

FAULT TOLERANT CONTROL OF A QUADROTOR UAV

M.Sc. THESIS

Majid MOGHADAM

Department of Control and Automation Engineering

Control and Automation Engineering Programme

DECEMBER 2015

FAULT TOLERANT CONTROL OF A QUADROTOR UAV

M.Sc. THESIS

Majid MOGHADAM
(504121141)

Department of Control and Automation Engineering

Control and Automation Engineering Programme

Thesis Advisor: Prof. Dr. Fikret CALISKAN

DECEMBER 2015

DÖRT ROTORLU BİR İHA'NIN ARIZA TOLERANSLI KONTROLÜ

YÜKSEK LİSANS TEZİ

**Majid MOGHADAM
(504121141)**

Kontrol ve Otomasyon Mühendisliği Anabilim Dalı

Kontrol ve Otomasyon Mühendisliği Programı

Tez Danışmanı: Prof. Dr. Fikret CALISKAN

ARALIK 2015

To my parents, my siblings, and my love,

FOREWORD

In the name of God, the Most Gracious, the Most Merciful,

I want to thank my lovely mother and father who have shown me the way, held my hand and had my back in every situation and each level of my life, including studying in such a precious university and graduating from it. My beautiful wife, you are amazing. You've been with me since I became student in university. It's been nine years of my unforgettable moments with you, I adore you and love you. I also want to thank my sister, brother and my sister-in-law. Thank you for being with mom and dad when I was overseas.

I also want to thank prof. Fikret Caliskan, prof. Habib Ghanbarpour Asl, prof. Turan Soylemez, and Mehdi Nasiri in helping me in graduating from Master of Science.

This thesis is dedicated to my lovely family, the symbol of sacrifice and affection.

December 2015

Majid MOGHADAM
(M.Sc. in Control and Automation)

TABLE OF CONTENTS

	<u>Page</u>
FOREWORD	ix
TABLE OF CONTENTS	xi
ABBREVIATIONS	xiii
LIST OF TABLES	xv
LIST OF FIGURES	xvii
SUMMARY	xix
ÖZET	xxi
1. INTRODUCTION	1
1.1 Purpose of Thesis	1
1.2 Literature Review	1
1.3 Roadmap.....	2
2. DYNAMICS AND EQUATIONS OF THE QUADROTOR	5
2.1 Input Description	5
2.2 Quadrotor Dynamics and Parameters	6
2.3 Linearization and State Space Description.....	7
3. STATE FEEDBACK CONTROLLER DESIGN	11
4. TSKF FOR STATE ESTIMATION AND FAULT IDENTIFICATION	13
4.1 TSKF Formulation.....	13
4.2 Testing Kalman Estimation and State Feedback Controller.....	15
5. ACTUAL SYSTEM AND LEADER-FOLLOWER CONTROLLER SIMULATION	19
5.1 Actual Sensors	19
5.2 Nonlinear Model in Simulink.....	20
5.3 Reference Follower Controller Design.....	20
5.4 Actual System Simulation Results	26
6. ACTUATOR FAULT	29
6.1 The Impact of Actuators Faults	30
6.2 Actuators Faults and Leader-Follower Controller	31
6.3 Actuator Fault Detection and Diagnosis.....	38
7. SENSOR FAULT	45
7.1 Sensor Fault detection and Diagnosis.....	45
7.2 Active Fault Tolerant Control on Sensor Fault	49
8. ACTUATOR FDD AND SENSOR AFTC	55
9. CONCLUSION	59
REFERENCES	61

CURRICULUM VITAE..... 63

ABBREVIATIONS

FTC	: Fault Tolerant Control
AFTC	: Active Fault Tolerant Control
FDD	: Fault Detection and Diagnosis
TSKF	: Two-Stage Kalman Filter
LQR	: Linear Quadratic Regulator
UAV	: Unmanned Aerial Vehicle
PWM	: Pulse-Width Modulation

LIST OF TABLES

	<u>Page</u>
Table 2.1 : Qball-X4 Quadrotor Parameters.....	7

LIST OF FIGURES

	<u>Page</u>
Figure 2.1 : Control surfaces and coordinate directions of the Qball-X4.	6
Figure 4.1 : The schematic diagram of the Two-Stage Kalman Filter.	15
Figure 4.2 : The overall structure of the closed-loop linear system with TSKF block.....	16
Figure 4.3 : System response to reference inputs and Kalman estimations.	17
Figure 4.4 : Measured and estimated Theta and Phi angles.....	17
Figure 5.1 : Simulating equation (2.3) in Simulink.	21
Figure 5.2 : Simulating angular velocity and orientation of quadrotor in Simulink.....	22
Figure 5.3 : Simulating velocity and position of quadrotor in Simulink.	23
Figure 5.4 : Setting function blocks in Fig. 5.3.	24
Figure 5.5 : Transferring state vector to output vector in Simulink.	25
Figure 5.6 : The structure of the system with reference follower controller (two feedback loops).....	26
Figure 5.7 : Overall structure of the simulated nonlinear system with actual number of sensors in.	28
Figure 5.8 : System in Fig. 5.7 responses to reference inputs.	28
Figure 6.1 : The overall structure of the linear system with faulty actuators in Sumulink.....	31
Figure 6.2 : The impact of 30% fault in actuator 1 on system response.....	32
Figure 6.3 : The effect of fault in actuator 1 on pitch angle.....	32
Figure 6.4 : Control input to actuators when actuator 1 is faulty.....	33
Figure 6.5 : Thrust produced by actuators when actuator 1 is faulty.	33
Figure 6.6 : The impact of faults in actuators 2 and 3 on system response.	34
Figure 6.7 : The effect of faults in actuators 2 and 3 on pitch and roll angles.....	34
Figure 6.8 : Control input to actuators when actuators 2 and 3 are faulty.	35
Figure 6.9 : Thrust produced by actuators when actuator 2 and 3 are faulty.	35
Figure 6.10 : The overall structure of the nonlinear reference follower system with faulty actuators in Sumulink.....	36
Figure 6.11 : Response of the system with faulty actuators and reference follower controller.....	37
Figure 6.12 : Pitch and roll angles of the system with faulty actuators and reference follower controller.....	37
Figure 6.13 : TSKF estimation of the actutors faults.	38
Figure 6.14 : The impact of 75% fault on actuator 3 on system response.	39
Figure 6.15 : The impact of 75% fault on actuator 1 in pitch and roll angles.....	40
Figure 6.16 : The acceleration toward y direction when actuator 3 is 75% faulty..	41
Figure 6.17 : The estimated actuators fault by TSKF when actuator 3 is faulty....	41

Figure 6.18: The impact of faults on all actuators on system response. 42

Figure 6.19: Pitch and roll angles when all actuators are faulty. 42

Figure 6.20: The estimated actuators fault by TSKF when all actuators are faulty. 43

Figure 7.1 : The overall structure of the system with faulty actuators and sensors. 46

Figure 7.2 : The impact of fault on z sensor on system response. 46

Figure 7.3 : Control inputs when fault happened in height sensor..... 47

Figure 7.4 : Height sensor’s residual when the sensor is faulty. 48

Figure 7.5 : Kalman filter residuals when height sensor is faulty. 48

Figure 7.6 : The impact of sensor fault on system response and Kalman estimation..... 49

Figure 7.7 : The impact of faults on three sensors on the system response. 50

Figure 7.8 : The impact of faults on three sensors on pitch and roll angle. 50

Figure 7.9 : Kalman filter residuals when three sensors are faulty. 51

Figure 7.10: Violating linearization condition by acceleration on y direction..... 51

Figure 7.11: The overall structure of the AFTC system. 52

Figure 7.12: Response of the AFTC system when one sensor is faulty. 53

Figure 7.13: Response of the AFTC system when multiple sensors are faulty..... 53

Figure 7.14: Pitch and roll angles of the faulty quadrotor equipped with AFTC. . 54

Figure 8.1 : Final system performance on faulty sensors and actuators. 55

Figure 8.2 : Final system performance on faulty sensors and actuators. 56

Figure 8.3 : Estimated vector in final system..... 56

FAULT TOLERANT CONTROL OF A QUADROTOR UAV

SUMMARY

This thesis addresses the problem of sensor and actuator fault detection and diagnosis (FDD) and as a result fault tolerant control (FTC) of an unmanned quadrotor helicopter in the presence of the observation and process noises. To this end a Two-Stage Kalman Filter (TSKF) to detect, isolate and identify faults is used. Sensor faults are assumed to be bias faults and actuator faults are modeled as a loss of effectiveness in the propellers. Also a reference follower controller using Linear Quadratic Regulator (LQR) technique is designed and used to make the system response to follow a desired trajectory.

Firstly, the system nonlinear dynamics are presented. These equations are linearized around an equilibrium point. The nominal point is assumed a fixed point in the world coordinate with small Euler angles. This indicates the hovering condition by the quadrotor. Using Jacobian matrices, linearized state space representation of the system around a mentioned nominal point is obtained. A feedback gain matrix controller using LQR technique is calculated and the system ability to follow reference input is discussed. However, in further experiments, availability of this controller in following reference inputs in faulty situation will be brought into question.

TSKF is a block which can be used to estimate the states of system and equivalently the loss of effectiveness of propellers. Control input to the system plant and system sensed outputs are two input vectors to this block. Outputs of TSKF block are the estimation of noisy states and loss of effectiveness of rotors (that assumed as a bias vector in state space representation) and calculated residuals. Formulation of TSKF is as straight as the Standard Kalman Filter (SKF) except an additional bias vector that exists in TSKF and indicates the actuators' loss of effectiveness. At first, it's assumed that there is a sensor for each state that measures the state value. This assumption enables us to test the performance of the TSKF in estimating the states. However, this is not the case in actual system where some parameters are measured by a number of sensors like three-axis gyroscope, three-axis accelerometer, indoor positioning system, and range finder (altitude measurement). Then using state estimator, like Kalman Filter, all states of the system (position, orientation and their velocities) are calculated. That's why we aim to take some steps closer to actual system.

After selecting a proper form for C matrix in state space representation which makes us getting closer to the actual system, we continue to simulate the nonlinear dynamics of the system in MATLAB Simulink. As the LQR controller is designed for linear system, the merit of controller in stabilization will be suspicious by substituting the nonlinear model of system with former linear one. In order to overcome this problem, a stabilizing-leader-follower controller design method is presented. Further, the capability of this controller in both normal and faulty situations will be tested.

Actuator faults are assumed as lack of efficiency in propellers and happens in a specific moment. While using linear controller in a linear system, the impact of these faults are obvious in the output. The inability of system response in following reference input after actuator faults is one the negative points of faults in propellers while using linear controller. Fortunately, by changing the current feed-back controller and exchanging with the mentioned reference-follower one, the eligibility of this controller will be proven. As a matter of fact, actuator fault will not cause any problem up to an extent.

By some change of variables and mathematical processes, the loss of effectiveness in rotors enter into the state space equations as new variables in a way that TSKF can estimate their values. By estimating the propellers' loss of effectiveness, the faulty rotor is detected, and isolated. Since, the estimated loss of effectiveness carries the information about time and percent of faults, after detecting these faults FDD is done for actuators. Also after actuator fault diagnosis the proper action can be done that quadrotor be able to continue its scenario. However, this left as a future study for prospective researchers.

Same as the actuator faults, sensor faults happens in a specific moment and assumed as a bias added to measured value. Residuals are defined as either the difference between sensed and estimated states or square of it. As bias faults on sensors occurs suddenly and the estimated states in TSKF are smoother than their sensed values, therefore, at the time of sensors bias fault there will be large deviation in the corresponding residuals. Crossing the deviated residual from its threshold is an alarm for occurred fault. Residuals carry the information about time, size and source of faults, hence, sensor fault identification can be done by observing the amplitude of faulty sensor's residual. Whenever a residual crossed its threshold, means that a sensor bias fault is occurred. As each sensor's measured value is assumed as a state, therefore there is a residual for each sensor and fault isolation, and as a result, FDD can be done this way. By analyzing the impact of measurement faults it will be seen that, as the incorrect value is fed back to controller, the reaction to this will cause the quadrotor deviate from its reference value. This is the vital impact of sensor faults. In seeking a solution to this crisis, we will see that, at the time of fault, magnitude of the residual belonging to faulty sensor is equal to the amplitude of faulty sensor's bias. This is an important fact, we realized that by taking advantage of this it's possible to compensate the faulty measurement and send the corrected value to the controller. By realization of this correction, a new method for Active Fault Tolerant Control (AFTC) on sensor faults for quadrotor is performed.

Finally, various faults on sensors and actuators happening on the either same or different moments are simulated and the ability of controller in stabilization and following the reference input together with the capability of FTC block in detecting, diagnosing, and controlling these faults is presented and analyzed.

DÖRT ROTORLU BİR İHA'NIN ARIZA TOLERANSLI KONTROLÜ

ÖZET

Bu tez sensör ve eyleyici arıza tespiti ve teşhisi (Fault Detection and Diagnosis) sorununu sunar, ve bunun yardımcısıyla bir dört rotorlu insansız hava aracının arıza tolerans kontrol problemini observation ve process gürültüleri olduğunda analiz etmeye çalışır. Bu amaçla, Two-Stage Kalman Filter (TSKF) arızanı bulmak, yalıtım, ve belirlemek için kullanılmaktadır. Sensor arızaları bias fault olarak kabul edilmişlerdir ve aktüatör arızaları pervanelerdeki etkinliğinin kaybı (loss of effectiveness) modellenmiştir. Ayrıca Lineer Kuadratik Regülatör (LQR) tekniğini kullanarak bir referans takipçisi kontrolü tasarlanmış ve kullanılmıştır.

İlk olarak, sistemin doğrusal olmayan dinamikleri sunulmuştur. Bu denklemler bir denge noktası etrafında doğrusallaştırılmışlardır. Nominal nokta dünya koordinatlarında sabit bir nokta küçük Euler açılarıyla sayılmıştır. Bu sabit uçmak (hovering) durumunu İHA da gösterir. Jacobian matrislerini kullanarak, sistemin doğrusallaştırılmış durum uzay modeli bir soylenmiş nominal noktası etrafında elde edilir. Bir geri kazanım matrisi kontrolörü doğrusal kuadratik regülatör tekniği kullanılarak hesaplanmıştır ve referans girişini takip sistemi kabiliyeti tartışılmıştır. Buna rağmen, Diğer deneylerde, arızalı durumda referans girişlerini takip etmek konusunda bu kontrolörün kullanılabilirliği sorgulanacaktır.

Two-Stage Kalman Filter sistemin durumlarını ve pervanelerdeki etkinliğinin kaybını tahmin etmek için kullanılabilir bir bloktur. Sisteme kontrol girişi ve ölçülen çıkışlar bu bloğun iki giriş vektörleridirler. Two-Stage Kalman Filter bloğun çıkışları gürültülü durumların tahmini ve rotorların etkinliğinin kaybı (ki durum uzay modelinde bir bias vektörü olarak hesaplanmıştır) ve hesaplanmış rezidüeldirler. Two-Stage Kalman Filterde bulunan ek bir bias vektörünün dışında TSKF'nin formülasyonu Standard Kalman Filter (SKF) gibi rahattır. Bu vektör aktüatörlerdeki etkinliğinin kaybı modelliyor. İlk başta, her duruma bir sensör bulunmakta olması varsaymıştır. Bu varsayım Two-Stage Kalman Filterin performansını durumların tahmininde test etmeği bize sağlar. Buna rağmen, gerçek sistemde durum böyle değil çünkü gerçek sistemde bir kaç sensörü kullanarak bazı parametreler hesaplanır. Three-axis gyroscope, three-axis accelerometer, indoor positioning system, ve range finder (yüksekliği ölçen) bahs edilen sensörlere dandılar. Sonra durum tahmincisini kullanarak (Kalman filtresi gibi) sistemin tüm uzay durumunda bulunan durumlar (pozisyon, oryantasyon ve onların hızları) hesaplanır. Bu nedenle, kaç adım atmayla gerçek sisteme yaklaşmayı amaçlıyoruz.

Uygun bir uzay durumunda bulunan C matrisi seçtikten sonra, ki bizi gerçek sisteme daha yaklaştırır, sistemin doğrusal olmayan dinamiklerini MATLAB Simulinkde simüle etmeyle devam edeceğiz. Linear Quadratic Regulator kontrolör lineer sistem için dizayn edilmiştir, o yüzden, eğer sistemim doğrusal modelini doğrusal olmayan

modelle deyiştirirsek kontrolör stabilizasyon konusunda başarısız olacaktır, bu sorunu çözmek için bir stabilize eden leader-follower (referans takipçısı) kontrolörün dizayn yöntemi sunulacaktır. Ondan sonra, normal ve arızalı durumlarda bu kontrolörün beceriyi denelenecektir.

Eyleyici hataları özel bir zamanda oluşan, pervanelerdeki verimlilik eksigi olarak kabul edilir. Doğrusal sistemde doğrusal kontrolör kullanılırken, bu hataların etkisi açık bir şekilde sistemin çıkışında gözükür. Eyleyici hatalarından sonra takip eden referans girişlerindeki sistemin yanıtındaki yetersizlik, doğrusal kontrolör kullanılmasından kaynaklanan olumsuz noktalardan birisidir. Neyse ki, geçerli geri beslemeli kontrolörün deęişimi ve sözü geçen referansı takibinin deęişmesi ile bu kontrolörün uygunluğu gösterilmiş olur. Nitekim buraya kadar, eyleyici hataları kapsamlı ölçüde soruna neden olmayacaktır.

Bi ğir sürü deęişkenlerin deęişimi ve matematiksel işlemler kullanarak, rotorlardaki etkinliğinin kaybı yeni bir deęişken olarak uzay durum denklemlerine girer, o yuzden Kalman filtresi bu arızaların büyüklüğünü, zamanını ve hangi motorda olduğunu tahmin edip ve bular. Pervanelerdeki etkinliğinin kaybı tahmin ettiğimiz için arızali rotor ayrılıp ve belirlenir. Mademki tahmin edilen etkinlik kaybı arızalardan bir sürü bilgi (zaman ve yuzdegibi) taşıyor. bu arızalar bulunduktan sonra Fault Detection and Diagnosis (FDD) eyleyicilerde yapılmış olur. Ayrıca, eyleyici arıza teşhisinden sonra oyle bir uygun hareket yapıla bilir ki insansız hava aracı uçuşuna devam etsin. Ancak bu konu prospektif araştırmacılar için geleceğin çalışması olarak açık bırakılmıştır.

Eyleyici hataları gibi, sensor hataları bir anda aluyor ve bir bias olarak ölçülmüş miktara ekleniyorlar. Rezidüel iki şekilde hesaplanabilirler. Biricisiyse, ölçülmüş miktarla tahmin edilen deęerin arasındaki fark, ikincisi, bah edilen farkın karesi. Bu bir gerçek ki sensorlardaki arızalar bir an oluyorlar o yuzden tahmin edilen durum uzaydaki durumlar ölçülmüş miktardan daha pürüzsüz (daha yumuşak), bu nedenle arızalı sensora ayıt olan rezidüel hata anında bir büyük miktar olacaktır. Her rezidüel için bir threshold (eşik) oluşmuştur, bu kırmızı çizgiyi geçmek sensörde arıza olmasına bir alarmdır. Rezidüeller zaman, büyüklük ve arızanın nerede olduğundan bilgi taşıyorlar, bundan dolayı hatalı sensörün rezidüelinin büyüklüğüne bakarak fault identification gerçekleşebilir. Her zamanki bir rezidüel kendine ait olan thresholdu geçerse bir sensör arızası oluşmuştur. Mademki her sensörün ölçen deęeri bir durum hesaplanmıştır, o yuzden her sensöre ait bir rezidüel vardır bu nedenle fault isolation ve dolayısıyla Fault Detection and Diagnosis sensörler için yapıla bilir. Ölçüm arızasının darbesini analiz etmeyle göreceğiz ki, yanlış deęer kontrolöre geri donuyor ve kontrolör ölçülen deęerin referans girişten uzaklaştığını zan eder, bu nedenle reaksiyon olarak doğru olmayan bir kontrol girişi üretir ve dolayısıyla uçak referans pozisyondan uzaklaşar. Bu sensör arızalarının hayati darbesidir. Bu kriz için bir çözüm bulma amacıyla buna şahit olacağız ki, hatalı sensöre ait olan rezidüelin büyüklüğü aynı sensörde ulaşan hatanın boyutuyla (arıza anında) eşittir. Bu çok önemli bir gerçektir, biz bunu farketdik ki bundan yararlanarak bu mümkünki arızalı sensörün yanlış deęerini düzelte biliriz. Ve bu düzeltilmiş deęeri yeniden kontrolöre geri göndermek olur. Bunu gerçekleştirdikten sonra göreceğiz ki artık yanlış deęer geri beslem yardımcıyla kontrolere girmeyecektir, dolayısıyla sistem referans girişi hatta hatanın olduğu zamandada takip etmeye devam edecek. Bu düzeltme gerçekleşmesi ile yeni bir yöntem Active Fault Tolerant Control (AFCT) İHANin sensörlerinin üzerinde bulunmuş olur.

Nihayet, birçok çeşitli arızalar, sensörler ve eyleyiciler üzerinde simule edilmiştir. Bu arızalar farklı ve ya aynı zamanda oluşturulmuşlar. Dolayısıyla, kontrolörü stabilizasyon ve referans takip etme konusunda artık Fault Tolerant Control (FTC) bloğunun başarıyla arızaların tesbit ve kontrol etmesini test ve analiz edilmişler.

1. INTRODUCTION

Unmanned Aerial Vehicles (UAVs) have experienced in recent years an important growth both in research activities and in the industrial development of platforms to be used in applications such as fields of security, management of natural risks, intervention in hostile environments, management of ground installations, agriculture, military, and also product delivery. Quadrotors share the notable portion of these UAVs. In most of the UAVs' applications, reliability, maintainability and survivability are essential parameters to be taken into account and these have drawn significant attention into Fault Detection and Diagnosis (FDD). For implementing a Fault Tolerant Control (FTC) system, FDD blocks are essential and implemented to detect and isolate actuator and sensor faults. The information of faults are taken in order to counteract the effects of faults on the system's response. To do so, a Two-Stage Kalman Filter (TSKF) scheme for actuator and sensor faults detection, isolation and identification (fault diagnosis) is used.

1.1 Purpose of Thesis

The purpose of this thesis is firstly to present an specific quadrotor's equations and parameters together with introducing two controller design methods to stabilize the system response and make it to follow reference inputs in normal and faulty situations. Secondly, we aim to analyze the system behavior in the existence of sensor and actuator faults, how they affect the system output and how they can be detected and diagnosed using TSKF. We also aim to control these faults using the information we have in hand to counteract the impact of faults, this is called Fault Tolerant Control (FTC).

1.2 Literature Review

In recent years, researchers have presented different control strategies for quadrotors, such as Linear Quadratic Regulator (LQR) [1] for linear model of a quadrotor, Backstepping [2, 3], PID [4], Sliding Mode [5], and Feedback Linearization [6] for the

nonlinear model of UAV. FDD has become focused in some researches within the last two decades [7], after FDD it's possible to counteract the effects of faults on system to increase the survivability and reliability of the system. Both sensor and actuator faults have been taken into account by several studies focusing on FTC of quadrotor using PID [4], Sliding Mode [5] and TSKF [1, 7] but none of the researches included both quadrotor's sensor and actuator faults simultaneously except [1]. Sensor and actuator FDI on an UAV is discussed in [8].

1.3 Roadmap

In this thesis, a nonlinear model of a quadrotor is given and linearized around an equilibrium point, this point is assumed the hovering mode of the quadrotor. Using LQR controller design, state-feedback gain is calculated such that the output of system follows a desired trajectory in no-fault mode. Then, formulation of the TSKF to estimate the states trajectories and to detect and diagnose actuator and sensor faults is given. By replacing the nonlinear model of system with the linearized representation we take a step closer to actual system. Obviously, LQR controller is not able to stabilize this nonlinear model of system, therefore, a leader-follower controller to stabilize the nonlinear system and to make the response follow reference input is formulated and designed.

Actuator faults are assumed as lack of efficiencies in the motors' propellers. They are simulated as a bias vector in state space representation of system and TSKF is set to estimate this vector. The impact of actuator faults on system response is discussed. We will see that, while using LQR controller in existence of faults, although the system is stable, it's not able to follow input references, to figure this out, leader-follower controller is substituted with the former one and capability of this is discussed.

When it comes to sensor faults, they are simulated as bias faults occurring at a specific time. As the incorrect measured value is fed back to controller, unintentionally it produce inaccurate control input and this lead to deviation from intended value in system response. Using residuals created by TSKF, sensor faults are detected and diagnosed. For the first time ever, the information hidden in residuals are taken and the correlation process for faulty sensors is done. By doing so, Active Fault Tolerant Control (AFTC) is performed for sensor faults.

This dissertation is organized in nine chapters as follows, chapter 2 describes the nonlinear dynamics of Qball X4 quadrotor and its linearized model in state space mode. State feed-back controller using LQR method is designed in chapter 3. TSKF equations together with closed-loop system simulation is discussed in chapter 4. In chapter 5 in order to get closer to the actual system two important steps are taken, in addition the formulation of reference-follower controller is given. Actuator faults' impacts and a method for counteracting them is presented in chapter 6. Also FDD for this faults is presented in this part. Sensor faults simulation and FDD of them is given in chapter 7. Finally, in chapter 8 actuator and sensor faults happening in either same or different times are simulated and the ability of closed-loop system including FTC algorithm is discussed.

2. DYNAMICS AND EQUATIONS OF THE QUADROTOR

2.1 Input Description

The complete dynamics of an aircraft, taking into account aero-elastic effects, flexibility of the wings, internal dynamics of the engine and the whole set of changing variables are quite complex and somewhat unmanageable for the purposes of control. Therefore, it is interesting to consider a simplified model of an aircraft formed by a minimum number of states and inputs, but retaining the main features that must be considered when designing control laws for a real aircraft. Conventional helicopters modify the lift force by varying the collective pitch. Such aerial vehicles use a mechanical device known as swashplate. This system interconnects servomechanisms and blade pitch links in order to change the rotor blades pitch angle in a cyclic manner, so as to obtain the pitch and roll control torques of the vehicle. In contrast, the quad-rotor does not have a swashplate and has constant pitch blades. Therefore, in a quad-rotor we can only vary the angular speed of each one of the four rotors to obtain the pitch and roll control torques.

The quadrotor is controlled by the angular speeds of four electric motors as shown in Fig. 2.1. Each motor produces a thrust and a torque, whose combination generates the main thrust, the yaw torque, the pitch torque, and the roll torque acting on the quadrotor [9]. Let T_i and τ_i be the torque and thrust for rotor respectively ($i= 1, \dots, 4$) and L be the distance from each rotor to center of mass of quadrotor. Now, a set of four control inputs as a functions of torques and thrusts are defined as follow

$$\begin{aligned} u_z &= T_1 + T_2 + T_3 + T_4 \\ u_\theta &= L(T_1 - T_2) \\ u_\phi &= L(T_3 - T_4) \\ u_\psi &= \tau_1 + \tau_2 - \tau_3 - \tau_4 \end{aligned} \tag{2.1}$$

where u_z is the main thrust and u_θ , u_ϕ and u_ψ are applied pitch, roll and yaw moments respectively which are the Euler angles. The main thrust is the sum of individual

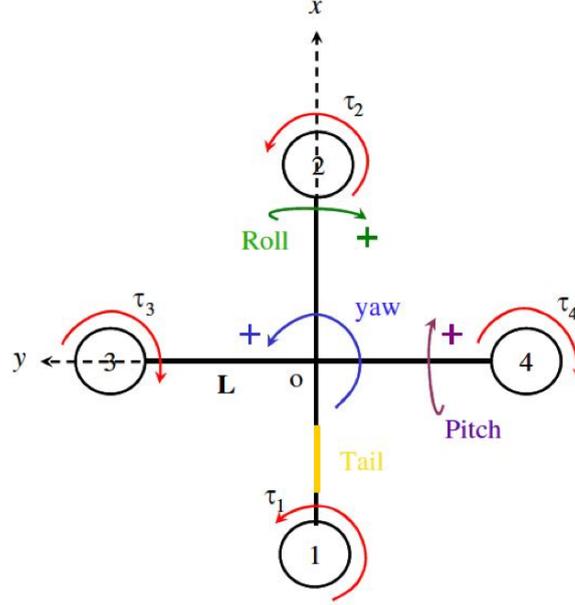


Figure 2.1: Control surfaces and coordinate directions of the Qball-X4.

thrusts of each motor. The pitch torque is a function of the difference $T_1 - T_2$, the roll torque is a function of $T_3 - T_4$, and the yaw torque is the sum $\tau_1 + \tau_2 - \tau_3 - \tau_4$. The torque produced by each rotor is proportional to its thrust via the relation of $\tau_i = K_\psi T_i$ where K_ψ is the constant of proportionality. In this article, we assume that the quadrotor is Qball-X4 which is developed by Quanser Inc. It is equipped with four 10-inch propellers and standard RC motors and speed controllers. The relation between generated thrust of each motor and Pulse Width Modulation (PWM) input to each motor is approximated by a zero-order transfer function [7]

$$T_i \approx K u_i; i = 1, \dots, 4 \quad (2.2)$$

where K is a positive gain. Therefore, the relation between applied main thrust and moments and PWM inputs to rotors are

$$\begin{bmatrix} u_z \\ u_\theta \\ u_\phi \\ u_\psi \end{bmatrix} = \begin{bmatrix} K & K & K & K \\ KL & -KL & 0 & 0 \\ 0 & 0 & KL & -KL \\ KK_\psi & KK_\psi & -KK_\psi & -KK_\psi \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \quad (2.3)$$

2.2 Quadrotor Dynamics and Parameters

Using both Euler-Lagrange and Newton-Euler approach [9], it's possible to show that the dynamics of the quadrotor UAV can be defined as following nonlinear equations

Table 2.1: Qball-X4 Quadrotor Parameters.

Parameter	Description	Value
K	Thrust gain	175
L	Distance from motor to center of gravity	0.2 m
K_ψ	Thrust-to-moment gain	0.023
m	Mass	1.42 kg
g	Gravitational acceleration	9.81 m/s ²
$J_1; J_2; J_3$	Moments of inertia	0.03; 0.03; 0.04 kg.m ²

$$\left\{ \begin{array}{l} m\ddot{x} = u_z (\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi) \\ m\ddot{y} = u_z (\cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi) \\ m\ddot{z} = u_z (\cos\phi \cos\theta) - mg \\ J_1 \ddot{\theta} = u_\theta \\ J_2 \ddot{\phi} = u_\phi \\ J_3 \ddot{\psi} = u_\psi \end{array} \right. \quad (2.4)$$

where x , y and z are the coordinates of the quadrotor UAV center of mass in the earth-frame (see Fig. 2.1). m is the mass, θ , ϕ and ψ are the pitch, roll and yaw Euler angles respectively, and J_i ($i=1,2,3$) are the moments of inertia along y , x and z directions respectively. Considered Qball-X4 quadrotor parameters and their numerical values are given in Table 2.1.

2.3 Linearization and State Space Description

Obviously, (2.4) is a nonlinear set of equations. In order to linearize these equations around a fixed equilibrium point state variables are defined as

$$\begin{aligned} \underline{x} &= [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7 \ x_8 \ x_9 \ x_{10} \ x_{11} \ x_{12}]^T \\ &= [x \ \dot{x} \ y \ \dot{y} \ z \ \dot{z} \ \theta \ \dot{\theta} \ \phi \ \dot{\phi} \ \psi \ \dot{\psi}]^T \end{aligned} \quad (2.5)$$

that underline implies the vector form. Nonlinear equations in (2.4) in the form of $\underline{\dot{x}}(t) = \underline{f}(\underline{x}(t), \underline{u}(t), t)$, become

$$\dot{\underline{x}} = \begin{bmatrix} f_1 \\ f_2 \\ f_3 \\ f_4 \\ f_5 \\ f_6 \\ f_7 \\ f_8 \\ f_9 \\ f_{10} \\ f_{11} \\ f_{12} \end{bmatrix} = \begin{bmatrix} x_2 \\ \frac{1}{m}u_z(\cos x_9 \sin x_7 \cos x_{11} + \sin x_9 \sin x_{11}) \\ x_4 \\ \frac{1}{m}u_z(\cos x_9 \sin x_7 \sin x_{11} - \sin x_9 \cos x_{11}) \\ x_6 \\ \frac{1}{m}u_z(\cos x_9 \cos x_7) - g \\ x_8 \\ \frac{1}{J_1}u_\theta \\ x_{10} \\ \frac{1}{J_2}u_\phi \\ x_{12} \\ \frac{1}{J_3}u_\psi \end{bmatrix} \quad (2.6)$$

To define the equations in the linear state space form we need to define a nominal point which the state variables stay around. To this end, we assume that the quadrotor stays in the predefined position with no yawing and small roll and pitch angles in the nominal point. Also, velocity of all variables are small. The nominal inputs are

$$[\tilde{u}_z \quad \tilde{u}_\theta \quad \tilde{u}_\phi \quad \tilde{u}_\psi]^T = [mg \quad 0 \quad 0 \quad 0]^T \quad (2.7)$$

where *tilde* denotes the nominal value and g is the gravitational acceleration. Linearized state space A , B , C and D matrices are the Jacobian matrices calculated at the nominal points [10]

$$\begin{aligned} \dot{\underline{x}} &= A\underline{x} + B\underline{u} \\ \underline{y} &= C\underline{x} + D\underline{u} \end{aligned} \quad (2.8)$$

where

$$\begin{aligned} A &= \frac{\partial f}{\partial x}(\tilde{x}, \tilde{u}, t), B = \frac{\partial f}{\partial u}(\tilde{x}, \tilde{u}, t) \\ C &= I_{12 \times 12}, D = 0_{12 \times 4} \end{aligned} \quad (2.9)$$

that I and 0 are the Identity and Zero matrices respectively and

$$\underline{u} = [u_z \quad u_\theta \quad u_\phi \quad u_\psi]^T \quad (2.10)$$

Let us take u_i (PWM inputs to propellers) as inputs to the system, using (2.3), (2.6) and (2.9), A and B matrices become

$$\begin{aligned}
A = & \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & g & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -g & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \\
B = & \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ K/m & K/m & K/m & K/m \\ 0 & 0 & 0 & 0 \\ KL/J_1 & -KL/J_1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & KL/J_2 & -KL/J_2 \\ 0 & 0 & 0 & 0 \\ KK_\psi/J_3 & KK_\psi/J_3 & -KK_\psi/J_3 & -KK_\psi/J_3 \end{bmatrix} \tag{2.11}
\end{aligned}$$

Note that in sixth row of (2.6) gravitational acceleration (g) is not eliminated by linearizing the equations therefore it should not be forgotten in simulation. We simulated this parameter as a trim point for $\underline{\dot{x}}$ in MATLAB Simulink. As we use the discrete TSKF, we are interested in the discrete-time state space representation

$$\begin{aligned}
A_k &= e^{AT_s}, \quad B_k = \int_0^{T_s} e^{A\tau} d\tau B \\
C_k &= I_{12 \times 12}, \quad D_k = 0_{12 \times 4}
\end{aligned} \tag{2.12}$$

where A_k , B_k , C_k and D_k are discrete-time state space matrices and T_s is the sampling period in seconds which is taken as 0.01 seconds and e denotes the matrix exponential. At first, we assume that all states are available to fed them back to the state feedback controller, this is possible if each state be measured by a sensor, however this is not true in real system since there exist gyroscopes, acidometers, position cameras and compass to calculate the position and orientation of the quadrotor as well as their derivatives and using a state estimator like Kalman Filter the value for states can be estimated. Therefore, we choose matrix C of the system as identical matrix which means that all

states are calculated in the output. As we go further in this dissertation we try to make simulated model of the system as close as possible to the actual system, therefore, matrix C will change to its actual value in next sections.

3. STATE FEEDBACK CONTROLLER DESIGN

All eigenvalues of A_k are equal to 1 so they are located on the unit circle in z-plane so that the system is unstable. Using discrete-time LQR method we can locate all the eigenvalues on the desired locations in z-plane. For a linear discrete-time state space model

$$x_{-k+1} = A_k x_{-k} + B_k u_{-r_k} \quad (3.1)$$

where subscript k implies the k^{th} sample and

$$u_{-r_k} = [u_{1k} \ u_{2k} \ u_{3k} \ u_{4k}]^T, \quad (3.2)$$

a control input

$$u_{-r_k} = r_{-k} - K_c x_{-k} \quad (3.3)$$

is determined, where K_c is the feedback gain matrix and

$$r_{-k} = [r_{z_k} \ r_{x_k} \ r_{y_k} \ r_{\psi_k}]^T \quad (3.4)$$

is the reference input vector for the specified directions, such that the closed-loop system

$$x_{-k+1} = [A_k - B_k K_c] x_{-k} + B_k r_{-k} \quad (3.5)$$

is stable and a performance measure

$$J = \sum_{k=1}^{\infty} (x_{-k}^T Q x_{-k} + u_{-r_k}^T R u_{-r_k}) \quad (3.6)$$

is minimized (Q and R are the weighting matrices). By choosing $Q = \text{diag}(0.115, 0.1, 0.115, 0.1, 12 \times 10^{-6}, , 0.029, 0.1)$ and $R = I_{4 \times 4}$ the following feedback gain matrix is calculated using MATLAB.

A future work is planned to develop this controller to the robust controller found by Pole Coloring and Genetic Algorithm so that the quadrotor parameters can change within the predetermined range. Consequently, even in the flying scenarios in which a mass is carried by quadrotor, the response will remain close to the reference input.

4. TSKF FOR STATE ESTIMATION AND FAULT IDENTIFICATION

4.1 TSKF Formulation

The TSKF was introduced and formulated by Keller and Darouach [11] in 1997 and applied to actuator fault detection and diagnosis by Amoozgar et al [7] in 2012 and Caliskan et al in 2014 [8]. Based on the linearized model of the open-loop UAV around a trim point, and a parameterization of actuator faults which is done in chapter 6, the following discrete time model with process and measurement noises is used as the design model of the TSKF

$$\begin{aligned} \underline{x}_{k+1} &= A_k \underline{x}_k + B_k \underline{u}_{r_k} - B_k U_k \underline{\gamma}_k + \underline{w}_k \\ \underline{y}_k &= C_k \underline{x}_k + D_k \underline{u}_{r_k} + \underline{v}_k \\ \underline{\gamma}_{k+1} &= \underline{\gamma}_k + \underline{w}_k^\gamma \end{aligned} \quad (4.1)$$

where $\underline{\gamma}$ is bias vector of dimension four, representing faults entering actuators. \underline{w}_k , \underline{v}_k and \underline{w}_k^γ are uncorrelated Gaussian random vectors with zero means and covariance matrices:

$$\begin{aligned} Q_k &= \text{diag}(10^{-12} I_{6 \times 6}, 10^{-24} I_{6 \times 6}), \\ R_k &= 10^{-12} \times \text{diag}(10^6, 1, 10^6, 1, 10^6, 1, 1, 1, 1, 1, 1, 1), \\ Q_k^\gamma &= 10^{-3} \times I_{4 \times 4} \end{aligned} \quad (4.2)$$

respectively. Actuators faults are simulated as loss of effectiveness in the propellers and modeled by the term $-B_k U_k \underline{\gamma}_k$ such that

$$\begin{aligned} \underline{\gamma}_k &= [\gamma_{1_k} \ \gamma_{2_k} \ \gamma_{3_k} \ \gamma_{4_k}]^T \quad 0 \leq \gamma_{i_k} \leq 1; \ i = 1, \dots, 4 \\ U_k &= \text{diag}(u_{1_k}, u_{2_k}, u_{3_k}, u_{4_k}) \end{aligned} \quad (4.3)$$

where $\gamma_{i_k} = 0$ and $\gamma_{i_k} = 1$ mean that the i^{th} actuator is completely healthy or faulty respectively. E.g. $\gamma_{3_k} = 0.35$ means that 3rd actuator has 35 percent loss of effectiveness and u_{i_k} is control input to i^{th} actuator at k^{th} sample. A random bias relation as (4.1) can help the true estimation of control effectiveness vector $\underline{\gamma}_k$, where

w_γ is an uncorrelated Gaussian random vector with zero mean and covariance matrix Q_k^γ . The minimum variance solution to estimate the true values of biases (fault parameters) and states is obtained by application of the Two-Stage Kalman Filter as follows.

The optimal bias estimator:

$$\hat{\gamma}_{k+1|k} = \hat{\gamma}_{k|k} \quad (4.4)$$

$$P_{k+1|k}^\gamma = P_{k|k}^\gamma + Q_k^\gamma \quad (4.5)$$

$$\hat{\gamma}_{k+1|k+1} = \hat{\gamma}_{k+1|k} + K_{k+1}^\gamma (\tilde{r}_{k+1} - H_{k+1|k} \hat{\gamma}_{k|k}) \quad (4.6)$$

$$K_{k+1}^\gamma = P_{k+1|k}^\gamma H_{k+1|k}^T (H_{k+1|k} P_{k+1|k}^\gamma H_{k+1|k}^T + \tilde{S}_{k+1})^{-1} \quad (4.7)$$

$$P_{k+1|k+1}^\gamma = (I - K_{k+1}^\gamma H_{k+1|k}) P_{k+1|k}^\gamma \quad (4.8)$$

The Bias-free state estimator:

$$\tilde{x}_{k|k+1} = A_k \tilde{x}_{k|k} + B_k u_k + W_k \hat{\gamma}_{k|k} - V_{k+1|k} \hat{\gamma}_{k|k} \quad (4.9)$$

$$\tilde{P}_{k+1|k}^x = A_k \tilde{P}_{k|k}^x A_k^T + Q_k^x + W_k P_{k|k}^\gamma W_k^T - V_{k+1|k} P_{k+1|k}^\gamma V_{k+1|k}^T \quad (4.10)$$

$$\tilde{x}_{k+1|k+1} = \tilde{x}_{k+1|k} + \tilde{K}_{k+1}^x (y_{k+1} - C_{k+1} \tilde{x}_{k+1|k}) \quad (4.11)$$

$$\tilde{K}_{k+1}^x = \tilde{P}_{k+1|k}^x C_{k+1}^T (C_{k+1} \tilde{P}_{k+1|k}^x C_{k+1}^T + R_{k+1})^{-1} \quad (4.12)$$

$$\tilde{P}_{k+1|k+1}^x = (I - \tilde{K}_{k+1}^x C_{k+1}) \tilde{P}_{k+1|k}^x \quad (4.13)$$

The filter residual and its covariance:

$$\tilde{r}_{k+1} = y_{k+1} - C_{k+1} \tilde{x}_{k+1|k} \quad (4.14)$$

$$\tilde{S}_{k+1} = C_{k+1} \tilde{P}_{k+1|k}^x C_{k+1}^T + R_{k+1} \quad (4.15)$$

The coupling equations:

$$W_k = A_k V_{k|k} - B_k U_k \quad (4.16)$$

$$V_{k+1|k} = W_k P_{k|k}^\gamma (P_{k+1|k}^\gamma)^{-1} \quad (4.17)$$

$$H_{k+1|k} = C_{k+1} V_{k+1|k} \quad (4.18)$$

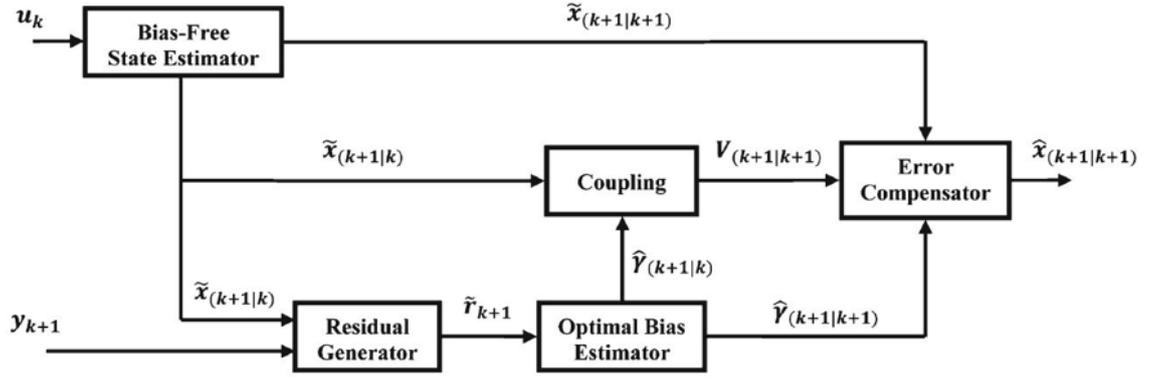


Figure 4.1: The schematic diagram of the Two-Stage Kalman Filter.

$$V_{k+1|k+1} = V_{k+1|k} - \tilde{K}_{k+1}^x H_{k+1|k} \quad (4.19)$$

The compensated error and covariance estimator:

$$\hat{x}_{k+1|k+1} = \tilde{x}_{k+1|k+1} + V_{k+1|k+1} \hat{Y}_{k+1|k+1} \quad (4.20)$$

$$P_{k+1|k+1} = \tilde{P}_{k+1|k+1}^x + V_{k+1|k+1} P_{k+1|k+1}^y V_{k+1|k+1}^T \quad (4.21)$$

Like other type of Kalman filters, the equations can be divided into two sets: the time update equations and the measurement update equations. The time update equations which can be distinguished by “(k + 1|k)” subscription are responsible to obtain a priori estimates by moving the state and error covariances one step ahead in the time domain. While the measurement update equations, shown by “(k + 1|k + 1)” subscription are responsible to obtain a posteriori estimates through feedbacking measurements into the a priori estimates. In other words, time update equations are used for prediction, while measurement updates are used for correction. Indeed, the whole prediction-correction process is used to estimate the states as close as possible to their real values. Fig. 4.1 shows a schematic flow diagram of the TSKF.

4.2 Testing Kalman Estimation and State Feedback Controller

The state feedback controller that is designed using LQR method in chapter 3 and estimation of the formulated TSKF are going to be tested in this section. To this end, system diagram is designed and simulated in MATLAB Simulink as shown in Fig. 4.2. In this simulation, the linearized model of the system is used, also, all the states are fed back to the controller gain which means that, it's assumed that all states are available either using a sensor for each state or using a state estimator like TSKF as discussed

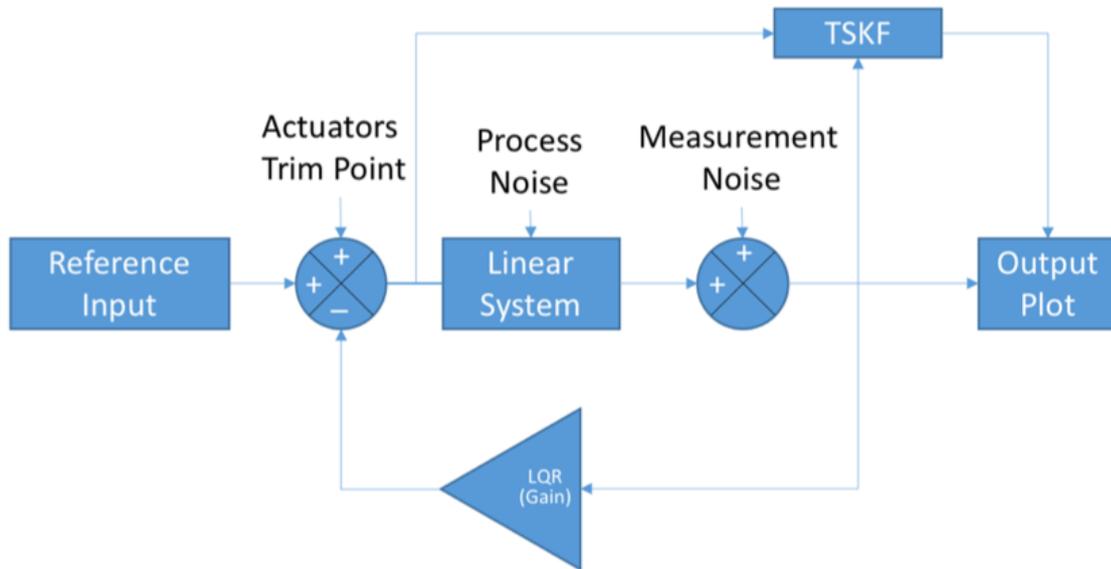


Figure 4.2: The overall structure of the closed-loop linear system with TSKF block.

in section 2.3. Note that Kalman filter estimates the states of the system by taking the control input and output of the system. In addition, input vector (PWM inputs) is produced by transferring reference inputs to PWM inputs using (2.3). By driving the inputs, system response is shown in Fig. 4.3. Obviously, thanks to controller, the output vector tracks the reference input without any overshoot. Rising time for these references is acceptable, however by tuning the Q and R matrix elements we can make the system follow the references faster but this may lead to overshoot. In this system we prefer slower response without overshoots. In addition, Fig. 4.3 and Fig. 4.4 reveals that Kalman filter estimates the states in an acceptable manner. In the following sections only orientation and position parameters (6 states) will be fed back to Kalman filter therefore TSKF will be challenged to estimate all states of the system. These estimated states will be fed back to LQR controller to create the proper control input.

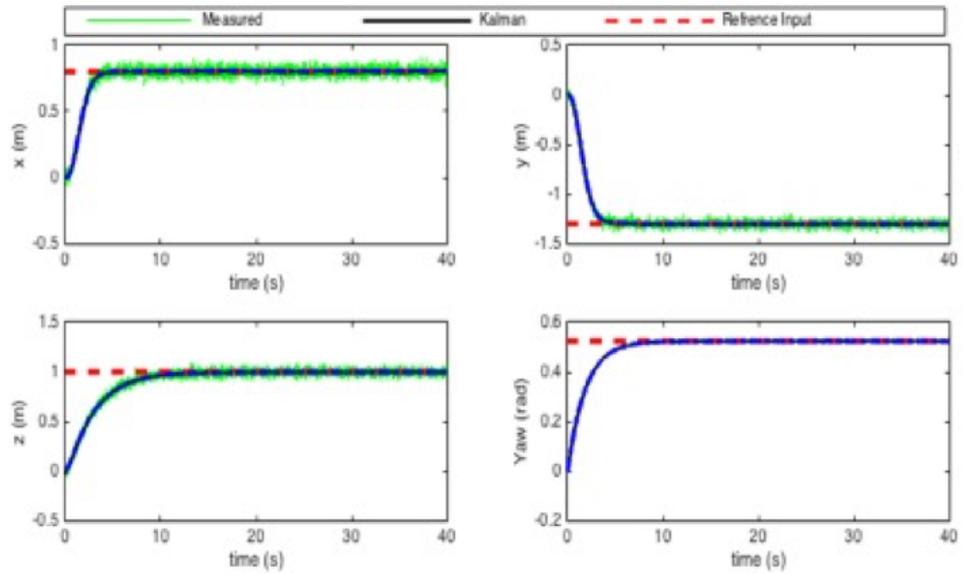


Figure 4.3: System response to reference inputs and Kalman estimations.

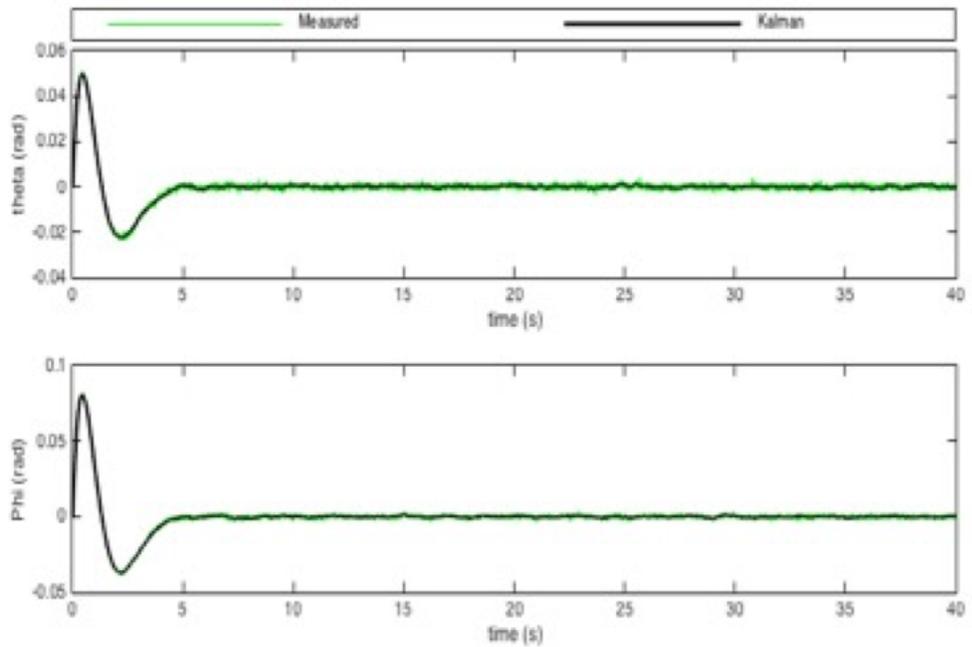


Figure 4.4: Measured and estimated Theta and Phi angles.

5. ACTUAL SYSTEM AND LEADER-FOLLOWER CONTROLLER SIMULATION

5.1 Actual Sensors

In the pervious section we assume that all states are available to be fed back to state feedback controller but as Fig. 4.2 reveals states value calculated by the Simulink are used, which is true only in the simulation not the actual system. Qball-x4 quadrotor is equipped with 3-axis accelerometer that measures the acceleration of the UAV in all exes, 3-axis gyroscope that measures the UAV's angular velocity in 3 axes, sonar height sensor, and a multi-agent platform using 6 indoor OptiTrack cameras to measure position and velocity of the UAV in all axes. It is possible to estimate all 12 states in (2.5) using mentioned sensors. In this section we assume that there are only two sensors that measure position and orientation of the vehicle in 3 axes, therefore, matrix C of the system in (2.8) become

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix} \quad (5.1)$$

The requirement to estimate the all 12 states using 6 position and orientation parameters (x, y, z, θ, ϕ and ψ) the system should be observable. Let us calculate the rank of observability matrix in order to find if the system is observable or not

$$\text{Rank} \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{11} \end{bmatrix} = 12 \quad (5.2)$$

Therefore, the system is observable in the nominal point. By calculating linearized equations using (2.9) we see that the rank of observability matrix does not change

by changing the nominal point, therefore the system is locally observable around the nominal point and it's possible to estimate the 12 states by only sensing 6 position and orientation parameters.

5.2 Nonlinear Model in Simulink

The aim of this section is to make the simulated model of the quadrotor one step closer to the actual system by substituting the linear state-space model with the nonlinear model defined in (2.6). To simulate (2.6) in Simulink we preferred to use blocks instead of programming in Matlab Editor. Fig. 5.1 to Fig. 5.5 show how this is done in Simulink. PWM inputs to actuators (control input) is transferred to lifting thrust and angular moments using (2.3) in Fig. 5.1. By driving these inputs, angular accelerations are achieved as shown in Fig. 5.2. Angular velocities and angular positions can be obtained by integrating the angular accelerations and angular velocities respectively. Also, initial conditions of the states are defined in integrators blocks. Fig. 5.3 reveals how first three equations in (2.6) are simulated. A screen shot of the function blocks (fcn) are shown in Fig. 5.4. Note that process noise (section 4.1) is added to each state using Gaussian Noise Generator block. By choosing different initial seeds to each Gaussian Noise Generator block, each block is initialized in different way to make the produced noise as close as possible to ideal Gaussian noise. Finally, by multiplying the state vector to matrix C , output vector is reached as shown in Fig. 5.5.

5.3 Reference Follower Controller Design

In this section we describe a state-space architecture that tracks the reference input. The approach can be viewed as a generalization of proportional integral (PI) control [12]. Our starting point is to augment the state vector to include the control error $e_k = r - y_k$. Specifically, we use the integrated control error, which describes the accumulated control error. Denoted by x_{I_k} , the integrated control error is computed as

$$x_{I_{k+1}} = x_{I_k} + e_k \quad (5.3)$$

The augmented state vector is $\begin{bmatrix} x_k \\ x_{I_k} \end{bmatrix}$ the control law becomes

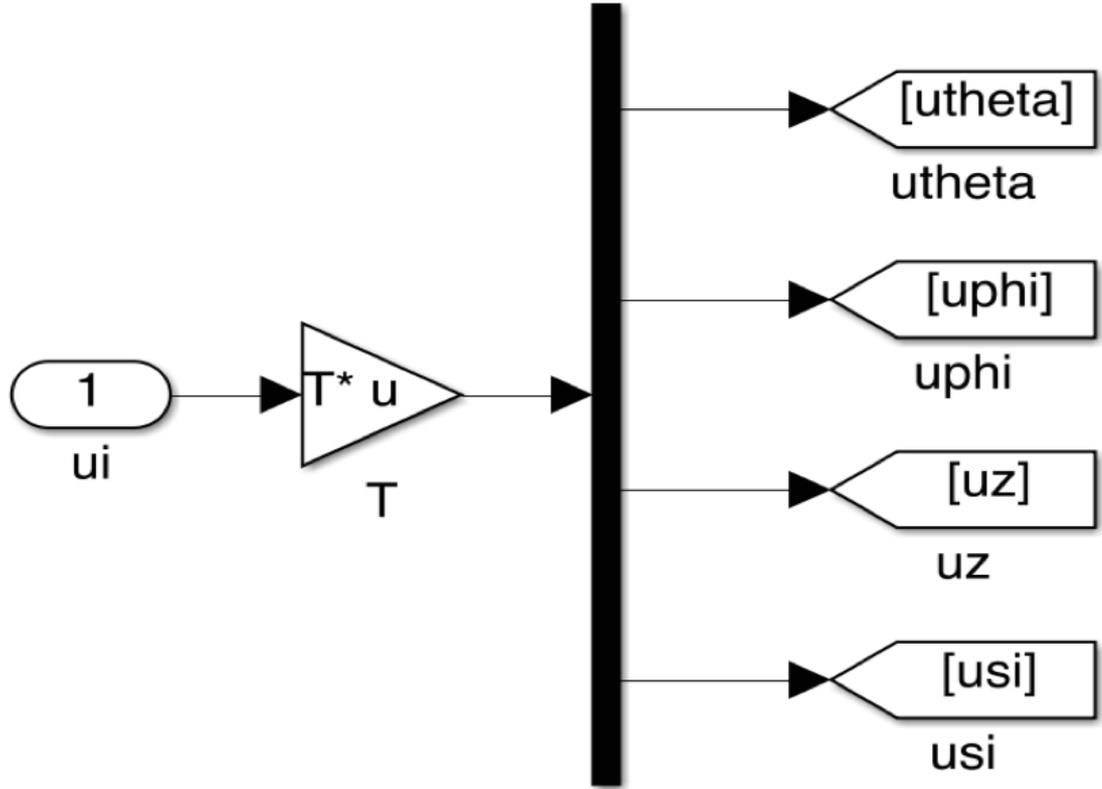


Figure 5.1: Simulating equation (2.3) in Simulink.

$$u_k = \begin{bmatrix} -K_p & -K_I \end{bmatrix} \begin{bmatrix} x_k \\ x_{I_k} \end{bmatrix} \quad (5.4)$$

where K_p denotes the feedback gain for x_k and K_I is the gain associated with x_{I_k} . Fig. 5.6 displays a block diagram of a system that uses the dynamic feedback control law in equation above. The augmented state-space model is

$$\begin{bmatrix} x_{k+1} \\ x_{I_{k+1}} \end{bmatrix} = \begin{bmatrix} A & 0 \\ C & I \end{bmatrix} \begin{bmatrix} x_k \\ x_{I_k} \end{bmatrix} + \begin{bmatrix} B \\ 0 \end{bmatrix} u_k + \begin{bmatrix} 0 \\ 1 \end{bmatrix} r \quad (5.5)$$

and

$$C_y = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (5.6)$$

It turns out that with dynamic state feedback the measured output converges to the reference input. This can be shown as follows. From above equations we know that at steady state $x_{I_{ss}} = x_{I_{ss}} + e_{ss}$. That is, $e_{ss} = 0$. This property holds regardless of the feedback gains K_p and K_I (as long as the closed-loop system is stable).

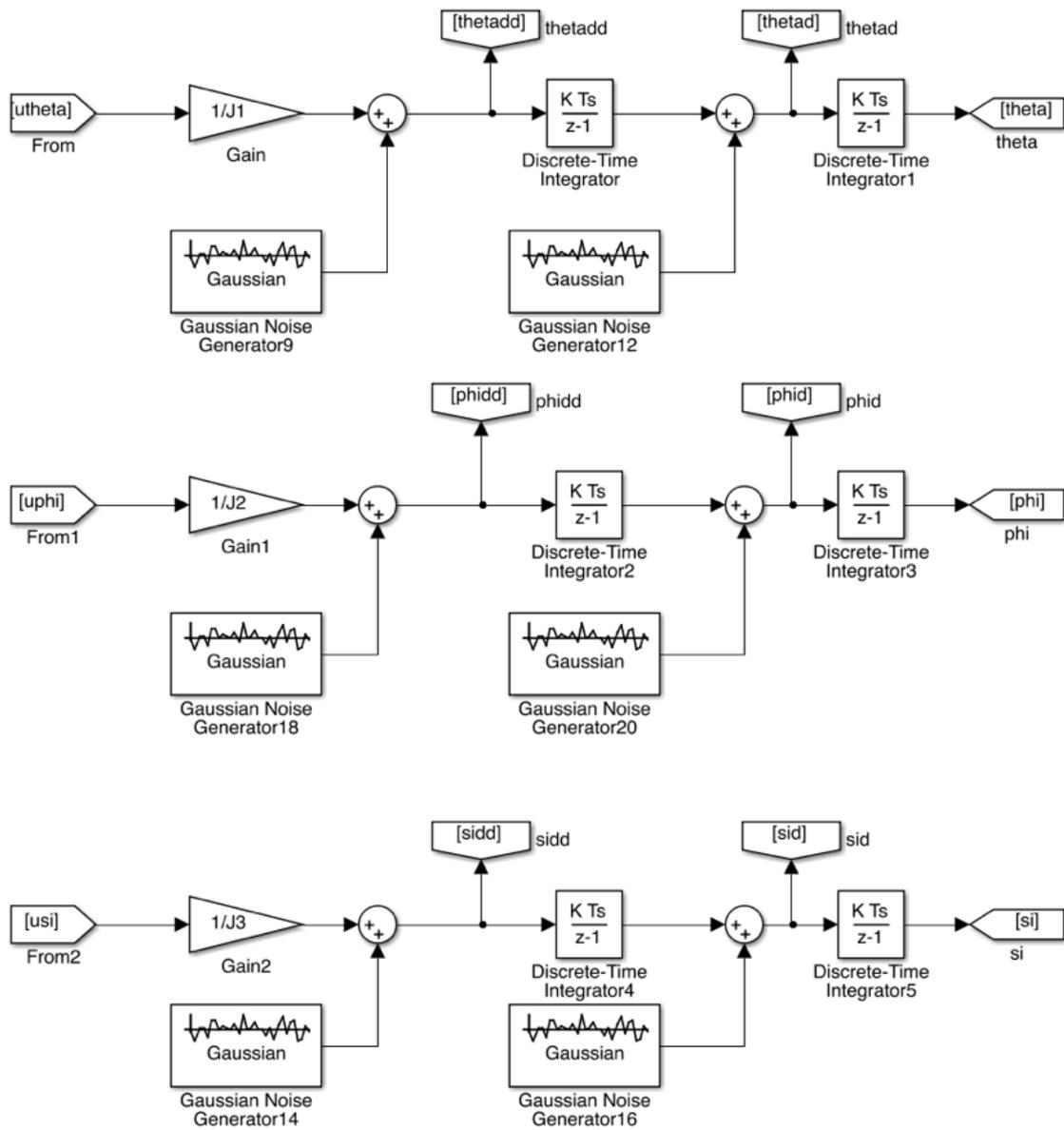


Figure 5.2: Simulating angular velocity and orientation of quadrotor in Simulink.

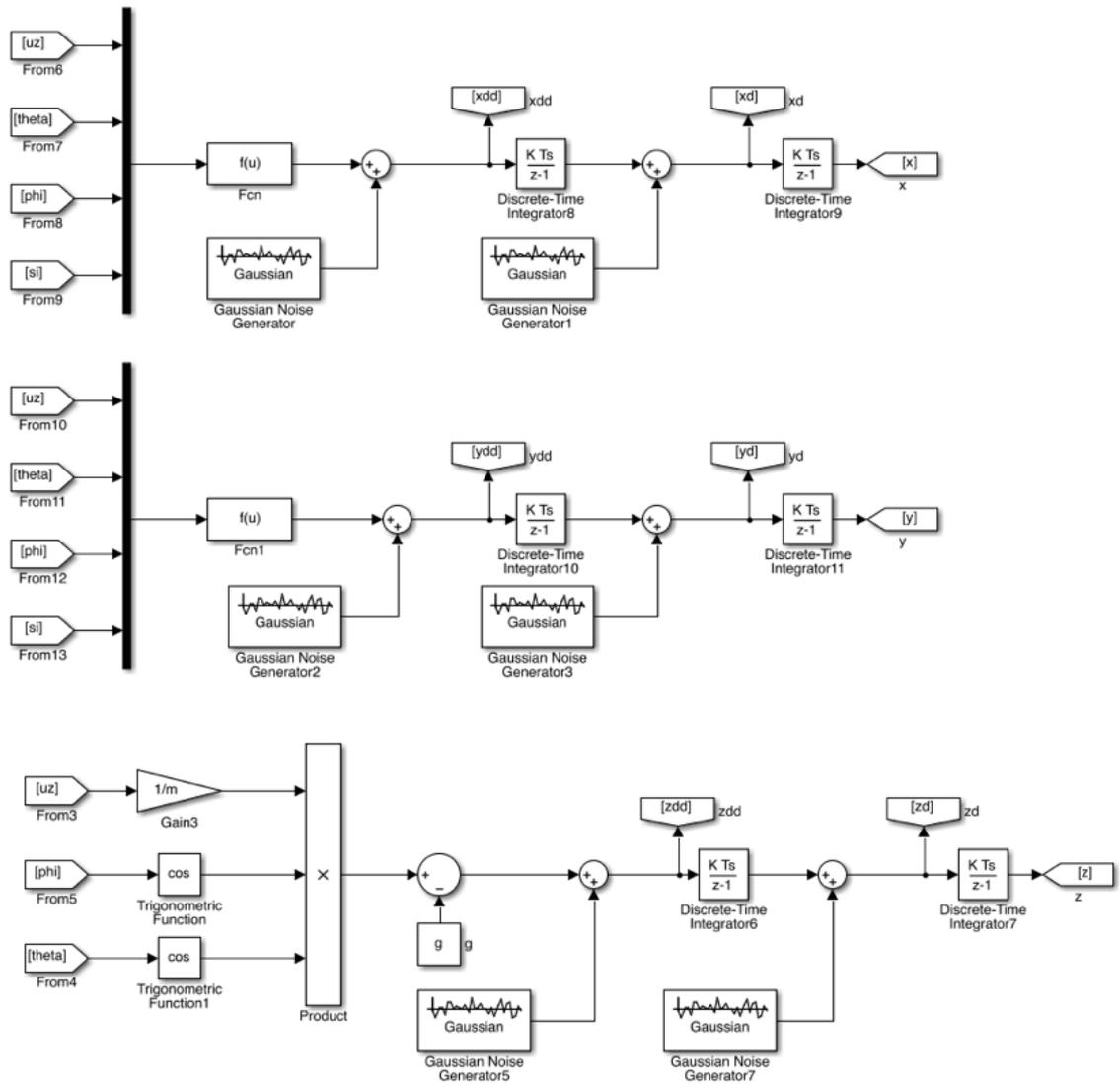


Figure 5.3: Simulating velocity and position of quadrotor in Simulink.

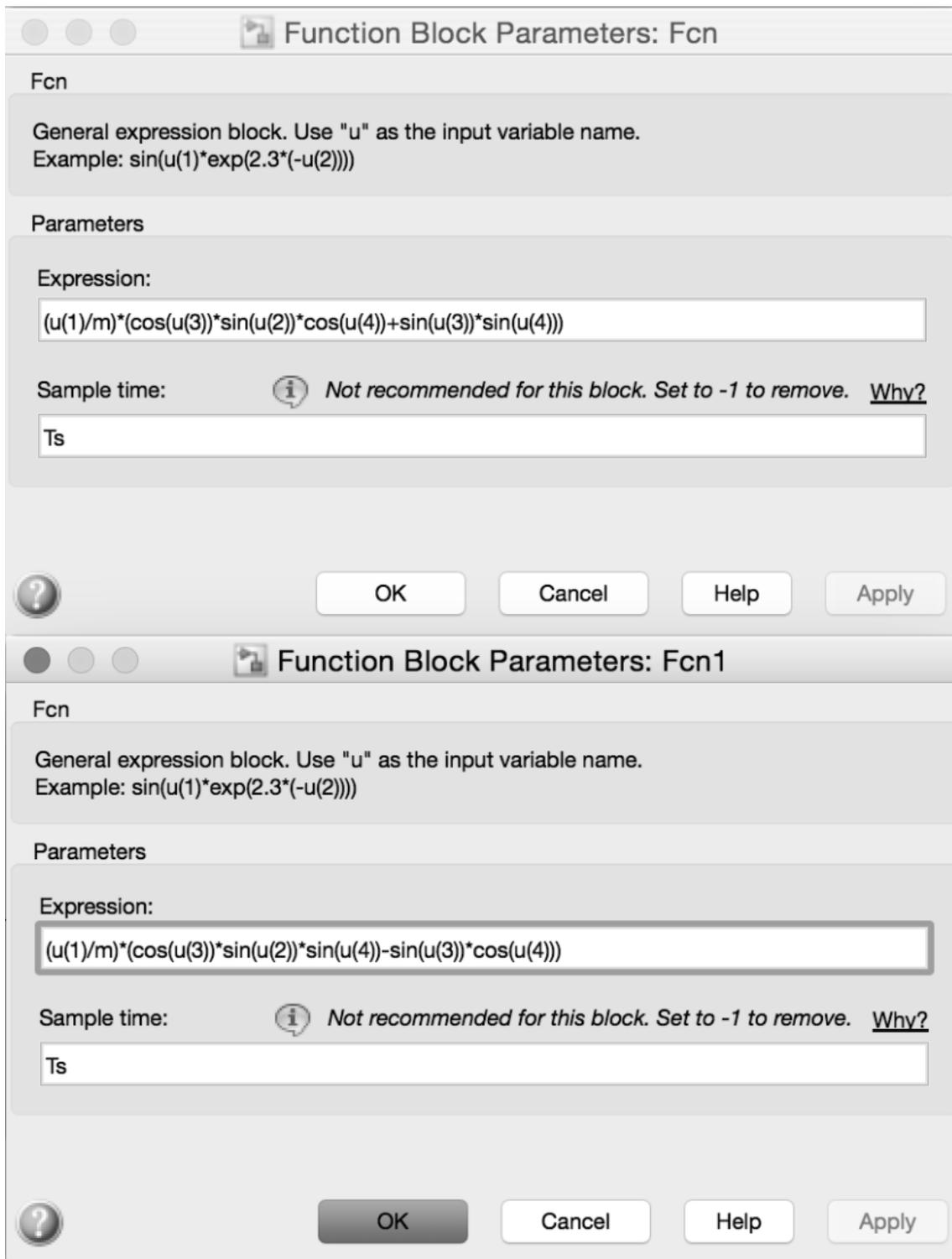


Figure 5.4: Setting function blocks in Fig. 5.3.

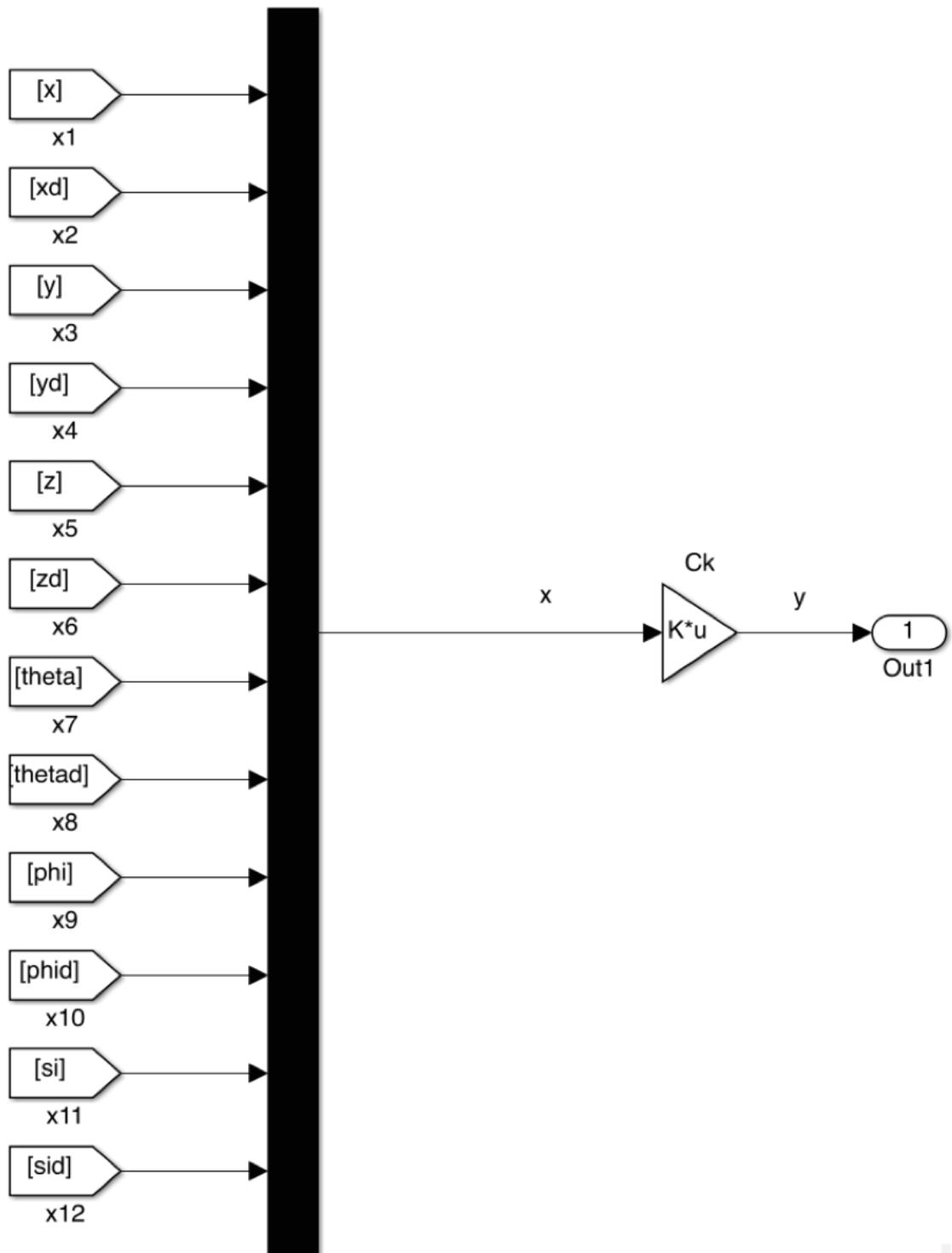


Figure 5.5: Transferring state vector to output vector in Simulink.

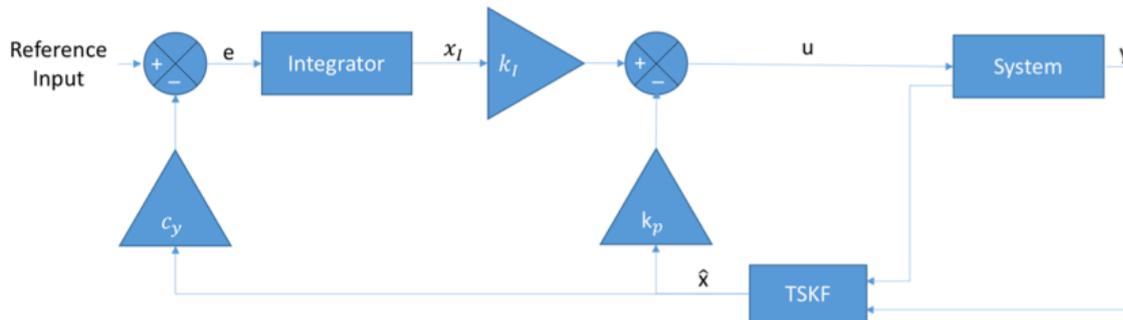


Figure 5.6: The structure of the system with reference follower controller (two feedback loops).

The obvious difference between reference follower controller and previous controller is that the latter one does not guarantee to follow the reference inputs. This will be a major problem when actuator fault happens and is discussed in chapter 7. Also, another important point is that, when nonlinear model is used, violating the linearization conditions is probable, therefore, for LQR controller with one feedback loop its not possible to control the system and finally the system output will be unstable. Therefore, if nonlinear model is used, controller must be reference follower, unless, system will be unstable as in nonlinear model, the system response makes a lot of sudden movements which result unstable response.

5.4 Actual System Simulation Results

By combining sections 5.1, 5.2 and 5.3 the overall structure of the simulated system is revealed in Fig. 5.7. Note that, the dimensions of output vector of nonlinear system block are 6x1 which means that only position and orientation parameters are measured. In addition, measurement noise is created by Gaussian Noise Generator block with different initial seed for each sensor. TSKF block gets 4 control inputs and 6 measured outputs and gives 12 estimated states which is fed back to LQR gain matrix. LQR controller takes estimated states, and together with reference inputs, creates control inputs to the actuators. However, in this structure actuators block does not exist because as mentioned in section 2.1, transfer function of the actuators are approximated as a zero-order transfer function in (2.2) and this equation is included in state-space equations. This means that inputs to system block in Fig. 5.7 are PWM inputs to actuators. Also, covariance matrices for Gaussian noises are as follows.

$$Q_k = \text{diag}(10^{-6} [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 10^{-6} \ 10^{-6} \ 10^{-6} \ 10^{-6} \ 10^{-6} \ 10^{-6}])$$
(5.7)

$$R_k = \text{diag} (10^{-4} [4 \ 4 \ 4 \ 1 \ 1 \ 1])$$
(5.8)

$$Q_k^y = 10^{-3} I_{4 \times 4}$$
(5.9)

Let's examine how Kalman filter estimates the states using control input and measured output vectors. By driving input vector in (2.10), system response together with Kalman filter estimations are plotted in Fig. 5.8. By examining system response, we can see that TSKF block works well in estimating 12 states and controller does its job in following reference input, however, in comparison with the responses in section 4.2, there are some fluctuations around reference input. This is normal as nonlinear model of the system is used and controller inputs are Kalman filter estimations, which estimates the states using linearized state-space model of the system around the nominal point. These oscillations are an alarm for us, that reminds us to not to violate linearization conditions as discussed in section 2.3. For instance, if we chose $u_\psi \geq \frac{\pi}{6}$ the system response will be unstable, because, yaw angle will move far away from its nominal point ($\psi \approx 0$), therefore, the linearized state space definition of the system in (2.11) which is used in TSKF block will no longer be reliable, accordingly, Kalman cannot estimate states correctly which are fed back to the controller and finally, system will be unstable.

The important point in this subject is that, in chapter 3, LQR gain matrix is designed such that system be stable and over-damped, so, system response reaches the reference value slowly without overshoot. For example, it takes almost 5 seconds to quadrotor to reach reference value in y direction in Fig. 5.8. This causes quadrotor moves slowly and consequently velocity of quadrotor in any direction be small, therefore, linearization conditions are not violated. This means that, if we design a controller that makes the quadrotor moves fast (e.g. small rising time), violating linearization conditions becomes probable.

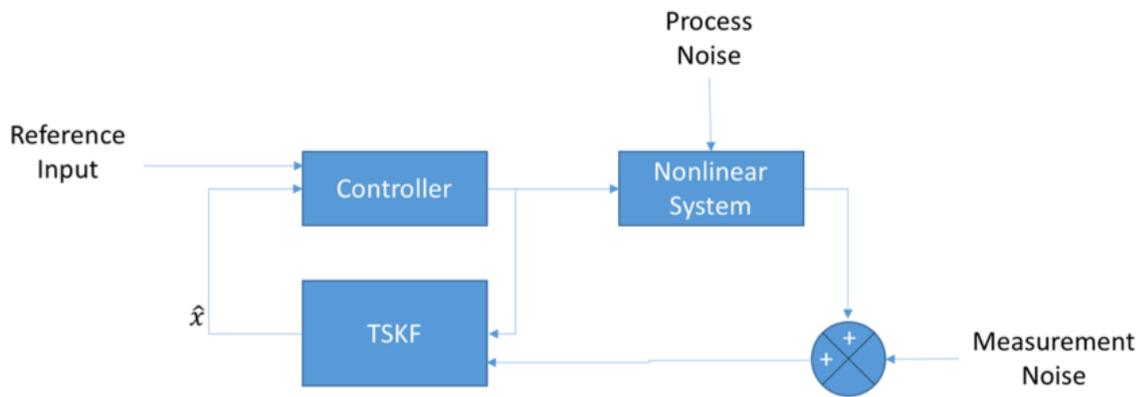


Figure 5.7

Figure 5.7: Overall structure of the simulated nonlinear system with actual number of sensors in.

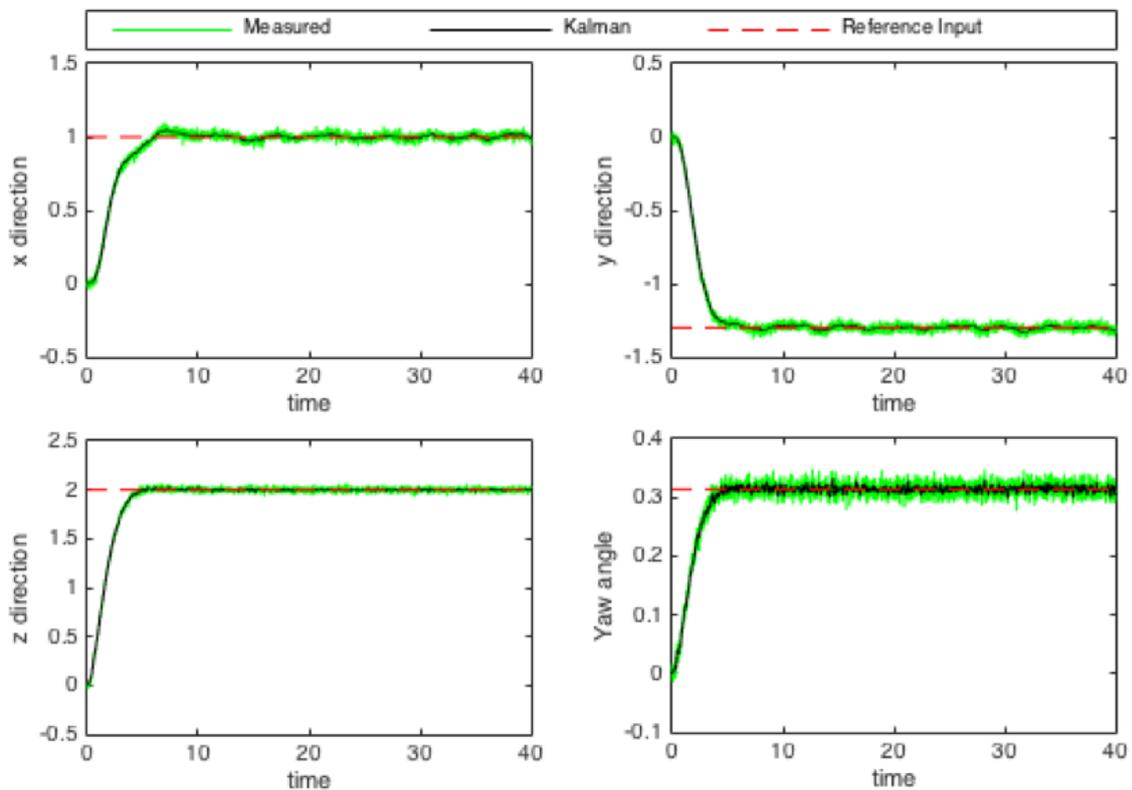


Figure 5.8: System in Fig. 5.7 responses to reference inputs.

6. ACTUTOR FAULT

Loss of control effectiveness in the actuators is the most common type of faults happening to the quadrotors' actuators. As discussed in chapter 2, each motor generates an upright thrust while working. Let us rewrite the equation (2.2) which is the approximated zero-order transfer function of each actuator

$$T_i \approx K u_i; i = 1, \dots, 4 \quad (6.1)$$

The generated thrust (T_i) by the faulty actuator is decreased by the percentage of its fault. As above equation suggests, this reduction can be simulated by decreasing the produced PWM (u_i) by the same percentage. As mentioned in section 4.1, γ_i is the percent of loss of control effectiveness in the i^{th} actuator. If u_{if} be the PWM input to the i^{th} faulty actuator, then

$$u_{if} = u_i - \gamma_i u_i = u_i (1 - \gamma_i) \quad (6.2)$$

by doing some simple mathematical simplifications, for all four actuators we have

$$\begin{bmatrix} u_1 - \gamma_1 u_1 \\ u_2 - \gamma_2 u_2 \\ u_3 - \gamma_3 u_3 \\ u_4 - \gamma_4 u_4 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} - \begin{bmatrix} \gamma_1 u_1 \\ \gamma_2 u_2 \\ \gamma_3 u_3 \\ \gamma_4 u_4 \end{bmatrix} = \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} - \begin{bmatrix} u_1 & 0 & 0 & 0 \\ 0 & u_2 & 0 & 0 \\ 0 & 0 & u_3 & 0 \\ 0 & 0 & 0 & u_4 \end{bmatrix} \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \\ \gamma_4 \end{bmatrix} = u_{-k} - U_k \gamma_{-k} \quad (6.3)$$

Now, state-space representation of the system with faulty actuators become

$$\dot{x}_{-k+1} = A_k x_{-k} + B_k u_{-kf} \quad (6.4)$$

where u_{-kf} is the faulty input vector calculated in (6.3),

$$u_{-kf} = u_{-k} - U_k \gamma_{-k} \quad (6.5)$$

therefore, state-space equation become,

$$x_{-k+1} = A_k x_{-k} + B_k u_{-k} - B_k U_k \gamma_{-k} \quad (6.6)$$

which is used in (4.1).

The implementation of actuators loss of effectiveness in Simulink is a pretty simple task. As (6.2) suggests, multiplying the control input vector by loss of effectiveness vector γ_{-k} , simulates the actuators' fault in MATLAB Simulink.

6.1 The Impact of Actuators Faults

The impact of the actuators faults can be discussed by plotting the system responses and Kalman filter estimations in faulty system using Simulink. If an actuator in a quadrotor is faulty, the thrust produced by the corresponding actuator is reduced by the percentage of fault, this is called “loss of control effectiveness” in an actuator. The overall structure of the faulty system in Simulink is drawn in Fig. 6.1. Note that in this diagram the controller is not the reference follower and therefore the linear system model is used (this is discussed on section 5.3) and the covariance matrices for Gaussian noises are as follows

$$Q_k = \text{diag}(10^{-6} [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 10^{-6} \ 10^{-6} \ 10^{-6} \ 10^{-6} \ 10^{-6} \ 10^{-6}]) \quad (6.7)$$

$$R_k = \text{diag}(10^{-12} [10^6 \ 1 \ 10^6 \ 1 \ 10^6 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 1]) \quad (6.8)$$

$$Q_k^\gamma = 10^{-3} I_{4 \times 4} \quad (6.9)$$

As a first faulty scenario, let us assume that there is a 30% loss of effectiveness in actuator 1 at 15 (s) of the flight. The impact of this fault on the system response is shown on Fig. 6.2. As the faulty actuator is located on the x direction (Fig. 2.1) the loss of control effectiveness on this actuator caused a sudden change on pitch angle (Fig. 6.3), consequently, a sudden reduction on x direction is happened. Note that, the quadrotor hit the ground after fault and a small change on ψ is unavoidable. This faulty scenario reveals the actuators' loss of effectiveness as a crisis when only one feedback loop for controller is used (Fig. 6.1). By plotting the PWM input (control

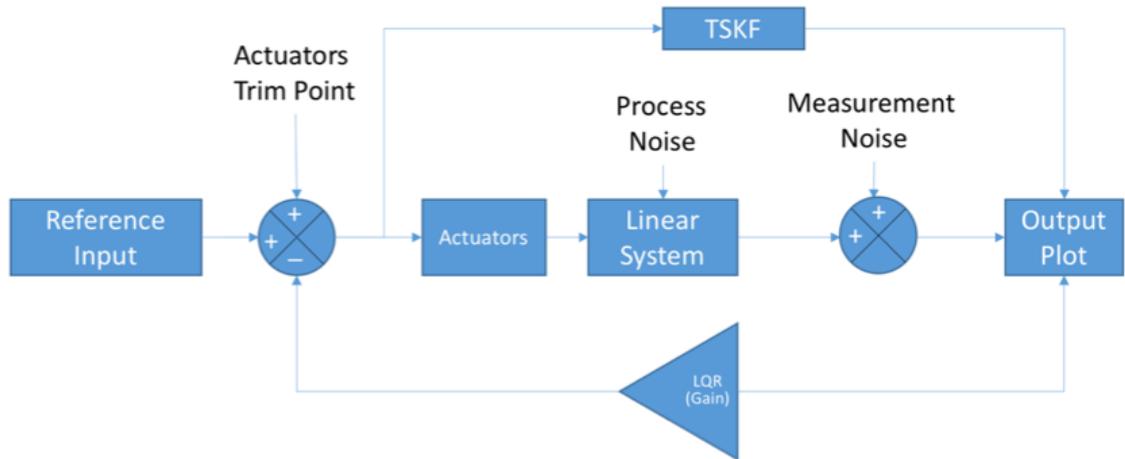


Figure 6.1: The overall structure of the linear system with faulty actuators in Simulink.

input) to actuators in Fig. 6.4 we can see that when a fault is occurred in actuator 1, in order to compensate the fault, the controller counteracts the fault by increasing the PWM input to the corresponding actuator. This compensation resulted in Fig. 6.5 in which the reduced thrust (produced by faulty motor) came back to pre-fault value. Also remaining actuators' thrusts are decreased by the controller to make the quadrotor stable while hovering.

As a second faulty scenario, we assume that there are 40% and 50% faults on actuators 2 and 3 at 15 (s) and 30 (s) respectively. The position and orientation of the quadrotor, control inputs produced by the controller and thrusts created by motors are plotted in Fig. 6.6. to Fig. 6.9. Again, it can be seen that how quadrotor gets away from the reference position after actuators faults. These two fault scenarios stimulate us to deal with this problem. In the upcoming section a solution to this problem is presented and the results are discussed.

6.2 Actuators Faults and Leader-Follower Controller

By examining the system response to the reference input when actuators loss of effectiveness exists, being actuators faults as a serious problem is concluded. In previous section, linear system mode was used and controller was the LQR controller (one loop feedback) calculated in chapter 3. As discussed, the actuators faults result in deviations from reference values on system response, therefore, using reference follower controller (two feedback loop) will be helpful to deal with this problem. The implementation of this controller is presented in section 5.3. By taking advantage of

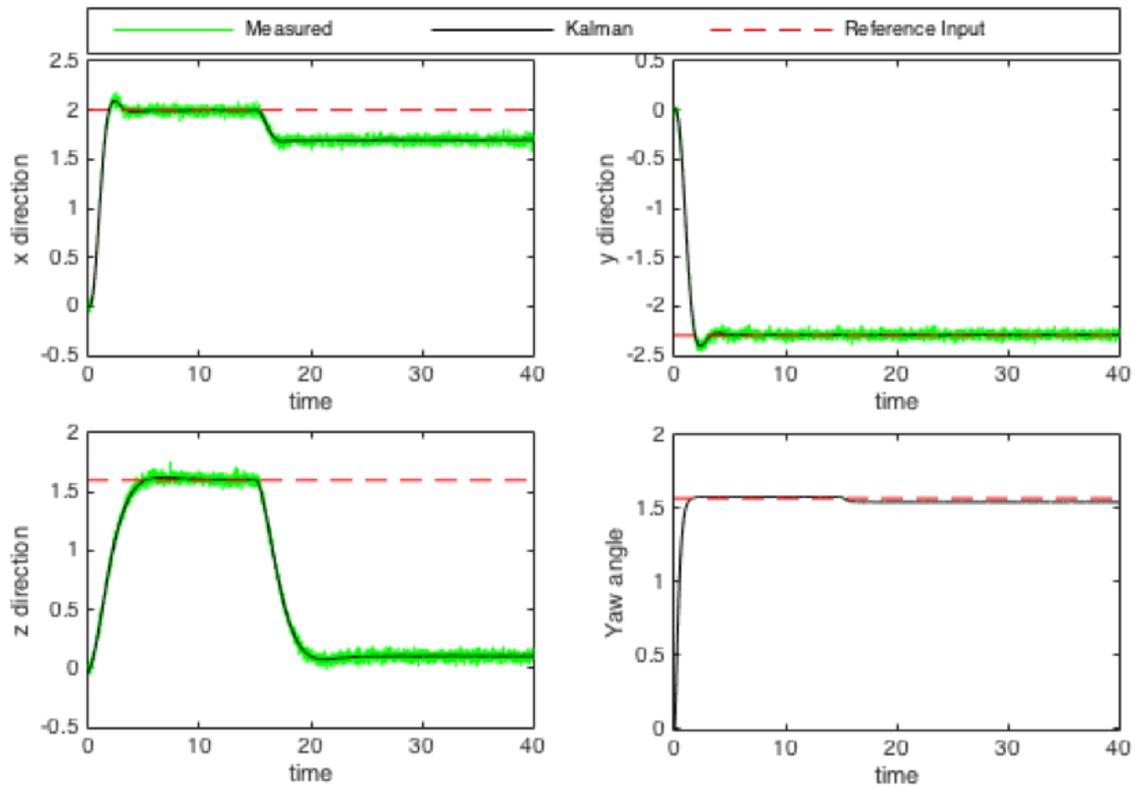


Figure 6.2: The impact of 30% fault in actuator 1 on system response.

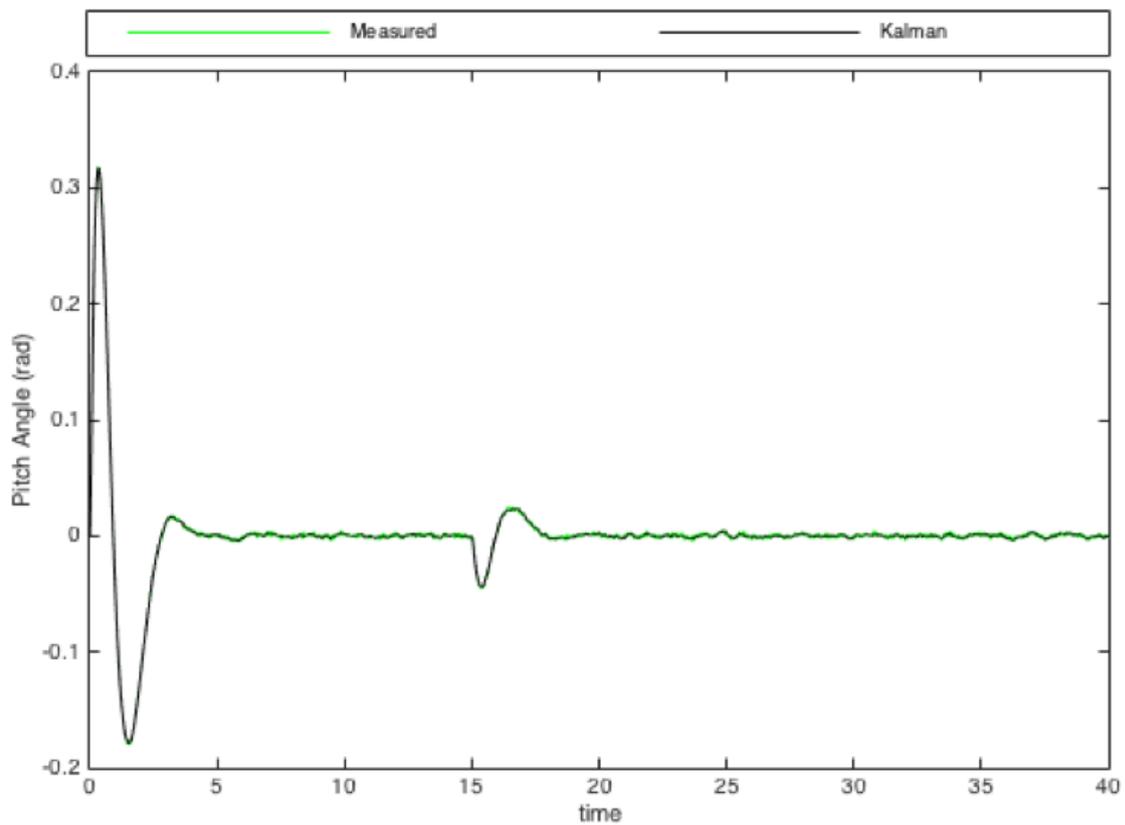


Figure 6.3: The effect of fault in actuator 1 on pitch angle.

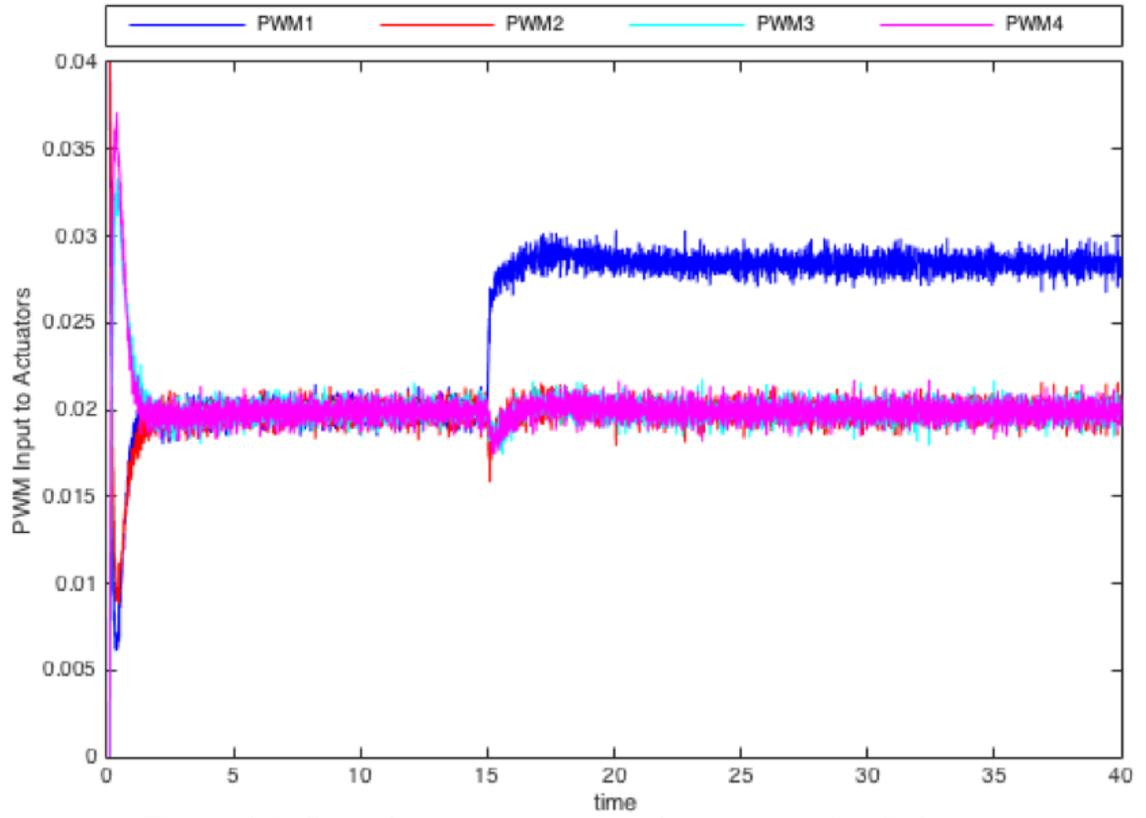


Figure 6.4: Control input to actuators when actuator 1 is faulty.

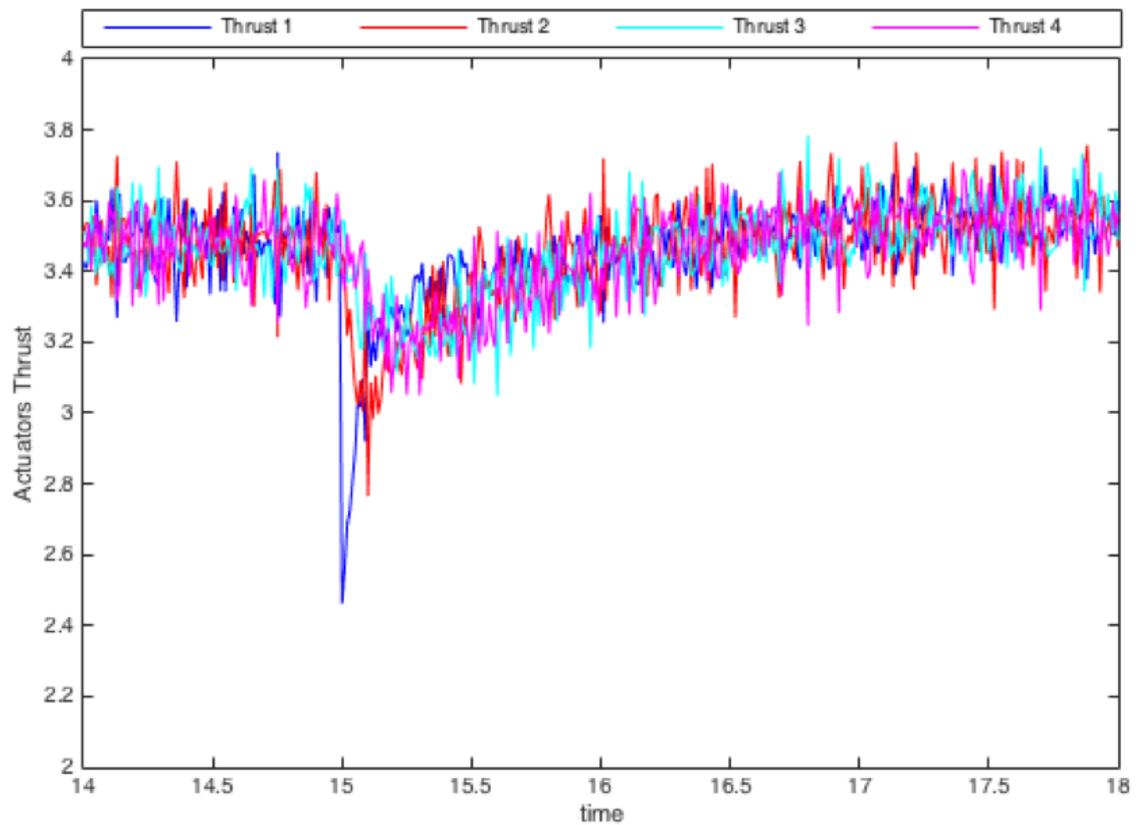


Figure 6.5: Thrust produced by actuators when actuator 1 is faulty.

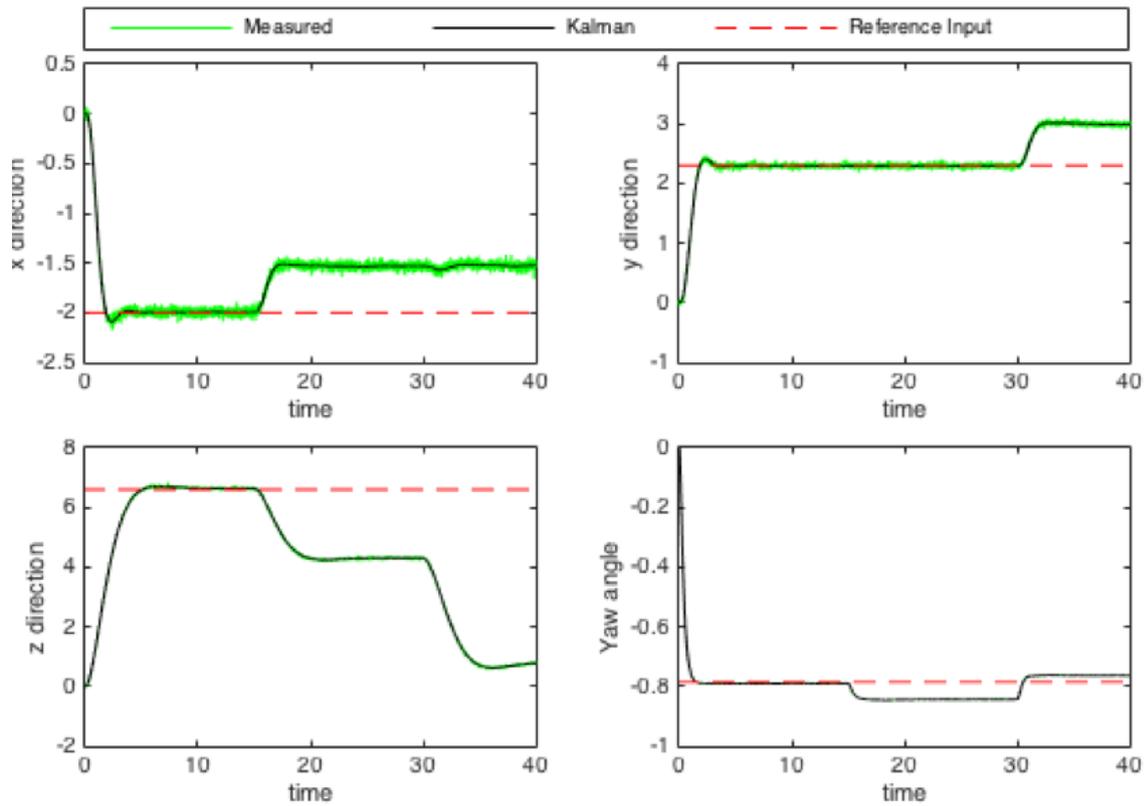


Figure 6.6: The impact of faults in actuators 2 and 3 on system response.

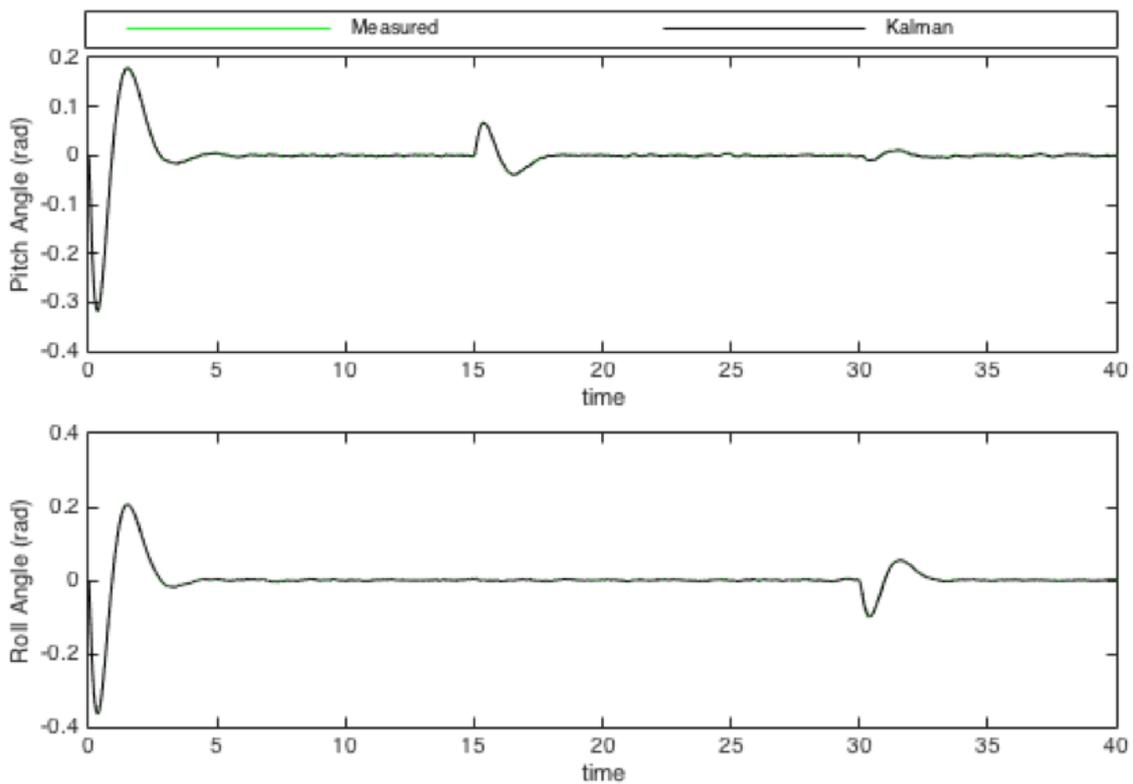


Figure 6.7: The effect of faults in actuators 2 and 3 on pitch and roll angles.

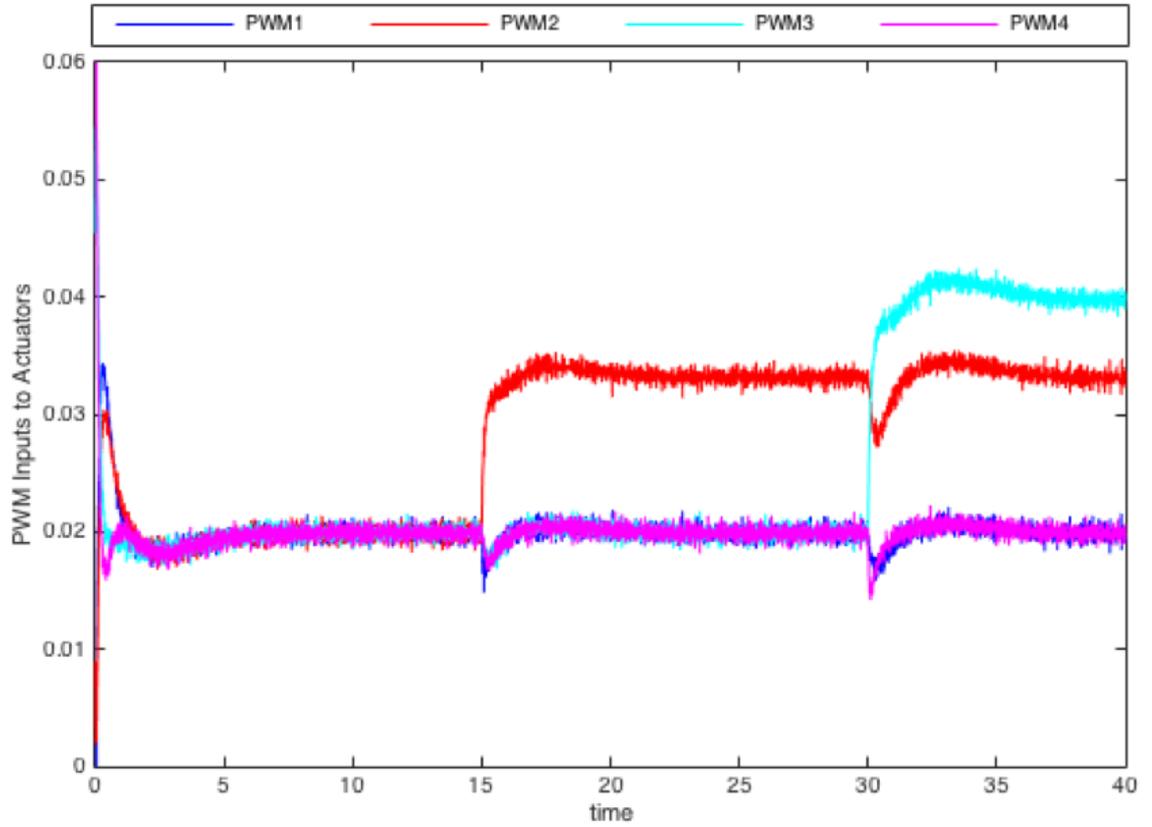


Figure 6.8: Control input to actuators when actuators 2 and 3 are faulty.

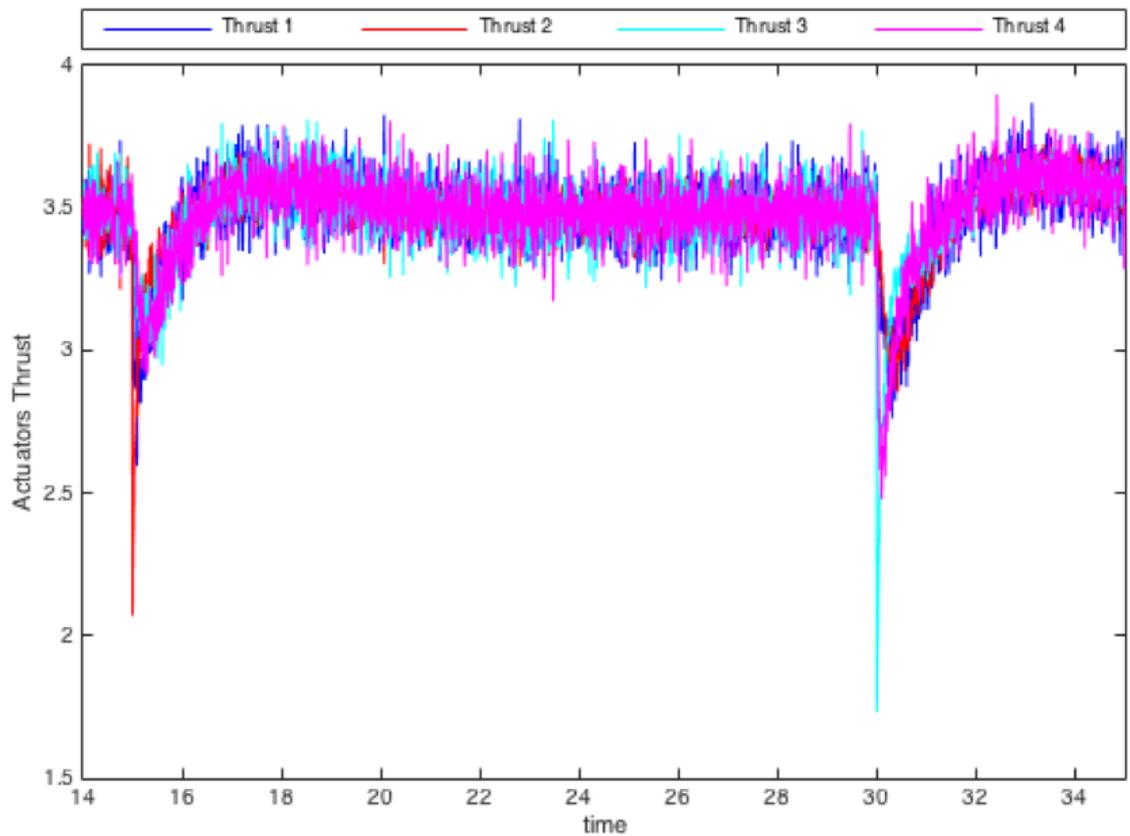


Figure 6.9: Thrust produced by actuators when actuator 2 and 3 are faulty.

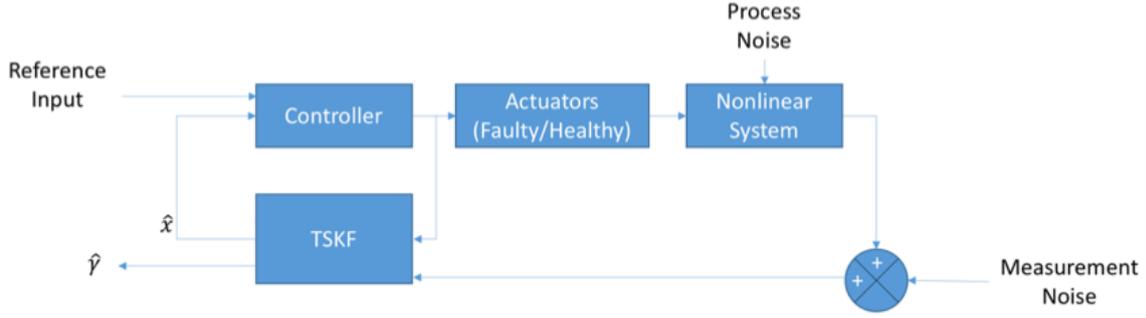


Figure 6.10: The overall structure of the nonlinear reference follower system with faulty actuators in Simulink.

this controller and using nonlinear model of the system instead of the linear model to challenge the controller performance, the overall system diagram in Simulink is presented in Fig. 6.10. Note that, $\hat{\gamma}$ is the estimated γ vector and faults on actuators occurs the same way in previous section.

In third faulty flight scenario, let us give the same faults as in second scenario. Therefore, there are 40% and 50% faults on actuators 2 and 3 at 15 (s) and 30 (s) respectively. In addition, covariance matrices for Gaussian noise generators are as follows

$$Q_k = \text{diag}(10^{-6} [1 \ 1 \ 1 \ 1 \ 1 \ 1 \ 10^{-6} \ 10^{-6} \ 10^{-6} \ 10^{-6} \ 10^{-6} \ 10^{-6}])$$
(6.10)

$$R_k = \text{diag} (10^{-4} [4 \ 4 \ 4 \ 1 \ 1 \ 1])$$
(6.11)

$$Q_k^\gamma = 10^{-3} I_{4 \times 4}$$
(6.12)

The system response in position and orientation of the quadrotor along with Kalman filter's state estimations are plotted in Fig. 6.11 and Fig. 6.12. By comparing these results with the results in section 6.1 we can see the major difference between one feedback loop controller with reference follower controller and two feedback loops. The system is able to deal with these faults on actuators. No more ground hit and deviations from reference values happened. The faults effects on x and y direction at 15 (s) and 30 (s) can be seen in Fig. 6.11 but these effect are too small to be taken into account and can be ignored. However, if loss of effectiveness in actuators and their consequences be too great to be ignored then actuators' fault will be important problem even in system with reference follower controller. This subject is discussed in next section.

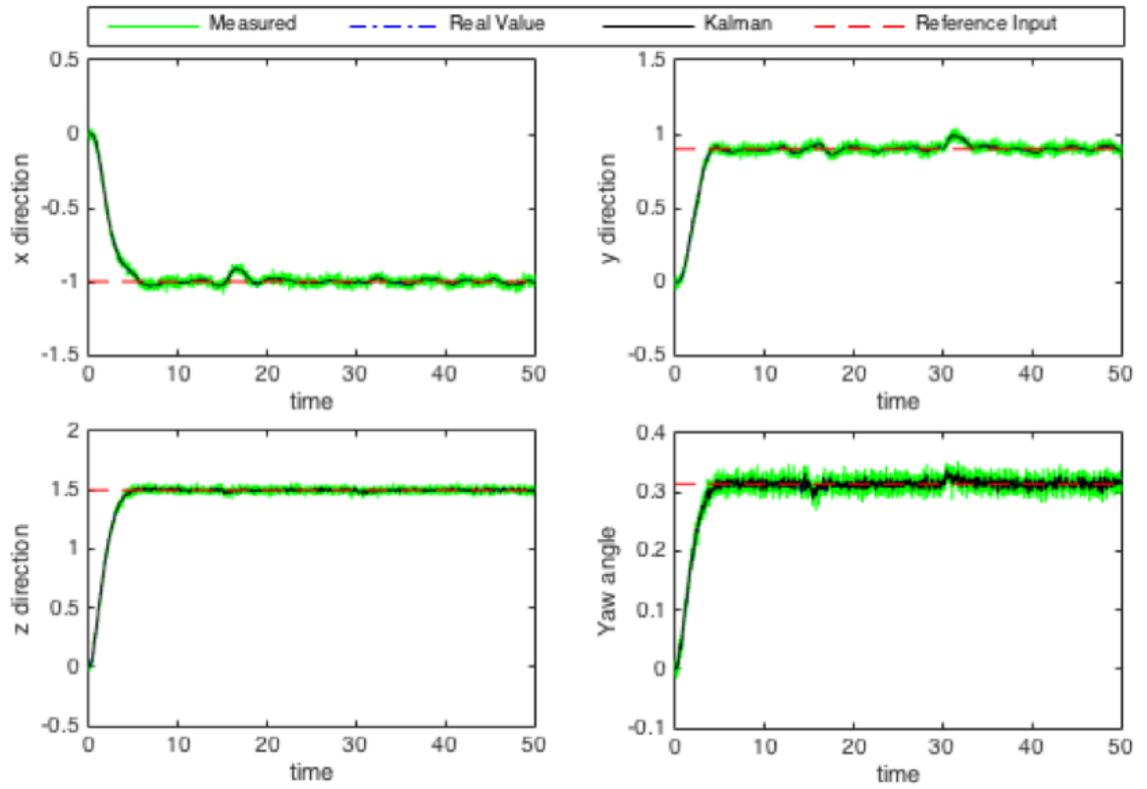


Figure 6.11: Response of the system with faulty actuators and reference follower controller.

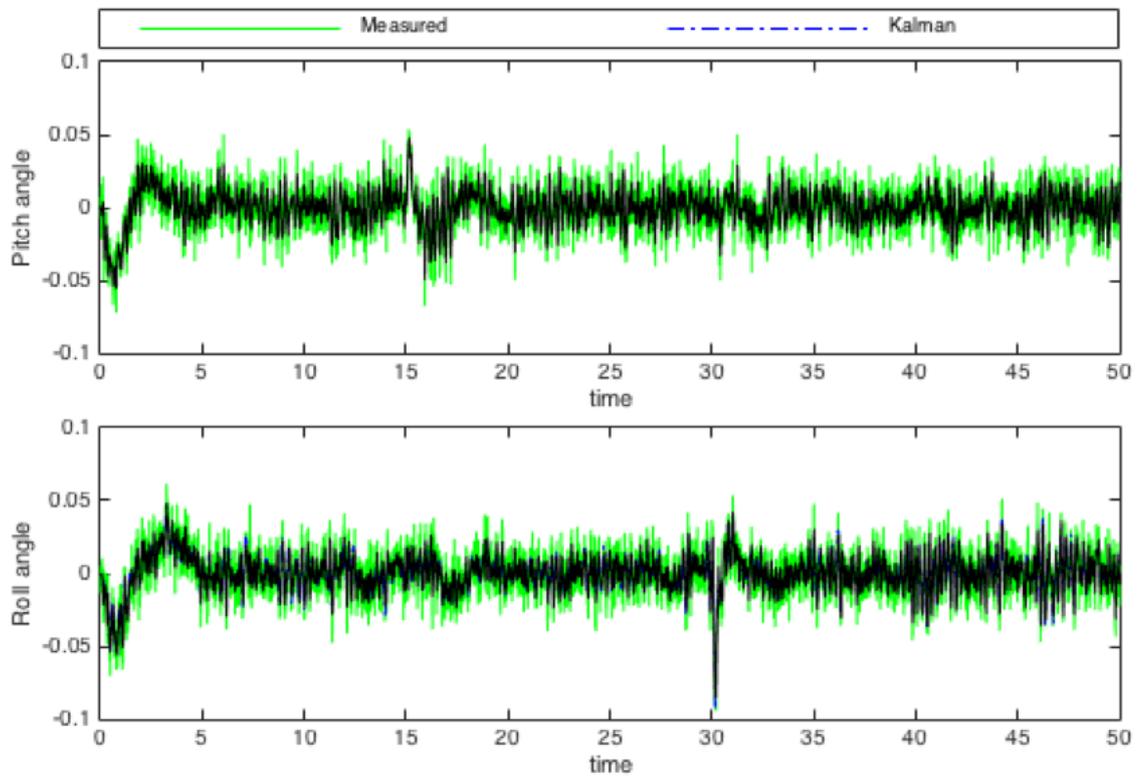


Figure 6.12: Pitch and roll angles of the system with faulty actuators and reference follower controller.

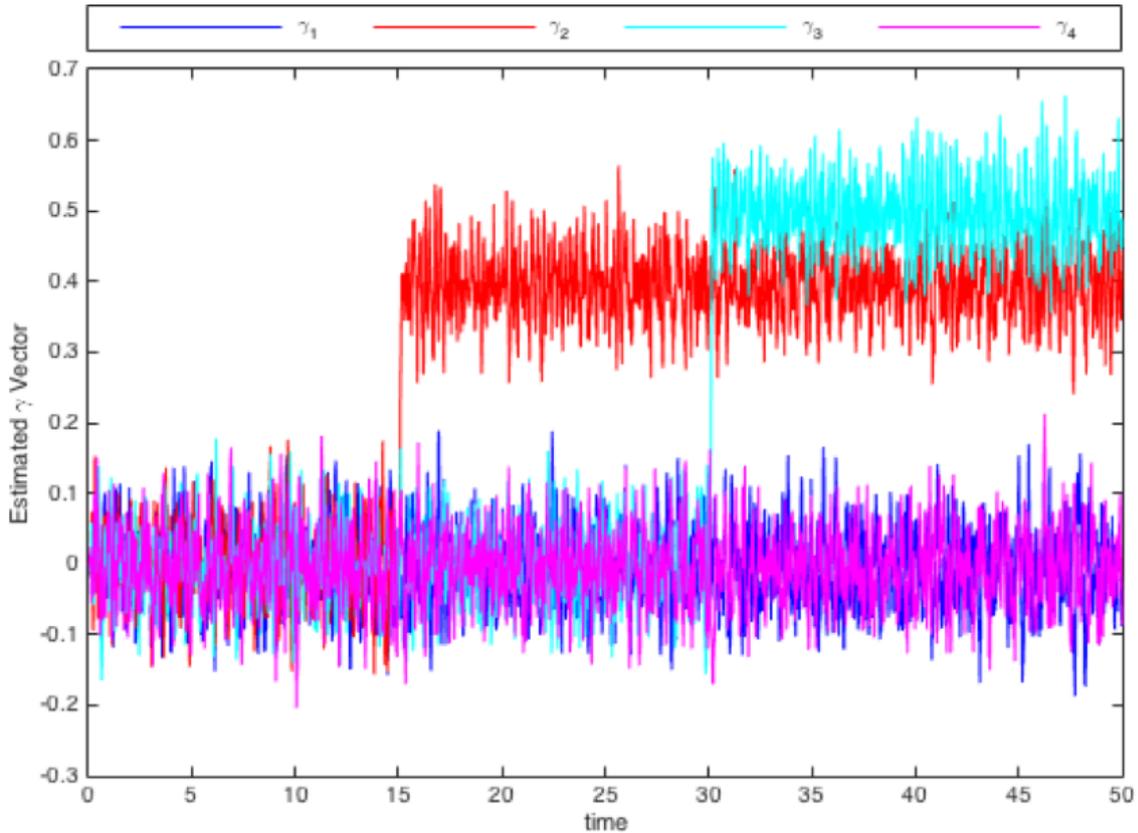


Figure 6.13: TSKF estimation of the actuators faults.

6.3 Actuator Fault Detection and Diagnosis

Faults that are occurred in actuators are assumed as loss of control effectiveness. The percent of fault describes the percent of thrust reduction created by each motor. This is discussed in first part of this section. The amount of fault existing on an actuator is described by the corresponding element in γ vector. Two-Stage Kalman Filter estimates the γ vector ($\hat{\gamma}$) along with the system states. The formulation of this is presented in section 4.1. When a bias fault occurs in an element(s) of the γ vector, using TSKF the estimated $\hat{\gamma}$ vector reveals the time, position and magnitude of the occurred fault. This is the reason why we call this procedure as FDD. By defining a threshold for $\hat{\gamma}$, the detection can be done when an element of $\hat{\gamma}$ crosses the corresponding threshold, also, the magnitude of the $\hat{\gamma}$ is equal to the percent of the actuator's fault. The system with nonlinear model of the plant, reference follower controller, actuators fault and TSKF block is shown on Fig. 6.10. By plotting the $\hat{\gamma}$ vector estimated by TSKF block in third fault scenario on section 6.2 the TSKF ability in FDD of the actuators is proved (Fig. 6.13).

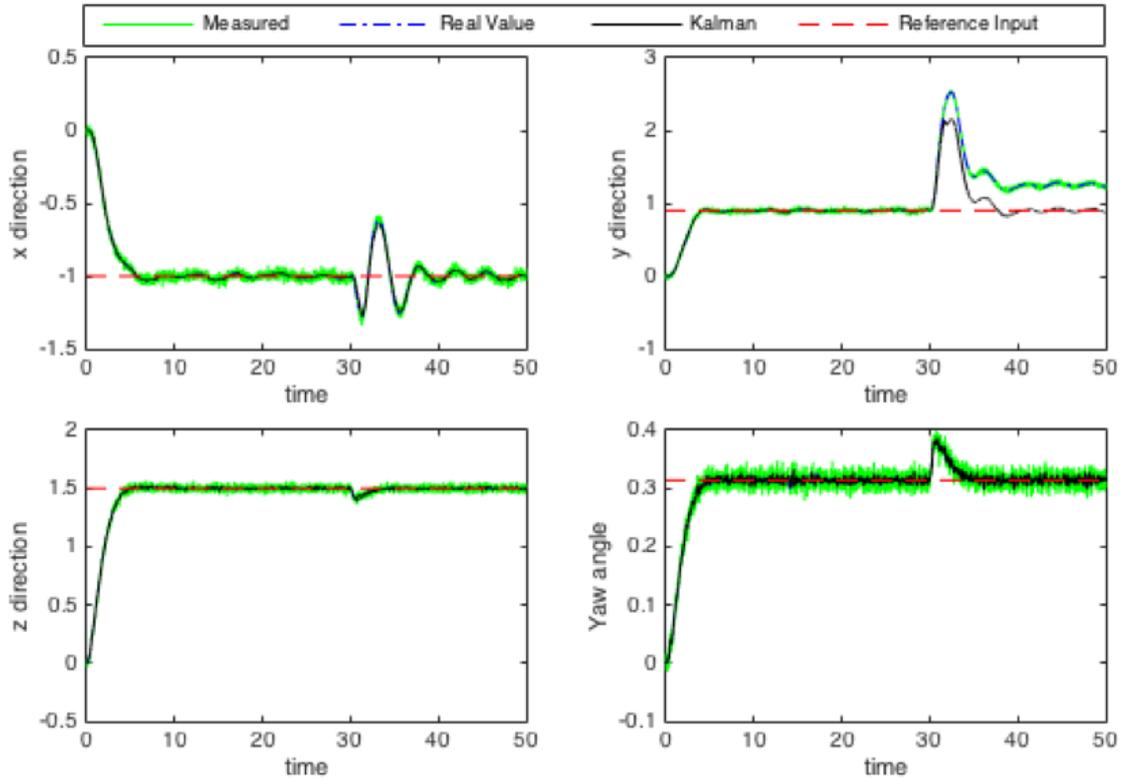


Figure 6.14: The impact of 75% fault on actuator 3 on system response.

By studying the system response in this scenario, a question arises in our minds. If the system is able to deal with the actuators faults using reference follower controller, why we need to detect and diagnose the actuators faults? To answer this question, we need to run a flight scenario with high percentage of actuator fault. As fourth faulty scenario, we assume that there is 75% fault on actuator 3 at 15 (s). The impact of this fault on system position and the orientation is plotted in Fig. 6.14. and Fig. 15. It's easy to conclude that, in contrast with previous flight scenario, reference follower controller is no longer reliable for high percentage of faults. System deviated from its reference position on y direction and made some oscillations in x direction when fault occurred. Kalman filter estimation is no longer equal to the measured value for y direction. Altogether, prove the need to counteract the actuator fault, to this end, FDD for actuators' faults is required.

Now, another question that arises is, why the reference follower controller was not able to counteract the high percent actuator fault? To reply this question let us plot the estimated and real value of the acceleration toward y direction in Fig. 6.16. It can be seen that, after fault occurs, Kalman filter cannot estimate this state accurately. The reason behind this is obvious, in the fault occurrence time controller produces great

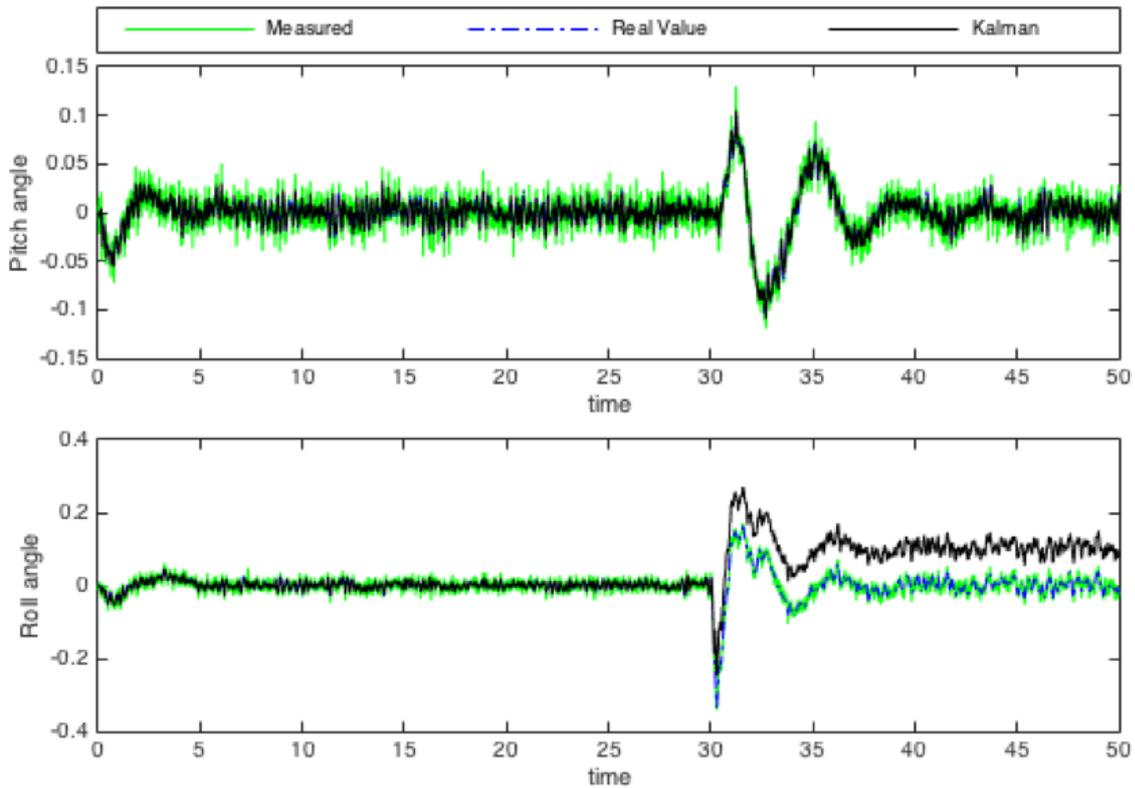


Figure 6.15: The impact of 75% fault on actuator 1 in pitch and roll angles.

control input in y direction to counteract the fault on actuator 3, therefore, acceleration in y direction becomes a big value and crosses the red line for linearization condition, as Kalman filter uses the linearized model of the system around the nominal point, after violating this condition, Kalman is not able to estimate accurately. The same reason holds true for the oscillations on x direction after fault. Let us see if TSKF was able to estimate γ vector. Fig. 6.17 proves that 75% loss of control effectiveness on actuator 3 is detected and diagnosed by TSKF. A future work for this is planned to counteract the consequence of actuator fault on system response by changing the controller gain matrix (reconfiguration) with respect to the size and position of the fault. This problem is called Active Fault Tolerant Control (AFTC) on actuators.

For the actuator fault detection and diagnosis, let us run the fifth faulty scenario in which, respectively, 15%, 25%, 70%, and 90% faults on actuators 1 to 4 at 15, 18, 18, and 25 (s) are happened. The impact of faults on system response and the estimated γ vector by TSKF are plotted in Fig. 6.18 to 6.20.

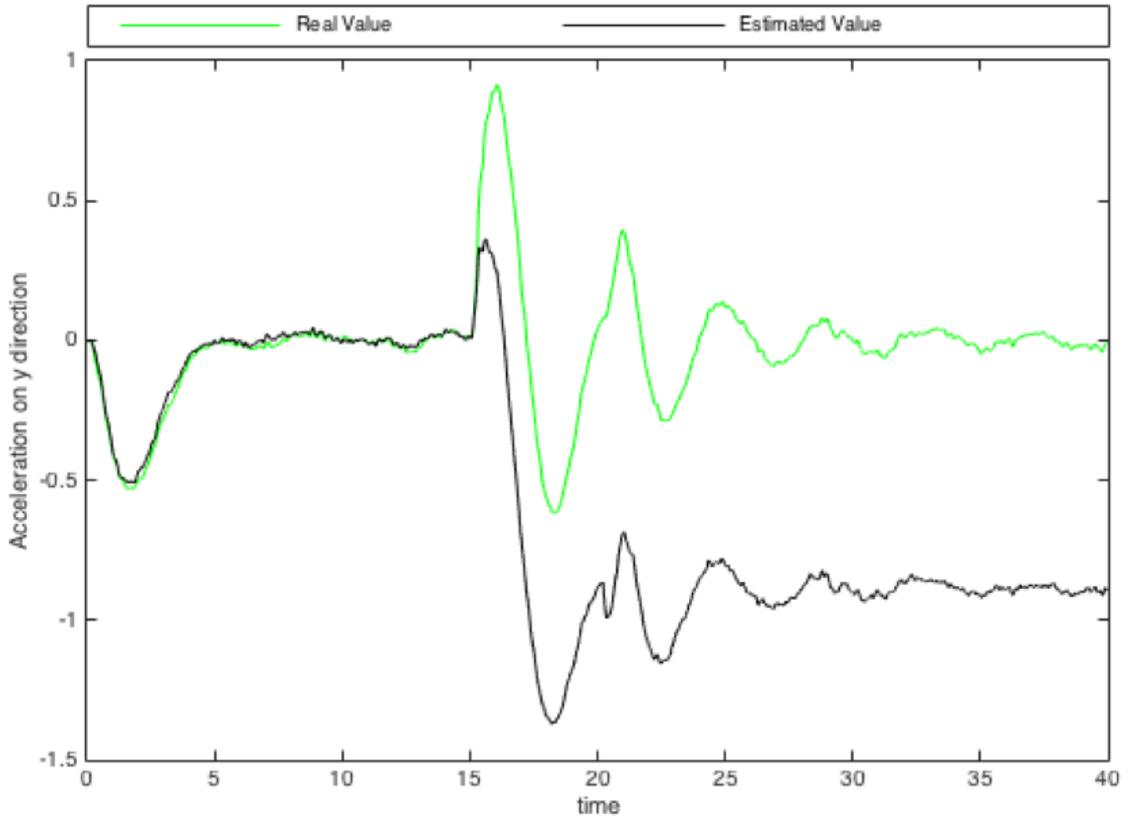


Figure 6.16: The acceleration toward y direction when actuator 3 is 75% faulty.

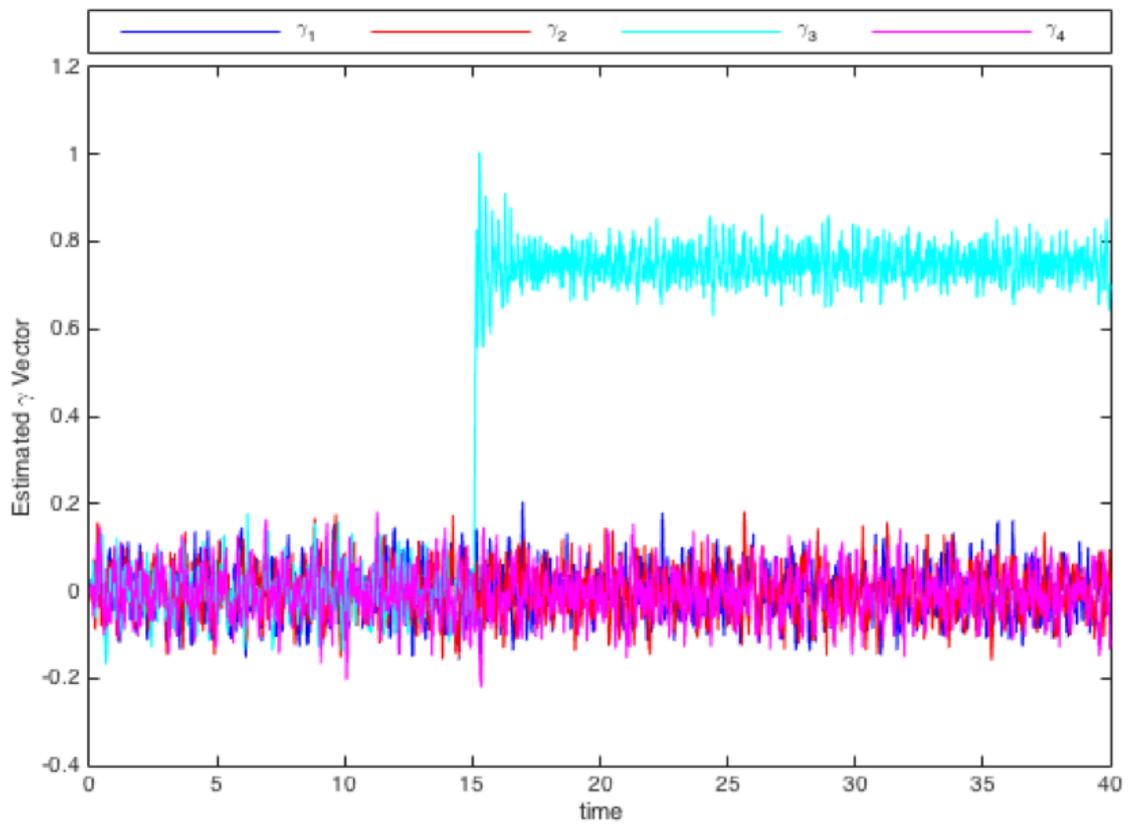


Figure 6.17: The estimated actuators fault by TSKF when actuator 3 is faulty.

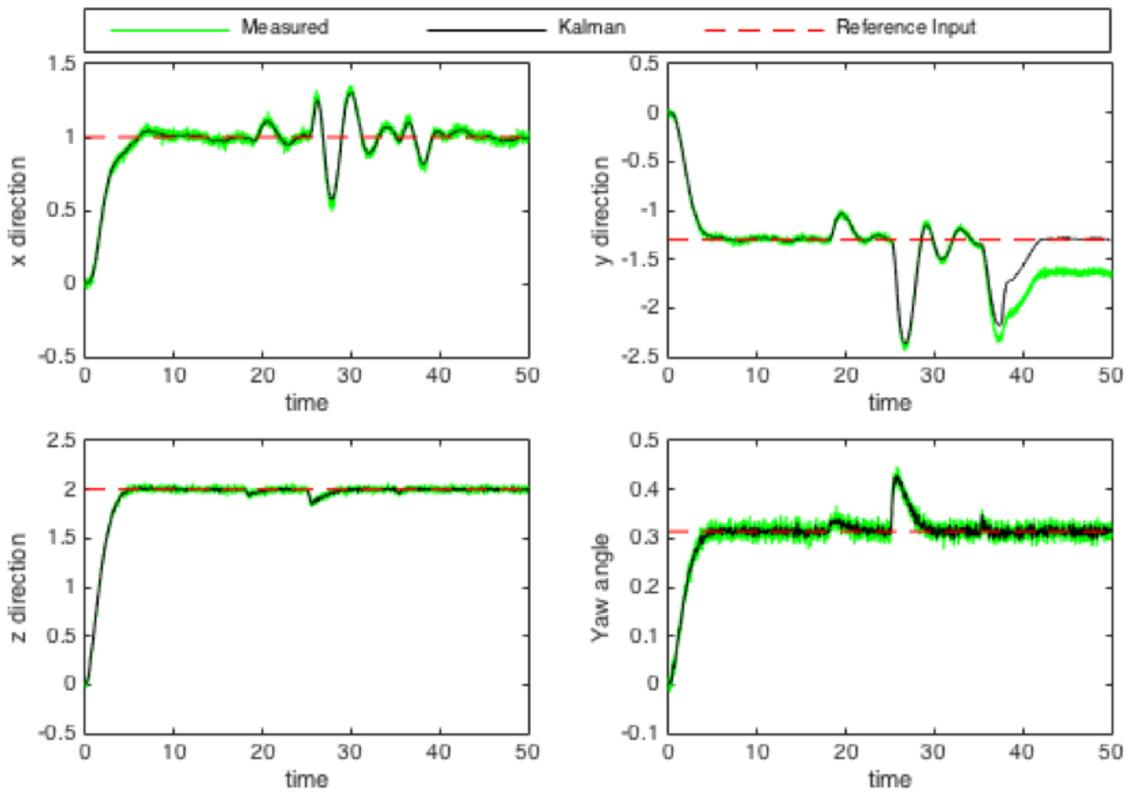


Figure 6.18: The impact of faults on all actuators on system response.

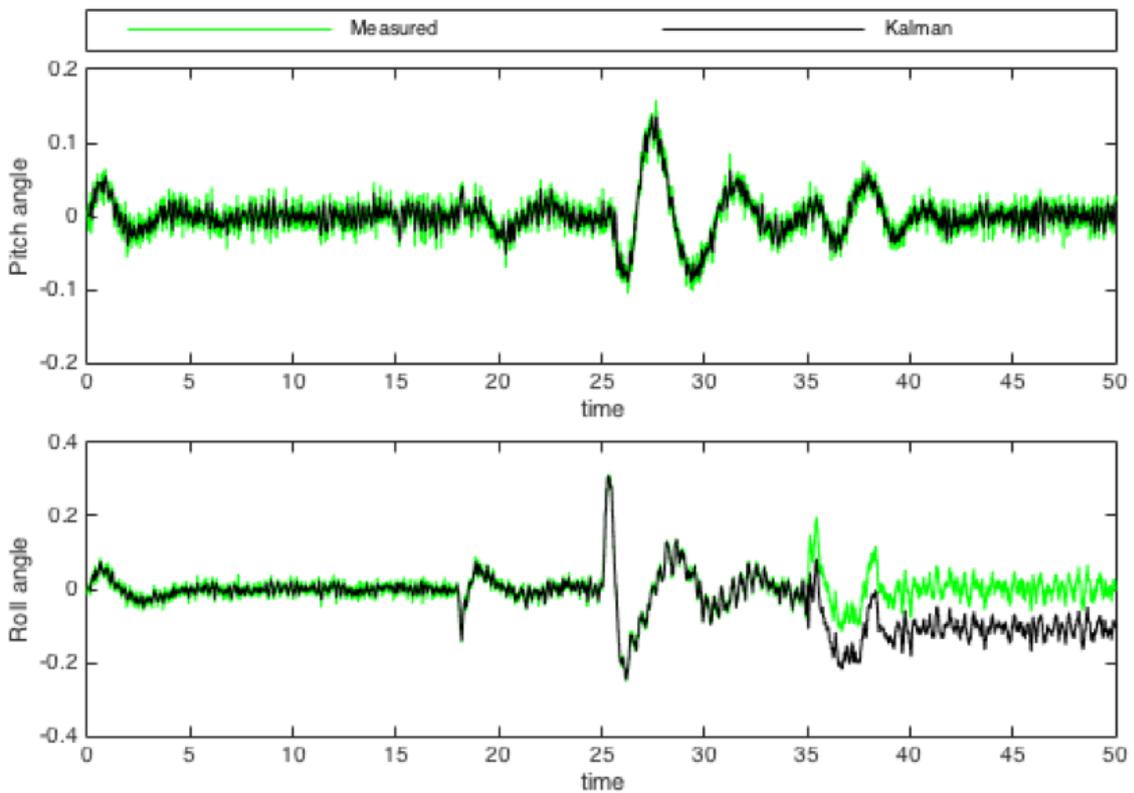


Figure 6.19: Pitch and roll angles when all actuators are faulty.

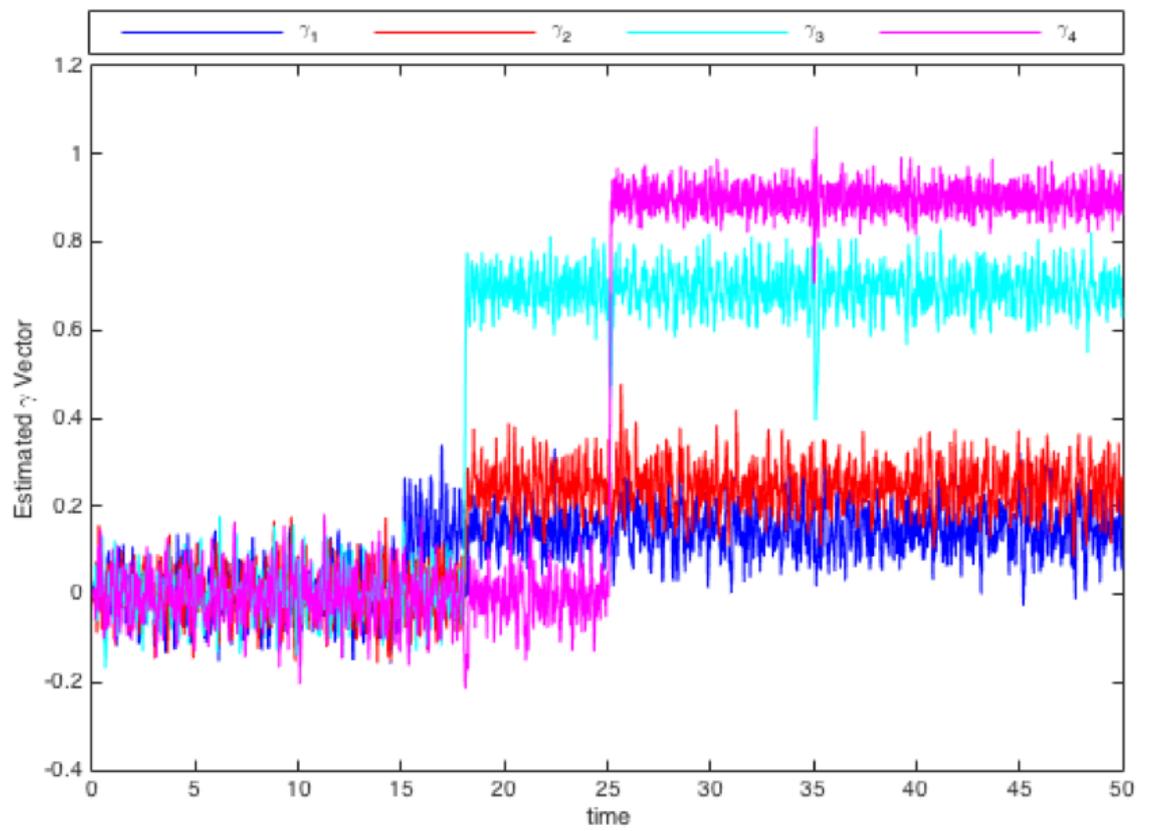


Figure 6.20: The estimated actuators fault by TSKF when all actuators are faulty.

7. SENSOR FAULT

Small UAVs are often equipped with low-cost and lightweight micro-electro-mechanical systems (MEMS) inertial measurement units including 3-axis gyro, accelerometer and magnetometer. The measurements provided by these sensors often suffer from bias and excessive noise as a result of temperature variations, vibration, etc. As these sensors serve an essential role in most quadrotor control schemes, therefore, the detection and estimation of sensor faults plays a crucial role in the safe operations of quadrotors. There are some types of sensor faults that are considered in some studies (mentioned in Introduction), but as a result of abundance of bias faults in low-cost sensors, in this dissertation, sensor faults are considered as bias faults and occur suddenly at a specific time and remain till the end of scenario.

7.1 Sensor Fault detection and Diagnosis

As discussed in section 5.1, we assume that there are three position and three orientation sensors measuring quadrotor position in Cartesian coordinate and orientation using Euler angles. As we consider the bias faults on sensors, implementation of these faults is an easy process. The measured value by the faulty sensor is summed with a step function which starts at the time of fault and its magnitude is equal to the fault size. The measured faulty/healthy values together with control inputs are fed back to TSKF block, therefore, the state vector is estimated and fed back to the controller (reference follower controller). The structure of the simulated system with faulty sensors is shown in Fig 7.1. Note that, actuators block considered in previous section also exists here and \tilde{r} is the residual vector calculated in chapter 4. As a reminder, residuals are the difference between measured value by sensors and estimated value by Kalman filter. As Kalman filter estimates the states using linear model of the system together with control inputs and measured values, if any sudden deviation will happen in measured value, Kalman estimation will not jump suddenly and will move smoothly to the measured faulty value. At the fault occurrence time,

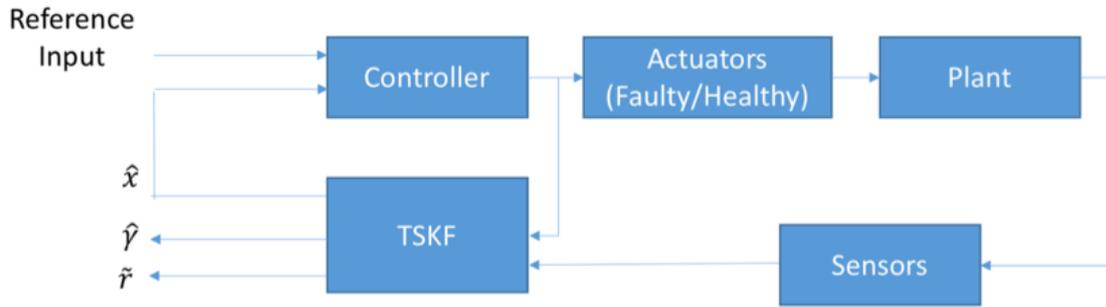


Figure 7.1: The overall structure of the system with faulty actuators and sensors.

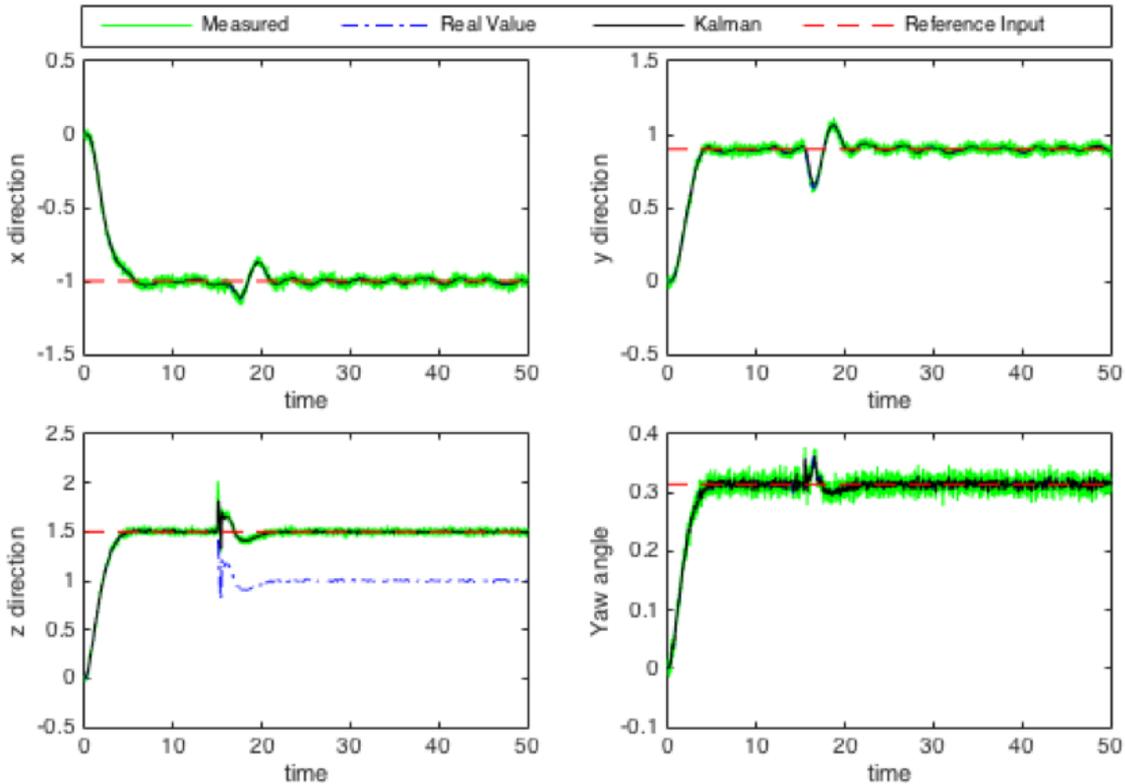


Figure 7.2: The impact of fault on z sensor on system response.

this smooth movement will cause a difference between measured and estimated value and will be observable in residuals.

Let us execute a scenario with faulty sensors and see how sensor bias faults effects the system response and how TSKF estimates the states and residuals. To this end, we assume that there is a 50 cm fault on z sensor at 15 (s). The impact of this fault on system response is plotted in Fig. 7.2.

As a bias fault happens in a sensor the measured value differs from its actual value, this causes a deviation in actual value of output from its reference value as incorrect value has been fed back to the controller. For instance, as fault is injected to z sensor at $t=15$ (s) controller tries to recover the difference between measured value and reference

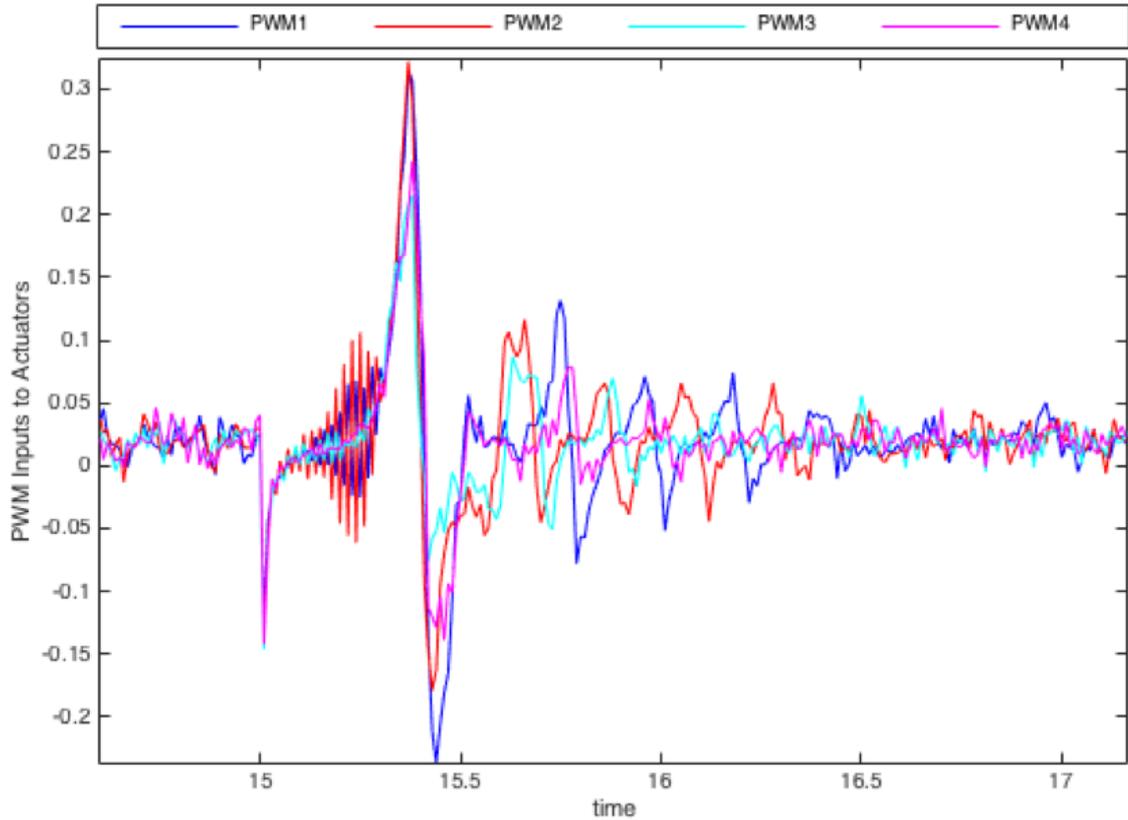


Figure 7.3: Control inputs when fault happened in height sensor.

input, whereas, the control inputs (Fig. 7.3) are decreased in order to reduce height and this, actually, makes the quadrotor go from $z=1.5$ m to $z=1$ m. This is the main negative effect of the sensor fault on system response. In the following section we will try to overcome this problem.

By plotting the height residual created by Kalman filter in Fig. 7.4 it can be seen that fault is detected and diagnosed when corresponding residual crossed its threshold. The fault isolation is done because remaining residuals are less than their thresholds during the flight (Fig. 7.5). For more information, the height residual crossed its threshold because Kalman estimated value for z direction did not jump at the fault occurrence time (Fig. 7.6).

In order to challenge more the Kalman filter in detecting and diagnosing the sensor faults, let us give 40 cm, 50 cm, and 7° bias faults at 15, 25, and 35 (s) on x , z , and roll sensors. The impact of faults on system position and orientation are revealed in Fig. 7.7 and 7.8 and residuals calculated by Kalman filter are shown in Fig. 7.9. Note that, the quadrotor's position in y direction deviated from reference value after fault on roll sensor, this also can be seen in y direction residual. This problem happened because

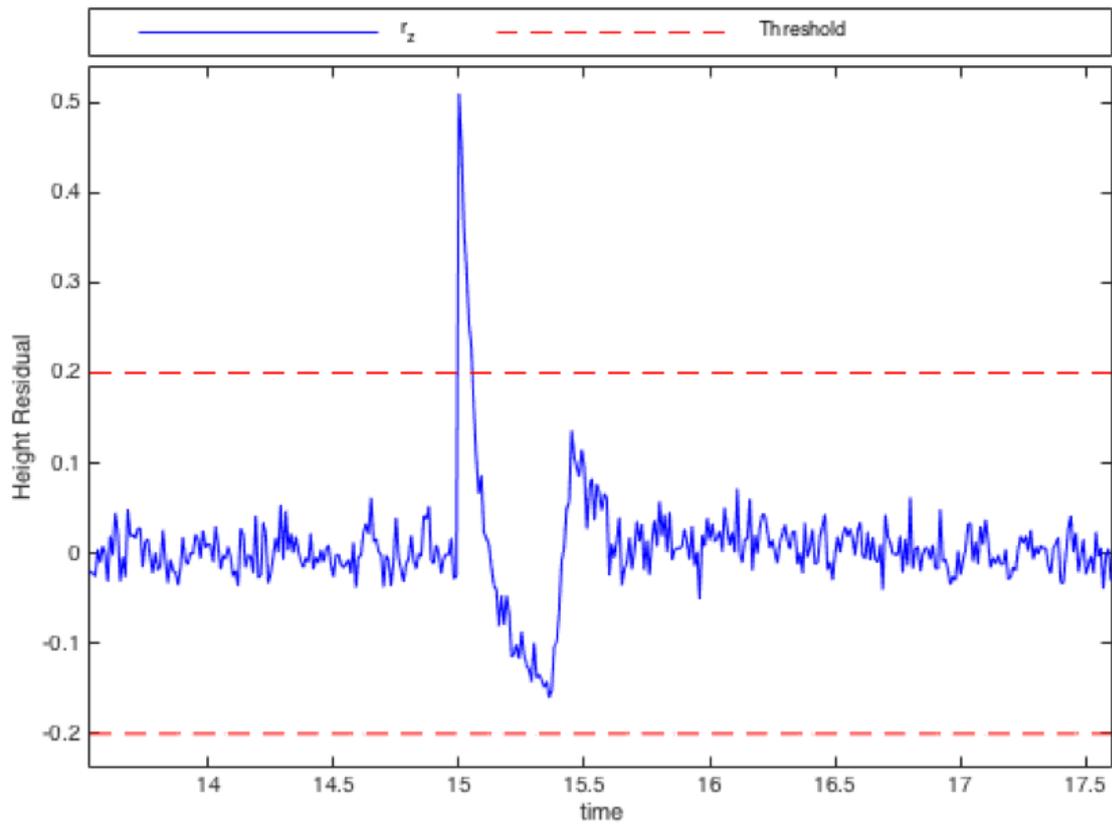


Figure 7.4: Height sensor's residual when the sensor is faulty.

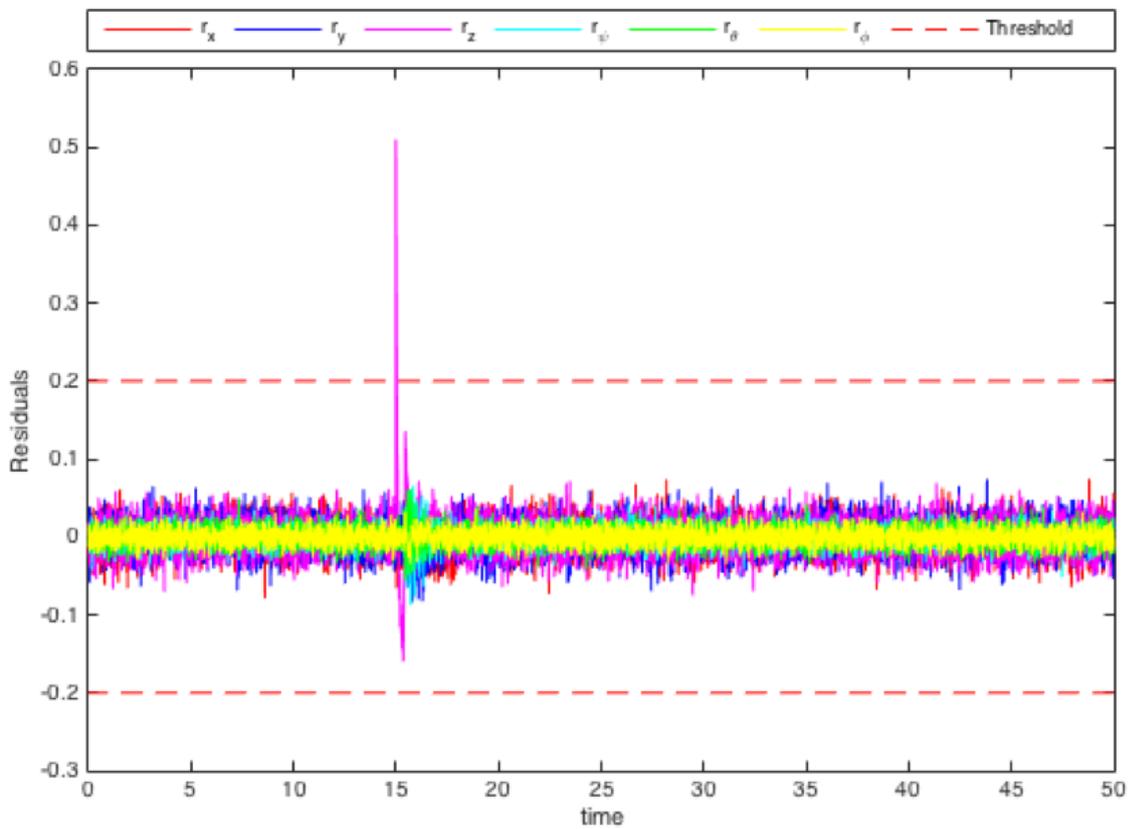


Figure 7.5: Kalman filter residuals when height sensor is faulty.

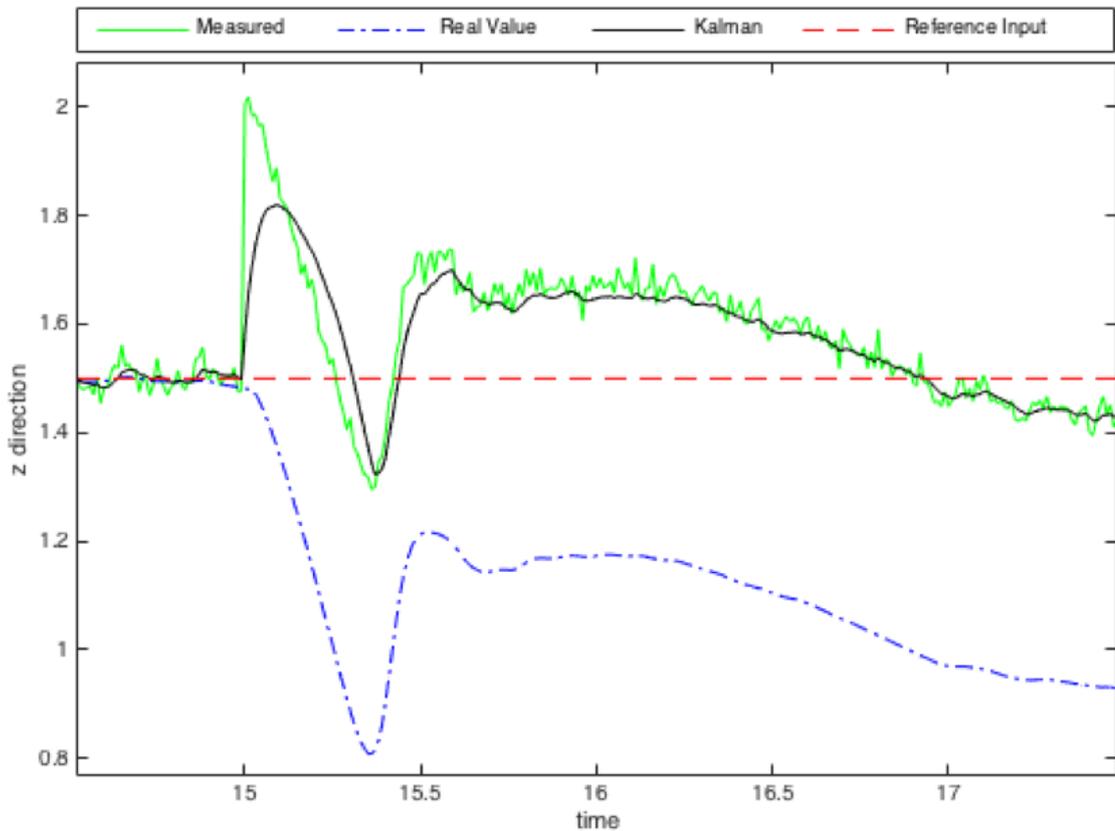


Figure 7.6: The impact of sensor fault on system response and Kalman estimation.

acceleration in y direction violated the linearization condition (Fig. 7.10), this problem is discussed and finally solved in next section.

7.2 Active Fault Tolerant Control on Sensor Fault

Active Fault Tolerant Control (AFTC) is a process of real-time reconfiguration of system parameters with respect to existing faults on system. In previous section we detected and diagnosed the sensor faults by watching the Kalman filter residuals, also, a small fault on roll sensor caused a permanent difference between estimated and measured y position of the quadrotor, this also has been seen in y sensor residual. Although, y sensor was not faulty, y residual deviated from zero but did not cross the threshold. By taking a closer look to the residuals (e.g. Fig. 7.4) it can be seen that, the value of the residual at the fault occurrence time is equal to the magnitude of bias fault happened on the corresponding sensor, however, this value goes to zero as Kalman estimation gets closer to measured value by faulty sensor. By taking advantage of this property of the residuals, Sensor Correction block is added to the system as shown in Fig. 7.11.

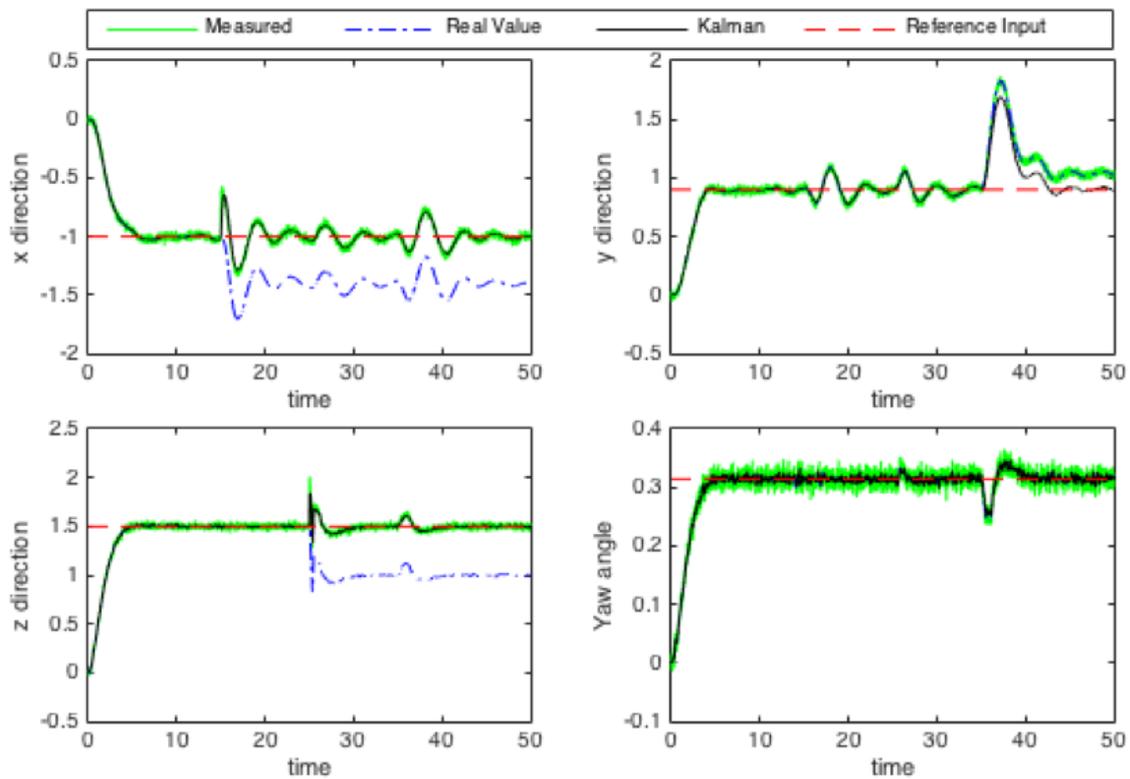


Figure 7.7: The impact of faults on three sensors on the system response.

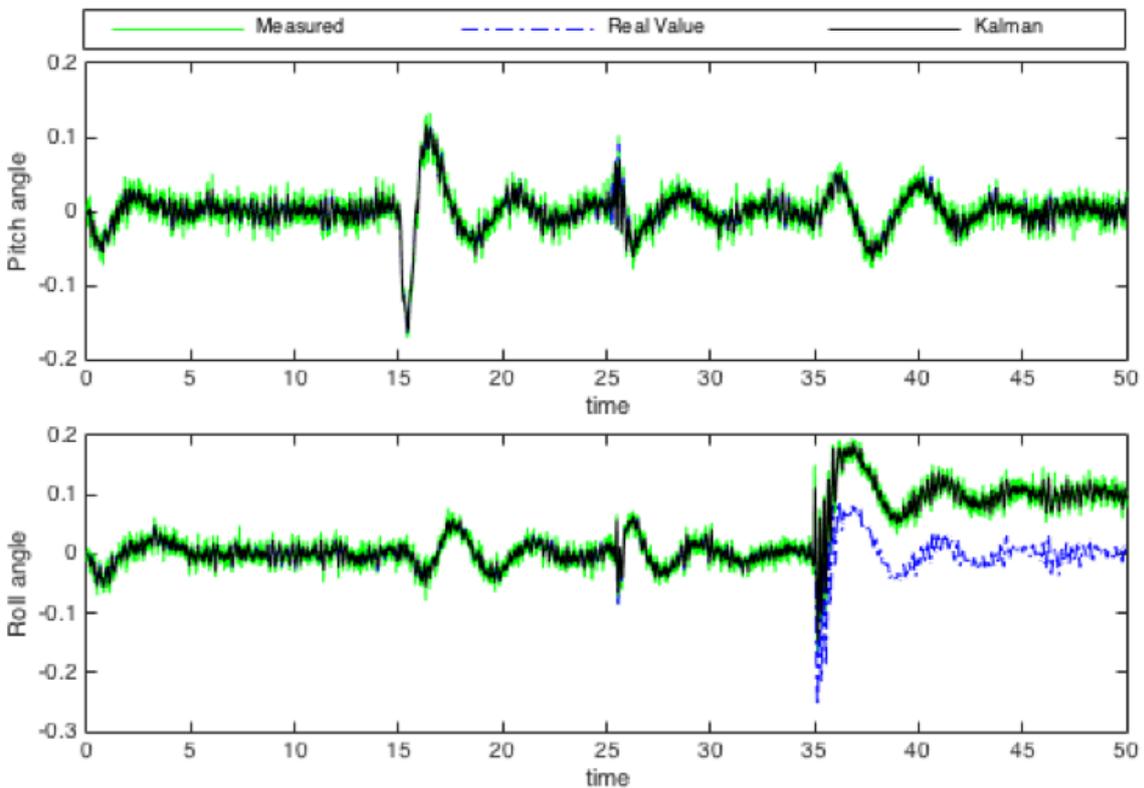


Figure 7.8: The impact of faults on three sensors on pitch and roll angle.

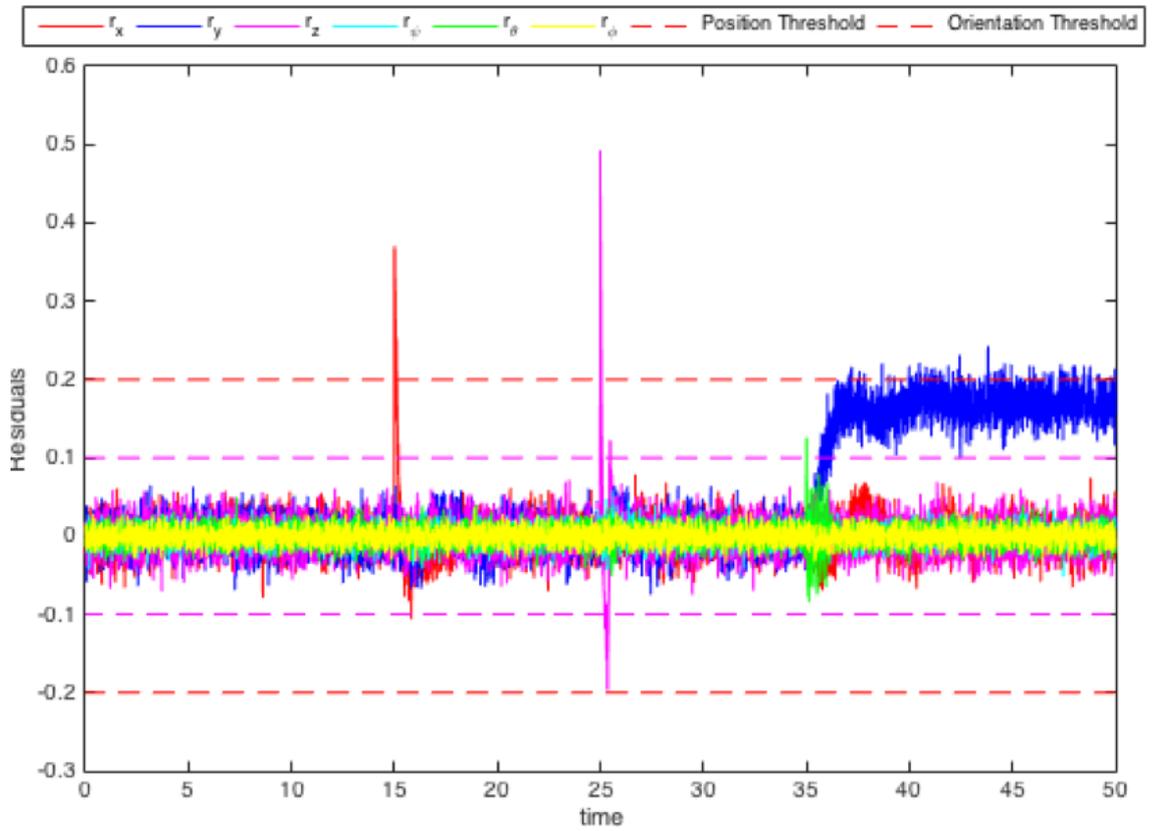


Figure 7.9: Kalman filter residuals when three sensors are faulty.

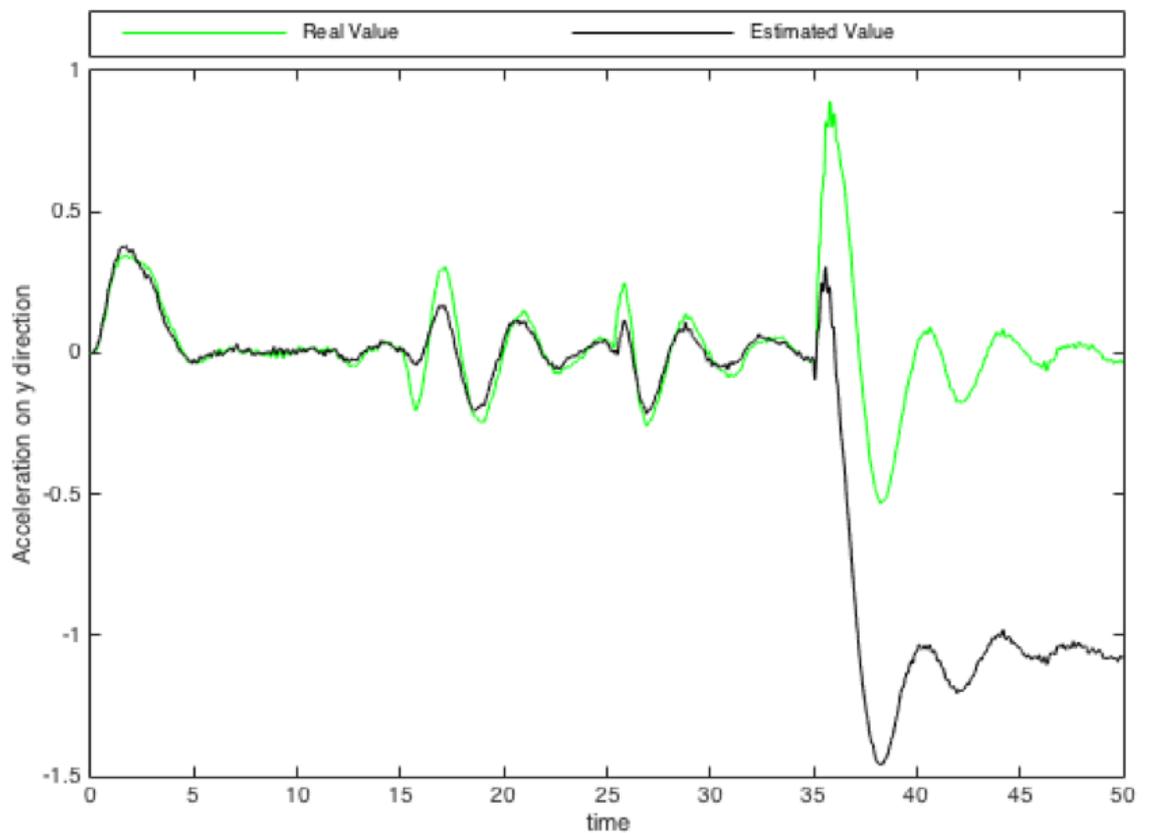


Figure 7.10: Violating linearization condition by acceleration on y direction.

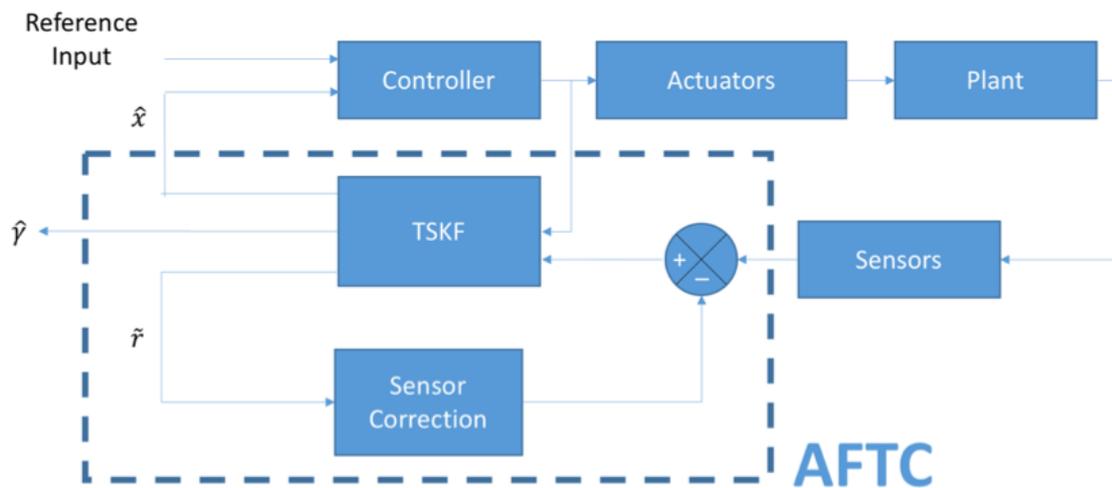


Figure 7.11: The overall structure of the AFTC system.

Sensor Correction block receives the residuals and compares them with the corresponding thresholds, if any residual crosses its threshold, the residual value is saved and considered as correction parameter for sensors. By subtracting the correction parameter of all sensors with measured values, the corrected measured value by faulty sensor is produced. After doing this, the corrected measured value by sensor that is created by Sensor Correction block is fed back to the TSKF block.

In order to check the ability of the AFTC block on overcoming the sensor faults let us execute some flight scenarios. In first scenario, we assume that there is 50 cm fault on height sensor at 15 (s). This fault is the same as fault on first flight scenario in section 7.1. The impact of fault on system response is plotted in Fig. 7.12.

By analyzing the system response and comparing it with the results on section 7.1 two main results are concluded

- AFTC block made the quadrotor to follow the reference input even when measured value by sensor is faulty
- AFTC block compensated the permanent deviation of Kalman estimation from healthy measured value in y direction ...

To challenge the AFTC system more, let us execute the last flight scenario in section 7.1. Let us give 40 cm, 50 cm, and 7° bias faults at 15, 25, and 35 (s) on x, z, and roll sensors. The system ability in dealing with faults on multiple sensors is obvious in Fig. 7.13 and Fig. 7.14.

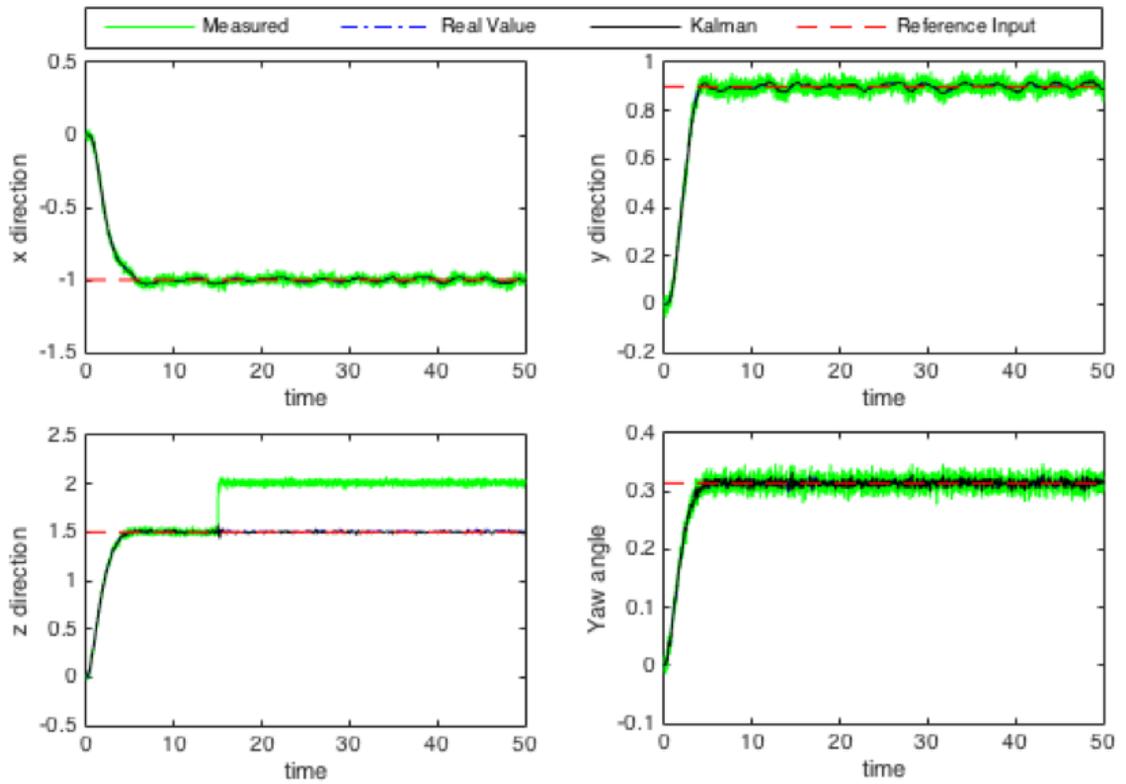


Figure 7.12: Response of the AFTC system when one sensor is faulty.

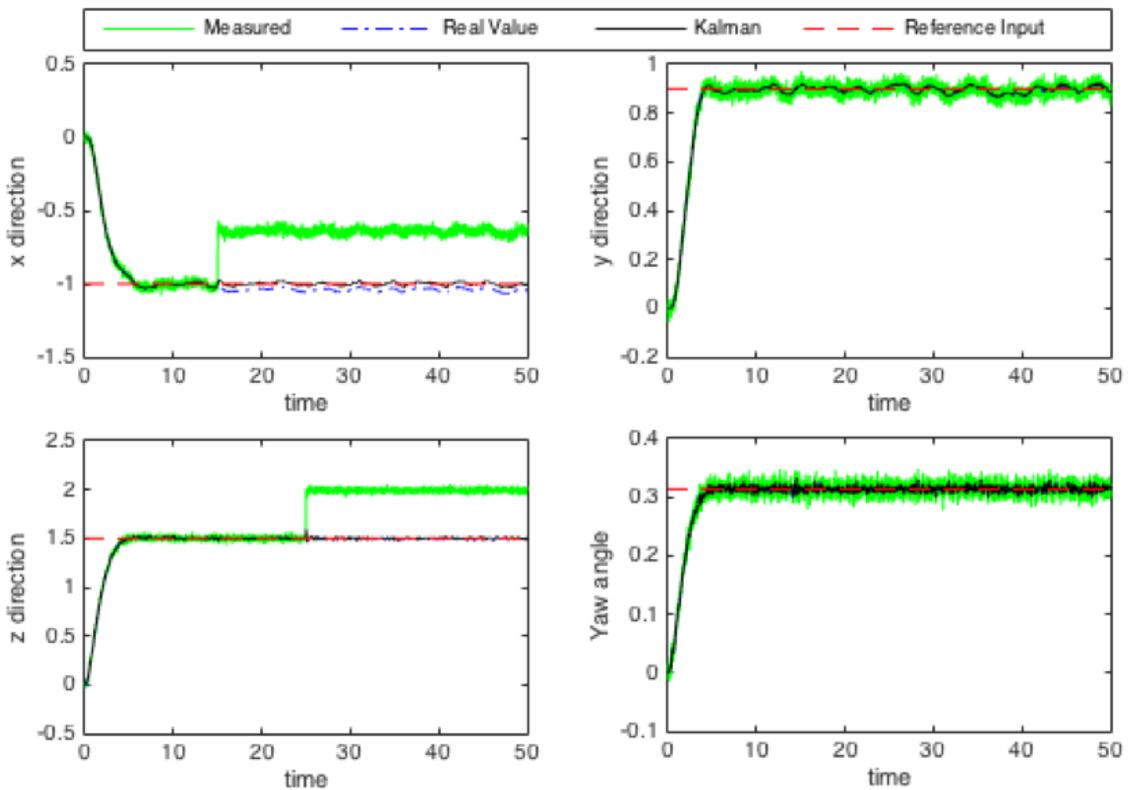


Figure 7.13: Response of the AFTC system when multiple sensors are faulty.

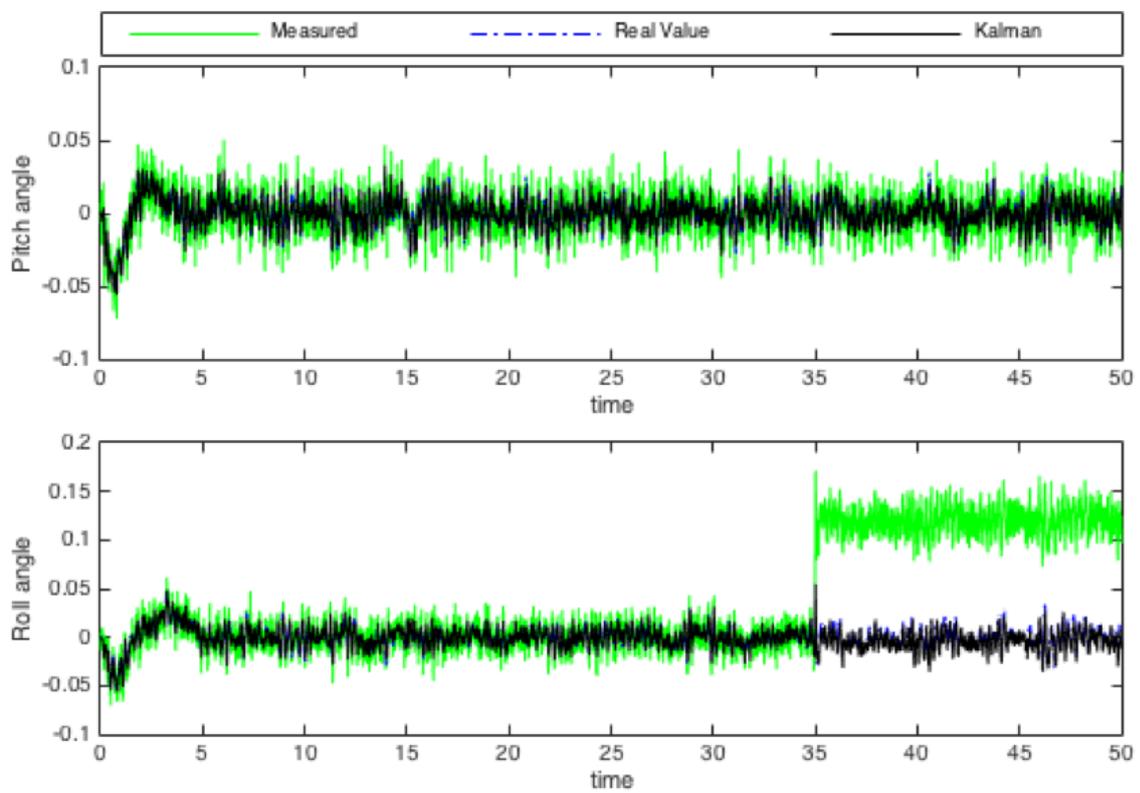


Figure 7.14: Pitch and roll angles of the faulty quadrotor equipped with AFTC.

8. ACTUATOR FDD AND SENSOR AFTC

In this section, the final version of the simulated system in Matlab Simulink is given. The structure of the system is the same as Fig. 7.11. System model is the nonlinear system and number of sensors are real. Controller is the reference follower controller in which the input is the reference input (by user) and estimated states by TSKE. Actuators faults are detected and diagnosed using $\hat{\gamma}$ vector. Sensors' FDD is done using Kalman filter's residuals. ATFC of sensor faults are completed with the help of Sensor Correction block.

As a final faulty flight scenario, let us assume there are -1 m, 2.3 m, 5.8° , 7° , 60%, and 75% faults on x, y, yaw, roll sensors, actuator 1, and actuator 2 faults at 15, 20, 10, 40, 18, and 5 (s). The system response to the reference inputs including the faults effects are plotted in Fig. fig8.1 to Fig. 8.3.

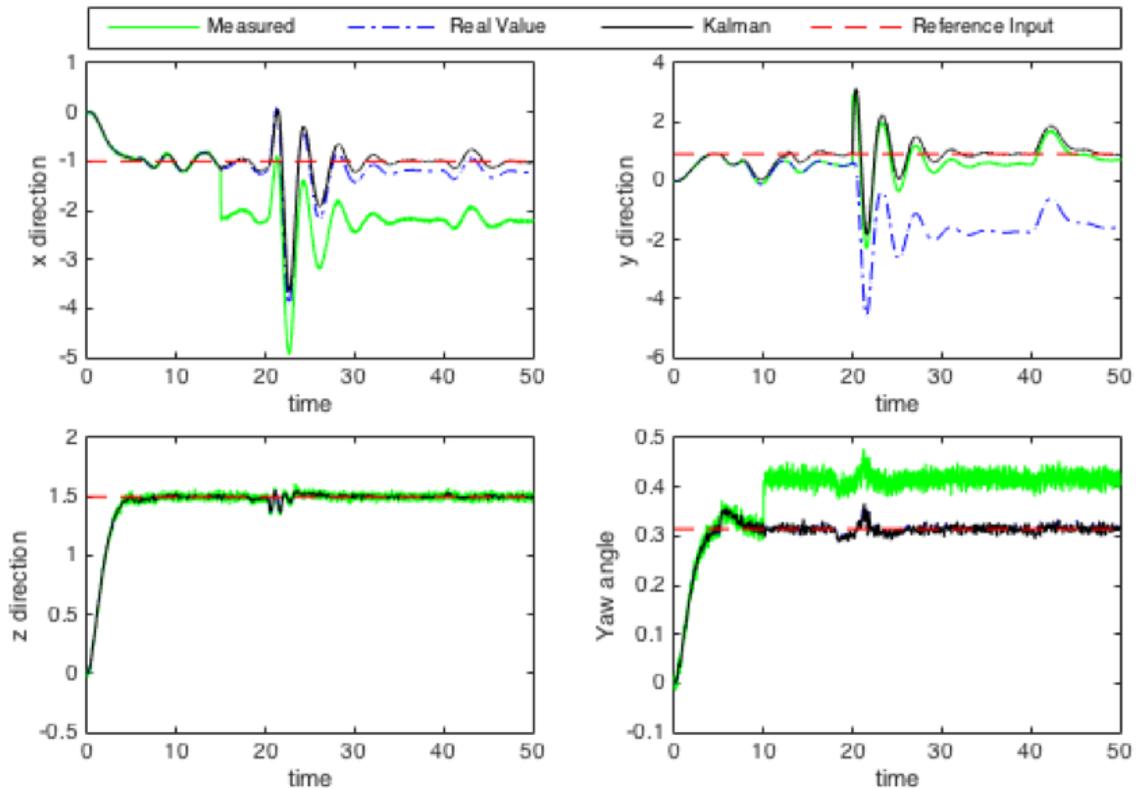


Figure 8.1: Final system performance on faulty sensors and actuators.

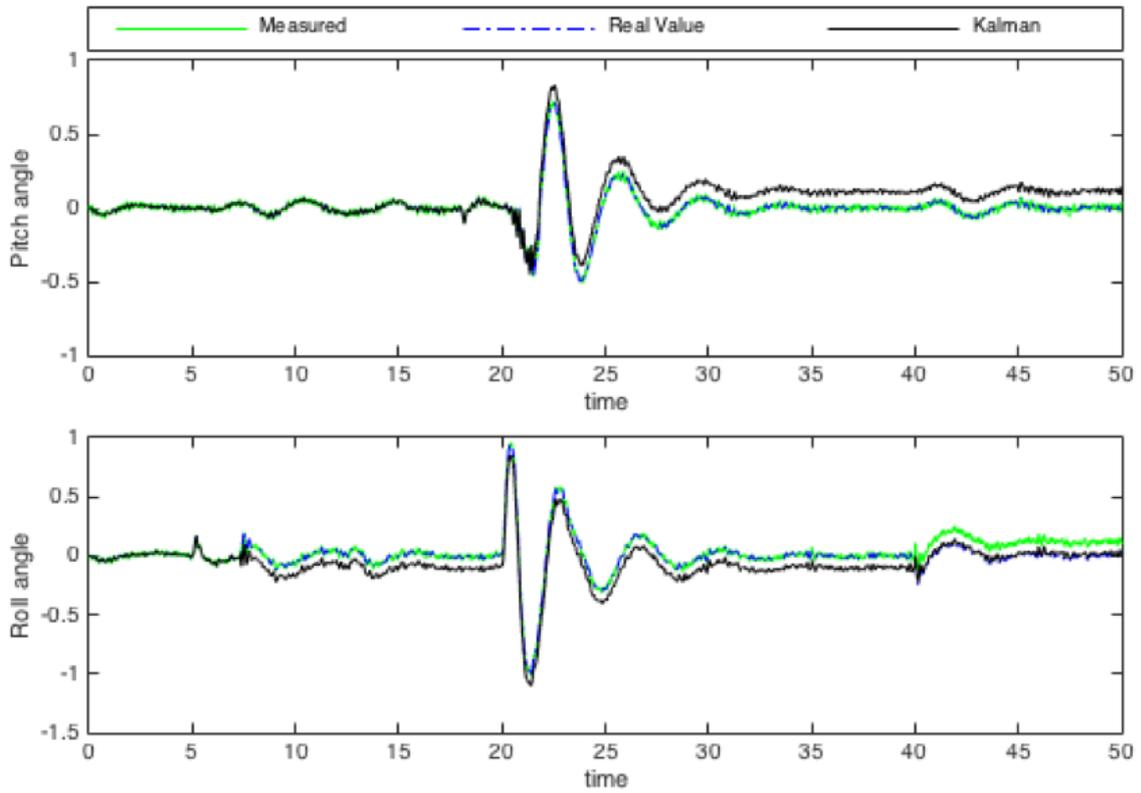


Figure 8.2: Final system performance on faulty sensors and actuators.

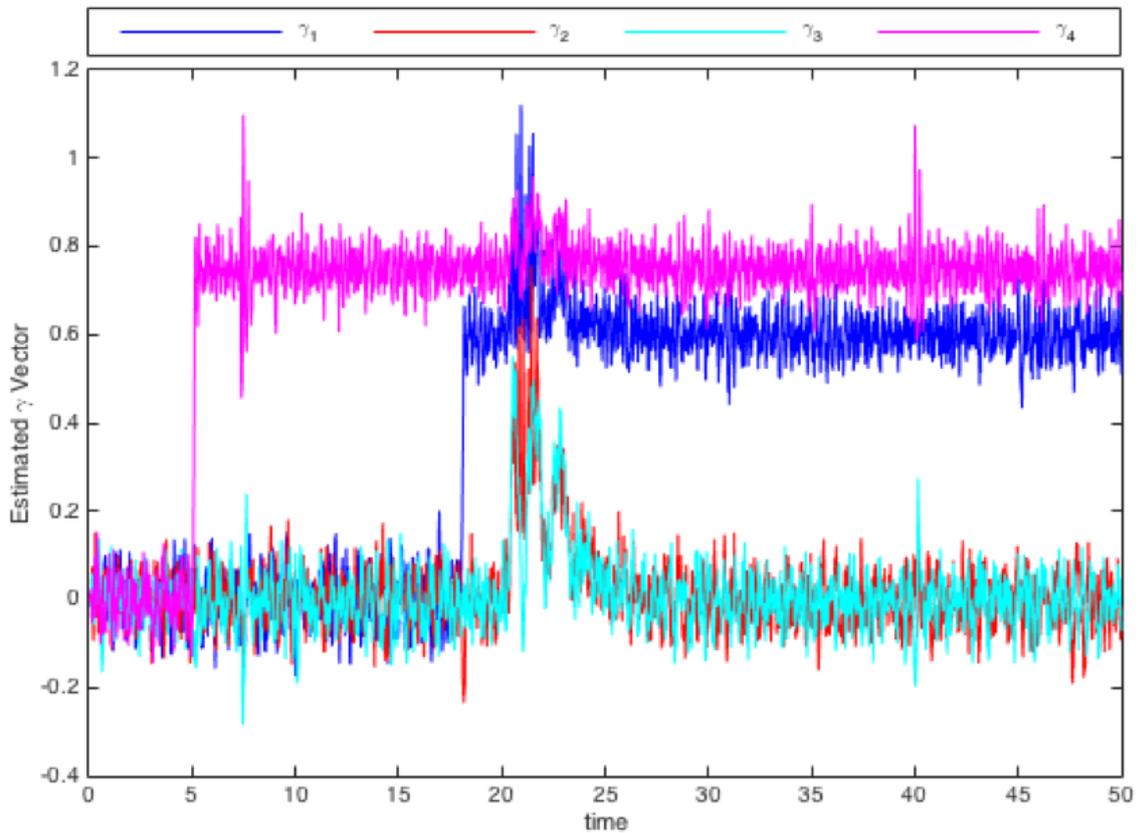


Figure 8.3: Estimated vector in final system.

This scenario proves the ability of the system in dealing with faults on different places of the system at any time. Also, reveals the weakness of the system in dealing with high percentage faults on actuators that is discussed in chapter 6.

In seeking a solution to this, for prospective researchers, we want to draw a path. In the FDD block, the time and size of both actuator and sensor faults are calculated. In sensor AFTC we saw that this information can be used to compensate the adverse effects of occurred faults. Obviously, when an actuator fault happens, the system response deviates from its desired reference input. Our experiences proved that, by reconfiguring the controller parameters this deviation can be compensated. The amount of parameter change in the controller is highly related to the amount of occurred actuator fault, as a result, this relation can be calculated and controller parameters can be reconfigured with respect to the actuator faults so that system response can follow the input trajectories even in the faulty situation on actuators. If this be performed, the AFTC for actuators can be done.

9. CONCLUSION

In this dissertation, we started with deriving the Qball-X4 nonlinear equations. By linearizing the nonlinear equations around a nominal point (hovering on the air) the state-space matrices are calculated. Using Linear Quadratic Regulator (LQR) controller design method, a state-feedback gain, to make the system stable around the equilibrium point is calculated. Two-Stage Kalman Filter (TSKF) is formulated and tested in estimating the states. At first, its assumed that there is a sensor for each state which is not true in real life. Substituting the three position and three orientation sensors are flowed by exchanging the linear model of the system with nonlinear model. Obviously, linear controller with one feedback loop is not able to deal with the nonlinear model of the system, therefore, a reference follower controller with two feedback loops is designed and tested. The simulation results for the nonlinear system containing TSKF block, actual numbers of sensors, and reference follower controller are plotted and analyzed.

When it comes to faults, actuator faults play an important role in making problems during flight scenarios. In chapter 6, actuators' faults and their consequences on system responses are studied and it is proved that how reference follower controller is able to deal with these faults. However, higher percentage of actuators' faults' seemed a problem in flights, which can be solved by taking the estimated $\underline{\gamma}$ vector to reconfigure the controller parameters in Fault Detection and Diagnosis (FDD) section. This subject is left as a future work.

Sensors' faults seem as a crisis when low-cost sensors are used in cheap quadrotors. Magnetic environment, vibrations and etc. can cause a bias fault on sensors. The impacts of sensors' bias faults on system response are discussed in chapter 7. We saw that when a sensor fault occurs, system response deviates from reference value, the reason behind this is that the faulty measured value is fed back to the TSKF block and consequently, the wrong value enters to the controller, therefore, the controller drives the control inputs to move the quadrotor to the reference position and this

causes a difference between reference and real position of the quadrotor. Using TSKF's residuals, sensor faults are detected and diagnosed. Finally, by taking the advantage of a property of residuals, in which, the magnitude of the faulty sensor's residual is equal to the size of the happened fault in corresponding sensor, the Sensor Correction block is introduced. This block corrects the faulty sensors values using the information in residuals and sends the corrected sensors' values to the Kalman filter. This causes the system stay on its reference position and orientation even when sensors are faulty. This problem is called Active Fault Tolerant Control (AFTC).

Finally, both actuator and sensors faults in either same or different times are given and the ability of system to control these faults is discussed. Last but not least, a motivational recommendation for prospective researchers is given in chapter 8, in which, a path to how implement AFTC for actuator faults is presented.

REFERENCES

- [1] **Moghadam, M. and Caliskan, F.** (2015). Actuator and sensor fault detection and diagnosis of quadrotor based on Two-Stage Kalman Filter, *Control Conference (AUCC), 2015 5th Australian*, pp.182–187.
- [2] **Madani, T. and Benallegue, A.** (2006). Backstepping Control for a Quadrotor Helicopter, *Intelligent Robots and Systems, 2006 IEEE/RSJ International Conference on*, pp.3255–3260.
- [3] **Bouabdallah, S. and Siegwart, R.** (2005). Backstepping and Sliding-mode Techniques Applied to an Indoor Micro Quadrotor, *Robotics and Automation, 2005. ICRA 2005. Proceedings of the 2005 IEEE International Conference on*, pp.2247–2252.
- [4] **Milhim, A., Zhang, Y. and Rabbath, C.A.**, (2010). chapterGain Scheduling Based PID Controller for Fault Tolerant Control of Quad-Rotor UAV, Infotech@Aerospace Conferences, American Institute of Aeronautics and Astronautics, 0.
- [5] **Sharifi, F., Mirzaei, M., Gordon, B. and Zhang, Y.** (2010). Fault tolerant control of a quadrotor UAV using sliding mode control, *Control and Fault-Tolerant Systems (SysTol), 2010 Conference on*, pp.239–244.
- [6] **A. Freddi, A.L. and Longhi, S.** (2011). A Feedback Linearization Approach to Fault Tolerance in Quadrotor Vehicles, *18th IFAC World Congress*, Milano, Italy, pp.5413–5418.
- [7] **Amoozgar, M.H., Chamseddine, A. and Zhang, Y.** (2012). Experimental Test of a Two-Stage Kalman Filter for Actuator Fault Detection and Diagnosis of an Unmanned Quadrotor Helicopter, *Journal of Intelligent & Robotic Systems*, **70**(1), 107–117.
- [8] **F. Caliskan, C.H.** (2014). Sensor and Actuator FDI Applied to an UAV Dynamic Model, *19th World Congress The International Federation of Automatic Control*, Cape Town, South Africa, pp.12220–12225.
- [9] **García Carrillo, L.R., Dzul López, A.E., Lozano, R. and Pégard, C.** (2013). *Quad Rotorcraft Control: Vision-Based Hovering and Navigation*, Springer London.
- [10] **Rugh, W.J.** (1996). *Linear System Theory (2Nd Ed.)*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA.
- [11] **Keller, J. and Darouach, M.** (1997). Optimal two-stage Kalman filter in the presence of random bias, *Automatica*, **33**(9).

- [12] **Hellerstein, J.L., Diao, Y., Parekh, S. and Tilbury, D.M.** (2004). *Feedback Control of Computing Systems*, John Wiley & Sons.