Fault Tolerant Control of a Quadrotor Helicopter

by

Yarkın Hocaoğlu

Submitted to
the Graduate School of Engineering and Natural Sciences
in partial fulfillment of
the requirements for the degree of
Master of Science

Sabanci University

July, 2021

Fault Tolerant Control of a Quadrotor Helicopter

Yarkın Hocaoğlu

APPROVED BY

DATE OF APPROVAL: 06/07/2021

Fault Tolerant Control of a Quadrotor Helicopter

Yarkın Hocaoğlu

ME Master's Thesis 2021

Thesis Advisor: Prof. Dr. Mustafa Ünel

Keywords: Unmanned Aerial Vehicles, Quadrotor, Fault Tolerant Control, Passive FTC, Two-Stage Kalman Filter, Extended Kalman Filter, Disturbance Observer

Abstract

Application areas of Unmanned Aerial Vehicles (UAVs) are widened significantly over the last decade. Off-the-shelf components such as low-cost sensors and actuators have broaden their usability. UAVs can be used in various missions from logistics to surveillance where ongoing research keeps encouraging new developments. Substituting a human operator with an on-board computer proposes a very appealing solution to improve the operation productivity and the cost. However, this replacement raises some safety concerns and it might be harder for a flight computer to recover from hazardous situations. Especially, UAVs that operate over crowded areas and high safety demanding environments introduce new constraints on their design process. Fault Tolerant Controllers (FTC) serve to reduce this safety gap by modeling and recovering faults during a mission. Fault recovery is a highly sought-after research topic where it is aimed to increase robustness and immunity to possible fault scenarios.

This thesis deals with developing a fault tolerant controller for a quadrotor helicopter. A high-fidelity nonlinear model of a quadrotor is constructed using Newton-Euler formulation where Dryden wind effects and sensor noise are included to simulate real-world flight conditions. A hierarchical control algorithm is employed for outer and inner control loops where PID-LQG controllers are designed to control position and attitude dynamics. For full state feedback, first a linear Two-Stage Kalman Filter (TSKF) is implemented to detect and estimate the faults and provide state estimates. Second, an Extended Kalman Filter (EKF) is used to provide more accurate state estimates. In order to increase robustness to external disturbances and uncertainties in the plant dynamics, a disturbance observer is designed and integrated to the control system. Simulations carried out with the high fidelity model have shown that the proposed fault tolerant control algorithms successfully detect and compensate for actuator and/or sensor failures in a trajectory tracking task, and hence provide good tracking performance with reasonable control effort.

Dört Pervaneli Helikopterin Hata Toleranslı Kontrolü

Yarkın Hocaoğlu

ME Master Tezi, 2021

Tez Danışmanı: Prof. Dr. Mustafa Ünel

Anahtar Kelimeler: İnsansız Hava Araçları, Dört Pervaneli Helikopter, Hata Toleranslı Kontrol, Pasif FTC, İki Kademeli Kalman Filtresi, Genişletilmiş Kalman Filtresi, Kargaşa Gözlemcisi

Özet

İnsansız hava araçlarının uygulama alanları son on yılda epey artmıştır. Rafta hazır, düşük maliyetli sensör ve eyleciler bu araçların kullanım imkanlarını genişletmiştir. Bu cihazlar lojistikten gözetlemeye kadar çeşitli görevlerde kullanılabilir ve bu durum araştırmada yeni gelişmelere öncelik etmektedir. Bir insan operatörü yerine gömülü bir bilgisayar üzerinden çalışma fikri, verim ve maliyet açısından önem arz etmektedir. Ancak tamemen bilgisayar tabanlı bir sistemde, güvenlik açısından başka kaygılar öne çıkabilir. Özellikle, kalabalık ortamlar gibi yüksek güvenlik önlemi gerektiren koşullarda yeni kısıtlamalar olması kaçınılmazdır. Hata toleranslı kontrolcü tasarımı, bu güvenlik açıklarına hizmet etmek amacıyla tasarlanır ve olabilecek hata senaryolarını modelleyip kurtararak görevi güvenilir kılmak için çalışır. Hata kurtarma çalışmaları, çok rağbette olan bir çalışma konusu olup sistemin gürbüzlüğünü arttırmaya ve hatalara karşı bağışıklı olmayı amaçlar.

Bu tezin konusu, dört pervaneli bir helikopter için hata toleranslı kontolcü geliştirme üzerinedir. Yüksek sadakatli doğrusal olmayan bir model Newton-Euler formülasyonuyla geliştirilmiştir ve gerçek uçuş senaryolarını yansıtması adına Dryden rüzgar efekti ve sensör gürültüsü de eklenmiştir. İç ve dış kontrol döngülerini içeren, hiyerarşik bir kontrol algoritması geliştirilmekle birlikte; PID-LQG kontolcüsü pozisyon ve oryantasyon kontrolü için kullanılmıştır. Bütün sistem parametrelerini geri beslemek adına, önce bir lineer İki Kademeli Kalman Filtresi kullanılmıştır. Bu filtre hem hata hem de sistem parametrelerini tahmin etmek için kullanılmıştır. İkinci olarak, bir Genişletilmiş Kalman Filtresi sistem parametre tahminini iyileştirmek için kullanılmıştır. Sistemi dış bozucu ve belirsizliklere karşı gürbüz kılmak adına kontrol sistemi üzerine bir kargaşa gözlemcisi entegre edilmiştir. Simülasyondan alınan sonuçlara istinaden, tasarlanan hata toleranslı kontrolcünün eyleyici ve sensör hatalarını bulup, yörünge takip senaryosında başarılı bir şekilde ve mantıklı kontrol eforu ile telafi edebildiği görülmüştür.

Acknowledgements

Foremost, I would like to express my deep and sincere gratitude to my thesis advisor Prof. Dr. Mustafa Ünel for his continuous support and his marvellous supervision. It was a great privilege and honor to work and study under his guidance. I am so grateful for his immense patience to me on this three years of journey where he kept his support all the time which is the main reason for concluding this work. I would like to thank him for his exceptional vision as an engineer where he taught me to show immense focus on every detail where nothing works by chance. I have also highly benefited from his remarkable multi-disciplinary background as a researcher where I have found great eagerness to discover. His passion and dedication to his profession has always mesmerized me and he stands as a role model for me to shape my career.

I would also like to thank Assoc. Prof. Dr. Kemalettin Erbatur and Assist. Prof. Dr. Hüseyin Üvet for spending their valuable time to review my thesis.

I am very grateful to my colleague Mehmet Emin Mumcuoğlu where his enormous help guided me to surpass bottle-necks that I have encountered highly. I greatly appreciate his passion to share and discuss during this thesis period with his humble approach.

I would also like to thank my former employer Turkish Aerospace Inc. for allowing me to carry through my Master's studies.

Last but not least, I would like to thank my family for all the unconditional love and support they have given me throughout my life. I would like to thank my parents Gül and Mahir Hocaoğlu for helping me to be grateful for life and appreciate every little beauty of it. I am very thankful to them for inspiring me to become a good person that has right ethical values and giving me power to strive independent of its consequence. I am also very grateful to my love İlayda Yelken for all the discussions we had to look hopefully to future together and her continuous support with every possible way.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Unmanned Aerial Vehicles' popularity is increasing rapidly as their application areas are widened through time. Due to enhanced availability of low-cost sensors and platforms, civil field has also benefited highly from UAVs regarding research and applications[1]. One of the reasons that enabled this circumstance is the increased trust on these devices due to ever-evolving advancements.

UAVs are employed for a variety of tasks such as photography, military defense, precision agriculture, logistics and etc.[2]. Different types of UAVs can be found on operation with respect to employed tasks and work loads. These differences basically can be classified according to structural properties which include rotary-wing, fixed-wing and hybrid configurations. Rotary-wings use multiple(3,4,6,8) rotors for actuation, fixed-wings use generated lift from its airfoils and hybrid configuration combines both properties in the case of a tilt-wing or tilt-rotor UAV. Separately from their structural properties; UAVs can be piloted by remote-control or autonomously and can be utilized as single or in multi-agent manner[3].

Although there is an active research environment for various UAV-based topics, many incremental innovations bring new challenges that are introduced with increased complexity and safety requirements. Especially, considering immensely growing market opportunities for logistic delivery UAVs where fast shipments are made possible. This inevitably increases market share of these devices and their operation intensity accordingly. Therefore, managing safe missions is and will be playing a very important role for preventing possible hazardous situations.

So, their safety under operation, concerns human safety factor if it crashes. This is a very crucial topic especially under UAV guidance because of the possibility they might operate under crowded environments. Therefore, it is possible to observe a tendency to operate fully autonomously for eliminating human feedback for control. This brings another challenge because even though ground robots are able to localize themselves and can create a map of their environment for some time, UAVs suffer from localizing themselves [4]. It is much harder to explore and map an unknown space. Enabling an autonomy to a UAV would cancel human feedback out of control loop. A related and very active field of research is improving the feedback quality for the UAV. For example, UAVs that use optical feedback for position estimation is a very active field of research in terms of control and obstacle avoidance. There are a great number of algorithms such as RANSAC [5] that try to optimize visual guidance for better pose estimation. This wide variety of research that combines various engineering disciplines leads to expand application areas every day. Few contemporary applications can be given as example such as;

- Structural inspections where buildings[6] or various infrastructures like road[7] investigation
- Transportation where landslide monitoring[8] is used to take precautions on the road and traffic monitoring[9]
- Historic preservation which monitors sensitive and cultural monuments[10]
- Progress monitoring where tracking materials[11] and fatigue inspection under unsafe worksites
- Search and rescue missions[12]

One other benefit for preventing any hazardous situations is disuse of expensive components accordingly which are valid apprehensions. However, techniques such as fault tolerant control serves to decrease this safety gap by modeling and controlling possible fault scenarios that may harm an aerial mission. For this reason, developing fault compensation mechanisms play a very important role to consolidate the role of UAVs in the future.

Fault-tolerant control systems become very handy if conventional feedback does not satisfy the system requirements and shows poor performance which is a possible case under a fault scenario. However, we can compensate the feedback signal with a fault tolerant controller in order to make the system perform better with these unexpected situations. Only solution however is not a controller design, one can make a system fault tolerant via hardware redundancy but this would increase cost and weight which is not very desirable. A FTC overview can be categorized in two sectors where fault diagnosis is one category that focuses on modeling or finding the fault which can be considered as a health monitoring system. Second category is fault recovery where solutions to compensate a detected fault are proposed. There are plenty of configurations for each field of research that collectively serve fault tolerant systems. This work combines UAV dynamics with fault scenarios that will be compensated by various controller and observer architectures.

1.1 Motivation

Fault tolerant control strategies constitute an active field of research in terms of employing an Unmanned Aerial Vehicle as a test bed. Combining a highly intricate platform such as UAV with possible recovery solutions under a high disturbance environment constitutes a challenging proposition. Idea of fault compensation is spread to a vast majority of engineering disciplines where safety critical systems are prioritized such as nuclear facilities and aircraft flight controls[13]. Even though these safety critical applications remain important, UAVs have many advantages over conventional aircraft for research purposes due to mechanical simplicity, ability to take-off and land vertically(VTOL) and proportionally small sizes with respect to a conventional aircraft[14].

Aerial vehicles also provide great benefits due to their wide operation environments. Their ability to perform with hard-to-reach operations puts them one move ahead over human personnel. High safety-demanding workloads such as fire rescue to military applications make UAVs capable tools.

Specialized missions can be achieved via blending this ease of applicability of a UAV and tolerance to various types of faults. Depending on the type of mission whether its a trajectory tracking or payload delivery target; robustness to any fault would be highly beneficial. Therefore, designing a control architecture which is robust to faults pose a challenge in terms of integrity of the model. Mentioned control architecture does not have to consider every possible scenario that may prevent mission completion but may also adapt to differentiating model characteristics with respect to faults, disturbances and noise. While faults basically can be classified into 3 major branches as actuator, sensor and structural; many other configurations within these branches may effect UAV dynamics. Therefore, apart from the control perspective; modeling each and every fault with accuracy in order to counter-act is also a major challenge within this area of research.

One of the highly worked-over test beds for experimentation are quadrotors which itself resents from its under-actuated architecture. Controlling 4 rotors to move in 6 degree-of-freedom while having sufficient bandwith to control all rotors simultaneously is a valid challenge still, depending on the workload and task[15]. Due to having a challenging platform with coupled high non-linearity and fault scenarios requests very precise and high-fidelity model in which a robust controller can act upon accordingly.

1.2 Contributions of the Thesis

The contributions of this thesis are outlined as follows:

- A high-fidelity non-linear model is constructed using Newton-Euler formulation for quadrotor type UAV.
  - No linearization is required for plant dynamics.

  - Developed model includes Dryden wind effects and measurement noise in order to simulate real-world scenarios.

- Two hierarchical controllers are designed where
  - A PID-PD architecture is employed for outer and inner controllers.
  - A PID-LQG architecture is employed for outer and inner controllers, respectively.
- Fault models have been integrated to the model which are comprised of;
  - Actuator faults
  - Sensor faults
  - Structural failures
- Various observers have been implemented for achieving accurate state and disturbance estimations with and without fault injection.
  - A Two-Stage-Kalman Filter which employs a linear model for state estimation and estimates faults independently.
  - An Extended Kalman Filter which provides more accurate state estimates due to its nonlinear prediction capability.
  - A Disturbance Observer which estimates a lumped disturbance to counter-balance actual disturbance on the plant.

1.3 Outline of the Thesis

This thesis consists of 8 chapters where;

Chapter 2 gives background information and literature survey on UAVs, fault modeling and detection, various controller architectures and possible aerial missions that are formed under these topics.

Chapter 3 gives a detailed model development using Newton-Euler formulation.

Chapter 4 is about modeling faults and articulates possible scenarios related to these fault types.

Chapter 5 accounts for controller types and reasons behind their selection. This section gives details about employed hierarchical control architecture where outer control loop is responsible for position dynamics and inner control loop is responsible for attitude dynamics.

Chapter 6 provides details about observer dynamics and mathematical background of all the observers that are used for state and fault estimation.

Chapter 7 presents simulation results for various scenarios where different faults have been considered and appropriate estimators, controllers and observers have been implemented.

Chapter 8 concludes the thesis with several remarks and indicates possible future directions.

Chapter 2

Background and Literature Survey

2.1 Unmmanned Aerial Vehicles

Unmanned Aerial Vehicle is a mobile robot that operates midair which can be controlled either remotely by a human operator or fully autonomously. These devices benefit from various sensors that helps it to navigate consciously in a controlled manner. It is possible to categorize UAVs in 3 categories which are based on weight, altitude and range, wings and rotors. Following list provides sub-types for each category of UAVs which is inspired from [16].

Classification with respect to weight

- Nano UAVs which weigh less than 250 grams
- Micro UAVs which weigh higher than 250 grams and less than 2 kilograms
- Small UAVs which weigh higher than 2 kilograms and less than 25 kilograms
- Medium UAVs which weigh higher than 25 kilograms and less than 150 kilograms
- Large UAVs which weigh higher than 150 kilograms

(a) Black Hornet Nano UAV[17] (b) DJI AGRAS Small UAV[18] (c) TAI Anka Large UAV[19]

Figure 2.1: UAVs with different sizes

Classification with respect to altitude and range

- Hand-held UAVs that flies altitude under 600 meters and has range lower than 2 kilometers
- Close UAVs that flies altitude under 1500 meters and has range lower than 10 kilometers
- NATO UAVs that flies altitude under 3000 meters and has range lower than 50 kilometers
- Tactical UAVs that flies altitude under 5500 meters and has range lower than 160 kilometers
- MALE (Medium Altitude Low Endurance) UAVs that flies altitude overr 9100 meters and has ranger higher than 200 kilometers
- Hypersonic UAVs that flies altitude around 15000 meters and has range higher than 200 kilometers

(a) Optimus Close UAV[20] (b) JOUAV NATO UAV[21] (c) D-21 Hypersonic UAV[22]

Figure 2.2: UAVs with different altitude and range

Classification with respect to actuation

- Fixed-Wing
- Single Rotor
- Multi Rotor
- Fixed-Wing Hybrid VTOL

(a) Insitu Fixed-Wing UAV[23] (b)Single Rotor UAV[24] (c) SUAVI Tilt-Wing UAV[25]

Figure 2.3: UAVs with different actuation types

Even though fault tolerant control studies on UAVs have huge research interest for the last decade, simulation-based experiments outweigh real-world test-beds. However, it is possible to find various test-beds that accompany different platforms such as rotary-wing and fixed-wing applications. Following sections will present a background information on UAV platforms that are test-beds under fault tolerant control, fault modeling and diagnosis, controller architectures within FTC scheme, respectively.

2.1.1 Rotary-Wing UAV

One of the most commonly used test-beds under UAV research activities are quadrotors for their relatively low price and ease of controllability by driving its actuators with Pulse Width Modulation (PWM) signals. There are a lot of different quadrotor brands which can be off-the-shelf or custom-built depending on use-cases. However, as fault tolerant flight is concerned; Quanser's Qball-X4 is a very actively used platform by Concordia University[26]. This test-bed has a carbon fibre cage which protects the drone from possible expenses and ensures its agility with preserving a low payload.

Figure 2.4: Qball-X4 test-bed used by Concordia University[26]

Rotary wing platforms include single rotor type helicopter UAVs where Georgia Tech's GTMax[27] platform is used as an experimental test bed which includes fault tolerant control studies. Single rotor helicopters are known to have high instability by not having an instant lift-to-power property due to lack of control surfaces to accommodate airfoil. Therefore, this platform can be a good test-bed for experimenting high variety of faults which even small ones might cause catastrophic consequences.

Figure 2.5: Georgia Tech GTMax Platform[27]

GTMax platform has a 2 cylinder internal combustion engine that powers its single rotor. Fault injection on this platform is conservative due to its expensive price therefore a typical fault scenario includes limiting the swash plate angle for introducing limited actuator command. Developed fault tolerant control module is able to cope with this type of fault as additive control effort is employed[27].

2.1.2 Fixed-Wing UAV

Fixed-wing UAV platforms have high impact factor on influencing FTC research because of its possible benefits on civil aviation. Fixed-wing UAVs have high stability over their control surfaces which opens a large envelope for applicable FTC techniques. However, due to disadvantages such as need of high speed take-off and landing, runway requirements and being always open to disturbances such as cross wind and gust[28]; puts this platform into a challenging situation.

Figure 2.6: 60% Wing Loss on F-18 Subscale UAV[29]

One of the used platforms for real-world testing experiments is a sub-scale of F-18 turbojet UAV whose damage tolerant controller is developed and tested by Rockwell Collins aerospace company. In this experiment, up to 60% wing loss has been implied on the UAV which is recovered by its on-board adaptive controller[29].

2.2 Fault Detection and Identification

Fault detection and identification is a very broad field of research due to its applicability on various engineering disciplines. Therefore, elaborating on its background has an importance on understanding its effects.

2.2.1 Description of Fault

It is possible to see an effort in literature on describing what is considered to be a fault and why it is important to make a distinction between a fault and a failure. A fault is defined to be a condition where an unallowed malfunction occurs within the system where this malfunction does not directly effect system dynamics[30]. On the other hand, a failure effects directly the functioning of a component on the system. It would be possible to say that multiple faults may occur on the system but the system may still be functioning via compensating a fault. However, if one or multiple faults prevent system dynamics to operate in a controlled fashion, this would be classified as a failure. A good example to make this distinction can be the relation between a sensor fault and an actuator failure. A sensor may read inaccurate readings due to sensor bias or calibration error where this wrong reading behaviour might not harm a component until some time is passed.

However, if there is an actuator failure such as its loss of effectiveness is reduced per se, instant effects can be observed due to component failure[30]. Some type of sensor faults can be seen in Figure 2.7[31].

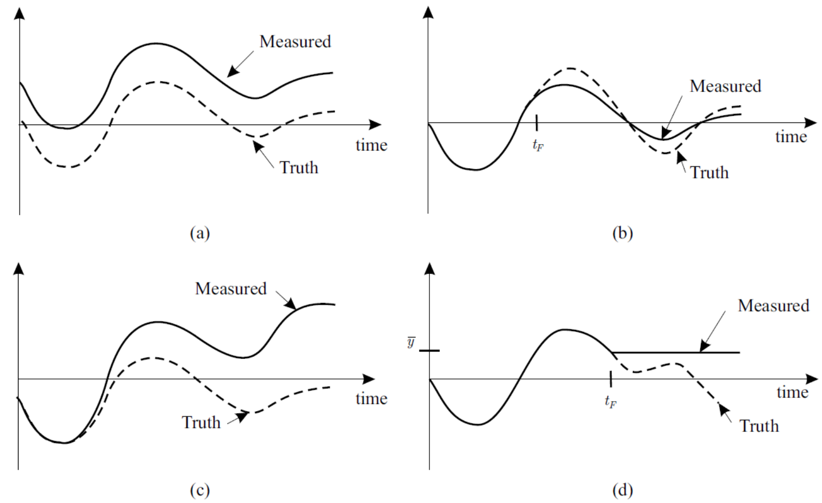

Figure 2.7: Type of sensor faults: (a) sensor bias; (b) loss of accuracy of calibration error; (c) sensor drift; (d) frozen sensor[31]

2.2.2 Detecting a Fault

In order to comprehend and tolerate any given fault, one can suggest to model each fault explicitly within system dynamics. This is a valid approach for consciously improving the system however it is not always the case. In FTC scheme, fault does not always have to be modelled and isolated from the system. Even though there are plenty of methods related to fault modeling, fault can be compensated by a robust controller architecture without the controller having any clue about any fault. In order to detect a fault, one can use the aforementioned fault model to counter-act. However, this is not the only option where fault detection methods can be categorized in two major branches which are model-based and model-free where model-based's name is self explanatory and model-free systems use data driven approaches and statistical proofing[32]. Data driven approaches are highly benefited from neural networks[33], SVM algorithms[34], reinforcement learning[35] and statistical methods become handy in terms of simplicity and ease of applicability[36]. Nevertheless, model-free methods need huge bank of data and computation effort can be costly therefore, one of the most encountered fault detection algorithm is model-based approach where an observer architecture is employed mostly. Some of the highly used techniques for model-based fault detection are Thau observers[37], sliding mode switching observers[38], high gain observers[39] and different variants of Kalman filters[31] such as a Two-Stage Kalman filter[40].

2.3 Fault Tolerant Controllers

Fault tolerant control (FTC) is an active field of research with improving control architectures to back-up system failures. Unmanned Aerial Vehicles are one of the highly benefited areas of work due to high non-

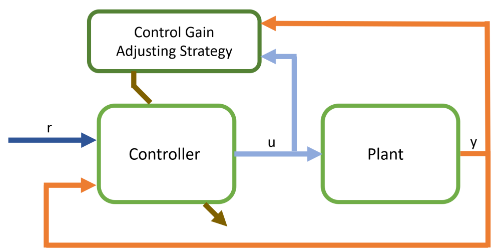

Figure 2.8: Active FTC

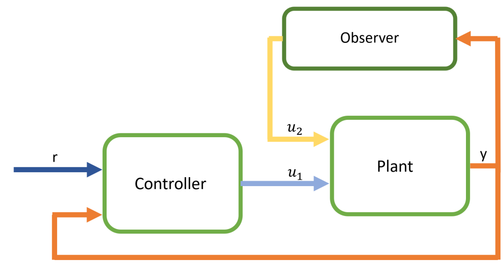

Figure 2.9: Passive FTC

linearity of aircrafts and high safety requirements. Therefore, finding a fault and isolating that from system dynamics via a controller plays a very important role for its robustness. So, designing a controller is as important as defining a fault accurately due to high disturbances and symmetric structures of most UAVs which augments the importance of fault identification techniques to estimate and verify how much a fault is qualified to be considered. This section presents a literature review on fault tolerant control schemes where fault diagnosis and reconfiguring a corresponding control architecture will be presented.

Fault tolerant control systems can be classified in two sections namely passive and active FTC schemes. Active type uses a strategy to modify control gains with respect to estimated fault or uncertainty during simulation. On the other hand, passive systems use one controller architecture but additive control input can be used for estimated faults and uncertainty.
Active techniques can either be active for using a separate fault detection and identification mechanism, or can be active regarding to controller which adapts to faults without having the knowledge of fault itself[41]. Even though we have branched fault detection techniques as model-based and model-free earlier, how it is detected during the simulation without emphasizing on the model stands as a category under fault tolerant control ideology. This section focuses on articulating the controller part due to scope of this work where comparisons will be laid accordingly. However interested reader can go through references [42]-[47] where detailed information about Thau observers as an active fault identification method is discussed extensively. Also, in this thesis we have used a Two-Stage Kalman Filter where active fault identification is achieved via simultaneously estimating system states and faults[26].

2.3.1 Active FTC

Unlike a traditional controller, active fault tolerant systems introduce more state parameters into the state vectors where these parameters are actively controlled depending on the architecture of the plant. As in the FTC case, estimated faults are included in the controller decision. It is possible to come across a high variety of active fault tolerant controllers in literature. One of the highly used architectures is adaptive feedback linearization where nonlinear state space dynamics are modeled[48]. Adaptive sliding mode[43] and adaptive sliding mode backstepping[44] are also useful controllers for their adaptivity and high robustness envelope. Model Reference Adaptive Control (MRAC) is another highly used application where controller parameters are updated with respect to changing system dynamics and faults in order to increase robustness to parametric uncertainties[49].These parametric uncertainties that are related to un-modelled dynamics can be actively estimated and used in control allocation where bounded uncertainties can be represented[50]. Rather than altering control law for controller as in the case of MRAC, gain scheduling[51] is also highly employed for dynamically changing control gains with fast response.

It is possible to come across hybrid control techniques such as [52] where a passivity-based adaptive backstepping technique is used to deal with mass uncertainties on the quadrotor due to unknown payload. Generally for the adaptive approach, parameter uncertainties are targeted to be eliminated and passive techniques such as backstepping[46] is good for overcoming under-actuation problem[53].
Rather than focusing solely on controller, some applications include altering the trajectory such that if a fault is introduced, an optimized easier to follow trajectory can be reconfigured for compensation[54].

2.3.2 Passive FTC

Passive FTC lays its foundations on robust control in which it defends having a single controller architecture compliant enough to cope with parametric uncertainties. Passive fault tolerant controllers have a fixed stability margin where faults, disturbances and noise can be compensated within that region. This controller type is especially beneficial for faster response however depending on disturbances, it may need high robustness envelope which can be hard to accomplish. LQR control[55] is an optimal control technique which has high robustness envelope and it is used extensively for UAVs. Backstepping[56] is also an effective way in terms of its adaptation capability without knowing a fault in the system.

Some works in literature use sliding mode control (SMC)[57][45] due to its high robustness margin. However due to its switching behaviour of SMC, discontinuities appear during SMC regulation which induces chattering. Even though SMC is known for providing asymptotic stability, unmatched disturbances such as fault introduction may be a problem due to its discontinuous nature[58]. However, SMC architecture is one of the highly worked-over controller configurations where different techniques such as terminal sliding mode control (TSMC) [59] are experimented to overcome this chattering problem. Normally linearly defined tracking error on SMC is defined in a non-linear manner in this work where error is estimated apriori for its compensation on the controller. By this way, controller can act faster and reduce chattering problem.

Passive FTC method is also highly used with simpler to implement controllers such as PID. It can show exceptional results such as in this work [60] where it uses a PD controller and a robust compensation mechanism to deal with uncertainties. Authors have managed to get successful results with a passive PD controller on a real experimental test-bed.

It should be noted that artificial intelligence based techniques are also applied on FTC which serves an advantage since not requiring a model to deal with uncertainties. This is defined under passive FTC because AI is mostly used for uncertainty estimation which backs up a passive controller. One example [61] can be usage of a radial basis function network (RBFNN) which can be employed as an uncertainty observer that uses prior flight data obtained from the system. Estimated uncertainty is lumped back to the controller to compensate the uncertainty in the system just like in the case of a disturbance observer. This work employs a backstepping controller which is unaware of neither a fault nor any uncertainty so it is a passive system however, used RBFNN estimates uncertainties online therefore fault detection is conveyed actively. These are hybrid systems where combining adaptation and robustness techniques aim to operate back to back for fault compensation. This aims for serving best of both worlds where robustness of controller is increased with adaptive fault estimation. There are also right opposite structures where highly adaptive controller's robustness is increased with a robust observer [62] to have better attitude dynamics.

Chapter 3

Nonlinear Model for Quadrotor

A nonlinear mathematical model of a quadrotor is presented in this section. This model will later employ various choice of controllers within the simulation that can be switched manually by the operator. This system takes reference trajectories as input and gives position, velocity and acceleration as output. A general view of modelled dynamics of quadrotor can be seen in Figure 3.1

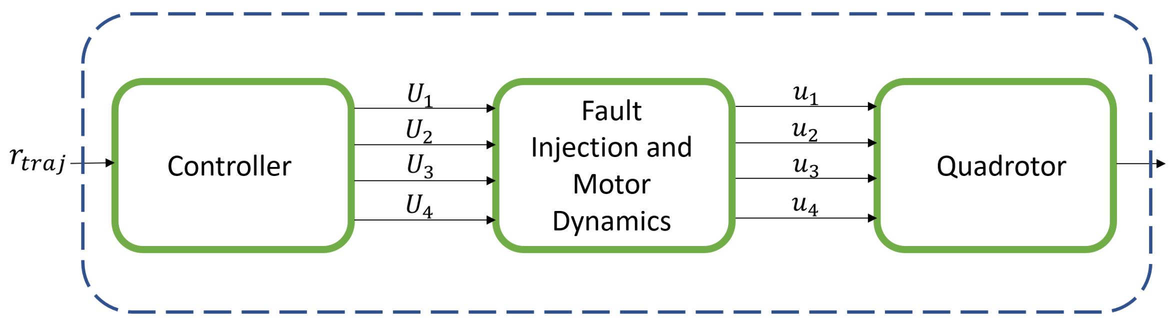

Figure 3.1: Quadrotor System Block Diagram

Fault injection and motor dynamics block contains motor dynamics where it takes force and torque control inputs and converts them to individual PWM motor inputs. Every motor is modelled with a linear first order transfer function which takes PWM input u_i and motor model $K\frac{w}{s+w}$ where K is the positive motor gain and w is the motor bandwith. Combining motor dynamics with PWM input of each rotor gives motor thrusts;

$$T_i = K\frac{w}{s+w}u_i \text{ ; } i = 1, 2, 3, 4 \tag{3.1}$$

For mapping between control inputs to motor PWM control inputs, we have used the following matrix;

$$\begin{bmatrix} U_1 \\ U_2 \\ U_3 \\ U_4 \end{bmatrix} = \begin{bmatrix} K & K & K & K \\ 0 & 0 & KL & -KL \\ KL & -KL & 0 & 0 \\ KK_\psi & KK_\psi & -KK_\psi & -KK_\psi \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \tag{3.2}$$

where L is the distance from the center of mass to each motor. K_ψ is a constant for yawing motion. U_1 is the total lifting force acting on the system, U_2, U_3, U_4 are torques for rolling, pitching and yawing motions respectively. For the output of PWM signals, we have also saturated the signals in between 0 and 0.05 where 0 denotes no power and 0.05 denotes maximum actuator effort. Rotor directions and corresponding movements can be visualized in Figure 3.2.

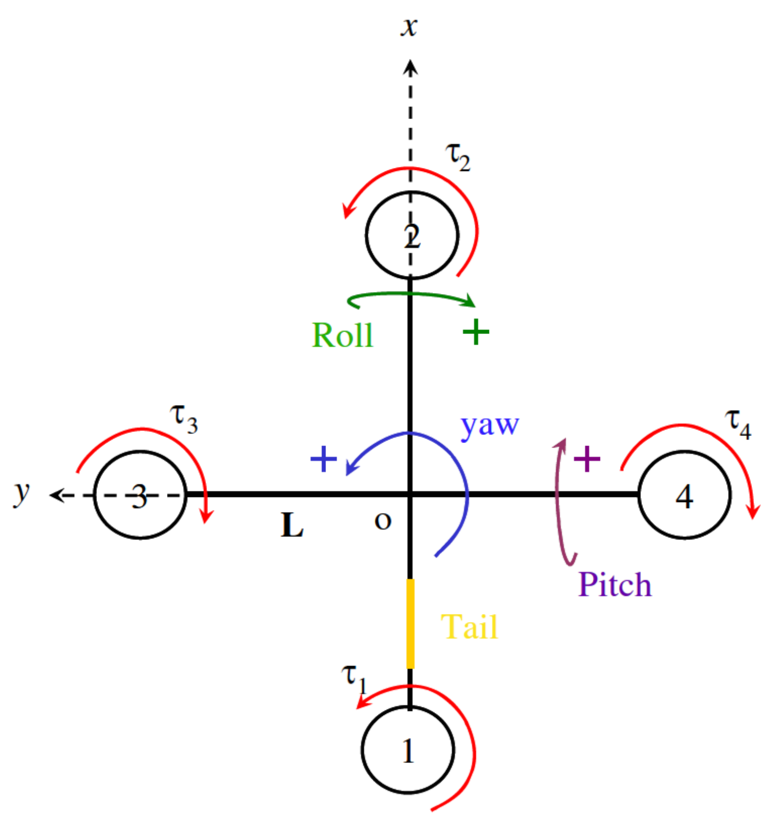

Figure 3.2: Schematic Representation of Quadrotor [26]

Newton-Euler formulation is employed in order to obtain dynamic models for the quadrotor. It is assumed that air frame is rigid body where it benefits from same dynamic force and moment equations. Dynamic equations are calculated with respect to body frame where it is denoted as subscript b.

$$m(w_b \times v_b + \dot{v_b}) = F \tag{3.3}$$

$$w_b \times (I_b w_b) + I_b \dot{w}_b = M \tag{3.4}$$

where m is the mass, $w_b = [p\ q\ r]^T$ is the angular velocity in body frame, $v_b = [u\ v\ w]^T$ is the linear velocity in body frame and F is the total force. I_b is the moment of inertia matrix in body frame and M is the total moment. It is possible to manipulate equations (3.3) and (3.4) to show linear and angular velocities in body frame at left hand side.

$$\begin{bmatrix} \dot{u} \\ \dot{v} \\ \dot{w} \end{bmatrix} = \frac{F}{m} - (w_b \times v_b) \tag{3.5}$$

$$\begin{bmatrix} \dot{p} \\ \dot{q} \\ \dot{r} \end{bmatrix} = {I_b}^{-1}(M - (w_b \times I_b w_b)) \tag{3.6}$$

Mathematical model of the quadrotor is derived from its force and moment balance equations given in (3.3) and (3.4) where they mostly rely on propeller thrust and gravity as force components and moment created under roll, pitch and yaw due to orientation imbalance and opposite propeller speeds. Vector $[x\ y\ z\ \phi\ \theta\ \psi]$ denotes position and orientation of the UAV in earth frame and vector $[u\ v\ w\ p\ q\ r]$ denotes linear and angular velocities in body frame.

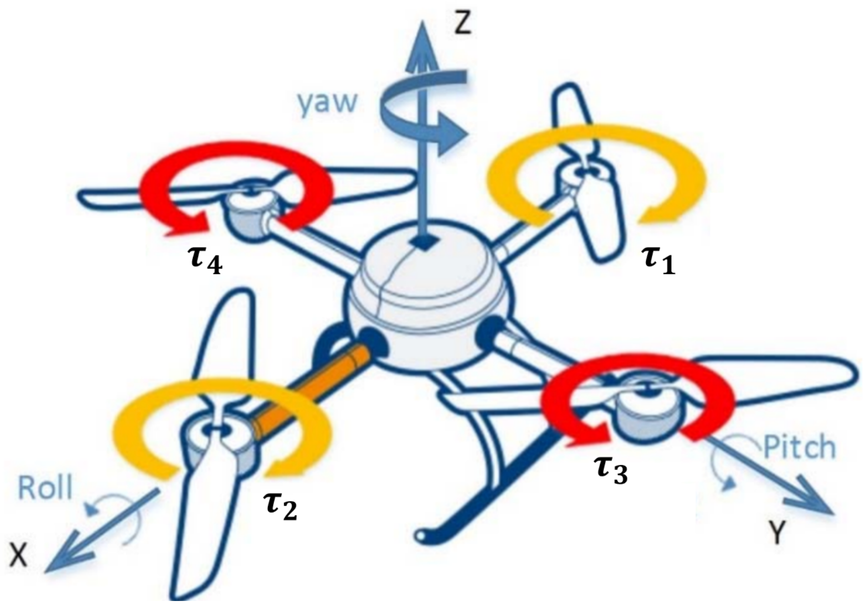

Figure 3.3: Body Coordinate System for a Quadrotor

Z-axis' direction is taken pointing upwards as can be seen in Figure 3.3 where gravitational forces are denoted negative and propeller thrust as negative. Body and earth frame can be linked by velocity transformation and rotation matrices such as;

$$v = Rv_b \tag{3.7}$$

$$w = Tw_b \tag{3.8}$$

where $v = [\dot{x}\ \dot{y}\ \dot{z}]^T$, $w = [\dot{\phi}\ \dot{\theta}\ \dot{\psi}]^T$, $v_b = [u\ v\ w]^T$ and $w_b = [p\ q\ r]^T$. R matrix enables transformation from body frame to inertial frame which is formulated by multiplying three rotation matrices at ZYX conversion;

$$R = \begin{bmatrix} c_\psi c_\theta & -s_\psi c_\phi + c_\psi s_\theta s_\phi & s_\psi s_\phi + c_\psi s_\theta s_\psi c_\phi \\ s_\psi c_\theta & c_\psi c_\phi + s_\psi s_\theta s_\phi & -c_\psi s_\phi + s_\psi s_\theta s_\psi c_\phi \\ -s_\theta & c_\theta s_\phi & c_\theta c_\phi \end{bmatrix} \tag{3.9}$$

T matrix is the rotation matrix that maps angular velocities from body frame to inertial frame.

$$T = \begin{bmatrix} 1 & s_\phi t_\theta & c_\phi t_\theta \\ 0 & c_\phi & -s_\phi \\ 0 & \frac{s_\phi}{c_\theta} & \frac{c_\phi}{c_\theta} \end{bmatrix} \tag{3.10}$$

Kinematic model becomes;

$$
\begin{aligned}
\dot{x} &= w[s(\phi)s(\psi) + c(\phi)c(\psi)s(\theta)] - v[c(\phi)s(\psi) - c(\psi)s(\phi)s(\theta)] + u[c(\psi)c(\theta)] \\
\dot{y} &= v[c(\phi)c(\psi) + s(\phi)s(\psi)s(\theta)] - w[s(\phi)c(\psi) - s(\psi)c(\phi)s(\theta)] + u[s(\psi)c(\theta)] \\
\dot{z} &= w[c(\phi)c(\theta) + c(\phi)c(\psi)s(\theta)] + v[c(\theta)s(\phi)] - u[s(\theta)] \\
\dot{\phi} &= p + r[c(\phi)t(\theta)] + q[s(\phi)t(\theta)] \\
\dot{\theta} &= q[c(\phi)] - r[s(\phi) \\
\dot{\psi} &= r\frac{c(\phi)}{c(\theta)} + q\frac{s(\phi)}{c(\theta)}
\end{aligned}
\tag{3.11}
$$

where s denotes sin, c denotes cos and t denotes tan. Next, total force acting on the system will be found by Newton's law;

$$
\begin{aligned}
f_x &= m(\dot{u} + qw - rv) \\
f_y &= m(\dot{v} - pw - ru) \\
f_z &= m(\dot{w} - pv - qu)
\end{aligned}
\tag{3.12}
$$

We have showed total force generated as U_1 such as $U_1 = f_x + f_y + f_z$. Total moment acting on the system can be found as;

$$
\begin{aligned}
m_x &= \dot{p}I_{xx} - qrI_{yy} + qrI_{zz} \\
m_y &= \dot{q}I_{yy} - prI_{xx} - prI_{zz} \\
m_z &= \dot{r}I_{zz} - pqI_{xx} + pqI_{yy}
\end{aligned}
\tag{3.13}
$$

We define forces and moments in body frame as $f_b = [f_x \ f_y \ f_z]$ and $m_b = [m_x \ m_y \ m_z]$ where gravity will be combined as well. But first gravity vector should be multiplied with R^T to body frame representation;

$$
f_b = mgR^T e_z - f_t e_3 + f_w \tag{3.14}
$$

where e_z is the unit vector in inertial z axis, e_3 is the unit vector in body z axis, f_b denotes external forces in body frame, f_t is the total thrust generated by rotors which can be expressed as $T_i = K\frac{w}{s+w}u_i$, K is thrust coefficient and f_w is the force caused by the wind. Similar formulation is again conveyed to calculate external moment balance.

$$
m_b = \tau_B - g_a + \tau_w \tag{3.15}
$$

where $\tau_B = [U_2 \ U_3 \ U_4]^T$ is control torques generated by motor speed difference, g_a is generated gyroscopic moments and τ_w is torques generated by wind. Generally, g_a is neglected for quadrotors due to very small inertia value for each rotor. If it would have been a tilt wing application per se, then rotor inertia should have been considered as well. Combining external force and moment equations gives;

$$\begin{aligned}
-mg[s(\theta)] + f_{wx} &= \dot{u} + qw - rv \\
mg[c(\theta)s(\phi)] + f_{wy} &= m(\dot{v} - pw - ru) \\
mg[c(\theta)c(\phi)] + f_{wz} - f_t &= m(\dot{w} - pv - qu) \\
U_2 + \tau_{wx} &= \dot{p}I_{xx} - qrI_{yy} + qrI_{zz} \\
U_3 + \tau_{wy} &= \dot{q}I_{yy} + prI_{xx} - prI_{zz} \\
U_4 + \tau_{wz} &= \dot{r}I_{zz} + pqI_{xx} - pqI_{yy}
\end{aligned} \tag{3.16}$$

Actuator dynamics can be modelled by using generated PWM thrusts in (3.1) where control inputs can then be found via (3.17).

$$\begin{aligned}
U_1 &= \sum_{i=1}^{4} T_i \\
U_2 &= LT_3 - LT_4 \\
U_3 &= LT_1 - LT_2 \\
U_4 &= K_\psi T_1 + K_\psi T_2 - K_\psi T_3 - K_\psi T_4
\end{aligned} \tag{3.17}$$

where L is the distance between the center of the quadrotor and the center of the rotor and K_ψ is the yawing constant. Combining all the previous equations provide us the position in earth frame and orientation in body frame which constitutes a hybrid frame for the quadrotor is as follows;

$$\begin{aligned}
\ddot{X} &= (s(\phi)s(\psi) + c(\psi)s(\theta)c(\phi))\frac{U_1}{m} \\
\ddot{Y} &= (-c(\psi)s(\phi) + s(\psi)s(\theta)c(\phi))\frac{U_1}{m} \\
\ddot{Z} &= (c(\phi)c(\theta))\frac{U_1}{m} - g \\
\dot{p} &= \frac{I_{yy} - I_{zz}}{I_{xx}}qr + \frac{U_2 + \tau_{wx}}{I_{xx}} \\
\dot{q} &= \frac{I_{zz} - I_{x}}{I_{yy}}pr + \frac{U_3 + \tau_{wy}}{I_{yy}} \\
\dot{r} &= \frac{I_{xx} - I_{y}}{I_{zz}}pq + \frac{U_4 + \tau_{wz}}{I_{zz}}
\end{aligned} \tag{3.18}$$

Chapter 4

Definition and Classification of Faults

Fault detection and isolation process can be utilized under phenomenon of hardware or software redundancy. Hardware redundancy employs duplicate of several hardware where measurement signal can be supplied from two independent sensors for example which are allocated for the same task. Therefore, if a fault occurs for a specific sensor, a correlated separate sensor would still send correct data for evaluation of that signal. Same redundancy can be applied to actuators or even computers to recover the system from a faulty condition likewise. Hardware redundancy is utilized frequently for aerospace applications where safety plays a high importance under certification purposes. However as one could expect, relying more on hardware introduces extra cost, increased weight and occupies more space which can not be desirable for plenty of low cost or space-restricted applications.

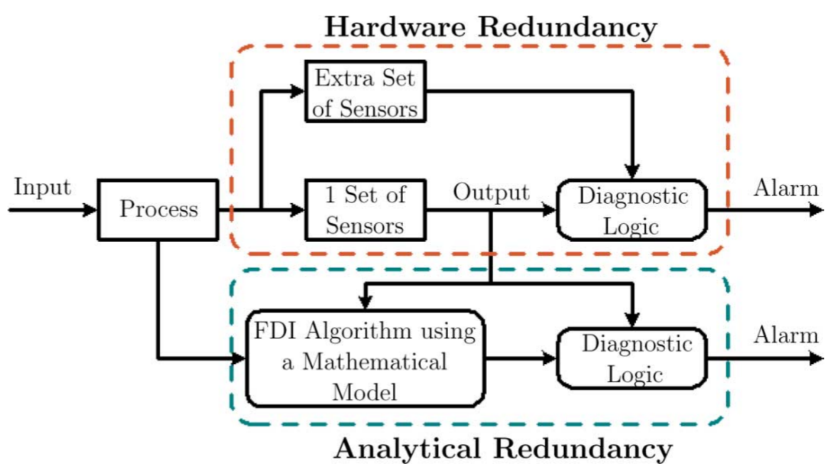

Figure 4.1: Hardware and analytical redundancy

On the other hand, software (analytical) redundancy employs a mathematical model of the system where it constantly checks deviations of measured output from estimated output via various estimation techniques which will be discussed further in this section.

This work focuses on analytical redundancy where fault modeling will be investigated for further isolate that fault out of the system and reconfigure the controller to take an action accordingly. The outline of this chapter will be as follows;

- System and Fault Modeling
- Residual Generation
- Fault Isolation
- Decision Making
- Reconfiguration
- Fault Injection Interface

Chapter 5 will be dedicated to controller design in which once an accurate fault model is obtained; various fault tolerant controller architectures will be discussed.

4.1 System and Fault Modeling

We can add disturbance terms to represent the plant dynamics of a system where observer dynamics are introduced via assuming a linear system. Leariazed system model becomes;

$$\begin{aligned} x(t+1) &= (A + \Delta A)x(t) + (B + \Delta B)u(t) + E_1 n_1(t) \\ y(t) &= (C + \Delta C)x(t) + (D + \Delta D)u(t) + E_2 n_2(t) \end{aligned} \tag{4.1}$$

where n_1 and n_2 are unknown disturbance vectors; ΔA and ΔB correspond to model uncertainties.
Fault vector is generally modelled including 3 states which are sensor, actuator and component faults. Actuator and sensor faults are modelled as 'additive fault' where a stuck elevator or faulty sensor reading can be examples. On the other hand component fault is modelled as 'multiplicative fault' where a permanent damage on fuselage or broken control surface. If we include fault terms to our system model;

$$\begin{aligned} x(t+1) &= (A + \Delta A + \Delta A_c)x(t) + (B + \Delta B + \Delta B_c)u(t) + E_1 n_1(t) + B f_a(t) \\ y(t) &= (C + \Delta C + \Delta C_c)x(t) + (D + \Delta D + \Delta D_c)u(t) + E_2 n_2(t) + f_s(t) \end{aligned} \tag{4.2}$$

where $f_a(t)$ is actuator faults, $f_s(t)$ is sensor faults; ΔA_c and ΔB_c represent component faults. As one can see from (4.2), a possible component fault disturbs the system dynamics highly and unwanted effects such as sensor noise, environmental disturbances and component faults are evaluated within the same model uncertainty matrix. This is not a desired calculation method because we would like to isolate disturbances specific to faults in order to reconfigure an updated control strategy. This is where a residual signal is introduced which is defined as the difference between measured output and estimated output. Goal for designing an effective residual is to minimize effects of disturbances and sensor noise to emphasize strictly on the fault itself which is a challenging objective. The ideology behind creating a 'pure' residual remains an active field of research. As in the scope of this work, we can embed the model uncertainties into the state matrices and take out multiplicative component fault to be added to a generalized additive fault vector. This notion of an isolated fault(f_t) can be modelled as;

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) + E_1(t)n_1(t) + Bf_a(t) + F_1(t)f_c(t) \\ y(t) &= Cx(t) + Du(t) + E_2(t)n_2(t) + f_s(t) + F_2(t)f_c(t) \end{aligned} \tag{4.3}$$

where $n = [n_1 \ n_2]^T$ is the noise vector, $f = [f_a \ f_s \ f_c]^T$ is the fault vector. If we squeeze (4.3) into an input-output scheme and take the z-transform of the equation;

$$y(z) = G(z)u + F(z)f + E(z)n \tag{4.4}$$

where $G(z), F(z), E(z)$ are transfer functions for how effective the control, fault or noise vectors on output. They are defined as;

$$G(z) = C(zI - A)^{-1}B + D$$
$$F(z) = [(zI - A)^{-1}E_1E_2]$$
$$E(z) = [(zI - A)^{-1}BI(zI - A)^{-1}F_1 + F_2]$$

4.2 Residual Generation

As stated earlier, residual signal is defined as the difference between measured output and estimated output which can be shown as;

$$r(t) = y(t) - \hat{y}(t) \tag{4.5}$$

If there is no fault in the system, the mean of the residual $E[r(t)]$ is zero. If there is a detected fault, then mean should diverge from zero.
There are various methods that tries to create the most robust residual that is independent of noise and disturbances. Some of them are;

4.2.1 Full-state observer-based methods

If we would write a simplified version of the existing fault model;

$$\begin{aligned} x(t+1) &= Ax(t) + Bu(t) + E_1n_1(t) + Bf_a(t) \\ y(t) &= Cx(t) \end{aligned} \tag{4.6}$$

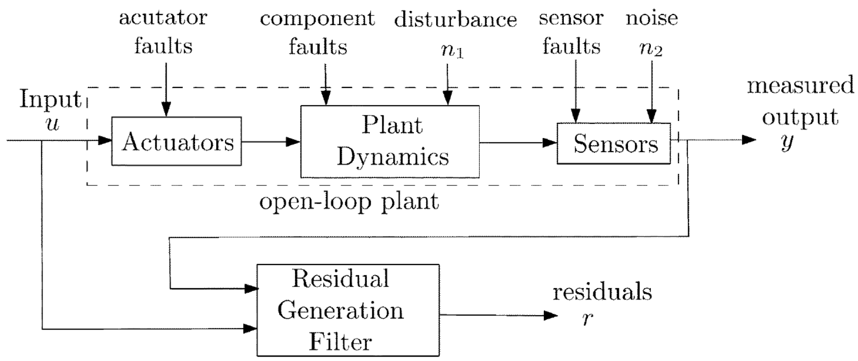

Figure 4.2: Residual generation

A full state observer would give an estimate such as;

$$\begin{aligned} \hat{x}(t+1) &= A\hat{x}(t) + Bu(t) + L(y(t) - \hat{y}(t)) \\ \hat{y}(t) &= C\hat{x}(t) \end{aligned} \tag{4.7}$$

where L is the typical observer gain. Residual should also be updated due to altered estimate;

$$r(t) = W(y(t) - \hat{y}(t)) = WC\xi(t) \tag{4.8}$$

where W is the weighting matrix and ξ is state estimation error $x - \hat{x}$. Next, error dynamics can be written as;

$$\xi(t+1) = (A - LC)\xi(t) + E_1 n_1(t) + Bf_a(t) \tag{4.9}$$

By using pole placement, one can choose such observer gain L that may lead the system to asymptotic stability. Also eigenstructure assignment can be employed other than pole placement where transfer function of noise in residual signal can be nulled via choosing such W and L.

4.2.2 Kalman filter-based approach

A Kalman filter based observer identify faults regarding to the whiteness, mean and covariance of residuals. Generally, such application is assembled to the system by assigning separate Kalman filters to every possible fault scenario. By this way, a bank of Kalman filters are employed as in Figure 4.3.

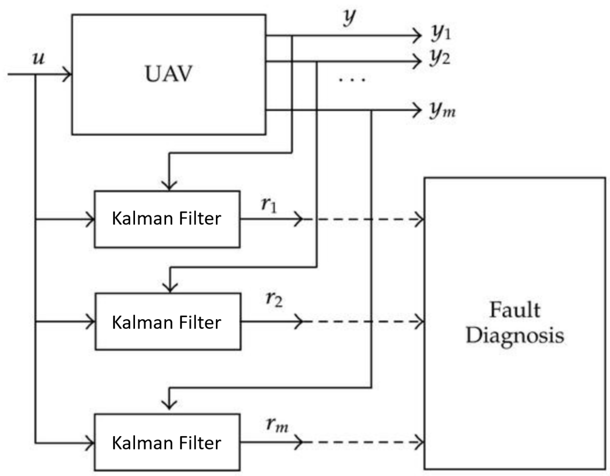

Figure 4.3: Parallel Kalman filters assigned to separate faults

This technique is very widely used with aerospace applications where Multiple Model Adaptive Estimation (MMAE) under reconfiguration chapter is a good example. With MMAE, each type of fault for parallel Kalman outputs are recorded and a specific model is assigned to each fault configuration. If the system detects a fault under a predefined condition, entire system model will switch to that faulty scenario model where again a predefined controller architecture will be assigned accordingly.

Actuator Fault Modeling

In this work Kalman filter-based approach will be employed for modeling actuator faults. An adaptive two stage Kalman filter(ATSKF) is used for this purpose where an augmented state matrix which embeds system states and actuator faults is configured. Augmenting the state vector enables us to make MIMO predictions separating states and faults. However, this method will be only used for estimating actuator faults with respect to loss of effectiveness factor where a γ parameter is constituted to represent how effective that actuator is. Loss of control effectiveness is reasoned to partial losses in either hydraulic or pneumatic pressures for control surfaces in fixed wing aircrafts or inadequate thrust for quadrotors due to limited motor bandwith. One can show the direct effect of partial actuator loss as;

$$u_i^f = (1 - \gamma_i)u_i \tag{4.10}$$

where u_i^f is the faulty control input, γ_i is the LOE factor which is bounded between $0 \leq \gamma_i \leq 1$ and u_i is the apriori control input before fault injection. This part mainly focuses on actuator faults due to loss of control effectiveness but three other actuator fault types are included into the fault injection block for simulation that are specific to fixed wing UAV types which are

- *Actuator lock* denotes a mechanical jam on the actuator that supplies a constant value due to its position for control input.

- *Float failure* denotes a hydraulic, pneumatic or electric failure for the actuator where a control pressure or electrical current is completely lost for that actuator. This results in a freely moving control surface where it generates no lift.

- *Hardover Actuator Failure* denotes the actuator at its maximum deflection where it supplies high lift without the need due to software or hardware faults.

Fault modeling techniques are not limited to Kalman filtering only. Even though only KF is employed for fault detection in this work, various compensation schemes are also available in literature. Some of the other techniques that are used for fault detection and residual generation are;

- Parity relations approach [63]
- Optimization-based approach [64]
- Stochastic approach [65]
- System identification methods [66]
- Nonlinear systems [67]
- Artificial intelligence approach [68],[69]

4.3 Fault Isolation

Now that is clear that an optimum residual is the one that is sensitive to faults; we need to define how that specific fault is isolated from other faults. Basically, an isolation is needed to distinguish each fault from another which is covered under two methods being directional residual approach and structured residual approach. In directional residual approach, direction of every residual is defined under a residual set where faults are classified regarding to their directional properties. In structural residual approach, every fault is quantified due to how similar it is to other faults. In other words, each fault has a weight that is assigned from similarity to other faults. By this way a certain fault characteristic can be chosen and can be isolated conforming to that fault property.

4.4 Decision Making

This part mostly rely on a statistical background where an identified fault is evaluated whether it should be taken into account for controller reconfiguration[70]. Essentially, a decision is made for a previously quantified fault to be worthy of a compensation. The most basic technique to make such decision is to define a threshold where consider every fault as valid if it exceeds that threshold. However, this is not a comprehensive method which is why various decision making techniques are employed. Some techniques use adaptive thresholds in which a probabilistic designation becomes handy. Some of the highly used examples are as follows;

- Sequential probability ratio test (SPRT) [71]
- Cumulative sum algorithm (CUSUM) [72]
- Local approach [73]

4.5 Reconfiguration

This part is about re-configuring a control action to back up the actual controller output. A control action is taken at this step in the light of quantified valid faults to be added to the real controller. A block diagram for tracing the signals can be seen in Figure 4.4.

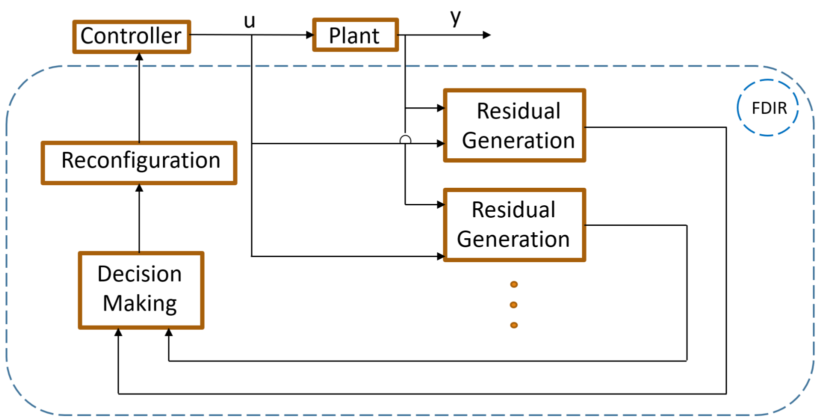

Figure 4.4: Fault detection, isolation and reconfiguration (FDIR) scheme on a system

Various controller architectures can be employed under 2 major branches which are passive and active control. Passive control method tries to find a robust control region where possible fault scenarios have already been modelled and these faults are compensated within this entire robust control envelope. This method can be very fast comparing to active control due to defined control actions however designing a wide robustness margin can make the system nonreactive under some conditions. With active FTC on the other hand, controller structure changes with respect to identified faults to improve real time fault handling characteristics. Even though active control reconfiguration sound promising, it requires high computation power. Some of the methods that are used for control reconfiguration in literature are classified as follows[74];

- Passive (Robust Control)
- Active
  - Multiple Model (Multiple Model Switching and Tuning)

– Adaptive (MRAC, Adaptive Feedback Linearization via ANN)
– Actuator Only (Sliding Mode Control, Control Allocation)
– Controller Synthesis where Fault Model is Assumed (Eigenstructure Assignmet, MPC)

4.6 Fault Injection

This part is dedicated to fault injection methods where one can change fault characteristics to his/her own preference. Depending on the origin of fault and its type, fault injection module has three branches where it lets user to set faults either one-by-one or simultaneously. Faults caused by actuators, sensors or components can be configured depending on the fault scenario.

Actuator faults are modeled in this work as LOE (Loss of Effectiveness) factors where injected step LOE value causes degradation in actuator operation. Sensor faults effect system output matrix where they are classified under 5 configurations. *Bias* represents a constant error in the output where a random noise will be generated for each click. *Loss of accuracy* is self explanatory for this sensor fault where an accuracy rating can be manually tuned in percentage. *Drift* is a cumbersome fault where the sensor output keeps diverging with respect to time. This fault can be modelled as a ramp function where the assigned slope would emulate rate of the drift. *Freezing* is again a straight forward fault where sensor reading is stuck at a specific value. This option will serve as a steady state sensor measurement. *Calibration Error* is the last sensor fault option where a sensor becomes unreliable to be read. If this fault is selected, a chosen sensor will continuously send uncorrelated set of random measurements.

One other fault type is structural failure where UAV looses a part from its body. This can be modeled as actuation loss if a fixed-wing UAV looses its rudder per se. However, if our quadrotor model looses an actuator totally like an entire motor for example; its balance would totally collapse and become very hard to control unlike rudder loss where a fixed-wing can still preserve its symmetry. So, structural failure on our model will be modeled as a battery-pack that is ruptured from the rigid body where it only effects weight not moments of inertia.

Chapter 5

Controller Design

This work lays its foundations on to Passive FTC scheme in which defends one and only robust controller can compensate effects of disturbances such as faults, failures and noise. Therefore, an attempt to compose a robust controller architecture is presented in this chapter. We have used a hierarchical control scheme due to under-actuated structure of the quadrotor.

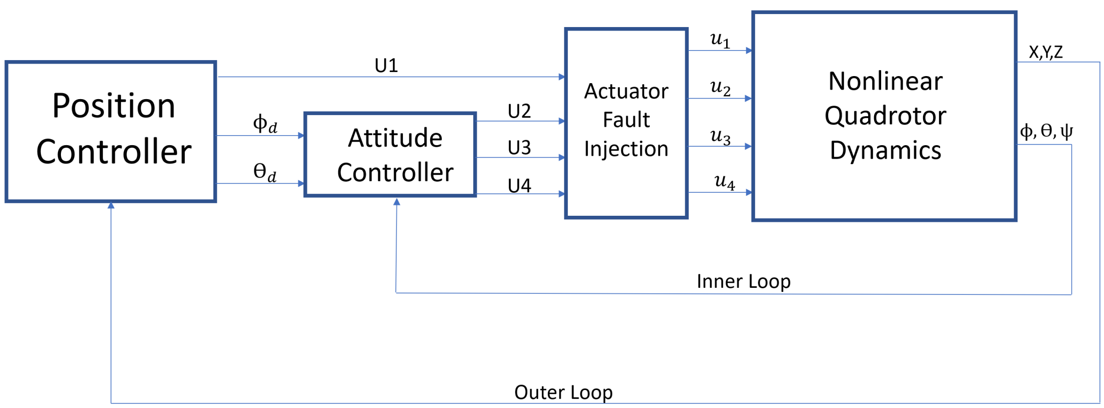

Figure 5.1: Hierarchical Control Architecture

Two controller loops are generated where a feed-forward PID architecture is employed for outer (Position) controller and two switchable controllers are embedded inside the inner (Attitude) controller. Inner controller's robustness plays a very important role for stabilizing the system therefore PD and LQR architectures are constituted for studying robustness.

5.1 Position Control

5.1.1 PID Control

Position controller is responsible from trajectory tracking in which deals to minimize error dynamics on reference states. It also provides desired pitch and roll commands for the attitude controller. Position controller employs virtual control inputs forces the errors on X,Y,Z states in order to converge to zero[75]. We can formulate these errors on position dynamics with the following error dynamics formulation;

$$\begin{aligned} e_x &= X_d - X \\ e_y &= Y_d - Y \\ e_z &= Z_d - Z \end{aligned} \tag{5.1}$$

Taking derivative of error terms yields;

$$\begin{aligned} \dot{e}_x &= \dot{X}_d - \dot{X} => \ddot{e}_x = \ddot{X}_d - \ddot{X} \\ \dot{e}_y &= \dot{Y}_d - \dot{Y} => \ddot{e}_y = \ddot{Y}_d - \ddot{Y} \\ \dot{e}_z &= \dot{Z}_d - \dot{Z} => \ddot{e}_z = \ddot{Z}_d - \ddot{Z} \end{aligned} \tag{5.2}$$

We can now define the virtual control inputs;

$$\begin{aligned} \mu_x &= \ddot{X}_d + K_{p,x} e_x + K_{d,x} \dot{e}_x + K_{i,x} \int e_x dt \\ \mu_y &= \ddot{Y}_d + K_{p,y} e_y + K_{d,y} \dot{e}_y + K_{i,y} \int e_y dt \\ \mu_z &= \ddot{Z}_d + K_{p,z} e_z + K_{d,z} \dot{e}_z + K_{i,z} \int e_z dt \end{aligned} \tag{5.3}$$

Desired feed-forward terms are obtained by taking double derivatives of desired X,Y,Z values outputted from trajectory generation block, and PID terms are feedback terms obtained from error dynamics. Next, we can determine the total thrust U_1 and the desired roll and pitch angles in terms of calculated virtual control inputs as [75];

$$\begin{aligned} U_1 &= m\sqrt{\mu_x^2 + \mu_y^2 + (\mu_z + g)^2} \\ \phi_d &= asin\left(\frac{s_{\psi_d}\mu_x - c_{\psi_d}\mu_y}{\sqrt{\mu_x^2 + \mu_y^2 + (\mu_z + g)^2}}\right) \\ \theta_d &= asin\left(\frac{c_{\psi_d}\mu_x - s_{\psi_d}\mu_y}{c_{\psi_d}\sqrt{\mu_x^2 + \mu_y^2 + (\mu_z + g)^2}}\right) \end{aligned} \tag{5.4}$$

5.2 Attitude Control

Attitude controller needs a well-designed controller architecture due to high frequency operating conditions. Therefore, maintaining robustness inside inner loop takes huge importance for overall stability of the system. In order to cope with high non-linearity that is brought by fault injection and disturbance; we have designed two switchable controllers to emphasize the importance of robust attitude control.

5.2.1 PD Control

Just as in the case of position controller, attide controller is benefited from PID control. However, control gains are higher with inner loop with respect to outer loop and even a small error can lead a control degradation which is why integral control is eliminated with attitude control.

Just as before, we can define error dynamics for attitude controller as follows;

$$\begin{aligned} \ddot{\phi} &= \ddot{\phi}_d + K_{p,\phi} e_\phi + K_{d,\phi} \dot{e}_\phi \\ \ddot{\theta} &= \ddot{\theta}_d + K_{p,\theta} e_\theta + K_{d,\theta} \dot{e}_\theta \\ \ddot{\psi} &= \ddot{\psi}_d + K_{p,\psi} e_\psi + K_{d,\psi} \dot{e}_\psi \end{aligned} \tag{5.5}$$

Now, we can calculate control inputs U_2, U_3, U_4 as follows;

$$\begin{aligned} U_2 &= I_{xx}(\ddot{\phi}_d + K_{p,\phi} e_\phi + K_{d,\phi} \dot{e}_\phi) \\ U_3 &= I_{yy}(\ddot{\theta}_d + K_{p,\theta} e_\theta + K_{d,\theta} \dot{e}_\theta) \\ U_4 &= I_{zz}(\ddot{\psi}_d + K_{p,\psi} e_\psi + K_{d,\psi} \dot{e}_\psi) \end{aligned} \tag{5.6}$$

5.2.2 LQG Control

Linear Quadratic control is an optimal control method which employs a state feedback for closed loop dynamics. Gains for state feedback controller can be chosen depending on demands of the system by minimizing a cost function[76]. LQR control places the eigenvalues on desired positions which ensures stability. In our case, we are using output of Kalman filter estimates to achieve full state feedback therefore, such applications are named as Linear Quadratic Gaussian control.

A finite horizon, linear quadratic regulator (LQR) is formulated as;

$$x_{k+1} = Ax_k + Bu_k \tag{5.7}$$

where a discrete linear state space model is given such that $x \in \Re^N$, $u \in \Re^N$, x_0 given. A cost function J is defined;

$$J = \sum_{k=1}^{\infty} x_k^T Q x_k + u_k^T R u_k \tag{5.8}$$

where $Q \geq 0$ is a positive semi-definite and $R > 0$ is a positive definite matrix. A control input is determined as;

$$u_k = r_k - K x_k \tag{5.9}$$

where K is the feedback gain matrix and r is the reference input vector. K is calculated by solving the algebraic Riccati equation. However, in MATLAB *lqr* command minimizes the cost function by placing eigenvalues automatically via given Q and R matrices such as $K = lqr(A, B, Q, R)$

Gains are chosen depending on what is required out of the system. Giving higher values for Q enables better trajectory tracking but increases control effort. Giving higher values for R reduces control effort by sacrificing less accurate trajectory tracking therefore, an optimal calibration is done via tuning them with trial and error.

Linearized system matrices at hovering condition and small angle approximation are as follows;

$$\begin{aligned} U_2 &= I_{xx}\ddot{\phi} \\ U_3 &= I_{yy}\ddot{\theta} \\ U_4 &= I_{zz}\ddot{\psi} \end{aligned} \tag{5.10}$$

which leads to obtain a linear state space model to calculate feedback gain;

$$\begin{bmatrix} \dot{\phi} \\ \ddot{\phi} \\ \dot{\theta} \\ \ddot{\theta} \\ \dot{\psi} \\ \ddot{\psi} \end{bmatrix} = \begin{bmatrix} 0\,1\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,1\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,0 \\ 0\,0\,0\,0\,0\,0\,1 \\ 0\,0\,0\,0\,0\,0\,0 \end{bmatrix} \begin{bmatrix} \phi \\ \dot{\phi} \\ \theta \\ \dot{\theta} \\ \psi \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} 0\;0\;0 \\ \frac{1}{I_{xx}}\;0\;0 \\ 0\;0\;0 \\ 0\;\frac{1}{I_{yy}}\;0 \\ 0\;0\;0 \\ 0\;0\;\frac{1}{I_{zz}} \end{bmatrix} \begin{bmatrix} U_2 \\ U_3 \\ U_4 \end{bmatrix} \tag{5.11}$$

Obtained A and B matrices are utilized along with Q and R matrices to compute the optimal state feedback gain.

LQR is also known for its disability to eliminate steady-state error even though it gives remarkable tracking results[77]. In order to compensate for this, we have added an integral action to error dynamics which eliminated previous steady-state error perfectly.

Chapter 6

Observer Design

In this section, some mathematical background is provided for observer development. We have first implemented a Two-Stage Kalman Filter (TSKF) which is designed to estimate injected faults and system states seperately. TSKF is a beneficial filter especially under FTC scheme however its driving equations are based on linearized system dynamics at a given hovering condition. Therefore, it may have some difficulties if nonlinearities are dominant.

In order to cope with nonlinearities in the system dynamics, an Extended Kalman Filter (EKF) has been implemented. Extended Kalman Filters provide more accurate state estimations due to their fully nonlinear architecture. This ability to perform estimations in a non-linear manner, removes necessity to estimate a separate fault value from residual.

These two observers are employed for improving state estimations in order to enable full state feedback condition. We have also designed another observer additively benefits system dynamics. A disturbance observer (DOB) is implemented in order to estimate a lumped disturbance value where a disturbance estimate can be added to the feedback control as a feedforward term for compensating effects of disturbance. Two relevant disturbance observer architectures are implemented which are called Acceleration-Based (AbDOB) and Velocity-Based disturbance observers (VbDOB).

6.1 State Estimation

6.1.1 Two-Stage Kalman Filter

This part is dedicated to deriving a TSKF for predicting loss of control effectiveness in an arbitrary actuator. This method is first formulated by [78] where two Kalman filters are combined for multiple predictions. First a state matrix should be formed to represent the dynamics which is why we should linearize both of our plants around a trim point. This trim point will be the hovering condition for quadrotor where there is no yaw and very small roll and pitch angles are assumed. Simplified model becomes[26];

$$\begin{aligned}
\ddot{x} &= \theta g \\
\ddot{y} &= -\phi g \\
\ddot{z} &= g - \frac{U_1}{m} \\
\ddot{\phi} &= \frac{U_2}{I_{xx}} \\
\ddot{\theta} &= \frac{U_3}{I_{yy}} \\
\ddot{\psi} &= \frac{U_4}{I_{zz}}
\end{aligned} \tag{6.1}$$

State vector is $[x, \dot{x}, y, \dot{y}, z, \dot{z}, \phi, \dot{\phi}, \theta, \dot{\theta}, \psi, \dot{\psi}]$, and $[U_1, U_2, U_3, U_4]$ are total lift force, moment of roll, moment of pitch and moment of yaw respectively. In order to stay in these trim points, quadrotor control input is taken to be $[U_1, U_2, U_3, U_4]$ where it covers all the rotors. Linearized state space matrices A, B, C and D are formulated by using Jacobian matrices.

$$\begin{aligned}
\dot{x}_p(t) &= A_p x_p(t) + B_p u_p(t) \\
y_p &= C_p x_p(t)
\end{aligned} \tag{6.2}$$

In order to include the fault model into the state space model, we can augment the states by including faulty actuator control inputs. First, state space model is designed that takes actuator dynamics into account.

$$\begin{aligned}
\dot{x}_a(t) &= A_a x_a(t) + B_a u(t) \\
y_a(t) &= C_a x_a(t)
\end{aligned} \tag{6.3}$$

where $x_a = [x_p \ u_p]^T$ is the augmented state vector, $u = [u_1, u_2, u_3, u_4]$ for quadrotor. If we consider control inputs to be modelled as PWM where a first order linear transfer function becomes

$$F_i = K \frac{w}{s+w} u_i = \bar{K} u_i \tag{6.4}$$

where K is a positive gain, w is motor bandwith and variation in thrust can be modeled accordingly. B_a becomes $B_a = [\bar{K} B_p \ K w]^T$ and $C_a = [C_p 0]$. $A_a = A_{a0} + \delta A_a$ where $A_{a0} = [A_p 0; 0 - w]_{2x2}$ and δA_a denotes unknown model uncertainties. If we combine actuator loss of control effectiveness (4.10) and augmented state space equations, following expression is obtained.

$$\dot{x}_a(t) = A_a x_a(t) + B_a (I - \Gamma) u(t) \tag{6.5}$$

where it can be converted into additive form such as;

$$\dot{x}_a(t) = A_a x_a(t) + B_a u_a(t) + E_a \gamma(t) \tag{6.6}$$

where $\Gamma = diag(\gamma_1, \gamma_2, \gamma_3, \gamma_4)$ for quadrotor, $\gamma = [\gamma_1, \gamma_2, \gamma_3, \gamma_4]$. $E_a = -B_a U$ where $U = diag(u_1, u_2, u_3, u_4)$. We can alter the equation once more to obtain;

$$\dot{x}_a(t) = A_a x_a(t) + B_a u_a(t) + \bar{E}_a f(t) \tag{6.7}$$

where $\bar{E}_a = -B_a$ and $f(t) = U\gamma(t)$.

Now that fault augmented state equations are obtained, we can design the two step Kalman filter. This filter is discrete therefore a discrete-time model should be obtained from upper continuous time representation. Sampling rate of 100 Hz has been chosen by using zero-order hold. Discrete time state equations become;

$$\begin{aligned} x_{k+1} &= A_k x_k + B_k u_k + E_k \gamma_k + w_k^x \\ y_k &= C_k x_k + v_k \end{aligned} \tag{6.8}$$

where w_k^x and v_k are uncorrelated white Gaussian noise sequences with covariences Q_k^x and R_k. These covariances are found by designing a state feedback that makes eigenvalues of the system matrix stable. LQR method is employed for optimization in order to obtain best Q and R values with respect to the cost function $J = \sum_{k=1}^{\infty} x^T Q x + u^T R u$. A quantified fault value is formulated as;

$$\gamma_{k+1} = \gamma_k + w_k^\gamma \tag{6.9}$$

where w_k^γ denotes the zero mean white noise with covarience Q_k^γ. This quantified fault value will be sent to TSKF in order to obtain a loss of control effectivess factor(LOE) for fault detection and isolation. TSKF employes two parallel filters that inputs the augmented state matrix to estimate the fault parameter and system state [79]. The output of the filter gives these estimations by minimizing variances for state and actuator fault. First filter that is responsible for fault estimation can be formulated as;

$$\hat{\gamma}_{k+1|k} = \hat{\gamma}_{k|k} \tag{6.10}$$

$$P^\gamma_{k+1|k} = P^\gamma_{k|k} + Q^\gamma_k \tag{6.11}$$

$$K^\gamma_{k+1} = P^\gamma_{k+1|k} H^T_{k+1|k} (H_{k+1|k} P^\gamma_{k+1|k} H^T_{k+1|k} + \bar{S}_{k+1})^{-1} \tag{6.12}$$

$$\hat{\gamma}_{k+1|k+1} = \hat{\gamma}_{k+1|k} + K^\gamma_{k+1}(r_{k+1} - H_{k+1|k}\hat{\gamma}_{k+1|k}) \tag{6.13}$$

$$P^{\gamma}_{k+1|k+1} = (I - K^{\gamma}_{k+1} H_{k+1|k}) P^{\gamma}_{k+1|k} \tag{6.14}$$

We can formulate fault free subfilter that represents states as;

$$\bar{x}_{k+1|k} = A_k \bar{x}_{k|k} + B_k u_k + W_k \hat{\gamma}_{k|k} - V_{k+1|k} \hat{\gamma}_{k|k} \tag{6.15}$$

$$P^{x}_{k+1|k} = A_k P^{x}_{k|k} A^T_k + Q^x_k + W_k P^{\gamma}_{k|k} W^T_k - V_{k+1|k} P^{\gamma}_{k+1|k} V_{k+1|k} \tag{6.16}$$

$$K^{x}_{k+1} = P^{x}_{k+1|k} C^T_{k+1} (C_{k+1} Pk + 1|k^x C^T_{k+1} + R_{k+1})^{-1} \tag{6.17}$$

$$\bar{x}_{k+1|k+1} = \bar{x}_{k+1|k} + K^{x}_{k+1} (y_{k+1} - C_{k+1} \bar{x}_{k+1|k}) \tag{6.18}$$

$$P^{x}_{k+1|k+1} = (I - K^{x}_{k+1} C_{k+1}) P^{x}_{k+1|k} \tag{6.19}$$

where residual is formulated as $r_{k+1} = y_{k+1} - C_{k+1} \bar{x}_{k+1|k}$ and its covariance matrix is formulated as

$$\bar{S}_{k+1|k} = R_{k+1} + C_{k+1} P^{x}_{k+1|k} C^T_{k+1} \tag{6.20}$$

The coupling equations are as follows;

$$W_k = A_k V_{k|k} + E_k \tag{6.21}$$

$$V_{k+1|k} = W_k P^{\gamma}_{k|k} (P^{\gamma}_{k+1|k})^{-1} \tag{6.22}$$

$$H_{k+1|k} = C_{k+1} V_{k+1|k} \tag{6.23}$$

$$V_{k+1|k+1} = V_{k+1|k} - K^{x}_{k+1} H_{k+1|k} \tag{6.24}$$

Redressed estimated state and its estimated covarience becomes

$$\hat{x}_{k+1|k+1} = \bar{x}_{k+1|k+1} + V_{k+1|k+1} \hat{\gamma}_{k+1|k+1} \tag{6.25}$$

$$P_{k+1|k+1} = V_{k+1|k+1} P^{\gamma}_{k+1|k+1} V^T_{k+1|k+1} + P^{x}_{k+1|k+1} \tag{6.26}$$

TSKF is implemented on MATLAB which takes control inputs and plant outputs as input and supplies full state estimate vector for controller.

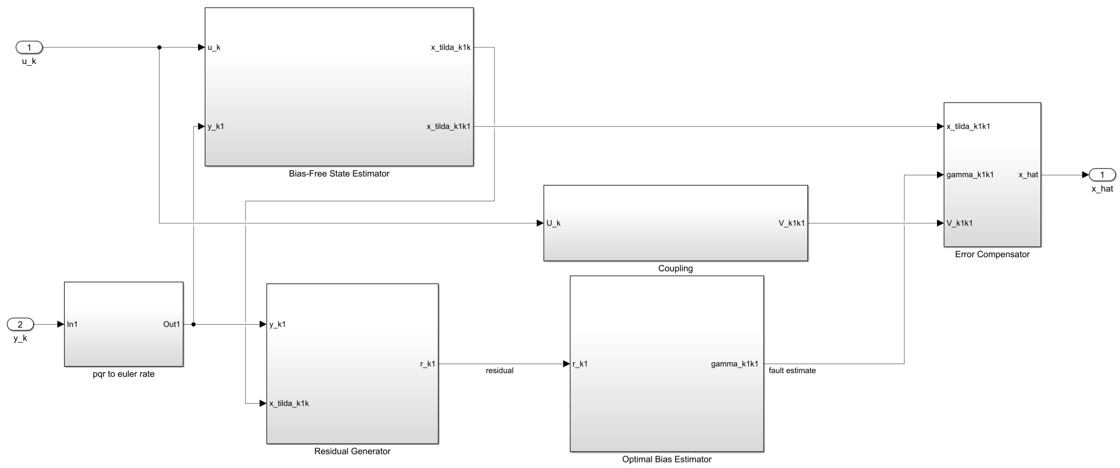

Figure 6.1: TSKF MATLAB/SIMULINK Implementation

Even though we have used step actuator LOE for our fault injection procedure, different LOE characteristics have also been tested on the filter for demonstration purposes. Given LOE factors:

Fault Type	Motor 1	Motor 2	Motor 3	Motor 4
Step	t = 38s, 50%	0	t = 22s, 45%	t = 48s, 15%
Sinusoid	t = 38s, 20%	0	t = 22s, 40%	0
Random	$(0 \leq LOE \leq 20)$	$(0 \leq LOE \leq 20)$	$(0 \leq LOE \leq 20)$	$(0 \leq LOE \leq 20)$
Pulse	0	t = 22s, 25%	0	t = 44s, 30%

Table 6.1: Injected Actuator LOEs

TSKF can successfully estimate different fault types as can be seen in Figure 6.2.

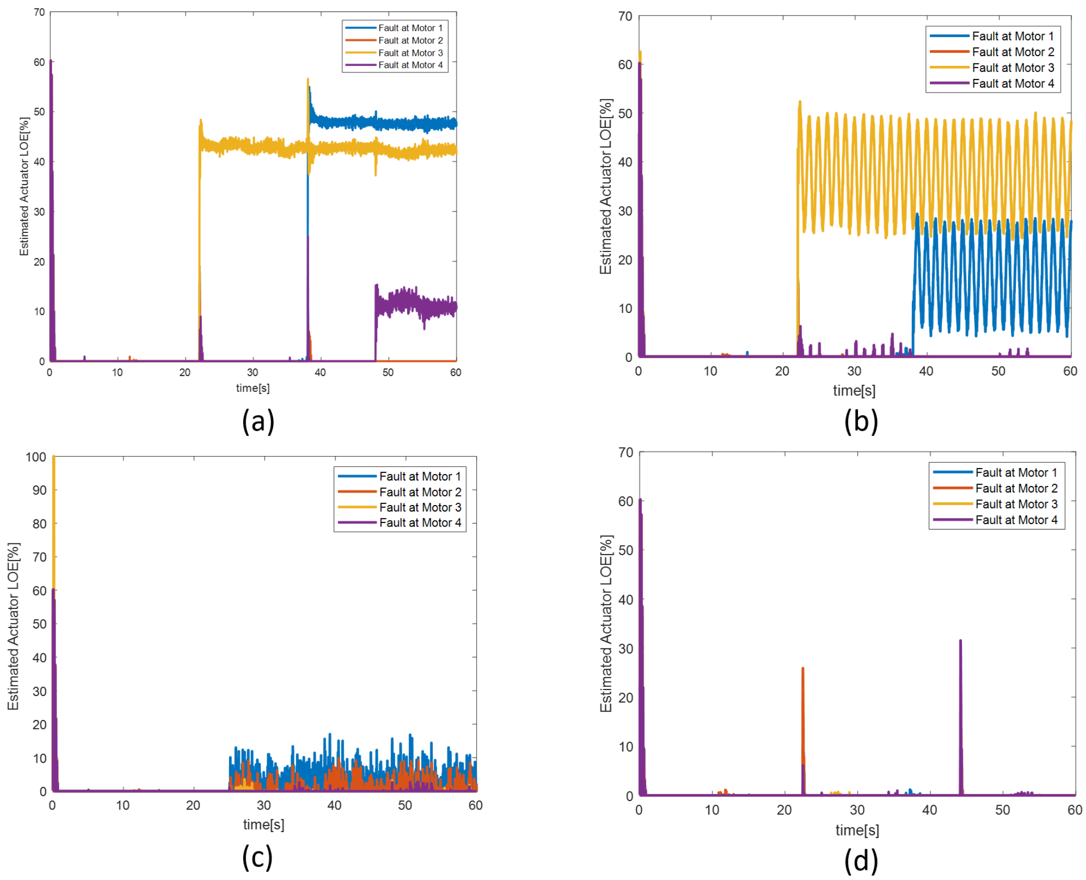

Figure 6.2: TSKF Estimated Faults: (a) Step; (b) Sinusuoid; (c) Random Number; (d) Pulse

Although it is possible to add multiple fault behaviours, we have chosen to use step LOE for all the simulations due to being the most challenging one.

6.1.2 Extended Kalman Filter

Rather than directly giving linear state space model as in the case of a regular Kalman filter, Extended Kalman filter takes nonlinear plant model where it linearizes the model itself in every simulation step by using Jacobians. EKF is a highly useful tool in terms of catching non-linearaties and improve estimations with this recursive behaviour accordingly. Although it is pretty obvious that its more computationally heavy with respect to regular Kalman observer; it may undeniably serve much more accurate state estimations especially as in our fault injected cases.

We have used Simulink's built-in EKF block where we have embedded our nonlinear quadrotor model. It should also be denoted that EKF block takes discretized inputs which is why zero-order-hold blocks are used which can be seen in Figure 6.3.

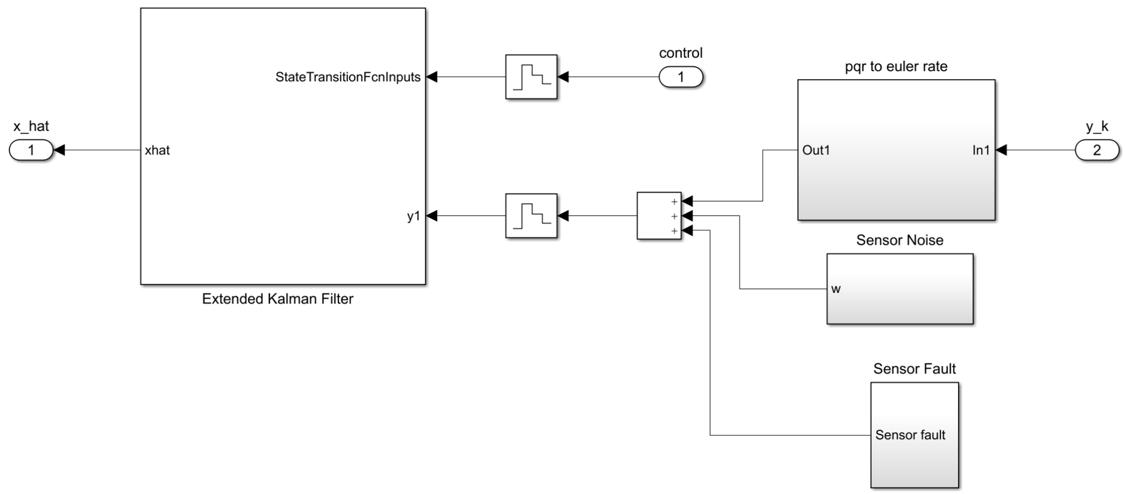

Figure 6.3: EKF Implementation in MATLAB/SIMULINK

Due to adaptive architecure of EKF where every simulation cycle is handled as a separate configuration, EKF does not need an extra fault estimation term. In TSKF we have used a separate embedded bias estimator in order to improve observer estimate by being aware of faults but with EKF this burden is removed where each full state estimation is indirectly manipulated by any disturbance respectively. Therefore, EKF's ability to adapt itself under nonlinearity, provided us full state estimate without needing an isolated fault estimation model.

6.2 Disturbance Estimation

Disturbance observers are very useful tools in order to cancel effects of uncertainties and external disturbances. An estimated lumped disturbance parameter is fed back as a compensation signal in which counteracts effects of that disturbance value[80]. Many applications[81] can be found in literature due to its simple architecture for implementation and abilty to reject disturbances. However, disturbances can be estimated if the disturbance signal is under coverage of used low-pass filter's bandwith for DOB[83].

An overview for DOB can be presented with the following simple to understand block diagram;

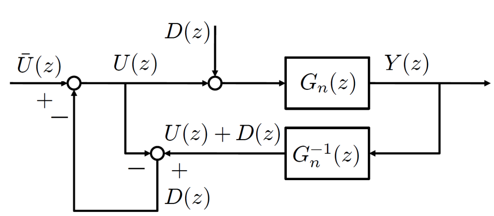

Figure 6.4: Block Diagram of a Reversible Plant[82]

One can see from this block diagram that ideally, we could obtain a disturbance term $D(z)$ by subtracting the control input $U(z)$ from nominal plant output $G_n^{-1}(z)$. This looks fairly simple however, $G_n^{-1}(z)$ is not realizable because given plant $G_n(z)$ might not be exactly as it is on the model. Therefore, even being not sure about reversing the plant as true as possible to the nominal plant, sensor noise will be added to reversed plant as well. Briefly, we can't be sure about modelling all the dynamics inside the plant which corrupts the notion of estimating a good disturbance term. At this point, we introduce a new term $Q(z)$ which is the low-pass filter.

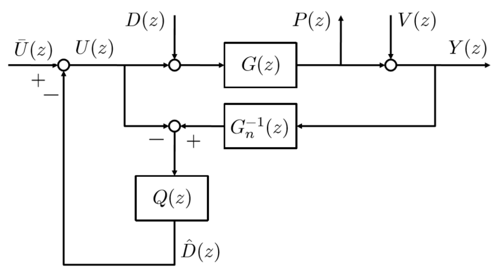

Figure 6.5: Realizable Disturbance Observer[82]

This filter is employed to make dynamics from $U(z)$ and $Y(z)$ to $\hat{D}(z)$ realizable. The task of our DOB is to estimate such lumped disturbance value for disturbance rejection. Figure 6.5 shows the block diagram of a conventional DOB based controller. On the other hand, we have implemented a realizable DOB structure which comes more handy for disturbance estimation.

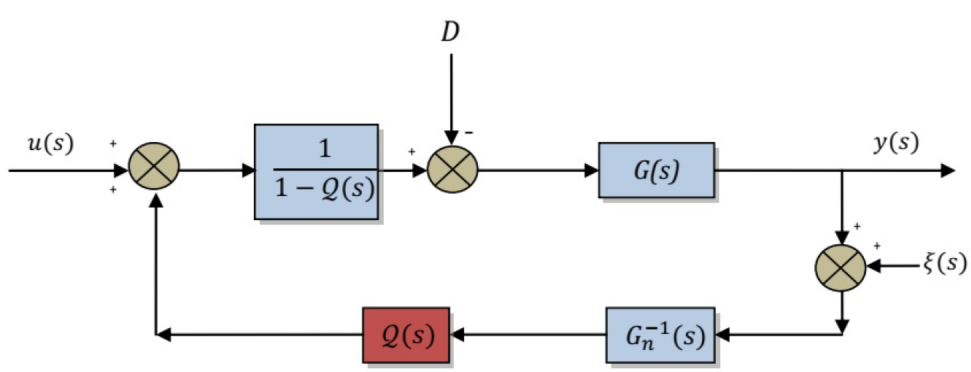

Figure 6.6: A More Convenient DOB Architecture [83]

We have implemented two types of DOB architectures which are Velocity-Based and Acceleration-Based disturbance observers. Their working principle is same but nominal plants and corresponding Q filters differ.

For AbDOB, nominal plant model is constructed as;

$$G_n(s) = \begin{bmatrix} \frac{1}{I_{xx}s^2} & 0 & 0 \\ 0 & \frac{1}{I_{yy}s^2} & 0 \\ 0 & 0 & \frac{1}{I_{zz}s^2} \end{bmatrix} \tag{6.27}$$

which uses as its filter.

$$Q_{AbDOB} = \frac{w^2}{s^2 + 2w + w^2} \tag{6.28}$$

For VbDOB, nominal plant model is constructed as;

$$G_n(s) = \begin{bmatrix} \frac{1}{I_{xx}s} & 0 & 0 \\ 0 & \frac{1}{I_{yy}s} & 0 \\ 0 & 0 & \frac{1}{I_{zz}s} \end{bmatrix} \tag{6.29}$$

which uses as its filter.

$$Q_{VbDOB} = \frac{w}{s + w} \tag{6.30}$$

Filter cut-off parameter w is chosen as $w = 5$ for each alternative.

Chapter 7

Simulation Results

In this chapter, simulation results are presented for various implemented configurations. Due to having plenty of configurations and corresponding result sets, only one trajectory configuration is presented for preventing excessive result deposition.

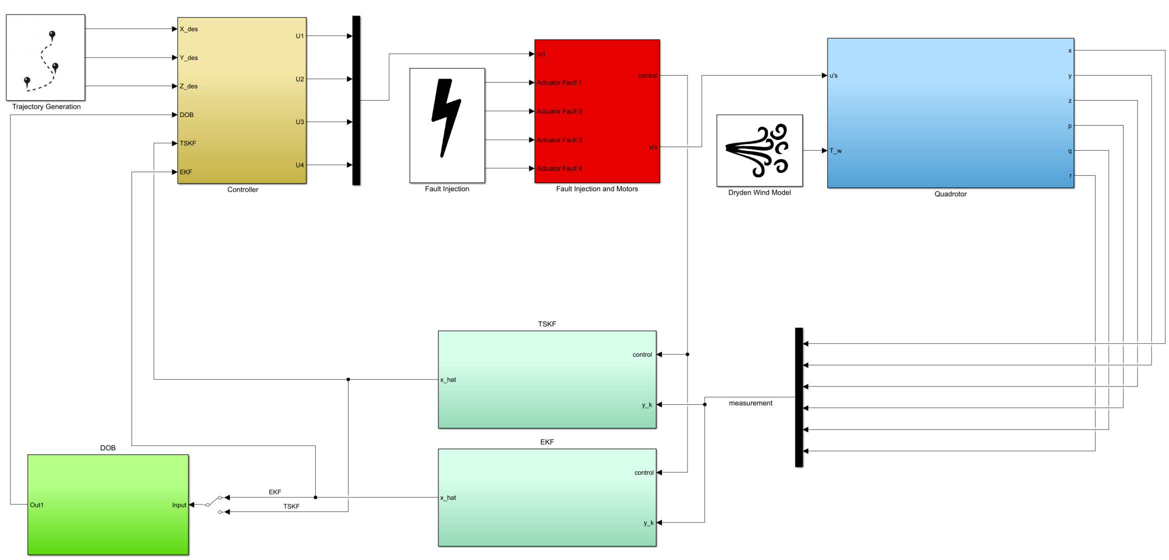

Figure 7.1: MATLAB/SIMULINK Model

Simulation parameters are originated from Concordia University's QBALL-X4 platform[26] which are tabulated in Table 7.1.

Table 7.1: Simulation Parameters

Parameter	Description	Value
K	Thrust Gain	175
w	Motor Bandwith	15 rad/s
L	Distance from Motor to C.G	0.2 m
K_ψ	Thrust to Moment Gain	0.023
m	Mass	1.42 kg
g	Gravity	9.81 m/s^2
$I_{xx}; I_{yy}; I_{zz}$	Moments of Inertia	0.03; 0.03; 0.04 $kg.m^2$

Results are presented with the following layout;

- **Baseline Configuration:** This is the reference result which neither contains any fault nor added compensation mechanism. It is presented to visualize designed cubic trajectory generation and corresponding basic controller response. Every other following section can be thought as a supplementation to this reference configuration.
- **Controller Effects:** In this section, effects of controller types are discussed. Due to having a singular PID outer controller for each controller architecture, only inner controller variations are discussed. Therefore, presented variations due to controllers are specified to inner controllers notably.
- **Disturbance Observer Effects:** Effects of disturbance observers are presented at this section.
- **State Observer Effects:** This section compares effects of TKSF and EKF state observers that are used under load of actuator faults. Results presented are targeted to present the importance of accurate state estimation methods with high non-linearity.
- **Various Types of Faults Effects:** This part is the only section where other fault types are injected on to the system. Multiple configurations of reaction to various faults are presented.

7.1 Baseline Configuration

A reference case is presented here to lay basis for every other following implementation. A trajectory is set to our quadrotor where it first hovers to 0.6 meters and draws a $1m^2$ square in air by remaining the same altitude. Dryden wind model is implemented and all the results presented are under its impact. Also, process and measurement noise is always acting on the system independently of any scenario. Ideal trajectory we would like to follow can be seen in Figure 7.2.

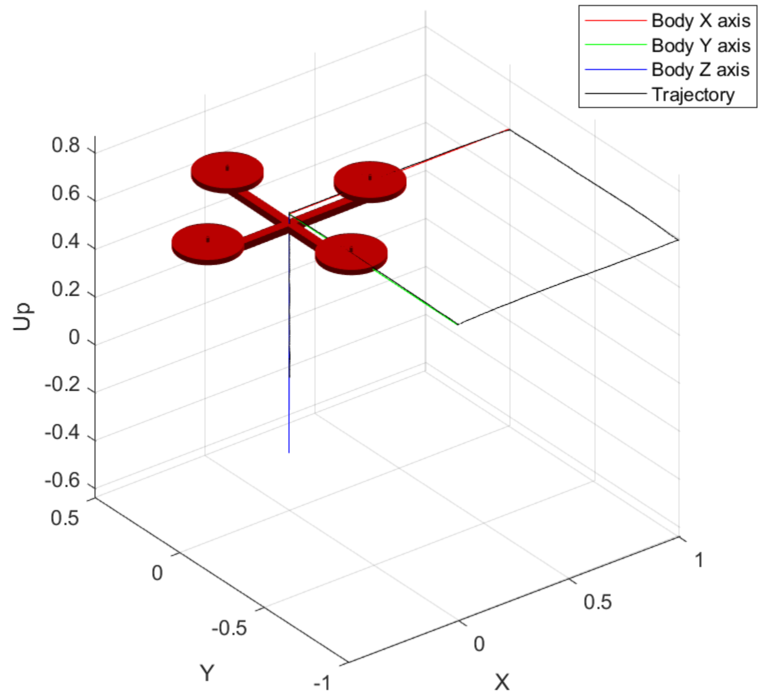

Figure 7.2: Visualized Quadrotor Trajectory

However, results will be presented in graphical form such as Figure 7.3 because later when we add faults, its effects can be visualized better on the graph.

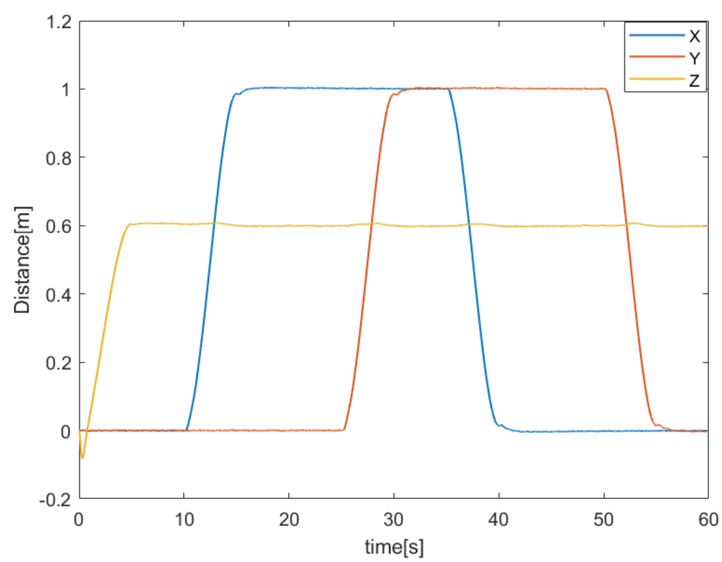

Figure 7.3: Quadrotor Trajectory in Baseline Configuration

For the baseline configuration, LQG controller is used in order to get the best response to assign as a reference. Following visualizations are given for understanding error dynamics and system performance.

Again, this is the lowest error generating condition in which a tabulated form will be presented for every result set.

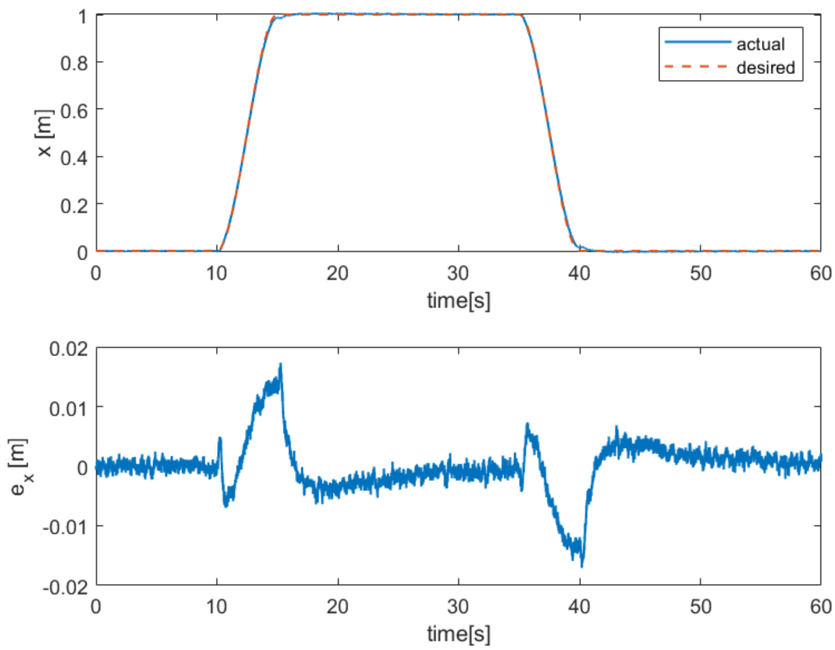

Figure 7.4: X position of the quadrotor (top), position error (bottom)

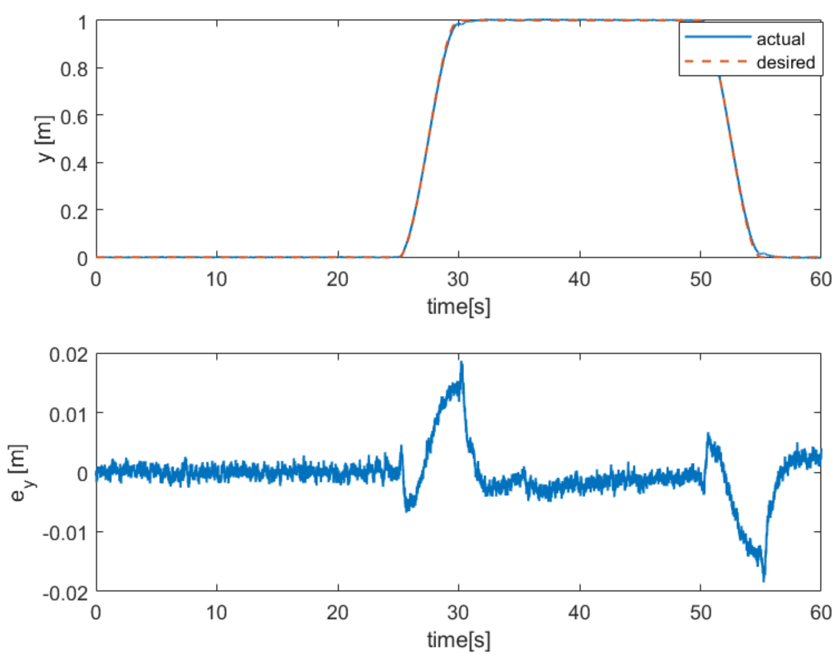

Figure 7.5: Y position of the quadrotor (top), position error (bottom)

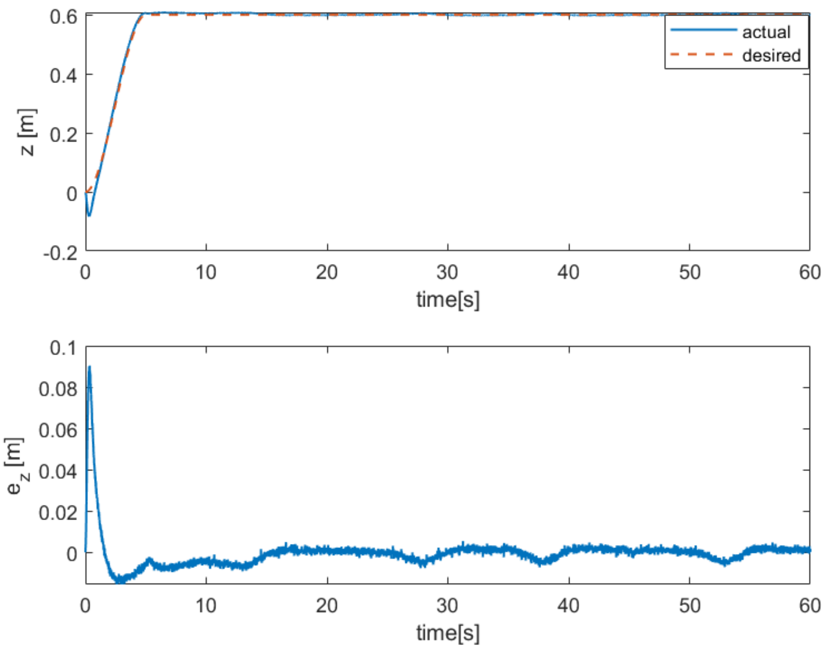

Figure 7.6: Z position of the quadrotor (top), position error (bottom)

Trajectory tracking is conveyed very smoothly and tracking errors are very low as expected.

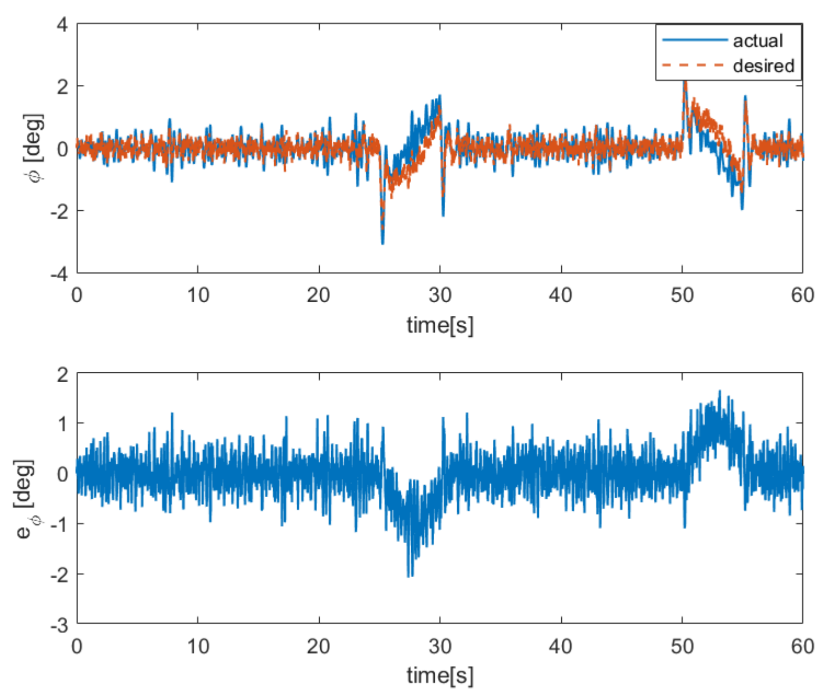

Figure 7.7: Roll angle of the quadrotor (top), tracking error (bottom)

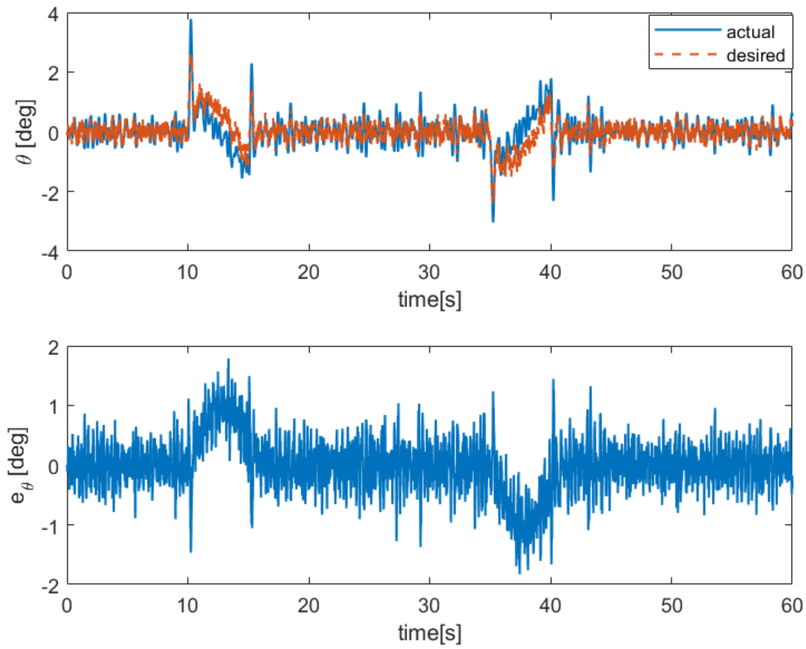

Figure 7.8: Roll angle of the quadrotor (top), tracking error (bottom)

Following table construction is used for presenting root mean square (RMS) and maximum values of errors for measurement states.

Table 7.2: Tracking Errors for Baseline Configuration

Error	RMS Errors	Max Errors
e_x	0.0044	0.0179
e_y	0.0042	0.0184
e_z	0.008	0.0907
e_ϕ	0.4520	1.5402
e_θ	0.4761	1.9486
e_ψ	0.0187	0.2310

PWM outputs for all motors are smooth and linear. Motors do not have to operate under full power and controller task is conveyed by an even distribution on control signal. Motor PWM output is modelled to be maximum 0.05 under full load and 0 under no load with respect to given actuator response.

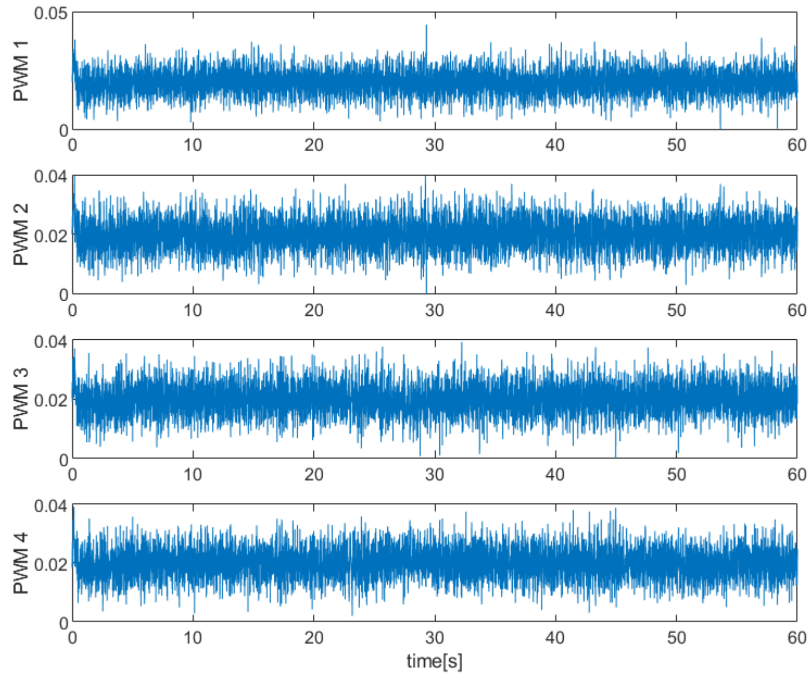

Figure 7.9: PWM Signals for All Motors

Controller effort is calculated for every scenario by summing all the PWM values and divide it by time interval to get a mean value. Average control effort for the baseline controller is 0.07947.

7.2 Controller

This section presents results for both controller architectures.

From now on, results presented have the same fault injection type which is actuator loss of control effectiveness until otherwise specified. Corresponding fault injection arrangement can be seen in Table 7.3.

Table 7.3: Actuator Fault (LOE) Injection

Motor 1	Motor 2	Motor 3	Motor 4
30%, t=38s	0%	20%, t=22s	15%, t=48s

LQG Control

First, LQG controller response will be shown after injecting actuator loss of effectiveness faults. Due to only making controller comparison, state feedback should remain the same which is supplied from the Extended Kalman filter.

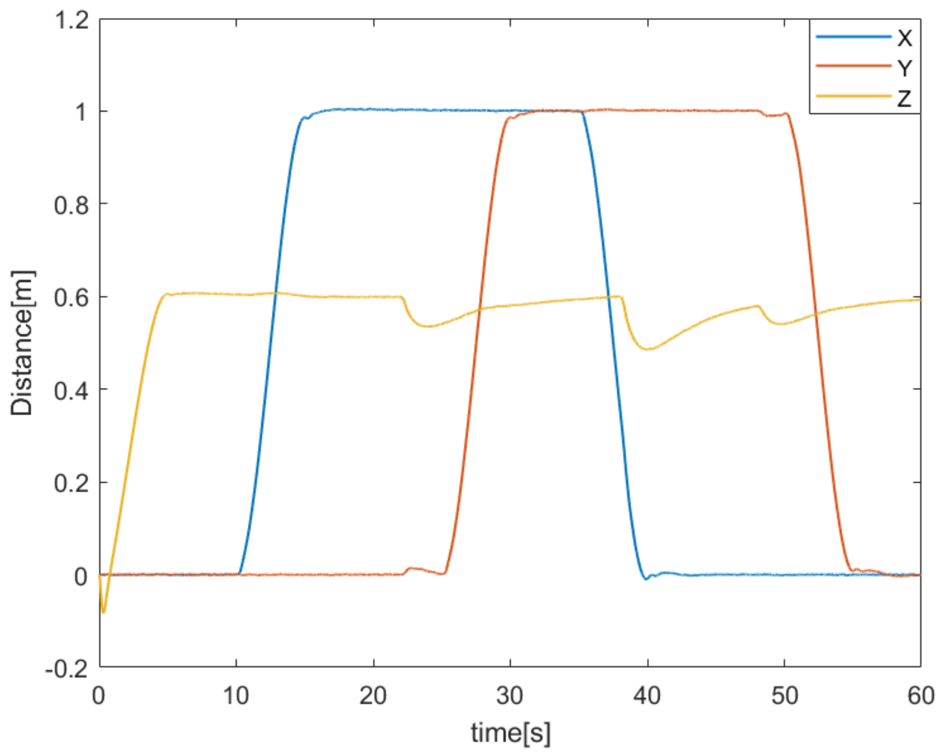

Figure 7.10: LQG Trajectory Following with Actuator LOE

One can immediately realize that fault injection has brought drop in altitude. Drops in altitude are proportional to amount of actuator loss as expected. Here, we can see that no matter which actuator looses control effectiveness; altitude is lost respectively. However, LQG controller performance is robust enough to compensate the injected faults.

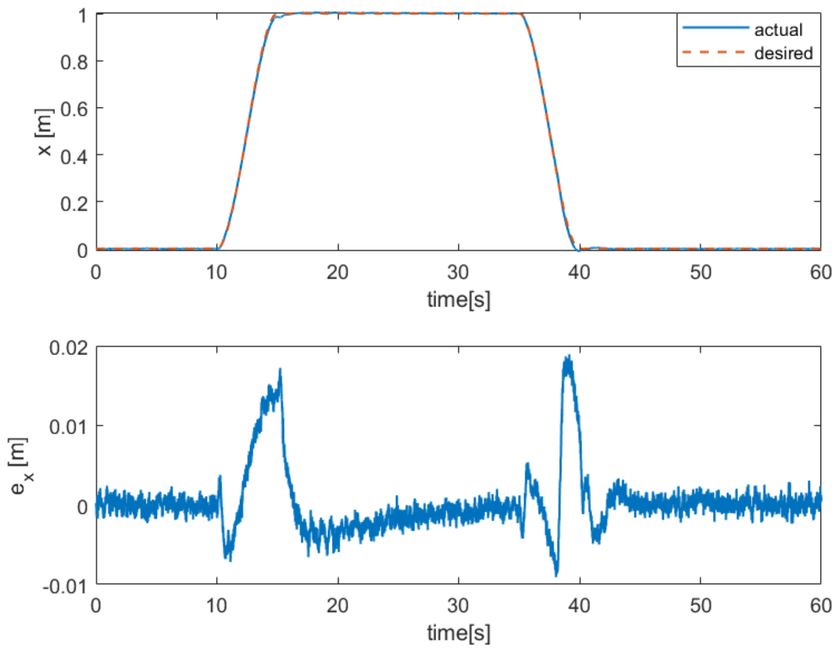

Figure 7.11: X position of the quadrotor (top), position error (bottom)

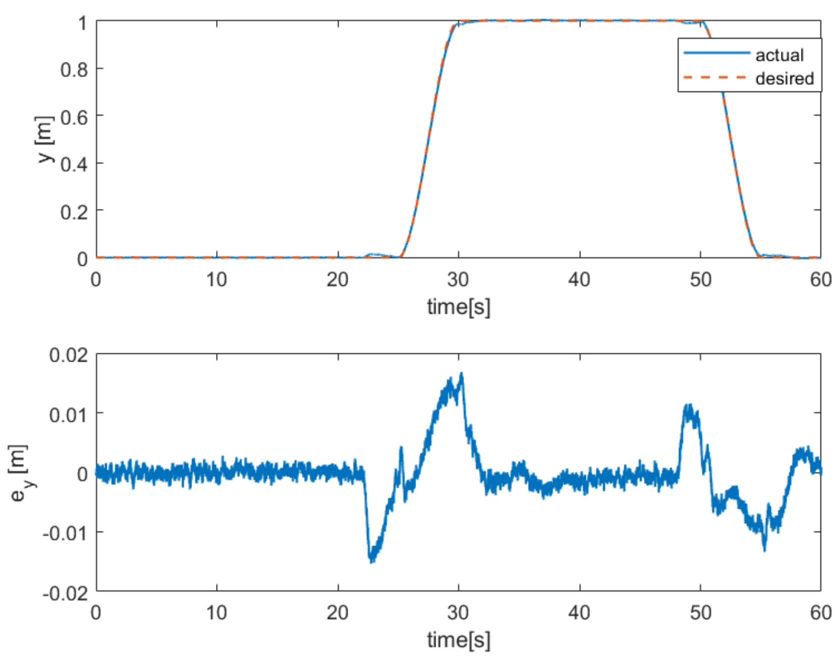

Figure 7.12: Y position of the quadrotor (top), position error (bottom)

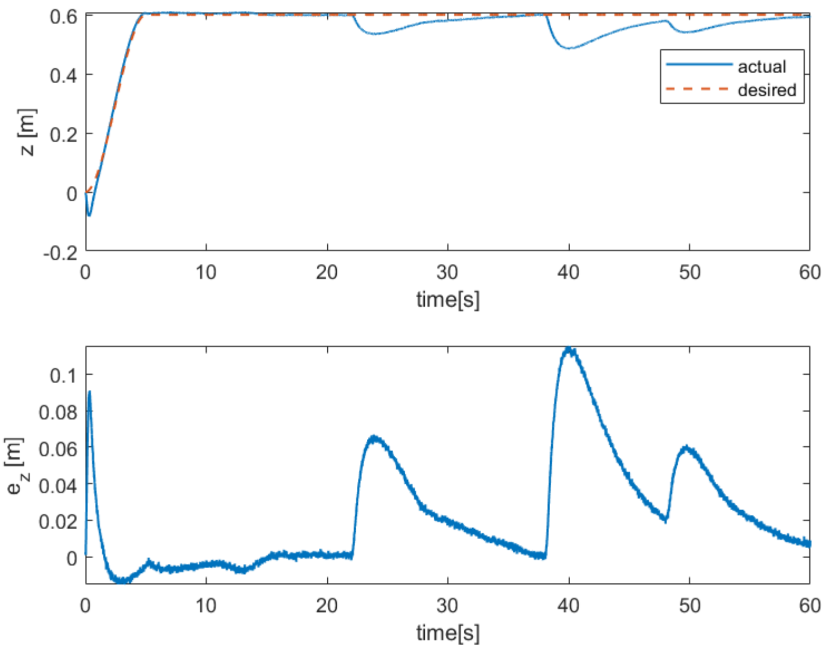

Figure 7.13: Z position of the quadrotor (top), position error (bottom)

Tracking on X and Y directions are not effected but changes in Z is pretty visible. This is because the feedforward gravity term at Z that is involved in the nonlinear quadrotor model (3.18).

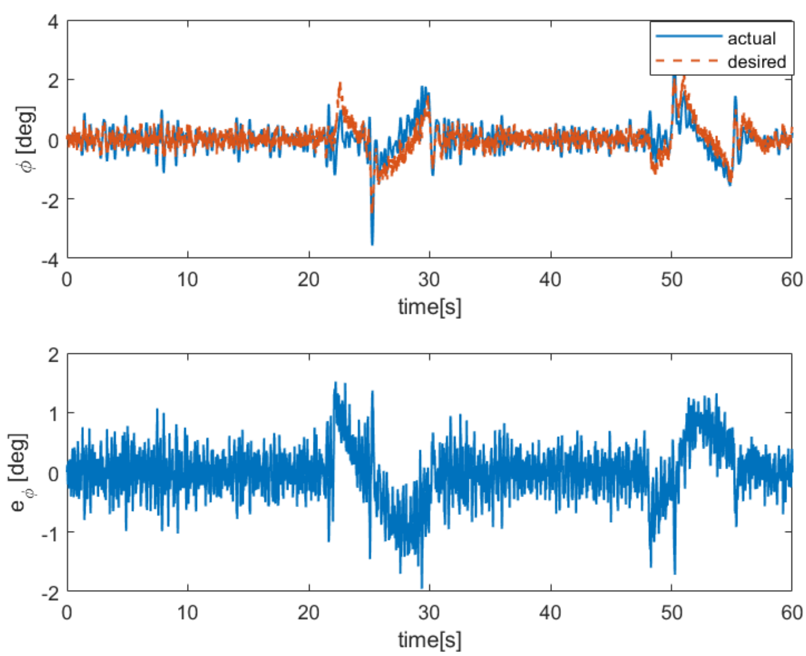

Figure 7.14: Roll angle of the quadrotor (top), tracking error (bottom)

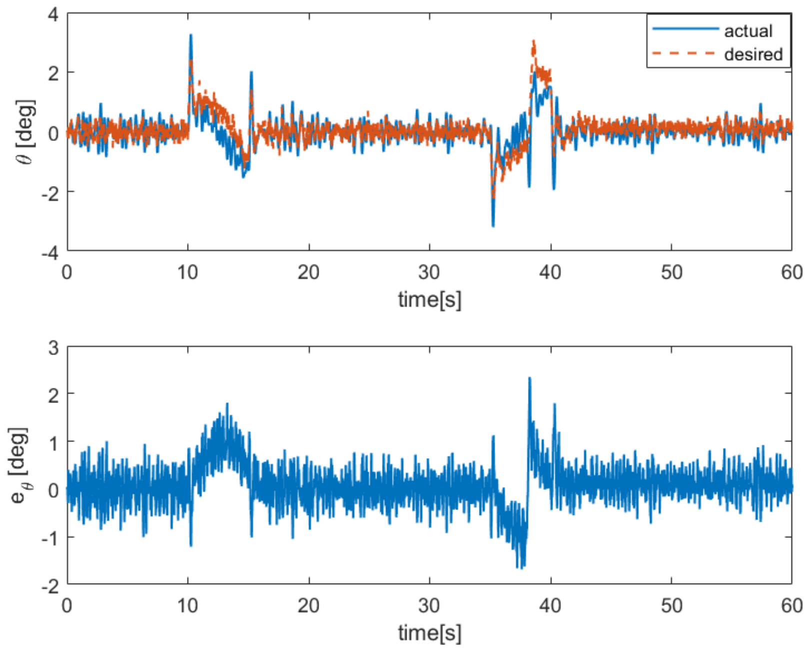

Figure 7.15: Pitch angle of the quadrotor (top), tracking error (bottom)

Tracking error is increased for attitude dynamics with fault injection as expected however error residuals are not high and controllable.

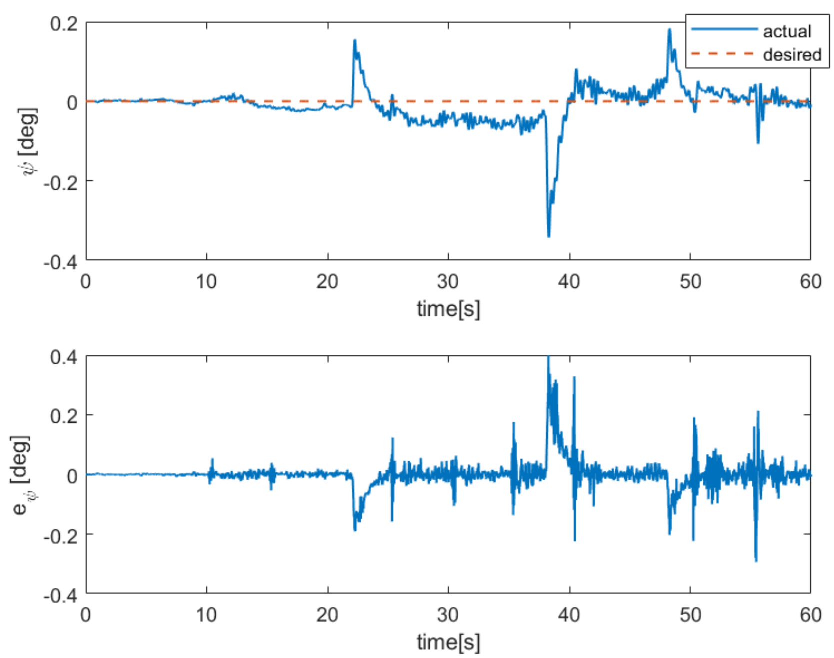

Figure 7.16: Yaw angle of the quadrotor (top), tracking error (bottom)

Table 7.4: Tracking Errors for LQG Controller with Actuator LOE

Error	RMS Errors	Max Errors
e_x	0.0040	0.0189
e_y	0.0046	0.0168
e_z	0.0376	0.1154
e_ϕ	0.4469	1.5214
e_θ	0.4413	2.3469
e_ψ	0.0439	0.3998

Fault injections on $1^{st}, 3^{rd}$ and 4^{th} motor is pretty visible on PWM signals as well. If controller detects actuator loss on a motor, it opens up that motor proportionally more to compensate fault. This is of course under saturation conditions where a limited increase in voltage is possible. One can see that highest fault value of 30% on first rotor creates a big margin over standard linear behaviour.

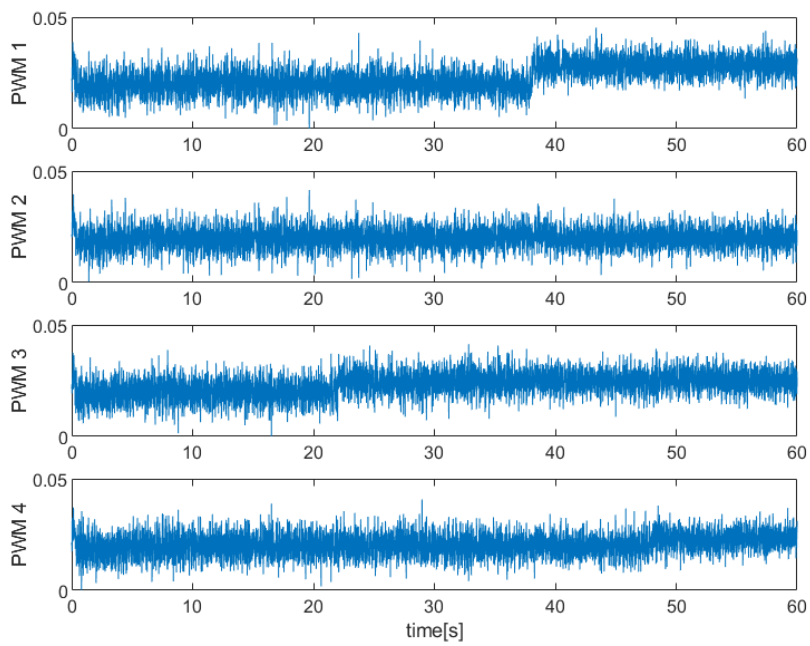

Figure 7.17: PWM Signals for LQG Controller with Actuator LOE

Due to increase in PWM signals, average control effort has increased excessively over baseline configuration having a value of 0.08645 on behalf of fault compensation.

PD Control

A PD controller is also implemented for controlling attitude dynamics under fault tolerant control scheme. Integral control is eliminated here because fault introduction combined with this controller architecture yields to high oscillations and hard to recover error residuals. Therefore, in order not to increase error and lack of observed steady state error; control dynamics let us to eliminate integral feedback. Presented PD controller results below are showing significantly poorer trajectory tracking outcomes but it constructs a good comparison study.

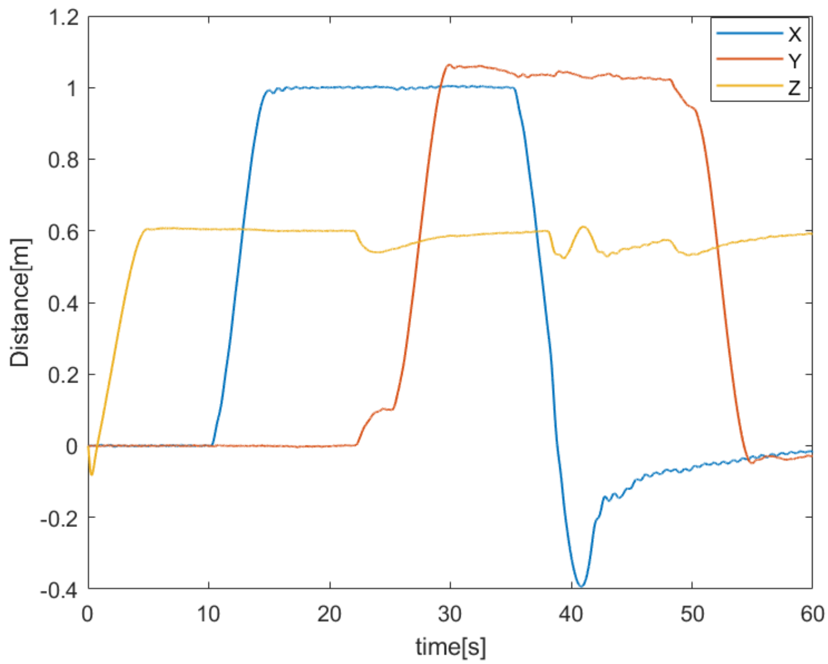

Figure 7.18: PD Trajectory Following with Actuator LOE

As can be seen from Figure 7.18, quadrotor is highly deflected from intended trajectory. Unlike LQG, PD controller is not a type of optimal controller and its results may be cumbersome if nonlinearity and amount of controlled states are high. In order to compensate sudden faults that has high burden on controller, control gains had to be very high initially. Even though this works for some simulation scenarios, many impudent configurations really challenge the system and makes PD controller fail. Therefore, we have added desired double derivatives of trajectory $\ddot{X}_d, \ddot{Y}_d, \ddot{Z}_d$ as feedforward to our controller which enabled a better response via employing lower gains. Accordingly, a more robust controller architecture is fulfilled and enabled us to experiment on various configurations.

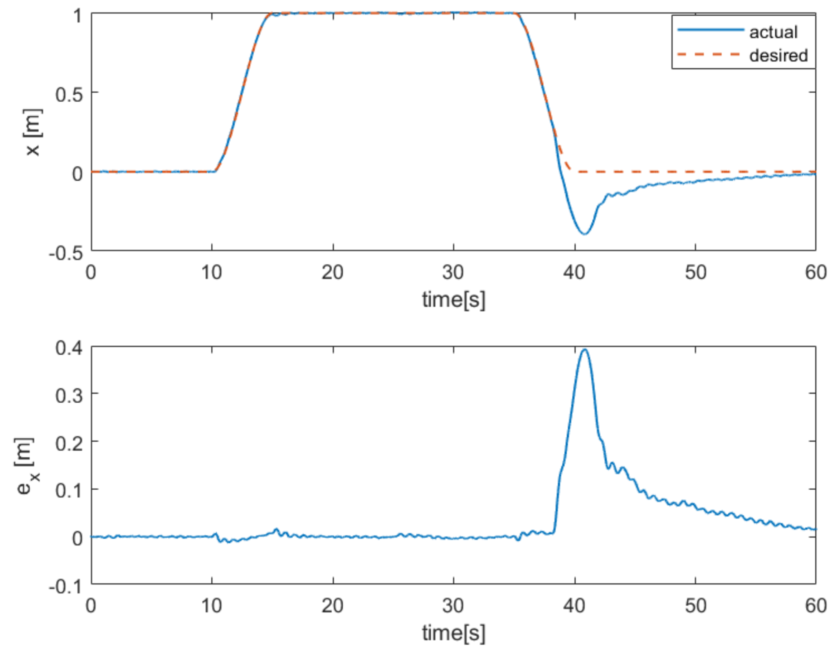

Figure 7.19: X position of the quadrotor (top), position error (bottom)

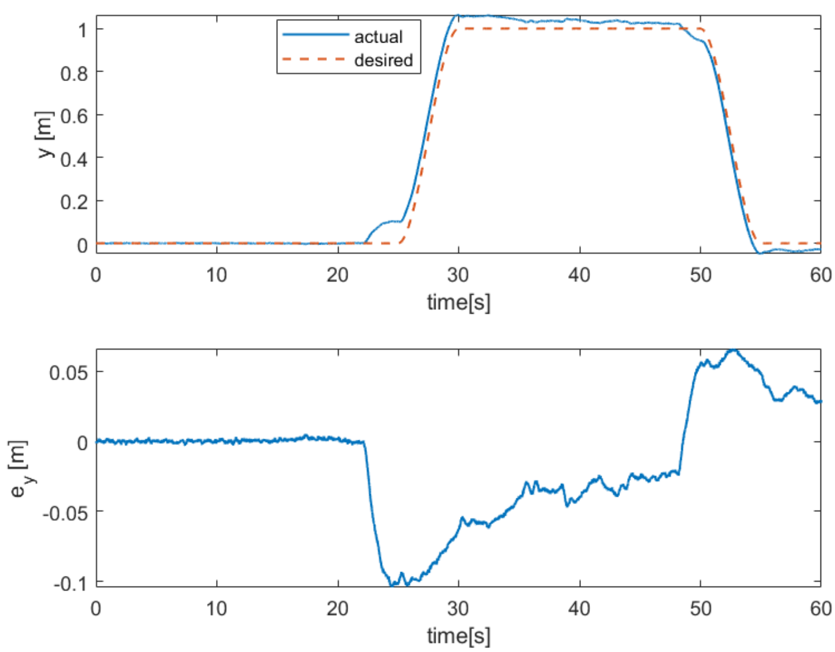

Figure 7.20: Y position of the quadrotor (top), position error (bottom)

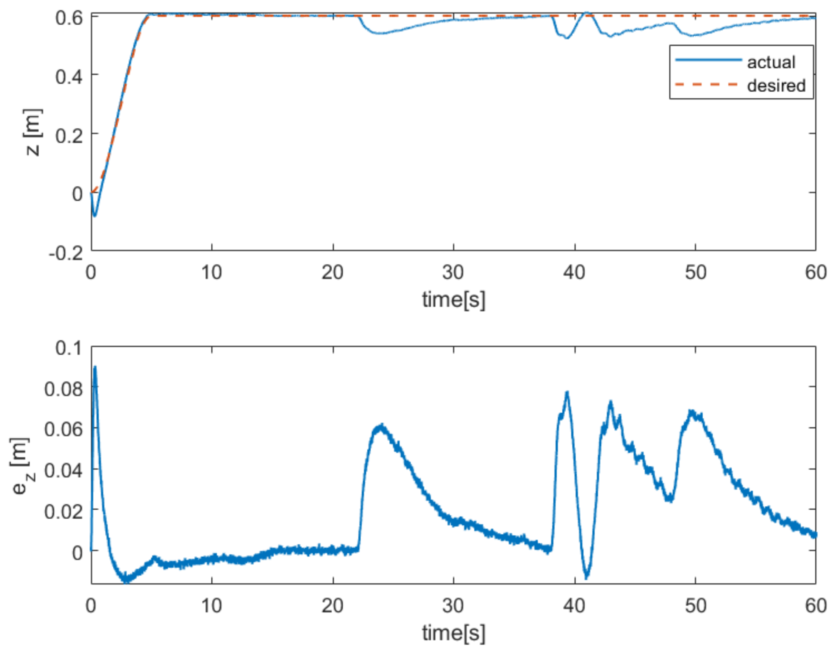

Figure 7.21: Z position of the quadrotor (top), position error (bottom)

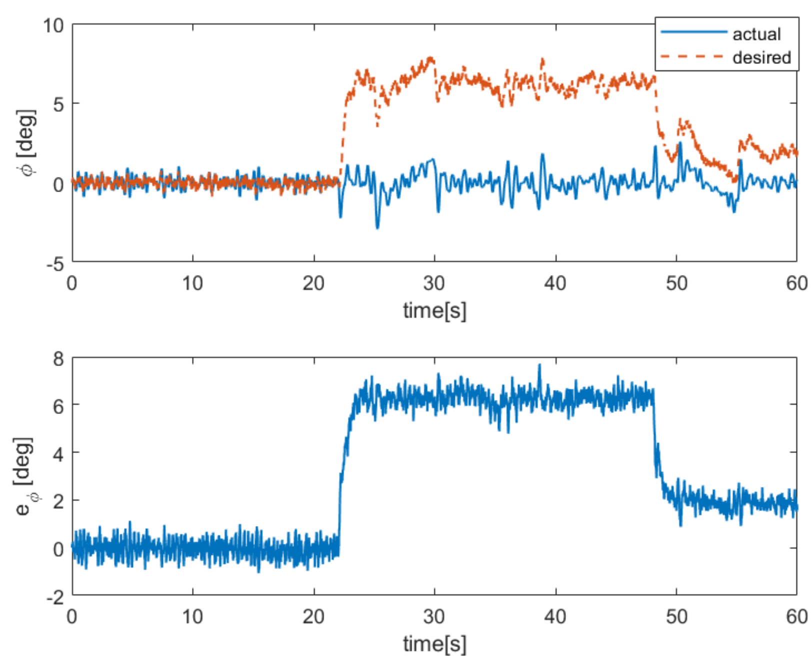

Figure 7.22: Roll angle of the quadrotor (top), tracking error (bottom)

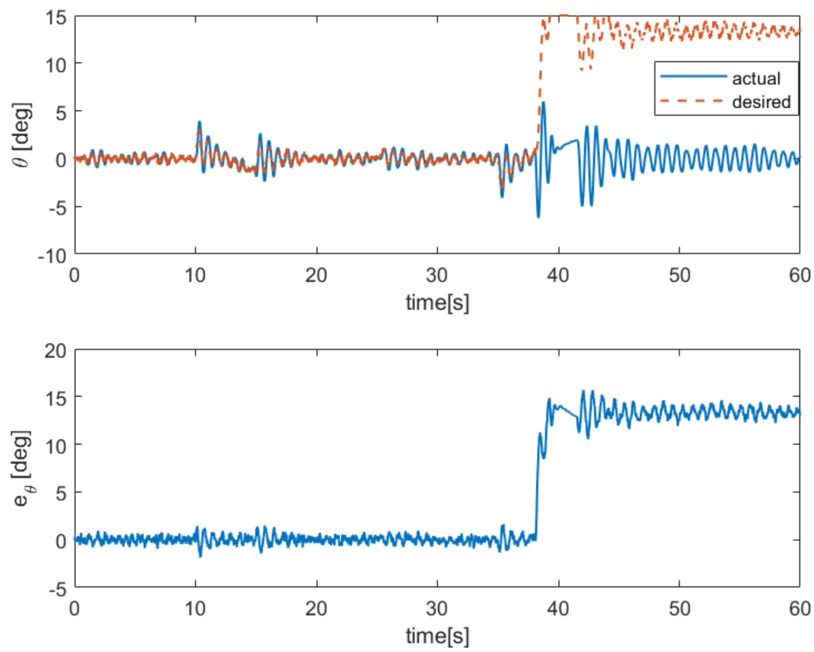

Figure 7.23: Pitch angle of the quadrotor (top), tracking error (bottom)

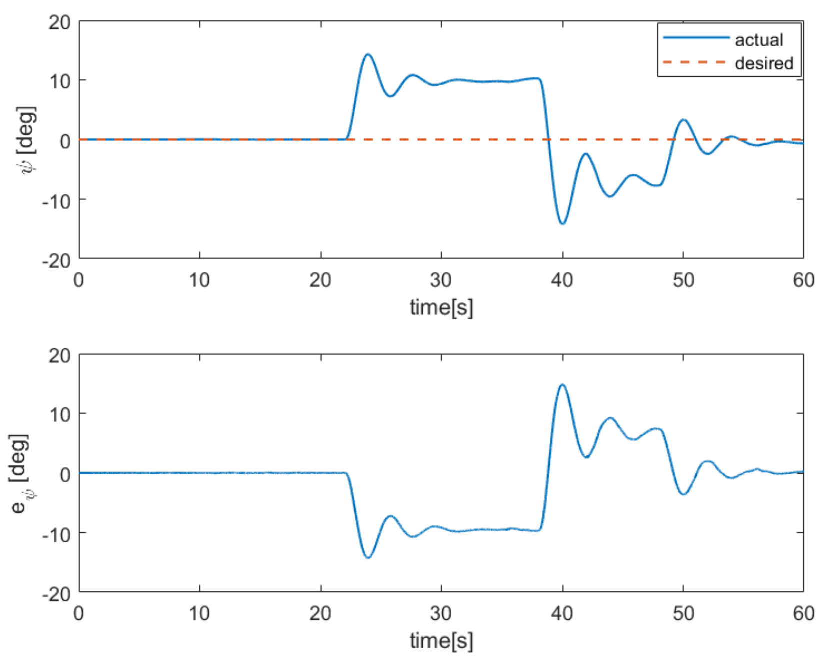

Figure 7.24: Yaw angle of the quadrotor (top), tracking error (bottom)

Lower frequency position controller can still show acceptable performance however it is not the same case with high frequency attitude controller. Tracking error on Euler angles have increased very significantly and

PD controller is having a very hard time coping with this error. Roll angle is controlled at some extent but pitch angle is completely lost after second fault injection to motor 1. Results may seem distant from LQG controller but with introduction of the disturbance observer at the next section, this gap will close spectacularly. This is why direct PD response to faults are shown here.

Table 7.5: Tracking Errors for PD Controller with Actuator LOE

Error	RMS Errors	Max Errors
e_x	0.0827	0.3937
e_y	0.04286	0.0661
e_z	0.0307	0.0902
e_ϕ	4.1892	7.7096
e_θ	7.9748	15.712
e_ψ	6.0053	14.8623

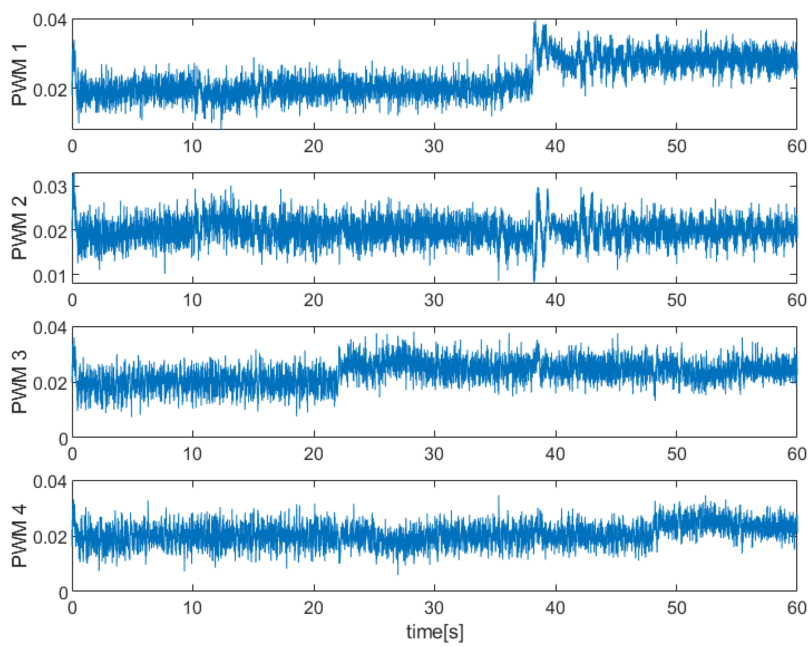

Figure 7.25: PWM Signals for PD Controller with Actuator LOE

Average control effort with this case is found to be 0.08665.

7.3 Disturbance Observer

This section shows differences of observer types for both controller architectures. Again, fault type is fixed to only actuator LOE. First implementation is configured to be added on PD controller with high tracking error. The velocity-based disturbance observer perfectly helps PD controller to reduce tracking error.

VbDOb on PD Controller with Actuator LOE

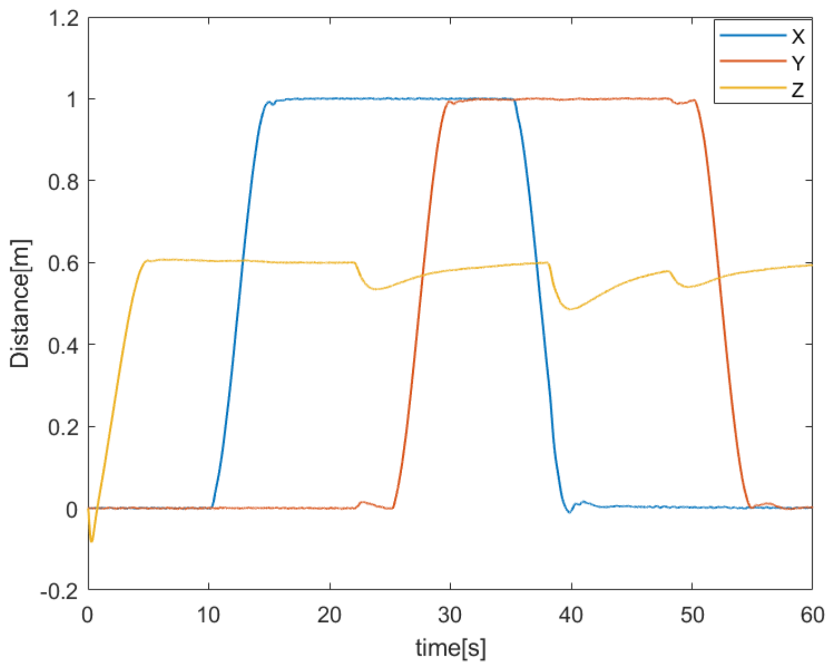

Figure 7.26: VbDOB on PD Controller Trajectory Following with Actuator LOE

As can be seen from Figure 7.26, Vbdob has improved results incredibly. If one would compare this figure with non-dob PD configuration at Figure 7.18, a huge difference can be seen. Vbdob is the best option to be applied on the PD controller where control performance is very close to LQG control.

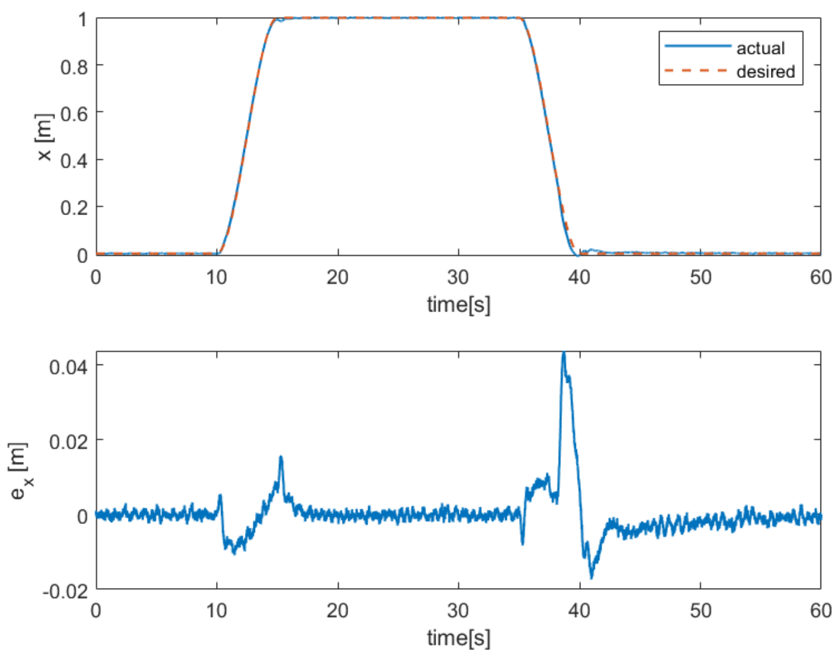

Figure 7.27: X position of the quadrotor (top), position error (bottom)

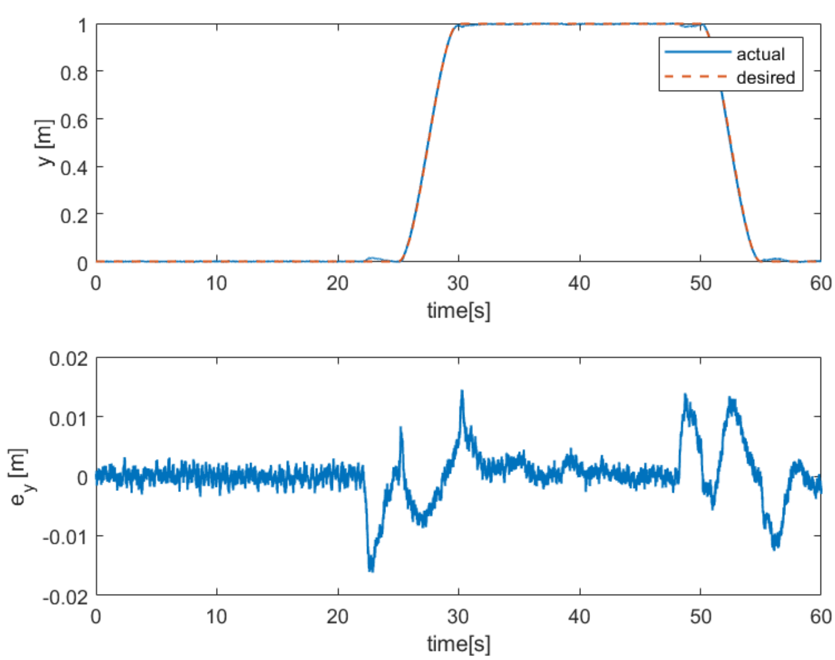

Figure 7.28: Y position of the quadrotor (top), position error (bottom)

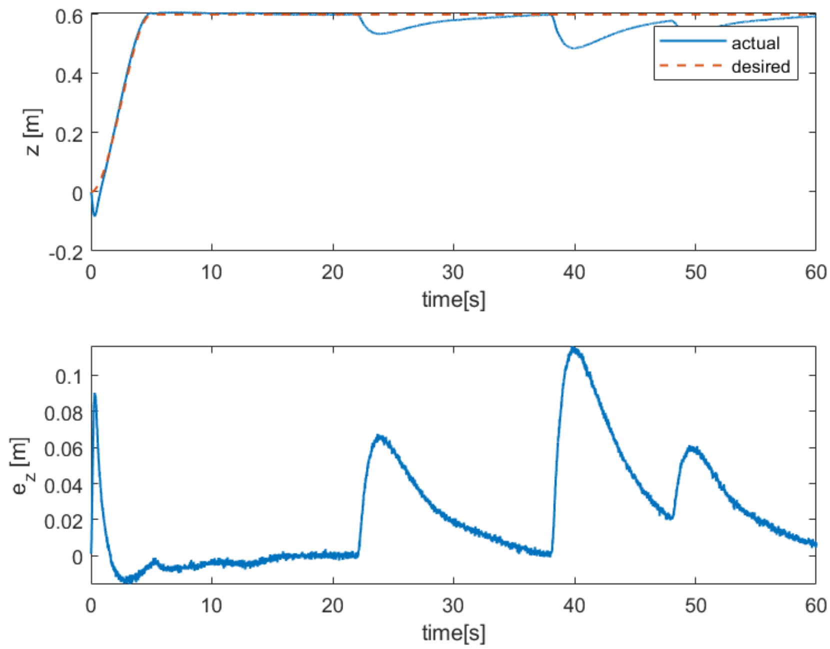

Figure 7.29: Z position of the quadrotor (top), position error (bottom)

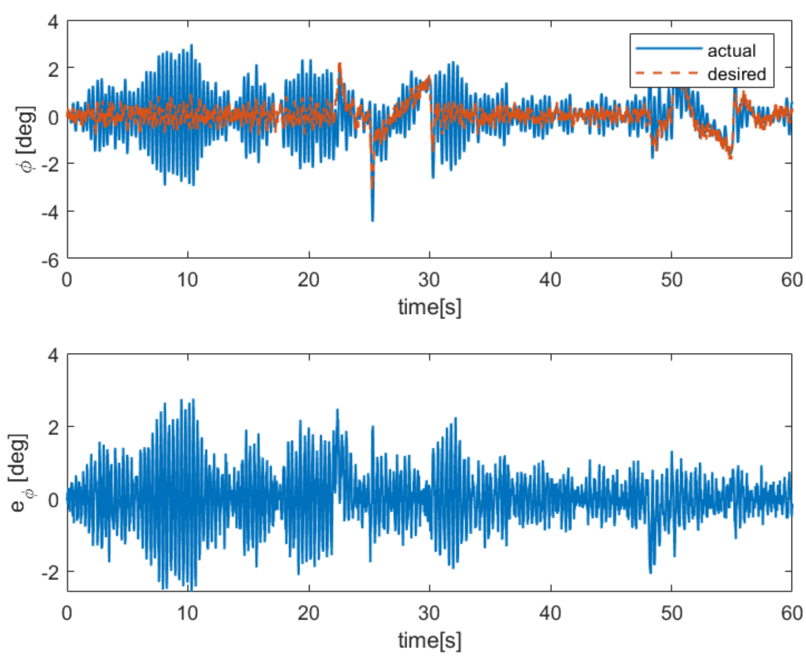

Figure 7.30: Roll angle of the quadrotor (top), tracking error (bottom)

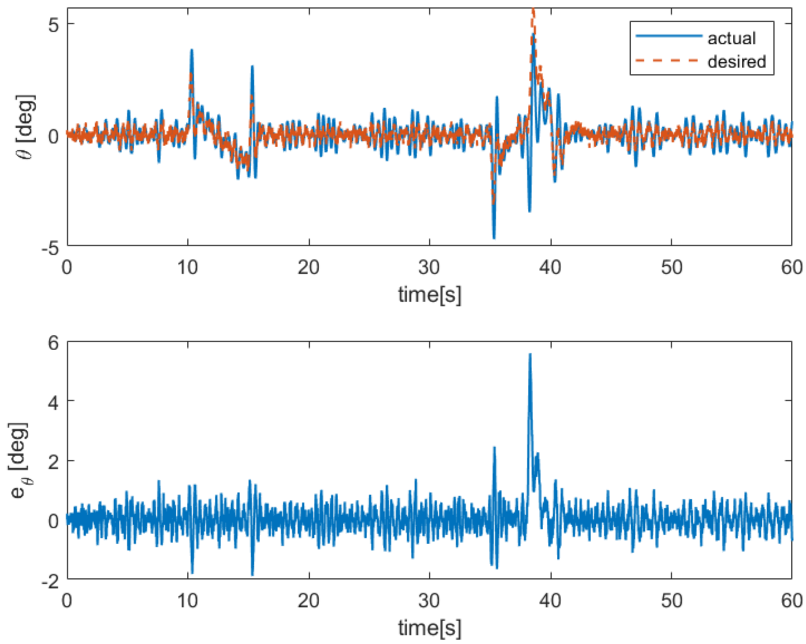

Figure 7.31: Pitch angle of the quadrotor (top), tracking error (bottom)

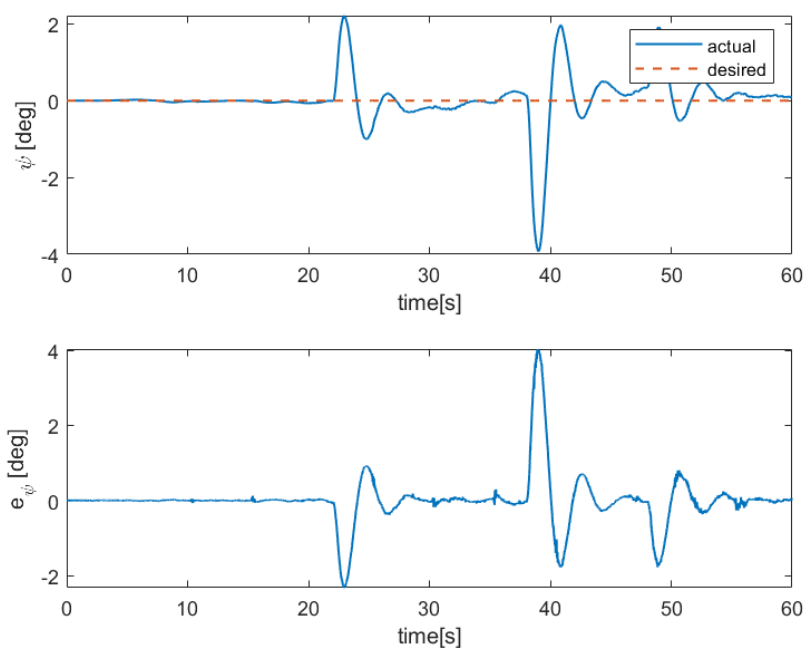

Figure 7.32: Yaw angle of the quadrotor (top), tracking error (bottom)

Disturbance observer has shown exceptional results in terms of error tracking.

Table 7.6: Tracking Errors for VbDOB ob PD Controller with Actuator LOE

Error	RMS Errors	Max Errors
e_x	0.0062	0.0437
e_y	0.0040	0.0144
e_z	0.0375	0.1161
e_ϕ	0.7817	2.7508
e_θ	0.5489	5.5796
e_ψ	0.6610	4.0297

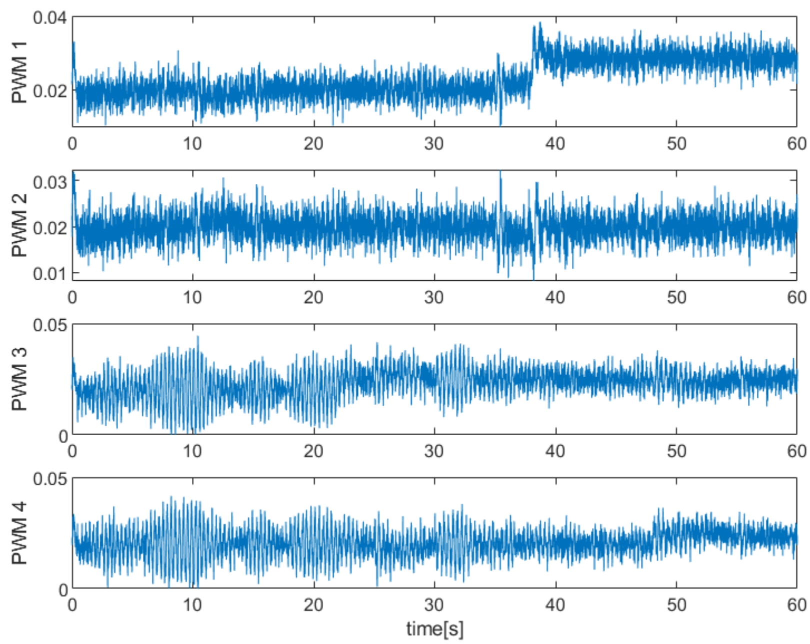

Figure 7.33: PWM Signals for VbDOB on PD Controller with Actuator LOE

Estimated lumped disturbance is directly added on calculated control input which is why increased oscillations on PWM signals are seen. Also, other than first motor; fault effects are minimized on PWM signals due to acted disturbance observer beforehand. Average control effort is reduced slightly comparing to non-DOb case which is calculated to be 0.08654.

VbDOB on LQG Controller with Actuator LOE

VbDOB does not change LQG response significantly as in the case of PD. It minimizes error even more on positional states but change in angular dynamics is negligibly small if we compare it to Table 7.4. LQG response while acting a VbDOB is as follows;

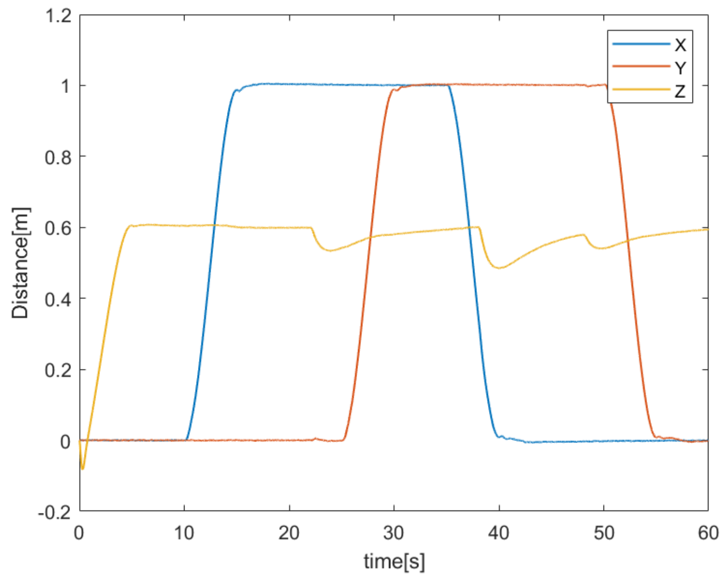

Figure 7.34: VbDOB on LQG Controller Trajectory Following with Actuator LOE

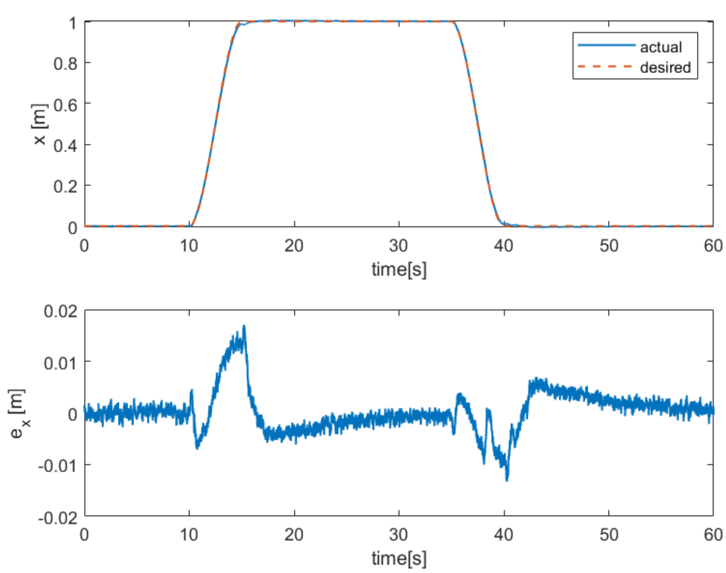

Figure 7.35: X position of the quadrotor (top), position error (bottom)

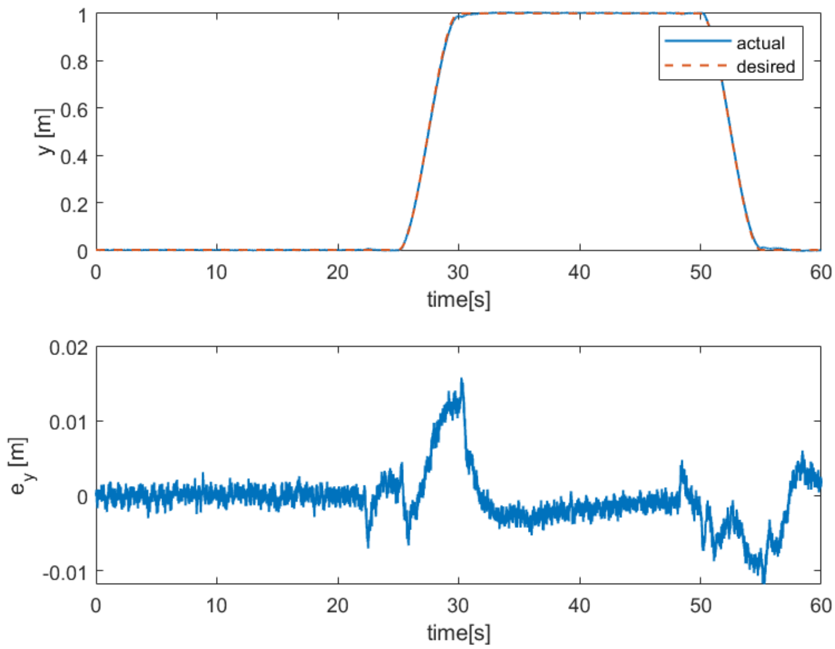

Figure 7.36: Y position of the quadrotor (top), position error (bottom)

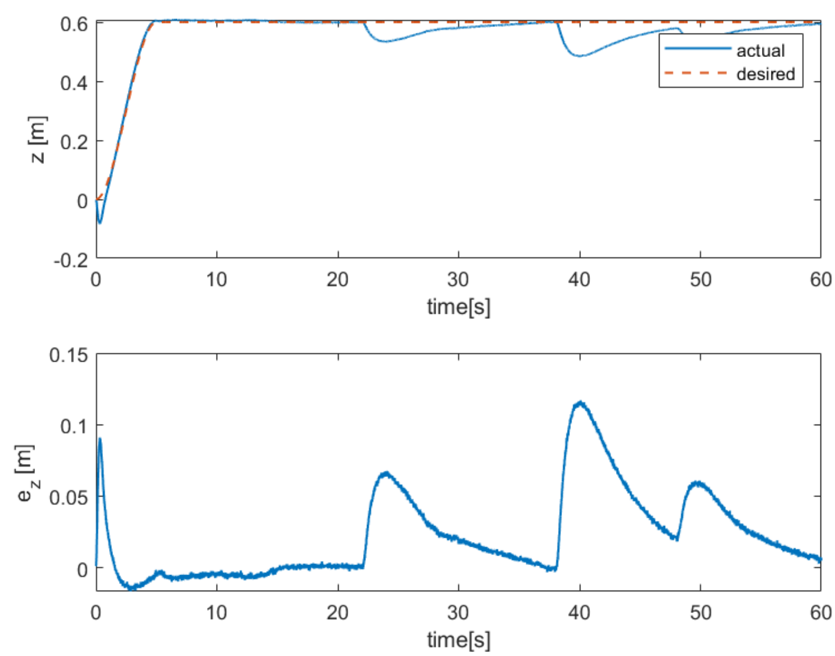

Figure 7.37: Z position of the quadrotor (top), position error (bottom)

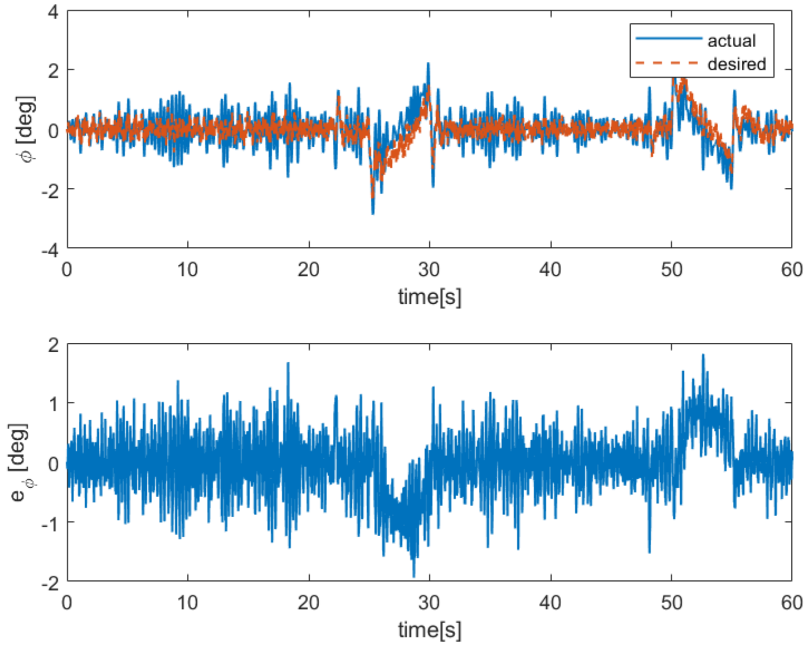

Figure 7.38: Roll angle of the quadrotor (top), tracking error (bottom)

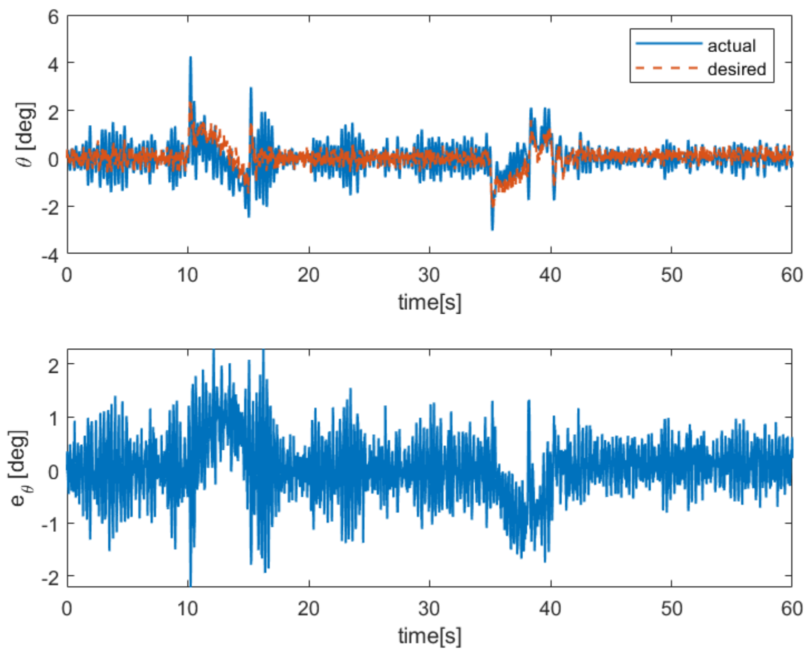

Figure 7.39: Pitch angle of the quadrotor (top), tracking error (bottom)

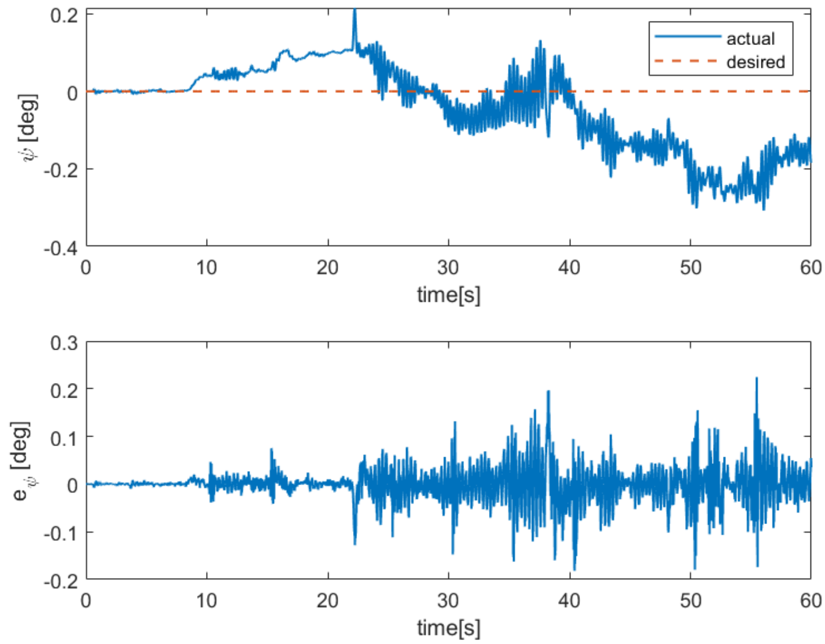

Figure 7.40: Yaw angle of the quadrotor (top), tracking error (bottom)

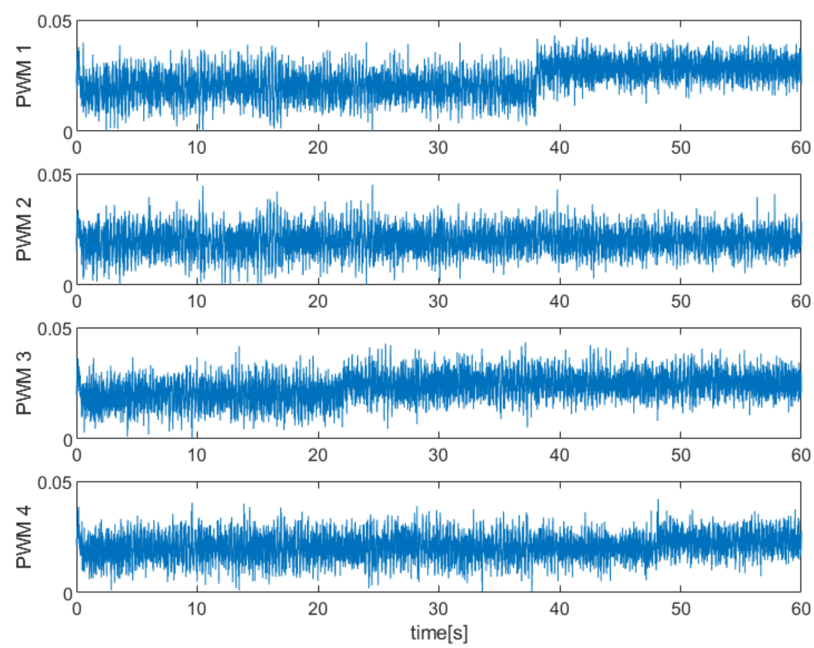

Figure 7.41: PWM Signals for VbDOB on LQG Controller with Actuator LOE

Table 7.7: Tracking Errors for VbDOB on LQG Controller with Actuator LOE

Error	RMS Errors	Max Errors
e_x	0.0039	0.0170
e_y	0.0036	0.0157
e_z	0.03771	0.1169
e_ϕ	0.5113	1.8184
e_θ	0.5804	2.2886
e_ψ	0.03465	0.2243

AbDOB on LQG Controller with Actuator LOE

An acceleration based velocity disturbance observer is tested on LQG controller to compare with velocity-based version. AbDOB uses a nominal plant with second order terms and acts on directly to Euler angles raher than Euler rates in the VbDOB case.

We have wanted to test this because due to VbDOB acting on Euler rates, high oscillation is observed under attitude dynamics. This is because improved trajectory response due to VbDOB requires higher demanding desired roll and pitch angles. One can see from Figures 7.38 and 7.39 tracking error is increased with these terms due to demanding position controller. In order to lift the burden from attitude dynamics, AbDOB is designed which improves navigational tracking by not extensively loading on attitude controller.

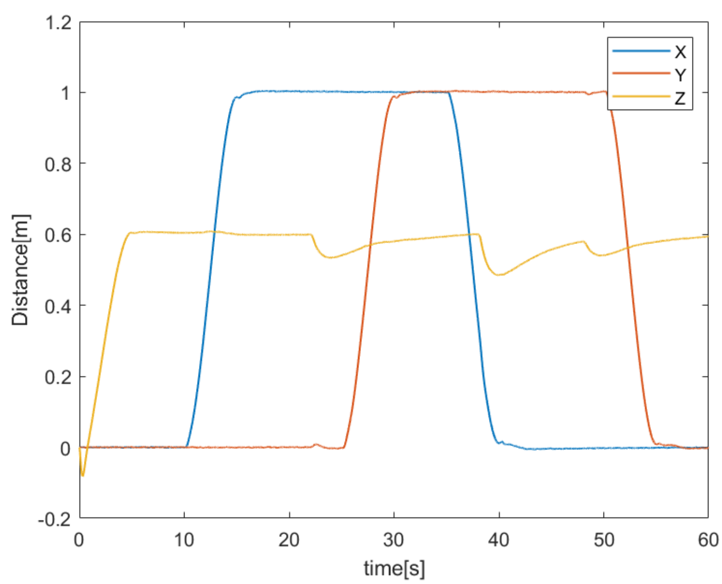

Figure 7.42: AbDOB on LQG Controller Trajectory Following with Actuator LOE

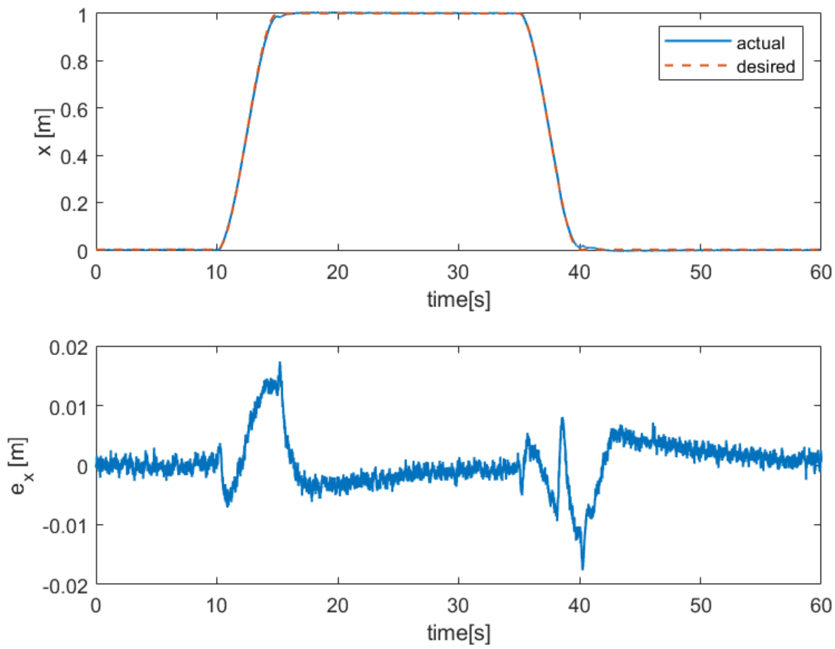

Figure 7.43: X position of the quadrotor (top), position error (bottom)

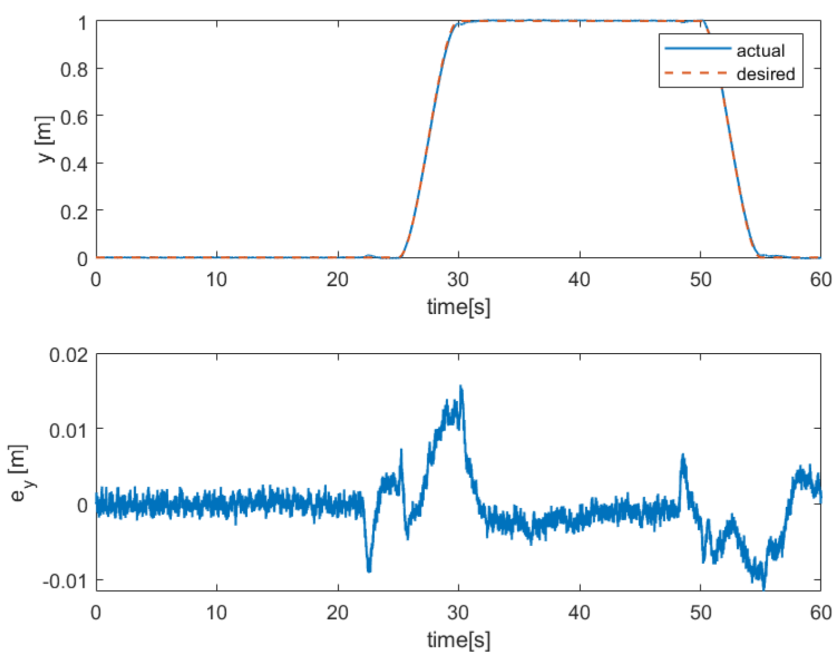

Figure 7.44: Y position of the quadrotor (top), position error (bottom)

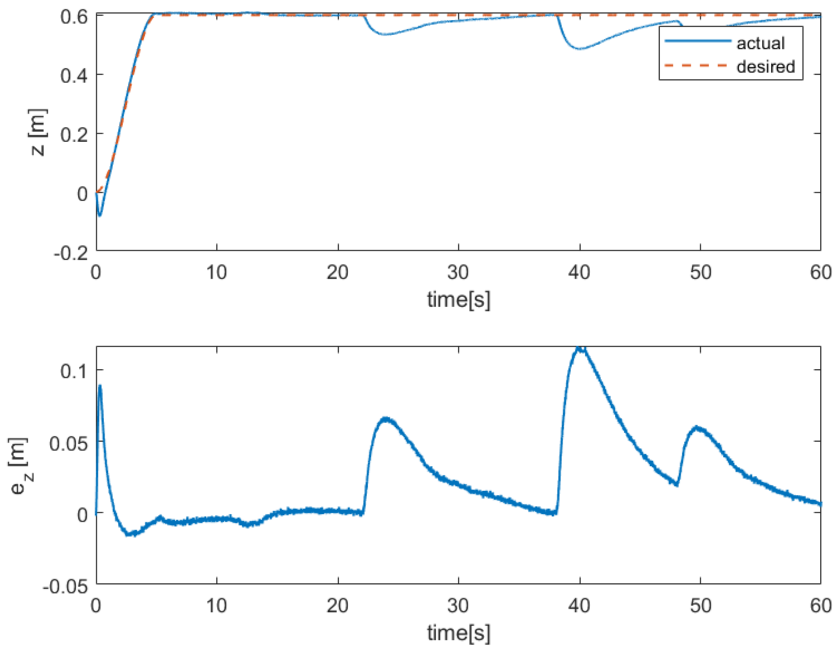

Figure 7.45: Z position of the quadrotor (top), position error (bottom)

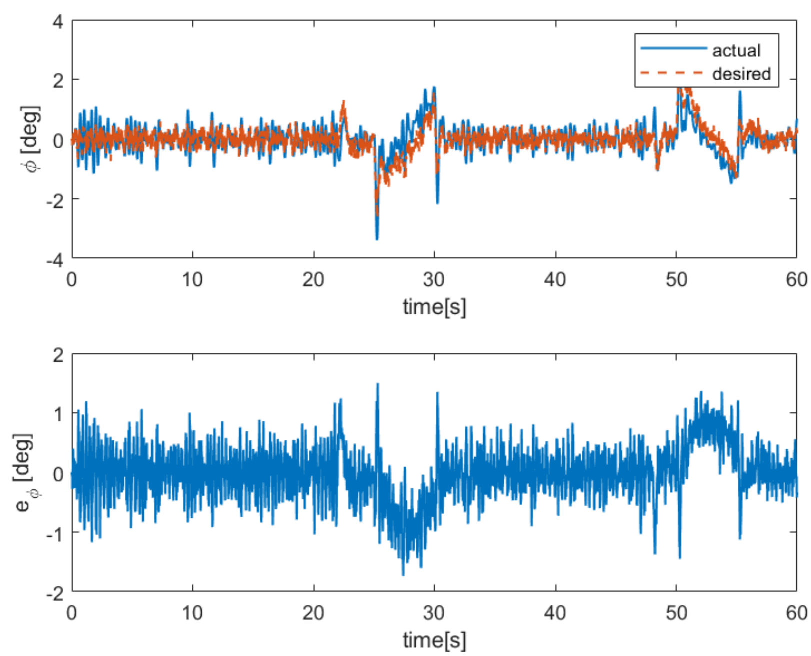

Figure 7.46: Roll angle of the quadrotor (top), tracking error (bottom)

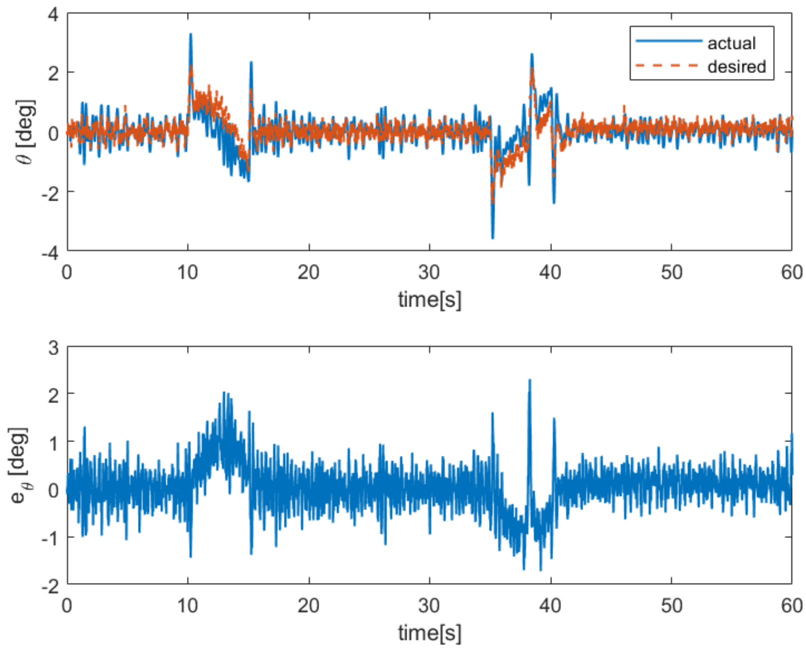

Figure 7.47: Pitch angle of the quadrotor (top), tracking error (bottom)

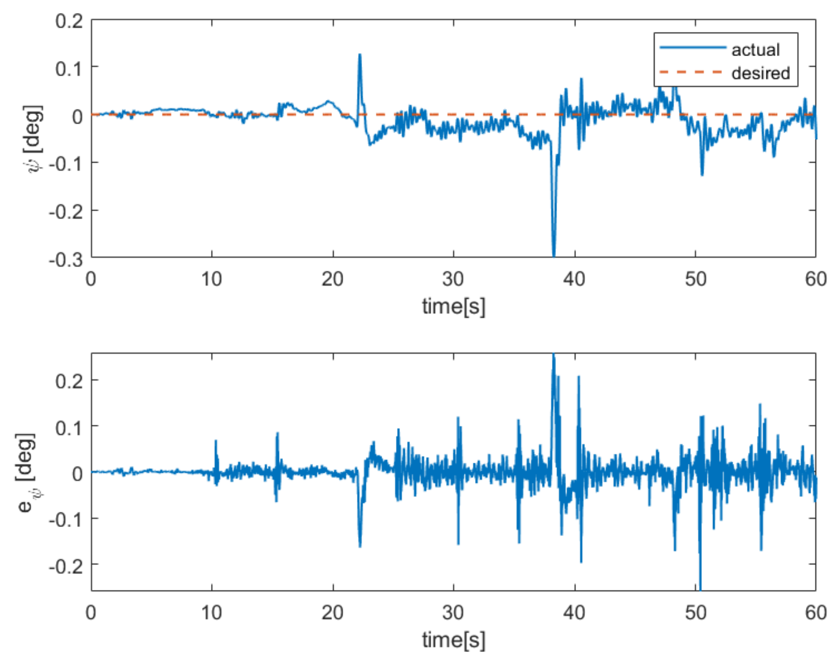

Figure 7.48: Yaw angle of the quadrotor (top), tracking error (bottom)

Inner controller's work load is eased simultaneously achieving very good trajectory following. This configuration gives the best results so far.

Table 7.8: Tracking Errors for AbDOB on LQG Controller with Actuator LOE

Error	RMS Errors	Max Errors
e_x	0.0040	0.01738
e_y	0.0037	0.01578
e_z	0.03777	0.1166
e_ϕ	0.4472	1.5005
e_θ	0.4655	2.3038
e_ψ	0.0302	0.2593

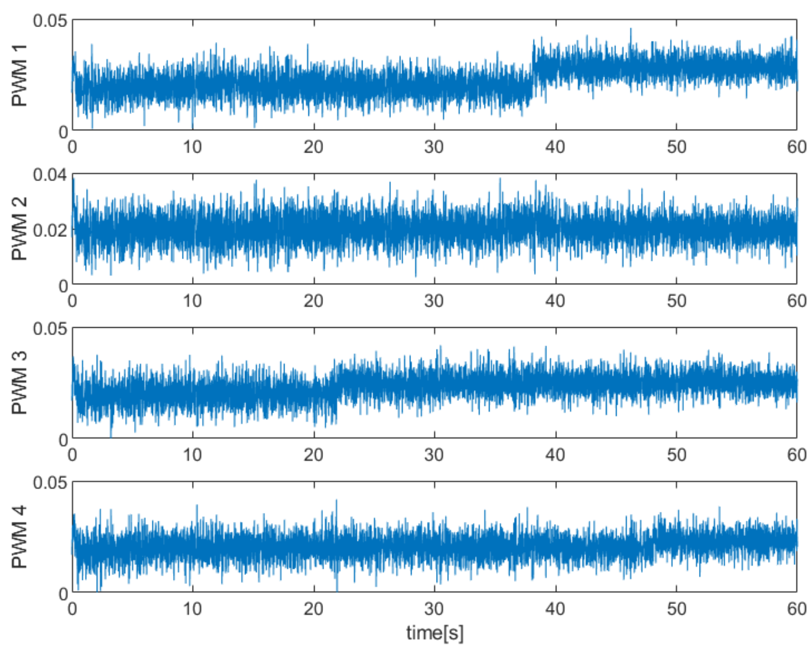

Figure 7.49: PWM Signals for AbDOB on LQG Controller with Actuator LOE

Average control effort is 0.08645 for this configuration.

7.4 State Observer

This section presents results for two state observer types. It is stated earlier that TSKF is an adaptive fault estimator which uses linear state space equations for state estimation. On the other hand EKF does not have a separate fault estimation unit but gives state estimations by directly using nonlinear equations.

Previously, all the results employed EKF for state feedback due to having slightly better estimation figures with respect to TSKF. This section presents results that employs TSKF as the state-feedback observer. Each result set has disturbance observer acting upon due to having improved reactions. Therefore, reader can compare correlating results with Section 7.3 that disturbance observer is activated at all cases.

TSKF - PD - AbDOB

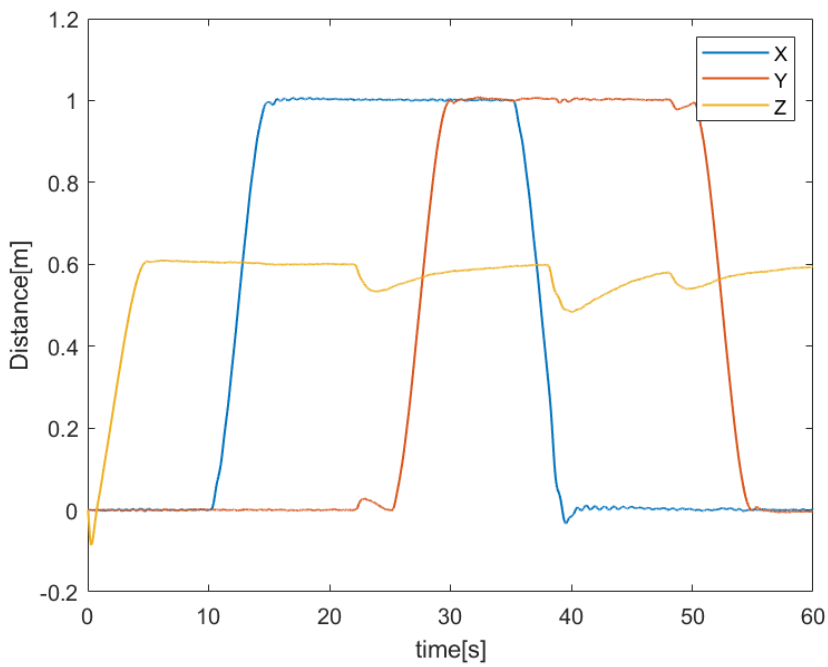

Figure 7.50: TKSF Feedback and AbDOB on PD Controller Trajectory Following with Actuator LOE

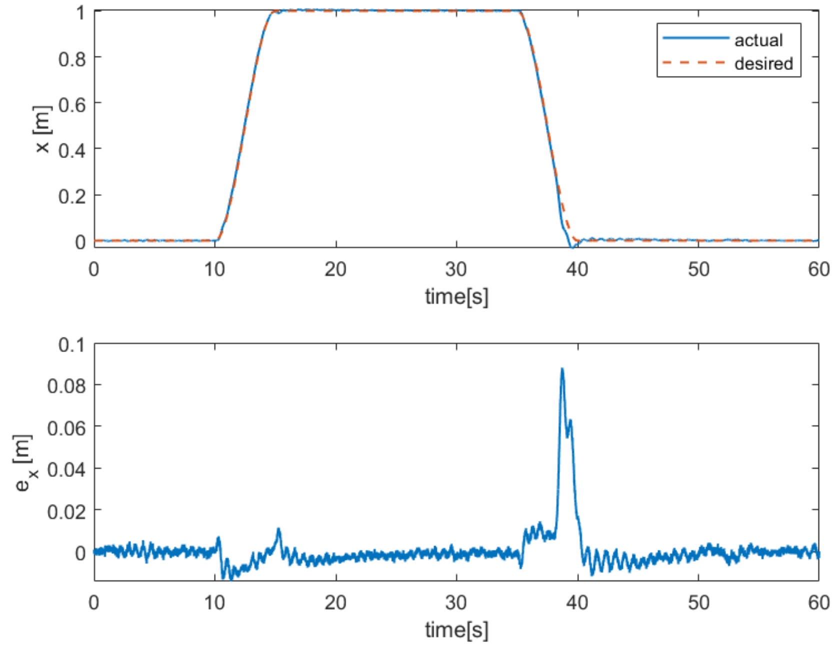

Figure 7.51: X position of the quadrotor (top), position error (bottom)

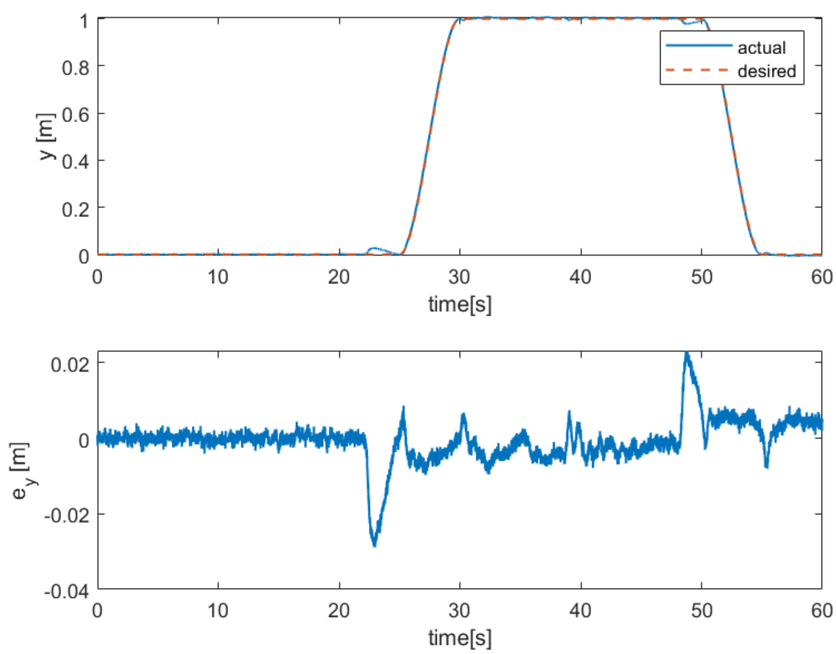

Figure 7.52: Y position of the quadrotor (top), position error (bottom)

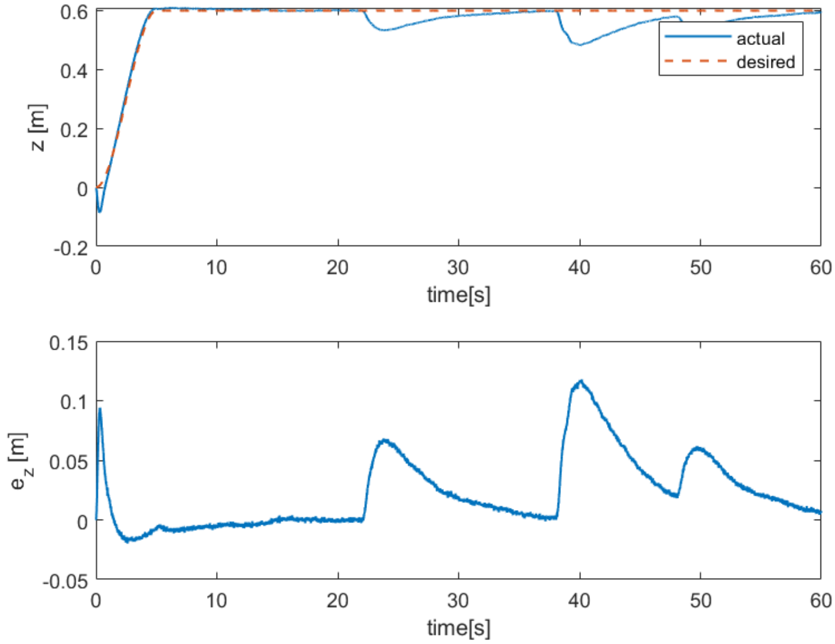

Figure 7.53: Z position of the quadrotor (top), position error (bottom)

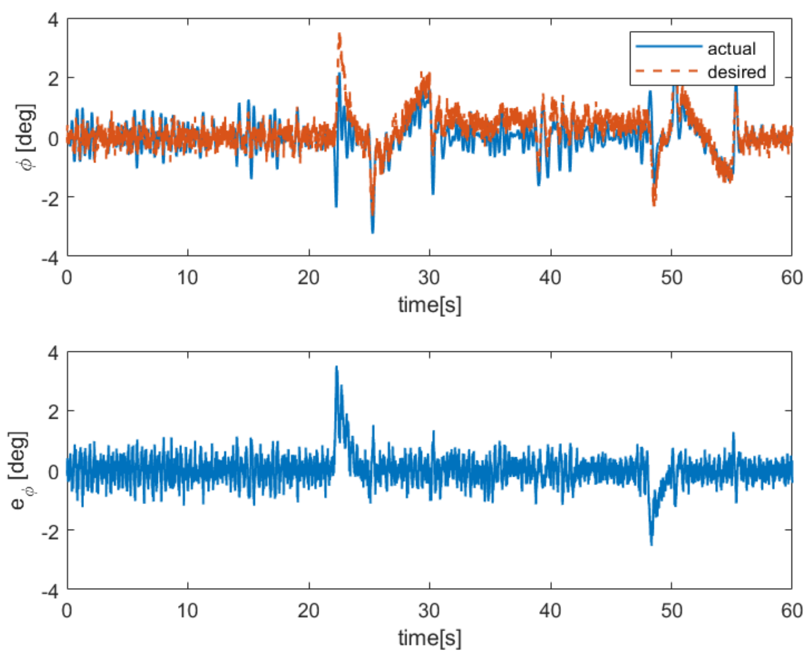

Figure 7.54: Roll angle of the quadrotor (top), tracking error (bottom)

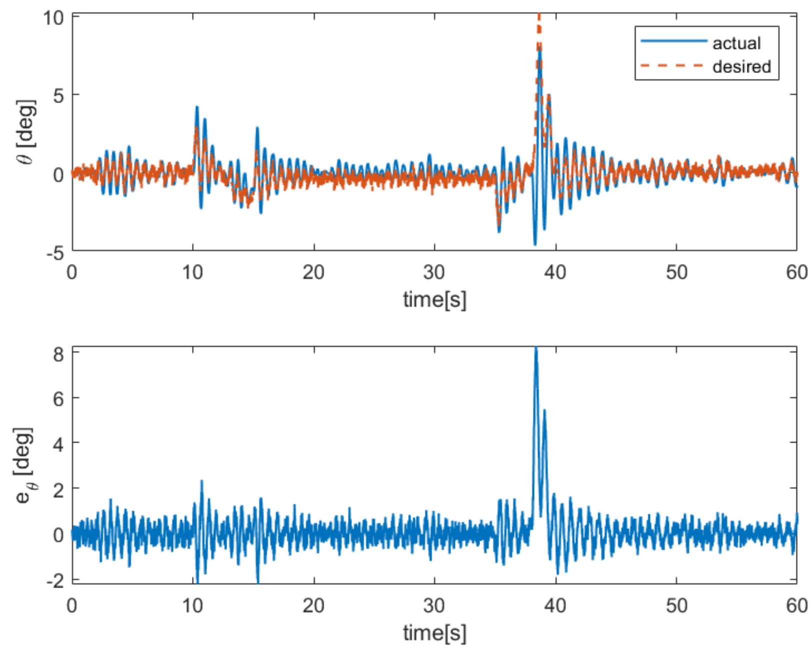

Figure 7.55: Pitch angle of the quadrotor (top), tracking error (bottom)

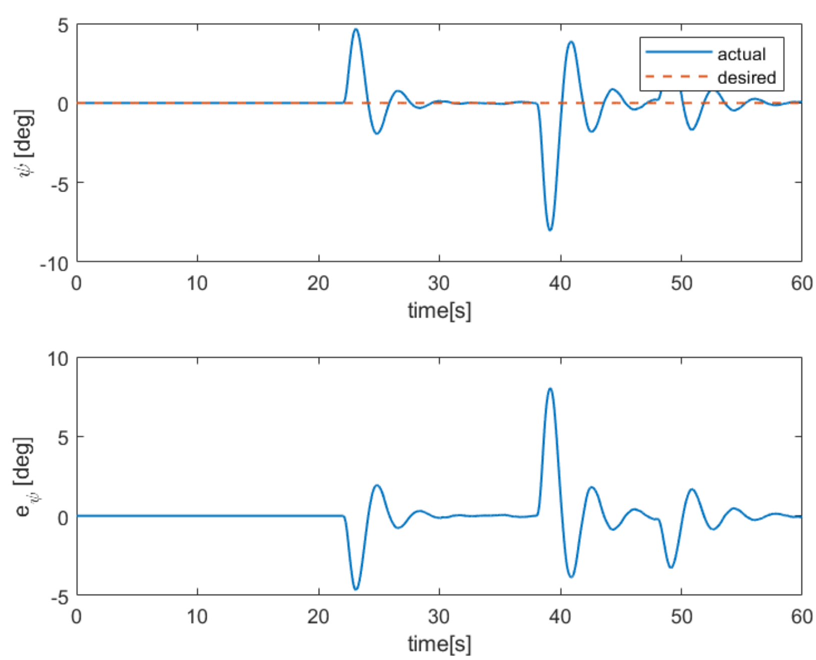

Figure 7.56: Yaw angle of the quadrotor (top), tracking error (bottom)

Even though taking state feedback from TSKF does not effect system dynamics highly when disturbance observer is acting on the system. A little oscillations are observed especially on X and Y states but overall performance is satisfactory. Also, due to used AbDOB rather than VbDOB for PD setup gave much better results again just like with the LQG case. So, it is safe to say that acceleration-based disturbance observer works better and much more efficiently if the position controller is slogging on.

Table 7.9: Tracking Errors for AbDOB on PD Controller with Actuator LOE and TSKF Feedback

Error	RMS Errors	Max Errors
e_x	0.01048	0.0880
e_y	0.0056	0.0232
e_z	0.03769	0.1178
e_ϕ	0.4923	3.5078
e_θ	0.8308	8.2729
e_ψ	1.3624	8.0259

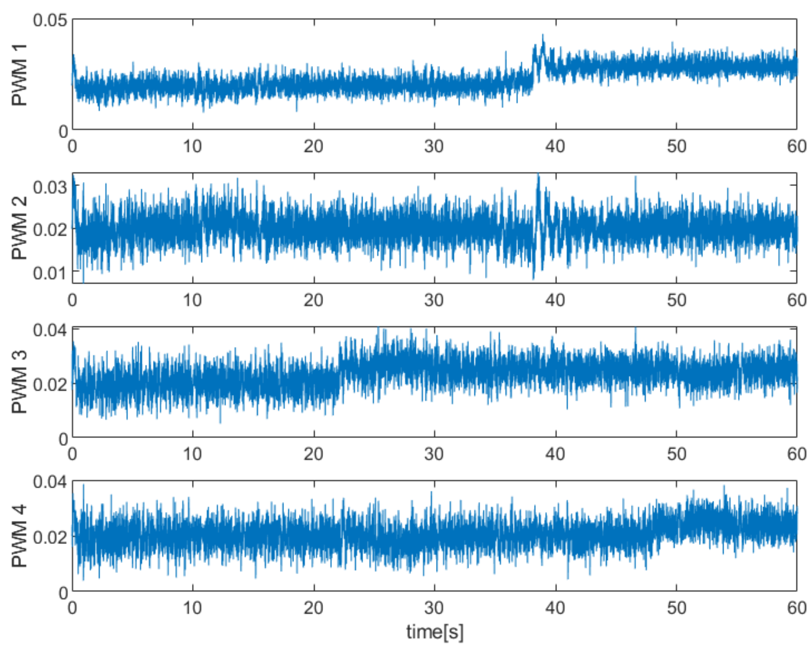

Figure 7.57: PWM Signals for AbDOB on PD Controller with Actuator LOE and TSKF Feedback

Average control effort is 0.08680 for this configuration.

TSKF - LQG - AbDOB

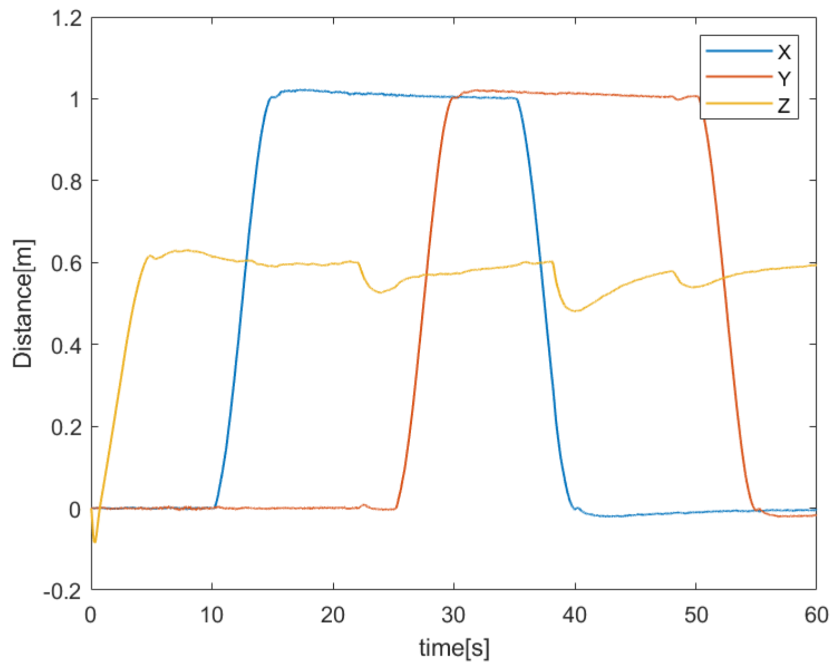

Figure 7.58: TKSF Feedback and AbDOB on LQG Controller Trajectory Following with Actuator LOE

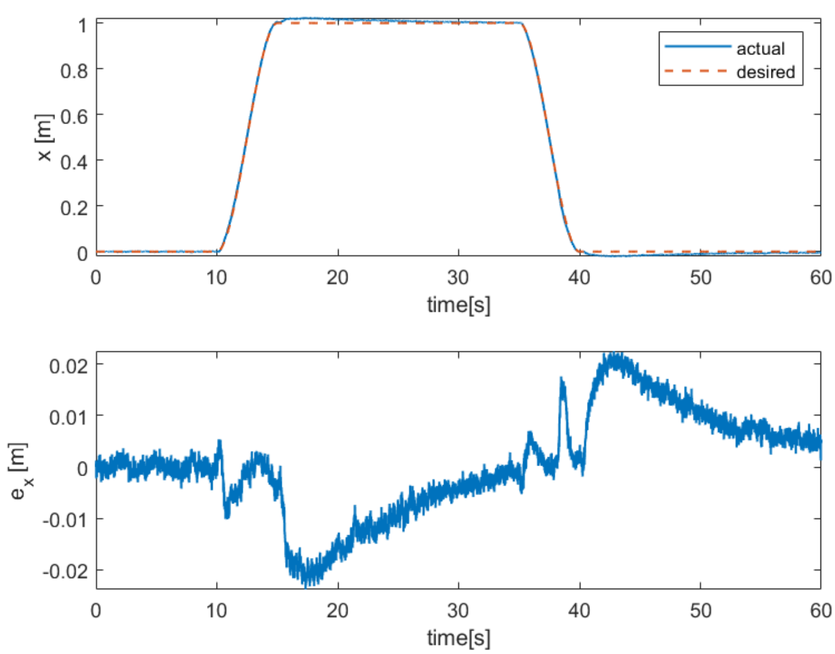

Figure 7.59: X position of the quadrotor (top), position error (bottom)

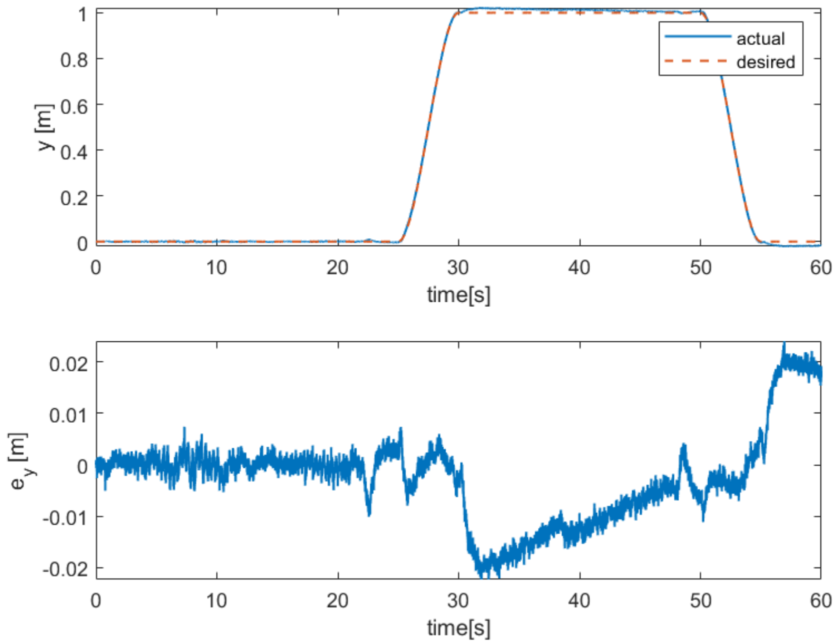

Figure 7.60: Y position of the quadrotor (top), position error (bottom)

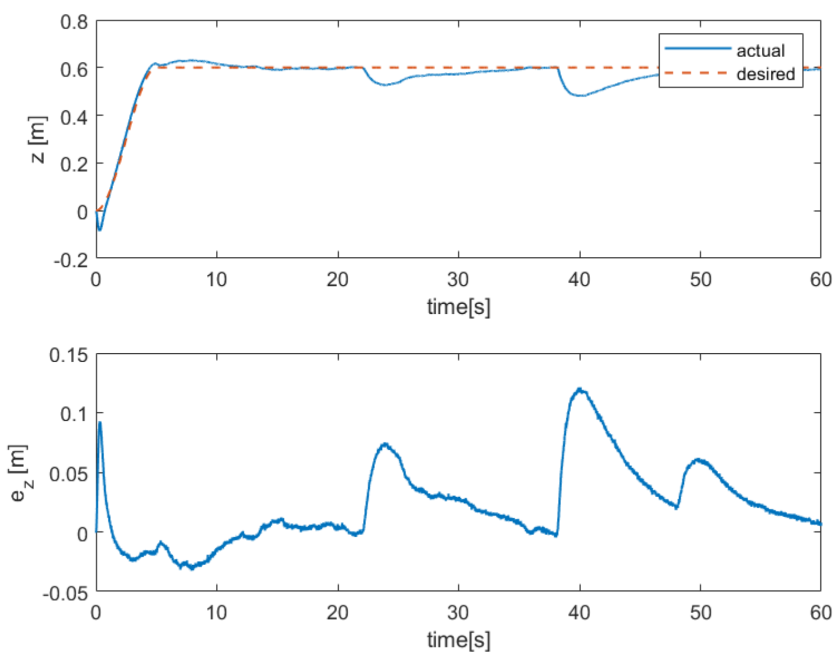

Figure 7.61: Z position of the quadrotor (top), position error (bottom)

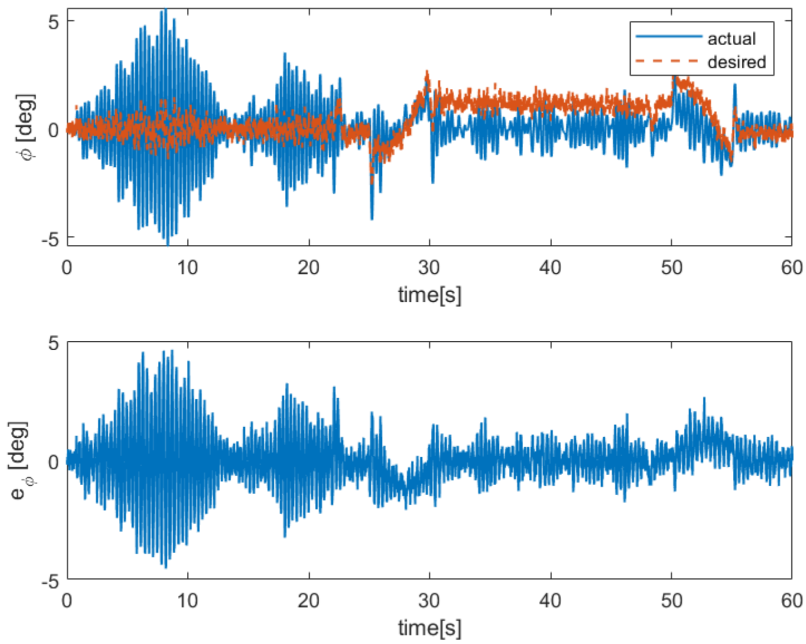

Figure 7.62: Roll angle of the quadrotor (top), tracking error (bottom)

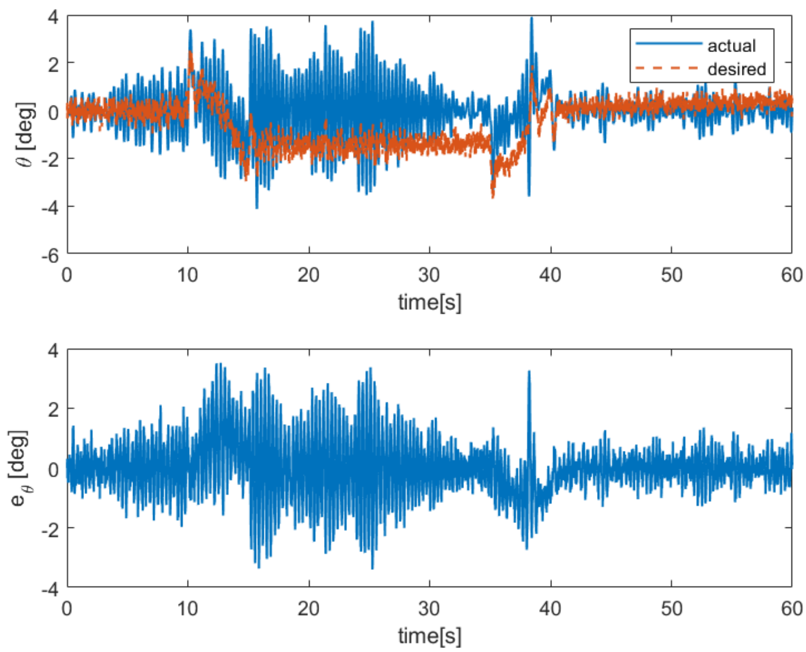

Figure 7.63: Pitch angle of the quadrotor (top), tracking error (bottom)

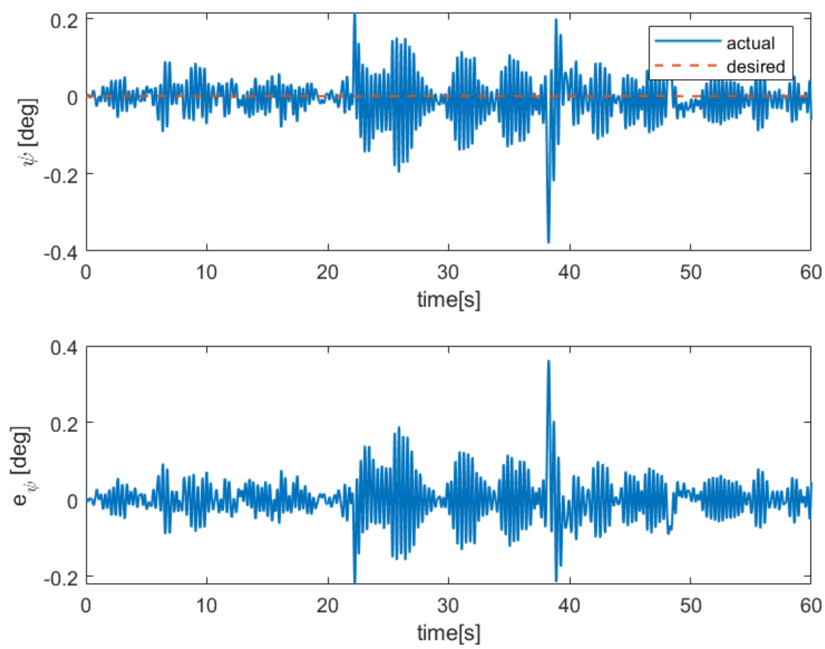

Figure 7.64: Yaw angle of the quadrotor (top), tracking error (bottom)

Surprisingly, TSKF feedback showed poorer performance on LQG controller comparing to PD controller. This is probably due to high error tracking requirement designed for LQG controller. LQG Q gain is selected conservative for best trajectory following however with poorer feedback response from TSKF, it prevents LQG to find an optimized strategy to smoothen control decision. In other words, LQG is trying to find the best solution with the upcoming feedback signal however the signal can't represent the actual dynamics as in the case of EKF feedback. This is why, TSKF feedback puts LQG controller into a chattering behaviour which decreases error tracking performance highly especially with attitude dynamics.

Table 7.10: Tracking Errors for AbDOB on LQG Controller with Actuator LOE and TSKF Feedback

Error	RMS Errors	Max Errors
e_x	0.0098	0.022
e_y	0.009	0.02410
e_z	0.0399	0.1212
e_ϕ	1.1494	4.6378
e_θ	0.9862	3.5211
e_ψ	0.0518	0.3628

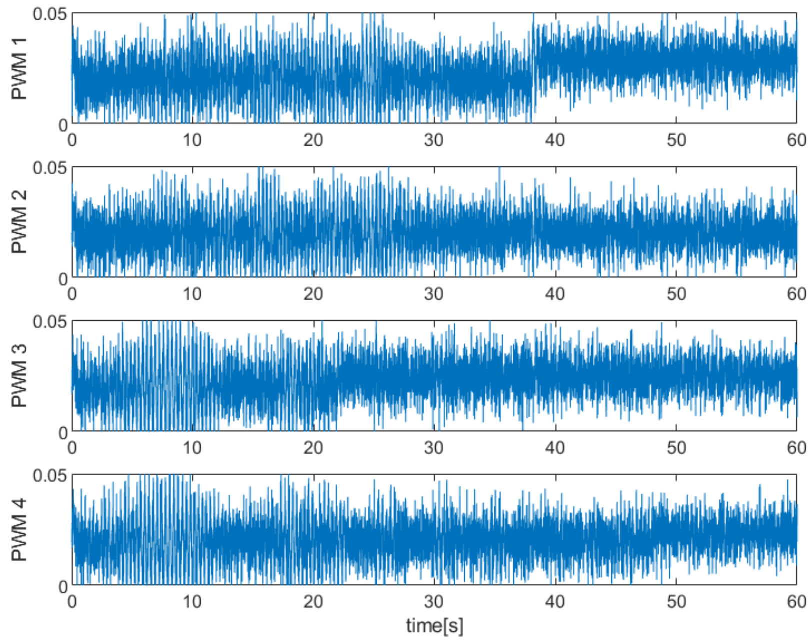

Figure 7.65: PWM Signals for AbDOB on LQG Controller with Actuator LOE and TSKF Feedback

One very important fact to note here is; normally LQG controller shows better performance with every simulation scenario we have tested so far. However, the only exception is with this configuration where PD performance out performs LQG performance if feedback signal has higher uncertainty. Therefore, it is possible to say that an LQG controller architecture is much beneficial under FTC scheme as long as we design an accurate state feedback for it. Mean control effort is 0.08657 for this configuration.

7.5 Fault Types

Last part is focused on reaction to different fault types. Only EKF is used to prevent any uncertainty related to state estimation. Injected fault and failure types are as follows;

- **Structural Failure:** A battery pack weighing 300 grams breaks of from quadrotor. It is assumed that the battery pack is lost right from c.g at $t = 22s$ which only causes sudden weight loss and does not effect any inertia term.
- **Sensor Bias:** It is assumed that the barometer that is responsible for measuring altitude has a constant bias of 10^{-3} right from beginning to end of the simulation cycle. Therefore, only fault reflection will be expected to be observed on altitude.
- **Sensor Drift:** Additive to bias case, in this scenario initial bias of 10^{-3} is increased linearly and proportionally to simulation time.

Following cases are experimented as in Table 7.11 to also see reactions of controllers. All of the defined scenarios are ran with disturbance observers to capture the best performance as possible. This section aims to give reader an intuitive view about possible fault scenarios and their respective dynamic response. Therefore, rather than providing detailed analysis of error dynamics on each state; figures that present general system response are provided in order not to go out of scope.

Table 7.11: Fault Injection and Structural Failure Simulation Scenarios

Case Number	Controller	Fault Type
Case # 1	LQG	Structural Failure
Case # 2	PD	Structural Failure
Case # 3	LQG	Sensor Bias
Case # 4	PD	Sensor Bias
Case # 5	LQG	Sensor Drift
Case # 6	PD	Sensor Drift
Case # 7	LQG	Sensor Bias + Structural Failure
Case # 8	LQG	Sensor Drift + Structural Failure+ Actuator LOE

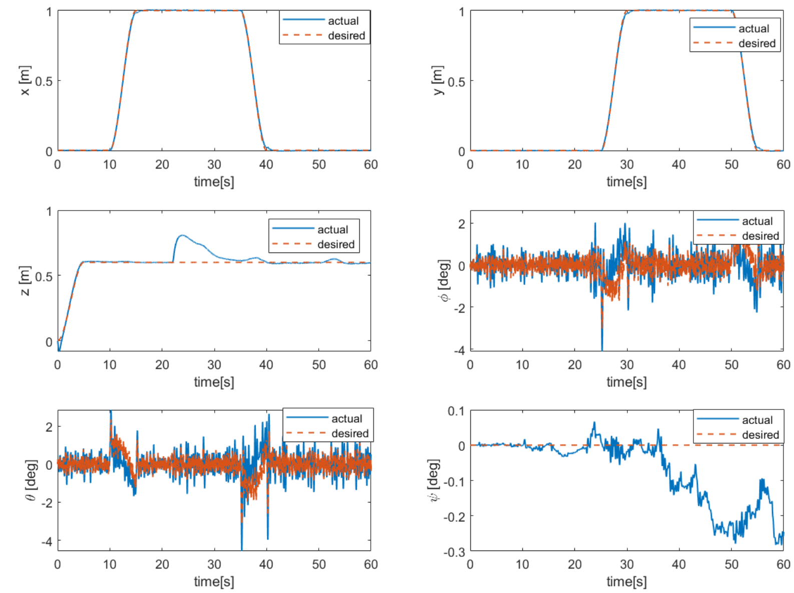

Figure 7.66: System Response to Case # 1

PD controller induces a chattering behaviour particularly on X and roll.

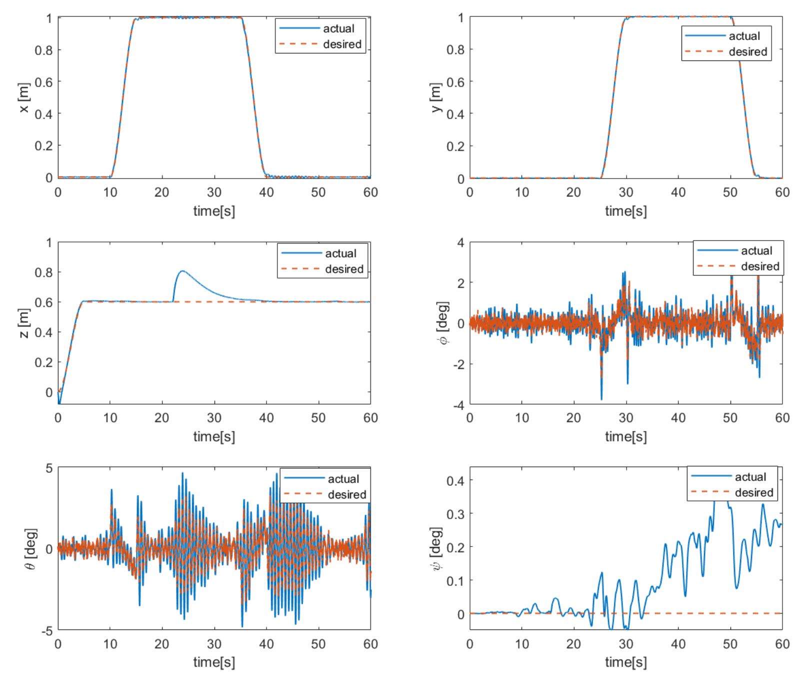

Figure 7.67: System Response to Case # 2

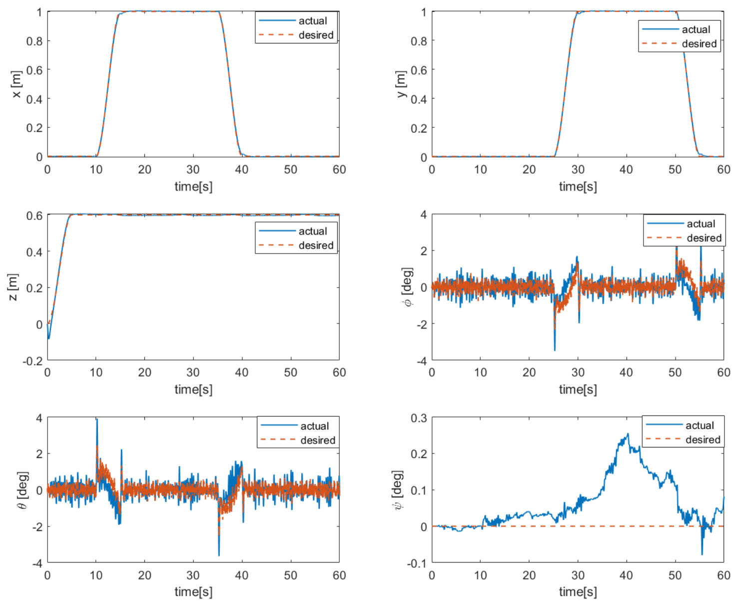

Figure 7.68: System Response to Case # 3

Sensor bias on barometer does not affect controller performance. Both systems are able to compensate for a fixed bias.

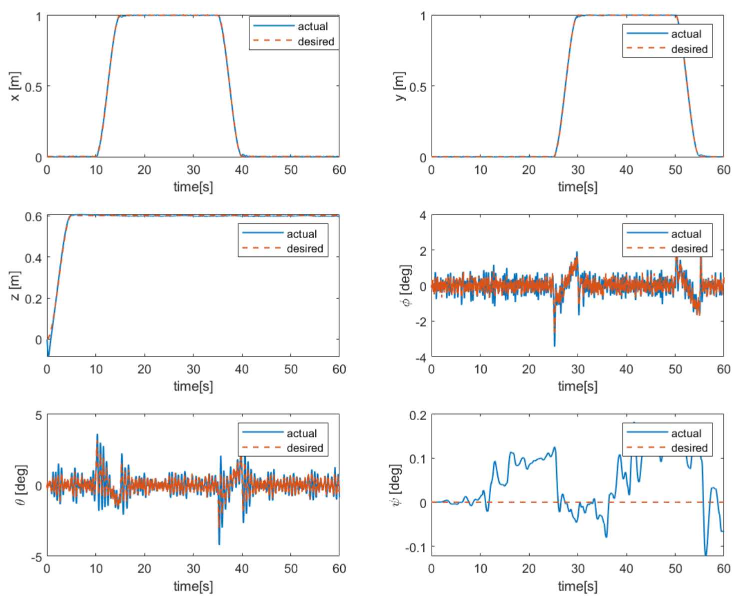

Figure 7.69: System Response to Case # 4

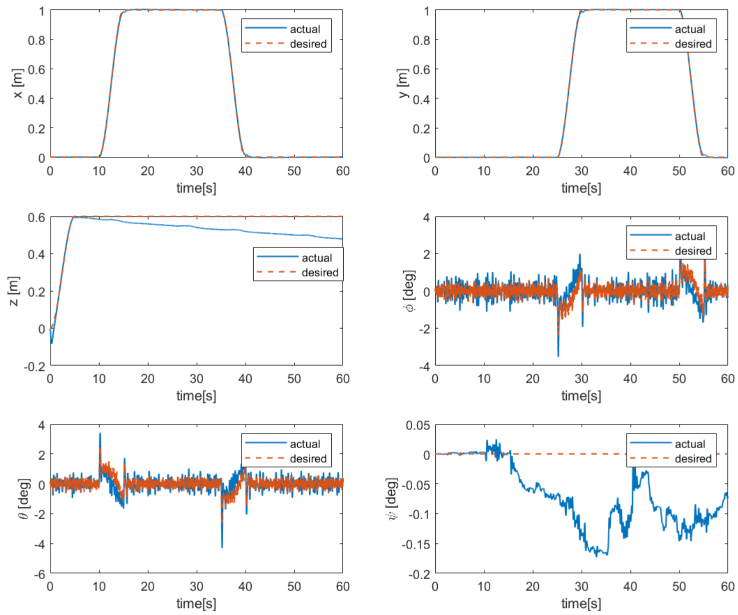

Figure 7.70: System Response to Case # 5

Both controllers are effected with barometer sensor drift. This is because drift is additively summed throughout simulation time which is why error on Z becomes harder to compensate as time is passed.

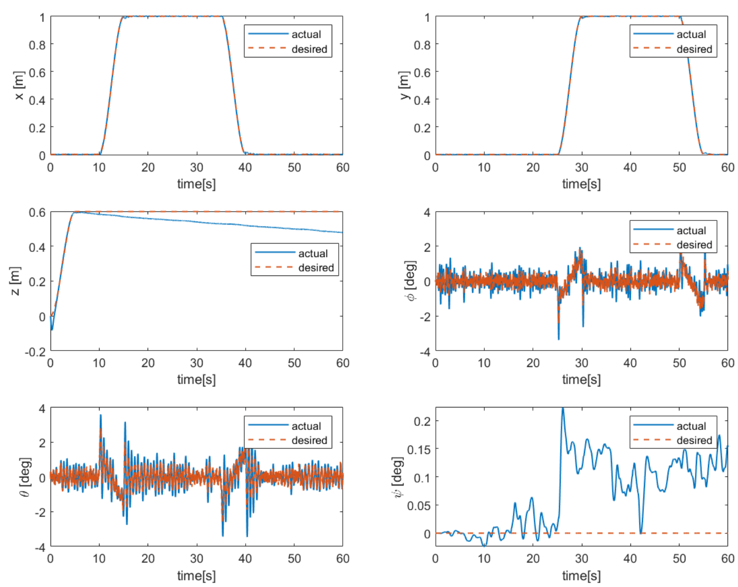

Figure 7.71: System Response to Case # 6

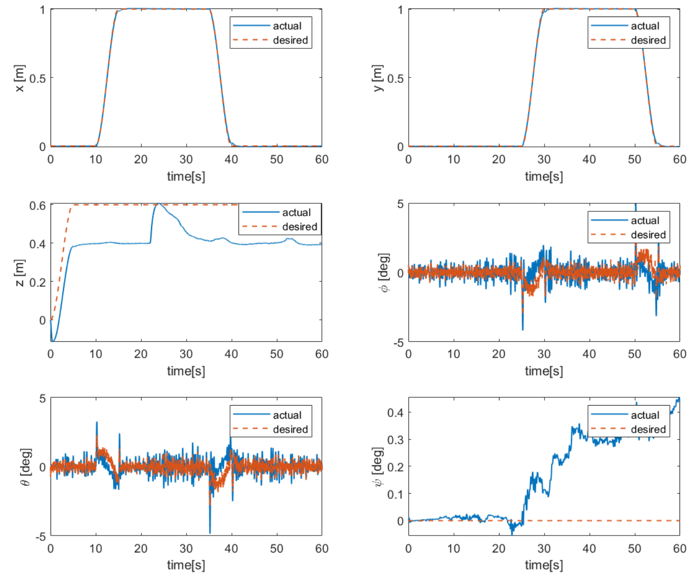

Figure 7.72: System Response to Case # 7

With scenario #7 bias on barometer is increased to 10^{-1} to visualize any changes. In before cases this bias could be compensated but with this scenario, effects of sensor bias and structural failure are highly visible. With increased bias, quadrotor constantly thinks its on the right altitude even though its not.

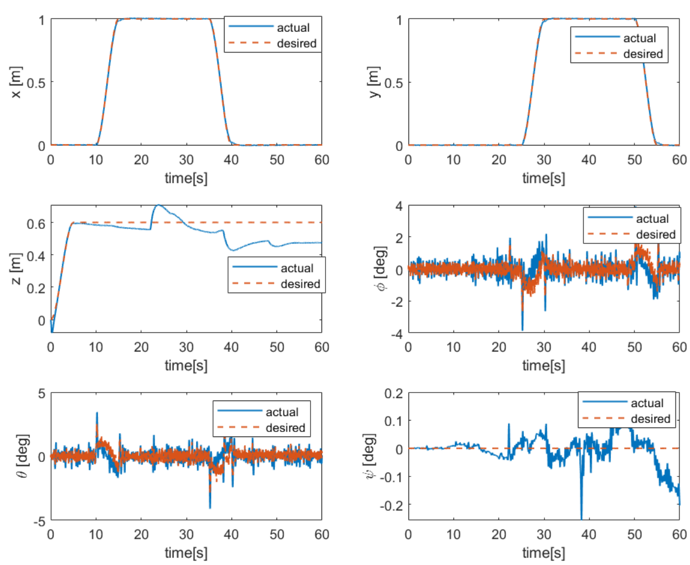

Figure 7.73: System Response to Case # 8

Case #8 is designed to be the most challenging scenerio which includes every type of fault and failure on top of the system. Our designed LQG controller gives exceptional results that can even recover extreme cases as such. To make a general commentary about this section, controller efforts for all the cases in Table 7.11 are given in Table 7.12.

Table 7.12: Fault Injection and Structural Failure Controller Efforts

Case Number	Mean Control Effort
Case # 1 (LQG and Structural Failure)	0.06911
Case # 2 (PD and Structural Failure)	0.06954
Case # 3 (LQG and Sensor Bias)	0.07947
Case # 4 (PD and Sensor Bias)	0.07955
Case # 5 (LQG and Sensor Drift)	0.07947
Case # 6 (PD and Sensor Drift)	0.07955
Case # 7 (LQG and Sensor Bias + Structural Failure)	0.06922
Case # 8 (LQG and Sensor Drift + Structural Failure+ Actuator LOE)	0.07485

One can immediately see by looking at Table 7.12 that sensor fault scenarios use the highest control effort. Also at first sight, decrease in control effort under structural failure would seem nonsensical because how can controller effort is reduced while requiring more of the controller under FTC scheme. This is due to reduced weight after a structural failure where motor voltages are decreased with less weight to carry. Also, LQG effort is always less than PD effort due to optimal control background.

Chapter 8

Conclusions and Future Work

In this thesis, we have presented a fault tolerant controller architecture via employing and comparing types of Kalman filters for state estimation and disturbance observers for disturbance estimation. Overall, a comprehensive result set is produced by interchanging between 2 controllers, 2 state estimators and 2 disturbance observers.

Our fault tolerant controller configuration which consists of both PID and LQG options have performed satisfactorily at a wide robustness envelope. It should be noted that LQG controller performed much more efficiently when compared to PD control in terms of both minimal control effort and maximum tracking error. However, PD feedback's ease of configurability has pushed itself forward from time to time, especially under finding instant solutions to problems.

Due to having very high gains for attitude response in order to stabilize the quadrotor under actuator fault, even small increments in fault would have resulted as a failure of control at first. By adding feed-forward terms of double-derivative trajectory references, we were able to reduce controller gains which directly increased robustness for more eager fault injection methods. Also, adding disturbance observer on PD control output was highly beneficial for enabling much better tracking error and better trajectory following. On the other hand, LQG is highly optimal as itself; does not need too much intervention. Because LQG structure is simply based on optimizing Q and R matrices where loop shaping is done by the controller itself to ensure stability which is why LQG does not need too much intervention unlike PID. Even disturbance observer did not make a lot of difference due to this self-maintained optimality for LQG. One fact that required intervention was due to inherent steady-state error problem of LQG which was solved via adding integral action. Therefore, it would be possible to say that even though LQG shows more promising performance at first, it does not mean PID can't. PID controller's enthusiasm on accepting new implementations and ease of applicability due to its relatively simpler architecture makes it still a very useful and reliable tool.

In this work, we have also benefited from diving deep into Kalman filters in order to have good state estimations which was also an advantage for observing effects of uncertainty over FTC. Knowing what the system is precisely doing is one of the most important aspects of designing a system especially when this system is subjected to high non-linearity. At first, a linear Two-Stage Kalman Filter is implemented which is a generally proposed filter for additive bias estimation from a residual output. This nested structure of TSKF comes very handy for preventing disadvantages of a linear filter because effects of fault can be embedded to state estimations anyway.

Every simulation scenario was able to operate with TSKF state feedback but we wanted to see possible effects that may come from uncertainty due to state estimations, which is why we have implemented an Extended Kalman Filter as well. Running two filters side-by-side showed that uncertainty can be a major problem even if we are using LQG controller such as in the case of Figure 7.65 where higher uncertainty due to TSKF might put extra burden on the controller in order to tolerate. Therefore, importance of accuracy of a state estimation has really justified itself via this comparison.

Furthermore, robustness of our controllers let us to experiment on various new implementations like introducing a disturbance observer to fault tolerant control scheme. This study has turned out to be very effective especially for PD control on attitude dynamics as discussed above. Disturbance observer is very handy for improving trajectory tracking as well. One other huge benefit of DOB is its feasibility for compensating a disturbance with an estimated anti-disturbance. This is why its applicability is not only restricted to disturbance estimation with fault only but to generate a lumped disturbance to counter-act on the control input generally. So, rather than compensating solely disturbance due to fault; we could cover an entire envelope of disturbances including wind, noise and any other possible unmodelled dynamics.

Overall, proposed methods showed successful results on their own way where fault and failure recovery was achieved for all the presented cases in this work. For future work, active fault tolerant control strategies can be developed either by adaptive controllers or adaptive fault detection observers. Or combinations of these methods can be studied; such as estimating faults online while using these faults to actively change control laws etc. Running all the scenarios and their corresponding solutions on a real test-bed would be a good opportunity as a future work as well.

Bibliography

[1] Giordan, D., Adams, M.S., Aicardi, I. et al. The use of unmanned aerial vehicles (UAVs) for engineering geology applications. Bull Eng Geol Environ 79, 3437–3481 (2020). https://doi.org/10.1007/s10064-020-01766-2

[2] H. Shakhatreh et al., Unmanned Aerial Vehicles (UAVs): A Survey on Civil Applications and Key Research Challenges, in IEEE Access, vol. 7, pp. 48572-48634, 2019, doi: 10.1109/ACCESS.2019.2909530.

[3] S. Hayat, E. Yanmaz and R. Muzaffar, Survey on Unmanned Aerial Vehicle Networks for Civil Applications: A Communications Viewpoint, in IEEE Communications Surveys and Tutorials, vol. 18, no. 4, pp. 2624-2661, Fourthquarter 2016, doi: 10.1109/COMST.2016.2560343.

[4] A. Bachrach, R. He, and N. Roy. Autonomous flight in unknown indoor environments. International Journal of Micro Air Vehicles, 1(4):217–228, 2009.

[5] K. C. Saranya, V. P. S. Naidu, V. Singhal and B. M. Tanuja, Application of vision based techniques for UAV position estimation, 2016 International Conference on Research Advances in Integrated Navigation Systems (RAINS), 2016, pp. 1-5, doi: 10.1109/RAINS.2016.7764392.

[6] Eschmann, C., Kuo, C.-M., Kuo, C.-H., and Boller, C. (2012). Unmanned Aircraft Systems for Remote Building Inspection and Monitoring. In 6th European Workshop on Structural Health Monitoring - Th.2.B.1

[7] Dobson, R. J., Brooks, C., Roussi, C., and Colling, T. (2013). Developing an unpaved road assessment system for practical deployment with high-resolution optical data collection using a helicopter UAV. In 2013 International Conference on Unmanned Aircraft Systems (ICUAS) (pp. 235–243). IEEE. https://doi.org/10.1109/ICUAS.2013.6564695

[8] Ruggles, S., Clark, J., Franke, K. W., Wolfe, D., Reimschiissel, B., Martin, R. A., Hedengren, J. D. (2016). Comparison of SfM computer vision point clouds of a landslide derived from multiple small UAV platforms and sensors to a TLS-based model. Journal of Unmanned Vehicle Systems, 4(4), 246–265. https://doi.org/10.1139/juvs-2015-0043

[9] Wierzbicki, D. (2018). Application Of Unmanned Aerial Vehicles In Monitoring Of Communication Routes On Country Areas. In 18th International Scientific Conference Engineering for Rural Development. Jelgava, Latvia. https://doi.org/10.22616/ERDev2018.17.N199

[10] Wefelscheid, C., Hänsch, R., and Hellwich, O. (2011). Three-Dimensional Building Reconstruction Using Images Obtained by Unmanned Aerial Vehicles. In International Conference on Unmanned Aerial Vehicle in Geomatics (UAV-g) (Vol. XXXVIII-1/, pp. 183–188). https://doi.org/10.5194/isprsarchives-XXXVIII1-C22-183-2011

[11] Hubbard, B., Wang, H., Leasure, M., Ropp, T., Lofton, T., Hubbard, S., and Lin, S. (2015). Feasibility Study of UAV use for RFID Material Tracking on Construction Sites. In 51st ASC Annual International Conference Proceedings. Retrieved from http://ascpro.ascweb.org/chair/paper/CPRT367002015.pdf

[12] P. Doherty and P. Rudol. A UAV search and rescue scenario with human body detection and geolocalization. In Mehmet Orgun and John Thornton, editors, AI 2007: Advances in Artificial Intelligence, volume 4830 of Lecture Notes in Computer Science, pages 1– 13. Springer Berlin / Heidelberg, 2007. ISBN 978-3- 540-76926-2.

[13] Amin, K.M. Hasan, A Review of Fault Tolerant Control Systems: Advancements and Applications, Measurement (2019)

[14] Nguyen, N.P.; Hong, S.K. Fault Diagnosis and Fault-Tolerant Control Scheme for Quadcopter UAVs with a Total Loss of Actuator. Energies 2019, 12, 1139. https://doi.org/10.3390/en12061139

[15] Gabriel Hoffmann, Haomiao Huang, Steven Waslander and Claire Tomlin. Quadrotor Helicopter Flight Dynamics and Control: Theory and Experiment, AIAA 2007-6461. AIAA Guidance, Navigation and Control Conference and Exhibit. August 2007.

[16] Chamola, Vinay; Kotesh, Pavan; Agarwal, Aayush; Naren, ; Gupta, Navneet; Guizani, Mohsen (2020). A Comprehensive Review of Unmanned Aerial Vehicle Attacks and Neutralization Techniques. Ad Hoc Networks, 102324–. doi:10.1016/j.adhoc.2020.102324

[17] https://en.wikipedia.org/wiki/Black Hornet Nano. Accessed: 24 June 2021.

[18] https://www.dji.com/mg-1. Accessed: 24 June 2021.

[19] Turkish Aerospace Industry (TAI) ANKA. https://www.tai.com.tr/ fotograf-galerisi/anka, 2021. Accessed: 24 June 2021.

[20] http://www.civi-uavs.com/images/EBrochures/Products/Optimus.pdf. Accessed: 24 June 2021.

[21] https://www.jouav.com/flightSystem/cw-100.html. Accessed: 24 June 2021.

[22] https://en.wikipedia.org/wiki/Lockheed D-21. Accessed: 24 June 2021.

[23] https://www.insitu.com/products/scaneagle. Accessed: 24 June 2021.

[24] https://umsskeldar.aero/v-200-skeldar/. Accessed: 24 June 2021.

[25] E. Cetinsoy, S. Dikyar, C. Hancer, K. Oner, E. Sirimoglu, M. Unel, and M. Aksit, Design and construction of a novel quad tilt-wing UAV, Mechatronics, vol. 22, no. 6, pp. 723-745, 2012.

[26] Amoozgar, M.H., Chamseddine, A., Zhang, Y. Experimental Test of a Two-Stage Kalman Filter for Actuator Fault Detection and Diagnosis of an Unmanned Quadrotor Helicopter. J Intell Robot Syst 70, 107–117 (2013). https://doi.org/10.1007/s10846-012-9757-7

[27] E.N Johnson, and D.P Schrage, System integration and operation of a research unmanned aerial vehicle, Journal of Aerospace Computing, Information, and Communication, Vol. 1, January 2004.

[28] Sadeghzadeh, Iman (2015) Fault Tolerant Flight Control of Unmanned Aerial Vehicles. PhD thesis, Concordia University.

[29] D.B. Jourdan, M.D. Piedmonte, V. Gavrilets, and D.W. Vos, Enhancing UAV survivability through damage tolerant control, AIAA Guidance, Navigation, and Control, Conference, August 2-5, 2010, Toronto, Ontario, Canada.

[30] R. Isermann, Fault-diagnosis systems, an introduction from fault detection to fault tolerance, Springer-Verlag, Berlin Heidelberg, 2006.

[31] J.J. Guillaume Ducard, Fault-tolerant flight control and guidance systems: practical methods for unmanned aerial vehicles, Springer, 2009.

[32] A. Fekih, Fault diagnosis and Fault Tolerant Control design for aerospace systems: A bibliographical review, 2014 American Control Conference, 2014, pp. 1286-1291, doi: 10.1109/ACC.2014.6859271.

[33] A. Younghwan. A Design Of Fault Tolerant Flight Control Systems For Sensor And Actuator Failures Using On-Line Learning Neural Networks. PhD thesis, West Virginia University, 1998.

[34] Murat Bronz, Elgiz Baskaya, Daniel Delahaye, Stephane Puechmorel. Real-time Fault Detection on Small Fixed-Wing UAVs using Machine Learning. DASC 2020 AIAA,IEEE 39th Digital Avionics Systems Conference, Oct 2020, San Antonio, United States.

[35] Arasanipalai, Rohitkumar, Agrawal, Aakriti, Ghose, Debasish. (2020). Mid-flight Propeller Failure Detection and Control of Propeller-deficient Quadcopter using Reinforcement Learning.

[36] P. A. Samara, G. N. Fouskitakis, J. S. Sakellariou, and S. D. Fassois. A Statistical Method for the Detection of Sensor Abrupt Faults in Aircraft Control Systems. IEEE Transactions on Control Systems Technology, 16(4):789–798, July 2008.

[37] Nguyen, N.P.; Hong, S.K. Sliding mode Thau observer for actuator fault diagnosis of quadcopter UAVs. Appl. Sci. 2018, 8, 1893.

[38] Zhang, Zhi-Hui., Li, Shujiang Li., Yan, Hua, Fan, Quan-Yong. (2019). Sliding mode switching observer-based actuator fault detection and isolation for a class of uncertain systems. Nonlinear Analysis: Hybrid Systems. 33. 322-335. 10.1016/j.nahs.2019.04.001.

[39] Hongjun, M, Liu, Y, Li, T, et al. Nonlinear high-gain observer-based diagnosis and compensation for actuators and sensors faults in a quadrotor unmanned aerial vehicle. IEEE Trans Ind Inform 2019; 15(1): 550–562.

[40] C. Hajiyev Et Al. , Two-Stage Kalman Filter for Fault Tolerant Estimation of Wind Speed and UAV Flight Parameters, Measurement Science Review , vol.20, no.1, pp.35-42, 2020

[41] H.Başak, E.Prempain, Switched fault tolerant control for a quadrotor UAV, IFAC-PapersOnLine, Volume 50, Issue 1,2017,Pages 10363-10368,ISSN 2405-8963, https://doi.org/10.1016/j.ifacol.2017.08.1686.

[42] A. Freddi, S. Longhi and A. Monteriu,. Actuator fault detection system for a mini-quadrotor. 2010 IEEE International Symposium on Industrial Electronics, 2010, pp. 2055-2060, doi: 10.1109/ISIE.2010.5637750.

[43] Wang, B.; Zhang, Y.M. Adaptive sliding mode fault-tolerant control for an unmanned aerial vehicle. Unmanned Syst. 2017, 5, 209–221.

[44] Nguyen, N.P.; Hong, S.K. Active Fault-Tolerant Control of a Quadcopter against Time-Varying Actuator Faults and Saturations Using Sliding Mode Backstepping Approach. Appl. Sci. 2019, 9, 4010.

[45] J. Tan, Y. Fan, P. Yan, C. Wang, and H. Feng, Sliding mode fault tolerant control for unmanned aerial vehicle with sensor and actuator faults, Sensors, vol. 19, no. 3, p. 643, 2019.

[46] Zhu, Zhiqiang, Cao, Songyin. (2019). Back-stepping sliding mode control method for quadrotor UAV with actuator failure. The Journal of Engineering. 2019. 10.1049/joe.2019.1084.

[47] Cen, Z., Noura, H., Susilo, T.B. et al. Robust Fault Diagnosis for Quadrotor UAVs Using Adaptive Thau Observer. J Intell Robot Syst 73, 573–588 (2014). https://doi.org/10.1007/s10846-013-9921-8

[48] M. Ranjbaran and K. Khorasani, Fault recovery of an under-actuated quadrotor Aerial Vehicle, 49th IEEE Conference on Decision and Control (CDC), 2010, pp. 4385-4392, doi: 10.1109/CDC.2010.5718140.

[49] Z. T. Dydek, A. M. Annaswamy and E. Lavretsky, Adaptive Control of Quadrotor UAVs: A Design Trade Study With Flight Evaluations, in IEEE Transactions on Control Systems Technology, vol. 21, no. 4, pp. 1400-1406, July 2013, doi: 10.1109/TCST.2012.2200104.

[50] Lavretsky, E. and Wise, K. A. (2013). Robust and adaptive control: with aerospace applications. Springer.

[51] D. Saussie, D. Nguyen, and L. Saydy, Quaternion-based robust fault-tolerant control of a quadrotor UAV, 2017 International Conference on Unmanned Aircraft Systems (ICUAS), 2017, pp. 1333-1342, doi: 10.1109/ICUAS.2017.7991516.

[52] Ha, C., Zuo, Z., Choi, F. B. and Lee, D. (2014). Passivity-based adaptive backstepping control of quadrotor-type UAVs. Robotics and Autonomous Systems, 62(9), 1305– 1315 Elsevier B.V.. doi:10.1016/j.robot.2014.03.019.

[53] Emran, B. J., Dias, J., Seneviratne, L. and Cai, G. (2015). Robust adaptive control design for quadcopter payload add and drop applications. Chinese Control Conference, CCC 2015–Septe: 3252–57. doi:10.1109/ChiCC.2015.7260141.

[54] Chamseddine, A., Theilliol, D., Zhang, Y.M., Join, C., and Rabbath, C.A. (2015), Active fault-tolerant control system design with trajectory re-planning against actuator faults and saturation: Application to a quadrotor unmanned aerial vehicle, Int. J. Adapt. Control Signal Process., 29, 1– 23, doi: 10.1002/acs.245

[55] Cowling, Ian, Whidborne, James, Cooke, Alastair. (2006). Optimal Trajectory Planning and LQR Control for a Quadrotor UAV.

[56] Benrezki, Riadh, Tadjine, M., Fouad, Yacef, Kermia, Omar. (2015). Passive fault tolerant control of quadrotor UAV using a nonlinear PID. 1285-1290. 10.1109/ROBIO.2015.7418948

[57] Merheb, Abdel-Razzak, Noura, Hassan, Bateman, François. (2015). Design of Passive Fault–Tolerant Controllers of a Quadrotor Based on Sliding Mode Theory. International Journal of Applied Mathematics and Computer Science. 25. 10.1515/amcs-2015-0042.

[58] K. D. Young, V. I. Utkin and U. Ozguner, A control engineer's guide to sliding mode control, in IEEE Transactions on Control Systems Technology, vol. 7, no. 3, pp. 328-342, May 1999, doi: 10.1109/87.761053.

[59] Xiong, J. J., and Zheng, E. H. (2014). Position and attitude tracking control for a quadrotor UAV. ISA Transactions, 53(3), 725–731 Elsevier. doi:10.1016/j.isatra.2014.01. 004

[60] Liu, H., Xi, J., Zhong, Y. (2014). Robust motion control of quadrotors. Journal of the Franklin Institute, 351(12), 5494–5510. doi:10.1016/j.jfranklin.2014.10.003.

[61] Peng, C., Bai, Y., Gong, X., Gao, Q., Zhao, C., Tian, Y. (2015). Modeling and robust backstepping sliding mode control with adaptive RBFNN for a novel. IEEE/CAA. Journal of Automatica Sinica, 2(1), 56–64. doi:10.1109/JAS.2015.7032906.

[62] Lee, Taeyoung. Robust Adaptive Attitude Tracking on SO(3) With an Application to a Quadrotor UAV. IEEE Transactions on Control Systems Technology 21 (2013): 1924-1930.

[63] J Gertler, Fault detection and isolation using parity relations, Control Engineering Practice, Volume 5, Issue 5, 1997, Pages 653-661.

[64] Daniel Ossmann, Optimization based tuning of fault detection and diagnosis systems for safety critical systems, IFAC Proceedings Volumes, Volume 47, Issue 3, 2014, Pages 8570-8575.

[65] Anna Hagenblad, Fredrik Gustafsson, Inger Klein, A comparison of two methods for stochastic fault detection: the parity space approach and principal components analysis, IFAC Proceedings Volumes, Volume 36, Issue 16, 2003, Pages 1053-1058.

[66] Zhang, X.; Zhao, Z.; Wang,Z.; Wang, X. Fault Detection and Identification Method for Quadcopter Based on Airframe Vibration Signals. Sensors 2021, 21, 581. https://doi.org/10.3390/s21020581

[67] Guo J, Qi J, Wu C. Robust fault diagnosis and fault-tolerant control for nonlinear quadrotor unmanned aerial vehicle system with unknown actuator faults. International Journal of Advanced Robotic Systems. March 2021. doi:10.1177/17298814211002734

[68] C. S. Jing and D. Pebrianti, Fault detection and identification in Quadrotor system (Quadrotor robot), 2016 IEEE International Conference on Automatic Control and Intelligent Systems (I2CACIS), 2016, pp. 11-16, doi: 10.1109/I2CACIS.2016.7885281.

[69] Iannace, Gino; Ciaburro, Giuseppe; Trematerra, Amelia. 2019. Fault Diagnosis for UAV Blades Using Artificial Neural Network Robotics 8, no. 3: 59. https://doi.org/10.3390/robotics8030059

[70] Paraskevi Samara. A Statistical Method for the Detection of Sensor Abrupt Faults in Aircraft Control Systems.IEEE Transactions on Control Systems Technology, Vol. 16, No. 4, July 2008

[71] Marzat J, Piet-Lahanier H, Damongeot F, Walter E. Model-based fault diagnosis for aerospace systems: a survey. Proceedings of the Institution of Mechanical Engineers, Part G: Journal of Aerospace Engineering. 2012;226(10):1329-1360. doi:10.1177/0954410011421717

[72] Avram, R. (2016). Fault Diagnosis and Fault-Tolerant Control of Quadrotor UAVs.

[73] Guo K, Liu L, Shi S, Liu D, Peng X. UAV Sensor Fault Detection Using a Classifier without Negative Samples: A Local Density Regulated Optimization Algorithm. Sensors (Basel). 2019;19(4):771. Published 2019 Feb 13. doi:10.3390/s19040771

[74] Micheal Verhaegen. Fault Tolerant Flight Control - A Survey. Fault Tolerant Flight Control. Lecture Notes in Control and Information Sciences, vol 399. Springer, Berlin, Heidelberg.

[75] G. Alcan and M. Unel, Robust hovering control of a quadrotor using acceleration feedback, 2017 International Conference on Unmanned Aircraft Systems (ICUAS), 2017, pp. 1455-1462, doi: 10.1109/ICUAS.2017.7991335.

[76] K. J. Astrom and R. M. Murray. Analysis and Design of Feedback Systems. Preprint, 2005. Available at http://www.cds.caltech.edu/ murray/am05.

[77] Hanmant G. Malkapure, M. Chidambaram, Comparison of Two Methods of Incorporating an Integral Action in Linear Quadratic Regulator, IFAC Proceedings Volumes, Volume 47, Issue 1, 2014, Pages 55-61.

[78] Jean-Yves, Keller, Darouach, Mohamed. (1997). Optimal two-stage Kalman filter in the presence of random bias. Automatica. 33. 1745-1748. 10.1016/S0005-1098(97)00088-5.

[79] Wu EN, Zhang YM, Zhou KM, 2000. Detection, estimation, and accommodation of loss of control effectiveness. Int J Adapt Contr Signal Process, 14(7):775-795.

[80] H. Zaki, M. Unel and Y. Yildiz, Trajectory control of a quadrotor using a control allocation approach, 2017 International Conference on Unmanned Aircraft Systems (ICUAS), 2017, pp. 533-539, doi: 10.1109/ICUAS.2017.7991344.

[81] W. Chen, J. Yang, L. Guo and S. Li, Disturbance-Observer-Based Control and Related Methods—An Overview, in IEEE Transactions on Industrial Electronics, vol. 63, no. 2, pp. 1083-1095, Feb. 2016, doi: 10.1109/TIE.2015.2478397.

[82] Kelman, Tony. ME233 Advanced Control Systems II, Disturbance Observers. UC Berkeley, Spring 2016

[83] H. Zaki and M. Unel, Control of a Hovering Quadrotor UAV Subject to Periodic Disturbances, 2018 6th International Conference on Control Engineering and Information Technology (CEIT), 2018, pp. 1-8, doi: 10.1109/CEIT.2018.8751902.