



4-2019

Model Predictive Control Synthesis for the Innovative Control Effector Tailless Fighter Aircraft

Christopher Proctor

Western Michigan University, proctor11@yahoo.com

Follow this and additional works at: https://scholarworks.wmich.edu/masters_theses



Part of the [Aerospace Engineering Commons](#)

Recommended Citation

Proctor, Christopher, "Model Predictive Control Synthesis for the Innovative Control Effector Tailless Fighter Aircraft" (2019). *Master's Theses*. 4304.

https://scholarworks.wmich.edu/masters_theses/4304

This Masters Thesis-Open Access is brought to you for free and open access by the Graduate College at ScholarWorks at WMU. It has been accepted for inclusion in Master's Theses by an authorized administrator of ScholarWorks at WMU. For more information, please contact maira.bundza@wmich.edu.



MODEL PREDICTIVE CONTROL SYNTHESIS FOR THE INNOVATIVE CONTROL
EFFECTOR TAILLESS FIGHTER AIRCRAFT

by

Christopher Proctor

A thesis submitted to the Graduate College
in partial fulfillment of the requirements
for the degree of Master of Science
Aerospace Engineering
Western Michigan University
April 2019

Thesis Committee:

Richard Meyer, Ph.D., P.E., Chair
Kapseong Ro, Ph.D.
Jennifer Hudson, Ph.D.

© 2019 Christopher Proctor

MODEL PREDICTIVE CONTROL SYNTHESIS FOR THE INNOVATIVE CONTROL EFFECTOR TAILLESS FIGHTER AIRCRAFT

Christopher Proctor, M.S.

Western Michigan University, 2019

A nonlinear model predictive control law was developed for the Lockheed Martin Innovative Control Effector tailless fighter aircraft to track way points. In general, aircraft are described by nonlinear dynamics that are dependent on the regime of flight. Additionally strict requirements on state and actuator constraints are common to all aircraft. Tailless aircraft are usually overdetermined systems, meaning solutions to control problems are not unique, and the system is non-affine. The proposed nonlinear control law considers those constraints during run-time, and solves the nonlinear control problem for a range of points within different flight regimes. The control law was developed using a computer simulation of the tailless fighter aircraft, and further simulation was used to validate the control law when applied to the aircraft. It was found that the controller was able to track reference step commands in altitude anywhere from sea level to 50,000 ft and remain stable. It is also shown that the single nonlinear controller is able to handle lateral translations at the same time as altitude commands, demonstrating its authority over the entire 6 degree of freedom system. The controller is not real time applicable but research indicates that it is possible to apply such a technique in real time. It was concluded that nonlinear model predictive control is a viable control synthesis technique for tailless fighter aircraft if real-time algorithms can be developed.

TABLE OF CONTENTS

LIST OF TABLES	iv
LIST OF FIGURES	vii
CHAPTER	
1 INTRODUCTION	1
1.1 Motivation	1
1.2 Literature Review: Tailless Atmospheric Flight Vehicle Control	3
1.2.1 Control Allocation	3
1.3 Dynamic Inversion	3
1.4 Other Control Synthesis Techniques	7
1.5 Literature Review: MPC applied to Flight Vehicles	8
1.6 Chapter Summary	10
1.7 Thesis Overview	10
2 THE MODEL	12
2.1 The General Fixed Wing Flight Vehicle Model	12
2.1.1 Axes Definitions	13
2.1.2 Kinematic State Variables and Propagation Equations	13
2.1.3 Aerodynamic Coefficient Buildup	15
2.1.4 Calculation of Body Forces and Moments	16
2.2 The ICE Model	17
2.2.1 Control Inputs	18
2.2.2 The Aerodynamic Tables	19
2.2.3 Aerodynamic Data and Limits	20
2.3 Chapter Summary	20
3 MODEL PREDICTIVE CONTROL REVIEW	21

Table of Contents–Continued

CHAPTER		
3.1	Linear MPC	21
3.2	Nonlinear MPC	23
3.3	Chapter Summary	25
4	SIMULATION SETUP	26
4.1	Trimming the Aircraft	26
4.2	MPC Properties	27
4.2.1	The Internal Model	28
4.2.2	Pseudo Inputs	28
4.2.3	Control Allocation	30
4.3	Infeasibility and Suboptimal Solutions	31
4.4	Chapter Summary	34
5	SIMULATION RESULTS	35
5.1	Initial Control Tuning	35
5.2	The Altitude Sweep	42
5.3	Descent	50
5.4	Tracking References in Two Position Coordinates	57
6	CONCLUSIONS AND FUTURE WORK	65
6.1	Observations Based on Climb Angle and Speed	65
6.2	MPC Performance Observations Based on the Simulations	66
6.3	Future Work	66
REFERENCES	69

LIST OF TABLES

2.1	6-DOF aircraft model state variables.	14
2.2	Table containing the dimensionalizing factors for kinetic values.	17
2.3	Relevant geometrical parameters for the ICE model	18
2.4	Relevant mass parameters for the ICE model	18
2.5	Control inputs to the ICE model	18

LIST OF FIGURES

1.1	A rendition of the ICE tailless fighter aircraft adopted from Dorsett et al. [1]	2
1.2	The ICE aircraft with control surfaces emphasized, adopted from Matamoros and Visser [15]	5
1.3	The ADMIRE aircraft with principal control surfaces emphasized, adopted from Hagström [24]	9
2.1	The body frame, adopted from Kale and Chipperfield [5]	12
2.2	An example of one of the tables used to determine aerodynamic force coefficients .	19
3.1	A convex function	22
3.2	A block diagram of a high level closed loop MPC implementation	23
3.3	The receding horizon idea visualized	24
4.1	A depiction of each optimization, their inputs, and outputs	32
4.2	The feasibility loop	33
5.1	Tuning: Commanded force and moment coefficients from the pseudo control optimization	36
5.2	Tuning: Thrust parameters	37
5.3	Tuning: Inboard and outboard leading edge flap deflections	37
5.4	Tuning: All moving wingtip and elevon deflections	38
5.5	Tuning: Spoiler slot and pitch flap deflections	38
5.6	Tuning: Positions and velocities	39
5.7	Tuning: Altitude tracking	39
5.8	Tuning: Y position tracking	40
5.9	Tuning: Euler angles and angular velocities	40
5.10	Tuning: Climb angle and velocity magnitude	41
5.11	Tuning: State tracking cost	42

List of Figures–Continued

5.12 Altitude Sweep: Commanded force and moment coefficients from the pseudo control optimization	43
5.13 Altitude Sweep: Thrust parameters	43
5.14 Altitude Sweep: Inboard and outboard leading edge flap deflections	44
5.15 Altitude Sweep: All moving wingtip and elevon deflections	44
5.16 Altitude Sweep: Spoiler slot and pitch flap deflections	45
5.17 Altitude Sweep: Positions and velocities	46
5.18 Altitude Sweep: Altitude tracking	47
5.19 Altitude Sweep: Y position tracking	47
5.20 Altitude Sweep: Euler angles and angular velocities	48
5.21 Tuning: Climb angle and velocity magnitude	48
5.22 Altitude Sweep: α plotted against time	49
5.23 Altitude Sweep: State tracking cost	50
5.24 Descent: Commanded force and moment coefficients from the pseudo control optimization	51
5.25 Descent: Thrust parameters	51
5.26 Descent: Inboard and outboard leading edge flap deflections	52
5.27 Descent: All moving wingtip and elevon deflections	52
5.28 Descent: Spoiler slot and pitch flap deflections	53
5.29 Descent: Positions and velocities	54
5.30 Descent: Altitude tracking	54
5.31 Descent: Y position tracking	55
5.32 Descent: Euler angles and angular velocities	55
5.33 Descent: Climb angle and velocity magnitude	56
5.34 Descent: State tracking cost	56

List of Figures–Continued

5.35 Two Position Reference: Commanded force and moment coefficients from the pseudo control optimization	57
5.36 Two Position Reference: Thrust parameters	58
5.37 Two Position Reference: Inboard and outboard leading edge flap deflections	58
5.38 Two Position Reference: All moving wingtip and elevon deflections	59
5.39 Two Position Reference: Spoiler slot and pitch flap deflections	59
5.40 Two Position Reference: Positions and velocities	60
5.41 Two Position Reference: Altitude tracking	61
5.42 Two Position Reference: Y position tracking	61
5.43 Two Position Reference: Positions in 3D	62
5.44 Two Position Reference: 3D positions assuming perfect control allocation	62
5.45 Two Position Reference: Euler angles and angular velocities	63
5.46 Two Position Reference: Climb angle and velocity magnitude	64
5.47 Two Position Reference: State tracking cost	64

CHAPTER 1

INTRODUCTION

In this chapter background information for the topic of this thesis is given. The first section describes the motivation for this research, the second examines publications on the topic, and the final section outlines the content of the thesis.

1.1 Motivation

Aircraft are complex nonlinear dynamic systems with strict requirements on performance and robustness of controllers used. To describe the dynamics, aircraft are generally considered rigid body structures whose equations of motion can be derived from Newton's laws of motion. Forces and moments experienced by an aircraft in flight usually come from propulsion sources such as: turbofans, turboprops, propellers, turbojets, etc., gravity, and the aerodynamic forces of lift and drag. The most complex forces to model are the aerodynamic forces. These are usually highly nonlinear functions that depend on a myriad of parameters such as aircraft geometry, flight conditions, and control surface deflections. For flight control design it is common to split aircraft dynamics into two pieces, longitudinal control, and lateral directional control. This can be done because the longitudinal states of the craft are not coupled with the lateral directional states, assuming no control surface contributions. Traditional aircraft take advantage of this and are designed such that the primary control surfaces do not couple the dynamics; this is not true for the ICE model whose control surfaces are nonlinearly coupled. The two sets of dynamics can then be linearized and have controllers designed independently around an operating point.

Aircraft are generally subject to constraints in both states and control inputs. Some common constraints are due to physical limitations of the structure, aerodynamics, and the pilot. Therefore it is necessary that controllers can account for system constraints and successfully operate near

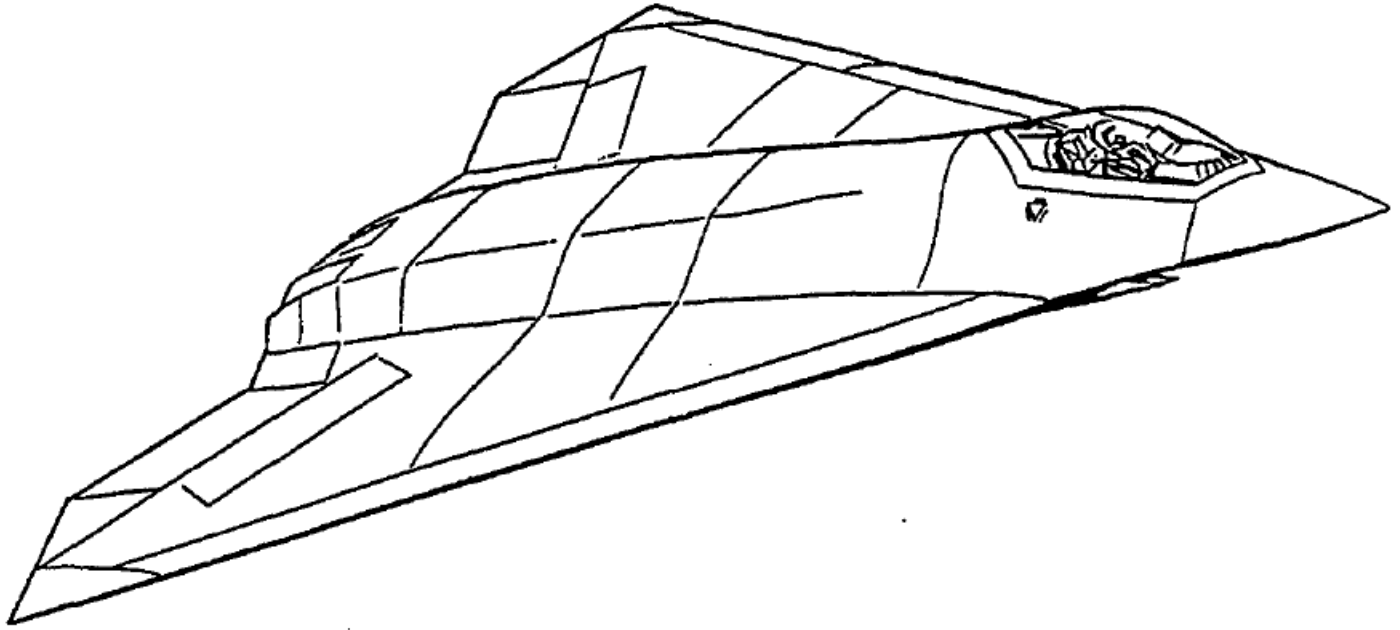


Figure 1.1: A rendition of the ICE tailless fighter aircraft adopted from Dorsett et al. [1]

them to maximize performance. In the case of fighter aircraft, which operate over a wide range of conditions, in order to allow a pilot to focus on completing the mission, and not on limitations of the aircraft, flight controllers have to be able to limit the response to pilot inputs so that events such as structural failure, stall, or black out of the pilot due to acceleration do not occur.

By adding constraints to states and control inputs the control problem becomes nonlinear, even if the plant itself has been linearized. One synthesis technique used for closed loop optimal control of constrained systems is called Model Predictive Control (MPC). In recent years MPC has garnered attention by the aeronautical community. Researchers have explored the validity of MPC to impact flight control, but MPC has yet to become widely applied [2] [3] [4]. A major research topic in aerospace control is reconfigurable control, or control laws that can adapt to unexpected control surface failures or aircraft damage. MPC can be intuitively applied to reconfigurable control due to the nature of its formulation [5] [6] [7].

The goal of this research is to lay the foundation to develop a real-time MPC algorithm to track commands in position for Lockheed Martin's Innovative Control Effector (ICE) tailless fighter aircraft, while respecting constraints in control inputs.

1.2 Literature Review: Tailless Atmospheric Flight Vehicle Control

Tailless aircraft suffer from several issues, with regards to control, that traditional aircraft do not [8].

1. The difficulties in generating enough yaw control power.
2. Multi-axis instabilities.
3. Multiple control surfaces can influence any single axis.
4. Highly nonlinear and coupled surfaces.

The most prominent of these issues is the control allocation problem, which is the combination of redundant and nonlinearly coupled control surfaces.

1.2.1 Control Allocation

In general, tailless aircraft have multiple redundant, coupled control surfaces [8]. Therefore it is of interest to determine how best to distribute control inputs across those surfaces to achieve a commanded state. The problem of control distribution is known as control allocation. Choosing which control surfaces are necessary to achieve a desired system response is not always intuitive for an over actuated system such as the tailless aircraft. The control allocation problem can be solved online via control methods or predetermined by the designer using intuition and/or numerical tools. Online control algorithms may find the control allocation that minimizes some objective function. A few examples might be prioritizing the input effectors that have the highest control authority, determining which combination of inputs results in the least amount of drag, or minimizing control input power. The following section looks at several types of controllers used for tailless aircraft, and their allocation methods, if they employ one.

1.3 Dynamic Inversion

The concept behind dynamic inversion (DI) can be summarized as given an invertible plant dynamics, the control commands needed to produce a desired response can be calculated by simply

inverting the known plant dynamics [9]. By applying DI a nonlinear system can be modified such that linear control methods may be applied [10]. DI is attractive for flight control because it has an intuitive architecture, explicit model-following behavior, the ability to be used to introduce fundamental-level simulated failures within the aerodynamic model for testing the performance of advanced control elements, and it can be included in the stability proofs for many advanced control schemes [11]. Some disadvantages of DI are that it lacks in robustness due to its reliance on a highly accurate model, it requires the system to be invertible on some level, and in its simplest form requires the system to be stable [8].

Ngo et al. [12] presents an outer loop robust controller, and an inner loop dynamic inversion scheme to control the ICE tailless fighter model, depicted in 1.2. The outer loop controller was designed using μ synthesis. This controller adds robustness in the presence of plant dynamic uncertainties, in this case imperfect inversion and system parameter variations within the model. μ is the smallest structured singular value related to uncertainties within a model of a system. This value can be used to determine whether or not the system will remain stable in the presence of plant uncertainties, this is known as the μ condition. μ synthesis is a class of techniques used to design controllers that meet the μ condition, and thereby guarantee stability under a range of plant uncertainties [13] [14]. After control inputs are available from the robust controller decides on inputs the DI determines how to achieve those inputs. The DI was designed to decouple the longitudinal and lateral axes, and steer the vehicle towards equilibrium if disturbed. The DI linearizes the plant by multiplying the plant by the inverse of its own nonlinear terms, thereby modifying the system dynamics to some desired dynamics. The desired dynamics are such that the aircraft's modified dynamic response is similar throughout the flight envelope. The controller itself is a static gain matrix that is updated as the flight conditions change. The inputs chosen as control variables are the pitch rate augmented to include the angle of attack, yaw rate, and roll rate. The controller assumes full state feedback. Control allocation is handled by a linear transformation from the generalized commands, given by the DI, to actual control surface deflection. The transformation is done by applying the generalized commands to a control effectiveness matrix. The value of each of the effectiveness of the control surfaces is a function of M and altitude, to update these values the control derivatives for several flight conditions are stored in a database. The controller was tested using M 0.3 to 0.5 and altitudes ranging 10,000 to 20,000 feet. The system showed robustness against

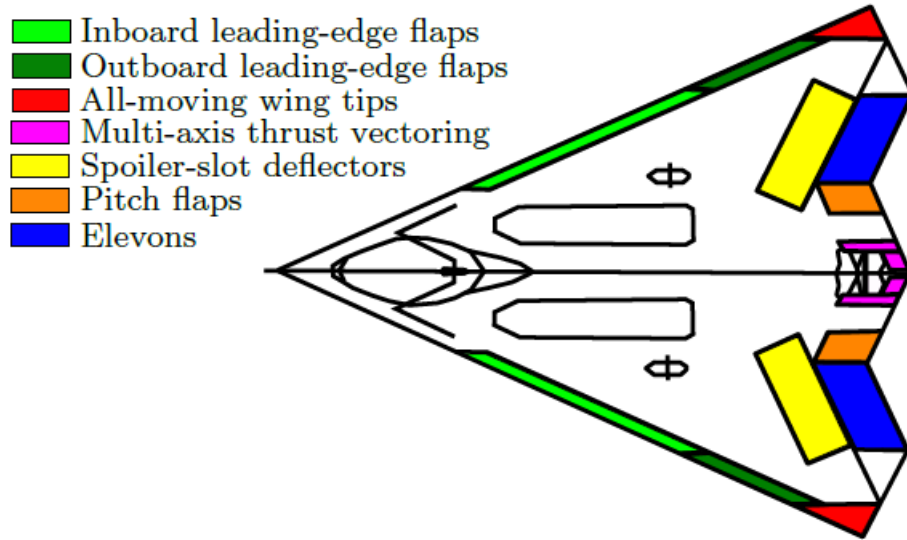


Figure 1.2: The ICE aircraft with control surfaces emphasized, adopted from Matamoros and Visser [15]

control derivative uncertainties up to 15% and stability derivative uncertainties up to 50%.

Bo et al.[16] compares two nonlinear dynamic inversion (NDI) controllers designed for a general tailless unmanned aerial vehicle. One controller uses an active disturbance rejection control (ADRC) technique to handle parameter modeling robustness issues, and the other does not. The study divides the system states into a fast response group and a slow response group. Controllers are then designed for each group and cascaded together. Inputs are provided to the control algorithm by a trajectory planner, passed to the fast group ADRC algorithm which passes generalized control inputs to the fast group NDI algorithm. The outputs of the fast group NDI are the input to the slow group controller which is structured the same way. The outputs of the slow group controller are passed to a control allocation algorithm which transforms the generalized commands into control surface deflections. The idea behind the ADRC synthesis technique is to use a trajectory planner, provide a noise tolerant tracking differentiator, employ nonlinear control laws, and provide total disturbance estimation and rejection [17]. By adding an unknown “total disturbance” term modeled as a function of states and known disturbances, and introducing additional states, an estimator can be built that will have inputs consisting of the controller command, and the known plant outputs. The outputs of this observer are the estimated system states and the previously unknown total disturbance. Application of this technique results in several tuning parameters that are

largely independent of the system model, but do rely on the system time scales [17]. The controller was tested with 30% variation in all aerodynamic coefficients, tracking references in angle of attack, side slip, and bank angle at seven different points within the flight envelope. The algorithm was not reported to have been tested against output disturbances.

Buffington [18] applies a multi-branch linear programming control allocation technique in conjunction with a DI algorithm on the ICE model for attitude control. DI inverts the command variable dynamics and proportional/integral feedback generates a desired robust loop shape. The output of the DI is a virtual control that the allocation algorithm turns into real control surface deflections. The multi-branch linear programming algorithm can be conceptualized as follows: check to ensure a command is feasible, if it is then use another solver to determine the best way to distribute that solution among control surfaces while optimizing a secondary objective. If the solution is not feasible then the control input solved for will be as close to the desired reference as possible and no further control objectives are optimized. The control scheme was able to utilize all available power without violating actuator limits, and minimizing drag. The flight regimes tested span combinations of M 0.35-0.85 and altitudes of 15000 to 500 ft. The major drawback of this approach was not being able to account for nonlinear control surface based interactions

Matamoros et al. [15] apply an incremental nonlinear dynamic inversion (INDI) law coupled with an incremental control allocation (INCA) algorithm to control the attitude of the ICE model. The incremental approaches separates a control problem up into small pieces and solves for small changes in control input that drive the states. Locally linearizing the system dynamics at every step using a first order Taylor series expansion and assuming a small incremental time change over which the state derivatives evolve much faster than the states themselves allows for a virtual control input to be solved for. The virtual input is only a function of a nonlinear control surface mapping from moments to control surface deflections and the states at the beginning of the solution. Due to the ICE model being overactuated the Jacobian of the control surface model is not square. This prevents the direct inversion of the model. Rather than inverting the Jacobian, the INCA algorithm takes a portion of the INDI solution, which is referred to as the pseudo control input and determines an increment in the control input vector such that the Jacobian of the control surface model multiplied by the increment is the pseudo control input. By doing so the INCA completes the inversion solution and outputs a control surface deflection increment. The incre-

mental formulation allows for the nonlinear allocation problem to be solved with linear allocation algorithms while still taking into account nonlinear interactions among control surfaces. The optimization routine selected for control allocation was the one that found accurate solutions in the least time, quadratic programming, which also took into account control surface constraints. This control scheme has been simulated in real-time at an altitude of 20000 ft and M 0.85. The group was able to track commands in all body rates using irregularly shaped command references.

Bacon and Ostroff [19] use DI to synthesize a reconfigurable flight control law for Lockheed Martin's reconfigurable control for tailless fighter aircraft (RESTORE) program, that would later become the ICE model. A reconfigurable control law is generally composed of three elements. Failure detection and isolation to determine which surface is no longer useful, on-line parameter identification to provide a model of the damaged vehicle, and on-line control design which utilizes the information from the other two elements to reestablish control [19]. The need for a nominal model is eliminated by an on-line parameter estimation that is applied to the aircraft itself. This estimation measures the rates of change of the states of the aircraft directly via instrumentation, and feeds them back into the DI. The DI control output is formulated by evaluating a Taylor series expansion using the previous states and control inputs to evaluate the needed partial derivatives. This yields a required change in surface position from its previous position, that is independent of a nominal model of the aircraft. The allocation problem is solved via a minimum norm solution, using a properly chosen weighting matrix. If surfaces are lost, the weights within the allocation problem that correspond to those surfaces can be changed to zero, and those surfaces will not be considered in the solution. It is shown the aircraft can track commands in body rates, sideslip angle, angle of attack, and stability axes angular rates for several types of failures including combinations of both complete losses of control surfaces and control surfaces being stuck in fixed positions.

1.4 Other Control Synthesis Techniques

Calise et al. [20] propose a direct adaptive reconfigurable controller with a baseline dynamic inverter for a general tailless aircraft. When an adaptive control is direct it does not attempt to identify the plant, instead an explicit model is used and the generated error is used in the adaptive control law [21]. Adaption is done by a feed forward neural network that is trained using sigmoidal

activation functions. The weights of the neural network are updated on-line by a law that depends on the feedback error of the states. The neural network takes in states of the system and outputs an adaption signal that is subtracted from a filtered acceleration and an error feedback to produce a psuedo input which feeds into the DI. The output of the DI is allocated by an unspecified control selector. A simulation at M 0.6 and altitude 15000 ft with the left aft body flap stuck at 30 degrees was run. The simulation found that the neural net adapted the control input to compensate and was able to track longitudinal and lateral references better than the same system without adaption.

Eberhardt and Ward [22] apply indirect adaption to the RESTORE tailless aircraft model. Indirect control estimates the parameters of the unknown plant from input-output data and, use these estimates to adjust the parameters of a controller such that the transfer function of the controlled plant evolves to that of a model [21]. The indirect adaption is done by a combination of model predictive control (MPC) and on-line parameter identification (PID). The PID estimates data as the system is in use to update a linear model. This updated model is used by the MPC to generate control pseudo commands in the form of forces and moments, which are given to a redistributed pseudo-inverse algorithm that allocates the control commands. The simulation results were able to track commands in pitch rate, roll rate, and side-slip angle at an unspecified flight condition with unspecified solution times. However, it is stated that the simulations were not run in real time.

Every work discussed above shares a common element: control allocation. Every work also has a different method for solving the control allocation problem. This is evidence the allocation problem and algorithms designed to solve it are a major focus of tailless fighter control research. Much of the published work on tailless control is based on DI synthesis and solving the problems associated with DI, due to DI naturally lending itself to the solution of control for MIMO systems, and therefore flight vehicles. Less work has been published under the guise of adaptive control, and less still as predictive control.

1.5 Literature Review: MPC applied to Flight Vehicles

Simon [23] proposes using linear MPC with feedback linearization to control the nonlinear ADMIRE model. ADMIRE is an advanced generic simulation model of a modern delta-canard fighter aircraft [24]. The ADMIRE model configuration is shown in 1.3. The MPC has a quadratic

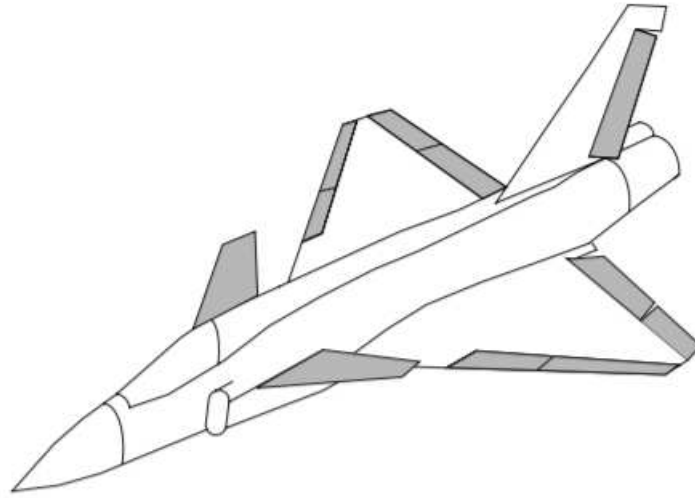


Figure 1.3: The ADMIRE aircraft with principal control surfaces emphasized, adopted from Hagström [24]

cost function that penalizes angle of attack and pitch rate reference tracking error and use of control input. The cost function is quadratic, so it can be used in a quadratic programming solver. The model of the aircraft is nonlinear in a single state, the nonlinear state is linearized through feedback using the control input. This makes the constraints applied to the MPC nonlinear in the control input, and dependent on the state. Simulation flight conditions were M 0.6 and altitude 1000 m with a sample rate of 60 Hz. The MPC regulated angle of attack and pitch rate to some equilibrium point over the course of a few seconds simulation time. The real-time solution times are not given.

Slegers et al. [25] present a nonlinear MPC technique for general unmanned air vehicles suited for real-time application. The work presents a method of finding a closed form solution to the nonlinear problem through expanding the output and control in a truncated Taylor series. Through proper expansion the cost function can be represented as a quadratic whose optimization parameters are the elements of each Taylor series. The elements of the weighting matrix are given as the square of the integral of the coefficients from the Taylor series expansion over the length of the prediction horizon. This method assumes the system has the same number of outputs as inputs. The algorithm is derived for a general case and applied to a simulated unmanned glider. The control inputs are the deflections on aileron, rudder, and elevator, the tracked outputs are the bank, pitch, and heading angles. The outputs converge on the desired values in seven seconds, but

the solution times for control inputs are not commented on.

1.6 Chapter Summary

MPC synthesis for tailless aircraft is a relatively unexplored topic in comparison to DI based techniques. DI has been successful in implementing real-time solutions to the entire flight envelope, with added complexity to address robustness issues and reconfigurable flight. However, the reviewed literature on DI do not discuss how states, actuator rates, or actuator limits can be taken into consideration while performing DI synthesis. It is likely that saturation limits are imposed on control inputs that are unknown to the controller itself and handled by the allocation algorithms.

MPC has been explored in many of the same ways DI has but for traditional aircraft, which in general do not require advanced allocation algorithms due to not having redundant and coupled control surfaces. MPC in flight vehicles has been successful in tracking both longitudinal models, and full 6 DOF models of traditional aircraft, while respecting state, and actuator constraints. Solution times in much of the literature are not mentioned, implying that they may not be real-time.

1.7 Thesis Overview

Chapter 2 of this work will attempt to lay the foundation for a general understanding of aircraft dynamics. It will begin with discussing axes followed by kinematic variables, aerodynamic coefficients, body forces and moments, and ending with ICE specific parameters. Chapter 3 will lay the foundation for a general MPC implementation. The first section will discuss linear MPC and the receding horizon idea, followed by a brief discussion of nonlinear MPC. Chapter 4 will deal with the MPC simulation specifics such as cost functions, constraints, and tuning parameters. This will be supplemented with other discussion about the internal model, and control allocation. The final section of Chapter 4 will discuss briefly how infeasibility and suboptimal solutions are used during simulation. Chapter 5 will discuss the results of simulation, including flight profiles, solution times, and control input values. There will be a brief mention of how the MPC was tuned. Chapter 6 is the final chapter and will begin by making conclusions based on the simulation results

and research tribulations. This will be followed with some recommendations for future work.

CHAPTER 2

THE MODEL

This chapter begins by setting forth the general equations of motion for a fixed wing aircraft. First relevant axes definitions are given for modeling an aircraft. The second section discusses the kinematic state variables and their propagation equations. The third and fourth sections discuss aerodynamic coefficient build ups and force and moment equations. The final section describes geometric properties and control inputs specific to the ICE model.

2.1 The General Fixed Wing Flight Vehicle Model

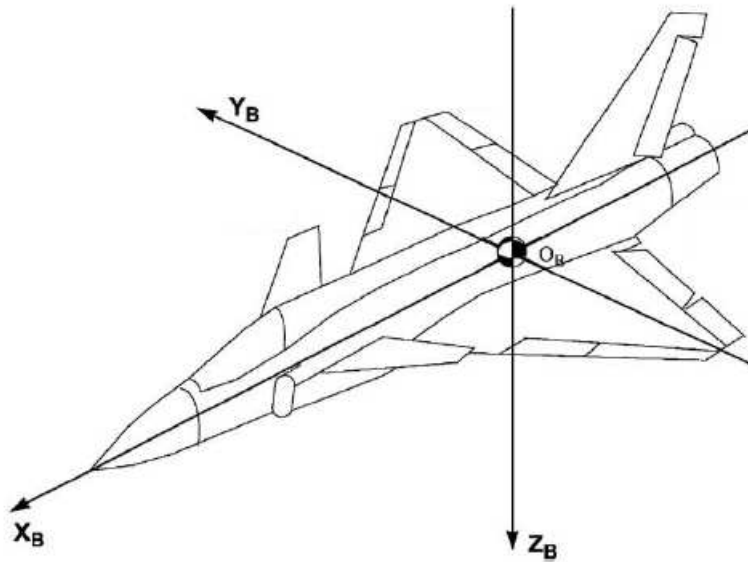


Figure 2.1: The body frame, adopted from Kale and Chipperfield [5]

2.1.1 Axes Definitions

The study of aircraft dynamics requires applying multiple frames of reference to an aircraft. The most relevant frames are the body frame, inertial frame, and stability frame. The body frame is defined as positive out of the nose of the aircraft, X_B , positive out of the right wing looking from the back of the craft, Y_B , and positive out of the bottom of the aircraft, Z_B . Additional notation associated with the body frame are the body velocities u , v , w , which align with the X_B , Y_B , and Z_B body axes respectively. These three axes are known as the roll, pitch, and yaw axes respectively. The origin of the body frame is placed at the center of mass of the craft, seen in 2.1.

For many atmospheric flight dynamics applications the Earth frame is assumed to be inertial. The origin of the inertial frame can be placed wherever is convenient as long as it is aligned with the Earth frame and fixed. The standard Earth frame has axes that are considered positive north, positive east, and positive towards the center of the earth.

To convert from the body and inertial frames a rotation matrix, L_{IB} , that transforms through the Euler angles: bank angle, Φ , pitch angle, θ , and heading angle, Ψ , is used.

$$L_{IB} = \begin{bmatrix} C_\theta C_\Psi & S_\Phi S_\theta C_\Psi - C_\Phi S_\Psi & C_\Phi S_\theta C_\Psi + S_\Phi S_\Psi \\ C_\theta S_\Psi & S_\Phi S_\theta S_\Psi + C_\Phi C_\Psi & C_\Phi S_\theta S_\Psi - S_\Phi C_\Psi \\ -S_\theta & S_\Phi C_\theta & C_\Phi C_\theta \end{bmatrix} \quad (2.1)$$

For brevity C , S , and T are used to represent the trigonometric functions sin, cos, and tan, respectively, when used in rotation matrices.

2.1.2 Kinematic State Variables and Propagation Equations

There are twelve kinematic states simulated in order to completely represent the motion of an aircraft. States that will be used hereafter are listed in table 2.1. Therefore there are twelve nonlinear differential equations that must be solved simultaneously to model an aircraft during flight in the body frame.

The first group of equations, 2.2, are the equations that describe the angular accelerations, \dot{p} , \dot{q} , \dot{r} in the roll, pitch, and yaw directions respectively. In this equation I with subscript denotes the moment of inertia about a particular axis, and L , M , N are the rolling moment, pitching moment,

Table 2.1: 6-DOF aircraft model state variables.

Variable	State
p	Body axis roll rate
q	Body axis pitch rate
r	Body axis yaw rate
u	Body axis X velocity
v	Body axis Y velocity
w	Body axis Z velocity
Φ	Bank angle
θ	Pitch angle
Ψ	Heading angle
x_I	Inertial X position
y_I	Inertial Y position
z_I	Inertial Z position

and yawing moment respectively.

$$\begin{aligned}
 \begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} &= - \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix}^{-1} \left\{ \begin{bmatrix} \dot{I}_{xx} & \dot{I}_{yy} & \dot{I}_{xz} \\ \dot{I}_{yx} & \dot{I}_{yy} & \dot{I}_{yz} \\ \dot{I}_{zx} & \dot{I}_{zy} & \dot{I}_{zz} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \right. \\
 &\quad \left. + \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{bmatrix} I_{xx} & I_{yy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \right\} + \begin{bmatrix} I_{xx} & I_{yy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix}^{-1} \begin{bmatrix} L \\ M \\ N \end{bmatrix} \quad (2.2)
 \end{aligned}$$

The second grouping, 2.3 describes the linear accelerations in the body frame along the X, Y and Z body axes.

$$\dot{u} = Rv - Qw - g \sin(\theta) + F_{XB}/m \quad (2.3a)$$

$$\dot{v} = Pw - Ru + g \cos(\theta) \sin(\Phi) + F_{YB}/m \quad (2.3b)$$

$$\dot{w} = Qu - Pv + g \cos(\theta) \cos(\Phi) + F_{ZB}/m \quad (2.3c)$$

The Euler angles, Φ , θ and Ψ propagation through time is shown next:

$$\dot{\Phi} = P + Q \sin(\Phi) \tan(\theta) + R \cos(\Phi) \tan(\theta) \quad (2.4a)$$

$$\dot{\theta} = Q \cos(\Phi) - R \sin(\Phi) \quad (2.4b)$$

$$\dot{\Psi} = Q \sin(\Phi) \sec(\theta) + R \cos(\Phi) \sec(\theta) \quad (2.4c)$$

The final grouping describes how the inertial position x_I y_I z_I propagates in time. To see a full derivation refer to [26]

$$\begin{bmatrix} \dot{x}_I \\ \dot{y}_I \\ \dot{z}_I \end{bmatrix} = \begin{bmatrix} C_\theta C_\Psi & S_\Phi S_\theta C_\Psi - C_\Phi S_\Psi & C_\Phi S_\theta C_\Psi + S_\Phi S_\Psi \\ C_\theta S_\Psi & S_\Phi S_\theta S_\Psi + C_\Phi C_\Psi & C_\Phi S_\theta S_\Psi - S_\Phi C_\Psi \\ -S_\theta & S_\Phi C_\theta & C_\Phi C_\theta \end{bmatrix} \begin{bmatrix} u \\ v \\ w \end{bmatrix} \quad (2.5)$$

2.1.3 Aerodynamic Coefficient Buildup

Aerodynamic coefficients are non-dimensional representations of the control surfaces and aircraft geometry that are used to build a linear estimate of the forces that act upon an aircraft; there are two types of aerodynamic coefficients used in these estimates: stability derivatives, and control derivatives. Control derivatives measure how much change in force or moment occurs when a control surface is deflected. Stability derivatives measure how much change in force or moments occur when there is a change in flight parameters such as angle of attack. In practice these coefficients are linear representations of how each factor in question effects each force and moment individually. The significance of non-dimensionalizing values is to allow for the data received from the testing of small models in wind tunnels to be scaled to the full size aircraft of the same geometry. For example, the data for the ICE model was collected using a 1/18th scale model at the Subsonic Aerodynamic Research Laboratory at Wright Laboratories wind tunnel [27].

In order to elaborate on the previous concept and become familiar with the notation consider this small example.

$$\frac{\partial C_M}{\partial \alpha} = C_{M\alpha}$$

This is the general notation used hereon. It is read: the partial derivative of the coefficient of pitching moment, C_M , with respect to angle of attack, α . The forces and moments are then built in the following manner.

$$C_{M\alpha}\alpha + C_{M\delta e}\delta_e = C_M$$

This buildup method is at its essence a slope multiplied by an independent variable. A compact way to represent these equations in general is given in equation 2.6 where i represents a general index standing in for one of the six forces and moments and n is an index that should be substituted for every control surface and every flight parameter that has an impact on that particular derivative.

$$C_i = \sum_n C_{in} \delta_n \quad (2.6)$$

2.1.4 Calculation of Body Forces and Moments

The group of variables representing the linear buildup of body forces and moments are shown in equations 2.7 and 2.8 using the general summation above. There is one equation for each kinetic value totaling six. These values are the body forces, F_{XB} , F_{YB} , and F_{ZB} aligned with the X_B , Y_B , and Z_B and the moments L_N , M_N , and N_N along the roll, pitch, and yaw axes respectively. These equations represent the six non-dimensional kinetic values that are then dimensionalized to calculate forces and moments. The prefix C is used to show that these are non dimensional coefficients.

$$\begin{aligned} C_{BX} &= \sum_n C_{XB_n} \delta_n \\ C_{BY} &= \sum_n C_{YB_n} \delta_n \\ C_{BZ} &= \sum_n C_{ZB_n} \delta_n \end{aligned} \quad (2.7)$$

Table 2.2: Table containing the dimensionalizing factors for kinetic values.

Kinetic value	Dimensionalizing factor
F_{XB}	$q_{\infty}S_{ref}C_{XB}$
F_{YB}	$q_{\infty}S_{ref}C_{YB}$
F_{ZB}	$q_{\infty}S_{ref}C_{ZB}$
L_L	$q_{\infty}S_{ref}bC_{LN}$
L_M	$q_{\infty}S_{ref}\bar{c}C_{MN}$
L_N	$q_{\infty}S_{ref}bC_{NN}$

$$\begin{aligned}
 C_{LN} &= \sum_n C_{LN_n} \delta_n \\
 C_{MN} &= \sum_n C_{MN_n} \delta_n \\
 C_{NN} &= \sum_n C_{NN_n} \delta_n
 \end{aligned} \tag{2.8}$$

The dimensionalizing factors are listed in Table 2.2. These factors are a direct result of non-dimensionalizing the forces and moments using the Buckingham Pi theory [28]. Notice that the factors for the body forces are in terms of only reference area, S_{ref} , dynamic pressure, q_{∞} and their respective coefficients while the factors for moments have an additional length element, either the span, b , or the chord length \bar{c} .

The effects of propulsion cannot be ignored. The previous table does not include additions for thrust in any direction, or the moment caused by the thrust due to being offset from the center of mass of the vehicle. Thrust can be included by summing the thrust in each direction with their respective directions and summing the moment caused by the thrust to the body moments.

2.2 The ICE Model

The aircraft used in this research is the ICE model a high-sweep tailless fighter concept introduced in January of 2017 for the purpose of innovative flight control research. This nonlinear flight model allows for 13 control inputs, including vectored thrust. Table 2.3 below is a list of constant geometric and mass properties of the craft used for simulation.

The mass properties of the ICE model are presented in table 2.4.

Table 2.3: Relevant geometrical parameters for the ICE model

Geometrical Attribute	Value
Reference Wing Area,	806.6 ft ²
Reference Span, b_{ref}	37.5 ft
Mean Aerodynamic Chord (MAC)	345 in
Fuselage Station Leading Edge MAC	160.84 in

Table 2.4: Relevant mass parameters for the ICE model

Mass and C.G. Locations		Moment of Inertia Values	
Weight	37084 lb	I_{xx}	42576 slug-ft ²
x_{cg}	0.36	I_{yy}	81903 slug-ft ²
y_{cg}	0	I_{zz}	118379 slug-ft ²
z_{cg}	88.97	I_{xz}	-525 slug-ft ²

2.2.1 Control Inputs

The ICE model has thirteen control inputs, Table 2.5 lists each and their respective deflection limits.

Table 2.5: Control inputs to the ICE model

Control Input	Upper and Lower Limit [Degrees]
Inboard leading edge flaps left and right	40, -0.001
Outboard leading edge flaps left and right	40, -40
All moving wingtip left and right	60, -0.001
Elevons left and right	30, -30
Spoiler slot deflector left and right	60, -0.001
Pitch flap left and right (Deflects together)	30, -30
Pitch thrust vector deflection	15, -15
Yaw thrust vector deflection	15, -15

The six degree of freedom simulation that solves the state equations for the ICE model assumes the mass of the aircraft will not change, meaning no fuel cost is calculated, and the moment

of inertia values remain constant. This is important to note, as the equations described by equation 2.2 are generalized to describe a time varying inertia matrix. Changes in mass are not considered in the ICE simulation, to reflect this the matrix that involves the time rate of change of inertia values will become a matrix of zeros during simulation.

2.2.2 The Aerodynamic Tables

The aerodynamic coefficients, presented previously, come from a group of 108 lookup tables. The six force and moment coefficients are comprised of the sum of up to 19 aerodynamic coefficients. These coefficients are nonlinear functions of several inputs and states. Each control input or state is used to determine a specific flight coefficient via interpolation. An example such as figure 2.2 takes in M , angle of attack, α , and right spoiler slot deflector, left spoiler slot deflector, and pitch flap deflection and calculates to contribution of all these to, C_{XB} . These tables were empirically gathered as stated before are non-smooth, which will later be discussed in relation to controller performance. An example table, produced by linear interpolation to give a 3-D slice of the 5-D coefficient table discussed, is shown in figure 2.2.

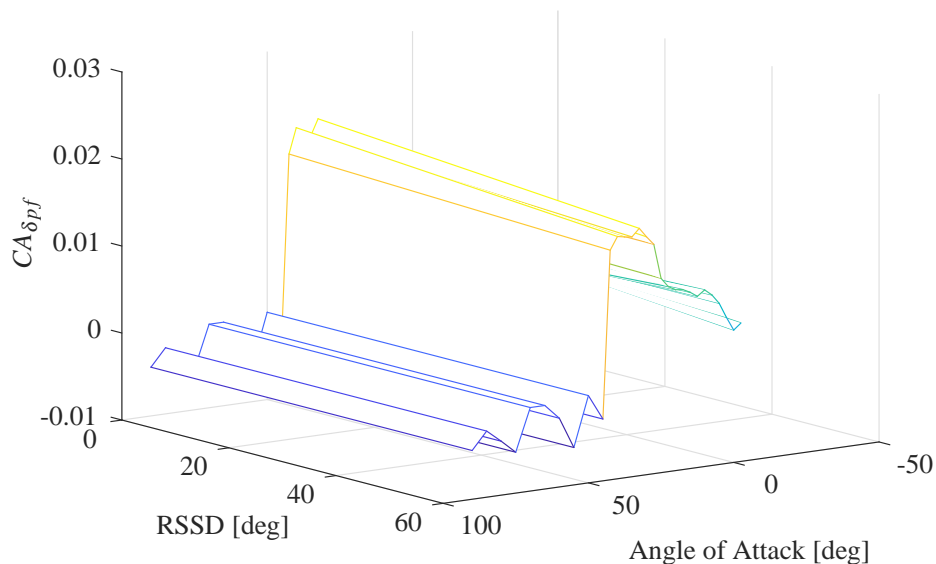


Figure 2.2: An example of one of the tables used to determine aerodynamic force coefficients

This table is a 3-D slice of the 5-D coefficient table discussed above.

2.2.3 Aerodynamic Data and Limits

The range of the angle of attack used during Lockheed Martins wind tunnel testing is -4° to 90° . The angle of attack, α , is defined as the angle between incoming air and the chord line of the wing about the pitch axis. This is stated to clarify that the model can simulate flight at angles greater than 90° relative to the Earth frame. The sideslip angle, β , was limited to $\pm 30^\circ$. Similarly this angle is measured relative to incoming wind from the roll axis about the yaw axis [27]. It is important to note when discussing controller constraints that not all 108 tables have data for the full range of α and β , meaning that some control surfaces are not accurately represented for the full range of test data.

2.3 Chapter Summary

There are 12 nonlinear differential equations of motion for an aircraft. These capture the angular accelerations, Euler angle velocities, inertial velocities, and linear body accelerations. The acceleration equations for the ICE model consider the body moments and forces that are built from the 108 aerodynamic coefficients and propulsion forces. Those aerodynamic coefficients account for aircraft geometry, control surface deflections, as well as various flight parameters. These equations and coefficients will be used in the solution to the optimal control problem formulated in Chapter 3, and applied in Chapter 5.

CHAPTER 3

MODEL PREDICTIVE CONTROL REVIEW

Model Predictive Control (MPC) is a form of optimal control that attempts to anticipate how a model will react to certain control inputs and drives the model based on these predictions to minimize a cost function. The solution to such a problem can take a relatively long time depending on the optimization algorithm and the system dynamics. Due to this, in its infancy MPC was generally used for processes that were slow, as to give the controller time to find solutions. Other major drawbacks include concerns about the feasibility of online optimization, model uncertainty limitations, stability, cost function term penalty weight updating for different operating points. Despite these drawbacks MPC is still alluring in the aerospace world, due to its ability to handle multi-variable control, actively consider actuator limitations, and allow for operation at performance constraints [29].

This section aims to convey a general mathematical understanding of MPC beginning with the linear MPC and following with a brief discussion of nonlinear MPC. MPC is most commonly formulated in discrete time, so the discussion will be limited to discrete time formulations. All MPC formulations result in nonlinear controllers if inequality constraints are enforced on the system. The term linear MPC refers to the plant model being linear, while nonlinear MPC refers to the plant model being nonlinear.

3.1 Linear MPC

Assume the system dynamics can be described by a discrete time difference Equation 3.1. Where the subscript k is a time index for sample time T_s .

$$x_{k+1} = Ax_k + BU_k \tag{3.1}$$

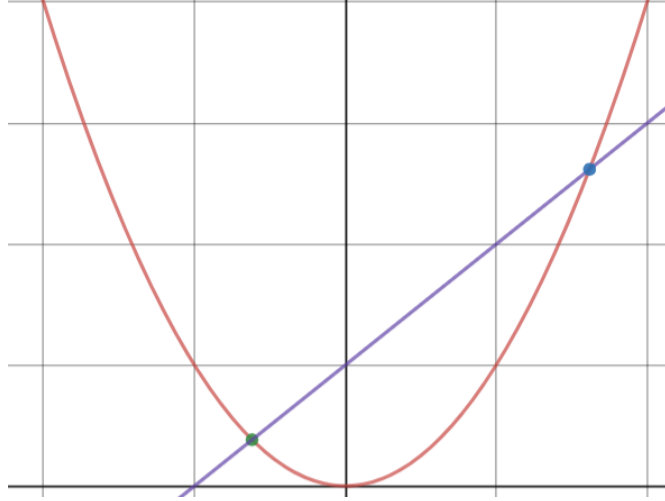


Figure 3.1: A convex function

Now consider a discrete time-constrained optimal control problem, with state x_k , input U_k , two weighting matrices Q and S and a horizon H_p .

$$\min_{U_{k+1}, \dots, U_{k+H_p-1}} (x_{H_p})^T Q (x_{H_p}) + \sum_{j=0}^{H_p-1} (x_{k+j}^T Q x_{k+j} + U_{k+j}^T S U_{k+j}) \quad (3.2)$$

subject to

$$x_{k+j+1} = Ax_{k+j} + BU_{k+j}$$

$$x_{k+j} \in \mathcal{X}$$

$$U_{k+j} \in \mathcal{U}$$

$$x_{H_p} \in \Gamma$$

Where the sets \mathcal{X} and \mathcal{U} are convex, this can be viewed as a general convex optimization. A convex function is one in which a function curve connecting two points lies below a line connecting the two points, this is shown in Figure 3.1, a more rigorous mathematical description can be found in [30]. Convex functions have only a single minimum, and thus the local minimum is a global minimum. This makes convexity a very attractive property of functions when applying typical gradient descent optimization methods, because it guarantees globally optimal solutions.

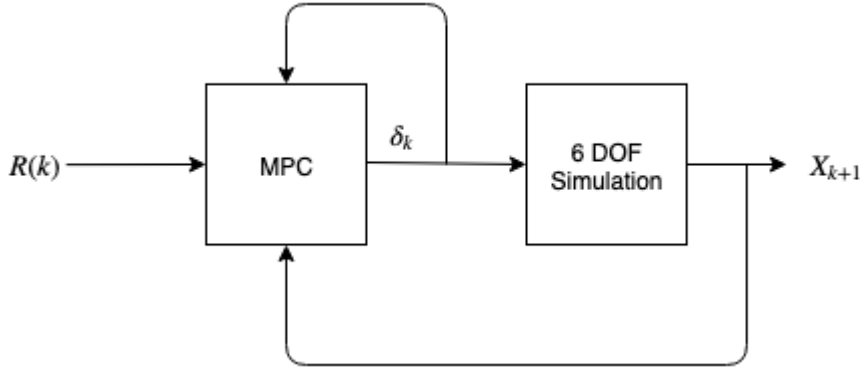


Figure 3.2: A block diagram of a high level closed loop MPC implementation

The selection of the bounding function and constraints, Γ , is discussed at length in Mayne et. al. [31].

Solving this optimal control problem once results in open loop control. To close the loop, only the first control in the optimal control sequence is applied from k to $k + 1$. At the next step, $k + 1$, we measure the current state and redo the optimization with the new current state as x_0 . This is known as Receding Horizon Control because the control problem is solved over a finite future horizon that moves into the future as T_s increments changes. The closed loop system can be represented by the block diagram shown in Figure 3.2.

Using the predicted outputs of the system, \hat{y} , the cost function can be reformulated to track a reference, $r(k + j)$, without a terminal penalty, where H_p is the number of partitions [32].

$$\min_{U_{k+1}, \dots, U_{k+H_p-1}} \sum_{j=0}^{H_p} (\hat{y}_{k+j} - r_{k+j})^T Q (\hat{y}_{k+j} - r_{k+j}) \quad (3.3)$$

3.2 Nonlinear MPC

Consider a nonlinear discrete time dynamic system

$$x_{k+1} = f(x_k, U_k) \quad (3.4)$$

The MPC formulation takes the form

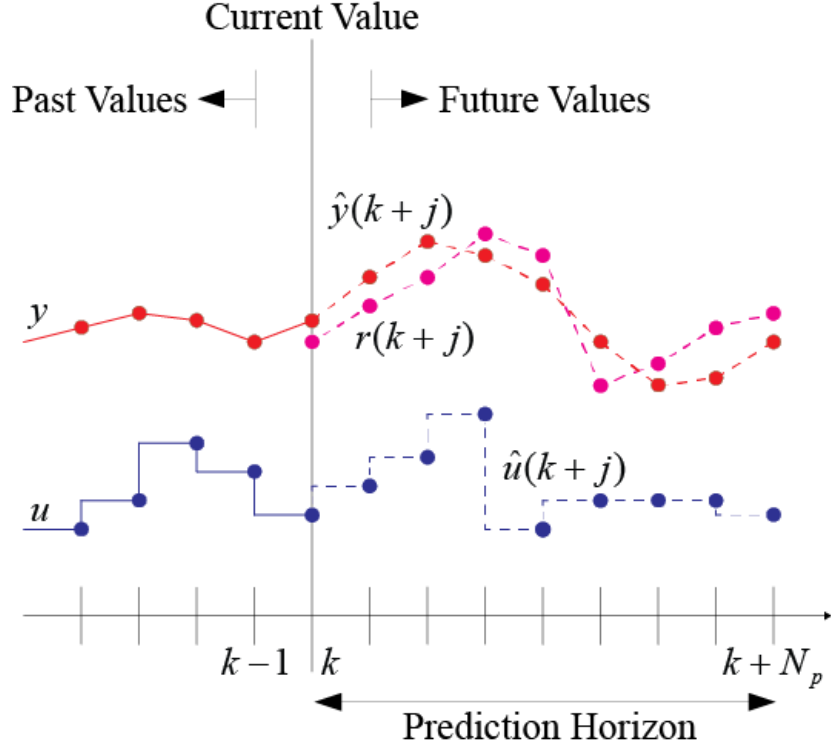


Figure 3.3: The receding horizon idea visualized

$$\min_{U_{k+1}, \dots, U_{k+N_p-1}} (x_{N_p})^T Q(x_{N_p}) + \sum_{j=0}^{N_p-1} P(x_{k+j}, U_{k+j}) \quad (3.5)$$

subject to

$$\begin{aligned} x_{k+j+1} &= f(x_{k+j}, U_{k+j}) \\ x_{k+j} &\in \mathcal{X} \\ u_{k+j} &\in \mathcal{U} \\ x_{N_p} &\in \Gamma \end{aligned} \quad (3.6)$$

The constraint set, Γ , is generally difficult to determine for flight vehicles due to variations in dynamics. Techniques to do so, while guaranteeing stability, are outside the scope of this research. The nonlinear system results in a nonconvex problem. Some popular approaches to solving this problem are to use Sequential quadratic programming (SQP), Euler-Lagrange and Hamilton-Jacobi-Bellman algorithms [33].

SQP is a method that models nonlinear programming (NLP) at a given approximate solu-

tion by a quadratic programming subproblem. The approximated solution of the subproblem is used to construct a better approximation. By iteration a sequence of approximations are created that are hoped to converge to a solution [34]. The Euler-Lagrange and Hamilton-Jacobi-Bellman techniques circumvent solving the nonconvex optimization problem by taking advantage of the nonlinear MPC problems formulation as an optimal control problem. The Euler-Lagrange method numerically solves the two point boundary value problem that arises from Pontryagin's Maximum Principle and the Hamilton-Jacobi-Bellman approach tries to numerically solve the Hamilton-Jacobi-Bellman partial differential equation [23]. SQP is the method by which the proposed MPC will solve its optimization problems. SQP was chosen as a solution method for three reasons: its ability to iterate through an infeasible solution space, being able to be modified such that all linear constraints are always satisfied and, the existence of many algorithms that solve quadratic programming problems [34].

3.3 Chapter Summary

MPC is a form of optimal control that predicts how a model will respond to inputs and drives the model based on these predictions to minimize a cost function. The minimization can be subject to both linear and nonlinear inequality and equality constraints that reflect the operating limits of the system. A receding horizon is used to close the MPC loop, so that it responds to feedback from the plant, and can minimize an error. There are several optimization methods used to solve nonlinear optimal control problems the one employed in this work is SQP.

CHAPTER 4

SIMULATION SETUP

4.1 Trimming the Aircraft

Trimming an aircraft is finding an equilibrium point for a given flight condition. Generally this means that the aircraft control inputs are set in such a way that the craft experiences steady level flight. To find the combination of control inputs to achieve such flight an optimization routine is used to satisfy Equation 4.1 where δ is a vector of control surface deflections, thrust vectoring, and thrust.

$$\min_{\delta} \sum_{i=1}^{14} \delta_i^2 \quad (4.1)$$

subject to

$$\delta \in \Delta$$

and Equations 4.2-4.7.

The constraints, imposed to achieve level flight in terms of state variables are as follows:

1. The difference between the actual velocity and the desired velocity, V_d , must be zero.

$$u^2 + v^2 + w^2 - V_d^2 = 0 \quad (4.2)$$

2. The flight path angle, γ , must be zero.

$$\gamma = \theta - \arctan(w/u) = 0 \quad (4.3)$$

3. The linear accelerations must be zero.

$$\dot{u} = rv - qw - g \sin(\theta) + F_{XB}/m = 0 \quad (4.4)$$

$$\dot{v} = pw - ru + g \cos(\theta) \sin(\Phi) + F_{YB}/m = 0 \quad (4.5)$$

$$\dot{w} = qu - pv + g \cos(\theta) \cos(\Phi) + F_{ZB}/m = 0 \quad (4.6)$$

4. The angular accelerations must be zero.

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = - \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix}^{-1} \left\{ \begin{bmatrix} 0 & -r & q \\ r & 0 & -p \\ -q & p & 0 \end{bmatrix} \begin{bmatrix} I_{xx} & I_{yy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} \begin{bmatrix} p \\ q \\ r \end{bmatrix} \right\} + \begin{bmatrix} I_{xx} & I_{yy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix}^{-1} \begin{bmatrix} L \\ M \\ N \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix} \quad (4.7)$$

4.2 MPC Properties

Closed loop simulation of the ICE model and MPC was done using MATLAB. The reference, $R(k)$, followed for all simulations is a series of discrete state vectors with commanded step changes. Where the state vector, X is defined by Equation 4.8.

$$X = [x_I \ y_I \ z_I \ u \ v \ w \ \Phi \ \theta \ \Psi \ p \ q \ r]^T \quad (4.8)$$

The input to `fmincon` is a column vector denoted Z_L , which will be referred to as the long vector. The elements of the long vector are the initial state vector repeated H_p times concatenated with the initial input vector repeated H_p times. After the first control input is found, the next iteration is warm started using the previous solution. i.e. $Z_L = x_k, u_k$. Warm starting the algorithm near the last solution is done in hopes that the next solution will not be far from the last, and computation time will be reduced.

The maximum magnitude of the bounds of each state were used to normalize the inputs to `fmincon`, a constrained gradient descent based optimization tool developed by MATLAB. When normalized the convergence rate of gradient descent is better bounded because the ratio of largest and smallest eigenvalues of the Hessian is small, aiding `fmincon` in converging more quickly and

more accurately [35].

4.2.1 The Internal Model

The general formulation of a continuous time nonlinear system in Equation 4.9 can describe the ICE aircraft, and will be used as the internal model the MPC uses for prediction. In this formulation let the input vector U be comprised of the six force and moment coefficients. Representing the system as Equation 4.9

$$X(t) = f(X(t), U(t)) \quad (4.9)$$

Where this is a vector of the functions described in Equations 2.2 - 2.5. During the search for a solution to this problem the optimizer only predicts what will happen when certain force and moment coefficients are applied, to Equations 2.2 and 2.3 the solver does not consider how they will be built from the aerodynamic data tables.

The internal model the MPC evaluates is a 6DOF continuous time representation of the aircraft without any aerodynamic data. In order to obtain discrete approximations of the continuous time state equations discussed in Chapter 2, two numerical methods are used: Runge-Kutta 45 and collocation [36]. The first partition, $k + 1$, is evaluated using the variable step solver Runge-Kutta 45. This was chosen to increase accuracy of the first partition step to more accurately predict the control input that will be applied. The remaining partitions are predicted using collocation, sacrificing accuracy for speed because the remaining partitions are only used to anticipate the aircrafts response. The difference between the internal model and the simulation model are the rate at which they are sampled. The internal model is sampled at some rate T_s while the simulation model is solved by MATLABs ode45 function, which is a variable step solver that provides more accurate outputs, across much smaller time steps.

4.2.2 Pseudo Inputs

The objective of the MPC is to track the reference vector, by optimizing across a prediction horizon, H_p , and applying optimal control until a control horizon, H_c , after which the control

input is held constant. The pseudo inputs used to track the reference consist of aerodynamic force and moment coefficients, thrust, pitch thrust vectoring, and yaw thrust vectoring. The variable T contains the three thrust inputs and is contained within U . The cost, J_1 , was selected to minimize state error between the reference and actual states, while minimizing the thrust inputs. Assuming full state information is available,

$$\begin{aligned}
 J_1 = & (X_{H_p} - R(k + H_p))^T Q (X_{H_p} - R(k + H_p)) \\
 & + \sum_{j=0}^{H_p-1} [(X_{k+j} - R(k+j))^T Q (X_{k+j} - R(k+j))] + \sum_{j=0}^{H_c-1} T_{k+j}^2
 \end{aligned} \tag{4.10}$$

J_1 was selected such that the cost is convex and quadratic to make it suitable to a minimization problem. Both the states and references are normalized by their maximum values as before. By doing so the states whose maximum magnitudes are orders larger than those of the smallest states will not dominate the direction the cost function travels, allowing for more intuitive choices of the weighting matrix, Q which is a 12 x 12 diagonal matrix whose diagonal elements are

$$\begin{array}{cccc}
 Q_{1,1} = 0 & Q_{2,2} = 5 & Q_{3,3} = 50 & Q_{4,4} = 0 \\
 Q_{5,5} = 1 & Q_{6,6} = 1 & Q_{7,7} = 1 & Q_{8,8} = 1 \\
 Q_{9,9} = 1 & Q_{10,10} = 1 & Q_{11,11} = 1 & Q_{12,12} = 1
 \end{array}$$

and whose off-diagonal elements are zero. Elements of Q were chosen based on an iterative tuning process. Notice that the 3,3 element of Q is heavily penalized, this is because the inputs to the cost function are scaled by the maximum magnitudes of their bounds. The bound on altitude is much larger than the bounds on the other states of the system. Therefore to penalize altitude appropriately, the weight on that state was increased. This is by no means the only reasonable choice of Q , it can be modified and still be effective depending on what the designer requires. Using J_1 the pseudo inputs are found by solving the following problem with nonlinear equality constraints and nonlinear inequality constraints, c .

$$\min_{U_k, \dots, U_{k+H_c-1}} J_1 \quad (4.11)$$

subject to

$$X_{k+j+1} - f(X_{k+j}, U_{k+j}) = 0 \quad j = 0, \dots, H_p - 1$$

$$X_{k+j} \in \mathcal{X}$$

$$U_{k+l} \in \mathcal{U} \quad l = 0, \dots, H_c - 1$$

$$U_{k+H_p-1} = \dots = U_{k+H_c} = U_{k+H_c-1}$$

where \mathcal{X} and \mathcal{U} are closed and convex, U is held constant past H_c and

$$\begin{bmatrix} \alpha_{k+j+1} \\ -\alpha_{k+j+1} \end{bmatrix} \leq \begin{bmatrix} 45 \\ -5 \end{bmatrix}$$

The constraints enforced in this optimization are:

1. The original nonlinear dynamics must be satisfied at the collocation points.
2. α cannot exceed 45°
3. α cannot be less than -5°

The additional constraints on α were selected so that the craft did not operate outside the bounds of the aerodynamic table data, increasing the chance that the control allocation solution will be feasible.

The solver proceeds in this manner until the predicted states match the internal model states. When this occurs the solver outputs its final estimate of what the aerodynamic force and moment coefficients to drive the aircraft to the reference state.

4.2.3 Control Allocation

In order to determine the control deflections necessary to produce the obtained force and moment coefficients, a second optimization is performed after the first.. It takes in vector C_L which

is comprised of a single set of control deflections representing the control deflections to produce the first partition's pseudo inputs. C_L is normalized similar to Z_L for the same reasons. This routine iterates over the aerodynamic derivative tables attempting to find a C_L that minimizes the cost function J_2 . Within J_2 , C_{FM} is a vector of force and moment coefficients which results from the optimizer's choices of C_L through the aerodynamic coefficient tables, the matrix A contains weights, and δ_k is the current control surface deflections

$$J_2 = (C_{FM} - U_k)^T A (C_{FM} - U_k) + (\delta_k - \delta_{k-1})^2 \quad (4.12)$$

Because the craft is a tailless fighter, and has redundant control surfaces, there are multiple solutions to every set of coefficients. To try and reduce the number of acceptable solutions the cost function penalizes the difference between the previous control input and the new one. As before the cost function is quadratic as to make it convex. The weighting matrix A is a 6x6 identity matrix, so that no particular coefficient is favored.

Unlike the first optimization there are no nonlinear constraints, but linear bounds on the limits of the control surfaces. The problem that must be solved is then

$$\min_{\delta_k} J_2 \quad (4.13)$$

subject to

$$\delta_k \in \Delta \quad (4.14)$$

where Δ is closed and convex. The entire process is summarized in Figure 4.1

In the problem formulation there are no constraints to account for control rates, or engine dynamics. An engine model was not provided by Lockheed Martin for this aircraft, and one was not assumed. Transfer functions for the control surfaces were provided but for this work were not implemented.

4.3 Infeasibility and Suboptimal Solutions

It is not always possible to find an optimal control problem solution in a reasonable amount of time, or at all. For the ICE model if first order optimality is nearly 0 and constraints are not being

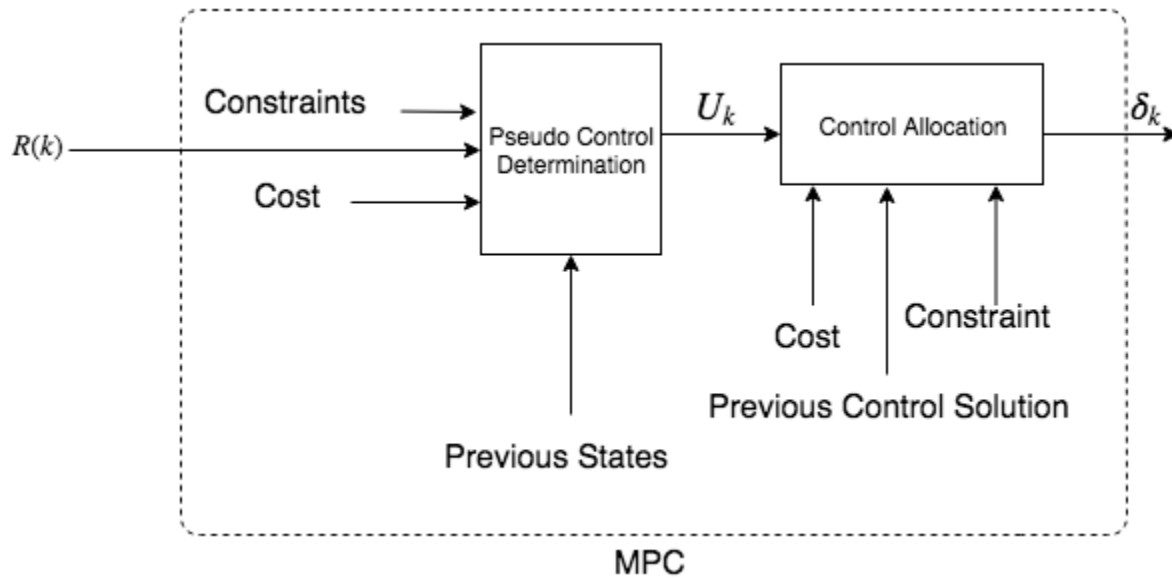


Figure 4.1: A depiction of each optimization, their inputs, and outputs

violated and the cost function is no longer decreasing, a new initial solver guess for the states can be implemented to try and change the optimizers, starting point and trajectory through the solution space to come to an optimal solution. If after a certain number of attempts an optimal solution still isn't found the suboptimal solution is accepted. This process is shown by the block diagram Figure 4.2. Generally speaking, suboptimal and infeasible solutions only occur when the system is forced to operate outside the nonlinear constraints. When suboptimal solutions are employed the subsequent solutions can return to being optimal but, when infeasibility occurs the controller has significant difficulty returning to an acceptable solution, optimal or otherwise.

The reason for infeasible solutions is due in part to poorly constrained pseudo control inputs, and in other part by the shape of the aerodynamic derivative tables. Without knowing what the maximum and minimum magnitudes for pseudo inputs are, which would require a global optimization over the aerodynamic tables for each pseudo input, the first optimization routine may ask for a solution that is not physically attainable. The shape of the aerodynamic tables, and the nonlinear dependence on each other makes it very difficult for the optimizer to find suitable solutions. In many cases there are local minima on the tables themselves, and sharp discontinuities that might cause the solver to chatter across them. Several strategies were considered and explored to address these issues, such as the use of radial basis neural networks to smoothly approximate

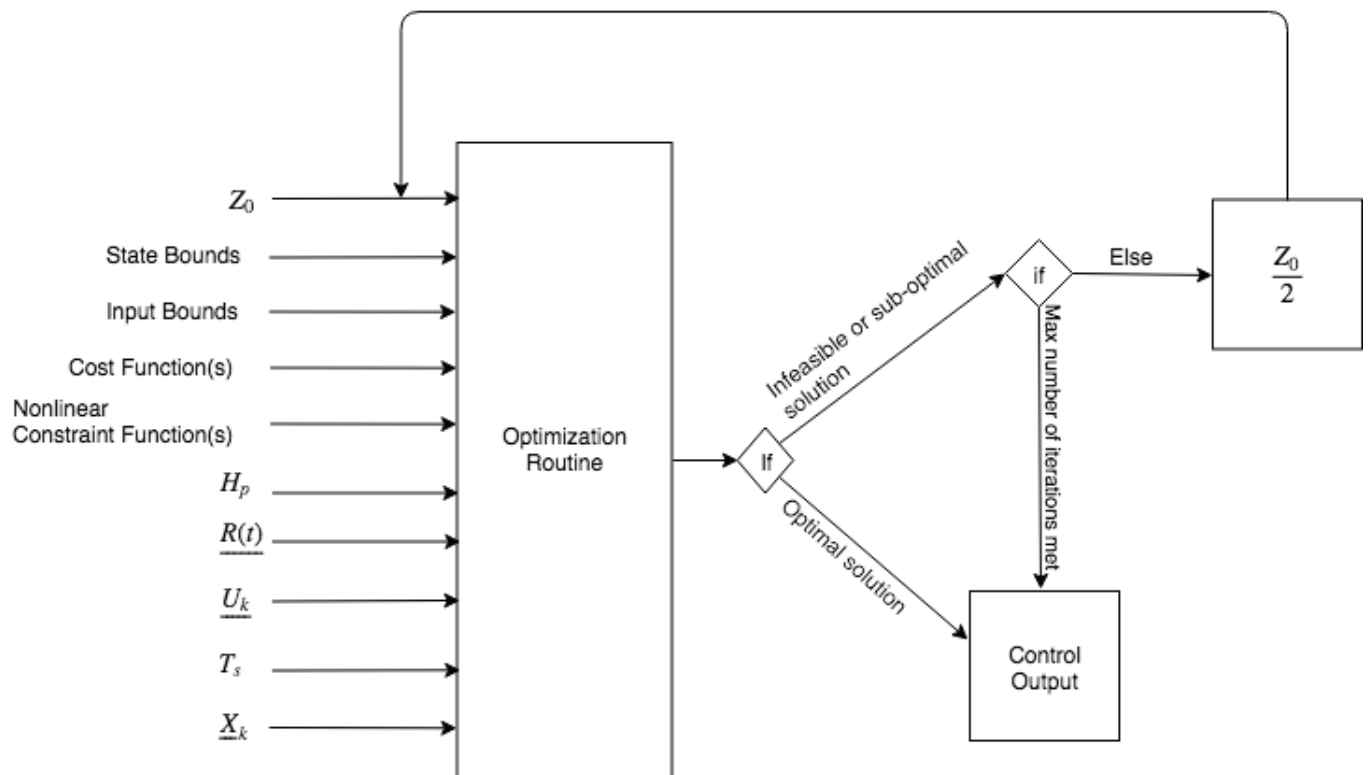


Figure 4.2: The feasibility loop

the tables, or locally linearizing the tables at each partition using a least squares approximation and bounding how far the solver could move on the linear plane. Neither scheme was successful.

4.4 Chapter Summary

The aircraft is trimmed, to find an equilibrium point at which to begin simulation. The conditions for this trim is are an altitude of 3000 ft with a forward velocity of 900 ft/s. The MPC parameters that are used in this work are set forth in section 4.2. The continuous time representations of the aircraft equations of motion are used to represent the internal model of the system and descritized using RK45 and collocation so that the discrete time MPC formulations can be applied. The MPC will only choose pseudo inputs to drive the system, while another optimization routine will choose control surface deflections to realize those pseudo inputs. Infeasibility is avoided and suboptimal solutions are used when necessary as dictated by the logic described in Figure 4.2.

CHAPTER 5

SIMULATION RESULTS

This chapter discusses how the MPC was tuned and the results from three simulation cases are presented. The reference in the first scenario asks the craft to drop in altitude by 1000 ft, climb to 20,000 ft then take step commands to increase altitude by 10,000 ft every 20 seconds until the ceiling of 50,000 ft is reached, referred to as the altitude sweep. The second simulation requires the aircraft to decrease altitude by 1000 ft. Lastly the aircraft is asked to track a 300 foot change in y_I position, and a 1000 ft increase in altitude simultaneously. Each section will begin by displaying the pseudo control inputs, followed by the allocation results, state tracking, climb angle, speed, and cost.

5.1 Initial Control Tuning

To tune MPC the sample time, prediction horizon, control horizon, and the weights of the optimization routine must all be chosen to obtain a desired behavior. The controller was tuned by choosing a sample time based on knowledge of the system. The sample time was chosen to be small enough so that the faster evolving states of the system could still be regulated, but not so small that the slower evolving states do not change. The prediction horizon was then fixed at 5 partitions, making the quantity $T_s H_p$ 1.25 seconds. This is how far ahead in time the controller is looking when choosing control inputs, 1.25 seconds was not chosen by any specific criteria. The weights were initialized such that all weights were equal, and simulations were run until a subjectively acceptable response was found. The control horizon was tuned by starting with a number of partitions equal to that of the prediction horizon, that is $H_p = H_c$ and iteratively reducing it by 1 after simulating the system, until the response was satisfactory. During tuning the aircraft was asked to climb 3000 ft, from 3000 ft to 6000 ft, while minimizing every state except forward

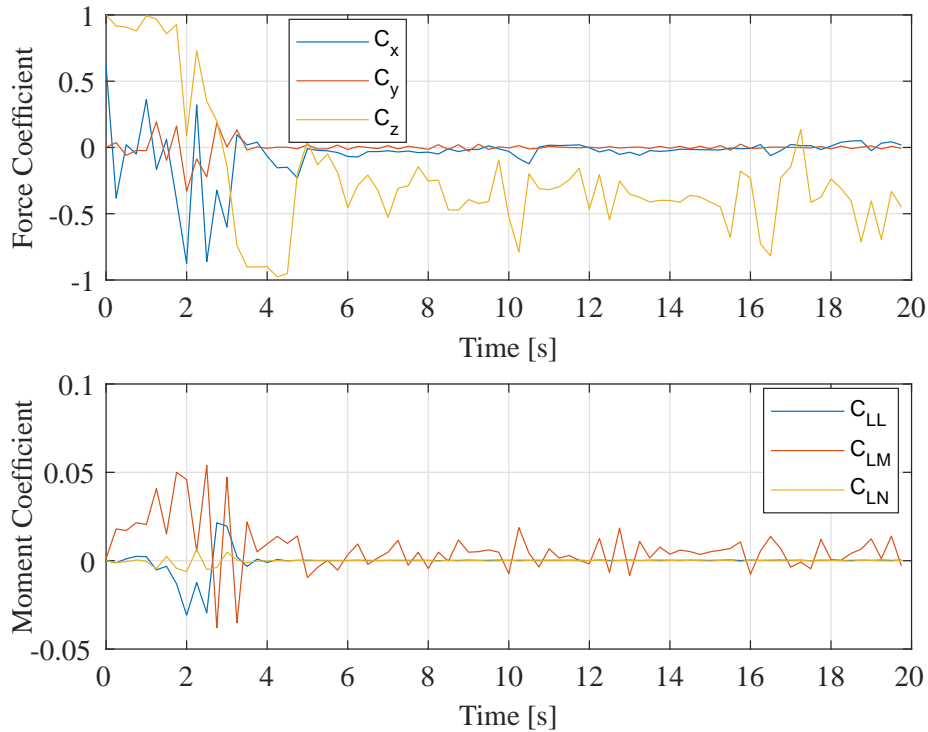


Figure 5.1: Tuning: Commanded force and moment coefficients from the pseudo control optimization

velocity. The results are shown in Figures 5.1 - 5.11.

Figure 5.1 shows the force and moment coefficients commanded by the psuedo controller. The moments are small for a climb, while the C_z gains magnitude to indicate a climb. Figure 5.2 shows the thrust parameters, as expected only pitch thrust vectoring is significant. Figure 5.3 shows significant deflection in the inboard and outboard leading edge flaps, indicating their importance in pitch control. Figure 5.4 shows the all moving wingtip and elevon deflections. Surprisingly there is significant use of the wingtips. This can be attributed to trying to stabilize yaw, by using differential drag. Figure 5.5 shows the spoiler slot and pitch flap deflections. The pitch flap is not as heavily deflected as one might expect, likely due to the heavy use of other control surfaces.

Figures 5.6 - 5.9 show the states for this simulation. Figure 5.6 includes both position and body velocity states, while Figures 5.7 and 5.8 take a close look at the two states with references. There is some overshoot in the altitude track with no sign of converging, but the Y position seems to meet its reference, with small variations. Figure 5.9 shows the euler angles during this motion. As expected the pitch increases, but unexpectedly the bank angle spikes near the beginning. As mentioned before the all moving wingtips are likely activated in response to this.

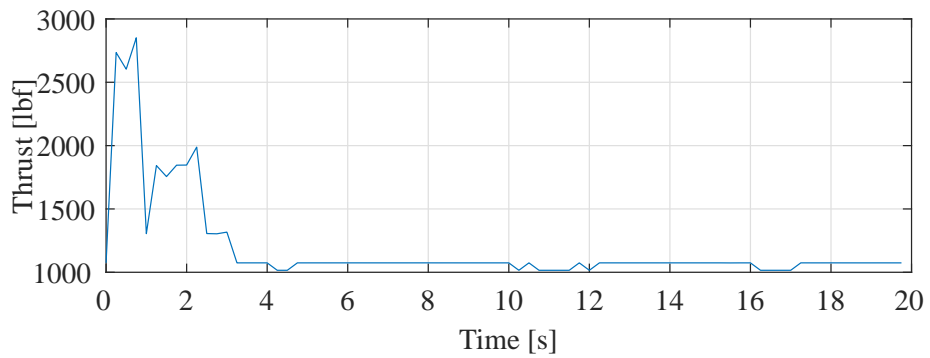
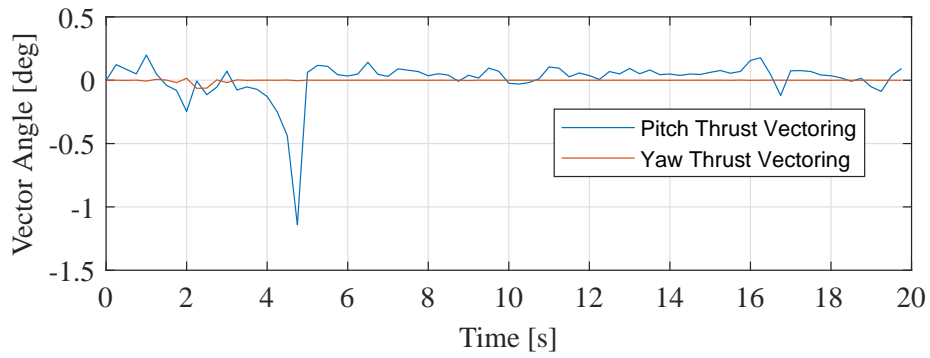


Figure 5.2: Tuning: Thrust parameters

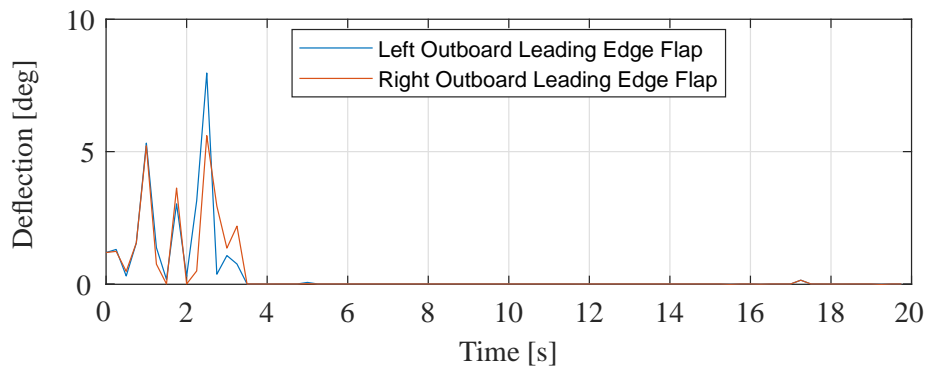
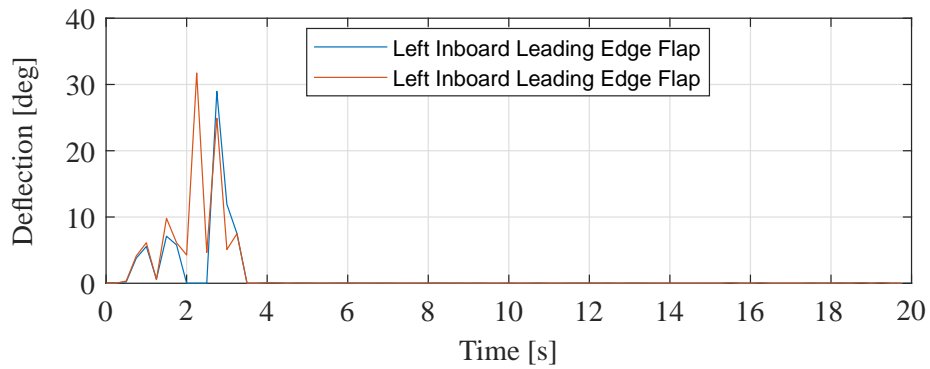


Figure 5.3: Tuning: Inboard and outboard leading edge flap deflections

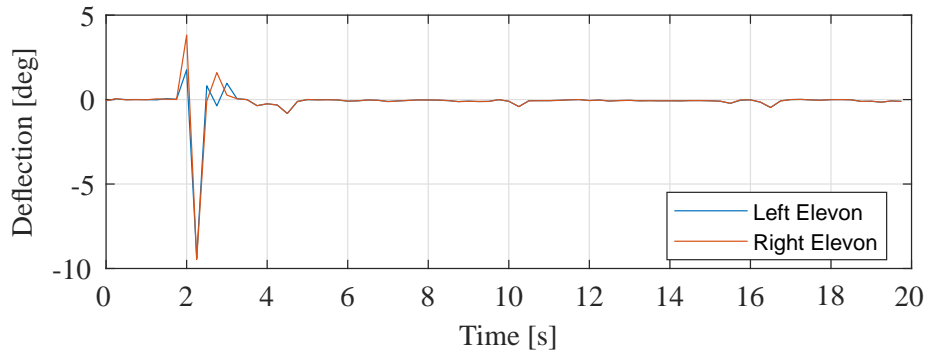
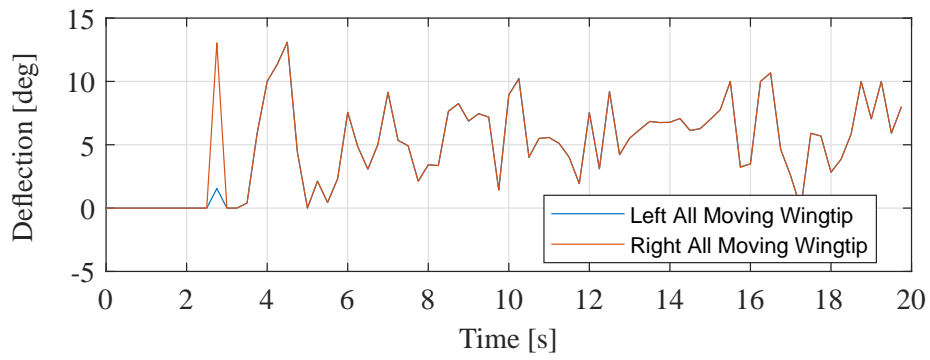


Figure 5.4: Tuning: All moving wingtip and elevon deflections

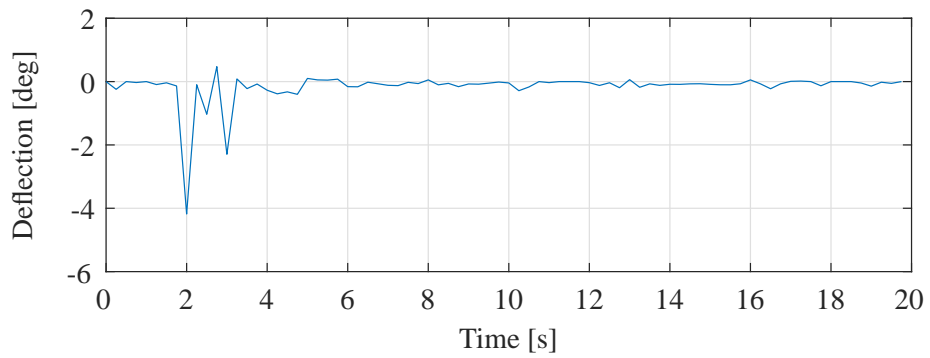
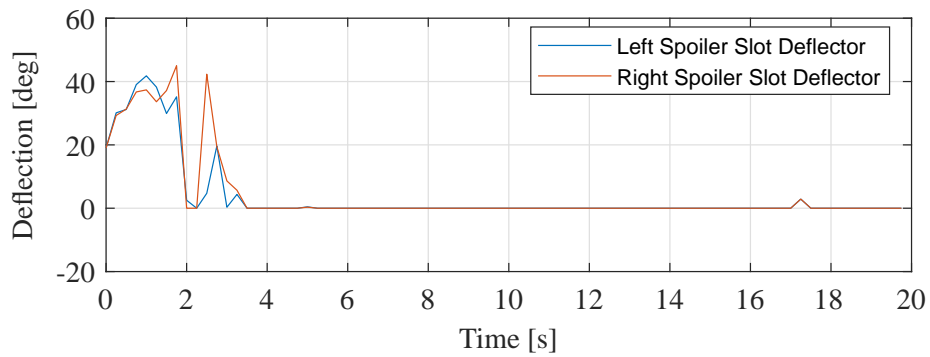


Figure 5.5: Tuning: Spoiler slot and pitch flap deflections

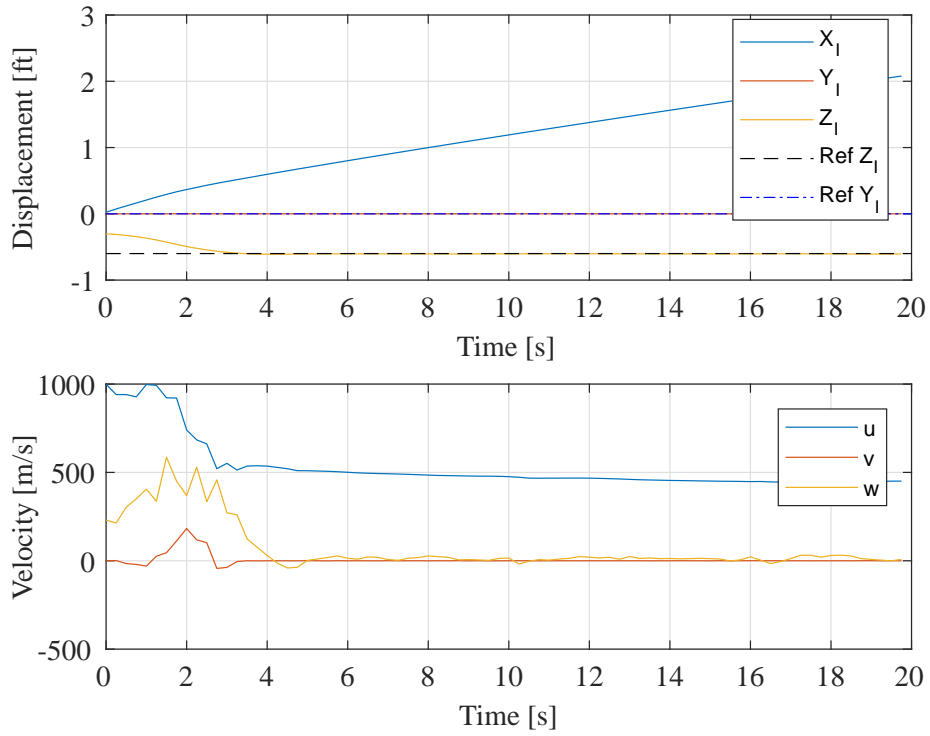


Figure 5.6: Tuning: Positions and velocities

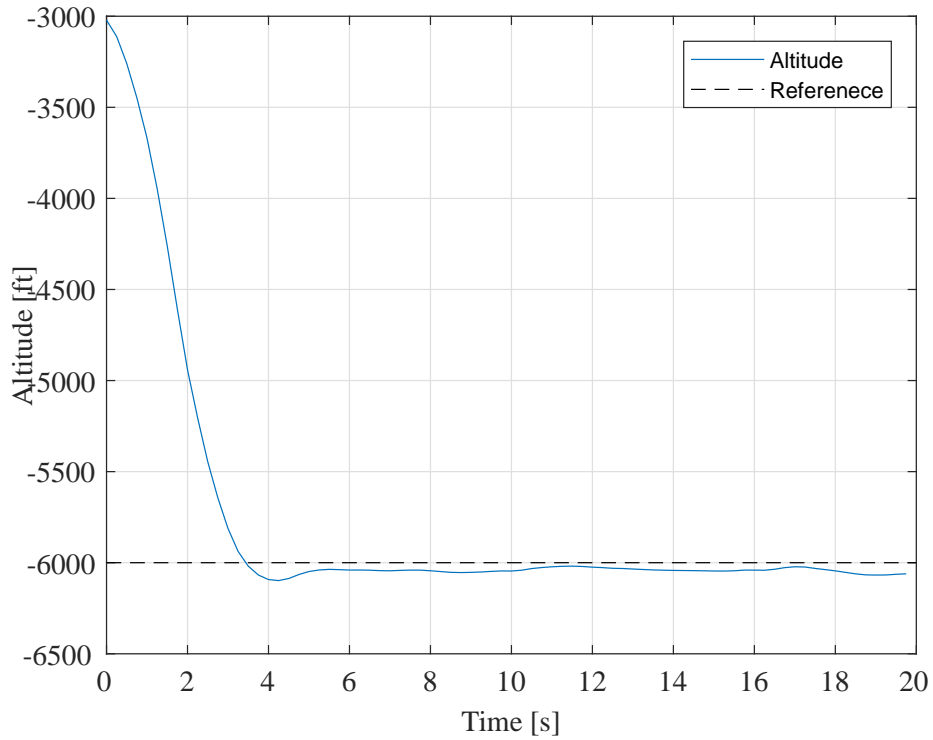


Figure 5.7: Tuning: Altitude tracking

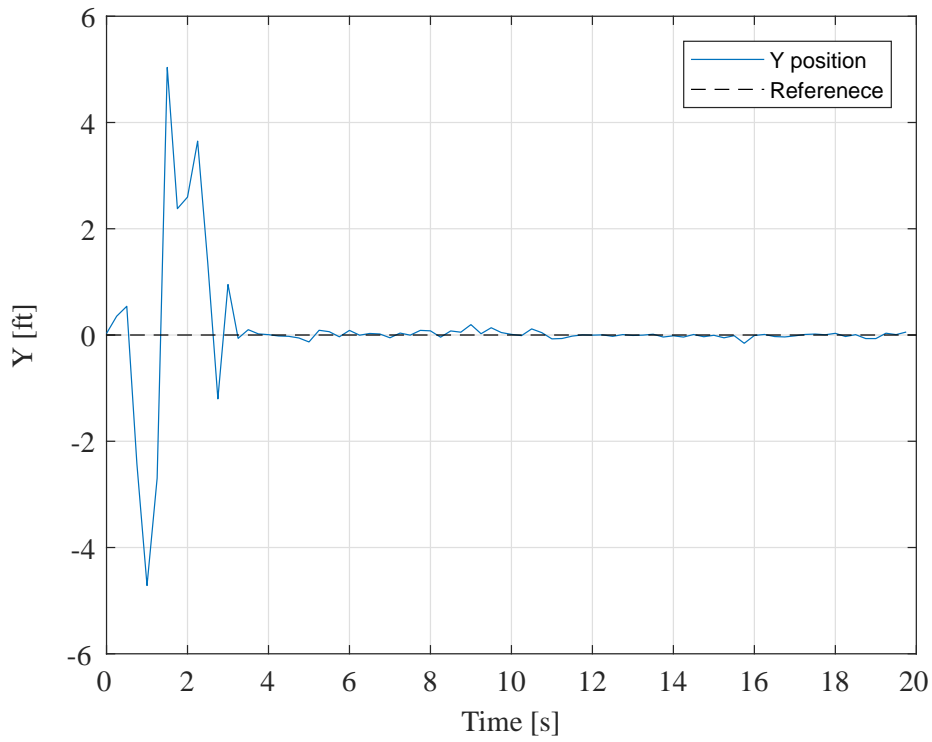


Figure 5.8: Tuning: Y position tracking

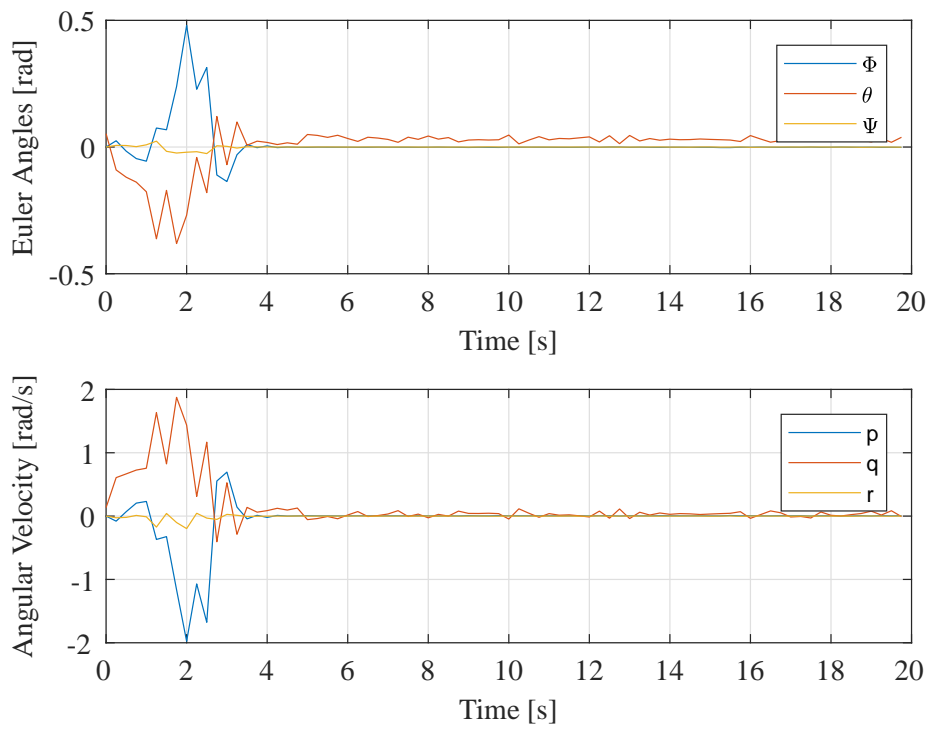


Figure 5.9: Tuning: Euler angles and angular velocities

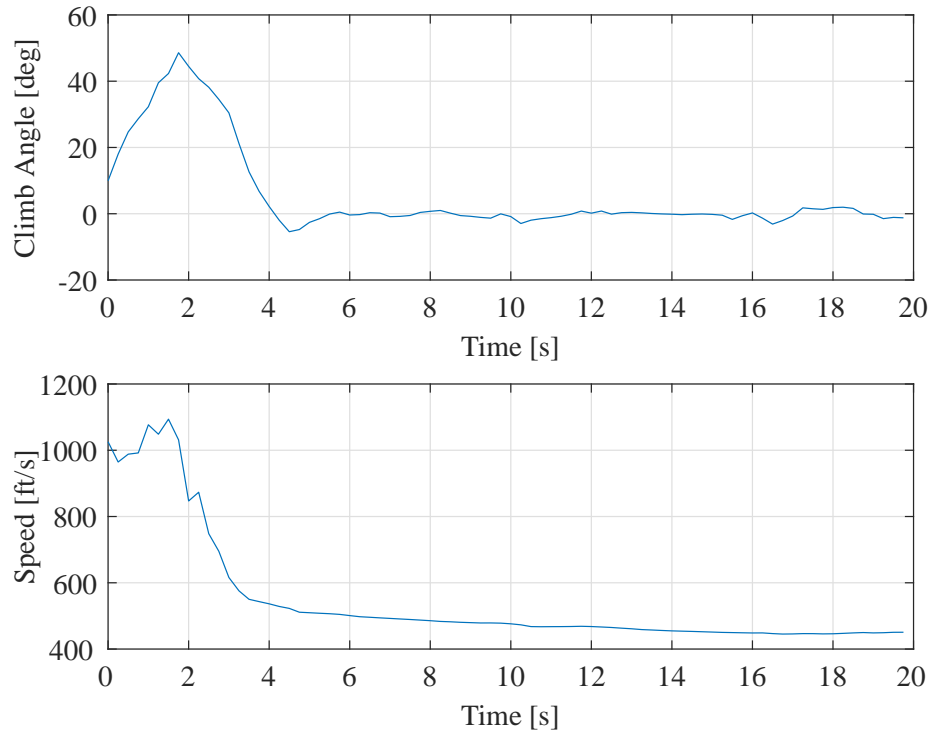


Figure 5.10: Tuning: Climb angle and velocity magnitude

Figure 5.10 shows the climb angle and speed for this motion. It reaches a maximum of 44° before coming back down over the duration of the motion. The speed increases near the beginning of the motion and drops off afterward once the craft has reach its new altitude.

Figure 5.11 shows a normalized cost function for the state tracking error.

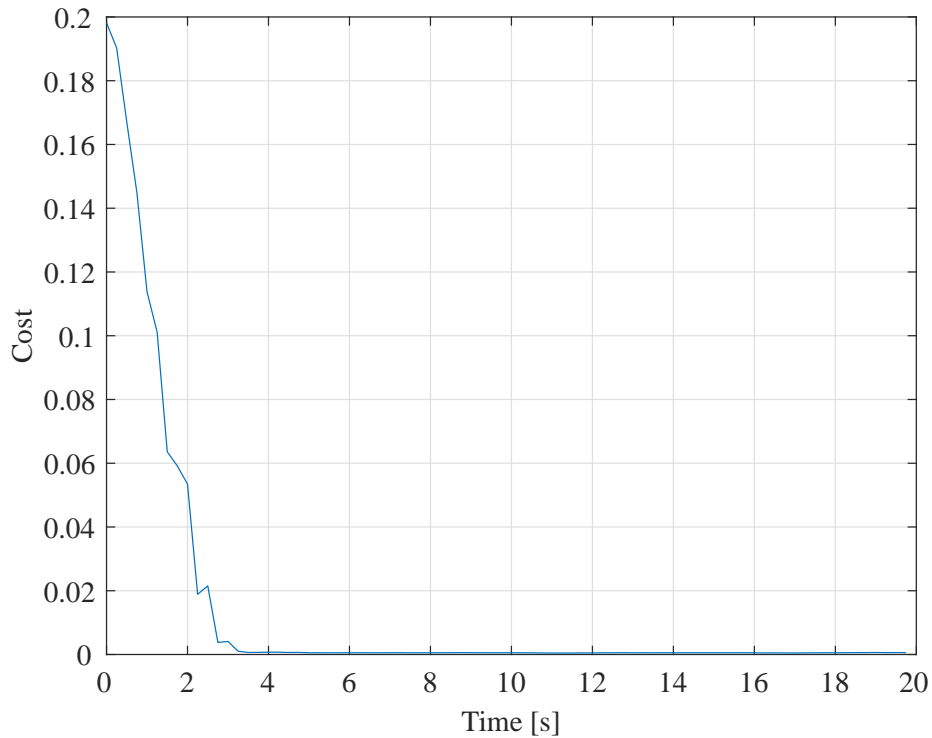


Figure 5.11: Tuning: State tracking cost

5.2 The Altitude Sweep

In this case the aircraft starts out at 3000 ft, is asked to decrease altitude to 2000 ft then climb to 20,000 ft. After which every 20 seconds step commands are given to climb by 10,000 ft until a final altitude of 50,000 ft is reached. The results are displayed in Figures 5.12 - 5.23.

The coefficient plots for this motion are shown in Figure 5.12. They oscillate with high frequencies during any part of the motion that is not a level flight. Figure 5.13 shows the thrust parameters for this motion. The magnitude of thrust increases during the climb phases as well as the thrust vectoring magnitudes. During some motions, the aircraft attempts to create moment to stabilize itself using yaw vectoring. Figure 5.14 shows the usage of inboard and outboard leading edge flaps, they behave as expected based on the tuning case. Figure 5.15 shows the all moving wingtip and elevon deflections. The elevon response seems delayed at the 20 second mark, but as the craft allocates stability to regulate the angular velocities, the elevons are prioritized for pitch control. Figure 5.16 looks at the spoiler slot deflector and pitch flap responses. The pitch flap seems to be underutilized for these motions, while the spoiler slot deflectors are given large commands.

Figures 5.17 - 5.20 show the state evolution for this motion. Figure 5.17 shows the overall

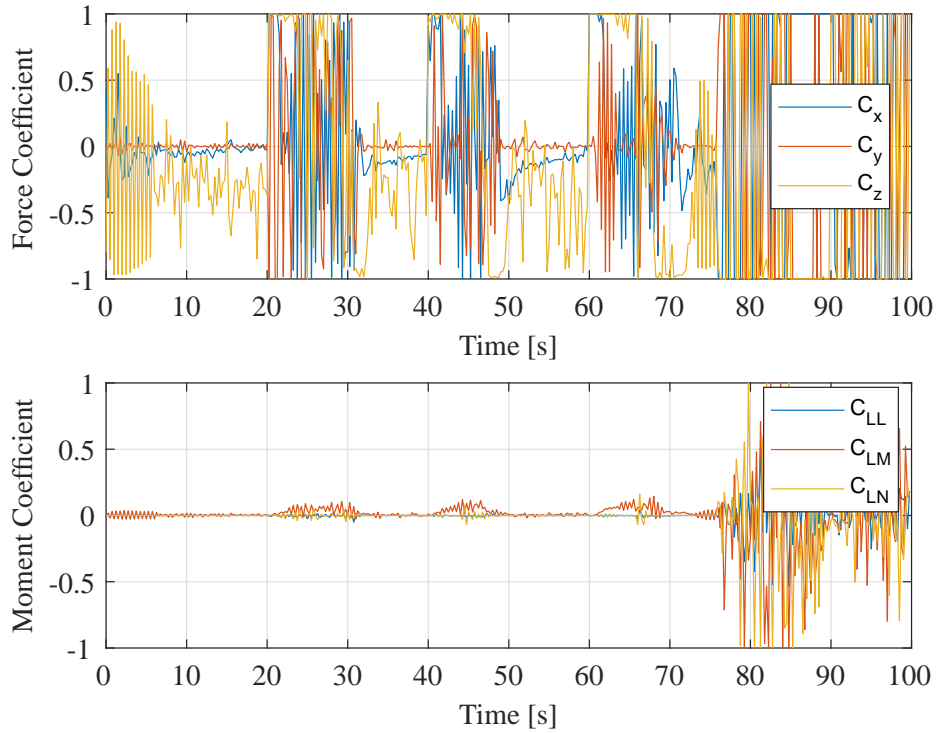


Figure 5.12: Altitude Sweep: Commanded force and moment coefficients from the pseudo control optimization

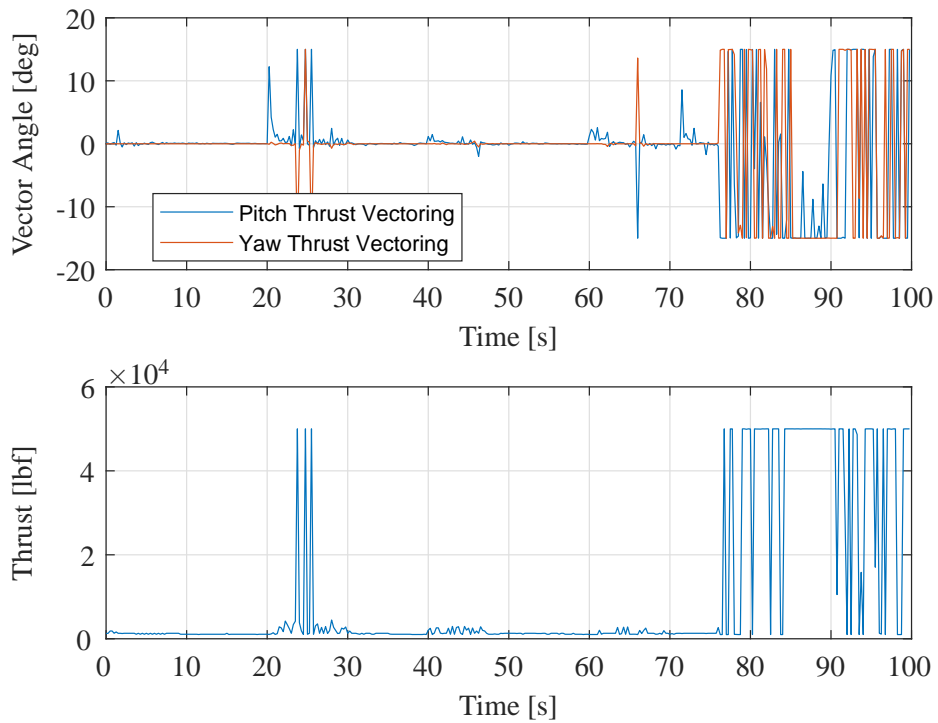


Figure 5.13: Altitude Sweep: Thrust parameters

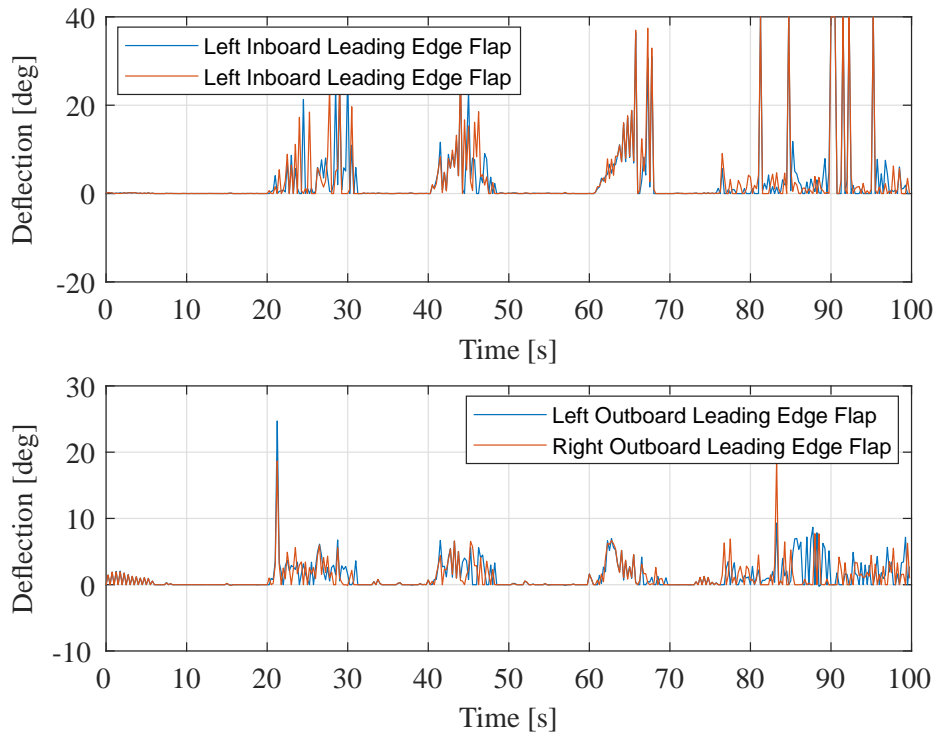


Figure 5.14: Altitude Sweep: Inboard and outboard leading edge flap deflections

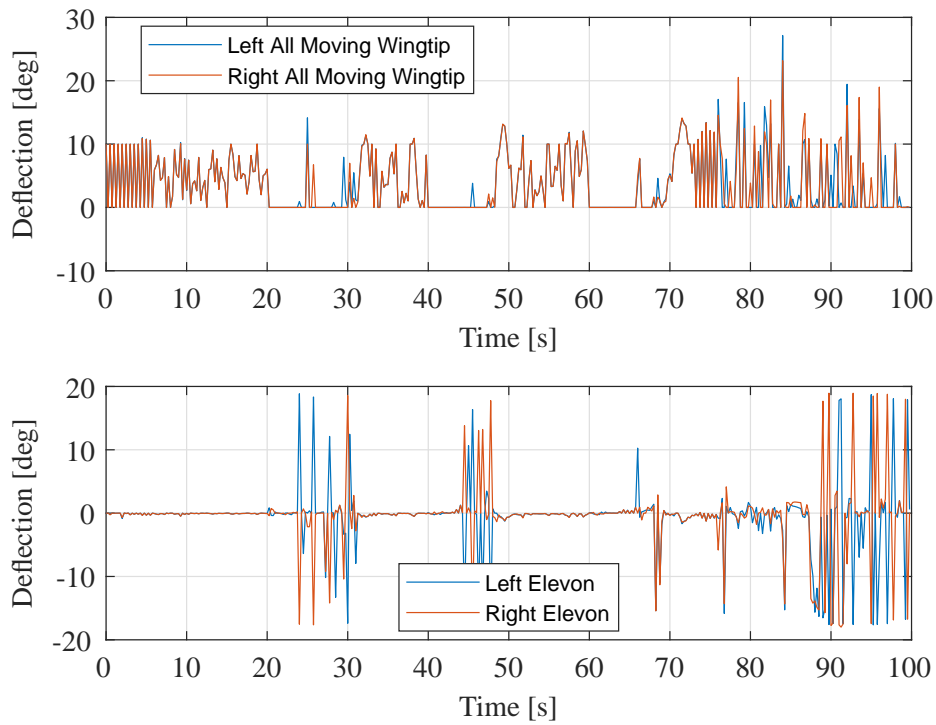


Figure 5.15: Altitude Sweep: All moving wingtip and elevon deflections

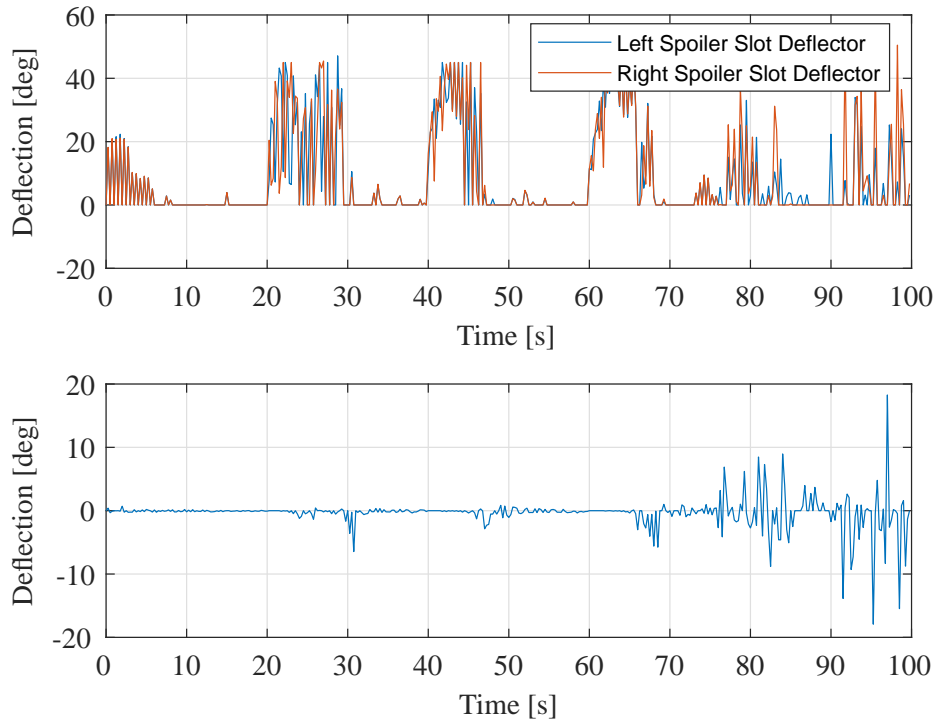


Figure 5.16: Altitude Sweep: Spoiler slot and pitch flap deflections

position of the system plotted against its reference over time and the body velocities of the system. Unexpectedly there are large v and w velocities during each climb. The controller tracks in the Y_I and Z_I directions very well until it gets near the final altitude. The problem was constrained such that a 52,000 ft altitude was the maximum altitude allowable. Operating near that constraint and experiencing overshoot caused the solver to use suboptimal solutions that violated the altitude constraint, which eventually led to infeasible solutions. A close up of just the altitude track is shown in Figure 5.18 and the Y track is shown in Figure 5.19. Figure 5.20 shows the Euler angles and angular velocities, again there are large unexpected magnitudes in the q and r states.

Upon closer inspection we can see that the overshoot is significant at higher altitudes. The system was limited to an angle of attack of 45 degrees, which is reflected by the slope of the climb decreasing as altitude increases. Figure 5.22 shows the angle of attack plotted against time. While α remains close to or at the maximum for most climb sections, as air density decreases, less lift can be produced at that α , and without thrust compensation the craft is unable to keep its rate of climb. A changing dynamic pressure, resulting in less control authority, and an MPC that was tuned at another operation point are likely the causes of this overshoot. Additionally, due to loss

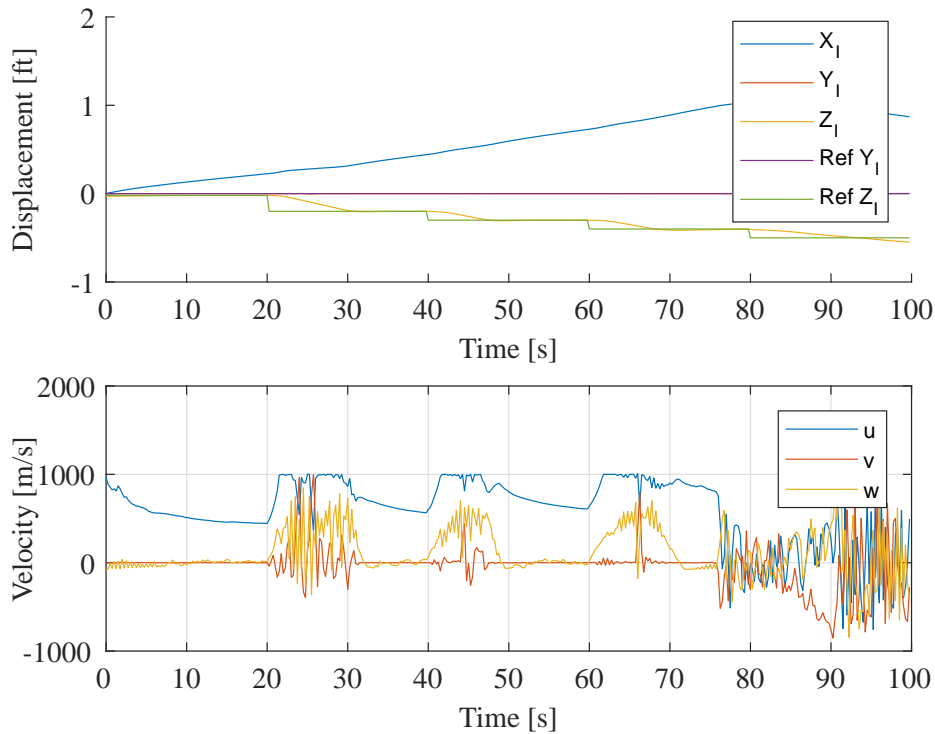


Figure 5.17: Altitude Sweep: Positions and velocities

in control power because air density is changing and dynamic pressure is not being compensated for by increasing velocity, the system response to control inputs is lessened, and the system is responding slower. It was found that if states evolve too slowly the optimization solver will lose its ability to predict across small sample times, indicating a need for an adjustment to sample time to account for control power loss due to dynamic pressure. Near the end of the simulation the only solutions that can be found are suboptimal or infeasible because of exceeding the constraint. This causes the large violations in the angle of attack constraints.

Figure 5.21 shows the climb angle and speed. The climb angle oscillates near 0, and behaves as expected otherwise. The speed track oscillates near the peaks likely in response to thrust input. At each time step the speed seems to approach some value that would be steady state if the system were given enough time. This is speculated on with regards to dynamic pressure in Chapter 6.

Figure 5.22 shows the angle of attack during the simulation. The angle of attack changes rapidly during ascent, and becomes more regular while in between step inputs.

Figure 5.23 shows the cost for the states during the simulation. There are unexpected spikes

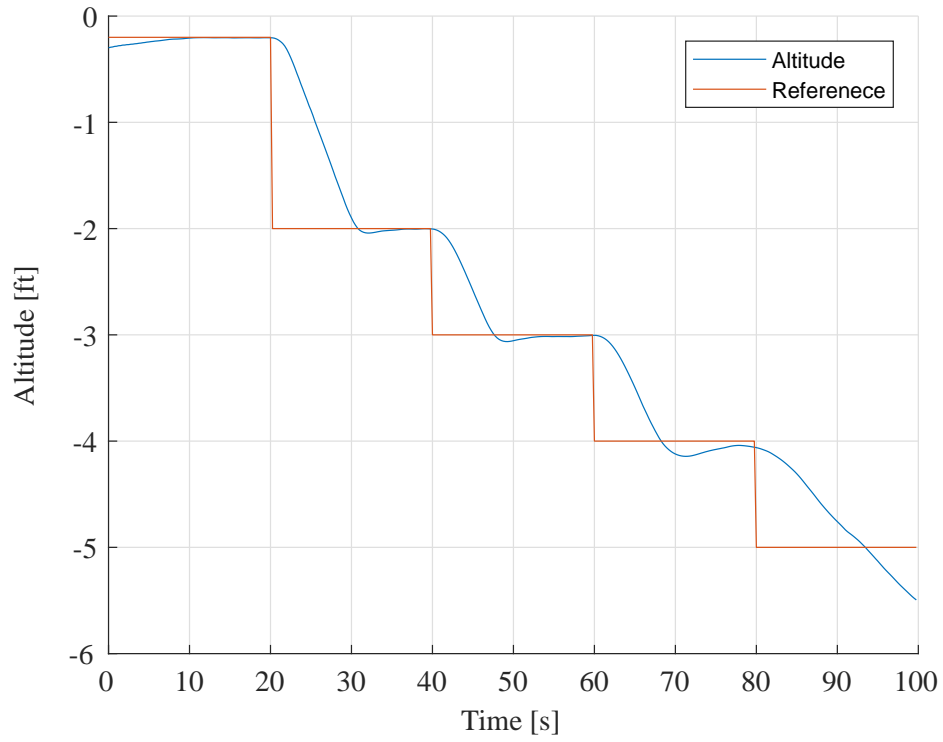


Figure 5.18: Altitude Sweep: Altitude tracking

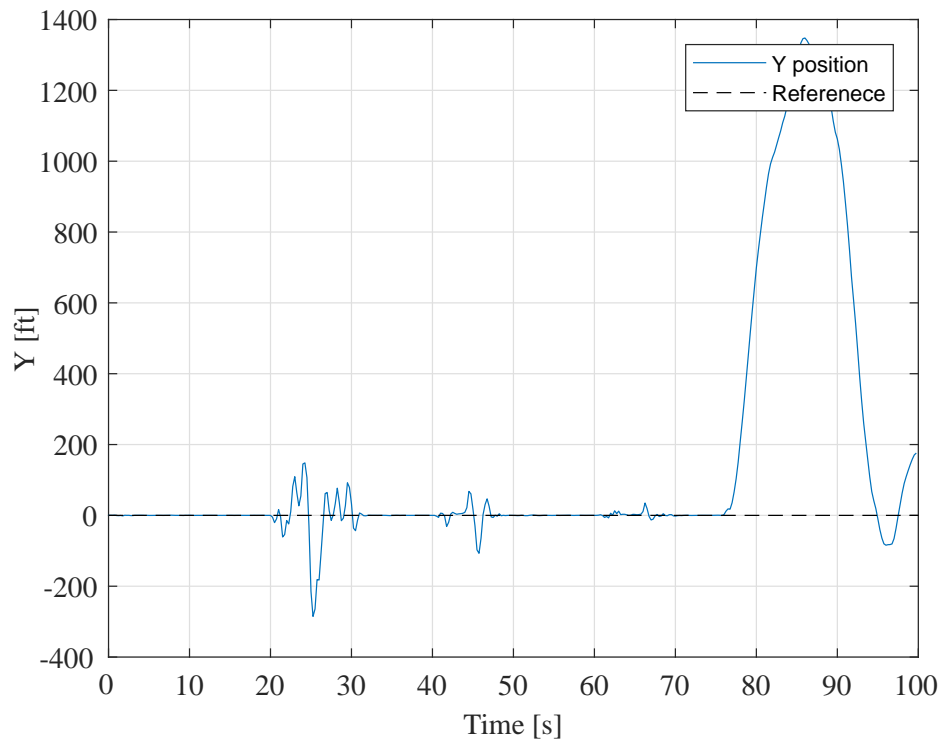


Figure 5.19: Altitude Sweep: Y position tracking

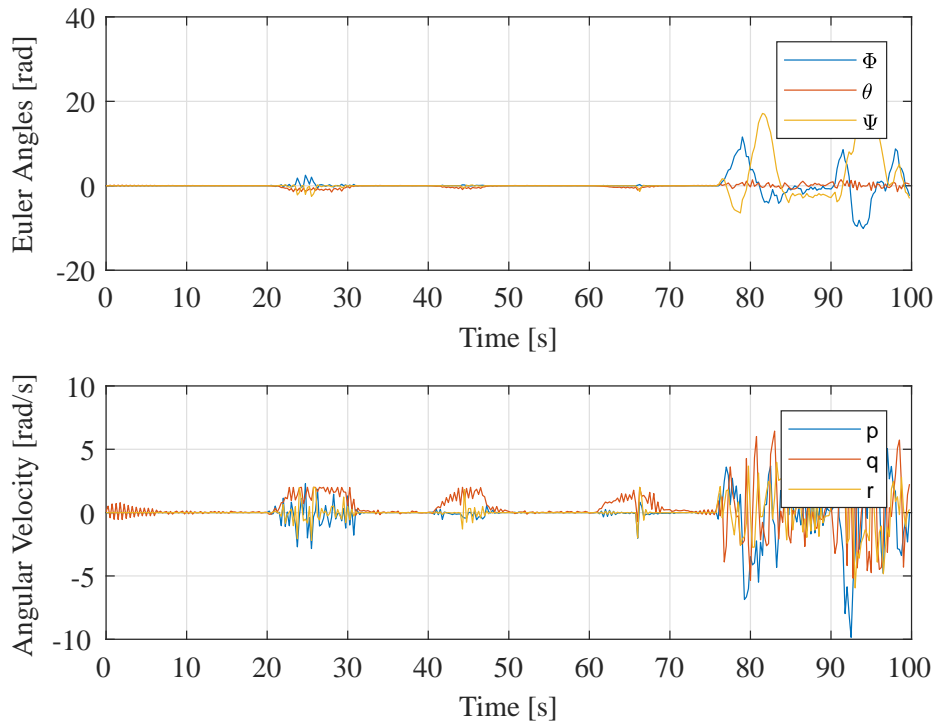


Figure 5.20: Altitude Sweep: Euler angles and angular velocities

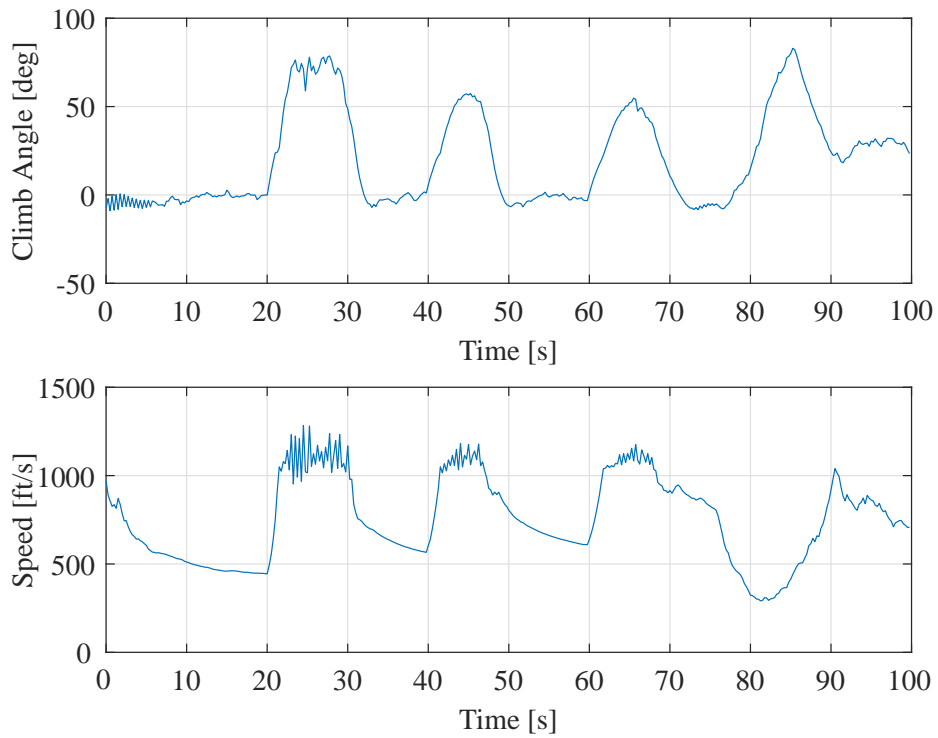


Figure 5.21: Tuning: Climb angle and velocity magnitude

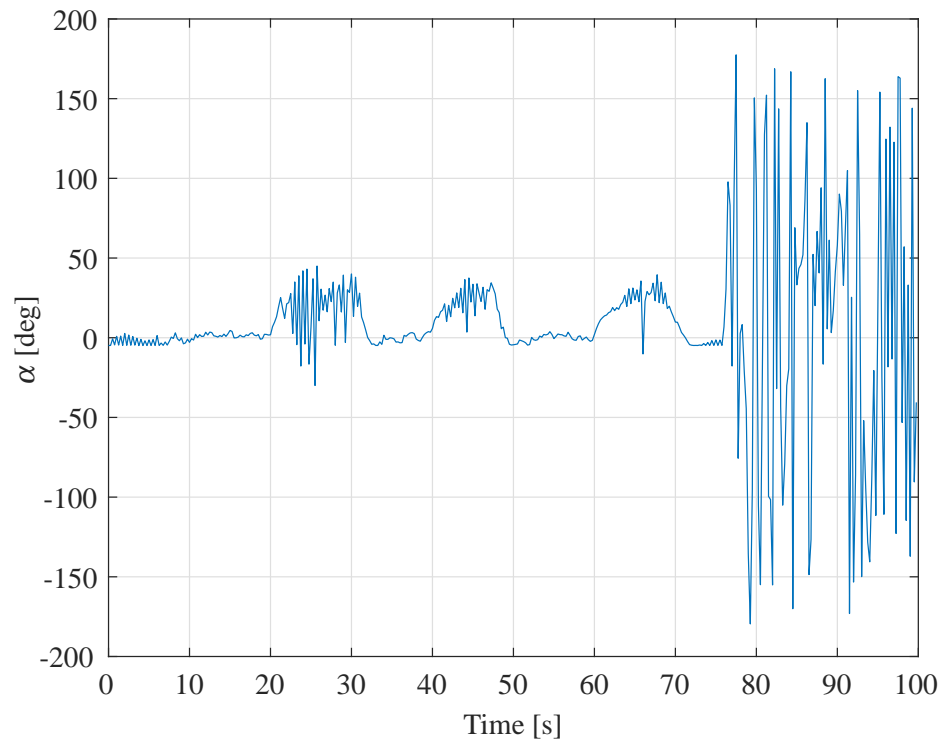


Figure 5.22: Altitude Sweep: α plotted against time

in the cost function during the first ascension but the cost does decrease over time.

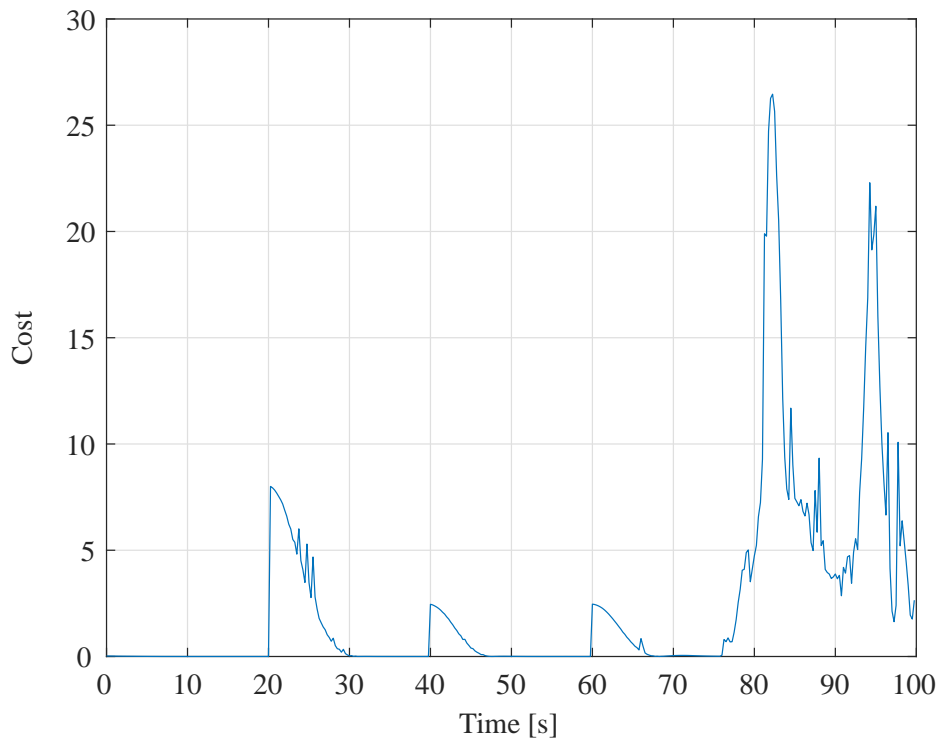


Figure 5.23: Altitude Sweep: State tracking cost

5.3 Descent

In the second simulation the aircraft starts at an altitude of 3000 ft and is asked to drop to 2000 ft. The results from this simulation are shown in Figures 5.24 - 5.34. Because of the lower limit on the angle of attack, and the need to minimize the Euler angles, the aircraft's rate of descent is very small. As previously discussed the lower limit on the angle of attack is due to the limit on the aerodynamic tables. What a pilot might have done in this case is roll the aircraft and pull up, essentially effecting a positive angle of attack while decreasing altitude.

Figure 5.24 shows the commanded pseudo inputs. There is an unexplained divergence in the C_z input that shows up in the state tracking which will be discussed later on. Figure 5.25 shows the thrust parameters for the system. The pitch vector input behaves the same during the first second as it did during ascent, which may be indicative that the system cannot predict the non minimum phase behavior of pitch. Figures 5.26, 5.27 and 5.28 all indicate that during descent the aircraft is relatively stable with little aerodynamic control surface input.

Figure 5.29 shows the position and body velocities. Figure 5.30 shows a better perspective on the altitude track. As mentioned before the C_z coefficient diverged and this is reflected in the

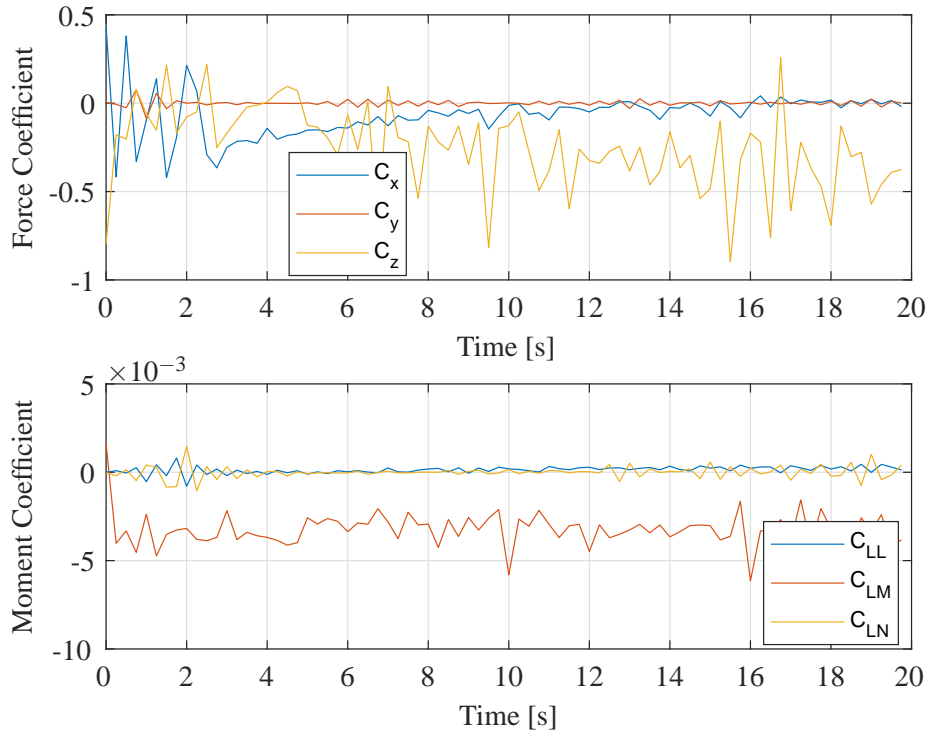


Figure 5.24: Descent: Commanded force and moment coefficients from the pseudo control optimization

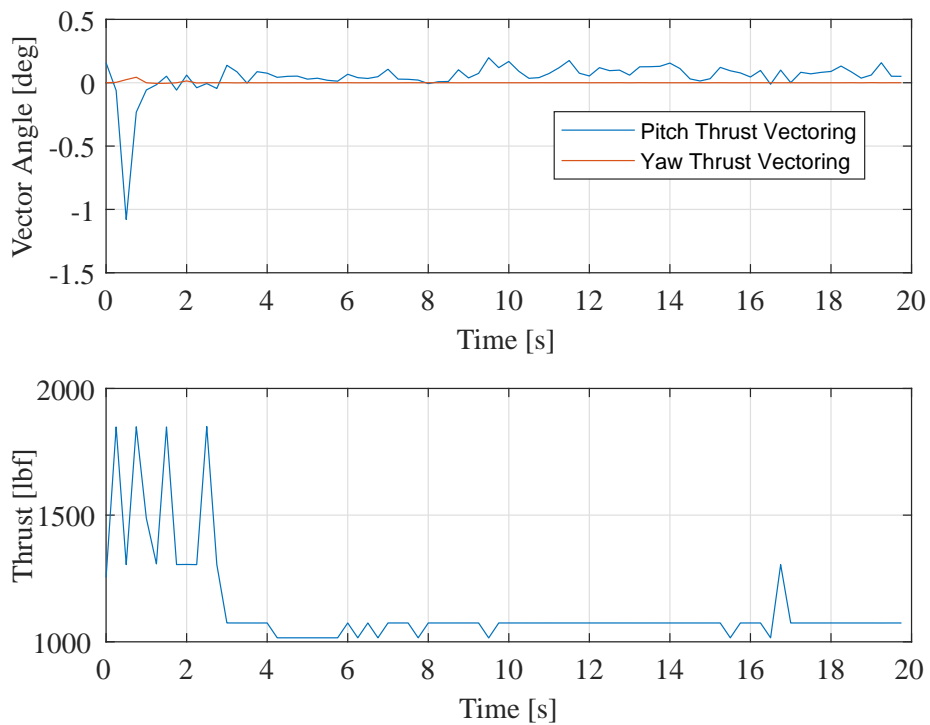


Figure 5.25: Descent: Thrust parameters

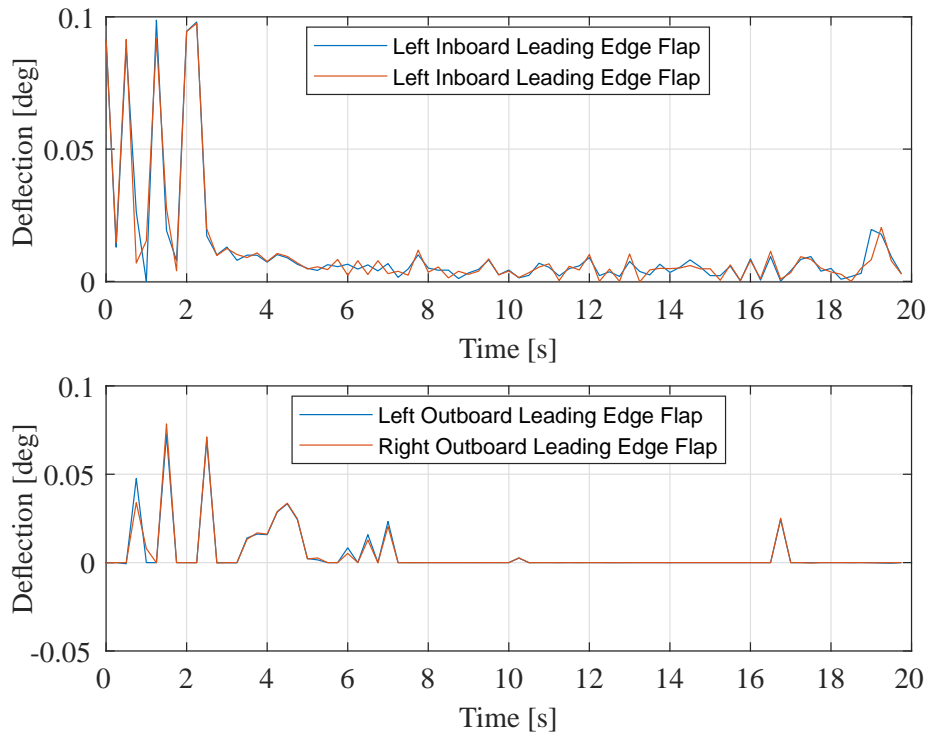


Figure 5.26: Descent: Inboard and outboard leading edge flap deflections

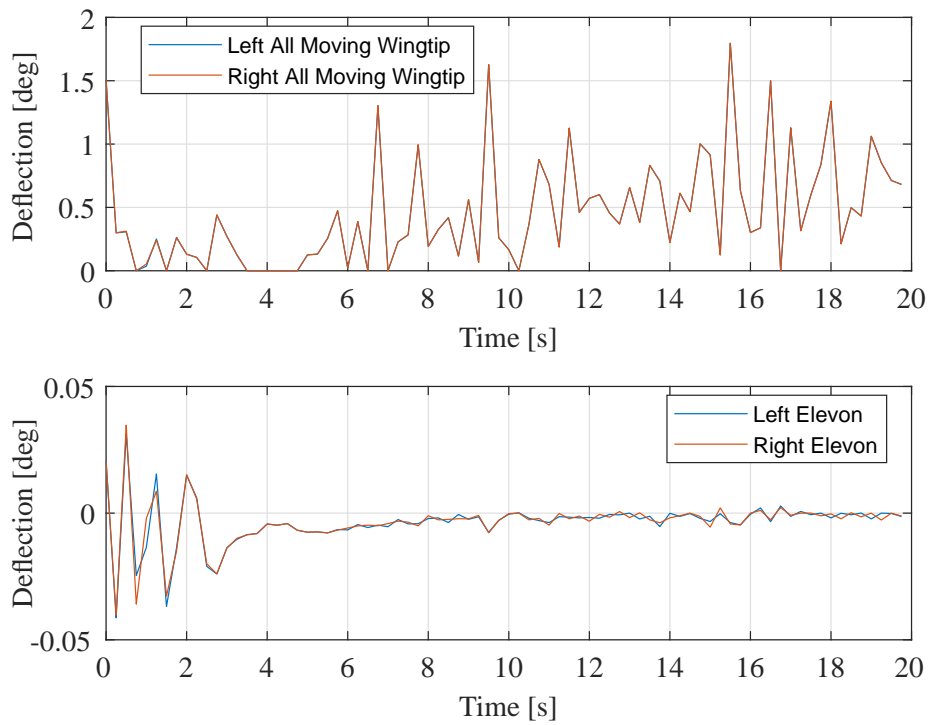


Figure 5.27: Descent: All moving wingtip and elevon deflections

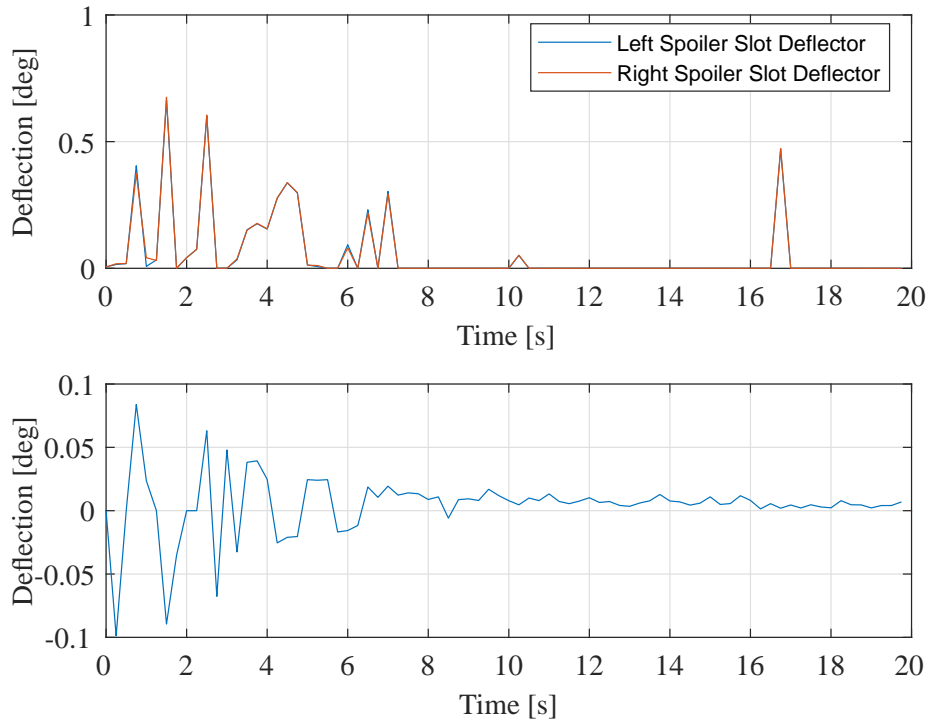


Figure 5.28: Descent: Spoiler slot and pitch flap deflections

altitude position coordinate. The reason for this divergence is likely due to poor tuning of the MPC for a descending motion. There is however, no overshoot while descending this is likely do to how slow the descent is compared to the climb and how much time the MPC has between predicting meeting the reference altitude and actually meeting it. This is indicative that the MPC would benefit from having a larger prediction horizon, given the quarter second sample time.

Figure 5.33 shows the climb angle and speed. The climb angle cannot be as negative as it can be positive due to the angle of attack constraint. The speed drops off very quickly as the aircraft glides to its refernce altitude.

Figure 5.34 shows the normalized cost for state tracking during the motion.

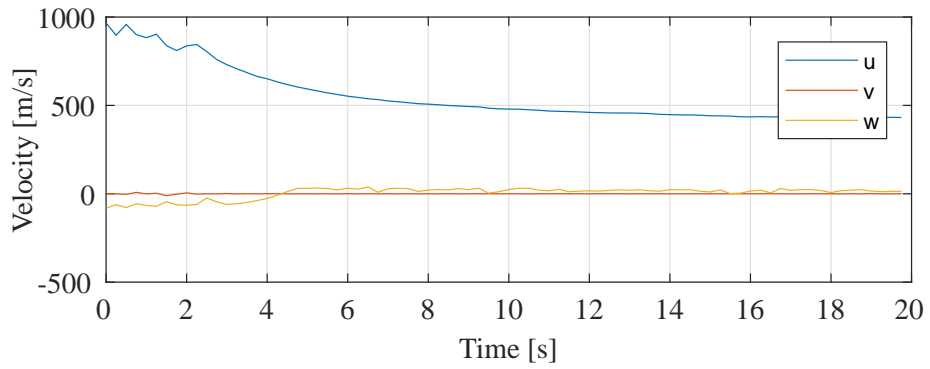
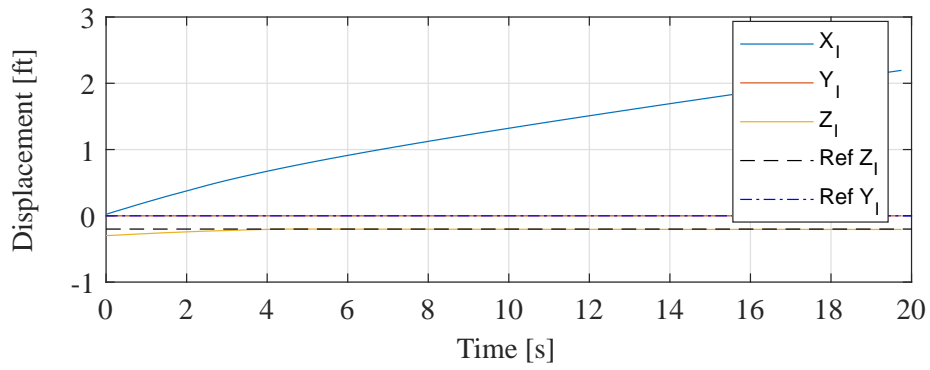


Figure 5.29: Descent: Positions and velocities

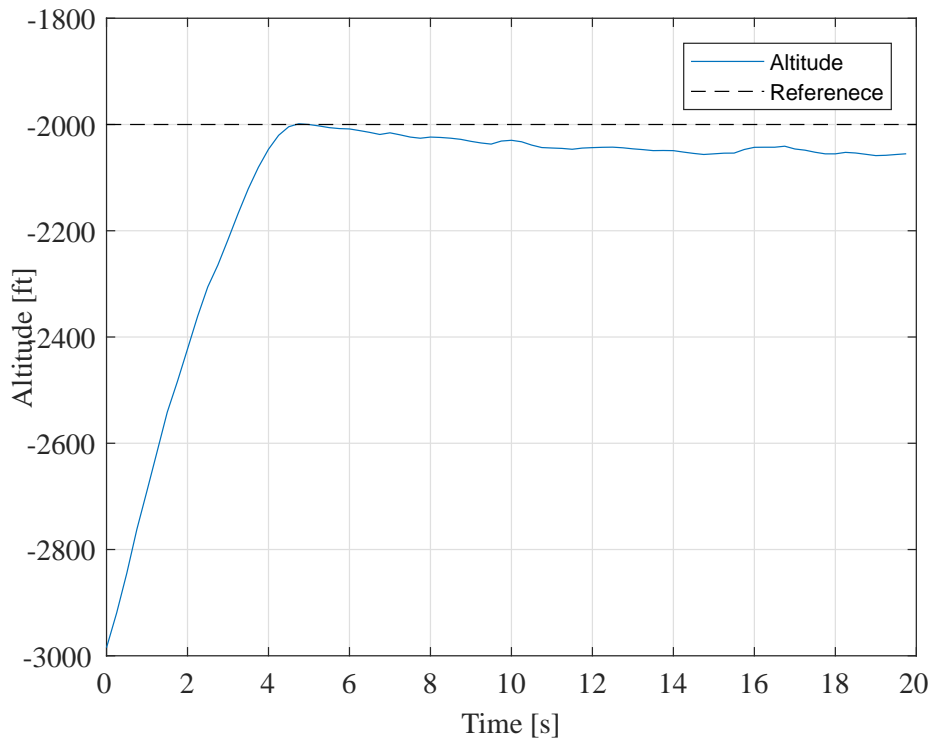


Figure 5.30: Descent: Altitude tracking

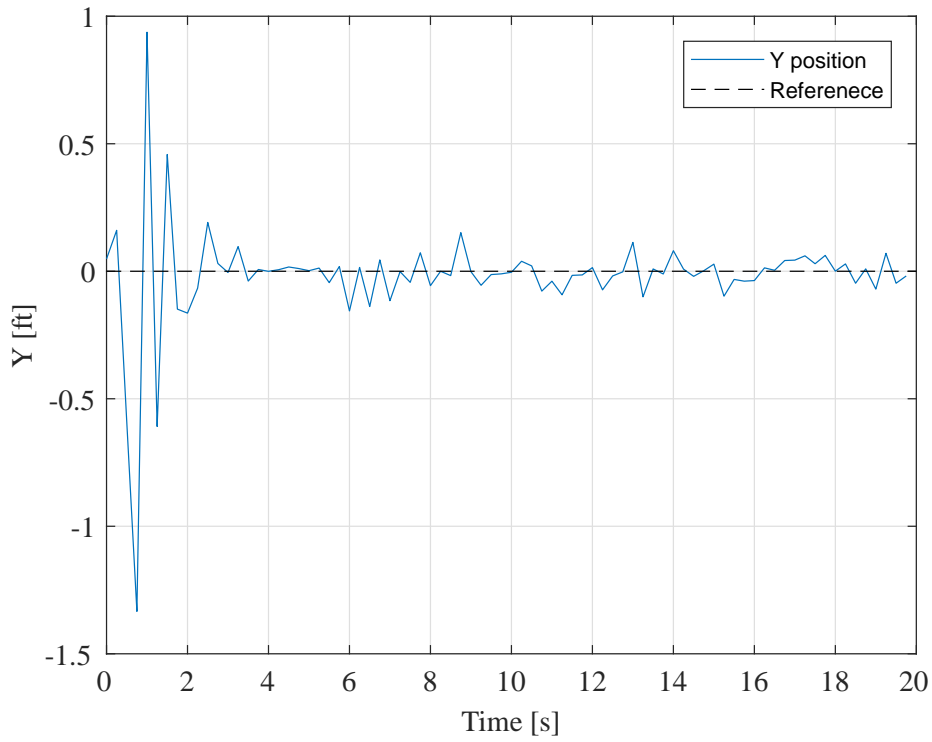


Figure 5.31: Descent: Y position tracking

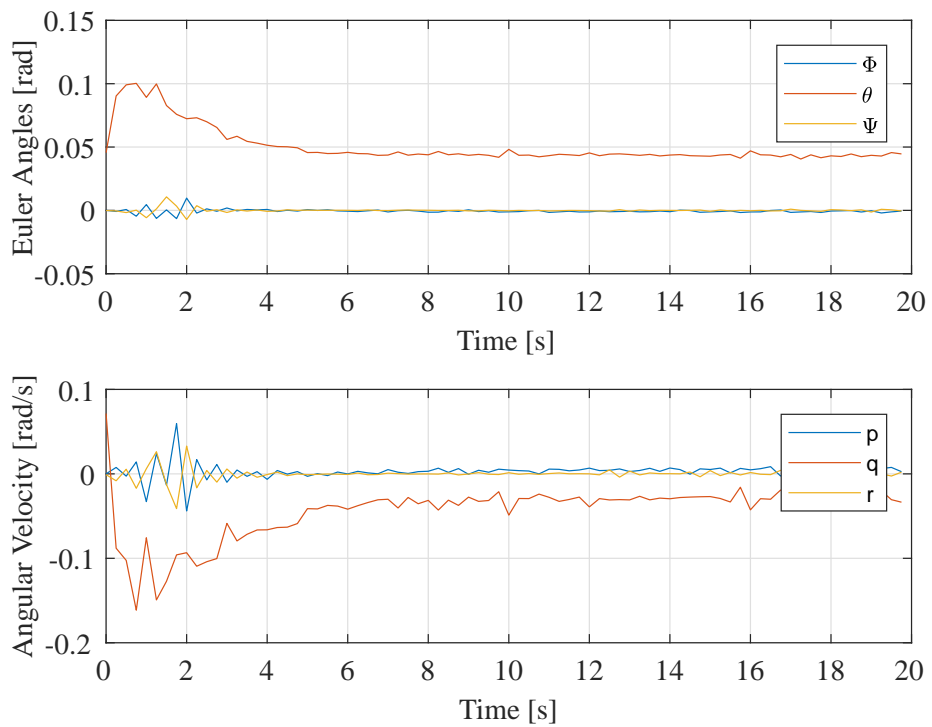


Figure 5.32: Descent: Euler angles and angular velocities

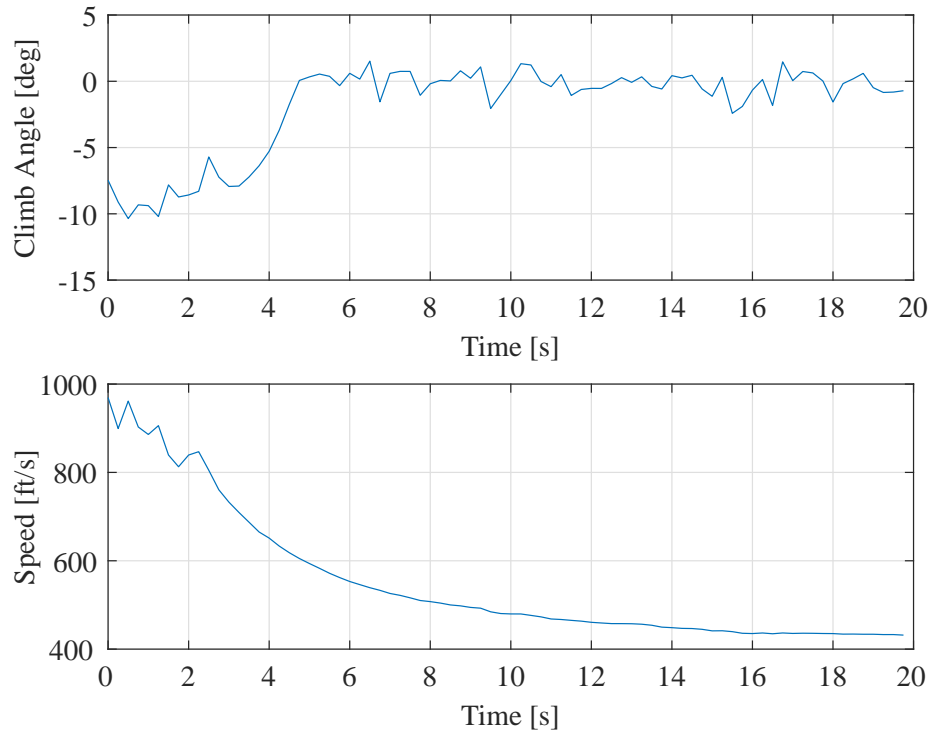


Figure 5.33: Descent: Climb angle and velocity magnitude

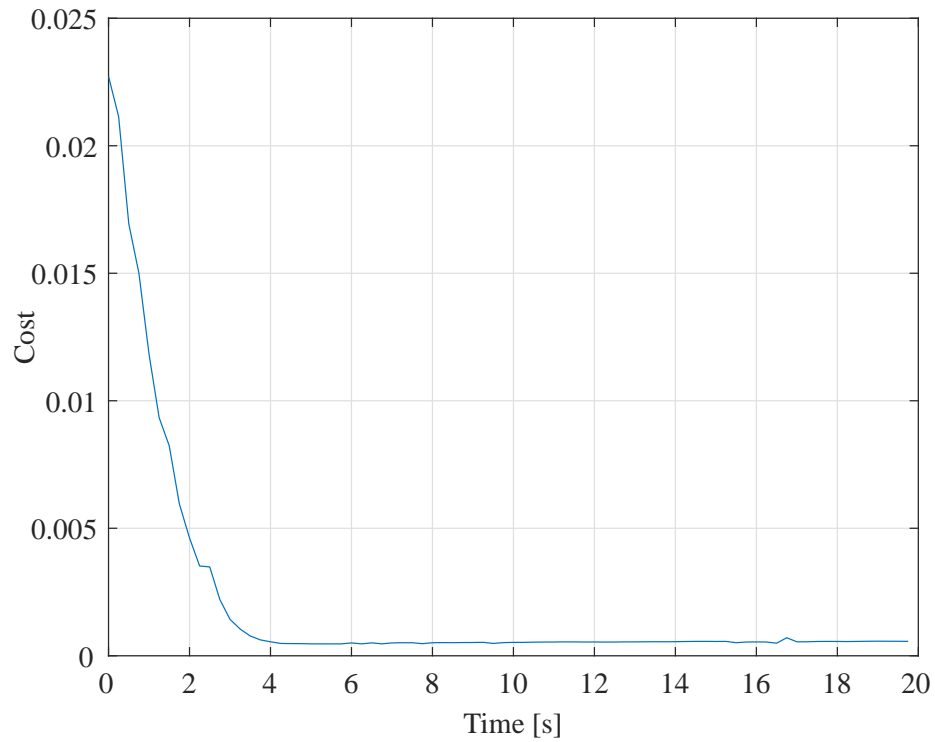


Figure 5.34: Descent: State tracking cost

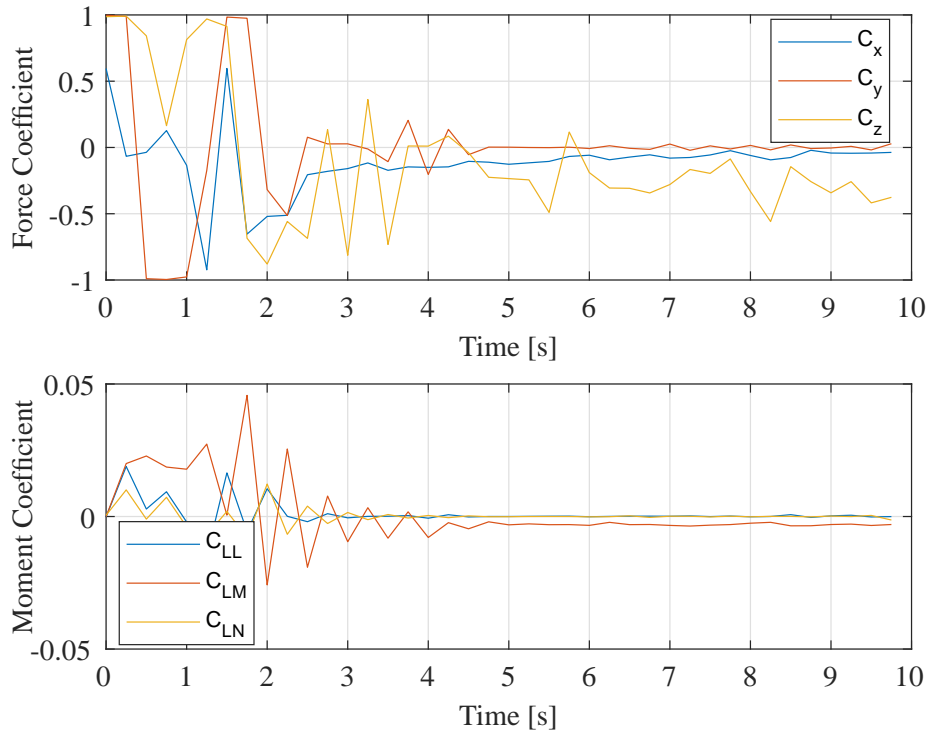


Figure 5.35: Two Position Reference: Commanded force and moment coefficients from the pseudo control optimization

5.4 Tracking References in Two Position Coordinates

The final simulation done was to track a reference in more than one position coordinate. The reference to track was a simultaneous step input change in Y_I and Z_I . The two reference values stepped were 300 ft in the Y_I and 1000 ft in the Z_I . The results are shown in Figures 5.35 - 5.47.

Figure 5.35 shows the coefficients of the system. This motion requires control over both altitude state, and Y position state, this is reflected by large coefficient commands on C_y and C_z . Figure 5.36 shows little involvement of the thrust vectoring system, and a relative to the other climbing motions a small magnitude of thrust. Figure 5.37, 5.38, and 5.39 indicate that no single control surface meets a constraint for this motion. Remarkably there appears to be very little control authority exercised over the system in comparison to other simulations where a climb is required. The reason for this is not known.

Figure 5.40 shows the position and body velocities. It can be seen in Figure 5.41 and 5.42 that the aircraft converges on both states. To get a better idea of what the maneuver looks like the positions were plotted in 3 space in Figure 5.43. Two semi transparent planes represent the

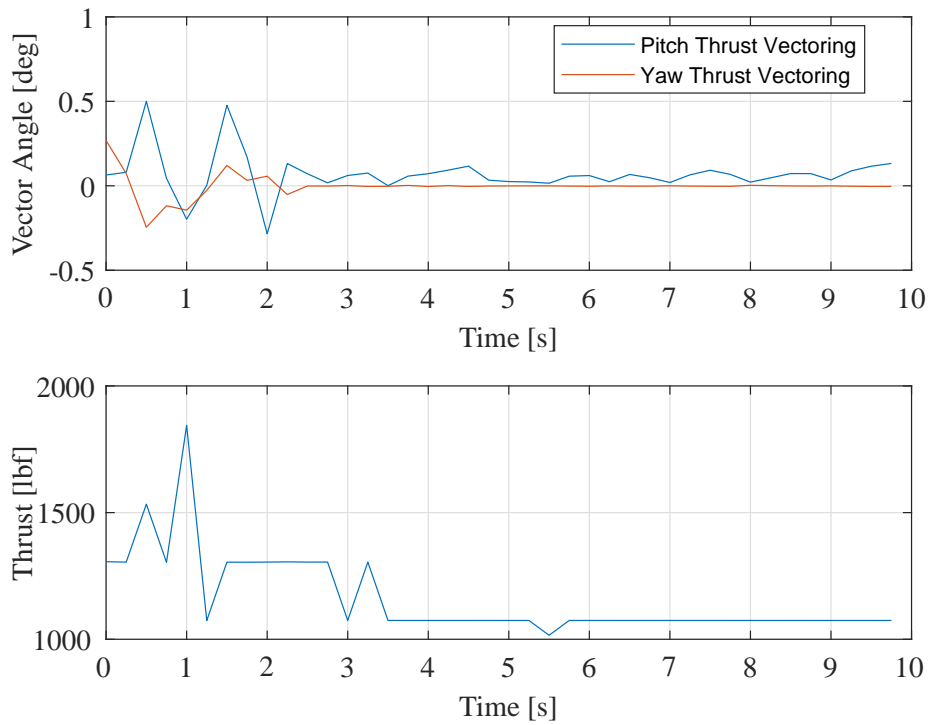


Figure 5.36: Two Position Reference: Thrust parameters

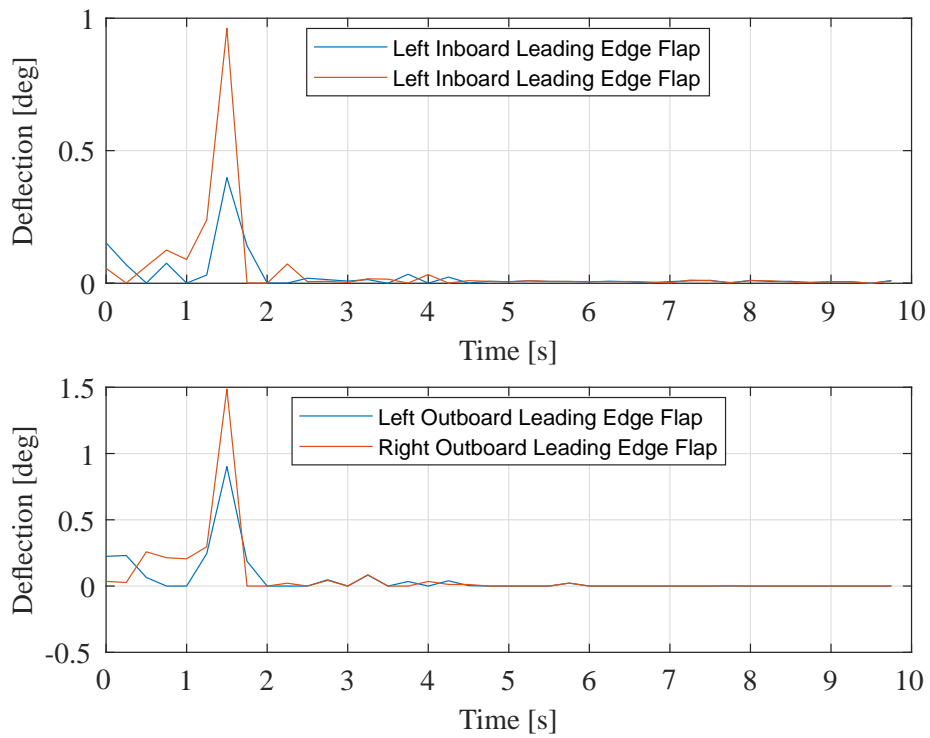


Figure 5.37: Two Position Reference: Inboard and outboard leading edge flap deflections

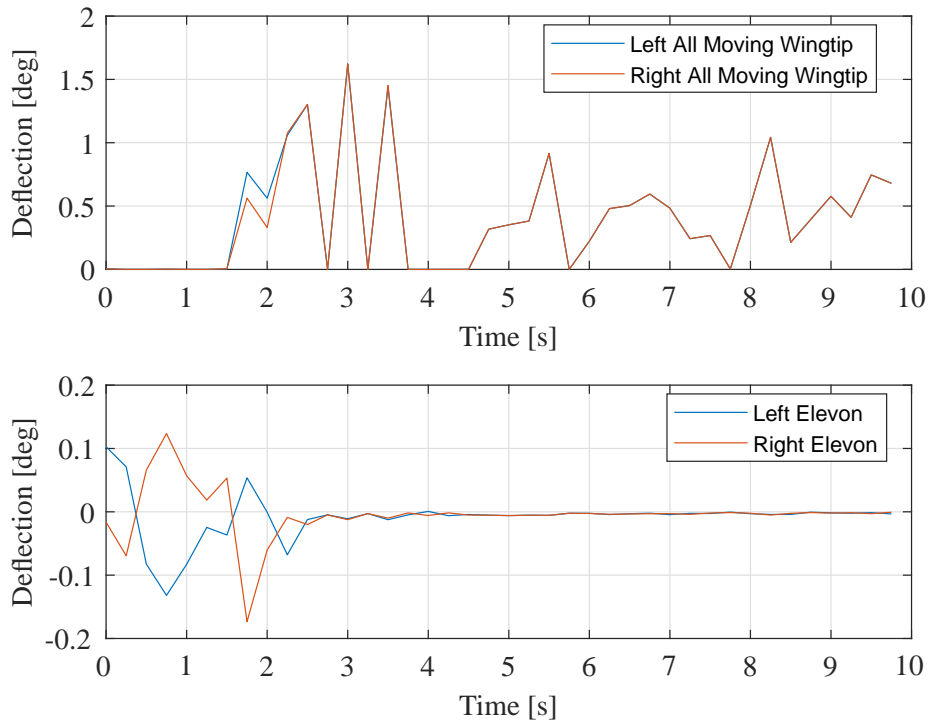


Figure 5.38: Two Position Reference: All moving wingtip and elevon deflections

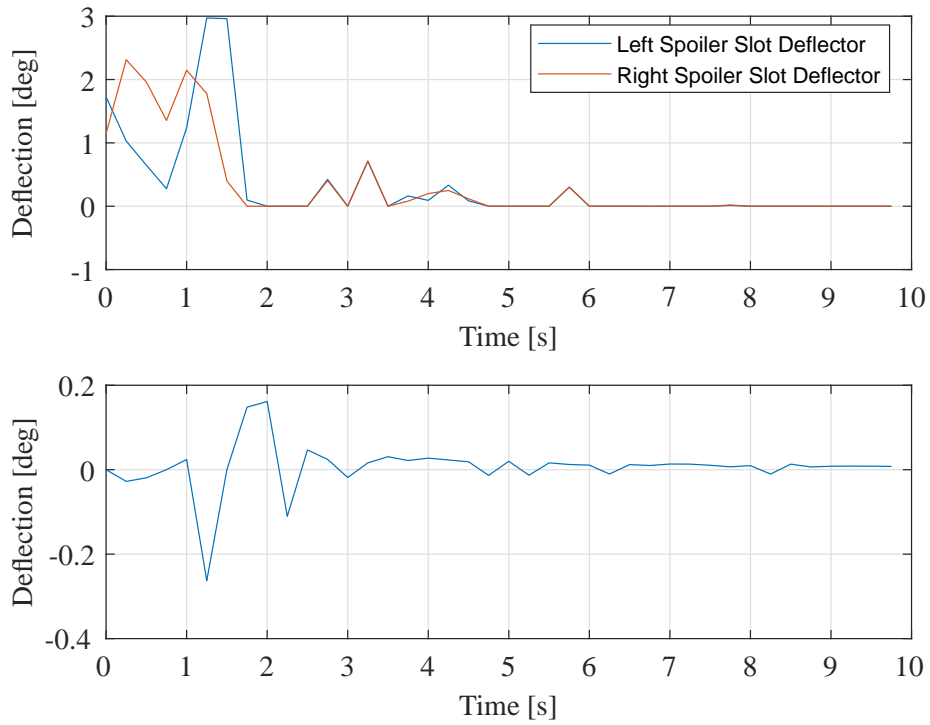


Figure 5.39: Two Position Reference: Spoiler slot and pitch flap deflections

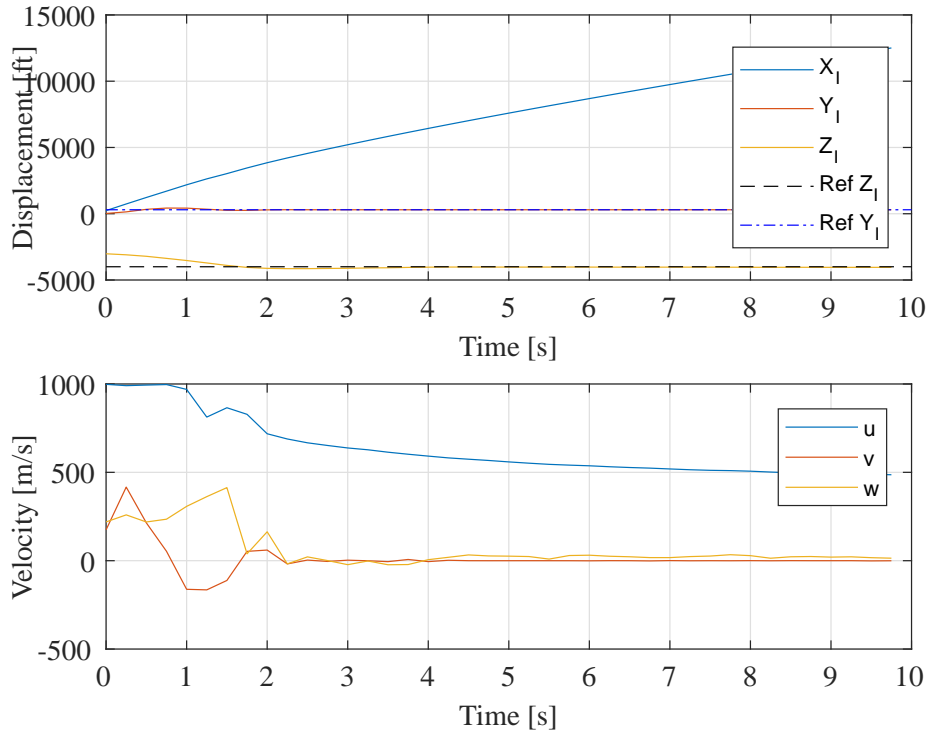


Figure 5.40: Two Position Reference: Positions and velocities

references for tracking. The pink plane, represents the Y_I reference, and the blue plane the Z_I reference. If there were zero error the flight trajectory would live on the intersection of these two planes. Figure 5.43 clearly shows overshoot in the Y_I and Z_I directions. The craft does converge on the Y_I reference but does not converge in the Z_I direction within the given time. It was at first suspected that this was due to allocation pseudo control mismatch. Contrary to that suspicion, Figure 5.44 shows a simulation of the same motion, but assuming perfect control allocation. While there is much better behavior in the Y_I direction, the MPC still cannot find a solution that converges on the Z_I position within the simulation time. It seems that regardless of the ability of the allocation algorithm to meet the requested pseudo control inputs, the craft cannot meet the reference. This can be once again attributed to the aforementioned problem of altitude overshoot recovery due to constraints in angle of attack. Figure 5.45 shows the Euler angles and angular velocities for the motion.

Figure 5.46 shows the climb angle and speed during the motion. Interestingly the climb angle returns to a value close to zero even though the altitude is not quite met. This is likely because the system is trying to balance a Y position reference and an altitude reference. Figure

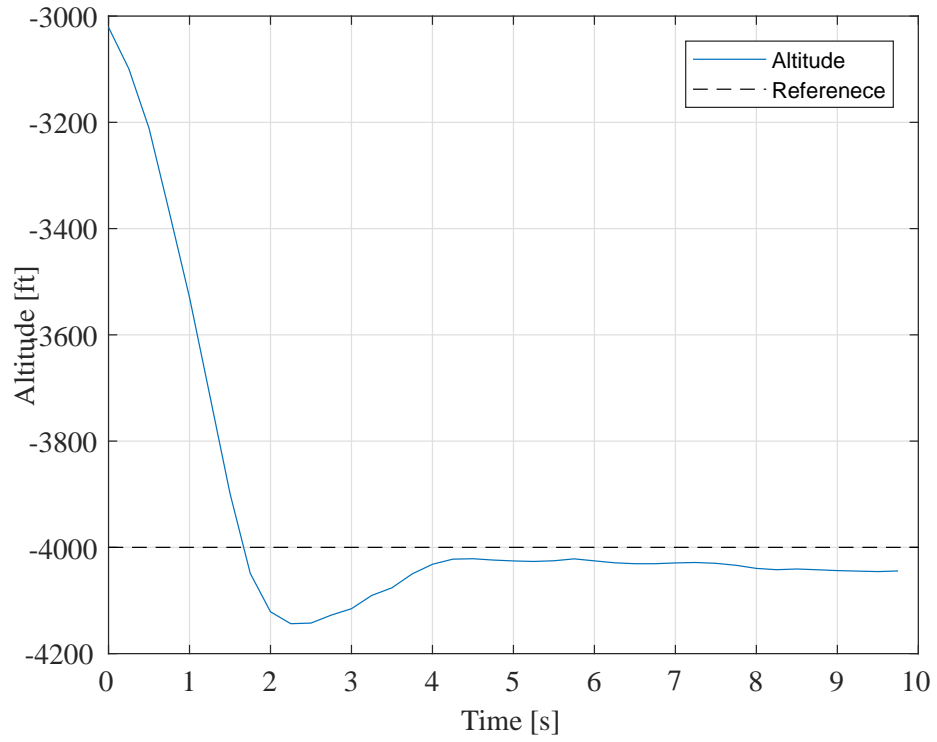


Figure 5.41: Two Position Reference: Altitude tracking

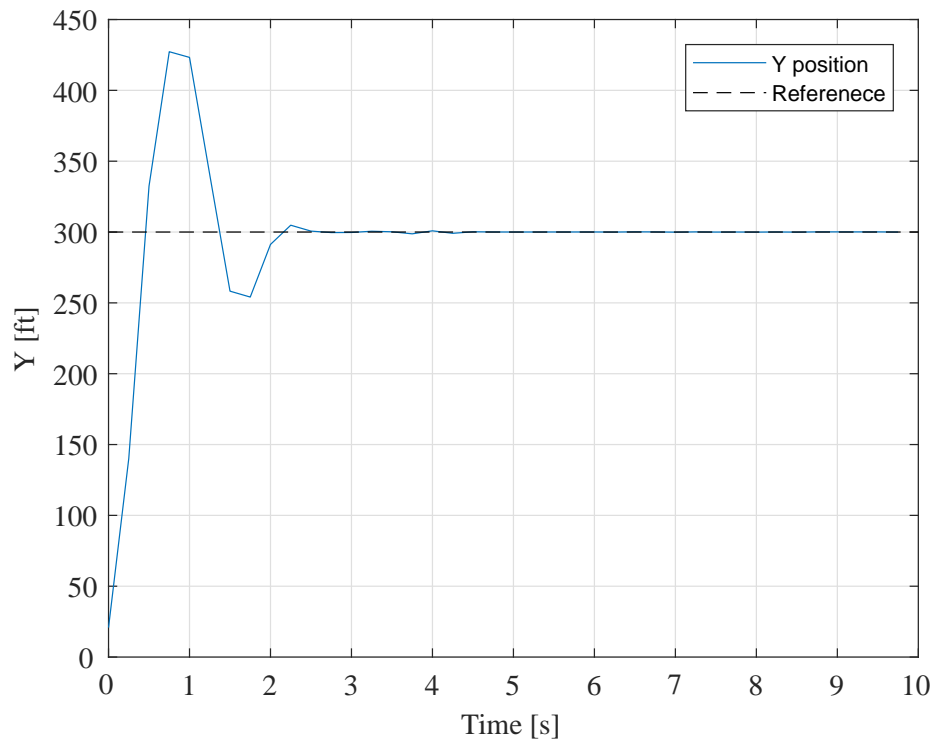


Figure 5.42: Two Position Reference: Y position tracking

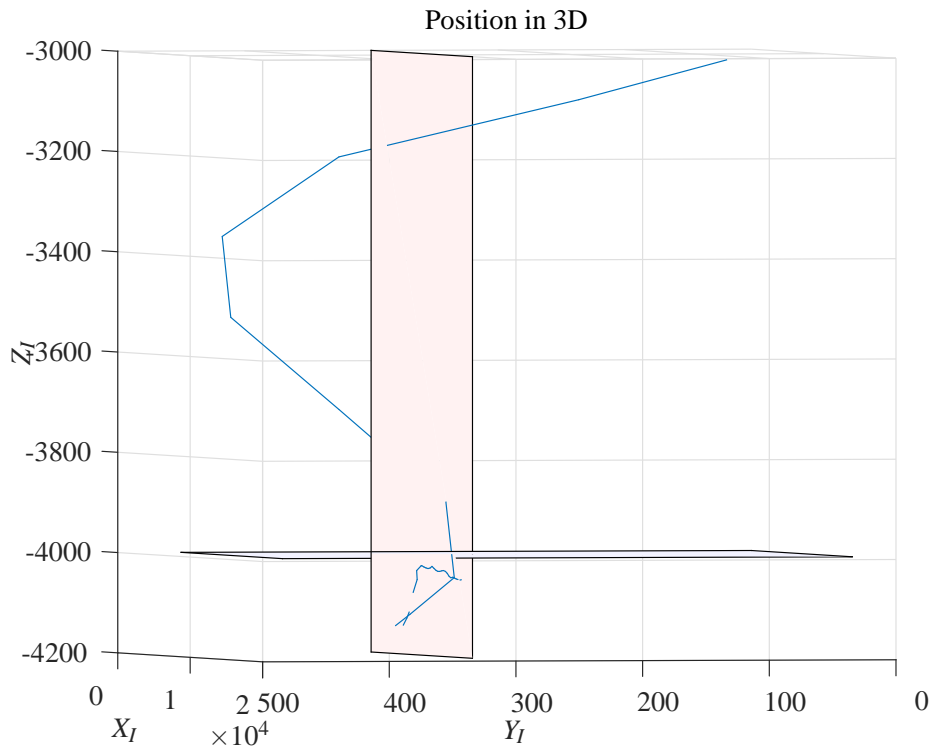


Figure 5.43: Two Position Reference: Positions in 3D

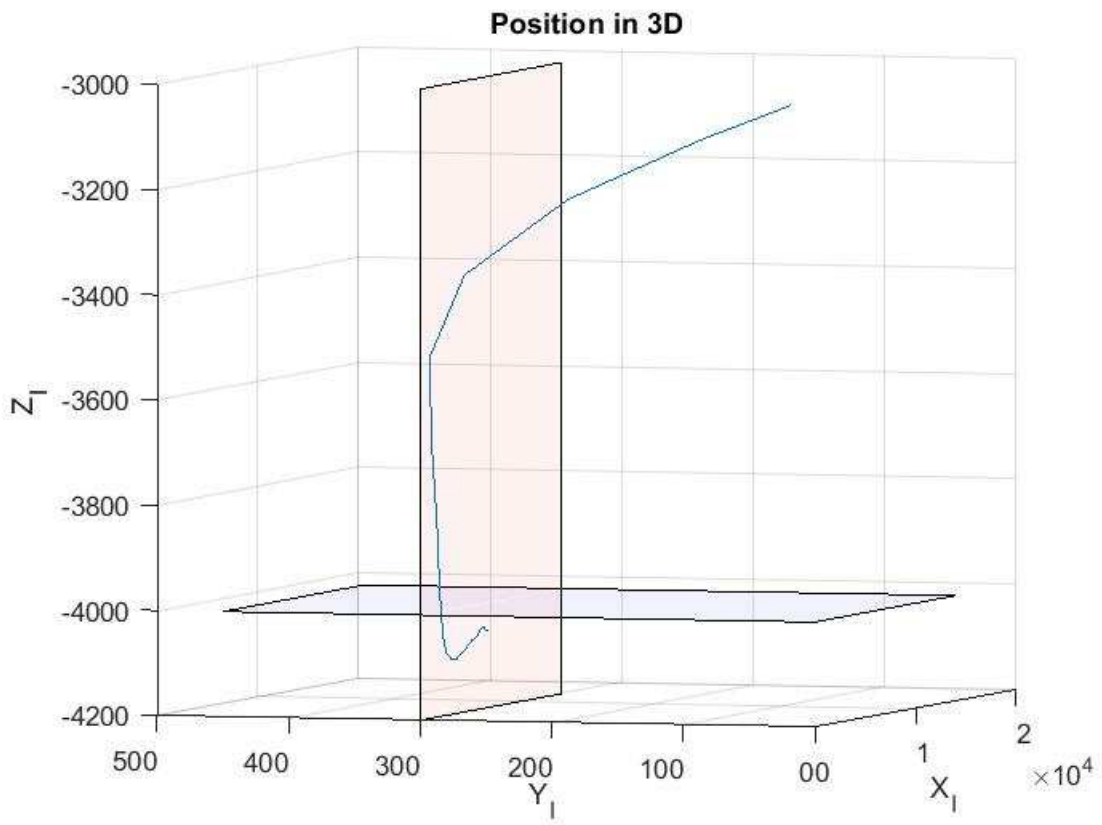


Figure 5.44: Two Position Reference: 3D positions assuming perfect control allocation

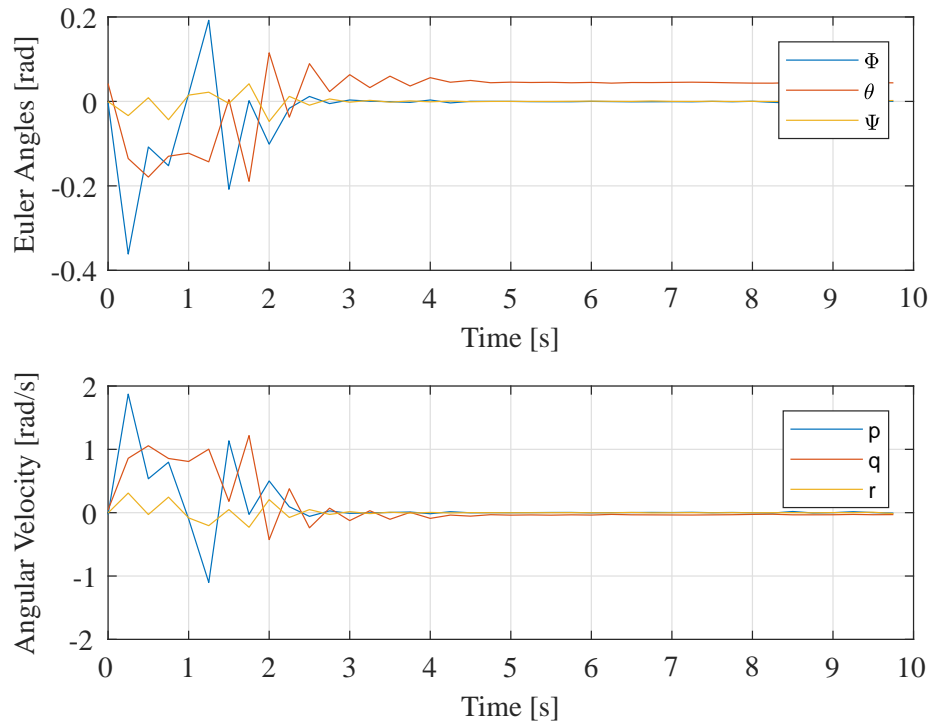


Figure 5.45: Two Position Reference: Euler angles and angular velocities

5.47 shows the normalized cost for the states. The overshoot in the Y position occurs at the peak less than 1 second, and the Z overshoot occurs at the peak in between 1 and 2 seconds. Otherwise the cost decreases as time goes on.

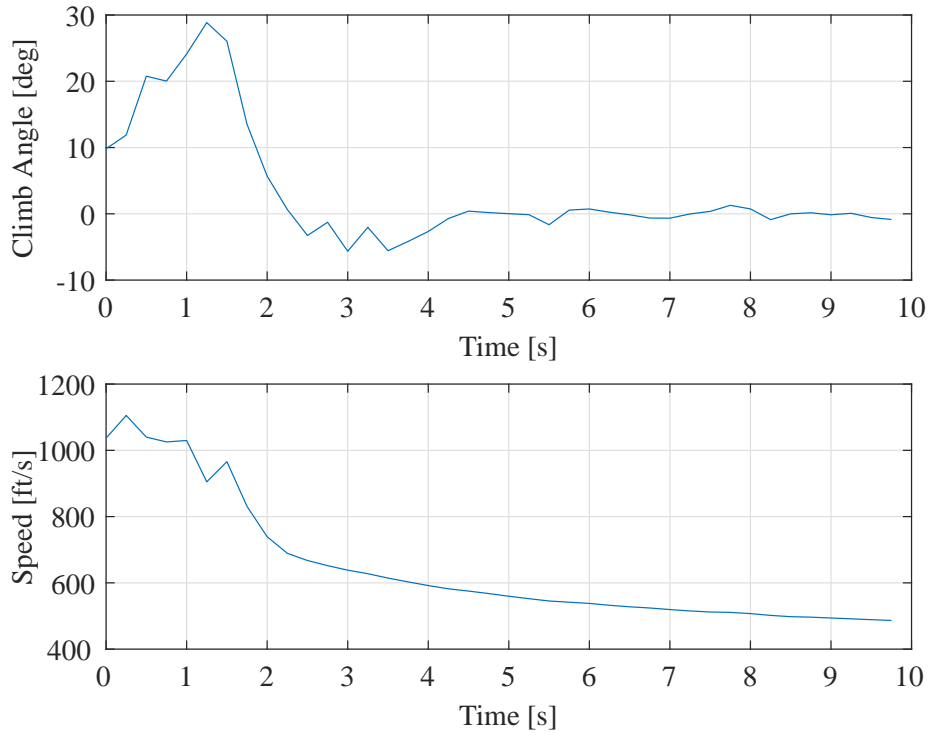


Figure 5.46: Two Position Reference: Climb angle and velocity magnitude

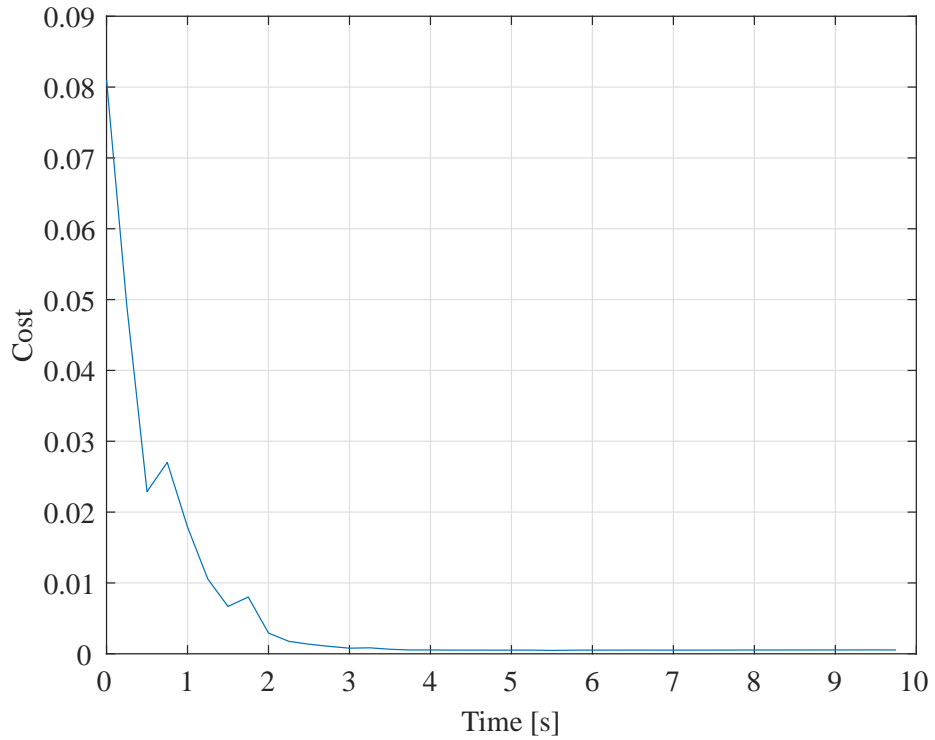


Figure 5.47: Two Position Reference: State tracking cost

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

This chapter will begin by drawing some major conclusions about the MPC design process and MPC performance for the ICE aircraft. The final section of this thesis will discuss what improvements can be made on the proposed MPC.

6.1 Observations Based on Climb Angle and Speed

Each simulation has additional plots to show the climb angle and speed of the aircraft. During simulation these climb angles are limited not by a thrust to weight ratio but by the angle of attack constraint. The thrust to weight ratio of the craft is 1.34, and without an engine model that reflects changes in air density the craft does not lose power with altitude. This results in the craft being able to climb at 90° relative to the inertial frame at any altitude. The limiting factor for this climb will be attitude stability when dynamic pressure becomes negligent. Without dynamic pressure the aerodynamic control surfaces cannot compensate for aircraft attitude, and everything must be done by the engine. In reality an engine with rotating subsystems such as a turbofan couldn't operate past a certain altitude because low air densities wouldn't allow enough oxygen into the engine for combustion.

The lowest speed employed by the control for any motion was 440 ft/s. During the simple tuning case, descent case, and two position reference cases, the speed settles down around this minimum. During the altitude sweep case speed at every step command speed increases during climb, and decreases after altitude is attained and before the new step command until instability occurs. Each step command the apparent speed the system would settle on is slightly higher, this is likely due to the system requiring more velocity to keep dynamic pressure, due to losses in air density as altitude increases. This demonstrates the MPC's ability to properly compensate for

changes to the dynamics of the system due to altitude.

6.2 MPC Performance Observations Based on the Simulations

It is clear that MPC has promise in the field of aircraft control. The position references were tracked, and the craft remained stable while the simulated aircraft operated within the given constraints. The algorithm was able to track step inputs in altitude across the full flight envelope, showing signs that it would converge given more time before the next reference update. This demonstrates the ability of a single nonlinear control methodology to be applied to a problem that would normally take several linear controllers. Unfortunately the constraints on α prevented the craft from being able to track decreases in altitude in a similar manner as increases, meaning if there was overshoot in altitude the craft needed much more time to recover from it. It was demonstrated by the third simulation that the control was able to handle tracking lateral states as well as longitudinal ones. This shows that it is capable of handling the entire 6 DOF control problem. In short, a single nonlinear MPC was developed that was shown to be able to optimally operate the simulated craft anywhere in the flight envelope.

Control solutions were found to take on average 1 minute each. That is for every sample time of 0.25 seconds the amount of real time it took to solve the optimal control problem was 60 seconds. In the final case where suboptimal solutions were accepted and infeasibility occurred the solution time was increased to an average of 2.3 minutes. This insinuates that solution times will have to be at least 240 times faster to be implemented in real time. One way that was attempted to increase solution speed was to linearize the model at each sample time and apply a MPC to the newly linearized model. This was unsuccessful, likely due to poor choice in the length of sample time, in that the system was evolving too much between each sample time for a linear model to properly represent the dynamics.

6.3 Future Work

While designing the nonlinear MPC several problems presented themselves. Perhaps the most difficult of these to deal with is determining adequate constraints to define the system. It was observed that when the controller meets a constraint that the solutions quickly become poor for

flight control. Perhaps one of the most restrictive constraint was on the angle of attack. This can be eased by constraint softening, gathering more aerodynamic data, or by implementing logic to roll the aircraft and perform a pull-up maneuver whenever large altitude descents are asked for. Appropriate bounds for the coefficients of force and moment are also difficult to determine. One solution could be to apply a global solver such as a genetic algorithm to the aerodynamic tables to attempt to find the maximum and minimum attainable values for each coefficient individually. One potentially simpler way around this is to make the MPC decide on forces and moment control variables rather than coefficients of force and moment, which are more easily bounded because of their more direct physical implications.

One way to eliminate infeasibility due to violation of state constraints is to employ barrier functions inside the cost rather than hard constraints. Barrier functions are those that cause the cost to incur heavy penalties as some boundary is approached. This is a reasonable way of handling the state constraints because in reality an aircraft is not incapable of exceeding any hard state limits that exist within its flight envelope. This solution would prevent infeasibility and cause the optimizer to stay away from some boundary as it attempts to minimize the cost function. This does of course allow the craft to violate some constraints if need be, which could be detrimental to pilot, structure, or aerodynamic force generation.

Control input rate responses were not considered during simulation. It is possible to implement them as linear constraints within the pseudo control, and control allocation algorithms. The control input rate likely to have the largest impact on controller performance is associated with the engine. As it stands engine thrust changes are considered instantaneous which allows the craft to compensate against undesired motion with instantaneous bursts in engine output in the vectoring directions. This causes an instantaneous moment which enables to engine to be used for angular rate control, which is unrealistic. Including rate dynamics in the aerodynamic control surface would slow down reference convergence, but is not likely to have a large impact on the ability of the controller to find optimal solutions.

Tuning an MPC is a difficult process. A more in depth study should be conducted on MPC tuning techniques for a general nonlinear system. Having a better tuning strategy would likely result in better performance over the entire flight envelope, and may reduce or eliminate the overshoot that was occurring at high altitudes. Related to tuning through choosing an appropriate

sample time is implementing a strategy to control nonlinear systems whose states evolve on different time scales. It was observed through the tuning process, and elaborated on in [15], that the angular velocities evolve much quicker than the positions. Being able to consider both time scales would improve attitude stabilization which would in turn improve the ability of the craft to track.

Lastly different methods of solving both the optimization problems should be considered. SQP is extremely computationally intensive, taking significant computation time. Some of the aforementioned works by other researchers have reformulated the flight control problem to make it more suitable for SQP algorithms, saving computation time. Validating these methods for a tailless fighter aircraft should be explored in the interest of real-time nonlinear MPC application. It was observed during tuning that on average solving the control allocation problem took three times the amount of time it took to solve the pseudo control problem. There are many methods that could be employed as per the Literature Review to shorten the time for control allocation.

REFERENCES

- [1] Kenneth Dorsett, Scott Fears, and Heather Houlden. Innovative control effectors (ice) phase ii. Technical report, Air Force Research Laboratory Wright-Patterson Air Force Base, 1999.
- [2] William Dunbar, Mark Milam, Ryan Franz, and Richard Murray. Model predictive control of a thrust-vectorred flight control experiment. In *IFAC Triennial World Congress*, 2002.
- [3] T. Keviczky and G. Balas. Receding horizon control of an f-16 aircraft: A comparative study. *Control Engineering Practice*, 14(9):1023–1033, Sept., 2006.
- [4] Sébastien Gros, Rien Quirynen, and Moritz Diehl. Aircraft control based on fast non-linear mpc & multiple-shooting. In *51st IEEE Conference on Decision and Control December 10-13*, 2012.
- [5] M Kale and A Chipperfield. Stabilized mpc formulations for robust reconfigurable flight control. *Control Engineering Practice*, 2005.
- [6] Fabio Almeida and Dirk Leissing. Fault-tolerant flight control system using model predictive control. In *Brazilian Symposium on Aerospace Eng. & Applications*, 2009.
- [7] Jan Maciejowski and Colin Jones. Mpc fault-tolerant flight control case study: Flight 1862. In *IFAC Fault Detection, Supervision and Safety of Technical Processes*, 2003.
- [8] John Bowlus, Dieter Multhopp, and Siva Banda. Challenges and opportunities in tailless aircraft stability and control. Technical report, American Institute of Astronautics and Aeronautics, 1997.
- [9] Christopher Miller. Nonlinear dynamic inversion baseline control law: Architecture and performance predictions. Technical report, NASA, 2011.

- [10] Uzair Ansari and Bajodah Abdulrahman. Robust launch vehicle's generalized dynamic inversion attitude control. *Aircraft Engineering and Aerospace Technology*, 2017.
- [11] C Hanson, J Schaefer, J Burken, Johnson, and N Nguyen. Handling qualities evaluations of low complexity model reference adaptive controllers for reduced pitch and roll damping scenarios. In *AIAA Guidance Navigation and Control Conference*, 2011.
- [12] Anhtuan Ngo, William Reigelsperger, Siva Banda, and John Bessolo. Multivariable control law design for a tailless airplane. In *AIAA, Guidance, Navigation and Control Conference*, 1996.
- [13] DW Gu, P.H. Petkov, and M.M. Konstantinov. *Robust Control Design with MATLAB*, chapter Robust Design Specifications, pages 22–30. Springer, 2013.
- [14] DW Gu, P.H. Petkov, and M.M. Konstantinov. *Robust Control Design with MATLAB*, chapter μ -Analysis and Synthesis, pages 65–71. Springer, 2013.
- [15] Matamoros I and de Visser C.C. Incremental nonlinear control allocation for a tailless aircraft with innovative control effectors. In *2018 AIAA Guidance, Navigation, and Control Conference*, 2018.
- [16] Zhang Bo, Zhu Xiaoping, Zhou Zhou, and XU Mingxing. Design of nonlinear flight control law for a tailless unmanned aerial vehicle.
- [17] Jingqing Han. From pid to active disturbance rejection control. In *IEEE Transactions on Industrial Electronics, Vol 56. No 3*, 2009.
- [18] James Buffington. Modular control law design for the innovative control effectors tailless fighter aircraft configuration 101-3. Technical report, Air Force Research Laboratory, 1999.
- [19] Barton Bacon and Aaron Ostroff. Reconfigurable flight control using nonlinear dynamic inversion with a special accelerometer implementation. In *AIAA Guidance, Navigation, and Control Conference and Exhibit*, 2000.
- [20] A Calise, S Lee, and M Sharma. Direct adaptive reconfigurable control of a tailless fighter aircraft. In *AIAA-98-4108*, 1998.

- [21] Kumpati Narendra and Lena Valavani. Direct and indirect adaptive control. Technical report, Department of Engineering and Applied Science, Yale University, 1978.
- [22] Rowena Eberhardt and David Ward. Indirect adaptive flight control system interactions. *International Journal of Robust and Nonlinear Control*, 1999.
- [23] Daniel Simon. Model predictive control in flight control design stability and reference tracking. Master's thesis, Linköping University, 2014.
- [24] Martin Hagström. *The ADMIRE Benchmark Aircraft Model*, pages 35–54. Springer Berlin Heidelberg, Berlin, Heidelberg, 2007.
- [25] Nathan Slegers, Jason Kyle, and Mark Costello. Nonlinear model predictive control technique for unmanned air vehicles. *Journal of Guidance, Control, and Dynamics Vol. 29, No. 5, September-October*, 2006.
- [26] Brian Stevens and Frank Lewis. *Aircraft Control and Simulation*, chapter Modeling the Aircraft, pages 59–107. Wiley, 2003.
- [27] M Niestroy, K Dorsett, and K Markstein. A tailless fighter aircraft model for control-related research and development. In *AIAA SciTech Forum, AIAA Modeling and Simulation Technologies Conference*, 2017.
- [28] Bernard Etkin and Lloyd Reid. *Dynamics of Flight Stability and Control*, chapter General Equations of Unsteady Motion, pages 115–118. John Wiley & Sons, 1996.
- [29] Jan Marian Maciejowski. *Predictive Control With Constraints*. Prentice Hall, 2002.
- [30] Eric Weisstein. Convex function. From MathWorld—A Wolfram Web Resource. <http://mathworld.wolfram.com/ConvexFunction.html>.
- [31] D Mayne, J Rawlings, C Rao, and P Scokaert. Constrained model predictive control: Stability and optimality. *Automatica*, 36(6):789-814, June, pages 23–44, 2000.
- [32] K Muske and J Rawlings. Model predictive control with linear models. *AIChE Journal* 39(2) Feb., 1993.

- [33] M Cannon. Efficient nonlinear model predictive control algorithms. *Reviews in Control*, 28(2) Jan., 2004.
- [34] Paul Boggs and Jon Tolle. Sequential quadratic programming. *Acta Numerica*, pp. 1-1000, 1996.
- [35] Dimitri P. Bertsekas. *Constrained Optimization and Lagrange Multiplier Methods*. Athena Scientific, 1996.
- [36] F Topputo and C Zhang. Survey of direct transcription for low-thrust space trajectory optimization with applications. *Abstract and Applied Analysis vol. 2014*, 2014.