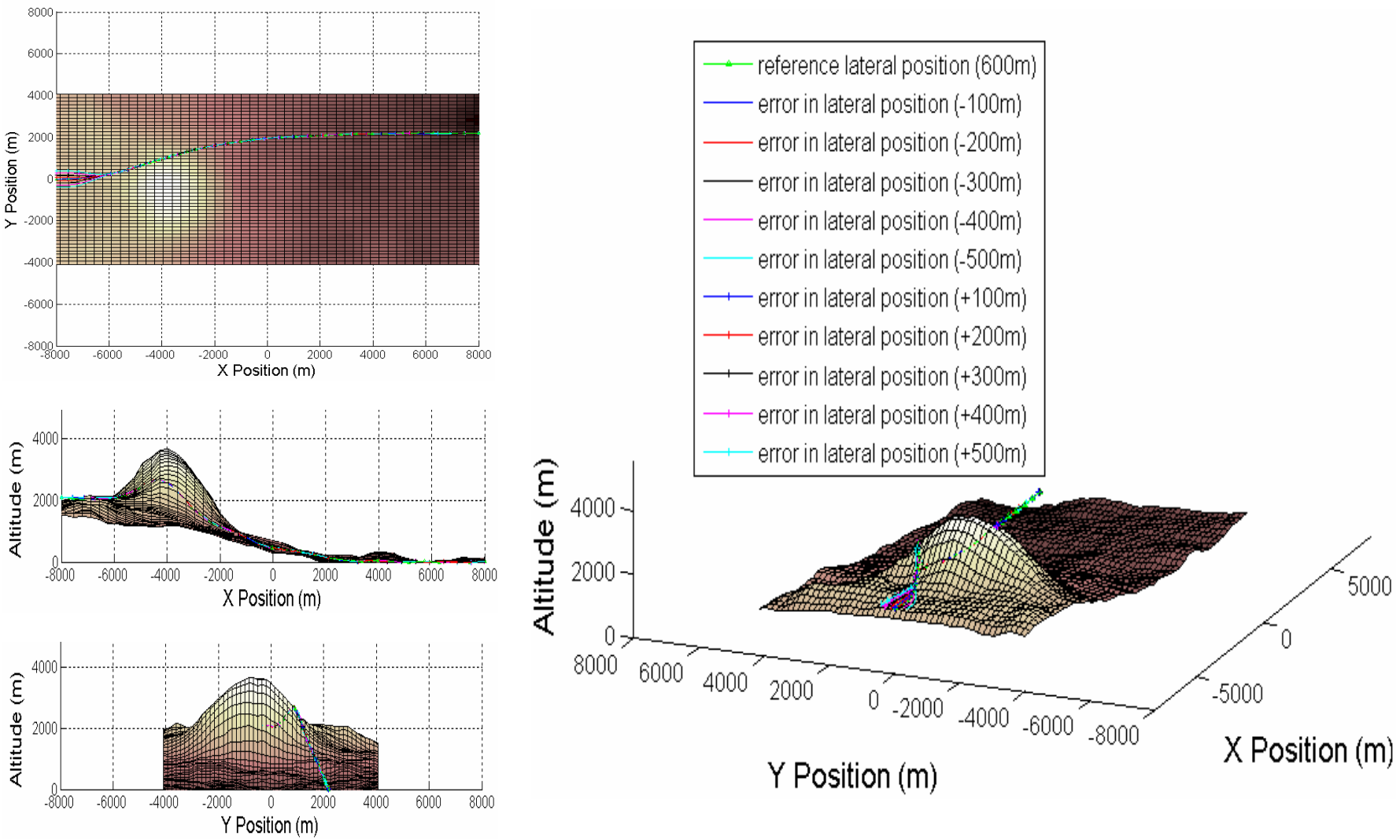


Figure 6.22: Error in lateral position for terrain 4 minimum clearance scenario

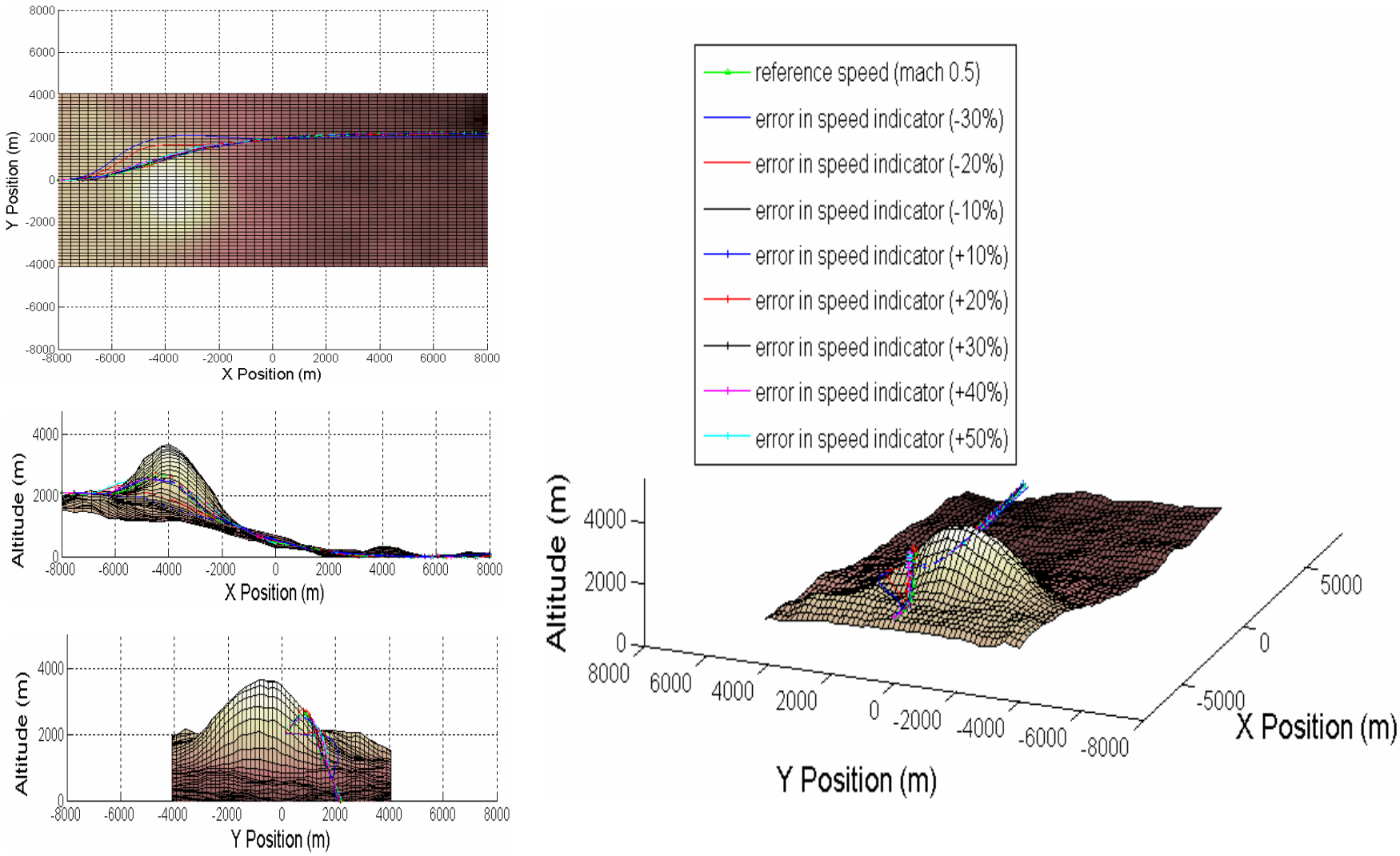


Figure 6.23: Error in speed indicator for terrain 4 minimum clearance scenario

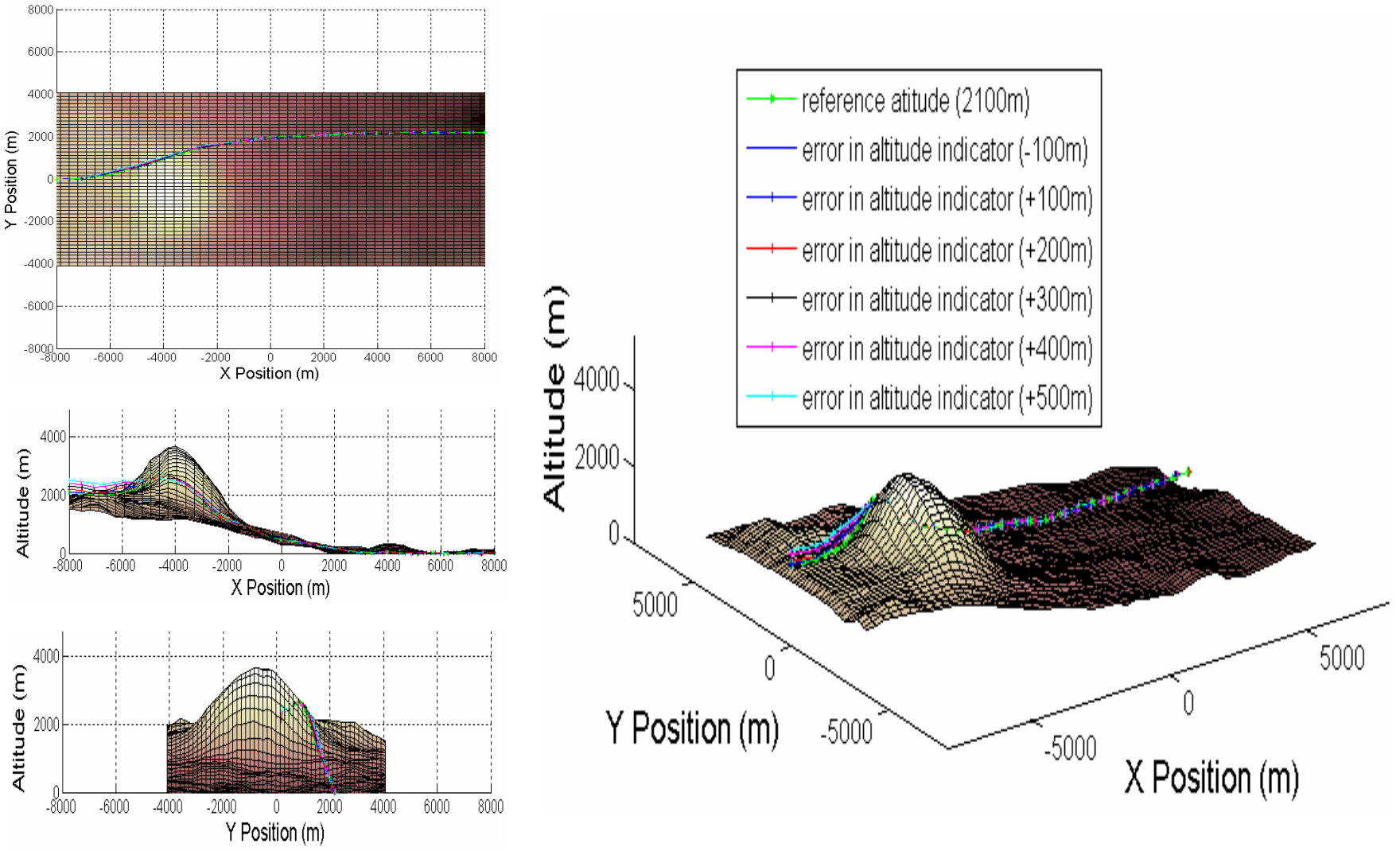


Figure 6.24: Error in altitude indicator for terrain 4 minimum clearance scenario

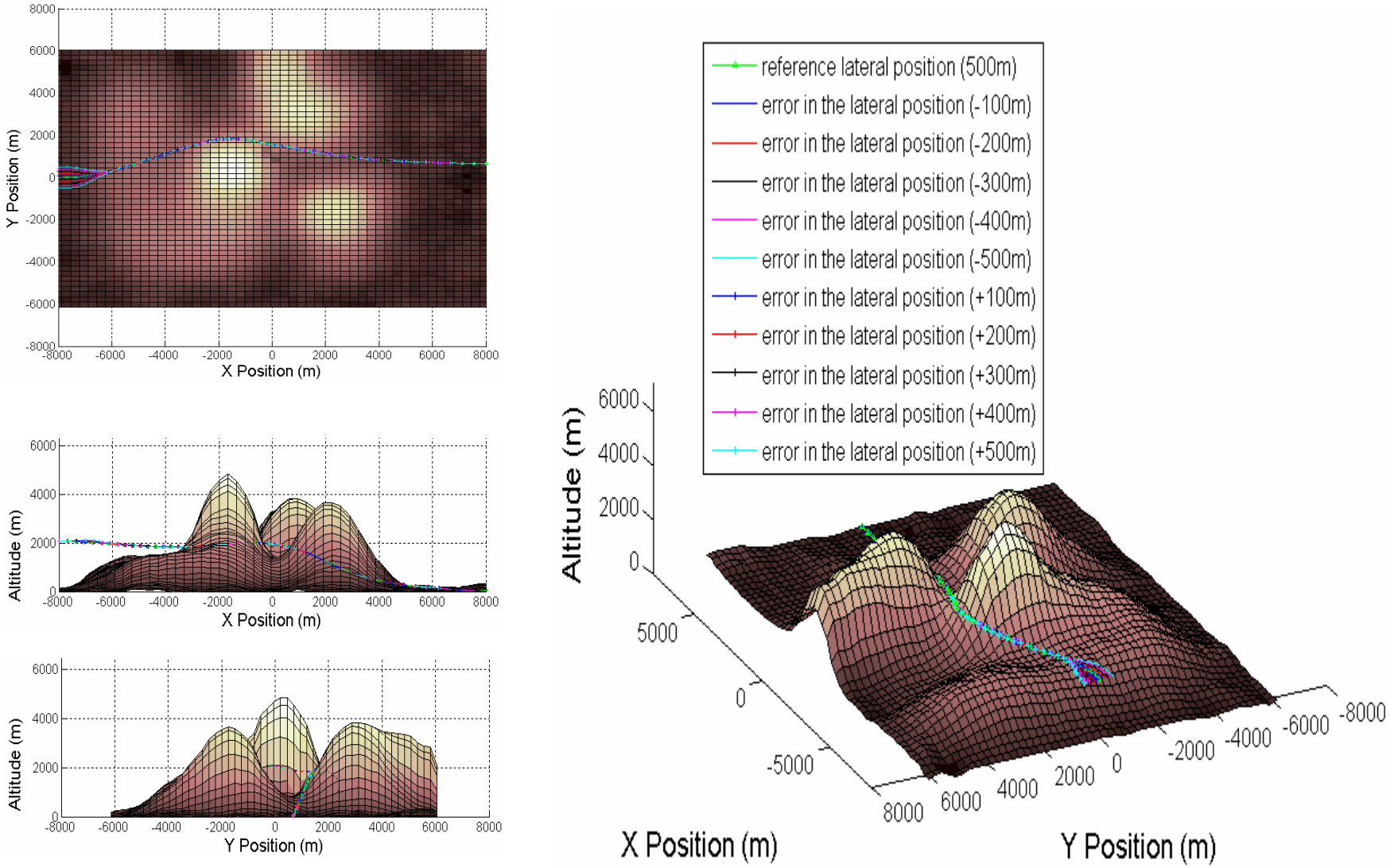


Figure 6.25: Error in lateral position for terrain 5 minimum time scenario

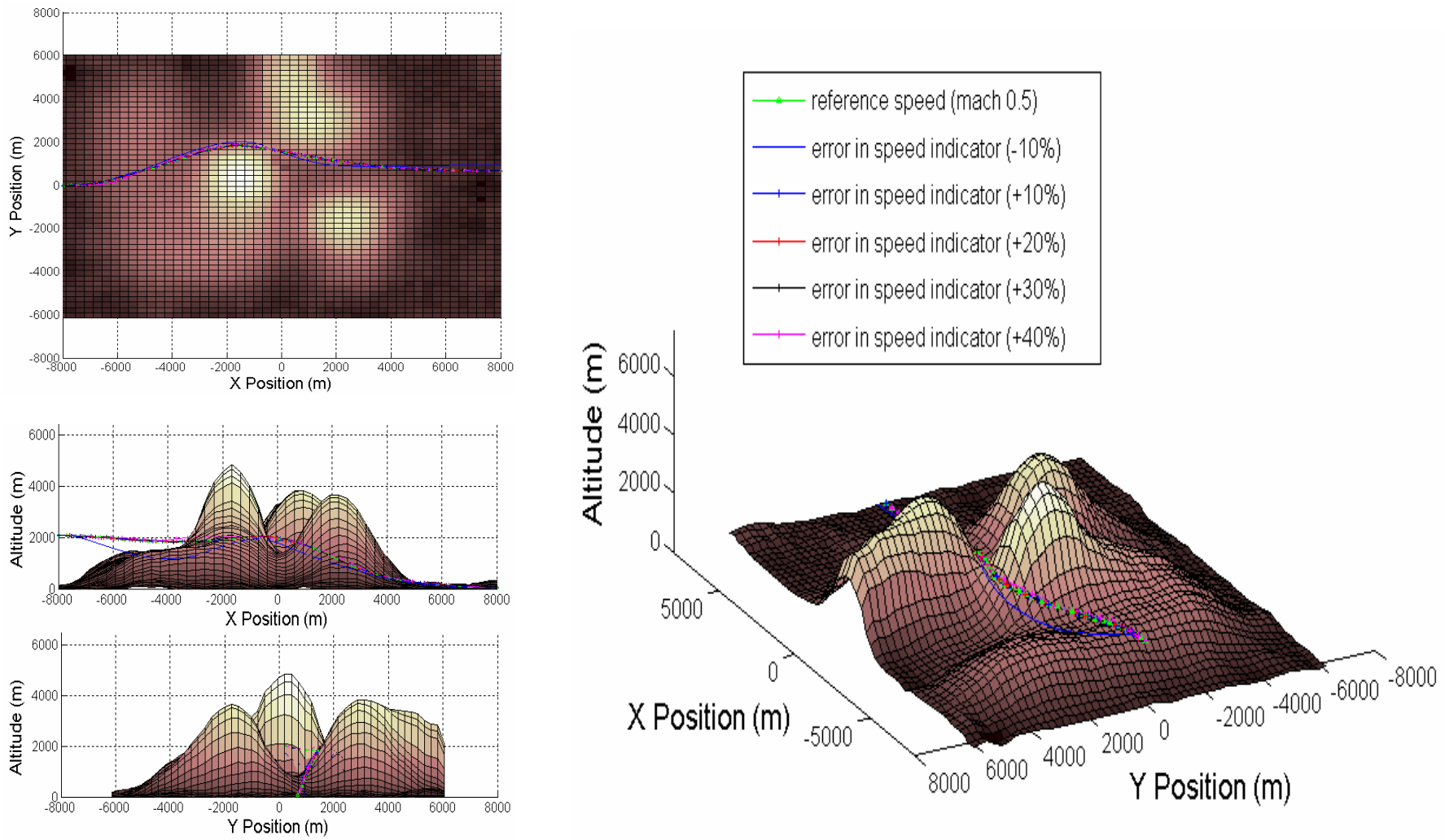
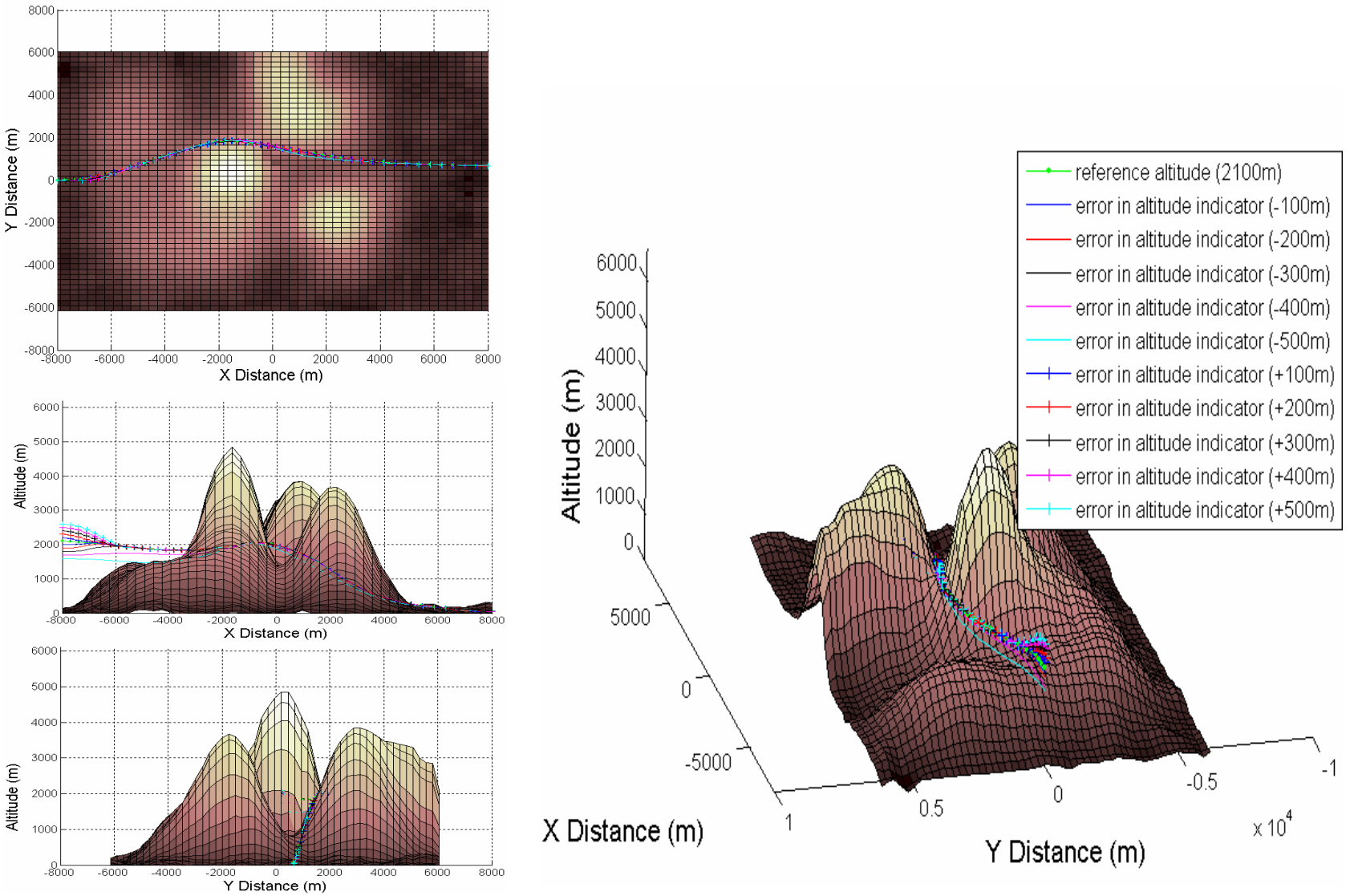


Figure 6.26: Error in speed indicator for terrain 5 minimum time scenario



]

Figure 6.27: Error in altitude indicator for terrain 5 minimum time scenario

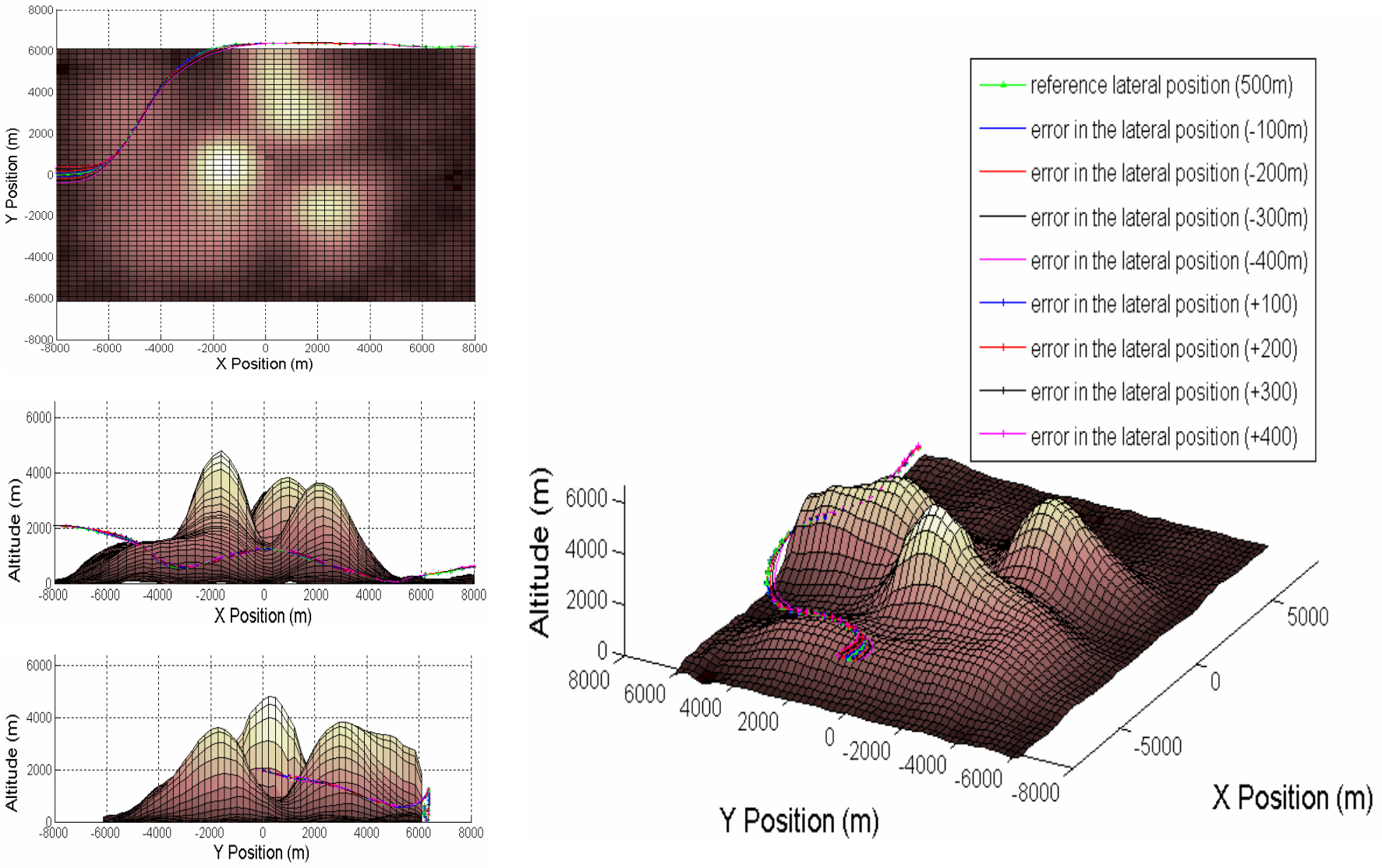


Figure 6.28: Error in lateral position for terrain 5 minimum clearance scenario

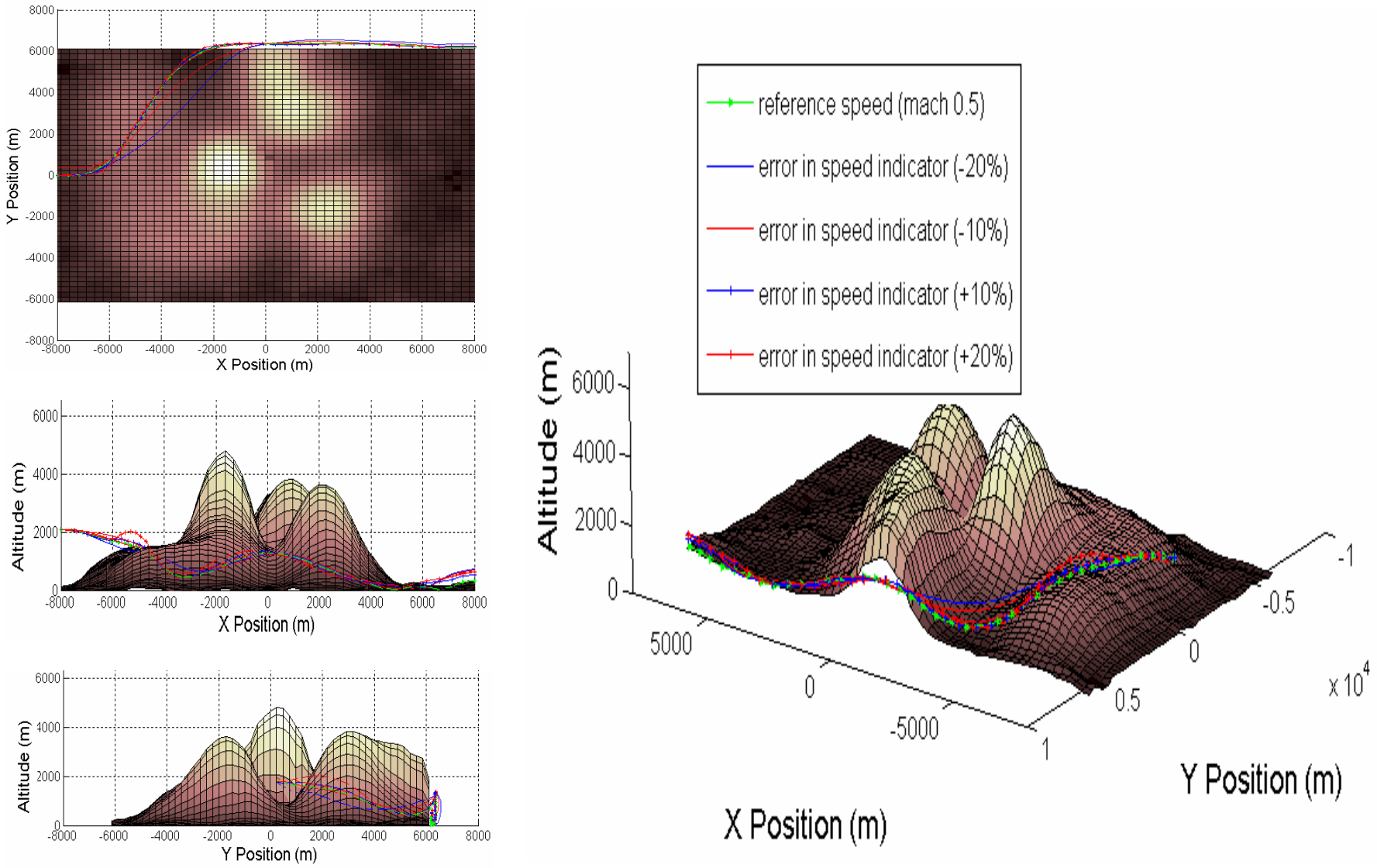


Figure 6.29: Error in speed indicator for terrain 5 minimum clearance scenario

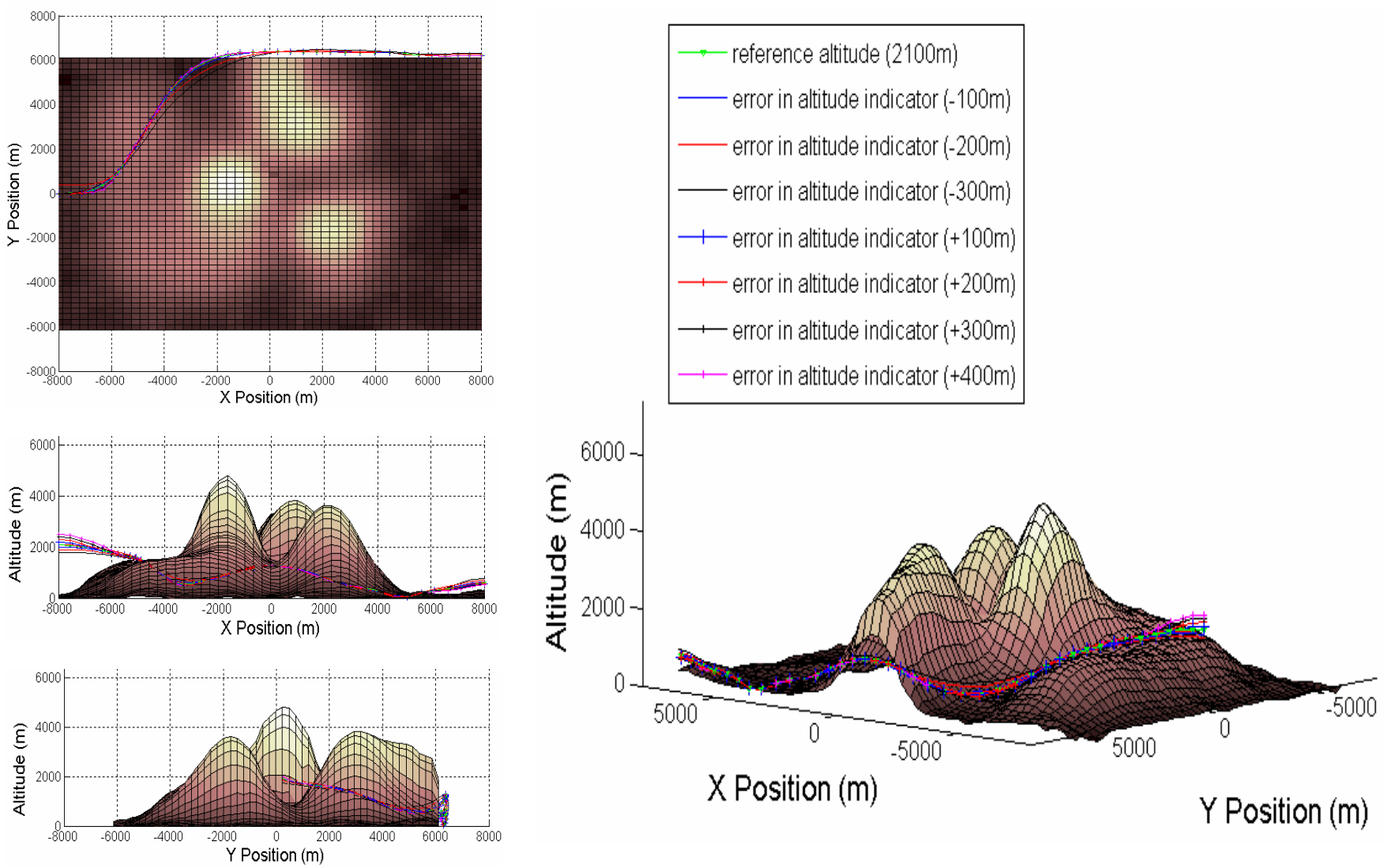


Figure 6.30: Error in altitude indicator for terrain 5 minimum clearance scenario

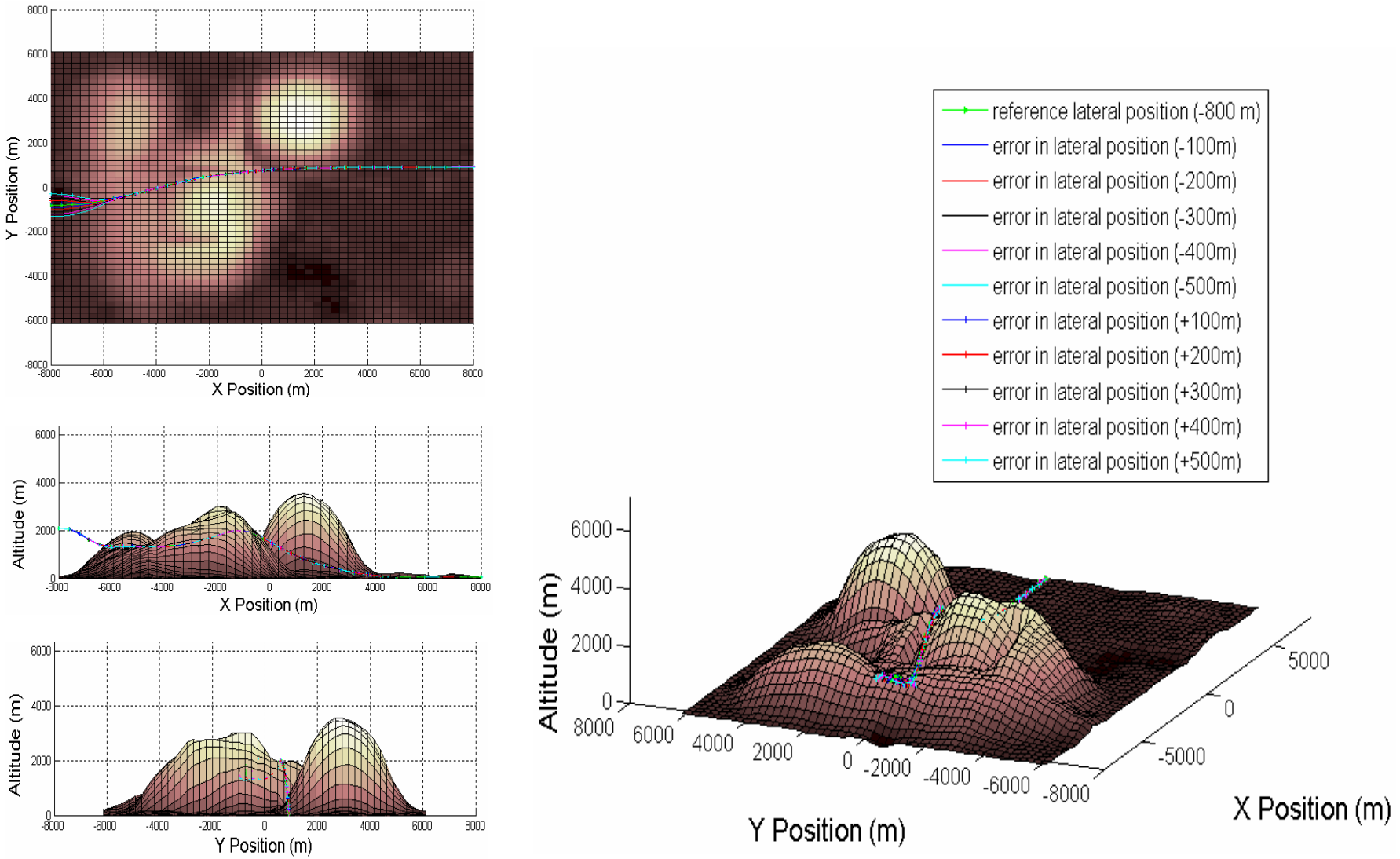


Figure 6.31: Error in lateral position for terrain 6 minimum time scenario

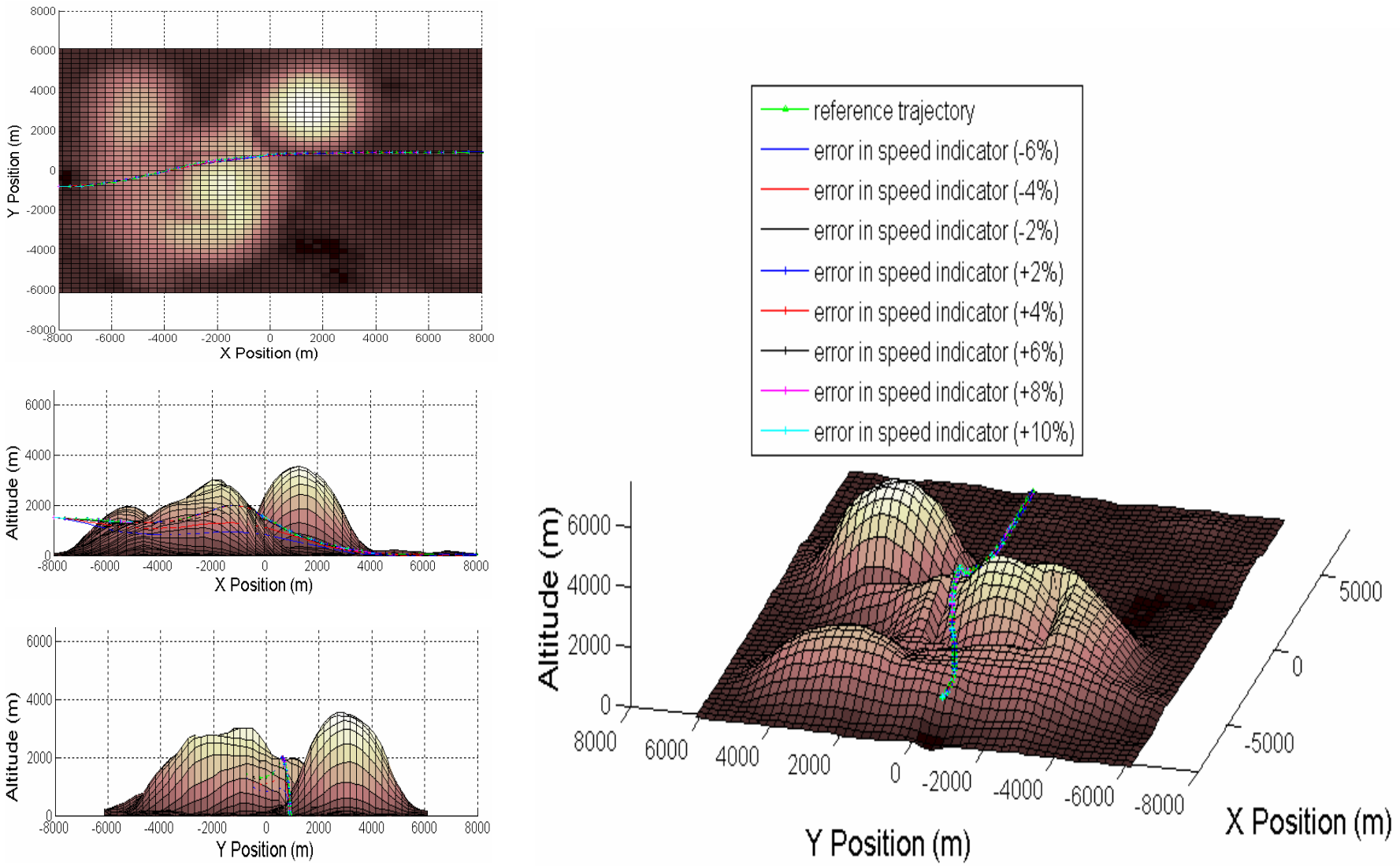


Figure 6.32: Error in speed indicator for terrain 6 minimum time scenario

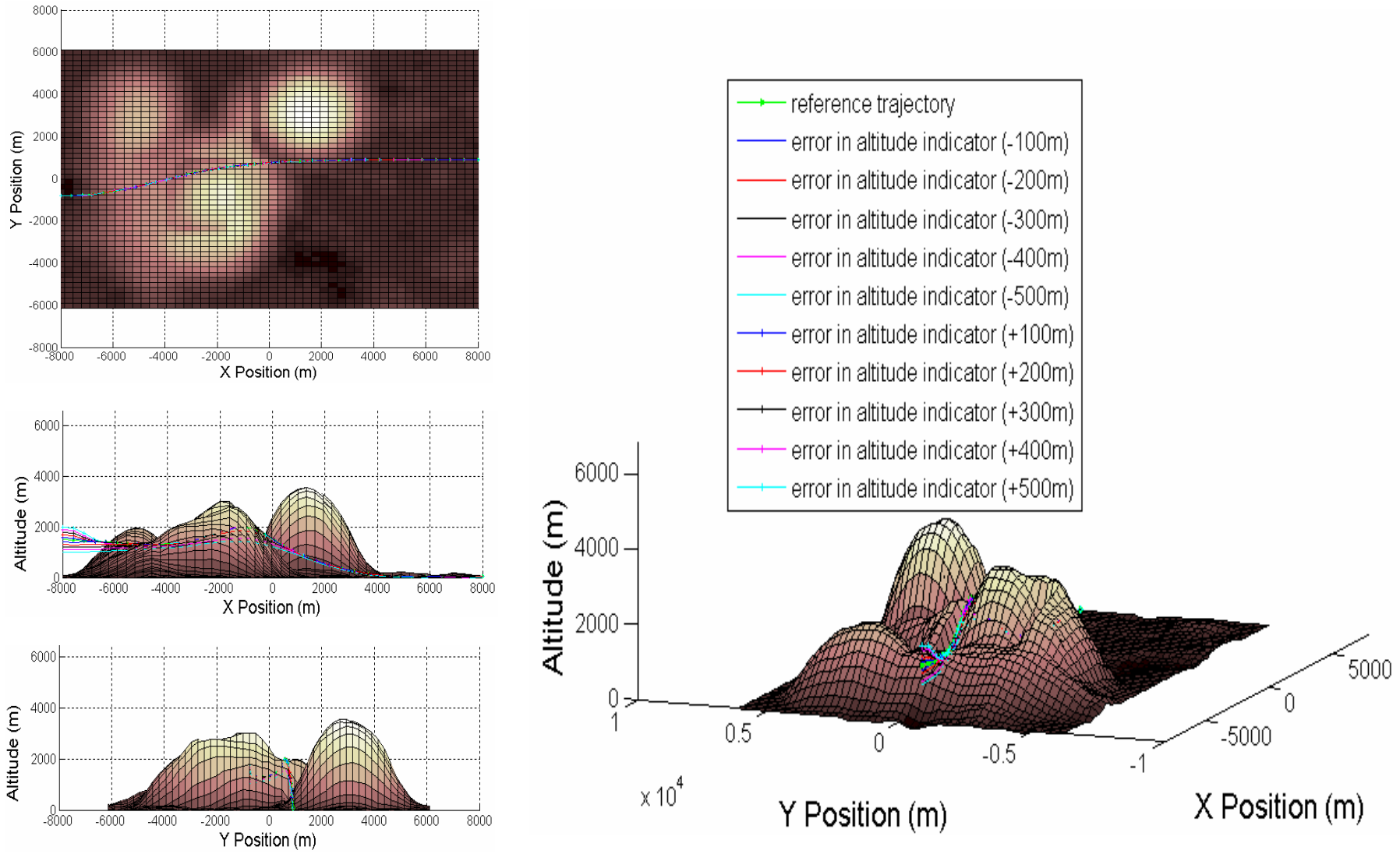


Figure 6.33: Error in altitude indicator for terrain 6 minimum time scenario

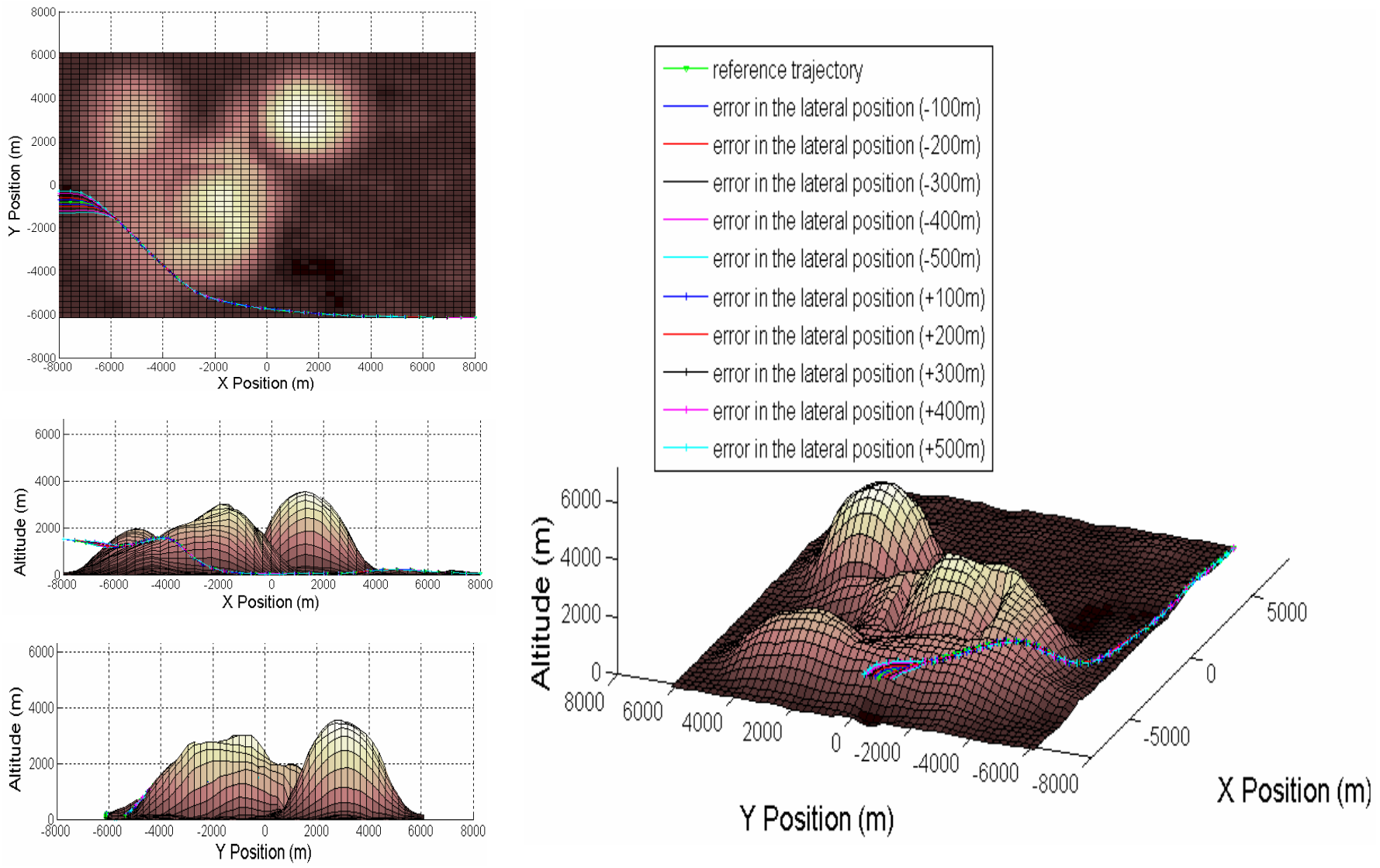


Figure 6.34: Error in lateral position for terrain 6 minimum clearance scenario

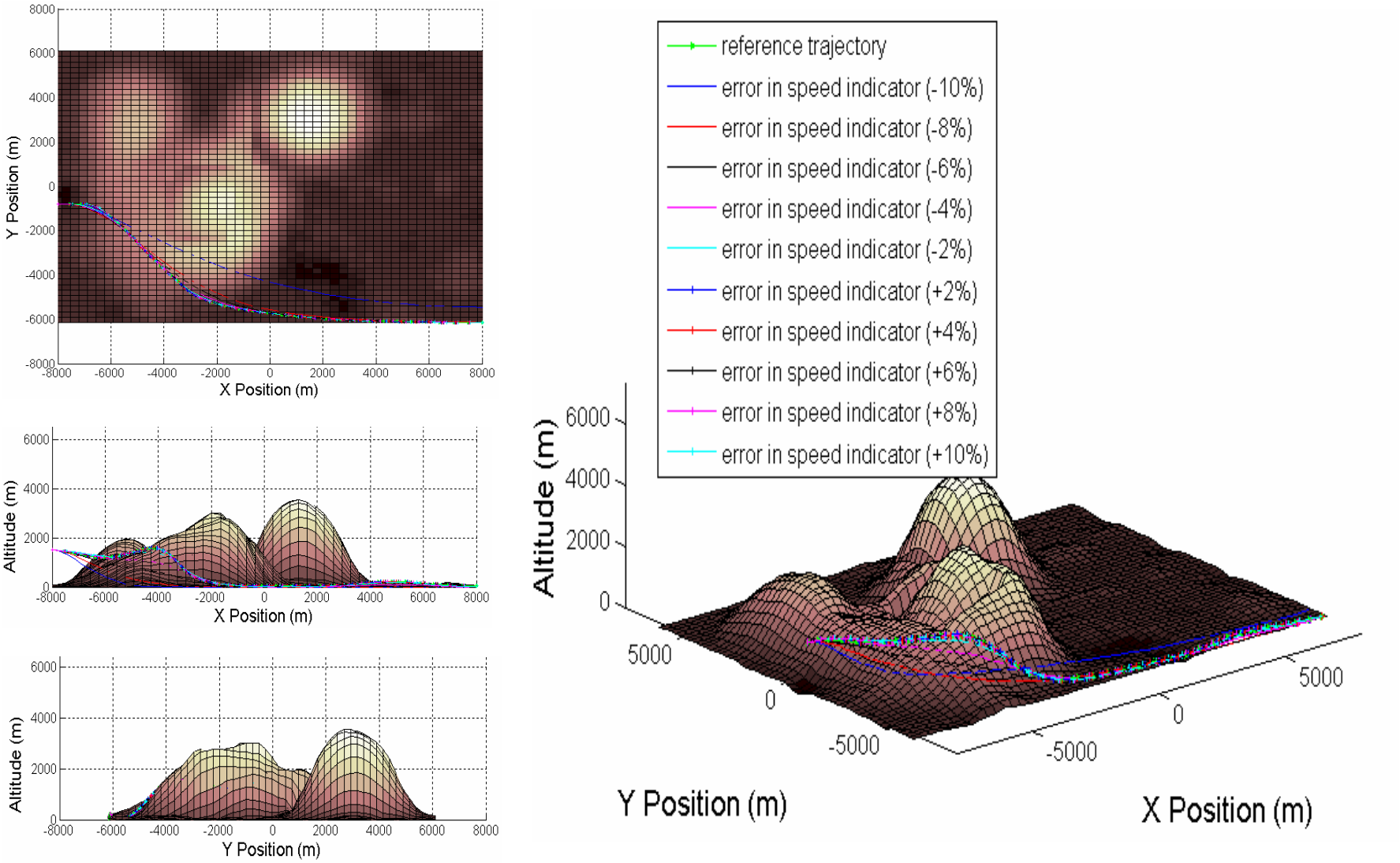


Figure 6.35: Error in speed indicator for terrain 6 minimum clearance scenario

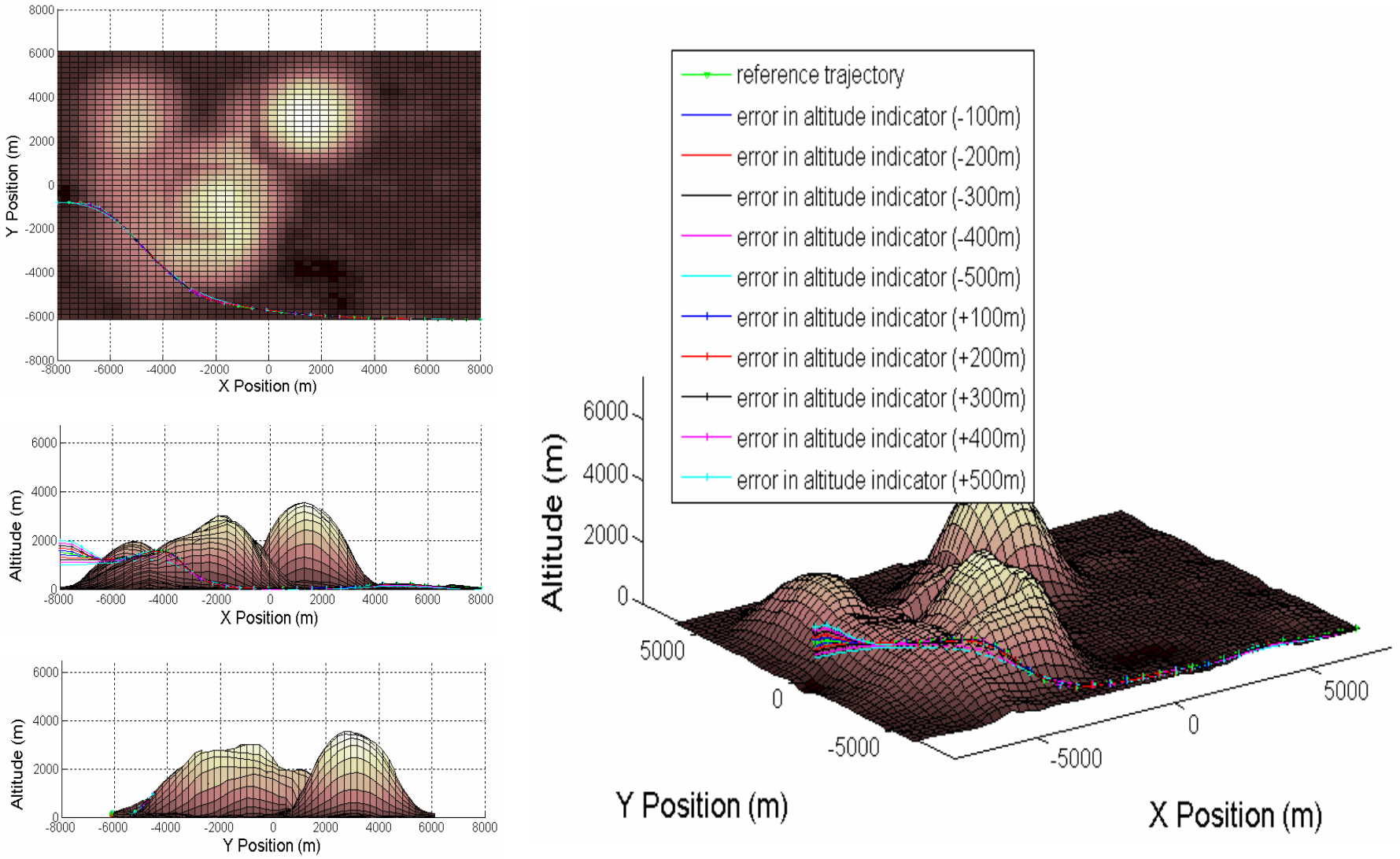


Figure 6.36: Error in altitude indicator for terrain 6 minimum clearance scenario

7 Conclusions and Recommendations

One of the major objectives of this thesis was to prove that escape trajectories can be generated in a three dimensional environment with the knowledge of terrain topology database and aircraft position only. The results presented in discussion chapter 5.1 indicate that this objective was achieved and that various terrain models produced different optimal solutions. In the case of terrain data 2, it was difficult to determine a safe trajectory as the terrain was densely mountainous providing little room for manoeuvre and the aircraft was struggling as the limits on the controls were exceeded.

Furthermore, an investigation was conducted to determine whether the minimum time and minimum clearance scenarios would produce different evasive paths. This is evident from looking at the results in the discussion chapter 5.1. It can be seen clearly that the different cost functions produced different trajectory paths. However, terrain data 4 produced similar trajectories for both minimum time and minimum clearance scenarios. Differences were observed for the minimum clearance scenario as indicated by the minimum clearance value obtained.

It was critical to investigate the effect of errors in the lateral position, speed and altitude indicators of the aircraft. As discussed, the variations in speed resulted in the different paths produced whereby in some cases, eventuated in a collision. For most cases, terrain modes, the errors in lateral position and altitude indicator did not seem to cause much variation. The exceptional case for terrain data 2 for which the densely mountainous terrain provided little room for the aircraft to conduct an evasive manoeuvre. Thus, it was found that the aircraft adhered to the optimised trajectory for errors in lateral positions and altitude indicators. For errors in speed, the provision for tolerances in the speed indicator would decrease the probability of a terrain collision.

It was possible to obtain all the solutions utilising a realistic terrain database. The aircraft was able to determine the clearance space via interpolating the x and y coordinates of the terrain. This method was effective for terrains that do not posses a very steep slope. Hence, the only exception is the case whereby a cliff is encountered. In this case, the optimiser will not be able to deduce a feasible solution.

Thus, the results obtained for this thesis demonstrate that collision avoidance is possible by performing a lateral manoeuvre rather than just conducting a regular pull up manoeuvre. The results produced in the scenario of figure 5.1 indicated that a go around manoeuvre prevented an aircraft from crashing into the terrain when the maximum pull up manoeuvre failed. However the results are preliminary indicating that further work needs to be conducted. The limitations set on the rate of change in angle of attack, bank angle and thrust settings have not been adhered to in accordance with the civil regulations. Therefore, the results obtained may not apply for a real life scenario. Additionally, it is essential that the equations of motion for a rigid body which includes the rolling moments and side slips are implemented. The addition of flight controls such as elevators, ailerons and stick force would ensure that the simulations are representative to a real life case. It is essential to consider the real time scenario as it would ensure that the aircraft is able to detect the terrain beforehand rather than just possessing a known database. For this thesis, the minimum clearance, minimum time, errors in lateral position, speed and altitude indicators results were obtained via utilising interpolative methods in ***Snopt*** and ***Direct***. Interpolation itself is a time consuming process. Although the results of this thesis were relatively accurate, the simulations were time consuming. On the average, the optimiser took 25 minutes for each simulation process. The simulation time can be minimised further via reducing the number of nodes utilised to represent the terrain database. An alternative method would be to divide the whole process into two parts. The interpolation could be done offline and this would cut the optimisation time by 70%. Thus the implementations would improve the simulations as they would accurately represent a real time and realistic military scenario.

8 References

1. Gill, P.E., Murray, W., and Saunders, M.A., "User's Guide for SNOPT 5.3: A Fortran Package for Large-Scale Nonlinear Programming", Technical report 97-5, Department of Mathematics, University of California, San Diego, La Jolla, CA, 1997" December 2002

2. Williams, P., "User's guide to Direct Version 1.16", 2005

3. Matlab 7.0 user guide

4. http://nemesis.thewavelength.net/index.php?c=2 (viewed on 03042004)

5. http://www.atlasaviation.com/AviationLibrary/CFIT/prevention_of_controlled_flight_into_terrain.htm (viewed on 14032006)

6. http://www.flightsafety.org/cfit1.html(viewed on 13022006)

7. http://www.flightsafety.org/pdf/Alert.pdf (viewed on 28012006)

8. http://www.boeing-727.com/Data/systems/infogpws.html (viewed on 10032006)

9. http://www.egpws.com (viewed on 13032006)

10. Jackson. Joseph. W. "Curved Path Approaches and Dynamic Interpolation" Aerospace and Electronic Systems Magazine, IEEE, Feb 1991, Pages 8 – 13, Volume: 6, Issue: 2

11. Hewitt .C, Hickey. A.J and Boyes. J.D "A Ground and Obstacle Collision Avoidance Technique (GOCAT)" IEEE AES Magazine, August 1991

12. Barfield Finley, Probert Judith and Browning Duke "All Terrain Ground Collision Avoidance and Manoeuvring Terrain Following for Automated Low Level Night Attack" IEEE AES Systems Magazine, March 1993

13. Heimdahl Mats P.E, Leveson Nancy G. and Reese Jon Damon "Experience from Flying the TCAS II Requirements Using RSML" IEEE 31 Oct-7 Nov 1998 ISBN: 0-7803-5086-3

14. Inder Lee F.W and Kuchar James K. "Evaluation of Collision Avoidance Manoeuvres for Parallel Approach" Journal Of Guidance, Control and Dynamics Vol. 22, No. 6, November – December 1999

15. Massink Mieke and Francesco Nicoletta De "Modelling Free Flight with Collision Avoidance" Seventh International Conference on Engineering of Complex Computer Systems, June 11 - 13, 2001

16. Swihart Donald and Barfield Finley "An Advanced Automatic Ground Collision Avoidance System for Fighter Aircraft" SAFE Association, Annual Symposium, 37th, Atlanta, GA; 6-8 Dec. 1999. 1999

17. Shandy Surya U. and Valasek John “Intelligent Agent for Aircraft Collision Avoidance” AIAA Guidance, Navigation, and Control Conference and Exhibit, Montreal, Canada, Aug. 6-9, 2001

18. James K. Kuchar “Markov Model of Terrain for Evaluating of Ground Proximity Warning System Thresholds” Journal Of Guidance, Control and Dynamics Vol. 24, No. 3, May – June 2001

19. Fayad, C. and Webb, P. “Optimized Fuzzy Logic Based on Algorithm for a Mobile Robot Collision Avoidance in an Unknown Environment”, 7th European Congress on Intelligent Techniques & Soft Computing, Aachen, Germany, September 13-16, 1999.

20. Richards, A. and How, J. P. “Aircraft trajectory planning with collision avoidance using mixed integer linear programming” Proceedings of the American Control Conference, 8-10 May 2002

21. R. Asep, A.K, Achaibou and F.Mora-Camino,”Automatic collision avoidance based on supervised predictive controllers” Control Engineering, Vol. 4, 8 pp. 1169 – 1175, 1996

22. Kumar, B. Ajith, Ghose, D. “Radar-Assisted Collision Avoidance Guidance Strategy for Planer Flight” IEEE Transactions on Aerospace and Electronic Systems, Vol. 37, No. 1, January 2001

23. Steele, K. L. and Egbert, P. K. “A Unified Framework for Collision Detection, Avoidance and Response” Proceedings of WSCG '98, The Sixth International Conference in Central Europe on Computer Graphics and Visualization '98, Feb. 9-13, 1998, pp 517-524.

24. Egbert, P. and Winkler, S. “Collision-Free Object Movement Using Vector Fields” IEEE Computer Graphics and Applications, 16, 4 July 1996, pp 18 – 24.

25. Bryson, J. R. and Arthur, F, “Energy-State Approximation in Performance Optimization of Supersonic Aircraft”, Journal of Aircraft, Vol. 6, No. 6, pp. 481 – 488, November – December 1969

26. Lu Ping, Pierson, B. L. “Optimal Aircraft Terrain-Following Analysis and Trajectory Generation” AIAA Atmospheric Flight Mechanics Conference, Baltimore, MD, Technical Papers; 7-10 Aug. 1995. pp. 555-560. 1995

27. Flood Cecilia, “Real time Trajectory Optimization for Terrain Following Based on Non-Linear Model Predictive Control”, Linkoping University, Masters Thesis, pp. 1 - 55, 2001

28. Ulf, Ringertz. “Flight Testing an Optimal Trajectory for the Saab J35 Draken” Journal of Aircraft, Vol. 37, No.1, pp. 187 – 189, 14 September 1999

29. Ulf, Ringertz. “Optimal Trajectory for a Minimum Fuel Turn” Journal of Aircraft, Vol.37, No. 5, pp. 932 – 934, 4 May 2000

30. Kim, S. K, Tilbury D. M, “Trajectory Generation for a Class of nonlinear Systems with Input and State Constraints”, American Control Conference, 2001. Proceedings of the 2001, Arlington, VA, USA, Volume: 6, pp. 4908 - 4913

31. Chen, D. Z., Szczebra, R. J. and Uhran J. J. J. “Determining Conditional Shortest Paths in an Unknown 3-D Environment Using Framed-Octrees” Proc. IEEE International Conf. on Systems, Man, and Cybernetics, pp. 4101- 4106.

32. Singh, L, “Trajectory Generation for a UAV in Urban Terrain, using Nonlinear MPC” Proceedings of the American Control Conference Arlington, VA, June 25 – 27, 2001

33. Trent, A., Venkataraman, R. and Doman, D. “Trajectory Generation Using A Modified Simple Shooting Method” (2004), Proc. IEEE Aerospace Conference, Big Sky, MT.

34. Holsapple, R. Venkataraman, R. and Doman, D. “A Modified Simple Shooting Method for Solving Two Point Boundary- Value Problems” (2003), Proc. IEEE Aerospace Conference,

35. Williams, P. “Hermite-Legendre-gauss-lobatto direct transcription methods in trajectory optimization” The American Astronautical Society, Vol. 5, No. 131 2005

36. Hargraves C. R and Paris S.W, ”Direct Trajectory Optimization Using Nonlinear Programming and Collocation” Journal of Guidance and Control, Vol. 10, No. 4, pp. 338 – 342, American Institute of Aeronautics and Astronautics, Inc 1987

37. Murray Jerry and Zampathas Jim “Digital Terrain for Research and Development Simulations” SAE, Aerotech '92 Conference, Anaheim, CA, Oct. 5-8, 1992

38. Petit, N. Milam, M. B. and Murray, R. M. *"Inversion Based Trajectory Optimization"*, in 2001 IFAC Symposium on Nonlinear Control Systems Design

39. Baldwin, J. C. R. and Smith, A. “GPS Based Terrain Avoidance Systems – A Solution for General Aviation Controlled Flight into Terrain” Rannoch Corporation 413-417

40. Bryson, A. E. “Dynamic Optimization” Addison-Wesley. Menlow Park, 1999

41. Betts John. T, “Practical Methods for Optimal Control Using Nonlinear Programming” Society for Industrial & Applied Mathematics, U.S.; 1st edition 1 May 2001

42. De Boor, C., “A Practical Guide to Splines” Springer – Verlag, New York, 1978

43. Nelson, R. C., “Flight stability and automatic control”, New York: McGraw-Hill, 1989

44. http://selair.selkirk.bc.ca/aerodynamics1/Lift/LF_Climb.html (viewed on 25052005)

45. http://selair.selkirk.bc.ca/aerodynamics1/Lift/Page10.html (viewed on 25052005)

46. Sharma T., Bil C. and Eberhard A. “Optimal Flight Trajectories for Terrain Collision Avoidance” AIAC, Eleventh Australian International Aerospace Congress Melbourne Australia, March 2005

47. Sharma T., Bil C. and Eberhard A. “Control System for Optimal Flight Trajectories for Terrain Collision Avoidance” KES, Ninth International Conference on Knowledge-Base Intelligence Information & Engineering Systems, 14 – 16 September, 2005 Hilton, Melbourne, Australia

48. Sharma T., Williams P., Bil. C. and Eberhard A. “Optimal Three-Dimensional Aircraft Terrain Following and Collision Avoidance” The 7th Biennial Engineering Mathematics and Applications Conference, Melbourne Victoria, 25 – 28 September 2005

49. Sharma T., Bil C. and Eberhard A. “Optimized Three Dimensional Collision Avoidance Trajectories into Terrain” The 1st Malaysian Software Engineering Conference, Copthorne Orchid Hotel Penang, 12th to 13th December 2005

50. Sharma T., Bil C. and Eberhart A. “Optimum Flight Trajectories and Sensitivity Analysis for Terrain Collision Avoidance Systems” 25th International Congress of the Aeronautical Sciences, Hamburg Germany, September 2006

A Appendix

A.1.1 Description of SQP method

This section describes some terminology used in the description of the subroutine and its arguments. The full terms are described in Gill, Murray and Saunders [1].

A.1.2 Constraints and slack variables

The upper and lower bounds on the m components of $f(x)$ and $A_L x$ are said to define the general constraints of the problem. ***Snopt*** converts the general constraints to equalities by the introduction of a set of slack variables s, where $s = (s_1, s_2, \ldots\ldots, s_m)^T$.
The problem (SparseNP) can be rewritten as

$$\underset{X,S}{\text{minimize}} \quad f_0(x)$$

$$\textit{subject to} \quad \begin{pmatrix} f(x) \\ A_L x \end{pmatrix} - s = 0, \qquad (1)$$

$$l \le \begin{pmatrix} x \\ s \end{pmatrix} \le u$$

Non linear constraint

Linear constraint

The linear and nonlinear general constraints become equalities of the form $f(x) - s_N = 0$ and $A_L x - s_L = 0$, where s_L and s_N are known as the linear and nonlinear slacks respectively.

A.1.3 Major iteration

The basic structure of ***Snopt*** involves major and minor iterations. The major iterations generate sequence of iterates (x_k) that satisfies the linear constraints and converge to a point that satisfies the first-order conditions for optimality. At each iterate a QP sub problem is used to generate a search direction towards the next iterate (x_{k+1}). The constraints of the sub problem are formed from the linear constraints $A_L x - s_L = 0$ and the non-linear constraint linearization.

$$f'(x_k) + f'(x_k)(x - x_k) - s_N = 0 \tag{2}$$

where, $f'(x_k)$ denotes the Jacobian matrix, whose elements are the first derivatives of f(x) evaluated at x_k. The QP constraints comprise of the m linear constraints.

$$\begin{aligned} f'(x_k) - s_N \quad &= -f(x_k) + f'(x_k)x_k \\ A_L x \quad &- s_l = 0 \end{aligned} \tag{3}$$

where x and s are bounded above and below by u and l as before. If the $m \times n$ matrix A and m-vector b are defined as

$$A = \begin{pmatrix} f'(x_k) \\ A_L \end{pmatrix} \text{ and } b = \begin{pmatrix} f'(x_k) + f'(x_k)x_k \\ 0 \end{pmatrix}, \tag{4}$$

then the QP can be written as

$\underset{x,s}{\text{minimize}}\, q(x)$ *subject to* $Ax - s = b$

, $l \le \begin{pmatrix} x \\ s \end{pmatrix} \le u$,

where q(x) is a quadratic approximation to a modified Lagrangian function.

A.1.4 Minor iterations

Solving the QP sub problem is itself an iterative procedure, with the minor iterations if an SQP method being the iterations of the QP method. At each minor iteration, the constraints $Ax - s = b$ are (conceptually) partitioned into the form:

$$Bx_B - Sx_S + Nx_N = b, \tag{5}$$

where the basic matrix B is square and non-singular. The elements of x_B, x_S and x_N are called the basic, super basic and nonbasic variables respectively; they are permutation of the elements of x and s. At a QP solution, the basic and super basic variables will lie somewhere between their bounds whilst the nonbasic variables will be equal to the upper and lower bounds. At each iteration, x_S is regarded as a set of independent variables that is free to move in any desired direction, specifically that will improve that will improve the value of the QP objective (or the sum of infeasibilities). The basic variables are then adjusted in order to ensure that (x, s) continues to satisfy Ax-s = b. The number of super basic variables (say n_s) indicates the number of degrees of freedom remaining after the constraints have been satisfied. In broad terms, n_s is a measure of how nonlinear the problem is. In particular, n_s will always be zero for Linear Programming (LP) problems. If it appears that no improvement can be made with the current definition of B, S and N, a nonbasic variable is selected to be added to S, and the process is repeated with the of value of n_s with increment of one. At all stages, if a basic or super basic variables encounters one of its bounds, the variables is made nonbasic and the value of n_s is decreased by one. Associated with each of the m equality constraints Ax - s = b a dual variables π is associated. Similarly, each variable in (x, s) is associated to a reduced gradient d_j. The reduced gradients for the variables x are the quantities (g – $A_T\pi$), where g is the gradient of the QP objective, and the reduced gradients for the slacks are the dual variables. The QP sub problem is optimal if $d_j \geq 0$ for all nonbasic variables at their lower bounds, $d_j \leq 0$ for all nonbasic variables at their upper bounds, and $d_j = 0$ for other variables, including super basics. In practice, an approximate QP solution is found by relaxing these conditions on d_j (see the Minor optimality tolerance described in §7.6).

A.1.5 Merit Function

After a QP sub problem has been solved, new estimates of the Non Linear Programming (NP) solution are computed using a line search on the augmented Lagrangian merit function

$$M(x,s,\pi) = f_0(x) - \pi^T(f(x) - s_N) + \frac{1}{2}(f(x) - s_N)^T D(f(x) - s_N) \qquad (6)$$

where D is a diagonal matrix of penalty parameters. If (x_k, s_k, π_k) denotes the current solution estimate and $(\hat{x}_k, \hat{s}_k, \hat{\pi}_k)$ denotes the optimal QP solution, the line search determines a step $a_k (0 < a_k \leq 1)$ such that the new point gives a sufficient decrease in the merit function.

$$\begin{pmatrix} x_k + 1 \\ s_k + 1 \\ \pi_k + 1 \end{pmatrix} = \begin{pmatrix} x_k \\ s_k \\ \pi_k \end{pmatrix} + \alpha_k \begin{pmatrix} \hat{x}_k - x_k \\ \hat{s}_k - s_k \\ \hat{\pi}_k - \pi_k \end{pmatrix} \tag{7}$$

When necessary, the penalties in D are increased by the minimum-norm perturbation that ensures descent for M [10]. As in NPSOL, s_N is adjusted to minimise the merit function as a function of s prior to the solution of the QP sub problem.

A.1.6 Treatment of constraint infeasibility

Snopt makes explicit allowances for infeasible constraints. Infeasible linear constraints are detected first by solving a problem of the form.

FLP $\underset{x,v,w}{\text{minimize}} \quad e^T(v + w)$

$$\text{Subject to } l \leq \begin{pmatrix} x \\ A_L x - v + w \end{pmatrix} \leq u, v \geq 0, w \geq 0, \tag{8}$$

where e is a vector of ones. This is equivalent to minimizing the sum of the general linear constraint violations subject to the simple bounds. (In the literature pertaining to linear programming, this approach is often called elastic programming)

If the linear constraints are feasible, all the subsequent iterates satisfy the linear constraints. (Such a strategy allows linear constraints to be used to define a region in which the function can be safely evaluated.) ***Snopt*** proceeds to solve (Npsparse) as given, using search directions obtained from a sequence of quadratic programming sub problems. If a QP subproblem proves to be infeasible or unbounded (or if the dual variables π for the non linear constraint become large), ***Snopt*** enters “elastic” mode to solve the problem.

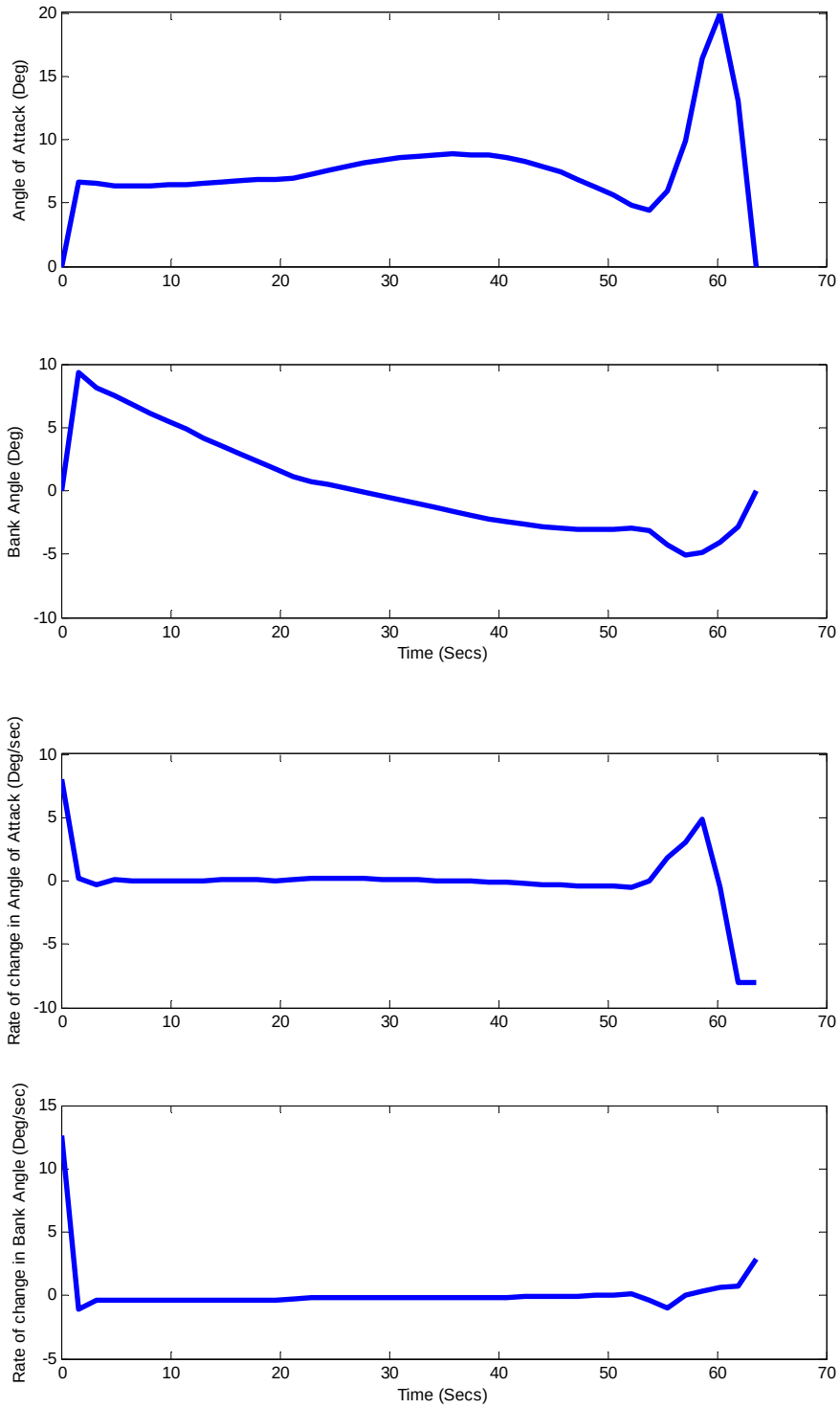
$$\text{NP}(\gamma) \qquad \underset{x,v,w}{\text{minimize}} \qquad f_0(x) + \gamma e^T (v + w)$$

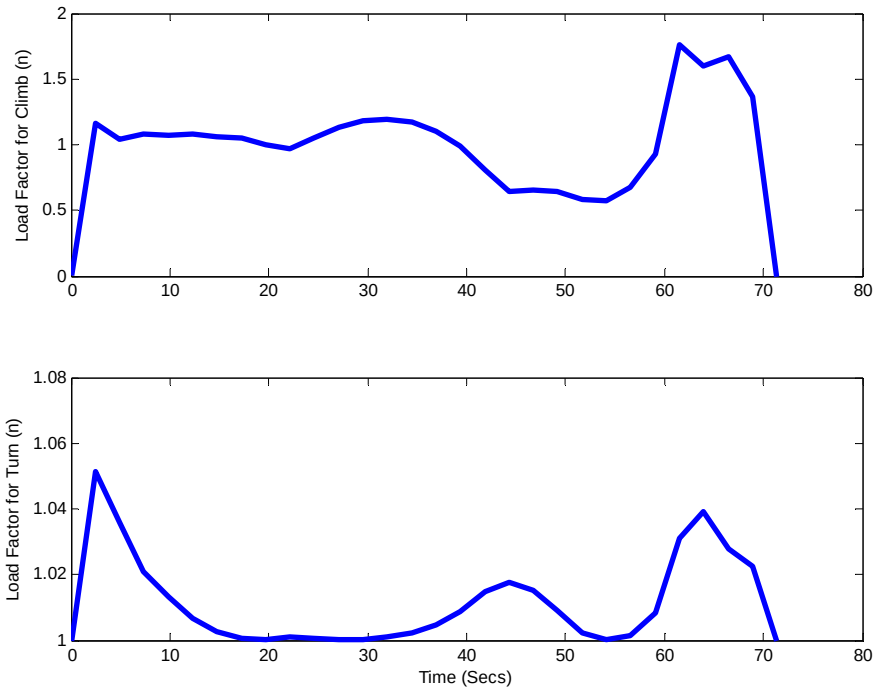
$$\text{Subject to } l \le \begin{pmatrix} x \\ f(x) - v + w \\ A_L x \end{pmatrix} \le u, v \ge 0, w \ge 0, \tag{9}$$

where γ is a nonnegative parameter (the elastic weight) and $f_0(x) + \gamma e^T (v + w)$ is known as the composite objective. If γ is sufficiently large, this is equivalent to minimising the sum of the non-linear constraint violations subject to the linear constraints and bounds.

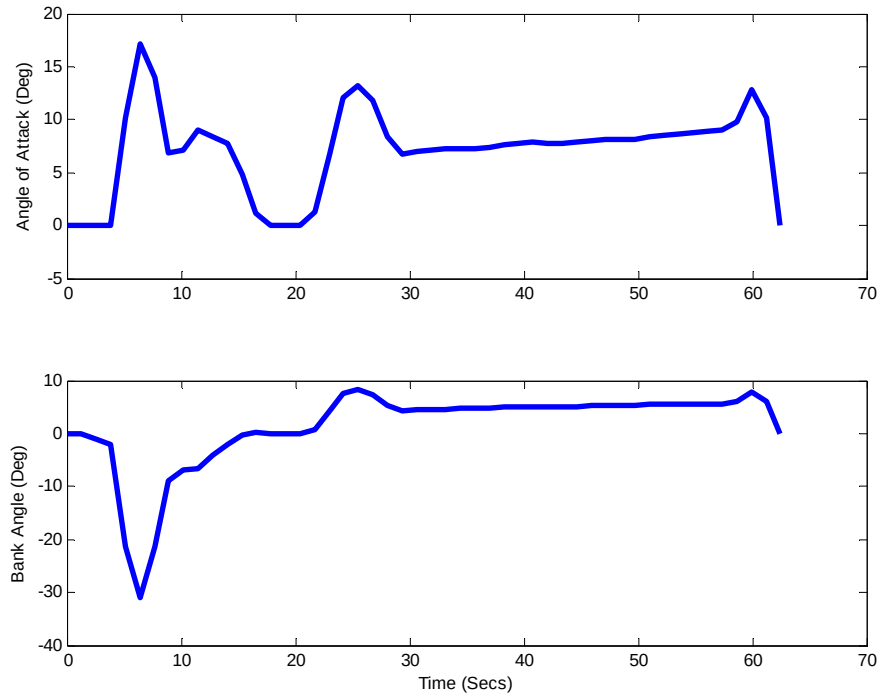
A.2.1 Minimum time and minimum clearance to terrain control plots

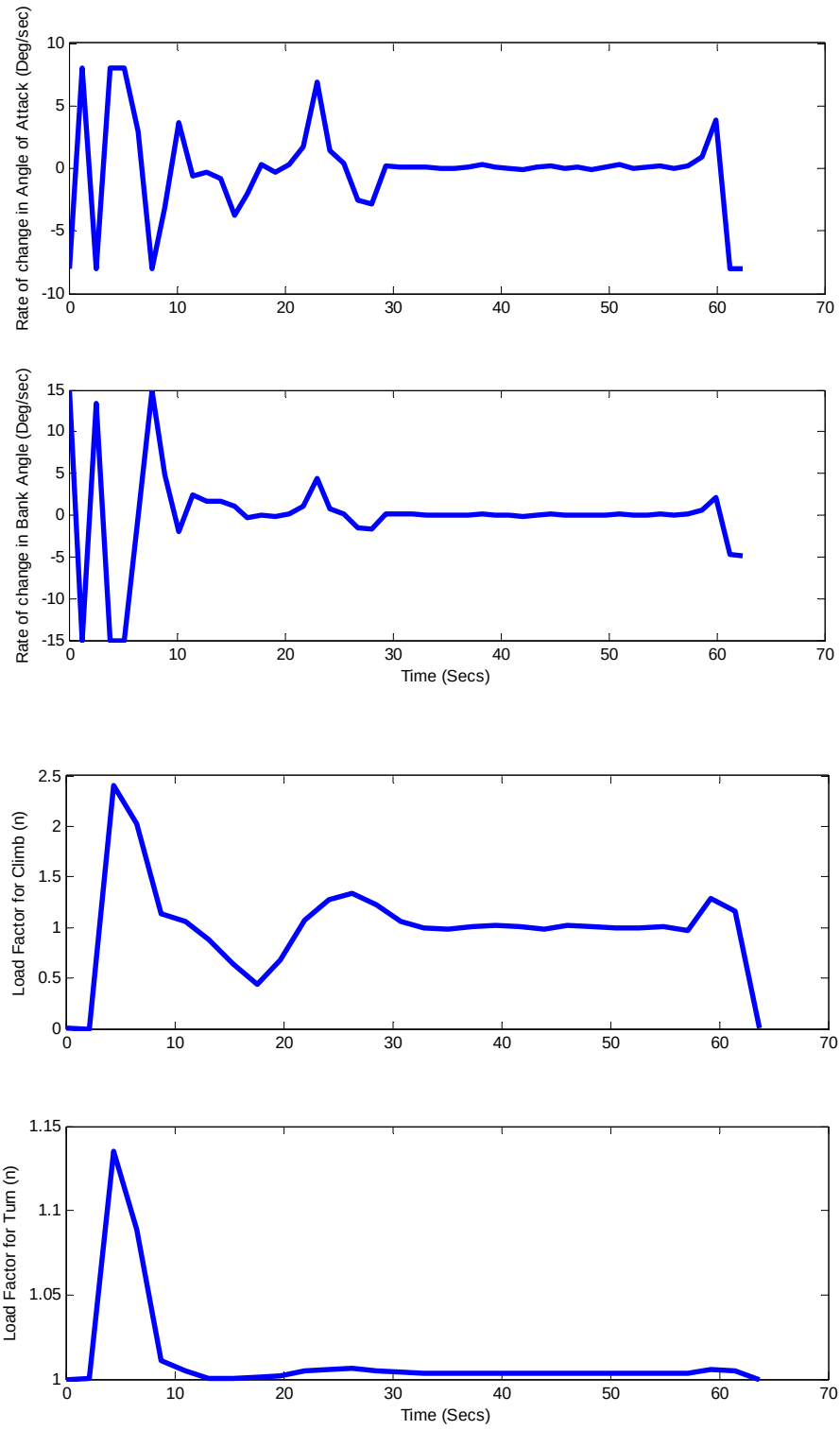
Control plots for Terrain 1 for minimum time



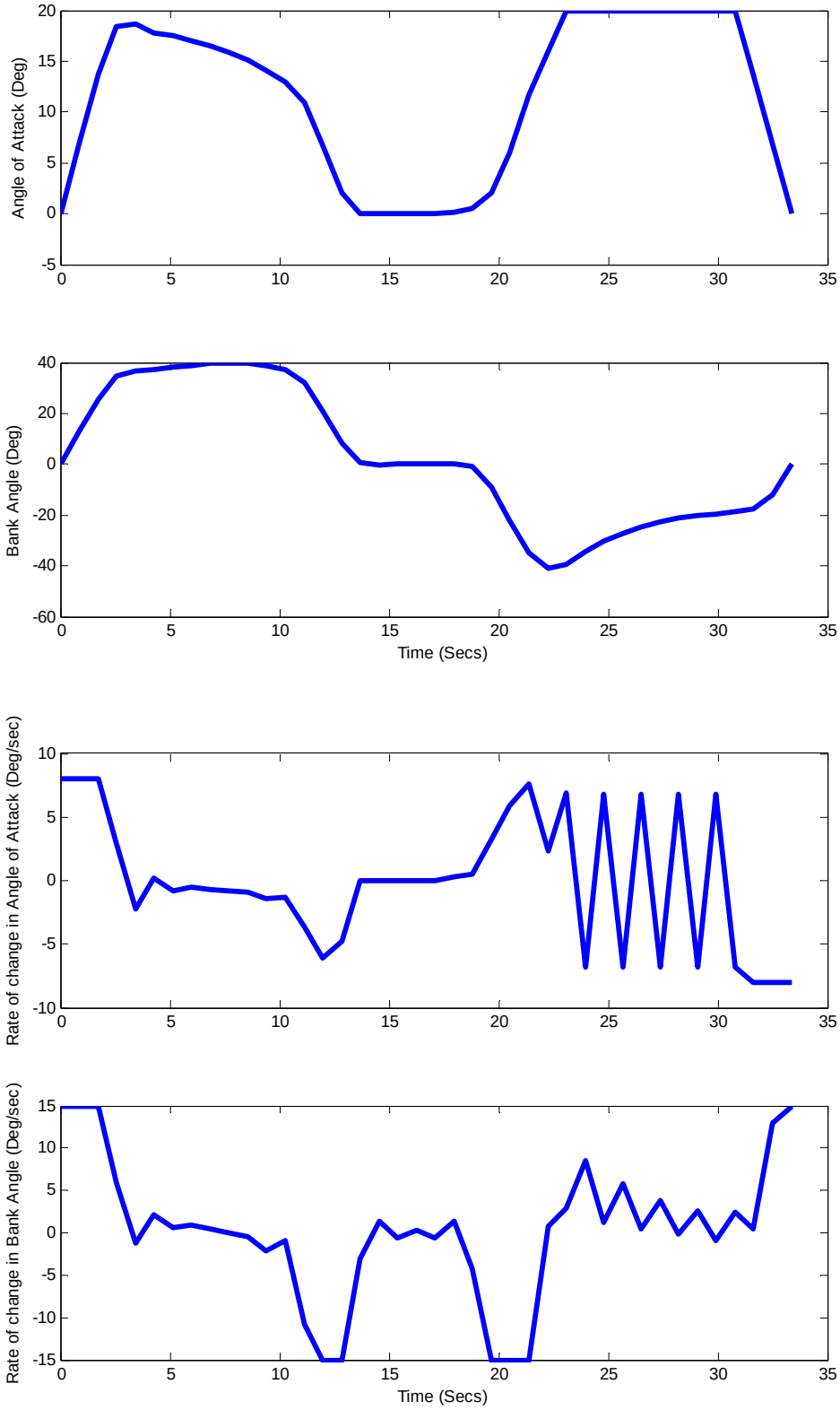


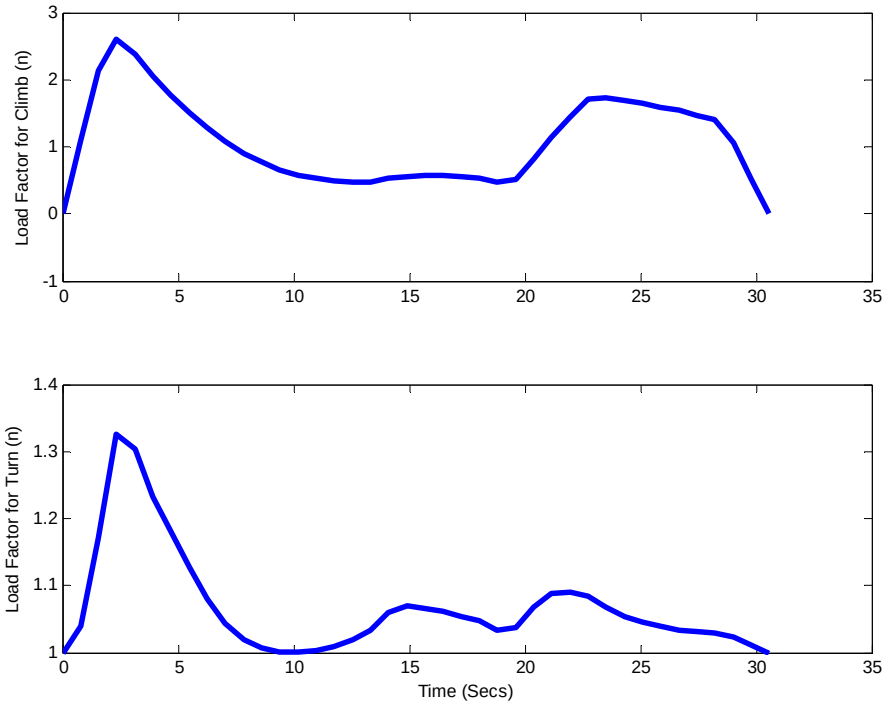
Control plots for Terrain 1 for minimum clearance to terrain



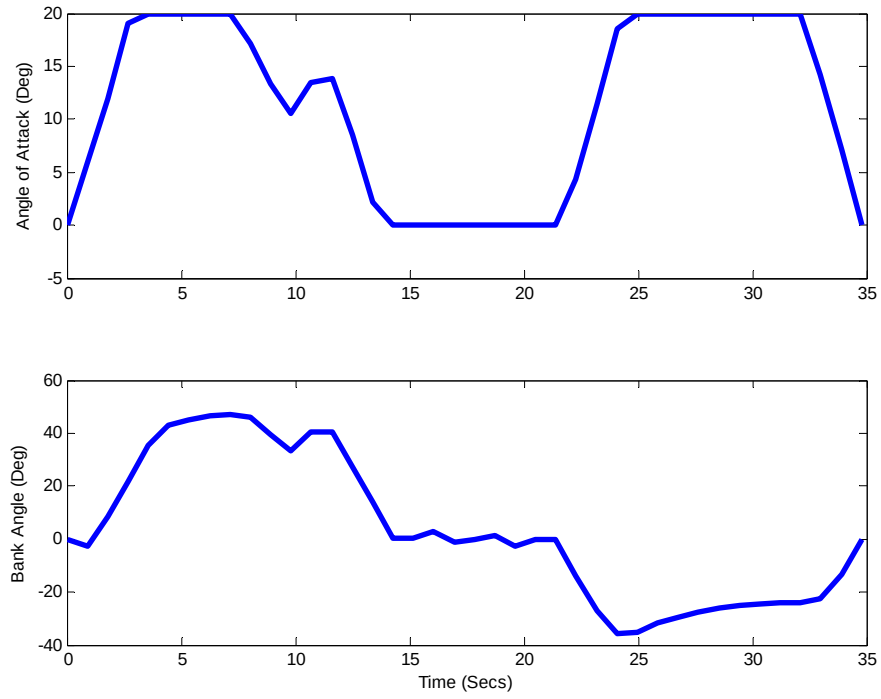

Rate of change in Angle of Attack (Deg/sec)
10
5
0
-5
-10
0
10
20
30
40
50
60
70
Rate of change in Bank Angle (Deg/sec)
15
10
5
0
-5
-10
-15
Time (Secs)
Load Factor for Climb (n)
2.5
2
1.5
1
0.5
0
Load Factor for Turn (n)
1.15
1.1
1.05
1
Time (Secs)

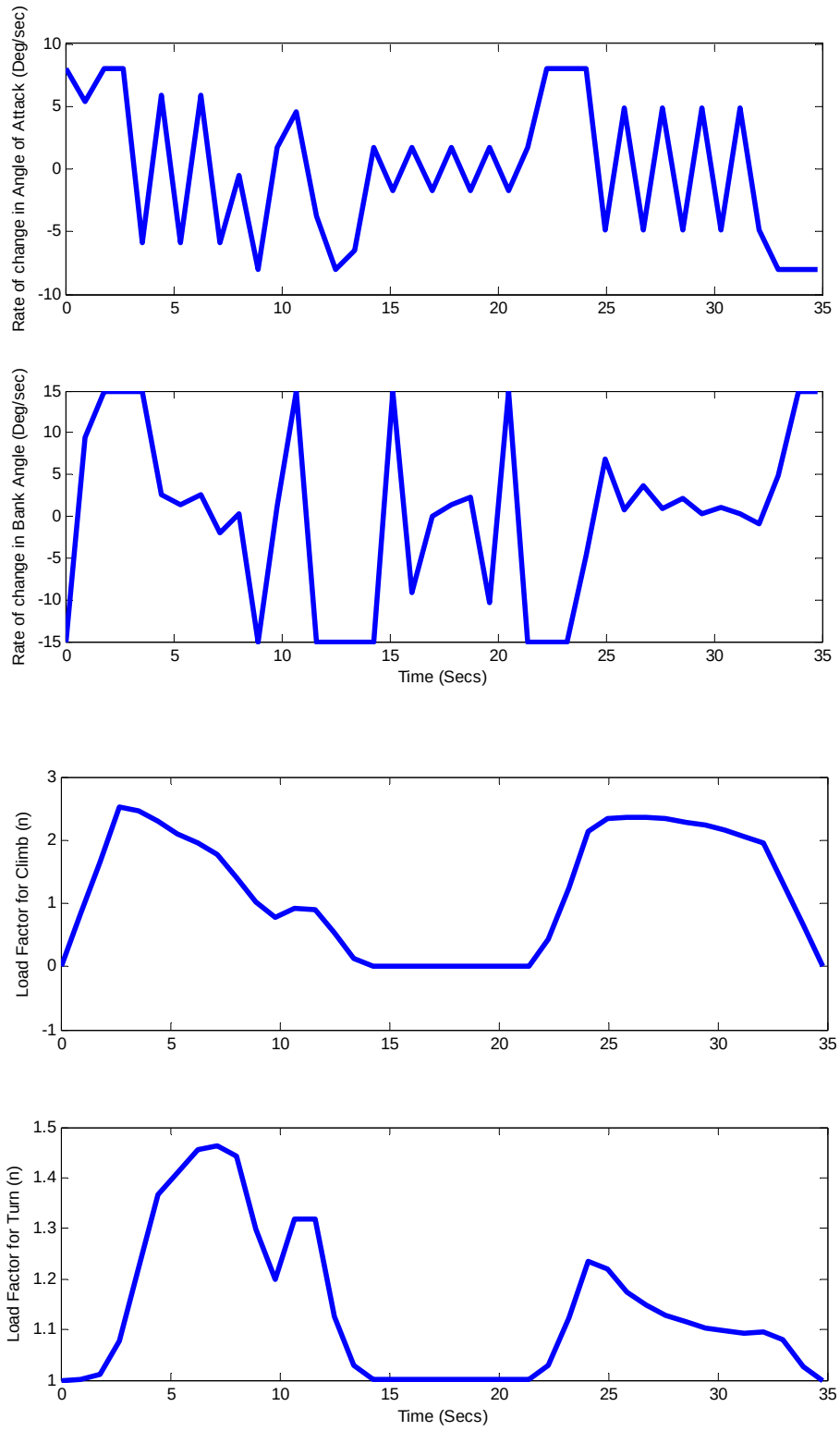
Control plots for Terrain 2 for minimum time



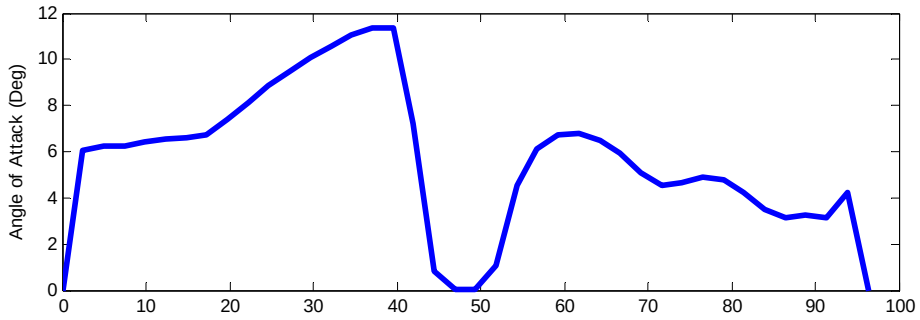


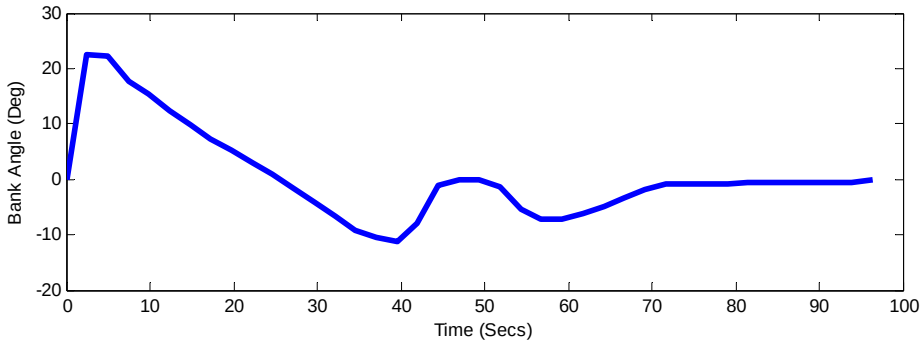
Control plots for Terrain 2 for minimum clearance to terrain

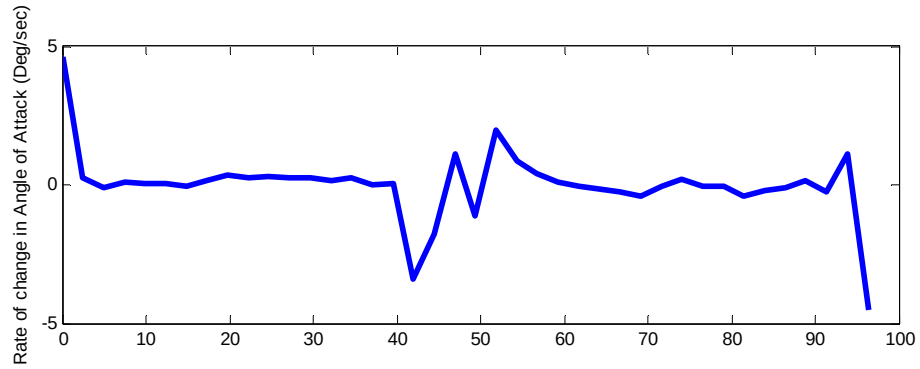


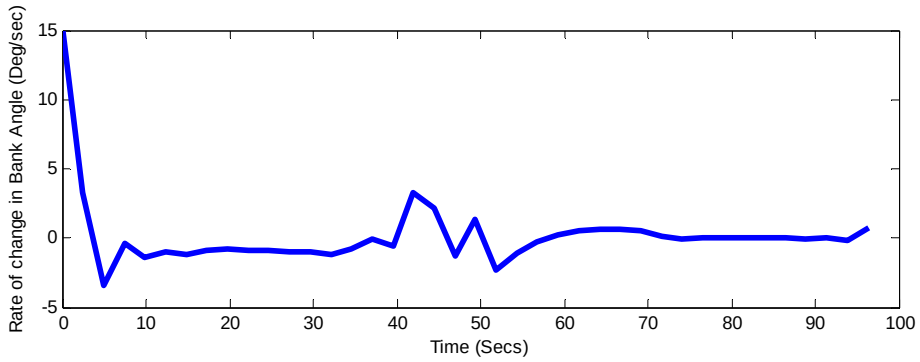

Rate of change in Angle of Attack (Deg/sec)
10
5
0
-5
-10
0
5
10
15
20
25
30
35
Rate of change in Bank Angle (Deg/sec)
15
10
5
0
-5
-10
-15
Time (Secs)
Load Factor for Climb (n)
3
2
1
0
-1
Load Factor for Turn (n)
1.5
1.4
1.3
1.2
1.1
1
Time (Secs)

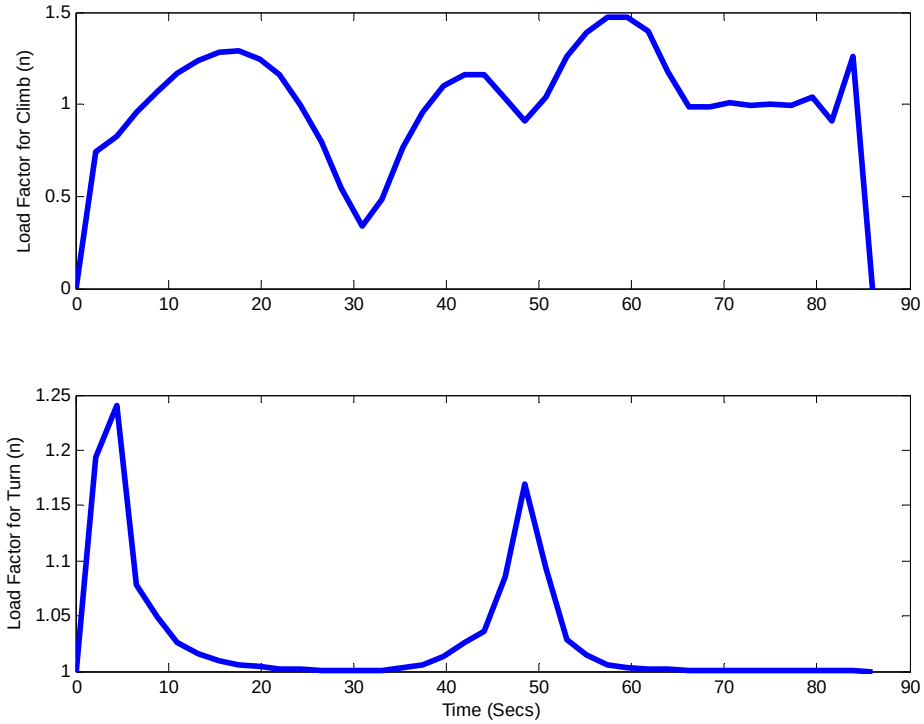
Control plots for Terrain 3 for minimum time



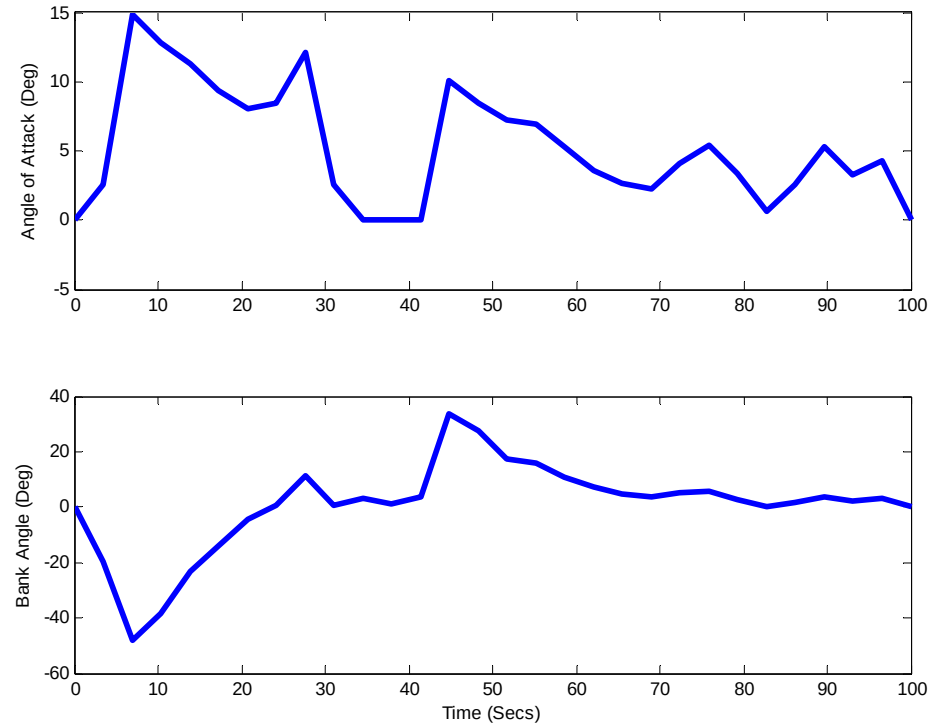


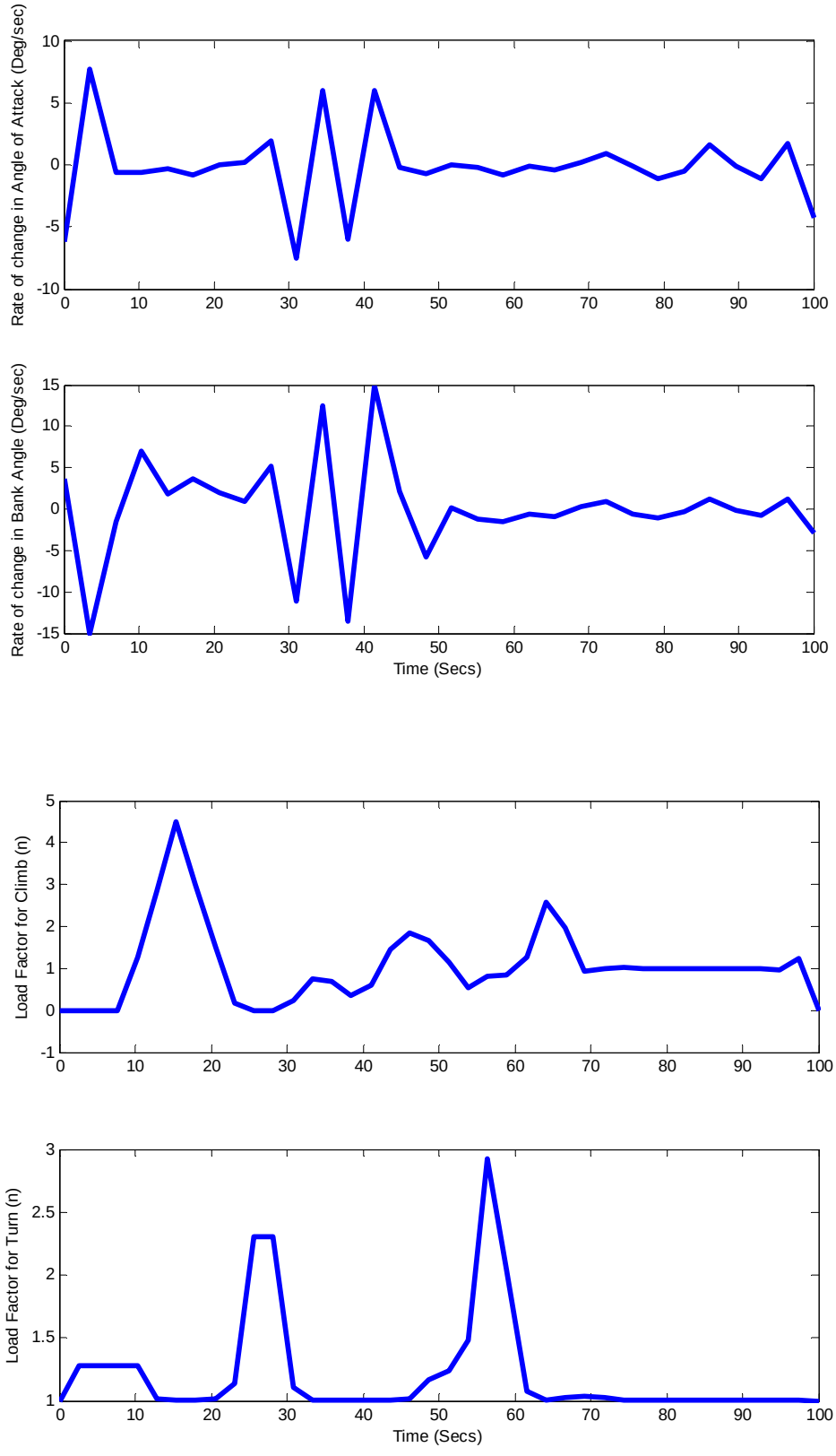




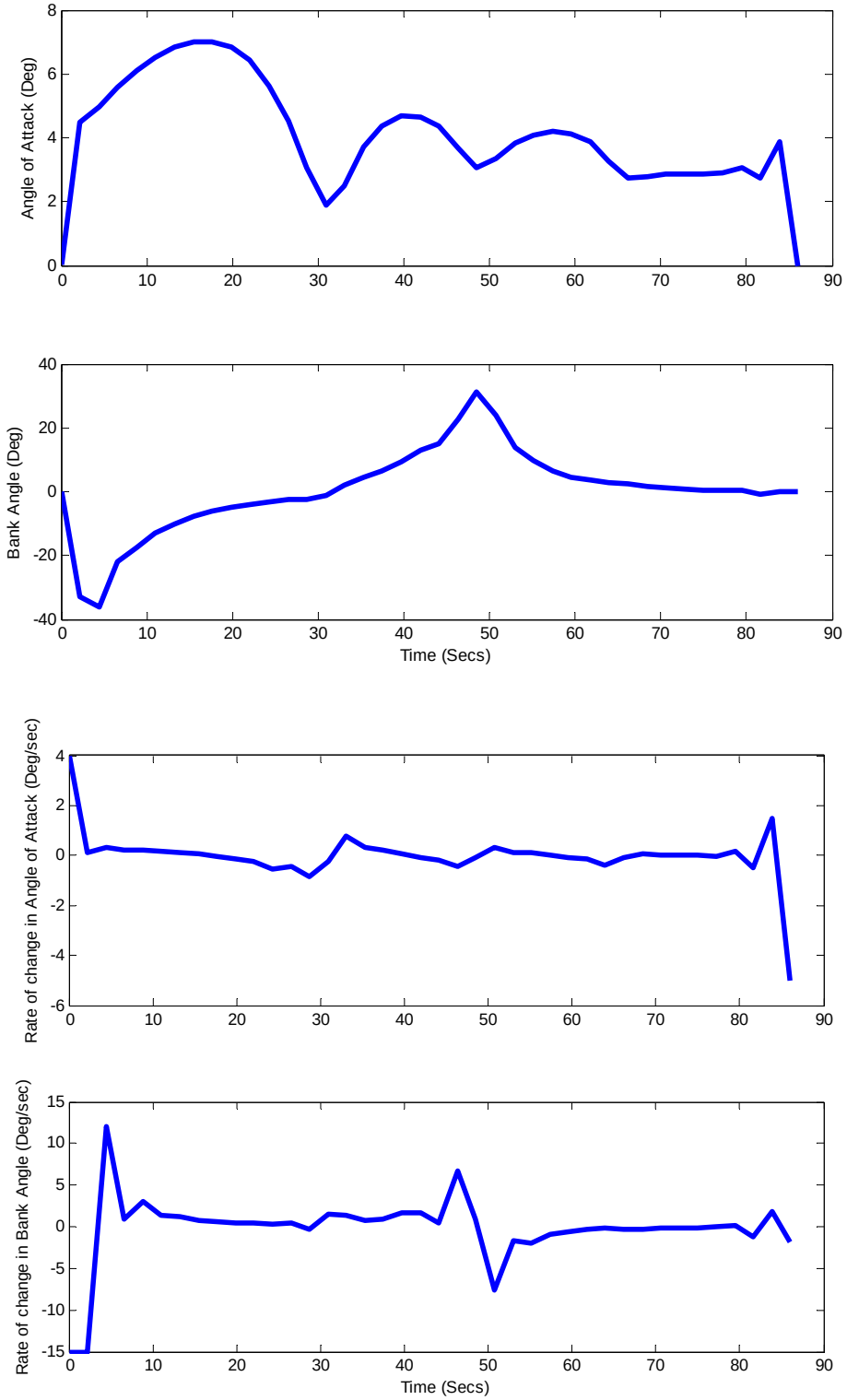


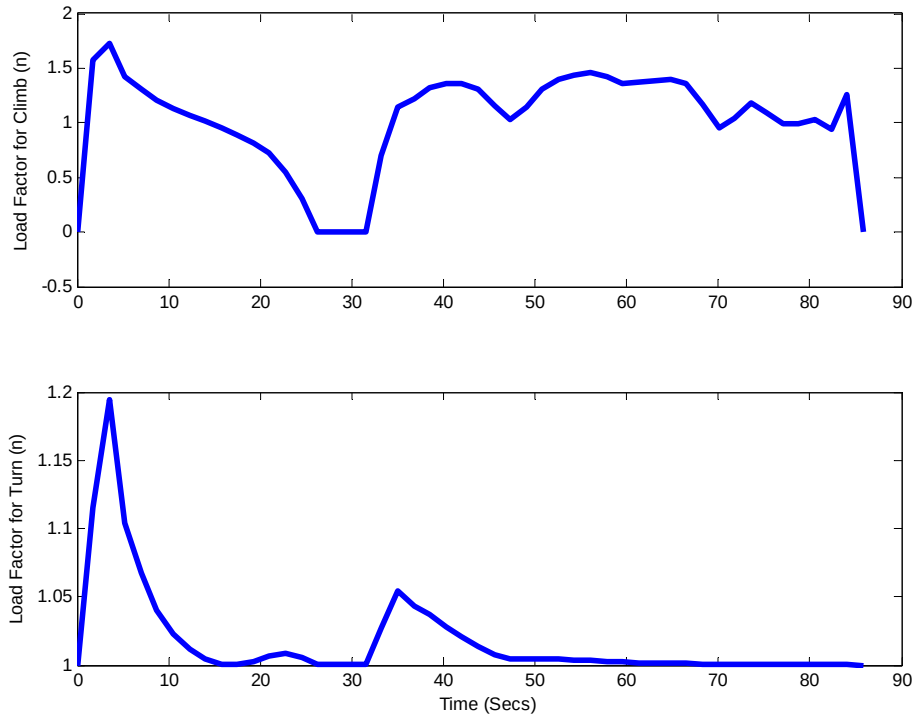
Control plots for Terrain 3 for minimum clearance to terrain



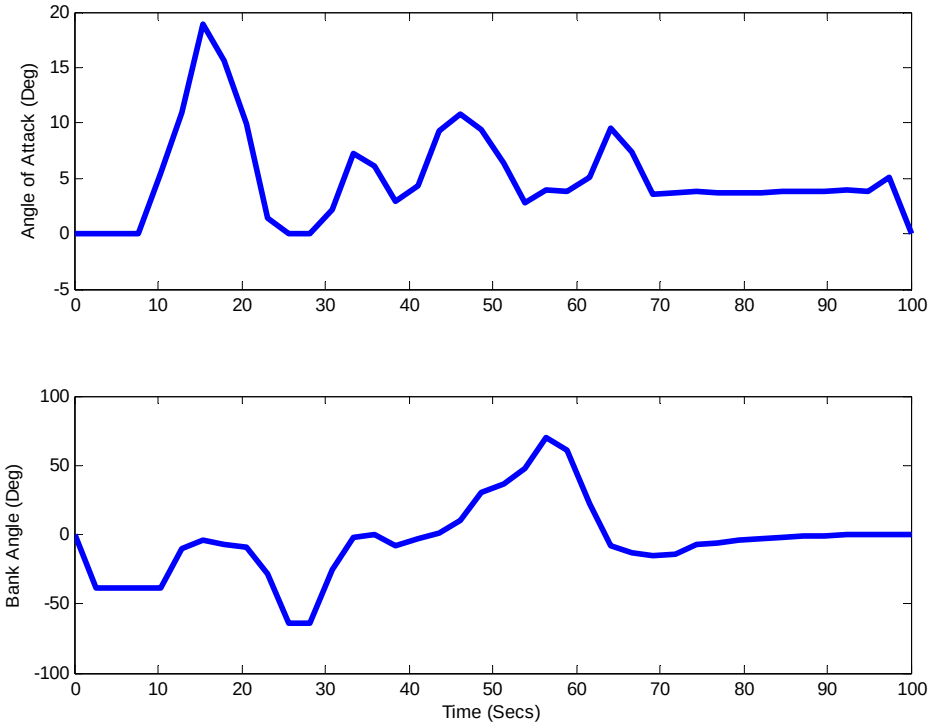

Rate of change in Angle of Attack (Deg/sec)
10
5
0
-5
-10
0
10
20
30
40
50
60
70
80
90
100
Rate of change in Bank Angle (Deg/sec)
15
10
5
0
-5
-10
-15
Time (Secs)
Load Factor for Climb (n)
5
4
3
2
1
0
-1
Load Factor for Turn (n)
3
2.5
2
1.5
1
Time (Secs)

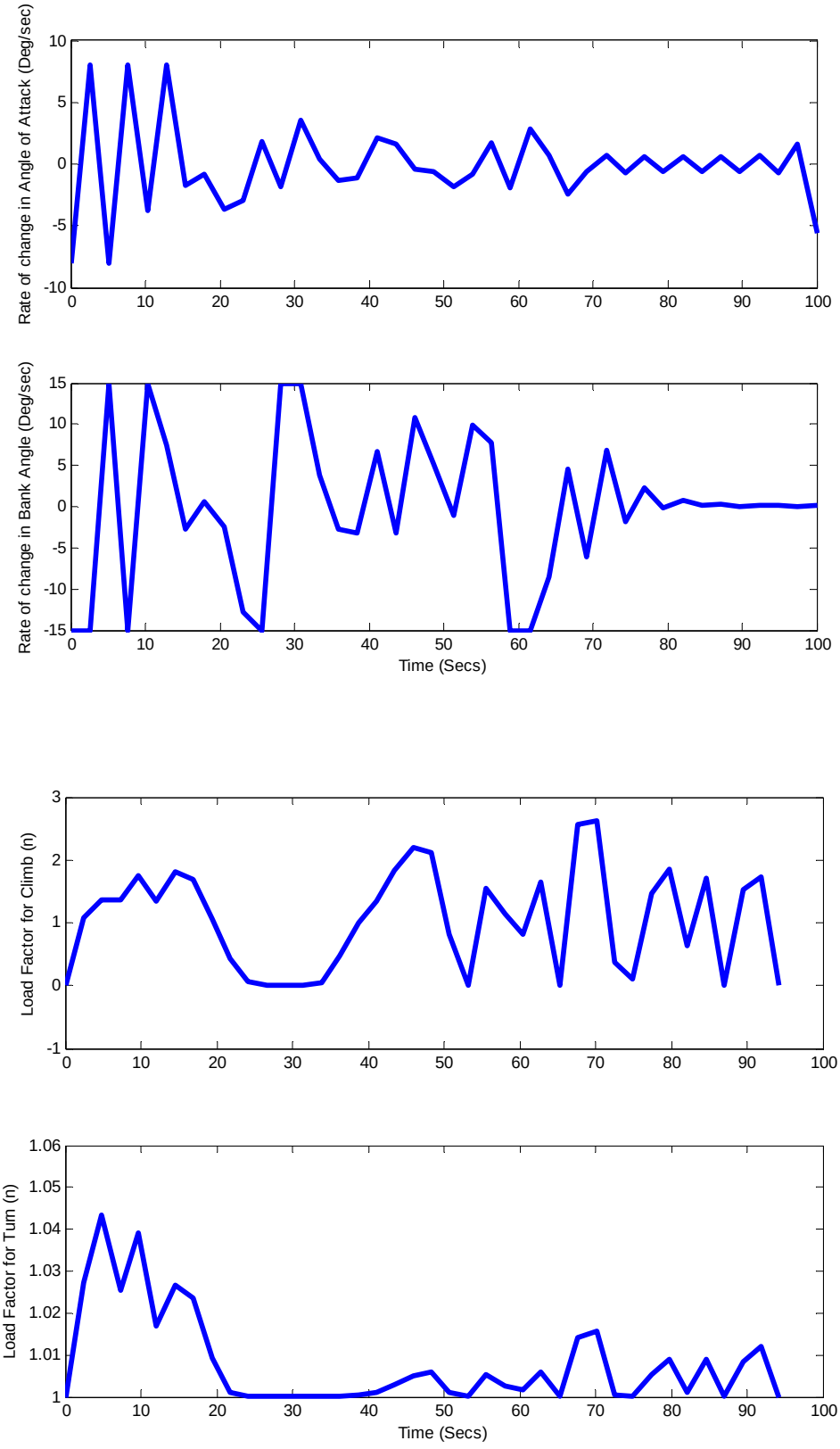
Control plots for Terrain 4 for minimum time



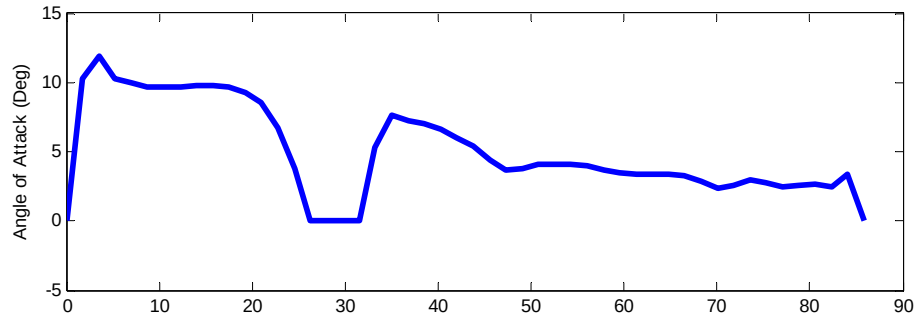


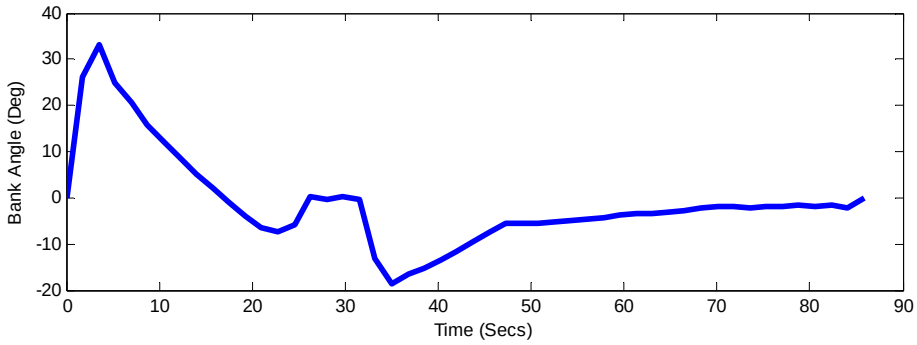
Control plots for Terrain 4 for minimum clearance to terrain

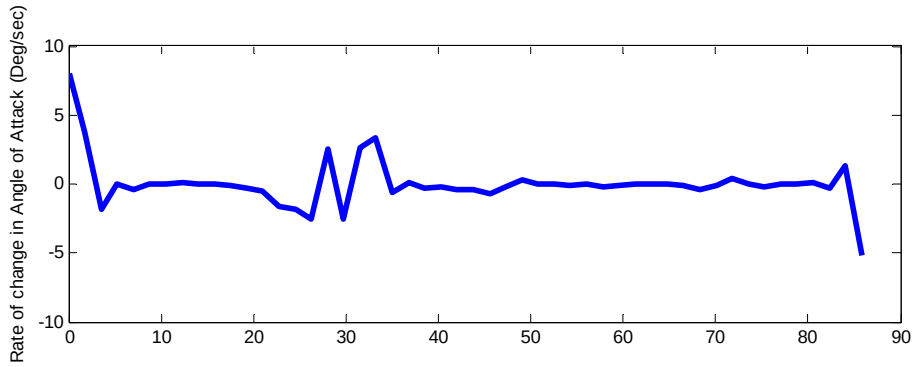


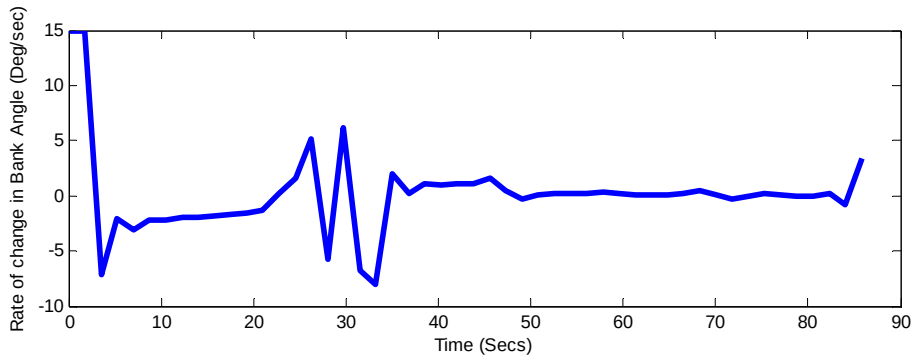

Rate of change in Angle of Attack (Deg/sec)
10
5
0
-5
-10
0
10
20
30
40
50
60
70
80
90
100
Rate of change in Bank Angle (Deg/sec)
15
10
5
0
-5
-10
-15
Time (Secs)
Load Factor for Climb (n)
3
2
1
0
-1
Load Factor for Turn (n)
1.06
1.05
1.04
1.03
1.02
1.01
1
Time (Secs)

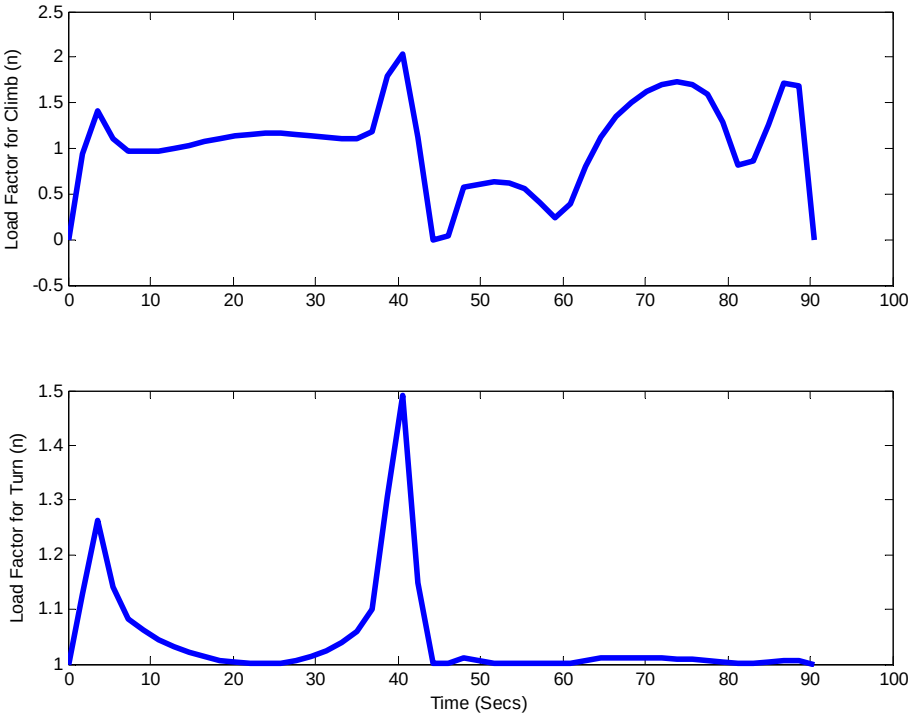
Control plots for Terrain 5 for minimum time



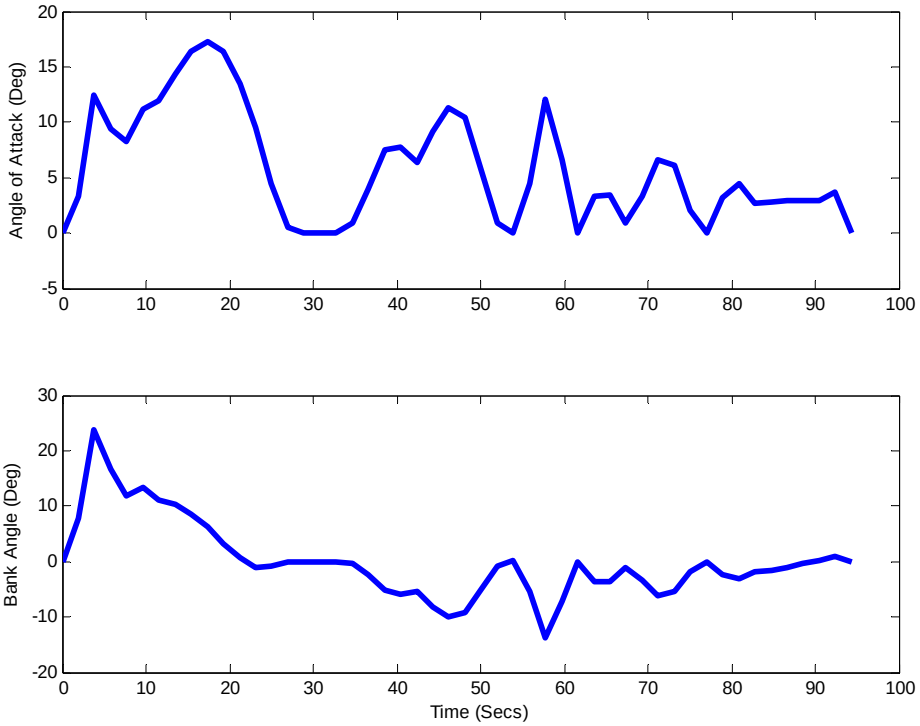


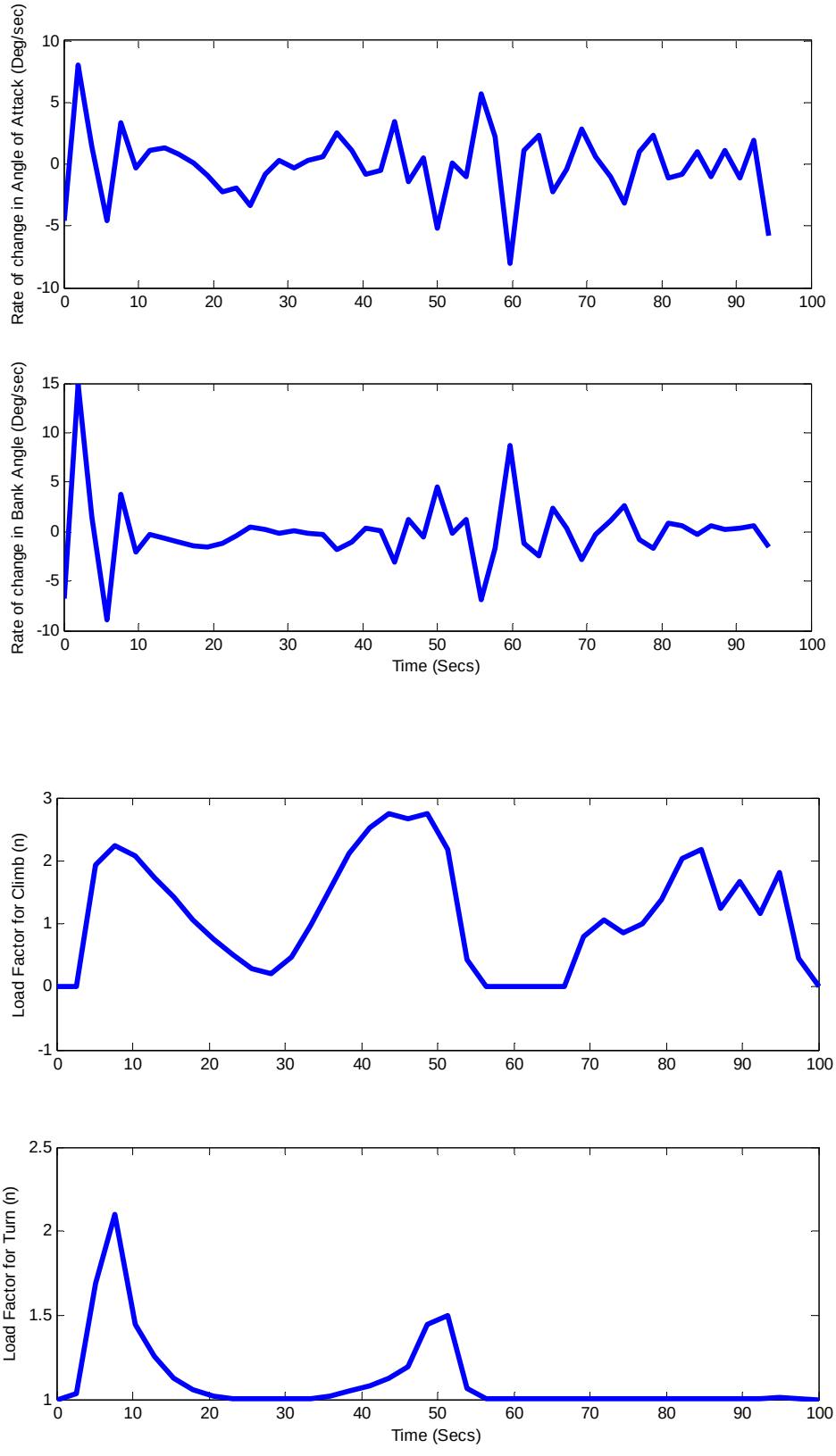




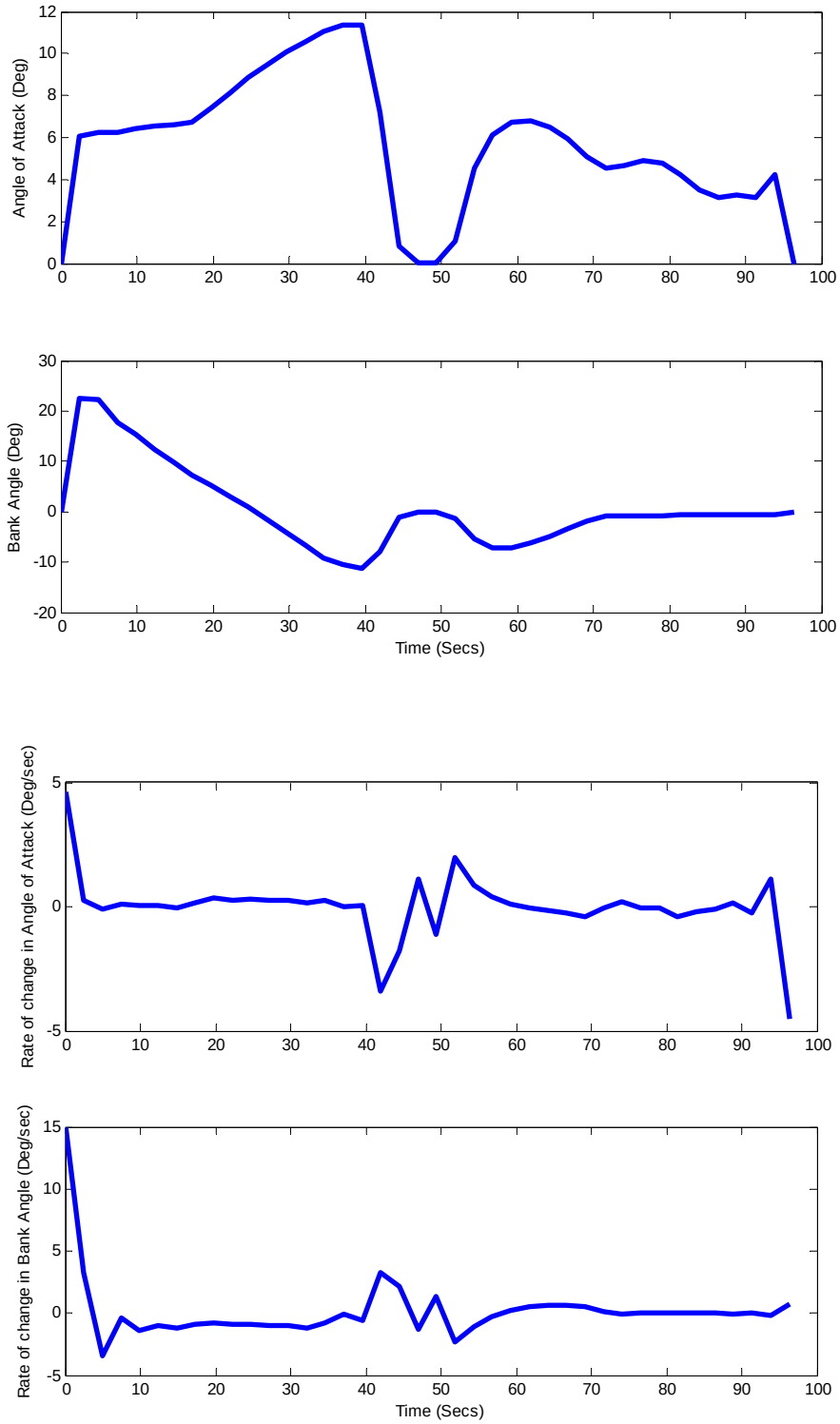


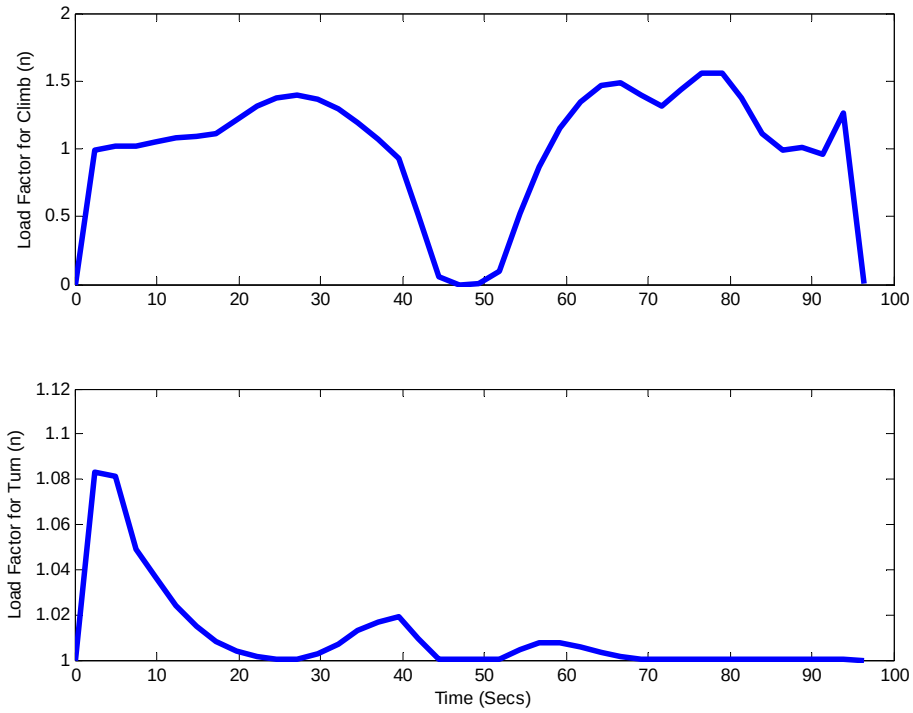
Control plots for Terrain 5 for minimum clearance to terrain



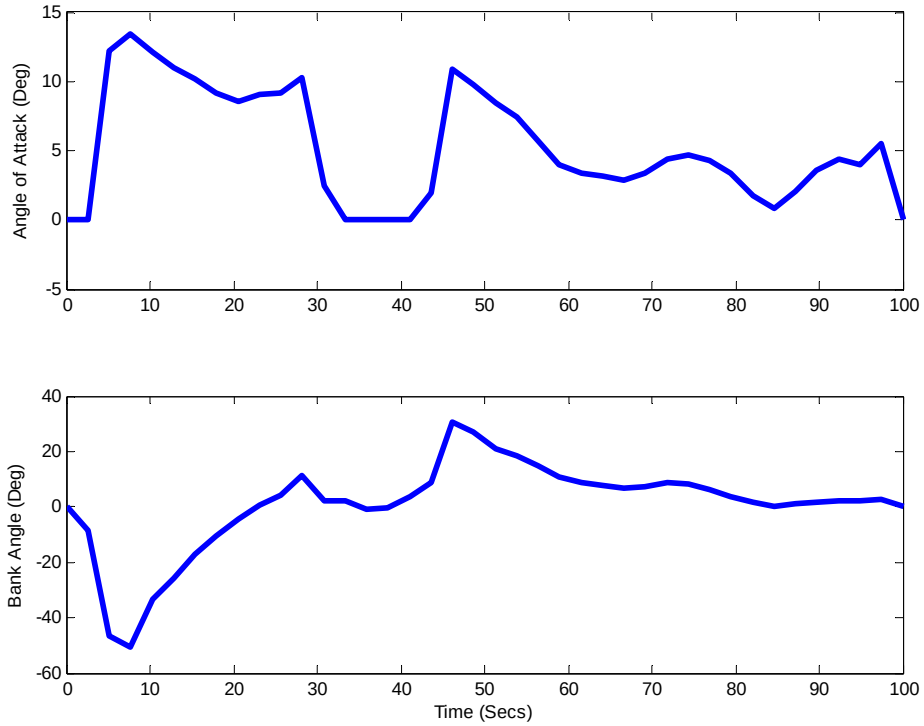

Rate of change in Angle of Attack (Deg/sec)
10
5
0
-5
-10
0
10
20
30
40
50
60
70
80
90
100
Rate of change in Bank Angle (Deg/sec)
15
10
5
0
-5
-10
Time (Secs)
Load Factor for Climb (n)
3
2
1
0
-1
Load Factor for Turn (n)
2.5
2
1.5
1
Time (Secs)

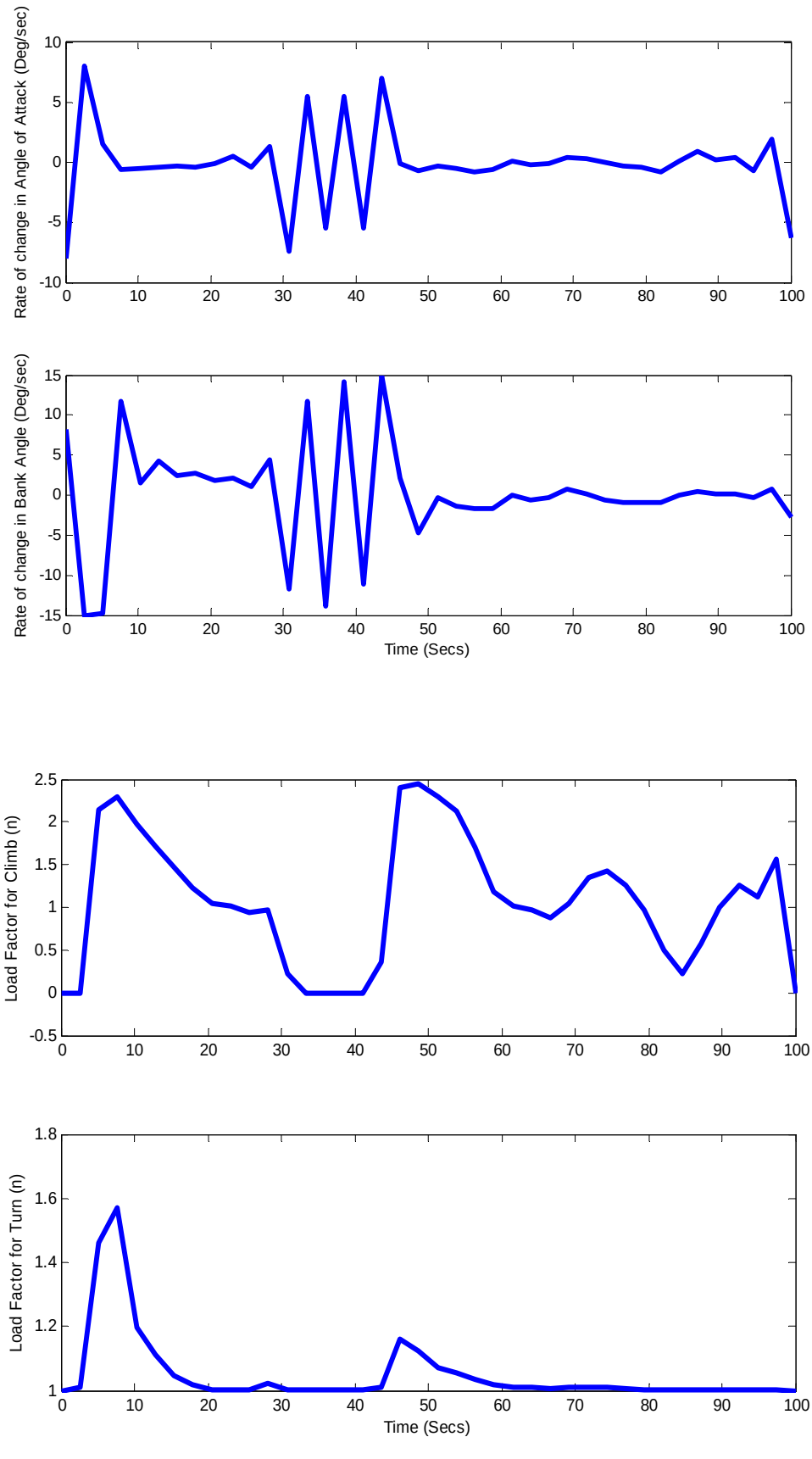
Control plots for Terrain 6 for minimum time





Control plots for Terrain 6 for minimum clearance to terrain




Rate of change in Angle of Attack (Deg/sec)
10
5
0
-5
-10
0
10
20
30
40
50
60
70
80
90
100
Rate of change in Bank Angle (Deg/sec)
15
10
5
0
-5
-10
-15
Time (Secs)
Load Factor for Climb (n)
2.5
2
1.5
1
0.5
0
-0.5
Load Factor for Turn (n)
1.8
1.6
1.4
1.2
1

A.3.1 Direct working files

Cost function file

```
function [Mayer,Integral] = Cost(y,u,p,t)
global cs;
global tref yref;
```

2) **Sensitivity analysis cost function**

```
temp1 = interp1(yref(1,:),yref(2,:),y(1,:),'pchip');
temp2 = interp1(yref(1,:),yref(3,:),y(1,:),'pchip');
Mayer = 0;
Integral = (temp1-y(2,:)).^2+(temp2-y(3,:)).^2;
```

2) **Minimum Time scenario cost function**

```
Mayer = t(end);
Integral = 0;
```

3) **Minimum Clearance to terrain cost function**

```
Mayer = 0 ;
xb = y(1,:); yb = y(2,:); zb = y(3,:);
g = 9.81; vs = 340.3;
```

Make sure the coordinates are scaled properly

```
X = xb*vs^2/g;
Y = yb*vs^2/g;
Z = zb*vs^2/g;

zterrain = diag(fnval({[X],[Y]},cs))';
Integral = ((Z + 50 - zterrain)/1000). ^2 + y (7, :). ^2 + y (8, :). ^2;
```

Boundary Conditions file

```
function BC = BoundaryConditions(y,u,t);
global data;

Put initial boundary conditions here

BC(1) = y(1,1) - data.y0(1);
BC(2) = y(2,1) - data.y0(2);
BC(3) = y(3,1) - data.y0(3);
BC(4) = y(4,1) - data.y0(4);
BC(5) = y(5,1) - data.y0(5);
BC(6) = y(6,1) - data.y0(6);
BC(7) = y(7,1) - data.y0(7);
BC(8) = y(8,1) - data.y0(8);
BC(9) = y(9,1) - data.y0(9);


Final boundary conditions

BC(10) = y(1,end) - data.yf(1);
BC(11) = 0; set free
BC(12) = 0; set free
BC(13) = 0; set free
BC(14) = y(5,end) - data.yf(5);
BC(15) = y(6,end) - data.yf(6);
BC(16) = y(7,end) - data.yf(7);
BC(17) = y(8,end) - data.yf(8);
BC(18) = y(9,end) - data.yf(9);
```

Path Constraint File

```
function g = pathconstraints(y,u,p,t)
global cs;

xb = y(1,:); yb = y(2,:); zb = y(3,:);
g = 9.81; vs = 340.3;
% Make sure the coordinates are scaled properly
X = xb*vs^2/g;
Y = yb*vs^2/g;
Z = zb*vs^2/g;

% now implement the path constraints
% g<=0

% this interpolates the terrain
zterrain = diag(fnval({[X],[Y]},cs))';
% yterrain = diag(fnval({[X],[Z]},cs))';


g=[];
g(1,:) = (zterrain + 50-Z)/1000; %+ (yterrain + 40-Y)/1000;
% g(2,:) = (yterrain + 40-Y)/1000;
% g(2,:) = (Z-zterrain)/1000-p(1);  % for minimax
```

Climb Load Factor

```
% % xb = y(1,:); yb = y(2,:); zb = y(3,:);
v = y(4,:); gamma = y(5,:); psi = y(6,:);

eta   = y(9,:);
alpha = y(7,:);
bank  = y(8,:);
m = 28030;
g0 = 9.81;
vs = 340.3;
S = 49.2;
V = v*vs;
% X = xb*vs^2/g;
% Y = yb*vs^2/g;
% Z = zb*vs^2/g;
rho = 1.225*(1-Z/44.331/1000).^4.256;
q = 0.5*rho.*V.^2;
CLalpha = 3.44+1./cosh((v-1)/0.06).^2;%
CL = CLalpha.*alpha;
CD = 0.013+0.0144*(1+tanh((v-0.98)/0.06))+(0.54+0.15*(1+tanh((v-
0.9)/0.06))).*CLalpha.*alpha.^2;
L = q.*S.*CL;
w = m*g0;
n  = L*(1/w);
```

Turn Load Factor

```
% g(2,:) = n;
g(2,:) = 0 - n;
g(3,:) = n - 9;

% n1 = sqrt((((y(6,:).*y(4,:))/g0).^2) + 1);

n1 = 1./cos(y(8,:));
%
% g(3,:) = n1;
g(4,:) = 0 - n1;
g(5,:) = n1 - 7.5;
```

State Equations file

```
function ydot = states(y,u,p,t)

xb = y(1,:); yb = y(2,:); zb = y(3,:); v = y(4,:); gamma = y(5,:); psi = y(6,:);

eta   = y(9,:);
alpha = y(7,:);
bank  = y(8,:);

g = 9.81; S = 49.2; m = 28030; vs = 340.3;

% Calculate dimensional values

V = v*vs;
X = xb*vs^2/g;
Y = yb*vs^2/g;
Z = zb*vs^2/g;

% Tmax = 16000;
% rho = 1.225;

hT = Z/3048;
Tmax = ((30.21-0.668*hT-6.877*hT.^2+1.951*hT.^3-0.1512*hT.^4) + ...
   v.*(-33.8+3.347*hT+18.13*hT.^2-5.865*hT.^3+0.4757*hT.^4) + ...
   v.^2.*(100.8-77.56*hT+5.441*hT.^2+2.864*hT.^3-0.3355*hT.^4) + ...
   v.^3.*(-78.99+101.4*hT-30.28*hT.^2+3.236*hT.^3-0.1089*hT.^4) + ...
   v.^4.*(18.74-31.6*hT+12.04*hT.^2-1.785*hT.^3+0.09417*hT.^4))*4448.22; % Newton's

rho = 1.225*(1-Z/44.331/1000).^4.256;

q = 0.5*rho.*V.^2;
CLalpha = 3.44+1./cosh((v-1)/0.06).^2;%
CL = CLalpha.*alpha;
CD = 0.013+0.0144*(1+tanh((v-0.98)/0.06))+(0.54+0.15*(1+tanh((v-
0.9)/0.06))).*CLalpha.*alpha.^2;
D = q.*S.*CD;
L = q.*S.*CL;
Thrust = eta.*Tmax;

Wxbar = 0;
Wybar = 0;
Wzbar = 0;
Zs = 1000;
Ax = 0; Ay = 0; DX = 500; DY = 500; Tx = 10; Ty = 20;

% remember that these are defined in terms of dimensional time!

Wx = Wxbar + Ax*Z/Zs.*sin(2*pi*X/DX).*sin(2*pi*Y/DY);
Wy = Wybar + Ay*Z/Zs.*sin(2*pi*X/DX).*sin(2*pi*Y/DY);
Wz = Wzbar*Z/Zs;
```

```
ydot(1,:) = v.*cos(gamma).*cos(psi)+Wx/vs;
ydot(2,:) = v.*cos(gamma).*sin(psi)+Wy/vs;
ydot(3,:) = v.*sin(gamma)+Wz/vs;

Wxdot =
2*pi*Ax/DX*cos(2*pi*X/DX).*sin(2*pi*Y/DY).*Z/Zs.*(V.*cos(gamma).*cos(psi)+Wx) ...

+2*pi*Ax/DY*sin(2*pi*X/DX).*cos(2*pi*Y/DY).*Z/Zs.*(V.*cos(gamma).*sin(psi)+Wy)+
Ax/Zs*sin(2*pi*X/DX).*sin(2*pi*Y/DY).*(V.*sin(gamma)+Wz);
Wydot =
2*pi*Ay/DX*cos(2*pi*X/DX).*sin(2*pi*Y/DY).*Z/Zs.*(V.*cos(gamma).*cos(psi)+Wx) ...

+2*pi*Ay/DY*sin(2*pi*X/DX).*cos(2*pi*Y/DY).*Z/Zs.*(V.*cos(gamma).*sin(psi)+Wy)+
Ay/Zs*sin(2*pi*X/DX).*sin(2*pi*Y/DY).*(V.*sin(gamma)+Wz);
Wzdot = Wzbar/Zs.*(V.*sin(gamma)+Wz);

ydot(4,:) = (Thrust.*cos(alpha)-D)/(m*g)-sin(gamma)-
1/g*(Wxdot.*cos(gamma).*cos(psi)+Wydot.*cos(gamma).*sin(psi)+Wzdot.*sin(gamma));
ydot(5,:) = ((Thrust.*sin(alpha)+L).*cos(bank)/(m*g)-
cos(gamma))./v+1/g*(Wxdot.*sin(gamma).*cos(psi)./v+Wydot.*sin(gamma).*sin(psi)./v-
Wzdot.*cos(gamma)./v);
ydot(6,:) =
(Thrust.*sin(alpha)+L).*sin(bank)./(m*g*v.*cos(gamma))+1/g*(Wxdot.*sin(psi)./(v.*cos(ga
mma))-Wydot.*cos(psi)./(v.*cos(gamma)));
ydot(7,:) = u(2,:)*vs/g;
ydot(8,:) = u(3,:)*vs/g;
ydot(9,:) = u(1,:);
```

Calling File

```
global data cs;
global tref yref;
load reftraj;
data.method = 'control';
% ignore these lines
data.nodes =20;       % Specify equivalent number of nodes for Euler method
data.intervals = 12;

%%%%%
data.n     = 9;   % Give number of state variables
data.m     = 3;   % Give number of control inputs
data.np    = 1;

data.time.range  = [0 100*9.81/340.3]; % Define time range (guess, maybe)
```

In this part of the programme the upper and lower limits of the state equations are added

```
data.states.lower = [-inf -inf 0 50/340.3 -inf -inf -10*pi/180 -70*pi/180 0];
data.states.upper = [inf inf 20000*9.81/340.3^2 inf inf inf 20*pi/180 70*pi/180 1];
```

In this part of the programme, the limits on the rate of change for the controls are added.

```
alphadotmax = 8*pi/180;
bankdotmax = 15*pi/180;

data.control.lower = [-0.25 -alphadotmax -bankdotmax];
data.control.upper = [0.25 alphadotmax bankdotmax];

data.p.lower = [0];
data.p.upper = [100];

data.time.lower = [0 0.001];
data.time.upper = [0 100*9.81/340.3];

% data.path.lower = [-inf -inf 0 -inf 0];
% data.path.upper = [0 0 .9 0 .8];
data.path.lower = [-inf];
data.path.upper = [0 ];
```

In this part of the file, the initial and final conditions of the aircraft are added.

```
data.y0 = [-8000*9.81/340.3^2, -800*9.81/340.3^2, 1500*9.81/340.3^2, 0.5, 0, 0, 0, 0, 0];
% Use nan, find(isnan(y0));
data.yf = [8000*9.81/340.3^2,  0*9.81/340.3^2, 900*9.81/340.3^2, 0.5, 0, 0, 0, 0, 0];
```

```
data.guess = 0;     % 0 = interpolate data given in structure .t,.states,.control
                    % 1 = linear between y0 and yf, or zero if no BC
                    % 2 = integrate states with control given in structures
                    % .t,.control

data.guess.t = [0 100];
data.guess.states(1,:) = [data.y0(1) data.y0(1)];
data.guess.states(2,:) = [data.y0(2) data.y0(2)];
data.guess.states(3,:) = [data.y0(3) data.y0(3)];
data.guess.states(4,:) = [data.y0(4) data.y0(4)];
data.guess.states(5,:) = [data.y0(5) data.y0(5)];
data.guess.states(6,:) = [data.y0(6) data.y0(6)];
data.guess.states(7,:) = [3*pi/180 3*pi/180];
data.guess.states(8,:) = [0*pi/180 0*pi/180];
data.guess.states(9,:) = [0 1];

data.guess.control(1,:) = [1 1];
data.guess.control(2,:) = [0*pi/180 0*pi/180];
data.guess.control(3,:) = [0 0];
data.guess.p = [0];

%get terrain model
cs = terrain;

data.outputscreen = 1;   % 1 = ON, 0 = OFF
data.Jacobian     = 1;
% data.Ne = (10*2-1)*2-1;
data.Ne = 20;

Discretization = ['simpsonc']; data.alpha = 0; data.beta = 0;
Direct_numpar(Discretization);
[y,u,p,t,f,inform,lambda,tlam,nu,mu,H,SolveTime] = Direct(Discretization);

%     for j = 1:2
%      if inform==1
%     break
%     end
%     if inform==3
%     else
%     data.Ne = data.Ne*2;
%     temp =['a' num2str(j)];
%     save(temp);
%     end
%     data.guess.t = t.tx;
%     data.guess.control = u;
%     data.guess.states = y;
%     data.time.range(2)= t.tx(end);
%     Direct_numpar(Discretization);
%     [y,u,p,t,f,inform,lambda,tlam,nu,mu,H,SolveTime] = Direct(Discretization);
% end
% [tout,yout,uout,Error] = feasible(t,y,u,p,'pchip');
```

```
% MaxError = max(max(Error));
% fprintf('Max error in States = %4.5e\n',MaxError);
%  [terror,relerror,meanrelerror,ERROR_RATIO] = StateError(t,y,u,p,'pchip');
```