ABSTRACT

Title of Document: EMERGENCY FLIGHT PLANNING FOR THE GENERALIZED TRANSPORT MODEL AIRCRAFT WITH LEFT WING DAMAGE

Yunshen Tang, Master of Science, 2006

Directed By: Assistant Professor, Ella M. Atkins, Department of Aerospace Engineering

A nontrivial fraction of aviation accidents are caused by in-flight damage or failures that reduce performance. Researchers are working to ensure future avionics recognize the impact of damage/failures and guide the aircraft to a safe landing. This thesis presents an end-to-end Adaptive Flight Planner (AFP) for such emergencies and applies it to a damage situation in which a Generalized Transport Model (GTM) aircraft loses a significant fraction of its left wingtip. Trimmed (non-accelerating) flight conditions define the post-damage/failure aircraft flight envelope. A landing site search algorithm is augmented to define the reachable landing footprint and to prioritize the feasible landing runways within this region. End-to-end landing trajectories are constructed as a sequence of trim states and corresponding transitions. An LQR-based PID nonlinear controller enables the damaged GTM aircraft to correctly track trajectory commands over trimmed flight and transition segments. A suite of emergency scenarios are used to evaluate AFP performance.

EMERGENCY FLIGHT PLANNING FOR THE GENERALIZED TRANSPORT MODEL AIRCRAFT WITH LEFT WING DAMAGE

by

Yunshen Tang

Thesis submitted to the Faculty of the Graduate School of the
University of Maryland, College Park, in partial fulfillment
of the requirements for the degree of
Master of Science
2006

Advisory Committee:

Assistant Professor Ella M. Atkins, Chair
Associate Professor Robert M. Sanner
Assistant Professor Benjamin Shapiro

Dedication

To my family.

Acknowledgements

I would like to express my gratitude to my advisor, Dr. Ella M. Atkins, for her guide and support. I would also like to acknowledge Dr. Rob M. Sanner for his valuable advice and insights throughout the thesis work, and Dr. Benjamin Shapiro for serving on my advisory committee.

I would like to thank NASA Ames Research Center for providing and sponsoring the research project to make everything possible. Thanks to Igor and Matt for building the foundation for this work.

I'm grateful to the Department of Aerospace Engineering at University of Maryland, College Park for giving me the opportunity to pursue my graduate study here. Thanks to the faculty and staff in the department for making my life much easier. Special thanks to the Chinese fellows in the department. Among them, Wei, Min, and Qina bring to me countless help and courage, as well as many unforgettable good times together.

Special thanks to my parents and my sister for encouraging and supporting me. Thanks to Lu for your patience. You are always there when I need help.

Table of Contents

List of Tables

List of Figures

Chapter 1: Introduction

Modern aviation is a safe and reliable form of transportation. In 1971, fatal accidents on commercial jetliners occurred approximately once in every 140 million miles flown. Thirty years later, jets fly 1.4 billion miles for every fatal accident. Although a ten-fold safety improvement has been made, accidents, especially the fatal ones, do occur every year [1]. In-flight aircraft damage and failure cause a significant fraction of remaining commercial aviation accidents. To make flying safer, research activities are being conducted to improve the reliability of aircraft avionics and mechanical systems to avoid or provide early warning of failures, as well as to provide pilots with the ability to cope with in-flight emergencies when they occur.

A number of aircraft failure and damage scenarios can severely reduce aircraft performance. For example, an aircraft cannot climb or maintain its altitude in the emergency situation where engine failure causes the total loss of thrust. When such failures occur, the pilot faces several challenges simultaneously. First, he must recognize the reduced aircraft performance sufficient to maintain long-term control of the aircraft, a problem managed today by researchers in adaptive control and system identification. Additionally, the pilot must quickly select a landing site and plan a landing trajectory to that site that is actually possible to fly given the reduced performance characteristics. This latter problem is less studied in the research community and is the focus of this work.

History contains numerous success and failure stories related to landing aircraft with failed or damaged systems which illustrate the complexity and importance of this work. On November 22, 2003, a European Air Transport A300B4-203F was hit by an SA-7 'Grail' missile after take-off from Baghdad International Airport. The aircraft rapidly lost all hydraulic pressure and thus controls. The crew found that after extending the landing gear to create more drag, they could pilot the plane using differences in engine thrust and managed to land the plane back at Baghdad airport with minimal additional damage. However, many emergencies result in tragedy due to failure to make decisions efficiently and correctly. As part of NASA Aeronautics' Damage Adaptive Control (DAC) program, researchers are collaborating to ensure future avionics systems ably assist the pilot with a safe landing when emergencies occur during flight. Other researchers have focused on system identification and adaptive control problems. The goal of this thesis, which builds on the previous work of Alonso Portillo [2] to select a feasible post-failure landing site and Strube [3] to build segmented landing trajectories, is to implement and evaluate an integrated software package for automatic post-failure flight planning to enable safe emergency landings and to apply this tool to a generalized transport model (GTM) aircraft with substantial structural damage.

1.1 Related Work

To cope with in-flight emergencies, numerous concepts and technologies have been developed. While some of them are trying to solve high-level issues, others are focused on more specific problems.

1.1.1 Advanced Flight Management System Architectures

To date, the primary function of a flight management system or FMS is to assist pilots with accurate autopilot guidance, navigation, and control (GNC), flight plan management given company route databases, and monitor performance and fuel throughout the flight. By providing inputs to the autopilot and throttles, the FMS can guide an aircraft through a complex set of speed, course and altitude changes from the airport of origin to the destination, greatly reducing pilot workload [4]. However, to extend assistance to cases with in-flight emergencies, the FMS must be expanded beyond route databases and tabulated performance models.

Researchers have begun to design flight management architectures that can help the pilots with decision-making during emergencies. An Emergency Flight Planner (EFP), proposed by Chen and Pritchett [5], is such a prototype. The modules of this proof-of-concept system include an automatic plan generator, trajectory predictor, autopilot, pilot interface, and an aircraft model identification toolbox. The core functionality of the planner is the ability to predict the aircraft trajectory resulting from a given plan, i.e., list of actions. Their experiment results show that an EFP may be a useful tool to prevent the pilots from performing unsafe tasks when emergencies occur. Although their architecture is general, Chen and Pritchett's work focuses on the interactions between automation and human. Similar research activities can be found in other literature [6-8].

While human factors are typically emphasized in research related to piloted aircraft, there has been a growing interest in designing highly efficient autonomous flight control systems for UAVs. It is envisioned that future UAVs will be required to perform complex tasks in presence of very large environmental uncertainties, including flight-critical failures [9]. Jovan, et al., present a multi-layer autonomous intelligent control architecture to achieve these highly autonomous objectives for UAVs [9]. Their Hierarchical Control Structure is consisted of four functional layers: Autonomous Decision Making Layer, Autonomous Path Planning Layer, Autonomous Trajectory Generation Layer, and Redundancy Management Layer. In this architecture, fault-tolerant and reconfigurable control was integrated with the guidance and path-planning loops to achieve truly autonomous operation under different upsets, failures and unanticipated events. While a Failure Detection and Identification block in the Redundancy Management Level detects and identifies emergencies, a novelty Achievable Dynamic Performance (ADP) Calculation block in the Autonomous Trajectory Generation Level calculates the ADP measure for planning and decision making in the higher levels. The ADP calculation module is the primary feature of this architecture. The layers in the architecture are connected through ADP calculation module, which results in a system in which all the decisions are made based on optimum use of the current available resources. Simulation studies show the potential of this architecture to achieve future highly autonomous flight control for UAVs.

In general, the key augmentations to a flight management system that enable emergency situation management typically are: system identification tools, an autopilot that can adapt to altered performance characteristics, a trajectory generator, and a full flight plan (waypoint) generator. A pilot or operator interface is also necessary so long as the aircraft is not fully-autonomous. In addition to these functionalities, this thesis also discusses an approach that enables the flight management system to automatically select the desired emergency landing site.

1.1.2 Enabling Technologies

The augmented FMS outlined above need the support from more specific technologies to enable "conceptual modules" in an architectural diagram to robustly perform their designated functions. As listed above, primary technologies that enable emergency flight management system include: adaptive or reconfigurable control, path/trajectory generation, and waypoint (target or landing site) selection. While specifically emergency landing site selection is unique to this line of research, summaries of the significantly more prevalent adaptive control as well as path and trajectory generation research are provided below.

- Adaptive/Reconfigurable Control

An emergency flight management system will require an inner-loop controller capable of assuring rapid stabilization of the overall system in the presence of failures/damages. As an integrated or distinct (but linked) capability, system identification is also required to improve knowledge of aircraft models through on-line adaptation. Significant related research has been conducted in this area. In their

research work, Ahmed-Zaid, et. al.[10], augmented the flight control system of an F-16 aircraft with a hybrid adaptive linear quadratic (LQR) control scheme, resulting in an improved system which had the on-line capability of learning and accommodating to drastic changes in the aircraft dynamics due to surface or hardware failure. This work also adopts an LQR strategy for controlling the damaged GTM aircraft.

Other adaptive flight controllers have been used to accommodate debilitating situations ranging from control surface failures [11] to airframe damage [12]. Much research has been conducted in the area of fault-tolerant and reconfigurable control designs for both piloted and unmanned aircraft [13-15]. Simulation and actual flight test results have shown the potential of reconfigurable control techniques to maintain the desired aircraft performance in presence of severe control surface failures and structural damage. As a new concept in the field of adaptive/reconfiguration control , intelligent control, the control techniques that use various Artificial Intelligence (AI) computing approaches, can achieve the objectives in the presence of very large uncertainties [9]. Intelligent flight controllers have been developed to achieve post-failure/damage aircraft control [16, 17]. A neural network-based flight controller was used to stabilize the F-15 aircraft under conditions including locked/failed control surfaces, as well as unforeseen in-flight damage [17]. Flight-test results show that the controller can adapt to substantial changes in aircraft dynamics by updating stability and control derivative values from an online learning neural network in conjunction with a real-time parameter identification algorithm. An intelligent flight control system, augmented with linear programming theory and adaptive critic technologies

was designed and tested for a simulated Boeing C-17 aircraft [16]. Full motion piloted simulation studies demonstrate the potential of this system for improving handling qualities and yielded significantly increased survivability rates under various simulated failure conditions.

- Path/Trajectory Planning

When emergencies occur, it is crucial for the flight management system to produce a feasible flight plan in near-real time. This task is typically performed by a path/trajectory generator in the flight management system. To achieve on-line real-time path planning, a variety of motion-planning techniques have been applied in 2-D robotic applications [18], and some of those techniques have been introduced to aerospace.[9] Other techniques that have been used in aerospace include probabilistic-map-based [19] and Voronoi diagram [20], particularly for cluttered military environments. Autonomous path planning is a popular topic among UAV researchers. For the path planner in their four-layer autonomous UAV control architecture [9], Boskovic and Mehra developed a simplified two dimensional path planning algorithm. Focused on military applications/collision avoidance, this algorithm can reconfigure/change the pre-computed path in the case of a pop-up threat. An evolution-based path planner was used for autonomous UAV oceanic search missions [21]. Simulation studies showed the capability of the algorithm to consider realistic weather information. A similar strategy has been used for multiple autonomous vehicles operating in dynamic uncertain environments [22]. These market-based planning algorithms were developed for allocating tasks and planning

paths simultaneously. Simulation results show that the planning system can be used effectively for both off-line planning and on-line re-planning in a dynamic uncertain environment. While path planner typically generates a desired flight path representing sequences of positions without considering aircraft's dynamic information, the objective of an automatic aircraft trajectory generator is to fit a feasible trajectory leading to the destination, based on the aircraft dynamics and control input constraints. Richards, Sharma, and Ward presented a path/trajectory planning approach for waypoint planning and obstacle avoidance [23]. They used a modified A* search algorithm to rapidly assemble UAV motion primitives from the motion-primitive libraries, resulting in trajectories that explicitly account for the vehicle's flight envelope and nonlinear motion constraints. Simulation examples have demonstrated the efficacy of this approach, as well as the ease of reconfigurable planning. Yakimenko proposed a direct method for real-time generation of near-optimal spatial trajectories for short-term maneuvers [24]. This method transforms the trajectory optimization problem into a nonlinear programming problem and then solves the problem using an appropriate algorithm with an accelerated time scale. Faiz, et al., used an approach based on differential flatness and Linear Matrix Inequalities to achieve on-line trajectory planning, in near real-time, given the state equations, path and actuator constraints, and the initial and terminal constraints [25]. Dever, Mettler, Feron, and Popovi synthesized feasible rotorcraft trajectories via parameterized maneuver classes. They developed a dynamically feasible trajectory interpolation algorithm to generate a continuous family of vehicle maneuvers across a

range of boundary conditions while enforcing nonlinear system equations of motions as well as nonlinear equality and inequality constraints.

While the trajectories generated by these approaches are appropriate for autonomous UAVs given that performance constraints are met, commercial and general aviation pilots prefer flight plans specified as a sequence of constant-trim flights and transitions between these trims because so-called “segmented” routes can be comprehended intuitively by human pilots and air traffic controllers [3, 26]. A good demonstration of this concept can be found in Frazzoli’s work [27]. Frazzoli defined a hybrid-automaton-based modeling language for UAV trajectories based on the interconnection of a finite number of suitably-defined motion primitives and maneuvers. Based on this modeling language, feasible and optimal motion plans can be computed efficiently by several algorithms. When applying this approach to motion planning for a closed-loop X-Cell 60 helicopter, he specifies the notion of motion primitives as trim trajectories, and maneuvers as a set of transitions between two different trim trajectories. To be clear here, trim trajectories are produced by an aircraft that maintains trimmed (non-accelerating) flight conditions. A precise definition of aircraft trim state will be given in Chapter 2. The concept of motion primitive has been successfully used in the path/trajectory planning by many researchers, thus providing an inspiration to this thesis work. As a fact, this thesis applies Frazzoli’s hybrid framework and trim-trajectory motion primitives to a Generalized Transport Model (GTM) aircraft with left wing damage, building upon

Strube's [3] initial development of a trim-based trajectory generation process given reduced performance characteristics.

1.1.3 Emergency Flight Management Heritage

As the foundation of this thesis, an Emergency Flight Management Architecture was presented in [2, 3], as illustrated in Figure 1-1. The Adaptive Flight Planner (AFP) in this architecture was designed to help pilots maintain aircraft safety when in the presence of failures and/or damage. The Flight Plan Monitor propagates the existing flight plan through the post failure/damage performance model to verify the feasibility of the flight plan. If the executing flight plan is infeasible, the pilot is notified via the Pilot Interface. Meanwhile, the AFP is activated to generate a new flight plan. The Landing Site Search (LSS) sub-module is responsible for finding a safe landing site, hopefully a runway that is judged safe based on the post-failure/damage aircraft performance model. The Segmented Trajectory Planner then constructs a dynamically-feasible trajectory to the landing site.

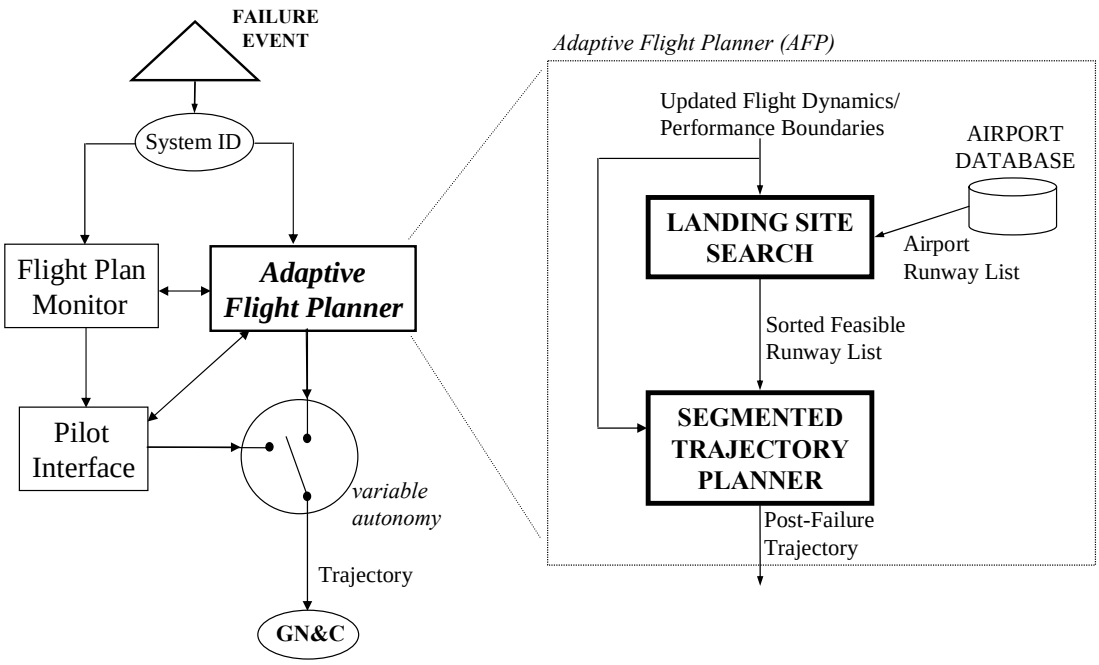

Figure 1-1: Emergency Flight Management Architecture [2]

1.2 Thesis Objectives and Method

The objective of this thesis is to describe and evaluate an automatic post-failure/damage flight planning capability to enable safe emergency landings. This work integrates previously-developed landing site search and path planning methods into the end-to-end Adaptive Flight Planner (AFP), as illustrated in Figure 1-1. Building upon previous efforts that focus on actuator jam failures and loss of thrust, a Generalized Transport Model (GTM) aircraft with substantial left wing damage is used as the model in this work. Use of the GTM requires conversion of an aerodynamic model for characterization of trim states and simulation-based validation of planned flight trajectories. The severity of the damage also requires careful control

law tuning to minimize transients and to maintain stability in the presence of disturbances.

Efficient algorithms are used to identify and rank feasible landing runways based on the post-failure/damage aircraft performance limits. First, the footprint generation algorithm computes the post-failure/damage aircraft footprint that takes into account the updated performance model and runway proximity to practically constrain remaining time aloft. The approach end and heading of the highest-priority feasible runway is sent to the trajectory generator as the terminal point (landing site).

This work augments an existing trim sequence trajectory planner to construct a complete trajectory leading the aircraft from the initial location at which the emergency occurs to the desired landing runway, by a sequence of trim trajectories and the maneuver transitions between neighboring trim states. The trajectory planner uses an efficient constant-time algorithm to rapidly generate the initial part of the trajectory bringing the aircraft from the initial location to the intermediate location from which the final landing trajectory will be planned. This initial trim sequence generated from static “descend and head toward landing site” rules are designed to provide sufficient time for generation of the precise landing trajectory. The “approach” trajectory planner uses a combination of discrete search and continuous local optimization algorithms to construct a precise final landing trajectory that connects the post-descent state to the desired landing runway.

The trajectory planner requires aircraft trim states and maneuver transitions as motion primitives. Once a suite of feasible trim states is developed, the effects maneuver transitions between two trim trajectories must be analyzed; otherwise plans either would not be executable (in the case where such transition is unstable/impossible) or else the plan would not result in an accurate landing (in the case where such transition results in appreciable deviation in aircraft state parameters). An LQR-based gain scheduling controller is developed and tuned in this work for the severely-damaged GTM to enable stable transitions between two trim flights as well as accurate closed-loop maintenance of trim states that are "stabilizable" but not inherently stable. The aircraft configuration changes during feasible trim state transitions are computed and stored so that the trajectory planner can account for them during its final approach planning processes.

1.3 Thesis Outline

Before further delving into the implemented AFP or simulation-based results, Chapter 2 presents a brief overview of rigid fixed-wing aircraft dynamics, as well as issues related to the Generalized Transport Model aircraft with left wing damage. Chapter 2 also provides a more formal definition of "aircraft trim state" for later discussions. Chapter 3 reviews the Landing Site Search module and defines a backward-compatible algorithm for footprint generation applicable to the damaged GTM. Chapter 3 also discusses the impact of utility function weight changes on the identification and ranking of candidate landing runway. Chapter 4 introduces the trimmed flight analysis processes, including the calculation of the trajectories incurred by trimming the flight over a non-zero but finite time interval. Chapter 4 also

presents the characterization of transition flights between trim states, as well as the development and the performance of an LQR-based gain scheduling controller tuned to minimize the impact of transitions through minimal overshoot and relatively fast response times. Chapter 5 presents the two-step end-to-end landing trajectory generation process, as well as the algorithms used in both steps. Chapter 6 presents a case study examining efficacy of the integrated Adaptive Flight Planner on the left wing damage scenario for a GTM aircraft. The thesis ends in Chapter 7 with a brief summary and the suggestions for future work.

Chapter 2: Aircraft Kinematics and Dynamics

This chapter first presents a brief review of flight dynamics and kinematics for rigid-body fixed-wing aircraft, and then describes the GTM aircraft model with damaged left wing used as an emergency flight planning case study in this thesis. The last section provides a formal definition of aircraft trim state.

2.1 Aircraft Flight Kinematics and Dynamics

With the origin of the body coordinate system chosen to be at the aircraft center of gravity (CG), the equations of motion can be decoupled into two independent sets of translational and rotational equations. The rotational motions of the body will then be the three rotational motions about the center of gravity. The other components of motion are the three translational motions of the CG, typically specified with respect to an Earth-based inertial reference frame. Twelve state variables are required to describe the state model for a six degree-of-freedom rigid body in three-dimensional space. Latitude, longitude, and altitude components specify inertial position with respect to the fixed Earth frame. Three components of linear velocity are needed to specify translational kinetic energy, and three components of angular velocity to specify rotational kinetic energy. Three final state variables are needed to specify the orientation of the body, with a ZYX Euler angle representation adopted for this work.

2.1.1 Axes and Angles

The conventional body-fixed coordinate system of an aircraft is shown with respect to the relative wind direction in Figure 2-1. The positive X axis goes out the nose of the aircraft; the positive Y axis goes out the right wing of the aircraft; and the positive Z axis goes out the belly of the aircraft.

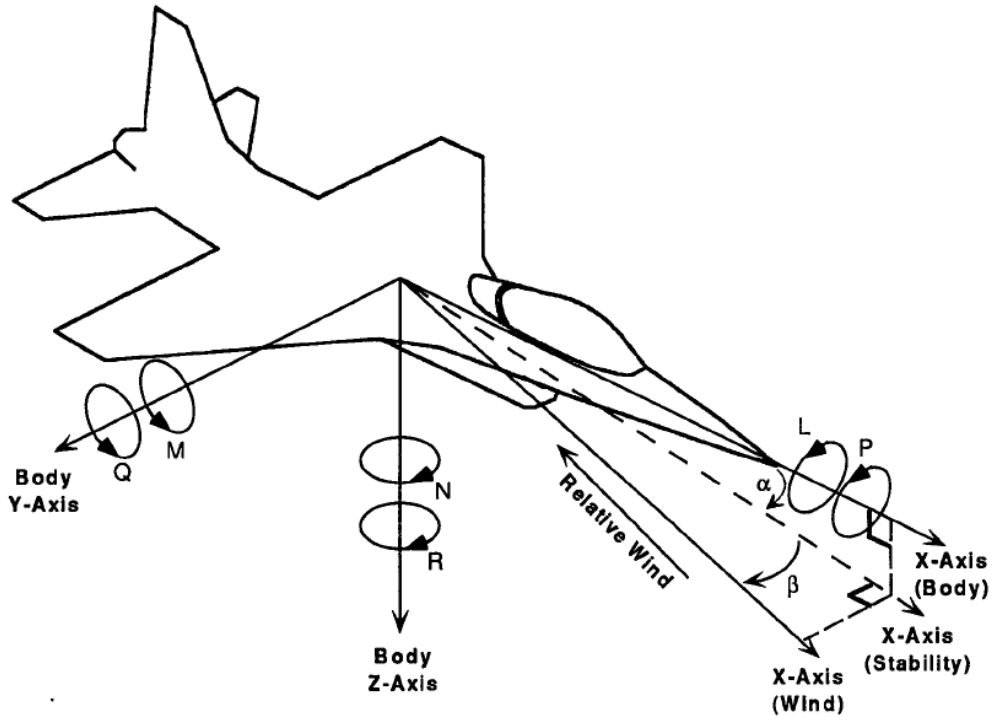

Figure 2-1: Definition of Axes, Aerodynamic Angles, Moments and Angular Velocities

The aerodynamic forces and moments on an aircraft are produced by the relative motion with respect to the air. Two orientation angles with respect to the airflow specify the aerodynamic forces and moments: angle of attack (α) and sideslip angle (β). These aerodynamic angles can be defined by performing a rotation α about the body Y axis, followed by a rotation β about the new Z axis, such that the final X axis is aligned directly into the relative wind. The first rotation defines the stability axes,

and the angle of attack is the angle between the body X axis and the stability X axis. α is positive if the first rotation about the body Y axis is negative; thus a positive α is shown in Figure 2-1. The second rotation β defines wind axes, indicating the angle between the stability X axis and the wind X axis. β is positive if the rotation about the stability Z axis is positive; thus a positive β is shown in Figure 2-1.

The transformation from body to stability axes is given by

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_S = \begin{bmatrix} \cos\alpha & 0 & \sin\alpha \\ 0 & 1 & 0 \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_B = R_\alpha \begin{bmatrix} x \\ y \\ z \end{bmatrix}_B \tag{2.1}$$

while the transformation from stability axes to wind axes is

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix}_W = \begin{bmatrix} \cos\beta & \sin\beta & 0 \\ -\sin\beta & \cos\beta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \end{bmatrix}_S = R_\beta \begin{bmatrix} x \\ y \\ z \end{bmatrix}_S \tag{2.2}$$

These two rotations are denoted by R_α and R_β as shown above, and the complete rotation from body axes to wind axes is denoted by ${}^W_B R$. Therefore, if v_B is a vector expressed in body axes, and v_W is the same vector expressed in wind axes,

$$v_w = R_\beta R_\alpha v_B = {}^W_B R v_B \tag{2.3}$$

where

$${}^W_B R = \begin{bmatrix} \cos\alpha\cos\beta & \sin\beta & \sin\alpha\cos\beta \\ -\cos\alpha\sin\beta & \cos\beta & -\sin\alpha\sin\beta \\ -\sin\alpha & 0 & \cos\alpha \end{bmatrix} \tag{2.4}$$

In this thesis, the subscript of a vector denotes the axes in which it is expressed. Therefore, subscript B stands for body axes; subscript S for stability axes; and subscript W for wind axes.

An inertial reference frame is needed to describe the lateral position (x and y) and altitude (h) of the aircraft. In this inertial frame, the X axis points North, the Y axis points East, and the Z axis points down, thus it is often called the north-east-down (NED) frame. The NED frame is considered fixed in space with its X-Y plane tangent to the surface of the Earth.

In order to express aircraft orientation in the NED frame, Euler angles are defined by performing a sequence of frame rotations. Starting from the inertial reference frame, the first rotation is about the Z axis, which defines yaw angle (ψ). The second rotation is performed about the new Y axis, which defines pitch angle (θ), and the last rotation is performed about the new X axis incurred by the second rotation, which defines roll angle (ϕ). Therefore, the following equation transforms a vector expressed in the NED frame to the equivalent vector expressed in the body frame:

$$v_B = R_\phi R_\theta R_\psi v_I = {}^B_I R v_I \tag{2.5}$$

where the subscript *I* denotes the vector is expressed in the inertial (NED) frame and the roll, pitch, and yaw rotation matrices are given by:

$$R_\phi = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & \sin\phi \\ 0 & -\sin\phi & \cos\phi \end{bmatrix},\ R_\theta = \begin{bmatrix} \cos\theta & 0 & -\sin\theta \\ 0 & 1 & 0 \\ \sin\theta & 0 & \cos\theta \end{bmatrix},\ R_\phi = \begin{bmatrix} \cos\psi & \sin\psi & 0 \\ -\sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{2.6}$$

2.1.2 Forces and Moments

The total forces and moments at the aircraft CG have components due to aerodynamic effects and engine thrust. The total force expressed in the body frame is given by

$$F_B = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = F_{BA} + F_{BT} \tag{2.7}$$

while the total moment vector expressed in the body frame is given by

$$T_B = T_{BA} + T_{BT} \tag{2.8}$$

In both equations, the subscripts A and T denote the components due to aerodynamic effects and engine thrust, respectively.

The velocity of the CG is defined most simply in wind axes, where its only non-zero component is true airspeed V_T in the X direction. The velocity vector expressed in body axes is given by

$$v_B = \begin{bmatrix} U \\ V \\ W \end{bmatrix} = {}^B_W R v_W = {}^B_W R \begin{bmatrix} V_T \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} V_T \cos\alpha \cos\beta \\ V_T \sin\beta \\ V_T \sin\alpha \cos\beta \end{bmatrix} \tag{2.9}$$

The following expressions can then be derived:

$$\begin{aligned} \tan\alpha &= \frac{W}{U} \\ \sin\beta &= \frac{V}{V_T} \\ V_T &= (U^2 + V^2 + W^2)^{1/2} \end{aligned} \tag{2.10}$$

α, β, and V_T are then used to determine aerodynamic forces and moments. The aerodynamic forces are the drag D, the lift L, and the sideforce Y. The aerodynamic moments are the rolling moment $\bar{L}$, the pitch moment M, and the yawing moment N. These forces and moments are defined in terms of dimensionless aerodynamic coefficients:

$$
\begin{aligned}
D &= \bar{q} S C_D \\
L &= \bar{q} S C_L \\
Y &= \bar{q} S C_Y \\
\bar{L} &= \bar{q} S b C_l \\
M &= \bar{q} S \bar{c} C_M \\
N &= \bar{q} S b C_N
\end{aligned}
\tag{2.11}
$$

where

$$
\bar{q} = \frac{1}{2} \rho V_T^2 \tag{2.12}
$$

In equation (2.11), forces are specified in wind axes, while the moments are specified in body axes. $\bar{q}$, ρ, S, b, and $\bar{c}$ are, respectively, the free-stream dynamic pressure, the atmospheric density, the wing reference area, the wing span, and the wing mean geometric chord. The dimensionless aerodynamic coefficients C_D, C_L, …, C_N, are primarily functions of aerodynamic angles α and β. They are also dependent on control surface deflection, making the aircraft controllable. Because these equations are not the focus of this work, the reader is referred to a more complete description of aerodynamic forces and moments in [28, 29].

2.1.3 The Nonlinear Aircraft Model

The standard 6-DOF equations of motion typically used for a conventional aircraft assume a flat Earth, since the accelerations associated with the Earth's rotation are negligible compared to the accelerations that can be produced by a maneuvering aircraft. The state variables used to describe the equations of motion for the aircraft includes the components (*U, V, W*) of the translational velocity vector v_B, angular rates (*p, q, r*), inertial position (*x, y, z*), and ZYX Euler angles (ϕ, θ, ψ):

$$z = \begin{bmatrix} U & V & W & p & q & r & x & y & z & \phi & \theta & \psi \end{bmatrix} \tag{2.13}$$

These standard equations of motion are defined in the body-fixed coordinate frame. The force equations are

$$\begin{aligned} \dot{U} &= rV - qW - g\sin\theta + \frac{F_x}{m} \\ \dot{V} &= -rU + pW + g\sin\phi\cos\theta + \frac{F_y}{m} \\ \dot{W} &= qU - pV + g\cos\phi\cos\theta + \frac{F_z}{m} \end{aligned} \tag{2.14}$$

where m is the mass of the aircraft. The kinematic conversion between body axis velocities (p, q, r) and Euler angle first derivatives is given by

$$\begin{aligned} \dot{\phi} &= p + \tan\theta(q\sin\phi + r\cos\phi) \\ \dot{\theta} &= q\cos\phi - r\sin\phi \\ \dot{\psi} &= \sec\theta(q\sin\phi + r\cos\phi) \end{aligned} \tag{2.15}$$

The moment equations are

$$\begin{aligned} \dot{p} &= (c_1 r + c_2 p)q + c_3\overline{L} + c_4 N \\ \dot{q} &= c_5 pr - c_6(p^2 - r^2) + c_7 M \\ \dot{r} &= (c_8 p - c_2 r)q + c_4\overline{L} + c_9 N \end{aligned} \tag{2.16}$$

where constants c_i are

$$c_1 = \frac{(I_y - I_z)I_z - I_{xz}^2}{I_x I_z - I_{xz}^2} \qquad c_4 = \frac{I_{xz}}{I_x I_z - I_{xz}^2} \qquad c_7 = \frac{1}{I_y}$$

$$c_2 = \frac{(I_x - I_y + I_z)I_{xz}}{I_x I_z - I_{xz}^2} \qquad c_5 = \frac{I_z - I_x}{I_y} \qquad c_8 = \frac{I_x(I_x - I_y)}{I_x I_z - I_{xz}^2}$$

$$C_3 = \frac{I_z}{I_x I_z - I_{xz}^2} \qquad c_6 = \frac{I_{xz}}{I_y} \qquad c_9 = \frac{I_x}{I_x I_z - I_{xz}^2}$$

The aircraft inertia matrix, under the assumption that $x - z$ is a plane of symmetry, is given by

$$J = \begin{bmatrix} I_x & 0 & -I_{xz} \\ 0 & I_y & 0 \\ -I_{xz} & 0 & I_z \end{bmatrix}$$

The navigation equations represent the North, East, and vertical components of the aircraft velocity vector expressed in the locally-level NED frame on the surface of the Earth and are given by

$$\begin{aligned} \dot{x} &= U\cos\theta\cos\psi + V(-\cos\phi\sin\psi + \sin\phi\sin\theta\cos\psi) + W(\sin\phi\sin\psi + \sin\phi\sin\theta\cos\psi) \\ \dot{y} &= U\cos\theta\sin\psi + V(\cos\phi\cos\psi + \sin\phi\sin\theta\sin\psi) + W(-\sin\phi\cos\psi + \cos\phi\sin\theta\sin\psi) \\ \dot{h} &= U\sin\theta - V\sin\phi\cos\theta - W\cos\phi\cos\theta \end{aligned} \tag{2.17}$$

Usually, it is convenient to replace state variables U, V, and W in the force equations by α, β, and V_T since they are the primary factors determining aerodynamic forces and moments. Then U, V, and W can be calculated from Equation (2.9) for the force equations. The following expressions can be derived from Equations (2.9) and (2.10):

$$\begin{aligned} \dot{V}_T &= \frac{U\dot{U} + V\dot{V} + W\dot{W}}{V_T} \\ \dot{\alpha} &= \frac{U\dot{W} - W\dot{U}}{U^2 + W^2} \\ \dot{\beta} &= \frac{\dot{V}V_T - V\dot{V}_T}{V_T^2\cos\beta} \end{aligned} \tag{2.18}$$

These expressions enable conversion to the 6-DOF state vector used for this work:

$$z^T = \begin{bmatrix} V_T & \alpha & \beta & p & q & r & x & y & z & \phi & \theta & \psi \end{bmatrix} \tag{2.19}$$

During the evaluation of these equations of motion, the derivatives $\dot{\alpha}$, $\dot{\beta}$ can be computed after the forces in Equations (2.14) have been evaluated. The state

variables can be grouped into two vectors: the 6-dimensional configuration of the aircraft

$$\eta^T = \begin{bmatrix} x & y & z & \phi & \theta & \psi \end{bmatrix} \tag{2.20}$$

and the linear and angular velocities of the aircraft

$$\nu^T = \begin{bmatrix} V_T & \alpha & \beta & p & q & r \end{bmatrix} \tag{2.21}$$

2.2 The Post-Damage GTM Aircraft Model

As indicated in Chapter 1, the damaged aircraft model primarily used in this thesis is a Generalized Transport Model (GTM) aircraft with significant damage to the left wing. Specifically, the damage results in a missing wingtip. Thus for simplicity, the "left wing damage" will translates to "missing wingtip" in the later discussions. This 6-DOF nonlinear aircraft model is based on the value data of stability derivatives and control derivatives from NASA Ames. As described in Section 2.1.2, the dimensionless aerodynamic coefficients defined in equation (2.11) are functions of control surface deflections, as well as aircraft state variables. Thus, these stability and control derivatives are actually the partial derivatives determined by deriving the Taylor series of these dimensionless aerodynamic coefficients about a point in the control and state variable space. The derivatives used for developing the damaged GTM model are summarized in Table 2-1. While the second column in Table 2-1 contains the constant values of the dimensionless aerodynamic coefficients at a given fixed point, columns 3 through 8 are respectively the values of the first-order partial derivatives of the coefficients, at this given point, with respect to the body-axes

angular velocity variables p, q, and r, as well as the elevator deflection variable μ_e, aileron deflection variable μ_a, and rudder deflection variable μ_r. In the Ames' data, the values p_0, q_0, r_0, μ_{e0}, μ_{a0}, and μ_{r0} of these given points are all zeros. Therefore, the entries in Table 2-1 are all functions of α, β, and V_T.

Table 2-1: Definition of Stability and Control Derivatives

Dimensionless Aerodynamic Coefficient	Variables						
		p	q	r	μ_e	μ_a	μ_r
C_L	C_{L0}	$\frac{\partial C_L}{\partial p}$	$\frac{\partial C_L}{\partial q}$	$\frac{\partial C_L}{\partial r}$	$\frac{\partial C_L}{\partial \mu_e}$	$\frac{\partial C_L}{\partial \mu_a}$	$\frac{\partial C_L}{\partial \mu_r}$
C_D	C_{D0}	$\frac{\partial C_D}{\partial p}$	$\frac{\partial C_D}{\partial q}$	$\frac{\partial C_D}{\partial r}$	$\frac{\partial C_D}{\partial \mu_e}$	$\frac{\partial C_D}{\partial \mu_a}$	$\frac{\partial C_D}{\partial \mu_r}$
C_Y	C_{Y0}	$\frac{\partial C_Y}{\partial p}$	$\frac{\partial C_Y}{\partial q}$	$\frac{\partial C_Y}{\partial r}$	$\frac{\partial C_Y}{\partial \mu_e}$	$\frac{\partial C_Y}{\partial \mu_a}$	$\frac{\partial C_Y}{\partial \mu_r}$
C_l	C_{l0}	$\frac{\partial C_l}{\partial p}$	$\frac{\partial C_l}{\partial q}$	$\frac{\partial C_l}{\partial r}$	$\frac{\partial C_l}{\partial \mu_e}$	$\frac{\partial C_l}{\partial \mu_a}$	$\frac{\partial C_l}{\partial \mu_r}$
C_M	C_{M0}	$\frac{\partial C_M}{\partial p}$	$\frac{\partial C_M}{\partial q}$	$\frac{\partial C_M}{\partial r}$	$\frac{\partial C_M}{\partial \mu_e}$	$\frac{\partial C_M}{\partial \mu_a}$	$\frac{\partial C_M}{\partial \mu_r}$
C_N	C_{N0}	$\frac{\partial C_N}{\partial p}$	$\frac{\partial C_N}{\partial q}$	$\frac{\partial C_N}{\partial r}$	$\frac{\partial C_N}{\partial \mu_e}$	$\frac{\partial C_N}{\partial \mu_a}$	$\frac{\partial C_N}{\partial \mu_r}$

Table 2-2: Values for α_0, β_0, and constant Mach number

	α_0 (deg)	β_0 (deg)	Constant Mach Number
Max	12	10	0.8
Min	-2	-10	0.1
Δ	2	5	0.1
Total pts.	320		

However, the values of the entries in Table 2-1 are only given on a discrete set of α_0's, β_0's, and Mach numbers, as shown in Table 2-2. Thus, a trilinear interpolation

algorithm[30] is used to approximate the true values of the derivatives for the speeds ranging from Mach 0.1 to Mach 0.8, α 's from -2º to 12º, and β's ranging from -10º to 10º. The stability and control derivatives values are then used for the calculation of the dimensionless aerodynamic coefficients, specifically the C_D, C_L, C_Y, C_l, C_M, and C_N defined in equation (2.11), which are presumed continuous and smooth functions of α, β, and airspeed. Figure 2-2 shows the dimensionless aerodynamic coefficients as numerical functions of α and β at an altitude of 5,000 feet and at a speed of 600 ft/sec.

As described in Section 2.1, the aerodynamic components of the forces and moments are derived from the dimensionless aerodynamic coefficients based on equations (2.11, 2.12). These forces and motions, together with the engine thrust terms mentioned in equations (2.7, 2.8), enable propagation of aircraft motion over time by equations (2.14-2.18).

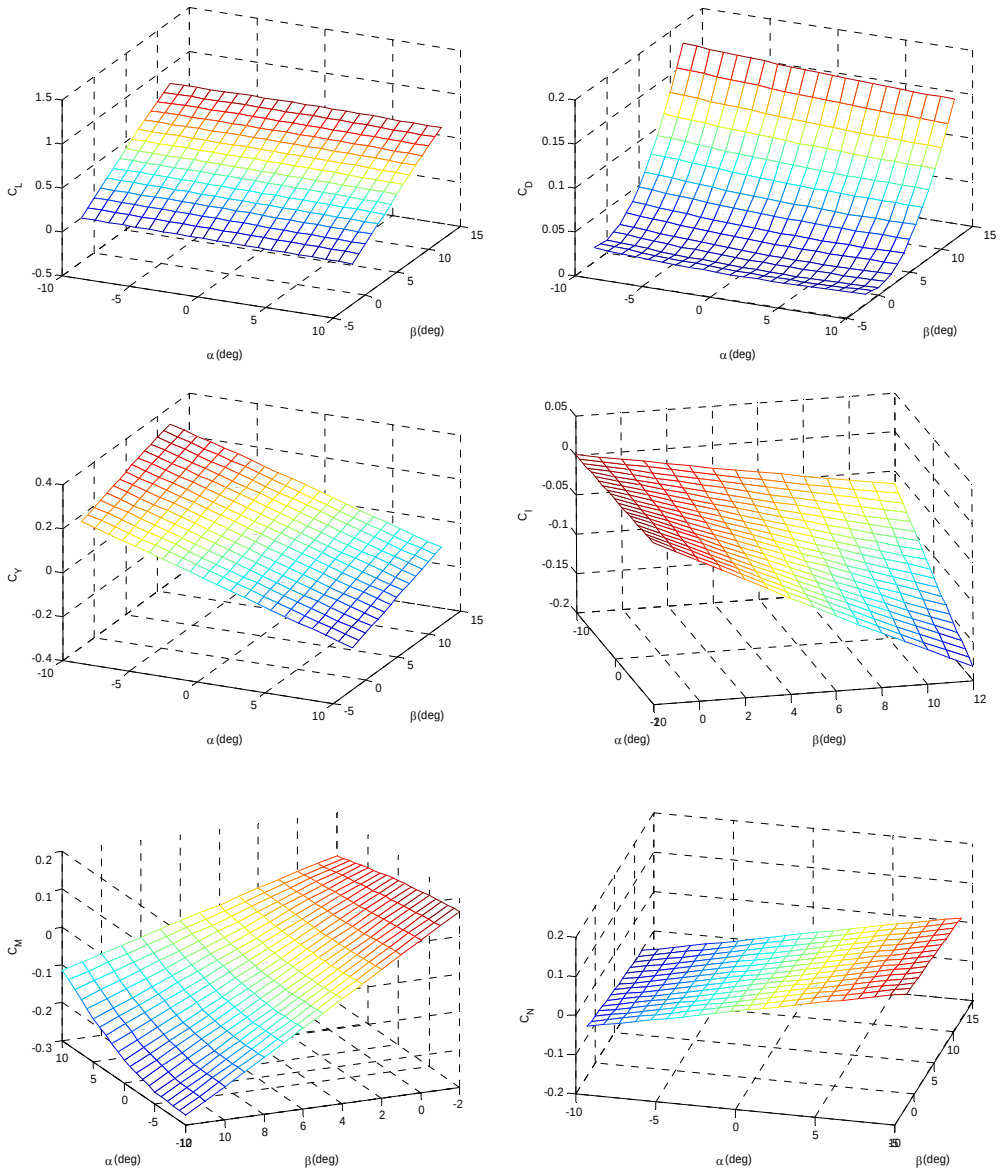

Figure 2-2: Dimensionless Aerodynamic Coefficients for the Damaged GTM aircraft

The control input vector μ for the damaged GTM includes standard elevator, aileron, and rudder deflections, as well as explicit left and right engine thrust terms representing differential throttle control:

$$\mu = \begin{bmatrix} \mu_{lt} & \mu_{rt} & \mu_e & \mu_a & \mu_r \end{bmatrix}^T \tag{2.22}$$

Full control authority is assumed for this structurally damaged aircraft model. Therefore, the deflection limits for the control surfaces μ_e, μ_a, and μ_r, the elevator, aileron, and rudder deflections, respectively, are assumed $\pm 30^\circ$. Left and right engine thrusts μ_{lt} and μ_{rt} are constrained between zero, for no thrust, and 1, for maximum thrust, which is 40,000 lbs adopted by this thesis. The controls are modeled to be ideal, neglecting the lag between issuing the command and the actual response.

2.3 Aircraft Trim State Definition

A *trim state* is the aircraft state under a trimmed flight condition. In a trim state, the aircraft travels in an equilibrium (non-accelerating) flight condition, that is, the aircraft linear and angular velocities (in body coordinates) are constant. Mathematically, the trimmed flight condition can be expressed as

$$\dot{\nu}^* = 0 \tag{2.23}$$

where the asterisk is used to denote equilibrium (trim). This general condition is necessary for all trimmed flight. Given our standard representation of aircraft state, steady trimmed flight is achieved given the following conditions:

$$\begin{aligned} &(\dot{V}_T, \quad \dot{\alpha}, \quad \dot{\beta}) \equiv 0 \\ &(\dot{p}, \quad \dot{q}, \quad \dot{r}) \equiv 0 \\ &(\dot{\phi} \quad \dot{\theta}) \equiv 0 \\ &\dot{\psi} \equiv \dot{\psi}^* \\ &\dot{h} \equiv \dot{h}^* \end{aligned} \tag{2.24}$$

where $\dot{\psi}^*$ and $\dot{h}^*$ are respectively the desired constant climb rate and constant turn rate for the trim condition. With these constraints, eight aircraft state variables can fully

define a steady climbing-turning trim state and can be grouped into a reduced state vector

$$\bar{z}^T = [V_T \quad \alpha \quad \beta \quad p \quad q \quad r \quad \phi \quad \theta] \tag{2.25}$$

By solving the aircraft equations of motion under the conditions expressed in Equation (2.24), trim state $\bar{z}^*$ and trim control settings μ^* can be found for a given velocity V_T, climb rate $\dot{h}^*$ and turn rate $\dot{\psi}^*$. $\bar{z}^*$ and μ^* also depend on aircraft altitude h since atmospheric density variation as a function of altitude requires control setting changes to maintain the trim state. Each trimmed flight state can typically be achieved over a range of airspeeds. Therefore, the trim state $\bar{z}^*$ and the corresponding trim control settings μ^* are actually functions of four parameters that define a steady climbing-turning trimmed flight condition, expressed by the quadruplet

$$(h^*, V_T^*, \dot{h}^*, \dot{\psi}^*) \tag{2.26}$$

where h^* and V_T^* are respectively the altitude and true airspeed associated with the trim state. For this work, given a non-zero climbing rate flight, h^* is presumed to be the initial altitude of that trim flight. This is a good assumption for relatively short flight segments or for descending flight segments typical for landing trajectories (since envelope typically degrades at higher altitudes), since the commanded trim state computed for the initial altitude can hold for a short period of time then can be actively maintained by varying control input settings.

Chapter 3: Landing Site Search

This chapter describes the process of selecting an appropriate emergency landing site, defined as a runway in this research given the use of an airport database. This Landing Site Search, one of the two primary components of the adaptive flight planning architecture (Figure 1-1), provides a position (approach end of the runway) and heading used as the "target" state by the trajectory planner described below. The Landing Site Search procedure and software used for this work are based on [2], which provided a complete LSS capability but that required some extension for use with the damaged GTM due to the specific focus in [2] on loss-of-thrust emergency scenarios.

Landing Site Search is a two-step process. First, a reachable footprint is generated taking into account the range constraints imposed by the aircraft damage or failure scenario. Then, all reachable runways within this footprint are examined to identify the feasible subset that can safely accommodate the emergency landing. In cases where more than one feasible runway is found, the LSS ranks the list of feasible runways based on a safety-oriented utility function [2]. The most desirable landing site (i.e., the highest-ranked feasible runway) is identified as the final state for the trajectory planner.

In this thesis, this Landing Site Search process is extended to accommodate emergencies that do not limit range, such as the wing damage situation described in

this thesis. Section 3.1 reviews the overall Landing Site Search procedure. Section 3.2 presents the modified footprint generation algorithm applicable to the GTM, while Section 3.3 reviews the feasible runway identification and ranking LSS processes.

3.1 Landing Site Search Architecture

Figure 3-1 shows the architecture of Landing Site Search module in the context of Adaptive Flight Planner. The inputs to the Landing Site Search include a U.S. airport database, the updated aircraft performance model, the initial aircraft state, and the airport wind/weather conditions. While the airport database is static, the updated aircraft performance model must be generated or identified from a database in real-time taking into account the aircraft failure/damage. The initial LSS aircraft state is specified by the instantaneous location (x, y), altitude (h), heading (ψ), and velocity (V_T) of the aircraft when the emergency occurs. Together with the runway data from the U.S. airport database, the real-time airport weather conditions are used to evaluate the feasibility and "quality" of a runway/airport to accommodate a safe emergency landing.

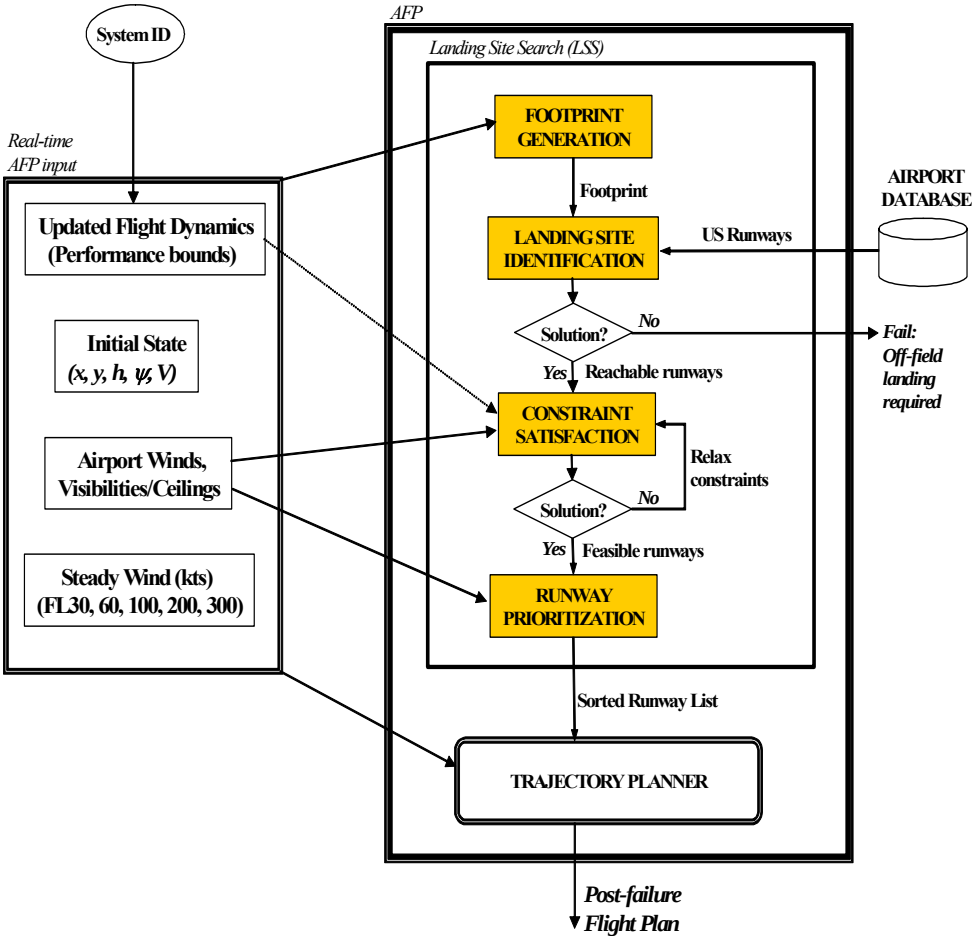

Figure 3-1: Landing Site Search Architecture [2]

The output of the Landing Site Search is a sorted runway list, which contains the candidate runways ranked according to their safety-oriented utility values. The top candidate will then be processed in by the AFP trajectory planner. At the highest level, the Landing Site Search includes: footprint generation, landing site identification, constraint satisfaction, and utility-based runway prioritization. Footprint generation calculates the approximate boundary of the region the post-failure aircraft can or should reach before it is forced to land. Landing site identification builds a list of all runways within this footprint, then constraint

satisfaction rules out the runways that cannot meet hard safety constraints such as minimum runway length or surface (e.g., asphalt vs. water). Particularly near urban areas, multiple feasible runways can be reached. The final step is to evaluate the feasible runways [2] in terms of their desirability ranging from safety-oriented factors (e.g., extra runway length, width, instrument procedures) to company preferences (e.g., availability of maintenance facilities). The final output is then the sorted runway list. Below, the footprint generation process is described, focusing on augmentations made in this work appropriate for situations such as the studied GTM damage case where range is limited by time aloft rather than physical limitations of the aircraft. The final section reviews the remaining LSS components described in [2] to facilitate understanding of the results presented in this thesis.

3.2 Footprint Generation

For the originally-studied loss-of-thrust emergency [2], aircraft *footprint* had the standard definition of "maximum region the aircraft can reach on the ground". As will be seen below, for the studied left wing damage scenario the aircraft can remain aloft and fly straight for an extended time. Thus, with its standard definition the "reachable" footprint region may extend for thousands of miles given sufficient fuel. However, as indicated in [2], given a failure or damage that renders flight difficult and that may potentially be progressive, "footprint" may also be artificially defined to constrain time aloft. Such an algorithm was posited but not implemented with work from [2], thus is developed as part of this thesis. Below, the loss-of-thrust algorithm is reviewed, followed by a description of the new artificial footprint generation algorithm.

3.2.1 Footprint Generation for Loss-of-Thrust

When aircraft thrust is totally lost during the flight, the un-powered aircraft cannot maintain altitude over time. In this situation, there is a hard constraint on time aloft, so an approximate maximum-range footprint is computed [2].

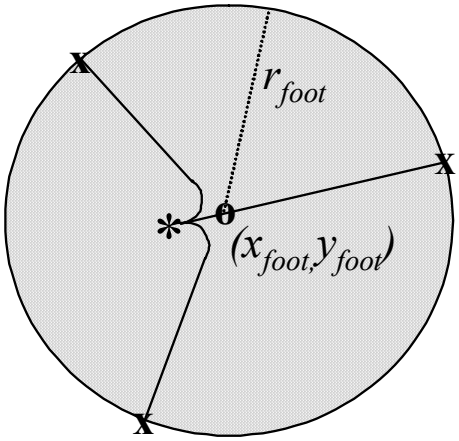

Figure 3-2: Footprint Generation for Loss-of-Thrust Emergency [2]

The approximation procedure is illustrated in Figure 3-2. The asterisk (*) marks the 2-D location (x, y) at which the aircraft loses its thrust with an initial East-Northeast heading. To explore the region in which the aircraft can glide given the initial state, three boundary points, marked as **x** in the figure, are computed. The location of the first boundary point is determined by projecting the aircraft straight ahead from its initial state to the ground along a best-glide path. The locations of the second and third boundary points are computed by turning the aircraft 120° to the right and left, respectively, presuming best-glide, 30° bank turns, then projecting straight best-glide paths to the ground. These three boundary points define a circular approximate footprint with radius r_{foot} and center location (x_{foot}, y_{foot}). Since strong winds have an

appreciable impact on a glider's flight trajectory, wind corrections that shift these three boundary points are included in this procedure.

3.2.2 Footprint Generation for Aircraft that Maintain Extended Range Capability

In damage or failure situations where the aircraft can remain aloft until fuel is exhausted, the practical requirement to safely land the disabled aircraft near-term rather than maximum range limits footprint size. For example, an F-16 aircraft with a 15º rudder jam failure can still perform straight, left, and right turning flight, as well as climbing and descending flight [3]. In such cases there is no hard range constraint beyond fuel use considerations. Furthermore, it is more difficult to describe the geometric characteristics of the footprint for such a disabled aircraft, since the aircraft has sufficient maneuver choices to make it capable of reaching many locations.

To solve this problem, a virtual footprint boundary is defined to artificially constraint the reachable region. For simplicity and compatibility with the loss-of-thrust footprint, a circular boundary is defined for this virtual aircraft footprint as well. The center of the circle is defined at the initial location at which the emergency occurs. The radius of this circle is initially set to be a "reasonable" user-defined constant (for example, 20 nautical miles for this work) which is then incremented until a feasible runway is found, or an upper bound value is reached, whichever comes first. The aircraft footprint is then defined as the region within this circular boundary. This procedure is shown in Figure 3-3.

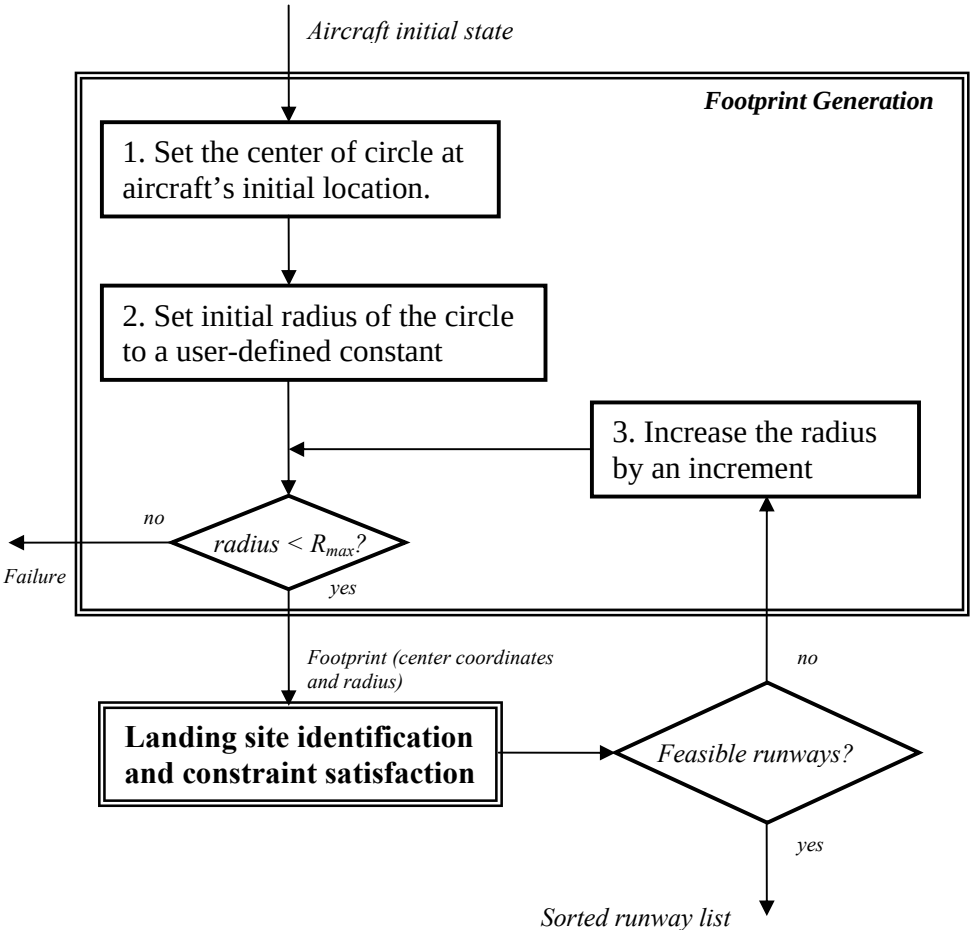

Figure 3-3: Footprint Generation for Damaged Aircraft to Limit Time Aloft.

This simple procedure was implemented within the LSS software and is shown to be effective below and in Chapter 6.

3.3 Feasible Runway Identification and Ranking

Once the post-failure/damage footprint has been defined, the set of feasible runways within this footprint are compiled and ranked. This section reviews the procedures from [2] to accomplish this goal. The reachable runways, defined as all runways within the defined circular footprint region, are identified from the U.S. airport database, based on the distance from the airport runway to the center of the circular

footprint. A runway is considered *reachable* if this distance is less than the radius of the aircraft footprint. Among these reachable runways, a set of hard constraints are checked against wind/weather conditions, as well as the published runway characteristics. The following hard constraints must be met:

- Minimum runway length
- Minimum runway width
- Maximum crosswind component
- Runway surface type
- Reported visibility vs. instrument approach minimum constraints

Those reachable runways that cannot meet one or more of these constraints are eliminated from the set, with the remaining runways defined as the feasible runways. There is a probability that all the reachable runways may be considered unqualified by this procedure. In this situation, the constraints must be relaxed, reducing safety margins until at least one feasible runway is identified. Once one or more feasible runways are identified, utility-based prioritization is used to rank these runways.

The utility function used for runway ranking is safety-oriented. This function takes into account the airport runway and wind conditions and is defined as a weighted sum:

$$U = \sum_i C_i \cdot w_i = C_1 \cdot \frac{r_l}{r_{l,\max}} + C_2 \cdot \frac{r_w}{r_{w,\max}} + C_3 \cdot q_I + C_4 \cdot \left(\frac{d}{d_{\max}} \right) + C_5 \cdot \frac{w_h}{w_{h,\max}} + C_6 \cdot \frac{\left(w_{c,\max} - w_c \right)}{\left(w_{c,\max} - w_{c,\min} \right)} + C_7 \cdot q_S + C_8 \cdot q_f \tag{3.1}$$

The utility is defined by evaluating the following eight parameters of a feasible runway:

- Runway length r_l
- runway width r_w
- Instrument approach quality q_I
- Distance d from the footprint boundary
- Headwind velocity w_h
- Crosswind velocity w_c
- Surface quality q_S
- Facility availability measure q_f

Numerical values $r_{l,max}$, $r_{w,max}$, d, $w_{h,max}$, and $w_{c,max}$ are, respectively, the extrema of parameters r_l, r_w, d, w_h, and w_c over all the feasible landing runways. The numerical costs of these parameters are normalized by these extreme values to guarantee individual cost values in the range [0.0 1.0]. Table 3-1 [4] includes a set of normalized values of quality measures q_I, q_S, and q_f. The facility availability measure q_f is a quality measure that favors landing sites where the aircraft can be repaired. Distance parameter d gives priority to runways well within the footprint boundary. The other parameters in the equation favor larger safety margins during landing.

Table 3-1: Quality Measures for Runway Utility Computation [2]

Instrument approach (q_I)		***Runway Surface (q_S)***		***Airport Facilities (q_f)***	
Description (max value used)	***Value***	***Description (one will apply)***	***Value***	***Description (Σ over all)***	***Value***
WAAS, ILS/MLS	1.0	Asphalt	1.0	Fuel of required type (Jet-A)	0.25
LOC with RWY designation	0.95	Concrete	1.0	Airframe maintenance: Major	0.25
LOC w/o RWY designation	0.85	Metal, brick, etc. (VTOL)	0.5	*Minor*	*0.125*
LDA w/ RWY designation	0.8	Wood	0.2	Power plant maintains.: Major	0.25
LDA w/o RWY designation	0.7	Turf/gravel/dirt	0.1	*Minor*	*0.125*
GPS,LORAN,RNAV w/ RWY	0.6			Bulk oxygen	0.25
VOR, NDB, SDF w/ RWY	0.5				
GPS, LORAN, RNAV, VOR, NDB, SDF w/o RWY	0.2				

While the normalized numerical costs of the function parameters for a runway are relatively fixed given a specific footprint, the weighting factors C_i (i = 1, 2, …, 8) can be set to different values in response to different emergency types, with the total sum constrained to be 1.0. The default weight set used for loss-of-thrust case study is

$$\{C_1, \quad C_2, \quad ..., \quad C_8\} = \{0.15 \quad 0.15 \quad 0.15 \quad 0.15 \quad 0.1 \quad 0.1 \quad 0.1 \quad 0.1\} \tag{3.2}$$

which gives the same preference to runway length, runway width, instrument approach quality, and runway distance to the footprint boundary, slightly more than to other parameters. Similarly to [4], the weights and perhaps even the utility function terms are subject to modification by the pilot and/or airline community. As will be

seen for our damaged GTM case, however, landings will be fast and with minimal control, practically suggesting a weight set that favors the long and wide runways:

$$\{C_1, \quad C_2, \quad ..., \quad C_8\} = \{0.4 \quad 0.4 \quad 0.05 \quad 0.01 \quad 0.05 \quad 0.03 \quad 0.03 \quad 0.03\} \tag{3.3}$$

As indicated by the values of the weighing factors, the runway length and secondarily runway width are prioritized well above the other parameters. However, non-zero weights still exist for the other parameters to distinguish runways of near-equivalent length and width.

To illustrate the effect of the different weight sets, the default weight set in equation (3.2) and the modified weight set in equation (3.3) were tested on the same set of feasible airport runways within the same footprint. In this emergency scenario, the aircraft is over the San Francisco Bay area when the left wingtip is damaged. The aircraft is initially located at (37.44^{o} North, 122.12^{o} West) with initial altitude of 10,000 ft. Footprint generation defines a footprint with radius of 100 nautical miles, which is a user-defined initial value. Within this region, 378 reachable runways are found out, out of which 18 feasible runways are identified. Table 3-2 is the ranked top ten runway list based on the default weight setting, while Table 3-3 contains the top ten results from the equation (3.3) weight set.

Table 3-2: Sorted Runway List (Top 10) with Default Weight Set

Ranking	Airport	Runway	Length (ft)	Width (ft)
1	SFO	28R	11870	200
2	SFO	28L	10600	200
3	SJC	20L	10200	150
4	OAK	29	10000	150
5	OAK	27R	5453	150
6	HWD	28L	5024	150
7	SJC	12R	10200	150
8	SCK	29R	8650	150
9	MRY	28L	7598	150
10	LVK	25R	5255	100

Table 3-3: Sorted Runway List (Top 10) with Weight Set Favoring Length and Width

Ranking	Airport	Runway	Length (ft)	Width (ft)
1	MHR	22L	11301	300
2	MHR	4R	11301	300
3	SUU	21R	11000	300
4	SUU	21L	10995	300
5	SUU	3L	11000	300
6	SUU	3R	10995	300
7	SFO	28R	11870	200
8	SFO	28L	10600	200
9	SFO	10L	11870	200
10	SFO	10R	10600	150

While the most desirable landing runway (runway 28R at SFO) in Table 3-3 has runway length of 11,870 ft and width of 200 ft, the top landing runway (runway 22L at MHR) in Table 3-4 is 11,301 ft in length and 300 ft in width. This example shows that the selection of runway utility weight set can significantly affect the ranking of feasible runways. While the default weight set performs a comprehensive evaluation on the feasible runways, the weight set in equation (3.3) favors the longer and wider runways.

Chapter 4: Trajectory Specification as Trim State Sequences

Once a landing runway is identified, the adaptive flight planner must compute a trajectory to this runway. This chapter describes a procedure originally developed in [3] by which trajectories are defined as sequences of trimmed flight segments connected by transitions between these segments, a design believed to maximize intuitive comprehension by pilots and air traffic controllers. In order to determine the feasible trim states the post-failure/damage aircraft can achieve, a discrete flight envelope is computed for the disabled aircraft. Recall that a trim state is defined as a non-accelerating flight condition that can be maintained indefinitely. Once trim and transition information is compiled, a simplified aircraft kinematic model is used to compute position and heading change incurred during trimmed flight over a finite time period. The emergency path planner sequences trim states and adjusts their duration to accurately guide the aircraft to the designated landing runway. To manage trim state transitions and reject disturbances during trimmed flight, a closed-loop PID controller is developed to reduce transition settling time and provide close tracking of a desired flight trajectory. Section 4.1 presents the analysis required to characterize the set of achievable post-failure/damage trim states, while Section 4.2 describes the transition analysis, as well as the transition controller design. Section 4.3 presents the kinematic representation used to describe trajectories as trim state sequences.

4.1 Trim Analysis

In this section, based on the definition of aircraft trim state presented in Chapter 2, a constrained nonlinear optimization technique is described for the computation of the trim states under given trimmed flight conditions. After the trim states are identified, the aircraft stability and controllability are examined in small neighborhoods surrounding a trim state. The results from this general procedure originally described in [26] and motivated by Frazzaoli [27] enable definition of an aircraft trim database that discretely represents the aircraft flight envelope.

4.1.1 Trim State Computation

Based on the definition of aircraft trim state presented in Chapter 2, a nonlinear constrained optimization technique is used to compute the trim state $\bar{z}^*$ and control settings under given flight conditions by minimizing the cost function

$$J_{trim}(z,\mu) = \frac{1}{2}\dot{\bar{z}}^T Q \dot{\bar{z}} \tag{4.1}$$

constrained by trimmed flight conditions

$$\begin{aligned} h &= h^* \\ V_T &= V_T^* \\ \tan\theta &= \frac{ab + \sin\gamma^* \sqrt{a^2 - \sin^2\gamma^* + b^2}}{a^2 - \sin^2\gamma^*} \\ p &= -\dot{\psi}^* \sin\theta \\ q &= \dot{\psi}^* \cos\theta \sin\phi \\ r &= \dot{\psi}^* \cos\theta \cos\phi \end{aligned} \tag{4.2}$$

where the trimmed flight path angle γ^* is subject to $\sin\gamma^* = h^* / V_T^*$, and

$$a = \cos\alpha\cos\beta$$
$$b = \sin\phi\sin\beta + \cos\phi\sin\alpha\cos\beta \tag{4.3}$$

The first two constraints in (4.2) directly constrain the aircraft altitude and airspeed. The third constraint indirectly specifies the desired climb rate, while the last three constraints indirectly specify the desired turn rate, as well as constrain the roll and pitch rates to be zero. Under the above constraints, the minimization of (4.1) over the aircraft state variables and the control inputs yields

$$J^*_{trim} = J_{trim}(z^*, \mu^*) = \min_{z,\mu}\{J_{trim}(z,\mu)\} \tag{4.4}$$

where z^* and μ^* are the solution to the minimization.

z^* is then a trim state if

$$J^*_{trim} = 0$$

Practically, a multi-dimensional numerical optimization algorithm is used to compute an approximate solution since this problem cannot be solved analytically. The algorithm iteratively varies the values of a set of independent variables until

$$J_{trim} < \varepsilon << 1 \tag{4.5}$$

where ε is small positive scalar. For the GTM aircraft, these independent variables include:

- Angle of attack α
- Sideslip angle β
- Roll angle ϕ
- Left and right thrust μ_{lt} and μ_{rt}
- Elevator deflection μ_e

- Aileron deflection μ_a
- rudder deflection μ_r

4.1.2 Aircraft Trim Database

The above procedure can be applied over a comprehensive set of trim conditions to define a full aircraft trim database. An *aircraft trim database* is a set of steady climbing-turning trimmed flight conditions and is a discrete representation of the continuous aircraft flight envelope that fully defines the flight dynamics characteristics. In order to practically compute the trim database for an aircraft, additional constraints are required, as described below.

- Control saturation constraints

A trim state z^*, solved by the procedure in Section 4.1.1, is only considered feasible if the corresponding control settings μ^* satisfy the constraints on the control input. For the GTM aircraft, these constraints are:

$$\begin{aligned} &|\mu_{lt} - 0.5| \le 0.5 \\ &|\mu_{rt} - 0.5| \le 0.5 \\ &|\mu_e| \le 30^\circ \\ &|\mu_a| \le 30^\circ \\ &|\mu_r| \le 30^\circ \end{aligned} \tag{4.6}$$

The first two constraints limit the left and right engine thrust (%) to a range [0 1], while the other constraints limit the control surfaces deflections between ±30º. A steady climbing-turning trimmed flight condition $(h^*, V_T^*, \dot{h}^*, \dot{\psi}^*)$ is considered feasible

if the trim state z^* is feasible under this flight condition. The aircraft flight envelope can then be defined as the complete set of feasible trimmed flight conditions $(h^*, V_T^*, \dot{h}^*, \dot{\psi}^*)$ and therefore can be represented by a volume in a four-dimensional flight condition space. The trajectory planner (described in Chapter 5) uses this trim database to generate the feasible landing trajectories that completely fall within the post-failure/damage flight envelope.

- Stability and controllability analysis

In order to further categorize these feasible trimmed flight conditions, the stability and controllability of the aircraft within a small neighborhood surrounding each corresponding trim state is assessed.

A system is *stable* at a trim state if it naturally converges to the trim state in the neighborhood of that trim state [31]. A system is considered *controllable* if there exists an input that can transfer the system between two arbitrary states in finite time [32]. Since the aircraft is a nonlinear system, a full nonlinear system analysis would be the best for characterizing aircraft dynamics at these trim states. However, several reasons justify the use of linear system theory when performing the stability and controllability analysis for the damaged GTM aircraft. First, the damaged GTM aircraft model is based on discrete aerodynamic data in tabular form, with no analytical model available. Second, a complete nonlinear analysis is very complicated, sometimes impossible, for the majority of nonlinear systems. Last, a

nonlinear system can be approximated by a linearization of its dynamics about an equilibrium point, in a small neighborhood surrounding that equilibrium point.

To perform a linear stability and controllability analysis for the aircraft system requires first the linearization of partial nonlinear aircraft dynamics represented as

$$\dot{\bar{z}} = \bar{f}(\bar{z}, \mu) \tag{4.7}$$

about a trim state $\bar{z}_k^*$. The linear perturbation model about that trim state is described by

$$\dot{x}_k = A_k x_k + B_k u_k \tag{4.8}$$

where

$$x_k = \bar{z} - \bar{z}_k^*$$

$$u_k = \mu - \mu_k^*$$

$$A_k = \left.\frac{\partial \bar{f}}{\partial \bar{z}}\right|_{\bar{z}=\bar{z}_k^*, \mu=\mu_k^*}$$

$$B_k = \left.\frac{\partial \bar{f}}{\partial \mu}\right|_{\bar{z}=\bar{z}_k^*, \mu=\mu_k^*}$$

In these equations, $\bar{f}$ is the system of nonlinear equations defining the dynamics of state $\bar{z}$ at trim altitude h_k^*, and A_k and B_k are the Jacobian matrices at the trim state. An analytic derivation of the Jacobian matrices is infeasible due to the use of tabular aerodynamic data. Instead, these two matrices can be approximated by deriving a set of the first-order differences:

$$A_{k,i} = \frac{\bar{f}(\bar{z}_k^* + \varepsilon e_i, \mu_k^*) - \bar{f}(\bar{z}_k^*, \mu_k^*)}{\varepsilon} \tag{4.9}$$

$$B_{k,i} = \frac{\bar{f}(\bar{z}_k^*, \mu_k^* + \varepsilon e_i) - \bar{f}(\bar{z}_k^*, \mu_k^*)}{\varepsilon} \tag{4.10}$$

where $A_{k,i}$ is the i^{th} column of A_k and $B_{k,i}$ is the i^{th} column of B_k, ε is a small, positive number (for example, 1×10^{-6}), and e^i is the i^{th} column of an n-dimensional identity matrix where n is the size of the corresponding $\bar{z}_k^*$ or μ_k^*.

Once the linear perturbation model is available, the results from linear control theory can be then used to analyze the system's stability and controllability about the trim state. The aircraft is considered *stable* in a small neighborhood of a trim state if all eigenvalues of the linear perturbation system (4.8) fall within the left-half complex plane, that is, the real parts of the system eigenvalues are strictly negative numbers, which can be represented as

$$\mathbf{R}\{\lambda_i(A_k)\} < 0 \qquad i = 1, ..., 8 \tag{4.11}$$

where $\lambda_i(A_k)$ is the i^{th} eigenvalue of A_k.

While stable trim states are preferred due to the fact that small perturbations about these trim states naturally decay asymptotically to zero, unstable trim states should be avoided if they are not *stabilizable*. A linear system (4.8) is *stabilizable* if there is a controller

$$u_k = -K_k x_k \tag{4.12}$$

that can make the closed-loop dynamics of (4.8) stable. This condition can be represented by

$$\mathbf{R}\{\lambda_i(A_k - B_k K_k)\} < 0 \qquad i = 1, ..., 8 \tag{4.13}$$

While an unstable trim state is acceptable if it is stabilizable, it is more favorable if it is *controllable*. The aircraft is considered controllable in a small neighborhood of a trim state z_k^* if the linear perturbation system (4.8) about this trim state has a controllable matrix pair $[\mathbf{A}_k, \quad \mathbf{B}_k]$, or equivalently, has a controllability matrix

$$\mathbf{U}_C = \begin{bmatrix} \mathbf{B}_k & \mathbf{A}_k\mathbf{B}_k & \mathbf{A}_k^2\mathbf{B}_k & \ldots & \mathbf{A}_k^{n-1}\mathbf{B}_k \end{bmatrix} \tag{4.14}$$

with full row rank *n*. The closed-loop eigenvalues of the linear perturbation dynamics (4.8) can be assigned arbitrarily with a linear controller (4.12) if the system (4.8) is controllable. Therefore, the controllability is a more dominant concern than stability and stabilizability, since a controllable trim state can be maintained despite disturbances given a capable closed-loop control law.

- Trim database representation

Based on the above procedures, the aircraft trim database can now be generated by characterizing each trim state in the four-dimensional ($h, V_T, \dot{h}, \dot{\psi}$) space in terms of feasibility (based on the nonlinear optimization), as well as stability and controllability.

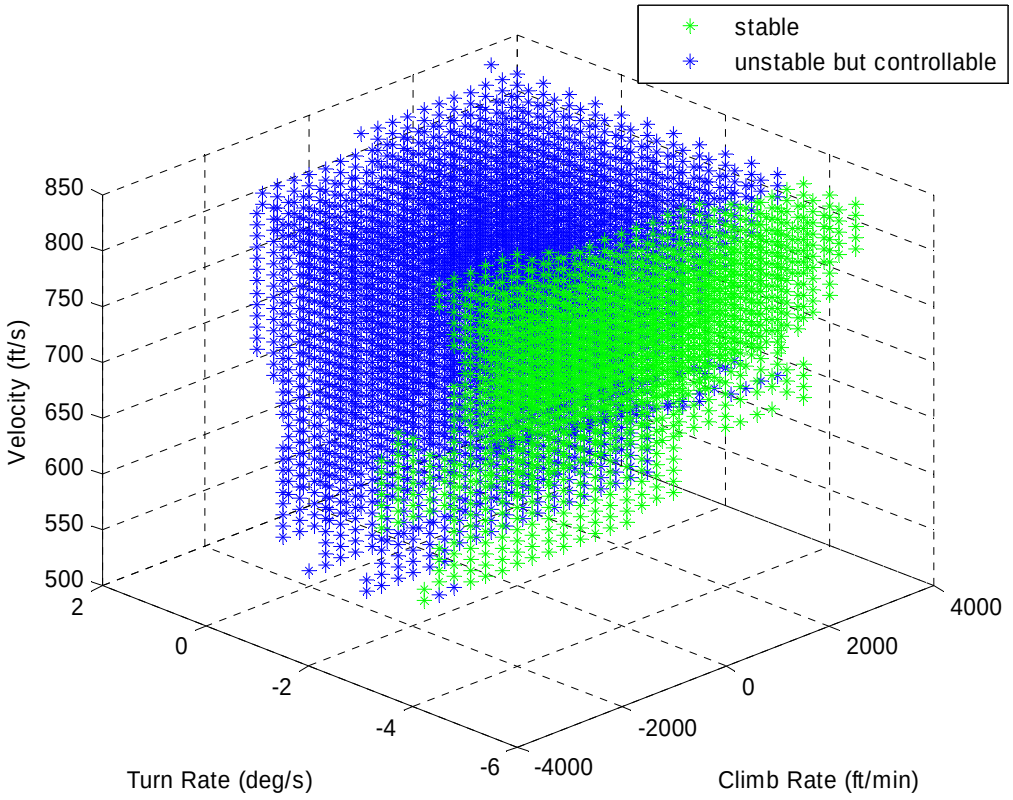

Figure 4-1: Flight Envelope for the Damaged GTM aircraft at altitude of 10 ft

By fixing one component of the trim state quadruplet, in this case altitude h^*, as well as plotting each type of trim state with a different colors in the three-dimensional space, a slice of the trim database can be presented graphically, which enables the intuitive comprehension of the viewers. Figure 4-1 shows of a slice of trim database at a fixed altitude of 10 ft for the left-wind-damaged GTM aircraft. In this figure, a green asterisk in $(V_T, \dot{h}, \dot{\psi})$ space indicates a naturally stable trim state for that trimmed flight condition quadruplet, while a blue dot represents an unstable, but controllable trim state. The unmarked area within the Figure 4-1 plot indicates the area of infeasible or uncontrollable combinations $(V_T, \dot{h}, \dot{\psi})$.

4.1.3 Trim Control

As discussed above, trimmed flight conditions define not only the trim aircraft state, but also the trim control settings. Therefore, a basic open-loop controller

$$C(h^*, V_T^*, \dot{h}^*, \dot{\psi}^*) = \mu^* \tag{4.15}$$

can be used to maintain the desired trimmed flight condition for stable trim states. However, an open-loop control strategy cannot maintain trimmed flight for the naturally unstable but controllable trim states. This problem is solved with an LQR-based PID controller that tracks the entire trajectory, including both trimmed flight segments and trim transition maneuvers. Section 4.2 presents the controller design.

4.1.4 Trimmed Flight Path Changes

In this work a trajectory is defined as a sequence of trim states. The trajectory planner searches over discrete sequences of trim states ($V_T, \dot{h}, \dot{\psi}$) and adjusts their durations to guide the aircraft to the landing site. In general, aircraft flight path change ($\Delta x, \Delta y, \Delta h, \Delta\psi$) over a trim state can be computed directly by integrating the linear velocity with respect to inertial space and yaw rate over the prescribed time interval. An intermediate velocity vector was defined to represent the aircraft linear velocity with respect to the pseudo-body axes as in [3]. The pseudo-body velocity of the aircraft that performs trimmed flight $(h^*, V_T^*, \dot{h}^*, \dot{\psi}^*)$ is computed as

$$v_P^* = R_{\theta^*}^T R_{\phi^*}^T R_{\beta^*}^T R_{\alpha^*}^T \begin{bmatrix} V_T^* & 0 & 0 \end{bmatrix}^T \tag{4.16}$$

which is constant since α^*, β^*, ϕ^*, θ^*, and V_T^* are all constant if the trimmed climb rate is zero. With non-zero climb rate, atmospheric density changes as a function of altitude result in variation of trim states $\bar{z}^*$ required to maintain a trim flight condition

involving non-zero climb rate. Therefore, the pseudo-body velocity varies along climbing trim trajectories, which makes the computation of flight path changes more complicated. However, empirical research has shown that the pseudo-body velocity varies roughly linearly with altitude, that is, dv_P / dh is roughly constant over a trim segment [3]. As a result, with non-zero climb rate the pseudo-body velocity of the aircraft during a trim segment can be expressed as a function of time:

$$v_P^*(t) = v_{P,i}^* + \frac{v_{P,i+1}^* - v_{P,i}^*}{t_{i+1} - t_i}(t - t_i) \tag{4.17}$$

where $v_{P,i+1}^*$ and $v_{P,i}^*$ are respectively the aircraft pseudo-body velocity at the beginning and end of a trim flight segment.

Based on the general method for computing flight path changes, as well as the definition of pseudo-body velocity, the yaw change during a steady climbing-turning trimmed flight is directly given by

$$\Delta\psi = \dot{\psi}^*(t_1 - t_0) \tag{4.18}$$

where the trim state is maintained from initial time t_0 to final time t_1. The position change over a trim flight segment is determined by

$$\begin{aligned} \Delta p &= \int_{t_0}^{t_1} v_I(t)dt \\ &= \int_{t_0}^{t_1} R_\psi^T v_P(t)dt \\ &= R_{\psi_0}^T(\Omega_1 v_{P,0}^* + \Omega_2 \frac{v_{P,1}^* - v_{P,0}^*}{t_1 - t_0}) \end{aligned} \tag{4.19}$$

where $R_{\psi_0} = R_\psi(t_0)$, I is the 3×3 identity matrix, and

$$\Omega_1 = \begin{cases} \dfrac{1}{\dot{\psi}^*} \begin{bmatrix} \sin\Delta\psi & \cos\Delta\psi - 1 & 0 \\ 1 - \cos\Delta\psi & \sin\Delta\psi & 0 \\ 0 & 0 & \Delta\psi \end{bmatrix}, & \text{if } \dot{\psi}^* \neq 0 \\ (t_1 - t_0) I, & \text{if } \dot{\psi}^* = 0 \end{cases} \tag{4.20}$$

$$\Omega_2 = \begin{cases} \dfrac{1}{\dot{\psi}^{*2}} \begin{bmatrix} \cos\Delta\psi + \Delta\psi\sin\Delta\psi - 1 & -\sin\Delta\psi + \Delta\psi\cos\Delta\psi & 0 \\ \sin\Delta\psi - \Delta\psi\cos\Delta\psi & \cos\Delta\psi + \Delta\psi\sin\Delta\psi - 1 & 0 \\ 0 & 0 & \Delta\psi^2 \end{bmatrix}, & \text{if } \dot{\psi}^* \neq 0 \\ \dfrac{(t_1 - t_0)^2}{2} I, & \text{if } \dot{\psi}^* = 0 \end{cases} \tag{4.21}$$

The above analytic expressions enable rapid computation of aircraft position and heading changes during steady climbing-turning trimmed flight thus are practical for use within the iterative trajectory generation algorithm.

4.2 Transition Analysis

In addition to trimmed flight segments, an emergency landing trajectory also requires accurate characterization of the transition maneuvers that connect neighboring trim segments. The post-failure/damage aircraft must be able to stably perform these transition maneuvers, ideally with smooth and rapid response characteristics. This section presents a maneuver strategy that smoothes transitions between two trim states, as well as the development of an LQR-based PID closed-loop controller that provides the left-wing-damaged GTM aircraft with configuration tracking capability.

4.2.1 Trim transitions

A trim transition is defined as a finite time evolution between two trim states. A transition from trim state *i* into trim state *j* is defined intuitively as the change in flight conditions

$$(V_{T,i}^{*},\ \dot{h}_{i}^{*},\ \dot{\psi}_{i}^{*}) \rightarrow\ (V_{T,j}^{*},\ \dot{h}_{j}^{*},\ \dot{\psi}_{j}^{*}) \tag{4.22}$$

over a finite transition time Δt, where $(V_{T,i}^{*},\dot{h}_{i}^{*},\dot{\psi}_{i}^{*})$ denotes the initial trim flight condition and $(V_{T,j}^{*},\dot{h}_{j}^{*},\dot{\psi}_{j}^{*})$ represents the terminal trim state. Correspondingly, the transition leads to a change in aircraft state

$$\bar{z}_{i}^{*}(h_{i},V_{T,i}^{*},\dot{h}_{i}^{*},\dot{\psi}_{i}^{*}) \rightarrow \bar{z}_{j}^{*}(h_{j},V_{T,j}^{*},\dot{h}_{j}^{*},\dot{\psi}_{j}^{*}) \tag{4.23}$$

where h_i and h_j are respectively the initial and resulting terminal aircraft altitude. Note that the trim state variables are dependent on aircraft altitude due to the variation of atmospheric density as a function of altitude. Similarly, the transition also results in the change in open-loop control settings

$$\mu_{i}^{*}(h_{i},V_{T,i}^{*},\dot{h}_{i}^{*},\dot{\psi}_{i}^{*}) \rightarrow \mu_{j}^{*}(h_{j},V_{T,j}^{*},\dot{h}_{j}^{*},\dot{\psi}_{j}^{*}) \tag{4.24}$$

4.2.2 Interpolated Transitions

While many approaches can be used to define the desired instantaneous flight condition and control settings during the transition, an intuitive strategy is to smoothly vary the flight condition, as well as the open-loop control settings, over transition interval Δt. With this strategy, the desired flight condition during a transition $(V_{T,i}^{*},\ \dot{h}_{i}^{*},\ \dot{\psi}_{i}^{*}) \rightarrow\ (V_{T,j}^{*},\ \dot{h}_{j}^{*},\ \dot{\psi}_{j}^{*})$ is given by

$$
\begin{aligned}
V_T^*(t) &= V_{T,i}^* + \frac{V_{T,j}^* - V_{T,i}^*}{\Delta t}(t - t_0) \\
\dot{h}^*(t) &= \dot{h}_i^* + \frac{\dot{h}_j^* - \dot{h}_i^*}{\Delta t}(t - t_0) \qquad\qquad \forall t \in [t_0,\ \ t_0 + \Delta t] \\
\dot{\psi}^*(t) &= \dot{\psi}_i^* + \frac{\dot{\psi}_j^* - \dot{\psi}_i^*}{\Delta t}(t - t_0)
\end{aligned}
\tag{4.25}
$$

where t_0 denotes the beginning time of the transition. Using the same linear interpolation strategy, the desired control settings and the desired aircraft state at time t during the transition $(V_{T,i}^*, \dot{h}_i^*, \dot{\psi}_i^*) \rightarrow (V_{T,j}^*, \dot{h}_j^*, \dot{\psi}_j^*)$ is given by

$$
\begin{aligned}
\mu^*(t) &= \mu_i^* + \frac{\mu_j^* - \mu_i^*}{\Delta t}(t - t_0) \qquad\qquad \forall t \in [t_0,\ \ t_0 + \Delta t] \\
\bar{z}^*(t) &= \bar{z}_i^* + \frac{\bar{z}_j^* - \bar{z}_i^*}{\Delta t}(t - t_0)
\end{aligned}
\tag{4.26}
$$

where the terminal trim state and trimmed control settings can be found by estimating the terminal altitude

$$
h_j = h_i + \frac{\dot{h}_j^* + \dot{h}_i^*}{2}\Delta t \tag{4.27}
$$

Equations (4.25) and (4.26) define a desired reference trajectory for the transition $(V_{T,i}^*, \dot{h}_i^*, \dot{\psi}_i^*) \rightarrow (V_{T,j}^*, \dot{h}_j^*, \dot{\psi}_j^*)$. However, open-loop control strategies cannot provide good tracking performance [3]. For the damaged GTM aircraft, an open-loop controller cannot even maintain stability during transitions involving naturally unstable but controllable trim states, which are used due to sparseness of naturally stable trim states. As a result, a closed-loop control strategy is required to stability such transitions and to characterize the nature of the closed-loop transition time response for our trajectory planner.

4.2.3 LQR-based PID Controller

A closed-loop controller for commanding the trim transitions is required to guarantee system stability, as well as to provide good configuration tracking ability. A nonlinear PID controller is used to meet these requirements, adapted from [3], for the GTM model. The development of this controller is described as follows.

- Configuration tracking control

As discussed in Section 4.1, the linear perturbation model of the aircraft about a trim state $\bar{z}_k^*$ is given by

$$\dot{x}_k = A_k x_k + B_k u_k \tag{4.28}$$

where the state vector $x_k = \bar{z} - \bar{z}^*$ represents the error between the actual aircraft state and the desired trim state. As an augmentation of open-loop control, a full state feedback control law $u_k = -K_k x_k$ can achieve the commanded trim state provided that the trim state is controllable. However, it is more favorable if the controller can also guarantee minimization of the configuration tracking error. One approach is to include the integral of the flight condition tracking error [3]. The flight condition tracking error is given by

$$y_k = \begin{bmatrix} V_T - V_T^* \\ \dot{h} - \dot{h}^* \\ \dot{\psi} - \dot{\psi}^* \end{bmatrix} \tag{4.29}$$

y_k can be expressed as a linear function of x_k

$$y_k = C_k x_k \tag{4.30}$$

where C_k is a constant matrix defined by

$$C_k = \left.\frac{\partial y_k}{\partial \bar{z}}\right|_{\bar{z}_k^*, \mu_k^*} \tag{4.31}$$

The climb rate and turn rate can be expressed analytically by

$$\begin{aligned} \dot{h} &= \cos\alpha\cos\beta\sin\theta - \sin\phi\sin\beta\cos\theta - \cos\phi\sin\alpha\cos\beta\cos\theta \\ \dot{\psi} &= \frac{q\sin\phi + r\cos\phi}{\cos}\theta \end{aligned} \tag{4.32}$$

An analytic form of C_k can be obtained by evaluating the partial derivatives at the trimmed flight condition. The non-zero components of C_k are:

$$c_{1,1} = 1$$

$$c_{2,1} = \cos\alpha_k^*\cos\beta_k^*\sin\theta_k^* - \sin\phi_k^*\sin\beta_k^*\cos\theta_k^* - \cos\phi_k^*\sin\alpha_k^*\cos\beta_k^*\cos\theta_k^*$$

$$c_{2,2} = V_T^*(-\sin\alpha_k^*\cos\beta_k^*\sin\theta_k^* - \cos\phi_k^*\cos\alpha_k^*\cos\beta_k^*\cos\theta_k^*)$$

$$c_{2,3} = V_T^*(\cos\alpha_k^*\cos\beta_k^*\cos\theta_k^* + \sin\phi_k^*\sin\beta_k^*\sin\theta_k^* + \cos\phi_k^*\sin\alpha_k^*\cos\beta_k^*\sin\theta_k^*)$$

$$c_{2,5} = V_T^*(-\cos\alpha_k^*\sin\beta_k^*\sin\theta_k^* - \sin\phi_k^*\cos\beta_k^*\cos\theta_k^* + \cos\phi_k^*\sin\alpha_k^*\sin\beta_k^*\cos\theta_k^*)$$

$$c_{2,6} = V_T^*(-\cos\alpha_k^*\sin\beta_k^*\cos\theta_k^* + \sin\phi_k^*\sin\alpha_k^*\cos\beta_k^*\cos\theta_k^*)$$

$$c_{3,3} = \frac{(q_k^*\sin\phi_k^* + r_k^*\cos\phi_k^*)\sin\theta_k^*}{\cos\theta_k^{*2}}$$

$$c_{3,4} = \frac{\sin\phi_k^*}{\cos\theta_k^*}$$

$$c_{3,4} = \frac{q_k^*\cos\phi_k^* + r_k^*\sin\phi_k^*}{\cos\theta_k^*}$$

$$c_{3,4} = \frac{\cos\phi_k^*}{\cos\theta_k^*}$$

where $c_{i,j}$ is the component of C_k located in the i^{th} row and j^{th} column.

The definition of a new state vector ξ, whose dynamics is defined by $\dot{\xi}_k = -y_k$, leads to an augmented system

$$\begin{bmatrix} \dot{x}_k \\ \dot{\xi}_k \end{bmatrix} = \begin{bmatrix} A_k & 0 \\ -C_k & 0 \end{bmatrix} \begin{bmatrix} x_k \\ \xi_k \end{bmatrix} + \begin{bmatrix} B_k \\ 0 \end{bmatrix} u_k \tag{4.33}$$

The substitution $\zeta_k = \begin{bmatrix} x_k^T & \xi_k^T \end{bmatrix}^T$ enables (4.33) to be written compactly as

$$\dot{\zeta}_k = \hat{A}_k \zeta_k + \hat{B}_k u_k \tag{4.34}$$

where

$$\hat{A}_k = \begin{bmatrix} A_k & 0 \\ -C_k & 0 \end{bmatrix}, \quad \hat{B}_k = \begin{bmatrix} B_k \\ 0 \end{bmatrix}$$

If this augmented system is controllable, there exists a feedback controller

$$u_k = -\hat{K}_k \zeta_k \tag{4.35}$$

that can make the closed-loop system

$$\dot{\zeta}_k = (\hat{A}_k - \hat{B}_k \hat{K}_k) \zeta_k \tag{4.36}$$

stable and the state vector ζ_k will converge to zero asymptotically. As a result, the integral of the tracking error decays to zero asymptotically, thereby improving the configuration tracking performance.

- Nonlinear PID controller for trim transitions

Extending the above control design to the transition control leads to nonlinear PID controller

$$\mu(h(t),V_T^*(t),\dot{h}^*(t),\dot{\psi}^*(t)) = \mu^*(t) - \hat{K}(t)\zeta(t)$$
$$= \mu^*(t) - K_x(t)x(t) + K_\xi(t)\int_{t_0}^{t} y(\tau)d\tau$$
$$= \mu^*(t) - K_x(t)(\bar{z}(t) - \bar{z}^*(t)) + K_\xi(t)\int_{t_0}^{t}\begin{bmatrix} V_T(\tau) - V_T^*(\tau) \\ \dot{h}(\tau) - \dot{h}^*(\tau) \\ \dot{\psi}(\tau) - \dot{\psi}^*(\tau) \end{bmatrix} d\tau \tag{4.37}$$

where $\hat{K}(t) = \begin{bmatrix} K_x(t) & K_\xi(t) \end{bmatrix}$ and t_0 is the transition start time. The desired control settings $\mu^*(t)$ and the desired aircraft states $\bar{z}^*(t)$ are given by (4.26), while the desired flight condition ($V_T^*(t)$, $\dot{h}^*(t)$, $\dot{\psi}^*(t)$) is determined by (4.25). Similarly, $\hat{K}(t)$ is computed with linear interpolation and given by

$$\hat{K}(t) = \hat{K}_i + \frac{\hat{K}_i + \hat{K}_j}{\Delta t}(t - t_0), \qquad \forall t \in [t_0, t_0 + \Delta t] \tag{4.38}$$

where $\hat{K}_i$ is the controller gain matrix designed for the initial trim state, and $\hat{K}_j$ is the gain matrix designed for the terminal trim state. A gain scheduling strategy is utilized for this nonlinear controller.

- Control gains by an LQR design

Different techniques can be used to design the controller gain matrix $\hat{K}_k$ for a trim state. For example, by the placement of the poles of the closed-loop system (4.36) at desired locations on complex plane, $\hat{K}_k$ can be determined for the trim state [3]. These desired closed-loop poles represent the nominal closed-loop dynamics of the aircraft.

However, this technique cannot be used in the controller gain design for the GTM aircraft since no nominal controllers are available. In this thesis, the controller gains are designed for the GTM aircraft by using LQR technique, which yields an optimal controller that minimizes a cost function evaluating both the system errors and control efforts.

Given a controllable aircraft about an trim state

$$\dot{\zeta}_k = \hat{A}_k \zeta_k + \hat{B}_k u_k \tag{4.39}$$

the LQR design finds an optimal control

$$u_k^* = -R_k^{-1} \hat{B}_k^T P \zeta_k \tag{4.40}$$

that minimizes the cost function

$$J_{LQR} = \int_{t_0}^{\infty} (\zeta_k^T Q_k \zeta_k + u_k^T R_k u_k) dt \tag{4.41}$$

where Q_k and R_k are respectively a $n \times n$ and a $m \times m$ positive definite matrix, where n is the length of ζ_k and m is the length of u_k. The P in Equation (4.40) can be defined by solving the Ricatti equation

$$\hat{A}_k^T P + P\hat{A}_k + Q_k = P\hat{B}_K R_k^{-1} \hat{B}_k^T P \tag{4.42}$$

The optimal control (4.39) automatically guarantees the stability of the closed-loop system

$$\dot{\zeta}_k = (\hat{A}_k - \hat{B}_k R_k^{-1} \hat{B}_k^T P) \zeta_k \tag{4.43}$$

if the weighting matrices Q_k and R_k are positive definite. In practice, Q_k and R_k are typically chosen to be diagonal matrices

$$
\begin{aligned}
Q_k &= diag\{\begin{bmatrix} q_{k,1} & q_{k,2}, & ..., & q_{k,n} \end{bmatrix}\} \qquad q_{k,i} > 0, \forall i = 1,2,...,n \\
R_k &= diag\{\begin{bmatrix} r_{k,1} & r_{k,2}, & ..., & r_{k,m} \end{bmatrix}\} \qquad r_{k,i} > 0, \forall i = 1,2,...,m
\end{aligned} \tag{4.44}
$$

Varying matrices Q_k and R_k yields the optimal controls over a suite of "optimal" closed-loop system responses. Generally, a dominant Q_k enables a closed-loop system with high control precision and high control effort, while a dominant R_k yields a closed-loop system with low control effort and low control precision.

In this thesis, the control gain matrix $\hat{K}_k$ in the controller (4.35) is defined by the LQR design described as above,

$$\hat{K}_k = R_k^{-1} \hat{B}_k^T P \tag{4.45}$$

Although different weighting matrices can be chosen for each trim state, constant Q_k and R_k are used in this thesis since they yield uniform closed-loop system performance characteristics well within the flight envelope. Furthermore, constant weighting matrices can significantly simplify the control design by avoiding the tedious work of manually tuning the controller at each trim state.

For the left-wing-damaged GTM aircraft, the diagonal constant matrix Q_k is chosen in a way such that the lateral motion state errors are weighted dominantly in the cost function (4.41) since the missing left wingtip results in more difficulty controlling lateral motions. Meanwhile, roughly even weights are enforced on five components of control effort by the matrix R_k defined as

$$R_k = r \times I_{5\times 5} \tag{4.46}$$

where *r* is a constant. The case study presented in Chapter 6 uses an *r* of 100.

The simulation results from the damaged GTM case study demonstrate the success of this control design strategy. The performances of the PID controller during a trim transition are shown in Figure 4-2 through 4-5. In the simulation, the aircraft at initial altitude of 5,000 ft is commanded to perform a desired transition leading it from an initial trimmed flight condition with $V_T^* = 750\,ft/s$, $\dot{h}^* = 0$, and $\dot{\psi}^* = 0$, to a terminal trimmed flight condition with $V_T^* = 750\,ft/s$, $\dot{h}^* = -300\,ft/\min$, and $\dot{\psi}^* = -0.5\,\deg/s$, over a typical time interval of 15 seconds, which is also adopted in the case study of this thesis. As shown in Figures 4-2 and 4-3, the controller provides good tracking of the desired trim states as well as the transition between them. The relatively large terminal location tracking errors shown in Figure 4-4 will not influence the subsequent trajectory planning since the trajectory planner uses the actual transition path changes rather than the desired transition path changes in its analyses. Last, Figure 4-5 shows that all the control inputs are well within their saturation limits.

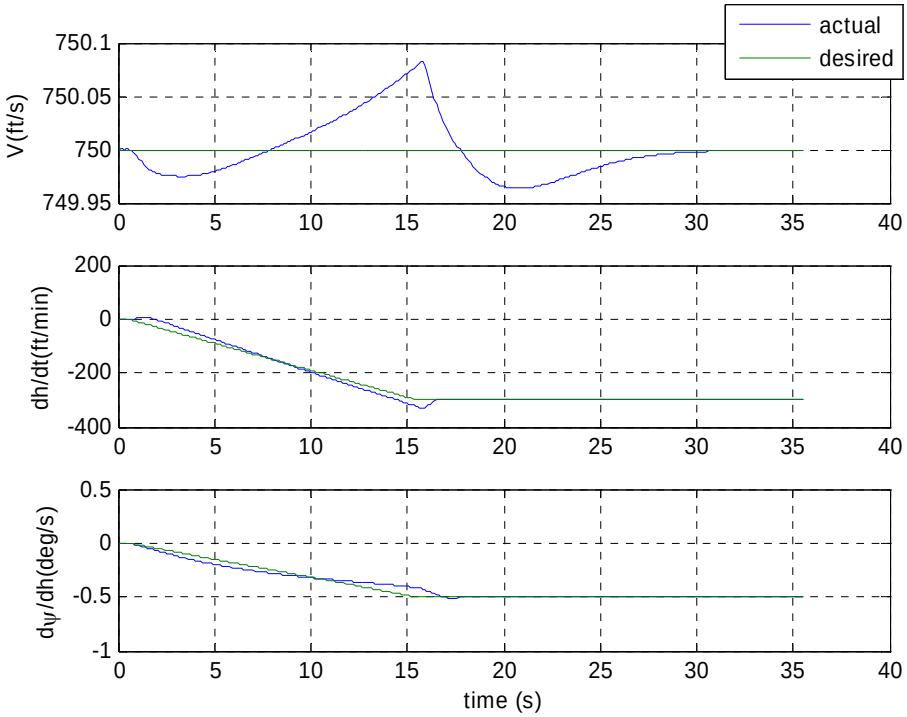

Figure 4-2: Flight Condition Response using LQR-based PID controller

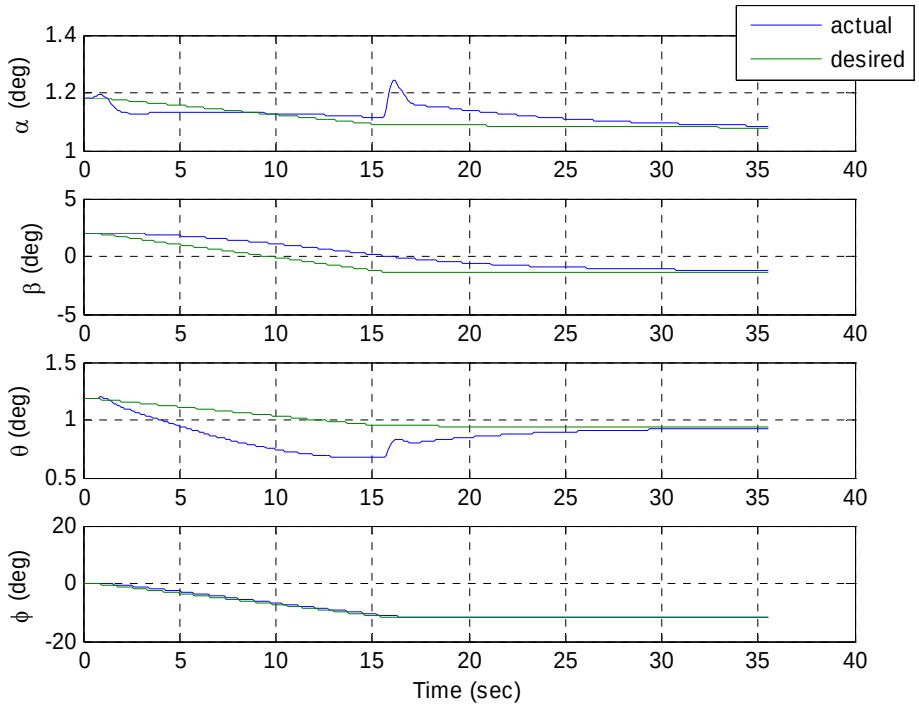

Figure 4-3: Angle Response using LQR-based PID controller

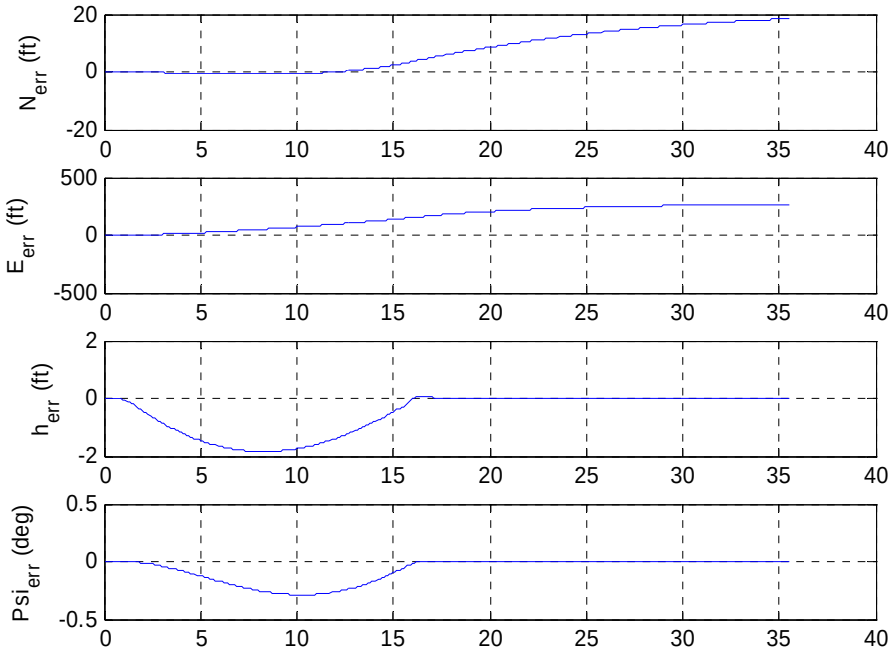

Figure 4-4: Ground Tracking Error using LQR-based PID controller

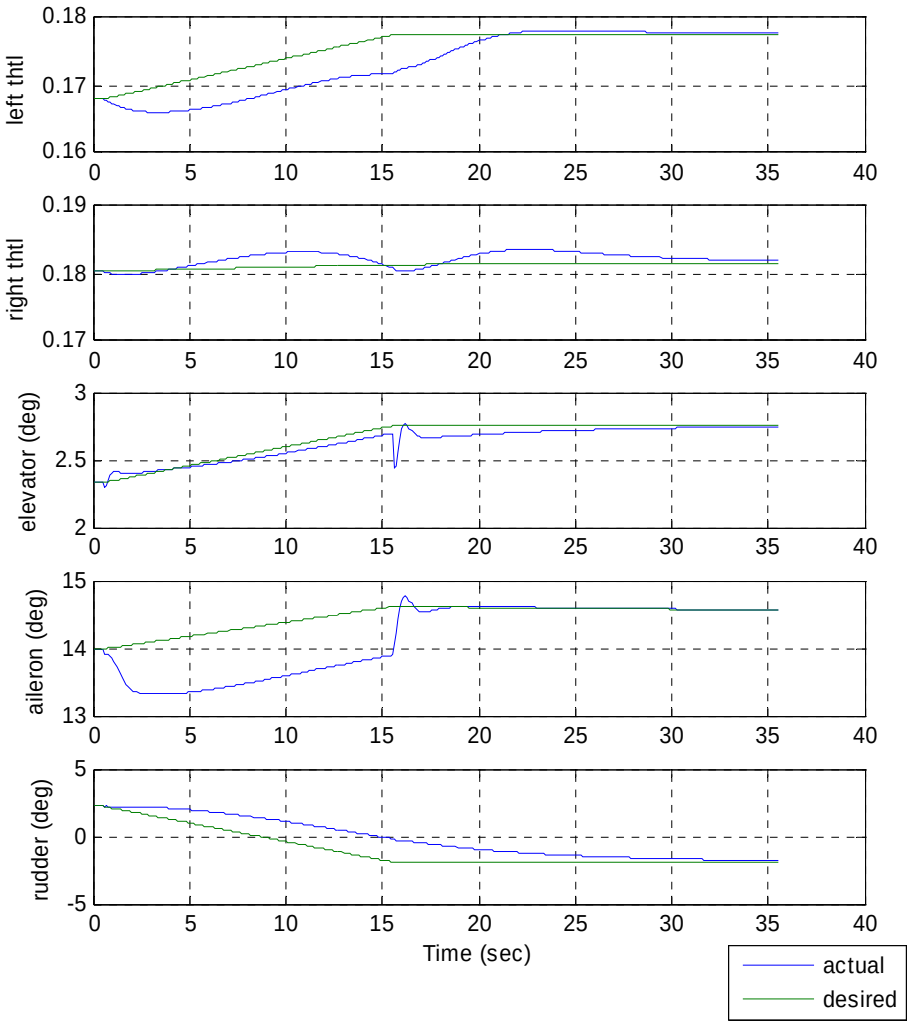

Figure 4-5: LQR-based PID Controller Response

Although the trimmed flight path displacement over a given time interval can be computed analytically (Section 4.1.4), the path displacement incurred by a transition between trim states can only be obtained via numerical simulations. If a transition is numerically simulated starting at time t_1 and all the transients have died away by time t_2, a time-parameterized position vector $p(t)$ and heading $\psi(t)$ are produced. Then, the transition position change is given by

$$\Delta p = p(t_2) - p(t_2) \tag{4.47}$$

while the transition heading change is given by

$$\Delta \psi = \psi(t_2) - \psi(t_1) \tag{4.48}$$

Performing the numerical simulations over the spectrum of potential trim transitions yields a transition database M from which the trajectory planner retrieves these pre-computed path changes that occur during each transition.

4.3 Aircraft Kinematic Model

A kinematic model [3] is used to specify the aircraft's motion, both position and heading, along a segmented flight path. The initial aircraft flight path configuration can be represented by a 4×4 matrix

$$F_0 = \begin{bmatrix} R_{\psi_0}^T & p_0 \\ 0 & 1 \end{bmatrix} \tag{4.49}$$

where p_0 is the 3×1 vector that specifies initial aircraft position in the inertial space, and

$$R_{\psi_0} = \begin{bmatrix} \cos\psi_0 & \sin\psi_0 & 0 \\ -\sin\psi_0 & \cos\psi_0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.50}$$

where ψ_0 is the aircraft's initial flight path heading.

Given the flight path change of the first flight path segment, namely, $\Delta p_1 = p_1 - p_0$ and $\Delta \psi_1 = \psi_1 - \psi_0$, where p_1 and ψ_1 are respectively the inertial aircraft flight path position and heading at the terminal point of the path segment, the

aircraft flight path configuration at the terminal point of this segment can by defined by

$$\begin{aligned} F_1 &= \begin{bmatrix} R_{\psi_1}^T & p_1 \\ 0 & 1 \end{bmatrix} \\ &= F_0 \begin{bmatrix} R_{\Delta\psi_1}^T & R_{\psi_0}\Delta p_1 \\ 0 & 1 \end{bmatrix} \\ &= F_0 G_{0,1} \end{aligned} \tag{4.51}$$

where $G_{0,1}$ represents the flight path change over the first flight path segment between the initial and the 1st flight path point. Therefore, the terminal aircraft flight path configuration after a sequence of N consecutive flight path segments can be represented by

$$F_N = F_0 \prod_{i=1}^{N} G_{i-1,i} \tag{4.52}$$

where $G_{i-1,i}$ represents the flight path change over the i^{th} flight path segment connecting the $(i\text{-}1)^{\text{th}}$ and i^{th} flight path point, and is defined as

$$G_{i-1,i} = \begin{bmatrix} R_{\Delta\psi_i}^T & R_{\psi_{i-1}}\Delta p_i \\ 0 & 1 \end{bmatrix} \tag{4.53}$$

where $\Delta p_i = p_i - p_{i-1}$ and $\Delta\psi_i = \psi_i - \psi_{i-1}$. p_i and ψ_i denote the inertial aircraft position and heading at the i^{th} flight path point, which is consistent with the denotation in (4.49) and (4.50).

As discussed in Sections 4.1.5 and 4.2.4, the flight path change over each flight path segment is determined analytically for trim state segments and numerically for transition segments. For the i^{th} trimmed segment in a flight path, the flight path

change is rapidly computed via equations (4.18) and (4.19), which show that a trimmed flight path change is function of the trimmed flight condition $((V_T^*)_i, \dot{h}_i^*, \dot{\psi}_i^*)$ and the duration Δt_i , and thus can be further denoted by

$$G((V_T^*)_i, \dot{h}_i^*, \dot{\psi}_i^*, \Delta t_i) = \begin{bmatrix} R^T_{\dot{\psi}_i^* \Delta t_i} & R_{\psi_{0,i}} \Delta p_i \\ 0 & 1 \end{bmatrix} \tag{4.54}$$

where ψ_{0i} denotes the initial heading of this trimmed flight segment, and Δp_i - the flight path position change over the duration Δt_i - is computed by equation (4.19). For a transition between trim states, the flight path change is pre-computed from numerical simulation and stored in a transition database for the trajectory planner to retrieve. Since a transition flight is determined by the initial and terminal trimmed flights, the flight path change over the transition connecting the (i-1)$^{\text{th}}$ and i^{th} trimmed flight segments can be further represent as

$$G_{i-1,i} = \begin{bmatrix} R^T_{\Delta\psi_{i-1,i}} & R_{(\psi_0)_{i-1,i}} \Delta p_{i-1,i} \\ 0 & 1 \end{bmatrix} \tag{4.55}$$

where $(\psi_0)_{i-1,i}$ denotes the initial heading over this transition, with the flight path position change $\Delta p_{i-1,i}$ defined by equation (4.47) and the heading change $\Delta\psi_{i-1,i}$ defined by equation (4.48).

As a result, the terminal flight path configuration over a flight path composed of a sequence of N trimmed flight segments can be further derived from equation (4.52) and represented as

$$F_N = F_0 \prod_{i=1}^{N} G_{i-1,i} G((V_T^*)_i, \dot{h}_i^*, \psi^*_i, \Delta t_i) \tag{4.56}$$

This kinematic path representation with the support of the trim and transition databases applicable to a particular failure/damage scenario provide the tools required to generate applicable segmented post-failure/damage landing trajectories.

Chapter 5: Trajectory Planning

This chapter presents a planning algorithm that constructs a feasible trajectory leading the aircraft from its initial state where the failure/damage is detected to the landing runway specified by the LSS (Chapter 3). As defined in Chapter 4, a trajectory is defined as a sequence of finite time trim states, accounting for the smooth transitions between neighboring trim segments. To design a feasible trajectory, the planner must identify an appropriate trim state sequence and assign each trimmed flight segment a duration. The timed trim state sequence is deemed a feasible trajectory if the aircraft terminal state predicted by the aircraft kinematic model coincides with the desired landing site position (latitude, longitude, altitude) and heading.

Identification of landing trajectories is a potentially time-consuming task given the size of typical trim and transition databases and the necessity to identify appropriate times to hold each segment. Since safety is the primary concern for landing an aircraft with damaged or failed systems that impact the performance envelope, the trajectory planner's goal is to find feasible rather than optimal solutions, allowing the planner to consider instead a significantly reduced set of trim states. Searching for a feasible solution in a highly reduced space makes it possible for the trajectory planning to be performed in real-time.

Section 5.1 describes the two-step trajectory planning procedure, which is designed to reduce computational complexity and increase the probability of finding solutions in

real-time. Section 5.1 also describes the trim database reduction process. Section 5.2 describes an algorithm contributed by this work that performs the first step of the trajectory planning process, deterministically commanding a trim sequence to bring the aircraft from its initial state where failure/damage is detected to an intermediate state relatively close to the desired landing site. Execution of this deterministic trajectory provides time for computing the final approaching trajectory, which need not be initiated until the aircraft reaches the "intermediate state". Section 5.3 describes a hybrid discrete/continuous algorithm [3] designed to plan the second part of the landing trajectory that uses the reduced trim and transition databases to guide the aircraft from the intermediate state to the runway.

5.1 Landing Trajectory Design

The task of the trajectory planner is to identify a sequence of trimmed flight conditions that allows the aircraft to reach the landing runway with the correct heading. Although trim transitions are necessary to connect neighboring trim segments, it is sufficient to represent a trajectory (flight plan) by only specifying each trim flight segment, since a constant-time trim transition is completely specified by the initial and terminal trim states it connects. Let each trimmed flight segment be specified as

$$s_i = ((V_T^*)_i, \dot{h}_i^*, \dot{\psi}_i^*, \Delta t_i) \tag{5.1}$$

where the triplet $((V_T^*)_i, \dot{h}_i^*, \dot{\psi}_i^*)$ specifies the segment's trimmed flight condition and Δt_i is the duration of the segment. A trajectory plan then can be expressed as a sequence of N trim segments:

$$P = \{s_i\}_{i=1}^{N} = \{((V_T^*)_i, \dot{h}_i^*, \dot{\psi}_i^*, \Delta t_i)\}_{i=1}^{N} \tag{5.2}$$

A candidate trajectory plan $\widetilde{P}$ is a partially-instantiated plan consisting of valid trim state triplets for each segment, but with no durations specified. Thus, a candidate plan $\widetilde{P}$ can be expressed as

$$\widetilde{P} = \{((V_T^*)_i, \dot{h}_i^*, \dot{\psi}_i^*)\}_{i=1}^{N} \tag{5.3}$$

5.1.1 Two-Step Trajectory Planning

The Landing Site Search discussed in Chapter 3 is fast (under one second execution time). Therefore, several aircraft instantaneous states can be considered approximately equivalent if the LSS is executed immediately after the failure or damage occurs. These instantaneous states are:

- state at the time when failure/damage occurs (or is detected, assumed concurrent in this work)
- state at the time when the LSS is executed
- state at the time the target landing runway is selected

The value for these near-equivalent states is straightforwardly set to the state just after the failure/damage occurs. This value will be used by the trajectory planner as the initial state, of which the position and heading components are represented as p_{init} and ψ_{init}. Correspondingly, the desired landing site can be represented as p_{des} and ψ_{des}.

The planner builds a feasible trajectory between the initial (p_{init} , ψ_{init}) and the desired (p_{des}, ψ_{des}) states in two stages. The first stage checks the relative position of p_{init} with respect to p_{des} and builds *trajectory part I* if p_{init} doesn't satisfy a three-dimensional geometric constraint, expressed as

$$\begin{aligned} &\underline{h} \le h_{init} - h_{des} \le \bar{h} \\ &(x_{init} - x_{des})^2 + (y_{init} - y_{des})^2 \le r^2 \end{aligned} \tag{5.4}$$

where (*x, y*) and *h* denotes respectively the location and the altitude of a 3-D point. By following *trajectory part I*, the aircraft flies to an intermediate position p_{int} , which falls in the cylindrical neighborhood region of p_{des} , with heading ψ_{int} . Then for the second step, a search-based path planner constructs *trajectory part II* that connects the intermediate $(p_{\text{int}}, \psi_{\text{int}})$ to the final (p_{des}, ψ_{des}). For the consistency in the later discussion, the initial state where *trajectory part I* is generated is denoted by (p_0^I, ψ_0^I) and the initial state for planning trajectory part II is denoted by (p_0^{II}, ψ_0^{II}). The subscript "0" denotes initial state for generating the trajectory parts, while the superscript *I* and *II* denotes *trajectory part I* and *trajectory part II* respectively.

This two-step procedure is illustrated in Figure 5-1 in the context of the Adaptive Flight Planner (AFP). In the figure, *PLANNER_I* and *PLANNER_II* generate respectively the partial flight plan P^I for *trajectory part I* and the other partial plan P^{II} for *trajectory part II*. As described in the previous paragraph, the initial state for procedure *PLANNER_I* is denoted by (p_0^I, ψ_0^I) with superscript *I* and subscript 0,

while the initial state for procedure *PLANNER_II is* represented by (p_0^{II}, ψ_0^{II}) with superscript *II* equivalently**.**

1. Set AFP flight envelope/algorithmic model based on type of failure *Fail*
2. Compute initial state (p_0^I, ψ_0^I)
3. Find sorted feasible landing site list *ls_list* via LSS
   $ls\_list \leftarrow LSS(p_0^I, \psi_0^I, Fail)$
4. **if** *ls_list* is empty
   alert pilot AFP cannot find a feasible runway; exit
   else
   determine the desired landing site (p_{des} , ψ_{des}) by copying the top node in the *ls_list*, then remove the top node
   end if
5. Execute trajectory planner I
   $(P^I, p_0^{II}, \psi_0^{II},) \leftarrow PLANNER\_I(p_0^I, \psi_0^I, p_{des}, \psi_{des})$
6. Compute the allowed planning time t_{plan} for trajectory planner II
   $t_{plan} = time(p_0^{II}, \psi_0^{II}) - time(p_0^I, \psi_0^I)$
7. **if** $t_{plan} < t_{min}$
   Choose next-highest ranked landing runway (goto step 4)
   end if
8. Initiate P^I (continue to step 9 concurrently)
9. Execute trajectory planner II until completed or until t_{plan} expires
   $L \leftarrow PLANNER\_II(p_0^{II}, \psi_0^{II}, p_{des}, \psi_{des}, t_{plan})$
10. **if** *L* is not empty
    $P^{II} \leftarrow L(1)$
    else
    alert pilot AFP was unable to find a landing trajectory; exit
    end if
11. Execute P^{II} to landing runway

Figure 5-1: Two-Step Trajectory Planning in the Context of AFP

In Figure 5-1, step 2 initiates (p_0^I, ψ_0^I) as (p_{init}, ψ_{init}). Besides P^I as the first part of the flight plan *P*, *PLANNER_I* also outputs the intermediate state $(p_{\text{int}}, \psi_{\text{int}})$, which is the initial state (p_0^{II}, ψ_0^{II}) for planning the landing *trajectory part II* (P^{II}). Time

t_{plan} is the time interval over which the aircraft flies from (p_0^I, ψ_0^I) to (p_0^{II}, ψ_0^{II}) by executing P^I, or the duration of trajectory part I. t_{plan} is also used as a planning time constraint for *PLANNER_II* since P^{II} must be planned before it can be executed. Step 7 evaluates whether t_{plan} is sufficient for *PLANNER_II* to complete at least one feasible plan, selecting an alternate landing site if t_{plan} is too brief. Time t_{min} is the minimum value set to test t_{plan} and is empirically selected to be 60 seconds for the case study in this thesis since it is sufficient for all the scenarios studied in this thesis. Step is also a simple test that is dependent on the existence of multiple feasible runways. In the absence of alternate landing sites, the aircraft can be commanded to fly away from the landing site (straight or in a holding pattern) then come back, making t_{plan} as large as possible provided sufficient fuel exists and the aircraft can maintain altitude long-term. The same argument holds for the "failure" in step 10: Even if L is empty, it may be possible to select another landing site (with (p_0^I, ψ_0^I) updated as the current aircraft state at step 10) for which a solution can be found with sufficient speed. While the above discussion gives a suggestion for future research, they are not the focus of this thesis. Thus the more detailed discussion will not be given in this thesis. The efficient execution (under one second) of Steps 1 through 7 guarantee the aircraft is still approximately in state (p_0^I, ψ_0^I) when P^I is initiated.

5.1.2 Database Reduction

The trajectory planners build their solutions as sequences of trim states found in the post-failure/damage database developed as described in Chapter 4. To enable real-

time plan development, the space of possible trim states to be sequenced must have a tractable size. The full set of controllable states in the trim database can be represented as the set

$$D = \left\{\left(h_k^*,\ (V_T^*)_k,\ \dot{h}_k^*,\ \dot{\psi}_k^*\right)\right\}_{k=1,\ldots,N_D} \tag{5.5}$$

where N_D is the total number of the trim states in database D. Because a full trim database is developed to provide understanding of flight characteristics as well as provide candidate trim states for the trajectory planners, N_D is very large even for the left-wing-damaged GTM aircraft. Tractable planning thus makes it necessary to reduce this database to a small subset of its original size. Since altitude cannot be independently specified, D is first contracted over altitude to produce the flight condition database D', the intersection of all three-dimensional altitude slices, from h_0 to h_n, in the trim database:

$$D' = \bigcap_{i=0}^{n} D(h_i^*, (V_T^*)_k, \dot{h}_k^*, \dot{\psi}_k^*) = \left\{\left((V_T^*)_k, \dot{h}_k^*, \dot{\psi}_k^*\right)\right\}_{k=1,\ldots,N_{D'}} \tag{5.6}$$

where $N_{D'} < N_D$. This procedure is successful so long as this intersection yields a sufficiently large set of trim states, which has been the case for all GTM and F-16 failures analyzed to-date. Further contraction of the database can be accomplished by removing additional climb rate, turn rate, airspeed points, retaining a sufficient set approximately spanning the flight envelope. The new, contracted database $\tilde{D}$ can be represented as

$$\tilde{D} = \left\{\left((V_T^*)_k, \dot{h}_k^*, \dot{\psi}_k^*\right)\right\}_{k=1,\ldots,N_{\tilde{D}}} \tag{5.7}$$

where $\tilde{D} \subset D'$ by definition and $N_{\tilde{D}}$ is the size of the contracted database with $N_{\tilde{D}} << N_D$. A heuristic method is presented in [3] to perform the database

contraction $D' \rightarrow \tilde{D}$. With this method, slices of the D volume are taken over a discrete set of airspeeds. Within each slice, a square is superimposed that encompasses the range of climb rates and turn rates of the two-dimensional slice. Over the discrete range of airspeeds, the combined three-dimensional shape well defines $\tilde{D}$. A complete discussion can be found in [3]. By following this approach, two boundary-altitude slices of an example trim database D of the left-wing-damaged GTM aircraft are depicted in Figure 5-2, and then is reduced to $\tilde{D}$ expressed in Table 5-1.

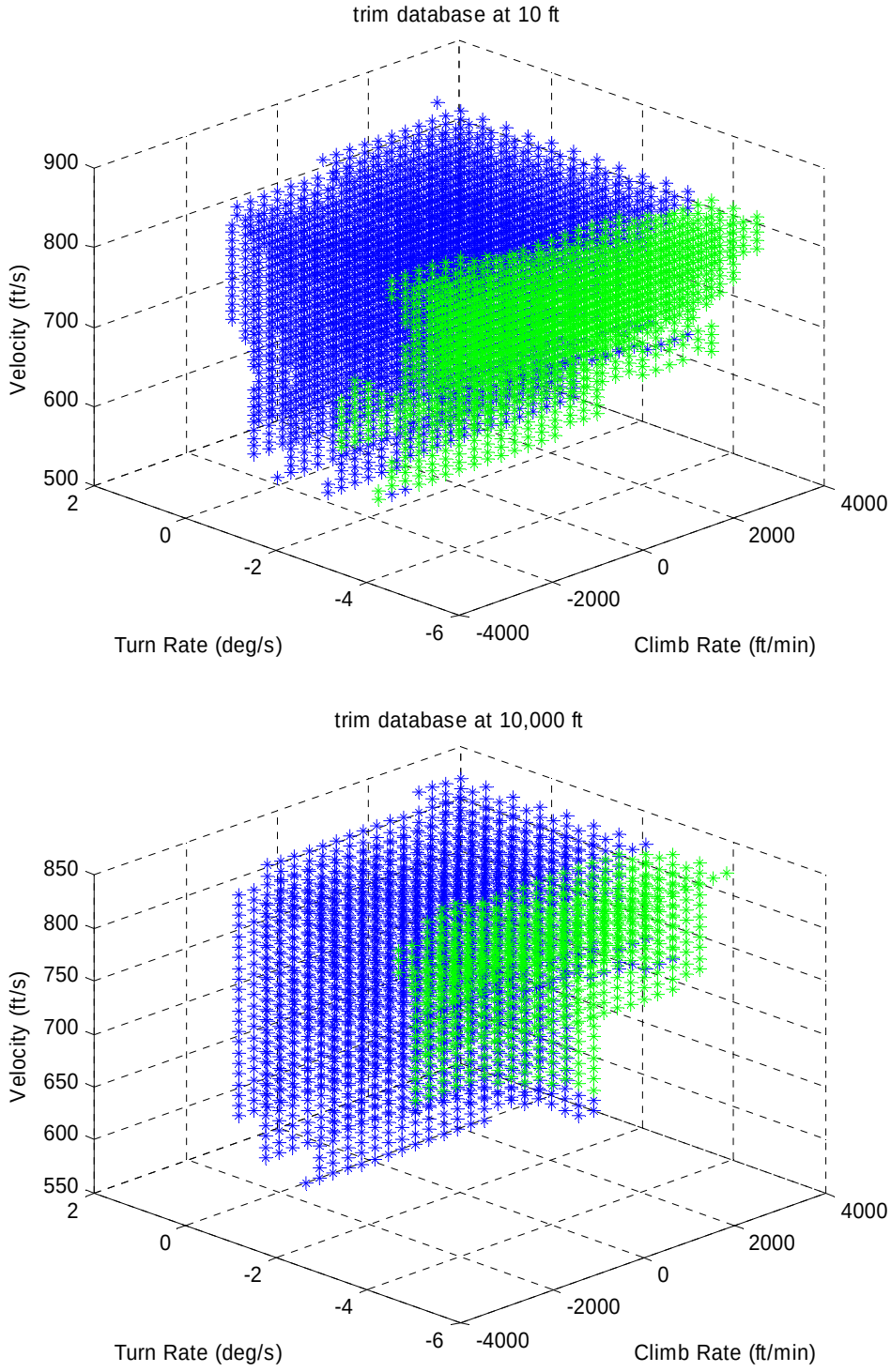

Figure 5-2: Trim Database for the damaged GTM aircraft at altitude of 10 ft and 10,000 ft

Table 5-1: A Reduced Trim Database $\tilde{D}$ for the Damaged GTM aircraft

Airspeed (ft/s)	Climb Rate (ft/s)	Turn Rate (deg/s)	points
800	0, ±5, ±10	-5, -2, 0, 0.5	20
750	0, ±5, ±10	-4.5, -2, 0, 0.5	20
700	0, ±5, ±10	-4.5, -2, 0	15
650	0, ±5, ±10	-2, -1, 0	15
Total points			70

In Figure 5-2, the size of the example database D is intuitively shown by the numerous colored points in trim state space, while the reduced database $\tilde{D}$ shown in Table 5-1 only contains 70 trim states. Thus, the trajectory planner can exhaustively search $\tilde{D}$ for solutions in real-time.

Transition database M is also straightforwardly reduced in accordance with $\tilde{D}$, resulting in databases $\tilde{M}$ to be used with $\tilde{D}$ by the trajectory planner.

5.2 Trajectory Planning Part I

An efficient algorithm *PLANNER_I* was developed to rapidly plan landing *trajectory part I*. As discussed in Section 5.1.1, *trajectory part I* has two effects. First, by following this trajectory the aircraft can fly into a neighborhood region of (p_{des}, ψ_{des}), as defined by (5.4). Next, the time required to fly this trajectory will allow search-based *PLANNER_II* sufficient time to find a feasible landing *trajectory part II*. The *PLANNER_I* algorithm is shown in Figure 5-3.

Algorithm: $PLANNER\_I(p_0^I, \psi_0^I, p_{des}, \psi_{des})$

1. Initialize P^I as an empty plan
2. Initialize $(p_{end}^I, \psi_{end}^I) \leftarrow (p_0^I, \psi_0^I)$
3. Read data from databases $\tilde{D}$, and $\tilde{M}$
4. Compute altitude offset: $\Delta h = h_0^I - h_{des}$
5. **if** $\Delta h > \bar{h}$
6. $(s_1^I, p_1^I, \psi_1^I) \leftarrow GetSpiralDown(p_0^I, \psi_0^I, p_{des}, \tilde{D}, \tilde{M}, ((V_T^*)_0^I, \dot{h}_0^I, \dot{\psi}_0^I))$
7. **else if** $\Delta h < \underline{h}$
8. $(s_1^I, p_1^I, \psi_1^I) \leftarrow GetSpiralUp(p_0^I, \psi_0^I, p_{des}, \tilde{D}, \tilde{M}, ((V_T^*)_0^I, \dot{h}_0^I, \dot{\psi}_0^I))$
9. **else**
10. Set segment duration #1: $\Delta t_1^I \leftarrow 60$
11. $(s_1^I, p_1^I, \psi_1^I) \leftarrow GetCircle(p_0^I, \psi_0^I, p_{des}, \Delta t_1^I \equiv 60, , \tilde{T}, \tilde{M}, ((V_T^*)_0^I, \dot{h}_0^I, \dot{\psi}_0^I))$
12. **end if**
13. Append flight segment s_1 to plan P^I: $P^I \leftarrow Append(P^I, s_1^I)$
14. Set segment duration #2: $\Delta t_2^I \leftarrow 0$
15. $(s_2^I, p_2^I, \psi_2^I) \leftarrow GetCircle(s_1^I, p_1^I, \psi_1^I, p_{des}, \tilde{D}, \tilde{M})$
16. Append flight segment s_2^I to plan P^I: $P^I \leftarrow Append(P^I, s_2^I)$
17. Compute the position offset: $d = \left|(x_2^I, y_2^I) - (x_{des}, y_{des})\right|$
18. **if** $d \le r$
19. $(p_{end}^I, \psi_{end}^I) \leftarrow (p_2^I, \psi_2^I)$
20. **else**
21. $(s_2^I, s_3^I, p_3^I, \psi_3^I) \leftarrow FlyToLS(s_2^I, p_2^I, \psi_2^I, p_{des}, \tilde{D}, \tilde{M})$
22. Modify segment s_2^I in P^I: $P^I \leftarrow Modify(P^I, s_2^I)$
23. Append segment s_3^I to P^I: $P^I \leftarrow Append(P^I, s_3^I)$
24. $(p_{end}^I, \psi_{end}^I) \leftarrow (p_3^I, \psi_3^I)$
25. **end if**
26. $(p_0^{II}, \psi_0^{II}) \leftarrow (p_{end}^I, \psi_{end}^I)$

return $(P^I, p_0^{II}, \psi_0^{II})$

Figure 5-3: *PLANNER_I* Algorithm

The inputs to *PLANNER_I* are the initial aircraft 3-D location and heading, and the desired terminal location and heading (p_{des}, ψ_{des}), the landing runway position and heading. *PLANNER_I* first reads databases $\tilde{D}$, and $\tilde{M}$. Trajectory plan P^I is initialized

as an empty plan and the terminal state $(p^I_{end}, \psi^I_{end})$ at the end of this set of trajectory segments is initially set to initial state (p^I_0, ψ^I_0). Then, the planner examines the difference Δh between initial and desired landing altitudes. If Δh is above upper bound $\bar{h}$, procedure *GetSpiralDown* will plan the first flight segment s^I_1, a trimmed steady turning-descending flight segment. By following this segment, the aircraft will spiral down to state (p^I_1, ψ^I_1) with altitude $h^I_1 = h_{des} + \bar{h}$. Similarly, if the initial aircraft altitude violates lower bound $h_{des} + \underline{h}$, the procedure *GetSpiralUp* will plan the first flight segment as a trimmed steady turning-ascending flight, which results in a terminal trajectory point with altitude h^I_1 equal to $h_{des} + \underline{h}$. However, if the aircraft initial altitude satisfies constraint $\underline{h} < h^I_1 - h_{des} < \bar{h}$, the procedure *GetCircle* will create a steady turning level trimmed flight as the first flight segment to effectively create a constant-altitude "holding pattern turn" during which trajectory *PLANNER_II* has time to build a landing trajectory. Note that the duration of the first segment is temporarily set to a positive constant, for example, 60 seconds as shown in the algorithm, since an altitude variation is not required. Once the first circle or spiral flight segment s^I_1 is determined, it is sequenced into flight plan P^I. In the algorithm, steps 14 and 15 plan the second flight segment s^I_2 as a trimmed level turning flight segment that points the aircraft toward the landing runway. The terminal aircraft state of segment s^I_2 will be (p^I_2, ψ^I_2). Afterwards, *PLANNER_I* examines the 2-D *x-y* distance d between p^I_2 and p_{des}. If p^I_2 falls within the region defined as a circle centered at (x_{des}, y_{des}) with radius r, it is considered to be a qualified initial location

for the planner *PLANNER_II* that is responsible for planning the landing *trajectory part II* and thus defined as the terminal location $(p^I_{end}, \psi^I_{end})$ of *trajectory part I.* If d is greater than r, the procedure *FlyToLS* designs a third flight segment s^I_3, a straight level flight with the terminal state (p^I_3, ψ^I_3), of which p^I_3 falls within the region and defines (p^I_3, ψ^I_3) as $(p^I_{end}, \psi^I_{end})$; then segment s^I_3 is sequenced into the plan P^I. The final step sets the terminal state $(p^I_{end}, \psi^I_{end})$ as (p^{II}_0, ψ^{II}_0). Then, the algorithm terminates by returning the plan P^I and the state (p^{II}_0, ψ^{II}_0).

The planner *PLANNER_I* assumes that the post-failure/damage aircraft is capable of trimmed spiral ascending and descending flight, as well as circling and straight level flights. In the case study of this thesis, these required trimmed turn rates and climb rates are selected manually from the trim database $\tilde{D}$ of the GTM aircraft, and the transitions between all the trim state pairs are feasible. In future work, the planner may select these trim states automatically from $\tilde{D}$. The Figure 5-3 algorithm was used for the results presented in this thesis. However, extensive simulations conducted as part of this research and for NASA Ames have revealed that the last flight segment s^I_3, a trimmed straight-line level flight toward the desired landing site, is unnecessary and potentially detrimental provided straight flight is possible since the range reduction over this segment can be handled in *trajectory part II.* Furthermore, in some situations, an initial state of *trajectory part II,* (p^{II}_0, ψ^{II}_0), that is too close to the desired landing site (p_{des}, ψ_{des}), may decrease the number of solutions from *PLANNER_II* if the aircraft is limited to shallow turn rates. In Figure 5-3

algorithm, an appropriate selection for the value of *r* at step 18 may prevent this problem. The computation of *r* is dependent on the turning ability of the aircraft. Simulations conducted at Ames shows that a large *r* will be required if the aircraft cannot perform fast turning, which indicates the value of *r* may be primarily dependent on the minimum turn radius that can be computed from the trim states in $\tilde{D}$. In this thesis, *r* is set manually to a safe static value for the case study. While this approach works well for the scenarios that will be presented in Chapter 6, it also suggests the need of an automatic algorithm that computes *r* based on the turning ability of the aircraft.

5.3 Trajectory Planning Part II

Once *PLANNER_I* has returned a plan P^I that defines *trajectory part I*, as well as the intermediate state that is defined as the initial state (p_0^{II}, ψ_0^{II}) of landing *trajectory part II*, *PLANNER_II* can then complete the entire flight plan to desired landing site P_{des} with desired heading ψ_{des}. At the top level, *PLANNER_II* combinatorially searches the reduced discrete trim state space for a partially instantiated solution $\tilde{P}^{II}$ composed of N^{II} ordered trim segments, and then it performs a continuous optimization over flight segment durations Δt_i to determine the existence of a landing solution P^{II} given $\tilde{P}^{II}$. The algorithm used in this thesis is adapted from the planner algorithm in [3] and shown in Figure 5-4. Also, a relatively complete discussion is presented in [3] on the determination of N^{II}-the number of the flight segments used for generating *trajectory part II*. As was the case in [3], N^{II} is set to 4 in this work. Given the size of the reduced trim database ($N_{\tilde{D}}$) and number of segments N^{II}, there are a total

$(N_{\tilde{D}})^{N^{II}}$ trim flight sequences possible. The i^{th} *while* loop examines the existence of solution given a unique sequence, that is, a unique trim state ordering P_i^{II} optimized over segment durations. The procedure *GetCandidatePlan* is responsible for generating partially instantiated plans P_i^{II} from the search space.

Algorithm: $PLANNER\_II(p_0^{II}, \psi_0^{II}, p_{des}, \psi_{des}, t_{plan})$

1. Adopt $\tilde{D}$, and $\tilde{M}$ from *PLANNER_I*
2. Determine the number of trim states in $\tilde{D}$: $N_{\tilde{D}} \leftarrow SizeOf(\tilde{D})$
3. Compute the total number of possible trim sequences:
   $i_{\max} \leftarrow Power(N_{\tilde{D}}, N^{II})$, clear index: $i \leftarrow 0$
4. Initiate an empty plan list: $L \leftarrow NIL$
5. **while** $(i < i_{max})$ **and** $(\delta t < t_{plan})$
6. $\tilde{P}_i^{II} \leftarrow GetCandidatePlan(i, \tilde{D}, N^{II})$
7. **if false** $\leftarrow ValidateSequence(\tilde{P}_i^{II}, \tilde{M})$
8. $i \leftarrow i+1$
9. **continue**
10. **end if**
11. **if false** $\leftarrow PlanConstraint(\tilde{P}_i^{II})$
12. $i \leftarrow i+1$
13. **continue**
14. **end if**
15. $(J_{plan}^*, P_i^{II}) \leftarrow GetDurations(\tilde{P}_i^{II}, \tilde{M}, p_0^{II}, \psi_0^{II}, p_{des}, \psi_{des})$
16. **if** $J_{plan}^* \le \varepsilon_{plan}$ (cost of plan is acceptably low indicating accurate solution)
17. **if false** $\leftarrow CheckAltitude(P_i^{II})$
18. $i \leftarrow i+1$
19. continue
20. **end if**
21. $l_i^{II} \leftarrow ComputeLength(P_i^{II})$
22. $L \leftarrow SortList(L, P_i^{II})$
23. $i \leftarrow i+1$
24. **continue**
25. **else**
26. $i \leftarrow i+1$
27. **continue**
28. **end if**
29. Update running time δt
30. **end while**

return (L)

Figure 5-4: *PLANNER_II* Algorithm

The procedure *ValidateSequence* checks if the transitions between the trimmed flight segments are feasible. The procedure *PlanConstraint* eliminates the candidate plans in which the final segment is not descending flight.

At step 15, the procedure *GetDurations* determines duration $\Delta t_{i,j}^{II}$ of each trimmed flight segment by performing a numerical optimization to minimize the cost function

$$J_{plan} = \left\| F_0^{II} \prod_{j=1}^{N^{II}} G_{j-1,j}^{II} G((V_T^*)_j^{II}, (\dot{h}^*)_j^{II}, (\dot{\psi}^*)_j^{II}, \Delta t_{i,j}^{II}) - F_{des} \right\|_2 \tag{5.8}$$

over variables $\Delta t_{i,j}^{II_i}$. As represented in equation (4.56), the expression $F_0^{II} \prod_{j=1}^{N^{II}} G_{j-1,j}^{II} G((V_T^*)_j^{II}, (\dot{h}^*)_j^{II}, (\dot{\psi}^*)_j^{II}, \Delta t_{i,j}^{II})$ computes the planned terminal flight path configuration by propagating initial flight path configuration F_0^{II} along the trajectory defined by trim sequence $\prod_{j=1}^{N^{II}} G_{j-1,j}^{II} G((V_T^*)_j^{II}, (\dot{h}^*)_j^{II}, (\dot{\psi}^*)_j^{II}, \Delta t_{i,j}^{II})$, and the matrix F_{des} defines the desired terminal flight path configuration. Therefore, the cost J_{plan} describes the magnitude of terminal position and heading error. A numerical optimization algorithm, namely the Nelder-Mead simplex algorithm, is utilized to minimize J_{plan} in (5.8). When (5.8) is minimized, the optimal solutions $\{\Delta t_{i,j}^{II}\}_{j=1}^{N^{II}}$ then defines a complete plan P_i^{II} .

The complete plan P_i^{II} is considered an acceptable plan if the associated cost J_{plan}^* is less than threshold ε_{plan} , a small positive scalar. Furthermore, plan P_i^{II} will only be

considered a feasible plan when *CheckAltitude* verifies the planned terminal aircraft position has altitude at or above the landing runway's altitude. Length l_i^{II} of feasible plan P_i^{II} is defined as the duration sum

$$l_i^{II} = \sum_{j=1}^{N^{II}} \Delta t_{i,j}^{II} \tag{5.9}$$

and is used by the procedure *SortList* to insert the plan P_i^{II} into the sorted plan list *L*, the top-ranked of which is selected as the best solution. Step 29 updates execution time δt, then a new candidate plan P_{i+1}^{II} will be examined in the next *while* loop until the solution space is exhaustively explored or else available planning time expires.

Although numerous local optimization approaches could be used, the Nelder-Mead simplex algorithm has been shown effective. I n this application, the minimization function is (5.8), while the variables are trimmed flight segment durations $\left\{\Delta t_{i,j}^{II}\right\}_{j=1}^{N^{II}}$. In order to reduce the planned touchdown position and heading errors at the desired landing site, smaller ε_{plan} can be used as the criterion for judging a solution. While this method can reduce the planned touchdown errors by eliminating the unqualified Nelder-Mead solutions, these errors can also be reduced in the optimization procedure by increasing the error weighting factors since the position and heading errors are weighted in cost function (5.8). Of course, reducing the planned touchdown errors by weighting these errors more heavily requires more computation time for each optimization. Although the Nelder-Mead simplex algorithm is computationally efficient, the accumulated computation time over all candidate plans is considerable

given the size of the search space. Thus, there is a tradeoff between reducing the planned touchdown errors and reducing the computation time, as well as the number of plans with acceptable touchdown errors, when choosing the design parameters – the touchdown error weighting factors and the acceptable plan criterion ε_{plan}. While this tradeoff would be a good topic in future work, this thesis will not extend the discussion on this topic and instead focuses on the existence of the feasible solutions given a set of design parameters. In the case study (Chapter 6), threshold ε_{plan} is set to 1, while the each component of the 3-D landing position error is evenly weighted by 1 and the landing heading error is weighted by 1000. Thus, the maximum acceptable 1-D landing position error is 1 ft, while the maximum landing heading error is 0.001 radians (approximately 0.06 deg).

Chapter 6: GTM Case Study

The adaptive flight planner was applied to a damaged general transport model (GTM) aircraft with missing left wingtip, as discussed in Chapter 2. Based on the method from Chapter 4, the full trim database was calculated to show the reduced flight envelope of the post-damage aircraft. This database was further contracted to facilitate real-time trajectory planning as described previously. This chapter shows the full and reduced damaged GTM trim databases as well as example emergency landing trajectories generated by the two-step trajectory planner. The emergency scenarios provided in this chapter show the adaptability of the adaptive flight planning approach to the different initial aircraft states when the wingtip damage occurs. To provide context for execution time statistics, this case study is conducted on a 2.20GHz AMD® Athlon® 64 processor. Section 6.1 presents the full trim database of the damaged GMT aircraft, as well as the reduced trim and transition databases for performing trajectory planning. Section 6.2 presents the flight planning results for four different scenarios.

6.1 Database Computation

In this section, a full trim database D of the damaged GTM aircraft is first computed by the method discussed in Chapter 4. Then, the reduced trim database $\tilde{D}$ is defined, and the corresponding reduced transition database is computed.

6.1.1 Full Trim Database

As discussed in Chapter 4, the full trim database is created by characterizing trim states over a discrete set of trimmed steady climbing-turning flight conditions, of which each flight condition is defined by a combination of values from Table 6-1.

Table 6-1: Trim Flight Parameter Ranges for Constructing the Trim Database

Variable	*Minimum Value*	*Maximum Value*	*Step Size*	*# of Pts*
h (ft)	10	30,010	10,000	4
V_T (ft/s)	450	850	20	21
$\dot{h}(ft/s)$	-50	50	10	11
$\dot{\psi}$(deg/s)	-6	2	0.5	17
Total trimmed flight conditions				15,708

The existence of each trimmed flight state, defined as a unique combination of Table 6-1 parameters, is determined by (4.5). A trim state is considered valid for this analysis with error parameter $\varepsilon_{trim} = 1\times10^{-6}$. To develop the linearized aircraft model (4.8), small perturbations in (4.9) and (4.10) are chosen to be 1×10^{-2} for airspeed and

1×10^{-4} for all other state and control variables. Instead of using the theoretical criterion (4.15), a more practical constraint

$$\boldsymbol{R}\{\lambda_i(A_k)\} < \varepsilon_{stab} \qquad i = 1, ..., 8$$

is used to determine the stability of the linearized dynamics with $\varepsilon_{stab} = 1\times10^{-3}$. The controllability of the linearized dynamics is defined by (4.18) and is checked in practice by utilizing the MATLAB built-in functions *ctrb* that returns the controllability matrix defined by equation (4.14), and *rank* that returns the row rank of the controllability matrix.

Figures 6-1 through 6-4 show the 10ft altitude slice of the full trim database. The color at each point indicates the characteristics of each trim state that represents a controllable trimmed flight condition. Green points represent stable and controllable trim states, while blue points indicate unstable but controllable trim states. Controllable trimmed flight is infeasible at all unmarked points.

The missing left wingtip has a significant impact on the flight envelope of the GTM. The aircraft cannot achieve trimmed flight with airspeed lower than 520 ft/sec since relatively high airspeeds are necessary to compensate for the lift loss due to the decrease in left wing area. While the aircraft can achieve a variety of trimmed left turn rates, it can only slightly turn right with the help of a controller to maintain closed-loop stability. This behavior is expected since the smaller lift on the left wing due to the decrease of left wing area causes a negative rolling moment. To counter this negative rolling moment requires a negative aileron deflection which

subsequently incurs the negative yawing moment. As a result, left turning flight is easier to trim than right turns. Compared to the limited turning capability, the aircraft has a wide range of climb rates, enabling the damaged aircraft to straightforwardly change altitude as required for landing.

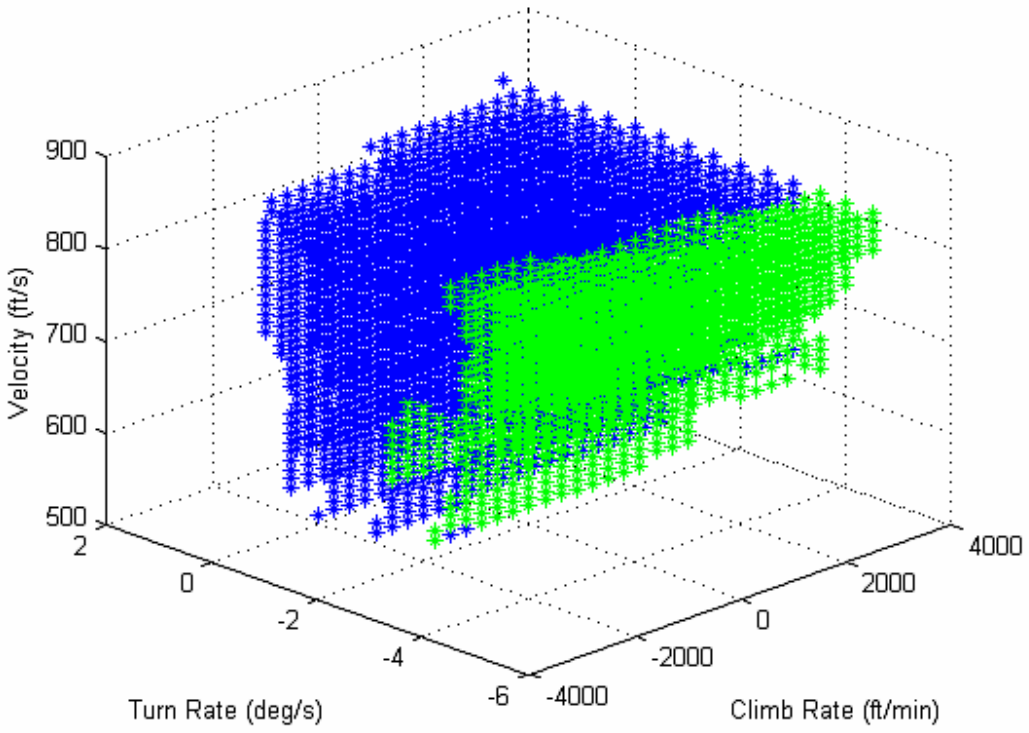

Figure 6-1: 3-D Trim states of the left wing damaged GTM at 10 ft

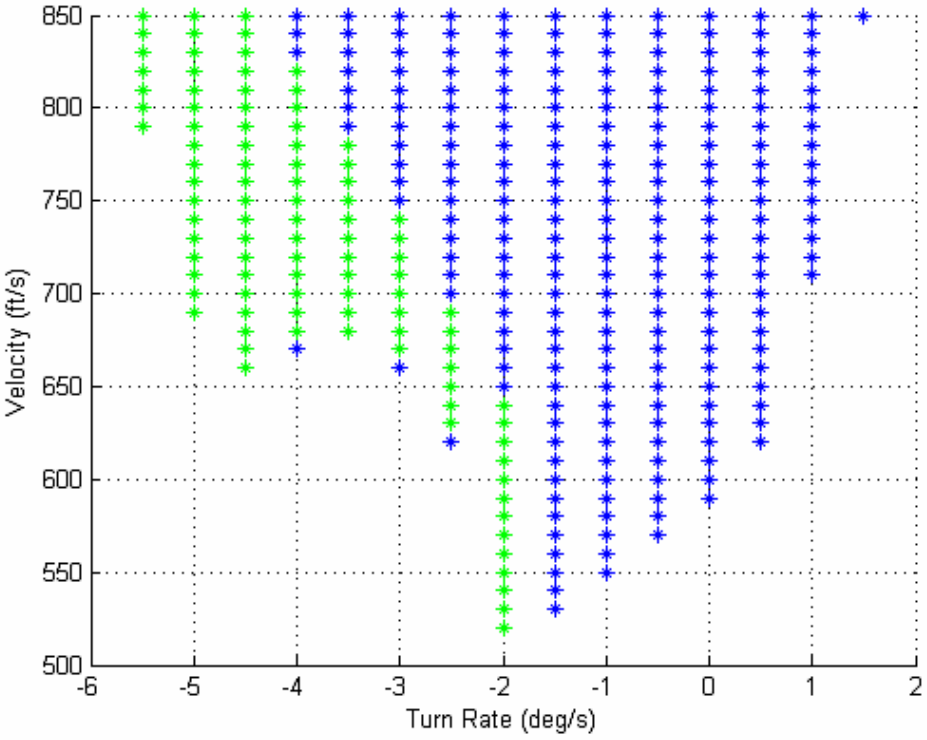

Figure 6-2: Front View of the left wing damaged GTM trim database at 10 ft

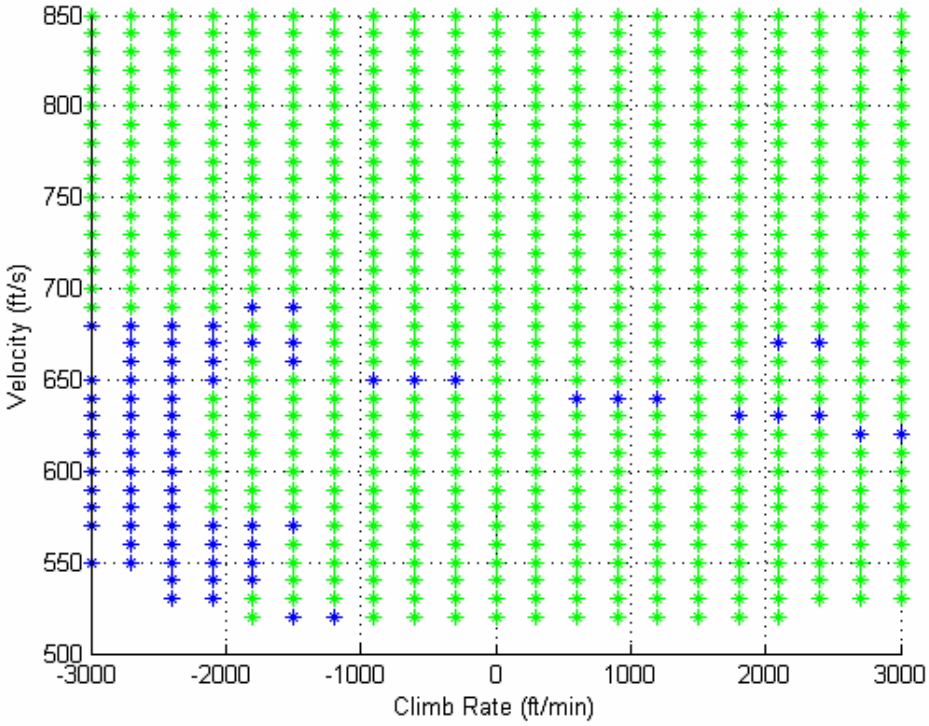

Figure 6-3: Side View of the left wing damaged GTM trim database at 10 ft

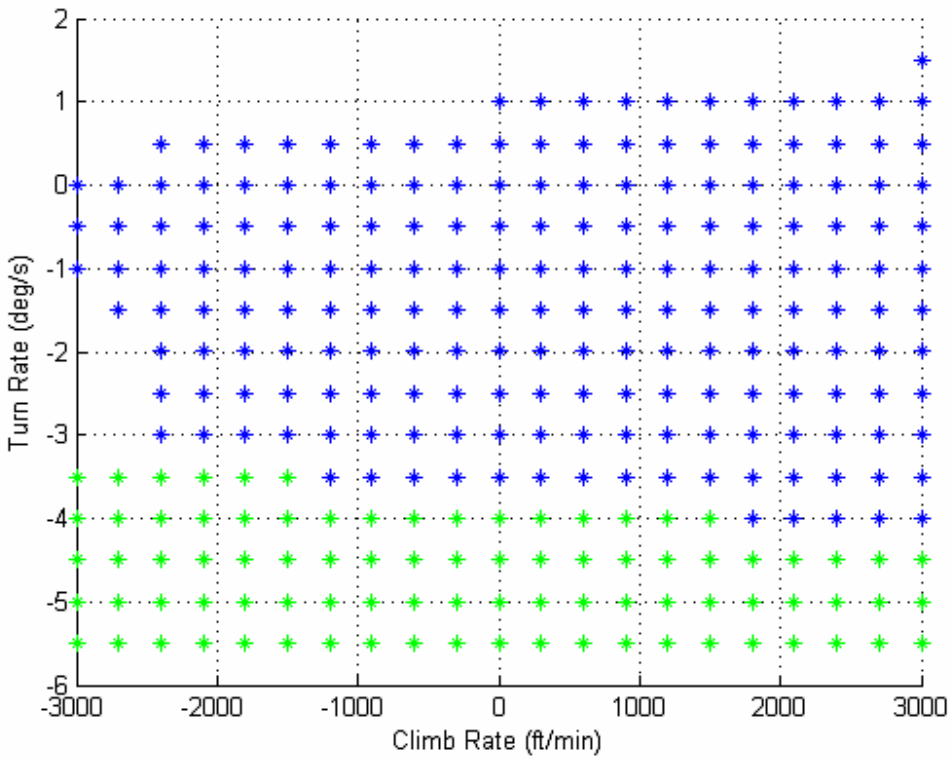

Figure 6-4: Top View of the left wing damaged GTM trim database at 10 ft

Figures 6-5 through 6-8 show the trim database at the altitude of 10,010 ft. This altitude slice shows similar dynamics characteristics but with a contracted flight envelope in all three dimensions. Higher airspeeds are required to trim the aircraft, while the trimmed turning and climbing capabilities are more limited. This contraction is more obvious as the altitude increases to 20,010 ft and 30,010 ft, as shown respectively in Figures 6-9 and 6-10. At altitude of 30,010ft, the few feasible trim points as shown are not sufficient to define a meaningful flight envelope for the damaged aircraft. By checking these altitude slices, we limit our case study examples to initial altitudes at 15,200 ft and below.

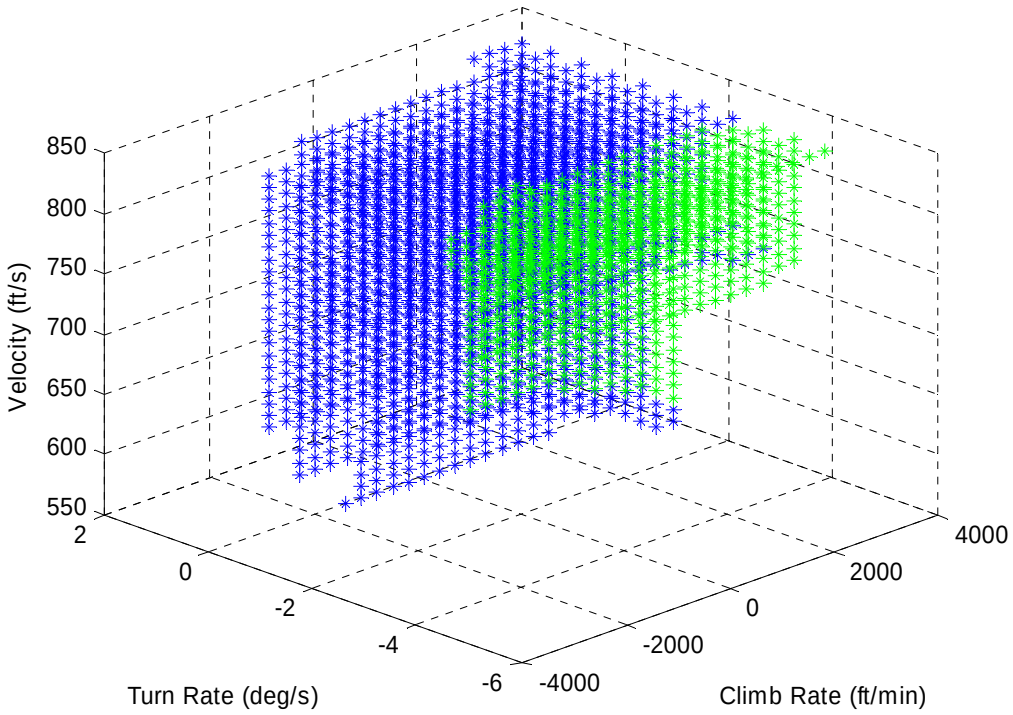

Figure 6-5: 3-D View of the left wing damaged GTM trim database at 10,010 ft

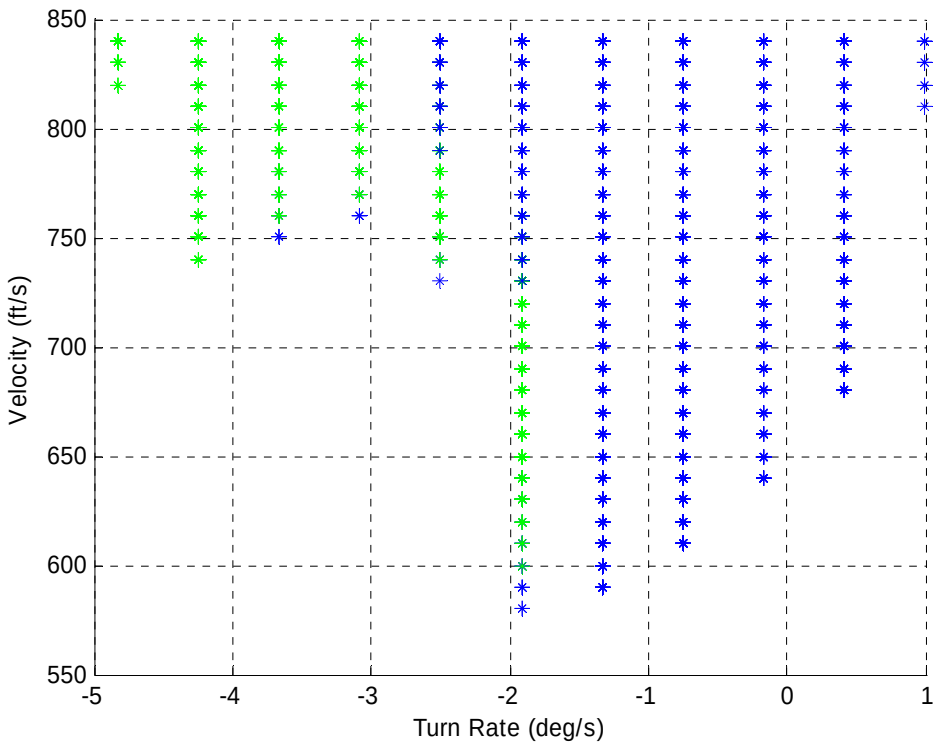

Figure 6-6: Front View of the left wing damaged GTM trim database at 10,010 ft

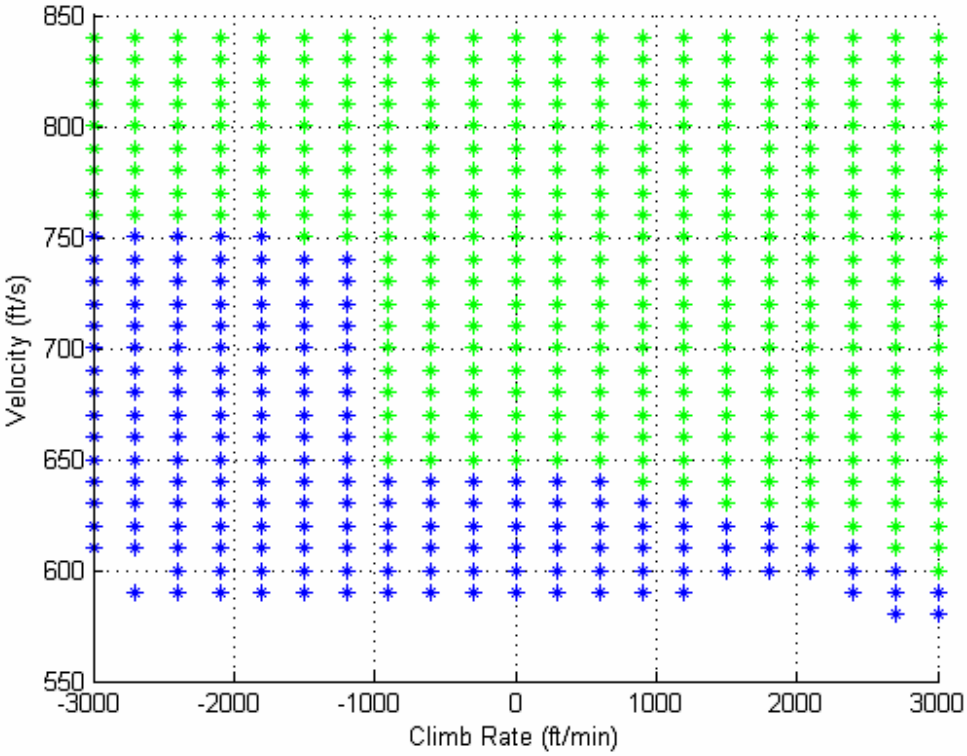

Figure 6-7: Side View of the left wing damaged GTM trim database at 10,010 ft

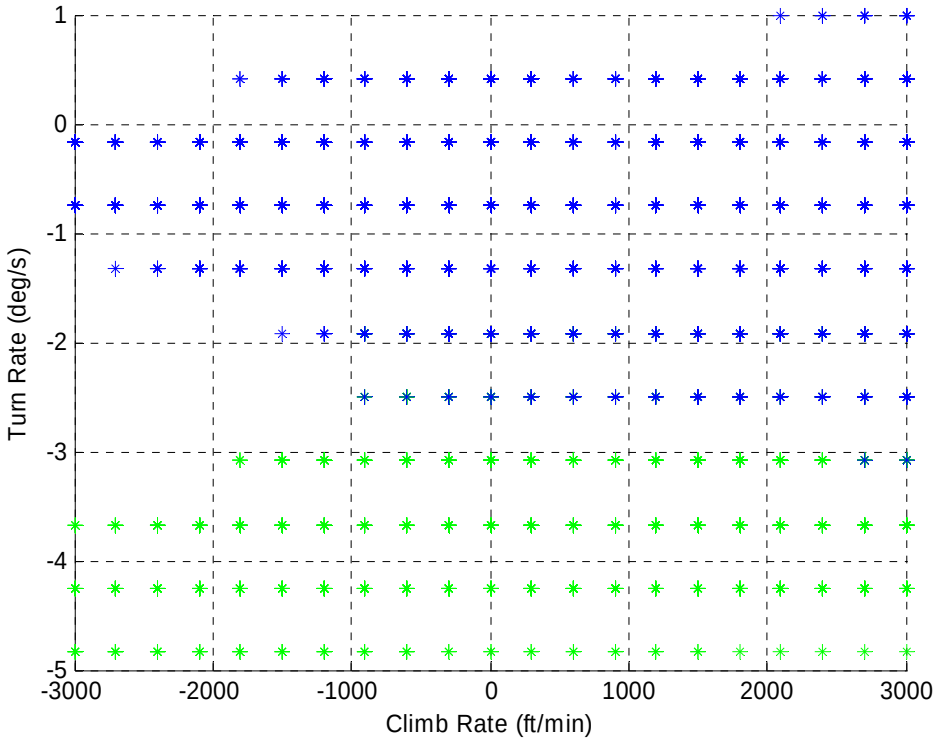

Figure 6-8: Top View of the left wing damaged GTM trim database at 10,010 ft

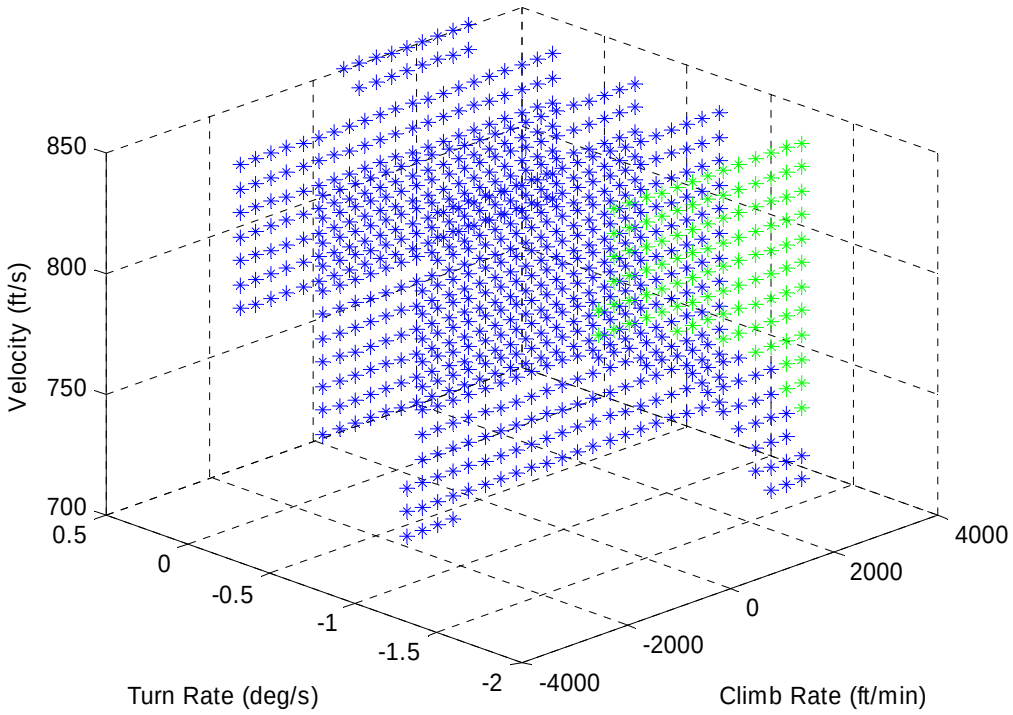

Figure 6-9: 3-D View of the left wing damaged GTM trim database at 20,010 ft

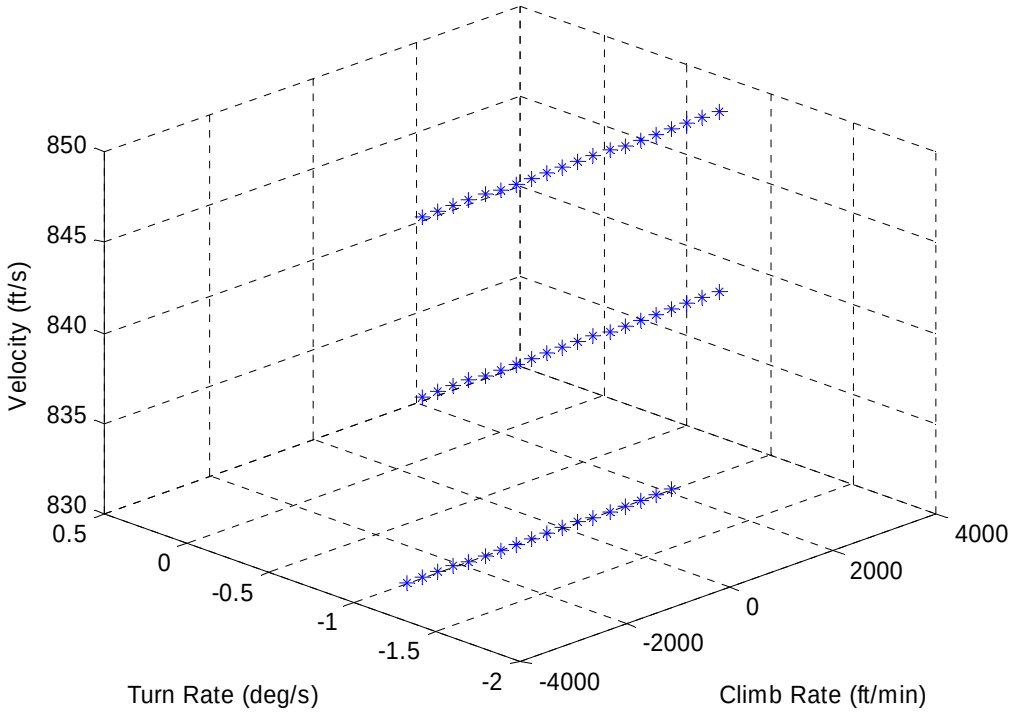

Figure 6-10: 3-D View of the left wing damaged GTM trim database at 30,010 ft

As shown in Figure 6-10, at the altitude of 30,010 ft the damaged GTM aircraft can only be trimmed with velocity of 830 ft/sec, 840 ft/sec, and 850 ft/sec, and with turn rate of -1 deg/sec. Although this thesis assumes that the initial aircraft altitude is no higher than 15,200 ft, it is very likely that the aircraft failure/damage occurs at such a high altitude. Therefore, in future work, by maintaining these few trim states the aircraft may fly down to a lower altitude where the larger trim databases would be applicable since negative flight is possible at 30,010 ft.

6.1.2 Reduced Trim and Transition Databases

Based on the full trim database, a reduced trim database $\tilde{D}$ is defined using the approach described in Section 5.1.2. Table 6-2 shows the flight condition values manually chosen as $\tilde{D}$ for the damaged GTM, representing values that can be trimmed up to altitudes of 15,200 ft.

Table 6-2: Definition of $\tilde{D}$ for the Damaged GTM Aircraft

Airspeed (ft/s)	Climb Rate (ft/s)	Turn Rate (deg/s)	No. Pts.
750	0, ±5	-2.5, -1, -0.5, 0, 0.5	15

Given the reduced trim database $\tilde{D}$, the corresponding transition database $\tilde{M}$ is computed for every thousand feet of altitude between 200 and 15,200 using the controller from Section 4.2.2 with $\Delta t = 15\,\text{sec}$ and $t_c = 20\,\text{sec}$. Transitions involving turn rate $\dot{\psi}$ transitions $-2.5 \leftrightarrow 0.5$ are infeasible for altitudes over 62,00ft since the aileron deflection transients are out of the deflection limits over these transitions. All other transitions are feasible and their kinematic effects (position and heading change) over interval $t_c + \Delta t\,\text{sec}$ are stored in $\tilde{M}$.

6.2 Emergency Scenarios

This section studies four different emergency scenarios, as well as the corresponding results from the Adaptive Flight Planner.

6.2.1 Scenario 1

In this example, the GTM aircraft is at an altitude of 14,000 ft over the San Francisco area when the damage occurs. Specifically, the initial aircraft's latitude and longitude are 37.44º and -122.12º respectively; the initial altitude is 14,000 ft MSL (mean sea level) and initial heading is 90º, a heading of due East. As discussed in previous chapters, the LSS first searches the U.S. airport database for nearby landing sites. Since the damaged aircraft must touchdown with a high airspeed of 750 ft/s, the minimum runway length and width for a feasible landing runway are set to 10,000ft and 200ft respectively. The runway utility weighting factors defined by equation (3.1) are as follows

$$\{C_1, \quad C_2, \quad ..., \quad C_8\} = \{0.4 \quad 0.4 \quad 0.05 \quad 0.01 \quad 0.05 \quad 0.03 \quad 0.03 \quad 0.03\}$$

The initial footprint radius is set to 20 nautical miles, within which the LSS finds eight feasible runways and sorts them as shown in Table 6-3. The entire LSS procedure executes in 0.241 seconds, which includes the time for data logging operations not required in a deployed LSS system.

Table 6-3: Feasible Landing Sites for Scenario 1

Rank	Airport	Runway	Utility
1	SFO	28R	0.9917
2	SFO	28L	0.9489
3	SFO	10L	0.9017
4	SFO	10R	0.8589
5	OAK	29	0.8131
6	SJC	30L	0.8107
7	SJC	12R	0.7667
8	OAK	11	0.7654

The top ranking runway, SFO/28R, is selected by the AFP as the desired landing site for the damaged GTM aircraft. The runway SFO/28R is located at 37.619002º N and 122.374843º W; its elevation is 11ft and the runway heading is 0.48692 radians. In the trajectory planner, the maximum altitude offset $\bar{h}$ described in Figure 5-3 *PLANNER_I* algorithm is set to be 2,000 ft. The trajectory planner generates an optimal flight plan, as shown in Table 6-4, where the i^{th} row describes the i^{th} trimmed flight segment. These trimmed flight segments, connected by the fixed-time kinematic position and heading changes stored in the transition database, are represented graphically in Figure 6-11.

Table 6-4: An Optimal Flight Plan for Scenario 1

Trajectory	i	$(V_T^*)_i$ (ft/sec)	$\dot{h}_i^*$ (ft/min)	$\dot{\psi}_i^*$ (deg/sec)	Δt_i (sec)
Part I	1	750	0	0	0.00
	2	750	-300	-1	2370.304
	3	750	0	-1	273.718
	4	750	0	0	72.313
Part II	5	750	-300	-1	48.481
	6	750	-300	-2.5	0.549
	7	750	-300	-1	143.510
	8	750	-300	-2.5	67.398

The first four trimmed flight segments form the landing *trajectory part I*, which is returned by *PLANNER_I* in 0.000107 seconds. The remaining segments form landing *trajectory part II*. The actual flight time along *trajectory part I* is more than 2,400 seconds. Although the 2,400 seconds duration for *trajectory part I* may be too long to be practical for an emergency landing, it is caused by the artificially-imposed 300 ft/sec descent rate that is consistently selected throughout the case study. The generation of the *trajectory part II* doesn't require such a long time. As shown in figures 6-1 through 6-10, the damaged GTM aircraft is capable of faster climb and descent rates which should be examined in future work. In this case, the duration over *trajectory part I* is sufficient for the *PLANNER_II* to generate a feasible plan for *trajectory part II* during the flight along *trajectory part I*. In fact, the optimal flight

plan shown in Table 6-3 was completed by *PLANNER_II* 4.4023 seconds after it is activated, well within the limit imposed by execution of *trajectory part I*.

A full path simulation was conducted to validate the optimal flight plan. In simulation, the damaged GTM aircraft is flown by the controller used to achieve the trim transitions, as described in Section 4.2.2. This is an ideal situation that serves to validate computed solutions given no disturbances or modeling discrepancies.

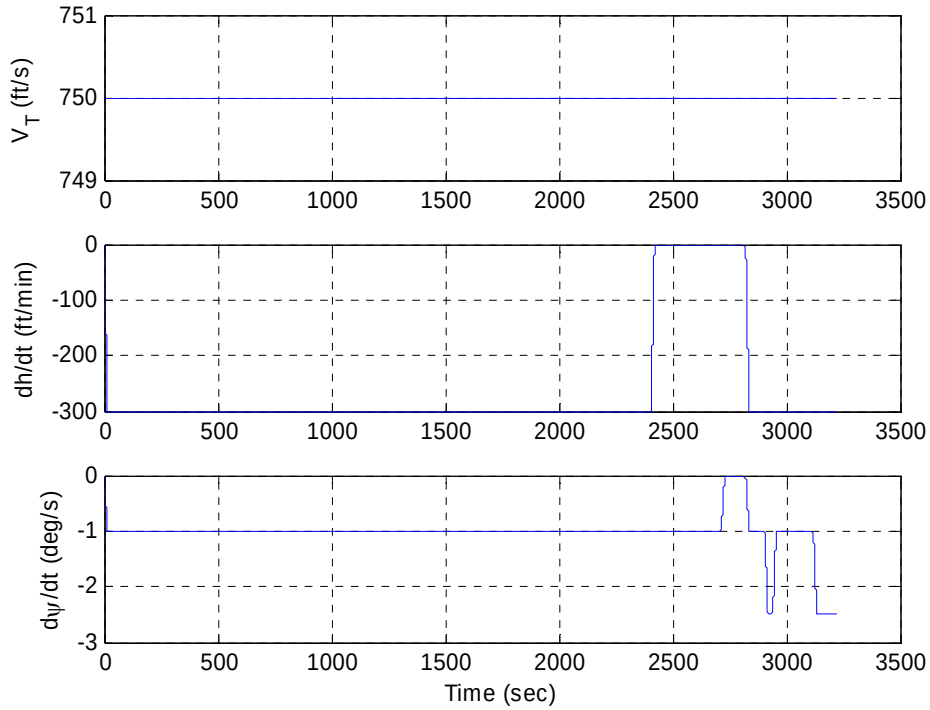

Figure 6-11: Optimal Flight Plan for Scenario 1

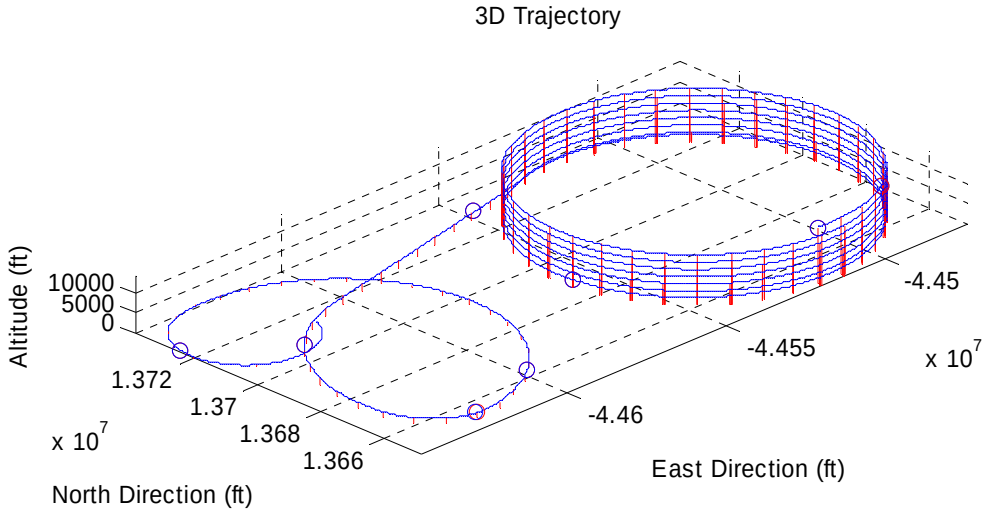

Figure 6-12: Solution Trajectory for Scenario 1

Simulation results are shown in Figures 6-12 through 6-15. Figure 6-12 is the trajectory generated by the controlled aircraft following the optimal flight plan. After a 3221.2-second flight, the final touchdown errors are 65.00 ft South, 14.67 ft West, 0.655 ft in altitude, and 0.0508 deg in heading. Figure 6-13 shows flight tracking performance. The aircraft tracks the turn rate commands better than it tracks the airspeed and climb rate commands. However, tracking errors primarily occur during the trim transitions, thus their impact on the kinematic information has been taken into account within the transition database.

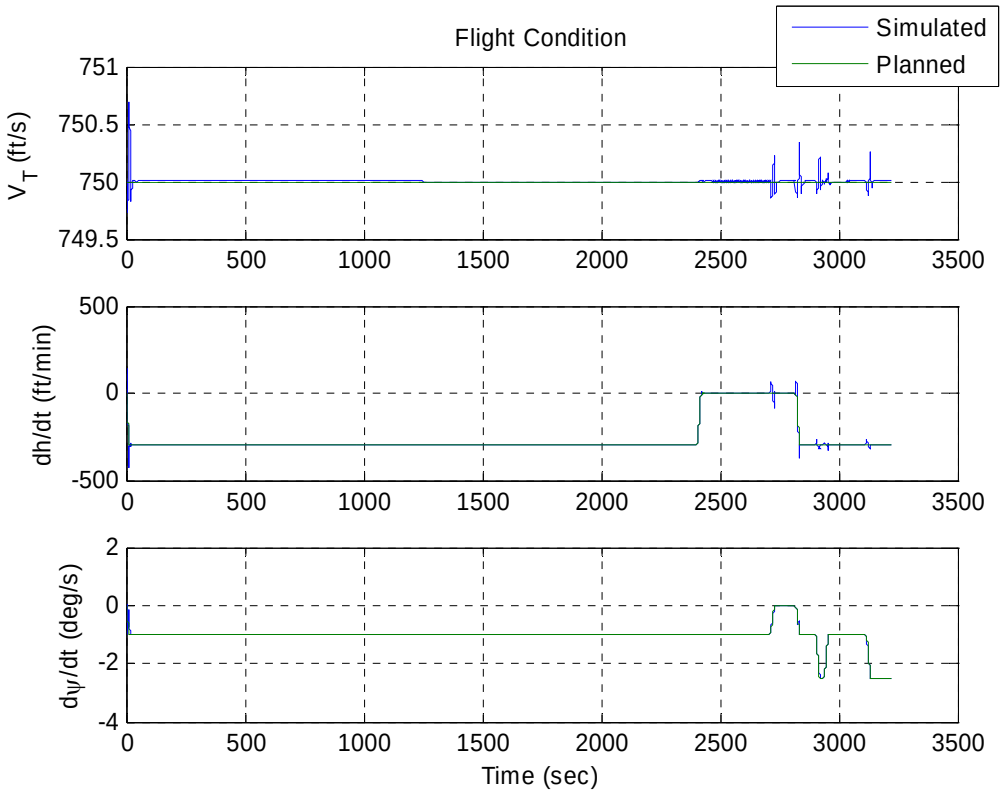

Figure 6-13: Flight Condition Tracking Performance for Scenario 1

Figure 6-14 shows aircraft orientation tracking performance. Again, the relatively larger tracking errors occur during the trim transitions. However, angle of attack and sideslip angle fall well within the valid data range. The trimmed roll angle value along the $\dot{\psi} = -2.5\,\mathrm{deg/sec}$ trimmed flight segments is large. While this value may exceed the limit for a normal commercial airplane, it may be acceptable when a safe landing is required in emergency situations.

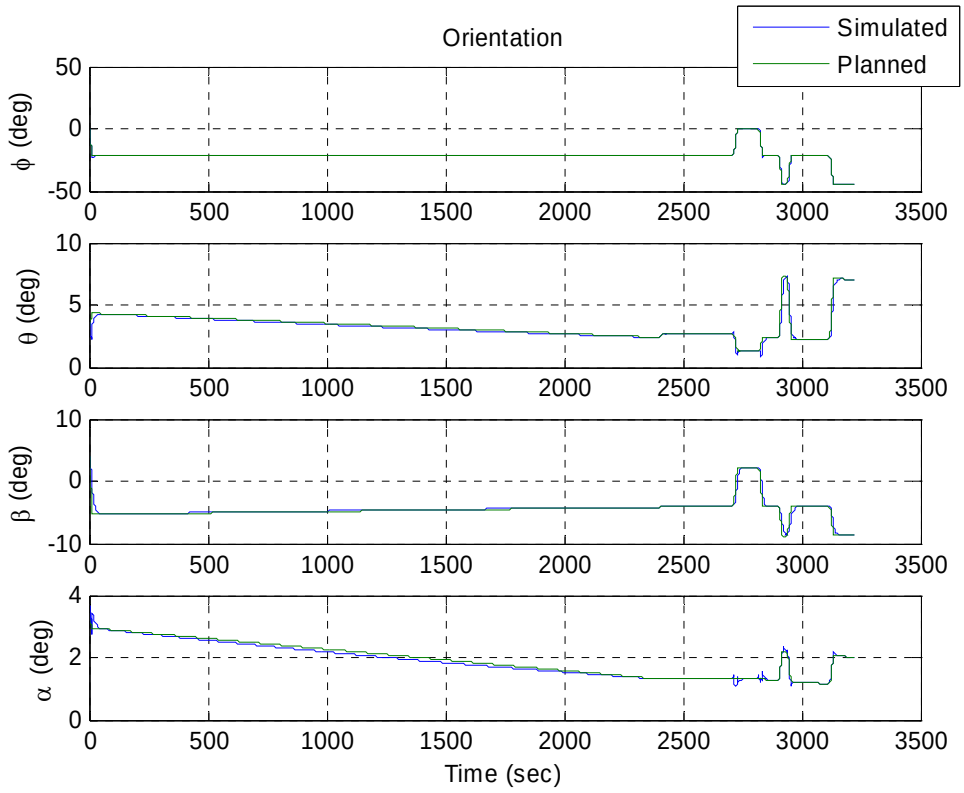

Figure 6-14: Orientation Information for Scenario 1

Figure 6-15 shows the controller is able to track the desired trajectory without violating actuator saturation constraints. During the 1st trimmed flight segment, a spiral descent, there are some tracking errors for the control inputs. This is reasonable since the "planned" control inputs in the figure are actually the "ideal" control linearly interpolated between two terminal trimmed control settings of a trimmed flight segment. The first trimmed flight segment involves a significant variation of atmospheric density due to the change of altitude, thus the actual ideal control inputs over this segment are a nonlinear function of time. For the segments that don't involve altitude change, the desired controls are constant over the segment. The tracking angle errors over the first trimmed flight segment seen in Figure 6-14 can be explained similarly.

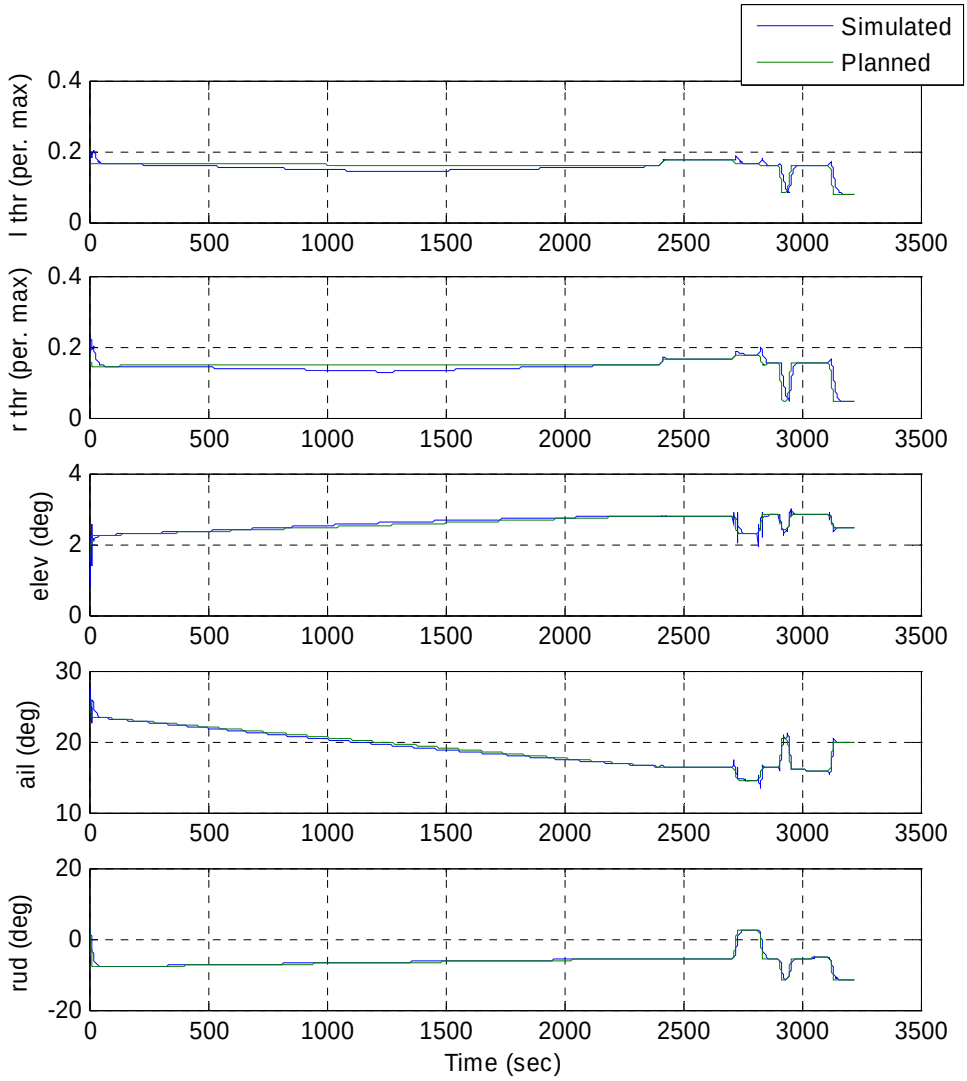

Figure 6-15: Controller Information for Scenario 1

6.2.2 Scenario 2

In this scenario, the GTM is also over the San Francisco area when the emergency occurs. However, unlike in Scenario 1, the aircraft's initial location is near the airport, and the initial altitude is 200 ft, much lower than in the previous scenario. The aircraft's initial latitude and longitude are 37.64º and -122.38º respectively. The initial aircraft heading is 90º (due East).

In the LSS, the minimum runway requirements and the feasible runway utility weighting factors are the same values as in Scenario 1. Within the initial 20 nm footprint region, the LSS finds the same sorted list of feasible runways as in Scenario 1 in 0.177 seconds. The same landing site, namely runway SFO/28R, is selected as the desired landing site for the damaged GTM aircraft. The trajectory planner then generates an optimal flight plan, as shown in Table 6-5 and Figure 6-16, which connects the initial state with runway SFO/28R.

Since the aircraft is initially at a very low altitude, the planner *PLANNER_I* generates the landing *trajectory part I* as a spiral-up trajectory in 0.028 seconds. Here, the minimum altitude offset $\underline{h}$ described in Figure 5-3 *PLANNER_I* algorithm is set to be 1,000 ft. Note that the duration of the third flight segment, a straight level flight, is zero since the initial point of this segment is sufficiently close to the desired landing site. It will take the aircraft approximately 400 seconds to complete *trajectory part I* by following the first three trimmed flight segments. In the meantime, *PLANNER_II* generates the remaining flight plan, *trajectory part II*, in less than 12 seconds. Therefore, the generation of the entire optimal flight plan meets the real-time requirements.

Table 6-5: Optimal Flight Plan for Scenario 2

Trajectory	*i*	$(V_T^*)_i$ (ft/sec)	$\dot{h}_i^*$ (ft/min)	$\dot{\psi}_i^*$ (deg/sec)	Δt_i (sec)
Part I	0	750	0	0	0.00
	1	750	300	-2.5	334.700
	2	750	0	-2.5	4.400
	3	750	0	0	0.000
Part II	4	750	-300	0	55.066
	5	750	-300	-2.5	54.386
	6	750	-300	-0.5	69.754
	7	750	-300	-2.5	95.783

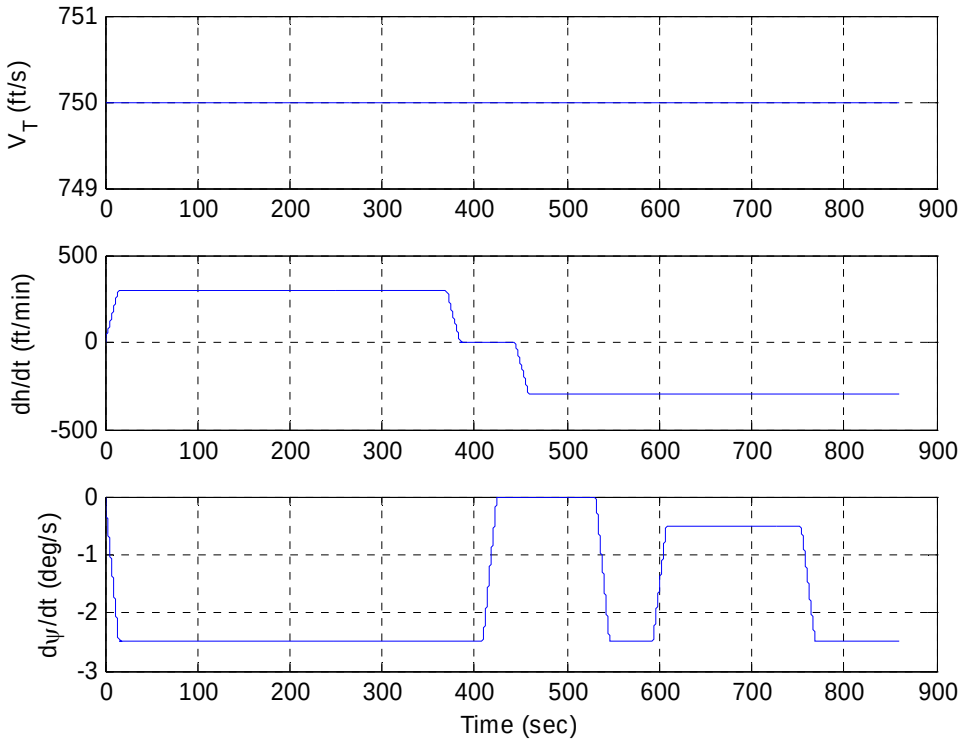

Figure 6-16: Optimal Flight Plan for Scenario 2

The results from the full path simulation are shown in Figures 6-17 through 6-20. Figure 6-17 shows the simulated 3-D trajectory which yields final touchdown errors of 60.96 ft South, 8.36 ft West, 0.50 ft in altitude, and -0.0004 deg in heading. In Figures 6-18 and 6-19, tracking errors occurring over the trim transitions converge to zero during the subsequent trimmed flight segments. Figure 6-20 shows the control efforts required for the aircraft to follow the planned trajectory. All control values satisfy saturation constraints, although the significant transients during trim transitions may require the unrealistically high-speed response from the actuators.

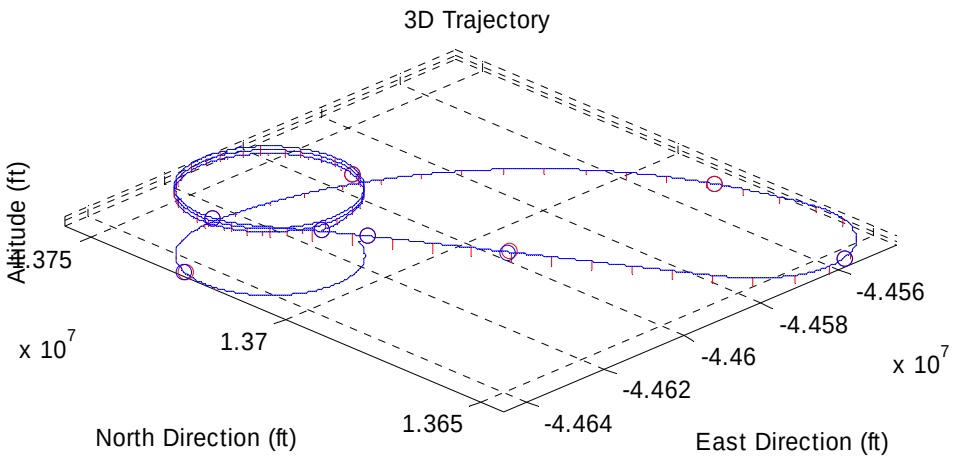

Figure 6-17: Solution Trajectory for Scenario 2

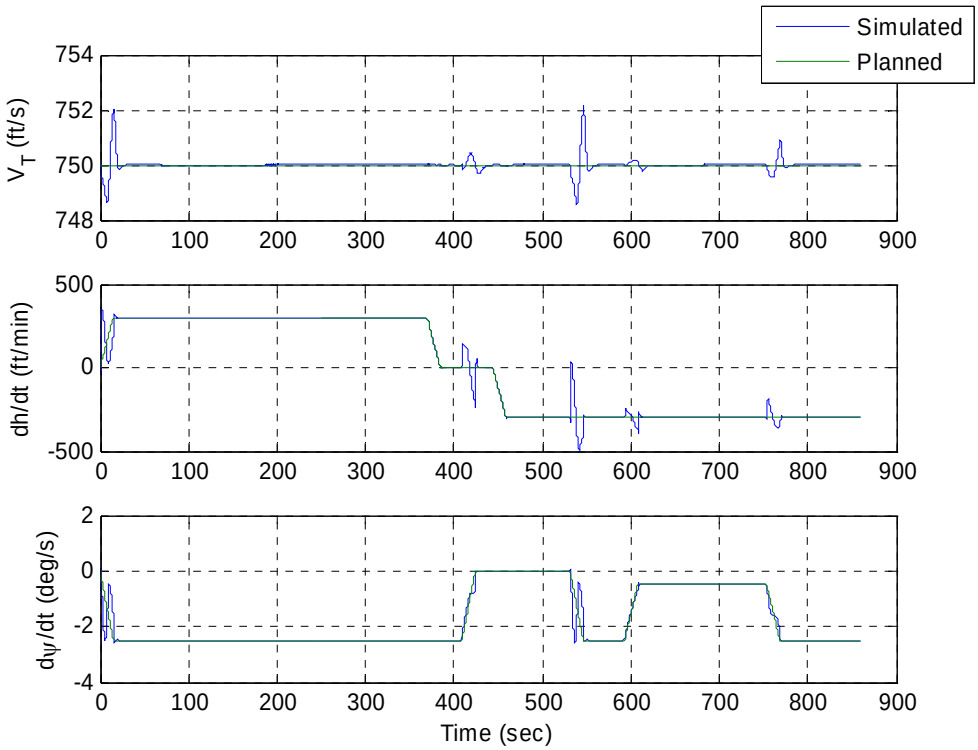

Figure 6-18: Flight Condition Tracking Performance for Scenario 2

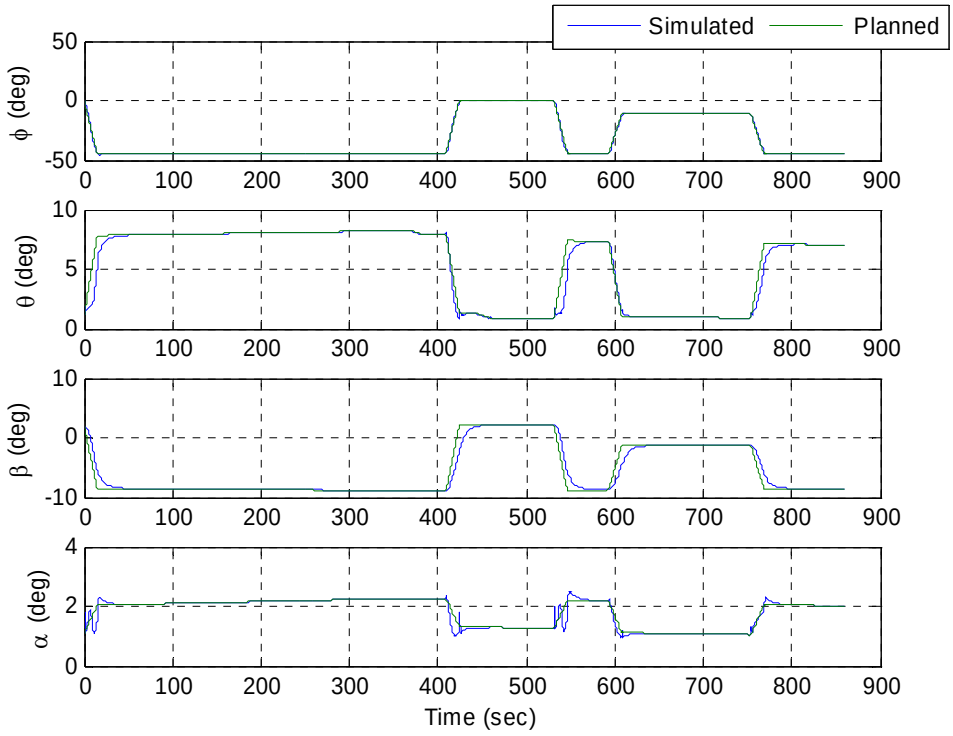

Figure 6-19: Orientation Information for Scenario 2

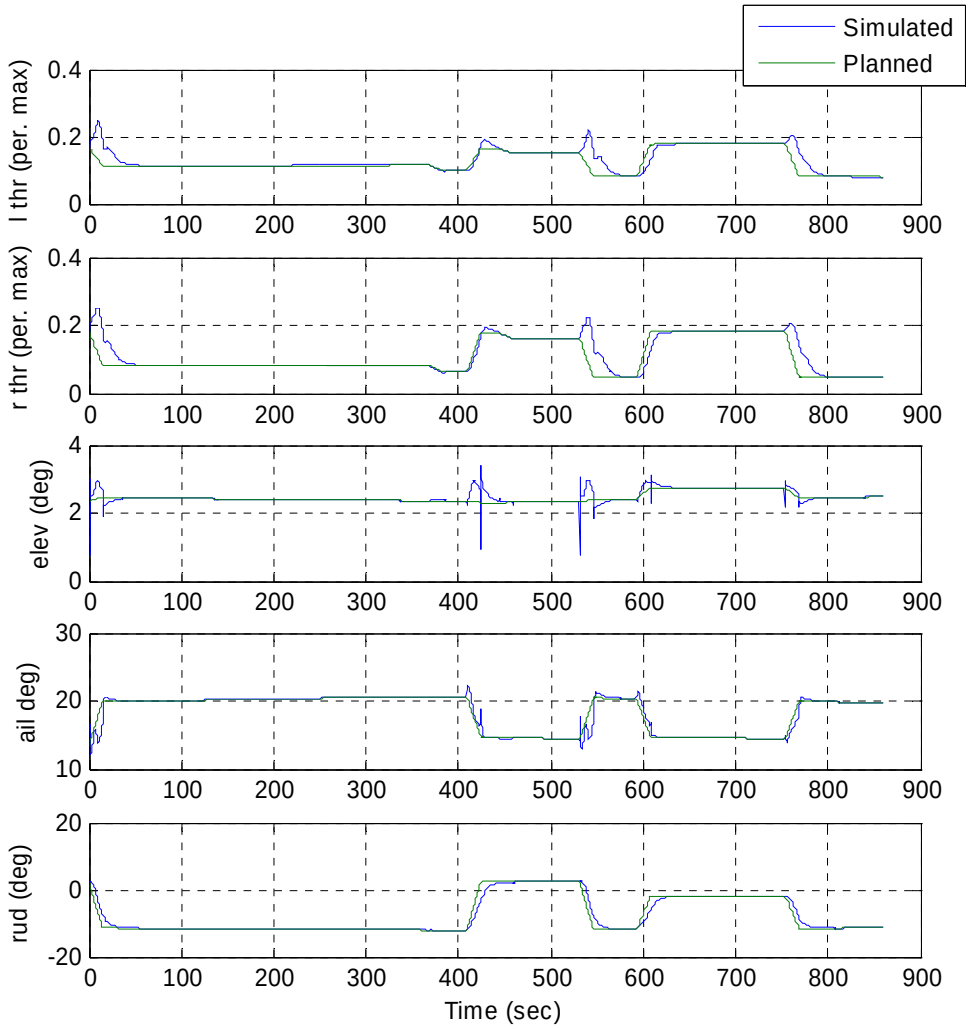

Figure 6-20: Controller Information for Scenario 2

To provide comparison between the "optimal" plan and other candidate plans in *L* found by the trajectory planner, full path simulations were also conducted by following the next three suboptimal flight plans in *L*. Figures 6-21 through 6-23 show these simulated trajectories. These trajectories yield similar final touchdown errors, thus they can be considered feasible for the aircraft.

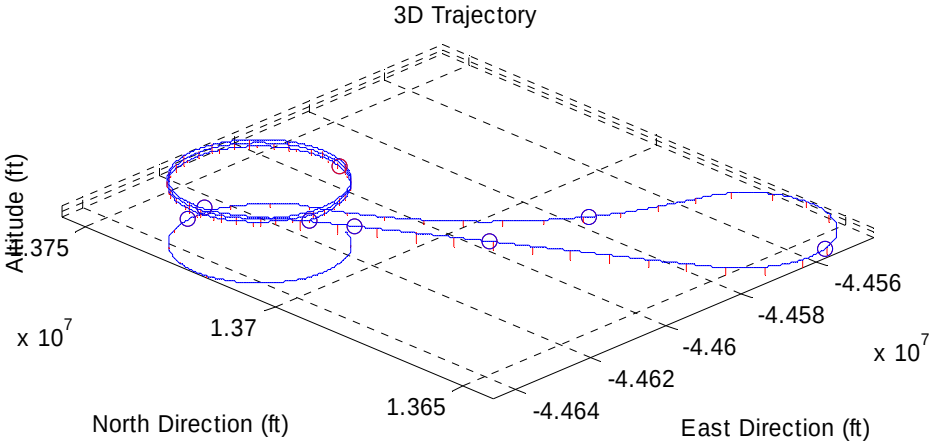

Figure 6-21: Second Solution Trajectory for Scenario 2

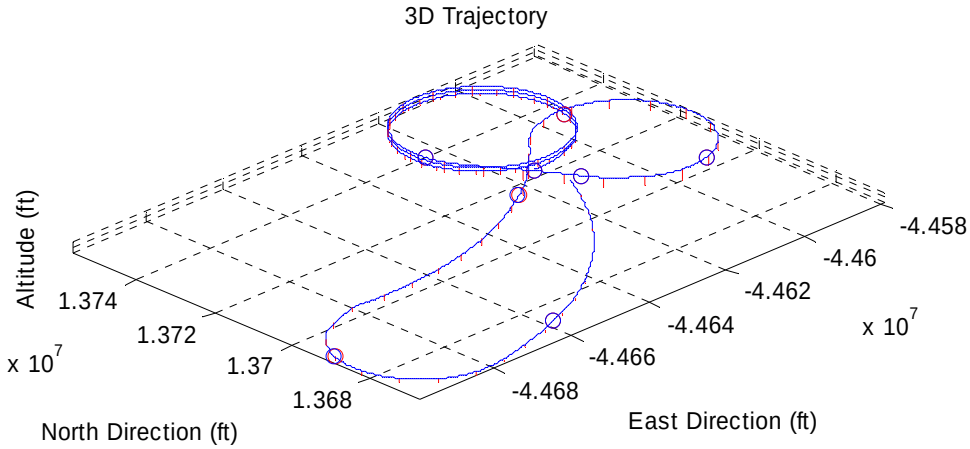

Figure 6-22: Third Solution Trajectory for Scnario 2

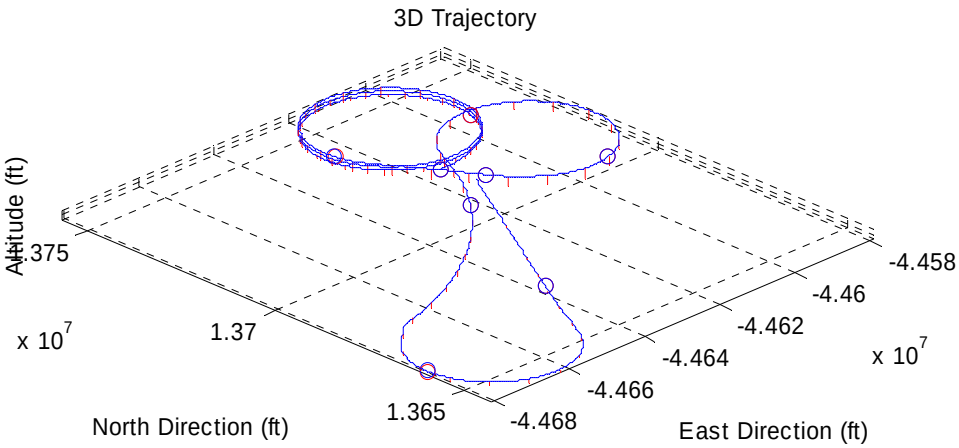

Figure 6-23: Fourth Solution Trajectory for Scenario 2

6.2.3 Scenario 3

In this scenario, the GTM aircraft is over a remote area when the damage occurs. The initial aircraft location is at 40.89º N and 94.01º W, which is in Iowa. The aircraft's initial altitude is 10,000 ft and initial heading is 210º (South-South-West). Since the damaged GTM aircraft is initially located in a remote area, identification of a feasible runway requires adjustment to the LSS parameters. The minimum runway width requirement is relaxed to be 150 ft. Other requirements are the same as for the previous scenarios. For the same reason, the initial footprint radius is manually set to be 50 nautical miles instead of 20 nautical miles used in the previous scenarios, and the radius increment is also adjusted to 100 nautical miles. In future work, this manual adjustment may be made by the software automatically without much more effort. Although 82 reachable airport runways are found within the initial footprint,

none of them meet the minimum feasible runway requirements. Therefore, the LSS increases the footprint radius to 150 nautical miles, and then finds 1028 reachable airport runways, out of which 6 feasible runways are identified. Based on the same runway utilities weighting factors used previously, these feasible runways are sorted and returned in a list, as shown in Table 6-6. The entire LSS procedure is completed in 0.383 seconds.

Table 6-6: Feasible Landing Sites for Scenario 3

Rank	Airport	Runway	Untility
1	OFF	30	0.9199
2	OFF	12	0.8699
3	SZL	19	0.7553
4	SZL	1	0.7452
5	MCI	19R	0.6759
6	MCI	1L	0.6659

The top ranking runway, OFF/30, is selected by the AFP as the desired landing site for the damaged GTM aircraft. The runway OFF/30 is located at 41.118332º N and 95.912511º W; its elevation is 1,048 ft and the runway heading is 0.52360 radians. The trajectory planner generates an optimal flight plan to runway OFF/30, as shown in Table 6-7 and Figure 6-24. The first four flight segments form *trajectory part I*, generated by *PLANNER_I* in 0.000071 seconds. The other four segments comprise *trajectory part II*. As the aircraft executes the flight plan part for trajectory part I, *PLANNER_II* builds the remaining flight segments to complete the plan. While

trajectory part I will take the aircraft over 2,400 seconds, PLANNER_II completes the entire plan in 6.492 seconds. Thus, the real-time requirement is met.

Table 6-7: Optimal Flight Plan for Scenario 3

Trajecotyr	i	$(V_T^*)_i$ (ft/sec)	$\dot{h}_i^*$ (ft/min)	$\dot{\psi}_i^*$ (deg/sec)	Δt_i (sec)
Part I	0	750	0	0	0.00
	1	750	-300	-1	1362.901
	2	750	0	-1	304.326
	3	750	0	0	880.538
Part II	4	750	-300	-0.5	64.789
	5	750	0	-2.5	15.460
	6	750	-300	-1	168.334
	7	750	-300	-2.5	61.867

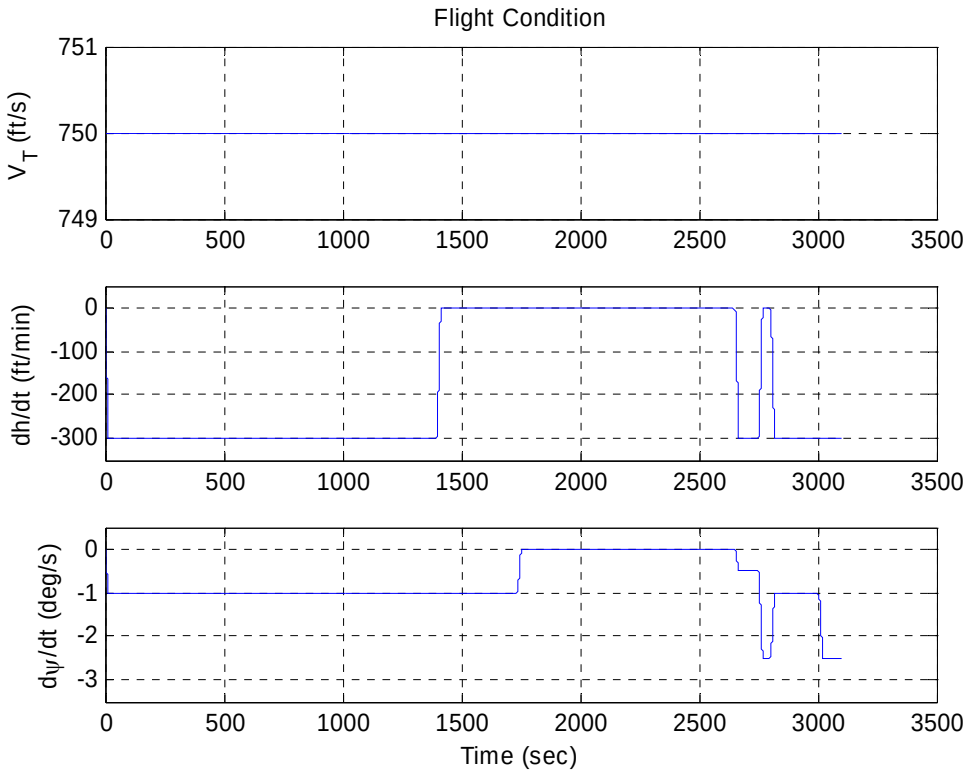

Figure 6-24: Optimal Flight Plan for Scenario 3

Full path simulation results are shown in Figures 6-25 through 6-28. The simulated 3-D trajectory shown in Figure 6-25 results in final touchdown errors of 41.51 ft South, 3.484 ft East, 0.125 ft in altitude, and -0.0144 deg in heading. Figure 6-26 shows that the actual flight conditions track their targets closely with some transients over the trim transitions. Heading angle tracking shows similar performance in Figure 6-27. The controller responses over the entire trajectory are shown in Figure 6-28. As with the previous examples, control inputs fall well within saturation limits while tracking the commands closely.

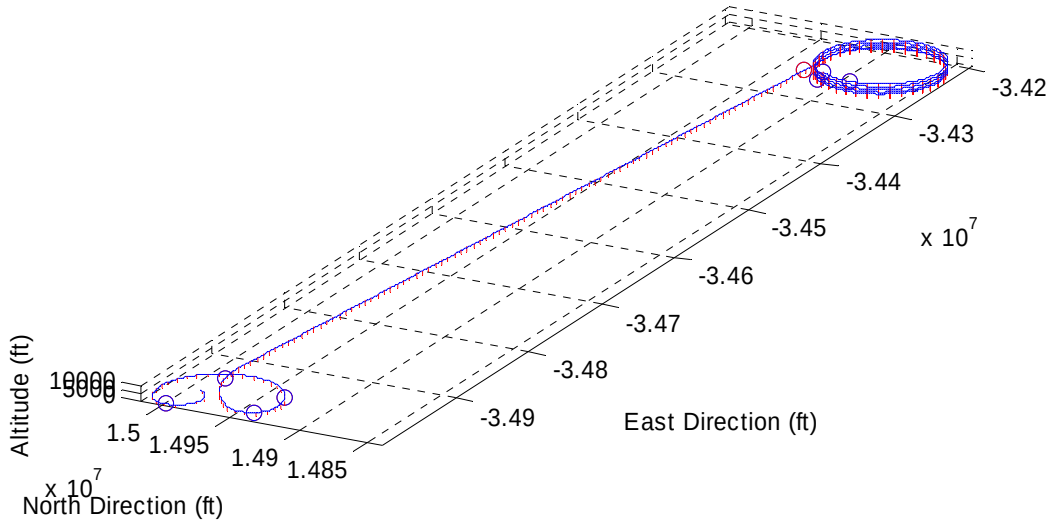

Figure 6-25: Solution Trajectory for Scenario 3

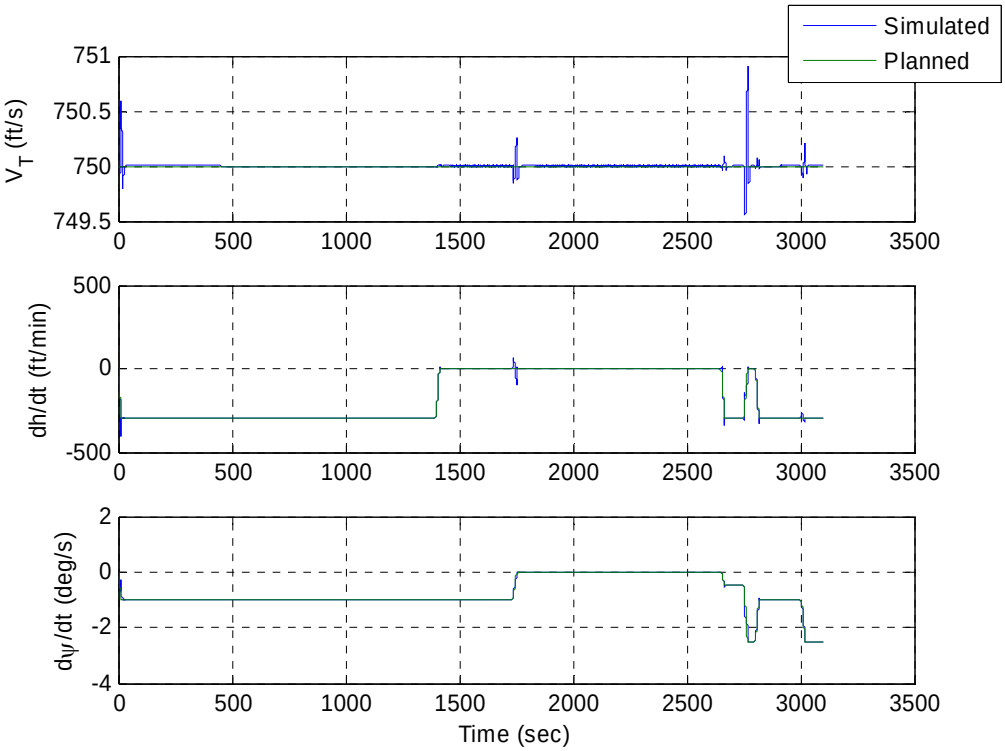

Figure 6-26: Flight Condition Tracking Performance for Scenario 2

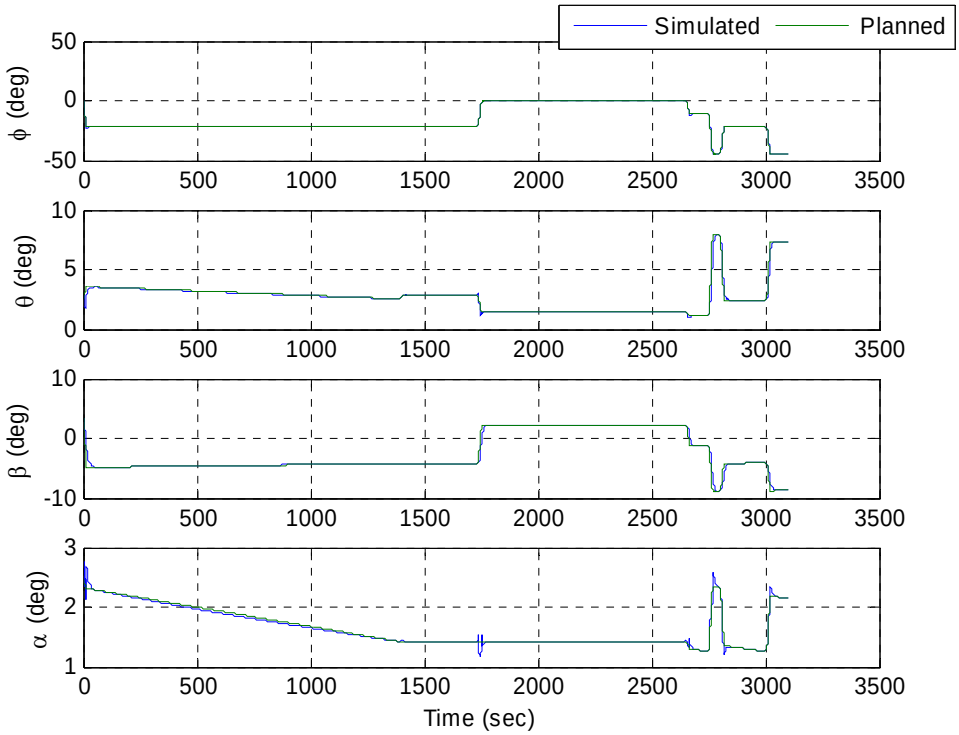

Figure 6-27: Orientation Information for Scenario 2

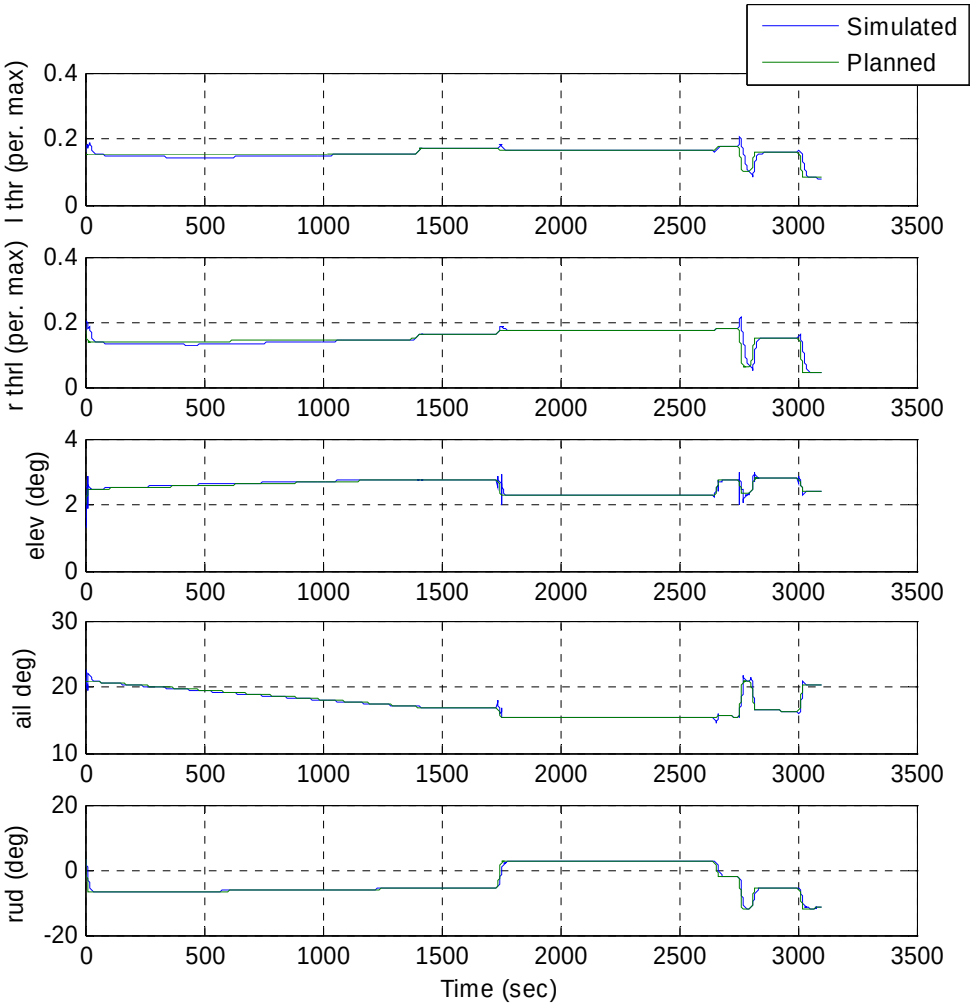

Figure 6-28: Controller Information for Scenario 3

Figures 6-29 and 6-30 present simulated 3-D trajectories obtained by executing the next two suboptimal flight plans in *L*. Due to use of the same flight plan for *trajectory part I*, the first segments of these trajectories are identical to their counterparts in the optimal solution. The second segments of these trajectories involve 0.5 deg/sec right turn trimmed flight, which is at the boundary of the flight envelope, as mentioned previously.

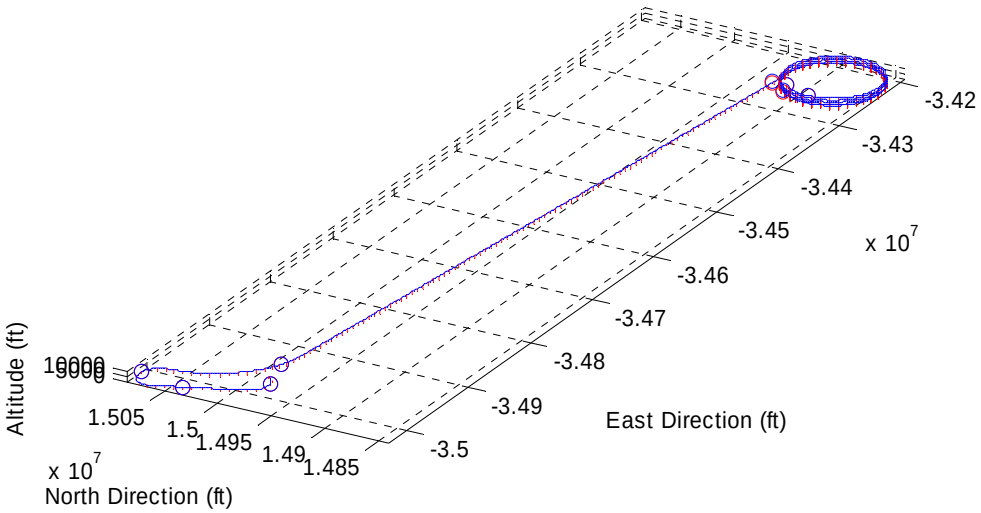

Figure 6-29: Second Solution Trajectory for Scenario 3

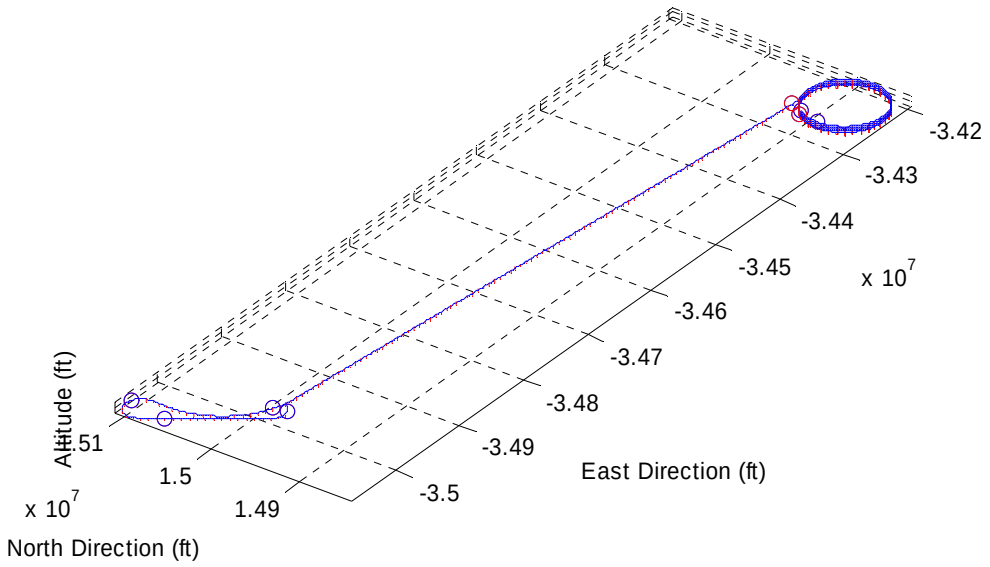

Figure 6-30: Third Solution Trajectory for Scenario 3.

6.2.4 Scenario 4

In the previous examples, the trajectory planner was capable of generating feasible solutions for *trajectory part II* before the aircraft finishes executing the planned *trajectory part I*. This is primarily because the aircraft was assumed able to perform a very steep turning flight with a turn rate $\dot{\psi} = -2.5\,\text{deg/sec}$. A -2.5 deg/sec trimmed turn flight results in a large aircraft roll angle of -44.62 deg. While these values may be acceptable when a safe landing is required in emergency situations, it is useful to examine how the trajectory planner would perform if a more practical constraint were placed on aircraft turn rate.

Practically, the nominal GTM aircraft's FMS does not allow the aircraft roll angle to exceed $\pm 30^{o}$. While this limit was not represented in the first three examples, it is required to be met in this final scenario. Therefore, a trimmed left turn with turn rate equal to -2.5 deg/sec is no longer used in this example. By manually checking the roll angle values accompanying the turn rate, we find the aircraft roll angle is approximately -27.68 deg when the damaged GTM performs a -1.3 deg/sec trimmed turn flight. Thus, -1.3 deg/sec is selected as the maximum value for the turn rate, and the reduced trim database $\tilde{D}$ must be redefined for this scenario.

Table 6-8 shows the flight condition values manually chosen as the new $\tilde{D}$ for the damaged GTM, representing values that can be trimmed up to altitudes of 15,200 ft.

Table 6-8: Definition of $\tilde{D}$ for the Damaged GTM Aircraft for Scenario 4

Airspeed (ft/s)	Climb Rate (ft/s)	Turn Rate (deg/s)	No. Pts.
750	0, ±5	-1.3, -0.5, 0, 0.5	12

Given the reduced trim database $\tilde{D}$, the corresponding transition database $\tilde{M}$ is recomputed for every thousand feet of altitude between 200 and 15,200 using the controller from Section 4.2.2 with $\Delta t = 15\,\text{sec}$ and $t_c = 20\,\text{sec}$. All the transitions are feasible and their kinematic effects (position and heading change) over interval $t_c + \Delta t\,\text{sec}$ are stored in $\tilde{M}$. These two databases are used in the following case study.

In this final scenario, the GTM is again over the San Francisco area when the emergency occurs. The initial aircraft altitude is 2,500 ft MSL. The aircraft's initial

latitude and longitude are 37.64º and -122.38º respectively. The initial aircraft heading is 90º (due East).

In the LSS, the minimum runway requirements and the feasible runway utility weighting factors are the same values as in Scenarios 1 and 2. Within the initial 20 nm footprint region, the LSS finds the same sorted list of feasible runways as in Scenarios 1 and 2 in 0.235 seconds. The same landing site, namely runway SFO/28R, is selected as the desired landing site for the damaged GTM aircraft. The trajectory planner quickly generates landing *trajectory part I*, requiring the damaged GTM aircraft 571 seconds to complete. Thus, the trajectory planner has 571 seconds to plan landing *trajectory part II*. In this case, the trajectory planner exhaustively explores the search space in 205.66 seconds, but fails to provide any solutions for *trajectory part II*.

Unlike in scenario 1, without the -2.5 deg/sec turning ability of the damaged GTM, the second trajectory planner *PLANNER_II* fails to find a solution for *trajectory part II* after PLANNER_I places the initial state of *trajectory part II* (p_0^{II}, ψ_0^{II}) too close to the desired landing site (p_{des}, ψ_{des}), which is the situation described in Chapter 5. Following the discussion in Chapter 5, we try to avoid this problem by eliminating the third flight segment s_3^I of *trajectory part I,* which is a trimmed straight-line level flight toward the desired landing site, and executing the entire trajectory planning procedure once again.

This time, *PLANNER_I* plans a two-segment *trajectory part I* that only kills the extra aircraft altitude offset without further guiding the aircraft toward the airport. The aircraft needs approximately 260 seconds to fly to (p_0^{II}, ψ_0^{II}) by following *trajectory part I* plan. Although *PLANNER_II* exhaustively explores the same search space in 248.42 seconds, it still cannot find any solutions for *trajectory part II*. This result is not surprising since the damage occurs very near SFO.

The failure of the trajectory planners (both with and without guiding the aircraft toward the airport) suggests a concrete topic for the future research work; that is, how *PLANNER_I* should appropriately place *PLANNER_II*'s initial state (p_0^{II}, ψ_0^{II}) with respect to the desired landing site (p_{des}, ψ_{des}). There are two aspects to be considered when solving this problem. First, as discussed in Chapter 5, an automatic approach for determining the 2-D distance *r* between p_0^{II} and p_{des} will be needed by *PLANNER_I* if a trimmed straight and level flight toward the airport is necessary for the emergency scenario (for example, scenario 3), since a (p_0^{II}, ψ_0^{II}) that is too close to p_{des}, ψ_{des} is actually detrimental to the search performed by *PLANNER_II* when aircraft turning ability is constrained by the practical FMS settings. Second, in situations where the damage occurs very near the desired landing site, PLANNER_I may actually need to guide the aircraft away from the airport with the last flight segment of *trajectory part I*. Thus, *PLANNER_II* will have sufficient space for using shallow turning flights to assemble feasible solutions for *trajectory part II*. While

these discussions provide a possible approach to this problem, this augmentation to the trajectory planner is left for future work beyond this thesis.

Chapter 7: Conclusions and Future Work

7.1 Conclusions

This thesis has outlined a set of general methods for implementing an end-to-end Adaptive Flight Planner (AFP) architecture for emergency flight planning. The primary contribution of this work is application of the AFP to the challenging emergency situation in which a Generalized Transport Model (GTM) aircraft is damaged (loses a significant fraction of its left wingtip) during flight.

To construct each landing trajectory as a sequence of intuitive constant-trim segments, a trim database was developed to define the flight envelope of the post-damage aircraft, and transitions between the valid trim states were characterized in simulation using a controller tuned to handle the post-damage aircraft dynamics. Based on the reduced aircraft dynamics, the Landing Site Search module defined a footprint region for the disabled aircraft and identified the most desirable runway within this footprint as the emergency landing site. Using the trim database and the corresponding transition database, the two-step trajectory planner generated the feasible landing flight plans as a sequence of trimmed flight sequences in real-time. An LQR-based nonlinear PID controller was developed to enable the damaged aircraft to correctly track the trajectory commands over both the trimmed flight and trim transition segments.

The success of the adaptive flight planner requires the aircraft's initial flight condition falls well within the post-damage flight envelope, which is assumed to be sufficiently large for the aircraft to perform necessary landing maneuvers. Furthermore, the AFP is only applicable in cases where the trim and transition databases that define the dynamics boundary and flight envelope for the damaged aircraft have been defined, since their comprehensive definition currently requires numerous optimizations and simulations that preclude real-time database definition.

In the damaged GTM case study, different scenarios were presented to examine the ability of the damaged aircraft to build plans with the AFP and successfully execute these plans in simulation. In the first three scenarios, the damaged GTM was assumed capable of steep turns, while in last scenario a practical 30 degree bank constraint was imposed to reflect current FMS constraints. For all scenarios, the LSS efficiently identified a nearby appropriate landing site by examining reachable runways, while feasible landing trajectories could be generated and executed only for the first three scenarios. The failure of the trajectory planning in the last scenario suggests future work to augment the current *PLANNER_I* so that it better directs the aircraft to an intermediate state relative to the landing site that *PLANNER_II* can handle.

7.2 Future Work

The incorporation of the damaged GTM model and the full integration of all AFP modules are important contributions to solve the emergency flight planning problems

in a relatively realistic context. However, future research is required for the Adaptive Flight Planner to be deployed in practice.

In this thesis, the Landing Site Search module must return at least one feasible landing runway before emergency trajectory planning can be performed. It is possible that there are no feasible runways within the reachable region of the post-failure/damage aircraft. Future work is required to enable the LSS to identify a feasible off-runway landing site through use of terrain and population database information not currently available to the AFP.

As mentioned previously, another future research topic is to initially guide the aircraft into a controllable trim state in cases where the aircraft state lies outside the post-failure/damage flight envelope when the failure/damage occurs, which will require future work to meet the challenge of understanding the suite of trim states and transition behaviors unique for each failure/damage type and develop the appropriate control strategy correspondingly.

As shown by Scenario 4 in Chapter 6, a systematic method is required for *PLANNER_I* to automatically determine how to plan for the initial location p_0^{II} of *trajectory part II* with respective to the desired landing site located at p_{des} so that the distance *r* between these two locations facilitate *PLANNER_II*'s search for *trajectory part II* solutions, especially when stringent but practical turn rate constraints are imposed. The "ideal" approach distance *r* used by *PLANNER_I* should be kept above

a minimum corresponding with the imposed turn rate constraints. Furthermore, if the aircraft completes its *trajectory_part_I* too close to the landing site, a flight segment that guides the aircraft away from the desired landing site may be necessary. The minimum value of *r* may be computed based on the aircraft's minimum turn radius, which is in turn determined by the maximum turning rates contained in the trim database $\tilde{D}$.

Although the two-step strategy presented in this thesis enables trajectory planning to be performed in real-time, search heuristics (i.e., efficient/logical trim state orderings such as "turn-fly-turn-fly" should be examined first) may be included to enable the *PLANNER_II* to more efficiently locate the optimal candidate plan for *trajectory part II* without exhaustive search.

Future research is also required to deploy the AFP in practice. As discussed in Chapter 4, although emergency flight plans are specified as sequences of trim states, the trim transitions are a necessary part of the trajectory plans and thus the trim transition kinematic information is important to the emergency trajectory planning. However, the difficulty of tuning a capable linear-system-based controller that allows the post-failure/damage aircraft to accurately follow the trajectory commands over the trim transitions suggests the need of developing a uniform fault-tolerant trajectory tracking control strategy that could control the post-failure/damage aircraft to decently follow the desired trajectory. Related research shows that nonlinear flight control techniques are promising in this area. In fact, NASA Ames is applying a

neural network based adaptive control technology to the Damage-Adaptive Control which provides a more general fault-tolerant capability to aircraft's flight control system [33].

Currently, the complexity of computing the post-failure/damage trim and transition databases prevents their real-time generation. These databases are computed off-line and then preloaded into the system in order to be instantly accessed by the emergency flight planning. However, this strategy cannot guarantee the AFP has the exact trim and transition databases it needs if the corresponding post-failure/damage aircraft dynamics were not pre-examined. Perhaps the most significant future work required to make the AFP a practical solution is the development of more efficient online techniques for performing the trim/transition analysis required for the AFP to plan feasible landing trajectories. This approach must be fully-automatic since a control engineer is usually not in the cockpit and will likely be tightly coupled with the system identification process.

Bibliography

[1] T. Grayling, Bishop, S. , "Sustainable Aviation 2030: Discussion Document," Institute for Public Policy Research, London, UK 2001, URL: www.ippr.org.uk/uploadedFiles/projects/s_a_2030_discuss.pdf

[2] E. Atkins, Alonso-Portollio, I., Strube, M. J., "Emergency Flight Planning Applied to Total Loss of Thrust," *Journal of Aircraft*, to apear.

[3] M. J. Strube, "Post-failure Trajectory Planning From Feasible Trim State Sequences," in *Aerospace Engineering, Dept.*, vol. Master of Science: University of Maryland, College Park, 2005.

[4] J. Roskam, *Airplane Design Part IV: Layout Design of Landing Gear & Systems*. Lawrence, Kansas: DARcorporation, 2000.

[5] T. L. Chen, Pritchett, A. R., "Development and Evaluation of a Cockpit Decision Aid for Emergency Trajectory Generation," *Journal of Aircraft*, vol. 38, pp. 935-943, 2001.

[6] T. L. Chen, Pritchett, A. R., "On-The-Fly Procedure Development for Flight Re-Planning Following system Failures," presented at AIAA 38th Aerospace Sciences Meeting and Exhibit, Reno, NV, 2000.

[7] T. Fan, Hyams, D. S., Kuchar, J.K., "Study of In-Flight Replanning Decision Aids," presented at AIAA Guidance Navigation and Control Conference, Boston, MA, 1998.

[8] B. Schroeder, Sarter, N., "Supporting Decision-makding and Action Selection Under Time Pressure and Uncertainty: The Case of In-Flight Icing," presented at The International Symposium on Aviation Psychology, Columbus, OH, 2001.

[9] J. D. Boskovic, Prasanth, R., Mehra, R. K., "A Multi-layer Autonomous Intelligent Control Architecture for Unmanned Aerial Vehicles," *Journal of Aerospace Computing, Information, and Communication*, vol. 1, pp. 605-628, 2004.

[10] F. Ahmed-Zaid, Ioannou, P., Gousman, K., Rooney, R., "Accommodation of Failures in the F-16 Aircraft Using Adaptive Control," in *IEEE Control Systems Magazine*, vol. 11, 1991, pp. 73-78.

[11] J. D. Boskovic, Mehra, R. K., "A Multiple Model Adaptive Flight Control Scheme for Accommodation of Actuator Failures," *Journal of Guidance, Control, and Dynamics*, vol. 25, pp. 712-724, 2002.

[12] M. Boskovic, R. K., "Intelligent Adaptive Control of a Tailess Advanced Fighter Aircraft Under Wing Damage," *Journal of Guidance, Control, and Dynamics*, vol. 23, pp. 876-884, 2000.

[13] M. Bodson, Groszkiewicz, J., "Mutlivariable Adpative Algorithms for Reconfigurable Flight Control," *IEEE Transactions on Control Systems Technology*, vol. 5, pp. 217-229, 1997.

[14] J. D. Boskovic, Mehra, R. K., "Failure Detection, Identification and Reconfiguration in Flight Control," in *Fault Diagnosis and Fault Tolerance for Mechatronic Systems, Recent Advances Series: Springer Tracts in*

Advanced Robotics, vol. 1, F. Caccavale, Villani, L., Ed. NY: Springer Verlag, 2002.

[15] P. Chandler, Pachter, M., Mears, M., "System Identification for Adaptive and Reconfigurable Control," *Journal of Guidance, Control, and Dynamics*, vol. 18, pp. 516-524, 1995.

[16] K. Gundy-Burlet, "Augmentation of an Intelligent Flight Control System for a Simulated C-17 Aircraft," *Journal of Aerospace Computing, Information, and Communication*, vol. 1, pp. 526-542, 2004.

[17] P. Williams, "Selected Flight Test Results for Online Learning Neural Network-Based Flight Control System," presented at AIAA 1st Intelligent Systems Technical Conference, Chicago, IL, 2004.

[18] J. C. Latombe, *Robot Motion Planning*. Boston, MA, 1991.

[19] S. M. LaValle, Kuffner, J. J., "Randomized Kinodynamic Planning," *International Journal of Robotics Research*, vol. 20, pp. 378-400, 2001.

[20] T. H. Cormen, Leiserson, C. E., Rivest, R. L., *Introduction to Algorithms*. Cambridge, MA: MIT Press, 1990.

[21] J. Rubio, Vagners, J., Rysdyk, R., "Adaptive Path Planning for Autonomous UAV Oceanic Search Missions," presented at AIAA 1st Intelligent Systems Technical Conference, Chicago, IL, 2004.

[22] A. Pongpunwattana, Rysdyk, R., "Real-Time Planning for Multiple Autonomous Vehicles in Dynamic Uncertain Environments," *Journal of Aerospace Computing, Information, and Communication*, vol. 1, 2004.

[23] N. D. Richards, Sharma, M., Ward, D. G., "A Hybrid A*/Automaton Approach to On-Line Path Planning with Obstacle Avoidance," presented at AIAA 1st Intelligent Systems Technical Conference, Chicago, IL, 2004.

[24] O. A. Yakimenko, "Direct Method for Rapid Prototype of Near-Optimal Aircraft Trajectories," *Journal of Guidance, Control, and Dynamics*, vol. 23, pp. 865-875, 2000.

[25] N. A. Faiz, S. K., Murray, R. M., "Trajectory Planning of Differentially Flat Systems with Dynamics and Inequalities," *Journal of Guidance, Control, and Dynamics*, vol. 24, 2001.

[26] F. Vormer, Mulder, M., Paassen, M. Mulder, J., "Design and Preliminary Evaluation of a Segment-based Routing Methodology," presented at AIAA Guidance, Navigation, and Control Conference, Monterey, CA, 2002.

[27] E. Frazzoli, "Maneuver-based Motion Planning and Coordination for Single and Multiple UAV's," presented at AIAA's 1st Technical Conference and Workshop on Unmanned Aerospace Vehicles, Portsmouth, VA, 2002.

[28] D. McRuer, Ashkenas, I., Graham, D., *Aircraft Dynamics and Automatic Control*: Princeton University Press, 1973.

[29] B. L. Stevens, Lewis, F. L., *Aircraft Control and Simulation*: John Wiley & Sons, Inc., 1992.

[30] S. R. Buss, *3D Computer Graphics: A Mathematical Introduction with OpenGL*. Cambridge, UK: Cambridge University Press, 2003.

[31] Khalil, *Nonlinear Systems*, 3rd ed: Prentice Hall, 2002.

[32] C. T. Chen, *Linear System Theory and Design*, 3rd ed. Oxford: Oxford, 1999.

[33] J. Urnes, Davidson, R., Jacbson, S., "A Damage Adaptive Flight Control System Using Neural Network Technology," presented at American Control Conference, Arlington, VA, 2001.