Report Documentation Page

On-Line Trajectory Optimization for Autonomous Air Vehicles

Technical Abstract

Successful operation of next-generation unmanned air vehicles will demand a high level of autonomy. Autonomous low-level operation in a complex environment dictates a need for on-board, robust, reliable and efficient trajectory optimization. In this report, we develop and demonstrate an innovative combination of traditional analytical and numerical solution procedures to produce efficient, robust and reliable means for nonlinear flight path optimization in the presence of time-varying obstacles and threats. The trajectory generation problem is first formulated as an optimization problem using reduced-order dynamics that result from the natural time-scale separation that exists in the aircraft dynamics. Terrain information is incorporated directly into the formulation of the reduced-order dynamics, which significantly reduces the computational load and leads to a path planning solution that can be implemented in real-time. Various cases of terrain, pop-up obstacles/threats, and targets are simulated. A representative optimal trajectory is generated with in a high fidelity full-order nonlinear aircraft dynamics and compared with a solution obtained from a reduced-order optimization. The developed algorithm is flight demonstrated with a fixed-wing unmanned aircraft test-bed in which a neural network-based adaptive autopilot is integrated with the on-line trajectory optimization algorithm.

List a Maximum of 8 Key Words that Describe the Project

Obstacle Avoidance
Singular Perturbation Theory
Guidance and Control
Trajectory Optimization
Optimal Guidance
Autonomous
Unmanned Air Vehicles
On-Line Path Planning

REPORT DOCUMENTATION PAGE	*Form Approved* *OMB No. 0704-0188*

1. REPORT DATE *(DD-MM-YYYY)* 31-07-2008	**2. REPORT TYPE** Final Technical Report	**3. DATES COVERED** *(From - To)* May 2004-April 2007

4. TITLE AND SUBTITLE

On-Line Trajectory Optimization
for Autonomous Air Vehicles

5a. CONTRACT NUMBER

FA9550-04-C-0046

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

J. Eric Corban, Shannon Twigg, Tobias Ries, Bong-Jun Yang, Eric Johnson, Anthony Calise

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

Guided Systems Technologies, Inc.
P.O. Box 1453
McDonough, Georgia 30253-1453

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

Air Force Office of Scientific Research
AFOSR/ND
875 N.Randolph St, Suite 325, Room 4111
Arlington, VA 22203-1768

10. SPONSOR/MONITOR'S ACRONYM(S)

AFOSR/ND

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

Unclassified Unlimited

13. SUPPLEMENTARY NOTES

14. ABSTRACT

Successful operation of next-generation unmanned air vehicles will demand a high level of autonomy. Autonomous low-level operation in a complex environment dictates a need for on-board, robust, reliable and efficient trajectory optimization. In this report, we develop and demonstrate an innovative combination of traditional analytical and numerical solution procedures to produce efficient, robust and reliable means for nonlinear flight path optimization in the presence of time-varying obstacles and threats. The trajectory generation problem is first formulated as an optimization problem using reduced-order dynamics that result from the natural time-scale separation that exists in the aircraft dynamics. Terrain information is incorporated directly into the formulation of the reduced-order dynamics, which significantly reduces the computational load and leads to a path planning solution that can be implemented in real-time.

15. SUBJECT TERMS

Obstacle Avoidance, Optimal Guidance, Singular Perturbation Theory, Autonomous Guidance and Control, Unmanned Air Vehicles, Trajectory Optimization, and On-Line Path Planning

16. SECURITY CLASSIFICATION OF:			**17. LIMITATION OF ABSTRACT**	**18. NUMBER OF PAGES**	**19a. NAME OF RESPONSIBLE PERSON**
a. REPORT	**b. ABSTRACT**	**c. THIS PAGE**			J. Eric Corban
UU	U	U	Same as Report	91	**19b. TELEPHONE NUMBER** *(Include area code)* 770-898-9100 ext 832

Reset

On-Line Trajectory Optimization for Autonomous Air Vehicles

Final Report
July 31, 2007

Report Number GST-06-1-F

STTR Phase II
Contract FA9550-04-C-0046

Prepared for:

Scott R. Wells, LtCol, USAF, PhD
Program Manager, Dynamics and Controls
Air Force Office of Scientific Research
AFOSR/ND
875 N. Randolph St, Suite 325, Room 4111
Arlington, VA 22203-1768
(703) 696-7796
scott.wells@afosr.af.mil

Prepared by:

J. Eric Corban, Shannon Twigg, Tobias Ries, Bong-Jun Yang, Eric Johnson, Anthony Calise
In Collaboration with the Georgia Institute of Technology and the University of Stuttgart

Guided Systems Technologies, Inc.

P.O. Box 1453
McDonough, Georgia 30253-1453
ec@guidedsys.com
(770) 898-9100, x832

ACKNOWLEDGEMENTS

The work presented herein was funded by the US Air Force Office of Scientific Research under STTR Phase I Contract F49620-02-C-0085 and Phase II Contract FA9550-04-C-0046. The phase I proposal was submitted in response to the 2002 Air Force STTR solicitation topic A02T002 entitled "Fast, Robust Real-Time Trajectory Generation for Autonomous and Semi-Autonomous Nonlinear Flight Systems." The team is especially thankful to Shannon Twigg who generated much of the work presented herein while a graduate student at the Georgia Institute of Technology. Special thanks also to Tobias Reis who contributed the full order optimization results while completing a Masters program and on loan from Prof. Klaus Well of the University of Stuttgart. The work reported herein also depends in part on previous research performed by P.K.A. Menon and his associates. Dr. Menon's discussion of this research topic with team members is much appreciated. Thanks also to Prof. Eric Johnson for his contribution of resources in the Georgia Tech School of Aerospace Engineering UAV lab that produced a flight demonstration of the work.

TABLE OF CONTENTS

1. Introduction 6
1.1 Background 6
1.2 Contributions of the PHASE II Efforts 9

2. Reduced-Order Formulation 13
2.1 Local Tangent Plane Equations of Motion 14
2.1.1 Legendre-Clebsch Necessary Condition 20
2.1.2 Weierstrass Test 21
2.2 Simplified Equations of Motion 22
2.2.1 Legendre-Clebsch Necessary Condition 24
2.2.2 Weierstrass Test 24
2.3 Interior point constraints (way points) 24
2.3.1 Legendre-Clebsch Necessary Condition 28
2.3.2 Weierstrass Test 28
2.4 Expansion to 3-D formulations 29

3. Six DOF Optimization by GESOP 31
3.1 Modeling 31
3.2 Issues for comparison with 3-D results 33

4. Numerical Procedure for Optimal Trajectories 35
4.1 Optimization Procedure for the reduced-order formulation 35
4.2 Optimization Procedure for GESOP 38

5. Numerical Results 40
5.1 Terrain Data 40
5.2 Pseudo 3-D and 3-D results 42
5.3 Six DOF optimal trajectories using GESOP 49
5.4 Comparisons between Pseud 3-D results and GEsOP solutions with Simulated hills 52
5.4.1 Flight around three mountains 52
5.4.2 Flight around Five mountains 53
5.4.3 Flights With Moving Targets and Threats 55
5.5 Comparisons between Pseud 3-D results and GEsOP solutions with the real terrain 58
5.5.1 Terrain-Masking (K=1) 58
5.5.2 Optimization of time (K=0) 59
5.5.3 Flight over terrain for a Threat 62
5.5.4 Flights with a Pop-UP Threat 64
5.5.5 Earlier pop-up Threat 67

6. Flight Demonstrations....69

6. 1 Integration Details....69

6.2 Simulation....70

6.3 Experimental Results....73

6.4 Relavant Files....73

7. Conclusions and Recommendations....82

References....87

1. Introduction

1.1 Background

High-flying unmanned reconnaissance and surveillance systems are now being used extensively in the United States military. Current development programs will soon produce demonstrations of next-generation unmanned flight systems that are designed to perform combat missions. In practice, these vehicles must achieve a high level of autonomy in operations to be successfully deployed in large numbers. Their use in first-strike combat operations will dictate operations in densely cluttered environments that include unknown obstacles and threats, and will require the use of terrain for masking. The demand for autonomy of operations in such environments dictates the need for an on-board trajectory optimization capability.

In any given mission scenario, the initial conditions, requirements, objectives, vehicle dynamics, and constraints can be analyzed at varying levels of detail to produce a number of feasible flight plans. However, in most situations conflicts will arise between the cited factors that necessitate trade-off of one for the other. Ad hoc methods of solution can be constructed for simple missions with relatively few constraints, but such methods are quickly overwhelmed as problem complexity grows. In such cases formal methods for trajectory optimization are needed to identify a dynamically feasible solution that is “best” in some sense.

Essentially all numerical methods for solving trajectory optimization problems incorporate nonlinear programming (NLP) methods. Depending on how NLPs are employed, methods available for a solution of trajectory optimization problems generally fall into two distinct categories: direct and indirect[1]. Indirect methods are characterized by explicitly solving the optimality conditions stated in terms of the adjoint differential equations, the Pontryagin's maximum principle, and associated boundary (transversality) conditions[2]. Using the calculus of variations, the necessary conditions are derived by setting the first variation of the Hamiltonian function to zero. The indirect approach usually requires the solution of a nonlinear multi-point boundary value problem. An indirect method for optimizing a function of n variables would require analytically computing the gradient and then locating a set of variables using a root-finding algorithm such that the gradient is zero. In contrast, direct methods do not require an analytic expression for the necessary condition and typically does not require initial guesses for the adjoint variables. Instead, the dynamic state and control variables are adjusted to directly optimize the objective function[3]. All direct methods introduce some parametric representation for the control variables which in general leads to larger number of variables in NLP.

While great success has been achieved in numerically solving complex nonlinear trajectory optimization problems using direct and indirect techniques, many short comings still exist. Direct optimization schemes tend to be tolerant of a poor initial guess but are in general slow to converge. Indirect methods, in contrast, can usually achieve greater accuracy in a few iterations but also exhibit much greater sensitivity to the initial guess. Essentially all of the available methods in both categories demand computational resources that until recently were simply unavailable in a flight environment. For complex problem formulations with full vehicle dynamics, most have run times that preclude real-time mission updates. Almost all suffer from great complexity of implementation.

For these reasons, past aircraft trajectory generation studies were rooted in classical optimization theory and the calculus of variations. A great variety of practical methods for optimizing the flight of aircraft both in two and three dimensions were developed in the 1960s and 70s. Much of this work began with the studies performed by Bryson based on minimizing time, fuel or range using reduced order modeling methods[4, 5]. Subsequently, the use of singular perturbation methods, and multi-time scale analysis for trajectory optimization were suggested by Kelley[6, 7], and later extensively developed and applied by Calise [8-11], Ardema [12] and others. In the work of Calise, the emphasis was on obtaining analytic or near-analytic feedback solutions that could be implemented with very limited computing resources in real-time.

In this research, we pay attention to natural time scale separation that exists in aircraft dynamics and attempt to exploit two-time scales in on-line trajectory optimization. The overall approach follows that in [13] in which a reduced-order terrain-following optimization problem is formulated for the nap-of-the-Earth guidance of helicopters under the assumption that terrain-following dynamics are effectively described by pseudo 3-D dynamics. Relying upon the time-scale separation assumption, we address various aspects of path-planning such as moving threats, interior point constraints, and pop-up obstacles besides terrain-masking. As a next step, we implement the resulting optimal trajectory from the reduced-order dynamics in a high fidelity flight simulator and assess the resulting trajectory flown in the high-fidelity simulator. To be precise, we compare the trajectory from the reduced-order formulation to both the trajectory followed in the high-fidelity simulator for the reduce-order solution and the optimal solution for full order 6DOF aircraft dynamics. In that procedure, we try to reveal various aspects that should be taken into account for the reduced-order optimal trajectory to be feasible with full-order 6 DOF vehicle dynamics. The report includes flight-test validation to demonstrate on-line re-planning of flight trajectory when a simulated pop-up obstacle appears.

1.2 Contributions of the PHASE II Efforts

This research expands on the work done by Menon and Kim[13]. The rationale in this research is to resort to analytical solutions to maximal degree possible for numerical efficiency. With this philosophy, it was decided to continue investigating using a reduced order formulation initiated in [13] and exploit the benefits of analytical methods of solving the optimal path planning problem before it became necessary to use any numerical methods. In most numerical methods, the time step for the discretization phase usually must be very small to avoid loss of information. This leads to very long solving times. Also, the graphical methods of solving this type of problem rarely lead to an optimal solution.

An underlying theory for the reduced-order formulation is singular perturbation. In general, singularly perturbed dynamic systems are characterized by a small parameter multiplying the derivatives of some state vector components. If the small parameter is set to zero, the order of the dynamic system is reduced. The solution to the reduced order problem that results when the parasitic small parameter is set to zero provides the leading term in a perturbation series. Of course, the reduced order solution is not able to satisfy all the boundary conditions of the original full order problem. Depending on the degree of sub-optimality of the reduced-order solution due to the effect of neglected dynamics, corrections can be made by constructing boundary layer transients that allow rapid variation of the fast states on a stretched time scale[14]. The boundary layer solutions are required to satisfy the boundary conditions violated by the reduced solution and to approach the reduced solution asymptotically. Natural time scale separation that exists in aircraft dynamics allows the successful application of singular perturbation methods to problems in atmospheric flight path optimization [15]. The reduced and boundary layer problems are often solvable using analytic methods alone. This has made a unified analysis of 3-D high-performance aircraft path optimization possible [16].

The components of the research presented here include:

- Two pseudo 3-D formulations that include wind effects, moving targets, interior point constraints in the form of waypoints, and moving threats.
- Derivation for second order variation conditions for each formulation which further assists finding the optimal initial conditions
- Expansion of pseudo 3-D equations of motion to 3-D equations of motion
- Further expansion of pseudo 3-D equations of motion to 3-D ones with addition of a velocity as a state. This constitutes a 3-D varying velocity formulation.

- Evaluations for the ability to track the pseudo 3-D optimal trajectories in a high-fidelity nonlinear simulation of 6 degree-of-freedom (DOF) high-performance aircraft operating at low altitude over simulated and real terrains with both stationary and moving pop-up threats
- Construction for the 6 DOF full-order optimal trajectories using GESOP (Graphical Environment for Simulation and Optimization), the optimization software provided by University of Stuttgart.
- Comparison and analysis of the full-order and pseudo 3-D optimal trajectories
- Flight demonstration of guided flight of an unmanned aerial vehicle (UAV) while employing the real-time path optimization algorithm by introducing virtual terrains and obstacles

While applying corrections for the optimal trajectory resulting from pseudo 3-D formulation by the method of matched asymptotic expansion as in [17] were proposed in the phase II proposal, the resulting optimal trajectories from pseudo 3-D and full 3-D formulations did not exhibit significant differences, both of which resulted in trajectories that exhibit discrepancies from the full-order GESOP optimal solutions. Various issues for comparing pseudo 3-D solutions and those of 6 DOF optimization are extensively addressed in [18] the work of which was supported by GST for this Phase II effort and performed at the Georgia Institute of Technology under the direction of Prof. Anthony J. Calise. The analysis using the singular perturbation method for full 6 DOF dynamics, including engine modeling and aerodynamic coefficients, is considerably complex and has not been addressed yet in the literature of optimal path planning for aerial vehicles that generally assumes a point mass model for optimization problem. This task, if undertaken, would require extremely complex algebra in order to obtain costate equations. Therefore, instead of attempting to carry out complex singular perturbation-based 6 DOF optimization analysis, we converted our attention to additional aspects of pseudo 3-D formulation, such as multi-vehicle formulation and efficient numerical algorithms for the case of increased search dimension due to multi-vehicle formulations. The results from this research constitutes the thesis of Shannon Twigg [19]. So additional contributions from the Phase II efforts are:

- Expansion of pseudo-3D and 3D equations of motion to handle cooperative path planning for multi-vehicles.

- Application of genetic algorithms (GA) to solve multiple initial conditions. The GA employed in this work is different from conventional ones as in [1] in a sense that GA is only used to set proper initial conditions. With a reasonably small number of variables to be solved, the employed algorithm resulted in run-time that is comparable to the variable sweep method for a single variable.

2. Reduced-Order Formulation

In this section, we provide two pseudo 3-D formulations and two full 3-D formulations extensively studied in the Phase II efforts. For ease of understanding, we concentrate our presentation on those of pseudo 3-D formulations. Full details on expansion of the methodology to full 3-D formulations and that of multiple vehicles are referred to the thesis[19] and related papers[20-24]. Figure 2.1 depicts a sample terrain profile with the X-Y-H coordinate system and a local x_1-y_1-z_1 coordinate system. The moving local coordinate system has its origin on the terrain surface at a current x, y position with the x_1-y_1 plane being the tangent plane.

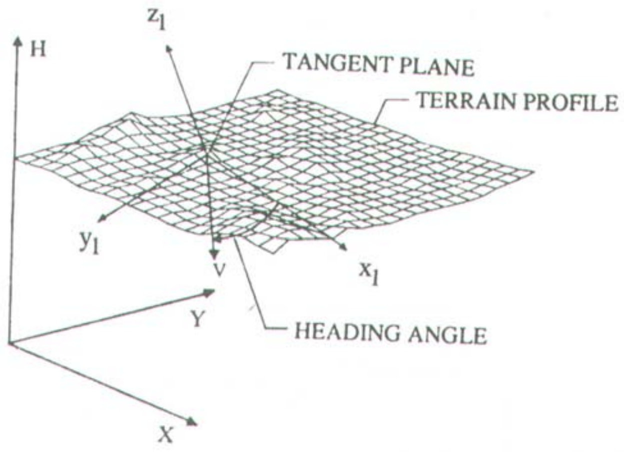


Figure 2.1: Relationship between Inertial Frame and Local Tangent Plane.

2.1 Local Tangent Plane Equations of Motion

The local tangent plane formulation incorporates the constraint that the vehicle flies tangentially to the local terrain directly into the equations of motion and can be written as

$$\dot{x} = \frac{V\cos\psi}{A_1} + \frac{Vf_x f_y \sin\psi}{A_1 A_2} + u(x, y) \tag{2.1}$$

$$\dot{y} = \frac{-VA_1 \sin\psi}{A_2} + v(x, y) \tag{2.2}$$

where *x* and *y* are the north and east components, respectively. *V* is the total aircraft velocity while *u* and *v* are the wind velocities in the x and y-directions, respectively. The heading of the vehicle is represented by ψ -- the heading angle measured with respect to the local tangent plane. Also, f_x and f_y are the partial derivatives of the terrain profile. A_1 and A_2 are given by

$$A_1 = \sqrt{1 + f_x^2} \tag{2.3}$$

$$A_2 = \sqrt{1 + f_x^2 + f_y^2} \tag{2.4}$$

The cost function for this problem can be seen in the following equation.

$$J = \int_0^{t_f} \left[(1 - K) + Kg(x, y, t)\right] dt \tag{2.5}$$

In this equation, the combined threat and terrain function, *g(x,y,t)*, is given as a function of time as well as the position and can be defined as follows.

$$g(x,y,t) = f(x,y) + f_T(x,y,t) \tag{2.6}$$

Here, *f(x,y)* is the function for the terrain profile and $f_T(x,y,t)$ is the function denoting the moving threat. The weighting parameter, *K*, can vary between 0 and 1 and determines the relative importance of time and terrain masking/threat avoidance used in the optimization. When *K* = 0, the equations are optimized with respect to time. When *K* is set to 1, the path is optimized with respect to the threats and the terrain. The Hamiltonian equation can then be given as

$$H = A_4 + \lambda_x\left[\frac{V\cos\psi}{A_1} + \frac{Vf_x f_y \sin\psi}{A_1 A_2} + u\right] + \lambda_y\left[\frac{-VA_1\sin\psi}{A_2} + v\right] \tag{2.7}$$

In this expression, λ_x and λ_y are the costate equations and A_4 can be seen in the following equation.

$$A_4 = 1 - K + Kg(x,y,t) \tag{2.8}$$

The moving threat and target equations of motion are, respectively:

$$\begin{aligned} \dot{x}_T &= V_T\cos\psi_T \\ \dot{y}_T &= V_T\sin\psi_T \end{aligned} \tag{2.9}$$

$$\begin{aligned} \dot{x}_{Tg} &= V_{Tg}\cos\psi_{Tg} \\ \dot{y}_{Tg} &= V_{Tg}\sin\psi_{Tg} \end{aligned} \tag{2.10}$$

In each expression, it is assumed that the respective velocity and heading angle are known at all times. The moving target then results in a new boundary condition.

$$\Psi(t_f) = \begin{bmatrix} x(t) - x_{Tg}(t) \\ y(t) - y_{Tg}(t) \end{bmatrix}_{t=t_f} \tag{2.11}$$

In this expression, it can be seen that $\Psi(t_f)$ has an explicit dependence on the final time as a consequence of the fact that the target coordinates are assumed to satisfy equation (2.10). Therefore, for a free final time, the Hamiltonian equation satisfies

$$H(t_f) = -\lambda^T \left[\frac{\partial \Psi}{\partial t}\right]_{t=t_f} = V_{Tg}\left[\lambda_x \cos\psi_{Tg} + \lambda_y \sin\psi_{Tg}\right]_{t=t_f} \tag{2.12}$$

Due to the moving threat, the Hamiltonian equation, (2.7), is explicitly dependent on time. Given this, the optimality condition for a solution along an extremal arc shows that

$$\dot{H} = \frac{\partial H}{\partial t} = Kg_t \tag{2.13}$$

where g_t denotes the partial derivative of the penalty function with respect to time. Assuming that the threat is constant when expressed in a coordinate system that is attached to the moving threat, then

$$g(x, y, t) = g[x - x_T(t), y - y_T(t)] \tag{2.14}$$

with the threat coordinates satisfying (2.9). Thus

$$\dot{H} = -KV_T\left(g_x \cos\psi_T + g_y \sin\psi_T\right) \tag{2.15}$$

Because the final time is free, the boundary condition for this expression is defined in (2.12).

The optimality condition for this problem is defined as

$$H_\psi = 0 \tag{2.16}$$

Evaluating this expression results in the following relationship

$$\lambda_y = \lambda_x \left[\frac{V f_x f_y \cos\psi}{A_1 A_2} - \frac{V \sin\psi}{A_1} \right] \frac{A_2}{V A_1 \cos\psi} \tag{2.17}$$

Equation (2.17) can then be substituted into the Hamiltonian equation, (2.7), to determine equations defining the two costates, λ_x and λ_y as follows.

$$\lambda_x = \frac{-\left(A_4 - H\right) A_1^2 \cos\psi}{Den} \tag{2.18}$$

$$\lambda_y = \frac{\left(A_4 - H\right) A_2 \sin\psi - \left(A_4 - H\right) f_x f_y \cos\psi}{Den} \tag{2.19}$$

where

$$Den = VA_1 + A_1^2 u \cos\psi + f_x f_y v \cos\psi - A_2 v \sin\psi \tag{2.20}$$

These new expressions for the costates can then be inserted into (2.12) to result in a new boundary condition for the Hamiltonian at the final time.

$$H\left(t_f\right) = \left[\frac{V_{Tg} A_4 \left(A_1^2 \cos\psi \cos\psi_{Tg} + f_x f_y \cos\psi \sin\psi_{Tg} - A_2 \sin\psi \sin\psi_{Tg}\right)}{V_{Tg}\left(A_1^2 \cos\psi \cos\psi_{Tg} + f_x f_y \cos\psi \sin\psi_{Tg} - A_2 \sin\psi \sin\psi_{Tg}\right) - Den} \right]_{t=t_f} \tag{2.21}$$

Differential equations for the costates can be found using

$$\begin{aligned} \dot{\lambda}_x &= -H_x \\ \dot{\lambda}_y &= -H_y \end{aligned} \tag{2.22}$$

This yields

$$\dot{\lambda}_x = -Kg_x - \lambda_x \left[\frac{D_2 \cos\psi + D_3 \sin\psi + D_1 u_x}{D_1} \right] - \lambda_y \left[\frac{D_4 \sin\psi + D_1 v_x}{D_1} \right] \tag{2.23}$$

$$\dot{\lambda}_y = -Kg_y - \lambda_x\left[\frac{D_5\cos\psi + D_6\sin\psi + D_1 u_y}{D_1}\right] - \lambda_y\left[\frac{D_7\sin\psi + D_1 v_y}{D_1}\right] \tag{2.24}$$

where

$$D_1 = A_1^3 A_2^3 \tag{2.25}$$

$$D_2 = -Vf_x f_{xx} A_2^3 \tag{2.26}$$

$$D_3 = VA_1^2 A_2^2 B_2 - VA_2^2 f_x^2 f_y f_{xx} - VA_1^2 B_1 f_x f_y \tag{2.27}$$

$$D_4 = VA_1^4 B_1 - VA_1^2 A_2^2 f_x f_{xx} \tag{2.28}$$

$$D_5 = -Vf_x f_{xy} A_2^3 \tag{2.29}$$

$$D_6 = VA_1^2 A_2^2 B_3 - VA_2^2 f_x^2 f_y f_{xy} - VA_1^2 B_4 f_x f_y \tag{2.30}$$

$$D_7 = VA_1^4 B_4 - VA_1^2 A_2^2 f_x f_{xy} \tag{2.31}$$

$$B_1 = f_x f_{xx} + f_y f_{xy} \tag{2.32}$$

$$B_2 = f_x f_{xy} + f_y f_{xx} \tag{2.33}$$

$$B_3 = f_x f_{yy} + f_y f_{xy} \tag{2.34}$$

$$B_4 = f_x f_{xy} + f_y f_{yy} \tag{2.35}$$

Next, the time derivative of either equation (2.18) or (2.19) is taken and set equal to its counterpart in equation (2.23) or (2.24). This expression can then be solved for the derivative of the heading angle such that

$$\dot{\psi} = \frac{T_1 + T_2 u + T_3 v + T_4 u_x + T_5 u_y + T_6 v_x + T_7 v_y}{T_8} \tag{2.36}$$

where

$$T_1 = -KVS_1 + V\left(A_4 - H\right)S_2 \tag{2.37}$$

$$T_2 = -KS_3 + \left(A_4 - H\right)S_4 \tag{2.38}$$

$$T_3 = KS_5 + \left(A_4 - H\right)S_6 \tag{2.39}$$

$$T_4 = \left(A_4 - H\right)S_7 \tag{2.40}$$

$$T_5 = \left(A_4 - H\right)S_8 \tag{2.41}$$

$$T_6 = \left(A_4 - H\right)S_9 \tag{2.42}$$

$$T_7 = \left(A_4 - H\right)S_{10} \tag{2.43}$$

$$T_8 = \left(A_4 - H\right)A_1^3 A_2^2 \tag{2.44}$$

$$S_1 = A_1^2 A_2 \left[A_2 g_x \sin\psi + \left(g_y + f_x^2 g_y - g_x f_x f_y\right)\cos\psi\right] \tag{2.45}$$

$$S_2 = \left(f_x f_y^2 f_{xx} - A_1^2 f_y f_{xy}\right)\sin\psi + A_2 f_y f_{xx} \cos\psi \tag{2.46}$$

$$S_3 = A_1^3 A_2 \left[A_2 g_x \sin\psi + \left(g_y + f_x^2 g_y - g_x f_x f_y\right)\cos\psi\right]\cos\psi \tag{2.47}$$

$$\begin{aligned} S_4 = {} & A_1 A_2 \left(A_1^2 f_x f_{xy} - f_x^2 f_y f_{xx} + f_y f_{xx}\right)\cos^2\psi + \\ & + A_1\left(A_1^2 f_x f_{xx} - A_1^2 f_y f_{xy} + 2 f_x f_y^2 f_{xx}\right)\sin\psi\cos\psi \end{aligned} \tag{2.48}$$

$$\begin{aligned} S_5 = {} & A_1 A_2^2 \left(A_1^2 g_y - 2 g_x f_x f_y\right)\sin\psi\cos\psi + \\ & + A_1 A_2^3 g_x \sin^2\psi + A_1 A_2 \left(g_x f_x^2 f_y^2 - A_1^2 f_x g_y f_y\right)\cos^2\psi \end{aligned} \tag{2.49}$$

$$S_6 = A_1 A_2 \left(f_y f_{xy} - f_x^2 f_y f_{xy} + A_1^2 f_x f_{yy} \right) \cos^2 \psi + \\ + A_1 \left(2 f_x f_y^2 f_{xy} + A_1^2 f_x f_{xy} - A_1^2 f_y f_{yy} \right) \sin \psi \cos \psi \tag{2.50}$$

$$S_7 = A_1^3 A_2 \left(A_2 \sin \psi - f_x f_y \cos \psi \right) \cos \psi \tag{2.51}$$

$$S_8 = A_1^5 A_2 \cos^2 \psi \tag{2.52}$$

$$S_9 = A_1 A_2^2 \left(2 f_x f_y \sin \psi \cos \psi - A_2 \right) + A_1 A_2 \left(A_2^2 - f_x^2 f_y^2 \right) \cos^2 \psi \tag{2.53}$$

$$S_{10} = A_1^3 A_2 \left(f_x f_y \cos \psi - A_2 \sin \psi \right) \cos \psi \tag{2.54}$$

This solution consists of four differential equations -- *x*, *y*, *H* and ψ -- and requires two initial conditions to be found -- *H* and ψ. The final value of the Hamiltonian is known, via equation (2.21). The solution is reached when the final values of the Hamiltonian and position are met and the cost is minimized. When there are no moving threats, the Hamiltonian is constant in value – so there are only three differential equations – and the final value is still known. When there is no moving target, the final value of the Hamiltonian is zero.

2.1.1 Legendre-Clebsch Necessary Condition

The Hamiltonian equation for the local tangent plane equations of motion is

$$H = A_4 + \lambda_x \left[\frac{V \cos \psi}{A_1} + \frac{V f_x f_y \sin \psi}{A_1 A_2} \right] + \lambda_y \left[\frac{-V A_1 \sin \psi}{A_2} \right] \tag{2.55}$$

and the algebraic equations for the costates are

$$\begin{aligned}\lambda_x &= \frac{-\left(A_4 - H\right)A_1 \cos\psi}{V} \\ \lambda_y &= \frac{\left(A_4 - H\right)\left(A_2 \sin\psi - f_x f_y \cos\psi\right)}{VA_1}\end{aligned} \tag{2.56}$$

Since there is only one control, the second partial derivative of the Hamiltonian with respect to the heading angle is a scalar value and is represented by

$$H_{uu} = \lambda_x \left[\frac{-V\cos\psi}{A_1} - \frac{Vf_x f_y \sin\psi}{A_1 A_2}\right] + \lambda_y \left[\frac{VA_1 \sin\psi}{A_2}\right] \tag{2.57}$$

Substituting in the equations for the optimal costates will result in

$$H_{uu} = \left(A_4 - H\right) \geq 0 \tag{2.58}$$

This condition must always be satisfied.

2.1.2 Weierstrass Test

The variational Hamiltonian can be found by substituting the costate equations from (2.56) for the optimal path into the Hamiltonian equation from (2.55) evaluated for any path. This yields

$$\begin{aligned}H(\psi) = A_4 + \left[\frac{-\left(A_4 - H\right)A_1 \cos\psi^o}{V}\right]\left[\frac{V\cos\psi}{A_1} + \frac{Vf_x f_y \sin\psi}{A_1 A_2}\right] + \\ \left[\frac{\left(A_4 - H\right)\left(A_2 \sin\psi - f_x f_y \cos\psi^o\right)}{VA_1}\right]\left[\frac{-VA_1 \sin\psi}{A_2}\right]\end{aligned} \tag{2.59}$$

This can be simplified to

$$H(\psi) = \left(A_4 - H\right)\left[1 - \cos\left(\psi^o - \psi\right)\right] \geq 0 \tag{2.60}$$

which will always be satisfied if equation (2.58) is satisfied.

2.2 Simplified Equations of Motion

The equations of motion used in the simplified formulation are:

$$\dot{x} = V\cos\psi + u(x,y) \tag{2.61}$$

$$\dot{y} = V\sin\psi + v(x,y) \tag{2.62}$$

These equations are written in the local level plane and neglect the effects of the terrain slope. The cost equation for this case is the same as earlier and can be found in equation (2.5). The corresponding Hamiltonian equation is therefore

$$H = A_4 + \lambda_x\left[V\cos\psi + u\right] + \lambda_y\left[V\sin\psi + v\right] \tag{2.63}$$

The equations governing the moving target and moving threat can be seen above in equations (2.9) and (2.10). Evaluating the optimality condition stated in equation (2.16) for this formulation results in the expression

$$\lambda_y = \lambda_x \frac{\sin\psi}{\cos\psi} \tag{2.64}$$

Substituting this into the Hamiltonian equation results in the following costate equations

$$\lambda_x = \frac{-\left(A_4 - H\right)\cos\psi}{V + u\cos\psi + v\sin\psi} \tag{2.65}$$

$$\lambda_y = \frac{-\left(A_4 - H\right)\sin\psi}{V + u\cos\psi + v\sin\psi} \tag{2.66}$$

Therefore, the Hamiltonian evaluated at the final time will be

$$H\left(t_f\right) = \left[\frac{V_{Tg}A_4\cos\left(\psi - \psi_{Tg}\right)}{V_{Tg}\cos\left(\psi - \psi_{Tg}\right) - \left(V + u\cos\psi + v\sin\psi\right)}\right]_{t=t_f} \tag{2.67}$$

The costate differential equations can then be found to be

$$\dot{\lambda}_x = -H_x = -Kg_x - \lambda_x u_x - \lambda_y v_x \tag{2.68}$$

$$\dot{\lambda}_y = -H_y = -Kg_y - \lambda_x u_y - \lambda_y v_y \tag{2.69}$$

As before, the time derivative of (2.65) or (2.66) is found and equated to either (2.68) or (2.69). This expression can then be rearranged to result in the following heading differential equation.

$$\dot{\psi} = \frac{R_1 + R_2 u + R_3 v + R_4\left(u_x - v_y\right) + R_5 u_y + R_6 v_x}{R_7} \tag{2.70}$$

with

$$R_1 = KV\left(g_y \cos\psi - g_x \sin\psi\right) \tag{2.71}$$

$$R_2 = K\left(g_y \cos\psi - g_x \sin\psi\right)\cos\psi \tag{2.72}$$

$$R_3 = K\left(g_y \cos\psi - g_x \sin\psi\right)\sin\psi \tag{2.73}$$

$$R_4 = \left(A_4 - H\right)\sin\psi\cos\psi \tag{2.74}$$

$$R_5 = -\left(A_4 - H\right)\cos^2\psi \tag{2.75}$$

$$R_6 = \left(A_4 - H\right)\sin^2\psi \tag{2.76}$$

$$R_7 = \left(A_4 - H\right) \tag{2.77}$$

Again, the inclusion of a moving target and moving threat results in a system of four differential equations with two initial parameters to be found.

2.2.1 Legendre-Clebsch Necessary Condition

The Hamiltonian equation for the simplified equations of motion is

$$H = A_4 + \lambda_x \left[V \cos\psi\right] + \lambda_y \left[V \sin\psi\right] \tag{2.78}$$

and the costate equations are

$$\begin{aligned} \lambda_x &= \frac{-\left(A_4 - H\right)\cos\psi}{V} \\ \lambda_y &= \frac{-\left(A_4 - H\right)\sin\psi}{V} \end{aligned} \tag{2.79}$$

The partial derivative of the Hamiltonian equation with respect to the heading angle is

$$H_{uu} = \left(A_4 - H\right) \geq 0 \tag{2.80}$$

whish must always be satisfied.

2.2.2 Weierstrass Test

Using the Hamiltonian equation in (2.78) and the costate equations in (2.79), the variational Hamiltonian can be written as

$$H = A_4 + \left[\frac{-\left(A_4 - H\right)\cos\psi^o}{V}\right]\left[V \cos\psi\right] + \left[\frac{-\left(A_4 - H\right)\sin\psi^o}{V}\right]\left[V \sin\psi\right] \tag{2.81}$$

This equation can be reduced to

$$H(\psi) = \left(A_4 - H\right)\left[1 - \cos\left(\psi^o - \psi\right)\right] \geq 0 \tag{2.82}$$

which will always be satisfied if equation (2.80) is satisfied.

2.3 Interior point constraints (way points)

One useful problem variation that can also be investigated is the implementation of interior point constraints during the flight. For this problem, the constraints will be implemented in the form of waypoints where a specific position is required in the middle of the flight. There can be n-number of waypoints during this flight, such that each waypoint – with a given x and y position – is reached at an unspecified time, t_i, in a specified order before ending at the specified final position.

In this type of problem, there are certain constraints on the costates and Hamiltonian that must be fulfilled at the interior points. They include

$$\begin{aligned}
\lambda_x(t_i^+) &= \lambda_x(t_i^-) + \upsilon_{1i} \\
\lambda_y(t_i^+) &= \lambda_y(t_i^-) + \upsilon_{2i} \\
H(t_i^+) &= H(t_i^-) = 0
\end{aligned} \tag{2.83}$$

This implies that the value of each of the costates will jump at each waypoint while the Hamiltonian will remain constant. Because of that, the heading angle will also jump at each waypoint. This will result in a trajectory such as that seen in Figure 2.2. In this

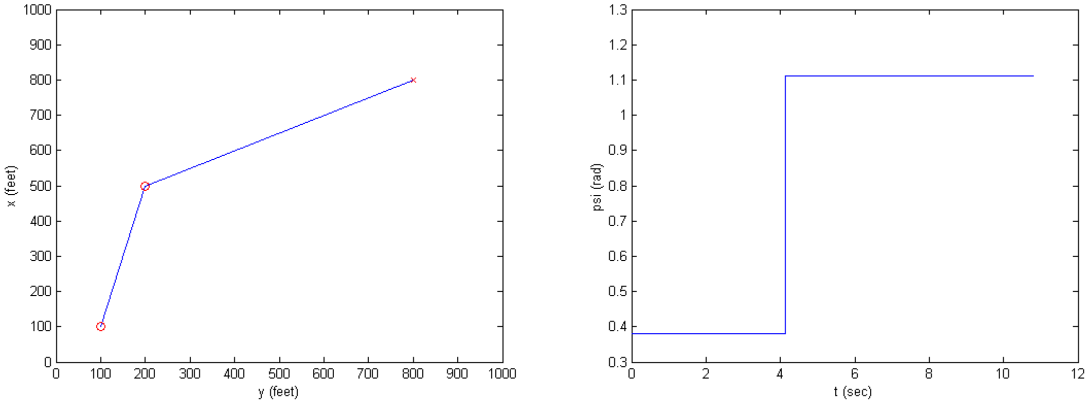


Figure 2.2: Solution with one waypoint.

example, flight over a flat plane with one waypoint is considered. Because all the terrain partial derivatives are zero, the local tangent plane equations of motion will reduce to the simplified equations of motion. In addition, both the costates will be at a constant value at all times with a jump at the time of reaching the waypoint. This results in the heading angle also being a constant value with a jump at the waypoint. Having a jump in the heading angle will create an optimal path that is not flyable. Therefore, the equations of motion for this section will be modified to ensure a smooth trajectory.

The equations of motion for this section will include the equations used earlier with the addition of ψ as an additional state. This results in equations of motion of

$$\begin{aligned} \dot{x} &= \frac{V\cos\psi}{A_1} + \frac{Vf_x f_y \sin\psi}{A_1 A_2} \\ \dot{y} &= \frac{-VA_1 \sin\psi}{A_2} \\ \dot{\psi} &= u \end{aligned} \tag{2.84}$$

for the local tangent plane or

$$\begin{aligned} \dot{x} &= V\cos\psi \\ \dot{y} &= V\sin\psi \\ \dot{\psi} &= u \end{aligned} \tag{2.85}$$

for the simplified equations of motion. In these equations, *u* designates the control variable for the system. The new cost equation is

$$J = \int_0^{t_f} \left\{ \left(1 - K + Kf\right) + Wu^2 \right\} dt = \int_0^{t_f} \left\{ A_4 + Wu^2 \right\} dt \tag{2.86}$$

which is the same as above with the inclusion of the control in the cost multiplied by a weighing factor.

The Hamiltonian for this problem is now

$$H = A_4 + Wu^2 + \lambda_x \dot{x} + \lambda_y \dot{y} + \lambda_\psi u \tag{2.87}$$

For both formulations, evaluating the optimality equation results in

$$H_u = 0 = 2Wu + \lambda_\psi \tag{2.88}$$

This yields the following equation for the control

$$u = \dot{\psi} = \frac{-\lambda_\psi}{2W} \tag{2.89}$$

At each interior point, the following conditions on the costates and the Hamiltonian must be met.

$$\begin{aligned} \lambda_x(t_i^+) &= \lambda_x(t_i^-) + \upsilon_{1i} \\ \lambda_y(t_i^+) &= \lambda_y(t_i^-) + \upsilon_{2i} \\ \lambda_\psi(t_i^+) &= \lambda_\psi(t_i^-) \\ H(t_i^+) &= H(t_i^-) = 0 \end{aligned} \tag{2.90}$$

Using these conditions from (2.90) as well as equations (2.87) and (2.86), an independent equation for λ_y is found as follows.

$$\lambda_y = \frac{Wu^2 - A_4 - \lambda_x \dot{x}}{\dot{y}} \tag{2.91}$$

Next the differential equations for the other two costates can be determined using

$$\begin{aligned} \dot{\lambda}_x &= -H_x \\ \dot{\lambda}_\psi &= -H_\psi \end{aligned} \tag{2.92}$$

This yields

$$\begin{aligned}\dot{\lambda}_x &= -Kf_x - \lambda_x\left[\frac{D_2\cos\psi + D_3\sin\psi}{D_1}\right] - \lambda_y\left[\frac{D_4\sin\psi}{D_1}\right]\\ \dot{\lambda}_\psi &= \lambda_x\left[\frac{V\sin\psi}{A_1} - \frac{Vf_x f_y\cos\psi}{A_1A_2}\right] + \lambda_y\left[\frac{VA_1\cos\psi}{A_2}\right]\end{aligned} \tag{2.93}$$

for the local tangent plane equations of motion with *D₁* through *D₇* defined in equations (2.25) – (2.31). For the simplified equations of motion, the costate differential equations are

$$\begin{aligned}\dot{\lambda}_x &= -Kf_x\\ \dot{\lambda}_\psi &= \lambda_x V\sin\psi - \lambda_y V\cos\psi\end{aligned} \tag{2.94}$$

This results in a system of six differential equations. The initial conditions for the two costates – λ_x and λ_ψ – must be found as well as υ_1 for each waypoint.

2.3.1 Legendre-Clebsch Necessary Condition

The Hamiltonian equation is stated in equation (2.87). The second partial derivative of it then

$$H_{uu} = 2W > 0 \tag{2.95}$$

which means that this condition is always satisfied.

2.3.2 Weierstrass Test

Using the Hamiltonian equations in (2.87) and the algebraic equations for λ_y and λ_ψ found in (2.91) and (2.89), the variational Hamiltonian can be found to be

$$H(u) = W\left(u - u^o\right)^2 \geq 0 \tag{2.96}$$

This shows that the Weierstrass test is always satisfied.

2.4 Expansion to 3-D formulations

The 3D equations of motion used are

$$\dot{x} = V\cos\gamma\cos\psi \tag{4.1}$$

$$\dot{y} = V\cos\gamma\sin\psi \tag{4.2}$$

$$\dot{z} = V\sin\gamma \tag{4.3}$$

Here, V is the vehicle velocity, γ represents the flight path angle and ψ is the heading angle. The cost equation used for this problem is

$$J = \int_0^{t_f} \{C_1 + C_2\} dt \tag{4.4}$$

$$C_1 = 1 - K + Kf(x,y) \tag{4.5}$$

$$C_2 = W\left[z - \left(f(x,y) + h_c\right)\right]^2 \tag{4.6}$$

This cost equation has two distinct parts. The first, and dominant part, is C_1 shown in (4.5). This part controls the importance of minimizing terrain masking versus minimizing flight time. The second part is C_2 as seen in (4.6). Here, h_c, the ground clearance, is a constant provided by the operator and represents the desired flight height above the terrain. This part is used to keep the flight path near the desired ground clearance throughout the flight. W is a weighing parameter supplied by the user.

In case the velocity of the vehicle is included as an additional state, the new equations of motion will now consist of the equations in (4.1-3) and also

$$\dot{V} = \frac{T - D}{m} - g \sin \gamma \tag{4.27}$$

In this equation, *m* is the mass of the vehicle, *g* is gravity, and *T* is the thrust of the vehicle, which, for optimal results should be held constant at its maximum value. The optimization procedure for this case is detailed in [19] and thus omitted here.

3. Six DOF Optimization by GESOP

3.1 Modeling

The 6 DOF full-order dynamics for GESOP optimization are constructed by incorporating engine dynamics and propeller rotation that are built into the *f-wing-simulator* (Simulator programmed and used by the Flight Mechanics and Controls Group at Georgia Tech.). This f-wing simulator is utilized to evaluate the feasibility of the pseudo 3-D optimal trajectory as well. The 6 DOF simulation dynamics comprise 14 states and described by

$$\dot{\vec{r}} = T_B^T \vec{V} \tag{1.7}$$

$$m\dot{\vec{V}} = -m\vec{\omega} \times \vec{V} + \vec{F} \tag{1.8}$$

$$\dot{\vec{\omega}} = -I^{-1}(\vec{\omega} \times I\vec{\omega}) + I^{-1}\vec{M} \tag{1.9}$$

$$\dot{\vec{q}} = -\frac{1}{2}\Omega_q \vec{q}\,, \tag{1.10}$$

where $\vec{r} \in R^3$ is the position vector, $\vec{V} \in R^3$ is the velocity vector, $\vec{\omega} \in R^3$ is the turn rate, and $\vec{q} \in R^4$ is the quaternion vector, $T_B \in R^{3\times3}$ and $\Omega_q \in R^{4\times4}$ are the directional cosine matrix and the quaternion matrix, m is the mass of the vehicle, and I is the moment of inertia. The force term $\vec{F}$ and the moment term $\vec{M}$ are determined by aerodynamic coefficients and engine characteristics. They are based on the Pioneer UAV

I_{yz}

with some changes in the physical dimensions. The calculation of the aerodynamic coefficients for *f-wing* and the 6DoF-model is based on the coefficient tables in [25]. For the GESOP-6DoF-Model these coefficient tables needed to be curve fitted using splines to provide smooth functions to the optimizer without a significant loss of performance. Table 3.1 gives an overview over the basic model characteristics used for the 6 DoF-Optimizations.

	SI-system	Non-SI-system
Wing area	1.8910 m^2	20.3546 ft^2
Wing width	4.3160 m	14.1600 ft
Chord width	0.4382 m	1.4375 ft
Air density	1.2250 kg/m^3	2.3770 slugs/ft^3
Mass	65.4 kg	4.4813 slugs
Earth acceleration g	9.8067 m/s^2	32.174 ft/s^2
Moments of inertia		
I_{xx}	7.0838 kg m^2	5.2248 slugs ft^2
I_{yy}	13.6422 kg m^2	10.062 slugs ft^2
I_{zz}	16.7213 kg m^2	12.333 slugs ft^2
I_{xy}	0.0 kg m^2	0.0 slugs ft^2
I_{xz}	-0.9965 kg m^2	-0.735 slugs ft^2
I_{yz}	0.0 kg m^2	0.0 slugs ft^2

Table 3.1: Model characteristics

3.2 Issues for comparison with 3-D results

Recall that the pseudo 3-D formulations in Sections 2.1 and 2.2 incorporate the vertical motion into the equations of motion so that the resulting trajectory maintains the same altitude over the terrain. The 3-D formulation in Section 2.4 or 6 DOF formulations in Section 3.1, raise the question of how to distinguish hills to prevent the optimizer from leading to a solution that flies through hills. One possibility is put a high penalty on changing the altitude over the ground so that the optimal path lays somewhere close to the constant altitude over the ground layer, which is precisely what has been done, as in the term C_4 in (4.6), in the 3-D formulation in Section 2.4. In case of the 6 DOF full aircraft dynamics, since the aircraft dynamics involve more states and control variables neglected in the reduced-order formulations, ensuring non-contact between the airplane and the terrain should be accomplished by properly maneuvering the airplane, which in general requires controlling the aircraft by control effectors and throttle. This point indicates that direct comparison between optimal trajectories from the reduced-order and the full 6 DOF aircraft dynamics is not straightforward. For example, even if the same cost functional is used, whereas the pseudo 3-D formulation leads to a trajectory of constant velocity, the 6 DOF optimizer cannot be forced to fly at a constant velocity and the cost integral over time will have little meaning for direct comparison.

This indicates that for a full 6 DOF optimal trajectory to be comparable to the reduced-order solution, we need to address the following issues:

- How to ensure for the flight path in the 6 DOF optimizer to be close to the optimal trajectory of the reduced-order formulation so that the cost functional in (2.5) makes sense in both cases

- How to set boundary conditions for extra states in addition to those states resulting from the reduced-order dynamics.
- How to transform constraints on the control effectors and the throttle into equivalent constraints on those states that are treated as controls in the reduce-order formulation.

Case studies for these complexities and some related simulations are presented in [18]. For example, an experiment with the terrain altitude as a path constraint in GESOP full order optimization required a large number of grid points for real terrain to cover all terrain peaks. Also, direct optimization of 6 DOF model produced heavy peaks and jumps in control signals, which are completely neglected in the reduced-order formulation. Therefore, to circumvent those complexities arising due to aircraft dynamics such as control peaks and engine characteristics, additional cost terms are included in the 6 DOF optimization procedure[18]. The goal for these additional costs is to realize the constant altitude flight path in the full aircraft dynamics as close as possible to those in the reduced-order formulation. The additional terms in GESOP optimization includes avoiding peaks in controls, altitude control by terrain-following, and inhibiting trajectory-terrain intersection. The additional cost induced by these terms for terrain-masking is calculated and compared to the original cost in a terrain-following scenario as well in [18].

4. Numerical Procedure for Optimal Trajectories

4.1 Optimization Procedure for the reduced-order formulation

In case of reduced-order formulation, a solution that satisfies a set of necessary conditions for optimality is found by seeking proper initial conditions. Two different methods were utilized to find these initial conditions, depending on the number necessary. When only one value was needed, a variable step sweep was employed to find it. Otherwise the GA was used[26-28].

To begin the genetic algorithm, a set of 48 chromosomes was initialized representing different sets of initial conditions to test. Each initial value in the chromosomes was represented by digits with five decimal places included. In addition the costates, flight path angles and Hamiltonian values included an extra digit to indicate a positive or negative value. After the chromosomes were initialized, they were each tested to determine their relative costs. To accomplish this, the current chromosome being tested was broken into its respective initial conditions, which were then used in the differential equations. The cost, *J*, was found for the run as well as the distance from the final position of the run to the final target position. The sum of these two values was used as the total cost for the chromosome.

After each chromosome was tested and a total cost assigned, the chromosomes were ranked from last to first based on a tournament procedure. Two chromosomes would be randomly chosen to compete and the one with the higher cost was placed in the next position on the list while the one with the lower cost was returned to the available set of chromosomes to be tested. This process was continued until all the chromosomes were ranked. The top 24 chromosomes were then kept to begin the next generation.

In all subsequent generations, the 24 available chromosomes were combined to create 24 new chromosomes to complete the population of 48. Here, chromosomes 1 and 2 would be combined to create two new chromosomes, and then chromosomes 3 and 4 would be combined to create two new chromosomes and so on until all the chromosomes were mixed. This was accomplished by first mixing the individual segments of the chromosomes so that each of the new chromosomes had some segments from each parent, where a segment consisted of the digits for each initial condition needed. Next a mutation was introduced into the new chromosomes such that up to about a third of the digits could be changed. The number of digits changed, which digits were changed, and their new values were all determined randomly. After all the new chromosomes were created, the cost assignment and tournament were repeated as before. This process was repeated until it converged on a solution.

The differential equations were solved using a standard fourth order Runge-Kutta method. In addition, a variable time step was implemented to decrease the time needed to numerically solve the set of differential equations. For most of the flight, the time step was 0.1 seconds; however, when the distance to the target final position was close enough, the time step was decreased to 0.01 seconds.

Another condition was added to the differential equation solver to help decrease the solving time of the genetic algorithms. For each formulation, inequalities were derived that had to be satisfied at all times in order for the Legendre-Clebsch necessary condition and the Weierstrass test to be satisfied. These inequalities were then tested at each time step. If either was violated, then the current run was ended at that point. This decreased the solving time significantly, but was even more useful in ensuring the convergence to a strong local minimum.

Table 4.1 contains some average run times for the different formulations using 1.8 GHz Pentium IV with 1 GB RAM. This table contains the number of initial conditions to be solved for, the run time length for each problem, then the average time it took to solve the problem. It can be seen that the single vehicle formulations were all generally solved in less than a minute for a case with a run time of 10 seconds. The multiple vehicle formulations took longer to solve because those cases tended to have a large number of local minima.

Formulation	Number of Variables	Run time	Time to Solve
Pseudo 3D	1	10 sec	25 sec
3D	2	10 sec	40 sec
Varying Velocity	3	10 sec	65 sec
2-vehicle	3	25 sec	6 min
3-vehicle	5	25 sec	15 min

4-vehicle	7	25 sec	60 min
2-vehicle 3D	5	10 sec	10 min

Table 4.1: Time to Solve for Reduced-order Formulations

4.2 Optimization Procedure for GESOP

GESOP provides several different optimizers. The two main types are methods either using a Direct Collocation Method or a Direct Multiple Shooting Method. All optimizations in this work were done with Multiple Shooting Method called SNOPT (Sparse Nonlinear OPTimizer)[29] .

The maximum number of iterations was changed from the default 80 to much higher values for the more complex iterations. Values around 1000 were used to make sure that the optimizer was able to find the optimal solution without any restarts. Initially restarts were used to obtain a stable solution, but later it was found that the majority of cases did not require a restart.

The values for “Real Workspace Size” and “Integer Workspace Size” had to be increased up to 1,000,000 and 300,000 for the most complex optimizations, to inhibit the raise of SNOPT error code 20 which indicates a too small workspace. The variables “Opt. Tolerance” and “Const. Tolerance” were left at1.0×10^{-6} . This causes relatively high iteration times but on the other side a very accurate solution. The change of the solution on the last 30%-50% of all iteration steps often consists of a very low and almost negligible change over a good part of the last iteration steps. For the Major Grid the number of points was increased to 50 for most cases. Especially the optimizations for flights over “Real Terrain” needed such high values to converge. The Constraints Grid was set to a single point since for the final cases no constraints were applied. The Controls Reference Grid was set to a relatively low value of 9 points in most cases. For the simulation of the complex optimizations the “Number of Points” for the output spacing had to be increased to 201.

The average run time needed to converge is mostly dependent on the accuracy required. With a very high accuracy setting the optimal solution is found in most cases after about 45-90 minutes CPU-time (1.8Ghz Pentium IV). With a lower accuracy setting the optimal solution can be found in about 30-45 minutes with just very little differences compared to the high accuracy setting. A good result which approximately follows the optimal solution is available already after a couple of minutes. The results of each step can be observed already during the optimization process within GESOP. In general optimizations going over real terrain need some more time compared to mathematical terrains due to the interpolation routine.

5. Numerical Results

5.1 Terrain Data

Two general types of terrain models are used for the results throughout this report. The first is a generic terrain model used for the initial testing of the equations. This consists of variations of a flat plane with one or more constructed hills. The second consists of actual terrain data for a larger area. This allows the opportunity for the various equations to be tested in a more realistic manner.

A sample terrain of the generic model is shown in Figure 5.1. In this case a mostly flat plane with a single hill is used. This hill in this terrain is formulated using the exponential function

$$f = Ae^{-r^2/b} \tag{5.1}$$

where A is the amplitude, b is a scaling factor to adjust the width and r is the distance from any position to the center of the threat.

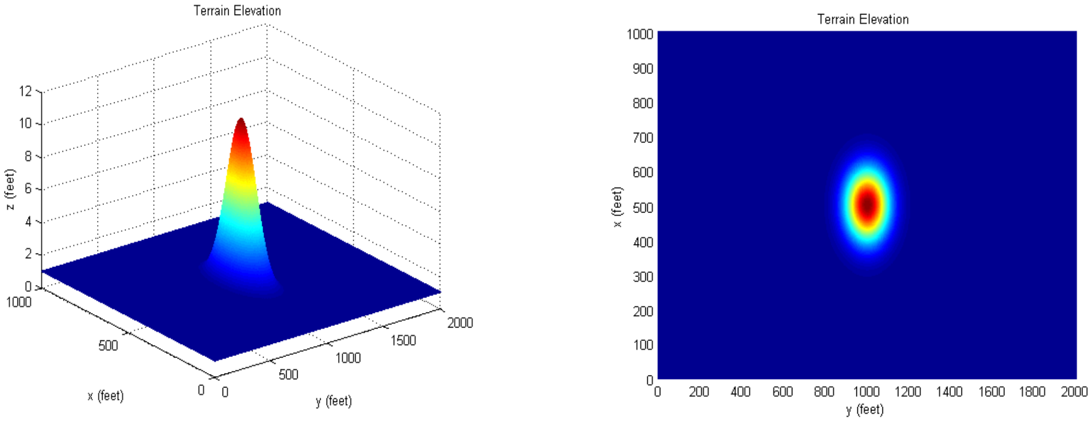


Figure 5.1: Terrain with threats formulated as an exponential function.

Real terrain data was acquired from the United States Geological Survey to incorporate into this model[30]. The data was found in tabular format relating the altitude to the locations longitude and latitude, with data points spaced approximately every 48 feet. This data was then converted to matrix form, from which it could then be used as $f(x, y)$. Because of the distance between the sampled altitude points in the matrix, the data was then smoothed to appear more continuous and to remove discontinuities in altitude. The gradients of this matrix, along both the x and y directions, were calculated numerically to form matrices representing $f_x(x, y)$ and $f_y(x, y)$. The gradients of these two matrices yielded matrices for $f_{xx}(x, y)$, $f_{yy}(x, y)$ and $f_{xy}(x, y)$.

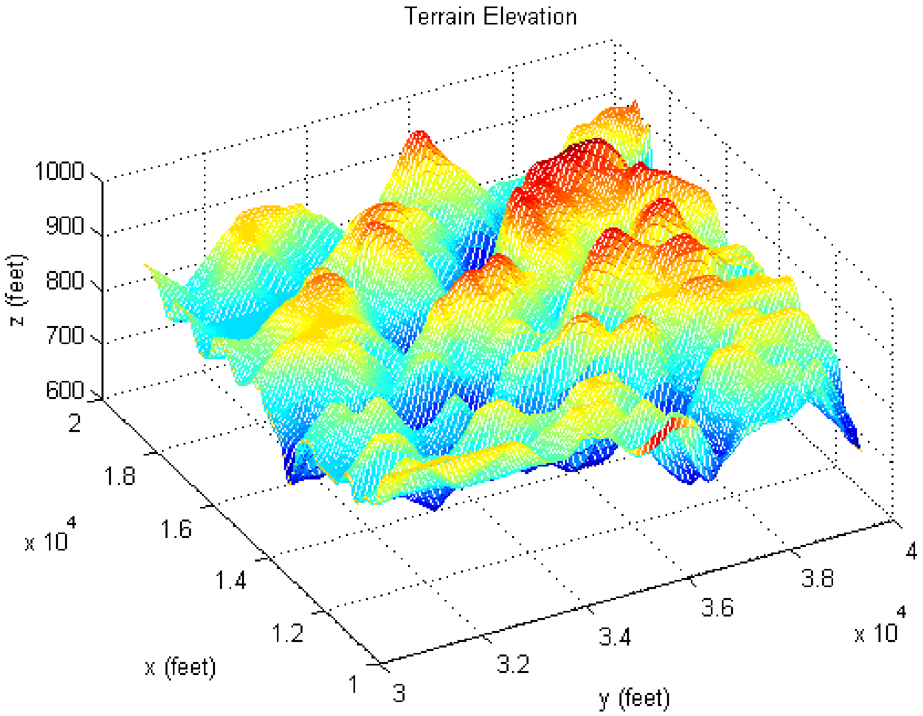


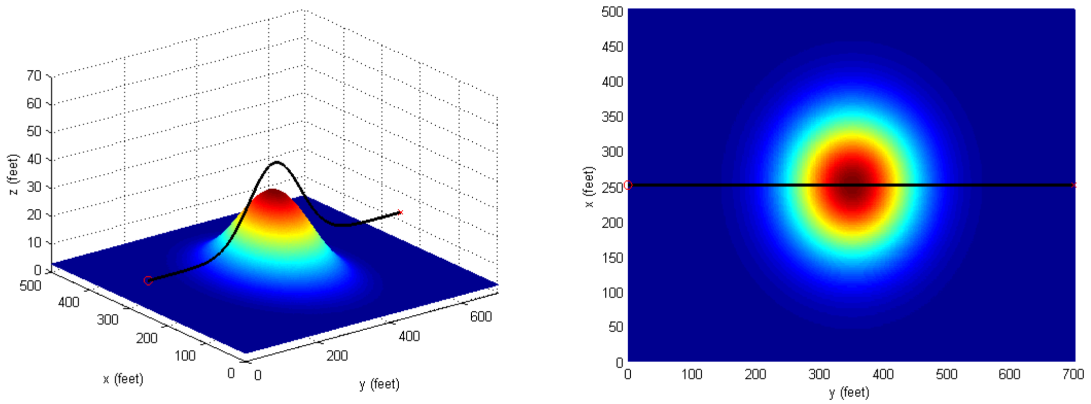
Figure 5.2: Terrain plot of an area near Columbus Ohio.

For this portion of the testing, it was decided to use a section of terrain near Columbus, Ohio. A profile of this terrain can be seen in Figure 5.2. In this graph, the x and y-axes depict the position coordinates, measured in feet, such that the x-axis point north and the y-axis points east. The altitude of the terrain is measured along the z-axis and is also given in feet. This plot depicts a square plot of land, with 10,000 feet to a side. The measurements along the x and y-axes are relative to a set origin.

5.2 Pseudo 3-D and 3-D results

Optimal trajectories from the pseudo 3-D formulations that include wind effects, moving targets, interior point constraints in the form of waypoints, and moving threats, and those from the 3-D formulation are found in the chapters 3 and 5 of the thesis [19]. The simulations were performed with generic terrain models in Figure 5.1 and the real terrain data in Figure 5.2. In this report, we compare the trajectories obtained from both pseudo 3-D and 3-D reduced-order formulations. For a given terrain, a flat plane with a single hill, the optimal trajectories found to navigate it for both minimum time and terrain masking flight are compared. The formulations considered include the simplified and local tangent plane equations of motion for the pseudo-3D case as well as the constant velocity and varying velocity 3D equations of motion. This is repeated for three different-steepness hills.

Figures 5.3 – 5.5 contain the results for the minimum time, $K = 0$, case. Each figure portrays the results from a different hill height. The top two plots are a 3D view and an overhead view of the trajectories from the pseudo-3D case while the bottom two plots depict the paths for the 3D cases. For these cases, the trajectories from using the simplified equations of motion and from using the local tangent plane equations of motion are the same and are represented by the black line. In the bottom two plots, the black line represents the simplified the constant velocity 3D equations of motion.



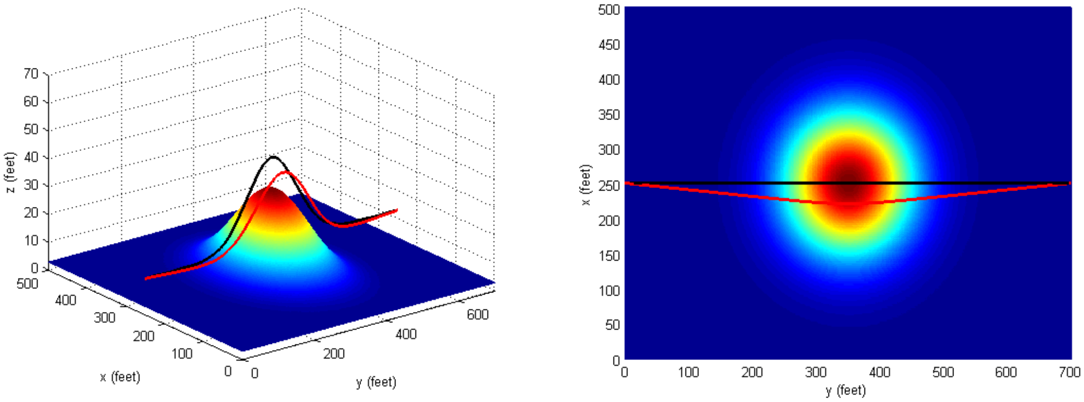
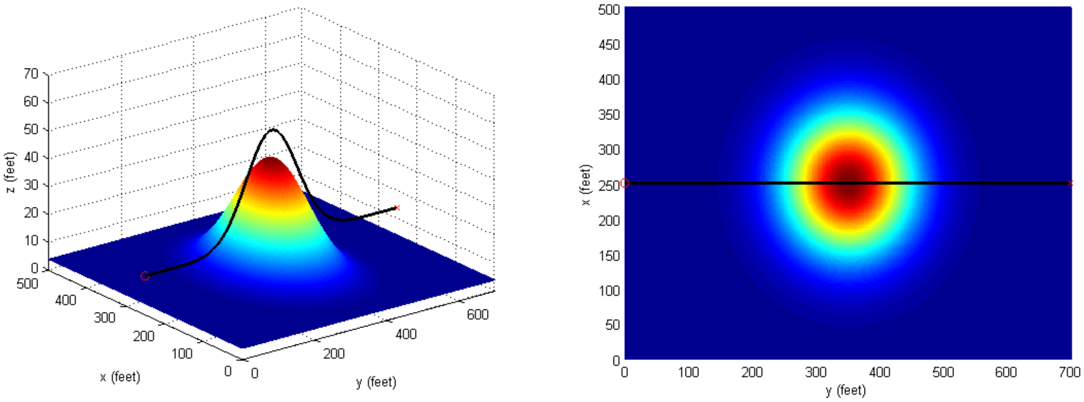


Figure 5.3: Trajectories for *K* = 0 and hill height = 30.

The red line is for the varying velocity 3D equations of motion. The plots for the 3D equations of motion trajectories are the same as depicted in Chapter 5 in [19], and are repeated here for convenience. In the first two cases, with hill heights of 30 and 40 feet, the constant velocity 3D trajectories are the same as the pseudo-3D formulations; however, in the steepest hill, the 3D trajectory begins to veer around the hill. In all three cases, the varying velocity trajectory veers around the hill to some extent.



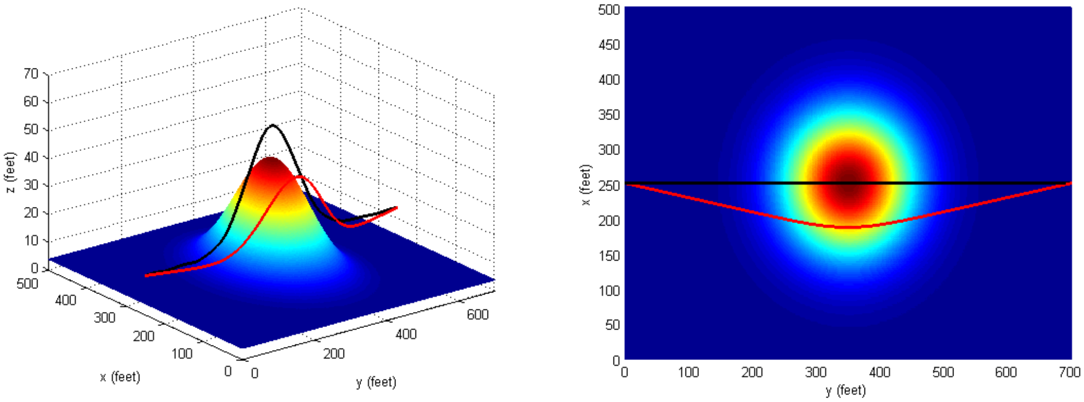


Figure 5.4: Trajectories for *K* = 0 and hill height = 40.

Some details from the various cases portrayed in these figures are tabulated below in Table 5.1. Here, the time for the flight and the cost of the flight can be compared for each set of equations of motion investigated. In this table, it can be seen that the flight time does not change with the simplified equations of motion, while the time increases for the local tangent plane and 3D equations of motion. This is because the time needed to fly vertically is ignored in the simplified equations of motion. This time is better accounted for with the local tangent plane equations of motion, but these final times are still slightly less than the final times with the 3D equations of motion, especially when the hill steepness is greater. The final times for the varying velocity cases are significantly greater due to the loss in velocity during the flights.

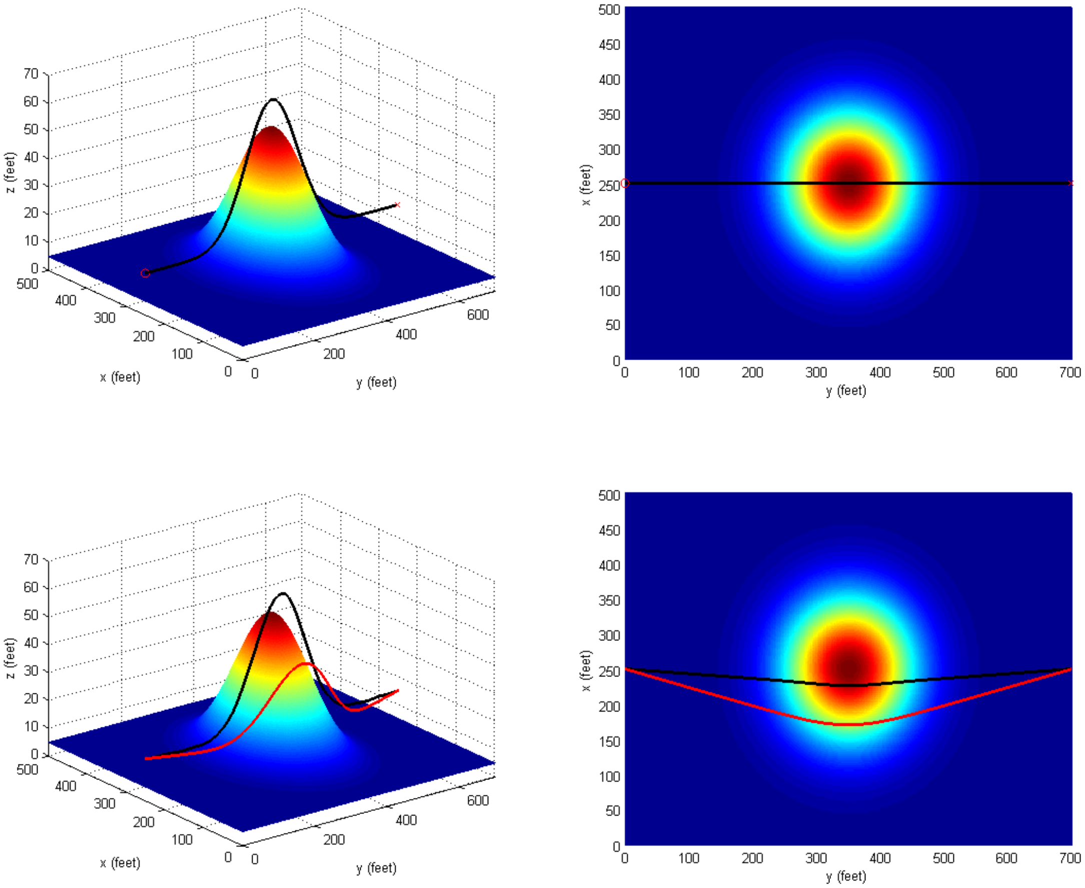


Figure 5.5: Trajectories for *K* = 0 and hill height = 50.

	Hill Height = 30		Hill Height = 40		Hill Height = 50	
	tf	cost	tf	cost	tf	cost
simplified	7	7	7	7	7	7
local tangent plane	7.06	7.06	7.1	7.1	7.15	7.15
3D	7.06	7.063	7.11	7.116	7.16	7.165
varying velocity	7.27	7.289	7.37	7.371	7.45	7.451

Table 5.1 Trajectory data for *K* = 0 flights

Figures 5.6 – 5.8 contain the plots for the results for K = 1. These plots are for cases parallel to those shown in Figures 5.3 – 5.5. The same three hill heights are displayed with the four sets of trajectories depicted. As in the earlier plots, the top plots

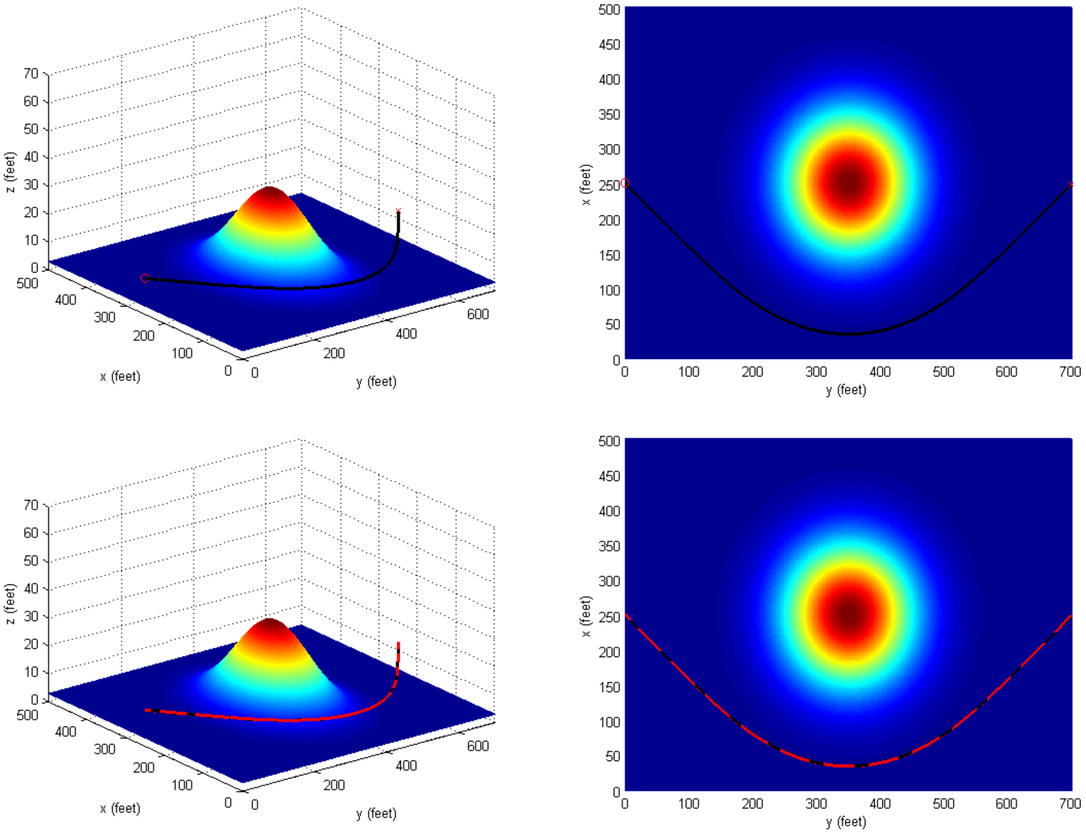


Figure 5.6: Trajectories for *K* = 1 and hill height = 30.

show the trajectories from using the simplified and local tangent plane equations of motion. The bottom two plots depict the results from Chapter 5 in [19] for the trajectories using the 3D constant velocity and varying velocity equations of motion. In each of the four cases, for each of the three hill steepness, the trajectories appear the same. The trajectory always curves around the given hill.

Table 5.2 contains the final time and cost information for the trajectories from each set of equations of motion and for each hill height for the *K* = 1 formulation. It can be seen here that there is very little difference in these results regardless of the equations of motion used or the hill steepness.

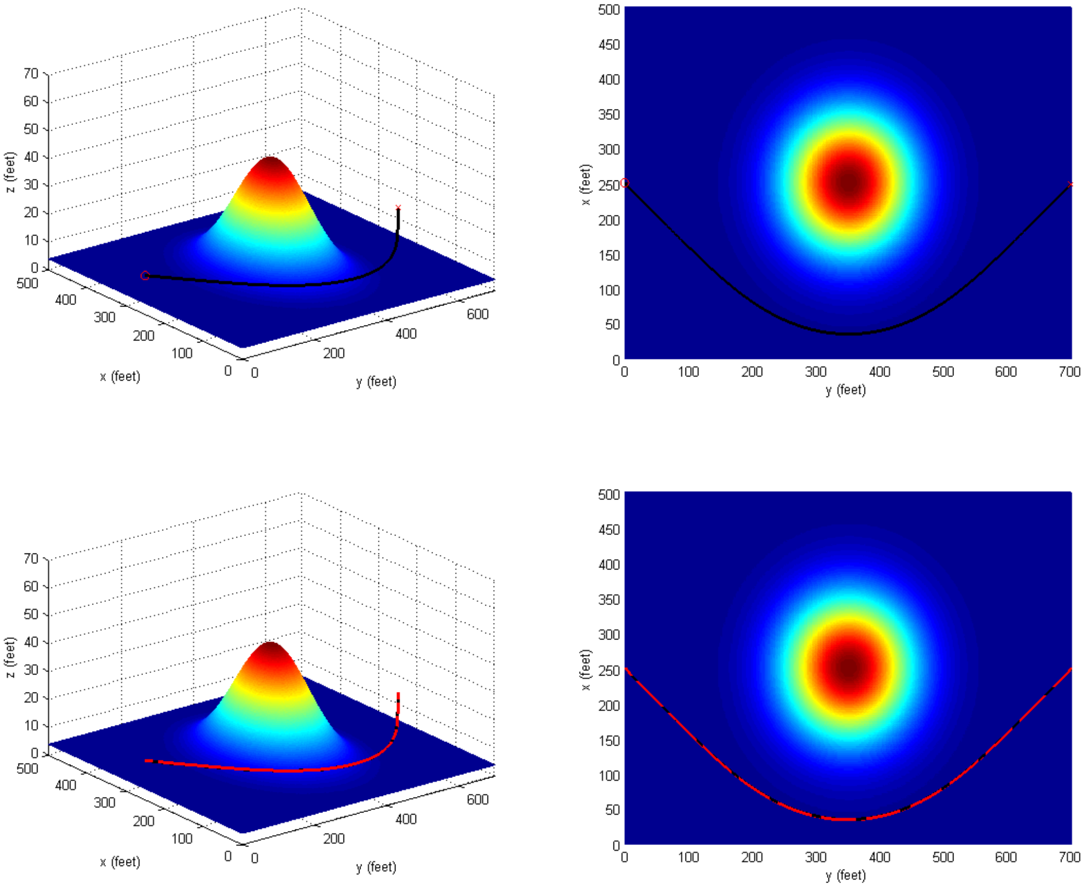


Figure 5.7: Trajectories for *K* = 1 and hill height = 40.

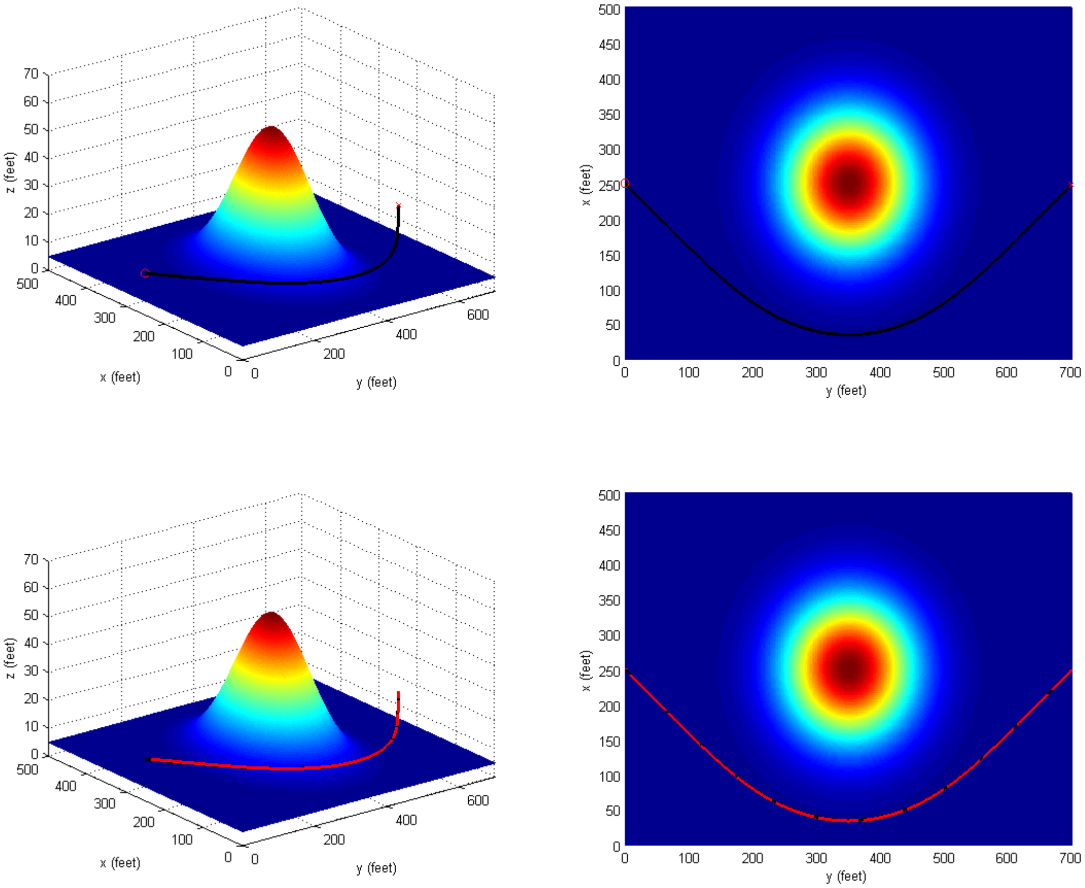


Figure 5.8: Trajectories for *K* = 1 and hill height = 50.

	Hill Height = 30		Hill Height = 40		Hill Height = 50	
	tf	cost	tf	cost	tf	cost
simplified	8.42	26.349	8.42	35.1314	8.42	43.9143
local tangent plane	8.42	26.348	8.42	35.1305	8.42	43.9132
3D	8.42	26.349	8.42	35.132	8.42	43.914
varying velocity	8.41	26.306	8.42	35.108	8.42	43.88

Table 5.2 Trajectory data for *K* = 1 flights

5.3 Six DOF optimal trajectories using GESOP

While reduced-order optimal trajectories are obtained by indirect methods, GESOP utilizes a direct method. Therefore, it is first verified that optimization results of GESOP are comparable with the results the optimizations in [19] by implementing the simplified equations in Section 2.2 into GESOP with the same boundary conditions enforced. It is shown that the resulting trajectories are almost identical (see Chapter 5 in [18]). This ensures that numerical methods employed for optimization do not generate differences. The first optimizations were the flights around one, three and five mountains. The 3DoF-Optimizations were used for the optimizer verification presented in Appendix B in [18]. Later when the 6DoF model was working, the same optimizations were done with it to investigate its behavior for flights over mathematical formed terrain as in (5.1).

For the 6DoF-Optimization *K* was set to 1 and Terrain-Following was activated. Since the ground under the optimal path is almost perfectly flat a constant absolute altitude must be the result. As shown in Figure 5.12 this is almost perfectly the case except a very small variation of about 7ft on a flight path length of more than 1600ft.

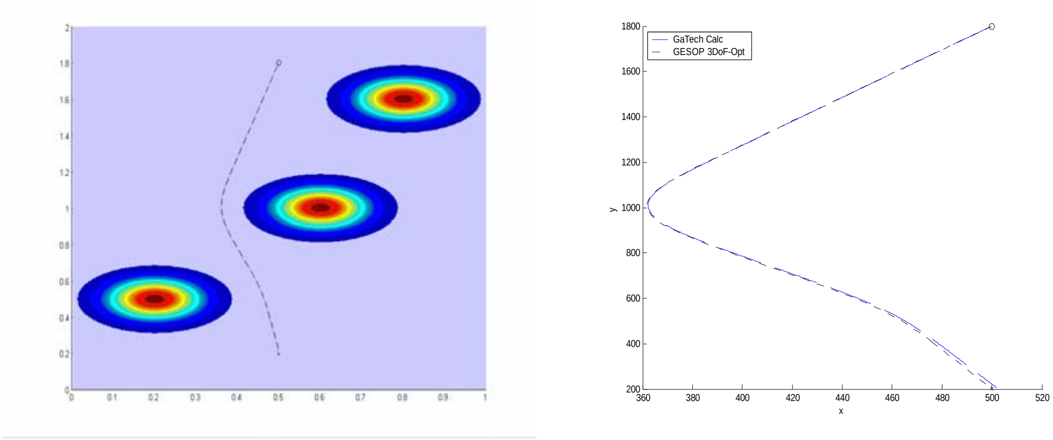


Figure 5.9. Trajectories for 3 Mountains – 3 DOF GaTech GESOP

In figure 5.9 practically identical results of the GaTech-Calculation (dashed) and the GESOP-3DoF-Optimization (dotted) are shown with the corresponding mountains. The *x*-*y*-units are in [feet] here. The zoomed trajectories are shown in the right figure in Figure 5.9.

As a next step, case studies with different operating conditions and cost functions are extensively carried out to understand the influence of different initial conditions, dynamic constraints, and different cost functions. Table 5.3 illustrate cases studied with the full 6 DOF GESOP optimizations, whose definitions for case details can be found in [18].

Case	Terrain Masking optimization (K=1)	Explicit Time optimization (K=0)	Terrain following on	Visibility Check	High Penalty for flying under Terrainlevel	Final altitude constraint	Comparable weighting	Flight altitude	Maximal Throttle
1. Reference I	X		X					High (50m)	1.0
2. Reference II	X		X		X	X	X	High (50m)	1.0
3. Reference I, Lower flight altitude	X		X					Low (18m)	1.0
4. Reference I, Lower flight altitude + penalty for flying under terrain level	X		X		X			Low (18m)	1.0
5. Reference I, Half Thrust	X		X					High (50m)	0.5
6. Reference I, deactivated engine moments	X		X					High (50m)	1.0
7.Reference II, Optimal Time		X	X		X	X	X	High (50m)	1.0
8.Reference II, Visibility Check	X			X	X	X	X	High (50m)	1.0
9.Reference II, Visibility Check + Terrain following	X		X	X	X	X	X	High (50m)	1.0
10. Reference I + Final Constraint	X		X			X		High (50m)	1.0
11. Reference I, Optimal Time		X	X					High (50m)	1.0
12. Reference II, Lower flight altitude + penalty for flying under terrain level	X		X		X	X	X	Low (18m)	1.0
13. Flight over terrain for pop up threat (threat not active)	X		X		X	X	X	Low (20m)	1.0
14. Flight over terrain for pop up threat (threat active)	X		X		X	X	X	Low (20m)	1.0

Table 5.3 Parametric Studies-Overview

5.4 Comparisons between Pseudo 3-D results and GESOP solutions with Simulated hills

5.4.1 Flight around three mountains

Figure 5.10 shows the dotted GESOP-pseudo 3-D Optimization, labeled as "GaTech 3DOF-Opt", now in comparison with the trajectory followed by the aircraft in the f-wing simulator, labeled as "GaTech Sim", and the solid 6DoF-GESOP-Optimization, labeled as "GESOP 6DOF-Opt".

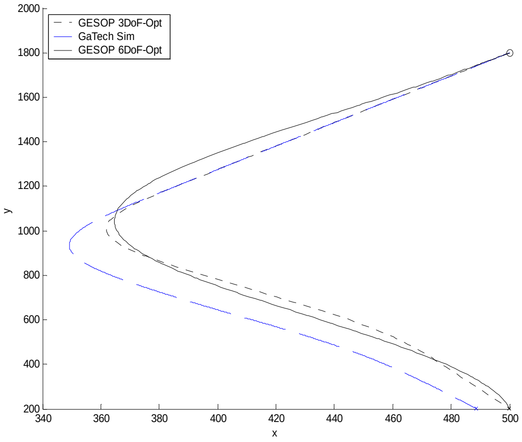


Figure 5.10 3 Mountains-3 DoF-GESOP vs GaTech-Sim & GESOP 6 DoF

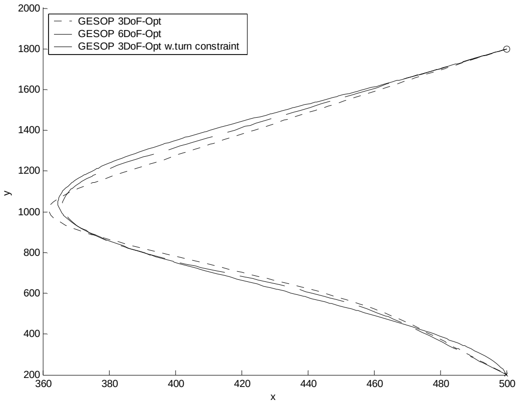


Figure 5.10 3 Mountains-3 DoF- with and without Turn rate constraint and GESOP 6 DoF

The GaTech-Simulation follows the commanded trajectory very closely up to the sharp turn. There it follows the command with a short delay and continues on a path parallel to the commanded trajectory with a constant shift until the end of the simulation. The solid 6DoF-Optimization starts and ends at the same point like the 3DoF-Optimization.

This is not surprising because these points are set as initial and final boundary conditions. The trajectory's course is also about the same. But the dynamics of the airplane model seem to cause a smoothening of the trajectory. The turn is less sharp, it begins earlier, ends later and the maximal extend to the left is smaller. It looks like the maximal turn rate is restricted by the airplane's dynamics. So an interesting question is how the 3DoF-Optimization would change if a maximal turn rate constraint would be set. The result is shown in figure 5.11. The dotted line is the 3DoF-Optimization as before, the solid line the 6DoF-Optimization as before, and the dashed line the new 3DoF-Optimization with the turn rate constraint. The 3DoF- and 6DoF-solutions are almost identical now. So for flights over flat terrain at a constant altitude a turn rate restriction in the simple 3DoF-Simulation can produce a solution very similar to the 6DoF-solution. Most probably the autopilot of the simulator would have minor problems to follow the commanded trajectory, too, because no impossible turn forcing the airplane to leave the commanded trajectory would be included any more.

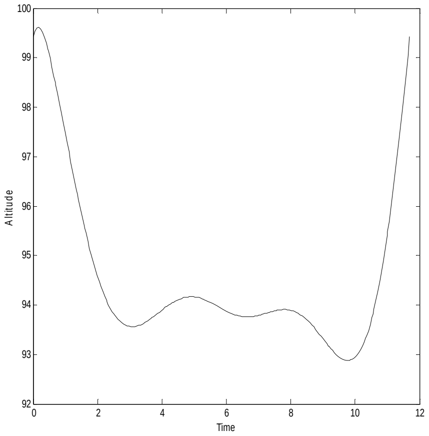


Figure 5.12: 3 Mountains – 6DoF – time-altitude

5.4.2 Flight around Five mountains

As can be seen in the figures 5.13 and 5.14 the 3DoF-Optimizations from GaTech and GESOP are identical again, in figure 5.15 gets clear again that the GaTech-Simulation follows with some delay and a persisting deviation while the 6DoF-

Optimization follows closely with a slightly lower maximal turn rate. In figure 5.16 the maximal turn rate of the dashed 3DoF-Gesop-Optimization was set to a value so that the 3DoF-Result is similar to the 6DoF-Trajectory. But at the first little turn it is obvious that for a good turn rate constraint approximation it must be not just constant, but velocity dependent. The limited solution has a bigger distance to the unlimited solution and the 6DoF-Optimization is able to follow that very closely. As shown in figure 5.17 the 6DoF-velocity is still low there. In the second turn the 6DoF-Simulation takes a wider way

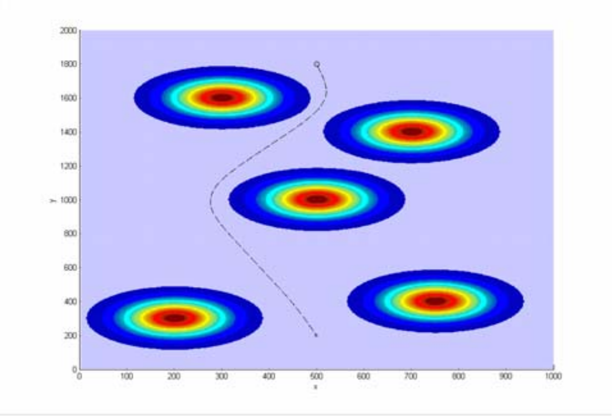


Figure 5.13: 5 Mountains – 3DoF GaTech-Gesop – xy a

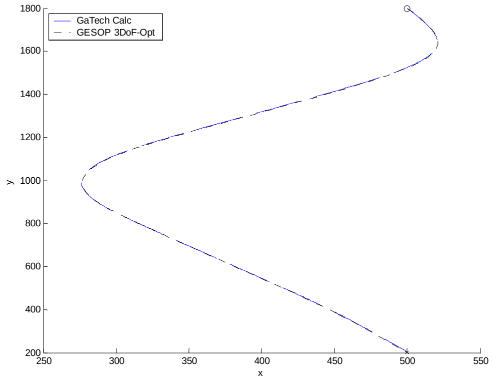


Figure 5.14 1: 5 Mountains – 3DoF GaTech-Gesop – xy b

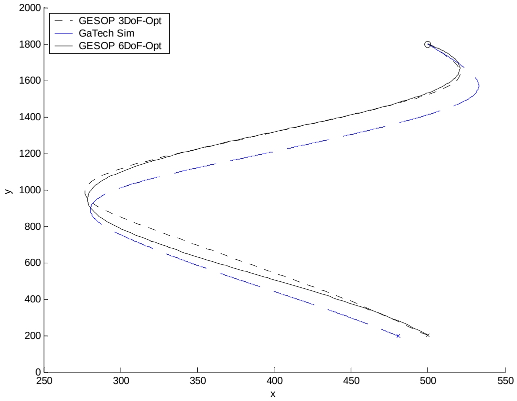


Figure 5.15: 5 Mountains – 3DoF Gesop vs. GaTech Sim & Gesop 6DoF - xy

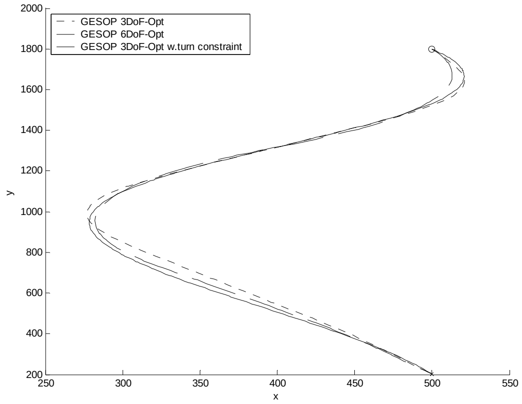


Figure 5.16: 5 Mountains – 3DoF with/-out turn constraint vs. 6DoF Gesop - xy

around the mountain and sticks closer to the limited solution due to its higher velocity. So a velocity dependent turn rate restriction in the 3DoF-Optimizations for flights in constant absolute altitude would approach the 6DoF-Optimization more closely than the constant restriction.

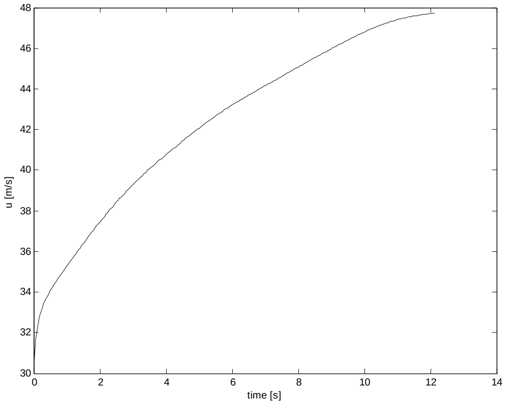


Figure 5.17: 5 Mountains – 6DoF – velocity u over time

5.4.3 Flights With Moving Targets and Threats

In addition to the comparisons of the 3DoF-Target-Threat-Optimizations, now the 6DoF-Optimizations and the 3DoF-Simulations will be compared to the 3DoF-Calculations. As 3DoF-Calculation always the GaTech solution is shown.

Figure 5.18 shows the 6DoF-Optimization in comparison with the GaTech-Calculation. There is no big surprise, the 6DoF-Optimization takes the turn a little smoother, but this is the only difference. In the GaTech-Simulation case in figure 5.19 the path is followed almost perfectly on the straight line up to the turn. There a shift is generated and doesn't get completely eliminated till the end point. It is anticipated that an I-factor in the autopilot is possibly too small there.

A completely different course than the GaTech calculation takes the 6DoF-Simulation in the 1 threat case (figure 5.20). Instead of passing in front, it passes behind. But examinations showed that this pass-behind solution can also be produced with the Gesop-3DoF-Optimization. Simply the heading for the initial guess needed to be changed and a solution very similar to the 6DoF-Optimization (dotted line) was the result. But it wasn't possible to do it the other way round so that the

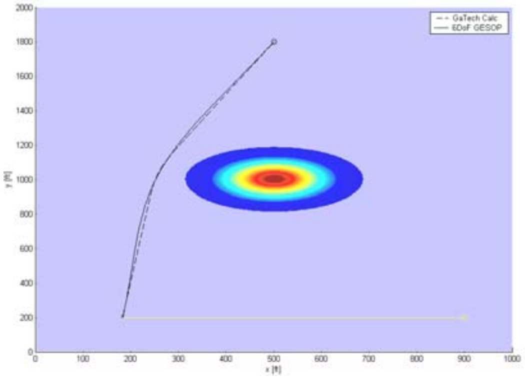


Figure 5. 18: Moving-Target: GaTech-Calc vs. 6DoF-Gesop - No Threat

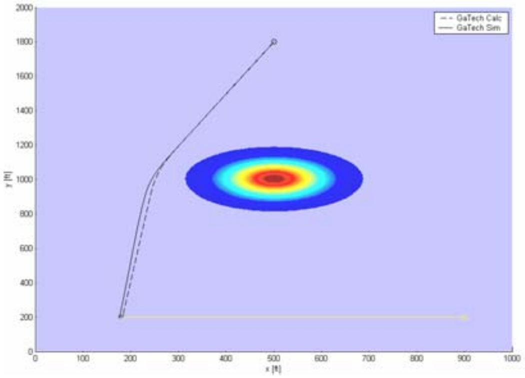


Figure 5.19.: Moving-Target: GaTech-Calc vs. GaTech-Sim. - No Threat

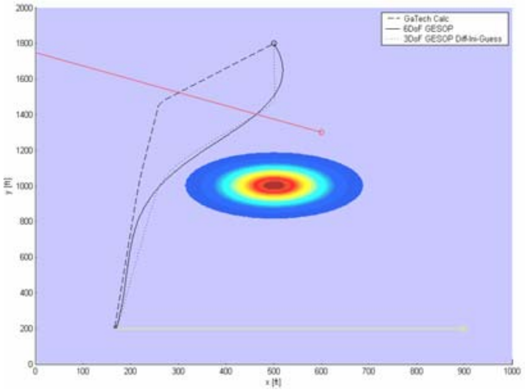


Figure 5.20: Moving-Target: GaTech-Calc vs. 6DoF-Gesop, 3DoF-Gesop restricted - One Threat

6DoF-case would follow the original 3DoF-case.

Of course the GaTech-Simulation's autopilot in figure 5.21 follows the commanded trajectory but here the gap between the commanded and the simulated path is wider.

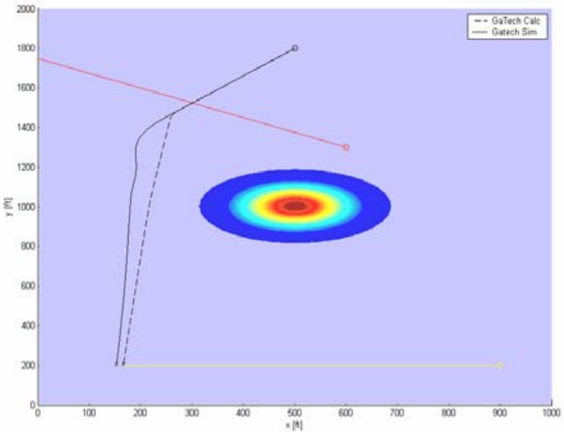


Figure 5.21: Moving-Target: GaTech-Calc vs. GaTech-Sim. - One Threat

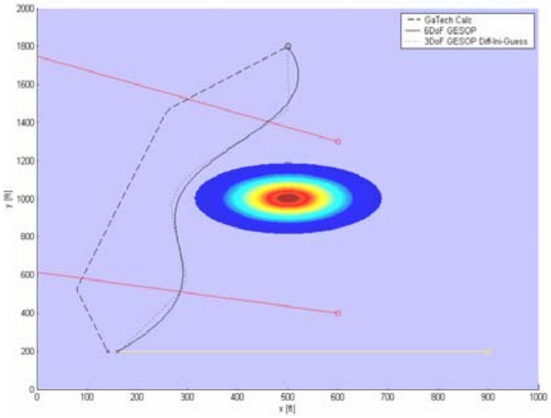


Figure 5.22: Moving-Target: GaTech-Calc vs. 6DoF-Gesop, 3DoF-Gesop restricted - Two Threats

It closes faster than before, but not completely till the end as well. The second threat doesn't change the situation in general any more. Also here the 6DoF-Optimization passes behind both mountains, the original 3DoF-Optimizations pass in front, and again a 3DoF-Optimization with a different heading for the initial guess produces a solution similar to the 6DoF-version. The autopilot in figure 5.23 reacts as described before, but it should be noticed that the gap after the second turn is bigger than after the first one. This could be because the second

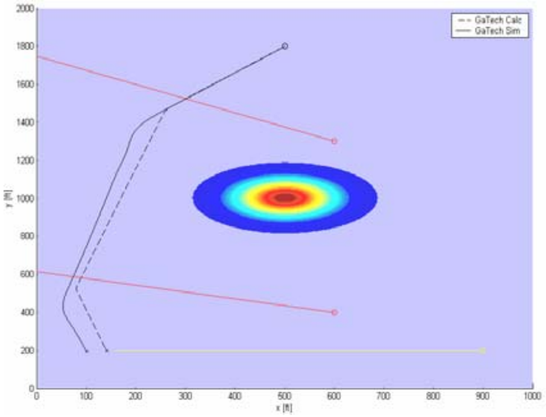


Figure 5.23: Moving-Target: GaTech-Calc vs. GaTech-Sim. - Two Threats

turn is simply harder, but at least a portion is owed to the fact that the gap isn't closed at the beginning of the second turn. Therefore more turns could lead to problems with the stability of the simulation when it gets too far off the commanded track.

5.5 Comparisons between Pseudo 3-D results and Gesop solutions with the real terrain

5.5.1 Terrain-Masking (K=1)

The first 3DoF-6DoF comparisons over real terrain are optimizations with K=1 (Terrain-Masking). The Gesop-Optimization is the lower case of the comparison with K=1 Low-Flight – High-Flight shown above. In figure 5.24 the GaTech (black) -Calculation (dotted), its simulation (solid), together with the Gesop-Optimization (white) are drawn. It can be seen immediately that the GaTech solution takes a completely different way compared to the Gesop trajectory. Despite the fact that the optimizer with K=1 is set to fly over terrain which is as low as possible, the GaTech-Calculation takes a path over the yellow ridge. In addition the simulation has significant problems to follow the commanded path (figure 5.25). This even leads to a path where the airplane would crash into the terrain at the first hill at this altitude (break

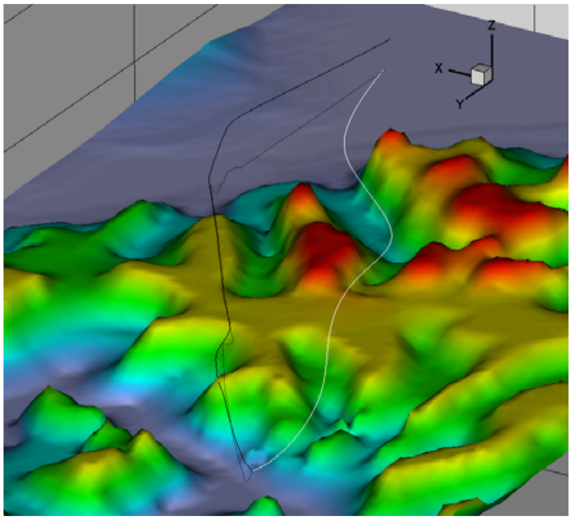


Figure 5.24: Ref.II, K=1 vs. GaTech-Calc & -Sim.-3D

in the solid black line in figure 5.25). Also the commanded final altitude cannot be reached. The reason for that is most probably the fact that the simulator tries to fly a constant velocity. A steeper decent would accelerate the airplane over the commanded constant velocity, so the Autopilot descends slower. Regarding the flight path length the Gesop-Optimization's path is a little shorter (figure 5.26). Also the integral of the terrain altitude over the path length shows the significant quality difference between the paths.

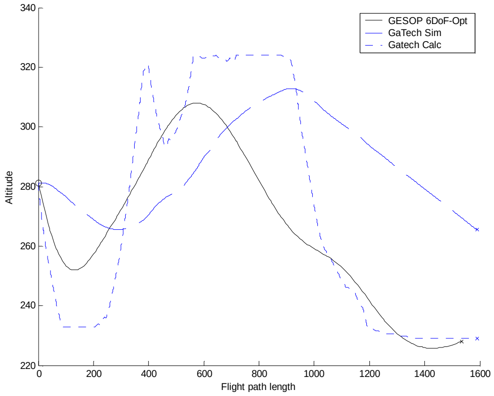


Figure 5.25: Ref.II, K=1 vs. GaTech-Calc & -Sim. - path-altitude

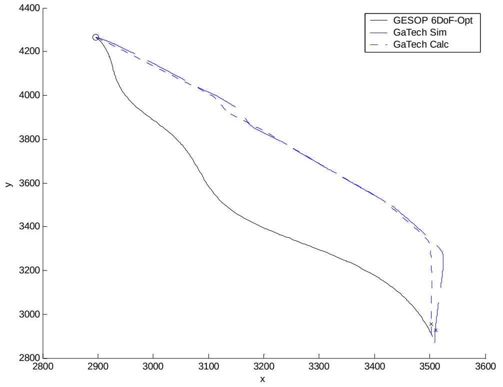


Figure 5.26: Ref.II, K=1 vs. GaTech-Calc & -Sim.-xy

The GaTech path integral with 115,838.64 is about 34.2% higher than the Gesop path integral with 86,308.24.

5.5.2 Optimization of time (K=0)

Now instead of using K=1 for Terrain-Masking optimization, K=0 for Time-Optimization was used in figures 5.27-5.29. The Gesop case this time is the higher case of the Time-Optimization shown above.

Compared to the previous case, the GaTech-Optimization again takes a different path than the 6DoF-Gesop solution. But an optimal solution with (unfortunately not comparable) different weightings on altitude change cost takes the same valley and sticks much closer to the GaTech solution. These results are shown in the figures 5.30-5.32 below.

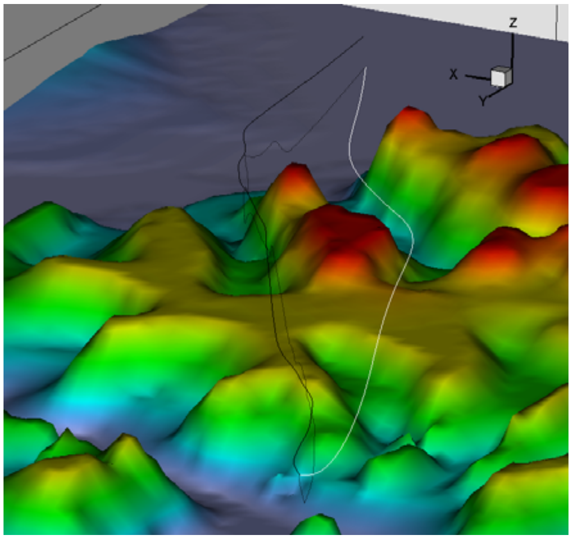
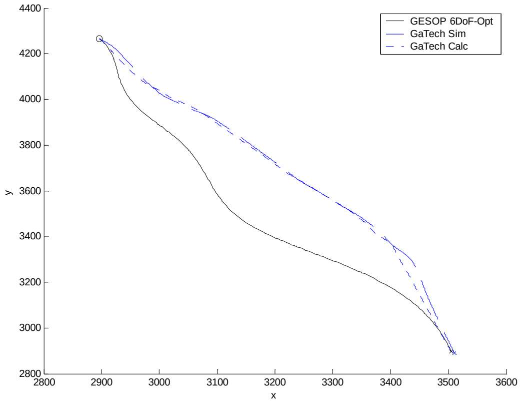


Figure 5.27: Ref.II, K=0 vs. GaTech-Calc & -Sim.-3D



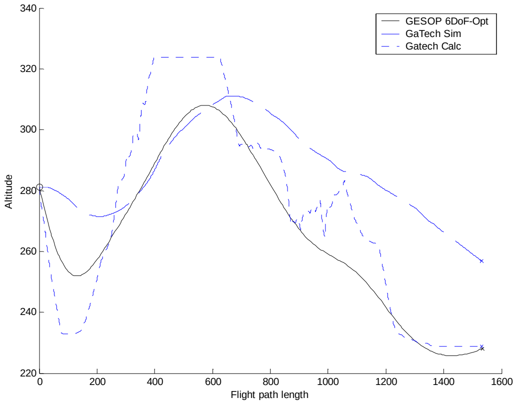


Figure 5.28: Ref.II, K=0 vs. GaTech-Calc & -Sim.-xy

Figure 5.29: Ref.II, K=0 vs. GaTech -Calc & -Sim. - path-altitude

The reason that despite the fact that for K=0 the GaTech Trajectory is not a straight line is the use of the pseudo 3-D formulation. They let the optimizer realize that flying over a hill lengthens the flight path but still doesn't tell it the differences in velocity caused by climbs / descents.

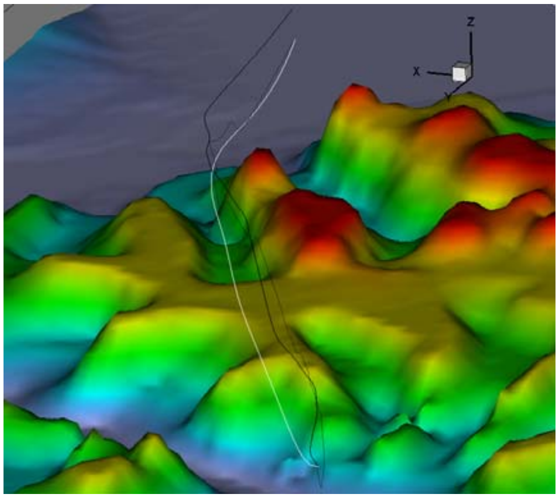


Figure 5.30: Ref.I, K=0 vs. GaTech -Calc & -Sim.-3D – unequal weighting

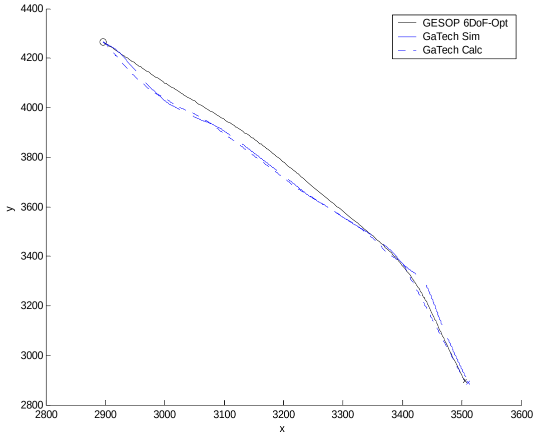


Figure 5.31: Ref.I, K=0 vs. GaTech-Calc & -Sim.-xy – unequal weighting

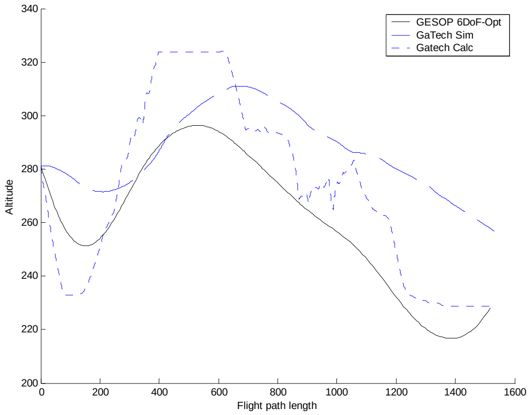


Figure 5.32: Ref.II, K=0 vs. GaTech Calc & Sim.- path-altitude – unequal weighting

Therefore the optimizer tries to avoid high slopes and at the same time it tries to find the shortest possible connection, which is probably the reason why it flies so close nearby the red top mountain at a relatively constant altitude without using the valley to accelerate and reaching the destination quicker. Again the constant velocity constraint inhibits the simulator to be able to reach the destination at the commanded altitude and it is not able to follow the altitude commands closely along the trajectory (figure 5.32). On the other side the *x*-*y*-command is followed very well like before (figure 5.31). This time the simulation's trajectory does not cross the terrain, but if the starting altitude would be lower it would.

In comparison the Gesop solution consequently tries to fly over low profile as much as possible. First the obligatory decent to gain speed and then it climbs the first hill at the latest position possible for this trajectory. Afterwards it flies directly in the middle of the first valley to descent as soon as possible again. So the average overflown altitude is lower in Gesop and therefore the Flight Path – Terrain altitude integral with 86.308.24 is about 31,59% smaller than 113,574,47 for the GaTech-Optimization.

5.5.3 Flight over terrain for a Threat

The white line in figure 5.33 represents the 6DoF-solution, the black dashed line the GaTech-Optimization, the black solid line the GaTech-Simulation when a threat pops-up.

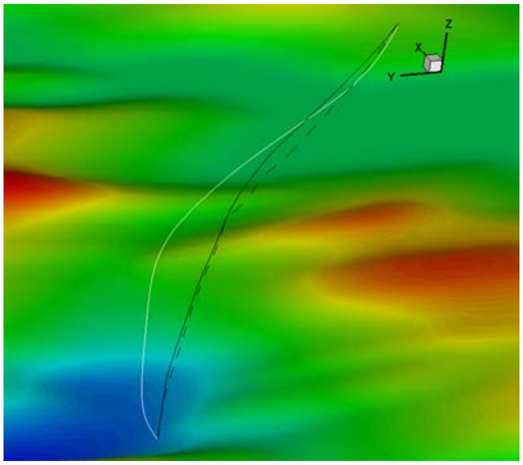


Figure 5.33: Pop-Up-Threat Gesop (white) vs. GaTech (black) - Calc.(dashed) & -Sim.(solid) - 3D - No Threat

In figure 5.34 the trajectories in the *x*-*y*-plane are shown and in figure 5.35 the altitude over the flight path length can be seen. Despite the fact that both optimizations were made with K=1 and the terrain level input for the Lagrange Cost function was normalized to the same values, there still remains a significant difference.

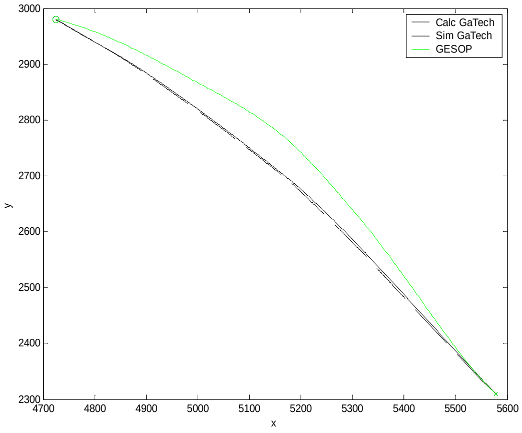


Figure 5.34: Pop-Up-Threat Gesop vs. GaTech-Calc. & -Sim. - xy - No Threat

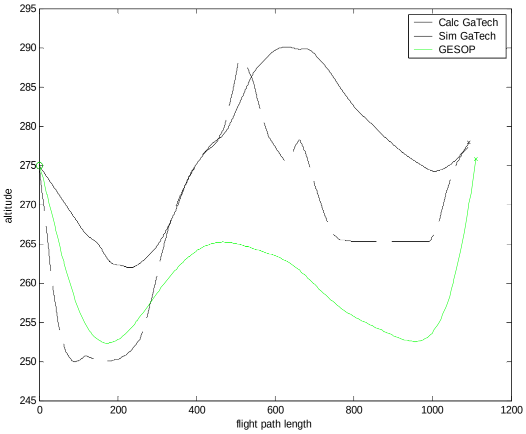


Figure 5.35: Pop-Up-Threat Gesop vs. GaTech-Calc. & Sim. - path-altitude - No Threat

The 6DoF solution takes a longer way compared to the GaTech solution. The reason for this difference is most probably caused by the advantage the 6DoF-Model draws out of the fact that it is able to gain speed much quicker by flying through the blue sink and even more avoiding the higher but shorter path the GaTech solution takes. As described earlier a constant velocity is commanded for the 3DoF-simulation, which does not take into consideration the extension of the path and slow down by first going up and afterwards going down a hill. For a flight going from a point A to a point B it is faster to fly through an imaginary valley rather than to fly over an imaginary mountain with the same size because the average speed is much higher when accelerating in the beginning of the maneuver. The 6DoF-optimizer realizes the advantage of a shorter flight time and therefore smaller Lagrange Integral by quickly gaining speed through the valley in the beginning and also tries to avoid higher mountains since they slow down the airplane. A look at figure 2.53 shows that.

The 6DoF flight is slightly longer but the average flight altitude of the Gesop solution is much lower than the GaTech-Calculation / -Simulation. The altitude difference at the end is most probably caused by the different interpolation methods used (Cubic-Spline-Int. in Gesop vs. Linear-Int. for GaTech). So at this example the differences between 3DoF and 6DoF-Optimiations come out very clearly.

5.5.4 Flights with a Pop-UP Threat

Now during the flight presented in the previous paragraph a Pop-Up-Threat suddenly appears after some seconds of the flight on the optimal trajectory of the airplane so that the airplane must find a new trajectory to avoid the threat. The threat itself is again represented by a Gaussian hill. Figure 5.36 shows the situation. The solid line shows the normal flight path, at the point where the dashed line comes out of the solid line, the wired hill representing the threat suddenly appears. At this point a new optimization is started with all actual state-values as initial boundary conditions for the new optimization. The optimizer calculates the new trajectory which is represented by the dashed line in Gesop. The GaTech-Solver doesn't need to use these initial boundary conditions since it just creates heading commands for the simulator which the Autopilot tries to follow then. So sudden turns are allowed there. The effect can be seen in figure 5.37. The dashed-dotted line is the heading command, the solid line is the simulator's trajectory trying to follow the heading command. Coming from the right bottom, after the appearance of the Pop-Up-Threat the heading command suddenly jumps clockwise. The airplane of course can't follow such quickly and tries to adapt with the delay it needs to turn.

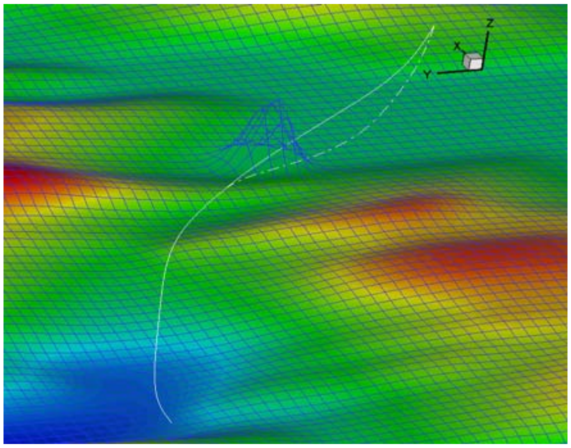


Figure 5.36 Pop-Up-Threat-Gesop Solution-3 D-with Threat (white solid)-without Threat (white dashed)

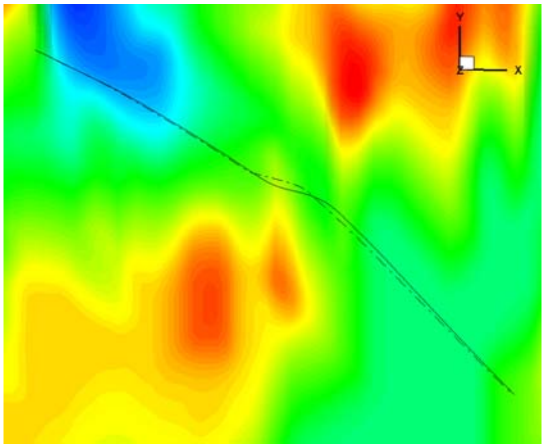


Figure 5. 37 Pop-Up-Threat-GaTech-Calc.(dashed) & -Sim (solid)-with Threat

The 6DoF-Optimization creates this smooth turn already during the optimization because the aircraft model with its dynamics is already built into the optimization loop. Since the basic trajectories in the paragraph before were too different to produce a comparable result with the same (threat-)hill, the hills for the two optimizations were located at different positions so that their tops intersect with the corresponding trajectory. Then the optimizations with the Pop-Up-Threat were set in such a way that the distances of the airplane to the appearing threat were about the same in both cases. In figure 5.38 a comparison between the GaTech-Simulation (black) and the Gesop-Optimization (white) for a Pop-Up-Threat appearing very close in front of the airplane is shown. It is the same case like in figure 5.36 and 5.37. The hills were left out to keep the figure understandable, but figure 5.36 shows the same case with the hill for the Gesop-part. The points where the threat appears are marked by additional small circles in the figures 5.39 and 5.40. The behavior of both solutions is very similar. Both try to avoid the hill by flying around it. The fact that they take different ways around the hills is not significant because slight hill position changes can already switch that.

As can be seen in figure 5.39 the turn rates in the GaTech-Simulation seem to be a little higher than in the 6DoF-Optimization. Reasons for that might be the penalty on the control derivatives, or the relatively wide-meshed control grid, but one certain reason is that the average velocity in the GaTech-Simulation was 100ft/s (30.48m/s) while the average speed of the 6DoF-Optimization with 109.25 ft/s (33.3m/s) was about 10% higher.

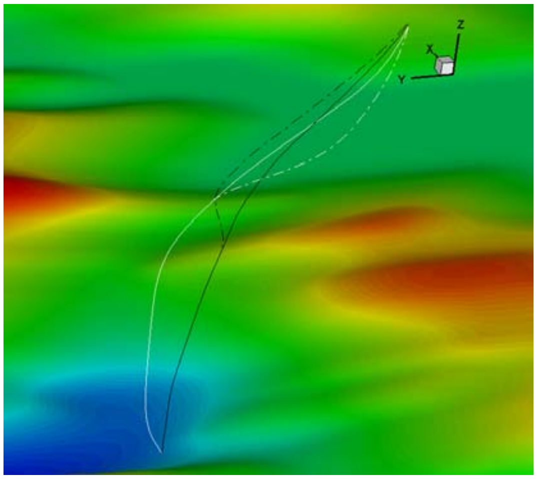


Figure 5.38: Pop-Up-Threat appearance Gesop (white) vs. GaTech Sim.(black) Dashed: With appeared threat, Solid: No appeared threat -3D

In contrast to the undisturbed flight both solution also fly a lower profile in the case with threat (figure 5.40). The explanation for that are most probably the sharp turns. In the 6DoF-case the optimizer tries to keep the velocity as high as possible which leads to a descent during the sharp turn because the lift decreases and a higher lift during the sharp turn would result in a higher drag.

On the other hand for the GaTech case where the simulator tries to fly a constant velocity the loss in lift by the sharp turn maybe can't be compensated.

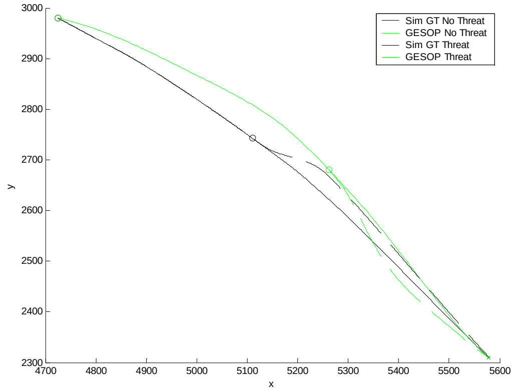


Figure 5.39: Pop-Up-Threat appearance Gesop (green) vs. GaTech Sim.(black) Dashed: With appeared threat, Solid: No appeared threat -xy

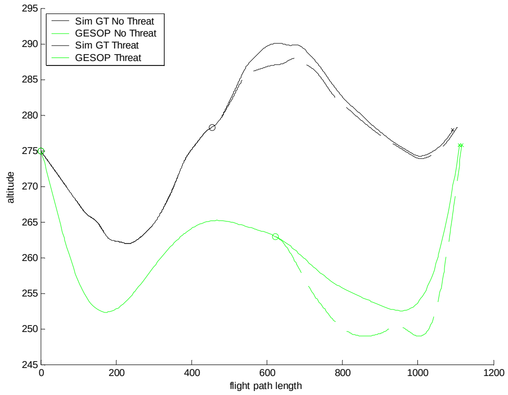


Figure 5. 40: Pop-Up-Threat appearance Gesop (green) vs. GaTech Sim.(black) Dashed: With appeared threat, Solid: No appeared threat - path-alt.

5.5.5 Earlier pop-up Threat

In the dotted lines in figure 5.41 shows the result when the Pop-Up-Threat appears earlier and the aircraft has more time to avoid it. As expected the trajectory leaves the original path earlier. After some time it matches the dashed trajectory of the first Pop-Up Threat in both cases. This shows that behind the threat the optimal path is about constant and can be reproduced even under different circumstances.

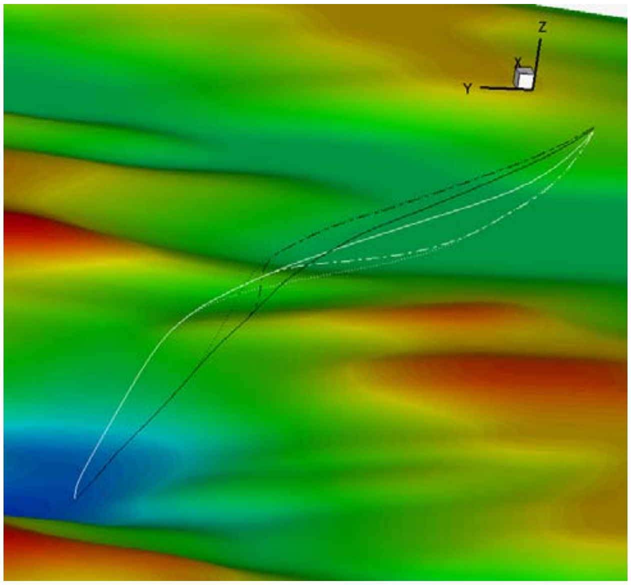


Figure 5.41: Early Popup-Threat Gesop (white) vs. GaTech (black) Calc. & Sim. - Dotted: early appearance of threat, Dashed: late appearance of threat - 3D

6. Flight Demonstrations

This section includes flight demonstration results in which an optimal trajectory is implemented in real-time using the reduced-order formulation. The pop-up obstacle is simulated as a pop-up mountain in a virtual environment.

6. 1 Integration Details

The planner was integrated into the Rmax environment for execution on the second flight computer (onboard2). A list of tasks being performed on onboard2 is shown in Figure 6.3.

The general sequence of events when flying the planner includes the following.

```
1. On gcs, load obstacle database into memory on the
   ground station (gcs).
     a. wdbset gcswdb
     b. wdbclear
     c. wdbdatumned
     d. @ropdb
     e. scene0.redraw=1
     f. scene1.redraw=1
     g. scene2.redraw=1
2. On gcs, instruct onboard1 to start sending extman0
   messages
     a. rc obDatalinkSet.sendExtMan0 = 1
3. On onboard2; Set planner parameters; an important
   parameter is, hc_const, which must be set to the
   nominal altitude above the terrain that one wants the
   vehicle to fly:
     a. cd reducedOrderPlanner
     b. W         = -0.05
```

```
        c. V_stall   = 5
        d. flag_real = 1
        e. e         = 1
        f. w_end     = 500
        g. hc_const  = 300
        h. scale_lam = 1
        i. T = 50
   4. On onbard2, set up history recording by executing
        a. @rophist
   5. On onbard2, get ready for loading the obstacle database
        a. wdbset ob2wdb
        b. wdbclear
   6. On gcs, issue a command that transmits the datum used
      to initialize onboard1, to onboard2.
        a. wdbdatumob2
   7. On onbard2, load the obstacle database
        a. @ropdb
   8. On onboard2, Start the rop planner task
        a. taskStart rop
   9. On onboard2, instruct the onboard2 main loop to call
      ropTraj when an external maneuver message (extman0) is
      received
        a. onboard2.runRop=1
```

The above steps are encoded as input files and executed in the following sequence.

```
   1. on gcs do @ropgcs
   2. on ob2 do @ropob2
   3. on gcs do wdbdatumob2
   4. on ob2 do @ropob2b
```

At this point, the planner task is executing on onboard2. Various waypoints may be setup on the gcs, some that may include passing through obstacles. The various pseudo-codes for the tasks executing on onboard2 are shown in Figure 6.3. An overview of the messages being send and received across the various computers are shown in Figure 6.4.

6.2 Simulation

As an example, the planner was executed in simulation. In Figure 6 1: **Example gcs view during before using the planner** segment1 passes through an obstacle. In this case the obstacles are hills with a maximum altitude of 300 ft.

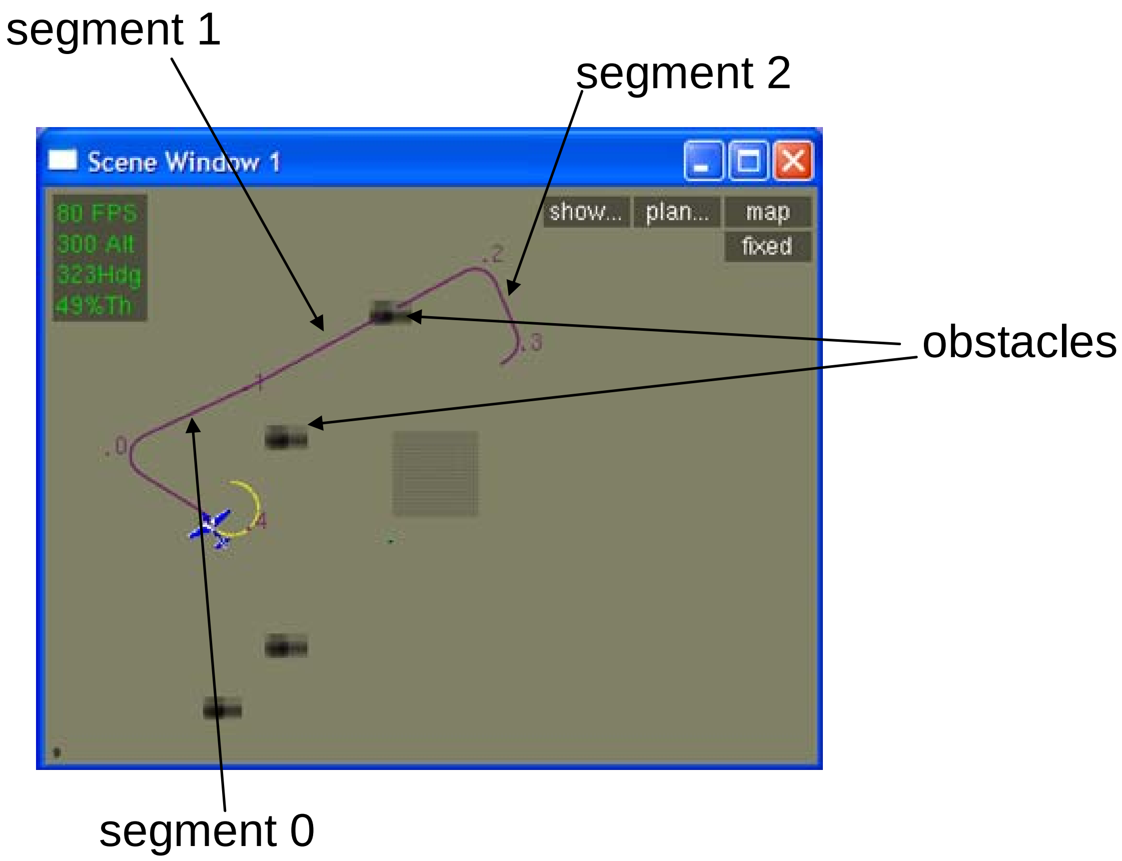


Figure 6 1: Example gcs view during before using the planner

The planner running on onbard2 may be instructed to replan segment 1 (between waypoint 1 and waypoint 2), by issuing

```
replan 1
```

at the gcs console. Figure 6 **2** now shows the trajectory for segment 1 generated by the planner in blue.

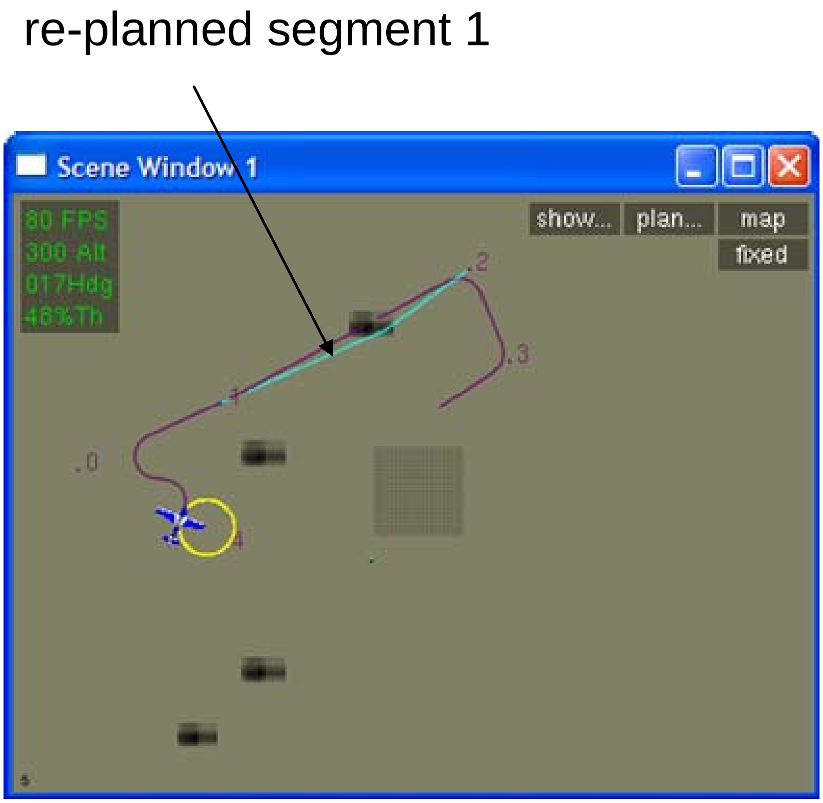


Figure 6 2: gcs after segment 1 has been replanned. The blue segment indicates trajectory generated by the planner

Making sure that waypoint 2 has the type MAN_EXT, the trajectory may be uploaded and executed using the following commands.

```
trajUpload

trajGo
```

Figure 6.5 shows the aircraft at various instants, tracking the planner generated trajectory. Figure 6.6, shows a close up of the position response around the hill and Figure 6.7 shows the position response for the entire set of waypoints.

6.3 Experimental Results

A flight test was conducted on the 11th of July 2007 – utilizing the GTYak research UAV.

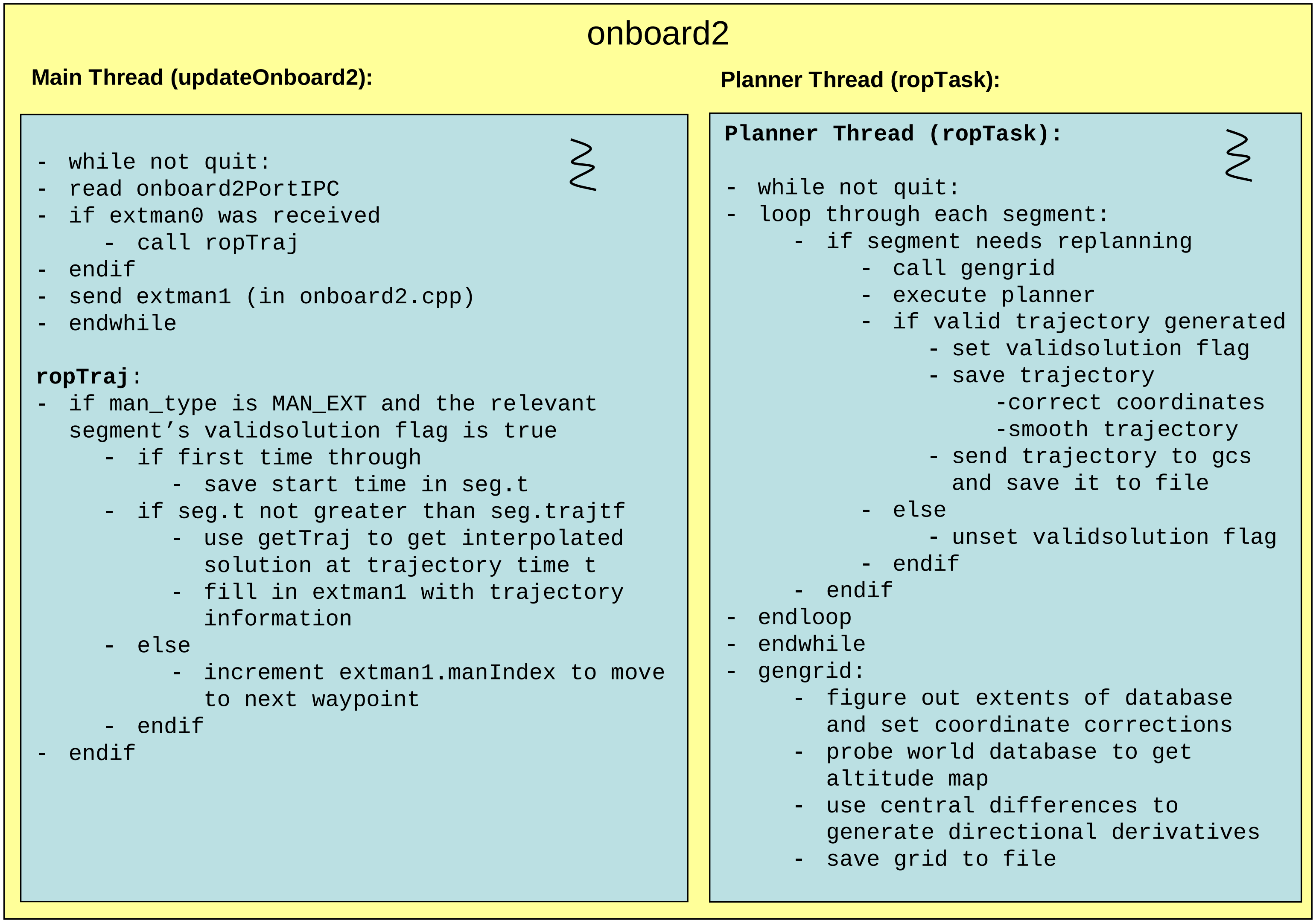
Pictures (files starting with GTYak): http://uav.ae.gatech.edu/pics/f070711_flight_FormationFlight_ObstacleAvoidance/
Video:
http://uav.ae.gatech.edu/videos/y070711b1_planner1.wmv

During the test, a simulated hill was placed at 0,0,0 with a maximum altitude of 300 ft. The setup for simulation and flight is exactly the same, except that gcs, onboard1 and onboard2 are executing on different computers. Figure 6.8, shows the trajectory around a simulated hill. The offset in lateral position is due to the initial condition offset when entering the external planner's segment. This offset may be alleviated by introducing a straight line segment that passes through the last waypoint before the planner's segment is executed. Figure 6.9 shows position plots versus time.

6.4 Relevant Files

File name	Description
rop.h, rop.cpp	Standalone planner algorithm
rop.db	declares data structures used by the planner during execution. It also contains data structures used to store planned segments
wdb.h, wdb.cpp	World Database. Contains code to represent, add, and remove obstacles. Also contains a function to query local terrain altitude taking obstacles into account.
wdb_scene.h, wdb_scene.c	World Database. Contains code to draw obstacles in the scene window
ropint.h, ropint.cpp	main code that integrates the planner into the GTMAX environment. Contains procedures to generate grid data from the world database, makes coordinate conversions between planner and GTMAX NED coordinates. It also contains functions which, given a pre-planned trajectory segment, will use interpolation to generate

	trajectories real-time during flight.
ropgcs.inp	execute during flight test at gcs, loads database from ropdb.inp, sends extman0
ropob2.inp	some parameters for the planner, activates ob2wdb, the onboard2 world database and clears it, also setups up history recording variables
ropob2b.inp	execute on ob2, loads database from ropdb.inp which should be uploaded to ob2, also starts the planner thread
ropdb.inp	contains list of obstacles and loaded at gcs and ob2
ropsim.inp	setup up sim environment for testing rop
rophist.inp	history variables for data recording



onboard2

Main Thread (updateOnboard2):

```
- while not quit:
- read onboard2PortIPC
- if extman0 was received
     - call ropTraj
- endif
- send extman1 (in onboard2.cpp)
- endwhile

ropTraj:
- if man_type is MAN_EXT and the relevant
  segment's validsolution flag is true
     - if first time through
          - save start time in seg.t
     - if seg.t not greater than seg.trajtf
          - use getTraj to get interpolated
            solution at trajectory time t
          - fill in extman1 with trajectory
            information
     - else
          - increment extman1.manIndex to move
            to next waypoint
     - endif
- endif
```

Planner Thread (ropTask):

```
Planner Thread (ropTask):

- while not quit:
- loop through each segment:
     - if segment needs replanning
          - call gengrid
          - execute planner
          - if valid trajectory generated
               - set validsolution flag
               - save trajectory
                    -correct coordinates
                    -smooth trajectory
               - send trajectory to gcs
                 and save it to file
          - else
               - unset validsolution flag
          - endif
     - endif
- endloop
- endwhile
- gengrid:
     - figure out extents of database
       and set coordinate corrections
     - probe world database to get
       altitude map
     - use central differences to
       generate directional derivatives
     - save grid to file
```

Figure 6.3: Overview of tasks on onboard2

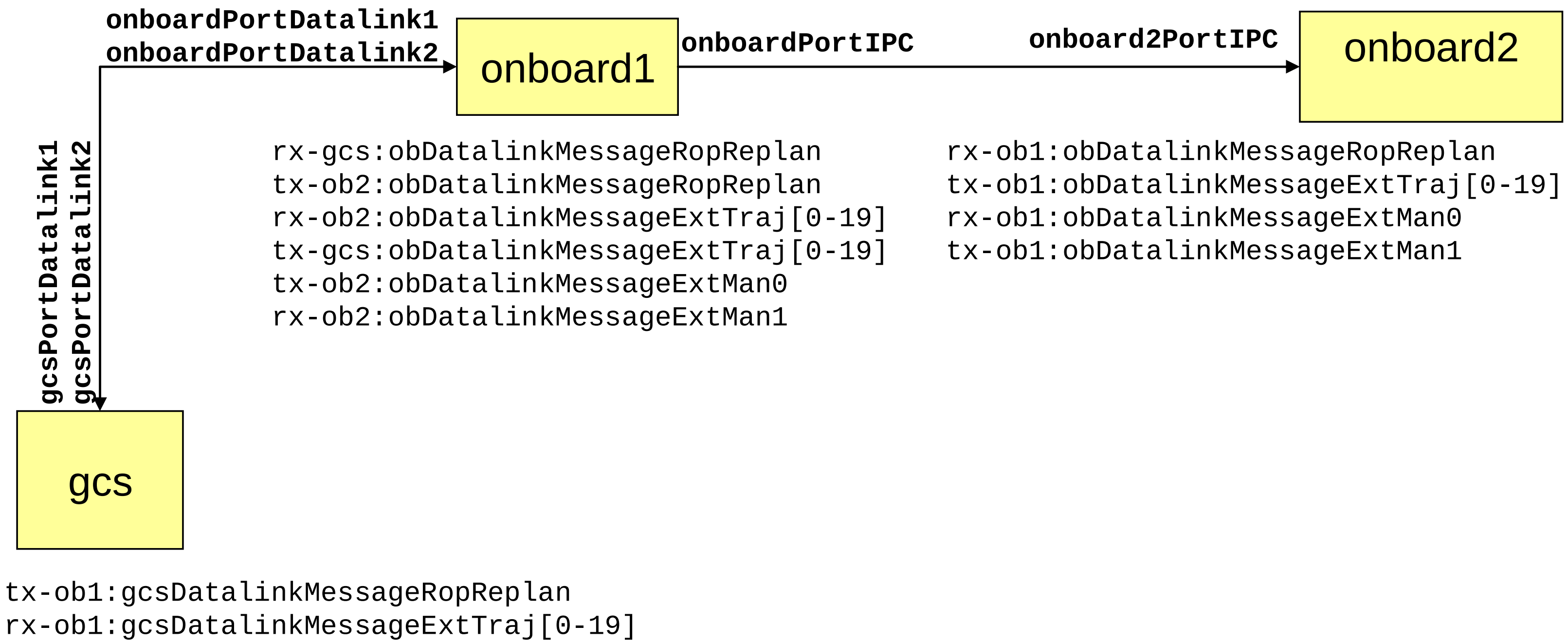

onboardPortDatalink1
onboardPortDatalink2
onboard1
onboardPortIPC
onboard2PortIPC
onboard2
gcsPortDatalink1
gcsPortDatalink2
rx-gcs:obDatalinkMessageRopReplan
tx-ob2:obDatalinkMessageRopReplan
rx-ob2:obDatalinkMessageExtTraj[0-19]
tx-gcs:obDatalinkMessageExtTraj[0-19]
tx-ob2:obDatalinkMessageExtMan0
rx-ob2:obDatalinkMessageExtMan1
rx-ob1:obDatalinkMessageRopReplan
tx-ob1:obDatalinkMessageExtTraj[0-19]
rx-ob1:obDatalinkMessageExtMan0
tx-ob1:obDatalinkMessageExtMan1
gcs
tx-ob1:gcsDatalinkMessageRopReplan
rx-ob1:gcsDatalinkMessageExtTraj[0-19]

Figure 6.4: Overview of transmission and reception of messages relevant to the planner

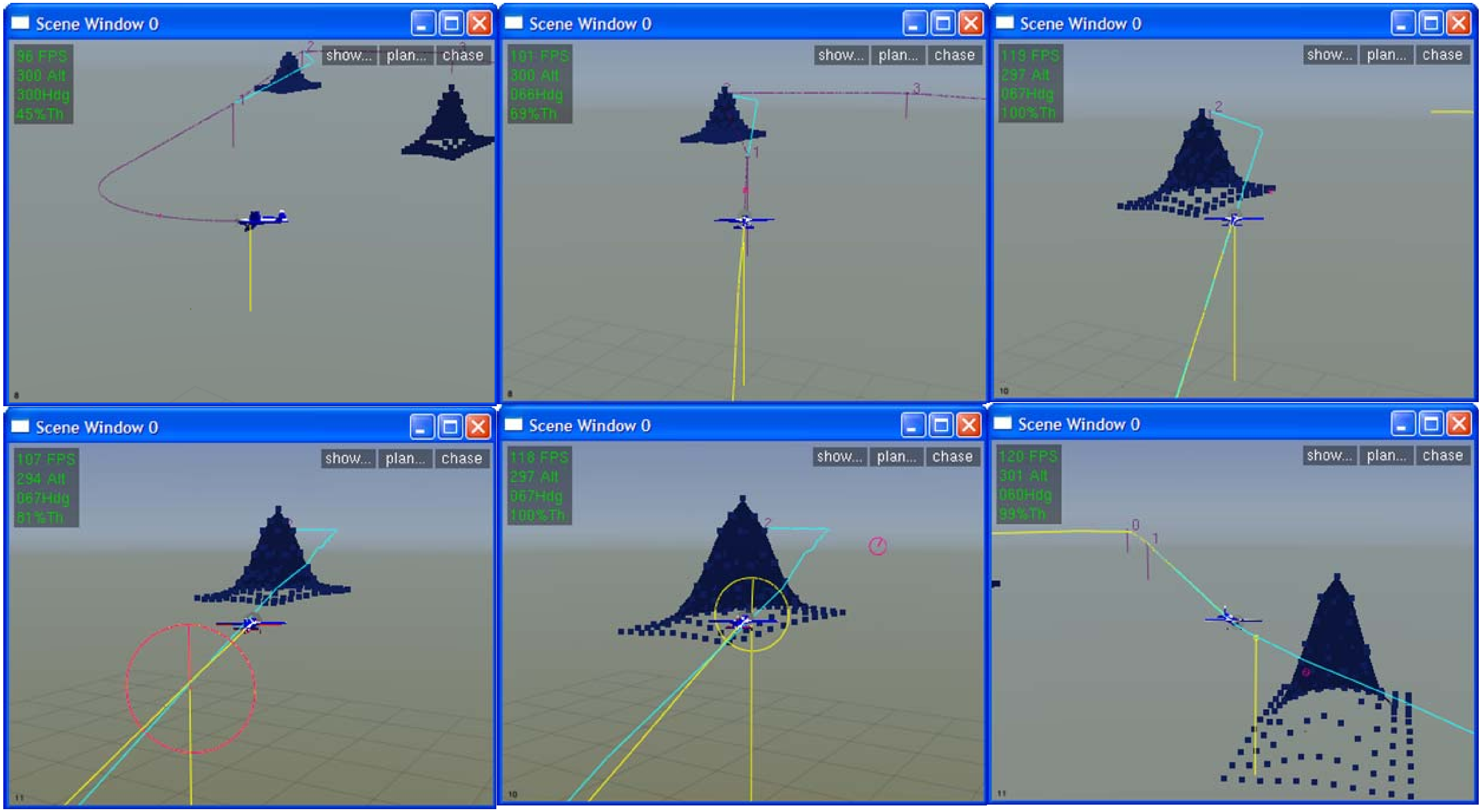


Figure 6.5: Images at various instants of the maneuver

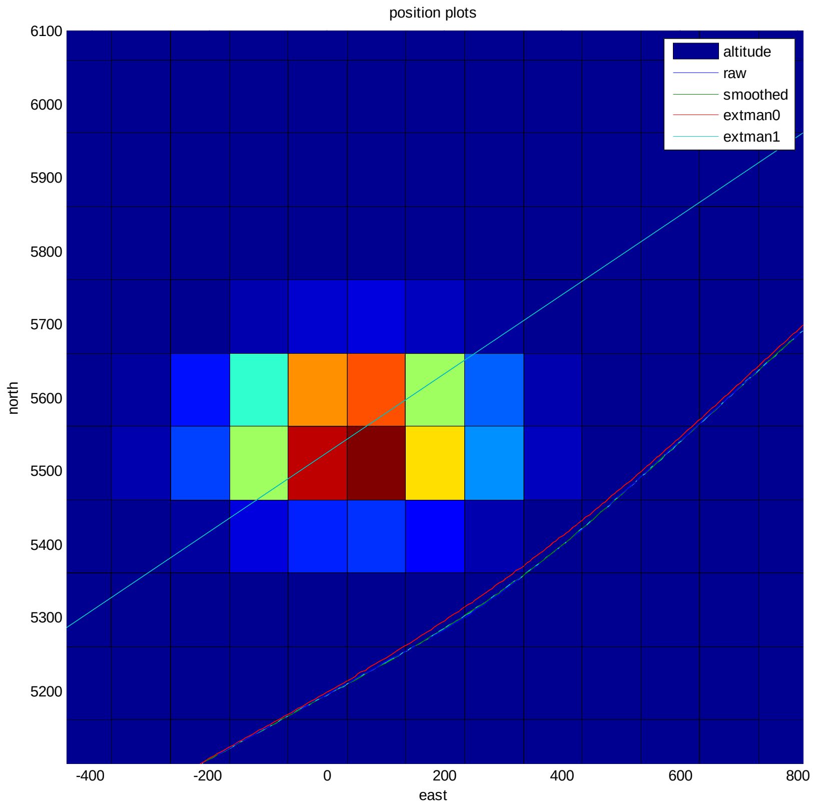


Figure 6.6: Plot of trajectory around hill. extman0 represents the actual trajectory of the vehicle, while extman1 represents the commanded trajectory

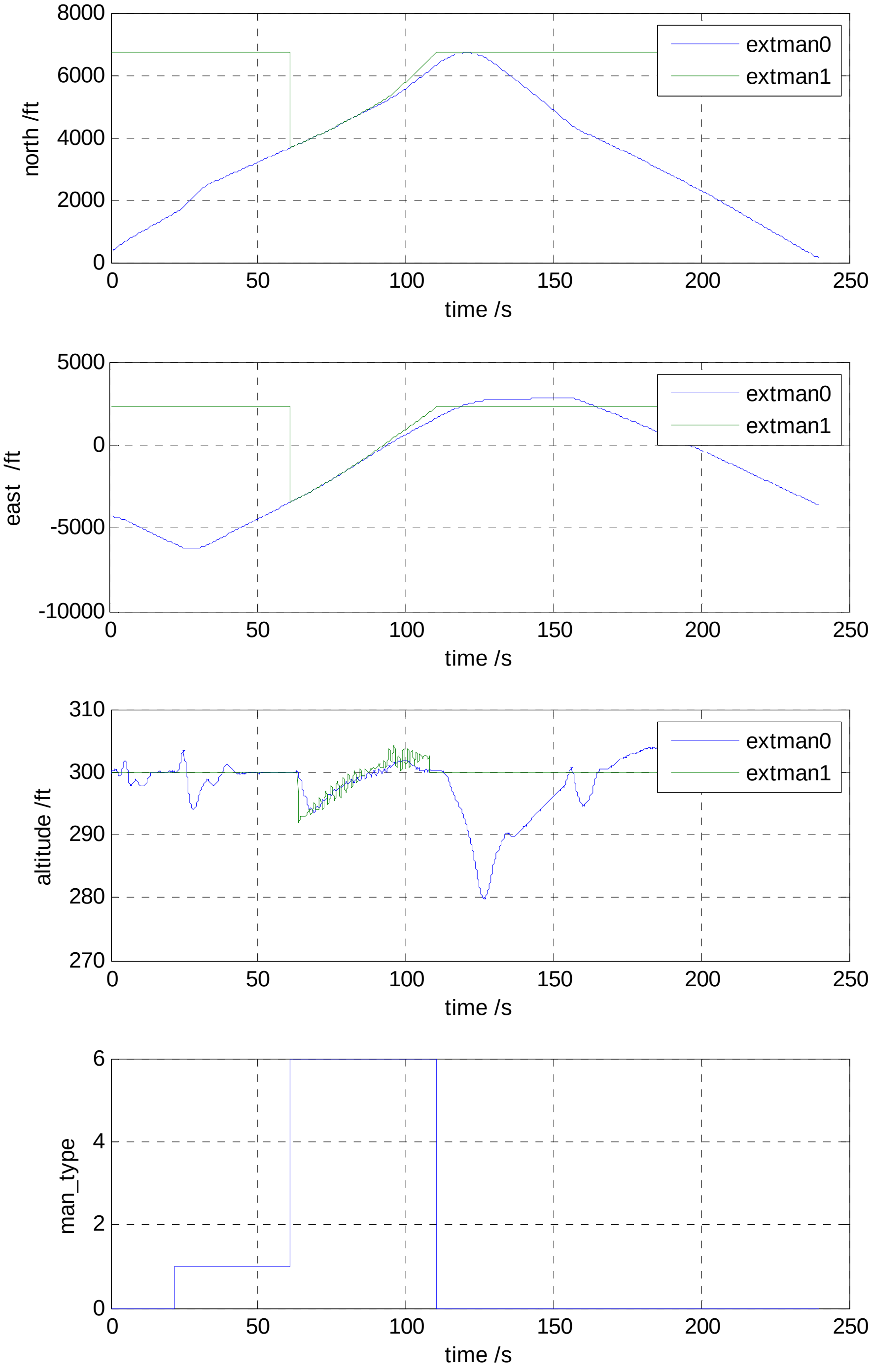


Figure 6.7: Plot of the various position command (extman1) and response (extman0) versus time. The segment where man_type is 6 corresponds to the segment generated by the planner.

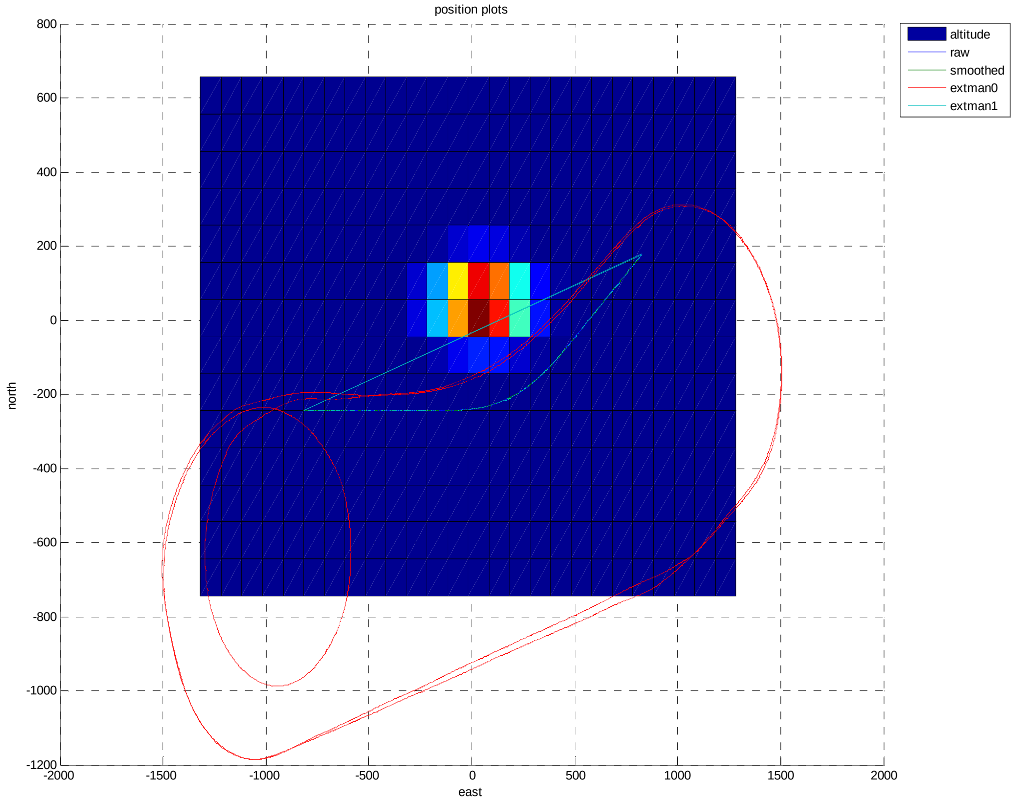


Figure 6.8: Plot of trajectory around a simulated hill.

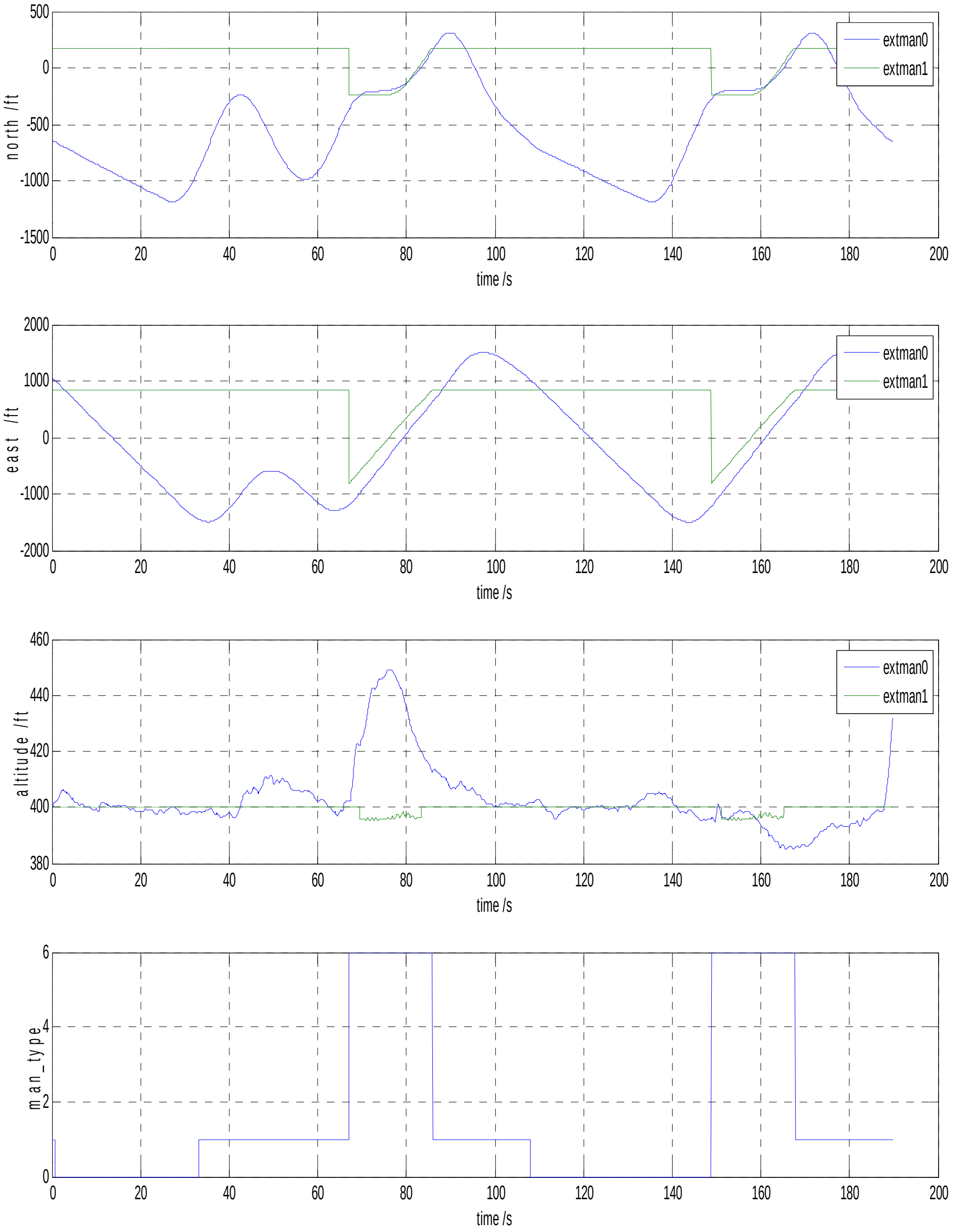


Figure 6.9: Plot of position responses during flight. extman1 represents the command and extman0 represents the response. The segment where man_type is 6 corresonds to the trajectoyr generated by the planner

7. Conclusions and Recommendations

In the Phase II effort we extensively addressed trajectory optimizations in incrementally complex formulations starting from the pseudo 3-D formulation, via 3-D point mass model to the full 6 DOF aircraft dynamics. Optimizations over mathematical and real terrains were investigated with moving targets, moving threats, and pop-up threats while the aircraft moves along the terrain. The pseudo 3-D reduced-order formulation is flight demonstrated for replanning an optimal path when a virtual obstacle pops up using a fixed wing flight test-bed. This illustrates that the resulting optimal trajectory from the pseudo 3-D formulation is numerically efficient and can be implemented for trajectory optimization for autonomous aerial vehicles.

Comparison studies of the pseudo 3-D solutions with the optimal solutions obtained using the full 6 DOF aircraft dynamics reveal various subtle issues in path planning regarding the sub-optimality of the reduced-order solution. In the case of terrain-masking and flight time minimization, our research during the Phase II efforts led to the following observations.

First, towards the task of terrain-following, the pseudo 3-D formulation is associated with the reduced-order point-mass equation that incorporates the terrain information in the phase of problem formulation. This leads to an trajectory that maintains a constant altitude over the terrain map with a constant maximal velocity to minimize the flight time[13]. While this is a reasonable assumption for those trajectories in which vertical portion of velocities are negligible, this assumption may be violated when the aircraft flies over a series of steep hills and deep valleys. In case of 6 DOF optimizations, terrain following is achieved by adding additional cost terms in the performance index. Figure 7.1 compares each costs in the case of terrain-following (K=1) for the optimal trajectory with the real terrain data, which reveals that the cost for the terrain-following dominates the overall performance index. Figure 7.2 shows the same costs as in Figure 7.2 for a non-optimal solution with the 6 DOF aircraft dynamics. Figure 7.2 reveals that there is a flight trajectory for which additional costs for maintaining a constant altitude over the terrain is at the same magnitude of the terrain following (at least for a while), and this implies that the reduced-order optimal solution that minimizes the only portion of the terrain-following cost may quite deviate from the 6 DOF trajectory.

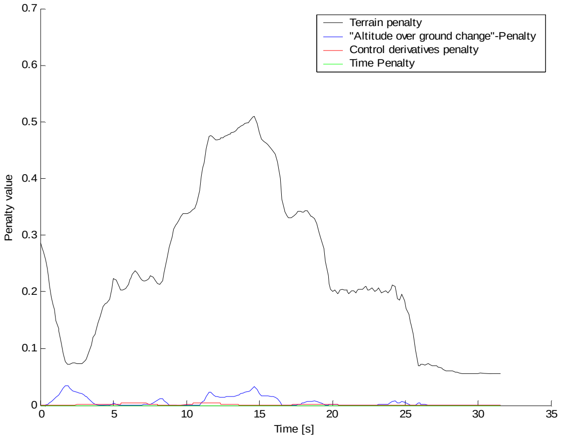


Figure 7.1: Comparison of costs for optimal trajectory for terrain-following

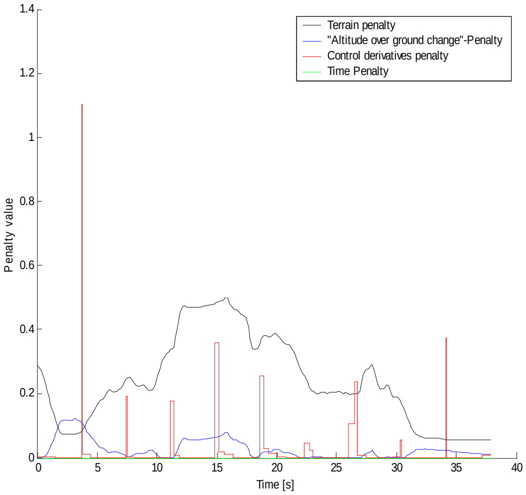


Figure 7.2: Comparison of costs for a non-optimal trajectory for terrain- following

The spikes shown in Figure 7.2 also imply that the rate of control signals should be constrained to result in a flyable trajectory. However, this has to induce additional constraints on the states used as controls in the reduced-order formulation. This is also related to the next issue.

Second, heading constrains with the reduced-order formulation leads to very close optimal trajectory to the 6 DOF optimal solution in case of generic hills in which optimal trajectories veer around hills (for example see Figure 5.10 and 5.16). With the real terrain data, the main differences between the pseudo 3-D solutions and 6 DOF optimal solutions result from altitude variations. As a matter of fact, the pseudo 3-D solutions are not flyable in the f-wing simulator in some cases. Considering that heading-constraint generates a trajectory close to an optimal trajectory with 6 DOF optimizer, this observation implies that additional constraint or formulation is required to address neglected part of the vertical dynamics.

Third, it was found that an optimal solution may be sensitive to initial conditions. For example, in Figure 5.20 in Section 5.4.3, the pseudo 3-D solution passes a threat in front while the 6 DOF optimal solution passes the target behind. Changing an initial guess for the heading angle resulted in a solution that is close to the 6 DOF optimal trajectory. An interesting question is how different costs are for these seemingly discrete trajectories. The study reveals that the difference in cost is not significant. Likewise, for 6 DOF optimization, several different trajectories of the same level of costs were available, and small changes in the weighting factors in the performance index could lead to quite different trajectories as a result of more possible combinations for states and control variables. Following the above observations, further research for on-line path optimization is recommended to address the following.

First, in the current effort, the pseudo 3-D optimization led to similar trajectories, with comparable cost, to those of 3-D formulation as shown in Section 5.2. This became a basis for ensuing study of extending the reduced-order formulation into multi-vehicle formulation that will benefit the research on multi-vehicle formation and cooperation in performing tasks. More quantitative analysis is, however, required with varying degree of complex terrains. In particular, analysis of the degree of the sub-optimality due to neglected portion of the dynamics will provide clear insight on the feasibility of the current formulation in more general terrain and mission scenarios.

Second, a main drive for the reduced-order formulation was numerical efficiency in a general setting of solving necessary conditions for optimality with the singular perturbation theory as an analysis tool for the neglected portion of the dynamics. However, formulating the aircraft dynamics in a singular perturbation form requires identification of a parasitic parameter which reveals inherent time scales in the aircraft dynamics, which is probably most difficult part of applying the singular perturbation method. As a matter of fact, a large effort has been made for a systematic methodology for selecting time-scaled state variables in general nonlinear optimal control[31-34], and writing the original dynamics in a singular perturbed form is still a main theoretical issue[14]. Performing more rigorous study on the form of the terrain-following dynamics may allow for proper corrections if the reduced-order solution leads to unflyable trajectories as has been done in [17]. An alternative method based on pseudo-control hedging[35] allowed the vehicle to fly a modified trajectory in simulation, which is discussed in the last paragraph of this section.

Third, finding equivalent constraints in the reduced-order formulation in case when the real control signals are position and rate limited is worth study. In particular, for the current scenario of terrain-following, the assumption of constant velocity over the constant altitude over the terrain is unrealistic with the real terrain and led to a solution that differs from the true 6 DOF optimization. However, properly constrained heading rate, the resulting trajectory is very close to the full-order optimizer. This implies that further study on adding constraints on the vertical motion may lead to similar results as was done for X-Y trajectories. Some directions, such as limiting the angle of attack or formulating the reduced-order problem in modified energy form, were proposed in Phase I final report.

Fourth, while the pseudo 3-D formulation employed in this study was an attempt to reduce the number of optimization variables by exploiting analyticity of optimally conditions in a singular perturbed dynamics, various schemes that tries to reduce the number of optimization variables also exist. For real-time trajectory generation, the notion of differential flatness has recently been introduced and attracted many researchers because the reduction of the number of variables can be performed in utilizing inherent structure of the dynamical systems in flat systems, and therefore dynamic constraints, the equation of states, are automatically met[36-38]. While this sounds very promising in terms of numerical computation, recent research reveals that the ease in computation offered by the reduction of the number of constraints in flatness-based methods can also be achieved by exploiting spare linear algebra in solving the underlying optimization problem [39]. Further, real-time trajectory generation is not merely driven by the number of constraints or optimization variables but fundamentally intertwined with optimization principles such as convexity[40]. We note that flatness-based methods belong to the category of direct methods[1], and direct comparison between flatness-based methods and the reduced-order formulation is not feasible. However, with the given scenario of terrain-masking, moving targets and obstacles, feasibility study on the structural property such as flatness of the air vehicle dynamics will further shed light on the structure of the air vehicle trajectory optimization and strengthen the understanding on the viability of the reduced-order formulation.

Finally, while the reduced-order optimal solution in a certain case leads to a trajectory that can not be tracked with the full aircraft dynamics, the neural network-based autopilot hedges the unfollowable portion of the guidance command by estimating achievable portion of trajectories by aircraft dynamics[41] and still maintains reasonable flights. For example, Figures 5.25, 5.29, and 5.32 shows that the altitude command is being tracked reasonably well considering significant fluctuation of original commands. This raises a question if guidance-command hedging mechanism can be used to correct the feasibility of the optimal solutions so that a resulting flight path is near optimal to the original full-order optimal trajectory. Some implications provided by this study, such as simplification of re-planning and robustness to the modeling error that will destroy the optimality anyway if the aircraft dynamics are uncertain, may be worth further study.

References

[1] J. T. Betts, "Survey of numerical methods for trajectory optimization," *Journal of Guidance Control and Dynamics*, vol. 21, pp. 193-207, 1998.

[2] A. E. Bryson and Y.-C. Ho, *Applied optimal control*. New York: Wiley, 1975.

[3] D. G. Hull, "Conversion of optimal control problems into parameter optimization problems," *Journal of Guidance Control and Dynamics*, vol. 20, pp. 57-60, 1997.

[4] A. E. Bryson, M. N. Desai, and W. C. Hoffman, "Energy-State Approximation in Performance Optimization of Supersonic Aircraft," *Journal of Aircraft*, vol. 6, pp. 481–488, 1969.

[5] A. E. Bryson and M. M. Lele, "Minimum Fuel Lateral Turns at Constant Altitude," *AIAA Journal*, vol. 7, pp. 559-560, 1969.

[6] H. J. Kelley, "Aircraft maneuver optimization of reduced-order approximation," *Control and dynamic systems, Advances in Theory and Applications*, vol. 10, pp. 131-178, 1973.

[7] H. J. Kelley and T. N. Edelbaum, "Energy Climbs, Energy Turns, and Asymptotic Expansions," *Journal of Aircraft*, vol. 7, pp. 93–95, 1970.

[8] A. J. Calise, "Singular Perturbation Methods for Variational Problems in Aircraft Flight," *Ieee Transactions on Automatic Control*, vol. 21, pp. 345-353, 1976.

[9] A. J. Calise, "Extended Energy Management Methods for Flight Performance Optimization," *Aiaa Journal*, vol. 15, pp. 314-321, 1977.

[10] A. Calise, "A new boundary layer matching procedure for singularly perturbed system," *Automatic Control, IEEE Transactions on*, vol. 23, pp. 434-438, 1978.

[11] A. J. Calise, "Singular Perturbation Techniques for Online Optimal Flight-Path Control," *Journal of Guidance and Control*, vol. 4, pp. 398-405, 1981.

[12] M. D. Ardema, "Singular Perturbations in Flight Mechanics," NASA TM X-62,380, August 1974.

[13] P. K. A. Menon, E. Kim, and V. H. L. Cheng, "Optimal Trajectory Synthesis for Terrain-Following Flight," *Journal of Guidance Control and Dynamics*, vol. 14, pp. 807-813, 1991.

[14] D. S. Naidu and A. J. Calise, "Singular perturbations and time scales in guidance and control of aerospace systems: A survey," *Journal of Guidance Control and Dynamics*, vol. 24, pp. 1057-1078, 2001.

[15] A. J. Calise, "Singular perturbations in flight mechanics," *IN: Applied mathematics in aerospace science and engineering(A 98-22897 05-31), New York, Plenum Press(Mathematical Concepts and Methods in Science and Engineering.*, vol. 44, pp. 115-132, 1994.

[16] D. B. Price, A. J. Calise, and D. D. Moerder, "Piloted Simulation of an Onboard Trajectory Optimization Algorithm," *Journal of Guidance Control and Dynamics*, vol. 7, pp. 355-360, 1984.

[17] J. E. Corban, A. J. Calise, and G. A. Flandro, "Rapid near-Optimal Aerospace Plane Trajectory Generation and Guidance," *Journal of Guidance Control and Dynamics*, vol. 14, pp. 1181-1190, 1991.

[18] T. Ries, "Full Dynamics 6 DOF-Trjaectory-Optimizations of an Unmanned Aeronautical Vehicle," in *Aerospace Engineering*, vol. Diploma: University of Stuttgart, 2005.

[19] S. Twigg, "Optimal Path Planning for Single and Multiple Aircraft using a Reduced Order Formulation," in *School of Aerospace Engineering*, vol. Ph. D.: Georgia Institute of Technology, 2007.

[20] S. Twigg, A. Calise, and E. Johnson, "On-line trajectory optimization for autonomous air vehicles," *AIAA Guidance, Navigation, and Control Conference*, pp. 2003–5522, 2003.

[21] S. Twigg, A. Calise, and E. Johnson, "On-Line Trajectory Optimization Including Moving Threats and Targets," *AIAA GNC Conference. August*, 2004.

[22] S. Twigg, A. Calise, and E. Johnson, "Trajectory Optimization for Multiple Vehicles Using a Reduced Order Formulation," *Infotech@ Aerospoace*, pp. 1-10, 2005.

[23] S. Twigg, A. J. Calise, and E. Johnson, "3D Trajectory Optimization for Terrain Following and Terrain Masking," presented at AIAA Guidance, Navigation, and Control Conference, Keystone, CO, 2006.

[24] S. Twigg, A. J. Calise, and E. Johnson, "Trajectory Optimization for n-Vehicles using a Reduced Order Formulation," presented at AIAA Guidance, Navigation, and Control Conference, Keystone, CO, 2006.

[25] R. M. Bray, "A Wind Tunnel Study of the Pioneer Remotely Piloted Vehicle," vol. MS Thesis. Monterey, CA: Naval Postgraduate School, 1991.

[26] E. Falkenauer, *Genetic algorithms and grouping problems*. New York: Wiley, 1998.

[27] P. Vas, *Artificial-intelligence-based electrical machines and drives : application of fuzzy, neural, fuzzy-neural, and genetic-algorithm-based techniques*. Oxford ; New York: Oxford University Press, 1999.

[28] K. Kulankara, "Machining fixture synthesis using the genetic algorithm," in *Mechanical Engineering*, Ph D. Thesis: Georgia Institute of Technology, 1999.

[29] P. E. Gill, W. Murray, and M. A. Saunders, "SNOPT: An SQP algorithm for large-scale constrained optimization," *Siam Journal on Optimization*, vol. 12, pp. 979-1006, 2002.

[30] USGS, "United States Geological Survey-Terrain Data." February 3: ftp://edcftp.cr.usgs.gov/pub/data/DEM/250/, 2003.

[31] H. J. Kelley, "Reduced-Order Modeling in Aircraft Mission Analysis," *AIAA Journal*, vol. 9, pp. 349-&, 1971.

[32] M. Ardema and N. Rajan, "Slow and Fast State Variables for 3-Dimensional Flight Dynamics," *Journal of Guidance Control and Dynamics*, vol. 8, pp. 532-535, 1985.

[33] M. D. Ardema and N. Rajan, "Separation of Time Scales in Aircraft Trajectory Optimization," *Journal of Guidance Control and Dynamics*, vol. 8, pp. 275-278, 1985.

[34] A. J. Calise, N. Markopoulos, and J. E. Corban, "Nondimensional forms for singular perturbation analyses of aircraft energy climbs," *Journal of Guidance, Control, and Dynamics*, vol. 17, pp. 584-590, 1994.

[35] E. N. Johnson and A. J. Calise, "Limited authority adaptive flight control for reusable launch vehicles," *Journal of Guidance Control and Dynamics*, vol. 26, pp. 906-913, 2003.

[36] S. K. Agrawal and N. Faiz, "Optimization of a class of nonlinear dynamic systems: New efficient method without Lagrange multipliers," *Journal of Optimization Theory and Applications*, vol. 97, pp. 11-28, 1998.

[37] N. Faiz, S. K. Agrawal, and R. M. Murray, "Trajectory planning of differentially flat systems with dynamics and inequalities," *Journal of Guidance Control and Dynamics*, vol. 24, pp. 219-227, 2001.

[38] M. Guay and J. F. Forbes, "Real-time dynamic optimization of controllable linear systems," *Journal of Guidance Control and Dynamics*, vol. 29, pp. 929-935, 2006.

[39] I. M. Ross and F. Fahroo, "Pseudospectral methods for optimal motion planning of differentially flat systems," *Ieee Transactions on Automatic Control*, vol. 49, pp. 1410-1413, 2004.

[40] I. M. Ross and F. Fahroo, "Issues in the real-time computation of optimal control," *Mathematical and Computer Modelling*, vol. 43, pp. 1172-1188, 2006.

[41] E. N. Johnson and S. K. Kannan, "Adaptive Trajectory Control for Autonomous Helicopters," *Journal of Guidance, Control, and Dynamics*, vol. 28, pp. 524-538, 2005.