Robust Optimal Controller Designs for Attitude and Altitude Control of a Quadrotor

Joseph Thomas

School of Engineering
February 2021

Supervised by:
Assoc. Professor David Wilson
Dr. Jonathan Currie

A thesis submitted to Auckland University of Technology in fulfilment of the requirements for the degree of Doctor of Philosophy.

Abstract

The Unmanned Aerial Vehicle (UAV) industry has seen a huge growth in the past couple of decades, driven mainly by availability of faster and affordable microcontrollers and sensors. The commercial and hobbyist sectors have been increasingly using rotorcraft UAVs for new and varied applications, all the way from asset inspection to bait deployment in long-line fishing. This has required control engineers to come up with systems that are extremely robust, and optimal so that the varying nature of payloads and the craft parameters itself does not affect the overall performance of the drone, while the best flight quality and maximum flight times are achieved.

In this research the overall aim was to develop tracking attitude/altitude robust, and optimal controllers for a quadrotor UAV where non-trivial model uncertainty is present. To achieve this goal, a comprehensive framework that would develop optimal weight designs along with the robust controller was developed. This was achieved by enclosing the $\mathcal{H}_\infty$ problem inside a constrained non-linear optimization problem. Separate algorithms were proposed for Single Input Single Output (SISO) and Multi Input Multi Output (MIMO) systems for all three main variants of robust controller design strategies; namely the Mixed Sensitivity Optimization (MSO), Loop Shaping Design Procedure (LSDP) 1 & 2 degree of freedoms, and μ-synthesis. A detailed linearized model of the multivariable plant and a decoupled version of it was developed in addition to which, worst-case plant models with significant model uncertainty were constructed.

In the SISO case, MSO, LSDP 1 & 2 DOF, and μ- synthesis based robust controllers were developed using the proposed algorithms for the decoupled plant. Satisfactory performances were obtained in all the cases with the controller achieving robust stability consistently. Performance comparisons were conducted among the developed controllers and benchmarked against a PID controlled system. For the systems containing model uncertainty the robust controllers performed better than that of the PID controlled, in terms of meeting the designs specifications, and the LSDP based controller performed the best among the robust controllers.

Similarly for the multivariable case, MSO, LSDP 1 DOF and μ-synthesis based controllers were developed using the proposed algorithms. While the controllers assured robust stability, however, it was found that uncertainty in mass of the craft, and that of the thrust coefficient caused systems to fall short of providing guaranteed robust performances. The controllers were compared against one another, and it was once again found that LSDP based systems provided the best performance in terms of reference tracking, and achieving the required design specifications, while the μ controller was found to be the most conservative, with the MSO controller occupying a position between the two other controllers.

A simulation based case study was performed inspired by the quadrotor being used in long-line fishing application with its unique time varying nature of the payload mass and slung load length. Two controllers, a PID based and a MSO robust controller based system were put to test on a quadrotor model carrying a slug load. It was observed that the PID based controller performed better than the robust controller for systems that are close to the nominal model. But performance deteriorated rapidly as the plant moved away from the nominal model with several models with uncertainty becoming unstable. The robust controller maintained stability even when tested over extreme plants with the performance deterioration taking a less steeper path than its PID counterpart.

The proposed algorithms enabled an efficient design process of the optimal controller weights and significantly quickened the process of developing robust controllers for rotorcraft drones. The MATLAB framework involved in developing the algorithms are provided in this work. The controller comparison studies shed new insights into the overall process of selecting robust control strategies for varied and challenging UAV applications.

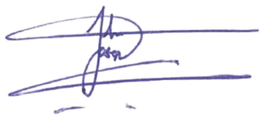
Contents

List of Figures

List of Tables

Attestation of Authorship

I hereby declare that this submission is my own work and that, to the best of my knowledge and belief, it contains no material previously published or written by another person (except where explicitly defined in the acknowledgements), nor material which to a substantial extent has been submitted for the award of any other degree or diploma of a university or other institution of higher learning.



Joseph Thomas ______________________________

Acknowledgements

I wish to express my gratitude to my academic supervisors Associate Professor David Wilson and Dr Jonathan Currie for their invaluable patience, support and guidance at every stage of this research. It is their constant encouragement and motivation that helped me complete this work.

My sincere thanks to the School of Engineering, Auckland University of Technology for providing me with a scholarship through the years. I would like to thank Industrial Information and Control Center for allocating funds to cover various aspects of this research. I thank all of the faculty from the School of Engineering and the lab technicians for their kindness and support.

I would like to thank Envirobotics Limited and especially Kyle Parshotam for his patience and support while helping me to learn more about industrial drone development. I would like to extend my thanks to some amazing people who helped me along the way– Professor Michael Petterson from the School of Science, Josephine Prasad from the Engineering Postgraduate School, Emiliana Faapoi from Graduate Research School, former employers Helen Page, Shama Sabir and Alex Shepherd all of whom constantly supported and motivated me through the work.

Finally I thank my family for supporting me, and my beloved partner Karen for her infinite patience and support through out this journey.

Nomenclature

Table 1: Thesis nomenclature

Symbol	Description
$\mathbf{x}$	State vector
$\mathbf{u}$	Input vector
$\mathbf{y}$	Output vector
x_i, y_i, z_i	Position in the inertial frame
ϕ_i, θ_i, ψ_i	Euler angles defining roll, pitch and yaw movements respectively in the inertial frame
x_b, y_b, z_b	Position in the body frame
ϕ_b, θ_b, ψ_b	Euler angles in the body frame
$\mathbf{R}$	Orthogonal rotation matrix
f_k	Force exerted by the k_{th} rotor
τ_k	Torque exerted by the k_{th} rotor on the craft
I_{xx}, I_{yy}, I_{zz}	Quadrotor moment of inertia along body x, y and z axis
$\mathbf{g_a}$	Gyroscopic moment vector
m	Mass of the quadrotor
c_t	Thrust coefficient
c_p	Power coefficient
ω	Angular velocity of a propeller
d	Diameter of a propeller
d_r	Distance of the centre of gravity of the quadrotor to the centre of gravity of the rotor
ρ	Density of air
$\boldsymbol{\vartheta}$	Generic parameter vector

The thesis will use the following notation for mathematical equations.

Table 2: Mathematical notation

Notation	Description
$\mathbf{X}$	Matrix (bold and capitalized)
$\mathbf{x}$	Column vector (bold, lower case)
x	Scalar (italic, lower case)
$\dot{x}$	First derivative of x with respect to time
$\ddot{x}$	Second derivative of x with respect to time
$\\|x\\|_\infty$	The infinity norm of x
$\sigma(\mathbf{X})$	Principal gain (singular value) of $\mathbf{X}$
$\bar{\sigma}(\mathbf{X})$	Maximum singular value of $\mathbf{X}$
$\underline{\sigma}(\mathbf{X})$	Minimum singular value of $\mathbf{X}$
Δ	Model Uncertainty
$\mu(\mathbf{X})$	Structured singular value of $\mathbf{X}$

In addition to the above the thesis will use the following acronyms.

Table 3: List of acronyms

Abbreviation	Definition
UAV	Unmanned Aerial Vehicle
PID	Proportional Integral Derivative
RCT	Robust Control Theory
MSO	Mixed Sensitivity Optimization
LSDP	Loop Shaping Design Procedure
PCL	Proportional Integral Derivative Controller based Loop
MCL	Mixed Sensitivity Optimization Controller based Loop
NS	Nominal Stability
NP	Nominal Performance
RS	Robust Stability
RP	Robust Performance
HJBI	Hamilton-Jacobi-Bellman-Isaacs
SISO	Single Input Single Output
MIMO	Multi Input Multi Output
NOMAD	Nonlinear Optimization using the Mesh Adaptive Direct Search Algorithm
LQR	Linear Quadratic Regulator
LQG	Linear Quadratic Gaussian
MPC	Model Predictive Control
NMPC	Nonlinear Model Predictive Control

Abbreviation	Definition
ANN	Artificial Neural Network
FC	Fuzzy Control
GA	Genetic Algorithm
AI	Artificial Intelligence
LFT	Linear Fractional Transformation
ARE	Algebraic Ricatti Equation
MAV	Micro Aerial Vehicle
LMI	Linear Matrix Inequality
EKF	Extended Kalman Filter
UKF	Unscented Kalman Filter
CIFER	Comprehensive Identification from Frequency Responses
MOI	Method Of Inequalities
MBP	Moving Boundaries Process
IAE	Integral Absolute Error
ITAE	Integral Time Absolute Error
ISE	Integral Square Error
ITSE	Integral Time Square Error
DAE	Differential Algebraic Equations
ODE	Ordinary Differential Equations
SVD	Singular Value Decomposition

List of Publications

1.) Thomas, J., Currie, J., and Wilson, D. "Automatic weight selection for mixed sensitivity optimization control of a quadrotor". In: 2017 24th International Conference on Mechatronics and Machine Vision in Practice (M2VIP). IEEE. 2017, pp. 1-6.

2.) Thomas, J., Currie, J., and Wilson, D. "Loop Shaping Design Procedure for Quadrotor Control with Weights Designed by Resolving a Constrained Nonlinear Optimization Problem". In: 2017 Australasian Conference on Robotics and Automation(ACRA). ARAA. 2017, pp. 1-7.

Chapter 1

Introduction

The popularity of rotorcraft drones with their superior agility and maneuvering abilities has risen in the past couple of decades among hobbyist communities as well as in the commercial sector. Unmanned Aerial Vehicles (UAVs) are increasingly being used in the commercial sector from monitoring and asset inspection applications [1–4], to search and rescue applications [5, 6], with the non-military drone market set to rise from its market value of 2.5 billion USD [7] in 2016 to an estimated 84 billion USD by 2025 [8].

In this context, stability, robustness, design optimality, and reliability have gained more importance with the manufacturers and research groups, than ever before. In this study, techniques are developed to design flight controllers for quadrotor UAVs that are both robust and optimal in its performance. This chapter provides a brief overview of the study in four sections. In the first section, an outline of the existing small UAV flight controller designs are presented. In section 1.2, the research objectives of the study are listed out. In section 1.3, the thesis contributions are discussed and in the final section a brief overview of the important chapters in the study are presented.

1.1 Background: Overview of Small Scale UAV Flight Controller Designs

Rotorcraft drones, classified in [9] as small scale, are those with a maximum payload capability of less than 10 kg and a flight time of less than 50 minutes. Commercial multi rotor drones generally fall in this category. Quadroters and other rotorcraft drones used in commercial applications mostly employ multiple PID based controllers [10], as they provide nominal stability and are easy to implement.

The stability and control of rotorcraft drones is a topic that has been researched extensively in the past couple of decades (see surveys [9–12]). Conventional control strategies such as the Proportional Integral Derivative controllers [13], adaptive controllers [14], Linear Quadratic Gaussian controllers [15], Model Predictive Control algorithms [16], as well as artificial intelligence based systems such as controllers based on Artificial Neural Networks [17], and Genetic Algorithm, [18] are among some of the favorite techniques that have been employed to find a solution to the quadrotor reference tracking problem.

Proportional Integral Derivative (PID) controllers, while they provide satisfactory nominal stability and performance, lack inherent optimality and robustness properties when significant model uncertainty is involved, as pointed out in [10]. Adaptive flight controllers while it is successful to a certain degree in tackling the problem of parametric model uncertainty, systems based on these have been observed to become unstable when the controllers are required to converge to accurate parameter estimates while tracking a constant steady state reference input as presented in [19]. The lack of guaranteed stability margins provided by Linear Quadratic Gaussian (LQG) regulators as noted in [20], is a trend reflected in studies that have developed LQG flight controllers, but then failed to have provided stability margins while considering model uncertainty. Model Predictive Control algorithms and techniques built on Artificial Intelligence (AI) based strategies, focus mostly on resolving the reference tracking/trajectory planning problem as compared to maintaining the attitude/altitude stability of the drone. The inability to guarantee satisfactory robust stability margins for plants with model uncertainty is seen to be a common characteristic of the strategies mentioned above.

Robust control strategies on the other hand, attempt to provide better stability and performance in the presence of model uncertainty, and have been employed in the development of quadroter UAV controllers (see [21–23]). Robust controllers are also used in combination with optimal control strategies, where the former tends to solve the 'tracking attitude and altitude' problem while the 'position tracking problem' is resolved by the latter [24–26]. The problem of multirotor model uncertainty has been addressed using the tools provided by linear robust control theory generally from three different angles, namely Mixed Sensitivity Optimization (MSO), Loop Shaping Design Procedure, (LSDP) and μ synthesis, with majority of the research undertaken using the Mixed Sensitivity Optimization technique.

Robust control theory, while offering enough leeway in terms of including model uncertainty in the design, brings in a development process that is quite tedious. A significant portion of the effort gets directed towards development of controller weights that play a vital role in the overall stability and performance of the system at the desired set-point. Even after such rigorous designs, the flight regimes that can be comfortably undertaken by these are found to be highly conservative reminding us the fact that important aspects of stability such as modelling high-frequency dynamics and dealing with non-deterministic elements such as wind, still remains an unsolved problem.

1.2 Research Objectives

Robust control theory applied in developing flight controllers for UAVs, generally follows the traditional method of loop shaping mostly based on of trial and error (see [21, 27, 28]) when it comes to selecting controller weights. This technique is both tedious and mostly results in sub-optimal controller designs.

Developing frameworks that involves the robust controller weight parameters, which minimize a cost function can be seen in various domains such as in [29–31] which employ fuzzy logic, particle swarm optimization, and genetic algorithm based optimization strategies respectively, although a general framework that can capture parametric uncertainty, uncertainties due to time delays or model non-linearity is largely left out.

This study aims to develop *optimal* and *robust* SISO and MIMO attitude-altitude rate tracking controllers for quadrotor UAVs with significant model uncertainty. To achieve this, comprehensive frameworks that would automate the weight selection procedure for the three main robust control techniques, will be developed. The following research objectives are examined and pursued in this study alongside the controller development.

1. To develop a framework which can take into account various linear constraints, such as limits on closed loop functionals as well as on the $\mathcal{H}_\infty$ problem, while automatically generating the controllers weights, alongside the process of developing an optimal and robust quadrotor controller for attitude and altitude rate tracking. Separate frameworks are to be developed for MSO, LSDP and μ -synthesis robust control design procedures.

2. To extend the functionality of the framework from single-input single-output systems (which is useful for a decoupled quadrotor system), into the *multivariable* domain and develop optimal robust attitude-altitude rate tracking Quadrotor controllers using the MSO, LSDP and μ-synthesis design procedures.

3. To conduct a simulation based comprehensive testing and performance com-

parison between the various developed controllers to determine the effectiveness of the design strategy.

The algorithms proposed, could be used effectively by the control engineer to come up with the most appropriate and optimal weights required by the controller. Methods to incorporate model parametric uncertainty and non-linearity into weight designs will be developed in the context of a quadrotor UAV.

The objectives are achieved by developing an outer framework over the conventional $\mathcal{H}_\infty$ cost function which solves the problem of weight selection by resolving a constrained non-linear optimization problem. The Nonlinear Optimization using the Mesh Adaptive Direct Search Algorithm (NOMAD) [32] is employed as the optimizer in the process.

1.3 Thesis contributions

The primary objective of this study is to develop optimal-robust tracking attitude/altitude rate controllers for quadrotors. To achieve this in line with the above identified research objectives, the following key contributions have been made:

1. Algorithms that enable the systematic development of the the optimal weights alongside the robust controller, have been developed for each of the three main branches of robust control; namely Mixed Sensitivity Optimization, Loop Shaping Design Procedure and μ-Synthesis for SISO systems.

2. The above algorithms and the framework have then been extended to the multivariable domain for the corresponding robust control branches.

3. The comparison of tracking quadrotor attitude/altitude rate controllers (and the associated weights), which were developed using these proposed algorithms, have been conducted and results presented.

4. Finally with the insights obtained from the various developed algorithms and adjoining comparison studies, a holistic framework that can be utilized by control engineers to develop optimal robust controllers is proposed.

To conduct the extensive testing that was required of the controllers, a new way to visualize the stochastic Monte-Carlo simulations is presented. Further, a non-trivial simulation based case study of the quadrotor carrying a slung load with time varying payload mass and slung length (inspired from a real-world fishing application) is discussed. For the case-study, a robust controller was developed and tested against the bench-mark cascaded PID based controllers, and results are presented.

The contributions of this study are summarized in section 7.2.

1.4 Thesis Outline

The thesis is divided into 6 subsequent chapters. In Chapter 2, the literature associated with previous research in the area of quadrotor flight controller development is examined. The flight controller designs developed using PID, adaptive, tracking LQR/LQG, model predictive control and artificial intelligence based controllers are examined alongside various non-linear and linear robust control strategies. Research work done previously on stability problems experienced by quadrotors are also examined here.

In chapter 3, the quadrotor problem statement of interest is examined. The quadrotor mathematical model and the model parameters used in this study are presented. Following this, the state space model and the methods of scaling are discussed. Further the sources of uncertainty considered, and the design specifications which form the physical constraints used in the robust controller designs are presented.

The Single Input Single Output (SISO) controller development and comparison studies are examined in Chapter 4. First the decoupled model of the quadrotor is developed and an introduction to robust control theory is presented in the chapter. Next, the frameworks for generating optimal control weights alongside the controller for Mixed Sensitivity Optimization, Loop Shaping Design Procedure, and μ-synthesis procedure are presented. Quadrotor controllers are also developed and the designs are analyzed in the chapter. Detailed comparisons are performed both among the developed controllers and also with a PID controlled system and conclusions are presented.

In chapter 5, the multivariable counter parts of the SISO controllers are developed. To design the optimal weights alongside the controllers, frameworks similar to those proposed in chapter 4 are developed. Multivariable controllers are designed for MSO, LSDP and μ-synthesis and a performance comparison is conducted among the controllers.

In chapter 6 a simulation based case study, based on aspects of an industrial application is presented. The developed control strategy is applied on an real world example inspired from a long-line fishing solution developed by Envirobotics Limited, a New Zealand based company that develops UAVs for industrial applications.

A comparison study is performed with a cascaded PID controller based system and conclusions are drawn.

In the final chapter a summary of the study and research findings are presented. The main conclusions drawn from the study and the future work is also discussed.

Chapter 2

Literature Review of Controller Designs

The chapter presents a literature review of previous research work performed in the field of developing flight controllers for small and micro aerial vehicles. In the first section, we look at the controller design developments in various branches of control literature and why these designs are inadequate for the problems of interest. In section 2.2 the need for robust control alongside the studies that have used robust control techniques to develop flight controllers are examined. In the section 2.3, some of the general problems affecting the stability of UAVs and the ways of resolving them are discussed. This is followed by the final section which lists the general conclusions drawn from the literature review.

2.1 UAV Flight Controller Designs

Non-linearity, modeling uncertainty and dynamics coupling are some of the important challenges that make the problem of designing a flight controller for a quadrotor UAV, challenging [33]. A number of different flight controller designs have been proposed in various studies to tackle different aspects of these general challenges. Cascaded PID controllers used mostly in commercial UAV flight controller designs, adaptive controllers, optimal control strategies like LQR-LQG designs, artificial intelligence based and robust controller designs are among the various techniques that have been proposed. The flight controllers designed using these strategies are examined in sections 2.1.1 - 2.2.

2.1.1 Proportional Integral Derivative Control

PID based controllers, although they lack optimal properties and doesn't offer robustness as such, are easy to implement on small platforms (such as on easily accessible STM32 family of micro-controllers [34]) making it the favourite choice for commercial autopilot manufacturers [10]. Typical PID flight controllers which achieves attitude and altitude stabilization are presented in [13, 35–38]. The controller constants are tuned for stability and optimal properties, and the controllers offer nominal stability.

Comparison studies [39–41], look at the PID and LQR controllers that provides attitude as well as the position stability for the UAV. In general, PID controllers provides better stability for the nominal plant, in terms of moving closed loop poles further into the left half plane, while better degree of optimality comes with LQR based controllers. A comparison of the path tracking ability of Generalized Predictive Control (GPC) and the PID method is presented in [42]. The cascade structure receives roll and pitch input coordinates which it translates to X and Y position coordinates. For this the system identification is performed using Matlab System ID toolbox. The PID controller achieves better robustness against model imperfections with respect to GPC controller.

Automatic tuning of PD controllers based on a scheme using an adaptive technique is presented in [43], the method being an improvement of [44]. The controller stabilizes the altitude (resembling a critically damped second order system), although tracking abilities in presence of disturbances haven't been discussed in the work. Automatic PID tuning using adaptive pole placement for attitude stability is studied in [45]. The author concludes that the time taken by the system identification mechanism adapting to different environments would inevitably lead to instability. A cascaded PID controller is discussed in [46], where a hierarchical order is followed at three levels of control, resulting in a highly conservative path tracking control algorithm.

Moving on to some of the hybrid-controllers we can see the actuator fault tolerant stability studied in [47] using a PID-LQR technique where we have the optimal properties of LQR, incorporated into the PID constants of the UAV controller. This is achieved by modifying the state space dynamics as the fault occurs by adjusting the 'fault' vector which represents the control effectiveness of the actuator.

PID controllers provide nominal stability and performance in general, and attempts such as [40] looks into disturbance rejection techniques, but a comprehensive control framework which takes into account parameter uncertainty, optimality, tracking control in the presence of varying frequency and higher order magnitude disturbances in the context of a highly coupled system like a quadrotor, seem to be beyond the scope of PID control.

2.1.2 Adaptive Control

As the name suggests the adaptive controller adapts to changes in the system or the surroundings, with the technique being employed when the control engineer has a highly non-linear or time varying system to stabilize. There would usually be two parts for such a control problem, an estimation problem, and controller design problem. There have been several variants for adaptive control techniques alongside the popular adaptive PID tuning, adaptive pole placement and minimum variance algorithms. Because these can be designed to adapt to parametric uncertainties such as say inertial matrix parameters for UAV flight controllers (as demonstrated in [14, 19]), the term 'robustness' has been sometimes used with adaptive controllers in literature although studies on certifiable stability margins for adaptive control is an ongoing field for research [48].

A robust version of adaptive control is presented in [49], where sliding mode control maintains position tracking and the adaptive control maintain attitude tracking. The parameter drift is reduced by using a function that contain noise signals derived from a reference model. Attitude stability realization during events of partial actuator failure is being discussed in [50]. A similar study which presents a robust adaptive controller that could deal with payload mass uncertainties in [19] points us to the so called 'dichotomy of adaptive control' (see [51] Chapter 7). The author notes that when the reference output was constant the controller found it impossible to converge to a parameter estimate although successful trajectory tracking was achieved.

In general in adaptive control, good estimation requires varying inputs and outputs although that conflicts with the general UAV requirement of a steady flight. The inherent ineffectiveness of adaptive control in the presence of un-modeled dynamics is explained well in [52].

2.1.3 Tracking LQR and LQG

The Linear Quadratic Regulator (LQR) and the Linear Quadratic Gaussian (LQG) are optimal controllers based on linear plant models with quadratic performance objectives [51]. The latter combines the former with a state estimator (presumably the 'Kalman Filter' (KF) or other forms of it like the Extended KF) to bring about a realistic controller design. Although optimal, the controllers designed using these techniques are proportional controllers hence in-order to compensate for the steady state error 'integral states' (such as the integral of the difference between the reference input and plant output) will need to be introduced to obtain tracking LQR and LQG controllers.

Flight controller designs for UAVs under the LQR and LQG framework have been intensely researched with several proposed design strategies examined. A double hierarchical control algorithm which uses a relatively accurate model of the quadrotor is presented in [53]. The settling time limits for the outer position tracking control loop is relaxed and a faster inner loop controller is used. These enable the system to minimize energy consumption similar to the time varying tracking LQ controller described in [54]. Path tracking problems are further resolved in [55–57] using LQR and its variants and in [58, 59] using LQG controllers. Other studies such as LQR integral states based (double layer) controller offering attitude stability [60], a switching PID-LQR hybrid controller (switches between 20 different models) [61] for a F-16 , comparison study between UKF and EKF being the state observer for the LQG controller [62] and a UAV modelled as a tail-sitter using LQG control theory [63], provide interesting insights into the capability and robustness margins that these optimal controllers could guarantee.

Reflecting on Doyle's 1978 article [20] (in which regarding the guaranteed stability margins offered by an LQG regulator, he famously concluded in a terse abstract that '*there are none*'), although the studies have discussed optimal properties of the algorithms they have failed to present a thorough robust stability and performance analysis for models containing plant uncertainty and this becomes one of the main sources of unreliability with regards to selecting LQR/LQG for quadrotor controller designs.

2.1.4 Model Predictive Controller (MPC)

Model Predictive Controller is an optimal controller that uses a cost function that accounts for both the error between the reference signal and model prediction over a 'prediction horizon', as well as a weighting on the usage of inputs over a 'control horizon'. The designed controller will be used for a single time step, the horizons would then recede and the whole process would be repeated, making the control algorithm computationally expensive especially when there are constrains involved with the decision variables.

MPC controllers are widely used for quadrotor position tracking control such as those seen in [64–67] and in limited studies where it has been employed for attitude tracking such as [68, 69]. MPC is one of the popular choices for control engineers when a hybrid controller is proposed to resolve the flight control problem of a UAV, with MPCs usually forming the outer loop to tackle the position-tracking problem as see in [70].

There have also been MPC applictions such as in [71] where a switching scheme is employed between five linearized plant models based on the performance requirement for trajectory tracking, and in others such as the obstacle avoidance scheme proposed in [72] and fault tolerant MPC that works with even two partially failed actuators [73].

Nonlinear MPCs (NMPCs) cater for highly non-linear dynamic applications where linearizing the model leads to significant plant-model mismatch. Two interesting studies, the first being a hybrid PID-NMPC controller is presented in [16] and the second being a comparison study between the LQR,MPC and NMPC [74], gives us valuable insights into capabilities of MPC. Both studies point to the fact that MPCs generally are poor performers when it comes to disturbance attenuation for attitude/altitude tracking, and hence limiting its influence to mainly position tracking controllers.

From the above studies we infer that, with the computational speed limits restricting the implementation capability of the controller, they are used widely as position tracking controllers and are not quite favoured when it comes to attitude-altitude tracking applications.

2.1.5 Artificial Intelligence Based : Artificial Neural Network (ANN), Fuzzy Control (FC) and Genetic Algorithm (GA)

Artificial Neural Networks can be viewed as a collection of artificial neurons with their functionality depending on the way they form connections. In the past decade ANN has been widely used as an autonomous trajectory solver for ground based applications [75, 76] and the trend has moved on to aerial applications. A Neural network with three specific areas the sensory, motor and a controller network built exclusively for trajectory tracking is presented in [77]. A similar study that presents a Dynamic NN, which comprises of a Recurrent NN and two feedforward NN can be seen in [78], the trajectory tracking in both studies being dependent on a recursive reward based learning strategy.

Fuzzy Control is a relatively new control approach which started becoming popular in 1980s. The approach provides a method to represent and implement commonsense heuristic knowledge about controlling a system, into a formal control framework ([79] see Chapter 1). The subject has been gaining popularity in UAV control in the past decade. A typical FC based trajectory tracking controller is presented in [80]. The 'computed torque controller' is compensated for uncertainties in the model by the fuzzy logic compensator. Simulation based test results show the quadrotor being robust against parametric as well as wind disturbances. A study which presents the implementation of the fuzzy controller for the quadrotor is presented in [81]. The controller is similar to the previous study with a PD Fuzzy controller combined with the traditional Integral controller. There are several variants for the fuzzy based controllers such as - [82] where an EKF adjusts the membership functions and the LQR controller is tuned based on that, or as in [83] where the fuzzy controller betters the sliding mode controller performance for the quadrotor UAV, or the multi-layered fuzzy controller the solves the stabilization problem of a hexacopter with a hanging payload [84].

A Genetic Algorithm is a global search algorithm introduced by Holland in 1975 [85] that uses an optimization technique where the candidate solutions are evolved and those with the best fitness survive forming the next generation of solutions. GA techniques have mainly been used for trajectory identification of the UAV (rather than stabilization) and some of these studies for optimal path determination are presented in [86–89]. With given obstacles to avoid, the algorithm attempts to find the global optimum. For the case of UAVs, the algorithms would work its way through 3-Dimensional environments, with obstacles partially known prior to the vehicle flight.

Apart from these AI based algorithms there have been other techniques, mostly variants of the above, that have been used for UAV flight control. However, similar to the mentioned techniques, i.e., ANN based methods which are used primarily for trajectory tracking, FC which is mainly used to complement the main controller and GA which are usually used mainly for trajectory planning, these AI based techniques fail to certify the robustness for a UAV with model uncertainty. Robustness properties have been analysed by random tests rather than theoretically, hence lacking the verifiability offered by Robust control based methods, which leads to the final part of flight control literature survey.

2.1.6 Summary of conventional control schemes

It was observed that conventional control schemes provide nominal stability for quadrotors and some studies have examined the robustness properties that these controllers can impart to the system. In general, the control schemes have failed to explicitly take into account model uncertainty, issues of non-linearity, and time delays during the design stage of the controller development. There have been attempts made to address the problem of model uncertainty in certain fields like adaptive controllers where the controller adapts towards changing model parameters, or through real-time parameter tuning for PID controllers or with the help of switching mechanisms when in comes to MPC controllers. The problem of model uncertainty have also been examined in studies based on artificial intelligence based controllers where results based on tests conducted have been used to measure robustness of the controller.

Lack of a systematic approach to deal with the model uncertainty remain an overall issue in all of the above discussed studies. In the next section we look at the need for robust control and at studies that have explicitly addressed the issue of model uncertainty for the case of the quadrotor using robust control techniques.

2.2 The Need for Robust Control

Robust Control Theory (RCT) began gaining more interest in the control domain during the 1970's as control engineers began to find that the classic theories of optimal control from the 1960's, failed to account for uncertainties in the plant model [90]. Robust control theory explicitly takes into account model uncertainty during the controller design and thus directly attempts to increase the robustness of the closed loop system. RCT not only provides tools to calculate the robust stability margins such as that of the μ-synthesis, but also design techniques that directly consider these stability margins and work towards increasing the maximum amount of model uncertainty the controller can tolerate.

The method of development can be broadly divided into three general areas depending on the technique used to design the controller. These are mixed sensitivity optimization, loop shaping design procedure and μ-controller theory. There also have been certain new developments in the area of robust control such as the polytopic uncertainty approach as discussed in [91], the probabilistic approach where the probability of parameter variation is taken into consideration while developing the controller as in [92], as well as some extensions to the original ideas such as the method of inequalities as discussed in [93].

With UAVs finding new and varied applications in the commercial and hobby flying sector, studies have increasingly relied on robust control to deal with modeling uncertainties. In robust control literature UAV flight controller designs are generally of two types, the linear and the non-linear controllers. In the following section these studies are examined.

2.2.1 Linear $\mathcal{H}_\infty$ Controllers

The linear $\mathcal{H}_\infty$ problem is to minimize the $\mathcal{H}_\infty$ norm of the Linear Fractional Transformation (LFT) between the uncertain plant and all stabilizing controllers. Solving for the optimal solution of this problem is mathematically complex and the controllers generated may at times be non-practical (due to difficulties in suitably implementing them on embedded platforms), hence usually a suboptimal controller would be adopted for actual applications. The algorithm which requires resolving a set of coupled algebraic Ricatti equations (ARE) developed in [94], remains fundamental to this study. In this section alongside quadrotor UAVs, we also examine studies that have developed robust controllers for other rotorcraft UAVs, as well as few fixed wing UAVs.

An interesting comparison study between three different controller design algorithms when it comes to the Linear $\mathcal{H}_\infty$ quadrotor tracking problem, is presented in [95]. The MATLAB `hifoo` and `hinfstruct` routines that produce lower order controllers are compared to the Glover-Doyle optimization algorithm that produce a relatively higher order controller. The Glover-Doyle algorithm which employs the mixed sensitivity 'S over KS over T' approach out performs the other two methods when it comes to both performance and computational time, but at the expense of the controller order. A similar mixed sensitivity $\mathcal{H}_\infty$ controller with a switching recovery scheme to deal with actuator failures is discussed in [21]. The weight functions are developed by trial and error and the recovery scheme which switches between two controllers, offers decent stability during actuator failure.

The studies [96] and [97] analyses the effectiveness of set point tracking ability of the $\mathcal{H}_\infty$ controller under actuator failures and parameter uncertainties, although they fail to comment on tracking abilities and performance in outdoor environments.

In [98] the $\mathcal{H}_\infty$ controller stabilizes a 3 DOF twin rotorcraft, and the controller achieves tracking stability with respect to a varying frequency reference signal as well as it maintains the attitude set point when wind disturbance with a maximum velocity of 4.5 m/s is applied. The coupling action between the translational and rotational action as well as the effect of wind disturbances while tracking a reference signal hasn't been examined in the study. An example of implementation of the $\mathcal{H}_\infty$ robust controller on a 6DOF miniature helicopter can be seen in [24] where we have the design specifications taken in line with the military UAV standards seen in [99]. The simulations depict controller reaction to a 'single frequency cosine wave wind disturbance' although effect on the controller to varying frequency wind disturbances have not been included. The actual test flight results show good hovering and tracking stability during wind gusts as high as 4 m/s. The reduced order controller takes into consideration uncertainty due to wind speed as the exogenous input during the design phase. In general, higher order controllers are generally less desirable than low order ones primarily due to the ease of implementation in the real world as argued in [100] where a new linear matrix inequality algorithm is being introduced to develop lower order $\mathcal{H}_\infty$ controllers.

In the study [25], the Dryden wind turbulence model (see [101]) is employed to generate the wind disturbance on the model, with the mean speed set to 2.5 m/s, shows successful attitude tracking of the linearised fixed wing model (with an EKF estimator) when a $\mathcal{H}_\infty$ controller is used, compared to unsuccessful attempts of regular PD controller. A procedure to calculate tunable gains for each state while attempting to design a tracking $\mathcal{H}_\infty$ controller in the presence of actuator command tracking error, and noisy sensor measurements is presented in [102]. Experiments show good tracking abilities with respect to position but poor attitude hold.

A flight controller for a fixed wing UAV is developed in [27] using the 2 degree of freedom $\mathcal{H}_\infty$ loop shaping technique (see [103]). The realistic controller design with 1st order Pade approximated delays on all four inputs and white sensor noise in the model provides acceptable robustness to the UAV. A study which uses a robust feedback linearisation based inner controller and a $\mathcal{H}_\infty$ outer controller to resolve the path tracking problem is presented in [104]. The controller performs well at 20% mass and inertia alongside aerodynamic force and moment disturbances. The study [102], mentioned previously also presents an LMI based algorithm which minimizes H_2 and $\mathcal{H}_\infty$ cost functions in order to stabilize the linear model of the quadrotor. The final controller is devised in a PID like format enabling easier tuning. However, for the above three studies a verification of Robust Stability (RS) and Robust Performance (RP) using μ-synthesis have been left out hence a lack of clarity in the overall performance remains.

A study that attempts to resolve the flight clearance problem for a X-fighter model can be seen in [105]. A Linear Fractional Transformation (LFT) of the linearised plant model is developed with the normalized uncertain block involving 4 key parameters of the plant. The μ-controller fails to achieve robust stability with the calculated μ value greater than one. Development and implementation of a μ-controller for a fixed wing micro air vehicle (MAV) is presented in [106]. The lateral and longitudinal dynamics are decoupled and control applied separately. Hardware in the loop testing using reduced order μ-controller gives poor control performance due model mismatch as identified by the author.

As a step towards the non-linear domain, the study [23] explains the concept of

Linear Parameter Varying Control where the nonlinearity is distributed among four linear models which switches between each other whenever required. Two sets of controllers are developed for each of four models, one being the LQR optimal controller and the second being the $\mathcal{H}_\infty$ robust controller. Both the controllers perform poorly in terms of reaching the set point although the author notes that $\mathcal{H}_\infty$ performs slightly better compared to the LQR controller when it comes to X and Y coordinate position control.

2.2.2 Non Linear H-infinity Controllers

The nonlinear version of the robust $\mathcal{H}_\infty$ control problem is resolved by solving the Hamilton-Jacobi-Bellman-Isaacs (HJBI) equation (instead of the Algebraic Ricatti Equations) for each of the γ, where γ is the L_2 gain from w to z in the general robust control framework (see page 49).

The path tracking problem for a 2 DOF helicopter is being studied in [22]. The nominal controller exhibits reasonable tracking performance for step reference inputs to yaw and pitch angles, achieving a PID-like format for tuning parameters although external disturbances like wind haven't been taken into account. In [70, 107, 108] hybrid controllers are presented where the attitude and the altitude of the UAV is being controlled by the nonlinear $\mathcal{H}_\infty$ controller and the path being controlled by a model predictive controller. The controller tracks the reference signal relatively well in the presence of uncertainty in the parameters, aerodynamic moments and forces. A similar study where a back stepping controller for dealing with translational motion and altitude control and the nonlinear $\mathcal{H}_\infty$ controller for the attitude control is presented in [26]. The system exhibits robust stability against uncertainty in the parameters.

The problem of under actuation of the quadrotor is being resolved in [109] and [110] using rotors that are free to tilt along the axis of the respective arm. The over actuated models use a feedback linearisation based controller in the former and an inverse dynamics based method in the latter study to achieve tracking. Although the $\mathcal{H}_\infty$ approach of solving the problem haven't been studied yet, a similar study which uses propellers mounted at an angle in order to solve the control problem is described in [111]. The nonlinear $\mathcal{H}_\infty$ controller takes into account a 40% uncertainty in mass as well as presence of disturbances, as it tracks the x, y, z coordinates and yaw angle while trying to maintain zero roll and pitch deviation. The tilt in the motor enables the thrust being distributed onto the body x and y axis giving a handle to stabilize (and there by removing the need to have an additional layer of control with regards to) the uncontrollable DOF. The effectiveness of the controller when the number of uncertain parameters are increased and the performance of a linear $\mathcal{H}_\infty$ controller for the same model would be an interesting problem to be investigated.

In non-linear $\mathcal{H}_\infty$ control theory, while currently there is no efficient method to systematically or numerically solve the HJBI partial differential inequalities as noted in [112], controllers for under-actuated systems like quadrotors have been developed as presented above. The designs still fall short of effectively analyzing the robustness of the overall system and these short comings mostly have been overcome through adequate testing. In this study we are concerned with developing controllers that could theoretically guarantee robust stability and performance before the systems can be deployed. Hence techniques and tools developed using the linear $\mathcal{H}_\infty$ control theory will be utilized to develop the required controllers, instead of relying on the non-linear $\mathcal{H}_\infty$ control techniques developed in the studies discussed.

2.2.3 Robust controllers - conclusion

To conclude, the quadrotor control problem although have been researched from within the robust control framework, there are certain areas that remain unexplored. These are listed below:

1. The μ-synthesis of developed controller-plant system when used to determine robust stability and performance of developed quadrotor controllers, work as a powerful tool to determine the how susceptible the closed loop would be to model uncertainty. The use of this tool has remained largely unexplored when it comes to developing quadrotor controllers.

2. Utilizing the weights developed to capture parameter as well as other uncertainties such as time delays and non-linearity of the plant in a systematic manner specifically for a quadrotor plant needs to be further explored and studied.

3. Finally and most importantly, the case of the optimality of the quadrotor controller weights is an area that remains unexplored in the above studies. For an application like the quadrotor which is heavily dependent on controller optimality for better flight quality and faster flight times [113], the lack of optimal weights form a significant shortfall when it comes to better controller designs.

From aside the above mentioned points, comparison of performances of robust controllers developed using the various robust control strategies (i.e, MSO, LSDP, and the μ-synthesis) also remain largely unexplored within the context of the quadrotor application. The stability problems faced by the quadrotor are discussed next.

2.3 Stability Problems due to Model Uncertainty

In this section the general stability problems (other than external sources of disturbances such as that of wind) which affect the flight of a UAV are listed and the approaches usually used to tackle these are presented briefly.

- Un-modelled Dynamics: Quadcopters, like most plants subjected to high frequency disturbance inputs, tend to exhibit nonlinearities which usually wouldn't be captured by the nominal model. Identifying these dynamics hence helps improving the control effectiveness of the UAV. Some of the common techniques used to deal with the problem are listed below.

  A frequency domain identification technique named CIFER (Comprehensive Identification from Frequency Responses), developed by US Military and NASA for rotorcraft applications have been used to perform frequency domain system identification in [114]. The high frequency dynamics being identified using data logged from an on-board data recorder.

  The inherent robustness of adaptive controllers such as that of [115], the sliding mode controller presented in [116] or the linear parameter varying controller in [117] have been exploited as a defense against un-modelled dynamics. Such techniques free the control engineer from trying to model the dynamics as the controller adapts as required in order to maintain stability.

  The robust control framework allows uncertainty being both neglected and un-modelled dynamics to be represented as complex perturbations in the frequency domain ([118] see pg. 271-274). The approach can be seen implemented on a fixed wing UAV based on μ-controller presented in [119].

- Parametric Uncertainties: The two ways of approach seen generally is either to have an adaptive control mechanism which takes into account the changes in the model, or use a (robust) controller that has considered the possible model parameter swings during development. In the general robust control framework, an unknown plant model with parameter uncertainties is expressed as a nominal plant alongside additive or multiplicative weights as explained in [118] (see pg. 265-274). This approach would be used in this study to model parametric uncertainties. Quadrotor parameters that exhibits uncertainty and their general uncertainty range are presented in Table 3.2.

- Model Linearisation Effects: Quadrotor dynamic equations are usually linearised at the hovering state as seen in [120] or would be linearised at various stages of flight such as hover, cruise, landing etc. and appropriate controllers are used at each modes using a switching mechanism. Linearisation effects always come into the picture when the linear controller attempts to control the actual non-linear plant. However these effects are superimposed alongside other disturbances and noise and usually hidden in error signals or cost functions. Some efforts to study such effects can be seen in [121] but wider studies involving comparison studies with robust controllers are yet to be done. In this study uncertainty brought about by model nonlinearity will be captured using control weights and efforts would be directed to attenuate the linearisation effects.

It can be noted that while various conventional control strategies mentioned in the previous section deal with individual problems that contribute towards system instability in flight, robust controllers remain well positioned to provide a comprehensive framework that can take into account uncertainties in dynamics, parameters and non-linearities. The important conclusions drawn from the literature review conducted are presented in the next section.

2.4 Conclusion

While various aspects of stabilizing and controlling a rotorcraft quadrotor UAV has been well researched in the academic literature, there are still areas that need to be examined when controllers that consider model uncertainties need to be developed. The following conclusions are drawn from the literature survey conducted:

1. Cascaded PID controllers are used mainly in the commercial sector owing to the simplicity in the design and development. While stability margins and optimality aspects of flight performance have been analyzed, performance of the plant in the presence of model uncertainties have not been examined fully. Moreover, apart from adaptive online tuning of parameters during flight and PID controller tuning ideas examined in [122], options to explicitly consider rotorcraft model uncertainty during controller development need to be explored further.

2. Adaptive controllers have played and contributed towards imparting 'robustness' to rotorcraft UAV designs. While adaptive designs have been reliable as the reference input commands change considerably, when constant reference

signals are employed, the ineffectiveness of the controller in the presence of changing model parameters that contribute to un-modeled dynamics (eg. dynamics arising from changing mass and slung load length of a payload attached to a quadrotor) makes it an unsuitable choice for developing robust control designs.

3. Optimal controllers like LQR and the LQG, perform well with nominal models, however as model uncertainty is introduced, guarantees on plant stability disappears, making it unsuitable for situations where significant model uncertainty is expected. Optimal MPC controllers have been effectively used in trajectory tracking have been found to be computationally expensive for implementation.

4. Artificial intelligence based controllers have been mostly used to develop effective trajectory tracking and planning systems. Robustness of the plant towards model uncertainty have been analysed mostly through experimental tests without verification from a theoretical standpoint.

5. Robust control techniques namely MSO, LSDP and μ-synthesis offer the control engineer options to consider model uncertainty during the developmental stage of the controllers as seen in different studies. Controller weights which play a crucial role in determining performance aspects have mostly been determined through a trial and error approach. Comparison studies between various robust control strategies when it comes to controlling small scale rotorcraft systems have also been largely left out in the literature.

To design controllers for systems that involve significant model uncertainty, be it parametric uncertainty or due to dynamics brought to the system by variation in payloads attached to the drone, or the inherent non-linearity of the system, robust control theory offers tools to ensure robustness. While types of robust controllers have been developed to solve the issue a comprehensive framework that can develop optimal controller weight designs while considering the above mentioned uncertainties is missing. Comprehensive comparison studies between different robust control techniques alongside the cascaded-PID controllers (those mostly preferred in the industry) have also been mostly left out.

In light of the literature review conducted the following research questions have been formulated and these would remain central to the rest of this study.

1. Is it possible to develop optimal controller weights alongside a robust controller, to resolve the tracking attitude/altitude problem of a quadrotor? Can these optimal robust designs be developed in all the three major branches of

robust control theory namely the Mixed Sensitivity Optimization, Loop Shaping Design Procedure and the μ-synthesis?

2. For a quadrotor model that is associated with modeling uncertainty is it possible to develop controllers the can theoretically guarantee *nominal performance*, *robust stability* and *robust performance*?.

3. How does Mixed Sensitivity Optimization, Loop Shaping Design Procedure and the μ-synthesis based controllers developed for the resolving the tracking attitude/altitude problem of the quadrotor, compare against each other in terms performance?

Efforts will be directed in upcoming chapters to effectively answer these questions. A framework that can generate optimal weights to be used further in design of the controller (as opposed to assumed after trial and error) will be developed for all the three major robust control strategies (MSO, LSDP and μ-synthesis). SISO and MIMO controller development will be treated separately. Comparisons will be preformed within the developed controllers and also against PID controllers. This brings us to the conclusion of the literature review section and in the next chapter the Quadrotor problem statement is examined.

Chapter 3

A Dynamic Model of the Quadrotor

The tracking attitude and altitude control problem of a quadrotor will be introduced in this chapter. The chapter is partitioned into five sections. In the first section, a general description about a quadrotor will be presented. This will be followed by development of equations of motion and the state space model that would be utilized through out the rest of the study. In section 3.3, details regarding scaling of the model will be provided. This is followed by section 3.4 where model uncertainty description will be presented. In section 3.5, the quadrotor design specifications will be listed, followed by the final section where the chapter summary will be presented.

3.1 General Description

In this thesis a quadrotor consist of 4 propellers placed at the ends of a 'X' frame (as opposed to a '+' frame) as seen in Figure 3.1.

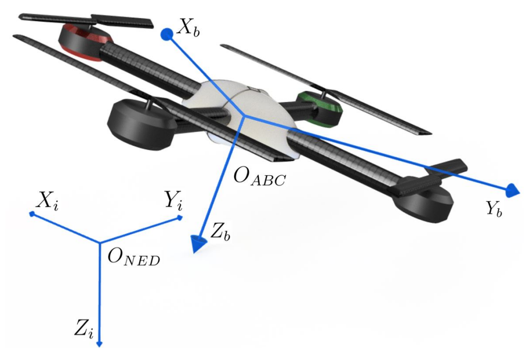


Figure 3.1: The quadrotor with the inertial frame X_i, Y_i, Z_i, and the body frame X_b, Y_b, Z_b.

The inherent under-actuation in a quadrotor arises from the difficulty of having to control 6 degrees of freedom, using only 4 propeller inputs. Alongside this, quadrotor flights are characterized by strong coupling between the rotational and translational forces that arise as a response to the inputs. This contributes towards the general non-linear dynamics that characterize the equations of motion. These factors make controlling a quadrotor a challenging problem.

3.1.1 Inertial and Body Frames

The quadrotor equations of motion are developed based on two different frames of references (see Figure 3.1). The unmovable reference frame, i.e the frame attached to *earth*, assumed to be fixed in space, is called the O_{NED} frame of reference where the subscripts stands for '*North*', '*East*' and '*Down*' respectively. The x, y and z axes associated with this frame have the subscript 'i' which stands for the '*inertial*' frame of reference. The Body Frame otherwise called the O_{ABC} where the subscript stand for '*Aircraft Body Center*' is the frame of reference whose origin coincides with the center of mass of the craft. The x, y and z axes associated with this frame have the subscript 'b' and this stands for the '*body*' frame of reference. The body x and y axis lie in the same plane as that of the rotors, at an angle of 45° with the arms of the quadrotor which are at right angles to each other.

The equations of motion are developed as required in either of these two frames. The variables to be observed and states to be controlled can be rotated into the O_{NED} or O_{ABC} frames as required. Further details will be discussed in section 3.2.1

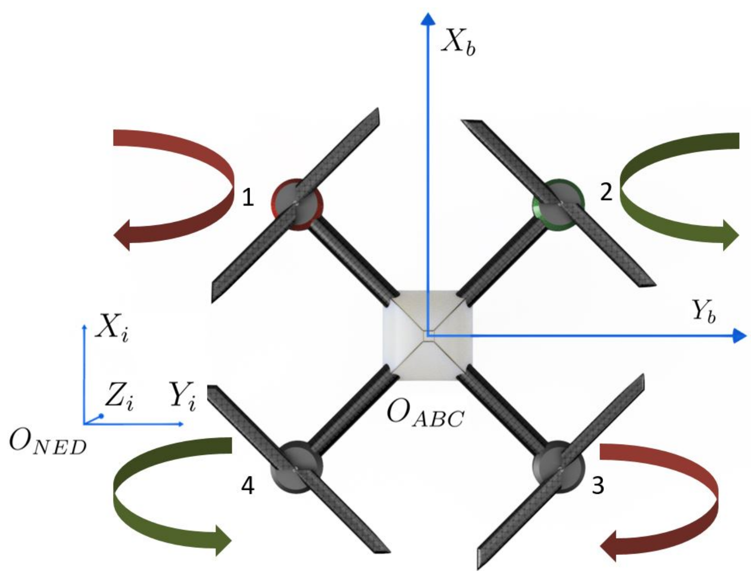
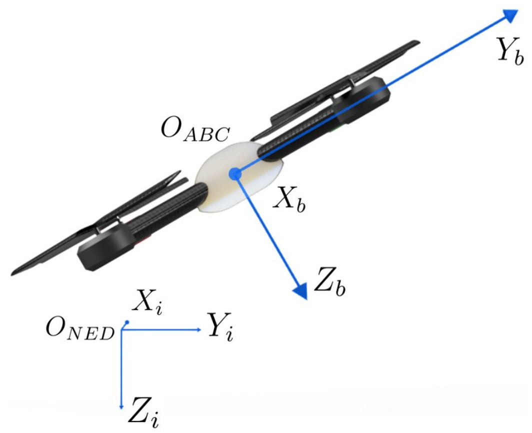


Figure 3.2: Clockwise and anticlockwise propeller movements

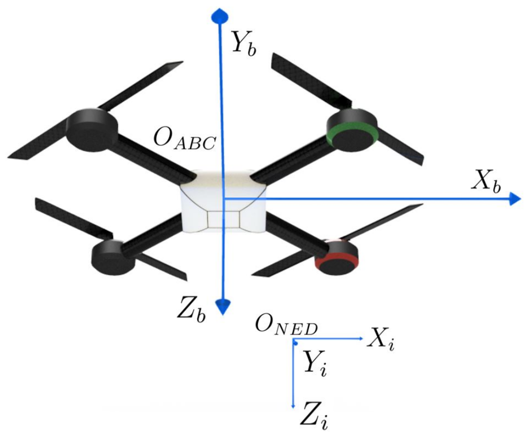
3.1.2 Quadrotor movements

In a quadrotor, opposite pairs of propellers rotate in the same direction while those adjacent to each other rotate in opposite directions. This can be seen in Figure 3.2 where propellers 1 and 3 rotate in clockwise direction, while 2 and 4 rotate in the anticlockwise directions.

The movements of the quadrotor can be classified into four basic categories. All movements of a quadrotor can be expressed as a combination of the these four basic actions. Theses are listed next.



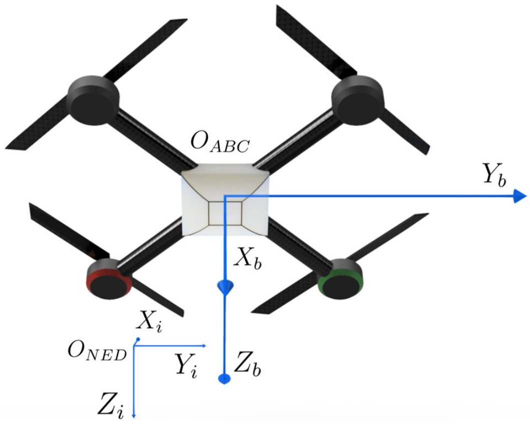
(a) Back elevation



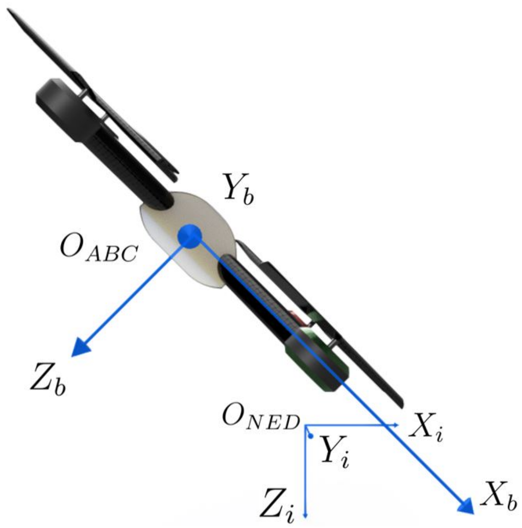
(b) Sideview

Figure 3.3: Roll movement

- Pure Roll: When Propellers 1 and 4 simultaneously decrease (or increase) and 2 and 3 simultaneously increase (or decrease) their angular velocity it results in a movement about the roll axis (See Figure 3.3). In case the changes to the propeller angular velocities are not proportional, it would lead to yawing action resulting from unbalanced torques from the four rotors.



(a) Back elevation



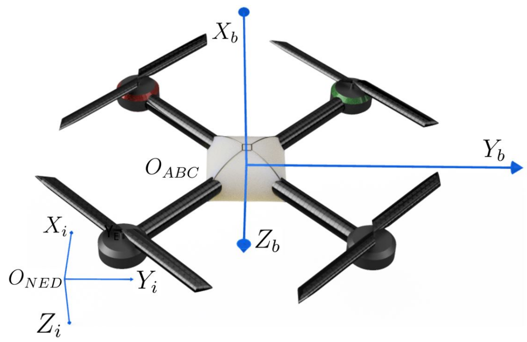
(b) Sideview

Figure 3.4: Pitch movement

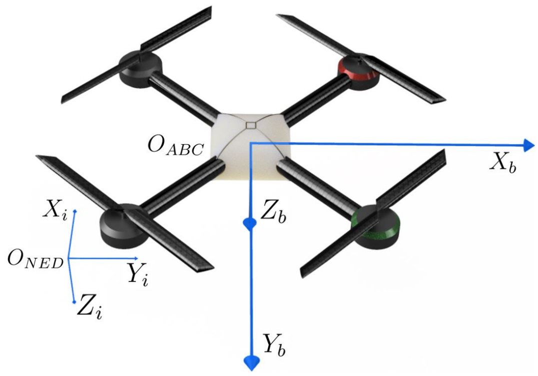
- Pure Pitch: When Propellers 1 and 2 simultaneously decrease (or increase) and 3 and 4 simultaneously increase (or decrease) their angular velocity, it results in a movement about the pitch axis (See Figure 3.4). Similar to roll, unbalanced torques resulting from disproportional angular velocities would result in a yawing motion alongside pitch.

- Yaw: When Propeller 1 and 3 simultaneously decrease and 2 and 4 simultaneously increase their angular velocity it results in a yaw motion. A yaw movement of 90° is shown in Figure 3.5.

- Vertical Thrust: When all the four propellers simultaneous increase or decrease their angular velocity, it results in a corresponding increase or decrease in the altitude of the craft. This is referred to as the vertical thrust action.



(a) Initial position



(b) Final position

Figure 3.5: Yaw movement

The equations of motion are developed in the next section.

3.2 Development of the Dynamic Model

The development of the equations of motion are based on the work presented in [123] and [124]. A quadrotor is an under-actuated mechanical system with 6 degrees of freedom these being position in the x, y, and z earth coordinate frame, and orientation based on Euler angles ϕ, θ and ψ, being the roll, pitch and yaw angles respectively.

As seen in Figure 3.6, any rotation of body frame O_{ABC} about X_B results in a change ϕ, recorded in the body frame as ϕ_b. Similarly changes resulting from rotation about Y_B are recorded as θ_b and that from rotation about Z_B axis as ψ_b. Here the subscript "b" stands for "body" and represents the variables measured in the O_{ABC} frame.

The position vector $\boldsymbol{\chi}_\mathbf{i}$ and orientation vector $\boldsymbol{\zeta}_\mathbf{i}$ as observed in the inertial frame are given by the following vector definitions:

$$\boldsymbol{\chi}_{\mathbf{i}} = \begin{bmatrix} x_i \\ y_i \\ z_i \end{bmatrix}, \quad \boldsymbol{\zeta}_{\mathbf{i}} = \begin{bmatrix} \phi_i \\ \theta_i \\ \psi_i \end{bmatrix}, \tag{3.1}$$

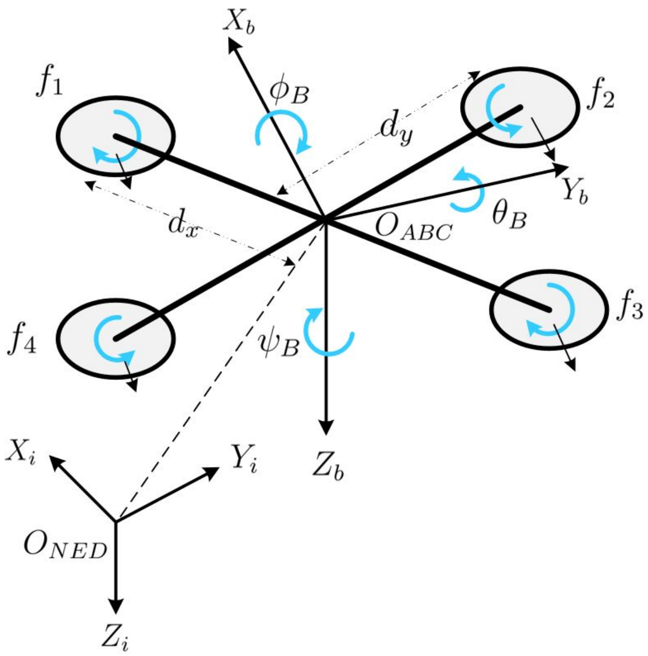


Figure 3.6: The quadrotor model

Notice the subscript 'i' which stands for '*inertial*' and represents the variables measured in the O_{NED} frame. Velocity vectors observed in the body frame- $\mathbf{v_b}$, representing the linear velocity (m/s) and $\boldsymbol{\omega}_{\mathbf{b}}$, representing the angular velocity (rad/s) of the craft are similarly given by the following vector definitions:

$$\mathbf{v_b} = \begin{bmatrix} \dot{x}_b \\ \dot{y}_b \\ \dot{z}_b \end{bmatrix}, \quad \boldsymbol{\omega}_{\mathbf{b}} = \begin{bmatrix} \dot{\phi}_b \\ \dot{\theta}_b \\ \dot{\psi}_b \end{bmatrix} \tag{3.2}$$

The velocity vector $\mathbf{v_b}$ can be rotated towards $\dot{\boldsymbol{\chi}}_i$ using the orthogonal rotation matrix $\mathbf{R}$ [125] :

$$\mathbf{R} = \begin{bmatrix} c_\psi c_\theta & c_\psi s_\theta s_\phi - s_\psi c_\phi & c_\psi s_\theta c_\phi + s_\psi c_\phi \\ s_\psi c_\theta & s_\psi s_\theta s_\phi + c_\psi c_\phi & s_\psi s_\theta c_\phi - c_\psi s_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix} \tag{3.3}$$

where $c_{\bullet}$ and $s_{\bullet}$ represent the trigonometric functions $\cos(\bullet)$ and $\sin(\bullet)$ respectively. Similarly, the angular velocity vector $\boldsymbol{\omega}_{\mathbf{b}}$ can be rotated into its O_{NED} counterpart

$\dot{\boldsymbol{\zeta}}_i$ using the matrix $\mathbf{W^{-1}}$ [125] given by:

$$\mathbf{W^{-1}} = \frac{1}{c_\theta} \begin{bmatrix} c_\theta & s_\phi s_\theta & c_\phi s_\theta \\ 0 & c_\phi c_\theta & -s_\phi c_\theta \\ 0 & s_\phi & c_\phi \end{bmatrix} \tag{3.4}$$

Hence we have $\dot{\boldsymbol{\zeta}}_{\mathbf{i}} = \mathbf{W}^{-1}\boldsymbol{\omega}_{\mathbf{b}}$. With these fundamental vector definitions examined, the equation of motion governing the quadrotor, will be presented in the next section.

3.2.1 Equations of Motion

The equations of motion are developed while maintaining four basic assumptions. Firstly, the mass of the quadrotor is assumed to be symmetric about the body x and y frame. Secondly, the free stream air velocity is considered to be zero. Thirdly, the body along with the propellers are assumed to be rigid, and finally the blades are considered inflexible. Based of Newton-Euler equations for rigid body, the relevant equations of motion can now be developed.

3.2.1.1 Equations of motion derived in the body frame

The equations of motion are first derived in the body frame. To balance out the translational forces acting on the body frame the net force generated by the motors $\mathbf{f}_b$ along with gravitational force $\mathbf{R}^T\mathbf{g}_i$ should be equal to centrifugal force $m(\boldsymbol{\omega}_b \times \mathbf{v}_b)$ plus $m\dot{\mathbf{v}}_{\mathbf{b}}$, the force required for acceleration of the craft. Here $\times$ represents the vector cross product. This can be written as follows:

$$\mathbf{R}^T\mathbf{g}_i + \mathbf{f}_b = m(\dot{\mathbf{v}}_b + \boldsymbol{\omega}_b \times \mathbf{v}_b) \tag{3.5}$$

where $\mathbf{f}_b = \begin{bmatrix} 0 & 0 & \sum_{k=1}^{4} f_k \end{bmatrix}^T$ and $\mathbf{g}_i = \begin{bmatrix} 0 & 0 & -g_i \end{bmatrix}^T$. Here f_k represents the force exerted by the k_{th} rotor and g represents the acceleration due to gravity. (See section 3.2.1.2 for details on f_k).

Similarly, to balance out the rotational forces affecting the craft, the net torque $\boldsymbol{\tau}$ realized on the frame should be equal to the net gyroscopic moments of all the rotors $\mathbf{g_a}$, centripetal forces $\boldsymbol{\omega}_b \times (\mathbf{I}\boldsymbol{\omega}_b)$ and $\mathbf{I}\dot{\boldsymbol{\omega}}_b$, where $\mathbf{I}$ represents the inertia matrix defined about the center of mass.

$$\mathbf{I} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{3.6}$$

and $\boldsymbol{\tau}_b = \left[\begin{array}{ccc} 0 & 0 & \sum_{i=1}^{4} \tau_k \end{array}\right]^T$, where τ_k represents the torque exerted by the k_{th} rotor on the craft (See section 3.2.2 for details on τ_k).

This can be formulated as below:

$$\boldsymbol{\tau}_b = \mathbf{g}_a + \boldsymbol{\omega}_b \times (\mathbf{I}\boldsymbol{\omega}_b) + \mathbf{I}\dot{\boldsymbol{\omega}}_\mathbf{b} \tag{3.7}$$

As pointed out in [126] the net gyroscopic moment produced by rotors $\mathbf{g}_a$, is negligible and hence ignored. Taking into account equations eq. (3.5) and eq. (3.7), and substituting in equations eqs. (3.1) to (3.4), the final equations of motion can be written as follows.

$$\begin{aligned}
\ddot{x}_b &= -g\, s_{\theta_i} + \dot{\psi}_b\, \dot{y}_b - \dot{\theta}_b\, \dot{z}_b \\
\ddot{y}_b &= g\, s_{\phi_i}\, c_{\theta_i} - \dot{\psi}_b\, \dot{x}_b + \dot{\phi}_b\, \dot{z}_b \\
\ddot{z}_b &= -(f_1 + f_2 + f_3 + f_4)\frac{1}{m} + g_i\, c_{\phi_i}\, c_{\theta_i} + \dot{\theta}_b\, \dot{x}_b - \dot{\phi}_b\, \dot{y}_b \\
\ddot{\phi}_b &= \frac{1}{I_{xx}}\big((f_1 - f_2 - f_3 + f_4)\, d_r + (I_{yy} - I_{zz})\, \dot{\theta}_b\, \dot{\psi}_b\big) \\
\ddot{\theta}_b &= \frac{1}{I_{yy}}\big((f_1 + f_2 - f_3 - f_4)\, d_r + (I_{zz} - I_{xx})\, \dot{\phi}_b\, \dot{\psi}_b\big) \\
\ddot{\psi}_b &= \frac{1}{I_{zz}}\big((\tau_1 - \tau_2 + \tau_3 - \tau_4) + (I_{xx} - I_{yy})\, \dot{\phi}_b\, \dot{\theta}_b\big)
\end{aligned} \tag{3.8}$$

Here d_r represents the distance between the center of gravity of the propeller and that of the quadrotor, m represents mass of the quadrotor, f_k represents the force generated by the k_{th} propeller, and τ_k represents the torque exerted by the k_{th} propeller on the center of gravity of the quadrotor. The equations for forces and torques are presented in section 3.2.2.

In eq. (3.8) the derivatives of position as well as translational velocities are described in terms of variables from the body frame. While this is useful in certain occasions (for example in applications that do not involve the use of an inertial frame of reference), it is always desirable for most of the present day applications that utilize a GPS or a similar position tracking mechanism, to identify these variables in an inertial frame of reference. This is examined in the next section.

3.2.1.2 Equations of motion examined in the inertial frame

In the inertial frame, analysing the translational forces, one finds the net linear acceleration equals acceleration produced by the rotor thrust alongside gravity.

$$\ddot{\boldsymbol{\chi}}_\mathbf{i} = \frac{\mathbf{R}\mathbf{f}_b}{m} + \mathbf{g} \tag{3.9}$$

When it comes to calculating the rotational forces acting on the inertial frame, two approaches can be taken. In the first approach the equation $\dot{\boldsymbol{\zeta}}_{\mathbf{i}} = \mathbf{W}^{-1}\boldsymbol{\omega}_{\mathbf{b}}$ is differentiated with respect to time, and eq. (3.7) is substituted to obtain the final expression. The second approach offers a simplification where the following assumption is made: at small angles we have $\dot{\boldsymbol{\zeta}}_{\mathbf{i}} = \boldsymbol{\omega}_{\mathbf{b}}$. As seen from eq. (3.4) the diagonal elements approach unity and all the other elements approach zero when the angles are close to zero. It is usually considered safe to make this assumption considering that the UAV operates most of its time near or at zero angle orientation, as can be seen in [127–130].

3.2.1.3 Equations of motion used in this study

In this study while the linear accelerations and velocity are derived in the inertial frame, the angular accelerations will be derived in body frame. This hybrid state vector is useful when the focus is to design a attitude-altitude stabilization controller [131]. This enables expressing the dynamics of the craft relative to the inertial frame fixed on the earth, and the use of orientation of the craft expressed in the body frame to achieve control and stability. The equation of motions in this approach based on eqs. (3.4), (3.7) and (3.9) can be derived as follows (further see [132] for the derivation, and [21, 95, 133] for the application of the same set of quadrotor equations of motion):

$$
\begin{aligned}
\dot{\phi}_i &= \dot{\phi}_b + (\dot{\theta}_b\, s_{\phi_i} + \dot{\psi}_b\, c_{\phi_i})\, t_{\theta_i} \\
\dot{\theta}_i &= \dot{\theta}_b\, c_{\phi_i} - \dot{\psi}_b\, s_{\phi_i} \\
\dot{\psi}_i &= (\dot{\theta}_b\, s_{\phi_i} + \dot{\psi}_b\, c_{\phi_i})\, \sec(\theta_i) \\
\ddot{x}_i &= -\frac{f_1+f_2+f_3+f_4}{m}\,(s_{\phi_i}\, s_{\psi_i} + c_{\phi_i}\, c_{\psi_i}\, s_{\theta_i}) \\
\ddot{y}_i &= -\frac{f_1+f_2+f_3+f_4}{m}\,(c_{\phi_i}\, s_{\psi_i}\, s_{\theta_i} - c_{\psi_i}\, s_{\phi_i}) \\
\ddot{z}_i &= g - \frac{f_1+f_2+f_3+f_4}{m}\,(c_{\phi_i}\, c_{\theta_i}) \\
\ddot{\phi}_b &= \frac{1}{I_{xx}}\big((f_1 - f_2 - f_3 + f_4)\, d_y + (I_{yy} - I_{zz})\, \dot{\theta}_b\, \dot{\psi}_b\big) \\
\ddot{\theta}_b &= \frac{1}{I_{yy}}\big((f_1 + f_2 - f_3 - f_4)\, d_x + (I_{zz} - I_{xx})\, \dot{\phi}_b\, \dot{\psi}_b\big) \\
\ddot{\psi}_b &= \frac{1}{I_{zz}}\big(\tau_1 - \tau_2 + \tau_3 - \tau_4 + (I_{xx} - I_{yy})\, \dot{\phi}_b\, \dot{\theta}_b\big)
\end{aligned}
\tag{3.10}
$$

This brings us to the end of development of quadrotor equations of motion. The selection of the parameters, used in this development, and system input definitions are presented next.

3.2.2 Model Parameters, Propeller Thrusts and Torques

The actual model parameters used in the study are based on the Ascending technologies (AscTec) Pelican Quadrotor, a commercially available quadrotor developed by Ascending Technologies GmbH (`http://www.asctec.de`), and used in [134]. The nominal values of thrust coefficient c_t and power coefficient c_p are based on information documented in the University of Illinois propeller database [135] for the $10'' \times 4.7''$ APC propeller. The length of the quadrotor arms have been slightly increased to accommodate the propellers into the original design. The basic dimensions of the quadrotor is shown in Fig. 3.7.

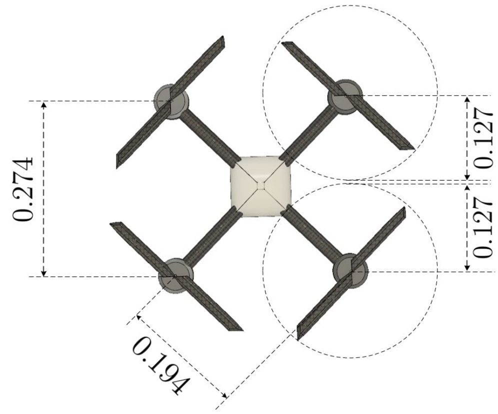


Figure 3.7: Quadrotor basic dimensions (units in [m]).

This particular quadrotor, which can be modified easily for research purposes [136], remains popular among academics [137–140]. The parameters used are listed in Table 3.1.

Table 3.1: The parameters of the quadrotor used in this study

Parameters	Value
m	1.27 kg
g	9.81 m/s^2
d_x	0.194 m
d_y	0.194 m
c_t	0.1
c_p	0.045
I_{XX}, I_{YY}	0.04339 kg m^2
I_{ZZ}	0.0705 kg m^2

Equations for the thrusts produced by the propeller are presented in [141]. The total thrust produced by the k_{th} rotor is given by

$$f_k = \frac{c_t \, \rho \, {n_k}^2 \, d^4}{3600}, \tag{3.11}$$

and net torque exerted by the k_{th} rotor on the body of the quadrotor is calculated as

$$\tau_k = \frac{c_p \, \rho \, {n_k}^2 \, d^5}{2\pi \, 3600} \tag{3.12}$$

where n_k is the angular velocity of the k_{th} propeller calculated in rotations per minute, ρ the density of air at sea level ($1.225\,\mathrm{kg\,m^{-3}}$) and d ($0.254\,\mathrm{m}$), the diameter of the propeller disk.

The maximum recommended RPM for the $10'' \times 4.7''$ APC propeller is 6500 [135]. The maximum thrust produced based on eq. (3.11) by a single rotor amounts to 5.985 N and hence maximum payload that the craft can lift excluding its own mass, is 1.17 Kg.

3.2.3 The Linearised State Space Model

Having determined the nonlinear dynamic equations of motion, and parameters of the model, the purpose of this section is to derive the linear state space model of the quadrotor attitude-altitude system. The section is subdivided into two parts. In the first subsection the control architecture as well as the state, input and output vectors will be defined. The system definition in terms of the output variables to be controlled, and states variables to be observed, are presented. In the second subsection the linearised model of the quadrotor is developed and the state space model is presented.

3.2.3.1 Control Architecture and Control System Vectors

The quadrotor control architecture can be structured in a two tier-format such that the position tracking control would be managed by an outer framework, while the inner loop handles attitude-altitude stabilization and tracking, following [142]. The outer loop receives reference commands in the form of position coordinates (x_i^D, y_i^D) which it converts to desired roll and pitch rates ($\dot{\phi}_b^D$, $\dot{\theta}_b^D$). This along with the desired yaw and altitude rates, obtained from the time series derivation of desired yaw orientation and altitude coordinates ($\dot{\psi}_b^D$, $\dot{z}_i^D$), will be fed to the inner loop controller. Aside from the subscripts 'i' and 'b' which indicate the frame of reference, the superscript 'D' represents the 'desired' values in the reference signal. This control architecture can be seen in Figure 3.8.

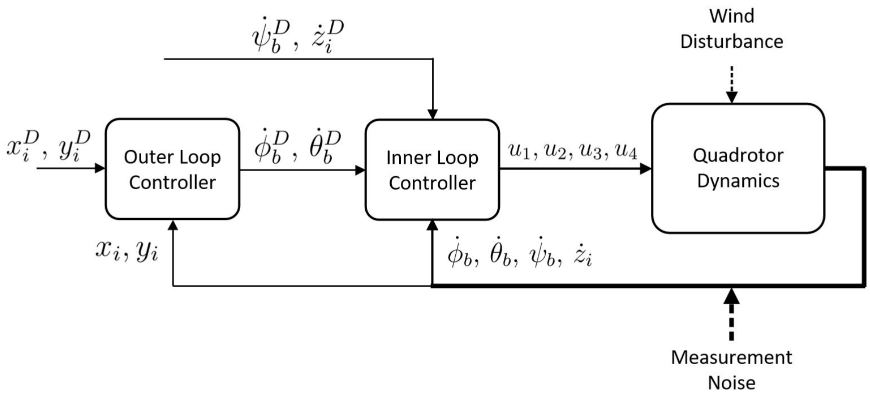


Figure 3.8: Control architecture

The focus of this study is to resolve attitude-altitude rate tracking problem managed by the inner loop controller. Hence the output variables that would be tracked and controlled are selected to be the roll, pitch, yaw and altitude rates.

The state, input and output vectors of the quadrotor model defined in accordance with the set of differential eq. (3.10), are as follows:

$$\begin{aligned} \mathbf{x} &= \begin{bmatrix} x_i \ y_i \ z_i \ \phi_i \ \theta_i \ \psi_i \ \dot{x}_i \ \dot{y}_i \ \dot{z}_i \ \dot{\phi}_b \ \dot{\theta}_b \ \dot{\psi}_b \end{bmatrix}^T, \\ \mathbf{y} &= \begin{bmatrix} \dot{\phi}_b \ \dot{\theta}_b \ \dot{\psi}_b \ \dot{z}_i \end{bmatrix}^T, \text{ and} \\ \mathbf{u} &= \begin{bmatrix} u_1 \ u_2 \ u_3 \ u_4 \end{bmatrix}^T, \end{aligned} \tag{3.13}$$

where $\mathbf{x}$ represents the state, $\mathbf{y}$, the output and $\mathbf{u}$, the input variables where u_i equals ω_i, the *angular velocity* of the i_{th} propeller in rotations per minute (RPM). Alongside the equation for output variables and differential eq. (3.10) rewritten as a first order differential equation (by substituting eq. (3.13)) the system definition can be presented as follows:

$$\begin{aligned} \mathbf{y} &= \mathbf{g}(\mathbf{x}, \mathbf{u}) \\ \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, \mathbf{u}), \end{aligned} \tag{3.14}$$

where $\mathbf{g}(\mathbf{x}, \mathbf{u})$ equals

$$\mathbf{g}(\mathbf{x}, \mathbf{u}) = \begin{bmatrix} \dot{\phi}_b \\ \dot{\theta}_b \\ \dot{\psi}_b \\ \dot{z}_i \end{bmatrix} \tag{3.15}$$

and $\boldsymbol{f}(\mathbf{x}, \mathbf{u})$ is given by

$$\mathbf{f}(\mathbf{x},\mathbf{u}) = \begin{bmatrix} \dot{x}_i \\ \dot{y}_i \\ \dot{z}_i \\ \dot{\phi}_b + (\dot{\theta}_b\, s_{\phi_i} + \dot{\psi}_b\, c_{\phi_i})\, t_{\theta_i} \\ \dot{\theta}_b\, c_{\phi_i} - \dot{\psi}_b\, s_{\phi_i} \\ (\dot{\theta}_b\, s_{\phi_i} + \dot{\psi}_b\, c_{\phi_i})\ \sec(\theta_i) \\ -\frac{f(u_1)+f(u_2)+f(u_3)+f(u_4)}{m}\,(c_{\phi_i}\, s_{\theta_i}\, c_{\psi_i}\ +\ s_{\phi_i}\, s_{\psi_i}) \\ -\frac{f(u_1)+f(u_2)+f(u_3)+f(u_4)}{m}\,(c_{\phi_i}\, s_{\theta_i}\, s_{\psi_i}\ -\ s_{\phi_i}\, c_{\psi_i}) \\ g - \frac{f(u_1)+f(u_2)+f(u_3)+f(u_4)}{m}\,(c_{\phi_i}\, c_{\theta_i}) \\ \frac{f(u_1)-f(u_2)-f(u_3)+f(u_4)}{I_{xx}}\, d_y + \frac{I_{yy}-I_{zz}}{I_{xx}}\, \dot{\theta}_b\, \dot{\psi}_b \\ \frac{f(u_1)-f(u_2)+f(u_3)-f(u_4)}{I_{yy}}\, d_x + \frac{I_{zz}-I_{xx}}{I_{yy}}\, \dot{\phi}_b\, \dot{\psi}_b \\ \frac{-\tau(u_1)-\tau(u_2)+\tau(u_3)+\tau(u_4)}{I_{zz}} + \frac{I_{xx}-I_{yy}}{I_{zz}}\, \dot{\phi}_b\, \dot{\theta}_b \end{bmatrix} \tag{3.16}$$

Here $f(u_k) \equiv f_k$ and $\tau(u_k) \equiv \tau_k$. To design a controller that stabilizes the quadrotor and achieves satisfactory reference tracking, the non-linear model of eq. (3.14) needs to be linearized at a point of equilibrium. This is discussed in the next subsection.

3.2.3.2 Jacobian Linearisation

For a non-linear system of the form $\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x},\mathbf{u})$, $\bar{\mathbf{x}}$ can be called an equilibrium point if an input $\bar{\mathbf{u}}$ can be identified such that $\mathbf{f}(\bar{\mathbf{x}},\bar{\mathbf{u}}) = 0$, such that when the system begins operation from an initial point $\mathbf{x}(t_0) = \bar{\mathbf{x}}$, with a persistent input $\bar{\mathbf{u}}\ \forall\ t > t_0$, $\mathbf{x}(t)$ remain equal to $\bar{\mathbf{x}}$ [143].

For an open loop quadrotor system, subject to zero external disturbances, the state of perfect hover, i.e. when all of the propellers rotate at the same RPM and the net thrust generated, precisely equals the weight of the craft, is the only such equilibrium point present. For this *marginally stable* equilibrium position we have $\bar{\mathbf{x}} = \mathbf{0}_{12\times1}$ and $\bar{\mathbf{u}} = \frac{m\,g}{4}\,\mathbf{I}_{4\times1}$. It is around this equilibrium point, that the craft will be expected to operate and controllers expected to provide asymptotic stability, during flight operations. Consider the system of eq. (3.14). We have at the equilibrium point:

$$\begin{aligned} \mathbf{f}(\bar{\mathbf{x}},\bar{\mathbf{u}}) &= \mathbf{0} \\ \mathbf{g}(\bar{\mathbf{x}},\bar{\mathbf{u}}) &= \mathbf{0} \end{aligned} \tag{3.17}$$

Now considering small perturbations to the system at this point, we can define the following perturbation variables:

$$\begin{aligned} \boldsymbol{\delta}_{\mathbf{x}}(t) &= \mathbf{x}(t) - \bar{\mathbf{x}} \\ \boldsymbol{\delta}_{\mathbf{u}}(t) &= \mathbf{u}(t) - \bar{\mathbf{u}} \end{aligned} \tag{3.18}$$

Corresponding system equations can be derived as follows:

$$\begin{aligned} \dot{\boldsymbol{\delta}}_{\mathbf{x}}(t) &= \mathbf{f}(\bar{\mathbf{x}} + \boldsymbol{\delta}_{\mathbf{x}}(t), \bar{\mathbf{u}} + \boldsymbol{\delta}_{\mathbf{u}}(t)) \\ \boldsymbol{\delta}_{\mathbf{y}}(t) &= \mathbf{g}(\bar{\mathbf{x}} + \boldsymbol{\delta}_{\mathbf{x}}(t), \bar{\mathbf{u}} + \boldsymbol{\delta}_{\mathbf{u}}(t)) \end{aligned} \tag{3.19}$$

These exact equations are expanded using a Taylor series expansion, with terms higher than the 1^{st} order neglected.

$$\begin{aligned} \dot{\boldsymbol{\delta}}_{\mathbf{x}}(t) &= \mathbf{f}(\bar{\mathbf{x}}, \bar{\mathbf{u}}) + \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\substack{\mathbf{x}=\bar{\mathbf{x}} \\ \mathbf{u}=\bar{\mathbf{u}}}} \boldsymbol{\delta}_{\mathbf{x}}(t) + \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{\substack{\mathbf{x}=\bar{\mathbf{x}} \\ \mathbf{u}=\bar{\mathbf{u}}}} \boldsymbol{\delta}_{\mathbf{u}}(t) \\ \boldsymbol{\delta}_{\mathbf{y}}(t) &= \mathbf{g}(\bar{\mathbf{x}}, \bar{\mathbf{u}}) + \left.\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\right|_{\substack{\mathbf{x}=\bar{\mathbf{x}} \\ \mathbf{u}=\bar{\mathbf{u}}}} \boldsymbol{\delta}_{\mathbf{x}}(t) + \left.\frac{\partial \mathbf{g}}{\partial \mathbf{u}}\right|_{\substack{\mathbf{x}=\bar{\mathbf{x}} \\ \mathbf{u}=\bar{\mathbf{u}}}} \boldsymbol{\delta}_{\mathbf{u}}(t) \end{aligned} \tag{3.20}$$

Substituting eq. (3.14) in the above equation and introducing the variables

$$\begin{aligned} \mathbf{A}_{n\times n} &= \left.\frac{\partial \mathbf{f}}{\partial \mathbf{x}}\right|_{\substack{\mathbf{x}=\bar{\mathbf{x}} \\ \mathbf{u}=\bar{\mathbf{u}}}} \boldsymbol{\delta}_{\mathbf{x}}(t)\ , \quad & \mathbf{B}_{n\times m} &= \left.\frac{\partial \mathbf{f}}{\partial \mathbf{u}}\right|_{\substack{\mathbf{x}=\bar{\mathbf{x}} \\ \mathbf{u}=\bar{\mathbf{u}}}} \boldsymbol{\delta}_{\mathbf{u}}(t) \\ \mathbf{C}_{p\times n} &= \left.\frac{\partial \mathbf{g}}{\partial \mathbf{x}}\right|_{\substack{\mathbf{x}=\bar{\mathbf{x}} \\ \mathbf{u}=\bar{\mathbf{u}}}} \boldsymbol{\delta}_{\mathbf{x}}(t)\ , \quad & \mathbf{D}_{p\times m} &= \left.\frac{\partial \mathbf{g}}{\partial \mathbf{u}}\right|_{\substack{\mathbf{x}=\bar{\mathbf{x}} \\ \mathbf{u}=\bar{\mathbf{u}}}} \boldsymbol{\delta}_{\mathbf{u}}(t) \end{aligned} \tag{3.21}$$

where n is the number of states of the system, m the number of inputs and p the number of outputs, we get the final form:

$$\begin{aligned} \dot{\boldsymbol{\delta}}_{\mathbf{x}}(t) &= \mathbf{A}\,\boldsymbol{\delta}_{\mathbf{x}}(t) + \mathbf{B}\,\boldsymbol{\delta}_{\mathbf{u}}(t) \\ \boldsymbol{\delta}_{\mathbf{y}}(t) &= \mathbf{C}\,\boldsymbol{\delta}_{\mathbf{x}}(t) + \mathbf{D}\,\boldsymbol{\delta}_{\mathbf{u}}(t) \end{aligned} \tag{3.22}$$

The Jacobian linearised form of the non-linear equation eq. (3.14) at the equilibrium point, is thus given by eq. (3.22). Now replacing the perturbation variables with system variables, we can write down the linearised spate space model as:

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{A}\,\mathbf{x} + \mathbf{B}\,\mathbf{u} \\ \mathbf{y} &= \mathbf{C}\,\mathbf{x} + \mathbf{D}\,\mathbf{u} \end{aligned} \tag{3.23}$$

At the equilibrium point $\bar{\mathbf{x}} = \mathbf{0}_{12\times 1}$ and $\bar{\mathbf{u}} = \frac{mg}{4}\,\mathbf{I}_{4\times 1}$ for the quadrotor model and parameters as given in Table 3.1, we have the state-space matrices

$$\mathbf{A_{12\times12}} = \begin{pmatrix} 0&0&0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&0&0&0&1&0&0&0&0\\ 0&0&0&0&0&0&0&0&1&0&0&0\\ 0&0&0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0&0&0&0&1\\ 0&0&0&0&-9.81&0&0&0&0&0&0&0\\ 0&0&0&9.81&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0&0 \end{pmatrix},$$

$$\mathbf{B_{12\times4}} = \begin{pmatrix} 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ -0.001046&-0.001046&-0.001046&-0.001046\\ 0.004562&-0.004562&-0.004562&0.004562\\ 0.004562&-0.004562&0.004562&-0.004562\\ -0.0003428&-0.0003428&0.0003428&0.0003428 \end{pmatrix},$$

$$\mathbf{C_{4\times12}} = \begin{pmatrix} 0&0&0&0&0&0&0&0&0&1&0&0\\ 0&0&0&0&0&0&0&0&0&0&1&0\\ 0&0&0&0&0&0&0&0&0&0&0&1\\ 0&0&0&0&0&0&0&0&1&0&0&0 \end{pmatrix}, \quad \mathbf{D_{4\times4}} = \begin{pmatrix} 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0\\ 0&0&0&0 \end{pmatrix} \tag{3.24}$$

To do a quick check with regards to whether the linear system is state controllable would be to check if the controllability matrix $\boldsymbol{\mathcal{C}}$ has a full rank [144]. The controllability matrix is given by:

$$\boldsymbol{\mathcal{C}} = \Big[\mathbf{B} \quad \mathbf{AB} \quad \mathbf{A^2B} \quad \cdots \quad \mathbf{A}^{n-1}\mathbf{B}\Big]. \tag{3.25}$$

Substituting for $\mathbf{A}$ and $\mathbf{B}$, the rank of $\boldsymbol{\mathcal{C}}$ for the quadrotor system defined in eq. (3.24) can be calculated to equal 12, i.e. same as the number of states. The concept of state controllability doesn't throw light into the quality of control for example say whether the system can be held at a particular state indefinitely.

Functional controllability, a concept introduced by Rosenbrock [145], which offers insight into the number of outputs that can be controlled independently [146] can be checked by calculating the normal rank of the system over all frequencies. To be fully functionally controllable this calculated rank needs to be equal to the number of outputs of the system. For the quadrotor model under consideration, the normal rank amounts to unity. Hence this lets us conclude that the system is not fully functionally controllable. Checking for functional controllability as well as designing robust controllers, depends on the proper scaling of the system. The next section deals with this subject.

3.3 System Scaling

Scaling the system is crucial for both developing the controller and uncertainty weights, as well as for implementation on a micro-processor. The method employed here is to normalize the variables entering and exiting the system with respect to the maximum expected change (please refer to section 1.4 in [118], for further details). To enable this, the following weights are first defined:

$$\mathbf{D_e} = \begin{bmatrix} 0.25 & 0 & 0 & 0 \\ 0 & 0.25 & 0 & 0 \\ 0 & 0 & 0.25 & 0 \\ 0 & 0 & 0 & 0.25 \end{bmatrix}$$

where $\mathbf{D_e}$ represents the maximum allowed control error. It is set to to 0.25 rad/s for roll, pitch and yaw rates and 0.25 m/s for altitude rate. These values were chosen based on the practical considerations of the AscTech Pelican quadrotor.

$$\mathbf{D_u} = \begin{bmatrix} 1810 & 0 & 0 & 0 \\ 0 & 1810 & 0 & 0 \\ 0 & 0 & 1810 & 0 \\ 0 & 0 & 0 & 1810 \end{bmatrix}$$

where $\mathbf{D_u}$ represents the maximum allowed change in propeller angular velocity from the steady state value of 4690 RPM (i.e, corresponding to a weight of $\frac{mg}{4}$). The maximum allowed propeller RPM is 6500, hence the diagonal elements of $\mathbf{D_u}$ calculates to $6500 - 4690 = 1810$ RPM.

$$\mathbf{D_d} = \begin{bmatrix} 0.5 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 0.5 \end{bmatrix}, \quad \mathbf{D_r} = \begin{bmatrix} 0.5 & 0 & 0 & 0 \\ 0 & 0.5 & 0 & 0 \\ 0 & 0 & 0.5 & 0 \\ 0 & 0 & 0 & 1.5 \end{bmatrix}$$

where $\mathbf{D_d}$ and $\mathbf{D_r}$ represents the maximum expected disturbance signal and maximum expected change in the reference signal. A maximum expected disturbance signal and reference change of 0.5 rad/s is expected for roll, pitch and yaw rates while 0.5 m/s is the expected altitude rate change. With the symbol ($\check{}$) representing the unscaled values, the original variables can be scaled as follows:

$$\mathbf{y} = \mathbf{D_e}^{-1}\check{\mathbf{y}}\ , \quad \mathbf{e} = \mathbf{D_e}^{-1}\check{\mathbf{e}}\ , \quad \mathbf{u} = \mathbf{D_u}^{-1}\check{\mathbf{u}} \tag{3.26}$$

where we have the outputs and errors scaled with respect to the error weight while the inputs are scaled with respect to the input weight. For the reference signal, a normalized variable $\tilde{\mathbf{r}}$ is introduced. We have $\check{\mathbf{r}} = \mathbf{D_r}\,\tilde{\mathbf{r}}$, and the original variable can now be scaled as $\mathbf{r} = \mathbf{D_e}^{-1}\,\check{\mathbf{r}}$. Let $\mathbf{D_e}^{-1}\,\mathbf{D_r} = \boldsymbol{\mathcal{R}}$. The final equation hence becomes

$$\mathbf{r} = \boldsymbol{\mathcal{R}}\,\tilde{\mathbf{r}} \tag{3.27}$$

The general control equations

$$\begin{aligned} \check{\mathbf{y}} &= \check{\mathbf{G}}\,\check{\mathbf{u}} + \check{\mathbf{G}}_\mathbf{d}\,\check{\mathbf{d}} \\ \check{\mathbf{e}} &= \check{\mathbf{y}} - \check{\mathbf{r}} \end{aligned} \tag{3.28}$$

can now be written as:

$$\begin{aligned} \mathbf{D_e}\,\mathbf{y} &= \check{\mathbf{G}}\,\mathbf{D_u}\,\mathbf{u} + \check{\mathbf{G}}\,\mathbf{D_d}\,\mathbf{d} \\ \mathbf{D_e}\,\mathbf{e} &= \mathbf{D_e}\,\mathbf{y} - \mathbf{D_e}\,\mathbf{r} \end{aligned} \tag{3.29}$$

Using the scaled transfer function matrices

$$\mathbf{G} = \mathbf{D_e^{-1}}\,\check{\mathbf{G}}\,\mathbf{D_u} \text{ and } \mathbf{G_d} = \mathbf{D_e^{-1}}\,\check{\mathbf{G}}_\mathbf{d}\,\mathbf{D_d} \tag{3.30}$$

eq. (3.28) can be rewritten as

$$\begin{aligned} \mathbf{y} &= \mathbf{G}\,\mathbf{u} + \mathbf{G_d}\,\mathbf{d} \\ \mathbf{e} &= \mathbf{y} - \mathbf{r} \end{aligned} \tag{3.31}$$

The resulting model in terms of scaled variables are shown in Figure 3.9.

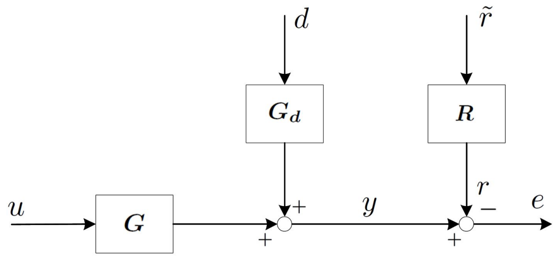


Figure 3.9: The model enabled by the scaled variables

The control objective now is to keep the error $\|\mathbf{e}\|_\infty \leq 1$ using an input $\|\mathbf{u}\|_\infty \leq 1$ subject to disturbances $\|\mathbf{d}\|_\infty \leq 1$ and a reference set point $\|\tilde{\mathbf{r}}\|_\infty \leq 1$. Here $\|\cdot\|_\infty$ refers to the $\mathcal{H}_\infty$ norm (see section 4.3.1). Sources of modelling uncertainty, considered during development of robust controllers are presented in the next section.

3.4 Uncertainty Sources

Accurately characterizing the model uncertainty becomes important while designing controllers for a plant, as it accounts for being able to quantify the unknown [147]. The approach taken is to consider the different sources of known uncertainty and design weights to capture them relative to the nominal model. Three types of uncertainty sources are considered in the study, these being parametric, time delay and uncertainty due to non-linearity. Table 3.2 shows the magnitude of sources considered.

Table 3.2: Uncertainty Sources

Parameters	% Uncertainty	Range
I_{XX}, I_{YY}	0.04339 ±10% $\mathrm{kg\,m^2}$	$0.0391 \leq I_{XX},\, I_{YY} \leq 0.0477$
I_{ZZ}	0.0705 ±10% $\mathrm{kg\,m^2}$	$0.0634 \leq I_{ZZ} \leq 0.0775$
m	1.27 ±50% kg	$0.6350 \leq m \leq 1.9050$
dr	0.194±30% m	$0.1358 \leq d_r \leq 0.2522$
c_t	0.1 ±50%	$0.0500 \leq c_t \leq 0.1500$
c_p	0.045 ±30%	$0.0315 \leq c_p \leq 0.0585$
t_D	0.004 ±10% s	$0.0036 \leq t_D \leq 0.0044$

Variables	Range for Model Linearisation
ϕ_i, θ_i, ψ_i	-20 ↔ 20 (deg)
$\dot{\phi}_b$, $\dot{\theta}_b$, $\dot{\psi}_b$	-0.2 ↔ 0.2 (rad/s)
$\dot{x}_i$, $\dot{y}_i$, $\dot{z}_i$	-0.5 ↔ 0.5 (m/s)
u_1, u_2, u_3, u_4	4590 ↔ 4790 (RPM)

The first section of the table presents the parametric uncertainty sources. These values are based on percentage changes considered generally in literature [26, 104, 107, 148, 149] where parametric uncertainties are modeled. Time delay uncertainty with a nominal value of 0.004 s is represented by t_D. The second section lists the ranges across with the model is linearised. The details regarding development of the weights will be discussed in later sections alongside the controller design.

3.5 Quadrotor design specifications

The design specifications for the nominal nonlinear plant are listed out below. The design specifications are based on both recommendations from literature [99, 150–153], and also based on discussions with the development team at Envirobotics Limited [154] (more of which will be explored in Chapter 6).

Among the constraints listed there are both soft constraints (colored in green) and hard constraints (colored in red). These form the basis for development of the constraints of the optimization problem in the final stage of the controller design. The soft constraints are flexible and while it would be excellent for the quadrotor to achieve them, these limits are flexible and can be relaxed to meet other more important constraints (eg. robustness margins). The hard constraints fall at the farthest end of the allowed performance limits. Failing to meet the hard limits will be considered as *not having met* the required performance standards in the study. It is important to note that surpassing the hard limits might not always result in instability.

Roll and Pitch Rate Loop:

1. For a reference input of $y(t) = 0.25$ rad/s
   - % Overshoot < 5, 8
   - Rise Time < 0.1 s, 0.65 s
   - Settling Time < 0.25 s, 0.85 s
2. The scaled inputs limited as -1 $\leq u(t) \leq$ 1
3. The closed loop bandwidth of atleast 10 rad/s

Yaw Rate Loop:

1. For a reference input of $y(t) = 0.5$ rad/s

- % Overshoot < 5, 20
- Rise Time < 0.1 s, 0.65 s
- Settling Time < 0.3 s, 0.85 s

2. The scaled inputs limited as -1 $\leq u(t) \leq$ 1
3. The closed loop bandwidth atleast 10 rad/s

Altitude Rate Loop:

1. For a reference input of $y(t) = 1.5$ m/s
   - % Overshoot < 20, 25
   - Rise Time < 0.3 s, 0.75 s
   - Settling Time < 1.25 s, 1.75 s
2. The scaled inputs limited as -1 $\leq u(t) \leq$ 1
3. The closed loop bandwidth atleast 2 rad/s

The limits are formulated into inequality constraints while developing the framework for controller design. These are discussed in subsequent chapters.

3.6 Chapter Summary

The main ideas presented in he chapter are summarized below:

- The general dynamic quadrotor mathematical model is introduced in this chapter and equations of motion that are used in the study are derived.
- The specific parameters used to develop the quadrotor model are presented.
- Jacobian linearisation principles are presented and numerical linearisation is carried out to derive the state space model of the plant.
- A system scaling is presented based on the limits of input, error, expected disturbance, and change in reference input signal.
- Expected model uncertainty based on uncertainty in parameter accuracy, presence of time delays and model non-linearity are presented. These estimates will used during controller development in following chapters.

- Design specifications that need to be met for the performance of the nominal plant design are presented. These specifications will be used as constraints that need to be met during development of the controllers in the upcoming chapters.

This brings us to the end of this chapter. In the next chapter the development of SISO robust controllers for the developed quadrotor model will be examined in detail.

Chapter 4

Robust Controller Designs: A Preview on SISO Systems

This chapter presents an introduction to Robust Control Theory together with a new method for selecting the design weights by resolving a constrained nonlinear optimization problem. The decoupled linear quadrotor model will be taken as the plant model for designing the controllers. The chapter is divided into ten sections. A general introduction to the robust control problem that is being examined is presented in the first section. In the section 4.2, the decoupled quadrotor model is presented. A general introduction to robust control theory is presented in the section 4.3. The three control strategies Mixed sensitivity Optimization (MSO) control, Loop Shaping Design Procedure (LSDP) and μ-synthesis are explained in the subsequent sections (sections 4.4, 4.5, 4.6, 4.7, and 4.8). In each of these sections, controllers for attitude and altitude tracking of the quadrotor model, will be developed and their performances, examined. In the next section 4.9, a comparison of performance of the various controller designs is presented. In the final section 4.10, conclusions drawn from the chapter are summarized.

4.1 Introduction

In Chapter 3, a Multi-Input Multi-Output (MIMO) model of the quadrotor model, developed for resolving the tracking problem was presented. In the commercial industry the attitude-altitude tracking problem is usually resolved by breaking down this MIMO problem into multiple 'Single-Input Single-Output' SISO problems. SISO controllers are relatively easy to implement compared to the MIMO counterparts and as long as a MIMO plant can be decoupled into individual SISO plants at least around the frequencies where the plant operates, industries opt for

SISO controllers owing to the relative easiness of design [155, 156]. As such, popular rotor craft autopilots as surveyed in [10, 157] and [158], use 4 PID controllers to control the SISO loops, while resolving the attitude-altitude tracking problem. These SISO loops will be examined in the first part of this chapter.

Robust Control Theory (RCT) explained in the later sections will be presented alongside controller development for the SISO loops, that concerns with pitch, roll, yaw and altitude rate of the quadrotor. The three controller development strategies explained, namely MSO, LSDP and μ-synthesis which are based on minimizing the $\mathcal{H}_\infty$ norm of certain specified objective functions, enable the control engineer to to shape and modify plant characteristics in the frequency domain. For example, the $\mathcal{H}_\infty$ MSO S/T strategy minimizes the peaks of Sensitivity and Complementary Sensitivity functions of the closed loop, alongside formulating the regulation-tracking problem into frequency domain constraints; the LSDP on the other hand enables the designer to shape the open-loop singular values of the plant alongside stabilizing the plant against co-prime factor uncertainties.

While controller design and loop shaping within the frequency domain is helpful, laying out performance objectives of the closed loop in terms of time domain specifications (for example, rise time and settling time of step responses) is a practice that is more commonly seen throughout the control industry [93]. The mathematical approach of designing controllers such that the constraints and performance specifications, represented as a set of algebraic inequalities, form the basis of determining the controller parameters, is termed as Method of Inequalities (MOI) [159]. In [93], MOI is combined with robust design procedures to form a framework where both time and frequency domain specifications can be combined as requisites for the controller design. A numerical search routine called the Moving Boundaries Process (MBP) is employed for this technique.

The above version of MOI is modified in this study to formulate the same problem of forming a framework for defining time and frequency domain specifications, but instead, by resolving a constrained nonlinear optimization problem. Substantial changes are made to this former technique when it comes to MSO and μ-synthesis designs, while significant changes are brought about to the 1 and 2 DOF LSDP design technique. The 'Nonlinear Optimization with MADS algorithm' (NOMAD) [32] as provided by the MATLAB toolbox OPTI [160], is employed to develop this framework.

While various other nonlinear optimization routines that comes with OPTI MATLAB suite, such as the Particle Swarm Pattern Search Method (PSWARM) or the NLopt Nonlinear-Optimization Package (NLOPT) among others, can be used to resolve the optimization problem at hand, NOMAD helps to provide the fastest convergence to the global optimum. The specific nature of the optimization problem results in the tendency of solvers to converge to local optima depending on the initial point of the variables. Of all the solvers listed, NOMAD consistently proves to be the best solver both in terms of fastness of convergence as well as robustness towards the initial point of the variables when it comes to reaching the globa optimum. The algorithms developed for each different control strategy will be explained in the respective sections.

4.2 Decoupling the Quadrotor

In commercially available autopilots for quadrotors, the control system, generally resolves the position, attitude and altitude tracking problem by decoupling the model twice, that is, the x, y coordinate position tracking, is decoupled from attitude-altitude control, and further the attitude-altitude model is further decoupled into four SISO systems [161].

Attitude control generally employs a double loop PID controller [162] where two PID controllers are allocated for each SISO channel. While the outer loop aims at reducing the absolute values of attitude tracking error, the inner loop focuses on minimizing the tracking error of their rates.

Decoupling can be brought about by redefining the inputs (recall inputs originally in the multivariable model were angular velocities of individual rotors, see section 3.2.3.1) of the quadrotor as follows:

$$\begin{aligned} u_1 &= f_1 - f_2 - f_3 + f_4 \\ u_2 &= f_1 + f_2 - f_3 - f_4 \\ u_3 &= f_1 - f_2 + f_3 - f_4 \\ u_4 &= f_1 + f_2 + f_3 + f_4 \end{aligned} \tag{4.1}$$

where f_k represents the force exerted by the k_{th} propeller. We can see here that inputs are categorized according to the force required to perform the the four basic manoeuvres- roll, pitch, yaw and thrust, in contrast to the input propeller speeds (ω_k), as used to derive eq. (3.10). The equations of motion now become:

$$\begin{aligned} \dot{\phi}_i &= \dot{\phi}_b + (\dot{\theta}_b \, s_\phi + \dot{\psi}_b \, c_\phi) \, t_\theta \\ \dot{\theta}_i &= \dot{\theta}_b \, c_\phi - \dot{\psi}_b \, s_\phi \\ \dot{\psi}_i &= (\dot{\theta}_b \, s_\phi + \dot{\psi}_b \, c_\phi) \sec(\theta) \\ \ddot{x}_i &= -\frac{u_4}{m} (s_\phi \, s_\psi + c_\phi \, c_\psi \, s_\theta) \end{aligned}$$

$$
\begin{aligned}
\ddot{y}_i &= -\frac{u_4}{m}\,(c_\phi\, s_\psi\, s_\theta - c_\psi\, s_\phi) \\
\ddot{z}_i &= g - \frac{u_4}{m}\,(c_\phi\, c_\theta) \\
\ddot{\phi}_b &= \frac{1}{I_{xx}}\,(u_1\, d_y + (I_{yy} - I_{zz})\,\dot{\theta}_b\,\dot{\psi}_b) \\
\ddot{\theta}_b &= \frac{1}{I_{yy}}\,(u_2\, d_x + (I_{zz} - I_{xx})\,\dot{\phi}_b\,\dot{\psi}_b) \\
\ddot{\psi}_b &= \frac{1}{I_{zz}}\,(u_3\, d\,\frac{c_p}{c_t} + (I_{xx} - I_{yy})\,\dot{\phi}_b\,\dot{\theta}_b)
\end{aligned}
\tag{4.2}
$$

When Jacobian linearisation is performed at the equilibrium point $\bar{\mathbf{x}} = \mathbf{0_{1\times 12}}$ and $\bar{\mathbf{u}} = \left[\; 0\; 0\; 0\; mg \;\right]^T$, the state space model essentially decouples into four SISO models, each representing the system from input actions causing the craft to roll, pitch, yaw and change altitude to their corresponding desired rates, i.e the outputs. The linearised model of the quadrotor is given by:

$$
\mathbf{A_{12\times 12}} = \begin{pmatrix}
0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\
0 & 0 & 0 & 0 & -9.81 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 9.81 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0
\end{pmatrix}
$$

$$
\mathbf{B_{12\times 4}} = \begin{pmatrix}
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & 0 \\
0 & 0 & 0 & -0.7872 \\
3.434 & 0 & 0 & 0 \\
0 & 3.434 & 0 & 0 \\
0 & 0 & 1.621 & 0
\end{pmatrix}
$$

$$\mathbf{C_{4\times12}} = \begin{pmatrix} 0&0&0&0&0&0&0&0&0&1&0&0 \\ 0&0&0&0&0&0&0&0&0&0&1&0 \\ 0&0&0&0&0&0&0&0&0&0&0&1 \\ 0&0&0&0&0&0&0&0&1&0&0&0 \end{pmatrix} ; \ \mathbf{D_{4\times4}} = \begin{pmatrix} 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \\ 0&0&0&0 \end{pmatrix} \tag{4.3}$$

Inspecting the input matrix $\mathbf{B}$ reveals the successful decoupling of the quadrotor model. As for scaling the model, the diagonal weighting matrices $\mathbf{D_e}$, $\mathbf{D_d}$ and $\mathbf{D_r}$, described in section 3.3 remains the same except for $\mathbf{D_u}$, which becomes

$$\mathbf{D_u} = \begin{bmatrix} 9.6184 & 0 & 0 & 0 \\ 0 & 9.6184 & 0 & 0 \\ 0 & 0 & 9.6184 & 0 \\ 0 & 0 & 0 & 11.4744 \end{bmatrix}$$

Here the values are calculated based on the maximum expected change in inputs (forces, in contrast to angular velocities), from their steady state values.The following sections will examine algorithms required to design the controllers for the SISO plant model.

4.3 Robust Control Theory

This section introduces the key concepts involved in defining the robust control problem. The strategy explained in later sections will be developed on, from the theory presented in this section.

4.3.1 The $\mathcal{H}_\infty$ norm

In robust control, objectives are often defined in terms of minimizing the maximum value of a particular function, like for example the sensitivity function of the closed loop, over the range of frequency that is of interest for the problem at hand. The maximum value in this 'min-max' problem is determined by calculating the $\mathcal{H}_\infty$ norm of the objective function. The $\mathcal{H}_\infty$ norm of scalars, vectors and matrices encountered throughout the rest of this study are defined as:

- Scalars - For a transfer function, the $\mathcal{H}_\infty$ norm is defined as the peak value of the Bode magnitude plot. For example for the scalar sensitivity function S, $\|S\|_\infty = \max_\omega |S(j\,\omega)|$

- Vectors - The $\mathcal{H}_\infty$ norm of a vector containing scalars is simply the absolute value of the element with the largest magnitude, i.e. for the input vector $\mathbf{u}$, the infinity norm $\|\mathbf{u}\|_\infty = \max_i |u_i|$.
  When the vector contains transfer functions, the approach is slightly different. For example the $\mathcal{H}_\infty$ norm of the objective function of the weighted S/T MSO problem is given by:

$$\left\| \begin{matrix} w_P S \\ w_I T \end{matrix} \right\|_\infty \triangleq \max_\omega \sqrt{|w_P S|^2 + |w_I T|^2} \tag{4.4}$$

- Matrices - For a matrix containing scalars, the $\mathcal{H}_\infty$ norm is equal to the maximum singular value of the matrix. For example, the $\mathcal{H}_\infty$ norm of $\mathbf{D_u}$ the input scaling matrix, $\|\mathbf{D_u}\|_\infty = \bar{\sigma}(\mathbf{D_u})$, where $\bar{\sigma}$ denotes the maximum singular value of the matrix.
  The $\mathcal{H}_\infty$ norm of matrix transfer function is the maximum value calculated, over all frequencies, of the maximum singular value of the matrix. For example the system matrix $\mathbf{G}$ we have $\|\mathbf{G}\|_\infty = \max_\omega \bar{\sigma}(\mathbf{G}(j\omega))$.

In the time domain, $\mathcal{H}_\infty$ norm can be interpreted as the worst case gain of a proper stable system subjected to sinusoidal inputs at any frequency. The symbol $\mathcal{H}_\infty$ is also associated with the $\mathcal{H}_\infty$ space, usually called the 'Hardy' space (a concept first appearing in [163], also see [164] and [118], page 60), which consist of the set of all proper and stable transfer functions.

The $\mathcal{H}_\infty$ norm alongside being an induced norm, plays an important role in representation of unstructured model uncertainty, hence remains popular in robust control theory.

4.3.2 The General Robust Control Problem

The method of formulating control problems into the structure shown in Fig. 4.1 first appeared in [165]. The objective function of such a formulation is to develop a controller $\mathbf{K}$ that minimizes the $\mathcal{H}_\infty$ norm of the transfer function between exogenous input signals $\mathbf{w}$ to the generalized plant $\mathbf{P}$, to outputs $\mathbf{z}$, using an input command $\mathbf{u}$ and measurement signal $\mathbf{v}$ that drives the controller. Here $\boldsymbol{\Delta}$ represents the model uncertainty with $\mathbf{y}_\Delta$ and $\mathbf{u}_\Delta$ representing its outputs and inputs respectively.

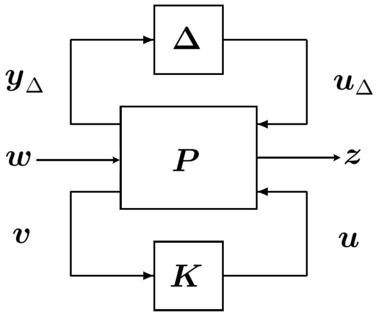


Figure 4.1: The general robust control problem formulation

The block diagram in Fig. 4.1 can be interpreted using the below set of equations:

$$\begin{gathered}\begin{bmatrix}\mathbf{y}_\Delta\\ \mathbf{z}\\ \mathbf{v}\end{bmatrix} = \mathbf{P}\begin{bmatrix}\mathbf{u}_\Delta\\ \mathbf{w}\\ \mathbf{u}\end{bmatrix}\\ \mathbf{u} = \mathbf{K}\mathbf{v}\\ \mathbf{u}_\Delta = \mathbf{\Delta}\mathbf{y}_\Delta\end{gathered} \tag{4.5}$$

The transfer function between $\mathbf{w}$ and $\mathbf{z}$ is derived by defining the relationships of the blocks in terms of Linear Fractional Transformations (LFTs). Accordingly we have the following relationship equations.

$$\begin{aligned}\mathbf{N} &= F_l(\mathbf{P},\mathbf{K}) \triangleq \mathbf{P}_{11} + \mathbf{P}_{12}\,\mathbf{K}(\mathbf{I}-\mathbf{P}_{22}\mathbf{K})^{-1}\,\mathbf{P}_{21}\\ \mathbf{F} &= F_u(\mathbf{N},\mathbf{\Delta}) \triangleq \mathbf{N}_{22} + \mathbf{N}_{21}\mathbf{\Delta}\,(\mathbf{I}-\mathbf{N}_{11}\mathbf{\Delta})^{-1}\,\mathbf{N}_{12}\end{aligned} \tag{4.6}$$

where $\mathbf{N}$ is the transfer function from $\mathbf{w}$ to $\mathbf{z}$ when $\mathbf{\Delta}$ equals zero, F_l the lower LFT and F_u the upper LFT.

4.3.3 Toolboxes and Solvers

Alongside the Control System Toolbox, the Symbolic Math Toolbox and other general toolboxes in MATLAB, the work presented in this study relies on the following:

- Robust Control Toolbox: Development of controllers using MSO, LSDP and μ-synthesis strategies, is enabled by calling routines from the Robust Control Toolbox [166].
- OPTI Toolbox : OPTI is a MATLAB toolbox that provides tools to resolve optimization problems. OPTI, which provides access to a suite of linear and non-linear optimizers, through a MEX interface that enables calling the routines from within MATLAB effortlessly [167], remains a popular toolbox for

resolving optimization problems among researchers [168–171].

The MATLAB Optimization Toolbox [172], similar to OPTI, is a widely popular optimization toolbox among researchers. Although it gives access to a number of optimization routines, fails to provide access to the optimization algorithm, NOMAD which is used in this study to form the optimization framework. Hence the framework, that enables the search for a controller that satisfies both time domain and frequency domain specifications, is developed using the OPTI toolbox. The optimization algorithm chosen to resolve the problem- NOMAD [32], is a black-box optimization routine, that can handle linear and nonlinear constraints while working with non-convex, derivative free optimization problems [169]. NOMAD uses the Mesh Adaptive Direct Search Algorithm [173], a direct search routine that uses non-smooth Calculus developed in [174]. Among various advantages of NOMAD as mentioned in [175] the global convergence of the solver using search routines such as Variable Neighborhood Search algorithm [176] among others, enables the solver to evade settling down at local optima and hence have found to be extremely helpful in resolving the optimization problem at hand compared to other solvers. The following sections examines in detail, each of the different algorithms employed to design the controller.

4.4 Mixed Sensitivity Optimization

The one degree of freedom negative feedback control system is presented in Fig. 4.2.

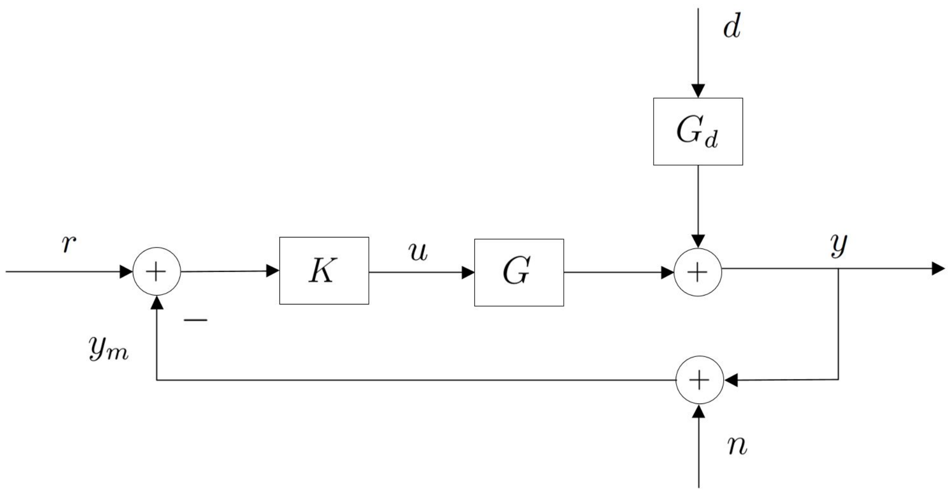


Figure 4.2: The one degree of freedom negative feedback control system.

Following the usual naming conventions, we have "*r*" representing the reference input, "*u*" the plant input, "*d*" the disturbance signal, "*y*" the plant output, "*n*" the measurement noise and "y_m" the measured output. The error signal "e" is defined as the difference between the actual output and the reference signal (in contrast to measured output "y_m" and reference signal), i.e. $e = y - r$.

From analysing Fig. 4.2 the following expressions can be derived:

$$y = T\,r + S\,G_d\,d - T\,n \tag{4.7}$$

$$e = -S\,r + S\,G_d\,d - T\,n \tag{4.8}$$

$$u = K\,S\,r - K\,S\,G_d\,d - K\,S\,n, \tag{4.9}$$

where we have the sensitivity function S and complementary sensitivity function T defined as

$$S = \frac{1}{1 + G\,K}\,, \quad T = \frac{G\,K}{1 + G\,K} \tag{4.10}$$

From eq. (4.7) we can see that, shaping the complementary sensitivity function such that its magnitude is close to unity around the bandwidth frequencies (in order for the output to match the reference signal), and rolls down quickly at higher frequencies (in order to eliminate effects from the noise signal), is important for resolving the tracking problem.

Similarly, from eq. (4.8), it can be deduced that in order to obtain a small error signal, the magnitude of the sensitivity function should be small at lower frequencies to curtail the effect of reference inputs, which generally operate at the lower end of the frequency spectrum. Alongside this, the magnitude of complementary sensitivity should be small at high frequencies to curtail the effects of high frequency noise on the system. This requirement in turn translates to shaping the magnitudes of closed loop functions S and T in the frequency domain utilizing weights.

4.4.1 The S/T Problem

Mixed sensitivity optimization S/T technique caters exactly to these above stated requirements. The S/T MSO problem can be developed from the Fig. 4.3.

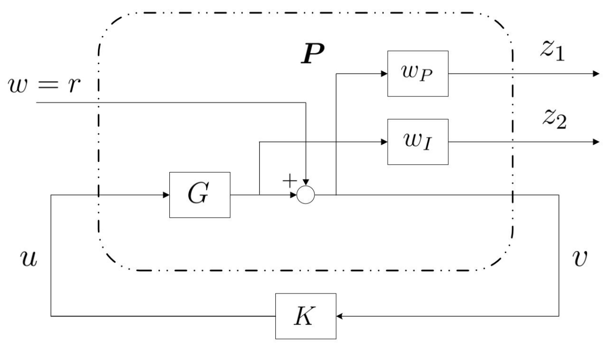


Figure 4.3: The S/T mixed sensitivity optimization problem

The objective function of the S/T MSO problem can be stated as to develop a controller that simultaneously stabilizes the closed loop and at the same time minimizes the $\mathcal{H}_\infty$ norm of the transfer function between output signal $\mathbf{z} \triangleq [z_1 \; z_2]^T$ and the input signal w. This transfer function can be expressed as the lower fractional linear transformation, $F_l(\mathbf{P},\, K)$. The system matrix $\mathbf{P}$ is given by:

$$\mathbf{P} = \left[\begin{array}{c:c} w_P & -w_P\, G \\ 0 & w_I\, G \\ \hdashline I & -G \end{array}\right] \tag{4.11}$$

where the dashed line, partitions the matrix such that P_{22} is compatible with K. Here w_P stands for the performance weight and is associated with shaping the sensitivity function S, and w_I is the weight associated with complementary sensitivity function T. The subscript 'I' represents the choice of considering the model uncertainty at inputs, rather than outputs (See section 4.4.2). G represents the plant model that is required to be controlled. The cost function of the S/T MSO problem is given by:

$$\gamma = \left\| \begin{matrix} w_P S \\ w_I T \end{matrix} \right\|_\infty \triangleq \max_\omega \sqrt{|w_P S|^2 + |w_I T|^2} \tag{4.12}$$

Designing the weights play an important role in obtaining a controller that is robust against model uncertainty. Although there are guidelines that can be followed while developing these weights (such that they shape the closed loop functions as required), in general a significant amount of fine tuning, that involves trial and error, is generally required to reach a satisfactory performance [93]. The following section presents a deterministic general procedure, that standardizes the development of the frequency weights, alongside the controller that satisfies the design constraints for a SISO system.

4.4.2 Developing the design weights w_I and w_P

Developing the weights, as noted in [93], can be formulated into the following problem. For the given control problem in Fig. 4.3 find $\tilde{W} = (w_I, w_P)$ such that

$$\gamma(\tilde{W}) \leq \epsilon_\gamma, \tag{4.13}$$

and

$$\Phi_i(\tilde{W}) \leq \epsilon_i, \tag{4.14}$$

while maintaining design optimality. Here $\gamma(\tilde{W})$ equals the $\mathcal{H}_\infty$ norm of eq. (4.12). The vector $\mathbf{\Phi}$ contains closed loop time domain functionals that are required to be constrained. These may include, but are not limited to, step response characteristics such as rise time, settling time or peak overshoot. The variables ϵ_γ and ϵ_i are values that represent the desired bounds on $\gamma(\tilde{W})$ and $\Phi_i(\tilde{W})$ receptively.

The development of $\tilde{W}$ is a two stage process. In the first stage the complementary sensitivity weight w_I is designed such that it captures the model uncertainty. In stage two, the sensitivity weight w_P is designed alongside the controller.

4.4.2.1 Stage 1: Capturing model uncertainty using w_I

Model uncertainty can be represented in various forms such as additive, multiplicative, inverse additive, and inverse multiplicative uncertainty. Uncertainty can also be classified based on its location being either at the input of the plant or at its outputs. The type of uncertainty can be chosen depending on the application and the plant model. In this study, we have chosen the uncertainty to be multiplicative input uncertainty, which gives the perturbed plant model $G_p = G(1 + w_I\Delta)$, where the subscript 'p' stands for 'perturbed' and 'I' indicates that the perturbations taken into account are present at the input of the plant. This is depicted in Fig. 4.4.

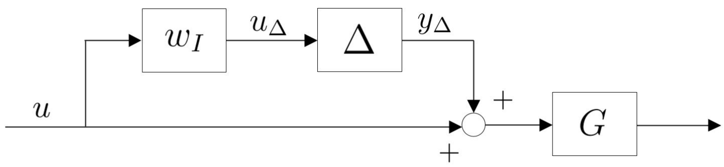


Figure 4.4: Multiplicative input uncertainty.

Here Δ represents any stable transfer functions with a magnitude less than or equal to 1 over all frequencies, i.e. $|\Delta(j\omega)| \leq 1 \,\forall\, \omega$. To develop w_I we first calculate l_I at each frequency as below:

$$l_I(\omega) = \max_{G_p \in \prod} \left| \frac{G_P(j\omega) - G(j\omega)}{G(j\omega)} \right| \tag{4.15}$$

The complementary sensitivity weight is then selected such that $|w_I(j\omega)| \geq l_I(j\omega) \,\forall\, \omega$. Here $\prod$ represents the set of all possible plants developed using the uncertainty descriptions in Table 3.2. The designed weights are required to be stable, minimum phase, real and rational transfer functions.

4.4.2.2 Stage 2: Developing w_P alongside the controller

The sensitivity weight w_P as seen in Fig.4.3 affects the outputs and hence the performance of the closed loop. Here the subscript 'P' stands for performance. The sensitivity weight needs to be parametrised as below, before the algorithm that resolves the non-linear optimization problem can be applied,

$$w_P = \frac{w_1(s + w_2)}{s + w_3}. \tag{4.16}$$

The initial values of the scalar parameters w_1, w_2 and w_3 are chosen such that $w_2 > w_3$ as w_P in most applications is chosen to be a high pass filter. Note that other choices are also possible here for the format of w_P (see equation 4.18). A crucial step in obtaining a successful convergence for the optimization problem while satisfying the constraints and resolving optimality, is the design of an efficient cost function. This is explained in the next section.

4.4.3 Choice of the cost function, $\mathcal{J}$.

The proposed cost function $\mathcal{J}$, which is a function of the controller K and design weight $\tilde{W}$ is given by

$$\mathcal{J}(K, \tilde{W}) = \underbrace{W1 \cdot \gamma}_{\mathcal{J}_1 \text{ - Robustness}} + \underbrace{W2 \cdot \mathrm{ITSE} + W3 \cdot \mathrm{J_u}}_{\mathcal{J}_2 \text{ - Optimality}} + \underbrace{W4 \cdot \texttt{order}(K)}_{\mathcal{J}_3 \text{ - Implementation cost}} \tag{4.17}$$

The cost function has the three different parts of it. These are listed below:

1. $\mathcal{J}_1$ - The first part is a function that reflects a penalty on the lack of robustness of the design which can take different forms. In this section this is represented by γ, the MSO frequency domain cost function that represents $\mathcal{H}_\infty$ norms of closed loop functionals, multiplied by a weight. Hence $\mathcal{J}_1 = W1 \cdot \gamma$

2. $\mathcal{J}_2$ - The second part is concerned with the optimality of the designed controller. This can be function that puts a penalty on tracking error and input usage. It is defined as $\mathcal{J}_2 = W2 \cdot \mathrm{ITSE} + W3 \cdot \mathrm{J_u}$, where ITSE indicates the Integral Time Square Error, while $\mathrm{J_u}$ represents the area under the input curve, which acts as a handle towards limiting the input usage.

3. $\mathcal{J}_3$ - The third part is an optional part which penalizes the cost of implementation of the controller. Robust controllers in general require large orders which causes difficulties in implementation and usually requires an extra step of model reduction. A penalty on the order of the controller thus helps to penalize this implementation cost. Hence this third part is defined as $\mathcal{J}_3 = W4 \cdot \texttt{order}(K)$.

4.4.4 Algorithm to develop the Controller

Now that the cost function is defined, the constrained nonlinear optimization problem to develop the controller and the sensitivity weight can be formulated. **Algorithm #1** gives the procedure to develop the sensitivity weight alongside the controller.

Algorithm # 1 Sensitivity weight and Controller design procedure for S/T MSO

Inputs: Plant model G, design constraints, model uncertainty estimate.
Outputs: K, w_I, w_P.

1. Scale the given plant G according to the criteria described in section 3.3
2. Calculate the complementary sensitivity weight w_I according to the procedure described in section 4.4.2.1.
3. Define the appropriate form of w_P (see equation 4.16,) and choose the initial values of the parameters.
4. Define the cost function $\mathcal{J}$ and initializes the weights W1 through W4.
5. Develop the vector $\boldsymbol{\Phi}$ which contains the following nonlinear closed loop functionals
   1. Maximum overshoot, M_p
   2. Rise time, t_p
   3. Settling time, t_s
   4. Maximum value of input, $u_{\max}$
   5. gamma, γ

   Here M_p, t_p and t_s, the step response characteristics and u_{max} are the time domain specifications while γ refers to the frequency domain specification.
6. Define the frequency domain bounds ϵ_γ and time domain specifications limit vector $\boldsymbol{\epsilon}$.
7. Solve the constrained non-linear optimization problem to obtain controller K, alongside the weights. This is achieved using the OPTI MATLAB toolbox. The black box optimizer NOMAD is chosen as the optimization routine. Nonlinear constraints are defined based on ϵ_γ and $\boldsymbol{\epsilon}$ for the vector $\boldsymbol{\Phi}$ that defines the closed loop functionals.
8. Depending on robustness requirements, modify the weights $W1$ through $W4$ to reach the required performance levels.
9. If the performance specifications are not met, decrease the % uncertainty considered while defining w_I and relax the bounds defined in step 6 and retry step 7.

Algorithm #1 can be implemented in commercially available software such as MATLAB that support optimization packages aforementioned. On successful completion the sensitivity weight w_P is obtained alongside the controller. While implementing the algorithm in MATLAB the following points should be noted.

1. During the initial runs, both the time and frequency domain specifications may need to be relaxed for pragmatic and numerical reasons by the control

engineer. This gives the optimizer enough leeway to generate search points and decreases the significance of the parameter initialization. As successful iterations are completed the specifications can be tightened.

2. The Robust Control Tool box routine `mixsyn`, is called in step 7 of **Algorithm #1** to develop the MSO controller.

3. The sensitivity weight w_P can also be modeled as below:

$$w_P = \frac{(s/M^{1/n} + \omega_B^*)^n}{(s + \omega_B^* A^{1/n})^n} \tag{4.18}$$

   with the parameters M, A representing the upper bounds on closed loop sensitivity $|S|$ at higher and lower frequencies respectively, and ω_B^* representing the bandwidth frequency. This format is usually used by control engineers while designing the weight by trial and error. As an alternative, the three parameters can also be searched for using **Algorithm #1**. The parameter n, which takes integer values, can be initialized with unity. If higher performance is required (for example, if we may need a steeper slope for the sensitivity, or the loop function below the bandwidth), its value can be increased.

4. ITSE has been used in step 4 while framing the cost function. Compared to the various other error performance indices namely IAE (Integral Absolute Error), ITAE (Integral Time Absolute) and ISE (Integral Square Error), ITSE has been found to facilitate the fastest convergence for the problem at hand.

Based on **Algorithm #1** a simple graphical depiction of the inner and the outer framework is represeted in Fig. 4.5. The darker shade of white represents the outer framework and the $\mathcal{H}_\infty$ problem form the inner framework.

The optimizer NOMAD is implemented in the outer framework encapsulating the $\mathcal{H}_\infty$ problem. Inside the outer framework the $\mathcal{H}_\infty$ problem is run continuously several times until a convergence to the global minimum of the cost function is reached. In case the optimizer failure to converge the acceptable limits of the model uncertianty might need to be revisited before the optimization problem can be rerun. This generic graphical depiction will be used a basis for development of all the other algorithms (related to the other branches of robust control) going forward.

Once the controllers are developed, the performance can be measured both in the frequency and time domain. The performance criteria is explained in the following subsection.

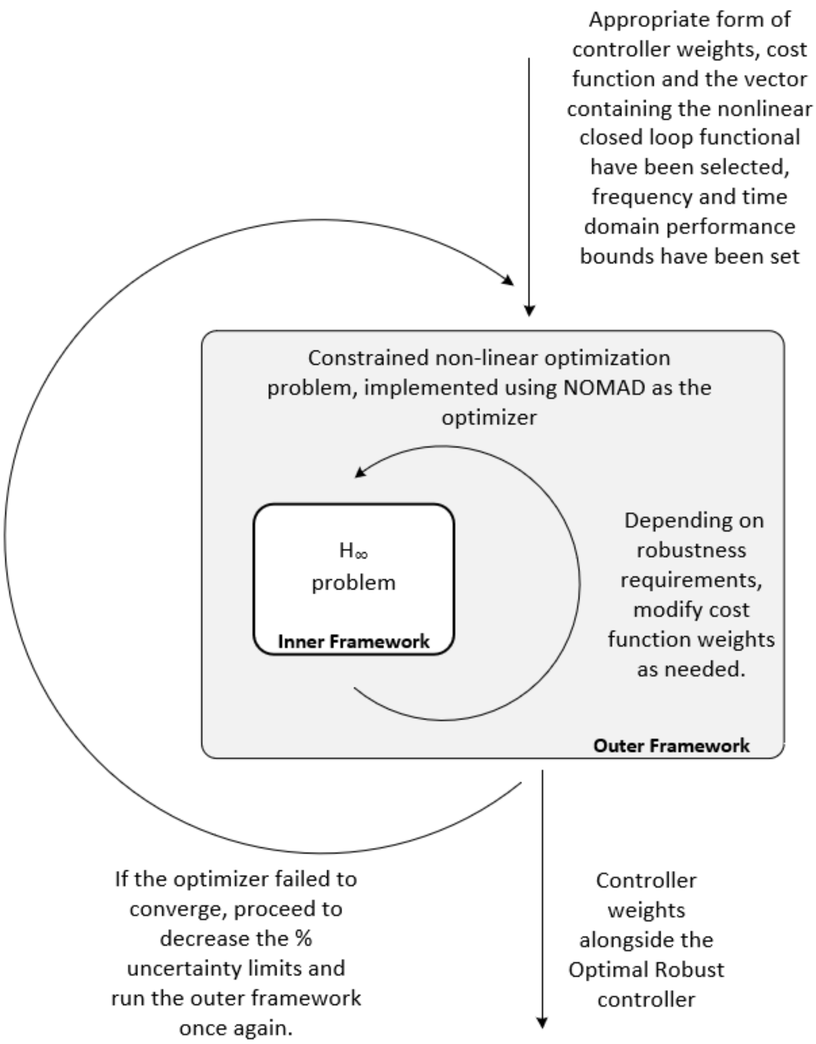


Figure 4.5: Graphical representation of the inner and outer framework of the control problem.

4.4.5 Performance Analysis Criteria

Following the procedures listed in 4.4.2, the effectiveness of the controller can be measured in both the frequency and time domain. In the frequency domain the performance criteria in the Table 4.1 can be used to analyse the closed loop stability.

Table 4.1: Performance Criteria

Performance Criteria	Necessary Constraint
Nominal Stability	Closed loop nominally stable
Nominal Performance	$\lvert w_P S\rvert < 1;\ \forall\omega$
Robust Stability	$\lvert w_I T\rvert < 1;\ \forall\omega$
Robust Performance	$(\lvert w_P S\rvert + \lvert w_I T\rvert) < 1;\ \forall\omega$

The nominal performance, robust stability and robust performance criteria are developed by measuring the distance of the closed loop of the nominal plant and uncertain model, from -1 in the Nyquist plot (for details refer Ch. 7, [118]).

To assess the time domain performance, step response characteristics of the nominal nonlinear plant can be analyzed. The step response of the extreme plants can also be plotted to check whether the response settles down towards the reference values. The extreme plants are generated using parameters at the vertices of the hypercube formed using values from Table 3.2.

This brings us to the end of the design procedure of MSO controllers. In the following section controllers are developed using this procedure, for attitude-altitude rate tracking control of the quadrotor, and performances are analyzed.

4.4.6 A Quadrotor Control Application

The controllers developed using the theory discussed in section 4.4, are presented in this section. A controller is developed for each of the four SISO loops described in section 4.2.

The design specifications for each loop of the quadrotor presented in section 3.5 are utilized to develop the optimization problem which forms the outer framework that designs the controller alongside the sensitivity weight w_P. The development of the complementary sensitivity weights w_I, and sensitivity weights w_P are presented next, followed by the controllers and the performance analysis. Parametric uncertainty based on values from Table 3.2 is considered as the only source of uncertainty for plants present in the uncertain plant array $\prod$. This helps to keep the design problem simple during this initial phase of design. Non-linearities and time delay uncertainties will be discussed later in Chapter 5 where MIMO controllers will be presented.

4.4.6.1 The Complementary Sensitivity weight (w_I)

The complementary sensitivity weight w_I developed based on eq. (4.15) is presented in Fig. 4.6. The relative error $l_I(j\omega)$, given by

$$l_I(j\omega) = \frac{G_P(j\omega) - G(j\omega)}{G(j\omega)}, \tag{4.19}$$

is calculated for all the uncertain plants in $\prod$. The light blue contours represent this relative error $l_I(j\omega)$, while the red line represents w_I (refer to section 4.4.2.1 for details).

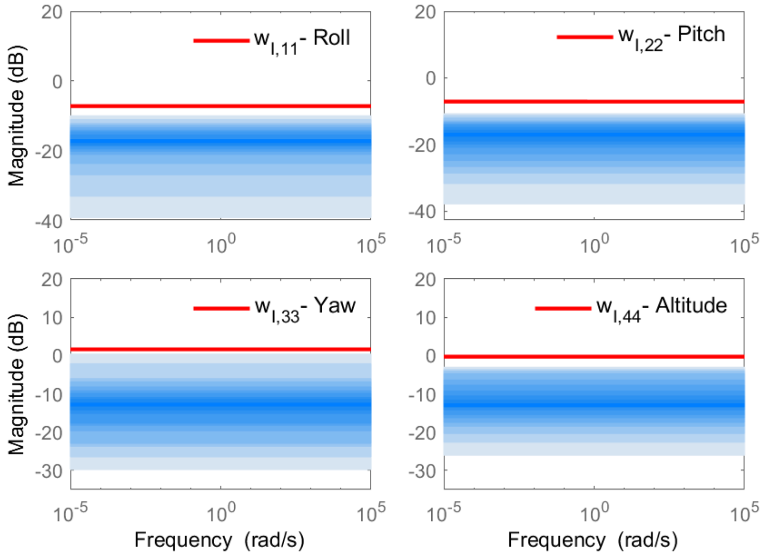


Figure 4.6: Bode magnitude plots of multiplicative input uncertainty and relative error between uncertain and nominal plant (Note $w_{I,11} = w_{I,22}$).

Since the SISO nominal model from input u_1 to roll rate y_1 is exactly the same as that from input u_2 to pitch rate y_2, the model uncertainty for both SISO models are captured by the same weight.

The individual weights $w_{I,jk}$ designed for the uncertain plant $G_p = G(1 + w_{I,jk}\Delta)$, where j and k represents the corresponding inputs and outputs clearly satisfies the requirement $|w_I(j\omega)| \geq l_I(j\omega)$ as seen from the plots (as the red line in the plot always falls above the blue lines of the density plot).

The developed weights are given below:

$$\begin{gathered} w_{I,11} = w_{I,22} = \frac{0.4393s + 0.4374}{s + 0.9954}, \\ w_{I,33} = \frac{1.209s + 1.204}{s + 0.9954} \quad , \quad w_{I,44} = \frac{0.9692s + 0.965}{s + 0.9954}. \end{gathered} \tag{4.20}$$

4.4.6.2 Optimization Problem

Using both designs specification presented in section 3.5 as well as based on performance requirements given in Table 4.1, the following Optimization problem is constructed.
For a scaled unit step reference input, minimize $\mathcal{J}(K, \tilde{W}) = \sum_{k=1}^{4} \mathcal{J}_k$ (see eq. (4.17)) subject to nonlinear constraints in Table 4.2 (developed based on the design speci-

fications listed in section 3.5).

Table 4.2: Nonlinear Constraints of the optimization problem

Nonlinear Constraints	Roll & Pitch Rate	Yaw Rate	Altitude Rate
Closed loop % Overshoot	≤ 5	≤ 5	≤ 20
Closed loop Rise Time (s)	≤ 0.1	≤ 0.1	≤ 0.3
Closed loop Settling Time (s)	≤ 0.25	≤ 0.3	≤ 1.25
Control effort $\|u(t)\|$	≤ 1	≤ 1	≤ 1
γ	≤ 1	≤ 1	≤ 1

where γ is the stacked mixed $\mathcal{H}_\infty$ MSO cost function (see eq. (4.12)).

The upper and lower bounds for parameters forming the sensitivity weight of the optimization problem arise primarily from the structure of eq. (4.16) where, we begin the design search with initial state of $w_2 \geq w_3$. The bounds for the variables must be such that the transfer function developed is stable and proper.

The optimization problem formed as a result is utilized in the development of the outer frame work using which the sensitivity weight and subsequently the controller is designed.

4.4.6.3 The Sensitivity weight (w_P)

The sensitivity weights developed using the framework presented in **Algorithm #1**, are listed below:

$$\begin{aligned} w_{P,11} &= w_{P,22} = 0.8861 \frac{s + 32.6141}{s + 0.7366} \\ w_{P,33} &= 0.0378 \frac{s + 174.7582}{s + 0.4198} \quad , \quad w_{P,44} = 0.538 \frac{s + 1.9591}{s + 0.006676} \end{aligned} \tag{4.21}$$

The time and frequency domain specifications obtained are presented in Table 4.3

Table 4.3: Time response characteristics and frequency domain specification of the linear closed loop model.

Closed Loop Constraints	Roll & Pitch Rate controller	Yaw Rate controller	Altitude Controller
% Overshoot	5.000	0	3.294
Rise Time (s)	0.025	0.055	0.206
Settling Time (s)	0.250	0.120	1.135
u_{max}	0.418	0.712	1.000
γ	0.893	**1.221**	0.979

The normalized reference signal employed to generate the response has a step size of unity. Apart from controller design for the yaw rate loop we have successful controller designs, i.e, convergence of the optimizer for roll, pitch and altitude rate loops. From the step response characteristics obtained from the initial runs, we could see that the roll and pitch controller induces an overshoot of 5.00% of the reference step input, and that of around 3.29% in the altitude rate loop. We have the maximum control effort u_{max}, (normalized to unity) for roll, pitch and altitude rate loops within the desired limit of ≤ 1.

In the case of the controller of the yaw rate loop, the optimizer failed to converge. The best possible result with the least number of constraint violations as obtained by NOMAD, are presented in the second column of Table 4.3. The constraint violation results from the frequency specification γ (the cost function from eq. (4.12)), of the yaw rate loop exceeding 1, which is not desirable and needs to be addressed.

To address the issue with the yaw controller % uncertainty (Table 3.2) is modified such that uncertainty in mass is reduced to 40% (from 50%) and the thrust coefficient c_t is reduced to 38% (from 50%). The satisfactory results obtained after tuning are given in Table 4.4.

Table 4.4: Time and frequency domain specifications after controller tuning.

Closed Loop Functionals	Yaw Rate Controller
% Overshoot	1.00
Rise Time (s)	0.05
Settling Time (s)	0.26
u_{max}	0.68
γ	**0.926**

The modified sensitivity and complementary sensitivity weights are given below:

$$w_{I,33} = \frac{0.9165s + 0.9126}{s + 0.9954} \quad , \quad w_{P,33} = 0.0315\,\frac{s + 173.1726}{s + 0.4895} \tag{4.22}$$

Inspecting the tuned results presented in Table 4.4 the following points can be noted:

1. Changing the uncertainty in mass and thrust coefficient affects only the development of the yaw controller, this is because, during the development phase, decoupled loops (in this case the yaw rate loop) are individually considered to develop the corresponding SISO controller. In the practical setting, this change can be explained as follows: With the given configuration of the quadrotor (i.e. the parameters used), when faced with an uncertainty of 50% in mass and that of 50% in thrust coefficient while robust stability and acceptable performance can be expected from the roll, pitch and altitude controllers, the performance of yaw controller might be poor. Subsequently if (say, by the limiting changes to the payload mass) the uncertainty in mass is limited to 40% and that of the thrust coefficient to less that 38% (by changing the propeller material) robust stability and acceptable performance can be guaranteed.

2. The new yaw controller achieves a γ that is less than unity at expense of an increase in settling time and appearance of a slight overshoot.

The above example demonstrates that the framework can be used effectively to tune the controller by both making changes to the cost function, as well as by reducing the model uncertainty captured by the sensitivity weight. The effectiveness of the controllers are examined in the following sections.

4.4.6.4 Robust stability analysis

The performance criteria presented in Table 4.1 is analyzed in this section. Nominal stability is guaranteed if closed loop system is nominally stable. The classical closed loop stability margins, corresponding crossover frequencies and closed loop stability of the loops are presented in the Table 4.5.

The plant and the controllers are discretized at a frequency of 100 Hz before the stability analysis is carried out. The sampling frequency was selected as this has been found to be a reasonable choice, in both academic and commercial environments [177–180].

Table 4.5: Classical Stability Margins and Crossover Frequencies

Stability Margins	Roll & Pitch Rate Loop	Yaw Rate Loop	Altitude Rate Loop
Gain Margin (abs)	2.781	4.206	13.641
GM Frequency (rad/s)	105.148	84.806	59.816
Phase Margin (rad)	1.047	1.128	1.293
PM Frequency (rad/s)	36.285	23.478	7.193
Delay Margin (s)	0.029	0.048	0.180
DM Frequency (rad/s)	36.285	23.478	7.193
Closed loop Stability	Stable	Stable	Stable

The gain margins of >2 and phase margins of >0.525 rad (30°) as is the case for the designed controllers is considered acceptable for practical engineering problems. The delay margin (row 5) gives the maximum amount of delay that can be tolerated by the pitch/roll rate, yaw rate and the altitude rate loops respectively before they become unstable.

Having established nominal stability and examined the closed loop stability margins, nominal performance, robust stability and performance can be assessed. According to the criteria presented in Table 4.1 the values of $|w_P S|$, $|w_I T|$ and $|w_P S| + |w_I T|$ should be less than unity at all frequencies for the system to achieve nominal stability, robust stability and performance. The infinity norms of these weighted closed loop functionals are given in the next table.

Table 4.6: Robust Stability Analysis

∞-Norm	Roll & Pitch Rate Loop	Yaw Rate Loop	Altitude Rate Loop
$\|w_P S\|_\infty$	0.890	0.223	0.611
$\|w_I T\|_\infty$	0.431	0.898	0.999
$\|(\lvert w_P S\rvert + \lvert w_I T\rvert)\|_\infty$	1.213	1.121	1.227

The values are calculated based on the curves presented in Fig. 4.7. On inspecting values in the Table 4.6 and Fig. 4.7, it is evident that except for $|w_P\,S| + |w_I\,T|$, the weighted sensitivity and complementary sensitivity values fall below unity at frequencies within the area of interest (which for the quadrotor application is taken to be within 10^{-4} rad/s to 10^4 rad/s, as in this range the dynamics/characteristics of the plant at steady state, bandwidth and high frequency can be observed comfortably). This satisfies the criteria for achieving nominal performance and robust stability.

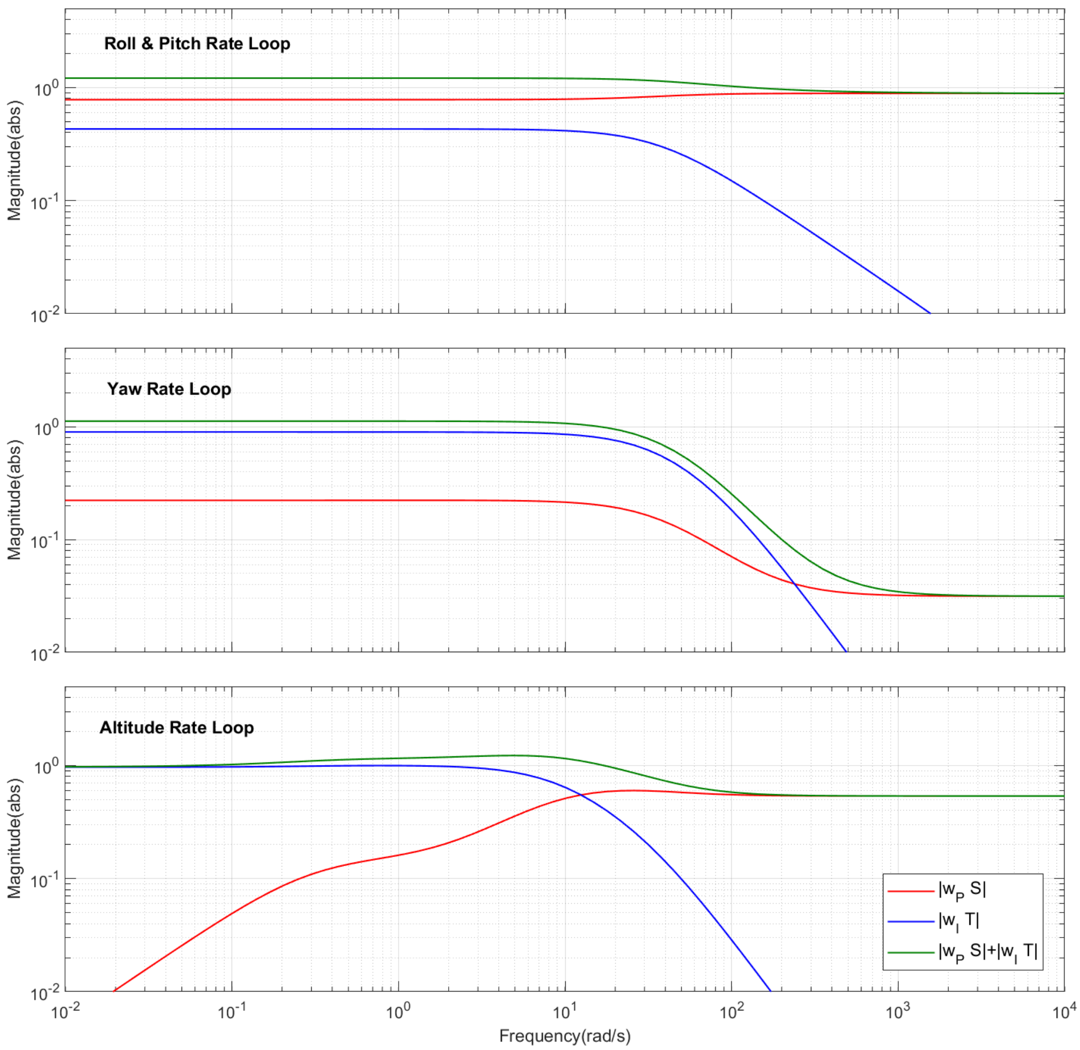


Figure 4.7: Robust stability analysis: Bode magnitude plots of $|w_P\,S|$, $|w_I\,T|$ and $(|w_P\,S| + |w_I\,T|)$

With regards to the robust performance criteria (row 4) in Table 4.1, the following condition can be taken into account:

$$\max_{\omega}\left(|w_P\,S| + |w_I\,T|\right) \;\leq\; \sqrt{2}\left\|\begin{matrix} w_P S \\ w_I T \end{matrix}\right\|_{\infty} \tag{4.23}$$

Refer to [118] pg. 531, for the proof of equation 4.23. This explains the reason why although the value of γ (i.e. the stacked infinity norm in eq. (4.23)) had been constrained to be less than one, robust performance criteria is violated (i.e. $\|(|w_P\, S|+|w_I\, T|)\|_\infty > 1$ as seen in row 3, Table 4.6). Nevertheless, since the values are close to unity, acceptable performance is to be expected.

4.4.6.5 Closed loop bandwidth

The bandwidth ω_B can be defined as the range of frequencies over which control is effective. In terms of closed loop sensitivity this translates to the frequency where $|S(j\omega)|$ first crosses $1/\sqrt{2}$ from below.

(Note: In literature bandwidth definitions have differed slightly among disciplines of study, chosen mostly from between ω_B based on the magnitude of Sensitivity function, ω_c the *gain crossover frequency*, and ω_{BT} based on the magnitude of Complimentary Sensitivity function.

From the closed loop control system presented in Figure 4.2 and having equations 4.7 to 4.9 derived in terms of S and T, we can see that for tracking performance calculations, we have

$$y = T\, r \quad \text{and} \quad e = -S\, r. \tag{4.24}$$

For simplicity the disturbance and noise terms are ignored. Without control, we have T equal to zero and for effective tracking we need the magnitude of T to be atleast over $1/\sqrt{2}$ ($\approx$ -3dB), with bandwidth ω_{BT} defined as the highest frequency at which $|T(j\omega)|$ crosses $1/\sqrt{2}$ from above.

While this definition might suffice in most disciplines, in feedback control in certain cases like that of the inverse response based system presented in [118] (see page. 40), between the the frequency of ω_B and ω_{BT}, the phase of T might drop substantially, even to the point where tracking becomes out of phase, resulting in poor control performance. Hence when considering complimentary sensitivity to define bandwidth, both the magnitude and phase of T will need to considered.

On the other hand for good tracking as shown in eq. (4.24), to attain minimal tracking error, the Sensitivity function should be close to zero. Hence, irrespective of the phase of S if the magnitude $|S|$, is close to zero, good performance tracking can be achieved. Hence for the feedback control problems discussed in this study we will use the definition of bandwidth ω_B based on the Sensitivity function.)

The bandwidths calculated according to this definition is listed in Table 4.7.

As such we can see that the fastest response in terms better rise times, can be expected in the roll and pitch rate loops, the yaw rate loop can be expected to respond a bit slower and the altitude rate would exhibit the slowest response among the three loops. This is very desirable for a quadrotor as it is the pitch and roll loops that is mainly responsible for stabilizing the quadrotor against disturbance signals, while yaw and altitude loops are responsible for way point tracking.

Table 4.7: Closed Loop Bandwidth

Loop	ω_b (rad/s)
Roll & Pitch Rate	37.6
Yaw Rate	20.2
Altitude Rate	6.125

The values in Table 4.7 are generated from Fig. 4.8. Inspecting the Fig. 4.8 in the context of eqs. (4.7) to (4.9) the following insights can be derived:

1. The absolute value of sensitivity $|S|$ has small values at low frequencies. This contributes towards both decreasing the impact of disturbance signals on output, as well as lowering magnitude of error at low frequencies. It also contributes towards decreasing the overall magnitude of the input signal at these frequencies.

2. Similarly the magnitude of complementary sensitivity $|T|$ has a small magnitude at high frequencies, hence it contributes towards both decreasing the impact of noise both at the output, as well as in lowering the magnitude of the overall error signal

3. The loop transfer function L assumes larger magnitudes at low frequencies. Around bandwidth frequencies, the magnitude rolls down at a constant rate. The roll off increases further at higher frequencies. The behavior is consistent for all three controllers. This loop shape enables better performance at lower frequencies in terms of tracking and increased noise reduction at high frequencies.

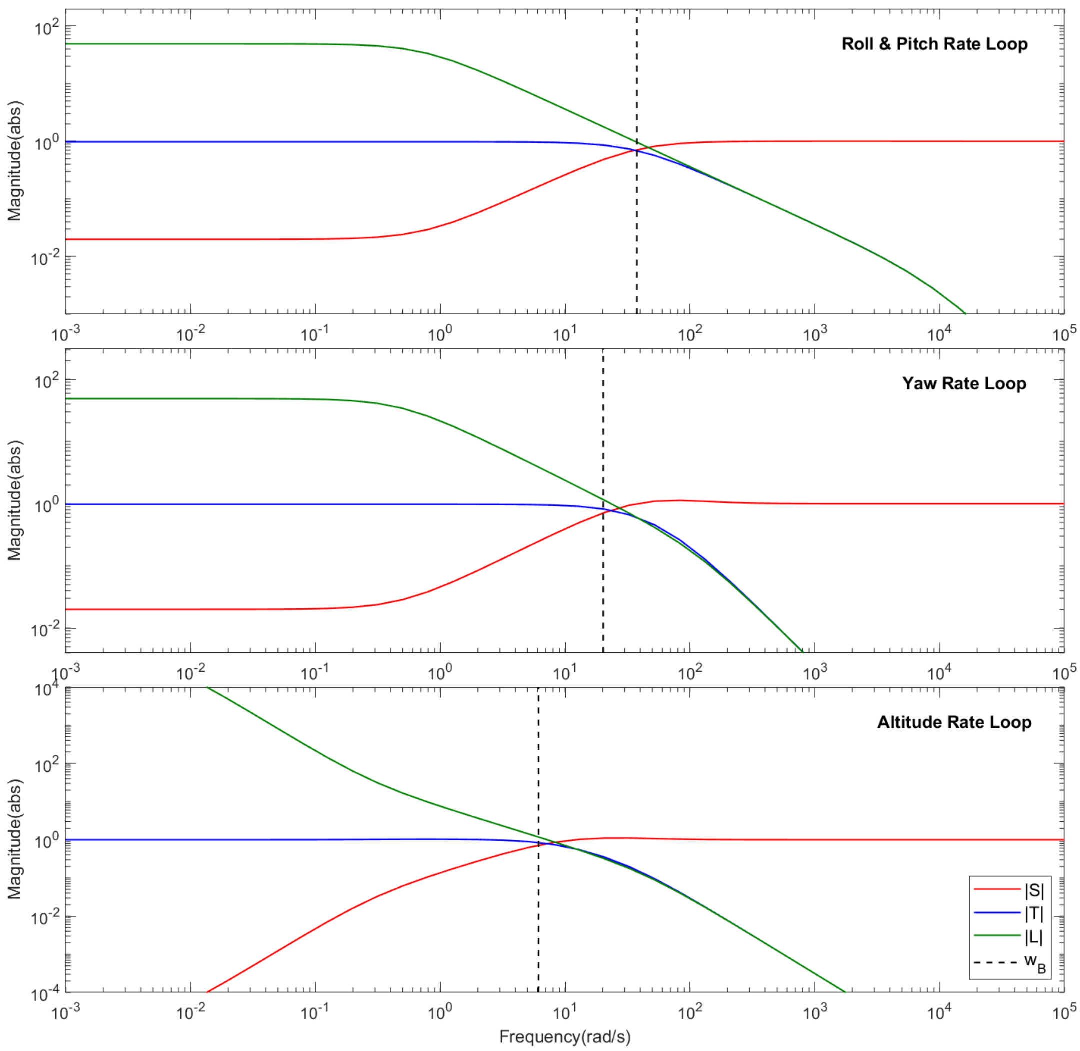


Figure 4.8: Bode magnitude plots of $|S|$, $|T|$ and $|L|$

4. The peak values of T (M_T) and S (M_S) are as follows:

Table 4.8: Robust Stability Analysis

	Roll & Pitch Rate Loop	Yaw Rate Loop	Altitude Rate Loop
M_T	1.004	1.132	1.031
M_S	1.000	1.000	1.108

The values fall below the typical values of $M_S = 2$ and $M_T = 1.25$, above which the system would exhibit poor robustness.

5. The $|S|$ and $|T|$ curves exhibit the 'water-bed' characteristic (see [118] page 167 for details), which results in a peak for systems with poor robustness. The tuning procedure has enabled the smoothing out of the peaks over the

frequency range of interest and hence have produced, almost flat curves, for all the three closed loops.

Having obtained satisfactory closed loop frequency characteristics, the time domain characteristics are examined in the next section.

4.4.6.6 Time Domain Response

In this section the time domain response of the nonlinear model controlled by the developed MSO controller is examined. The section is subdivided into three parts. In the first part, the nonlinear plant response to a step input is examined and step response characteristics are calculated. In the second part the response of a sampled collection of the uncertain plant models with the designed controllers towards a complex manoeuvre is plotted. Simulations are performed again on the same uncertain plant models, but this time controlled by set of 4 PID controllers. Details regarding the responses of both sets of simulations are discussed. In the final part of the section robust performance of the system is analyzed in the context on the obtained results.

Throughout the study, reference signals are coloured in shades of grey, output signals in shades of red, input signals in shades of blue, and disturbance signals in shades of green.

The response of the non-linear plant subjected to a reference pulse and a square disturbance input is shown in Fig. 4.9.

In the first part of the analysis, two sets of simulations are conducted. In the first, set the plant is subjected to a pulse reference setpoint change and a square wave disturbance input signal. The results from this set of simulations are used to calculate the step response characteristics of the system.

In the second set, a normally distributed (with zero mean) noise signal, with a standard deviation equal to 15% of the maximum reference value, is injected at the output alongside a delay of 1 time sample.

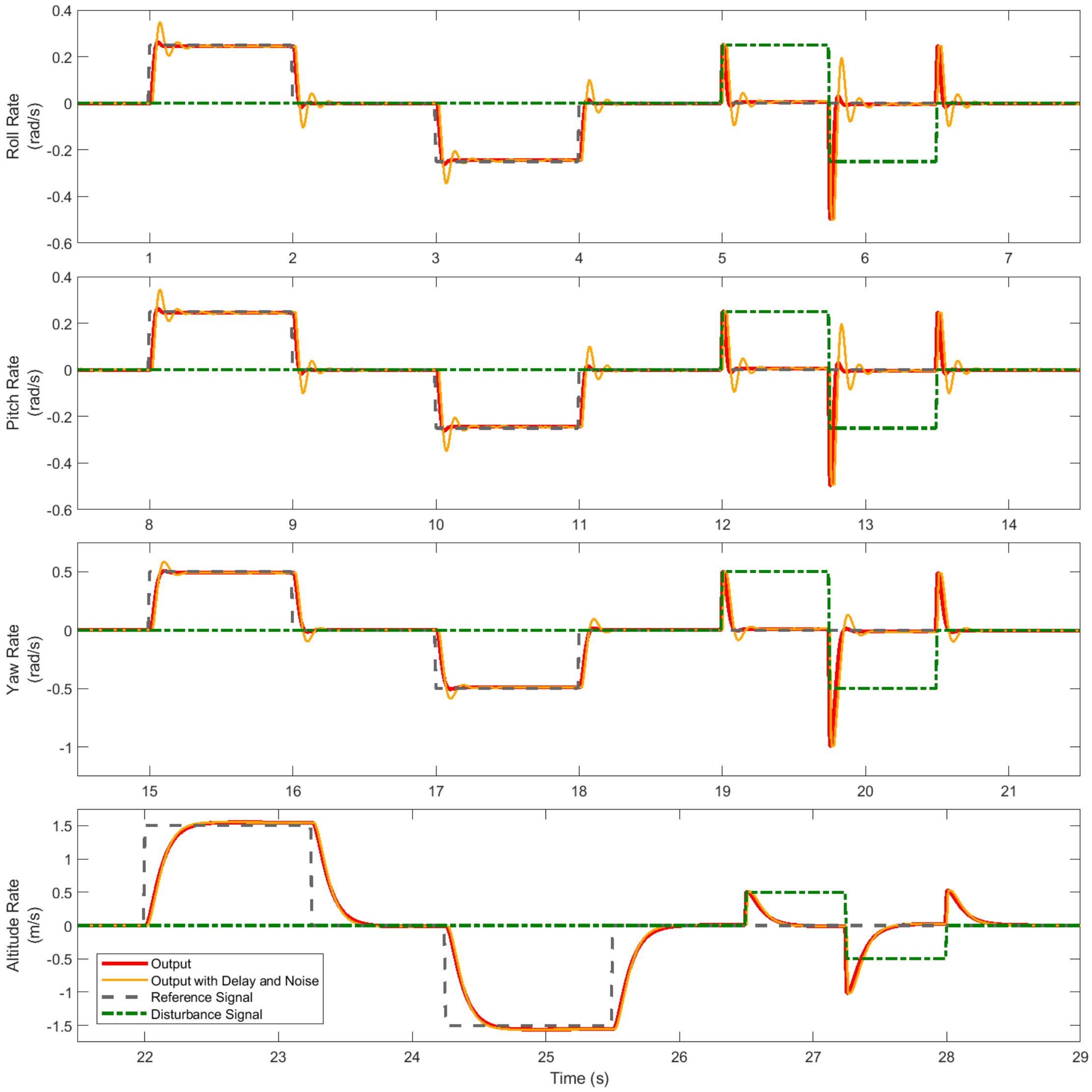


Figure 4.9: Responses to 1(or 1.25) second long reference pulse and 1.5 second long disturbance input: Setpoint change of 0.25 rad/s at $t = 1$ and of -0.25 rad/s at $t = 3$, disturbance square wave input of magnitude 0.25 rad/s at t=5 in the Roll rate loop; setpoint change of 0.25 rad/s at $t = 8$ and of -0.25 at $t = 10$, disturbance square wave input of magnitude, 0.25 rad/s at t = 12 in the Pitch rate loop; setpoint change of 0.5 rad/s at $t = 15$ and of -0.5 at $t = 17$, disturbance square wave input of magnitude 0.5 rad/s, at t = 19 in the Yaw rate loop; setpoint change of 1.5 m/s at $t = 22$ and of -1.5 at $t = 24.25$, disturbance square wave input of magnitude 0.5 m/s at $t = 26.5$ in the Altitude rate loop.

In the input signals presented in Figure 4.10, the dark blue curve represents the signals in the loop where noise and delays are absent while the light blue curve represents the case where noise and delay signals are present in the loop. The effects of coupling, i.e. the effect on a particular output channel due to the changes in other channels, are not presented in Fig.4.9, as these are negligible (see Fig. 4.15 and 4.17 to see the effects in more detail).

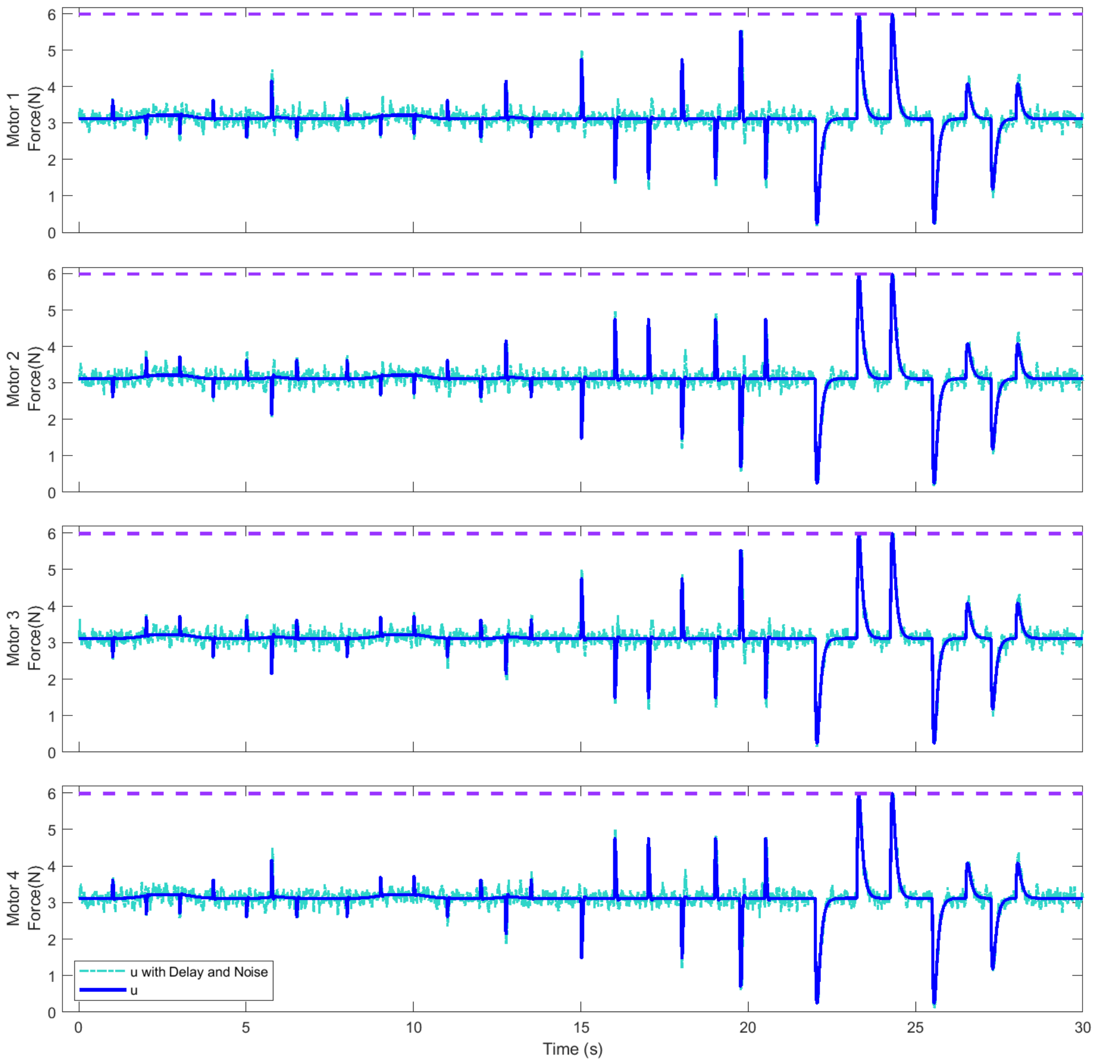


Figure 4.10: Forces exerted by individual motors in response to reference signal

Examining the results from 4.9 and 4.10, the following points can be observed.

1. The nonlinear plant responses (Fig. 4.9) shows acceptable performances both with regards to reference tracking and disturbance rejection. The acceptability is assessed from short rise times, marginal peak overshoots and short settling times. The values of step response characteristics are presented in Table 4.9.

2. The inputs i.e., the forces exerted by the propellers presented in the Fig. 4.10 (dark blue curve) stay within the maximum force that can be exerted corresponding to the maximum RPM limits designated for the selected motors. This maximum limit equal to 5.984 N is indicated in the figure by the violet dashed line.

3. The response towards the same reference inputs and disturbance signal in the presence of noise signal and time delay is indicated by the orange line in Fig.

4.9. Acceptable tracking and disturbance rejection is observed. The overshoot is higher than the case without noise and delay disturbance. In Table 4.5 it was noted that altitude controller had the largest delay margin. This is evident in the altitude rate response, which is seen to be the most immune towards delay and noise injection.

4. In Fig. 4.10 the curve shaded in light blue represents the force exerted by the individual motors in the presence of delay and the noise signal. The inputs reach the saturation limit, although they do not affect the plant response considerably.

5. Further simulations showed that the system remains stable when the delay is increased to two sample times, although it induces substantial oscillations in Roll, Pitch and Yaw loops.

In the following section, the step response characteristics are presented.

4.4.6.7 Step Response Characteristics

The step response characteristics are calculated based on the output response presented in the Fig. 4.9. Before listing down the response characteristics the terms are briefly defined below:

- Rise Time (t_r): The time taken by response signal to move from 10% to 90% of the stead state final value.
- Settling Time (t_s): Time taken for the error, defined as the difference between the steady state and current value of the signal, to reach and stay below 5% of the steady state value.
- Settling Maximum/Peak: This is the maximum value of the response once the signal has risen.
- Peak Time (t_p): The time taken for the response to reach the primary peak of the overshoot.
- % Overshoot: Overshoot is the magnitude of the peak measured from the steady state value, relative to the the steady state value. It is usually calculated as a percentage as below:

$$\%\ \text{Overshoot} = \frac{y_{\text{peak}} - y_{\text{final}}}{y_{\text{final}}} \times 100\% \tag{4.25}$$

where y_{peak} indicates peak value of the response and y_{final} the final steady state value.

- % Undershoot: The difference between steady state value and the minimum value that the response signal swings to below its steady state value after it has risen to its peak (y_{min}), calculated relative to the steady state value. It is calculated as a percentage as below:

$$\%\ \text{Undershoot} = \frac{y_{\text{final}} - y_{\text{min}}}{y_{\text{final}}} \times 100\% \tag{4.26}$$

The response characteristics are further presented in Figure 4.11.

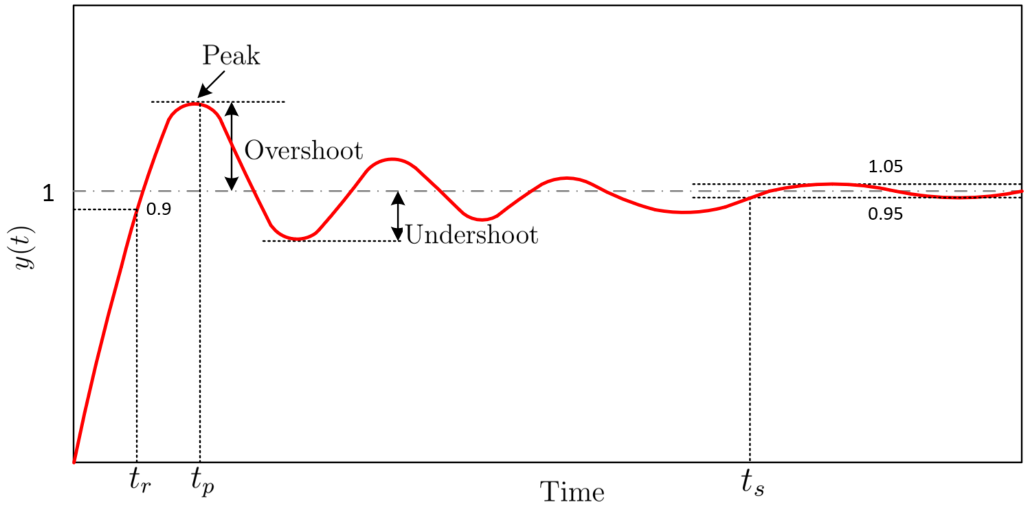


Figure 4.11: Response characteristics of the closed loop to a unit step input signal.

The values of these characteristics calculated accordingly are presented in the Table 4.9. (Note: The peak values should be read in the context of the reference signal applied to the corresponding loops; refer Fig. 4.9.)

Table 4.9: Time and frequency domain specifications after controller tuning

Response Characteristics	Roll & Pitch Rate	Yaw Rate	Altitude Rate
Rise Time (s)	0.025	0.049	0.206
Settling Time (s)	0.250	0.263	0.989
Peak	0.262	0.505	1.495
Peak Time (s)	0.060	0.100	0.720
% Overshoot	5.000	1.006	3.246
% Undershoot	0.000	0.000	0.061

Having examined the response characteristics for the nominal plant model, in the following sections performance plants containing model uncertainty will be examined.

4.4.6.8 The Parameter Hypercube

The Monte-Carlo simulations will have to be undertaken to check whether the developed controllers provide adequate stability when plants with model uncertainty are used instead of the nominal plant. The array of uncertain models are build using the parameter set at the vertices of the parameter uncertainty hypercube, as well as using a collection of randomly sampled points from the inside of the uncertain parameter space.

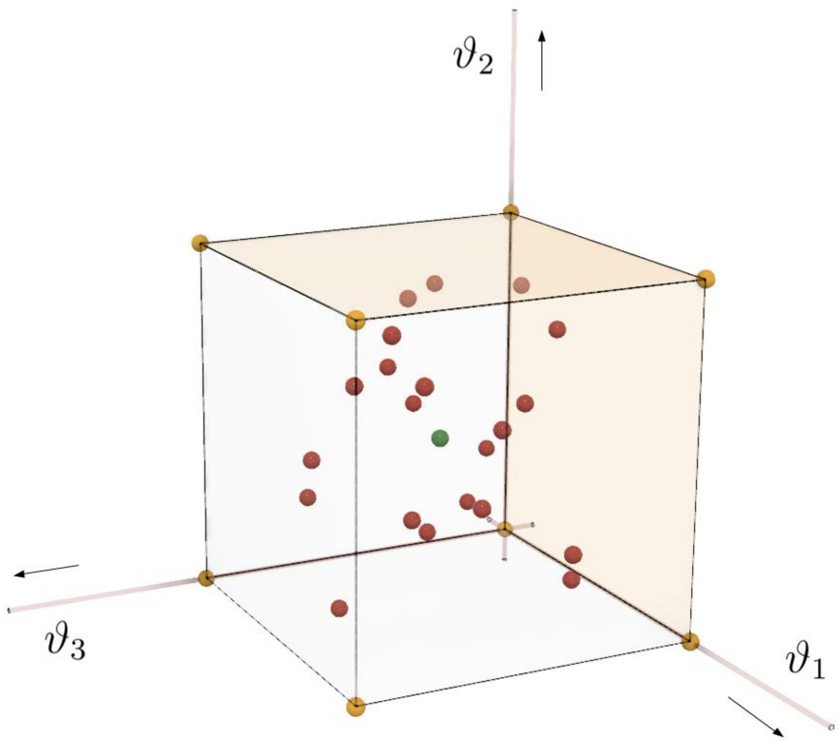


Figure 4.12: A three dimensional representation of the randomly sampled parameter hypercube.

This is depicted in Fig.4.12 where for arguments sake, a three dimensional uncertain parameter space is presented. The three axis represent the magnitude of the 3 parameters ϑ_1, ϑ_2 and ϑ_3 respectively. The yellow dot at the vertices represents the set of parameters which are used to build the extreme plants, while the red points are those sets of parameters randomly distributed points within the hyperspace. A Monte-Carlo analysis of the plant responses are preformed using the parameter sets from these red dots. The green dot located at the center of the cube represents the parameter set of the nominal plant.

A realization of uncertain plant models are generated based on parameter uncertainty definitions examined previously, and each of these models are subjected to a reference tracking objective. This objective spreads over a length of 15 seconds in which the first half tracks a simultaneous change in roll, yaw and altitude rates while in the second half a simultaneous change in pitch, yaw and altitude rates are tracked. Two sets of simulations are carried out. In the first set robust controllers are used to generate the responses. In the second set, 4 PID controllers are used to generate the responses. The PID controllers are employed primarily to perform a comparison study with the MSO, and the other robust controllers developed further on.

In total, 256 simulations corresponding to maximum and minimum values of 8 different parameters (dr is replaced by dx and dy corresponding to x-y coordinates of the rotor with respect to the center of mass, representing the uncertainty in the location of the center of mass) are conducted to plot the yellow shaded areas. Similarly 256 simulations are conducted in the Monte-Carlo analysis which uses the uniformly distributed parameter sets from inside the uncertain parameter hypercube to plot the red shaded areas. The performance plots are presented in the section after the following section.

4.4.6.9 PID Controller Setup

The PID controllers are tuned on the nominal plant model using the SIMULINK PID Tuner [181], a highly regarded, commercial PID tuning software available within the MATLAB environment.

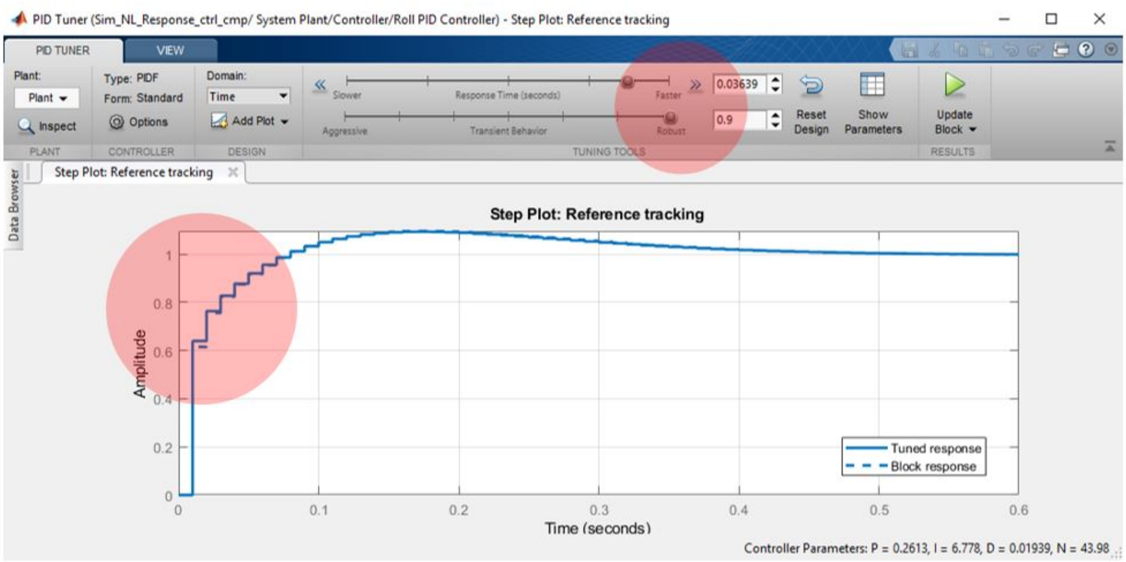


Figure 4.13: PID Controllers are tuned using the fast and robust tuning settings

The controllers are tuned to obtain the most 'robust' response as well as acceptable rise times, through out the four individual loops. As seen from Fig.4.13 the tuning dials are adjusted until the response time is the fastest and no induced oscillatory responses that develop as the signal rises (the area, indicated by the bigger red circle in the figure) are present.

The discrete time PID controller takes the Parallel format with a filtered derivative component. The settings can be observed in Fig. 4.14

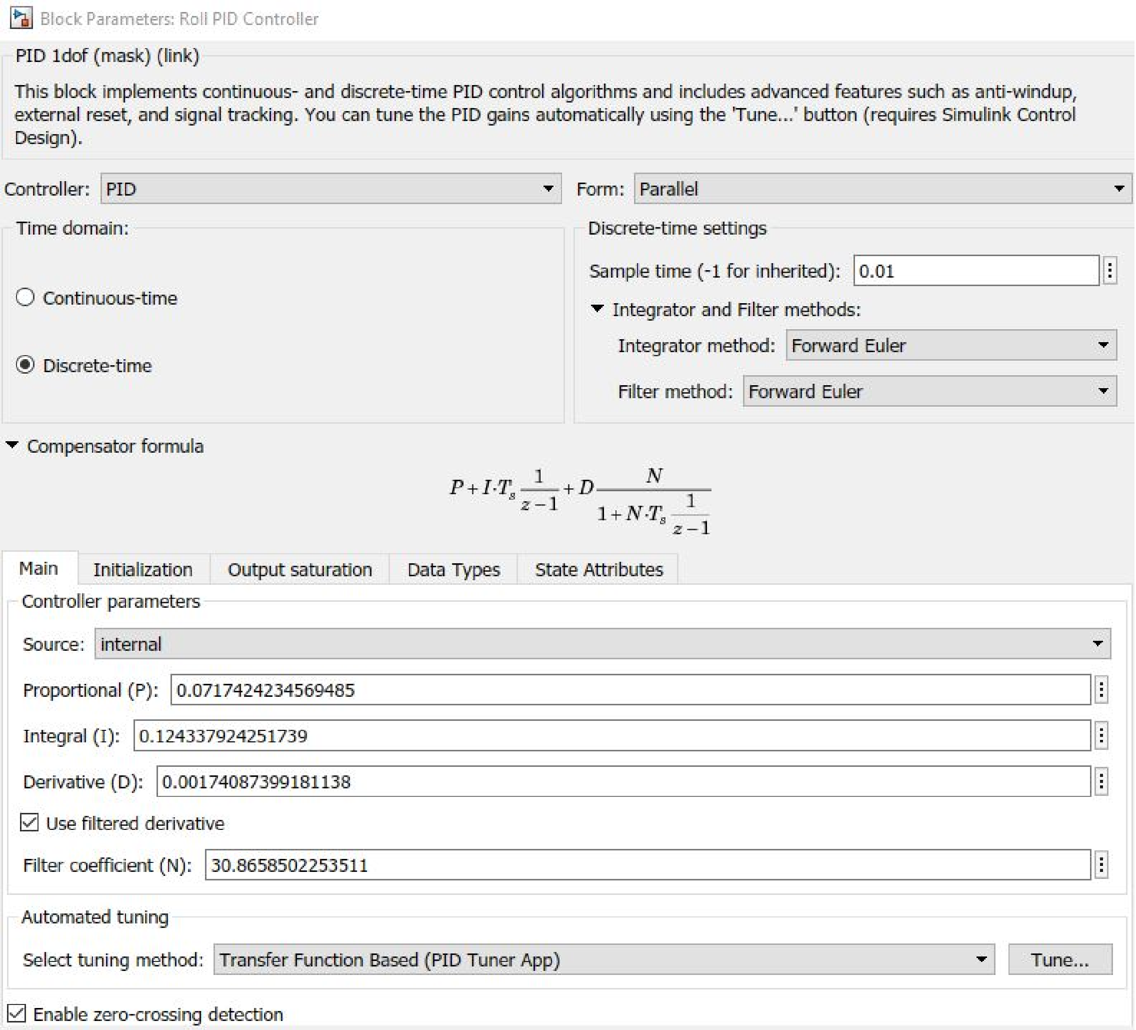


Figure 4.14: The PID Controller set-up

4.4.6.10 Performance Plots

In the output response plots (Fig. 4.15, and 4.17) the reference input command is marked using dashed black lines. The signals from all the plants formed by the parameters at the *vertices* of the uncertainty parameter hypercube are shaded in yellow while the remaining Monte-Carlo parameter simulations are shaded in red. In each sub-plot, areas marked in dashed-line boxes are zoomed out. These present a more detailed view of the responses.

As can be observed, while a square wave is applied towards roll, pitch, and yaw rate loops, for the altitude rate a filtered square wave is applied. While the roll, pitch and yaw rate loops are expected to accommodate sharp changes in reference commands, the altitude rate changes are expected to be more gentler.

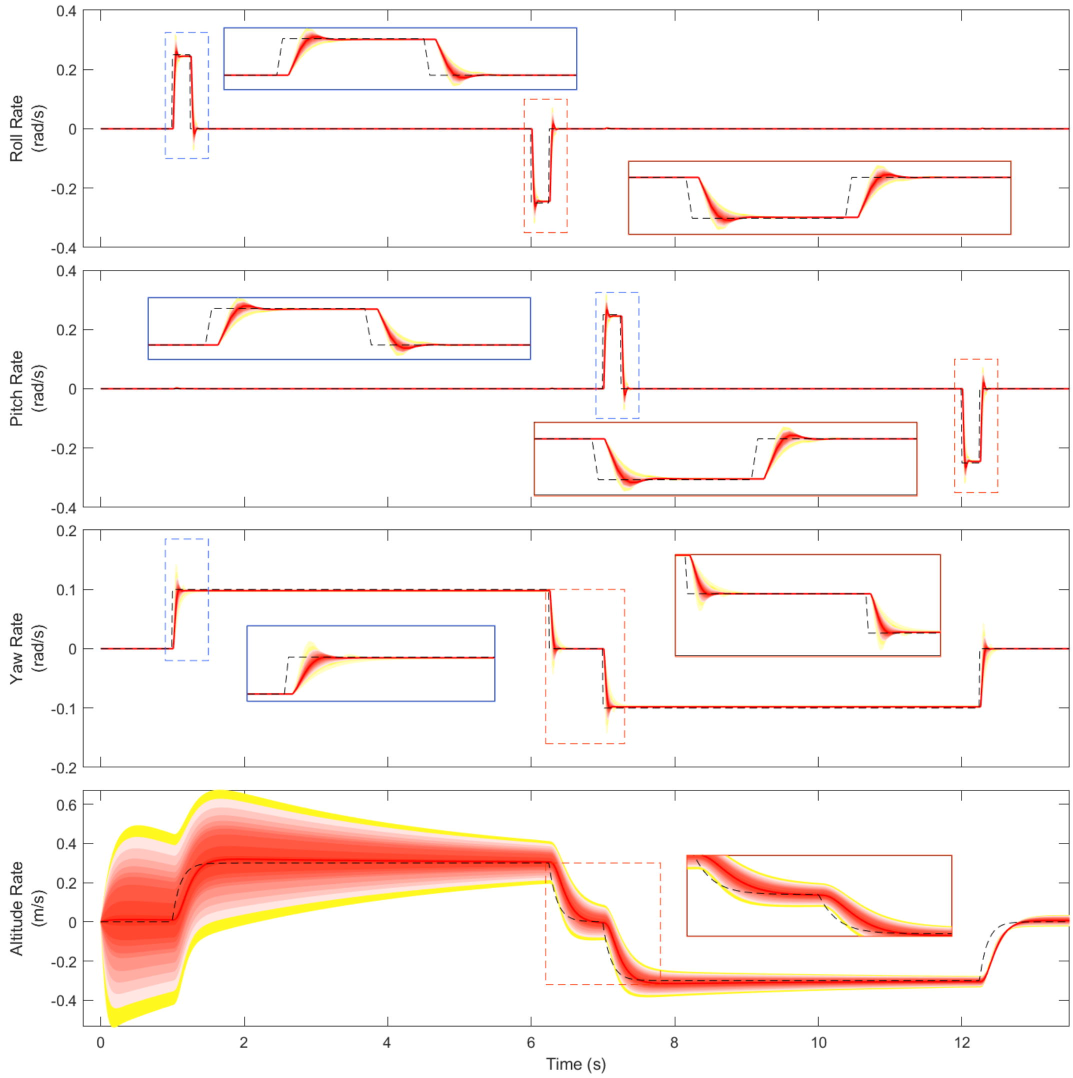


Figure 4.15: The response of the MSO based plant with model uncertainty to the complex manoeuvre reference command. The inserts show zoomed out plots of the dynamics.

The responses to the reference commands by the system controlled by the MSO controller are presented in Fig.4.15. It can be noted that red bands (performances of the plants from within the parameter hypercube) are always inside the yellow bands (performances of the plants from the vertices of the hypercube)

Fig.4.16 presents the input trends of the uncertain plants. The signals represent the forces exerted by the individual motors as the simulation progresses. The inputs to those plants at the vertices of the uncertain parameter hypercube are shaded in light blue while to those models used for the Monte-Carlo analysis are shaded in Navy blue. The dashed light blue lines indicates the maximum and minimum limits of the forces that can be exerted by individual motors.

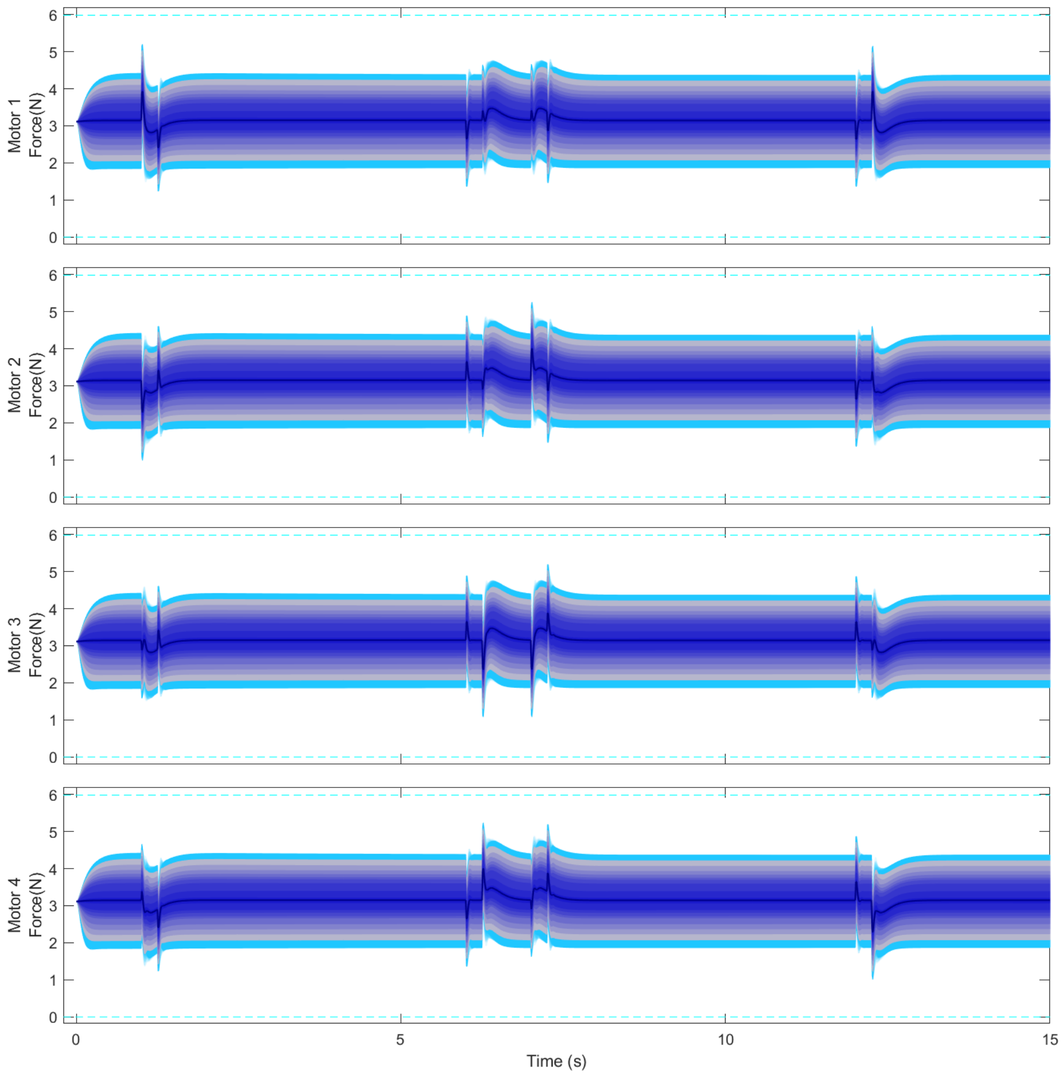


Figure 4.16: MSO controlled plant rotor responses to the complex manoeuvre reference signal.

In Fig.4.17 the responses to the same reference input by the uncertain plants controlled by the PID controller are presented for comparison. The zoomed portions marked by the dashed boxes in both Figures 4.15 and 4.17 represent the same areas measured in the time and the corresponding output axis.

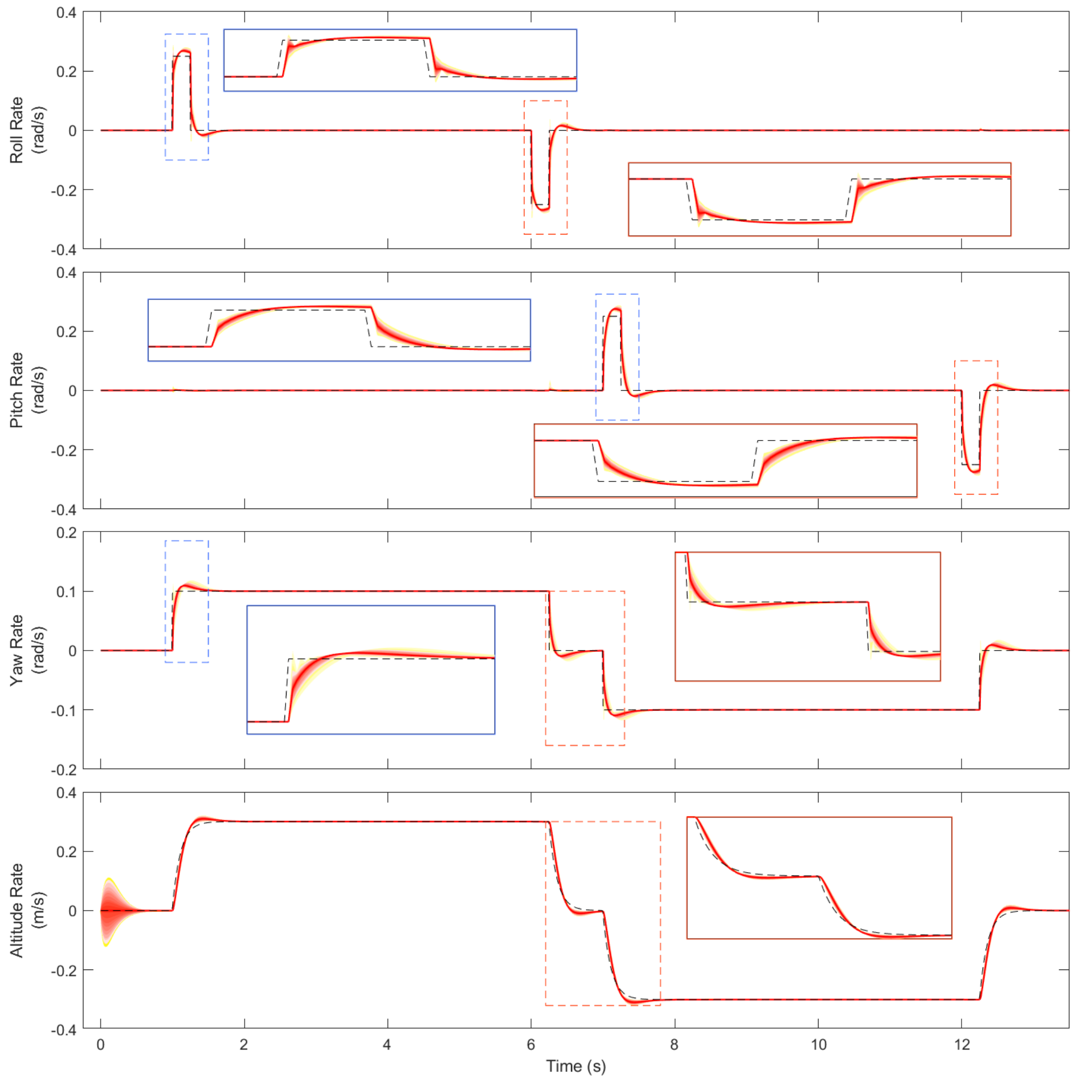


Figure 4.17: The response of the PID based plant with model uncertainty to the complex manoeuvre reference command (compare with Fig.4.15).

Fig. 4.18 presents the corresponding inputs to the PID controlled uncertain plants. Comparing the response and input signals generated by the MSO and PID controlled plants, the following points can be noted.

- The output response towards the step signals (i.e. in the roll, pitch and yaw rate loops) indicate that the PID controlled Loop (PCL), although tuned to provide the fastest response, produces a slower response compared to the MSO Controlled Loop (MCL). The spread of the output signals in that of the PID loop although remain smaller, this comes at the expense of the slower response. Tuning the PID controller to be any more faster would induce oscillations.

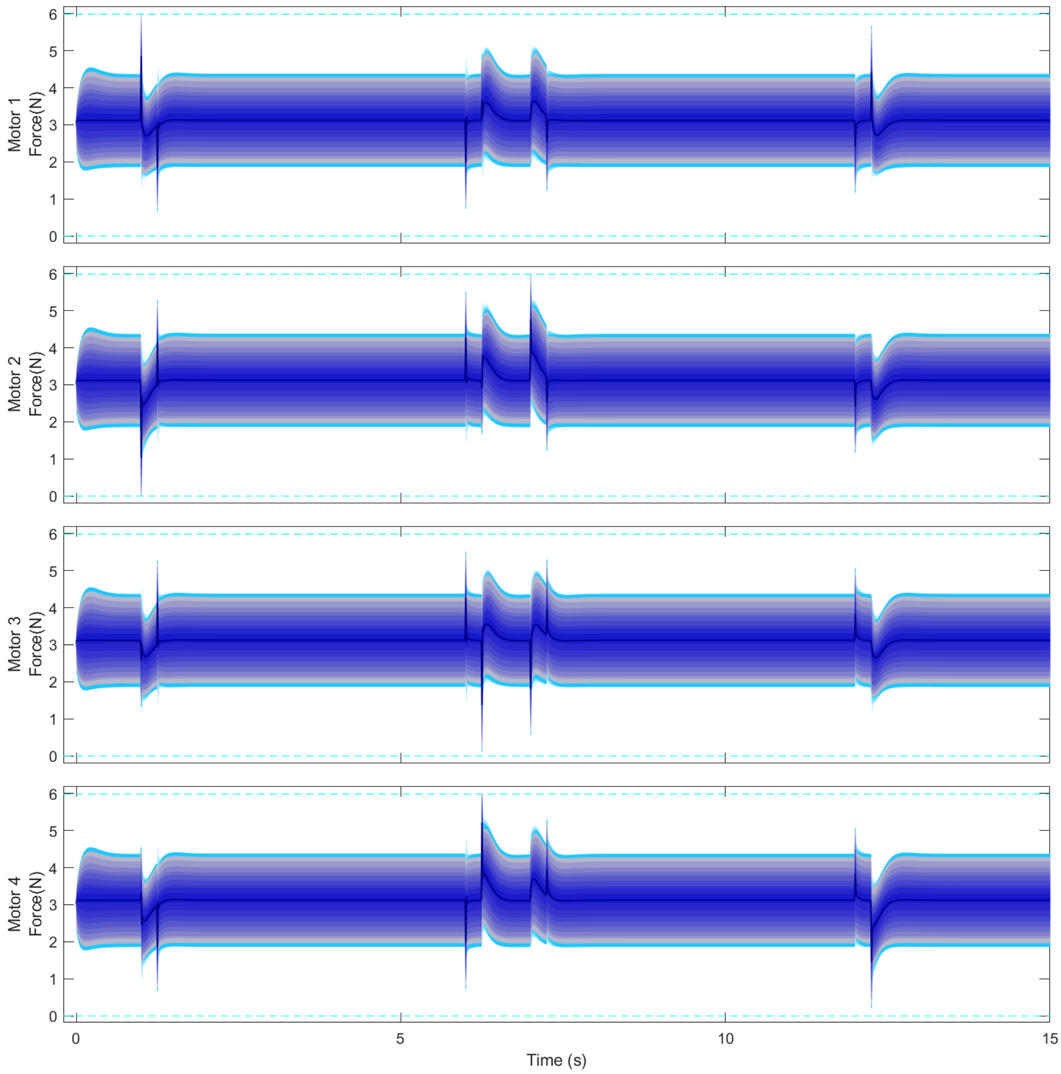


Figure 4.18: PID controlled plant rotor responses to the complex manoeuvre reference signal (compare with Fig.4.16).

- In MCL, except for the yaw response (where one can observe gentle oscillations), the uncertain plant models from the hypercube vertices do not produce oscillatory responses. While in PCL careful examination reveals gentle oscillatory responses, which can be observed in pitch and roll and yaw rate loops, for the extreme plants.

- The differences in the altitude rate responses seen between MCL and the PCL are the most significant. The spread observed in PCL are significantly smaller. To understand the reason behind this difference in performance we need to compare the motor inputs from both cases

  It can be observed that in PCL, the inputs exact an extremely fast motor input change (extremely small motor start-up time). In the real world, extremely fast changes in motor RPM are not desirable both due to the difficultly in achieving it and due to maintenance related reasons.

Interesting performance characteristics are observed when the systems are tested with an input that changes slowly. To simulate this situation, a low pass filter is introduced with response characteristics similar to that of a first order system. As the filter cut-off frequency is decreased the PCL begins to exhibit greater input oscillations, taller input usage spikes and ultimately becomes unstable. While MCL, maintains stability to a greater extent while exhibiting smaller input oscillations and shorter usage spikes.

- The characteristic bulge observed at the start of the altitude rate responses of both PCL and MCL are brought about by the variation of mass. A $\pm$ 40% variation of mass is incorporated into the model uncertainty and the altitude rate changes at the start to compensate for this change. This also accounts for the spread in inputs, as when mass increases, in order to maintain the altitude a higher RPM is exacted, and vice versa.

- Saturation of inputs that occur in PCL (Fig. 4.18), magnifies the effects of the inherent coupling between loops, which can be observed in roll and pitch rates as spikes. From the inputs plots, compared to the MCL, it can be seen that PCL tracks the reference command at the expense of inputs which are larger in magnitude.

- Careful examination reveals gentle oscillations in all four inputs signals in the PCL response (Fig. 4.18), while in MCL inputs are oscillation free.

Summing up, compared to PCL the MSL provides a crisper attitude rate and a reliable altitude rate response (as opposed to a sluggish response) alongside input demands that are cheaper as well as oscillation free. Hence the claim that, if the uncertainty in the plant model can be quantified a dedicated robust controller would perform better than its PID counterpart, can be made. Having analysed the robust performance (RP) of the system in the time domain the next section examines the RP characteristics from the frequency domain.

4.4.6.11 Robust Performance Analysis

Having calculated the $\mathcal{H}_\infty$ norm of the weighted sensitivity, complementary sensitivity functions and their sum in Table 4.6, we had observed that the loops failed to achieve strict robust performance. In light of this earlier observation, the values in Table 4.10 provides limits for the uncertainty that would guarantee robust performance. (Note: MATLAB commands `robgain` or `robustperf` which is an earlier version, can be used to calculate the robust performance margins.)

Table 4.10: Robust Performance Analysis

Loops	Roll & Pitch Rate Loop	Yaw Rate Loop	Altitude Rate Loop
Uncertainty upper bound	0.825	0.892	0.815
Critical frequency (rad/s)	0.0013	0.1737	4.8143

From the calculated values, it can inferred that the uncertainty in roll and pitch rate loops needs to be lowered to a level of 82.5% of its current value at steady state, (frequency of 0.00133 rad/s) to achieve robust performance. For the yaw rate loop model uncertainty at a frequency of 0.17 rad/s should be lowered to 89% of the current value to achieve robust performance. Similarly at a frequency of 4.814 rad/s the model uncertainty in altitude rate loop should be lowered to 81.5% of the current value to achieve RP. The table also points towards the limitations of the stacked S/T MSO technique as the algorithms available currently in commercial software do not develop controllers that can guarantee RP although brings the system quite close to guaranteed RP (as mentioned earlier in eq. (4.23)).

This brings us to the end of the design procedure using mixed sensitivity optimization strategy. In the following section Loop Shaping Design Procedure (LSDP) will be explained and controllers for the quadrotor will be designed and analysed.

4.5 Loop Shaping Design Procedure

In MSO, we had designed frequency weights that depended on the parametric model uncertainty of the problem. LSDP eliminates this problem-specific uncertainty representation, by using a generalized co-prime factor uncertainty [182, 183]. Loop Shaping Design Procedure essentially combines the classical loop shaping strategies for shaping the frequency response of open-loop plant, with the $\mathcal{H}_\infty$ robust stabilization.

Coprime factor uncertainty evolved from coprime factor representation of the plant model. It involves factorising the plant model G as follows:

$$G(s) = {M_l}^{-1}(s)N_l(s) \tag{4.27}$$

where the subscript l stands for left-coprime factorization and M and N are stable and coprime transfer functions. The stability condition requires that all right half plane poles of $G(s)$, would be contained in $M_l(s)$ as RHP zeros, and the RHP zeros of $G(s)$ will be contained in $N_l(s)$. Coprimeness on the other hand requires that, while there exist stable transfer functions $U_l(s)$ and $V_l(s)$, M_l and N_l satisfy the Bezout identity:

$$N_l\,U_l + M_l\,V_l = I. \tag{4.28}$$

This form of representation allows for both poles and zeros to cross over to RHP and hence enable the representation of an unstable transfer function, as two stable transfer functions. The *perturbed* plant model G_p can now be expressed as follows:

$$G_p(s) = (M_l + \Delta_M)^{-1}(N_l + \Delta_N). \tag{4.29}$$

where Δ_M and Δ_N represents unknown stable transfer functions representing additive uncertainty in coprime factors. The perturbed plant is shown in Fig. 4.19

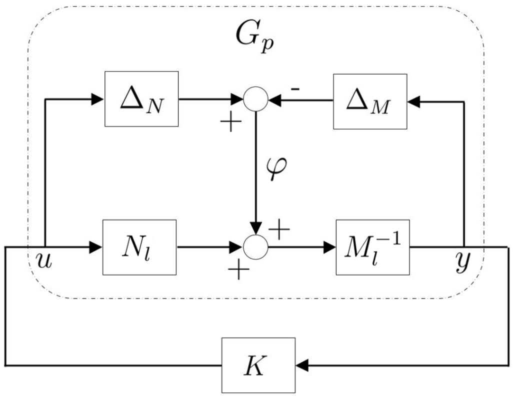


Figure 4.19: Coprime factorisation of the perturbed plant

In the next section the controllers based on the 1 and 2 Degree of Freedom designs of the LSDP will examined. Similar to the MSO development, the important concepts of each of these design procedures will be briefly introduced. For detailed analysis of development of the LSDP controllers, refer [184].

4.6 1 DOF LSDP

As noted in Fig. 4.19, in the co-prime factor representation of the uncertain plant, the perturbations carry no weights. The magnitude of these uncertainty representations in the perturbed plant $G_p(s)$ (eq. (4.29)) are such that $||[\Delta_M\ \Delta_N]||_\infty \leq \epsilon$. Here ϵ >0 is the stability margin of the system. The system attains robust stability if and only if the controller stabilizes the nominal feedback loop and achieves the following:

$$\gamma_K \triangleq \left\| \begin{bmatrix} K \\ I \end{bmatrix} (I - GK)^{-1} M^{-1} \right\|_\infty \leq \frac{1}{\epsilon}. \tag{4.30}$$

Such a controller will stabilize all the plants in the family of perturbed plant G_p. The cost function γ_K represents the $\mathcal{H}_\infty$ norm from φ to $[u\ y]^T$. With this arrangement, as observed in [182], the smallest magnitude of the cost function $\gamma_{\min}$ is given by:

$$\gamma_{\min} = \frac{1}{\epsilon_{\max}} = \sqrt{1 - \|[N\ M]\|_H^2} = \sqrt{1 + \rho_S(X\ Z)}, \tag{4.31}$$

where $\|\cdot\|_H$ represents the Hankel norm and ρ_S the spectral radius. For a minimally realized plant G (with the state space matrices A, B, C and D) Z is the unique positive solution to the Generalized Control Algebraic Ricatti Equation (GCARE)

$$(A - BS^{-1}D^TC)Z + Z(A - BS^{-1}D^TC)^T - ZC^TR^{-1}CZ + BS^{-1}B^T = 0 \tag{4.32}$$

where $R = I + DD^T$ and $S = I + D^TD$. X is the unique positive definite solution to the Generalized Filter Algebraic Riccati Equation (GFARE):

$$(A - BS^{-1}D^TC)X + Z(A - BS^{-1}D^TC)^T - XBS^{-1}B^TX + C^TR^{-1}C = 0 \tag{4.33}$$

For practical problems a $\gamma > \gamma_{min}$ is used to calculate the controller. For such a γ, the controller that satisfies

$$\left\| \begin{bmatrix} K \\ I \end{bmatrix} (I - GK)^{-1} M^{-1} \right\|_\infty \leq \gamma \tag{4.34}$$

is given by :

$$K = \left[\begin{array}{c|c} A + BF + \gamma^2(L^T)^{-1}ZC^T(C + DF) & \gamma^2(L^T)^{-1}ZC^T \\ \hline B^TX & -D^T \end{array} \right] \tag{4.35}$$

where

$$\begin{aligned} F &= -S^{-1}(D^TC) + B^TX, \text{ and} \\ L &= (1 - \gamma^2)I + XZ. \end{aligned} \tag{4.36}$$

Although γ_{min} can be calculated without the usual γ-iteration procedure by resolving eq. (4.32) and eq. (4.33) (use `icare` in MATLAB), the subsequent controller implementation runs into problems as L (see eq. (4.36)) turns singular when $\gamma = \gamma_{min}$, hence the choice of a sub-optimal γ.

In the traditional LSDP development, there are three main steps:

1. The open loop singular values of the nominal plant are shaped using pre and post compensators. The desired loop shape would encompass the open loop bandwidth requirements, as well as the usual loop gain requirements of higher gain at lower frequencies for reference tracking, and lower gain at the high frequencies for disturbance and noise rejection.

2. The shaped plant denoted by $G_S = W_2\, G\, W_1$ (see section 4.6.1) is then stabilized according to steps mentioned earlier using the controller K.

3. The feedback controller is developed by augmenting K with the designed pre and post compensators as $K_{final} = W_1\, K\, W_2$.

The controller implementation can also take the form presented in Fig. 4.20 in contrast to the general format of Fig. 4.2, as in this format that the controller doesn't directly get excited from reference inputs.

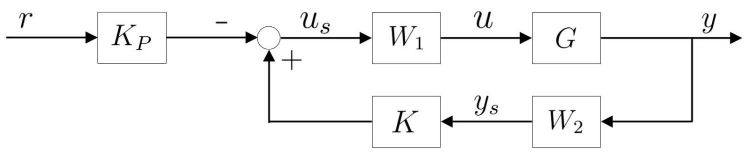


Figure 4.20: 1-DOF LSDP alternate implementation format

The reference pre-filter K_P ensures that the steady state error is zero. In this approach, to reach satisfactory weight designs alongside the controller development, trial and error procedures are employed as recommended in [185] and [186]. It is also common to use the weights based on a initial controller design as seen in [187], where the weights are designed for a LQG based controller and further reused for the $\mathcal{H}_\infty$ controller.

In the algorithm presented in this section, steps 1 and 2 are combined to a single step. An outer framework (similar to **Algorithm #1** in MSO on page 55) provides options to provide time domain specification constraints. This, in addition to the LSDP which defines the frequency domain specifications, provide for both a faster design in terms of pre and post compensator development, and an optimal controller in terms of considering tracking error and disturbance rejection.

4.6.1 Design weights W_1, W_2 and the prefilter K_P

The selection of weights W_1, W_2, and the prefilter K_P can be formulated as the following problem: For the plant configuration presented in Fig. 4.19, find $\tilde{W} = (W_1, W_2)$ and K_P such that

$$\gamma_0(\tilde{W}) \leq \epsilon_\gamma \tag{4.37}$$

and

$$\Phi_k(\tilde{W}, K_P) \leq \epsilon_k \tag{4.38}$$

alongside the controller K that stabilizes the nominal plant, while maintaining design optimality. Here $\gamma_0(\tilde{W})$ is equal to the infinity norm in eq. (4.34) but for the shaped plant G_S. So $\gamma_0(\tilde{W})$ is given by

$$\gamma_0(\tilde{W}) = \left\| \begin{bmatrix} W_1^{-1} K \\ W_2 \end{bmatrix} (I - GK)^{-1} [W_2^{-1} \, G \, W_1] \right\|_\infty \tag{4.39}$$

where K is the controller that stabilizes the system. $\Phi_k(\tilde{W}, K_P)$, an element of $\mathbf{\Phi}$ $= [\ \Phi_1\ \Phi_2 \cdots \Phi_n]$ represents the closed loop functional required to be constrained. Similar to the design procedure in MSO vector, $\mathbf{\Phi}$ contains step response characteristics. Here ϵ_γ and ϵ_k are the real numbers that represents the desired upper limits of γ_0 and Φ_k respectively. Development of the weights and the prefilter, first involves parametrizing them. While K_P can take the form of simple scalars, W_1 and W_2 takes the form of a transfer function as below:

$$W_k = \frac{w_1(s + w_2)}{s + w_3}. \tag{4.40}$$

Once parameterized, the weights and the prefilter alongside the controller can be developed using **Algorithm #2**. The design optimality is ensured by the optimization problem central to the algorithm.

Algorithm # 2 Weight, prefilter selection and controller design for 1 DOF LSDP

Inputs: Plant model G, design constraints.
Outputs: K, W_1, W_2, K_P.

1. Scale the given plant G according to the criteria described in section 3.3.
2. Define the appropriate form of W_1, W_2 and K_P and choose the initial values of the parameters.
3. Choose the magnitude of the factor with which γ_{min} needs to be multiplied to obtain γ
4. Define the cost function $\mathcal{J}$ (see eq. (4.17)) and initialize the weights W1 though W4 (see section 4.4.3 for details regarding development of the cost function).
5. Develop the vector $\mathbf{\Phi}$ which contains the following nonlinear closed loop functionals:
   1. Maximum overshoot, M_p
   2. Rise time, t_p
   3. Settling time, t_s
   4. Maximum value of input, u_{max}
   5. Gamma, γ

   Here M_p, t_p and t_s, the step response characteristics and u_{max} are the time domain specifications, while γ refers to the frequency domain specification.

---

6. Define the frequency domain bounds ϵ_γ and time domain specifications limit vector $\boldsymbol{\epsilon}$.
7. Implement the constrained non-linear optimization problem. This is achieved using the OPTI MATLAB toolbox. The black box optimizer NOMAD is chosen as the optimization routine. Nonlinear constraints are defined based on ϵ_γ and $\boldsymbol{\epsilon}$ for the vector $\boldsymbol{\Phi}$ that defines the closed loop functionals.
8. Calculate the highest obtainable stability margin $1/\gamma_{\text{min}}$. If this value is less than .25, vary the forms of the weights W_1, W_2 (see 4.18) as well relax the limits of the linear constraints and repeat step 7 again.

As in **Algorithm #1**, the algorithm can be implemented in MATLAB or similar software which supports the NOMAD optimization package. Successful implementation produces the weights W_1, W_2, and K_P alongside the controller K. The following points can be noted while implementing the algorithm in MATLAB:

1. During initial runs, the time and frequency domain constraints can be relaxed. These can be tightened as successful iterations are completed.
2. Matlab routine `ncfsyn` (see Page 235 [166]) is called in step 7 to develop the controller.
3. ITSE has been used in step 4, for developing the cost function. Among the various error performance indices (IAE, ITAE, ISE, ITSE etc.) ITSE has been found to facilitate the fastest convergence.

4.6.2 Performance Criteria

Since the coprime factor uncertainty representation isn't plant specific, the robust coprime stability margin $\epsilon_{max} = 1/\gamma_{min}$ is considered as the scale with which the robustness is assessed. In practical problems if $\epsilon_{max} >.25$, the designs are considered successful as noted in [118].

The aforementioned stability margin is also known as the gap metric stability margin. The gap metric [188] and the ν-gap metric [189, 190] is a criteria used to measure distance between different plant models based on their difference in closed loop performance. For both metrics given a controller K and plants G_0 and G_1 the following identity holds:

$$\sin^{-1}\big(b(G_1, K)\big) \geq \sin^{-1}\big(b(G_0, K)\big) - \sin^{-1}\big(\delta_\nu(G_0, G_1)\big) \tag{4.41}$$

where $b(G_k, K)$ represents the gap metric stability margin and δ_ν represents the metric criteria. Calculating the stability criteria for a given plant and controller, enables us to calculate the distance (in the metric criteria) of the furthest plant from the nominal plant, that can be stabilized by the same controller.

The metric always takes a value between 0 and 1 i.e.

$$0 \leq \delta_\nu(G_0, G_1) \leq 1 \tag{4.42}$$

The gap and ν metric criteria will be used to analyse the robustness properties of the controller. The stability margin will also be used to calculate guaranteed lower bounds of the classical gain and phase margins given by the following expressions:

$$\begin{aligned} GM &\geq \frac{1+SM}{1-SM} \\ PM &\geq 2\sin^{-1}(SM) \end{aligned} \tag{4.43}$$

where SM represents the gap metric stability margin. For further details regarding the gap and ν-gap metric criteria including their expressions refer to [191].

This brings us to the end of LSDP 1-DOF controller design procedure. The attitude and altitude controller designs using this strategy will be explored in the next section.

4.6.3 Attitude-Altitude controllers for a Quadrotor

In this section the development of the LSDP 1 DOF controllers for the attitude and altitude tracking of the quadrotor model from section 4.2 will be examined. The section is composed of different parts. In the first part the loop shaping weights and the values of the precompensator are presented. The loops shapes alongside the sensitivity and complementary sensitivity are examined next following which the closed loop bandwidth and the classical stability margins will be presented.

In the final parts of the section, the nonlinear responses, responses of the uncertain plant model, and robustness of the closed loop will be examined.

4.6.3.1 Optimization Problem

The optimization problem for development of the loop shaping weights is exactly the same as that presented in section 4.4.6.2 except for the parameters searched and constraint limits of γ_0 (eq. (4.37)). The total number of parameters that need to be determined equals 7 (in contrast to three — w_1, w_2, w_3, for MSO controller), these being 3 parameters each for W_1 and W_2 (equation 4.40), and one parameter that constitutes K_P. The constraints for γ_0 are set as 4 as an upper bound and 0.25 as a lower bound.

4.6.3.2 The Designed Loop Shaping Weights and the prefilter

The weights developed to shape the open loop, i.e, W_1, W_2 and the prefilter K_P for the attitude-altitude controllers are listed below:

Roll and pitch rate:

$$W_1 = 110.4597\,\frac{s+48.8819}{s+91.8782}\,, \quad W_2 = 0.0022\,\frac{s+34.2769}{s+16.3131} \quad K_P = 0.9802 \tag{4.44}$$

Yaw rate:

$$W_1 = 100\,\frac{s+40.1646}{s+27.8011}\,, \quad W_2 = 0.0045\,\frac{s+18.4082}{s+16.0159} \quad K_P = 0.9800 \tag{4.45}$$

Altitude rate:

$$W_1 = 218.0153\,\frac{s+30.1754}{s+96.9799}\,, \quad W_2 = 0.0012\,\frac{s+48.4101}{s+6.9793} \quad K_P = 0.9820 \tag{4.46}$$

The controllers themselves are presented in Appendix D.1. The time and frequency domain specifications of the modified closed loop of the linearised plant are given in the following table:

Table 4.11: Time response characteristics and frequency domain specification of the linear closed loop model

Closed Loop Constraints	Roll & Pitch Rate controller	Yaw Rate controller	Altitude Controller
% Overshoot	0.528	0.985	19.495
Rise Time (s)	0.050	0.046	0.250
Settling Time (s)	0.079	0.068	1.171
$u_{\max}$	0.490	1.000	1.000
$\gamma_{\min}$	1.441	1.509	1.629

The cost function $\gamma_{\min}$ is multiplied with a factor of 1.1 to generate γ which is used to develop the controller using eq. (4.35). The normalized reference signal of unit magnitude is employed to generate the step responses. Having presented the loop shaping weights and the step response characteristics of the linearised model, in the next sub section the stability specifications of the loops are analyzed.

4.6.3.3 Classical Stability Margins and Nominal Stability

The classical stability margins from analyzing the shaped loops are presented in the following table. The nominal plant as well as the controller, are discretized at a frequency of 100 Hz (sample time of 0.01s).

Table 4.12: Classical Stability Margins and Crossover Frequencies

Stability Margins	Roll & Pitch Rate Loop	Yaw Rate Loop	Altitude Rate Loop
GainMargin (abs)	4.700	6.274	33.051
GM Frequency (rad/s)	312.15	311.8	314.159
PhaseMargin (rad)	1.369	1.274	1.043
PM Frequency (rad/s)	36.986	35.334	5.557
DelayMargin (s)	0.037	0.036	0.188
DM Frequency (rad/s)	36.986	35.334	5.557
Closed loop Stability	Stable	Stable	Stable

Achievement of acceptable gain and phase margins (GM >2 and PM >30°) for all the three shaped loops can be observed from Table 4.12. The gain crossover frequencies for the three loops are close to approximately close to 313 rad/s.

The delay margin proves a safety of 0.037 s, 0.036 s and 0.188 s for the roll and pitch, yaw loops and altitude loop respectively. By analysing the Nyquist plot the loops are also found to be nominally stable. The closed loop bandwidths are further examined in the next section.

4.6.3.4 Closed Loop Bandwidth

The closed loop bandwidth as defined in section 4.4.6.5 for the loops is presented in Table 4.13. It can be noted that the altitude controller has a smaller bandwidth which should hence reflect in a longer step response rise time.

Table 4.13: Closed Loop Bandwidth

Loops	ω_b (rad/s)
Roll & Pitch Rate	36.92
Yaw Rate	31.75
Altitude Rate	3.82

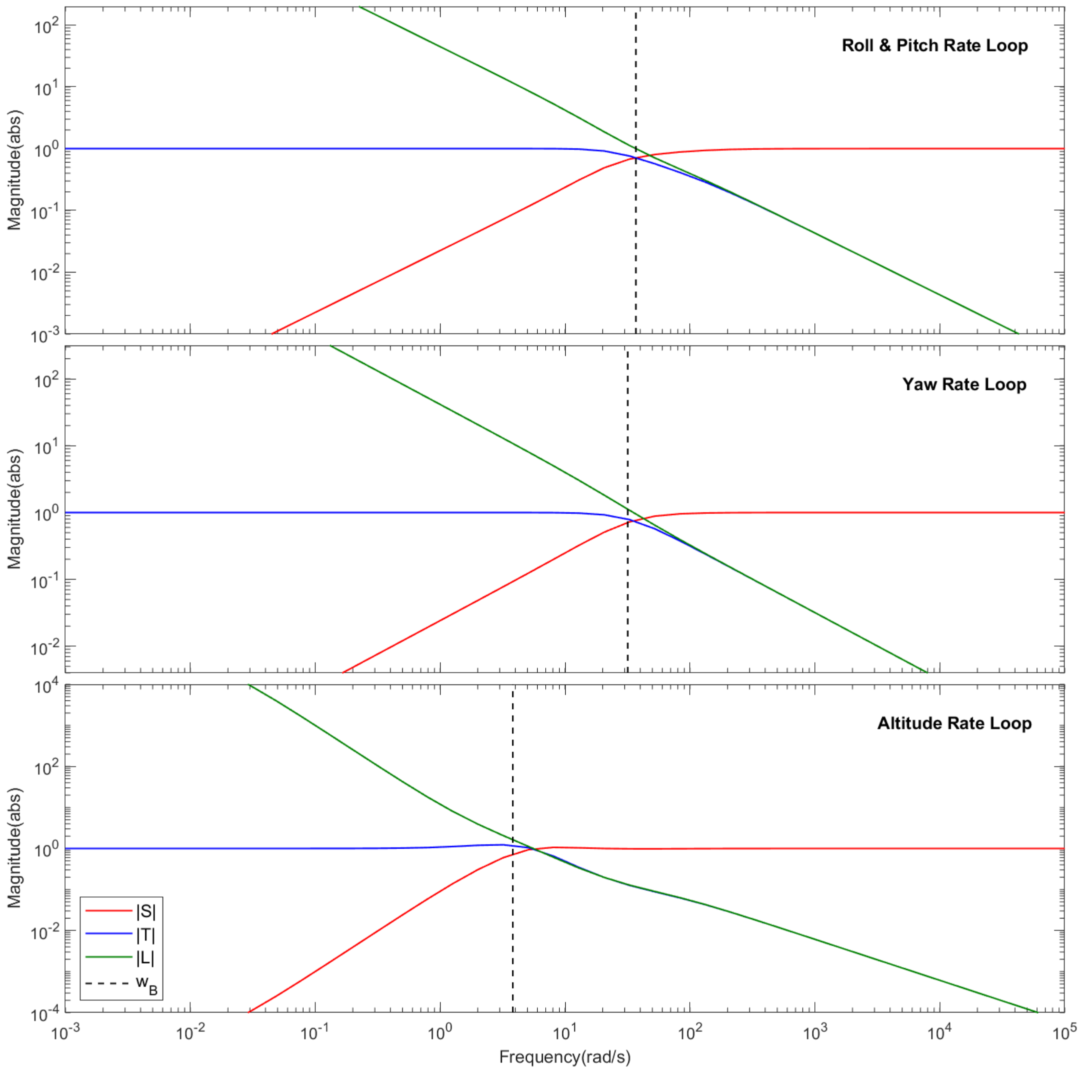


Figure 4.21: LSDP 1-DOF based plant: Bode magnitude plots of $|S|$, $|T|$ and $|L|$

The values in Table 4.13 are arrived from the Fig. 4.21. Similar to the MSO controllers developed earlier, the LSDP controllers also managed to smooth out the peaks of $|S|$ and $|T|$, ensuring better robustness against uncertainties.

The nature of curves in Fig. 4.21 resemble those from Fig. 4.8. The magnitude of $|S|$ remain low at low frequencies up to around closed loop bandwidth frequency

ω_B after which it plateaus around unity. The magnitude of $|T|$ on the other hand remain constant around unity at low frequencies and starts falling after ω_B alongside the loop transfer function. At lower frequencies the magnitude of loop shape remains high. The implications of these curves have been discussed previously in section 4.4.6.5 so will not be repeated here.

The peak values of $|T|$ and $|S|$ which provides insights into the robustness properties are presented in Table 4.14. As mentioned earlier, the peaks of $|T|$ and $|S|$ that usually characterize the "water-bed" formation have been smoothed out, and this results in the values of M_T and M_S falling below the stipulated values (of M_T <1.25 and M_S <2).

Table 4.14: Robust Stability Analysis

	Roll & Pitch Rate Loop	Yaw Rate Loop	Altitude Rate Loop
M_T	1.000	1.000	1.230
M_S	1.000	1.000	1.067

Having analysed characteristics of some of the closed loop transfer functions, we will now see how these translate to their responses in time domain.

4.6.3.5 Time Domain Response

The discussions on the time domain response of the nonlinear plant are presented in two parts. In the first part the nominal nonlinear plant is subjected to step response. The responses as well as the force generated by each propeller (the inputs), as a response to the controller outputs are analyzed. In the second part of the study the same complex manoeuvre used in analyzing MSO controller (see section 4.4.6.6), is used to study the effect of plants with uncertainty in the parameters in order to replicate an actual drone trajectory.

(Note: The amount of noise injected into the simulation may differ with different controllers for example MSO, LSDP 1 DoF etc., in sections where the controller development is presented. The quantity of noise would depend on the amount each individual controller can accommodate without inducing instability or deterioration in performance. Disturbances that are *equal* in magnitude will be introduced in the Performance Comparison section (see section 4.9) where comparisons between the various developed controllers are presented.)

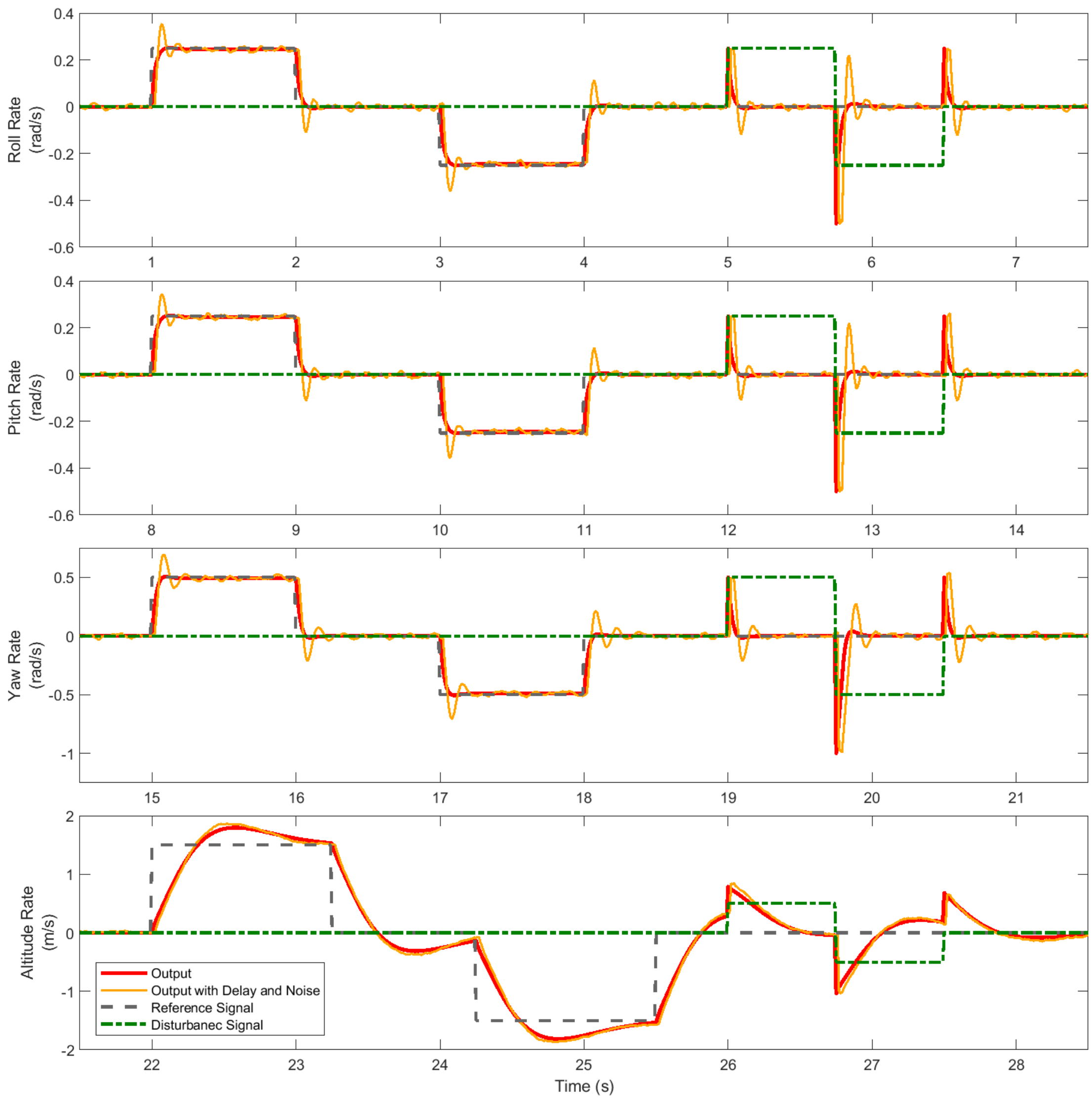


Figure 4.22: Step response of the nominal LSDP 1-DOF controlled plant.

In Fig. 4.22 the response towards square pulse input signals to the plant are presented. The input as well as the disturbance signals have been decoupled in time in order for the responses of each loop and the effects of others on it, to be analysed independently. Fig. 4.23 shows the forces generated by each propeller.

Two sets of simulations are conducted. In the first set the nominal plant is subjected to a noise and delay free input signal. In the second set a delay of two samples are introduced at the output and a zero mean noise signal, with a standard deviation of 15% of the maximum value of the reference input is injected at the output. The Signal-to-Noise Ratio (SNR) of the measured output signals are calculated in to be 17.284 dB, 17.374 dB, 17.473 dB and 12.799 dB in the roll, pitch, yaw and altitude rate loops respectively.

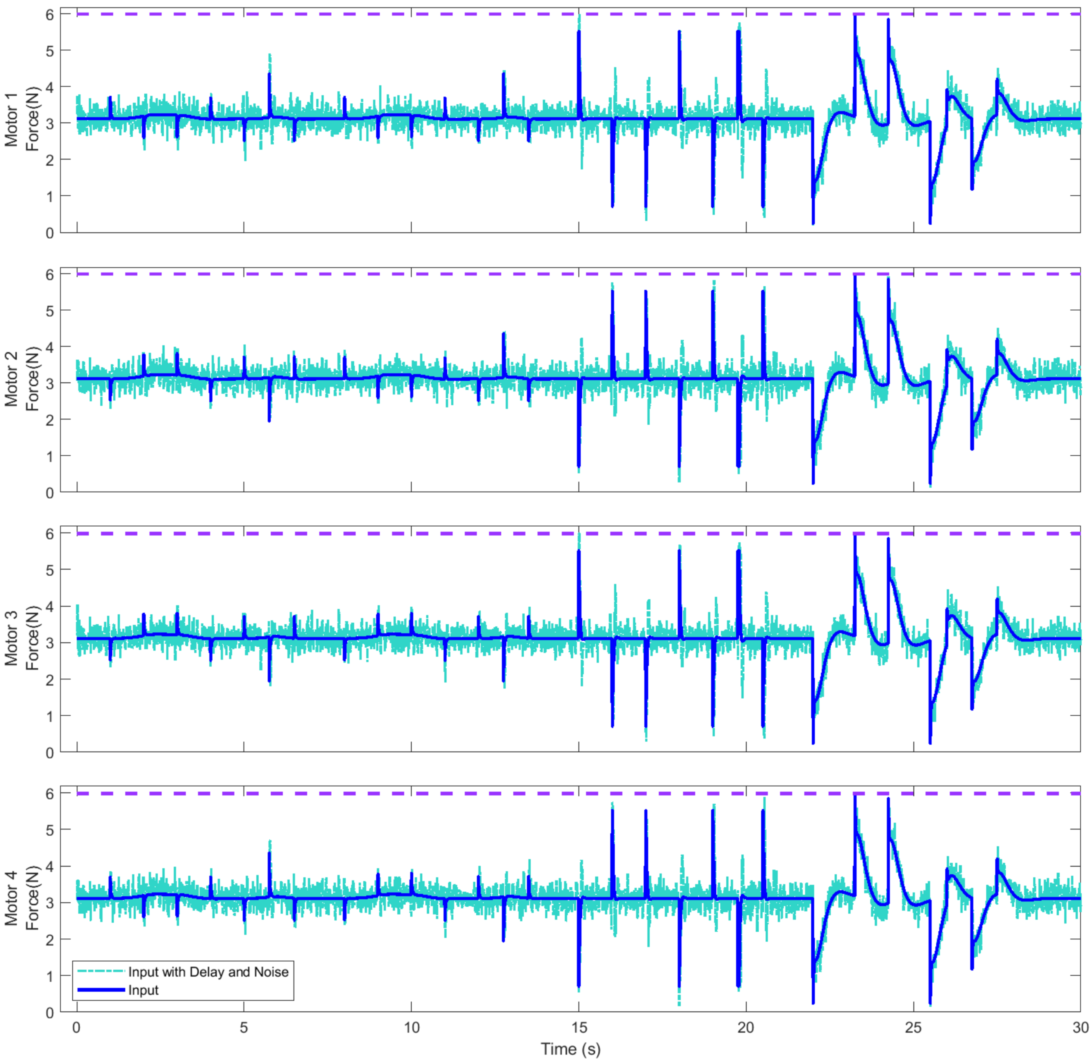


Figure 4.23: Forces exerted by individual motors in response to reference signal

Examining the figures the following points can be noted :

1. Acceptable performances in terms of short rise times, acceptable settling times and peak overshoots can be observed in all the four loops for the case of the delay and noise free model response. The step response characteristics calculated based on this response signals are presented in Table 4.15.

2. The case where a delay as well as a noise signal is present, oscillations arising primarily due to the presence of the delay block can be observed except for in the altitude rate loop. The altitude rate loop which had the largest delay margin of 13 time samples remain oscillation free in the face of the introduced delays.

3. Good immunity to noise signals (even with low SNRs) can be observed in the output responses.

4. The input forces generated in the second case with the injected noise signal, reach the upper saturation limits, the effects of which are negligible on the output response.

5. Subsequent simulations show that the system with a delay of two samples remain stable even at SNRs of below 10. When the delay in increased to three samples, roll, pitch and yaw rates show significant oscillations while altitude rate loop remain oscillation free.

The step response characteristics based on delay and noise free response of the nonlinear nominal plant presented in Fig. 4.22 are given in Table 4.15. For the definitions of response specifications see section 4.4.6.7. The peak values listed in the table should in seen in the context of the maximum values of the reference signal, being 0.25 rad/s for roll and pitch rate, 0.5 rad/s for yaw rate and 1.5 m/s for altitude rate.

Table 4.15: Time and frequency domain specifications of the non-linear plant model after controller design

Response Characteristics	Roll & Pitch Rate	Yaw Rate	Altitude Rate
Rise Time (s)	0.050	0.046	0.249
Settling Time (s)	0.079	0.068	1.182
Peak	0.251	0.505	1.796
Peak Time (s)	0.120	0.100	0.580
% Overshoot	0.528	0.985	19.740
Undershoot	0.000	0.000	0.000

Having examined the responses of the of the nominal plant, we now analyze the response towards plants with model uncertainty. In LSDP since the controller is designed such that it tries to maximize the coprime factor uncertainty limit in the model that the controller can stabilize effectively, to analyze the robustness the model uncertainty is increased until the responses being to show oscillatory behaviour. These responses are presented in Fig. 4.24 and the forces (inputs) generated by individual motors as a response to the reference inputs in Fig. 4.25.

The model uncertainty limits employed to generate the plots are given in Table 4.16. These limits are different from those presented in Table 3.2. And the reason for this is that depending on the nature of the controller, these limits can be stretched in order to accommodate a greater degree of uncertainty in certain parameter. Modeling uncertainty that are *equal* in magnitude will be used in the Performance Comparison section (see section 4.9), where comparisons between the various developed controllers are presented.

Table 4.16: The parameter uncertainty limits of the quadrotor used for testing the controller, with the results presented in Figures 4.24 and 4.25.

Parameters	% Uncertainty
I_{XX}, I_{YY}, I_{ZZ}	$\pm 50\%$
m	$\pm 50\%$
dx	$\pm 40\%$
dy	$\pm 40\%$
c_t	$\pm 50\%$
c_p	$\pm 30\%$

Similar to previous Monte-Carlo plots we have the extreme plants output responses in Fig. 4.24 coloured in yellow, while the inputs in Fig. 4.25 coloured in light blue.

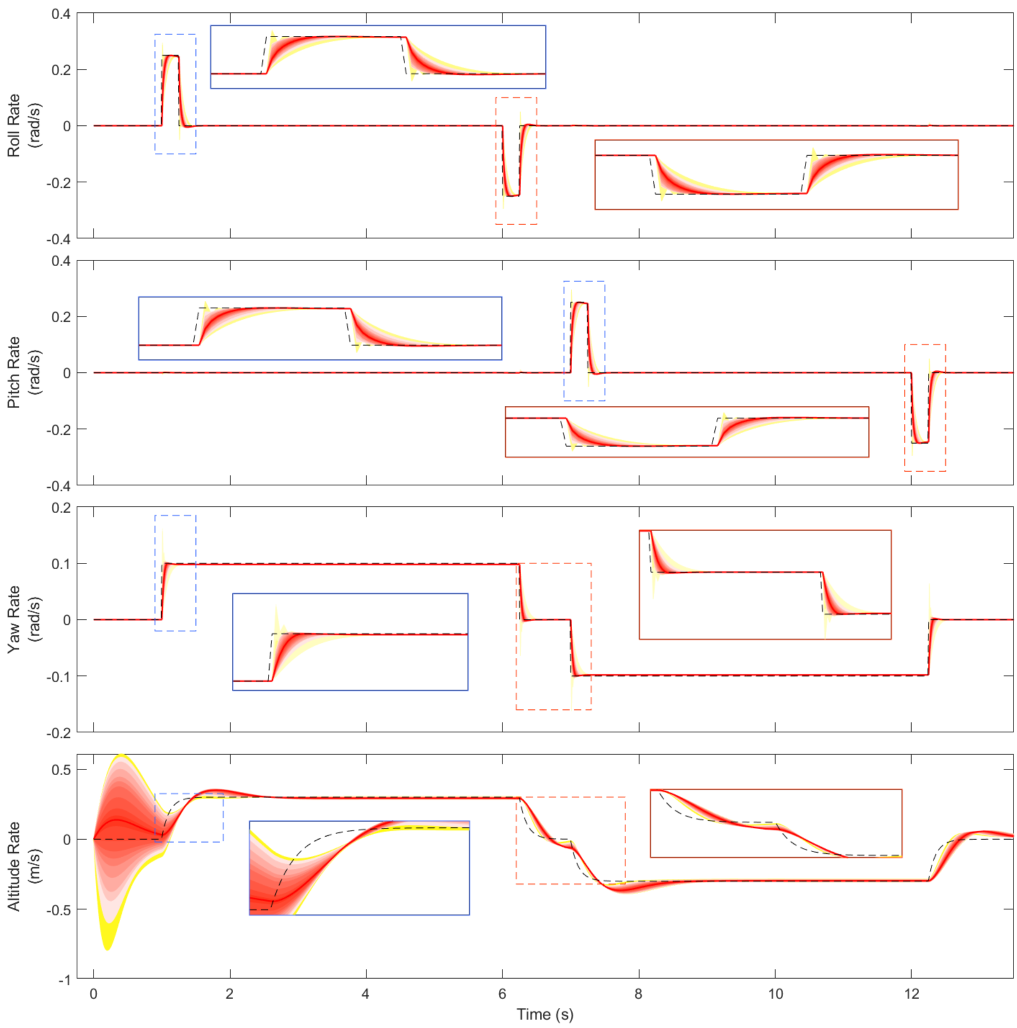


Figure 4.24: The response of the LSDP 1-DOF based plant with model uncertainty to the complex manoeuvre reference command.

The response of the plants surrounding the nominal plant in the interior of those in the hypercube (i.e. those with parameters marked in red in Fig. 4.12), are in turn marked in red in Fig. 4.24 and dark blue in Fig. 4.25. The reference command is represented by black dashed lines.

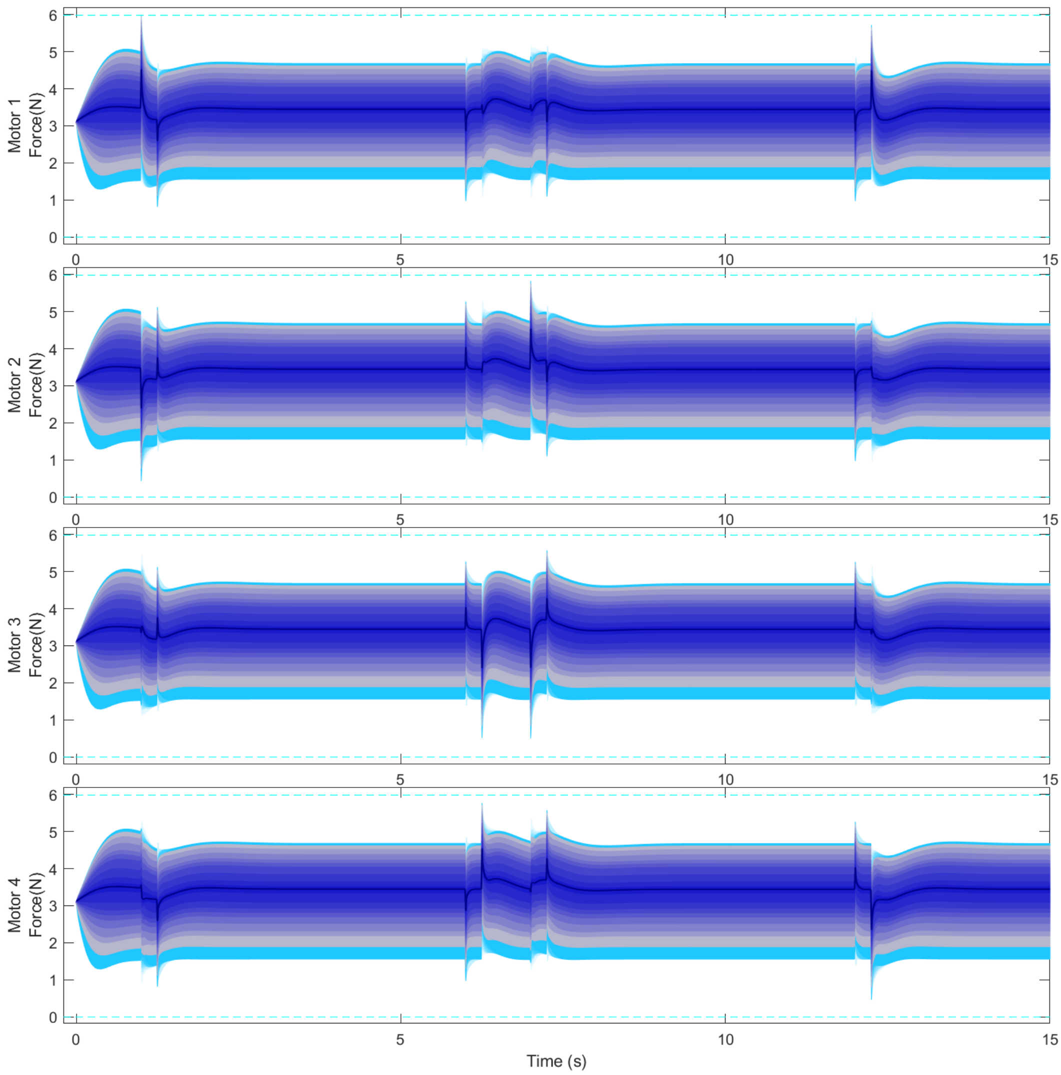


Figure 4.25: LSDP 1-DOF controlled plant rotor responses to the complex manoeuvre reference signal.

The following points can be noted from the two plots.

- In the output response plot (Fig. 4.24), for the plants within the hypercube, overshoots are absent in the roll, pitch and yaw rates, while the extreme plant responses (those at the hyper cube corners) exhibit overshoots, and overshoots can be observed in altitude rates for all models in general.
- Oscillatory responses are absent in roll, pitch, yaw and altitude rate responses for the plants within the hypercube. The extreme plant responses exhibit

oscillations for the case of roll, pitch and yaw rates. Oscillatory response are absent in the case of altitude rates for extreme plants.

- Input forces (Fig. 4.25) generated as a response to the reference commands fall below the saturation limits.
- The fluctuation observed in altitude rates characterized by a blob in the first couple of seconds of the simulation (as explained earlier is the response of MSO and PID controlled loops), happens as a response towards the uncertainty in mass and settles down as the system stabilizes.

Having examined the responses of plants containing model uncertainty in the time domain, the analysis is now presented as observed from the frequency domain. The performance criteria explained in section 4.6.2 is used in this analysis.

4.6.3.6 Robustness Analysis

The normalized coprime stability margins calculated for each of the SISO loops alongside the lower limits of the classical stability margins (SM) and phase margins (PM) are presented in Table 4.17.

Table 4.17: Coprime stability margin and classical gain & phase margins

Response Characteristics	Roll & Pitch Rate	Yaw Rate	Altitude Rate
Normalized Coprime Stability Margin	0.694	0.663	0.614
Classical SM Lower Limit	5.537	4.933	4.178
Classical PM Lower Limit (deg)	87.899	83.039	83.039

The values in the first row gives the limit of uncertainty that the controller can accommodate in each loop, before the loop becomes unstable. This uncertainty is measured as the distance between the nominal model and the uncertain model in the gap metric criterion. Hence the controller K that stabilizes the nominal model G_0, can stabilize an uncertain model G_{unc} if the following condition holds

$$\delta_\nu(G_0, G_{unc}) < b(G_0, K) \tag{4.47}$$

The normalized coprime stability margins fall in the acceptable range of >0.25, hence from the point of view of robustness, the controller designs are considered successful. This brings us to the end of the single degree of freedom LSDP controller design. In the following section the two degree of freedom LSDP are explored.

4.7 Two Degree of Freedom LSDP

The 2-DOF configuration enables providing an additional handle on the model matching capabilities for the control engineer. When strict model matching becomes the primary motive for the controller design the 2DOF LSDP is employed to design the robust controller [103].

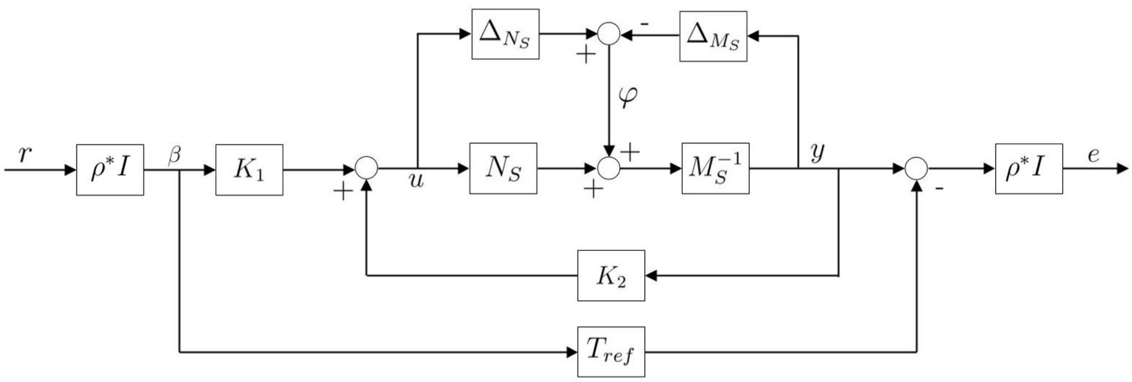


Figure 4.26: Two degrees of freedom $\mathcal{H}_\infty$ loop shaping design problem

The 2-DOF LSDP problem can be seen in Fig. 4.26. Here T_{ref} is the model that the engineer is trying to match, while β is the reference input after scaling and ρ^* ($\geq$ 1) takes a value depending on whether preference is given towards model matching over robustness. Higher values of ρ^* (generally $\rho^* \leq 3$) put a greater emphasis on model matching. The subscript in the plant $G_S = M_S^{-1} N_S$ and in coprime factor uncertainty uncertainty Δ_{N_S} and Δ_{M_S} indicates the shaped plant $G_S = G W_1$.

In the designed controller $K = [K_1 \; K_2]$, K_1 plays the role of the prefilter while K_2 forms the feedback controller. K aims to minimize the $\mathcal{H}_\infty$ norm of the transfer function of the signals from $\left[r^T \; \varphi^T\right]^T$ and $\left[u^T \; y^T \; e^T\right]^T$. With $G_s = [A_s B_s; C_s D_S]^T$ and $T_{\text{ref}} = [A_r B_r; C_r D_r]^T$, the system matrix P can be defined as below

$$\begin{bmatrix} u_s \\ y \\ e \\ \hdashline \beta \\ y \end{bmatrix} = \left[\begin{array}{cc|cc:c} A_s & 0 & 0 & (B_S D_S^T + Z_S C_S^T) R_S^{-1/2} & B_S \\ 0 & A_R & B_R & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & I \\ C_S & 0 & 0 & R_S^{-1/2} & D_S \\ \rho^* C_S & -{\rho^*}^2 C_R & -{\rho^*}^2 D_R & \rho^* R_S^{-1/2} & \rho^* D_S \\ \hdashline 0 & 0 & \rho^* I & 0 & 0 \\ C_S & 0 & 0 & R_S^{-1/2} & D_S \end{array}\right] \begin{bmatrix} r \\ \varphi \\ \hdashline u_s \end{bmatrix} \tag{4.48}$$

Here $R_S = I + D_S D_S^T$, $S_S = I + D_S^T D_S$ and Z_S can be obtained by solving the matrix Riccati equation

$$(A_S - B_S S_S^{-1} D_S^T C_S) Z_S + Z_S (A_S - B_S S_S^{-1} D_S^T C_S)^T - Z_S C_S R^{-1} C_S Z_S + B_S S_S^{-1} B_S^T = 0$$

(Refer to [182] and [184], chapter 18, for derivation). The controller K_2 aims for internal robust stability while K_1 enables model matching by ensuring

$$||(I - GK_2^{-1}GK_1 - T_{ref})||_\infty \leq \gamma {\rho^*}^{-2} \tag{4.49}$$

To eliminate the steady state error and thereby match the closed loop to the reference model T_{ref}, the reference signals can be scaled by W_i, a constant. This alongside the plant shaping weight W_1, and the model matching factor ρ^*, can be generated using the **Algorithm #3** explained in section 4.7.1.

In the traditional design procedure, once form and values of parameters of the weights are decided the $\mathcal{H}_\infty$ controller is obtained through γ iterations. After completion of the design of the controller K ($[K_1\ K_2]$), the controller-plant system is implemented as shown in Fig. 4.27

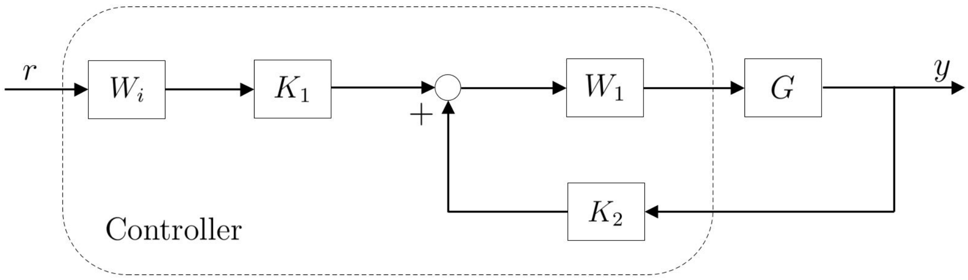


Figure 4.27: Two degree of freedom LSDP controller implementation

As shown in the figure 4.27, both the precompensator and pre filter is incorporated into the controller structure in the implementation phase.

In this study the three quantities W_i, W_1 and ρ^* are designed with the help of an outer frame work that works towards satisfying the time domain specifications required from the controller. The development of these weights are discussed in the next section.

4.7.1 Choosing the design weight W_1, prefilter W_i and model matching parameter ρ_*

Similar to selection of weights described in section 4.6.1, the selection of the design weight W_1, the prefilter W_i and model matching parameter ρ^* can be formulated into the following problem.

For the plant configuration presented in Fig. 4.26 find $\tilde{W} = (W_1, W_i)$ and ρ^* such that

$$\gamma(\tilde{W}, \rho^*) \leq \epsilon_\gamma \tag{4.50}$$

and

$$\Phi_k(\tilde{W}, \rho^*) \leq \epsilon_k \tag{4.51}$$

alongside the controller K that stabilizes the nominal plant, while maintaining design optimality. Here γ is $\mathcal{H}_\infty$ norm of the matrix $\mathbf{P}$ from equation 4.48. If required, one can also constrain γ_{min} from equation 4.31 such that it is ≤ 4. In this study this is not considered as it has been observed that sufficiently minimizing $\mathcal{H}_\infty$ of $\mathbf{P}$, limits the normalized co-prime stability margins to be ≥ 0.25. Φ_k represent the closed loop functional required to be constrained. ϵ_γ and ϵ_k are real numbers that represents the limits for γ and Φ_k respectively.

The prefilter W_i and the ρ_* are simple scalars while W_1, once again takes the form of a transfer function as below:

$$W_1 = \frac{w_1(s + w_2)}{s + w_3} \tag{4.52}$$

The parametrised weights can now be developed using **Algorithm #3**. The optimization problem central to the algorithm maintains the design optimality.

Algorithm # 3 Weight, prefilter, model matching parameter selection and controller design for a 2 DOF LSDP

Inputs: Plant model G, design constraints.
Outputs: K, W_1, W_i, ρ_*.

1. Scale the given plant G according to the criteria described in section 3.3
2. Define the appropriate form for W_1, W_i and choose the initial values of the parameters (see 4.7.1).
3. Choose the simple plant model T_{ref} which reflects the desired closed loop response and initialize the value of the model matching parameter ρ^*.
4. Define the cost function $\mathcal{J}$ (see eq. (4.17)) and initialize the weights W1 though W4 (see section 4.4.3 for details regarding development of the cost function).
5. Develop the vector $\mathbf{\Phi}$ which contains the following nonlinear closed loop functionals
   1. Maximum overshoot, M_p
   2. Rise time, t_p
   3. Settling time, t_s
   4. Maximum value of input, u_{max}
   5. Gamma, γ

   Here M_p, t_p and t_s, the step response characteristics and u_{max} are the time domain specifications while γ refers to the frequency domain specification.
6. Define the frequency domain bounds ϵ_γ and time domain specifications limit vector $\boldsymbol{\epsilon}$.

---

7. Implement the constrained non-linear optimization problem. This is achieved using the OPTI MATLAB toolbox. The black box optimizer NOMAD is chosen as the optimization routine. Nonlinear constraints are defined based on ϵ_γ and $\boldsymbol{\epsilon}$ for the vector $\mathbf{\Phi}$ that defines the closed loop functionals.
8. Calculate the highest obtainable stability margin $1/\gamma_{\min}$. If this value is less than .25, vary the forms of the weights W_1 (see 4.18), as well as relax the limits of the linear constraints and repeat step 7 again.

On successful implementation and execution of the algorithm the design weights, parameters and the controller are obtained. The algorithm can be implemented in MATLAB or similar software with toolboxes that support the NOMAD optimization package. For implementation in MATLAB the following points can be noted.

1. During initial runs, the time and frequency domain constraints can be relaxed. These can be tightened as successful iterations are completed.
2. T_{ref} is chosen based on requirements in design specifications section 3.5.
3. The Matlab routine `hinfsyn` is called in step 7 to develop the controller.
4. ITSE has been used in step 4, for developing the cost function. Among the various error performance indices (IAE, ITAE, ISE etc.) ITSE has been found to facilitate the fastest convergence.

In the next section the performance criteria that will be used to check the design effectiveness is examined.

4.7.2 Performance Criteria

Since co-prime factor uncertainty is the source of uncertainty considered in the model, the criteria used will be the gap-metric stability margin, similar to that used to assess the controllers in 1-DOF, (see section 4.6.2).

This brings us to the end of the LSDP 2-DOF design procedure. The development of the attitude and altitude controllers for the quadrotor will be discussed in the next section.

4.7.3 Attitude-Altitude controller for a Quadrotor

The attitude-altitude controller of the decoupled quadrotor developed using the above discussed 2-DOF LSDP strategy is presented here. The section is divided into different parts. In the first part the optimization problem is stated. The reference models, the loop shaping weights, the prefilter and the model matching parameter ρ^* are then presented. Thereafter the time domain response characteristics and frequency domain specification of the linear models are examined. This is followed through by examining the classical stability margins, the loop shape and the closed loop bandwidth. Sensitivity and complementary sensitivity function plots are then presented.

In the final part of the section, the responses of the non-linear nominal model and that with model uncertainty are examined followed by the robust stability margins of the closed loop system.

4.7.3.1 Optimization Problem

As mentioned in **Algorithm #3** the weights, the prefilter and the model matching parameter selection are carried out by resolving an optimization problem. The framework explained in section 4.4.6.2 is used to develop the problem. The only difference is the constraint limits set for γ - the $\mathcal{H}_\infty$ norm of matrix $\mathbf{P}$ from eq. (4.48) and for ρ^* the model matching parameter. The lower limit for γ is set as a soft bound of 0.25 and the upper limit is set as a hard bound of 3. The upper limit is taken from a practical stand point, so that the co-prime factor stability margin falls above 0.25. For ρ^*, the lower limit is set at 1 and the upper limit is set at 3. A total of 5 decision parameters are being searched for, these being 3 for the loop shaping weight W_1, 1 for the prefilter W_i and 1 for ρ_*.

4.7.3.2 Reference Model and Design Weights

The reference model M_0 is selected such that each of it satisfies the basic design specifications given in section 3.5. The reference models are selected to be first order transfer functions. They are as follows:

$$M_{0_{\text{Roll and Pitch Rate}}} = \frac{1}{0.0567s+1} \quad , \quad M_{0_{\text{Yaw Rate}}} = \frac{1}{0.0315s+1} \quad , \quad \text{and} \quad M_{0_{\text{Altitude Rate}}} = \frac{1}{0.1134s+1} \tag{4.53}$$

The developed loop shaping weight W_1, the prefilter W_i and model matching parameter ρ^* are listed below: Roll and pitch rate

$$W_1 = 0.1860\frac{s+1.0753\cdot 10^{-4}}{s+0.0978}, \quad W_i = 1.3043, \quad \rho^* = 2.0957 \tag{4.54}$$

Yaw Rate:

$$W_1 = 0.1836\frac{s+0.0053}{s+0.2981}, \quad W_i = 3.4095, \quad \rho^* = 1.2016 \tag{4.55}$$

Altitude Rate:

$$W_1 = 1.119\frac{s+3.8052}{s+0.01177}, \quad W_i = 1.1472, \quad \rho^* = 2.7617 \tag{4.56}$$

The controllers themselves are presented in Appendix D.1. The time and frequency domain specification of the closed loop system is given in the table below:

Table 4.18: Time response characteristics and frequency domain specification of the linear closed loop model

Closed Loop Constraints	Roll & Pitch Rate controller	Yaw Rate controller	Altitude Controller
% Overshoot	0.00	0.00	3.84
Rise Time (s)	0.10	0.06	0.13
Settling Time (s)	0.24	0.11	0.55
u_{max}	0.37	0.36	1.00
γ	2.56	1.93	3.16

The step responses are obtained by feeding in a normalized unit reference input. As seen from Table 4.18, all the time and frequency domain specifications lie within the acceptable range of values. The stability characteristics of the closed loops are analysed next.

4.7.3.3 Classical Stability Margins and Nominal Stability

To calculate the stability margins, the open loop transfer function defined as the ratio of the feedback signal to the actuating error signal, in this case is equal to the product between the shaped plant and the feedback controller K_2. The classical stability margins are presented in the Table 4.19. The controller and the plant is discretized at a frequency of 100Hz (sample time = 0.01s).

Table 4.19: Classical Stability Margins and Crossover Frequencies

Stability Margins	Roll & Pitch Rate Loop	Yaw Rate Loop	Altitude Rate Loop
GainMargin (abs)	6.433	20.221	10.360
GMFrequency (rad/s)	312	312.25	314.159
PhaseMargin (rad)	1.411	1.536	1.282
PMFrequency (rad/s)	31.912	11.093	20.377
DelayMargin (s)	0.044	0.138	0.063
DMFrequency (rad/s)	31.912	11.093	20.377
Closed loop Stability	Stable	Stable	Stable

The following points can be noted regarding the values presented in the Table 4.19.

1. The gain and phase margins of all the loops lie within the acceptable limits, i.e. GM >2 and PM >30° or 0.52 rad.
2. The gain crossover frequencies all lie around 312 rad/s which can be confirmed by observing the plots in Fig. 4.28.

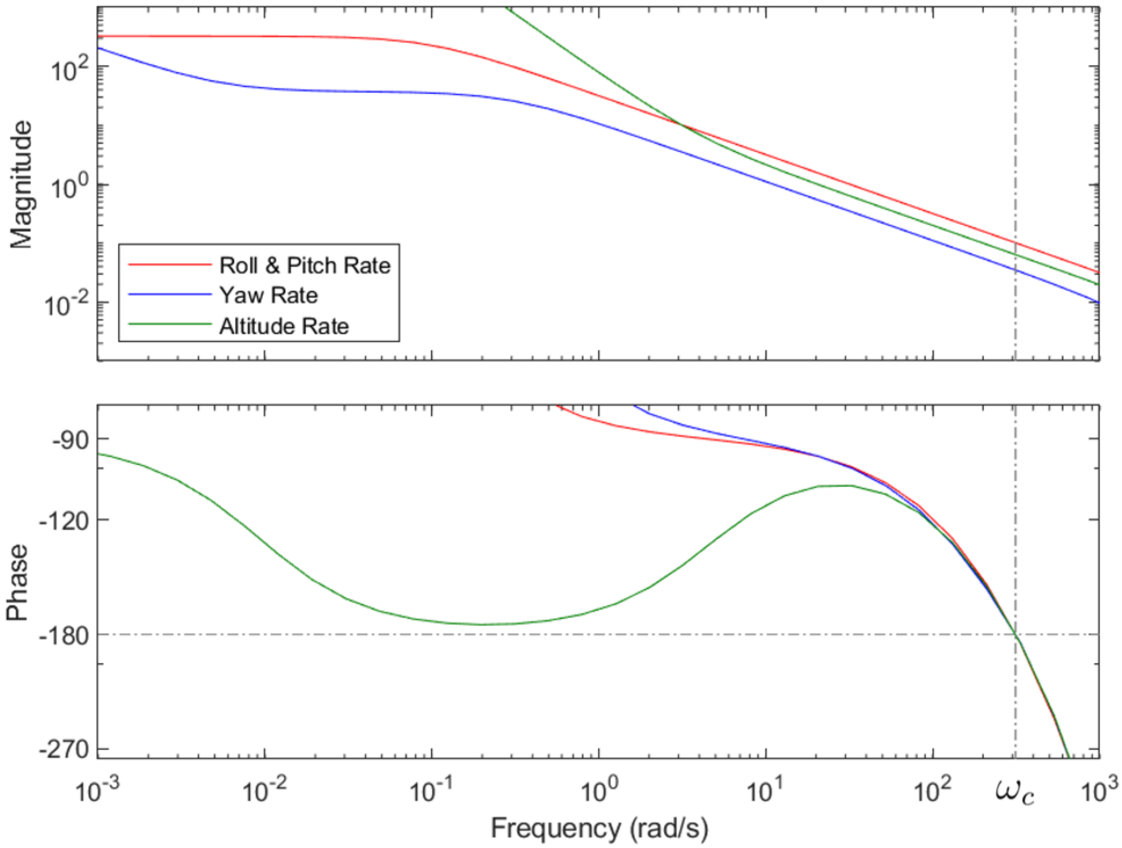


Figure 4.28: LSDP 2-DOF shaped plant: Bode plot of $L(j\omega)$.

3. The yaw rate loop is the most immune towards instabilities due to delays, with a delay margin of 0.138 seconds, followed by the altitude rate and then by the roll and pitch rate loops.

4. As noted in the table, nominal stability is achieved by all of the loops.

As observed in Fig. 4.28 the loop shapes suggest that the yaw rate loop has the least bandwidth and the highest bandwidth are for roll and pitch rate loops. The details of the closed loop bandwidth are examined in the next section.

4.7.3.4 Closed Loop Bandwidth

Before presenting the closed loop bandwidth, let us recall the definitions of sensitivity and complimentary sensitivity functions. Sensitivity can be defined as the closed loop transfer function from output disturbance to the output, while complementary sensitivity function can be defined as the closed loop transfer function between the reference to the outputs.

Hence for the 2DOF positive feedback loop configuration, (see Fig. 4.27), the sensitivity and complementary sensitivity are obtained as follows:

$$S = \frac{1}{1 - G_S\,K_2} \quad \text{and}$$
$$T_F = T \cdot K_1 = \frac{G_S\,K_2\,K_1}{1 - G_S\,K_2} \tag{4.57}$$

(Note: The complimentary sensitivity function defined as the closed loop transfer function from reference signal to output, assumes the form of T_F eq. (4.57), as opposed to T, where subscript $_F$ indicates presence of a pre-filter in the control system. To use Bode's original definition of the term Sensitivity function (refer [192]) we have,

$$S = \frac{d\mathcal{T}/\mathcal{T}}{d\mathcal{G}/\mathcal{G}}, \tag{4.58}$$

where $\mathcal{T}$ represents the complimentary sensitivity function, and $\mathcal{G}$, the plant model. Consider the transfer function

$$G_{yr} = \frac{G_S\,K_2\,K_1}{1 - G_S\,K_2} \tag{4.59}$$

where G_{yr} is the transfer function from reference signal to outputs as derived from Figure 4.27, with $W_i = 1$. Now differentiating G_{yr} with respect to the plant model G_S we get:

$$\frac{dG_{yr}}{dG_S} = \frac{K_2\,K_1}{1 - G_S\,K_2} - \frac{-G_S\,K_2\,K_1\,K_2}{(1 - G_S\,K_2)^2} = \frac{K_2\,K_1}{(1 - G_S\,K_2)^2} = S\frac{G_{yr}}{G_S} \tag{4.60}$$

Hence the sensitivity function S for Fig. 4.27 is derived as:

$$S = \frac{dG_{yr}/G_{yr}}{dG_S/G_S} \tag{4.61}$$

Comparing equations 4.58 and 4.61, we could see that in order to derive the sensitivity function as defined by Bode, we need to consider the transfer function T_F instead of T which forms the rationale behind choosing T_F ($\equiv G_{yr}$) as the performance indicator (as opposed to T) in the subsequent performance analysis. The transfer function T_F, also features as one among the *Gang of Six* important transfer functions that captures the properties of a control system alongside the conventional complimentary function T, as defined by Karl Johan Aström, in [193].)

The closed loop bandwidth (see section 4.4.6.5) for the loops are presented in Table 4.20.

Table 4.20: Closed Loop Bandwidth

Loops	ω_b (rad/s)
Roll & Pitch Rate loop	31.900
Yaw Rate Loop	11.380
Altitude Rate Loop	16.500

As shown from the curves in Fig. 4.28, the yaw rate loop has the lowest closed loop bandwidth suggesting a slower response to reference signals. The closed loop bandwidth had been calculated based on the values from the plots in Fig. 4.29.

Observing the plots, the following points can be noted.

1. The inherent peaks of the S and T_F curves have been smoothed out by the controller action.
2. From the bandwidth frequency indicated by the dotted grey lines to around 10^3 rad/s magnitude of $|T_F|$ falls at an almost equal pace for all the three loops after which the rate of descent increases substantially, indicating a effective noise cancellation at higher frequencies.
3. The magnitude of the open loop remains high at lower frequencies, consistently around the bandwidth frequency and decreases at higher frequencies.
4. The values of the $|S|$ and $|T_F|$ curve peaks are presented in Table 4.21. The values of the peaks fall well below the stipulated values (of M_T <1.25 and M_S <2), with M_T falling below M_S in each of the loops.

Table 4.21: Robust Stability Analysis

	Roll & Pitch Rate Loop	Yaw Rate Loop	Altitude Rate Loop
M_T	0.985	0.975	1.011
M_S	1.003	1.005	1.003

The implications of the curves have been discussed previously in section 4.4.6.5 hence are not be repeated here.

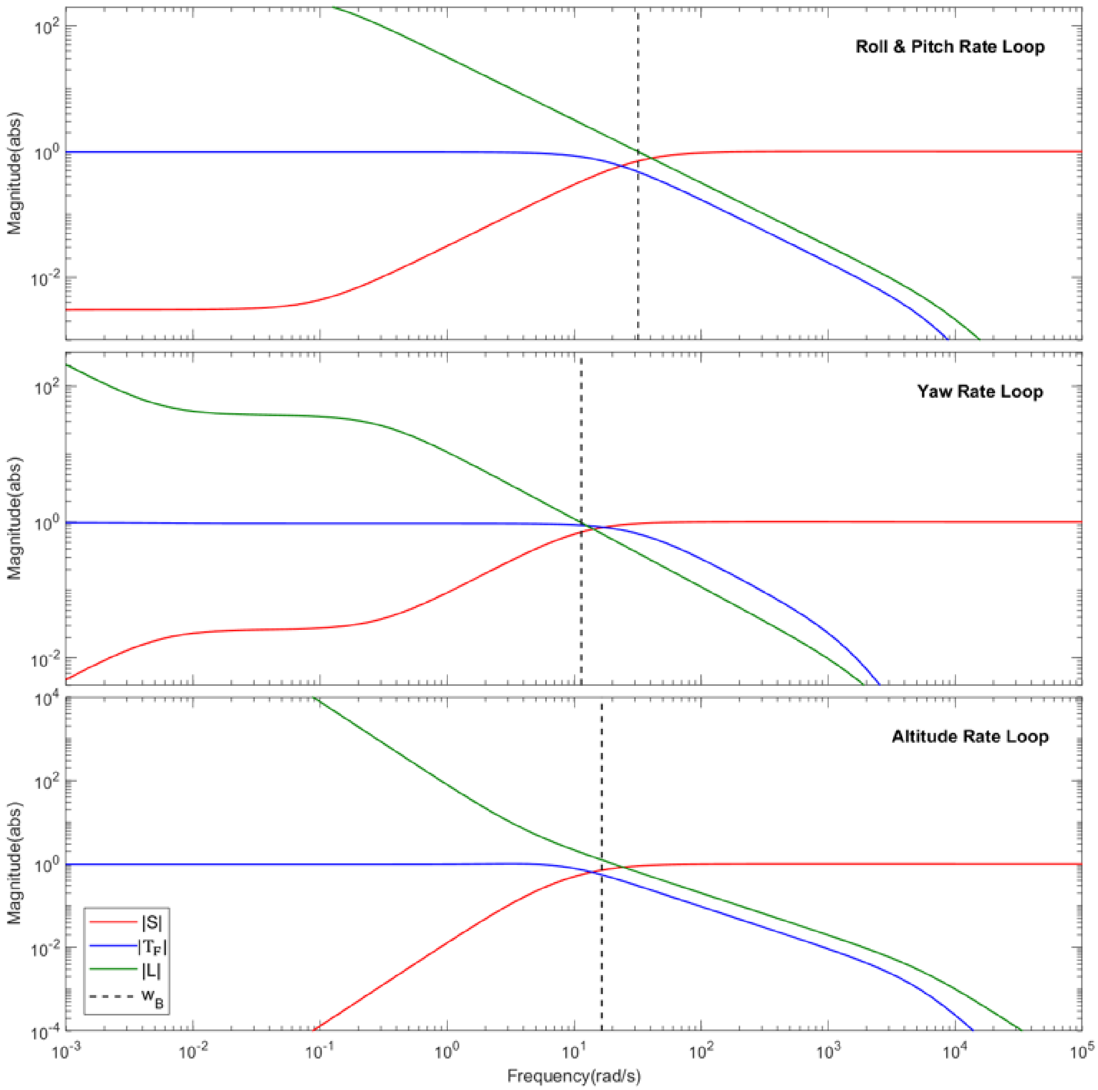


Figure 4.29: LSDP 2-DOF system: Bode magnitude plots of $|S|$, $|T_F|$ and $|L|$

In the next section the time domain responses of the plant towards reference and disturbance inputs are analysed .

4.7.3.5 Time Domain Response

The closed loop time domain response of the non-linear plant is discussed in this section. The section is presented in two parts. In the first part the response of the plant towards unit step inputs both in the absence and presence of noise and time delays are presented. In the second part, response of plants containing model uncertainty is examined.

The response towards a step reference input signal is presented in Fig. 4.30. The inputs as well as disturbances are injected such that the effects are separated. This enables to observe the plant responses in each case separately.

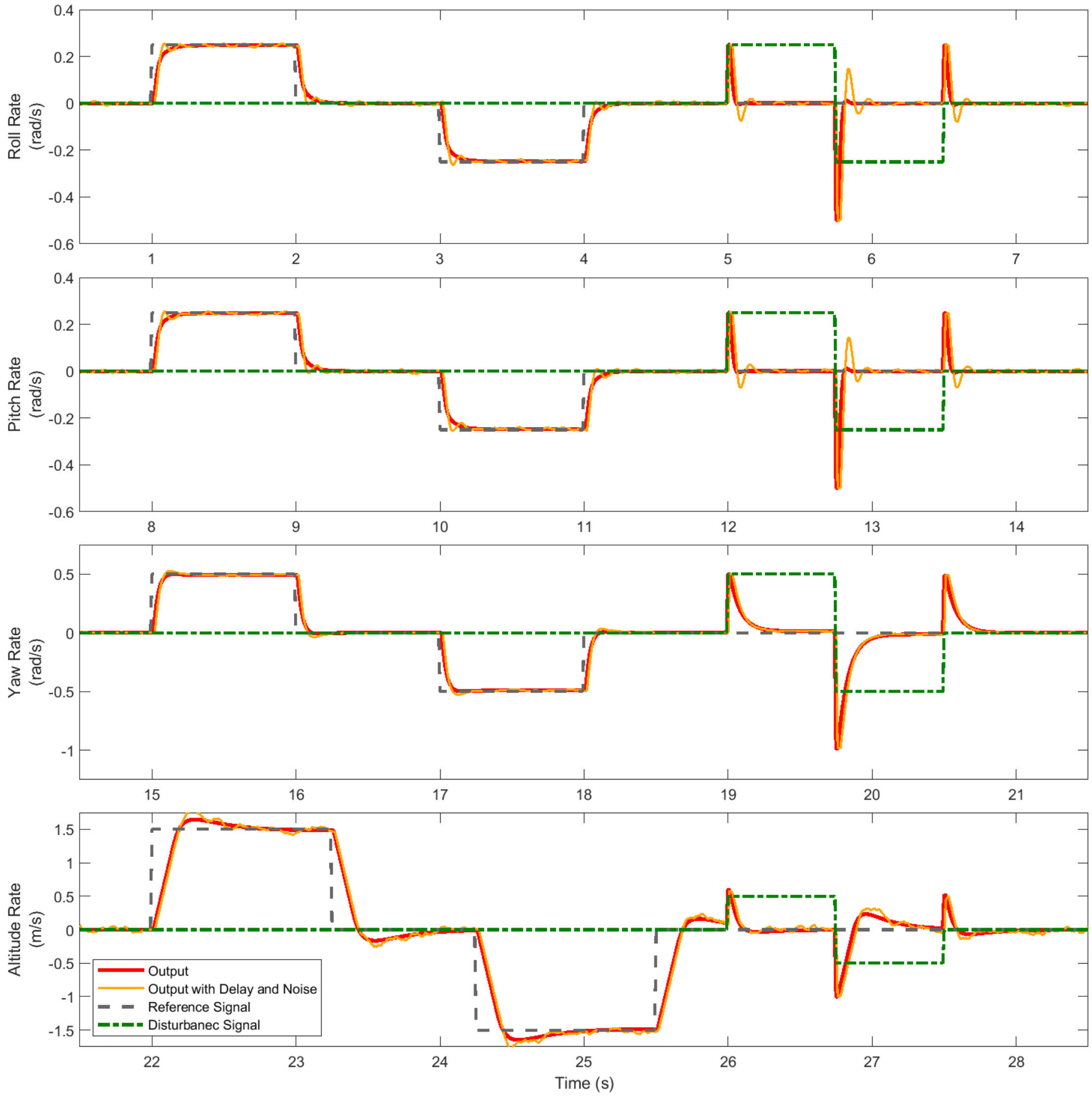


Figure 4.30: Step response of the nominal LSDP 2-DOF controlled plant

The inputs, or forces generated by each propeller responding to the reference input are presented in Fig. 4.31. As in previous cases of controller analysis (see section 4.4.6.6 and section 4.6.3.5) two sets of simulations are performed, i.e. with and without noise and time delays.

In the second set of simulations a delay of a single sample time alongside a zero mean noise signal with a standard deviation 10% of the maximum of the reference signal, injected at the output is included. The SNR of the output measured signals are calculated to be 20.575 dB, 20.707 dB, 20.948 dB, 15.704 dB in the roll, pitch, yaw and altitude rate loops respectively.

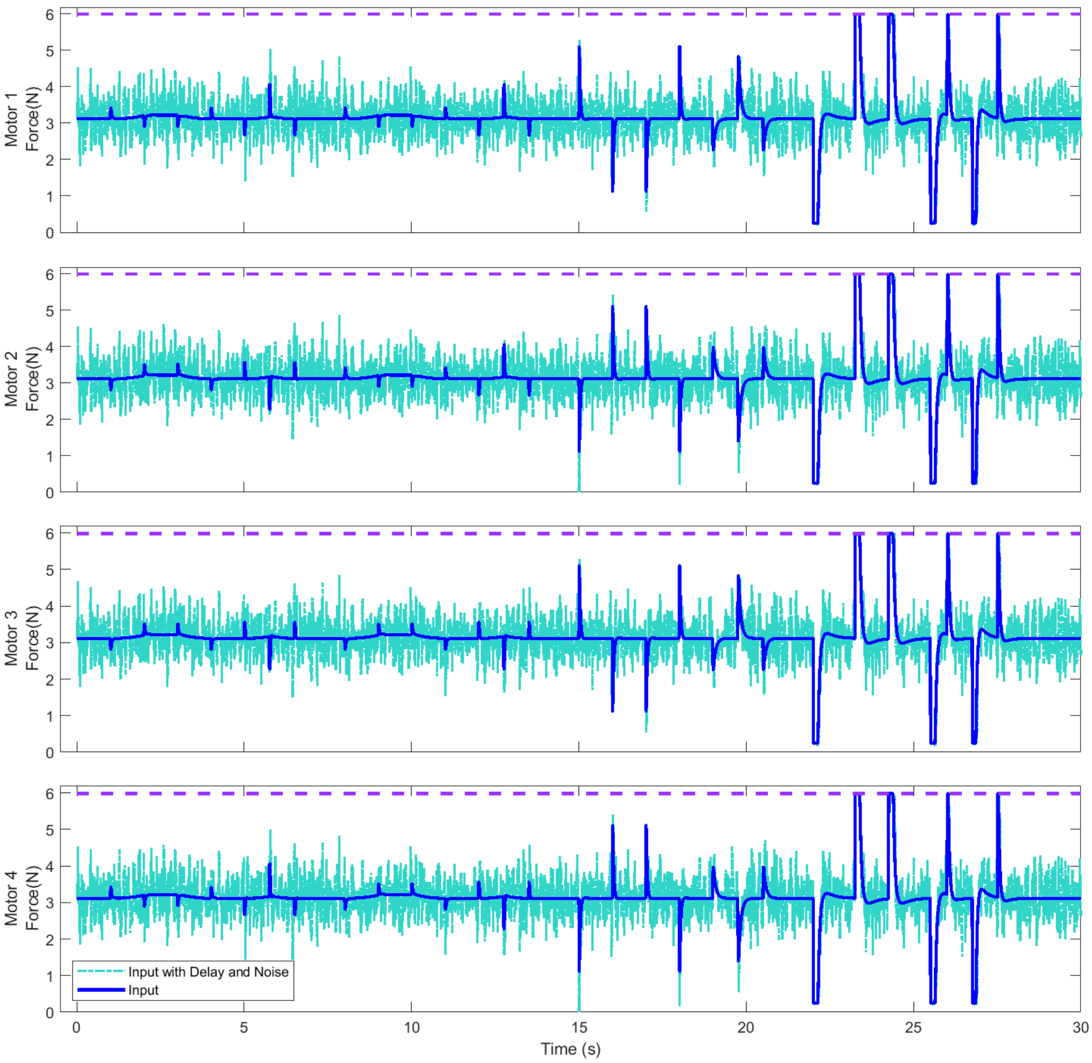


Figure 4.31: Forces exerted by individual motors in response to reference signal

Examining figures 4.30 and 4.31 and from conducting simulations the following points can be noted.

1. Acceptable rise times and settling times are observed in all the four loops. Overshoots are absent in the roll, pitch and yaw rate loop responses to the reference inputs for the noise and delay free simulation. Altitude rate shows a slight overshoot. The step response characteristics are presented in Table 4.22.

2. For the case with the presence of the unit time delay and noise, oscillations can be observed in the roll, pitch, and altitude rate responses, while yaw rate response exhibit no oscillations. The oscillation are brought about by the delay in the loop. In the roll, pitch, and altitude rate loops the oscillations increase substantially with a delay of two sample times, while yaw rate loop exhibit good immunity which can be explained by its larger delay margins.

3. Good noise immunity to noise signal (even with SNR <10) can be observed in the output responses.

4. The input signals are seen to reach the saturation limits although its effect on the outputs are negligible.

The step response characteristics based on delay and noise free response of the nonlinear nominal plant presented in Fig. 4.30 are given in Table 4.22. For the definitions of response specifications see section 4.4.6.7.

The peak values listed in the table should be seen in the context of the maximum values of the reference signal, being 0.25 rad/s for roll and pitch rate, 0.5 rad/s for yaw rate and 1.5 m/s for altitude rate.

Table 4.22: Closed loop time and frequency domain specifications from the nonlinear plant model response

Response Characteristics	Roll & Pitch Rate	Yaw Rate	Altitude Rate
Rise Time (s)	0.096	0.060	0.134
Settling Time (s)	0.235	0.110	0.659
Peak	0.248	0.498	1.646
Peak Time (s)	0.990	0.160	0.300
% Overshoot	0.000	0.000	9.711
% Undershoot	0.000	0.000	0.000

Having met the required performance objectives of the nominal non-linear plant model, now the the responses of the plant with model uncertainties will be examined. Like the case of one degree of freedom, the controller tries to maximize co-prime factor uncertainty in the model that it can stabilize effectively.

The uncertain nonlinear plant is subjected towards the same complex manoeuvre used in section 4.4.6.10 and section 4.6.3.5. The percentage parameter uncertainty limits used in modelling the uncertain nonlinear plant are presented in Table 4.23.

Table 4.23: Parameter uncertainty limits of the quadrotor used for testing the controller

Model Parameters	% Uncertainty
I_{XX}, I_{YY}, I_{ZZ}	$\pm 40\%$
m	$\pm 50\%$
dx	$\pm 40\%$
dy	$\pm 40\%$
c_t	$\pm 50\%$
c_p	$\pm 30\%$

The Monte-Carlo plots of the output responses are presented in Fig. 4.32. The significance of the colours used in the plots have been explained previously (see section 4.6.3.5), hence will not be repeated here.

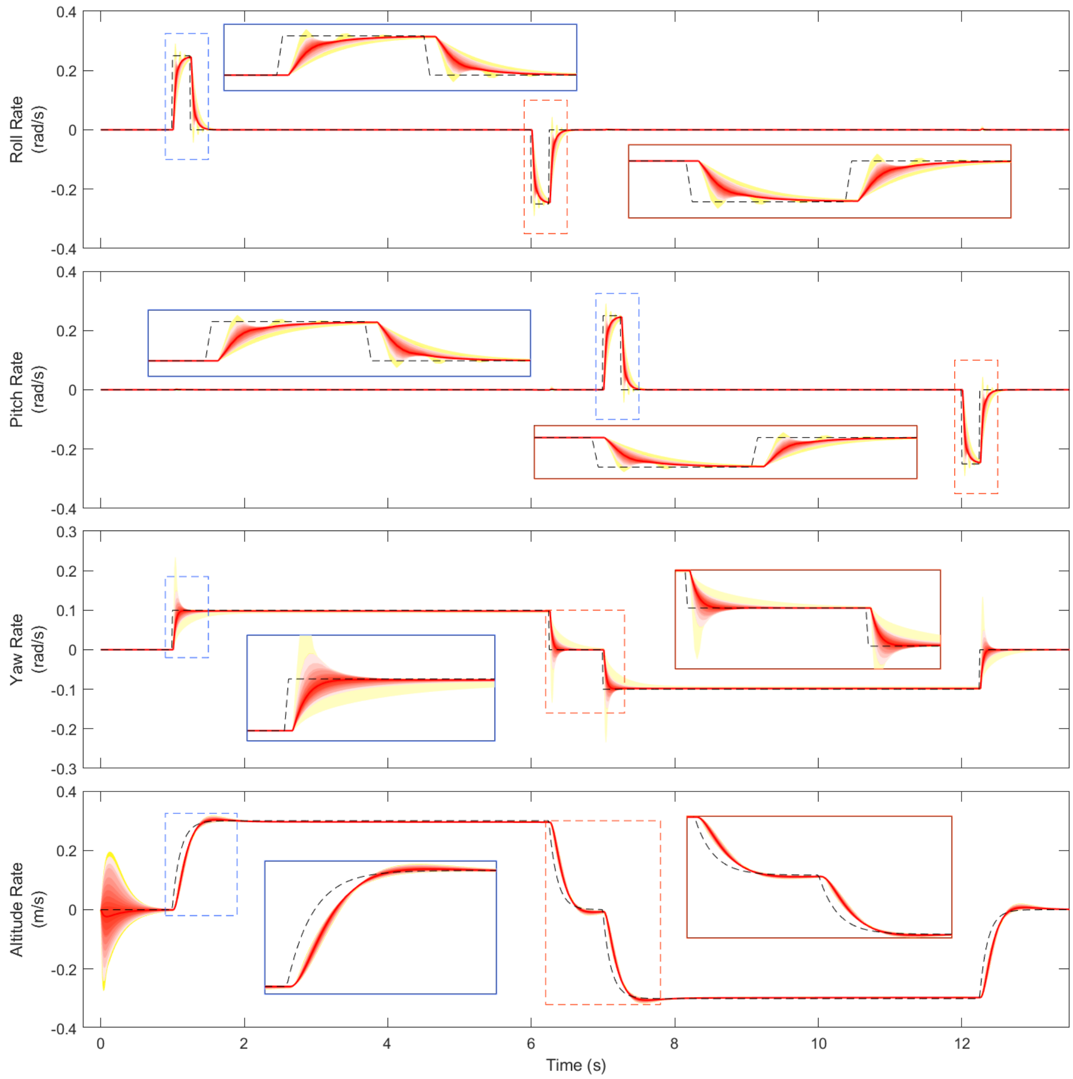


Figure 4.32: The response of the LSDP 2-DOF based plant with model uncertainty to the complex manoeuvre reference command.

The forces produced by each motor as a response to the reference input is presented in 4.33.

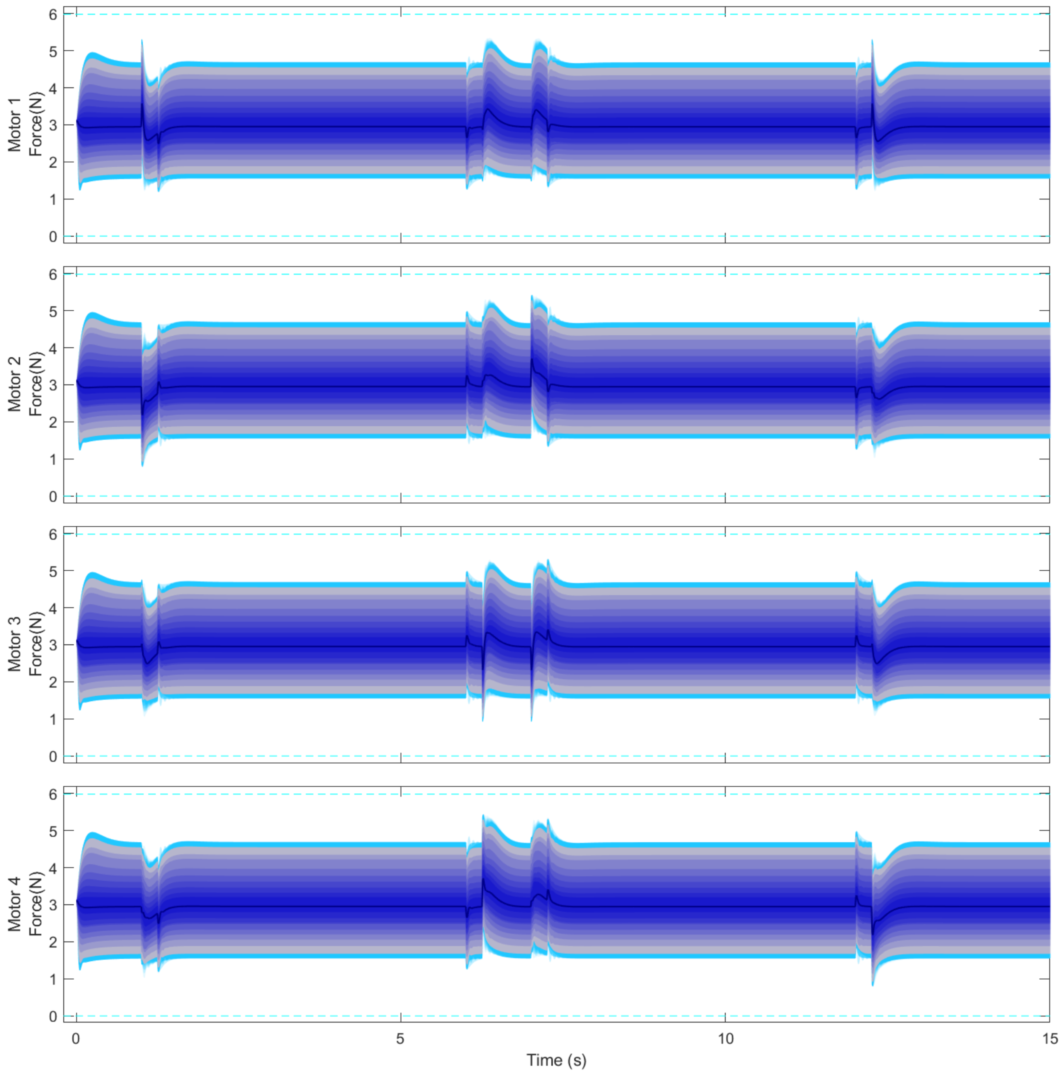


Figure 4.33: LSDP 2-DOF controlled plant rotor responses to the complex manoeuvre reference signal.

The following points can be noted from examining the figures.

- In the output response plot, (Fig. 4.32), slight overshoots are present in the roll, pitch and altitude rates for the extreme plants (plots marked in yellow). Overshoots can also be observed in the yaw rates response for the extreme plants.
- Oscillatory responses are present in roll and pitch rate responses. These are absent in yaw and altitude rate responses.
- Spikes can be noticed after the 6th second in pitch rate response and after the 7th second in roll rate response in Fig. 4.32 corresponding to the saturation

of inputs as seen in Fig. 4.33 at those respective times.

Having observed the responses of the uncertain nonlinear plants in the time domain and having recorded satisfactory responses, we now look at the frequency domain robustness aspects of the controller.

4.7.3.6 Robustness Analysis in the Frequency Domain

The normalized coprime stability margins calculated for each of the SISO loops alongside the lower limits of the classical stability margins (SM) and phase margins (PM) are presented in Table 4.24.

Table 4.24: Coprime stability margin and classical gain & phase margins

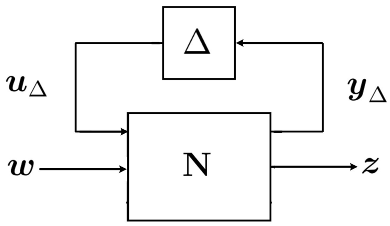
Response Characteristics	Roll & Pitch Rate	Yaw Rate	Altitude Rate
Normalized Coprime Stability Margin	0.412	0.604	0.494
Classical SM Lower Limit	2.403	4.049	2.953
Classical PM Lower Limit (deg)	48.699	74.294	59.214

The coprime stability margins are greater than 0.25 hence from the point of view of robustness the controller designs can be considered as successful. The stability margin provides enough guarantee against uncertain plant models as seen from the perspective of the gap metric criterion (see eq. (4.47)). This brings us to the end of the two degree of freedom LSDP controller design. In the next section μ-Synthesis controller designs will be explored.

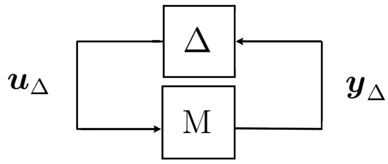
4.8 μ-Synthesis

In this section the controller design using μ-synthesis will be examined. In Table 4.6 of the MSO design we had observed that the controller fell short of guaranteeing robust performance to the plant in the presence of input multiplicative model uncertainty. Whilst the plant is represented as in Fig.4.1, the μ-synthesis problem can be pictured as aiming to tackle this shortfall.

From eq. (4.6) we have defined **F** as the upper LFT of **N** and Δ. This N-Δ structure can be observed in Fig. 4.34(a).



(a) The N-Δ Structure.



(b) The M-Δ Structure.

Figure 4.34: μ-synthesis motivation.

We can also observe from the eq. (4.6) that for a nominally stable system the source of instability arises from the term $(I - N_{11}\Delta)^{-1}$. Hence to assess the robust stability and performance we begin analysing the stability of the M-Δ structure (see Fig. 4.34(b)), where M equals N_{11}. With this given M-Δ structure the robust stability criteria can now be stated as follows: For a stable nominal system M(s) and a stable convex set of perturbations Δ with $\|\Delta\|_\infty \leq 1$, robust stability is achieved if and only if

$$\det(I - M\Delta(j\omega)) \neq 0 \quad \forall\omega, \forall\Delta. \tag{4.62}$$

(See eq. 8.104 and further section 8.5 in [118]). The uncertainty Δ in eq. (4.62) can be both unstructured (full-block matrix) or structured (diagonal matrix). Controller designs based on unstructured uncertainty can be generally more conservative compared to structured uncertainty. To take advantage of the fact that parameter uncertainty, delay uncertainty and non-linearity can be expressed in terms of structured uncertainty, the parameter μ can be used in the design [194]. The structured singular value of μ can be defined as follows:

For a given complex matrix M and a block diagonal complex uncertainty matrix Δ (i.e. $\Delta = \text{diag}\ \{\Delta_i\}$ where some of the Δ_i can be real or repeating), the real non-negative function μ(M) is given by

$$\mu(M) \triangleq \frac{1}{\min\{k_m \,|\, \det(I - k_m M\Delta) = 0 \text{ for structured}\, \Delta,\, \bar{\sigma}(\Delta) \leq 1\}} \tag{4.63}$$

The larger the value of μ, the smaller the value of Δ that can make the quantity $(I - M\Delta)$ singular. Having defined the function μ, we can now look at how it can be used to assess the robust stability and robust performance of a system with model uncertainty.

For the N-Δ structure from Fig. 4.34 with a nominally stable system M, and real or complex stable perturbations Δ, with $\bar{\sigma}(\Delta) \leq 1, \forall\omega$, robust stability is achieved if and only if $\mu(M(j\omega)) \leq 1$, $\forall\,\omega$.

Similarly robust performance for the same system is achieved if and only if $\mu_{\widehat{\Delta}}(N(j\omega)) \leq 1\,\forall\,\omega$. Here $\widehat{\Delta}$ represents the diagonal structure,

$$\widehat{\Delta} = \begin{bmatrix} \Delta & 0 \\ 0 & \Delta_P \end{bmatrix} \tag{4.64}$$

Here Δ is a pure diagonal matrix, while Δ_P is always a full block matrix. Δ_P is a perturbation block that is associated with performance weight w_P (see [118] pg. 316-319). While μ-synthesis can be used to analyse the stability and performance characteristics, a controller that strives to reduce the value of μ can also be developed. Although currently the problem of resolving a controller that minimizes μ remains unsolved, an iteration procedure named as the D-K iteration from [195] gives good results. With the matrix $\mathbf{D}$, chosen as any block diagonal matrix that commutes with $\boldsymbol{\Delta}$, the over all objective of D-K iteration can be stated as to find the controller that minimizes the function

$$\min_{\mathbf{K}}(\min_{\mathbf{D}\in\boldsymbol{\mathcal{D}}} \left\|\mathbf{D}\,\mathbf{N}(\mathbf{K})\,\mathbf{D}^{-1}\right\|_\infty) \tag{4.65}$$

by switching consecutively between the controller $\mathbf{K}$ and matrix $\mathbf{D}$. Here $\mathcal{D}$ represents the set of block diagonal matrices whose structure is compatible with that of $\boldsymbol{\Delta}$. The iteration process has three important steps.

1. K-Step: By holding $\mathbf{D}$ constant a $\mathcal{H}_\infty$ controller that resolves the problem $\min_{\mathbf{K}} \left\|\mathbf{D}\,\mathbf{N}(\mathbf{K})\,\mathbf{D}^{-1})\right\|_\infty$ is developed.
2. Using the developed controller $\mathbf{K}$ i.e, by holding $\mathbf{N}$ constant find the $\mathbf{D}(j\omega)$ that minimises $\bar{\sigma}(\mathbf{D}\,\mathbf{N}\,\mathbf{D}^{-1})$.
3. The magnitude of $\mathbf{D}(j\omega)$ is fitted to a minimum phase transfer function. The iteration now begins again starting in step 1.

While the minimization problem in steps 1 and 2 are individually convex, they may or may not be jointly convex. This lack of guarantee sometimes results in the iterative optimization problem getting stuck in a localized optima. But overall, in general the D-K iteration has proved to give good results (see [118] pg. 328-338).

4.8.1 Design weights w_I and w_P in μ synthesis

The development of the design weights used in μ-synthesis to an extent is similar to that of MSO control in 4.4.2. There are minor differences however. The control problem can be restated as to finding a stabilizing optimal controller for the control problem in Fig. 4.3 alongside the weight $\tilde{W} = (w_I, w_P)$ such that

$$\gamma(\tilde{W}) \leq \epsilon_\mu \tag{4.66}$$

and

$$\Phi_i(\tilde{W}) \leq \epsilon_i, \tag{4.67}$$

where ϵ_μ represents the upper bound on $\gamma(\tilde{W})$, which is the value of function presented in eq. (4.65), developed in the context of the weights w_I and w_P. The weight w_I is developed such that it captures model uncertainty while the performance weight w_P is developed by the outer framework that encapsulates the D-K iteration algorithm. The performance weight w_P takes the format as that specified in eq. (4.16).

The outer framework as stated earlier (i.e, as in the development of the MSO control strategy), resolves an optimization problem to generate both the performance weight and a controller that satisfies the performance specifications. The cost function for the optimiser is similar to the one explained in section 4.4.3, except for weight W1, which represents the penalty on the value of μ (instead of γ). The algorithm for developing the μ-controller is presented next.

4.8.2 Algorithm for μ-Controller development

To obtain the performance weight alongside the μ-controller that satisfies the required time and frequency domain specifications the following algorithm can be followed:

Algorithm # 4 Sensitivity weight and Controller design procedure for μ-Synthesis

Inputs: Plant model G, design constraints, model uncertainty estimate.
Outputs: K, w_i, w_p.

1. Scale the given plant G according to the criteria described in section 3.3
2. Calculate the complementary sensitivity weight w_I according to the procedure described in section 4.4.2.1.

3. Define the appropriate form of w_P and choose the initial values of the parameters.
4. Define the cost function $\mathcal{J}$ and initializes the weights W1 through W4.
5. Develop the vector $\mathbf{\Phi}$ which contains the following nonlinear closed loop functionals
   1. Maximum overshoot, M_p
   2. Rise time, t_p
   3. Settling time, t_s
   4. Maximum value of input, u_{max}
   5. μ(N)

   Here M_p, t_p and t_s, the step response characteristics and u_{max} are the time domain specifications while μ(N) refers to the frequency domain specification.
6. Define the frequency domain bounds ϵ_μ and time domain specifications limit vector $\boldsymbol{\epsilon}$.
7. Implement the constrained non-linear optimization problem. This is achieved using the OPTI MATLAB toolbox. The black box optimizer NOMAD is chosen as the optimization routine. Nonlinear constraints are defined based on ϵ_μ and $\boldsymbol{\epsilon}$ for the vector $\mathbf{\Phi}$ that defines the closed loop functionals.
8. Depending on robustness requirements modify the weights W1 through W4 to reach the required performance levels.
9. If the performance specifications are not met, decrease the % uncertainty considered while defining w_I and relax the bounds defined in step 6 and retry step 7.

Existing commercial software such as MATLAB can be used to develop the algorithm. While implementing and executing **Algorithm #4** the following points should be noted.

1. The time and frequency domain specifications can be relaxed in the initial runs. These can be tightened as successful iterations are completed.
2. The Matlab routine `dksyn` (part of the Robust Control Toolbox) is used in step 7 to develop the controller.
3. ITSE has been used in step 4, for developing the cost function. Among the various error performance indices (IAE, ITAE, ISE etc.) ITSE has been found to facilitate the fastest convergence.

The performance of the developed controllers can be measured once they are designed and tested. The criteria to be followed is given in the next section.

4.8.3 Performance Analysis Criteria

The closed loop stability and performance of the system can be analysed with the help of the criteria listed in Table 4.25. The criteria follows naturally when we have a performance requirement of $\|F\|_\infty \leq 1$

Table 4.25: Performance Criteria - μ-Synthesis

Performance Criteria	Necessary Constraint
Nominal Stability	$\mathbf{N}$ internally stable
Nominal Performance	$\bar{\sigma}(N_{22}) = \mu_{\Delta_P} < 1;\ \forall\omega$, and NS
Robust Stability	$\mu_{\Delta}(N_{11}) < 1;\ \forall\omega$, and NS
Robust Performance	$\mu_{\widehat{\Delta}}(\mathbf{N}) < 1;\ \forall\omega, \widehat{\Delta} = \left[\begin{smallmatrix}\Delta & 0\\ 0 & \Delta_P\end{smallmatrix}\right]$, and NS

Consider the system in Fig.4.35. The system is modelled with multiplicative uncertainty and the performance weight w_P is present at the output.

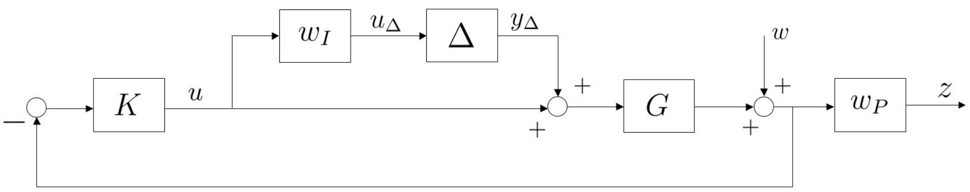


Figure 4.35: System modelled with multiplicative uncertainty

Converting the system into the N-Δ structure presented in Fig. 4.34.(a) by analysing the closed loop system from the inputs $[u_\Delta \quad w]^T$ to outputs $[y_\Delta \quad z]^T$, we obtain N as below:

$$\mathbf{N} = \begin{bmatrix} -W_I KG(I+KG)^{-1} & -W_I K(I+GK)^{-1} \\ W_P G(I+KG)^{-1} & W_P(I+GK)^{-1} \end{bmatrix} \tag{4.68}$$

and M $= N_{11} = W_I T_I$, where the negative sign is ignored as we have $\mu(\mathbf{N}) = \mu(\mathbf{U}\,\mathbf{N})$ with $\mathbf{U} = \left[\begin{smallmatrix}-I & 0\\ 0 & I\end{smallmatrix}\right]$. T_I represents the input complementary sensitivity function. The performance criteria in Table 4.25 for the system in Fig 4.35 is hence restated in Table 4.26.

(In the final criteria w.r.t to RP, calculating $\mu_{\widehat{N}}$ follows directly from the following identity: when a, b, c, and d are complex scalars and $\Delta = \mathrm{diag}\{\delta_1\ \delta_2\}$ we have $\mu\left[\begin{smallmatrix}ab & ad\\ bc & cd\end{smallmatrix}\right] = |ab| + |cd|$. It can also be noted that the μ-Synthesis performance criteria is exactly same as that of the MSO criteria presented in Table 4.1.)

Table 4.26: Performance Criteria - μ-Synthesis for the system in Fig.4.35

Performance Criteria	Necessary Constraint
Nominal Stability	**N** internally stable $\equiv$ S, SG, KS and T_I being stable
Nominal Performance	$\bar{\sigma}(N_{22}) = \|w_P S\| < 1;\ \forall\omega$
Robust Stability	$\mu_{\Delta}(N_{11}) = \mu(M) = \|w_I T_I\| < 1;\ \forall\omega$
Robust Performance	$\mu_{\widehat{\Delta}}(\mathbf{N}) = \|w_P S\| + \|w_I T_I\| < 1;\ \forall\omega$

This brings us to the end of the design and performance analysis procedure of μ-synthesis for single input single output systems. In the following section tracking altitude-attitude rate controllers for the quadrotor model explained in section 4.2 will be presented.

4.8.4 Attitude-Altitude controller for a Quadrotor

The attitude-altitude controller developed using the μ-Synthesis procedure for the decoupled quadrotor is presented in this section. Initially the complementary sensitivity weights w_I, which capture the model uncertainty, are presented. The optimization problem is presented next followed by the sensitivity weights alongside the step response characteristics, which are derived by resolving the optimization problem.

An analysis of the designed controllers is then presented. The classical stability margins, peak values of closed loop functions and the closed loop bandwidth are examined in the frequency domain. In the time domain the responses to a step signal is examined. The performance of plants with model uncertainty are also examined. In the final section the Robust performance of the plant using the designed controller will be examined.

4.8.4.1 Complementary Sensitivity weight w_I

The complementary sensitivity weight (similar to those developed for MSO) absorbs the model uncertainty. For sake of avoiding repetition, the procedure to develop w_I will not be explained (refer to 4.4.2.1). The % parameter uncertainty absorbed into the weights are based on the values presented in Table 4.27

Table 4.27: Parameter uncertainty limits of the quadrotor used for testing the controller

Parameters	% Uncertainty
I_{XX}, I_{YY}, I_{ZZ}	$\pm 10\%$
m	$\pm 30\%$
dx	$\pm 30\%$
dy	$\pm 30\%$
c_t	$\pm 30\%$
c_p	$\pm 30\%$

Following **Algorithm #4**, the developed weights are listed below:

$$\begin{aligned} w_{I,11} = w_{I,22} &= \frac{0.4044s + 0.4027}{s + 0.9954} \\ w_{I,33} = \frac{0.9684s + 0.9643}{s + 0.9954} \quad &, \quad w_{I,44} = \frac{0.4278s + 0.4260}{s + 0.9954} \end{aligned} \tag{4.69}$$

4.8.4.2 Optimization Problem

The optimization problem is exactly same as that of one developed in 4.4.6.2 except for the non-linear constraint on μ (instead of γ). μ is bound to be less than 1. The Optimization problem is used to develop the sensitivity weight as well as the controller that achieves robust stability. In the cost function defined in section 4.17 $\mathcal{J}_1$, is redefined as the weight associated with μ, i.e. $\mathcal{J}_1 = W1 \cdot \mu$.

4.8.4.3 Sensitivity weight w_P

The Sensitivity weights developed following **Algorithm #4** are given below.

$$\begin{aligned} w_{P,11} = w_{P,22} &= 0.0003016\,\frac{s + 200}{s + 0.00125} \\ w_{P,33} = 0.0106\,\frac{s + 36.1226}{s + 0.1485} \quad &, \quad w_{P,44} = 0.005812\,\frac{s + 4.9553}{s + 0.07274} \end{aligned} \tag{4.70}$$

The controllers themselves are presented in Appendix D.1. The time and frequency domain specifications obtained while developing the controllers based for the linearised model are given below.

Table 4.28: Time response characteristics and frequency domain specification of the linear closed loop model

Closed Loop Constraints	Roll & Pitch Rate controller	Yaw Rate controller	Altitude Controller
% Overshoot	4.160	4.871	19.857
Rise Time (s)	0.047	0.037	0.225
Settling Time (s)	0.142	0.116	1.542
u_{max}	0.246	0.857	0.757
μ	0.408	0.980	0.430

From the results in the Table is can be deduced that the optimization routine have successfully managed to generate controllers for the loops that satisfy the non-linear constraints alongside providing stability. In the next subsection the classical stability margins provided by the controllers are examined.

4.8.4.4 Classical Stability Margins and Nominal Stability

The classical stability margins of the closed loops are presented in Table 4.29

Table 4.29: Classical Stability Margins and Crossover Frequencies

Stability Margins	Roll & Pitch Rate Loop	Yaw Rate Loop	Altitude Rate Loop
GainMargin (abs)	10.106	9.341	42.170
GMFrequency (rad/s)	136.308	158.647	88.291
PhaseMargin (rad)	1.230	1.242	1.087
PMFrequency (rad/s)	23.346	27.573	5.188
DelayMargin (s)	0.053	0.045	0.209
DMFrequency (rad/s)	23.346	27.573	5.188
Closed loop Stability	Stable	Stable	Stable

The gain and phase margins lie above the acceptable ranges of GM >2 and PM >30°. The nominal stability is established by examining the Nyquist plots of the closed loop systems.

Having established nominal stability alongside acceptable stability margins, the criteria presented in Table 4.26 is now checked.

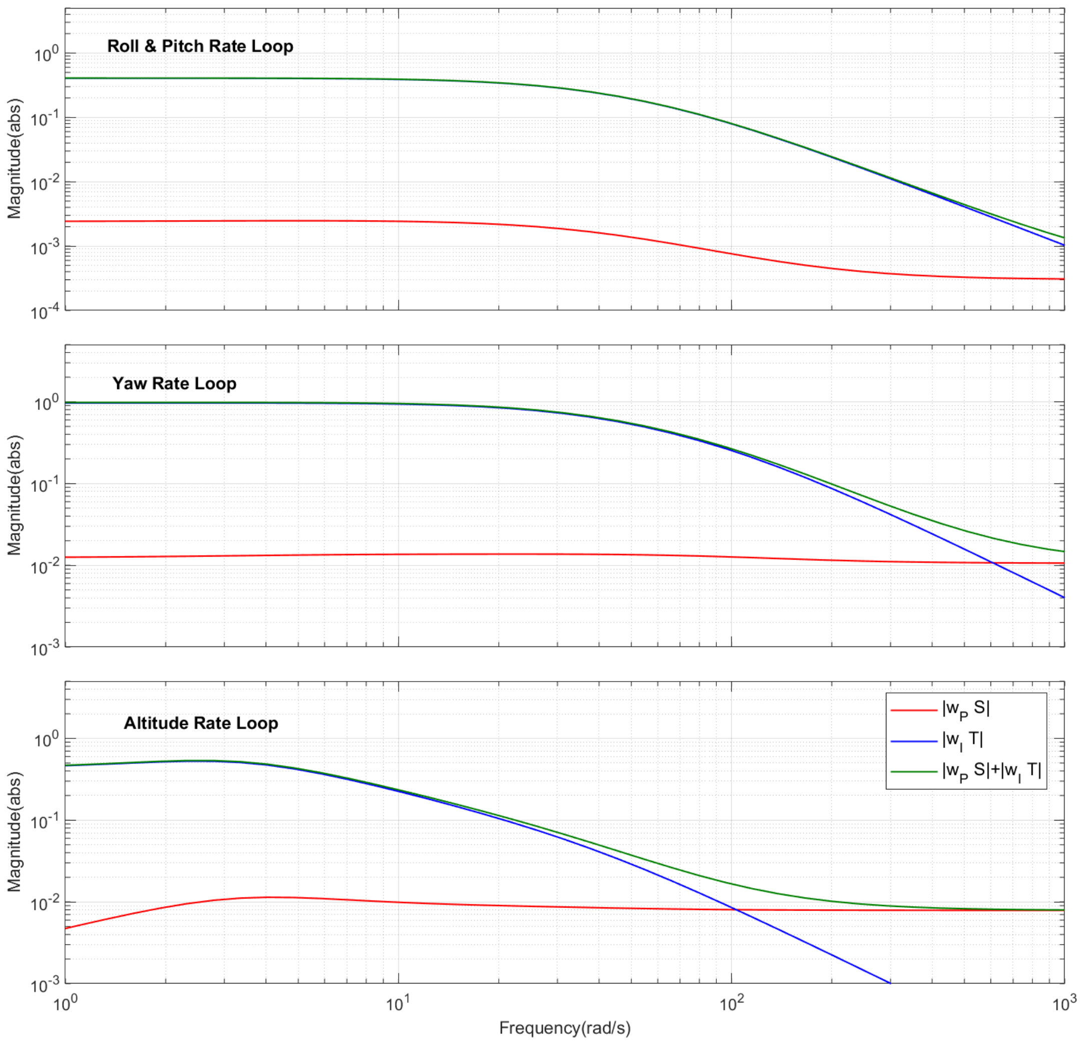


Figure 4.36: Robust stability analysis: Bode magnitude plots of $|w_P\, S|$, $|w_I\, T|$ and $(|w_P\, S| + |w_I\, T|)$

The values in the Table 4.30 are calculated based on the curves of weighted closed loop functions presented in Figure 4.36.

As observed from the values presented in Table 4.30, as well as from the curves in Fig. 4.36 it can be concluded that the system will exhibit nominal performance, robust stability and robust performance in the presence of the considered model uncertainty (as values of $|w_P\, S|$, $|w_I\, T|$ and $(|w_P\, S| + |w_I\, T|)$ fall below unity $\forall\, \omega$).

Table 4.30: Robust Stability Analysis

∞-Norm	Roll & Pitch Rate Loop	Yaw Rate Loop	Altitude Rate Loop
$\|w_P S\|_\infty$	0.002	0.014	0.0111
$\|w_I T\|_\infty$	0.406	0.969	0.526
$\|(\|w_P S\|+\|w_I T\|)\|_\infty$	0.408	0.980	0.535

The closed loop bandwidth is examined next.

4.8.4.5 Closed Loop Bandwidth

The closed loop bandwidth (calculated according to the definition in 4.4.6.5) is presented in Table 4.31

Table 4.31: Closed Loop Bandwidth

Closed loops	ω_b (rad/s)
Roll & Pitch Rate loop	19.605
Yaw Rate Loop	23.950
Altitude Rate Loop	3.435

The values obtained have been arrived at from the Fig. 4.37 and it can be noted that acceptable bandwidths have been obtained. We can see that the the peaks of $|S|$ and $|T|$ have been smoothed out by the controller ensuring better robustness properties. The curves resemble those presented in Fig. 4.8 and the reader is referred to section 4.4.6.5 for the discussion around their significance.

The peak values of sensitivity and complementary sensitivity function are listed in Table 4.32 and the resulting values fall within the target values of M_T <1.25 and M_S <2.

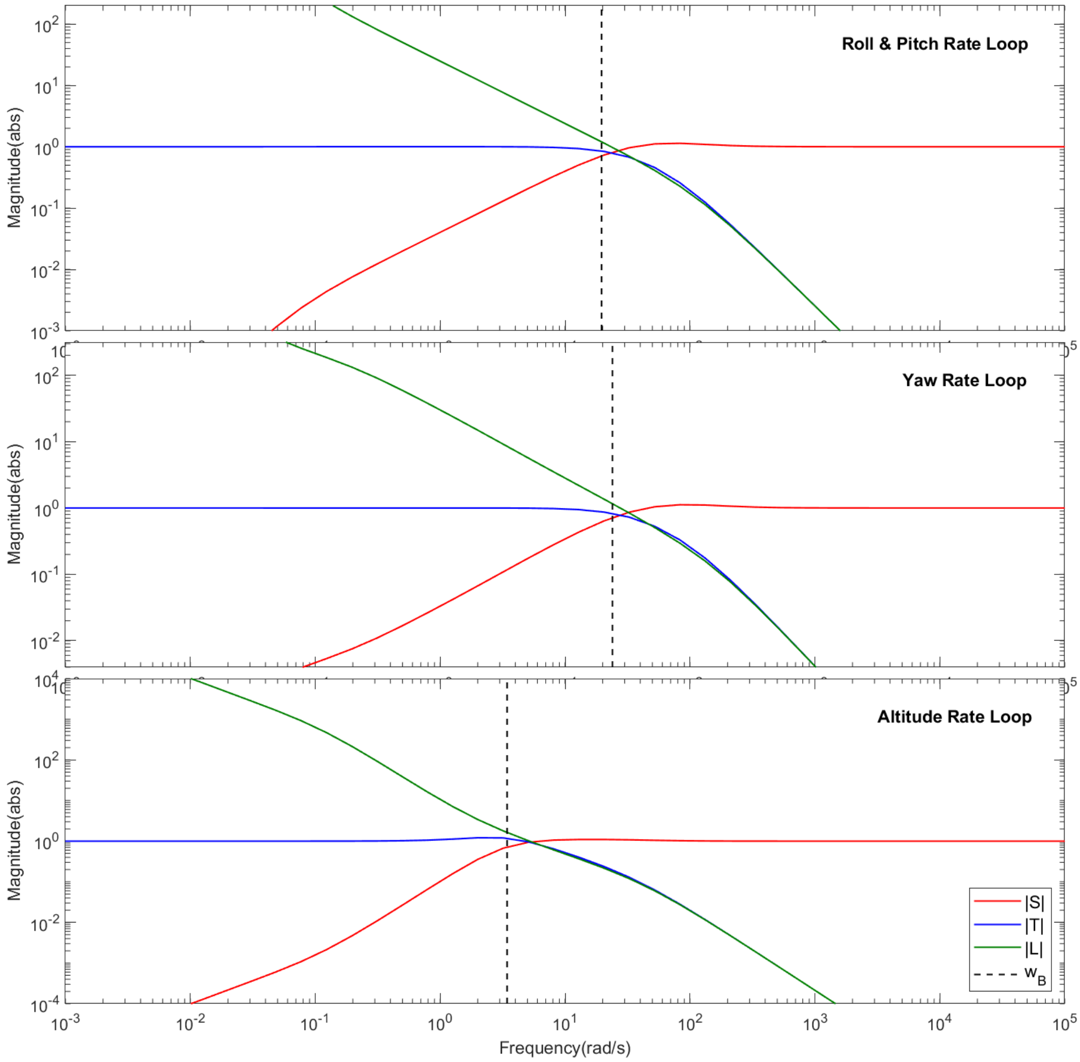


Figure 4.37: The μ-controller based system: Bode magnitude plots of $|S|$, $|T|$ and $|L|$

Table 4.32: Robust Stability Analysis

	Roll & Pitch Rate Loop	Yaw Rate Loop	Altitude Rate Loop
M_T	1.138	1.119	1.213
M_S	1.003	1.000	1.098

Having obtained satisfactory closed loop characteristics in the frequency domain, the time domain responses are analysed in the next subsection.

4.8.4.6 Time Domain Responses

The time responses are presented in two parts. In the first part the nominal non-linear plant responses and motor inputs toward a pulse reference signal is presented. In the second part the plant containing model uncertainty is subjected to a complex manoeuvre and the effectiveness of the controller with respect to the robustness of the plant is examined.

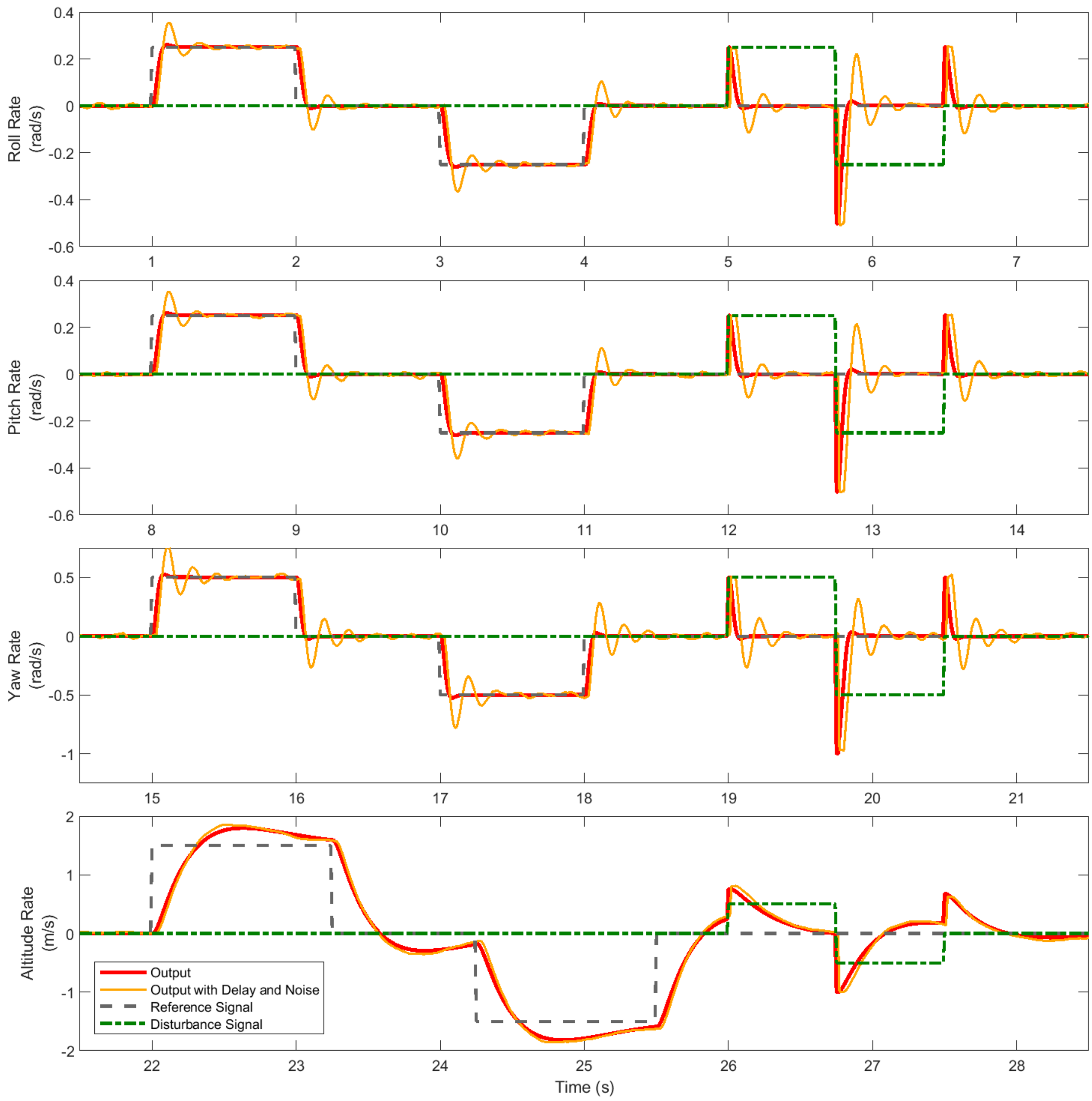


Figure 4.38: Step response of the nominal μ-controller based plant

The nonlinear nominal plant response is presented in Fig. 4.38. Two sets of simulations are conducted. In the first simulation the plant is subjected to a reference signal in a delay and noise free environment.

In the second simulation the plant is subjected to a delay of two sample times and a zero mean noise signal with standard deviation that measures to about 15% of the maximum value of the reference input. In the latter simulation, the signal to noise ratio is calculated to be 17.718 dB, 17.841 dB, 17.886 dB and 10.642 dB in the roll, pitch, yaw and altitude rate loops respectively.

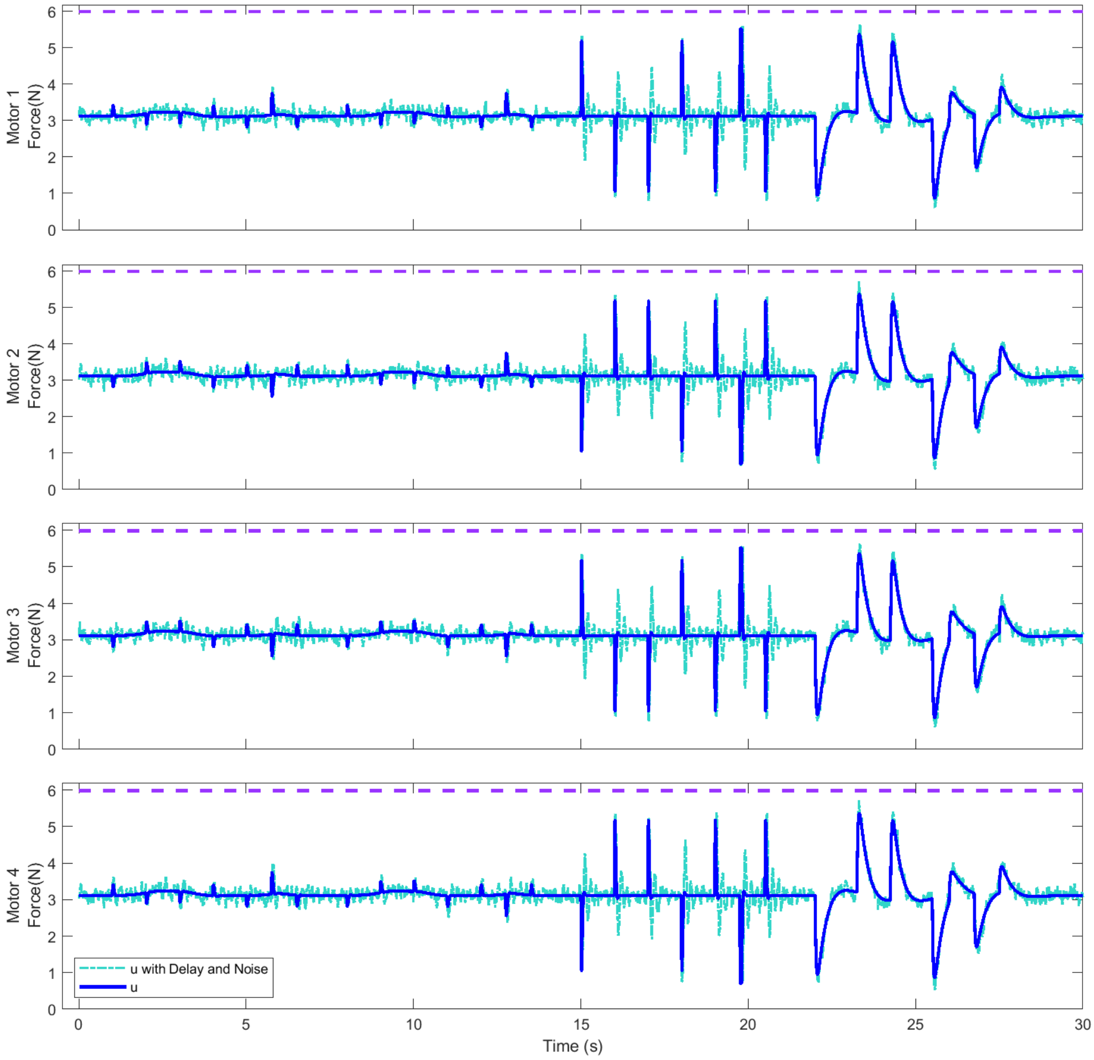


Figure 4.39: Forces exerted by individual motors in response to reference signal

The forces generated by each motor are presented in Fig. 4.39. Examining both figures and from subsequent simulations the following points can be noted.

1. Acceptable over and undershoots, rise times and settling times (see Table 4.33) are observed in the delay and noise free responses.
2. Delay induces oscillations in the responses. A delay of two sample times introduces substantial oscillations in the roll, pitch and yaw rate responses, which was expected from the delay margins the controller provided. The altitude rate shows relatively higher immunity towards delays.
3. Acceptable noise immunity can be observed towards noise injected at the output of the system.

4. Inputs reach saturation limits although the effects on performance are negligible.

The step response characteristics of the nonlinear nominal plant are presented next. The peak values (3rd row in Table 4.33) should be seen in the context of the maximum values of the reference signal, being 0.25 rad/s for roll and pitch rate, 0.5 rad/s for yaw rate and 1.5 m/s for altitude rate.

Table 4.33: Time and frequency domain specifications after controller tuning

Response Characteristics	Roll & Pitch Rate	Yaw Rate	Altitude Rate
Rise Time (s)	0.047	0.037	0.225
Settling Time (s)	0.142	0.116	1.446
Peak	0.260	0.524	1.798
Peak Time (s)	0.100	0.080	0.630
% Overshoot	4.160	4.871	19.855
% Undershoot	0.000	0.000	0.002

From the results, it can be seen the acceptable response characteristics have been obtained. In the next part the responses of the plant with model uncertainty is examined.

In this second part of the section the plant is subjected to a complex manoeuvre (identical to those in section 4.4.6.10). The % parameter uncertainty incorporated into the nonlinear plant is the same used to develop the complementary sensitivity weight w_I (see Table 4.27).The response of the uncertain plant to the reference input is presented in Fig. 4.40. The forces produced by the motors in response towards the reference inputs are presented in 4.41.

From examining both figures the following points can be noted.

- Slight overshoots can be observed in roll and pitch rate responses for the extreme plants (coloured in yellow). Overshoots can also be observed in the yaw and altitude rate loops, for extreme plants as well as those plant models in between the nominal and extreme plants.

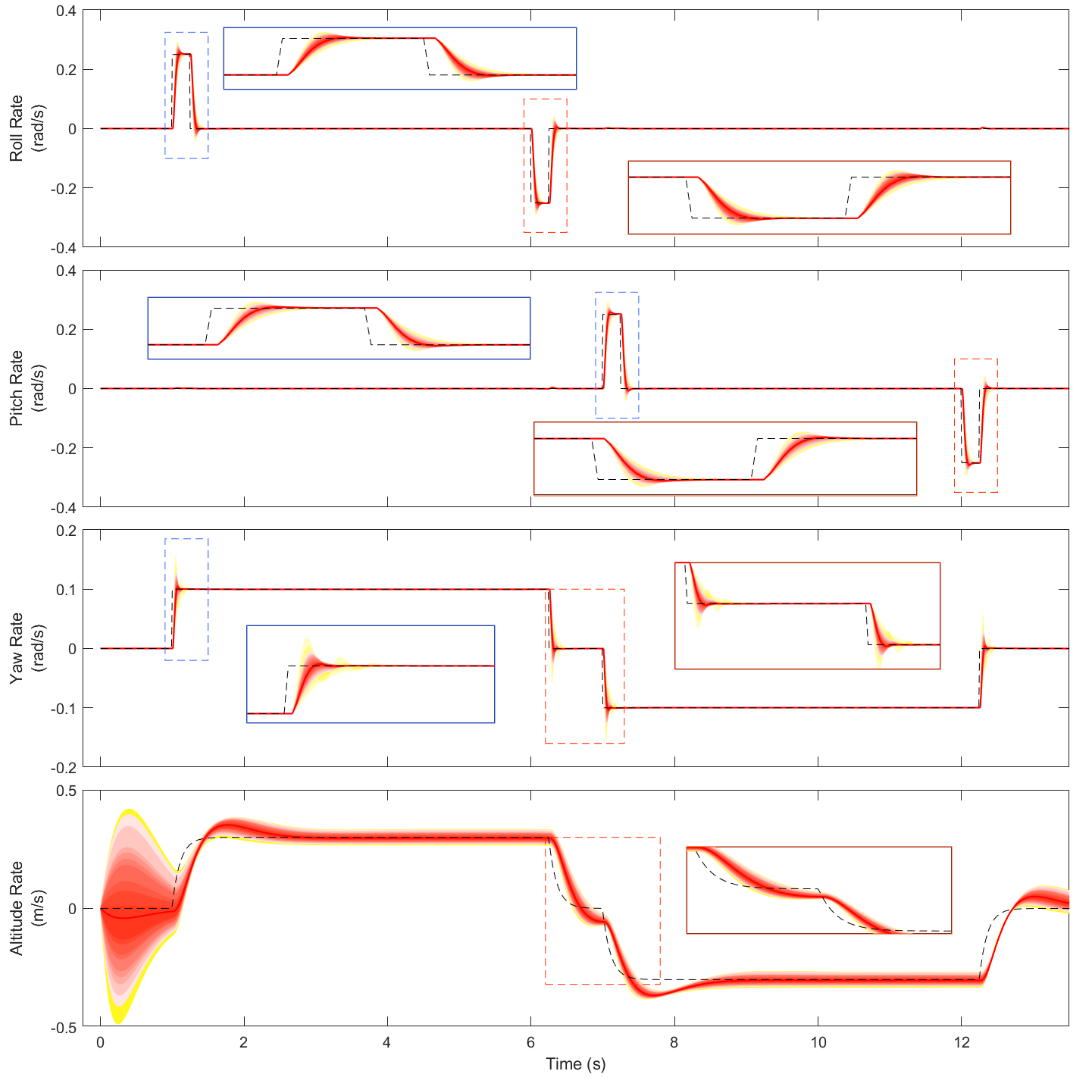


Figure 4.40: The response of the μ-controller based plant with model uncertainty to the complex manoeuvre reference command.

- Oscillatory responses can be observed in the extreme plants in the yaw rate loop. Oscillations are absent in the pitch, roll and altitude rate responses.

- The input forces generated fall below their saturation limits. Oscillations are absent in the motor reactions to the reference commands in the roll and yaw rate loops. Slight oscillations can be observed in the forces generated in the pitch and altitude rate loops for the extreme plants (coloured in light blue)

This brings us to the end of the examining time domain responses of the non-linear uncertain plant. Both stability and acceptable performances have been observed in the simulations. In the next section the robustness aspects in the frequency domain will be examined.

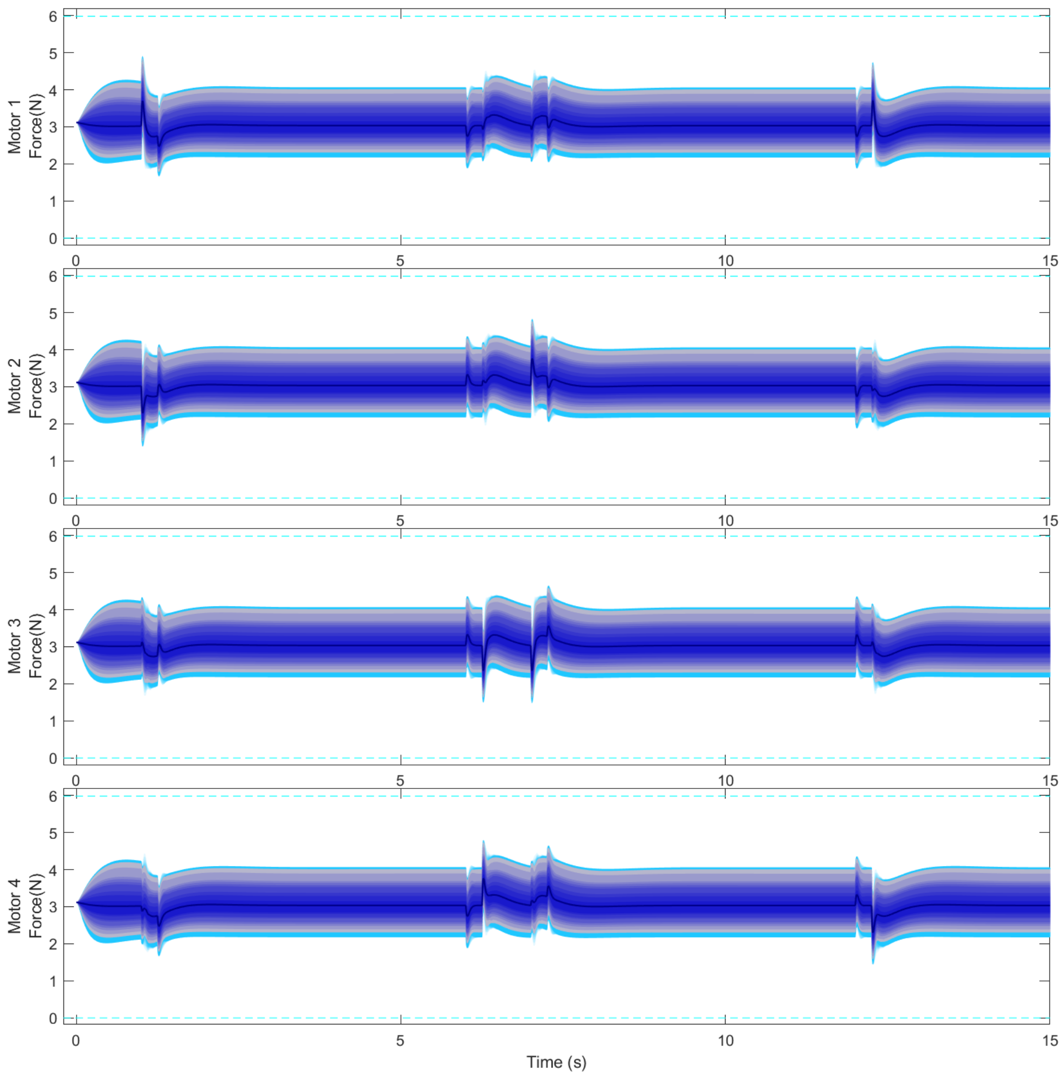


Figure 4.41: The μ-controller based plant rotor responses to the complex manoeuvre reference signal.

4.8.4.7 Robustness Analysis in the Frequency Domain

From the values of μ displayed in Table 4.28, the limits of uncertainty that each loop can withstand are calculated and the values listed in Table 4.34. It can be seen that the roll and pitch rate loop can tolerate 2.451 times the current model uncertainty at a frequency of 0.749 rad/s and still exhibit robust performance. The yaw rate can tolerate 1.020 times and altitude rate can tolerate 1.862 times the model uncertainty at frequencies of 2.574 rad/s and 2.591 rad/s respectively and still exhibit robust performance.

The table points toward the superiority of μ controllers compared to MSO controllers in terms of its ability to guarantee robust performance during the development phase of the controller. The controller designs have been acceptable in terms of its ability to guarantee robust stability and performance w.r.t to the model uncertainty considered.

Table 4.34: Robust Performance Analysis

Loops	Roll & Pitch Rate Loop	Yaw Rate Loop	Altitude Rate Loop
Uncertainty upper bound	2.451	1.020	1.862
Critical frequency (rad/s)	0.749	2.574	2.591

This brings us to the end of the design procedure of μ controllers. In the final section of this chapter a comparison of various controllers and their performances are presented.

4.9 Performance Comparisons

In this section, comparisons between the different controllers developed in the previous sections are performed. The section is subdivided into five parts. In the first part, the time domain response characteristics of the closed loop system subjected to step responses are examined and performances compared. The performance of the nominal plant as well as the plant with model uncertainty is assessed and compared.

In the second part, the classical margins of each of the robust control systems are examined. Like in the first part, the margins of the nominal plant as well as those with model uncertainty are examined. In the third part of the section, a comparison is made on 1 DOF controllers based on value of the performance characteristic μ obtained over frequency. MSO, LSDP 1 DOF and the μ-controller are compared in this section.

In the fourth part of the section, those controllers that exhibited robust performance characteristics, as seen from the analysis performed during the controller development phase, are analysed using the gap-metric stability criterion and comparisons are made. This is followed by the final part where certain conclusions are drawn from the observations made.

4.9.1 Time domain step response characteristics

As observed from figures 4.42 to 4.45 three aspects of the step responses are examined in the section; these being the percentage overshoot, rise time and the settling time. To obtain the step responses (both for the nominal plant and for those with model uncertainty) the plant is subjected to the same reference signal presented in Fig. 4.9. For incorporating model uncertainty, parameters in the range described in Table 3.2 are selected. In this first part of the section all the five developed controllers, (i.e. the four robust controllers alongside the PID controller) are compared based on these step response characteristics of their respective closed loop systems. Before we examine the results, the method of presentation is briefly examined in the following paragraph.

Violin plots are used, to present the results. A violin plot presents the probability density of the data points at different values after it is (approximately) smoothed by a kernel density estimator (see [196]). The sample points are also plotted in the figure. Unlike box plots which presents the user with the four quartiles of the data alongside the outliers, and which remain unaffected by the distribution of the samples, the violin plots gives a visual representation of the distribution of the data, and more suitable for unusual distributions.

In the data presented, the violin plots are shaded in light blue. The blue dots (•), represents the response of the systems with model uncertainty, the median value is represented using the red cross (+) and the value of the nominal plant is represented by the green circle (●). The design constraints (see section 3.5) are represented by grey patched area. The responses from a total of 512 systems have been taken into account to generate each violin plot. Of this 256 systems are those from the vertices of the hypercube (see section 4.12) and the remaining 256 systems are those which are uniformly distributed in the hyperspace around the nominal plant.

Each set of violin plots can be inspected for the following four aspects.

1. If the median value of the response characteristic lies outside the patched grey area in the plot, where the patched grey area refers to soft constraints of the design specifications presented in 3.5.

2. If the violin plot is completely outside of the patched grey area.

3. The nature of the spread of points and the resulting shape of the probability density distribution.

4. The nature of each individual plot in comparison to the rest of the four violin plots.

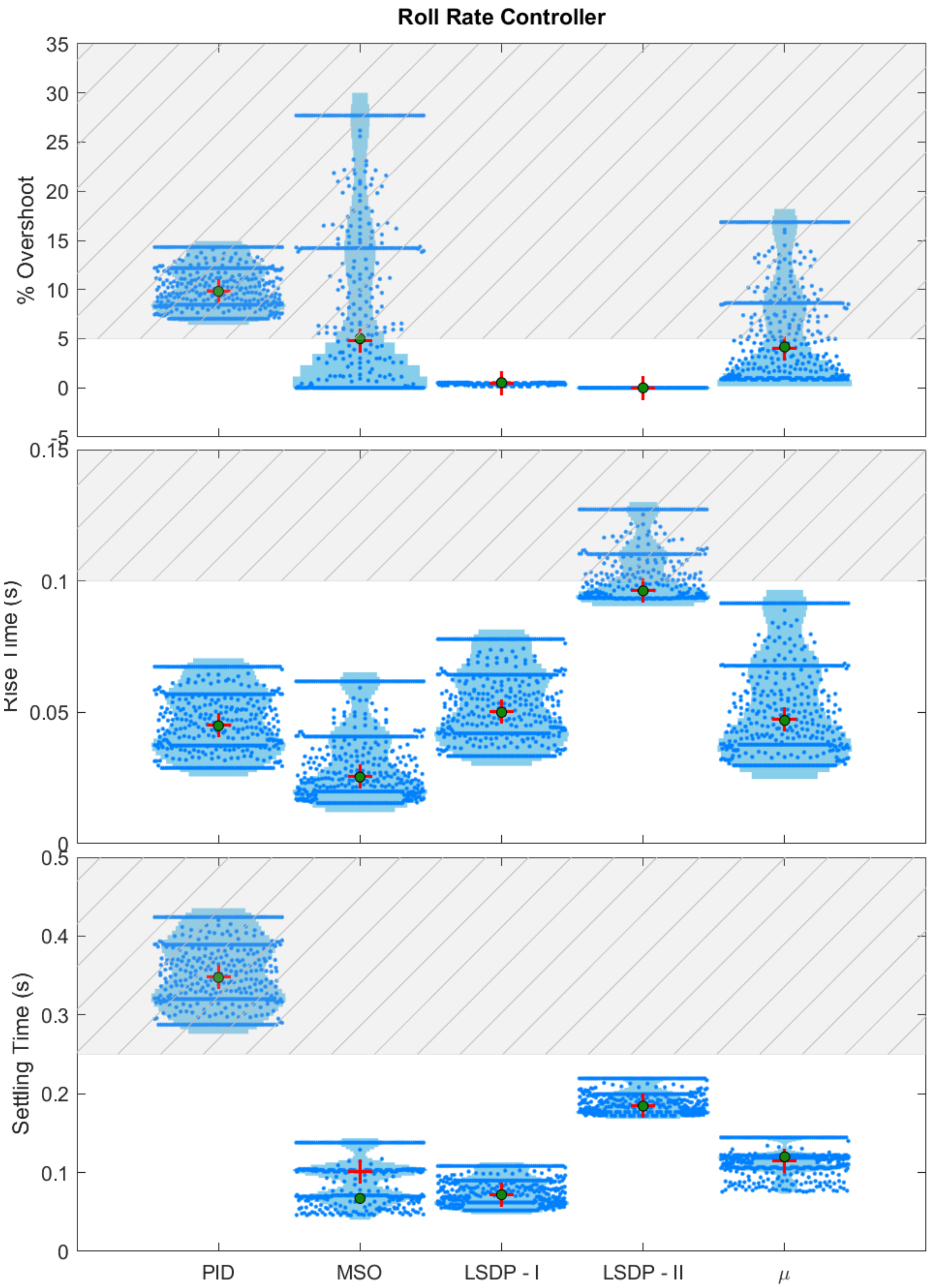


Figure 4.42: Roll rate controller- time domain response comparison using violin plots ([196]): The markers displayed have the following meanings - • : The response of the systems with model uncertainty, + : Median value of the respective response characteristic and ● : The value of the response characteristic of the nominal plant.

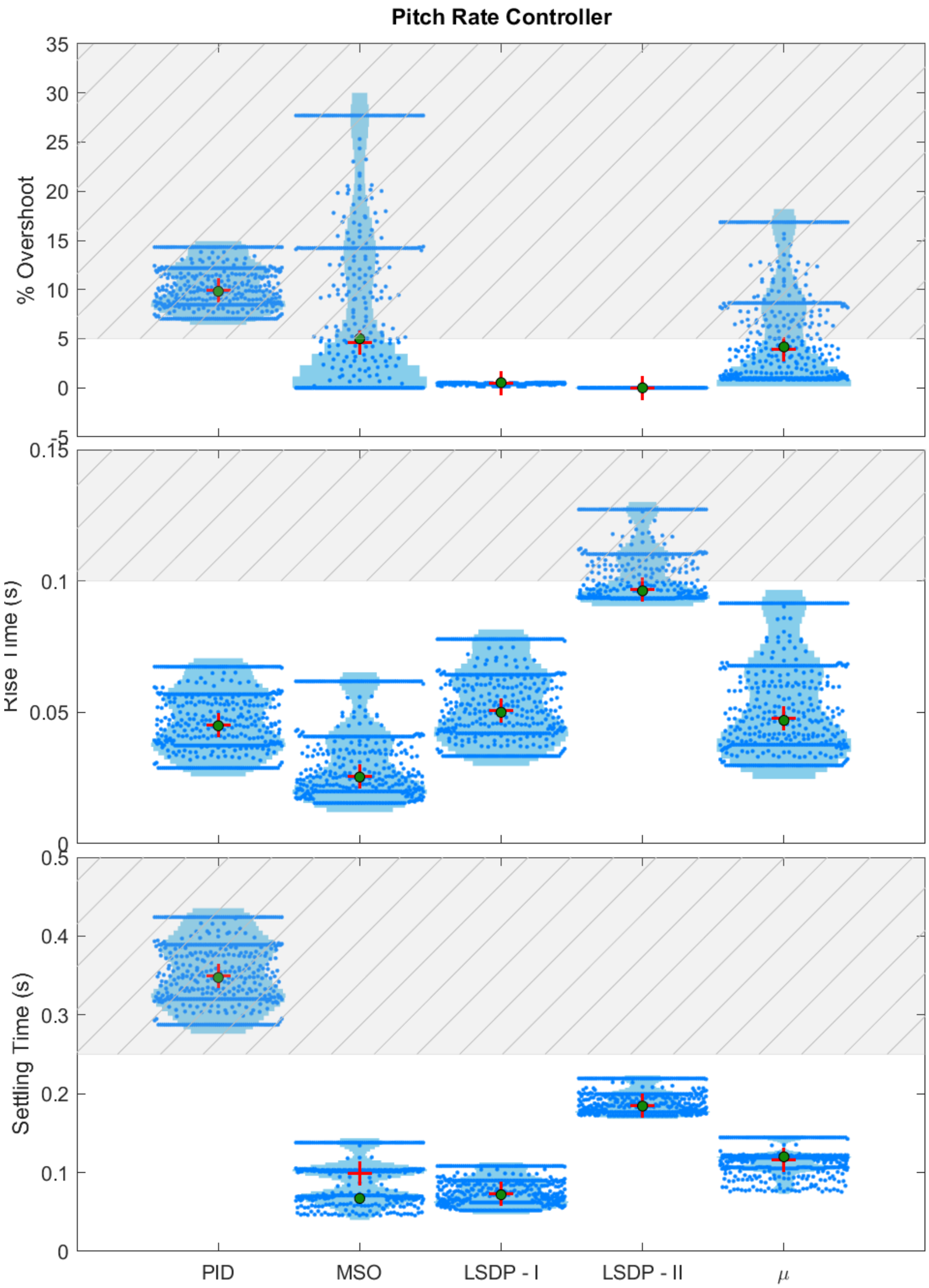


Figure 4.43: Pitch rate controller- time domain response comparison: The markers displayed have the following meanings - • : The response of the systems with model uncertainty, + : Median value of the respective response characteristic and ● : The value of the response characteristic of the nominal plant.

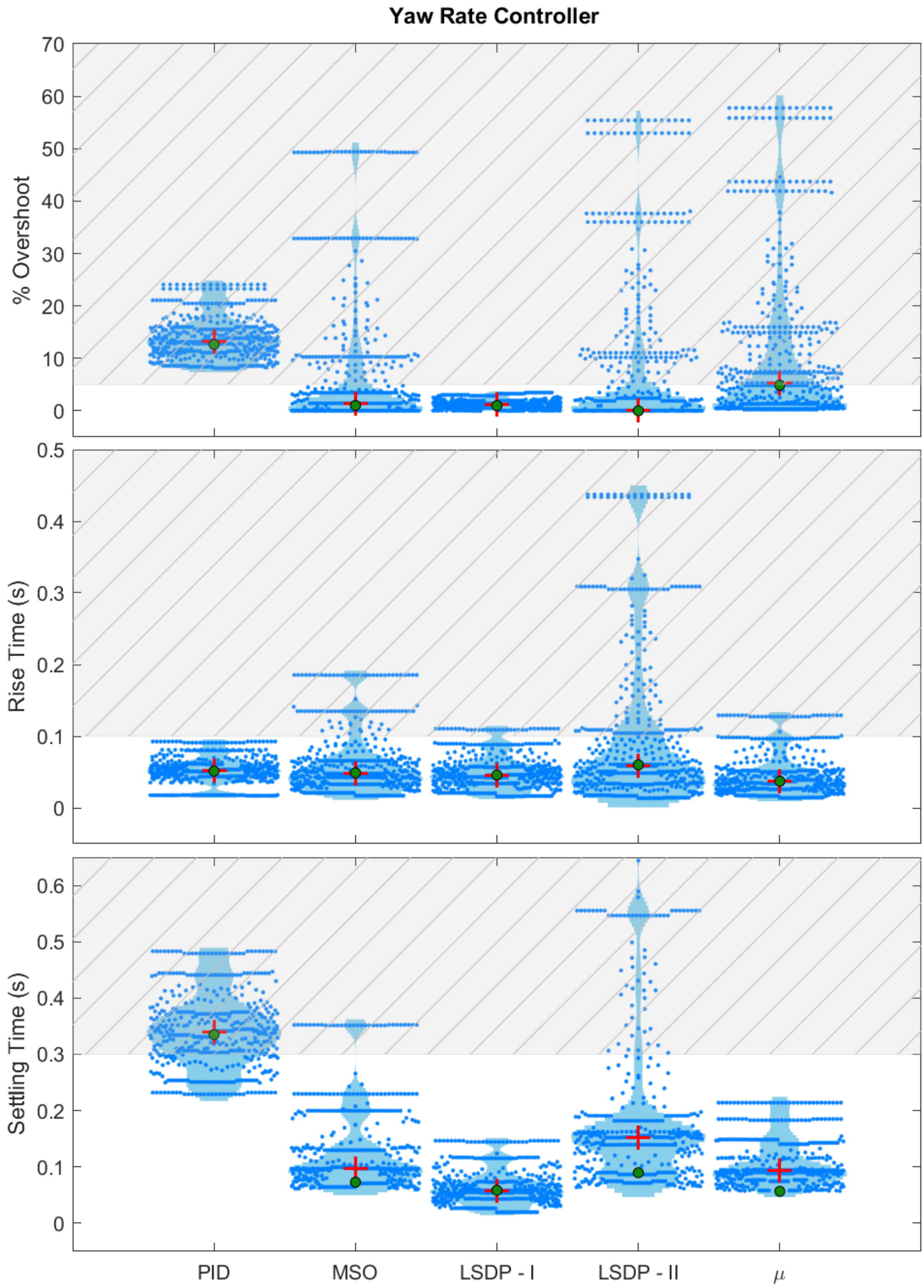


Figure 4.44: Yaw rate controller- Time domain response comparison: The markers displayed have the following meanings - • : The response of the systems with model uncertainty, + : Median value of the respective response characteristic and ● : The value of the response characteristic of the nominal plant.

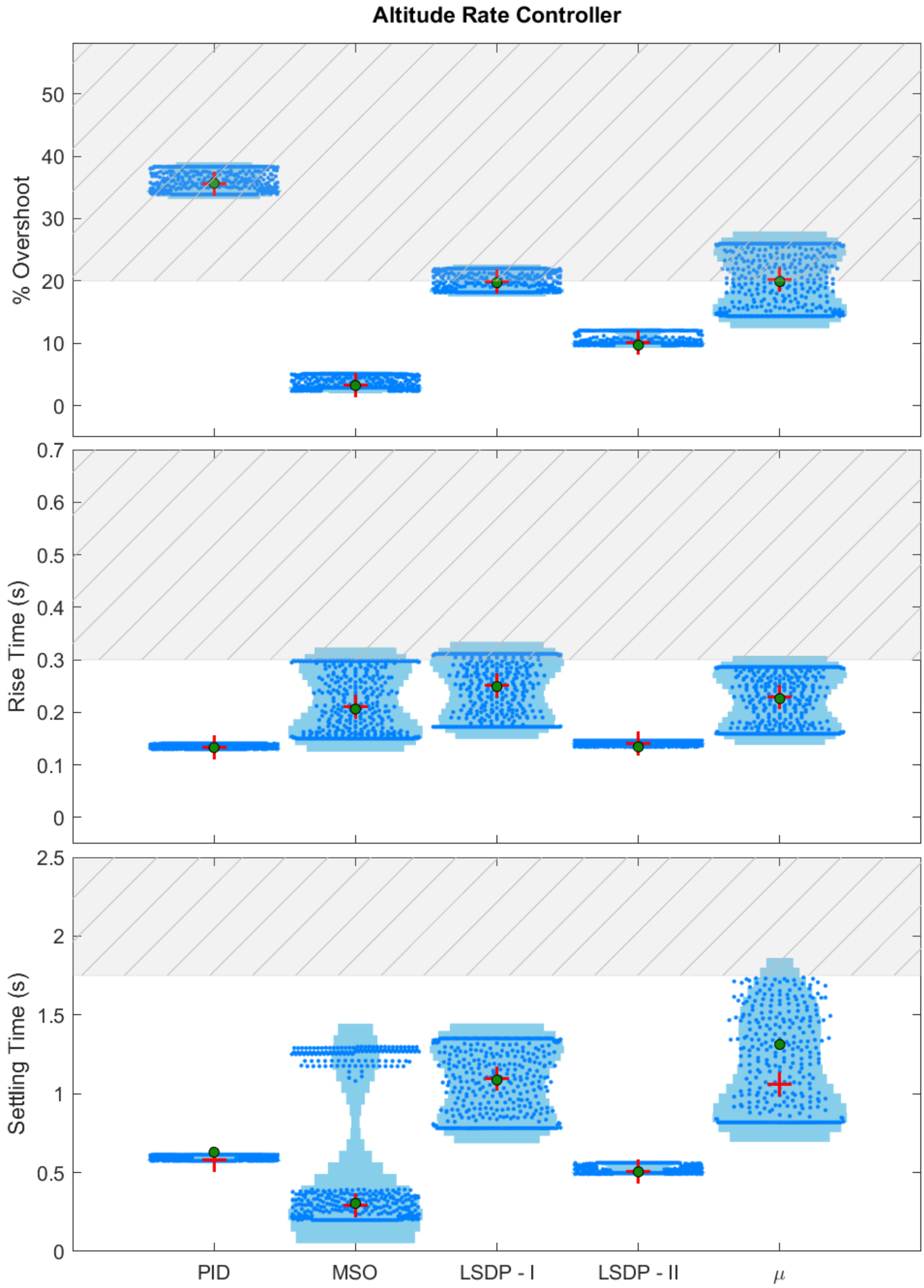


Figure 4.45: Altitude rate controller- Time domain response comparison: The markers displayed have the following meanings - • : The response of the systems with model uncertainty, + : Median value of the respective response characteristic and ● : The value of the response characteristic of the nominal plant.

The following broad conclusions can be drawn from figures 4.42 to 4.45.

1. The PID controller designed for highest robustness and acceptable rise times, while it maintains plant stability for all uncertain plants (thanks to tuning for robustness), designs specifications were not fully met. Acceptable performance is registered w.r.t rise times for all of the 512 models, but with regards to overshoot and settling times the required standards haven't been met.

2. The MSO controller in general registered a performance which is close to acceptable in many regards. Although the extreme plants in cases produced a response that fell in the grey hatched unacceptable zone, to a major extent 50% of plants with model uncertainty exhibited robust performance.

3. The LSDP 1 DOF controller registers acceptable performance. Robust performance for all the 512 models are achieved for roll and pitch rate loops. The yaw and altitude rate loops, except for a portion of the models at the vertices, satisfied the design constraints.

4. The LSDP 2DOF controller achieves acceptable performance for more than 50% of all models in the four loops. A portion of the extreme plat models are characterized by slow rise times, in roll, pitch and yaw rate loops, and for yaw rate loop performance decline is observed for a set of extreme plants in the % overshoots and settling times.

5. The μ-controller registers robust performance w.r.t to some aspects of the system. A portion of extreme plants experience unacceptable overshoots in roll, pitch, yaw and altitude rate loops and slower rise times in the yaw rate loops

These observations based on whether or not, the performance fell in the acceptable zone are further presented in Table 4.35.

Table 4.35: Time domain Performance Comparison

Controllers	PID	MSO	LSDP 1DOF	LSDP 2 DOF	μ
% Overshoot	○ ○ ○ ○	◒ ◒ ◒ ●	● ● ● ◒	● ● ◒ ●	◒ ◒ ◒ ◒
Rise Time	● ● ● ●	● ● ◒ ◒	● ● ◒ ◒	◒ ◒ ◒ ●	● ● ◒ ◒
Settling Time	○ ○ ◒ ●	● ● ◒ ●	● ● ● ●	● ● ◒ ◒	● ● ● ●

Here full circles indicate the controller achieves acceptable performance for all the 512 plant models. The amount of fullness of each circle, corresponds to the number of plant models that falls in the region of acceptable performance. The four circles at each entry corresponds to roll and pitch rate loop in the first row from the left and to yaw and altitude rate loop in the second row from the left.

A quick look at the table suggests that the single degree of freedom LSDP controller seems to be the most effective in delivering the required performance for the models with model uncertainty. While MSO, LSDP 2 DOF and μ remains reasonably effective, tuning the PID for robustness haven't proved to be fully effective as anticipated. For the case of the PID controller, tuning the system for the maximum robustness leaves the control engineer with very few options to simultaneously meet all design specifications. This lack of tuning option is evident from the performance registered with the uncertain plant models as observed above.

The robust controllers by design strives to push down the peak magnitudes of sensitivity, complementary sensitivity and the closed loop function. This in the Nyquist plot translates towards healthy gain and phase margins and further translating into acceptable time domain performances and robustness. While the guaranteed frequency domain performance might not translate into guaranteed time domain performance in terms of the user defined requirements (as we have witnessed the cases of some extreme plant time responses lying in the hatched area), the controller skews the performance of uncertain models in the hypercube, favorably towards the required designs standards.

In the next part the classical margins as well as the bandwidth of the closed loop will be examined for those controllers which were effective namely the robust controllers (PID controller performances are excluded).

4.9.2 Classical Margins

The effect of model uncertainty on the classical margins and bandwidth of the system and the effectiveness of the robust controllers in mitigating it, is examined in this part.

The gain phase and delay margin as well as the bandwidth of the closed loop system for all the 512 models with uncertainty are calculated and the values are plotted. Violin plots are employed to present the data in Fig. 4.46 to Fig. 4.49.

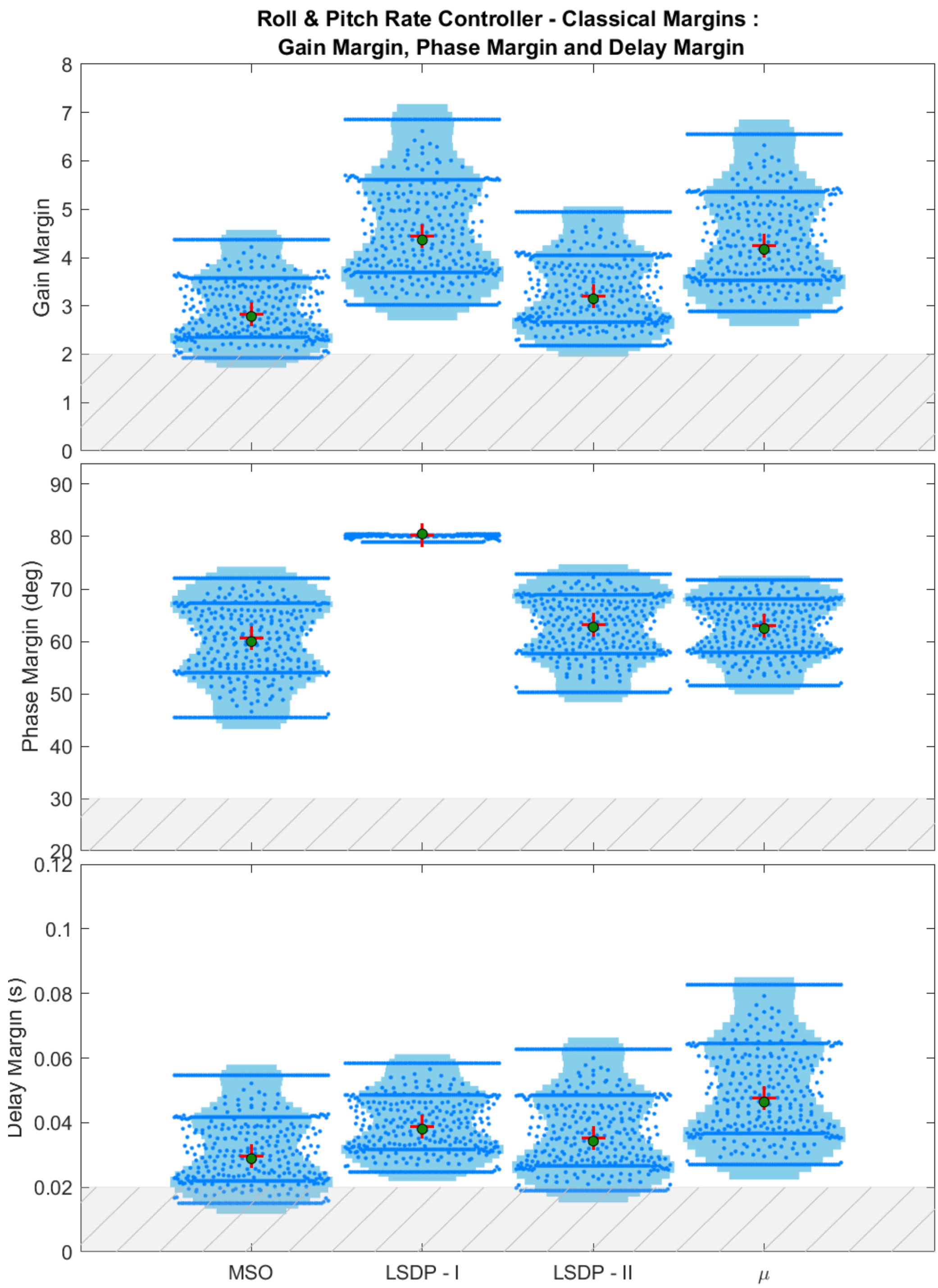


Figure 4.46: Roll & Pitch rate controller- Classical Margins Gain, Phase and Delay Margin: The markers displayed have the following meanings - • : The response of the systems with model uncertainty, + : Median value of the respective response characteristic and ● : The value of the response characteristic of the nominal plant.

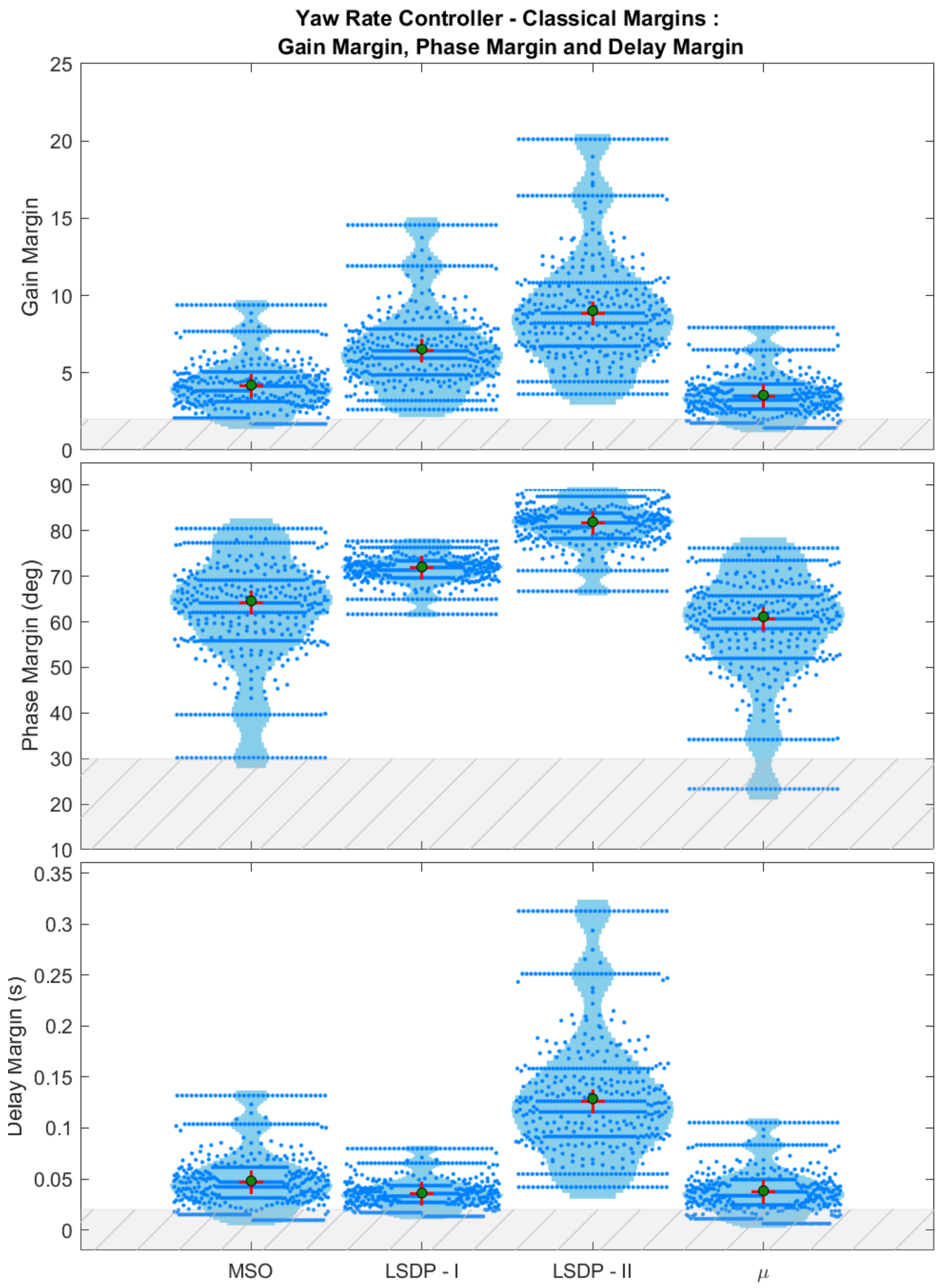


Figure 4.47: Yaw rate controller- Classical Margins Gain, Phase and Delay Margin: The markers displayed have the following meanings - • : The response of the systems with model uncertainty, + : Median value of the respective response characteristic and ● : The value of the response characteristic of the nominal plant.

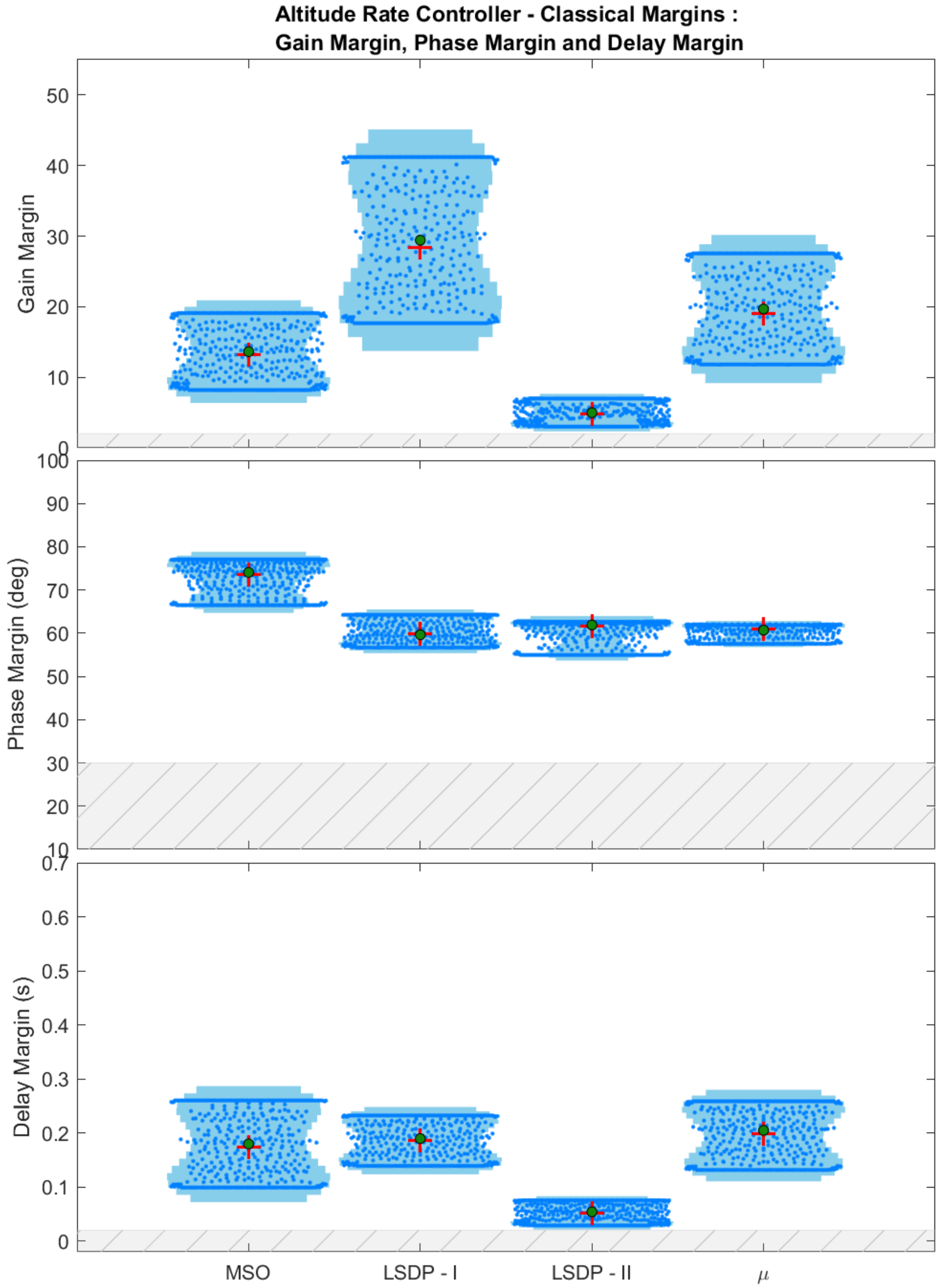


Figure 4.48: Altitude rate controller- Classical Margins Gain, Phase and Delay Margin: The markers displayed have the following meanings - • : The response of the systems with model uncertainty, + : Median value of the respective response characteristic and ● : The value of the response characteristic of the nominal plant.

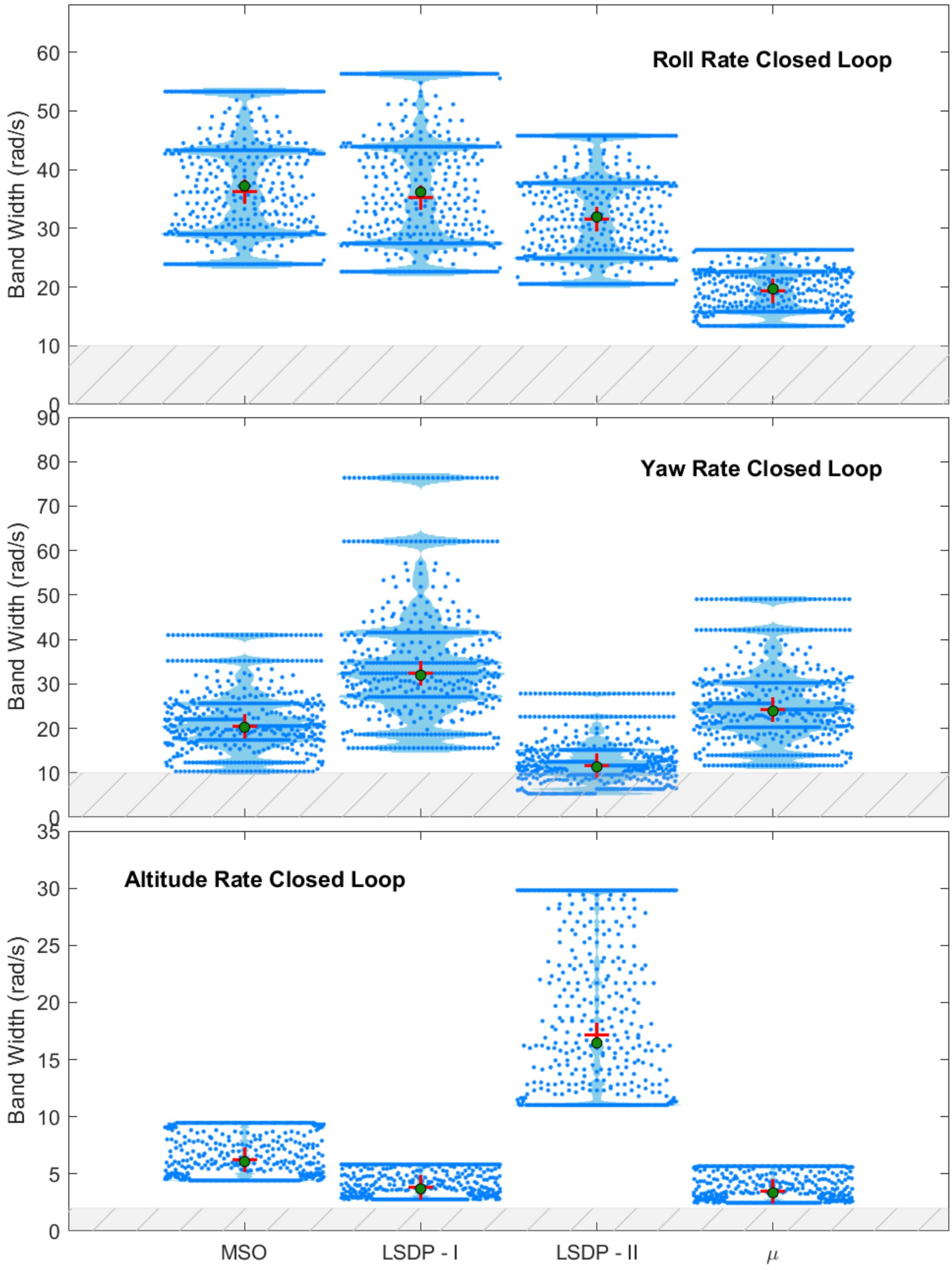


Figure 4.49: Closed loop bandwidth: The markers displayed have the following meanings - • : The response of the systems with model uncertainty, + : Median value of the respective response characteristic and ● : The value of the response characteristic of the nominal plant.

From examining the plots in Fig. 4.46 to Fig. 4.49 the following conclusions are drawn:

1. The MSO, LSDP 2DOF and the μ controller provides acceptable gain (>2), phase (>30°) and delay margins (2 time samples) for majority of the uncertain plant models. Similarly the three controllers ensures the bandwidth falls within acceptable margins (see section 3.5). A small sectin of the extreme plants however fails to provide acceptable margins and bandwidths.

2. The LSDP 1DOF controllers provide by far most acceptable margins as well as reasonable closed loop bandwidth for the uncertain plant models. This performance agrees with the conclusions drawn from the previous part.

In further simulations conducted to examine M_T and M_S (although not presented here due to the obvious similarities to those of the classical margin plots), acceptable peak values (M_T <1.25 and M_S <2) have been observed in all but one of the instances. The only exception was with the values of M_T in the yaw rate loop where extreme plants tended to have peak values of the complementary sensitivity function to fall above the acceptable levels. For further details regarding the relations between M_T and M_S to the classical gain and phase margins refer page 36, Chapter 2 in [118].

In the violin plots the characteristic horizontal lines arise from the performances of the plants from the edges of the parameter hypercube (i.e. the extreme plants) and those performance points that are scattered arise from within the hypercube. The light blue distribution shades sometimes overshoot the actual data points and these overshoots can be ignored.

In the next section we examine single degree of freedom controllers and make comparisons in the frequency domain using the μ criteria.

4.9.3 1 DOF controller comparison

In this part the μ-synthesis is utilised to compare the single degree of freedom robust controllers. The approach taken here, is to use the sensitivity and complementary sensitivity weights designed during the development of the μ controller (see section 4.8.4). The system presented in Fig. 4.35 with an input multiplicative uncertainty and performance weights is considered for the analysis.

The values of μ to determine nominal performance, robust stability and robust performance as listed in Table 4.26 have been calculated and plotted as a function of frequency. The plots are calculated for systems controlled by MSO, LSDP 1DOF and μ-controllers.

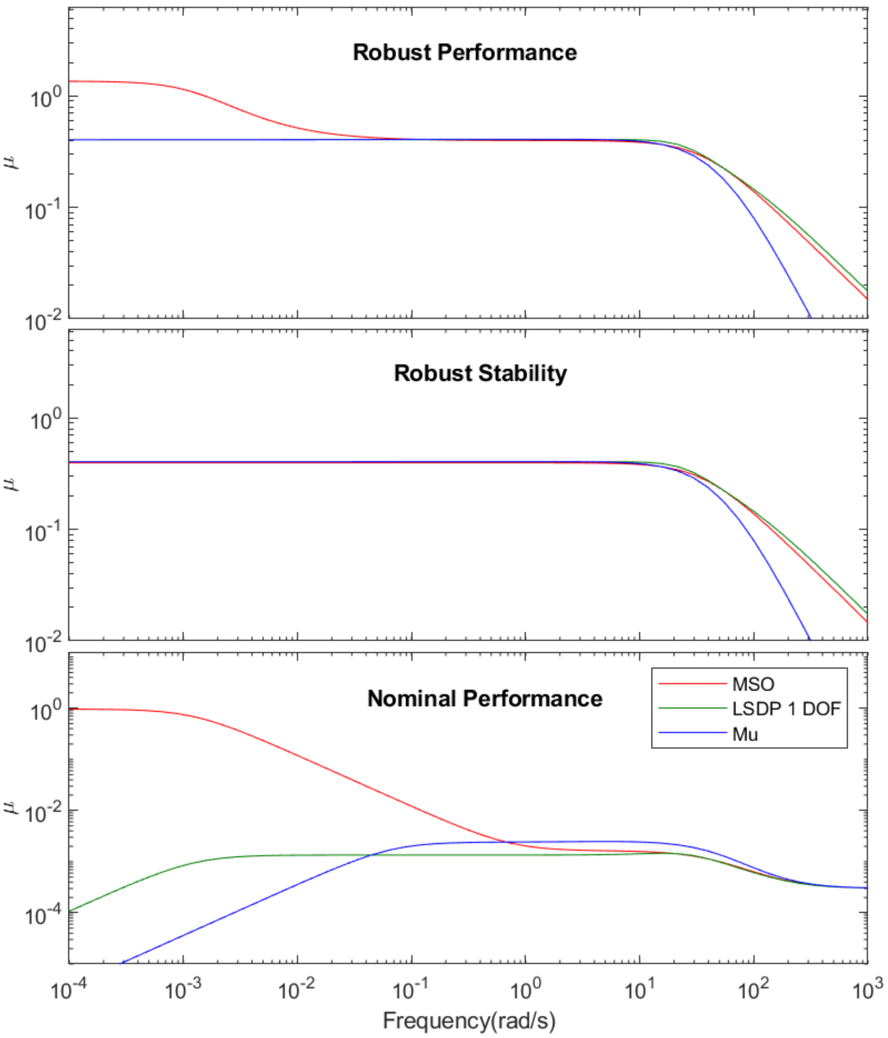


Figure 4.50: Roll and pitch rate controller: μ plotted as a function of frequency

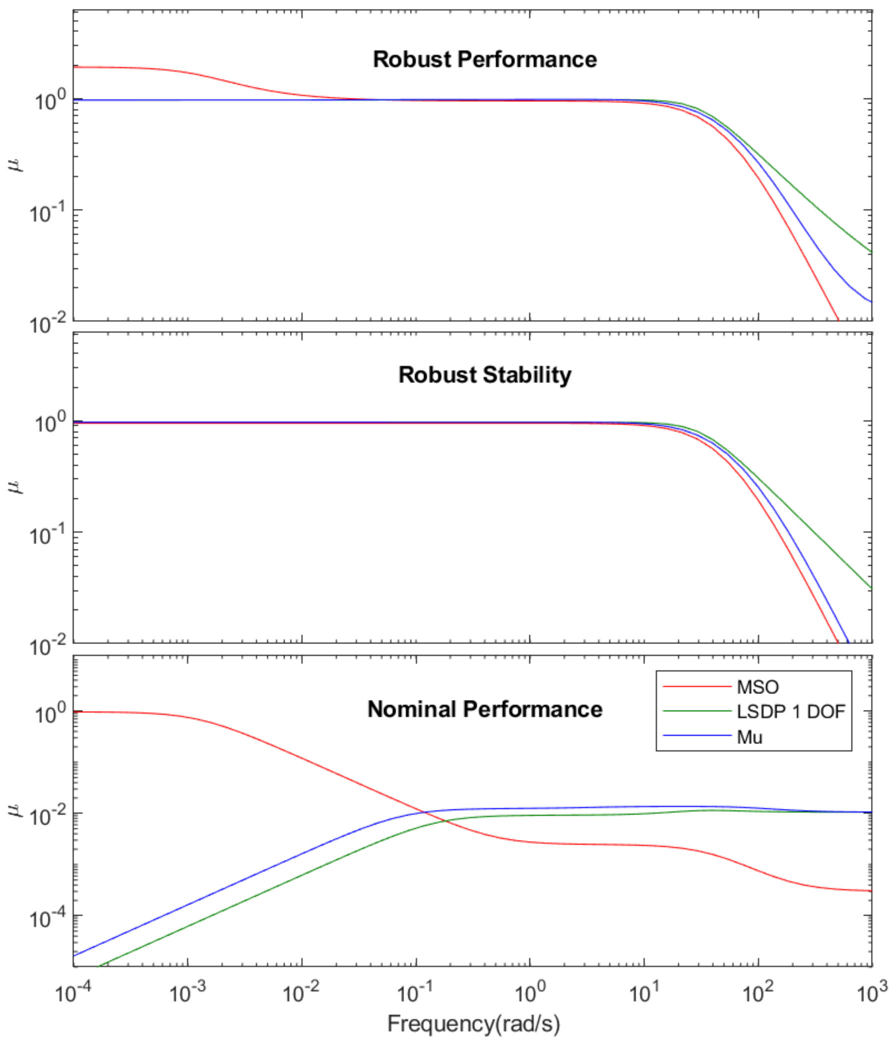


Figure 4.51: Yaw rate controller: μ plotted as a function of frequency

From inspecting the plots presented in Fig. 4.50 to Fig. 4.52 the following points can be noted.

- For the given performance weight and The MSO controller fails to limit the value of μ, to below unity w.r.t to the robust performance criteria, below a frequency of approximately 0.002 rad/s (0.1 Hz) for roll and pitch rate controller and below a frequency of 0.02 rad/s (1.1 Hz) for yaw rate controller.

- As far Robust Stability and Nominal Performance is concerned, the 1 DOF controllers perform well. All of the controllers are able to limit the value of μ in all the four loops.

- A closer inspection of the plots shows that LSDP 1DOF controller consistently limits the value of μ below that of the μ controller at all frequencies except for those in a narrow band around the bandwidth frequencies, where the μ-controller very slightly outperforms the LSDP 1DOF controller.

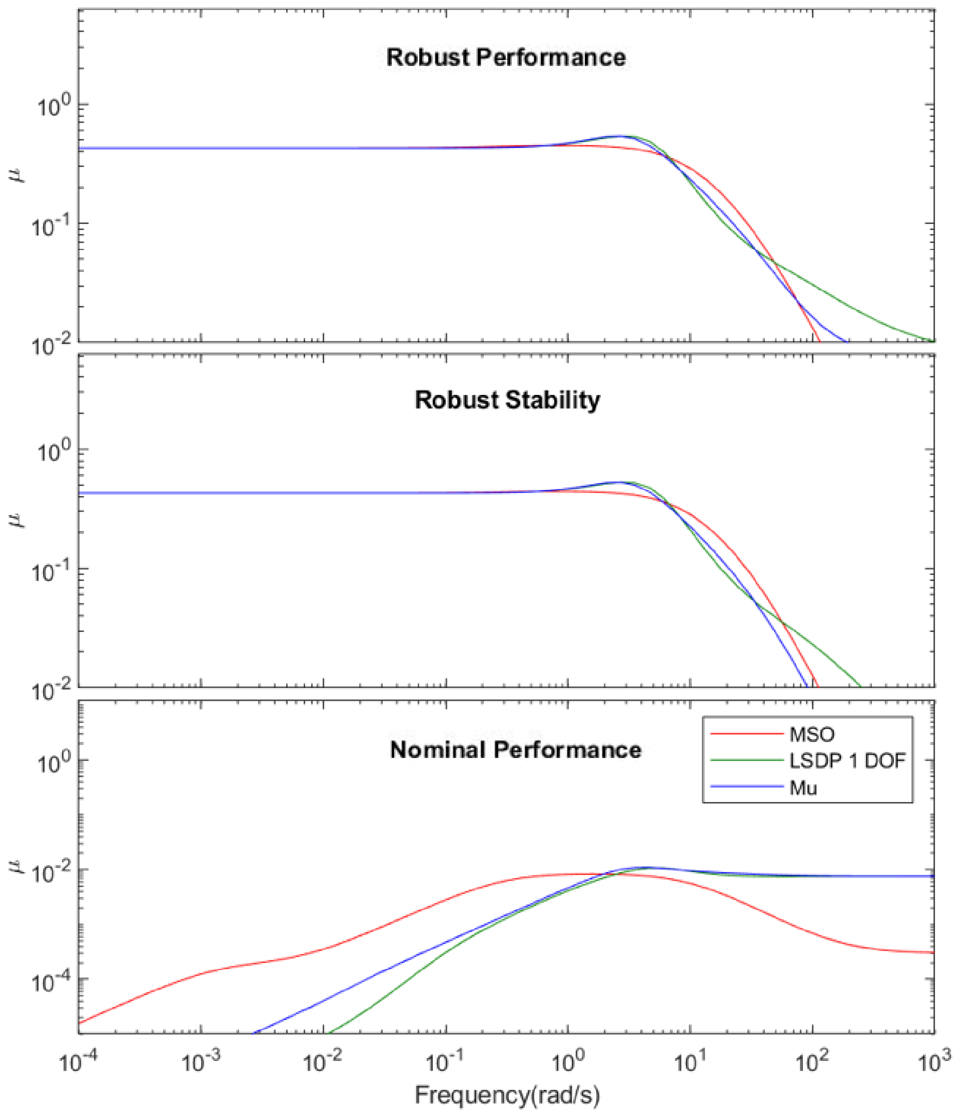


Figure 4.52: Altitude rate controller: μ plotted as a function of frequency

The significance of limiting the value of μ below unity can be explained in terms of dynamics in the Nyquist plot. As seen in the figure the performance weight or the sensitivity weight w_P can be visualized as the radius of sphere in the Nyquist plot centered at -1 on the imaginary axis. The term $L(j\omega)$ in the figure represents the loop transfer function of the systems. The term $|w_I(j\omega)L(j\omega)|$ can be visualized as the radius of a sphere centered on the Nyquist plot.

In the figure, while w_P represents the closeness to instability, w_I represents model uncertainty. As the value of μ falls above 1 these two spheres intersect each other resulting in a deterioration of the system performance, although stability is still maintained.

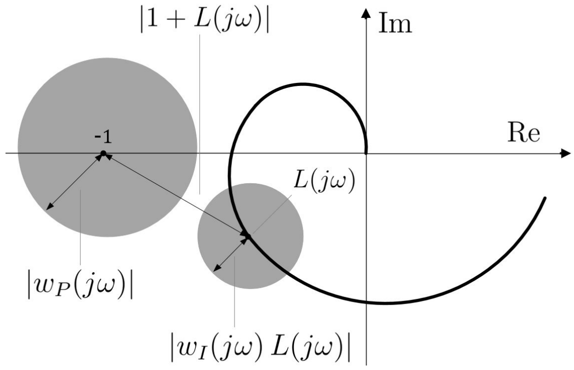


Figure 4.53: Nyquist plot dynamics

Hence lower the value of μ better the performance. According to this criteria in overall LSDP 1DOF controller ranks the best followed by μ and MSO controller.

In the next part those controllers that have been identified as the ones which can provide acceptable robust performance are compared using the the gap-metric criterion.

4.9.4 Comparison using the Gap-Metric Stability Criterion

The gap-metric stability margin as previously defined in section 4.6.2 calculates the distance of the furthest plant from the nominal model that can be stabilized by the controller at hand. In this part, this criterion is used to compare those controllers that have shown promising robust performance properties, namely the LSDP 1DOF, LSDP 2DOF and the μ controller.

Table 4.36 presents the stability margins.

Table 4.36: Robust Performance Analysis - Stability margins

Loops	Roll & Pitch Rate Loop	Yaw Rate Loop	Altitude Rate Loop
LSDP 1 DOF	0.559	0.537	0.590
LSDP 2 DOF	0.412	0.604	0.682
μ	0.137	0.244	0.240

As can be seen the μ-controller offer the lowest stability margins for model uncertainty. LSDP 1DOF and LSDP 2DOF provides good margins, in terms promising superior performance in the face of uncertainty. In the next and final part of this chapter, the summary of the work performed in the chapter is presented.

4.10 Chapter - Summary

The work presented in the chapter is summarized below:

1. The MIMO model of the quadrotor presented in chapter 3 (3) is decoupled after redefining the inputs and converted into SISO models. Numerical linearisation is performed and a state space model is developed.

2. The S/T MSO problem is implemented and **Algorithm #1** (4.4.4) is proposed. SISO controllers are developed using this strategy while ensuring the design specification are met for the nominal plant and that $\mathcal{H}_\infty$ falls below unity. Stability analysis is carried out in the both time and frequency domain.

3. 1 DOF LSDP is implemented and **Algorithm #2** (4.6.1) is presented. SISO controllers are developed alongside design weights and pre-filters, followed by a stability analysis. Co-prime stability margins are calculated and satisfactory performance is observed.

4. 2 DOF LSDP is implemented and **Algorithm #3** (4.7.1) is presented. Design weights alongside the controller weights are developed and a stability analysis is performed and co-prime stability margins calculated. Satisfactory performance is observed.

5. μ-synthesis is implemented and **Algorithm #4** (4.8.2) is presented. Design weights are developed alongside the controllers. A μ value of less than unity is attained for all the controllers. A stability analysis is carried out and satisfactory performance is observed.

6. Performance comparison between the 4 developed controllers are performed. In the time domain analysis, comparison is also performed against the plant controlled by a PID controller. The robust controllers performed markedly better than PID controller for plants containing model uncertainty, with regards to meeting design specifications. In the time domain step response analysis the LSDP based controllers gave the best overall performances.

7. Comparisons are also performed based on the classical stability margins. Here too the LSDP based controllers provide the best performances. A μ synthesis based performance comparison is further performed. It was seen that when it comes to RP, the MSO controller fails to limit the value of μ below unity at

certain frequencies, while in all other cases the controllers successfully limit the values of μ below unity, for the model uncertainty considered.

8. A comparison based on gap-metric stability criterion shows that LSDP based controllers remain better equipped in the presence of model uncertainty than μ controllers.

This brings us to the end of the chapter. In the next chapter more realistic multivariable robust controllers will be developed for the quadrotor, and plant performances will be analyzed.

Chapter 5

Multivariable Robust Controller Designs

In this chapter the SISO robust controller design concepts from chapter 4 are expanded upon to the multivariable domain. The theory and techniques from the the Chapter 4 will be referred to, and augmented as required. The nonlinear model of the quadrotor developed in Chapter 3 will be utilized to test the developed algorithms.

The chapter is composed of 6 sections following an approach similar to Chapter 4. A general introduction towards multivariable robust control theory is presented in the first section. In the subsequent three sections (sections 5.2, 5.3 and 5.4) the MSO, LSDP and μ-controller MIMO controller development using the proposed algorithms will be presented. In these sections the tracking attitude-altitude controller for the quadrotor will be developed and the performances will be examined. In section 5.5, a performance comparison between the developed controllers will be presented which will followed by the final section 5.6, where the ideas presented in the chapter will be summarized.

5.1 Introduction

The main difference between the SISO and the MIMO model of the plant, is the absence of directions in the former and that of its presence in the latter. The term 'direction(s)' of a MIMO model can be defined as the normalized column vector(s) obtained from the singular value decomposition of the MIMO model's frequency response matrix (i.e, the singular vectors). Input directions influence the gain of a system. As the input directions are varied, the gain of the system varies between its maximum and minimum singular values. Alongside the direction and singular values of the MIMO model, the condition number of the system will also be briefly introduced in the section. These characteristics will be evaluated based on the MIMO quadrotor model developed in Chapter 3.

The singular values of a matrix $\mathbf{G}$ can be computed by the following expression:

$$\sigma_k(\mathbf{G}) = \sqrt{\lambda_k(\mathbf{G}^H\,\mathbf{G})} \tag{5.1}$$

where λ_k represents the k^{th} eigenvalue of $(\mathbf{G}^H\,\mathbf{G})$ where $\mathbf{G}^H$ represents the complex conjugate transpose of $\mathbf{G}$. The largest singular value of the matrix (represented by $\bar{\sigma}$) gives the maximum gain of the matrix in any given direction and the smallest singular value (represented by $\underline{\sigma}$) similarly gives the minimum gain in any given input direction. As such, plotting the maximum and minimum singular values of the quadrotor over the frequencies of interest gives us Fig. 5.1

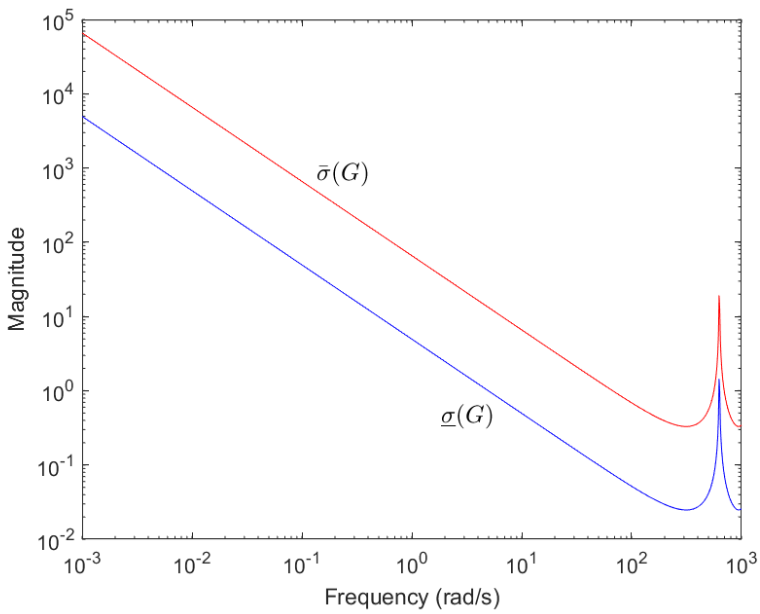


Figure 5.1: The frequency dependent maximum and minimum singular values of the quadrotor

The maximum and minimum singular values can be derived from the singular value decomposition (SVD) of the system matrix at various frequencies. A SVD essentially decomposes the system matrix $\mathbf{G}$ as follows:

$$\mathbf{G} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^H \tag{5.2}$$

where $\mathbf{U}$ and $\mathbf{V}$ represents the output, and input singular vectors respectively. The input directions corresponding to the maximum and minimum singular values are first and last columns of $\mathbf{V}$ respectively. The matrix $\boldsymbol{\Sigma}$ in eq. (5.2) which is a diagonal matrix, contains the singular values arranged in decreasing order. $\boldsymbol{\Sigma}$ for the quadrotor model at around the bandwidth frequency (see section 5.2.5) calculates to below matrix:

$$\mathbf{\Sigma} = \begin{bmatrix} 5.9172 & 0 & 0 & 0 \\ 0 & 5.9172 & 0 & 0 \\ 0 & 0 & 0.6782 & 0 \\ 0 & 0 & 0 & 0.4446 \end{bmatrix} \tag{5.3}$$

Alongside the interactions observed in the singular input and output directions, the values from the diagonal of the $\mathbf{\Sigma}$ matrix can be used to measure if the system is ill-conditioned. The condition number defined as $\bar{\sigma}/\underline{\sigma}$ can be calculated as a way to determine significance of large off-diagonal elements of the system matrix , alongside effects of multi-variable uncertainty. The quadrotor model remains at and around 13.308 through out the frequency of interest.

In general, plants with condition number greater than 10 usually exhibit control problems (such as those with an inverse based controller [197] as well as with multivariable parameter uncertainty [198]) although there are exceptions to this ([118] pg. 82). With a condition number of 13.308, effects on performance due to unstructured (and to a certain extent, structured) uncertainty, can be expected during the controller development.

Based on the insights obtained from the previous discussions, the development of multivariable robust controllers are discussed in the following sections. The MIMO quadrotor model will be used to test the developed frameworks.

5.2 Mixed Sensitivity Optimization

This section is further subdivided into two main parts. In the first part the theory behind multivariable controller development will be introduced and in the second part the controller development based on the MIMO quadrotor will be presented.

The theoretical background of mixed sensitivity optimisation explained in section 4.4 in the previous chapter directly translates over to its multivariable counterpart. The differences in formulation of the $\mathcal{H}_\infty$ problem, those while developing the design weights as well as the differences in the performance criteria will be presented in the following subsection.

5.2.1 Multivariable controller development framework

The S/T problem described in section 4.4.1 will remain as the framework for development of the robust control MSO problem. With the model presented in Fig. 4.3, the $\mathbf{P}$ matrix remains the same (eq. (4.11)) and the cost function is given by

$$\gamma = \left\| \begin{matrix} \mathbf{W}_P\mathbf{S} \\ \mathbf{W}_I\mathbf{T} \end{matrix} \right\|_\infty \tag{5.4}$$

where the infinity norm of the matrix transfer function is now calculated as a explained in section 4.3.1, namely the peak of the maximum singular value, $\bar{\sigma}$ over the frequency of interest.

Similarly with regards to developing the design weights, i.e. the sensitivity and complimentary sensitivity weights, the steps presented in section 4.4.2 directly translates to the multivariable domain with few changes. The concepts relating to eq. (4.13) and eq. (4.14) remain the same.

Sensitivity weights and complimentary sensitivity weights turn out to be matrices with transfer function weights along its diagonals. The complimentary sensitivity weight matrix, which captures the model uncertainty (categorised as 'Structured' input multiplicative uncertainty in this case) is given by $\mathbf{W}_I = \mathrm{diag}\{w_{I_1}, w_{I_2}, \cdots, w_{I_k}\}$ where k denoted the number of inputs. Here w_{I_k} is derived similar to eq. (4.15), by calculating $l_{I_k}(j\omega)$ as follows:

$$l_{I_k}(j\omega) = \max_{\mathbf{G}_p \in \prod} \bar{\sigma}(\mathbf{G}(k,:)^{-1}(\mathbf{G_p}(k,:) - \mathbf{G}(k,:))) \tag{5.5}$$

with w_{I_k} developed such that $w_{I_k}(j\omega) \geq l_I(\omega) \, \forall \, \omega$. For successful development of the controller it is required that $\|\mathbf{W}_I\|_\infty$ remain less than unity. The sensitivity function $\mathbf{W}_P$, similarly is a matrix with transfer functions along the diagonal. To develop $\mathbf{W}_P$, similar to its SISO counter part, we develop an outer framework that resolves a nonlinear constrained optimisation problem. The cost function essentially remains the same (see section 4.4.3). The optimisation framework both develops the controller alongside the sensitivity weight. The algorithm is presented in the next subsection.

5.2.2 Algorithm to Develop the Multivariable MSO Controller

Algorithm #5 can be followed to develop the sensitivity weights alongside the multivariable optimal MSO controller.

Algorithm # 5 Sensitivity weight and Multivariable Controller design procedure for S/T MSO

Inputs: Plant model $\mathbf{G}$, design constraints, model uncertainty estimate.
Outputs: $\mathbf{K}$, $\mathbf{W}_I$, $\mathbf{W}_P$.

1. Scale the given plant $\mathbf{G}$ according to the criteria described in section 3.3
2. Calculate the complementary sensitivity weight $\mathbf{W}_I$ according to the procedure described in section 5.2.1.
3. Depending on the form of $\mathbf{W}_I$ (based on unstructured or structured uncertainty that is used to model the system), define the appropriate form for $\mathbf{W}_P$ (scalar or diagonal matrix transfer function) and choose the initial parameters.
4. Define the cost function $\mathcal{J}$ and initialize the weights W1 through W4.
5. Develop the vector $\mathbf{\Phi}$ which contains the following nonlinear closed loop functionals
   1. Maximum overshoot, M_p
   2. Rise time, t_p
   3. Settling time, t_s
   4. Maximum value of input, u_{max}
   5. gamma, γ

   Here M_p, t_p and t_s, the step response characteristics and u_{max} are the time domain specifications while γ refers to the frequency domain specification. To constrain the closed loop functionals calculated for every output, they are stacked together into a single vector.
6. Define the frequency domain bounds ϵ_γ and time domain specifications limit vector $\boldsymbol{\epsilon}$.
7. Implement the constrained non-linear optimization problem. This is achieved using the OPTI MATLAB toolbox. The black box optimizer NOMAD is chosen as the optimization routine. Constraints are defined based on ϵ_γ and $\boldsymbol{\epsilon}$ for the vector $\mathbf{\Phi}$ that defines the closed loop functionals.
8. Depending on robustness requirements, modify the weights $W1$ through $W4$ to reach the required performance levels.
9. If satisfactory performance specifications are not met, decrease the % uncertainty considered while defining $\mathbf{W}_I$ and relax the bounds defined in step 6 and retry step 7.

The algorithm might seem very much similar to that of **Algorithm #1** (see section 4.4.4), and in many ways it is. The differences between the two are in aspects of the form of the weights $\mathbf{W}_I$ and, $\mathbf{W}_I$, the structure of $\mathbf{\Phi}$, and the way the cost function is calculated considering the various outputs from the system. For further details regarding other aspects concerning the algorithm see section 4.4.4.

5.2.3 Performance Analysis Criteria - Multivariable MSO

When it comes to analysing the performance of the controller in the frequency domain the criteria described in Table 5.1 can be used.

Table 5.1: Performance Criteria

Performance Criteria	Necessary Constraint
Nominal Stability	N internally stable
Nominal Performance	$\bar{\sigma}(\mathbf{N}_{22}) < 1;\ \forall\omega$
Robust Stability	$\mu_{\Delta}(\mathbf{N}_{11}) < 1;\ \forall\omega$
Robust Performance	$\mu_{\Delta}(\mathbf{N}) < 1;\ \forall\omega$

For the definition of $\mathbf{N}$ see section 4.3.2. Here μ_{Δ} represents the value of μ while the uncertainty Δ is structured. In addition to the frequency domain response the time domain step response characteristics will also be analysed based on the design specifications listed in section 3.5. In the following section these concepts will be applied to design the multivariable attitude-altitude tracking controller for the quadrotor model, and the performances will be analysed.

5.2.4 Multivariable MSO controller for the Quadrotor Application

In this section the tracking attitude-altitude quadrotor controller developed based on ideas discussed previously will be presented. The complimentary sensitivity weight, the sensitivity weight, the closed loop bandwidth, the time and frequency domain characteristics of the closed loop system will be explored.

5.2.4.1 Complimentary sensitivity weight

The uncertainty sources described in Table 3.2, being uncertainty in parameters, time delays and model uncertainty due to linearisation, has been considered while $\mathbf{W}_I$, the complimentary sensitivity weight was developed. As described earlier, the chosen structured multiplicative input uncertainty, leads to the following description of the uncertain plant: $\mathbf{G_p} = \mathbf{G}(1 + \mathbf{W}_I\boldsymbol{\Delta})$. The developed weight matrix is given below:

$$\mathbf{W}_I = \begin{bmatrix} \dfrac{1.933s + 34.17}{s + 57.2} & 0 & 0 & 0 \\ 0 & \dfrac{1.995s + 41.11}{s + 57.88} & 0 & 0 \\ 0 & 0 & \dfrac{1.989s + 17.24}{s + 49.44} & 0 \\ 0 & 0 & 0 & \dfrac{1.982s + 42.11}{s + 57.82} \end{bmatrix}$$

The weight development was completed by lowering the uncertainty (see Table 5.2) for example in the thrust coefficient c_t to 20 % (from 50 %) and lowering uncertainty in mass to 20 % (from 50 %), among others. This adjustment had be to made inorder to develop a weight with an infinity norm less than unity.

While this might seem like a hack, what this essentially means in the real world, is that we can only permit for the model to have, say 20% uncertainty in mass and thrust coefficient, if we are hoping to design a robust MSO controller that upholds the required design standards. Hence rather than a hack, it turns out to be a way in which the nature of the physical plant, shines light on the constraints it wields on the achievable limits of robustness, on all possible MSO controller designs.

5.2.4.2 Optimisation Problem

The formulation of the optimisation problem remains the same as that given in 4.4.6.2, except for γ which represents the stacked cost function given in eq. (5.4). The new objective w.r.t γ would be to minimize its magnitude instead of constraining it to less that unity. The optimisation problem will be used in the development of the sensitivity weight and the controller, which is presented in the next section.

5.2.4.3 Sensitivity weight choice and the MSO controller

As noted in step 9 of **Algorithm #5**, the % uncertainty is needs to be decreased as required so that the design constraints are met. This means, the developed controller can meet the design requirements only when model uncertainty is below the lowered new values. As such, for the development the new uncertainty values that were used are listed in the Table 5.2:

Table 5.2: Uncertainty Sources

Parameters	% Uncertainty	Range
I_{XX}, I_{YY}	0.04339 $\pm$10% $\mathrm{kg\,m^2}$	$0.0391 \leq I_{XX},\, I_{YY} \leq 0.0477$
I_{ZZ}	0.0705 $\pm$10% $\mathrm{kg\,m^2}$	$0.0634 \leq I_{ZZ} \leq 0.0775$
m	1.27 $\pm$20% kg	$1.0160 \leq m \leq 1.5240$
dr	0.194$\pm$20% m	$0.1552 \leq d_r \leq 0.2328$
c_t	0.1 $\pm$20%	$0.0800 \leq c_t \leq 0.1200$
c_p	0.045 $\pm$20%	$0.0360 \leq c_p \leq 0.0540$
t_D	0.004 $\pm$10% s	$0.0036 \leq t_D \leq 0.0044$

Variables	Range for Model Linearisation
ϕ_i, θ_i, ψ_i	-10 $\leftrightarrow$ 10 (deg)
$\dot{\phi}_b$, $\dot{\theta}_b$, $\dot{\psi}_b$	-0.1 $\leftrightarrow$ 0.1 (rad/s)
$\dot{x}_i$, $\dot{y}_i$, $\dot{z}_i$	-0.25 $\leftrightarrow$ 0.25 (m/s)
u_1, u_2, u_3, u_4	4490 $\leftrightarrow$ 4890 (RPM)

The designed sensitivity weight $\mathbf{W}_P$ is given below:

$$\mathbf{W}_P = \begin{bmatrix} \frac{0.356s+10.03}{s+0.187} & 0 & 0 & 0 \\ 0 & \frac{1.969s+53.05}{s+0.09886} & 0 & 0 \\ 0 & 0 & \frac{0.06848s+2.375}{s+0.0482} & 0 \\ 0 & 0 & 0 & \frac{0.1426s+3.489}{s+0.02092} \end{bmatrix}$$

The controller itself is presented in Appendix D.2. The resulting closed loop gave the following time and frequency domain specifications with the linear plant model.

Table 5.3: Time response characteristics and frequency domain specification of the linear closed loop MIMO model

Closed Loop Constraints	Roll Rate	Pitch Rate	Yaw Rate	Altitude Rate
% Overshoot	0	0	1.779	1.140
Rise Time (s)	0.105	0.041	0.489	0.349
Settling Time (s)	0.175	0.078	0.739	0.540
u_{max}	0.999			
γ	1.989			

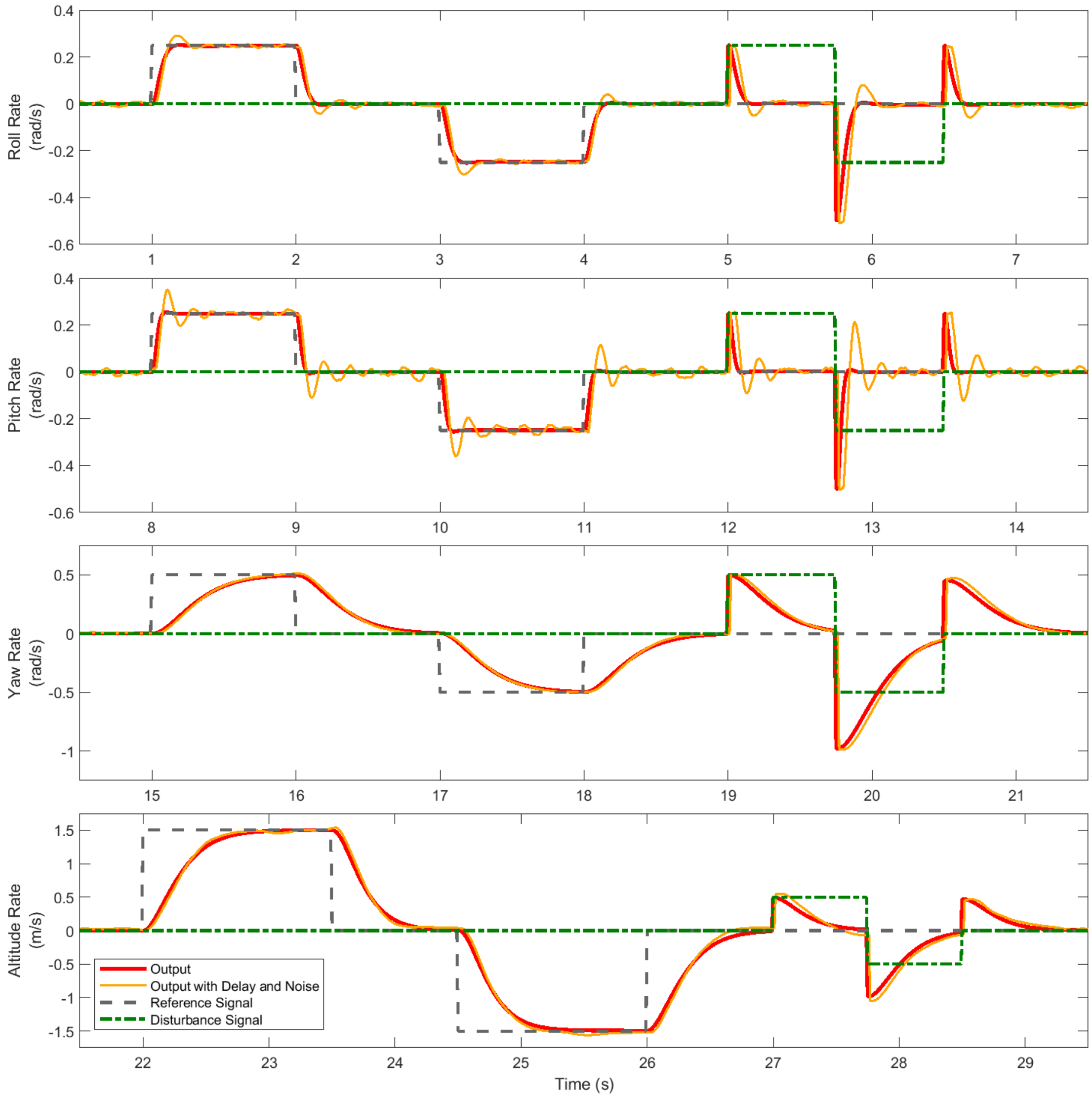


Figure 5.2: Step response of the nominal multivariable MSO controlled plant

The developed controller performs well in terms of achieving the desired performance for the nominal plant as can be seen in the nonlinear plant response in Figures 5.2 and 5.3. The orange curve in Fig. 5.2 represents the response to the reference input alongside a time delay of two sample units and a zero mean noise signal with a standard deviation equal to 30% of the maximum reference value, injected at the output of the plant. The light blue curve in Fig.5.3 shows the corresponding inputs.

Further experiments showed that the system remains stable even when the standard deviation of the noise signal is increased to 50% and time delay is increased to 3 sample times (0.03s). The yaw and altitude rates showed excellent resistance to both the noise and time delay disturbance.

(Note: The differences in elements of the sensitivity function $W_P(1,1)$, and $W_P(2,2)$, corresponding to the roll and pitch rates, and the subsequent variation/differences observed in the nature of roll and pitch rate plant responses in Fig. 5.2, can be attributed towards the nature of the path taken by the optimizer while solving the nonlinear optimization problem. Similar weights and hence symmetric reactions can be elicited from the optimizer if the upper and lower bounds of the parameters of both $W_P(1,1)$, and $W_P(2,2)$ are identically tightened.)

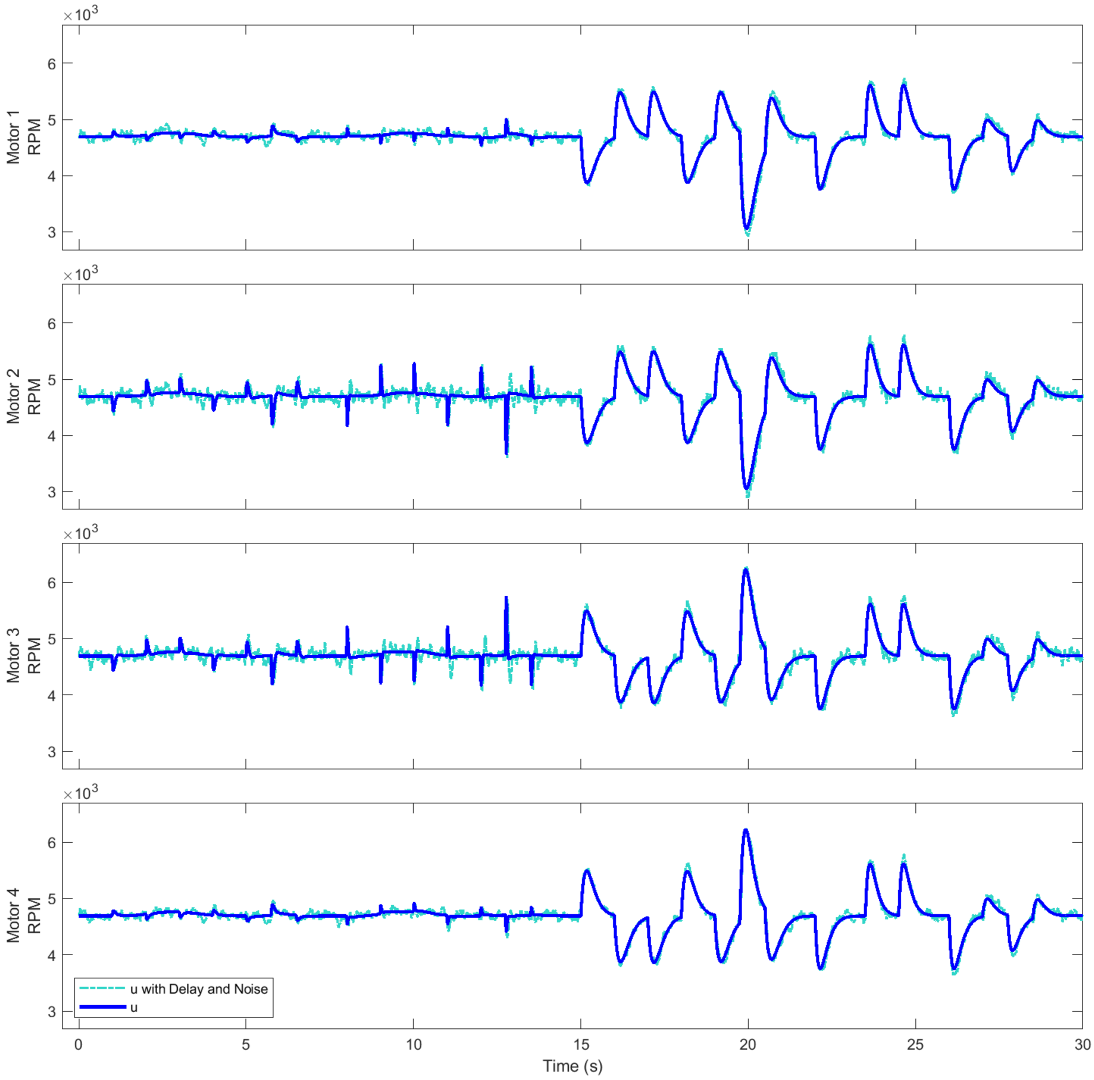


Figure 5.3: Control inputs, the angular velocities of individual motors in response to reference signal

The response of the plants with model uncertainty is presented in Fig. 5.4. The plant is subjected to the same complex manoeuvre, that was initially presented in section 4.4.6.10. (For the explanation for the various shades please see section 4.4.6.10).

While for the nominal plant, satisfactory performance and disturbance rejection is achieved, in the case with model uncertainty, the altitude rate response appears to lack the integral action. This issue can be resolved in two ways.

The traditional way to approach the problem is by slightly tuning the altitude rate integrator parameter (parameter $\mathbf{A}(9,9)$ in the controller state matrix manually. The second way to approach the problem would be by referring to Step 9 in **Algorithm #5**. The constraint parameters are slightly relaxed until sufficient integral action is obtained in the response.

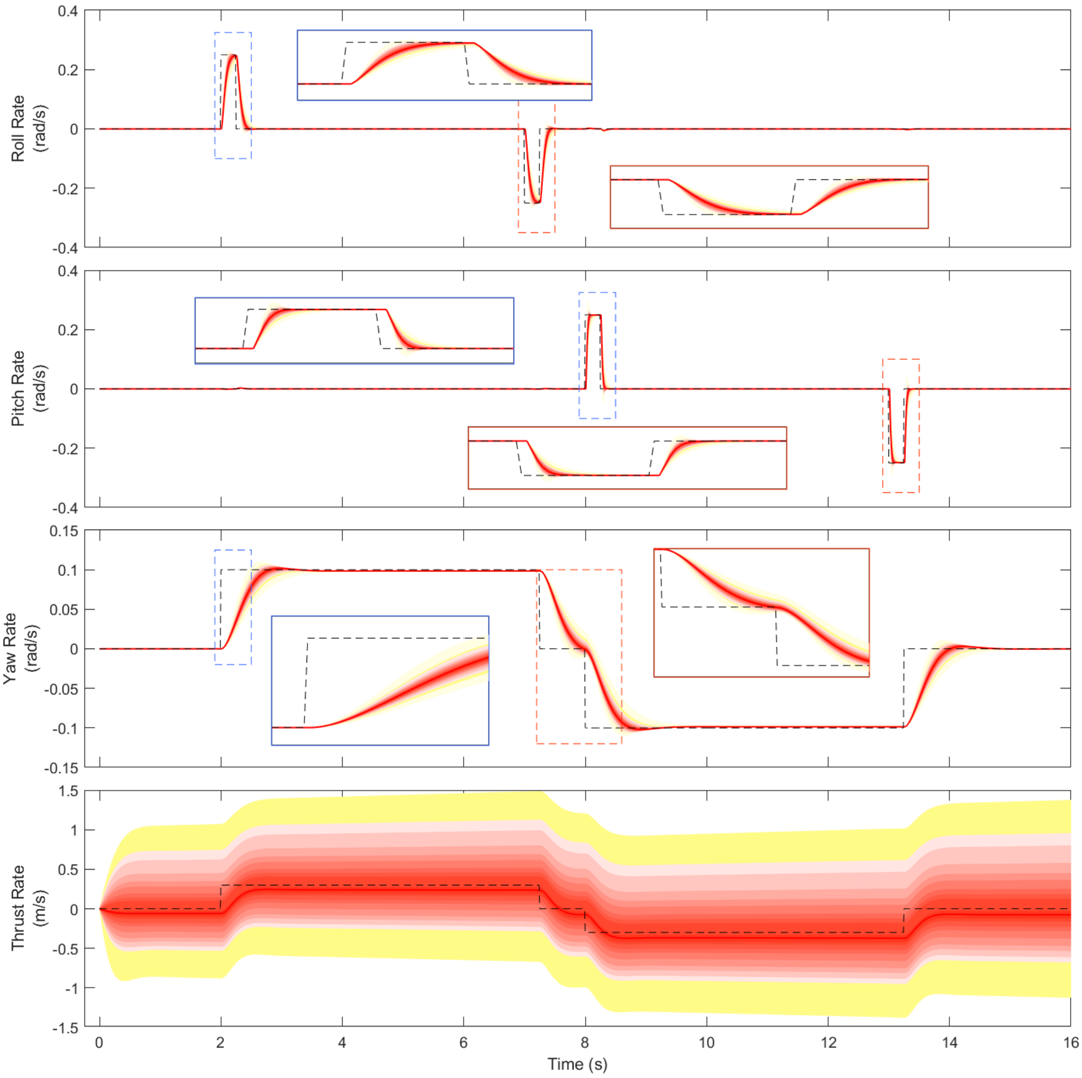


Figure 5.4: The response of the Multivariable MSO based plant with model uncertainty to the complex manoeuvre reference command- initial attempt.

While the issue is resolved this comes at the expense of 32.41 % overshoot and a settling time of 1.493 seconds for the nominal plant response, and the extra oscillatory dynamic at the beginning of the simulation.

The Inputs to the system i.e the motor RPM is presented in the Fig. 5.6 and the non-linear plant step response characteristics after the final tuning is listed in Table 5.4. We have satisfactory performances for the uncertain plant models. The nonlinear plant response characteristics are, except for Altitude rate overshoots, which had moved past the hard limit set earlier in section 3.5.

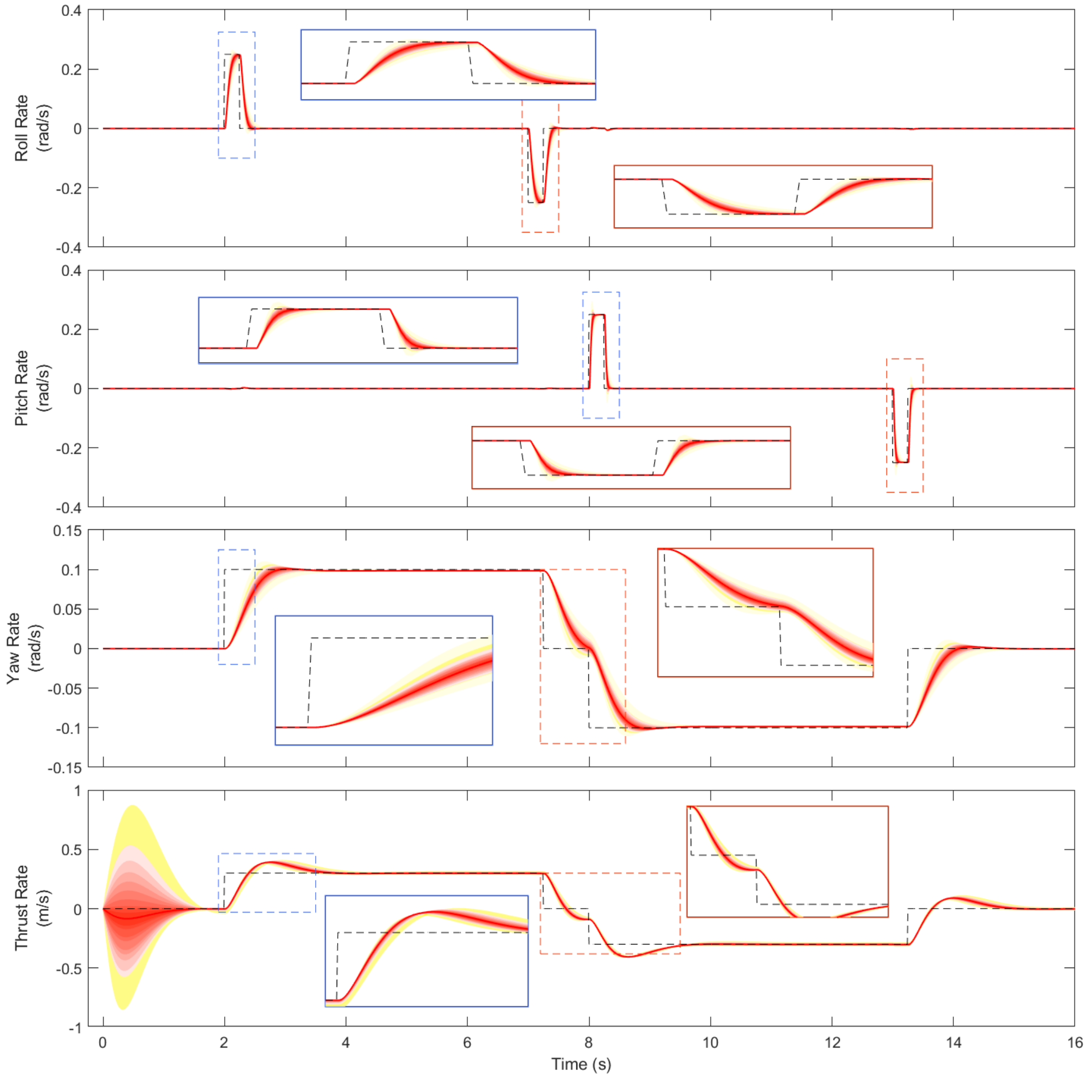


Figure 5.5: Multivariable MSO based plant response to the complex manoeuvre reference signal- after tuning

Although the loop maintains nominal as well as robust stability, relaxing the limits set in design specifications (section 3.5) resulted in a single instance of constraint violation. This violation is compensated on the other hand through an enhanced the integral action in the altitude rate the system as we have seen above.

From the experiments conducted and observations made it was concluded that uncertainty in craft mass makes it difficult to design controllers that can guarantee robust performance, while simultaneously maintaining strict performance standards.

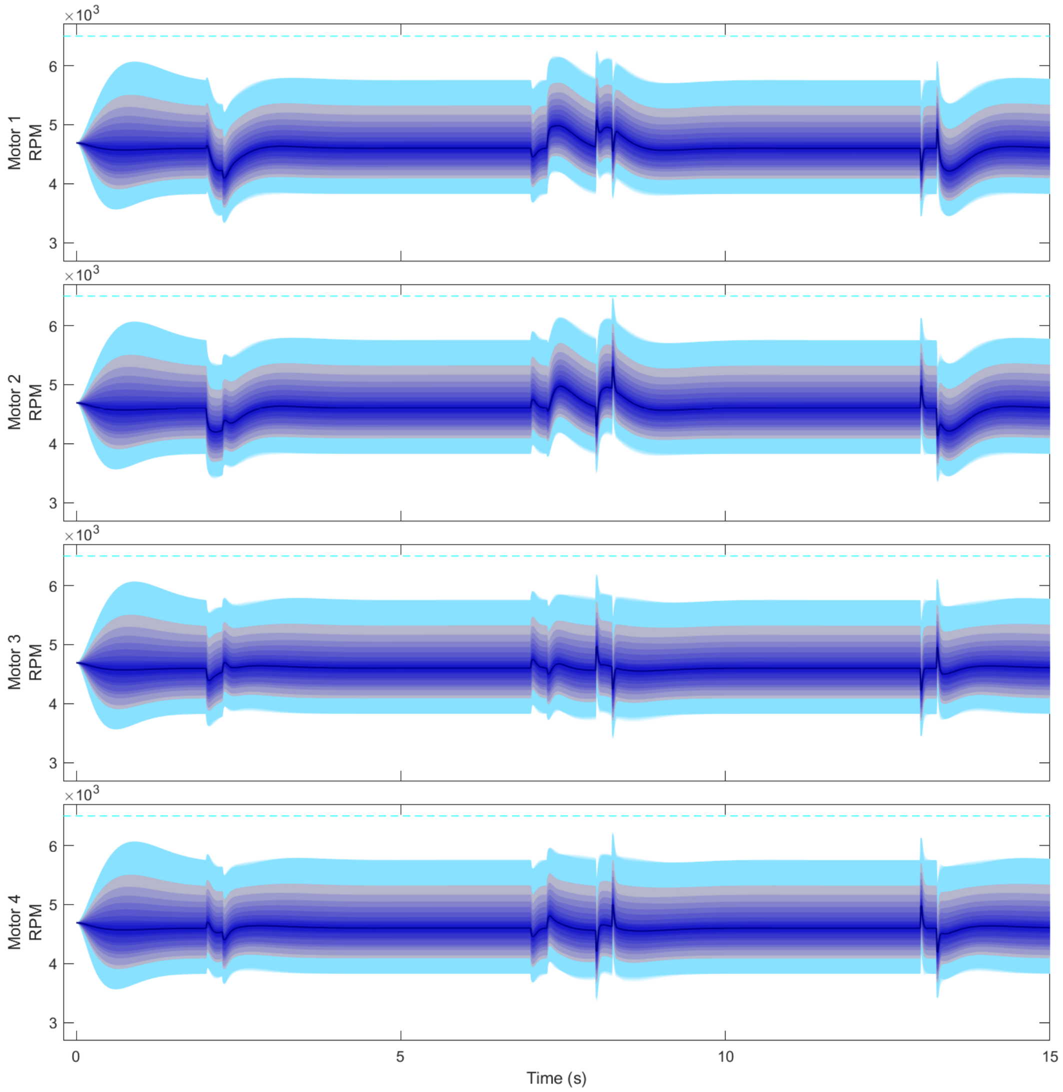


Figure 5.6: Multivariable MSO controlled plant rotor responses to the complex manoeuvre reference signal.

Table 5.4: Closed loop time domain specifications after controller tuning

Response Characteristics	Roll Rate	Pitch Rate	Yaw Rate	Altitude Rate
Rise Time (s)	0.090	0.047	0.526	0.284
Settling Time (s)	0.149	0.081	0.728	1.493
Peak	0.250	0.248	0.505	1.986
Peak Time (s)	0.121	0.150	0.566	0.760
% Overshoot	0.000	0.000	0.934	**32.413**
Undershoot	0.000	0.000	0.237	0.000

In the next section the performance criteria listed in Table 5.1 will be presented. The frequency dependent values of μ calculated for determining nominal stability, robust stability and robust performance will also be presented.

5.2.4.4 Performance criteria

The performance criteria, calculated as explained in 5.1 are listed out in Table 5.5. For *nominal performance* and *robust stability*, we have have the infinity norm of μ calculated over the frequency spectrum of interest to be below unity, while that of the *robust performance* falls above unity.

This can further be observed in the μ-frequency plots presented in Fig. 5.7. As can be seen the values of μ calculated for robust performance falls above unity (the black dashed line) at steady state frequencies.

Table 5.5: Robust Performance Analysis

	Nominal Performance	Robust Stability	Robust performance
μ	0.489	0.49	2.42

As previously observed in 4.6 where in SISO MSO could fail to strictly guarantee RP, in MIMO systems MSO similarly cannot also guarantee to push μ values to below unity across the entire frequency spectrum.

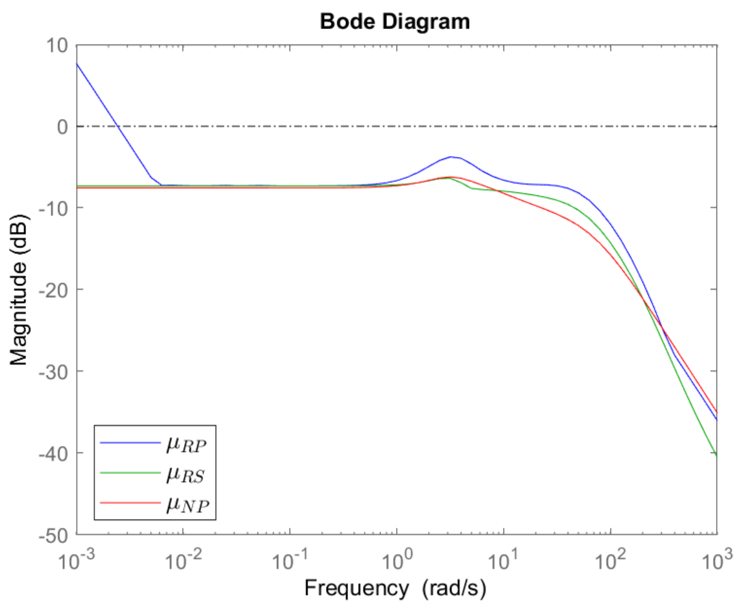


Figure 5.7: Frequency dependent μ-plots

In the following section we look at the closed loop bandwidth properties of the system.

5.2.5 Closed loop bandwidth

Closed loop bandwidth in multivariable systems, as presented in [118] (pg. 81) can be defined as the frequency where $\bar{\sigma}$(S) crosses 0.707 from below. This definition also translates as to the frequency where $\underline{\sigma}$ crosses unity.

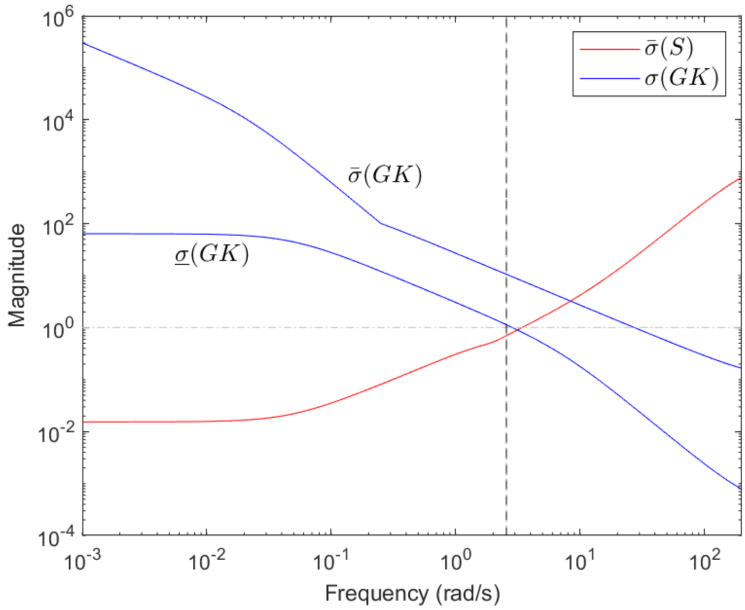


Figure 5.8: Multivariable MSO system: Singular value plots of Sensitivity and Loop Function

The singular values of sensitivity function (S) and the loop function ($L = GK$) are plotted against frequency in Fig. 5.8. The dashed line indicates the frequency where the sensitivity function crosses 0.707 from below. The bandwidth calculates to 2.56 rad/s (0.41 Hz). This indicates that while designed controller is resistant to model uncertainty arising from parametric, time delay and irregularities in linearisation, the quadrotor would be effective to reference commands that have a time period of about 2.4 seconds and slower (0.41 Hz). While effectiveness can be expected at higher frequencies in certain directions, as we can observe that $\bar{\sigma}(GK)$) crosses unity at 27 rad/s or 4.3 Hz (time periods of 0.23 seconds and above), the overall effective time period of reference commands tends to be around 2 to 2.4 seconds.

5.2.6 Conclusion of multivariable MSO controller design

This concludes the end of multivariable MSO controller design and analysis. The section details the formulation of a framework (**Algorithm #5**- see section 5.2.2) that can generate optimal (or suboptimal as needed) controllers alongside the controller weights. The designed quadrotor attitude/altitude controller, while remaining suboptimal (as μ falls above unity for *robust performance*), gives a satisfactory responses for models with uncertainty. The bandwidth indicates that the closed loop system will react robustly to slow commands (i.e. those with a time period of around 2.5 seconds). This conservative nature is expected owing to the coupled nature of the plant outputs controlled by a multivariable controller that strives to maintain step response constraints alongside robustness. In the next section we will look at the design and implementation of 1 DOF LSDP multivariable controllers.

5.3 Loop Shaping Design Procedure - 1 DOF

In this section the development of the multivariable robust controller using the loop shaping procedure single degree of freedom approach is discussed. The design procedure is based on a technique presented in [199], where a $\mathcal{H}_\infty$ controller is cascaded with a PID controller for stabilizing an open loop unstable system. The technique employed reduces the complexity of overall optimization problem, there by ensuring convergence alongside the development of a satisfactory controller.

The section is divided into two main parts. In the first part the controller development framework which is based on the concepts presented in section 4.6 will be briefly explained. In the second part, the developed multivariable controller for the quadrotor will be presented.

5.3.1 Multivariable controller development framework

Similar to the MSO multivariable desgin in section 5.2.4, the concepts from the SISO LSDP theory (see section 4.6) translates directly to the multivariable domain with few changes. The $\mathcal{H}_\infty$ norm of the cost function γ given in equation eq. (4.39) becomes $\bar{\sigma}$ of the cost function matrix.

The weights $\mathbf{W_1}$, and $\mathbf{W_2}$ are matrices with transfer functions and simple scalars arranged along its diagonals respectively. The prefilter $\mathbf{K_P}$ is also a matrix with simple scalars arranged along the diagonal.

Unlike the design presented in SISO LSDP 1DOF controller development, in the weight $\mathbf{W_2}$, transfer functions are replaced by simple scalars. In the initial experiments with transfer functions along the diagonal of $\mathbf{W_2}$, the resulting optimization problem was too complex for the optimizer to resolve satisfactorily. The new technique while it would require the control engineer to further design the the cascaded PI controllers, reduced the overall complexity of the optimization problem (as well as the overall order of the multivariable controller), resulting in a satisfactory controller.

The transfer functions along the diagonal of $\mathbf{W_1}$ takes the format from eq. (4.40). In the next subsection the algorithm to be followed to develop the optimization framework which leads to the development of the multivariable controller is presented.

5.3.1.1 Algorithm to Develop the Multivariable LSDP 1 DoF Controller

The **Algorithm #6** can be used to develop the multivariable controller alongside the loop shaping design weights and the prefilter.

Algorithm # 6 Design weights, prefilter and Multivariable Controller development algorithm for LSDP 1DOF

Inputs: Plant model $\mathbf{G}$, design constraints.
Outputs: $\mathbf{K}$, $\mathbf{W_1}$, $\mathbf{W_2}$, $\mathbf{K_P}$.

1. Scale the given plant $\mathbf{G}$ according to the criteria described in section 3.3
2. Define the appropriate form of the designs weights $\mathbf{W_1}$, $\mathbf{W_2}$ and the prefilter $\mathbf{K_P}$ and initialize the weights.
3. Choose the magnitude of the γ_{min}
4. Follow step 4 through 7 from **Algorithm #2** (on page 85).
5. If the optimizer fails to converge, linear constraints can be relaxed and the form of $\mathbf{W_1}$ can be changed (if required), after which step 4 can be repeated. As the optimizer achieves convergence, these constraints can tightened and brought back to the required values.

Unlike the SISO case, in the multivariable optimization framework the nonlinear constraints (some of which are themselves vectors) are stacked together to form a column vector with 17 rows. The upper and lower bounds are then assigned depending on a) the constraint, and b) the loop that is being considered. While developing the cost function, the impact of various outputs on the system can be separated (if required), and priorities assigned by using separate weights. For more details regarding other aspects concerning the algorithm, see section 4.4.4

While **Algorithm #6** develops the multivariable controller, the 4 PI controllers (corresponding to each of the output) that are cascaded with it are tuned using the SIMULINK PID Tuner [181]. The tuning process is relatively straight forward and can be performed with ease after the LSDP controller is designed (see section 4.4.6.9). The controllers are tuned such that the bandwidth of the closed loop complies with that of design requirements (section 3.5) and required closed loop stability margins are achieved.

5.3.2 Performance criteria

The gap metric stability criterion introduced in section 4.6.2 will be used to analyse the effectiveness of the multivariable controller. If we have the value of ϵ_{max} >0.25, then the design will be considered successful.

Alongside the stability margin criteria, the time domain specifications of the nominal closed loop needs to be within the required limits (as noted in section 3.5). As a part of further analysing the system, the frequency dependent μ values will also be calculated and plotted.

5.3.3 Designing the multivariable LSDP 1DoF controller

This section presents the developed prefilter and controller weights, the developed PI controllers alongside the results obtained from subjecting the system to step inputs and complex manoeuvres. The time domain characteristics, the closed loop bandwidth and the frequency domain characteristics will also be explored here.

5.3.3.1 Controller weights, Prefilter and the PI controllers

While in MSO we require to capture uncertainty using the complimentary sensitivity weight, in LSDP the co-prime factor uncertainty representation enables the designer to directly develop the controller that can provide the maximum stability margins against the uncertainty. As such following **Algorithm #6**, the following weights are developed for the tracking controller, for the quadrotor model presented in section 3.2.

$$\mathbf{W_1} = \begin{bmatrix} \dfrac{216.9s+3880}{200.6s+2387} & 0 & 0 & 0 \\ 0 & \dfrac{1722s+2133}{606.6s+927.1} & 0 & 0 \\ 0 & 0 & \dfrac{5.139s+185.1}{2.964s+86.07} & 0 \\ 0 & 0 & 0 & \dfrac{2.115s+330.7}{1.977s+151.4} \end{bmatrix}$$

$$\mathbf{W_2} = \begin{bmatrix} 0.2733 & 0 & 0 & 0 \\ 0 & 0.3260 & 0 & 0 \\ 0 & 0 & 0.4537 & 0 \\ 0 & 0 & 0 & 0.3550 \end{bmatrix}, \ \mathbf{Ks0} = \begin{bmatrix} 0.9913 & 0 & 0 & 0 \\ 0 & 0.9782 & 0 & 0 \\ 0 & 0 & 1.0208 & 0 \\ 0 & 0 & 0 & 1.0137 \end{bmatrix}$$

.

The form of the tuned PI controller as well as the values of the constants are presented in Table 5.6.

Table 5.6: Proportional and integral constants of the cascaded PI controller

$\mathrm{P}\big(1+\mathrm{I}\cdot\mathrm{T}_s\frac{1}{z-1}\big)$	Roll Rate	Pitch Rate	Yaw Rate	Altitude Rate
P	1.676	0.709	6.445	2.438
I	0.305	0.240	2.025	3.893

The LSDP controller itself is presented in Appendix D.2.

Table 5.7: Closed loop time domain specifications after controller tuning

Response Characteristics	Roll Rate	Pitch Rate	Yaw Rate	Altitude Rate
Rise Time (s)	0.031	0.083	0.161	0.201
Settling Time (s)	0.062	0.143	0.626	1.407
Peak	0.259	0.248	0.562	1.835
Peak Time (s)	0.060	0.380	0.270	0.500
% Overshoot	3.747	0.00	12.439	22.32
Undershoot	0.00	0.00	0.221	0.039

5.3.3.2 Closed loop time domain characteristics

The time and frequency domain characteristics of the developed controller plant model are presented next. The true nonlinear system is subjected to step inputs which are exactly the same inputs as those presented in section 4.4.6.6, except for altitude rate where the square wave has a length of 1.5 seconds (as opposed to 1.25s). The time domain closed loop response characteristics are presented in Table 5.7. The peak values should be seen in the context of the reference signal (see page 94 for details).

As observed in Table 5.7 acceptable performances in terms of meeting the design constraints have been achieved. The actual responses to these setpoints are shown in Figures 5.9 (outputs) and 5.10 (inputs).

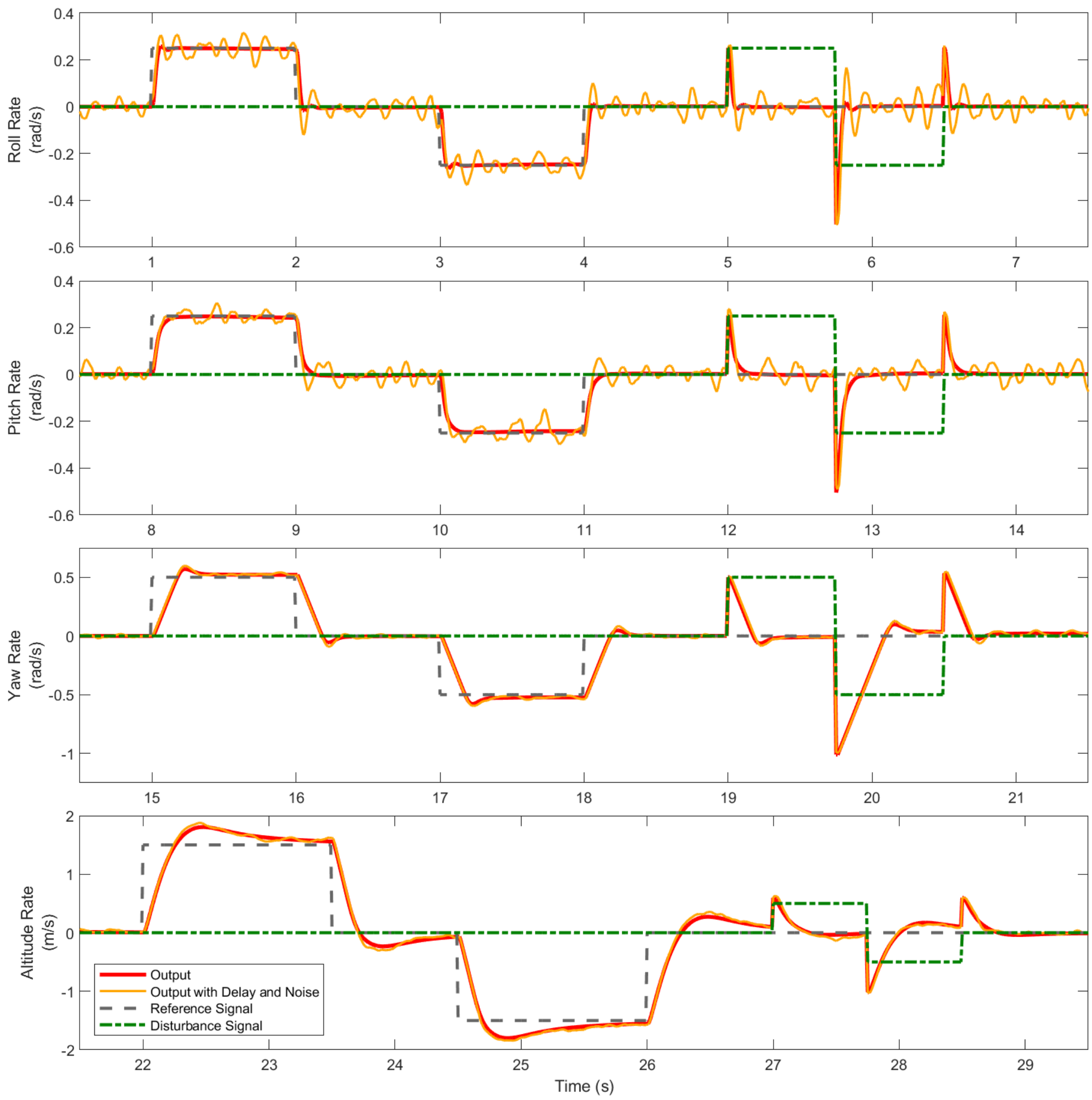


Figure 5.9: The outputs of the 1 DoF LSDP-PI controlled nonlinear plant.

In Fig. 5.9 the orange curve represents the response to the reference input alongside the presence of unit delay and zero mean noise signal with a standard deviation equal to 15% of the maximum reference value, injected at the output of the plant to simulate measurement noise. As can be observed the roll and pitch rate loops are relatively more susceptible to noise signal, compared to the yaw and altitude rate loops.

Further experimentation showed that, the yaw and altitude rate loops are resistant towards both greater time delay and larger noise, while the pitch and roll rate loops appear vulnerable especially to stronger noise signals.

Fig. 5.10 presents the motor inputs in response to the reference setpoint. The simulation conducted in the presence of the injected noise signals are indicated using the light blue curves. The signals can be seen to reach the saturation limits although the controller maintains stability and the system offers acceptable performance.

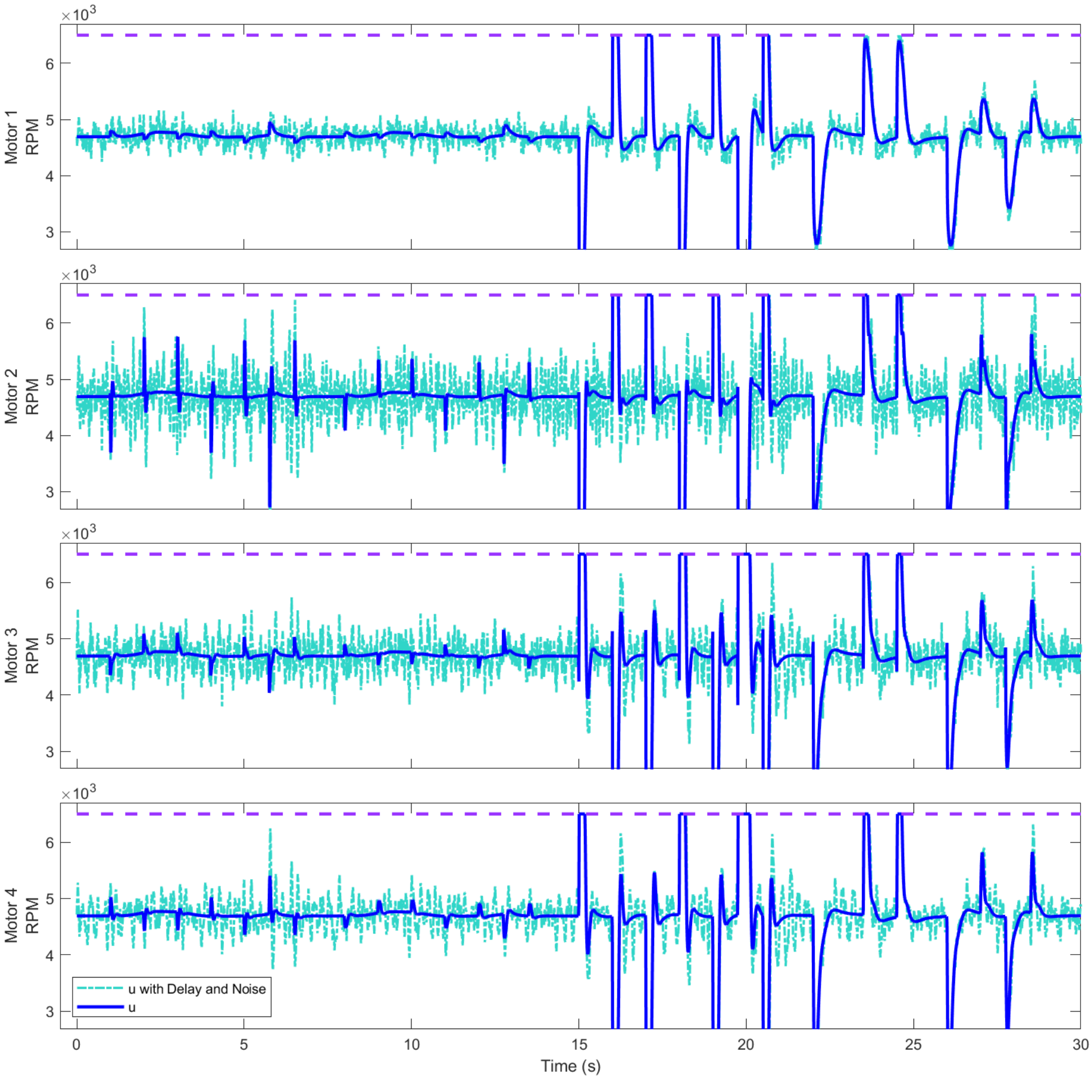


Figure 5.10: Motor inputs of the LSDP-PI controlled system in response to the step commands

The performance of the plant in the presence of parameter uncertainty is explored next. The parametric uncertainty source ranges are those presented in first list of Table 5.2. The plant is subjected to the complex manoeuvre presented in section 4.4.6.10.

The response on the plants with parametric uncertainty is presented in Fig. 5.11. For details regarding the meanings of the colours used, refer section 4.4.6.10. As observed in the figure, the plants are able to track the step inputs closely. An oscillatory dynamic is observed at the beginning of the simulation which dies out fast. The dynamic has its largest magnitude in the altitude rate loop. Overshoots are observed in pitch, yaw and altitude rates loops.

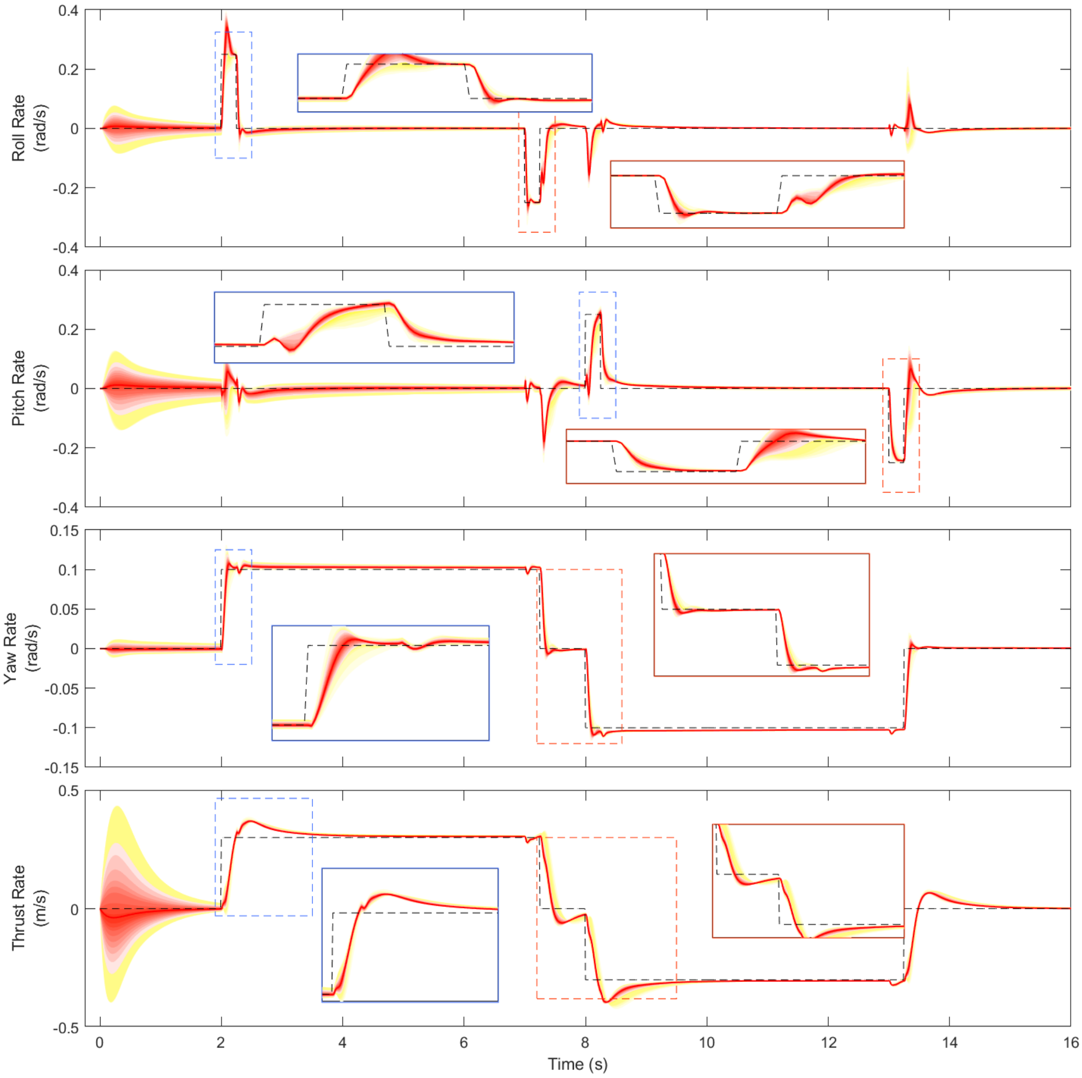


Figure 5.11: The response of the Multivariable LSDP 1-DOF based plant with model uncertainty to the complex manoeuvre reference command.

The response of the uncertain plants to the reference input is observed in Fig. 5.12. The light blue dotted line represents the motor RPM saturation limits. Spikes can be observed in the system response corresponding to the points where the inputs hits this saturation limit.

In the following section the performance criteria will be examined in the context of the developed controller. The frequency domain characteristics of the system will also be analyzed.

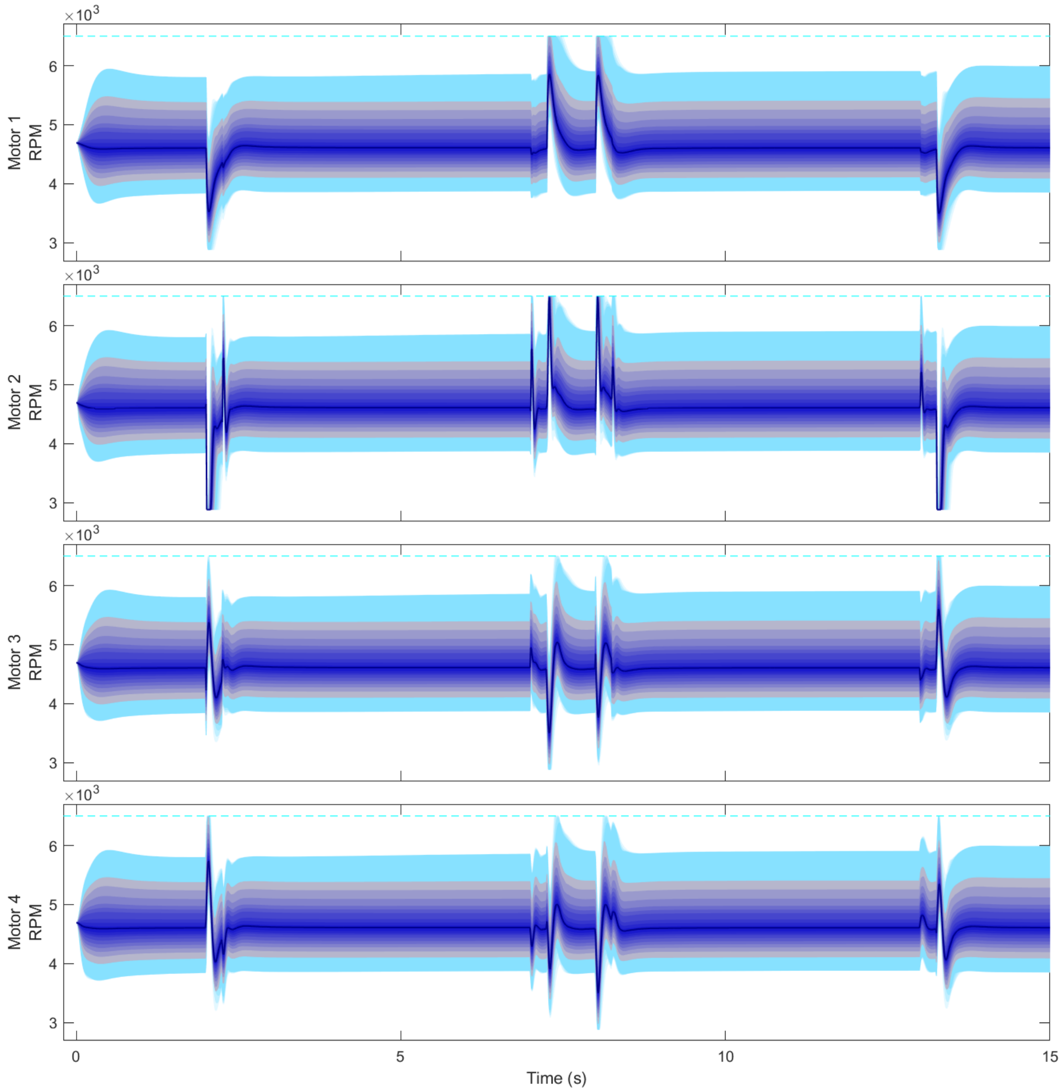


Figure 5.12: Multivariable LSDP 1-DOF controlled plant rotor responses to the complex manoeuvre reference signal.

5.3.3.3 Performance

The normalized co-prime stability margin of the closed loop system controlled by the MIMO LSDP controller is given in Table 5.8. As noted earlier in section 4.6.3.6, the co-prime stability margin provide the limit of co-prime factor uncertainty that the system can accommodate, where the co-prime factor uncertainty is the distance measure in gap metric criterion from the nominal model to the uncertain model.

As observed in Table 5.8, values of above 0.25 are achieved, hence the designs are considered to successful from the point of view of robustness.

Table 5.8: Coprime stability margin and classical gain & phase margins

Response Characteristics	MIMO System
Normalized Coprime Stability Margin	0.672
Classical SM Lower Limit	5.106
Classical PM Lower Limit (deg)	84.515

The closed loop stability margins of individual loops after the tuning of the PI controllers have been completed are listed out in Table 5.9.

Table 5.9: Closed loop stability margins

Response Characteristics	Roll Rate	Pitch Rate	Yaw Rate	Altitude Rate
Gain margin (dB)	11.2	16	12.1	21.7
Phase margin (Deg)	62.4	75.5	55	63

Acceptable stability margins can be observed in the loops. The values of the structured singular value calculated using the weights derived in section 5.2.4.1 and section 5.2.4.1, and plotted across the frequency spectrum, is presented in Fig. 5.13.

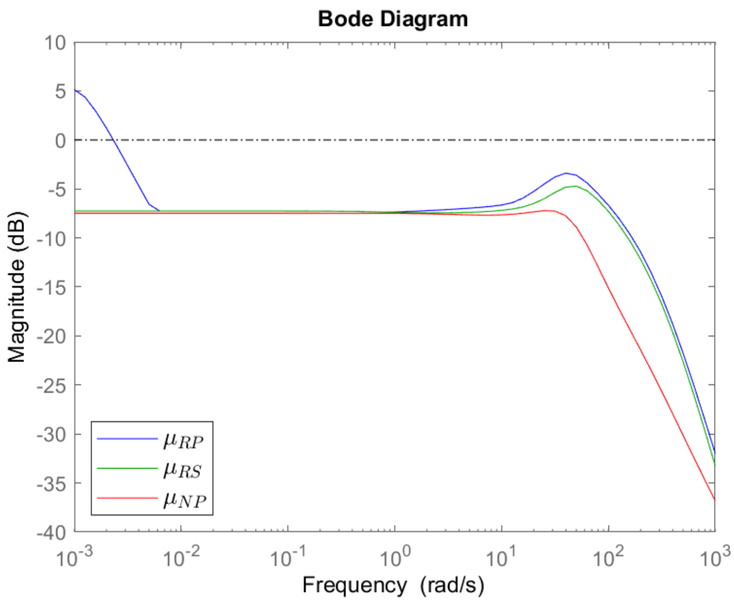


Figure 5.13: Frequency dependent μ-plots

It can be observed that the curves lie below unity except for that of robust performance, which while at low frequencies fall above unity. While LSDP controller proves effective against co-prime factor uncertainty, the controller by itself doesn't strive towards constraining the singular structured value concerning performance to below unity.

The peak values of the curves are presented in Table 5.10. As can be seen, the peak nominal and robust performance μ values of the system, as controlled by the multivariable controller falls below unity while, (as noted above) that of the robust performance falls above unity.

Table 5.10: Robust Performance Analysis

	Nominal Performance	Robust Stability	Robust Performance
μ	0.44	0.58	1.8

In the next subsection the closed loop bandwidth of the system will be examined.

5.3.3.4 Closed loop bandwidth

Using the bandwidth definitions presented in 5.2.5 the closed loop bandwidth of the system as controlled by the LSDP multivariable controller is calculated based on the curves in Fig. 5.14.

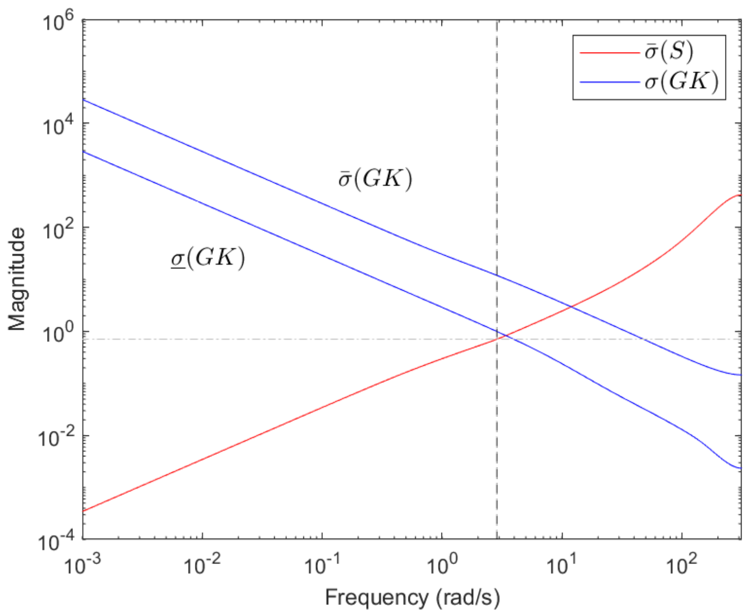


Figure 5.14: Multivariable LSDP 1-DOF system: Singular value plots of Sensitivity and Loop Function

The bandwidth, i.e. the frequency where $\bar{\sigma}(S)$ crosses 0.707 from below, calculates to 2.866 rad/s (0.456 Hz). This indicates that the quadrotor would be most effective to reference commands that have a time period of about 2.2 seconds and slower. Faster response can be expected in certain directions as we can see $\bar{\sigma}(GK)$ crosses unity at 47.91 Hz (time periods of 0.13 s and above), but overall effective time period for reference commands can be noted to be around 2 seconds.

5.3.4 Conclusion

This brings to the end of the LSDP 1 DOF implementation and formulation of the framework **Algorithm #6** (5.3.1.1), that can be used to develop optimal robust controllers alongside design weights. As presented, the hybrid LSDP 1 DOF - PI provides satisfactory performance with regards to reference tracking. The μ values similar to what we observed with the Multivariable MSO controller lie below unity for NP and RP but not for RP in lower frequencies. From calculating the bandwidths we can observe the fact that the system would respond robustly to input commands that have a time period of around 2 seconds.

The implementation of a framework, as well as the development of a multivariable controller for the 2-DOF LSDP for the quadrotor model is not presented here in the study. A framework for the 2-DOF can be based on **Algorithm #3** (4.7.1) and developed for the multivariable controller in a manner very similar to how **Algorithm #2** was extended to **Algorithm #6**. In the next section the development of the multivariable μ controller will be examined.

5.4 μ-Synthesis

In this section the design procedure for the development of the multivariable μ controller will be presented. The section is divided into two parts. In the first part the theory behind the controller development will be introduced. This is mainly an extension of the μ-controller concepts explained previously in section 4.8.

In the second part, the multivariable quadrotor μ-controller developed based on the concepts discussed will be presented. The performance of the controller will be analysed and response to step inputs will be examined.

5.4.1 Multivariable controller development framework

Similar to the SISO counter part, the MIMO μ-controller development has similarities to the multivariable MSO controller development procedure. Developing the controller depends on appropriately selecting the sensitivity and complimentary sensitivity weights. The complimentary sensitivity weight w_I is developed as explained in section 5.2.1.

The sensitivity weight is derived alongside the optimal controller which stabilizes the control problem from Fig. 4.3 while satisfying equations 4.66 and 4.67. To develop this weight alongside the controller an external framework is introduced which resolves an optimization problem while ensuring the time and frequency domain performance constraints are met. The D-K iteration algorithm which develops the μ-controller is placed inside this framework. The algorithm to develop the framework is presented in the next subsection.

5.4.2 Algorithm to Develop the Multivariable μ-Controller

Algorithm #7 can be used to develop the μ-controller in the multi-input multi-output plant.

Algorithm # 7 Sensitivity weight and Multivariable Controller design procedure for μ-synthesis

Inputs: Plant model $\mathbf{G}$, design constraints, model uncertainty estimate.
Outputs: $\mathbf{K}$, $\mathbf{W}_I$, $\mathbf{W}_P$.

1. Scale the given plant G according to the criteria described in section 3.3
2. Calculate the complementary sensitivity weight $\mathbf{W}_I$ according to the procedure described in section 5.2.1.
3. Define the appropriate form of elements of the diagonal matrix $\mathbf{W}_P$ and choose the initial value of the parameters.
4. Define the cost function $\mathcal{J}$ and initialize the weights W1 through W4.
5. Develop the vector $\mathbf{\Phi}$ which contains the following nonlinear closed loop functionals
   1. Maximum overshoot, M_p
   2. Rise time, t_p
   3. Settling time, t_s
   4. Maximum value of input, u_{max}
   5. μ(N)

   Here M_p, t_p and t_s, the step response characteristics and u_{max} are the time domain specifications while μ(N) refers to the frequency domain specification. To constrain the closed loop functionals calculated for every output, they are stacked together into a single vector.
6. Define the frequency domain bounds ϵ_μ and time domain specifications limit vector $\boldsymbol{\epsilon}$.

7. Implement the constrained non-linear optimization problem. This is achieved using the OPTI MATLAB toolbox. The black box optimizer NOMAD is chosen as the optimization routine. Constraints are defined based on ϵ_μ and $\boldsymbol{\epsilon}$ for the corresponding elements of vector $\mathbf{\Phi}$, that defines the closed loop functionals.
8. Depending on robustness requirements modify the weights $W1$ through $W4$ (in the cost function $\mathcal{J}$) to reach the required performance levels.
9. If satisfactory performance specifications are not met, decrease the % uncertainty considered while defining $\mathbf{W}_I$ and relax the bounds defined in step 6 and retry step 7.

Comparing **Algorithm #7** with **Algorithm #4**, many similarities can be noted, specially with regards to how the overall optimization problem is formulated. The differences mainly come from the weights $\mathbf{W}_I$ and $\mathbf{W}_P$, the structure of $\mathbf{\Phi}$, the calculation of the cost function and the consideration of outputs from the system. For further details concerning other aspects of the algorithm see section 4.8.2.

The performance analysis criteria defined in section 4.8.3 for the SISO system, can be directly applied to the MIMO case and hence is not be restated. It can be noted that $\mathbf{N}_{11}$, as well as other components of $\mathbf{N}$ in eq. (4.68) for the MIMO case, would be matrices as opposed to scalars in the SISO case.

This brings us to the end of the controller development description. In the following section these developed concepts will be applied to the multivariable quadrotor model.

5.4.3 Multivariable μ-controller for the Quadrotor Application

The tracking attitude-altitude controller developed based on concepts presented in the previous section, will be analyzed and responses of the plant to specific input signals presented, in this section. The complimentary sensitivity weight, the sensitivity weight, the closed loop bandwidth, and the time and frequency domain characteristics will be analyzed.

5.4.3.1 Complimentary sensitivity weight

While developing the complimentary sensitivity weight, the uncertainty values presented in Table 5.2 are considered to begin with. As listed in step 9 of **Algorithm #7**, as uncertainty is decreased until the constraints given in section 3.5 are fully met. The parameter uncertainty range from Table remain the same, except for mass and thrust coefficient c_t. The uncertainty of both these parameters are reduced to zero in order to satisfy the constraints.

Uncertainty in quadrotor parameter values, w.r.t to time delay and uncertainty due to linearization are absorbed by the complimentary sensitivity weights. Considering multiplicative input uncertainty and the uncertain plant $\mathbf{G}_p = \mathbf{G}(\mathbf{I} + \mathbf{W}_I\mathbf{\Delta})$, the following complimentary sensitivity weight is developed.

$$\mathbf{W}_I = \begin{bmatrix} \frac{0.9859s+15.15}{s+56.37} & 0 & 0 & 0 \\ 0 & \frac{0.985s+11.77}{s+53.16} & 0 & 0 \\ 0 & 0 & \frac{0.984s+15.07}{s+56.34} & 0 \\ 0 & 0 & 0 & \frac{0.9851s+12.05}{s+53.45} \end{bmatrix}$$

Having developed $\mathbf{W}_I$ the next step is to develop the sensitivity weight and the controller. **Algorithm #7** will be used for this purpose.

5.4.3.2 Optimization problem, the sensitivity weight and the μ-controller

The framework of the optimization problem remains the same (see section 4.4.6.2), and the additional changes to this optimization problem w.r.t developing the μ-controller comes from section 4.8.4.2.

The sensitivity weight that is developed by following **Algorithm #7** is given below:

$$\mathbf{W}_P = \begin{bmatrix} \frac{-6.482s-1671}{s+100000} & 0 & 0 & 0 \\ 0 & \frac{53.77s-1874}{s+99960} & 0 & 0 \\ 0 & 0 & \frac{0.148s+3144}{s+21150} & 0 \\ 0 & 0 & 0 & \frac{-0.01026s-36.51}{s+225.1} \end{bmatrix}$$

The controller itself is presented in Appendix D.2. In the next section the time and frequency domain characteristics of the non-linear closed loop system will be presented.

5.4.3.3 Closed loop characteristics

The time and frequency domain characteristics of the closed loop plant is analysed in this section. In Table 5.11 the time domain characteristic of the closed loop nonlinear plant is presented. The plant is subjected to step inputs (see section 4.4.6.6) and responses calculated.

Table 5.11: Closed loop time domain specifications after controller tuning

Response Characteristics	Roll Rate	Pitch Rate	Yaw Rate	Altitude Rate
Rise Time (s)	0.464	0.453	0.515	0.634
Settling Time (s)	0.646	0.631	0.669	0.855
Peak	0.249	0.248	0.501	1.464
Peak Time (s)	0.990	0.990	0.990	0.990
% Overshoot	0.000	0.000	0.189	0.000
Undershoot	0.000	0.000	0.000	0.144

As observed acceptable responses in terms of the meeting the design constraints, have been achieved.

The actual response of the system to these step input are presented in Figures 5.15 and 5.16. In Fig. 5.15, the red curve represents the response of the nonlinear model and the orange curve represents that response in the presence of a time delay equal to the duration of three sample times, alongside the presence of a zero-mean noise signal, with a standard deviation equal to 50% of the maximum reference value, injected at the output of the plant.

As can been seen from Fig. 5.15, the μ-controller offers excellent noise and time delay rejection capabilities. Further experiments with noise signals with a higher % standard deviation, showed that the controller can tolerate noise signals with a standard deviation upto around 70% of the reference value and a delay of three sample times.

(Note: In subsequent simulations it was identified that in case the it requires that the system needs to exhibit a higher rise time and faster settling times, it could be achieved by introducing a cascaded proportional controller before the μ-controller. While this hybrid controller can speed up the closed loop performance, it comes at the cost of the presences of oscillations in the signal. This isn't unexpected as the higher bandwidth offered by the hybrid controller lets through a portion of the high frequency noise signal into the system.

Hence while introducing the cascading proportional controller might improve the performance (in terms of desired closed loop characteristics) of uncertain plant models, this comes at the cost of decreased noise rejection capabilities. The control engineer will need to find the balance between uncertain plant performance and noise rejection capabilities as required, while tuning the proportional controller.)

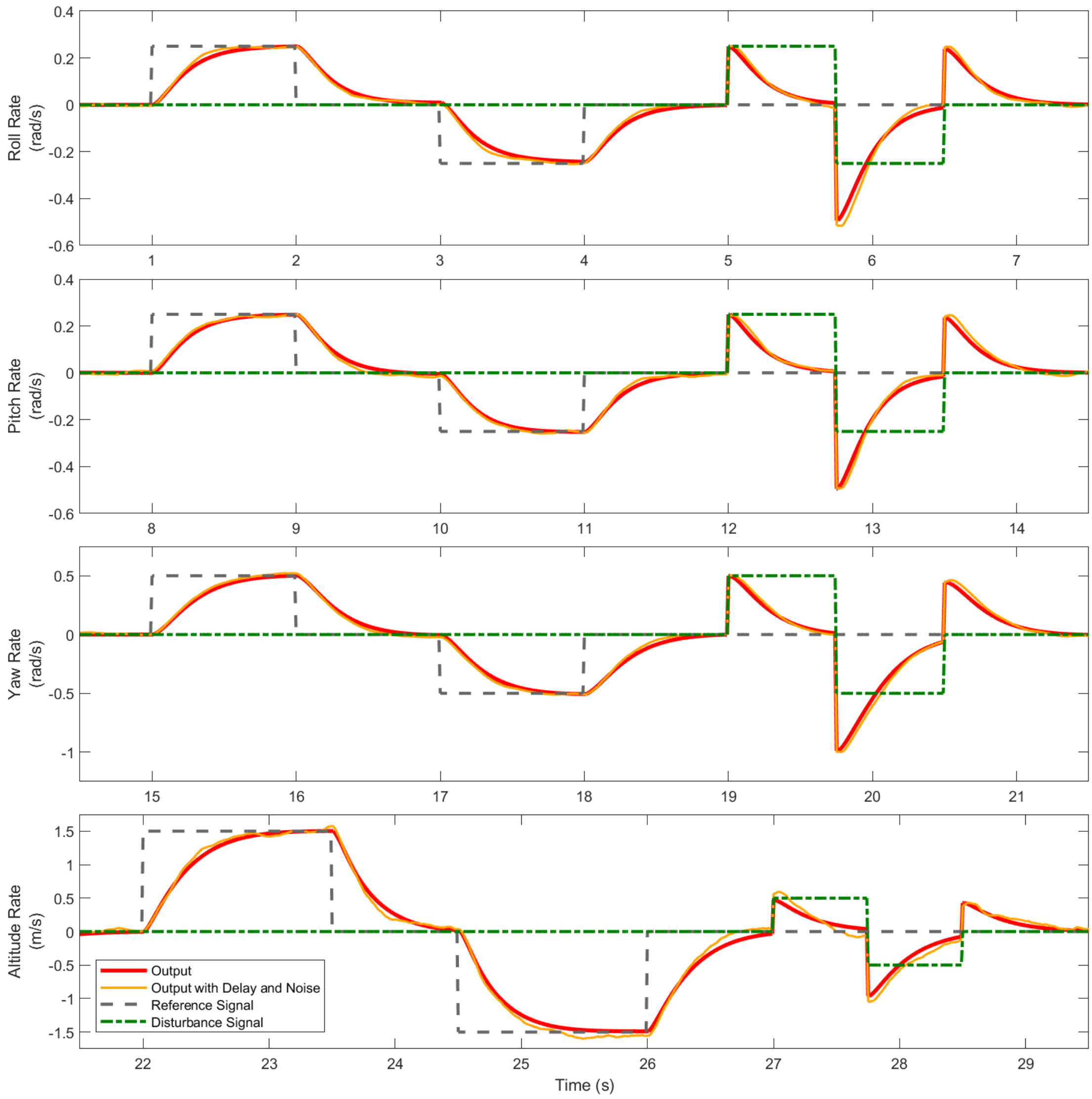


Figure 5.15: Step response of the multivariable μ-controller based system

Fig. 5.16 presents the responses of the motors to the reference input command. The dark and light blue curves represents the responses from the systems without and with the noise/time delay injection. The motor responses are satisfactory with no oscillations or overshoots.

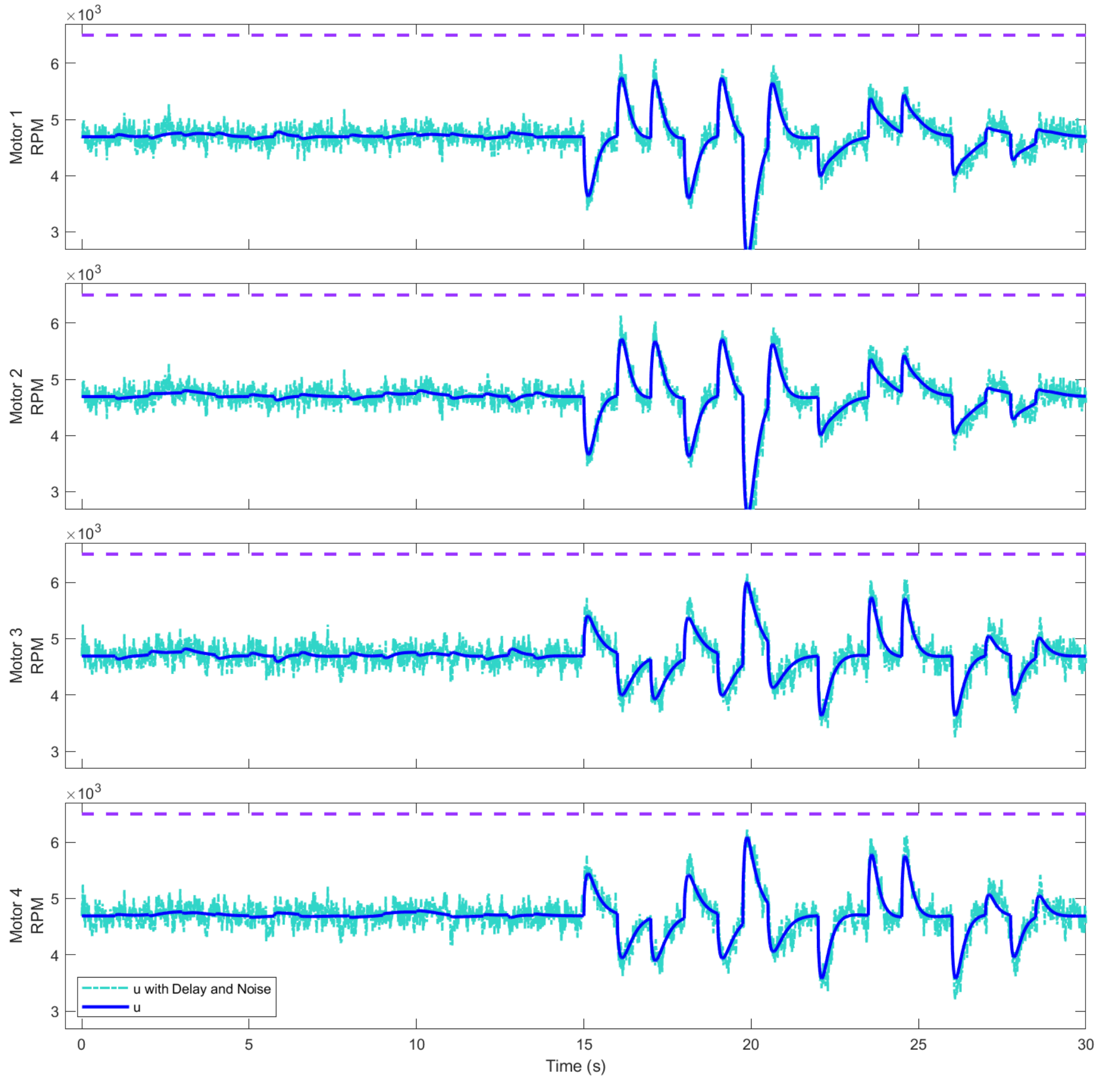


Figure 5.16: Forces exerted by individual motors in response to reference signal

The performance of the uncertain plant models, that is, those plants with the uncertainty in parameters are examined next. The uncertainty source ranges are those present in Table 5.12, and the plant subjected to the complex manoeuvre of section 4.4.6.10.

Table 5.12: Uncertainty Sources

Parameters	% Uncertainty	Range
I_{XX}, I_{YY}	0.04339 $\pm$30% $\mathrm{kg\,m^2}$	$0.0304 \leq I_{XX}, I_{YY} \leq 0.0564$
I_{ZZ}	0.0705 $\pm$30% $\mathrm{kg\,m^2}$	$0.0493 \leq I_{ZZ} \leq 0.0916$
dr	0.194$\pm$30% m	$0.1358 \leq d_r \leq 0.2522$
c_p	0.045 $\pm$30%	$0.0315 \leq c_p \leq 0.0585$

As observed from Fig.5.17 oscillations are absent throughout the responses. The response are also characterised by the absence of overshoots. Pitch and Roll response fail to the reach the peak reference input set point, while altitude rate response can be observed to have recorded the smallest spread.

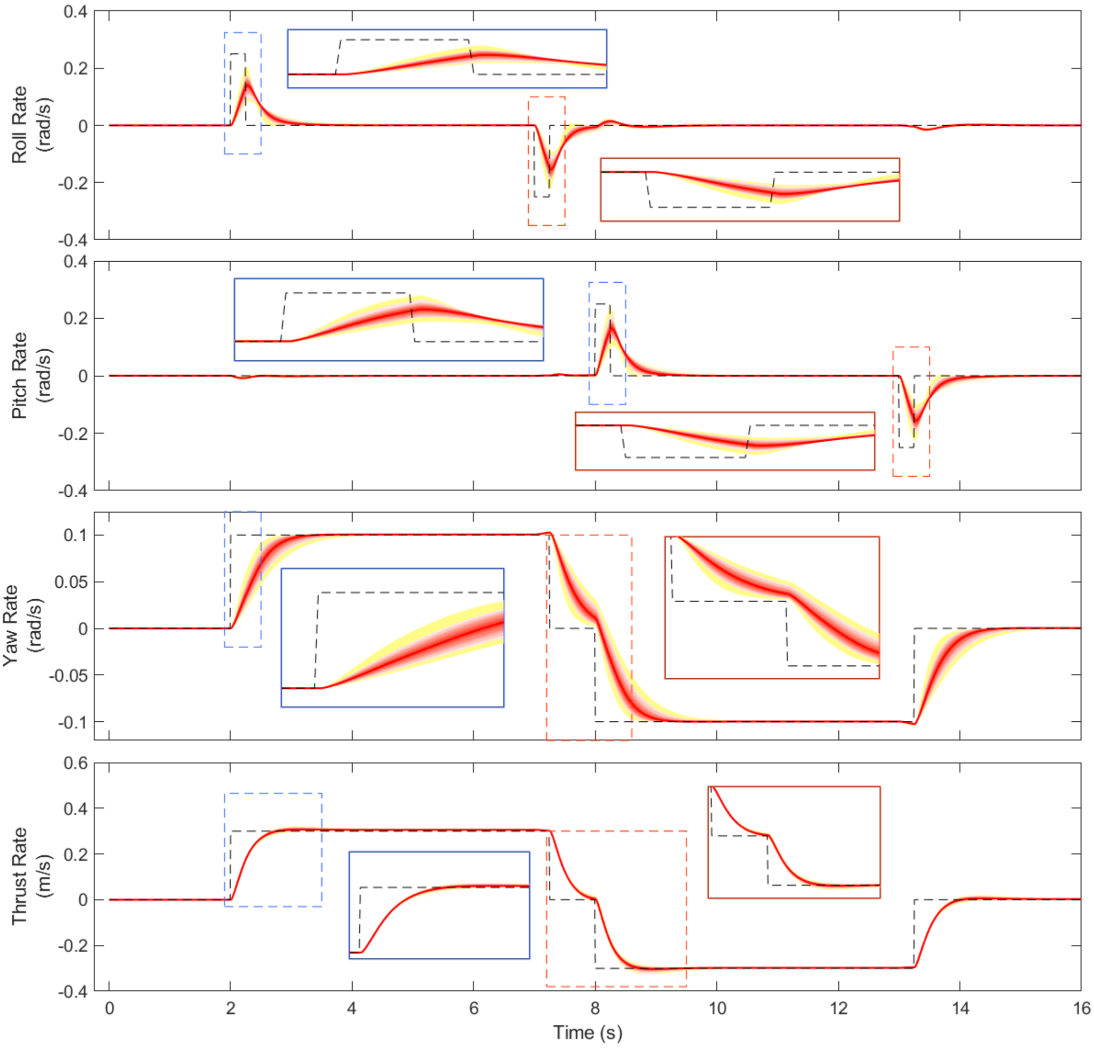


Figure 5.17: The response of the Multivariable μ-controller based plant with model uncertainty to the complex manoeuvre reference command.

The inputs associated with the uncertain plants are presented in Figure 5.18. Similar to the the plant response (observed in Fig. 5.17), the oscillations are absent in the associated inputs. The inputs also do not reach the point of saturation (6500 RPM).

(Note: The input plots are devoid of the characteristic spread observed in those plots from previous control designs. This is due to the absence of uncertainty in mass considered during the simulations.)

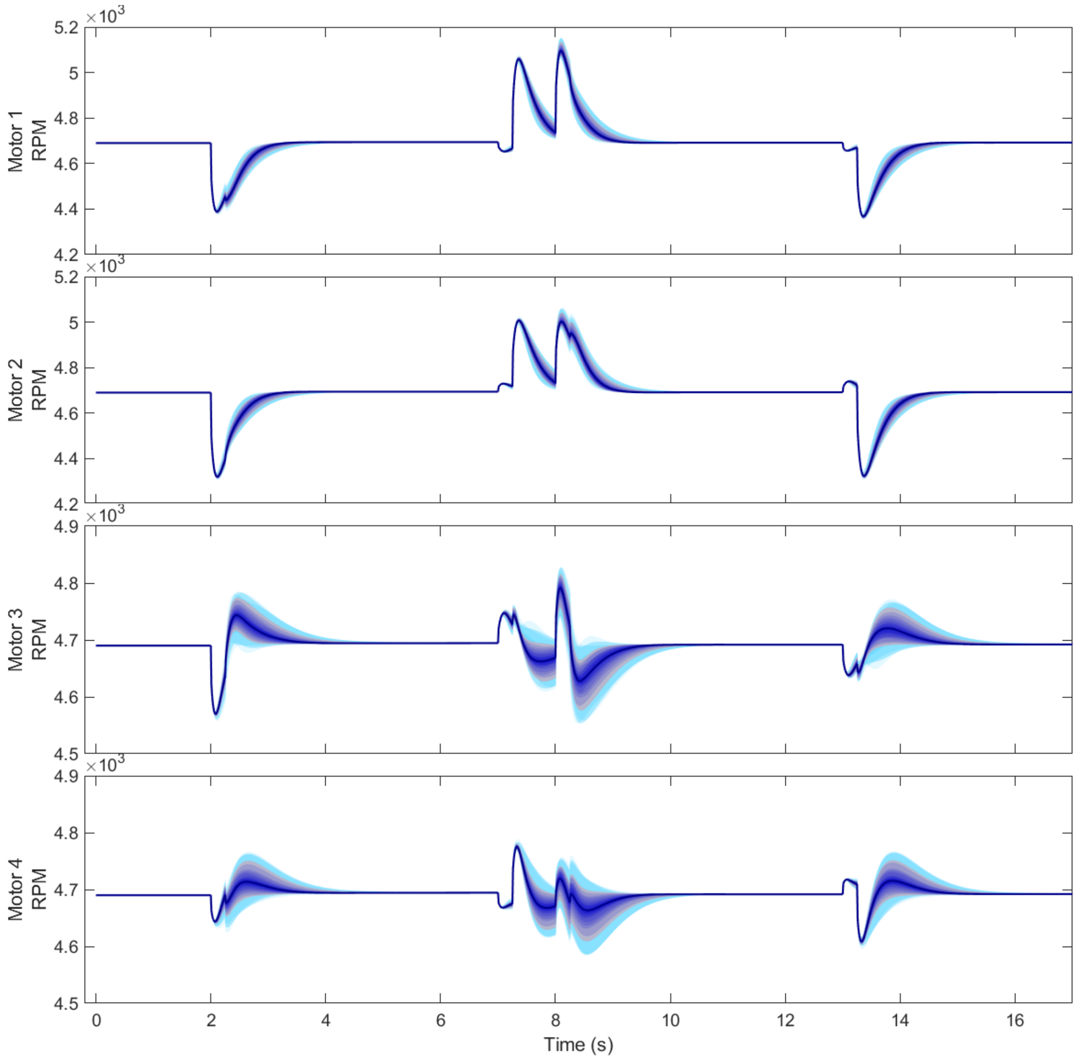


Figure 5.18: Multivariable μ-controller controlled plant rotor responses to the complex manoeuvre reference signal.

In the next section the performance of the plant with respect to the criteria mentioned earlier (see end of section 5.4.2) will be examined.

5.4.3.4 Performance

In this section the performance of the plant will be analysed. The values of μ calculated based on the weights presented in section 5.4.3.1 and section 5.4.3.2, and based on a $\mathbf{\Delta}$ allowed to take both real and complex values, plotted across the frequency spectrum is presented in Fig. 5.19.

Compared to the corresponding plots of multivariable MSO and LSDP (Figures 5.7 and 5.13 respectively), it can be observed that the controller brings the values of μ to below unity.

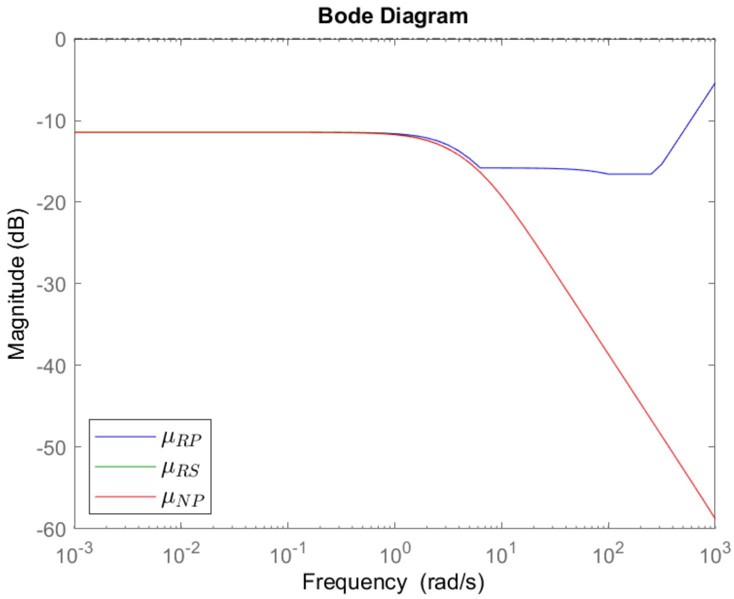


Figure 5.19: Frequency dependent μ-plots

The peak values of the curves are presented Table 5.13.

Table 5.13: Robust Performance Analysis

	NP	RS	RP
μ	0.27	0.27	0.54

As explained in section 4.8, the controller attempts to bring the μ values to below unity inorder to ensure robust performance and stability of the plant in the presence of uncertainty. The D-K performance routine within the optimization framework, in this case has successfully achieved this task.

Using the closed loop bandwidth definitions from section 5.2.5, the bandwidth of the system as controlled by the μ-controller, is calculated based on the plots in Fig. 5.20. The bandwidth calculates to 1.86 rad/s (0.296 Hz). This would mean that the controller, despite model uncertainty will be able to effectively follow reference commands which have a time period of 3.4 seconds or higher. While this might seem slow, the controller this time period can guarantee robust performance against uncertainties.

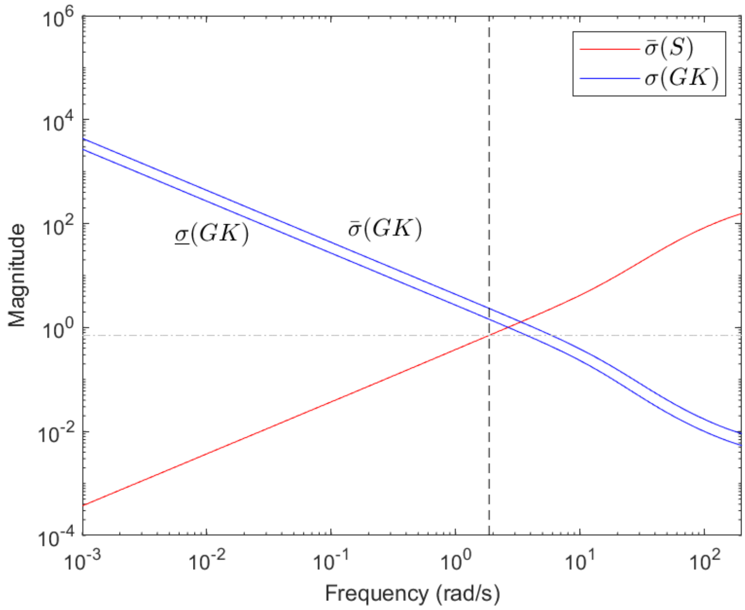


Figure 5.20: Multivariable μ-controller system: Singular value plots of Sensitivity and Loop Function

It can be observed that $\bar{\sigma}(GK)$ crosses unity at 4.25 rad/s (0.676 Hz), corresponding to a a reference command time period of 1.5 seconds. Hence faster response can be expected in certain directions, while an overall effective time period for reference command can be expected to be around 3.4 seconds.

5.4.4 The μ-synthesis controller development - Conclusion

This concludes the development of the μ controller. The framework to generate the controller and the design weights are explained through **Algorithm #7** (5.4.2). As can be seen from 5.20, while the values of μ fall below unity for NP, RS and RP this comes at the expense of developing a complimentary sensitivity weight that do not consider uncertainty in the mass and thrust coefficient. In light of the performance analysis done on MSO and LSDP controllers in the previous sections we can assume that uncertainty in mass, and thrust coefficient of the craft restricts the overall system from achieving robust performance i.e., to limiting μ_{RP} below unity.

The μ-controller achieves a overall system bandwidth of 1.84 rad/s which translates to an effective response to signal with a time period of around 3.4 seconds. This once again points to the conservative nature of the controller owing to the fact that in the presence of the considered uncertainties it guarantees complete robustness of the plant.

(Note: The uncertainty in mass and thrust coefficient can be included if need be, but that would result in a $\mu_{RP} \geq 1$. This is a characteristic of the problem which we are trying to solve. There are other studies that have come across the same problem. Consider the study [200], where a μ-controller is developed to control an X-fighter plane with mass uncertainty of $\pm10\%$. The developed controller while it was able to provide stability, it could constraint the value of μ to only about 1.87. This meant the controller will not be able to guarantee *robust performance*.

This reminds us that, sometimes the nature of the problem physically poses restrictions on the performance, that the controller can achieve in the presence of model uncertainty. The following two points can be inferred from the work presented on μ-controller development using the proposed framework.

1. Theoretically it is possible to isolate those parameter uncertainties that can contribute significantly towards performance deterioration of the plant.

2. Optimal robust controllers can be developed in spite of constraints imposed by the physical characteristics and model uncertainties of the plant, using the proposed framework.

Depending on requirements, i.e, say if it is okay for the value of mu to be ≤ 2 as opposed to $\leq$1, as many a time performance requirements in practical problems may not be as stringent as it had been in this study, the proposed framework would act as excellent tool to develop robust optimal controllers.)

5.5 Performance comparison of the developed robust controllers

In this section the performance of the 3 multivariable robust controllers, developed in the previous sections are compared. The comparisons are performed in a manner similar to those presented in section 4.9. Plants with parameter uncertainty are developed using the uncertainty description from Table 5.12. The section is divided into 4 parts. In the first part, various aspects of the time domain response of the plants for each of the different systems are compared. The response of the uncertain plants alongside the nominal plant is presented in each case.

In the second part, the closed loop bandwidth is defined in terms singular values of the sensitivity function ($\bar{\sigma}(S)$) and the loop function ($\bar{\sigma}(GK)$ and $\underline{\sigma}(GK)$). Similar to the first part, the band widths of the uncertain and the nominal plant are presented. In the third section, a comparison is made based on values of μ calculated for nominal performance, robust stability and robust performance explained in the end of section 5.4.2 and in the final section the conclusions are presented based on the observations made.

5.5.1 Time domain step response characteristics

Figures 5.21 to 5.24 represents the step response characteristics. The characteristics are based on the step response signal from 5.15 with a minor change, being the altitude pulse signal having a time period of two seconds (as opposed to a single second). This change allows to record the settling times more accurately. Alongside the settling time, the % overshoots and rise times are examined as well. Details regarding the Violin plots and the data points are presented in section 4.9.1. Similarly each of the Violin plots are inspected for the four aspects presented in section 4.9.1. These are given below.

1. Position of median value of the response characteristic with respect to the patched grey area in the plot.
2. Position of the violin plot with respect to the patched grey area.
3. The spread of the data points and the shape of the probability density distribution.
4. The nature of individual violin plots as compared with the other two plots for a characteristic.

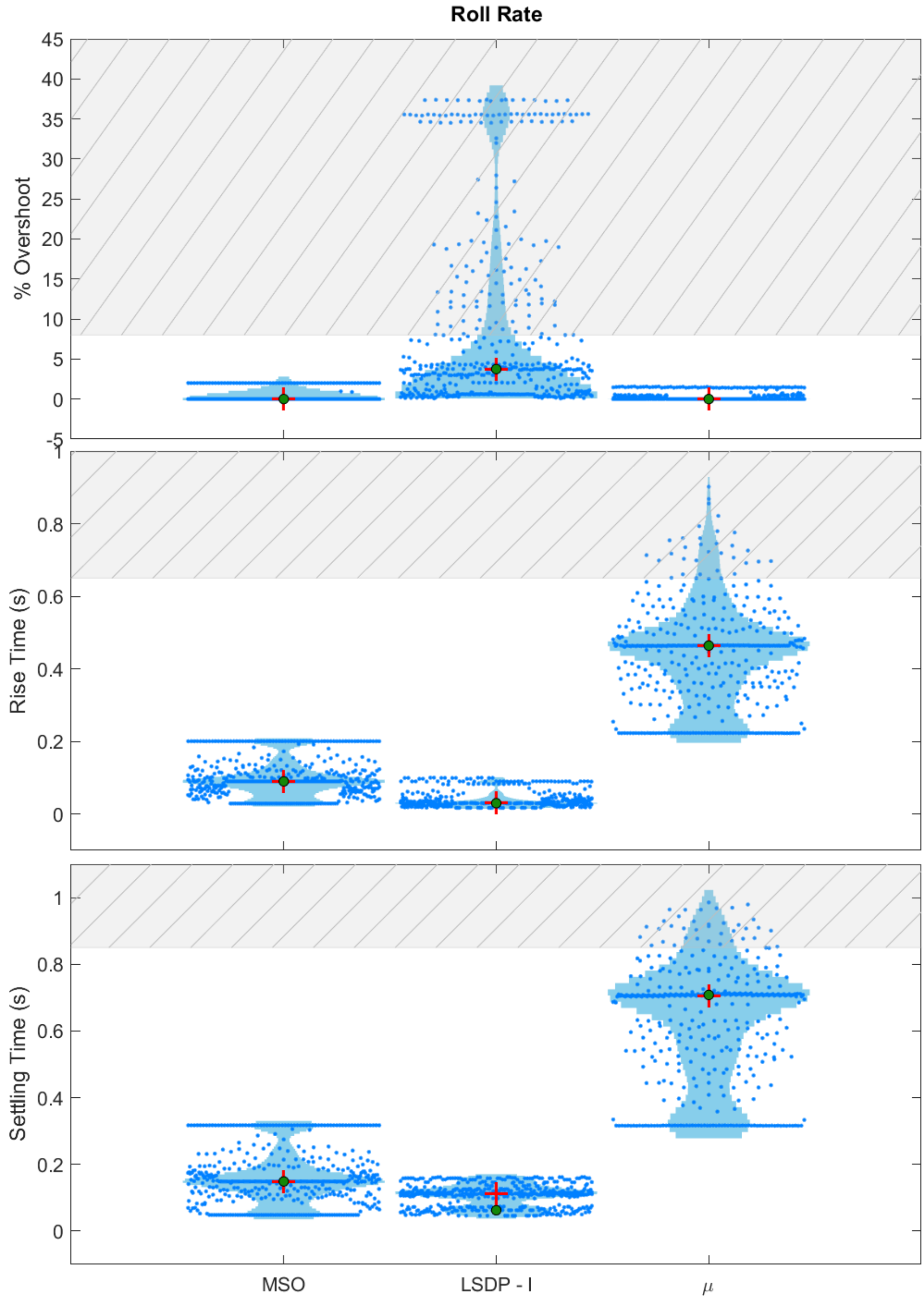


Figure 5.21: Multivariable system roll rate- time domain response comparison: The markers displayed have the following meanings - • : The response of the systems with model uncertainty, + : Median value of the respective response characteristic and ● : The value of the response characteristic of the nominal plant.

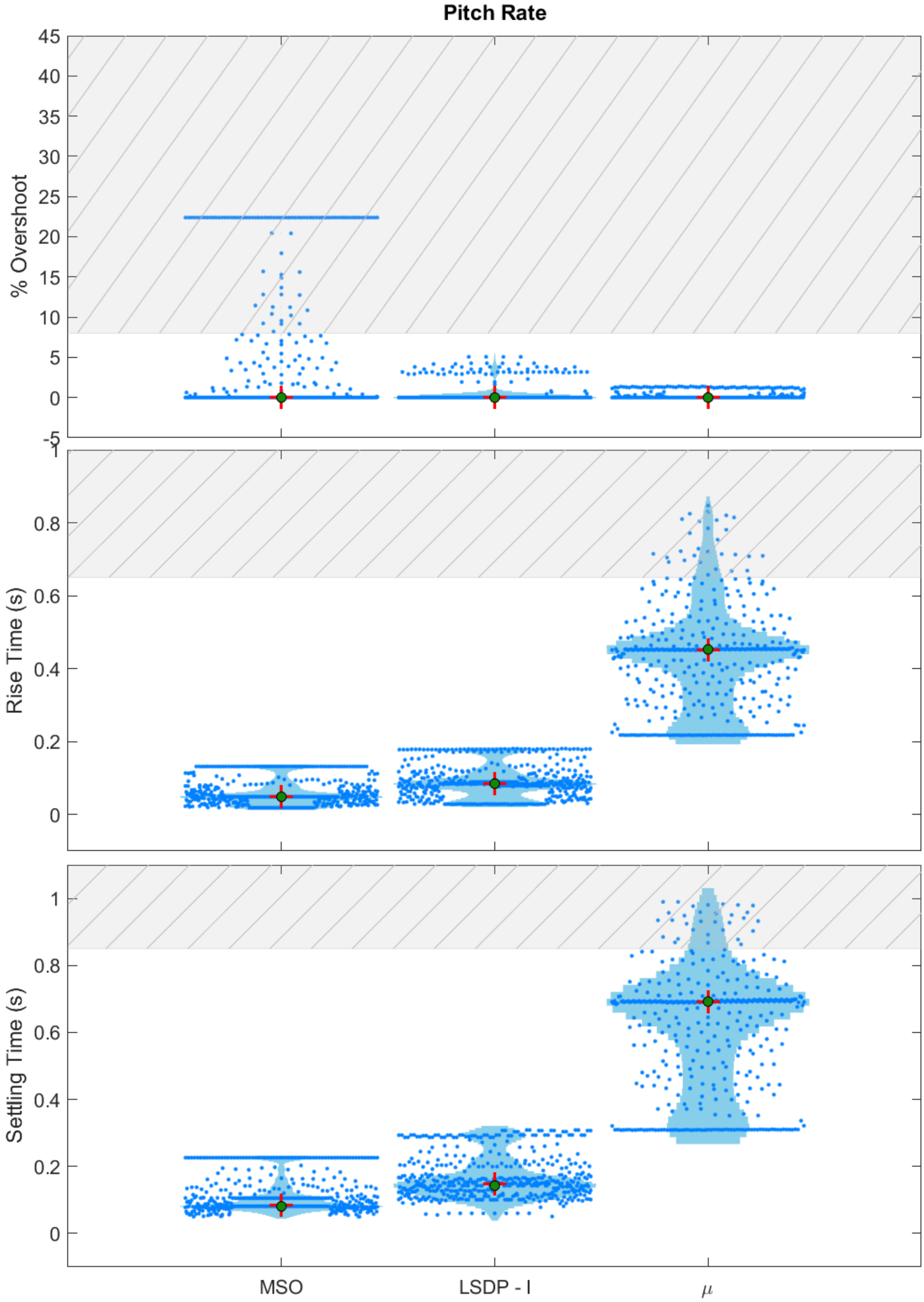


Figure 5.22: Multivariable system pitch rate- time domain response comparison: The markers displayed have the following meanings - • : The response of the systems with model uncertainty, + : Median value of the respective response characteristic and ● : The value of the response characteristic of the nominal plant.

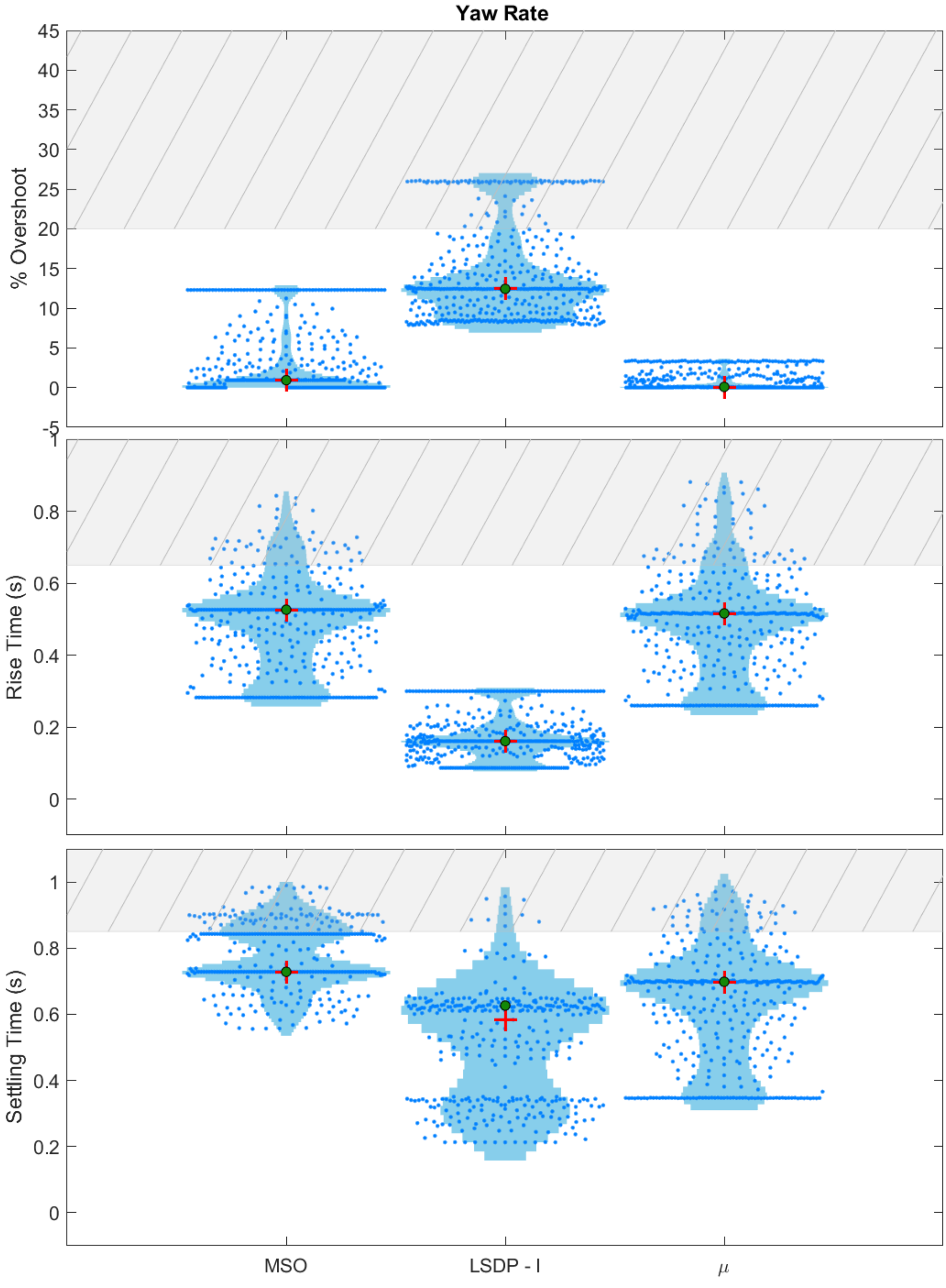


Figure 5.23: Multivariable system yaw rate- time domain response comparison: The markers displayed have the following meanings - • : The response of the systems with model uncertainty, + : Median value of the respective response characteristic and ● : The value of the response characteristic of the nominal plant.

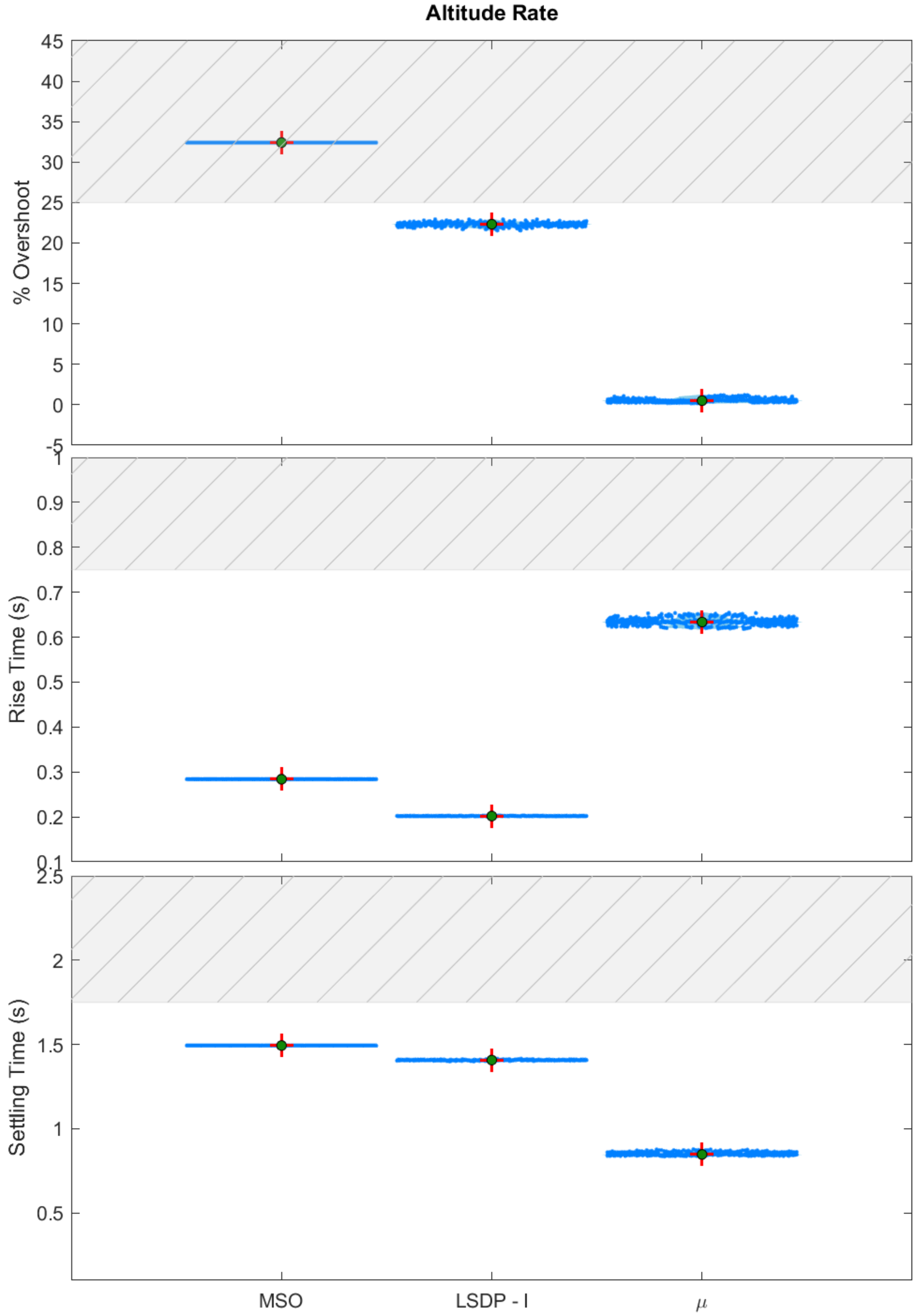


Figure 5.24: Multivariable system altitude rate- time domain response comparison: The markers displayed have the following meanings - • : The response of the systems with model uncertainty, + : Median value of the respective response characteristic and ● : The value of the response characteristic of the nominal plant.

Inspecting the Figures from 5.21 to 5.24, the following points are observed.

1. MSO controlled plants shows acceptable performance when to comes to the roll rate loop. A portion of the extreme plants in the case of the pitch rate overshoots fall in the grey patched zone. Similarly for the yaw rate rise and settling times, a portion of the extreme plant responses fall in the grey patched zone.

   The altitude rate plot, shows that the overshoots responses of all systems controlled by the MSO controller fall in the unacceptable areas. This as explained earlier (see section section 5.2.4.3) comes from the relaxation of constraints performed in order to bring more integral action into the closed loop for the altitude rate response of the plants with model uncertainty.

2. The LSDP 1 DOF controlled plants in general give acceptable response characteristics with a few exceptions. The % overshoots of some of the extreme plants for the roll and yaw rate, and the settling times of few plants for the yaw rate fall in the grey patched area. The cascaded PI controller do contribute to the higher overshoots in the altitude rate although it still falls in the acceptable region.

3. With the μ-controllers, the system shows acceptable the roll, pitch and yaw rate characteristics expect for the few extreme plants, for which the rise times and the settling times fall in the grey patched zone. Compared to the MSO and LSDO controller systems the μ controlled systems shows the largest spread in terms of the area of the probability density indicator.

While all the three controllers contribute towards the robustness of the system certain distinct characteristics can be observed in the plots. The LSDP controlled systems are characterised by faster rise times, indicating a susceptibility towards noise signals (owing to larger bandwidth). Yaw rates are characterised by a largest probability density spread while altitude rates have the least spread. The μ-controlled systems are characterised by the slowest rise times and the smallest overshoots. This points towards the relative conservative nature of the μ-controllers and the faster LSDP systems with MSO taking a place between both. Details regarding bandwidth of the systems are examined next.

5.5.2 System Bandwidths

In this section the effect of model uncertainty on the bandwidth of the multivariable system is examined. The bandwidth descriptions are those explained in 5.2.5.

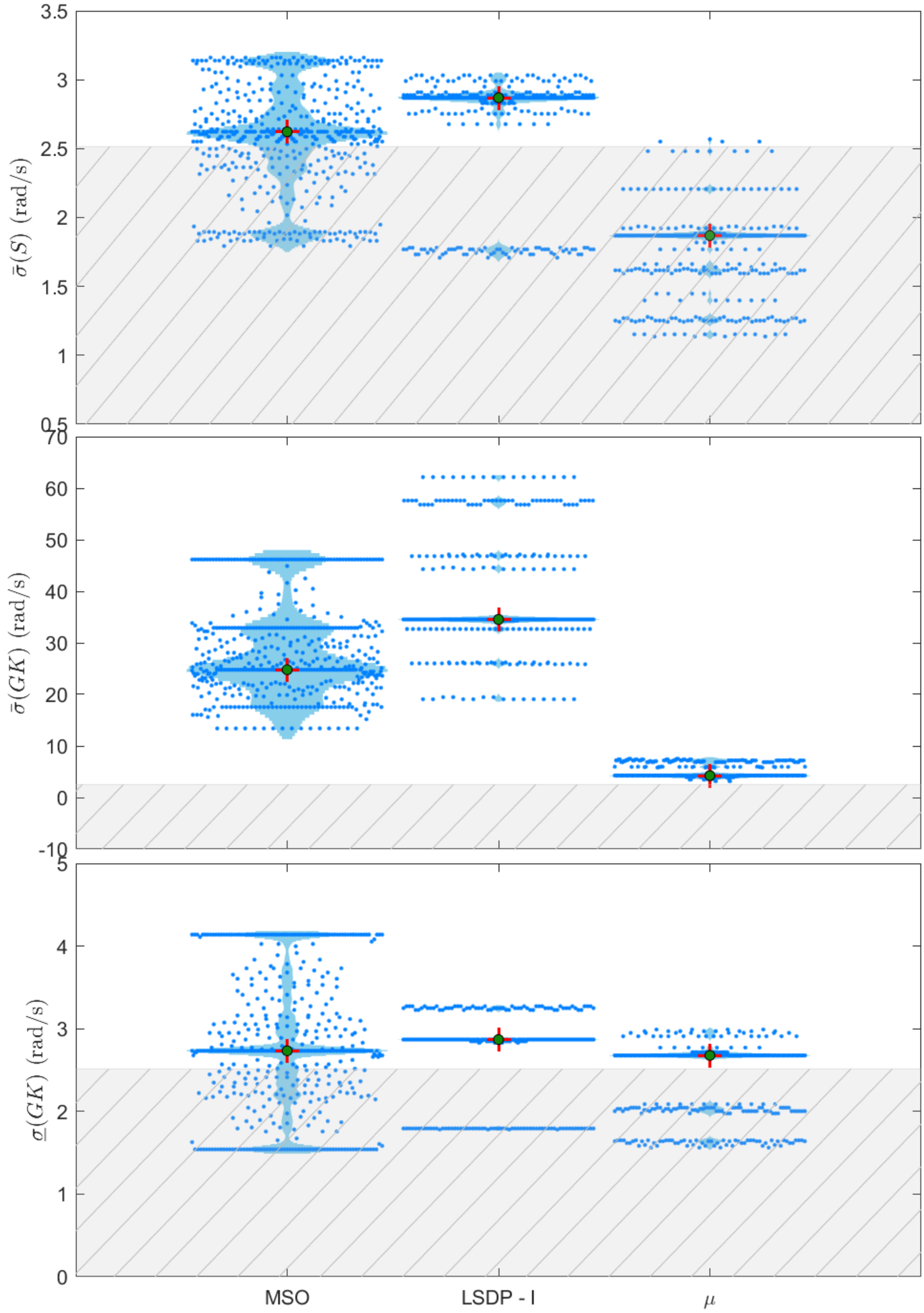


Figure 5.25: Multivariable system closed loop bandwidth comparison: The markers displayed have the following meanings - • : The bandwidth of the systems with model uncertainty, + : Median value of the respective characteristic and ● : The value of the characteristic of the nominal plant.

An overall system bandwidth of 2.5 rad/s is considered as a limit against which performances are compared. From examining the system bandwidth and magnitudes of the maximum and minimum singular values of the loop function the following observations are made:

1. MSO controlled systems the overall system bandwidth for the uncertain plots fall between approximately 1.8 and 3.2 rad/s. The MSO plots are characterized by the largest spread among the three controllers. For the loop function in the direction of $\bar{\sigma}(GK)$, the magnitude of all of the uncertain plants fall above 10 rad/s.

2. LSDP controlled systems offers the most acceptable bandwidth among the three controllers. This is considering both the overall system bandwidth as well as the loop function singular value magnitudes. The system bandwidth for the uncertain plants fall between 1.4 and 3.1 rad/s approximately.

3. Systems controlled by the μ-controller are characterized by the smallest bandwidth among the three controllers. The overall system bandwidths of the uncertain plants fall between approximately 1 and 2.6 rad/s. While $\bar{\sigma}(GK)$ values fall in the acceptable region a portion of the $\underline{\sigma}(GK)$ uncertain plant values falls in the grey patched zone.

   These bandwidth estimates concur with the observations made from the time response violin plots in the previous section. The conservative nature of the μ-controller arises from the efforts of the D-K algorithm which tries to ensure the system's robust performance and stability during the controller development stage. While this is achieved, the swiftness of the system response to the reference signal is sacrificed.

Compared to the bandwidth discussion of the SISO counter parts in section 4.9.2, it is clear that the system bandwidth of the multivariable quadrotor system is significantly lesser. While the SISO controllers offer a bandwidth of above 10 rad/s for roll, pitch and yaw rate loops and above 2 rad/s for the altitude rate, for a majority of the uncertain plants, MIMO controllers are much more conservative. They tend to offer an overall system bandwidths of less than 3.5 rad/s, with higher bandwidths in certain directions. Now that the bandwidth proprieties have been examined the robustness of the different systems will be analysed with the help of μ values.

5.5.3 Analysing the system robustness

In this section the robustness of the various systems as controlled by the developed controllers are analysed by calculating the values of μ. The values are calculated based on the definitions for μ described in Table 4.25.

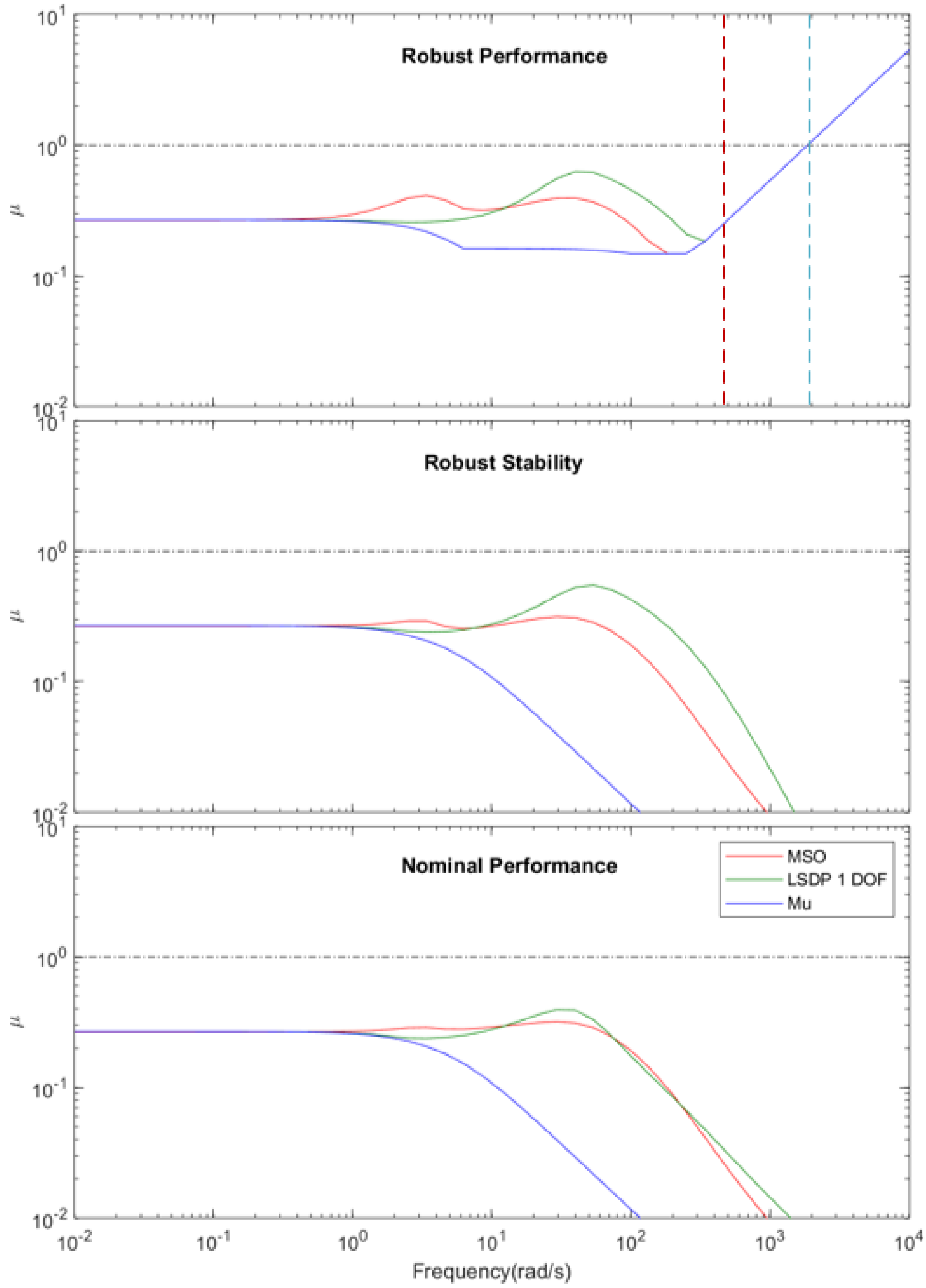


Figure 5.26: μ plotted as a function of frequency

To calculate the μ values the sensitivity and complimentary sensitivity weights developed in section 5.4.3.1 and section 5.4.3.2 are used in the system with input multiplicative uncertainty (i.e, ignoring the uncertainty in mass of the craft coefficient of thrust). The values have been calculated across the frequency spectrum and the plots are presented in Fig. 5.26. The following points can be noted from the figure.

1. The values of μ of all the three systems from steady state frequency to around a 1100 rad/s (dashed blue line in the first subplot) falls below unity for robust performance, robust stability and nominal performance indicator plots.

2. Values of μ calculated for robust performance overshoots unity at high frequencies in all the three systems, indicating the susceptibility of the system towards model uncertainty and deteriorating performance at these frequencies.

3. Considering a frequency range from steady state up to a 100 Hz, in all the three plots LSDP controlled systems exhibit the peak μ values, followed by MSO and μ controlled systems (see Table 5.14).

4. In the absence uncertainty in mass and thrust coefficient all the three controllers readily guarantee robust performance at steady state.

From the plots it is observed that the range where robust performance and stability can be guaranteed in the presence of model uncertainty, can be found out with some effort in the case of the multivariable controller design. For the case of the quadrotor, uncertainty in mass and thrust coefficient play significant roles when it comes to limiting guaranteed robust performance of the system. Efforts to decrease the model uncertainty for these two parameters would significantly make designing robust quadrotor systems easier.

The peak values of the plots are presented in Table 5.14. For the robust performance curve while as noted earlier the value of μ overshoots unity after around 1100 rad/s, the value remain below unity around system bandwidth frequencies. The peaks are calculated using those values upto around 350 rad/s (i.e. upto the critical frequency range around the system bandwidth frequency- dashed dark red line in the first subplot), hence ignoring the obvious high values at higher frequencies.

Table 5.14: Robust performance analysis - peak μ values

Systems	RP	RS	NP
LSDP 1 DOF	0.639	0.553	0.398
MSO	0.414	0.314	0.320
μ	0.340	0.269	0.267

As observed in the table, when considering robust stability, the peak μ value, that is $\|M\|_{\Delta_I}$ for the LSDP controlled plant equals 0.553. This would mean the parameter and time delay uncertainty can be increased by a factor of $1/0.553 = 1.808$ before which the worst case model uncertainty can lead to instability. The corresponding factors for MSO and μ controlled systems calculates to 3.18 and 3.71 respectively.

Similarly with regards to robust performance, the peak μ value, that is $\|N\|_{\widehat{\Delta}}$, with $\widehat{\Delta} = \left[\begin{smallmatrix} \Delta_I & 0 \\ 0 & \Delta_P \end{smallmatrix}\right]$ for the LSDP controller plant equals 0.639. Hence a parameter and time delay uncertainty can be increased by a factor of $1/0.639 = 1.565$ before significant deterioration in performance of the system can be noticed. The corresponding factors for MSO and μ controlled systems calculates to 2.415 and 2.941 respectively.

5.5.4 Performance Comparison - Conclusion

The comparison between the various developed controllers point towards certain aspects of the systems. While satisfactory robust stability margins are observed in all the three systems, the comparisons based on time domain response, system bandwidth and robustness it can be concluded that the LSDP based controllers with their fast responses, larger bandwidths, and higher stability margins can be safely chosen from between the three, when there is no uncertainty with mass and thrust coefficient of the craft or that of the payload.

The μ controllers with the smallest bandwidth and slowest in terms of favourable reference command time periods, but with the highest stability margins can be considered as the most conservative of all the three controllers. The MSO multivariable controller lies in between both these extremes. This concludes the performance comparison section and in the next part the important ideas noted in the chapter are summarized.

5.6 Chapter - Summary

The important aspects of the chapter is summarised in the below list of points. For further details refer to the corresponding sections.

- The Multivariable MSO S/T problem is implemented and **Algorithm #5** (5.2.2) which enables the controller and optimized weight development, is proposed. A multivariable controller was developed and the performance of the closed loop system examined. Satisfactory performance is observed with the developed suboptimal controller.
- LSDP 1DOF multivariable problem is defined for the quadrotor and **Algo-**

rithm #6 (5.3.1.1) is proposed. A hybrid LSDP 1-DOF - PI controller is developed for the quadrotor model and satisfactory performance is observed.

- The μ-synthesis problem is implemented for the multivariable system and **Algorithm #7** (5.4.2) is presented. A multivariable controller is developed with good robust stability margins. The μ controllers is found to be the most conservative among all the three developed controllers.

- A detailed performance comparison is conducted between the three developed controllers. While all of the controllers impart robustness to the overall system, the μ controller is found to be the most conservative and LSDP 1DOF to be the most effective when uncertainty in mass and thrust coefficient is absent.

In the next chapter a simulation based case study inspired by aspects of an industrial application is presented. Robust design strategies developed in this chapter are used to design a multivariable controller and a comparison is performed against a cascaded PID based system which are commonly used in the commercial industry.

Chapter 6

Case Study

As mentioned at the beginning of this study, (see Chapter 1), UAVs are increasingly being used in challenging environments and for resolving still harder problems. In this chapter an interesting application related to the recreational fishing industry, is discussed.

The chapter is divided into 3 sections. In the first section the industrial application is discussed and challenges observed during the flight process are listed out. In the second section a multivariable robust attitude/altitude controller and a cascaded PID trajectory tracking controller is developed for a plant model resembling aspects of the industrial scenario. Simulations are presented in this chapter for various cases and results are examined. In the final section conclusions are drawn from the case study and the case for future work is presented. The case study has been conducted using simulations, and real world experiments haven't been performed using the developed controller.

6.1 Long line fishing and the 'Aerokontiki'

Longline fishing by enthusiasts and hobbyists enabled by water borne drones and kites have been a part of recreational fishing in the past decades. The idea of drawing the line along with the baits out to the sea using a rotorcraft drone mainly evolved out of a need for increasing the reliability as well decreasing the time taken to put out the baits. While water borne drones required around 20 to 30 minutes to draw the line a kilometer out to the ocean depending on the roughness of the ocean, dropping the baits with help of a kite requires right wind directions to enable the drop.

The New Zealand based Envirobotics Limited presented their initial design ‘The Aerokontiki - Evolution I’ that involved a ‘Y-6’ configuration based drone that could lift and deliver bait payloads as heavy as 2.5 kilograms in late 2014 and from then on have graduated to newer ‘X-8’ configuration based designs that could lift and deliver up to 7 kg across the distance of a kilometer in headwinds of up to 40 km/hr [154]. The industry is rapidly growing with a user based focused mainly in New Zealand but also extended to countries ranging from the United States and to Italy. From 2017 to 2019 the author of this study had the chance to be employed in the development and testing of the drones which enabled him to have a first hand experience of working with these machines.

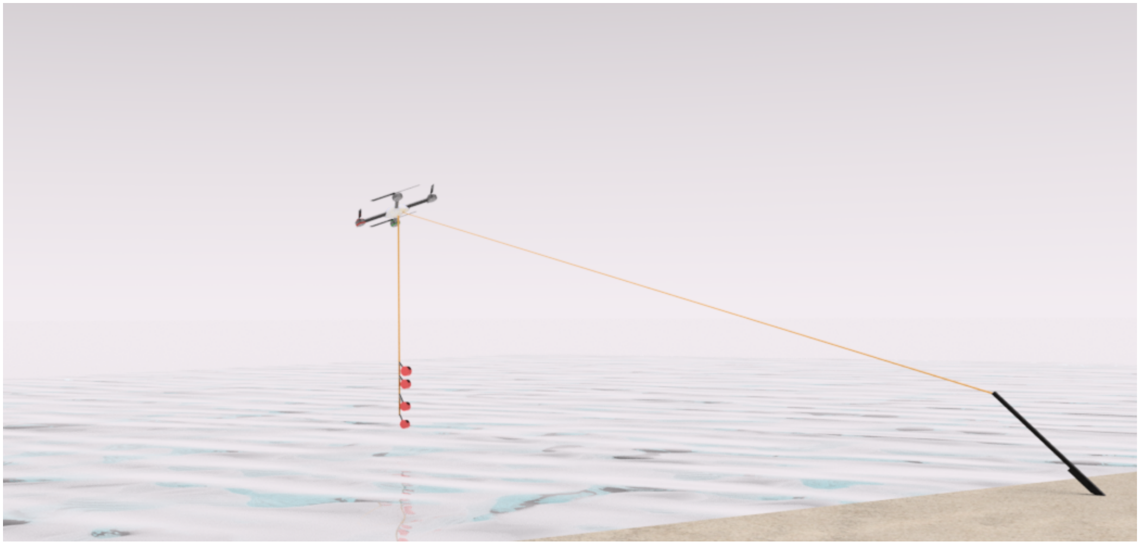


Figure 6.1: The flight of the Aerokontiki: The general flight regime is in a straight line once a height of around 60 meters is attained. The baits are suspended at a length of around 5-10 meters depending on their number. The figure is not drawn to scale.

6.1.1 The flight regime

The goal of every flight is to drop the baits at a predetermined distance from the shore line. The typical flight of an Aerokontiki, involves three main stages. These being:

1. Lift off: The craft lifts off picking up the baits rising to a height of around 60m. The baits typically number anywhere between 1 to 25 and with the leaders lines spaced at a distance of about 50cm. The bait and the leader lines are laid out on the sand or arranged using a launching trace board.
2. Flight out to the drop destination: Once the height of about 60 meters is attained, the craft flies out to the ocean usually following a straight line.

3. Bait drop and return home: This is final stage of the flight where, once the destination (usually a distance between 600 m to 1000 m from the shore line) is reached, the baits are dropped and the craft returns back.

The general set up during the initial stages of the flight of the Aerokontiki- Evolution III, the ‘X-8’ configuration based model, is presented in Figure 6.1. The practical case of the Aerokontiki makes an especially challenging control problem to resolve. Unlike traditional Micro, and Mini UAV payloads, the Aerokontiki firstly lifts a slung load, and secondly pulls the main fishing line as it flies forward. The environmental conditions, given the flight occurs at the sea in windy conditions also makes it particularly challenging.

The uncertainty in mass, occurs from the varying nature of baits and more importantly from the dips in altitude of the craft, that would sometimes result in the baits getting pulled through the water. This is in addition to the main line getting dragged through water. As the distance increases so does the friction exerted by water on the main line and to this adds additional dynamics such as flapping (oscillations) exhibited by the main line. Flapping occurs as a result of vibrations from the drone being translated to the main line, and it is seen to occur sometimes at distances above 500 m. The flapping occasionally becomes significant enough to disrupt flight and hence when this occurs the craft is brought to a stand still-hover to wait until the oscillations die out.

Figure 6.2: The Aerokontiki in action at Takerau Beach, Doubtless Bay New Zealand.

Figure 6.2 shows the Aerokontiki- Evolution II, THE ‘Y-6’ configuration based model, in action at Takerau Beach, Doubtless Bay, New Zealand.

In the next section, the effect of the slung load on the Aerokontiki flight is reproduced using experiments conducted in MATLAB. A multivariable robust attitude-altitude controller and a cascaded PID based controller for trajectory tracking is developed to control the craft. A comparison with the conventional PID based SISO controllers, based on the effectiveness of trajectory tracking is also presented.

6.2 A Quadrotor with Suspended Load

The problem of the Aerokontiki delivering bait payloads attached to the fishing main line can be approximated by modeling the problem as a quadrotor carrying a suspended load. In this section we concentrate on developing robust tracking attitude/altitude controllers for a quadrotor carrying a suspended load. The dynamics involved in pulling the main line, as it is drawn out from the reel by the Aerokontiki will not be modeled, as the design of a robust controller that can accommodate variations in system parameters, is the primary concern of this study.

Various aspects of the problem of a quadrotor carrying a slung load has been previously examined in detail in several studies which can be broadly grouped into two sections. The first section concentrates on developing optimal swing free trajectories for the flight as reported in [201–204], and the second section focuses on developing robust attitude/altitude controllers as in [205–209]. In this section since we are primarily focused on developing attitude/altitude controllers, so we will briefly examine the ideas presented in this latter group.

In [205], the quadrotor with the suspended load is established to be a deferentially-flat hybrid system (see [210] for more details) and a non-linear geometric controller to stabilize the system is presented. The closed loop system is stable and a satisfactory set point tracking is achieved. The method of separating the overall flight regime into differentially flat portions is further examined in [207], where the solution to the problem of the quadrotor with a suspended load achieving a smooth lift off is presented. The issue of payload uncertainty is examined in [206], where a Proportional Derivative controller is tuned for the quadrotor carrying the nominal payload mass and a Retrospective Cost Adaptive Controller (RCAC) is used to deal with the uncertainty in mass. Smaller settling times and overshoots are observed when the RCAC controller is compared to a fixed gain controller.

The development of an anti-swing controller is examined in [208]. While separate controllers are designed for achieving position and attitude tracking, the anti-swing mechanism is developed by modifying the reference input command using an algorithm highly dependent on the system model. In [211], an iterative LQR controller is developed to control the position and attitude of the controller. The quadrotor model is presented as a set of Differential Algebraic Equations, DAEs (a key point ignored by most other studies done in the area), although the equations are significantly simplified in subsequent analysis. Comparisons are presented with a traditional LQR and it can be observed that an iLQR gives a better performance when the integral absolute error is considered.

A simplified version of the model presented in [211] (where the DAEs are replaced after simplification with a set of ODEs), is examined in [212]. The model considered in this chapter is developed based on this model. Of all the studies summarized above, none of the studies consider model uncertainties in particular. The effects that model parameter uncertainties as well as that of time varying payload masses and time varying payload cable length, exert on the performance of the craft are omitted in these studies. This is precisely the problem that the Aerokontiki is trying to resolve. Hence in the next section, the control and response of a quadrotor with model uncertainties carrying a suspended load to complex trajectories in the presence of both time varying mass, and payload cable length, are examined.

6.3 Multivariable attitude-altitude robust controller for a Quadrotor carrying a suspended load

The equations of motion presented are based on [212] and [211]. The model presents a quadrotor that carries a slung load attached using a non-stretchable cable.

$$\begin{aligned}
\ddot{x}_i &= \frac{f_1+f_2+f_3+f_4}{m}\,(s_{\phi_i}\,s_{\psi_i} + c_{\phi_i}\,c_{\psi_i}\,s_{\theta_i}) - \frac{T}{m}\,(s_{\theta_{iL}}\,c_{\phi_{iL}})\\
\ddot{y}_i &= \frac{f_1+f_2+f_3+f_4}{m}\,(c_{\phi_i}\,s_{\psi_i}\,s_{\theta_i} - c_{\psi_i}\,s_{\phi_i}) + \frac{T}{m}\,(s_{\theta_{iL}}\,s_{\phi_{iL}})\\
\ddot{z}_i &= -g + \frac{f_1+f_2+f_3+f_4}{m}\,(c_{\phi_i}\,c_{\theta_i}) - \frac{T}{m}(c_{\phi_{iL}})\\
\ddot{\phi}_b &= \frac{1}{I_{xx}}\,((f_1-f_2-f_3+f_4)\,d_y + (I_{yy}-I_{zz})\,\dot{\theta}_b\,\dot{\psi}_b)\\
\ddot{\theta}_b &= \frac{1}{I_{yy}}\,((f_1+f_2-f_3-f_4)\,d_x + (I_{zz}-I_{xx})\,\dot{\phi}_b\,\dot{\psi}_b)\\
\ddot{\psi}_b &= \frac{1}{I_{zz}}\,(\tau_1-\tau_2+\tau_3-\tau_4\,+(I_{xx}-I_{yy})\,\dot{\phi}_b\,\dot{\theta}_b)\\
\ddot{\phi}_L &= -\frac{T}{m_L}(L\,s_{\theta_{bL}}\,c_{\phi_{bL}}) + \frac{(f_1-f_2-f_3+f_4)\,d_y}{m_L}\\
\ddot{\theta}_L &= -\frac{T}{m_L}(L\,s_{\theta_{bL}}\,s_{\phi_{bL}}) + \frac{(f_1+f_2-f_3-f_4)\,d_x}{m_L}
\end{aligned} \tag{6.1}$$

Refer to 3.2.1 for details on the variable naming conventions. The subscript L

represents the values of variable associated with the slung payload and T represents the magnitude of the tension in the cable.

The multivariable plant has four inputs which are the respective forces exerted by the propellers and 6 outputs being the attitude-altitude rates of the quadrotor as well as the roll and pitch angles of the payload with respect to the quadrotor.

The parameters of the model are those from Table 3.2 with the nominal mass of the slung load at 0.25 kg and the nominal length of the cable that attaches the load to the quadrotor at 2 meters.

The system is modeled as that containing an inner loop controller managing the attitude and altitude rate tracking and an outer loop that takes care of tracking the X/Y position coordinates. The inner loop is controlled by a MSO $\mathcal{H}_\infty$ controller and the outer loop using two sets of cascaded PID controllers. The cascaded PID controller is modeled similar to those reported in [213–215], and as presented in [216].

Here the choice for the multivariable MSO controller (as opposed to an LSDP or μ)-synthesis based controller) comes from the insights received from the performance comparison study presented in section 5.5. While the LSDP based controller was seen to be more susceptible to noise, the μ-synthesis based approach seemed to be overly conservative. The MSO based controller was observed to take a position in between the two. In this specific case, where payload mass is a significant uncertainty faced by the craft, it is crucial that the controller is able to target and absorb this uncertainty during the controller development phase. The MSO controller, while it is not too conservative and with its ability to target specific parameter uncertainties, (as opposed to LSDP which doesn't offer this option), will hence be selected for developing the controller.

In the case of the comparison study, the PID controller structure in the outer loop is kept the same while in the inner loop, the multivariable controller is replaced by SISO PID controllers. The model here remains the same while in the SISO case the inputs to the plant are grouped as explained previously in 4.2.

Three sets of experiments are conducted through simulations. In all the three sets, the quadrotor is made to trace a vertical helix stretching across a height of 200 meters, with a radius of 10 meters within a time period of 200 seconds. This trajectory is different from that taken during a usual Aerokontiki flight, although it proves ideal for testing. While the Aerokontiki trajectory generally follows a straight line, the vertical helix is slightly more challenging, and helps the engineer to test the developed controllers more thoroughly. The experiments are presented in the following sections.

6.3.1 Nominal plant response as payload mass is varied

In the first set of experiments the mass of the slung load is varied from 0.125 kg to 0.55 kg (the nominal mass being 0.25 kg), with all the other parameters held constant. The results of the simulations are presented in Figure 6.3. The same experiment is conducted on a plant where the inner and outer controllers are PID based (similar to the control strategy adopted by the industrial and hobbyist sectors [217]), the results of which are presented in Figure 6.4.

The PID controllers are tuned to provide the most robust outcomes while maintaining system stability. In both the cases, Gaussian white noise (with a noise power or height of the power spectral density equal to 10^{-6}) is injected at the attitude/altitude output which is of our primary concern. The responses are colour coded such that they follow a graduation from violet, where the payload weighs 0.125 kg, to yellow where the payload weighs 0.55 kg.

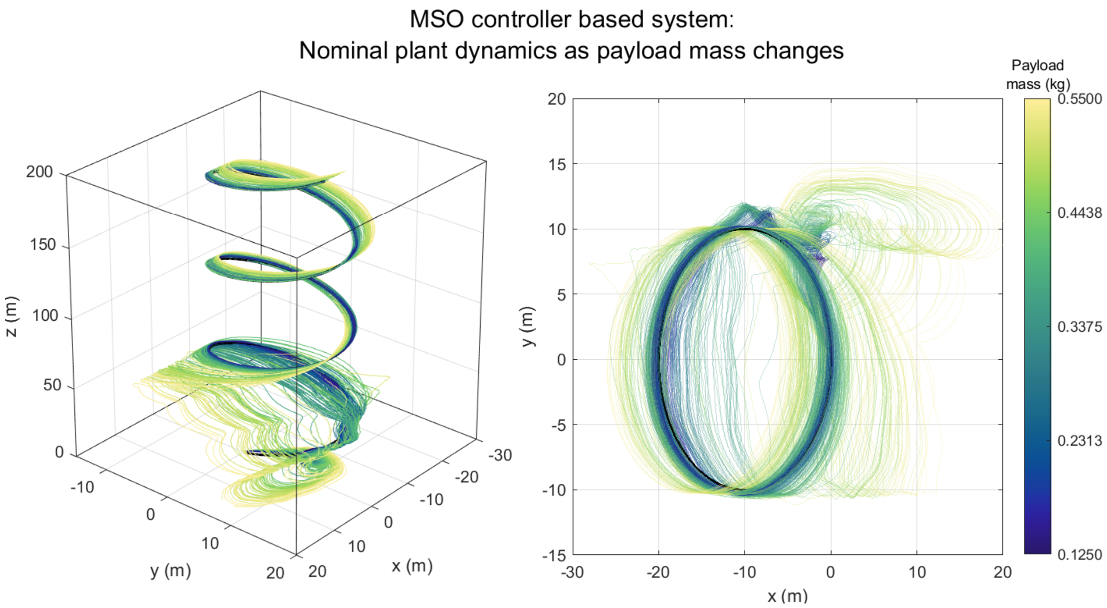


Figure 6.3: Multiple simulations showing the MSO based plant dynamics as payload mass is varied over a period of 200 seconds. The Quadrotor traces a vertical helix with a diameter of 10 meters and a height of 200 meters. The payload mass is varied from 0.125 kg to 0.55 kg (see the box on the right hand side).

The response of the MSO controller can be observed in Fig. 6.3 which can be compared to that of the PID based controller in Fig. 6.4. As can be observed in the figures, the MSO controller system performs significantly better and to an extent gives a predictable response when the mass is varied. The PID controlled system, while it gives a satisfactory response to plants with the payload mass is close the the nominal load mass of 0.25 kg, the response becomes chaotic and unpredictable as the mass is increased substantially.

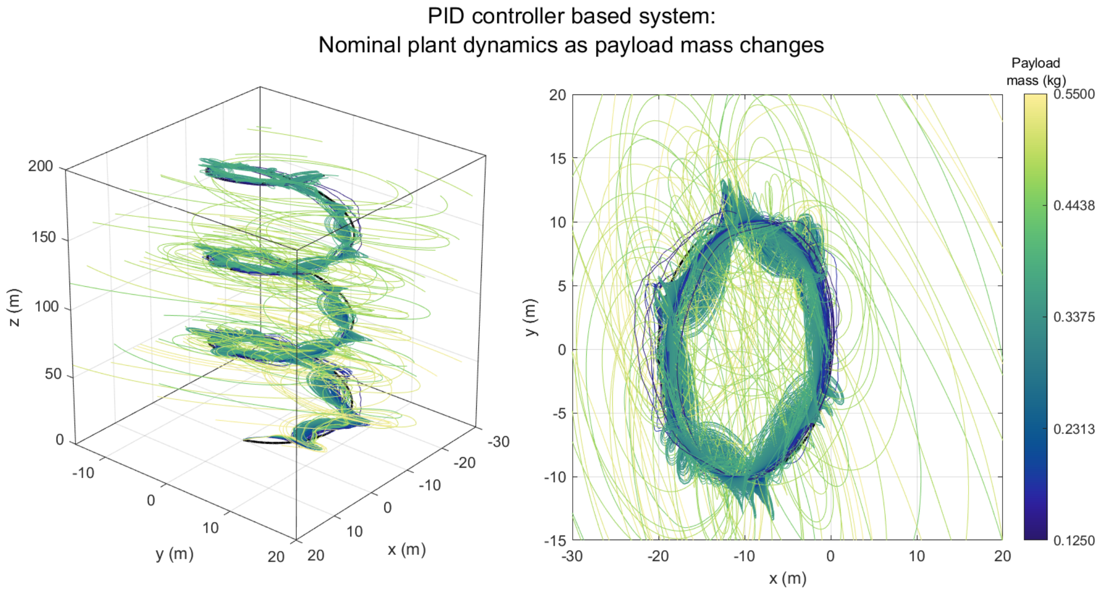


Figure 6.4: PID based plant dynamics as payload mass is varied over a period of 200 seconds. The payload mass is varied from 0.125 kg to 0.55 kg.

A point to note here is that while the PID controller performs more consistently than the MSO controller (see the dark blueish regions in Fig. 6.4) during when the load is closer to the nominal value of 0.25 for which the system was tuned, the performance deteriorates significantly as payload mass is increased. The MSO controller on the other hand with its initial offsets presents a steady but slower deterioration as mass is increased. This in short is the whole design point of robust control.

6.3.2 Plant response as the slung length of the payload is varied

In the second of experiments the length at which the slung load is attached from the quadrotor is varied from the nominal length of 2 meters to a length of 10 meters. The results are presented from Fig.6.5 to Fig.6.8. The responses are colour coded such that they follow a gradation from violet which represents a length of the attached payload of 2 meters to yellow at a length of 10 meters.

Experiments are conducted with two different sets of payload masses. In the first set the payload mass is kept at the nominal value of 0.25 kg and slung length is varied from 2 to 10 meters. The results are presented in Fig. 6.5 which shows the MSO controlled plant response, and Fig. 6.6 which gives the response of the PID controller plant.

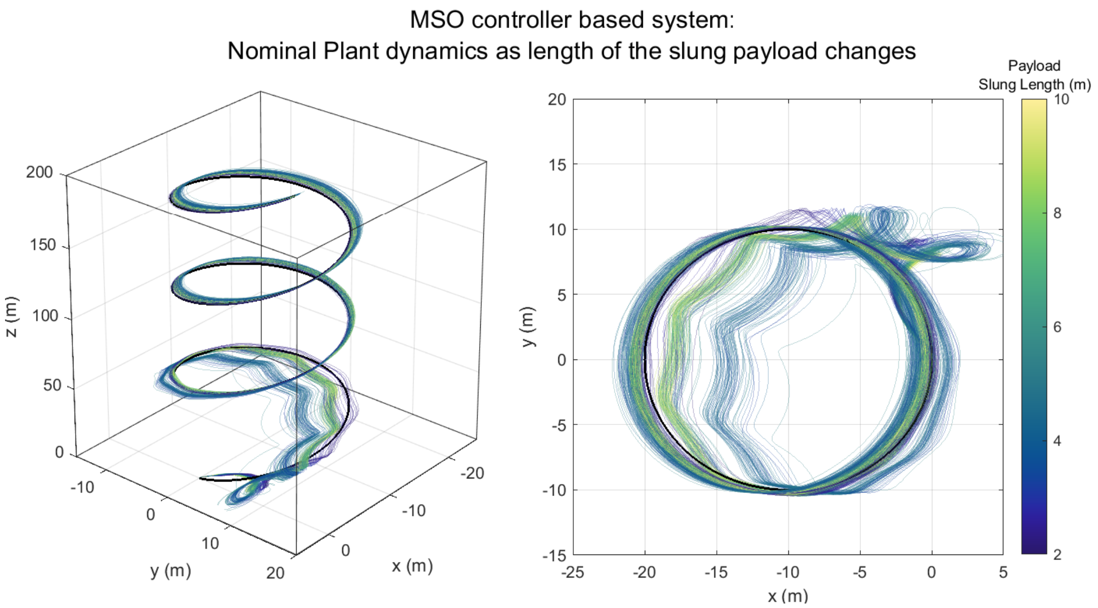


Figure 6.5: MSO based plant dynamics as the slung length of the payload is varied and payload mass is held at the nominal value. The Quadrotor traces a vertical helix with a diameter of 10 meters and a height of 200 meters. The slung load length is varied from 2 m to 10 m.

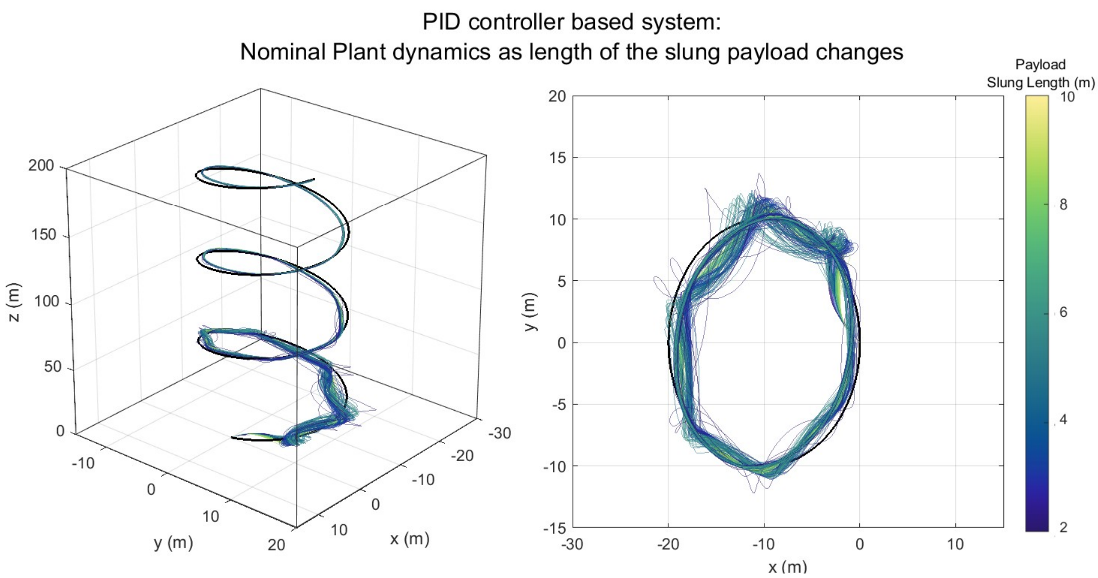


Figure 6.6: PID Plant dynamics as the slung length of the payload is varied and payload mass is held at the nominal value. The slung load length is varied from 2 m to 10 m.

As the payload mass is held at the nominal value, the PID controller plant gives a performance better than the MSO plant, except for certain short lengths. Both the experiments show that the system is susceptible to oscillations originating at the payload at shorter lengths (as seen from the blueish portion of the responses), mostly around a length of 5 meters. As the slung length is increased further the system shows better performance.

In the second set the payload mass is increased to 0.395 kg and the experiments are repeated. This time the performance of the PID controlled system can be seen have deteriorated considerably (see Fig. 6.8) compared to that of the MSO controlled plant (Fig. 6.7).

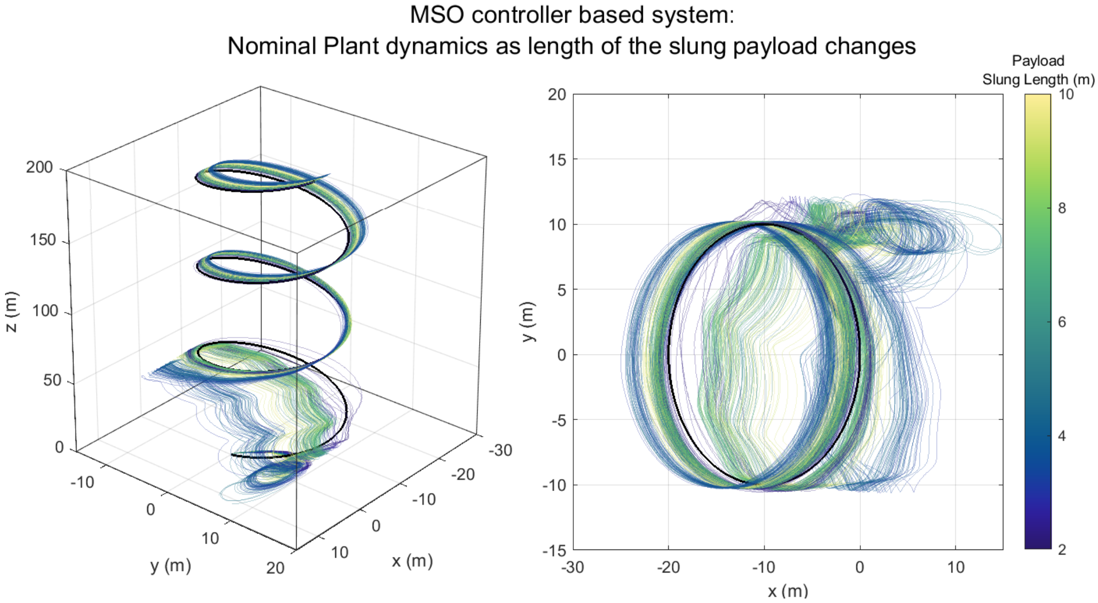


Figure 6.7: MSO based plant dynamics as the slung length of the payload is varied while payload mass is held at 0.395 kg. The slung load length is varied from 2 m to 10 m.

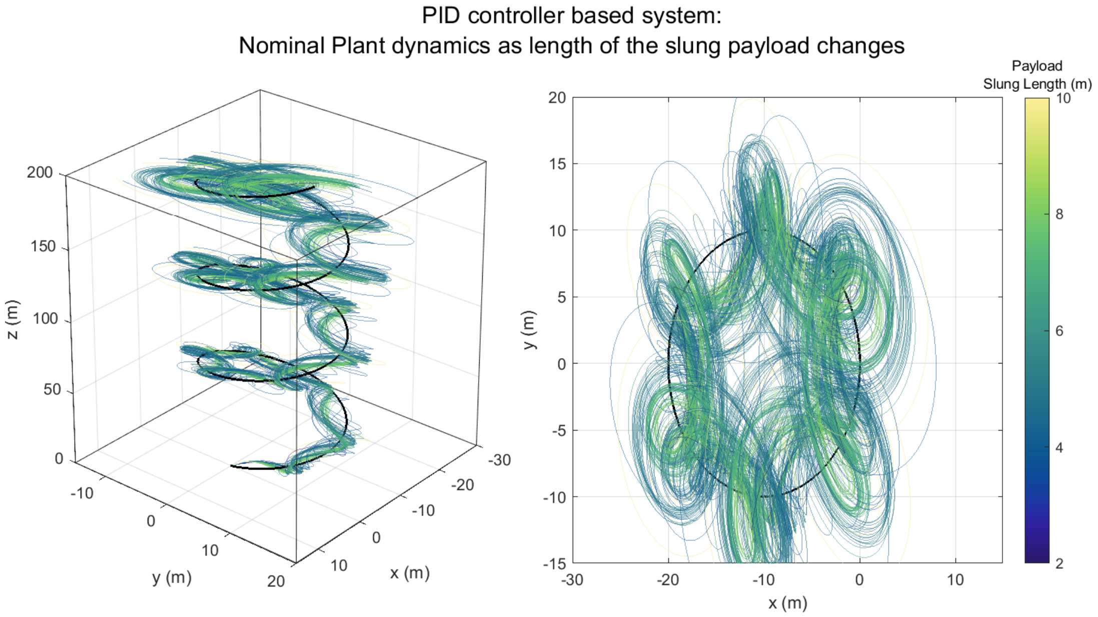


Figure 6.8: PID Plant dynamics as the slung length of the payload is varied while payload mass is held at 0.395 kg. The slung load length is varied from 2 m to 10 m.

A comparison between performances of respective systems with the nominal and increased payload mass, as the slung load is varied, presents the advantage the robust controller has, in the presence of varying model parameters. While performance deterioration is observed, stability is maintained in the case of the robust controller, but for the case of the PID based system, instability for the case plants with model uncertainty can be expected.

6.3.3 Plant response as both the slung length and payload mass is varied simultaneously

In the third set of experiments the quadrotor is made to trace the vertical helix and with the payload mass and the length at which it is attached are changed simultaneously. This simultaneous change (in the payload related parameters) is the characteristic that makes the flight of the Aerokontiki unique. We are trying to the emulate these time-varying characteristics in our experiments.

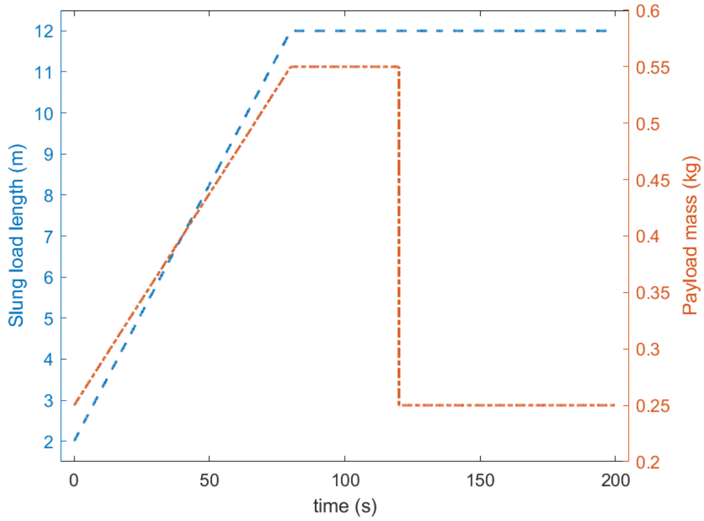


Figure 6.9: Experiment Set 3: The change in the value of the payload parameters as plotted against time.

Table 6.1: Uncertainty Sources

Parameters	% Uncertainty	Range
I_{XX}, I_{YY}	$0.04339 \pm 10\%$ kg m^2	$0.0391 \leq I_{XX}, I_{YY} \leq 0.0477$
I_{ZZ}	$0.0705 \pm 10\%$ kg m^2	$0.0634 \leq I_{ZZ} \leq 0.0775$
m	$1.27 \pm 2\%$ kg	$1.2446 \leq m \leq 1.2954$
dx, dx	$0.194 \pm 20\%$ m	$0.1552 \leq d_r \leq 0.2328$
c_t	$0.1 \pm 10\%$	$0.0500 \leq c_t \leq 0.1500$
c_p	$0.045 \pm 10\%$	$0.0315 \leq c_p \leq 0.0585$

In Figure 6.9 the time varying payload parameters are plotted. The experiments are conducted on plants with parameter uncertainties presented in Table 6.1. The uncertainty in mass of the quadrotor is only a modest 2%. This is because a time varying payload mass uncertainty is already being considered.

The results of the experiments are presented in Fig. 6.10 and Fig. 6.11. The plots are colour coded with red lines representing the responses of plants from within the parameter hypercube and blue lines representing the plants from the vertices of the hypercube.

As can be observed in the case of the MSO controlled plant (Fig.6.10), the controllers are able to stabilize the plants with model uncertainties and track the reference signal. While those plants that lie significantly further away from the nominal plant model do take time to reach the steady state (as can be noted from some off the responses in the blue lines), the responses eventually reaches the steady state and begins tracking the reference trajectory.

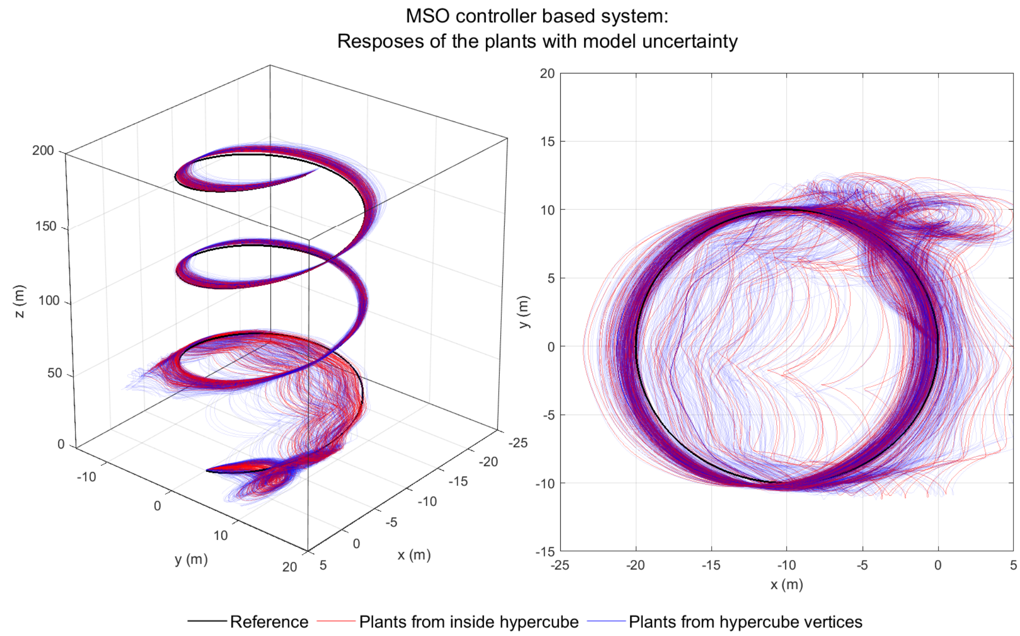


Figure 6.10: Dynamics of MSO controlled plant with model uncertainties, while payload mass and the slung length is varied.

In the case of the PID controlled plant (Fig. 6.11) the oscillations that the plants experience results in instability for many plants both from within, and from the vertices of the hypercube. While the PID controller does provide satisfactory responses for many models, the unpredictability makes it necessarily dangerous to employ them before understanding the limits of each tuned configuration. This is at a stark contrast when compared to the plant controlled by the MSO controller where the control engineer can expect a steady deterioration of performance as opposed to catastrophic failures when model parameters are unknown and some of them are vary substantially. In either cases the expected disturbance at the 120th second when the payload mass drops back to the nominal value, while is difficult to observe in the plots on a Z axis scale of 200 meters, the quick correction were observed in the pitch, roll and altitude rates in the experiments. In the experiments conducted, the developed model contain saturation elements that limit inputs.

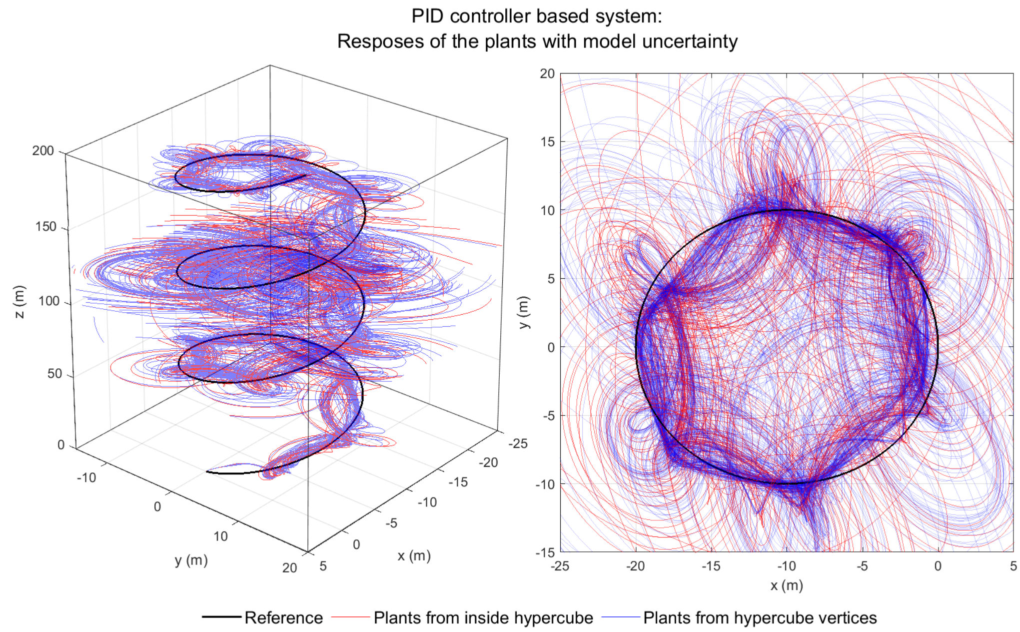


Figure 6.11: Dynamics of the PID controller plant with model uncertainties, while payload mass and the slung length is varied.

The main conclusions that are drawn from the three sets of experiments are listed out below:

1. The mixed sensitivity optimization based robust controller developed can effectively stabilize systems such as those developed for the Aerokontiki fishing application. The controller can not only handle model parameter uncertainties, but also can handle time varying payload parameters during the flight.

2. As observed in the previous chapters once the uncertainties are estimated and effectively modelled during the development of the controller, the performance outcomes become predictable even in the face of substantial model uncertainties.

3. With the PID controller while it provides satisfactory response with the nominal plants, the performance becomes chaotic and unpredictable as the payload parameters are varied. The effect of the oscillations transferred from the payload to the plant model are also seen to increase as the payload parameters move away from the nominal value.

4. The effects of the payload oscillations transferred to the quadrotor are minimum except of the initial offset (see 6.12) from the reference in the case of the MSO controlled plant.

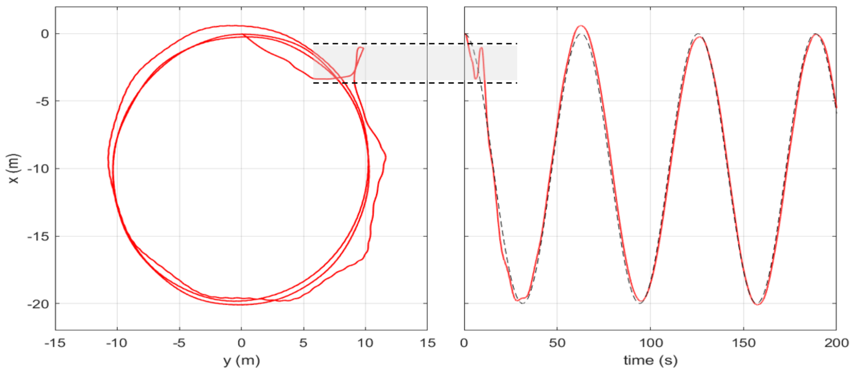


Figure 6.12: Observed offset in the response

The offset can be explained as part of an interplay between the initial overshoot of the plant response in the X axis and attempts of the controller to stabilize the oscillations translated to the craft from the swinging payload. To an extent these initial offsets are also observed in the PID controlled plant response. In the MSO controlled plant where the offsets are significant in some cases, in all of the simulations the plant output reaches the steady state within a time period of 20 seconds.

5. The steady state error in position coordinates observed in Fig. 6.10 can be accounted to the PID controllers from the outer loop. The steady state error in the altitude remains less than $\pm$ 1 meter in either cases.

The MSO robust controller development algorithm outlined in **Algorithm #5** developed in Chapter 5, enables an efficient way to develop an optimal and robust controller in this case. The algorithm allows the engineer to capture model uncertainties encountered in industrial applications such as that of the Aerokontiki flight and design effective controllers. This brings us to the conclusion of the case study.

6.4 Conclusion

The chapter takes a look at the case of the Aerokontiki, an industrial application that involves a payload that is attached to the quadrotor using a non stretchable cable. The time varying nature of the payload mass and the length at which the payload is attached to the craft, makes it a challenging problem to resolve.

A MSO attitude/altitude controller is developed to control the inner loop of the system and a PID based controller takes care of tracking the position coordinates. A comparison is performed with a system where the plant is controlled using a cascaded PID controller. The MSO based controller is able to stabilize the system and effectively track the reference set-point even in the presence of model uncertainties and time varying payload parameters. The MSO controller performs significantly better than the PID counter part, in terms of stability, rejection of noise and in terms of rejecting oscillations transferred from the payload to the craft.

The future work in this area would look at modelling the disturbance induced by the main line onto the Aerokontiki during the flight. The effectiveness of the proposed controller can then be tested more effectively. Ways to develop a robust controller to track the position coordinates will also be studied. This would remove the sub-optimality introduced into the system by the PID based position controllers.

Chapter 7

Conclusion

The overall aim of this research was to develop robust optimal quadrotor controllers, especially intended for situations where significant model uncertainty is present in the plant. To meet this goal, both SISO as well as multivariable robust controllers were developed. One of the key contributions of this thesis is the development of algorithms that could automatically develop controller weights thereby removing the ambiguity brought about by relying on the trial and error approach currently used widely.

The chapter is divided into four sections. In the first section the main ideas developed in this work are summarized. This is followed by section 7.2, where the contributions of this study are presented. Next in section 7.3, the important ideas that emerged from the study are discussed and a high-level comparison of the various controllers developed in this work is presented. In the final section 7.4, the work that will be pursued in future will be examined.

7.1 Summary

The summaries from the important chapters in this study are presented below.

7.1.1 Problem statement

After the literature review (presented in Chapter 2) of existing techniques employed to design quadrotor controllers that would perform well against model uncertainties, the robust control paradigm was selected to design controllers for resolving the tracking attitude/altitude control of the quadrotor. While attempts have been made in some studies to automatically develop optimal weights for the designed controllers, a comprehensive framework for SISO and MIMO systems that could be used was absent. It was identified that when it comes to quadrotor UAV designs, a comprehensive comparison study has not been performed among various developed robust controllers as well as with PID controllers in general.

The quadrotor problem was stated in Chapter 3 and both a non-linear and numerically linearised model was developed. To use a concrete example, the Ascending technologies (AscTec) Pelican Quadrotor, a popular drone among UAV researchers, was selected. The model uncertainty that would be considered during the controller design phase was listed. Seven parameters as identified from the literature review were considered for inducing model parametric uncertainty for later simulation. In addition to this time delay uncertainty and uncertainty due to model non-linearity is also considered (See Section 3.4). Further the design specifications that are required to be met during the controller development were listed in Section 3.5.

7.1.2 SISO systems

In Chapter 4, a decoupled version of the quadrotor was developed such that it suits the SISO controller design and further, robust control theory was briefly introduced. Algorithms were proposed for Mixed Sensitivity Optimization, Loop Shaping Design Procedure 1 & 2-DOF and μ-synthesis (**Algorithm #1 — #4**), that could be used to generate optimal weight designs alongside the development of the controller. The performances of individual closed loop systems were tested and satisfactory stability margins were obtained in each of the cases. Further, a PID controlled system was developed for the purposes of a base-test comparison. All the five closed loop systems were compared with each other and conclusions were presented.

During the design phase of the robust controllers, (except for those based on Loop Shaping Design Procedure), the value of μ was constrained to less than unity in the presence of considerable model uncertainty. For LSDP based controllers the coprime stability margins of that greater than 0.25 were achieved. From the time domain analysis, it was observed that the robust controllers gave a better performance in the presence of model uncertainty, with regards to meeting design specifications, than the PID controller. Comparisons among robust controllers were also performed based on classical stability margins observed in the systems, μ values plotted over the frequency spectrum and using the gap-metric stability criterion. Among the designed robust controllers, the LSDP based controllers offered better robustness against uncertainty, both in the terms of overall performance as well as for robust stability margins.

7.1.3 Multivariable controllers

Chapter 5 begins with a discussion of the MIMO quadrotor model properties which includes the singular value decomposition and the condition number of the plant around the bandwidth frequency. The multivariable robust controllers for the quadrotor were developed using MSO, LSDP 1-DOF and μ-synthesis strategies. Algorithms which could be utilized inorder to generate optimal control weights were proposed (**Algorithm #5 — #7**). The LSDP 1-DOF procedures were slightly modified to produce a LSDP-PI hybrid which enabled a reduction in overall controller order as well as the complexity of the optimisation problem. All three controllers were able to achieve nominal stability, nominal performance and robust stability irrespective of uncertainty in parameters that were expected.

It was observed that robust performance could be achieved in all the three cases when the uncertainty in the payload mass and thrust coefficient was removed during controller development, resulting in a value of μ_{RP} below unity. It was recognized that to develop robust quadrotor systems it was crucial to reduce uncertainty in both these parameters.

It was observed that MIMO controllers are more conservative than its SISO counterparts and hence are comfortable only with smaller percentage uncertainties. Comparison between the three developed controllers were performed based on step response characteristics, closed loop systems bandwidths and by calculating μ-values across the frequency spectrum. Similar to the SISO cases, the LSDP based controllers were found to be the most effective (in terms of larger bandwidths and higher robust stability margins) when there was no uncertainty in mass and thrust coefficient of the craft. The μ controller was the most conservative of all the three with MSO taking a position in between.

7.1.4 Simulation based case study

In Chapter 6 a simulation based case study was performed by modelling a quadrotor carrying a suspended load. The inspiration for the study comes from the case of the Aerokontiki, an industrial drone used for longline fishing. Simulated experiments were conducted with cases where payload masses and the slung load lengths were varied individually as well as simultaneously. A cascaded PID controller was also developed for the system and experiments were repeated on the same.

Comparisons between both the systems revealed that while the PID based plant performed better than the robust controller based system when the plants are close to the nominal model, the performance deteriorated substantially with significant unpredictability, for systems far away from the nominal plant. The robust controller on the other hand, offered a steady but predictable decline in the performance even as the systems moved further away from the nominal plant (as opposed to the significant unpredictability of PID controlled plants). The unique nature of time varying payload mass and the slung load length, provided a interesting setting for testing the robustness of the designed controllers. In the next section the key contributions of this study are presented.

7.2 The Contributions of this Thesis

The contributions from this research are discussed in this section. To recall the research objectives that were listed in section 1.2, the main objective of this study was to develop optimal and robust quadrotor controllers. To achieve this, three research objectives were outlined. These being: (i) Developing a framework for Single Input Single Output systems using which optimal controller weights could be developed alongside the robust quadrotor controller (ii) Extending the functionality of the framework to the multivariable domain, and finally (iii) To conduct a comprehensive in-simulation performance testing and comparison among the developed controllers from the three main branches of Robust Control Theory, namely the MSO, LSDP and μ- synthesis.

In light of these objectives, the contributions are now discussed. Robust controllers for a quadrotor model were developed in chapter 4 (where controllers for a decoupled quadrotor model involving 4 SISO systems were presented), and in chapter 5 (where multivariable controllers for the MIMO quadrotor model was presented). The associated controller weights which enabled meeting the required performance standards were developed such that design optimality was maintained in the overall system. For the SISO system MSO, LSDP (1 and 2 DoF) and μ-synthesis based controllers were developed and for the MIMO model MSO, LSDP (1 Dof) and μ-synthesis were developed. The development and testing of the controllers were enabled and complemented by the following key contributions.

1. Separate algorithms developed for MSO (**Algorithm #1**), LSDP - 1 DoF (**Algorithm #2**), LSDP - 2 Dof (**Algorithm #3**) and μ-synthesis (**Algorithm #4**) were presented in Chapter 4. These algorithms act as frameworks that could generate optimal controller weights alongside the robust controller, while providing options to incorporate model uncertainties, as well as time and frequency domain design specifications.

2. In Chapter 5, algorithms for the multivariable system, for each of the different robust control strategies namely, MSO (**Algorithm #5**), LSPD 1 DoF (**Algorithm #6**) and μ-synthesis (**Algorithm #7**) were developed. Similar to the SISO case, the multivariable frameworks allow for capturing model uncertainties as well as incorporating design standards from both time and frequency domains thus bringing in optimality to the controller weights alongside development of the robust controller.

3. Comprehensive comparison studies conducted between the developed robust controllers both in the SISO and MIMO systems provided insights into the effectiveness of various developed controllers when it came to controlling the quadrotor. Insight into ways to choose between the various robust control strategies depending upon the application for which the quadrotor controller is being designed also crystallized in these comparison studies.

4. The various algorithms proposed can be consolidated into a holistic general framework that can be used whenever an optimal-robust controller is to be designed. As can be observed in Fig. 7.1, the decision tree branches first depending on whether the linearized plant model is a SISO or a MIMO system. Then depending on information that the designer has regarding model uncertainty, the tree further spreads out.
   For cases where there is information on modeling uncertainty, depending on required performance standards the control engineer can select either between MSO or μ-synthesis designs, or on the other hand, if no model uncertainty is known the engineer could choose LSDP algorithms. The LSDP algorithms further branches out into the 1 & 2 DoF designs depending on whether there is a reference model that the closed loop system is required to emulate.

Alongside these, significant efforts had been directed towards developing a new way to visualize the stochastic Monte Carlo results obtained from performance of quadrotors with model uncertainty. These visualizing techniques made it possible to study the effects of modeling uncertainty in quadrotors from within the parameter hypercube and its vertices. These techniques helped to complement and validate the performance and comparison analysis made through out the study.

A non-trivial case study with contributions to the real world practical problems was presented in Chapter 6. The simulations that presented the case of a quadrotor with modeling uncertainties, carrying a time varying mass attached to a time varying slung load length, have provided valuable insights that the control engineer can utilize when she/he is required to make design considerations and select controllers.

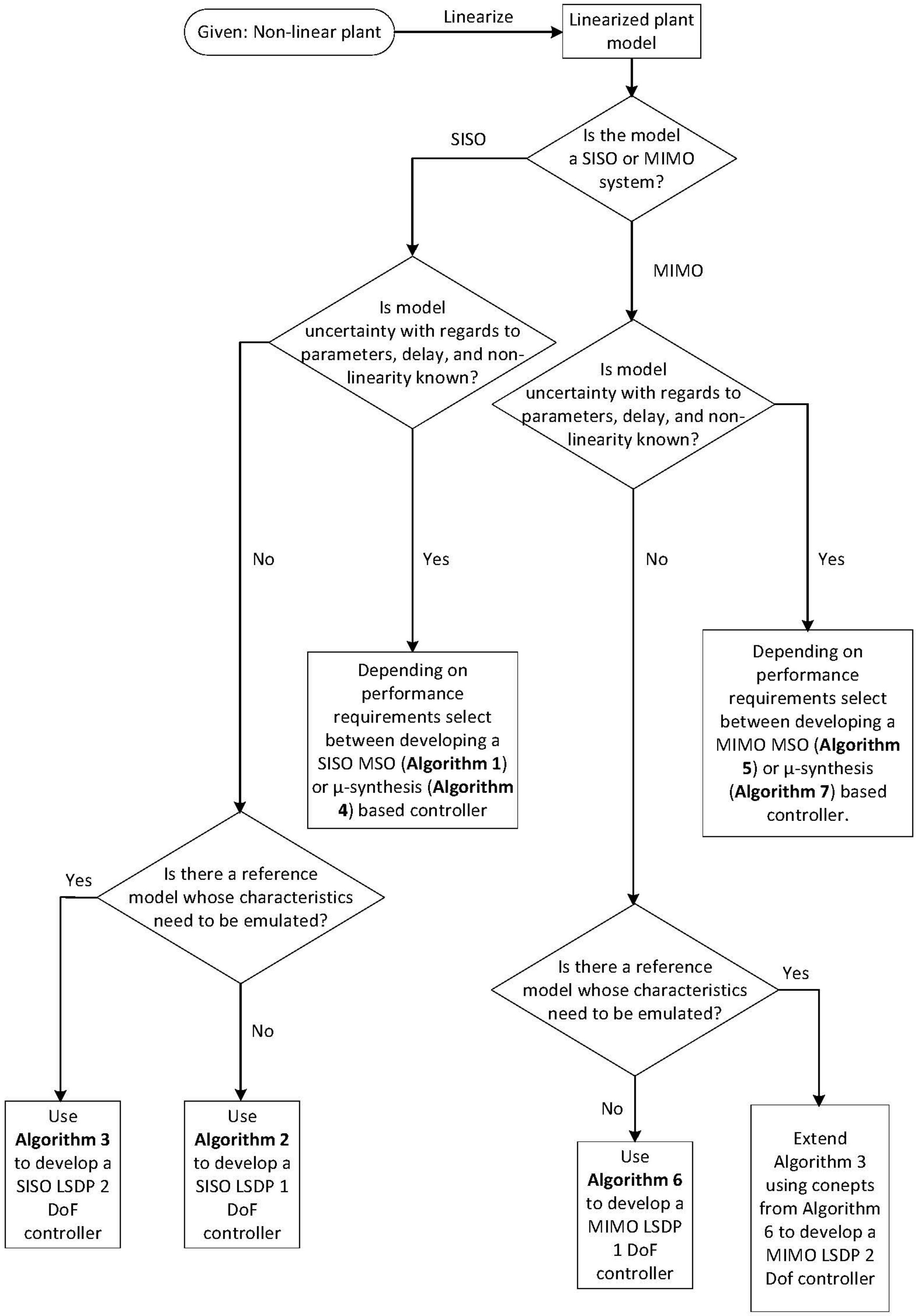


Figure 7.1: The general decision tree that can be followed to choose the appropriate algorithms for developing optimal robust controllers

In the next section a discussion on ideas that emerged from the study are presented.

7.3 Discussion

Developing optimal weights has become important in robust control designs, when gains in the form of flight quality and flight time are to be achieved during the plant operation. While a conventional trial and error approach helps the engineer to develop robust control designs, the optimality is sacrificed and the work is overly ad-hoc. In this study, the algorithms were developed that provides the control engineer options to formulate constraints, in (i) the time domain in the form of step response characteristics, and (ii) in the frequency domain as $\mathcal{H}_\infty$ norm of the closed loop functionals, during the robust quadrotor controller development phase. Algorithms for all the three major robust control designs strategies, namely the MSO, LSDP and μ-synthesis were proposed, and they were used to resolve the tracking attitude/altitude problem for a quadrotor. The successful implementation of the algorithms imply that the methods can be used in the future for other systems as well to develop optimal weights alongside design of the robust controller.

MIMO controllers, while they are conservative compared to the SISO counterparts, when designed for optimality should bring in a higher degree of savings in terms of the energy expended during flight, as the controllers are designed to keep the overall input to a minimum while achieving the required performance objectives. In the case of the SISO controller, while inputs are constrained to a minimum for individual loops but when operated in parallel (as they would be during flight), the overall design optimality may weaken. A comparison study between the MIMO and SISO systems should reveal this and would be undertaken as a part of future work.

The optimality and savings would become more significant when we take into account model uncertainties. For plants with high condition numbers, significant deterioration in performance can be expected with multi-dimensional model uncertainties. For a simple plant that can be effectively decoupled the SISO controllers may suffice, but for new and chaotic applications (like that of the Aerokontiki), a MIMO controller that presupposes modeling uncertainty would bring about significant gains in terms of flight time and performance. For the case of the quadrotor, the work performed in the Chapter 5, enabled isolating those model parameters which contributed to performance deterioration, which the control engineer need to watch for.

The time taken for the optimization problem to converge to the global optimum differed with the different robust control strategies discussed. On average the convergence took around 7-8 minutes for SISO loops and around 20-30 minutes for the MIMO systems on an HP EliteDesk 800 G4 TWR desktop operating on 8th Generation Intel® Core™ i7+ processor. These times should not be generalized on to other quadrotor configurations or applications. These convergence patterns are highly problem specific and depends to an extent on the initial values of the variables and the nature of the performance demands.

While during the development of MSO and μ-synthesis based controller, the parameter uncertainty, time delays, and non-linearity are explicitly absorbed into complimentary sensitivity weights, in LSDP the controller by nature, aims for the maximum co-prime factor uncertainty that the system can accommodate. Both these approaches offer advantages and disadvantages. Both in SISO and MIMO designs, LSDP based controllers performed better than MSO and μ controllers, but it was during the development of the latter pair, that the control engineer was able to identify and isolate model parameter uncertainties that contributed the most toward performance deterioration.

During the development of MIMO controllers, uncertainty in mass, and thrust coefficient of the craft restricted the system from achieving theoretically guaranteed robust performance i.e. $\mu_{RP} \leq 1$ (as seen for the case of both MSO and LSDP 1DOF were the values of μ_{RP} exceeded unity). Although in these cases, robust stability could still be guaranteed. We observe this in the presented case study where, while the performance deteriorates steadily, stability is still maintained.

From the work with PID controllers, (analyzed both in Chapter 4 and 6), and its comparison with robust controllers, it was observed that PID controlled systems perform more satisfactorily than robust controllers when it comes to trajectory tracking for systems that are close to the nominal plant. Not unexpectedly, as system move away from the nominal plant the performance of the PID controlled system deteriorate significantly. The slope of this deterioration is steeper when compared to that of a system controlled by a robust controller. The *Goldilocks* zone within which the PID controller outperforms the robust controller also depends on a number of other factors such as input saturation and high frequency dynamics of the plant.

To choose between selecting a PID and a robust controller when it comes to UAV controller design, model uncertainty limits will need to be estimated accurately and experiments performed through numerical simulations. This would enable the control engineer to identify the the areas where the PID based system would behave unpredictably leading to instability. For applications (like that of the Aerokontiki fishing drone), with multi-dimensional uncertainty it would be safe to develop a robust controller that guarantees robust stability with a predictable performance decline, as opposed to catastrophic and unpredictable failures which the PID controllers might bring about.

Table 7.1 presented next, gives an overview of the comparison of various controllers developed through the study. The table gives a quick overview of what each controller can achieve in the context of attitude/altitude rate tracking of a quadrotor. Having been able to develop this table, partially justifies the reason behind the development of various controllers from all the three main branches of robust control theory.

Table 7.1: A general comparison of various control strategies developed in this study in context of the tracking attitude/altitude rate control of a quadrotor.

Control Strategies	Comparison criteria	SISO Controllers	MIMO Controller
Mixed Sensitivity Optimization	• Ability to analyze and isolate individual parameters of the Quadrotor that contribute towards performance deterioration. Uncertainty in mass and thrust coefficient were noted and isolated both in the SISO (see page 61), and the MIMO case (see page 157), during controller development.	✔	✔
	• Effective against unstructured model uncertainty arising from the multivariable nature of the system	✖	✔
	• Ability to guarantee *nominal stability*, *nominal performance*, and *robust stability* considering all or a selected set of the model uncertainty by the automated algorithm- **Algorithm #1** for SISO and **Algorithm #5** for the MIMO system (see page 63 for SISO and page 164 for the MIMO case)	✔	✔
	• Ability to guarantee *robust performance* considering all or a selected set of the model uncertainty by the automated algorithm (see page 63 for SISO and page 164 for the MIMO case)	✖	✖
Loop Shaping Design Procedure - 1 Degree of Freedom	• Ability to analyze and isolate individual parameters of the Quadrotor that contribute towards performance deterioration	✖	✖
	• Effective against unstructured model uncertainty arising from the multivariable nature of the system	✖	✔
	• Ability to guarantee *nominal stability*, and *nominal performance* considering all or a selected set of the model uncertainty by the automated algorithm- **Algorithm #2** for SISO and **Algorithm #6** for the MIMO system (see page 89 for SISO and page 171 for the MIMO case)	✔	✔

Table 7.1 continued...

	• Ability to guarantee *robust stability*, and *robust performance* considering all or a selected set of the model uncertainty by the automated algorithm • Effectiveness against coprime-factor uncertainty	✖ ✔	✖ ✔
Loop Shaping Design Procedure - 2 Degree of Freedom	• Ability to design the controller so that it can follow the closed loop characteristics of a predefined model (refer page 99). • Ability to analyze and isolate individual parameters of the Quadrotor that contribute towards performance deterioration • Effective against unstructured model uncertainty arising from the multivariable nature of the system • Ability to guarantee *nominal stability*, and *nominal performance* considering all or a selected set of the model uncertainty by the automated algorithm- **Algorithm #3** for SISO system which can be extended to the MIMO system (see page 105 for the SISO case). • Ability to guarantee *robust stability*, and *robust performance* considering all or a selected set of the model uncertainty by the automated algorithm • Effectiveness against coprime-factor uncertainty	✔ ✖ ✖ ✔ ✖ ✔	✔ ✖ ✔ ✔ ✖ ✔
μ-Synthesis	• Ability to analyze and isolate individual parameters of the Quadrotor that contribute towards performance deterioration. Uncertainty in mass or thrust coefficient were noted and isolated in the SISO (see Table 4.27), and the MIMO case (see page 179), during controller development. • Effective against unstructured model uncertainty arising from the multivariable nature of the system	✔ ✖	✔ ✔

Table 7.1 continued…

	• Ability to guarantee *nominal stability*, *nominal performance*, *robust stability*, and *robust performance* considering all or a selected set of the model uncertainty by the automated algorithm- **Algorithm #4** for SISO and **Algorithm #7** for the MIMO system (see page 125 for SISO and page 186 for the MIMO case)	✔	✔
Proportional Integral Derivative Controller	• Ability to analyze and isolate individual parameters of the Quadrotor that contribute towards performance deterioration	✖	—
	• Effective against unstructured model uncertainty arising from the multivariable nature of the system	✖	—
	• Ability to guarantee *nominal stability*, and *nominal performance* of the plant while ensuring a degree of robustness	✔	—
	• Ability to guarantee *robust stability*, and *robust performance* considering all or a selected set of the model uncertainty by the automated algorithm	✖	—

7.4 Future Work

There are five areas that have been identified for future work. The first three would be those that would immediately supplement the conducted research and the final two areas would follow on naturally from the work presented in the study.

The initial priority would be on developing a robust observer that would enable state estimation. Measurement inaccuracy that comes along from sensor noise affects accurate state estimation especially for a UAV when the noise has been observed to be coloured as opposed to Gaussian white noise [218]. As noted in [219], in such situations an Unscented Kalman Filter (UKF) would prove to be more effective than the traditional Extended Kalman Filter to produce accurate estimation results. Ways to effectively estimate the attitude, altitude, and their rates using an UKF have been explored previously in [220, 221] and these studies would be examined and developed on for designing the state estimator for the model presented in the thesis.

Secondly, efforts will be directed towards developing an anti-windup scheme to deal with the actuator saturation experienced by the plant. One of the sources of non-linearity in quadrotor is actuator saturation. The Anti-windup compensator presented in [222] is well suited for this work with the key objective of the compensator to minimize the $\mathcal{L}_2$ norm u_{in} and y_d as seen in the Figure 7.2.

Here M represents the parameterized version of the compensator, G_2 the plant feedback part, with K being a stabilizing controller of the plant G. The Linear Matrix Equalities listed in the study can be resolved using the MATLAB LMI toolbox.

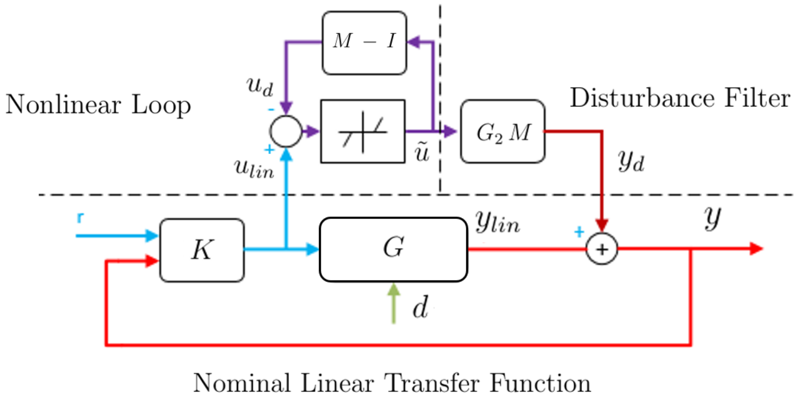


Figure 7.2: The anti-windup scheme.

Next, work could be performed on reducing the controller order for the designed systems. This becomes important for implementing the robust controllers of real-world platforms especially for the multivariable controllers. As can seen from Table

7.2, the orders of designed robust controllers are higher compared to those of the PID controllers.

While commands from the MATLAB Robust Control Toolbox significantly simplifies the effort required to reduce the controller order, non-trivial testing needs to be preformed before the controllers can be safely deployed.

Table 7.2: Controller order – SISO Systems

Type	Roll & Pitch	Yaw	Altitude
MSO	3	3	3
LSDP 1 DOF	4	4	4
LSDP 2 DOF	3	3	3
μ	5	5	6
PID	2	2	2

Controller order – MIMO Systems

Type	Order
MSO	12
LSDP 1 DOF	11
μ	12

To follow on from the research work presented in this study, the primary work will be directed towards developing an effective methodology to compare the the PID and Robust controller based systems developed in Chapter 6. Currently the systems have only been compared by means of stability imparted by the controller to the closed loops. This will be further expanded using traditional methodologies such as calculating the Integral Absolute Error with regards to trajectory tracking and also using those extended from Robust Control literature. Efforts will also be directed to perform comparisons between SISO and MIMO robust controllers. The impact of model uncertainty and the gains that could obtained when the plant is stabilized using both the controllers will be studied.

Finally efforts will be directed at developing a framework in the context of UAVs to determine the zones where a robust controller would outperform the PID controller. This would make it easier to isolate those systems and applications for which robust controllers would prove better and safer than working with a PID based system.

This brings us to the end of this chapter, and the thesis. References used in the study is presented next, followed by the Appendix.

Bibliography

[1] Khaleghi, A. M., Xu, D., Wang, Z., Li, M., Lobos, A., Liu, J., and Son, Y.-J. "A DDDAMS-based planning and control framework for surveillance and crowd control via UAVs and UGVs". In: *Expert Systems with Applications* 40.18 (2013), pp. 7168–7183.

[2] McGill, P., Reisenbichler, K., Etchemendy, S., Dawe, T., and Hobson, B. "Aerial surveys and tagging of free-drifting icebergs using an unmanned aerial vehicle (UAV)". In: *Deep Sea Research Part II: Topical Studies in Oceanography* 58.11-12 (2011), pp. 1318–1326.

[3] Metni, N. and Hamel, T. "A UAV for bridge inspection: Visual servoing control law with orientation limits". In: *Automation in construction* 17.1 (2007), pp. 3–10.

[4] Immerzeel, W., Kraaijenbrink, P., Shea, J., Shrestha, A., Pellicciotti, F, Bierkens, M., and De Jong, S. "High-resolution monitoring of Himalayan glacier dynamics using unmanned aerial vehicles". In: *Remote Sensing of Environment* 150 (2014), pp. 93–103.

[5] Varela, G., Caamaño, P., Orjales, F., Deibe, Á., López-Peña, F., and Duro, R. J. "Autonomous UAV based search operations using constrained sampling evolutionary algorithms". In: *Neurocomputing* 132 (2014), pp. 54–67.

[6] Tuna, G., Nefzi, B., and Conte, G. "Unmanned aerial vehicle-aided communications system for disaster recovery". In: *Journal of Network and Computer Applications* 41 (2014), pp. 27–36.

[7] Weissbach, D. and Tebbe, K. "Drones in sight: rapid growth through M & A's in a soaring new industry". In: *Strategic Direction* 32.6 (2016), pp. 37–39.

[8] Watkins, L., Sartalamacchia, S., Bradt, R., Dhareshwar, K., Bagga, H., Robinson, W. H., and Rubin, A. "Defending Against Consumer Drone Privacy Attacks: A Blueprint For A Counter Autonomous Drone Tool". In: *Workshop on Decentralized IoT Systems and Security (DISS)* (2020). DOI: 10.14722/diss.2020.23010.

[9] Kendoul, F. "Survey of advances in guidance, navigation, and control of unmanned rotorcraft systems". In: *Journal of Field Robotics* 29.2 (2012), pp. 315–378.

[10] Chao, H., Cao, Y., and Chen, Y. "Autopilots for small unmanned aerial vehicles: a survey". In: *International Journal of Control, Automation and Systems* 8.1 (2010), pp. 36–44.

[11] Hayat, S., Yanmaz, E., and Muzaffar, R. "Survey on unmanned aerial vehicle networks for civil applications: A communications viewpoint". In: *IEEE Communications Surveys & Tutorials* 18.4 (2016), pp. 2624–2661.

[12] Cai, G., Dias, J., and Seneviratne, L. “A survey of small-scale unmanned aerial vehicles: Recent advances and future development trends”. In: *Unmanned Systems* 2.02 (2014), pp. 175–199.

[13] Mustapa, Z., Saat, S., Husin, S., and Abas, N. “Altitude controller design for multi-copter UAV”. In: *Computer, Communications, and Control Technology (I4CT), 2014 International Conference on*. IEEE. 2014, pp. 382–387.

[14] Fernando, T., Chandiramani, J., Lee, T., and Gutierrez, H. “Robust adaptive geometric tracking controls on SO(3) with an application to the attitude dynamics of a quadrotor UAV”. In: *Decision and Control and European Control Conference (CDC-ECC), 2011 50th IEEE Conference on*. IEEE. 2011, pp. 7380–7385.

[15] Schioler, H., Leth, J., Leth, T., and Totu, L. “Stochastic design of switching controller for quadrotor UAV under intermittent localization.” In: *2016 Australian Control Conference (AuCC)*. 2016, pp. 243–248. DOI: `10.1109/AUCC.2016.7868196`.

[16] Tanveer, M. H., Hazry, D, Ahmed, S. F., Joyo, M. K., Warsi, F. A., Kamaruddin, H, Razlan, Z. M., Wan, K., and Shahriman, A. “NMPC-PID based control structure design for avoiding uncertainties in attitude and altitude tracking control of quad-rotor(UAV)”. In: *Signal Processing & its Applications (CSPA), 2014 IEEE 10th International Colloquium on*. IEEE. 2014, pp. 117–122.

[17] Hercus, R., Kong, H. S., and Ho, K. F. “Control of an unmanned aerial vehicle using a neuronal network”. In: *2013 IEEE Symposium on Computational Intelligence, Cognitive Algorithms, Mind, and Brain (CCMB)*. 2013, pp. 73–79. DOI: `10.1109/CCMB.2013.6609168`.

[18] Roberge, V., Tarbouchi, M., and Labonté, G. “Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning”. In: *IEEE Transactions on Industrial Informatics* 9.1 (2013), pp. 132–141.

[19] Min, B.-C., Hong, J.-H., and Matson, E. T. “Adaptive robust control (ARC) for an altitude control of a quadrotor type UAV carrying unknown payload”. In: *Control, Automation and Systems (ICCAS), 2011 11th International Conference on*. IEEE. 2011, pp. 1147–1151.

[20] Doyle, J. “Guaranteed margins for LQG regulators”. In: *IEEE Transactions on automatic Control* 23.4 (1978), pp. 756–757.

[21] Başak, H. and Prempain, E. “Switching recovery control of a quadcopter UAV”. In: *Control Conference (ECC), 2015 European*. IEEE. 2015, pp. 3641–3646.

[22] Lopez-Martinez, M, Vivas, C, and Ortega, M. “A multivariable nonlinear H_∞ controller for a laboratory helicopter”. In: *Decision and Control, 2005 and 2005 European Control Conference. CDC-ECC'05. 44th IEEE Conference on*. IEEE. 2005, pp. 4065–4070.

[23] Sadeghzadeh, I., Chamseddine, A., Theilliol, D., and Zhang, Y. “Linear Parameter Varying control synthesis: State feedback versus H_∞ technique with application to quadrotor UAV”. In: *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*. IEEE. 2014, pp. 1099–1104.

[24] Cai, G., Chen, B. M., and Lee, T. H. “Design and implementation of robust flight control system for a small-scale UAV helicopter”. In: *Asian Control Conference, 2009. ASCC 2009. 7th*. IEEE. 2009, pp. 691–697.

[25] Kohno, S. and Uchiyama, K. “Design of robust controller of fixed-wing UAV for transition flight”. In: *Unmanned Aircraft Systems (ICUAS), 2014 International Conference on*. IEEE. 2014, pp. 1111–1116.

[26] Raffo, G. V., Ortega, M. G., and Rubio, F. R. “Backstepping/nonlinear H_∞ control for path tracking of a quadrotor unmanned aerial vehicle”. In: *American Control Conference, 2008*. IEEE. 2008, pp. 3356–3361.

[27] López, J., Dormido, R., Dormido, S., Gómez, J. P., and Gómez, I. “An H-infinite robust tracker controller for an UAV under realistic simulated environmental effects”. In: *IFAC Proceedings Volumes* 41.2 (2008), pp. 6136–6141.

[28] Gaitan, A. T. and Bolea, Y. “Modeling and robust attitude control of a quadrotor system”. In: *2013 10th international conference on electrical engineering, computing science and automatic control (CCE)*. IEEE. 2013, pp. 7–12.

[29] Donha, D. C. and Katebi, M. “Automatic weight selection for controller synthesis”. In: *International Journal of Systems Science* 38.8 (2007), pp. 651–664.

[30] Kaur, R. and Ohri, J. “PSO Based Weight Selection and Fixed Structure Robust Loop Shaping Control for Pneumatic Servo System with 2DOF Controller”. In: *World Academy of Science, Engineering and Technology InternationalJournal of Electrical, Computer, Energetic, Electronic and Communication Engineering* 8.8 (2014).

[31] Lo, K. and Khan, L. “Hierarchical micro-genetic algorithm paradigm for automatic optimal weight selection in H_∞ loop-shaping robust flexible AC transmission system damping control design”. In: *IEE Proceedings-Generation, Transmission and Distribution* 151.1 (2004), pp. 109–118.

[32] Le Digabel, S. “Algorithm 909: NOMAD: Nonlinear optimization with the MADS algorithm”. In: *ACM Transactions on Mathematical Software (TOMS)* 37.4 (2011), p. 44.

[33] Michailidis, M. G., Rutherford, M. J., and Valavanis, K. P. “A survey of controller designs for new generation UAVs: The challenge of uncertain aerodynamic parameters”. In: *International Journal of Control, Automation and Systems* 18.4 (2020), pp. 801–816.

[34] Brown, G. “Discovering the STM32 microcontroller”. In: *Cortex* 3 (2012), p. 34.

[35] Sá, R. C., Araújo, A. L. C. de, Varela, A. T., and Barreto, G. d. A. “Construction and PID control for stability of an unmanned aerial vehicle of the type quadrotor”. In: *2013 Latin American Robotics Symposium and Competition*. IEEE. 2013, pp. 95–99.

[36] Rafique, S. F., Bodla, M. K., Ahmed, Z., Nasir, U., Zaidi, A., and Saleem, M. “Design and implementation of a UAV for power system utility inspection”. In: *2014 16th International Power Electronics and Motion Control Conference and Exposition*. IEEE. 2014, pp. 1146–1150.

[37] Abdou, L., Glida, H.-E., et al. "Parameters tuning of a quadrotor PID controllers by using nature-inspired algorithms". In: *Evolutionary Intelligence* (2019), pp. 1–13.

[38] Pratama, N., Manggau, F. X., and Betaubun, P. "Attitude quadrotor control system with optimization of PID parameters based on fast genetic algorithm". In: *International Journal of Mechanical Engineering and Technology (IJMET)* 10.1 (2019), pp. 335–343.

[39] Khatoon, S., Gupta, D., and Das, L. "PID & LQR control for a quadrotor: Modeling and simulation". In: *2014 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. IEEE. 2014, pp. 796–802.

[40] Sahul, M., Chander, V. N., and Kurian, T. "A novel method on Disturbance Rejection PID Controller for Quadcopter based on Optimization algorithm". In: *IFAC Proceedings Volumes* 47.1 (2014), pp. 192–199.

[41] Argentim, L. M., Rezende, W. C., Santos, P. E., and Aguiar, R. A. "PID, LQR and LQR-PID on a quadcopter platform". In: *2013 International Conference on Informatics, Electronics and Vision (ICIEV)*. IEEE. 2013, pp. 1–6.

[42] Criado, R. M. and Rubio, F. R. "Autonomous path tracking control design for a comercial quadcopter". In: *IFAC-PapersOnLine* 48.9 (2015), pp. 73–78.

[43] Tnunay, H., Abdurrohman, M. Q., Nugroho, Y., Inovan, R., Cahyadi, A., and Yamamoto, Y. "Auto-tuning quadcopter using Loop Shaping". In: *2013 International Conference on Computer, Control, Informatics and Its Applications (IC3INA)*. IEEE. 2013, pp. 111–115.

[44] Gaikwad, S., Dash, S., and Stein, G. "Auto-tuning PID using loop-shaping ideas". In: *Proceedings of the 1999 IEEE International Conference on Control Applications (Cat. No. 99CH36328)*. Vol. 1. IEEE. 1999, pp. 589–593.

[45] Yang, J., Cai, Z., Lin, Q., and Wang, Y. "Self-tuning PID control design for quadrotor UAV based on adaptive pole placement control". In: *2013 Chinese Automation Congress*. IEEE. 2013, pp. 233–237.

[46] Orsag, M. and Bogdan, S. "Hybrid control of quadrotor". In: *2009 17th Mediterranean Conference on Control and Automation*. IEEE. 2009, pp. 1239–1244.

[47] Yu, B., Zhang, Y., and Qu, Y. "Fault tolerant control using PID structured optimal technique against actuator faults in a quadrotor UAV". In: *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE. 2014, pp. 167–174.

[48] Nguyen, N. T. "Verifiable Adaptive Control with Analytical Stability Margins by Optimal Control Modification". In: *AIAA Guidance, Navigation, and Control Conference*. 2010, p. 7770.

[49] Emran, B. J. and Yesildirek, A. "Robust nonlinear composite adaptive control of quadrotor". In: *International journal of digital information and wireless Communications* 4.2 (2014), pp. 213–225.

[50] Chen, F., Wu, Q., Jiang, B., and Tao, G. "A reconfiguration scheme for quadrotor helicopter via simple adaptive control and quantum logic". In: *IEEE Transactions on Industrial Electronics* 62.7 (2015), pp. 4328–4335.

[51] Wilson, D. I. *Advanced control using MATLAB: or Stabilising the unstabilisable*. David I. Wilson, 2020.

[52] Rohrs, C, Valavani, L., Athans, M., and Stein, G. "Robustness of continuous-time adaptive control algorithms in the presence of unmodeled dynamics". In: *IEEE Transactions on Automatic Control* 30.9 (1985), pp. 881–889.

[53] Hernandez-Martinez, E. G., Fernandez-Anaya, G., Ferreira, E., Flores-Godoy, J. J., and Lopez-Gonzalez, A. "Trajectory tracking of a quadcopter UAV with optimal translational control". In: *IFAC-PapersOnLine* 48.19 (2015), pp. 226–231.

[54] Suicmez, E. C. and Kutay, A. T. "Optimal path tracking control of a quadrotor UAV". In: *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE. 2014, pp. 115–125.

[55] Reyes-Valeria, E., Enriquez-Caldera, R., Camacho-Lara, S., and Guichard, J. "LQR control for a quadrotor using unit quaternions: Modeling and simulation". In: *CONIELECOMP 2013, 23rd International Conference on Electronics, Communications and Computing*. IEEE. 2013, pp. 172–178.

[56] Cowling, I. D., Yakimenko, O. A., Whidborne, J. F., and Cooke, A. K. "A prototype of an autonomous controller for a quadrotor UAV". In: *2007 European Control Conference (ECC)*. IEEE. 2007, pp. 4001–4008.

[57] Sasongko, R., Sembiring, J, Muhammad, H, and Mulyanto, T. "Path following system of small unmanned autonomous vehicle for surveillance application". In: *2011 8th Asian Control Conference (ASCC)*. IEEE. 2011, pp. 1259–1264.

[58] Minh, L. D. and Ha, C. "Modeling and control of quadrotor MAV using vision-based measurement". In: *International Forum on Strategic Technology 2010*. IEEE. 2010, pp. 70–75.

[59] Zhang, Y., Wang, T., Liang, J., Wang, C., Chen, Y., Zhou, Y., Luan, Y., and Gao, H. "An implement of RPV control system for small unmanned helicopters". In: *2012 IEEE International Conference on Robotics and Biomimetics (ROBIO)*. IEEE. 2012, pp. 1141–1145.

[60] Tang, Y.-R. and Li, Y. "Design of an optimal flight control system with integral augmented compensator for a nonlinear UAV helicopter". In: *Proceedings of the 10th World Congress on Intelligent Control and Automation*. IEEE. 2012, pp. 3927–3932.

[61] Shao, J., Fei, Q., Hu, Q., and Geng, Q. "Multi-model control design for longitudinal dynamics of fixed-wing UAV". In: *Proceedings of the 33rd Chinese Control Conference*. IEEE. 2014, pp. 8826–8830.

[62] Hendrix, C. D., Veth, M. J., and Carr, R. W. "LQG control design for a hovering micro air vehicle using an optical tracking system". In: *2009 IEEE Aerospace conference*. IEEE. 2009, pp. 1–14.

[63] Hang, Y., Jihong, Z., Xiaming, Y., and Chao, Z. "Robust hover control of thrust-vectored unmanned tail-sitter aircraft against gust load". In: *2013 10th IEEE international conference on control and automation (ICCA)*. IEEE. 2013, pp. 735–738.

[64] Alexis, K., Papachristos, C., Nikolakopoulos, G., and Tzes, A. "Model predictive quadrotor indoor position control". In: *2011 19th Mediterranean Conference on Control & Automation (MED)*. IEEE. 2011, pp. 1247–1252.

[65] Ganga, G and Dharmana, M. M. “MPC controller for trajectory tracking control of quadcopter”. In: *2017 International Conference on Circuit, Power and Computing Technologies (ICCPCT)*. IEEE. 2017, pp. 1–6.

[66] Alexis, K., Nikolakopoulos, G., and Tzes, A. “On trajectory tracking model predictive control of an unmanned quadrotor helicopter subject to aerodynamic disturbances”. In: *Asian Journal of Control* 16.1 (2014), pp. 209–224.

[67] Bangura, M. and Mahony, R. “Real-time model predictive control for quadrotors”. In: *IFAC Proceedings Volumes* 47.3 (2014), pp. 11773–11780.

[68] Alexis, K., Nikolakopoulos, G., and Tzes, A. “Model predictive quadrotor control: attitude, altitude and position experimental studies”. In: *IET Control Theory & Applications* 6.12 (2012), pp. 1812–1827.

[69] Chen, X. and Wang, L. “Cascaded model predictive control of a quadrotor UAV”. In: *2013 Australian Control Conference*. IEEE. 2013, pp. 354–359.

[70] Raffo, G. V., Ortega, M. G., and Rubio, F. R. “An underactuated H_∞ control strategy for a quadrotor helicopter”. In: *2009 European Control Conference (ECC)*. IEEE. 2009, pp. 3845–3850.

[71] Joelianto, E., Sumarjono, E. M., Budiyono, A., and Penggalih, D. R. “Model predictive control for autonomous unmanned helicopters”. In: *Aircraft Engineering and Aerospace Technology* (2011).

[72] Joelianto, E., Budiyono, A., Wijayanti, I. E., Megawati, N. Y., et al. “Model predictive control for obstacle avoidance as hybrid systems of small scale helicopter”. In: *2013 3rd International Conference on Instrumentation Control and Automation (ICA)*. IEEE. 2013, pp. 127–132.

[73] Izadi, H. A., Gordon, B. W., and Zhang, Y. “A data-driven fault tolerant model predictive control with fault identification”. In: *2010 Conference on Control and Fault-Tolerant Systems (SysTol)*. IEEE. 2010, pp. 732–737.

[74] Trachte, J., Gonzalez, F., and McFadyen, A. “Nonlinear model predictive control for a multi-rotor with heavy slung load”. In: *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE. 2014, pp. 1105–1110.

[75] Abe, A. “Trajectory planning for flexible Cartesian robot manipulator by using artificial neural network: numerical simulation and experimental verification”. In: *Robotica* 29.5 (2011), p. 797.

[76] Liu, C., Chen, Q., and Wang, D. “CPG-inspired workspace trajectory generation and adaptive locomotion control for quadruped robots”. In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 41.3 (2011), pp. 867–880.

[77] Hercus, R., Kong, H.-S., and Ho, K.-F. “Control of an unmanned aerial vehicle using a neuronal network”. In: *2013 IEEE Symposium on Computational Intelligence, Cognitive Algorithms, Mind, and Brain (CCMB)*. IEEE. 2013, pp. 73–79.

[78] Atmeh, G. and Subbarao, K. “A dynamic neural network with feedback for trajectory generation”. In: *IFAC-PapersOnLine* 49.1 (2016), pp. 367–372.

[79] Passino, K. M., Yurkovich, S., and Reinfrank, M. *Fuzzy control*. Vol. 42. Citeseer, 1998.

[80] Rodić, A. D. and Stojković, I. R. "Dynamic Inversion Control of quadrotor with complementary Fuzzy logic compensator". In: *11th Symposium on Neural Network Applications in Electrical Engineering*. IEEE. 2012, pp. 53–58.

[81] Maj, W. S. and Butkiewicz, B. "Flying n-copter with fuzzy logic control". In: *2013 Signal Processing Symposium (SPS)*. IEEE. 2013, pp. 1–6.

[82] Liu, Z., Yuan, C, Zhang, Y., and Luo, J. "A learning-based fuzzy lqr control scheme for height control of an unmanned quadrotor helicopter". In: *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE. 2014, pp. 936–941.

[83] Yang, Y. and Yan, Y. "Attitude regulation for unmanned quadrotors using adaptive fuzzy gain-scheduling sliding mode control". In: *Aerospace Science and Technology* 54 (2016), pp. 208–217.

[84] Koslosky, E., Wehrmeister, M. A., Fabro, J. A., and Oliveira, A. S. de. "On using fuzzy logic to control a simulated hexacopter carrying an attached pendulum". In: *Designing with Computational Intelligence*. Springer, 2017, pp. 1–32.

[85] Holland, J. H. "Genetic algorithms and the optimal allocation of trials". In: *SIAM Journal on Computing* 2.2 (1973), pp. 88–105.

[86] Qu, Y.-h., Pan, Q., and Yan, J.-g. "Flight path planning of UAV based on heuristically search and genetic algorithms". In: *31st Annual Conference of IEEE Industrial Electronics Society, 2005. IECON 2005.* IEEE. 2005, 5–pp.

[87] Roberge, V., Tarbouchi, M., and Labonté, G. "Comparison of parallel genetic algorithm and particle swarm optimization for real-time UAV path planning". In: *IEEE Transactions on industrial informatics* 9.1 (2012), pp. 132–141.

[88] Nikolos, I. K., Valavanis, K. P., Tsourveloudis, N. C., and Kostaras, A. N. "Evolutionary algorithm based offline/online path planner for UAV navigation". In: *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)* 33.6 (2003), pp. 898–912.

[89] Pehlivanoglu, Y. V. "A new vibrational genetic algorithm enhanced with a Voronoi diagram for path planning of autonomous UAV". In: *Aerospace Science and Technology* 16.1 (2012), pp. 47–55.

[90] Safonov, M. G. "Origins of robust control: Early history and future speculations". In: *Annual Reviews in Control* 36.2 (2012), pp. 173–181.

[91] Boyd, S., El Ghaoui, L., Feron, E., and Balakrishnan, V. *Linear matrix inequalities in system and control theory*. SIAM, 1994.

[92] Yaesh, I., Boyarski, S., and Shaked, U. "Probability-guaranteed robust H_∞ performance analysis and state-feedback design". In: *Systems & control letters* 48.5 (2003), pp. 351–364.

[93] Whidborne, J., Murad, G, Gu-D-W, and Postlethwaite. "Robust control of an unknown plant -the IFAC 93 benchmank". In: *International Journal of Control* 61.3 (1995), pp. 589–640.

[94] Doyle, J., Glover, K., Khargonekar, P., and Francis, B. "State-space solutions to standard H_2 and H_∞ control problems". In: *1988 American Control Conference*. IEEE. 1988, pp. 1691–1696.

[95] Başak, H. and Prempain, E. “Low order H_∞ controllers for a quadrotor UAV”. In: *2014 UKACC International Conference on Control (CONTROL)*. IEEE. 2014, pp. 127–132.

[96] Athanasius, G. X., Zhu, J. G., and Sreenatha, A. G. “Robust control of unmanned arial vehicles”. In: *2009 International Conference on Advances in Computing, Control, and Telecommunication Technologies*. IEEE. 2009, pp. 153–155.

[97] Kis, L. and Lantos, B. “Quadrotor control based on partial sensor data”. In: *19th International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD 2010)*. IEEE. 2010, pp. 43–48.

[98] Wang, X., Lu, G., and Zhong, Y. “Robust H_∞ attitude control of a laboratory helicopter”. In: *Robotics and Autonomous Systems* 61.12 (2013), pp. 1247–1257.

[99] Downs, J., Prentice, R., Alzell, S, Besachio, A., Ivler, C., Tischler, M. B., and Mansur, M. “Control system development and flight test experience with the MQ-8B fire scout vertical take-off unmanned aerial vehicle (VTUAV)”. In: *Annual Forum Proceedings-American Helicopter Society*. Vol. 63. 1. American Helicopter Society, Inc. 2007, p. 566.

[100] Shiau, J.-K. “Structurally Constrained H_∞ Suboptimal Control Problems: Low-order Controller Design”. In: *Tamkang Journal of Science and Engineering* 2.4 (2000), pp. 205–212.

[101] Chalk, C. R., Neal, T., Harris, T., Pritchard, F. E., and Woodcock, R. J. *Background Information and User Guide for Mil-F-8785B (ASG),'Military Specification-Flying Qualities of Piloted Airplanes'*. Tech. rep. CORNELL AERONAUTICAL LAB INC BUFFALO NY, 1969.

[102] Ryan, T. and Kim, H. J. “LMI-based gain synthesis for simple robust quadrotor control”. In: *IEEE Transactions on Automation Science and Engineering* 10.4 (2013), pp. 1173–1178.

[103] Hoyle, D., Hyde, R. A., and Limebeer, D. J. “An H_∞ approach to two degree of freedom design”. In: *[1991] Proceedings of the 30th IEEE Conference on Decision and Control*. IEEE. 1991, pp. 1581–1585.

[104] Mokhtari, A., Benallegue, A., and Daachi, B. “Robust feedback linearization and GH_∞ controller for a quadrotor unmanned aerial vehicle”. In: *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE. 2005, pp. 1198–1203.

[105] Xi, Z., Lin, L., CHEN, Z.-j., and DUAN, H.-b. “Validation of flight control law based on LFT and structured singular value”. In: *Chinese Journal of Aeronautics* 20.1 (2007), pp. 60–65.

[106] Mystkowski, A. “Implementation and investigation of a robust control algorithm for an unmanned micro-aerial vehicle”. In: *Robotics and Autonomous Systems* 62.8 (2014), pp. 1187–1196.

[107] Raffo, G. V., Ortega, M. G., and Rubio, F. R. “An integral predictive/nonlinear H_∞ control structure for a quadrotor helicopter”. In: *Automatica* 46.1 (2010), pp. 29–39.

[108] Raffo, G. V., Ortega, M. G., and Rubio, F. R. “Path Tracking of a UAV via an Underactuated H_∞ Control Strategy”. In: *European Journal of Control* 17.2 (2011), p. 194.

[109] Ryll, M., Bülthoff, H. H., and Giordano, P. R. "Modeling and control of a quadrotor UAV with tilting propellers". In: *2012 IEEE International Conference on Robotics and Automation*. IEEE. 2012, pp. 4606–4613.

[110] Falconi, R. and Melchiorri, C. "Dynamic model and control of an over-actuated quadrotor uav". In: *IFAC Proceedings Volumes* 45.22 (2012), pp. 192–197.

[111] Raffo, G. V., Ortega, M. G., and Rubio, F. R. "Nonlinear H_∞ controller for the quad-rotor helicopter with input coupling". In: *IFAC Proceedings Volumes* 44.1 (2011), pp. 13834–13839.

[112] Aliyu, M. *Nonlinear H-infinity control, Hamiltonian systems and Hamilton-Jacobi equations*. CRC Press, 2017.

[113] Giernacki, W., Horla, D., Báča, T., and Saska, M. "Real-Time Model-Free Minimum-Seeking Autotuning Method for Unmanned Aerial Vehicle Controllers Based on Fibonacci-Search Algorithm". In: *Sensors* 19.2 (2019), p. 312.

[114] Niermeyer, P., Raffler, T., and Holzapfel, F. "Open-loop quadrotor flight dynamics identification in frequency domain via closed-loop flight testing". In: *AIAA Guidance, Navigation, and Control Conference*. 2015, p. 1539.

[115] Achtelik, M., Bierling, T., Wang, J., Höcht, L., and Holzapfel, F. "Adaptive control of a quadcopter in the presence of large/complete parameter uncertainties". In: *Infotech@ Aerospace 2011*. 2011, p. 1485.

[116] Sumantri, B., Uchiyama, N., Sano, S., and Kawabata, Y. "Robust tracking control of a Quad-Rotor helicopter utilizing sliding mode control with a nonlinear sliding surface". In: *Journal of System Design and Dynamics* 7.2 (2013), pp. 226–241.

[117] Ma, S. H., Shao, P. Y., and Wu, C. F. "LPV Based Robust Gain–Scheduling Control for Transient Mode of Morphing UAV". In: *Advanced Materials Research*. Vol. 622. Trans Tech Publ. 2013, pp. 1368–1372.

[118] Skogestad, S. and Postlethwaite, I. *Multivariable feedback control: analysis and design*. Vol. 2. Wiley, New York. 2007.

[119] Paw, Y. C. and Balas, G. J. "Development and application of an integrated framework for small UAV flight control development". In: *Mechatronics* 21.5 (2011), pp. 789–802.

[120] Samar, R. and Postlethwaite, I. "Design and implementation of a digital multimode H_∞ controller for the Spey turbofan engine". In: *Journal of dynamic systems, measurement, and control* 132.1 (2010).

[121] Miller, K. "Path tracking control for quadrotor helicopters". In: *Computing & Control Engineering Journal* (2008).

[122] Skogestad, S. and Grimholt, C. "The SIMC method for smooth PID controller tuning". In: *PID Control in the Third Millennium*. Springer, 2012, pp. 147–175.

[123] Luukkonen, T. "Modelling and control of quadcopter". In: *Independent research project in applied mathematics, Espoo, Finland* Vol.: 22 (2011).

[124] Sabatino, F. "Quadrotor control: modeling, nonlinear control design, and simulation". MA thesis. KTH, School of Electrical Engineering (EES), Automatic Control, 2015. URL: http://urn.kb.se/resolve?urn=urn:nbn:se:kth:diva-175380.

[125] Diebel, J. "Representing attitude: Euler angles, unit quaternions, and rotation vectors". In: *Matrix* 58.15-16 (2006), pp. 1–35.

[126] Mellinger, D., Lindsey, Q., Shomin, M., and Kumar, V. "Design, modeling, estimation and control for aerial grasping and manipulation". In: *Intelligent Robots and Systems (IROS), 2011 IEEE/RSJ International Conference on.* IEEE. 2011, pp. 2668–2673.

[127] Das, A., Subbarao, K., and Lewis, F. "Dynamic inversion with zero-dynamics stabilisation for quadrotor control". In: *IET control theory & applications* 3.3 (2009), pp. 303–314.

[128] Jung, D. and Tsiotras, P. "Inertial attitude and position reference system development for a small UAV". In: *AIAA Infotech@ Aerospace 2007 Conference and Exhibit.* 2007, p. 2763.

[129] Jung, D. and Tsiotras, P. "Modeling and hardware-in-the-loop simulation for a small unmanned aerial vehicle". In: *AIAA Infotech@ Aerospace 2007 Conference and Exhibit.* 2007, p. 2768.

[130] Whitehead, B. and Bieniawski, S. "Model reference adaptive control of a quadrotor UAV". In: *AIAA Guidance, Navigation, and Control Conference.* 2010, p. 8148.

[131] Bresciani, T. "Modelling, identification and control of a quadrotor helicopter". MA thesis. Department of Automatic Control, Lund University, 2008.

[132] Freddi, A., Lanzon, A., and Longhi, S. "A feedback linearization approach to fault tolerance in quadrotor vehicles". In: *IFAC proceedings volumes* 44.1 (2011), pp. 5413–5418.

[133] Lanzon, A., Freddi, A., and Longhi, S. "Flight control of a quadrotor vehicle subsequent to a rotor failure". In: *Journal of Guidance, Control, and Dynamics* 37.2 (2014), pp. 580–591.

[134] Chowdhary, G., Sobers, D. M., Pravitra, C., Christman, H. C., Wu, A., Johnson, E., Hashimoto, H., Ong, C., and Kalghatgi, R. "Integrated guidance navigation and control for a fully autonomous indoor uas". In: *AIAA Guidance, Navigation, and Control Conference.* 2011, p. 6720.

[135] J.B. Brandt R.W. Deters, G. A. and Selig, M. *UIUC Propeller Database, University of Illinois at Urbana-Champaign.* 2015. URL: `http://m-selig.ae.illinois.edu/props/propDB.html` (visited on 10/17/2018).

[136] Stevens, B. L., Lewis, F. L., and Johnson, E. N. "Aircraft control and simulation: dynamics, controls design, and autonomous systems". In: John Wiley & Sons, 2015. Chap. 8. Modeling and Simulation Of Miniature Aerial Vehicles.

[137] Fumagalli, M., Naldi, R., Macchelli, A., Forte, F., Keemink, A. Q., Stramigioli, S., Carloni, R., and Marconi, L. "Physical Interaction with the Environment". In: *IEEE robotics & automation magazine* 1070.9932/14 (2014).

[138] Yanmaz, E., Kuschnig, R., and Bettstetter, C. "Achieving air-ground communications in 802.11 networks with three-dimensional aerial mobility". In: *INFOCOM, 2013 Proceedings IEEE.* IEEE. 2013, pp. 120–124.

[139] Naseer, T., Sturm, J., and Cremers, D. "Followme: Person following and gesture recognition with a quadrocopter". In: *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on.* IEEE. 2013, pp. 624–630.

[140] Wallar, A., Plaku, E., and Sofge, D. A. "Reactive motion planning for unmanned aerial surveillance of risk-sensitive areas". In: *IEEE Transactions on Automation Science and Engineering* 12.3 (2015), pp. 969–980.

[141] Brandt, J. and Selig, M. "Propeller performance data at low reynolds numbers". In: *49th AIAA Aerospace Sciences Meeting including the New Horizons Forum and Aerospace Exposition*. 2011, p. 1255.

[142] Nagaty, A., Saeedi, S., Thibault, C., Seto, M., and Li, H. "Control and navigation framework for quadrotor helicopters". In: *Journal of intelligent & robotic systems* 70.1-4 (2013), pp. 1–12.

[143] Lebovitz, N. *Ordinary Differential Equations*. Mathematics Series. Brooks/-Cole, 1999.

[144] Southall, B., Buxton, B. F., and Marchant, J. A. "Controllability and Observability: Tools for Kalman Filter Design." In: *BMVC*. 1998, pp. 1–10.

[145] Rosenbrock, H. and MacFarlane, A. "State-space and multivariable theory". In: *IEEE Transactions on Systems, Man, and Cybernetics* 2 (1972), pp. 295–296.

[146] Ludlage, J. H. A. "Controllability analysis of industrial processes: Towards the industrial application." PhD thesis. Technische Universiteit Eindhoven, 1999.

[147] de Vries, D. K. and Van den Hof, P. M. J. "Quantification of model uncertainty from data: input design, interpolation, and connection with robust control design specifications". In: *1992 American Control Conference*. 1992, pp. 3170–3175. DOI: 10.23919/ACC.1992.4792733.

[148] Xiao-lu, R., Chang-hong, W., Guo-xing, Y., Wang, Y., and Han-yu, S. "Robust fault-tolerant method in the ducted fan UAV attitude control". In: *IET Conference Proceedings*. The Institution of Engineering & Technology. 2012.

[149] Ton, C. T. and Mackunis, W. "Robust attitude tracking control of a quadrotor helicopter in the presence of uncertainty". In: *2012 IEEE 51st IEEE Conference on Decision and Control (CDC)*. IEEE. 2012, pp. 937–942.

[150] Elruby, A., El-Khatib, M., El-Amary, N., and Hashad, A. "Dynamic modeling and control of quadrotor vehicle". In: *The International Conference on Applied Mechanics and Mechanical Engineering*. Vol. 15. 15th International Conference on Applied Mechanics and Mechanical Engineering. Military Technical College. 2012, pp. 1–10.

[151] Trirattanawananon, N., Panomruttanarug, B., Higuchi, K., and Mora-Camino, F. "Simulation and experimental study on attitude control of quadrotor". In: *2014 IEEE International Conference on Robotics and Biomimetics (ROBIO 2014)*. IEEE. 2014, pp. 1719–1724.

[152] Thomas, J., Currie, J., and Wilson, D. "Automatic weight selection for mixed sensitivity optimization control of a quadrotor". In: *2017 24th International Conference on Mechatronics and Machine Vision in Practice (M2VIP)*. IEEE. 2017, pp. 1–6.

[153] Thomas, J., Currie, J., and Wilson, D. "Loop Shaping Design Procedure for Quadrotor Control with Weights Designed by Resolving a Constrained Nonlinear Optimization Problem". In: *2017 Australasian Conference on Robotics and Automation(ACRA)*. ARAA. 2017, pp. 1–7.

[154] Envirobotics Limted. *Drone Fishing With an Aerokontiki*. 2020. URL: https://www.aerokontiki.com/ (visited on 08/22/2020).

[155] Lieslehto, J. "MIMO controller design using SISO controller design methods". In: *IFAC Proceedings Volumes* 29.1 (1996), pp. 1152–1156.

[156] Koivo, H. and Tanttu, J. "Tuning of PID Conrollers: Survey of SISO and MIMO Techniques". In: *Intelligent tuning and adaptive control*. Elsevier, 1991, pp. 75–80.

[157] Ebeid, E., Skriver, M., and Jin, J. "A survey on open-source flight control platforms of unmanned aerial vehicle". In: *2017 Euromicro Conference on Digital System Design (DSD)*. IEEE. 2017, pp. 396–402.

[158] Yang, Z., Lin, F., and Chen, B. M. "Survey of autopilot for multi-rotor unmanned aerial vehicles". In: *IECON 2016-42nd Annual Conference of the IEEE Industrial Electronics Society*. IEEE. 2016, pp. 6122–6127.

[159] Zakian, V. and Al-Naib, U. "Design of dynamical and control systems by the method of inequalities". In: *Proceedings of the Institution of Electrical Engineers*. Vol. 120. 11. IET. 1973, pp. 1421–1427.

[160] Currie, J. *Inverse Problem Limited*. 2015. URL: https://www.inverseproblem.co.nz/OPTI/ (visited on 10/11/2019).

[161] Curi, S., Mas, I., and Pena, R. S. "Autonomous flight of a commercial quadrotor". In: *IEEE Latin America Transactions* 12.5 (2014), pp. 853–858.

[162] Julkananusart, A. and Nilkhamhang, I. "Quadrotor tuning for attitude control based on double-loop PID controller using fictitious reference iterative tuning (FRIT)". In: *Industrial Electronics Society, IECON 2015-41st Annual Conference of the IEEE*. IEEE. 2015, pp. 004865–004870.

[163] Hardy, G. H. "The mean value of the modulus of an analytic function". In: *Proceedings of the London Mathematical Society* 2.1 (1915), pp. 269–277.

[164] Nikolski, N. *Hardy spaces*. Vol. 179. Cambridge University Press, 2019.

[165] Doyle, J. C. "Synthesis of robust controllers and filters". In: *The 22nd IEEE Conference on Decision and Control*. IEEE. 1983, pp. 109–114.

[166] Balas, G., Chiang, R., Packard, A., and Safonov, M. "Robust control toolbox". In: *For Use with Matlab. User's Guide, Version* 3 (2005).

[167] Currie, J., Wilson, D. I., Sahinidis, N., and Pinto, J. "OPTI: Lowering the barrier between open source optimizers and the industrial MATLAB user". In: *Foundations of computer-aided process operations* 24 (2012), p. 32.

[168] Belotti, P., Kirches, C., Leyffer, S., Linderoth, J., Luedtke, J., and Mahajan, A. "Mixed-integer nonlinear optimization". In: *Acta Numerica* 22 (2013), pp. 1–131.

[169] Qi, J., Sun, K., and Kang, W. "Optimal PMU placement for power system dynamic state estimation by using empirical observability Gramian". In: *IEEE Transactions on power Systems* 30.4 (2015), pp. 2041–2054.

[170] Luo, W., Tay, W. P., and Leng, M. "How to identify an infection source with limited observations". In: *IEEE Journal of Selected Topics in Signal Processing* 8.4 (2014), pp. 586–597.

[171] Abdelnasser, A., Hossain, E., and Kim, D. I. "Tier-aware resource allocation in OFDMA macrocell-small cell networks". In: *IEEE Transactions on Communications* 63.3 (2015), pp. 695–710.

[172] Coleman, T., Branch, M. A., and Grace, A. "Optimization toolbox". In: *For Use with MATLAB. User's Guide for MATLAB 5, Version 2, Relaese II* (1999).

[173] Audet, C. and Dennis Jr, J. E. "Mesh adaptive direct search algorithms for constrained optimization". In: *SIAM Journal on optimization* 17.1 (2006), pp. 188–217.

[174] Clarke, F. H. *Optimization and nonsmooth analysis*. Vol. 5. Society for Industrial and Applied Mathematics, 1990.

[175] Berthold, T., Gamrath, G., Gleixner, A. M., Heinz, S., Koch, T., and Shinano, Y. "Solving mixed integer linear and nonlinear problems using the SCIP optimization suite". In: *Research Association of Mathematical Programming (RAMP)* (Jan. 2012).

[176] Audet, C., Béchard, V., and Le Digabel, S. "Nonsmooth optimization through mesh adaptive direct search and variable neighborhood search". In: *Journal of Global Optimization* 41.2 (2008), pp. 299–318.

[177] Shen, S., Mulgaonkar, Y., Michael, N., and Kumar, V. "Vision-Based State Estimation and Trajectory Control Towards High-Speed Flight with a Quadrotor." In: *Robotics: Science and Systems*. Vol. 1. Citeseer. 2013.

[178] Zhang, T., Kang, Y., Achtelik, M., Kuhnlenz, K., and Buss, M. "Autonomous hovering of a vision/IMU guided quadrotor". In: *2009 International Conference on Mechatronics and Automation*. IEEE. 2009, pp. 2870–2875.

[179] Powers, C., Mellinger, D., Kushleyev, A., Kothmann, B., and Kumar, V. "Influence of aerodynamics and proximity effects in quadrotor flight". In: *Experimental robotics*. Springer. 2013, pp. 289–302.

[180] Mellinger, D., Michael, N., and Kumar, V. "Trajectory generation and control for precise aggressive maneuvers with quadrotors". In: *Experimental Robotics*. Springer. 2014, pp. 361–373.

[181] Abu-Khalaf, M., Chen, R., and Turevskiy, A. "PID control design made easy". In: *Matlab Digest* (2009).

[182] Glover, K. and McFarlane, D. "Robust stabilization of normalized coprime factor plant descriptions with H_∞ bounded uncertainty". In: *IEEE transactions on automatic control* 34.8 (1989), pp. 821–830.

[183] McFarlane, D. C. and Glover, K. *Robust controller design using normalized coprime factor plant descriptions*. Vol. 138. Springer, 1990.

[184] Zhou, K., Doyle, J. C., and Glover, K. *Robust and optimal control*. Vol. 40. Prentice Hall New Jersey, 1996.

[185] Prempain, E. and Postlethwaite, I. "Static H_∞ loop shaping control of a fly-by-wire helicopter". In: *Automatica* 41.9 (2005), pp. 1517–1528.

[186] Mohamed, M. K. and Lanzon, A. "Design and control of novel tri-rotor UAV". In: *Control (CONTROL), 2012 UKACC International Conference on*. IEEE. 2012, pp. 304–309.

[187] Rich, M., Elia, N., and Jones, P. “Design and implementation of an H_∞ controller for a quadrotor helicopter”. In: *Control & Automation (MED), 2013 21st Mediterranean Conference on*. IEEE. 2013, pp. 1189–1198.

[188] El-Sakkary, A. “The gap metric: Robustness of stabilization of feedback systems”. In: *IEEE Transactions on Automatic Control* 30.3 (1985), pp. 240–247.

[189] Vinnicombe, G. *Uncertainty and Feedback: H_∞ Loop-shaping and the ν-gap Metric*. World Scientific, 2001.

[190] Vinnicombe, G. “Measuring robustness of feedback systems”. PhD thesis. University of Cambridge, 1992.

[191] Zhou, K. and Doyle, J. C. *Essentials of robust control*. Vol. 104. Prentice hall Upper Saddle River, NJ, 1998.

[192] Bode, H. W. et al. “Network analysis and feedback amplifier design”. In: (1945).

[193] Åström, K. J. “Control system design lecture notes for me 155a”. In: *Department of Mechanical and Environmental Engineering University of California Santa Barbara* 333 (2002).

[194] Doyle, J. “Analysis of control systems with structured uncertainty”. In: *IEE Proc. Part D* 129 (1982), pp. 242–250.

[195] Doyle, J. C. “Structured uncertainty in control system design”. In: *1985 24th IEEE Conference on Decision and Control*. IEEE. 1985, pp. 260–265.

[196] Hintze, J. L. and Nelson, R. D. “Violin plots: a box plot-density trace synergism”. In: *The American Statistician* 52.2 (1998), pp. 181–184.

[197] Doherty, J. F. and Mammone, R. J. “A fast method for regularized adaptive filtering”. In: *Digital Signal Processing* 2.1 (1992), pp. 14–26.

[198] Kakar, M. R., Hamzah, M. O., Akhtar, M. N., and Woodward, D. “Surface free energy and moisture susceptibility evaluation of asphalt binders modified with surfactant-based chemical additive”. In: *Journal of cleaner production* 112 (2016), pp. 2342–2353.

[199] Abara, D. N. and Lanzon, A. “H-infinity-PID controller for an open-loop unstable system”. In: *2017 IEEE 3rd International Conference on Electro-Technology for National Development (NIGERCON)*. 2017, pp. 134–138. DOI: 10.1109/NIGERCON.2017.8281886.

[200] Zhou, X., Liu, L., Chen, Z. ji, and Duan, H. bin. “Validation of Flight Control Law Based on LFT and Structured Singular Value”. In: *Chinese Journal of Aeronautics* 20.1 (2007), pp. 60–65. ISSN: 1000-9361.

[201] Palunko, I., Fierro, R., and Cruz, P. “Trajectory generation for swing-free maneuvers of a quadrotor with suspended payload: A dynamic programming approach”. In: *2012 IEEE International Conference on Robotics and Automation*. IEEE. 2012, pp. 2691–2697.

[202] Sreenath, K., Michael, N., and Kumar, V. “Trajectory generation and control of a quadrotor with a cable-suspended load-a differentially-flat hybrid system”. In: *2013 IEEE International Conference on Robotics and Automation*. IEEE. 2013, pp. 4888–4895.

[203] Cruz, P. J. and Fierro, R. "Cable-suspended load lifting by a quadrotor UAV: hybrid model, trajectory generation, and control". In: *Autonomous Robots* 41.8 (2017), pp. 1629–1643.

[204] Faust, A., Palunko, I., Cruz, P., Fierro, R., and Tapia, L. "Learning swing-free trajectories for UAVs with a suspended load". In: *2013 IEEE International Conference on Robotics and Automation*. IEEE. 2013, pp. 4902–4909.

[205] Sreenath, K., Lee, T., and Kumar, V. "Geometric control and differential flatness of a quadrotor UAV with a cable-suspended load". In: *52nd IEEE Conference on Decision and Control*. IEEE. 2013, pp. 2269–2274.

[206] Dai, S., Lee, T., and Bernstein, D. S. "Adaptive control of a quadrotor UAV transporting a cable-suspended load with unknown mass". In: *53rd IEEE Conference on Decision and Control*. IEEE. 2014, pp. 6149–6154.

[207] Cruz, P. J., Oishi, M., and Fierro, R. "Lift of a cable-suspended load by a quadrotor: A hybrid system approach". In: *2015 American Control Conference (ACC)*. IEEE. 2015, pp. 1887–1892.

[208] Sadr, S., Moosavian, S. A. A., and Zarafshan, P. "Dynamics modeling and control of a quadrotor with swing load". In: *Journal of Robotics* 2014 (2014).

[209] Alothman, Y., Guo, M., and Gu, D. "Using iterative LQR to control two quadrotors transporting a cable-suspended load". In: *IFAC-PapersOnLine* 50.1 (2017), pp. 4324–4329.

[210] Fliess, M., Lévine, J., Martin, P., and Rouchon, P. "Flatness and defect of non-linear systems: introductory theory and examples". In: *International journal of control* 61.6 (1995), pp. 1327–1361.

[211] Alothman, Y. and Gu, D. "Quadrotor transporting cable-suspended load using iterative linear quadratic regulator ILQR optimal control". In: *2016 8th Computer Science and Electronic Engineering (CEEC)*. IEEE. 2016, pp. 168–173.

[212] Alothman, Y., Jasim, W., and Gu, D. "Quad-rotor lifting-transporting cable-suspended payloads control". In: *2015 21st International Conference on Automation and Computing (ICAC)*. IEEE. 2015, pp. 1–6.

[213] Zhou, J., Deng, R., Shi, Z., and Zhong, Y. "Robust cascade PID attitude control of quadrotor helicopters subject to wind disturbance". In: *2017 36th Chinese Control Conference (CCC)*. IEEE. 2017, pp. 6558–6563.

[214] Bo, G., Xin, L., Hui, Z., and Ling, W. "Quadrotor helicopter Attitude Control using cascade PID". In: *2016 Chinese Control and Decision Conference (CCDC)*. IEEE. 2016, pp. 5158–5163.

[215] Idres, M, Mustapha, O, and Okasha, M. "Quadrotor trajectory tracking using PID cascade control". In: *Materials Science and Engineering Conference Series*. Vol. 270. 1. 2017, p. 012010.

[216] Wang, L. *PID control system design and automatic tuning using MATLAB/Simulink*. John Wiley & Sons, 2020.

[217] Kim, J., Gadsden, S. A., and Wilkerson, S. A. "A Comprehensive Survey of Control Strategies for Autonomous Quadrotors". In: *Canadian Journal of Electrical and Computer Engineering* 43.1 (2019), pp. 3–16.

[218] Luo, C., McClean, S. I., Parr, G., Teacy, L., and De Nardi, R. "UAV Position Estimation and Collision Avoidance Using the Extended Kalman Filter". In: *IEEE Transactions on Vehicular Technology* 62.6 (2013), pp. 2749–2762. DOI: 10.1109/TVT.2013.2243480.

[219] Vitanov, I. and Aouf, N. "Fault detection and isolation in an inertial navigation system using a bank of unscented H_∞ filters". In: *2014 UKACC International Conference on Control (CONTROL)*. IEEE. 2014, pp. 250–255.

[220] Gośliński, J., Giernacki, W., and Gardecki, S. "Unscented Kalman Filter for an orientation module of a quadrotor mathematical model". In: *2013 9th Asian Control Conference (ASCC)*. IEEE. 2013, pp. 1–6.

[221] Sui, L., Mou, Z., Gan, Y., and Huang, X. "Unscented Kalman filter algorithm with colored noise and its application in spacecraft attitude estimation". In: *The 1st International Workshop on the Quality of Geodetic Observation and Monitoring Systems (QuGOMS'11)*. Springer. 2015, pp. 95–100.

[222] Turner, M. C. and Postlethwaite, I. "A new perspective on static and low order anti-windup synthesis". In: *International Journal of Control* 77.1 (2004), pp. 27–44.

Appendix A

Algorithms

The code snippets from proposed seven algorithms will be listed in this Appendix. While the algorithms are different from each other, only steps from **Algorithm #1** will be fully listed. For the other algorithms, the the cost functions from other will be presented. The other steps can be constructed by examining the steps from **Algorithm #1** and corresponding sections presented in the study. The MATLAB listings are ordered such that they follow the steps from the Algorithms presented in the work.

A.1 Algorithm #1

Step 1: Develop the linear plant and scale the plant G according to the criteria described in section 3.3

Listing A.1: Parameters based on AsTec Pelican Quadrotor

```
% Chowdhary, G. V., et al. Integrated Guidance Navigation and Control
% for a Fully Autonomous Indoor UAS
mass = 1.2703;      % Kg ; Mass
Ixx = 0.04339;      % Kg m^2 ; Moment of Inertia of the quad - x component;
Iyy = 0.04339;      % Kg m^2 ; Moment of Inertia of the quad - y component;
Izz = 0.0705;       % Kg m^2 ; Moment of Inertia of the quad - z component;
dx = 0.194;         % m ; Location of propeller in body x-direction
dy = 0.194;         % m ; Location of propeller in body y-direction
Ct = 0.1;           % Thrust coefficient nominal value
Cp = 0.045;         % Power coefficient nominal value
d = .254;           % m ; Propeller diameter
Ir = 3.375e-5;      % Kg m^2 ; Rotor Inertia
g = 9.81;           % m s^-2  ; Accelaration due to gravity
Ts = 0.01;          % s ; Sample time
```

Listing A.2: Developing the SISO system

```
quad_params = [ mass g dx dy Ixx Iyy Izz Ct Cp d Ir]';
lin_pts = [zeros(15,1)' mass*g ]; % Nominal_Plant_tf

% SISO Sytem function declaration
G_hat = SISO_quad(quad_params,lin_pts);

% The function
function [Plant] = SISO_quad(quad_params,lin_pts)
% Define symbolic parameters
% [ x y z ] -> Position vector in Inertial Frame
% [ phi theta yaw ] -> Euler angles in Inertial Frame
% [ x_dot y_dot z_dot ] -> Velocity vector in Inertial Frame
% [ p q r ] -> Angular veloctiy vector (equal to [phi_dot theta_dot ...
%                       ... yaw_dot] at small angles)

syms x y z phi theta yaw x_dot y_dot z_dot p q r u1 u2 u3 u4 f1 f2 f3 f4

% Initialize quadrotor parameters
mass = quad_params(1); g = quad_params(2); dx = quad_params(3);
dy = quad_params(4) ; Ixx = quad_params (5) ; Iyy = quad_params(6);
Izz = quad_params(7); Ct = quad_params(8); Cp = quad_params(9);
d = quad_params(10); Ir = quad_params(11);

% States, inputs and outputs
X = [ x y z phi theta yaw x_dot y_dot z_dot p q r ];   % state vector
U = [ u1 u2 u3 u4 ];                                   % input vector
Y = [ p q r z_dot ];                                   % output vector

rho = 1.225; Dia = 0.254;        % Deriving propeller angular velcotiy
F2U = [ 1 -1 -1 1 ; 1 1 -1 -1 ; 1 -1 1 -1 ; 1 1 1 1 ];F = F2U\U';
n = (sqrt(F(1)) - sqrt(F(2)) + sqrt(F(3)) - sqrt(F(4)))/(Ct*rho*Dia^4);

% Quadrotor equations of motion
x_d = x_dot;
y_d = y_dot;
z_d = z_dot;
phi_d = p + (q*sin(phi) + r*cos(phi))*tan(theta);
theta_d = q*cos(phi) - r*sin(phi);
yaw_d = (q*sin(phi) + r*cos(phi))*sec(theta);
x_dot_d = -(u4/mass)*(sin(phi)*sin(yaw) + cos(phi)*cos(yaw)*sin(theta));
y_dot_d = -(u4/mass)*(cos(phi)*sin(yaw)*sin(theta) - cos(yaw)*sin(phi));
z_dot_d = g -(u4/mass)*(cos(phi)*cos(theta));
p_d = 1/(Ixx)*(u1*dy + (Iyy - Izz)*q*r - Ir*q*n);
q_d = 1/(Iyy)*(u2*dx + (Izz - Ixx)*p*r + Ir*p*n);
r_d = 1/(Izz)*(u3*d*(Cp/Ct) + (Ixx - Iyy)*p*q);

% non-linear system, dX(t)/dt = g(X,U,t), y(t) = h(X,U,t)
G = [ x_d y_d z_d phi_d theta_d yaw_d ...
      x_dot_d y_dot_d z_dot_d p_d q_d r_d]';
H = [ p q r z_dot ]';

% compute jacobian
A.symbolic = jacobian(G, X);
B.symbolic = jacobian(G, U);
C.symbolic = jacobian(H, X);
D.symbolic = jacobian(H, U);

% linearizing plant at lin_pnt
A.algebraic = simplify(subs(A.symbolic, {x y z phi theta ...
```

```
        yaw x_dot y_dot z_dot p q r u1 u2 u3 u4},lin_pts));
B.algebraic = simplify(subs(B.symbolic, {x y z phi theta ...
        yaw x_dot y_dot z_dot p q r u1 u2 u3 u4},lin_pts));
C.algebraic = simplify(subs(C.symbolic, {x y z phi theta ...
        yaw x_dot y_dot z_dot p q r u1 u2 u3 u4},lin_pts));
D.algebraic = simplify(subs(D.symbolic, {x y z phi theta ...
        yaw x_dot y_dot z_dot p q r u1 u2 u3 u4},lin_pts));

% compute numerical values
A.eval = eval(A.algebraic);
B.eval = eval(B.algebraic);
C.eval = eval(C.algebraic);
D.eval = eval(D.algebraic);

% linearized system
Plant = ss(A.eval, B.eval, C.eval, D.eval);
return
```

Listing A.3: Scaling the system

```
%% Scaling the nominal system matrix - dividing each variable by ...
%               ... its maximum expected or allowed change
% Maximum expected changes
Du = diag([9.6184 9.6184 9.6184 11.4744]);% Hover propeller RPM - 4690...
%                     ...Max propeller RPM 6500. Change 6500-4690 = 1810.
De = diag([0.25 0.25 0.25 0.5]); % Max allowed error- 0.25 rad/s ...
%              ... for angualr velocity & 0.5 m/s for altitude climb rate
Dd = diag([0.5 0.5 0.5 0.5]);  % Max expected change in disturbance - ...
%     ... 0.5 rad/s for angular veloctiy & 0.5 m/s for altitude climb rate
Dr = diag([0.5 0.5 0.5 1.5]); % Max expected change in ref setpoint - ...
%     ... 0.5 rad/s for angular veloctiy & 1.5 m/s for altitude climb rate

% Scaled system matrix
G = De^-1*G_hat*Du; G_d = c2d(G,Ts);
Gd = De^-1*eye(4)*Dd;
R = De^-1*Dr;
```

Listing A.4: SISO systems

```
% TF from Input U1 to roll Output
U1_tf_to_roll = minreal(G(1,1));

% TF from Input U2 to theta Output
U2_tf_to_theta = minreal(G(2,2));

% TF from Input U3 to yaw Output
U3_tf_to_yaw = minreal(G(3,3));

% TF from Input U4 to altitude Output
U4_tf_to_altitude = minreal(G(4,4));
```

Step 2: Calculate the complimentary sensitivity weight, by capturing the uncertainty in the model.

Listing A.5: Capturing Parameter Uncertainty

```
% Parameter Uncertainty is captured by generating 'm' different ...
%                      ... uncertain plants around the nominal plant
s = tf('s');
m = 100; Uc_G_array = s * zeros(4,4,m);
PUc_I = 10; PUc_Ct = 38; PUc_Cp = 30;
PUc_dx = 30; PUc_dy = 30; PUc_mass = 40;

U_mass = ureal('U_mass',mass,'Percentage',PUc_mass);
U_mass_sample =  usample(U_mass,m);
U_mass_sample = reshape(U_mass_sample,[m 1]);

U_dx = ureal('U_dx',dx,'Percentage',PUc_dx);
U_dx_sample =  usample(U_dx,m);
U_dx_sample = reshape(U_dx_sample,[m 1]);

U_dy = ureal('U_dy',dy,'Percentage',PUc_dy);
U_dy_sample =  usample(U_dy,m);
U_dy_sample = reshape(U_dy_sample,[m 1]);

U_Ixx = ureal('U_Ixx',Ixx,'Percentage',PUc_I);
U_Ixx_sample =  usample(U_Ixx,m);
U_Ixx_sample = reshape(U_Ixx_sample,[m 1]);

U_Iyy = ureal('U_Iyy',Iyy,'Percentage',PUc_I);
U_Iyy_sample =  usample(U_Iyy,m);
U_Iyy_sample = reshape(U_Iyy_sample,[m 1]);

U_Izz = ureal('U_Izz',Izz,'Percentage',PUc_I);
U_Izz_sample =  usample(U_Izz,m);
U_Izz_sample = reshape(U_Izz_sample,[m 1]);

U_Ct = ureal('U_Ct',Ct,'Percentage',PUc_Ct);
U_Ct_sample =  usample(U_Ct,m);
U_Ct_sample = reshape(U_Ct_sample,[m 1]);

U_Cp = ureal('U_Cp',Cp,'Percentage',PUc_Cp);
U_Cp_sample =  usample(U_Cp,m);
U_Cp_sample = reshape(U_Cp_sample,[m 1]);

for i = 1:m
    Uc_quad_params = [U_mass_sample(i) g U_dx_sample(i) U_dy_sample(i) ...
                      U_Ixx_sample(i) U_Iyy_sample(i) U_Izz_sample(i)...
                      U_Ct_sample(i) U_Cp_sample(i) d Ir]';
    Uc_G_hat = SISO_quad_ss2(Uc_quad_params,lin_pts);
    Uc_G_array(:,:,i) = minreal(De^-1*Uc_G_hat*Du);
end

% Develop complimentary senstivity weight to capture the uncertainty
% Roll and Pitch rate
[Uc_roll_param,Uc_info_roll_param] = ucover(Uc_G_array(1,1,:), ...
                                     G(1,1),1,'InputMult');
Wi_roll = Uc_info_roll_param.W1;

% Yaw rate
[Uc_yaw_param,Uc_info_yaw_param] = ucover(Uc_G_array(3,3,:), ...
                                   G(3,3),1,'InputMult');
Wi_yaw = Uc_info_yaw_param.W1;

% Altitude rate
```

```
[Uc_alt_param,Uc_info_alt_param] = ucover(Uc_G_array(4,4,:), ...
                                   G(4,4),1,'InputMult');
Wi_alt = Uc_info_alt_param.W1;
```

Step 3: Define the data structure necessary for the optimization framework as well the form of complimentary sensitivity weight w_P.

Listing A.6: Data structures to develop the optimization framework

```
% Generate reference and disturbance signals
Tot_time = 10; Time_step = Tot_time/5; Ts = 0.01;
Step_r = [];
Ref = [ 1 0 -1 0 0 ];
for j = 1:5
    Step_r = vertcat(Step_r,Ref(j)*ones(floor(Time_step*(1/Ts)),1));
end

Step_d = [];
Dist = [ 0 1 0 -1 0 ];
for j = 1:5
    Step_d = vertcat(Step_d,Dist(j)*ones(floor(Time_step*(1/Ts)),1));
end

% Structure containing the data
Roll_data = struct('Plant',U1_tf_to_roll,'Comp_sens_wt', Wi_roll, ...
            'Ref_wt', R(1,1), 'Dist_wt', Gd(1,1), 'Ref_sig', Step_r, ...
            'Dist_sig', Step_d, 'Sample_time', Ts);
Yaw_data = struct('Plant',U3_tf_to_yaw,'Comp_sens_wt', Wi_yaw, ...
           'Ref_wt', R(3,3), 'Dist_wt', Gd(3,3), 'Ref_sig', Step_r, ...
           'Dist_sig', Step_d, 'Sample_time', Ts);
Alt_data = struct('Plant',U4_tf_to_altitude,'Comp_sens_wt', Wi_alt, ...
           'Ref_wt', R(4,4), 'Dist_wt', Gd(4,4), 'Ref_sig', Step_r, ...
           'Dist_sig', Step_d, 'Sample_time', Ts);
```

Listing A.7: w_P form definition

```
% Wp form definition
% Wp = a (s + b)
%      --------    where b > c   ; a high pass filter
%       (s + c)

% Lower and upper bounds lb <= x <= ub & x0 for the optimisation framework
lb = [0.001 0.01 0.01];
ub = [100 200 100];
x0 = [1 28 2];
```

Step 4: Define the cost function $\mathcal{J}$ and initialize the weights W1 through W4.

Listing A.8: Sensitivity Weight & Controller Design :- Roll rate

```
% Cost function declaration
fun = @(x)foptHinf_SISO_MS_S_T(x,Roll_data);

% Cost function definition
```

```
function [j] = foptHinf_SISO_MS_S_T(x,Data)

% Wp form definition
% Wp = a (s + b)
%      --------    where b > c   ; a high pass filter
%       (s + c)
%
s = tf('s');
Wp = x(1)*((s + x(2))/(s + x(3)));

try
    [K,~,GAMA,~] = mixsyn(Data.Plant,Wp,[],Data.Comp_sens_wt);

    % Discretize the controller and the plant
    K_d = c2d(K,Data.Sample_time);G_d = c2d(Data.Plant,Data.Sample_time);

    % Model setup
    %                                                   |
    %                                                   | d
    %                                                ------- 
    %                                               | Gd_ss |
    %                                                ------- 
    %                                                   | dout
    %                                                   |
    % r   ______ rout    e    _____   u    _____  y_G_d |   y
    % -- | R_ss | --> + ---- | K_d | ---- | G_d | ----> + ---->
    %     ------    - |       -----        -----        |
    %                 |                                 |
    %                  ---------------------------------
    %
    R_ss = ss([],[],[],Data.Ref_wt); Gd_ss = ss([],[],[],Data.Dist_wt);
    R_ss.InputName = 'r'; R_ss.OutputName = 'rout';
    Sum_E = sumblk('e = - y + rout',1);
    K_d.InputName = 'e'; K_d.OutputName = 'u';
    G_d.InputName = 'u'; G_d.OutputName = 'y_G_d';
    Gd_ss.InputName = 'd'; Gd_ss.OutputName = 'dout';
    Sum_D = sumblk('y = y_G_d + dout',1);
    T = connect(R_ss,Sum_E,K_d,G_d,Gd_ss,Sum_D,{'d' ; 'r'},'y');
    sys = idss(T.A,T.B,T.C,T.D,'Ts',Data.Sample_time);

    % simulate & generate 'y' for reference input
    udata = [Data.Dist_sig Data.Ref_sig];
    y = sim(sys,udata);
    t = 0:Data.Sample_time:((length(udata)/ ...
        (1/Data.Sample_time))-Data.Sample_time);
    e = y - Data.Ref_sig*Data.Ref_wt;
    K_model = idss(K_d.A,K_d.B,K_d.C,K_d.D,'Ts',Data.Sample_time);
    u = sim(K_model,e);
    ctrl_effort = max(abs(u));

    % Calculate j
    e_area = sum(trapz(t,(t'.* (abs(e(:,:)).^2))));
    j = (W1*abs(e_area)+ W2*order(K_model)) + W3*GAMA + W4*ctrl_effort;
catch
    j =10^8;
    disp('Good Luck next time')
end
return
```

Step 5: Develop the vector $\mathbf{\Phi}$ which contains the following nonlinear closed loop functionals.

1. Maximum overshoot, M_p
2. Rise time, t_p
3. Settling time, t_s
4. Maximum value of input, u_{max}
5. γ.

Here M_p, tp and ts, the step response characteristics and u_{max} are the time domain specification while γ refers to the frequency domain specification

Listing A.9: Vector containing the closed loop functionals

```
% Nonlinear Constraints (cl <= nlcon(x) <= cu) a.k.a Closed loop ...
%                                                          ... functionals
nlcon = @(x)[ Clp_overshoot(x,Roll_data) ;
              Clp_risetime(x,Roll_data) ;
              Clp_settlingtime(x,Roll_data) ;
              Clp_ctrl_effort(x,Roll_data) ;
              Clp_gama(x,Roll_data) ];
```

The first three functions can be developed by using the MATLAB command `stepinfo` while the fourth and fifth function can be developed by modifying the Listing A.8.

Step 6: Define the frequency domain bounds ϵ_γ and time domain specifications limit vector $\boldsymbol{\epsilon}$.

Listing A.10: Vector containing the closed loop functionals

```
% Constraints
cl = [-1; 0; 0; 0; .25];
cu = [5; .5; 1 ; 1; 1.0];
```

Step 7: Implement the constrained non-linear optimization problem. This is achieved using the OPTI MATLAB toolbox. The black box optimizer NOMAD is chosen as the optimization routine. Nonlinear constraints are defined based on ϵ_γ and $\boldsymbol{\epsilon}$ for the vector $\mathbf{\Phi}$ that defines the closed loop functionals.

```
% Define the solver to be used -> NOMAD in OPTI MATLAB
opts = optiset('solver','nomad','display','iter', ...
       'maxtime',1000,'tolafun',1e-4);

Opt = opti('fun',fun ,'nl',nlcon,cl,cu,'bounds',lb,ub, ...
      'x0',x0,'options',opts);
tic
[x,~] = solve(Opt,x0);  % solve for x
toc
```

The controller can now be developed once the optimization problem is resolved.

Listing A.11: Develop the SISO Roll rate controller

```
% Develop the controller
s = tf('s');
Wp_roll = x(1)*((s + x(2))/(s + x(3)));
[K_roll,CL_roll,GAM_roll,~] = mixsyn(U1_tf_to_roll,Wp_roll,[],Wi_roll);
```

Further tuning and improvements (**Step 8** and **9**) can now be undertaken as needed once the preliminary tests are performed with the controller.

A.2 Cost Functions

In this section cost functions from various algorithms presented in the study will be listed out.

Algorithm #2 – **Step 4**:

Listing A.12: SISO LSDP 1 DOF - Cost function definition

```
% Loop shaping weight and pref-filter form definition
% W1 = W1_a(s + W1_b) ;
%      -----------
%        s + W1_c
% W2 = W2_a(s + W2_b) ;
%        -----------
%          s + W2_c
% Ks0 = Ks0;
% and x = [ W1_a W1_b W1_c W2_a  W2_b  W2_c Ks0]'
W1_a= x(1); W1_b= x(2); W1_c= x(3);
W2_a= x(4); W2_b= x(5); W2_c= x(6);
Ks0=x(7);

s = tf('s');
W1 = W1_a*(s + W1_b)/ (s + W1_c);
W2 = W2_a*(s + W2_b)/ (s + W2_c);

try
    [K,~,GAMA,~]=ncfsyn(-Data.Plant,W1,W2);

    % Discretize the controller and the plant
    K_d = c2d(K,Data.Sample_time);G_d = c2d(Data.Plant,Data.Sample_time);

% Model setup
%                                                                   |
%                                                                   | d
%                                                               -------
%                                                              | Gd_ss |
%                                                               -------
```

```
%                                                                  | dout
%                                                                  |
% r   ______  rs  ________ rout     e    _____   u    _____  y_G_d |   y
% -- | R_ss | -- | Ks0_ss | ---> + ---- | K_d | ---- | G_d | ----> + ---->
%     ------      --------     - |       -----        -----        |
%                                |                                 |
%                                 ---------------------------------
%
    R_ss = ss([],[],[],Data.Ref_wt); Gd_ss = ss([],[],[],Data.Dist_wt);
    Ks0_ss = ss([],[],[],Ks0);
    R_ss.InputName = 'r'; R_ss.OutputName = 'rs';
    Ks0_ss.InputName = 'rs'; Ks0_ss.OutputName = 'rout';
    Sum_E = sumblk('e = - y + rout',1);
    K_d.InputName = ''e'; K_d.OutputName = 'u';
    G_d.InputName = 'u'; G_d.OutputName = 'y_G_d';
    Gd_ss.InputName = 'd'; Gd_ss.OutputName = 'dout';
    Sum_D = sumblk('y = y_G_d + dout',1);
    T = connect(R_ss,Ks0_ss,Sum_E,K_d,G_d,Gd_ss,Sum_D,{'d' ; 'r'},'y');
    sys = idss(T.A,T.B,T.C,T.D,'Ts',Data.Sample_time);

    % simulate & generate 'y' for reference input
    udata = [Data.Dist_sig Data.Ref_sig];
    y = sim(sys,udata);
    t = 0:Data.Sample_time:((length(udata)/ ...
        (1/Data.Sample_time))-Data.Sample_time);

    e = y - Data.Ref_sig*(Ks0*Data.Ref_wt);

    K_model = idss(K_d.A,K_d.B,K_d.C,K_d.D,'Ts',Data.Sample_time);
    u = sim(K_model,e);
    ctrl_effort = max(abs(u));

    % Calculate j
    e_area = sum(trapz(t,(t'.* (abs(e(:,:)).^2))));
    j = (W1*abs(e_area)+ W2*order(K_model)) + W3*GAMA + W4*ctrl_effort;
catch
    j =10^8;
    disp('Good Luck next time')
end
return
```

Algorithm #3 – Step 4:

Listing A.13: SISO LSDP 2 DOF - Cost function definition

```
% Loop shaping weights form definition
function [j] = foptHinf_LS_2DOF(x,Data)
% W1 = (W1_a*s + W1_b)
%      ---------------
%      (W1_c*s + W1_d)
% W2 = W2_a;
% Rho
% and x = [ W1_a  W1_b  W1_c  W1_d  rho  Wi]'
W1_a= x(1); W1_b= x(2); W1_c= x(3); W1_d= x(4); rho=x(5); Wi = x(6);
s = tf('s');
W1 = (W1_a*s + W1_b)/ ((W1_c*s + W1_d + 1));
Gs = Data.Plant*W1;
try
```

```
    [K,GAMA] = Lsdp2dof(Gs,Data.Tref,rho);
    [A,B,C,D] = ssdata(K);
    A = real(A); B = real(B); C = real(C); D = real(D);
    K1 = ss(A,B(:,1),C,D(:,1));
    K2 = ss(A,B(:,2),C,D(:,2));

    % Discretize the controller and the plant
    K_d = c2d(K,Data.Sample_time); K1_d = c2d(K1,Data.Sample_time);
    K2_d = c2d(K2,Data.Sample_time);
    G_d = c2d(Data.Plant,Data.Sample_time);

% Model setup
%                                                                   |
%                                                                   | d
%                                                               -------
%                                                              | Gd_ss |
%                                                               -------
%                                                                   | dout
%                                                                   |
% r  ______ rs ____ rw  ______ rk      uw  ______  u   _____ y_G_d  |   y
% --| R_ss |--| Wi |---| K1_d |---> + ----| W1_d |----| G_d |---->  + ---->
%    ------     ----     ------    + |       ------      -----        |
%                                    | yk    ______                   |
%                                     -----| K2_d |------------------
%                                          ------
    R_ss = ss([],[],[],Data.Ref_wt,Data.Sample_time);
    Gd_ss = ss([],[],[],Data.Dist_wt,Data.Sample_time);
    Wi_ss = ss([],[],[],Wi,Data.Sample_time);
    W1_d = c2d(W1,Data.Sample_time);
    R_ss.InputName = 'r'; R_ss.OutputName = 'rs';
    Wi_ss.InputName = 'rs'; Wi_ss.OutputName = 'rw';
    K1_d.InputName = 'rw'; K1_d.OutputName = 'rk';
    Sum_E = sumblk('uw = yk + rk',1);
    W1_d.InputName = 'uw'; W1_d.OutputName = 'u';
    G_d.InputName = 'u'; G_d.OutputName = 'y_G_d';
    Gd_ss.InputName = 'd'; Gd_ss.OutputName = 'dout';
    Sum_D = sumblk('y = y_G_d + dout',1);
    K2_d.InputName = 'y'; K2_d.OutputName = 'yk';
    T = connect(R_ss,Wi_ss,K1_d,Sum_E,W1_d,G_d,Gd_ss, ...
                Sum_D,K2_d,{'d' ; 'r'},'y','u');
    sys = idss(T.A,T.B,T.C,T.D,'Ts',Data.Sample_time);

    % simulate & generate 'y' for reference input
    udata = [Data.Dist_sig*0 Data.Ref_sig];
    y = sim(sys,udata); t = 0:Data.Sample_time:((length(udata)/ ...
        (1/Data.Sample_time))-Data.Sample_time);
    e = y - Data.Ref_sig*Data.Ref_wt;
    K_model = idss(K_d.A,K_d.B,K_d.C,K_d.D,'Ts',Data.Sample_time);

    % determine input signal 'input'
    udata = [Data.Dist_sig Data.Ref_sig];
    rd2u = getIOTransfer(T,{'r','d'},'u');
    u_sys = idss(rd2u.A,rd2u.B,rd2u.C,rd2u.D,'Ts',Data.Sample_time);
    u = sim(u_sys,udata);
    ctrl_effort = max(abs(u));

    % Calculate j
    e_area = sum(trapz(t,(t'.* (abs(e(:,:)).^2))));
    j = (W1*abs(e_area)+ W2*order(K_model)) + W3*GAMA + W4*ctrl_effort;
catch
j =10^8;
```

```
disp('Good Luck next time')
end
return
```

Listing A.14: Lsdp2dof function definition

```
function [K,GAM] = Lsdp2dof(Gs,Tref,rho) %#ok<*ASGLU>
[As,Bs,Cs,Ds] = ssdata(balreal(Gs));
[Ar,Br,Cr,Dr] = ssdata(Tref);
[nr,nr] = size(Ar); [lr,mr] = size(Dr);
[ns,ns] = size(As); [ls,ms] = size(Ds);
Rs = eye(ls) + Ds*Dr' ; Ss = eye(ms) + Ds'*Dr;
A = (As - (Bs/(Ss))*Ds'*Cs);
B = sqrtm((Cs'/Rs)*Cs);
Q = (Bs/Ss)*Bs';
[Zs,L,~,REPORT] = care(A,B,Q);
A = blkdiag(As,Ar);
B1 = [zeros(ns,mr) ((Bs*Ds')+(Zs*Cs'))/(sqrtm(Rs));
Br zeros(nr,ls)];
B2 = [Bs ; zeros(nr,ms)];
C1 = [zeros(ms,ns+nr); Cs zeros(ls,nr);rho*Cs -rho*rho*Cr];
C2 = [zeros(mr,ns+nr); Cs zeros(ls,nr)];
D11 = [zeros(ms,mr+ls); zeros(ls,mr) sqrtm(Rs); -rho*rho*Dr rho*sqrtm(Rs)];
D12 = [eye(ms) ; Ds; rho*Ds];
D21 = [rho*eye(mr) zeros(mr,ls); zeros(ls,mr) sqrtm(Rs)];
D22 = [zeros(mr,ms);Ds];
B = [B1 B2]; C = [C1;C2]; D = [D11 D12;D21 D22];
P = ss(A,B,C,D);
[l1,m2] = size(D12); [l2,m1] = size(D21);
nmeas = l2; ncon = m2; gmin = 1;gmax = 100; gtol = .01;
[K,CLP,GAM] = hinfsyn(P,nmeas,ncon,'GMIN',gmin,'GMAX', ...
              gmax,'TOLGAM',gtol,'DISPLAY','off');
return
```

Algorithm #4 – Step 4:

Listing A.15: SISO μ - Cost function definition

```
% Sensitivity weight form definition
function [j] = MuSyn(x,Data)
% Wp = a (s + b)
%       --------    where b > c    ; a high pass filter
%        (s + c)
%
% and x = [ a  b  c ]'
s = tf('s');
G = Data.Plant; %#ok<*NASGU>
Wp = x(1)*((s + x(2))/(s + x(3)));
Wi = Data.Comp_sens_wt;
omega = logspace(-3,3,61);
try
    % Generalized Plant P
    systemnames = 'G Wp Wi';
    inputvar = '[udel(1); w(1); u(1)]';
    outputvar = '[Wi; Wp; -G-w]';
    input_to_G = '[u+udel]';
    input_to_Wp = '[G+w]';
```

```
    input_to_Wi = '[u]';
    sysoutname = 'P'; cleanupsysic = 'yes';
    sysic;
    P = minreal(ss(P));
    Delta = ultidyn('D_1',[1 1]);
    Punc = lft(Delta,P);
    opt = dkitopt('FrequencyVector', omega);
    [K,clp,bnd,dkinfo] = dksyn(Punc,1,1,opt);

    % Discretize the controller and the plant
    K_d = c2d(K,Data.Sample_time);
    G_d = c2d(Data.Plant,Data.Sample_time);

% Model setup
%                                                   |
%                                                   | d
%                                                -------
%                                               | Gd_ss |
%                                                -------
%                                                   | dout
%                                                   |
% r   ______ rout    e    _____   u    _____  y_G_d |   y
% -- | R_ss | --> + ---- | K_d | ---- | G_d | ----> + ---->
%     ------    - |       -----        -----        |
%                 |                                 |
%                  ---------------------------------
%
    R_ss = ss([],[],[],Data.Ref_wt);
    Gd_ss = ss([],[],[],Data.Dist_wt);
    R_ss.InputName = 'r'; R_ss.OutputName = 'rout';
    Sum_E = sumblk('e = - y + rout',1);
    K_d.InputName = 'e'; K_d.OutputName = 'u';
    G_d.InputName = 'u'; G_d.OutputName = 'y_G_d';
    Gd_ss.InputName = 'd'; Gd_ss.OutputName = 'dout';
    Sum_D = sumblk('y = y_G_d + dout',1);
    T = connect(R_ss,Sum_E,K_d,G_d,Gd_ss,Sum_D,{'d' ; 'r'},'y');
    sys = idss(T.A,T.B,T.C,T.D,'Ts',Data.Sample_time);

    % simulate & generate 'y' for reference input
    udata = [Data.Dist_sig Data.Ref_sig];
    y = sim(sys,udata);
    t = 0:Data.Sample_time:((length(udata)/ ...
        (1/Data.Sample_time))-Data.Sample_time);
    e = y - Data.Ref_sig*Data.Ref_wt;
    K_model = idss(K_d.A,K_d.B,K_d.C,K_d.D,'Ts',Data.Sample_time);
    u = sim(K_model,e);
    ctrl_effort = max(abs(u));

    gama = bnd;

    % Calculate j
    e_area = sum(trapz(t,(t'.* (abs(e(:,:)).^2))));
    j = (W1*abs(e_area)+ W2*order(K_model)) + W3*bnd + W4*ctrl_effort;
catch
j =10^8;
disp('Good Luck next time')
end
return
```

Similarly the Cost function definitions for the multivariable controller development frameworks.

Algorithm #5 – Step 4:

Listing A.16: MIMO MSO cost function definition

```
function [j] = foptHinf_MIMO_MSO(x,Data,Wi)
% Performance Weight 'Wp' form definition
% Wp_i = a_i (s + b_i)
%         ----------    where b_i > c_i ...
%          (s + c_i)    ... a high pass filter where i = 1:4
% Wp = [ Wp_1    0     0      0
%        0      Wp_2   0      0
%        0       0    Wp_3    0
%        0       0     0     Wp_4 ]
s = tf('s');
Wp_1 = x(1)*((s + x(2))/(s + x(3)));
Wp_2 = x(4)*((s + x(5))/(s + x(6)));
Wp_3 = x(7)*((s + x(8))/(s + x(9)));
Wp_4 = x(10)*((s + x(11))/(s + x(12)));
Wp = [ Wp_1    0     0      0
        0     Wp_2   0      0
        0      0    Wp_3    0
        0      0     0     Wp_4 ];
try
    [K,~,GAMA,~] = mixsyn(Data.Plant,Wp,[],Wi);

    % Discretize the controller and the plant
    K_d = c2d(K,Data.Sample_time);
    G_d = c2d(Data.Plant,Data.Sample_time);

% Model setup
%                                                    |
%                                                    | d
%                                                 --------
%                                                | Gd_ss |
%                                                 --------
%                                                    | dout
%                                                    |
% r    ______ rout    e    _____    u    _____  y_G_d |    y
% -- | R_ss | --> + ---- | K_d | ---- | G_d | ----> + ---->
%      ------    - |       -----        -----        |
%                  |                                 |
%                   ---------------------------------
%
    R_ss = ss([],[],[],Data.Ref_wt); Gd_ss = ss([],[],[],Data.Dist_wt);
    R_ss.InputName = 'r'; R_ss.OutputName = 'rout';
    Sum_E = sumblk('e = - y + rout',4);
    K_d.InputName = 'e'; K_d.OutputName = 'u';
    G_d.InputName = 'u'; G_d.OutputName = 'y_G_d';
    Gd_ss.InputName = 'd'; Gd_ss.OutputName = 'dout';
    Sum_D = sumblk('y = y_G_d + dout',4);
    T = connect(R_ss,Sum_E,K_d,G_d,Gd_ss,Sum_D,{'d' ; 'r'},'y');
    sys = idss(T.A,T.B,T.C,T.D,'Ts',Data.Sample_time);

    % simulate & generate 'y' for reference input
    udata = [Data.Dist_sig Data.Ref_sig];
    y = sim(sys,udata);
```

```
    t = 0:Data.Sample_time:((length(udata)/ ...
        (1/Data.Sample_time))-Data.Sample_time);
    e = y - Data.Ref_sig*Data.Ref_wt;
    K_model = idss(K_d.A,K_d.B,K_d.C,K_d.D,'Ts',Data.Sample_time);
    u = sim(K_model,e);
    ctrl_effort = norm(max(abs(u)),inf);
    % Calculate j
    e_area = sum(trapz(t,(t'.* (abs(e(:,:)).^2))));
    j = (W1*abs(e_area)+ W2*order(K_model)) + W3**GAMA + W4*ctrl_effort;
catch
j =10^8;
disp('Good Luck next time')
end
return
```

Algorithm #6 – Step 4:

Listing A.17: MIMO LSDP 1 DOF - Cost function definition

```
function [j] = coprimeunc_MIMO(x,Data)
% Loop Shaping Design Weight form defintions.
% W1 = diag( [ (a1*s + b1)/(c1*s + d1) ;
%              (a2*s + b2)/(c2*s + d2) ;
%              (a3*s + b3)/(c3*s + d3) ;
%              (a4*s + b4)/(c4*s + d4) ]
% W2 = diag( [ w2a1 w2a2 w3a3 w4a4 ]);
% Ks0 = diag( [ks1 ks2 ks3 ks4 ]);
% and x = [ a1 a2 a3 a4 ...
%           b1 b2 b3 b4 ...
%           c1 c2 c3 c4 ...
%           d1 d2 d3 d4 ...
%           w2a1 w2a2 w3a3 w4a4 ...
%           ks1 ks2 ks3 Ks4 ]

x = reshape(x,[4 6])';
s = tf('s');
W1 = s* diag(ones(4,1));
W2 = diag(zeros(4,1));
Ks0 = diag(zeros(4,1));
for i = 1:4
    for j = 1:4
        if i==j
            W1(i,j) = (x(1,j)*s + x(2,j))/(x(3,j)*s + x(4,j));
            W2(i,j) = x(5,j);
            Ks0(i,j) = x(6,j);
        end
    end
end

try
    [K,~,GAMA,~]=ncfsyn(-Data.Plant,W1,W2);

    % Discretize the controller and the plant
    K_d = c2d(K,Data.Sample_time);
    G_d = c2d(Data.Plant,Data.Sample_time);

% Model setup
%                                                           |
```

```
%                                                                  | d
%                                                              -------
%                                                             | Gd_ss |
%                                                               -------
%                                                                  | dout
%                                                                  |
% r   ______  rs  ________ rout     e    _____   u    _____  y_G_d |   y
% -- | R_ss | -- | Ks0_ss | ---> + ---- | K_d | ---- | G_d | ----> + ---->
%     ------      --------     - |       -----        -----        |
%                                |                                 |
%                                 ---------------------------------
%
    R_ss = ss([],[],[],Data.Ref_wt);
    Gd_ss = ss([],[],[],Data.Dist_wt);
    Ks0 = ss([],[],[],Ks0);
    R_ss.InputName = 'r'; R_ss.OutputName = 'rs';
    Ks0.InputName = 'rs'; Ks0.OutputName = 'rout';
    Sum_E = sumblk('e = - y + rout',4);
    K_d.InputName = 'e'; K_d.OutputName = 'u';
    G_d.InputName = 'u'; G_d.OutputName = 'y_G_d';
    Gd_ss.InputName = 'd'; Gd_ss.OutputName = 'dout';
    Sum_D = sumblk('y = y_G_d + dout',4);
    T = connect(R_ss,Ks0,Sum_E,K_d,G_d,Gd_ss,Sum_D,{'d' ; 'r'},'y');
    sys = idss(T.A,T.B,T.C,T.D,'Ts',Data.Sample_time);

    % simulate & generate 'y' for reference input
    udata = [Data.Dist_sig Data.Ref_sig];
    y = sim(sys,udata);
    t = 0:Data.Sample_time:((length(udata)/ ...
        (1/Data.Sample_time))-Data.Sample_time);
    e = y - Data.Ref_sig*Data.Ref_wt;
    K_model = idss(K_d.A,K_d.B,K_d.C,K_d.D,'Ts',Data.Sample_time);
    u = sim(K_model,e);
    ctrl_effort = norm(max(abs(u)),inf);

    % Calculate j
    e_area = sum(trapz(t,(t'.* (abs(e(:,:)).^2))));
    j = (W1*abs(e_area)+ W2*order(K_model) + W3*GAMA + W4*ctrl_effort);
catch
    j =10^8;
    disp('Good Luck next time')
end
return
```

Algorithm #7 – Step 4:

Listing A.18: MIMO μ - Cost function definition

```
function [j] = MuSyn_MIMO(x,Data,Wi)
% Performance Weight 'Wp' form definiton
% Wp_i = a_i (s + b_i)
%         ----------    where b_i > c_i
%          (s + c_i)    a high pass filter where i = 1:4
% Wp = [ Wp_1    0     0      0
%        0      Wp_2   0      0
%        0       0    Wp_3    0
%        0       0     0     Wp_4 ]
s = tf('s');
```

```
G = Data.Plant; %#ok<*NASGU>
Wp_1 = x(1)*((s + x(2))/(s + x(3)));
Wp_2 = x(4)*((s + x(5))/(s + x(6)));
Wp_3 = x(7)*((s + x(8))/(s + x(9)));
Wp_4 = x(10)*((s + x(11))/(s + x(12)));
Wp = [ Wp_1    0     0      0
        0     Wp_2   0      0
        0      0    Wp_3    0
        0      0     0     Wp_4 ];
omega = logspace(-3,3,61);

try
    % Generalized plant P
    systemnames = 'G Wp Wi';
    inputvar = '[ydel(4); w(4); u(4)]';
    outputvar = '[Wi ; Wp; -G-u]';
    input_to_G = '[u+ydel]';
    input_to_Wp = '[G+w]';
    input_to_Wi = '[u]';
    sysoutname = 'P';
    cleanupsysic = 'yes';
    sysic;
    P = minreal(ss(P));
    Delta = ultidyn('D_1',[4 4]);
    Punc = lft(Delta,P);
    opt = dkitopt('FrequencyVector', omega);
    [K,clp,bnd,dkinfo] = dksyn(Punc,4,4,opt);

    % Discretize the controller and the plant
    K_d = c2d(K,Data.Sample_time);
    G_d = c2d(Data.Plant,Data.Sample_time);

% Model setup
%                                                     |
%                                                     | d
%                                                  -------
%                                                 | Gd_ss |
%                                                  -------
%                                                     | dout
%                                                     |
% r    ______ rout    e    _____    u    _____  y_G_d |    y
% -- | R_ss | --> + ---- | K_d | ---- | G_d | ----> + ---->
%      ------    - |       -----        -----         |
%                  |                                  |
%                   ----------------------------------
%
    R_ss = ss([],[],[],Data.Ref_wt);
    Gd_ss = ss([],[],[],Data.Dist_wt);
    R_ss.InputName = 'r'; R_ss.OutputName = 'rout';
    Sum_E = sumblk('e = - y + rout',4);
    K_d.InputName = 'e'; K_d.OutputName = 'u';
    G_d.InputName = 'u'; G_d.OutputName = 'y_G_d';
    Gd_ss.InputName = 'd'; Gd_ss.OutputName = 'dout';
    Sum_D = sumblk('y = y_G_d + dout',4);
    T = connect(R_ss,Sum_E,K_d,G_d,Gd_ss,Sum_D,{'d' ; 'r'},'y');
    sys = idss(T.A,T.B,T.C,T.D,'Ts',Data.Sample_time);

    % simulate & generate 'y' for reference input
    udata = [Data.Dist_sig Data.Ref_sig];
    y = sim(sys,udata);
    t = 0:Data.Sample_time:((length(udata)/ ...
```

```
        (1/Data.Sample_time))-Data.Sample_time);
    e = y - Data.Ref_sig*Data.Ref_wt;
    K_model = idss(K_d.A,K_d.B,K_d.C,K_d.D,'Ts',Data.Sample_time);
    u = sim(K_model,e);
    ctrl_effort = norm(max(abs(u)),inf);

    gama = bnd;

    % Calculate j
    e_area = sum(trapz(t,(t'.* (abs(e(:,:)).^2))));
    j = (W1*abs(e_area)+ W2*order(K_model)) + W3*gama + W4*ctrl_effort;
catch
    j =10^8;
    disp('Good Luck next time')
end
return
```

Appendix B

SIMULINK Diagrams

In this section, the SIMULINK layouts followed throughout the study are presented. The figures presented are those from SISO MSO controlled plant. The layout is similar for all the systems except for the replacement in the controllers used.

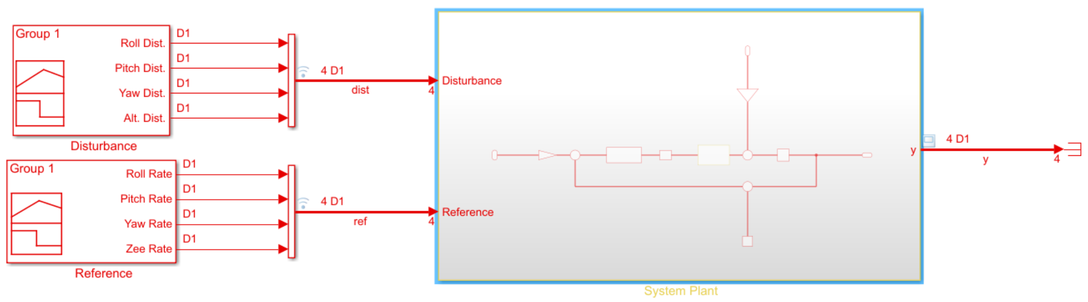


Figure B.1: Disturbance and Reference Signals

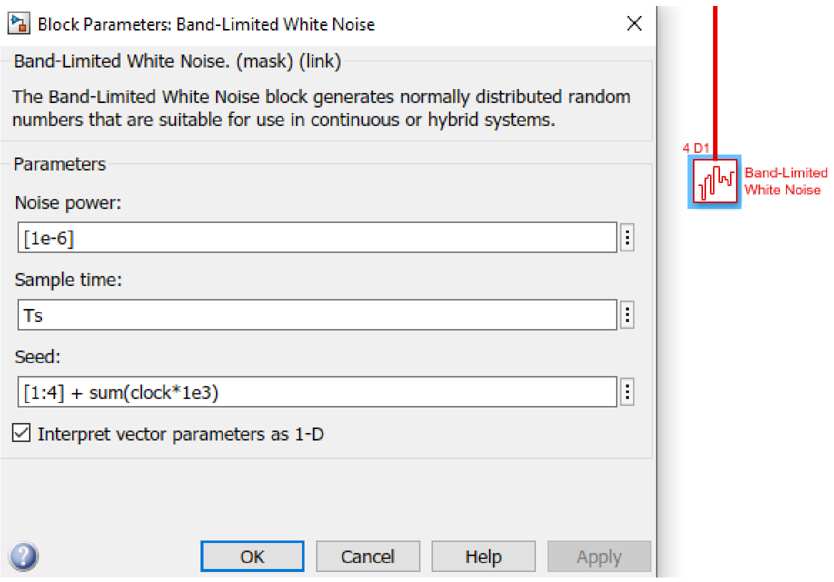


Figure B.2: The Band-Limited White Noise signal block

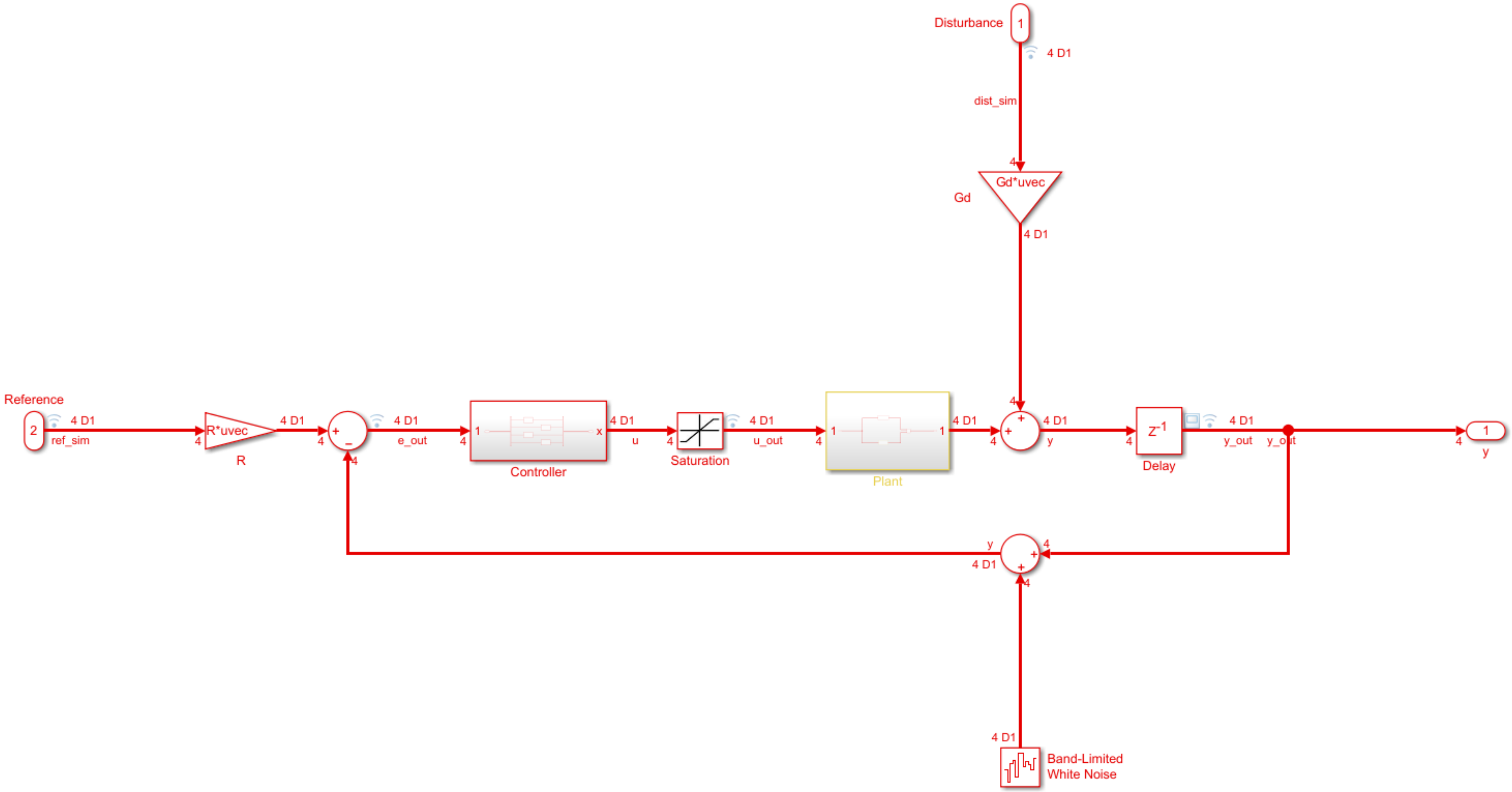


Figure B.3: The general layout

The 'Band-Limited White Noise' block (Figure B.2) with this specific value of the variable 'Seed' enables random white noise to be injected into the system in SIMULINK. The block have been used in Chapter 6 for the case study. The controller block (Figure B.4) changes as the different controllers are used. In some cases there might also be pre-filters added to the system, but the general layout stays the same.

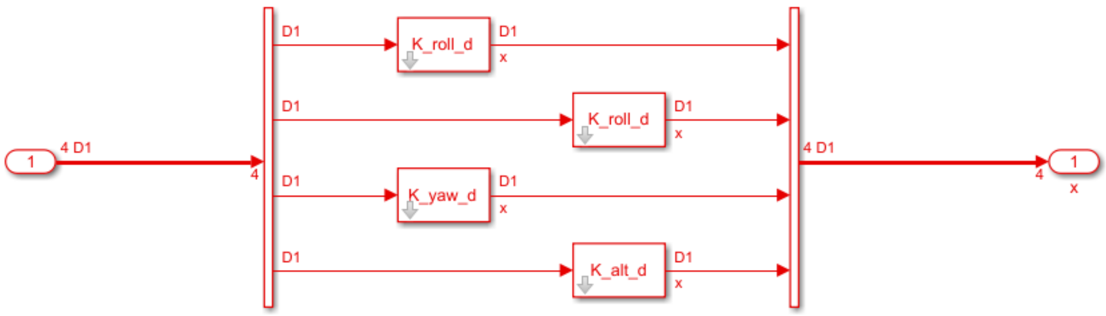


Figure B.4: The Controller Block

Inside the 'Plant' block, options are presented to select between the Linearized plant or Non-Linear plant. The Non-Linear plant block is presented in Fig. B.6.

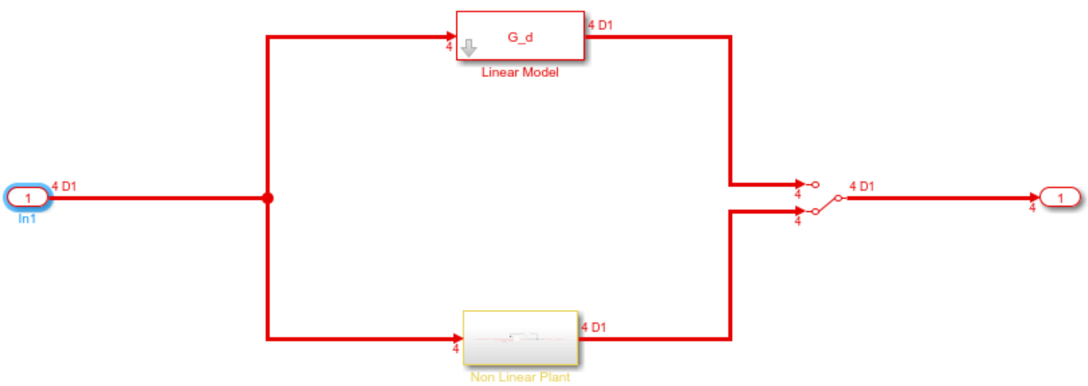


Figure B.5: The 'Plant' Block containing both a non-linear and linearized version of the plant model.

In the 'Non Linear Plant' block B.6 the options to select between the nominal plant parameters and that for the plant with model uncertainty can be seen. These parameters are used when model with parameter uncertainty are to be tested. The 'Saturation' block B.7 makes sure that the input do not overshoot the practically realizable limits.

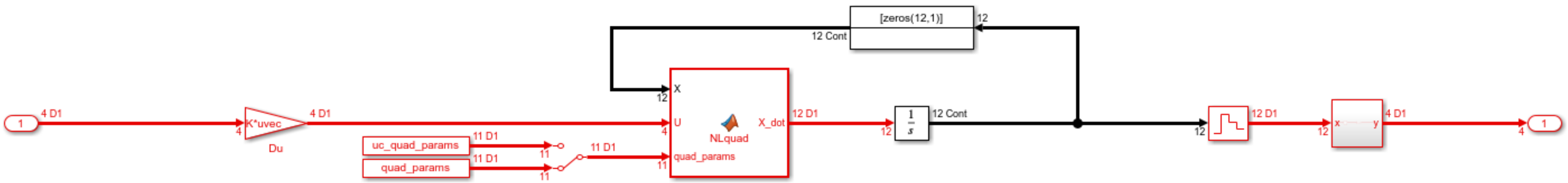


Figure B.6: The 'Non Linear Plant' Block

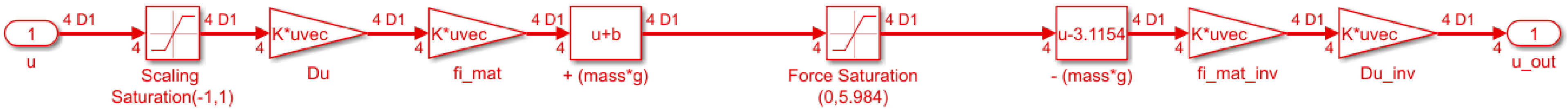


Figure B.7: The 'Saturation' Block

The Non Linear plant equation present in the ‘NLquad’ is presented in Listing B.1

Listing B.1: Non-Linear plant model

```
function X_dot = NLquad(X,U,quad_params)
x= X(1); y= X(2); z= X(3); phi= X(4); theta= X(5); yaw= X(6); x_dot= X(7);
y_dot=X(8); z_dot=X(9); p= X(10); q= X(11); r= X(12);

mass = quad_params(1); g = quad_params(2); dx = quad_params(3);
dy = quad_params(4) ; Ixx = quad_params (5) ; Iyy = quad_params(6);
Izz = quad_params(7); Ct = quad_params(8); Cp = quad_params(9);
d = quad_params(10); Ir = quad_params(11);

U_f = U + [zeros(3,1); 1.2703*g ];
u1= U_f(1); u2= U_f(2); u3= U_f(3); u4= U_f(4);

rho = 1.225; Dia = 0.254;  % Deriving propeller angular velcotiy
F2U = [ 1 -1 -1 1 ; 1 1 -1 -1 ; 1 -1 1 -1 ; 1 1 1 1 ];F = abs(F2U\U_f);
n = (sqrt(F(1)) - sqrt(F(2)) + sqrt(F(3)) - sqrt(F(4)))/(Ct*rho*Dia^4);

% Quadrotor equations of motion
x_d = x_dot;
y_d = y_dot;
z_d = z_dot;
phi_d = p + (q*sin(phi) + r*cos(phi))*tan(theta);
theta_d = q*cos(phi) - r*sin(phi);
yaw_d = (q*sin(phi) + r*cos(phi))*sec(theta);
x_dot_d = -(u4/mass)*(sin(phi)*sin(yaw) + cos(phi)*cos(yaw)*sin(theta));
y_dot_d = -(u4/mass)*(cos(phi)*sin(yaw)*sin(theta) - cos(yaw)*sin(phi));
z_dot_d = g -(u4/mass)*(cos(phi)*cos(theta));
p_d = 1/(Ixx)*(u1*dy + (Iyy - Izz)*q*r - Ir*q*n);%
q_d = 1/(Iyy)*(u2*dx + (Izz - Ixx)*p*r + Ir*p*n);
r_d = 1/(Izz)*(u3*d*(Cp/Ct) + (Ixx - Iyy)*p*q);

X_dot = [ x_d y_d z_d phi_d theta_d yaw_d ...
          x_dot_d y_dot_d z_dot_d p_d q_d r_d]';
end
```

Appendix C

Additional Code Snippets

In this chapter important portions of the code from the work presented in the study are included. In Listing C.1, C.2 and C.3 capturing plant-non linearity and uncertainty due to time delays and absorbing it into the complimentary sensitivity weight is presented.

(Note: MATLAB code and SIMULINK files used in the study can be downloaded from the link : `https://1drv.ms/u/s!AspeQzJQstwqqWIozR3FWTbR_MIq?e=mOs3s6`)

Listing C.1: Capturing non-linearity

```
% Non linearity is to be captured by linearizing the quadrotor at
% 'n' points around the hover point
n = 100;

% angles taken from -10 to +10 . Generate 'n' sample points.
Eu_angle_range = deg2rad([-10,10]);
u_phi = ureal('u_phi',0,'Range',Eu_angle_range);
u_phi_sample =  usample(u_phi,n);
u_phi_sample = reshape(u_phi_sample,[n 1]);

u_theta = ureal('u_theta',0,'Range',Eu_angle_range);
u_theta_sample =  usample(u_theta,n);
u_theta_sample = reshape(u_theta_sample,[n 1]);

u_yaw = ureal('u_yaw',0,'Range',Eu_angle_range);
u_yaw_sample =  usample(u_yaw,n);
u_yaw_sample = reshape(u_yaw_sample,[n 1]);

% angular rates -0.1 rad/s to +0.1 rad/s. Generate 'n' sample points.
AngVel_range = [-.1,0.1];
u_p = ureal('u_p',0,'Range',AngVel_range);
u_p_sample =  usample(u_p,n);
u_p_sample = reshape(u_p_sample,[n 1]);

u_q = ureal('u_q',0,'Range',AngVel_range);
u_q_sample =  usample(u_q,n);
u_q_sample = reshape(u_q_sample,[n 1]);
```

```
u_r = ureal('u_r',0,'Range',AngVel_range);
u_r_sample =  usample(u_r,n);
u_r_sample = reshape(u_r_sample,[n 1]);

% linear velocity -.25 m/s to +.25 m/s. Generate 'n' sample points.
LinVel_range = [-.25,.25];
u_u = ureal('u_u',0,'Range',LinVel_range);
u_u_sample =  usample(u_u,n);
u_u_sample = reshape(u_u_sample,[n 1]);

u_v = ureal('u_v',0,'Range',LinVel_range);
u_v_sample =  usample(u_v,n);
u_v_sample = reshape(u_v_sample,[n 1]);

u_w = ureal('u_w',0,'Range',LinVel_range);
u_w_sample =  usample(u_w,n);
u_w_sample = reshape(u_w_sample,[n 1]);

% rotor RPM inthe range -> 4490 - 4690 - 4890
RPM_Range = [4490,4890];
u_Omega_1 = ureal('u_Omega_1',4690,'Range',RPM_Range);
u_Omega_1_sample =  usample(u_Omega_1,n);
u_Omega_1_sample = reshape(u_Omega_1_sample,[n 1]);

u_Omega_2 = ureal('u_Omega_2',4690,'Range',RPM_Range);
u_Omega_2_sample =  usample(u_Omega_2,n);
u_Omega_2_sample = reshape(u_Omega_2_sample,[n 1]);

u_Omega_3 = ureal('u_Omega_3',4690,'Range',RPM_Range);
u_Omega_3_sample =  usample(u_Omega_3,n);
u_Omega_3_sample = reshape(u_Omega_3_sample,[n 1]);

u_Omega_4 = ureal('u_Omega_4',4690,'Range',RPM_Range);
u_Omega_4_sample =  usample(u_Omega_4,n);
u_Omega_4_sample = reshape(u_Omega_4_sample,[n 1]);

% linearize at n different points
clear Non_linear_G_vector;
for i = 1:n
    lin_pts = [ 0 0 0 ...
                u_phi_sample(i) u_theta_sample(i) u_yaw_sample(i) ...
                u_u_sample(i) u_v_sample(i) u_w_sample(i) ...
                u_p_sample(i) u_q_sample(i) u_r_sample(i) ...
                u_Omega_1_sample(i) u_Omega_2_sample(i) ...
                u_Omega_3_sample(i) u_Omega_4_sample(i) ];
    Non_linear_G = tf(MIMO_quad_ss(quad_params,lin_pts));
    Non_linear_G_vector(:,:,i) = De^-1*Non_linear_G*Du;
end
Non_linear_G_vector = frd(Non_linear_G_vector,[6.28,628.3]); % 1 Hz ...
%                                                          ...  to 10 Hz
```

Listing C.2: Capturing time-delay uncertainty

```
% Delay Uncertainty is captured by generating 'o' different
% uncertain plants around the nominal plant
o = 100; Tau = .004; PUc_Tau = 10;

% Uncertainty in the range Tau +-10%.
U_Tau = ureal('U_Tau',Tau,'Percentage',PUc_Tau);
```

```
s = tf('s');
[num,den] = tfdata(G_hat);

for i = 1:o
    Sample_pts = [ usample(U_Tau) 0 0 0 ; ...
                   0 usample(U_Tau) 0 0 ; ...
                   0 0 usample(U_Tau) 0 ; ...
                   0 0 0 usample(U_Tau) ];

    %   Unc_TD_G = tf(num,den,'InputDelay',Sample_pts(:,:));
    Unc_TD_G = tf(num,den,'IODelay',Sample_pts);
    Unc_TD_G_vector(:,:,i) = De^-1*Unc_TD_G*Du; %#ok<SAGROW>
end
```

Listing C.3: Developing the complimentary sensitivity weight

```
% Concatenate the vectors. Here Non_linear_G_vector contains the models
% with parameter uncetainty. The other two vectors are defined above.
NL_TD_Param_Unc_G_vector = cat(3,Unc_TD_G_vector, Unc_G_vector, ...
                           Non_linear_G_vector);

% Structured Parametric, Time Delay Output and Nolinear Input uncertainty
S_NL_TD_Param_Ip_w = s*eye(4);
for j = 1:4
    % 1st order Structured Parametric, Time Delay & Non Linear Output ...
    %                                                         ... uncertainty
    [S_G_NL_TD_Param,S_Info_G_NL_TD_Param] = ...
           ucover(NL_TD_Param_Unc_G_vector(j,:,:),G(j,: ),1,'InputMult');
    S_NL_TD_Param_Ip_w(j,j) = tf(S_Info_G_NL_TD_Param.W1);
end
```

The plots where percentile based shading is used (eg. see Figure 4.24) is developed by modifying the MATLAB file `fanChart.m`. Similarly the Violin plots (see Figure 4.42) is based on the MATLAB file `distributionPlot.m` (click to follow through to the hyperlink).

Appendix D

The Controllers

In order to easily reproduce these results, each of the optimised controllers are listed following the order in which they appear in this thesis. The continuous controller, K, is given in packed matrix form [145], in the units given in chapter 3.

D.1 Chapter 4 – SISO Systems

D.1.1 Pitch and Roll Rate

D.1.1.1 MSO Controller

$$K \cong \left[\begin{array}{ccc|c} -0.7366 & 2.082\cdot 10^{-16} & -1.342\cdot 10^{-13} & 4 \\ 8.197\cdot 10^{-7} & -0.9954 & 0.05844 & -1.206\cdot 10^{-9} \\ 1.874\cdot 10^{4} & 12.84 & -8280 & -2.055\cdot 10^{-8} \\ \hline 435.9 & 0.2985 & -192.5 & 0 \end{array}\right] \tag{D.1}$$

D.1.1.2 LSDP 1 DoF Controller

$$K \cong \left[\begin{array}{cccc|c} -91.88 & -29.89 & -218.8 & 10.41 & 0.1449 \\ 0 & -1.257 & -42.78 & 0.07331 & 0.00102 \\ 0 & 42.75 & -97.04 & -0.5363 & -0.007463 \\ 0 & 0 & 0 & -16.31 & 0.25 \\ \hline -74.21 & -51.59 & -377.6 & 17.97 & 0.2501 \end{array}\right] \tag{D.2}$$

D.1.1.3 LSDP 2 DoF Controller

$$K \cong \left[\begin{array}{ccc|cc} -10.01 & -7.07 & 8.743 & 3.704 \cdot 10^{-20} & 0.004397 \\ -1.287 \cdot 10^{4} & -9089 & 1.124 \cdot 10^{4} & -1.442 \cdot 10^{-13} & 5.636 \\ -6.217 \cdot 10^{-11} & -4.391 \cdot 10^{-11} & -17.64 & 1.909 & 2.747 \cdot 10^{-14} \\ \hline -2268 & -1602 & 1988 & 0 & 0 \end{array}\right] \tag{D.3}$$

D.1.1.4 μ Controller

$$K \cong \left[\begin{array}{ccccc|c} 166.4 & -3.481 & -1.404 & 2.231 & 68.8 & 3.965 \cdot 10^{-17} \\ 323.2 & -112.6 & -7.457 & 4.781 & 130.8 & -25.29 \\ 3.299 \cdot 10^{-7} & -1.094 \cdot 10^{-7} & -0.00125 & 3.133 \cdot 10^{-9} & 1.334 \cdot 10^{-7} & -0.25 \\ 3.026 & -0.06329 & -0.02553 & -0.9552 & 1.251 & 5.594 \cdot 10^{-15} \\ -397.3 & 3.628 & 3.143 & -5.263 & -164.4 & -1.118 \\ \hline 215.9 & -4.515 & -1.821 & 2.871 & 89.22 & 0 \end{array}\right] \tag{D.4}$$

D.1.2 Yaw Rate

D.1.2.1 MSO Controller

$$K \cong \left[\begin{array}{ccc|c} -0.4895 & 8.619 \cdot 10^{-33} & -2.106 \cdot 10^{-31} & 2 \\ 9.469 \cdot 10^{-8} & -0.9954 & 0.08441 & -5.521 \cdot 10^{-9} \\ 333.3 & 4.565 & -111.5 & -6.51 \cdot 10^{-8} \\ \hline 21.37 & 0.2928 & -7.153 & 0 \end{array}\right] \tag{D.5}$$

D.1.2.2 LSDP 1 DoF Controller

$$K \cong \left[\begin{array}{cccc|c} -27.8 & -91.27 & -9.364 & 3.125 & 0.1633 \\ 0 & -33.82 & -6.455 & 0.247 & 0.01291 \\ 0 & -6.444 & -20.96 & 0.0253 & 0.001322 \\ 0 & 0 & 0 & -16.02 & 0.125 \\ \hline 38.64 & -285.2 & -29.26 & 9.764 & 0.5102 \end{array}\right] \tag{D.6}$$

D.1.2.3 LSDP 2 DoF Controller

$$K \cong \left[\begin{array}{ccc|cc} -32.76 & -13.61 & 11.06 & -1.167 \cdot 10^{-15} & 0.02525 \\ -4351 & -1807 & 1470 & -2.823 \cdot 10^{-14} & 3.268 \\ -2.238 \cdot 10^{-12} & -9.295 \cdot 10^{-13} & -31.75 & 3.329 & 1.519 \cdot 10^{-15} \\ \hline -1289 & -535.4 & 438.1 & 0 & 0 \end{array}\right] \tag{D.7}$$

D.1.2.4 μ Controller

$$K \cong \left[\begin{array}{ccccc|c} 129.6 & -17.97 & -4.965 & 2.344 & 128.7 & -1.994 \cdot 10^{-16} \\ 170.4 & -167.5 & -16.77 & 2.07 & 165 & -34.01 \\ 1.794 \cdot 10^{-7} & -1.724 \cdot 10^{-7} & -0.1485 & 1.874 \cdot 10^{-9} & 1.738 \cdot 10^{-7} & -0.5 \\ 1.626 & -0.2254 & -0.06227 & -0.9663 & 1.614 & -1.381 \cdot 10^{-14} \\ -114.5 & -0.5017 & 3.218 & -2.16 & -114.2 & -3.869 \\ \hline 74.96 & -10.39 & -2.871 & 1.341 & 74.39 & 0 \end{array}\right] \tag{D.8}$$

D.1.3 Altitude Rate

D.1.3.1 MSO Controller

$$K \cong \left[\begin{array}{ccc|c} -0.006676 & -6.225 \cdot 10^{-17} & 1.489 \cdot 10^{-15} & 1 \\ 3.454 \cdot 10^{-7} & -1.005 & 0.0434 & -1.409 \cdot 10^{-8} \\ 156.9 & 1.717 & -43.07 & -3.231 \cdot 10^{-7} \\ \hline -17.37 & -0.1901 & 4.738 & 0 \end{array}\right] \tag{D.9}$$

D.1.3.2 LSDP 1 DoF Controller

$$K \cong \left[\begin{array}{cccc|c} -69.22 & -266.5 & -193.8 & -28.09 & -0.1801 \\ 0 & -4.427 & -14.56 & -0.4605 & -0.00291 \\ 0 & 14.44 & -96.12 & 0.3378 & 0.002151 \\ 0 & 0 & 0 & 0.01991 & 0.2415 \\ \hline -113.8 & -531.4 & -384.8 & -56.25 & -0.3394 \end{array}\right] \tag{D.10}$$

D.1.3.3 LSDP 2 DoF Controller

$$K \cong \left[\begin{array}{ccc|cc} -2892 & 1095 & 1876 & -1.164 \cdot 10^{-17} & 1.131 \\ 7756 & -2937 & -5032 & 1.835 \cdot 10^{-14} & -2.99 \\ -1.957 \cdot 10^{-11} & 7.411 \cdot 10^{-12} & -8.818 & 1.448 & 7.849 \cdot 10^{-15} \\ \hline 2549 & -964.9 & -1658 & 0 & 0 \end{array}\right] \tag{D.11}$$

D.1.3.4 μ Controller

$$K \cong \left[\begin{array}{cccccc|c} -0.009512 & 0.002744 & 0.03722 & 0.03774 & 0.0101 & 0.03948 & 0 \\ 0 & -0.01 & 0.06552 & 0.06644 & 0.01779 & 0.0695 & 0 \\ 0 & 0 & -0.1067 & 0.9012 & 0.2413 & 0.9427 & 0 \\ 0 & 0 & 0 & -0.9944 & 0.2447 & 0.9559 & 0 \\ 0 & 0 & 0 & 0 & -3.681 & 0.256 & 0 \\ 0 & 0 & 0 & 0 & 0 & -44.3 & 8 \\ \hline 0.05876 & -0.1035 & -1.403 & -1.423 & -0.381 & -1.488 & 0 \end{array}\right] \tag{D.12}$$

D.2 Chapter 5 – MIMO Systems

D.2.1 MSO Controller

$$K \cong \left[\begin{array}{cccccccccc}
-0.0001 & 5.337 \cdot 10^{-29} & 2.962 \cdot 10^{-28} & -7.969 \cdot 10^{-29} & -1.093 \cdot 10^{-27} & 1.845 \cdot 10^{-31} & 5.951 \cdot 10^{-29} & -2.088 \cdot 10^{-29} & 7.691 \cdot 10^{-29} & -1.034e-26 \\
1.86 \cdot 10^{-11} & -0.1679 & -7.921 \cdot 10^{-17} & -1.264 \cdot 10^{-17} & -1.727 \cdot 10^{-12} & -1.878 \cdot 10^{-14} & 9.761 \cdot 10^{-19} & 3.163 \cdot 10^{-19} & 3.031 \cdot 10^{-17} & -1.617 \cdot 10^{-11} \\
-1.902 \cdot 10^{-9} & 3.447 \cdot 10^{-13} & -0.04651 & -1.824 \cdot 10^{-14} & 1.766 \cdot 10^{-10} & -6.183 \cdot 10^{-14} & -7.615 \cdot 10^{-18} & 4.565 \cdot 10^{-16} & 4.375 \cdot 10^{-14} & 1.653 \cdot 10^{-9} \\
1.23 \cdot 10^{-9} & -7.111 \cdot 10^{-14} & 5.536 \cdot 10^{-15} & -0.0001 & -1.142 \cdot 10^{-10} & 1.275 \cdot 10^{-14} & -6.821 \cdot 10^{-17} & -3.041 \cdot 10^{-16} & -2.915 \cdot 10^{-14} & -1.069 \cdot 10^{-9} \\
0.0003452 & 1.488 \cdot 10^{-14} & 8.631 \cdot 10^{-15} & -1.416 \cdot 10^{-15} & -57.5 & -2.654 \cdot 10^{-15} & -1.15 \cdot 10^{-16} & 4.795 \cdot 10^{-17} & 3.434 \cdot 10^{-15} & 32 \\
7.589 \cdot 10^{-12} & 1.685 \cdot 10^{-7} & -1.049 \cdot 10^{-15} & 7.649 \cdot 10^{-16} & -7.045 \cdot 10^{-13} & -56.64 & 1.293 \cdot 10^{-17} & -1.914 \cdot 10^{-17} & -1.835 \cdot 10^{-15} & -6.596 \cdot 10^{-12} \\
3.897 \cdot 10^{-11} & -1.017 \cdot 10^{-13} & 3.753 \cdot 10^{-08} & 9.624 \cdot 10^{-16} & -3.618 \cdot 10^{-12} & 1.824 \cdot 10^{-14} & -55.28 & -2.409 \cdot 10^{-17} & -2.308 \cdot 10^{-15} & -3.387 \cdot 10^{-11} \\
1.155 \cdot 10^{-11} & -3.074 \cdot 10^{-13} & -2.137 \cdot 10^{-14} & 4.096 \cdot 10^{-8} & -1.072 \cdot 10^{-12} & 5.514 \cdot 10^{-14} & 2.633 \cdot 10^{-16} & -58.11 & 16 & -1.004 \cdot 10^{-11} \\
-1.976 \cdot 10^{-11} & -4.402 \cdot 10^{-14} & 8.389 \cdot 10^{-13} & 14.51 & 1.646 \cdot 10^{-12} & 2.237 \cdot 10^{-14} & -3.809 \cdot 10^{-14} & 3.802 & -15.83 & 1.719 \cdot 10^{-11} \\
8077 & 1.524 \cdot 10^{-9} & -9.459 \cdot 10^{-10} & 6.744 \cdot 10^{-10} & -742.2 & 7.091 \cdot 10^{-11} & -1.891 \cdot 10^{-10} & 2.733 \cdot 10^{-10} & -7.145 \cdot 10^{-10} & -7021 \\
2.249 \cdot 10^{-11} & 420.4 & 1.071 \cdot 10^{-11} & -1.539 \cdot 10^{-12} & -2.284 \cdot 10^{-12} & 1.402 & -7.404 \cdot 10^{-13} & 4.024 \cdot 10^{-13} & -3.115 \cdot 10^{-13} & -1.925 \cdot 10^{-11} \\
5.426 \cdot 10^{-11} & 1.234 \cdot 10^{-14} & 5.147 & 2.377 \cdot 10^{-15} & -5.036 \cdot 10^{-12} & -3.817 \cdot 10^{-15} & 1.034 & 3.993 \cdot 10^{-16} & -4.04 \cdot 10^{-15} & -4.716 \cdot 10^{-11} \\
\hline
244.5 & 12.73 & -2.074 & -1.917 & -22.47 & 0.04245 & -0.4166 & -0.5021 & 1.85 & -212.6 \\
-244.5 & -12.73 & -2.074 & -1.917 & 22.47 & -0.04245 & -0.4166 & -0.5021 & 1.85 & 212.6 \\
-244.5 & 12.73 & 2.074 & -1.917 & 22.47 & 0.04245 & 0.4166 & -0.5021 & 1.85 & 212.6 \\
244.5 & -12.73 & 2.074 & -1.917 & -22.47 & -0.04245 & 0.4166 & -0.5021 & 1.85 & -212.6
\end{array}\right.$$

$$\left.\begin{array}{cc|cccc}
-3.457 \cdot 10^{-29} & -4.411 \cdot 10^{-28} & 4 & 2.566 \cdot 10^{-33} & 5.848 \cdot 10^{-39} & -1.969 \cdot 10^{-38} \\
-8.205 \cdot 10^{-14} & 3.932 \cdot 10^{-16} & -1.259 \cdot 10^{-14} & 4 & 8.538 \cdot 10^{-17} & 1.27 \cdot 10^{-17} \\
-2.701 \cdot 10^{-13} & -3.067 \cdot 10^{-15} & 1.288 \cdot 10^{-12} & -2.486 \cdot 10^{-14} & 1 & 1.833 \cdot 10^{-14} \\
5.572 \cdot 10^{-14} & -2.748 \cdot 10^{-14} & -8.326 \cdot 10^{-13} & 5.127 \cdot 10^{-15} & -5.967 \cdot 10^{-15} & 2 \\
-1.166 \cdot 10^{-14} & -4.298 \cdot 10^{-14} & -2.337 \cdot 10^{-07} & -1.068 \cdot 10^{-15} & -9.347 \cdot 10^{-15} & 1.452 \cdot 10^{-15} \\
16 & 5.207 \cdot 10^{-15} & -5.138 \cdot 10^{-15} & -1.215 \cdot 10^{-8} & 1.131 \cdot 10^{-15} & -7.687 \cdot 10^{-16} \\
7.969 \cdot 10^{-14} & 32 & -2.639 \cdot 10^{-14} & 7.332 \cdot 10^{-15} & -4.045 \cdot 10^{-8} & -9.672 \cdot 10^{-16} \\
2.409 \cdot 10^{-13} & 1.061 \cdot 10^{-13} & -7.818 \cdot 10^{-15} & 2.217 \cdot 10^{e-14} & 2.303 \cdot 10^{-14} & -4.116 \cdot 10^{-8} \\
1.77 \cdot 10^{-14} & -1.001 \cdot 10^{-12} & -2.188 \cdot 10^{-14} & 8.373 \cdot 10^{-16} & 8.65 \cdot 10^{-14} & -1.495 \cdot 10^{-7} \\
-1.007 \cdot 10^{-9} & 1.429 \cdot 10^{-9} & -4.2 \cdot 10^{-7} & -2.246 \cdot 10^{-17} & -1.856 \cdot 10^{-14} & 3.005 \cdot 10^{-15} \\
-272.3 & -1.613 \cdot 10^{-11} & -2.294 \cdot 10^{-14} & -4.301 \cdot 10^{-8} & 2.622 \cdot 10^{-16} & -3.747 \cdot 10^{-16} \\
-8.455 \cdot 10^{-15} & -7.665 & -3.677 \cdot 10^{-14} & -1.14 \cdot 10^{-18} & -6.988 \cdot 10^{-8} & -8.538 \cdot 10^{-16} \\
\hline
-8.245 & 3.088 & 0 & 0 & 0 & 0 \\
8.245 & 3.088 & 0 & 0 & 0 & 0 \\
-8.245 & -3.088 & 0 & 0 & 0 & 0 \\
8.245 & -3.088 & 0 & 0 & 0 & 0
\end{array}\right] \tag{D.13}$$

D.2.2 LSDP 1DOF Controller

$$
K \cong \left[\begin{array}{ccccccccccc|cccc}
-11.9 & 0 & 0 & 0 & -31.52 & 0.3401 & -6.496 & 0.1504 & -0.08653 & 0.01209 & -0.003862 & 0.002508 & -0.003153 & 0.0002533 & -0.000313 \\
0 & -1.528 & 0 & 0 & 11.74 & -0.9519 & 14.65 & -0.01374 & -0.1843 & 0.04385 & 0.001779 & 0.009151 & -0.0115 & 0.000924 & -0.001142 \\
0 & 0 & -29.04 & 0 & -1.788 & -21.25 & 1.573 & 2.624 & 11.1 & -2.528 & -0.08921 & -0.5503 & 0.6918 & -0.05557 & 0.06867 \\
0 & 0 & 0 & -76.59 & -8.508 & -40.26 & -5.194 & -6.216 & -26.15 & 5.912 & 0.2027 & 1.229 & -1.545 & 0.1241 & -0.1534 \\
0 & 0 & 0 & 0 & -542 & 9.037 & -340.9 & -45.84 & 8.276 & 0.379 & 4.399 & -5.105 & -5.872 & 3.66 & 3.489 \\
0 & 0 & 0 & 0 & 8.802 & -81.21 & 20.73 & 1.804 & 3.478 & -12.01 & -0.1946 & 0.3351 & 0.1659 & -3.206 & 2.75 \\
0 & 0 & 0 & 0 & -325 & 20.29 & -355 & -26.55 & 4.099 & 2.463 & -0.2225 & -4.098 & -4.534 & -1.739 & -1.307 \\
0 & 0 & 0 & 0 & 16.17 & -3.871 & -3.846 & -16.57 & -18.39 & 9.553 & 2.232 & 0.141 & -0.2135 & 0.02975 & -0.009456 \\
0 & 0 & 0 & 0 & -9.024 & -16.81 & -3.398 & -17.52 & -88.73 & 41.69 & 0.9349 & 0.6739 & -0.8418 & 0.07067 & -0.09009 \\
0 & 0 & 0 & 0 & 1.005 & 10.15 & -0.4495 & -0.2188 & -1.235 & -30.15 & -0.8307 & 0.09997 & -0.1249 & 0.004565 & -0.007574 \\
0 & 0 & 0 & 0 & -4.222 & 0.3277 & -1.374 & 3.081 & -0.2486 & -0.8603 & -1.85 & -0.0001526 & 0.000409 & -0.0007606 & -0.0004473 \\
\hline
3.237 & 0 & 0 & 0 & -17.04 & 0.1839 & -3.512 & 0.08133 & -0.04678 & 0.006536 & -0.002088 & 0.001356 & -0.001705 & 0.0001369 & -0.0001692 \\
0 & -0.8222 & 0 & 0 & 33.33 & -2.702 & 41.58 & -0.03901 & -0.5233 & 0.1245 & 0.005049 & 0.02598 & -0.03266 & 0.002623 & -0.003242 \\
0 & 0 & 3.023 & 0 & -0.7749 & -9.21 & 0.6818 & 1.137 & 4.813 & -1.096 & -0.03867 & -0.2385 & 0.2999 & -0.02409 & 0.02977 \\
0 & 0 & 0 & 10.67 & -1.138 & -5.384 & -0.6947 & -0.8314 & -3.497 & 0.7907 & 0.0271 & 0.1644 & -0.2066 & 0.0166 & -0.02051
\end{array}\right] \tag{D.14}
$$

D.2.3 μ Controller

$$
K \cong \left[\begin{array}{cccccccccccc|cccc}
-704.1 & 20.44 & -716.6 & 97.1 & -227.1 & 8593 & 0.01914 & -2.441 & -1.633 & 7.073 & -18.37 & -1559 & 6.472\cdot 10^{-8} & 1.105\cdot 10^{-6} & 4.24\cdot 10^{-8} & -324.8 \\
-232.4 & -5581 & 3121 & -431.6 & -966.2 & -3.743\cdot 10^{4} & -0.08337 & -10.39 & -6773 & -81.89 & 21.35 & 73.21 & -1417 & 1.284\cdot 10^{-8} & 8.668\cdot 10^{-7} & 3.206\cdot 10^{-8} \\
-232.6 & -89.15 & -8866 & -422.6 & 990.9 & 3.836\cdot 10^{4} & -0.08351 & -10.41 & -1.548 & -6789 & 71.68 & 30.66 & 1.323\cdot 10^{-8} & -1417 & 5.014\cdot 10^{-8} & 5.519\cdot 10^{-7} \\
-34.98 & -13.27 & 470.8 & -787.5 & 149.2 & -5646 & 0.01243 & -1.567 & 6.15 & 2.115 & -1021 & -3.356 & 2.139\cdot 10^{-7} & 1.318\cdot 10^{-8} & -212.9 & 5.4\cdot 10^{-9} \\
-9.009 & 3890 & 121 & -16.74 & -1.003\cdot 10^{5} & -1451 & -0.003232 & -0.4026 & -264.1 & -1.647 & 2.368 & 1.308 & -54.92 & -4.675\cdot 10^{-9} & 4.456\cdot 10^{-8} & 7.872\cdot 10^{-10} \\
-18.12 & -6.947 & 9349 & -32.93 & 77.22 & -1.195\cdot 10^{5} & -0.006507 & -0.8108 & -3.906 & -525.2 & 1.798 & 6.141 & 9.38\cdot 10^{-8} & -110.4 & 5.415\cdot 10^{-8} & -3.996e-09 \\
-0.02544 & -0.00965 & 0.3424 & 15.43 & 0.1085 & -4.105 & -2.115\cdot 10^{4} & -0.001139 & 0.006685 & 0.003734 & -0.7447 & -0.004639 & 7.659\cdot 10^{-10} & -1.999\cdot 10^{-10} & -0.1548 & -1.174\cdot 10^{-11} \\
3.144 & 0.3739 & -13.11 & 1.776 & -4.155 & 157.2 & 0.0003502 & -225.2 & 0.1353 & 0.2928 & -0.1708 & -28.35 & -3.407\cdot 10^{-10} & 2.027\cdot 10^{-8} & 6.713\cdot 10^{-10} & -5.941 \\
1.312 & -0.4973 & -17.62 & 2.438 & 5.586 & 211.3 & 0.0004708 & 0.05865 & -17.9 & 0.2405 & -0.3444 & -0.1911 & 1.316\cdot 10^{-11} & -1.202\cdot 10^{-12} & -3.561\cdot 10^{-13} & -8.767\cdot 10^{-14} \\
1.313 & 0.5034 & 17.65 & 2.387 & -5.596 & -211.7 & 0.0004716 & 0.05876 & 0.2307 & -15.05 & -0.1828 & -0.3931 & -6.521\cdot 10^{-12} & 1.678\cdot 10^{-11} & 3.119\cdot 10^{-13} & 6.859\cdot 10^{-14} \\
1.314 & 0.4986 & -17.69 & -2.44 & -5.607 & 212.1 & -0.0004725 & 0.05888 & -0.3454 & -0.193 & -17.86 & 0.2397 & 7.872\cdot 10^{-14} & -7.621\cdot 10^{-14} & -1.253\cdot 10^{-13} & 7.315\cdot 10^{-14} \\
1.312 & -0.5035 & 17.65 & -2.392 & 5.595 & -211.7 & -0.0004715 & 0.05876 & -0.1822 & -0.3943 & 0.23 & -15.27 & -2.032\cdot 10^{-12} & 3.13\cdot 10^{-12} & -1.163\cdot 10^{-13} & 1.463\cdot 10^{-15} \\
\hline
0.164 & -0.06216 & -2.203 & 0.3047 & 0.6983 & 26.42 & 5.885e-05 & 0.007331 & 4.808 & 0.03006 & -0.04305 & -0.02389 & 0 & 0 & 0 & 0 \\
0.1642 & 0.06293 & 2.207 & 0.2983 & -0.6995 & -26.46 & 5.895e-05 & 0.007345 & 0.02883 & 4.764 & -0.02285 & -0.04913 & 0 & 0 & 0 & 0 \\
0.1643 & 0.06233 & -2.211 & -0.305 & -0.7009 & 26.52 & -5.907e-05 & 0.00736 & -0.04318 & -0.02412 & 4.81 & 0.02997 & 0 & 0 & 0 & 0 \\
0.164 & -0.06294 & 2.206 & -0.299 & 0.6994 & -26.46 & -5.894e-05 & 0.007345 & -0.02278 & -0.04929 & 0.02875 & 4.771 & 0 & 0 & 0 & 0
\end{array}\right] \tag{D.15}
$$