MODEL PREDICTIVE CONTROLLER IMPLEMENTATION TO A MULTICOPTER FOR STABILITY AUGMENTATIONS

MODEL ÖNSEZİLİ KONTROL İLE ÇOK ROTORLU HAVA ARACI'NIN KARARLILIĞININ İYİLEŞTİRİLMESİ

CANTÜRK SANAN

ASSOC. PROF. DR S. ÇAĞLAR BAŞLAMIŞLI
Supervisor

Submitted to
Graduate School of Science and Engineering of Hacettepe University
As a Partial Fulfillment to the Requirements for the Award of the Degree of Master of Science in Mechanical Engineering.

2021

ABSTRACT

MODEL PREDICTIVE CONTROLLER IMPLEMENTATION TO A MULTICOPTER FOR STABILITY AUGMENTATIONS

Cantürk SANAN

Master of Science Degree, Department of Mechanical Engineering
Supervisor: Assoc. Prof. Dr. S. Çağlar BAŞLAMIŞLI
February 2021, 152 pages

Nowadays, multi-rotor systems are widely used in military, civil and academic applications thanks to their cheap, simple and suitable structures for use in different areas such as surveillance and load carrying. On the other hand, thanks to processor and sensor technologies' progress, modern and high computation power-requiring control methods, such as model predictive controller, have become applicable even on these small systems. This thesis study aims to improve the stability and performance of a multi-rotor system with a model predictive controller.

The thesis study can be summarized in six main processes. During the study, firstly, a system model consistent with system identification was obtained. Subsequently, this model was verified with flight tests. In the second part, a simulation model was established using MATLAB-Simulink software and, a model predictive controller was designed with "MPC Designer". This controller has been tested in the simulation environment and compared with the reference PID controller. After satisfactory performance results were obtained, C code was generated with "Embedded Coder", it was added to the existing autopilot software, and software in the loop tests were completed. After this step, the generated code was implemented on the Jetson NANO board and the hardware in the loop tests were started. After accomplishing sufficient performance in these tests, flight tests were started. Finally, the system was tested in a real application and compared with the reference controller.

Keywords: Model Predictive Control, System Identification, Multirotor, Quadcopter, Software in the Loop, Hardware in the Loop

ÖZET

MODEL ÖNSEZİLİ KONTROL İLE ÇOK ROTORLU HAVA ARACI'NIN KARARLILIĞININ İYİLEŞTİRİLMESİ

Cantürk SANAN

Yüksek Lisans, Makina Mühendisliği Bölümü
Danışman: Doç. Dr. S. Çağlar BAŞLAMIŞLI
Şubat 2021, 152 sayfa

Çok rotorlu sistemler ucuz, basit olmaları ve gözetleme, yük taşıma gibi farklı alanlarda kullanılmaya elverişli yapıları sayesinde günümüzde askeri, sivil ve akademik uygulamalarda yaygın olarak kullanılmaktadır. Öte yandan, işlemci ve sensör teknolojilerindeki ilerleme sayesinde, bu küçük sistemler üzerinde bile model önsezili kontrolcü gibi modern ve yüksek işlem gücü gerektiren kontrol yöntemleri uygulanabilir hale gelmiştir. Bu tez çalışması da temelde model önsezili kontrolcü ile çok rotorlu bir sistemin kararlılığında ve performansında iyileştirme sağlanmasını hedeflemektedir.

Tez çalışması temel olarak altı ana süreçte özetlenebilir. Çalışma sırasında öncelikle sistem tanımlama ile tutarlı bir sistem modeli elde edilmiştir. Ardından elde edilen bu modelin uçuş testleri ile doğrulaması yapılmıştır. İkinci bölümde ise MATLAB-Simulink yazılımları kullanılarak bir benzetim modeli kurulmuş ve "MPC Designer" ile model önsezili kontrolcü tasarımı yapılmıştır. Tasarlanan bu kontrolcü benzetim ortamında sınanmış ve referans PID kontrolcü ile karşılaştırılmıştır. Tatmin edici performans sonuçlarının alınmasının ardından "Embedded Coder" ile C kodu üretilmiş, mevcut otopilot yazılımına eklenmiş ve döngüde yazılım testleri tamamlanmıştır. Bu aşamadan sonra üretilen kod Jetson NANO kartı üzerine yerleştirilerek döngüde donanımsal benzetim safhasına geçilmiştir. Bu testlerde de yeterli performansın görülmesinin ardından ise uçuş testlerine geçilmiştir. Son olarak uçuş testlerinde sistemin gerçek uygulamada sınanmış ve referans kontrolcü ile karşılaştırılmıştır.

Anahtar kelimeler: Model Önsezili Kontrol, Sistem Tanımlama, Çok-rotorlu, Dört-rotorlu, Döngüde Yazılım, Döngüde Donanımsal Benzetim

ACKNOWLEDGEMENTS

I would like to express my special thanks to my supervisor, Assoc. Prof. Dr. S. Çağlar BAŞLAMIŞLI for his supports and guidance during the entire period. Our meetings were always inspiring and fulfilling.

I would like to express my sincere thanks to Alperen KALE and Fetihhan GÜRAN, who have always been by my side and have enjoyed all the work we have done since the first day I started university.

In addition, I would like to thank A.Erkin ARSLAN, F. Ersel ÖLÇER, K. Burak ÜNAL and Sinan PAKKAN from Aselsan A.Ş., who spared me time and solved my problems with their knowledge and experience whenever I needed.

Finally, I would like to express my deepest gratitude to my mother, father, and sister, who brought me to these days and made me better every day with their supports. They always trusted and encouraged me under though times. Besides, I sincerely thank my dear wife, who has made me happier and calmer since the day she came into my life.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

LIST OF SYMBOLS & ABBREVIATIONS

List of Abbreviations

LQR	Linear Quadratic Regulator
MIMO	Multiple Input, Multiple Output
MPC	Model Predictive Control
PID	Proportional, Integral, Derivative
RMS	Root Mean Square
SIL	Software in the Loop
SISO	Single Input, Single Output
HIL	Hardware in the Loop
UAV	Unmanned Aerial Vehicle
MV	Manipulated Variables
OV	Output Variables
LBMPC	Learning Based Model Predictive Controller
UART	Universal Asynchronous Receiver Transmitter
NMPC	Non-linear Model Predictive Controller
QP	Quadratic Programming

List of Symbols

x	The inertial position of the UAV towards x_I in C_I
y	The inertial position of the UAV towards y_I in C_I
z	The inertial position of the UAV towards z_I in C_I
u	The body frame velocity of the UAV measured along x_b in C_b
v	The body frame velocity of the UAV measured along y_b in C_b
w	The body frame velocity of the UAV measured along z_b in C_b
ϕ	The roll angle of the UAV defined with respect to C_2
θ	The pitch angle of the UAV defined with respect to C_1
ψ	The yaw angle of the UAV defined with respect to C_v
p	The roll rate of the UAV measured along x_b in C_b
q	The roll rate of the UAV measured along x_b in C_b
r	The yaw rate of the UAV measured along z_b in C_b
h	The altitude of the aircraft measured along z_v in C_v
X	External longitudinal force on the aircraft center of gravity
Y	External lateral force on the aircraft center of gravity
Z	External vertical force on the aircraft center of gravity
L	External roll moments about the aircraft center of gravity
M	External pitch moments about the aircraft center of gravity
N	External yaw moments about the aircraft center of gravity
δ_{ail}	Lateral pilot reference input in PWM
δ_{ele}	Longitudinal pilot reference input in PWM
δ_{rud}	Directional pilot reference input in PWM
δ_{thr}	Throttle pilot reference input in PWM

X_u	Longitudinal force that is generated by the longitudinal vehicle speed
M_u	Longitudinal moment that is generated by the longitudinal vehicle speed
Y_v	Lateral force that is generated by the lateral vehicle speed
L_v	Lateral force that is generated by the lateral vehicle speed
N_r	Directional moment that is generated by the yaw rate
Z_w	Vertical forces that is generated by the vertical vehicle speed
$X_{\delta_{ele}}$	Longitudinal force that is generated by the longitudinal pilot reference input
$M_{\delta_{ele}}$	Longitudinal moment that is generated by the longitudinal pilot reference input
$Y_{\delta_{ail}}$	Lateral force that is generated by the lateral pilot reference input
$L_{\delta_{ail}}$	Lateral moment that is generated by the lateral pilot reference input
$N_{\delta_{rud}}$	Directional moment that is generated by the directional pilot reference input
$Z_{\delta_{thr}}$	Vertical force that is generated by the vertical pilot reference input
n_ω	Number of frequency points
$\omega_1, \omega_{n_\omega}$	Number of frequency points
$\vert \quad \vert$	Magnitude (dB)at each frequency ω
$\angle$	Phase(deg) at each frequency ω

1. INTRODUCTION

1.1. Problem Definition

Quadcopters are aggressive systems due to their low inertia and weight. In outdoor conditions, this can bring along stability problems, and the system dynamic model must be modelled in a realistic way to control this aggressive system successfully. Today, the assumptions used in the generating of mathematical models of small-sized UAVs can cause inadequate modelling, as they are actually prepared for larger-sized systems[1]. On the other hand, since many studies in the literature remain in the simulation environment, the effects of these errors on the model predictive controller cannot be fully observed.

Designing a control system with models with low accuracy can cause catastrophic consequences. Therefore, in many systems, loose controllers have been used not to risk the security of the platform. But it decreases the performance and robustness of the controller. This is one reason why PIDs, which are widely used today, cannot be optimally designed. In another example, model predictive controllers can encounter performance and stability problems during flight, since they include a model implicitly and create all control outputs according to this model's responses in the following time steps. The first problem aimed to be solved in the thesis is to eliminate the model-based deficiencies.

Another element that is tried to be improved during the thesis is the improvement of the controller performance. In addition to the performance loss arising from modelling deficiencies, PID controllers also have some deficiencies according to advance control methods. By decoding the MPC system in a MIMO way, it can respond faster than the decoupled SISO controller structure and can control with low overshoots. While doing this, it also increases the flight safety of the platform by taking the limits into account that can be entered implicitly in the controller design.

1.2. Scope of the Thesis

The first aim of this thesis is to obtain a dynamic model with high accuracy. To achieve this goal, system identification works in frequency domain have been done by using CIFER software. The system identification process was performed parametrically, and parameters were obtained in accordance with the projected system model. Subsequently, this model was verified by flight tests.

In the second stage of the thesis work, the aim is to design a stable and high performance MPC compared to the reference controller and perform tests in MATLAB-Simulink environment.

In the third stage of the thesis, verification tests have been carried out in SIL and HIL environment with the code generated from the designed MPC controller. Then, the performance of the controller on the real system has been tested with the flight test.

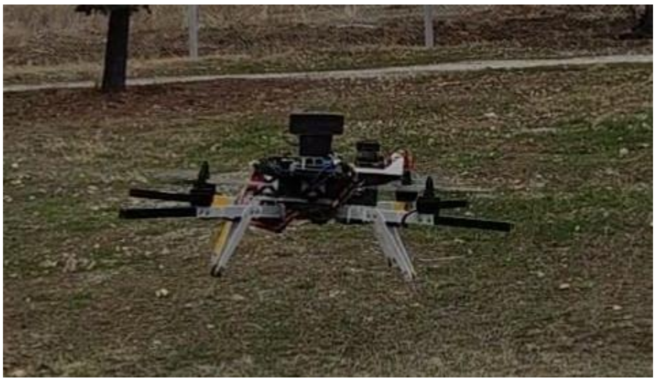


Figure 1-1 Quadcopter test platform

1.3.Thesis Outline

The first part of the thesis is divided into three. These are the thesis contents sections that include the problem definition, the purpose of the thesis, and a summary of the thesis flow. In these sections, general information about the content of the thesis is given.

The second chapter includes a literature review on model predictive controllers and system identification procedure in multi-rotor aircraft. In this section, firstly, general information about model predictive controllers is given, and similar studies in the literature are mentioned. Then, information is given about the QP solver of the model predictive controller used in this thesis. The last part of the literature review is focused on the system identification procedure and the CIFER program.

The third chapter includes the process of deriving the mathematical model in multi-rotor platforms and information about the test platform used. Detailed information about the test system is given in the first part of the chapter. In the following section, information is given about the coordinate systems used in the thesis. After that, obtaining the mathematical model of multi-rotor systems is explained by making an overview of the system identification process. In the last part of this section, the studies and results of the system model are included.

The fourth chapter explains the model predictive controller development studies that form the basis of the thesis. In this section, firstly, the definitions of the parameters that are important in model predictive controller design and the issues to be considered when selecting the values of these parameters are mentioned. During this thesis, MATLAB software was used to develop a model predictive controller. A six degrees of freedom MPC has been developed using the MPC toolbox. In the last part of this section, SIL and HIL tests, which are two important verification methods in model-based design studies, and how these tests are applied in controller development work are explained.

The fifth part of the thesis includes the results obtained from the studies so far, verification tests and comparisons of the developed controller with the reference controller. In this section, firstly, the test results in the simulation model and the comparison of these test results with SIL and HIL tests are included. Finally, with the flight tests, it is confirmed that the implementation has been successfully performed.

In the last chapter, results and future work sections are included. In this section, the studies carried out during the thesis study are explained in detail. In addition, not only the points that are open to improvement in the thesis study, but also ideas about the future studies are included.

2. LITERATURE SURVEY

In this section, besides the abstracts of other studies similar to the content of this thesis, there are titles related to the definition of the coordinate systems of UAVs, the system identification process and the model predictive controller design.

2.1. Model Predictive Controllers (MPC)

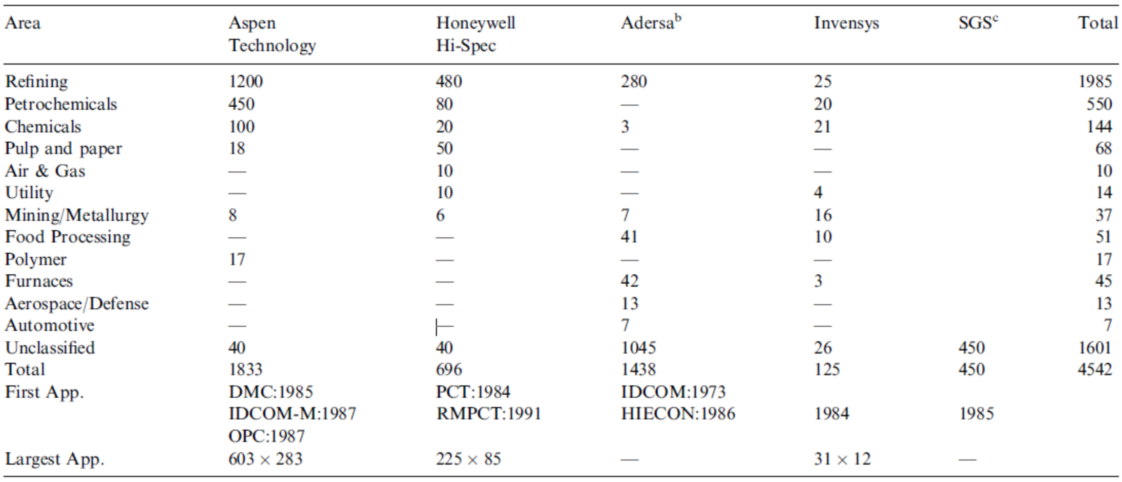
Model predictive control is an advanced controller technique for MIMO systems. We can summarize the working principle of MPC as follows. If we have an accurate dynamic model, it means we can successfully predict the future values of the system states. This knowledge allows us to calculate changes of each input variable to get the desired output. Also, MPC can handle inequality constraints on the input and outputs. In MPC application, output variables of the MPC are called controlled variables (CV), and input variables are called manipulated variables (MV).

Model predictive control technique comes with various advantages.

- MPC can capture the relations between input, output and disturbances.
- MPC can work with constraints implicitly both on inputs and outputs.
- MPC can work with Non-Linear and MIMO systems.
- MPC can ensure early warnings of potential issues using the accurate dynamic model.

MPC is introduced in the 1960s conceptually, but first industrial MPC systems were developed by two initiative group which are Dynamic Matrix Control (DMC) intended by Shell Oil and a correlated approach by ADERSA. It has started to be used in the chemical industry and oil refineries first because of its high computational load and memory needs. For the last few years, MPC has begun to be used in Aerospace and Automotive industries [2].

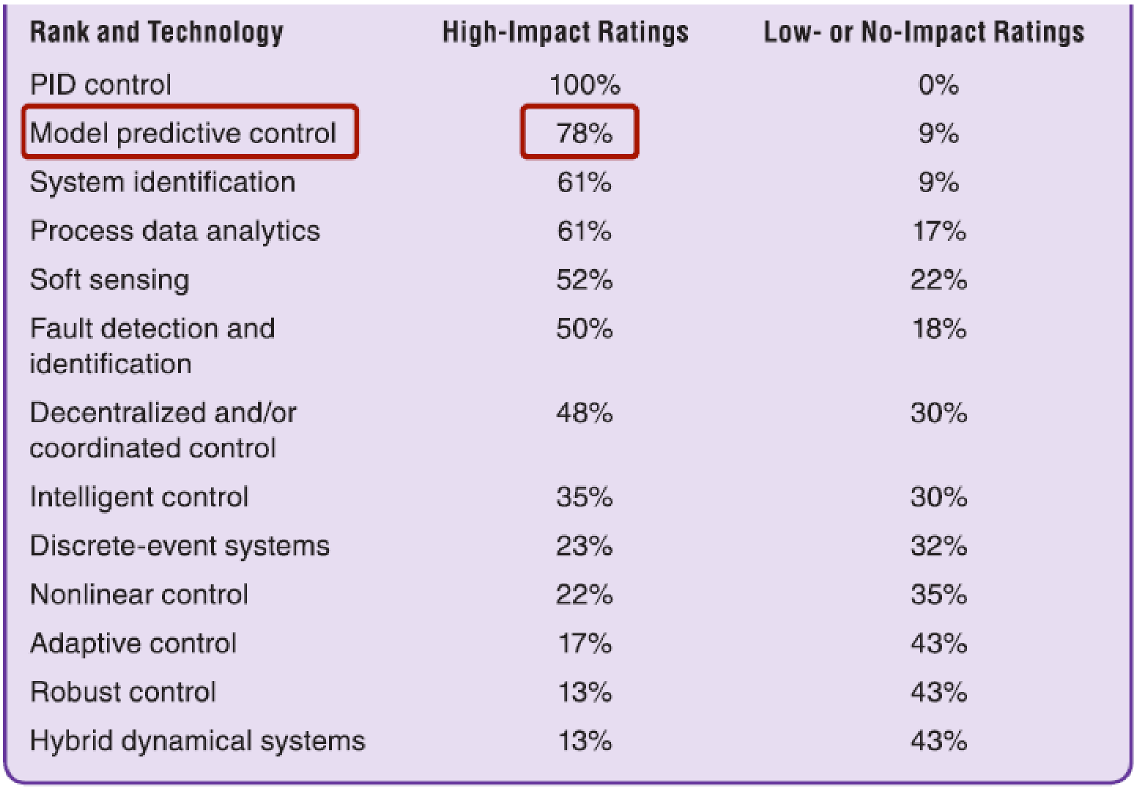
Table 2-1 Industrial survey of MPC applications conducted in mid-1999 [1]



Area	Aspen Technology	Honeywell Hi-Spec	Adersa[b]	Invensys	SGS[c]	Total
Refining	1200	480	280	25		1985
Petrochemicals	450	80	—	20		550
Chemicals	100	20	3	21		144
Pulp and paper	18	50	—	—		68
Air & Gas	—	10	—	—		10
Utility	—	10	—	4		14
Mining/Metallurgy	8	6	7	16		37
Food Processing	—	—	41	10		51
Polymer	17	—	—	—		17
Furnaces	—	—	42	3		45
Aerospace/Defense	—	—	13	—		13
Automotive	—	⊢	7	—		7
Unclassified	40	40	1045	26	450	1601
Total	1833	696	1438	125	450	4542
First App.	DMC:1985 IDCOM-M:1987 OPC:1987	PCT:1984 RMPCT:1991	IDCOM:1973 HIECON:1986	1984	1985	
Largest App.	603 × 283	225 × 85	—	31 × 12	—	

According to the research published by IEEE in 2017, the industrial effect of MPC and system identification has been shown to be high.

Table 2-2 Impact of advanced control technologies in industry [1]



Rank and Technology	High-Impact Ratings	Low- or No-Impact Ratings
PID control	100%	0%
Model predictive control	78%	9%
System identification	61%	9%
Process data analytics	61%	17%
Soft sensing	52%	22%
Fault detection and identification	50%	18%
Decentralized and/or coordinated control	48%	30%
Intelligent control	35%	30%
Discrete-event systems	23%	32%
Nonlinear control	22%	35%
Adaptive control	17%	43%
Robust control	13%	43%
Hybrid dynamical systems	13%	43%

Following the technological leap in the production of processors and electrical components in the recent past, MPC has become applicable to small systems such as the quadcopter.

There are some researches related to quadcopters with MPC. First work is introduced by Bemporad et al. in 2009. It uses a hierarchical MPC method, in this method a linear MPC controller stabilizes the quadcopter around the target points and uses a slower sampling rate Hybrid MP controller in the upper layer [3]. This second hybrid MPC is responsible for generating desired target points to avoid obstacles. They test their controller approach in the simulation environment with the Core 2 Duo CPU under MS Windows.

In another research, Kostas Alexis et al. introduced the Switching Model Predictive Controller method based on Piecewise Affine (PWA) dynamics modelling [4]. They used three cascaded switching MP controllers. In this method, the position controller generates commands for attitude controller and attitude controller has MP controller for each vertical, planar and rotary motion. Then, in 2012 Patrich Bouffard et al. introduced another advanced MPC method which is called as Learning-Based MPC [5]. They used an online learning method to improve MPC performance and robustness. In 2017, Wang et al. presented a Non-linear MPC method for a quadcopter [6]. In 2018 Chinedu Amata Amadi implemented a Model predictive controller on Pixhawk flight controllers angular rate loop [7]. In a study conducted in the Autonomous System Lab at ETH Zurich in 2017, the aggressive trajectory tracking performances of Linear MPC and Non-Linear MPC controllers were compared. In this study, controller performances are also tested on the real system [8]. In addition, studies aimed at improving robustness and performance have been carried out with LBMPC (Learning Based Model Predictive Controller), which is designed to be used together with learning algorithms [5, 9, 10].

When the studies in the literature are examined, it is seen that the articles are basically derived from the flight dynamics equations of the models. Since the performance of MPC, which is a model-based controller, is directly related to model accuracy, it was aimed to increase model accuracy with system identification in the thesis study. On the other hand, most of the studies do not contain comparisons and the performance of MPC against PID, the most common control method, is not understood. This thesis study basically aimed to compare MPC, which is designed by developing an accurate model, and a reference PID controller.

2.1.1. Overview of MPC

MPC can work with MIMO systems and can handle all relations between states of the system. This ability brings design advantage, and it makes MPC design comparatively more intuitive and predictable than the cascaded PID design. For instance, while we design a speed controller for a quadcopter, we know that essential parameters for us are body velocities of the quadcopter. So, we can increase the weight of the body velocities and decrease the weight of the angular rates etc. to obtain the desired performance. On the other hand, there are zero weights at the attitude states in lateral and longitudinal dynamics of the quadcopter.
The overall objectives of an MPC can be summarized as noted below.

- Satisfy inequality constraints on both input and output.
- Developing appropriate control input to drive outputs to their optimal setpoints while the others within their limitations.
- Limiting excessive variance of the input variables.
- Control as many manipulated variables as possible when a feedback channel or actuator is not available.

A simple schematic of MPC workflow can be shown below:

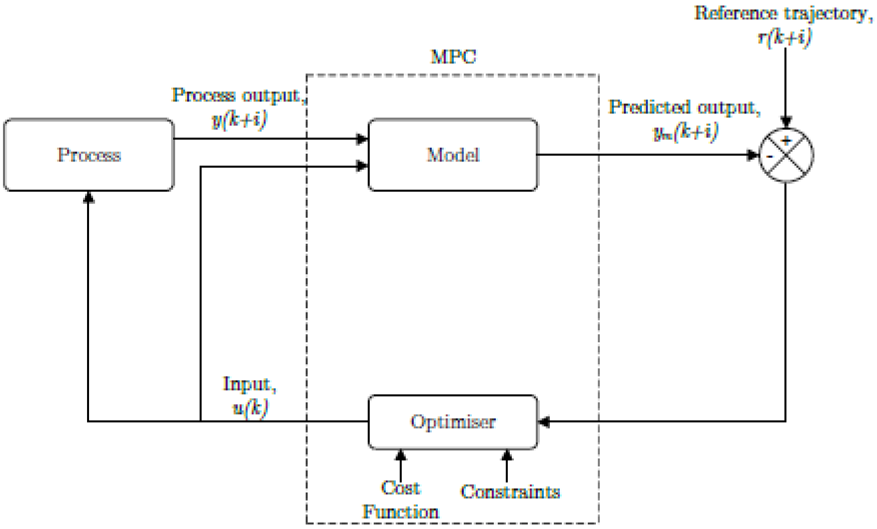


Figure 2-1 Principle of MPC [7]

MPC procedure:

- Generate prediction of future state values of the system with the help of the accurate dynamic model.
- Minimize cost function inside the limits for the decided prediction and control horizon to get optimal control sequence.
- Apply first control input to the real plant.
- Get the current measurement of the real system responses.
- Repeat the procedure from step one for each sampling instant.

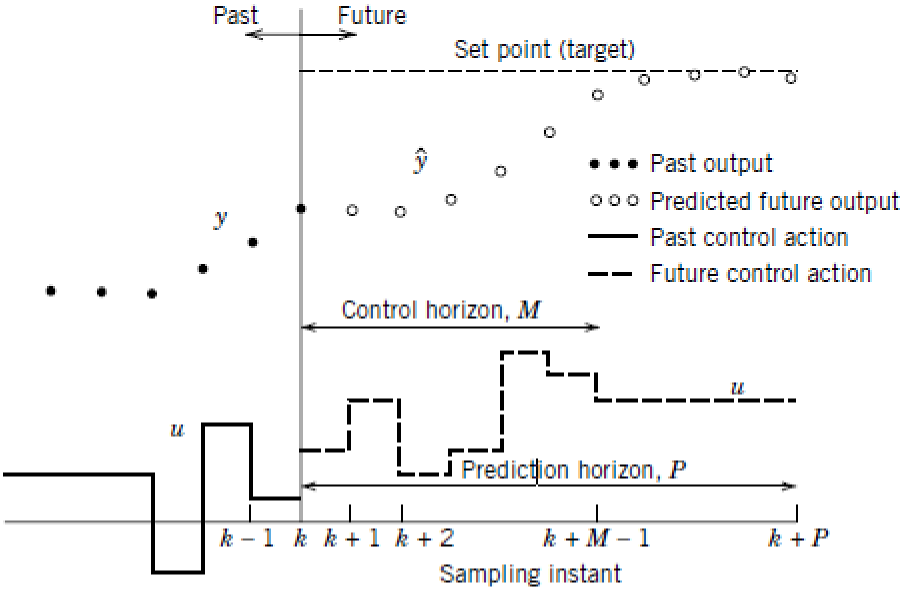


Figure 2-2 Basic concept for MPC [11]

2.1.2. Principles of MPC

Linear MPC uses linear discrete system model to predict the behaviour of the future states. Linear system model can be obtained from Newton-Euler equations or system identification etc. Detailed information about the system model can be found in Section 3.

In this section, generic MPC design is expounded. At first, we need an accurate system model. This model will be used to generate model output at each time step, y_{k+1} ,of the system for a horizon, p, which is called prediction horizon. In this notation we will use k as additional time step, p as prediction horizon, m as control horizon and n as number of states.

In this thesis, constrained Linear MPC has been used. Generic system model can be shown as follow [12]:

$$x_{(k+1)} = Ax_k + \ Bu_k \tag{2.1}$$

$$y_k = Cx_k \tag{2.2}$$

Where;

$$x_0 = x(t) \tag{2.3}$$

$$x_k = x(t+k|t) \tag{2.4}$$

$$u_k = u(t+k|k) \tag{2.5}$$

$$x \in \mathbb{R}^n, u \in \ \mathbb{R}^m, y \in \ \mathbb{R}^p$$

$$u_{min}, u_{max} \in \ \mathbb{R}^m, \qquad y_{min}, y_{max} \in \ \mathbb{R}^p$$

The main objective of MPC is finding a control sequence, $\boldsymbol{z}$, which is minimizing a cost function or a performance index in each time step while satisfying the constraints.
Performance Index [12]:

$$\min_{z} \quad J(z, x_0) = x_N' P x_N + \sum_{k=0}^{N-1} x_k' \, Qx_k + u_k' R u_k \tag{2.6}$$

$$R = R' > 0 \, , \qquad Q = Q' \geq 0, \qquad P = P' \geq 0$$

$$z = \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix}$$

$$J(z, x_0) = x_0'Qx_0 + \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix}' \begin{bmatrix} Q & 0 & 0 & \cdots & 0 \\ 0 & Q & 0 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots & \vdots \\ 0 & \cdots & \cdots & Q & 0 \\ 0 & 0 & \cdots & 0 & P \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_{N-1} \\ x_N \end{bmatrix} + \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-1} \end{bmatrix}' \begin{bmatrix} R & 0 & \cdots & 0 \\ 0 & R & \cdots & 0 \\ 0 & \vdots & \ddots & \vdots \\ 0 & \cdots & 0 & R \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-1} \end{bmatrix} \tag{2.7}$$

$$\begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{bmatrix} = \begin{bmatrix} B & 0 & \cdots & 0 \\ AB & B & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ A^{N-1}B & A^{N-2}B & \cdots & B \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_{N-1} \end{bmatrix} + \begin{bmatrix} A \\ A^2 \\ \vdots \\ A^N \end{bmatrix} x_0 \tag{2.8}$$

$$J(z, x_0) = (\bar{S}z + \bar{T}x_0)'Q\ (\bar{S}z + \bar{T}x_0) + z'\bar{R}z + x_0'Q\ x_0 \tag{2.9}$$

$$J(z, x_0) = \frac{1}{2}z'2(\bar{R} + \bar{S}'\bar{Q}\bar{S})z + x_0'2\bar{T}\bar{Q}\bar{S}z + \frac{1}{2}x_0'2(Q + \bar{T}'\bar{Q}\bar{T})x_0 \tag{2.10}$$

$$J(z, x_0) = \frac{1}{2}z'Hz + x_0'F'z + \frac{1}{2}x_0'Yx_0 \tag{2.11}$$

After that we can find optimum point by zeroing the gradient

$$\nabla_z J(z, x_0) = Hz + Fx_0 = 0 \tag{2.12}$$

from the equation above we can find control sequence $z^* = \begin{bmatrix} u_0^* \\ u_1^* \\ \vdots \\ u_{N-1}^* \end{bmatrix} = -H^{-1}Fx_0$ as batch solution.

When we want to implement constraints to MP controller, we can rearrange the equations. Constraints to satisfy:

$$u_{min} \leq u(t) \leq u_{max} \tag{2.13}$$

$$y_{min} \leq y(t) \leq y_{max} \tag{2.14}$$

Linear prediction model:

$$x_k = A^k x_0 + \sum_{i=0}^{k-1} A^i B u_{k-1-i} \tag{2.15}$$

Constrained optimal control problem:

$$\begin{aligned} \min_z \quad & x_N' P x_N + \sum_{k=0}^{N-1} x_k'\, Q x_k + u_k' R u_k \\ s.t. \quad & u_{min} \le u_k \le u_{max},\;\; k = 0, \dots , N-1 \\ & y_{min} \;\le y_k \le y_{max},\;\; k = 1, \dots , N \end{aligned} \tag{2.16}$$

Optimization problem in quadratic form is denoted below:

$$\begin{aligned} V(x_0) = \frac{1}{2} x_0' Y x_0 + \min_z \frac{1}{2} z' H z + x_0' F z \\ s.t. Gz \le W + S x_0 \end{aligned} \tag{2.17}$$

Note: H is positive definite matrix and H, F, Y, G, W, S related to Q, R, P, constraints and model matrices.

Input constraints:

$$u_{min} \le u_k \le u_{max},\;\; k = 0, \dots , N-1 \tag{2.18}$$

We can combine the bound in a single matrix form.

$$\begin{aligned} u_k &\le u_{max} \\ -u_k &\le\; -u_{min} \end{aligned} \tag{2.19}$$

$$\begin{bmatrix} 1 & 0 & \cdots & 0 \\ 0 & \ddots & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & 1 \\ -1 & 0 & \cdots & 0 \\ 0 & -1 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ 0 & \cdots & 0 & -1 \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix} \le \begin{bmatrix} u_{max} \\ u_{max} \\ \vdots \\ u_{max} \\ -u_{min} \\ -u_{min} \\ \vdots \\ -u_{min} \end{bmatrix} \tag{2.20}$$

Output constraints:

$$y_{min} \leq y_k \leq y_{max}, \ \ k = 1, \ldots, N \tag{2.21}$$

Where;

$$y_k = CA^k x_0 + \sum_{i=0}^{k-1} CA^i B u_{k-1-i} \tag{2.22}$$

$$\begin{bmatrix} CB & 0 & \cdots & 0 \\ CAB & CB & \cdots & 0 \\ \vdots & & & \vdots \\ CA^{N-1}B & \cdots & CAB & CB \end{bmatrix} \begin{bmatrix} u_0 \\ u_1 \\ \vdots \\ u_{N-1} \end{bmatrix} \leq \begin{bmatrix} y_{max} \\ y_{max} \\ \vdots \\ y_{max} \end{bmatrix} - \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^N \end{bmatrix} x_0 \tag{2.23}$$

Upon this point, controller structure type was regulator. After now, we can rearrange the equations to achieve reference tracking.

In this case we will use r(t) as reference signal and $\Delta u(t)$ as input increments.

$$\Delta u(t) = u(t) - u(t-1) \tag{2.24}$$

$$x(t+1) = Ax(t) + Bu(t-1) + B\Delta u(t) \tag{2.25}$$

$$x_u(t-1) = x_u(t) + \Delta u(t) \tag{2.26}$$

$$\begin{bmatrix} x(t+1) \\ x_u(t+1) \end{bmatrix} = \begin{bmatrix} A & B \\ 0 & I \end{bmatrix} \begin{bmatrix} x(t) \\ x_u(t) \end{bmatrix} + \begin{bmatrix} B \\ I \end{bmatrix} \Delta u(t)$$
$$y(t) = \begin{bmatrix} C & 0 \end{bmatrix} \begin{bmatrix} x(t) \\ x_u(t) \end{bmatrix} \tag{2.27}$$

In this implementation, system states are $x(t)$,$x_u(t)$ and input is $\Delta u(t)$. With this implementation quadratic performance index is also reorganized.

$$\min_z \sum_{k=0}^{N-1} \|W^y(y_{k+1} - r(t)\|_2^2 + \|W^{\Delta u}\, \Delta u_k\|_2^2 \tag{2.28}$$

2.2.Quadratic Programming and QP Solvers

Quadratic programming is an optimization problem with quadratic objective function and linear constraints[13]. The general form of quadratic programming problem is mentioned below:

$$\min_{x} q(x) = \frac{1}{2} x^T G x + x^T c$$

$$subject\ to\ \ a_i^T x = b_i, \qquad i \in \mathcal{E} \tag{2.29}$$

$$a_i^T x \geq b_i, \qquad i \in \mathcal{I}$$

Where G is symmetric nxn matrix, $\mathcal{E}$ and $\mathcal{I}$ are finite sets of indices, c, x and $\{a_i\}, i \in \mathcal{E} \cup \mathcal{I}$ in $\mathbb{R}^n$. G is Hessian matrix which is positive semidefinite.

2.2.1. Active-Set method for convex QPs

Active-set method is an iterative optimization method that includes two phases. The main objective of the first phase "The feasibility phase" is defining an algorithm which is a set of constraint called the working set or active set. The working set is a subset of the constraints at the current active point. In this phase, the working set must cover the current point. After that, the algorithm solves an equality constraint problem. And if all the Lagrange multipliers are non-negative, a local solution has been found. If there is a negative Lagrange multiplier, constraints need to be relaxed [13].

Lagrangian of the problem is:

$$\mathcal{L}(x, \lambda) = \frac{1}{2} x^T G x + x^T c - \sum_{i \in I \cup \mathcal{E}} \lambda_i (a_i^T x - b_i) \tag{2.30}$$

Now we can define the active set $(A(x^*))$ that consists of the indices of the constraints for which equality holds at x^*:

$$\{A(x^*) = i \in \mathcal{E} \cup I \mid a_i^T x^* = b_i \} \tag{2.31}$$

$$\min_{x} q(x) = \frac{1}{2}x^T Gx + x^T c$$
$$a_i^T x = b_i, \;\; i \in A(x^*). \tag{2.32}$$

Let define p as step

$$p = x - x_k, \qquad g_k = Gx_k + c \tag{2.33}$$

by substituting for x into equation 2.29 we can get

$$q(x) = q(x_k + p) = \frac{1}{2}p^T Gp + g_k^T p + \rho_k \tag{2.34}$$

Where $\rho_k = \frac{1}{2}x_k^T Gx_k + c^T x_k$ is independent of p. After clearing ρ_k from the objective without changing the solution, now we can get kth iteration as QP subproblem.

$$\min_{p} \frac{1}{2}p^T Gp + g_k^T p$$
$$s.t. \qquad a_i^T p = 0, \qquad i \in W_k. \tag{2.35}$$

For each $i \in W_k$, $a_i^T x$ does not change as we move along p_k, since $a_i^T (x_k + \alpha p_k) = a_i^T x_k = b_i$ for all α. Every constraint in W_k was satisfied at x_k, also satisfied at $x_k + \alpha p_k$ for any value of α.

After solving equation 2.35, if we assume that the optimal p_k is non-zero, we can decide step-length in this direction. If $x_k + p_k$ is feasible with respect to all constraints, $x_{k+1} = x_k + p_k$, otherwise;

$$x_{k+1} = x_k + \alpha_k p_k \tag{2.36}$$

The step-length parameter, α_k, is the largest value in the range [0,1] for which all constraints are satisfied. Since $i \in W_k$ has certainly satisfied regardless of the choice of a_k, explicit definition of a_k can be derived. If $a_i^T p_k < 0$ for some $i \notin W_k$, then for all $\alpha_k \geq 0$, $a_i^T (x_k + \alpha_k p_k) \geq a_i^T x_k \geq b_i$. Therefore, for all non-negative choices of the step length parameter, the constraint i will be satisfied. But for some $i \notin W_k$ when $a_i^T p_k < 0$, however, $a_i^T (x_k + \alpha_k p_k) \geq b_i$ only if

$$\alpha_k \leq \frac{b_i - a_i^T x_k}{a_i^T p_k} \tag{2.37}$$

α_k should be as large as possible to maximize decrease in q. So;

$$\alpha_k \overset{\text{def}}{=} min\, 1, \underset{i\notin W_k, a_i^T p_k<0}{min} \left(\frac{b_i - a_i^T x_k}{a_i^T p_k}\right) \tag{2.38}$$

The constraints i for which the minimum equation above is called blocking constraints. If $\alpha_k = 1$ and no new constraints are active at $x_k + \alpha_k p_k$ then there are no blocking constraints on that iteration. If $\alpha_k < 1$, that is, if the step along p_k is blocked by a constraint not in W_k, a new working set W_{k+1} is created by adding one of the blocking constraints to W_k. By continuing to iterate in this way, we add constraints to the working set until we reach a point $\hat{x}$ that minimizes the quadratic target function over the current working set $\widehat{W}$.

```
Algorithm:
Compute a feasible starting point $x_0$;
Set $W_0$ to be a subset of the active constraints at $x_0$;
**for** $k = 0, 1, 2, \ldots$
    Solve to find $p_k$;
    **if** $p_k = 0$
      Compute Lagrange multipliers $\widehat{\lambda_\imath}$ that satisfy ()
        With $\widehat{W} = W_k$;
        if  $\widehat{\lambda_\imath} \geq 0$ for all $i \,\epsilon\, W_k \cap I$
          **stop** with solution $x^* = x_k$;
        **else**
            $\boldsymbol{j} \leftarrow \mathbf{argmin}_{j\epsilon W_k \cap I}\, \lambda_j$
            $x_{k+1} \leftarrow x_k$;  $W_{k+1} \leftarrow W_k \backslash \{j\}$;
    **else** $(* p_k \neq 0^*)$
        Compute $\alpha_k$ from ()
        $x_{k+1} \leftarrow x_k + \alpha_k p_k$;
        **if** there are blocking constraints
              Obtain $W_{k+1}$ by adding one of the blocking
                  Constraints to $W_k$;
        **else**
              $W_{k+1} \leftarrow W_k$
  **end(for)**
```

2.3. System Identification

The system identification process is the process of obtaining the system through motion measurements as mentioned before. This process can be done over the frequency response or directly in the time-domain. Since most of the assumptions and mathematical models in aviation are not written to fit small class UAVs, modelling UAVs in this class over these equations may give erroneous results. On the other hand, the inconsistency of these models and the simulation with the real system responses may make it difficult to develop a control system and may cause developing non-optimal controllers [1].

Modelling studies performed with the system identification method provide more accurate models for small class UAVs, and this increases the consistency of the simulation. System identification applications for unmanned aerial vehicles can be frequently encountered in the literature [14-18]. The system identification process in the frequency domain is advantageous against the work done in the time domain in many ways. Such as removing the bias error by its nature, allowing fast and accurate identification through the bode graph, and the ability to make quick reviews of the system (whether it is stable or not via bode plot) [19].

System identification can be made in the frequency domain in parametric (Gray-box) or non-parametric (Black-box) manner.

Non-parametric System Identification:

It is a system identification process that does not require information, for example, the system structure, how many states there are, or which order can be expressed with the transfer function, etc. With this identification process, the system's bandwidth, time delay, system stability and system order can be identified.

Parametric system identification:

In the parametric system identification, the determining parameters are defined by adding the information of the system to the model. In this method, adding kinematic relations to the model can reduce the number of parameters to be defined and provide more accurate and faster results. This method is more suitable for designing a control system or a mathematical model.

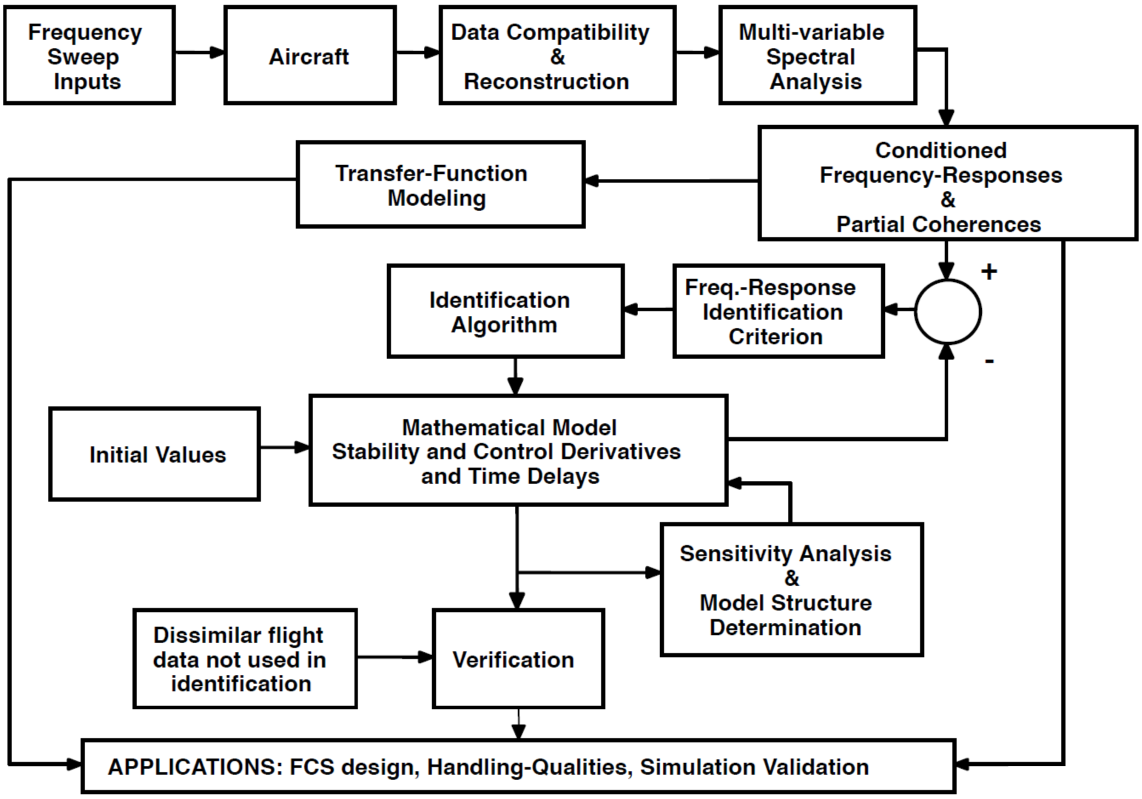

Frequency Sweep Inputs
Aircraft
Data Compatibility & Reconstruction
Multi-variable Spectral Analysis
Conditioned Frequency-Responses & Partial Coherences
Transfer-Function Modeling
Identification Algorithm
Freq.-Response Identification Criterion
+
-
Initial Values
Mathematical Model Stability and Control Derivatives and Time Delays
Sensitivity Analysis & Model Structure Determination
Dissimilar flight data not used in identification
Verification
APPLICATIONS: FCS design, Handling-Qualities, Simulation Validation

Figure 2-3 Frequency-Response Identification Method [1]

The summary of the system identification process in the frequency domain is shown in Figure 1. As can be seen in the image, the first step of the process is to generate the signal that can appropriately excite the frequency range of interest. The method to be applied here is the automatic application of the sine-sweep signal covering the frequency range of interest. In this way, it is seen that the coherence value of the data obtained is higher. "Coherence" is a metric that expresses the linear similarity between two signals. The closer the coherence function is to 1, the higher linear the relationship between the two signals and the lower the noise ratio. "Coherence" values above 0.7 are considered good, and above 0.6, system identification is considered sufficient for the procedure[1].

$$Frequency - Response = \qquad \hat{\text{H}} = \frac{|G_{xy}|}{|G_{xx}|} \tag{2.39}$$

$$Coherence = \qquad \widehat{\gamma_{xy}^{2}} = \frac{|G_{xy}|^{2}}{|G_{xx}||G_{yy}|} \tag{2.40}$$

Time delay can be defined in the frequency domain as follows;

$$\Phi = -\omega\,\tau \tag{2.41}$$

Basically, time delays that caused by process delay rotor delays, actuator delays, motor delays, produce a linear roll-off in phase as a function of frequency. And, it can be determined, by looking at the slope of the curve at high frequency.

We can formulate a model with a time delay as follows:

$$\dot{x} = Fx + Gu(t - \tau) + bias \tag{2.42}$$

$$\dot{y} = Hx + ju(t - \tau) + yref \tag{2.43}$$

$$T(s) = (H[sI - F]^{-1}G + j)e^{\tau s} \tag{2.44}$$

In order to perform system definitions in the time domain, the data must be free from reference and bias errors. On the other hand, in the frequency domain, these bias errors disappear because they are static.

The use of physics-based models during the design of small-sized UAVs have significant deficiencies. For example, calculations of aerodynamic forces on the blade cannot give accurate results for these size of blade dimensions and these rotation speeds. Trying to obtain these data with CFD analysis is a process that will take a long time since all rotation speeds must be scanned. System identification process can be completed with a flight test of about a few hours. Since small scale multi-rotor systems are quite unstable systems, it is not possible to apply a sine-sweep signal without a controller. For this reason, it is necessary to define the system with low-gain controllers for these vehicles during identification. Then, system identification can be done through the inputs of the mixer matrix, which are outputs of the rate controllers.

Some of the steps to be considered for the system identification process are as follows [1, 19]:

- The frequency range of interest to the sine-sweep should be determined. This value can start with 0.1-0.15 Hz for small-scale UAVs and go up to 2-4 Hz.
- Recording time should be 4 or 5 times of the longest period of interest. For example, 40-50 seconds for the above situation.
- Since the definition of 4 channels will be made, the total flight takes 160-200 seconds.
- Since direct control cannot be made on the control surfaces, the model will be obtained with the actuator delays included in the model.

- In some cases, because the Kalman filter fuses the sensor measurements, it may cause inaccurate results in system identification when it is not well-designed. Therefore, it should be checked whether the Kalman filter has an effect, especially on low frequencies.
- The recording frequency is suggested to be around 25 times the highest frequency of interest.
- In order to cover the system model for the entire flight profile, the tests must be repeated in different trim conditions. In forward flight, 10 kts steps for low speeds and 20 kts steps for high speeds are considered adequate.
- Input rates for the test are recommended to be 20-30 degrees/s.
- It is recommended to do a doublet manoeuvre for the verification test. The Doublet manoeuvre consists of a two-way sharp step signal. With this signal, even if the input and output signals are noisy, high coherence value can be obtained.

It is always preferable to do the system identification process with low gain controllers over an open-loop system. However, if there is an obligation to perform closed-loop testing as in the above example, it should be considered that low coherence can be caused at low frequencies.

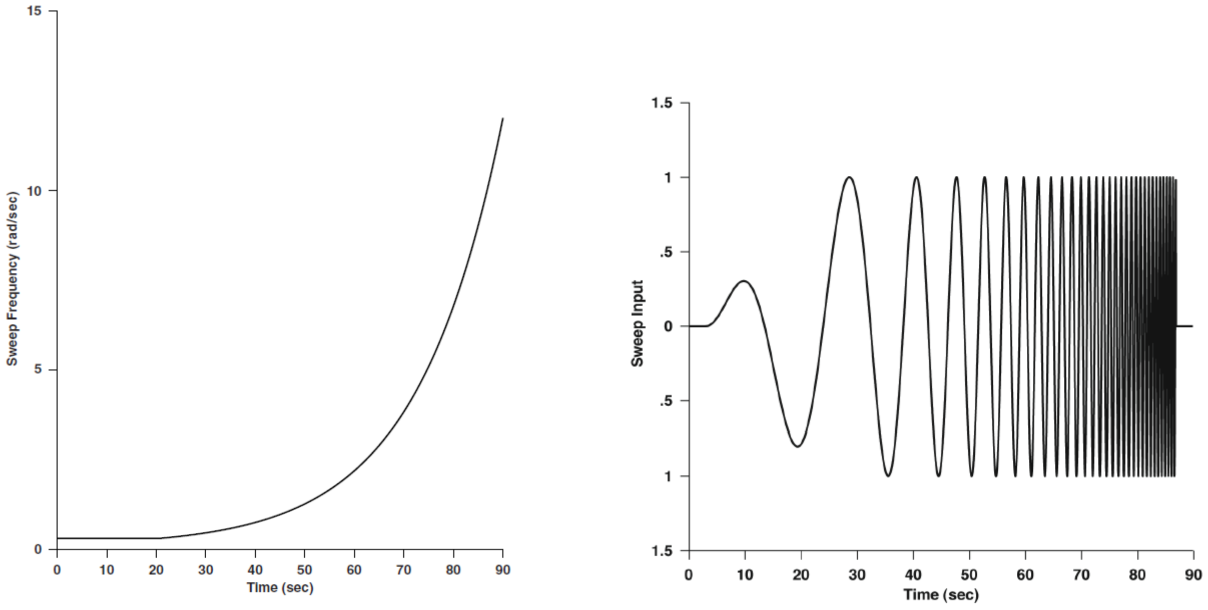


Figure 2-4 Sine-sweep signal [1]

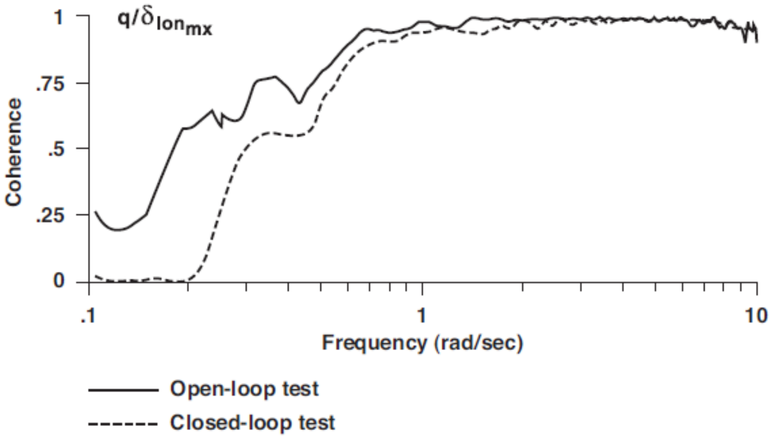


Figure 2-5 Coherence of open-loop and closed loop tests[1]

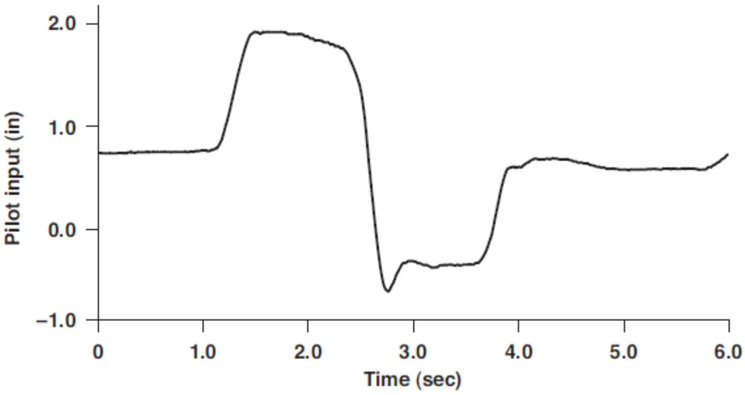


Figure 2-6 Doublet manoeuvre[1]

2.3.1. System Identification process

The roll-rate response of the aircraft, which is a one DOF transfer function, can be defined by the following transfer function.

$$\dot{p} = L_p p + L_\delta \delta \qquad (2.45)$$

After Laplace transformation;

$$sp(s) - p(0) = L_p p(s) + L_\delta \delta \qquad (2.46)$$

$$\frac{p(s)}{\delta(s)} = \frac{L_\delta}{s - L_p} \quad ; \quad p(0) = 0 \qquad (2.47)$$

Foruier Transform = Laplace Transform $s => j\omega$:

$$H(j\omega) = H(j2\pi f) = \frac{L_\delta}{j\omega - L_p} \tag{2.48}$$

$$H_{db} = 20\log_{10}|H(j\omega)| \tag{2.49}$$

$$H_{deg} = phase\ of\ H(j\omega)\ in\ degrees \tag{2.50}$$

Although the amplitude graph is similar for stable and unstable systems, the stability of the system can be understood by looking at the phase graph. If the phase graph increases with the frequency, this indicates that the system is unstable and L_p will take a positive value. In stable systems, the phase graph shows a decreasing trend, and in this case, the L_p value becomes negative. System modelling can be done without the need for an assumption in transfer function modelling. In this modelling approach, all parameters can be obtained through the bode chart. As can be seen in the graphics below; low frequencies amplitude $\left|\frac{L_\delta}{L_p}\right|$ gives the rate. This equation can be obtained by making ω = 0 in the transfer function in the frequency domain. The magnitude of the L_p is equal to the frequency at the point where the phase graph reaches a 45-degree slope, or the amplitude graph breaks down.

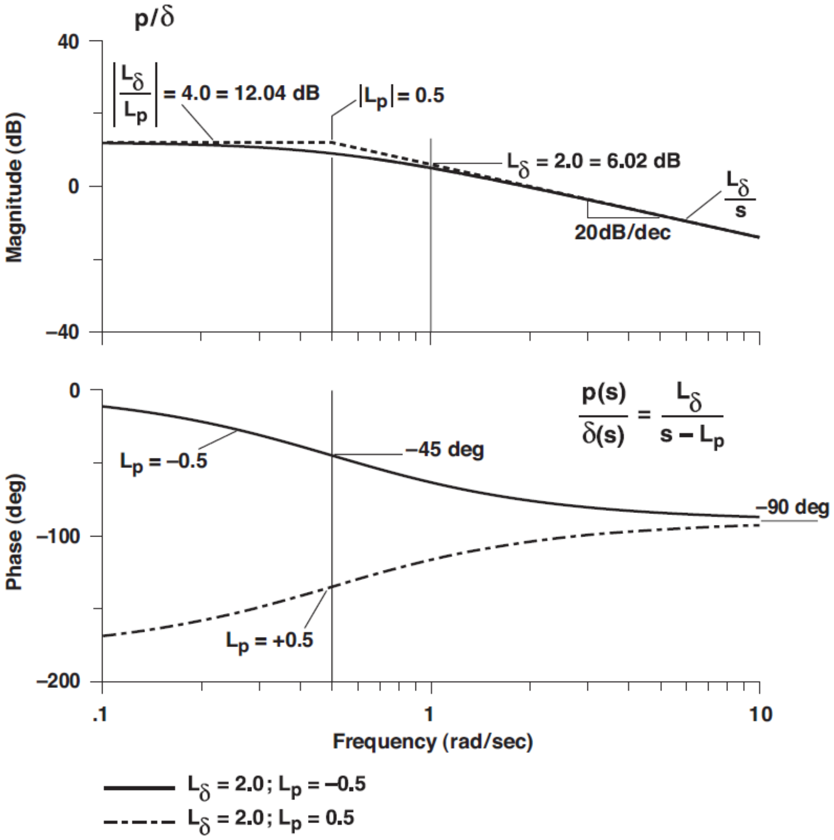


Figure 2-7 Bode plot of a SISO transfer function[1]

As briefly mentioned in the previous sections, coherence is a metric for defining the linear relationship between signals.

$$\gamma_{xy}^2 = \frac{\left|G_{xy}\right|^2}{G_{xx}G_{yy}} \tag{2.51}$$

Coherence can be affected by non-linearity of the system, disturbances and noise on the system. The effect of system noise on coherence can be expressed as follows.

$$\gamma_{xy}^2 = \frac{1}{1 + \, \mathcal{E}} \tag{2.52}$$

Here, the parameter refers to the noise/signal ratio. This parameter is expected to be less than 0.3, which ensures the coherence value to be around 0.77. To expand further the noise ratio, for example, during a flight around the trim, the angular rates of the system can change ±2°/s. That may cause the noise on the system to be relatively close with the measurements made during the system identification process. Therefore, the fact that this excitation is around ±10/20°/s during the tests for system identification will ensure that the effect of this noise is low, and small-scale UAV platforms can easily provide this requirement.

The effects of gust and non-linearities or non-diagonal terms of the system can also be reduced by windowing during FFT. Thanks to the 80% overlap windowing, the effect of random errors at low frequencies can be reduced. Wide windows are effective at low frequencies, and narrow screens are effective at higher frequencies [19].

$$\varepsilon_r = \left(\sqrt{0.50} \right) \frac{\left[1 - \gamma_{xy}^2\right]^{1/2}}{\left|\gamma_{xy}\right|\sqrt{2n_d}} \tag{2.53}$$

$$n_d = \frac{T_{rec}}{T_{win}} = Number\ of\ independet\ time\ history\ averages$$

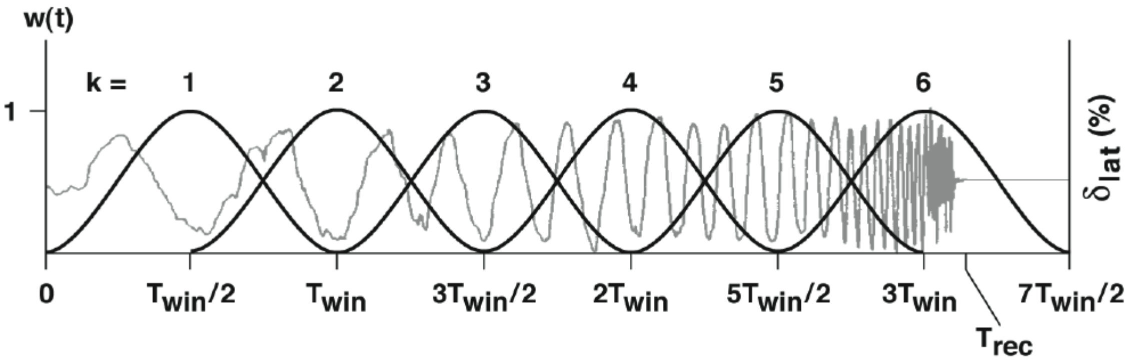


Figure 2-8 Overlapped windowing of flight data[1]

A flat amplitude and phase responses can be seen in the frequency response at very high noise. The reason for this misleading effect is the noise in the feedback signal itself. The amplitude here will be equal to $-\frac{1}{Gc}$.

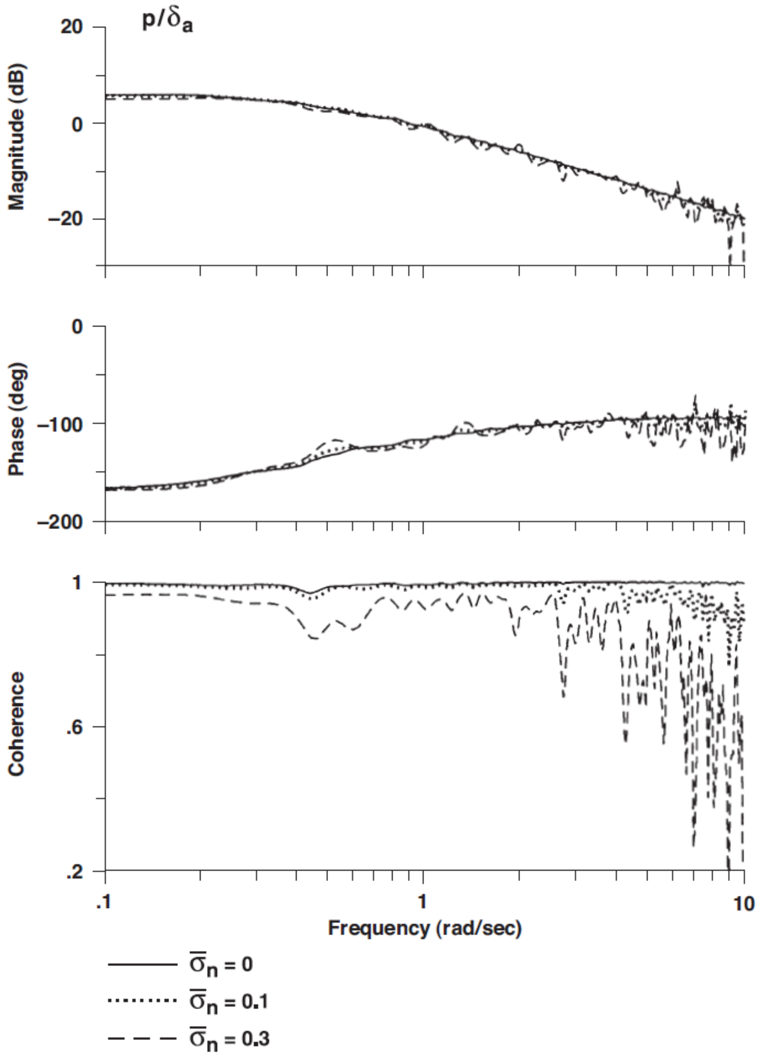


Figure 2-9 Effect of low noise ratios on Bare Airframe ID[1]

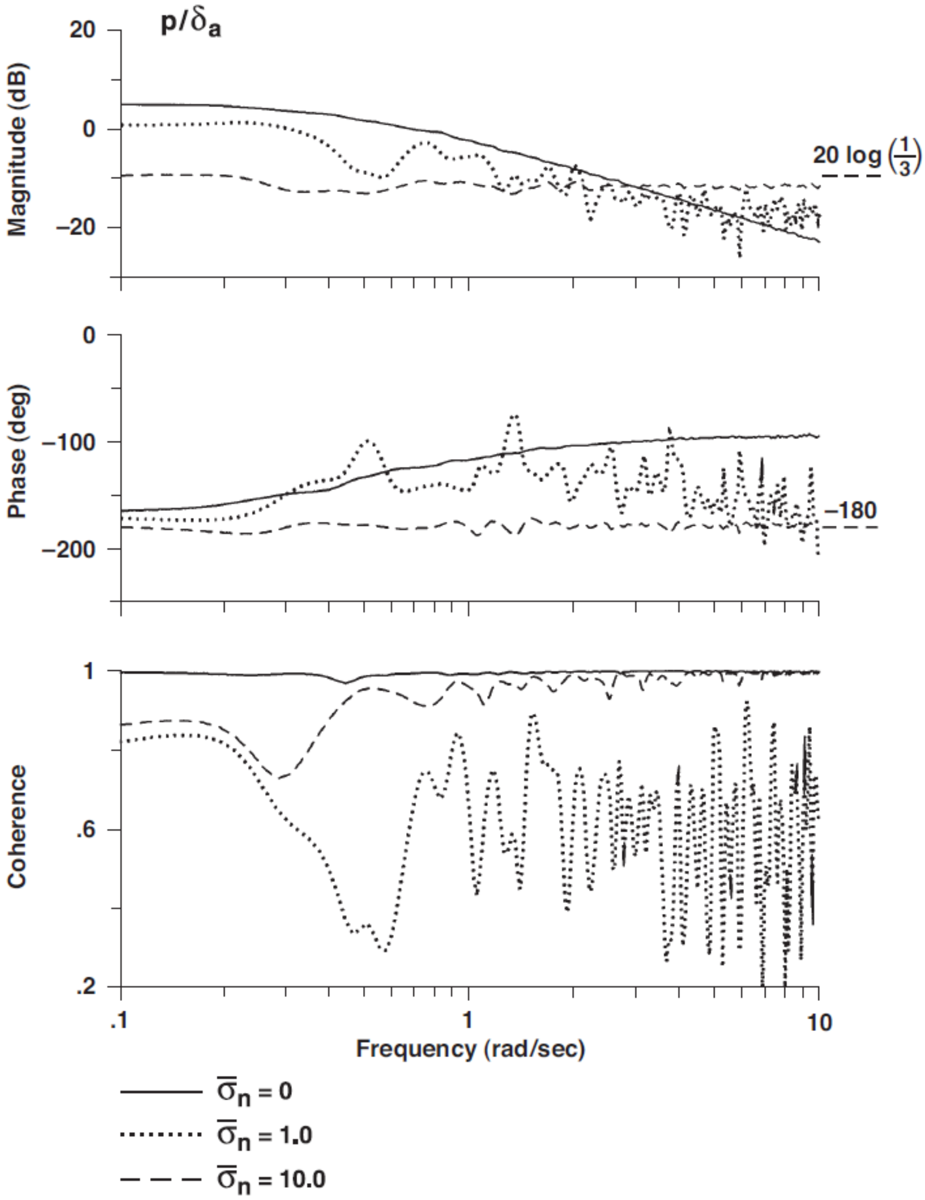


Figure 2-10 Effect of high noise ratios on bare airframe[1]

CIFER NAVFIT program defines transfer function parameters with non-linear (Rosenbrock) least-squares minimization cost function [20]:

$$J = \frac{20}{n_w} \sum_{\omega_1}^{\omega_{n_\omega}} W_\gamma \left[W_g (|T_c| - |T|)^2 + W_p (\angle T_c - \angle T)^2 \right] \tag{2.54}$$

Where;

n_ω = number of frequency points

$\omega_1, \omega_{n_\omega}$ = starting and ending frequencies of fit

$|\quad|$ = magnitude (dB)at each frequency ω

$\angle$ = phase(deg) at each frequency ω

$W_r =$ coherence weighting to emphasize most reliable data

$W_g =$ weighting on gain error;default=1

$W_p =$ weighting on phase error;default $= 0.01745 => (7.57deg-\ error\text{: } 1\ dB\ mag - error)$

If the cost value is J <100, it is acceptable, if J <50, it indicates a very realistic definition.

Another method of system identification through CIFER is to define through state-space matrices.

Here, state-space matrices are defined in the following format:

$$M\dot{x} = Fx + Gu(t-\tau) \tag{2.55}$$

Here M, F, G and τ are unknown stability derivatives. Some terms here can be known from physical relationships and can be included in the matrix during definition.

The output format is also defined as follows:

$$\dot{x} = Ax + Bu(t-\tau), y = Cx + Du(t-\tau)$$

$$A = M^{-1}F, \quad B = M^{-1}G \tag{2.56}$$

$$C = H_0 + H_1M^{-1}F, \quad D = H_1M^{-1}G$$

In the system definition made from state-space matrices, the expanded form of SISO formulation is used directly.

$$J = \sum_{l=1}^{n_{TF}} \left\{ \frac{20}{n_w} \sum_{\omega_1}^{\omega_{n_\omega}} W_\gamma \left[W_g(|T_c| - |T|)^2 + W_p(\angle T_c - \angle T)^2 \right] \right\} \tag{2.57}$$

Unlike the definition made with the transfer function in the cost function above, the parameters $\omega_1, \omega_{n_\omega}$ and W_r are uniquely defined for each transfer function.

In system definitions made over state-space, it is sufficient if the average cost, $J_{ave} = \frac{J}{n_{TF}}$, is less than 100, and it is considered perfect if it is less than 50. It is sufficient for each transfer function to be less than 200.

$$J(\partial\Theta) = \left(\frac{20}{n_w}\right) [(\varepsilon_0 + D\partial\Theta)^T W(\varepsilon_0 + D\partial\Theta) + H.O.T.] \tag{2.58}$$

$$H = \frac{\partial^2 J}{\partial\Theta} \cong \left(\frac{20}{n_w}\right) [2D^I W D] \qquad (CR_i)_{CIFER} \equiv 2\sqrt{(H^{-1})_{ii}} \approx \sigma_i \tag{2.59}$$

Cramer-Rao Bound and insensitivity ratios are other parameters that are obtained and should be considered during the system identification process.

$$\overline{CR_i} = \left|\frac{CR_i}{\theta_i}\right| \times 100\% \quad where\ \theta_i = converged\ ID\ parameter \tag{2.60}$$

If $\overline{CR_i} \leq 20\%$, it is ideal for system identification, lower than 40% is sufficient. A high value of $\overline{CR_i}$ increases parameter insensitivity and/or parameter correlation, making that parameter ineffective on the cost function.

$$I_i = [H_{ii}]^{-\frac{1}{2}}, \quad \overline{I_\imath} = \frac{I_i}{\theta_i} \times 100\% \quad ve\,\overline{I_\imath} \leq \overline{CR_\imath} \tag{2.61}$$

For a correct identification, it is recommended to fulfill $\overline{I_i} \leq 10\%$ condition.

The increase in parameter correlation indicates that two or more parameters have a very similar effect on the cost function, which causes the parameters not to be defined independently. If a parameter has a very high Cramer-Rao ratio and insensitivity, this parameter can be deleted from the system model.

3. MODELLING

3.1. Introduction

Multicopters are very popular in research and industrial areas because of its simplicity and effectiveness. In many research areas such as image-processing, control, artificial intelligence etc. quadcopters have a significant role. Despite its popularity, many of these research areas don't require accurate system model to work. Different from these research areas, model predictive control approach highly dependent on model accuracy. Higher fidelity models require more computational power, and they are hard to implement on real applications. In this thesis, the system identification method has been used to model the quadcopter system. The system identification method gives the accurate linear model of the system. Before proceeding the details of the system identification method, basic concept of the quadcopters has been expounded in this section.

In this work, a quadcopter has been used as a test platform. The test platform has an X type of configuration shown in Figure 3-3. In this configuration, the motor rotations have set to CCW, CW, CCW, CW in order.

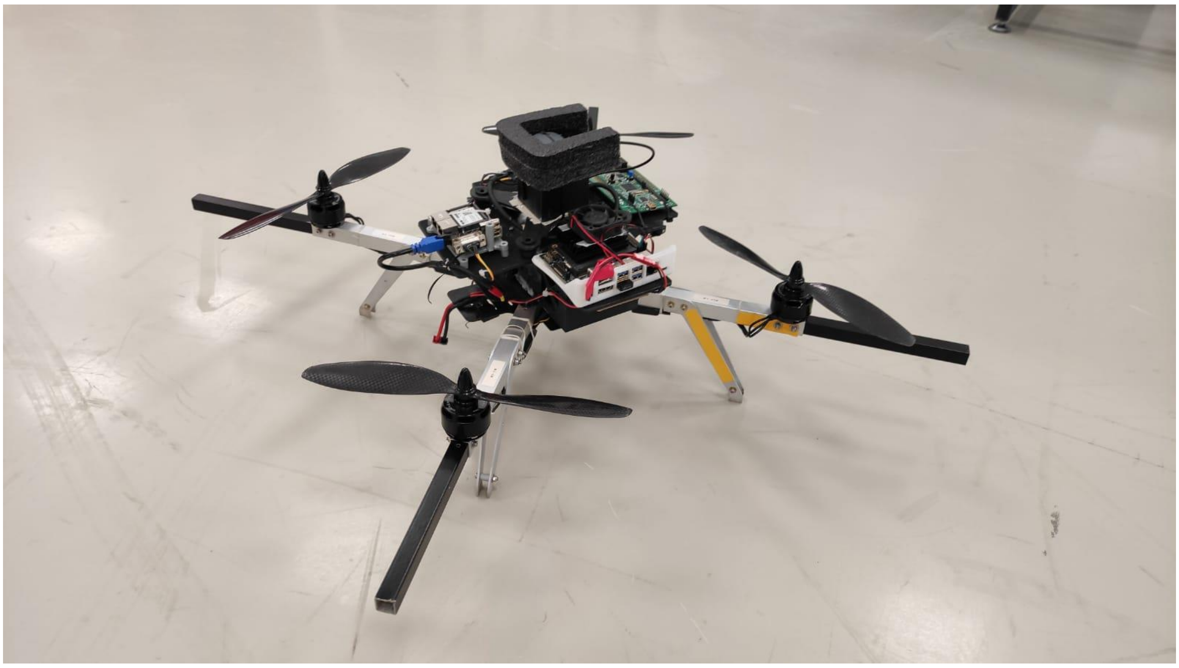


Figure 3-1 Test Platform

Test platform specifications are detailed below:

Table 3-1 Test Platform Specifications

Total Mass	Wheelbase	Motor	Propeller	Navigation	Hardware	ESC	Battery
2.65 kg	0.7 m	T-motor U3 KV700	Carbon Fibre Propeller 11x4.7	Advanced Navigation Spatial	STM32F04 + Jetson Nano	T-Motor Air 40A	Tattu 4S1P 9000 mah

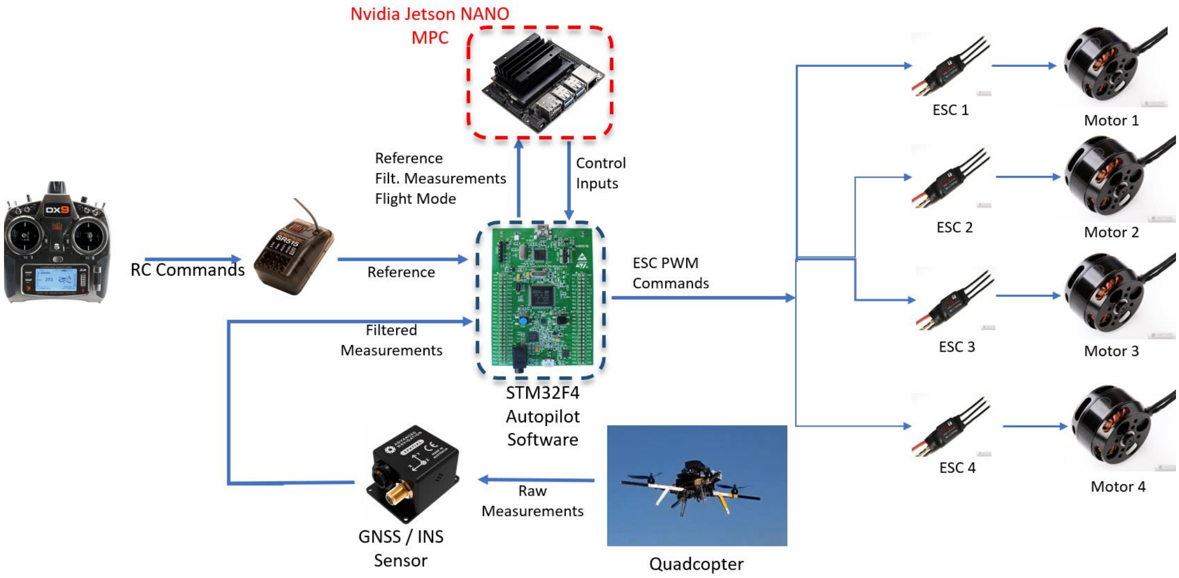


Figure 3-2 System Overview

Table 3-2 Motor Specifications

	KV	700
	Internal resistance	50mΩ
	Max. efficiency current	(4-10A)>82%
	Max Continuous Power(W)180S	500W
	Max Continuous current(A)180S	25A
	No.of Cells(Lipo)	3-4S
	ldle current(10)@10v(A)	0.5A
	Weight(g)	97g
	Motor Dimension(Dia.*Len)	Φ41.8×30.75mm
	Shaft Diameter	4mm
	Configuration	12N14P

Table 3-3 Propeller Specifications

	Diameter	11inch
	Pitch	4.7 inch
	Hub thickness	8 mm
	Shaft diameter	6 mm
	Weight(each)	15 g

Table 3-4 ESC Specifications

	Continuous Current	40A
	Peak Current	60A
	BEC	NO
	Lipo	2-6S
	Weight	26 g

Table 3-5 Navigation System

	Horizontal Position Accuracy	2.0 m
	Vertical Position Accuracy	3.0 m
	Horizontal Position Accuracy (with RTK)	0.02 m
	Vertical Position Accuracy (with RTK)	0.03 m
	Horizontal Position Accuracy (Kinematica Post Processing)	0.01 m
	Vertical Position Accuracy (Kinematica Post Processing)	0.02 m
	Velocity Accuracy	0.05 m/s
	Roll & Pitch Accuracy	0.1 °
	Heading Accuracy (Dynamic with GNSS)	0.2 °
	Heading Accuracy (Magnetic Only)	0.8 °
	Roll & Pitch Accuracy (Kinematica Post Processing)	0.04 °
	Heading Accuracy (Kinematica Post Processing)	0.08 °
	Heave Accuracy	5 % or 0.05 m (whichever is greater)
	Orientation Range	Unlimited
	Hot Start Time	500 ms
	Internal Filter Rate	1000 Hz
	Output Data Rate	Up to 1000 Hz
	Latency	0.4 ms

Table 3-6 STM32F04 Discovery Board Specifications

	Microcontroller	32-bit Arm® Cortex®-M4
	Flash memory	1-Mbyte
	RAM	192-Kbyte
	Accelerometer	ST MEMS 3-axis
	Power Supply	3 V and 5 V

Table 3-7 NVIDIA Jetson Nano Specifications

	GPU	128-core NVIDIA Maxwell™ architecture-based GPU
	CPU	Quad-core ARM® A57
	Video	4K @ 30 fps (H.264/H.265) / 4K @ 60 fps (H.264/H.265) encode and decode
	Camera	MIPI CSI-2 DPHY lanes, 12x (Module) and 1x (Developer Kit)
	Memory	4 GB 64-bit LPDDR4; 25.6 gigabytes/second
	Connectivity	Gigabit Ethernet
	OS Support	Linux for Tegra®

Table 3-8 Battery Specifications

	Capacity	9000mAh
	Voltage	14.8V
	Max Continuous Discharge	25C
	Max Burst Discharge	50C
	Weight	733g
	Dimensions	207*73*26 mm
	Charge Rate	1-3C Recommended, 5C Max

3.2. Coordinate Frames

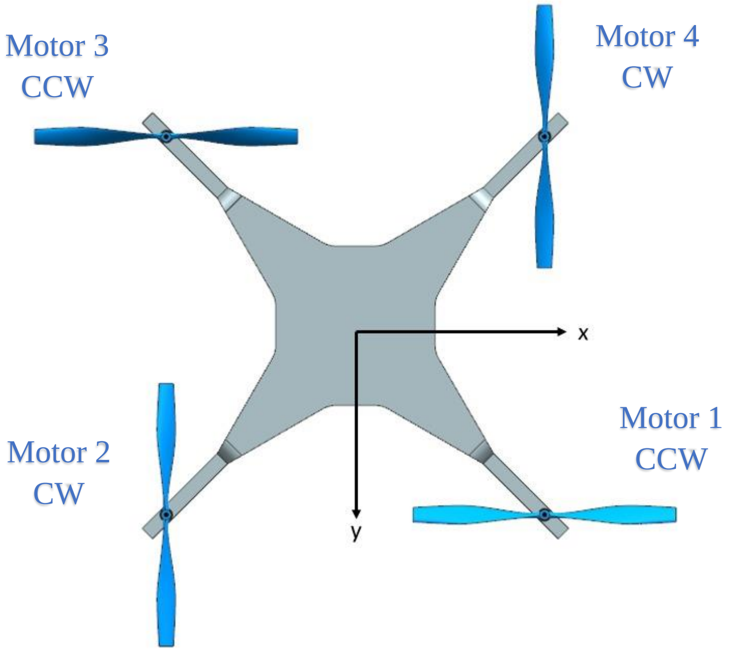


Figure 3-3 Quadcopter Configuration

The inertial frame C_I, also known as earth fixed frame, uses North-East-Down convention. In this frame, origin stays at the home location. The x-axis points North, y-axis points east, and the z-axis points to the center of the earth.

The vehicle frame C_v, also known as NED (North-East-Down) frame, the origin of this frame is at the center of mass of the UAV, but its axes are aligned with the inertial frame. So, its x-axis points north, y-axis points east and the z-axis points the center of the earth or down in another words.

The body frame C_B has the origin at the center of mass of the UAV, uses x-axis as the forward direction, the z-axis as down and the y-axis points right-side of the UAV [21].

States of the UAV’s

$x =$ the inertial position of the UAV towards x_I in C_I

$y =$ the inertial position of the UAV towards y_I in C_I

$z =$ the inertial position of the UAV towards z_I in C_I

$u = the\ body\ frame\ velocity\ of\ the\ UAV\ measured\ along\ x_b$ in C_b

$v =$ the body frame velocity of the UAV measured along y_b in C_b

$w =$ the body frame velocity of the UAV measured along z_b in C_b

$\phi =$ the roll angle of the UAV defined with respect to C_2 (3.1)

$\theta =$ the pitch angle of the UAV defined with respect to C_1

$\psi =$ the yaw angle of the UAV defined with respect to C_v

$p =$ the roll rate of the UAV measured along $x_b\ in\ C_b$

$q =$ the pitch rate of the UAV measured along y_b in C_b

$r =$ the yaw rate of the UAV measured along z_b in C_b

$h =$ the altitude of the aircraft measured along z_v in C_v

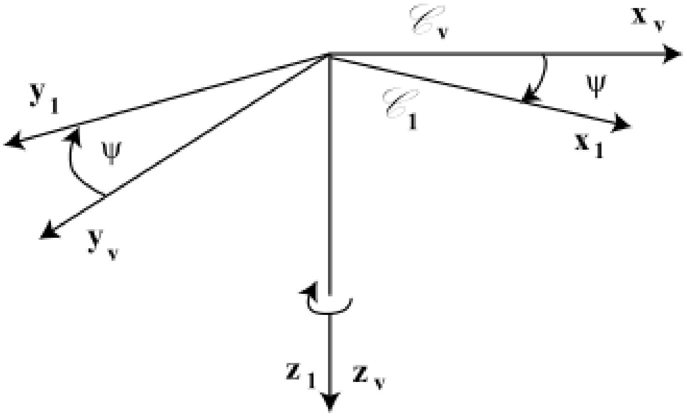


Figure 3-4 Yaw Angle Definiton [21]

$$R_{yaw}(\psi) = R_{v\to 1} = \begin{pmatrix} cos(\phi) & sin\ (\phi) & 0 \\ -sin\ (\phi) & cos\ (\phi) & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (3.2)$$

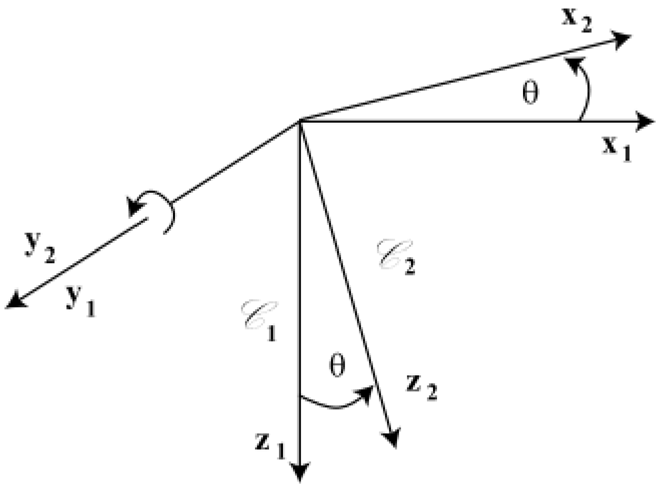


Figure 3-5 Pitch Angle Definition [21]

$$R_{pitch}(\theta) = R_{1\to 2} = \begin{pmatrix} cos(\theta) & 0 & -sin\,(\theta) \\ 0 & 1 & 0 \\ sin\,(\theta) & 0 & cos\,(\theta) \end{pmatrix} \qquad (3.3)$$

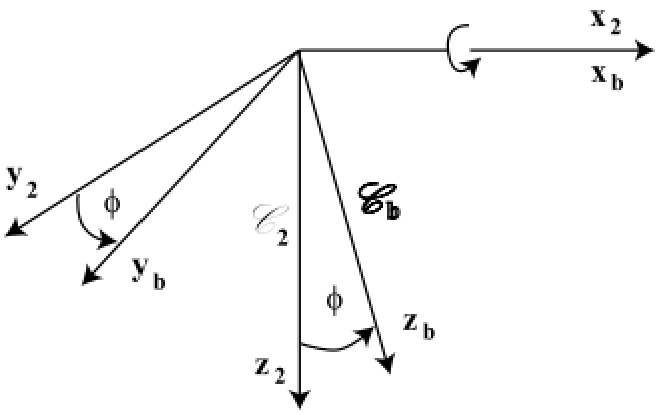


Figure 3-6 Roll Angle Definition [21]

$$R_{pitch}(\theta) = R_{1\to 2} = \begin{pmatrix} 1 & 0 & 0) \\ 0 & cos(\phi) & sin\,(\phi) \\ 0 & -sin\,(\phi) & cos\,(\phi) \end{pmatrix} \qquad (3.4)$$

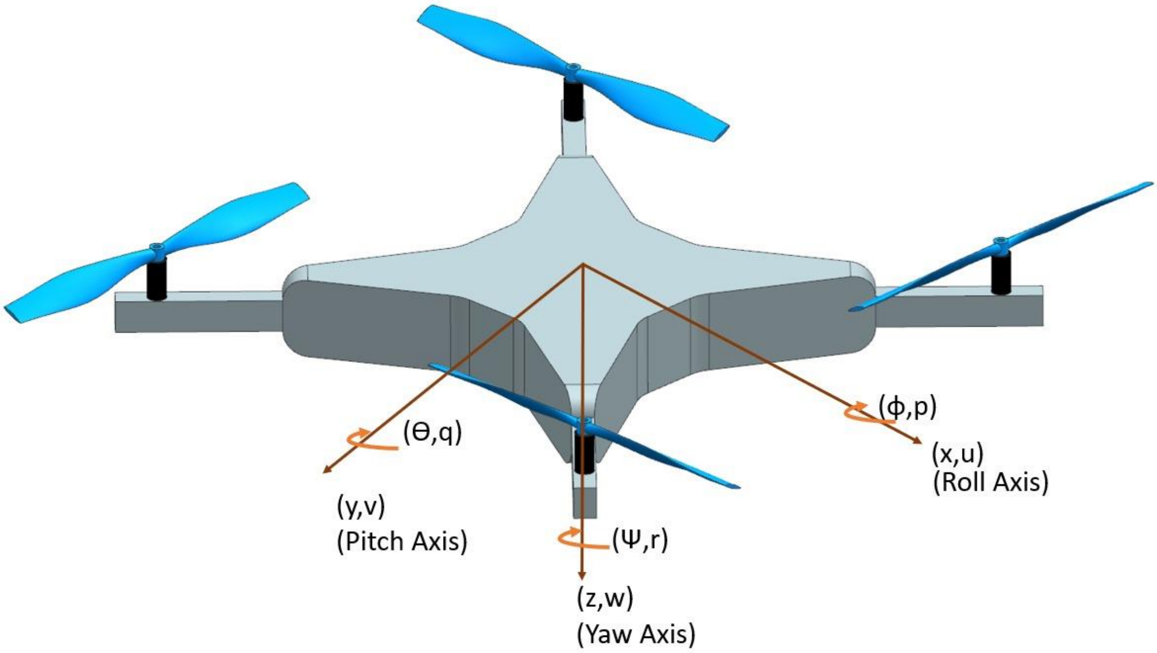


Figure 3-7 Axes Definitions and Euler Angles

3.3.System Identification Overview

System identification, in summary, is the process of obtaining the mathematical model of the system through movement measurements. The system identification process can be implemented efficiently and quickly also the identified model of the systems gives the closest linear approach to the real system. System identification can be done by using the time or frequency responses of the system. In this thesis, system identification is made on the system responses received in the frequency domain using CIFER software [1].

The diagram of the system identification process, performed over the frequency responses of the system, is shown in Figure 3-8.

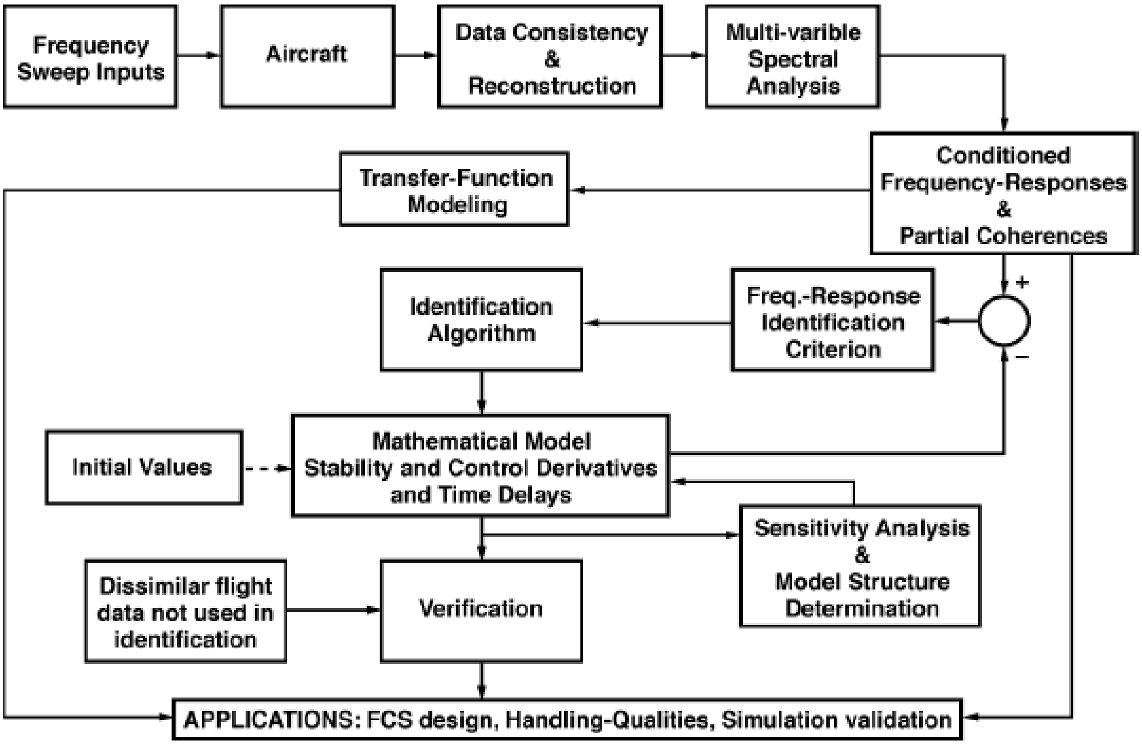

Frequency Sweep Inputs
Aircraft
Data Consistency & Reconstruction
Multi-varible Spectral Analysis
Conditioned Frequency-Responses & Partial Coherences
Transfer-Function Modeling
Identification Algorithm
Freq.-Response Identification Criterion
+
−
Initial Values
Mathematical Model Stability and Control Derivatives and Time Delays
Sensitivity Analysis & Model Structure Determination
Dissimilar flight data not used in identification
Verification
APPLICATIONS: FCS design, Handling-Qualities, Simulation validation

Figure 3-8 Frequency-Response Identification Method[1]

3.3.1. CIFER Software

CIFER, developed by the U.S Army Aviation Directorate, is a program that can make system identification from the frequency responses. Thanks to the programs inside, CIFER can make system identification quickly and with high accuracy. The schema of the programs in CIFER can be seen in Figure 3-9.

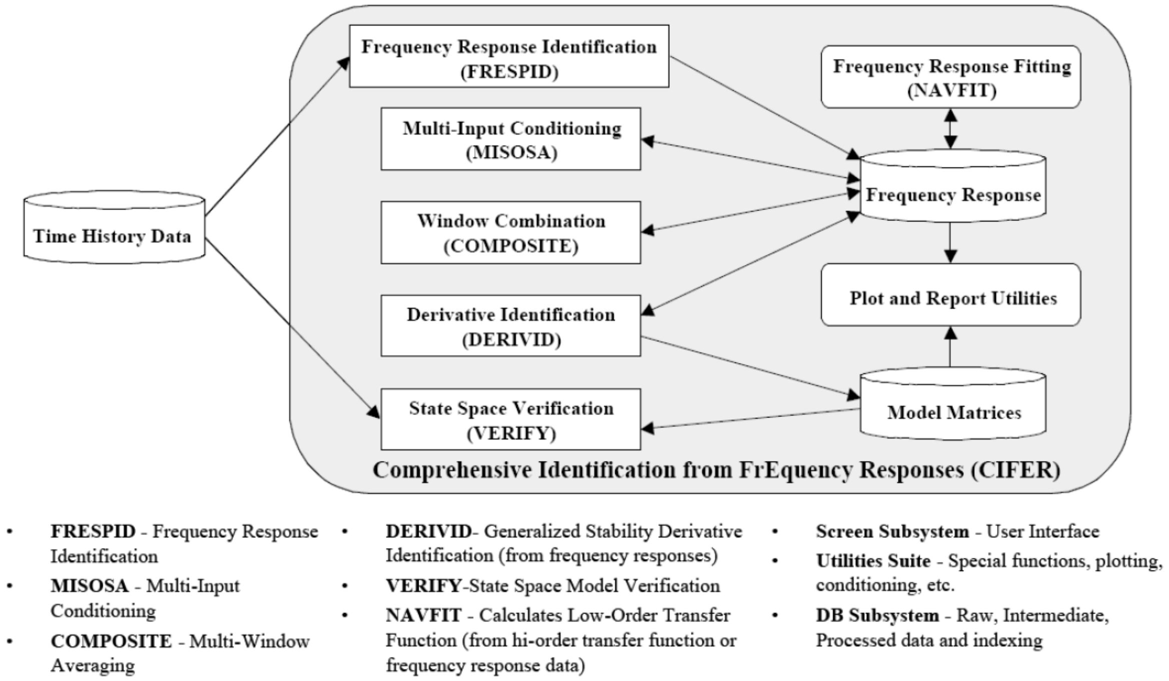
During this study, FRESPID, COMPOSITE, NAVFIT and DERIVID functions were used.

FRESPID: It is used to obtain frequency responses from time-dependent data.

COMPOSITE: It enables frequency responses to be weighted with different screenings (windowing).

NAVFIT: It allows to obtain a lower-level system model from the frequency response or a high-level transfer function.

DERIVID: Defines the parameter from the frequency response from the state space model.


Frequency Response Identification (FRESPID)
Multi-Input Conditioning (MISOSA)
Window Combination (COMPOSITE)
Derivative Identification (DERIVID)
State Space Verification (VERIFY)
Time History Data
Frequency Response Fitting (NAVFIT)
Frequency Response
Plot and Report Utilities
Model Matrices
Comprehensive Identification from FrEquency Responses (CIFER)

- **FRESPID** - Frequency Response Identification
- **MISOSA** - Multi-Input Conditioning
- **COMPOSITE** - Multi-Window Averaging
- **DERIVID**- Generalized Stability Derivative Identification (from frequency responses)
- **VERIFY**-State Space Model Verification
- **NAVFIT** - Calculates Low-Order Transfer Function (from hi-order transfer function or frequency response data)
- **Screen Subsystem** - User Interface
- **Utilities Suite** - Special functions, plotting, conditioning, etc.
- **DB Subsystem** - Raw, Intermediate, Processed data and indexing

Figure 3-9 CIFER Organization Scheme[1]

3.3.2. System Identification Process for Multicopters

The development of a six-degree of freedom model of a multi-rotor model requires less workload than other aircraft since all transport and control elements are identical rotors. Still, the main advantage of these systems is that they can be divided into four main structures under reasonable acceptances into linear or non-linear modelling environments. The main reason for this is that pitch and yaw movements show similar characteristics to each other since they generally have axial symmetry, or they are quite close even if they do not. Besides, the vertical and head angle channels can be solved more easily and separately from other channels [1, 18].

3.4.System Model

In this study, the system identification process was performed by linearizing the equations of motion obtained from Newton-Euler equations with assumptions in accordance with the dynamics of multirotor systems. Linearized versions of six-degrees of freedom equations of motion (Newton-Euler) for aircraft can be found in aviation books. Also, there are different linear systems with the differentiation of acceptances specific to the platform. For example, while these systems are often offered around the hover for a helicopter, full systems with 8-9 states depending on the flight speed and subsystems with 2,3 or 4 states are also widely available for fixed-wing aircraft. When the literature is reviewed, it is seen that most of the modelling studies for multi-rotor systems do not include aerodynamic forces. This leads to a decrease in the accuracy of the mathematical model.

In the example here, the state-space model presented for a quadcopter is essentially inclusive enough to model any multirotor.

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} X_u & 0 & 0 & 0 & 0 & 0 & 0 & -g & 0 \\ 0 & Y_v & 0 & 0 & 0 & 0 & g & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & L_v & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ M_u & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & N_r & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \\ \phi \\ \theta \\ \psi \end{bmatrix} + \begin{bmatrix} 0 & 0 & X_{\delta_{ele}} & 0 \\ 0 & Y_{\delta_{ail}} & 0 & 0 \\ Z_{\delta_{thr}} & 0 & 0 & 0 \\ 0 & L_{\delta_{ail}} & 0 & 0 \\ 0 & 0 & M_{\delta_{ele}} & 0 \\ 0 & 0 & 0 & N_{\delta_{rud}} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_{thr} \\ \delta_{ail} \\ \delta_{ele} \\ \delta_{rud} \end{bmatrix}$$

Figure 3-10 Quadcopter Model for System Identification[18]

Notation for State Space:

Table 3-9 Linear model notation

Direction	Forces	Moment	Directional Speed	Angular Speed	Euler Angles
x	X	L	u	p	θ
y	Y	M	v	q	ϕ
z	Z	N	z	r	ψ

When the system above is examined, the terms heave damping (Z_w), pitch and roll rate damping terms (M_q,M_θ,L_p,L_ϕ etc.) are missing compared to a conventional helicopter model. These terms have negligible values for multi-rotors, and it has been deemed appropriate to neglect them. However, in the tests carried out on the system, it was observed that the Z_w value had an undeniable effect, and this term was added to the model. On the other hand, the value of these terms around the hover comes entirely from rotor aerodynamics and can be calculated theoretically. By including the missing parameter, the state matrices can be edited as follows.

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} x_u & 0 & 0 & 0 & 0 & 0 & 0 & -g & 0 \\ 0 & Y_v & 0 & 0 & 0 & 0 & g & 0 & 0 \\ 0 & 0 & Z_w & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & L_v & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ M_u & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & N_r & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \\ \phi \\ \theta \\ \psi \end{bmatrix} + \begin{bmatrix} 0 & 0 & X_{\delta_{ele}} & 0 \\ 0 & Y_{\delta_{ail}} & 0 & 0 \\ Z_{\delta_{thr}} & 0 & 0 & 0 \\ 0 & L_{\delta_{ail}} & 0 & 0 \\ 0 & 0 & M_{\delta_{ele}} & 0 \\ 0 & 0 & 0 & N_{\delta_{rud}} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_{thr} \\ \delta_{ail} \\ \delta_{ele} \\ \delta_{rud} \end{bmatrix} \tag{3.5}$$

As can be seen mathematically, this system of equations can be divided into four independent channels. These are vertical (altitude), directional, longitudinal and lateral channels. These channels can be identified by independent tests.

3.4.1. Vertical (Altitude) Channel

Since it is a SISO system and does not show excessive instability, it can be driven without a controller, it is the channel where system diagnosis procedures can be applied easily.

Transfer function:

$$\dot{w} = Z_{\delta_{thr}} * \delta_{thr} + Z_w * w \tag{3.6}$$

$$w.s = Z_{\delta_{thr}} * \delta_{thr} + Z_w * w \tag{3.7}$$

$$w(s - Z_w) = Z_{\delta_{thr}} * \delta_{thr} \tag{3.8}$$

$$\frac{w}{\delta_{thr}}(s) = \frac{Z_{\delta_{thr}}}{(s - Z_w)} \tag{3.9}$$

The vertical channel represented by the equation above shows marginal stability and can be directly controlled by the pilot. Here, the relationship between pilot input and vertical speed is visible. With one pole and one gain, the above equation can be identified.

Test Input:

During flight test of the vertical channel, an automated sine-sweep signal has been applied to throttle channel. During the test, the pilot had to intervene to stabilize the platform. The shifts due to this intervention are seen in Figure 3-11. The specification of the sine-sweep tabulated below:

Table 3-10 Vertical channel sine-sweep input parameters

Starting Freq.	Ending Freq.	Number of Freq. Points	Test Period	Amplitude
0.1 Hz	15 Hz	8192	81.92s	± 50 PWM

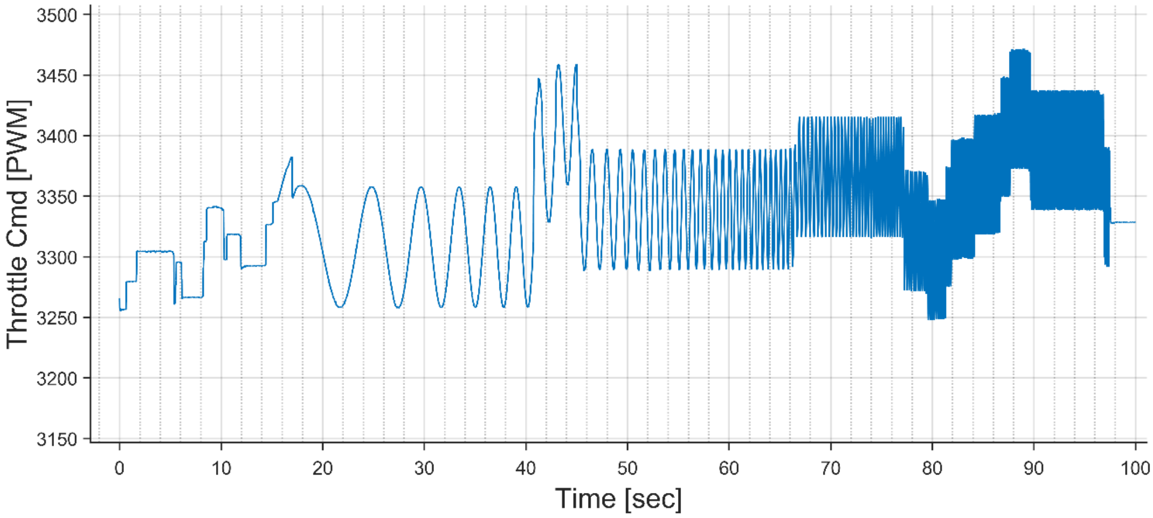


Figure 3-11 Throttle command during flight test

Test Measurements:

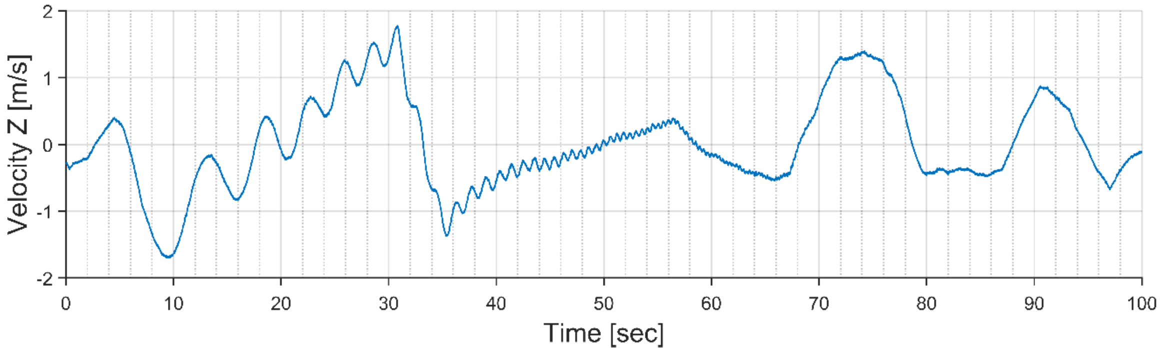


Table 3-11 Vertical velocity(w) response

Frequency Responses:

Frequency responses of the signals have been calculated via FFTs by FRESPID and composed via the COMPOSITE function of the CIFER. After determining the window sizes, CIFER can adjust adequate settings for FFT process. The windowing settings for FFTs are tabulated below:

Table 3-12 Windowing setup for FFT

Window Size (sec)	Number of input points	Number of output points	Minimum Frequency (Hz)	Maximum Frequency (Hz)
13	1300	748	0.0769	20
8	800	224	0.1250	20
6	600	424	0.1667	20
4	400	112	0.2500	20
2.1	210	302	0.4762	20

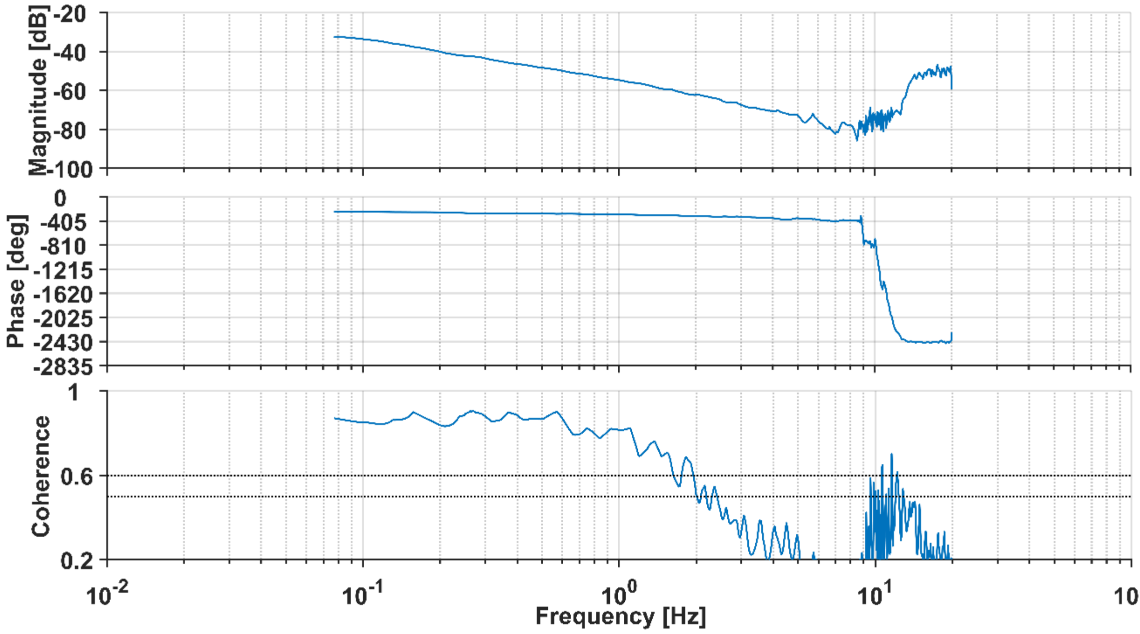


Figure 3-12 Frequency response of vertical velocity

As can be seen in the figures above the coherence values are acceptable 0.09 Hz to 2 Hz, that is why that particular frequency band has been selected for the identification process.

Table 3-13 Vertical channel identification frequency band

Transfer Function	**Start Frequency [Hz]**	**Stop Frequency [Hz]**
w/δ_{ele}	0.09091	2

After running DERIVID program with the dedicated model in Equation 3.8 the following results have been obtained.

Table 3-14 Identified parameters

PARAM	VALUE	CR_BOUND	CR-%	INSENS-%
Z_w	-0.2518	0.06854	27.22	12.75
$Z_{\delta_{thr}}$	-0.01229	5.24e-04	4.263	2.081
τ	0.06986	0.01081	15.47	7.407

Cost: 13.141

Results:

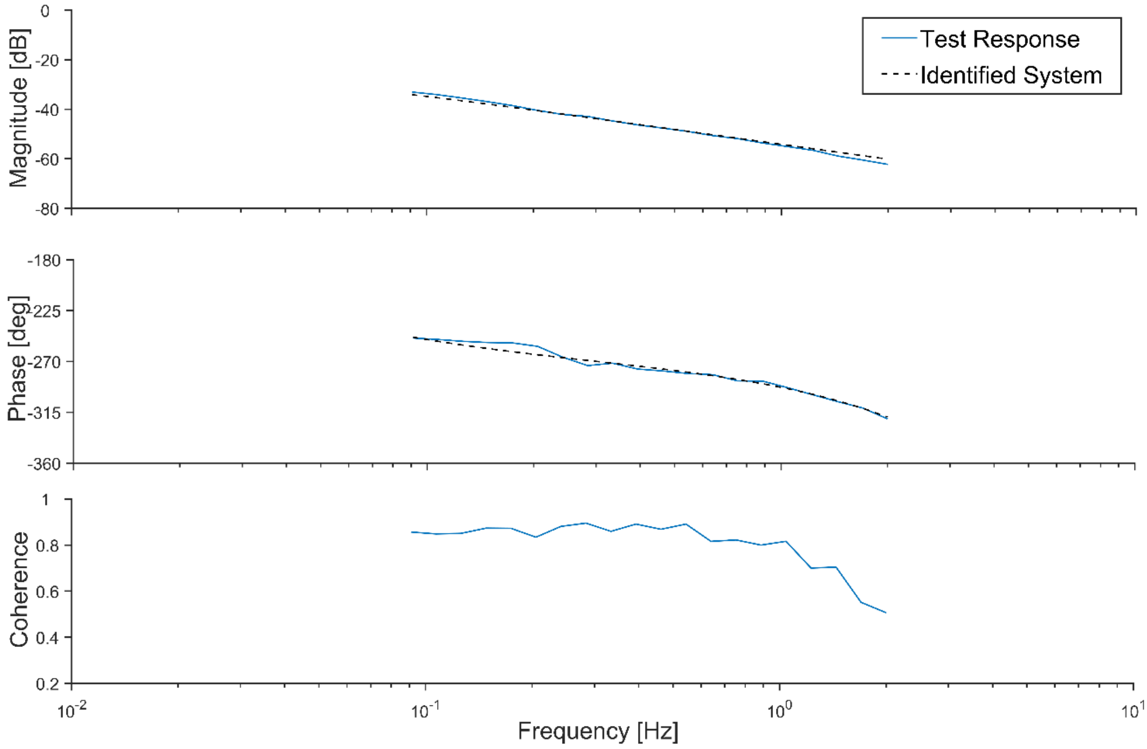


Figure 3-13 Comparison of ID system vs Real system in Frequency Domain

As it can be seen in Table 3-14, and Figure 3-13, system identification provided the sufficient performances criteria. %CR is lower than 40%, and insensitivity values are close to 10%. In the next step, step inputs have been applied in a flight test to validate the identified model. After obtaining an accurate result in the frequency domain, validation tests have been done. Throttle command that is shown in Figure 3-14 has been applied during the validation tests. The results have been shown in Figure 3-15, Figure 3-16. J_{RMS} value is around 1, so validation has been done successfully.

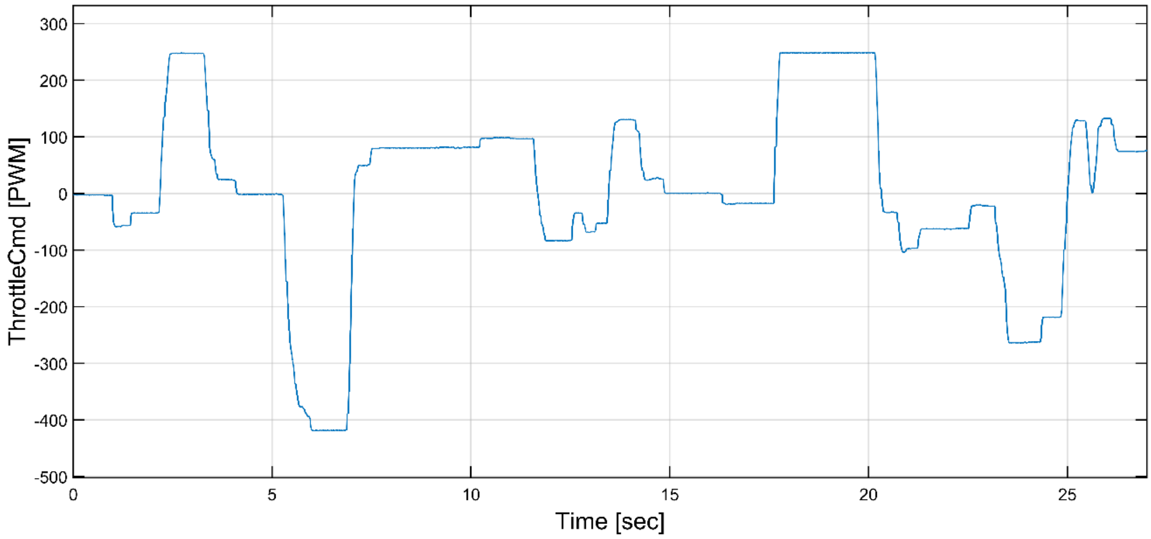


Figure 3-14 Throttle command around trim point to doublet manoeuvre

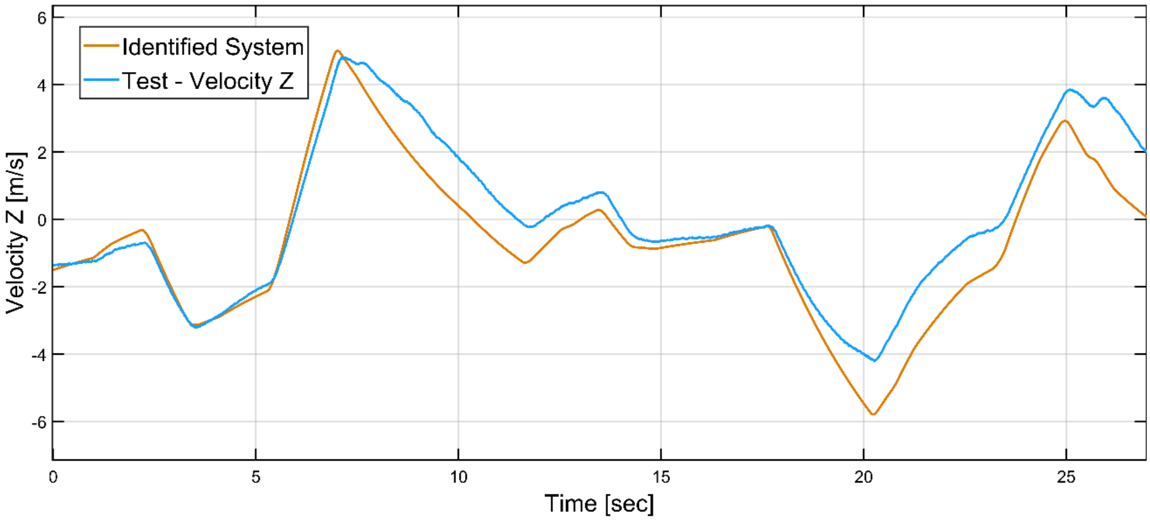


Figure 3-15 Vertical velocity responses to doublet manoeuvre

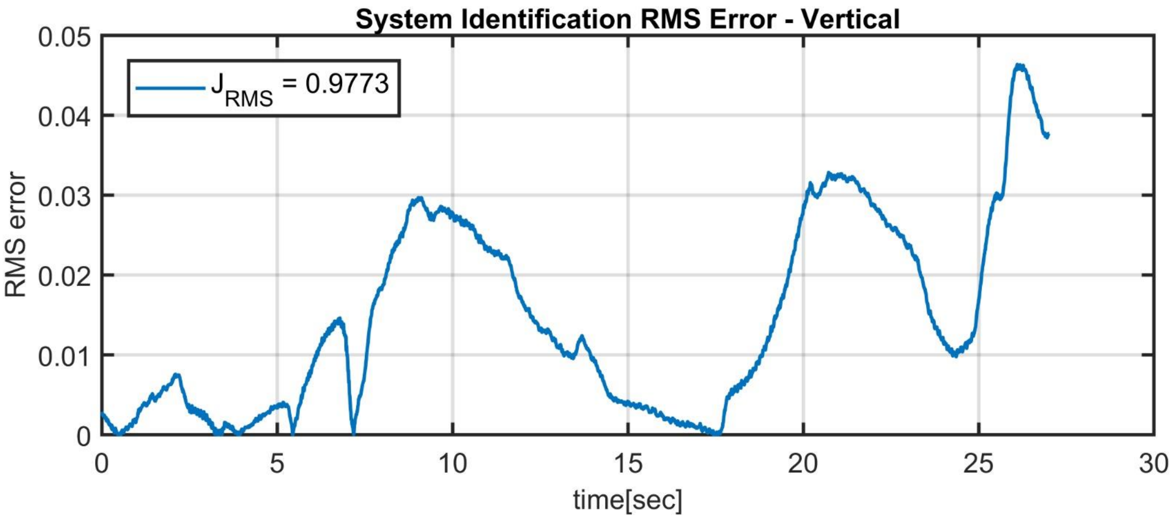


Figure 3-16 Vertical velocity responses RMS error

3.4.2. Directional Channel

The directional channel can be represented by a damping value and a control value. It has low damping and can be controlled directly by the pilot since it is not at very high speeds. The head angle channel also has a very similar structure to the vertical channel.

Transfer Function:

$$\begin{bmatrix} \dot{r} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} N_r & 0 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} r \\ \psi \end{bmatrix} + \begin{bmatrix} N_{\delta_{rud}} \\ 0 \end{bmatrix} \delta_{rud} \tag{3.10}$$

$$\dot{r} = N_{\delta_{rud}} * \delta_{rud} + N_r * r \tag{3.11}$$

After the Laplace transform, it can be expressed as follows.

$$\frac{r}{\delta_{rud}}(s) = \frac{N_{\delta_{rud}}}{(s - N_r)} \tag{3.12}$$

As can be seen in the transfer function above, it can be identified with one pole and one gain, like the vertical channel. Identification of these two channels is relatively easier than the horizontal channels.

Test Input:

During flight test of the directional channel, an automated sine-sweep signal has been applied to directional channel. The specification of the sine-sweep tabulated below. The generated sine sweep signal has shown in Figure 3-17. During identification, the directional command, which is the output of the rate controller and input of the mixer matrix, has been used to identify bare airframe.

Table 3-15 Directional channel sine-sweep input parameters

Starting Freq.	Ending Freq.	Number of Freq. Points	Test Period	Amplitude
0.1 Hz	8 Hz	16384	163.84s	± 0.4 rad/s

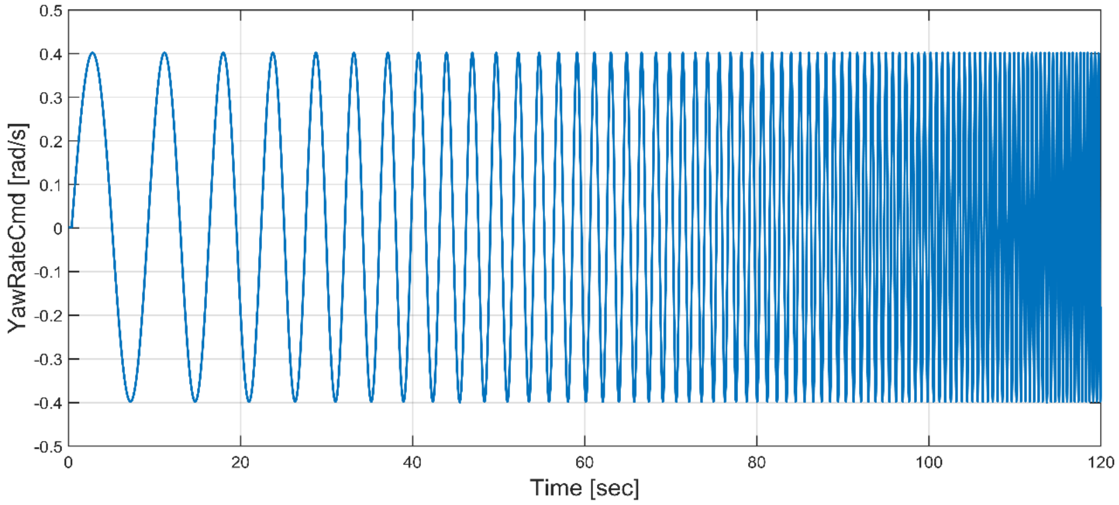


Figure 3-17 Yaw-rate command during flight test

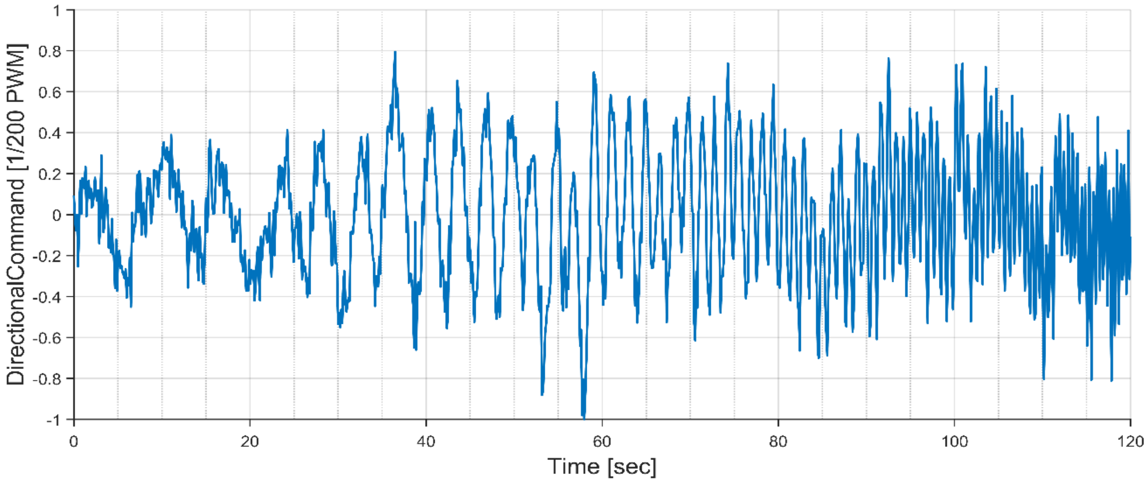


Figure 3-18 δ_{rud} during flight test

Test Measurement:

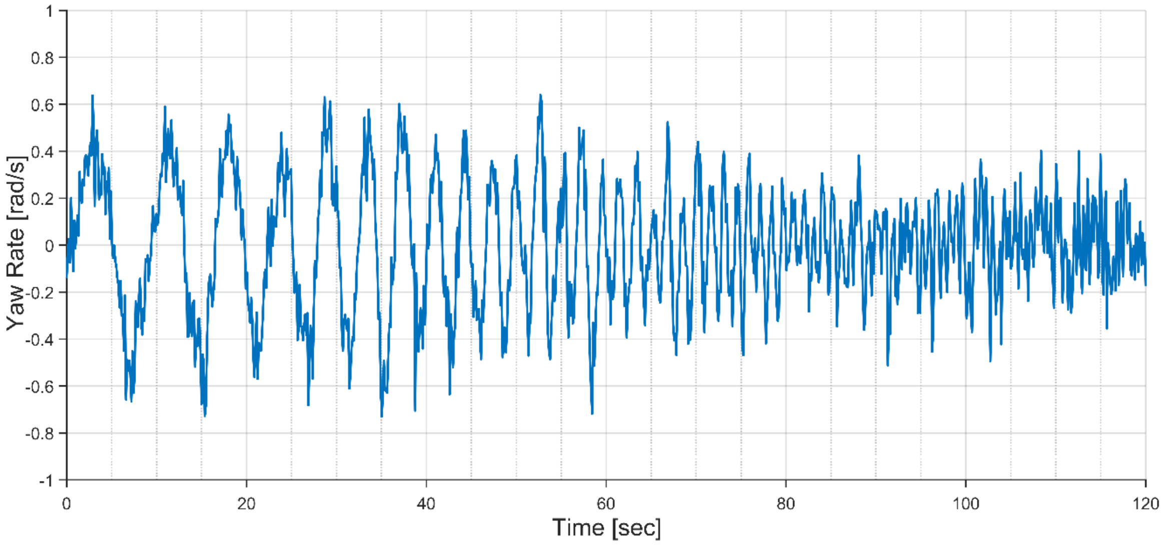


Figure 3-19 Yaw-rate during flight test

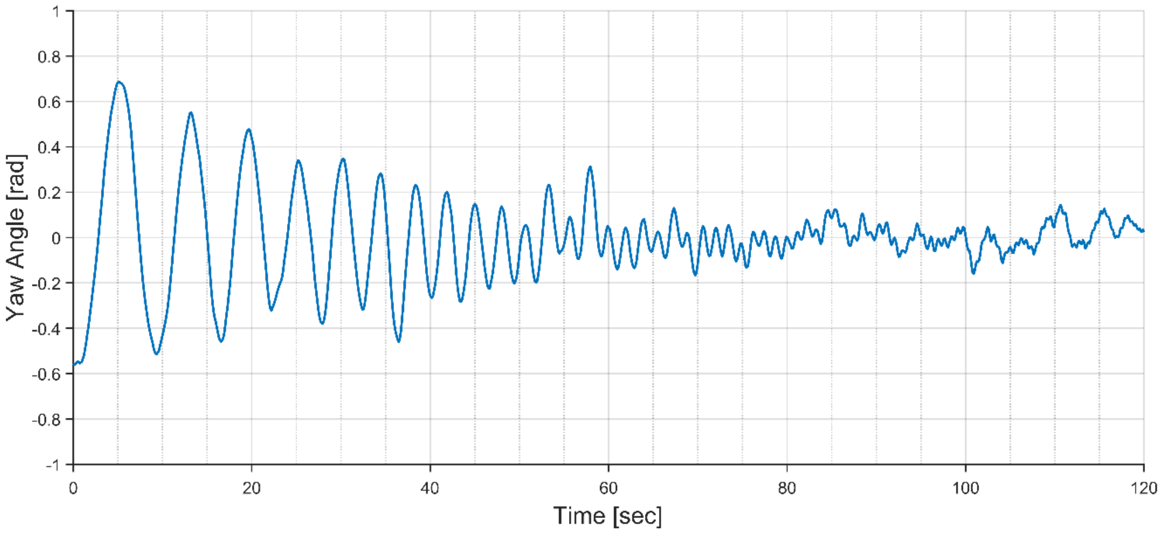


Figure 3-20 Yaw angle during flight test

The results of the test measurements have been shown in Figure 3-19 and Figure 3-20. After these tests, identification processes between directional input to yaw rate and yaw angle has been started.

Frequency Responses:

Frequency responses of the signals have been calculated via FFTs by FRESPID and composed via the COMPOSITE function of the CIFER. After determining the window sizes, CIFER can adjust adequate settings for FFT process. The windowing settings for FFTs are tabulated below:

Table 3-16 Windowing setup for FFT

Window Size (sec)	**Number of input points**	**Number of output points**	**Minimum Frequency (Hz)**	**Maximum Frequency (Hz)**
10	1000	1048	0.10000	8
7	700	324	0.14286	8
5	500	524	0.20000	8
3	300	212	0.33333	8
1.1	110	146	0.90909	8

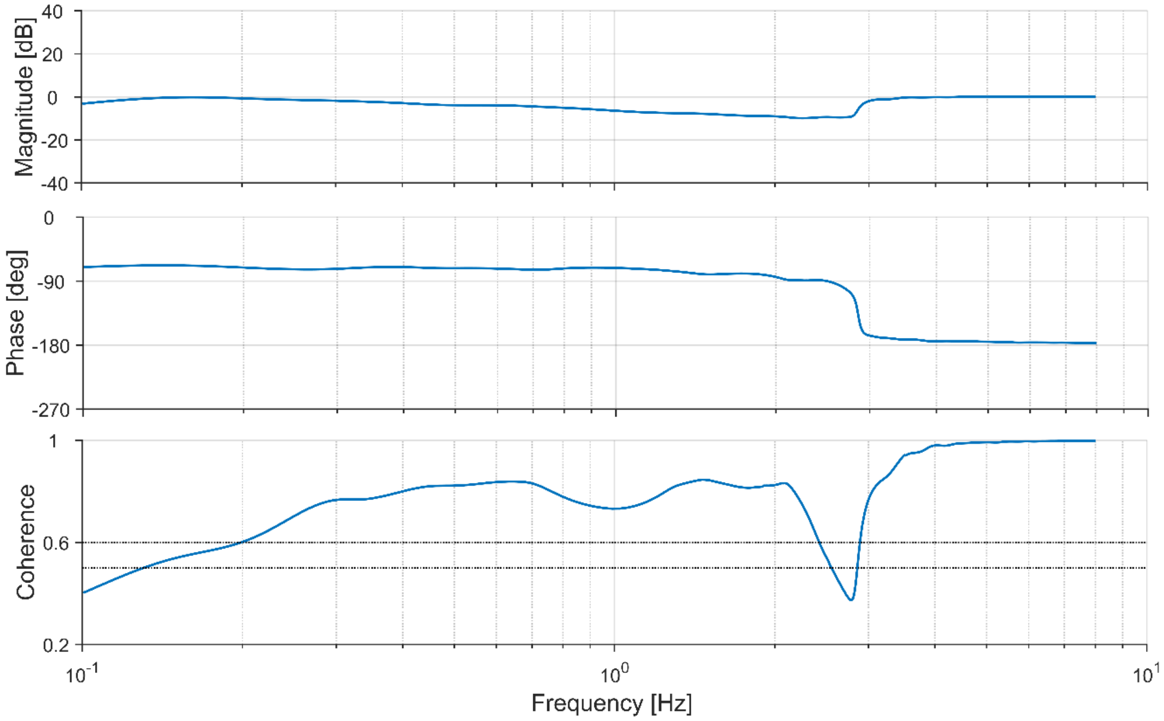


Figure 3-21 Frequency response of yaw-rate

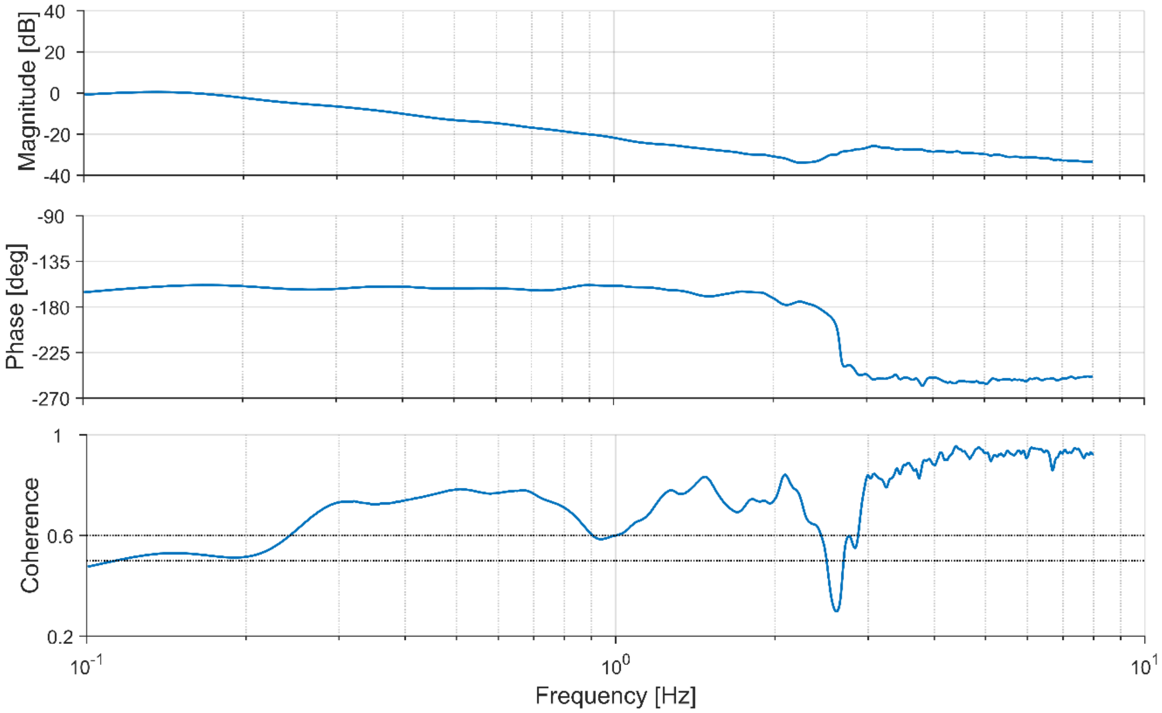


Figure 3-22 Frequency response of yaw angle

As it can be seen in the figures above the coherence values are acceptable 0.3 Hz to 2 Hz, that is why that particular frequency band has been selected for the identification process.

Table 3-17 Directional channel identification frequency band

Transfer Function	Start Frequency [Hz]	Stop Frequency [Hz]
r/δ_{rud}	0.3	2.4
ϕ/δ_{rud}	0.3	2.4

After running DERIVID program with the dedicated model in Equation 3.8 the following results have been obtained.

Table 3-18 Identified parameters

PARAM	VALUE	CR_BOUND	CR-%	INSENS-%
N_r	-1.8180	0.2020	11.11	4.190
$N_{\delta_{rud}}$	3.101	0.1100	3.548	1.531
τ_{rud}	0.006	0.0056	97.10	40.79

Cost:

$\boldsymbol{r/\delta_{rud}}$: 101.8668

$\boldsymbol{\psi/\delta_{rud}}$: 80.28391

Average cost: 91.0754

Results:

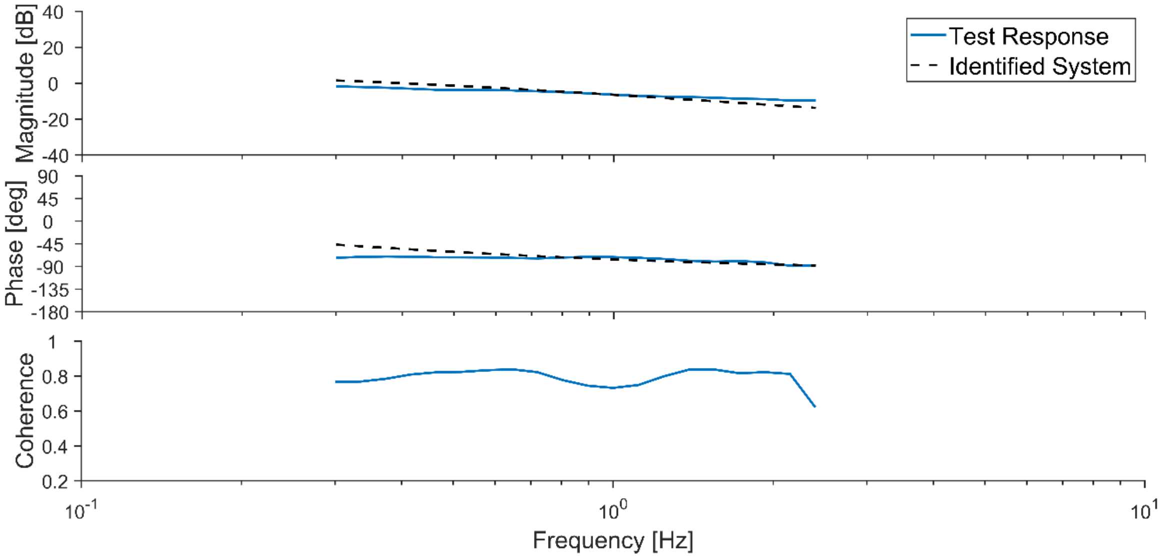


Figure 3-23 Comparison of ID TF of r/δ_{rud} vs Real system in Frequency Domain

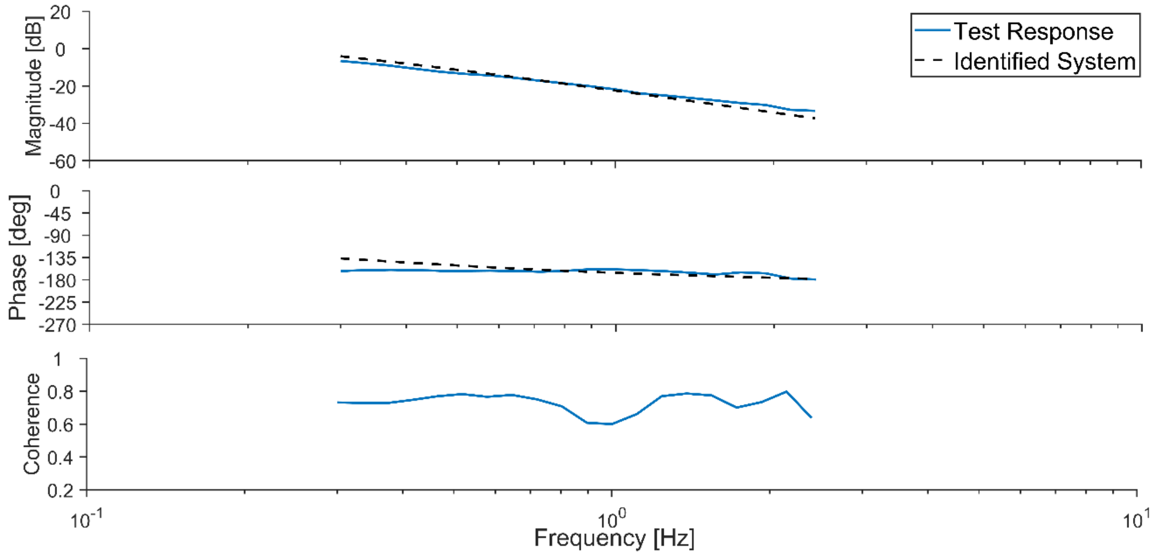


Figure 3-24 Comparison of ID TF of ψ/δ_{rud} *vs Real system in Frequency Domain*

As it can be seen in Table 3-18, Figure 3-23, and Figure 3-24 system identification provided the sufficient performances criteria. %CR is lower than 40%, and insensitivity values are close to 10% except τ. In the next step, step inputs have been applied in a flight test to validate the identified model. After obtaining an accurate result in the frequency domain, validation tests have been done. Yaw-rate command and responses of the identified and real system have been shown in that is shown in Figure 3-25 and Figure 3-26. J_{RMS} value is under 1, so validation has been done successfully.

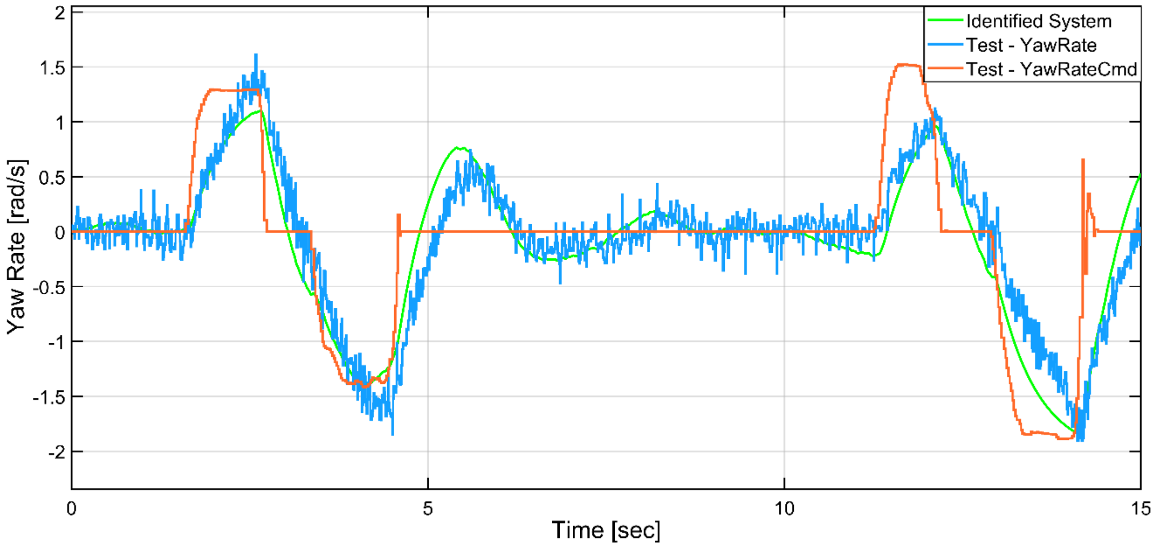


Figure 3-25 Yaw-rate response to doublet manoeuvre

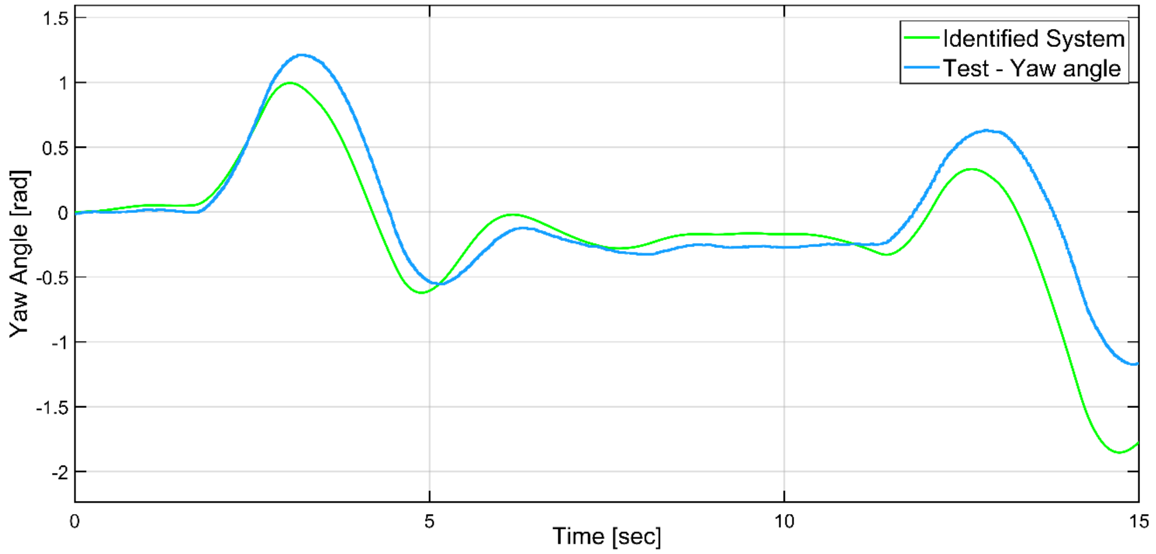


Figure 3-26 Yaw angle responses to doublet manoeuvre

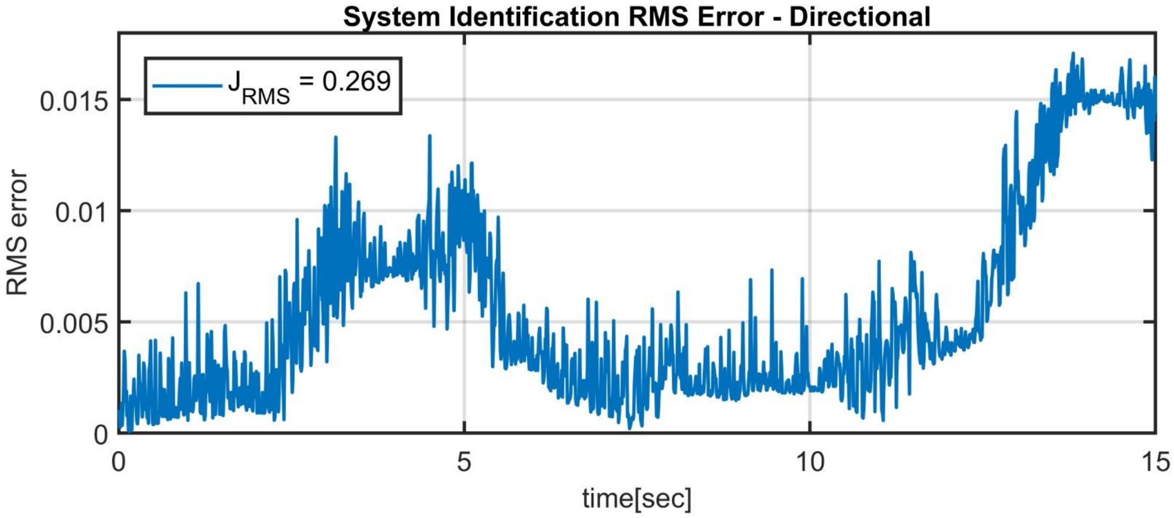


Figure 3-27 Directional response RMS error

3.4.3. Lateral and Longitudinal Channels

$$\begin{bmatrix} \dot{v} \\ \dot{p} \\ \phi \end{bmatrix} = \begin{bmatrix} Y_v & 0 & g \\ L_v & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} v \\ p \\ \phi \end{bmatrix} + \begin{bmatrix} Y_{\delta_{ail}} \\ L_{\delta_{ail}} \\ 0 \end{bmatrix} \delta_{ail} \tag{3.13}$$

$$\begin{bmatrix} \dot{u} \\ \dot{q} \\ \theta \end{bmatrix} = \begin{bmatrix} X_u & 0 & -g \\ M_u & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} u \\ q \\ \theta \end{bmatrix} + \begin{bmatrix} X_{\delta_{\mathrm{ele}}} \\ M_{\delta_{\mathrm{ele}}} \\ 0 \end{bmatrix} \delta_{\mathrm{ele}} \tag{3.14}$$

As examined, these channels have the same architecture, and their values should be very close physically. Another determination that can be made is that these channels have a pure damped mode and an oscillating unstable mode. As can be seen from the sample study at the source, this unstable mode, which is getting faster and faster as the size of the platform decreases, shows that multi-rotor platforms cannot be flown without control support. The following transfer functions can be extracted from here,

$$\dot{v} = Y_v v + g\phi + Y_{\delta_{ail}} \delta_{ail} \tag{3.15}$$

After the Laplace transform, the following equation is obtained.

$$vs - Y_v v = g\phi + Y_{\delta_{\mathrm{ail}}} \delta_{\mathrm{ail}} \tag{3.16}$$

$$v(\mathrm{s}) = \frac{g\phi + Y_{\delta_{\mathrm{ail}}}}{s - Y_v} \tag{3.17}$$

$$\dot{\mathrm{p}} = L_v v + L_{\delta_{\mathrm{ail}}} \delta_{\mathrm{ail}} \delta_{\mathrm{ail}} \tag{3.18}$$

Here, if we include the equation we got above:

$$ps = L_v \frac{g\phi + Y_{\delta_{ail}} \delta_{ail}}{s - Y_v} + L_{\delta_{ail}} \delta_{ail} \tag{3.19}$$

$$\mathrm{ps} = \frac{L_v(g\phi + Y_{\delta_{\mathrm{ail}}} \delta_{\mathrm{ail}})}{s - Y_v} + \frac{L_{\delta_{\mathrm{ail}}} \delta_{\mathrm{ail}} (s - Y_v)}{s - Y_v} \tag{3.20}$$

$$\phi = p/s \tag{3.21}$$

$$\mathrm{p}s^2 - \mathrm{p}\frac{L_v g}{s} - \mathrm{p}\frac{\mathrm{s}^2 Y_v}{s} = \frac{s L_v Y_{\delta_{\mathrm{ail}}} \delta_{\mathrm{ail}}}{s} + \frac{s^2 L_{\delta_{\mathrm{ail}}} \delta_{\mathrm{ail}}}{s} - \frac{s L_{\delta_{\mathrm{ail}}} \delta_{\mathrm{ail}} Y_v}{s} \tag{3.22}$$

$$\mathrm{p}(s^3 - \mathrm{s}^2 Y_v - L_v g) = \delta_{\mathrm{ail}}(s^2 L_{\delta_{\mathrm{ail}}} + s(L_v Y_{\delta_{\mathrm{ail}}} - L_{\delta_{\mathrm{ail}}} Y_v)) \tag{3.23}$$

$$\frac{\mathrm{p}}{\delta_{\mathrm{ail}}}(\mathrm{s}) = \frac{s^2 L_{\delta_{\mathrm{ail}}} + s(L_v Y_{\delta_{\mathrm{ail}}} - L_{\delta_{\mathrm{ail}}} Y_v)}{s^3 - \mathrm{s}^2 Y_v - L_v g} \tag{3.24}$$

As can be seen in the equation above, the relationship between p and δ_{ail} can be expressed by 2 zeros (one of which is at 0) and 3 poles. Or the equation can be rearranged according to Θ and diagnosed with 1 zero and 3 poles.

The linear channel can be obtained completely similarly as follows.

$$\frac{\mathrm{u}}{\delta_{\mathrm{ele}}}(\mathrm{s}) = \frac{s^2 M_{\delta_{\mathrm{ele}}} + s(L_u X_{\delta_{\mathrm{ail}}} - M_{\delta_{\mathrm{ele}}} X_u)}{s^3 - \mathrm{s}^2 X_u + M_u g} \tag{3.25}$$

3.4.3.1. Lateral Channel

Test Input:

For lateral and longitudinal channels, the identification process has been done while the attitude controllers active. Otherwise, the quadcopter cannot be stabilized during tests. Even though the controllers have been activated, the system identification process has used the output of the controllers, lateral command, δ_{ail} ,which is also the input of the mixer matrix. The sine input applied to attitude loop shown in Figure 3-28, and the δ_{ail} commands created by it are shown in Figure 3-29. The specification of the sine-sweep tabulated below:

Table 3-19 Sine-sweep Input Parameters

Starting Freq.	Ending Freq.	Number of Freq. Points	Test Period	Amplitude
0.1 Hz	15 Hz	16384	163.84s	± 0.4 rad

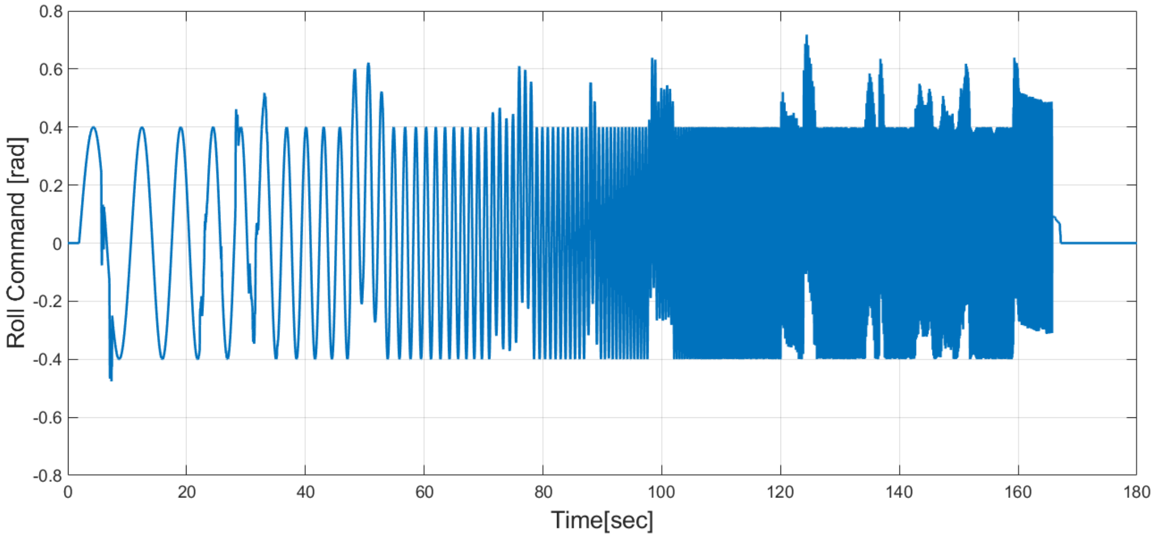


Figure 3-28 Roll command during flight test

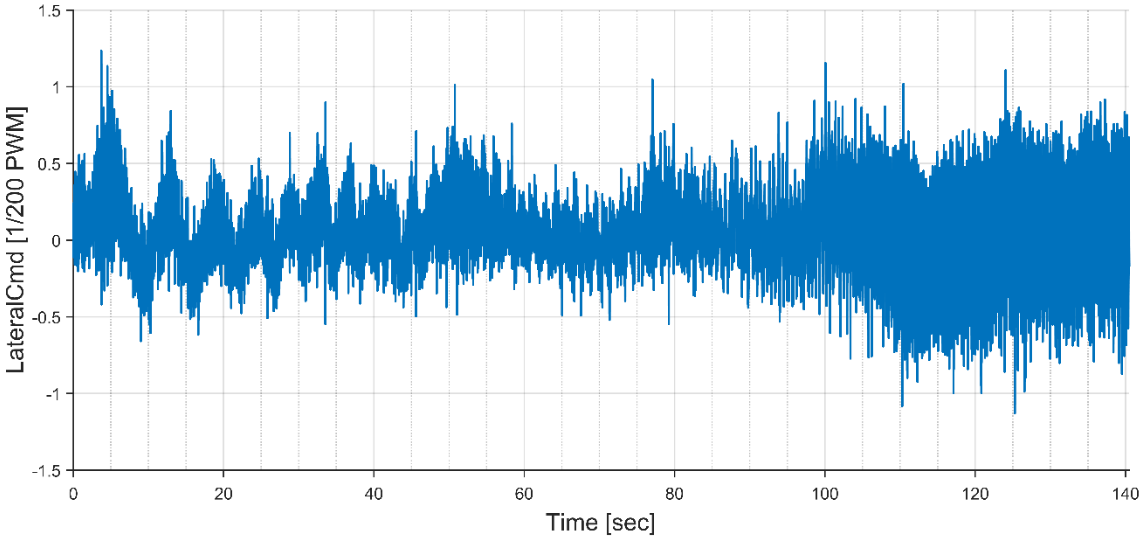


Figure 3-29 δ_{ail} *command during flight test*

Test Measurements:

Roll, roll-rate and lateral velocity measurements as a result of the applied sine input are shown in Figure 3-30, Figure 3-31 and Figure 3-32. When the data are examined, the system can follow the input quite well at low frequencies, and it can also respond to high frequencies, but its amplitude decreases. Another remarkable element here is that the output of the system at low frequencies cannot reach the input amplitude of 0.4 rad, because the roll angle controller does not have an integrator term.

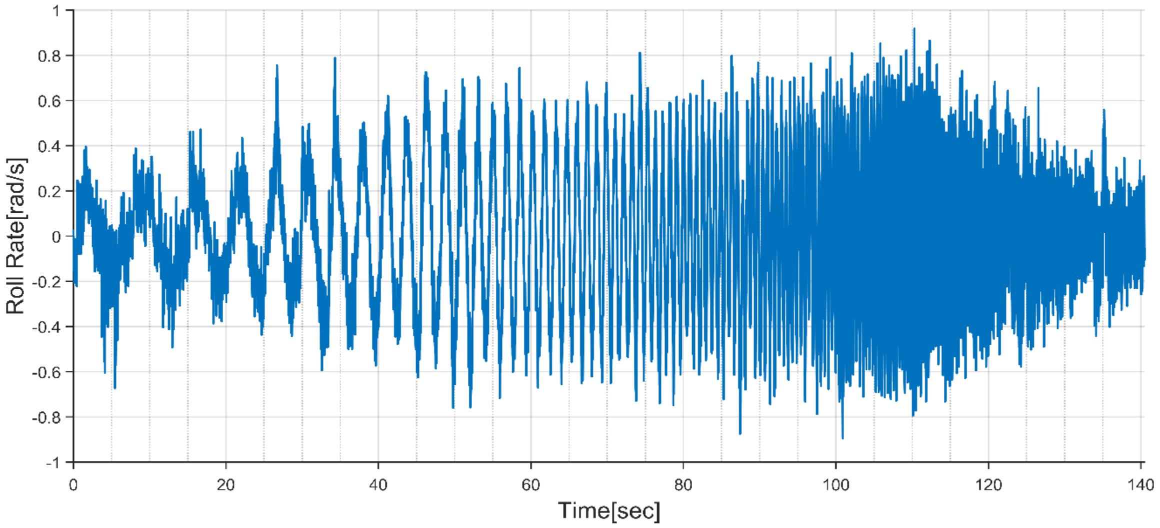


Figure 3-30 Roll-rate during flight test

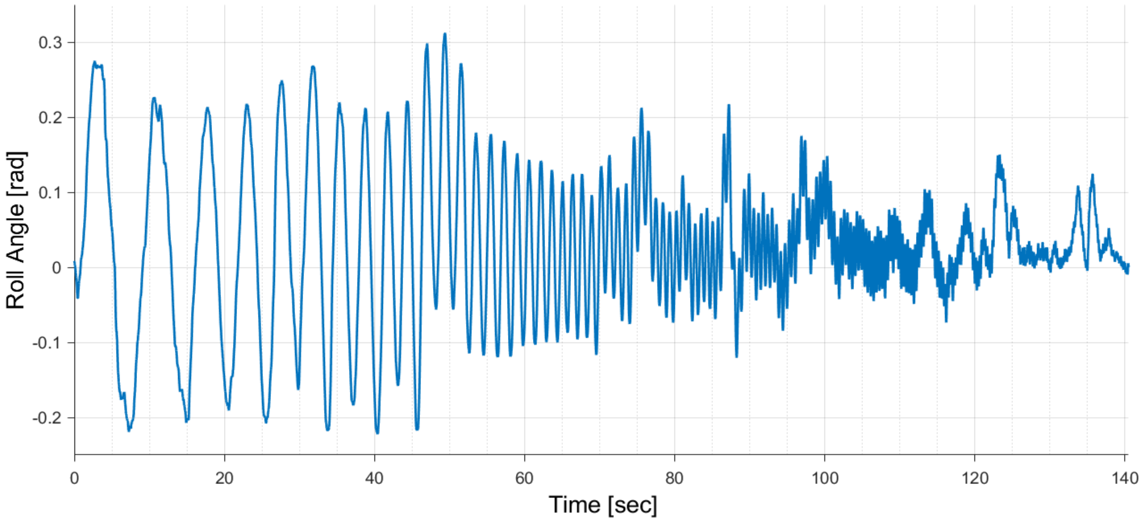


Figure 3-31 Roll angle during flight test

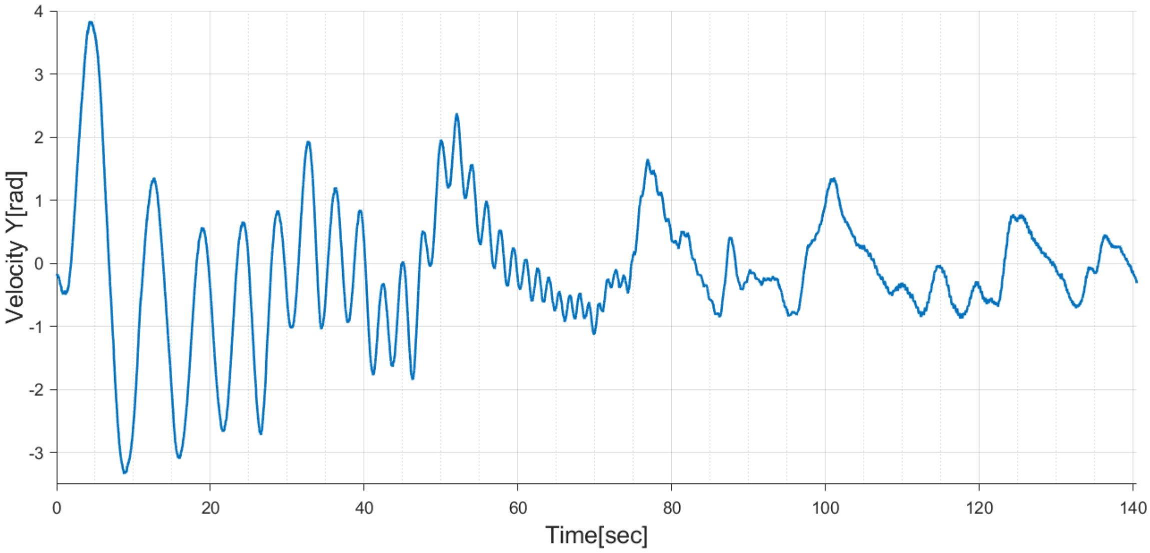


Figure 3-32 Lateral velocity during flight test

Frequency Responses:

Frequency responses of the signals have been calculated via FFTs by FRESPID and composed via the COMPOSITE function of the CIFER. After determining the window sizes, CIFER can adjust adequate settings for FFT process. The windowing settings for FFTs are tabulated below:

Table 3-20 Windowing setup for FFT

Window Size (sec)	Number of input points	Number of output points	Minimum Frequency (Hz)	Maximum Frequency (Hz)
23	2300	1796	0.0435	15
15	1500	548	0.0667	15
8	800	224	0.1250	15
4	400	112	0.2500	15
2.2	220	292	0.4546	15

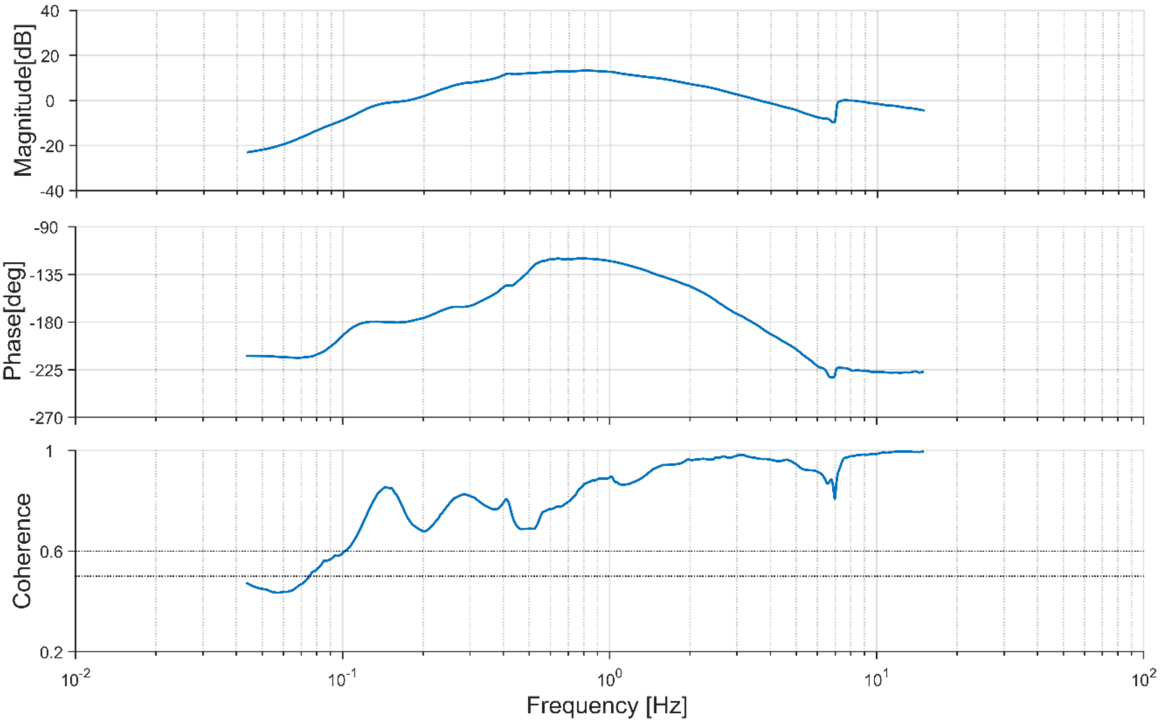


Figure 3-33 Frequency response of roll-rate

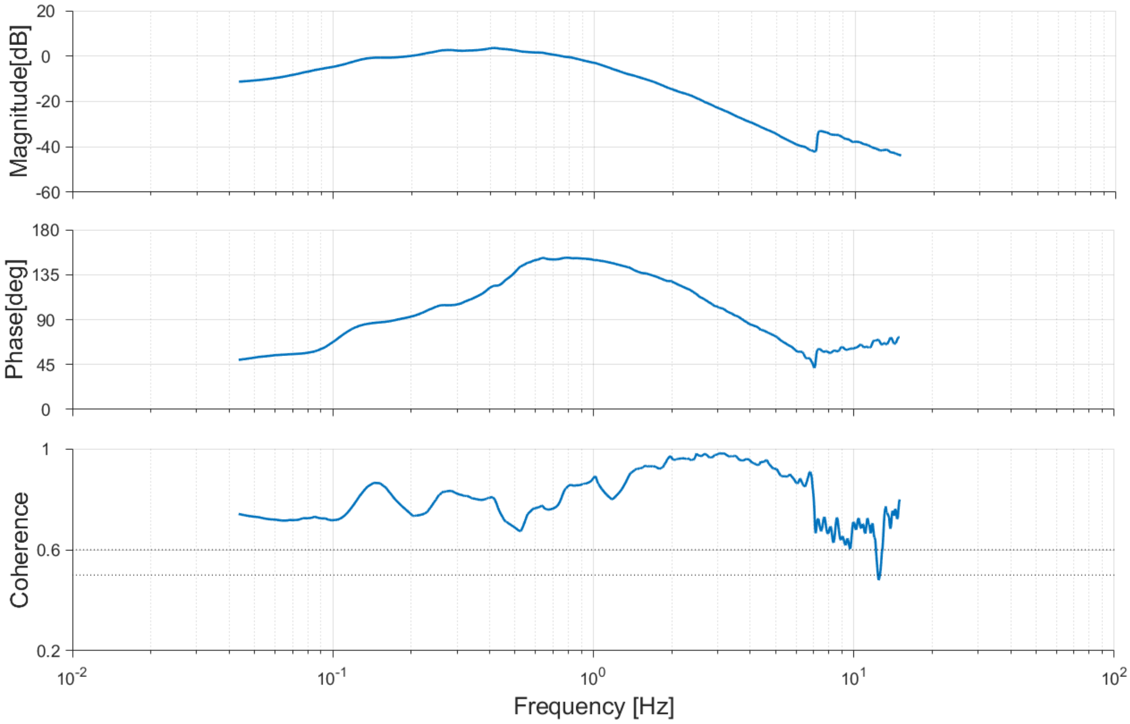


Figure 3-34 Frequency response of roll angle

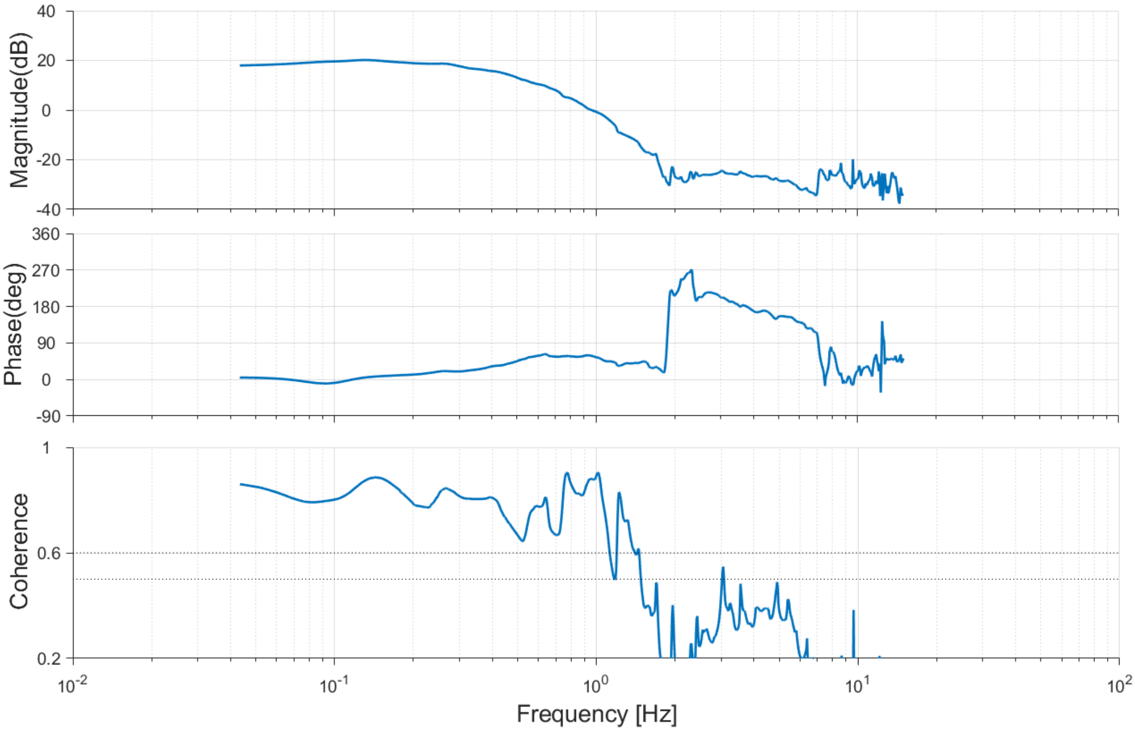


Figure 3-35 Frequency response of lateral velocity

When the frequency responses are examined, the frequency ranges that the system can respond to can be clearly seen. Although the coherence values seen at high frequencies are high, there is an inconsistent result in the frequency response due to the noise on the system. Because of that reason, the system is defined in the frequency ranges below.

Table 3-21 Lateral channel identification frequency band

TF Name	Start Frequency	Stop Frequency
v/δ_{ail}	0.05	1.7
p/δ_{ail}	0.05	7
ϕ/δ_{ail}	0.05	7

Table 3-22 Lateral channel identified parameters

PARAM	VALUE	CR_BOUND	CR-%	INSENS-%
Y_v	-0.1975	0.0525	26.57	9.402
L_v	-2.674	0.1372	5.130	1.729
$Y_{\delta_{ail}}$	0.4513	0.3040	67.37	23.28
$L_{\delta_{ail}}$	22.64	0.8152	3.601	1.196
τ_v	0.07597	0.0135	17.76	8.799
τ_p	0.06351	0.0030	4.568	2.282
τ_ϕ	0.06024	0.0030	4.869	2.433

Costs:

$\boldsymbol{v/\delta_{ail}}$: 34.45490

$\boldsymbol{p/\delta_{ail}}$: 70.49017

$\boldsymbol{\phi/\delta_{ail}}$: 73.32174

The average cost function is: 59.4223

Results:

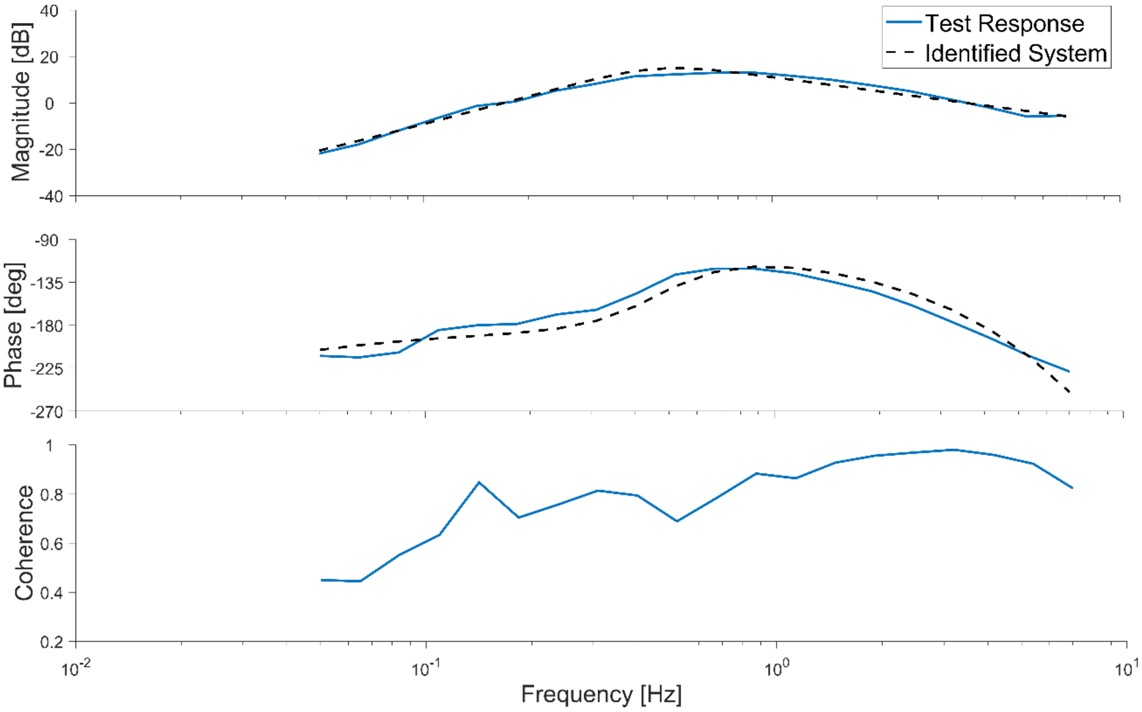


Figure 3-36 Comparison of ID TF of p/δ_{ail} vs Real system in Frequency Domain

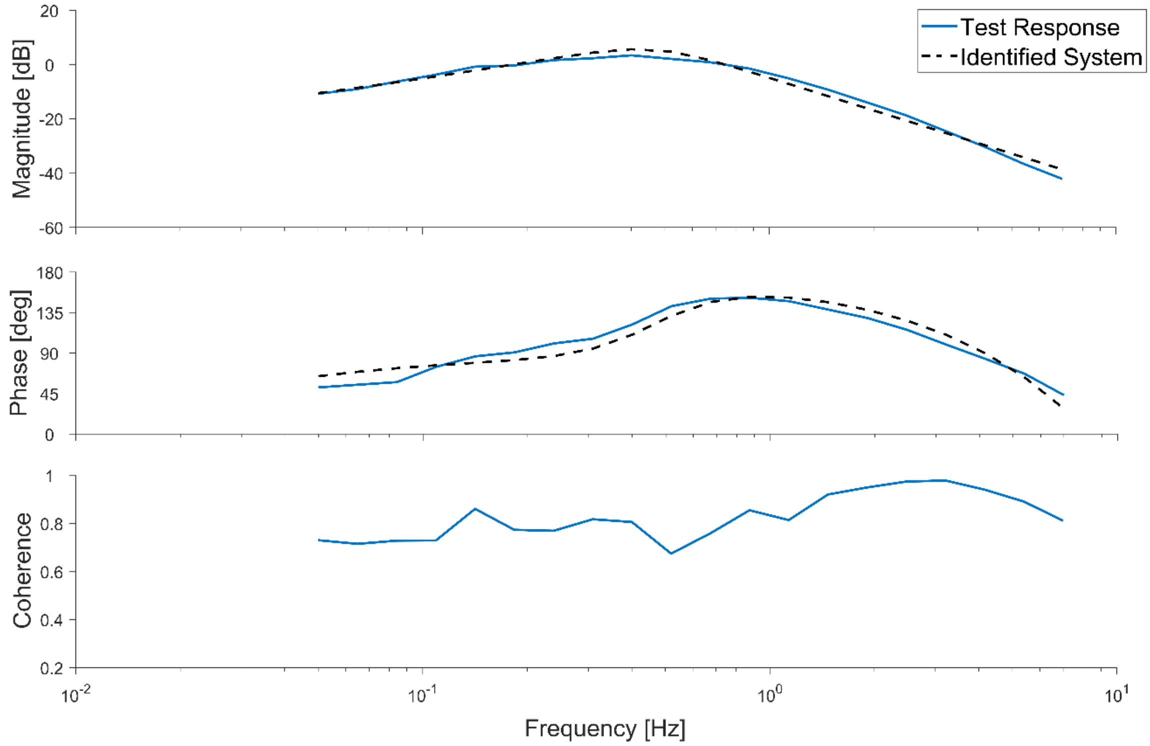


Figure 3-37 Comparison of ID TF of ϕ/δ_{ail} vs Real system in Frequency Domain

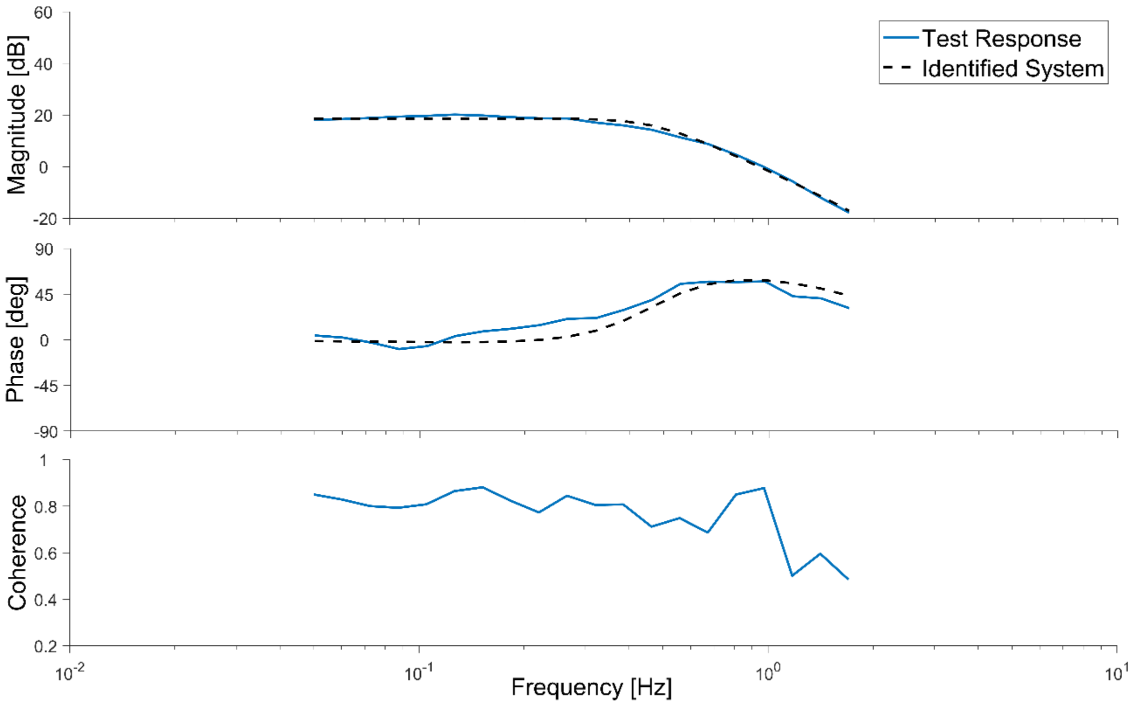


Figure 3-38 Comparison of ID TF of v/δ_{ail} vs Real system in Frequency Domain

As seen in Figure 3-36, Figure 3-37, Figure 3-38, the system identification has been done successfully in the frequency ranges of interest. However, the comparison in the frequency domain alone is not sufficient to assume that the system has been successfully defined. For this reason, a "doublet" manoeuvre has been applied to the system, as stated in the previous chapters. In the figure below, the comparison of the identified system and flight test can be found.

During the validation tests roll command, which has been shown in Figure 3-40, has been applied to the system. Meanwhile, roll angle and roll-rate controller was active. So, the roll angle controller has generated roll-rate command, which is also shown in Figure 3-39. The responses of the system have been stated with reference commands in the same figures.

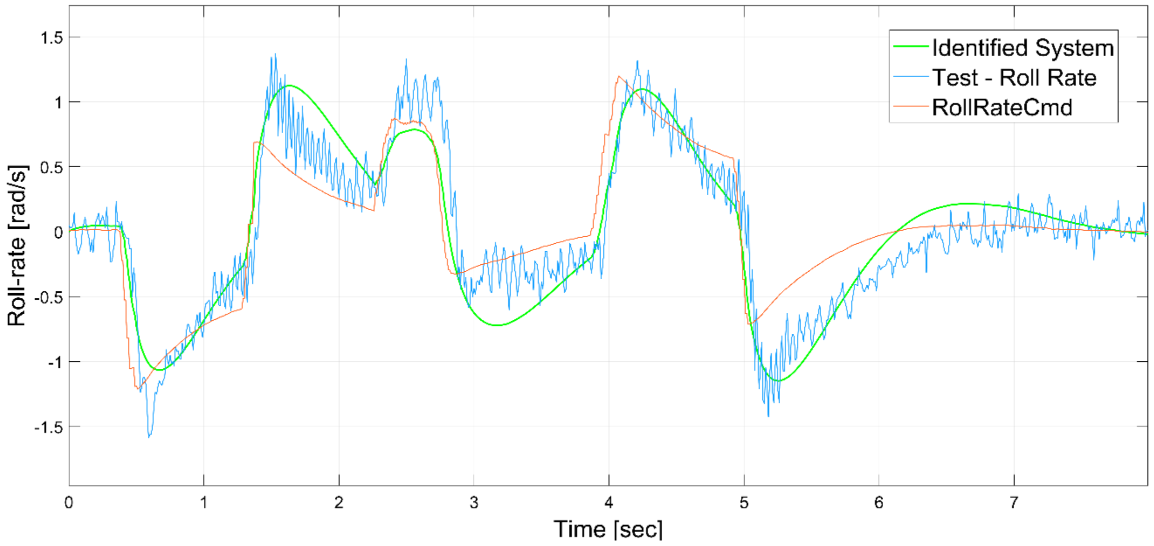


Figure 3-39 Roll-rate responses to doublet manoeuvre

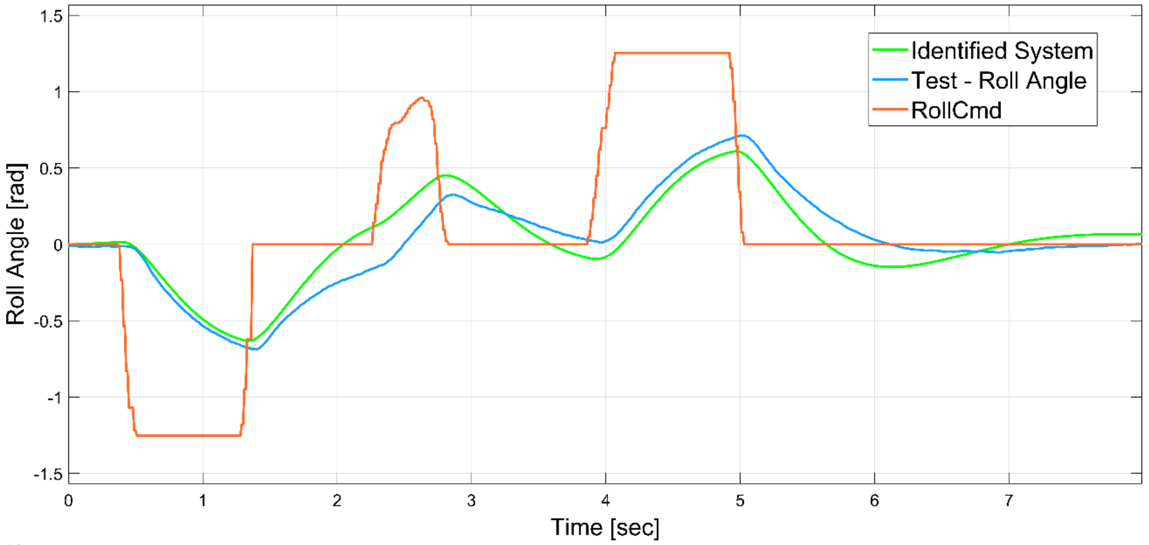


Figure 3-40 Roll angle responses to doublet manoeuvre

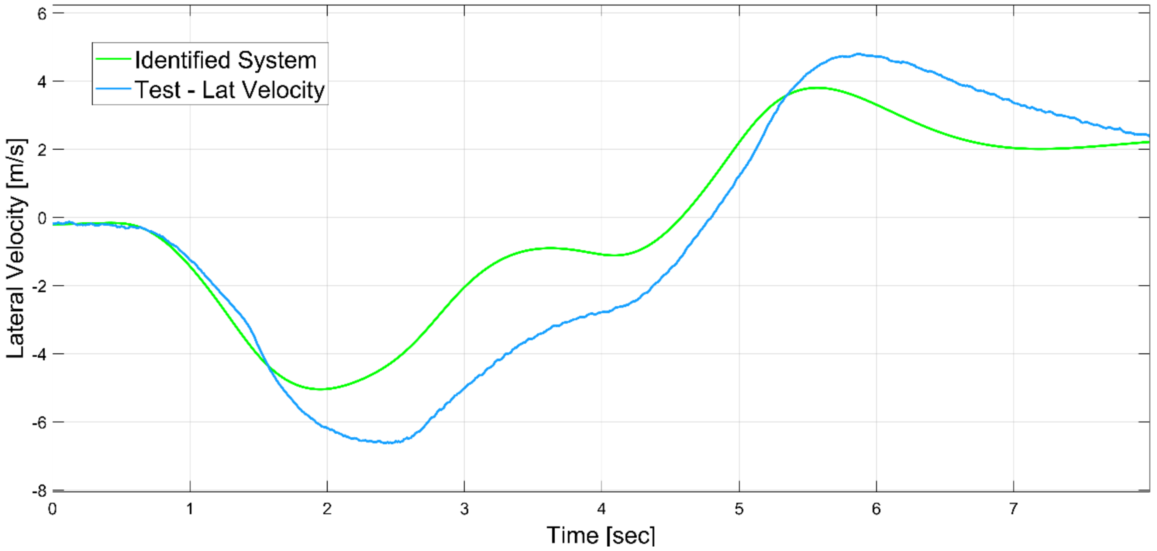


Figure 3-41 Lateral velocity responses to doublet manoeuvre

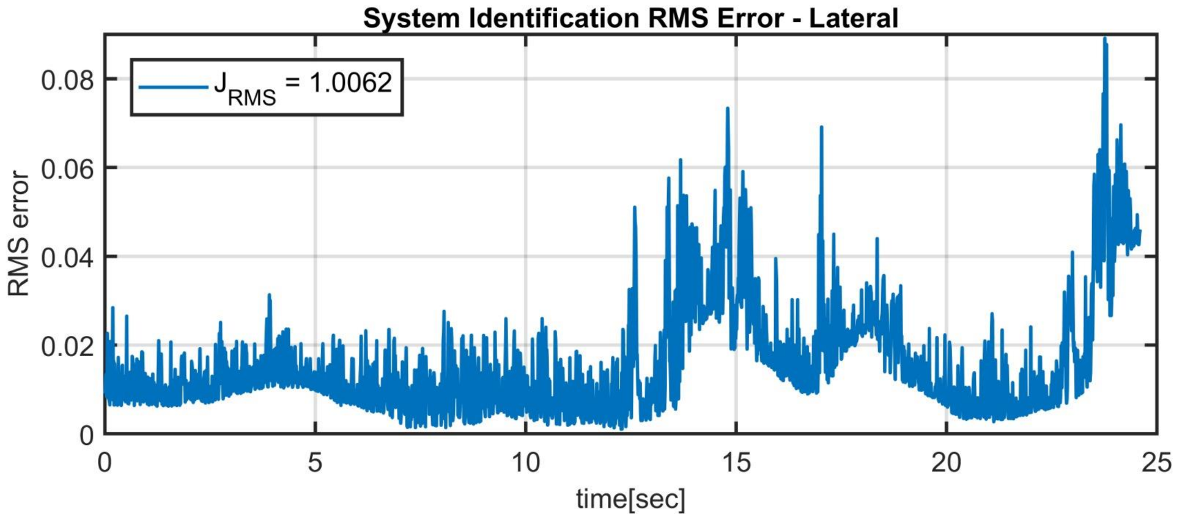


Figure 3-42 Lateral velocity response RMS error

When Figure 3-39, Figure 3-40 and Figure 3-41 are examined, it shows that the responses given to the doublet manoeuvre in the time domain can be quite adequate for angular velocity and angle channels. When looking at the speed channel, there is a difference at low frequencies, although trends have been caught. It is considered that this difference may occur due to wind effect. But the J_{RMS} error was around 1, and that shows the validation has been done successfully.

3.4.3.2. Longitudinal Channel

The longitudinal channel is almost the same as the lateral channel in structure, so it is expected that the parameter values are also very similar. These two channels can be differed, because of the reasons, such as asymmetry in the body of the platform, unbalanced center of gravity and the inertia differences on the axes.

During the identification process for the longitudinal channel, some problems were encountered due to test conditions or reasons mentioned above. Like the system lateral channel, when all parameters are tried to be defined freely, it is seen that the time delay is negative for the speed loop and the X_u damping parameter is positive. Although the tests were repeated, these parameters could not be obtained in accordance with physics. For this reason, the time delay value of the speed loop during the definition of this channel was forced as 0.075s.

Test Input:

The applied sine-sweep signal information is given in the table below.

Table 3-23 Sine-sweep Input Parameters

Starting Freq.	Ending Freq.	Number of Freq. Points	Test Period	Amplitude
0.05 Hz	8 Hz	16384	163.84s	± 0.3 rad

The above signal was produced and applied to the angle channel as shown in Figure 6 during the test. Distortions on the signal occurred due to the pilot's corrections.

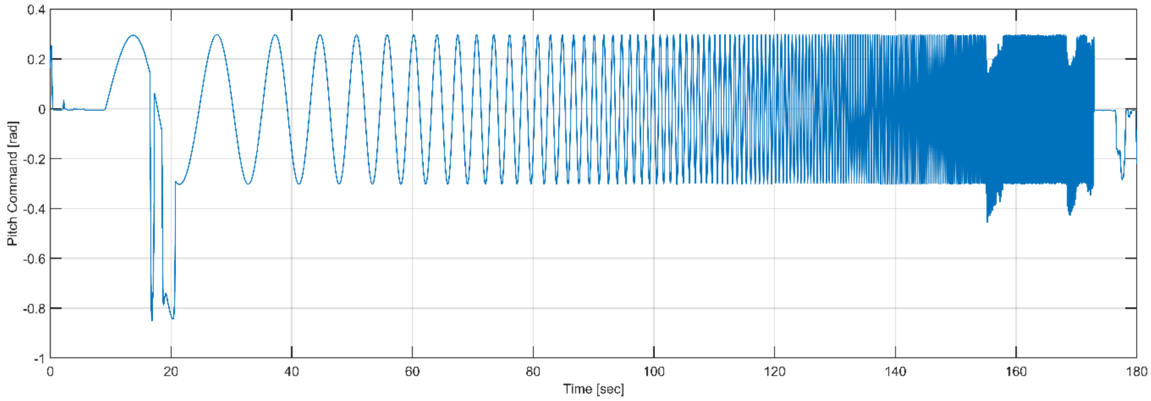


Figure 3-43 Pitch angle command during flight test

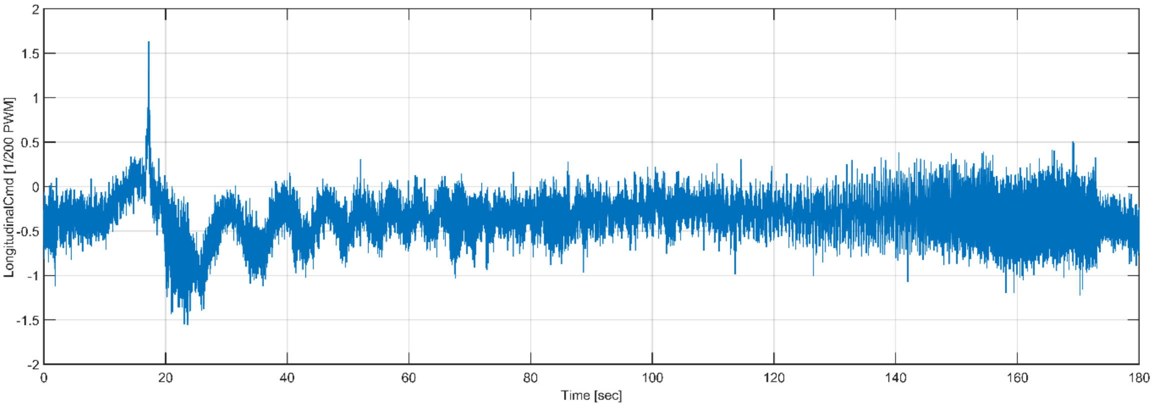


Figure 3-44 δ_{ele} command during flight test

Test Measurements:

Angular velocity, angle and velocity measurements, which are results of the sine sweep tests, are shown in Figure 22, Figure 23, Figure 24. When the data are examined, the system can follow the input quite well at low frequencies and respond to high frequencies, but its amplitude decreases. Another remarkable element here is that the output of the system at low frequencies cannot reach the input amplitude of 0.3 rad. The reason for this is that the angle controller does not have the integrator term.

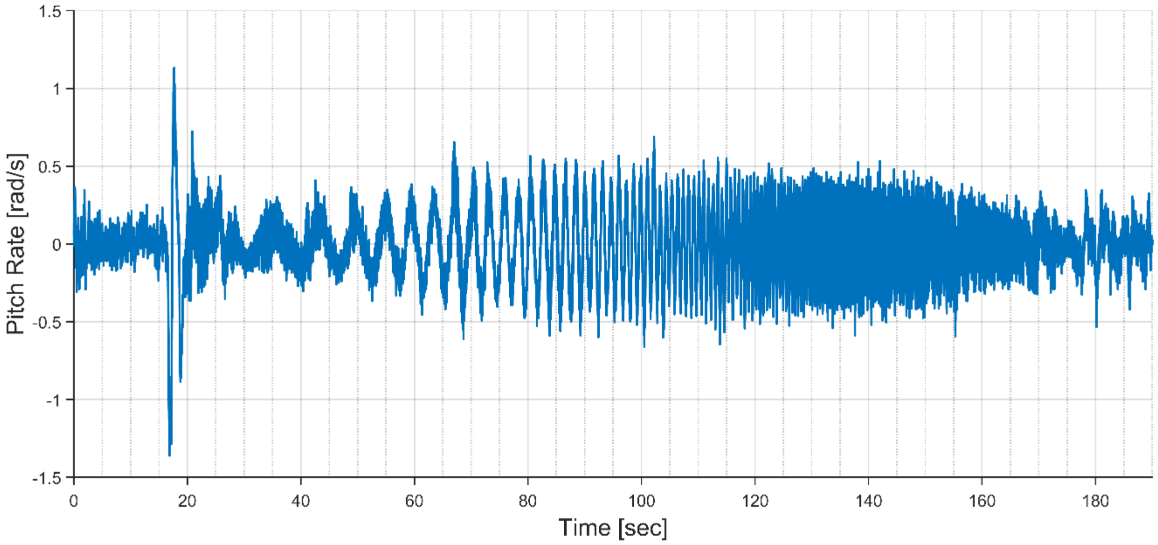


Figure 3-45 Pitch-rate response during flight test

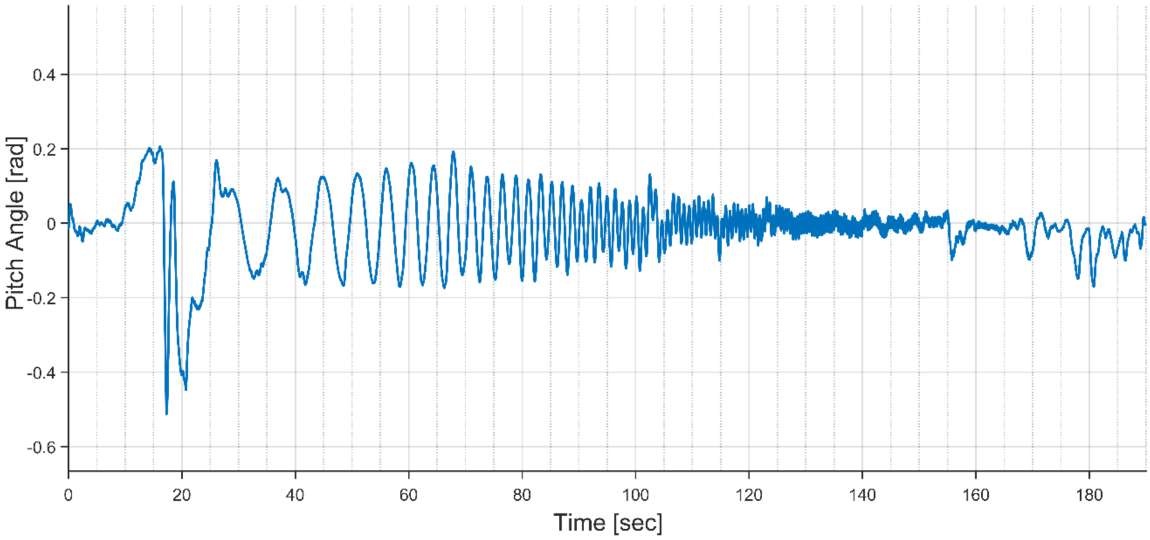


Figure 3-46 Pitch angle response during flight test

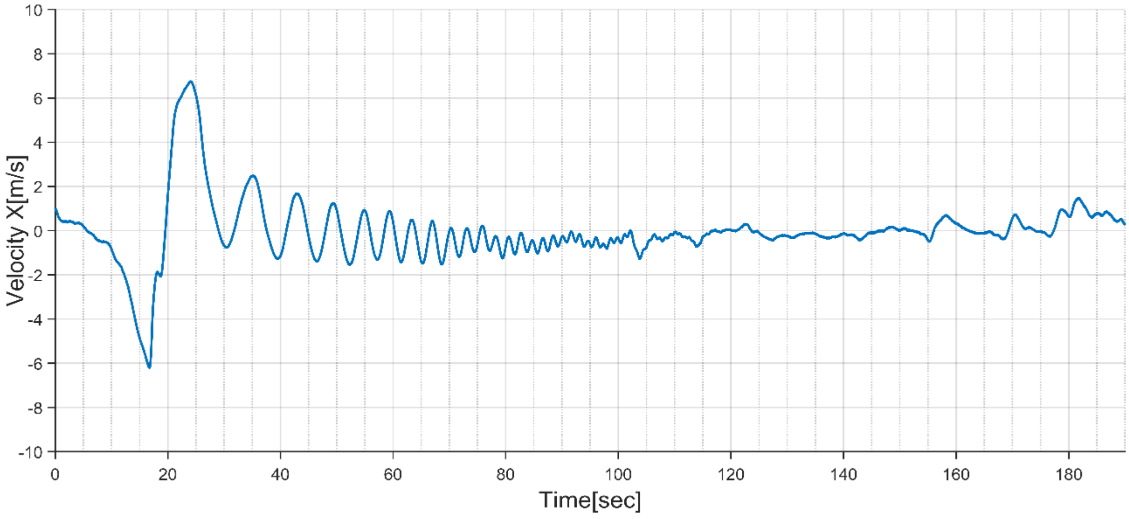


Figure 3-47 Longitudinal velocity response during flight test

Frequency Responses:

Frequency responses of the signals have been calculated via FFTs by FRESPID and composed via the COMPOSITE function of the CIFER. After determining the window sizes, CIFER can adjust adequate settings for FFT process. The windowing settings for FFTs are tabulated below:

Table 3-24 Longitudinal channel windowing setup for FFT

Window Size (sec)	Number of input points	Number of output points	Minimum Frequency (Hz)	Maximum Frequency (Hz)
25	2500	1596	0.0400	10
9	900	124	0.1111	10
4	400	112	0.2500	10
2.5	250	262	0.4000	10
1.2	120	136	0.8333	10

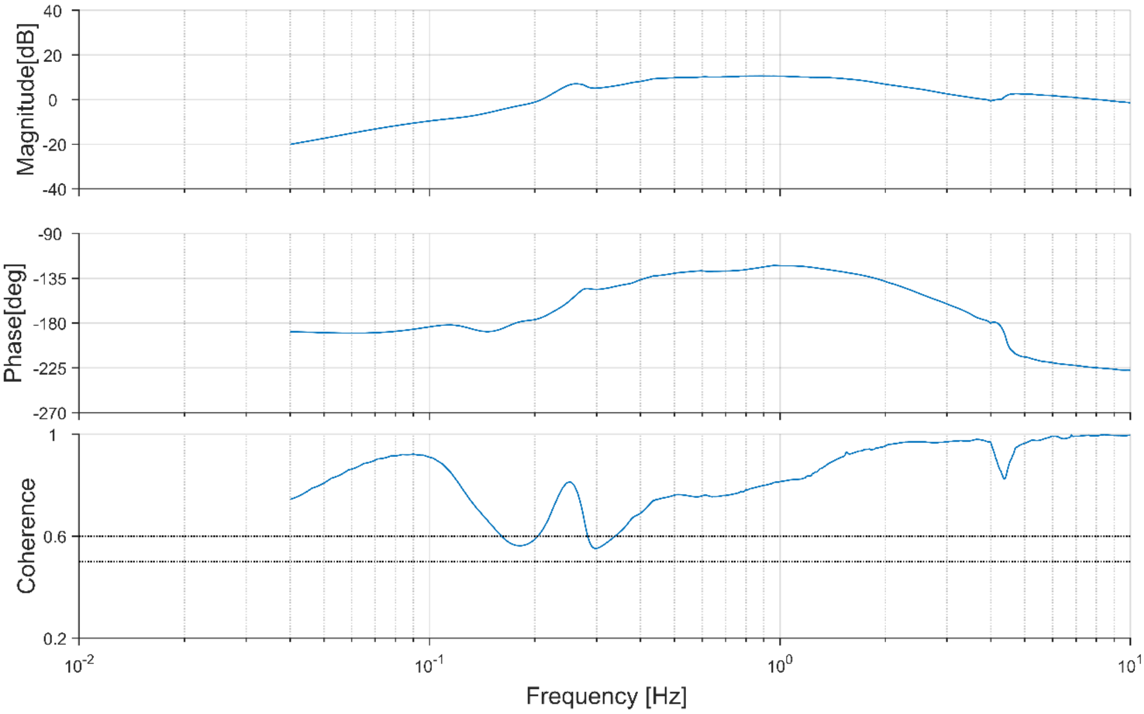


Figure 3-48 Pitch-rate frequency response

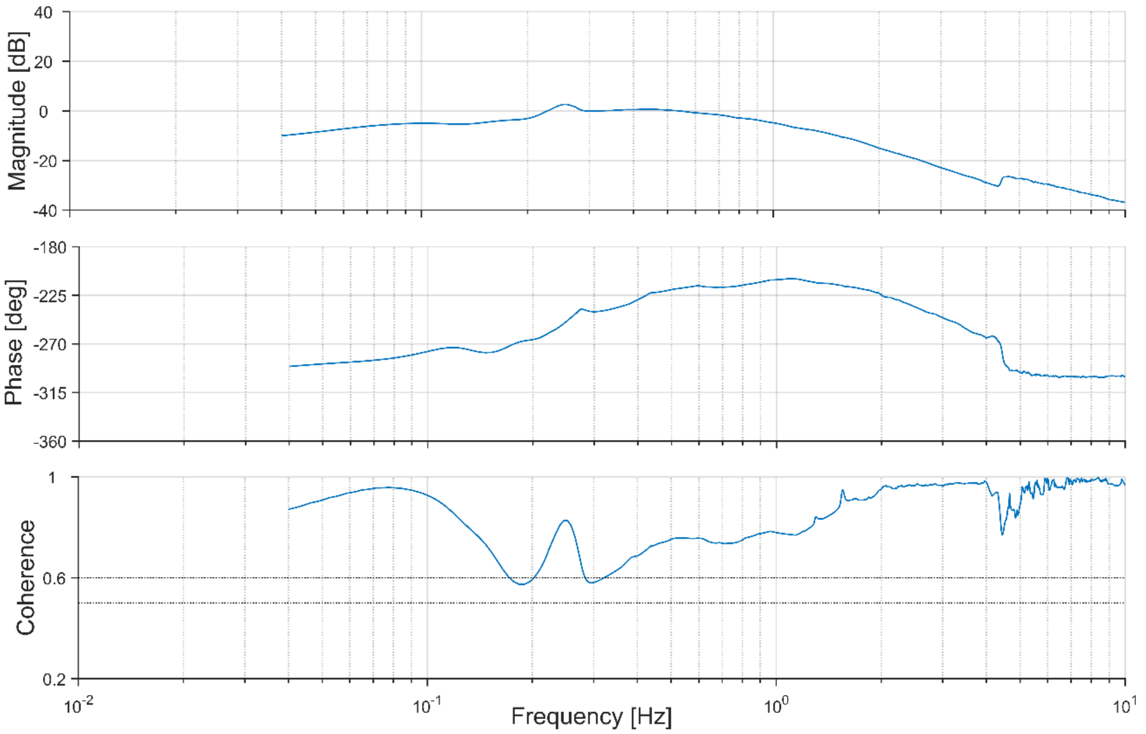


Figure 3-49 Pitch angle frequency response

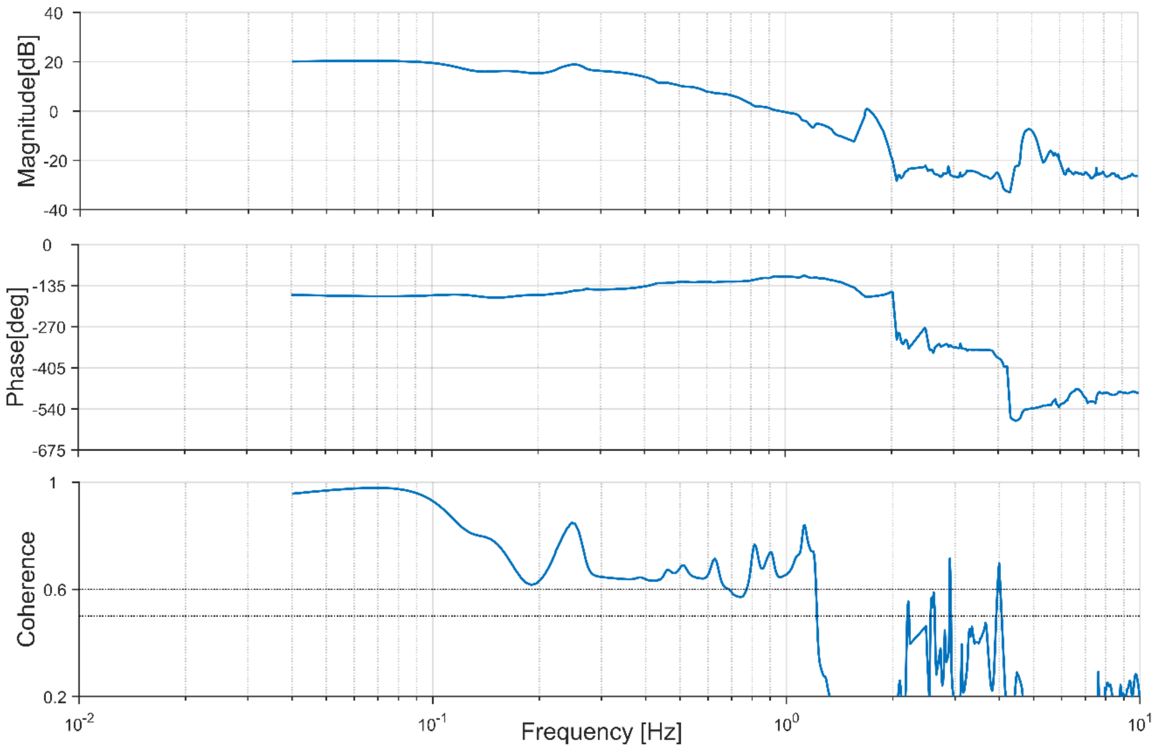


Figure 3-50 Longitudinal velocity frequency response

When the frequency responses are examined, we can clearly see the frequency band of the system. The reason for the artificial improvement we see here at high frequencies is that the system noise becomes dominant. The relevant frequency ranges required for system identification are given in the table below.

Table 3-25 Longitudinal channel identification frequency band

TF Name	Start Frequency	Stop Frequency
u/δ_{ele}	0.04	1.5
q/δ_{ele}	0.04	4
θ/δ_{ele}	0.04	4

Results:

The system identification has been performed in the frequency ranges shown in Table 3-25 Longitudinal channel identification frequency band. The results are given in the Table 3-26. During the definition of the linear channel, unrealistic results were obtained in the parameters $X_{\delta_{ele}}$ and τ_u. Therefore, a more consistent result is achieved in the system identification process by taking these parameters from the lateral channel, which has similar characteristics. Even though this change caused the cost to appear higher, it gave a more consistent result with the real dynamics.

Table 3-26 Longitudinal channel identified parameters

PARAM	VALUE	CR_BOUND	CR-%	INSENS-%
X_u	-0.02514	0.02347	93.35	46.44
M_u	2.462	0.1237	5.024	1.607
$X_{\delta_{ele}}$	0.4513	-	-	-
$M_{\delta_{ele}}$	20.820	0.7954	3.820	1.212
τ_u	0.07500	-	-	-
τ_q	0.06343	0.005	7.516	3.754
τ_θ	0.05914	0.005	8.073	4.033

Costs:

u/δ_{ele} : 118.0624

q/δ_{ele} : 143.1461

θ/δ_{ele} : 113.8166

Average Cost: 125.0084

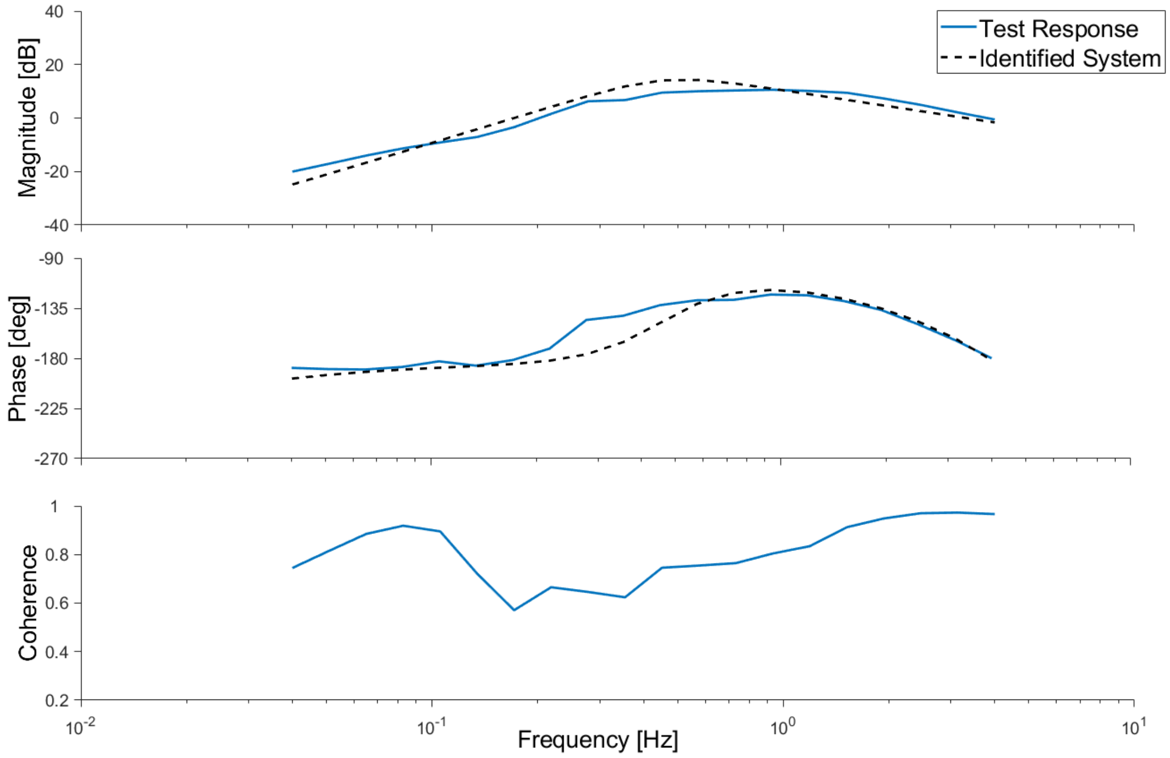


Figure 3-51 Comparison of ID TF of q/δ_{ele} vs Real system in Frequency Domain

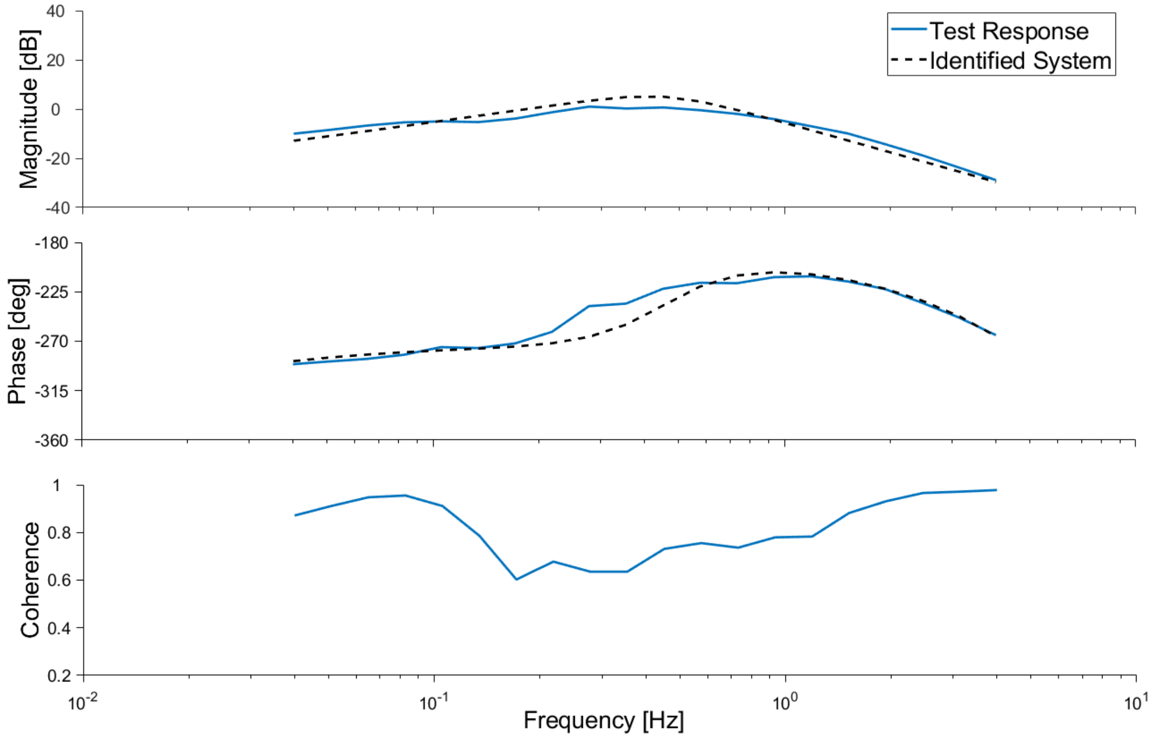


Figure 3-52 Comparison of ID TF of θ/δ_{ele} vs Real system in Frequency Domain

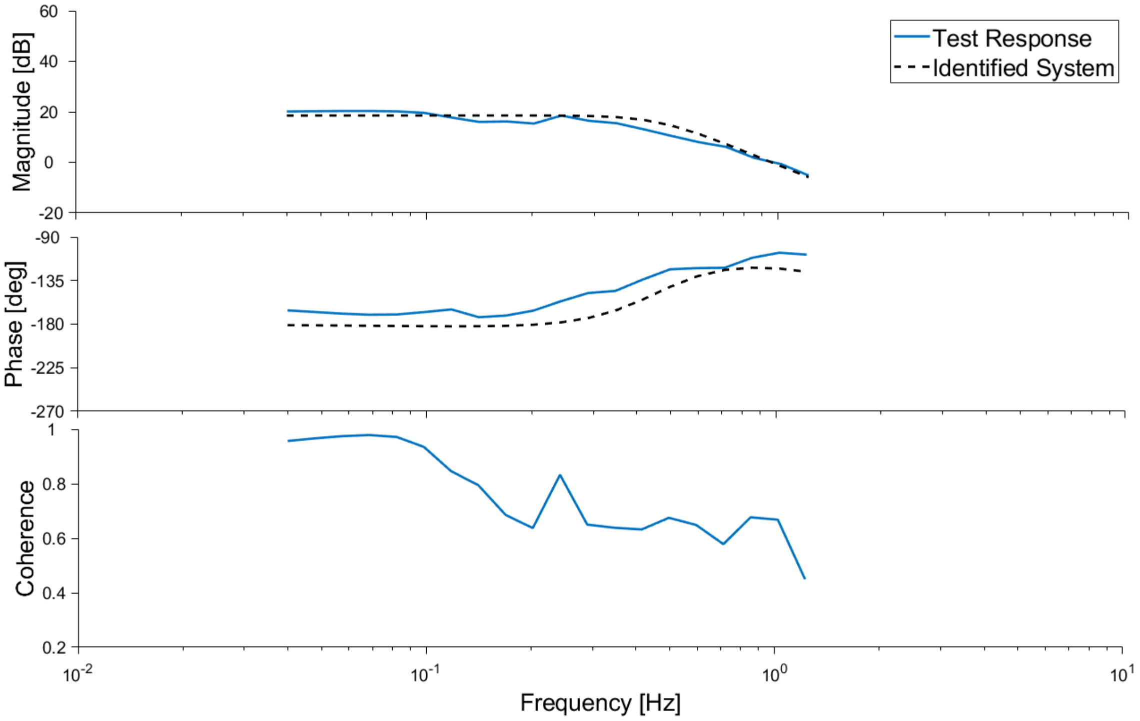


Figure 3-53 Comparison of ID TF of u/δ_{ele} vs Real system in Frequency Domain

As seen in Figure 3-51, Figure 3-52, Figure 3-53the system identification has been done successfully in the frequency ranges of interest. These results are satisfying enough to continue the next process, which is validation.

During the validation tests pitch command, which has been shown in Figure 3-55, has been applied to the system. Meanwhile, pitch angle and pitch-rate controller were active. So, the pitch angle controller has generated pitch-rate command, which is also shown in Figure 3-54Figure 3-39. The responses of the system have been stated with reference commands in the same figures.

The system responses measured after the doublet manoeuvre are shown in the Figure 3-54, Figure 3-55, Figure 3-56. The J_{RMS} error value is around 1.46, which is acceptable also shown in Figure 3-57.

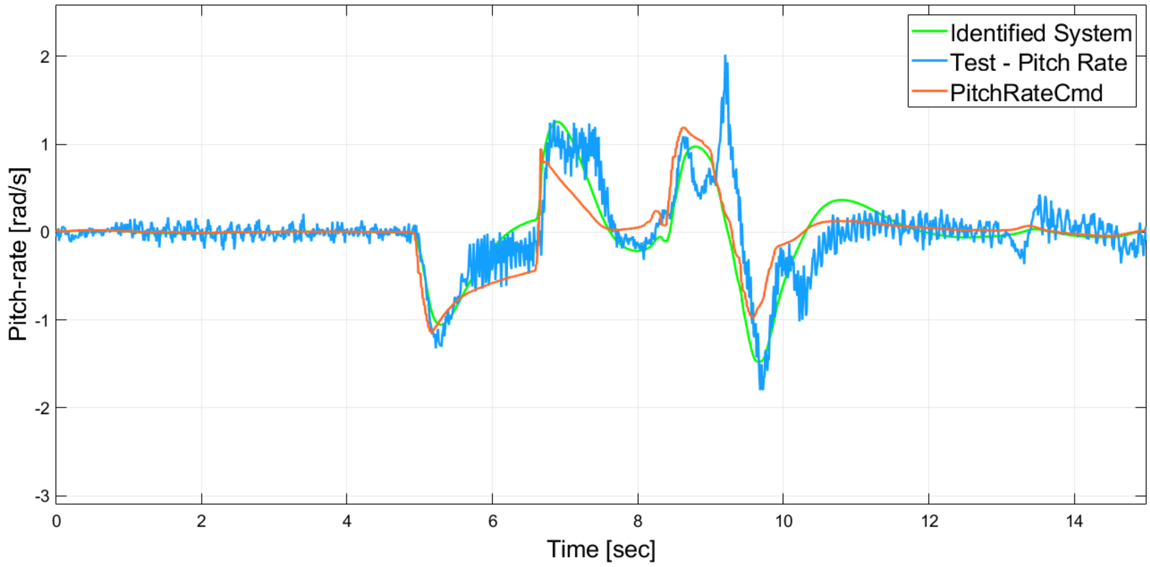


Figure 3-54 Pitch-rate responses to doublet manoeuvre

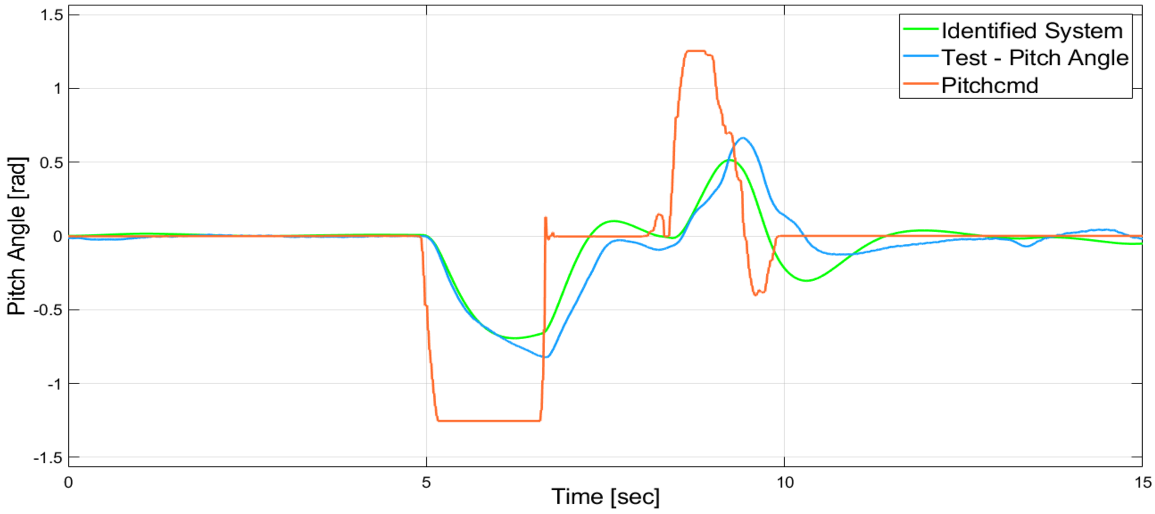


Figure 3-55 Pitch angle responses to doublet manoeuvre

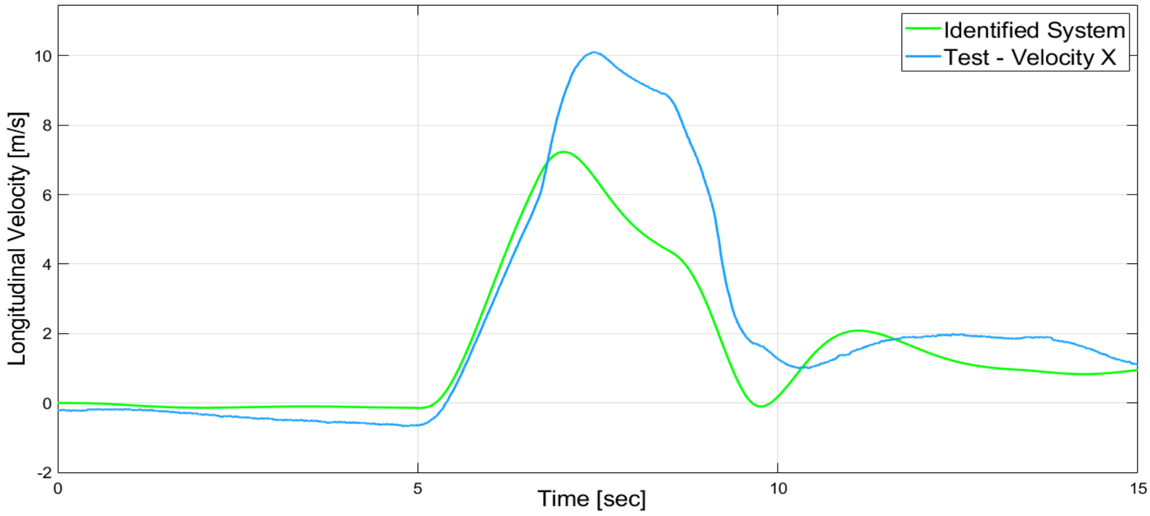


Figure 3-56 Longitudinal velocity responses to doublet manoeuvre

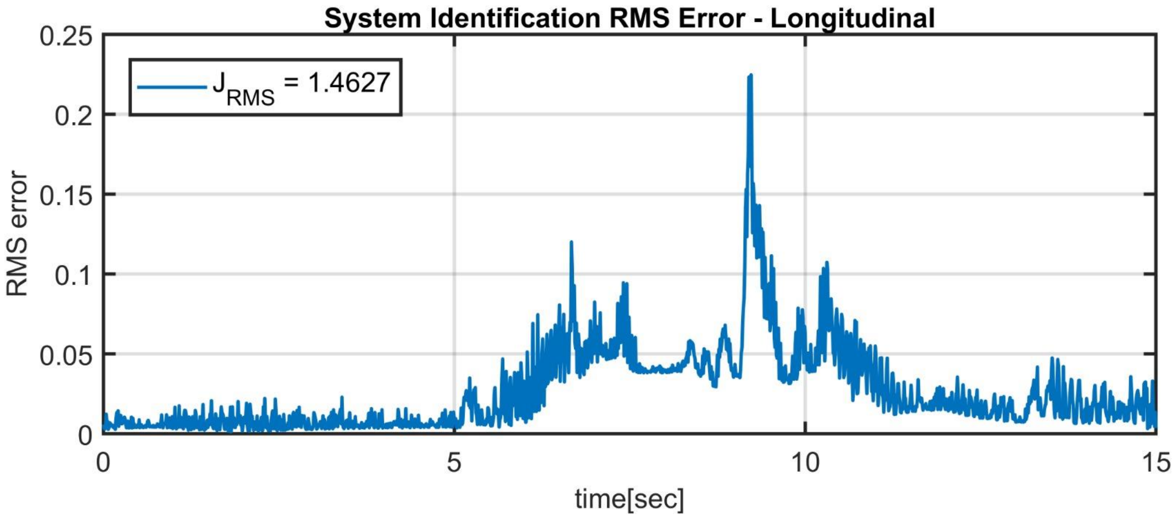


Figure 3-57 Longitudinal velocity response RMS error

Summary of the dynamic model:

$$\begin{bmatrix}\dot{u}\\ \dot{v}\\ \dot{w}\\ \dot{p}\\ \dot{q}\\ \dot{r}\\ \dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\end{bmatrix}=\begin{bmatrix}-0.02514 & 0 & 0 & 0 & 0 & 0 & 0 & -9.81 & 0\\ 0 & -0.1975 & 0 & 0 & 0 & 0 & 9.81 & 0 & 0\\ 0 & 0 & -0.2518 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & -2.674 & 0 & 0.8812 & 0 & 0 & 0 & 0 & 0\\ 2.462 & 0 & 0 & 0 & 1.818 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & -1.818 & 0 & 0 & 0\\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}u\\ v\\ w\\ p\\ q\\ r\\ \phi\\ \theta\\ \psi\end{bmatrix}$$

$$+\begin{bmatrix}0 & 0 & 0.4513 & 0\\ 0 & 0.4513 & 0 & 0\\ -0.01229 & 0 & 0 & 0\\ 0 & 22.64 & 0 & 0\\ 0 & 0 & 20.82 & 0\\ 0 & 0 & 0 & 3.101\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}\delta_{thr}\\ \delta_{ail}\\ \delta_{ele}\\ \delta_{rud}\end{bmatrix} \qquad (3.26)$$

4. MODEL PREDICTIVE CONTROLLER DESIGN

The model predictive control is an optimal control method which contains the system model to predict future motions of the system to generate optimal control sequence. In this section, the controller design is discussed step by step. In the first part, generic definitions and notation are explained.

4.1. Design Parameters

Sample time: Defines the sample rates of the controller. In the controller design, the correct selection of sample time is important to get the desired performance. Lower sample time may cause performance degradation, on the other hand, higher sample time needs much more computation power, but also it increases the disturbance rejection performance. Therefore, optimal sample time must be found for satisfying performance. During this thesis work, some different sample time and the prediction horizon has been tried. The best performance has been observed with 80 ms sample time.

Prediction horizon: defines the number of iteration step for prediction. The prediction horizon duration, $p \cdot Ts$, should be bigger than settling time or rise time depending on the desired focus. In general, $p \cdot Ts$ can be selected as %10-25 of minimum desired closed-loop response [22]. Also, low sampling time and high prediction horizon might cause infeasible QP solution, especially in open-loop unstable systems like quadcopters.

The system identification process showed that the system could respond to up to 3-4 Hz. So, controller sample time has been selected as 12.5 Hz to cover all system dynamic with an acceptable computational load. After, the system model has been implemented into MATLAB workspace in state-space form. The detailed information about the system model could be found in section 3.3.

Control Horizon: It is the length of the control input sequence to be optimized at each time step. Broader control horizon increases computation times dramatically.

Sampling time has selected as 0.08s besides 13 step prediction horizon and 7 step control horizon in this thesis.

Constraints: MPC can handle constraints both on control inputs and states. The critical point is avoiding unnecessary constraints, particularly hard ones. More constraint can increase computation time and might come with an infeasible QP solution [23].

$$\frac{y_{j,min}(i)}{s_j^y} - \varepsilon_k V_{j,min}^y(i) \le \frac{y_j(k+i|k)}{s_j^y} \le \frac{y_{j,max}(i)}{s_j^y} - \varepsilon_k V_{j,max}^y(i), \qquad (4.1)$$
$$where \qquad i = 1{:}\,p, \qquad j = 1{:}\,n_y$$

$$\frac{u_{j,min}(i)}{s_j^u} - \varepsilon_k V_{j,min}^u(i) \le \frac{u_j(k+i-1|k)}{s_j^u} \le \frac{u_{j,max}(i)}{s_j^u} - \varepsilon_k V_{j,max}^u(i), \qquad (4.2)$$
$$where \qquad i = 1{:}\,p, \qquad j = 1{:}\,n_u$$

$$\frac{\Delta\, u_{j,min}(i)}{s_j^u} - \varepsilon_k V_{j,min}^{\Delta\, u}(i) \le \frac{\Delta\, u_j(k+i-1|k)}{s_j^u} \le \frac{\Delta\, u_{j,max}(i)}{s_j^u} - \varepsilon_k V_{j,max}^{\Delta\, u}(i), \qquad (4.3)$$
$$where \qquad i = 1{:}\,p, \qquad j = 1{:}\,n_u$$

In these equations, the V refers to ECR (constraint softening variable) which is used for constraint softening weights.
ε_k is scalar QP slack variable for softening,
s_j^y and s_j^u stands for scale factor of outputs and manipulated variables,
$y_{j,min}$ and $y_{j,max}$ are lower and upper bounds of output variables,
$u_{j,min}$ and $u_{j,max}$ are lower and upper bounds of output variables.

For a quadcopter, there are several constraints which might be essential or beneficial. For example, hard constraints on control inputs are necessary to avoid unfeasible physical forces. In the system, used in this thesis, manipulated variables are:

$$\begin{bmatrix} -400 \\ -3 \\ -3 \\ -3 \end{bmatrix} \le \begin{bmatrix} \delta_{thr} \\ \delta_{ail} \\ \delta_{ele} \\ \delta_{rud} \end{bmatrix} \le \begin{bmatrix} 400 \\ 3 \\ 3 \\ 3 \end{bmatrix} \qquad (4.4)$$

$$\begin{bmatrix} -20\,m/s \\ -20\,m/s \\ -20\,m/s \\ -\frac{\pi}{4}\,rad/s \\ -\frac{\pi}{4}\,rad/s \\ -\frac{\pi}{4}\,rad/s \\ -\frac{\pi}{4}\,rad \\ -\frac{\pi}{4}\,rad \\ -\pi\,rad \end{bmatrix} \leq \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \\ \phi \\ \theta \\ \psi \end{bmatrix} \leq \begin{bmatrix} 20\,m/s \\ 20\,m/s \\ 20\,m/s \\ \frac{\pi}{4}\,rad/s \\ \frac{\pi}{4}\,rad/s \\ \frac{\pi}{4}\,rad/s \\ \frac{\pi}{4}\,rad \\ \frac{\pi}{4}\,rad \\ \pi\,rad \end{bmatrix} \tag{4.5}$$

Constraint softening: In some conditions, some hard constraints cause infeasible QP solution. If the designer has an extra margin, MPC can use this margin when it is necessary with constraint softening.

Constraint Softening Matrices of Output Variables:

$$minECR = \begin{bmatrix} 3 \\ 3 \\ 3 \\ 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \end{bmatrix} \tag{4.6}$$

$$maxECR = \begin{bmatrix} 3 \\ 3 \\ 3 \\ 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \end{bmatrix} \tag{4.7}$$

Constraint Softening Matrices of Manipulated Variables:

$$minECR = \begin{bmatrix} 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \end{bmatrix} \tag{4.8}$$

$$maxECR = \begin{bmatrix} 0.1 \\ 0.1 \\ 0.1 \\ 0.1 \end{bmatrix} \tag{4.9}$$

Scale Factor: If there is scaling differences in both control inputs and states, scale factor eases the weighting by normalizing them. The simplest way to specify scale factor is by using the span of the variables.

Scale Factors of Output Variables:

$$s^y = \begin{bmatrix} 10 \\ 10 \\ 10 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{bmatrix} \tag{4.10}$$

Scale Factors of Manipulated Variables:

$$s^u = \begin{bmatrix} 200 \\ 1 \\ 1 \\ 1 \end{bmatrix} \tag{4.11}$$

Weights and Cost Function: Similar to LQR, MPC uses weights both on control inputs and states to define cost function. Higher weight causes aggressive responses on the related variable. On the other hand, like constraints, it is sensible to avoid weight on states if it is unnecessary. Depending on the mode; weight matrices are selected differently.

During this thesis, attitude and speed mode controllers, which are the basic two modes of the reference autopilot module, have been replaced with the MPC controller. In order to enable the mode transition here, the online weighting has been added to the MPC. Additionally, to facilitate the pilot, the directional channel commands angular velocity, and holds the angle when an angular velocity command is received. Therefore, the angular velocity command input is controlled, and when it comes close to zero, the weight of the head is pulled to zero.

In the attitude mode, the inputs of the system are pitch angle, roll angle and yaw rate. The vertical channel is directly controlled by the pilot. In this mode, the body velocities weights have been set to zero. During flight tests, low weights in rate states caused oscillations. In the simulation, the oscillation observed with higher time delays. Increasing the rate states weights have significantly decreased the oscillations magnitude and also gave the system smoother responses.

$$W_{OV_{attitude}} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 1 \\ 1.5 \\ 5 \\ 5 \\ 5 \end{bmatrix} \tag{4.12}$$

$$W_{MV_{attitude}} = \begin{bmatrix} 0.2 \\ 0.2 \\ 0.2 \\ 0.2 \end{bmatrix} \tag{4.13}$$

In speed mode, the main objective is tracking the pilot's speed, and yaw rate commands with less overshoot and lower rise time. Higher weights are given in controller design to increase the aggressiveness of the controller in these channels. Also, pitch and roll channels do not have weight (they are equal to 0) to give controller flexibility. Roll rate, pitch rate and yaw rate also have weights to prevent oscillations.

$$W_{OV_{speed}} = \begin{bmatrix} 10 \\ 10 \\ 10 \\ 1 \\ 1 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.14}$$

$$W_{MV_{speed}} = \begin{bmatrix} 0.2 \\ 0.2 \\ 0.2 \\ 0.2 \end{bmatrix} \tag{4.15}$$

QP Solver: MPC uses QP solver to find optimal control sequence. In this thesis, active-set method solver has been selected for optimal performance.

Model Time Delay: During flight tests, different model time delays have been tried to observe its effect on system stability. Tests showed that modelling of the time delay is crucial for stability. And, the best performance has been observed with 80 ms model time delay. Around 10-15ms calculation time of MPC and approximately 65-70 ms physical time delay, which was found through the system identification process.

4.2. Creating MPC Controllers in Simulink with Linear Model

To create the MPC object, MPC designer has been work during the thesis. After that, the MPC object has been implemented in a Simulink model to test its performance. This Simulink model contains three major parts which are controller, input generator and dynamic model. To generate test input, Signal Generator has been used. The generated inputs have shown in section 5. On the other hand, the dynamic model contains the identified linear model. The identification processes and the linear model can be found in section 3. In this section, the details of the controller block will be covered.

MPC Controller has three blocks and a reference selector switch. The MPC block contains MPC object, yaw - yaw-rate selector block, MPC mode selector block, and MPCMode - OVweight block.

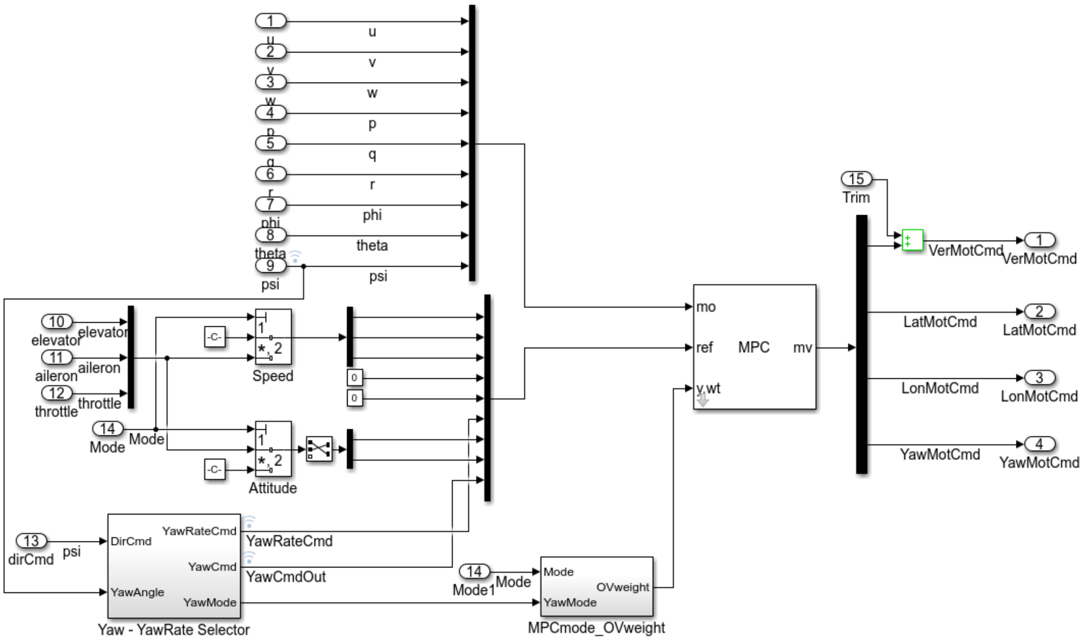


Figure 4-1 MPC scheme in Simulink

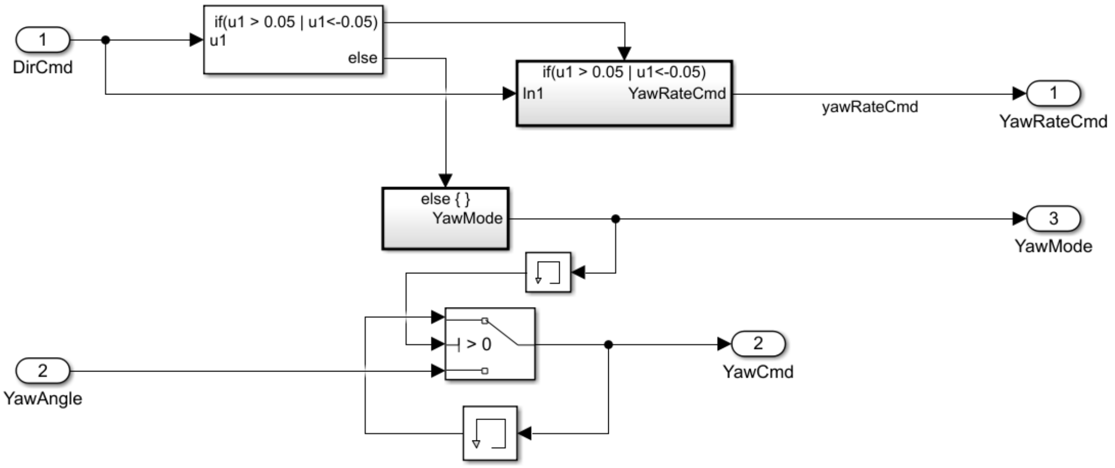


Figure 4-2 Yaw – Yaw rate selector

Yaw mode selector has been designed to control yaw angle while the reference is zero. The controller sets the yaw angle weight to 2 if the directional input is zero, and 0 if not.

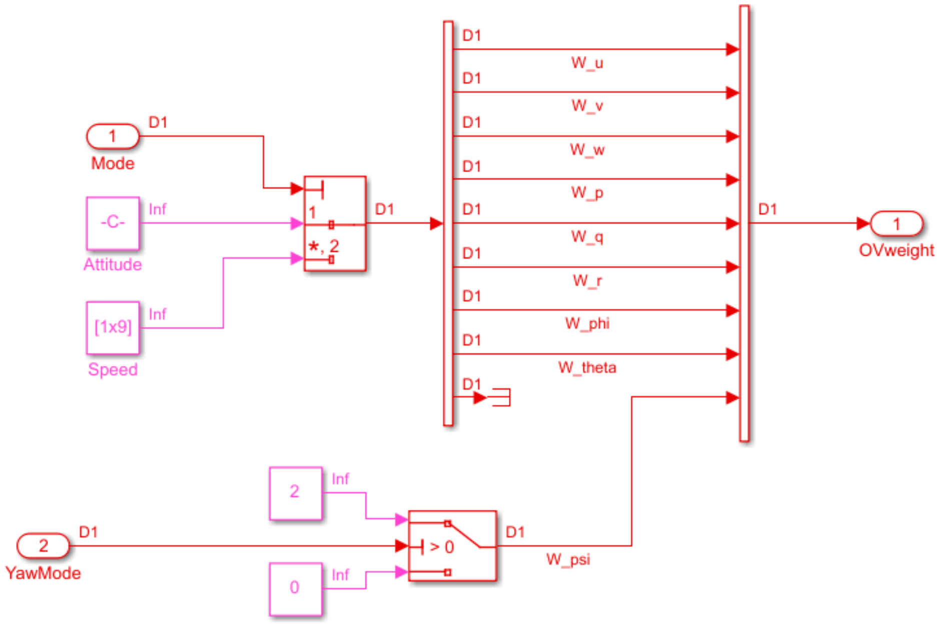


Figure 4-3 MPCmode_OVweight

MPCmode_OVweight block is responsible with the changing the MPC weights while mode changes.

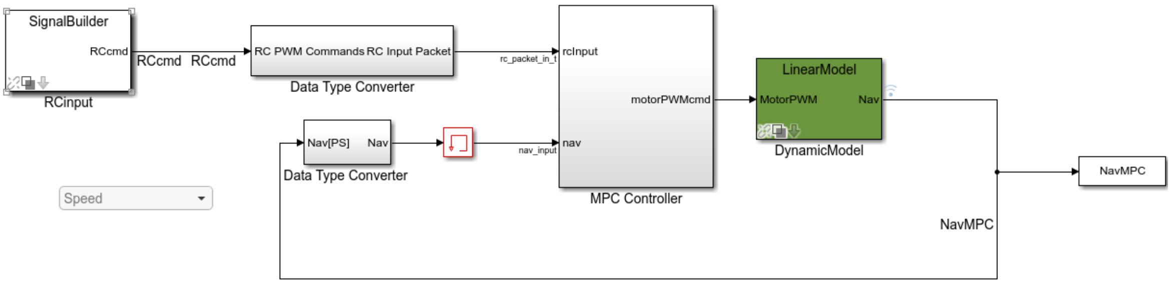


Figure 4-4 MPC simulation model

The simulation model basically consists of three parts. These are the RCInput block where the reference input is generated, the Controller block containing the MPC, the DynamicModel block that contains the linear plant.

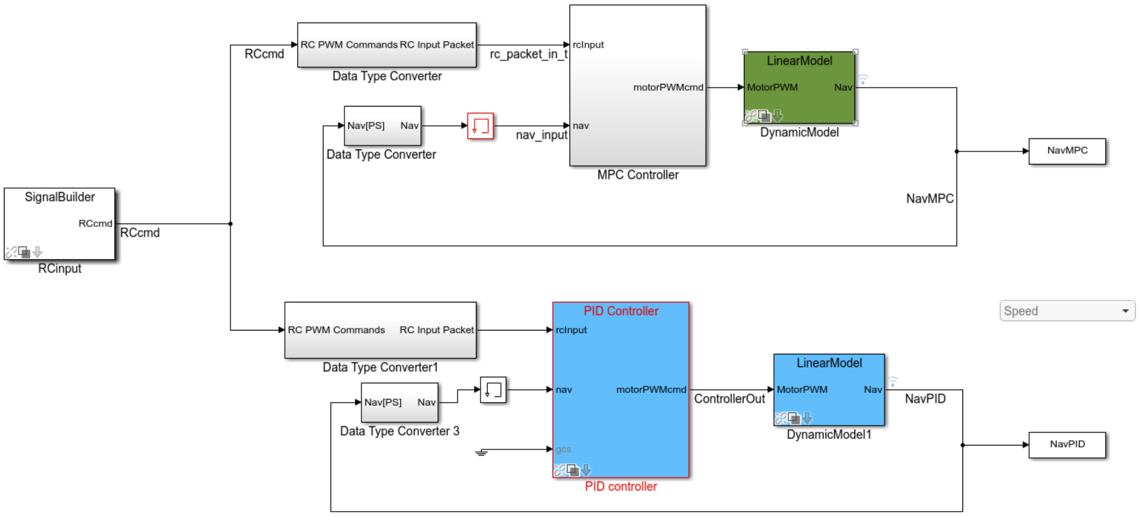


Figure 4-5 MPC-PID comparison Simulink model

5. RESULTS

This section includes the results of the tests run on the Simulink model shown in Figure 4-4. To compare MPC's performance, a cascaded PID controller has been used.

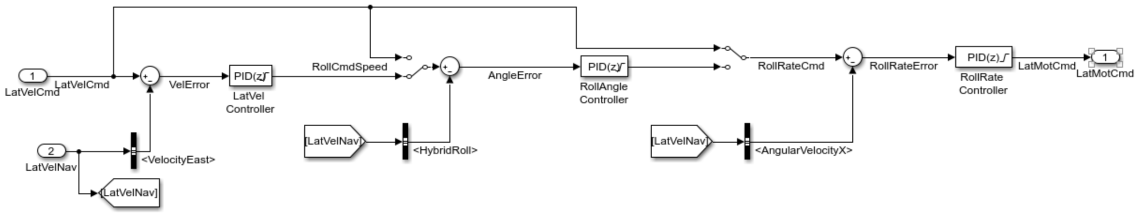


Figure 5-1 Cascaded PID contoller for lateral and longitudinal dynamics

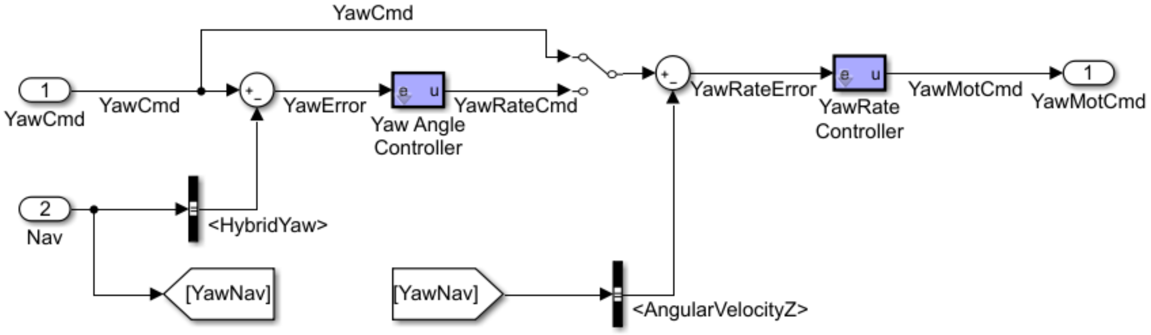


Figure 5-2 Cascaded PID loop for directional dynamic

The gains of the PID controllers have been tabulated in Table 5-1.

Table 5-1 PID controller gains

Channel	Loop	Kp	Ki	Kd	CutOff Frequency	Anti-wind up	Saturation Min	Saturation Max
Lateral and Longitudinal	Position	0.2	0.1	0	-	1.1	-2	2
	Velocity	0.55	0.05	0.015	50	1.1	-1	1
	Angle	1	0	0	-	1.1	-4	4
	Rate	0.5	1	0.015	50	1.1	-5	5
Vertical	Position	1.65	0.45	0	-	1.1	-2	2
	Velocity	3.85	4.95	0	-	1.1	-2	2
Directional	Angle	1.35	0.22	0	-	1.1	-1	1
	Rate	1	2.57	0	-	1.1	-1	1

5.1. Simulation Results

5.1.1. Results of Attitude Mode

In this section, the results of the model in the loop tests against the doublet manoeuvre have been shown. As the pilot in attitude mode controls the vertical channel, no input is given to the vertical channel during this test. In order to keep the system stable, this channel input was fed as zero in the dynamic model during the simulation.

5.1.1.1. Directional Channel

In the directional channel, if the input is non-zero, the controller operates in the angular velocity loop, and if the command is zero, the controller operates in the angle loop.

Test Input:

To perform this test, 0.3 rad/s step yaw-rate command has been applied.

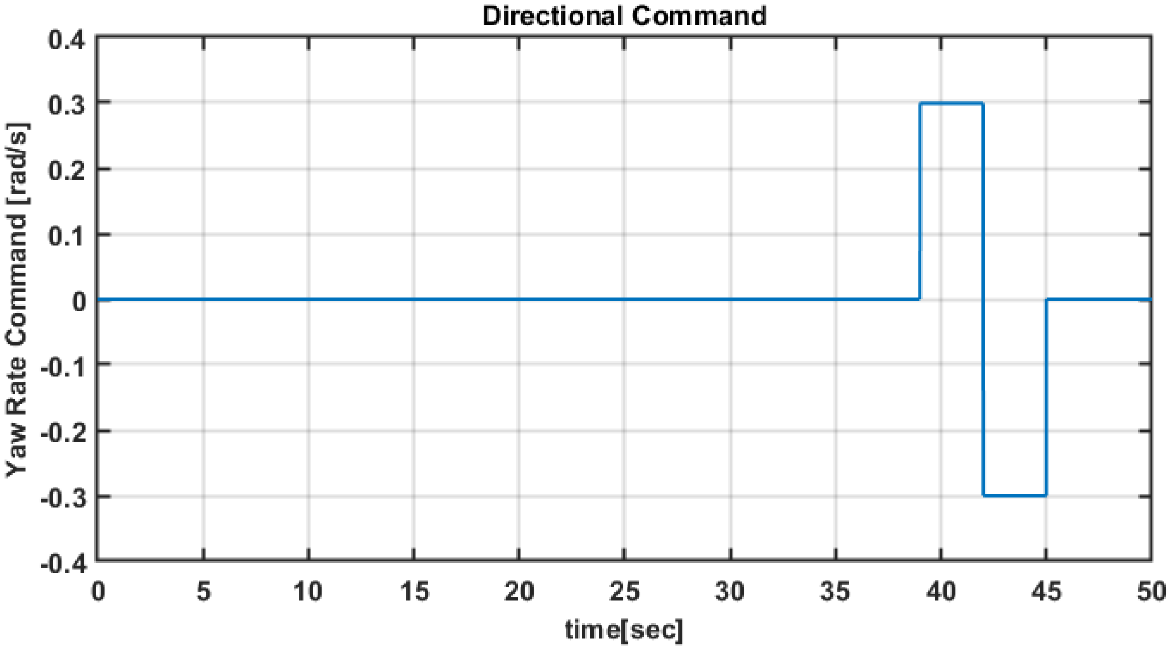


Figure 5-3 MPC vs PID - Directional command

Responses:

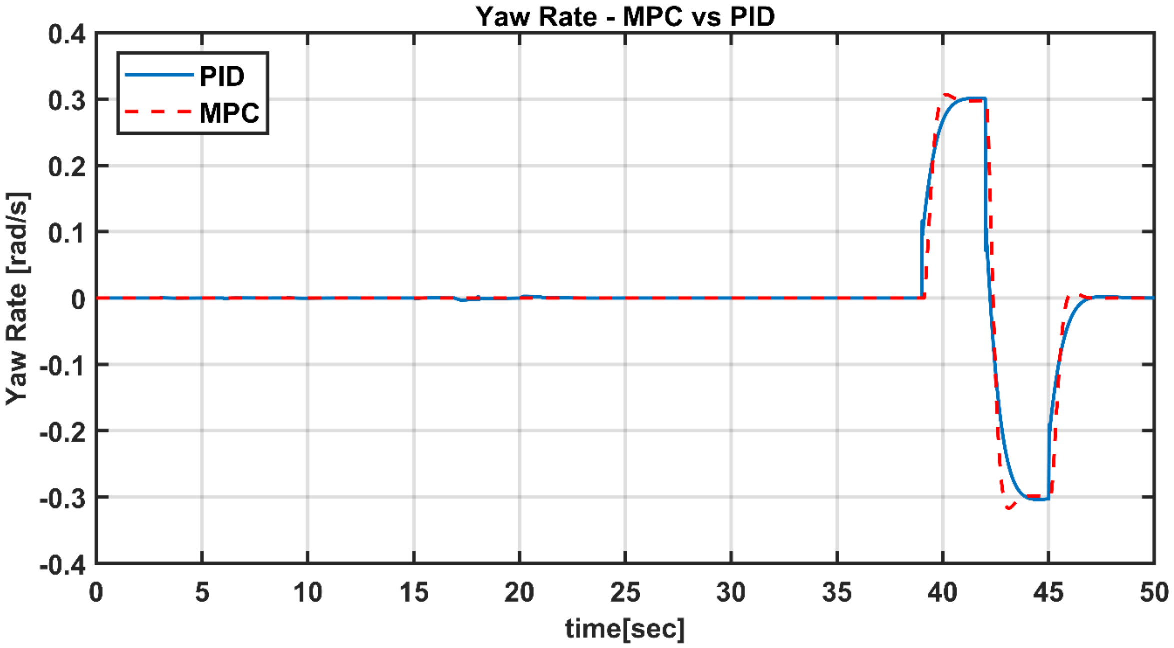


Figure 5-4 MPC vs PID - Yaw rate responses

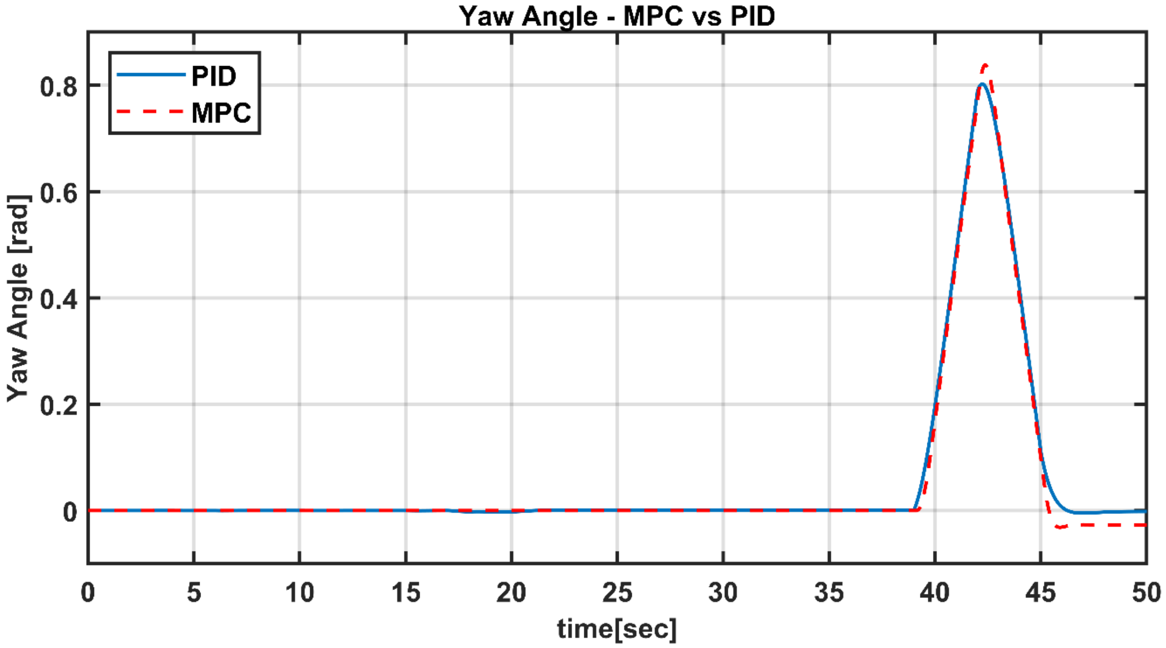


Figure 5-5 MPC vs PID - Yaw angle responses

5.1.1.2. Lateral Channel

In the lateral channel, the system input is given as angle command. Test input is shown in Figure 5-6. In the simulation of this mode, the integrator and derivative terms in PID controller are not included in the angle loop. For this reason, even if the controller has steady state error in the simulation, the controller performance is satisfactory for manual mode.

Test Input:

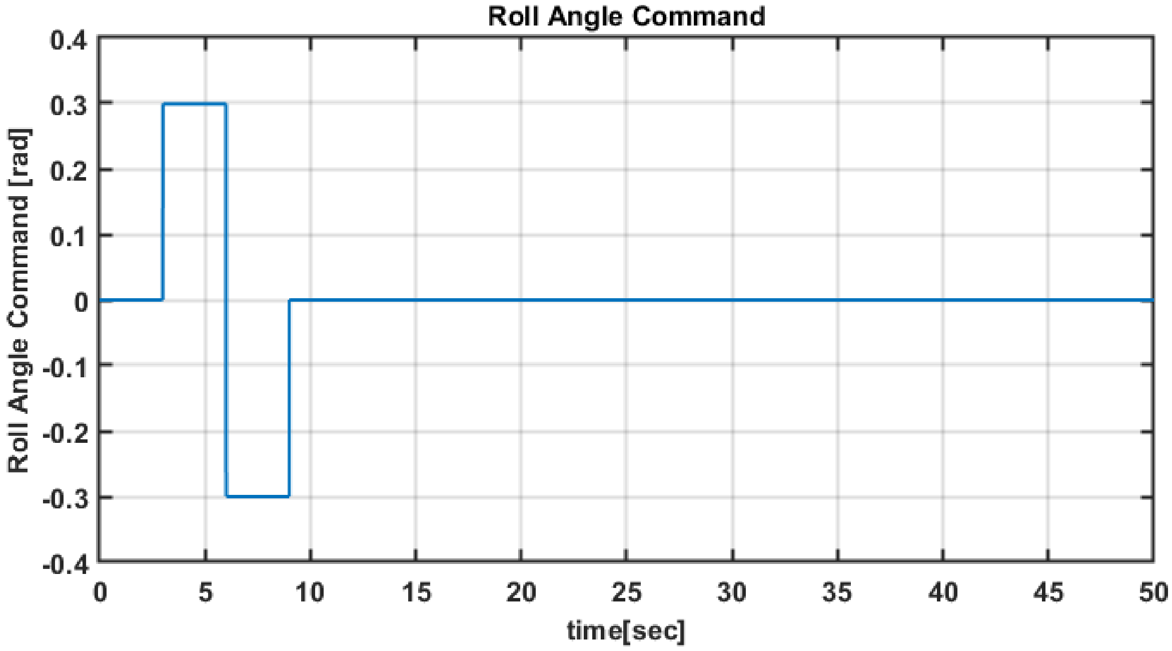


Figure 5-6 MPC vs PID - Roll command

Responses:

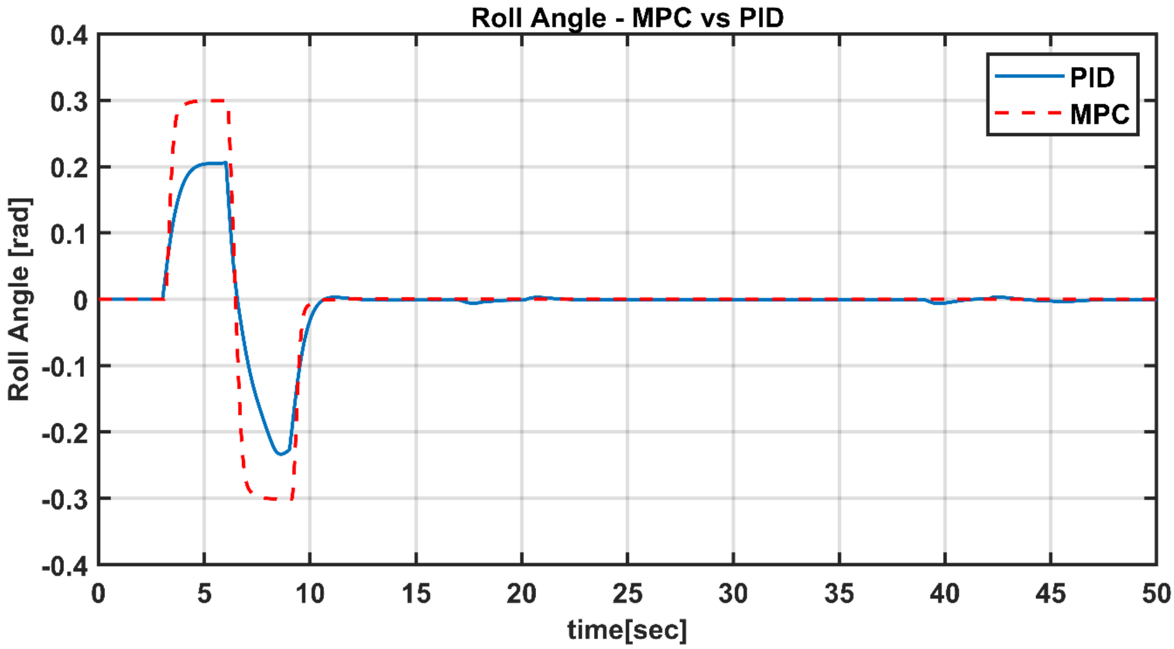


Figure 5-7 MPC vs PID - Roll angle response

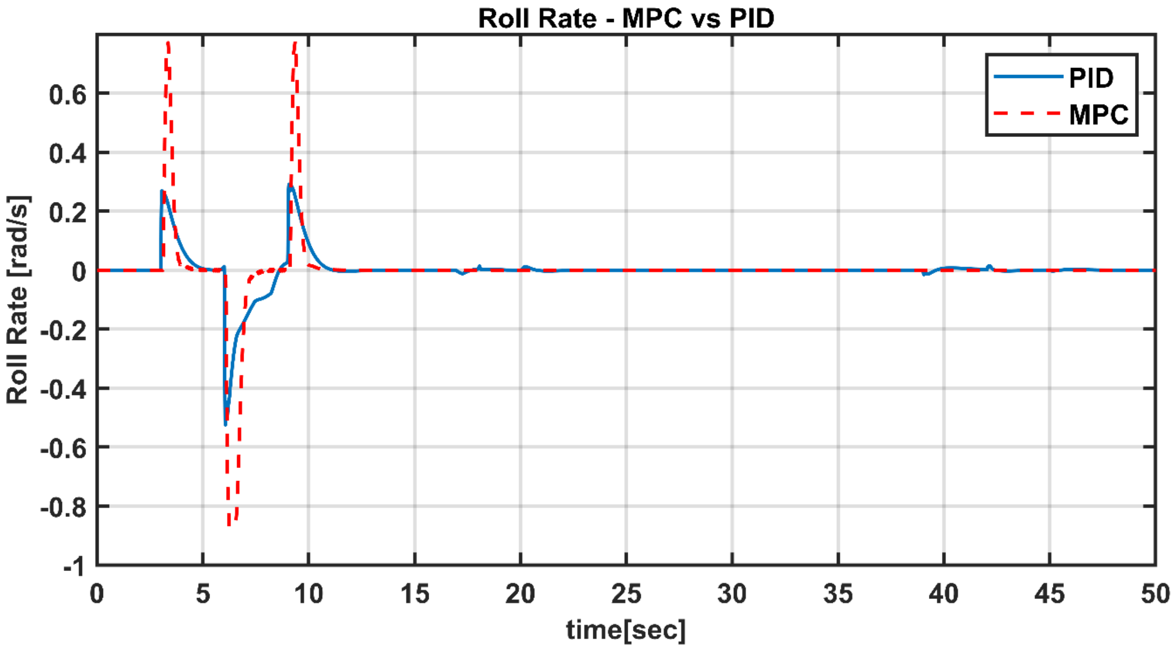


Figure 5-8 MPC vs PID - Roll rate response

5.1.1.3. Longitudinal Channel

In the longitudinal channel, the system input is given as angle command. Test input is shown in Figure 5-9. In the simulation of this mode, the terms integrator and derivative in PID controller are not included in the angle loop. For this reason, even if the controller has steady state error in the simulation, the controller performance is satisfactory for manual mode.

Test Input:

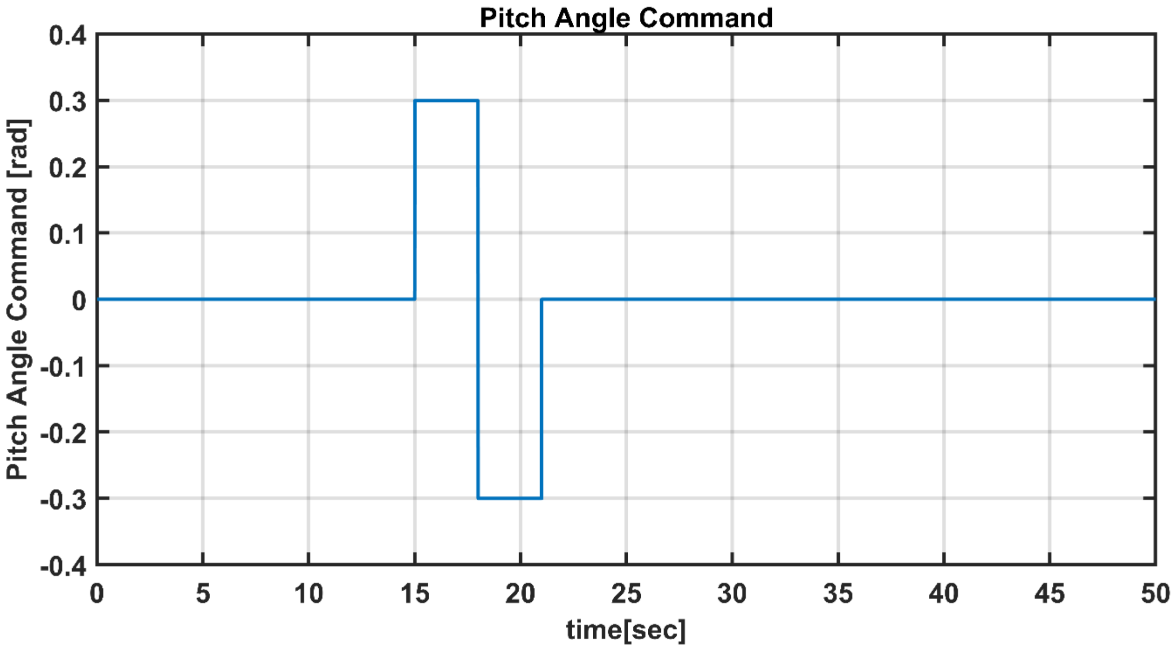


Figure 5-9 MPC vs PID - Pitch command

Responses:

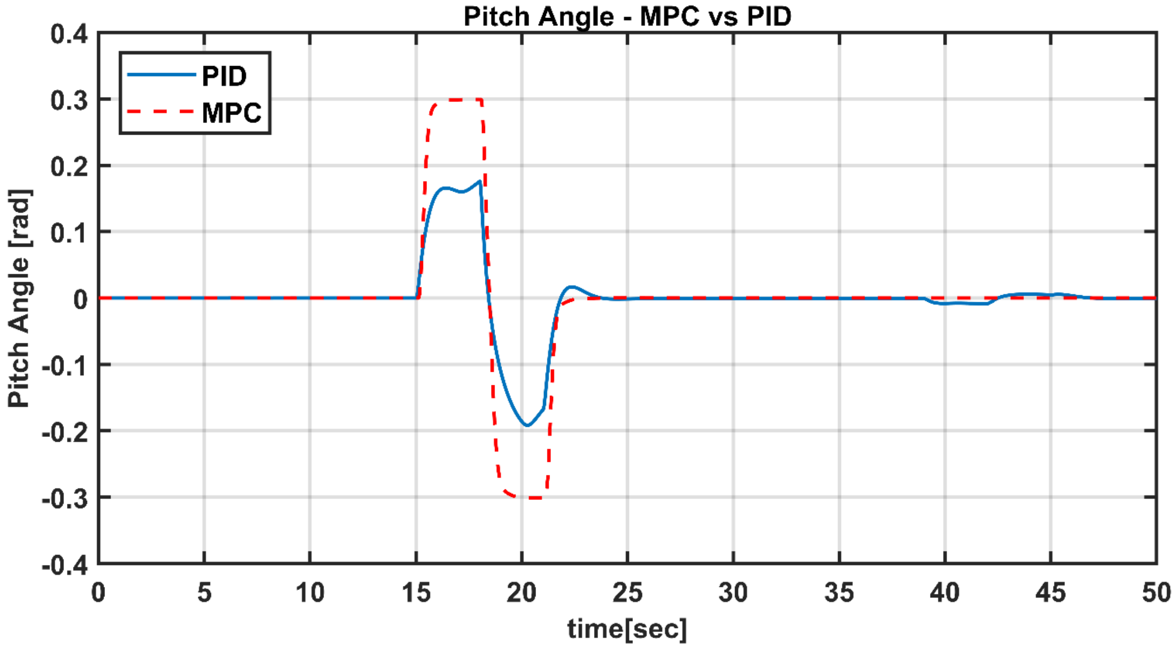


Figure 5-10 MPC vs PID - Pitch angle response

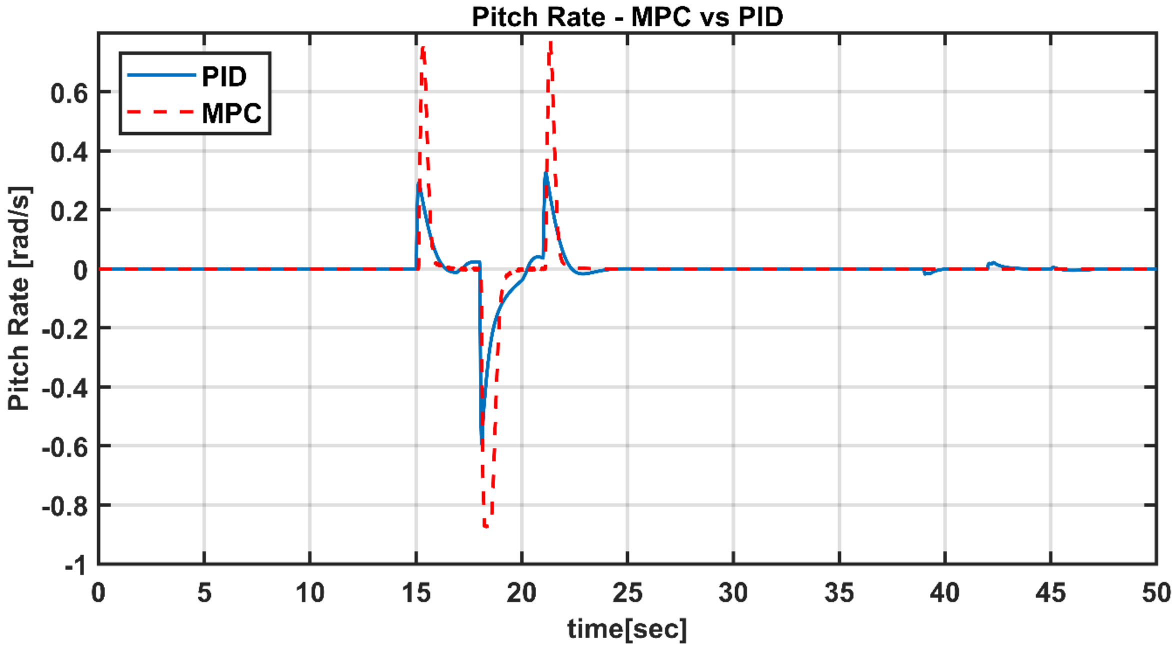


Figure 5-11 MPC vs PID - Pitch rate response

5.1.2. Results of Speed Mode

In this section, the results of the model in the loop tests against the doublet manoeuvre have been shown. During the tests performed in this mode, the pilot commanded speed in the longitudinal, lateral and vertical channels, and yaw rate in the directional channel.

5.1.2.1. Directional Channel

In the directional channel, if the input is non-zero, the controller operates in the angular rate, and if the command is zero, the controller operates in the angle loop.

Test Input:

To perform this test, 0.3 rad/s yaw rate command has been applied.

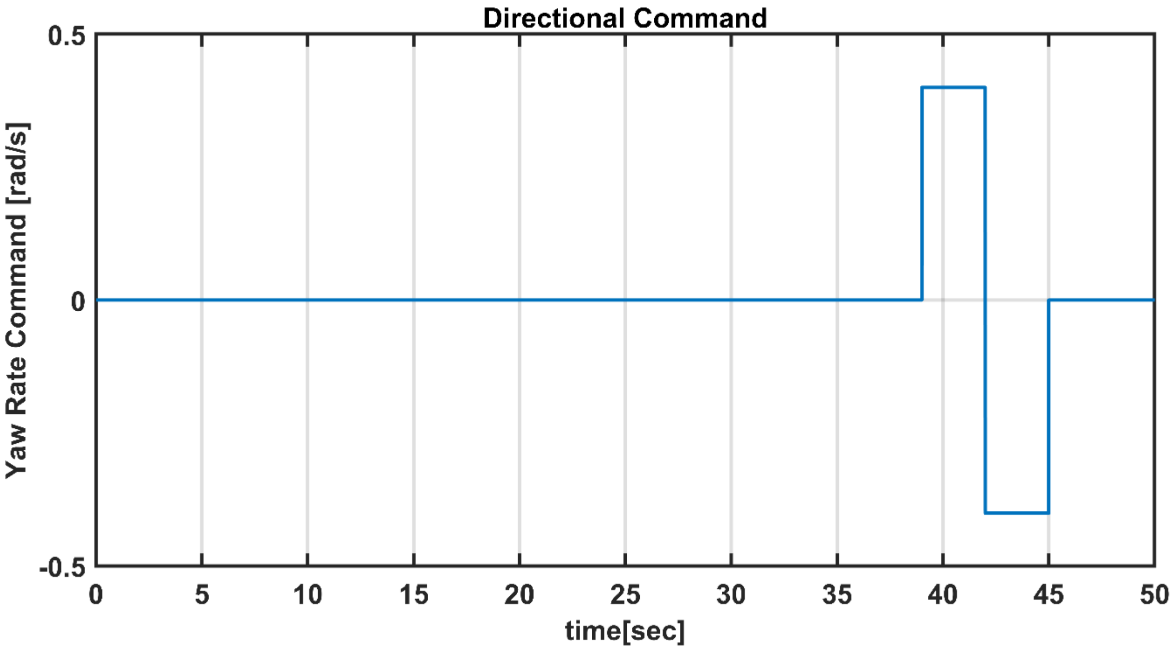


Figure 5-12 MPC vs PID - Directional command

Responses:

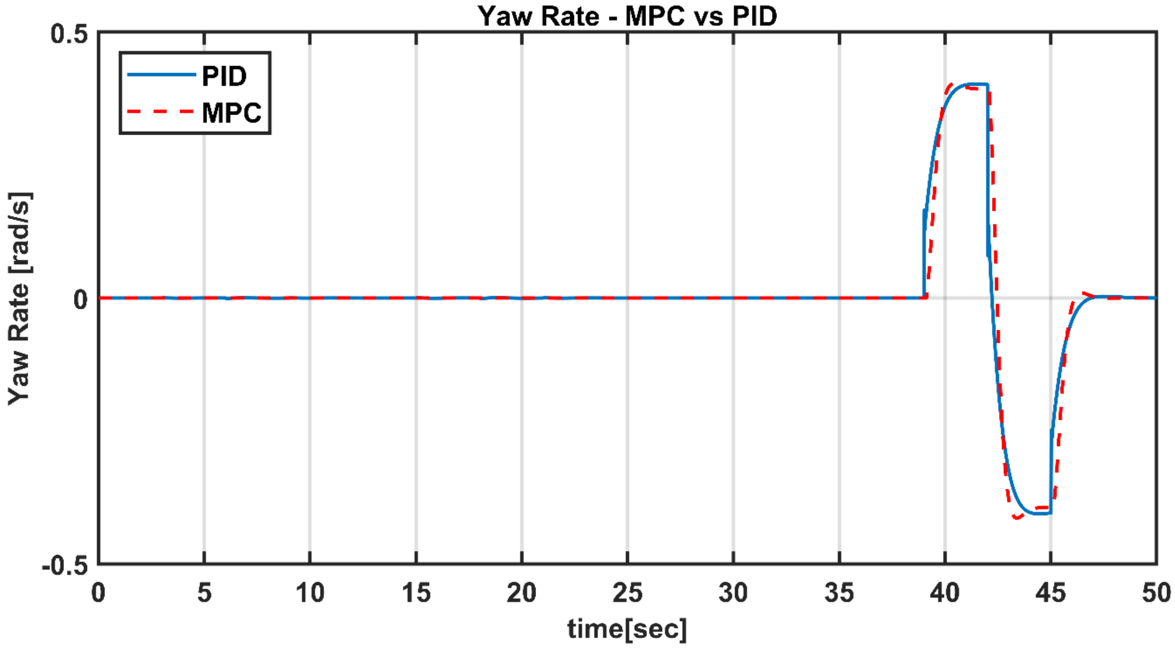


Figure 5-13 MPC vs PID - Yaw rate responses

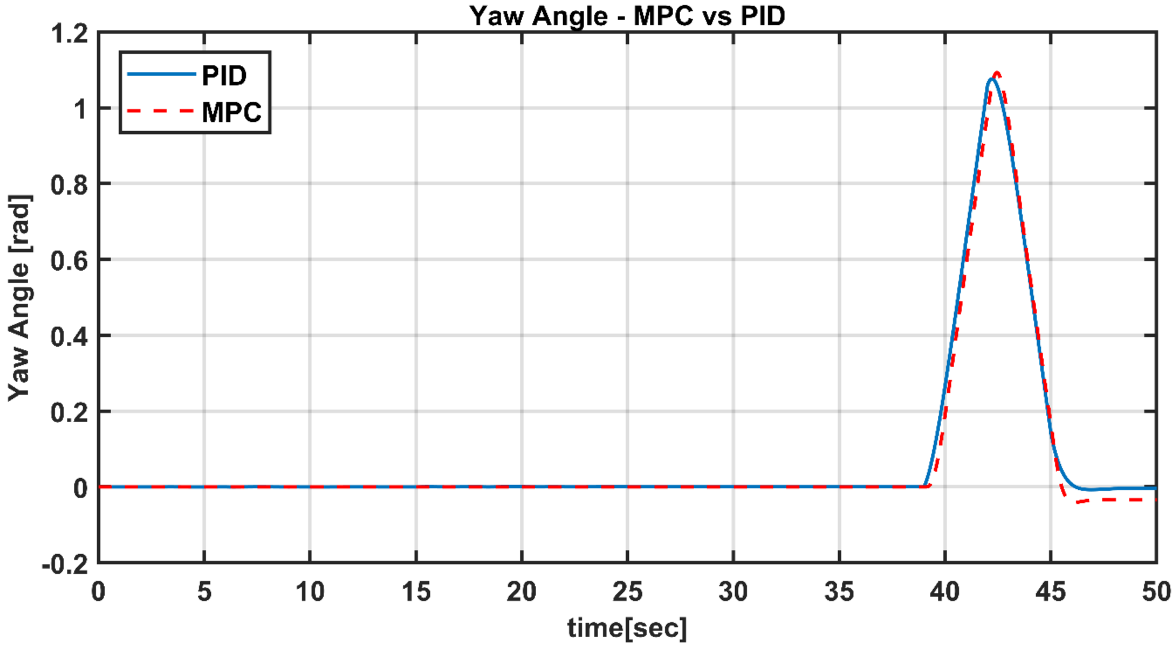


Figure 5-14 MPC vs PID - Yaw angle responses

As it can be seen in Figure 5-13, Figure 5-14 both controllers has a satisfying response in the directional channel.

5.1.2.2. Lateral Channel

In the lateral channel, the system input is given as angle command. Test input is shown in Figure 5-15.

Test Input:

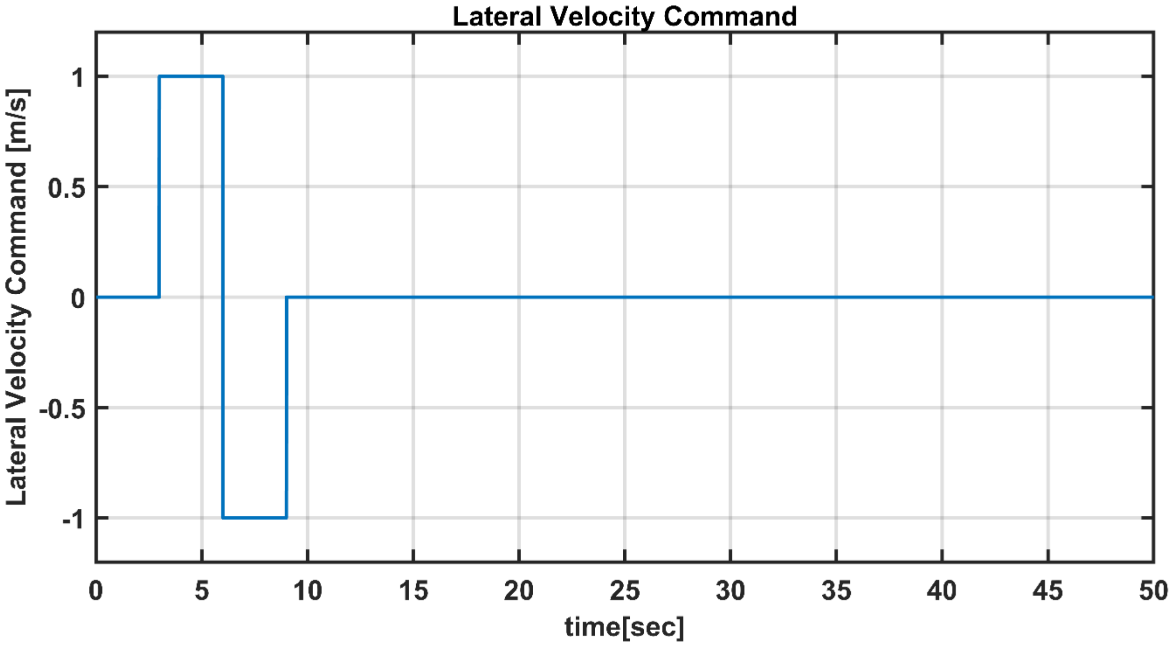


Figure 5-15 MPC vs PID – Lateral velocity command

Responses:

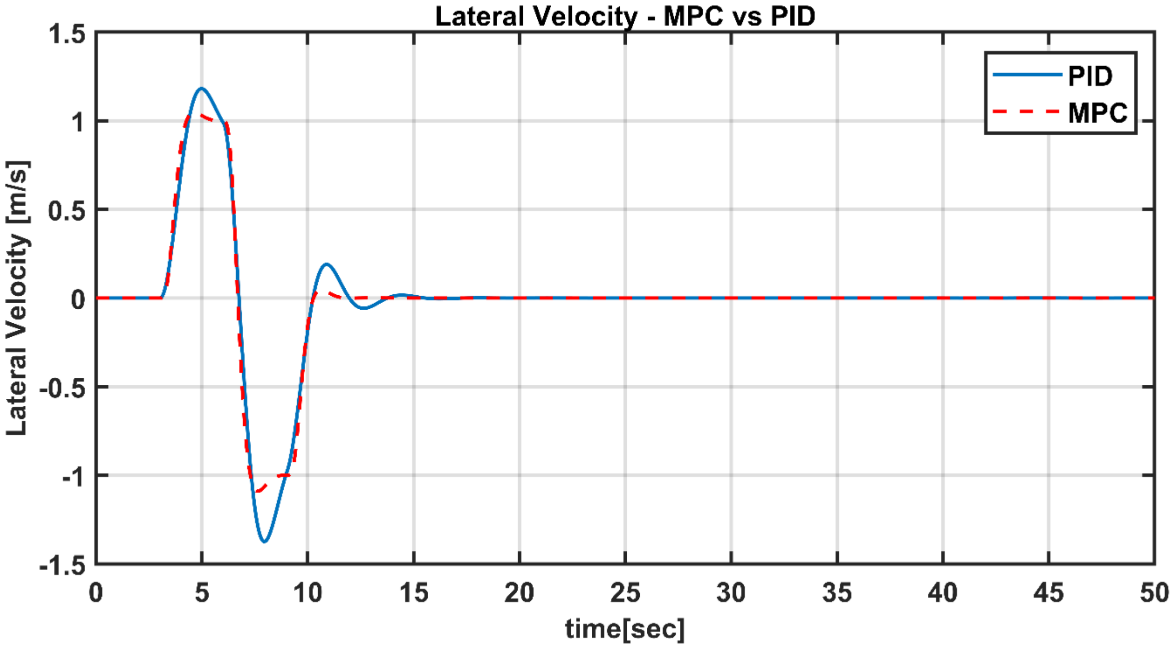


Figure 5-16 MPC vs PID – Lateral velocity response

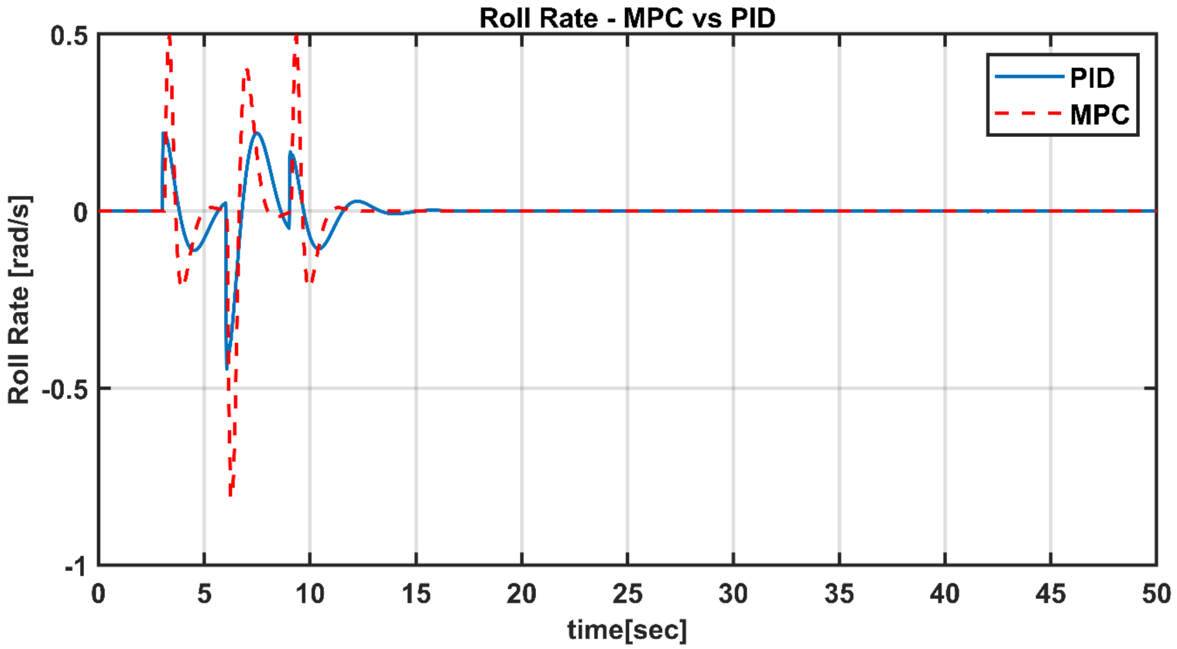


Figure 5-17 MPC vs PID - Roll rate response

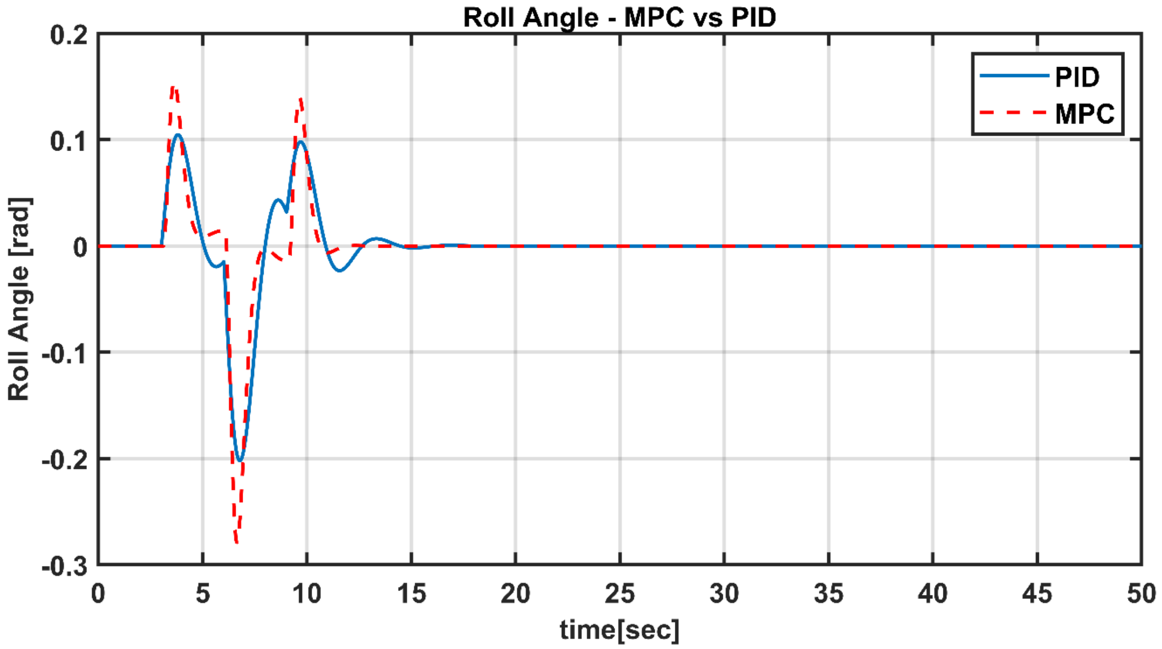


Figure 5-18 MPC vs PID – Roll angle response

In the lateral channel, MPC has a faster response with less overshoot. Also, it has less oscillation than PID.

5.1.2.3. Longitudinal Channel

In the longitudinal channel, the system input is given as angle command. Test input is shown in Figure 5-47.

Test Input:

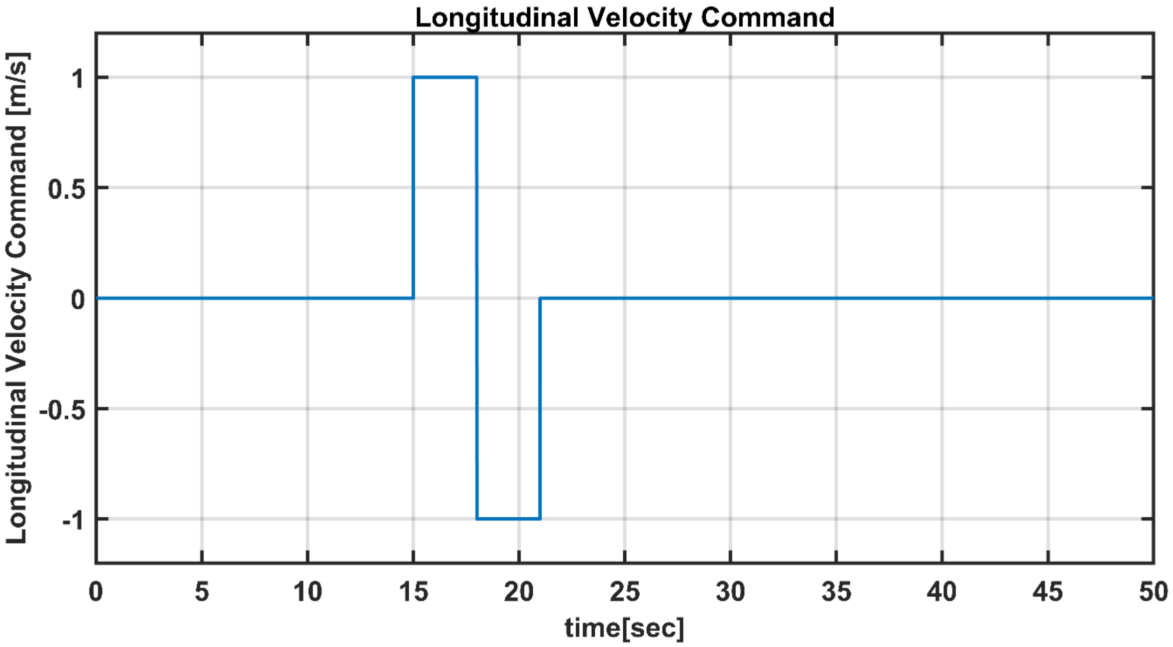


Figure 5-19 MPC vs PID – Longitudinal velocity command

Responses:

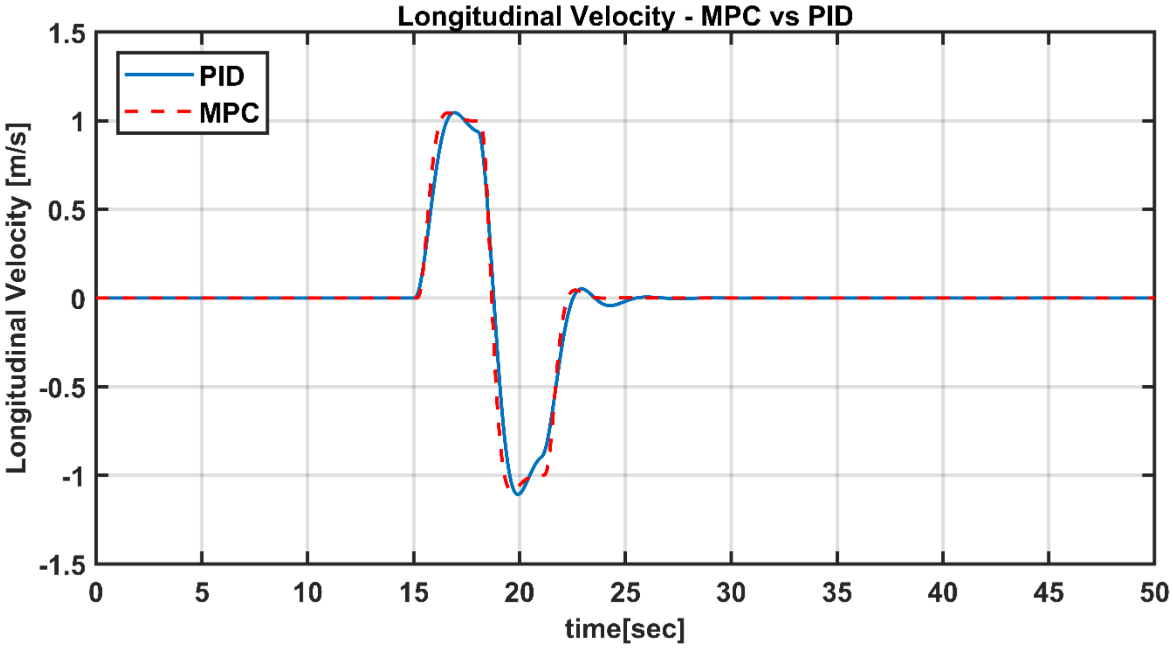


Figure 5-20 MPC vs PID – Longitudinal velocity response

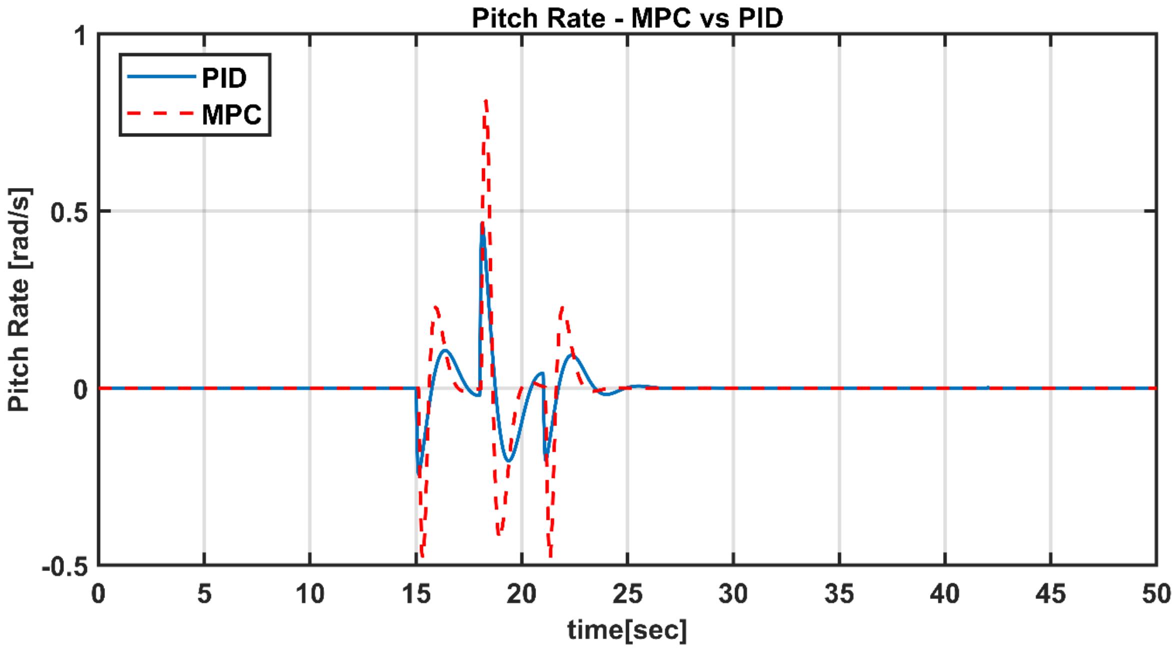


Figure 5-21 MPC vs PID - Pitch rate response

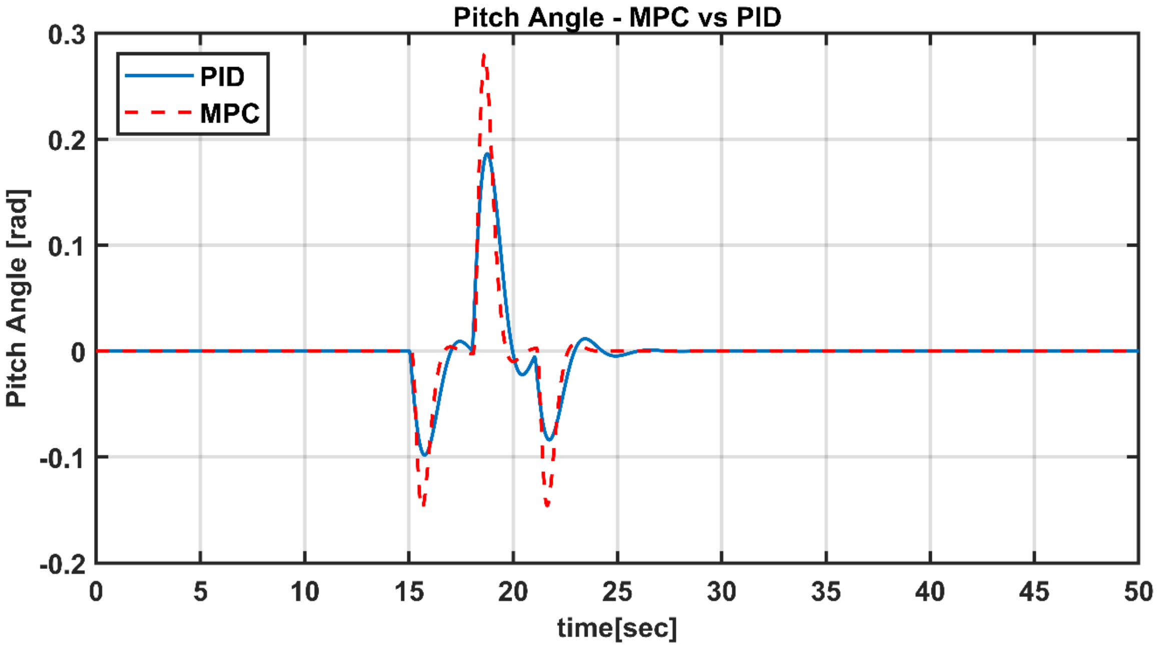


Figure 5-22 MPC vs PID – Pitch angle response

In the longitudinal channel, MPC has a faster response, but they have similar overshoot. Also, it has less oscillation than PID.

Vertical Channel

In the longitudinal channel, the system input is given as angle command. Test input is shown in Figure 5-23.

Test Input:

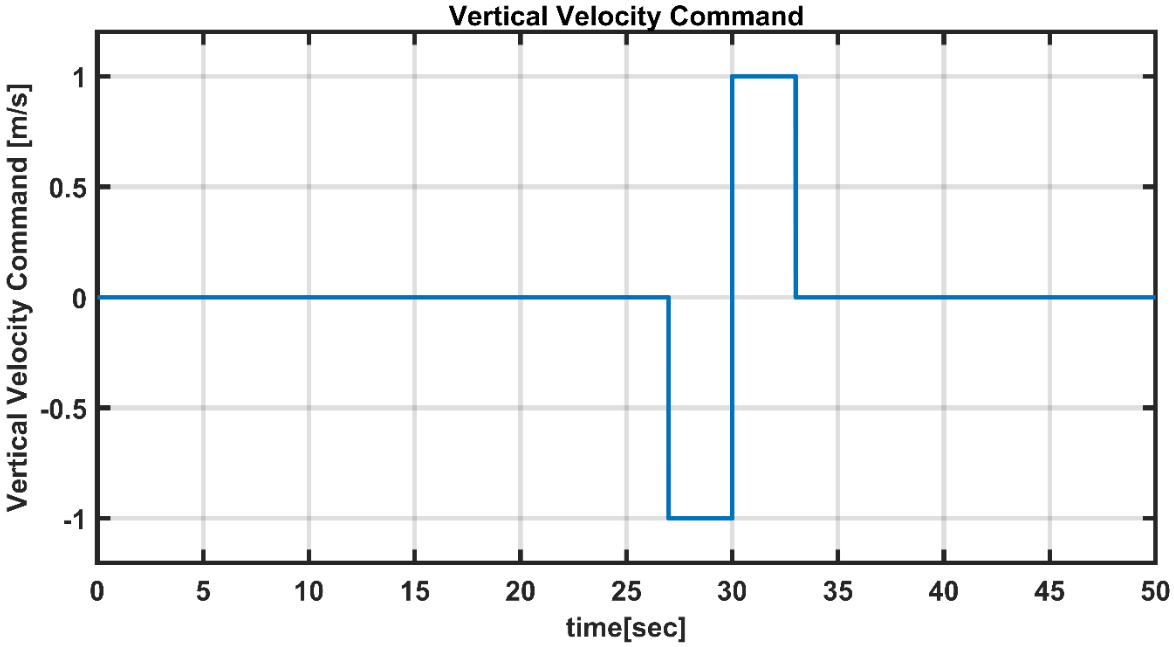


Figure 5-23 MPC vs PID – Vertical velocity command

Responses:

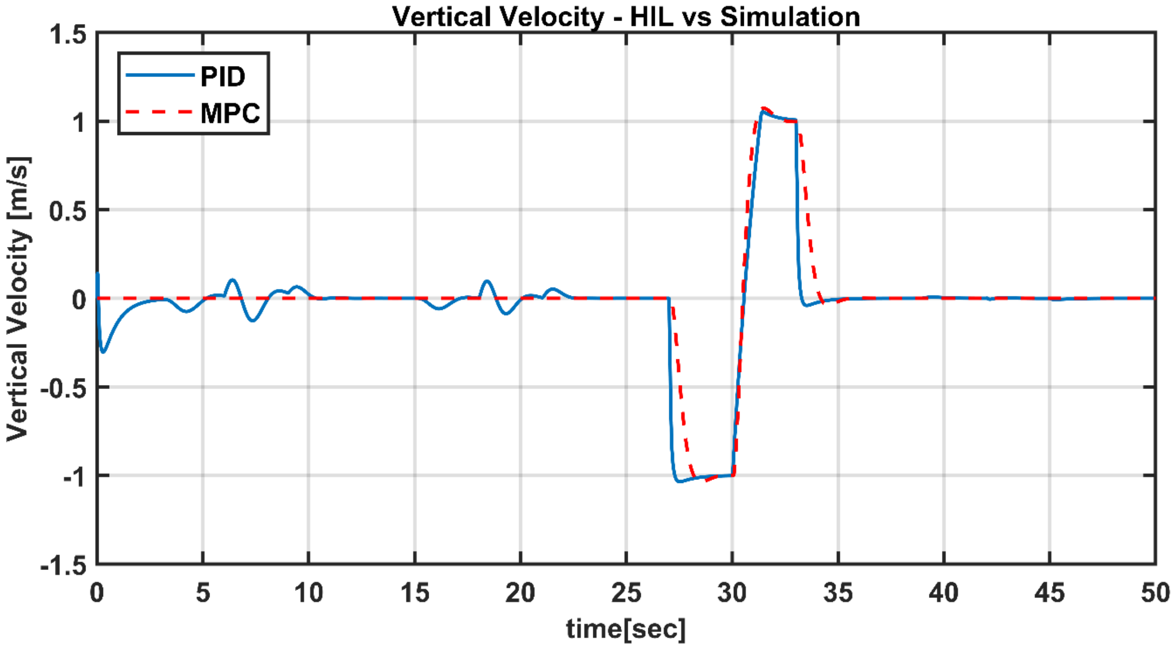


Figure 5-24 MPC vs PID – Vertical velocity response

For the vertical channel, PID's response was faster, and it has an acceptable overshoot. But there is another remarkable point. PID controller has deviations during other channels inputs. However, MPC can handle these inputs without deviation.

5.1.3. Results of Attitude Mode Under Input Disturbance

Input disturbance rejection performance has been tested via step inputs. In these tests, PID performed better than MPC. The state estimator has been selected as faster in MPC Designer to increase the input disturbance rejection performance of MPC. This improvement gave MPC slightly better performance, but PID still responded better during simulations.

In this section, the results of simulations against the step input have been shown. As the pilot in attitude mode controls the vertical channel, no input is given to the vertical channel during this test. In order to keep the system stable, this channel input was fed as zero in the dynamic model during the test.

System model for disturbance rejection mentioned below.

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \\ \dot{p} \\ \dot{q} \\ \dot{r} \\ \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} = \begin{bmatrix} x_u & 0 & 0 & 0 & 0 & 0 & 0 & -g & 0 \\ 0 & Y_v & 0 & 0 & 0 & 0 & g & 0 & 0 \\ 0 & 0 & Z_w & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & L_v & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ M_u & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & N_r & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} u \\ v \\ w \\ p \\ q \\ r \\ \phi \\ \theta \\ \psi \end{bmatrix} + \begin{bmatrix} 0 & 0 & X_{\delta_{ele}} & 0 \\ 0 & Y_{\delta_{ail}} & 0 & 0 \\ Z_{\delta_{thr}} & 0 & 0 & 0 \\ 0 & L_{\delta_{ail}} & 0 & 0 \\ 0 & 0 & M_{\delta_{ele}} & 0 \\ 0 & 0 & 0 & N_{\delta_{rud}} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_{thr} \\ \delta_{ail} \\ \delta_{ele} \\ \delta_{rud} \end{bmatrix} \qquad (5.1)$$

5.1.3.1. Directional Channel

Test Input:

To perform this test, 0.2 directional command (δ_{rud}) has been applied as input to dynamic model.

Responses:

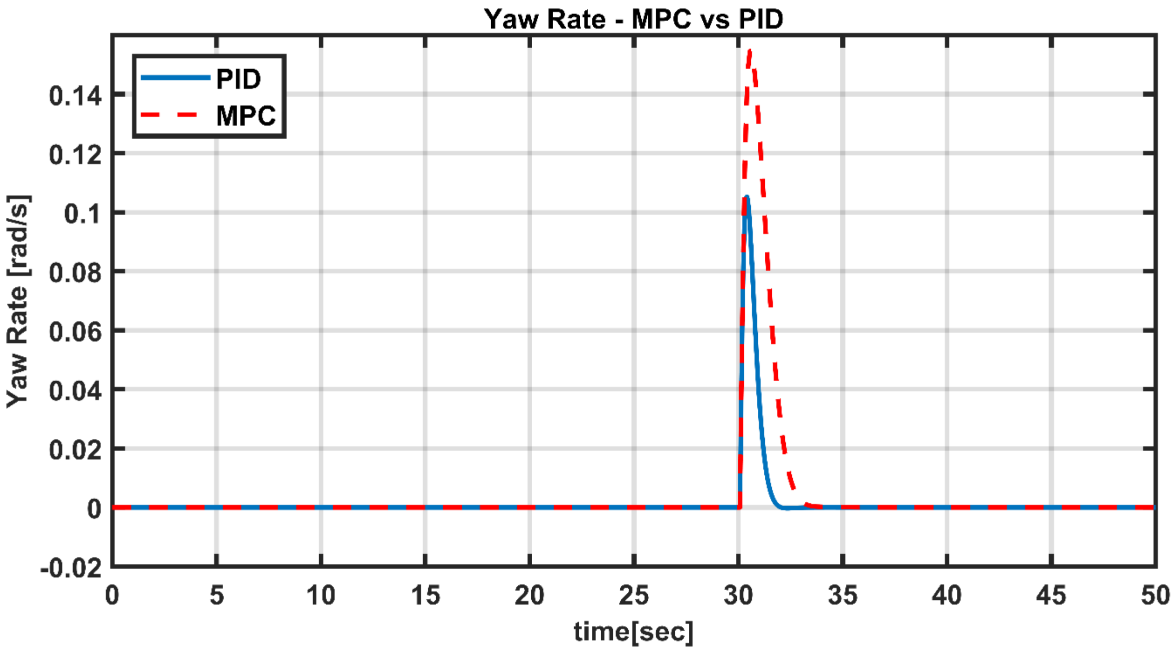


Figure 5-25 Input disturbance rejection - Yaw rate responses

5.1.3.2. Lateral Channel

In the lateral channel, the system input is given as angle command.

Test Input:

To perform this test, 0.2 lateral command (δ_{ail}) has been applied as input to dynamic model.

Responses:

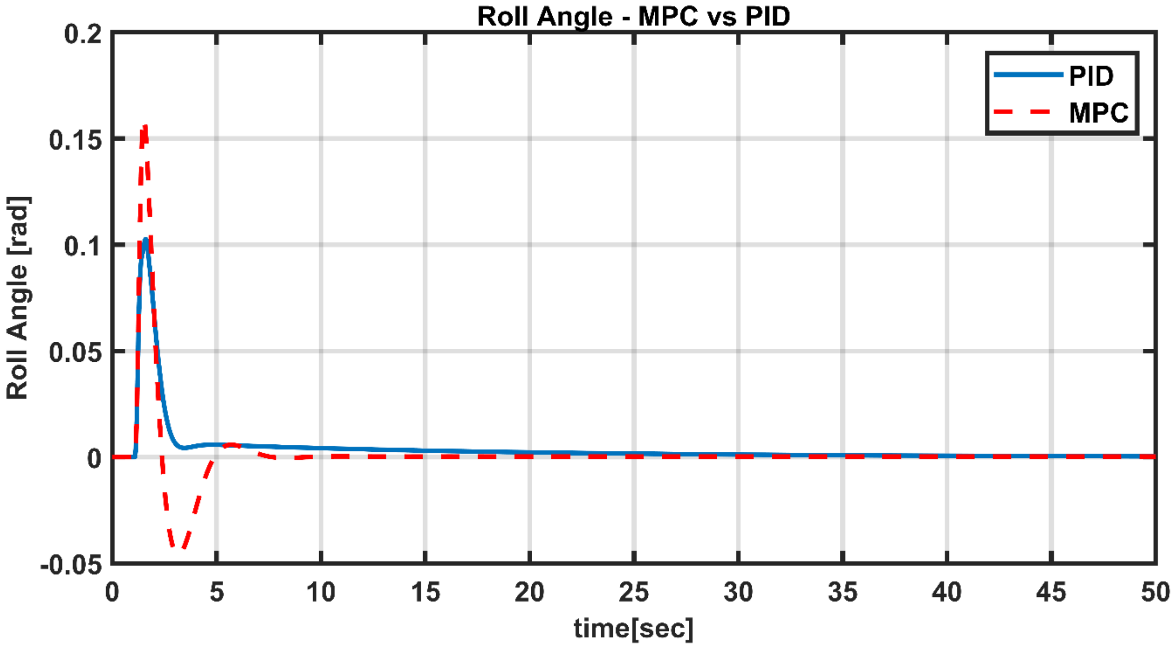


Figure 5-26 Input disturbance rejection - Roll angle response

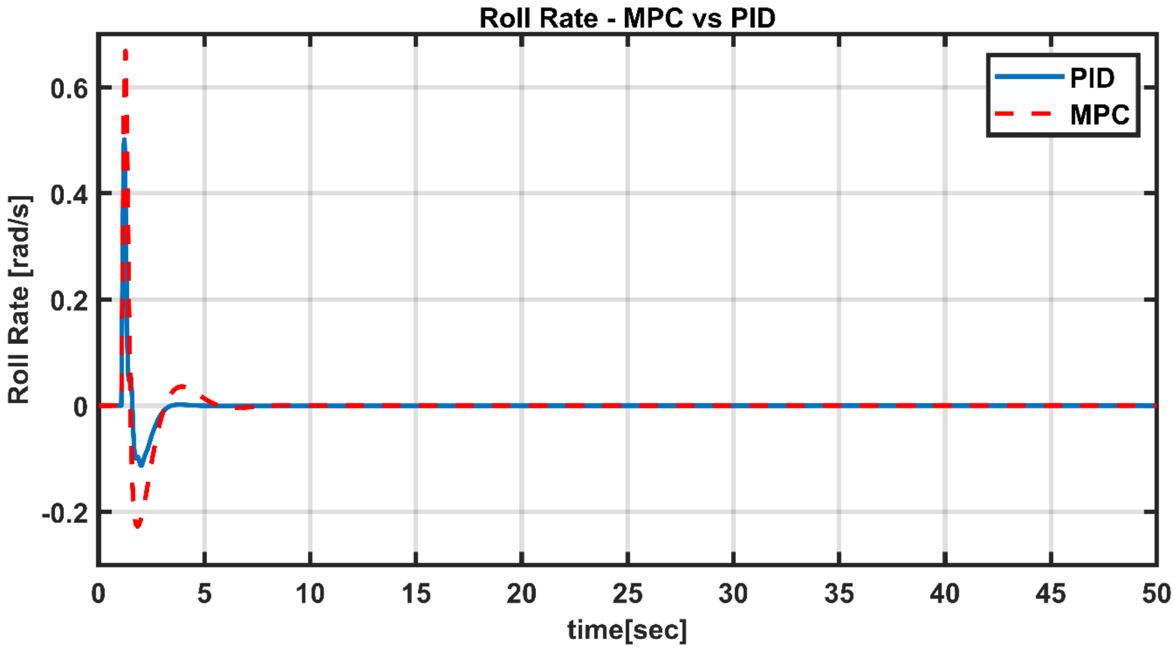


Figure 5-27 Input disturbance rejection - Roll rate response

5.1.3.3. Longitudinal Channel

In the longitudinal channel, the system input is given as angle command.

Test Input:

To perform this test, 0.2 longitudinal command (δ_{ele}) has been applied as input to dynamic model.

Responses:

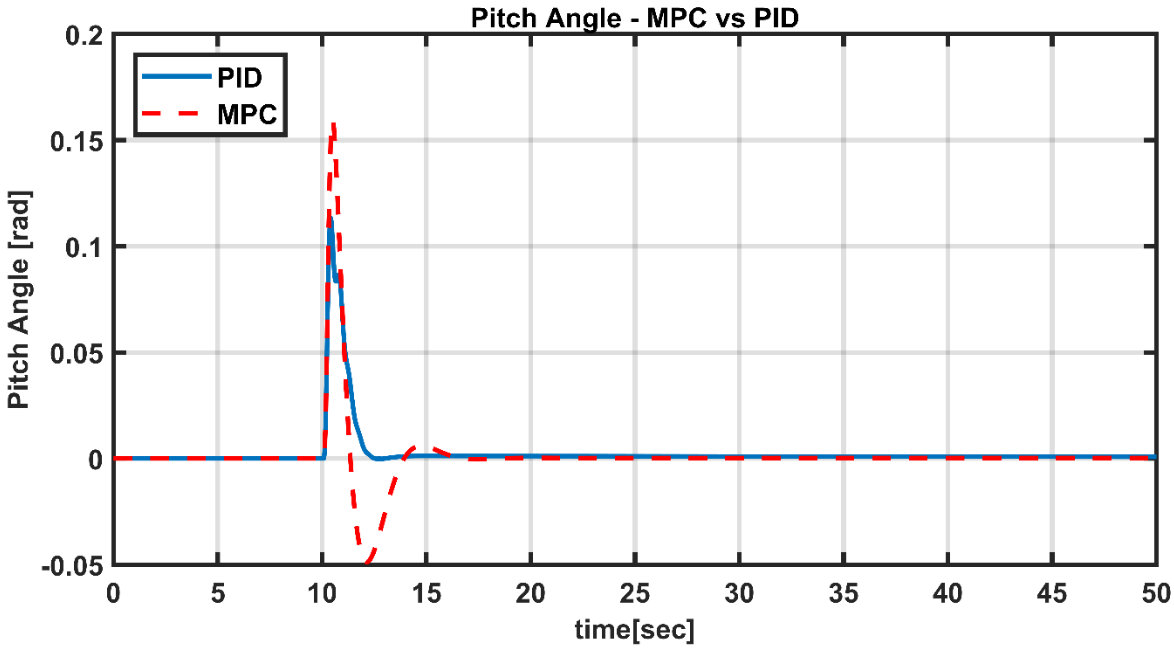


Figure 5-28 Input disturbance rejection - Pitch angle response

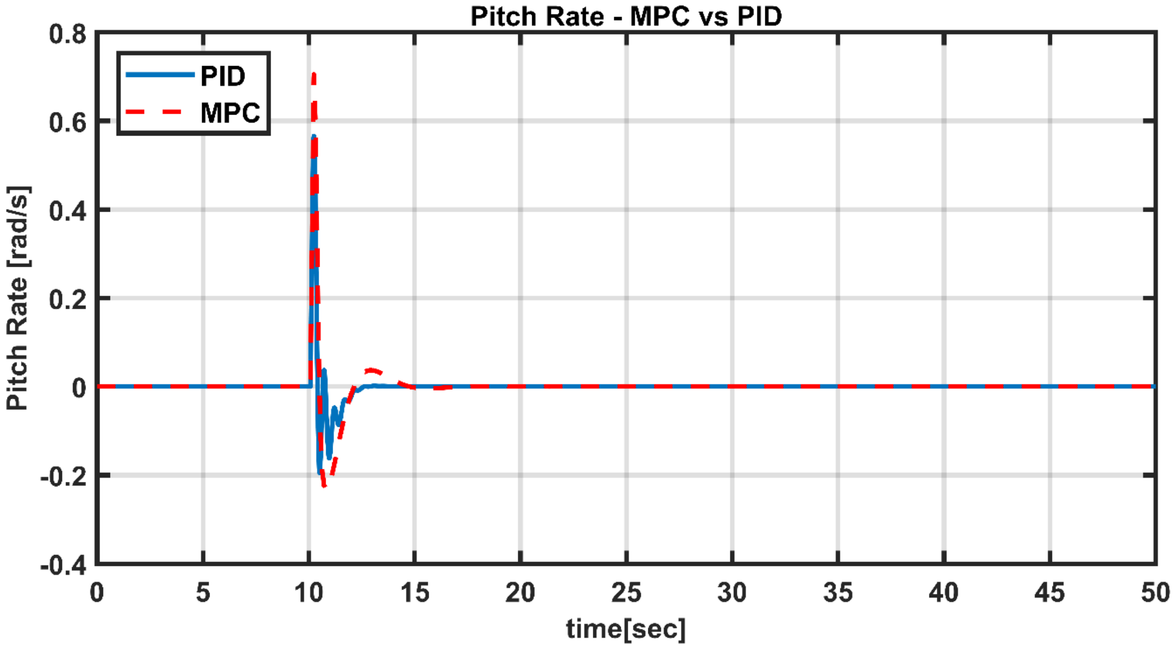


Figure 5-29 Input disturbance rejection - Pitch rate response

5.1.4. Results of Speed Mode Under Input Disturbance

In this section, the results of simulations against the step input have been shown.

5.1.4.1. Directional Channel

Test Input:

To perform this test, 0.2 directional command (δ_{rud}) has been applied as input to dynamic model.

Responses:

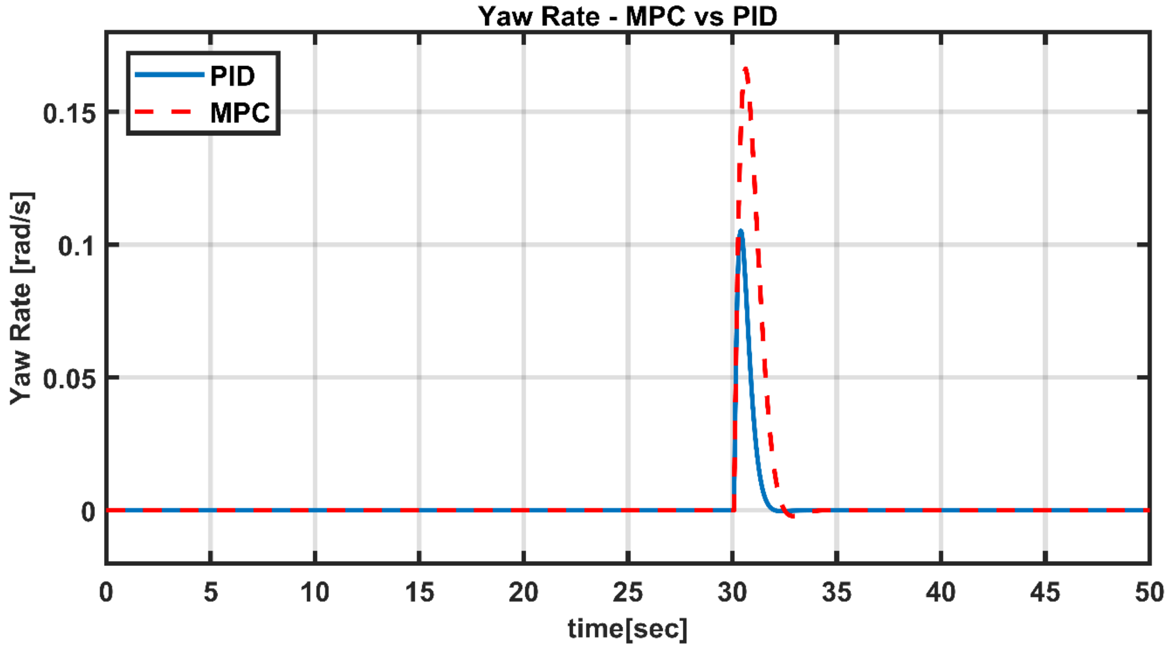


Figure 5-30 Input disturbance rejection - Yaw rate responses

5.1.4.2. Lateral Channel

Test Input:

To perform this test, 0.2 lateral command (δ_{ail}) has been applied as input to dynamic model.

Responses:

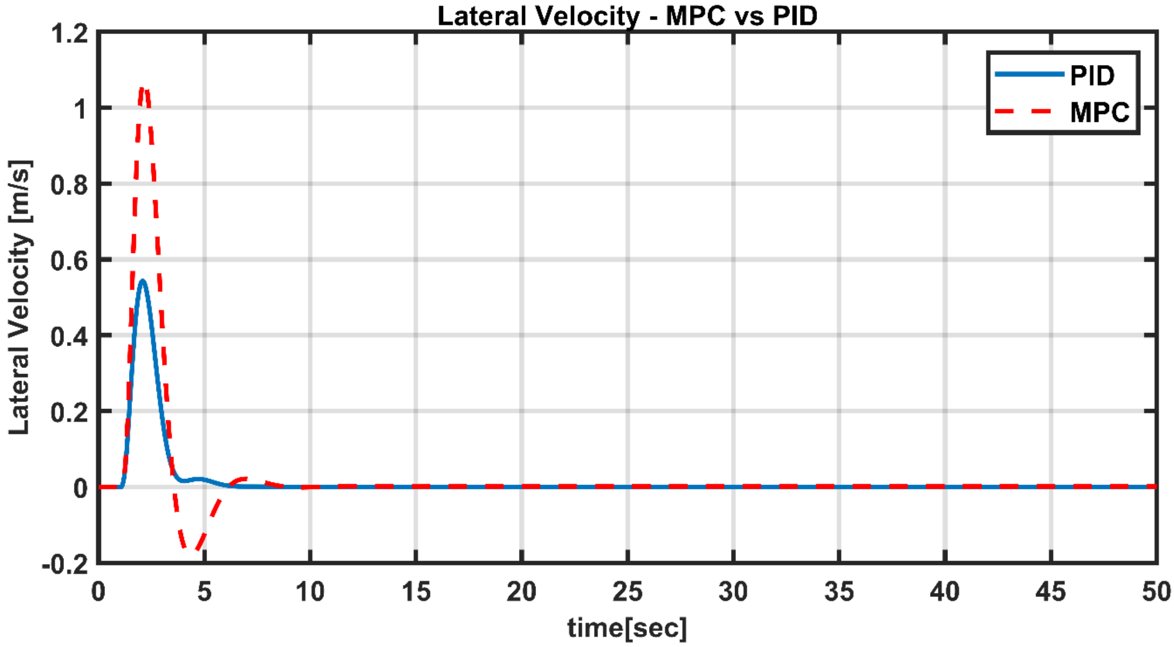


Figure 5-31 Input disturbance rejection – Lateral velocity response

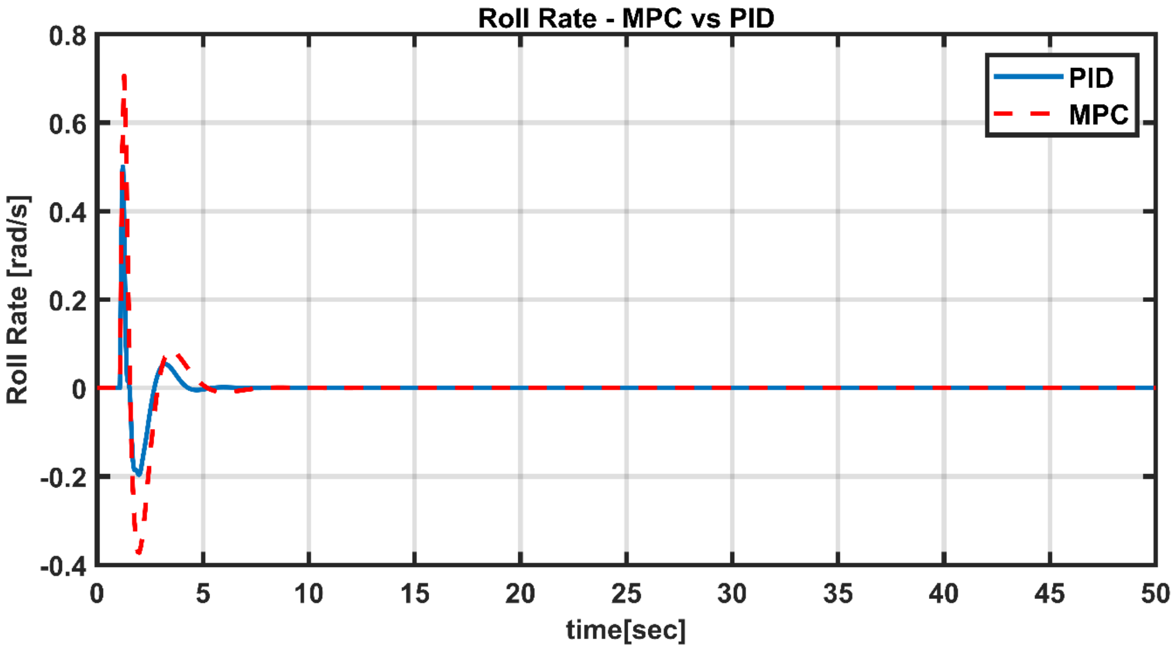


Figure 5-32 Input disturbance rejection - Roll rate response

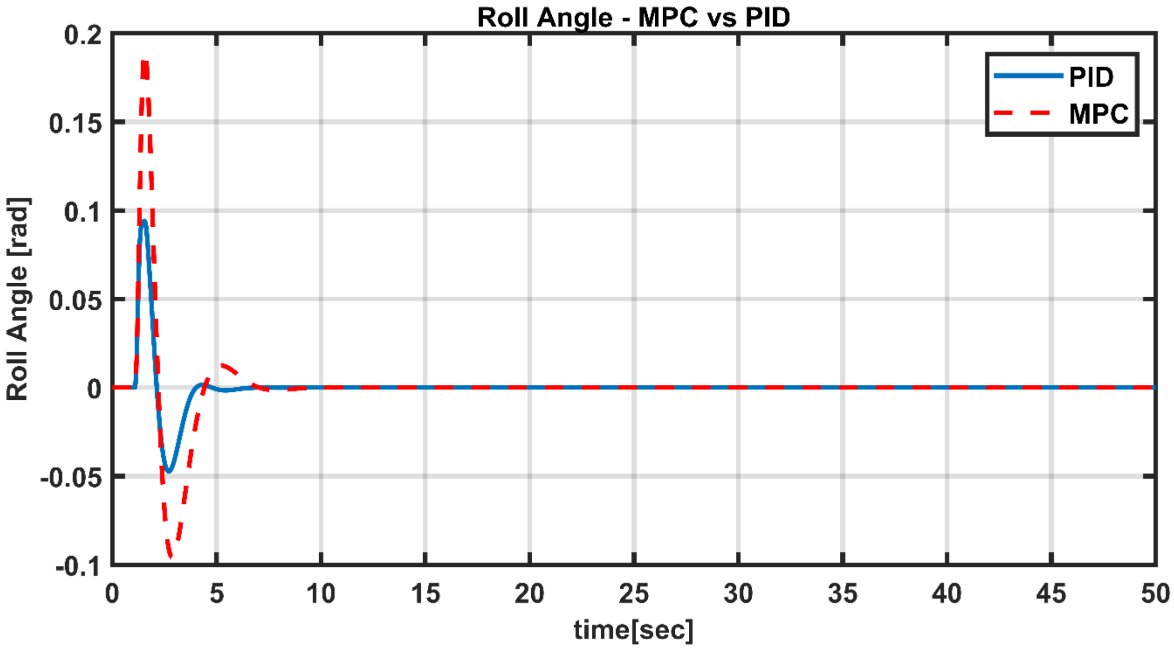


Figure 5-33 Input disturbance rejection – Roll angle response

5.1.4.3. Longitudinal Channel

Test Input:

To perform this test, 0.2 longitudinal command (δ_{ele}) has been applied as input to dynamic model.

Responses:

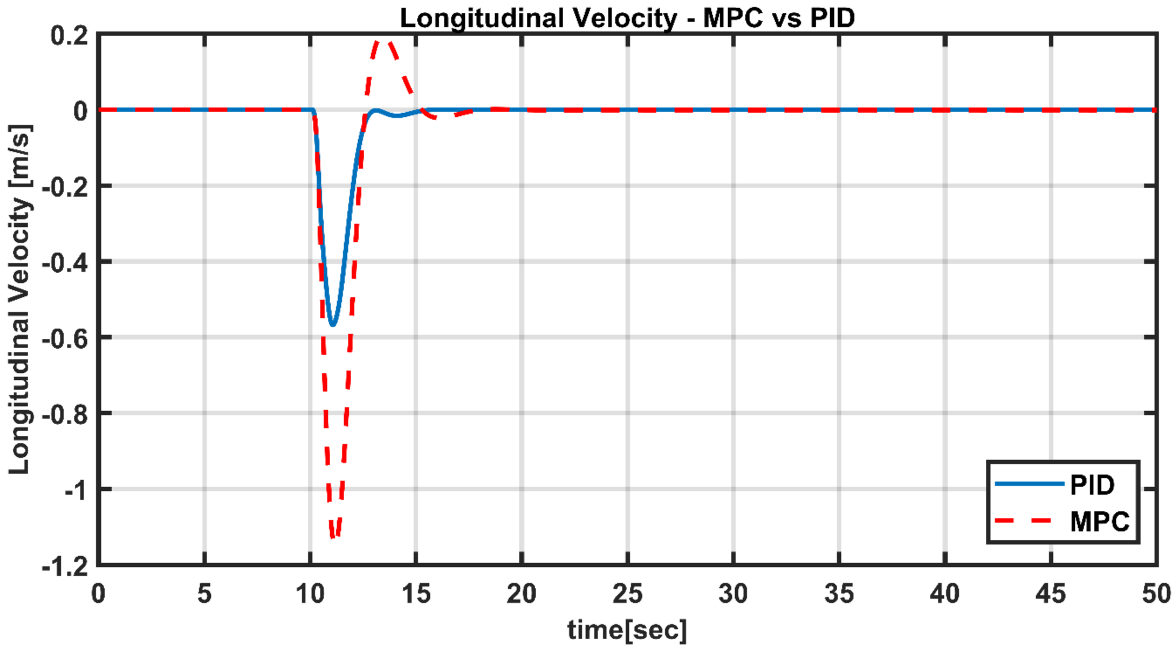


Figure 5-34 Input disturbance rejection – Longitudinal velocity response

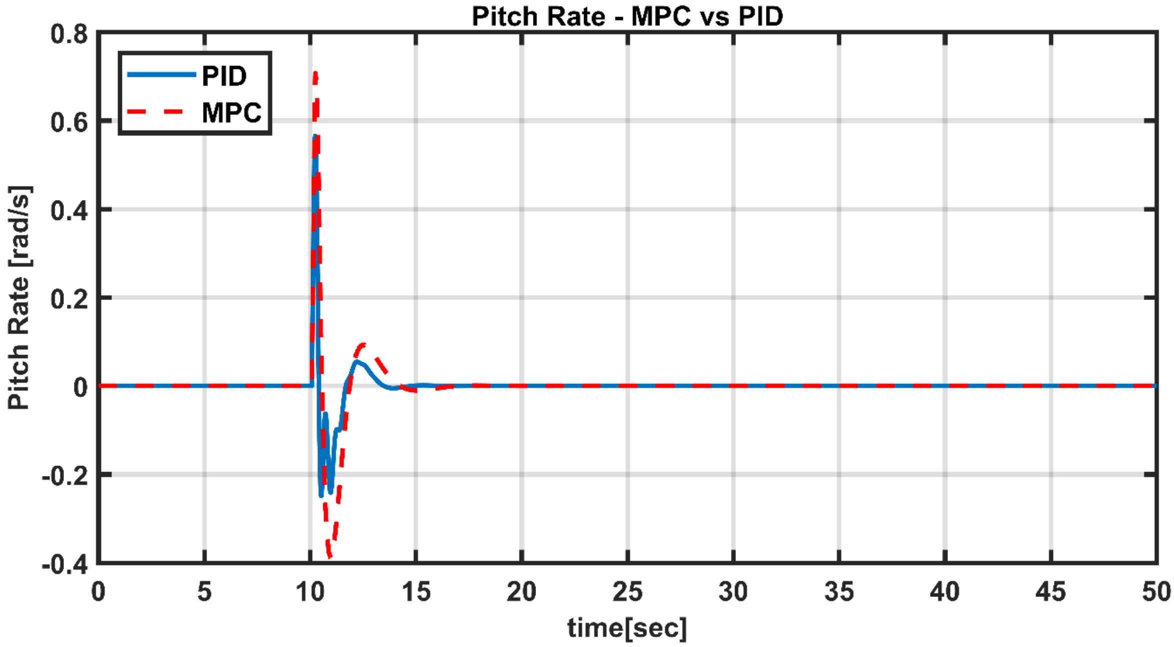


Figure 5-35 Input disturbance rejection - Pitch rate response

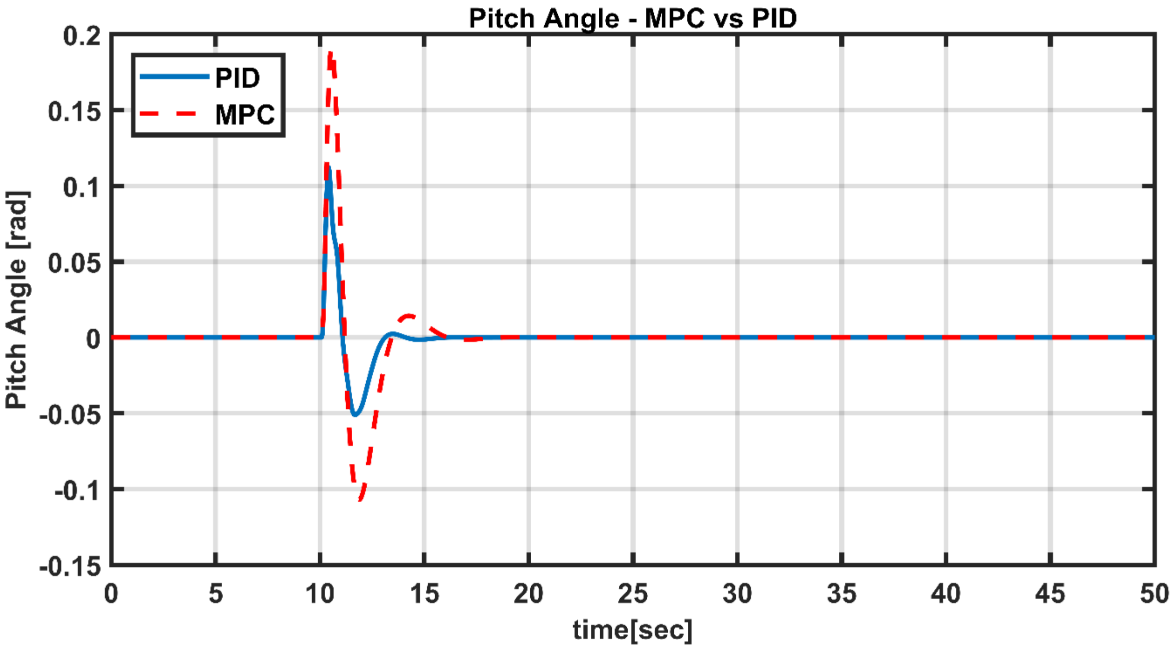


Figure 5-36 Input disturbance rejection – Pitch angle response

5.1.4.1. Vertical Channel

Test Input:

To perform this test, 40 PWM vertical command (δ_{thr}) has been applied as input to dynamic model.

Responses:

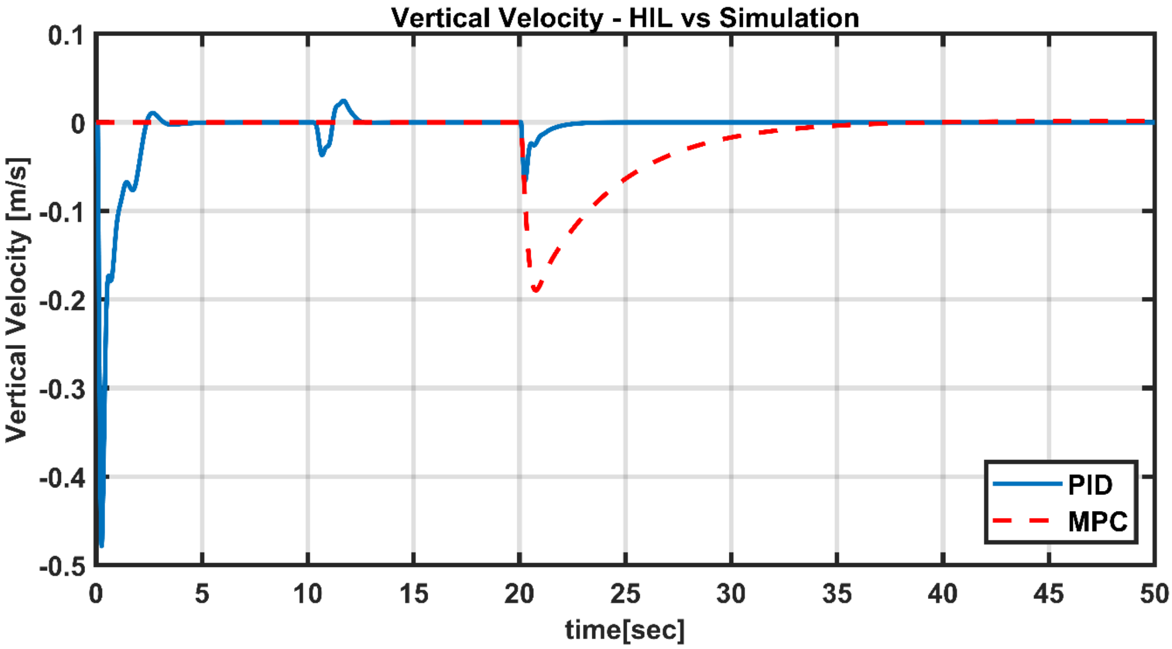


Figure 5-37 Input disturbance rejection – Vertical velocity response

5.2. Software in the Loop Tests

C / C ++ codes of algorithms, which are designed in MATLAB, can be generated via embedded coder software. The simulation environment created to test the accuracy of these codes is called software in the loop environment. In the software in the loop test, the generated C / C ++ codes are included in the simulation with s-functions. These s-functions take over the wrapping function between MATLAB and C codes. After the S-function is prepared, the test is performed by calling C / C ++ codes. The generated code and the model simulation responses should be the same or very close. Another purpose of performing SIL testing is to see possible software problems before hardware implementation.

In order to perform this test, first, the c code is generated in accordance with the controller block for ARM processors with the help of the embedded coder. This generated C code has been added to the project folder. And, they have been called in attitude and speed modes from the autopilot software.

The autopilot software operates at 100 Hz and performs tasks such as reading sensors, establishing communication with the ground control station, making navigation solutions and providing mode transitions. In addition, the conversions of PWM inputs into angle and speed commands have also been done on the autopilot software. It also runs the MPC code at every 12.5 Hz and takes control inputs and converts them to PWM. This test can run in simulation time without the need for real-time operation.

The reason for the error seen during the SIL test is due to the losses during the conversion of commands given over "%" to PWM and sending as “uint16”. This difference is observed as the autopilot code works in accordance with driving with RC control as input. In the simulation environment, since the commands are given in rad and m/s units, $10^{-3}/10^{-4}$ unit errors occurred in the test at points where the command is different from zero.

5.2.1. Results of Attitude Mode

In this section, the results of the software in the loop tests against the doublet manoeuvre have been shown. As the pilot in attitude mode controls the vertical channel, no input is given to the vertical channel during this test. In order to keep the system stable, this channel input was fed as zero in the dynamic model during the test.

5.2.1.1. Directional Channel

In the directional channel, if the input is non-zero, the controller operates in the angular velocity loop, and if the command is zero, the controller operates in the angle loop.

Test Input:

To perform this test, 0.3 rad/s yaw rate command has been applied.

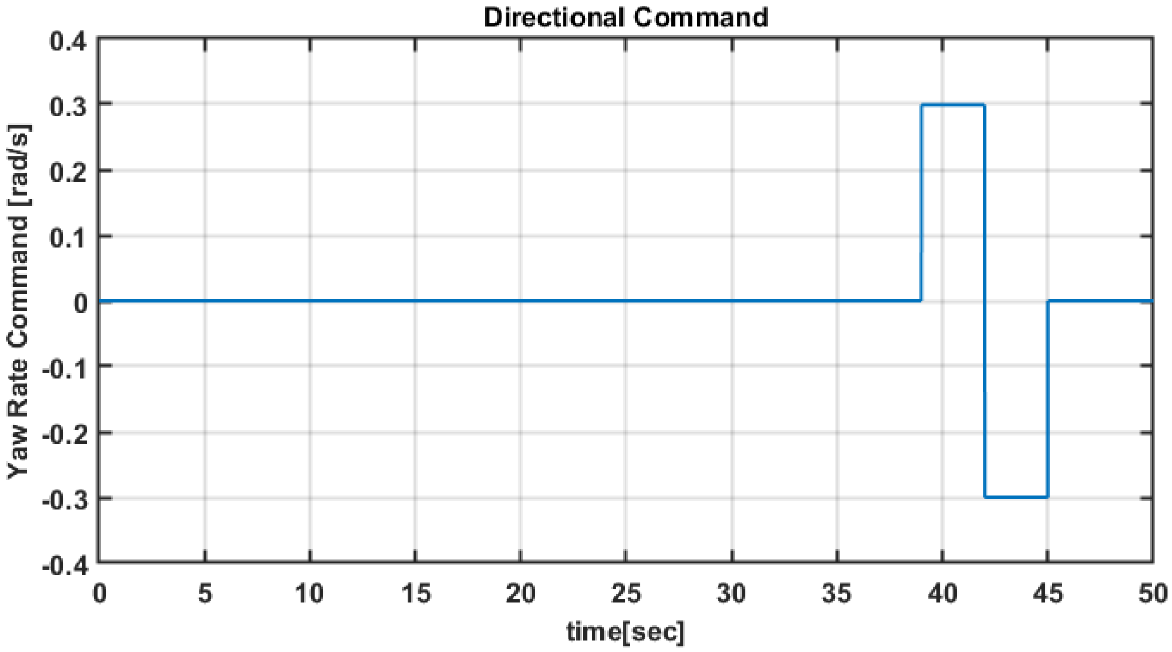


Figure 5-38 SIL test - Directional command

Responses:

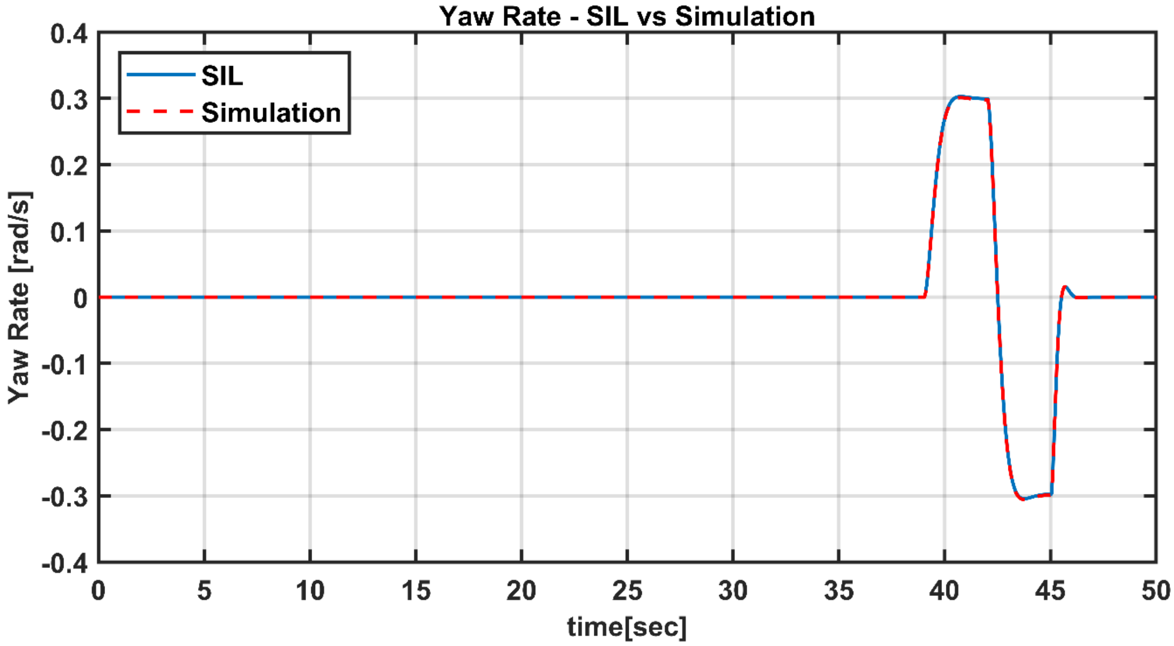


Figure 5-39 SIL test - Yaw rate responses

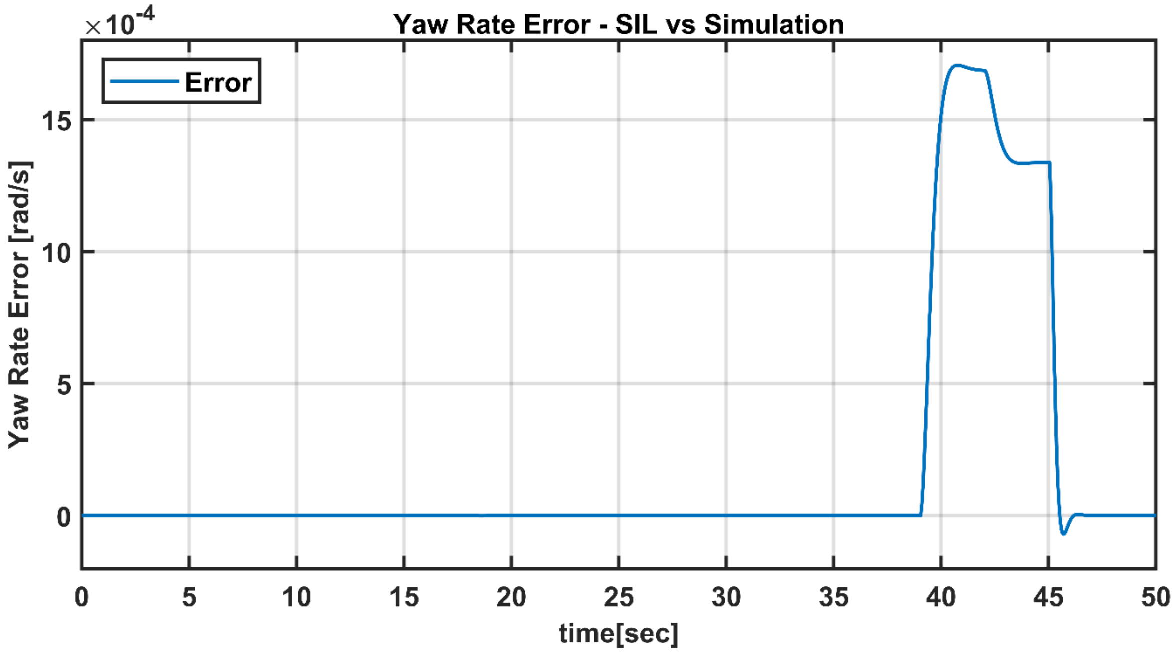


Figure 5-40 SIL test - Yaw rate error

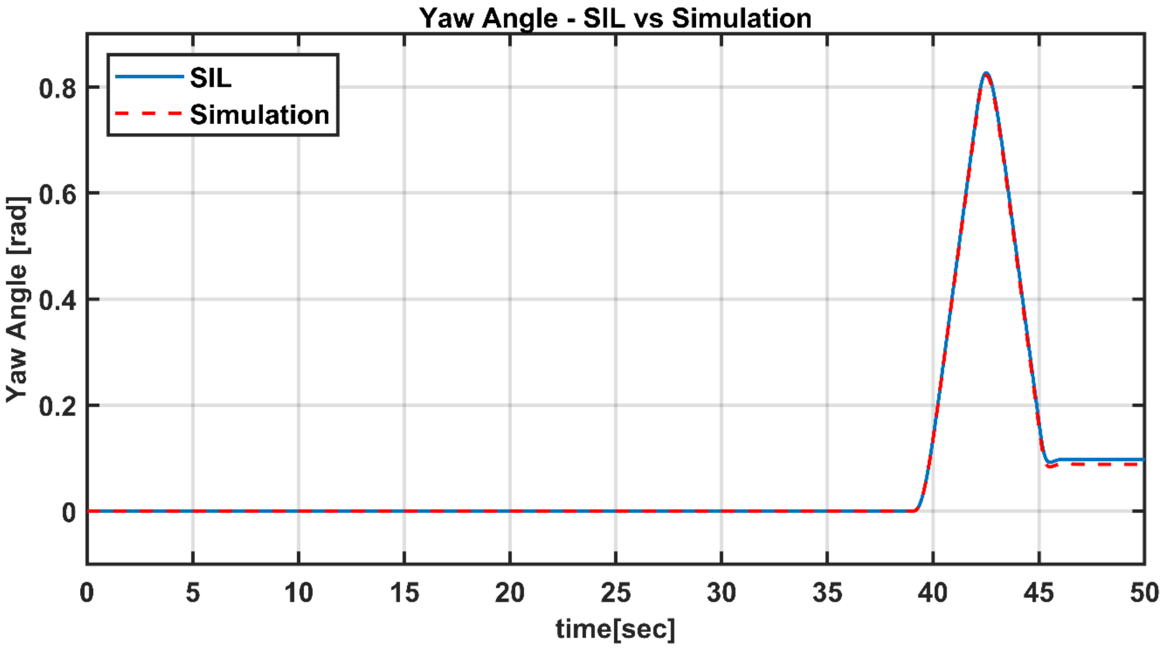


Figure 5-41 SIL test - Yaw angle responses

As seen in Figure 5-39,Figure 5-40,Figure 5-41, the SIL tests of the directional channel were performed with very low errors. As mentioned before, except for the error caused by the PWM conversion, the test was performed successfully.

5.2.1.2. Lateral Channel

In the lateral channel, the system input is given as angle command. Test input is shown in Figure 5-42.

Test Input:

To perform this test, 0.3 rad roll angle command has been applied.

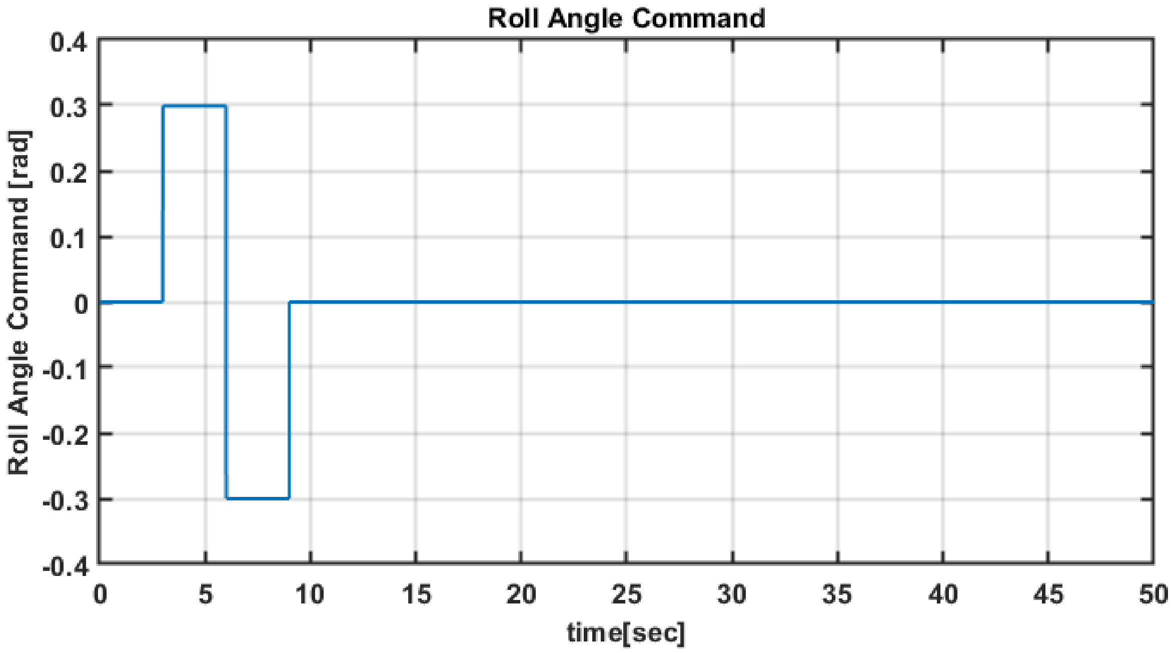


Figure 5-42 SIL test - Roll command

Responses:

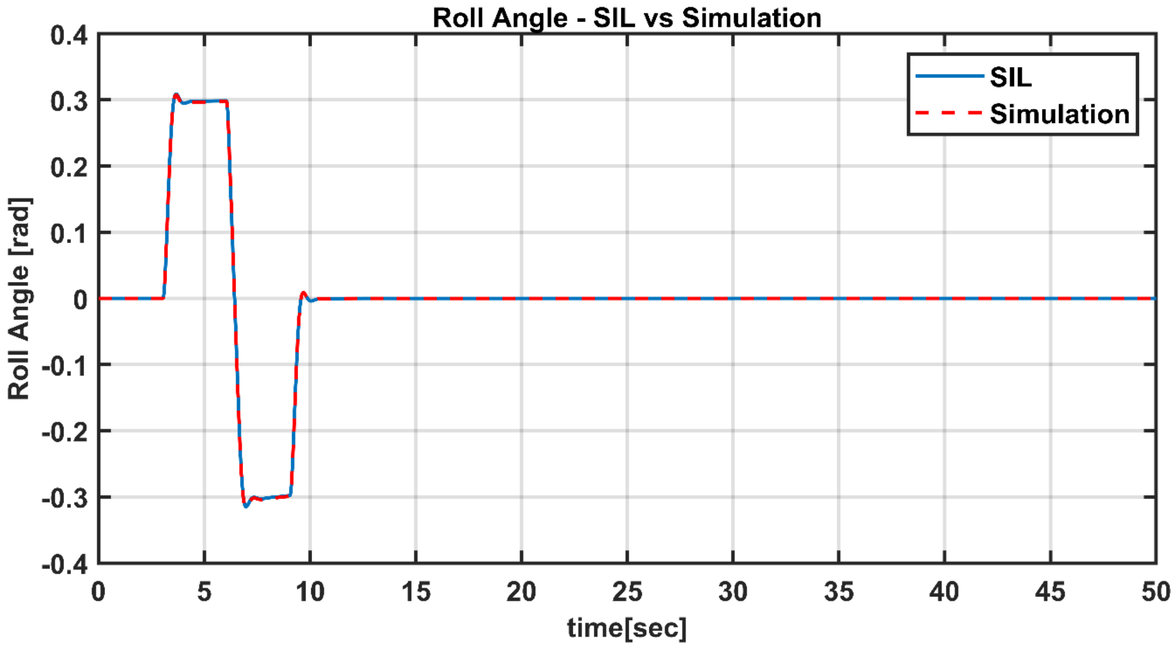


Figure 5-43 SIL test - Roll angle response

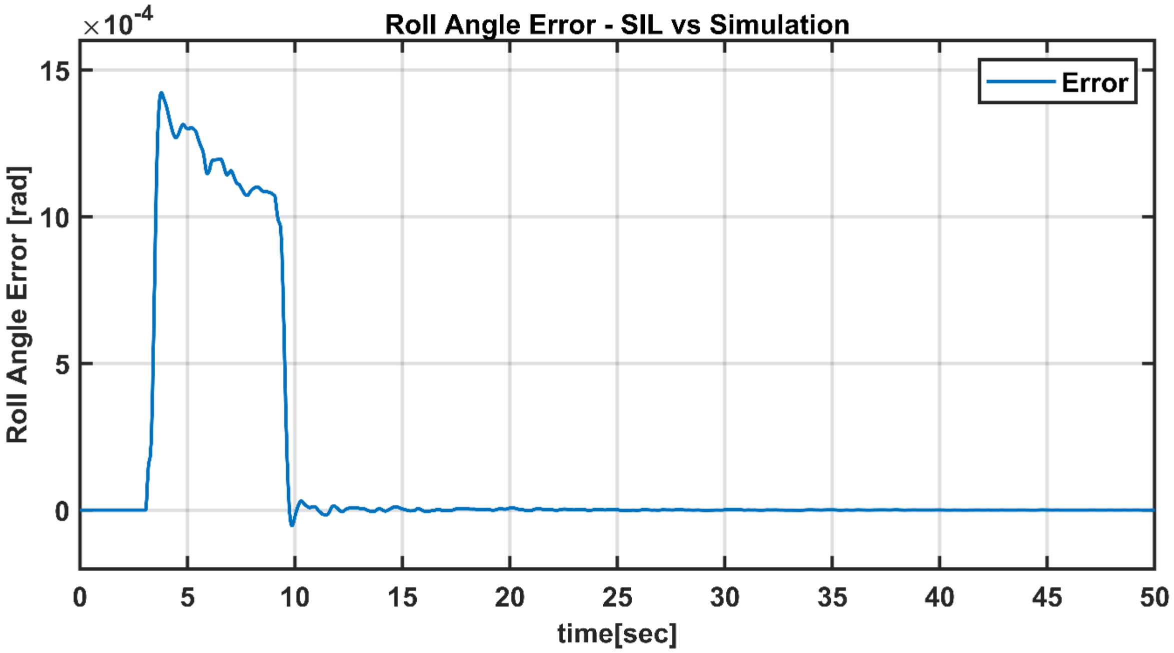


Figure 5-44 SIL test - Roll angle error

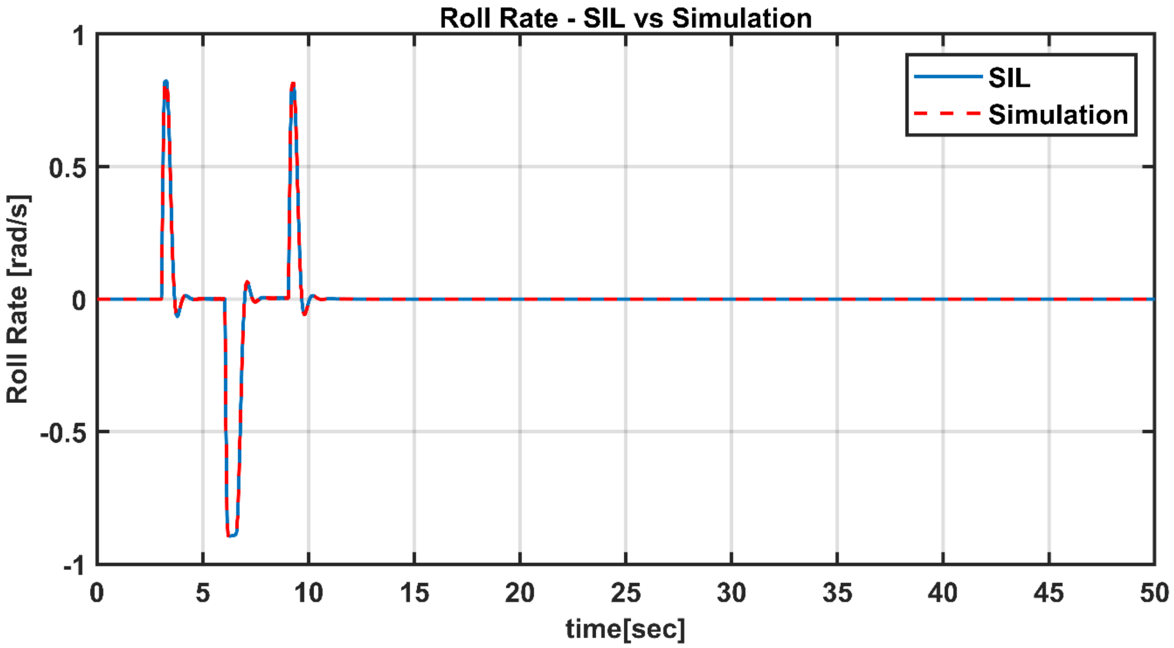


Figure 5-45 SIL test - Roll rate response

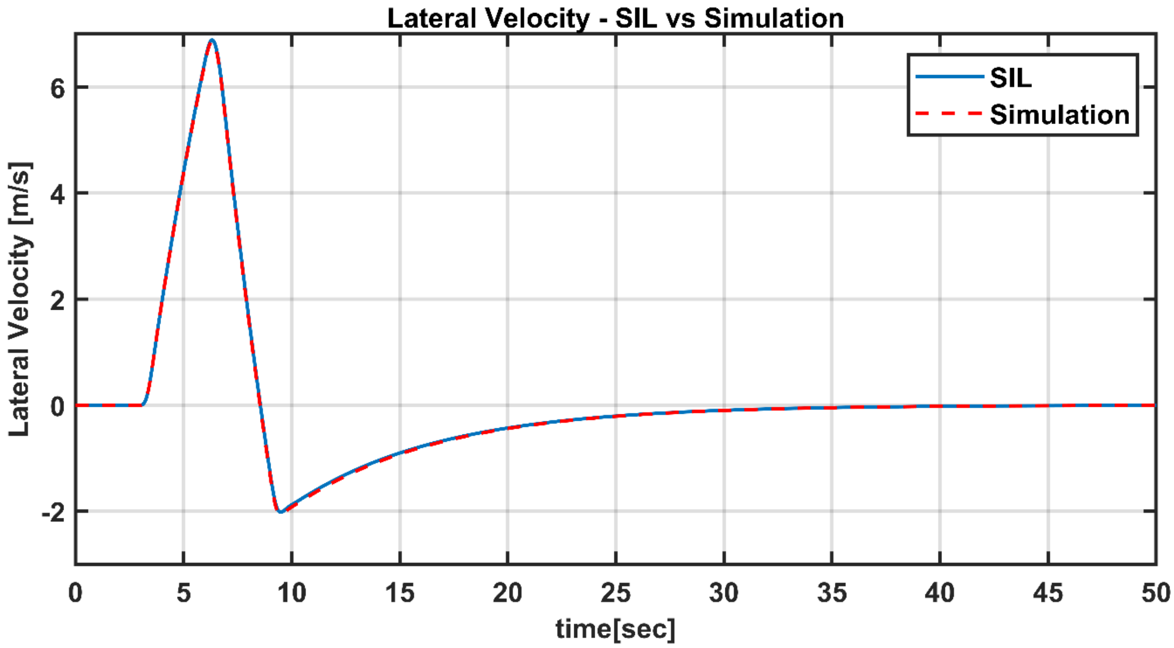


Figure 5-46 SIL test - Lateral velocity response

5.2.1.3. Longitudinal Channel

In the longitudinal channel, the system input is given as angle command. Test input is shown in Figure 5-47.

Test Input:

To perform this test, 0.3 rad pitch angle command has been applied.

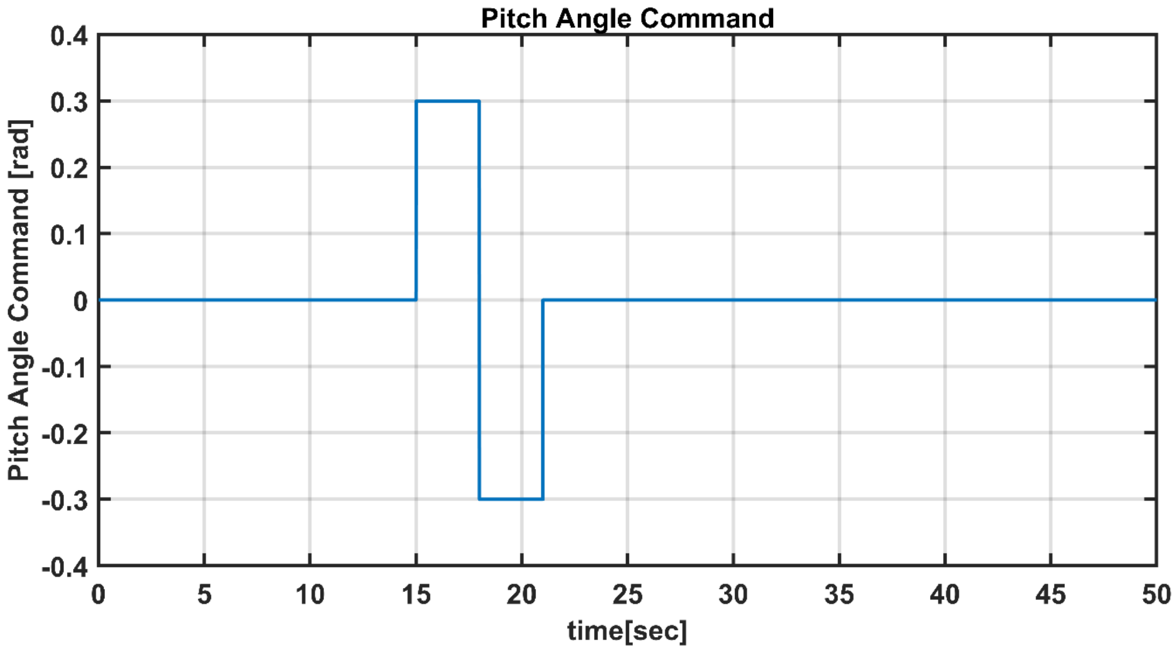


Figure 5-47 SIL test - Pitch command

Responses:

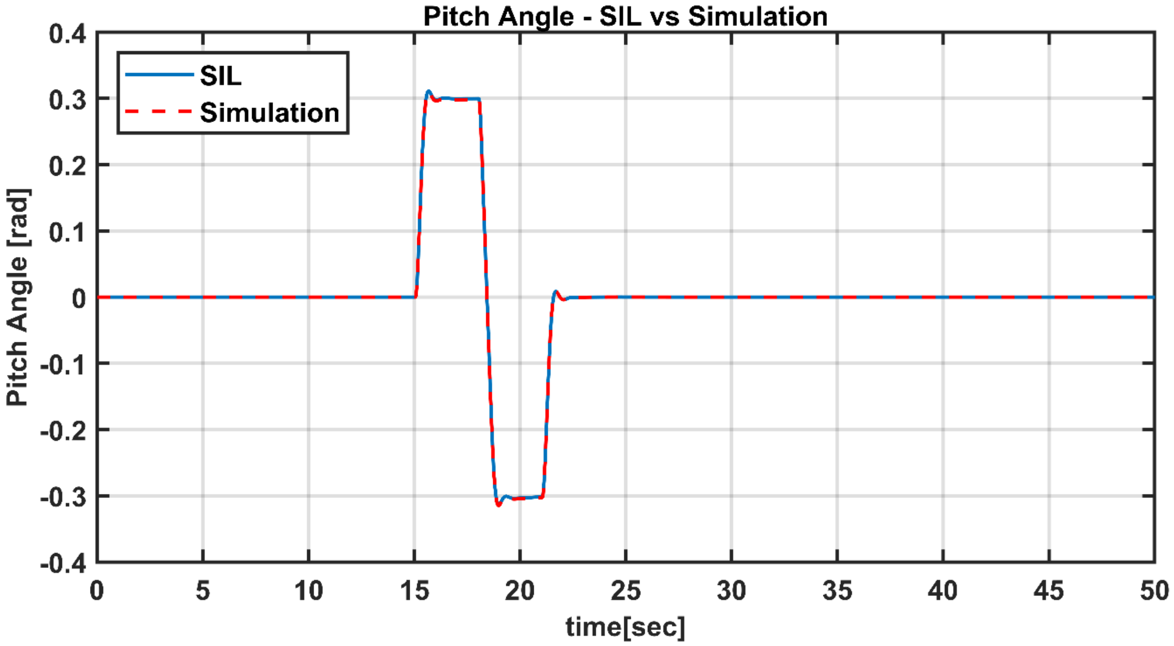


Figure 5-48 SIL test - Pitch angle response

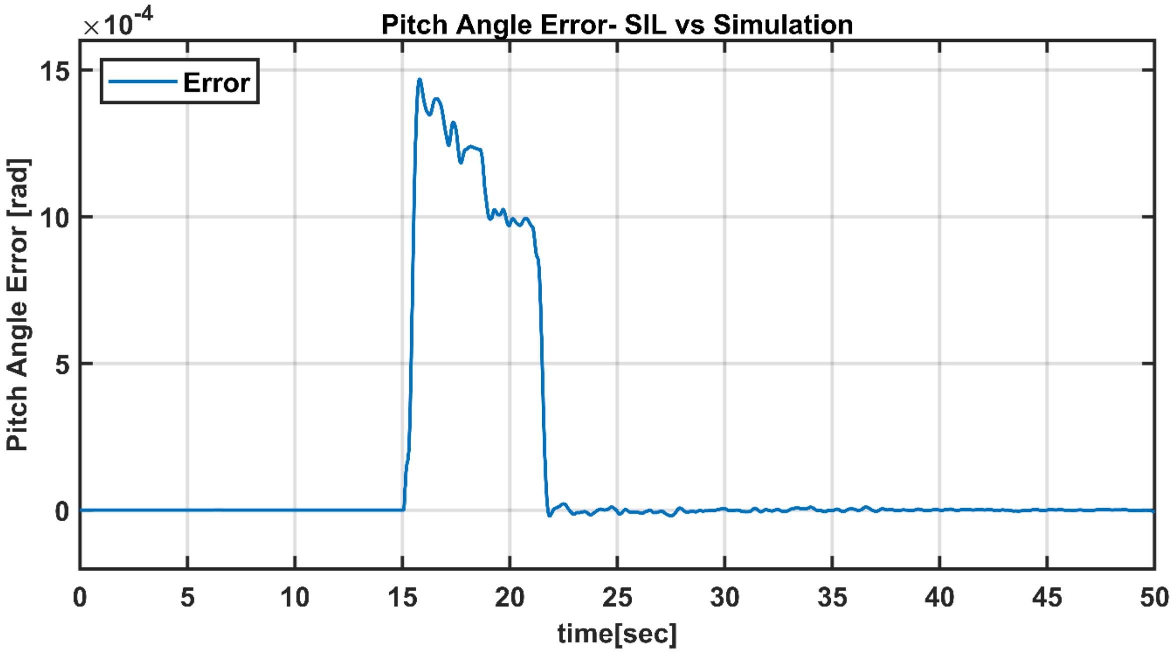


Figure 5-49 SIL test - Pitch angle error

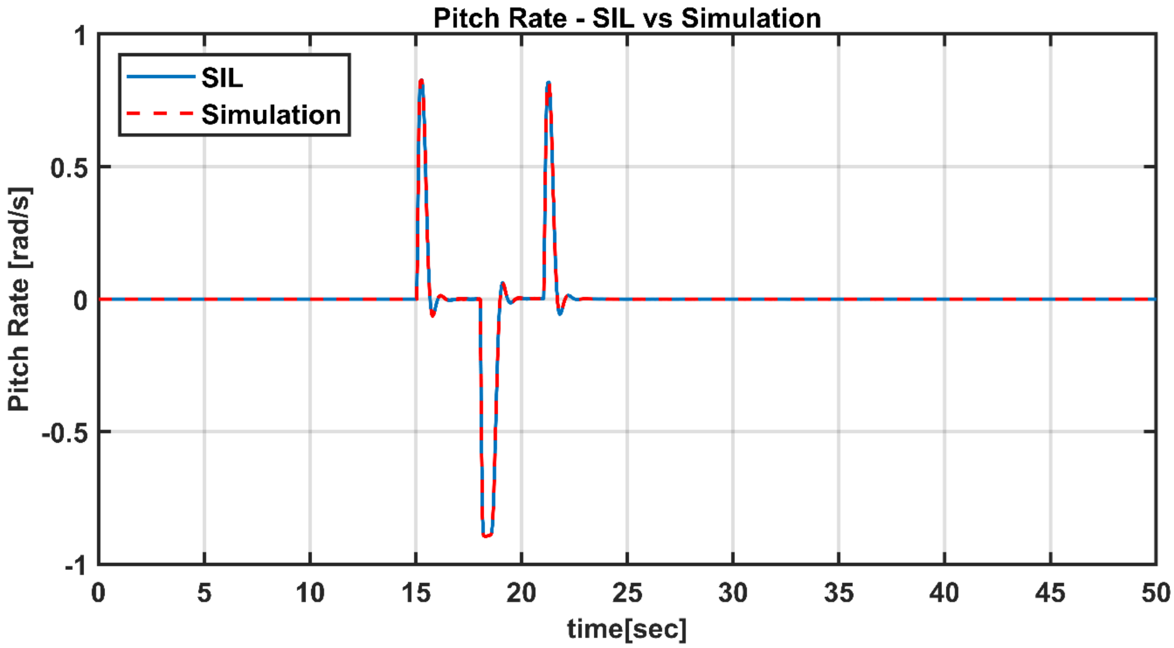


Figure 5-50 SIL test - Pitch rate response

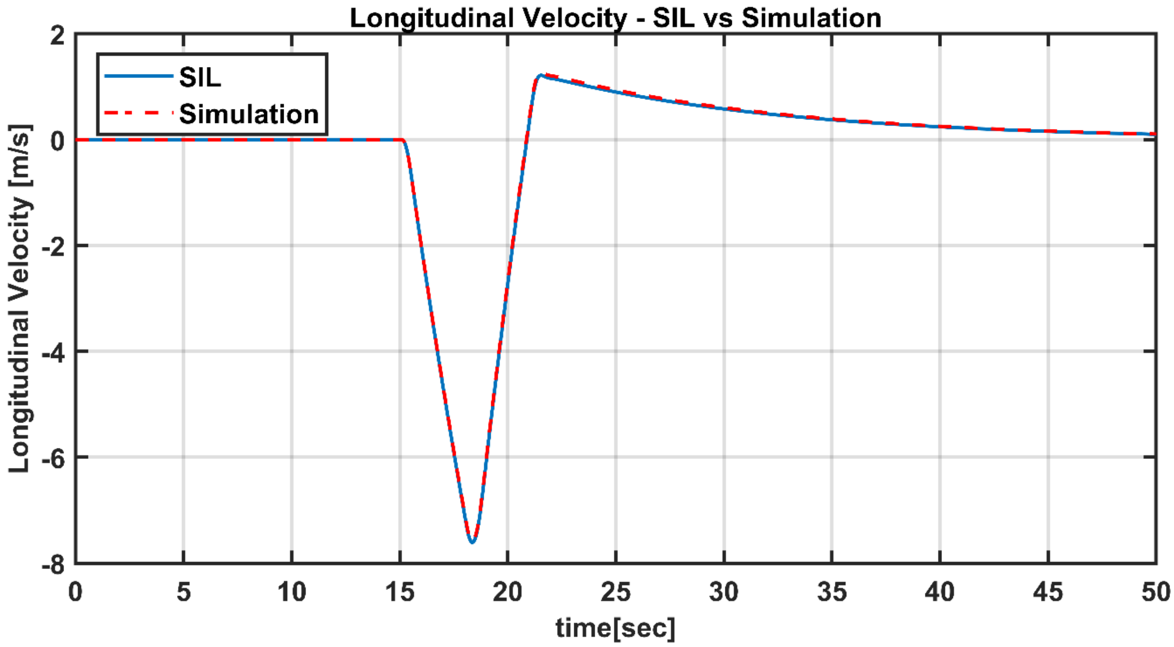


Figure 5-51 SIL test - Longitudinal velocity response

5.2.2. Results of Speed Mode

In this section, the results of the software in the loop tests against the doublet manoeuvre have been shown.

5.2.2.1. Directional Channel

In the directional channel, if the input is non-zero, the controller operates in the angular velocity loop, and if the command is zero, the controller operates in the angle loop.

Test Input:

To perform this test, 0.3 rad/s yaw rate command has been applied.

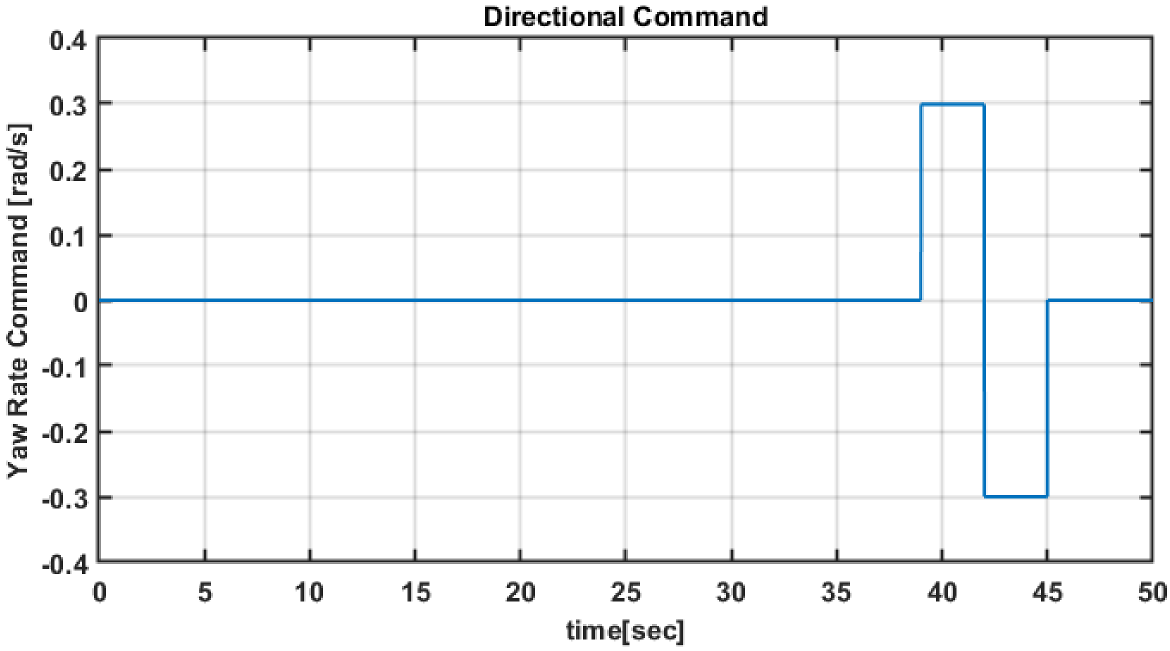


Figure 5-52 SIL test - Directional command

Responses:

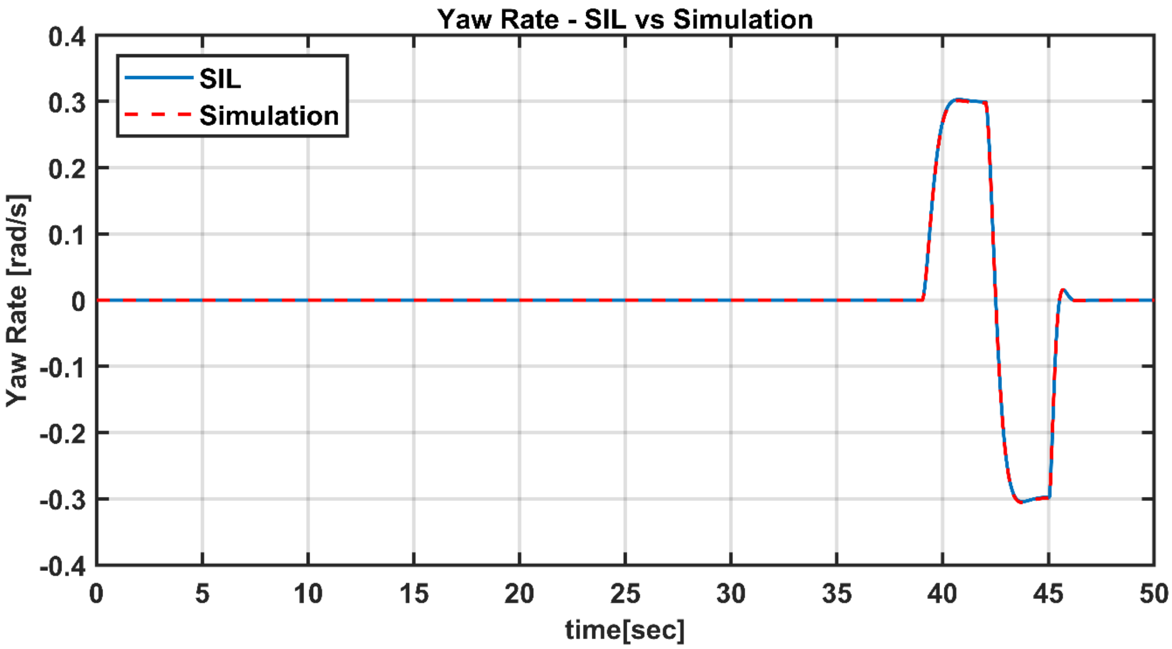


Figure 5-53 SIL test - Yaw rate responses

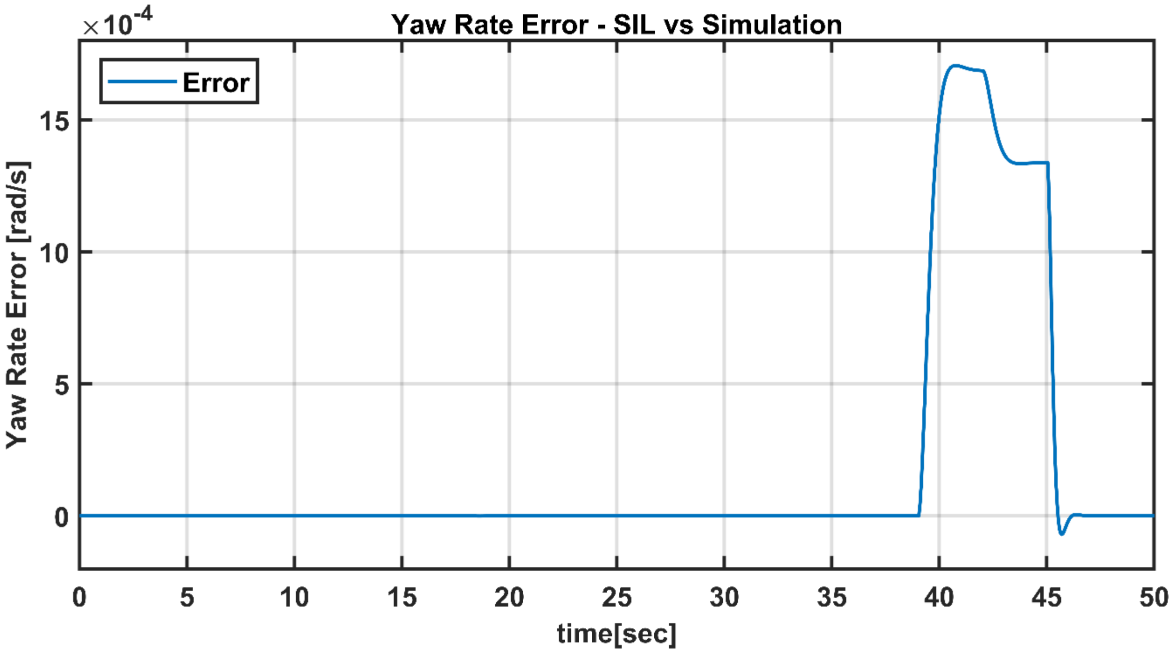


Figure 5-54 SIL test - Yaw rate error

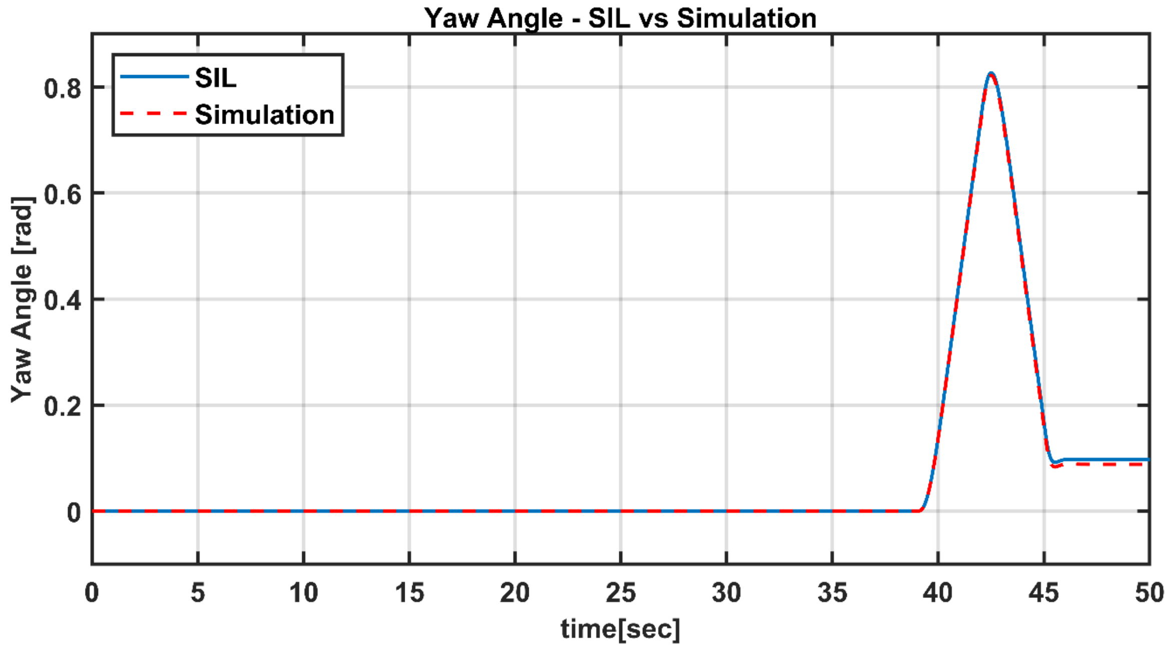


Figure 5-55 SIL test - Yaw angle responses

As seen in Figure 5-53, Figure 5-54, Figure 5-55, the SIL tests of the directional channel were performed with very low errors. As mentioned before, except for the error caused by the PWM conversion, the test was performed successfully.

5.2.2.2. Lateral Channel

In the lateral channel, the system input is given as angle command. Test input is shown in Figure 5-56.

Test Input:

To perform this test, 1 m/s lateral velocity command has been applied.

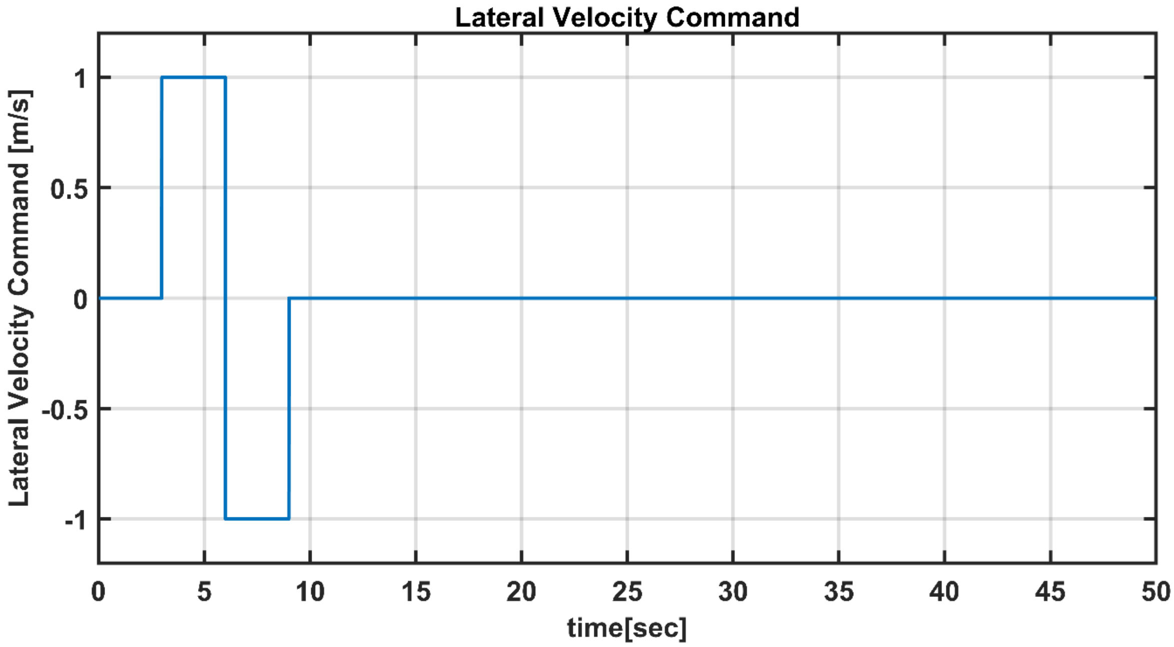


Figure 5-56 SIL test – Lateral velocity command

Responses:

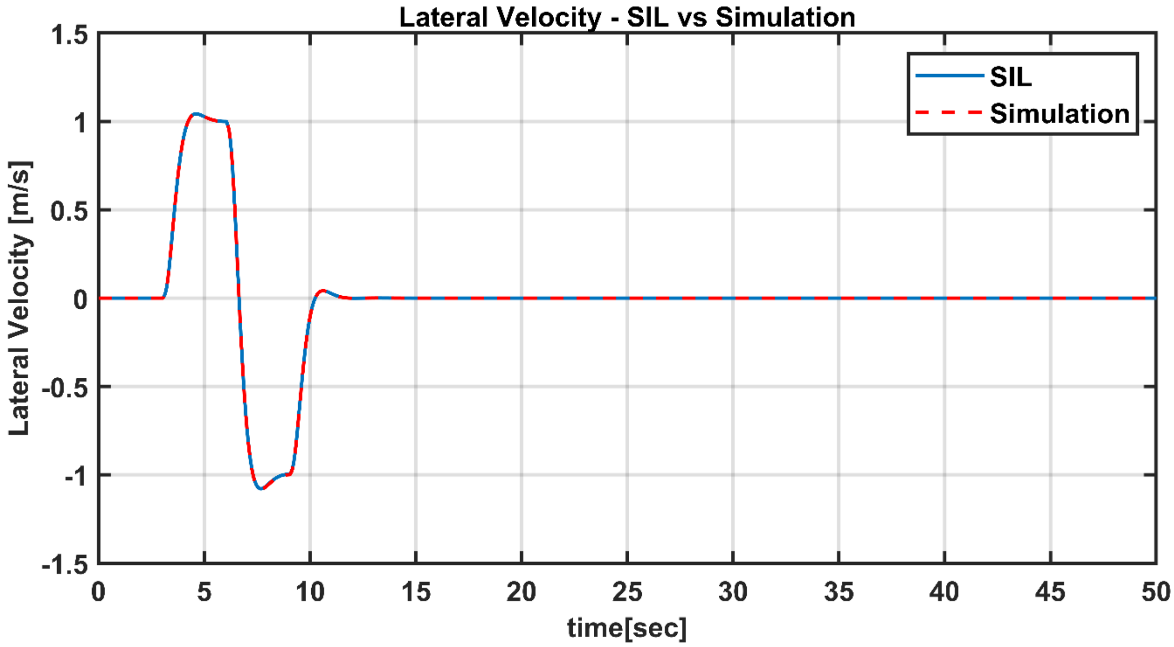


Figure 5-57 SIL test – Lateral velocity response

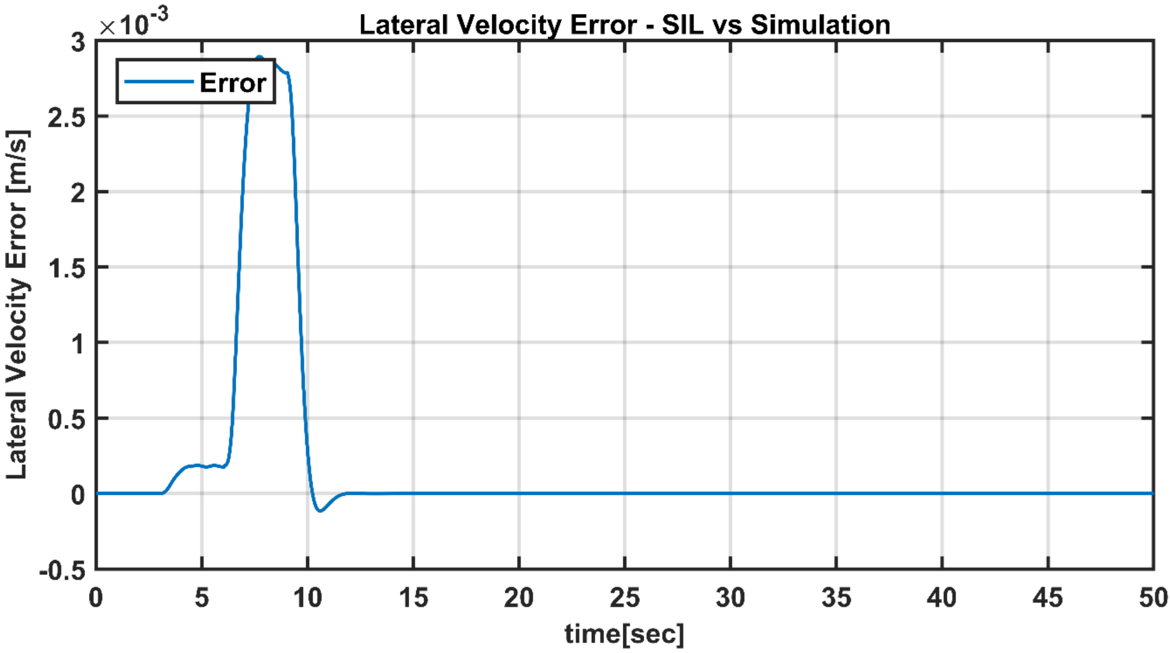


Figure 5-58 SIL test – Lateral velocity error

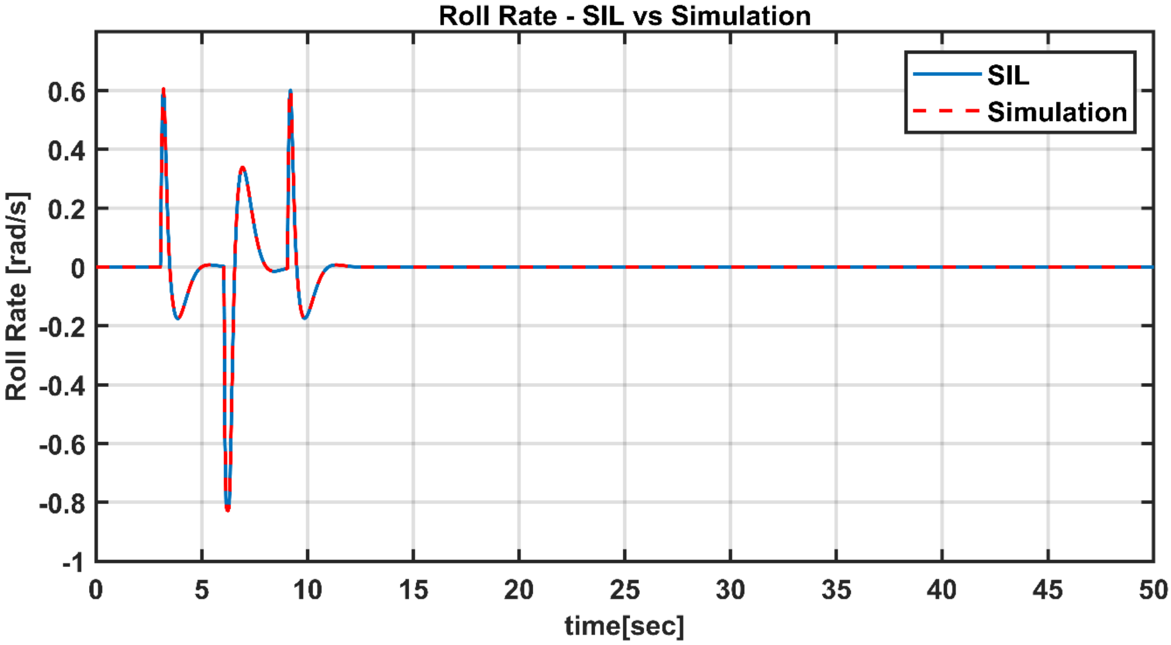


Figure 5-59 SIL test - Roll rate response

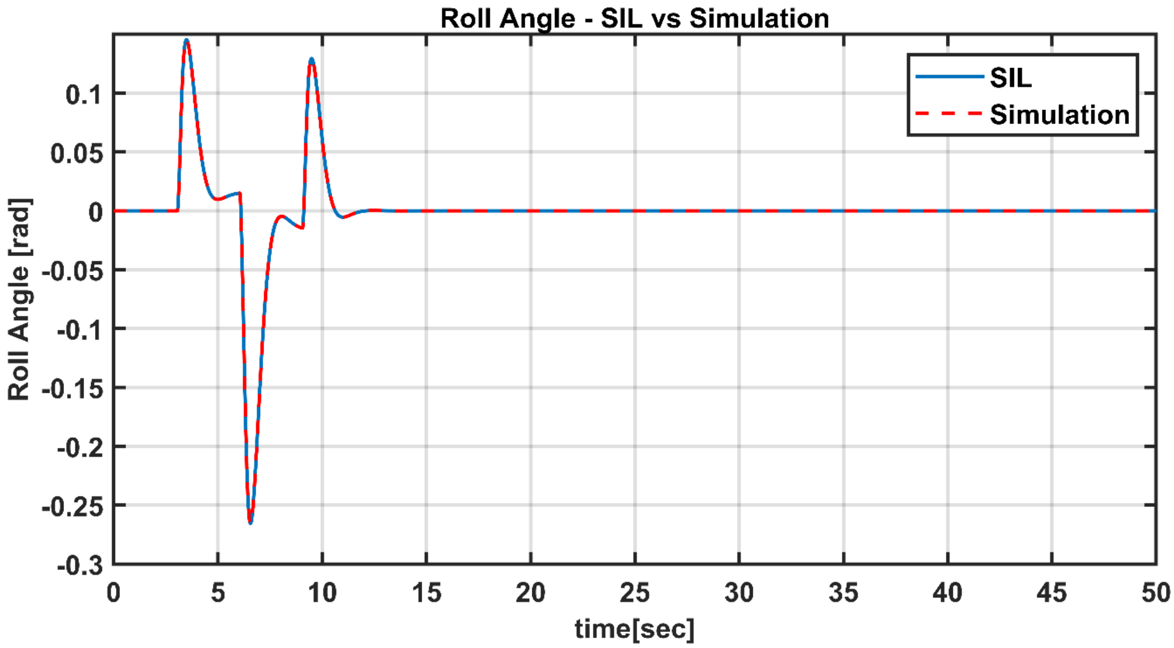


Figure 5-60 SIL test – Roll angle response

5.2.2.3. Longitudinal Channel

In the longitudinal channel, the system input is given as angle command. Test input is shown in Figure 5-61.

Test Input:

To perform this test, 1 m/s longitudinal velocity command has been applied.

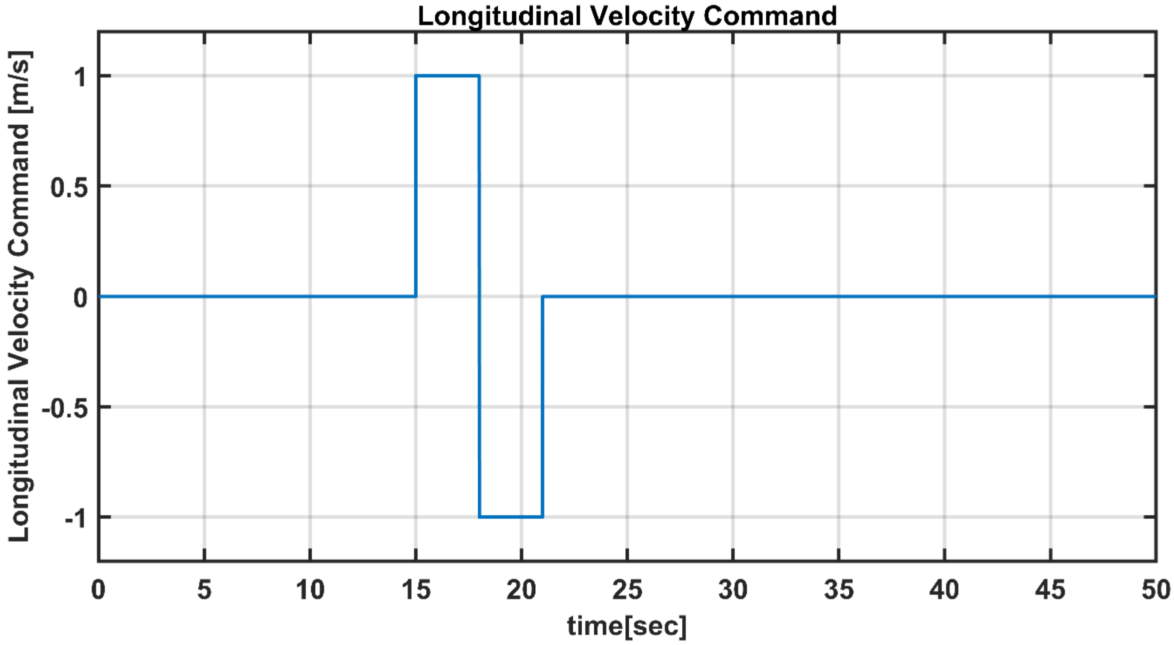


Figure 5-61 SIL test – Longitudinal velocity command

Responses:

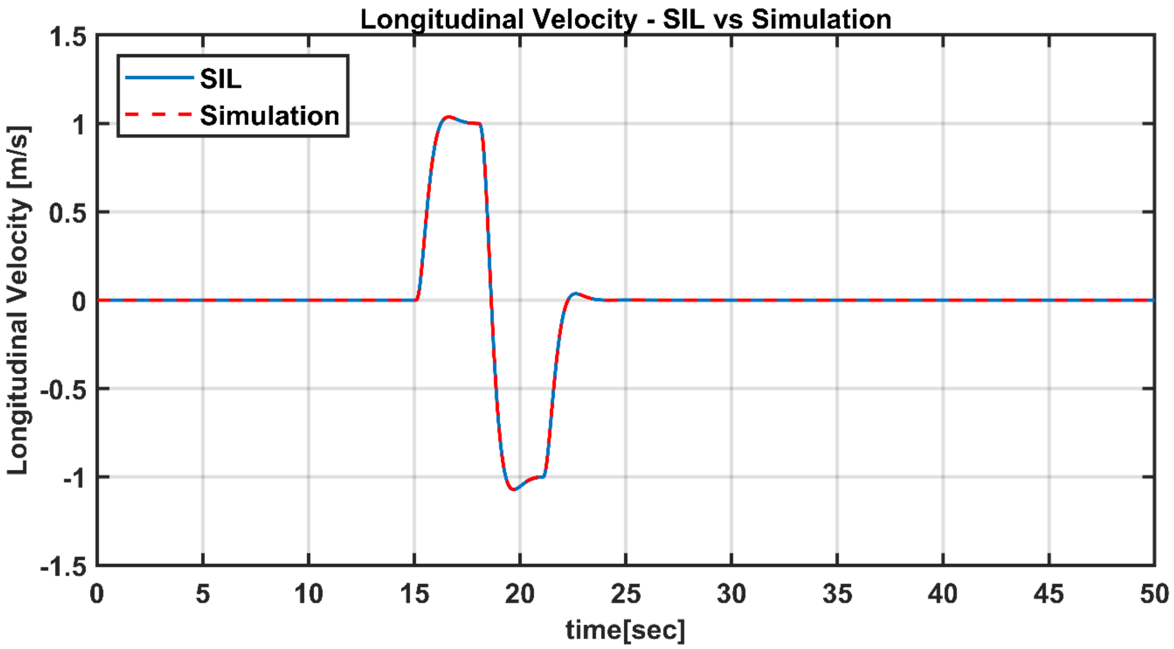


Figure 5-62 SIL test – Longitudinal velocity response

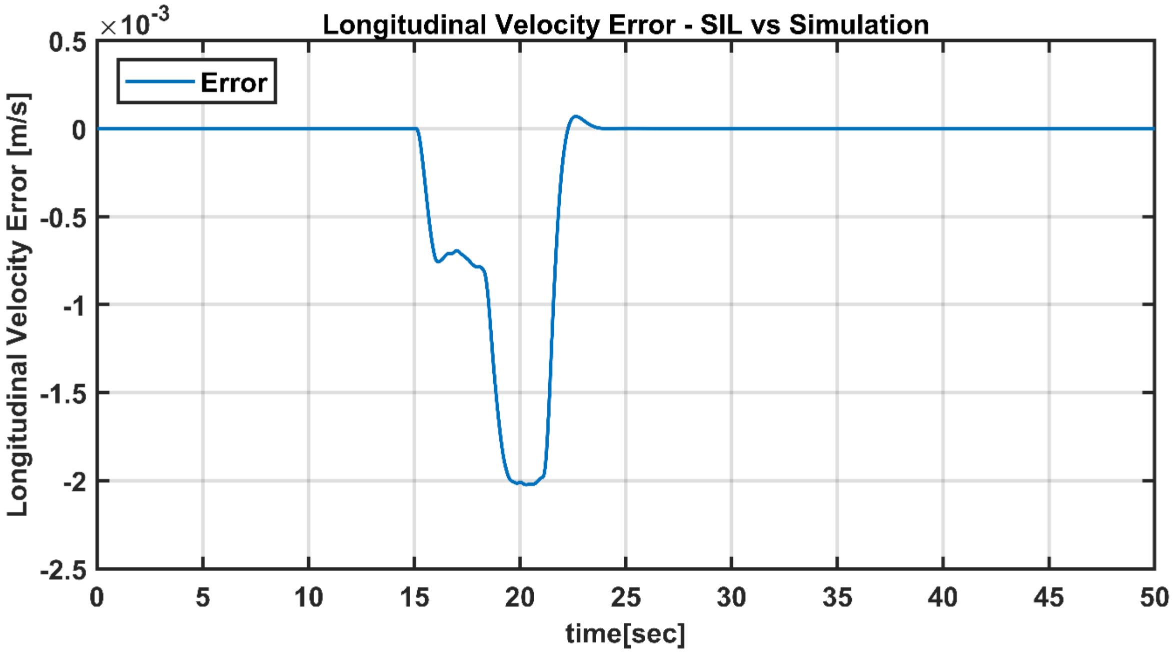


Figure 5-63 SIL test – Longitudinal velocity error

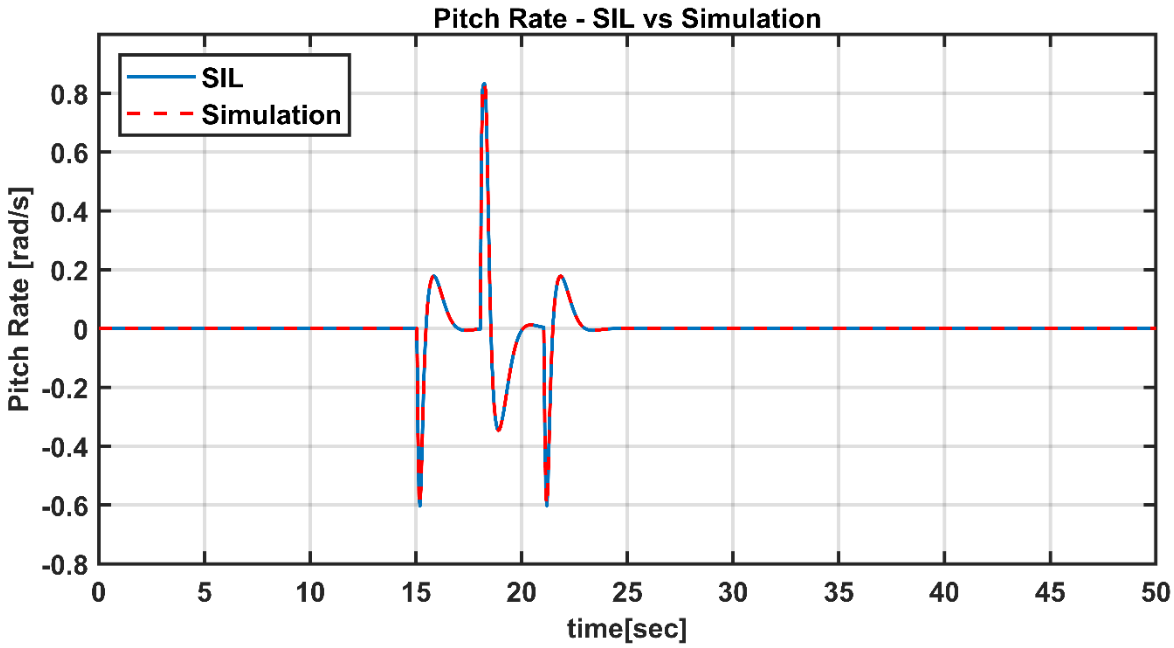


Figure 5-64 SIL test - Pitch rate response

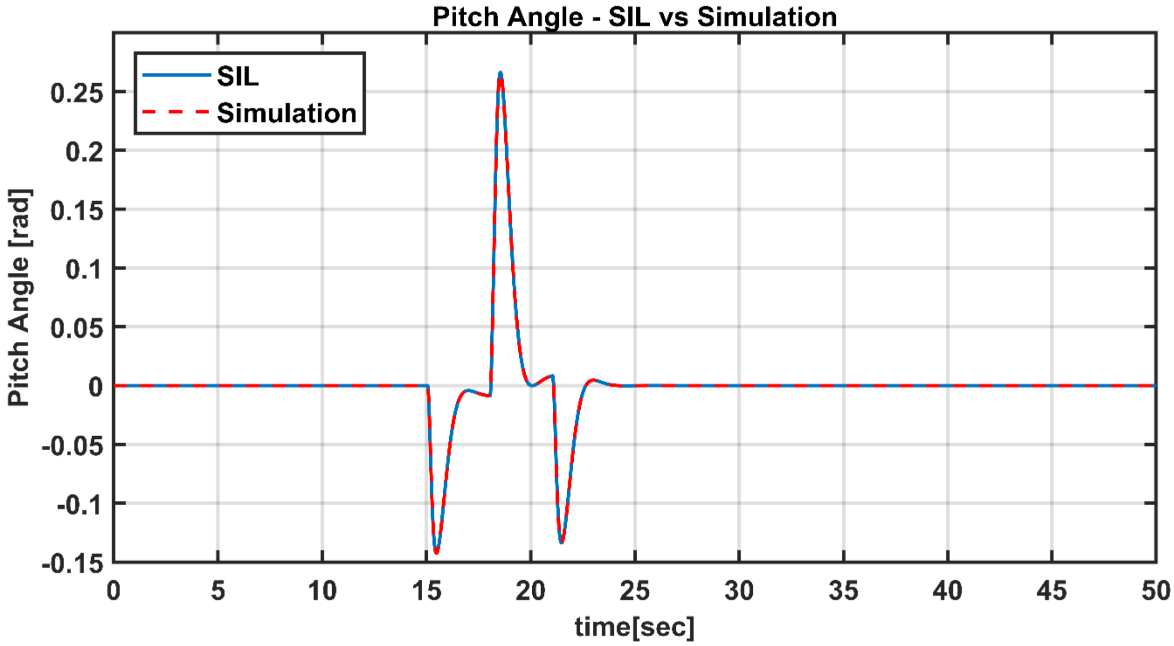


Figure 5-65 SIL test – Pitch angle response

5.2.2.4. Vertical Channel

In the longitudinal channel, the system input is given as angle command. Test input is shown in Figure 5-66.

Test Input:

To perform this test, 1 m/s vertical velocity command has been applied.

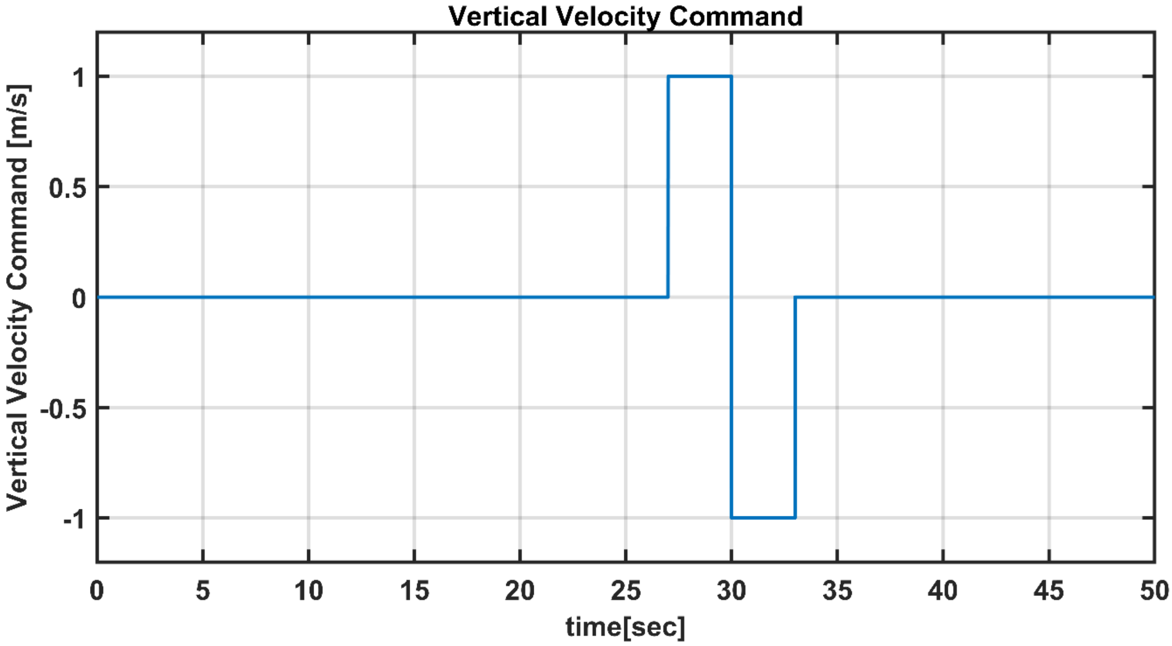


Figure 5-66 SIL test – Vertical velocity command

Responses:

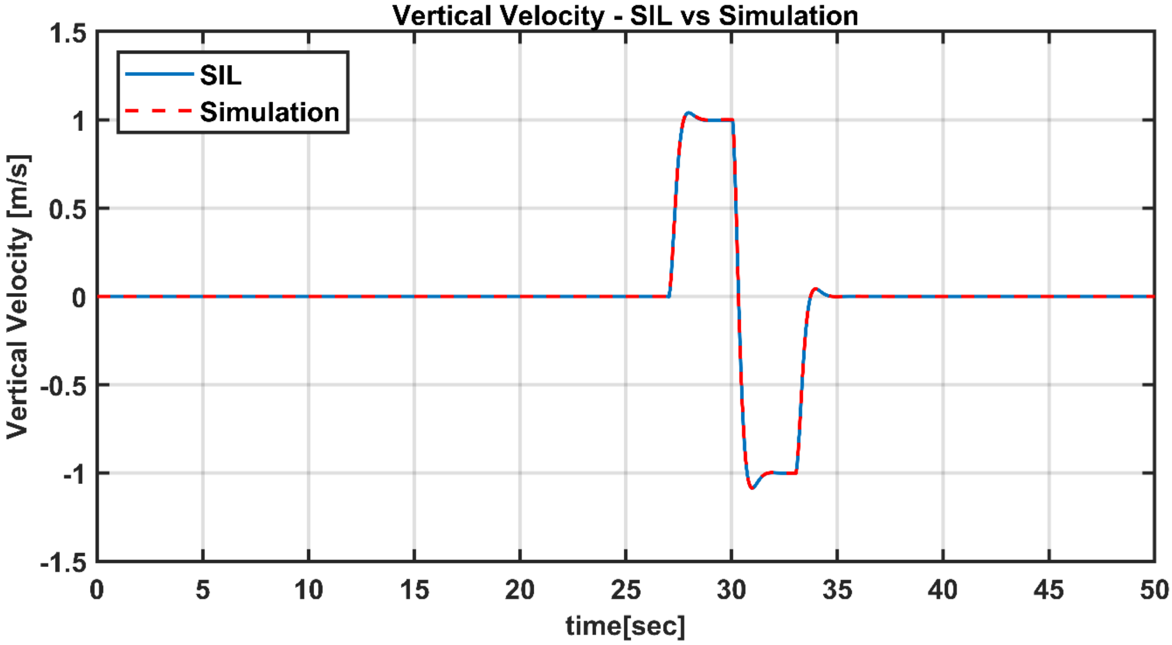


Figure 5-67 SIL test – Vertical velocity response

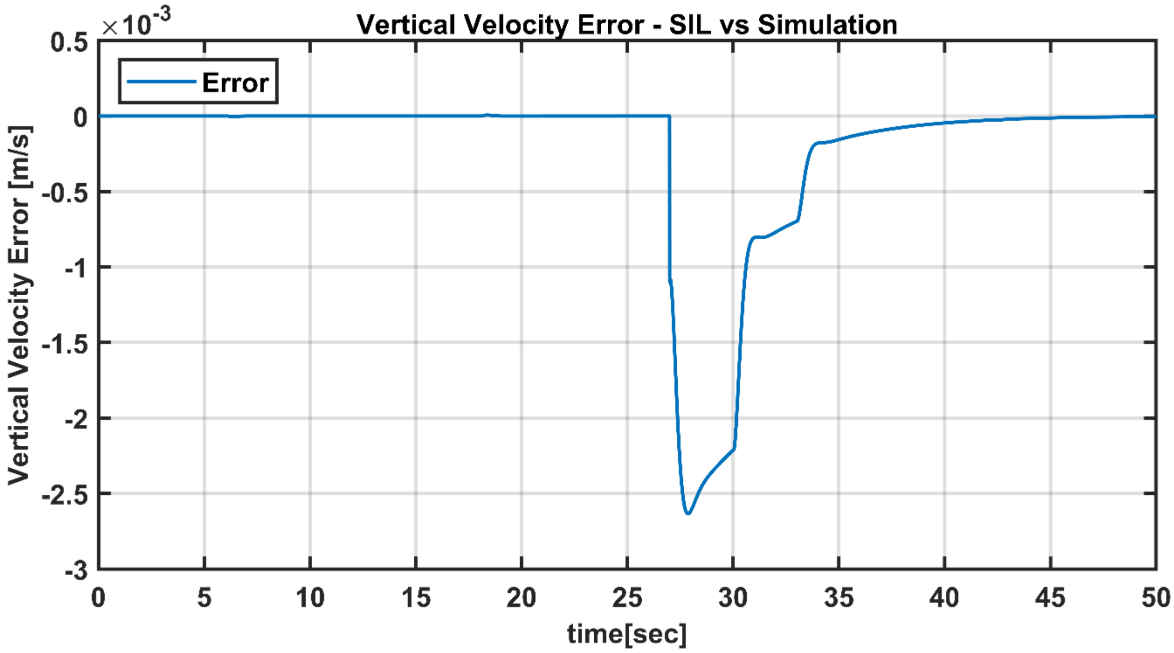


Figure 5-68 SIL test – Vertical velocity error

5.3.Hardware in the Loop Tests

Hardware in the loop test is a test environment where real-time hardware and software tests can be performed without risking the system. It is widely used in industries where testing on real systems is difficult and risky, such as aerospace, automotive. During these tests, while the software runs on the hardware, the dynamic model and other helper blocks runs on real-time computers called "target computers". In some studies, sensors are also included in the loop in order to see the effect of the sensors on the system on the software. On the other hand, HIL test is critical for controllers with high processing power such as MPC, as it shows the effects of factors affecting controller performance such as execution time and communication time.

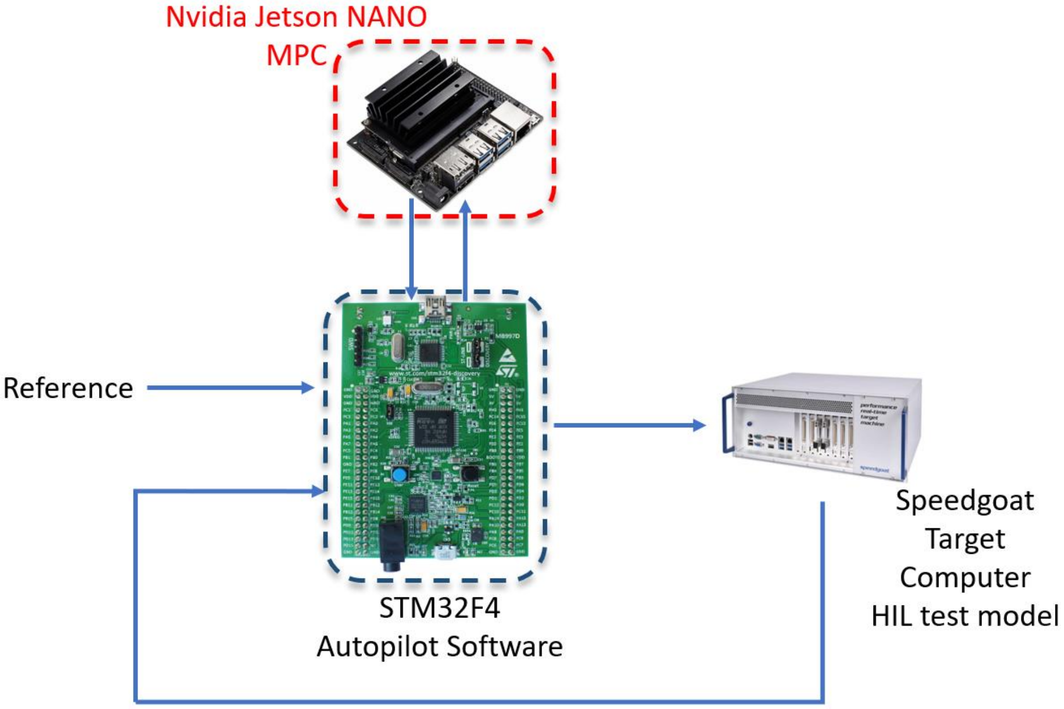


Figure 5-69 Hardware in the Loop Tests Overview

Computation time is crucial for real-time MPC performance. Optimization of the cost function in each time step requires high computation power and creates latency for the control system. This latency degrades MPC performance. For example, in this thesis work, LMPC has run on STM32F04 board at first, and it comes with around +100 ms computation time. MPC has been designed to work at 12.5 Hz, so it was impossible to run MPC on this board. Therefore, Jetson Nano has been chosen as hardware. Jetson NANO has shown acceptable performance around 10 ms.

5.3.1. Hardware in the Loop Test Setup

The setup for hardware in the loop test has been shown in Figure 5-70. In this structure, the system model and the signal generator that will simulate the system inputs worked on a Speedgoat target computer at 400 Hz. The existing autopilot software is included in the system for tasks such as providing communication with sensors and RC control and turning off the engines in emergency situations. Therefore, it is also included in the HIL test to see the effect of transactions and communication.

The test scenario can be summarized as follows. The commands generated from the HIL Simulink model operating at 400 Hz are sent to the STM32F4 card over the UART channel together with the responses of the dynamic model. This communication was at 100 Hz and at 460800 baud rate. Then, the autopilot software checks the flight mode and arm status. If the system is in speed or attitude mode, it sends information including the flight mode, reference commands and sensor measurements to Jetson NANO at 460800 baud rate via another UART channel. MPC software runs on Jetson NANO and transmits the control inputs that should be applied to STM32F04 by using the relevant weight set according to the flight mode. Then, STM32F04 multiplies these control inputs with the mixing matrix, converts them into PWM as motor commands and sends them back to the target computer. The target computer finally closes the loop by obtaining the results of the dynamic model against the control inputs and sending it to STM32F04 with the new reference commands. In order to measure the delay occurring on this system, it is logged again on the Speedgoat by proceeding with a counter-message created on target PC. Due to asynchronous communication, one or two data set may be missed from time to time on the STM32F04. This can create an extra 10 or 20 ms delay on the system.

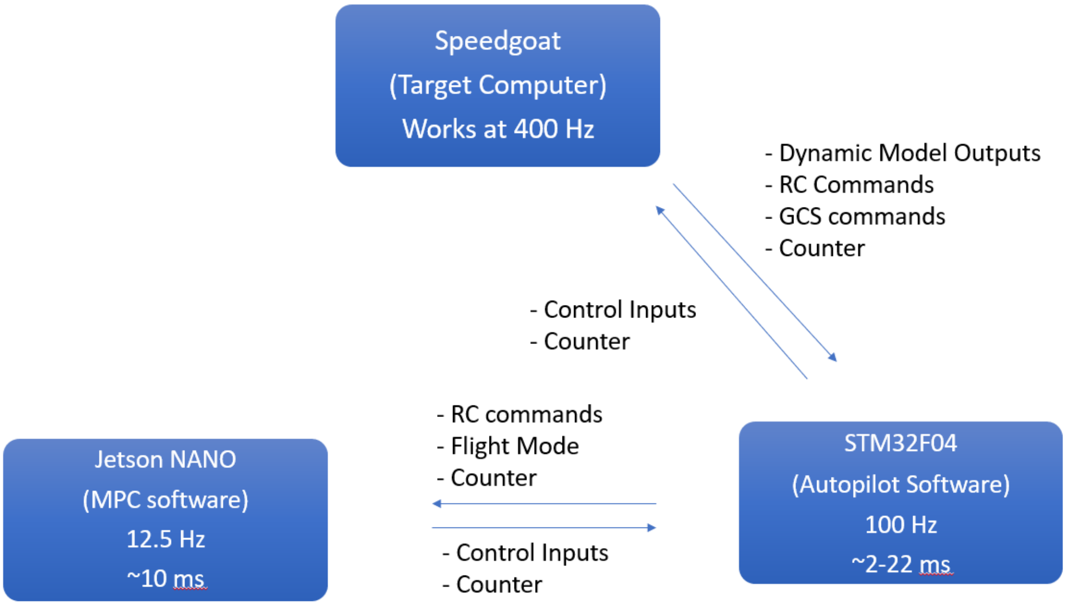


Figure 5-70 HIL test setup connections

RC Commands [rad] in Attitude [m/s] in Speed	GCS commands	Control Inputs	Dynamic Model Outputs
Lateral Cmd.	Flight Mode (Speed / Attitude)	Lateral Motor Cmd.	Longitudinal Speed in Body Frame (u)
Longitudinal Cmd.	Arm / Disarm	Longitudinal Motor Cmd.	Lateral Speed in Body Frame (v)
Vertical Cmd.		Vertical Motor Cmd.	Vertical Speed in Body Frame (w)
Directional Cmd.		Directional Motor Cmd.	Roll Rate (p)
			Pitch Rate (q)
			Yaw Rate (r)
			Roll Angle (ϕ)
			Pitch Angle (θ)
			Yaw Angle (ψ)
			Velocity North in NED Frame
			Velocity East in NED Frame
			Velocity Down in NED Frame

Table 5-2 Message Packet Contents

5.3.2. Results of Attitude Mode

In this section, the results of the hardware in the loop tests against the doublet manoeuvre have been shown. As the pilot in attitude mode controls the vertical channel, no input is given to the vertical channel during this test. In order to keep the system stable, this channel input was fed as zero in the dynamic model during the test.

5.3.2.1. Directional Channel

Test Input:

To perform this test, 0.3 rad/s yaw rate command has been applied.

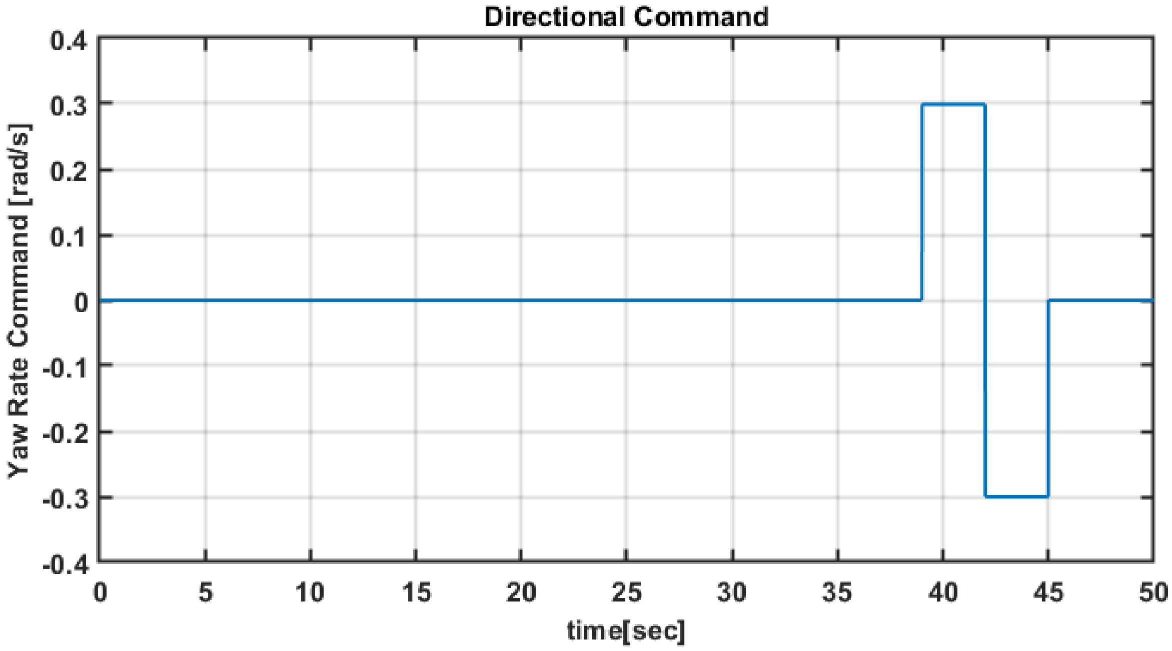


Figure 5-71 HIL test - Directional command

Responses:

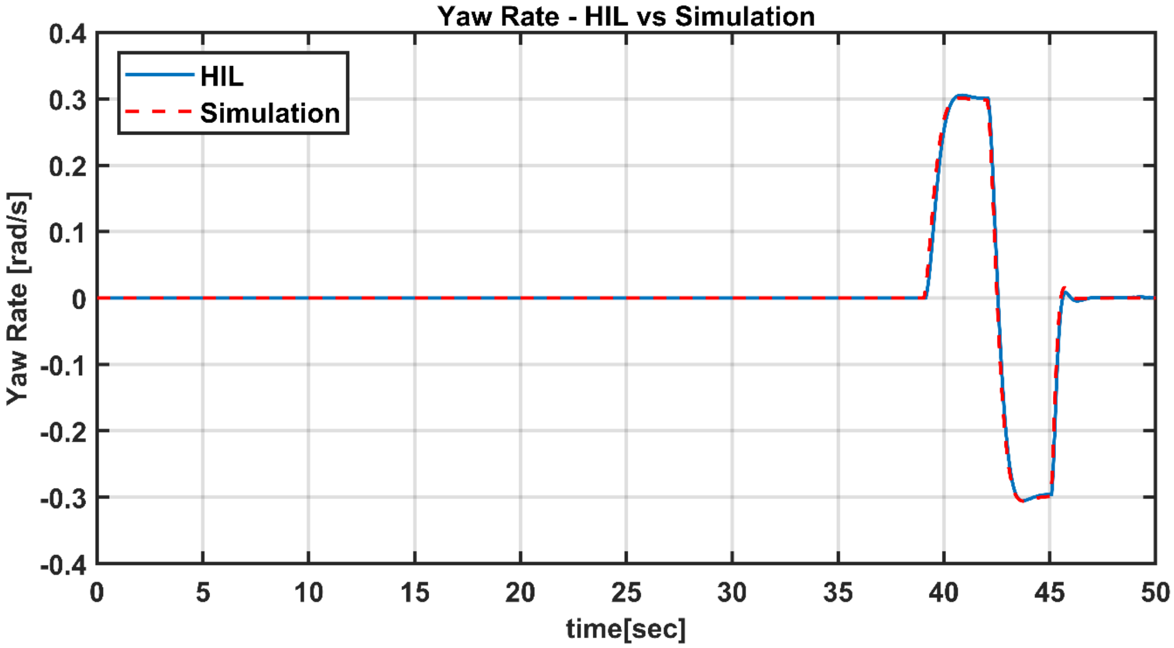


Figure 5-72 HIL test - Yaw rate responses

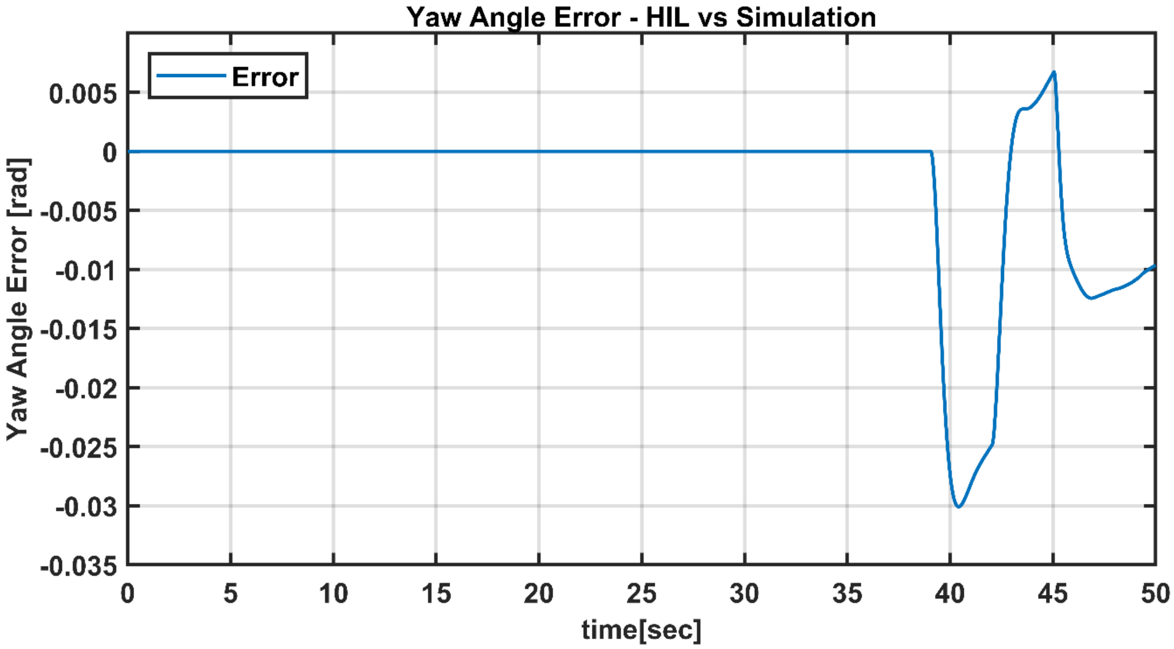


Figure 5-73 HIL test - Yaw rate error

As seen in Figure 5-72, Figure 5-73 the HIL tests of the directional channel were performed with very low errors. As mentioned before, except for the error caused by the PWM conversion, the test was performed successfully.

5.3.2.2. Lateral Channel

In the lateral channel, the system input is given as angle command. Test input is shown in Figure 5-74.

Test Input:

To perform this test, 0.3 rad/s roll angle command has been applied.

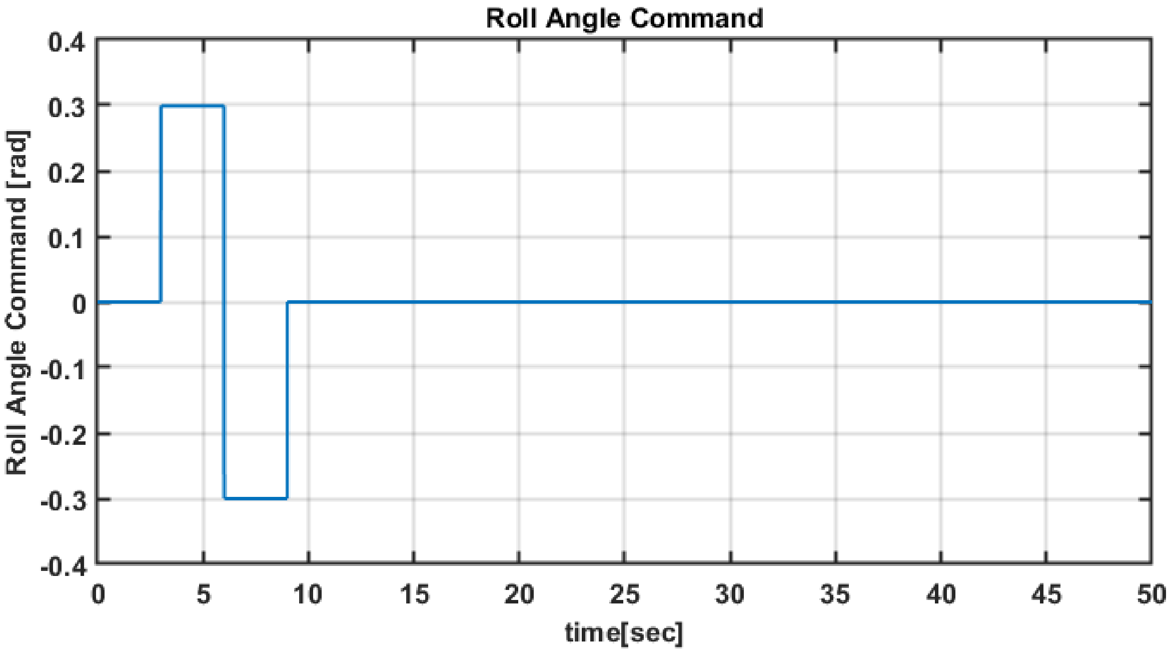


Figure 5-74 HIL test - Roll command

Responses:

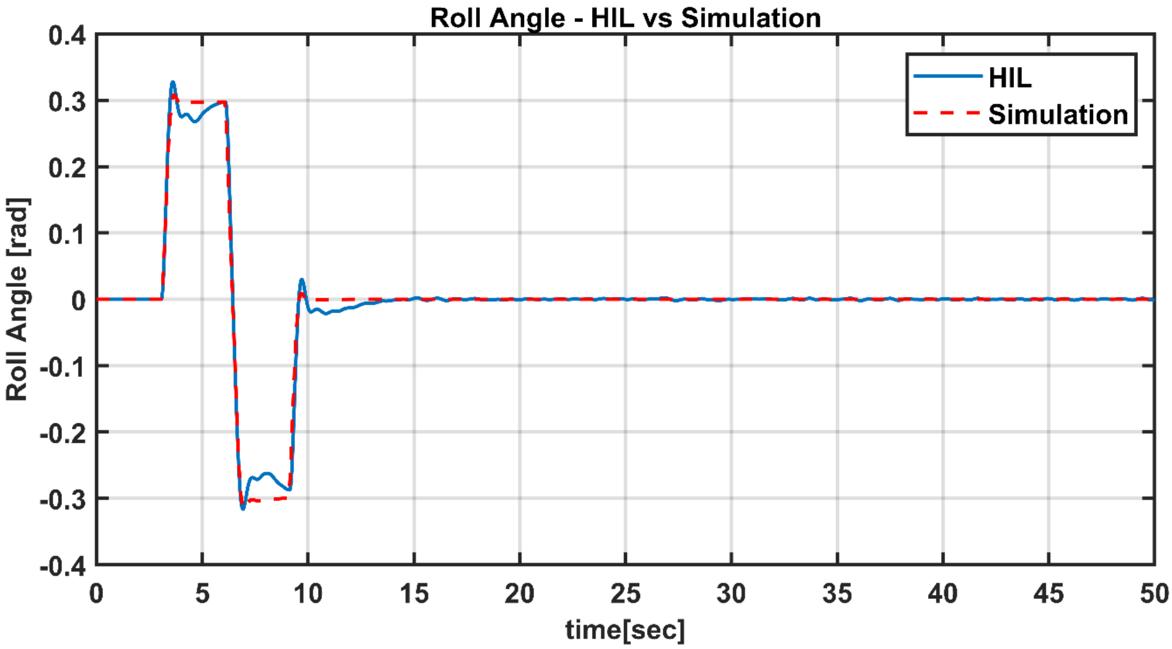


Figure 5-75 HIL test - Roll angle response

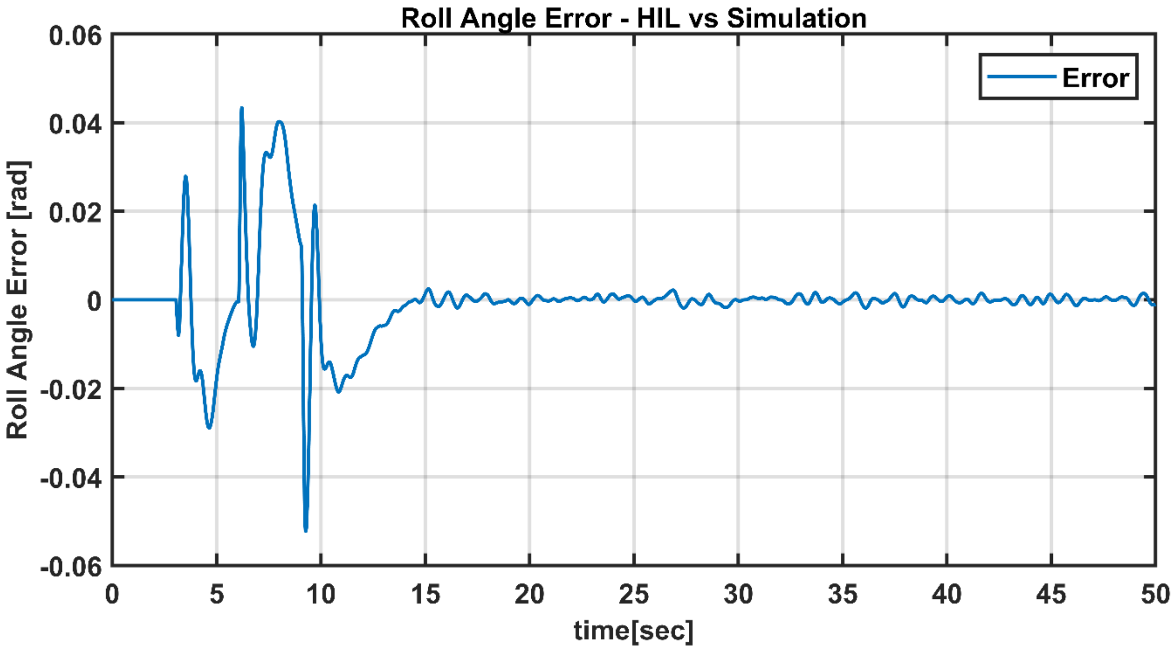


Figure 5-76 HIL test - Roll angle error

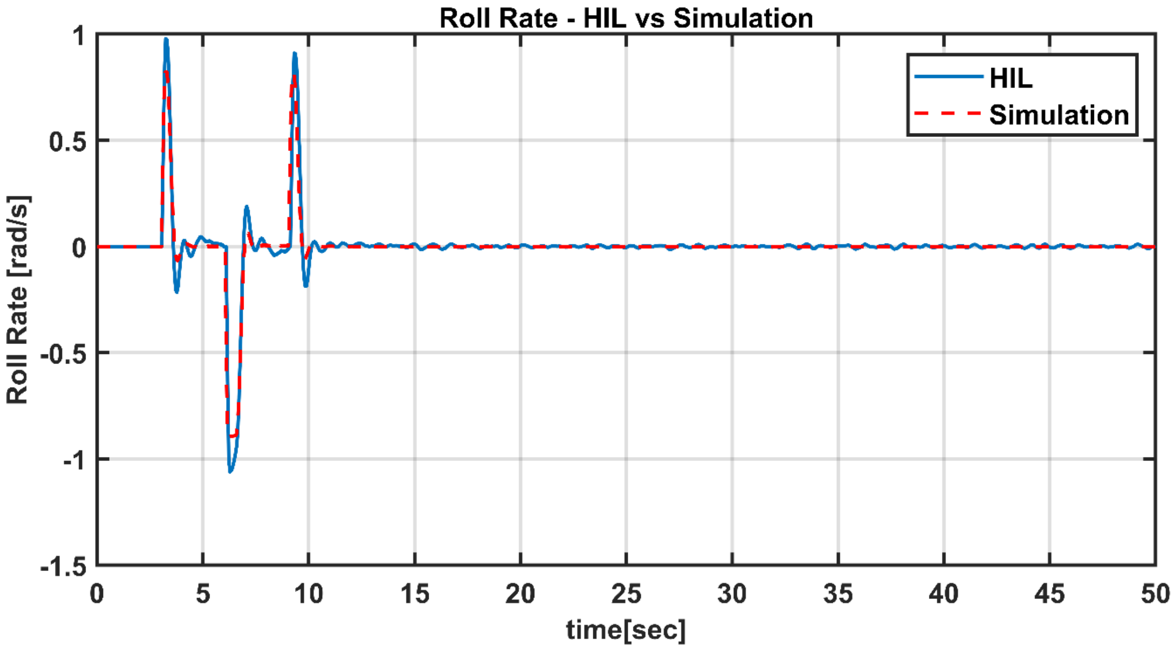


Figure 5-77 HIL test - Roll rate response

During the lateral channel tests, HIL tests results were not as successful as the directional channel. As mentioned before, because of the asynchronous communication, there were uncertain latencies on the test system. This latency probably affected the lateral channel more than the directional, because of its aggressive and unstable dynamics.

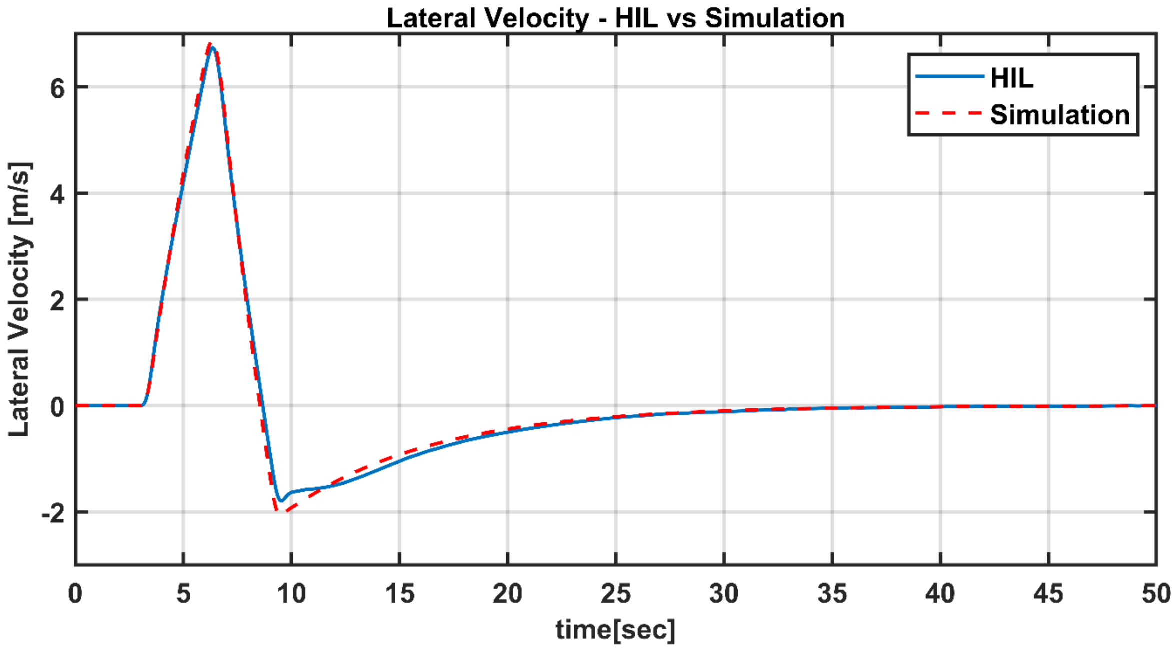


Figure 5-78 HIL test - Lateral velocity response

5.3.2.3. Longitudinal Channel

In the longitudinal channel, the system input is given as angle command. Test input is shown in Figure 5-79.

Test Input:

To perform this test, 0.3 rad pitch angle command has been applied.

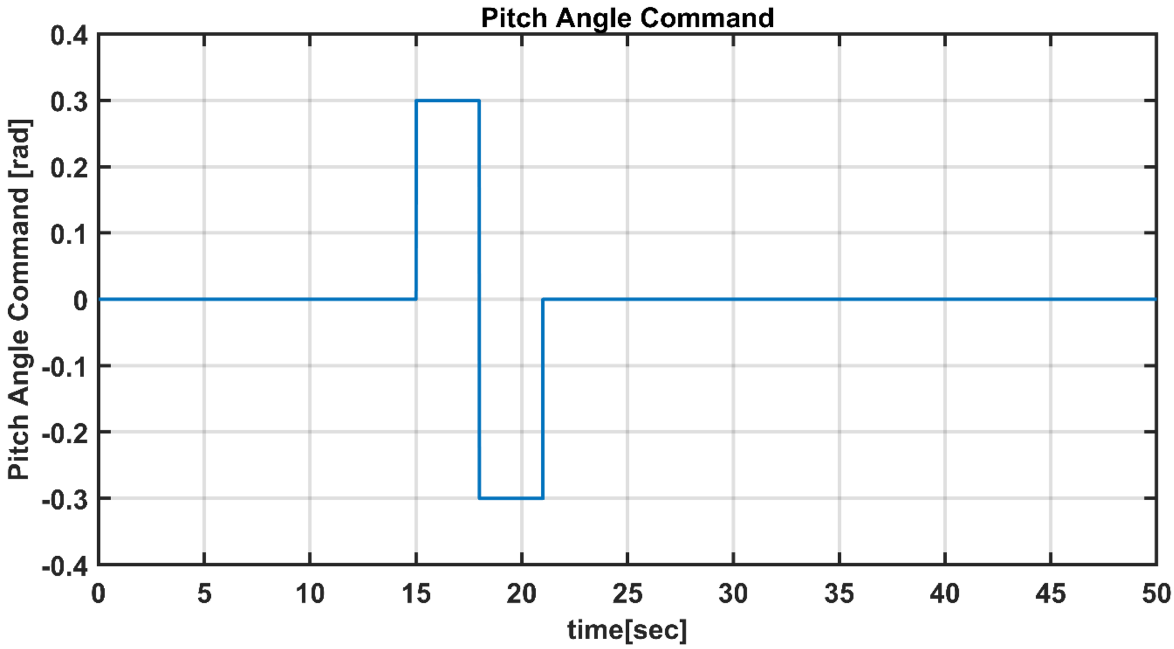


Figure 5-79 HIL test - Pitch command

Responses:

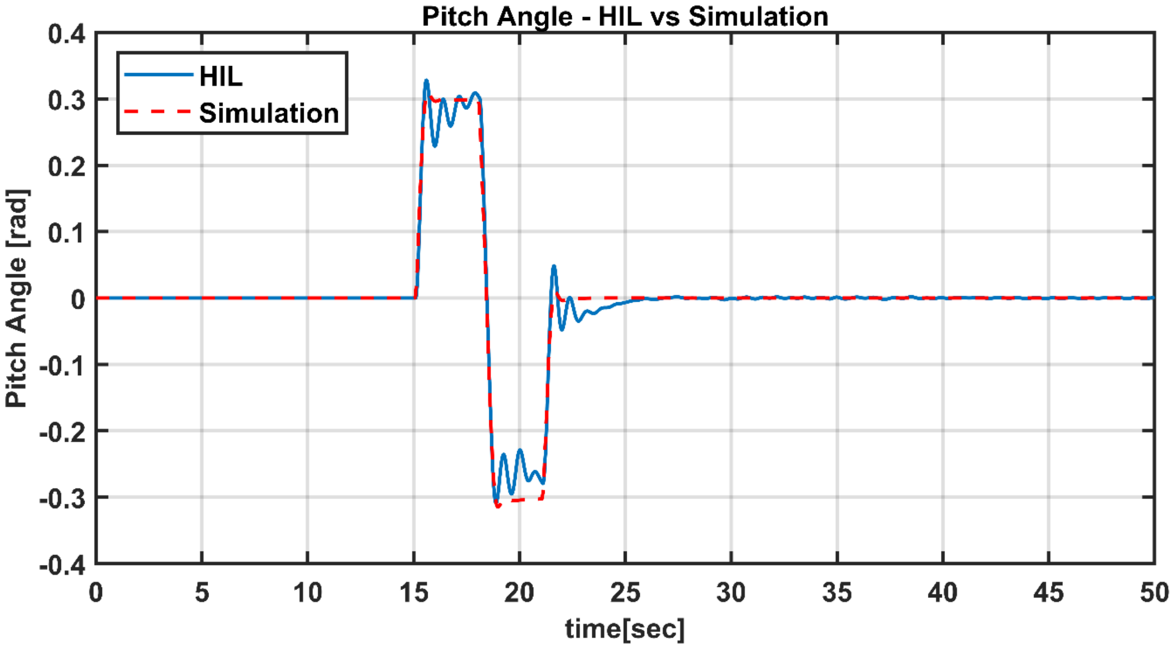


Figure 5-80 HIL test - Pitch angle response

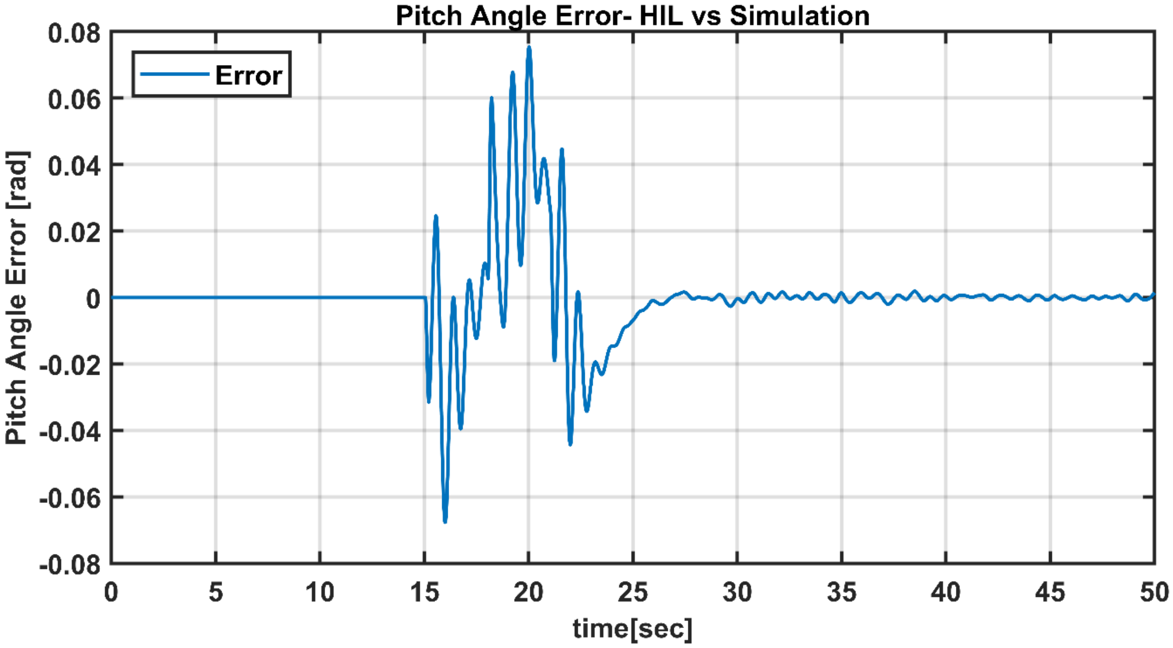


Figure 5-81 HIL test - Pitch angle error

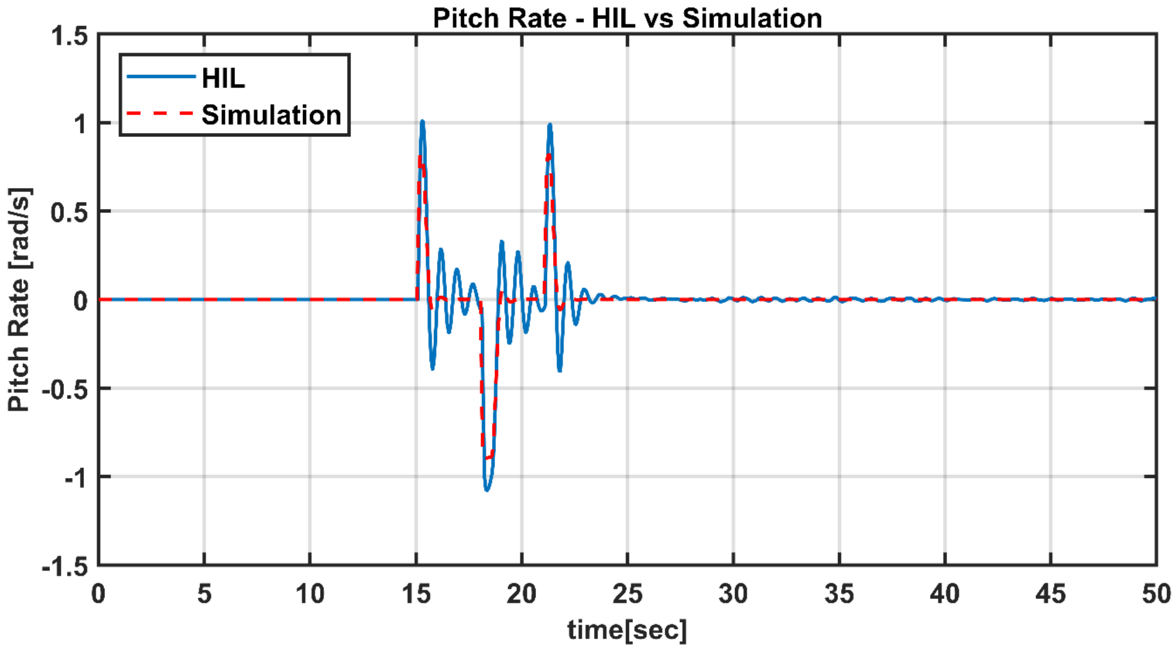


Figure 5-82 HIL test - Pitch rate response

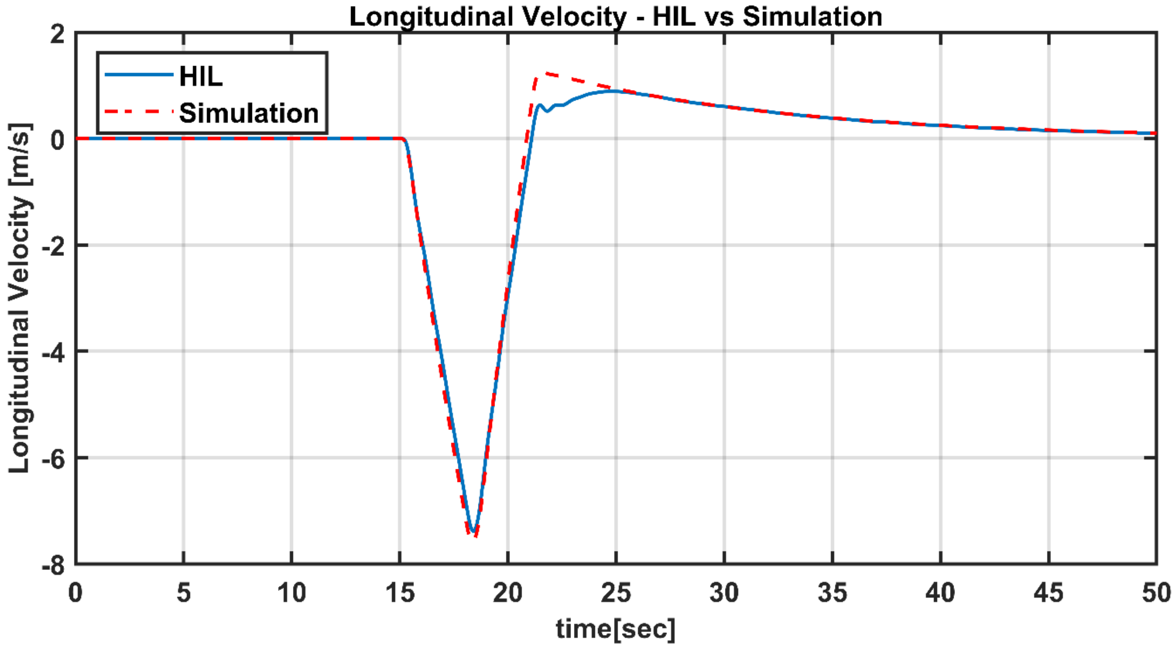


Figure 5-83 HIL test - Longitudinal velocity response

During the longitudinal channel tests, HIL tests results were not as successful as the directional channel. As mentioned before, because of the asynchronous communication, there were uncertain latencies on the test system. This latency probably affected the lateral channel more than the directional, because of its aggressive and unstable dynamics.

5.3.3. Results of Speed Mode

In this section, the results of the software in the loop tests against the doublet manoeuvre have been shown. During the tests performed in this mode, the pilot commanded speed in the longitudinal, lateral and vertical channels, and yaw rate in the directional channel.

5.3.3.1. Directional Channel

In the directional channel, if the input is non-zero, the controller operates in the angular velocity loop, and if the command is zero, the controller operates in the angle loop.

Test Input:

To perform this test, 0.3 rad/s yaw rate command has been applied.

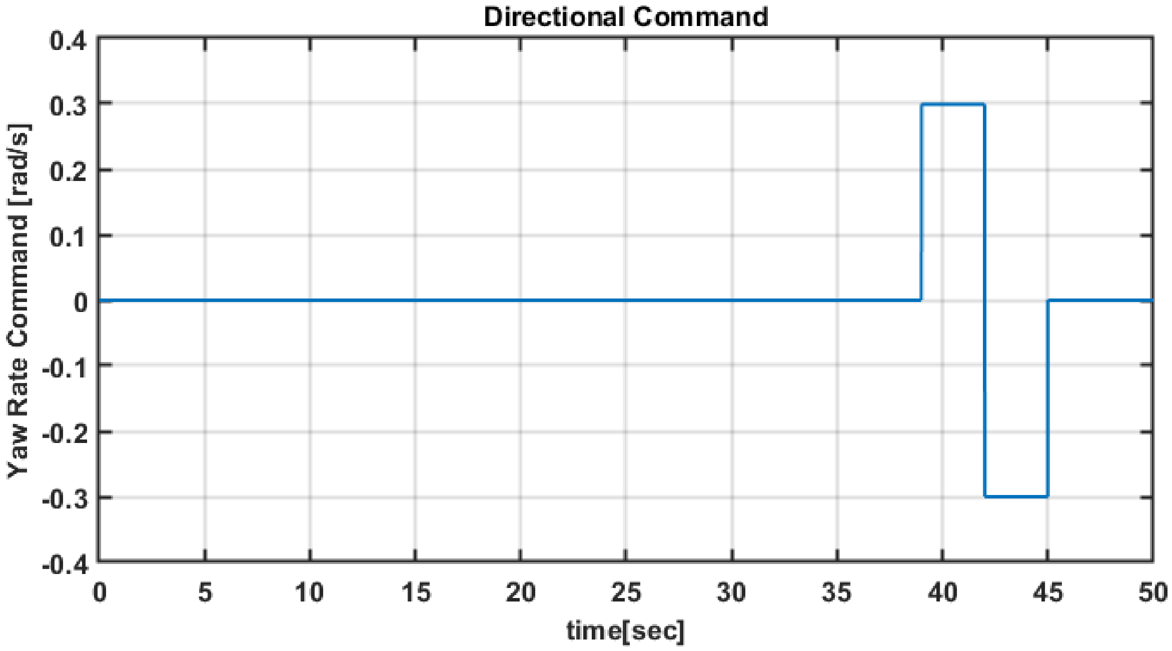


Figure 5-84 HIL test - Directional command

Responses:

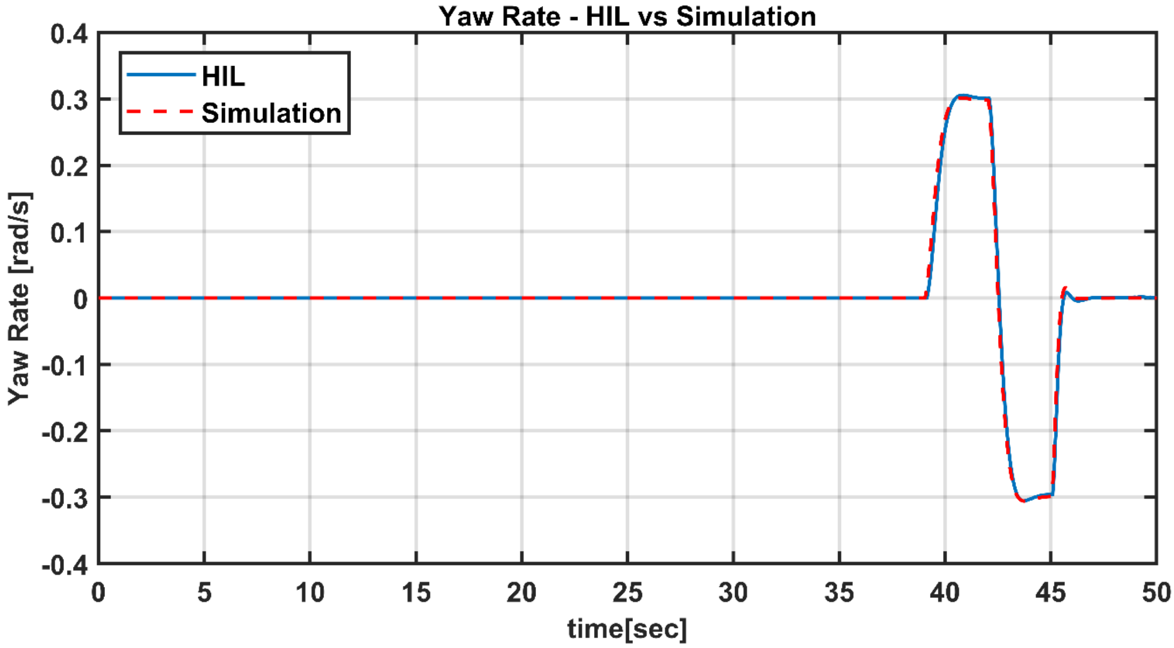


Figure 5-85 HIL test - Yaw rate responses

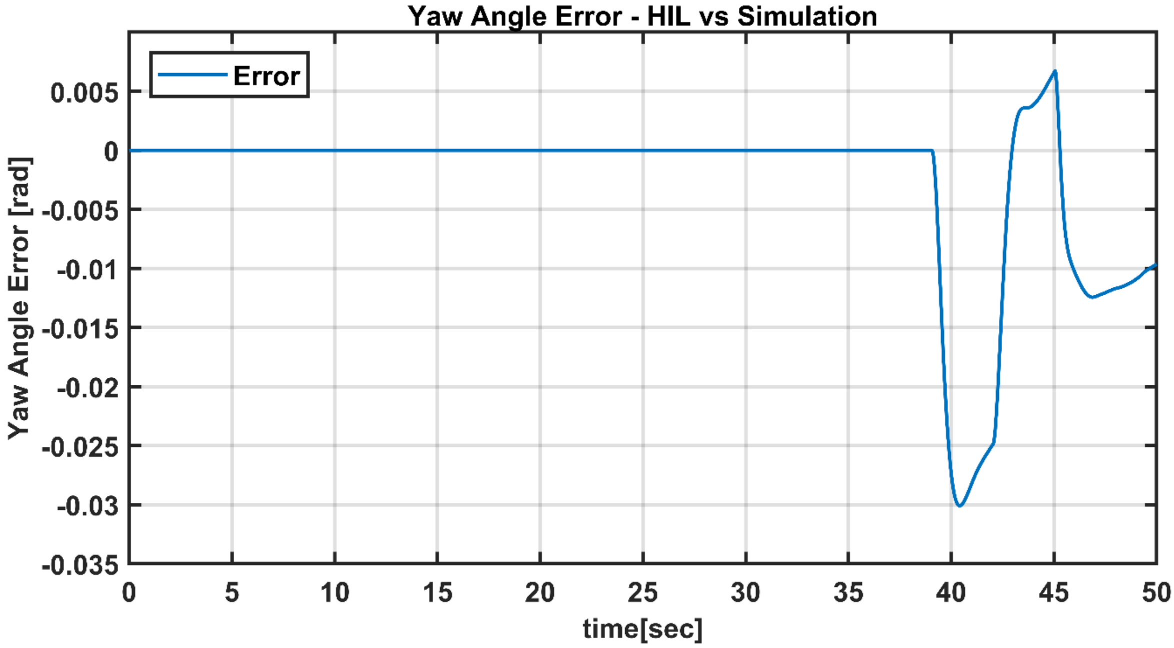


Figure 5-86 HIL test - Yaw rate error

As seen in Figure 5-85, Figure 5-86 the HIL tests of the head angle channel were performed with fewer errors. As mentioned before, except for the error caused by the PWM conversion, the test was performed successfully.

5.3.3.2. Lateral Channel

In the lateral channel, the system input is given as angle command. Test input is shown in Figure 5-87.

Test Input:

To perform this test, 1 m/s lateral velocity command has been applied.

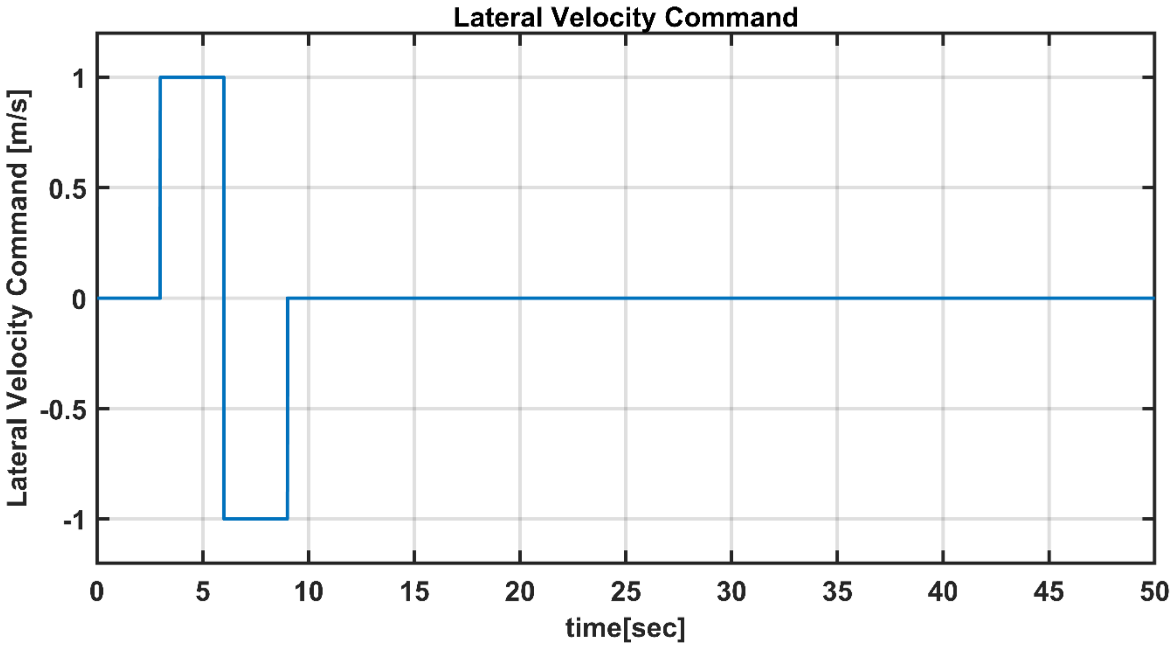


Figure 5-87 HIL test – Lateral velocity command

Responses:

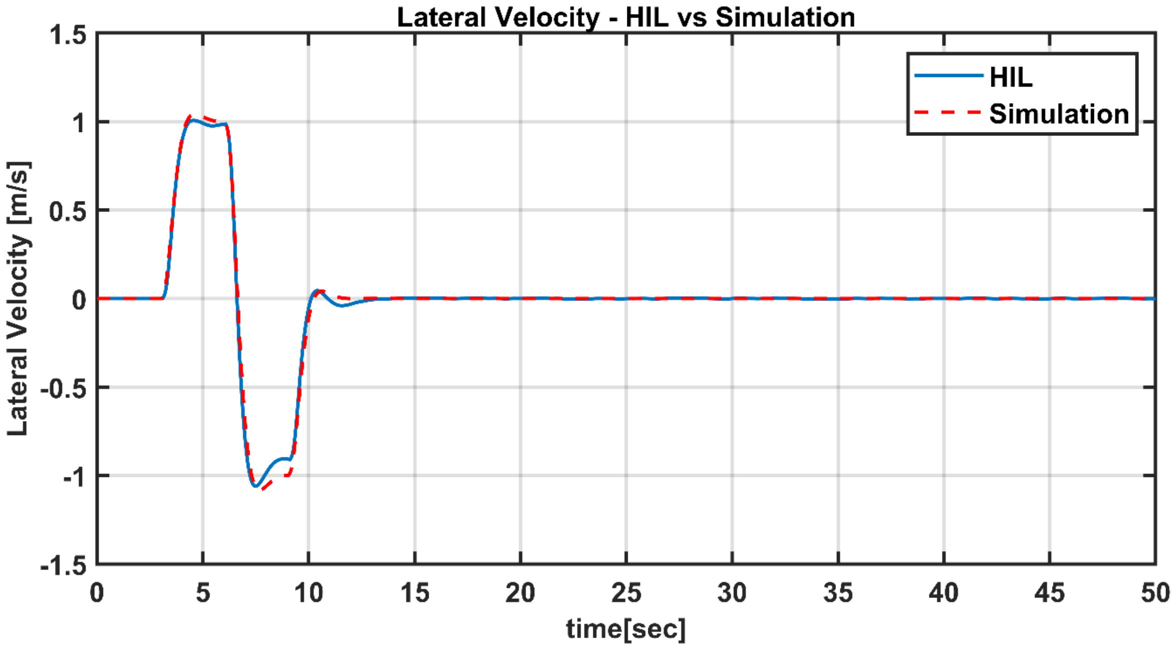


Figure 5-88 HIL test – Lateral velocity response

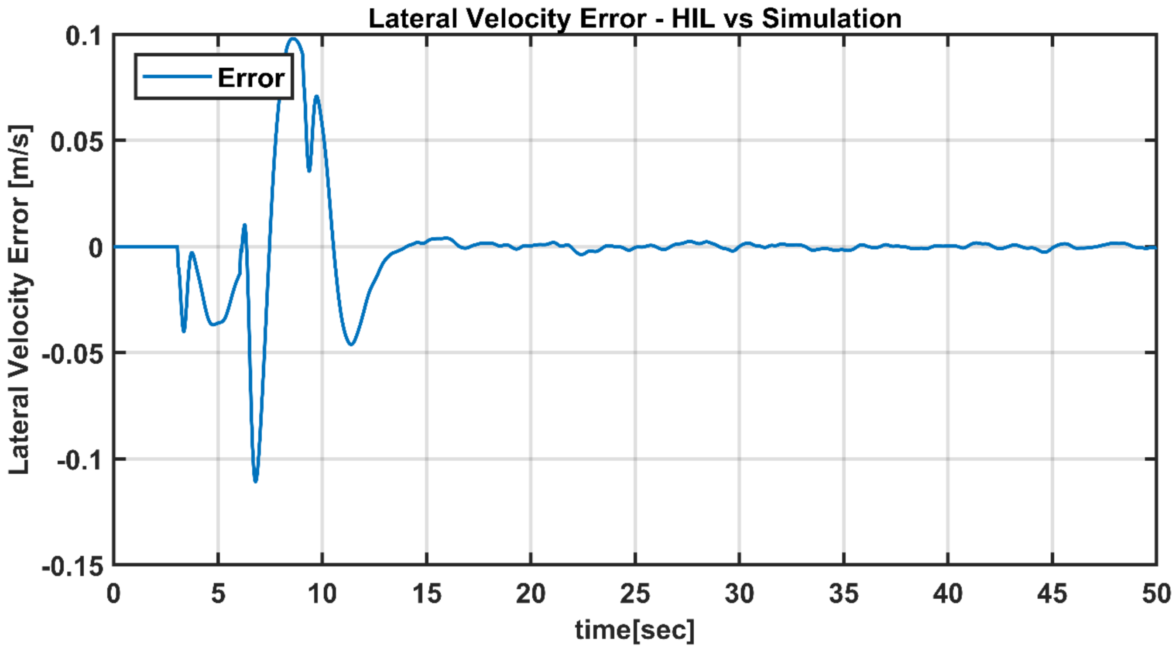


Figure 5-89 HIL test – Lateral velocity error

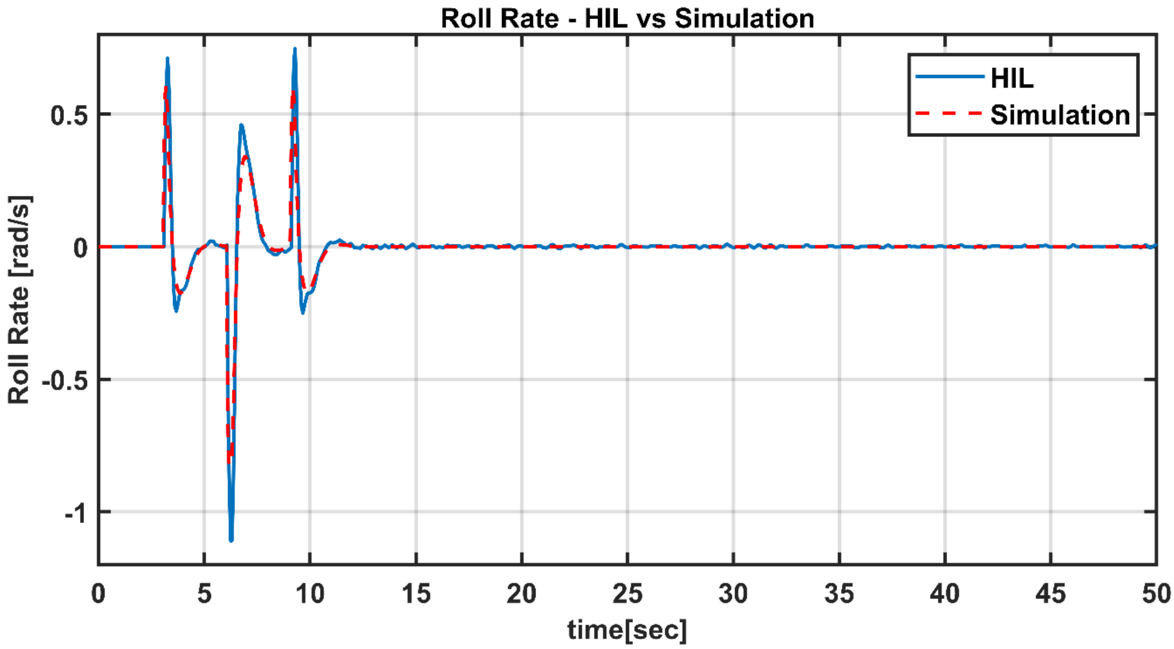


Figure 5-90 HIL test - Roll rate response

During the lateral channel tests, HIL tests results were not as successful as the directional channel. But in this mode, controller is not as aggressive as attitude mode. So, the controller can handle the uncertainties more appropriate.

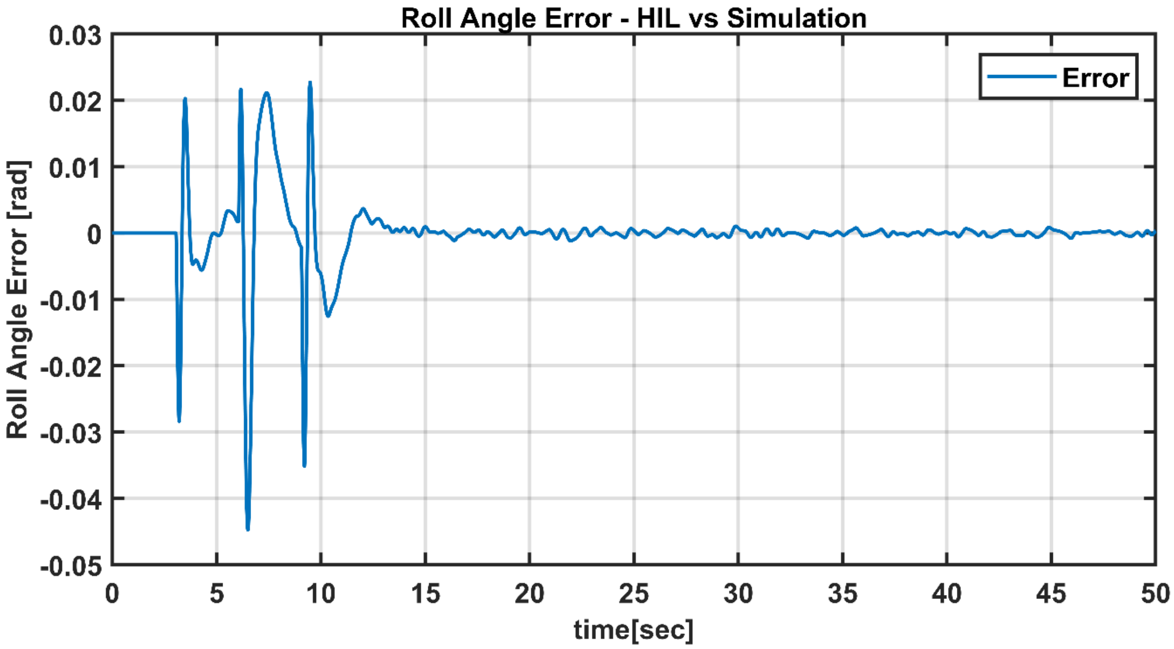


Figure 5-91 HIL test – Roll angle response

5.3.3.3. Longitudinal Channel

In the longitudinal channel, the system input is given as angle command. Test input is shown in Figure 5-47.

Test Input:

To perform this test, 1 m/s longitudinal velocity command has been applied.

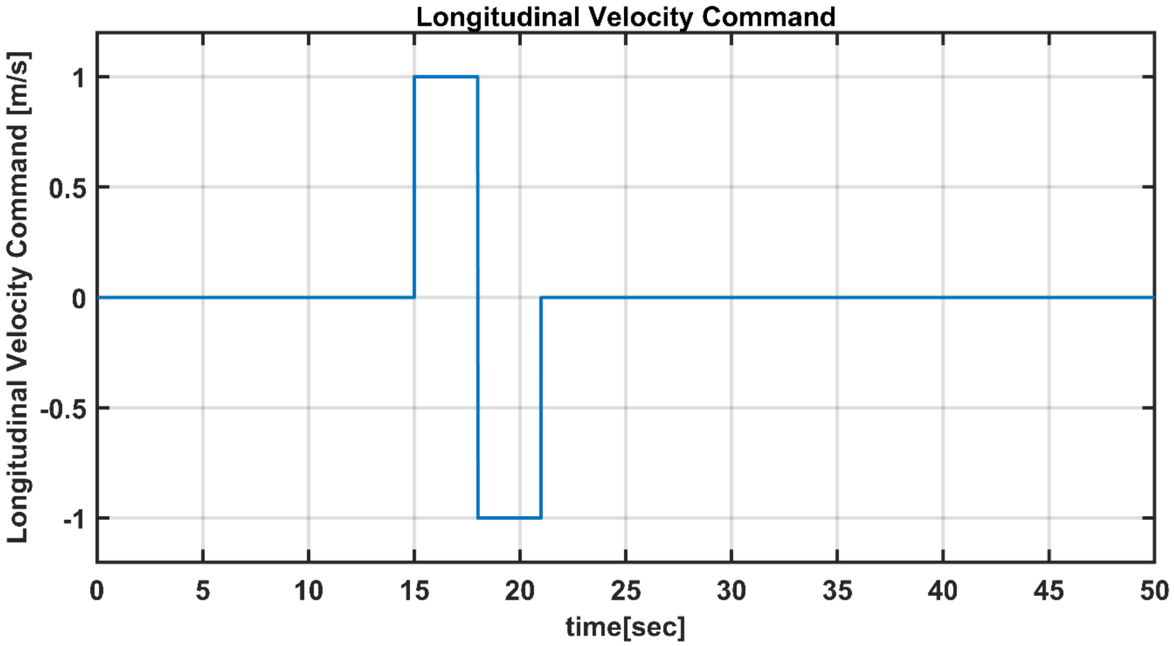


Figure 5-92 HIL test – Longitudinal velocity command

Responses:

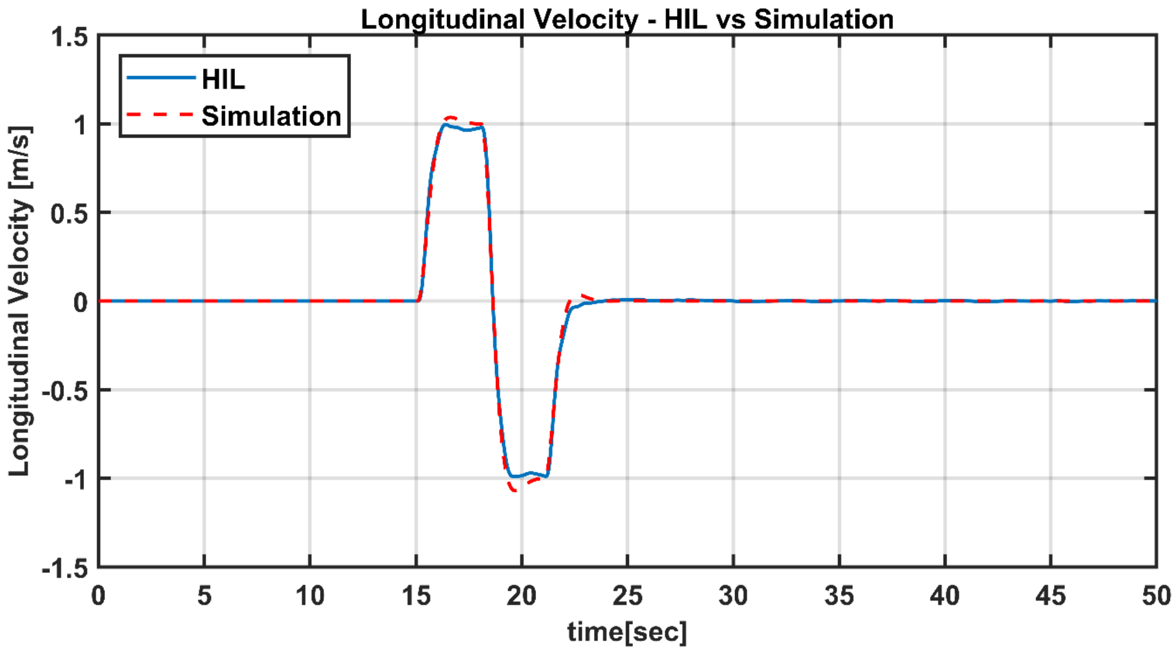


Figure 5-93 HIL test – Longitudinal velocity response

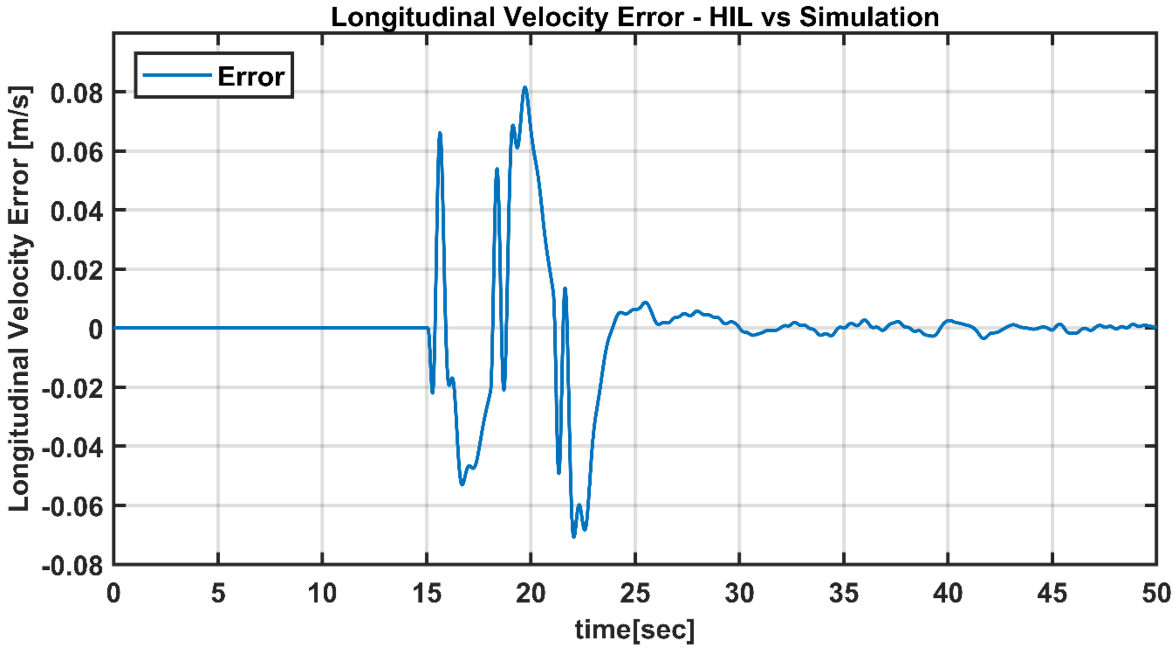


Figure 5-94 HIL test – Longitudinal velocity error

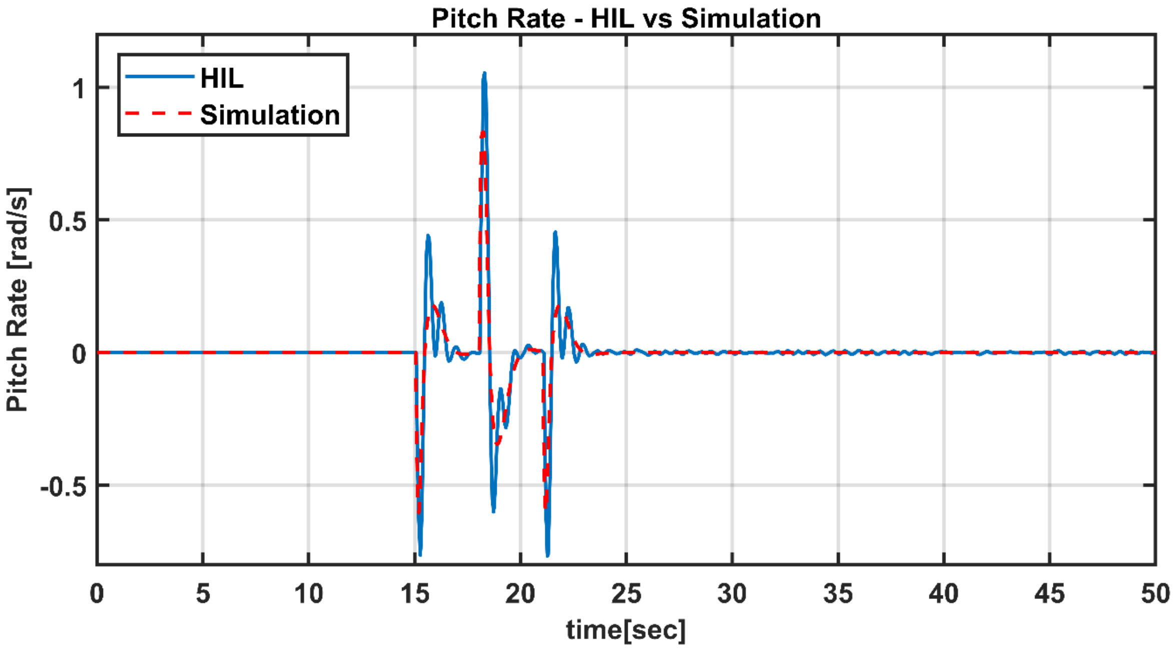


Figure 5-95 HIL test - Pitch rate response

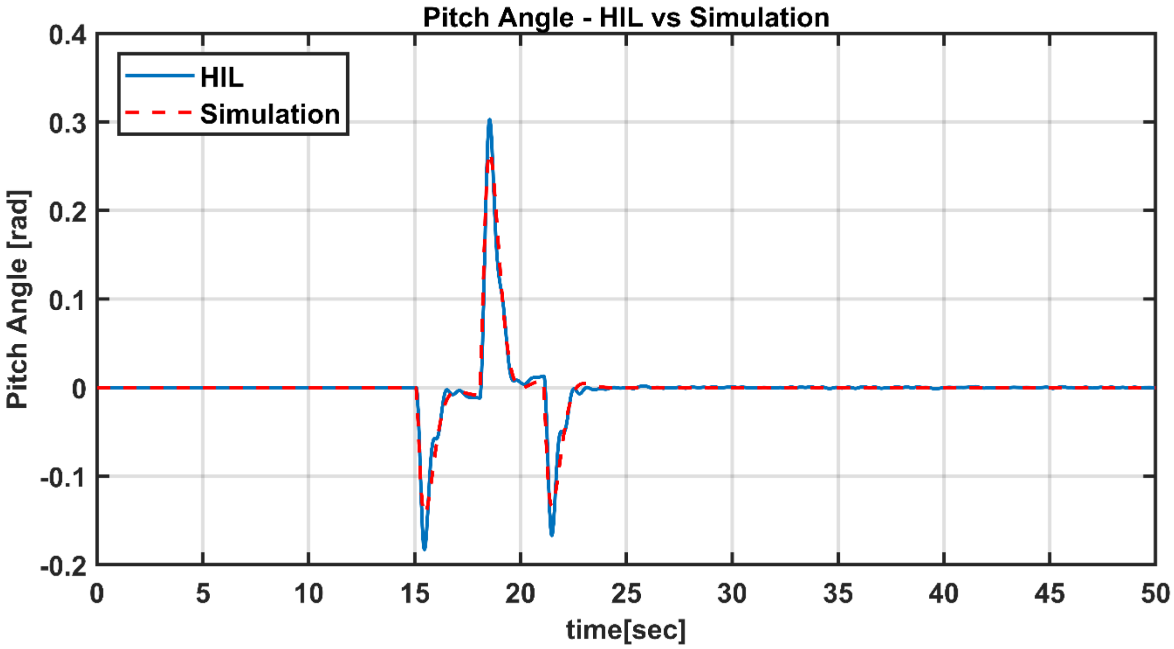


Figure 5-96 HIL test – Pitch angle response

During the lateral channel tests, HIL tests results were not as successful as the directional channel. But in this mode, controller is not as aggressive as attitude mode. So, the controller can handle the uncertainties more appropriate.

5.3.3.4. Vertical Channel

In the longitudinal channel, the system input is given as angle command. Test input is shown in Figure 5-97.

Test Input:

To perform this test, 1 m/s vertical velocity command has been applied.

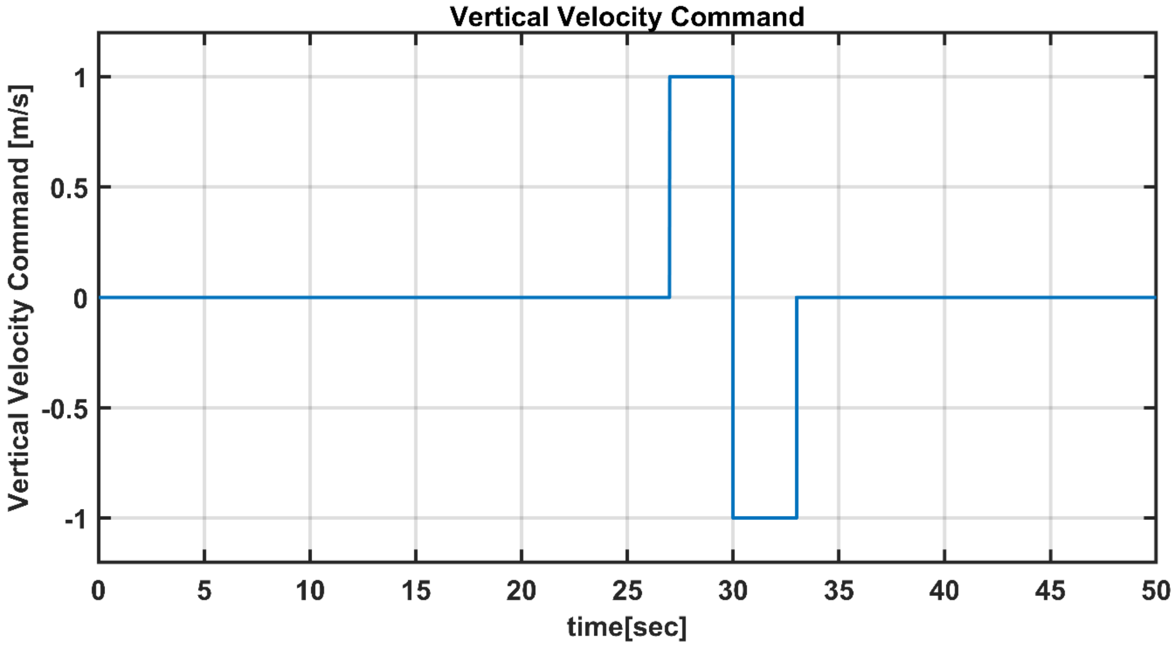


Figure 5-97 HIL test – Vertical velocity command

Responses:

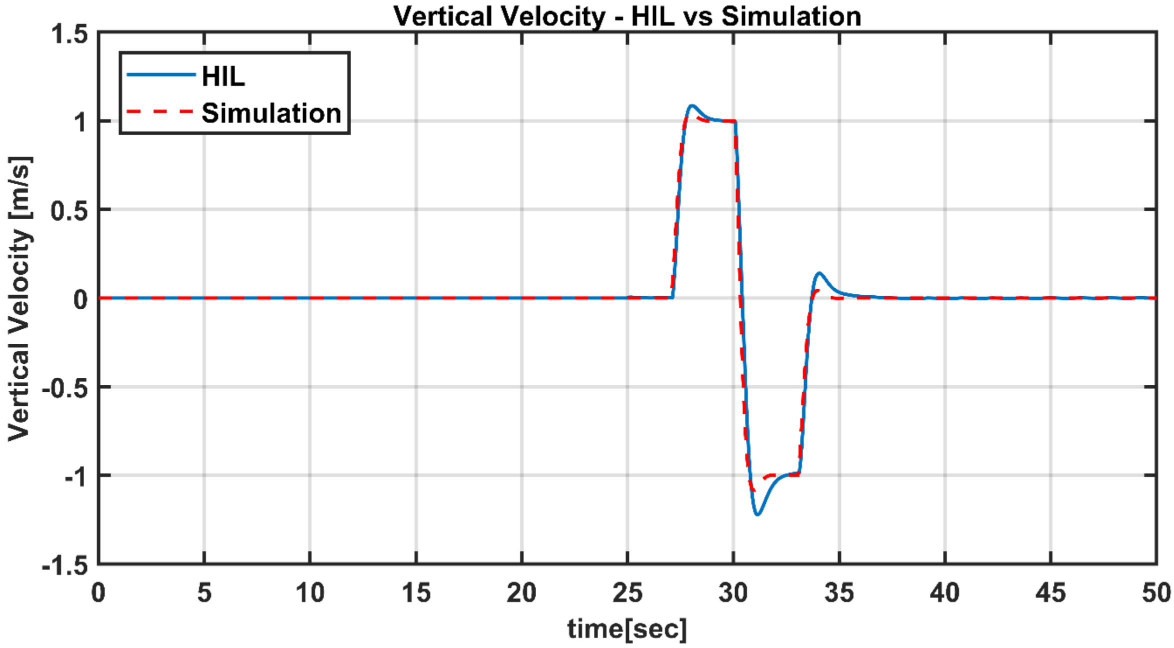


Figure 5-98 HIL test – Vertical velocity response

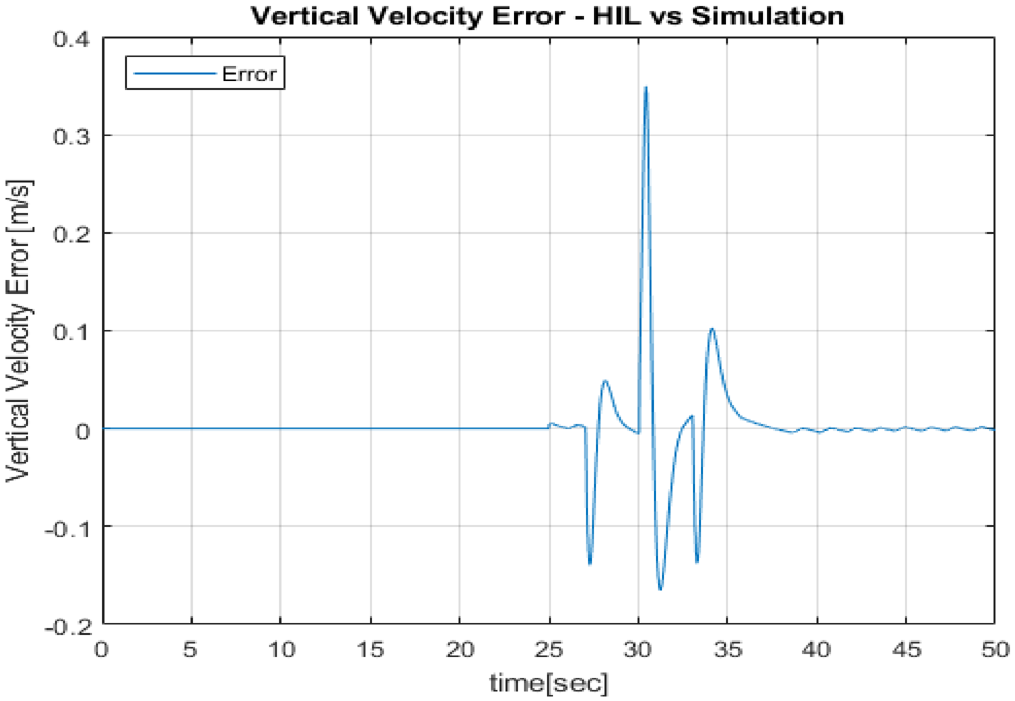


Figure 5-99 HIL test – Vertical velocity error

The vertical channel also had some error due to uncertain communication error, but these errors are in an acceptable range.

5.4. Flight Tests

In this part of the thesis work, the generated code has been implemented to the hardware with the HIL test configuration. Flight tests were repeated many times with different model time delays and controller weights. After obtaining a satisfactory result in attitude mode, tests have been done in speed mode. For a fair comparison with reference PID controller, flight tests have been done in speed mode. In earlier tries, the MPC has been responded aggressively and responded like a sine wave around 1.2 Hz. After this, the controller's responses were softened by increasing the controller's roll and pitch rate weights. In order to see the effect of time delay on the system during the test studies, the controllers with different time delays were tested with the same weight set. These time delays are chosen as 20-40-60-80-100 ms. During the design process, it was observed that the 80ms time delay agreed in Chapter 4 gave the most successful result. This time delay includes ~ 65 ms model time delay, which is obtained from system identification and ~10-15 ms MPC execution time. The delay between 0-20 ms due to asynchronous communication is considered as uncertainty and is not included in the model time delay.

Tests were carried out in outdoor conditions due to the lack of suitable indoor environment. For this reason, although tests have been tried in the most stable weather possible, the effect of wind has been observed from time to time. Inputs were given manually during the tests. RMS values of the errors were examined in order to compare the controller performances fairly.

Another factor that affected the data during the tests was the small structural changes that had to be made on the platform. During the system identification tests, Jetson NANO card, which was not included in the system because it was not intended to be used. It was added to the system after STM32 was found to be insufficient in HIL tests. This insertion has affected the dynamics in the roll and yaw axis. On the other hand, the damage that occurred on the platform over time also affected the platform dynamics.

5.4.1. Results of Speed Mode

This section covers MPC performance in speed mode during flight tests. System performance was tested by manually giving step inputs during the tests.

5.4.1.1. Directional Channel

Test Input:

To perform this test, ~0.5 rad/s yaw rate command has been applied.

Responses:

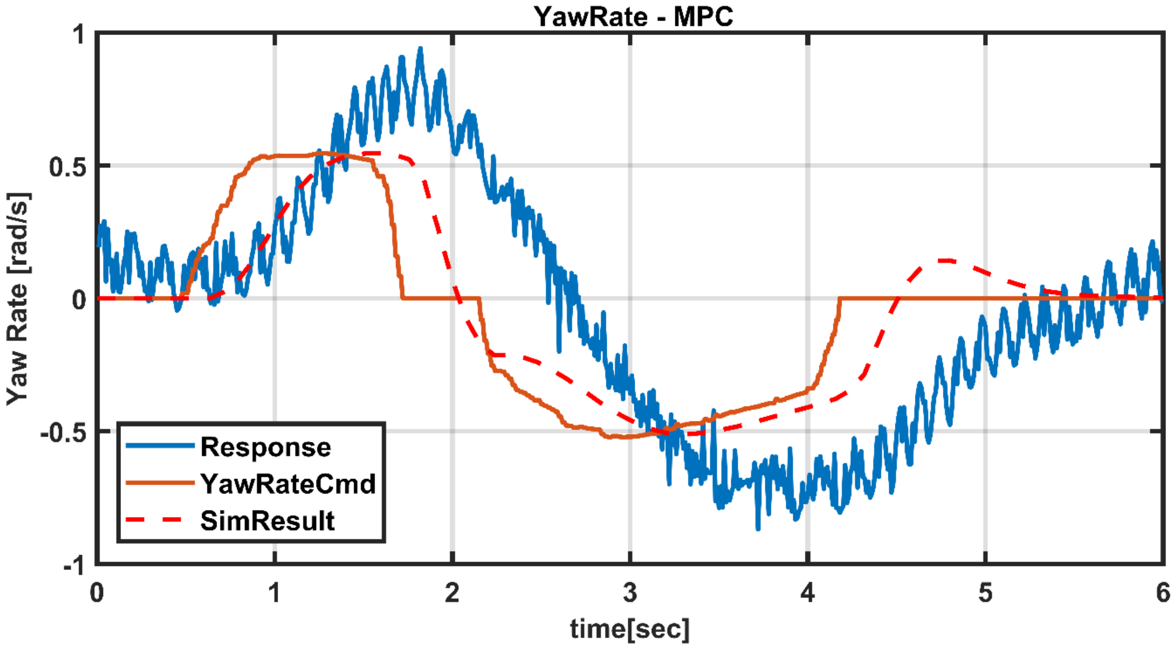


Figure 5-100 Flight Test - Yaw rate responses, MPC

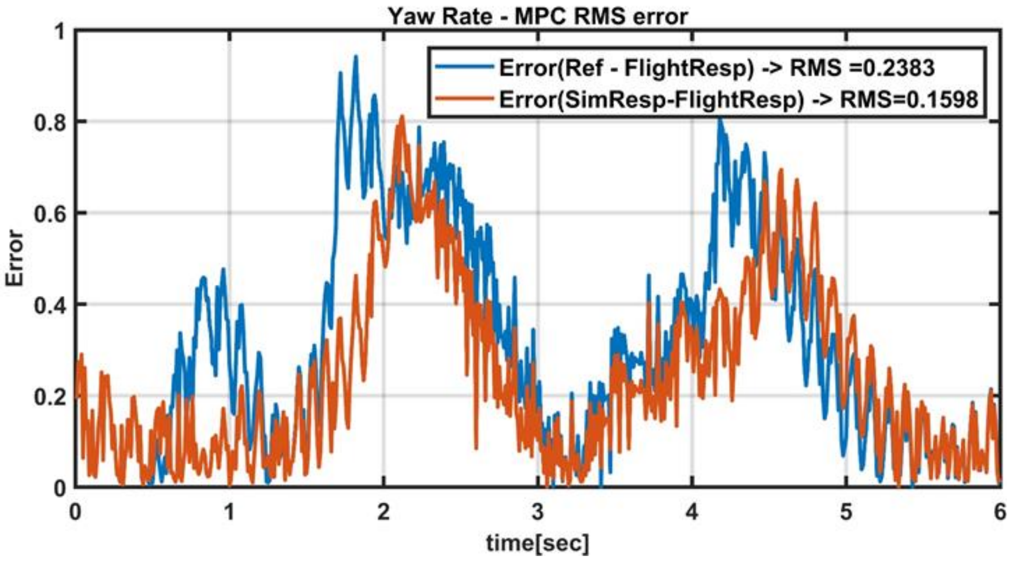


Figure 5-101 Yaw rate responses RMS error, MPC

RMS errors of reference - flight response and simulation response-flight response is given in Figure 5-101.

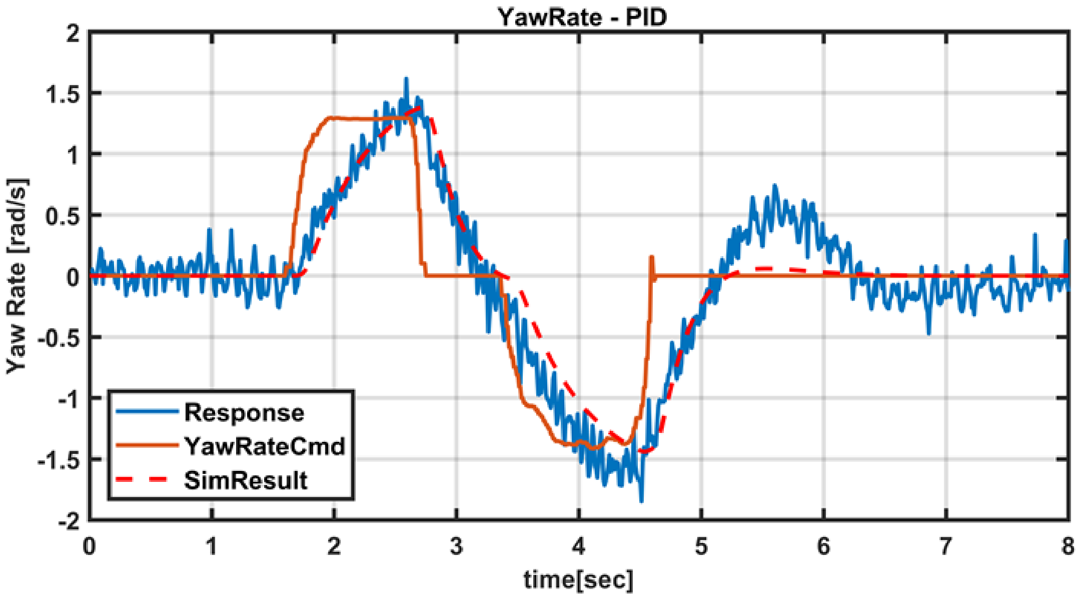


Figure 5-102 Flight Test - Yaw rate responses, PID

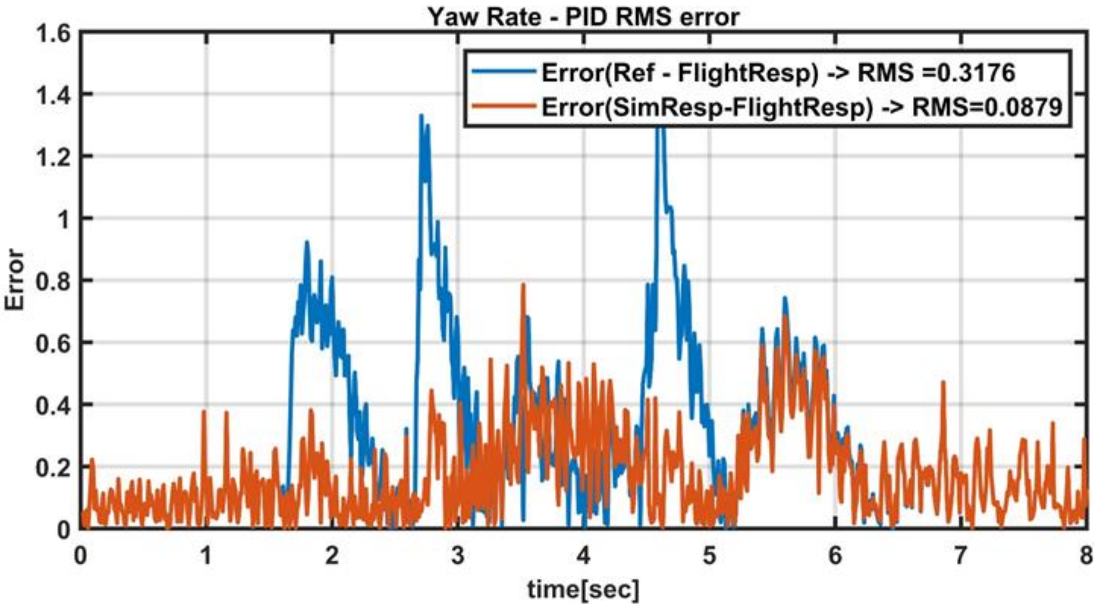


Figure 5-103 Yaw rate responses RMS error, PID

When the results in Figure 5-85 of the test performed in the Yaw channel are examined, it is seen that the system produces more aggressive responses than the results obtained in the simulation. When the Figure 3-25 obtained during the system definition is examined, it is seen that the system definition has been done successfully. Therefore, it is evaluated that the difference in response in this channel may also be due to structural problems. The performance of the directional channel has been deemed adequate as it has a low effect on platform stability. But, PIDs response were so close to its simulation. This showed that PID controller is more robust than the model uncertainties as expected.

5.4.1.2. Lateral Channel

Test Input:

To perform this test, ~0.7 m/s lateral velocity command has been applied for MPC and around 1.2 m/s lateral velocity for PID during flight.

Responses:

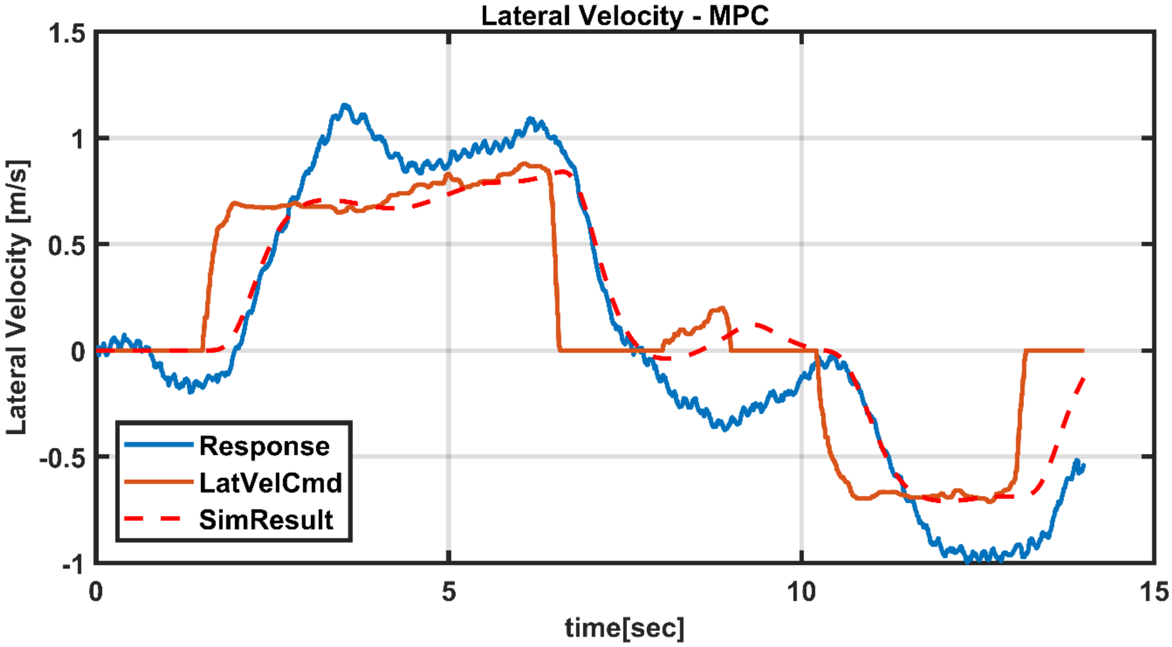


Figure 5-104 Flight test – Lateral velocity responses, MPC

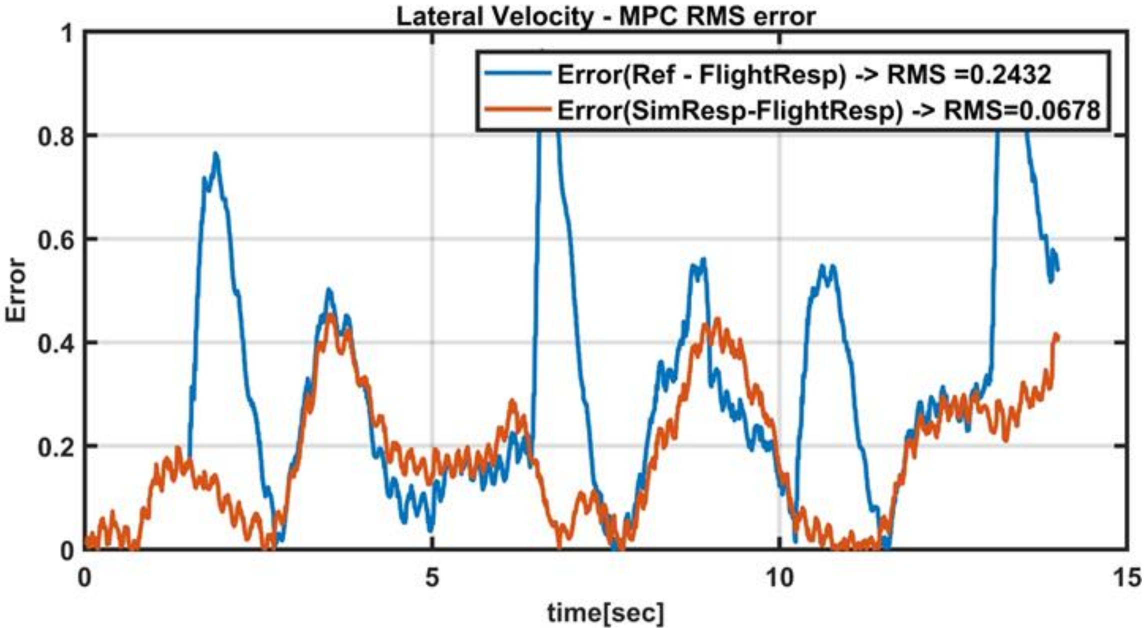


Figure 5-105 Lateral velocity responses RMS error, MPC

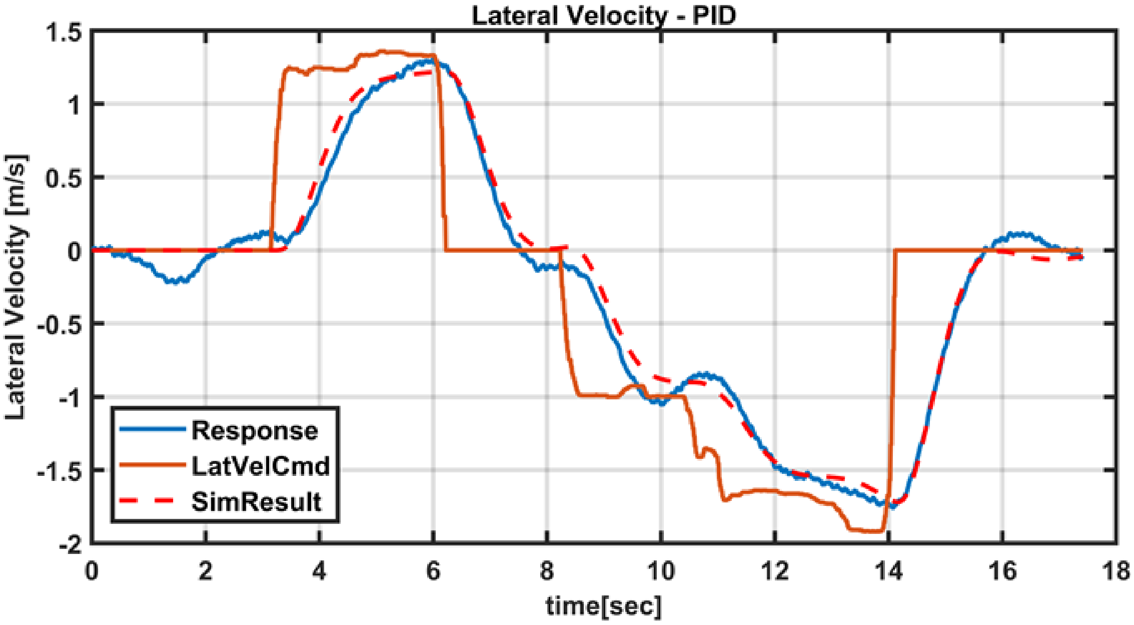


Figure 5-106 Lateral velocity responses, PID

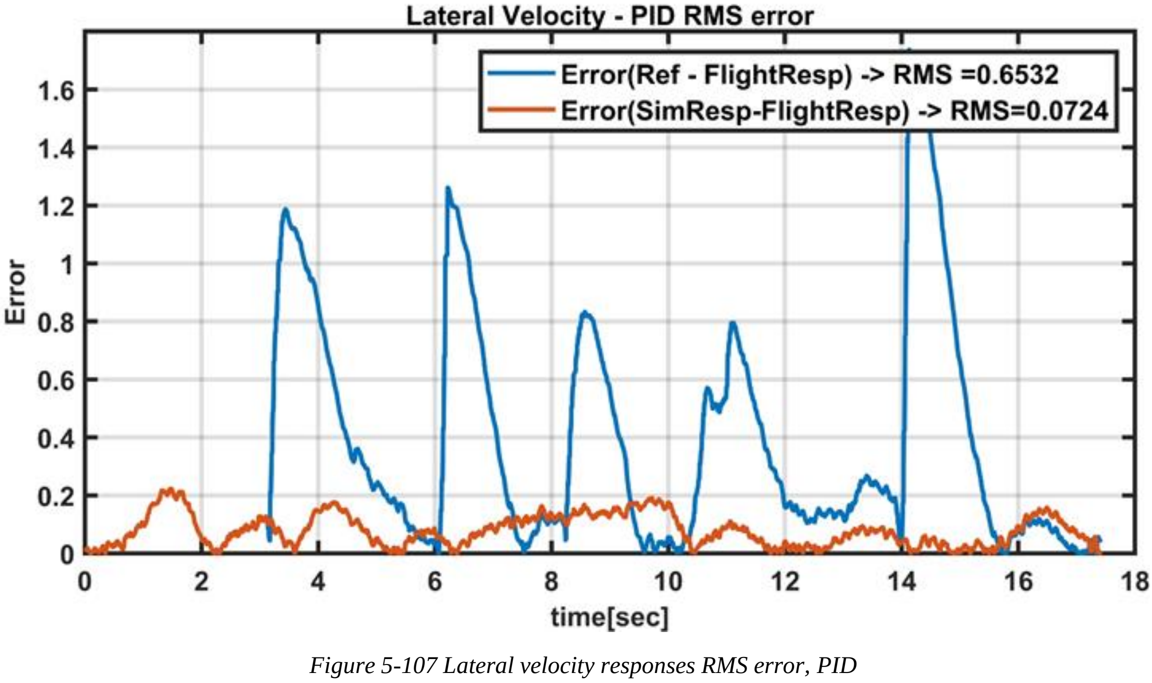


Figure 5-107 Lateral velocity responses RMS error, PID

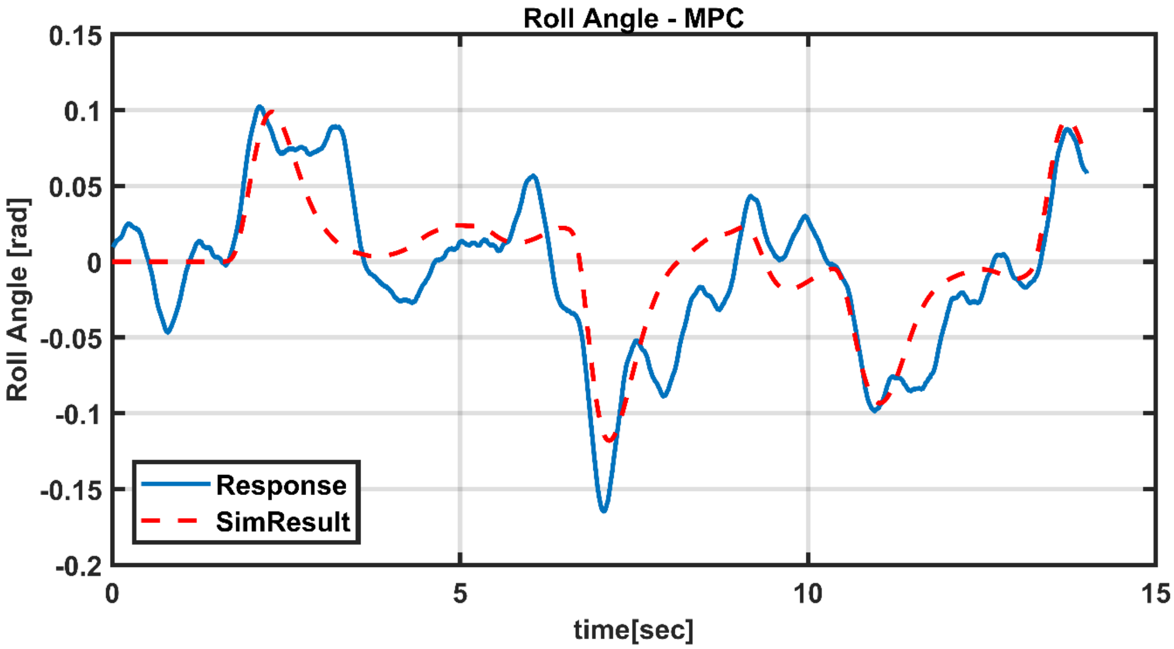


Figure 5-108 Flight test – Roll angle

The system responses in the lateral channel were more aggressive than the simulation results, though the identification was successful. But, looking at these channel responses, the decrease and increase were modest. However, in the longitudinal channel test, which has similar dynamics; the simulation and flight test are quite consistent. The mass increase along this axis may affect the performance. In Chapter 6, it is mentioned how these citations can be reduced as future work. PIDs simulation and flight test results were more consistent than the PID. It might show that PID is more robust against the model uncertainties.

5.4.1.3. Longitudinal Channel

Test Input:

To perform this test, ~0.8 m/s lateral velocity command has been applied during flight.

Responses:

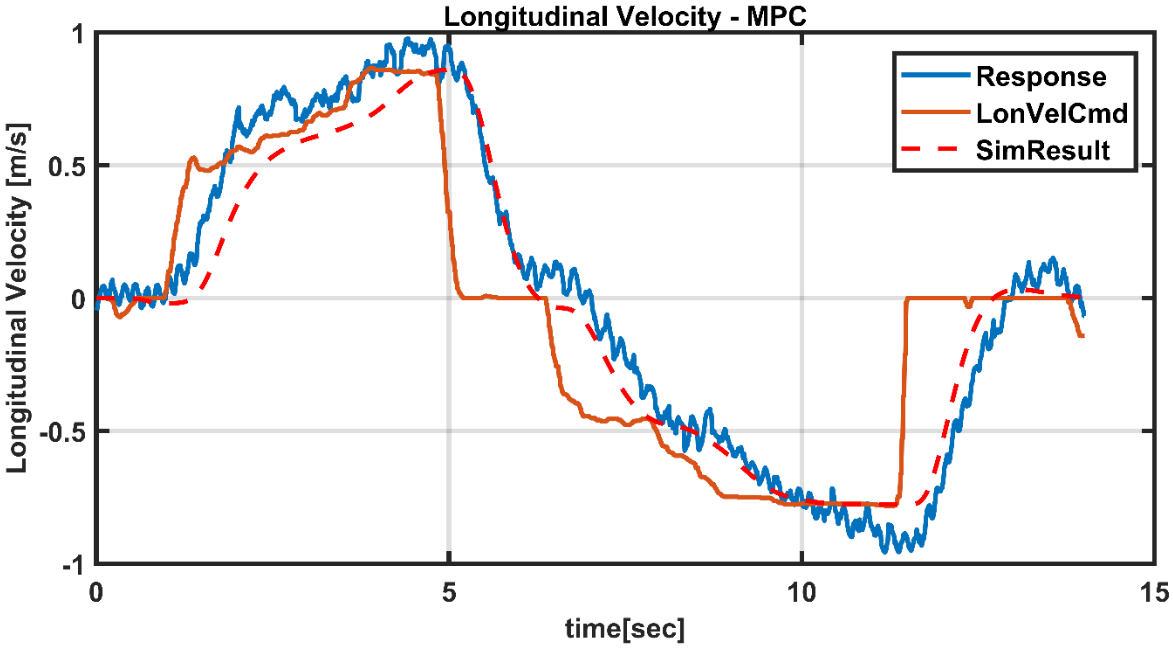


Figure 5-109 Flight test – Longitudinal velocity responses, MPC

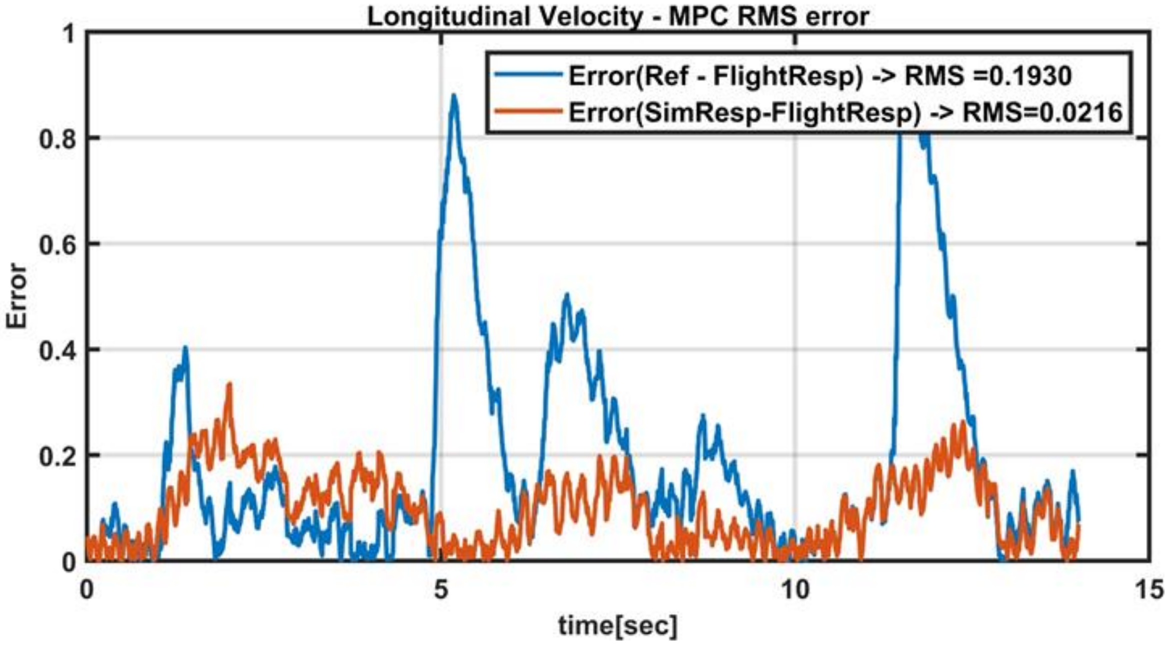


Figure 5-110 Longitudinal velocity responses RMS error, MPC

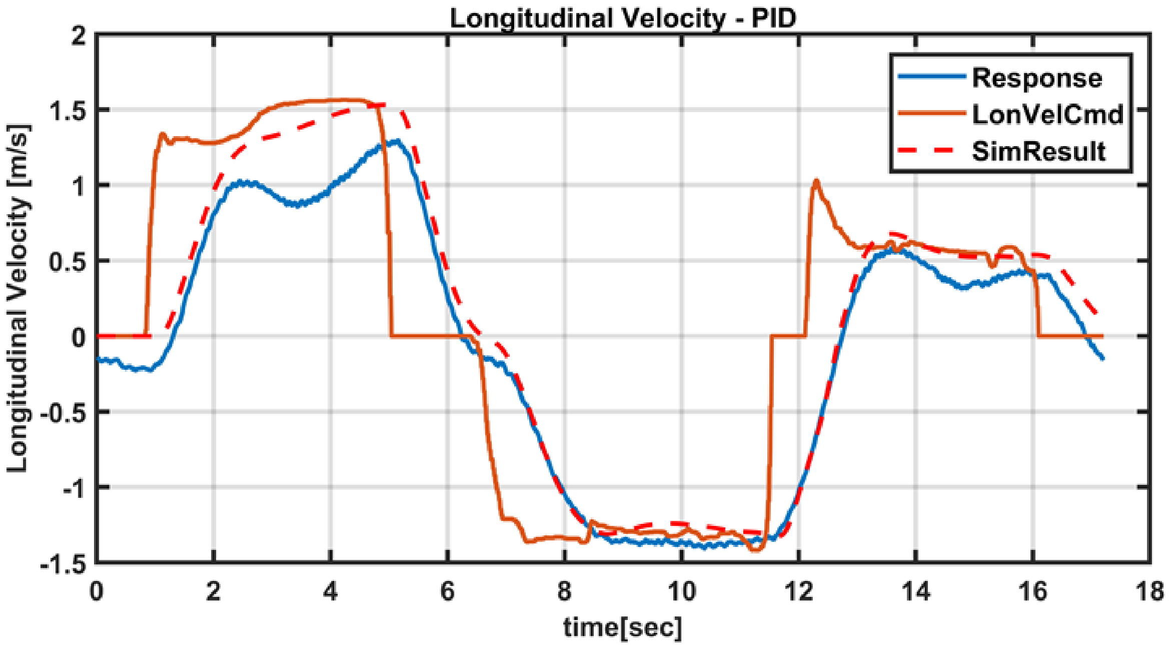


Figure 5-111 Longitudinal velocity responses, PID

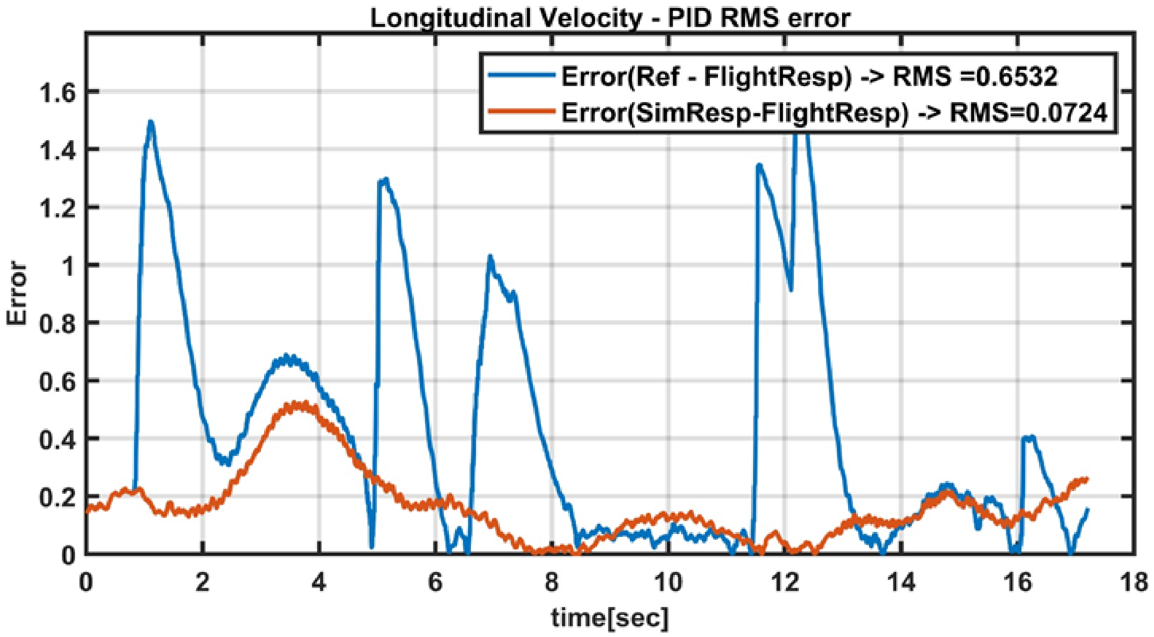


Figure 5-112 Longitudinal velocity responses RMS error, PID

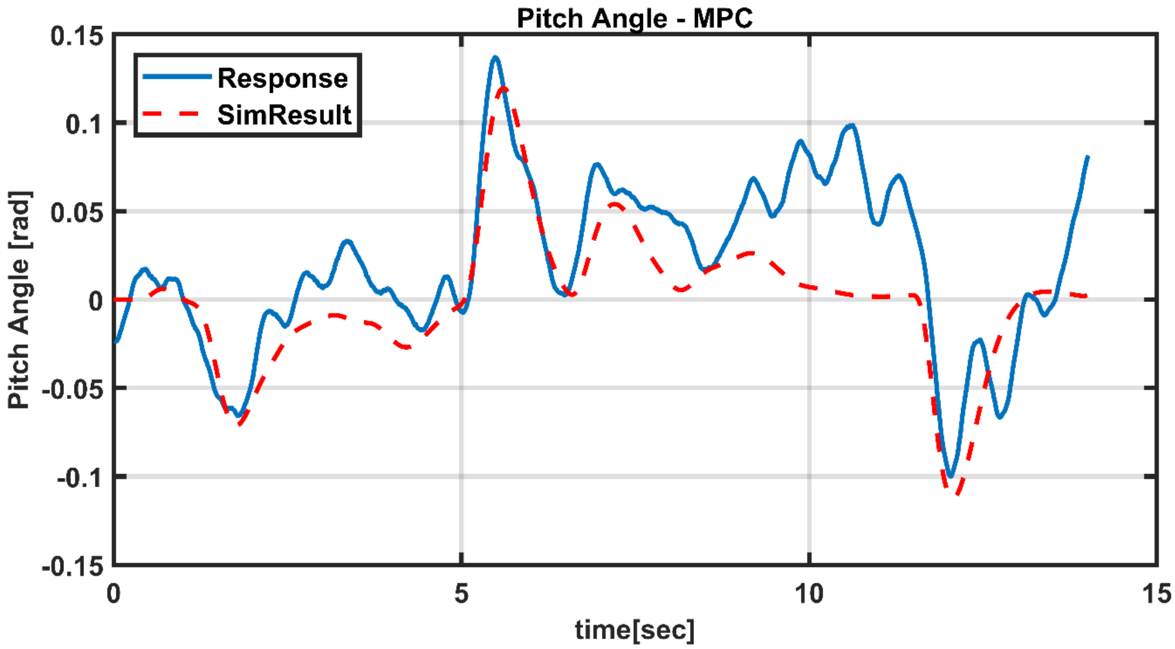


Figure 5-113 Flight test – Pitch angle response

The channel with the most satisfactory results during flight tests is the longitudinal channel. Also, this channel performed better than PID during tests.

5.4.1.4. Vertical Channel

In the longitudinal channel, the system input is given as angle command. Test input is shown in Figure 5-47.

Test Input:

To perform this test, ~0.8 m/s lateral velocity command has been applied during flight.

Responses:

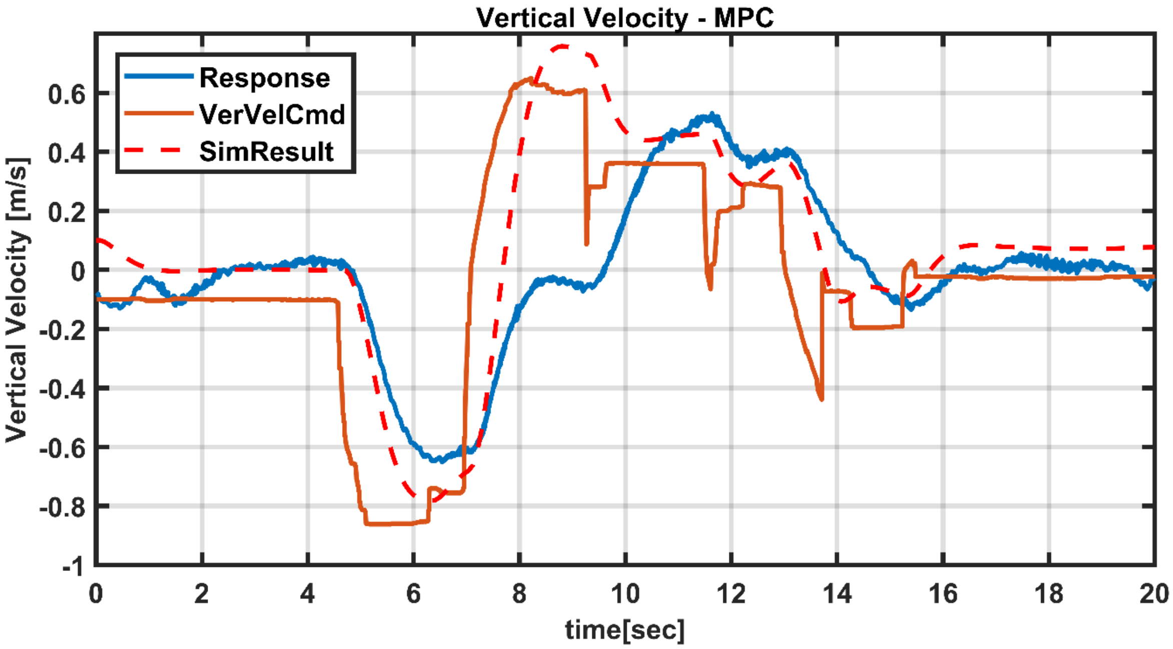


Figure 5-114 Flight test – Vertical velocity responses, MPC

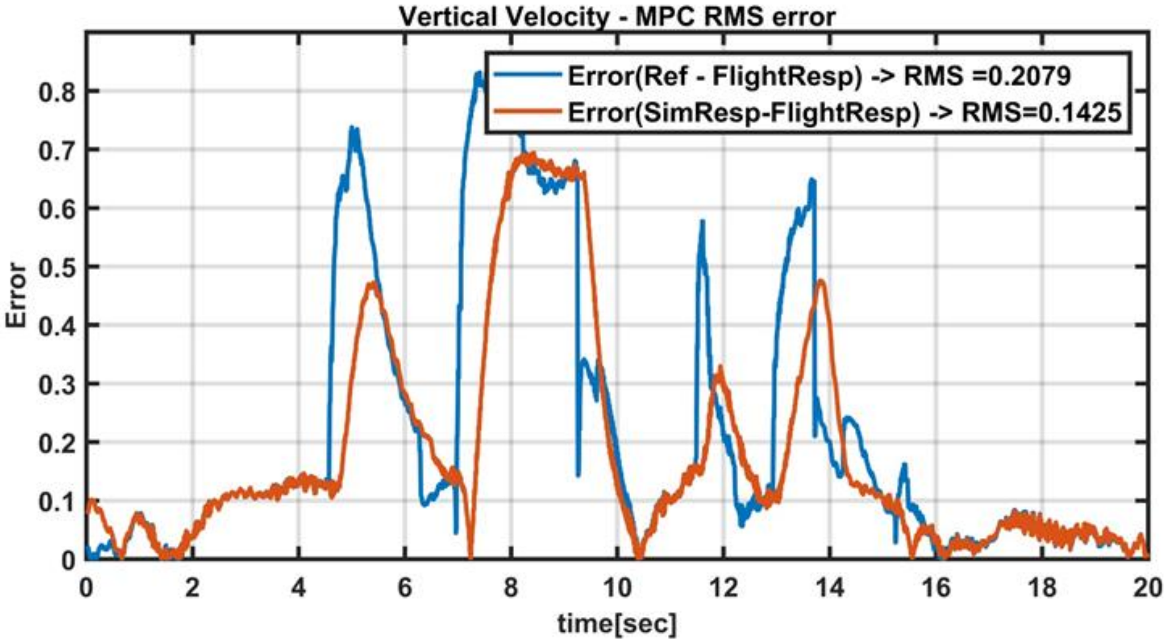


Figure 5-115 Vertical velocity responses RMS error, MPC

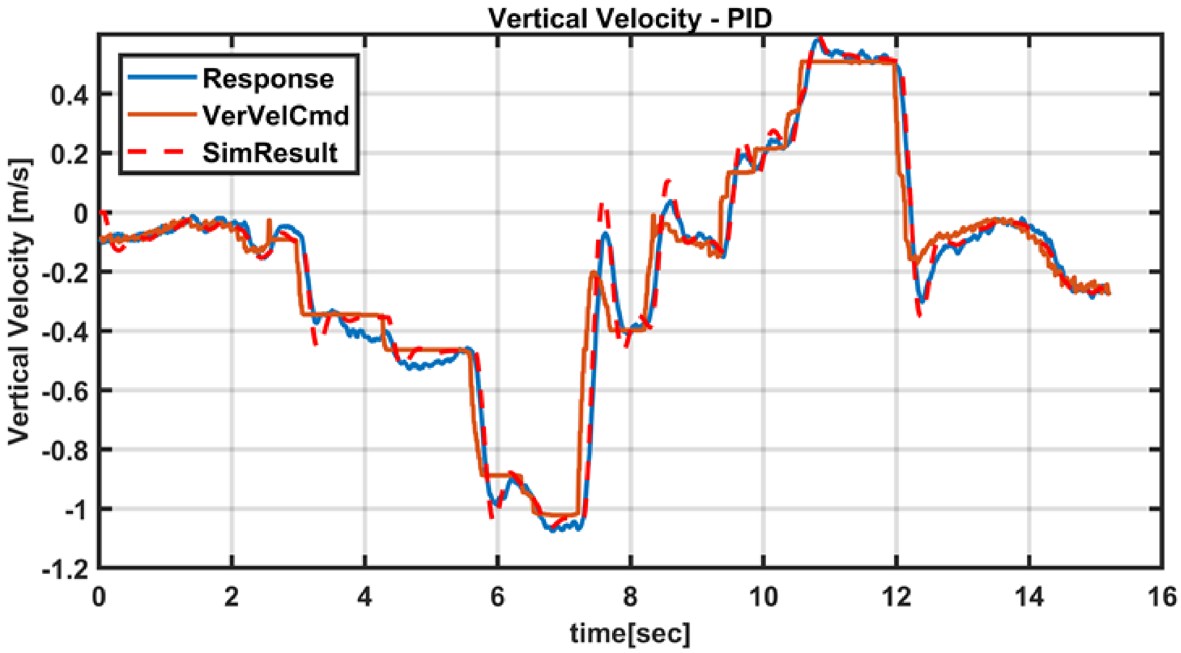


Figure 5-116 Vertical velocity responses, PID

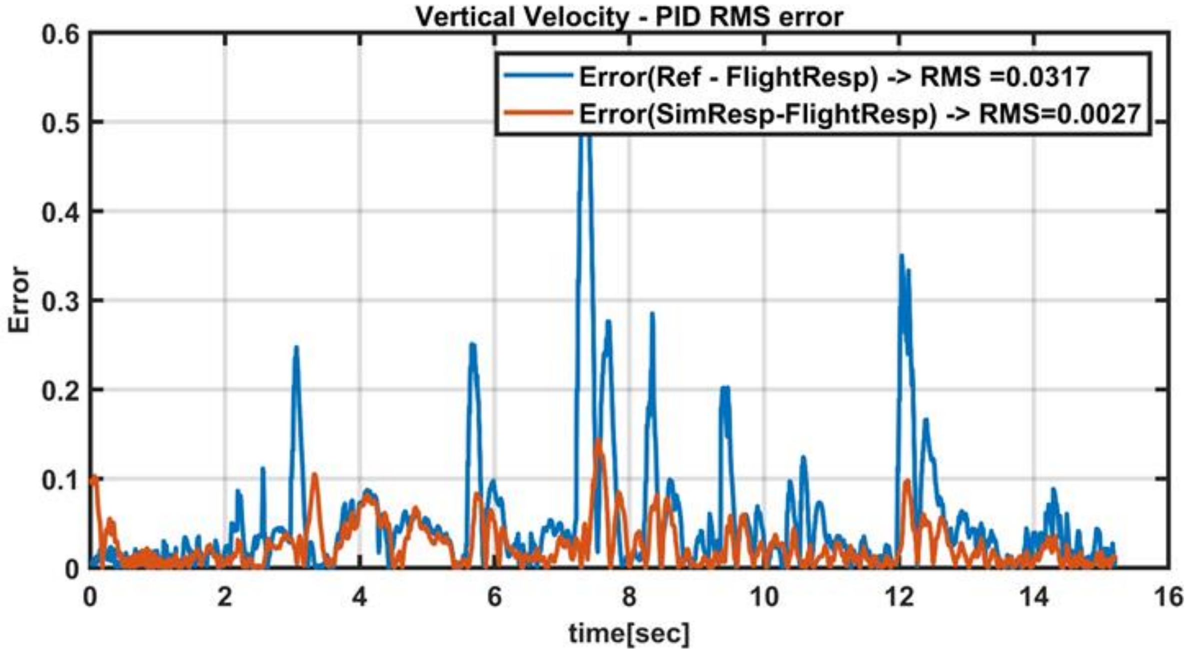


Figure 5-117 Vertical velocity responses RMS error, PID

In the vertical channel, system reactions were able to perform their movements with low acceleration due to the weight added on the system after identification, but still showed the adequate performance. PID was still given a consistent result with simulation.

6. CONCLUSION AND FUTURE WORKS

6.1. Conclusion

In this part of the thesis, the studies and results during the thesis study are briefly mentioned. As cited in the 1st chapter, system identification tests were carried out in order to obtain the system model. The system model obtained in these identification studies is shown in equation 3.26. The details of the results obtained during the system identification process are explained in detail under Chapter 3. The relevant graphics of the verification tests of the system model obtained are also included in this section.

Controller development work, which is the main purpose of the thesis study, has been covered in Chapter 4. The studies conducted here were first tested for different conditions in the simulation environment, and then SIL, HIL and flight tests were performed individually. During the controller design, the control structure was first decided. Then, determined in which flight mode, which states would be used, and which would have zero weights. Accordingly, the weights of the longitudinal, lateral and vertical speeds are zeroed for the attitude mode, and the weights of the pitch, roll and yaw angles of the system are zeroed in the speed mode. Subsequently, the prediction horizon, control horizon and sampling time suitable for the system dynamics were selected. Finally, fine-tuning was performed using the MPC designer to achieve the expected performance of the system. Performance comparisons with the reference controller, PID, are also made under this section. Finally, when the system performance was deemed sufficient, the code was generated, and SIL and HIL tests were passed.

In the SIL test, the accuracy of the code was tested, and it was confirmed that very similar results could be obtained with simulation. An important part of this phase is that the code could be correctly implemented into the existing autopilot code. Thus, it has been verified that the code will not cause a software problem before proceeding to the next stage.

In the HIL test, the computational performance was examined by embedding the generated code on the hardware. During the HIL test, the STM32F04, which was the first card to be tested, could not give sufficient performance. After that, the Jetson NANO board was used, and sufficient execution time was obtained. At this point, the remaining tasks of the autopilot software were left on the STM32F04, and only the MPC was moved to the Jetson NANO. However, making the communication between these two boards over the UART communication protocol added an extra 10-20 ms delay to the system since its an asynchronous protocol. It was revealed during HIL tests that these delays should be entered into the model in the MPC, and the average communication and calculation delays were included in the model. Then, when it was seen that sufficient performance could be achieved, the flight test has begun.

During flight tests, firstly, attitude mode was preferred, in which the system responses can be understood easier. In this mode, the vertical channel of the system is directly left to the pilot control. In these tests, the first controllers, which were designed primarily, produced very aggressive commands, and the platform could not take off safely. After examining the data, it was seen that the system oscillated with a sine wave around 1 - 1.5 Hz. Responses in similar frequency ranges could also be obtained in the simulation environment with different time delays. However, as mentioned before, the uncertainty of this delay on the system prevents it from being properly inserted into the model. In order to solve this problem, firstly the weight of the rate states was increased, and the system was aimed to produce softer responses. After these changes, the system was able to fly successfully. Later, the system was tested in speed mode, and it was seen that it could successfully fly in this mode. Relevant results are given in Chapter 5.

6.2. Future Works

Studies have shown that MPC brings a flexible structure and is suitable for designing high-performance controllers. It is also clear that the performance of the model predictive controller developed with high accuracy dynamic models can be improved further. On the other hand, the disadvantages of MPC are decreasing day by day thanks to developing technology. For example, sensors with high sensitivity and accuracy have become smaller day by day, and thanks to these sensors that can operate at high frequencies, system identification operations have become possible even on small systems. Another advantage of the developing technology is that the hardware with high processing power is now very light and in small volumes, making MPC studies on small class UAVs convenient.

During the thesis study, the insufficiency of some test conditions and the asynchronous communication of the cards on the system with each other clearly affected the performance of the controller. It is seen that with the improvements to be made in the test environment, controller performance can improve, and the consistency of flight tests and simulations will increase.

In the thesis study, obtaining the dynamic model covering dynamics around the hover showed the improvement in the controller's performance in this equilibrium position. System dynamics in forward flight and different trim values have not been examined in this thesis. With the models obtained by repeating the system identification processes in different flight conditions, adaptive MPC design can be made to improve the controller performance. Another similar method is to use Non-Linear MPC or LBMPC to ensure that the system performs well in dynamics outside the hover.

In a similar study [8], Zurich University study compared non-linear and linear MPC controllers. However, the models obtained here are obtained by using the flight dynamics equations. As Tischler has mentioned [1], these equations are actually not accurate enough for small class UAVs. Similarly, this test can be repeated between the MPC using the linear model obtained by the system identification process and the Non-linear MPC to contribute to the literature.

7. References

[1] M. Tischler, Aircraft and Rotorcraft System Identification: Engineering Methods for UAV Applications, (**2018**).

[2] A. Bemporad, Model Predictive Control - Lecture 1, (**2020**).

[3] C.A.P. A. Bemporad, C.Rocchi, Hierarchical and hybrid model predictive control of quadcopter air vehicles, (**2009**).

[4] K. Alexis, C. Papachristos, G. Nikolakopoulos, A. Tzes, Model predictive quadrotor indoor position control, IEEE.

[5] P. Bouffard, A. Aswani, C. Tomlin, Learning-based model predictive control on a quadrotor: Onboard implementation and experimental results, IEEE, **2012**.

[6] Y. Wang, A. Ramirez-Jaime, F. Xu, V. Puig, Nonlinear Model Predictive Control with Constraint Satisfactions for a Quadcopter, Journal of Physics: Conference Series, 783 (**2017**) 012025.

[7] C.A. Amadi, Design and Implementation of Model Predictive Control on Pixhawk Flight Controller, (**2018**).

[8] M. Kamel, M. Burri, R. Siegwart, Linear vs Nonlinear MPC for Trajectory Tracking Applied to Rotary Wing Micro Aerial Vehicles, IFAC-PapersOnLine, 50 (**2017**) 3463-3469.

[9] A. Aswani, H. Gonzalez, S.S. Sastry, C. Tomlin, Provably safe and robust learning-based model predictive control, Automatica, 49 (**2013**) 1216-1226.

[10] A. Loquercio, A.I. Maqueda, C.R. Del-Blanco, D. Scaramuzza, DroNet: Learning to Fly by Driving, IEEE Robotics and Automation Letters, 3 (**2018**) 1088-1095.

[11] T.F.E. Dale E. Seborg, and Duncan A. Mellichamp, Process Dynamics and Control, (**2011**) 414-438.

[12] L. Wang, Model Predictive Control System Design and Implementation Using MATLAB®, (**2009**).

[13] S.W. Jorge Nocedal, Numerical Optimization, Springer (**2006**).

[14] U. Ahmad, M. Ahsan, A.I. Qazi, M.A. Choudhry, Ieee, Modeling of Lateral Dynamics of a UAV using System Identification Approach, **2015**.

[15] D.M. Filatov, A.V. Devyatkin, A.I. Friedrich, Quadrotor parameters identification and control system design, IEEE.

[16] Y. Nong, Z. Qi, D. Lin, Ieee, System Identification of A Small Unmanned Aerial Vehicle Based on Time and Frequency Domain Technologies, **2011**.

[17] S. Sakulthong, S. Tantrairatn, W. Saengphet, Frequency Response System Identification and Flight Controller Tuning for Quadcopter UAV, IEEE.

[18] K.C. Wei Wei, Mark B. Tischler System identification and controller optimization of a quadrotor UAV, (**2015**).

[19] M.B.T.a.R.K. Remple, Aircraft and Rotorcraft System Identification - Engineering Methods With Flight-Test Examples, (**2013**).

[20] C. CORP, CIFER Software Users Manuel, (**2005**).

[21] R. Beard, UAV Coordinate Frames and Rigid Body Dynamics, (**2004**).

[22] N.L.R. Alberto Bemporad, Manfred Morari, Model Predictive Control Toolbox™ - User Guide, (**2019**).

[23] N.L.R. Alberto Bemporad, Manfred Morari, Model Predictive Control Toolbox™, (**2020**).