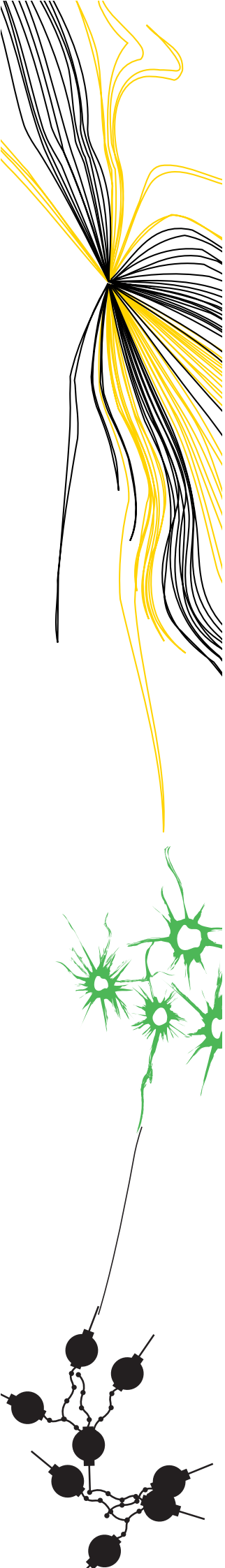


RAM ROBOTICS AND MECHATRONICS

DATA-DRIVEN MODELING AND CONTROL OF MULTI-ROTOR AERIAL VEHICLES IN CONFINED ENVIRONMENTS

R. (Rohit) Aggarwal

MSC ASSIGNMENT

Committee:
dr. ir. R.G.K.M. Aarts
A. Alharbat
prof. dr. ir. A. Franchi
dr. ir. A.Y. Mersha, Saxion

November, 2023

055RaM2023
Robotics and Mechatronics
EEMCS
University of Twente
P.O. Box 217
7500 AE Enschede
The Netherlands

UNIVERSITY OF TWENTE. | TECHMED CENTRE UNIVERSITY OF TWENTE. | DIGITAL SOCIETY INSTITUTE

Abstract

Multi-aerial rotor vehicles, such as quadcopters and hexacopters, are increasingly finding application in confined environments, ranging from indoor surveillance and inspection to warehouse automation. Achieving precise and agile control in these constrained spaces presents unique challenges, that includes navigating limited space, coping with intricate environmental dynamics, the aerodynamic disturbances of the confined spaces and real-time decision-making constraints. Data-driven modeling incorporated Nonlinear Model Predictive Control (NMPC) are proposed as suitable solutions for addressing these challenges, with a focus on learning for improved adaptability and the capacity to model and adapt to disturbances in confined space environments.

This thesis explores if the integration of data-driven modeling and model-based controllers can improve the performance of Multi Rotor Aerial Vehicles(MRAV) in confined environments. The potential combination of data-driven modeling and NMPC control offers several advantages in the context of confined environment applications. The major advantage is that the data-driven model provides a more accurate representation of the disturbances, allowing for improved tracking and robustness in the face of external disturbances which is crucial when navigating around obstacles. Firstly, the dynamic effects of flying an MRAV in a confined space is analysed based on measurements collected from physical experiments. Then, an NMPC controller that incorporates a Gaussian Process model is explored, proposed, implemented, analysed, and validated with real-time simulations. The proposed controller is compared with a nominal NMPC and NMPC controller that incorporates a state of the art Disturbance Observer. The results shows that the proposed controller outperformes both the nominal NMPC controller and the NMPC controller with the Disturbance Observer.

Acknowledgement

I'd like to begin by expressing my sincere appreciation to my daily supervisor, Ayham Alharbat, for the unwavering support he offered throughout the duration of the assignment. Ayham consistently delved into the intricacies, engaged in discussions, and ensured that I could always maintain my focus on the assignment. Furthermore, I extend my heartfelt thanks to Davide Bicego for our progress meetings, productive discussions, the valuable insights you provided, and your meticulous feedback that always enhanced my learnings.
I would also like to express my gratitude to Antonio Franchi for providing me with the invaluable opportunity to work under his guidance. Antonio's remarkable work and contributions have been a significant source of inspiration for my involvement in this demanding field. His insights and valuable input have been instrumental in shaping the outcome of my work, and I have gained extensive knowledge from our interactions. Additionally, I extend my appreciation to Abeje Mersha for the support and valuable input that have made a substantial contribution to my research. I am deeply grateful to my parents for their unwavering support, boundless love, and unshakable trust in me throughout my journey. Their constant encouragement and the freedom they granted me to think and choose have been instrumental in shaping the person I've become. I feel blessed to have parents who have always stood by my side, empowering me to explore my path and make my own decisions. Their love and guidance have been my greatest blessings. Thank you for everything.
I'd like to extend my gratitude to both my friends Samir and Remco for their unwavering support and the valuable insights they provided on various matters. Their presence have been a source of strength and wisdom throughout my journey. I would also like to express my heartfelt thanks to everyone who was there to support me. Your collective encouragement and presence have made a significant difference in my journey, helped me overcome obstacles, and I'm truly grateful for it.

With regards
Rohit

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter aims to give an introduction to the assignment. The chapter is divided in four sections namely: motivation, goals of the assignment, experimental validation, and the thesis outline.

1.1 Motivation

Advances in the field of multi-Rotor Aerial Vehicles (MRAVs) have made it possible for vehicles to fly autonomously in a 3D indoor environment. The wide range of applications of such vehicles include inspection and maintenance, mapping, monitoring, surveying, security, physical interaction with environment and others. In case of an earthquake structures such as a small bridge under railways, buildings, sewer tunnels, and mines can be difficult for adults to inspect. Also, it is logistically time-consuming depending on the size of the structure. Furthermore, there can be potential hazards such as harmful gases and chemicals, venomous animals, or insects in addition to the unstable structures [3].
Based on these potential risks, Defense Advanced Research Projects Agency (DARPA) has introduced a sub-terrain challenge [4]. The challenge is to come up with innovative solutions for aid disaster response and combat operation in underground tunnels and natural caves environment. This enhanced interest and innovation in indoor aerial vehicles and also highlighted the potential risks in these environment.
The motivation for beginning this work is taken from the high demand of a precise hovering and maneuvering operations in close quarters for aerial vehicles performing inspection and maintenance tasks. These tasks generally requires the MRAVs to fly in close proximity to the boundaries such as the walls, ceiling, ground, or pillars with stability. However, when flying in close proximity to the boundaries, perturbations are induced on the vehicle because of the aerodynamic interaction between vehicle propellers and the boundary surfaces and as well as the lack of space to regulate the perturbed position and attitude of the vehicle by these disturbances. Hence, the proximity effects can have significant impact on the vehicle stability and affect its ability to perform tasks [3], [5]–[7].

1.2 Goals of the assignment

This section presents the questions that will be answered by this assignment. The assignment research question is to find if data driven model can improve the performance of model based controllers in confined environment?
This research question can further be divided into three sub research questions as:

1. How to model these erratic disturbances on MRAVs in confined environment?
2. How to extend the dynamic model of MRAVs to account for these disturbances induced by the air flows in confined environments?
3. How to extend the existing controllers to ensure accurate trajectory tracking in confined environments?

1.3 Thesis outline

This section describes the structure of the report.
Chapter 2 presents the accumulated research and results obtained by various groups and gives a comparison of the results while introducing the contribution of thesis.
Chapter 3 presents the experimental validation of the proximity effect and it's impact on the MRAV.
Chapter 4 presents the background used for the methodology in chapter5.
Chapter 5 presents the detailed methodology of the control scheme used to model and eliminate the disturbances.
Chapter 6 presents the results from the approach used in chapter 5.
Chapter 7 presents the final conclusion based on the results and future recommendation.

Chapter 2

Literature review

This chapter aims at accumulating and discussing the research and results obtained by the research group regarding the study of proximity effects on multirotor aerial vehicles(MRAVs) from a modeling and control point of view. The chapter is divided in three primary sections namely: proximity effects, control of MRAVs in confined spaces, and contribution.

2.1 Proximity effects

This section aims to discuss the research to understand the proximity effects and how these effects have been modeled analytically, and empirically as well as understand the impact on MRAVs. The presence of any boundary such as the ground, walls, ceiling or any other boundary that changes the flow into the rotors, can affect the performance of MRAVs. These proximity effects induce perturbations in the flight stability, when operating near these boundaries. The figure 2.1 depicts the flow of rotors in MRAVs under proximity effect.

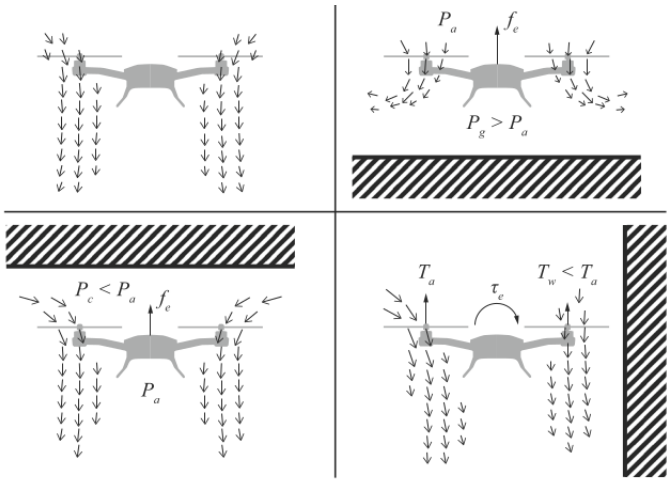


Figure 2.1: Flow vectors out of proximity effects: in ground effect, in ceiling effect, and in wall effect. Pressure is denoted as P, thrust as T, force as f, and torque as τ. Subscripts a, g, c, w, and e indicate away from surface, near ground, near ceiling, near wall, and external, respectively [1]

2.1.1 Ground effects

The study of proximity effects of the rotorcrafts can be traced back to the 1930s. The studies until the mid 1980s proposed few analytical approximations of functions based on thrust and power of one rotor disk also referred to as Helicopter In Ground effect(IGE). IN IGE one rotor disk is oriented parallel to ground up [1], [8]–[10]. [8] described an estimate of the power of a propeller, operating at constant thrust. By including image vortices below the ground plane, [9] described an analytical model to estimate torque of a rotor. Using the method of images, [10] described an analytical model to describe rotor thrust for a helicopter in ground effect. But it was only valid for $\frac{z}{R}$ >0.5, as the model did not take into consideration the effect of viscosity or boundary layer effects [1]. Also, some researchers carried out experimental studies for full scale helicopter for ground effects [11]. Numerous analytical models have been obtained and experimentally validated by various researches and can be seen in figure 2.2.

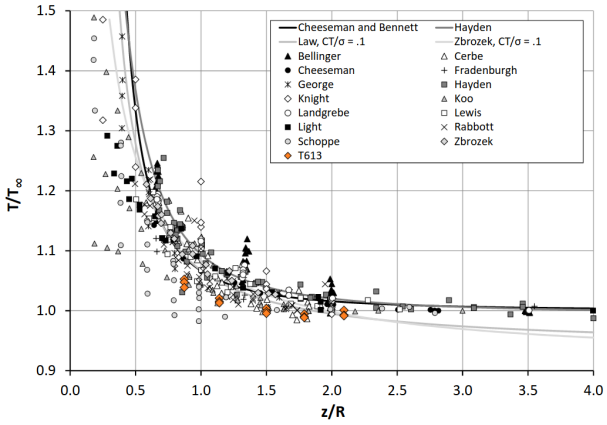


Figure 2.2: Ground effects experimentally validated from various research [2]

The y-label show the thrust ratio in hover and x-label show the normalized attitude. The ∞ show the baseline values. z and R denote the attitude over ground and radius of the rotor respectively. [12] studied the impact of ground on helicopter in hovering and forward flight scenarios by using the unsteady Reynold averaged Navier stokes method. The results showed unsteady flow as well as the flow which is recirculating the rotor region. It also showed that as the gap between the rotors and ground reduces, the lift is increased.

The study of multirotors experiencing the ground effect is inspired from the helicopter in the study of ground effect. The constant number 4 in mathematical model by [10] is experimentally replaced by [13] using identification with empirical data for the quadcopter under ground effect. [14] also followed the similar approach and the ground effect was seen up to $\frac{z}{R}$=5, that is more than what was predicted by model in [10].

[15] developed a method to predict power and thrust for small UAV experiencing ground effect numerically. The model combined blade element momentum theory with the model from [10]. The developed model effectively predicts the ground effect for the specific geometry of the propeller. The potential rotor-rotor interaction is not considered while developing

the model.
With the emergence of popularity of UAVs and advancements in computation technology, the interaction of the rotor in ground effect were studied using computational fluid dynamics(CFD), Particle Image Velocimetry(PIV) and advances were seen in control of rotorcrafts operating in region of the ground effects.
[7] has shown that the correlation between the mathematical model used by [10] and experimental data of the quadcopter is not good. A new model was proposed for the quadcopter in ground effect by using method of images and potential field. The model includes a single source situated at rotor geometric center and a term with coefficient. The model was empirically found that to incorporate recirculating flow and generated lift on the central body. The model was validated by physical experimentation and CFD. When all the rotors are not hovering close to the ground, asymmetrical thrust is caused between the rotors of quadcopter and the phenomenon is called the multirotor partial ground effect. This effect was recreated during flight with a PID controller. The observation showed that the effect obtained during flight and one obtained from the test bench are similar but during flight the effect is greater in magnitude.
The author also assessed the implementation of [10] for small rotor hovering over the ground, to find whether it was experiencing ground effect [16]. It shows that [10] is good approximation for single small rotors of size varying from 6 to 20 in. Following up, [17] developed a method to generate a map of the ground effect that denotes the change in thrust in environments with different obstacles. The approach works by testing the effects close to surfaces and then the obtained data is used to account for different obstacles. [18] has also studied the impact of the distance between rotors, rotational speed of the rotor and size of rotor on the quadcopter in ground effect. The thrust performance was evaluated by taking ratio of thrust generated by quadcopter out of ground effect to thrust in the ground effect. The results clearly showed that the influences of three parameters on the lift profile are different. It can be concluded that when compared with [7], the work by [18] underestimated the lift profile for the quadcopter IGE [3] but they agreed with [7] that [10] cannot accurately predict the ground effect. Following up on his work, the author presented a ground effect model based on empirical data [19]. The model is a rational polynomial for the torque as a function of standardized height, the radius of the propeller, and the propeller spacing. The coefficient of polynomials are obtained by data fitting using a cross-validation procedure to prevent selection biases and over fitting. The mean accuracy of model was within the range of $\pm 5\%$. The comparison of the models obtained using cross-validation and other methods is not discussed.
A modified model for the quadcopter experiencing ground effect is presented by [20],incorporating a ground surface quality coefficient. It was seen that for flat and hard surfaces, the ground effect is at its maximum compared to uneven surface such as grass. A backstepping control approach is used with the proposed model for reference trajectory tracking and validated in a simulated environment.
The empirical models are proposed by [21] for a quadcopter experiencing ground effect during hover and forward flight respectively. Each model further has two variants to account for

low to medium and high velocities. Although the results were obtained from Asctec Hummingbird quadcopter, the model was also validated for Bitcraze Crazyflie 2.0. The results indicated the model can be generalized for a quadcopter with different scale, control and trajectories.

2.1.2 Ceiling effects

The ceiling effect is another scenario that has been discussed by the researchers. The ceiling effect for a micro helicopter has been studied by [6] using CFD. The results showed that when the rotor ceiling gap is decreased, the local effective angle of attack along the blade span increases. The rotor efficiency is improved but when compared to rotor IGE, the efficiency is less.
Similar to the ground effect, [17] developed a method to generate a map of the ceiling effect that denotes the change in thrust in environment with different obstacles. The constant number 4 in mathematical model by [10] is replaced. The approach works by testing the effects close to surfaces and then the obtained data is used to account for different obstacles. The correction factor is included in mathematical model by [10] is experimentally replaced by [13] using identification with empirical data for the quadcopter experiencing ceiling effect. The research presented in [22] shows that the mathematical model based on [10] is accurate for single rotors In Ceiling Effect(ICE) and the model is experimentally validated. However for the multirotor experiencing the ground effect, the conclusion differs and the model in [10] is not accurate as suggested by [7]. [23] presented a solution to stay in the contact to inspect bridges and other infrastructure using the ceiling effect. The ceiling effect has been modelled for single rotor by changing 4 in the model by [10]. It is found that more thrust is produced within the multirotor but the shape is similar. The special fairing is used to isolate the aerodynamic effects for rotors of multirotor so as to replicate the performance as single rotor case. The distance was found to be 0.36R for maintaining the contact with the lower part of the bridge which was experimentally validated.
[24] also included a correct factor in the model by [10] by estimating the correction factor using CFD simulations.
[25] derived analytical models using the momentum theory and the blade element method for a spinning propeller ICE. The results were consistent with the performance but it does not give insight into the model's impact on MRAV and scalability.

2.1.3 Wall effects

Although number of studies have focused on the multirotor IGE or ICE, the scenario that is relatively new and has been less investigated despite the increase in research and use of UAV in indoor environments, is flying in proximity of walls or obstacles.
The interaction of the rotor has been investigated by [5] for a micro helicopter hovering near the wall. The results from the CFD simulations showed that due to hovering near the wall, an asymmetry is induced in the position and the circulation of the vortex wake. The

wake asymmetry, induces asymmetry in the lift forces that are acting on the rotor and the lift forces fluctuates as a function of the rotor azimuth angle. The fluctuation in lift forces induces disturbance moments that acts on the rotor blade. These disturbing moments are a function of rotor attitude and rotor-wall gap and have caused an adverse impact on the stabilization of the attitude of a micro helicopter.
[19] is one of the first study to investigate the near wall effects for the quadcopter. By early research it was suggested that the attractive force between the wall and quadcopter is induced by the pitching moment when the thrust vector changes. The results showed that attraction force between the quadcopter and the wall, and the pitching moment are not related and thus coupled. The study also proposed empirical models to model ground effect accurately based on numerical data.
A similar investigation for the wall effect is done by [3] using CFD and experimental methods for a quadcopter hovering in a tunnel of square cross-section. The study agreed that there is a disturbing moment acting on the quadcopter when near to the wall. The combined proximity effects are dominated by inviscid flow forces.

2.2 Control of MRAVs subjected to proximity effects

This section focus on the various control approaches used to model and eliminate the proximity effects for the MRAVs experiencing proximity effects.

2.2.1 Control

An adaptive altitude controller is proposed by [26] to avoid the ground effect for a quadcopter. The controller estimates the global thrust coefficient online, assuming that the thrust coefficient are same for all the rotors. The control law is:

$$u = mg + mk_1(k_1 + k_2)\delta_2 \tag{2.1}$$

where, δ_2, k_1 and k_2 are control parameters. k_1 and k_2 are fixed and δ_2 is a variable parameter that is changed when thrust estimation is changed. The ultrasonic sensor returned values with noise which is dealt by designing a position and velocity estimator. It was noted that perturbation resulting from the bad reading from the ultrasonic sensor can destabilize the system. The results showed that the controller works well but the results do not show controller performance below the altitude of 50cm as the ground effects might not be observed significantly above 50cm based on the drone size.
[27] presented a comparison in simulation between a feedback linearization(FL) controller and an adaptive sliding mode(ASD) controller to estimate the ground effects. The ground effect have been modelled as an additional acceleration term is:

$$g_r(z) = \left\{ \begin{array}{ll} \frac{A}{(z+z_{cg})^2} - \frac{A}{(z_o+z_{cg})^2}, & 0 < z \leq z_o \\ 0, & \text{else} \end{array} \right\} \tag{2.2}$$

where, A is ground effect constant, z_{cg} is the z component of the center of gravity, z_o is altitude below which ground effect affects the MRAV. The FL controller is designed as a position control of UAV and the ASD controller is designed as reference trajectory tracking controller. It was seen from the results that the FL controller against perturbation and sensor noise. For ASD controller, the input matrix was of size 6×4, hence was not invertible. An augmented input matrix is designed by introducing slack variables into the input matrix. The augmented control input is:

$$U = G^{-1}(x)[F(x, x_d) + \nu - f_r(x)] \tag{2.3}$$

where, x_d is the desired state, x is the state, G(x) is the augmented matrix, ν is the slack variable estimates, and f_r(x) is the estimate of ground effects. The results showed that the proposed control scheme works well with presence of sensor noise and also is able to compensate the ground effects.

The control scheme is proposed by [28] to land a quadcopter on a vertical oscillating platform. The control structure is divided into three modules: the motion estimation module, the trajectory generation module and an adaptive robust controller(ARC) as a tracking control module. The ground effects are adapted robustly using ARC. ARC estimated an equivalent inertia(m/k_G) online using measurements from MoCap system and inertial measurement unit(IMU). The results shows that using the tracking control module reduces the tracking error when the quadcopter approaches the platform.

[29] presented a model reference adaptive controller(MRAC) scheme for compensating ground effect on mini quadcopters. A mathematical model was experimentally formulated and validated. It was found that when the rotor to rotor distance is small, the total thrust of the quadcopter increases linearly as the quadcopter approaches the ground. When the distance is found to be large, the total thrust is found to be quadratic as proven in [10]. The control scheme used a linear quadratic regulator(LQR) control and a feedforward controller in inner loop for stabilizing quadcopter and altitude tracking. The PID controller and MRAC is used in outer loop for controlling position and countering the ground effects. The MRAC is developed by considering an unknown matched uncertainty to the original linearized system. The unknown uncertainty F(X) is written as a linear combination of basis functions $\phi(x)$ with unknown coefficient θ and is shown in equation 2.4.

$$\begin{aligned} \dot{x} &= Ax + B(u + F(x)) \\ F(x) &= \theta^T \phi(x) \end{aligned} \tag{2.4}$$

To cancel this uncertainty the adaptive control input is given by equation 2.5.

$$U_{adp} = -\hat{\theta^T}\phi(x) \tag{2.5}$$

The study considered two basis functions, namely, linear function and radial basic functions(RBFs). The rate of change of $\hat{\theta}$ is given by the difference between measured state and reference. The results showed that including MRAC into the control scheme outperforms the multirotor using only PID controller. Also, The MRAC using RBFs performs better as

compared to the linear model although computation is increased for RBFs.
The robust controller is applied by [30], by performing numerical simulation on the quadcopter. A position controller is developed using the saturated robust integral of sign of error(RISE). The ground effects are obtained from equation 2.2. The control law is:

$$u = G^{-1}(x)[\gamma_1 tanh(\nu)] \tag{2.6}$$

Here, γ_1 is a control gain>0, ν is Filippov solution obtained from the differential equation containing the tracking errors. The result showed that the proposed controller is able to compensate for the ground effect and added noise while guaranteeing that the saturation limit is not transcended in addition to using advantages of high gain control.
[31] presented a cascaded control approach to compensate for the ceiling effect. The cascaded control consists of an attitude controller that incorporate a PID controller and a position controller consisting of Nonlinear model predictive controller(NMPC). The system identification is performed using UKF and constraints for the parameters are estimated using a sigmoid based function. The results shows that the position controller is effective.
Following up to [31], [32] presented a cascaded control scheme with a nonlinear force estimator for close proximity to ceiling. The force estimator identifies forces online based on the pose measurement. The results show that the quadcopter can handle the ceiling effect upto 1 cm below ceiling.
A semi autonomous control scheme is proposed by [3], [33] for a quadcopter flying in a tunnel like environment. The system used a combination of a Kalman filter for translational position and velocity and a Hough scan matching(HSM) for computation of position information with fusing acceleration data. The Kalman filter is used because the update rate for position is slow from HSM. The estimated states are used to track the error and an integral controller is used for self stabilization. The control scheme is only valid in y-z frame. The control law is:

$$\begin{aligned} u_1 &= \frac{1}{\cos\phi\cos\theta}(g + \ddot{z}_d + a_1 e_{p_z} + a_2 \dot{e_{p_z}} + a_3 \beta_z) \\ \sin\phi &= \frac{-1}{u_1}(\ddot{y}_d + c_1 e_{p_y} + c_2 \dot{e_{p_y}} + c_3 \beta_y) \end{aligned} \tag{2.7}$$

where, a_1, a_2, a_3, c_1, c_2 and c_3 are positive control gains, $\ddot{z}$ and $\ddot{y}$ are the desired accelerations, and e_p and $\dot{e}_p$ are tracking and velocity errors. ϕ is the target input roll angle. The result shows that the position estimation and velocity estimation from Kalman filter are smooth. The two types of experiments, step input tracking and trajectory tracking are performed to validate the control scheme. For the step input tracking, the quadcopter is able to follow the z position with negligible errors, however, the system has lateral oscillations of $\pm$50mm when trying to reach y position, experiencing significant errors. For the trajectory tracking, the control scheme is able to follow the trajectory in a stable way, however, the performance deteriorates over time. The errors are present in both positions, 50 mm in z-axis and 80mm in y-axis. The performance is compared with PID controller and it shows to outperform the PID controller.

2.2.2 Observers

This section focuses on active rejection control i.e. when observers are used to estimate and compensate for the disturbances(internal or external) online [1].
The wall effects have been estimated and compensated by [34], using a Luenberger observer. The state and disturbance estimator gain is estimated using h_∞ synthesis using onboard IMU sensor. The results showed that the proposed control scheme can passively estimate the wall effect, however, the estimation has errors and the estimate encounters numerical instability as the quadcopter goes further away from the surface.
The control system of the quadcopter is decoupled by [35] into three loops that are inner, angle and position loops. The dynamic inversion is implemented to design a control law for position and angle loops. The incremental nonlinear dynamic inversion(INDI) is implemented to design a control law for the inner loop. The integral extended state observer(IESO) is incorporated with the inner loop to achieve ground effect rejection. The arranged transient process(ATP) is used to smooth out and smooth transition in the desired reference. The control law for the inner loop is given as in 2.8 [1] which is similar to 2.3.

$$U = w_o + G^{-1}(x)[F(x, x_d) - \Delta_{dis}] \tag{2.8}$$

Where, w_o are the velocities at last sample time, and Δ_{dis} are the disturbances that account for the ground effects. The results showed that IESO can observe and mitigate the ground effect in simulations. Additionally, the control scheme is shown to be effective for trajectory tracking in simulation.
Similar to the IESO, some non-linear disturbance observer(NDO) based control scheme is also used. [36] proposed a control scheme to predict and compensate the ground effects by incorporating an adaptive non-linear disturbance observer(ANDO) in the inner loop for attitude control. The inner loop consists of ANDO and a propotional-integral-velocity(PIV) controller while the outer loop for position is controlled by a PID controller. The ANDO is used to estimate the disturbance in torque and forces due to the ground effect. The ground effects are compensated by converting the disturbance estimates to pulse width modulation(PWM) and adding the obtained PWM values to the motor mixer. An adaptive law is designed for the motor parameters to estimate the global thrust coefficient and for simplification of the modeling process. The adaptive law is:

$$\begin{aligned} T_{i\infty} &= k_m W_i^2 \\ \dot{\hat{k}}_m &= c_m c_t e_z \end{aligned} \tag{2.9}$$

where, $T_{i\infty}$ is generalized torque, k_m is the motor parameters, c_m is adaptive gain, c_t is the control output, e_z is the height tracking error and W_i is the PWM command of the i^{th} motor. The results showed that the proposed ANDO effectively compensate the disturbances by ground effect and settling time is reduced by 60$\%$ in a simulation environment.
[37] focused on designing and validating the NDO from [36] without adaptive law in combination of the model based feed forward controller to compensate for the ground effect. The linear velocity and angular velocity along height are estimated using a barometer and an

IMU respectively. The compensator constraints the thrust to set limits. The tracking performance is improved and error is reduced by 23$\%$ when compared to without NDO. [7] also implements NDO to estimate the force and torque for the quadcopter experiencing ground effect, compared the performance with the conventional PID and feed forward scheme. It is shown from the obtained result when a quadcopter is in the test bench that it outperforms the conventional PID controller, however the feed forward scheme provides the best response.

2.2.3 Estimator

This section focuses on active rejection control i.e. when estimators are used to estimate and compensate for the disturbances(internal or external) online [1].
[38] used a continuous ant colony filter to estimate and compensate the ground effects. The filter used the dynamic model of the quadcopter. The ground effect is modeled using a coefficient provided by Hayden [39]. The equivalent radius is used instead of the radius of single rotor to scale the ground effects for quadcopters. The inaccurate modeling of ground effect is dealt by using the filter to estimate and compensate the errors online. It was seen that the algorithm works in all flight conditions except for hovering because in hovering the parameters are not observable. This was dealt with by proposing a special landing procedure introducing sinusoidal motion in phase two instead on near hover condition. Simulations results shows that an accurate landing is performed.
[40] presented an UKF based approach to estimate the torque and force disturbance for the wall effect. It uses the Gaussian based regression model as UKF for predicting the forces and torques. The data is sent to a support vector machine(SVM) and a warning is triggered if proximity to a wall is detected. It has been shown that the proposed controller converges to true value for ground effect and stays robust in a simulation environment. The proposed scheme is also experimentally validated to estimate force with machine learning to respond to tasks such as wall detection, holding position relative to wind and avoiding downwash.

2.2.4 Neural networks

A deep leaning based nonlinear robust controller, termed as neural lander is presented by [41], to improve the performance of the MRAV during landing. The control approach uses a combination of a Deep Neural Network(DNN) and nominal dynamics models to learn high order interaction. The output of DNN is limited by the spectral normalization. The limits on the output are leveraged to design a nonlinear feedback linearization(FL) controller that uses the learned model and prove system stability in addition to disturbance rejection. The learned model is trained offline. The control law is:

$$u = B_o^{-1}(u_1 - mg - \hat{f}_a) \tag{2.10}$$

where B_o is the input matrix, u_1 is the PD controller output, gravity vector is given by g and $\hat{f}_a$ is defined as DNN prediction to the disturbances induced by the ground effect. It is experimentally validated that the proposed neural lander outperforms the baseline controller in all

three axes. Additionally, it successfully captures the ground effect with other aerodynamics effects while assuring the stability of the controller.
As an extension to [41], [42] implemented DNN to include online learning with the goal of adapting to the new unknown situations. The model was tested on the quadcopter in the simulation environment with similar nonlinear FL controller as in [41]. The simulations showed positive and even better results than a model trained offline but the stablity is not assured for a closed loop scheme.
The Modular Deep Recurrent Neural Network(MODERNN) framework is used by [43] to model a multi-input-multi-output(MIMO) quadcopter model experiencing the ground effects. The ground effects have been roughly mapped in the force generated by motors. Although, ground effect is modeled, the influence of the ground effect on system has not been shown.
An adaptive controller and deep neural network based controller is presented by [44] to control the quadcopter experiencing ground effect during landing and levitating. The adaptive controller uses an additional force in z-direction as disturbance. The ground effect is estimated using an orthogonal trigonometric basis function. For the neural network based controller, data containing the relationship between the motion status and disturbance from ground effect is obtained offline by manually flying the drone up and down. The ground effect is then calculated from motion status and controller is used to compensate the ground effect. The control input is:

$$\begin{aligned} u &= u_n - u_d \\ u_n &= mg + m\dot{v}_d - Ke \end{aligned} \tag{2.11}$$

where, u_n and u_d are the nominal control input and compensation input respectively. Here, e is the state error, K is gain of the controller and $\dot{v}_d$ is desired acceleration. ReLu based learning is selected as an activation function. The measurement noise for the disturbance is reduced by using extended kalman filter.The experiments show that the neural network based controller has smooth landing and levitation as compared to the adaptive controller and they both outperform PID controller.
A novel learning based approach is proposed by [45] for modelling of multirotors experiencing near surface effects for arbitrary geometry of the platform. The approach uses three neural networks to predict the robot states and the control inputs. The approach starts by predicting the thrust for a single rotor and then corrects the predictions for the multi rotors. The model learns latent feature space of the landing platform for a robot centered occupancy grid using auto encoders. The proposed approach is validated using the single rotor and multirotor models, The robot centered occupancy grids by auto encoding are compared with the ground truth occupancy grids. The result showed that the approach using only rotor velocities and thrusts for a single rotor is not enough to predict the thrust correction for the multirotor. Also, height needs to be considered as input as well, however, it is not yet included in the model.
A self adapting sliding mode control(SMC) based on combination of sliding surface control based PID controller and a neuro-fuzzy controller is proposed by [46]. The output from a sliding surface based PID controller is used to update the parameter vector of neuro fuzzy

controller. The learning structure for the fuzzy controller is self evolving based on the balance between variance and bias. The MRAV is made to follow a circular and eight-shaped trajectory. It is shown that the proposed control scheme outperforms PID controller and NMPC controller in close proximity flight for tracking circular and eight shaped trajectories and holding the attitude. The stability analysis is provided to show that the error is bounded for the system having disturbances. The table 2.1 shows the overview of various control approaches discussed.

Here, the proximity disturbances are classified into ground effect, ceiling effect and wall effect and are denoted by G,C and W.

Table 2.1: Comparison of various control scheme for the multirotors. A: Proximity disturbances B: Data driven approach C: Model based control D: Controller validation through real experiment

References	A	B	C	D
Guenard et al. [26]	G	×	×	✓
D. Lee et al. [27]	G	×	×	✓
Botao Hu et al. [28]	G	×	✓	✓
Peng wei et al. [29]	G	✓	×	✓
Waheed et al. [30]	G	×	×	✓
Hei Vong et al. [3], [33]	G,W	×	✓	✓
Kocer et al. [31], [32], Robinson et al. [34]	C	×	✓	✓
Robinson et al. [34]	W	×	✓	✓
Han Du et al. [35], Xiang He et al. [36]	G	×	✓	×
Xiang He et al. [37]	G	×	✓	×
Sanchez-Cuevas et al. [7]	G	×	✓	✓
H. Nobahari and A. Sharifi [38]	G	×	✓	×
Mckinnon [40]	G,W'	×	×	✓
Guanya Shi et al. [41]	G	✓	×	✓
Guoxi Xu et al. [44]	W	✓	×	✓
Helen [42]	G	✓	×	✓
Edmonds et al. [45]	W	✓	×	×
Mohamad Abdul Hady et al. [46]	G,C	✓	×	✓
Learning Based MATMPC (Proposed solution)	✓	✓	✓	×

2.3 Contribution

It can be seen from the above that most of the literature is focused on the modeling and control of the MRAV experiencing ground effect and little literature is focused on the ceiling effect and even fewer on the wall effect. The models of ground effects based on an analytical model are not completely valid for the multirotor. The models obtained empirically by fitting

the data from experiments are able to explain the ground effect to an extent. However it does not comment on the scalability of the multirotors, lacks adaptability, and also may include lots of parameters that might be complex to work with. Section 2.2.1 gives an overview of various control schemes that are being used with the conventional controllers to model and compensate primarily ground effects. Based on the study so far, there is a lack of study to the best of my knowledge that comments on the proximity effects as a whole and also uses data data-driven approach with model-based control. Thus, this assignment intends to provide and if possible experimentally validate to model and compensate for the proximity effects during hovering maneuvers of the MRAVs using data data-driven approach. The method proposed by [47] is considered as the base motivation to model and compensate for the proximity effect as it uses a Gaussian process learning-based NMPC. Based on [37], a nonlinear disturbance observer is chosen as a state-of-the-art control scheme from the literature to compare our presented approach.

Chapter 3

Data Acquisition and Analysis

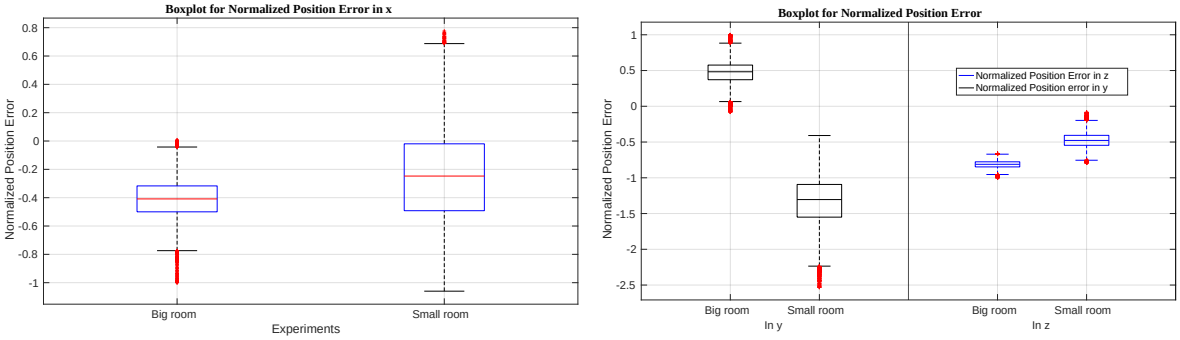
From the literature, it is seen that there exist proximity effects when MRAV moves in close proximity to the obstacles. This chapter intends to experimentally validate the proximity effect and use data analysis from the experiments to see the impact of these disturbances on the MRAV.
The first experiment was conducted at the Saxion University in a big room where it is assumed that there are no proximity effects on MRAV from the surrounding environment during the hovering phase. The size of the big room in Saxion was roughly $|4\text{m}\times4\text{m}\times6\text{m}|$. The other experiments were conducted at the University of Twente in a small room where the disturbances were present. The MRAV used for all three experiments is called Fiberthex, which is a six-tilted rotor platform. The same Nonlinear model predictive controller(NMPC) [48] with the same parameters was used to conduct the experiments. Figure 3.1 shows the test setups for the experiments. big room shows the data from hovering MRAV without experiencing any proximity effects i.e. showing the data from the experiment performed in the big room and small room has the datasets from hovering MRAV experiencing proximity effects i.e. showing the data from the experiment performed in small room.

Figure 3.1: Test Setup

3.1 Time domain

This section presents the effects on the MRAV experiencing proximity effects compared to when the effects are not present in time domain. The chapter A shows that there is a significant difference between datasets. The experiments are performed twice to check repeatability. Here, we chose to present the comparison of two datasets where each dataset represents the proximity effect present during a small room and the absence of proximity effect during a big room respectively.

Figure 3.2 shows the boxplot of data obtained for the error between positions and desired



(a) Boxplot for position error in x

(b) Boxplot for position error in y and z

Figure 3.2: Boxplot for Normalized Position Error

positions in x, y, z axes. The data has been normalized with respect to data from big room to clearly show the relative difference between the datasets due to the disturbances.

Table 3.1 shows the median, the mean, IQR(interquartile range) and the range of data for

Table 3.1: Data obtained from box plots

States and Errors	Axis	Median		Mean		Interquartile Range		Range	
		Big	Small	Big	Small	Big	Small	Big	Small
Position error	x	-0.0034	-0.0021	-0.0034	-0.0021	0.0015	0.0039	0.0061	0.0146
	y	0.0034	-0.0092	-0.0033	-0.0094	0.0014	0.0032	0.0058	0.0129
	z	-0.0452	-0.0266	-0.0454	-0.0266	0.0040	0.0078	0.0158	0.0310
Orientation error	x	0	-0.0052	0.0297	-0.0052	0.0225	0.0278	0.0674	0.1110
	y	-0.0105	0.0112	-0.0112	0.0108	0.0172	0.0234	0.0571	0.0930
	z	0.0391	0.0401	0.0393	0.0393	0.0074	0.0156	0.0297	0.0468
Linear velocity error	x	0.0020	0.0082	0.0019	0.0086	0.0132	0.0452	0.0525	0.1498
	y	-0.0033	0.0107	-0.0035	0.0113	0.0108	0.0272	0.0433	0.1086
	z	-0.0034	0.00084	-0.0033	0.00010	0.0281	0.0559	0.1121	0.2233
Angular velocity error	x	0.0015	-0.0024	0.0011	-0.000929	0.0615	0.1190	0.2460	0.4750
	y	-0.0014	0.00068	-0.0010	0.000955	0.0730	0.1355	0.2918	0.5406
	z	-0.0024	0.000022	-0.0020	0.000708	0.0378	0.0456	0.1509	0.1816
Thrust	Propeller 1	3.8864	3.9701	3.8590	4.0482	0.5720	0.9924	2.1691	3.6896
	Propeller 2	3.5114	3.9487	3.5349	3.9398	0.4728	1.0596	1.8036	3.8859
	Propeller 3	4.6647	4.3562	4.6823	4.3545	0.4792	0.9453	1.8188	3.4837
	Propeller 4	3.2402	3.5527	3.2216	3.6309	0.5496	1.0068	2.0806	3.7268
	Propeller 5	3.7728	4.5538	3.7679	4.5612	0.4638	1.0355	1.8459	3.5545
	Propeller 6	3.9813	3.7906	3.9944	3.8134	0.4618	0.9296	1.8311	3.4299

various error and states of the MRAV. IQR shows the spread of data in box or spread of 50 percentile of data. Range shows the spread of hundred percentile of data.

It is seen from figure 3.2 and table 3.1 that the medians of error in x and z axes are higher

for big room as compared to median of the error in small room, whereas median of error in y axis is higher for small room as compared to error in big room. The median is chosen over mean because median is more robust than mean.
Comparing the median and mean of errors in x, y and z axes, it is seen that in y-axis, the error is left skewed for big room and right skewed for small room. The right(positively) skewed distribution implies that the error density below median is high that drags the mean lower than the median and there are relatively fewer larger errors compared to the number of small errors instances that makes the distribution skewed whereas the left(negatively) skewed distribution implies that the error density above median is high that drags the mean higher than median and there are relatively fewer small errors compared to the number of large errors instances that makes the distribution skewed.
This suggests that the error in y-axis for big room has lower error at the center but there are few instances of large errors that make the distribution skewed whereas for small room have higher error at center but there are few instances of smaller error that makes the distribution skewed. For the error in z-axis, the error in big room has a right skewed distribution, whereas the error in small has a symmetric distribution. The same can be seen in figure 3.2. This suggests that the error in z axis for big room has a higher error at the center. However, there are a few instances of smaller errors that makes the distribution skewed. For the error in x-axis, the error is symmetrically distributed. Additionally, IQR and the range in table 3.1, shows that the variation of errors is significantly higher for small room as compared to big room.

Figure 3.3 shows the normalized orientation error in the x, y, and z axes and the errors

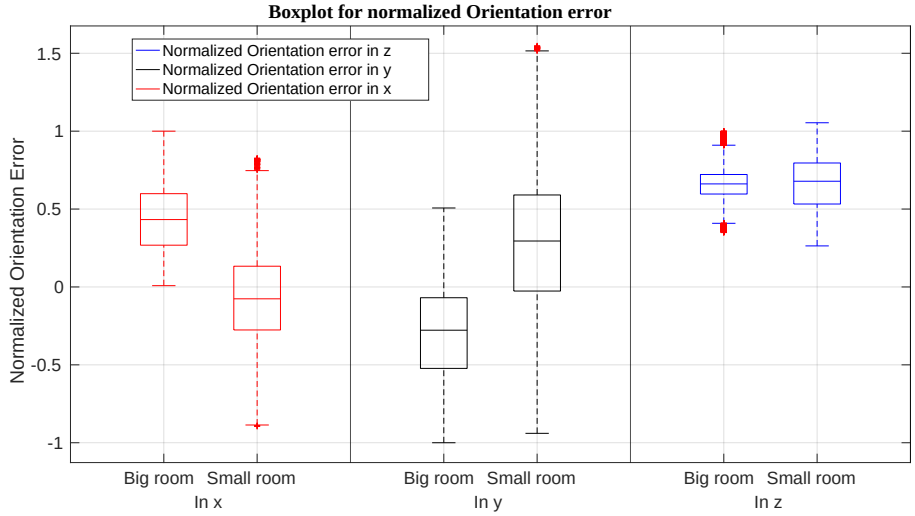


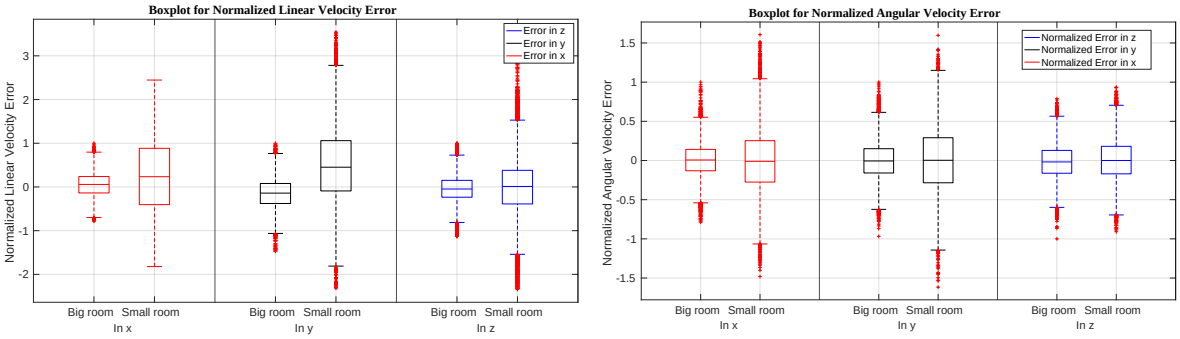
Figure 3.3: Boxplot for Normalized Orientation Error

have been normalized w.r.t. the error from big room. Figure 3.3 and table 3.1 shows that that median of the error in x axis is higher in magnitude for big room as compared to small room showing higher error for big room. However, for errors in y and z axes respectively, the median of errors is larger for small room as compared to errors from big room. Additionally, table 3.1 and figure 3.3 show that the variation of errors is significantly higher for small room

as compared to big room.

When comparing the median and mean of errors in x,y and z axes from table 3.1 and figure 3.3, it is seen that for the error in x axis, big room has right skewed distribution. However, small room has a symmetric distribution. The error in y axis for big room has a right skewed distribution and a left skewed distribution for small room. The error in z axis for big room has a right skewed distribution while the error for the small room has a left skewed distribution.

Figure 3.4a shows the normalized data obtained for the linear velocities in x, y and z axes



(a) Boxplot for Normalized Linear Velocity Error

(b) Boxplot for Normalized Angular Velocity Error

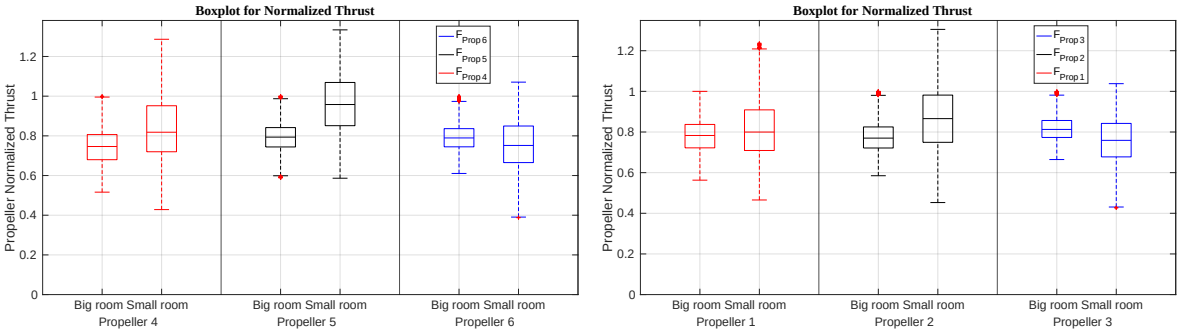
Figure 3.4: Boxplot for Normalized velocity Error

where the errors is normalized wrt errors from big room. Figure 3.4a and table 3.1 show that median of errors in x and y axes are higher in magnitude for the small room as compared to big room. The median of error in z axis is higher in magnitude for big room as compared to the small room. It is seen that from table 3.1, that the variation of errors is significantly higher for small room as compared to big room.

When comparing the median and mean of errors in x,y and z axes from table 3.1, it is seen that for the error in x axis, big room has a left skewed distribution and has a right skewed distribution for small room. The error in y axis for both room has right skewed distribution. The error in z for both the rooms have left skewed distribution.

Figure 3.4b shows the data obtained for the angular velocity errors in x, y, and z axes. It is seen from the figure 3.4b and table 3.1 that the median of the error in the x axis is higher in magnitude for small room as compared to big room. However, for the error in y and z axes, the median of error is higher in magnitude for small room as compared to the median of error for big room. It is seen that the variation of errors is significantly higher for small room as compared to big room. When comparing the median and mean of errors in x,y and z axes from table 3.1, it is seen that for the error in x axis for both the rooms has a left skewed distribution. The error in y axis for small room have right skewed distribution, whereas big room has a left skewed distribution. The error in z for big room has a left skewed distribution, whereas small room has a right skewed distribution.

Figure 3.5 shows the normalized thrusts data where thrusts are normalized w.r.t. thrusts from big room. It is seen from the figure 3.5a, figure 3.5b and table 3.1 that medians of thrusts from the propeller 1, propeller 2, propeller 4 and propeller 5 are higher in magnitude for small room as compared to big room.



(a) Boxplot for thrust for propellers

(b) Boxplot for thrust for propellers

Figure 3.5: Box plot for thrusts

The median of thrusts from propellers 3 and 6 are higher in magnitude for big room as compared to small room. It is seen that the variation of all six thrusts is significantly higher for small room as compared to big room.

When comparing the median and mean of errors of thrusts from table 3.1, it is seen that for thrusts in big room for propeller 1 and propeller 4 have a left skewed distribution and for small room have a right skewed distribution. The thrusts of propeller 2 and propeller 3 in big room have a right skewed distribution and a left skewed distribution for small room. The thrust of propeller 5 for big room has a left skewed distribution and for small room has a right skewed distribution. The thrust of propeller 6 for both room has a right skew distribution.

It is seen that except for the position in the x-axis, all other datasets have skewed distribution. During the repeatability of the experiments in the small room, it was seen that there is a bit of inconsistency in thrust from propeller 1 for experiments as compared to other thrusts. The possible reason for this is that one of the obstacles(wall in our case) was relatively closer to the propeller 1, so the effect is more dominant.

It is observed from IQR and range in table 3.1 and seen from all boxplots, that the spread or variation of all the states and errors are significantly higher for small room as compared to big room. This is expected because the disturbances have not been accounted for by the controller. When the controller reaches the desired position, there are unmodeled dynamics of disturbances that makes the controller deviate from the current position. When this happens, the controller again needs to reach the desired position.

3.2 Frequency analysis

This section presents the frequency analysis of data to show the impact on states and errors in the frequency domain due to presence of proximity effects. Table 3.2 shows the dominant frequencies and peak amplitude of the Fast Fourier Transform(FFT) of various states and errors. The fft plots are shown in appendix B. Figure 3.6 shows the logarithmic plots and the cumulative frequency distribution of the FFT of position error in x, y and z axes for experiments in the small room and the big room respectively. As seen from table 3.2, the position error has the same dominant frequency for both rooms suggesting that the periodic or os-

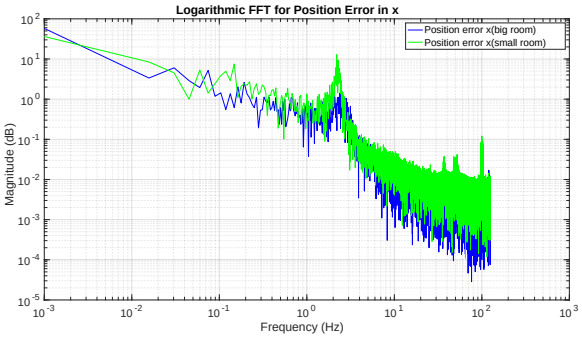
Table 3.2: Data from FFT plots

States and errors	Axis	Dominant frequency(Hz)		Peak Amplitude	
		A	B	A	B
Position error	x	0	0	57.8025	36.0901
	y	0	0	56.0059	159.7277
	z	0	0	771.0234	452.6878
Orientation error	x	0	0.0147	505.2573	109.6253
	y	0	0	191.1030	183.1622
	z	0	0	668.6526	668.5632
Linear velocity error	x	0	2.19105	31.7477	159.2447
	y	0	0	59.5080	192.7302
	z	0	2.36751	56.2915	270.5061
Angular velocity error	x	2.36751	2.19105	84.8752	185.2759
	y	0.441151	2.19105	83.5769	143.7759
	z	37.6448	37.1596	44.3157	81.3403
Thrust	Propeller 1	0	0	65610	68820
	Propeller 2	0	0	60100	66980
	Propeller 3	0	0	79600	74030
	Propeller 4	0	0	54770	61730
	Propeller 5	0	0	64060	77540
	Propeller 6	0	0	67910	64830

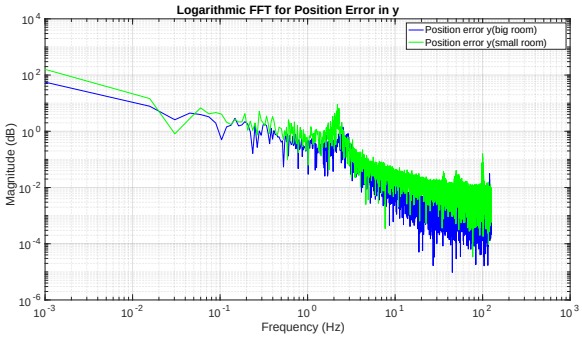
cillatory component is the same. Additionally, the peak amplitude of position errors in x and z axes at the dominant frequency has higher amplitude in big room as compared to small room suggesting that the errors are larger in big room as compared to small room at the dominant frequency. The amplitude of the position error in y-axis at the dominant frequency has a higher amplitude in small room as compared to big room. This suggests a higher error in small room as compared to big room at the dominant frequency.

From figure 3.6a, figure 3.6b and figure 3.6c, it is seen that as the frequency increases, the amplitude of the x,y, and z axes components for small room are higher but overall the amplitude keeps decreasing and not very significant when compared to amplitude around and at the dominant frequency. It is also interesting to note there are few exceptions that there exist few frequency components like between 5 to 7 Hz and around 240 to 245 Hz where the amplitude of fft of errors in big room is higher as compared to small room that might be highlighting the stochastic nature of noise that affects system dynamics at that particular frequencies. From figure 3.6d, it is seen that the proximity disturbances are more dominant in low frequency regions for errors in the x and y axes. The errors in the z-axis are similar in magnitude.

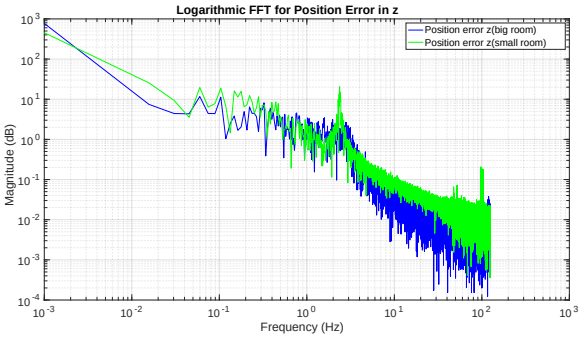
Figure 3.7 shows the logarithmic plots and cumulative frequency distribution of FFT of orientation error in x, y, and z axes for small and big rooms respectively. As seen from table 3.2 the orientation error has the same dominant frequency for both the rooms for y and z axes, suggesting that the periodic or oscillatory component is the same. However, for error in x axis, the dominant frequency is higher for the error in small room which is due to proximity effect affecting the system dynamics. Additionally, the peak amplitudes of orientation errors in all three axes at the dominant frequency have higher amplitudes in big room, as compared to small room. This suggests that the errors are larger in big room, as compared to small room at the dominant frequencies.



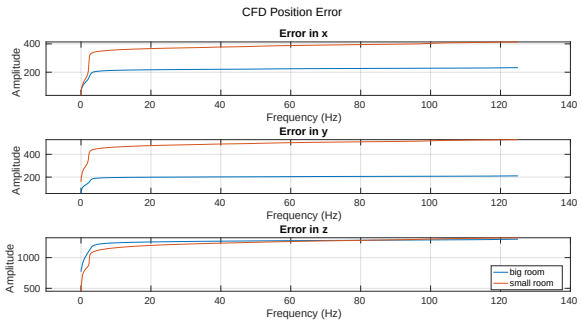
(a) Logarithmic plot of FFT of Position Error in x



(b) Logarithmic plot of FFT of Position Error in y



(c) Logarithmic plot of FFT of Position Error in z



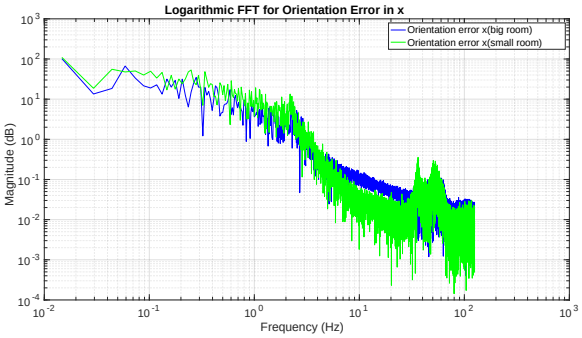
(d) CFD of Position Error

Figure 3.6: Frequency plot for position error

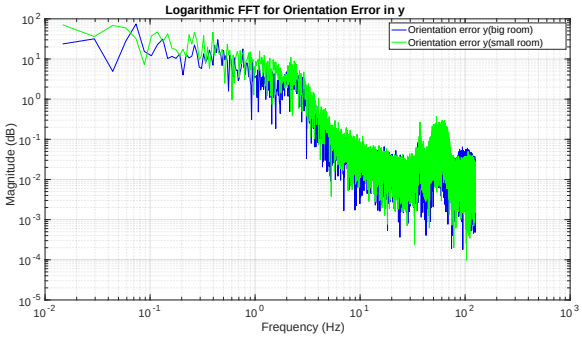
From figure 3.7a, figure 3.7b and figure 3.7c, it is seen that as the frequency increases, the amplitude of x, y and z components for small room are higher. This implies higher noise but, as a whole the amplitude keeps decreasing and is not very significant when compared to amplitude around and at the dominant frequency. It is also interesting to note that there exist a few frequency components for example between 10 to 60 Hz for error in x axis and around 240 to 245 Hz for all components where the amplitude of FFT of errors in big room is higher as compared to small room error in x. From figure 3.7d, it is seen that the proximity disturbances are more dominant in low frequency regions for errors in the x, y, and z axes as thats where the max errors are added.

Figure 3.8 shows the logarithmic plots and Cumulative Frequency Distribution (CFD) of the FFT of linear velocity error in the x, y, and z axes in the big and the small room. Table 3.2 shows that the linear velocity error has the same dominant frequency for both rooms for the y-axis suggesting that the periodic or oscillatory component is the same. However for the errors in x and z axes, the dominant frequency is higher for the error in small room which is again due to proximity effect affecting system dynamics. Additionally, the peak amplitude of linear velocity errors in all three axes at the dominant frequency have lower amplitudes in big room as compared to small room suggesting that the errors are less in big room as compared to small room at the dominant frequency.

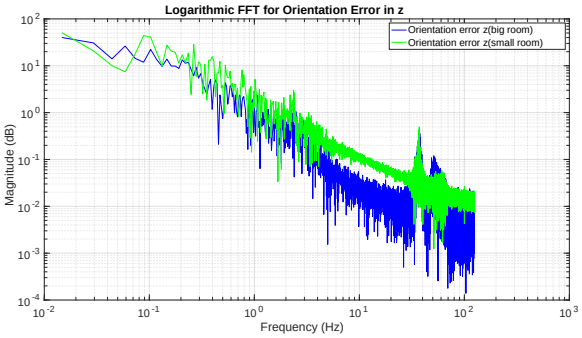
From figure 3.8a, figure 3.8b and figure 3.8c, it is seen that for the low frequencies, the amplitude of the x and y axes components for small room are higher, whereas for z-axis it is similar. This implies higher noise at low frequency for x and y axes, but as frequency increases the amplitude in all three axes for small room are lower than the amplitudes in



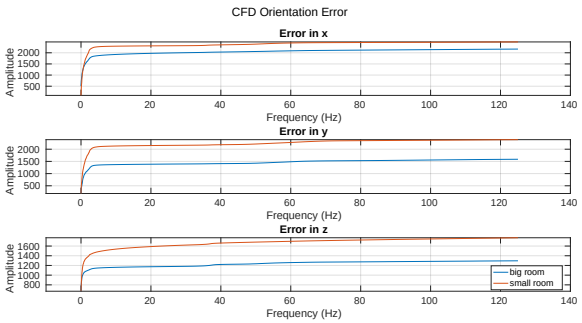
(a) Logarithmic plot of FFT of Orientation Error in x



(b) Logarithmic plot of FFT of Orientation Error in y



(c) Logarithmic plot of FFT of Orientation Error in z



(d) CFD of Orientation Error

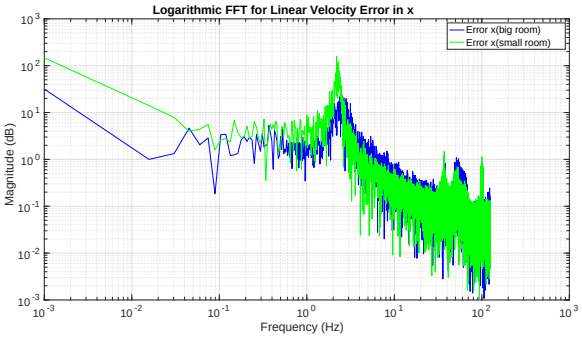
Figure 3.7: Frequency plot for Orientation error

big room. However, throughout bandwidth, the amplitude keeps decreasing and is not very significant compared to the amplitude around and at the dominant frequencies. From figure 3.8d, it is seen that the proximity disturbances are more dominant in low frequency regions for errors in the x and y axes. The errors in the z-axis are similar in magnitude.

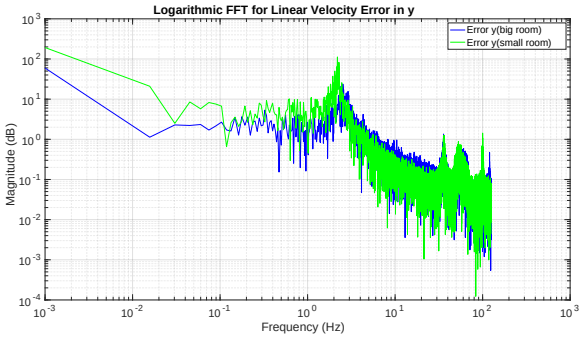
Figure 3.9 shows logarithmic plots and Cumulative Frequency Distribution (CFD) of the FFT of angular velocity error in x, y, and z axes in the big and the small room respectively. Table 3.2 shows that, the dominant frequency is higher for the error in small room compared to big room for the error in y-axis and dominant frequency being higher for big room in x-axis as compared to the small room whereas for error in z axis the dominant frequencies are close to each other. Additionally, the peak amplitude of errors in all three axes at the dominant frequency have lower amplitude in big room as compared to small room. This suggests that the errors are smaller in big room at the dominant frequency.

From figure 3.9a, figure 3.9b and figure 3.9c, it is seen that for the error in x and y axes the harmonics components for small room are higher than big room. For the error in z axis, the harmonic components for the small room are higher in the small room, upto 60 hz. However, as frequency increases, the magnitude of harmonic components in small room have similar amplitude compared to big room. The amplitude of high frequency harmonic components are significant compared to peak at the dominant frequency and contributes to the error. From figure 3.9d, it is seen that the proximity disturbances are more dominant in between 40 to 60 Hz for errors in the x, y, and z axes.

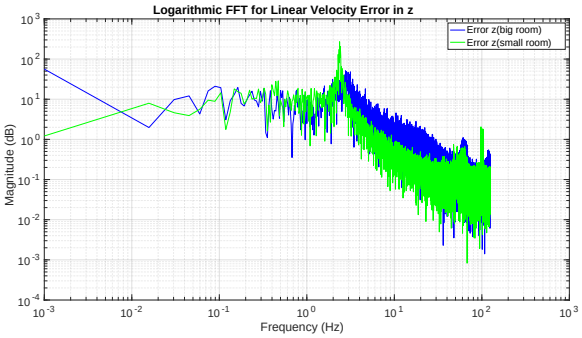
Figure 3.10 shows the logarithmic plots and Cumulative Frequency Distribution (CFD) of thrust for the propellers. It is seen from table 3.2 that the dominant frequency for all frequen-



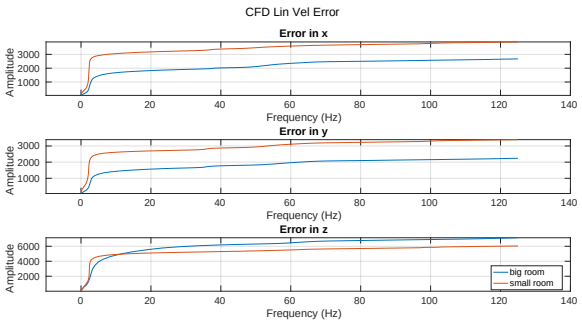
(a) Logarithmic plot of FFT of Linear velocity Error in x



(b) Logarithmic plot of FFT of Linear velocity Error in y



(c) Logarithmic plot of FFT of Linear velocity Error in z



(d) CFD of Linear Velocity Error

Figure 3.8: Frequency plot for Linear velocity error

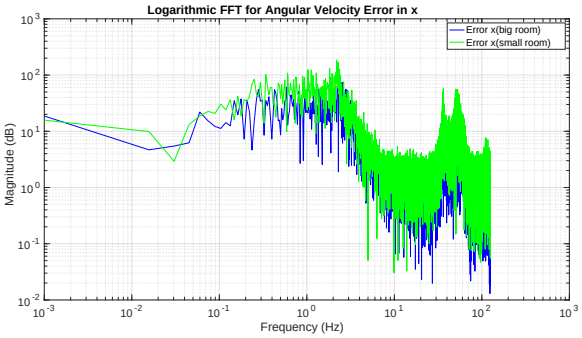
cies are same for all propellers. The peak amplitude in small room for thrusts for propeller 1, 2, 4 and 5 is higher and lower for thrusts for propeller 3 and 6 as compared to amplitude of big room. From figure 3.10a, 3.10b, 3.10c, 3.10d, 3.10e, 3.10f, it is seen that the amplitude of thrusts is higher for small room except around 10 to 25 Hz but as frequency increases the amplitude of frequency components keep decreasing and become relatively insignificant compared to peaks around the dominant frequencies. It is also seen from figure 3.10g that the proximity effects are more dominant in low frequencies as that's where the maximum increase in amplitude is observed.

3.3 Cross-spectral analysis

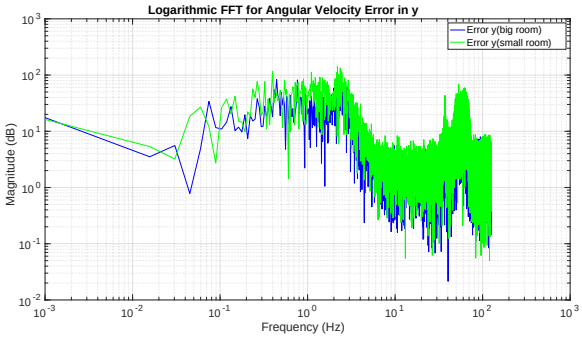
This section presents the cross-spectral analysis of the data. It was performed to measure the frequency domain relationship between two signals and how it correlates or co-varies with other signal at same frequencies. The cross spectral power density is obtained by taking Fourier transform of the cross correlation function.

Figure 3.11a shows the cross-spectral power density (CPSD) between different position errors in x, y and z for the big room and the small room. It is seen that the amplitude for small room are higher than amplitude for big room for the correlations between errors. CPSD suggest very weak or negligible relationship between aforementioned errors.

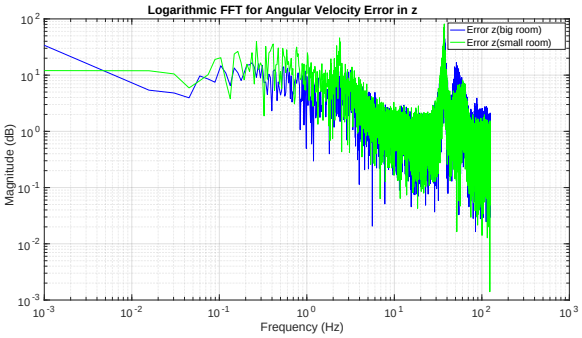
The figure 3.11b shows the cross-spectral power density (CPSD) between different orientation error in x, y and z in the big room and the small room. It is seen that the amplitude for



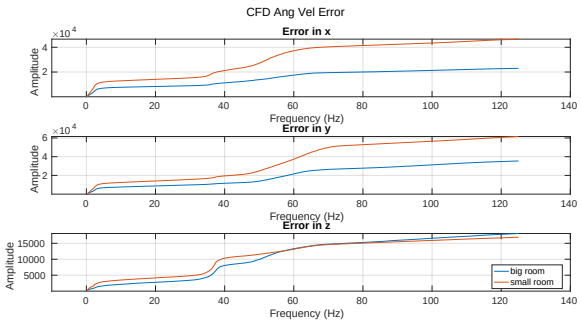
(a) Logarithmic plot of FFT of Angular velocity Error x



(b) Logarithmic plot of FFT of Angular velocity Error y



(c) Logarithmic plot of FFT of Angular velocity Error z



(d) CFD of Angular Velocity Error

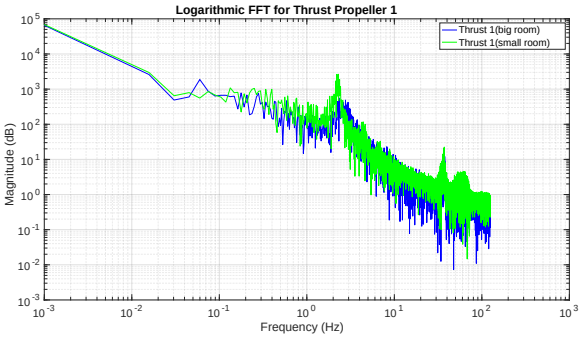
Figure 3.9: Frequency plot for Angular velocity error

big room is higher in amplitude for the small room for all the errors. CPSD suggest weak relationship for error between x and y for big room however the correlation is negligible for the small room. CPSD for orientation error between y and z suggests a moderate relationship for both the rooms. CPSD for error between x and z have a strong correlation for big room, a moderate correlation for the small room.

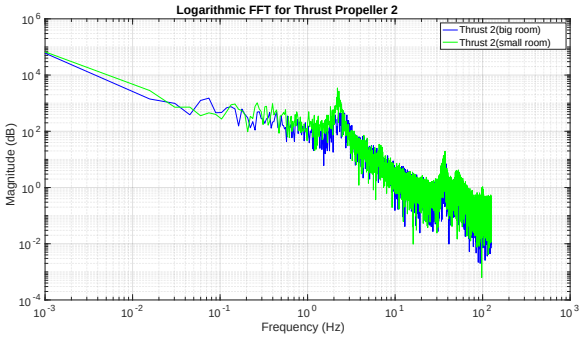
The figure 3.11c shows the cross-spectral power density (CPSD) between different linear velocity errors in x, y and z for the small room and the big room. From CPSD, it is suggested the correlation between all errors are negligible although it is interesting to note that the correlation for errors for small room.

The figure 3.11d shows the cross-spectral power density (CPSD) between different angular velocity error in x, y and z for the big room and the small room. From CPSD, it is suggested the correlation between all errors are negligible although it is interesting to note that the correlation for errors for small room.

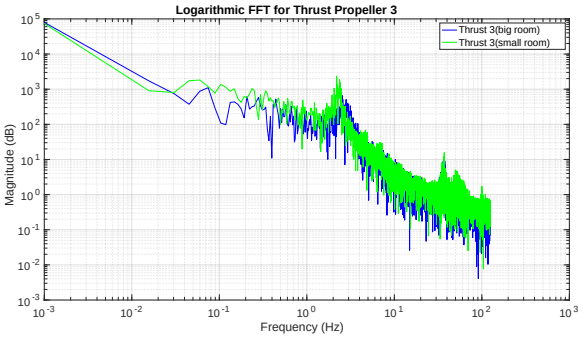
The figure 3.12 shows the CPSD between different thrusts for the big room and the small room. It is seen that all thrust have strong correlation with each other. There is more strong correlation for small room compared to big room for propellers 1 and 2, 1 and 4, 1 and 5, 1 and 6, 2 and 4, 2 and 5, 2 and 6, 3 and 5, 4 and 5, 5 and 6. There is similar correlation between propellers 1 and 3, 2 and 3, 3 and 4, 4 and 6 for both rooms.



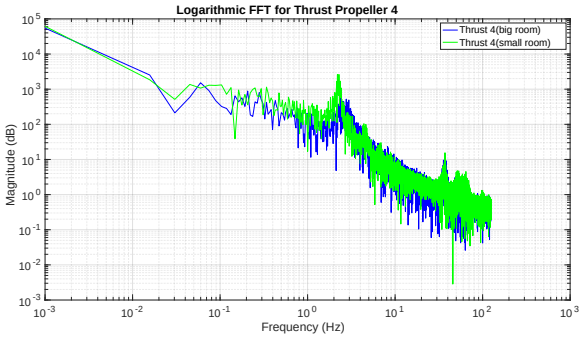
(a) Logarithmic plot of FFT of Thrust from propeller 1



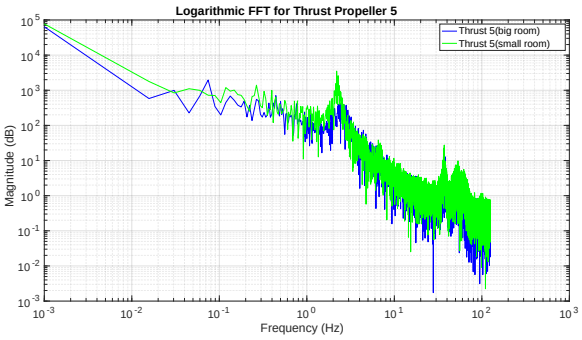
(b) Logarithmic plot of FFT of Thrust from propeller 2



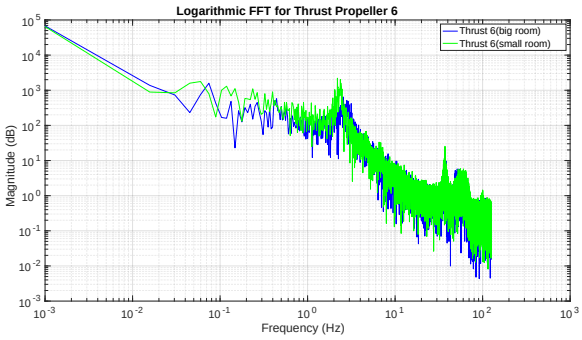
(c) Logarithmic plot of FFT of Thrust from propeller 3



(d) Logarithmic plot of FFT of Thrust from propeller 4

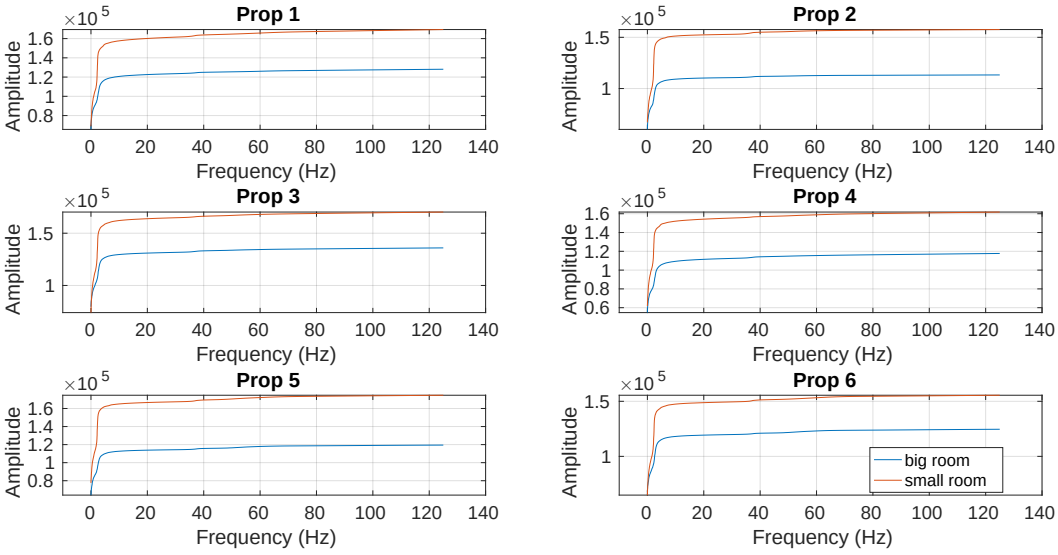


(e) Logarithmic plot of FFT of Thrust from propeller 5



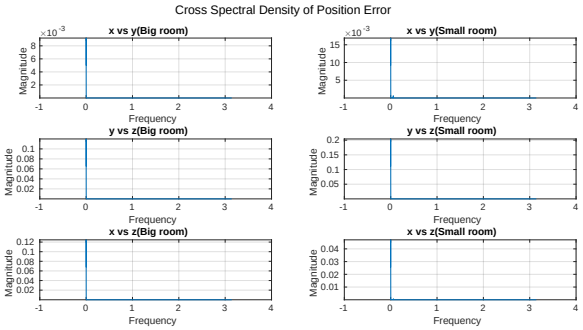
(f) Logarithmic plot of FFT of Thrust from propeller 6

CFD Propeller Thrust

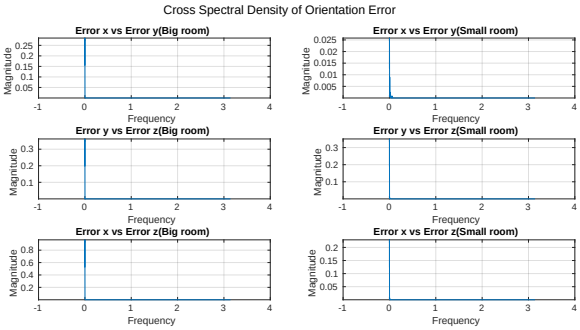


(g) CFD of Thrusts

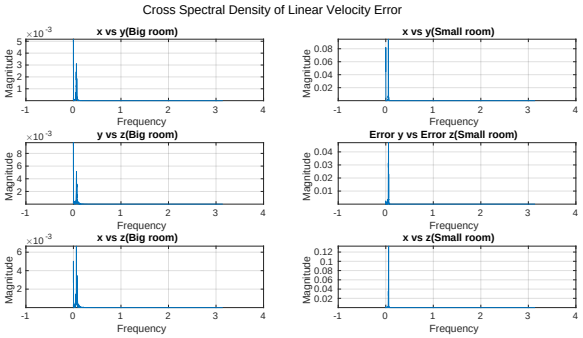
Figure 3.10: Frequency plot for Thrust



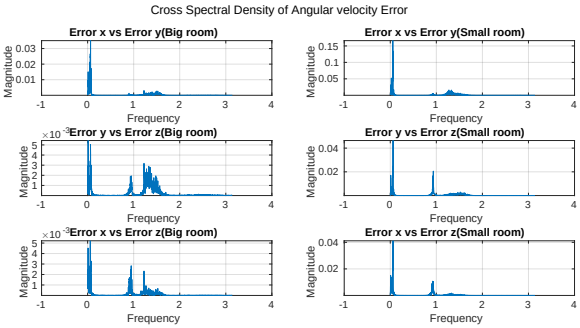
(a) Cross spectral density for position error



(b) Cross spectral density for orientation

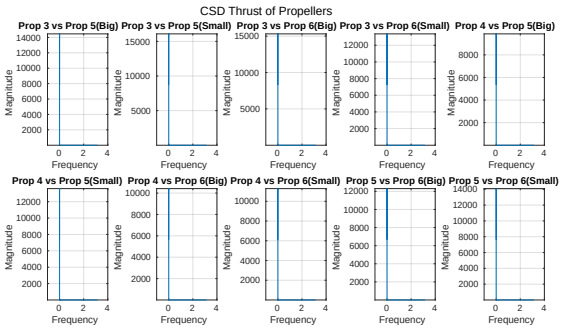


(c) Cross spectral density for linear velocity error

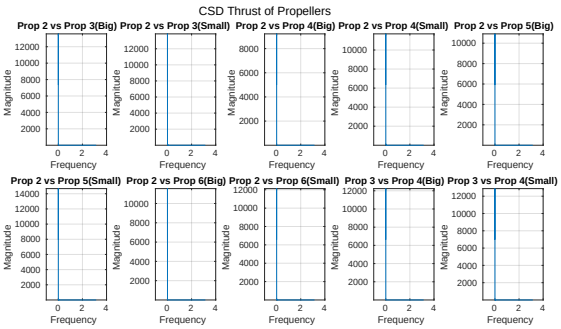


(d) Cross spectral density for angular velocity error

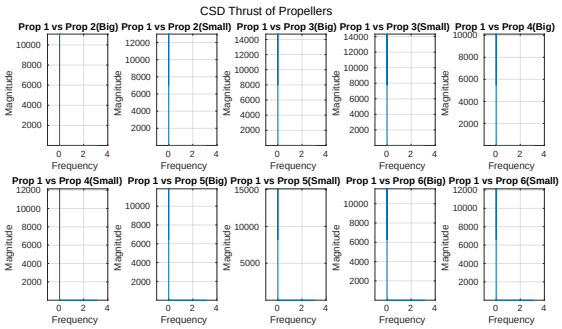
Figure 3.11: Cross spectral density



(a) CSD for propeller thrust



(b) CSD for propeller thrust



(c) CSD for propeller thrust

Figure 3.12: Cross spectral density for propeller thrust

Chapter 4

Theoretical Background

This chapter introduces the theoretical background used to derive the method to model and compensate for the proximity effect. The chapter first introduces the used controller namely, NMPC (Nonlinear Model predictive controller), then Gaussian process based modeling that will be used in the method and finally the state-of-the-art disturbance observer.

4.1 NMPC

The MPC(model predictive control) is based on the study of the LQR(Linear Quadratic Regulator) based optimization. For a continuous system, the model dynamics can be represented in the general form:

$$\dot{x}(t) = f(x(t), u(t), t, p) \tag{4.1}$$

where, x $\in \mathbb{R}^{n_x}$ is the state vector with n_x being number of states, u $\in \mathbb{R}^{n_u}$ is the input vector with n_u being number of inputs, p $\in \mathbb{R}^{n_p}$ are the parameters. An infinite dimension OCP (optimal control problem) defined for the continuous time of Bolza type objective function for the above equation is:

$$u^* = \underset{x,u}{\operatorname{argmin}} \;\; E(x(t_f)) + \int_{t_0}^{t_f} L(x(t), u(t))dt \tag{4.2}$$

$$\begin{aligned} r(x, u) &= 0, \\ s(x, u) &\leq 0 \end{aligned} \tag{4.3}$$

where L(x(t),u(t)) is stage cost and E(x(t_f)) is the terminal cost. The MPC computes the optimal solution i.e. the sequence u^* such as to minimize the cost or objective function along the future time window [t_0 t_f] called prediction horizon subjected to constraints, realized by functions r(x,u) and s(x,u) in equation 4.3. By standard, r includes the model dynamics constraints that evolve from equation 4.1 and real system measurements. The states and input constraints are included in s. The MPC solves the OCP at each sample time(Ts) iteratively. In other words, the principle of MPC can be summarized as:

- the measurement of the states are obtained $\hat{x}_k$ at time instant t_k where t_k is the current time

- solve the OCP for prediction horizon and get the first solution of u^* given by u_0
- obtained u_0 is applied to the real system for time Ts
- the process is repeated after each time Ts

The predictive horizon is shifted forward in the case of a receding horizon and shrank in the case of a diminishing horizon. It can be seen that the MPC has three primary blocks, namely the convex cost function reflecting the goals of the control tasks, the dynamics model of the physical system and the constraints imposed on the states, and the control input for the entirety of the control tasks.
The variant of MPC can be termed NMPC if it takes into account the nonlinear model dynamics of the plant and the constraints that can be either linear or nonlinear. The primary advantage of NMPC over reactive controllers where the control input at any time instant is computed with the state errors at the current time or past time, does not generate an intrinsic delay and guarantees that the physical constraints from the real system are satisfied by the input or the state.
The OCP is usually solved by three methods that can be categorized into dynamic programming, indirect methods and direct methods. The dynamic programming method uses Hamilton-Jacobi-Bellman(HJB) equations to compute a solution that gives the necessary optimal conditions for the OCP. The main drawback of this method is the high complexity and computation requirement for the moderate dimensions. The indirect method uses the minimum principles to get the necessary conditions for the OCP to define multipoint boundary value problem(MPBVP) i.e. differential algebraic equations solved usually by Newton optimization schemes iteratively. The main disadvantage of this method is deriving the DAEs can be intractable and the control elimination through algebraic methods might be impossible. The direct method converts the OCP into a nonlinear programming(NLP) problem with finite dimensions. The sufficient optimal conditions are obtained and solved by NLP solvers. The problem is first discretized and then solved for optimization and the direct method allows flexible implementation on the dynamic system and can handle the inequality constraints.
The direct method control can be further classified into two methods, direct shooting method and multiple shooting method. The direct shooting method for the forward simulation considers only the initial state. The method is not very useful for nonlinear system because of the high nonlinear propagation across time horizon during numerical integration. The solution to this problem is the direct multiple shooting method where the prediction horizon is split into a set of smaller intervals and each of these intervals is taken as an independent direct single shooting method while taking into account the constraints for continuity.

4.1.1 NLP formulation

The NLP is formulated after discretizing the OCP in equation 4.2 by applying the direct multiple shooting method over the prediction horizon with N shooting points and assuming system dynamics is locally Lipschitz i.e. smooth over the prediction horizon. The NLP with

constraints is:

$$\min_{x,u} \sum_{k=0}^{N-1} L(x_i, u_i, p_i) + E(x(t_f)) \tag{4.4}$$

$$\begin{aligned} \text{s.t.} \quad & x_0 = \bar{x}_0 \\ & x_{k+1} = \phi(x_k, u_k) \quad \text{k=0,1,....,N-1} \\ & x_k^l \leq x_k \leq x_k^u \quad \text{k=1,2,....,N-1} \\ & u_k^l \leq u_k \leq u_k^u \quad \text{k=0,1,....,N-1} \\ & g_k^l \leq G_k \leq x_k^u \quad \text{k=0,1,....,N-1} \\ & g_N^l \leq G_N \leq g_N^u \end{aligned} \tag{4.5}$$

where ϕ(x,u) is the discrete-time model dynamics, G_k and G_N represent the path constraints imposed on shooting points and x_0 is the initial solution. The discrete dynamics is used in constraints to include the continuity constraints during shooting intervals. The problem is converted to the form of quadratic programming(QP) by using Sequential Quadratic Programming(SQP) algorithm as discussed in the next section.

4.1.2 SQP

Assuming the initial solution (s_o,μ_o,λ_o), the NLP is computed using SQP by reformulating the NLP for a given iteration in the quadratic problem(QP) iteratively. The cost function and the constraints are substituted by its local quadratic approximation and the local affine approximation. The QP formed at the mth iteration is shown by 4.6.

$$\begin{aligned} \min_{\Delta \mathbf{s}} \quad & \sum_{k=0}^{N-1} (\frac{1}{2} \begin{bmatrix} \Delta x_k \\ \Delta u_k \end{bmatrix}^T H_k^m \begin{bmatrix} \Delta x_k \\ \Delta u_k \end{bmatrix} + \begin{bmatrix} g_{x_k}^m \\ g_{u_k}^m \end{bmatrix}^T \begin{bmatrix} \Delta x_k \\ \Delta u_k \end{bmatrix}) \\ & + \frac{1}{2} \Delta x_N^T H_N^m \Delta x_N + g_{x_N}^{m^T} \Delta x_N \\ s.t. \quad & \Delta x_0 = \hat{x}_0 - x_0, \\ & \Delta x_{k+1} = A_k^m \Delta x_k + B_k^m \Delta u_k + a_k^m, \; k = 0, \ldots, N-1 \\ & \underline{x}_k - x_k^m \leq \Delta x_k \leq \overline{x}_k - x_k^m, \; k = 1, \ldots, N \\ & \underline{u}_k - u_k^m \leq \Delta u_k \leq \overline{u}_k - u_k^m, \; k = 0, \ldots, N-1 \\ & \underline{c}_k^m \leq C_k^m \Delta x_k + D_k^m \Delta u_k \leq \overline{c}_k^m, \; k = 0, \ldots, N-1 \\ & \underline{c}_n - c_N^m \leq C_N^m \Delta x_N \leq \overline{c}_n - c_N^m, \end{aligned} \tag{4.6}$$

where, H_m and H_N are the Hessian matrices of the Lagrangian and are estimated by Gauss Newton(GN) approach or Generalized Gauss Newton(GGN) approach and the g_{x_k}, g_{u_k}, and g_{x_N} represent the first-order Lagrangian terms. The Gauss hessian Newton approach is used and hessian is reported to be always positive semi-definite.

$$\begin{aligned} \Delta \mathbf{x}^m &= \mathbf{x} - \mathbf{x}^m, \\ \Delta \mathbf{u}^m &= \mathbf{u} - \mathbf{u}^m \end{aligned} \tag{4.7}$$

where, x^m =[x_0^m,....,x_{N-1}^m,x_N^m] and u^m =[u_0^m,....,u_{N-2}^m,x_{N-1}^m]. The linear matrices are shown in 4.12. The ϕ represents the dynamics of the model and the r_k denotes f, the inequality constraint. The Explicit Runge-kutta 4(ERK4) method is used as an integrator to obtain the discrete dynamics. ERK4 is preferred for its convergence speed over Euler Method(EM).
For the NLP defined as:

$$\min_{s} \; h(s) \tag{4.8}$$

$$\begin{aligned} \text{s.t.} \quad & g(s) = 0 \\ & f(s) \leq 0 \end{aligned} \tag{4.9}$$

where h represents the objective function, s is the variable containing x and u, g represents the equality and f represents inequality constraints. As it is hard to find the global optimal solution for the optimal solution, we analyze the local optimal points similar to global solutions. These solutions are based on minimizing the Lagrangian function and the optimal solution needs to satisfy Karush-Kuhn-Tucker(KKT) conditions [49]. The Lagrangian function can be written as:

$$L(s, \lambda, \mu) = h(s) + \lambda^T g(s) + \mu^T f(s) \tag{4.10}$$

where λ and μ are the Lagrange multipliers. It is assumed that if s^* is a feasible solution, then there exists λ^* and μ^* such that,

$$\begin{aligned} & \Delta_s h(s^*) + \Delta_s g(s^*)^T \lambda^* + \Delta_s f(s^*)^T \mu^* = 0 \\ & g(s^*) = 0 \\ & f(s^*) \leq 0 \\ & \mu^* \geq 0 \\ & \mu_i f_i(s^*) = 0 \end{aligned} \tag{4.11}$$

where the first condition represents the stationary conditions, the second and third represent primal feasibility, the fourth represents dual feasibility and the last one represents complementary slackness. The primal solution s_* and the dual solutions λ_* and μ_* that satisfy the KKT conditions are also referred to as KKT points.

$$\begin{aligned} & A_k^m = \frac{\partial \phi_k}{\partial x_k}, \quad B_k^m = \frac{\partial \phi_k}{\partial u_k}, \quad a_k^m = \phi(x_k^m, u_k^m) - x_{k+1}^m, \\ & C_k^m = \frac{\partial r_k}{\partial x_k}, \quad D_k^m = \frac{\partial r_k}{\partial u_k}, \quad C_N^m = \frac{\partial r_N}{\partial x_N}, \\ & \overline{c}_k^m = \overline{r}_k - r_k(x_k^m, u_k^m), \quad \underline{c}_k^m = \underline{r}_k - r_k(x_k^m, u_k^m), \\ & \overline{c}_N^m = \overline{r}_N - r_N(x_N^m), \quad \underline{c}_N^m = \underline{r}_N - r_N(x_N^m) \end{aligned} \tag{4.12}$$

The primal and dual solutions provide the optimal values Δx and Δu for increments in x and u and new multipliers that are used to update the trajectory and Lagrange multipliers as:

$$\begin{aligned} x^{m+1} &= x^m + \alpha^m \Delta x^m \\ u^{m+1} &= u^m + \alpha^m \Delta u^m \\ \lambda^{m+1} &= \lambda^m + \alpha^m (\lambda^{m+1} - \lambda^m) \\ \mu^{m+1} &= \mu^m + \alpha^m (\mu^{m+1} - \mu^m) \end{aligned} \tag{4.13}$$

where α is called step-size and given by globalization strategies. The iteration continues until the KKT conditions are satisfied for the given value of accuracy. Once the value is obtained the first control input is applied to the system and moves the trajectory at the next sampling time instant. For real-time iteration α=1 is sufficient. The different global strategies like trust region or line search are more often required as SQP method or Interior Point(IP) methods don't guarantee that the solution will converge to a local minimum for the above problem. The globalization method used in this assignment is called line search.
The KKT conditions were discussed as it is the fundamental part of most of the solvers used to solve QP. The QP problem in equation 4.6 can be solved by available solvers. For this assignment, HPIPM [50] is considered and used.

4.1.3 MRAV model

The NMPC uses the detailed model of the physical system to produce an optimal input of the system. The model is taken from [48] for Fiberthex and is given as:

$$\begin{bmatrix} mI_3 & 0_3 \\ 0_3 & J \end{bmatrix} \begin{bmatrix} \ddot{p} \\ \dot{\omega_B^B} \end{bmatrix} = \begin{bmatrix} -\ddot{m}ge_3 \\ -\omega_B^B \times J\omega_B^B \end{bmatrix} + \begin{bmatrix} R & 0_3 \\ 0_3 & I_3 \end{bmatrix} G\gamma \tag{4.14}$$

where the symbols are shown in Table 4.1. The control input is denoted by u=$\dot{\gamma}$ and the constraints are defined as:

$$\begin{aligned} \gamma^l \leq \gamma \leq \gamma^u \\ \dot{\gamma}^l \leq \dot{\gamma} \leq \dot{\gamma}^u \end{aligned} \tag{4.15}$$

where, γ^l and γ^u denote the lower and upper bound on propeller forces, and $\dot{\gamma}^l$ and $\dot{\gamma}^u$ denote the lower and upper bound on the control inputs. γ and $\dot{\gamma}$ denotes the propeller forces and rate of change of propellers forces. The state vector x is defined as:

$$x_k = \Big[p, \eta, \dot{p}, \omega, \gamma\Big]^T \tag{4.16}$$

The mapping function to relate $\dot{x}$ and x is obtained from equation 4.14. The discrete time model for the control purpose can be discretized using various techniques such as fourth order Explicit Runge-kutta integrator as:

$$x_{k+1} = \phi(x_k, u_k) \tag{4.17}$$

Definition	Symbol
Body frame of MRAV with origin O_B and axes $\left(x_B, y_B, z_B\right)$	F_W
Inertial frame with origin O_w and axes $\left(x_W, y_w, z_W\right)$	F_B
Propeller frame with origin O_p and axes $\left(x_p, y_p, z_p\right)$	F_P
mass of MRAV	m
Inertia of the MRAV with respect to F_B and expressed in F_B	J
position of O_B at time t in F_w	p
velocity of O_B at time t in F_w	$\dot{p}$
Acceleration of O_B at time t in F_w	$\ddot{p}$
Rotation matrix expressing the orientation of F_B with respect to F_w	R^B
Gravitation acceleration constant	g
Angular velocity of F_B with respect to F_W and expressed in F_B	ω_B^B
Angular acceleration of F_B with respect to F_W and expressed in F_B	$\dot{\omega}_B^B$
Allocation matrix	G
Propeller forces in F_P	γ

Table 4.1: overview of the symbols used in this section

where, k=0,1...N-1. Since the NMPC is intended to be used for reference tracking, the reference signal y_r(t) is given as:

$$y_r = \left[p_r(t), \eta_r(t), \dot{p}_r(t), \ddot{p}_r(t), \omega_r(t), \dot{\omega}_r(t)\right]^T \tag{4.18}$$

and the output can be defined as:

$$y(t) = H(x(t), u(t)) = \left[p(t), \eta(t), \dot{p}(t), \ddot{p}(t), \omega(t), \dot{\omega}(t)\right]^T \tag{4.19}$$

The discretized versions of $y_r(t)$,and $y(t)$ are defined as $y_{r,k}$ and y_k respectively. The cost function in equation 4.20 is a function of the desired reference, output prediction, and control input.The NLP to be solved tries to minimize the cost function at some time kT and satisfy the imposed model constraints, with the current state given as x_k [48] can be formulated as:

$$\min_{x,u} = \sum_{i=0}^{N-1} \left(\|\hat{y}_i - y_{r,k+i}\|_{Q_i}^2 + \|\hat{u}_i\|_{R_i}^2 \right) + \|\hat{y}_N - y_{r,k+N}\|_{Q_N}^2 \tag{4.20}$$

$$\begin{aligned} \text{s.t.} \quad & x_0 = \bar{x}_0 \\ & x_{k+1} = \phi(x_k, u_k) \quad \text{k=0,1,....,N-1} \\ & y_i = h(x_k, u_k) \quad \text{k=1,2,....,N} \\ & \gamma_k^l \leq M x_i \leq \gamma_k^u \qquad \text{k=0,1,....,N} \\ & \dot{\gamma}_k^l \leq u_i \leq \dot{\gamma}_k^u \qquad \text{k=0,1,....,N-1} \end{aligned} \tag{4.21}$$

where Q_i and R_i are positive semi-definite weight matrices and M is a selection matrix to only constrain the n elements of the state vector defined as:

$$M = \begin{bmatrix} 0_{n_x\text{-}n} & I_n \end{bmatrix} \tag{4.22}$$

where n_x is the number of states and 0 and I denote the null matrix and identity matrix respectively. The lower and upper bounds on the propeller forces are denoted by γ_k^l and γ_k^u respectively. The lower and upper bounds on the changing rate of forces are denoted by $\dot{\gamma}_k^l$ and $\dot{\gamma}_k^u$ respectively.

4.2 Gaussian Process

This part aims to introduce the Gaussian Process(GP) based regressions and how it is incorporated into NMPC to account for unmodelled dynamics from observed data.
The Gaussian process is a stochastic process with multivariate Gaussian probability distribution as a finite dimension distribution. The Probability Density function(PDF) by Gaussian process for a random variable x given by:

$$P_X(x) = \frac{1}{\sqrt{(2\pi)^j det(\sum)}} \exp(-\frac{1}{2}(x-\mu)^T(\sum)^{-1}(x-\mu)) \tag{4.23}$$

where, j is number of dimension, x=[x_1,x_2,....,x_j], x is a real argument, μ=[m(t_1,m(t_2,.....,m(t_j)))] $\in \mathbb{R}^{n_j}$ denotes the mean and $\sum \in \mathbb{R}^{n_j \times n_j}$ denotes the symmetric covariance matrix containing the covariance of all jointly modeled x. The Gaussian Process(GP) can be denoted by f and the Euclidean space $\in \mathbb{R}^j$ contains the index set.

$$f(x) \sim GP(\mu(x), k(x, x', \theta)) \tag{4.24}$$

Equation 4.24 denotes the scalar GP with K as the kernel function or covariance matrix, $\theta \in \mathbb{R}^n$ as the hyperparameter that the covariance matrix depends on, and μ as the mean.

4.3 Gaussian Process Regression

Let's consider a certain dataset given by (x_j,y_j) where j=0,1,2....N, x $\in \mathbb{R}$ and y $\in \mathbb{R}$ represents input and measured output, we want to model the relationship between x and y. The model developed by using GP denoted by f with the index given by the dimension of the input, the variable is given as:

$$\begin{bmatrix} y_1 \\ y_2 \\ . \\ . \\ . \\ y_N \end{bmatrix} = \begin{bmatrix} f(x_1) \\ f(x_2) \\ . \\ . \\ . \\ f(x_N) \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \epsilon_2 \\ . \\ . \\ . \\ \epsilon_N \end{bmatrix} \quad \text{where, } \begin{cases} f(x) \sim GP(0, k(x, x', \theta)) \\ \epsilon \sim N(0, \sigma_y^2 I_N) \end{cases} \tag{4.25}$$

Here, the function f is a zero mean Gaussian process with covariance K and the error is zero mean with standard deviation σ_n. y have a normal distribution with the probability density given as:

$$y \sim N(0, K(X, X^*) + \sigma_y^2 I_N) \tag{4.26}$$

where X denotes the collective input, I denotes the identity matrix, and $K(X, X^*)$ is the covariance matrix between two random variables in X. Now that the probabilistic model is given, it is possible to make predictions in another point x′. For making the prediction, the conditional probability of f(x′) needs to be computed with respect to y. The joint probability of f(x′) and y is given as:

$$\begin{bmatrix} f(x') \\ y \end{bmatrix} \sim N\left(0, \quad \begin{bmatrix} K(x', x') & K(X, x') \\ K(x', X) & K(X, X) + \sigma_y^2 \mathbb{I} \end{bmatrix}\right) \tag{4.27}$$

The conditional probability can now be computed as:

$$f(x'|y, X) \sim N\left(\bar{\mu}, \bar{\sum}\right) \tag{4.28}$$

$$\begin{cases} \bar{\mu} = K(x', X)\left(K(X, X) + \sigma_y^2 \mathbb{I}\right)^{-1} y \\ \bar{\sum} = K(x', x') - K(x', X)\left(K(X, X) + \sigma_y^2 \mathbb{I}\right)^{-1} K(x', X) \end{cases} \tag{4.29}$$

where, K(x′,X) denotes the covariance between x′ and X, and K(x′,X)=K(X,x′)^T. The f(x′) can now be estimated by a Machine Learning(ML) estimator. The estimator in the case of Gaussian coincides with the mean of conditional probability:

$$\hat{f}(x') = \bar{\mu} = K(x', X)\left(K(X, X) + \sigma_y^2 \mathbb{I}\right)^{-1} y \tag{4.30}$$

It is also noted that the estimation strongly depends on the choice of kernel function. Interested readers can read more in detail on GP and it's implementation can be read by [51] and [52].

4.3.1 Hyperparameters

As regression depends on the kernel function, the kernel function further depends on the assigned hyperparameters θ. Various methodologies exist that can be used to learn the hyperparameters, but for the scope of this thesis, the marginal likelihood maximization approach is used. The measured output y is distributed for the Gaussian case for marginal likelihood according to [52] as:

$$\log(p(y|X; \theta)) = -\frac{n}{2} y^T K^{-1} y - \frac{n}{2} \log(2\pi det(K)) \tag{4.31}$$

with,

$$K = K_\theta(X, X) + \sigma_y^2 I_y \tag{4.32}$$

The optimization problem is then formed as:

$$\theta^* = \underset{\theta}{\operatorname{argmax}} \log(p(y|X; \theta)) \tag{4.33}$$

The objective function is differentiated with respect to hyperparameters under conditions such as kernel is differentiable to set a gradient:

$$\frac{\partial \log(p(y|X; \theta))}{\partial \theta_i} = \frac{1}{2} \text{tr}\left((\alpha\alpha^T - K_\theta^{-1}) \frac{\partial K_\theta}{\partial \theta_i}\right) \tag{4.34}$$

with,

$$\alpha = K_{\theta}^{-1} y \tag{4.35}$$

The various available gradient descent algorithms are applied to obtain the hyperparameters.

4.4 Disturbance Observer(DO)

The section introduces the extension of the dynamic model to incorporate a state-of-the-art disturbance observer (DO) from the literature in the control scheme to see how the proposed solution performs compared to it. The NMPC makes its prediction based on the model; however, the proximity effects are not accounted for in the model. The DO is intended to estimate the disturbances on the wrench of the MRAV and provide that disturbance to NMPC, so NMPC receives the updated state for the propeller forces and it is possible to get more consistent closed-loop performance. The force and torque sensors can be used as well and provide more reliable measurements, but that also adds to the cost and weight of the MRAV. Therefore, a more feasible solution is to use a wrench estimator that can provide a more accurate wrench estimation.
The wrench is given by $\hat{W}_B$=$\left[\hat{f}_B^T, \hat{\tau}_B^T\right] \in \mathbb{R}^6$, where $\hat{W}_B$ are the wrench estimations, $\hat{f}_B^T$ are the estimated forces and $\hat{\tau}_B^T$ are the estimated torques on the body frame of MRAV exerted by the environment. The wrench estimation can provide sufficient accurate estimation provided accurate measurements of position, velocities, and if possible acceleration. The forces are estimated using an acceleration based observer presented by [53] and the torques are estimated using a momentum based observer by [54].

Force estimator

The forces are estimated as:

$$\begin{aligned} \dot{\hat{f}}_B &= L(f_B^T \text{-} \hat{f}_B^T) \\ \dot{\hat{f}}_B &= \text{-} f_B^T \text{+} L(m\ddot{p}_B \text{+} mge_3 - RF_1 u) \end{aligned} \tag{4.36}$$

where L is the observer gain, f_B^T is the force on the body frame of MRAV, m is mass of MRAV, g is the gravity matrix, R is the rotation matrix expressing the body frame to world frame, $F_1 \in \mathbb{R}^{3\times6}$is the matrix to account for the physical and geometric properties of MRAV and u = $[u_1,u_2,u_3,u_4,u_5,u_6]^T$ is the controller input denotes the vector that gathers the squares of the spinning velocities of the six propeller. The observer error is defined as e_L=f_B^T-$\hat{f}_B^T$. The error dynamics, when there is a constant presence or slowly varying presence of external force, can be expressed as [53]:

$$\dot{e}_f = \text{-}Le_f \tag{4.37}$$

Therefore, the error dynamics exhibit exponential convergence towards the origin for any positive-definite gain matrix L.

Torque estimator

From the equation of motion of MRAV, the angular momentum in body frame (F_B) can be expressed as:

$$q^B = J\omega_B^B \tag{4.38}$$

Differentiating W.R.T. time,

$$\dot{q}^B = J\dot{\omega}_B^B = -\omega_B^B \times J\omega_B^B + F_2 u + \tau_B^B \tag{4.39}$$

From equation 4.39, the estimated value of torque τ_B in body frame can be regarded as the residual vector.

$$\hat{\tau}_B^B = k_I \left[(q^B(t) - q^B(t_0)) + \int_{t_0}^{t} \omega_B^B \times J\omega_B^B - F_2 u - \tau_B^B \right] \tag{4.40}$$

where t denotes the current time, t_0 denotes the initial time, $F_2 \in \mathbb{R}^{3\times6}$is the matrix to account for the physical and geometric properties of MRAV, J $\in \mathbb{R}^{3\times3}$ is MRAV inertia in body frame, $\omega_B^B \in \mathbb{R}^3$ is the angular velocity in body frame and k_I is the positive definite gain matrix, assuming $\omega_B^B(t_0)$=0 $\in \mathbb{R}^3$, it follows that the $q^B(t_0)$=0_3. Upon differentiating equation 4.40 with respect to time and using equation 4.39, we obtain the following dynamics for the residual vector:

$$\dot{\hat{\tau}}_B^B = k_I(\tau_B^B - \hat{\tau}_B^B) \tag{4.41}$$

It is seen from equation 4.41 denotes the dynamics of the first-order low pass system. It's readily apparent that as time t approaches infinity, the estimated vector $\hat{\tau}_B^B$ converges to the true vector τ_B^B for any positive-definite gain matrix k_I. The selection of the k_I matrix involves a trade-off between the convergence rate and the filtering characteristics of the observer. The larger gain values result in faster convergence, whereas smaller values help in filtering out the high-frequency noise.

4.4.1 DO Implementation with NMPC

This section shows how NMPC with DO is applied to the physical model. The control scheme for the controller implemented to MRAV is shown in Figure 4.1. The first input value from NMPC is applied as u indicates $\dot{\gamma}$, and an integrator is used to convert it to propeller forces. Then the propeller forces are converted to rotor velocity, which can be applied to actuators. The states are measured by the use of sensors and sent as feedback to NMPC. The states are also sent to DO. The DO output is the estimated wrench for the disturbance containing estimated forces and torque disturbances on the body of the MRAV, and this wrench is

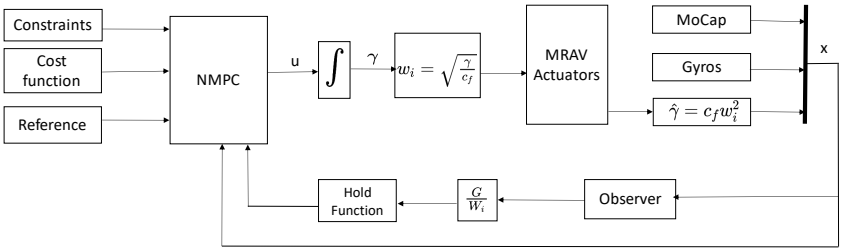


Figure 4.1: Control scheme of NMPC with DO implemented to MRAV

converted to propeller forces using G, where G is the allocation matrix. The hold function block holds the output until the observer converges, usually between 5 and 7 seconds. The output force disturbances are then added to the state responsible for the propeller forces, and NMPC takes into account the disturbances.

Chapter 5

Methodology

This chapter introduces the method containing the data-driven model used to model and compensate for the proximity effects.

5.1 Method

The method talks about the extension of the dynamic model to incorporate GP-based regression in NMPC. The GPR is intended to model the residual dynamics of the proximity effects. The predicted dynamics are incorporated into the nominal model, allowing NMPC to be aware of the disturbances. It allows NMPC to have more consistent closed-loop performance.

5.1.1 Training the GP and collecting data

In the context of dynamical systems, the feature space typically encompasses past and present inputs as well as state variables. The GP takes physical variables measured at the same time instant 'k' as input and generates predictions for the relative prediction error at the subsequent time instant 'k + 1.'
For the physical system of MRAV, states are defined as we define state variables as:

$$x_k = \Big[p, \eta, \dot{p}, \omega, \gamma\Big]^T \tag{5.1}$$

where p $\in \mathbb{R}^{n_p}$ defines the position, $\eta \in \mathbb{R}^{n_\eta}$ defines the orientation,$\dot{p} \in \mathbb{R}^{n_{\dot{p}}}$ defines the linear velocities,$\omega \in \mathbb{R}^{n_\omega}$ defines the angular velocities and γ defines the propeller forces respectively for the MRAV. With N time instants, we have the opportunity to organize the observed states and inputs into groups:

$$\begin{aligned} X &= \Big\{x_0, x_1....x_{N-1}\Big\} \\ u &= \Big\{u_0, u_1....u_{N-1}\Big\} \\ \hat{X} &= \Big\{\hat{x}_0, \hat{x}_1....\hat{x}_{N-1}\Big\} \end{aligned} \tag{5.2}$$

where $\hat{X}$ is the predicted state from the nominal model of MRAV and X and U are the actual states and control input. The needed data can be obtained by implementing the NMPC controller based on the nominal dynamics of MRAV. The GPR have been implemented on the dataset mentioned in equation 6.4. To address each component of state vectors separately, an independent GP has been employed. At each time instant t_j, the input GP vectors are denoted by s_j=[x_{j-1}, u_{j-1}]. The output of kth GP corresponding to s_j is defined as y_j^k=x_j^k-$\hat{x}_j^k$, where x_j^k and $\hat{x}_j^k$ denote the observed and predicted values of states. The GPR probabilistic model at the kth instant is as follows:

$$y^k = \begin{bmatrix} y_1^k \\ y_2^k \\ . \\ . \\ y_N^k \end{bmatrix} = \begin{bmatrix} x_1^k - \hat{x}_1^k \\ x_2^k - \hat{x}_2^k \\ . \\ . \\ x_2^k - \hat{x}_2^k \end{bmatrix} = \begin{bmatrix} \tilde{\psi}_x^k(s_1) \\ \tilde{\psi}_x^k(s_2) \\ . \\ . \\ \tilde{\psi}_x^k(s_N) \end{bmatrix} + \begin{bmatrix} \epsilon_1^k \\ \epsilon_1^k \\ . \\ . \\ \epsilon_1^k \end{bmatrix} = \tilde{\psi}_x^k + \epsilon^k \tag{5.3}$$

where, $\epsilon \sim$ N(0,$\sigma_y^2 I_N$) with σ_y^2 being variance and $\tilde{\psi}_x^k \sim$ N(0,K^k) with K^k denoting the covariance and can be defined by kernel function $K^k(x_i,x_j)$. From [52], it is stated that for the general input s_j, the posterior probability of $\tilde{\psi}_x^k$ is Gaussian in nature and thus the maximum estimator is given by the posterior mean and can be computed from equation 4.30 as:

$$\tilde{\psi}_x^k(s_j) = \Big[K^k(s_j,s_1),K^k(s_j,s_2),.....,K^k(s_j,s_N)\Big] \left(K^k + \sigma_y^2 \mathbb{I}\right)^{-1} y^k \tag{5.4}$$

The $\tilde{\psi}_x$ can be expressed as:

$$\tilde{\psi}_x(s_j) = \begin{bmatrix} \tilde{\psi}_x^1(s_j) \\ \tilde{\psi}_x^2(s_j) \\ . \\ . \\ \tilde{\psi}_x^{n_x}(s_j) \end{bmatrix} = \begin{bmatrix} K_j^1 \alpha^1 \\ K_j^2 \alpha^2 \\ . \\ . \\ K_j^{n_x} \alpha^{n_x} \end{bmatrix} \tag{5.5}$$

where α^k=$\left(K^k + \sigma_y^2 \mathbb{I}\right)^{-1} y^k$

It can be seen that the choice of kernel function is an important aspect of GPR. There are many kernels available such as Exponential kernel, Square Exponential(SE) kernel or Radial Basis Function(RBF) kernel, Matern Kernel, Polynomial kernel, and Trigonometric kernel. For the scope of this assignment, it was considered to use the SE kernel because the SE kernel is smooth and infinite times differentiable,a universal approximator and it has two hyperparameters making it easier to tune and less computationally expensive. The SE kernel is defined as:

$$K^k(\tilde{s}_i,\tilde{s}_j) = e^{-\|\tilde{s}_i - \tilde{s}_j\|_{\sum^k}} = e^{-\left(\tilde{s}_i - \tilde{s}_j\right)^T \sum^{k^{-2}} \left(\tilde{s}_i - \tilde{s}_j\right)} \tag{5.6}$$

where $\sum^k$ is a diagonal matrix with hyperparameters called lengthscales as its elements. It is seen that the SE covariance kernel evaluates the correlation based on the distance between two GP inputs. The second hyperparameter σ_y defines the tradeoff between accuracy

and smoothness. The higher values of σ_y leads to more smooth $\tilde{\psi}_x(s_j)$ but less accuracy and low values of σ_y lead to non-smooth $\tilde{\psi}_x(s_j)$. For the present framework, once the hyperparameters are selected by ML optimization and then the maximization of the marginal likelihood approach is implemented using the Python library "GPR pyTorch" to find the optimized hyperparameters.

5.1.2 GP based model

Now, once the GP model has been obtained, the discrete dynamic model of the system can be modified by including these unmodeled dynamics:

$$x_{k+1} = \phi(x_k,u_k) + \psi_x(x_k,u_k) \tag{5.7}$$

where ϕ is the nominal discrete dynamic models and ψ_x is the estimate of residual dynamics modeling proximity effects.
Following the dynamic model, the NLP for N shooting points is updated from equation 4.20 as:

$$\min_{x,u} = \sum_{i=0}^{N-1} \left(\|\hat{y}_i - y_{r,k+i}\|^2_{Q_i} + \|\hat{u}_i\|^2_{R_i} \right) + \|\hat{y}_N - y_{r,k+N}\|^2_{Q_N} \tag{5.8}$$

$$\begin{aligned} \text{s.t.} \quad & x_0 = \bar{x}_0 \\ & x_{k+1} = \phi(x_k, u_k) + \psi(x_k, u_k) \quad \text{k=0,1,....,N-1} \\ & y_i = h(x_k, u_k) \quad \text{k=1,2,....,N} \\ & \gamma_k^l \leq M x_i \leq \gamma_k^u \qquad \text{k=0,1,....,N} \\ & \dot{\gamma}_k^l \leq u_i \leq \dot{\gamma}_k^u \qquad \text{k=0,1,....,N-1} \end{aligned} \tag{5.9}$$

Then next QP constraints will be updated in equations 4.6 and 4.12 as:

$$\begin{aligned} & \Delta x_{k+1} = (A_k^m + A1_k^m)\Delta x_k + (B_k^m + B1_k^m)\Delta u_k + a_k^m, , k = 0, \ldots, N-1 \\ & A_k^m = \frac{\partial \phi_k}{\partial x_k}, \quad B_k^m = \frac{\partial \phi_k}{\partial u_k}, \quad a_k^m = \phi(x_k^m, u_k^m) + \psi(x_k, u_k) - x_{k+1}^m, \\ & A1_k^m = \frac{\partial \psi_k}{\partial x_k}, \quad B1_k^m = \frac{\partial \psi_k}{\partial u_k} \end{aligned} \tag{5.10}$$

where A1 and B1 represent the Jacobian with respect to residual dynamics. Finally, in the KKT conditions for equality constraints corresponding to the Lagrange multiplier λ and given by $\frac{\partial \phi(s^*)^T}{\partial s}$ where s contains the states and control input will be updated as $\frac{\partial \phi(s^*)^T}{\partial s} + \frac{\partial \psi(s^*)^T}{\partial s}$.

5.1.3 Implementation of GP incorporated NMPC with MRAV

This section shows how the control approach is applied to the physical model. The block diagram for the controller implemented with MRAV is shown in Figure 5.1. The NMPC gives the solution to the NLP at time instant k and it consists of the optimal values $x_{0,k}$,$x_{1,k}$,.....$x_{N,k}$ and $u_{0,k}$,$u_{1,k}$,.....,$u_{N-1,k}$. Here N shows the number of shooting points and x,u denotes the

optimal states and control input from NMPC. The first input value is applied to the system and all the values of u for different shooting point is applied to the GP prediction model to predict from trained model. As u indicates $\dot{\gamma}$, an integrator is used to convert it to propeller forces and then the propeller forces are converted to rotor velocity that can be implemented to the actuator. Then by use of sensors, the states are measured and sent as feedback to the NMPC and GP prediction model to estimate the control input at the next sample time instant.

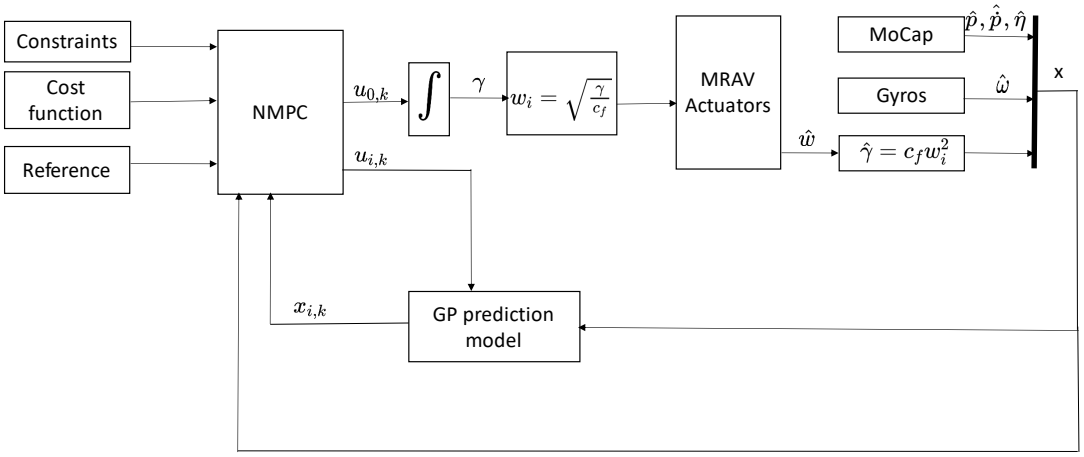


Figure 5.1: Control scheme of NMPC incorporated with GP implemented to MRAV

5.2 MATMPC

This section gives an insight in the framework used for NMPC implementation. The NMPC is implemented using the NMPC genome module developed and used previously by [55] based on a software called MATMPC [56]. Once the Optimal Control Problem(OCP) and respective Jacobian and Hessian matrices are defined using the Casadi framework [57], MATMPC uses MEX files to set up an interface between functions written as C code files. This allows MATMPC to call these functions during the real-time execution of the codes. Then these models are discretized using ERK4 for direct multiple shooting. It is followed by solving the dense QP and finding an explicit solution using solvers like HPIPM [50] and QPOASES [58] based on the Real Time Iteration(RTI) scheme.
RTI represents a cutting-edge, efficient algorithm designed for addressing Nonlinear Model Predictive Control (NMPC) problems. It operates by executing a single Sequential Quadratic Programming (SQP) iteration to solve the Nonlinear Programming (NLP) problem, meaning that the associated Quadratic Programming (QP) problem for the NLP is resolved in just one

iteration. The RTI scheme's effectiveness lies in its initial value embedding strategy, which enables it to rapidly address a series of comparable QP problems in real time, even when faced with varying initial conditions.

5.3 Gazebo

The simulations are carried out in the Gazebo environment, so this section intends to give an insight into the Gazebo simulation environment [59]. One of its notable features is its robust physics engine, which enables the MRAV to engage in physical interactions with objects present within the simulated world. Thus, the Gazebo simulation allowed us to get insight into the implementation of the proposed scheme in relatively real-life environment. The MRAV and the environment are defined in the Simulation Descriptive Format(SDF). While the controller continues to operate within MATLAB/Simulink, the physical models of both the MRAV and the surrounding environment are established and simulated in Gazebo. The rotorcraft module from Genom3 allows the interface between rotor velocities command obtained after integrating the control inputs and low-level ESCs. The commands are sent through Genom3 mrsim module that functions as low-level ESC and sent it to the mrsim-Gazebo plugin that evaluates the wrench that will be applied to the model. The rotorcraft module is further connected to POM module. The optitract Genom3 module is used to export the position and orientation data from Gazebo and to POM module of Gazebo. The POM module gives the estimates of states back to MATLAB. Interested readers can read in more detail about it in [55]. The block diagram of how NMPC interface between MATLAB/simulink, Genom3, and Gazebo is shown in Figure 5.2.

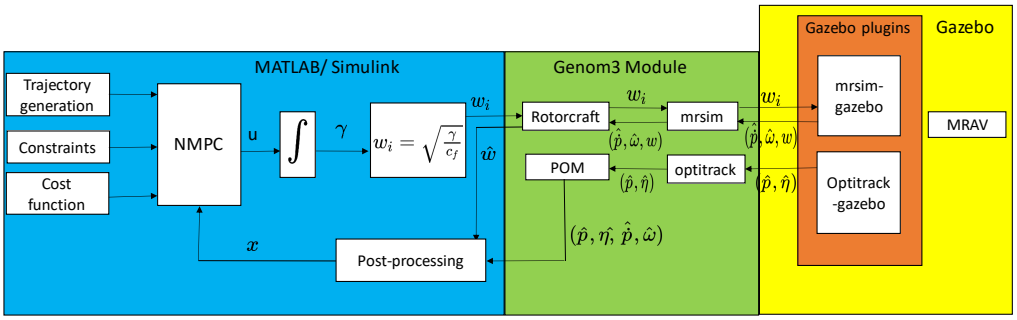


Figure 5.2: Interface of NMPC with Gazebo

Chapter 6

Results

This chapter presents the results based on the method discussed in the previous chapter. The chapter starts with the model training for the proposed method, then presents the results from the simulations of the proposed GP-based NMPC controller and its comparison with nominal NMPC, and state-of-the-art disturbance observer.

6.1 Data collection and model training

This section introduces the simulation setups and the model training to estimate the proximity effect. The simulations were performed in a Gazebo environment. In this simulation, the MRAV was commanded to hover and it was controlled with NMPC with the nominal model. Then, a virtual constant disturbance of 2.6 N was applied to propellers 3 and 6. Finally, the state and control input measurements were recorded. From the collected dataset, states, and control input were used as input to the GP model, whereas, the output was given as prediction error i.e. the difference between the predicted states and the measured states. The two different models were chosen based on the states, namely the minimized model and the extended model.

6.1.1 The minimized model

The minimized model consists of a GP model trained using position in the position vector and the control inputs as input, and the prediction errors of position vectors. The training data contained 700 points for each input and the output. The number of predictions for the NMPC controller was set to three. The inputs dataset is represented as:

$$X = \Big[p, u\Big] \tag{6.1}$$

and the output dataset is represented as:

$$\Big[p - \hat{p}\Big] \tag{6.2}$$

where $\hat{p}$ denotes the predicted position vector, U denotes the control input and p denotes the measured position vector. The control input is the time derivative of propeller forces.

6.1.2 The extended model

The extended model consists of a GP model trained using position vector, the thrusts from all the propellers and the control inputs as input, and the prediction errors of position vector and of the thrusts from respective propellers. The training data contained 140 points for each input and the output. The number of predictions for the NMPC controller was set to three. The inputs dataset is represented as:

$$X = \Big[p, \gamma, u\Big] \tag{6.3}$$

and the output dataset is represented as:

$$\Big[p - \hat{p}, \gamma - \hat{\gamma}\Big] \tag{6.4}$$

where $\hat{p}$ denotes the predicted position vector, U denotes the control input and p denotes the measured position vector. The control input is the time derivative of propeller forces. The γ denotes the respective propeller thrusts.

6.2 Comparison of GP models with NMPC

It is observed from the previous section that the training dataset does not use the complete states of the MRAV and the number of data points is limited. This is because including more states in the training dataset and the number of data points leads to the complex GP model. The complexity of the model results in more computation time for the NMPC. For NMPC to work the computation time needs to be less than the sampling time. The sampling time for this assignment was set to be 4ms. Figure 6.1 shows the spread of the computation time of

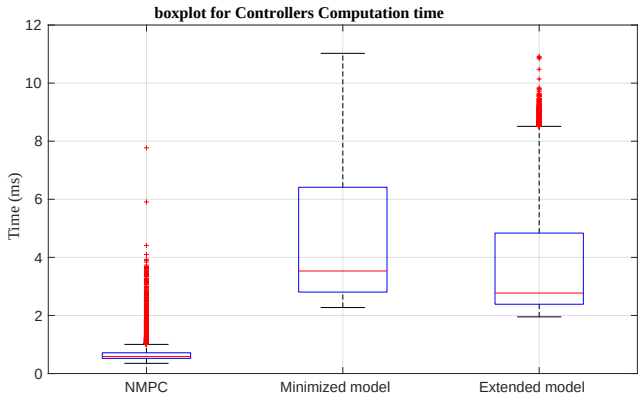


Figure 6.1: Computation time comparison of different controller

the minimized model and extended model with the NMPC. It is seen that for the extended model and the minimized model, the median and mean of the computation time are higher as compared to the NMPC. Also, the data is skewed i.e. more than fifty percent of the values of computation time are between 2ms and 4ms. If more data points are added in the minimized model and extended model then more values surpass 4ms. Similarly, if more

states are incorporated, the computation time passes the 4ms limit leading to the crash of the solver.

6.2.1 Comparison of errors

This section shows the comparison of GP models with NMPC and shows the spread of the error and thrusts. The table A.3 in appendix A.2 shows the mean, median and variation from the boxplot.

Figure 6.2a and figure 6.2b show the boxplot of position error in the x,y, and z axes. It is seen for the position error in the x,y, and z axes that the minimized model has the least median of the error compared to NMPC and the extended model. The median of the errors in the extended model is less than NMPC in the x,y, and z axes. The variability is highest for all errors in the extended model compared to the NMPC and Minimized model. The variability of error is higher in the minimized model when compared to the NMPC model in the x and y axes but that is expected as the controller is accounting for the disturbances. However, for the error in the z-axis, the variation of the minimized model is less than the NMPC.

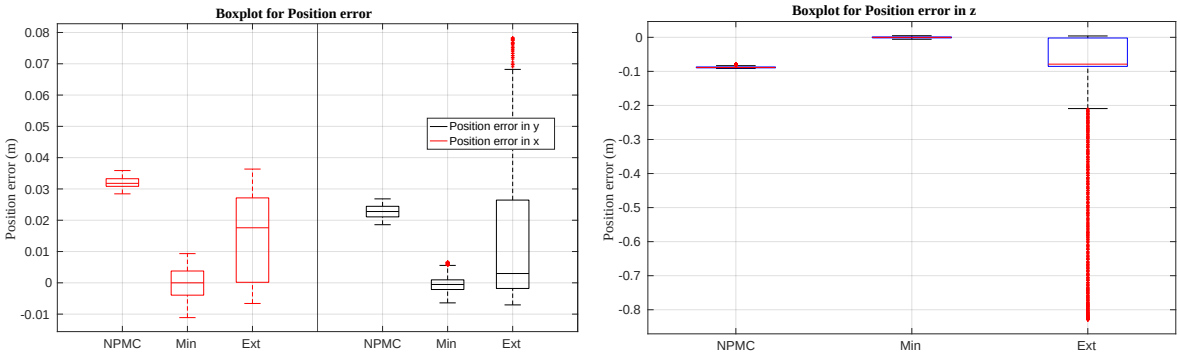
The error in the x-axis has a right-skewed distribution for the NMPC and minimized model but a left-skewed distribution for the extended model. That suggests that for the NMPC and minimized model, there are relatively fewer larger errors compared to the number of small errors and for the extended model there are relatively more large errors in the x-axis but overall the extended model has less errors than NMPC. The error in x is the least for the minimized model.

The error in the y-axis has a right-skewed distribution for the NMPC and extended model and a left-skewed distribution for the minimized model. That suggests that for the NMPC and extended model there are relatively fewer larger errors compared to the number of small errors and for the minimized model there are relatively more large errors in the y-axis but overall the minimized model has least errors compared to the NMPC and extended model.

The error in the z-axis has a left-skewed distribution for NMPC and a right-skewed distribution for the minimized model and extended model. That suggests that for NMPC, there are relatively fewer large errors compared to the number of small errors and for the minimized model and extended model, there are relatively more large errors in the z-axis but overall the errors are still least for the minimized model.

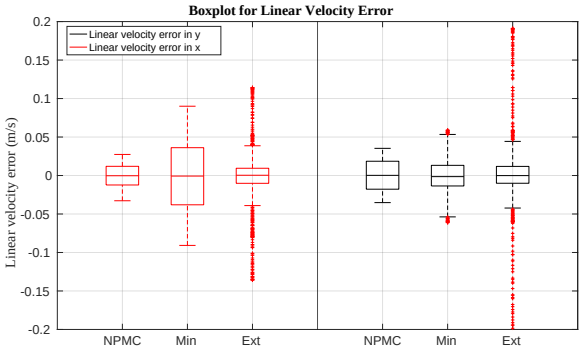
Figure 6.2c and figure 6.2d show the boxplot of linear velocity error in the x,y, and z axes. It is seen for the linear velocity error in the x and y axes that the extended model has the least median of the error compared to NMPC and the minimized model. However, for the z-axis, NMPC has the least median of errors. The median of the errors in the NMPC is less than the minimized model in the x,y, and z axes. The variability is highest for all errors in the extended model compared to the NMPC and Minimized model. The variability of error is higher in the minimized model when compared to the NMPC model in the x and y axes.

The error in the x-axis has a right-skewed distribution for the minimized model and extended model but a left-skewed distribution for the NMPC. That suggests that for the extended model

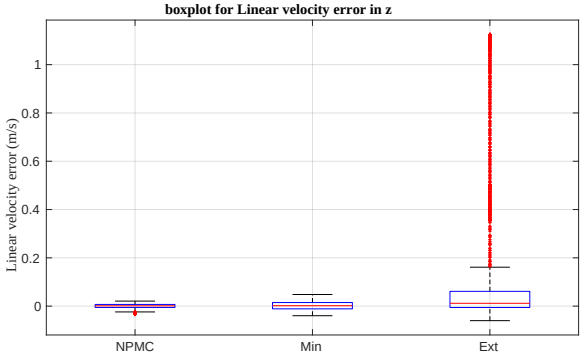


(a) Boxplot for Position Error

(b) Boxplot for Position Error



(c) Boxplot for Linear Velocity Error



(d) Boxplot for Linear Velocity Error

Figure 6.2: Boxplot for Position Error and Linear Velocity Error

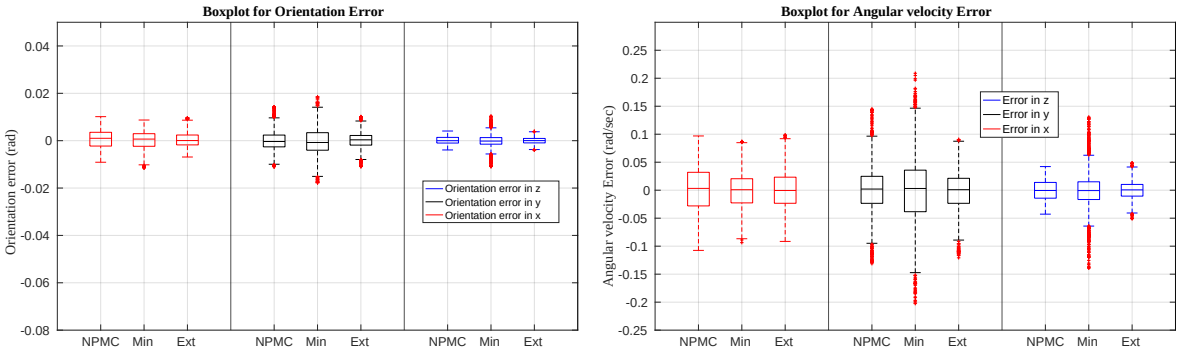
and minimized model, there are relatively fewer large errors compared to the number of small errors and for the NMPC there are relatively more large errors in the x-axis but overall extended model has similar performance to NMPC.

The error in the y-axis has a right-skewed distribution for the extended model and a left-skewed distribution for the NMPC and minimized model. That suggests that for the extended model, there are relatively fewer larger errors compared to the number of small errors and for the NMPC and minimized model, there are relatively more large errors in the y-axis but overall the NMPC has fewer errors than the minimized model and extended model although the performance of NMPC and the minimized model is close.

The error in the z-axis has a left-skewed distribution for NMPC and a right-skewed distribution for the minimized model and extended model. That suggests that for NMPC, there are relatively fewer large errors compared to the number of small errors and for the minimized model and extended model, there are relatively more large errors in the z-axis but overall the errors are still least for NMPC although the performance of the minimized model and NMPC is similar.

Figure 6.3a show the boxplot of orientation error in the x,y, and z axes. It is seen for the orientation error in the x-axis that the extended model has the least median of the error compared to NMPC and the minimized model. However, for the y and z axes, NMPC has the least median of errors. The median of the errors in the extended model is less than the minimized model in the x,y, and z axes. The variability is highest for all errors in the minimized model compared to the NMPC and extended model. The variability of error is higher

in the NMPC when compared to the extended model in the x, y, and z axes.
The error in the x-axis has a right-skewed distribution for the extended model but a left-skewed distribution for the NMPC and the minimized model. That suggests that for the extended model, there are relatively fewer large errors compared to the number of small errors and for the NMPC and the minimized model, there are relatively more large errors in the x-axis but overall extended model has the least errors.
The error in the y-axis has a right-skewed distribution for the NMPC and a left-skewed distribution for the extended model and minimized model. That suggests that for the NMPC, there are relatively fewer larger errors compared to the number of small errors and for the extended model and minimized model, there are relatively more large errors in the y-axis but overall the extended model has fewer errors than the minimized model and NMPC although the performance of NMPC and the extended model is close.
The error in the z-axis has a left-skewed distribution for the extended model and a right-skewed distribution for the minimized model and NMPC. That suggests that for the extended model, there are relatively fewer large errors compared to the number of small errors and for the minimized model and the NMPC, there are relatively more large errors in the z-axis but overall the errors are still the least for extended model although the performance of the all the controllers are similar.



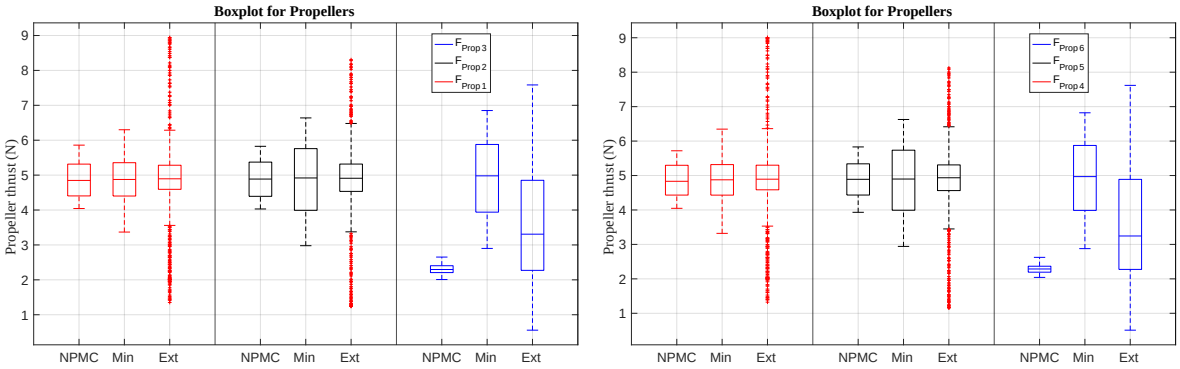
(a) Boxplot for Orientation Error

(b) Boxplot for Angular Velocity Error

Figure 6.3: Boxplot for Orientation Error and Angular Velocity Error

Figure 6.3b show the boxplot of angular velocity error in the x,y, and z axes. It is seen for the angular velocity error in the x and y axes that the extended model has the least median of the error compared to NMPC and the minimized model. However, for the z-axis, NMPC has the least median of errors although the value is close for all three controllers. The median of the errors in the extended model is less than the minimized model in the x,y, and z axes. The variability is highest for errors in the x and y axes in the NMPC compared to the minimized model and extended model. However, for the errors in the z-axis, the variability is highest in the minimized model.
The error in the x-axis has a left-skewed distribution for the NMPC, minimized model, and extended model. That suggests that for the NMPC, minimized model, and extended model, there are relatively more large errors in the x-axis but the overall minimized model has the least errors although all three controllers have similar performances.

The error in the y-axis has a right-skewed distribution for the extended model and a left-skewed distribution for the NMPC and minimized model. That suggests that for the extended model, there are relatively fewer larger errors compared to the number of small errors and for the NMPC and minimized model, there are relatively more large errors in the y-axis but overall the extended model has fewer errors than the minimized model and NMPC.
The error in the z-axis has a left-skewed distribution for the minimized model and extended model and a right-skewed distribution for the NMPC. That suggests that for the NMPC, there are relatively fewer large errors compared to the number of small errors and for the minimized model and the extended model, there are relatively more large errors in the z-axis but overall the errors are still the least for NMPC although the performance of the NMPC and extended model is similar.
Figure 6.4a and figure 6.4b show the boxplot of thrust from the propellers. The variation of the thrust is higher for the extended model and the minimized model compared to the NMPC. It can be seen that propeller 3 and propeller 6 where the constant disturbances, the extended model predicts the disturbance up to a certain extent however the minimized model can predict the constant disturbance well.



(a) Boxplot for Propeller Thrusts

(b) Boxplot for Propeller Thrusts

Figure 6.4: Boxplot for Thrusts

For NMPC, thrust from all propellers except propeller 5 has a right skew distribution. Thrust from all propellers except propeller 1 and propeller 4 have left skew distribution. For the extended model, propellers 1,3, and 6 have a right skew distribution whereas propellers 2,4 and 5 have a left skew distribution.
The table 6.1 shows the dominant frequency and the amplitude at dominant frequency for various errors and states of the MRAV with each controller respectively. The summarized data in the table 6.1 is obtained from FFT plots of errors and states shown in appendix C. It is seen that the GP-based controller has the highest dominant frequency compared to the NMPC-based controller. The minimized model has the highest dominant frequency except for orientation error in z and angular velocity error in z. The highest dominant frequency suggests more variation in the errors that is due to GP based model accounting for the disturbances. The thrusts have the same dominant frequencies. For the position error in the x,y, and z axes, the orientation error in the x-axis, and the angular velocity error in the x-axis, the amplitude is least at the dominant frequency for the minimized model. For linear velocity

Table 6.1: Data from FFT plots

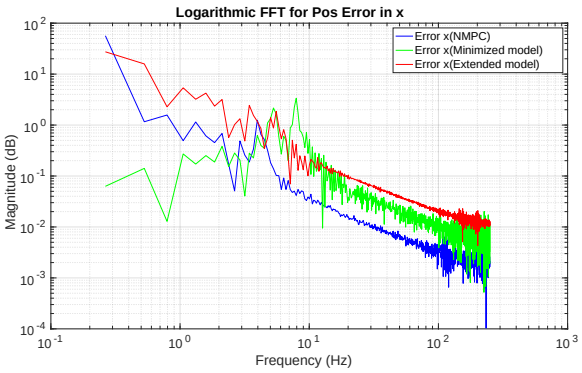
States and errors	Axis	Dominant frequency(Hz)			Peak Amplitude		
		NMPC	Min	Ext	NMPC	Min	Ext
Position error	x	0.2643	7.9281	0.2643	56.5187	3.3471	27.3815
	y	0.2643	7.9281	0.2643	39.3432	2.0830	24.1581
	z	0.2643	7.9281	0.2643	136.150	1.3502	255.91
Orientation error	x	4.4926	8.1924	5.5497	3.5254	1.6284	1.7929
	y	4.4926	5.2854	4.7569	1.6596	3.0645	1.5001
	z	4.4926	6.3425	7.3996	1.5438	6.3425	0.8337
Linear velocity error	x	3.9641	7.9281	5.5497	12.2201	32.5115	15.6833
	y	3.9641	7.9281	2.6427	18.2028	20.2206	23.0184
	z	3.9641	7.9281	0.2643	9.5728	12.5414	294.711
Angular velocity error	x	4.4926	8.1924	5.5497	38.1881	16.3701	18.8626
	y	4.4926	5.2854	4.7569	19.1199	30.9574	16.1851
	z	4.4926	6.3425	7.3996	18.0148	19.9354	12.192
Thrust	Propeller 1	0.2643	0.2643	0.2643	9246.56	9229.15	9261.74
	Propeller 2	0.2643	0.2643	0.2643	9215.06	9214.39	9275.21
	Propeller 3	0.2643	0.2643	0.2643	4365.88	9271.16	6731.96
	Propeller 4	0.2643	0.2643	0.2643	9223.85	9233.64	9241.41
	Propeller 5	0.2643	0.2643	0.2643	9218.93	9186.99	9269.35
	Propeller 6	0.2643	0.2643	0.2643	4344.27	9292.15	6726.12

errors in the x,y, and z axes, the amplitude is the least for the NMPC. For orientation error and angular velocity errors in the y and z axes, the amplitude is least for the extended model. For the thrust, it is seen that the thrust in propellers 3 and 6 has more consistent amplitude compared to other propellers for the minimized model.

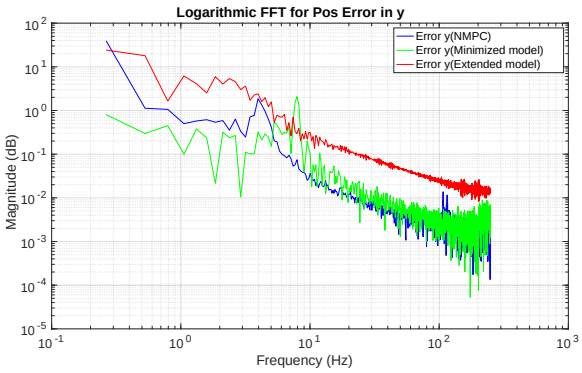
Figure 6.5 shows the logarithmic plot of FFT and cumulative frequency distribution of position error in the x,y, and z axes. From figure 6.5a, figure 6.5b, and figure 6.5c it is seen that for the low frequencies the minimized model has the least amplitude and the extended model has the highest amplitude. As the frequency increases, the minimized model has more variations in amplitude compared to the NMPC and extended model. From figure 6.5d it is seen that the fluctuation in amplitude of the minimized model does not contribute to the errors significantly as the amplitude is negligible, and the minimized model still has the least collective errors.

Figure 6.6 shows the logarithmic plot of FFT and cumulative frequency distribution of orientation error in the x,y, and z axes. From figure 6.6a, figure 6.6b, and figure 6.6c it is seen that for the low frequencies, all three controllers have similar amplitude with minimized model having more variation in amplitude for error in y and z axes and extended model having more variation in amplitude for error in x-axis. As the frequency increases, all three controllers have more variations in amplitude. From figure 6.6d, it is seen the extended model has the least collective errors for error in the x-axis and the NMPC has the least collective errors for error in y. Both NMPC and the extended model have similar performances for error in the z-axis. The most error is contributed near the dominant frequency for all three controllers and for the minimized model, error in the z-axis is contributed due to fluctuations at high frequency as well.

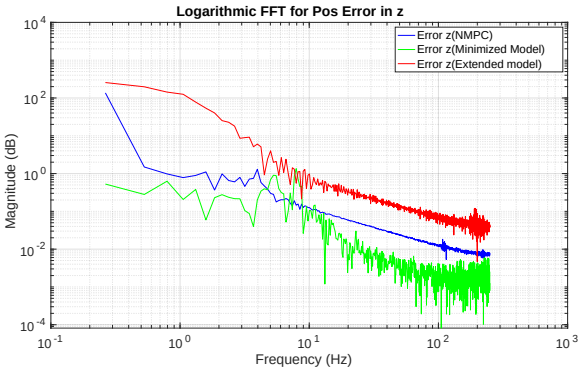
Figure 6.7 shows the logarithmic plot of FFT and cumulative frequency distribution of linear velocity error in the x,y, and z axes. From figure 6.7a, figure 6.7b, and figure 6.7c, it is seen



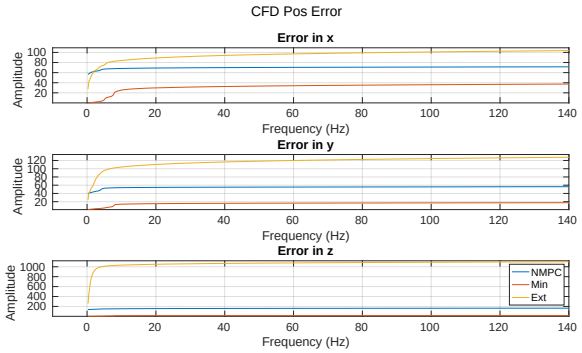
(a) Logarithmic plot of FFT for Position Error in x



(b) Logarithmic plot of FFT for Position Error in y



(c) Logarithmic plot of FFT for Position Error in z



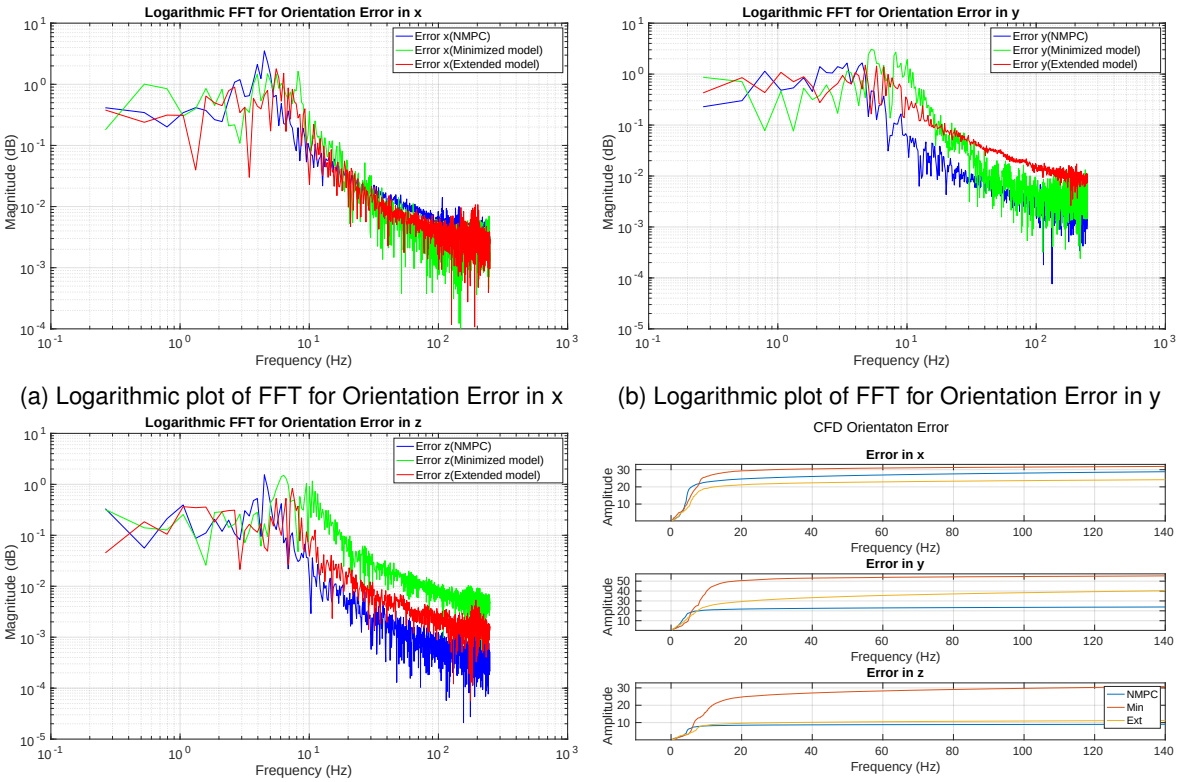
(d) CFD of Position Error

Figure 6.5: Frequency Spectrum for Position Error

that for the low frequencies the minimized model has the least amplitude and the extended model has the highest amplitude. As the frequency increases, all three controllers have more variations in amplitude. From figure 6.7d it is seen that the fluctuation in amplitude of the minimized model has the highest collective error in x-axis due to the addition of the error at the dominant frequency and NMPC has the least collective errors. For errors in the y-axis, the minimized model and NMPC have the least collective errors, and errors are similar in amplitude. For the z-axis, the extended model has the highest collective error due to the dominant peak. NMPC has the least collective errors although the performance of the minimized model is close to the performance of the NMPC. It is also seen that fluctuation in amplitude at high frequency does not contribute to the collective errors significantly.

Figure 6.8 shows the logarithmic plot of FFT and cumulative frequency distribution of angular velocity error in the x,y, and z axes. From figure 6.8a, figure 6.8b, and figure 6.8c, it is seen that as the frequency increases, all three controllers have more variations in amplitude with the highest amplitude for the minimized model. From figure 6.8d it is seen that for error in the x-axis, the fluctuation in amplitude of the minimized model and NMPC has the highest error due to the addition of the error at the dominant frequency and also fluctuations at high frequency and extended model has the least errors.

For errors in the y-axis, the NMPC has the least errors, and errors are high for the extended model and the minimized model. For the z-axis, the minimized model has the highest collective error due to the dominant peak. NMPC and extended model has the least errors and



(a) Logarithmic plot of FFT for Orientation Error in x

(b) Logarithmic plot of FFT for Orientation Error in y

(c) Logarithmic plot of FFT for Orientation Error in z

(d) CFD of Orientation Error

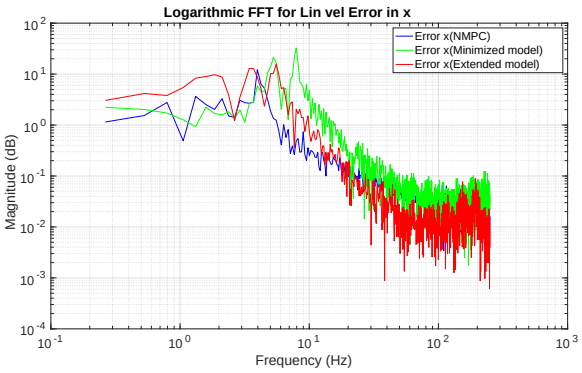
Figure 6.6: Frequency Spectrum for Orientation Error

the performance of the extended model is close to the performance of the NMPC. It is also seen that fluctuation in amplitude at high frequency for errors in x,y, and z axes is similar for all controllers.

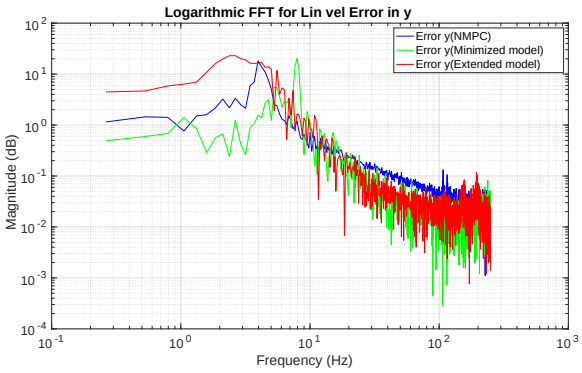
Here, after comparing all three controllers, the minimized model is chosen as it is able to estimate the disturbances and also provides good close-loop performance for the position. As it has been trained for position, it is giving good performance in position. It also provided good close loop performance in velocity for the y and z-axes and angular velocity in the x-axis. Although it is interesting to mention that the extended model also gives a good closed-loop performance, especially for the orientation, and is able to estimate the disturbances to an extent, however, it does not estimate the disturbances completely. Also, the performance for the position needs to be improved compared to the NMPC and minimized model. One possible reason for this is that the extended model is limited by training data points. It is expected that with more training data points the performance will improve. Therefore, the chosen model is the minimized model.

6.3 Comparison of GP-NMPC with DO-NMPC

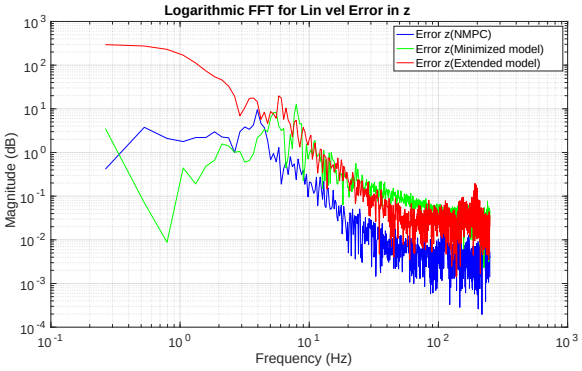
This section shows the comparison of DO with NMPC and the chosen GP model by observing the spread of the errors and thrust. Here only the performance of DO is compared to the NMPC and minimized model as the performance of NMPC and minimized model has



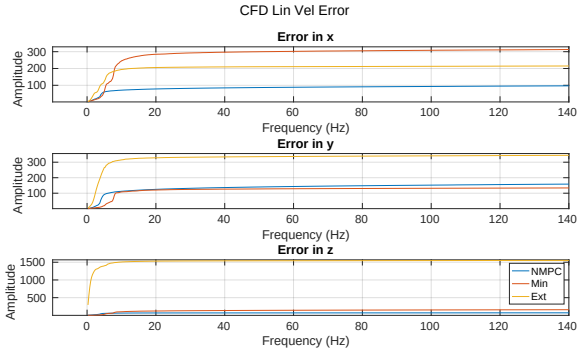
(a) Logarithmic plot of FFT for Linear Velocity Error in x



(b) Logarithmic plot of FFT for Linear Velocity Error in y



(c) Logarithmic plot of FFT for Linear Velocity Error in z



(d) CFD of Linear Velocity Error

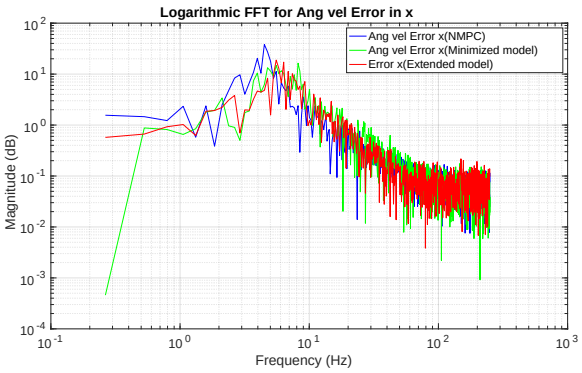
Figure 6.7: Frequency Spectrum for Linear Velocity Error

already been compared in the previous section. The table A.4 in appendix A.3 shows the mean, median and variation from the boxplot.

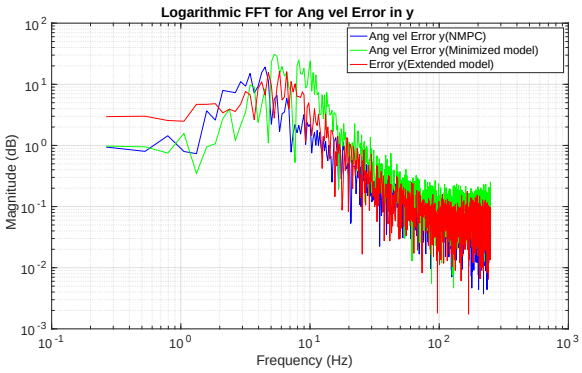
Figure 6.9a and figure 6.9b show the boxplot of position error in the x,y, and z axes. It is seen for the position error in the x,y, and z axes that the minimized model has the least median of the error but DO has less median of error than NMPC. The variation is highest in DO compared to the NMPC and minimized model. The position error for DO in the x,y, and z axes has the right skewed distribution. The overall error is highest in the DO incorporated controller.

Figure 6.9c and figure 6.9d show the boxplot of linear velocity error in the x,y, and z axes. It is seen for the linear velocity error in the x,y, and z axes that the NMPC has the least median of the error and DO has the largest median of error. The variation is highest in DO compared to the NMPC and minimized model. The linear velocity error for DO in the x, and y axes has the left skewed distribution and right skewed distribution for the z-axis. The overall error is highest in the DO incorporated controller.

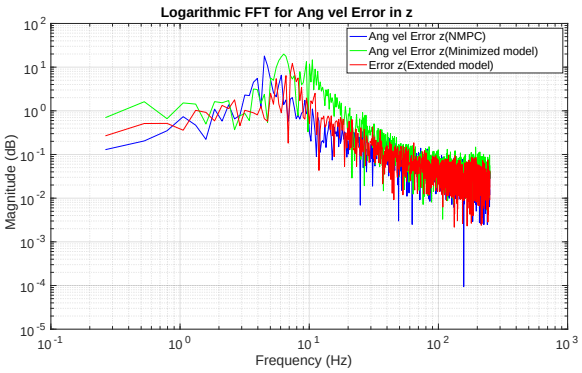
Figure 6.10a show the boxplot of orientation error in the x,y, and z axes. It is seen for the orientation error in the x and y axes that the DO has the least median of the error compared to NMPC and the minimized model. However, for the z-axis, NMPC has the lowest median of errors and DO has the highest median of error. The variation is highest in DO for error in the x-axis compared to the NMPC and minimized model. For error in y and z axes the



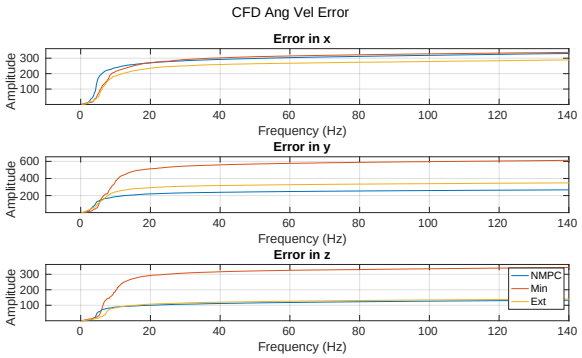
(a) Logarithmic plot of FFT for Angular Velocity Error in x



(b) Logarithmic plot of FFT for Angular Velocity Error in y



(c) Logarithmic plot of FFT for Angular Velocity Error in z

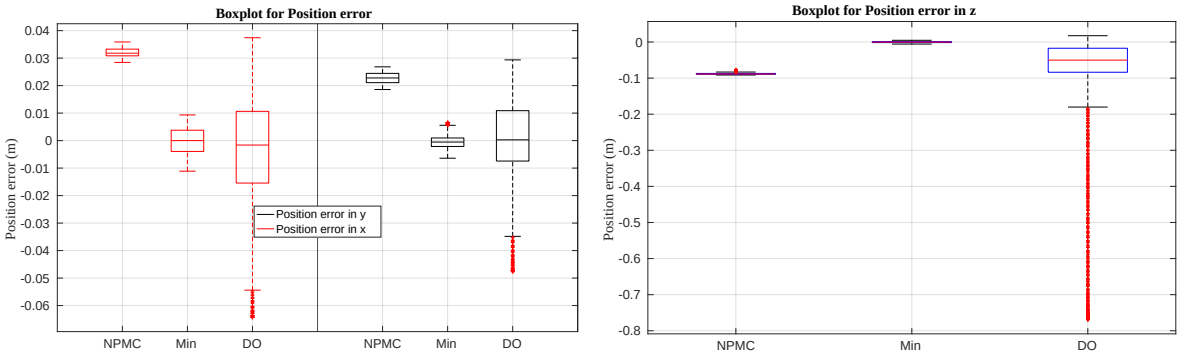


(d) CFD of Angular Velocity Error

Figure 6.8: Frequency Spectrum for Angular Velocity Error

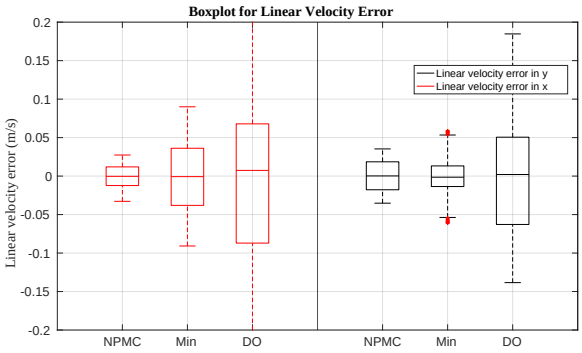
variation of DO is between NMPC and minimized model. The orientation error for DO in the x, and z axes has the left skewed distribution and right skewed distribution for the y-axis.
Figure 6.10b show the boxplot of angular velocity error in the x,y, and z axes. It is seen for the angular velocity error in the x and y axes that the DO has the median of the error between NMPC and the minimized model. However, for the z-axis, DO has the highest median of errors although the value is close for all three controllers. For error in y-axis, the median is lowest compared to NMPC and the minimized model.
The variation of DO is between variations of NMPC and minimized model for the angular velocity error in the x,y, and z axes. The angular velocity error for DO in the x, and z axes has the left skewed distribution and right skewed distribution for the y-axis. The overall error in x is highest in the NMPC incorporated controller and the minimizes model for error in y and z axes. Figure 6.11a and figure 6.11b show the boxplot of thrust from the propellers. The variation of the thrust is highest for DO incorporated model. It can be seen that in propeller 3 and propeller 6 where the constant disturbance was applied, the DO predicts the disturbances up to a certain extent better than NMPC but the minimized model still outperforms DO. The thrust distribution of all the propellers except propeller 5 has a right-skewed distribution.

Table 6.2 shows the dominant frequency and the amplitude at the dominant frequency for various errors and states of the MRAV with each controller respectively. The summarized

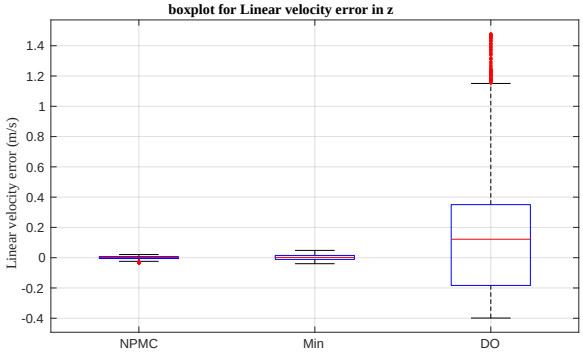


(a) Boxplot for Position Error

(b) Boxplot for Position Error



(c) Boxplot for Linear Velocity Error



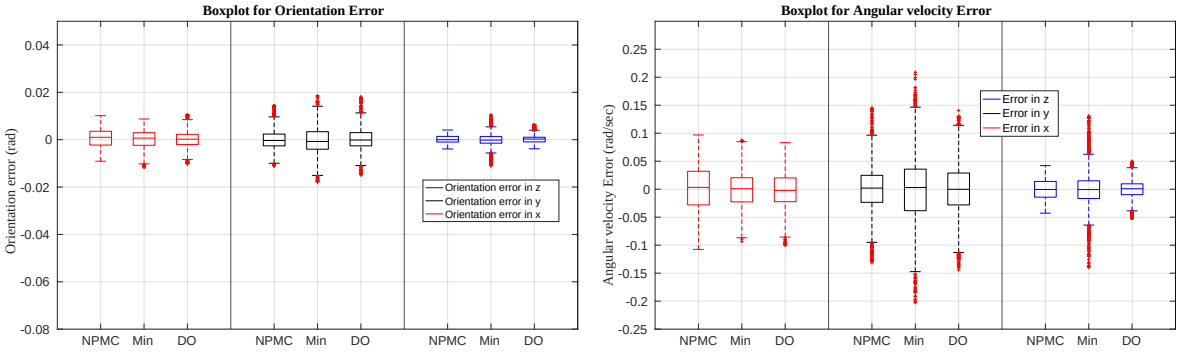
(d) Boxplot for Linear Velocity Error

Figure 6.9: Boxplot for Position Error and Linear Velocity Error

data in the table 6.2 is obtained from FFT plots of errors and states shown in appendix D. It is seen that except for the position error, the DO has the dominant frequency less than NMPC and minimized model. For position error, the dominant frequency for DO is equal to NMPC and greater than NMPC for error in the x and y axes. The dominant peak for DO is lowest for orientation error in the z-axis however for all other errors the amplitude is higher than the minimized model. For linear velocity errors in x,y, and z axes and angular velocity errors in z, the amplitude of DO is higher than NMPC as well.

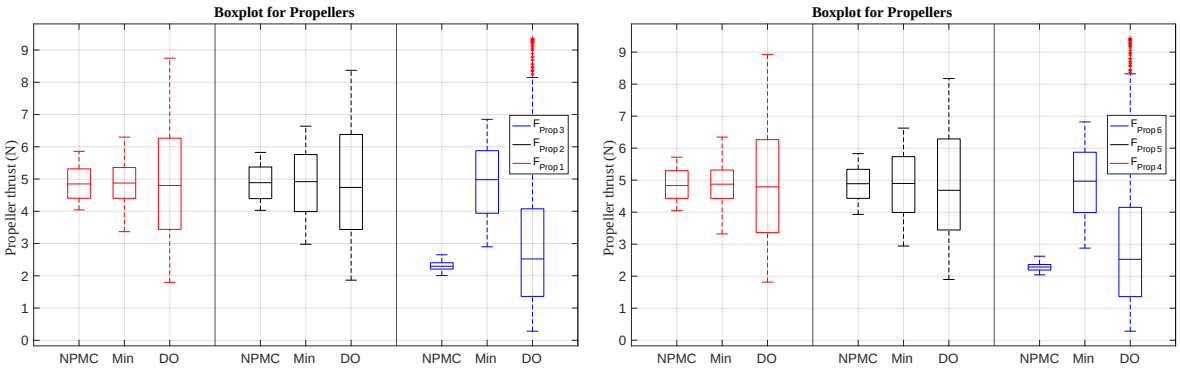
Figure 6.12 shows the logarithmic plot of FFT and cumulative frequency distribution of position error in the x,y, and z axes. From figure 6.12a, 6.12b, and figure 6.12c, it is seen that the DO has the highest amplitude of errors in the x,y, and z axes compared to NMPC and minimized model except around the dominant frequency of the minimized model in x and y axes. From figure 6.12d, it is seen that the DO has the highest cumulative error in the x,y, and z axes, and the minimized model still has the least collective errors. The most error is contributed near the dominant frequency for all three controllers, and the fluctuation is high for DO at high frequency and contributes to the error as there is an increase in error in the x,y, and z axes at high frequency.

Figure 6.13 shows the logarithmic plot of FFT and cumulative frequency distribution of orientation error in the x,y, and z axes. From figure 6.13a, 6.13b, and figure 6.13c, it is seen that for the low frequencies, all three controllers have similar amplitudes but as the frequency increases, the DO have higher variation in amplitude. From figure 6.11b, it is seen that for errors in x, DO has the least errors but for errors in the y and z axes the cumulative error is



(a) Boxplot for Orientation Error (b) Boxplot for Angular Velocity Error

Figure 6.10: Boxplot for Orientation Error and Angular velocity error



(a) Boxplot for Propeller Thrusts (b) Boxplot for Propeller Thrusts

Figure 6.11: Boxplot for Thrusts

between NMPC and minimized model.

Figure 6.14 shows the logarithmic plot of FFT and cumulative frequency distribution of the linear velocity error in the x,y, and z axes. From figure 6.14a, 6.14b, and figure 6.14c, it is seen that the DO has the highest amplitude of errors in the x,y, and z axes compared to NMPC and minimized model except around dominant frequency of the minimized model in x and y axes. From figure 6.14d, it is seen that the DO has the highest cumulative error in the x,y, and z axes compared to NMPC and minimized model. The most error is contributed near the dominant frequency for all three controllers, and the fluctuation is high for DO at high frequency and contributes to the error as there is an increase in error in the x,y, and z axes at high frequency.

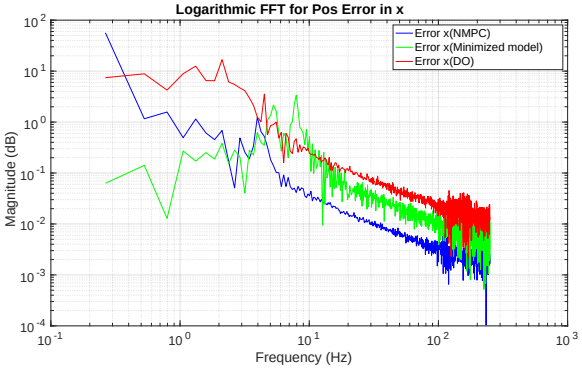
Figure 6.14 shows the logarithmic plot of FFT and cumulative frequency distribution of angular velocity error in the x,y, and z axes. From figure 6.13a, 6.13b, and figure 6.13c, it is seen that for the low frequencies, all three controllers have similar amplitudes but as the frequency increases, the DO have higher variation in amplitude similar to NMPC but less than the minimized model. From figure 6.11b, it is seen that for errors in x, DO has the least errors compared to NMPC and minimized model but for errors in the y and z axes the cumulative error is between NMPC and minimized model.

Table 6.2: Data from FFT plots (DO)

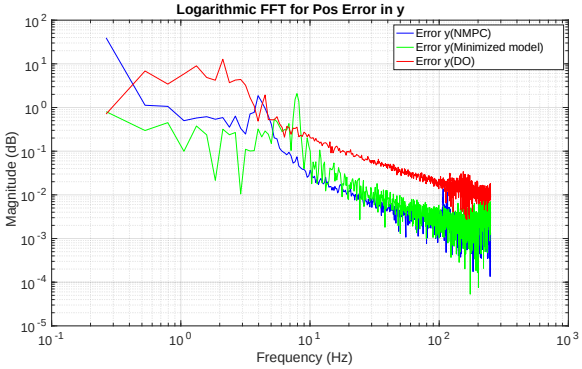
States and errors	Axis	Dominant frequency(Hz)			Peak Amplitude		
		NMPC	Min	DO	NMPC	Min	DO
Position error	x	0.2643	7.9281	2.1142	56.5187	3.3471	16.7458
	y	0.2643	7.9281	2.1142	39.3432	2.0830	12.7413
	z	0.2643	7.9281	0.2643	136.150	1.3502	258.21
Orientation error	x	4.4926	8.1924	4.2283	3.5254	1.6284	2.4154
	y	4.4926	5.2854	4.2283	1.6596	3.0645	4.1922
	z	4.4926	6.3425	3.6998	1.5438	6.3425	1.2709
Linear velocity error	x	3.9641	7.9281	2.1142	12.2201	32.5115	86.434
	y	3.9641	7.9281	2.1142	18.2028	20.2206	66.8242
	z	3.9641	7.9281	0.5285	9.5728	12.5414	343.455
Angular velocity error	x	4.4926	8.1924	4.2283	38.1881	16.3701	23.9291
	y	4.4926	5.2854	4.2283	19.1199	30.9574	39.9158
	z	4.4926	6.3425	3.6998	18.0148	19.9354	11.2254
Thrust	Propeller 1	0.2643	0.2643	0.2643	9246.56	9229.15	9282.66
	Propeller 2	0.2643	0.2643	0.2643	9215.06	9214.39	9241.26
	Propeller 3	0.2643	0.2643	0.2643	4365.88	9271.16	5461.59
	Propeller 4	0.2643	0.2643	0.2643	9223.85	9233.64	9265.16
	Propeller 5	0.2643	0.2643	0.2643	9218.93	9186.99	9222.76
	Propeller 6	0.2643	0.2643	0.2643	4344.27	9292.15	5509.77

6.4 Discussion

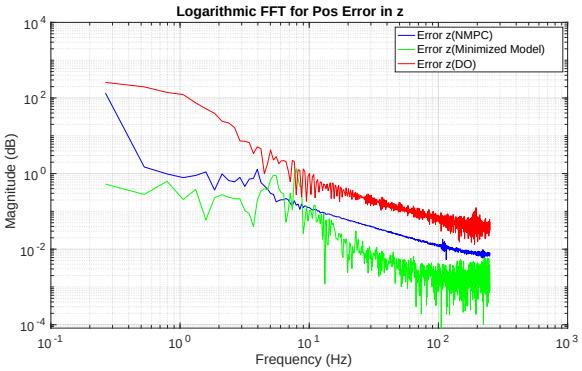
This chapter presented the results from the simulation validation in the Gazebo environment of controllers with NMPC and GP-based models. The GP-based models were classified as minimized model and extended model. It was clear that all three controllers tried to adjust for the disturbances. The minimized model-based controller was chosen based on the results. While implementing the controllers the essence of the assignment was maintained. Then, the performance of the chosen GP based controller was compared with the state-of-the-art disturbance observer-based controller. The Disturbance controller performed better for orientation and angular velocity in the x,y, and z axes. It was seen that for position and velocity in the x, y, and z axes, the chosen controller outperformed the disturbance observer based controller. The disturbance observer based controller tried to estimate the disturbances on propeller 3 and propeller 6 however it was only able to estimate to a certain extent compared to the chosen controller.



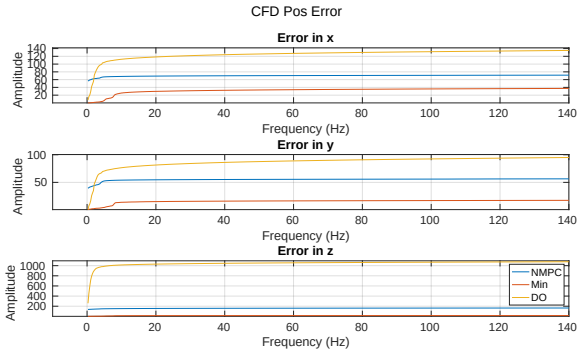
(a) Logarithmic plot of FFT for Position Error in x



(b) Logarithmic plot of FFT for Position Error in y

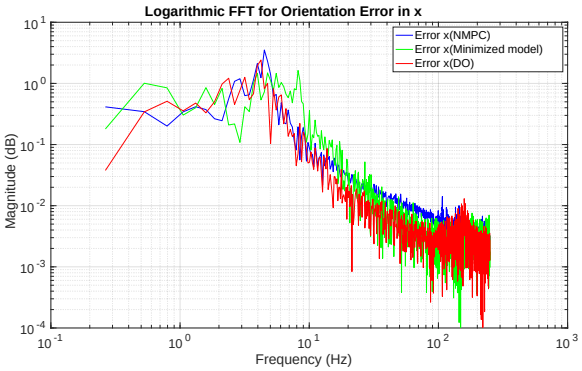


(c) Logarithmic plot of FFT for Position Error in z

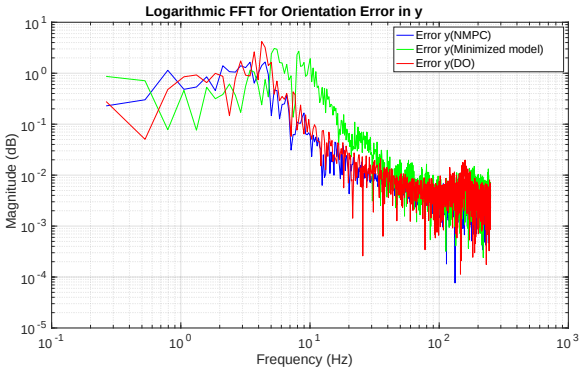


(d) CFD of Position Error

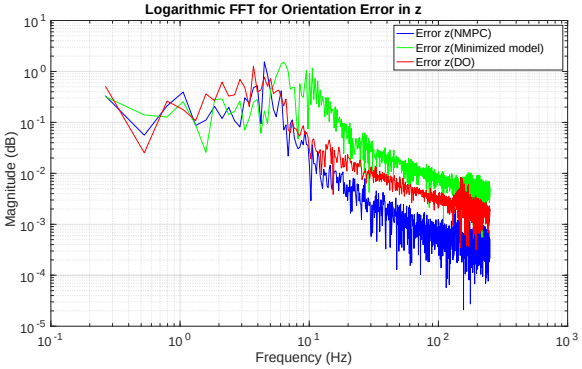
Figure 6.12: Frequency Spectrum for Position Error



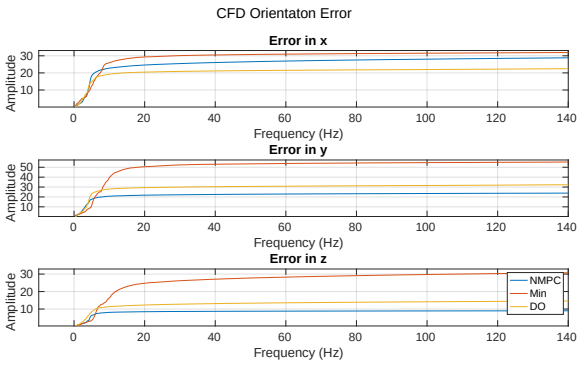
(a) Logarithmic plot of FFT for Orientation Error in x



(b) Logarithmic plot of FFT for Orientation Error in y

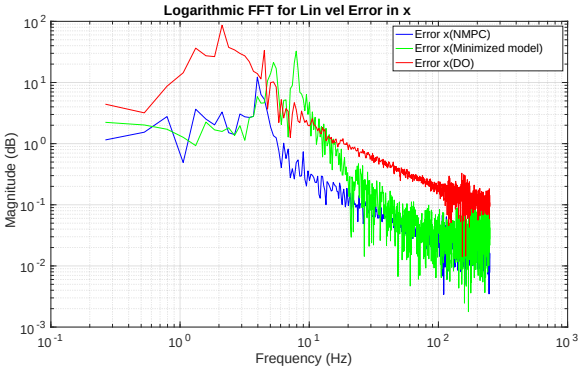


(c) Logarithmic plot of FFT for Orientation Error in z

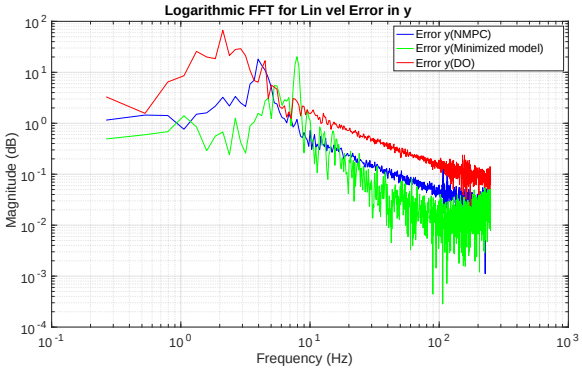


(d) CFD of Orientation Error

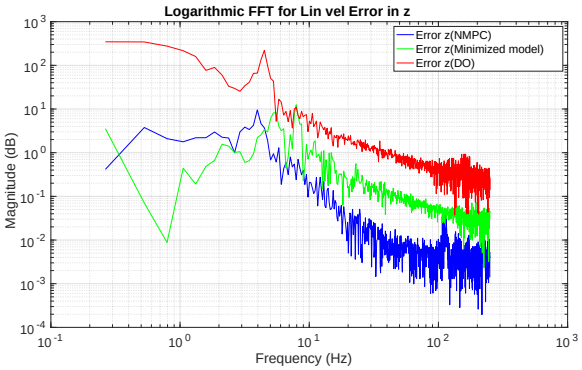
Figure 6.13: Frequency Spectrum for Orientation Error



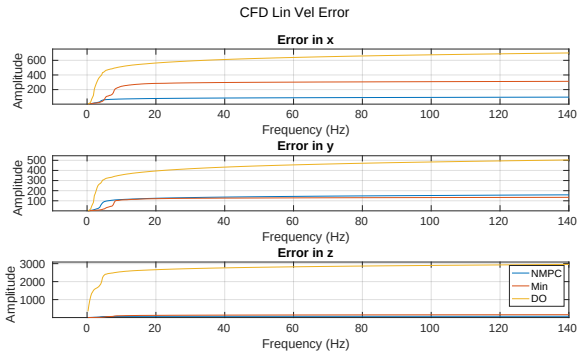
(a) Logarithmic plot of FFT for Linear Velocity Error in x



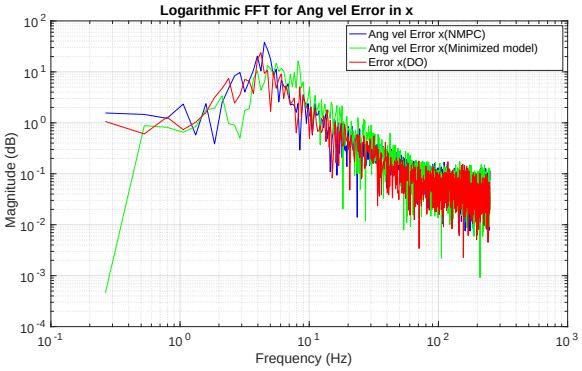
(b) Logarithmic plot of FFT for Linear Velocity Error in y



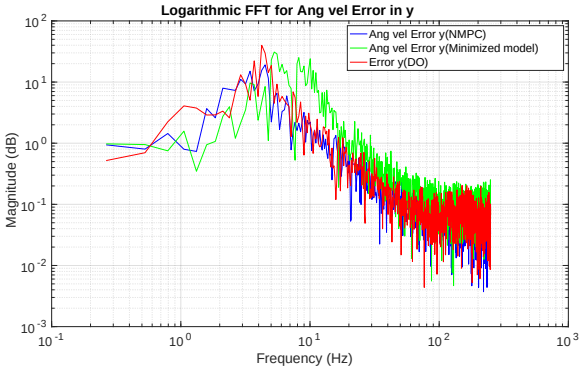
(c) Logarithmic plot of FFT for Linear Velocity Error in z



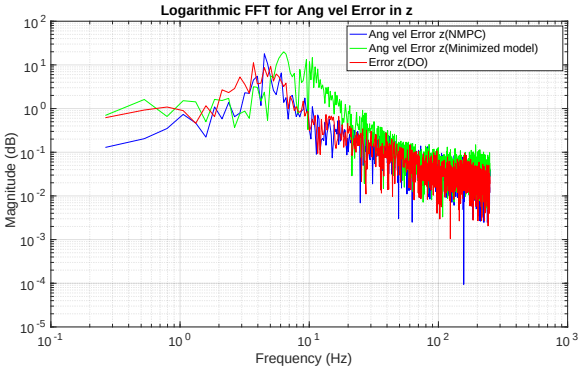
(d) CFD of Linear Velocity Error



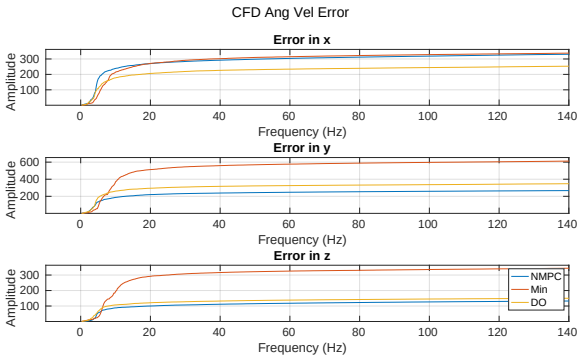
(e) Logarithmic plot of FFT for Angular Velocity Error in x



(f) Logarithmic plot of FFT for Angular Velocity Error in y



(g) Logarithmic plot of FFT for Angular Velocity Error in z



(h) CFD of Angular Velocity Error

Figure 6.14: Frequency Spectrum for Velocity Error

Chapter 7

Conclusions and recommendations

This chapter consists of two sections. The first section concludes with an overview of the work done during the thesis while answering the research questions. The second section presents future recommendations based on the presented work and the current limitations of the work.

7.1 Conclusions

In this section, the research questions that are proposed in chapter 1 are reviewed, and the summary of the work and related conclusions are presented. The primary research question is given by:

Can data-driven model improve the performance of model-based controllers in confined environment?
The data-driven model has potential to improve the performance of model-based controllers in a confined environment. This conclusion is based on the performance of the GP-based models incorporated with the NMPC controller to estimate and eliminate the disturbances induced by air flows interacting with the obstacles in close proximity in simulation environment. Further, the comparison of GP-based models with the NMPC and state-of-the-art disturbance observer based controllers has been discussed in the chapter 6. It is seen that data-driven controllers can improve performance with sufficient training data points. The primary advantage of using the data-driven model is that the states that are actually responsible can be considered as GPs are independent of each other. That means if only the position is interesting and MRAV needs to move in a confined environment, then the GP-based model can only be incorporated for the position as was shown in the minimized model.
Now that the primary question is answered, the sub-research questions are discussed that were used to achieve the above conclusion.

1. **How to model these erratic disturbances on MRAVs in a confined environment?**
   It was shown that the disturbances were modeled using Gaussian Process-based Re-

gression(GPR). GPR combined prior and the likelihood probabilistic functions based on training samples giving a probabilistic approach to prediction by giving mean and standard deviation as output. Then the mean is used as a prediction model for the error as seen in chapter 5.

2. **How to extend the dynamic model of MRAVs to account for these disturbances induced by the air flows in confined environments and How to extend the existing controller to ensure accurate trajectory tracking in confined environments?** The second and third sub-questions are concluded together as they are dependent on each other. It was seen that once the prediction model is obtained, it is included with the dynamic model of the MRAV in the system constraints while formulating the Nonlinear Programming(NLP) problem. The controller was extended by updating the NLP problem and its transformation into the Sequential Quadratic Problem (SQP). Further, the Jacobians used to convert the NLP into the sequential quadratic problem were extended to include the prediction model of the errors and the controller. This allows the NMPC to take into account the disturbances when solving SQP while imposing constraints on the system. Finally, the solver was able to solve the updated problem and provide the optimal control input that was transformed into velocities given as input to MRAV.

7.2 Limitation and Future Recommendations

This section introduces the current limitation of the work and future recommendations on how it can be improved. The limitations of the current work are:

1. The current control scheme has been validated for the constant disturbances in the simulation environment. The control scheme needs to be tested for the more real life scenario where proximity effects are erratic in nature as seen in chapter 3.

2. The control scheme has yet to be validated experimentally to get more real insight into the potential and the limitations of the controller.

3. The current model has only been tested for the hovering mode of the MRAV. So, the controller has only been tested for position control, and is yet to be determined what will be the impact of added rotation in MRAV maneuvering.

4. The GP-based model requires more computation time as compared to NMPC. Also, as more GPs are introduced to include more states the complexity of the GP-based model is increased as seen in chapter 6. That limits the training of the GP-based model and that affects the performance of the GP-based model as seen for the extended model.

Based on the work done in the assignment and the current limitations, following recommendations can be made:

1. The experimental validation of the control scheme to get more real insight into the potential and the limitations of the controller.

2. Bayesian optimization can be investigated for hyperparameter tuning and ensembling techniques to combine multiple GP-based regression models to enhance the proposed methodology.

3. The current controller framework uses a combination of Matlab, C++, and Python. Therefore, the GP-based controller can be implemented completely in C++/python to reduce the computation burden as they are better suited for computation purpose than Matlab.

4. New research on Real-time Neural MPC [60] shows faster results than previously Learned model-based MPC including GP based model, that can be explored.

5. At present, the GP models are being computed in series. Finding ways for parallelization of the computation process for each GP as all GPs are independent of each other, can also reduce the computation burden and allow the incorporation of more prediction error for more states in the controller.

Bibliography

[1] A. Matus-Vargas, G. Rodriguez-Gomez, and J. Martinez-Carranza, "Ground effect on rotorcraft unmanned aerial vehicles: a review," *Intelligent Service Robotics*, vol. 14, no. 1, pp. 99–118, Jan. 2021. [Online]. Available: https://doi.org/10.1007/s11370-020-00344-5

[2] P. E. Tanner, A. D. Overmeyer, L. N. Jenkins, Yao, and S. M. Bartram, "Experimental investigation of rotorcraft outwash in ground effect." [Online]. Available: https://ntrs.nasa.gov/api/citations/20160006428/downloads/20160006428.pdf

[3] CHI HEI VONG, "Control and aerodynamic analysis of quadcopters in confined spaces," 2021. [Online]. Available: https://bridges.monash.edu/articles/thesis/Control_and_Aerodynamic_Analysis_of_Quadcopters_in_Confined_Spaces/14583393/1

[4] "Darpa subterranean challenge." [Online]. Available: https://www.subtchallenge.com/

[5] D. C. Robinson, H. Chung, and K. Ryan, "Computational investigation of micro rotorcraft near-wall hovering aerodynamics," in *2014 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, May 2014. [Online]. Available: https://doi.org/10.1109/icuas.2014.6842357

[6] David C. Robinson, Hoam Chung and Kris Ryan, "Numerical investigation of a hovering micro rotor in close proximity to a ceiling plane," *Journal of Fluids and Structures*, vol. 66, pp. 229–253, Oct. 2016. [Online]. Available: https://doi.org/10.1016/j.jfluidstructs.2016.08.001

[7] P. Sanchez-Cuevas, G. Heredia, and A. Ollero, "Characterization of the aerodynamic ground effect and its influence in multirotor control," *International Journal of Aerospace Engineering*, vol. 2017, pp. 1–17, 2017. [Online]. Available: https://doi.org/10.1155/2017/1823056

[8] Albert B, "The ground effect on lifting propellers. technical report 835, National Advisory Committee for Aeronautics, Washington, DC, USA,1937." [Online]. Available: https://ntrs.nasa.gov/citations/19930094580

[9] Knight and Hefner(1941), "Analysis of ground effect on the lifting airscrew. technical report 835, National Advisory Committee for Aeronautics, Washington, DC, USA." [Online]. Available: http://hdl.handle.net/2060/19930081580

[10] Cheeseman and Bennett(1955), “The effect of the ground on a helicopter rotor in forward flight.” [Online]. Available: https://reports.aerade.cranfield.ac.uk/handle/1826.2/3590

[11] E. A. Fradenburgh, “The helicopter and the ground effect machine,” *Journal of the American Helicopter Society*, vol. 5, no. 4, pp. 24–33, Oct. 1960. [Online]. Available: https://doi.org/10.4050/jahs.5.4.24

[12] B. Kutz, U. Kowarsch, M. Kessler, and E. Kraemer, “Numerical investigation of helicopter rotors in ground effect,” in *30th AIAA Applied Aerodynamics Conference*. American Institute of Aeronautics and Astronautics, Jun. 2012. [Online]. Available: https://doi.org/10.2514/6.2012-2913

[13] T. Belatti, “Quadrotor flight in constrained environments NSF summer undergraduate fellowship in sensor technologies,” 2012.

[14] C. Powers, D. Mellinger, A. Kushleyev, B. Kothmann, and V. R. Kumar, “Influence of aerodynamics and proximity effects in quadrotor flight,” in *International Symposium on Experimental Robotics*, 2012, pp. 289–302.

[15] G. M. Eberhart, “Modeling of ground effect benefits for multi-rotor small unmanned aerial systems at hover,” Master’s thesis, 2017. [Online]. Available: https://etd.ohiolink.edu/apexprod/rws_etd/send_file/send?accession=ohiou1502802483367365&disposition=inline

[16] P. J. Sanchez-Cuevas, G. Heredia, and A. Ollero, “Experimental approach to the aerodynamic effects produced in multirotors flying close to obstacles,” in *ROBOT 2017: Third Iberian Robotics Conference*. Springer International Publishing, Nov. 2017, pp. 742–752. [Online]. Available: https://doi.org/10.1007/978-3-319-70833-1_60

[17] P. J. Sanchez-Cuevas, V. Martín, G. Heredia, and A. Ollero, “Aerodynamic effects in multirotors flying close to obstacles: Modelling and mapping,” in *Advances in Intelligent Systems and Computing*. Springer International Publishing, Nov. 2019, pp. 63–74. [Online]. Available: https://doi.org/10.1007/978-3-030-35990-4_6

[18] S. A. Conyers, M. J. Rutherford, and K. P. Valavanis, “An empirical evaluation of ground effect for small-scale rotorcraft,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, May 2018. [Online]. Available: https://doi.org/10.1109/icra.2018.8461035

[19] S. A. Conyers, “Empirical evaluation of ground, ceiling, and wall effect for small-scale rotorcraft (2019). electronic theses and dissertations. 1570,” Ph.D. dissertation, University of Denver. [Online]. Available: https://digitalcommons.du.edu/etd/1570

[20] H. Keshavarzian and K. Daneshjou, “Modified under-actuated quadrotor model for forwarding flight in the presence of ground effect,” *Aerospace Science*

and Technology, vol. 89, pp. 242–252, Jun. 2019. [Online]. Available: https://doi.org/10.1016/j.ast.2019.04.001

[21] X. Kan, J. Thomas, H. Teng, H. G. Tanner, V. Kumar, and K. Karydis, "Analysis of ground effect for small-scale UAVs in forward flight," *IEEE Robotics and Automation Letters*, vol. 4, no. 4, pp. 3860–3867, Oct. 2019. [Online]. Available: https://doi.org/10.1109/lra.2019.2929993

[22] S. A. Conyers, M. J. Rutherford, and K. P. Valavanis, "An empirical evaluation of ceiling effect for small-scale rotorcraft," in *2018 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, Jun. 2018. [Online]. Available: https://doi.org/10.1109/icuas.2018.8453469

[23] P. J. Sanchez-Cuevas, G. Heredia, and A. Ollero, "Multirotor UAS for bridge inspection by contact using the ceiling effect," in *2017 International Conference on Unmanned Aircraft Systems (ICUAS)*. IEEE, Jun. 2017. [Online]. Available: https://doi.org/10.1109/icuas.2017.7991412

[24] X. Wang, S. Du, and Y. Liu, "Research on ceiling effect of quadrotor," in *2017 IEEE 7th Annual International Conference on CYBER Technology in Automation, Control, and Intelligent Systems (CYBER)*. IEEE, Jul. 2017. [Online]. Available: https://doi.org/10.1109/cyber.2017.8446371

[25] Y. H. Hsiao and P. Chirarattananon, "Ceiling effects for surface locomotion of small rotorcraft," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, Oct. 2018. [Online]. Available: https://doi.org/10.1109/iros.2018.8593726

[26] N. Guenard, T. Hamel, and L. Eck, "Control laws for the tele operation of an unmanned aerial vehicle known as an x4-flyer," in *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, Oct. 2006. [Online]. Available: https://doi.org/10.1109/iros.2006.282432

[27] D. Lee, H. J. Kim, and S. Sastry, "Feedback linearization vs. adaptive sliding mode control for a quadrotor helicopter," *International Journal of Control, Automation and Systems*, vol. 7, no. 3, pp. 419–428, May 2009. [Online]. Available: https://doi.org/10.1007/s12555-009-0311-8

[28] B. Hu, L. Lu, and S. Mishra, "Fast, safe and precise landing of a quadrotor on an oscillating platform," in *2015 American Control Conference (ACC)*. IEEE, Jul. 2015. [Online]. Available: https://doi.org/10.1109/acc.2015.7171928

[29] P. Wei, S. N. Chan, S. Lee, and Z. Kong, "Mitigating ground effect on mini quadcopters with model reference adaptive control," *International Journal of Intelligent Robotics and Applications*, vol. 3, no. 3, pp. 283–297, Jul. 2019. [Online]. Available: https://doi.org/10.1007/s41315-019-00098-z

[30] S. Waheed, A. U. Awan, K. F. A. Khan, and M. Liaquat, “Robust tracking of quadrotor UAV using RISE controller with input saturation,” in *2015 34th Chinese Control Conference (CCC)*. IEEE, Jul. 2015. [Online]. Available: https://doi.org/10.1109/chicc.2015.7260499

[31] B. B. Kocer, T. Tjahjowidodo, and G. G. L. Seet, “Centralized predictive ceiling interaction control of quadrotor VTOL UAV,” *Aerospace Science and Technology*, vol. 76, pp. 455–465, May 2018. [Online]. Available: https://doi.org/10.1016/j.ast.2018.02.020

[32] B. B. Kocer, M. E. Tiryaki, M. Pratama, T. Tjahjowidodo, and G. G. L. Seet, “Aerial robot control in close proximity to ceiling: A force estimation-based nonlinear mpc,” 2019. [Online]. Available: https://arxiv.org/abs/1907.13594

[33] C. H. Vong, K. Ryan, and H. Chung, “Integral backstepping position control for quadrotors in tunnel-like confined environments,” in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, May 2019. [Online]. Available: https://doi.org/10.1109/icra.2019.8793893

[34] D. Robinson, “Modelling and estimation of aerodynamic disturbances acting on a hovering micro helicopter in close proximity to planar surfaces,” 2017. [Online]. Available: https://bridges.monash.edu/articles/thesis/Modelling_and_estimation_of_aerodynamic_disturbances_acting_on_a_hovering_micro_helicopter_in_close_proximity_to_planar_surfaces/4713655

[35] H. Du, Z. Pu, J. Yi, and H. Qian, “Advanced quadrotor takeoff control based on incremental nonlinear dynamic inversion and integral extended state observer,” in *2016 IEEE Chinese Guidance, Navigation and Control Conference (CGNCC)*. IEEE, Aug. 2016. [Online]. Available: https://doi.org/10.1109/cgncc.2016.7829076

[36] X. He, M. Calaf, and K. K. Leang, “Modeling and adaptive nonlinear disturbance observer for closed-loop control of in-ground-effects on multi-rotor UAVs,” in *Volume 3: Vibration in Mechanical Systems Modeling and Validation Dynamic Systems and Control Education Vibrations and Control of Systems Modeling and Estimation for Vehicle Safety and Integrity Modeling and Control of IC Engines and Aftertreatment Systems Unmanned Aerial Vehicles (UAVs) and Their Applications Dynamics and Control of Renewable Energy Systems Energy Harvesting Control of Smart Buildings and Microgrids Energy Systems*. American Society of Mechanical Engineers, Oct. 2017. [Online]. Available: https://doi.org/10.1115/dscc2017-5210

[37] X. He, G. Kou, M. Calaf, and K. K. Leang, “In-ground-effect modeling and nonlinear-disturbance observer for multirotor unmanned aerial vehicle control,” *Journal of Dynamic Systems, Measurement, and Control*, vol. 141, no. 7, May 2019. [Online]. Available: https://doi.org/10.1115/1.4043221

[38] H. Nobahari and A. Sharifi, “Continuous ant colony filter applied to online estimation and compensation of ground effect in automatic landing of quadrotor,” *Engineering Applications of Artificial Intelligence*, vol. 32, pp. 100–111, Jun. 2014. [Online]. Available: https://doi.org/10.1016/j.engappai.2014.03.004

[39] J. S. Hayden, “The effect of the ground on helicopter hovering power required,” 1976. [Online]. Available: https://api.semanticscholar.org/CorpusID:115406795

[40] C. D. McKinnon, “Data driven, force based interaction for quadrotors,” Master’s thesis, 2015. [Online]. Available: http://hdl.handle.net/1807/70478

[41] G. Shi, X. Shi, M. O'Connell, R. Yu, K. Azizzadenesheli, A. Anandkumar, Y. Yue, and S.-J. Chung, “Neural lander: Stable drone landing control using learned dynamics,” in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, May 2019. [Online]. Available: https://doi.org/10.1109/icra.2019.8794351

[42] H. From, “Predicting the ground effect in drone landing with online learning,” Master’s thesis, KTH, School of Electrical Engineering and Computer Science (EECS), 2019. [Online]. Available: http://www.diva-portal.org/smash/record.jsf?pid=diva2%3A1383531&dswid=4761

[43] N. Mohajerin and S. L. Waslander, “Modelling a quadrotor vehicle using a modular deep recurrent neural network,” *2015 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 376–381, 2015.

[44] G. Xu, Z. Sun, H. Liu, Y. Zhou, X. Gong, and S. Gong, “Identification of ground effect and intelligent control of unmanned aerial vehicles,” *Aerospace Science and Technology*, vol. 131, p. 107976, Dec. 2022. [Online]. Available: https://doi.org/10.1016/j.ast.2022.107976

[45] M. Edmonds and J. Yi, “Learning-based near-surface modeling for predictive multirotor landing control,” *IFAC-PapersOnLine*, vol. 54, no. 20, pp. 711–716, 2021. [Online]. Available: https://doi.org/10.1016/j.ifacol.2021.11.255

[46] M. A. Hady, B. B. Kocer, H. Kandath, and M. Pratama, “Real-time uav complex missions leveraging self-adaptive controller with elastic structure,” 2019. [Online]. Available: https://arxiv.org/abs/1907.08619

[47] E. Picotti, A. D. Libera, R. Carli, and M. Bruschetta, “LbMATMPC: an open-source toolbox for gaussian process modeling within learning-based nonlinear model predictive control,” in *2022 European Control Conference (ECC)*. IEEE, Jul. 2022. [Online]. Available: https://doi.org/10.23919/ecc55457.2022.9838016

[48] D. Bicego, “Design and control of multi-directional thrust multi-rotor aerial vehicles with applications to aerial physical interaction tasks,” in *Automatic. INSA de Toulouse, 2019. English.NNT : 2019ISAT0025. tel-02433940v2*. HAL, 2020. [Online]. Available: https://hal.laas.fr/tel-02433940v2

[49] R. Quirynen, “Numerical simulation methods for embedded optimization,” 2017. [Online]. Available: http://rgdoi.net/10.13140/RG.2.2.16335.28323

[50] G. Frison and M. Diehl, “Hpipm: a high-performance quadratic programming framework for model predictive control,” 2020. [Online]. Available: https://arxiv.org/abs/2003.02547

[51] J. Wang, “An intuitive tutorial to gaussian processes regression,” 2020.

[52] F. Diego, “Application of generalized kernels to learning-based non-linear model predictive control,” Master’s thesis, 2021. [Online]. Available: https://hdl.handle.net/20.500.12608/40466

[53] B. Yuksel, C. Secchi, H. H. Bulthoff, and A. Franchi, “A nonlinear force observer for quadrotors and application to physical interactive tasks,” in *2014 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*. IEEE, Jul. 2014. [Online]. Available: https://doi.org/10.1109/aim.2014.6878116

[54] A. de Luca and R. Mattone, “Sensorless robot collision detection and hybrid force/motion control,” in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*. IEEE. [Online]. Available: https://doi.org/10.1109/robot.2005.1570247

[55] A. Alharbat, “Exploiting nonlinear model predictive control in contact-based aerial physical interaction,” October 2021. [Online]. Available: http://essay.utwente.nl/89141/

[56] Y. Chen, M. Bruschetta, E. Picotti, and A. Beghi, “MATMPC - a MATLAB based toolbox for real-time nonlinear model predictive control,” in *2019 18th European Control Conference (ECC)*. IEEE, Jun. 2019. [Online]. Available: https://doi.org/10.23919/ecc.2019.8795788

[57] J. A. E. Andersson, J. Gillis, G. Horn, J. B. Rawlings, and M. Diehl, “CasADi – A software framework for nonlinear optimization and optimal control,” *Mathematical Programming Computation*, vol. 11, no. 1, pp. 1–36, 2019.

[58] H. Ferreau, C. Kirches, A. Potschka, H. Bock, and M. Diehl, “qpOASES: A parametric active-set algorithm for quadratic programming,” *Mathematical Programming Computation*, vol. 6, no. 4, pp. 327–363, 2014.

[59] N. Koenig and A. Howard, “Design and use paradigms for Gazebo, an open-source multi-robot simulator,” in *2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS) (IEEE Cat. No.04CH37566)*. IEEE. [Online]. Available: https://doi.org/10.1109/iros.2004.1389727

[60] T. Salzmann, E. Kaufmann, J. Arrizabalaga, M. Pavone, D. Scaramuzza, and M. Ryll, “Real-time neural MPC: Deep learning model predictive control for quadrotors and agile robotic platforms,” *IEEE Robotics and Automation Letters*, vol. 8, no. 4, pp. 2397–2404, Apr. 2023. [Online]. Available: https://doi.org/10.1109/lra.2023.3246839

Appendix A

Statistics test

A.1 Disturbance and Non-disturbance based data comparison

This section presents the results from the two independent samples. The first set of sample contains the error and thrust datasets from the MRAV in big room i.e. not experiencing the proximity effects. The second set of sample contains the error and thrust datasets from the MRAV in small room i.e. experiencing the proximity effects. These two samples are then analysed to check certain hypothesis and to conclude if there are significant difference in two samples due to the disturbances.

Table A.1: Hypothesis Test Summary

States and Errors	Axis	Sig.[a]	Decision on null hypothesis
Posititon error	x	.000	Reject
	y	.000	Reject
	z	.000	Reject
Orientation error	x	.000	Reject
	y	.000	Reject
	z	.018	Reject
Linear velocity error	x	.000	Reject
	y	.000	Reject
	z	.000	Reject
Angular velocity error	x	$<$.001	Reject
	y	.062	Retain
	z	$<$.001	Reject
Thrust	Propeller 1	.000	Reject
	Propeller 2	.000	Reject
	Propeller 3	.000	Reject
	Propeller 4	.000	Reject
	Propeller 5	.000	Reject
	Propeller 6	.000	Reject

The test that is performed is Independent-Samples Mann-Whitney U test. The Mann-Whitney U test is non parametric test to compare the distributions of independent groups and to determine if there exists a statistically difference between distributions of two samples across two categories. Further, Mann-Whitney U test can differences in shape and central ten-

dency of the distribution. The hypothesis also called null hypothesis is that the error and state datasets have same distribution for both cases i.e. when disturbances are present and when the disturbances are not present. The table A.1 shows the summary of results from the analysis. The p-value shown by Sig. in table A.1 to determine if there exists statistical differences between sample. If p-value is less than chosen significance level then the null hypothesis is rejected. The significance level is chosen to be 0.50 and represented by a in table A.1. The analysis is conducted using IBM statistics software.
It can be concluded that from the table A.1 that there exist significant differences in the two datasets except for the case in angular velocity error data in y-axis. For the angular velocity data in y axis, there is lack of evidence to reject the null hypothesis.

A.1.1 Disturbances

Table A.2: Hypothesis Test Summary

States and Errors	Axis	Sig.[a,b]	Decision on null hypothesis
Posititon error	x	.000	Reject
	y	.000	Reject
	z	.000	Reject
Orientation error	x	.000	Reject
	y	$<$.001	Reject
	z	.247	Retain
Linear velocity error	x	$<$.001	Reject
	y	.000	Reject
	z	.495	Retain
Angular velocity error	x	.021	Reject
	y	.707	Retain
	z	.006	Reject
Thrust	Propeller 1	.000	Reject
	Propeller 2	.003	Reject
	Propeller 3	.000	Reject
	Propeller 4	.000	Reject
	Propeller 5	$<$.001	Reject
	Propeller 6	.000	Reject

This section presents the results from the two related samples i.e. the experiment are repeated twice with same parameter and controller in small room i.e. experiencing the proximity effects. Both the set of samples contain the error and thrust datasets from the MRAV in small room. These two samples are then analysed to check certain hypothesis and to conclude if there are significant difference in two samples due to the disturbances. The test performed is Friedman's two-way Analysis of Variance by ranks to check if there exist statically significant difference between distribution of errors and thrusts between two samples. The analysis is conducted using IBM statistics software. The summary of result are shown in table A.2. It can be concluded that from the table A.2 that there exist significant differences in the two datasets except for the case in the orientation error in z, the linear velocity error in z, and the angular velocity error in y data. For the orientation error in z, the linear velocity error in z, and the angular velocity error in y there is lack of evidence to reject the null hy-

pothesis. It can be concluded that the error is erratic and can influence system differently as there is small but significant difference between two datasets.

A.2 Boxplot data for NMPC and GP-NMPC

The table A.3 shows the median, mean, and range of data for various errors and states of the MRAV with each controller respectively. One thing that can be concluded from table A.3 is that every dataset distribution is skewed irrespective of the controller.

Table A.3: Data from Box plots

States and errors	Axis	Median			Mean			Range		
		NMPC	Min	Ext	NMPC	Min	Ext	NMPC	Min	Ext
Position error	x	0.0318	-0.0016e-03	0.0176	0.0320	0.0332e-03	0.0145	0.0074	0.0205	0.0429
	y	0.0228	-0.5312e-03	0.0030	0.0229	0.4244e-03	0.0128	0.0083	0.0131	0.0854
	z	-0.0886	-0.2236e-03	-0.0788	-0.0881	-0.2790e-03	-0.1353	0.0133	0.0105	0.8331
Orientation error	x	0.0010	0.6762e-03	0.0604e-03	0.7342e-03	0.0951e-03	0.2005e-03	0.0192	0.0202	0.0165
	y	0.0003	-0.7696e-03	0.3752e-03	-0.0993e-03	-0.4581e-03	0.2266e-03	0.0252	0.0364	0.0211
	z	0.0000	-0.1674e-03	0.0365e-03	0.1549e-03	-0.1693e-03	-0.0239e-03	0.0084	0.0213	0.0078
Linear velocity error	x	-0.0003	-0.0007	0.0003	-0.1900e-03	-0.0012	-0.0016	0.0602	0.1808	0.2503
	y	0.0002	-0.0014	-0.0001	-0.0448e-03	-0.0003	-0.0024	0.0705	0.1202	0.4498
	z	0.0013	0.0017	0.0117	0.0209e-03	0.0019	0.1558	0.0545	0.0879	1.1856
Angular velocity error	x	0.0032	0.0009	-0.4661e-03	0.8573e-03	-0.0002e-03	0.0003	0.2044	0.1807	0.1902
	y	0.0021	0.0031	0.9123e-03	0.0444e-03	0.5155e-03	-0.0016	0.2761	0.4115	0.2115
	z	-0.0005	-0.0007	0.6109e-03	-0.1126e-03	-0.3691e-03	0.0001	0.0850	0.2697	0.1001
Thrust	Propeller 1	4.8482	4.8759	4.8947	4.8616	4.8806	4.8978	1.8158	2.9313	7.5920
	Propeller 2	4.8872	4.9189	4.9103	4.8943	4.8728	4.9049	1.7951	3.6611	7.0678
	Propeller 3	2.2962	4.9795	3.3082	2.3085	4.9028	3.5600	0.6437	3.9507	7.0265
	Propeller 4	4.8324	4.8756	4.8935	4.8547	4.8829	4.8871	1.6707	3.0262	7.6833
	Propeller 5	4.8901	4.8979	4.9349	4.8858	4.8583	4.9018	1.8982	3.6847	6.9624
	Propeller 6	2.2868	4.9683	3.2447	2.2925	4.9139	3.5569	0.5817	3.9437	7.1073

A.3 Boxplot data for nominal NMPC, GP-NMPC and DO-NMPC

Table A.4 shows the median, mean, and range of data for various errors and states of the MRAV with nominal NMPC, minimized model model, and DO respectively. One thing that can be concluded from table A.4 is that every dataset distribution is skewed for DO.

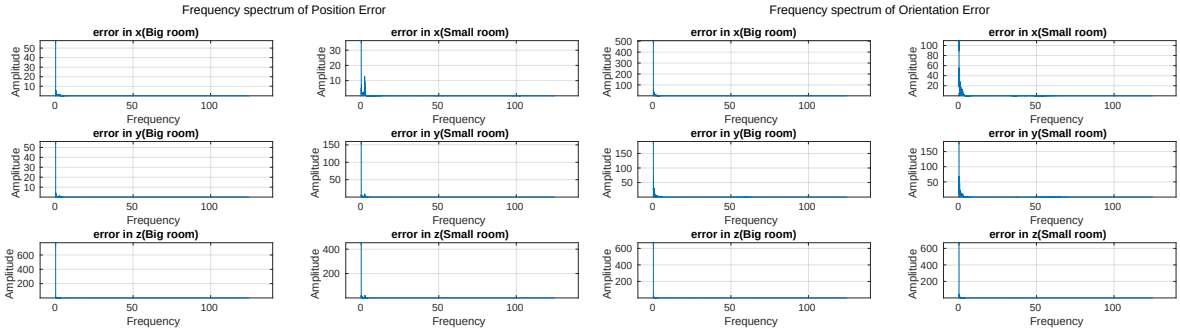
Table A.4: Data from Box plots

States and errors	Axis	Median			Mean			Range		
		NMPC	Min	DO	NMPC	Min	DO	NMPC	Min	DO
Position error	x	0.0318	-0.0016e-03	-0.0016	0.0320	0.0332e-03	-0.0039	0.0074	0.0205	0.1018
	y	0.0228	-0.5312e-03	0.0003	0.0229	0.4244e-03	-0.0004	0.0083	0.0131	0.0771
	z	-0.0886	-0.2236e-03	-0.0501	-0.0881	-0.2790e-03	-0.1294	0.0133	0.0105	0.7867
Orientation error	x	0.0010	0.6762e-03	0.2337e-03	0.7342e-03	0.0951e-03	0.0200e-03	0.0192	0.0202	0.0207
	y	0.0003	-0.7696e-03	-0.1404e-03	-0.0993e-03	-0.4581e-03	0.1470e-03	0.0252	0.0364	0.0328
	z	0.0000	-0.1674e-03	0.2826e-03	0.1549e-03	-0.1693e-03	0.2719e-03	0.0080	0.0213	0.0101
Linear velocity error	x	-0.0003	-0.0007	0.0074	-0.1900e-03	-0.0012	-0.0023	0.0602	0.1808	0.4141
	y	0.0002	-0.0014	0.0020	-0.0448e-03	-0.0003	-0.0017	0.0705	0.1202	0.3231
	z	0.0013	0.0017	0.1218	0.0209e-03	0.0019	0.1829	0.0545	0.0879	1.8755
Angular velocity error	x	0.0032	0.0009	-0.0022	0.8573e-03	-0.0002e-03	-0.5624e-03	0.2044	0.1807	0.1827
	y	0.0021	0.0031	-0.0002	0.0444e-03	0.5155e-03	-0.2750e-03	0.2761	0.4115	0.2848
	z	-0.0005	-0.0007	0.0008	-0.1126e-03	-0.3691e-03	-0.3269e-03	0.0850	0.2697	0.1013
Thrust	Propeller 1	4.8482	4.8759	4.8010	4.8616	4.8806	4.9089	1.8158	2.9313	6.9517
	Propeller 2	4.8872	4.9189	4.7394	4.8943	4.8728	4.8870	1.7951	3.6611	6.5073
	Propeller 3	2.2962	4.9795	2.5223	2.3085	4.9028	2.8882	0.6437	3.9507	9.0744
	Propeller 4	4.8324	4.8756	4.7938	4.8547	4.8829	4.8996	1.6707	3.0262	7.1153
	Propeller 5	4.8901	4.8979	4.6839	4.8858	4.8583	4.8772	1.8982	3.6847	6.2787
	Propeller 6	2.2868	4.9683	2.5281	2.2925	4.9139	2.9137	0.5817	3.9437	9.1434

Appendix B

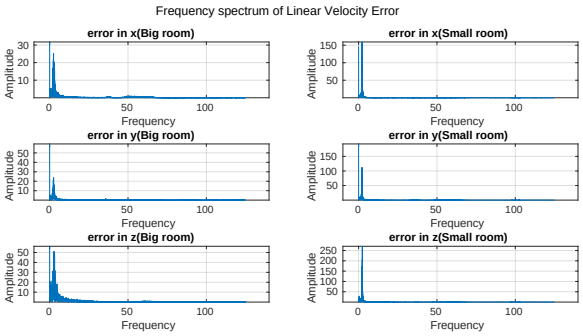
FFT of experimental data

This section shows the fft plots of error and thrust for the data from experimentation. The results from the plots have been summarized in the table 3.2.

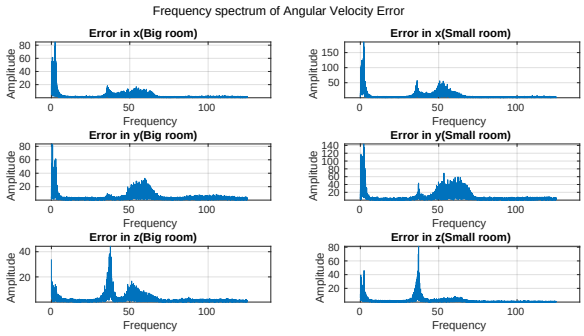


(a) FFT of Position Error

(b) FFT of Orientation Error

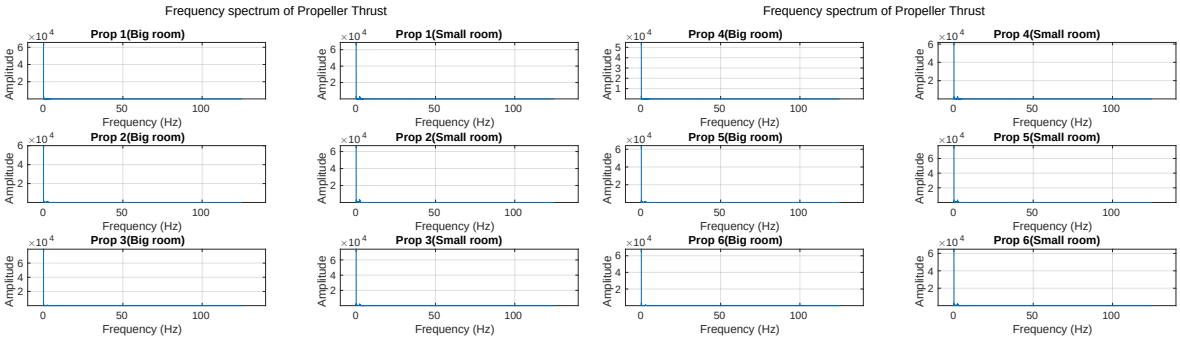


(c) FFT of Linear Velocity Error



(d) FFT of Angular Velocity Error

Figure B.1: Frequency plot for Errors



(a) FFT of Thrusts

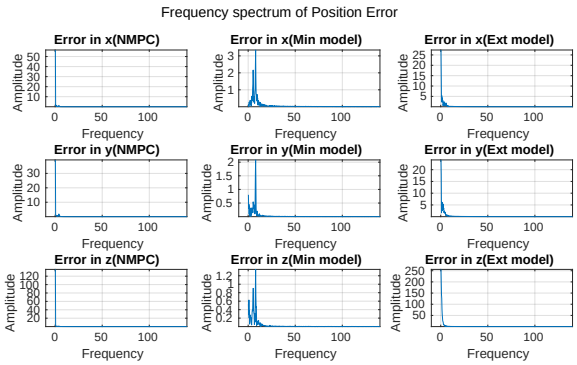
(b) FFT of Thrusts

Figure B.2: Frequency plot for Thrusts

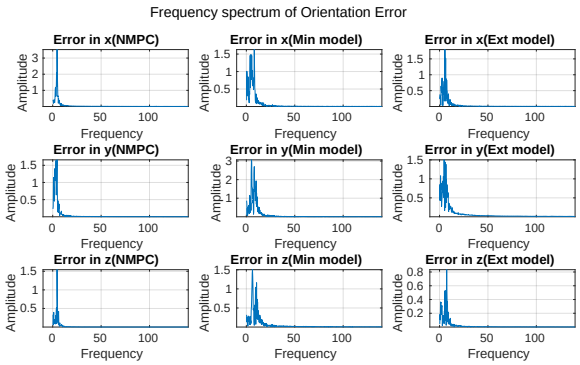
Appendix C

FFT of results

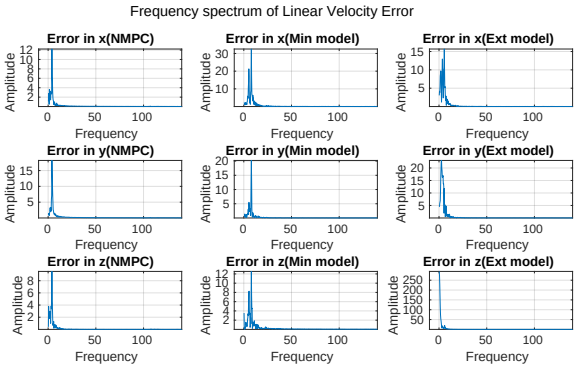
This section shows the fft plots of error and thrust for the data from results and compare the results of NMPC and GP based model. The results from the plots have been summarized in the table 6.1.



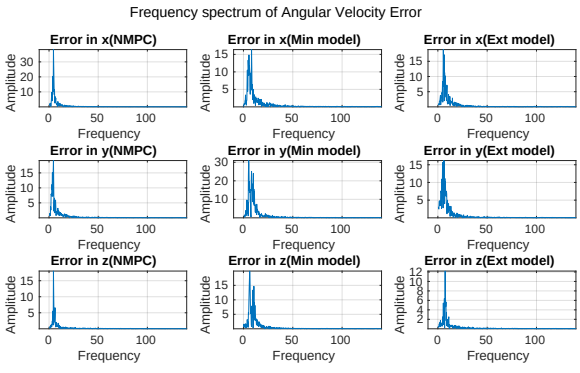
(a) FFT of Position Error



(b) FFT of Orientation Error

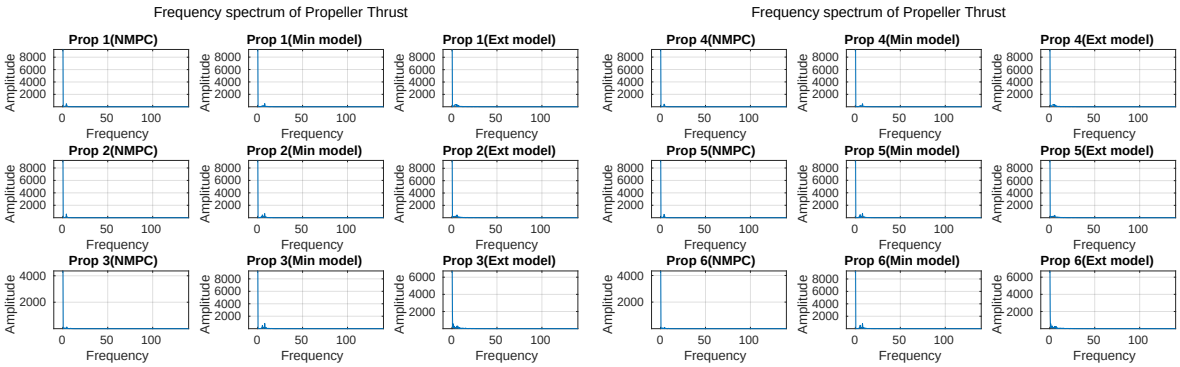


(c) FFT of Linear Velocity Error



(d) FFT of Angular Velocity Error

Figure C.1: Frequency plot for Errors



(a) FFT of Thrusts

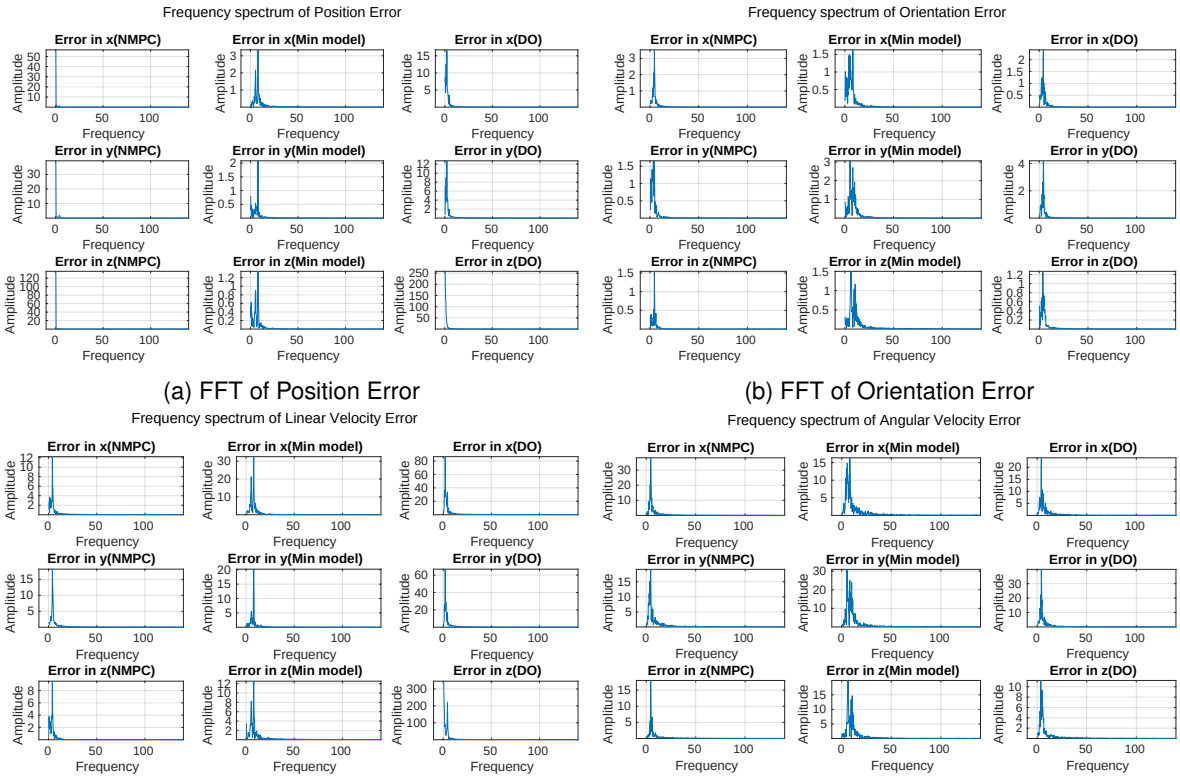
(b) FFT of Thrusts

Figure C.2: Frequency plot for Thrusts

Appendix D

FFT of results(DO)

This section shows the fft plots of error and thrust for the data from results and compare the results of NMPC and minimized model with Disturbance observer. The results from the plots have been summarized in the table 6.2.



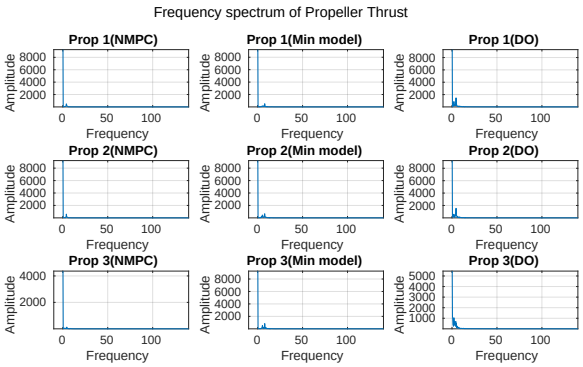
(a) FFT of Position Error

(b) FFT of Orientation Error

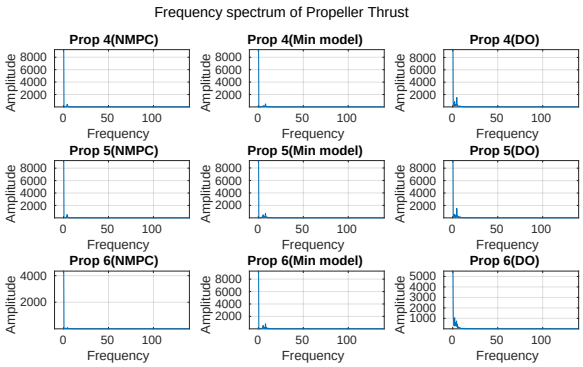
(c) FFT of Linear Velocity Error

(d) FFT of Angular Velocity Error

Figure D.1: Frequency plot for Errors



(a) FFT of Thrusts



(b) FFT of Thrusts

Figure D.2: Frequency plot for Thrusts