5-1-2020

SINGLE AXIS PITCH CONTROL AUTOPILOT DESIGN FOR A GENERAL AVIATION AIRCRAFT

Lorenzo Novoa

SINGLE AXIS PITCH CONTROL AUTOPILOT DESIGN FOR A GENERAL AVIATION AIRCRAFT

by

LORENZO NOVOA

Presented to the Faculty of the Honors College of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

HONORS BACHELOR OF SCIENCE IN AEROSPACE ENGINEERING

THE UNIVERSITY OF TEXAS AT ARLINGTON

May 2020

ACKNOWLEDGMENTS

The project to develop a pitch control autopilot for an aircraft was proposed by Dr. Dudley Smith from the Mechanical and Aerospace Engineering department at the University of Texas at Arlington. I would like to express my gratitude to Dr. Smith for proposing this idea. Similarly, I appreciate my teammate Josiah Everhart for his collaboration. This project would have been incomplete without his support and collaboration.

Likewise, I wish to express my gratitude to Dr. Animesh Chakravarthy for offering his feedback and guidance toward the project. His experience and expertise cover a wide range of topics within the guidance, navigation, and control field; and aided me in exploring new solution paths.

May 08,2020

ABSTRACT

SINGLE AXIS PITCH CONTROL AUTOPILOT
DESIGN FOR A GENERAL AVIATION
AIRCRAFT

Lorenzo Novoa, B.S. Aerospace Engineering

The University of Texas at Arlington, 2020

Faculty Mentor: Dudley Smith

Autopilot design is a technical and essential skill prevalent in the guidance, navigation, and control field of the aerospace industry. A Simulink example flight simulator based on the D-200 Sky Hogg Design Proposal was utilized as the simulation platform to attempt to design a single axis pitch controlling autopilot. The task was approached by reviewing the simulation framework capabilities, designing a control for the pitch angle, and simulating the vehicle with the adjusted controller. The design of the controller consisted of developing transfer functions for the elevator actuator and aircraft longitudinal dynamics, integrating a two loop autopilot into the simulator, and selecting the gains for the controller using the Root Locus plots of the inner and outer loops. The results of the controller design were an unstable closed loop system for any combination of values for the two loop gains. The simulation with the controller design resulted in

divergence of the aircraft from the commanded altitude. The uncontrollability of the system was likely due to an error in the vehicle modeling or the implementation of the two-loop autopilot into the simulation. Many lessons were learned about controller design and implementation into a flight simulator in the process of trouble shooting and modigying the simulator and controller design. Many of these lessons will directly translate to relevant skills necessary to excel in the aerospace industry. All in all, the project was a useful exercise to gain experience in topics outside of the scope of general coursework at The University of Texas at Arlington (UTA).

TABLE OF CONTENTS

LIST OF ILLUSTRATIONS

CHAPTER 1

INTRODUCTION

1.1 Autopilot Overview

An autopilot is a control scheme that can be utilized to define the behavior of a system from the reference of a current state. As the name suggests, an autopilot can achieve a desired behavior for a system without the intervention of a human control input. One simple example of an autopilot design in the cruise control feature in a modern car. The desired speed is the reference input. The controller within the cruise control feature quantifies the error between the desired speed and the current speed calculated by the car sensors to determine the acceleration required to achieve the desired speed. If the feature is designed poorly then it would be possible to run into issues such as: never reaching the desired speed, taking too long to accelerate to the desired speed, accelerating at an uncomfortable rate for the passengers, and oscillating around the desired speed while never matching it. Some of these problems and the methods required to solve them will be further discussed in the methodology section for this report.

Autopilot design is a specific topic that fits into a larger field within the aerospace industry known as guidance, navigation, and control (GNC). Guidance is the desired path a vehicle should take to meet the mission requirements, such as the shortest trip time or the most fuel-efficient path. Navigation is the current location and orientation of the vehicle as defined to an inertial reference frame. Navigation usually depends on an inertial measurement unit (IMU) to determine the relative accelerations present on the vehicle and

a global positioning system (GPS) to determine the coordinates of the vehicle in an established coordinate system. Control relates to the schematics utilized to interpret the navigation sensor data and use it to maintain the desired guidance trajectory with the vehicle controls. The common forms of controls available on an aircraft include control surfaces, such as the elevators and rudder, and throttle setting. An autopilot design fits into the control category of GNC but is dependent on guidance and navigation to properly function in a vehicle. Being such a broad topic, the use of an autopilot as the focus of this project required some constraints to be placed on the scope of the autopilot design. The next section will discuss the basic concepts necessary to understand how the project was constrained and some of the types of vehicle information required for the project.

1.2 Overview of Essential Concepts

A conventional fixed-wing aircraft has few common components. They usually have a fuselage containing the cockpit, propulsion device, and most of the aircraft subsystems running along the centerline of the vehicle. Additionally, a set of wings symmetrically placed on the fuselage and near the center of gravity. An empennage, often referred to as the tail of an aircraft, is comprised of the horizontal and vertical stabilizers located at the back of an aircraft, offset from the center of gravity. This is the conventional configuration, but there are numerous variations, as well as many different configurations with their own unique features and benefits.

The aircraft body can be further defined using a coordinate system convention, such as the one illustrated in Figure 1.1 (Etkins, 1958). This Cartesian coordinate system is fixed to the aircraft airframe, and therefore does not rotate as the aircraft rotates. Each of the coordinate directions corresponds to a velocity and force component of the vehicle, as

well as an axis of rotation related to the moments present on the aircraft. The y-axis of the body attached coordinate system is responsible for the pitching moment and pitching velocity of the vehicle. The scope of the project was limited by assigning the autopilot design to only operate as a controller for the pitch angle of the vehicle. As a result, the design would only have to be implemented in the control of a single axis of rotation. Furthermore, the only control surface utilized in the controller would be the elevator located on the horizontal stabilizer of the empennage assembly.

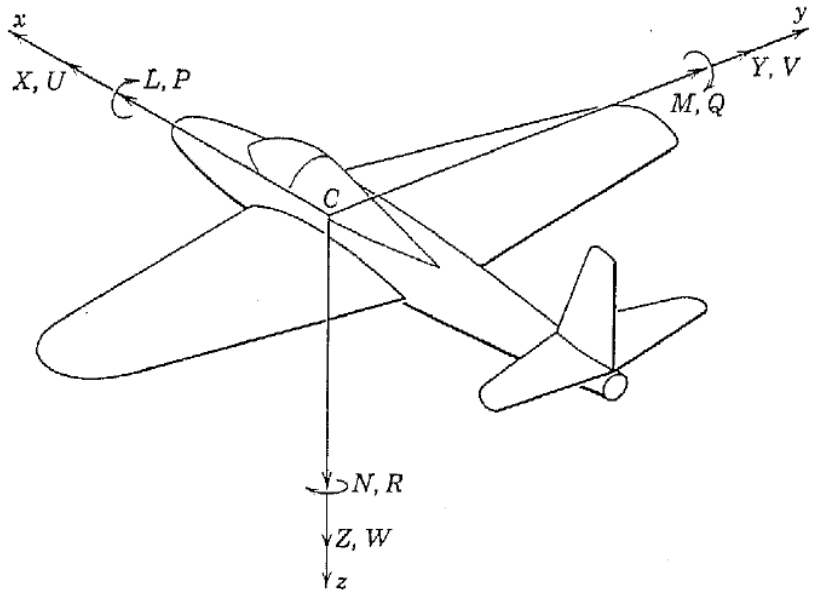

FIG. 1.3. Notation.

L = rolling moment P = rolling velocity
M = pitching moment Q = pitching velocity
N = yawing moment R = yawing velocity
$[X, Y, Z]$ = components of resultant aerodynamic force
$[U, V, W]$ = component of velocity of C

Figure 1.1: Aircraft Coordinate System and Notation

During flight, all the forces present on the vehicle can be separated into four categories: lift, drag, weight, and thrust. Figure 1.2 depicts the forces as well as the relevant angles to define the vehicle attitude. The pitch angle is taken to be the angle between the

body axis and the reference horizontal axis. The angle of attack is taken to be the angle between the body axis and the velocity vector. The flight path angle is taken to be the angle between the velocity vector and the reference horizontal. As a result of these definitions, the pitch angle is the sum of the angle of attack and flight path angle. Under normal operating conditions for a conventional fixed-wing aircraft, the weight and thrust of the vehicle vary in magnitude with altitude but are independent of rotational orientation. The lift and drag, however, are dependent on both the flight conditions, such as atmospheric data, and vehicle rotational orientation. Due to the wide range of aircraft types, sizes, and speed regimes, it is common to nondimensionalize various parameters for an aircraft to compare their performance to other vehicles. The forces and moments can be nondimensionalized by the dynamic pressure and reference dimensions. The reference dimensions utilized for a fixed wing aircraft include the planform area, S, and the mean geometric chord of the wing, $\bar{c}$. Equation 1 details the general from utilized to nondimensionalize a force into a force coefficient, where x represents the force being normalized (Anderson, 2017). The three forces necessary for this project are lift, drag and weight. Equation 2 details the form utilized to nondimensionalize the moment present on the aircraft into the moment coefficient (Anderson, 2017).

$$C_X = \frac{X}{\frac{1}{2}\rho V^2 S} \tag{1}$$

$$C_M = \frac{M}{\frac{1}{2}\rho V^2 S \bar{c}} \tag{2}$$

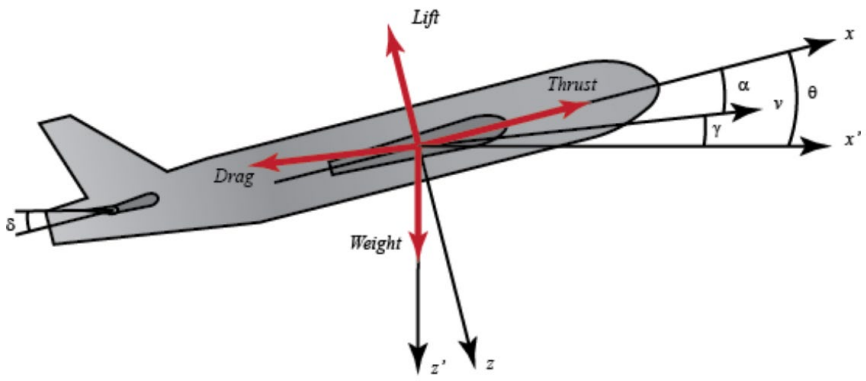

Figure 1.2: Force and Angle Diagram

The motion of a dynamic system can be defined by the governing equations of motion. These equations usually are linked to a specific degree of freedom of a system. As a result, an aircraft has six total equations of motion. Due to the complexity of the force interactions and dynamic phenomena present on an aircraft, the equations of motion are complex and not linear in behavior. These equations are heavily coupled, meaning that each state has a direct effect on the dynamics of the other states. Through various simplifying assumptions, some of the equations can be decoupled. Additionally, the system can be linearized through a method of applying small perturbations to the system, since the behavior is approximately linear for such a small range of values. Once the equations are decoupled and linearized, it is possible to reduce the system to a two axis or single axis system. Lastly, the linearized equations can be converted into functions of the flight conditions, reference dimensions, and nondimensional coefficients.

The objective of this honors senior capstone project is to develop a single axis pitch control autopilot for a fixed-wing aircraft. This task consists of designing a pitch controller, integrating the controller into a simulation of the aircraft, and simulating the aircraft motion for a specified time interval of interest. This task was selected to give the student more

exposure to topics relevant to GNC, which will aid him in his future career endeavors. Specifically, the student will get exposure designing an autopilot controller, integrating a model into a simulation, and running a simulation for varying flight conditions.

CHAPTER 2

METHODOLOGY

2.1 Simulation Framework

The simulation utilized for this project was the D-200 Sky Hogg Design Proposal flight simulator developed as an example for a Simulink aircraft model ("Lightweight Airplane Design"). The simulation consisted of an environment model, a vehicle system model, and a pilot model. The environment model contains an atmosphere model, gravity model, and wind model. The vehicle system model contains all the subsystems necessary to simulate the vehicle response to an input command or environment interaction. The pilot model consists of a commanded altitude.

The environment subsystem utilizes many of the built in Simulink blocks. The atmosphere model uses the 1976 Committee on Extension to the Standard Atmosphere (COESA) block to determine the variation in pressure, density, temperature, and speed of sound with altitude. This model is valid for altitudes up to around 85,000 meters. The gravity model is a Simulink block based on the 1984 World Geodetic System (WGS 84) gravity model to account or the oblate shape of the Earth. The wind model consists of a wind shear Simulink block, a wind turbulence block, and a wind gust block that were summed to determine the overall wind velocity in the three coordinate directions.

The vehicle system model consisted of a subsystem for the vehicle, a subsystem for the flight sensors, and a subsystem for the avionics. The vehicle system model depended on the output file from the Digital DATCOM software developed by the military to use

empirical and qualitative data to determine the approximate values of the force and moment coefficients present on a vehicle during a sweep of Mach values, altitudes, and angles of attack. Digital DATCOM primarily requires geometric data about the vehicle and is a quick first estimate at the performance of a vehicle before leaving the conceptual design phase and entering preliminary design.

The subsystem for the vehicle consisted of a nonlinear second order model for the elevator actuator which incorporated saturation for both a maximum and minimum deflection of 20 degrees. The model also incorporated a propulsion block defined by a thrust curved inputted into the simulation. An aerodynamic subsystem consisted of an Aerodynamic Forces and Moments block to determine the forces and moments present using the aerodynamic coefficients, and a Digital DATCOM Forces and Moments block to define the forces and moments along the body axis due to the stability derivatives output by Digital DATCOM. Lastly, a three degree of motion (3DOF) block was utilized to transform the forces and moments into positional data (such as pitch angle, body accelerations, and velocities) using the integrated equations of motion in the block.

The flight sensor subsystem consisted of a Simulink block to convert the pressure into a sensed altitude, and a block to convert the body velocity into a calibrated airspeed. The velocity conversion functions on a compressible dry air with constant specific heat ratio assumption.

The avionics subsystem consisted of a three-axis IMU model and an autopilot block. The IMU model used the plant output to generate measured accelerations and angular velocities for an accelerometer and gyroscope with noise. The autopilot consisted of an altitude Controller to turn the difference between the command altitude and the

sensed altitude into a commanded pitch angle, and a controller to convert the pitch angle command into a required elevator deflection. The focus of this project consisted of designing the pitch angle controller used to determine the required elevator deflection.

The pilot model for the simulation consisted of a throttle setting input, an initial altitude command, and a step command to represent the desired final altitude. By default, the throttle setting was set to 50% maximum power and the initial altitude was set at 2000 meters. This simulation framework allowed the focus of the project to be refined to just the pitch autopilot design as the simulation framework could stand alone and properly simulate the environment and flight vehicle system.

2.2 Controller Design

The design of the pitch controller was based on a two-loop design consisting of a feedback loop for the pitch rate as well as one for the pitch angle. Figure 2.1 depicts the general architecture for a two-loop pitch controller (Roskam, 2011). The controller design process consisted of constructing transfer functions for the aircraft and actuator model, implementing the two-loop architecture into the simulation framework, and selecting gain values to achieve the desired performance from the controller. Once the controller design was complete, the simulation was run to observe the performance of the controller during various flight conditions and vehicle attitudes.

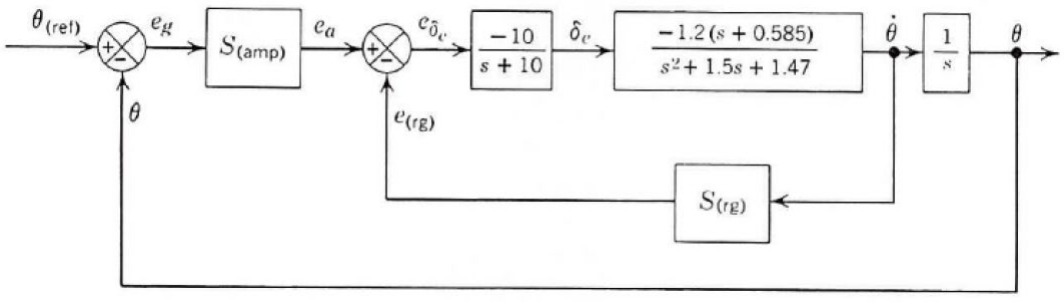

Figure 2.1: Two-loop Pitch Autopilot Architecture

2.2.1 Transfer Function Construction

The two necessary transfer functions for the two-loop controller design chosen are the aircraft elevator deflection to pitch rate transfer function and the actuator command deflection to actual deflection transfer function. The aircraft pitch rate transfer function was initially developed as a state space system with three states: angle of attack, pitch rate, and pitch angle. The input for the state space was the elevator deflection and the output was the pitch rate. Equations 3 through 5 are the longitudinal equations of motion utilized to define the state space system, these were developed in the Control Tutorial for MATLAB & SIMULINK website developed by the University of Michigan. These equations of motion assume steady cruise at constant altitude and velocity. Additionally, the equations of motion have an unrealistic assumption of the pitch angle not affecting the vehicle speed. The nominal flight conditions were used in conjunction with the Digital DATCOM output file to determine the relevant force and moment coefficient for the equations of motion. Therefore, Equations 6 through 9 were utilized to calculate the common constant terms necessary to define the state space matrix entries. The actuator was modeled using the standard second order linear system transfer function, see Equation 10. The natural frequency for the actuator was defined to be 44 rad/s, while the damping ration was defined as 0.7.

$$\dot{\alpha} = \mu\Omega\sigma\left[-(C_L + C_D)\alpha + \frac{1}{(\mu - C_L)}q - (C_W \sin\gamma)\theta + C_L\delta\right] \tag{3}$$

$$\dot{q} = \frac{\mu\Omega}{2i_{yy}}\left[[C_M - \eta(C_L + C_D)]\alpha + [C_M + \sigma C_M(1 - \mu C_L)]q + (\eta C_W \sin\gamma)\delta\right] \tag{4}$$

$$\dot{\theta} = \Omega q \tag{5}$$

$$\mu = \frac{\rho S\bar{c}}{4m} \tag{6}$$

$$\Omega = \frac{2U}{\bar{c}} \tag{7}$$

$$\sigma = \frac{1}{1 + \mu C_L} \tag{8}$$

$$\eta = \mu \sigma C_M \tag{9}$$

$$G_{act} = \frac{{\omega_n}^2}{s^2 + 2\zeta\omega_n s + {\omega_n}^2} \tag{10}$$

Once the aircraft pitch rate state space representation was constructed as matrices in MATLAB, the "ss2tf" command in MATLAB was utilized to transform the state space matrices into a transfer function. For the actuator transfer function, both the numerator and denominator coefficient vector were constructed using the natural frequency and damping ratio of the actuator. The numerator and denominator vectors were then plugged into the "tf" MATLAB function to define the transfer function object for the actuator dynamics.

2.2.2 Block Diagram Construction

Within the Simulink autopilot model inside the avionics subsystem of the vehicle system model, the two-loop autopilot architecture needed to be implemented into the place of the original pitch controller for the block diagram construction. The inner loop of the controller was added by inserting a summing block, transfer function block, state space system block, and gain to the model. The summing block was utilized to implement the negative feedback required to close the loop. The transfer function block was then used to implement the actuator dynamics through the coefficient numerator and denominator developed in the MATLAB script. The state space system was used to implement the pitch rate state space matrices into the Simulink model from the MATLAB workspace. Finally, the gain was employed to represent the rate gyro in the feedback loop. The actuator

dynamics in the vehicle plant model are nonlinear second order actuators with saturation limits due to the physical constrains imposed by the range of angle deflections possible for the actuators in the specific vehicle configuration. This actuator model was added as a parallel to the linear second order transfer function to observe if there are any differences in the performance imposed by the saturation limits for tested flight conditions. Figure 2.2 depicts the inner loop construction.

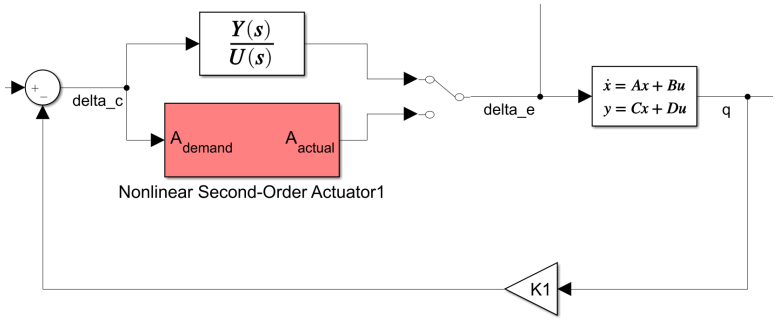

Figure 2.2: Simulink Inner Loop Block Diagram

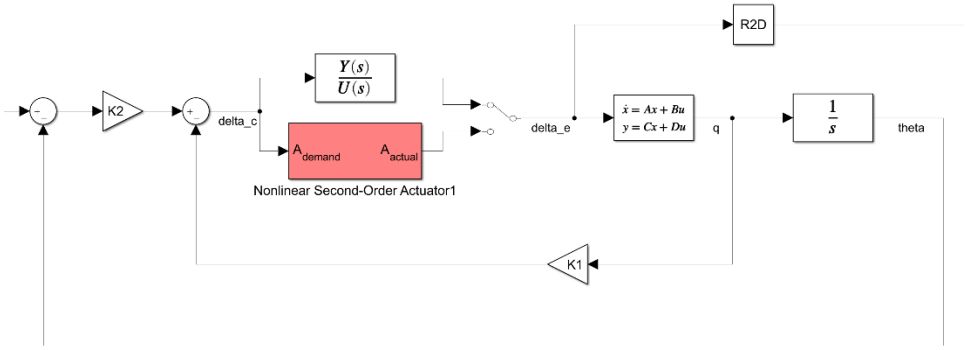

Figure 2.3: Simulink Outer Loop Block Diagram

The outer loop construction consisted of a summing block, gain block, and integrator block. The summing block was utilized to incorporate the negative feedback imposed by the pitch angle. The gain block served as the amplifier for the system to convert the reference pitch angle error into an elevator deflection command. Lastly, the integrator was placed after the inner loop to convert the pitch rate signal into a pitch angle to feedback and compare with the reference. Since the elevator deflection was the value of interest for

this controller, the signal between the actuator transfer function and the pitch rate state space plot was taken off the loop as an output since it represented the elevator deflection. The elevator deflection was determined in radians but was required in degrees in the simulation, therefore a radian to degrees block was utilized to accomplish this conversion of the outgoing signal. Figure 2.3 depicts the inner and outer loop construction in the block diagram.

2.2.3 Gain Selection

Once the transfer functions were constructed and the block diagram was inserted into the current simulation model, it was necessary to select gain values for the controller. The gains were selected using the root locus plot for the open loop transfer function of the two loops. First, the inner loop open transfer function was determined by combining the actuator transfer function and the aircraft transfer function in series using the "series" function in MATLAB. Next the "damp" function was utilized to print out the open loop poles and the "rlocus" was utilized to generate the Root Locus plot. The data selection tool in the MATLAB figure tools was utilized to pick a specific pole location and determine the corresponding gain. This inner loop gain was then added to the script to be used in the outer loop. The inner loop closed loop transfer function was developed by using the "feedback" command in MATLAB to represent the gain in the feedback loop of the system. Next, the open loop transfer function for the outer loop was developed using the closed loop transfer function of the inner loop in series with the integrator. Similarly, the "damp" command was utilized to view the poles of the open loop system and "rlocus" was used to plot the Root Locus. The second gain was then selected in a similar method to the first gain. These gains were then passed onto the gain blocks in the Simulink block diagram to

be utilized in the pitch controller within the autopilot for the flight simulation. Unfortunately, due to some errors in the modeling of the aircraft system, a stable solution did not exist for the system.

CHAPTER 3

DISCUSSION

After the gains were selected using the root locus for the open loop transfer functions, it was necessary to run the simulation to observe the behavior of the system for the selected gain in order to determine if it satisfied the performance requirements or needed refinement. This chapter will focus on the system behavior defined by the root locus of the open loop transfer function as well as the performance of the new controller design.

3.1 Root Locus for Gain Selection

The Root Locus of the inner loop is comprised of a fifth order system. This was a result of the presence of a second order actuator model in series with a third order aircraft dynamics model. Figure 3.1 shows the poles of the open loop transfer function for the inner loop. The actuator model had one zero in the numerator, causing the open loop transfer function to have an even number of combined total poles and zeros. One of the rules for drawing the Root Locus states that the "Locus lies to the left of an odd number of poles and zeros" (Dorf, Bishop, 2017, p. 435). As a result of this rule, it is known that all the points on the real axis further in the positive direction than the most positive real part of any pole will be part of the Locus. This means that the system will tend to be unstable as the gain is increased. For the given system, if one of the original poles is unstable, the system is unstable for all values of the gain. Nevertheless, a gain was selected for which the root locus for the remaining four poles is stable. Figure 3.2 depicts the Root Locus plot for the open loop transfer function of the inner loop.

Pole	Damping	Frequency (rad/seconds)	Time Constant (seconds)
-8.14e-05	1.00e+00	8.14e-05	1.23e+04
3.80e-01	-1.00e+00	3.80e-01	-2.63e+00
-8.06e-01	1.00e+00	8.06e-01	1.24e+00
-3.08e+01 + 3.14e+01i	7.00e-01	4.40e+01	3.25e-02
-3.08e+01 - 3.14e+01i	7.00e-01	4.40e+01	3.25e-02

Figure 3.1: Open Loop Transfer Function Poles for Inner Loop

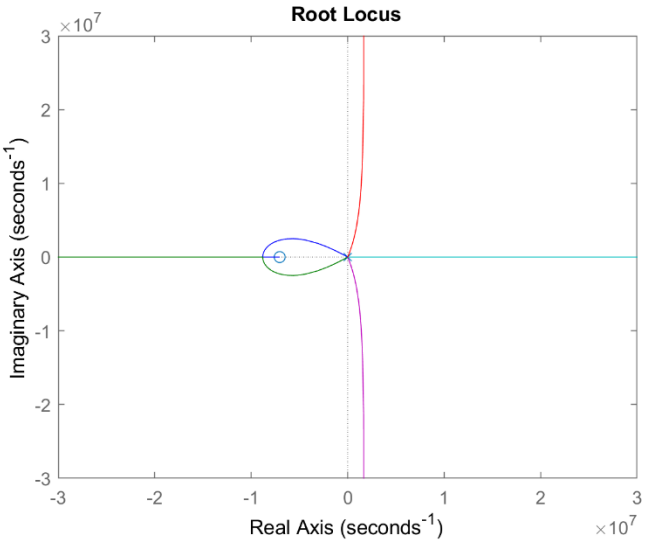

Figure 3.2: Root Locus Plot for Inner Loop

The Root Locus of the outer loop requires the gain for the inner loop to define the open loop transfer function. As a result, the performance of the outer loop is directly dependent on the value selected for the gain of the inner loop. The poles of the open loop transfer function for the outer loop can be seen in Figure 3.3. Figure 3.4 depicts the Root Locus plot for the open loop transfer function of the outer loop. The presence of an integrator causes there to be a pole at the origin on the Root Locus plot. Due to the previously noted Root Locus plotting rule, the combined number of poles and zeros for the outer loop causes the system to have no viable gain solution to make the system stable.

Pole	Damping	Frequency (rad/seconds)	Time Constant (seconds)
0.00e+00	-1.00e+00	0.00e+00	Inf
9.28e+00 + 1.23e+01i	-6.01e-01	1.54e+01	-1.08e-01
9.28e+00 - 1.23e+01i	-6.01e-01	1.54e+01	-1.08e-01
-2.10e+01	1.00e+00	2.10e+01	4.76e-02
-2.98e+01 + 3.29e+01i	6.71e-01	4.44e+01	3.36e-02
-2.98e+01 - 3.29e+01i	6.71e-01	4.44e+01	3.36e-02

Figure 3.3: Open Loop Transfer Function Poles for Outer Loop

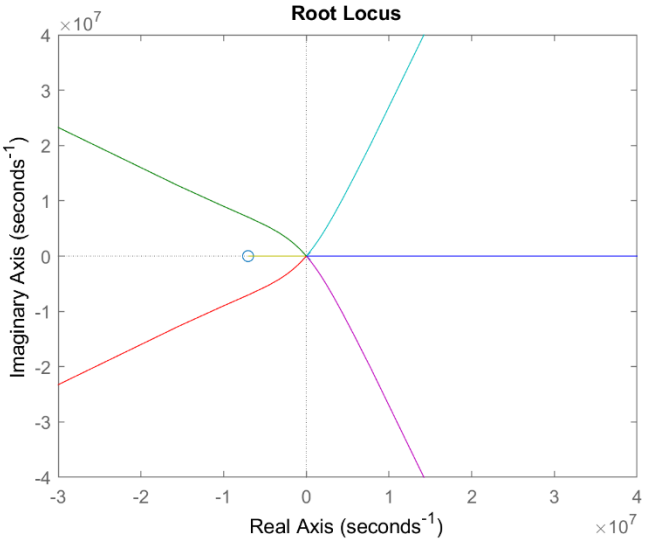

Figure 3.4: Root Locus Plot for Outer Loop

The gains selected are not viable solutions for the controller design, therefore the simulation will not observe the desired behavior. This is possibly due to an error in the simplification of the equations of motion or the presence of an uncontrollable system. For the purposes of showing the simulation output altitude plot, an inner loop gain of 3060 was used and an outer loop gain of 5000 was used.

3.2 Simulation Results

The flight simulator was run for thirty seconds using the controller design discussed in the Root Locus discussion above. Figure 3.5 is a plot of the desired altitude along with the time history of the aircraft altitude throughout the simulation. The aircraft observed the

unstable behavior expected due to the Root Locus for the outer loop being unstable for all values of gain.

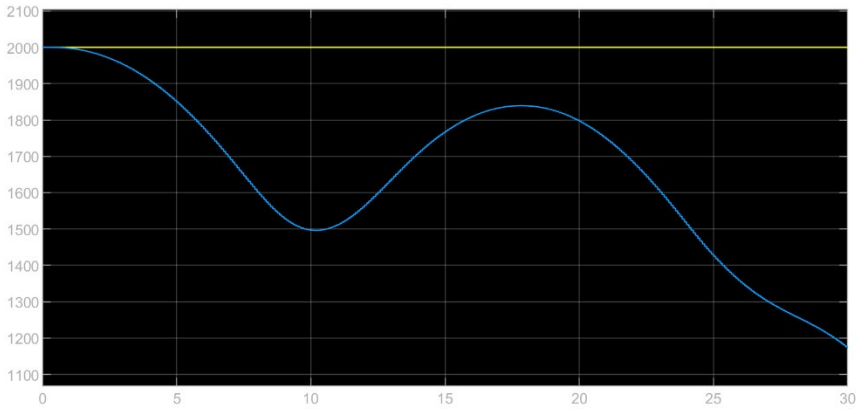

Figure 3.5: Simulation Output Altitude Plot

CHAPTER 4

CONCLUSION

This project allowed for the exploration of a controller design for a flight simulator of a fixed-wing aircraft. The project did not turn out as expected; from the beginning, many hurdles were faced in finding a viable flight simulator. As the project progressed, additional challenges were faced due to the vehicle being a design proposal with little available documentation. Lastly, the presence of the COVID-19 pandemic, which caused campus to close, affected mentoring availability to aid in progressing toward a viable autopilot design for the selected vehicle simulator. However, in the process of trouble shooting and modifying the simulator and controller design, many lessons were learned about the relevant topics in control design and implementation in a flight simulator. These topics will directly translate to relevant skills necessary to excel in a career as a GNC Engineer. All in all, the project was a useful exercise to gain experience in topics outside of the scope of general coursework at UTA.

APPENDIX A

MATLAB SCRIPT FOR GAIN SELECTION

```
%{
Lorenzo Novoa Spring 2020
Honors Senior Capstone Project
Autopilot Design for a Fixed-Wing Aircraft Simulator
%}

clear, clc, close all

% Load file containing Digital DATCOM
load('init_file.mat')

% Inputs for
rho = 1.00649; % Density [kg/m^3]
V = 93.08; % Speed [m/s]
gamma = .01714-.0171; % Flight path angle [rad]
m = 770.8437;%5670.23; % [kg] Based on Cessna Denali Weight Limit

Mach = 0.28649; % Mach [N/a]
alpha = .0171; % angle of attack [rad]
alpha_deg = alpha*180/pi; % angle of attack [degrees]
alt = 2000; % altitude [m]

S = statdyn{1,1}.sref/(3.28^2); % Planform area [m^2]
cbar = statdyn{1,1}.cbar/(3.28); % mean geomentric chord [m]

C_L = statdyn{1,1}.cl(7,3,2); % coefficient of lift [N/a]
C_D = statdyn{1,1}.cd(7,3,2); % coefficient of drag [N/a]
C_M = statdyn{1,1}.cm(7,3,2); % coefficient of moment [N/a]

C_W = m*9.81/(S*.5*rho*V^2); % coefficient of weight [N/a]
Iyy = 1693.416634404; % moment of inertia [kg-m^2] based on Cherokee PA-28
iyy = Iyy/(S*m); % normalized moment of inertia [N/a]

mu = rho*S*cbar/(4*m);
Omega = 2*V/cbar;
sigma = 1/(1+mu*C_L);
eta = mu*sigma*C_M;

% pitch rate to elevator deflection State-space system
A(1,:) = mu*Omega*sigma*[-(C_L+C_D), 1/(mu-C_L), -(C_W*sin(gamma))];
A(2,:) = mu*Omega/(2*iyy)*[(C_M-eta*(C_L+C_D)), C_M+sigma*C_M*(1-
mu*C_L),0];
A(3,:) = [0, Omega, 0];
B = [mu*Omega*sigma*C_L; mu*Omega*eta*C_W*sin(gamma)/(2*iyy); 0];
C = [0 0 1];
D = [0];
```

```
[num_ac,den_ac] = ss2tf(A,B,C,D);
G_ac = tf(num_ac,den_ac);

% Actuator Transfer Function Construction
num_act = [wn_act^2];
den_act = [1 2*z_act*wn_act wn_act^2];
G_act = tf(num_act,den_act);

% Root Locus Plot for inner loop gain
[A1,B1,C1,D1] = tf2ss(conv(num_act,num_ac),conv(den_act,den_ac));
OLTF_IL_ss = ss(A1,B1,C1,D1);
OLTF_IL_ss = prescale(OLTF_IL_ss);
[num_il,den_il] = ss2tf(A1,B1,C1,D1);
OLTF_IL = tf(num_il,den_il)
damp(OLTF_IL)
rlocus(OLTF_IL_ss)

K1 = 3060;

% Root Locus Plot for outer loop gain
integrator = tf([1],[1 0]);
CLTF_IL = feedback(OLTF_IL,-K1)
damp(CLTF_IL)
OLTF_OL = series(CLTF_IL,integrator)
damp(OLTF_OL)
figure()
rlocus(OLTF_OL)

K2 = 5000;
```

REFERENCES

(n.d.). Retrieved from http://ctms.engin.umich.edu/CTMS/index.php?example=AircraftPitch§ion

Anderson, J. D. (2017). *Fundamentals of aerodynamics*. New York, NY: McGraw-Hill Education.

Dorf, R. C., & Bishop, R. H. (2017). *Modern control systems*. Essex: Pearson Education.

Etkin, B. (1959). *Dynamics of flight Stability and Control*. New York: John Wiley & Sons.

Lightweight Airplane Design. (n.d.). Retrieved April 10, 2020, from https://www.mathworks.com/help/aeroblks/lightweight-airplane-design.html

McRuer, D. T., Ashkenas, I., & Graham, D. (1973). *Aircraft dynamics and control*. Princeton: Princeton University Press.

Roskam, J. (2011). *Airplane flight dynamics and automatic flight controls*. Lawrence, KS: DARcorporation.

BIOGRAPHICAL INFORMATION

Lorenzo Novoa is a senior studying Aerospace Engineering at the University of Texas at Arlington (UTA). He has been an active member in the Honors College since the start of his education at UTA. In addition to the Honors College, Lorenzo has been a member of a few organizations on campus including: Aero Mavericks, the UTA AIAA student branch, Vertical Flight Society UTA Chapter, and the Texas Eta Tau Beta Pi Chapter. During his time at UTA, Lorenzo found a passion for the field of Guidance, Navigation, and Control (GNC) and is intent on pursuing a career with a focus on GNC after completing his degree. During the Summer of 2019, Lorenzo participated in an internship with Lockheed Martin Missile and Fire Control (MFC) in Grand Prairie as a GNC Intern on a hypersonic program. At the end of the Summer, he was offered a fulltime employment offer to return to Lockheed Martin MFC as a GNC Engineer upon completion of his degree. Lorenzo accepted the offer and will be starting his new position in June 2020.