DESIGN AND DEVELOPMENT OF CONTROL SYSTEMS FOR UAVs

Alua Yermekova, BET

Submitted in fulfilment of the requirements for the degree of Master of Science in Electrical and Computer Engineering

School of Engineering and Digital Sciences
Department of Electrical and Computer Engineering
Nazarbayev University

53 Kabanbay Batyr Avenue,
Nur-Sultan, Kazakhstan, 010000

Supervisors: Prashant Jamwal
Refik Kizilirmak

18.03.2022

DECLARATION

I hereby, declare that this manuscript, entitled "Design and development of control systems for UAVs", is the result of my own work except for quotations and citations which have been duly acknowledged.

I also declare that, to the best of my knowledge and belief, it has not been previously or concurrently submitted, in whole or in part, for any other degree or diploma at Nazarbayev University or any other national or international institution.

Name: Alua Yermekova

Date: 18.03.2022

Table of Contents

Abstract

In this work, the closed-loop control of a fixed-wing UAV system was designed and a mathematical model was derived and stabilized using the PID controller with the Feed-forward compensation method to reduce the non-minimum phase. In the first part of the thesis project, a comprehensive systematic literature review on general UAV structures and their types, as well as their application was presented. Later the overall aircraft dynamics alongside the generalized roll angle model of a fixed-wing was pointedly described. Moreover, the nonlinear dynamics of both matrix differential equations and vector forms (traditional state-space form) were derived.

After modelling the system, the closed-loop control law was designed using the PID controller in MATLAB script. The reference generation was performed by using the two types of reference: simple (unit step signal and ramp) and complex (sine wave, square wave and their combination). Generated reference points were applied to test the controller behavior in various cases. The obtained closed-loop model had a negative transient behavior, which means it has a positive zero and caused the non-minimum phase system with even-numbered RHP (Right-hand plane) zeros. To reduce it, the various methods were tested and among them, the feed-forward compensation method showed the best performance in terms of usage, easy implementation, settling time, overshoot, and margin values.

Acknowledgements

Primarily, I would like to gratitude to the Lord of the Universe Allah for everything in my life and for having an opportunity to study and be a part of Nazarbayev University. I would also like to extend my deepest gratitude to Professor Prashant Jamwal and Farzad Salmasi for the knowledge that they shared with me and their guide during my project. I'm extremely grateful to Professor Refik Kizilirmak who gave me the chance to work in the Unmanned Aerial Vehicle (UAV) lab and for his support during the project. Through this project, I started to explore the UAV world and found wonderful teammates. I would like to thank Professor Mehdi for his understanding and support during master's program. Great thanks to my husband Kuanysh for his patience and support. I would like to express my sincere gratitude to my parents for supporting me in any situation.

List of Abbreviations & Symbols

UAV	Unmanned Aerial Vehicle
PID	Proportional Integrative Derivative Controller
PI	Proportional Integrative Controller
EKF	Extended Kalman Filter
GPS	Global Positioning System
AMM	Autonomous Mission Manager
HILS	Hardware-in-the-loop simulation
DOF	Degree of Freedom
PFCD	Parallel feedforward compensation with derivative method
MIMO	Multi-input- Multi- output
RHP	Right hand plane
LHP	Left hand plane
UFRD	Uncertain Frequency Response Data
FFC	Feed- Forward Compensation

List of Tables

List of Figures

Chapter 1- Introduction

1.1 UAV background dynamics

Nowadays UAV design and application are becoming more popular since it is used in the military [1], agriculture [2], photography [3], position tracking [4] and artificial techniques [5]-[7].

Electromagnetic jamming plays an important role in the UAV security system, for that reason, security awareness based on semantic analysis was developed and tested in [1]. For example, in military services the UAV is applied to notify the security situation awareness, which can be categorized into three main stages as situation parameter extraction, understanding the situation and future situation prediction. Therefore, it is very important developing the UAV electromagnetic interference perception of UAV [1].

In addition, UAV is used to protect the large-scaled plants and spray the pesticides. It has a large flight time and GPS flight control [2]. Researchers compared single rotor and multi-rotor UAVs and design features of common and plant protection UAVs and gave suggestions on how to enhance the design structure. Plant protection UAV is applied in large-scale farmland and helps to protect the environment and has such drawbacks as hose of the plant can be short circuit, heavy system, instability due to medicine box, poor quality of the battery. To overcome these issues, researchers in [2] investigated the multi-rotor plant protection UAV. So, they assumed that in-plant protection, usage of multi-rotor UAV gives good quality of battery and stability of the system rather than single-rotor UAV [2].

In some natural conditions, there is a necessity of taking a photo, for that reason, the photography system of UAV plays a significant role. The authors in [3] made experiments for four

years. Therefore, the given system helps to monitor various situations and take full-color, high-quality images.

If a UAV works with long distances, finding the correct location is vital. So, authors in [4] developed UAVs active geolocation method which can measure multiple angles and minimize the noise of sensors. To achieve the high accuracy of this method, the linear regression algorithm will be applied for mounting error which is identified as a combination of navigation module and electro-optical device. There are two types of electro-optical payload (significant sensor of UAV) active and passive. The passive localization couldn't work with long distances and the accuracy of active localization is better.

In addition, UAVs with Artificial Intelligence are rapidly developing and enhancing the borders of UAVs application. One of such inventions is the autonomous mission manager [5] Autonomous mission manager (AMM)) in a small UAV system is the breaking news for the AI world. To implement this management system, hybrid and hierarchical structures were taken into account in [5].

The authors in [6] suggested a flexible way of designing UAV based on computer-aided design, which will optimize the developing time of UAV and the cost of production. First of all, the authors highlighted the steps of UAV construction and drawbacks of nowadays design methods. Therefore, processors that consist of thematic coprocessors with dynamic data integration will help to cope with this issue. Thematic coprocessors have various applications and they are divided as table, text, math, graphics coprocessors. In addition, these coprocessors can be used in complex operations.

The authors in [7] discussed the importance of collision avoidance with smart UAV and hardware-in-the-loop simulation (HILS) was tested in case of multiple distributions which were

created by visual aircraft. There are different approaches to avoid collisions such as optical sensor, radar, and Automatic Dependent Surveillance where they have drawbacks in case of volume and effective usage.

UAV gave the possibility to save human life without any risk, since it is applicable for military operations, civilian applications and care for nature. In addition, UAV has reduced cost for that reason, it is commonly used in the market [8].

1.2 Fixed-wing UAV Control

There are several types of UAVs: fixed-wing and quad-rotor. Quad-rotor UAV has a small size and simple design structure, but due to the nonlinear stability, it has some issues with attitude and altitude. Due to the vulnerability to wind of the quad-rotor, as the control option the fixed-wing was chosen since, it has more precise stability. The main distinction between fixed-wing and quad-rotor UAV is that the quadrotor has four motors, where 2 of them move in clockwise and remaining in an anti-clockwise direction [8].

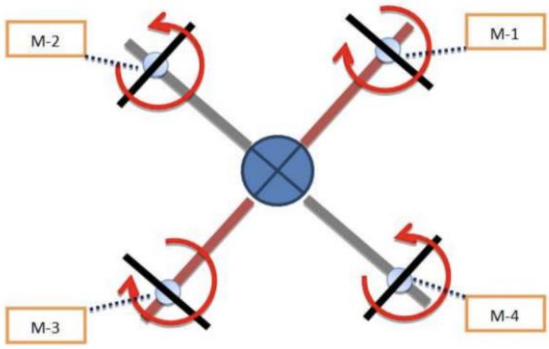

Figure 1.2.1 Quad-rotor UAV and its direction [8]

Fixed -wing UAV is the most common type among researchers. The main issue with this UAV is the stabilization of roll angle due to its nonlinear nature [9]. Fixed-wing UAV has the 6

Degree of Freedom (DOF) which can be categorized as Forward, sideway, vertical, yaw, pitch and roll[10]. In order to analyze the six degrees of freedom (DOF) in [10] the model was constructed and simulated with MATLAB Simulink. Let's identify how to evaluate the six degrees of freedom (DOF). In order to model the dynamics of UAVs the information about the internal frame, which is North, East, Down (NED) and body-fixed frame should be identified (Fig.1.2.2).

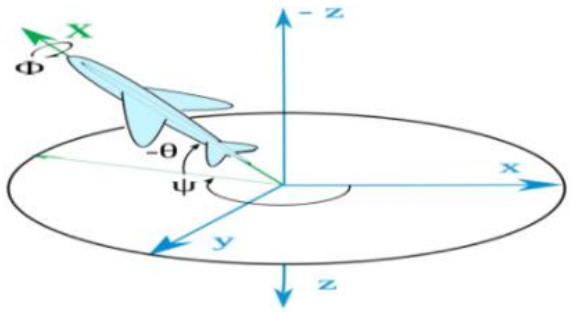

Figure 1.2.2 NED and body frame of fixed-wing UAV [10]

The following rotation matrix is used to fit the rotation and motion in the Body frame:

$$R_b^n = \begin{bmatrix} c_\Psi c_\theta & -s_\Psi c_\phi + c_\psi s_\theta s_\phi & s_\psi s_\phi + c_\Psi s_\theta c\phi \\ s_\psi c_\theta & c_\psi c_\phi + s_\Psi s_\theta s_\phi & -c_\psi s_\phi + s_\Psi s_\theta c_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix} \quad \textbf{(1.2.1)}$$

Table 1.2.1: Rotation matrix coefficients

Abbreviation	Description
θ	pitch angle
Ψ	roll angle
ϕ	yaw angle
s	sine
c	cosine

f_0^b	3 forces: aerodynamic, actuator, and gravitational
m_0^b	3 moments

As a result, 6 Degree of Freedom can be identified in the following way:

$$\begin{bmatrix} f_0^b \\ m_0^b \end{bmatrix} = \begin{bmatrix} X \\ Y \\ Z \\ L \\ M \\ N \end{bmatrix} = \begin{bmatrix} Forward\ force \\ Sideway\ force \\ Vertical\ force \\ Roll\ moment \\ Pitch\ moment \\ Yaw\ moment \end{bmatrix} \quad \textbf{(1.2.2)}$$

Therefore, the overall dynamics is defined by:

$$M\dot{v} + C(v)v + g(n) = \tau + \omega \quad \textbf{(1.2.3)}$$

Table 1.2.2: Overall dynamics coefficients

Abbreviation	Description
M	Inertia matrix
$C(v)$	Coriolis matrix
g(η)	Centripetal matrix
τ	The vector of control inputs
ω	The vector of disturbances

So, the equation of motion can be generalized in the following way:

$$\begin{cases} X = m(\dot{U} + QW - RV + gsin\theta) \\ Y = m(\dot{V} + UR - PW - g\cos\theta\, sin\emptyset) \\ Z = m(\dot{W} + PV - QU - gcos\theta cos\emptyset) \\ L = I_X\dot{P} - I_{XZ}\left(\dot{R} + PQ\right) + (I_Z - I_Y)QR \\ M = I_Y\dot{Q} - I_{XZ}(P^2 - R^2) + (I_X - I_Z)PR \\ N = I_Z\dot{R} - I_{XZ}\dot{P} + (I_Y - I_X)PQ + I_{XZ}QR \end{cases} \quad \textbf{(1.2.4)}$$

The system responses and impact of disturbances and noise on the system were analyzed for PID controllers with and without Extended Kalman Filter (EKF). Basically, the PID controller will be implemented to stabilize the system according to the desired input. PID controller fits well with controllability, however it is not appropriate for noisy conditions. Therefore, the Extended Kalman Filter was added to remove the additional noises of the system [9].

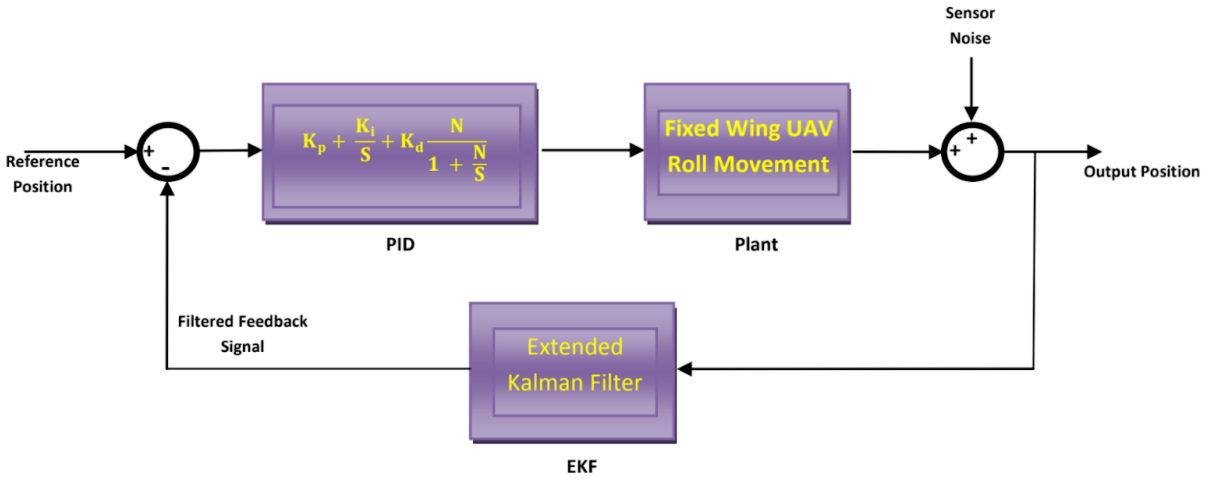

Figure 1.2.3 Block diagram of Feedback Control System with EKF[9]

Besides, the fixed-wing UAV movement can be divided into lateral and longitudinal ways. The main difference between these two subsystems is that in the longitudinal the traditional PID controller will be used to get desired altitude, while in the lateral subsystem the non-linear control methods will be taken into consideration to find the desired trajectory [12].

Depending on the aim of the UAV, the controller structure can be different. Basically, the controller has three main loops as inner [8], [11]-[14], outer [3], [12]-[17] and autonomous [12], [18]-[20].

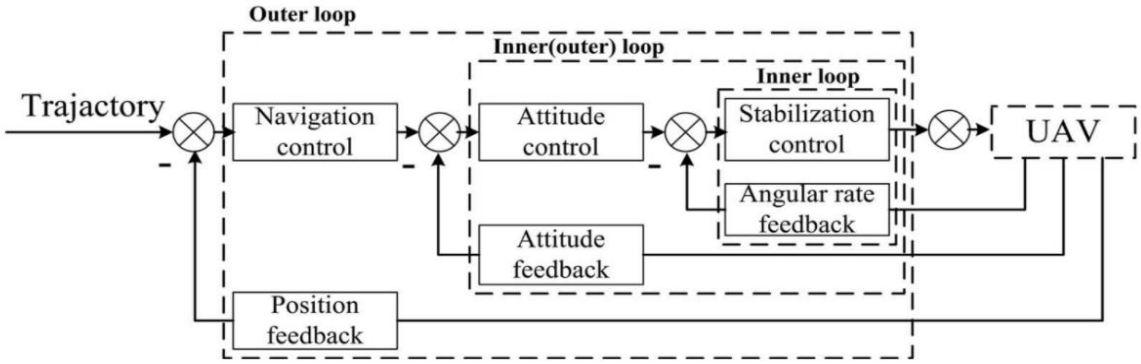

Figure 1.2.4 The basic control principal diagram [12]

Basically, the outer loop is responsible for navigation control, while to stabilize the system [12] inner loop is used, shown in Figure 1.2.4, where the inner loop is used to stabilize the system. Basically, the design part starts from inner loop control. The attitude control is in second place. Only after designing the inner(outer) loop, the navigation control can be investigated. The attitude control is in second place. Only after designing the inner(outer) loop, the navigation control can be investigated. Besides the classical controller structure, the UAV controller can be based on fuzzy logic [14], Lagrange Euler [21], integral backstepping and hierarchical [22], delayed control systems [23], and aerial refueling [24]. In order to stabilize the UAV parameters, the PID controller was applied by authors in [11]- [12], [18], [25].

For example, to work effectively with nonlinear UAVs, longitudinal altitude with the fuzzy PID will be implemented in [14]. Classical PID controllers are not convenient in terms of contradiction between dynamic characteristics and steady-state characteristics and unstable behavior under unfavorable natural conditions. On the other hand, usage of fuzzy controllers gives

accurate control even in high temperatures and stronger robustness. The comparison of two types of PID with pitch control are given and fuzzy PID showed the best result in [14].

In [21], the authors gave a detailed explanation of the dynamic modeling of fixed-wing UAVs with the Lagrange- Euler formula. After identifying the forces which affect the UAV, the concept of Lagrange was applied, then the equation of the system was under small-angle approximation.

Integral backstepping and hierarchical control types were applied for UAVs in [22]. The mathematical model of UAV was derived in stabilization, body, and wind-axes coordinates. The first control type helped to minimize the disturbance effects and guarantee closed-loop stability. To work without any issues with the high velocity and height of the system the hierarchical approach was investigated in [22].

In this paper [23], the importance of decreasing the time delay of UAV in in-the-loop was analyzed, since it affects the system instability. UAV control can be autonomous or main in the loop control, the first control method does not fit in case of latency of the system. To overcome this problem, researchers used a state predictive compensation algorithm and tested it with the example of pitch control and verified the improvement of time delay.

Authors in [24] simulated UAVs during aerial refueling and redesigned and simplified the traditional control theory and control laws. The given control system was simulated with MATLAB Simulink and after that, it was analyzed with Microsoft Visual Studio and the results of redesigned control with aerial refueling were confirmed in [24].

Researchers in [11] analyzed the robust deadbeat PID controller with one cascade gain tuning method via the pitch axis of the UAV. The controller implementation with a closed-loop control system was provided and two types of flight loops were categorized as inner and outer

loops. Several calculations and results showed that the given PID approach has high robustness and good performance of time-domain in dependency of system fluctuations.

In this paper [12], the authors identified two control loops of fixed-wing UAVs as outer which is responsible for navigation control which works with heading angle and inner is responsible for attitude and stabilization control where such parameters as roll and pitch angles are controlled. In addition, the navigation control was observed in lateral and longitudinal sides. PID controller was applied in longitudinal control to take good altitude. The flight test of this control system showed that it has good performance to be applied for navigation purposes.

Authors in [25] identified and simulated with PID controller to design the controller of Kiteplane with three types of error. In addition, altitude control was not considered in this paper, because of stable dynamics and the wind estimation approach was formulated and verified. There were several challenges during these experiments such as position and speed which means three types of error as altitude, horizontal position, and direction error, for each of them PID structure was applied. Moreover, the design method showed good performance, when THE robust analysis was applied.

Moreover, with PID, Extended Kalman Filters were applied to stabilize the system under disturbance condition [8]-[10], [17]. Extended Kalman Filter used to remove the noises caused by sensors. The comparison of results with Proportional (P) controller, Integral (I) controller and Derivative (D) controllers and Extended Kalman Filter was provided in papers [9]- [10], [13], [26]. In some cases, these types of control were not effective, therefore, to get good performance adaptive and expert types of PID were implemented by authors in [14], [27]-[28].

So, traditional PID controller has three terms [29]:

$$G(s) = K_P + K_I\frac{1}{s} + K_D s = K_P\left(1 + \frac{1}{T_I s} + T_D s\right) \qquad \textbf{(1.2.5)}$$

where K_P is the proportional gain, K_I the integral gain, K_D the derivative gain, T_I the integral time constant and, T_D the derivative time constant. In order to understand the effect of the proportional, integral and derivative gains, the following information was summarized by the authors in [29]:

Table 1.2.3: The effect of P,I , D parameters

Closed loop response	Rise time	Overshoot	Settling Time	Steady-State Error	Stability
Increasing K_P	Decrease	Increase	Small increase	Decrease	Degrade
Increasing K_I	Small Decrease	Increase	Increase	Large decrease	Degrade
Increasing K_D	Small Decrease	Decrease	Decrease	Minor Change	Improve

Therefore, the proportional part provides the overall control action, integral part reduces the steady state error and with the help of derivative part the transient response can be improved. During my thesis project, I will check the effect of the P, I, D controller to the system overall and by the combination of them.

1.3 Motivation

These days, the most commonly used UAV is the fixed-wing UAV, since it has longer flight capabilities. Therefore, it is used in such situations as localization, agriculture and mapping. Basically, fixed-wing UAV is non-linear, so the stabilization of UAV is a significant task for

engineers around the world. In this thesis, the fixed-wing UAV's roll control enhancement will be discussed.

Authors in [9] implemented the controller with PID and Extended Kalman Filter (EKF) to stabilize the roll angle of the fixed-wing UAV. The comparison of the usage of only PID and PID and EKF was considered in [10].

Taking account of all of the papers discussed in this work, the several conclusions can be made. There is still a gap in developing an optimal control law for the fixed-wing UAV under uncertainty conditions. Especially in my Thesis work I focus on roll-angle control and stabilization of fixed-wing type of UAV. Secondly, because of UAVs high cost in the market, the economic design methods which helped to optimize UAVs cost were revealed.

In order to get the accurate stabilization under uncertainty conditions, a Proportional Integrative Derivative (PID) controller will be implemented to the lateral subsystem of UAVs. In order to check the stability of the system for the different cases, robustness of the parallel feedforward compensation with derivative (PFCD) method will be proved in this thesis work.

1.4 Goals and Objectives

The key goal of the thesis is to design the stability control system for UAVs under perturbed conditions. To achieve that the following objectives were classified:

1. Firstly, the dynamical state equations of UAV will be found and the state equations will be transformed to transfer functions for the purpose of controller design.
2. In the second step, controller performance will be simulated with MATLAB Simulink. In addition, the step responses will be taken from this software, to understand how the UAV will change during the simulation time.

3. In the third step, the necessary PID controller parameters will be taken by using the autotuning method to get the accurate stabilization under uncertainty conditions.
4. In order to check the stability of the system for the different cases, robustness of the parallel feedforward compensation with derivative (PFCD) method will be proved in this thesis work.

Chapter 2- Mathematical modelling of fixed-wing UAV

2.1 General information about fixed-wing UAV

In order to derive the system dynamics, firstly let's define the type of the UAV which will be investigated during my master thesis. The physical characteristics of F-16 model can be identified in the following way:

Table 2.1.1: The physical characteristics of F-16

UAV Property	Abbreviation	Value
Weight	W	20.500
Moment of Inertia:	J_{xx}	9.456
	J_{yy}	55.814
	J_{zz}	63.100
	J_{xz}	982
Wing dimensions:	Span	30 ft
	Area	300 ft^2
	m.a.c	11.32 ft
Engine Angular Momentum		160 slug-ft2/s

Table 2.1.2: Control Surface Actuator Models

	deflection	rate limit	time constant
Elevator	±25.0°	60°/s	0.0495 s lag
Airlerons	±21.5°	80°/s	0.0495 s lag
Rudder	±30.0°	120°/s	0.0495 s lag

Basically, there are 3 main control surfaces of fixed-wing UAV: ailerons(rolling), an elevator(pitching), rudder(yawing). The detailed information about control surfaces of fixed-wing UAV is provided below:

$$X = \begin{bmatrix} x \\ y \\ z \\ U \\ V \\ W \\ \phi \\ \theta \\ \Psi \\ P \\ Q \\ R \end{bmatrix} \qquad \textbf{(2.1.1)}$$

Table 2.1.3: The detailed information about the control surface coefficients

Abbreviation	Aircraft states	Description
x	position	
y	position	
z	position	
U	linear velocity	
V	linear velocity	
W	linear velocity	
ϕ	angular position-attitudes	roll
θ	angular position-attitudes	pitch
Ψ	angular position-attitudes	yaw
P	angular velocity	
Q	angular velocity	
R	angular velocity	

The moments and forces which affect the aircraft are summarized in Fig. 2.1.2 below.

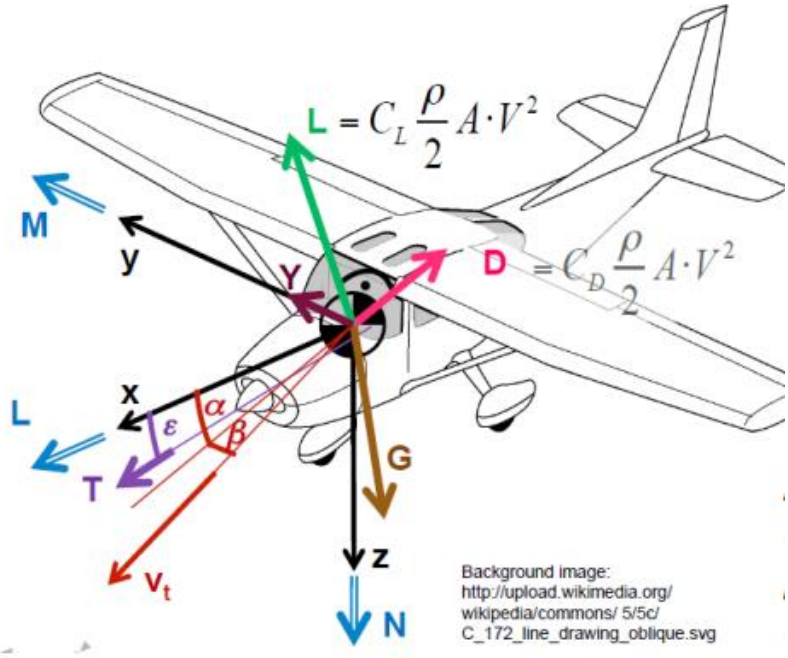

Figure 2.1.2 The moments, forces and angles of fixed-wing UAV [30]

In order to understand the formulas and characteristics which affect the UAV the following tables will be identified:

Table 2.1.4: The detailed information about the forces, moments and angles of UAV

Moments	Forces	Angles
L:Roll moment	L:Lift	α: :angle of attack
M: Pitch moment	D:Drag	β: Sideslip angle
N:Yaw moment	Y: Sideslip force	ε : Thrust-vector angle
	T:Thrust	
	G:Weight	

Table 2.1.5: The UAV moments and formulas

Description	Abbreviation	Formula
drag	D	$\bar{q}SC_D$
lift	L	$\bar{q}SC_L$
crosswind force	C	$\bar{q}SC_c$
rolling moment	l_w	$\bar{q}SbC_l$
pitching mom	m_w	$\bar{q}S\bar{c}C_m$
yawing moment	n_w	$\bar{q}SbC_n$

2.2 Control subsystems of fixed-wing UAV

To control the UAV system, the plant will be divided into two subsystems as longitudinal and lateral control. In Fig. 2.2.1, the two subsystems are shown.

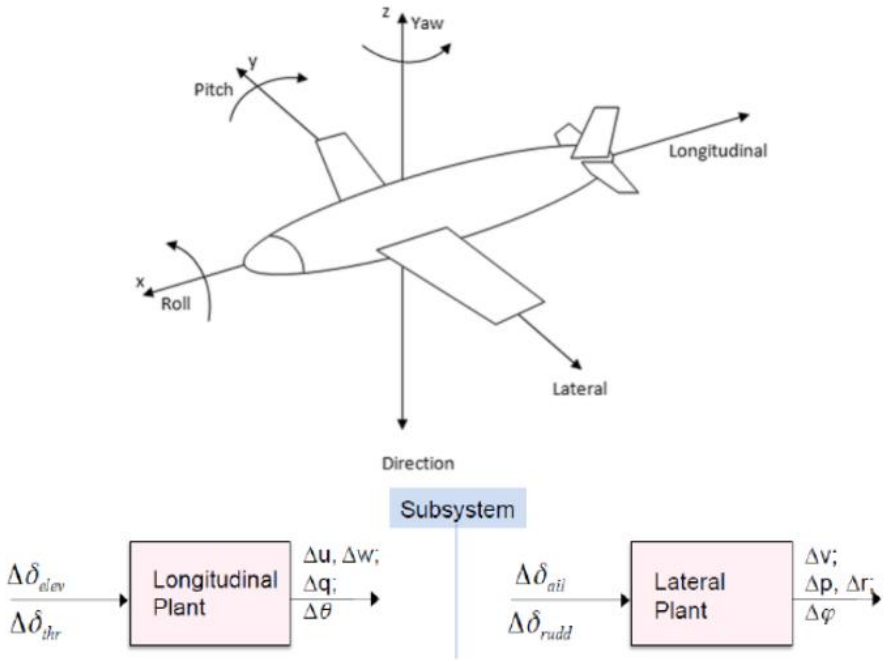

Figure 2.2.1 Longitudinal and later control of fixed-wing UAV[30]

For example, the longitude control has two inputs elevator and throttle. In the lateral subsystem, there are two inputs such as aileron and rudder.

Our project is focused on the inner loop which means stability augmentation systems. Inner loop controls: pitch, yaw, and roll damper of UAV. The inner control system of the UAV consists of four inputs like elevator, rudder, throttle, and aileron. Fig. 2.2.2 shows the inner and outer loops of the control system where the inner loop is responsible for stabilization and angular rate feedback. While the outer loop is responsible for navigation and position control.

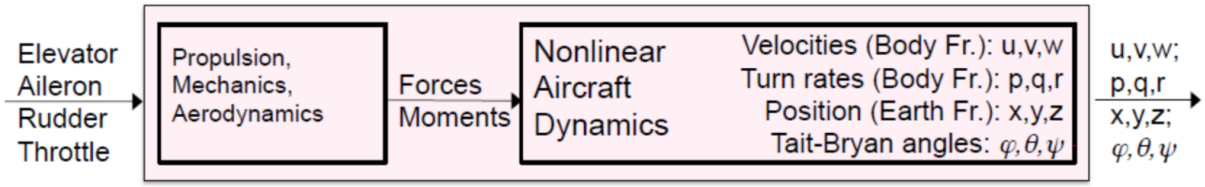

Figure 2.2.2 Inner and outer loop of flight control system [30]

Where,

Input vector:

$$u = \begin{bmatrix} \delta_{elev} \\ \delta_{ail} \\ \delta_{rudd} \\ \delta_{thr} \end{bmatrix} \quad \textbf{(2.2.1)}$$

State vector:

$$x = [u, v, w, p, q, r, x, y, z, \phi, \theta, \varphi]^T \quad \textbf{(2.2.2)}$$

Output vector:

$$y = \begin{bmatrix} V_T = \sqrt{u^2 + w^2} \\ v \\ \phi \\ \theta \end{bmatrix} \quad \textbf{(2.2.3)}$$

In the longitudinal subsystem, the dynamics will be derived in the following way:

$$x = [\alpha\ q\ V_T\ \ \theta]^T \quad \textbf{(2.2.5)}$$

$$u = [\delta_e\ \delta_t]^T \quad \textbf{(2.2.6)}$$

$$E\dot{x} = Ax + Bu \quad \textbf{(2.2.7)}$$

$$E = \begin{bmatrix} V_{T_E} - Z_{\dot{\alpha}} & 0 & 0 & 0 \\ -M_{\dot{\alpha}} & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \textbf{(2.2.8)}$$

$$B = \begin{vmatrix} Z_{\delta e} & -X_{\delta t}\sin(\alpha_e + \alpha_T) \\ M_{\delta e} & M_{\delta t} \\ X_{\delta e} & X_{\delta t}\cos(\alpha_e + \alpha_{T)} \\ 0 & 0 \end{vmatrix} \quad \textbf{(2.2.9)}$$

$$A = \begin{bmatrix} Z_\alpha & V_{T_e} + Z_q & Z_V - X_{T_V}\sin(\alpha_e + \alpha_T) & -g_D \sin y_e \\ M_\alpha + M_{T_\alpha} & M_q & M_V + & 0 \\ X_a & 0 & X_V + X_{T_V}\cos(\alpha_e + \alpha_T) & -g_D \cos y_e \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad \textbf{(2.2.10)}$$

After substitution the necessary values to the formulas provided above, the longitudinal subsystem matrixes will be identified in the following way:

$$A = \begin{bmatrix} -2.0244E-02 & 7.8763E+00 & -3.2170E+01 & -6.5020E-01 \\ -2.5372E-04 & -1.01900E+00 & 0.0000E+00 & 9.0484E-01 \\ 0.0000E+00 & 0.0000E+00 & 0.0000E+00 & 1.0000E+00 \\ 7.9472E-11 & -2.4982E+00 & 0.0000E+00 & -1.3861E+00 \end{bmatrix} \quad \textbf{(2.2.11)}$$

Table 2.2.1: The longitudinal subsystem characteristics

Description	Value	Mode description	Mode value
angle of attack α	$-1.2039 \pm j1.4922$	short-period mode	$T = 4.21s$,$\delta = 0.628$
pitch rate q	$-1.2039 \pm j1.4922$	short-period mode	$T = 4.21s, \delta = 0.628$
total velocity V_T	-0.0087297±j0.073966	phugoid mode	$T = 89.4\ s, \delta = 0.117$
pitch θ	-0.0087297±j0.073966	phugoid mode	T=89.4 s, δ=0.117

Basically, in my thesis project, the F-16 UAV model with only a rolling option will be considered. The main focus will be on the aileron, exactly on roll control which is the lateral subsystem. Dynamics of the system: In the lateral subsystem, the A

matrixes will be in the following way:

$$x = [\beta\ \phi\ \ p_s\ r_s]^T \quad \textbf{(2.2.12)}$$

$$u = [\delta_a\ \delta_r]^T \quad \textbf{(2.2.13)}$$

$$E\dot{x} = Ax + Bu \quad \textbf{(2.2.14)}$$

$$E = \begin{bmatrix} V_{T_E} & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \textbf{(2.2.15)}$$

$$B = \begin{bmatrix} Y_{\delta_a} & Y_{\delta_r} \\ 0 & 0 \\ \acute{L_{\delta_a}} & \acute{L_{\delta_r}} \\ \acute{N_{\delta_a}} & \acute{N_{\delta_r}} \end{bmatrix} \quad \textbf{(2.2.16)}$$

$$A = \begin{bmatrix} Y_\beta & g_D cos\theta_e & Y_P & Y_r - V_{T_E} \\ 0 & 0 & \frac{cy_e}{e\theta_e} & \frac{sy_e}{c\theta_e} \\ \acute{L_\beta} & 0 & \acute{L_p} & \acute{L_r} \\ \acute{N_\beta} & 0 & \acute{N_p} & \acute{N_r} \end{bmatrix} \quad \textbf{(2.2.17)}$$

After substitution the necessary values to the formulas provided above, the lateral subsystem matrixes will be identified in the following way:

$$\dot{x} = Ax + Bu \qquad \textbf{(2.2.18)}$$

$$A = \begin{bmatrix} -3.2200E-01 & 6.4032E-02 & 3.8904E-02 & -9.9156E-01 \\ 0.0000E+00 & 0.0000E+00 & 1.0000E+00 & 3.9385E-02 \\ -3.0919E+01 & 0.0000E+00 & -3.6730E+00 & 6.7425E-01 \\ 9.4724E+00 & 0.0000E+00 & -2.6358E-02 & -4.9849E-01 \end{bmatrix} \qquad \textbf{(2.2.19)}$$

Table 2.2.2: The lateral subsystem characteristics

Description	Value	Mode description	Mode value
roll rate p and yaw rate r	-0,4399±j3.220	dutch roll mode	T=1.95 s, δ=0.135
roll rate p	-3.601	roll subsidence mode	τ=0.28 s
roll angle ϕ	-0.0128	spiral mode	τ=77.9

The two subsystems of the system dynamics were derived. To sum up, in this stage the separate longitudinal and lateral plant models were obtained, and the various mathematical models such as roll to aileron, yaw rudder and pitch elevator relationships.

2.3 The lateral-directional equations of a fixed-wing UAV

From the state matrices below, the lateral-directional equations can be grouped from the full state set, the coefficient matrices are considered as :

$$A = \begin{bmatrix} -0.13150 & 0.1458 & 0.0 & 0.32434 & -0.93964 \\ 0.0 & 0.0 & 0.0 & 1.0 & 0.33976 \\ 0.0 & 0.0 & 0.0 & 0.0 & 1.0561 \\ -10.614 & 0.0 & 0.0 & -1.1793 & 1.0023 \\ 0.99655 & 0.0 & 0.0 & -0.0018174 & -0.25855 \end{bmatrix} \quad \textit{(2.3.1)}$$

$$B = \begin{bmatrix} \delta_a & \delta_r \\ 0.00012049 & 0.00032897 \\ 0.0 & 0.0 \\ 0.0 & 0.0 \\ -0.1031578 & 0.020987 \\ -0.0021330 & -0.010715 \end{bmatrix} \quad \textit{(2.3.2)}$$

$$C = \begin{bmatrix} 0.0 & 0.0 & 57.29578 & 0.0 \\ 0.0 & 0.0 & 0.0 & 57.29578 \end{bmatrix} \quad \textit{(2.3.3)}$$

$$D = \begin{bmatrix} 0 & 0 \\ 0 & 0 \end{bmatrix} \quad \textit{(2.3.4)}$$

Since this is the MIMO (Multi-Input-Multi-output). the system, the regular "**ss2tf**" function doesn't apply to this case. Instead, the usage of the ss function in MATLAB can help to retrieve the corresponding numerator and denominator values of the 4-transfer function.

The null column in the A-matrix shows that the state ψ (3rd column)is not coupled back to any other states, and it can be omitted from the state equations when designing an augmentation system. The C-matrix has been used to convert the output quantities to degrees, to match the control surface inputs. The transfer functions of primary interest are

$$\frac{p}{\delta_a} = \frac{-5.911s^3 - 2.393s^2 - 6.952s + 0.3628}{s^4 + 1.551s^3 + 4.848s^2 + 3.25s + 0.199} \quad \textit{(2.3.5)}$$

$$\frac{r}{\delta_a} = \frac{-0.1222s^3 - 0.1403s^2 - 2.338s - 1.068}{s^4 + 1.551s^3 + 4.848s^2 + 3.25s + 0.199} \quad \textit{(2.3.6)}$$

$$\frac{p}{\delta_r} = \frac{1.202s^3 - 0.3684s^2 - 5.064s + 0.2685}{s^4 + 1.551s^3 + 4.848s^2 + 3.25s + 0.199} \quad \textbf{(2.3.7)}$$

$$\frac{r}{\delta_r} = \frac{-0.6139s^3 - 0.7769 - 1.784s - 0.7901}{s^4 + 1.551s^3 + 4.848s^2 + 3.25s + 0.199} \quad \textbf{(2.3.8)}$$

Corresponding poles of the characteristic equation given below are:

$$Q(\lambda) = \lambda^4 + 1.551\lambda^3 + 4.848\lambda^2 + 3.25\lambda + 0.199 \quad \textbf{(2.3.9)}$$

$$Z_G = \begin{bmatrix} -0.3930 + 2.0113i \\ -0.3930 - 2.0113i \\ -0.6970 + 0.0000i \\ -0.0680 + 0.0000i \end{bmatrix}$$

(2.3.10)

It is clear the system behavior is stable since the poles have a negative real part. Step response results for pitch rate to aileron position, roll rate to aileron position, pitch rate to rudder position, and roll rate to rudder position.

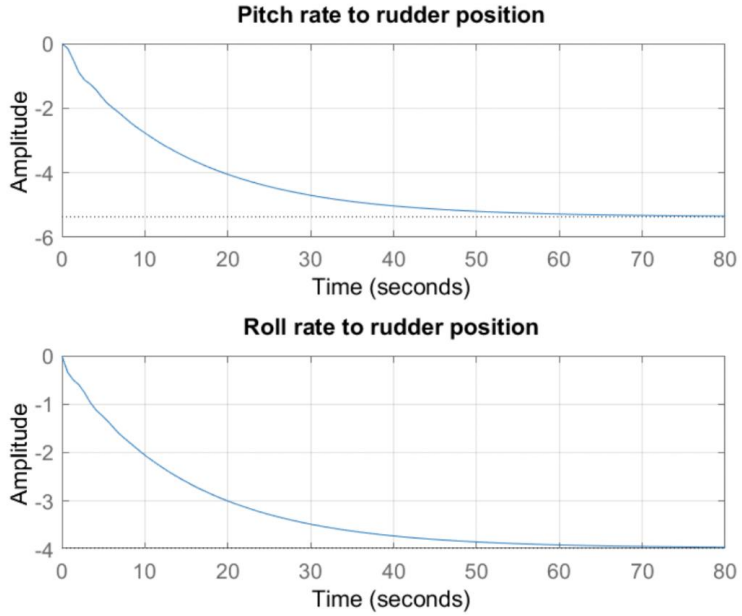

Figure 2.3.1: Step response of pitch and roll rates with respect to the rudder position

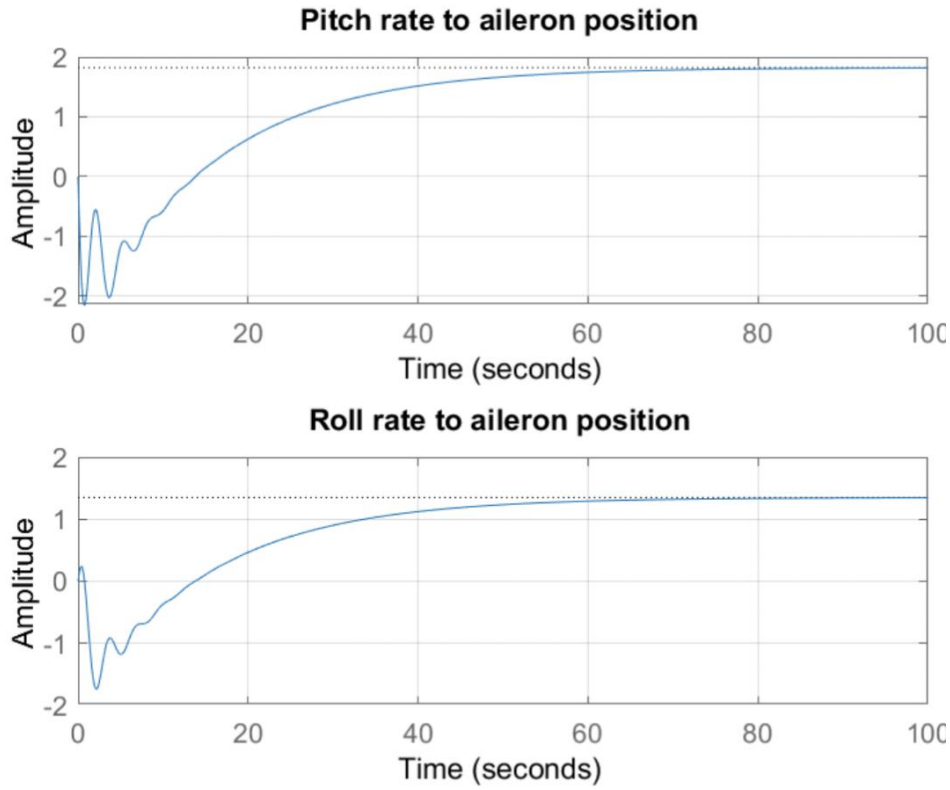

Figure 2.3.2: Step response of pitch and roll rates with respect to the aileron position

It is obvious from the figures above that the system behavior is stable since the after 50 seconds the amplitude of a roll and pitch to aileron position tends to the desired reference value.

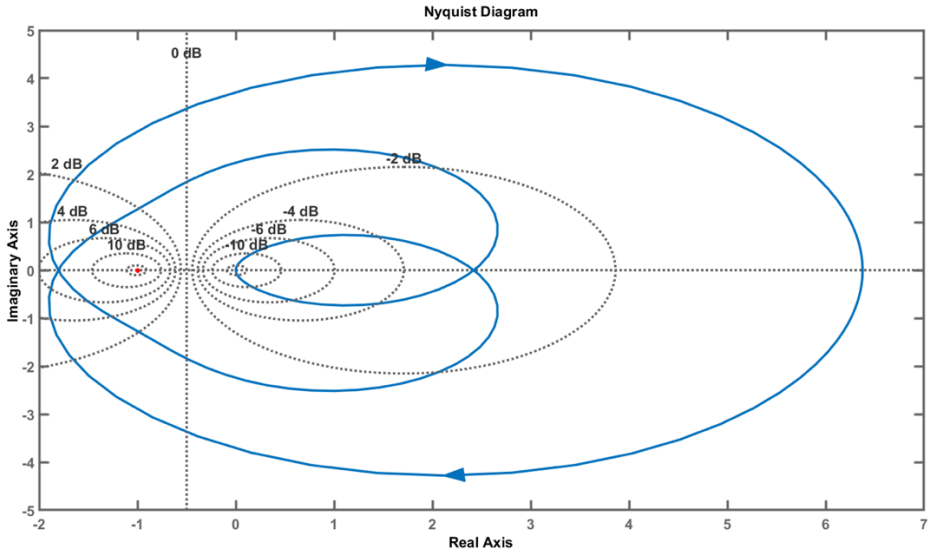

Figure 2.3.3: Nyquist plot of an open-loop system

From the Nyquist plot, it can be seen that the number of encirclements is equal to 2, however open-loop system does not have a positive poles. That is why, the root locus plot was introduced to access the location of the poles and zeros in s-plane. If we analyze the root locus plot, we can observe that there are unstable zeros, which causes a negative transient response at the beginning of the process.

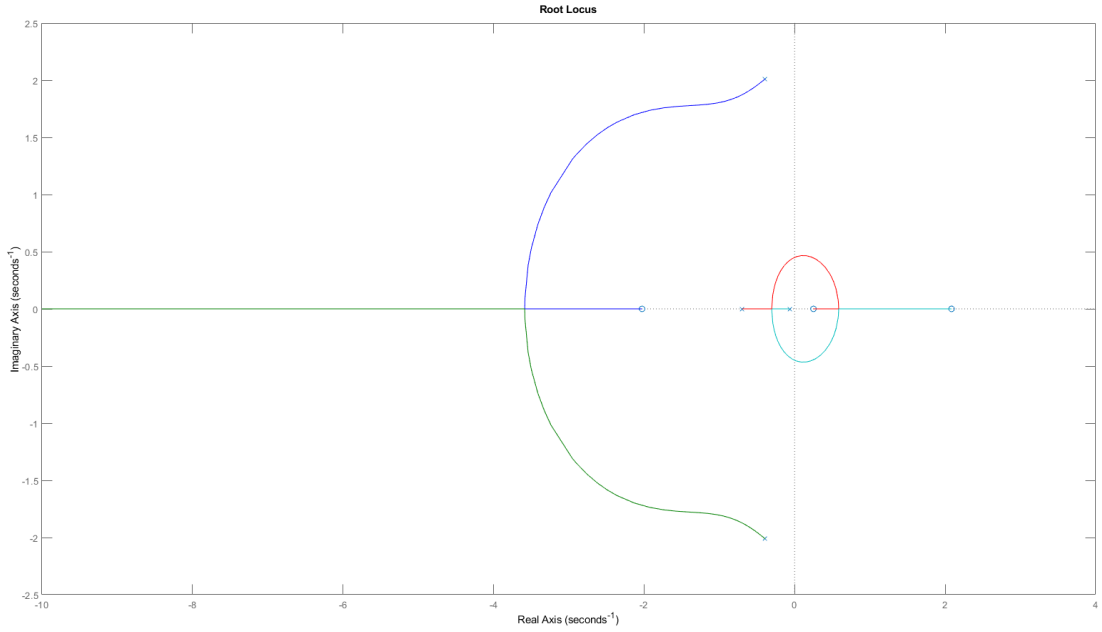

Figure 2.3.4: Root-locus of the open-loop system

From the root locus plot shown above, it is observable that all of the poles of an open-loop system are on the Left-hand plane, however there are two zeros which are on the right-hand plane. From now on, the closed-loop block diagram and the results of the F-16 model roll-to-aileron angle control will be shown.

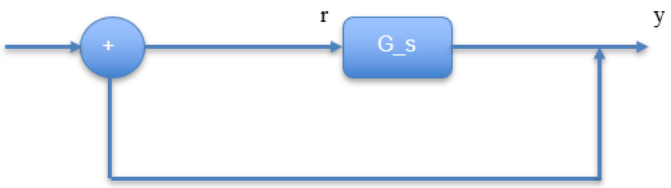

Figure 2.3.5: Closed loop block diagram

$$G_{closed}(s) = \frac{(1.202s^3 - 0.3684s^2 - 5.064s + 1.268)}{(s^4 + 2.753s^3 + 4.48s^2 - 1.814s + 1.468)} \quad \textbf{(2.3.11)}$$

If we analyze the root locus plot of this transfer function, we notice that even the poles are located on the right-hand plane, since the characteristic equation of $G_{closed}(s)$

$$Q(\lambda) = \lambda^4 + 2.753\lambda^3 + 4.848\lambda^2 - 1.814\lambda + 1.468 \quad \textbf{(2.3.12)}$$

has a different signed coefficients which leads to the unstable poles of:

- $z_1 = -1.6007 + 1.7677\text{i}$
- $z_2 = -1.6007 - 1.7677i$
- $z_3 = 0.2312 + 0.4512\text{i}$
- $z_4 = 0.2312 - 0.4512i$

$z_3$and z_4 are positive poles causing the instability of the system, where the step response of the closed-loop system without controller and root locus are shown below.

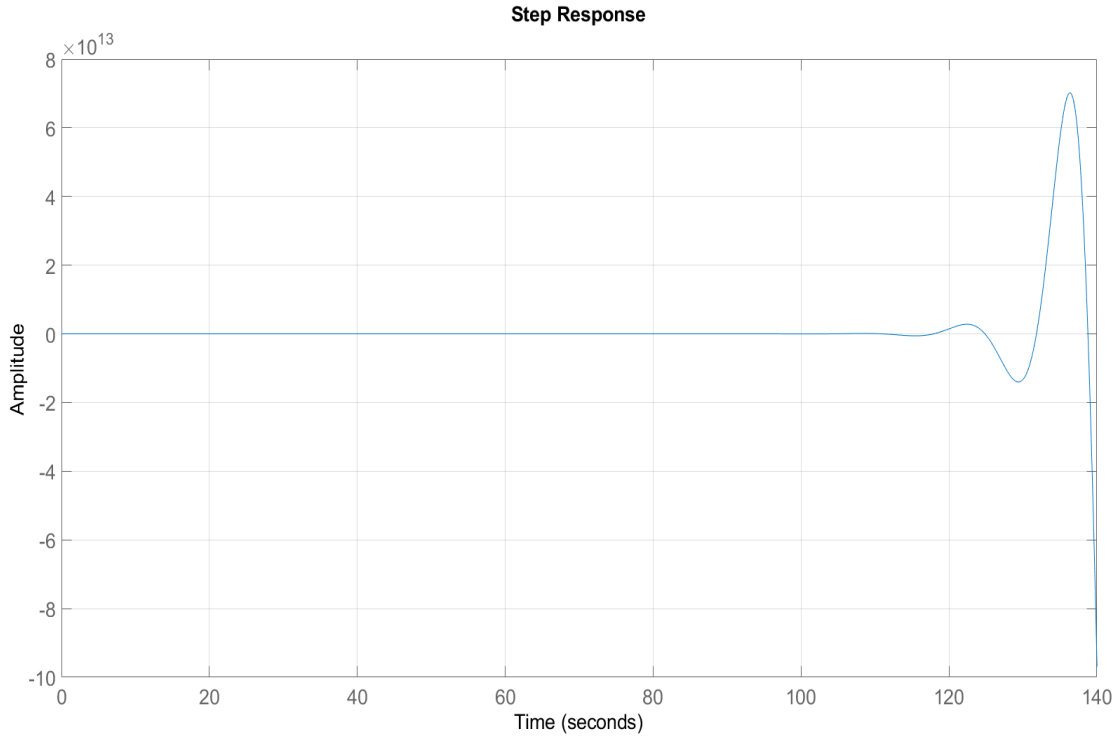

Figure 2.3.6: Step response of the closed-loop system without a controller

As we can observe, the step response is unstable since the amplitude of the transfer function will not converge to the set point as time approaches to infinity. Corresponding root locus plot is also shown below, where the “x” crossing are denoted as system poles, whereas the “o” aare denoted as system zeros. Thus it is clear from the figure that the 2 poles and 2 zeros are located on the right-hand plane, causing the instability of the system (positive poles makes system unstable), thus the feedback gain or PID controller needs to be designed to relocate the poles to the left-hand plane.

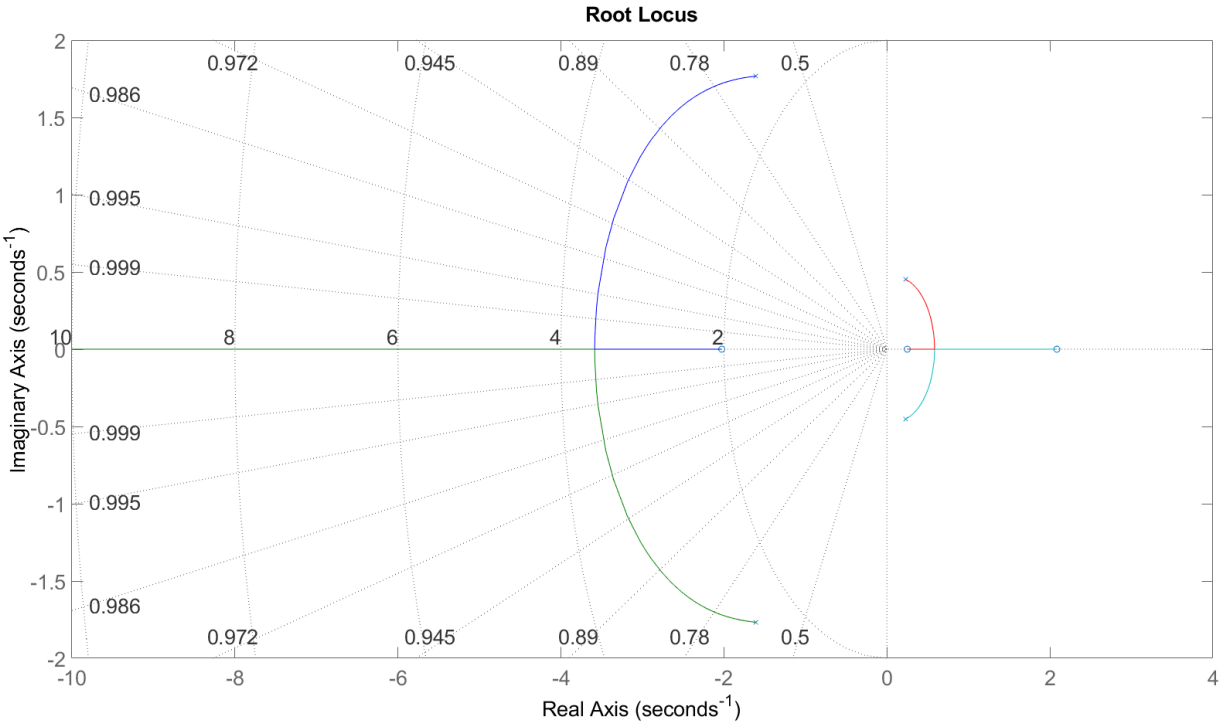

Figure 2.3.7: Root-locus plot of a closed-loop system without controller

Chapter 3- Closed-loop control of a F-16 fixed-wing UAV using PID controller

3.1 PID controllers and transfer function

From the previous section, it was obvious from controllability matrix that the system is controllable, thus the necessary controller needs to be designed to stabilize the system.

For this section, the PID (Proportional-Integrator-Derivative) controller will be designed in the following order:

- Closed-loop control with the proportional gain K. P controller
- Closed-loop control with the proportional and integrator controller. PI controller
- Closed-loop control with the proportional, integrator and derivative controller. PID controller

3.2 Designing the PID controllers with AutoTuner MATLAB

General block diagram of a closed-loop control is shown below:

Figure 3.2.1 : Block diagram of a closed-loop control of a roll angle transfer function using PID controller

After designing the Simulink block diagram and the PID auto tuner as well as with the MATLAB script the following results were achieved.

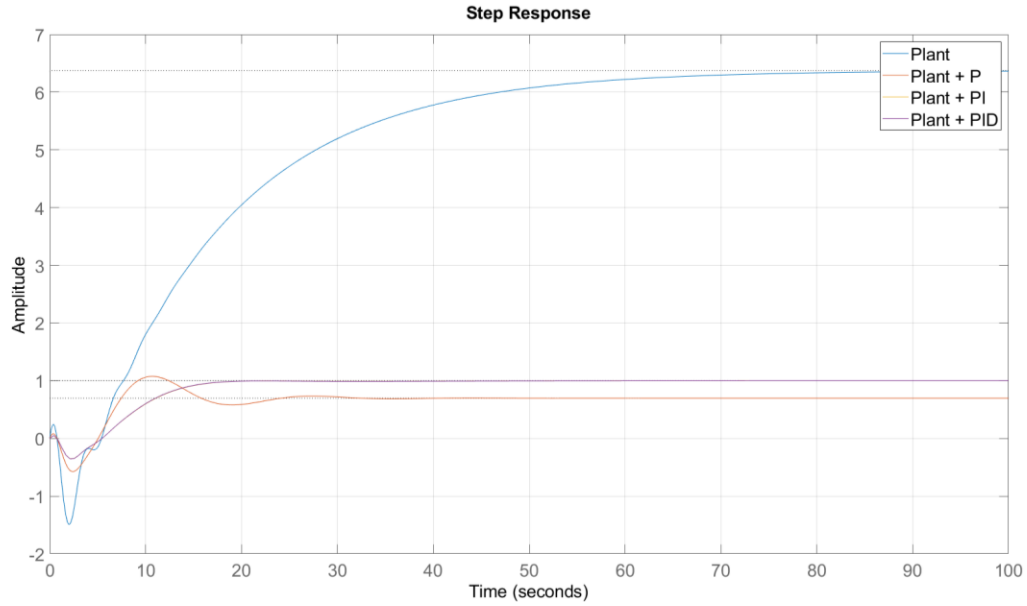

Figure 3.2.2: Tuning PID controllers to stabilize the open-loop plant

From the figure above, a label "Plant" means the open-loop step response. It's clear that the PI and PID controllers overlaps each other, denoting that they have a common coefficients P and I. From now on, we will consider the PI controller with K_P and K_i equal to $K_p = 0.223, K_i = 0.0143$. If we look closer to the PI controller response shown below, we observe that it stabilizes the system with the following parameters shown in Table 3.2.1.

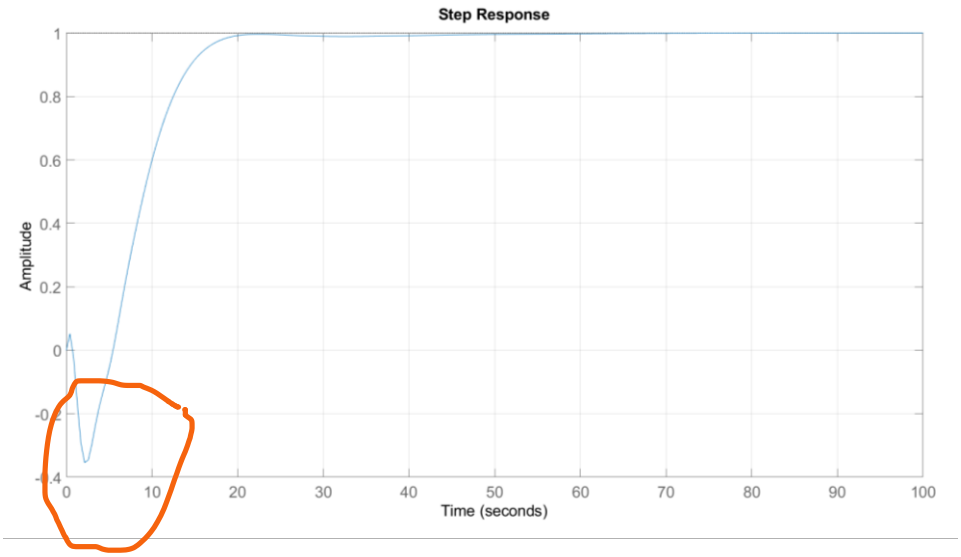

Figure 3.2.3: PI controller response

Table 3.2.1: PI controller step response info

Parameter name	Value
Rise Time	8.3899
Settling Time	17.6106
Settling Min	0.9009
Settling Max	0.9009
Overshoot	0
Undershoot	35.7232
Peak	0.9962
Peak Time	22.4702

Corresponding Nyquist plot and root-locus diagrams are shown in Figure 3.2.4 and Figure 3.2.5 respectively.

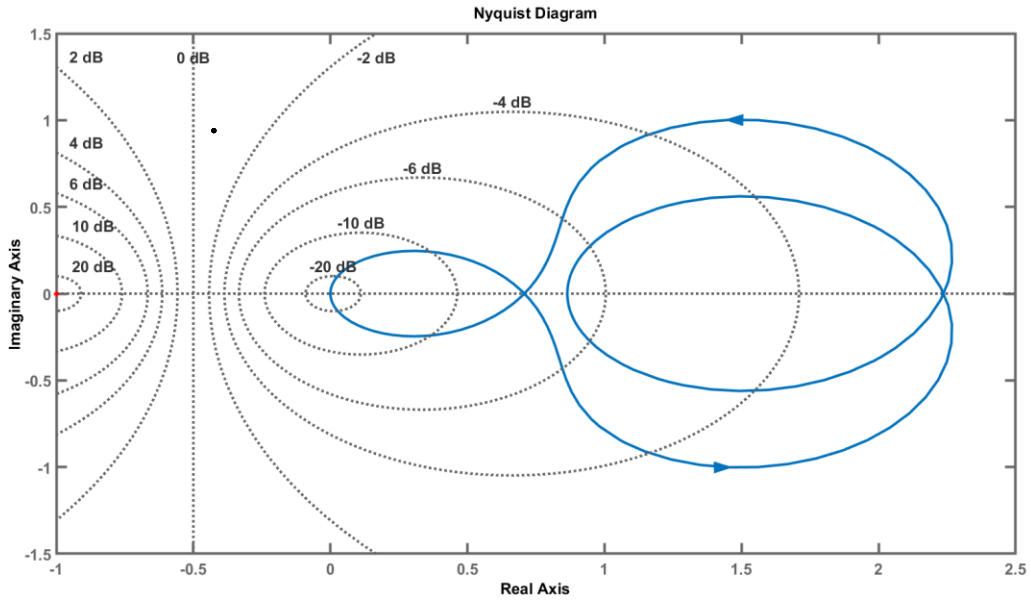

Figure 3.2.4: Nyquist plot of the closed-loop control system using PI controller

After applying a PI controller to the system, a new closed-loop system has no poles in RHP, that is why in the Figure above, there is no encirclements around -1.

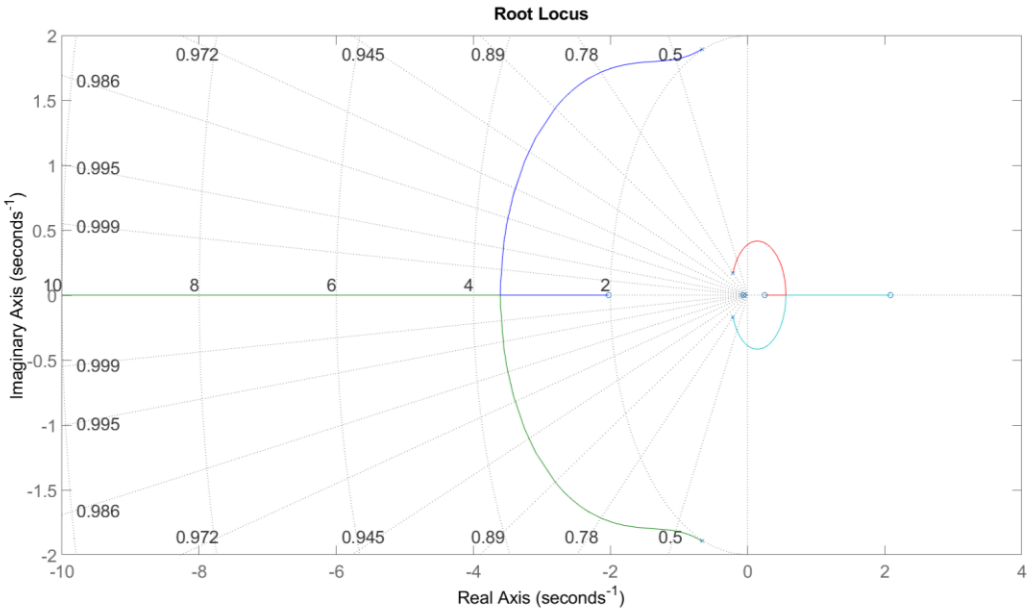

Figure 3.2.5: Root-locus plot of the closed-loop control system transfer function

From the step response figure shown above, a highlighted part which shows the negative transient for the first 5 seconds is a proof of an existing of positive zeros. This behavior is the distinct representation of a non-minimum phase system.

Chapter 4-Feed-forward compensation method for eliminating the non-minimum phase zeros

4.1 Non-minimum phase systems

In the Linear Control System theory, the investigation of the transfer function and its root-locus, step response behaviors can face the issue with positive zeros or in other words RHP-zeros (right-hand plane). It has some constraints under feedback control, given that some zeros in the right-hand plane. Choosing the limited and an arbitrary high feedback gain in closed-loop control of a system is one of the distinctive limitations. Since the increase of a feedback gain under the allowable range can lead to the positive poles thus cause the instability of the closed-loop system.

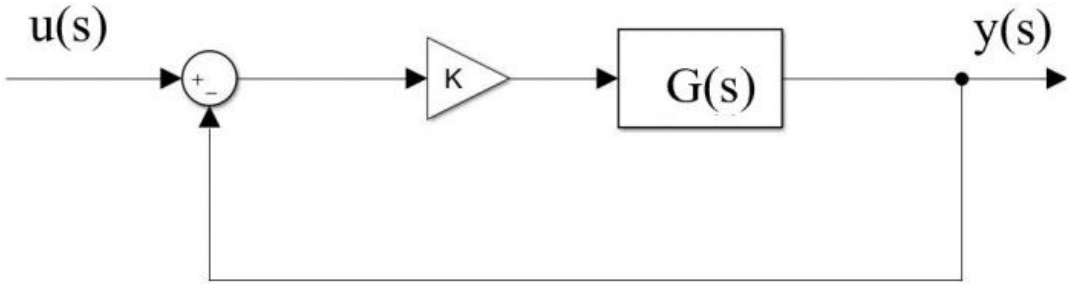

Figure 4.1.1: Closed-loop block diagram with a feedback gain K [31]

For a open-loop transfer function

$$G(s) = \frac{a_0 s^n + a_1 s^{n-1} + a_2 s^{n-2} + \cdots + a_n}{b_0 s^m + b_1 s^{m-1} + b_2 s^{m-2} + \cdots + b_m} \qquad \textbf{(4.1.1)}$$

Where n<=m, with a feedback gain K, a several conclusions can be drawn

- Open-loop transfer function G(s) can be a stable having all poles in LHP
- Closed-loop transfer function $G_{cloop}(s) = \frac{K*G(s)}{1+K*G(s)}$ can be a stable system under the allowable set of values
- However, it might have a positive pole which lead to an unstable system

- Despite that having a positive zero within the poles creates the non-minimum phase unstable system

Moreover, it has peculiar limitations in which one of them is the case for our system.

1. Undershoot issue

Depending on the number of positive zeros (namely the odd number of zeros) in RHP, the step response behavior of a system will vary its direction before settling to a reference point. In fact, that the number of times when the step response plot changes its direction directly depends on the number of odd zeros located in the right-hand plane.

2. Zero-crossings

According to the undershoot issue, it's retrieved that the phenomena depend on the odd-number zeros. Unlike that the zero crossings happen when we are dealing with the even-number zeros, such that the process plant response plot (behavior) will change its direction resulting in zero crossings.

A set of works have been conducted in reducing the non-minimum phase system and the main goal was to relocate the RHP-zeros to the LHP (Left-hand plane). Following are the classical and novel approaches to eliminate the NMP systems:

- Pole-zero cancellation
- Feed-forward compensation method (FFC)
- Parallel feed-forward compensator and derivative method (PFCD)

In this section three pointed methods will be applied and verified for a stability and other metrics, such as overshoot/undershoot, settling time and rise time.

4.2 Pole-zero cancellation

Pole-zero cancellation method often used in the case, when the transfer function of a plant is simplified and expressed in terms of poles and zeros, thus can be cancelled in case of match. This process facilitates and simplify the transfer function and, in some cases, the RHP zeros can also be eliminated in this case. The closed-loop transfer function of roll-to-aileron angle using PID controller can be written as

$$G(s) = \frac{0.2675\ s\text{^}4 - 0.06481\ s\text{^}3 - 1.132\ s\text{^}2 + 0.21\ s + 0.01812}{s\text{^}5 + 1.818\ s\text{^}4 + 4.783\ s\text{^}3 + 2.118\ s\text{^}2 + 0.409\ s + 0.01812} \qquad \textbf{\textit{(4.2.1)}}$$

It is straightforward that it cannot be eliminated and poles and zeros will not compensate each other since they have various values shown below:

Stable real and complex conjugate poles

- $p_1 = -0.6636 + 1.8849i$
- $p_2 = -0.6636 - 1.8849i$
- $p_3 = -0.2152 - 0.1656i$
- $p_4 = -0.2152 + 0.1656i$
- $p_5 = -0.061$

Unstable real zeros

- $z_1 = 2.0846$
- $z_2 = -2.0278$
- $z_3 = 0.2497$
- $z_4 = -0.0642$

One of the main disadvantages of this method that it cannot handle with the non-minimum phase and unstable and the systems which have complex conjugate poles. Since the transfer

function of the roll-to-aileron control system have complex conjugate poles, this method will not work.

4.3 Feed-forward compensation method

Authors in [31] proposed the updated method of feed-forward compensation where the key point is to decompose a plant into two subsystems of the stable but non-minimum phase and unstable system (with positive poles) with minimum phase subsystem designing the gain L.

Given G(s) – open-loop transfer function of the system with the following parts:

- R(s) – transfer function reference input signal. This can be a simple step signal, sine wave or even a complex one
- Y(s) – transfer function of the output signal
- E(s) – Transfer function of the error signal which can be calculated using the difference between the reference and output signal
- C(s) – Transfer function of the controller. This can be a simple feedback gain K, PID in our case.
- Gc(s) – transfer function of the parallel compensator
- G(s) – transfer function of the plant
- Yp(s) – transfer function of the output plant
- Yc(s) – transfer function of the output of parallel compensator

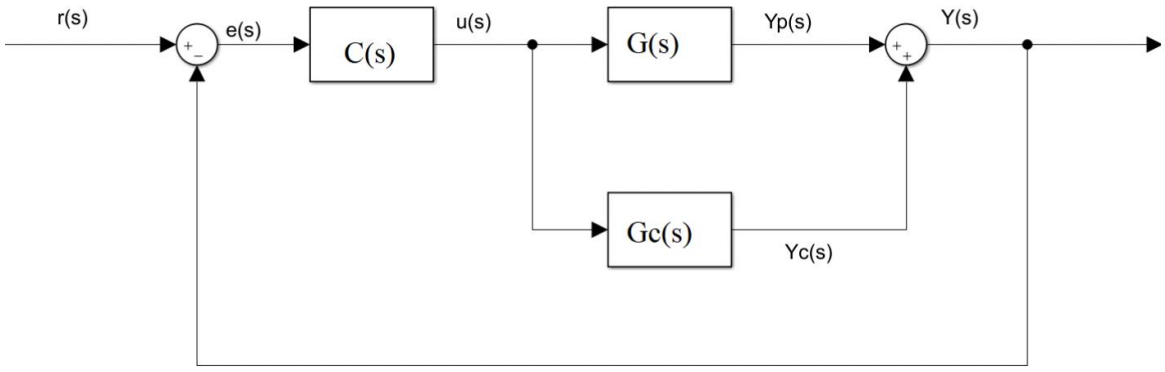

Figure 4.3.1: Block diagram of a feed-forward compensator controller [31]

The transfer function of the plant is denoted by G(s) can be written as

$$G(s) = \frac{Y_p(s)}{u(s)} \qquad \textbf{(4.3.1)}$$

$$G_c(s) = \frac{Y_c(s)}{u(s)} = G_1(s) - G(s) \qquad \textbf{(4.3.2)}$$

$$G_r(s) = \frac{Y(s)}{u(s)} = G(s) + G_c(s) = G(s) + G_1(s) - G(s) = G_1(s) \qquad \textbf{(4.3.3)}$$

Where $G_1(s)$ is the transfer function that should be selected carefully. The choice of is determined by the application. The precision of the consistent steady state is a worry for us. Because of the restricted feedback gain, which is dependent on the placement of the plant's zeros and poles, we have a limited ability to shape transient response in non-minimum phase systems.

Therefore, the decomposition of the open-loop transfer function of the plant should be applied to the system shown below.

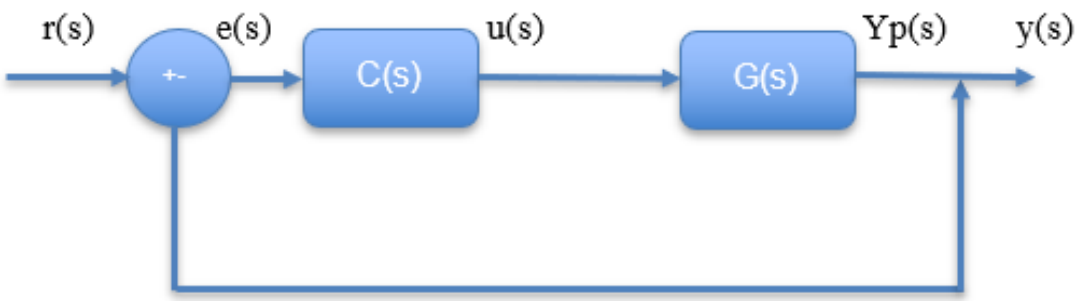

Figure 4.3.2: Block diagram of the cascaded control system

Where

- G(s) = G1(s)*G2(s)
- G1(s) = MP (Minimum phase) and stable/unstable
- G2(s) = NMP (Non-minimum phase) and stable and unstable

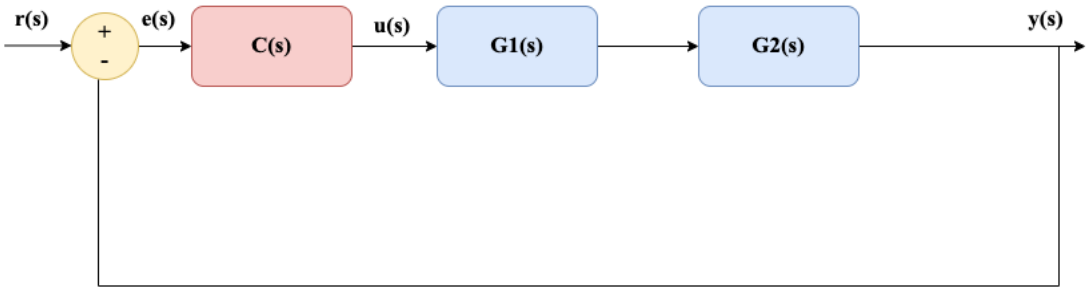

Figure 4.3.3: Simplified block diagram of decomposed control system

Assuming the $G(s)$ is the transfer function with n zeros and m poles, where $m \geq n$. The major step in this method is to divide the G(s) into two subsystems G1(s) and G2(s). Let's denote the denominator of the transfer function as D(s) and N(s) as the numerator, with the following relationship:

$$G(s) = \frac{N(s)}{D(s)} \qquad \textbf{(4.3.4)}$$

$$N(s) = (s - z_1)(s - z_2)(s - z_3) \dots (s - z_n) \qquad \textbf{(4.3.5)}$$

$$D(s) = (s - p_1)(s - p_2)(s - p_3) \dots (s - p_m) \qquad \textbf{(4.3.6)}$$

Since this proposed method works for both stable/unstable with non-minimum phase system, the further calculations will be made with the non-minimum phase stable system with complex conjugate real roots. The decomposition and selection of G1(s) and G2(s) process is the following:

- $G_1(s)$ should be selected in a such way that the denominator should have a stable zero (which means minimum phase) and stable poles

- $G_2(s)$should be selected in a such way that the denominator should have unstable zero (which means non-minimum phase) and stable poles

$$G_1(s) = \frac{(s+ z_{w+1})(s+ z_{w+2})(s+ z_{w+2})\dots(s+ z_n)}{(s+ p_1)(s+ p_2)(s+ p_3)\dots(s+ p_{m-w+1})} \quad \textbf{(4.3.7)}$$

$$G_2(s) = \frac{(s- z_1)(s- z_2)(s- z_3)\dots(s- z_w)}{(s+ p_{m-w+2})(s+ p_{m-w+3})(s+ p_{m-w+4})\dots(s+ p_m)} \quad \textbf{(4.3.8)}$$

Our transfer function is in the form

$$G(s) = \frac{1.202s^3-0.3684s^2-5.064s+1.268}{s^4+1.551s^3+4.848s^2+3.25s+0.199} \quad \textbf{(4.3.9)}$$

Where it has 2 complex conjugate pairs

$p_1 = -0.393 + 2.0112i$

$p_2 = -0.393 - 2.0112i$

and 2 positive zeros $z_1 = 2.085$ and $z_2 = 0.2499$

This is the case when the system is non-minimum phase and stable with the complex conjugate pairs. In fact that the complex poles exits in pairs, there is an additional, modified approach beside the what was discussed earlier.

In case of complex conjugate pairs, an additional set of stable left-hand poles/zeros factors will be multiplied to both numerator and denominator. Furthermore, the decomposition process will be performed based on this new transfer function. In the transfer function G(s), the number of poles is m, and there are m/2 complex conjugate pairs. In total it has k (out of n maximum zeros) non-minimum phase zeros. In total there are k non-minimum phase and (n - k) minimum phase zeros, such that

$$G(s) = \frac{(s + z_1)(s + z_2)\dots(s + z_{n-k})(s + z_k)}{(s \pm p_1)(s \pm p_2)\dots(s \pm p_{m/2})} \quad \textbf{(4.3.10)}$$

The next procedure is to apply stable pole-zeros to the plant. In fact that the number of pole-zero pairs is directly dependent on the number of k non-minimum phase zeros. In our case this is equal to k = 2.

From now on, the resultant transfer function after applying stable pole-zeros to the plant is can be denoted by $G_P(s)$, while the transfer function of stable pole-zeros is denoted by $G_C(s)$, in the following dependence.

$$G_c(s) = \frac{(s+a_1)(s+a_2)\ldots(s+a_m)}{(s+b_1)(s+b_2)\ldots(s+b_m)} \qquad \textbf{(4.3.11)}$$

$$G_p(s) = \frac{(s+z_1)(s+z_2)\ldots(s+z_{n-k})(s+z_k)(s+a_1)(s+a_2)\ldots(s+a_m)}{(s\pm p_1)(s\pm p_2)\ldots(s\pm p_{m/2})(s+b_1)(s+b_2)\ldots(s+b_m)} \qquad \textbf{(4.3.12)}$$

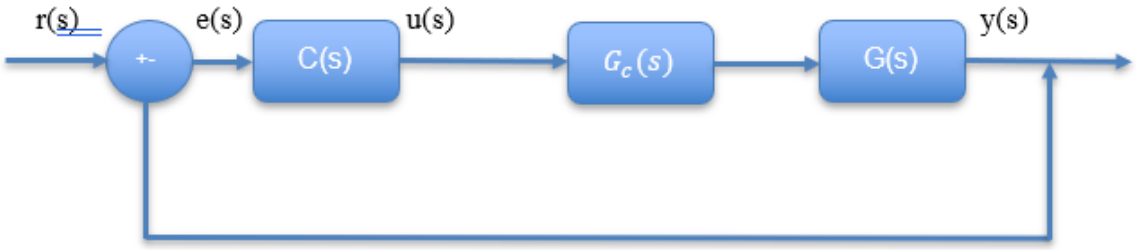

Figure 4.3.4: The block diagram of the cascaded pole-zero system

Furthermore the $G_p(s)$ is decomposed into $G_1(s)$,which minimum phase (stable zeros) and stable/unstable system, and $G_2(s)$,non-minimum phase and stable.

4.4 Tuning the parameters a,b and L

For a moment, the transfer functions $G_c(s)$ and $G_p(s)$ are modelled as

$$G_c(s) = \frac{(s+a_1)(s+a_2)}{(s+b_1)(s+b_2)} \qquad \textbf{(4.4.1)}$$

$$G(s) = \frac{1.202(s+2.208)(s-2.085)(s-0.2499)}{(s+0.06798)(s+0.6971)(s^2+0.7859s+4.199)} \qquad \textbf{(4.4.2)}$$

$$G_p(s) = \frac{1.202(s+2.208)(s-2.085)(s-0.2499)(s+a_1)(s+a_2)}{(s+0.06798)(s+0.6971)(s^2+0.7859s+4.199)(s+b_1)(s+b_2)} \quad \textbf{(4.4.3)}$$

Choosing the coefficients a_1, a_2 and b_1, b_2 is arbitrary and then can be tuned by MATLAB script. For now, let's take

$$a_1 = 5, a_2 = 1$$

$$b_1 = 6, b_2 = 10$$

$$G_p(s) = \frac{1.202(s+2.208)(s-2.085)(s-0.2499)(s+5)(s+1)}{(s+0.06798)(s+0.6971)(s^2+0.7859s+4.199)(s+6)(s+10)} \quad \textbf{(4.4.4)}$$

And decomposed $G_1(s)$ and $G_2(s)$

$$G_1(s) = \frac{1.202(s+2.208)(s+5)(s+1)}{s^2+0.7859s+4.199} \quad \textbf{(4.4.5)}$$

$$G_2(s) = \frac{(s-2.085)(s-0.2499)}{(s+0.06798)(s+0.6971)(s+6)(s+10)} \quad \textbf{(4.4.6)}$$

$$\hat{G}_2(s) = \frac{(s-2.085)(s-0.2499)}{(s+0.06798)(s+0.6971)(s+6)(s+10)} + L \quad \textbf{(4.4.7)}$$

Choosing L is also an arbitrary selection depending on the purpose. Let's for a moment take L = 1, which means that from unstable 2.085 and 0.2499 it can be shifted to -9.626, -6.438, -0.6126 and -0.08862.

$$\hat{G}_2(s) = \frac{(s+9.626)(s+6.438)(s+0.6126)(s+0.08862)}{(s+0.06798)(s+0.6971)(s+6)(s+10)} \quad \textbf{(4.4.8)}$$

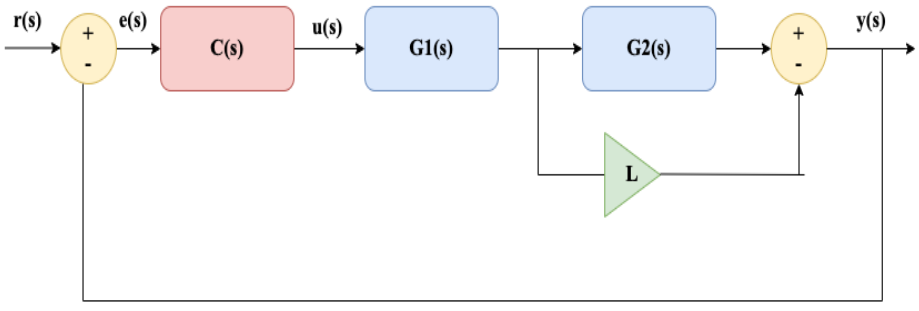

Figure 4.4.1: General block diagram of a feed-forward compensation method for a non-minimum phase stable system with complex conjugate pole pairs

Corresponding Simulink diagram is shown in figure below:

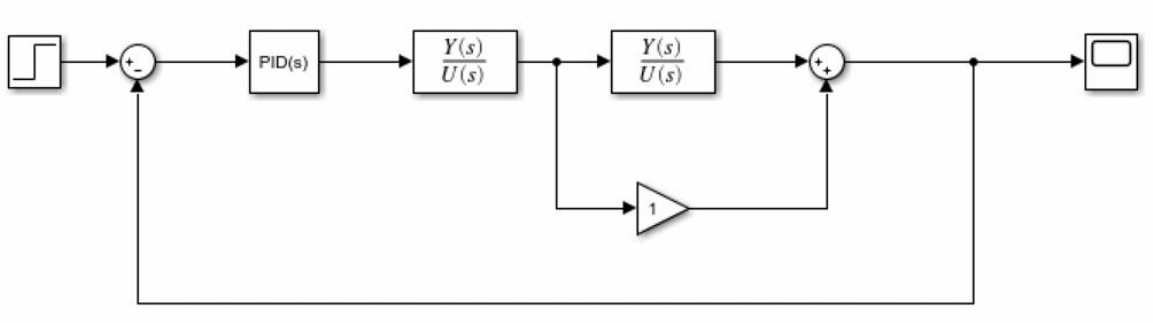

Figure 4.4.2: Simulink block diagram of a feed-forward compensation method for a non-minimum phase stable system with complex conjugate pole pairs

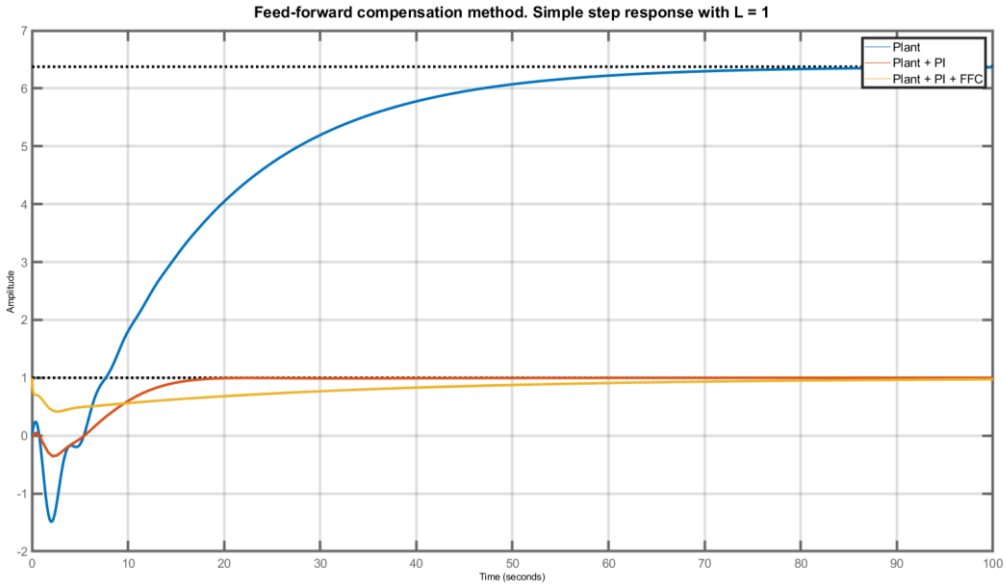

Figure 4.4.3: Feed-forward compensation method. Simple reference: step response with L = 1

As we can observe, the initial undershoot to the negative value is eliminated and the system is stabilized and tracked the reference compared to the PI controller case. However, the settling time in this case was around 126 seconds, which is not reasonable for a fixed-wing UAV to stabilize it. In order to investigate deeper, the closed-loop control of fixed-wing UAV using PI controller and feed-forward compensation method with a sine wave response result is shown in figure 4.4.4.

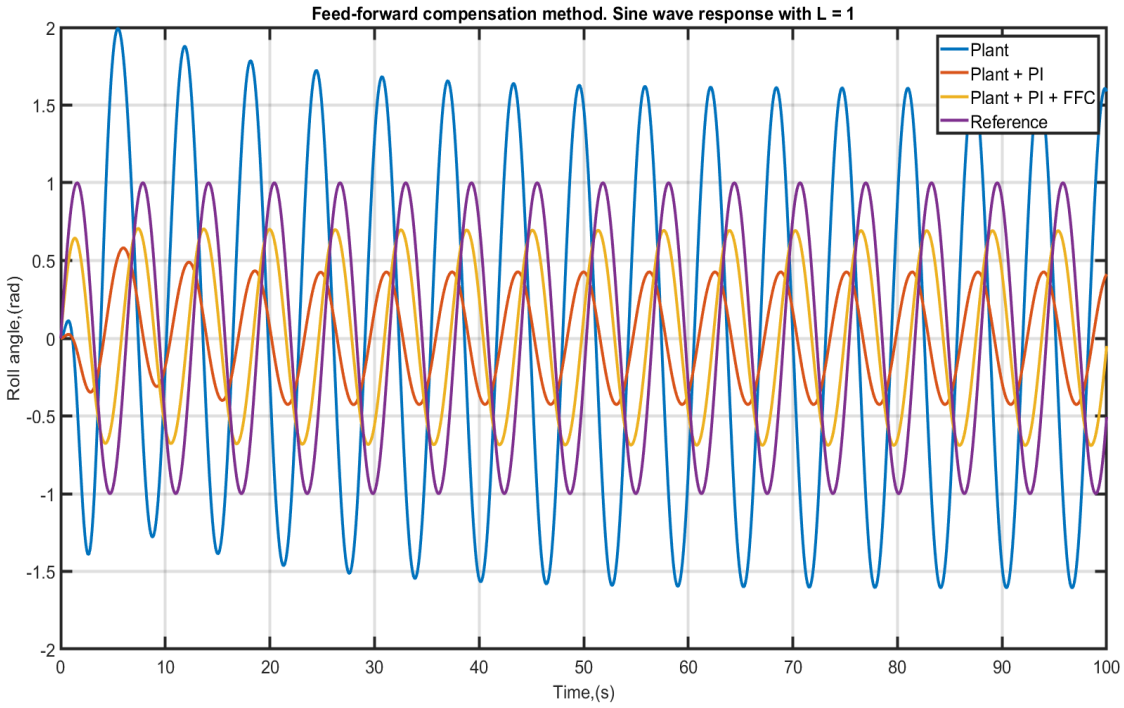

Figure 4.4.4: Feed-forward compensation method. Sine wave response with L = 1

It is evidently clear, that the response of a design with the feed-forward compensator method has the same phase with the original sine wave, while the PI controller has a phase shift of approximately 5 seconds. However, the issue with an amplitude exists in this case. From now, on the parameter L will be tuned in the range of 1 to 100, satisfying a condition and tracking the both a simple step response and complex reference (sine wave, square wave and their cumulation).

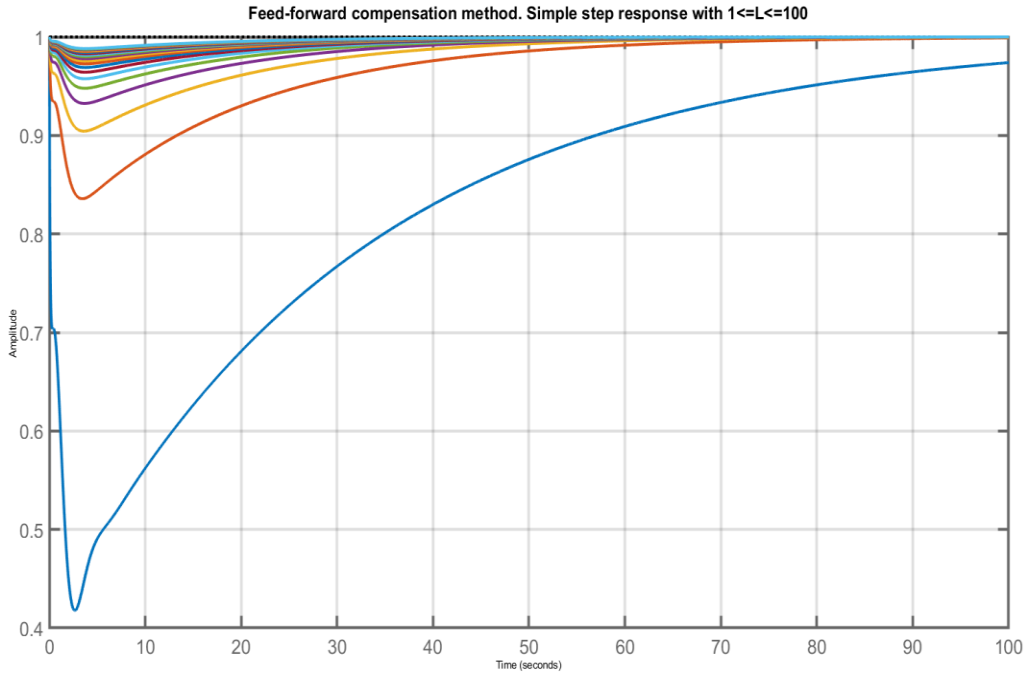

Figure 4.4.5: Feed-forward compensation method. Simple reference: Step response with $1 \leq L \leq 100$, with step 5

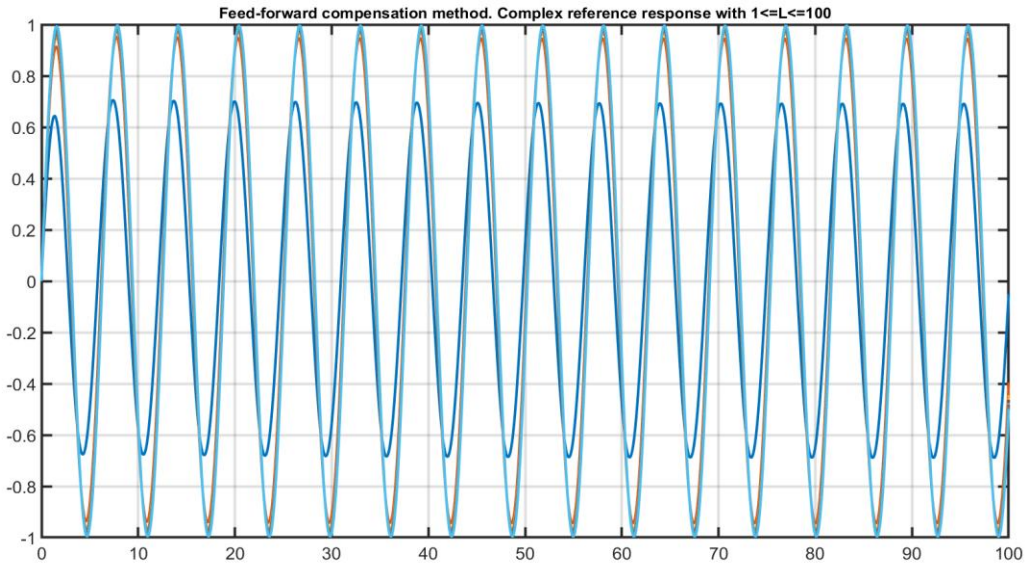

Figure 4.4.6: Feed-forward compensation method. Complex reference: Sine wave response with $1 \leq L \leq 100$, with step 5

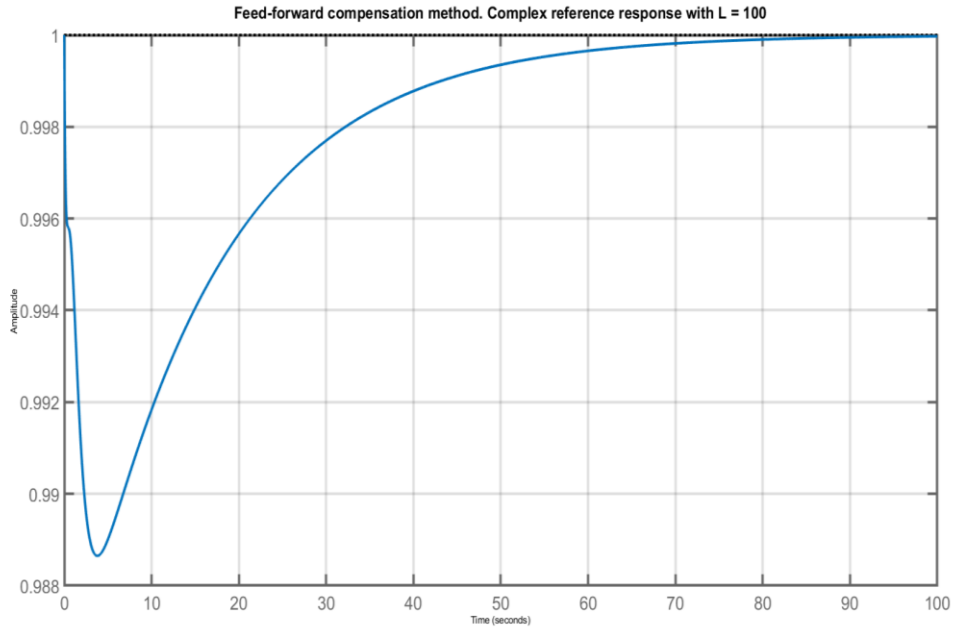

Figure 4.4.7: Feed-forward compensation method. Simple reference: Step response with $L = 100$

From the parameter tuning process, the **L = 100** was the optimal value of a coefficient L. As well as the undershoot problem at the initial phase was resolved and the whole step response information is provided in the Table 4.4.1 below:

Table 4.4.1: Step response information of a closed-loop model using [31]

Parameter name	Value
Rise Time	2.9875e-13
Settling Time	66.5522
Settling Min	0.9886
Settling Max	1.0000
Overshoot	1.3101e-12
Undershoot	0
Peak	1
Peak Time	0

Then it was tested for more complex references such as sine wave, square and combined complex reference responses.

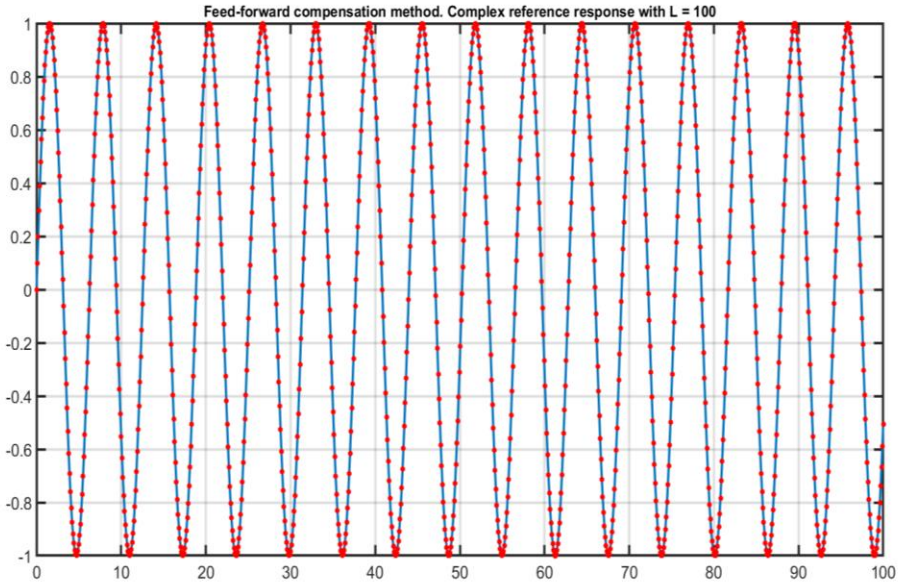

Figure 4.4.8: Feed-forward compensation method. Complex reference: Sine wave response with $L = 100$

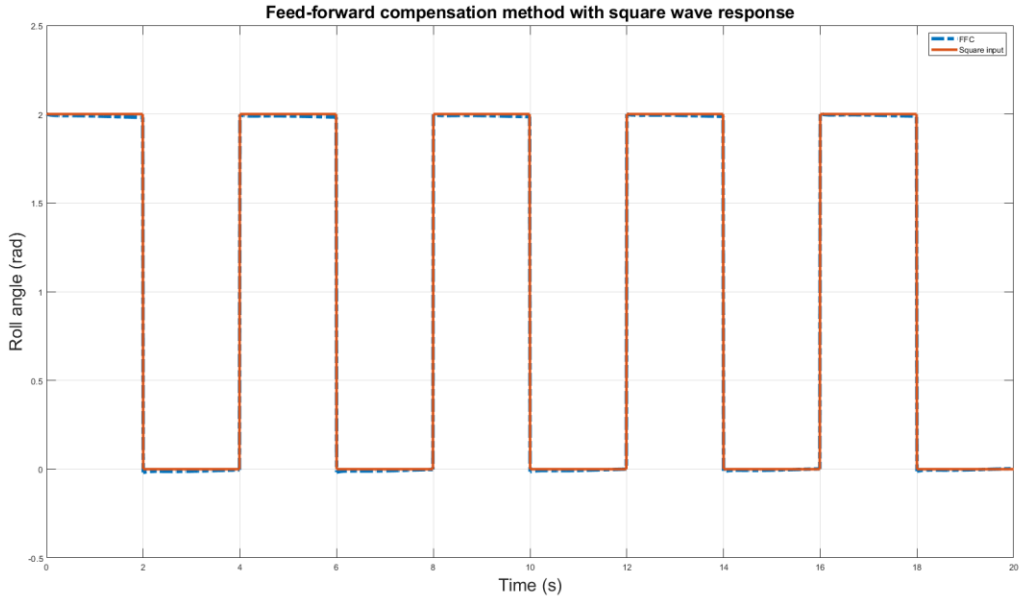

Figure 4.4.9: Feed-forward compensation method. Complex reference: Square wave response with $L = 100$

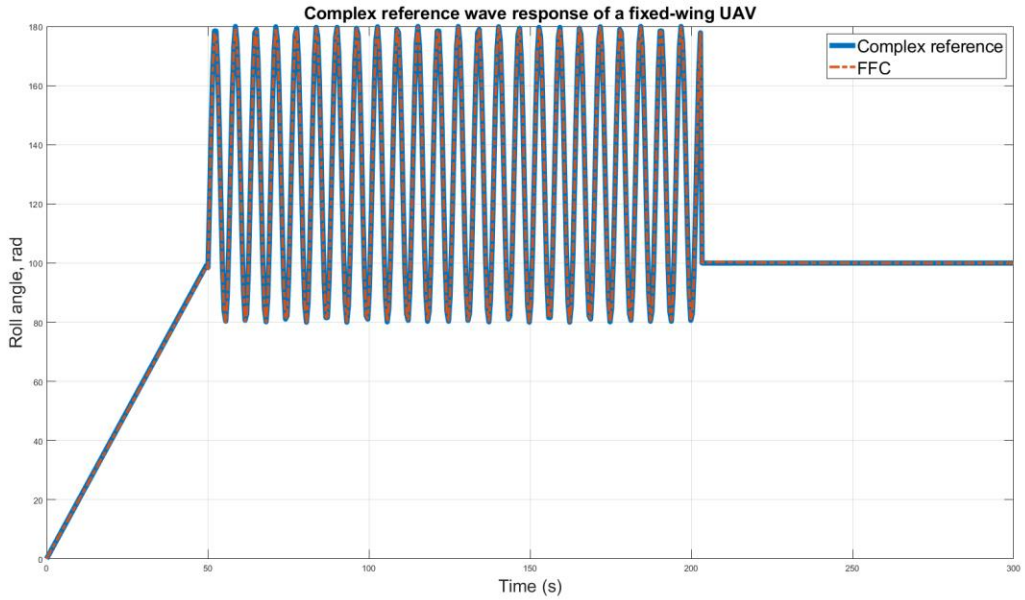

Figure 4.4.10: Feed-forward compensation method. Complex reference: Ramp + Sine wave response with $L = 100$

As we can observe from the figures above, closed-loop control using FFC method perfectly follows the different trajectories that was applied to the system. Corresponding Nyquist and root-locus plots are shown in Figure 4.4.11 and Figure 4.4.12 respectively:

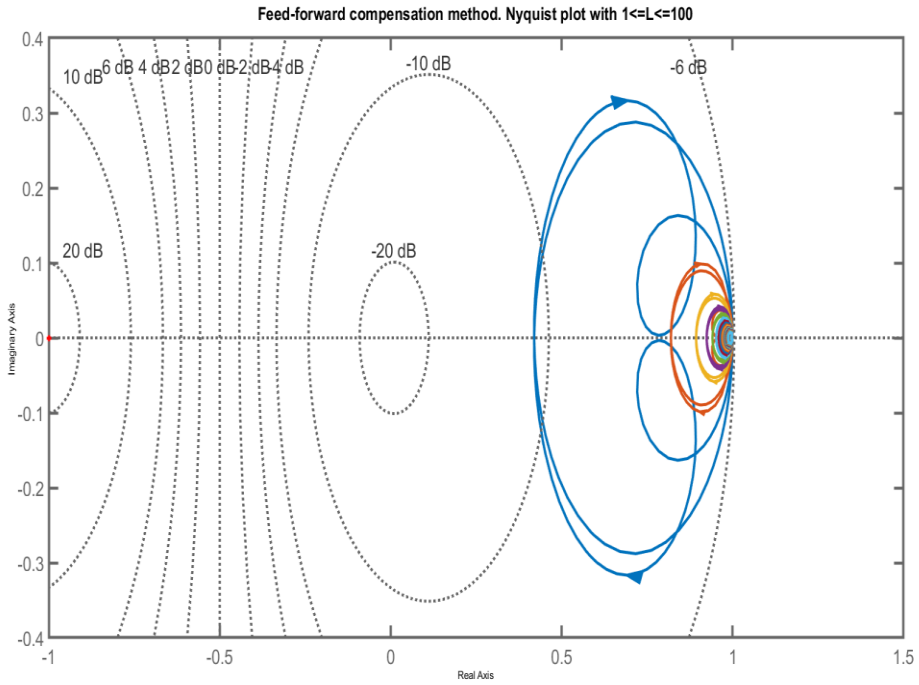

Figure 4.4.11: Feed-forward compensation method. Nyquist plot with 1≤L≤100

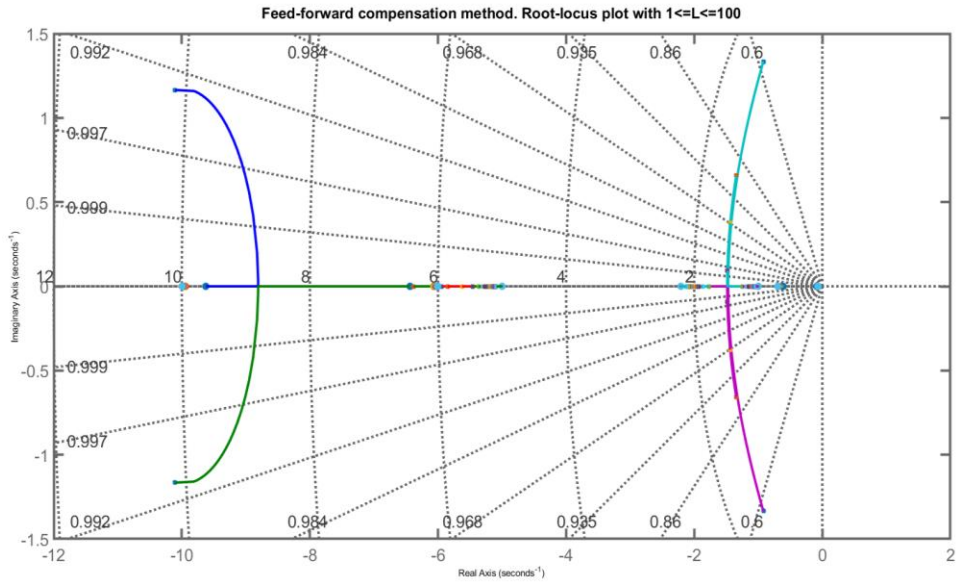

Figure 4.4.12: Feed-forward compensation method. Root-locus plot with 1≤L≤100

4.5 Robustness check

In order to research the robustness of the feed-forward compensation method, the uncertainty model based on the 5 parameters(a_1, a_2, b_1, b_2, L) . Based on this uncertain model, the following model with corresponding criteria performed:

$$\mathrm{a_{1nom}} = 5, \mathrm{a_1} = \mathrm{a_{1nom}} \pm 20\% \quad \textbf{(4.5.1)}$$

$$a_{2nom} = 1, a_2 = a_{2nom} \pm 20\% \quad \textbf{(4.5.2)}$$

$$b_{1nom} = 6, b_1 = b_{1nom} \pm 20\% \quad \textbf{(4.5.3)}$$

$$b_{2nom} = 8, b_2 = b_{2nom} \pm 20\% \quad \textbf{(4.5.4)}$$

$$L_{nom} = 100, L = L_{nom} + 50\% \quad \textbf{(4.5.5)}$$

The nominal values of coefficients are chosen randomly and checked with MATLAB script. However, the main criteria to choose these values was to select a stable poles and zeros which are not too close to the unstable poles/zeros. The robustness test will be conducted in 2 different ways, such as

- Monte-Carlo based approach using “robstab” MATLAB function
- Testing the performance for noise function

Moreover, there are other ways of testing for the worst-case scenario such as H-infinity and mu-synthesis. However, for the first sense the Monte-Carlo based approach, that focuses on iterating through each individual scenario within the range, helps to determine the robustness of the system. The code package,script and used functions for robustness check are shown in Appendix B. MATLAB robstab command retrieved the result of the test in the way that is shown in Figure 4.5.1.

```
System is not uncertain, robustness margins are infinite (assuming nominal stability for UFRD).
StabilityMargin =
  struct with fields:

          LowerBound: Inf
          UpperBound: Inf
    CriticalFrequency: NaN
```

Figure 4.5.1: "Robstab" MATLAB command result

Monte-Carlo based approach is used to iterate over the allowed range of uncertain parameters and check whether in that range the instability occurs. If so, the output from "robstab" command retrieves the upper and lower bound tolerance of selected parameters. In our case, It states that the system is not uncertain and not sensitive for these 5 parameters and the robustness margins are infinite. As well as the UFRD (uncertain frequency response data) models are nominally stable (This fact is shown in the Figure 4.5.2, where the MATLAB documentation about robstab command is displayed).

```
robstab   Robust stability of uncertain system.

   [STABMARG,WCU] = robstab(USYS) calculates the robust stability margin
   for the uncertain system USYS (USS or UFRD). This margin is relative to
   the  uncertainty level specified in USYS. A robust stability margin
   greater than 1 means that USYS is stable for all values of its modeled
   uncertainty. A robust stability margin less than 1 implies that USYS
   becomes unstable for some values of the uncertain elements within their
   specified ranges. For example, a margin of 0.5 implies that
     * USYS remains stable as long as the uncertain element values stay
       strictly within 0.5 normalized units of their nominal values
     * There is a destabilizing perturbation of size 0.5 normalized units.
   Use NORMALIZED2ACTUAL or USCALE to convert normalized deviations from
   nominal values to actual uncertainty ranges. The stability margin is 0
   if the nominal value of USYS is unstable. Note that UFRD models are
   always assumed to be nominally stable.
```

Figure 4.5.2: MATLAB documentation of the "robstab" command

4.6 Testing for noise function

Assume the disturbance function

$$N(s) = \frac{0.5}{s+4} \qquad \textit{(4.6.1)}$$

Then the time plots with step response and sine wave behaviors are shown in Figure 4.6.1, Figure 4.6.2 respectively.

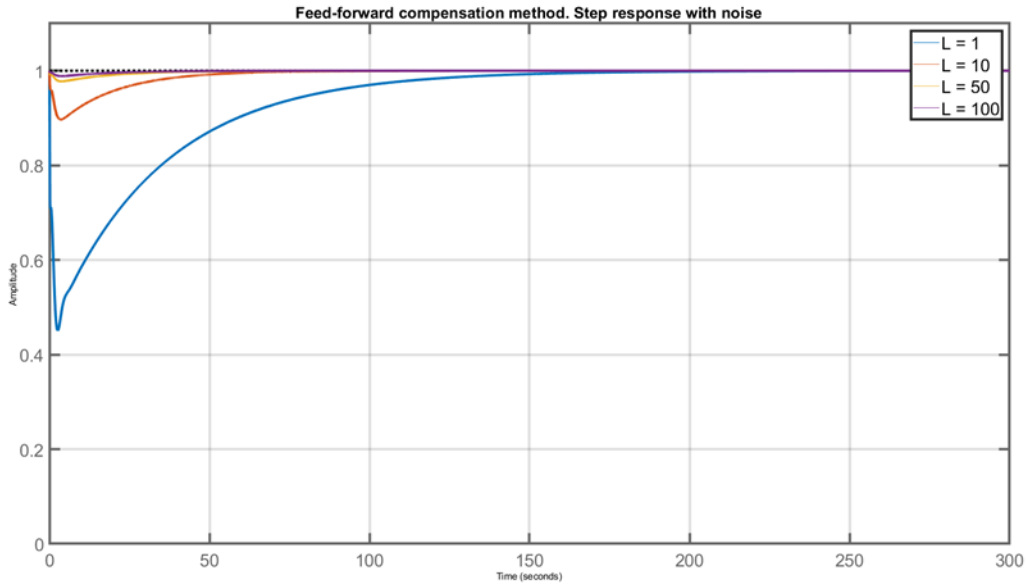

Figure 4.6.1: Feed-forward compensation method. Step response with noise

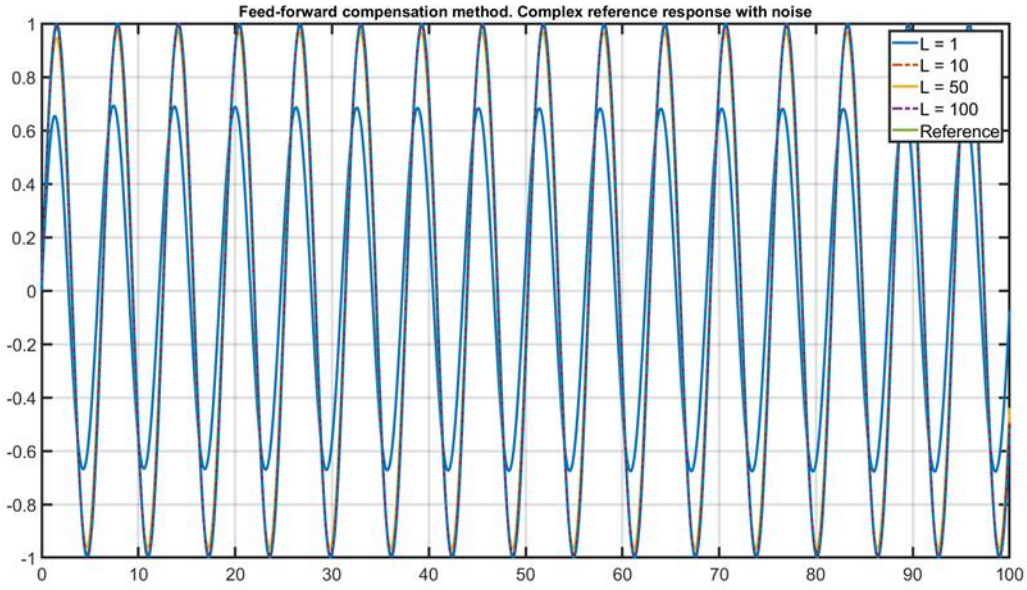

Figure 4.6.2: Feed-forward compensation method. Complex reference response with noise

It is clear from the figures that the simulation results with noise shows the similar performance, when it was without it, thereby proving that the designed FFC controller is robust stable and not sensitive for external disturbances.

Chapter 5- Conclusions

During the simulation and parameter tuning process the step response information of the closed-loop model using the PID controller and feed-forward compensation method showed an overshoot of 1.31e-12 which is close to zero and a rise time of almost 3e-13 seconds with the L = 100, which was the optimal value, as well as the undershoot problem at the initial phase was resolved.

These findings are the positive origin to continuing with the different UAV types. Furthermore, the simulation results presented in this Thesis can be verified experimentally by a real fixed-wing UAV, which was the primary limitation of this work, due to the Covid-19 and contract regulation issues. By building the setup in the laboratory, a real hardware-software alongside robustness check can be performed and verified in different disturbances. Moreover, this work can also be extended to the Artificial Intelligence controllers such as designing the Neural Network based controllers, various non-linear controllers and making a comparison with an existing metrics. Also, the parallel-feedforward compensation and derivative method can also be designed and tested for a fixed-wing UAV to see whether it improves the performance and stabilizes the system. Also, as the system is more complicated, and in this Thesis work only the roll angles-based model was considered, the other aircraft components should also be tested. In this case, the number of decision variables will increase (the dynamic equation is more complex), and in the same way, it is computationally expensive in terms of compilation time compared to simple structures. Also, the high nonlinearity of the dynamics and constraints can limit to perform it to the complex systems.

References

[1] Gao, X., Jia, H., Chen, Z., Yuan, G., Yang, S. (2020). UAV security situation awareness method based on semantic analysis. Proceedings of 2020 IEEE International Conference on Power, Intelligent Computing and Systems, ICPICS 2020, 272–276. https://doi.org/10.1109/ICPICS50287.2020.9201954

[2] Wang, X., Gu, Y., Ye, C. (2020). Improvement design of agricultural plant protection UAV. Proceedings - International Conference on Artificial Intelligence and Electromechanical Automation, AIEA 2020, 479–482. https://doi.org/10.1109/AIEA51086.2020.00108

[3] Li, X., Yang, L. (2012). Design and implementation of UAV intelligent aerial photography system. Proceedings of the 2012 4th International Conference on Intelligent Human-Machine Systems and Cybernetics, IHMSC 2012, 2, 200–203. https://doi.org/10.1109/IHMSC.2012.144

[4] Yang, X., Lin, D., Zhang, F., Song, T., Jiang, T. (2019). High Accuracy Active Stand-off Target Geolocation Using UAV Platform. ICSIDP 2019 - IEEE International Conference on Signal, Information, and Data Processing 2019, 2019–2022. https://doi.org/10.1109/ICSIDP47821.2019.9172919

[5] Yanying, T., Rongchun, Z., Xiaoping, Z., Zhang, B. (2007). The design and implementation of autonomous mission managers for small UAVs. 2007 IEEE International Conference on Control and Automation, ICCA, 00, 177–181. https://doi.org/10.1109/ICCA.2007.4376342

[6] Sineglazov, V. M., & Godny, A. P. (2015). Model of computer-aided design environment for UAV projects. 2015 IEEE 3rd International Conference Actual Problems of Unmanned Aerial Vehicles Developments, APUAVD 2015 - Proceedings, 41–44. https://doi.org/10.1109/APUAVD.2015.7346554

[7] Yoo, C. S., Cho, A., Park, B. J., Kang, Y. S., Shim, S. W., & Lee, I. H. (2012). Collision avoidance of Smart UAV in multiple intruders. International Conference on Control, Automation and Systems, 443–447.

[8] Hassan Tanveer, M., Faiz Ahmed, S., Hazry, D., Warsi, F. A., & Kamran Joyo, M. (2013). Stabilized controller design for attitude and altitude controlling of quad-rotor under disturbance and noisy conditions. American Journal of Applied Sciences, 10(8), 819–831.

[9] Desa, H., Lumpur, K., Kamran, M., Universiti, J. (2013). Roll Angle Stabilization of Fixed-wing UAVs in occurrence of noises by using PID with EKF controller. 63(September), 189–200.

[10] Warsi, F. A., Hazry, D., Ahmed, S. F., Joyo, M. K., Tanveer, M. H., Kamarudin, H., Razlan, Z. M. (2014). Yaw, Pitch and Roll con- troller design for fixed-wing UAV under uncertainty and perturbed condition. Proceedings - 2014 IEEE 10th International Colloquium on Signal Processing and Its Applications, CSPA 2014, May, 151–156. https://doi.org/10.1109/CSPA.2014.6805738

[11] Kada, B., Ghazzawi, Y. (2011). Robust PID Controller Design for an UAV Flight Control System. Lecture Notes in Engineering and Computer Science, 2194(1), 945–950.

[12] Zhai, R., Zhou, Z., Zhang, W., Sang, S., Li, P. (2014). Control and navigation system for a fixed-wing unmanned aerial vehicle. AIP Advances, 4(3). https://doi.org/10.1063/1.4866169

[13] Johansen, T. A., Fossen, T. I. (2020). Guidance, Navigation, and Control of Fixed-Wing Unmanned Aerial Vehicles. Encyclopedia of Robotics, 1–9.https://doi.org/10.1007/978-3-642-41610-167-1

[14] Peng-Ya, X., Yun-Jie, W., Jing-Xing, Z., Ling, C. (2018). Longitudinal attitude control of UAV based on fuzzy PID. 2018 IEEE CSAA Guidance, Navigation and Control Conference, CGNCC 2018. https://doi.org/10.1109/GNCC42960.2018.9019030

[15] Ren, W., Beard, R. W. (2004). Constrained nonlinear tracking control for small fixed-wing unmanned air vehicles. Proceedings of the American Control Conference, 5, 4663–4668. https://doi.org/10.1109/ACC.2004.182688

[16] Jackson, S., Tisdale, J., Kamgarpour, M., Basso, B., Hedrick, J. K. (2008). Tracking controllers for small UAVs with wind disturbances: Theory and flight results. Proceedings of the IEEE Conference on Decision and Control, 564–569. https://doi.org/10.1109/CDC.2008.4739415

[17] M. Kamran Joyo, Ahmed, S. F., Hazry, D., Tanveer, M. H., Warsi, F. A. (2013). Position Controller Design for Quad-rotor under Perturbed Condition. Wulfenia, 20(7),178–189.https://bit.ly/3veRAZU

[18] Wong, S., Yu, J. (2012). Autonomous Stable Flight with a PID Controller

[19] Lee, J. H., Min, B. M., Kim, E. T. (2007). Autopilot design of tilt-rotor UAV using particle swarm optimization method. ICCAS 2007 - International Conference on Control, Automation and Systems, 1629–1633. https://doi.org/10.1109/ICCAS.2007.4406594 https://doi.org/10.1109/ICCA.2007.4376342

[20] Kang, Y. S., Park, B. J., Cho, A., Yoo, C. S., Koo, S. O. (2012). Flight test of flight control performance for airplane mode of Smart UAV. International Conference on Control, Automation and Systems, 2, 1738–1741.

[21] Noth, a, Bouabdallah, S., Siegwart, R. (2006). Dynamic Modeling of Fixed-Wing UAVs. Master Course in Mechanical Engineering, 0–11.

[22] Bodo, Z., Lantos, B. (2019). Modeling and control of fixed-wing UAVs. SACI 2019 - IEEE 13th International Symposium on Applied Computational Intelligence and Informatics, Proceedings, 332–337. https://doi.org/10.1109/SACI46893.2019.9111573

[23] Zhang, J., Zhu, X., Zhou, Z. (2010). Design of time delayed control systems in UAV using model based predictive algorithm. CAR 2010 - 2010 2nd International Asia Conference on Informatics in Control, Automation and Robotics, 1, 269–272. https://doi.org/10.1109/CAR.2010.5456849

[24] Liu, Z., Zhu, M., Tan, F. (2014). Control system design and simulation for UAV during aerial refueling. 26th Chinese Control and Decision Conference, CCDC 2014, 1, 3482–3486. https://doi.org/10.1109/CCDC.2014.6852781

[25] Mooney, J. G., Johnson, E. N. (2014). A Comparison of Automatic Nap-of-the-earth Guidance Strategies for Helicopters. Journal of Field Robotics, 33(1), 1–17. https://doi.org/10.1002/rob

[26] Mobarez, E. N., Sarhan, A., Mohamed, A. M. (2019). Modeling of fixedwing UAV and design of multivariable flight controller using PID tuned by local optimal control. IOP Conference Series: Materials Science and Engineering, 610(1). https://doi.org/10.1088/1757- 899X/610/1/012016

[27] Hu, X., Liu, J. (2020). Research on UAV Balance Control Based on Expert-fuzzy Adaptive PID. Proceedings of 2020 IEEE International Conference on Advances in Electrical Engineering and Computer Applications, AEECA 2020, 787–789. https://doi.org/10.1109/AEECA49918.2020.9213511

[28] Xiao, Y., Fu, Y., Wu, C., Shao, P. (2016). Modified model reference adaptive control of UAV with wing damage. Proceedings - 2016 the 2nd International Conference on Control, Automation and Robotics, ICCAR 2016, 189–193. https://doi.org/10.1109/ICCAR.2016.7486724

[29] Ang, K. H., Chong, G., & Li, Y. (2005). PID control system analysis, design, and technology. IEEE Transactions on Control Systems Technology, 13(4), 559–576. https://doi.org/10.1109/TCST.2005.847331

[30] Stevens, Brian L., et al. Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous Systems. 3rd ed., John Wiley & Sons, 2016.

[31] Dudiki, V. (2018). Feed-Forward Compensation of Non-Minimum Phase Systems FEED-FORWARD COMPENSATION OF NON-MINIMUM PHASE SYSTEMS A thesis submitted in partial fulfillment.

Appendices

MATLAB script of closed-loop control of a fixed-wing UAV using PID controller and eliminating the non-minimum phase system using Feed-forward compensation method.

Appendix A

```
clear all;clc;close all;format compact;
den = [1 1.551 4.848 3.25 0.199];
num = [1.202 -0.3684 -5.064 1.2685];
G_prime = tf(num,den);
G_closed_prime = feedback(G_prime,1);
% Tuning P controller
[C_p,info] = pidtune(G_prime,'P');
T_p = feedback(C_p*G_prime,1);
% Tuning PI controller
[C_pi,info] = pidtune(G_prime,'PI');
T_pi = feedback(C_pi*G_prime,1);
% Tuning PID controller
[C_pid,info] = pidtune(G_prime,'PID');
T_pid = feedback(C_pid*G_prime,1);
%% Tuning uncertainty parameters of Feed-forward compensation method
% We tune a,b and l
% where a and b are the corresponding pole-zero pairs (s + a) and (s + b)
index = 1;
figure();
a1 = 5; b1 = 6;
a2 = 1; b2 = 10;
% for a = 1:5
%  for b = 1:5
u = sin(k*t);
for L = [1 10 50 100]
%         fprintf("%d,%d ",a,b)
        fprintf("L = %d\n",L)
        Z1 = [-2.208 -a1 -a2 ];
        P1 = [-0.393-2.01i -0.393 + 2.01i];
        K1 = 1.202;
G1_prime = zpk(Z1,P1,K1);
% G2(s) part1,P,K);
Z2 = [2.085 0.2499];
P2 = [-0.06798 -0.6971 -b1 -b2];
K2 = 1.0;
G2_prime = zpk(Z2,P2,K2);
% Designing L
%   L = 1.5;
```

```
        [num_G1,den_G1] = tfdata(G1_prime);
        num_G1 = cell2mat(num_G1);
        den_G1 = cell2mat(den_G1);
        [num_G2,den_G2] = tfdata(G2_prime);
        num_G2 = cell2mat(num_G2);
        den_G2 = cell2mat(den_G2);

        G_noise = tf(0.5,[1 4]);
        C_s = C_pi;
        G1_s = G1_prime; G2_s = G2_prime;
%       L = 2;
        G_tot1 = C_s*G1_s;
        G_tot2 = G2_s + L;
        G_tot12 = G_tot1*G_tot2;
        G_tot13 = G_tot12 + G_noise;
        G_total_noise = feedback(G_tot13*1,1);
        G_total = feedback(G_tot12*1,1);
        t_step = 300;
        y1 = lsim(G_total_noise, u, t);
        plot(t,y1,'Linewidth',2)
        hold on
        plot(t,u,'-.','Linewidth',4)
        hold on
%       step(G_total_noise,t_step)
%       hold on
%       step(G_total,t_step)
%       hold on
%       rlocus(G_total_noise)
%       nyquist(G_total_noise)
        hold on
        index = index + 1;

end
grid on
% ylim([0 1.1])
legend('L = 1','L = 10','L = 50','L = 100','Reference')
set(findall(gcf,'-property','FontSize'),'FontSize',20)
set(findall(gcf,'-property','Linewidth'),'Linewidth',3)
title('Feed-forward compensation method. Complex reference response with noise','FontSize',18)
% title('Feed-forward compensation method. Step response with noise','FontSize',18)
% title('Feed-forward compensation method. Root-locus plot with noise','FontSize',18)
% title('Feed-forward compensation method. Step response with noise','FontSize',18)
set(gcf, 'Position', get(0, 'Screensize'));

%% Sine wave response
```

```
t = linspace(0, 100, 1000);
figure()
for k = 1:1
    u = sin(k*t);
    y1 = lsim(G_total, u, t);
    y2 = lsim(G_prime,u,t);
    y3 = lsim(T_pi,u,t);

    plot(t,y2,t,y3,t,y1,t,u)
%     hold on
%     plot(t,y3,'k-*','Linewidth',2)
%     hold on
%     plot(t,y1,'b-','Linewidth',2)
%     plot(t,u,'r-.','Linewidth',2)
%     hold on
    grid on
end
legend('Plant','Plant + PI','Plant + PI + FFC','Reference')
xlabel('Time,(s)')
ylabel('Roll angle,(rad)')
title('Feed-forward compensation method. Sine wave response with L = 1','FontSize',18)
set(gcf, 'Position', get(0, 'Screensize'));
set(findall(gcf,'-property','FontSize'),'FontSize',18)
set(findall(gcf,'-property','Linewidth'),'Linewidth',3)

%% Square wave response
h=2;TimeEnd=20; % inputs
 t=0:0.01:TimeEnd;
 x=h/2:2*h:t(end);
 n=length(x);
 u_square=h*rectpuls(t-x(1),h);
 for i=2:n
     u_square=u_square+h*rectpuls(t-x(i),h);
 end

 y_square = lsim(G_total,u_square,t);
 plot(t,y_square,'-.','Linewidth',4)
 hold on
 plot(t,u_square,'-','Linewidth',3)
 grid on
 set(gcf, 'Position', get(0, 'Screensize'));
 legend('FFC','Square input')
 xlabel('Time (s)','FontSize',20)
 ylabel('Roll angle (rad)','FontSize',20)
 title('Feed-forward compensation method with square wave response','FontSize',20)
%% Complex reference
```

```
t_complex = linspace(0,300,600);
u1_t = 2*t_complex(1:100);
u2_t = 50*sin(t_complex(100:405)) + 130;
u3_t = 100*ones(1,194);
u_complex = [u1_t u2_t u3_t];
figure()
y_ffc_complex = lsim(G_total,u_complex,t_complex);
y_plant_complex = lsim(G_prime,u_complex,t_complex);
y_pid_complex = lsim(T_pi,u_complex,t_complex);
plot(t_complex,u_complex,'Linewidth',6)
hold on
plot(t_complex,y_ffc_complex,'-.','Linewidth',3)
hold on
% plot(t_complex,y_plant_complex,t_complex,y_pid_complex)
% grid on
legend('Complex reference','FFC','Plant','PI','FontSize',20)
xlabel('Time (s)','FontSize',20)
ylabel('Roll angle, rad','FontSize',20)
title('Complex reference wave response of a fixed-wing UAV','FontSize',20)
set(gcf, 'Position', get(0, 'Screensize'));
grid on
```

Appendix B

```
%% Robustness check
% Define 5 uncertain parameters
a1 = ureal('a1',5,'percent',20);
a2 = ureal('a2',1,'percent',20);
b1 = ureal('b1',6,'percent',20);
b2 = ureal('b2',8,'percent',20);
L = ureal('L',100,'percent',50);

s = zpk('s');
G1_s = 1.202*(s + 2.208)*(s + a1)*(s + a2)/(s^2 + 0.7859*s + 4.199);
G2_s = (s - 2.085*s)*(s - 0.2499)/((s + 0.06798)*(s + 0.6971)*(s + b1)*(s + b2));

% Uncertain state-space models
C_s = C_pi;
%       L = 2;
G_tot1 = C_s*G1_s;
G_tot2 = G2_s + L;
G_tot12 = G_tot1*G_tot2;
G_tot13 = G_tot12 + G_noise;
G_total = feedback(G_tot13,1);
```

```
Tnom = zpk(G_total.nominal);
maxrealpole = max(real(pole(Tnom)))

opt = robOptions('Display','on','Sensitivity','on');
[StabilityMargin,wcu] = robstab(Tnom,opt)
```