The University Of Sheffield.

Design of sliding mode-based nonlinear control systems with nonlinear full-order state observers for underactuated coupled systems

Thesis submitted to the University of Sheffield for the degree of
Doctor of Philosophy

by

Ahmad Riyad Firdaus

Department of Automatic Control and Systems Engineering
The University of Sheffield
United Kingdom

July 2018

Abstract

This thesis presents the development of new nonlinear control algorithms with full-order state observers to overcome the control challenges encountered with nonlinear and underactuated systems. Quadcopter unmanned aerial vehicles are good example of underactuated systems, and this is selected in this research for validating the developed control and observer systems.

It is shown through review of the literature that *sliding mode control* and *sliding mode observer* have many advantages over other control and observer methods in dealing with nonlinearity, underactuation, and coupled systems. However, the chattering phenomenon noted in such approaches can degrade the overall performance of the system. In this work, new methods to tackle the issue are proposed by employing *sliding mode-based interval type-2 fuzzy control* and *sliding mode-based interval type-2 fuzzy observer*. Moreover, in terms of control system, *set-point weighting function* and *integral term* are proposed and incorporated into the control system to improve the overall performance of the system. Furthermore, the *weighting* and *integral terms* are utilized to design further sliding mode-based control systems, including: *quasi-sliding mode control*, *super-twisting algorithm of sliding mode control*, and *dynamic sliding mode control*. In terms of observer, for the purposes of performance comparison, other nonlinear observer approaches are designed including: *quasi-sliding mode observer*, *super-twisting algorithm of sliding mode observer*, *high order sliding mode observer*, and *extended Kalman filter*.

All designed control and observer methods are evaluated and compared their performances based on predefined performance criteria through numerical simulations. Well-performing control and observer methods in simulated exercises are selected for further analysis and validation with real-time experimental investigations. From the simulation re-

sults obtained, overall *set-point integral quasi-sliding mode control* and *quasi-sliding mode observer* have performed better than other methods. The proposed methods of employing *set-point weighting function* and *integral term* to sliding mode-based control systems have performed very well in reducing overshoot, rise time, and steady-state errors. Meanwhile, in terms of experimental validations, *set-point integral super-twisting algorithm of sliding mode control* has shown better ability than *set-point integral quasi-sliding mode control*, and the observer method of *quasi-sliding mode observer* has yielded 2^{nd} - order state well.

Acknowledgements

All price be to Allah, the Most Merciful and Beneficent. There is no God but Allah and Prophet Muhammad (Peace be upon Him) is the Messenger of Allah. I would like to pay special thankfulness, warmth and appreciation to the persons and institutions below that made my research successful and assisted me at every point to cherish my goal:

1. My supervisor Professor M.O. Tokhi for the continuous support of my PhD study and related research, for his patience, motivation, and immense knowledge. It is whole-heartedly expressed that his advices for my research proved to be a landmark effort towards the success of my project.

2. My second supervisor, Dr George Panoutsos, whose help and sympathetic attitude at every point during my research helped me to work in time.

3. Indonesian Endowment Fund for Education (LPDP) and Politeknik Negeri Batam Indonesia, who have supported me financially during my PhD study in The University of Sheffield, United Kingdom.

4. All the faculty, staff members and lab technicians of Automatic Control and Systems Engineering, whose services turned my research a success.

5. All my colleagues at Politeknik Negeri Batam Indonesia, especially big family of Electrical Engineering Departement who have supported me morally and have been wonderful fellow discussions so I can complete this research very well.

6. My fellow office and lab-mates: Radhwan, Basim, Sapto, Omar, Abdullah Benomair, Abdullah Al-Shatti, Hyreil, Nafri, Hadi, Normaniha, Ghasaq, Siti, Daniela, Fiza, Ruzaini, Bukhari, Hassrizal, Mohammed Y. Hazim. You gave me all the things

I did not get from my education – determination, motivation and inspiration. Thanks for being colleagues and teachers, two in one.

7. The big family of KIBAR and PPI Sheffield, you make happiness worth sharing and you make my dreams worth believing. Thanks for being by my side and making my life worth living.

8. Ibu, Mamah Nini, and all family members, without you I was nothing; you extended your support morally and emotionally.

9. My beloved wife: Yesi Sugiarti and children: Muhammad Belva Zulfan Firdaus, Muhammad Nabil Najmuddaulah Firdaus, and Muhammad Alby Saqif Firdaus, you are the reason for my pride, but you have also made me humble. You are the reason for my satisfaction, but you have also made me want more for you. You are the reason for my strength, but you have also made me cry tears of joy.

Allah will exalt those who believe among you, and those who have been granted knowledge to high ranks. Allah is aware of all that you do **(The Quran 58:11)**

Contents

List of Figures

List of Tables

Acronyms

3D	three dimension
AHRS	attitude and heading reference system
BC	block control
BS	backstepping
DOF	degrees of freedom
DSMC	dynamic sliding mode control
EKF	extended Kalman filter
Ess	steady-state error
FL	feedback linearisation
FLS	fuzzy logic system
GA	genetic algorithm
HOSMO	high order sliding mode observer
IT2FLS	interval type-2 fuzzy logic system
IT2FSMC	sliding mode-based interval type-2 fuzzy control
IT2FSMO	sliding mode-based interval type-2 fuzzy observer
ITAE	integral of time-weighted absolute error
KF	Kalman filter
LMI	linear matrix inequality
LQR	linear quadratic regulator
MF	membership function
MPC	model-predictive control
MSE	mean squared error
PD	proportional derivative

PID proportional integral derivative

PWM pulse width modulation

QuasiSMC quasi-sliding mode control

QuasiSMO quasi-sliding mode observer

Tr rise time

SD steepest descent

SMC sliding mode control

SMO sliding mode observer

SNR signal to noise ratio

STASMC super-twisting algorithm of sliding mode control

STASMO super-twisting algorithm of sliding mode observer

UAV unmanned aerial vehicle

VSC variable structure control

VTOL vertical take-off and landing

Chapter 1

Introduction

1.1 Background

Underactuation is a unique property of the nonlinear systems with fewer control inputs than the degrees of freedom (DOFs) that need to be controlled. This causes a control problem that cannot be resolved by conventional control theory. Therefore, this typical class of nonlinear systems has attracted significant attention of researchers for development of suitable approaches to resolve associated control issues. Underactuated systems generally show some advantages, including more straightforward structure, less energy utilization, and better operational adaptability. Examples of underactuated systems include aircraft, unmanned aerial vehicles (UAVs), spacecraft, mobile and walking robots, flexible systems, and underwater vehicles.

The research related to the design of control systems for underactuated systems is numerous. In the author's knowledge, the first research addressing underactuated system was conducted by Henning (1919) in developing an artificial hand, following which research in this area grew rapidly. Over the last few decades, control system design of underactuated systems has been studied extensively. Examples include inverted pendulum (Riachy et al., 2008; López-Martínez, 2009; Wang et al., 2017c; Yang and Zheng, 2018); TORA systems (Choukchou-Braham et al., 2011; Mobayen, 2015; Sun et al., 2017b); autonomous underwater vehicles (Zain et al., 2011; Li et al., 2017a); robotic systems (Pucci et al., 2015; Cao et al., 2018); cranes (Tuan and Lee, 2016; Su et al., 2017; Sun et al., 2017a; Wu et al., 2017; Lu et al., 2018); spacecraft (Li et al., 2017b); hovercraft (Jeong and Chwa, 2018). In

addition to the systems mentioned, broad research on control design has been conducted to deal with underactuated systems in vertical take-off and landing (VTOL) aircraft especially in quadcopter-based UAVs.

There is significant interest shown by researchers in the development of control mechanisms for UAVs, as detailed later in Chapter 2. This trend is noted because such vehicles have several challenging conditions and constitute complex systems that a control system has to deal with. Firstly, the system dynamic model is highly nonlinear, considering the effects of complicated aerodynamics of such rotorcraft, non-minimum phase, and subject to non-holonomic constraints. Moreover, the system has six degrees of freedom, namely rotational motions: pitch, roll, and yaw actions; and translational motions: x, y and z - axes movements, four control inputs and four independent actuation inputs (rotor speeds). The system is highly underactuated, and the rotational and translational motions are coupled. Furthermore, disturbances and uncertainties due to the operational environment of the system pose challenges for design of controllers for such systems.

Availability of information on all states is a further matter of consideration in quadcopter UAVs. Such information plays a critical role in controlling underactuated systems. Sensor availability and noise interference are issues that limit the acquisition of states information on the system. Hence, such issues have triggered several studies to estimate unmeasured states and to deal with uncertainties and disturbances, as detailed later in Chapter 2.

Research investigations have led to the conclusion that an appropriate control system and observer method should be designed to handle the complexity of such systems, including: nonlinearity, underactuated and coupled dynamics, uncertainties and disturbances which otherwise may lead to overall performance degradation of the system.

1.2 Aim of the research

The aim of the research is to develop robust control mechanisms to cope with high nonlinearity, underactuated and coupled dynamics in the presence of disturbances and uncertain-

ties, and further to develop robust state observer for estimation of unmeasured information on system states, and to verify the performance of the developed mechanisms within simulated and experimental environments of quadcopter-based UAVs

1.3 Research objectives

The main objectives of the research are as follows,

- To investigate new robust nonlinear control and observer method to deal with nonlinear underactuated systems,
- To investigate the development of new robust control and observer system with application to quadcopter UAVs,
- To investigate the robustness of the developed control and observer system subject to disturbances and parameter uncertainties,
- To carry out a comprehensive comparative performance assessment of the developed control and observer system with other control and observer methods subject to disturbances and parameter uncertainties within simulated environment of quadcopter UAV,
- To design a quadcopter prototype and use as an experimental platform for validation of the developed control and obsrever approach.

1.4 Contributions

New contributions to knowledge have been made in this research via the development and investigation of the following:

- Set-point weighting function for integral sliding mode control systems,
- Integral sliding mode-based interval type-2 fuzzy control (IT2FSMC) with application to quadcopter UAVs,

- Sliding mode-based interval type-2 fuzzy observer (IT2FSMO) with application to quadcopter UAVs,
- A set of integral sliding mode-based control systems, including: quasi-sliding mode control (QuasiSMC), super-twisting algorithm of sliding mode control (STASMC), and dynamic sliding mode control (DSMC) with application to quadcopter UAVs,
- A set of sliding mode-based observer methods, including quasi-sliding mode observer (QuasiSMO), super-twisting algorithm of sliding mode observer (STASMO), and high order sliding mode observer (HOSMO) with application to quadcopter UAVs,
- The state estimator of extended Kalman filter (EKF) for quadcopter UAVs,
- Comparative assessment of performances of developed sliding mode controls (SMCs) with sliding mode observers (SMOs) as well as the extended Kalman filter (EKF).

1.5 Methodological approaches

In this section, the methodology of the research is set out to achieve the objectives of the research. This comprises the following:

(a) *Literature review.*

- Identify the general intersection areas of research.
- Identify areas that need detailed assessment to establish unresolved issues and gaps in knowledge.
- Collect previous works from credible sources.
- Sort documents based on closeness to the scope of the project and reviewing these in detail to map to the gaps in the literature.

(b) *Control design and simulation.*

- Develop the mathematical and the state space model of quadcopter UAVs.

- As has been investigated in the literature review, select a control system that has the best control performance in handling nonlinear underactuated systems and has the opportunity to be further developed.
- Develop the selected control method for quadcopter UAVs and improve its performance with possible new methods.
- Conduct comparative studies of possible new methods by simulating and analysing their performance in MATLAB
- Select an observer method that has the best performance in dealing with nonlinearity, underactuation, noise disturbance and parameter uncertainity, and has the opportunity to be further developed.
- Develop the selected observer method for quadcopter UAVs and improve its performance with possible new methods.
- Design the new control method with the new nonlinear full-state observer for quadcopter UAV
- Conduct comparative studies of possible new methods by simulating and analysing their performance in MATLAB with introduction of parameter mismatch, noise and air drag disturbance.

(c) *Experimental validation*

- Design test bed and embedded systems of a quadcopter UAV.
- Test all electronic parts of the systems.
- Select a candidate new control and observer for experimental test.
- Code and implement the new control and observer algorithm on the experimental test bed.
- Acquire, record and analyse the acquired data.
- Optimise the overall quadcopter UAV design.

1.6 Publications

Publications arising from this research, in either accepted or in print mode, are given below. There are also further publications in preparation for submission, which are not listed.

1. Firdaus A.R., Tokhi, M.O. (2015). Sliding Mode Control–Based Interval Type-2 Fuzzy Logic Controller For Quadcopter UAVs. In ASSISTIVE ROBOTICS: Proceedings of CLAWAR 2015 18th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines, HangZhou, China, 06–09 September 2015, World Scientific, pp. 555–563.

2. Firdaus A.R., Tokhi, M.O. (2015). Robust sliding mode-based interval Type-2 fuzzy logic observer for quadcopter UAVs. Proceedings of ICSTCC 2015: 19th International Conference on System Theory, Control and Computing, Romania, 14 – 16 October 2015, pp. 559 – 564.

3. Firdaus A.R., Tokhi, M.O. (2015). Soft-computing control paradigm: Sliding mode – based Interval Type-2 Fuzzy Logic control and observer for Quadcopter UAVs. Poster session presented at the ACSE PGR Symposium 2015. Department of Automatic Control and Systems Engineering, The University of Sheffield, United Kingdom.

4. Benomair A.M., Firdaus A.R., Tokhi, M.O. (2016). Fuzzy sliding control with nonlinear observer for magnetic levitation systems. Proceedings of MED2016: 24th Mediterranean Conference on Control and Automation, Athens, Greece, 21-24 June 2016, pp. 256-261.

5. Ali S.K., Firdaus A.R., Tokhi, M.O., Al-Rezage G. (2016). Tracking human upper-limb movements with sliding mode control type-II fuzzy logic. Proceedings of 21st International Conference on Methods and Models in automation and Robotics (MMAR), Miedzyzdroje, Poland, 29 August–01 September 2016, pp. 426-431.

6. Firdaus A.R., Tokhi, M.O. (2017). Real-time Embedded Sliding Mode Observer for Quadcopter UAVs. In MOBILE MACHINE: Proceedings of CLAWAR2017 -

20th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines, Porto, Portugal, 11–13 September 2017, pp. 205–212.

7. Basri H.B.H., Rossiter J. A., and Firdaus A.R. (2018). Feasibility Study of Switching Function Approaches in Sliding Mode Control for a Spacecraft's Attitude Control System. Proceedings of CoDIT'18 - 5th International Conference on Control, Decision and Information Technologies, Thessaloniki, Greece, 10-13 Apr 2018.

1.7 Thesis outline

This thesis comprises seven chapters describing the stages of the research set out in the methodological approaches to achieve the predetermined objectives. A brief summary of contents of the thesis is given as follows:

Chapter 1 - This chapter presents the background, aims, objectives, contributions, and the methodological approaches of the research. In addition, some of the publications which are the output of this research are presented in this chapter.

Chapter 2 - This chapter describes the various uses of control systems and observer methods in rotorcraft based-UAVs both simulations and experiments, the development of research that has been conducted in the field of sliding mode control and observer in handling the chattering issue in these methods.

Chapter 3 - This chapter describes the kinematic and dynamic model of quadcopter UAVs using newton-euler formalism. In addition, some system identification techniques for obtaining the required quadcopter parameters are also presented in this chapter.

Chapter 4 - This chapter describes basic design of sliding mode control, role of set-point weighting function in the control system, design of sliding mode control for quadcopter UAVs, and methods to handle chattering issue in sliding mode control.

Chapter 5 - This chapter describes basic design of sliding mode observer, design of sliding mode observer for quadcopter UAV, and methods to handle chattering issue in sliding mode observer.

Chapter 6 - This chapter describes block diagram of the overall control and observer system, the simulation setup used for validating set-point integral quasi-sliding mode control that has been developed for quadcopter UAV, and performance criteria to evaluate the performance of the developed system. In addition, this chapter presents the results of simulation, comparative analysis, and discussion.

Chapter 7 - This chapter describes the simulation setup used for validating set-point integral sliding mode-based interval type-2 fuzzy logic control that has been developed for quadcopter UAV, and performance criteria to evaluate the performance of the developed system. In addition, this chapter presents the results of simulation, comparative analysis, and discussion.

Chapter 8 - This chapter describes the simulation setup used for validating set-point integral super-twisting algorithm of sliding mode control that has been developed for quadcopter UAV, and performance criteria to evaluate the performance of the developed system. In addition, this chapter presents the results of simulation, comparative analysis, and discussion.

Chapter 9 - This chapter describes the simulation setup used for validating set-point integral dynamic sliding mode control that has been developed for quadcopter UAV, and performance criteria to evaluate the performance of the developed system. In addition, this chapter presents the results of simulation, comparative analysis, and discussion.

Chapter 10 - This chapter describes block diagram of the overall control and observer system, the experimental setup used for validating the selected control systems and observer method for quadcopter UAV, and performance criteria to evaluate the performance of the developed system. In addition, this chapter presents the results of experimental work, comparative analysis, and discussion.

Chapter 11 - This chapter highlights the achievements of the research, drawback, and limitations. Future directions and recommendations are also presented in this chapter.

Chapter 2

State of the art of control and observer systems for quadcopter UAVs

2.1 Introduction

As mentioned in Chapter 1, nonlinear underactuated coupled mechanical systems have triggered many studies to develop proper control systems due to its unique properties. Although it shows many advantages, the structure of such systems lead to design of complex control algorithms. Opportunities to improve the control performance of such systems will be open due to the challenging characteristics of the system. Therefore, this research focuses on developing new control algorithms with full-order state observers to deal with the challenging issues of underactuated systems. To validate the performance of proposed control and observer methods, quadcopter UAVs are selected as one of the applications. The reason for choosing this application is that it has a high enough complexity to verify the performance of the proposed control and observer methods. Therefore, the previous works regarding the design of control systems and observers for quadcopter UAVs are highlighted in this literature review.

2.2 Quadcopter UAVs

Quadcopter UAVs create complex conditions, including: high nonlinear system, highly underactuated and coupled dynamics, and subject to disturbances and uncertainties. Therefore, the aircraft may need to have a good control formalism to tackle the conditions.

Previous works related to the control and observer system design utilized in this vehicle have been conducted to deal with such issues as presented in Table 2.1

Table 2.1. Previous works matrix of rotorcraft-based UAVs

Authors	Control Methods										Observer Methods					Disturbances			Output	
	PID	MPC	NN	SMC	LQR	T1FL	BS	Automata	FL	LMI	KF	NO	EO	TO	SMO	Wind gust	Noise	No	Simulation	Real imp.
Bouadi et al. 2007				X								X							X	
Benallegue et al. 2007									X						X	X	X		X	
Sanca et al. 2010							X												X	
Alexis et al. 2011		X									X					X				X
Hoffmann et al. 2011	X										X							X		X
Nicol et al. 2011			X																X	X
Zhang et al. 2011				X									X			X			X	
Cabecinhas et al. 2012	X							X								X			X	
Capello et al. 2012	X										X					X	X		X	
Grzonka et al. 2012	X										X							X		X
Hehn et al. 2012					X														X	
Leong et al. 2012	X				X													X	X	X
Olivares-Mendez at el. 2012	X					X													X	X
Sa and Corke, 2012	X																	X		X
Vega et al. 2012				X											X	X			X	
Adir and Stoica, 2012					X											X			X	
Marks et al. 2012	X															X			X	
Leishman et al. 2012					X						X							X		X
Alaimo et al. 2013	X																	X	X	
Liu et al. 2013					X											X			X	X
Alwi at al. 2013	X															X			X	
Satici at al. 2013							X												X	X
Elamine et al. 2013							X					X								
Baklanov et al. 2013					X											X			X	
Chee and Zhong, 2013	X															X				X
Argentim et al. 2013	X				X													X	X	
Chen et al. 2013	X										X							X		X
Escareno et al. 2013				X												X			X	
Fatan et al. 2013	X															X	X		X	
Honglei et al. 2013							X												X	X
Magnussen et al. 2013	X										X							X		X
Nagaty et al. 2013	X						X											X	X	
Nikolakopoulos et al. 2013										X						X			X	
Omurlu et al. 2013	X					X										X			X	X
Ryan at al. 2013										X							X			X
Cen et al. 2013	X													X		X				X
Belokon et al. 2013	X										X						X			X
Liu et al. 2014	X										X						X			X
Nemati et al. 2014	X																		X	
Sebesta et al. 2014	X										X						X			X
Zhao at al. 2014		X														X				X
Zheng et al. 2014				X												X			X	
Wang et al. 2014	X														X		X		X	
Alaimo et al. 2014	X																	X	X	
Magnusson, 2014	X															X			X	X
Bergman et al. 2014	X															X			X	X
Chamseddine et al. 2014					X															X
Wang et al. 2015									X						X					
Gonzalez- H. al. 2017				X											X			X		X
Ibarra et al. 2017				X														X	X	

2.3 Control system methods

2.3.1 Proportional-integral-derivative controller

Many research studies have looked into how to apply a conventional control method, namely proportional integral derivative (PID) control for controlling the attitude and altitude of quadcopter UAVs. This method is well-known due to its simplicity, generating good performance, and easiness in tuning without availability of plant model (Leong et al., 2012). Similar research has been conducted to apply this control technique to several rotorcraft-based UAVs, such as quadcopter, and hexacopter (Cabecinhas et al., 2012; Grzonka et al., 2012; Alaimo et al., 2013; Magnusson, 2014). The results show that they still have significant problems with steady state errors and overshoot around the reference. Therefore, these results have triggered further research to devise new methods that can enhance the performance of PID controller.

Capello et al. (2012) investigated the advantages of Kalman filter (KF) and L1 adaptive controller to improve the PID controller in dealing with uncertainties and disturbances of a quadcopter platform. They found that there was a better manoeuvring performance of such vehicle in three dimension (3D) axes despite some small oscillations appearing in speed throughout vertical take-off. Similar with Capello et al. (2012), an adaptive PID controller was proposed by Fatan et al. (2013) to overcome force disturbances, and present noise in quadcopter platform. They investigated steepest descent (SD) algorithm for online updating of the controller parameters and genetic algorithm (GA) as optimization function to improve the controller performance. Although GA can enhance the performance of PID, this method cannot deal with disturbances and noises shown by significant changes in tracking error. In addition, the performance of adaptation depends on the selection of learning rate. The lower learning rate tend to slow down the response of the controller to adapt to varying conditions, whereas higher learning rate could lead to instability.

An alternative method based on fuzzy logic was proposed by Ömürlü et al. (2013) to deal with any disturbances. A proportional derivative (PD) controller was used in adjusting the fuzzy PD parameters. This platform can deal with non-identical motor dynamics

by reaching the steady state around $\pm$ 2 degrees. Ryan and Kim (2013) built a robust controller to address the attitude stability of quadcopter subject to disturbances, nonlinearity, uncertainty, and integrator in saturation conditions. They investigated the controller based on approximate feedback linearization to enable the gain tuning of PID-like controller. They used linear matrix inequality (LMI)-based gain synthesis to synthesize the gains of the controller. The simulations show that there were oscillations in the yaw angle, which is caused by the difficulty in acquiring accurate estimate of yaw angle close to motor magnetic fields.

Liu et al. (2014) proposed a robust compensator to improve the PD performance for automatic VTOL system, and trajectory tracking of quadcopter. They applied the approach to an experimental prototype, and the results show that there were still significant deviations around the reference especially in the horizontal positions. Further approaches include hybrid controllers incorporating different types of estimator and PID controllers to deal with disturbances (Sa and Corke, 2012; Chee and Zhong, 2013; Magnussen et al., 2013; Bergman and Ekström, 2014). These platforms show better performance (both simulation and prototyping) in terms of tracking errors by showing a very small standard deviation in spite of slight overshoot around references.

There are some experimental investigations of applying hybrid PID controller to rotorcrafts reported without considering disturbances and uncertainties. Leong et al. (2012) proposed linear quadratic regulator (LQR) scheme to obtain appropriate parameters that will decrease the undesirable altitude deviation of quadcopter. Their proposed method was implemented experimentally, and the results show that the vehicle can be stable in hovering with a deviation of around $\pm$ 2 cm. Using a different method of Leong et al. (2012), Argentim et al. (2013) attempted to compare the performance of PID controller tuned by integral of time-weighted absolute error (ITAE), and LQR. Based on simulation, they found that LQR-based PID had the fastest response in vertical speed. However, in the time responses of vertical position, ITAE-based PID was better than others with overshoot of less than 10 percent. The backstepping (BS) approach was employed to cooperate with PID controller in controlling the attitude and altitude of quadcopter (Nagaty et al., 2013).

It still had a problem in controlling the pitch angle of the vehicle which resulted in increasing the linear acceleration and overshoot. Finally, a quaternion was proposed by Alaimo et al. (2014) and combined with PID controller in establishing the hexacopter orientation. They reported good controller performance in simulation results in tracking the reference. However, the simulations were carried out without any disturbances.

2.3.2 Sliding Mode Control

SMC has been popularly used as a robust control system to deal with nonlinearity, disturbances, and uncertainties. Zhang et al. (2011) investigated the advantages of SMC by modifying the control law outputs from discrete to continuous. The scheme was applied to quadcopter platform to control the attitude of the vehicle subject to disturbances. The extended observer was also proposed to approximate the disturbances. The numerical simulations show that the modified control scheme could deal with any kind of disturbances as well as no chattering around sliding surface. Meanwhile, Adir and Stoica (2012) attempted to improve their prior work by changing the control strategy. An SMC scheme was proposed to address the stabilization of the rotorcraft's movement. Despite achieving positive results, the effectiveness of this control system was simply shown by numerical simulations.

A novel control design was proposed by Luque-Vega et al. (2012) in tracking and controlling the attitude and absolute position of quadcopter. They used the block control (BC) algorithm as a main controller to generate a smooth control input for altitude state. A super-twisting SMC estimator was used to ensure the robustness of rotorcraft when unmatched perturbations appeared. According to simulations, the controller was able to deny sufficiently the aerodynamic force and moment disturbance. Escareño et al. (2013) investigated the robustness of adaptive SMC technique in dealing with various wind conditions of quadcopter vehicle. This technique was also demonstrated in simulation. Good performance was achieved in tracking the references in any situations. Meanwhile, Zheng et al. (2014) designed 2nd order SMC for tracking control in both position and attitude to deal with wind gust. In the simulations, this method shows robustness in tackling the change of

reference values and disturbances.

2.3.3 Linear Quadratic Regulator

LQR has widely been used for controlling various plants. Adir and Stoica (2012) proposed an LQR control method and waypoint navigation to improve the behaviour and altitude stability of octorotor UAV. Their simulations show that the method can deal with payload change by reaching the reference with stable and fast enough response. Liu et al. (2013) attempted to combine robust compensator and LQR to control the tracking errors of quadcopter. Parameter uncertainties and wind gust were considered in designing this control method. The results show that the tracking errors were small enough, less than 0.1 degree in both numerical simulation and real application.

2.3.4 Backstepping controller

BS is a control method that can deal with nonlinearity. The method is based on a recursive Lyapunov approach, which has been popular in the early 1990s (Sanca et al., 2010). Sanca et al. (2010) investigated such a scheme to control the manoeuvring and hovering of hexacopter vehicle. In simulations, the controller has shown good performance in error tracking.

Other researchers have attempted to apply BS using inverse optimal control referred to as BS - based inverse optimal attitude control in quadcopter (Honglei et al., 2013). They investigated this method in both simulation and prototyping. In comparing with PID control, this technique could perform faster although their vehicle had limitations of maximum torque. Unfortunately, this performance was without consideration of disturbances.

Furthermore, Nagaty et al. (2013) investigated the ability of BS controller, and the results show that this method did not maintain the stability of pitch action and showed significant overshoot around references.

2.3.5 Model Predictive Control

MPC may give a promising performance in all areas of control design (Zhao and Go, 2014). Alexis et al. (2011) have investigated the capability of MPC to control the attitude of quadcopter subject to wind disturbances. A hybrid MPC with piecewise affine (PWA), and extended Kalman filter (EKF) was proposed to enhance the ability of such controller. Experimentally, the influence of wind gust on dynamic behaviour of such vehicle is significantly evident. However, this approach is still better than PID/LQR control. They later proposed a similar method, but with a modified experimental setup. The results still did not have significant differences (Alexis et al., 2012).

Robust feedback linearisation (FL) was introduced by Zhao and Go (2014) to improve the ability of MPC. The technique was applied to the quadcopter control. They attempted practical implementation with known obstacles, and the results show that the method could achieve satisfactory performance in maintaining the desired formation and avoiding the obstacles.

2.3.6 Other control methods

In addition to the control approaches mentioned above, there are further methods applied to rotorcraft-based UAVs. For example, Hehn et al. (2012) investigated time-optimal technique to control manoeuvring and hovering of quadcopter. This scheme still cannot perform as expected, inaccuracies in trajectories of pitch angle are noted and the system loses acceleration when the speeds go higher. Olivares-Mendez et al. (2012) used visual servoing-based type-1 fuzzy logic control on quadcopter platform to avoid obstacles when manoeuvring. Cross-entropy theory was applied to optimise the parameters of such controller. They found small time response and errors to achieve references during manoeuvring.

Baklanov and Morozov (2013) applied quadratic cost function to control a quadcopter to have a stable motion throughout arbitrary trajectories in 3D space. They considered air resistance in designing such controller. They found that there were still overshoot and steady state errors in pitch, yaw, and roll of the vehicle. An LMI scheme was proposed

by Nikolakopoulos and Alexis (2013) for controlling the attitude of quadcopter to deal with additive disturbances. There were no adequate data to show the performance of such controller. However, they stated that their control scheme had good performance to control such vehicle.

2.3.7 Summary of control systems survey

Regarding the implementation of several control techniques to rotorcraft systems, from the previous works which have been reviewed it can be summarized that LQR and SMC have more contribution to better performance of rotorcraft-based UAVs than other methods. However, in terms of dealing with nonlinearity, underactuated systems, uncertainties and disturbances, SMC outperforms LQR. This is because the SMC scheme is designed to be insensitive in such issues by employing sliding surface. Sliding surface which is designed in the control method has a role in maintaining trajectory on the surface for subsequent time at any conditions.

SMC is optimal control-based approach which needs the availability of all internal states of plant to be fed back to input. This issue can be a matter for development of experimental validation of such controllers. Therefore, the use of appropriate sensors for acquiring all required states or a method for estimating unknown states is another work to be considered for further developments.

2.4 State observer

Although information on states can be obtained accurately by high precision sensor, such as VICON motion capture as conducted by Satici et al. (2013), the system will become more expensive. Unmeasured states (such as velocity) can be acquired as well by other sensors, but the information is usually corrupted by noise. Moreover, by increasing the number of sensors, the system will become more expensive and complex (Benallegue et al., 2008). Therefore, the use of an observer can be a solution to address such problems.

Previous works have included the use of different types of observer methods in quad-

copter UAVs as shown in Table 2.1. These include nonlinear observer (Bouadi and Tadjine, 2007; Elamine et al., 2013); sliding mode observers (SMOs) (Benallegue et al., 2008; Luque-Vega et al., 2012; Wang and Shirinzadeh, 2015); Thau observer Cen et al. (2014) and Kalman filter (KF) (Hoffmann et al., 2011; Alexis et al., 2011; Leishman et al., 2012; Grzonka et al., 2012; Capello et al., 2012; Liu et al., 2013; Magnussen et al., 2013; Belokon' et al., 2013; Sebesta and Boizot, 2014). Although results show that the methods can improve the performance of the controller, some shortcomings are still associated with such methods. The fundamental disadvantage of nonlinear observer is connected to the expanding power of the gain parameter which leads to hard task numerical execution when the gain and the system order are very large (Astolfi and Marconi, 2015). Furthermore, a comparative study between SMO and KF was conducted by Chen and Dunnigan (2002). They pointed out that SMO is more robust than KF in terms of parameter uncertainty and noise cancellation. Meanwhile, Thau observer may not deal with error model of the system (Schröder et al., 2000). However, although SMO has shown better performance, the observer still has associated problem due to chattering, which affects its overall performance with noise measurement.

2.5 Summary

Various control systems and observer methods have been developed for performance enhancement of quadcopter UAVs. Among these, The SMC and SMO schemes show better performance in terms of robustness. However, the chattering phenomenon that may affect the system performance, is still an issue to be resolved. Some methods have been proposed to eliminate this phenomenon, such as boundary layer method (Baik et al., 2000; Xu, 2007; Shtessel et al., 2010); higher order sliding mode (Benallegue et al., 2008; Ozer et al., 2017; Gonzalez-Hernandez et al., 2017); super-twisting algorithm (Salgado et al., 2011; Luque-Vega et al., 2012; Ibarra and Castillo, 2017); and dynamic sliding mode control (DSMC) (Liu and Wang, 2011; Wang et al., 2017b). Basically, those methods can tackle the problem of chattering. However, this does not mean they are free from shortcoming. The

boundary layer method might not be effective in reducing chattering when systems require high accuracy control signals. Although higher order sliding mode approach can perform better than boundary layer method, the technique still cannot address such problem in n-th derivative states. Super-twisting algorithm is more effective than higher order sliding mode approach in eliminating chattering, but such method still cannot offset uncertainty and disturbance on state variables. Among the methods, DSMC has a better way to deal with noise measurement as well as chattering reduction. However, this approach has a difficulty to evaluate a new sliding variable and to tune the parameters to provide a system stability.

Ultimately, the research is to design a novel sliding mode-based nonlinear control system with non-linear full-order states observer to enhance the performance of the system and apply to quadcopter-based UAVs, investigate several methods of sliding mode control systems and nonlinear observers, and validate it in numerical simulations. Comparative studies will be carried out to show the performance of controllers and observers. Furthermore, experimental works will be conducted to verify the performance of selected control systems and observer method in real-time application.

Chapter 3

System modelling and identification of quadcopter UAVs

This chapter presents modelling and identification of a quadcopter. A Newton-Euler formalism is utilized to derive the kinematic and dynamic models of such vehicle with the assumption that:

- The construction is symmetrical and inflexible.
- The quadcopter's center of mass coincides with a body-fixed frame.
- The propellers are stiff.
- Drag and thrust force of the vehicle are relative to the square of propeller's speed.

Aerodynamic effects working on the quadcopter are described in this chapter as well as rotor dynamics of the rotorcraft. Formulation of the quadcopter state space model is determined for designing control and observer in subsequent chapters. This chapter will be ended by system identification to obtain quadcopter parameters, including moment of inertia, thrust force, motor speed, and power consumption.

3.1 Kinematic model

A quadcopter UAV as illustrated in Figure 3.1 has 4 fixed rotary blades connected to motors and is lifted by total forces of F_i $(i = 1, 2, 3, 4)$. This rotorcraft is formed with two clockwise pivoting rotors (2,4), and two counter-clockwise turning rotors (1,3), so that by increasing the speed of rotors at a similar rate will build the height of rotorcraft, and

the other way around. Forward movement can be accomplished by decreasing the rotor speed (1) and increasing the speed of rotor (3) at the same time. In a similar way reverse, rightward, and leftward movements can be realised. Moreover, the yaw movement can be achieved by accelerating or slowing down the rotation of clockwise rotors which rely upon yaw heading. Nevertheless, the aggregate energy of this rotorcraft must have the same value to keep up the height level of the vehicle.

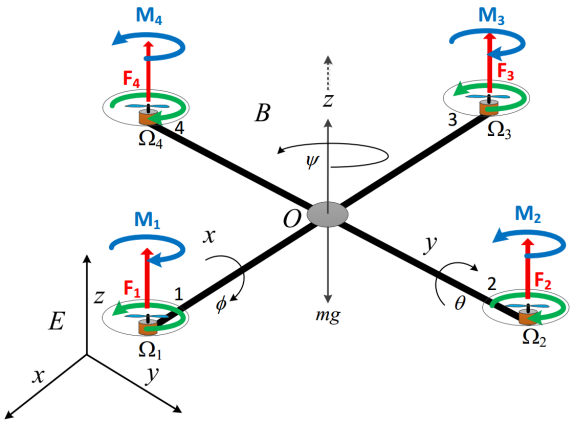

Figure 3.1. + type of quadcopter UAVs general model

The framework of the vehicle can be isolated into two sections, to be specific the "body frame" $B-(Oxyz)$, and the "earth frame" $E-(Oxyz)$. m, g, and l designate the rotor-craft mass, gravity acceleration, and the distance of each rotor to centre of gravity of rotor-craft respectively. The orientation of the rotocraft from body frame to earth frame is represented by rotation matrices of $R : e \longrightarrow B$, where these matrices depend on the Euler angle (ϕ, θ, ψ). Two intermediate coordinate systems are defined to illustrate the quadcopter orientation in space: the rotorcraft-1 $(r1)$ frame and the rotorcraft-2 $(r2)$ frame together with earth frame (E) and body frame (B)(Elkholy, 2014).

The earth frame is turned around its y-axis to obtain the $r1$ frame by the yaw angle. The matrix transformation for turning the earth to $r1$ frame is defined as,

$$R(r1, E) = \begin{bmatrix} cos\psi & sin\psi & 0 \\ -sin\psi & cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3.1}$$

The $r1$ frame is then pivoted around its y-axis by the pitch angle θ to obtain the $(r2)$

frame. The matrix transformation for turning the $r1$ to the $r2$ frame is defined as,

$$R(r2, r1) = \begin{bmatrix} cos\theta & 0 & -sin\theta \\ 0 & 1 & 0 \\ sin\theta & 0 & cos\theta \end{bmatrix} \tag{3.2}$$

To obtain the body frame B, the $r2$ frame is turned around its x-axis. Hence, The matrix transformation is defined as,

$$R(B, r2) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & cos\phi & sin\phi \\ 0 & -sin\phi & cos\phi \end{bmatrix} \tag{3.3}$$

By using equations (3.1) - (3.3), the transformation matrix to alter from the earth to the body frame is expressed as,

$$R(B, E) = R(B, r2)R(r2, r1)R(r1, E) \tag{3.4}$$

therefore,

$$R(B, E) = \begin{bmatrix} c\theta c\psi & c\theta s\psi & -s\theta \\ c\psi s\phi s\theta - c\phi s\psi & c\phi c\psi + s\phi s\psi s\theta & c\theta s\phi \\ s\phi s\psi + c\phi c\psi s\theta & s\phi s\psi s\theta - c\psi s\phi & c\phi c\theta \end{bmatrix} \tag{3.5}$$

The angles (ϕ, θ, ψ) represent roll, pitch, and yaw respectively, while s, c, and t represent $sine$, cos, and $tangent$ respectively. The movement of roll (ϕ), pitch (θ) and yaw (ψ) are bounded as: $(-\pi/2 < \phi < \pi/2)$, $(-\pi/2 < \theta < \pi/2)$ and $(-\pi < \psi < \pi)$ respectively.

The rotation matrix $R(B, E)$ obtained above can be transposed to obtain transformation from body frame to earth frame as,

$$R = R(B, E)^T = R(E, B) \tag{3.6}$$

Hence, the new rotation matrix is given as,

$$R = \begin{bmatrix} c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & s\phi s\psi + c\phi s\theta c\psi \\ c\theta s\psi & c\phi c\psi + s\phi s\psi s\theta & c\phi s\psi s\theta - s\phi c\psi \\ -s\theta & s\phi c\theta & c\phi c\theta \end{bmatrix} \tag{3.7}$$

3.2 Dynamic model

The system dynamics of quadcopter is nonlinear and comprises two multiple subsystems including fully-actuated subsystem which is rotational motions ($roll$, $pitch$ and yaw) and z axis movement, whereas translational movements in x and y axes form the under-actuated subsystem.

3.2.1 Rotational motions

The equations of rotational movement in the body frame are obtained by utilizing the general equation of $Newton - Euler$ method as,

$$J\dot{\omega} + \omega \times J\omega + M_G = M_B \tag{3.8}$$

where

J	The inertia matrix of quadcopter
ω	Body angular velocity
M_G	Gyroscopic moments because of inertia produced by rotors' motion
M_B	Moments working on the body frame of quadcopter.

Furthermore, the gyroscopic moment of the rotorcraft is defined as $\omega \times [0 \quad 0 \quad J_r\Omega_r]^T$, thus the rotational motion of the quadcopter can be expressed as,

$$J\dot{\omega} + \omega \times J\omega + \omega \times [0 \quad 0 \quad J_r\Omega_r]^T = M_B \tag{3.9}$$

where

J_r	rotors' inertia

Ω_r residual of angular rotors velocity $\Omega_r = -\Omega_1 + \Omega_2 - \Omega_3 + \Omega_4$

The quadcopter is designed to be symmetrical so that its inertial matrix for each component x, y and z is diagonal while others are zero as,

$$J = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \tag{3.10}$$

where I_{xx}, I_{yy} and I_{zz} are the mass moments of inertia for x, y and z axes respectively in the body frame reference.

There are two physical effects which produced by rotors forces: aerodynamic force F_i and aerodynamic moments M_i as illustrated in Figure 3.1. The aerodynamic force is the force applied on a body of rotorcraft by the air in which the body is submerged, and is because of the relative motion between the body and the air generated by rotors. An aerodynamic moment is the reactive torque caused by the rotor motion on the air. Those aerodynamic effects are defined as,

$$F_i = \frac{1}{2}\rho A C_T r^2 \Omega_i^2 \tag{3.11}$$

$$M_i = \frac{1}{2}\rho A C_D r^2 \Omega_i^2 \tag{3.12}$$

where

ρ air density

A blade area

C_T, C_D coefficients for aerodynamic

r blade radius

Ω_i angular velocity of rotor i

Since the aerodynamic moments and forces rely on the air density and the propeller geometry, and the maximum altitude of the quadcopter is typically restricted, so that the density

of air is considered constant. Therefore, the equations (3.11) and (3.12) are simplified as,

$$F_i = K_f \Omega_i^2 \tag{3.13}$$

$$M_i = K_M \Omega_i^2 \tag{3.14}$$

where K_f, K_M and Ω_i are respectively the aerodynamic force constant, moment constant and the angular velocity of rotor i. The constants of those parameters can be acquired by experiment.

The quadcopter moments M_B can be obtained by determining the aerodynamic forces and moments resulted by propellers. Figure 3.1 illustrates the aerodynamic forces and moments working on the quadcopter. Each propeller results in an upwards thrust force F_i and produces a moment M_i with the opposite direction of the rotational direction of the corresponding rotor i.

In the body frame's x-axis, the rolling torque derived by multiplying F_2 with the arm length l of rotorcraft produces a negative moment in the x-axis, while in the same way, F_4 produces a positive moment. Hence, the rolling torque is defined as,

$$\begin{aligned} M_\phi &= -F_2 l + F_4 l \\ &= -(K_f \Omega_2^2) l + (K_f \Omega_4^2) l \\ &= l K_f (-\Omega_2^2) + \Omega_4^2)) \end{aligned} \tag{3.15}$$

Furthermore, in the body frame's y-axis, the thrust of rotor 1 produces a positive moment, while rotor 3 delivers a negative moment in the y-axis. Therefore, the pitching torque is defined as,

$$\begin{aligned} M_\theta &= F_1 l - F_3 l \\ &= (K_f \Omega_1^2) l - (K_f \Omega_3^2) l \\ &= l K_f (\Omega_1^2) - \Omega_3^2)) \end{aligned} \tag{3.16}$$

For the motion in the body frame's z-axis, The moment caused by the rotors' rotation is presented in equation (3.14). Thus, the yawing torque in the body frame's z-axis is defined as,

$$\begin{aligned} M_\psi &= M_1 - M_2 + M_3 - M_4 \\ &= (K_M\Omega_1^2) - (K_M\Omega_2^2) + (K_M\Omega_3^2) - (K_M\Omega_4^2) \\ &= K_M(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \end{aligned} \tag{3.17}$$

In vector form, the moments expressed in equations (3.15), (3.16) and (3.17) can be presented as,

$$M_B = \begin{bmatrix} lK_f(-\Omega_2^2) + \Omega_4^2)) \\ lK_f(\Omega_1^2) - \Omega_3^2)) \\ K_M(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \end{bmatrix} \tag{3.18}$$

where l is the distance between the centre of rotation of each rotor and central point of the body reference frame of the quadcopter UAV.

3.2.2 Translational motions

The translation equations of quadcopter motion in the earth frame are derived by utilizing Newton's second law as,

$$m\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + RF_B \tag{3.19}$$

where

$r = [x\ y\ z]^T$	Quadcopter's distance from the Earth frame
m	Quadcopter's mass
g	Gravitational acceleration g = 9.81 m/s^2
R	Rotation matrix
F_B	Non-gravitational forces working on the quadcopter in the body frame

When the quadcopter is in a steady-state condition, the only non-gravitational force acting on it, is the thrust generated by motion of propellers as defined in equation (3.13). Thus, the non-gravitational force acting on the quadcopter, F_B, is given as,

$$F_B = \begin{bmatrix} 0 \\ 0 \\ K_f(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \end{bmatrix} \tag{3.20}$$

The zero values in the force vector as shown in equation (3.20) indicate that there is no force in the X and Y headings, while the third line in the equation is essentially an expansion of the thrust forces delivered by the four propellers. Furthermore, to obtain the thrust forces of the rotors in the Earth frame, F_B should be multiplied by the rotation matrix R, so that the equation is applicable for any orientation of the quadcopter UAV

3.3 Aerodynamic effects

In the previous sections, the aerodynamic effects acting on the quadcopter body were ignored. Nevertheless, these effects should be incorporated into the model to provide realistic and accurate model for use in simulation. There are two types of aerodynamic effects, namely drag forces and drag moments.

3.3.1 Drag forces

A force acts on the body of the quadcopter will resist movement of such vehicles caused by friction between a moving quadcopter and air. Drag force will increase as the speed of quadcopter movement increases. Therefore, the drag forces F_a can be approached by,

$$F_a = K_t \dot{r} \tag{3.21}$$

where K_t and $\dot{r}$ represent coefficient matrix of aerodynamic translation and the time derivative of the position vector r respectively. This shows that there is an additional force acting on the quadcopter body. The translational motion in equation (3.19) can be rewritten as,

$$m\ddot{r} = \begin{bmatrix} 0 \\ 0 \\ -mg \end{bmatrix} + RF_B - F_a \tag{3.22}$$

3.3.2 Drag moments

Similar to the drag force, a drag moment M_a acting on the quadcopter body caused by air friction, is approximated as,

$$M_a = K_r \dot{\eta} \tag{3.23}$$

where K_r and $\dot{\eta}$ denote coefficient matrix of aerodynamic rotation and Euler rates respectively. Therefore, the rotational motion as shown in equation (3.9) is rewritten to as,

$$J\dot{\omega} + \omega \times J\omega + \omega \times [0 \quad 0 \quad J_r\Omega_r]^T = M_B - M_a \tag{3.24}$$

3.4 Rotor dynamics

Brushless DC motor which has high torque and small friction is a typical motor used in quadcopters. A brushless DC motor and a conventional DC motor have the same dynamic model at steady state condition. The dynamic model of brushless DC motor is derived by assuming that the rotor is gear-less with stiff mechanical coupling between the shaft of motor and propeller. The internal schematic of a brushless DC motor at steady state is depicted in Figure 3.2,

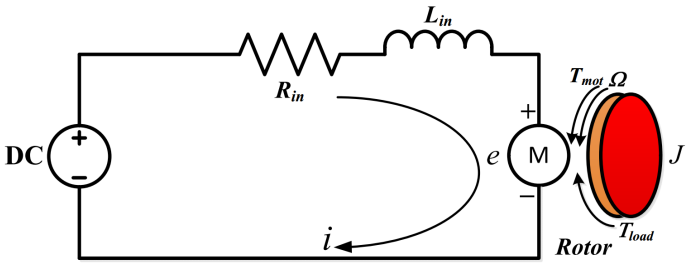

Figure 3.2. Internal schematic of a brushless DC motor

The Kirchhoff's voltage law of DC motor is thus,

$$v_{DC} = R_{in}i + L_{in}\frac{di}{dt} + K_{mot}\Omega \tag{3.25}$$

where R_{in} and L_{in} are respectively the resistance and inductance of motor, while i and v_{DC} are the armature current and the input voltage respectively. $K_{mot}\Omega$ represents generated emf e with K_{mot} as the constant of motor torque. Since the quadcopter is constructed with

small motors where the inductance is very small, equation (3.25) can be expressed as,

$$v_{DC} = R_{in} i + K_{mot} \Omega_i \tag{3.26}$$

or

$$i = \frac{v_{DC} - K_{mot} \Omega_i}{R_{in}} \tag{3.27}$$

By utilizing the Newton's 2^{nd} law, the relationship between torque and angular velocity of propeller is defined as,

$$J\dot{\Omega}_i = T_{mot} - T_{load} \tag{3.28}$$

where T_{mot} is the torque generated by the motor which is equal to $K_e i$. K_e is the motor's electric constant and considered as equal to K_{mot} for small motor. T_{load} is the load torque produced from the propeller system. Using equations (3.14) and (3.27), the angular velocity of the propeller can be expressed as,

$$J\dot{\Omega}_i = K_{mot} \frac{v_{DC} - K_{mot} \Omega_i}{R_{in}} - K_M \Omega_i^2 \tag{3.29}$$

Therefore,

$$v_{DC} = \frac{R_{in}}{K_{mot}} J\dot{\Omega}_i + K_{mot} \Omega_i + K_M R_{in} \Omega_i^2 \tag{3.30}$$

Brushless DC motor is typically equipped with embedded controllers so that the motor work with a pulse width modulation (PWM) signal. Furthermore, the voltage supplied to the motor is directly related proportionally to the speed of propeller rotation. The relationship can be obtained by a system identification process for the motor and propeller in pairs.

3.5 State space representation

A state-space model is a numerical model of a physical system as a set of input, output and state variables associated by first-order differential equations. ”State space” refers to the Euclidean space in which the variables on the axes are the state variables. The state of the system can be represented as a vector within that space.

3.5.1 State vector X

It is important to determine the state vector of quadcopter UAVs before designing the state observer for the system. Hence, the state vector of the system is defined as,

$$X = [x_1 \;\; x_2 \;\; x_3 \;\; x_4 \;\; x_5 \;\; x_6 \;\; x_7 \;\; x_8 \;\; x_9 \;\; x_{10} \;\; x_{11} \;\; x_{12}]^T \tag{3.31}$$

The state vector is mapped to the degrees of freedom of the quadcopter, as expressed,

$$X = [\phi \quad \dot{\phi} \quad \theta \quad \dot{\theta} \quad \psi \quad \dot{\psi} \quad z \quad \dot{z} \quad x \quad \dot{x} \quad y \quad \dot{y}]^T \tag{3.32}$$

The state vector defines the orientations and positions of the quadcopter in space and its angular and linear velocities.

3.5.2 Control input *u*

A vector of control input of quadcopter UAVs is defined as,

$$u = [u_1 \quad u_2 \quad u_3 \quad u_4]^T \tag{3.33}$$

where

$$u_1 = K_f(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2) \tag{3.34}$$

$$u_2 = K_f(-\Omega_2^2 + \Omega_4^2) \tag{3.35}$$

$$u_3 = K_f(\Omega_1^2 - \Omega_3^2) \tag{3.36}$$

$$u_4 = K_M(\Omega_1^2 - \Omega_2^2 + \Omega_3^2 - \Omega_4^2) \tag{3.37}$$

In a matrix form, equations (3.34) - (3.37) can be expressed as,

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} K_f & K_f & K_f & K_f \\ 0 & -K_f & 0 & K_f \\ K_f & 0 & -K_f & 0 \\ K_M & -K_M & K_M & -K_M \end{bmatrix} \begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} \tag{3.38}$$

From equation (3.28), it can be deduced that u_1 is the resultant upward force of four rotors which has a role in maintaining and change the altitude of the quadcopter and its rate of change (z, $\dot{z}$). u_2 is the thrust difference between two rotors, namely rotor 2 and 4 which will cause roll motions (ϕ, $\dot{\phi}$). u_3 is the thrust difference between two rotors, namely rotor 1 and 3 to generate pitch motions (θ, $\dot{\theta}$). Finally u_4 is the torque difference between two clockwise and two counter-clockwise rotating rotors which will cause yaw motion (ψ, $\dot{\psi}$).

Velocities of rotors are obtained by inverting the matrix equation (3.38) as,

$$\begin{bmatrix} \Omega_1^2 \\ \Omega_2^2 \\ \Omega_3^2 \\ \Omega_4^2 \end{bmatrix} = \begin{bmatrix} \frac{1}{4K_f} & 0 & \frac{1}{2K_f} & \frac{1}{4K_m} \\ \frac{1}{4K_f} & -\frac{1}{2K_f} & 0 & -\frac{1}{4K_m} \\ \frac{1}{4K_f} & 0 & -\frac{1}{2K_f} & \frac{1}{4K_m} \\ \frac{1}{4K_f} & \frac{1}{2K_f} & 0 & -\frac{1}{4K_m} \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} \tag{3.39}$$

Thus

$$\begin{aligned} \Omega_1 &= \sqrt{\frac{1}{4K_f}u_1 + \frac{1}{2K_f}u_3 + \frac{1}{4K_M}u_4} \\ \Omega_2 &= \sqrt{\frac{1}{4K_f}u_1 - \frac{1}{2K_f}u_2 - \frac{1}{4K_M}u_4} \\ \Omega_3 &= \sqrt{\frac{1}{4K_f}u_1 - \frac{1}{2K_f}u_3 + \frac{1}{4K_M}u_4} \\ \Omega_4 &= \sqrt{\frac{1}{4K_f}u_1 + \frac{1}{2K_f}u_2 - \frac{1}{4K_M}u_4} \end{aligned} \tag{3.40}$$

In terms of thrust force, the control input can expressed as,

$$
\begin{aligned}
u_1 &= F_1 + F_2 + F_3 + F_4 \\
u_2 &= -F_2 + F_4 \\
u_3 &= F_1 - F_3 \\
u_4 &= d(F_1 - F_2 + F_3 - F_4)/b
\end{aligned}
\tag{3.41}
$$

where b, and d represent the lift coefficient and the scaling factor of force to moment. Equation (3.41) can be expressed in matrix form as,

$$
\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 0 & -1 & 0 & 1 \\ 1 & 0 & -1 & 0 \\ \frac{d}{b} & -\frac{d}{b} & \frac{d}{b} & -\frac{d}{b} \end{bmatrix} \begin{bmatrix} F_1 \\ F \\ F_3 \\ F_4 \end{bmatrix}
\tag{3.42}
$$

By inverting equation (3.42), the thrust force required for each rotor is obtained as,

$$
\begin{aligned}
F_1 &= \frac{1}{4}u_1 + \frac{1}{2}u_3 + \frac{b}{4d}u_4 \\
F_2 &= \frac{1}{4}u_1 - \frac{1}{2}u_2 - \frac{b}{4d}u_4 \\
F_3 &= \frac{1}{4}u_1 - \frac{1}{2}u_3 + \frac{b}{4d}u_4 \\
F_4 &= \frac{1}{4}u_1 + \frac{1}{2}u_2 - \frac{b}{4d}u_4
\end{aligned}
\tag{3.43}
$$

3.5.3 Rotational motions

Considering equations (3.18) and (3.35) - (3.37), the total moments acting on the quadcopter can be expressed as,

$$
M_B = \begin{bmatrix} lu_2 \\ lu_3 \\ u_4 \end{bmatrix}
\tag{3.44}
$$

Furthermore, substituting for M_B from equation (3.44) into the rotational motion equation (3.9) and expanding each term, the new rotational motion can be expressed as,

$$\begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \ddot{\phi} \\ \ddot{\theta} \\ \ddot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix} \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} + \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ J_r\Omega_r \end{bmatrix} = \begin{bmatrix} lu_2 \\ lu_3 \\ u_4 \end{bmatrix} \tag{3.45}$$

and,

$$\begin{bmatrix} I_{xx}\ddot{\phi} \\ I_{yy}\ddot{\theta} \\ I_{zz}\ddot{\psi} \end{bmatrix} + \begin{bmatrix} I_{zz}\dot{\theta}\dot{\psi} - I_{yy}\dot{\theta}\dot{\psi} \\ I_{xx}\dot{\phi}\dot{\psi} - I_{zz}\dot{\phi}\dot{\psi} \\ I_{yy}\dot{\theta}\dot{\phi} - I_{xx}\dot{\theta}\dot{\phi} \end{bmatrix} + \begin{bmatrix} \dot{\theta}J_r\Omega_r \\ -\dot{\phi}J_r\Omega_r \\ 0 \end{bmatrix} = \begin{bmatrix} lu_2 \\ lu_3 \\ u_4 \end{bmatrix} \tag{3.46}$$

Therefore, the angular accelerations of the rotorcraft can be presented as,

$$\ddot{\phi} = \dot{\psi}\dot{\theta}\left(\frac{I_{yy} - I_{zz}}{I_{xx}}\right) - \frac{J_r}{I_{xx}}\dot{\theta}\Omega_r + \frac{l}{I_{xx}}u_2 \tag{3.47}$$

$$\ddot{\theta} = \dot{\phi}\dot{\psi}\left(\frac{I_{zz-I_{xx}}}{I_{yy}}\right) + \frac{J_r}{I_{yy}}\dot{\phi}\Omega_r + \frac{l}{I_{yy}}u_3 \tag{3.48}$$

$$\ddot{\psi} = \dot{\phi}\dot{\theta}\left(\frac{I_{xx} - I_{yy}}{I_{zz}}\right) + \frac{1}{I_{zz}}u_4 \tag{3.49}$$

From the rotational motion equations, it can be seen that each control input corresponds directly to each rotational motion state. Therefore, these states are considered as fully-actuated subsystems.

3.5.4 Translational motions

The non-gravitational force acting on the quadcopter related to control input can be obtained by substituting for angular velocities of rotors from equations (3.34) - (3.37) and into equation (3.20) as,

$$F_B = \begin{bmatrix} 0 \\ 0 \\ u_1 \end{bmatrix} \tag{3.50}$$

Therefore, the translational equation can be obtained by substituting for R and F_B from the equations (3.7) and (3.50) into equation (3.19),

$$m\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix} = \begin{bmatrix}0\\ 0\\ -mg\end{bmatrix} + \begin{bmatrix}c\theta c\psi & s\phi s\theta c\psi - c\phi s\psi & s\phi s\psi + c\phi s\theta c\psi\\ c\theta s\psi & c\phi c\psi + s\phi s\psi s\theta & c\phi s\psi s\theta - s\phi c\psi\\ -s\theta & s\phi c\theta & c\phi c\theta\end{bmatrix}\begin{bmatrix}0\\ 0\\ u_1\end{bmatrix} \tag{3.51}$$

or

$$m\begin{bmatrix}\ddot{x}\\ \ddot{y}\\ \ddot{z}\end{bmatrix} = \begin{bmatrix}0\\ 0\\ -mg\end{bmatrix} + \begin{bmatrix}s\phi s\psi + c\phi s\theta c\psi(u_1)\\ c\phi s\psi s\theta - s\phi c\psi(u_1)\\ c\phi c\theta(u_1)\end{bmatrix} \tag{3.52}$$

The dynamic system of translational motion in equation (3.52) can be rewritten as,

$$\ddot{x} = \frac{u_1}{m}(\sin\phi\sin\psi + \cos\phi\sin\theta\cos\psi) \tag{3.53}$$

$$\ddot{y} = \frac{u_1}{m}(\cos\phi\sin\psi\sin\theta - \sin\phi\cos\psi) \tag{3.54}$$

$$\ddot{z} = -g + \frac{u_1}{m}(\cos\phi\cos\theta) \tag{3.55}$$

Rewritten in the state variable X is expressed as,

$$\ddot{x} = \frac{u_1}{m}(\sin x_1\sin x_5 + \cos x_1\sin x_3\cos x_5) \tag{3.56}$$

$$\ddot{y} = \frac{u_1}{m}(\cos x_1\sin x_5\sin x_3 - \sin x_1\cos x_5) \tag{3.57}$$

$$\ddot{z} = -g + \frac{u_1}{m}(\cos x_1\cos x_3) \tag{3.58}$$

It can be seen that translational motions in x and y directions depend on rotational motion states (θ and ϕ), while control signal u_1 is maintaining the altitude at the same level. Therefore, x and y motions are considered as under-actuated subsystem while z is fully-actuated subsystem.

3.5.5 State space representation

From the state vectors in equations (3.31) and (3.32), rotational motion in equations (3.47) - (3.49) and translational motions in equations (3.53) - (3.55), the dynamic model of the quadcopter in state space representation can be expressed as,

$$f(x,u) = \begin{bmatrix} \dot{x_1} \\ \dot{x_2} \\ \dot{x_3} \\ \dot{x_4} \\ \dot{x_5} \\ \dot{x_6} \\ \dot{x_7} \\ \dot{x_8} \\ \dot{x_9} \\ \dot{x_{10}} \\ \dot{x_{11}} \\ \dot{x_{12}} \end{bmatrix} = \begin{bmatrix} x_2 \\ x_4 x_6 (\frac{I_{yy} - I_{zz}}{I_{xx}}) - \frac{J_r}{I_{xx}} x_4 \Omega_r + \frac{l}{I_{xx}} u_2 \\ x_4 \\ x_2 x_6 (\frac{I_{zz} - I_{xx}}{I_{yy}}) - \frac{J_r}{I_{yy}} x_2 \Omega_r + \frac{l}{I_{yy}} u_3 \\ x_6 \\ x_2 x_4 (\frac{I_{xx} - I_{yy}}{I_{zz}}) + \frac{1}{I_{zz}} u_4 \\ x_8 \\ -g + \frac{u_1}{m} (\cos x_1 \cos x_3) \\ x_{10} \\ \frac{u_1}{m} (\sin x_1 \sin x_5 + \cos x_1 \sin x_3 \cos x_5) \\ x_{12} \\ \frac{u_1}{m} (\cos x_1 \sin x_5 \sin x_3 - \sin x_1 \cos x_5) \end{bmatrix} \tag{3.59}$$

3.6 System identification

System identification utilizes statistical approaches to build numerical models of dynamical systems from measurement data and experiment. System identification also incorporates an optimal experimental design to generate informative data efficiently for fitting such models as well as model reduction.

In this section, presents methods of obtaining the required parameters of the quadcopter model. Moment of inertia is identified by calculation and bifilar pendulum approach. Moreover, certain motor and propeller parameters are acquired using a device designed by RC benchmark company.

3.6.1 Moment of inertia identification method

The moment of inertia, also called rotational inertia or angular mass, of an inflexible body is a tensor that specifies the torque required for a desired angular acceleration of a rotational axis. It relies upon the distribution of body mass and the selected axis, with larger moments needing more torque to alter body's rotation.

The mass moment of inertia J of an object plays much the same role in rotational motions as in translational motions. The mass moment of inertia determines how rotational velocity is influenced by applied torque. This obviously relies on the mass of the object and how it is distributed around the axis of rotation. Information on this parameter is required for the quadcopter vehicle so that control system can act properly. It is essential to note that the rotorcraft is assumed to be symmetric consummately about each the x, y, and z axes and to have its centre of gravity at the geometric center of the arms. With these assumptions, the J matrix as presented in equation (3.10) will be a diagonal. The I_{xx} and I_{yy} terms are considered as identical because of this symmetry.

There are two well known approaches to compute the moment of inertia:

1. Calculation method. Some basic moment of inertia equations are utilized to obtain the parameter.
2. Bifilar pendulum approach. This is an experimental approach for estimating rotational moments of inertia.

A. Calculation method

Some dimensional units of quadcopter are required to calculate the moment of inertia of the vehicle, including length (l) and mass (m). Calculation of the parameter can be conducted by the following steps (Hartman, 2014):

- Break quadcopter into separate parts, and model each part as a simplified geometric shape of constant internal density.
- Measure length and weigh each part.
- Use parallel-axis theorem to calculate the mass moment of inertia contribution of each component about the x, y, and z axes of the vehicle.

- Sum up the calculated inertias for every component of each axis to find the total mass moment of inertia values for the vehicle.

Parallel-axis theorem is utilized to calculate the mass moment of inertia of each component for every chosen axis. It is calculated through the centre of gravity of each component and the distance of perpendicular between the two axes. The parallel-axis equation can be presented as,

$$I_{parallel-axis} = I_{COM} + mr^2 \tag{3.60}$$

where I_{COM}, m, and r represent the moment of inertia of the vehicle part about its own axis parallel to the axis where the vehicle will move, the mass of individual component, and the distance of perpendicular between the parallel axes respectively.

a. Motors: solid cylinders ($I_{xx,M}$, $I_{yy,M}$, and $I_{zz,M}$)

Motors used in the quadcopter will affect the moment of inertia of the rotor-craft itself. By modelling motor as a solid cylinder, some variables should be measured to obtain the moment of inertia, as shown in Figure 3.3 (Hartman, 2014),

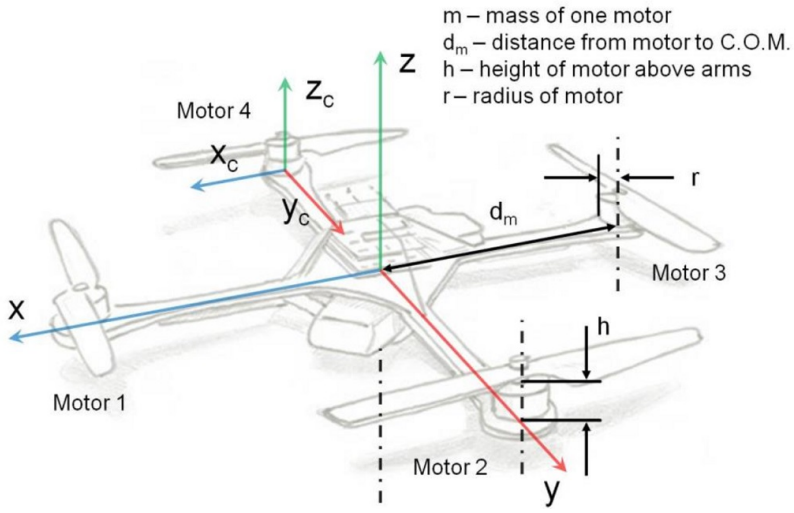

Figure 3.3. Motor - solid cylinder ($I_{xx,M}$, $I_{yy,M}$ and $I_{zz,M}$)

From Figure 3.3, each motor can be modelled as cylinder rotating around an end diameter (x and y axes) and cylinder rotating around a central axis (z axis) as depicted in Figures 3.4 and 3.5.

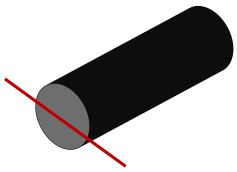

Figure 3.4. x and y axes

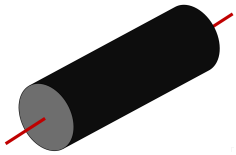

Figure 3.5. z axis

Therefore, the I_{COM} for x and y axes as well as for z axis can be obtained as,

$$I_{xx,M} = I_{yy,M} = I_{COM} = \frac{1}{4}mr^2 + \frac{1}{3}mh^2 \tag{3.61}$$

$$I_{zz,M} = I_{COM} = \frac{1}{2}mr^2 \tag{3.62}$$

$I_{xx,M}$ and $I_{yy,M}$ can be obtained by utilizing a cylinder turning around an end diameter as seen in Figure 3.4. For quadcopter systems, the moment of inertia in x and y axes caused by motors utilized in the vehicle can be expressed as,

$$I_{xx,M} = I_{yy,M} = I_{COM} = 2\left[\frac{1}{4}mr^2 + \frac{1}{3}mh^2\right] + 2\left[\frac{1}{4}mr^2 + \frac{1}{3}mh^2 + md_m^2\right] \tag{3.63}$$

Note in the above equation that the first bracketed term is for motors 1 and 3, which turn around an end diameter to coincide with the x-axis of the rotor-craft, thus the distance term (mr^2– theorem of parallel axis) is zero. The second bracketed term is for motors 2 and 4, which are at a diameter that is offset but parallel to the x-axis of the quadcopter. The distance term at the end is considered (mr^2) but the variable used is d_m instead of r for clarity. The variable d_m is the distance of perpendicular between the axis of rotation of the motor and the actual x-axis of the quadcopter. The quadcopter is assumed symmetric so that $I_{xx,M}$ has the same value as $I_{yy,M}$.

Furthermore, $I_{zz,M}$ can be obtained by utilizing a cylinder turning around a central axis (see Figure 3.5). In this situation, 4 motors are turning around a central axis that is offset but parallel to the z-axis of the quadcopter. The distance term md_m^2 is considered. This term is the distance of perpendicular between the turning axis of the motor and the actual z-axis of the quadcopter. All motors have the same distance term d_m, thus the moment of inertia in z-axis is defined as,

$$I_{zz,M} = I_{COM} = 4\left[\frac{1}{2}mr^2 + md_m^2\right] \tag{3.64}$$

b. Central HUB: solid cylinder ***($I_{xx,H}$, $I_{yy,H}$, $I_{zz,H}$)***

Another component that affects the moment of inertia of the quadcopter is the central hub of the vehicle. The central hub shape can be approximated with a solid cylinder as seen in Figure 3.6 (Hartman, 2014).

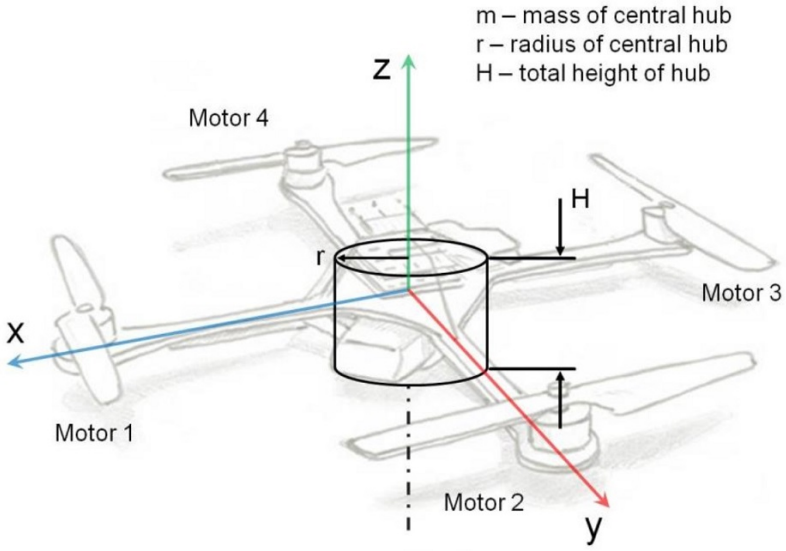

Figure 3.6. Central HUB: solid cylinder

The moment of inertia in x, y, and x - axes can be calculated by considering a solid cylinder turning around a central diameter for x and y axes, and a solid cylinder turning around a central axis for z axis, as depicted in Figures 3.7 and 3.8.

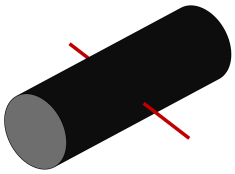

Figure 3.7. Central HUB x and y axes

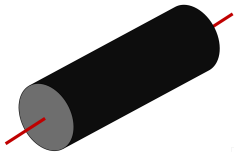

Figure 3.8. z axis

Considering the distance term (mr^2– parallel axis theorem) is zero, the calculation of inertia moment $I_{xx,H}$ or $I_{yy,H}$ for central hub can be expressed as follows:

$$I_{xx,H} = I_{yy,H} = I_{COM} = \left[\frac{1}{4}mr^2 + \frac{1}{12}mH^2\right] \tag{3.65}$$

The vehicle is assumed as symmetric, so $I_{yy,H}$ is the same as $I_{xx,H}$. Furthermore, $I_{zz,H}$ can be calculated by considering a cylinder turning around a central axis. In this condition

the hub is turning around a central axis that coincides with the z-axis of the quadcopter, so the distance term for this axis is considered zero. The moment of inertia for this axis is defined as,

$$I_{zz,H} = I_{COM} = \left[\frac{1}{2}mr^2\right] \tag{3.66}$$

c. Arms: long cylindrical rods ***($I_{xx,A}$, $I_{yy,A}$, $I_{zz,A}$)***

The length of the arm of the quadcopter will greatly determine the magnitude of the moment of inertia of the quadcopter, so calculating the parameter for this component is required. Some variables need to be measured to calculate this parameter as shown in Figure 3.9 (Hartman, 2014),

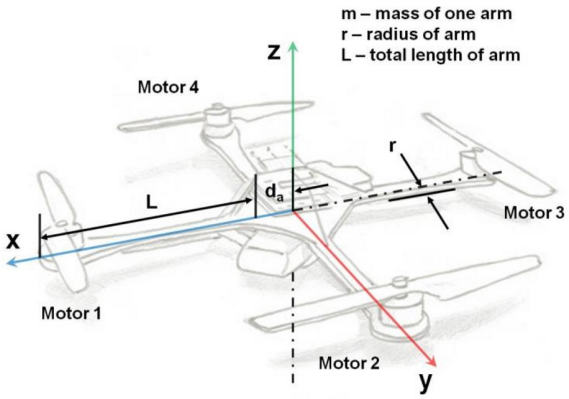

Figure 3.9. Arms: long cylindrical rods

The arm is modelled as a solid cylinder rotating around a central axes for x and y axes, and turning around an end diameter for x, y and z axes as can be seen in Figures 3.10 and 3.11.

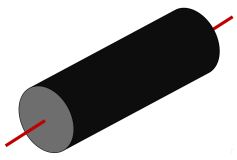

Figure 3.10. x and y axes

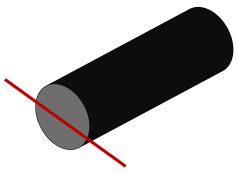

Figure 3.11. x, y and z axes

The basic calculation of I_{COM} for x and y axes as well as for x, y and z axes can be expressed as,

$$I_{COM-xyaxes} = \frac{1}{2}mr^2 \tag{3.67}$$

$$I_{COM-xyz-axes} = I_{COM} = \frac{1}{4}mr^2 + \frac{1}{3}mL^2 \tag{3.68}$$

The moments of inertia $I_{x,A}$ and $I_{y,A}$ can be calculated by considering equations (3.67) and (3.68). Therefore, this parameter is defined as,

$$I_{xx,A} = I_{yy,A} = 2\left[\frac{1}{2}mr^2\right] + 2\left[\frac{1}{4}mr^2 + \frac{1}{3}mL^2 + md_A^2\right] \tag{3.69}$$

It can be seen in equation (3.69) that the first bracketed term is for arms 1 and 3, which turn around a central axis to coincide with the x-axis of the quadcopter, thus the distance term (mr^2– theorem of parallel axis) is zero. The second bracketed term in the equation is for arms 2 and 4 which rotate around an end diameter located at a distance "d_a" from the x-axis of the vehicle, so the theorem of parallel axis term applied is md_a^2. The vehicle is assumed symmetric, so $I_{yy,A}$ has the same value as $I_{xx,A}$. Furthermore, $I_{zz,A}$ can be calculated by considering a cylinder rotating around an end diameter as seen in Figure 3.11. In this condition, 4 arms are turning around an end diameter located at a distance "d_a" from the z-axis of the vehicle, so the theorem of parallel axis term is md_a^2. Because of symmetrical shape and each arm having the same length, the moment of inertia for this axis is defined as,

$$I_{zz,A} = 4\left[\frac{1}{4}mr^2 + \frac{1}{3}mL^2 + md_A^2\right] \tag{3.70}$$

d. Calculation method results

Referring to equations (3.63), (3.64), (3.65), (3.66), (3.69), and (3.70), the total moment of inertia for the quadcopter UAV is obtained as shown in Table 3.1,

Table 3.1. Calculation of moment of inertia of quadcopter UAV

Inertia motor		**Units**	**Long Cylindrical rods**		**Units**	**Inertia HUB**		**Units**	**Total inertia**		**Units**
m	0.111	kg	m	0.038	kg	m	1.127	kg	I_{xx}	0.023219654	kgm^2
r	0.014	m	r	0.007	m	r	0.079	m	I_{yy}	0.023219654	kgm^2
h	0.035	m	L	0.229	m	H	0.125	m	I_{zz}	0.043138088	kgm^2
d_m	0.285	m	d_A	0.075	m						
I_{xx}=I_{yy}	0.018235006	kgm^2	I_{xx}=I_{yy}	0.001758798	kgm^2	I_{xx}=I_{yy}	0.00322585	kgm^2			
I_{zz}	0.036107412	kgm^2	I_{zz}	0.003513873	kgm^2	I_{zz}	0.003516804	kgm^2			

B. Bifilar pendulum approach

A bifilar pendulum as seen in Figure 3.12 (Jardin and Mueller, 2009) is a torsional pendulum comprising an object hanged by two thin parallel wires whose moment of inertia is measured. The object is rotated to oscillate around the vertical axis. The reinstating torque of the bifilar pendulum is given by the gravitational force as turns from the initial state causes the object to rise slightly. The mass moment of inertia is calculated utilizing dynamic modelling, the oscillation period measurements, and the information of physical dimensions of the pendulum including length of wires and the distance between pendulum wires.

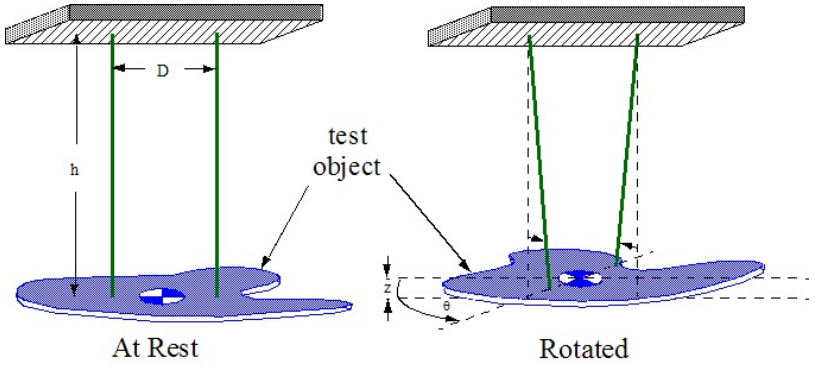

Figure 3.12. Bifilar pendulum diagram

Therefore, the moment of inertia is calculated by utilizing the following equation,

$$I = \frac{mgD^2}{4h\omega^2} \tag{3.71}$$

where m, g, and D denote the mass of the object, the gravity acceleration, and the distance between the wires respectively, while h is the length of the wire, and ω is angular frequency. Equation (3.71) is obtained from mathematical model of a bifilar pendulum, which is derived from Lagrange equations as shown in the following equation,

$$0 = I\ddot{\theta} + K_d\dot{\theta}^2 + C\dot{\theta} + \frac{mgD^2 \sin\theta\cos\theta}{4h\sqrt{1 - \left(\frac{D\sin\theta}{2h}\right)^2}} \tag{3.72}$$

The final form of the equation is obtained by assuming that the pendulum has small deflections, damping (viscous (C) and aerodynamic (K_d) are zero.

a. Equipments and method

Materials and equipments required for obtaining the moment of inertia of quadcopter are as follows:

- quadcopter
- bifilar pendulum test bench
- ropes
- stop watch
- IMU sensor
- digital scale
- measuring tape

After preparation, the following steps need to be followed:

Step 1: Both sides of quadcopter arms are tied with rope

Step 2: The quadcopter is hanged on a stable rack. For acquiring moment of inertia z-axis (I_{zz}), the position of quadcopter can be seen as in Figure 3.13, meanwhile for I_{yy} and I_{xx} the setups are as in Figures 3.14 and 3.15 respectively.

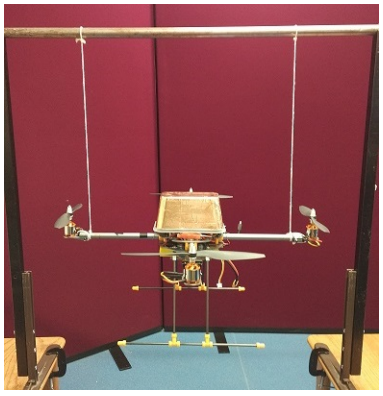

Figure 3.13. Bifilar - z-axis

Figure 3.14. Bifilar - y-axis

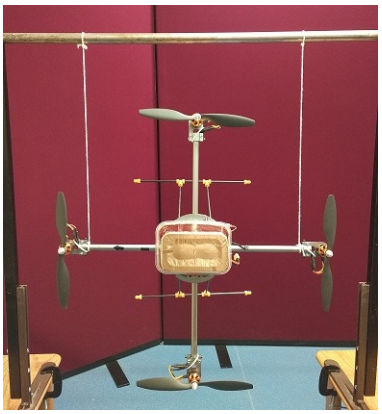

Figure 3.15. Bifilar - x-axis

Step 3: The quadcopter is rotated along z-axis by hand and released free. The same is done for y-axis and x-axis for obtaining I_{yy} and I_{xx} respectively.

Step 4: The time required by the quadcopter to swing about the axis is recorded using stop watch.

Step 5: The data of angular rotation of the quadcopter is acquired from IMU sensor.

Step 6: Step 1 to step 5 are repeated more than 10 times to obtain more accurate measurement.

Step 7: The mass (m) of the quadcopter is obtained by placing the quadcopter on a digital weight scale.

Step 8: The length, D from motor to motor and h (length of rope) are measured by using a measuring tape

The following precautions need to be considered to obtain valid data:

- The two ropes are in the same length and in vertical positions.
- The ropes are hanged parallel to each other vertically.
- Only the oscillations around a vertical axis through the center of the rack rod are considered.

b. Measurement results

The parameters used for calculating the moment of inertia through bifilar approach are the following,

Start angle	46	deg
Length of rops (h)	0.545	m
Length between ropes (D)	0.534	m
Quadcopter mass (m)	1.800	kg
Gravity (g)	9.812	$\frac{m}{s^2}$
Number of period (T)	300	times

From the experiment, the period of oscillation is obtained as seen in Tables 3.2, 3.3, and 3.4. Therefore, using equation (3.71), the moment of inertia in each axis can be obtained as,

- I_{xx} = 0.026282 kgm^2
- I_{yy} = 0.027484 kgm^2
- I_{zz} = 0.045603 kgm^2

Table 3.2. Bifilar experiment for I_{xx}

Number of experiment	***t*(300)**	***t*(1 period)**	**Unit**
Experiment 1	201.120	0.6704	second
Experiment 2	200.970	0.6699	second
Experiment 3	201.100	0.670333	second
Experiment 4	201.110	0.670367	second
Experiment 5	201.080	0.670267	second
Experiment 6	200.720	0.669067	second
Experiment 7	201.140	0.670467	second
Experiment 8	200.940	0.6698	second
Experiment 9	201.020	0.670067	second
Experiment 10	201.220	0.670733	second
Average	**201.042**	**0.67014**	**second**

Table 3.3. Bifilar experiment for I_{yy}

Number of experiment	***t*(300)**	***t*(1 period)**	**Unit**
Experiment 1	205.180	0.683933	second
Experiment 2	205.540	0.685133	second
Experiment 3	205.770	0.6859	second
Experiment 4	205.150	0.683833	second
Experiment 5	205.060	0.683533	second
Experiment 6	205.620	0.6854	second
Experiment 7	204.890	0.682967	second
Experiment 8	206.310	0.6877	second
Experiment 9	206.290	0.687633	second
Experiment 10	206.100	0.687	second
Average	**205.591**	**0.685303**	**second**

Table 3.4. Bifilar experiment for I_{zz}

Number of experiment	*t*(300)	*t*(1 period)	Unit
Experiment 1	264.350	0.881167	second
Experiment 2	264.750	0.8825	second
Experiment 3	264.920	0.883067	second
Experiment 4	264.830	0.882767	second
Experiment 5	265.070	0.883567	second
Experiment 6	264.980	0.883267	second
Experiment 7	264.960	0.8832	second
Experiment 8	264.960	0.8832	second
Experiment 9	264.590	0.881967	second
Experiment 10	264.830	0.882767	second
Average	**264.824**	**0.882747**	**secon**

C. Summary

In the calculation method, moment of inertia for x and y axes have the same value, it is assumed that each axis is symmetric. With the bifilar method, the moments of inertia for x and y are slightly different. This is due to the fact that the load of each axis is slightly different because of the placement and installation of some electronic components. Furthermore, the value of moment of inertia obtained by calculation method and bifilar method do not have big different as seen in Table 3.5. This shows that both methods can be used to identify the parameters of quadcopter UAVs. However, the differences in outcomes of the two methods can be attributed to several things, including: the asymmetry of the system, air disturbance during experiment, inaccurate load measurement, and the simplification of modelling for each part.

Table 3.5. Mass moment of inertia comparison

Moment of inertia	Calculation	Bifilar (exp)	Units
I_{xx}	0.023219654	0.026281674	kgm^2
I_{yy}	0.023219654	0.027484487	kgm^2
I_{zz}	0.043138088	0.045603074	kgm^2

3.6.2 Motor and propeller identification method

Motor and propeller data play a vital role in designing real application of rotorcraft based unmanned aerial vehicles (UAVs), inculding quadcopter. The size of quadcopter and the lifetime of battery should be taken into account to choose proper motor and propeller so that the vehicle can be hovering and manoeuvring according to the desired length of time. Therefore, some data regarding motor and propeller are required including: torque, thrust,

voltage and current consumption, motor speed, motor winding resistance, vibration, motor efficiency, and propeller efficiency.

A. General description of system identification device

In this research, the thrust stand RC Benchmark 1580 dynamo-meter series is used as seen in Figure 3.16. The device is equipped with load cell for torque and thrust measurement and some sensors to measure voltage and current consumption, rotation speed, and vibration. Basically, this device works by giving pulse width modulation (PWM) signal to the motor. The magnitude of the thrust force and torque generated by the propeller rotation is measured by the load cell.

Figure 3.16. Thrust stand - RC Benchmark 1580 dynamometer series

The technical specifications of this device are given below.

a *Direct measurements*

- Voltage (0 - 35 V).
- Current (40A continuous, 50A burst).
- Power (0 - 1400W).
- Thrust ($\pm$ 5kg).
- Motor speed (100k RPM).
- Torque ($\pm$ 1.5Nm).
- Precision ohmmeter (0.003 - 240 ohm).
- Accelerometer.

b *Derived measurements*

- Motor efficiency (%).
- Propeller efficiency (g/W).
- Vibration level.

Moreover, this device comes with software to log, process data and save to file as seen in Figure 3.17. Before carrying out the measurement, the device needs to be calibrated to provide accurate and valid data.

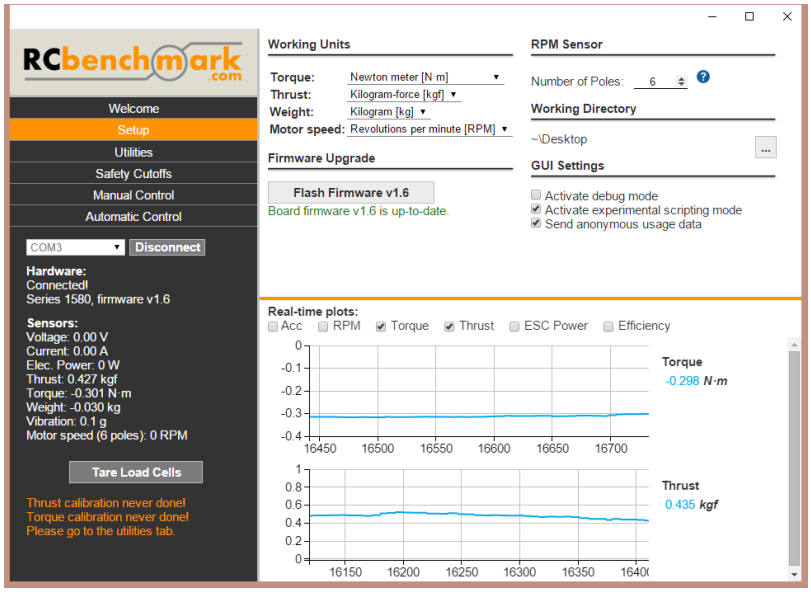

Figure 3.17. RC Benchmark 1580 dynamometer software

B. Motor and propeller data

The motor's type used for driving quadcopter is Turnigy D2836/8 1100KV and 10x4.5" Gemfan propeller. From the experiments, the data shown in Table 3.6 was obtained. Furthermore, by using polynomial method, the relationship model between PWM signals and thrust force (F) was obtained as in Figure 3.18.

Table 3.6. Motor and propeller data

ESC signal (μs)	%PWM	Thrust (N)	Torque (N·m)	Voltage (V)	Current (A)
1092	0	-0.0142647	0.000214395	12.49377882	0
1098	0.8174387	0.29095887	0.004937631	12.48831551	0.225908887
1109	2.3160763	0.37843288	0.007625374	12.48331675	0.305709007
1115	3.133515	0.45609002	0.009359799	12.48076873	0.358697979
1120	3.8147139	0.57467876	0.011102604	12.4768578	0.416671613
1130	5.1771117	0.61788551	0.012362755	12.47379689	0.525361991

Table 3.6. Motor and propeller data

ESC signal (μs)	%PWM	Thrust (N)	Torque (N·m)	Voltage (V)	Current (A)
1140	6.5395095	0.76725005	0.013375995	12.46825636	0.636304976
1150	7.9019074	0.77226473	0.017381981	12.46236774	0.766263702
1160	9.2643052	0.99283158	0.018659937	12.45567464	0.885954971
1170	10.626703	1.19419241	0.021329445	12.44817948	1.02790633
1181	12.125341	1.32974945	0.023891625	12.44004359	1.177456998
1198	14.441417	1.51651521	0.027562603	12.43017432	1.469058648
1218	17.166213	1.76800934	0.031249916	12.41873027	1.791478876
1238	19.891008	2.025406	0.034407294	12.40617029	2.12465502
1253	21.934605	2.21187749	0.037950782	12.39463496	2.413710717
1269	24.114441	2.43889065	0.040182812	12.38184955	2.707964303
1284	26.158038	2.61733014	0.04382624	12.36780677	2.9840259
1303	28.746594	2.87508844	0.048687509	12.35122138	3.410854454
1323	31.47139	3.21358662	0.05203605	12.33262536	3.841285867
1345	34.468665	3.56138463	0.058078182	12.31035586	4.385705902
1366	37.3297	3.86794431	0.063490801	12.2877107	4.897864478
1383	39.645777	4.0904605	0.067629357	12.26799633	5.293601127
1398	41.689373	4.27918757	0.070869518	12.24898481	5.619876377
1426	45.504087	4.71895993	0.07850464	12.22098115	6.361853135
1450	48.773842	5.06983258	0.083677019	12.19691931	6.961993235
1471	51.634877	5.36098294	0.087532945	12.1712413	7.473961651
1495	54.904632	5.79515716	0.094534335	12.13891057	8.233612619
1515	57.629428	6.15101558	0.099652946	12.10898025	8.915261798
1533	60.081744	6.5778319	0.10570248	12.07637831	9.63786147
1551	62.53406	6.86255702	0.111345828	12.04487024	10.29804476
1576	65.940054	7.44064893	0.120266921	11.99666365	11.42591244
1600	69.209809	7.86278163	0.129001457	11.94869997	12.47708836
1623	72.343324	8.17635113	0.13513675	11.90252266	13.41673254
1618	71.662125	8.01288302	0.132836708	11.89391174	13.08611197
1638	74.386921	8.36555274	0.139558474	11.85275821	13.94829952
1655	76.702997	8.59289557	0.143324039	11.8177176	14.56868483
1677	79.700272	9.04377407	0.15138839	11.76965078	15.64140416
1696	82.288828	9.59746617	0.159552683	11.71369839	16.83209676
1715	84.877384	9.94869775	0.165675978	11.66831213	17.7931778
1734	87.46594	10.2948064	0.171936631	11.61988983	18.77476756
1751	89.782016	10.527757	0.176316052	11.56943069	19.63790197
1771	92.506812	10.8609479	0.182111963	11.51034639	20.7209396
1787	94.686649	11.0046021	0.185931927	11.46482772	21.50060283
1810	97.820163	11.4019938	0.19323085	11.40184527	22.94106767
1826	100	11.5823938	0.19609444	11.36011844	23.53869824

The x and y axes in Figure 3.18 represent thrust force (F) and pulse width modulation (PWM) signals. Therefore, the polynomial equation representing the relationship between PWM signals and thrust force is given as,

$$PWM_i = 0.0047F_i^4 - 0.0899F_i^3 + 0.2919F_i^2 + 10.025F_i - 1.1474 \tag{3.73}$$

where PWM_i $(i = 1 - 4)$ is pulse width modulation signals for motor to generate thrust

force F_i $(i = 1 - 4)$ from each motor or propeller.

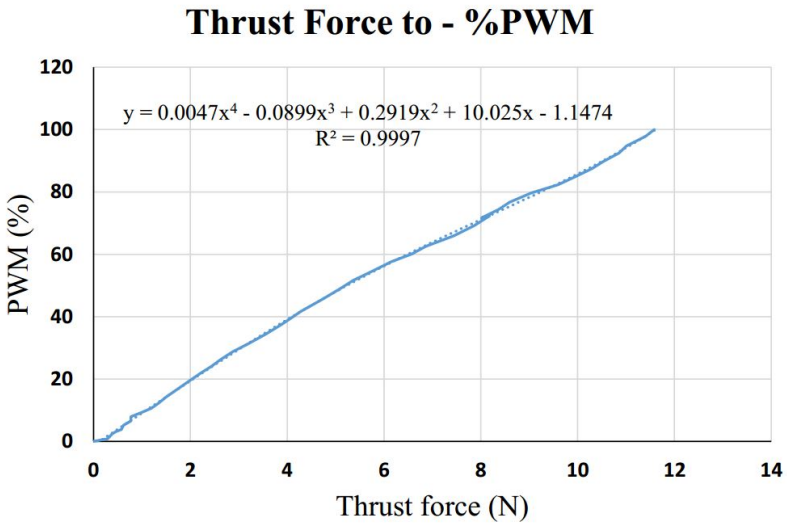

Figure 3.18. Thrust force to PWM signals

3.7 Summary

Mathematical model of systems play a critical role in applying model based-control systems, including sliding mode control (SMC). As a result, system parameters must be available for the control system to work properly and accurately. Therefore, system identification approaches play an important role in identifying and measuring system parameters. In this thesis, the kinematic and dynamic models of quadcopter were derived based on a Newton-Euler formalism because it is computationally efficient. Meanwhile, in order to obtain the moment of inertia of quadcopter, calculation method and bifilar approach were used. Both methods yielded relatively similar values. Furthermore, in a real time system, thrust force and torque information of the motor and propeller used is required. These informations are obtained by giving PWM signal to motor and measuring the lifting power generated by the motor. The correlation between PWM signals and thrust force was obtained by plotting the data acquired.

Chapter 4

Design of sliding mode control for quadcopter UAVs

4.1 Introduction

This chapter presents the design of sliding mode control formalism so as to deal with nonlinearity, underactuation and coupled states noted in quadcopter UAVs. The controller is further able to handle uncertainty and imprecision in the model which may appear in the system.

Model imprecision of a nonlinear system may originate from deliberate decision to simplify system dynamics, or from real uncertainty about the system (e.g., unidentified system parameters). Modelling imprecision is characterized into two main types, namely: structured and unstructured uncertainties. The first type correlates with "inaccuracies on the terms actually" incorporated into the model, and the second type relates to "inaccuracies on the system order" (Slotine and Li, 1988). Model imprecision will give adverse influence on overall control system performance. This issue may be tackled by robust control approaches which can deal with nonlinearity, uncertainty, and disturbance.

Sliding mode control (SMC) is a prominent robust control method which offers an effective way to deal with the issue of stability and performance consistency notwithstanding nonlinearity and model imperfection. SMC is a nonlinear control approach that adjusts a nonlinear system dynamics by utilizing a switching control input that compels the system to "slide" on the sliding surface all the time. The control law is formed with a discontinuous function of time. In fact, it can change from one continuous form to another based on

the current position in the state space. Hence, SMC is a variable structure control method. With regards to modern control theory, any variable structure system, including SMC, might be seen as a special case of a hybrid dynamical system as the system both moves through a continuous state space yet also flows through various discrete control modes.

4.2 Sliding surface

This section examines variable structure control (VSC) as a high frequency switched feedback control in sliding mode. The switching control law has a role to bring the trajectory of nonlinear state onto a pre-defined surface in state space representation and to keep the trajectory of the system's state on this surface over time. Commonly, this surface is called sliding or switching surface, as the feedback path will have different gains when the state trajectory is over or under the surface. A proper switching behaviour is directed by the rule defined by the surface. A Lyapunov approach is utilized to describe this task (DeCarlo et al., 1988).

Lyapunov stability criterion is commonly used to determine the stability of dynamical systems for which the solution is close enough or converge to an equilibrium point without solving the state equation. Consider a system $\dot{x} = f(x)$ which has an equilibrium point at $x = 0$, and a function $V(x) : \mathbb{R}^n \longrightarrow \mathbb{R}$ with the end goal that

- $V(x) = 0$ if and only if $x = 0$
- $V(x) > 0$ if and only if $x \neq 0$
- $\dot{V}(x) = \frac{d}{dt}V(x) = \sum_{i=0}^{n} \frac{\partial V}{\partial x_i} f_i(x) \leq 0$ for all values of $x \neq 0$

If the system meets these requirements, then the stability is definitely assured (note that $V(0) = 0$ is required). It means that the trajectory of the state is guaranteed to be kept on the surface for subsequent times. Furthermore, the $V(x)$ is called a Lyapunov function candidate.

4.2.1 Basic sliding surface design

A nonlinear system can in general be defined as,

$$\dot{x}_n = f(\mathbf{x}, t) + b(\mathbf{x}, t)u(t) \tag{4.1}$$

where $\mathbf{x}(t) \in \mathbb{R}^n$ and $u(t) \in \mathbb{R}^m$ represent n-dimensional state vector and m-dimensional control input vector that will be utilized for state feedback respectively. The subscript n on $\dot{x}$ represents the number of states. $f(\mathbf{x}, t) \in \mathbb{R}^n$ and $b(\mathbf{x}, t) \in \mathbb{R}^{n \times m}$ represent general nonlinear functions and gain of control input respectively. In addition, those functions are assumed to be continuous and adequately smooth. The functions are not accurately known, but the degree of inaccuracy on those function is enclosed by a known continuous derivative function with respect to **x**. The role of the control input is to convey the state **x** to follow a desired state $\mathbf{x}_d$ despite of model imperfection on $f(\mathbf{x}, t)$ and $b(\mathbf{x}, t)$. A time varying switching surface $s(t)$ is characterized in the system's dynamic by equalizing the variable $s(\mathbf{x}; t)$ to zero as,

$$s(\mathbf{x}; t) = \left(\frac{d}{dt} + \lambda\right)^{n-1} \mathbf{e}(t) \tag{4.2}$$

where $\lambda > 0$ (must be Hurwitz) and $\mathbf{e}(t)$ = $\mathbf{x}(t)$ - $\mathbf{x}_d(t)$ represent the bandwidth of the system and the output state error respectively. The theorem of first-order stabilization of s can be used to simplify the problem of tracking n-dimensional vector $\mathbf{x}_d(t)$. In addition, bounds on switching surface s can be simply converted into bounds on the vector of tracking error **x** , and thus the scalar switching surface s denotes an actual value of tracking performance. By assuming $e(0)$ = 0, the relating transformations of performance measures is defined as (Slotine and Li, 1988),

$$\forall t \geq 0, |s(t)| \leq \phi \Rightarrow \forall t \geq 0, |e^i(t)| \leq (2\lambda)^i \varepsilon \tag{4.3}$$

where $\varepsilon = \dfrac{\phi}{\lambda^{n-1}}$, and n is system's order. The problem of tracking in n^{th}- order can be simplified by utilizing a theorem of 1^{st}- order stabilization. The simplified tracking problem for maintaining the value of s at zero can be accomplished by picking the control

law u in equation (4.1) with the end goal that is outside of $s(t)$.

$$\frac{1}{2}\frac{d}{dt}s^2 \leq -\eta|s| \tag{4.4}$$

where $\eta > 0$. Basically, equation (4.4) describes that the squared "gap" to the switching surface, as quantified by s^2, diminishes along all trajectories of system. In this way, it restricts trajectories to direct towards the sliding surface $s(t)$. Especially, when it has been on the surface as shown in Figure 4.1, the trajectories stay on the sliding surface for subsequent times. In other words, fulfilling the sliding condition lead to stable invariant set of the system that will keep the system robust to uncertainty and disturbance.

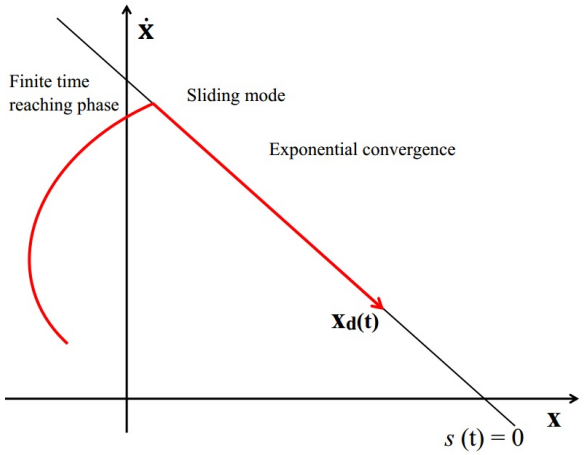

Figure 4.1. Sliding manifold

Finally, fulfilling the condition in equation (4.4) ensures that if in fact $\mathbf{x}(t=0)$ is off $\mathbf{x}_d(t=0)$, the surface $s(t)$ will be achieved in a finite time by less than $\frac{|s(t=0)|}{\eta}$. For example, if $s(t=0) > 0$, and let t_{reach} be the time it takes to reach the surface $s = 0$. By integrating equation (4.4) from $t = 0$ to t_{reach} will prompt

$$0 - s(t=0) = s(t=t_{\text{reach}}) - s(t=0) < -\eta(t_{\text{reach}} - 0) \tag{4.5}$$

which

$$t_{\text{reach}} \leq \frac{s(t=0)}{\eta} \tag{4.6}$$

The same outcome will be acquired as condition (4.6) for $s(t=0) > 0$. Beginning from any starting point, the time-varying sliding surface will be achieved by state trajectory in a finite time by less than $\frac{s(t=0)}{\eta}$, and after that slides on the surface towards desired state

$x_d(t)$ exponentially over times.

In summary, by utilizing a good function of switching surface $s(t)$ as presented in equation (4.2) and choosing the control law u as expressed in equation (4.1) such that s^2 stays as a closed-loop system in spite of the existence of inaccurate model and disturbances, is the idea of developing the control method.

4.2.2 Integral sliding surface design

To reduce the steady-state error, an integral term of tracking error is introduced into equation (4.2), which makes up the traditional integral sliding manifold as

$$s(\mathbf{x};t) = \left(\frac{d}{dt} + \lambda\right)^{n-1} \mathbf{e}(t) + \lambda_i \int_0^{\infty} \mathbf{e}(t)dt \tag{4.7}$$

where λ_i is a strictly positive constant.

4.2.3 Set-point weighting

Employing integral action in sliding mode control is believed to improve the controller performance by reducing steady-state error. However, an overshoot in output states can be produced by the following issues (Mantz et al., 1999):

- $|\mathbf{e}(t)| > |\mathbf{e}_{ss}|$, where $\mathbf{e}_{ss}$ is steady-state error of $\mathbf{e}(t)$,
- $sign(\mathbf{e}(t)) \neq sign(\mathbf{e}_{ss})$

Furthermore, considering an issue following typical feedback controller design, such as PID, is how to achieve a high performance controller to track the reference fast and to reject disturbance at the same time. Generally, high-gain control parameters are required to deal with load disturbance. However at the same time, this will lead to oscillation or high overshoot of step response. This problem can be solved by introducing "*set-point weighting* to increase the frequency of Zero, and thus reduce the overshoot in the output following step changes in the set-point" (Visioli, 2006).

This approach is adopted in this research to handle the problem of utilizing integral term in sliding mode control to reduce overshoot in output states while maintaining ro-

bustness of the controller. Therefore, the new sliding mode approach by applying set-point weighting can be designed as shown in Figure 4.2.

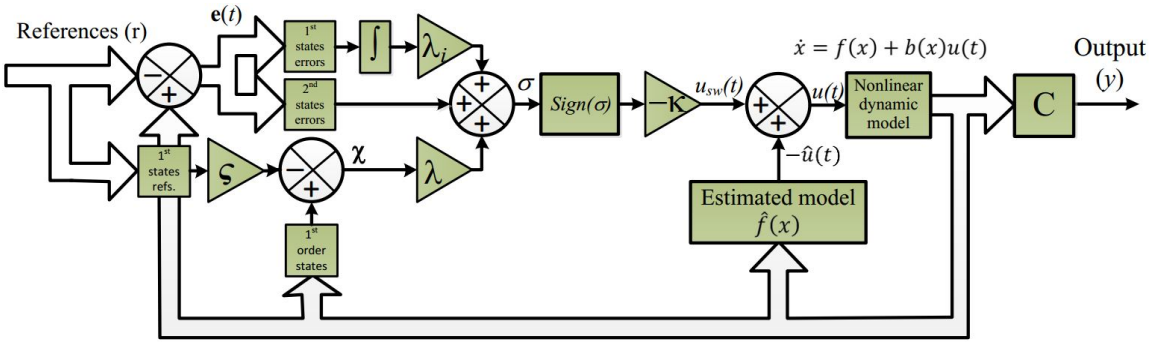

Figure 4.2. Set-point weighting function

Accordingly, a new error $\chi(t)$ for proportional action is presented as,

$$\chi(t) = e(t) + r(1 - \varsigma) \tag{4.8}$$

where r is reference and ς is constant between 0 and 1. Reducing ς will lead to decreasing overshoot and increasing rise time, and vice versa. Consequently, the new sliding surface can be defined as

$$s(\mathbf{x}; t) = \left(\frac{d}{dt} + \lambda\right)^{n-1} \chi(t) + \lambda_i \int_0^{\infty} \mathbf{e}(t)dt \tag{4.9}$$

4.3 Control law

In order to account for uncertainties and disturbances, the control law should be discontinuous across $s(t)$. However, this will result in chattering as seen in Figure 4.3 due to the imperfect implementation of such control law. This phenomenon is not expected in real-time application, because it let in high control action and can generate high-frequency dynamics that are ignored in process of modelling.

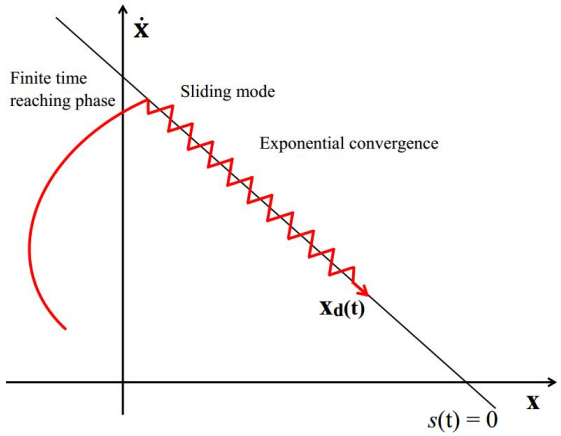

Figure 4.3. Chattering phenomenon

Consider a general form of 2^{nd}-order system as,

$$\ddot{x} = f(x,t) + u(t) \tag{4.10}$$

where $f(x,t)$ is time varying nonlinear system in general and approximated as $\hat{f}(x,t)$, while $x(t)$ and $u(t)$ are, respectively, the state variable to be controlled and control input so that it tracks a reference trajectory $x_d(t)$. The error of estimation on $f(x,t)$ is assumed to be restricted by some given function F=$F(x,t)$, so that

$$|\hat{f}(x,t) - f(x,t)| \leq F(x,t) \tag{4.11}$$

By utilizing equation (4.9), a 2^{nd}-order sliding surface is expressed as,

$$s(t) = (\frac{d}{dt} + \lambda)\chi(t) + \lambda_i \int_0^\infty e(t)dt = \dot{e}(t) + \lambda\chi(t) + \lambda_i \int_0^\infty e(t)dt \tag{4.12}$$

The first derivative of sliding surface is presented as,

$$\dot{s}(t) = \ddot{e}(t) + \lambda\dot{e}(t) + \lambda_i e(t) \tag{4.13}$$

where $\ddot{e}$ = $\ddot{x}$ - $\ddot{x}_d$, so equation (4.13) can be rewritten as,

$$\dot{s}(t) = \ddot{x}(t) - \ddot{x}_d(t) + \lambda\dot{e}(t) + \lambda_i e(t) \tag{4.14}$$

By substituting for $\ddot{x}$ from equation into equation (4.10) and (4.14) and simplifying yield,

$$\dot{s}(t) = f(x,t) + u(t) - \ddot{x}_d(t) + \lambda\dot{e}(t) + \lambda_i e(t) \tag{4.15}$$

In order to achieve $\dot{s}(t) = 0$, then the approximation of control law ($\hat{u}$) is defined as,

$$\hat{u}(t) = -\hat{f}(x,t) + \ddot{x}_d(t) - \lambda\dot{e}(t) - \lambda_i e(t) \tag{4.16}$$

where $\hat{u}(t)$ can be defined as equivalent control generated from the best approximated model. Furthermore, to account for uncertainty in nonlinear systems while fulfilling the sliding condition as stated in equation (4.4), the overall control law is defined as,

$$\begin{aligned} &u(t) = \hat{u}(t) - u_{sw}, \\ &\text{where } u_{sw} \text{ is switching control} \\ &u_{sw} = k(x,t)\text{sign}(s) \end{aligned} \tag{4.17}$$

by selecting an appropriate gain $k(x,t)$, such as $k(x,t) = F(x,t) + \eta$ ensures the fulfilment of condition in equation (4.4), since

$$\begin{aligned} &\frac{1}{2}\frac{d}{dt}\big(s(t)^2\big) = \dot{s}(t)s(t) = \big(f(x,t) - \hat{f}(x,t)\big)s(t) - k(x,t)|s(t)| \leq \eta|s(t)| \\ &\eta > 0 \end{aligned} \tag{4.18}$$

Therefore, by employing equation (4.17), the system trajectory will hit the sliding surface $s(t)$ in finite time, after which the state errors will converge to zero exponentially.

Now consider the 2^{nd}-order system by introducing control input gain $b(x,t)$ as follows,

$$\ddot{x} = f(x,t) + b(x,t)u(t) \tag{4.19}$$

where $b(x,t)$ is bounded as,

$$0 \leq b_{min}(x,t) \leq b(x,t) \leq b_{max}(x,t) \tag{4.20}$$

Function $b(x,t)$ and its boundary can be state dependent or time varying. Because the control input $u(t)$ is gained by $b(x,t)$ in the system dynamic, the lower and upper bound of $b(x,t)$ is defined as,

$$\hat{b}(x,t) = \sqrt{b_{min}(x,t)b_{max}(x,t)} \tag{4.21}$$

Therefore, the boundaries can then be defined as,

$$\begin{aligned} &\beta^{-1} \leq \frac{\hat{b}(x,t)}{b(x,t)} \leq \beta, \text{where} \\ &\beta = \sqrt{\frac{b_{max}}{b_{min}}} \end{aligned} \tag{4.22}$$

The control law is designed to be robust against the constrained increasing uncertainty, therefore b is known as the design margin gain.

The final control law is defined as,

$$u(x,t) = \frac{1}{\hat{b}(x,t)}\Big(\hat{u}(x,t) - k(x,t)\text{sign}(\alpha(t)\Big) \tag{4.23}$$

with

$$k(x,t) \geq \beta(x,t)(F(x,t)+\eta) + (\beta(x,t)-1)|\hat{u}(t)| \tag{4.24}$$

meets the sliding condition. The similar approach can be conducted to derive the control law for higher order systems.

4.4 Sliding surface and control law design for quadcopter UAVs

The architecture of flight control system can be illustrated as in Figure 4.4. One of its output performances depends on sliding surface design. Before designing such surface, it is required to define the states associated with the fully and under-actuated subsystems to ensure that those states converge to the reference (desired values). Therefore, for the fully-actuated subsystem, there are two states, namely z and ψ, while there are 4 states, namely x, y, θ and ϕ for the under-actuated subsystem.

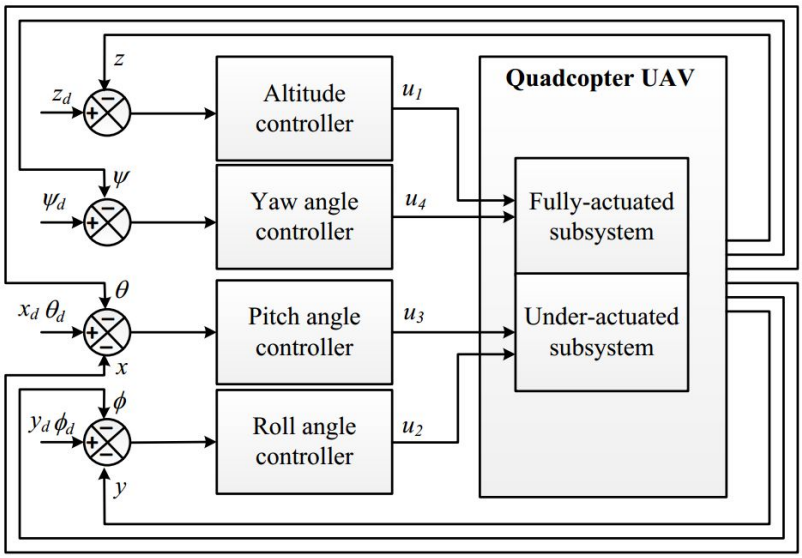

Figure 4.4. Quadcopter control architecture

Referring to equation (4.12), the sliding manifolds for quadcopter UAVs can be expressed by the following equations,

$$s_\phi = (\dot{\phi} - \dot{\phi}_d) + \lambda_\phi[(\phi - \phi_d) + \phi_d(1 - \varsigma)] + \lambda_{i\phi} \int_0^\infty (\phi - \phi_d)dt \tag{4.25}$$

$$s_\theta = (\dot{\theta} - \dot{\theta}_d) + \lambda_\theta[(\theta - \theta_d) + \theta_d(1 - \varsigma)] + \lambda_{i\theta} \int_0^\infty (\theta - \theta_d)dt \tag{4.26}$$

$$s_\psi = (\dot{\psi} - \dot{\psi}_d) + \lambda_\psi[(\psi - \psi_d) + \psi_d(1 - \varsigma)] + \lambda_{i\psi} \int_0^\infty (\psi - \psi_d)dt \tag{4.27}$$

$$s_z = (\dot{z} - \dot{z}_d) + \lambda_z[(z - z_d) + z_d(1 - \varsigma)] + \lambda_{iz} \int_0^\infty (z - z_d)dt \tag{4.28}$$

$$s_x = (\dot{x} - \dot{x}_d) + \lambda_x[(x - x_d) + x_d(1 - \varsigma)] + \lambda_{ix} \int_0^\infty (x - x_d)dt \tag{4.29}$$

$$s_y = (\dot{y} - \dot{y}_d) + \lambda_y[(y - y_d) + y_d(1 - \varsigma)] + \lambda_{iy} \int_0^\infty (y - y_d)dt \tag{4.30}$$

The tracking errors will asymptotically converge to zero with the conditions $s = 0$ and $\dot{s} = 0$ are fulfilled at every $t \geq t_0$.

Considering the dynamic model system in equations (3.47) - (3.55), sliding surface in equations (4.25) - (4.28), the general control law in equations (4.23) and sliding condition $\dot{s} = 0$, control input can be expressed as,

$$u_1 = \frac{m}{c\phi c\theta}\Big(- k_z \text{sign}(s_z) - \mu_z s_z - \lambda_z \dot{e}_z - \lambda_{iz} e_z + \ddot{z}_d + g \Big) \tag{4.31}$$

$$u_2 = \frac{I_x}{l}\Big(- k_\phi \text{sign}(s_\phi) - \mu_\phi s_\phi - \lambda_\phi \dot{e}_\phi - \lambda_{i\phi} e_\phi - \frac{\dot{\theta}\dot{\psi}}{I_x}(I_y - I_z) - \frac{J_r}{I_{xx}}\dot{\theta}\Omega_r + \ddot{\phi}_d\Big) \quad (4.32)$$

$$u_3 = \frac{I_y}{l}\Big(- k_\theta \text{sign}(s_\theta) - \mu_\theta s_\theta - \lambda_\theta \dot{e}_\theta - \lambda_{i\theta} e_\theta - \frac{\dot{\phi}\dot{\psi}}{I_y}(I_z - I_x) + \frac{J_r}{I_{yy}}\dot{\phi}\Omega_r + \ddot{\theta}_d\Big) \quad (4.33)$$

$$u_4 = I_z\Big(- k_\psi \text{sign}(s_\psi) - \mu_\psi s_\psi - \lambda_\psi \dot{e}_\psi - \lambda_{i\psi} e_\psi - \frac{\dot{\phi}\dot{\theta}}{I_z}(I_x - I_y) + \ddot{\psi}_d\Big) \quad (4.34)$$

where μ is a positif constant. Furthermore, by using dynamical model of x and y as stated in equations (3.53) and (3.54), sliding manifolds in equations (4.29) and (4.30), and general form of control law in equation (4.23), the calculation of the desired pitch (θ_d) and roll (ϕ_d) angles with respect to x and y axes errors are presented as,

$$\theta_d = \text{atan}\left(\frac{(-k_x \text{sign}(s_x) - \mu_x s_x + \ddot{x}_d - \lambda_x \dot{e_x} - \lambda_{ix} e_x) c\phi c\theta}{-k_z \text{sign}(s_z) - \mu_z s_z + \ddot{z} - \lambda_z \dot{e_z} - \lambda_{iz} e_z + g}\right) \quad (4.35)$$

$$\phi_d = -\text{atan}\left(\frac{(-k_y \text{sign}(s_y) - \mu_y s_y + \ddot{y}_d - \lambda_y \dot{e_y} - \lambda_{iy} e_y) c\phi c\theta}{-k_z \text{sign}(s_z) - \mu_z s_z + \ddot{z} - \lambda_z \dot{e_z} - \lambda_{iz} e_z + g}\right) \quad (4.36)$$

4.5 Chattering avoidance: elimination and attenuation

A robust sliding mode appears only when the trajectory of state $x(t)$ of the controlled plant concurs with the expected trajectory at every $t \geq t_0$ for some t_0. This may need high-frequency switching infinitely. In real-time application systems, a switched controller as stated in equation (4.23) has limitations which restrict switching to a limited frequency. The oscillation occurs within a neighbourhood of the sliding surface (DeCarlo et al., 1988). This oscillation as illustrated in Figure 4.3 is called chattering.

Control laws which meet the sliding condition in equation (4.4), lead to robust performance in the face of model inaccuracy, are discontinuous along the switching surface $s(t)$, thus this is generating control chattering. This phenomenon is unwanted due to high frequency control operation, and may evoke high-frequency dynamics disregarded in the process of modelling. In numerous real-time applications of control systems including aircraft systems, smooth control signal plays a crucial role in developing high performance of such systems. For example, the aerodynamic surfaces of aircraft cannot move backward and forward with high frequency, however in the meantime, it is important to keep the

robustness of the control system in dealing with uncertainty and disturbance.

4.5.1 Chattering elimination: quasi-sliding mode

The first solution to eliminate the chattering is to employ QuasiSMC by replacing the discontinuous function u_{sw} in the control law in equation (4.17) with continuous term, which can be expressed as,

$$\text{sign}(s) \approx \frac{s}{|s| + \varepsilon} \tag{4.37}$$

where ε is a small positive constant. It can be seen that,

$$\lim_{\varepsilon \longrightarrow 0} \frac{s}{|s| + \varepsilon} = \text{sign}(s) \tag{4.38}$$

for $s \neq 0$. The value of ε should be selected to lead to a smooth control signal while maintaining an ideal performance of control method. The function in equation (4.37) is shown in Figure 4.5.

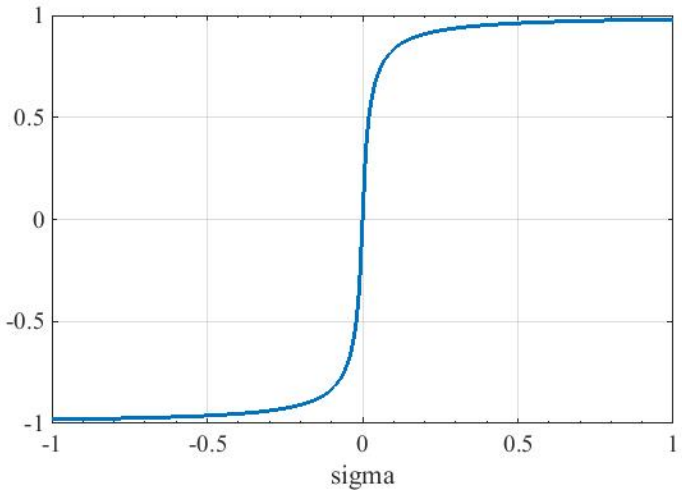

Figure 4.5. Sigmoid function

Hence, the control law as stated in equation (4.23) is expressed as,

$$u(x,t) = \frac{1}{\hat{b}(x,t)} \left(\hat{u}(x,t) - k(x,t) \frac{s}{|s| + \varepsilon} \right) \tag{4.39}$$

The complete control input for quadcopter UAVs can be presented as,

$$u_1 = \frac{m}{c\phi c\theta}\left(- k_z \frac{s_z}{|s_z| + \varepsilon} - \mu_z s_z - \lambda_z \dot{e}_z - \lambda_{iz} e_z + \ddot{z}_d + g \right) \tag{4.40}$$

$$u_2 = \frac{I_x}{l}\left(- k_\phi \frac{s_\phi}{|s_\phi| + \varepsilon} - \mu_\phi s_\phi - \lambda_\phi \dot{e}_\phi - \lambda_{i\phi} e_\phi - \frac{\dot{\theta}\dot{\psi}}{I_x}(I_y - I_z) - \frac{J_r}{I_{xx}}\dot{\theta}\Omega_r + \ddot{\phi}_d \right) \tag{4.41}$$

$$u_3 = \frac{I_y}{l}\left(- k_\theta \frac{s_\theta}{|s_\theta| + \varepsilon} - \mu_\theta s_\theta - \lambda_\theta \dot{e}_\theta - \lambda_{i\theta} e_\theta - \frac{\dot{\phi}\dot{\psi}}{I_y}(I_z - I_x) + \frac{J_r}{I_{yy}}\dot{\phi}\Omega_r + \ddot{\theta}_d \right) \tag{4.42}$$

$$u_4 = I_z\left(- k_\psi \frac{s_\psi}{|s_\psi| + \varepsilon} - \mu_\psi s_\psi - \lambda_\psi \dot{e}_\psi - \lambda_{i\psi} e_\psi - \frac{\dot{\phi}\dot{\theta}}{I_z}(I_x - I_y) + \ddot{\psi}_d \right) \tag{4.43}$$

The calculation of the desired pitch (θ_d) and roll (ϕ_d) angles with x and y axes errors are given as,

$$\theta_d = \text{atan}\left(\frac{(-k_x \frac{s_x}{|s_x| + \varepsilon} - \mu_x s + \ddot{x_d} - \lambda_x \dot{e_x} - \lambda_{ix} e_x) c\phi c\theta}{-k_z \frac{s_z}{|s_z| + \varepsilon} - \mu_z s_z - \frac{k_3 \dot{z}}{m} + \ddot{z} - \lambda_z \dot{e_z} - \lambda_{iz} e_z + g} \right) \tag{4.44}$$

$$\phi_d = -\text{atan}\left(\frac{(-k_y \frac{s_y}{|s_y| + \varepsilon} - \mu_y \sigma_y + \ddot{y_d} - \lambda_y \dot{e_y} - \lambda_{iy} e_y) c\phi c\theta}{-k_z \frac{s_z}{|s_z| + \varepsilon} - \mu_z s_z - \frac{k_3 \dot{z}}{m} + \ddot{z} - \lambda_z \dot{e_z} - \lambda_{iz} e_z + g} \right) \tag{4.45}$$

4.5.2 Chattering elimination: sliding mode-based interval type-2 fuzzy control

In this subsection, interval type-2 fuzzy logic systems (IT2FLSs) are proposed to improve the performance of the controller, such as eliminating chattering phenomenon, dealing with nonlinearity, and being insensitive to uncertainties and disturbances of SMC. There are two types of fuzzy logic systems (FLSs), including: “type-1 FLSs” and “type-2 FLSs”. Type-1 FLSs are developed with crisp exact membership functions (MFs) that do not permit any uncertainties in the membership values, whereas type-2 FLSs have a “Footprint of Uncertainty”, and are constructed with three dimensional MFs that can deal with uncertainties. Furthermore, type-2 FLSs may provide a better method to deal with nonlinear systems, multivariable, and uncertainty models caused by imprecision or unknown physical parameters (Mendel et al., 2014).

In general type-2 FLSs, there are 2 types of MFs, namely primary MFs and secondary MFs. Computational burden is a constrain of applying such MFs type due to varying

grades of membership for primary and secondary MFs (from 0 to 1) as seen in Figure 4.6. Therefore, in order to reduce such burden, the grade of secondary MFs is set to a value 1. This simplification is called IT2FLSs as shown in Figure 4.7.

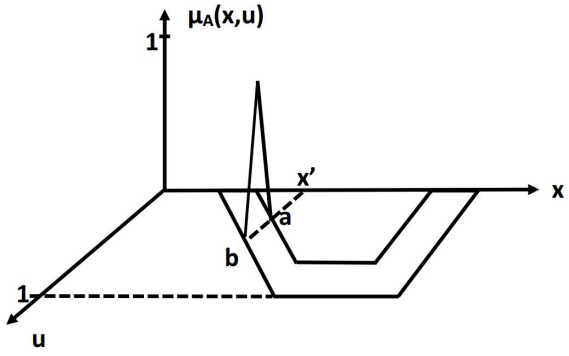

Figure 4.6. General type-2 fuzzy MFs

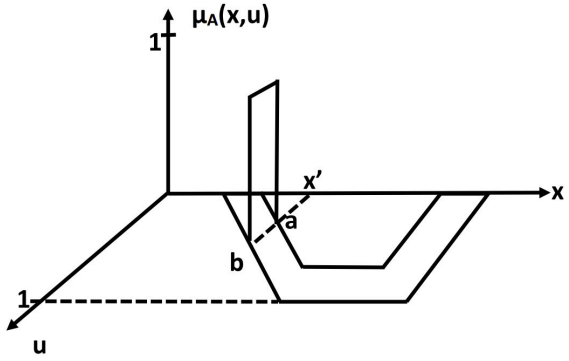

Figure 4.7. Interval type-2 fuzzy MFs

Generally, the architecture of IT2FLSs consists of fuzzification, fuzzy inference and rule base, type reduction, and deffuzification as shown in Figure 4.8. This configuration is similar to type-1 FLSs except in the output processing block. In the IT2FLSs, the fuzzy output sets proceed to type reduction, then defuzzification to produce crisp outputs.

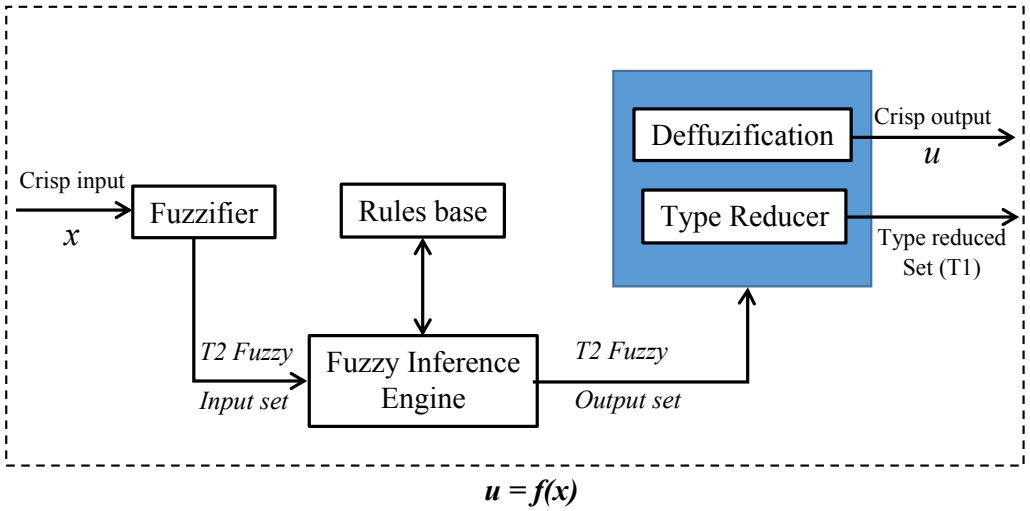

Figure 4.8. Architecture of IT2FLS

In terms of SMC, the chattering phenomenon is produced by switching control signal ($sign$ function) which has high speed switching across the surface to maintain sliding condition. Therefore, the proposed method of sliding mode-based interval type-2 fuzzy control (IT2FSMC) tries to eliminate the chattering during sliding mode condition in the sliding surface by employing IT2FLSs in the control signal.

The membership function of fuzzy input is built by the values of sliding surface (s) as

seen in Figure 4.9, while the fuzzy output is depicted in Figure 4.10. Each of these has 7 membership functions, namely: Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (Z), Positive Small (PS), Positive Medium (PM), and Positive Big (PB).

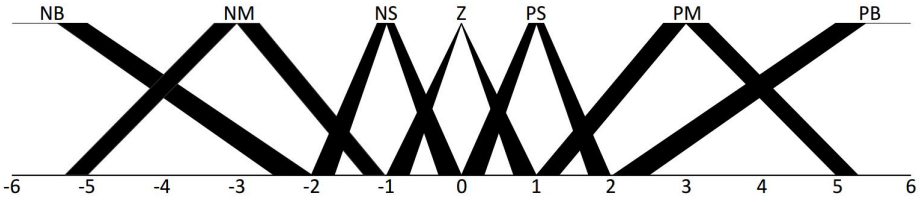

Figure 4.9. Sliding surface (s) MFs

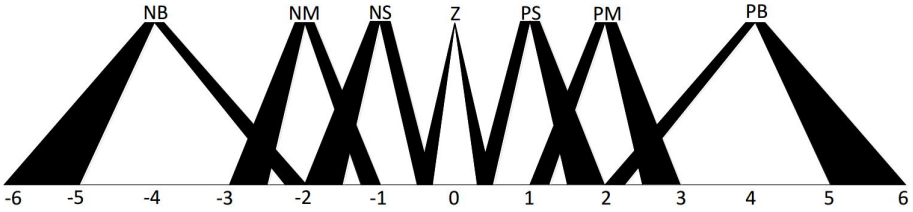

Figure 4.10. Fuzzy output MFs of SMC

Considering the fuzzy input and output, the rules-base to produce the desired switching control in equation (4.17) is shown in Table 4.1.

Table 4.1. The Rules of IT2FSMC

	s **(Sliding surface)**						
	NB	NM	NS	Z	PS	PM	PB
u_{f2}	PB	PM	PS	Z	NS	NM	NB

The switching control is obtained after type reduction and defuzzification process, and the control signal can be expressed as,

$$u_{sw} = ku_{f2} \tag{4.46}$$

where k is a gain and u_{f2} represents the crisp values obtained from IT2FSMC. Thus, using equation (4.23), the control law of SMC is expressed as:

$$u(x,t) = \frac{1}{\hat{b}(x,t)}\left(\hat{u}(x,t) - ku_{f2}\right) \tag{4.47}$$

The control input for quadcopter UAVs can be presented as:

$$u_1 = \frac{m}{c\phi c\theta}\left(- k_z u_{f2z} - \mu_z s_z - \lambda_z \dot{e}_z - \lambda_{iz} e_z + \ddot{z}_d + g\right) \tag{4.48}$$

$$u_2 = \frac{I_x}{l}\left(- k_\phi u_{f2\phi} - \mu_\phi s_\phi - \lambda_\phi \dot{e}_\phi - \lambda_{i\phi} e_\phi - \frac{\dot{\theta}\dot{\psi}}{I_x}(I_y - I_z) - \frac{J_r}{I_{xx}}\dot{\theta}\Omega_r + \ddot{\phi}_d \right) \quad (4.49)$$

$$u_3 = \frac{I_y}{l}\left(- k_\theta u_{f2\theta} - \mu_\theta s_\theta - \lambda_\theta \dot{e}_\theta - \lambda_{i\theta} e_\theta - \frac{\dot{\phi}\dot{\psi}}{I_y}(I_z - I_x) + \frac{J_r}{I_{yy}}\dot{\phi}\Omega_r + \ddot{\theta}_d \right) \quad (4.50)$$

$$u_4 = I_z\left(- k_\psi u_{f2\psi} - \mu_\psi s_\psi - \lambda_\psi \dot{e}_\psi - \lambda_{i\psi} e_\psi - \frac{\dot{\phi}\dot{\theta}}{I_z}(I_x - I_y) + \ddot{\psi}_d \right) \quad (4.51)$$

The calculation of the desired pitch (θ_d) and roll (ϕ_d) angles with x and y axes errors are presented as,

$$\theta_d = \text{atan}\left(\frac{(-k_x u_{f2x} - \mu_x s_x + \ddot{x_d} - \lambda_x \dot{e_x} - \lambda_{ix} e_x) c\phi c\theta}{-k_z u_{f2z} - \mu_z s_z + \ddot{z} - \lambda_z \dot{e_z} - \lambda_{iz} e_z + g} \right) \quad (4.52)$$

$$\phi_d = -\text{atan}\left(\frac{(-k_y u_{f2y} - \mu_y s_y + \ddot{y_d} - \lambda_y \dot{e_y} - \lambda_{iy} e_y) c\phi c\theta}{-k_z u_{f2z} - \mu_z s_z + \ddot{z} - \lambda_z \dot{e_z} - \lambda_{iz} e_z + g} \right) \quad (4.53)$$

4.5.3 Chattering attenuation: super-twisting algorithm

Super-twisting algorithm of sliding mode control (STASMC) has been worked out for the case of systems with relative degree one so as to keep off chattering in variable structure systems. The state trajectory of the s and $\dot{s}$ phase plane is shown in Figure 4.11. It twists and approaches the starting point on the state space. At last, it unites to the root of the phase plane.

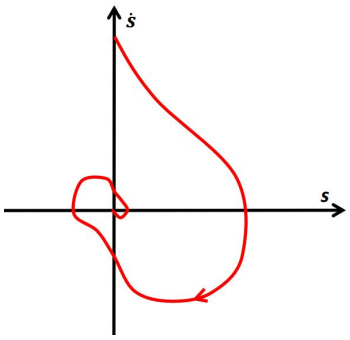

Figure 4.11. Phase plane trajectory of super-twisting algorithm

The STASMC can be viewed as a nonlinear form of the conventional PI controller. This similarity is clearer by observing Figure 4.12.

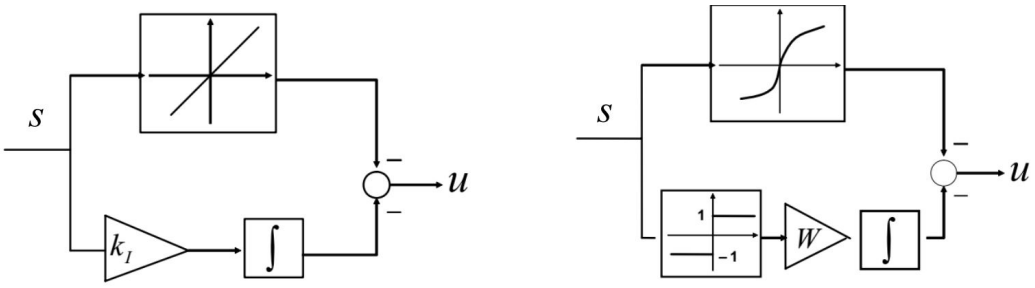

Figure 4.12. Block scheme of PI (left) and Super-Twisting (right) algorithm

Furthermore, the STASMC for perturbation and chattering attenuation is defined by

$$\begin{aligned} u_{sw} &= -k_1\sqrt{|s|}\text{sign}(s) + v \\ \dot{v} \quad &= -k_2\text{sign}(s) \end{aligned} \tag{4.54}$$

A suitable method for tuning its parameters is the accompanying pair of relationships

$$\begin{aligned} k_1 &= \sqrt{\kappa} \\ k_2 &= 1.1\kappa \end{aligned} \tag{4.55}$$

where $\kappa > 0$ is chosen to be adequately large. In practice, increase κ progressively until good results are noticeable in the closed loop system. The tuning of single-parameter through "trial and error" is suitable especially in practical implementation.

Therefore, the total control signal for nonlinear systems is expressed as,

$$u(x,t) = \frac{1}{\hat{b}(x,t)}\left(\hat{u}(x,t) - k_1\sqrt{|s|}\text{sign}(s) - k_2\int_0^\infty \text{sign}(s)dt\right) \tag{4.56}$$

Furthermore, the switching control law for z, roll (ϕ), pitch(θ), yaw(ψ), x, and y states are defined as,

$$u_{swz} = -k_{1z}\sqrt{|s_z|}\text{sign}(s_z) - k_{2z}\int_0^\infty \text{sign}(s_z)dt \tag{4.57}$$

$$u_{sw\phi} = -k_{1\phi}\sqrt{|s_\phi|}\text{sign}(s_\phi) - k_{2\phi}\int_0^\infty \text{sign}(s_\phi)dt \tag{4.58}$$

$$u_{sw\theta} = -k_{1\theta}\sqrt{|s_\theta|}\text{sign}(s_\theta) - k_{2\theta}\int_0^\infty \text{sign}(s_\theta)dt \tag{4.59}$$

$$u_{sw\psi} = -k_{1\psi}\sqrt{|s_\psi|}\text{sign}(s_\psi) - k_{2\psi}\int_0^\infty \text{sign}(s_\psi)dt \tag{4.60}$$

$$u_{swx} = -k_{1x}\sqrt{|s_x|}\text{sign}(s_x) - k_{2x}\int_0^\infty \text{sign}(s_x)dt \tag{4.61}$$

$$u_{swy} = -k_{1y}\sqrt{|s_y|}\text{sign}(s_y) - k_{2y}\int_0^\infty \text{sign}(s_y)dt \tag{4.62}$$

Therefore, the control input for quadcopter UAVs can be presented as,

$$u_1 = \frac{m}{c\phi c\theta}\left(u_{swz} - \mu_z s_z - \lambda_z \dot{e}_z - \lambda_{iz} e_z + \ddot{z_d} + g\right) \tag{4.63}$$

$$u_2 = \frac{I_x}{l}\left(u_{sw\phi} - \mu_\phi s_\phi - \lambda_\phi \dot{e}_\phi - \lambda_{i\phi} e_\phi - \frac{\dot{\theta}\dot{\psi}}{I_x}(I_y - I_z) - \frac{J_r}{I_{xx}}\dot{\theta}\Omega_r + \ddot{\phi}_d\right) \tag{4.64}$$

$$u_3 = \frac{I_y}{l}\left(u_{sw\theta} - \mu_\theta s_\theta - \lambda_\theta \dot{e}_\theta - \lambda_{i\theta} e_\theta - \frac{\dot{\phi}\dot{\psi}}{I_y}(I_z - I_x) + \frac{J_r}{I_{yy}}\dot{\phi}\Omega_r + \ddot{\theta}_d\right) \tag{4.65}$$

$$u_4 = I_z\left(u_{sw\psi} - \mu_\psi s_\psi - \lambda_\psi \dot{e}_\psi - \lambda_{i\psi} e_\psi - \frac{\dot{\phi}\dot{\theta}}{I_z}(I_x - I_y) + \ddot{\psi}_d\right) \tag{4.66}$$

The calculation of the desired pitch (θ_d) and roll (ϕ_d) angles with x and y axes errors are given as,

$$\theta_d = \text{atan}\left(\frac{(u_{swx} - \mu_x s_x + \ddot{x_d} - \lambda_x \dot{e_x} - \lambda_{ix} e_x)c\phi c\theta}{u_{swz} - \mu_z s_z + \ddot{z} - \lambda_z \dot{e_z} - \lambda_{iz} e_z + g}\right) \tag{4.67}$$

$$\phi_d = -\text{atan}\left(\frac{(u_{swy} - \mu_y s_y + \ddot{y_d} - \lambda_y \dot{e_y} - \lambda_{iy} e_y)c\phi c\theta}{u_{swz} - \mu_z s_z + \ddot{z} - \lambda_z \dot{e_z} - \lambda_{iz} e_z + g}\right) \tag{4.68}$$

4.5.4 Chattering attenuation: dynamic sliding mode control

Dynamic sliding method control (DSMC) is utilized to create a new sliding function. It is used to build a sliding s in the conventional sliding mode into a new sliding function σ. The switching function is with respect to the first or high order derivative in the control input. Additionally, it can move the discontinuous terms into the first or high order derivative in the control input. Consequently, a continuous dynamic sliding mode control pattern is obtained and the chattering problem can be reduced adequately.

Consider a 2^{nd}- order of nonlinear system as,

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= f(\mathbf{x}) + b(\mathbf{x})u(t) + d(t) \\ y &= x_1 \end{aligned} \tag{4.69}$$

where **x** = [x_1 x_2] represents states vector while x_1 and x_2 are system states. y and $u(t)$ denote output state and control input respectively, while $f(x)$ and $b(x)$ represent general nonlinear functions. $d(t)$ is uncertainty, and $|d(t)| \leqslant D_0$, $|\dot{d}(t)| \leqslant D$.

The tracking error and the integral sliding function are defined as,

$$\begin{aligned} e &= y - y_d \\ s &= \dot{e} + ce + \int_0^\infty \lambda_i e \end{aligned} \tag{4.70}$$

where c is a positive constant and must be Hurwitz. Thus,

$$\begin{aligned} \dot{s} &= \ddot{e} + c\dot{e} + \lambda_i e \\ \dot{s} &= f(x) + b(x)u(t) + d(t) - \ddot{y}_d + c\dot{e} + \lambda_i e \end{aligned} \tag{4.71}$$

Furthermore, a new dynamic sliding manifold is constructed as (Liu and Wang, 2011),

$$\sigma = \dot{s} + \lambda s \tag{4.72}$$

where λ is a positive constant and must be Hurwitz. When $\sigma = 0$, $\dot{\sigma} = 0$ is a asymptotically stable, hence, $e \longrightarrow 0$ and $\dot{e} \longrightarrow 0$. From equation (4.72), stability analysis is expressed as,

$$\begin{aligned} \sigma &= \dot{s} + \lambda s \\ &= f(x) + b(x)u(t) + d(t) - \ddot{y}_d + c\dot{e} + \lambda_i e + \lambda s \end{aligned} \tag{4.73}$$

Thus,

$$\begin{aligned} \dot{\sigma} &= \dot{f}(x) + \dot{b}(x)u(t) + b(x)\dot{u}(t) + \dot{d}(t) - \dddot{y}_d + c\ddot{e} + \lambda_i\dot{e} + \lambda\dot{s} \\ &= \dot{f}(x) + \dot{b}(x)u(t) + b(x)\dot{u}(t) + \dot{d}(t) - \dddot{y}_d + c(f(x) + b(x)u(t) + d(t) \\ &\quad - \ddot{y}_d) + \lambda_i\dot{e} + \lambda(f(x) + b(x)u(t) + d(t) - \ddot{y}_d + c\dot{e} + \lambda_i e) \\ &= \dot{f}(x) - (c+\lambda)\ddot{y}_d - \dddot{y}_d + \dot{d}(t) + (c+\lambda)d(t) \\ &\quad + (\dot{b}(x) + cb(x) + \lambda b(x))u(t) + (c+\lambda)f(x) + b(x)\dot{u}(t) + \lambda\lambda_i e + \lambda c\dot{e} + \lambda_i\dot{e} \end{aligned} \tag{4.74}$$

Select the dynamic controller law as,

$$\begin{aligned}\dot{u}(t) = \frac{1}{b(x)}\Big(&- \dot{f}(x) + (c+\lambda)\ddot{y}_d + \dddot{y}_d - (\dot{b}(x) + cb(x) + \lambda b(x))u(t) \\ &- (c+\lambda)f(x) - \lambda\lambda_i e - \lambda c\dot{e} - \lambda_i \dot{e} - k\text{sign}(\sigma) - \mu\sigma\Big)\end{aligned} \tag{4.75}$$

where k and μ are positive constants. Therefore, stability analysis is expressed as,

$$\dot{\sigma} = \dot{d}(t) + (c+\lambda)d(t) - k\text{sign}(\sigma) \tag{4.76}$$

Let $k > D + (c+\lambda)D_0$, therefore,

$$\begin{aligned}\sigma\dot{\sigma} &= \sigma(\dot{d}(t) + (c+\lambda)d(t) - k\text{sign}(\sigma)) \\ &= \sigma(\dot{d}(t) + (c+\lambda)d(t)) - k|\sigma| \leqslant (D + (c+\lambda)D_0)\sigma - k|\sigma| < 0\end{aligned} \tag{4.77}$$

Sliding surface for the quadcopter UAVs can be rewritten as,

$$\begin{aligned}
s_\phi &= c_\phi\chi_\phi + \dot{e}_\phi + \lambda_{i\phi}\int_0^\infty e_\phi \\
s_\theta &= c_\theta\chi_\theta + \dot{e}_\theta + \lambda_{i\theta}\int_0^\infty e_\theta \\
s_\psi &= c_\psi\chi_\psi + \dot{e}_\psi + \lambda_{i\psi}\int_0^\infty e_\psi \\
s_z &= c_z\chi_z + \dot{e}_z + \lambda_{iz}\int_0^\infty e_z \\
s_x &= c_x\chi_x + \dot{e}_x + \lambda_{ix}\int_0^\infty e_x \\
s_y &= c_y\chi_y + \dot{e}_y + \lambda_{iy}\int_0^\infty e_y
\end{aligned} \tag{4.78}$$

Referring to equation (4.71), the dynamic of sliding surface can be expressed as,

$$\begin{aligned}
\dot{s}_\phi &= c_\phi\dot{e}_\phi + f_\phi + b_\phi u_2 - \ddot{\phi}_d + \lambda_{i\phi}e_\phi \\
\dot{s}_\theta &= c_\theta\dot{e}_\theta + f_\theta + b_\theta u_3 - \ddot{\theta}_d + \lambda_{i\theta}e_\theta \\
\dot{s}_\psi &= c_\psi\dot{e}_\psi + f_\psi + b_\psi u_4 - \ddot{\psi}_d + \lambda_{i\psi}e_\psi \\
\dot{s}_z &= c_z\dot{e}_z + f_z + b_z u_1 - \ddot{z}_d + \lambda_{iz}e_z \\
\dot{s}_x &= c_x\dot{e}_x + f_x + b_x u_1 - \ddot{x}_d + \lambda_{ix}e_x \\
\dot{s}_y &= c_y\dot{e}_y + f_y + b_y u_1 - \ddot{y}_d + \lambda_{iy}e_y
\end{aligned} \tag{4.79}$$

Furthermore, the new dynamic sliding manifold is constructed as,

$$\begin{aligned}
\sigma_\phi &= \dot{s}_\phi + \lambda_\phi s_\phi \\
\sigma_\theta &= \dot{s}_\theta + \lambda_\theta s_\theta \\
\sigma_\psi &= \dot{s}_\psi + \lambda_\psi s_\psi \\
\sigma_z &= \dot{s}_z + \lambda_z s_z \\
\sigma_x &= \dot{s}_x + \lambda_x s_x \\
\sigma_y &= \dot{s}_y + \lambda_y s_y
\end{aligned} \tag{4.80}$$

When $\sigma = 0$, $\dot{\sigma} = 0$ is asymptotically stable and referring to equation (4.75), the control input of quadcopter UAVs can be defined as,

$$\begin{aligned}
u_1 = \int_0^\infty \frac{1}{b_z}\Big(&- \dot{f}_z + (c_z + \lambda_z)\ddot{z}_d + \dddot{z}_d - (\dot{b}_z + c_z b_z + \lambda_z b_z)u_1 \\
&- (c_z + \lambda_z)f_z - \lambda_z c_z \dot{e}_z - \lambda_{iz}\dot{e}_z - \lambda_z \lambda_{iz} e_z - k_z \text{sign}(\sigma_z) - \mu_z \sigma_z \Big) dt
\end{aligned} \tag{4.81}$$

$$\begin{aligned}
u_2 = \int_0^\infty \frac{1}{b_\phi}\Big(&- \dot{f}_\phi + (c_\phi + \lambda_\phi)\ddot{\phi}_d + \dddot{\phi}_d - (\dot{b}_\phi + c_\phi b_\phi + \lambda_\phi b_\phi)u_2 \\
&- (c_\phi + \lambda_\phi)f_\phi - \lambda_\phi c_\phi \dot{e}_\phi - \lambda_{i\phi}\dot{e}_\phi - \lambda_\phi \lambda_{i\phi} e_\phi - k_\phi \text{sign}(\sigma_\phi) - \mu_\phi \sigma_\phi \Big) dt
\end{aligned} \tag{4.82}$$

$$\begin{aligned}
u_3 = \int_0^\infty \frac{1}{b_\theta}\Big(&- \dot{f}_\theta + (c_\theta + \lambda_\theta)\ddot{\theta}_d + \dddot{\theta}_d - (\dot{b}_\theta + c_\theta b_\theta + \lambda_\theta b_\theta)u_3 \\
&- (c_\theta + \lambda_\theta)f_\theta - \lambda_\theta c_\theta \dot{e}_\theta - \lambda_{i\theta}\dot{e}_\theta - \lambda_\theta \lambda_{i\theta} e_\theta - k_\theta \text{sign}(\sigma_\theta) - \mu_\theta \sigma_\theta \Big) dt
\end{aligned} \tag{4.83}$$

$$\begin{aligned}
u_4 = \int_0^\infty \frac{1}{b_\psi}\Big(&- \dot{f}_\psi + (c_\psi + \lambda_\psi)\ddot{\psi}_d + \dddot{\psi}_d - (\dot{b}_\psi + c_\psi b_\psi + \lambda_\psi b_\psi)u_4 \\
&- (c_\psi + \lambda_\psi)f_\psi - \lambda_\psi c_\psi \dot{e}_\psi - \lambda_{i\psi}\dot{e}_\psi - \lambda_\psi \lambda_{i\psi} e_\psi - k_\psi \text{sign}(\sigma_\psi) - \mu_\psi \sigma_\psi \Big) dt
\end{aligned} \tag{4.84}$$

The calculation of the desired pitch (θ_d) and roll (ϕ_d) angles with x and y axes errors is given as,

$$\begin{aligned}
\theta_d = \int_0^\infty \frac{1}{u_1}\Big(&- \dot{f}_x + (c_x + \lambda_x)\ddot{x}_d + \dddot{x}_d - (\dot{u}_1 + c_x u_1 + \lambda_x u_1)\theta_d \\
&- (c_x + \lambda_x)f_x - \lambda_x c_x \dot{e}_x - \lambda_{ix}\dot{e}_x - \lambda_x \lambda_{ix} e_x - k_x \text{sign}(\sigma_x) - \mu_x \sigma_x \Big) dt
\end{aligned} \tag{4.85}$$

$$\begin{aligned}
\phi_d = -\int_0^\infty \frac{1}{u_1}\Big(&- \dot{f}_y + (c_y + \lambda_y)\ddot{y}_d + \dddot{y}_d - (\dot{u}_1 + c_y u_1 + \lambda_y u_1)\phi_d \\
&- (c_y + \lambda_y)f_y - \lambda_y c_y \dot{e}_y - \lambda_{iy}\dot{e}_y - \lambda_y \lambda_{iy} e_y - k_y \text{sign}(\sigma_y) - \mu_y \sigma_y \Big) dt
\end{aligned} \tag{4.86}$$

4.6 Summary

In this chapter, four methods of sliding mode controls (SMCs) including quasi-sliding mode control (QuasiSMC), sliding mode-based interval type-2 fuzzy control (IT2FSMC), super-twisting algorithm of sliding mode control (STASMC) and dynamic sliding mode control (DSMC) have been designed to control dynamic system of quadcopter UAV. The main idea of using those methods is that to reduce chattering phenomenon associated with SMC which may degrade the overall system performance. Set-point weighting function has been employed to diminish the overshoot occurrence due to integral terms in SMC. The overall performance of the methods will be evaluated and compared through numerical simulation and experimental validation.

Chapter 5

Design of sliding mode observer for quadcopter UAVs

5.1 Introduction

In control engineering theory, a state estimator is a system that gives an approximation of a real system internal state, from measurements of the input and actual system output. It is normally a computer-based implementation, and becomes the basis for various practical applications.

The availability of system state information is crucial to solving many control application problems; for instance, designing a stable system utilizing state feedback. The sliding mode controller is developed with the assumption that all required state information are available for the control process. Practically, in most cases the physical state information of the plant cannot be acquired by direct measurement. This is because of sensor limitation, or high cost to make accurate measurements. Instead, the required internal state can be estimated by observing the system outputs. If a system is observable, it is viable to completely reconstruct the state information from measured state using the state estimator or observer.

Sliding mode observers (SMOs), as nonlinear systems, have special properties in that they have the capacity to produce a sliding motion on the error between measured and estimated state of the observer. Thus, an SMO generates estimated states that are accurately proportionate with actual output of the plant. Moreover, the observers are believed to be at the cutting-edge of robust approaches for state estimation by demonstrating the ability to

deal with model errors and nonlinear systems.

5.2 Basic observer design

5.2.1 Systems with a single measurement

Consider a 2^{nd}- order system of nonlinear system as,

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= f(\mathbf{x}) + b(\mathbf{x})u(t) + d(t) \\ y &= x_1 \end{aligned} \tag{5.1}$$

where f is a nonlinear dynamic system and **x** = $[x_1 \quad x_2]^T$ is the state vector of the system. If the measured state is x_1, an observer structure is expressed as (Slotine et al., 1986),

$$\begin{aligned} \dot{\hat{x}}_1 &= -\alpha_1 e_1 + \hat{x}_2 - k_1 \text{sign}(e_1) \\ \dot{\hat{x}}_2 &= -\alpha_2 e_1 + \hat{f} + \hat{b}u - k_2 \text{sign}(e_1) \end{aligned} \tag{5.2}$$

where e_1 = $\hat{x}_1$ - x_1, $\hat{x}_i$, and u represent estimation error, estimated state of x_i, and control law respectively, while $\hat{f}$ and $\hat{b}$ denote the best estimated models of f and b respectively, and the constants α_i are selected as in a Luenberger estimator (which relate to k_1 = 0, k_2 = 0) so as to locate the poles of the linearised system at expected places $-\zeta_i$. The sliding gain k_1 is selected as a bound on estimated state error of x_2 in the steady state condition, and k_2 is selected to be larger than modelling errors. The resulting error dynamics of the approach can be composed as (Slotine et al., 1986),

$$\begin{aligned} \dot{e_1} &= -\alpha_1 e_1 + e_2 - k_1 \text{sign}(e_1) \\ \dot{e_2} &= -\alpha_2 e_1 + \Delta f - k_2 \text{sign}(e_1) \end{aligned} \tag{5.3}$$

The value of $\Delta f = \hat{f} - f$ relies on the computational intricacy and the modelling effort allowed in the estimator. In this research, dynamic uncertainty Δf is assumed to be bounded explicitly. For simplicity, the known nonlinear terms may likewise be handled as limited error (using known bounds on the state of the actual system) and incorporated into Δf.

The effect of Δf is adjusted by using the knowledge of its (generally time-varying) bound.

The approach can be directly applied to n^{th}- order systems in companion form as,

$$x_1^{(n)} = f(\mathbf{x}) + b(\mathbf{x})u(t) \tag{5.4}$$

where x_1 is the only measured state of the system. Therefore, the observer structure is expressed as (Slotine et al., 1986),

$$\begin{aligned}
\dot{\hat{x}}_1 &= -\alpha_1 e_1 + \hat{x}_2 - k_1 \text{sign}(e_1) \\
\dot{\hat{x}}_2 &= -\alpha_2 e_1 + \hat{x}_3 - k_2 \text{sign}(e_1) \\
&..... \\
\dot{\hat{x}}_n &= -\alpha_n e_1 + \hat{f} + \hat{b}u - k_n \text{sign}(e_1)
\end{aligned} \tag{5.5}$$

The n - 1 poles related with the implied dynamics on sliding area are defined as (Slotine et al., 1986),

$$det(p\mathbf{I}_{n-1} - \begin{bmatrix} -\frac{k_2}{k_1} & 1 & 0 & ... & 0 \\ -\frac{k_3}{k_1} & 0 & 1 & ... & 0 \\ & & & . & \\ & & & . & \\ & & & . & 1 \\ -\frac{k_n}{k_1} & 0 & 0 & ... & 0 \end{bmatrix}) = 0 \tag{5.6}$$

where the $\mathbf{I}_{n-1}$ is the identity matrix of order n - 1. In this way, the poles on sliding area can be located randomly by appropriate choice of the ratios (k_i/k_1), [i = 2,..., n]. A feasible selection is to characterize k_1 as the coveted precision in e_2, by defining

$$k_n \geqslant |\Delta f| \tag{5.7}$$

and in a constant proportion with k_1, and lastly determining the rest of the poles k_i, [i = 2,..., n-1] so that the implied dynamics related with sliding area are critically damped. In conclusion, reducing k_l increases the bandwidth on sliding area, but may potentially reduce sensitivity to measurement noise by reducing the amplitude of the discontinuity in

$\dot{e}_1$.

5.2.2 Effects of measurement noise

Considering a single measurement of second order system corrupted by noise $v = v(t)$, the resulting error dynamics of the method can be defined as (Slotine et al., 1986),

$$\begin{aligned} \dot{e}_1 &= -\alpha_1(e_1 + v) + e_2 - k_1 \text{sign}(e_1 + v) \\ \dot{e}_2 &= -\alpha_2(e_1 + v) + \Delta f - k_2 \text{sign}(e_1 + v) \end{aligned} \tag{5.8}$$

Despite the fact that the appearance of the terms in $\text{sign}(e_1 + v)$ makes a precise stochastic examination genuinely included, helpful knowledge can be acquired by utilizing proper simplifying approximations.

Assume that v is a deterministic signal of bounded range, as can be defined as (Slotine et al., 1986),

$$0 \leqslant \omega < \omega_- \quad \text{or} \quad \omega > \omega_+ \Longrightarrow F_v(\omega) = 0 \tag{5.9}$$

where F_v is the Fourier transform of v. The sliding motion can occur on the surface if the following is fulfilled,

$$e_1 + v = 0 \tag{5.10}$$

Consequently, the sliding area is then characterized by,

$$|e_2 + \dot{v}| \leqslant k_1 \tag{5.11}$$

and the equivalent dynamics are defined by

$$\begin{aligned} e_1 &= -v \\ \dot{e}_2 + \frac{k_2}{k_1} e_2 &= -\frac{k_2}{k_1}\dot{v} + \Delta f \end{aligned} \tag{5.12}$$

Two constraining cases merit specific consideration (Slotine et al., 1986):

1. $\omega_+ << \dfrac{k_2}{k_1}$. If Δf=0, then,

$$e_2 \approx -\dot{v} << \frac{k_2}{k_1} v_{max}$$

Particularly, the estimation of x_2 is precise if the error of measurement in x_1 is constant.

2. $\omega_- >> \dfrac{k_2}{k_1}$. If Δf=0, then,

$$e_2 \approx 0$$

Nevertheless, the bound on k_2/k_x likewise means that the estimator's robustness to model uncertainty is directly restricted by the value of ω_-. The corresponding preciseness in e_2 is expressed as,

$$|e_2| \leqslant \frac{rF}{\omega_-} \tag{5.13}$$

where $r \approx 3$ is the coveted proportion between ω_- and (k_2/k_l), and F is the available bound on Δf. It is acquired by selecting k_1 and k_2 as per

$$\begin{aligned} k_1 &\geqslant |\dot{v}| + \frac{rF}{\omega_-} \\ k_2 &= \frac{k_1 \omega_-}{r} \end{aligned} \tag{5.14}$$

so as to fulfill the condition in equation (5.11) while keeping k_2 greater than or equal to F.

In can be summarized that the system cannot stay in a pristine sliding mode with the appearance of randomly noise of measurement, since this may need utilizing an infinitely large **k**. Instead, expecting that the noise of measurement is limited by some constant v_0, the system will stay in a region of the x_2 axis of width v_0. The significant potential benefit of SMOs, over e.g. an EKF, is that the SMOs can still be made extensively more vigorous to parametric uncertainty. This can be simply comprehended by taking into account the 'average' error of the estimator, $\mathbf{e}_a$ where the error dynamics can be approximated as (Slotine et al., 1986),

$$\begin{aligned}\dot{e}_{1a} &= -\alpha_1 e_{1a} + e_{2a} - k_1 \text{Average}[\text{sign}(e_{1a} + v)] \\ \dot{e}_{2a} &= -\alpha_2 e_{1a} - k_2 \text{Average}[\text{sign}(e_{1a} + v)] + \Delta f_a\end{aligned} \tag{5.15}$$

where Average $[e_{1a} + v]$ is calculated over 'short' time periods during which e_a is dealt with as a constant, and Δf_a is characterized as Average $[\Delta f]$. It is assumed that the state is corrupted by white noise $v(t)$, thus

$$\begin{aligned}\text{Average}[\text{sign}(e_1 + v)] &= \text{Expectation}[\text{sign}(e_1 + v)] \\ &= \int_{-\infty}^{\infty} \text{sign}(e_1 + v) p(v) dv \\ &= 2 \int_0^{e_1} p(v) dv\end{aligned} \tag{5.16}$$

where it is assumed that the probability density $p(v)$ is symmetric. Therefore, the average value is an odd continuous function of e_1. For example, if v is evenly distributed on the interval $[|v_0, v_0]$, then

$$\text{Expectation}[\text{sign}(e_1 + v)] = \frac{e_1}{v_0}$$

so that the average dynamics can be expressed as,

$$\begin{aligned}\dot{e}_{1a} &= -(\alpha_1 + \frac{k_1}{v_0}) e_{1a} + e_{2a} \\ \dot{e}_{2a} &= -(\alpha_2 + \frac{k_2}{v_0}) e_{1a} + \Delta f_a\end{aligned} \tag{5.17}$$

Hence, the role of the switching function is to alter the effective bandwidth of the average dynamics in accordance with the current level of the measurement noise. Particularly, the sliding dynamics of Fillipov's equivalent is recovered as the noise level v_0 tends to zero.

$$\frac{1}{v_0} \to \infty \Longrightarrow e_{1a} \to 0 \Longrightarrow \dot{e}_{1a} \to 0$$

so that

$$\begin{aligned}\dot{e}_{2a} &\longrightarrow -\left[\frac{(\alpha_2 + \frac{k_2}{v_0})}{(\alpha_1 + \frac{k_1}{v_0})}\right] e_{2a} + \Delta f_a \\ &\longrightarrow -\frac{k_1}{k_2} e_{2a} + \Delta f_a\end{aligned} \tag{5.18}$$

A simplified investigation above can be utilized to help the selection of the switching gains k_j. For example, the dynamics of average error of a third-order system is expressed as,

$$\begin{aligned}
\dot{e}_{1a} &= -(\alpha_1 + \frac{k_1}{v_0})e_{1a} + e_{2a} \\
\dot{e}_{2a} &= -(\alpha_2 + \frac{k_2}{v_0})e_{1a} + e_{3a} \\
\dot{e}_{3a} &= -(\alpha_3 + \frac{k_3}{v_0})e_{1a} + \Delta f_a
\end{aligned} \tag{5.19}$$

Uniform bounded white noise of amplitude v_0 is considered, and α_i is selected as in a conventional Kalman filter. It is then sensible to choose the gain k_j so that the average dynamics become critically damped, e.g.:

$$\begin{aligned}
\alpha_1 + \frac{k_1}{v_0} &= 3\lambda \\
\alpha_2 + \frac{k_2}{v_0} &= 3\lambda^2 \\
\alpha_3 + \frac{k_3}{v_0} &= \lambda^3
\end{aligned} \tag{5.20}$$

Furthermore, the minimum satisfactory value of λ is resolved by the condition

$$|e_{1a}| \geqslant v_0 \tag{5.21}$$

which can be expressed as,

$$v_0 \geqslant \frac{F}{\lambda^3} \tag{5.22}$$

where F is a constant upper bound on Δf. The value of λ that generates the smallest k_j is defined as,

$$\lambda = \left(\frac{F}{v_0}\right)^{\frac{1}{3}} \tag{5.23}$$

which represents a sensible selection as long as the corresponding k_j stay positive. Consequently, the bounds on e_{2a} and e_{3a} can be calculated. Specifically, it can be easily analysed in the domain of frequency. The Laplace variable is represented by p, then

$$
\begin{aligned}
e_{2a} &= \left[1 - \frac{3\lambda^2 p + \lambda^3}{(p+\lambda)^3}\right]\frac{\Delta f_a}{p^2} \\
e_{3a} &= \left[1 - \frac{\lambda^3}{(p+\lambda)^3}\right]\frac{\Delta f_a}{p}
\end{aligned}
\tag{5.24}
$$

Therefore, In order to make the observer fully effective, the value of λ must be greater than the frequency value of Δf_a, which, if the equation (5.23) is utilized, imposes in turn an upper limit on the noise level v_0. As an alternative, the condition can be fulfilled by raising λ to a value which is greater than that in equation (5.23), thus also increasing the k_j as well as the noise content of the state estimation.

5.2.3 Noise sensitivity and implementation aspects

In light of the previous averaging investigation, the calculation might be adjusted in order to set the amplitude of the switching terms proportionately to the average values of e_1 over a short time of past event; especially, the gains can be decreased to small value when the average is near to zero, which indicates small values of Δf_a and hence sufficiency of the Kalman filter part alone. A simple approach to do so is to multiply the gain vector **k** by a saturation of smoothing factor $sat(\frac{e_{1f}}{\phi})$, where ϕ is a constant to be selected appropriately as boundary layer width. It should be noted that the multiplication of **k** by the smoothing factor makes the ratios k_j/k_l unchanged.

Furthermore, practically, the ratios k_j/k_l are restricted by the sampling rate, i.e., the bandwidth of the equivalent dynamics on the sliding area becomes smaller than the frequency of sampling by a factor 2 or 3. In addition, in order to effectively interpolate boundary layer, the boundary layer width should be modulated, so that the estimation error time e_1 would be required to cross the boundary layer, is at least two sampling periods, in the free of measurement noise.

5.2.4 Observer structure

A general n^{th}- order nonlinear system can be expressed as,

$$
\dot{\mathbf{x}} = \mathbf{f}(\mathbf{x}, t), \quad \mathbf{x} \in \mathbb{R}^n \tag{5.25}
$$

and a measurements vector which is proportionally connected to the state vector is defined as,

$$\mathbf{z} = \mathbf{C}\mathbf{x}, \quad \mathbf{z} \in \mathbb{R}^n \tag{5.26}$$

Furthermore, the observer structure is defined as (Slotine et al., 1986),

$$\dot{\hat{\mathbf{x}}} = \hat{\mathbf{f}}(\mathbf{x}, t) - \mathbf{H}\tilde{\mathbf{z}} - \mathbf{K}\mathbf{l}_s \tag{5.27}$$

where $\hat{\mathbf{x}} \in \mathbb{R}^n$, $\hat{\mathbf{f}}$ is the estimated model of $\mathbf{f}$, while $\mathbf{H}$ and $\mathbf{K}$ denote $n \times p$ gain matrices to be defined, and $\mathbf{l}_s$ is the $p \times 1$ vector and defined as,

$$\mathbf{l}_s = [\text{sign}(e_{z1}) \quad \text{sign}(e_{z2}) \quad \text{sign}(e_{z3})...\text{sign}(e_{zp})]^T \tag{5.28}$$

where

$$\tilde{z} := \mathbf{C}_i\hat{\mathbf{x}} - z_i \tag{5.29}$$

and $\mathbf{C}_i$, is the ith row of the $p \times n$ $\mathbf{C}$ matrix. Also, error vectors can be defined as,

$$\mathbf{s} := \tilde{\mathbf{z}} = \mathbf{C}(\hat{\mathbf{x}} - \mathbf{x}) \tag{5.30}$$

$$\mathbf{e} := \hat{\mathbf{x}} - \mathbf{x} \tag{5.31}$$

By utilizing equations (4.25) and (4.26), the error dynamics can be expressed as,

$$\dot{\mathbf{e}} = \Delta\mathbf{f} - \mathbf{H}\tilde{\mathbf{z}} - \mathbf{K}\mathbf{l}_\sigma \tag{5.32}$$

where

$$\Delta\mathbf{f} := \hat{\mathbf{f}}(\hat{\mathbf{x}}, t) - \mathbf{f}(\mathbf{x}, t) \tag{5.33}$$

Therefore, equation (4.32) can be expressed as,

$$\dot{\mathbf{e}} = \tilde{\mathbf{f}}, \quad \tilde{\mathbf{f}} = \Delta\mathbf{f} - \mathbf{H}\tilde{\mathbf{z}} - \mathbf{K}\mathbf{l}_\sigma \tag{5.34}$$

The p dimension of sliding surface, and s = 0, will be fascinating if

$$s_i \dot{s}_i < 0, \quad i = 1, ..., p \tag{5.35}$$

Sliding will take place on the sliding surface in a randomly small region of s_i if

$$s_i \dot{s}_i \leqslant -\eta |s_i|, \quad i = 1, ..., p \tag{5.36}$$

In the sliding motion "the system dynamics are reduced effectively from nth order system to a $n - p$ equivalent or reduced-order system" (Slotine et al., 1986).

The approximate dynamics on the reduced-order form can be determined formally by utilizing "equivalent control" approach as described by Utkin (1977), which is similar to Fillipov's method in the special case of linear input switching. In the sliding condition, the switching function in equation (5.29) is working to keep s = 0, and $\dot{s}$ = 0. Therefore, this condition can be expressed as,

$$\text{grad}(s) \bullet \tilde{\mathbf{f}}(\mathbf{e}, \tilde{I}_s) = 0 \tag{5.37}$$

where

$$\tilde{\mathbf{f}} := \Delta \mathbf{f} - \mathbf{H}\tilde{\mathbf{z}} - \mathbf{K}\tilde{\mathbf{I}}_s \tag{5.38}$$

and $\tilde{\mathbf{I}}_s$ is the equivalent sliding vector, which can be acquired from equations (5.30), (5.37), and (5.38), then

$$\mathbf{C}(\Delta \mathbf{f} - \mathbf{H}\tilde{\mathbf{z}} - \mathbf{K}\tilde{\mathbf{I}}_s) = 0 \tag{5.39}$$

so that

$$\tilde{\mathbf{I}}_s = (\mathbf{CK})^{-1}\mathbf{C}\Delta \mathbf{f} \tag{5.40}$$

Hence, the equivalent dynamics in the reduced order form are expressed as,

$$\begin{aligned} \dot{\mathbf{e}} &= (\mathbf{I} - \mathbf{K}(\mathbf{CK})^{-1}\mathbf{C})\Delta \mathbf{f} \\ \mathbf{Ce} &= 0 \end{aligned} \tag{5.41}$$

5.2.5 Observability prerequisites

Consider the general nonlinear system

$$\begin{aligned} \dot{\mathbf{x}} &= \mathbf{f}(\mathbf{x}, t), \quad \mathbf{x} \in \mathbb{R}^n \\ \mathbf{z} &= \mathbf{b}(\mathbf{x}, t), \quad \mathbf{z} \in \mathbb{R}^p \end{aligned} \tag{5.42}$$

Any observer method will succeed in reconstructing the required state **x** from the measurement **z**, if the system is observable. It is not easy to find observability condition of (F) and **z** mathematically in nonlinear systems. Hermann and Krener (1977) investigated the utilization of Lie derivatives to create local conditions. Basically, if the system is observable one must have the capacity to act progressive differential operations on **b(x)** until an inversion is accomplished implicitly to acquire **x**.

For example, given the 2^{nd}-order nonlinear system,

$$\begin{aligned} \dot{x}_1 &= f_1(x_1, x_2) \\ \dot{x}_2 &= f_2(x_1, x_2) \\ z &= x_1 \end{aligned} \tag{5.43}$$

The system f_1 will be observable, if it has a single valued function of x_2. As has been seen in equation (5.37) that Δf_1 must be a function of e_2 so that the control term $-(k_2/k_1)\Delta f_1$ can have an impact on the dynamics of error.

Generally, the observability requirement is connected to equation (5.36) through the $\Delta\mathbf{f}$ vector structure, and an unobservable system will make the dynamics of error uncontrollable.

5.3 Observer design for quadcopter UAVs

Before designing states observer for quadcopter UAVs system, defining states vector of the system is required. The states vector has been defined in equations (3.31) and (3.32), where the measured states are roll (x_1), pitch (x_3) and yaw (x_5) angles, and linear movements in $z(x_7)$, $x(x_9)$, and $y(x_{11})$ axes. SMC as state feedback control system needs the availability

of full state to control the system. Therefore, the observer plays a critical role in estimating the velocity of the rotor-craft, such as x_2, x_4, x_6, x_8, x_{10}, and x_{12}.

Considering the dynamic system of quadcopter in equations (3.47) - (3.49), (3.56) - (3.58), second order observer structure in equation (5.2), the states estimator for quadcopter UAV systems can be expressed as,

$$
\begin{aligned}
\dot{\hat{x}}_1 &= -\alpha_1 e_1 + \hat{x}_2 - K_1 \text{sign}(e_1) \\
\dot{\hat{x}}_2 &= -\alpha_2 e_1 + \hat{x}_4 \hat{x}_6 \left(\tfrac{I_y - I_z}{I_x}\right) + \tfrac{l}{I_x} u_2 - \frac{J_r}{I_{xx}} \hat{x}_4 \Omega_r - K_2 \text{sign}(e_1) \\
\dot{\hat{x}}_3 &= -\alpha_1 e_3 + \hat{x}_4 - K_1 \text{sign}(e_3) \\
\dot{\hat{x}}_4 &= -\alpha_2 e_3 + \hat{x}_2 \hat{x}_6 \left(\tfrac{I_z - I_x}{I_y}\right) + \tfrac{l}{I_y} u_3 + \frac{J_r}{I_{yy}} \hat{x}_2 \Omega_r - K_2 \text{sign}(e_3) \\
\dot{\hat{x}}_5 &= -\alpha_1 e_5 + \hat{x}_6 - K_1 \text{sign}(e_5) \\
\dot{\hat{x}}_6 &= -\alpha_2 e_5 + \hat{x}_2 \hat{x}_4 \left(\tfrac{I_x - I_y}{I_z}\right) + \tfrac{1}{I_z} u_4 - K_2 \text{sign}(e_5) \\
\dot{\hat{x}}_7 &= -\alpha_1 e_7 + \hat{x}_8 - K_1 \text{sign}(e_7) \\
\dot{\hat{x}}_8 &= -\alpha_2 e_7 - g + \frac{cx_1 cx_3}{m} u_1 - K_2 \text{sign}(e_7) \\
\dot{\hat{x}}_9 &= -\alpha_1 e_9 + \hat{x}_{10} - K_1 \text{sign}(e_9) \\
\dot{\hat{x}}_{10} &= -\alpha_2 e_9 + \frac{cx_1 sx_3 cx_5 + sx_1 sx_5}{m} u_1 - K_2 \text{sign}(e_9) \\
\dot{\hat{x}}_{11} &= -\alpha_1 e_{11} + \hat{x}_{12} - K_1 \text{sign}(e_{11}) \\
\dot{\hat{x}}_{12} &= -\alpha_2 e_{11} + \frac{cx_1 sx_3 sx_5 - sx_1 cx_5}{m} u_1 - K_2 \text{sign}(e_{11})
\end{aligned}
\tag{5.44}
$$

where $e = \hat{x} - x$ denotes the estimation error of states, while J_r, Ω_r, $\hat{x}$, α_i and K_i represent rotors' inertia, rotors' relative speed, estimated states, and positive constants respectively.

5.4 Chattering avoidance: elimination and attenuation

Similar to sliding mode control, the use of discontinuous term in sliding mode observer results in chattering phenomenon. Despite the fact that this issue is just connected to numerical implementation instead of "hard" mechanical restriction, the chattering is still harmful for obtaining highly accurate estimated states.

5.4.1 Chattering elimination: quasi-sliding mode observer

The first solution to eliminate the chattering is to employ QuasiSMO by replacing the discontinuous term $K\text{sign}(e)$ with continuous/smooth term which is expressed as (Shtessel et al., 2010),

$$\text{sign}(e) \approx \frac{e}{|e| + \varepsilon} \tag{5.45}$$

where ε is a small positive scalar. Hence, from equation (5.44), the observer can be expressed as,

$$\begin{aligned}
\dot{\hat{x}}_1 &= -\alpha_1 e_1 + \hat{x}_2 - K_1 \tfrac{e_1}{|e_1|+\varepsilon} \\
\dot{\hat{x}}_2 &= -\alpha_2 e_1 + \hat{x}_4 \hat{x}_6 \left(\tfrac{I_y - I_z}{I_x}\right) + \tfrac{l}{I_x} u_2 - \frac{J_r}{I_{xx}} \hat{x}_4 \Omega_r - K_2 \tfrac{e_1}{|e_1|+\varepsilon} \\
\dot{\hat{x}}_3 &= -\alpha_1 e_3 + \hat{x}_4 - K_1 \tfrac{e_3}{|e_3|+\varepsilon} \\
\dot{\hat{x}}_4 &= -\alpha_2 e_3 + \hat{x}_2 \hat{x}_6 \left(\tfrac{I_z - I_x}{I_y}\right) + \tfrac{l}{I_y} u_3 + \frac{J_r}{I_{yy}} \hat{x}_2 \Omega_r - K_2 \tfrac{e_3}{|e_3|+\varepsilon} \\
\dot{\hat{x}}_5 &= -\alpha_1 e_5 + \hat{x}_6 - K_1 \tfrac{e_5}{|e_5|+\varepsilon} \\
\dot{\hat{x}}_6 &= -\alpha_2 e_5 + \hat{x}_2 \hat{x}_4 \left(\tfrac{I_x - I_y}{I_z}\right) + \tfrac{1}{I_z} u_4 - K_2 \tfrac{e_5}{|e_5|+\varepsilon} \\
\dot{\hat{x}}_7 &= -\alpha_1 e_7 + \hat{x}_8 - K_1 \tfrac{e_7}{|e_7|+\varepsilon} \\
\dot{\hat{x}}_8 &= -\alpha_2 e_7 - g + \frac{cx_1 cx_3}{m} u_1 - K_2 \tfrac{e_7}{|e_7|+\varepsilon} \\
\dot{\hat{x}}_9 &= -\alpha_1 e_9 + \hat{x}_{10} - K_1 \tfrac{e_9}{|e_9|+\varepsilon} \\
\dot{\hat{x}}_{10} &= -\alpha_2 e_9 + \frac{cx_1 sx_3 cx_5 + sx_1 sx_5}{m} u_1 - K_2 \tfrac{e_9}{|e_9|+\varepsilon} \\
\dot{\hat{x}}_{11} &= -\alpha_1 e_{11} + \hat{x}_{12} - K_1 \tfrac{e_{11}}{|e_{11}|+\varepsilon} \\
\dot{\hat{x}}_{12} &= -\alpha_2 e_{11} + \frac{cx_1 sx_3 sx_5 - sx_1 cx_5}{m} u_1 - K_2 \tfrac{e_{11}}{|e_{11}|+\varepsilon}
\end{aligned} \tag{5.46}$$

where $e = \hat{x} - x$ denotes the estimation error of states, while J_r, Ω_r, $\hat{x}$, α_i and K_i represent rotors' inertia, rotors' relative speed, estimated states, and positive constants respectively.

5.4.2 Chattering elimination: sliding mode-based interval type-2 fuzzy observer

Similar to that in SMC, in the SMO, IT2FLSs will play a role in replacing discontinuous term of SMO in equation (5.2) to eliminate the chattering phenomenon. Furthermore, membership functions (MFs) of fuzzy input and output should be designed to achieve such goal. Therefore, the MFs of fuzzy input are built with values of estimation errors (e) as shown in Figure 5.1, while the fuzzy output is depicted in Figure 5.2. Each input and output has 7 membership functions, namely: Negative Big (NB), Negative Medium (NM), Negative Small (NS), Zero (Z), Positive Small (PS), Positive Medium (PM), and Positive Big (PB).

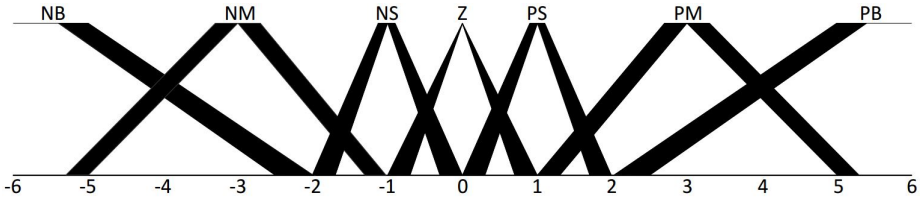

Figure 5.1. Estimation errors (e) MFs

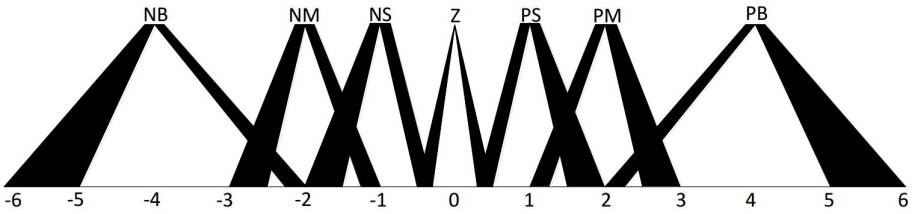

Figure 5.2. Fuzzy output MFs

Furthermore, considering the fuzzy input and output, the rules-base to produce the desired estimated states $(\hat{x})$ is shown in Table 5.1.

Table 5.1. The Rules of IT2FSMO

	e **(Estimation error)**						
	NB	NM	NS	Z	PS	PM	PB
Out$_{fuzzy2}$	NB	NM	NS	Z	PS	PM	PB

Hence, from equation (5.44), the IT2FSMO for the quadcopter UAV model is given as,

$$
\begin{aligned}
\dot{\hat{x}}_1 &= -\alpha_1 e_1 + \hat{x}_2 - K_1 \mathbf{Out}_{fuzzy2}(e_1) \\
\dot{\hat{x}}_2 &= -\alpha_2 e_1 + \hat{x}_4 \hat{x}_6 \left(\tfrac{I_y - I_z}{I_x}\right) + \tfrac{l}{I_x} u_2 - \frac{J_r}{I_{xx}} \hat{x}_4 \Omega_r - K_2 \mathbf{Out}_{fuzzy2}(e_1) \\
\dot{\hat{x}}_3 &= -\alpha_1 e_3 + \hat{x}_4 - K_1 \mathbf{Out}_{fuzzy2}(e_3) \\
\dot{\hat{x}}_4 &= -\alpha_2 e_3 + \hat{x}_2 \hat{x}_6 \left(\tfrac{I_z - I_x}{I_y}\right) + \tfrac{l}{I_y} u_3 + \frac{J_r}{I_{yy}} \hat{x}_2 \Omega_r - K_2 \mathbf{Out}_{fuzzy2}(e_3) \\
\dot{\hat{x}}_5 &= -\alpha_1 e_5 + \hat{x}_6 - K_1 \mathbf{Out}_{fuzzy2}(e_5) \\
\dot{\hat{x}}_6 &= -\alpha_2 e_5 + \hat{x}_2 \hat{x}_4 \left(\tfrac{I_x - I_y}{I_z}\right) + \tfrac{1}{I_z} u_4 - K_2 \mathbf{Out}_{fuzzy2}(e_5) \\
\dot{\hat{x}}_7 &= -\alpha_1 e_7 + \hat{x}_8 - K_1 \mathbf{Out}_{fuzzy2}(e_7) \\
\dot{\hat{x}}_8 &= -\alpha_2 e_7 - g + \frac{cx_1 cx_3}{m} u_1 - K_2 \mathbf{Out}_{fuzzy2}(e_7) \\
\dot{\hat{x}}_9 &= -\alpha_1 e_9 + \hat{x}_{10} - K_1 \mathbf{Out}_{fuzzy2}(e_9) \\
\dot{\hat{x}}_{10} &= -\alpha_2 e_9 + \frac{cx_1 sx_3 cx_5 + sx_1 sx_5}{m} u_1 - K_2 \mathbf{Out}_{fuzzy2}(e_9) \\
\dot{\hat{x}}_{11} &= -\alpha_1 e_{11} + \hat{x}_{12} - K_1 \mathbf{Out}_{fuzzy2}(e_{11}) \\
\dot{\hat{x}}_{12} &= -\alpha_2 e_{11} + \frac{cx_1 sx_3 sx_5 - sx_1 cx_5}{m} u_1 - K_2 \mathbf{Out}_{fuzzy2}(e_{11})
\end{aligned}
\tag{5.47}
$$

where $e = \hat{x} - x$ denotes the estimation error of states, while J_r, Ω_r, $\hat{x}$, α_i and K_i represent rotors' inertia, rotors' relative speed, estimated states, and positive constants respectively. $\mathbf{Out}_{fuzzy2}$ is interval type-2 fuzzy logic output.

5.4.3 Chattering attenuation: super-twisting algorithm of sliding mode observer

Historically, the twisting mode algorithm has features to twist trajectory around the origin of the 2^{nd} order sliding plane. The trajectories demonstrate an unlimited number of rotations while uniting in certain time to the origin as shown in Figure 4.11.

Considering the system as in equation (5.1), the STASMO used for this class of systems has the following structure (Salgado et al., 2011),

$$
\begin{aligned}
\dot{\hat{x}}_1 &= \hat{x}_2 - K_1 \sqrt{|e_1|}\text{sign}(e_1) \\
\dot{\hat{x}}_2 &= f(x,t) + b(x,t)u(t) - K_2 \text{sign}(e_1)
\end{aligned}
\tag{5.48}
$$

In this structure, the gains K_i $(i = 1, 2)$ must be selected in such a way as to ensure finite time convergence.

Hence, from equations (5.44) and (5.48), the STASMO for quadcopter UAVs can be defined as,

$$
\begin{aligned}
\dot{\hat{x}}_1 &= \hat{x}_2 - K_1\sqrt{|e_1|}\text{sign}(e_1) \\
\dot{\hat{x}}_2 &= \hat{x}_4\hat{x}_6\left(\tfrac{I_y - I_z}{I_x}\right) + \tfrac{l}{I_x}u_2 - \frac{J_r}{I_{xx}}\hat{x}_4\Omega_r - K_2\text{sign}(e_1) \\
\dot{\hat{x}}_3 &= \hat{x}_4 - K_1\sqrt{|e_3|}\text{sign}(e_3) \\
\dot{\hat{x}}_4 &= \hat{x}_2\hat{x}_6\left(\tfrac{I_z - I_x}{I_y}\right) + \tfrac{l}{I_y}u_3 + \frac{J_r}{I_{yy}}\hat{x}_2\Omega_r - K_2\text{sign}(e_3) \\
\dot{\hat{x}}_5 &= \hat{x}_6 - K_1\sqrt{|e_5|}\text{sign}(e_5) \\
\dot{\hat{x}}_6 &= \hat{x}_2\hat{x}_4\left(\tfrac{I_x - I_y}{I_z}\right) + \tfrac{1}{I_z}u_4 - K_2\text{sign}(e_5) \\
\dot{\hat{x}}_7 &= \hat{x}_8 - K_1\sqrt{|e_7|}\text{sign}(e_7) \\
\dot{\hat{x}}_8 &= -g + \frac{cx_1cx_3}{m}u_1 - K_2\text{sign}(e_7) \\
\dot{\hat{x}}_9 &= \hat{x}_{10} - K_1\sqrt{|e_9|}\text{sign}(e_9) \\
\dot{\hat{x}}_{10} &= \frac{cx_1sx_3cx_5 + sx_1sx_5}{m}u_1 - K_2\text{sign}(e_9) \\
\dot{\hat{x}}_{11} &= \hat{x}_{12} - K_1\sqrt{|e_{11}|}\text{sign}(e_{11}) \\
\dot{\hat{x}}_{12} &= \frac{cx_1sx_3sx_5 - sx_1cx_5}{m}u_1 - K_2\text{sign}(e_{11})
\end{aligned}
\tag{5.49}
$$

where $e = \hat{x} - x$ denotes the estimation error of states, while J_r, Ω_r, $\hat{x}$, and α_i represent rotors' inertia, rotors' relative speed, estimated states, and positive constants respectively.

5.4.4 Chattering attenuation: higher order sliding mode observer

Second-order sliding modes cause the switching variables to converge to zero in finite time, when the relative degree of the variable equals two. In addition, the sliding modes can solve the chattering issue by employing continuous term if the relative degree is one. This may help to cancel harmful high-energy oscillation. HOSMO can handle the problem for arbitrary relative degrees. The implementation of the approach needs more information: usually it needs to compute a number of sequential time derivatives of the switching

variables.

The general form of HOSMO is given as (Gonzalez-Hernandez et al., 2017),

$$\begin{aligned}\dot{\hat{x}}_1 &= \hat{x}_2 + K_1|e_1|^{\frac{2}{3}}\text{sign}(e_1)\\ \dot{\hat{x}}_2 &= \hat{x}_3 + K_2|e_1|^{\frac{1}{3}}\text{sign}(e_1) + f(x,t) + b(x,t)u(t)\\ \dot{\hat{x}}_3 &= K_3\text{sign}(e_1)\end{aligned} \tag{5.50}$$

The dynamics of the estimated states errors are defined as,

$$\begin{aligned}\dot{e}_1 &= e_2 - K_1|e_1|^{\frac{2}{3}}\text{sign}(e_1)\\ \dot{e}_2 &= -\hat{x}_3 - K_2|e_1|^{\frac{1}{3}}\text{sign}(e_1) + \zeta\\ \dot{e}_3 &= K_3|e_1|^{\frac{2}{3}}\text{sign}(e_1)\end{aligned} \tag{5.51}$$

Define a new variable of estimation error variable $e_3 = \zeta - \hat{x}_3$, if the rate of the disturbance is restricted $|\dot{\zeta}| < \zeta^+$, then the dynamics of the estimated states errors can be expressed as,

$$\begin{aligned}\dot{e}_1 &= e_2 - K_1|e_1|^{\frac{2}{3}}\text{sign}(e_1)\\ \dot{e}_2 &= e_3 - K_2|e_1|^{\frac{1}{3}}\text{sign}(e_1)\\ \dot{e}_3 &= -K_3|e_1|^{\frac{2}{3}}\text{sign}(e_1) + \dot{\zeta}\end{aligned} \tag{5.52}$$

In this structure, the gains K_i ($i = 1, 2, 3$) must be selected in such a way as to ensure finite time convergence.

Before deriving HOSMO for quadcopter UAVs, it is important to redefine the states vector of such vehicle. The new states information is defined as,

$$\begin{aligned}&x_1 = \phi; \quad x_4 = \theta; \quad x_7 = \psi; \quad x_{10} = z; \quad x_{13} = x; \quad x_{16} = y\\ &x_2 = \dot{\phi}; \quad x_5 = \dot{\theta}; \quad x_8 = \dot{\psi}; \quad x_{11} = \dot{z}; \quad x_{14} = \dot{x}; \quad x_{17} = \dot{y}\end{aligned}$$

Referring to equations (5.44) and (5.50), the states estimation for quadcopter UAVs can be obtained as,

$$
\begin{aligned}
\dot{\hat{x}}_1 &= \hat{x}_2 + K_1|e_1|^{\frac{2}{3}}\text{sign}(e_1) \\
\dot{\hat{x}}_2 &= \hat{x}_3 + \hat{x}_5\hat{x}_8\left(\tfrac{I_y - I_z}{I_x}\right) + \tfrac{l}{I_x}u_2 - \frac{J_r}{I_{xx}}\hat{x}_5\Omega_r + K_2|e_1|^{\frac{1}{3}}\text{sign}(e_1) \\
\dot{\hat{x}}_3 &= K_3\text{sign}(e_1) \\
\dot{\hat{x}}_4 &= \hat{x}_5 + K_1|e_4|^{\frac{2}{3}}\text{sign}(e_4) \\
\dot{\hat{x}}_5 &= \hat{x}_6 + \hat{x}_2\hat{x}_8\left(\tfrac{I_z - I_x}{I_y}\right) + \tfrac{l}{I_y}u_3 + \frac{J_r}{I_{yy}}\hat{x}_2\Omega_r + K_2|e_4|^{\frac{1}{3}}\text{sign}(e_4) \\
\dot{\hat{x}}_6 &= K_3\text{sign}(e4) \\
\dot{\hat{x}}_7 &= \hat{x}_8 + K_1|e_7|^{\frac{2}{3}}\text{sign}(e_7) \\
\dot{\hat{x}}_8 &= \hat{x}_9 + \hat{x}_2\hat{x}_5\left(\tfrac{I_x - I_y}{I_z}\right) + \tfrac{1}{I_z}u_4 + K_2|e_7|^{\frac{1}{3}}\text{sign}(e_7) \\
\dot{\hat{x}}_9 &= K_3\text{sign}(e_7) \\
\dot{\hat{x}}_{10} &= \hat{x}_{11} + K_1|e_{10}|^{\frac{2}{3}}\text{sign}(e_{10}) \\
\dot{\hat{x}}_{11} &= \hat{x}_{12} - g + \frac{cx_1cx_4}{m}u_1 + K_2|e_{10}|^{\frac{1}{3}}\text{sign}(e_{10}) \\
\dot{\hat{x}}_{12} &= K_3\text{sign}(e_{10}) \\
\dot{\hat{x}}_{13} &= \hat{x}_{14} + K_1|e_{13}|^{\frac{2}{3}}\text{sign}(e_{13}) \\
\dot{\hat{x}}_{14} &= \hat{x}_{15} + \frac{cx_1sx_4cx_7 + sx_1sx_7}{m}u_1 + K_2|e_{13}|^{\frac{1}{3}}\text{sign}(e_{13}) \\
\dot{\hat{x}}_{15} &= K_3\text{sign}(e_{13}) \\
\dot{\hat{x}}_{16} &= \hat{x}_{17} + K_1 sign|e_{16}|^{\frac{2}{3}}\text{sign}(e_{16}) \\
\dot{\hat{x}}_{17} &= \hat{x}_{18} + \frac{cx_1sx_4sx_7 - sx_1cx_7}{m}u_1 + K_2|e_{16}|^{\frac{1}{3}}\text{sign}(e_{16}) \\
\dot{\hat{x}}_{18} &= K_3\text{sign}(e_{16})
\end{aligned}
\tag{5.53}
$$

where $e = \hat{x} - x$ denotes the estimation error of states, while J_r, Ω_r, $\hat{x}$, and α_i represent rotors' inertia, rotors' relative speed, estimated states, and positive constants respectively.

5.5 Summary

In this chapter, four methods of sliding mode observers (SMOs) including quasi-sliding mode observer (QuasiSMO), sliding mode-based interval type-2 fuzzy observer (IT2FSMO), super-twisting algorithm of sliding mode observer (STASMO), and high order sliding

mode observer (HOSMO) have been designed to estimate unmeasured states of quadcopter UAV system. The main idea of using those methods is that to resolve chattering issue in SMO which may lead to performance degradation of the observer. The overall performance of the methods will be evaluated and compared with extended Kalman filter (EKF) through numerical simulation and experimental validation in the following chapter.

Chapter 6

Numerical simulations: set-point integral quasi-sliding mode control

6.1 Introduction

This chapter presents numerical simulation results to highlight the performance of set-point integral quasi-sliding mode control (QuasiSMC) with the developed nonlinear full-order state observers, namely:

- quasi - sliding mode observer
- sliding mode-based interval type-2 fuzzy observer
- super-twisting algorithm of sliding mode observer
- higher order sliding mode observer
- extended Kalman filter

Performance comparison between sliding mode-based observers and extended Kalman filter (EKF) is conducted to highlight the robustness of each method. Through numerical validation, the performance of the control system and the performance comparisons of the observer methods are carried out and presented in this chapter.

The overall control and observer system is shown in Figure 6.1. In this work, sliding mode control is employed to control the altitude and attitude of the quadcopter. The aim of such controller is to force the system dynamics to track desired trajectories, whereas SMO or EKF is utilized to estimate unmeasured states of the vehicle that are required for the control process.

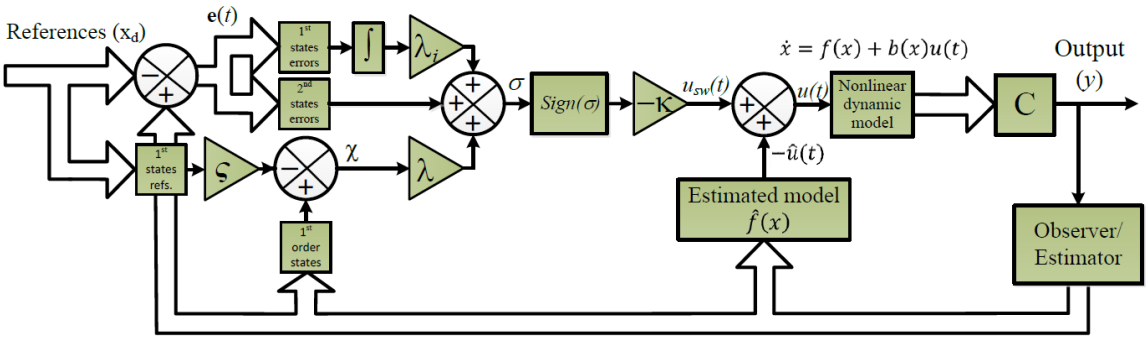

Figure 6.1. Block diagram of the overall control and observer system

6.2 Simulation setup

In this section, numerical simulations are done in MATLAB R2014b run in a personal computer (PC) specification as follows: Intel Core i5-2500 CPU @3.30GHz (4 CPUs), RAM 8MB.

The moment of inertia (Turnigy H.A.L frame), mass, and length of quadcopter arms are real parameters obtained by measurement. All quadcopter parameters used in the simulation are shown in Table 6.1, meanwhile the parameters of observers can be seen in Table 6.2. Selected weighting factors (ς) are 0.01 for x and y states, and 0.8 for others. However, for dynamic sliding mode control, the weighting factors used are 0.9 for x and y states, 0.688 for others. The simulation run with sampling (t_s) time of 0.01 seconds. In addition to those parameters, white Gaussian noise disturbance to measured states are selected to test the performance of the methods as well as parameters mismatch. The parameters mismatch is restricted to moment of inertia of the UAV. The signal to noise ratio (SNR) introduced to measured states are 10 dB for roll, pitch, and yaw angle, and 37 dB for x, y, and z movement, while parameters mismatch to be tested is 50%. In addition, the highest parameter mismatch will be sought to determine the robustness limits of each controller and observer.

Table 6.1. Quadcopter parameters

Variables	Values		
	Observer model	Plant	Unit
m	1.79	1.79	kg
l	0.29	0.29	m
I_x	0.026281674	0.026281674	kgm^2
I_y	0.027484487	0.027484487	kgm^2
I_z	0.045603074	0.045603074	kgm^2
$k_i (i = 1, 2, 3)$	0.37	0.37	Ns/m
$k_j (j = 4, 5, 6)$	0.0005	0.0005	Ns/m
g	9.81	9.81	m/s^2

Table 6.2. Observer parameters

Observer									
QuasiSMO		IT2FSMO		STASMO		HOSMO		EKF	
α_1	85.8	α_1	85.8	K_1	1.0	K_1	6.0	R	0.001
α_2	7.2	α_2	7.2	K_2	1.1	K_2	11.0	Q	0.1
K_1	0.1	K_1	0.1			K_3	6.0		
K_2	2.0	K_2	2.0						

There are four conditions to test the performance of the control and observer methods, including:

- without noise and no parameter mismatch
- without noise and with parameter mismatch
- with noise and no parameter mismatch
- with noise and with parameter mismatch

The purpose of this approach is to determine the ability of the control system and observer methods to deal with several conditions.

The reference frame of the quadcopter movements in this numerical simulation can be seen in Figure 6.2

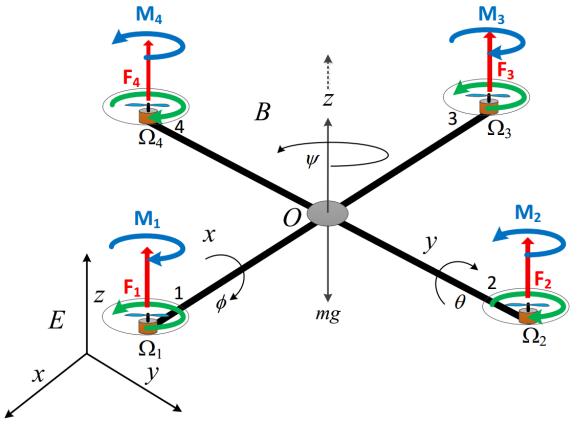

Figure 6.2. Reference frame of quadcopter movements

where x, y, and z represent the movements of the quadcopter in those axes, while ϕ, θ, and ψ represent roll, pitch, and yaw motions of the quadcopter respectively.

6.3 Performance criteria

A set of simulation exercises is presented to examine and demonstrate the performance of sliding mode-based nonlinear control systems with nonlinear full-order state observer approaches in controlling the quadcopter and estimating the unmeasured states of such vehicle. Comparative assessment of such controllers and estimators is presented. Maximum overshoot, rise time, chattering issue, steady-state error, and control loop time are taken into account to evaluate controller performance relating to the use of several different observers, while chattering phenomenon, estimation errors and time required to track true states are the factors evaluated to facilitate comparative studies of observers. Disturbances and uncertainties such as e.g. air drag, white Gaussian noise, and parameters mismatch are also incorporated into the simulations to demonstrate the robustness of the system. The combination of several nonlinear observers with selected sliding mode control method is applied to show the behaviour of each observer affecting the overall control performance.

6.4 Simulation results

After a series of investigations, the following sliding mode parameter were found suitable for the desired performance achievement,

$$\begin{array}{cccccc} \lambda_\phi = 16.5 & \lambda_\theta = 16.5 & \lambda_\psi = 3.2935 & \lambda_x = 2.3 & \lambda_y = 2.3 & \lambda_z = 3.3950 \\ \lambda_{i\phi} = 30.0 & \lambda_{i\theta} = 30.0 & \lambda_{i\psi} = 1.4 & \lambda_{ix} = 1.82 & \lambda_{iy} = 1.82 & \lambda_{iz} = 1.45 \\ k_\phi = 10.0 & k_\theta = 10.0 & k_\psi = 5.0 & k_x = 2.0 & k_y = 2.0 & k_z = 5.0 \end{array}$$

6.4.1 Without noise and no parameter mismatch

Noise-free disturbance and no uncertainty are the ideal conditions used to assess the behaviour of selected controller and observers. The simulation results obtained are shown in

Figures 6.3 – 6.24.

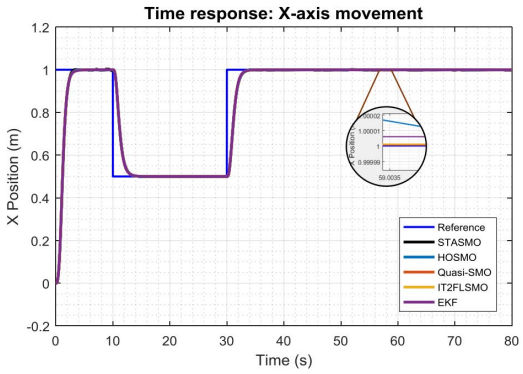

Figure 6.3. x-axis movement

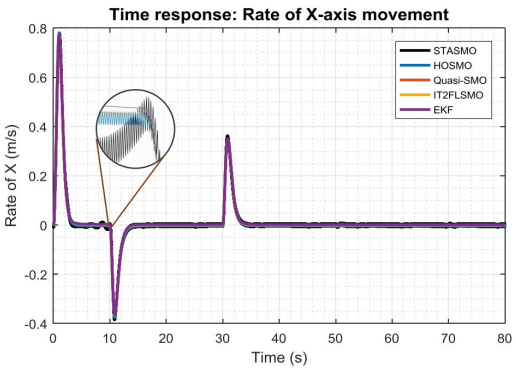

Figure 6.4. Rate of x-axis movement

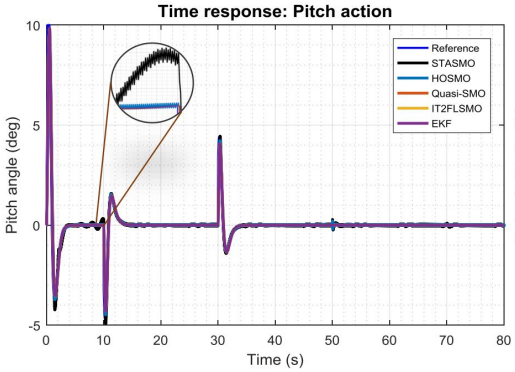

Figure 6.5. Pitch action

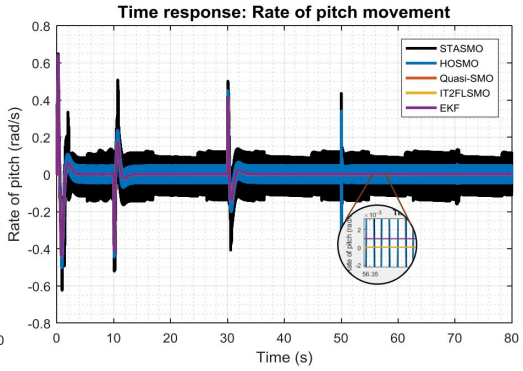

Figure 6.6. Rate of pitch movement

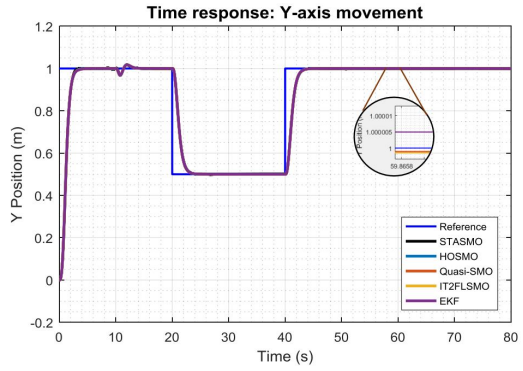

Figure 6.7. y-axis movement

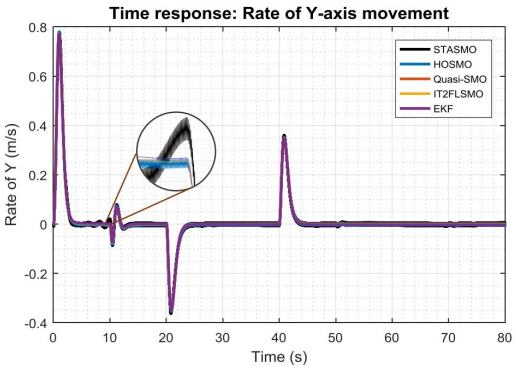

Figure 6.8. Rate of y-axis movement

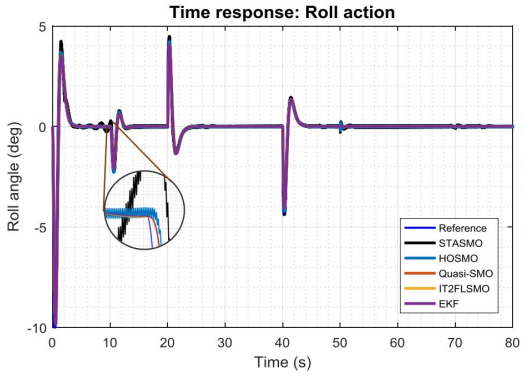

Figure 6.9. Roll action

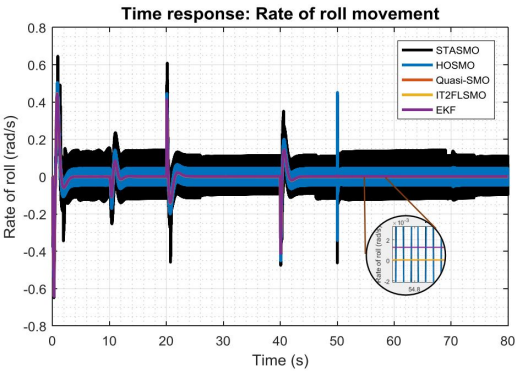

Figure 6.10. Rate of roll movement

Figure 6.11. z-axis movement

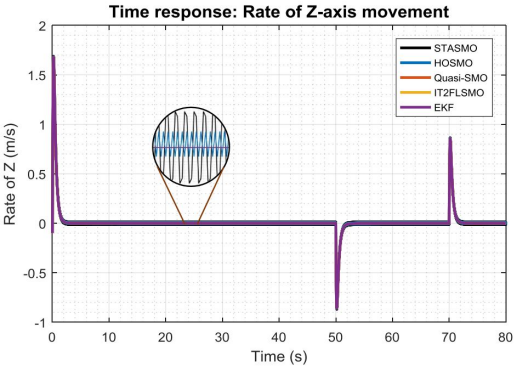

Figure 6.12. Rate of z-axis movement

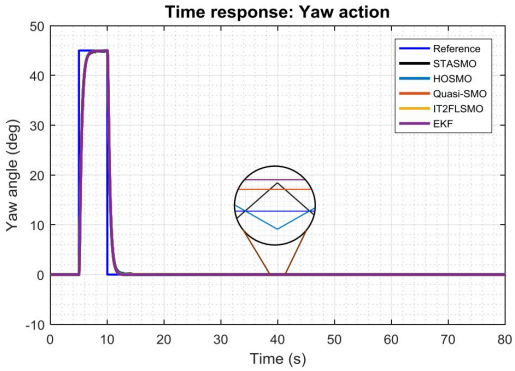

Figure 6.13. Yaw action

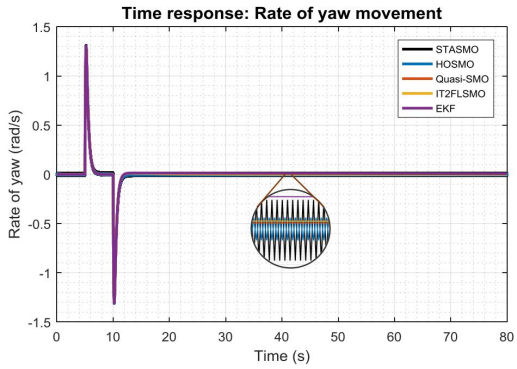

Figure 6.14. Rate of yaw movement

Generally, in terms of observer performance, all observers worked in estimating unmeasured states of the quadcopter UAV as can be seen in Figures 6.4, 6.6, 6.8, 6.10, 6.12, and 6.14. However, high frequency oscillations still appeared clearly on the use of STASMO and HOSMO approaches especially in estimating second states of pitch and roll motions as shown in Figures 6.6 and 6.10 with significant amplitude of oscillation. Although the chattering issue was reduced significantly with these methods, switching function utilized in both observer is still the main cause of the oscillation of the estimated states.

Furthermore, the mean squared error (MSE) calculation method was used to measure the quality of observers, as presented in Table 6.3. It is noted that, QuasiSMO had the smallest MSE of all estimated states. In other words, QuasiSMO outperformed other observer approaches in this case.

Table 6.3. Mean squared errors of estimated states

Observer methods	MSE estimated state of					
	x rate	pitch rate	y rate	roll rate	yaw rate	z rate
QuasiSMO	2.76081E-10	7.7897E-10	1.26399E-10	6.34289E-10	1.5927E-06	1.25385E-09
IT2FSMO	7.81835E-10	2.39267E-09	3.48071E-10	1.47978E-09	3.09E-06	1.27202E-09
STASMO	3.18186E-05	0.01090956	3.23353E-05	0.012470166	0.000141577	0.000139121
HOSMO	6.52389E-06	0.002133593	6.54453E-06	0.002166132	2.10799E-05	2.11742E-05
EKF	2.15319E-06	9.23902E-07	2.85031E-06	1.48113E-06	9.38941E-05	1.87837E-09

Referring to Figures 6.15 - 6.20, it is noted that each observer showed different performance in tracking true states. It can be seen that QuasiSMO and IT2FSMO showed small estimation errors of true states in steady state condition, whereas EKF performed better in true states change condition by showing smaller estimation error of true states. Furthermore, MSE was calculated as presented in Table 6.4. In the condition of without noise and no parameters mismatch, the two observers including QuasiSMO and IT2FSMO outperformed other observer methods by showing smaller MSE for almost all states.

Table 6.4. Mean squared errors of true states tracking

Observer method	MSE true state tracking of					
	x	pitch	y	roll	yaw	z
QuasiSMO	3.92201E-10	1.20594E-11	3.91856E-10	5.30348E-12	1.54953E-09	3.90873E-12
IT2FSMO	4.00058E-10	1.591E-11	3.97554E-10	6.29034E-12	1.922E-09	3.95141E-12
STASMO	5.09916E-09	6.39323E-09	5.18376E-09	6.47623E-09	5.66181E-09	9.1892E-09
HOSMO	3.00745E-09	2.97663E-09	3.01046E-09	2.97566E-09	2.97356E-09	2.97574E-09
EKF	3.72463E-10	2.14412E-09	2.60373E-10	2.20845E-09	1.28466E-08	5.1922E-09

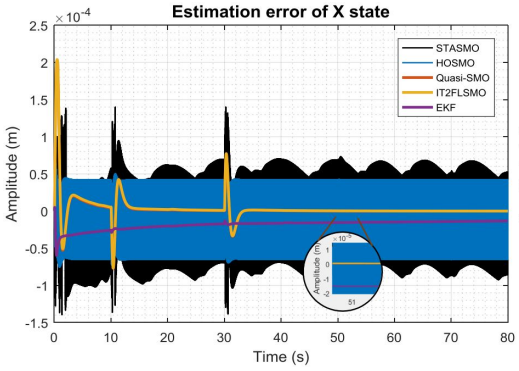

Figure 6.15. Estimation error of x

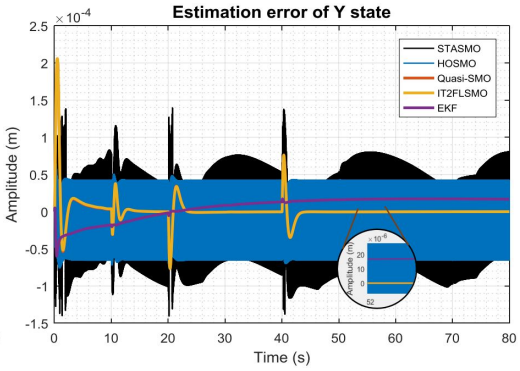

Figure 6.16. Estimation error of y

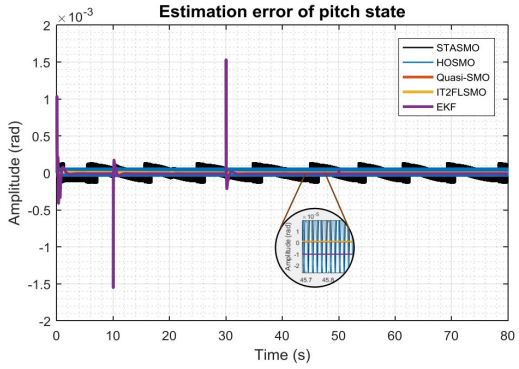

Figure 6.17. Estimation error of pitch

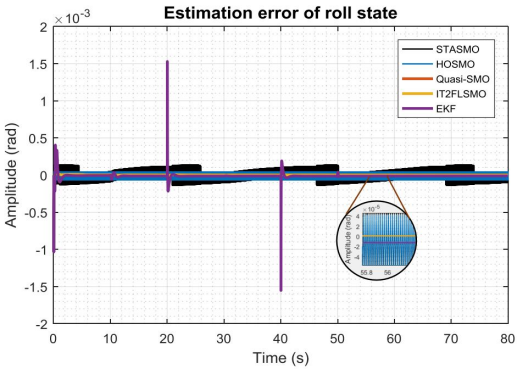

Figure 6.18. Estimation error of roll

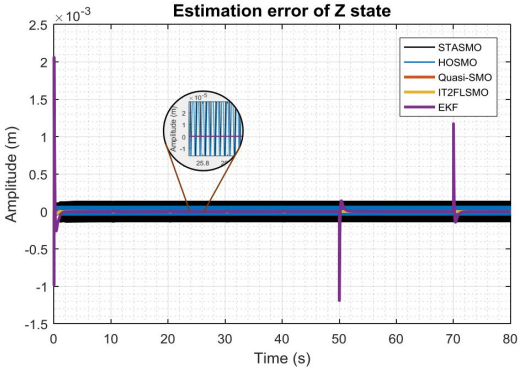

Figure 6.19. Estimation error of z

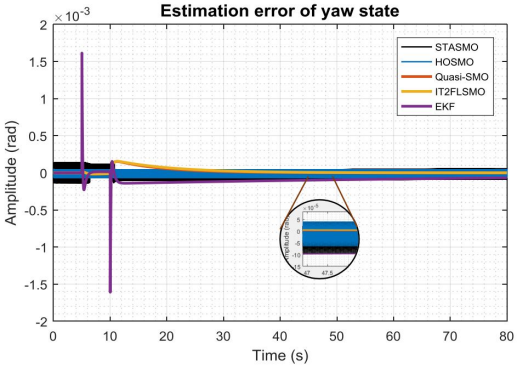

Figure 6.20. Estimation error of yaw

In addition to estimating unmeasured states and measuring estimation errors of true states, the performance of observers was evaluated by time required to estimate unknown states and to track true states since this parameter plays a critical role in real-time applications. It was noted in chapter 5 that each observer has different numerical methods to observe the unknown states. These differences will result in varying amount of time required of estimation process as presented in Table 6.5.

Table 6.5. Time required to estimate unmeasured states

	QuasiSMO	**IT2FSMO**	**STASMO**	**HOSMO**	**EKF**
Time required (ms)	0.04125	4.02375	0.05925	0.0755	0.1425

Although IT2FSMO had good performance in estimating unmeasured states and tracking true states, this method needed more time to track output states and estimate second order of states than other methods. Type reduction method in type-2 fuzzy logic is one of the processes consuming excessive time. Type reduction is still fascinating topic to be researched in type-2 fuzzy logic field so that it can be used in wide range real-time applications. Among the observers, QuasiSMO was the fastest estimator followed by STASMO,

HOSMO and EKF. Therefore, QuasiSMO will be a promising observer method to implement in real-time application.

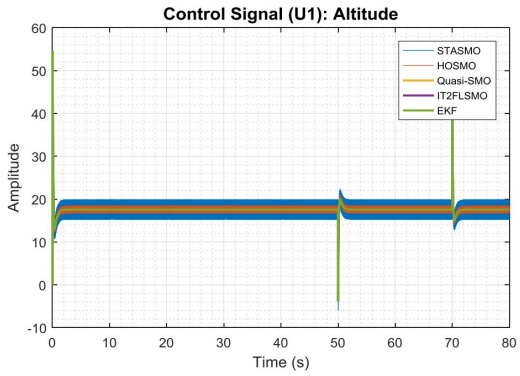

Figure 6.21. Control signal u_1

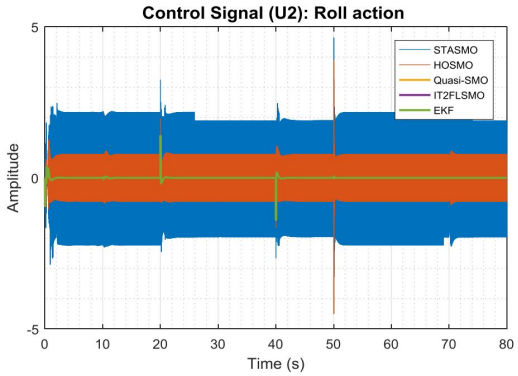

Figure 6.22. Control signal u_2

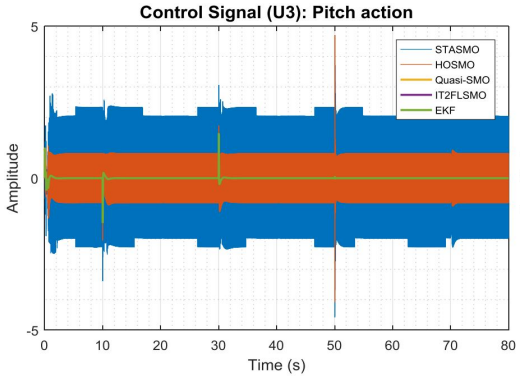

Figure 6.23. Control signal u_3

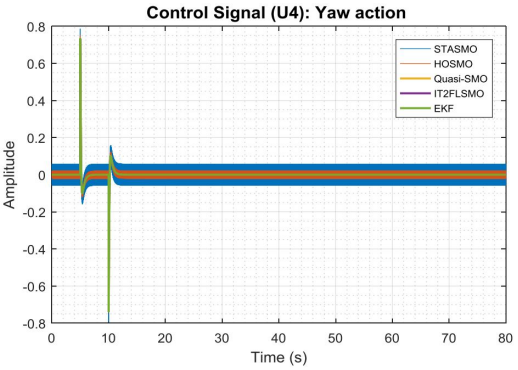

Figure 6.24. Control signal u_4

On the grounds of control as noted, the issues following each observer including small chattering, and time consumption were not major problems as the selected controller could control altitude and attitude of quadcopter UAVs adequately. It can be proved by showing good performance of the controller in tracking references as presented in Figure 6.3 - 6.14. The time demand to process the sequence of controller method was reasonable short around 0.59455 ms. However, not all observers led the controller to the same performance.

The QuasiSMC showed good performance in controlling the copter with various observers. It can be seen in the figures that there was no overshoot step response of the system, and same rise time was achieved with all observers methods except STASMO and HOSMO with which the response was slightly faster. However, high frequency oscillations were seen in the control inputs following the use of STASMO and HOSMO as can be seen in Figures 6.21 - 6.24, whereas QuasiSMO, IT2FSMO and EKF gave smoother control signals for the aircraft.

Another factor that needs to be considered in evaluating the selected controller performance which relates to the use of several observers is steady-state errors. Table 6.6 shows MSE of output states of quadcopter UAV in steady-state condition. Generally, QuasiSMO and IT2FSMO outperformed other observers in improving the performance of selected controller by showing smaller MSE in all output states except yaw for which EKF performed better than others.

Table 6.6. Mean squared errors of steady-states errors

Observer methods	**MSE steady-state error of**					
	x	**pitch**	**y**	**roll**	**yaw**	**z**
QuasiSMO	5.67918E-12	1.00446E-13	5.50607E-13	4.08506E-13	3.70166E-08	2.12579E-10
IT2FSMO	7.9161E-12	3.91245E-13	9.65327E-13	5.31172E-13	4.0398E-08	2.12336E-10
STASMO	1.13841E-07	3.94927E-07	1.61175E-07	4.65012E-07	1.03704E-08	2.10011E-08
HOSMO	1.3982E-08	1.01132E-07	1.76781E-08	1.10949E-07	4.19838E-09	4.33663E-09
EKF	7.29825E-11	1.31074E-10	7.33541E-11	2.13591E-10	1.0394E-08	2.41632E-10

In summary, in the case of free of noise and no parameters mismatch, QuasiSMC shows good performance in controlling the dynamics of quadcopter UAVs with no response overshoot, small rise time, free from chattering when using QuasiSMO, IT2FSMO and EKF observer method, small steady-states errors, and short computational time. As for the observers, QuasiSMO method outperforms other methods by showing smooth estimated states, smaller states estimation errors, and faster in tracking true states. Therefore, the observer led to improved overall performance of selected controller.

6.4.2 Without noise and with parameter mismatch

In this case, parameter uncertainty is incorporated into the system to assess the performances of the control and observer method in dealing with this condition. The performances of the selected control system and observers are presented in Figure 6.25 – 6.46.

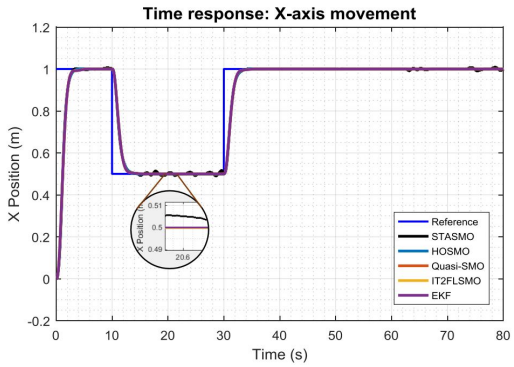

Figure 6.25. x-axis movement

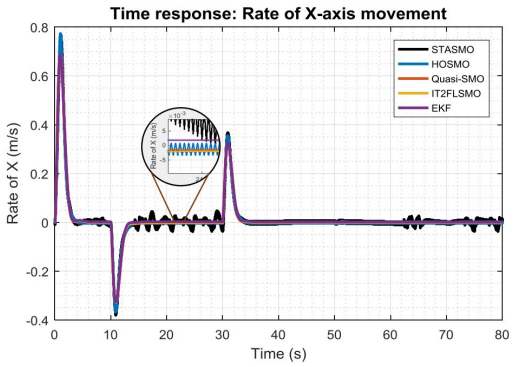

Figure 6.26. Rate of x-axis movement

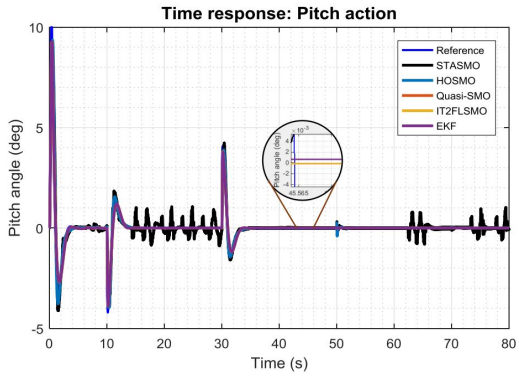

Figure 6.27. Pitch action

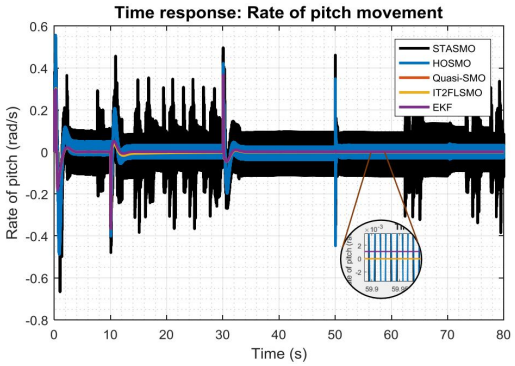

Figure 6.28. Rate of pitch movement

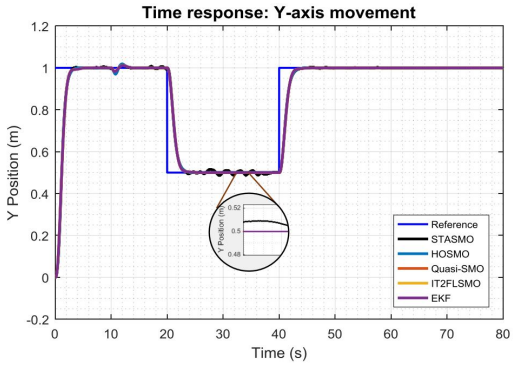

Figure 6.29. y-axis movement

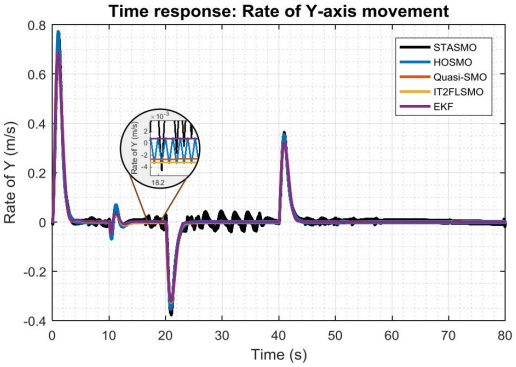

Figure 6.30. Rate of y-axis movement

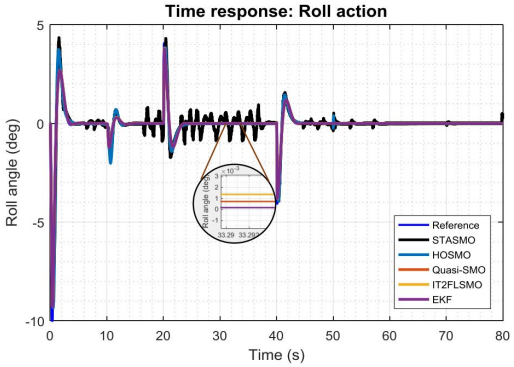

Figure 6.31. Roll action

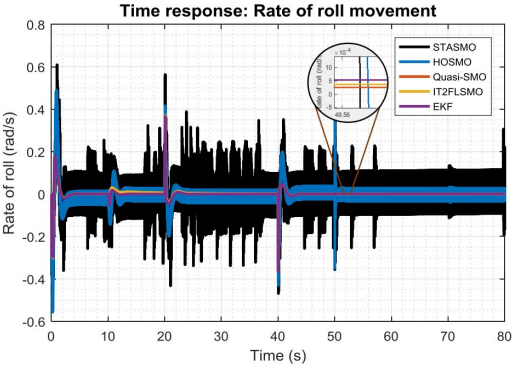

Figure 6.32. Rate of roll movement

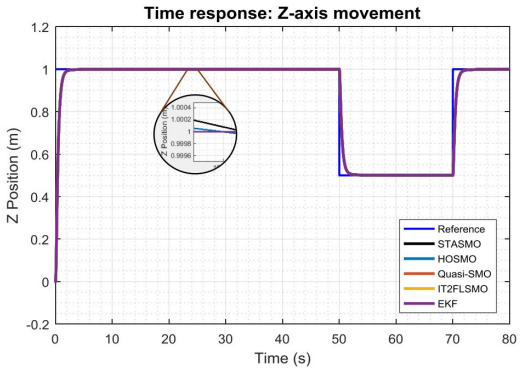

Figure 6.33. z-axis movement

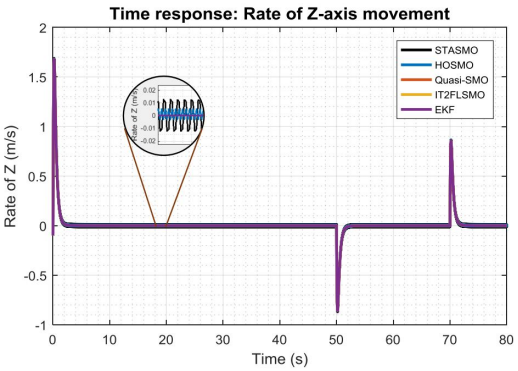

Figure 6.34. Rate of z-axis movement

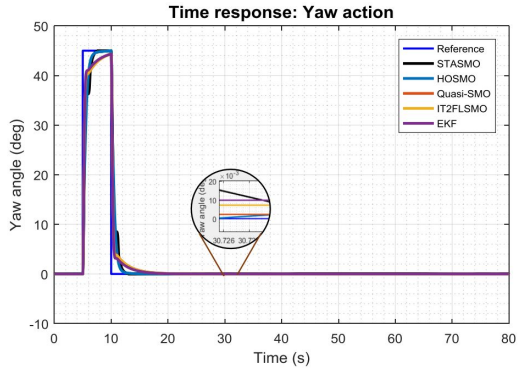

Figure 6.35. Yaw action

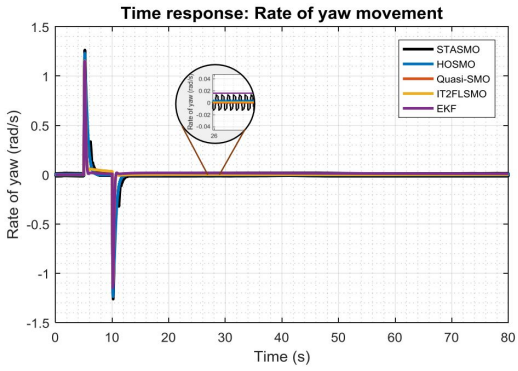

Figure 6.36. Rate of yaw movement

It is noted in the simulation results presented in Figures 6.26, 6.28, 6.30, 6.32, 6.34, 6.36, that the performance of STASMO in estimating some unmeasured states especially rate of pitch and roll has been affected as shown in Figures 6.28 and 6.32. In addition to high frequency oscillation caused by switching function, the amplitude of estimated states fluctuated randomly. Furthermore, the estimating process of x and y rate states were influenced by pitch and roll rate, because those states are coupled. From the results it can be seen that QuasiSMO, IT2FSMO, HOSMO, and EKF methods are more robust than STASMO in dealing with parameters mismatch by showing similar performance as the first case of no parameters mismatch.

For further evaluation, MSE of estimated states as presented in Table 6.7 was analysed to determine which estimator performed well in estimating unmeasured states. From the table it can be seen clearly that QuasiSMO still showed good performance by displaying smaller MSE for all unmeasured states.

Table 6.7. Mean squared errors of estimated states

Observer methods	MSE estimated state of					
	x rate	pitch rate	y rate	roll rate	yaw rate	z rate
QuasiSMO	2.91382E-09	9.00771E-09	2.84873E-09	9.55774E-09	1.72393E-08	1.50343E-10
IT2FSMO	8.20677E-09	2.38031E-08	7.7586E-09	2.40787E-08	5.92642E-07	1.6753E-10
STASMO	6.50345E-05	0.01016915	4.09194E-05	0.009117207	0.000141224	0.00013918
HOSMO	6.58723E-06	0.001036607	6.61224E-06	0.001059928	1.58288E-05	2.11738E-05
EKF	2.49688E-06	1.09824E-06	3.99952E-07	2.50842E-07	0.000168438	5.72571E-10

The Figures 6.37 - 6.42 show the different responses of observers methods in tracking measured states. Generally, all approaches worked well in tracking true states by showing small estimation errors for all measured states. However, some observers including QuasiSMO, IT2FSMO, and EKF performed better than STASMO and HOSMO. Fluctuated errors were noticeable clearly following the use of STASMO and HOSMO methods. Furthermore, similar to the previous case of no parameters mismatch, it can be seen that QuasiSMO and IT2FSMO performed better in steady-state condition by showing smaller estimation errors than EKF, whereas performance of EKF was better than those of other two in dynamic condition. The overall performance of tracking true states computed by MSE is presented in Table 6.8.

Table 6.8. Mean squared errors of true states tracking

Observer method	MSE true state tracking of					
	x	pitch	y	roll	yaw	z
QuasiSMO	1.42879E-10	9.52222E-08	2.39742E-10	9.54722E-08	9.8084E-07	7.61766E-13
IT2FSMO	1.85256E-10	9.57316E-08	3.35342E-10	9.61861E-08	9.87966E-07	7.73805E-13
STASMO	6.16589E-09	2.24582E-06	6.59567E-09	1.86926E-06	0.000215571	9.21118E-09
HOSMO	2.9991E-09	2.18417E-07	2.99873E-09	2.10726E-07	1.67686E-06	2.97573E-09
EKF	3.9264E-10	7.36446E-08	1.49266E-10	7.38305E-08	7.81748E-07	5.22319E-09

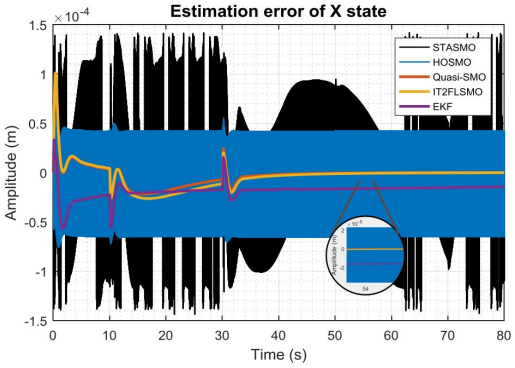

Figure 6.37. Estimation error of x

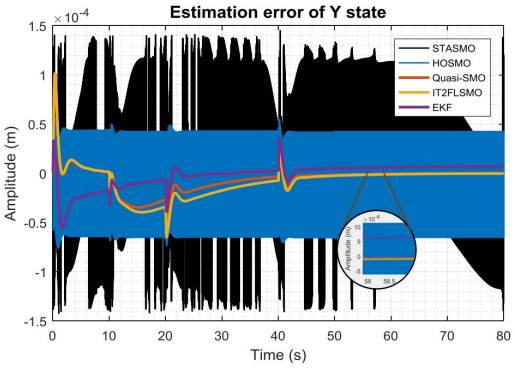

Figure 6.38. Estimation error of y

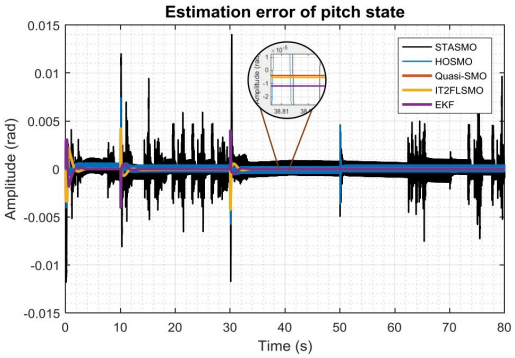

Figure 6.39. Estimation error of pitch

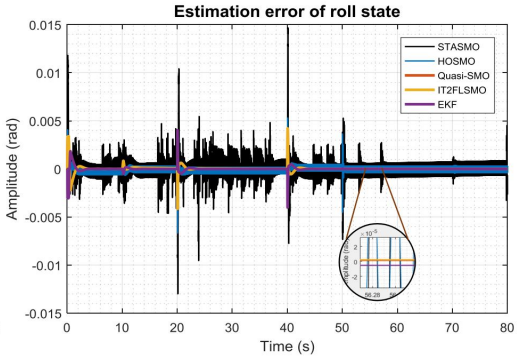

Figure 6.40. Estimation error of roll

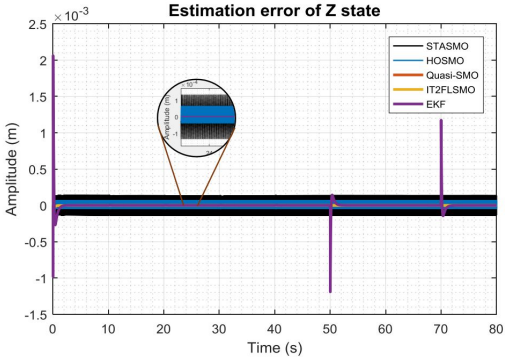

Figure 6.41. Estimation error of z

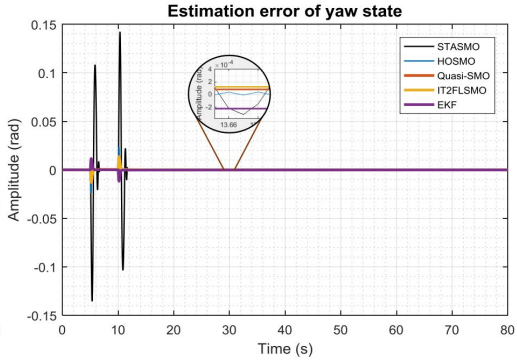

Figure 6.42. Estimation error of yaw

On the grounds of control, it is noted that parameter uncertainty influenced significantly the control performance in managing roll, pitch, and yaw as seen in Figures 6.27, 6.31, and 6.35 when STASMO is employed. In steady-state condition of roll and pitch motions, the states fluctuated around $\pm 0.8^{\circ}$ and $\pm 1^{\circ}$ respectively. Since pitch and x, and roll and y states are coupled, so tracking references of x and y were affected as seen in Figures 6.25 and 6.29. Small fluctuation around $\pm 2cm$ happened in x and y movement when the references changed.

In addition, different behaviours were noted in yaw motion response as seen in Figure 6.35. The controller responded slower rise time when employing QuasiSMO, IT2FSMO and EKF, and faster with STASMO and HOSMO. It means that STASMO and HOSMO performed better than other approaches for yaw movement. However, the selected controller achieved step response with no overshoot for all the states with smooth tracking in z-axis movement.

High frequency chattering was still clearly visible in the control inputs following the use of STASMO and HOSMO as seen in Figures 6.43 - 6.46, while QuasiSMO, IT2FSMO,

and EKF provided smooth control signals for the quadcopter UAV.

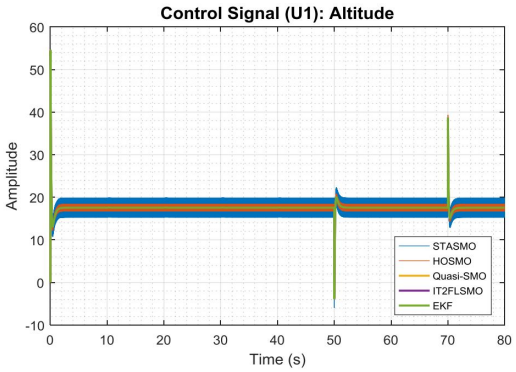

Figure 6.43. Control signal u_1

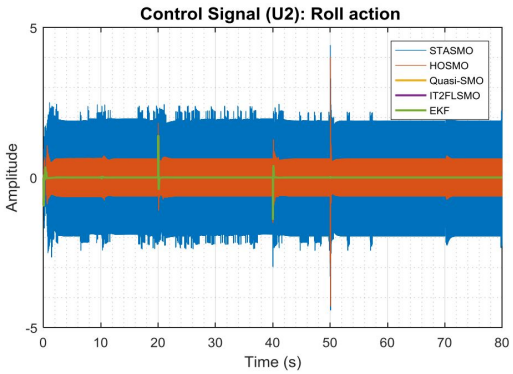

Figure 6.44. Control signal u_2

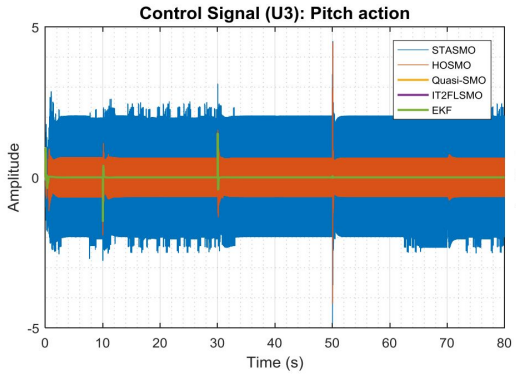

Figure 6.45. Control signal u_3

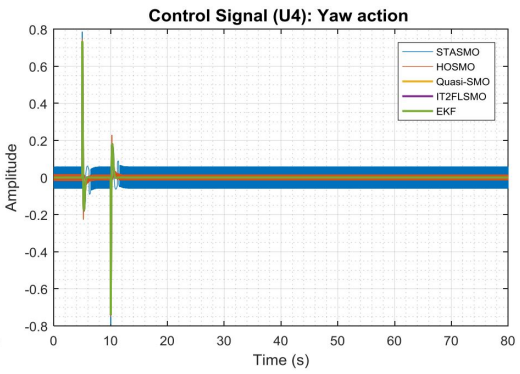

Figure 6.46. Control signal u_4

As noted further, QuasiSMO still outperformed other observers in enhancing the performance of QuasiSMC. It can be seen in the Table 6.9 that by applying QuasiSMO, the MSE of steady-state errors of the output states was smaller than other observers except yaw state for which HOSMO outperformed others.

Table 6.9. Mean squared errors of steady-states errors

Observer methods	**MSE steady-state error of**					
	x	**pitch**	**y**	**roll**	**yaw**	**z**
QuasiSMO	3.95479E-11	1.05333E-12	3.05006E-11	4.46757E-12	4.58096E-08	1.82786E-10
IT2FSMO	8.20855E-11	2.72824E-12	6.53954E-11	6.98284E-12	6.77775E-08	1.88921E-10
STASMO	4.64354E-06	9.01832E-06	2.04711E-07	8.71109E-07	9.29432E-08	2.30286E-08
HOSMO	5.22754E-08	7.46459E-08	6.87178E-08	9.3777E-08	1.48958E-09	4.33569E-09
EKF	6.50067E-11	1.82835E-10	5.20342E-11	4.24129E-11	5.1749E-08	1.59673E-10

In summary, in the presence of parameters mismatch, QuasiSMC still shows good performance in controlling the dynamic of quadcopter UAVs with no response overshoot, small rise time for the use of STASMO and HOSMO, free from chattering when applying QuasiSMO, IT2FSMO and EKF observer method, small steady-states errors, and

short computational time. In terms of observer, QuasiSMO method still outperforms other methods by demonstrating smooth estimated states, smaller states estimation errors of true states, and faster in tracking true states. Therefore, the observer has influenced and improved the overall performance of selected controller.

6.4.3 With noise and no parameter mismatch

In this case, white noise disturbance is introduced into the output states. The performances of selected control system and observer methods are presented in the simulation results shown in Figures 6.47 – 6.68.

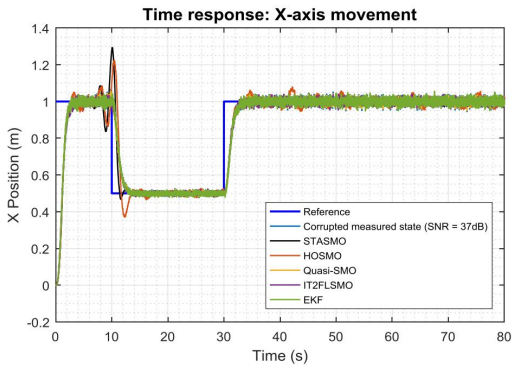

Figure 6.47. x-axis movement

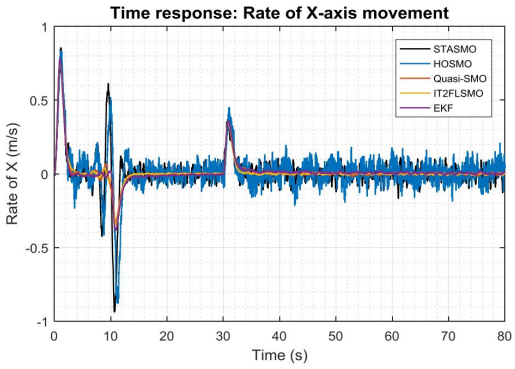

Figure 6.48. Rate of x-axis movement

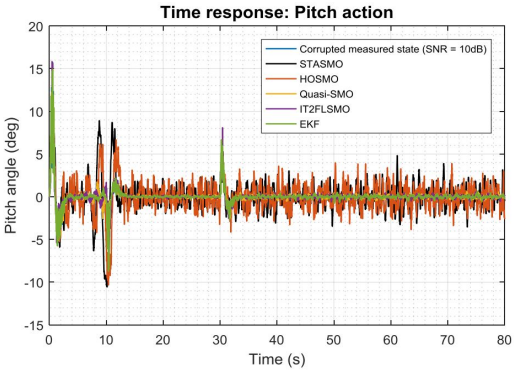

Figure 6.49. Pitch action

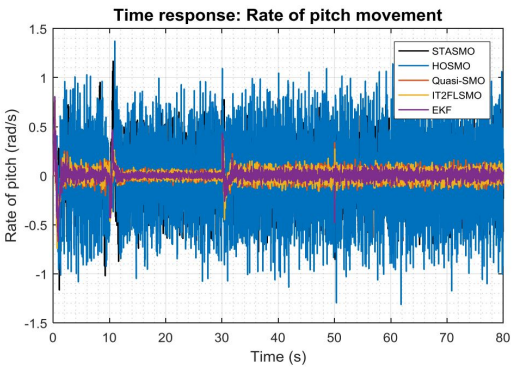

Figure 6.50. Rate of pitch movement

Figure 6.51. y-axis movement

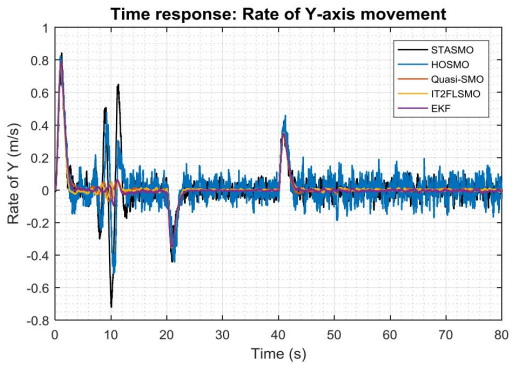

Figure 6.52. Rate of y-axis movement

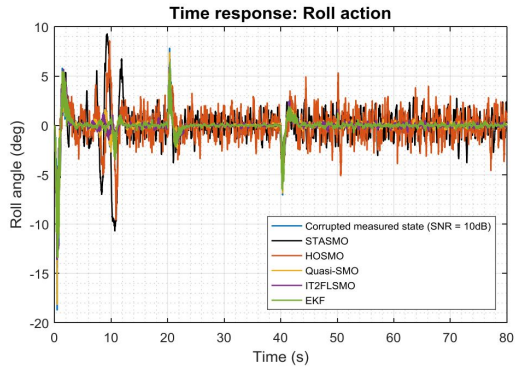

Figure 6.53. Roll action

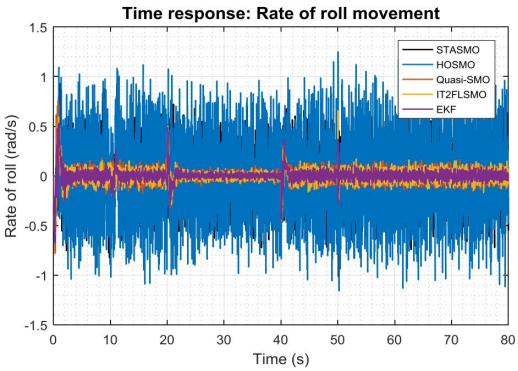

Figure 6.54. Rate of roll movement

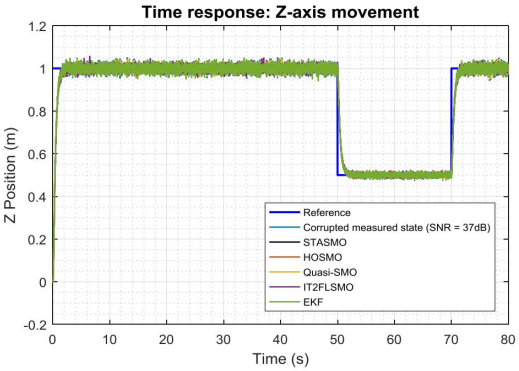

Figure 6.55. z-axis movement

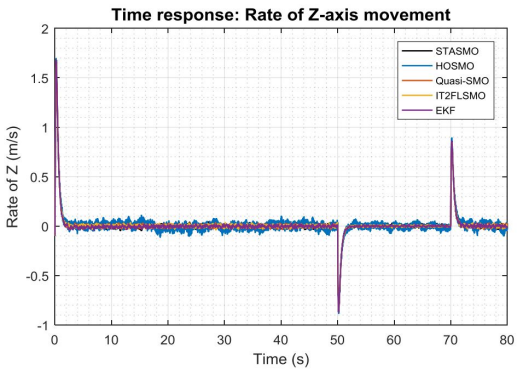

Figure 6.56. Rate of z-axis movement

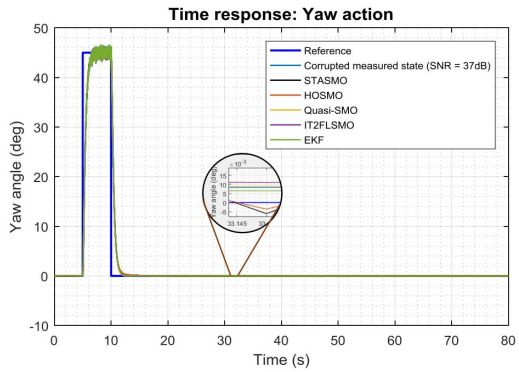

Figure 6.57. Yaw action

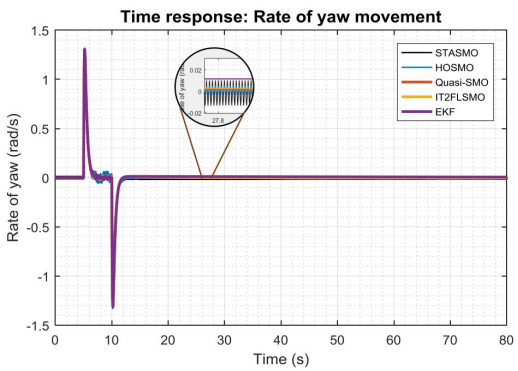

Figure 6.58. Rate of yaw movement

As noted in the simulation results presented in Figures 6.48, 6.50, 6.52, 6.54, 6.56 and

6.58, the effect of noise was reduced sufficiently by QuasiSMO, IT2FSMO, and EKF so that second order states were generated quite smoothly. Noise disturbance on the output states did not affect significantly the performance of the observers. In contrast, corrupted output states influenced the performances of STASMO and HOSMO quite significantly, especially in estimating second order states of pitch and roll as shown in Figures 6.50 and 6.54. The noise amplitude on the output states was amplified high enough due to the effect of switching functions on the methods.

Although the SNR on the output states x and y were the same as yaw and z states, that is 37dB where it was smaller than the noise on the pitch and roll states, that is 10dB, the effect of noise received by x and y was greater than yaw and z, as can be seen in Figures 6.48, 6.52, 6.56, and 6.58. This is because x and pitch, y and roll are dependent states which will affect each other.

Observer performance can be assessed also by analysing MSE of estimated states as presented in Table 6.10. From the table, it can be seen that QuasiSMO had smaller MSE in second order of x, y, yaw, and z than others, whereas EKF was better in estimating unmeasured states of pitch and roll rates.

Table 6.10. Mean squared errors of estimated states

Observer methods	MSE estimated state of					
	x rate	pitch rate	y rate	roll rate	yaw rate	z rate
QuasiSMO	2.52137E-05	0.002559446	2.38203E-05	0.002723063	1.5905E-06	0.000114864
IT2FSMO	1.94489E-05	0.002503558	2.4303E-05	0.002784655	2.96616E-06	0.000117486
STASMO	0.001805485	0.08552365	0.001478628	0.079785338	0.000140838	0.000255525
HOSMO	0.003835474	0.177975446	0.003706968	0.182360143	2.10793E-05	0.001335712
EKF	7.09669E-05	0.001330108	4.96901E-05	0.001316925	8.84775E-05	0.000115932

In the presence of noise, true states tracking errors appeared similar for all observer methods as shown in Figures 6.59 - 6.64. However, noting the computed MSE carefully as presented in Table 6.11, each observer technique displayed different performance in tracking true states. Although not able to estimate second order states as good as QuasiSMO, IT2FSMO and EKF; STASMO and HOSMO showed better performance in tracking true states. It is noticeable in Table 6.11 that those two observers had smaller MSE than other approaches in tracking the measured states of x, y, yaw and z, while QuasiSMO and EKF showed better performance in tracking pitch and roll states respectively. However,

STASMO and HOSMO did not perform as well as other approaches in estimating second order states so that the overall control performance would be affected.

Table 6.11. Mean squared errors of true states tracking

Observer method	MSE true state tracking of					
	x	pitch	y	roll	yaw	z
QuasiSMO	0.000270197	3.46874E-05	0.000273579	4.58094E-05	1.11938E-05	0.000282911
IT2FLO	0.000285062	4.23751E-05	0.000277166	4.67292E-05	1.14757E-05	0.000275964
STASMO	0.000178422	0.00012466	0.000173385	0.000136709	8.37523E-06	0.000179653
HOSMO	0.0001869	0.000124625	0.00018877	0.000111398	8.00168E-06	0.000188284
EKF	0.000314681	6.93525E-05	0.000300049	4.37804E-05	1.53207E-05	0.000311919

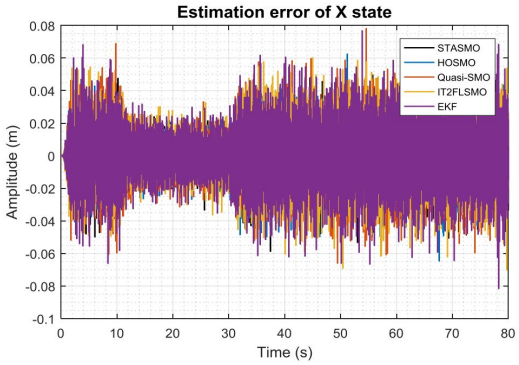

Figure 6.59. Estimation error of x

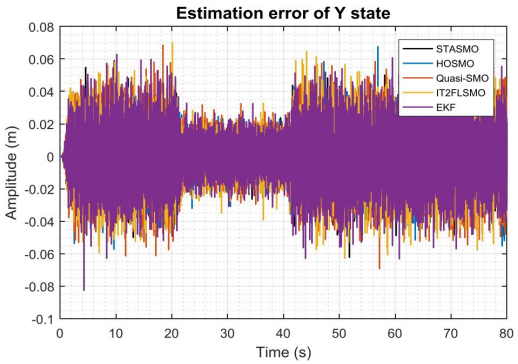

Figure 6.60. Estimation error of y

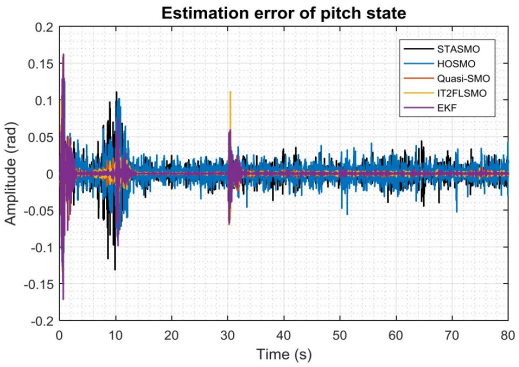

Figure 6.61. Estimation error of pitch

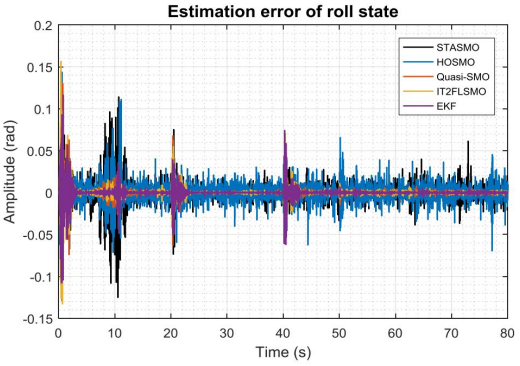

Figure 6.62. Estimation error of roll

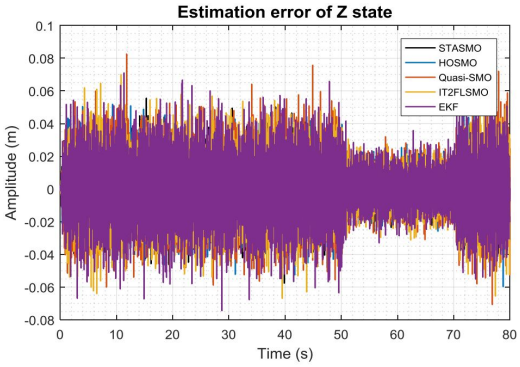

Figure 6.63. Estimation error of z

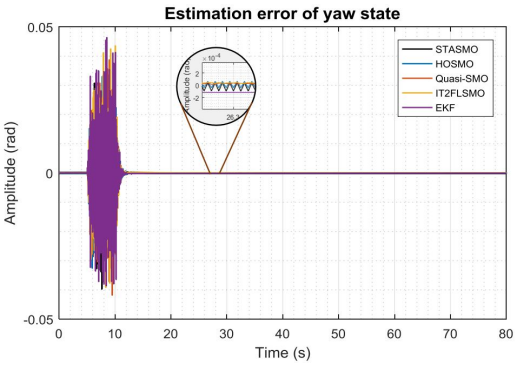

Figure 6.64. Estimation error of yaw

Noise is an inevitable fact in real-time applications. The presence of noise will certainly affect the performance of controller. However, the effect of noise on the systems depends on how to manage and select control and observer systems. It is noted in Figures 6.47, 6.49, 6.51, 6.53, 6.55, and 6.57, that the use of STASMO and HOSMO as states estimator had much effect on the control performance in controlling pitch, roll, x, and y. There were significant overshoot and undershoot when references changed around maximum 29.4% and 22.0% for STASMO and HOSMO respectively in x-axis movement, meanwhile 35.0% and 18.5% for STASMO and HOSMO respectively in y-axis moment. Higher overshoot was noticeable clearly in controlling pitch by around 8.9° and 8.58° for the use of STASMO and HOSMO respectively, while in roll state the overshoot was about 9.27° and 8.6° for STASMO and HOSMO respectively.

In contrast, the selected controller performed more stable in controlling all output states when using QuasiSMO, IT2FSMO, and EKF. However, the effect of noise was still noticeable clearly around references and control inputs as seen in Figures 6.65 - 6.68. This the trade-off of choosing parameters for the observers. On the one hand, increasing or decreasing observer gains will speed up tracking true value, yet may potentially increase sensitivity to measurement noise and vice versa. In this research, the speed of tracking measured states is more considered (without neglecting noise effect) to make the controller more responsive.

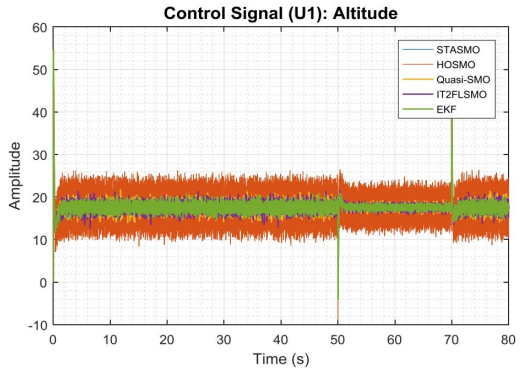

Figure 6.65. Control signal u_1

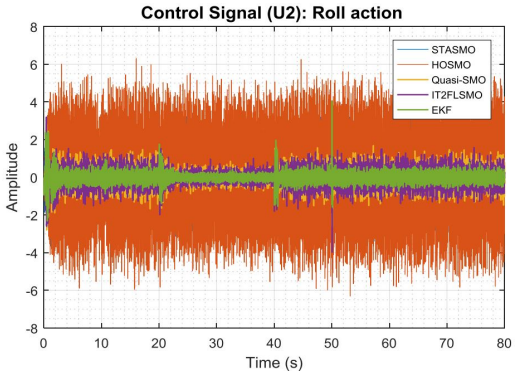

Figure 6.66. Control signal u_2

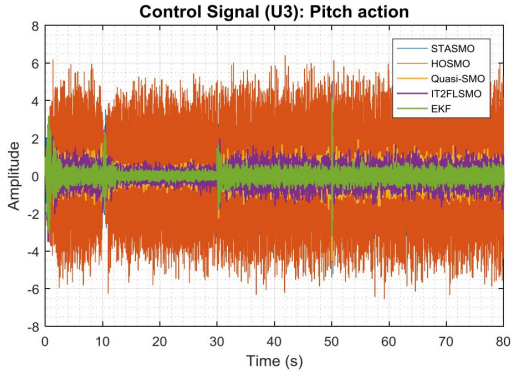

Figure 6.67. Control signal u_3

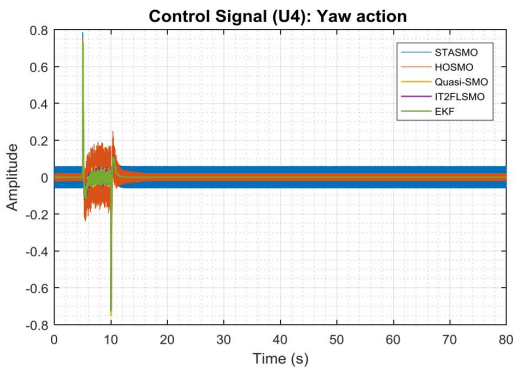

Figure 6.68. Control signal u_4

The steady-states error of tracking reference as a means of assessing the robustness of selected controller and observers in dealing with noise disturbance is shown in Table6.12. As noted, although STASMO and HOSMO have problem in handling noise effect in estimating unmeasured states, the observers have influenced control system to reduce steady-state error well. STASMO and HOSMO mostly had smaller steady-state error (except roll and pitch) computed by MSE method than other methods. Switching function in the algorithms has had the main role of keeping observed state tracking measured state all the time. However, due to noise disturbance, the control signals produced were not suitable for real-time embedded systems.

Table 6.12. Mean squared errors of steady-states errors

Observer methods	**MSE steady-state error of**					
	x	**pitch**	**y**	**roll**	**yaw**	**z**
QuasiSMO	0.000150976	0.000300922	0.000150774	0.000292534	3.69241E-08	0.000152913
IT2FSMO	0.000156631	0.000296579	0.00015233	0.000286674	3.88555E-08	0.00014974
STASMO	7.57813E-05	0.000725773	6.89261E-05	0.000621363	1.09064E-08	1.77685E-05
HOSMO	0.000511607	0.00075665	0.000386272	0.000835203	4.15275E-09	8.70393E-05
EKF	0.000206693	0.00029586	0.000187929	0.000288015	9.4867E-09	0.00019756

In summary, in the presence of noise, QuasiSMC still maintains good performance in controlling the dynamics of quadcopter UAV with no response overshoot except for roll and pitch action when applying STASMO and HOSMO, considered small rise time, free from chattering when using QuasiSMO, IT2FSMO and EKF observer methods, small steady-states errors, and short computational time. However, fluctuated signals randomly on the references still occurs due to noise disturbance. The selection of observer gain is being a trade-off. Reducing or increasing the gain may make observers become insensitive to noise, however, on the other hand it may delay tracking true states. As for the ob-

servers, generally QuasiSMO method still outperforms other methods by showing smooth estimated states, smaller MSE of estimated states and true states tracking errors, and faster in tracking true states. Therefore, the observer has influenced and improved the overall performance of selected controller.

6.4.4 With noise and with parameter mismatch

In this case, the performances of control system and observers are evaluated by introducing noise and uncertainty at the same time into the quadcopter UAV system. This situation maybe close to actual condition of real-time application. The performances of selected control system and observers are presented in Figures 6.69 - 6.90.

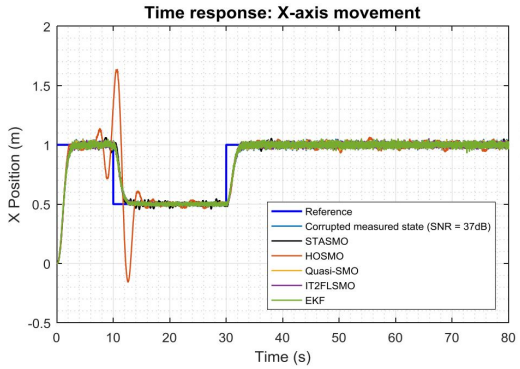

Figure 6.69. x-axis movement

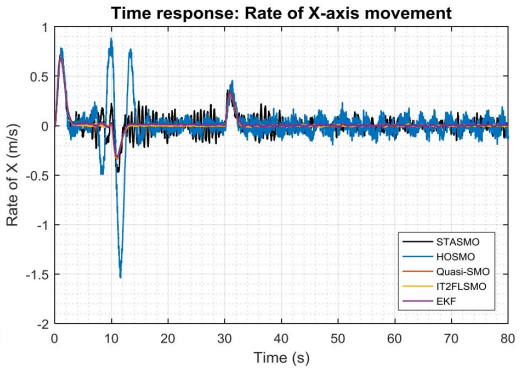

Figure 6.70. Rate of x-axis movement

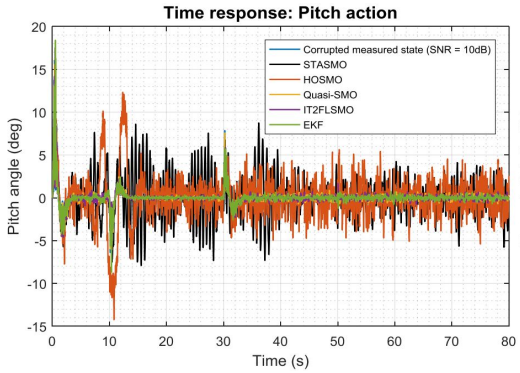

Figure 6.71. Pitch action

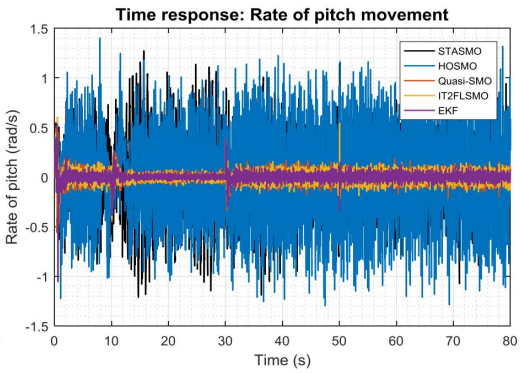

Figure 6.72. Rate of pitch movement

Figure 6.73. y-axis movement

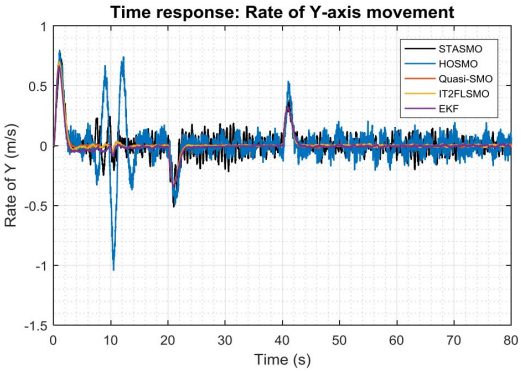

Figure 6.74. Rate of y-axis movement

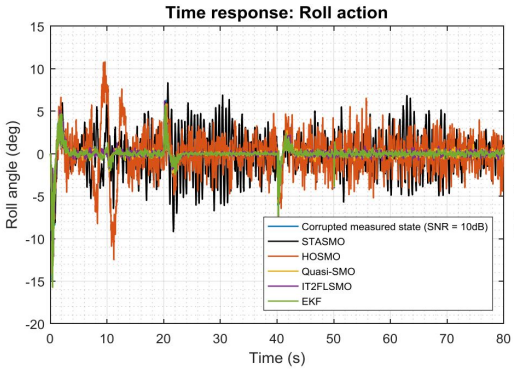

Figure 6.75. Roll action

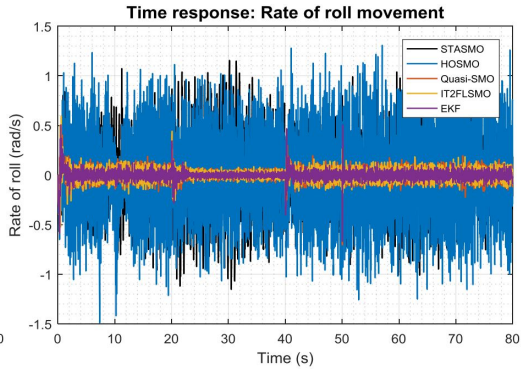

Figure 6.76. Rate of roll movement

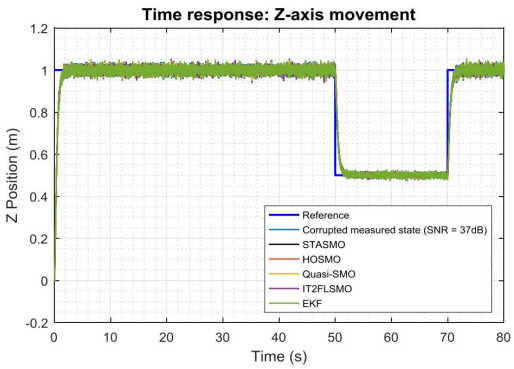

Figure 6.77. z-axis movement

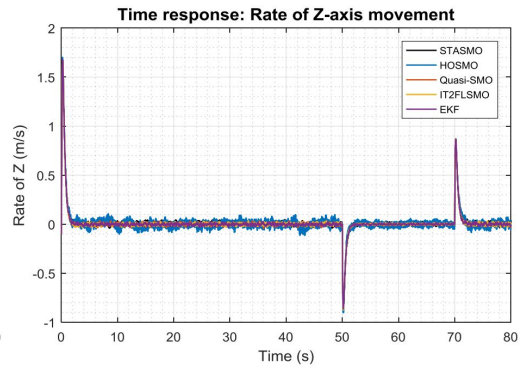

Figure 6.78. Rate of z-axis movement

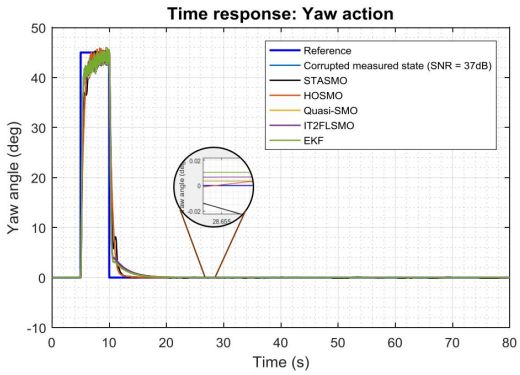

Figure 6.79. Yaw action

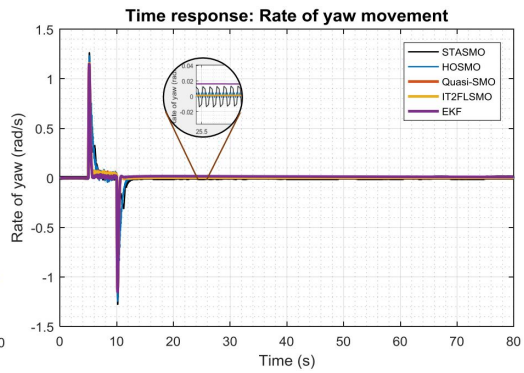

Figure 6.80. Rate of yaw movement

The same noise power as in the previous case with 50% parameters mismatch was ap-

plied to the system. As noted in the simulation results presented in Figures 6.70, 6.72, 6.74, 6.76, 6.78, and 6.80, in spite of the presence of high power noise and high uncertainty, the effect of disturbances were reduced sufficiently by QuasiSMO, IT2FSMO, and EKF so that second order states were generated quite smoothly. In other words, noise disturbance on the output states as well as parameters mismatch did not affect significantly the performance of the observers. In contrast, corrupted output states influenced the performance of STASMO and HOSMO quite significantly, especially in estimating second order states of pitch and roll as shown in Figures 6.72 and 6.76. The noise amplitude on the output states was amplified high enough due to the effect of switching functions on the methods. In addition, HOSMO could not deal with parameters mismatch well, as shown with high amplitude oscillation the estimation of x, and y movement rate when the state of x state changes.

Although the SNR on the output states x and y was the same as yaw and z states, that is 37dB where it was smaller than the noise on the pitch and roll states, that is 10dB, the effect of noise received by x and y was greater than yaw and z, as can be seen in Figures 6.70, 6.74, 6.78, and 6.80. This is because x and pitch, y and roll are dependent states which will affect each other.

MSE of estimated states was also calculated for further assessment of observers as presented in Table 6.13. As noted, QuasiSMO and EKF performed better in different unmeasured states estimation. In the use of QuasiSMO as estimator, the second order states of x, y, and yaw were generated nearly as expected by showing smaller values of MSE for those states, whereas EKF outperformed others in estimating unknown states of pitch, roll, and z rates. However, actually IT2FSMO had small MSE close to QuasiSMO and EKF.

Table 6.13. Mean squared errors of estimated states

Observer methods	**MSE estimated state of**					
	x rate	**pitch rate**	**y rate**	**roll rate**	**yaw rate**	**z rate**
QuasiSMO	2.37168E-05	0.002670402	2.52396E-05	0.003422104	1.94216E-08	0.000124037
IT2FSMO	2.68872E-05	0.002765641	2.9855E-05	0.002612052	2.14745E-07	0.00010497
STASMO	0.002081294	0.106644476	0.003412332	0.136032571	0.000143995	0.000337818
HOSMO	0.004053943	0.205306307	0.004175267	0.206927452	1.3365E-05	0.001194322
EKF	2.87102E-05	0.00129526	4.08341E-05	0.00199128	0.000158351	9.89014E-05

In the presence of noise and parameters mismatch, true states tracking errors appeared similar for all observer methods as presented in Figures 6.81 - 6.86. However, noting the computed MSE as presented in Table 6.14, each observer technique showed different performance in tracking true states. Despite of inability to estimating second order states as good as QuasiSMO, IT2FSMO and EKF; STASMO and HOSMO showed better performance in tracking true states. It is noticeable in the Table 6.14 that STASMO and HOSMO had smaller MSE than other approaches in tracking the measured states of x, y, yaw and z, while QuasiSMO and IT2FSMO showed better performance in tracking pitch and roll states. However, STASMO and HOSMO did not perform quite well as other approaches in estimating second order states so that the overall control performance would be affected.

Table 6.14. Mean squared errors of true states tracking

Observer method	**MSE true state tracking of**					
	x	**pitch**	**y**	**roll**	**yaw**	**z**
QuasiSMO	0.000269974	4.19776E-05	0.000270555	4.03963E-05	1.35795E-05	0.000285504
IT2FSMO	0.000279191	4.81532E-05	0.000281139	4.24167E-05	1.22059E-05	0.00028852
STASMO	0.000191511	0.002116093	0.000196571	0.001886163	0.000270709	0.000178146
HOSMO	0.00018736	0.000524501	0.000186359	0.000441203	1.19589E-05	0.000187872
EKF	0.000312682	5.39668E-05	0.000315449	4.31432E-05	1.59712E-05	0.0003217

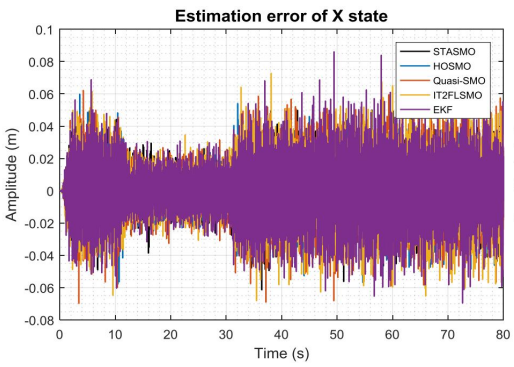

Figure 6.81. Estimation error of x

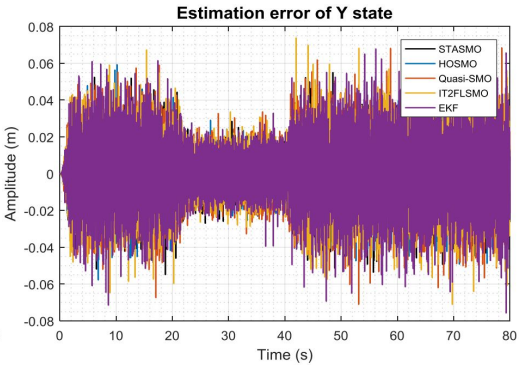

Figure 6.82. Estimation error of y

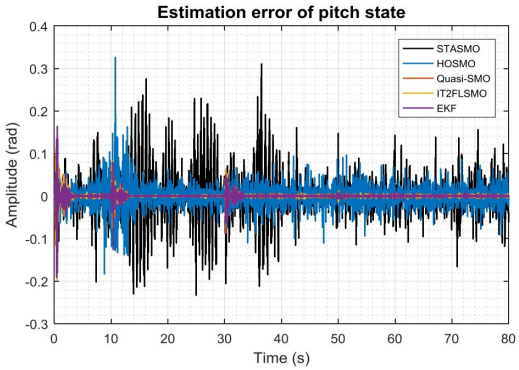

Figure 6.83. Estimation error of pitch

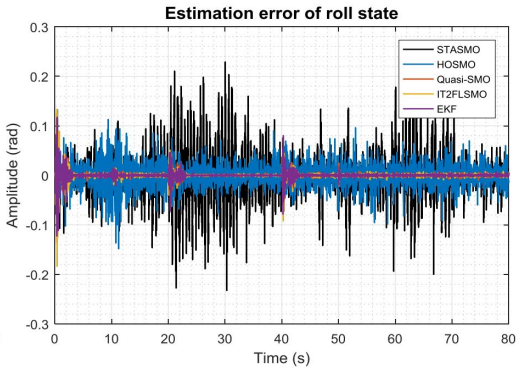

Figure 6.84. Estimation error of roll

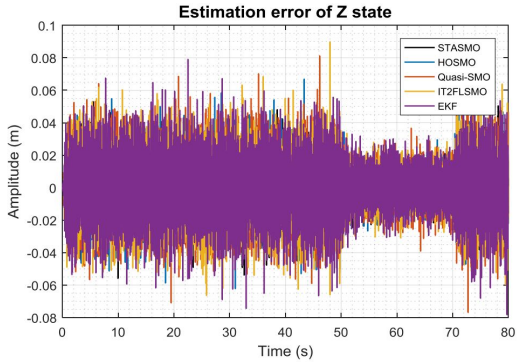

Figure 6.85. Estimation error of z

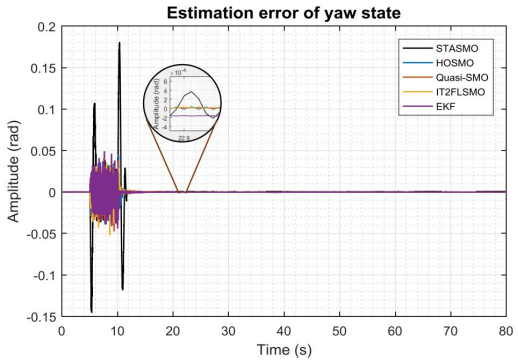

Figure 6.86. Estimation error of yaw

The presence of noise and parameters mismatch might certainly affect the performance of controller. However, the effect of noise on systems depends on how to manage and select control and observer systems. Notice in the results in Figures 6.69, 6.71, 6.73, 6.75, 6.77, and 6.79, the use of STASMO and HOSMO as states estimator had much effect on the control performance in controlling pitch, roll, x, and y. There was significant overshoot and undershoot especially when references changed around maximum 63.5% and 6.0% for HOSMO and STASMO respectively in x-axis movement, meanwhile 47.5% and 6.67% for HOSMO and STASMO respectively in y-axis moment. Not only high amplitude of random fluctuated states due to noise disturbance, higher overshoot also was noticeable clearly in controlling pitch by around $10.0°$ and $7.6°$ for HOSMO and STASMO respectively, while in roll state the overshoot was about $10.8°$ and $5.5°$ for HOSMO and STASMO respectively.

In contrast, the selected controller performed more stably in controlling all output states when employing QuasiSMO, IT2FSMO, and EKF. However, the noise effect was still noticeable clearly around references and control inputs as seen in Figures 6.87 - 6.90. This is the trade-off of choosing parameters for the observers. On the one hand, increasing or decreasing observer gains will speed up tracking true value, yet may potentially increase sensitivity to measurement noise and vice versa. In this research, the speed of tracking measured states is considered more (without neglecting noise effect) to make the controller more responsive.

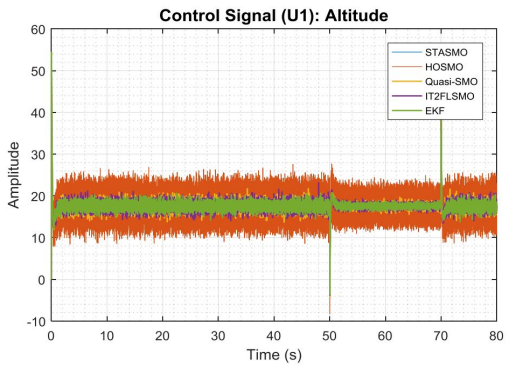

Figure 6.87. Control signal u_1

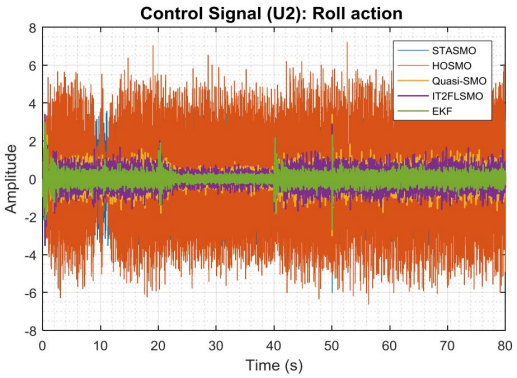

Figure 6.88. Control signal u_2

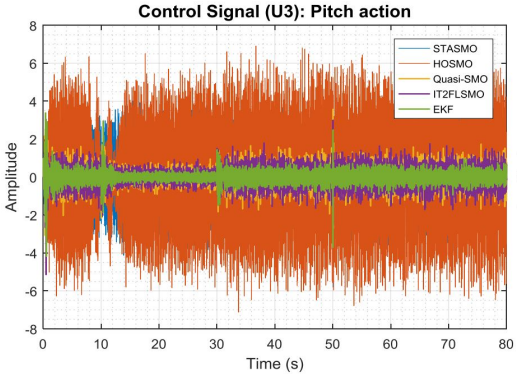

Figure 6.89. Control signal u_3

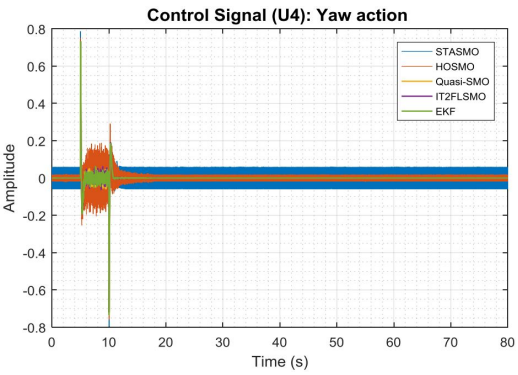

Figure 6.90. Control signal u_4

Table 6.15 shows, steady-states error as a further measure to evaluate the robustness of selected controller and observers in dealing with noise disturbance and parameters mismatch. As noted, generally steady-states errors of tracking reference were relatively small, which means that even though noise and parameters mismatch have put the quadcopter system in bad condition, the observers have influenced control system to stabilize the quadcopter UAV and reduce steady-state error quite well. No one or more observer methods showed dominating smaller steady-state error in every states as computed by MSE method. However, due to noise disturbance, the use of STASMO and HOSMO made the produced control signals not suitable for real-time embedded systems.

Table 6.15. Mean squared errors of steady-states errors

Observer methods	**MSE steady-state error of**					
	x	**pitch**	**y**	**roll**	**yaw**	**z**
QuasiSMO	0.000145758	0.005043404	0.000150085	0.005520106	4.58221E-08	0.000156155
IT2FSMO	0.000157451	0.005030683	0.000156812	0.005456679	5.38052E-08	0.000157494
STASMO	0.000137492	0.005354519	0.000230147	0.005658108	4.4223E-07	2.21427E-05
HOSMO	0.000565426	0.005952894	0.000493621	0.006463658	2.44169E-09	7.87743E-05
EKF	0.000201403	0.005033605	0.000204094	0.00544635	4.92746E-08	0.000197065

In summary, in the presence of noise and parameter mismatch, QuasiSMC maintains

good performance in controlling the dynamics of the quadcopter UAV with no response overshoot except for roll and pitch action when applying STASMO and HOSMO, considered small rise time, free from chattering when using QuasiSMO, IT2FSMO and EKF observer method, small steady-states errors, and short computational time. However, fluctuated signals randomly on the references still appeared due to noise disturbance. The selection of observer gain is being a trade-off. Reducing or increasing the gain may make observers become insensitive to noise, however, on the other hand it may delay tracking true states. As for the observers, QuasiSMO method still outperformed other methods by showing smooth estimated states, smaller states estimation errors, and faster in tracking true states. Therefore, the observer has influenced and improved the overall performance of selected controller.

6.5 Performance summary

After conducting performance evaluation of the control system and the observer methods with 4 predefined scenarios, it can be summarized that set-point integral quasi-sliding mode control (QuasiSMC) can still maintain the stability of its performance to control the dynamics of quadcopter UAV well in all conditions. Integral term plays an important role in reducing steady-state error of the output states while set-point weighting function plays a role in reducing overshoot due to the integral term. Oscillations on pitch and roll motions only occur in the scenario of with noise and parameter mismatch (scenario 2-4) when STASMO and HOSMO observer methods were employed. In terms of observer, actually QuasiSMO and IT2FSMO have similar performance in estimating unmeasured states. However, IT2FSMO has slower computation time than QuasiSMO. Hence, generally QuasiSMO outperformed than others and has contributed to improved control system performance in dealing with all predefined conditions

Chapter 7

Numerical simulations: set-point integral sliding mode-based interval type-2 fuzzy logic control

7.1 Introduction

This chapter presents numerical simulation results to highlight the performance of set-point integral sliding mode-based interval type-2 fuzzy control (IT2FSMC) with the developed nonlinear full-order state observers, namely:

- quasi - sliding mode observer
- sliding mode-based interval type-2 fuzzy observer
- super-twisting algorithm of sliding mode observer
- higher order sliding mode observer
- extended Kalman filter

Performance comparison between sliding mode-based observers and extended Kalman filter (EKF) is conducted to highlight the robustness of each method. Through numerical validation, the performance of the control system and the performance comparisons of the observer methods are carried out and presented in this chapter.

The overall control and observer system diagram, simulation setup and performance criteria are the same as in Chapter 6. In addition, the same scenerios as in Chapter 6 are introduced to test the performance of the control and observer methods, including:

- without noise and no parameter mismatch
- without noise and with parameter mismatch
- with noise and no parameter mismatch
- with noise and with parameter mismatch

The purpose of this approach is to determine the ability of the controller and observer to deal with several conditions.

The reference frame of the quadcopter movements in this numerical simulation can be seen in Figure 7.1

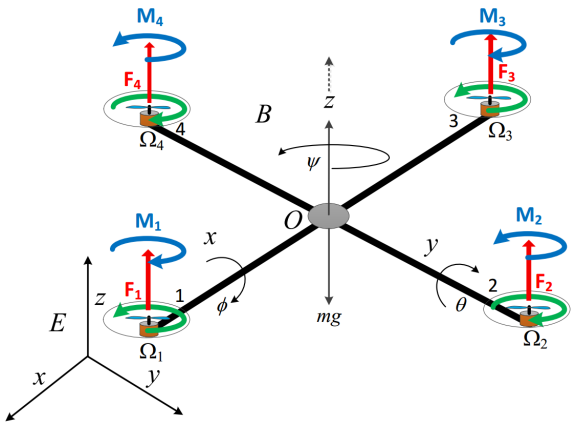

Figure 7.1. Reference frame of quadcopter movements

where x, y, and z represent the movements of the quadcopter in those axes, while ϕ, θ, and ψ represent roll, pitch, and yaw motions of the quadcopter respectively.

7.2 Simulation results

After a series of investigations, the following sliding mode parameter were found suitable for the desired performance achievement,

$$\lambda_\phi = 16.5 \quad \lambda_\theta = 16.5 \quad \lambda_\psi = 3.2935 \quad \lambda_x = 2.3 \quad \lambda_y = 2.3 \quad \lambda_z = 3.3950$$

$$\lambda_{i\phi} = 30.0 \quad \lambda_{i\theta} = 30.0 \quad \lambda_{i\psi} = 1.4 \quad \lambda_{ix} = 1.82 \quad \lambda_{iy} = 1.82 \quad \lambda_{iz} = 1.45$$

$$k_\phi = 10.0 \quad k_\theta = 10.0 \quad k_\psi = 5.0 \quad k_x = 2.0 \quad k_y = 2.0 \quad k_z = 5.0$$

7.2.1 Without noise and no parameter mismatch

Noise-free disturbance and no uncertainty are the ideal conditions used to assess the behaviour of selected controller and observers. For this condition, the simulation results obtained are presented in Figures 7.2 – 7.23.

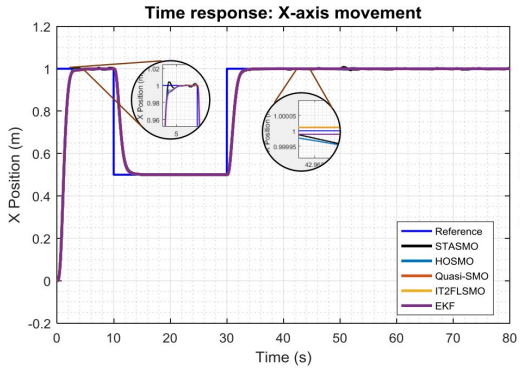

Figure 7.2. x-axis movement

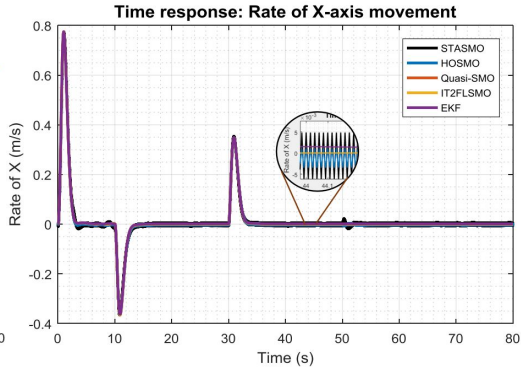

Figure 7.3. Rate of x-axis movement

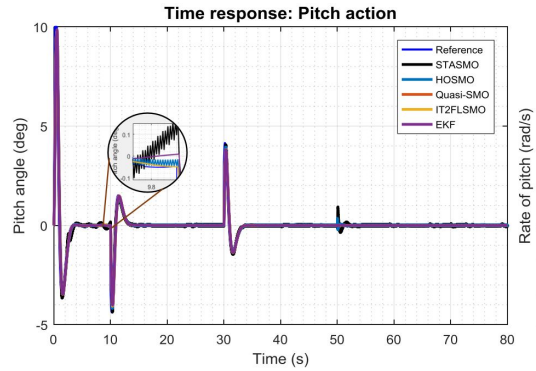

Figure 7.4. Pitch action

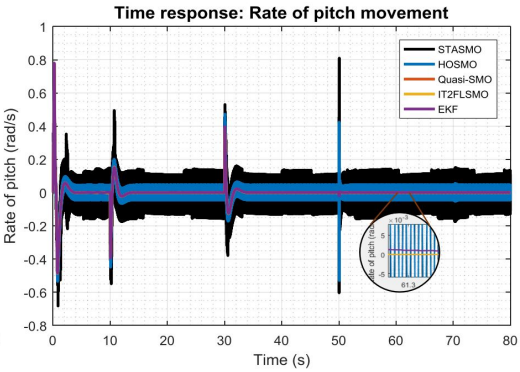

Figure 7.5. Rate of pitch movement

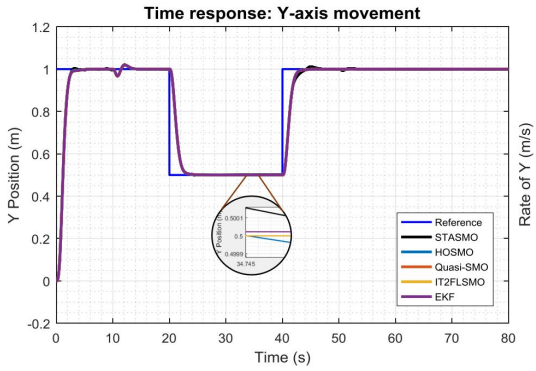

Figure 7.6. y-axis movement

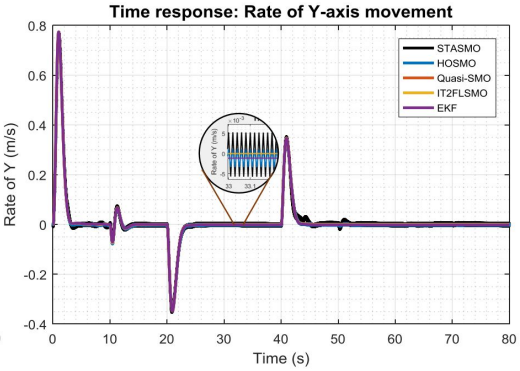

Figure 7.7. Rate of y-axis movement

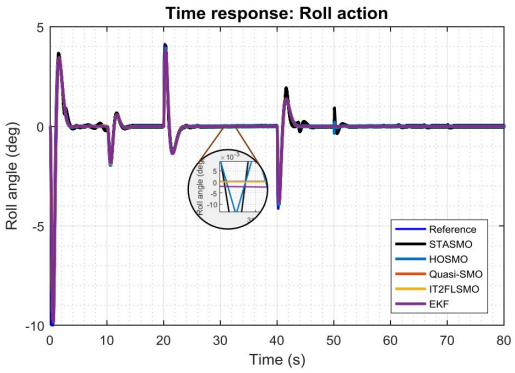

Figure 7.8. Roll action

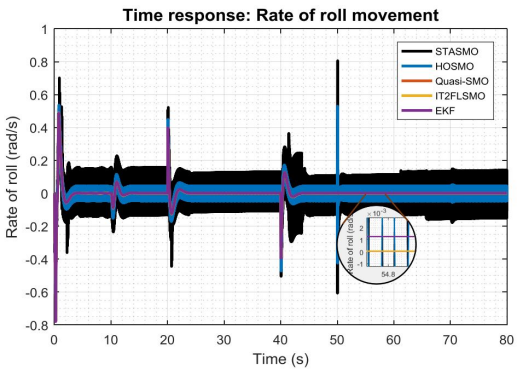

Figure 7.9. Rate of roll movement

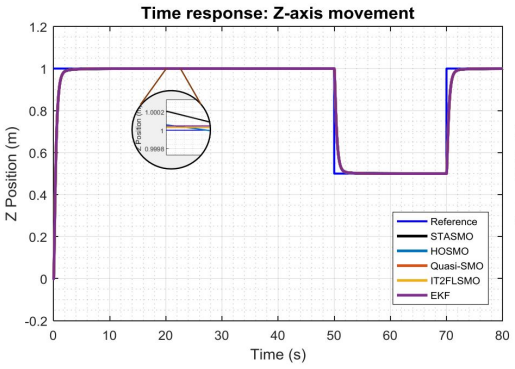

Figure 7.10. z-axis movement

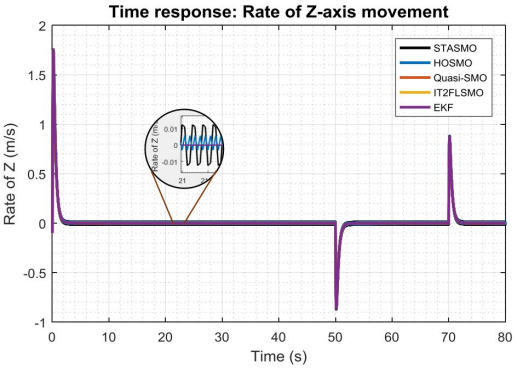

Figure 7.11. Rate of z-axis movement

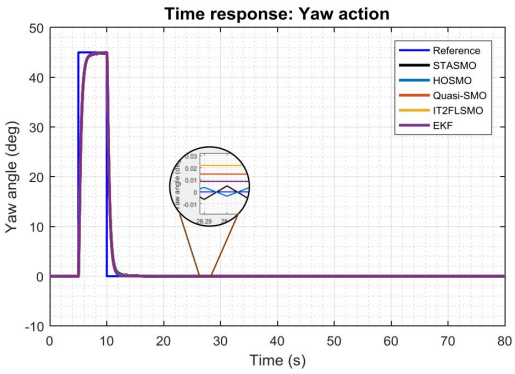

Figure 7.12. Yaw action

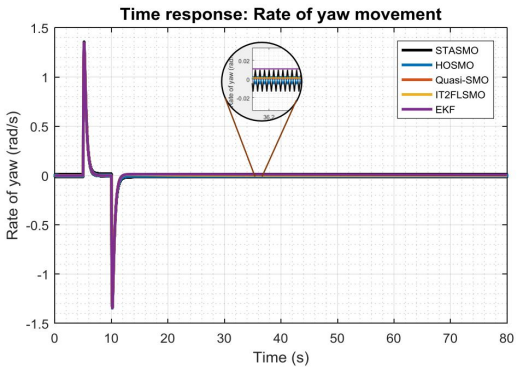

Figure 7.13. Rate of yaw movement

Generally, in terms of observer performance, all observers performed well in estimating unmeasured states of the quadcopter UAV as can be seen in Figures 7.3, 7.5, 7.7, 7.9, 7.11, and 7.13. However, oscillation still appeared when using STASMO and HOSMO approaches especially in estimating second order states of pitch and roll as shown in Figures 7.5 and 7.9 with significant amplitude of oscillation. Although the chattering was reduced significantly in these methods, switching function used in both observers is still the main cause of oscillation in the estimated states.

To validate the performance of observers, the MSE of estimated states in steady-state

condition was computed as presented in the Table 7.1. Although different control systems were applied, QuasiSMO still showed promising performance in this case. In addition to smooth estimated states, the observer generated the second order states with smaller MSE than others. It means that the estimated states were obtained as expected so the control system can perform well.

Table 7.1. Mean squared errors of estimated states

Observer methods	MSE estimated state of					
	x rate	pitch rate	y rate	roll rate	yaw rate	z rate
QuasiSMO	2.39749E-10	6.41929E-10	8.8172E-11	5.0959E-10	1.59899E-06	1.14843E-09
IT2FSMO	6.82505E-10	2.03702E-09	2.52365E-10	1.1549E-09	3.07835E-06	1.20673E-09
STASMO	3.57669E-05	0.014701626	3.86094E-05	0.015663435	0.000138843	0.000136884
HOSMO	6.62949E-06	0.002133262	6.68746E-06	0.002183735	2.07331E-05	2.09153E-05
EKF	2.04492E-06	1.31699E-06	2.88918E-06	1.44045E-06	9.44783E-05	2.22096E-09

Each observer showed different performance in tracking true states. With reference to Figures 7.14 - 7.19, QuasiSMO and IT2FSMO exhibited small estimation errors in steady state condition, whereas EKF performed better in states change condition of x and y movement by showing smaller estimation error. Furthermore, MSE calculation method was utilized to measure the quality of observers, as presented in Table 7.2. As noted, QuasiSMO and IT2FSMO outperformed other sliding mode-based observers and EKF by achieving smaller MSE for almost all states except x, and y states for which EKF performed better than those two.

Table 7.2. Mean squared errors of true states tracking

Observer method	MSE true state tracking of					
	x	pitch	y	roll	yaw	z
QuasiSMO	3.57967E-10	1.21045E-11	3.59388E-10	5.54736E-12	1.55107E-09	3.77395E-12
IT2FSMO	3.62949E-10	1.59756E-11	3.62805E-10	6.49756E-12	1.92545E-09	3.81899E-12
STASMO	5.17942E-09	6.51204E-09	5.4246E-09	6.34426E-09	5.65467E-09	9.19845E-09
HOSMO	3.00697E-09	2.9766E-09	3.00993E-09	2.97562E-09	2.97354E-09	2.97575E-09
EKF	2.40973E-10	2.53348E-09	1.98774E-10	2.58893E-09	1.37554E-08	6.48458E-09

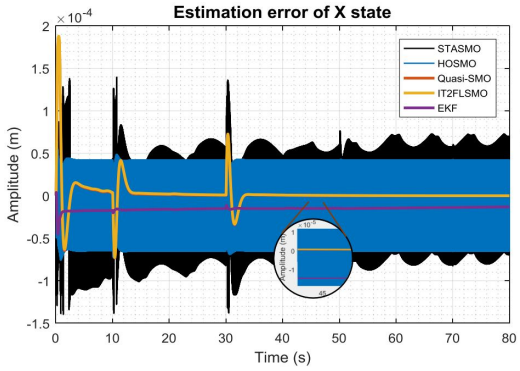

Figure 7.14. Estimation error of x

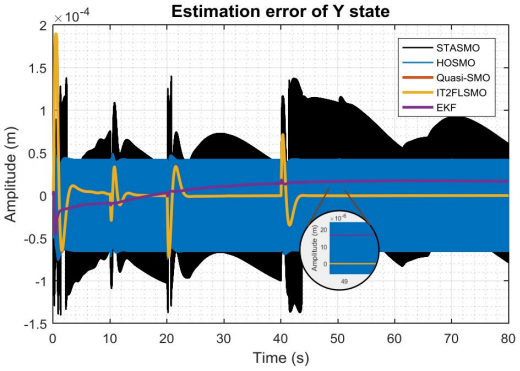

Figure 7.15. Estimation error of y

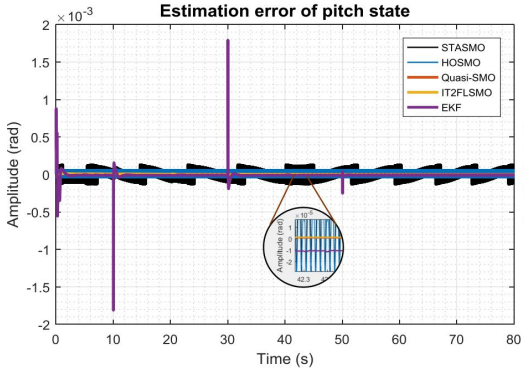

Figure 7.16. Estimation error of pitch

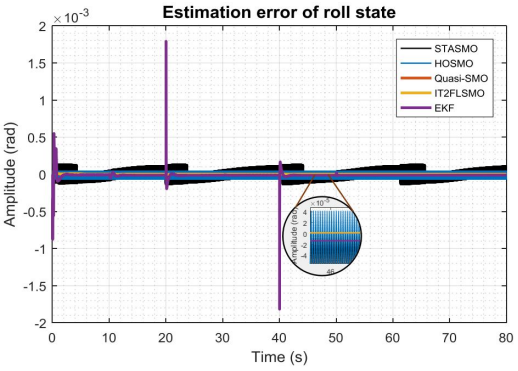

Figure 7.17. Estimation error of roll

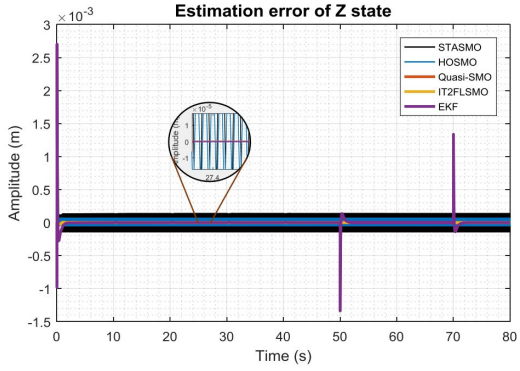

Figure 7.18. Estimation error of z

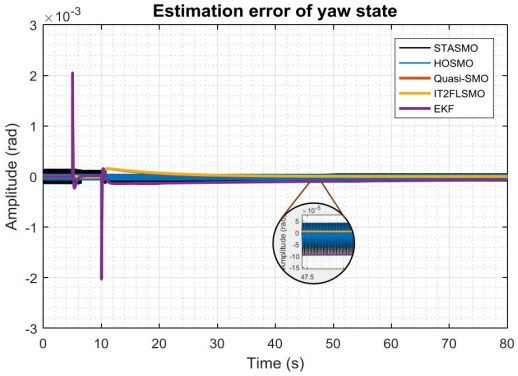

Figure 7.19. Estimation error of yaw

On the grounds of control, good performance in tracking references was achieved as shown in Figures 7.2 - 7.13. The IT2FSMC showed good performance in controlling the quadcopter with various observers. As noted there was no overshoot in step response of the system with all observers with similar rise time. However, high frequency oscillations were seen in the control inputs with using STASMO and HOSMO as can be seen in Figures 7.20 - 7.23, whereas QuasiSMO, IT2FSMO and EKF could influence the control system to generate smoother control signals for the aircraft. However, the time to process the control sequence was larger than for QuasiSMC around 3.6573 ms which might affect the overall

performance in real-time applications.

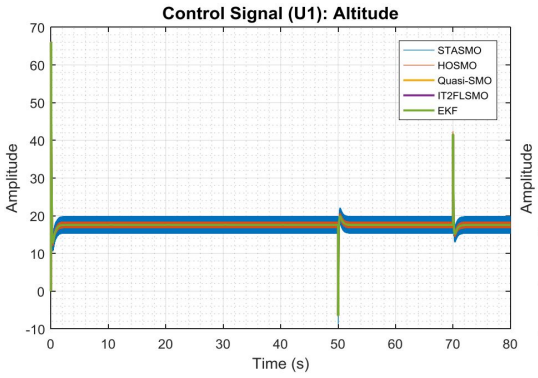

Figure 7.20. Control signal u_1

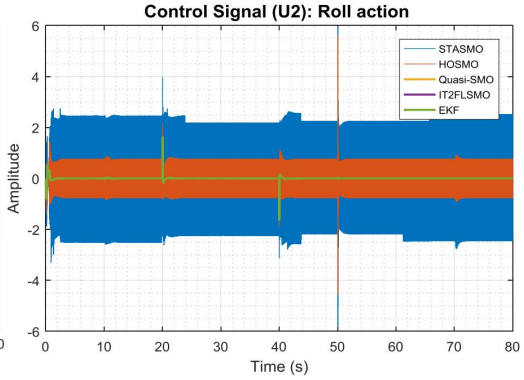

Figure 7.21. Control signal u_2

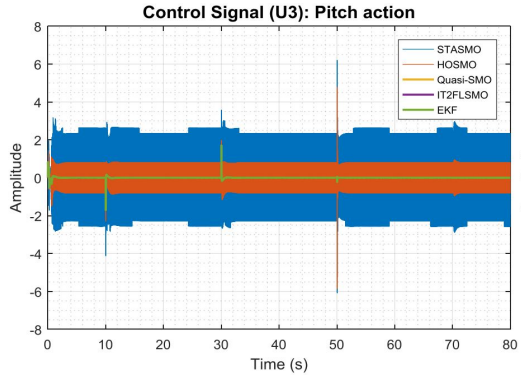

Figure 7.22. Control signal u_3

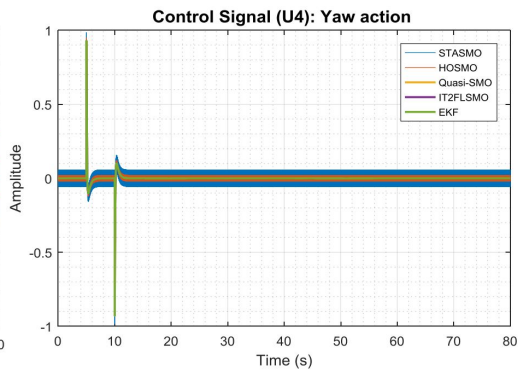

Figure 7.23. Control signal u_4

Table 7.3 shows mean squared error of output states of the quadcopter UAV in steady-state condition. Generally, QuasiSMO outperformed other observers in improving the performance of selected controller by showing smaller MSE in almost all output states except yaw for which HOSMO performed better than others.

Table 7.3. Mean squared errors of steady-states errors

Observer methods	**MSE steady-state error of**					
	x	**pitch**	**y**	**roll**	**yaw**	**z**
QuasiSMO	8.37836E-12	8.97454E-14	7.01513E-13	2.06849E-12	4.25678E-08	6.48996E-10
IT2FSMO	1.16551E-11	4.12576E-13	1.55266E-12	2.21758E-12	4.5355E-08	6.37865E-10
STASMO	7.12884E-07	1.39214E-06	1.09761E-06	1.71975E-06	1.01748E-08	2.12865E-08
HOSMO	4.03366E-08	1.24548E-07	5.31153E-08	1.42662E-07	4.29632E-09	5.09664E-09
EKF	1.33487E-09	2.53979E-09	5.87196E-11	2.04178E-10	1.26037E-08	6.73338E-10

In summary, in the condition of free of noise and no parameters mismatch, IT2FSMC shows good performance in controlling the dynamics of the quadcopter UAV with no response overshoot, small rise time, free from chattering when employing QuasiSMO, IT2FSMO and EKF observer methods, and small steady-states errors. However, the con-

troller has longer computational time than QuasiSMC. Type reduction method in type-2 fuzzy logic is one component which leads to such delay. Type reduction is still fascinating topic to be researched in type-2 fuzzy logic field to make this method feasible to be used in wide range of real-time applications. In terms of observer, QuasiSMO method outperforms other methods by showing smooth estimated states, smaller states estimation errors of true states, and faster in tracking true states. Therefore, the observer has influenced and improved the overall performance of selected controller.

7.2.2 Without noise and with parameter mismatch

Parameters mismatch was introduced into the system to assess the performance of the selected control system and proposed observer method. The simulation results obtained are presented in Figures 7.24 – 7.45.

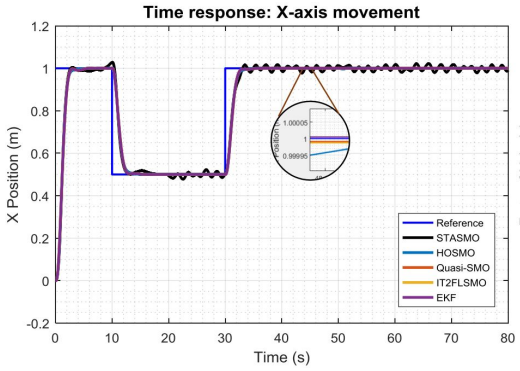

Figure 7.24. x-axis movement

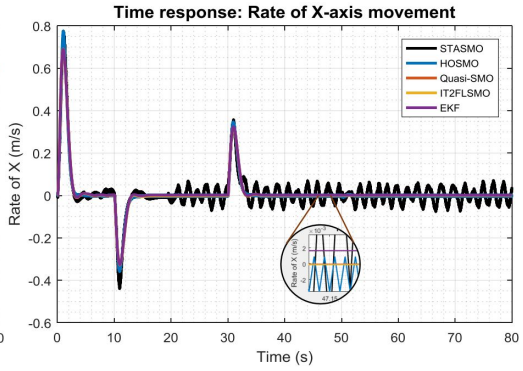

Figure 7.25. Rate of x-axis movement

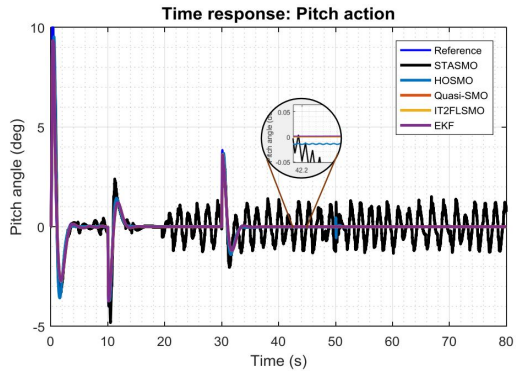

Figure 7.26. Pitch action

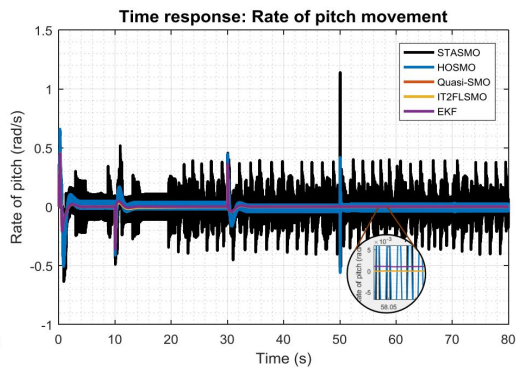

Figure 7.27. Rate of pitch movement

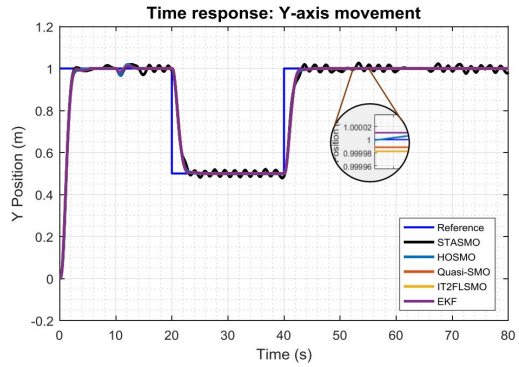

Figure 7.28. y-axis movement

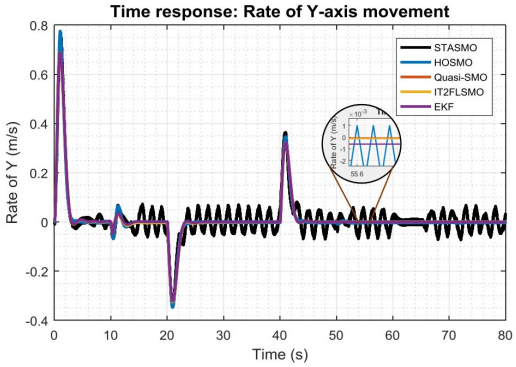

Figure 7.29. Rate of y-axis movement

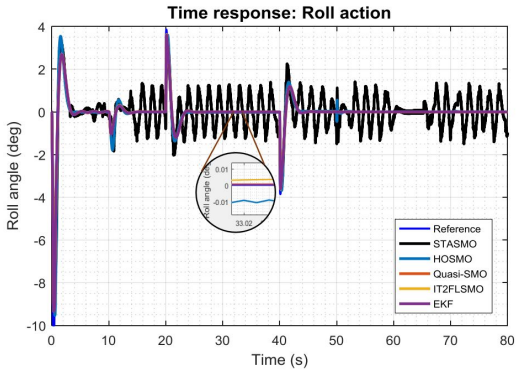

Figure 7.30. Roll action

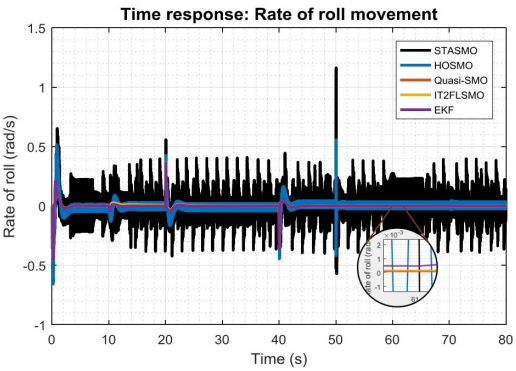

Figure 7.31. Rate of roll movement

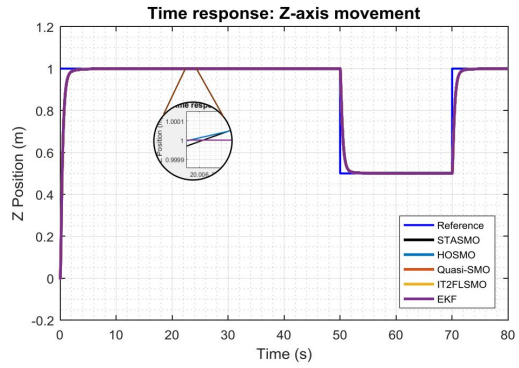

Figure 7.32. z-axis movement

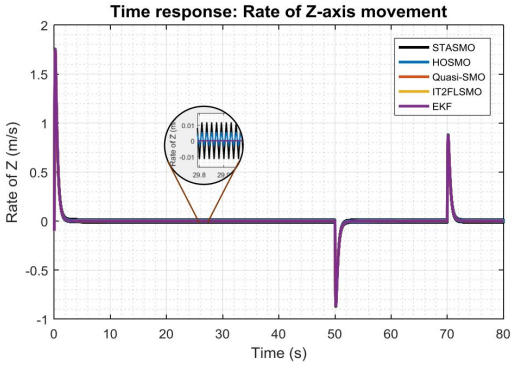

Figure 7.33. Rate of z-axis movement

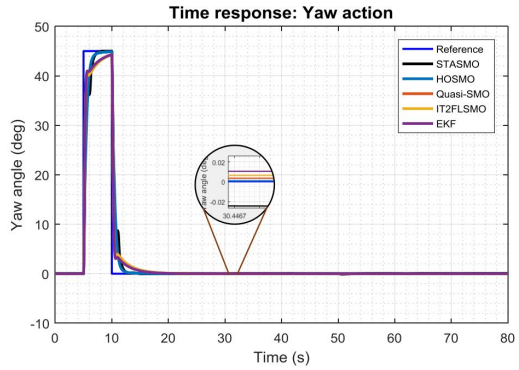

Figure 7.34. Yaw action

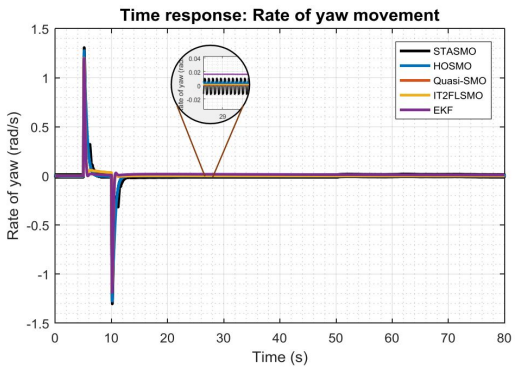

Figure 7.35. Rate of yaw movement

In this case, parameter uncertainty was introduced to conduct further testing of selected

controller and observers. Observing the simulation results presented in Figures 7.25, 7.27, 7.29, 7.31, 7.33, 7.35, the situation significantly affected the performance of STASMO in estimating some unmeasured states especially rate of x, pitch, y, and roll as depicted in Figures 7.25, 7.27, 7.29, and 7.31. In addition to high frequency oscillation caused by switching function, the amplitude of estimated states fluctuated randomly. Furthermore, the estimating process of x and y rate states were influenced by pitch and roll rate, because those states are coupled. QuasiSMO, IT2FSMO, HOSMO, and EKF methods were more robust than STASMO in dealing with parameters mismatch by showing similar performance with the system without parameters uncertainty.

In the steady-state condition, each observer showed different performance in estimating the unknown states as seen in the Table 7.4. In addition to generating smooth estimated states, QuasiSMO, IT2FSMO and EKF exhibited their advantages in estimating second order states by having small MSE of all estimated states. However, QuasiSMO was still the best among other observers by showing smaller values than others.

Table 7.4. Mean squared errors of estimated states

Observer methods	**MSE estimated state of**					
	x rate	**pitch rate**	**y rate**	**roll rate**	**yaw rate**	**z rate**
QuasiSMO	2.53689E-09	7.94116E-09	2.97629E-09	1.13403E-08	3.02786E-08	2.49775E-10
IT2FSMO	6.86499E-09	2.02171E-08	9.48718E-09	3.08426E-08	6.88646E-07	2.72383E-10
STASMO	0.001045385	0.020893679	0.000893303	0.022146886	0.000140813	0.000137553
HOSMO	7.03848E-06	0.001109202	7.17371E-06	0.00111765	1.56732E-05	2.09147E-05
EKF	2.41038E-06	1.12172E-06	4.08395E-07	2.44015E-07	0.000179461	5.15093E-10

Figures 7.36 - 7.41 show the different responses of observers in tracking measured states. Generally, all approaches performed well in tracking true states by achieving small estimation errors for all measured states. However, some observers including QuasiSMO, IT2FSMO, and EKF performed better than STASMO and HOSMO. Fluctuated errors were noticeable with using STASMO and HOSMO methods. Similar with the first condition, in steady-state condition the estimated states of true states generated by QuasiSMO and IT2FSMO had expected results by showing smaller estimation errors of true states than EKF. The overall performance of tracking true states evaluated by MSE calculation is presented in Table 7.5 where as noted EKF outperformed others by showing smaller MSE for almost all states except x and z states for which QuasiSMO performed better than

others.

Table 7.5. Mean squared errors of true states tracking

Observer method	MSE true state tracking of					
	x	pitch	y	roll	yaw	z
QuasiSMO	1.2304E-10	9.97789E-08	2.5828E-10	1.00255E-07	1.01214E-06	6.01419E-13
IT2FSMO	1.59312E-10	1.0016E-07	3.79838E-10	1.01059E-07	1.01944E-06	6.05809E-13
STASMO	6.975E-09	4.56671E-06	7.07056E-09	4.95733E-06	0.000235138	4.89481E-09
HOSMO	3.00048E-09	2.60717E-07	3.00042E-09	2.50405E-07	2.34066E-06	2.97571E-09
EKF	3.21091E-10	7.79136E-08	9.97841E-11	7.78931E-08	8.09099E-07	6.51213E-09

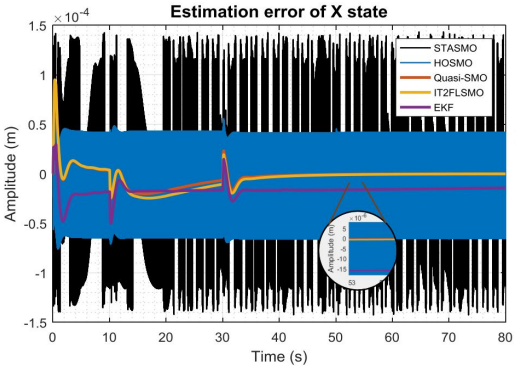

Figure 7.36. Estimation error of x

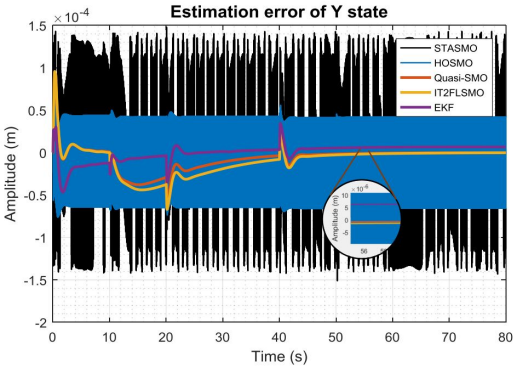

Figure 7.37. Estimation error of y

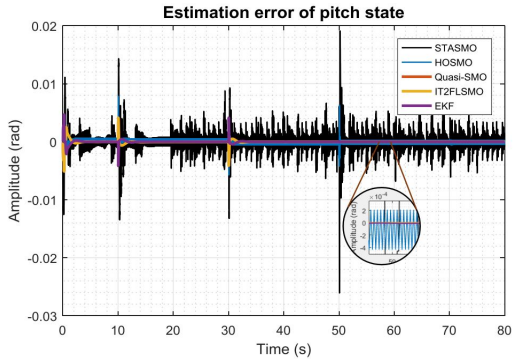

Figure 7.38. Estimation error of pitch

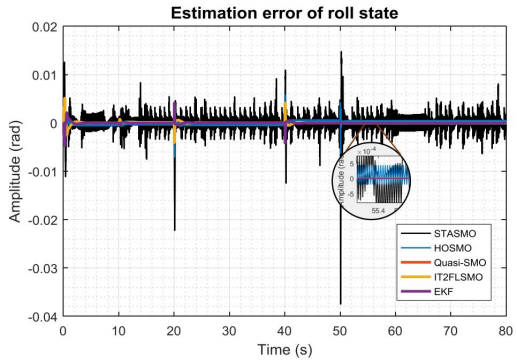

Figure 7.39. Estimation error of roll

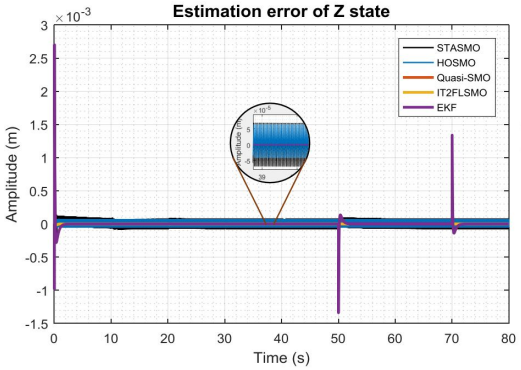

Figure 7.40. Estimation error of z

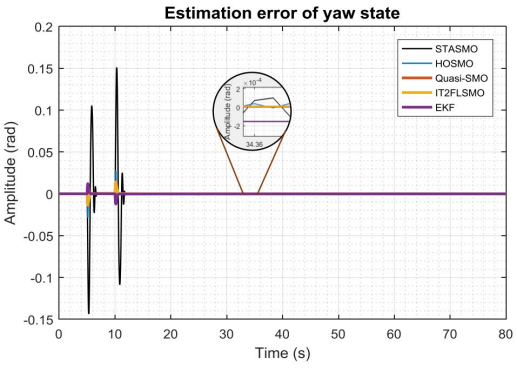

Figure 7.41. Estimation error of yaw

The presence of parameter uncertainty has influenced significantly the control performance in managing x, pitch, y and roll as seen in Figures 7.24, 7.26, 7.28, and 7.30 when

employing STASMO. In steady-state condition of roll and pitch motion, the states fluctuated around $\pm 1.25^{\circ}$ and $\pm 1.51^{\circ}$ respectively. Because pitch and x, and roll and y states are coupled, the reference tracking of x and y were affected as seen in Figures 7.24 and 7.26. It can be seen that small fluctuation around $\pm 2cm$ appeared in steady-state condition of x and y movements.

In addition, different behaviour can be seen in yaw action response as in Figure 7.34. The control system responded with slower rise time when employing QuasiSMO, IT2FSMO and EKF, but had faster response when using STASMO and HOSMO methods. It means that in yaw motion, STASMO and HOSMO influenced the control system to perform better than QuasiSMO, IT2FSMO and EKF. However, in z-axis movement, the selected controller responded to step inputs well by tracking the reference smoothly with no overshoot.

Furthermore, high frequency chattering was still clearly visible in the control inputs with using STASMO and HOSMO as seen in Figures 7.42 - 7.45, while QuasiSMO, IT2FSMO, and EKF could improve the control method to generate smooth control signals for the quadcopter UAV.

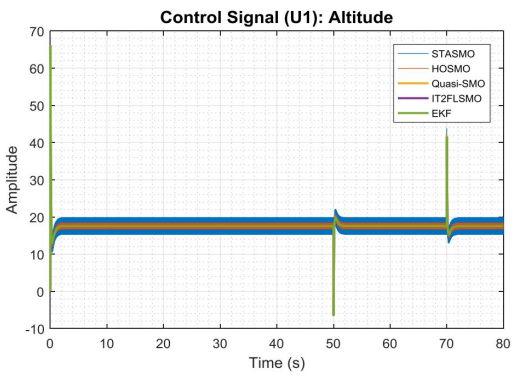

Figure 7.42. Control signal u_1

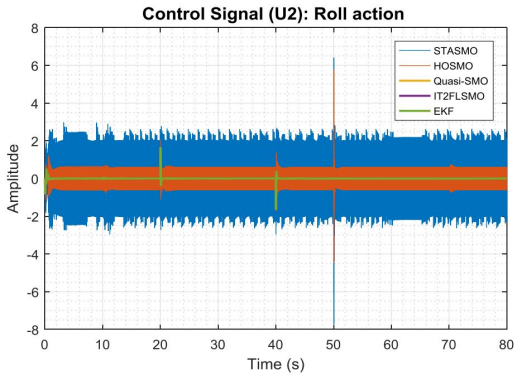

Figure 7.43. Control signal u_2

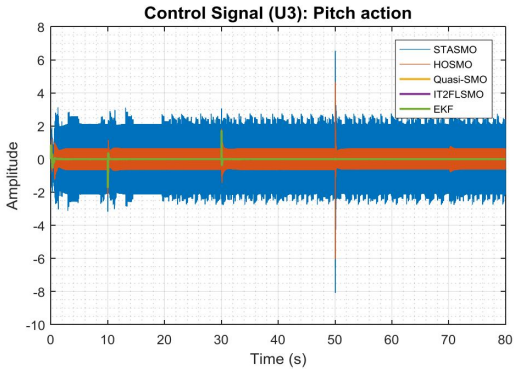

Figure 7.44. Control signal u_3

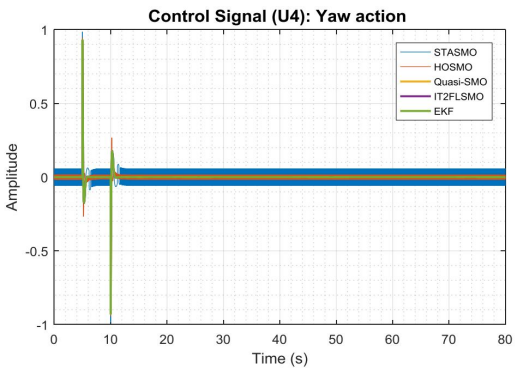

Figure 7.45. Control signal u_4

Further evaluation was carried out by measuring steady-states errors of output states of quadcopter UAV relating to the use of several observer techniques. Similar to the first condition, QuasiSMO outperformed other observers in enhancing the performance of IT2FSMC. It can be seen in Table 7.6 that by employing QuasiSMO, the MSE of steady-state errors of the output states was smaller than other observers except for yaw and z motion for which HOSMO and EKF outperformed others.

Table 7.6. Mean squared errors of steady-states errors

Observer methods	**MSE steady-state error of**					
	x	**pitch**	**y**	**roll**	**yaw**	**z**
QuasiSMO	5.52471E-11	1.00049E-12	5.1726E-11	9.15602E-11	5.33884E-08	8.93841E-10
IT2FSMO	1.33953E-10	2.72661E-12	1.55599E-10	9.31348E-11	8.58659E-08	9.02465E-10
STASMO	0.000118959	0.000150467	0.000114322	0.000119048	5.47741E-08	1.93307E-08
HOSMO	1.26582E-07	1.97406E-07	1.6713E-07	2.44952E-07	1.59769E-09	5.09605E-09
EKF	6.60559E-11	7.55064E-10	7.73953E-11	1.22817E-10	6.31791E-08	7.53523E-10

In summary, in the presence of parameters mismatch, IT2FSMC still shows good performance in controlling the dynamics of the quadcopter UAV with no response overshoot, small rise time by using STASMO and HOSMO in yaw movement, free from chattering when using QuasiSMO, IT2FSMO and EKF observer methods, small steady-states errors. However, computational time is the issue with this control method. On the other hand, QuasiSMO method still outperforms other methods by showing smooth estimated states, smaller states estimation errors, and faster in tracking true states. Therefore, the observer has influenced and improved the overall performance of selected controller.

7.2.3 With noise and no parameter mismatch

The performance of selected controller IT2FSMC is evaluated in dealing with high power white noise disturbance introduced into the output states of the quadcopter UAV. A set of results acquired is shown in Figures 7.46 – 7.67 relating to the use of several observer methods.

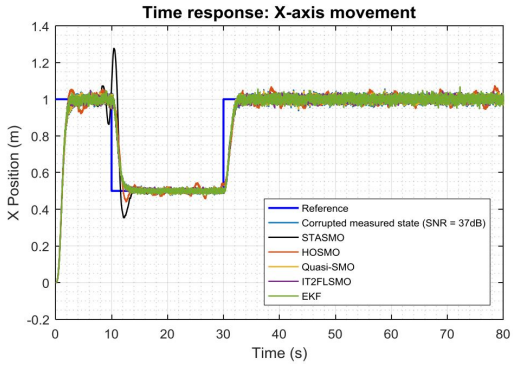

Figure 7.46. x-axis movement

Figure 7.47. Rate of x-axis movement

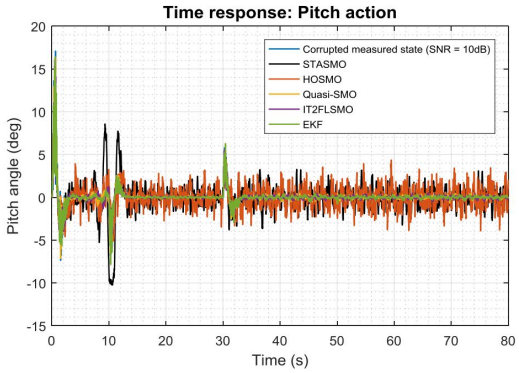

Figure 7.48. Pitch action

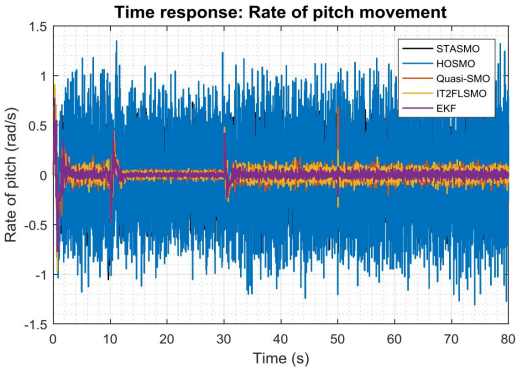

Figure 7.49. Rate of pitch movement

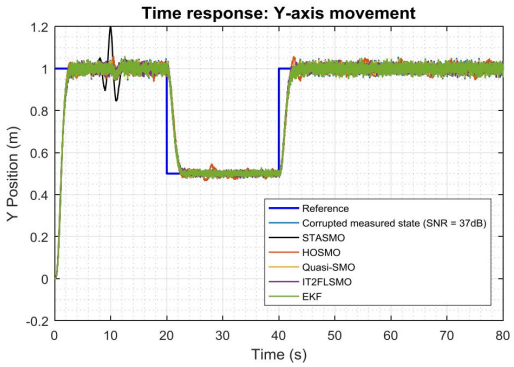

Figure 7.50. y-axis movement

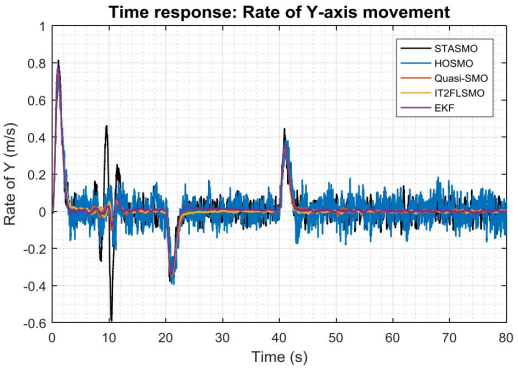

Figure 7.51. Rate of y-axis movement

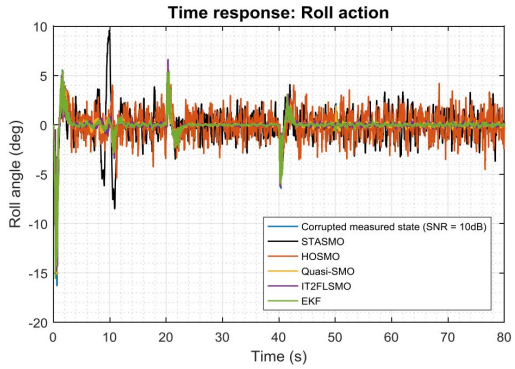

Figure 7.52. Roll action

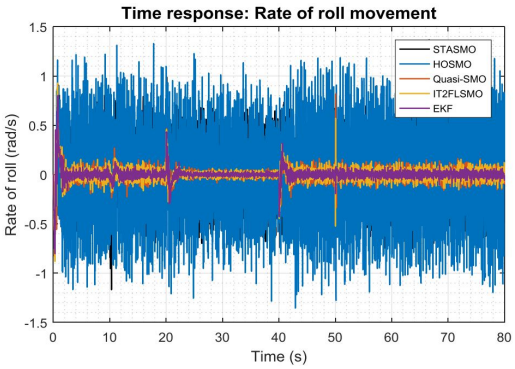

Figure 7.53. Rate of roll movement

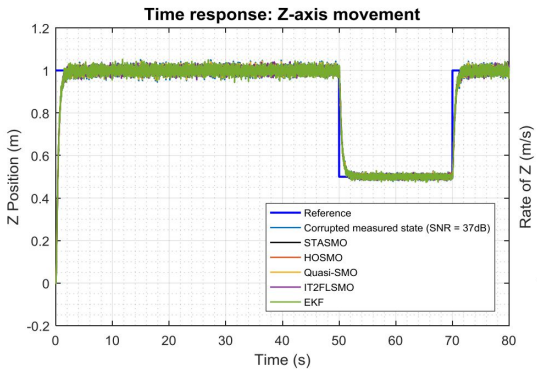

Figure 7.54. z-axis movement

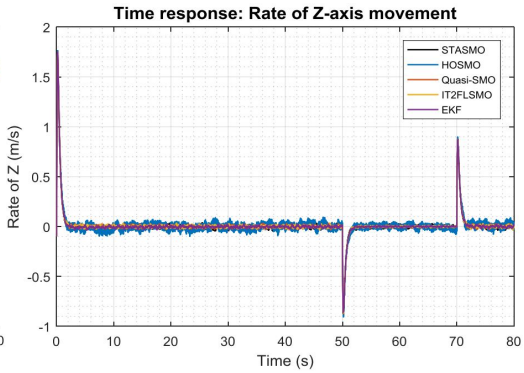

Figure 7.55. Rate of z-axis movement

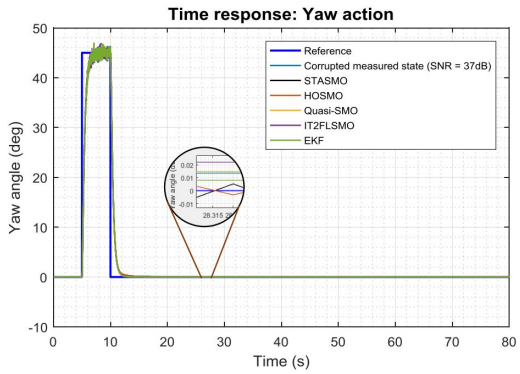

Figure 7.56. Yaw action

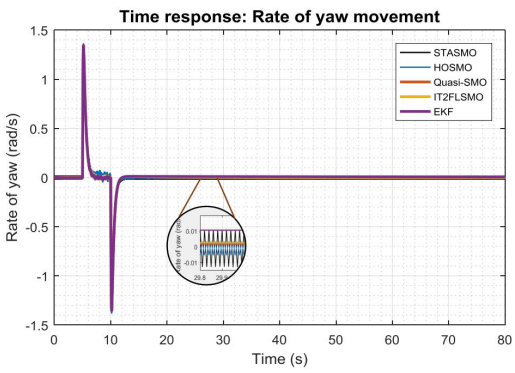

Figure 7.57. Rate of yaw movement

As noted in Figures 7.47, 7.49, 7.51, 7.53, 7.55, and 7.57, the effect of noise was reduced sufficiently by QuasiSMO, IT2FSMO, and EKF so that second order states were generated quite smoothly. Noise disturbance on the output states did not affect significantly the performance of the observers. In contrast, corrupted output states influenced the performances of STASMO and HOSMO quite significantly, especially in estimating second order state of pitch and roll as depicted in Figures 7.49 and 7.53. The noise on the output states was amplified high enough due to the effect of switching functions on the methods.

Although the SNR on the output states x and y was the same as yaw and z states, that is 37dB where it was smaller than the noise on the pitch and roll states, that is 10dB, the effect of noise received by x and y was greater than yaw and z, as can be seen in Figures 7.47, 7.51, 7.55, and 7.57. This is because x and pitch, y and roll are dependent states which will affect each other.

Table 7.7 presents the MSE of estimated states calculated in steady-states condition. It is noted that EKF estimated the second order of pitch and roll rate quite well whereas QuasiSMO and IT2FSMO estimated the states of x, y, z, and yaw rates quite accurately by showing smaller MSE for those states. However, a careful observation reveals that, QuasiSMO has achieved better performance than others by small MSE of estimated states.

Table 7.7. Mean squared errors of estimated states

Observer methods	**MSE estimated state of**					
	x rate	**pitch rate**	**y rate**	**roll rate**	**yaw rate**	**z rate**
QuasiSMO	2.38778E-05	0.002259182	3.26696E-05	0.002590403	1.5993E-06	8.38021E-05
IT2FSMO	2.50844E-05	0.001969311	1.73434E-05	0.002038015	3.06507E-06	8.8796E-05
STASMO	0.00143613	0.093098926	0.001842389	0.097381193	0.000137971	0.00025662
HOSMO	0.003291679	0.230352726	0.003186477	0.242077495	2.07475E-05	0.001126682
EKF	3.64674E-05	0.000738315	2.29225E-05	0.000733391	7.28828E-05	8.63146E-05

In the presence of noise, true states tracking errors appeared similar for all observer methods as depicted in Figures 7.58 - 7.63. However, based on computed MSE as presented in Table 7.8, each observer technique showed different performance in tracking true states. Although not able to estimate second order states as good as QuasiSMO, IT2FSMO and EKF; STASMO and HOSMO showed better performance in tracking true states. It is noticeable in Table 7.8 that STASMO and HOSMO had smaller MSE than other approaches in tracking the measured states of x, y, yaw and z, while QuasiSMO and EKF showed better performance in tracking roll and pitch states respectively. However, STASMO and HOSMO did not perform quite well in estimating second order states of the system compared to other methods so that the overall control performance would be affected.

Figure 7.58. Estimation error of x

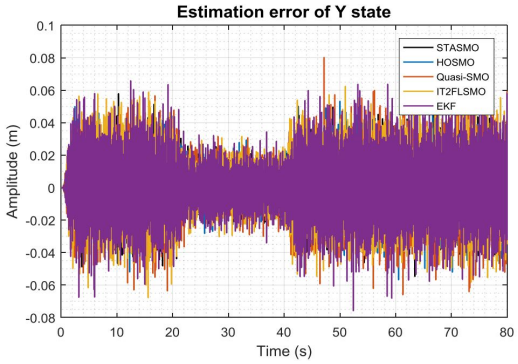

Figure 7.59. Estimation error of y

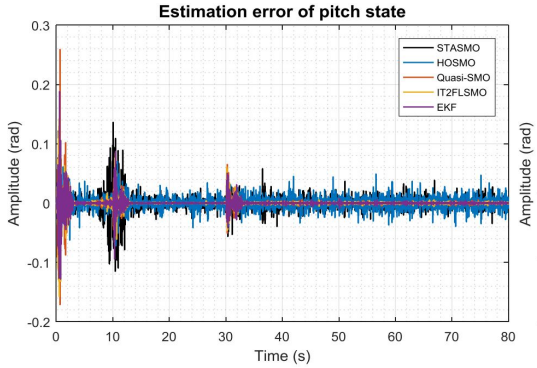

Figure 7.60. Estimation error of pitch

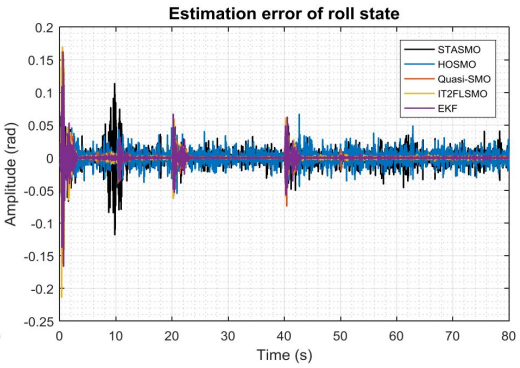

Figure 7.61. Estimation error of roll

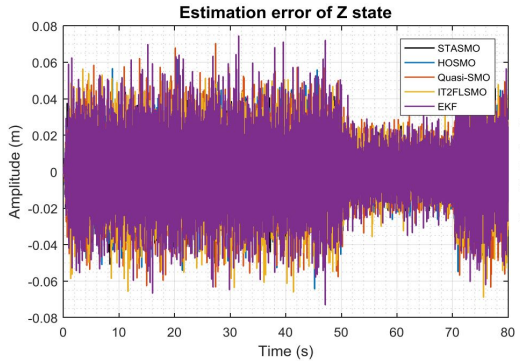

Figure 7.62. Estimation error of z

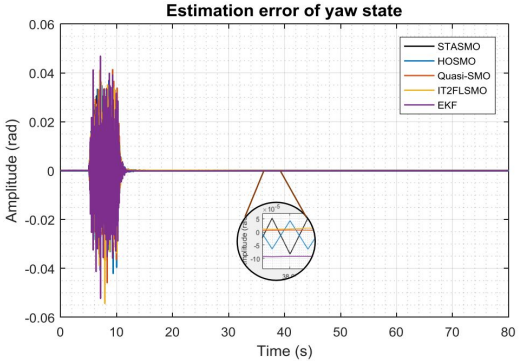

Figure 7.63. Estimation error of yaw

Table 7.8. Mean squared errors of true states tracking

Observer method	MSE true state tracking of					
	x	pitch	y	roll	yaw	z
QuasiSMO	0.00028229	6.22855E-05	0.000272046	5.18932E-05	1.22901E-05	0.000279554
IT2FSMO	0.000277683	5.3681E-05	0.000276559	5.88667E-05	1.24873E-05	0.000285219
STASMO	0.000174383	0.000122361	0.000174343	0.00011096	7.43237E-06	0.00017616
HOSMO	0.000186913	8.30383E-05	0.000181357	8.1605E-05	8.86969E-06	0.00019275
EKF	0.000303853	4.97156E-05	0.000309327	5.40894E-05	1.38318E-05	0.000339493

The presence of noise will certainly affect the performance of controller. However, the effect of noise on systems depends on how to manage and select control and observer systems. Note in Figures 7.46, 7.48, 7.50, 7.52, 7.54, and 7.56 that the use of

STASMO and HOSMO as states estimator had much effect on the control performance in controlling pitch, roll, x, and y. There were significant overshoot and undershoot in x-axis movement when at change of references around maximum 27.8% and 6.0% for STASMO and HOSMO respectively, meanwhile in y-axis movement the overshoots were around 19.8% and 5.75% for STASMO and HOSMO respectively. Higher overshoot was noticeable clearly in controlling pitch by around 8.57° and 3.26° for the use of STASMO and HOSMO respectively, while in roll state the overshoot is about 9.57° and 4.05° for STASMO and HOSMO respectively.

In contrast, the selected controller performed more robust in controlling all output states when QuasiSMO, IT2FLO, and EKF were employed. However, the effect of noise was still noticeable clearly around estimated states of true states and control inputs as seen in Figures 7.64 - 7.67. This is the trade-off of choosing parameters for the observers. On the one hand, increasing or decreasing observer gains will speed up tracking true value, yet may potentially increase sensitivity to measurement noise and vice versa. In this research, the speed of tracking measured states is more considered (without neglecting noise effect) to make the controller more responsive.

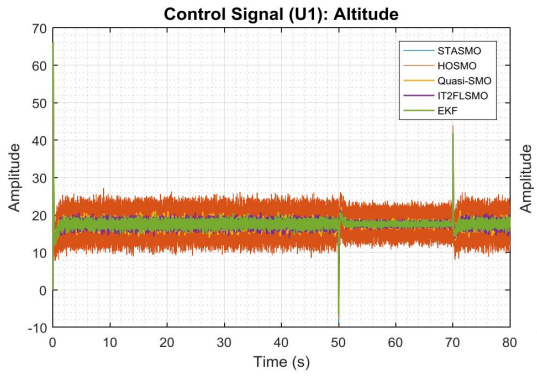

Figure 7.64. Control signal u_1

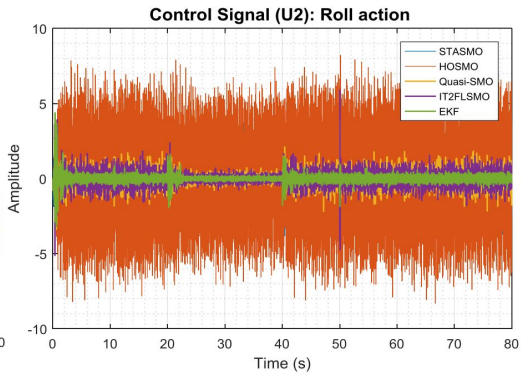

Figure 7.65. Control signal u_2

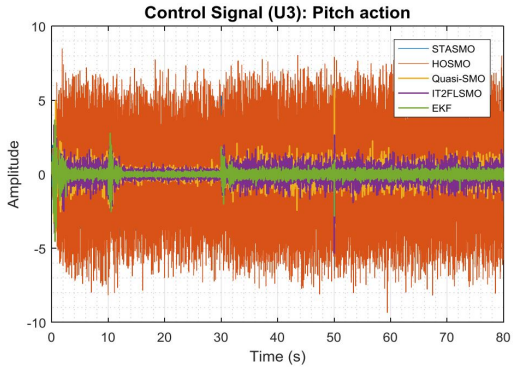

Figure 7.66. Control signal u_3

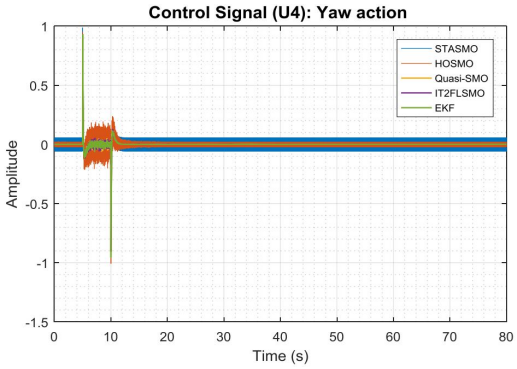

Figure 7.67. Control signal u_4

Furthermore, Table 7.9 shows steady-states error of output states in the presence of noise disturbance. It is noted that, although STASMO and HOSMO had problem in handling noise effect in estimating unmeasured states, the observers influenced control system to reduce steady-state error quite well. STASMO and HOSMO mostly had smaller steady-state error (except roll and pitch) computed by MSE method than other methods. Switching function in the algorithms had the main role of keeping observed state tracking measured state all the time. However, due to noise disturbance, the control signals produced are not suitable for real-time embedded systems as seen in Figures 7.64 - 7.67.

Table 7.9. Mean squared errors of steady-states errors

Observer methods	**MSE steady-state error of**					
	x	**pitch**	**y**	**roll**	**yaw**	**z**
QuasiSMO	0.000155848	0.000273005	0.000152914	0.000268313	4.25236E-08	0.00015306
IT2FSMO	0.000154799	0.000268148	0.000148609	0.000267108	4.51698E-08	0.000155317
STASMO	7.66478E-05	0.000611054	8.55343E-05	0.000708412	1.10923E-08	1.99439E-05
HOSMO	0.000534733	0.000735355	0.000408658	0.00072766	6.25239E-09	8.20339E-05
EKF	0.000198713	0.000267953	0.000200446	0.000265009	1.01776E-08	0.000209673

In summary, in the presence of noise, IT2FSMC maintains good performance in controlling the dynamics of the quadcopter UAV with no overshoot response except for x and y movements, pitch and roll actions when applying STASMO and HOSMO, considered small rise time, free from chattering when using QuasiSMO, IT2FSMO and EKF observer methods, and small steady-states errors. However, computational time is the constraint of the proposed control method. Furthermore, fluctuated signals randomly on the references still appear due to noise disturbance. This is because the selection of observer gain is being a trade-off. Reducing or increasing the gain may make observers become insensitive to noise, however, on the other hand it may delay tracking true states. In terms of observer,

QuasiSMO method still outperforms other methods by showing smooth estimated states, smaller states estimation errors, and faster in tracking true states. Therefore, the observer has influenced and improved the overall performance of selected controller.

7.2.4 With noise and with parameter mismatch

The last stage of testing the performance of IT2FSMC and observers are introducing noise and uncertainty at the same time to the quadcopter UAV system. The responses of the methods through simulation results are presented in Figures 7.68 - 7.89.

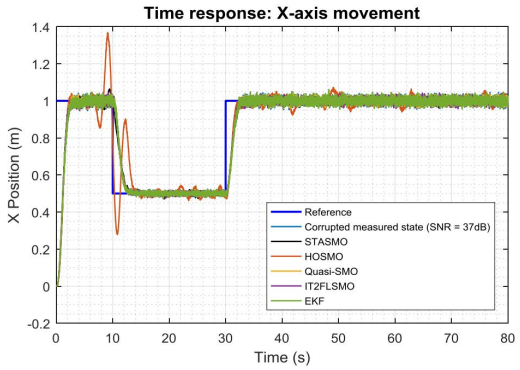

Figure 7.68. x-axis movement

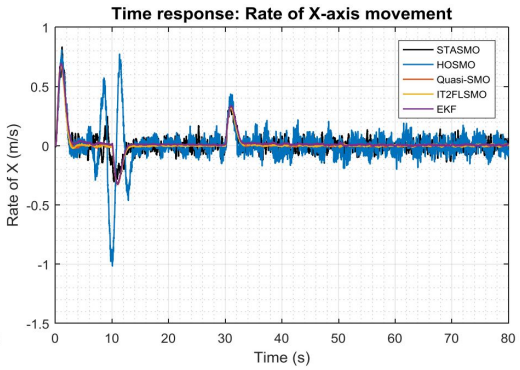

Figure 7.69. Rate of x-axis movement

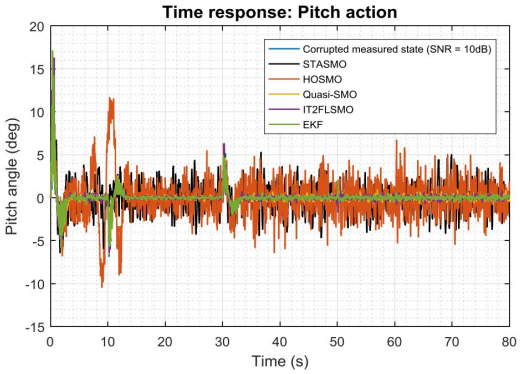

Figure 7.70. Pitch action

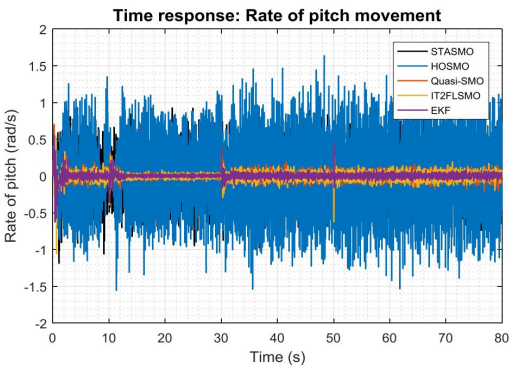

Figure 7.71. Rate of pitch movement

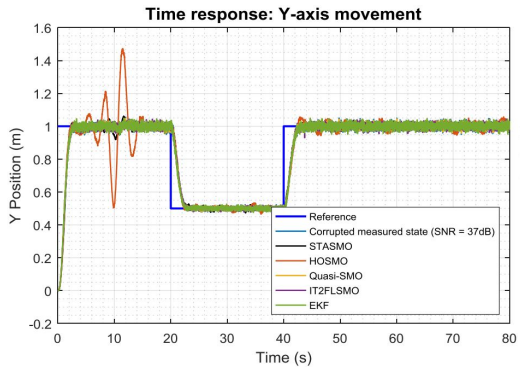

Figure 7.72. y-axis movement

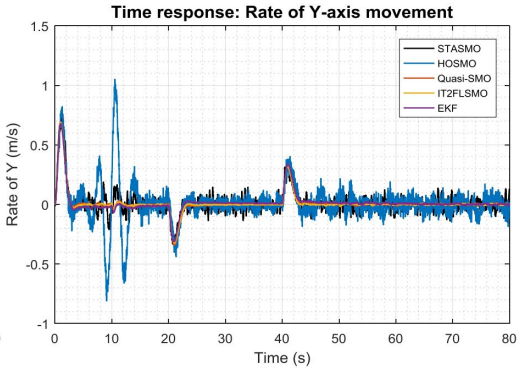

Figure 7.73. Rate of y-axis movement

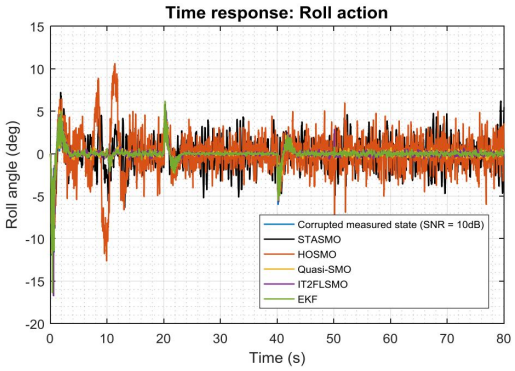

Figure 7.74. Roll action

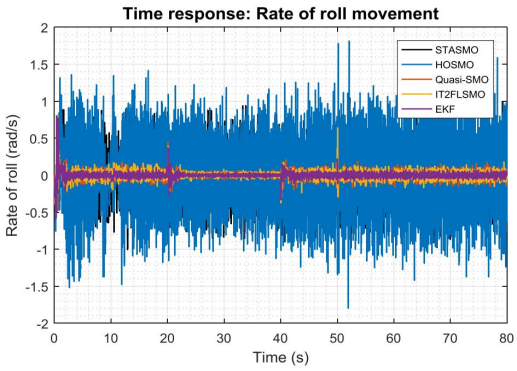

Figure 7.75. Rate of roll movement

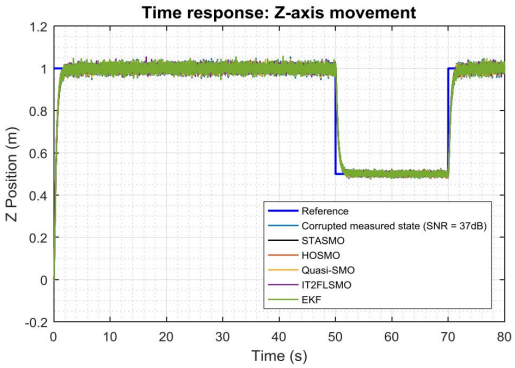

Figure 7.76. z-axis movement

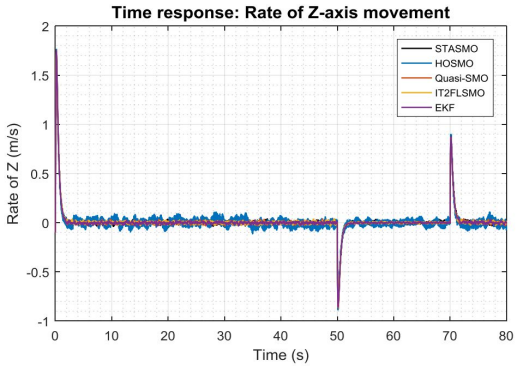

Figure 7.77. Rate of z-axis movement

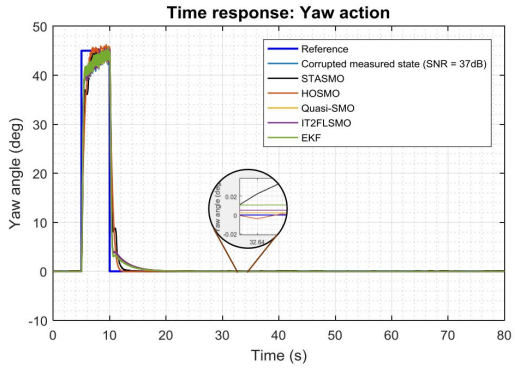

Figure 7.78. Yaw action

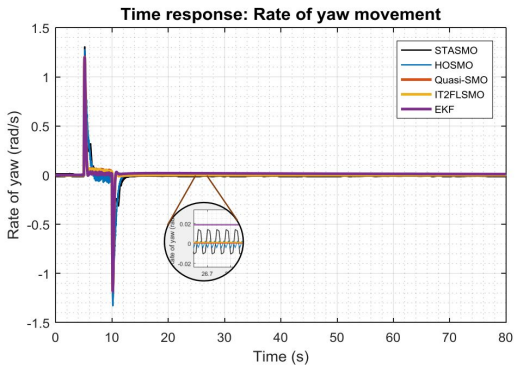

Figure 7.79. Rate of yaw movement

The same white noise power was introduced as in previous test to the output states and 50% parameters mismatch was incorporated into the UAV system. Note in the simulation results presented in Figures 7.69, 7.71, 7.73, 7.75, 7.77, and 7.79, that although in the presence of high power noise and high uncertainty, the effect of such disturbances were reduced adequately by QuasiSMO, IT2FSMO, and EKF so that second order states of the system were generated quite smoothly. In other words, noise disturbance on the output states as well as parameters mismatch did not affect significantly the performance of the observers. In contrast, corrupted output states influenced the performances of STASMO

and HOSMO significantly, especially in estimating second order states of pitch and roll as shown in Figures 7.71 and 7.75. The noise amplitude on the output states were amplified high enough due to the effect of switching term on the methods. In addition, HOSMO could not deal with parameters mismatch well, this is shown with high amplitude oscillation for the estimation of x, and y movement rate as the x state changed.

Although the SNR on the output states x and y were the same as yaw and z states, that is 37dB where it was smaller than the noise on the pitch and roll states, that is 10dB, the effect of noise received by x and y was greater than yaw and z, as can be seen in Figures 7.69, 7.71, 7.73, and 7.75. It is because x and pitch, y and roll are dependent states which will affect each other.

Due to noise disturbance and parameters mismatch, steady-state errors of estimated states increased as seen in Table 7.10. However, it is clearly noticeable that QuasiSMO, IT2FSMO and EKF still had good performance in estimating unmeasured states even in the presence of disturbances. MSE of estimated states computed for those observers, are considered as small values. This means that those observer performed as expected in estimating second order states.

Table 7.10. Mean squared errors of estimated states

Observer methods	**MSE estimated state of**					
	x rate	**pitch rate**	**y rate**	**roll rate**	**yaw rate**	**z rate**
QuasiSMO	2.41843E-05	0.002493785	2.33605E-05	0.002466661	2.90281E-08	0.000101505
IT2FSMO	2.22254E-05	0.002287918	1.86288E-05	0.002401759	3.06749E-07	9.08523E-05
STASMO	0.002047201	0.12161332	0.002353832	0.134846871	0.000139816	0.000240764
HOSMO	0.003776574	0.293485018	0.003530244	0.284777168	1.3223E-05	0.001473934
EKF	6.85287E-05	0.00086798	5.75095E-05	0.00080983	0.000224693	8.60903E-05

In the presence of noise and parameters mismatch, true states tracking errors appeared similar for every observer method as presented in Figures 7.80 - 7.85. However, observing the computed MSE as presented in Table 7.11, each observer technique exhibited different performance in tracking true states. Although not able to estimate second order states as good as QuasiSMO, IT2FSMO and EKF; STASMO and HOSMO showed better performance in tracking the true states. It is noticeable in Table 7.11 that STASMO and HOSMO had smaller MSE in tracking the measured states of x, y, yaw and z, while QuasiSMO and IT2FSMO showed better performance in tracking pitch and roll states. However, STASMO

and HOSMO did not perform quite well in estimating second order states of the system compared to other approaches, hence the overall control performance would be affected.

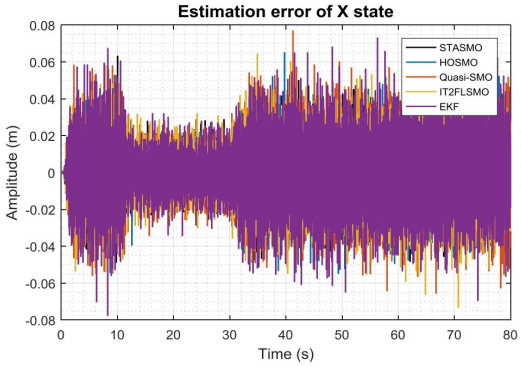

Figure 7.80. Estimation error of x

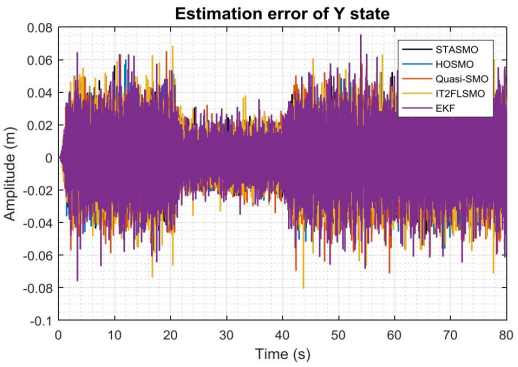

Figure 7.81. Estimation error of y

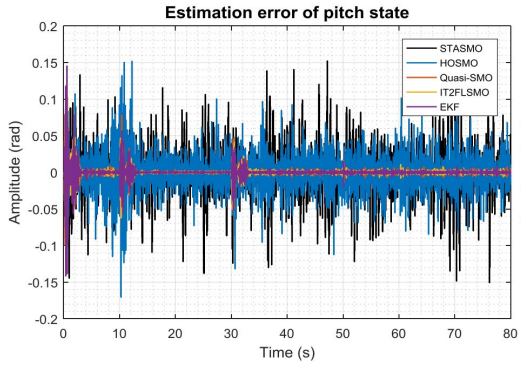

Figure 7.82. Estimation error of pitch

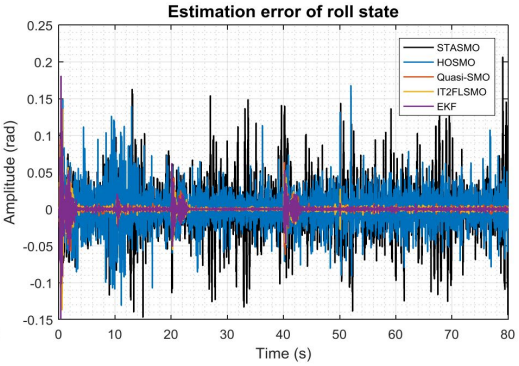

Figure 7.83. Estimation error of roll

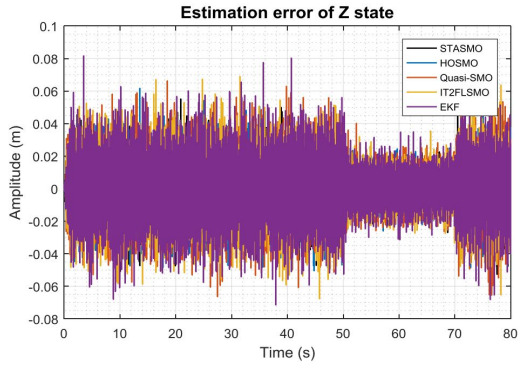

Figure 7.84. Estimation error of z

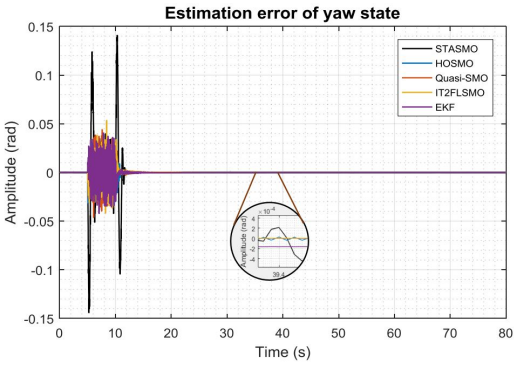

Figure 7.85. Estimation error of yaw

Table 7.11. Mean squared errors of true states tracking

Observer method	MSE true state tracking of					
	x	pitch	y	roll	yaw	z
QuasiSMO	0.000278813	3.51442E-05	0.000277396	4.25992E-05	1.42731E-05	0.000277907
IT2FSMO	0.000273066	3.54466E-05	0.000278449	4.08499E-05	1.27143E-05	0.000285055
STASMO	0.000183609	0.000908805	0.000182696	0.001043624	0.000216499	0.000179644
HOSMO	0.000192673	0.000492755	0.000190455	0.000491228	1.189E-05	0.000194719
EKF	0.000324987	4.71384E-05	0.000322694	4.26852E-05	1.39451E-05	0.000319785

The presence of noise and parameters mismatch might affect the performance of controller. However, the effect of noise on systems depends on how to manage and select control and observer systems. As noted in Figures 7.68, 7.70, 7.72, 7.74, 7.76, and 7.78, the use of STASMO and HOSMO as states estimator had much effect on the control performance in controlling pitch, roll, x, and y. There were significant overshoot and undershoot in x-axis movement especially when references changed around maximum 40.0% and 6.3% for HOSMO and STASMO respectively, meanwhile in y-axis movement the overshoots were around 49.6% and 8.0% for HOSMO and STASMO respectively. In addition to high amplitude random fluctuated states due to noise disturbance, higher response overshoot was also noted in controlling pitch around 11.66° and 6.0° for HOSMO and STASMO respectively, while in roll state the overshoots were 10.6° and 4.5° for HOSMO and STASMO respectively.

In contrast, the selected controller performed in a stable manner in controlling all output states when employing QuasiSMO, IT2FSMO, and EKF. However, the noise effect was still noticeable around estimated states of true state and control inputs as seen in Figures 7.86 - 7.89. This is the trade-off of choosing parameters for the observers. On the one hand, increasing or decreasing observer gains will speed up tracking true value, yet may potentially increase sensitivity to measurement noise and vice versa. In this research, the speed of tracking measured states was considered (without neglecting noise effect) to make the controller more responsive.

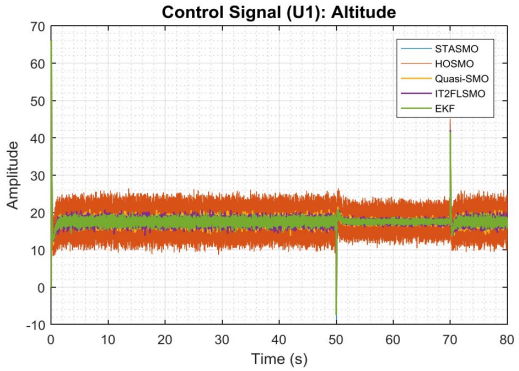

Figure 7.86. Control signal u_1

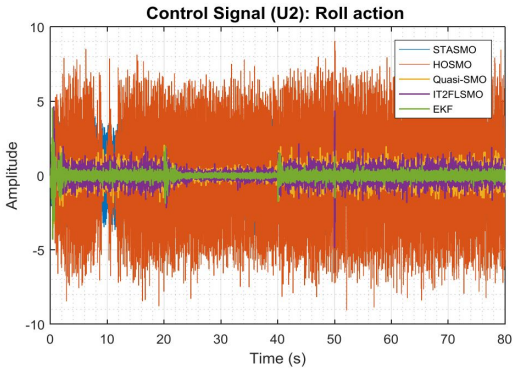

Figure 7.87. Control signal u_2

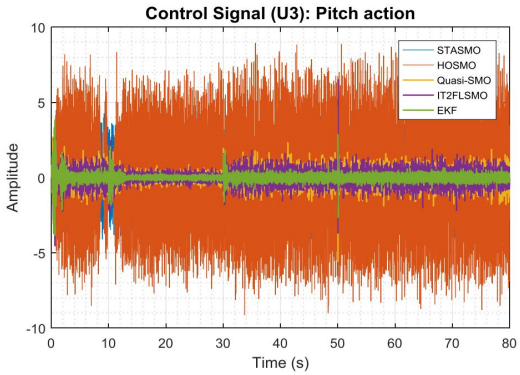

Figure 7.88. Control signal u_3

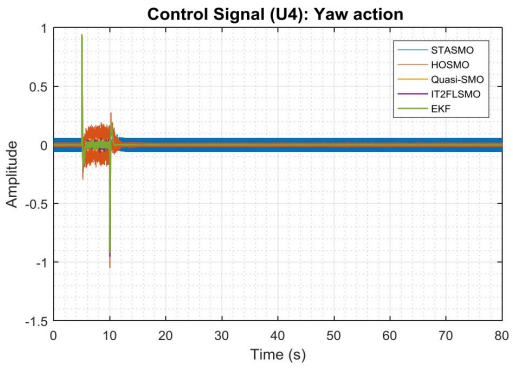

Figure 7.89. Control signal u_4

Steady-states error of the output states was considered to evaluate further the robustness of selected controller and observers in dealing with noise disturbance and parameters mismatch. As noted in Table 7.12, in the presence of such issues, in general, steady-states errors were small. It means that even though noise and parameters mismatch placed the quadcopter system in bad condition, the observers influenced control system to stabilize the quadcopter UAV and reduce steady-state error quite well. In this case, STASMO observer method performed better than the other estimators by showing smaller steady-state error in almost all states computed by MSE method. However, due to noise disturbance, the use of STASMO and HOSMO made the control signals produced not suitable for real-time embedded systems. In addition, the step response of the controller was not smooth when applying STASMO and HOSMO by demonstrating high overshoot and undershoot when references changed.

Table 7.12. Mean squared errors of steady-states errors

Observer methods	**MSE steady-state error of**					
	x	**pitch**	**y**	**roll**	**yaw**	**z**
QuasiSMO	0.000157397	0.000273325	0.000157207	0.000263628	5.4305E-08	0.000151842
IT2FSMO	0.000156073	0.000267858	0.000153038	0.00025932	6.51513E-08	0.000151393
STASMO	0.00014284	0.001173998	0.000147351	0.001302302	3.3771E-07	1.76463E-05
HOSMO	0.000557031	0.001214822	0.000434513	0.001168721	4.97105E-09	0.00012142
EKF	0.000203475	0.000269616	0.000207877	0.000258611	6.82787E-08	0.000205464

In summary, in the presence of noise and parameter mismatch, IT2FSMC maintains good performance in controlling the dynamics of the quadcopter UAV with no response overshoot except for x and y movements, pitch and roll actions when employing STASMO and HOSMO, considered small rise time, free from chattering when using QuasiSMO, IT2FSMO and EKF observer method, and small steady-states errors. However, computa-

tional time is still an issue with applying selected controller. In addition, fluctuated signals randomly on the references still appear due to noise disturbance. The selection of observer gain is a trade-off. Reducing or increasing the gain may make observers become insensitive to noise, however, on the other hand it may delay tracking true states. The QuasiSMO method still outperforms other methods by showing smooth estimated states, smaller states estimation errors, and faster in tracking true states. Therefore, the observer has influenced and improved the overall performance of selected controller.

7.3 Performance summary

After conducting performance evaluation of the control system and the observer methods with 4 predefined scenarios, it can be summarized that set-point integral sliding mode-based interval type-2 fuzzy control (IT2FSMC) can still maintain the stability of its performance to control the dynamics of quadcopter UAV well in all conditions. Integral term plays an important role in reducing steady-state error of the output states while set-point weighting function plays a role in reducing overshoot due to the integral term. Oscillations on x, y, pitch and roll motions only occur in the scenario of with noise and parameter mismatch (scenario 2-4) when STASMO and HOSMO observer methods were employed. The disadvantage of the control system is slow computation time to generate control inputs due to type reduction process in the type-2 fuzzy logic method. In terms of observer, actually QuasiSMO and IT2FSMO have similar performance in estimating unmeasured states. However, the same as IT2FSMC, IT2FSMO has slower computational time than QuasiSMO due to the same problem as IT2FSMC. Hence, generally QuasiSMO outperformed than others and has contributed to improved control system performance in dealing with all predefined conditions

Chapter 8

Numerical simulations: set-point integral super-twisting algorithm of sliding mode control

8.1 Introduction

This chapter presents numerical simulation results to highlight the performance of set-point integral super-twisting algorithm of sliding mode control (STASMC) with the developed nonlinear full-order state observers, namely:

- quasi - sliding mode observer
- sliding mode-based interval type-2 fuzzy observer
- super-twisting algorithm of sliding mode observer
- higher order sliding mode observer
- extended Kalman filter

Performance comparison between sliding mode-based observers and extended Kalman filter (EKF) is conducted to highlight the robustness of each method. Through numerical validation, the performance of the control system and the performance comparisons of the observer methods are carried out and presented in this chapter.

The overall control and observer system diagram, simulation setup and performance criteria are the same as in Chapter 6. In addition, the same scenerios as in Chapter 6 are introduced to test the performance of the control and observer methods, including:

- without noise and no parameter mismatch
- without noise and with parameter mismatch
- with noise and no parameter mismatch
- with noise and with parameter mismatch

The purpose of this approach is to determine the ability of the controller and observer to deal with several conditions.

The reference frame of the quadcopter movements in this numerical simulation can be seen in Figure 8.1

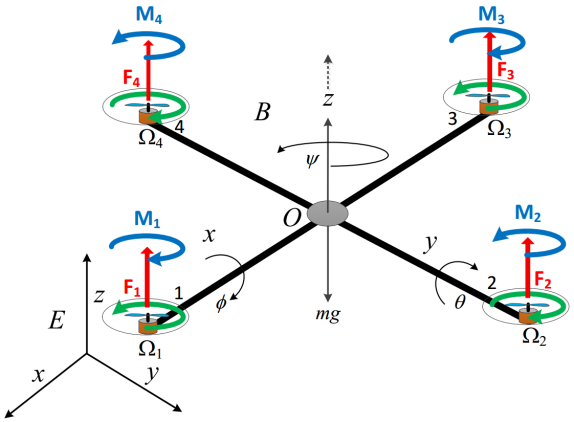

Figure 8.1. Reference frame of quadcopter movements

where x, y, and z represent the movements of the quadcopter in those axes, while ϕ, θ, and ψ represent roll, pitch, and yaw motions of the quadcopter respectively.

8.2 Simulation results

After a series of investigations, the following sliding mode parameters were found to be suitable for the desired performance achievement.

$$\lambda_\phi = 16.5 \quad \lambda_\theta = 16.5 \quad \lambda_\psi = 3.2935 \quad \lambda_x = 2.6599 \quad \lambda_y = 2.6599 \quad \lambda_z = 3.3950$$
$$\lambda_{i\phi} = 30.0 \quad \lambda_{i\theta} = 30.0 \quad \lambda_{i\psi} = 1.4 \quad \lambda_{ix} = 1.82 \quad \lambda_{iy} = 1.82 \quad \lambda_{iz} = 1.45$$
$$\kappa_\phi = 10.0 \quad \kappa_\theta = 10.0 \quad \kappa_\psi = 5.0 \quad \kappa_x = 2.0 \quad \kappa_y = 2.0 \quad \kappa_z = 5.0$$

8.2.1 Without noise and no parameter mismatch

The simulation results obtained in this case are depicted in Figures 8.2 – 8.23.

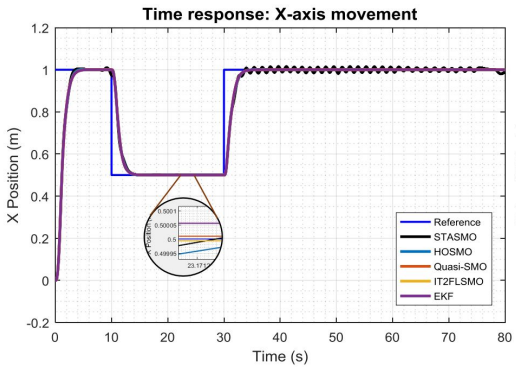

Figure 8.2. x-axis movement

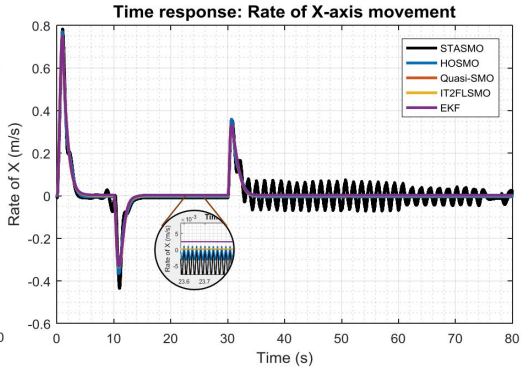

Figure 8.3. Rate of x-axis movement

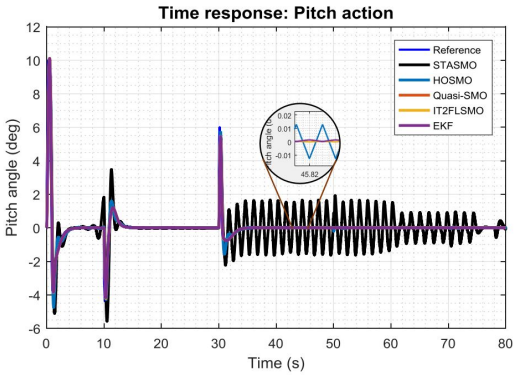

Figure 8.4. Pitch action

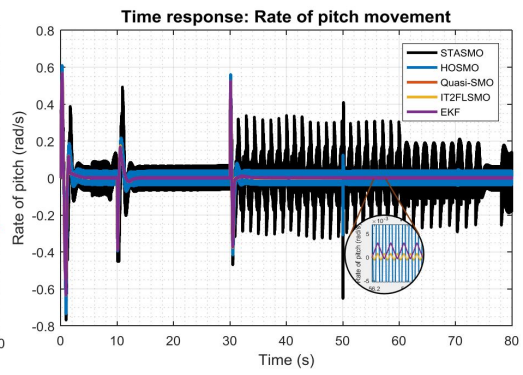

Figure 8.5. Rate of pitch movement

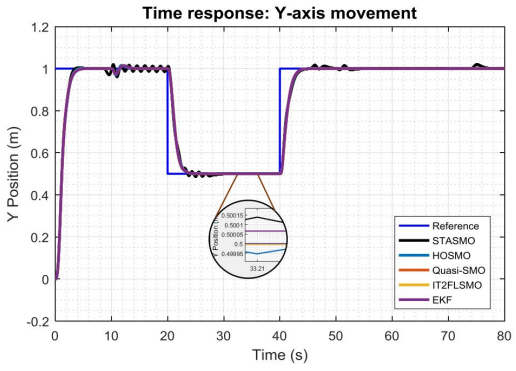

Figure 8.6. y-axis movement

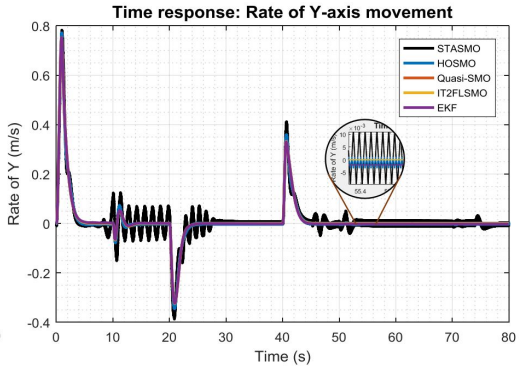

Figure 8.7. Rate of y-axis movement

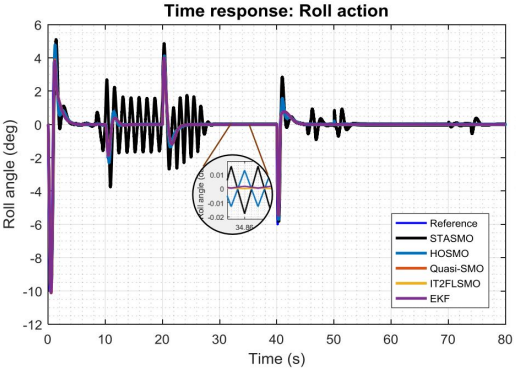

Figure 8.8. Roll action

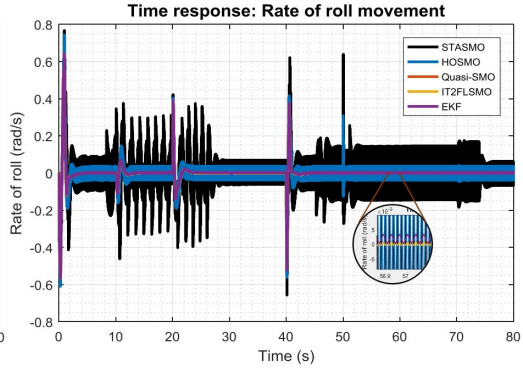

Figure 8.9. Rate of roll movement

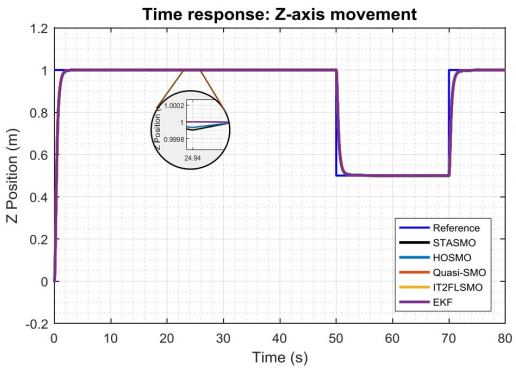

Figure 8.10. z-axis movement

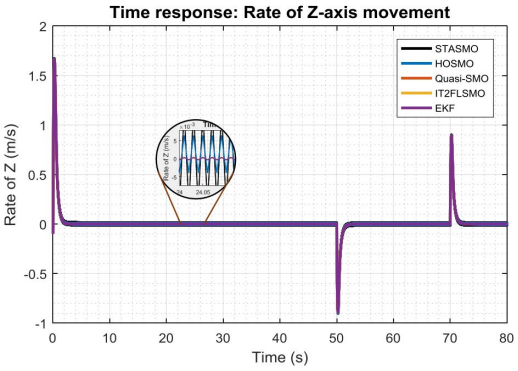

Figure 8.11. Rate of z-axis movement

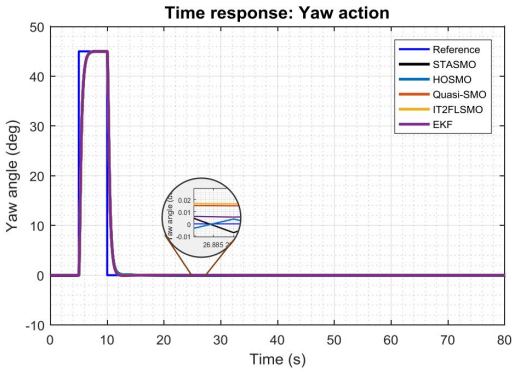

Figure 8.12. Yaw action

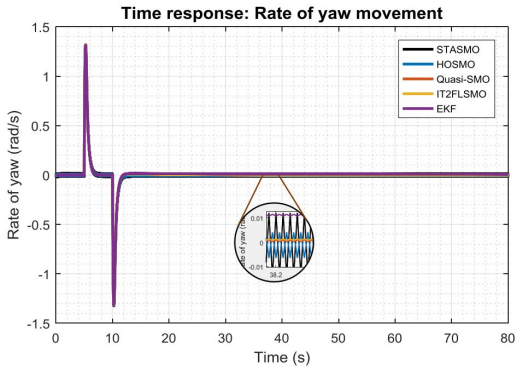

Figure 8.13. Rate of yaw movement

It is noted in the simulation results in Figures 8.3, 8.5, 8.7, 8.9, 8.11, and 8.13, that most of the observers performed well to estimate second order states of the quadcopter UAV except STASMO method especially in estimating rate of x, y - axes movements as well as pitch and roll motions. In addition to the chattering phenomenon caused by switching term in the observer method, significant oscillations were clearly visible in the estimated states. In previous selected controllers QuasiSMC and IT2FSMC, STASMO and HOSMO also showed unsatisfied performance in estimating the second order states of roll and pitch by showing significant chattering in the estimated states. The chattering phenomenon and dependent states of x and pitch as well as y and roll make the states estimation process of unmeasured states affect each other so that imperfect estimation results of the second order of those states will always be visible.

The MSE of estimated states in steady-state condition using STASMC is presented in Table 8.1. As noted, QuasiSMO, IT2FSMO and EKF have smaller MSE for every estimated states. They performed quite well. However, QuasiSMO showed more stable performance than other observer methods by presenting smallest MSE among others as

seen in the table.

Table 8.1. Mean squared errors of estimated states

Observer methods	MSE estimated state of					
	x rate	pitch rate	y rate	roll rate	yaw rate	z rate
QuasiSMO	7.24898E-10	6.02997E-07	1.26132E-09	6.0674E-07	1.90458E-06	1.30465E-07
IT2FSMO	1.85708E-09	6.10293E-07	2.10246E-09	6.12838E-07	3.51474E-06	1.30485E-07
STASMO	0.001108163	0.012892968	0.000112158	0.009765464	0.000169024	0.000169405
HOSMO	6.48735E-06	0.001377853	6.49962E-06	0.001392873	2.89284E-05	2.87166E-05
EKF	3.57044E-06	2.82689E-06	4.53367E-06	3.90637E-06	0.000102171	1.28176E-07

Furthermore, in addition to estimating unmeasured states, each observer showed different performance in tracking true states as seen in Figures 8.14 - 8.19. QuasiSMO and IT2FSMO showed good performance in tracking true states by achieving small estimation errors in steady state condition, whereas EKF performed better in states change condition by showing smaller estimation error. For providing better assessment, MSE calculation method was utilized to measure the quality of observers. The computed MSE is presented in Table 8.2. In the condition of without noise and no parameters mismatch, the two observers namely QuasiSMO and IT2FSMO outperformed other methods by exhibiting smaller MSE for all states.

Table 8.2. Mean squared errors of true states tracking

Observer method	MSE true state tracking of					
	x	pitch	y	roll	yaw	z
QuasiSMO	4.24659E-10	1.5885E-11	4.20319E-10	8.29605E-12	1.66524E-09	5.00184E-12
IT2FSMO	4.36769E-10	2.09106E-11	4.28966E-10	1.00113E-11	2.0713E-09	5.07911E-12
STASMO	6.81699E-09	5.62301E-09	7.28629E-09	7.3012E-09	5.45659E-09	4.90067E-09
HOSMO	3.00224E-09	2.97621E-09	3.00473E-09	2.97522E-09	2.97365E-09	2.97573E-09
EKF	6.48537E-10	2.68967E-09	4.39563E-10	2.77565E-09	1.33174E-08	4.82026E-09

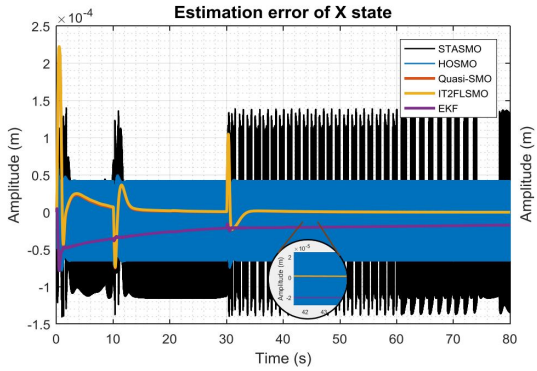

Figure 8.14. Estimation error of x

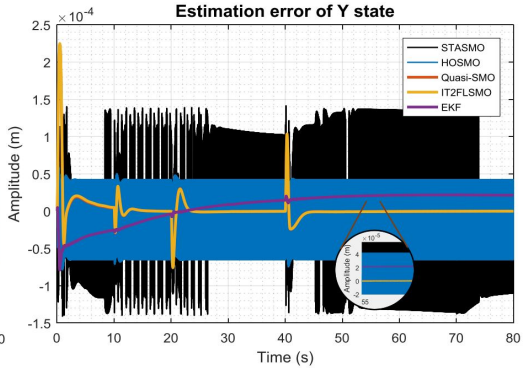

Figure 8.15. Estimation error of y

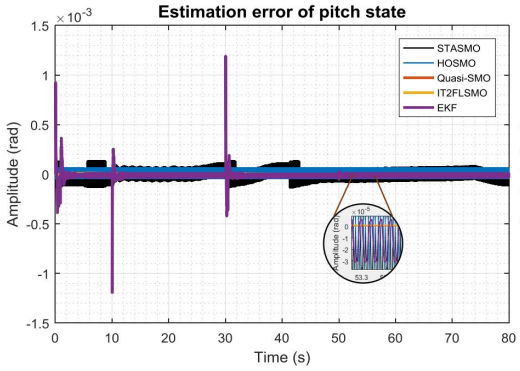

Figure 8.16. Estimation error of pitch

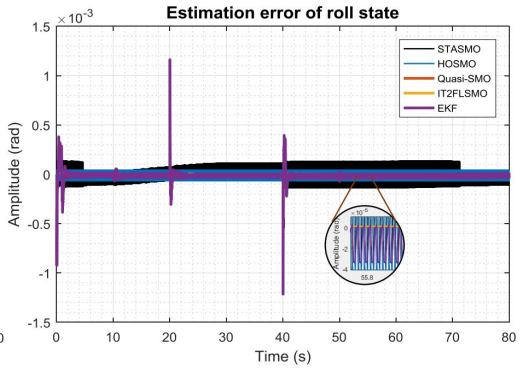

Figure 8.17. Estimation error of roll

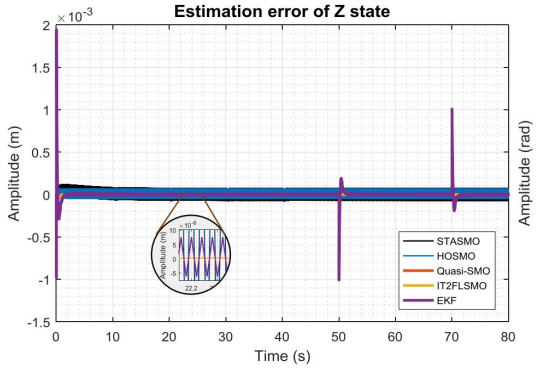

Figure 8.18. Estimation error of z

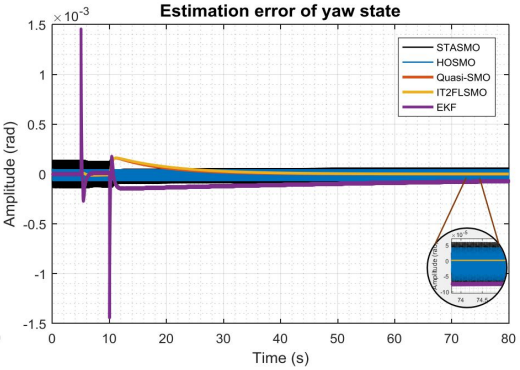

Figure 8.19. Estimation error of yaw

It can be seen in Figures 8.2, 8.4, 8.6, and 8.8 that there were oscillations in controlling pitch and roll of around $\pm 1.92°$ and $\pm 2.7°$ respectively. Therefore, due to coupled states, step responses for x and y axes movements were affected by showing small oscillation around $\pm 1.5cm$ and $\pm 1.4cm$ respectively. However, the controller performed well in z-axis movement and yaw action.

Generally, the time demand to process the sequence of controller method was quite short around 0.6350 ms, and the controller showed good performance in controlling the quadcopter when QuasiSMO, IT2FSMO, HOSMO, and EKF were employed. It can be noticed in the figures that the controller responded to step inputs with no overshoot and with acceptable rise time. However, high frequency oscillations were still seen in the control inputs using STASMO and HOSMO as can be seen in Figures 8.20 - 8.23, whereas QuasiSMO, IT2FSMO and EKF generated smoother control signals for the aircraft which is good for real-time application.

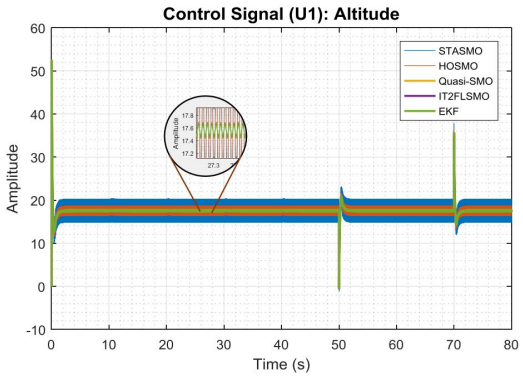

Figure 8.20. Control signal u_1

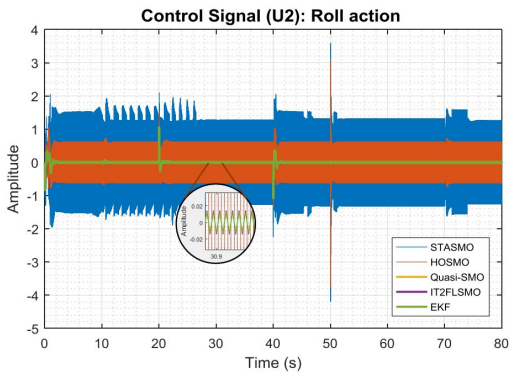

Figure 8.21. Control signal u_2

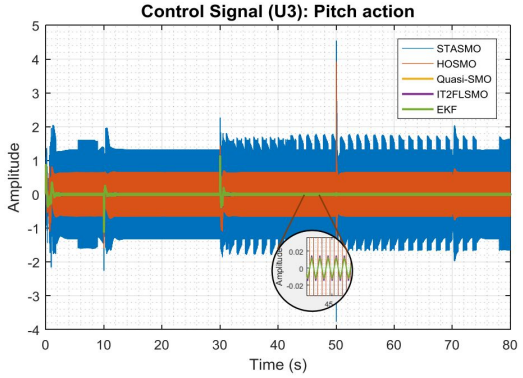

Figure 8.22. Control signal u_3

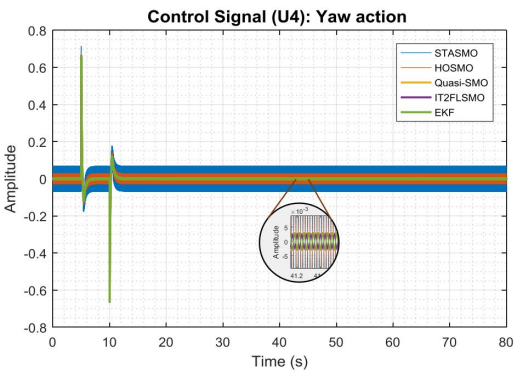

Figure 8.23. Control signal u_4

Table 8.3 presents MSE of output states of the quadcopter UAV in steady-state condition. Generally, QuasiSMO and IT2FSMO have outperformed other observers in improving the performance of selected control system by displaying smaller MSE in almost all output states except y and yaw motion for which EKF performed better than others.

Table 8.3. Mean squared errors of steady-states errors

Observer methods	**MSE steady-state error of**					
	x	**pitch**	**y**	**roll**	**yaw**	**z**
QuasiSMO	1.42914E-11	1.57624E-11	1.84017E-10	1.14073E-10	2.32891E-08	1.20287E-10
IT2FSMO	1.7754E-11	1.7091E-11	1.86648E-10	1.1605E-10	2.57844E-08	1.21411E-10
STASMO	7.19112E-05	0.000215364	1.88767E-05	7.46548E-06	1.13855E-08	1.49778E-08
HOSMO	1.08052E-08	6.95205E-08	1.3457E-08	7.49636E-08	4.70825E-09	4.70438E-09
EKF	1.02605E-10	4.81379E-10	1.69105E-10	7.71068E-10	6.79887E-09	1.85467E-10

In summary, in the condition of free of noise and no parameters mismatch, STASMC has shown good performance in controlling the dynamics of the quadcopter UAV with no response overshoot, small rise time, free from chattering when QuasiSMO, IT2FSMO and EKF observer methods were utilized, small steady-states errors, and low computational time. In terms of observer, QuasiSMO method has outperformed other methods by exhibiting smooth estimated states, smaller states estimation errors, and faster in tracking

true states. Therefore, the observer has influenced and improved the overall performance of selected controller.

8.2.2 Without noise and with parameter mismatch

The robustness of STASMC was tested through simulation in this condition. The results obtained are presented in Figures 8.24 – 8.25.

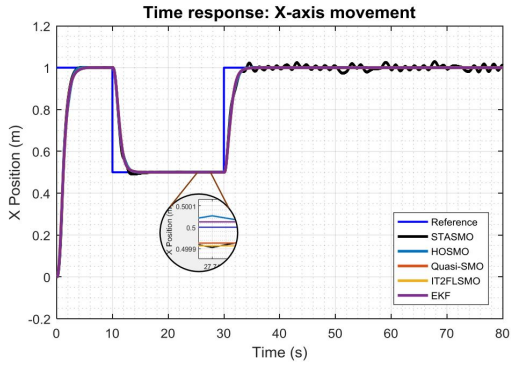

Figure 8.24. x-axis movement

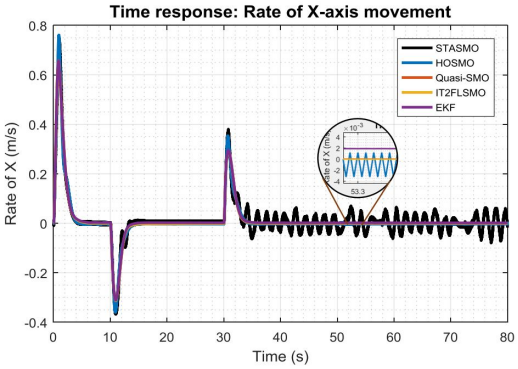

Figure 8.25. Rate of x-axis movement

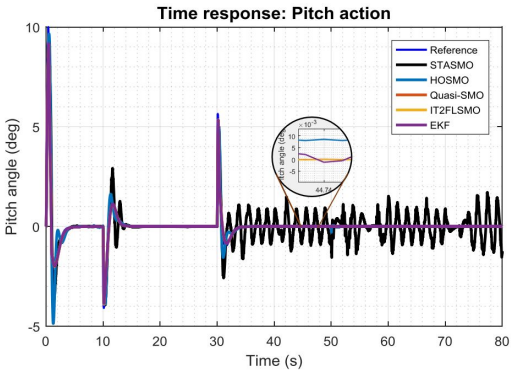

Figure 8.26. Pitch action

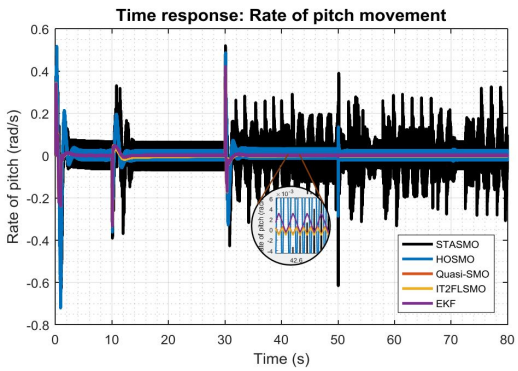

Figure 8.27. Rate of pitch movement

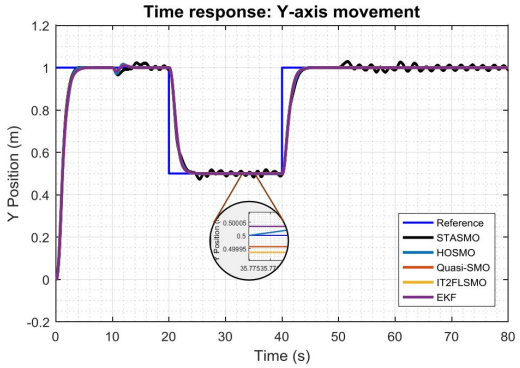

Figure 8.28. y-axis movement

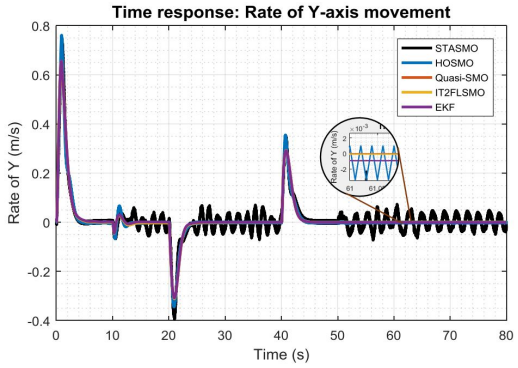

Figure 8.29. Rate of y-axis movement

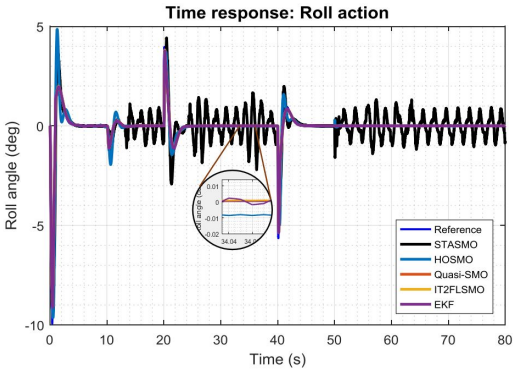

Figure 8.30. Roll action

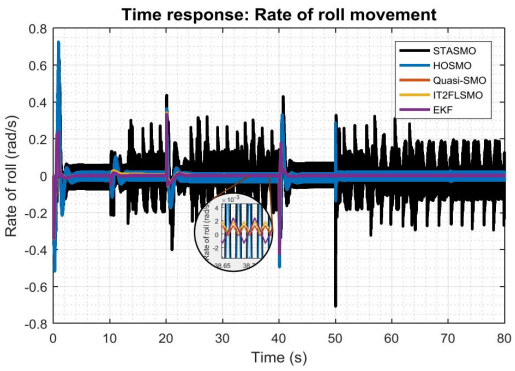

Figure 8.31. Rate of roll movement

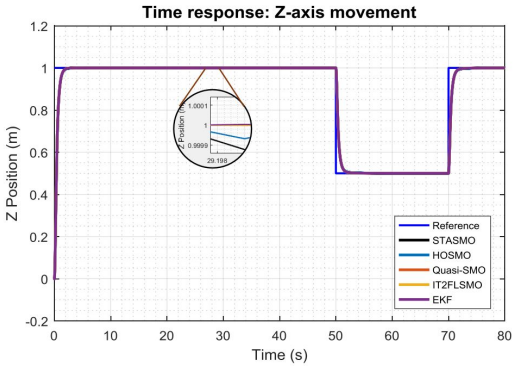

Figure 8.32. z-axis movement

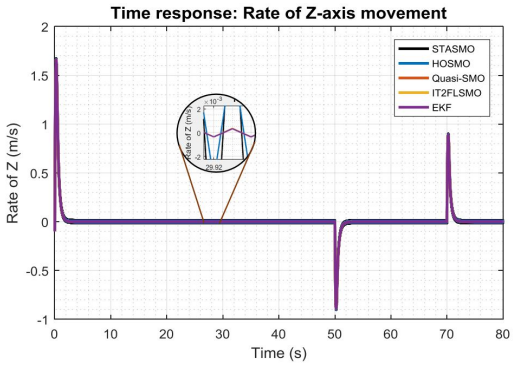

Figure 8.33. Rate of z-axis movement

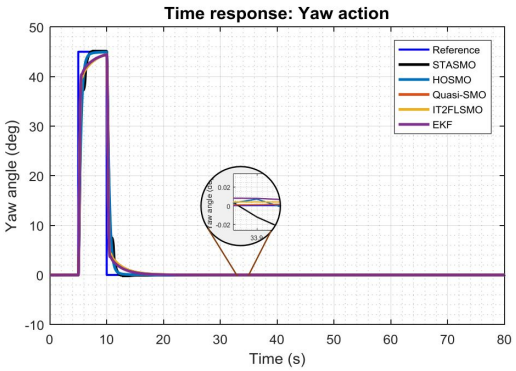

Figure 8.34. Yaw action

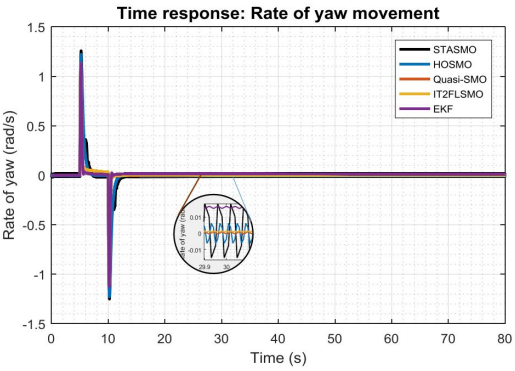

Figure 8.35. Rate of yaw movement

As noted in the simulation results presented in Figures 8.25, 8.27, 8.29, 8.31, 8.33, 8.35, parameters mismatch significantly affected the performance of STASMO in estimating some unmeasured states especially rate of x, y, pitch and roll. In addition to high speed oscillation caused by switching function, the amplitude of states estimation fluctuated randomly. Furthermore, the estimating process of x and y rate states was influenced by pitch and roll rate, because those states are coupled. QuasiSMO, IT2FSMO, HOSMO, and EKF methods performed in a more robust manner than STASMO in dealing with uncertainty.

The steady-state error represented by MSE of estimated states, is shown in Table 8.4.

As noted QuasiSMO has generated the smallest value of MSE among other observers in almost all estimated states except z-rate.

Table 8.4. Mean squared errors of estimated states

Observer methods	MSE estimated state of					
	x rate	pitch rate	y rate	roll rate	yaw rate	z rate
QuasiSMO	1.57782E-09	1.00376E-06	3.87437E-09	5.61595E-07	2.64322E-07	1.29236E-07
IT2FSMO	7.78173E-09	1.02141E-06	1.42264E-08	6.06107E-07	8.7829E-07	1.2926E-07
STASMO	0.000863278	0.010252766	0.000804644	0.010495916	0.000177176	0.000169417
HOSMO	6.42665E-06	0.000602754	6.48343E-06	0.000587765	2.53336E-05	2.87161E-05
EKF	3.15193E-06	3.20693E-06	8.73009E-07	2.37287E-06	0.00017463	1.23502E-07

Figures 8.36 - 8.41 shows the different responses of observers methods in tracking measured states. Generally, all approaches performed well in tracking the true states by showing small estimation errors for all measured states. However, some observers including QuasiSMO, IT2FSMO, and EKF performed better than STASMO and HOSMO by generating smoother estimation errors than the rest of observer approaches. Fluctuated errors were noticeable clearly following the use of STASMO and HOSMO methods. QuasiSMO and IT2FSMO performed better than EKF in steady-state condition by presenting smaller estimation errors of true states, whereas EKF was better than those two observers in states change condition as seen in Figures 8.36 and 8.37. The overall performance of tracking true states computed by MSE as presented in Table 8.5 shows that QuasiSMO outperformed others by showing smaller MSE in x, y and z states for which EKF performed better in pitch, roll, and yaw than others.

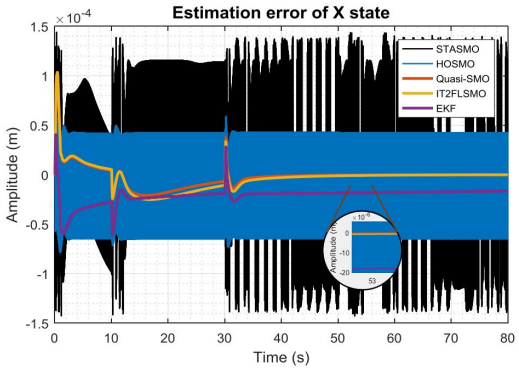

Figure 8.36. Estimation error of x

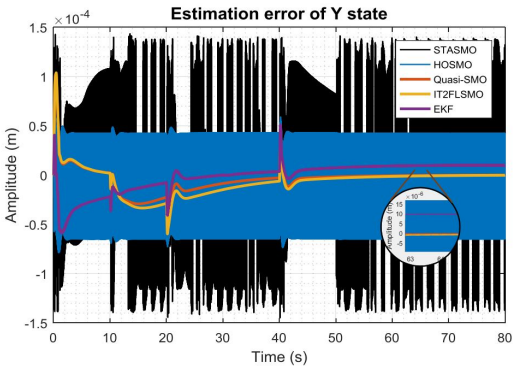

Figure 8.37. Estimation error of y

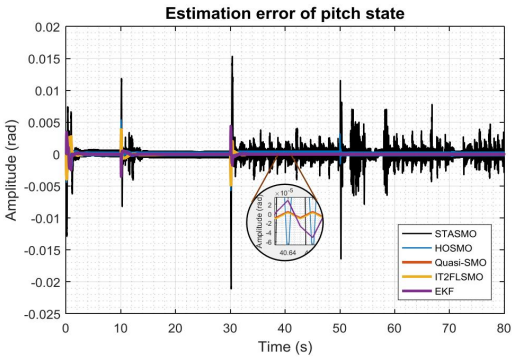

Figure 8.38. Estimation error of pitch

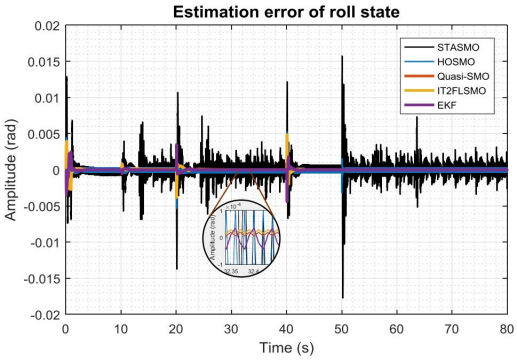

Figure 8.39. Estimation error of roll

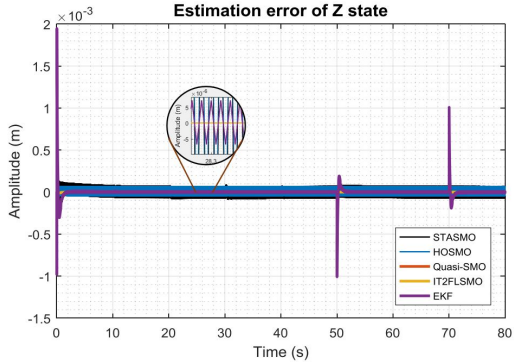

Figure 8.40. Estimation error of z

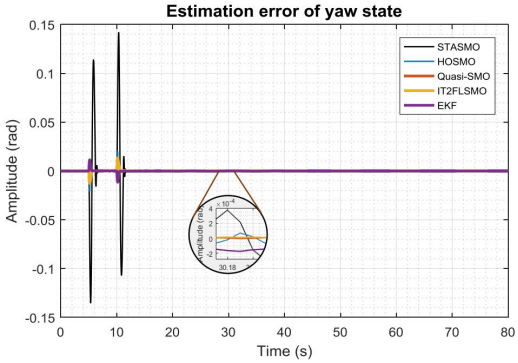

Figure 8.41. Estimation error of yaw

Table 8.5. Mean squared errors of true states tracking

Observer method	MSE true state tracking of					
	x	pitch	y	roll	yaw	z
QuasiSMO	1.47706E-10	1.14292E-07	2.0352E-10	1.13978E-07	9.98569E-07	1.01679E-12
IT2FSMO	1.87792E-10	1.14892E-07	2.76137E-10	1.14526E-07	1.00592E-06	1.04679E-12
STASMO	6.78558E-09	3.27439E-06	6.94394E-09	3.0371E-06	0.00021995	4.87478E-09
HOSMO	2.99117E-09	1.7202E-07	2.99217E-09	1.65871E-07	1.18299E-06	2.97571E-09
EKF	5.14285E-10	8.83578E-08	2.24911E-10	8.90096E-08	7.95164E-07	4.86299E-09

On the grounds of control, the presence of parameter uncertainty influenced significantly the control performance in managing x, pitch, y and roll as seen in Figures 8.24, 8.26, 8.28, and 8.30 using STASMO. In steady-state condition of roll and pitch motions, states fluctuated around $\pm 2.20°$ and $\pm 1.70°$ respectively. Since pitch and x, and roll and y states are coupled, the tracking references of x and y was affected as can be seen in Figures 8.24 and 8.28. Small fluctuation on the states around $\pm 2.25cm$ and $\pm 2.91cm$ appeared respectively in steady-states condition.

In addition, different behaviour were seen in yaw action response as presented in Figure 8.34. The controller responded with slower rise time when QuasiSMO, IT2FSMO and

EKF were utilized, and faster when STASMO and HOSMO were employed. It means that STASMO and HOSMO performed better than QuasiSMO, IT2FSMO and EKF for yaw movement. However, the selected controller responded to step inputs for all states with no overshoot and produced smooth reference tracking in z-axis movement.

Furthermore, high frequency chattering was still clearly visible in the control inputs with using STASMO and HOSMO as depicted in Figures 8.42 - 8.45, while QuasiSMO, IT2FSMO, and EKF generated smooth control signals for the quadcopter UAV.

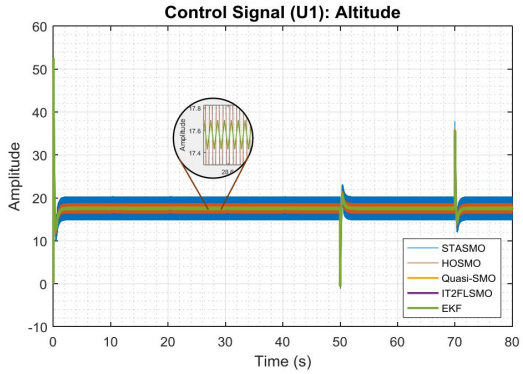

Figure 8.42. Control signal u_1

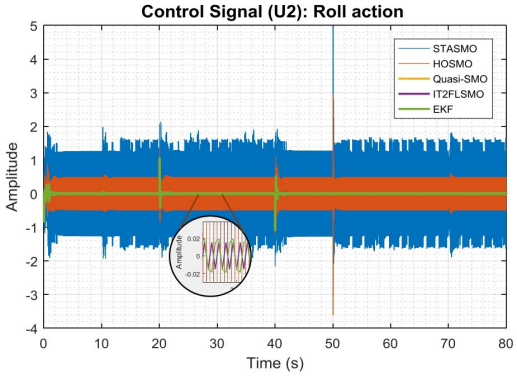

Figure 8.43. Control signal u_2

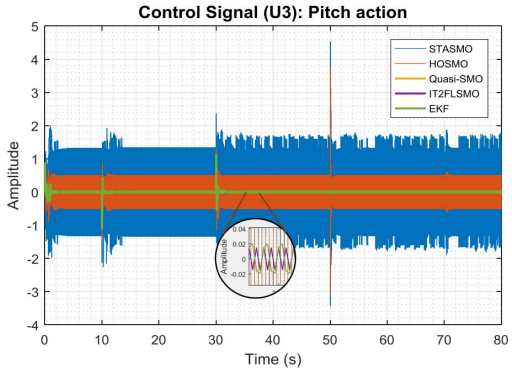

Figure 8.44. Control signal u_3

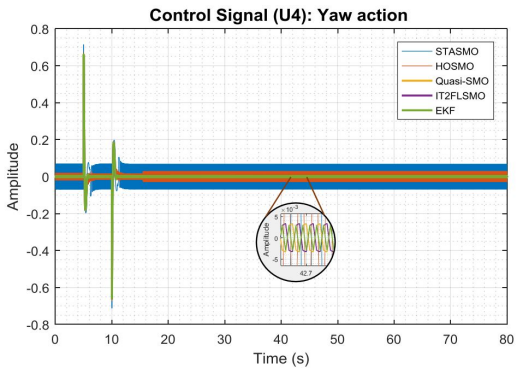

Figure 8.45. Control signal u_4

It can be seen as in Table 8.6 that QuasiSMO improved the controller performance by demonstrating smaller steady-state errors of the output states than other observers.

Table 8.6. Mean squared errors of steady-states errors

Observer methods	**MSE steady-state error of**					
	x	**pitch**	**y**	**roll**	**yaw**	**z**
QuasiSMO	5.55239E-11	7.67234E-11	4.45361E-11	6.982E-12	2.74156E-08	2.07698E-11
IT2FSMO	1.1796E-10	7.62668E-11	1.12335E-10	7.41354E-12	4.30448E-08	2.2918E-11
STASMO	0.000126945	0.000120853	0.000120408	9.93095E-05	7.80992E-08	1.59643E-08
HOSMO	2.18724E-08	4.18404E-08	2.81358E-08	4.85754E-08	8.6927E-09	4.70374E-09
EKF	6.74428E-11	1.09636E-09	1.48999E-10	9.56543E-10	2.96763E-08	6.3628E-11

In summary, despite of in the presence of parameters mismatch, STASMC still showed

good performance to stabilize the dynamics of the quadcopter UAV with no response overshoot, small rise time with using STASMO and HOSMO in yaw movement, free from chattering when QuasiSMO, IT2FSMO and EKF observer method were employed, and small steady-states errors. In terms of observer, QuasiSMO method still acted with high performance by showing smooth estimated states, smaller states estimation errors, and faster in tracking true states. Therefore, the observer has influenced and improved the overall performance of selected controller.

8.2.3 With noise and no parameter mismatch

In this case, high power white noise disturbance was incorporated into the output states of the UAV system. This condition was invoked to highlight the performance of selected control system and observer techniques in dealing with the disturbance. The responses of the methods are presented in Figures 8.46 – 8.67.

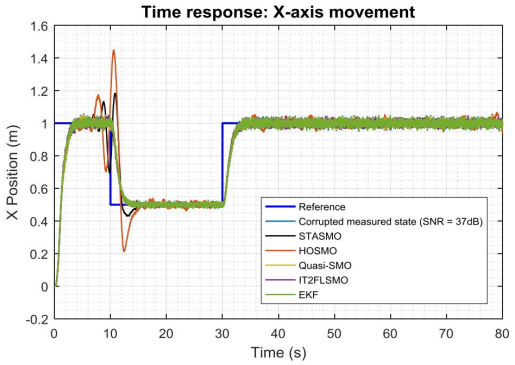

Figure 8.46. x-axis movement

Figure 8.47. Rate of x-axis movement

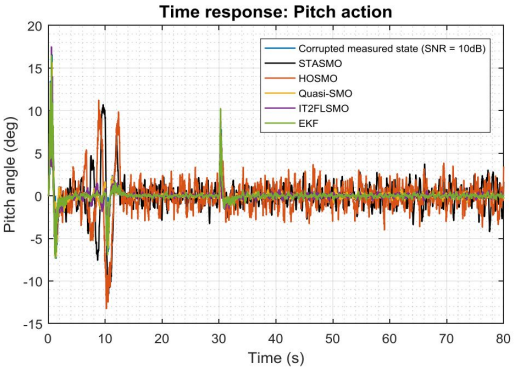

Figure 8.48. Pitch action

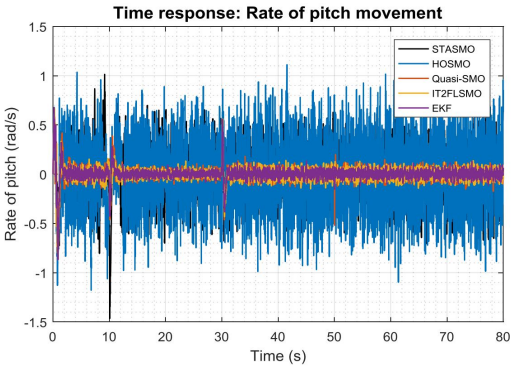

Figure 8.49. Rate of pitch movement

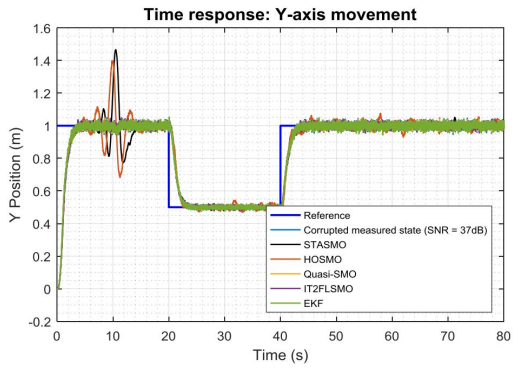

Figure 8.50. y-axis movement

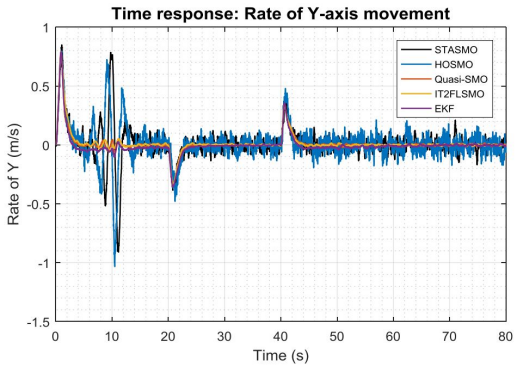

Figure 8.51. Rate of y-axis movement

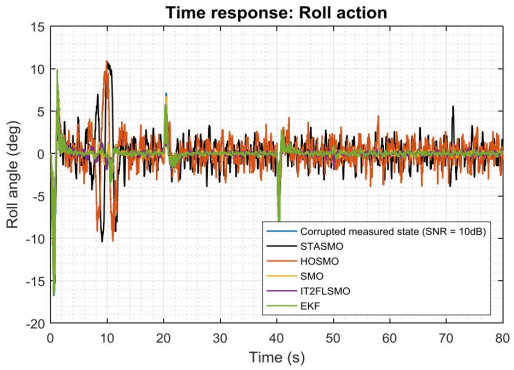

Figure 8.52. Roll action

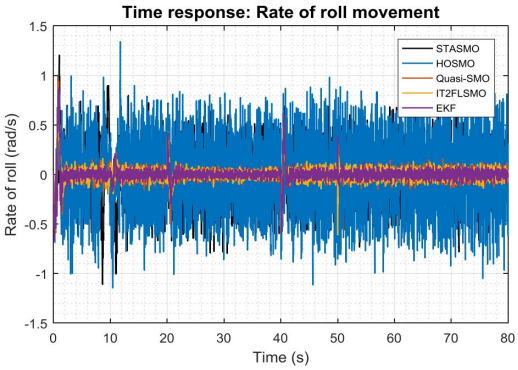

Figure 8.53. Rate of roll movement

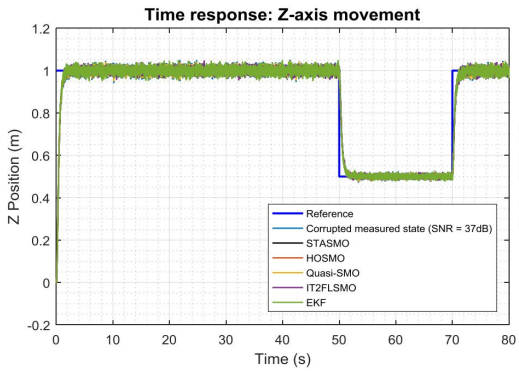

Figure 8.54. z-axis movement

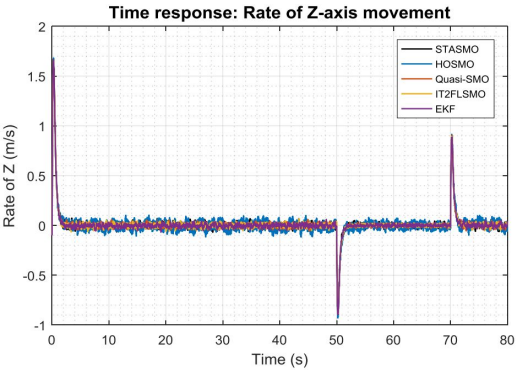

Figure 8.55. Rate of z-axis movement

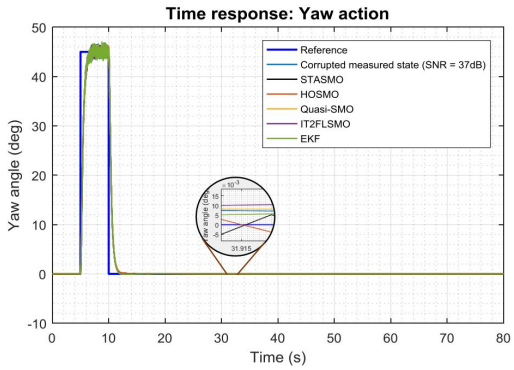

Figure 8.56. Yaw action

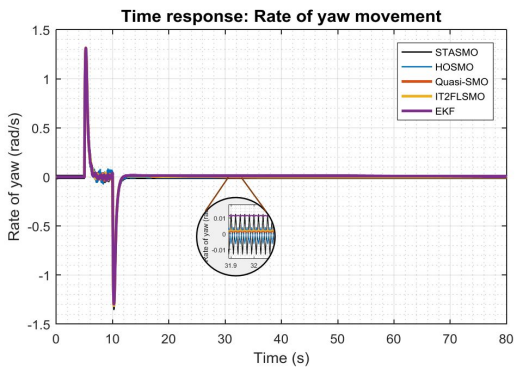

Figure 8.57. Rate of yaw movement

It is noted in the simulation results presented in Figures 8.47, 8.49, 8.51, 8.53, 8.55,

and 8.57, that the effect of noise was reduced sufficiently by QuasiSMO, IT2FSMO, and EKF so that second order states were generated quite smoothly. Noise disturbance on the output states did not affect significantly the performance of the observers. In contrast, corrupted output states influenced the performances of STASMO and HOSMO significantly, especially in estimating second order state of pitch and roll as shown in Figures 8.49 and 8.53. The noise amplitude on the output states were amplified high enough due to the effect of switching terms on the methods.

Furthermore, although the SNR on the output states x and y were the same as yaw and z states, that is 37dB where it was smaller than the noise on the pitch and roll states, that is 10dB, the effect of noise received by x and y was greater than yaw and z, as can be seen in Figures 8.47, 8.51, 8.55, and 8.57. It is because x and pitch, y and roll are dependent states which will affect each other.

The presence of noise in the output states might affect the performance of estimator in generating second order states of systems. Beside smoothness, steady-states error of the estimated states is one of the important factors to measure the quality of observer. Steady-states error of estimated states was calculated by MSE method as presented in Table 8.7. From the table, it can be seen that QuasiSMO, IT2FSMO and EKF performed as expected by presenting smaller MSE values than others. In other words, those observers were able to reduce the effect of noise well.

Table 8.7. Mean squared states of estimated states

Observer methods	MSE estimated state of					
	x rate	pitch rate	y rate	roll rate	yaw rate	z rate
QuasiSMO	3.95071E-05	0.00197069	3.33963E-05	0.002258451	1.88945E-06	0.00028979
IT2FSMO	3.32656E-05	0.001905274	2.85976E-05	0.002155247	3.44584E-06	0.000273046
STASMO	0.001895702	0.060469865	0.002339077	0.066979186	0.000168501	0.000511565
HOSMO	0.003835517	0.130963062	0.003988358	0.133632394	2.97262E-05	0.001573807
EKF	4.95782E-05	0.001197792	0.000136579	0.001397282	9.2184E-05	0.000281564

In the presence of noise, true states tracking errors appeared similar for all observer methods as presented in Figures 8.58 - 8.63. However, observing computed MSE carefully as presented in Table 8.8, each observer technique showed different performance in tracking true states. Although STASMO and HOSMO could not estimate second order states as well as QuasiSMO, IT2FSMO and EKF; they performed better in tracking true

states. It is noticeable clearly in the Table 8.8 that STASMO and HOSMO had smaller MSE than other approaches in tracking the measured states of x, y, yaw and z, while QuasiSMO and IT2FSMO showed better performance in tracking pitch and roll states respectively. However, STASMO and HOSMO did not work as well as other approaches in estimating second order states so that the overall control performance would be affected.

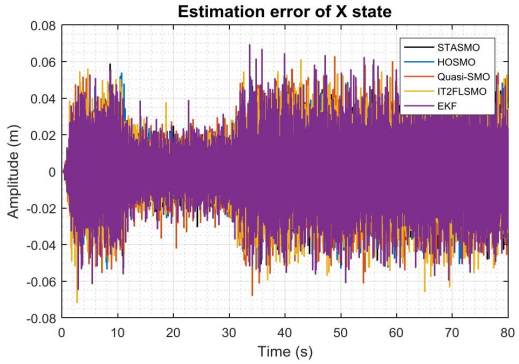

Figure 8.58. Estimation error of x

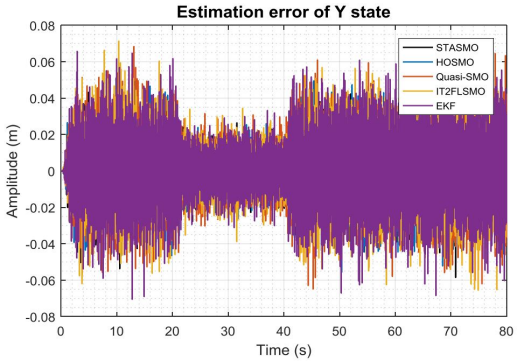

Figure 8.59. Estimation error of y

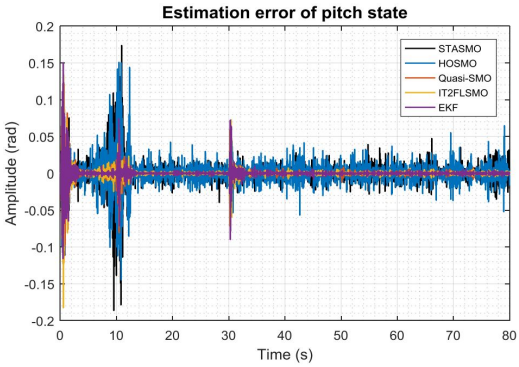

Figure 8.60. Estimation error of pitch

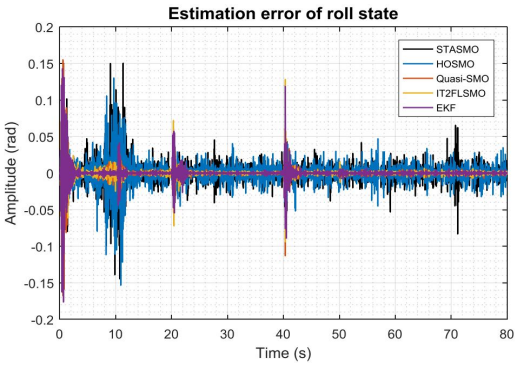

Figure 8.61. Estimation error of roll

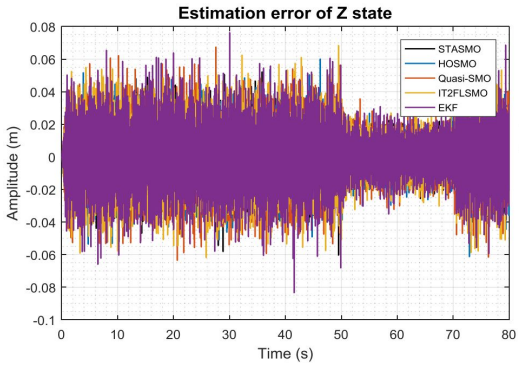

Figure 8.62. Estimation error of z

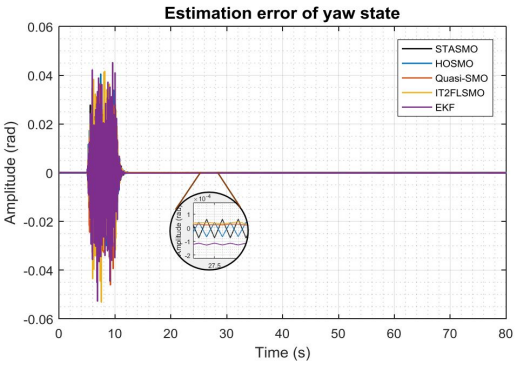

Figure 8.63. Estimation error of yaw

Table 8.8. Mean squared errors of true states tracking

Observer method	MSE true state tracking of					
	x	pitch	y	roll	yaw	z
QuasiSMO	0.000268934	4.79577E-05	0.000276107	6.011E-05	1.10571E-05	0.000280511
IT2FSMO	0.000278753	5.18486E-05	0.000271722	4.24766E-05	1.28945E-05	0.000283318
STASMO	0.000174721	0.000177572	0.000174562	0.000166051	8.30785E-06	0.000183073
HOSMO	0.000192143	0.000178546	0.00018732	0.000160282	8.49957E-06	0.00019333
EKF	0.000302629	5.27997E-05	0.000311298	7.42692E-05	1.52056E-05	0.000305571

The presence of noise will certainly affect the performance of controller. However, the effect of noise on systems depends on how to manage and select control and observer systems. It is noted in Figures 8.46, 8.48, 8.50, 8.52, 8.54, and 8.56, that using STASMO and HOSMO as states estimator had much effect on the control performance in controlling pitch, roll, x, and y. There were significant overshoot and undershoot in x-axis movement when references changed around maximum 18.20% and 44.86% for STASMO and HOSMO respectively, meanwhile in y-axis moment the overshoots were 46.67% and 30.53% for STASMO and HOSMO respectively. Higher overshoot was noticeable clearly in controlling pitch motion by around 10.69° and 11.22° when using STASMO and HOSMO respectively, while in roll state the overshoot was about 10.82° and 10.95° with STASMO and HOSMO respectively.

In contrast, the selected controller performed in a more robust manner in controlling all output states when QuasiSMO, IT2FSMO, and EKF were employed. However, the effect of noise was still noticeable clearly around the output states of the UAV system and control inputs as seen in Figures 8.64 - 8.67. This is the trade-off of choosing parameters for the observers. On the one hand, increasing or decreasing observer gains will speed up tracking true value, yet may potentially increase sensitivity to measurement noise and vice versa. In this research, the speed of tracking measured states is more considered (without neglecting noise effect) to make the controller more responsive.

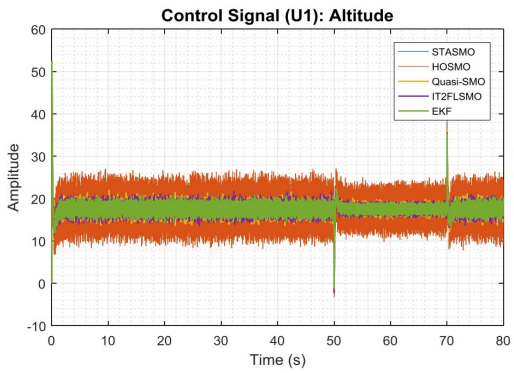

Figure 8.64. Control signal u_1

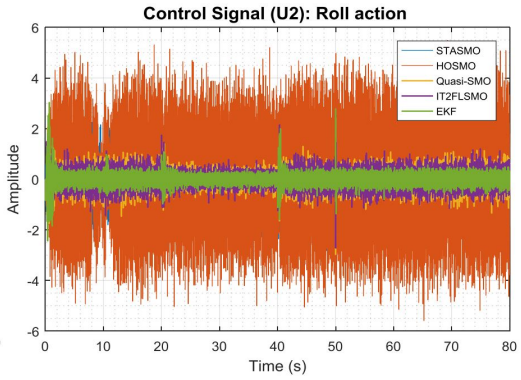

Figure 8.65. Control signal u_2

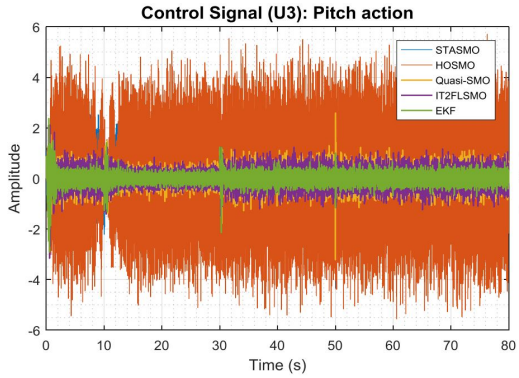

Figure 8.66. Control signal u_3

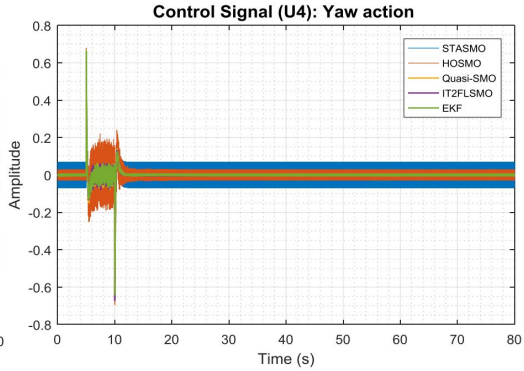

Figure 8.67. Control signal u_4

Steady-states error of the output states was also considered to further highlight the robustness of selected controller and observers in dealing with noise disturbance. As noted in Table 8.9, although STASMO and HOSMO had problem in handling the effect of noise in estimating unmeasured states, the observers influenced control system to reduce steady-state error well. STASMO and HOSMO mostly had smaller steady-state errors (except roll and pitch) computed by MSE method than other methods. Switching term in the algorithms has had the main role of keeping observed states tracking the measured states all the time. However, due to noise disturbance, the control signals produced were not suitable for real-time embedded systems as seen in Figures 8.64 - 8.67.

Table 8.9. Mean squared errors of steady-states errors

Observer methods	**MSE steady-state error of**					
	x	**pitch**	**y**	**roll**	**yaw**	**z**
QuasiSMO	0.000149088	0.005048901	0.000154376	0.005171346	2.2355E-08	0.000157083
IT2FSMO	0.000155418	0.00504895	0.000149175	0.005154261	2.48281E-08	0.000158504
STASMO	0.000104144	0.005237038	0.000132782	0.005341325	1.16982E-08	1.83708E-05
HOSMO	0.0004098	0.005617762	0.000510022	0.005538562	5.55784E-09	7.34468E-05
EKF	0.000194229	0.005049715	0.000210614	0.005159779	6.18652E-09	0.000198527

In summary, in the presence of noise, STASMC maintains good performance to stabi-

lize the dynamics of the quadcopter UAV with no response overshoot except for x and y movements, pitch and roll actions when applying STASMO and HOSMO, small rise time, free from chattering when employing QuasiSMO, IT2FSMO and EKF observer methods, and small steady-states errors. Furthermore, fluctuated signals randomly on the references still appear due to noise disturbance. The selection of observer gain is a trade-off. Reducing or increasing the gain may make observers become insensitive to noise, however, on the other hand it may delay tracking true states. In the observer side, QuasiSMO method still outperforms other methods by showing smooth estimated states, smaller states estimation errors, and faster in tracking true states. Therefore, the observer has influenced and improved the overall performance of selected controller.

8.2.4 With noise and with parameter mismatch

At this stage of testing the performance of STASMC and observers, noise and uncertainty were introduced into the quadcopter UAV system at the same time. The simulation results acquired are presented in Figures 8.68 – 8.89.

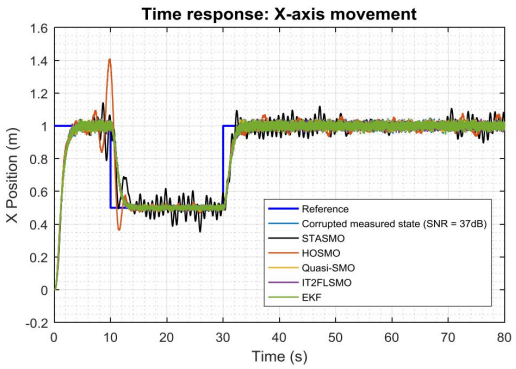

Figure 8.68. x-axis movement

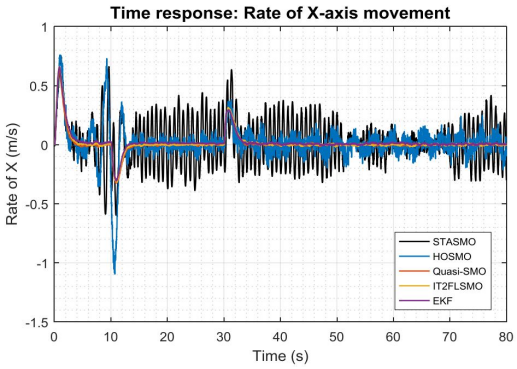

Figure 8.69. Rate of x-axis movement

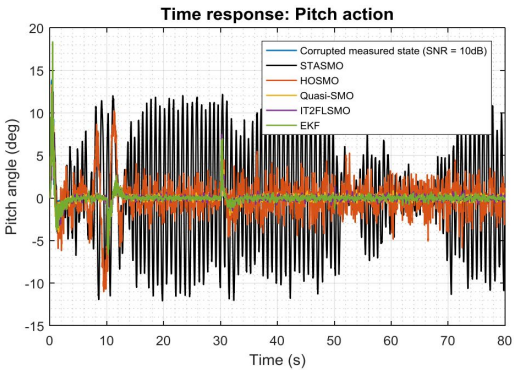

Figure 8.70. Pitch action

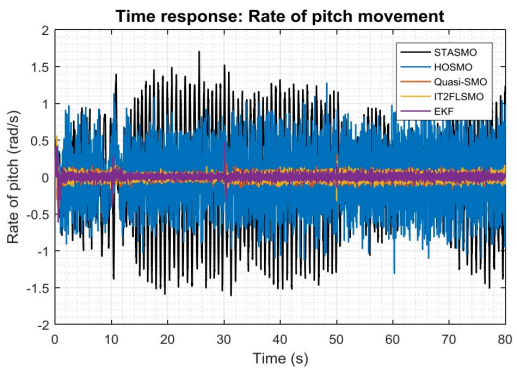

Figure 8.71. Rate of pitch movement

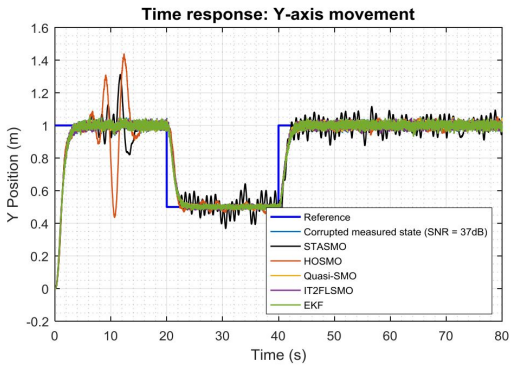

Figure 8.72. y-axis movement

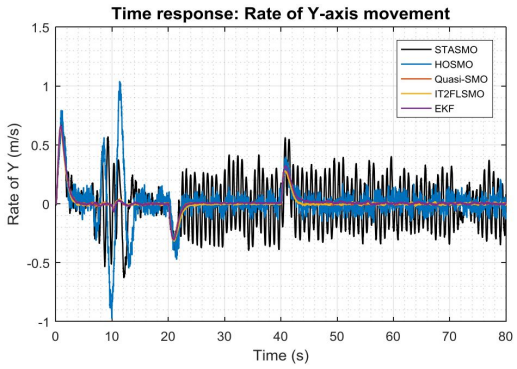

Figure 8.73. Rate of y-axis movement

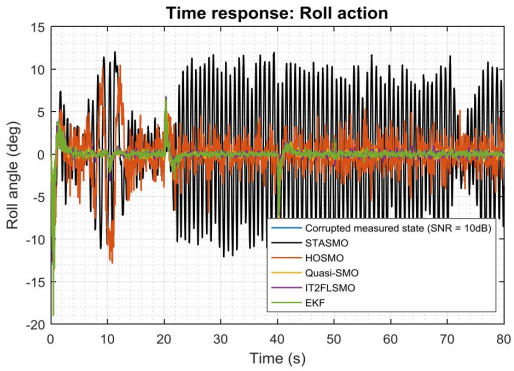

Figure 8.74. Roll action

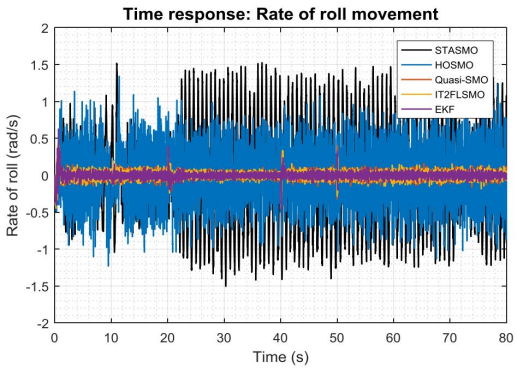

Figure 8.75. Rate of roll movement

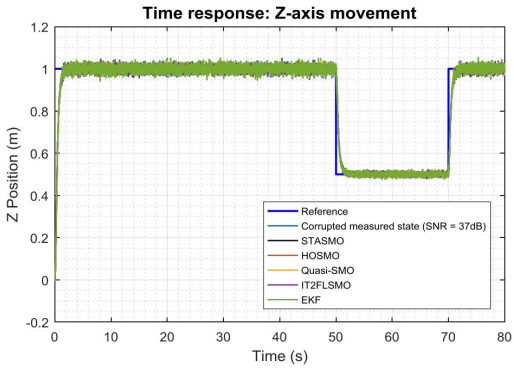

Figure 8.76. z-axis movement

Figure 8.77. Rate of z-axis movement

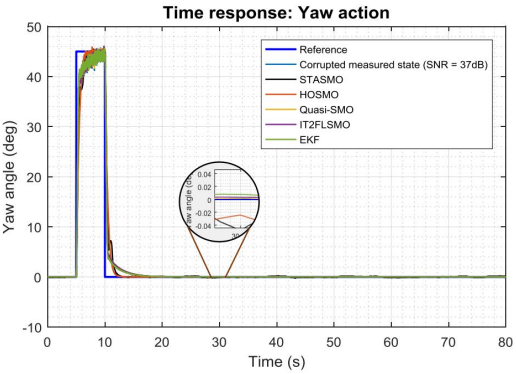

Figure 8.78. Yaw action

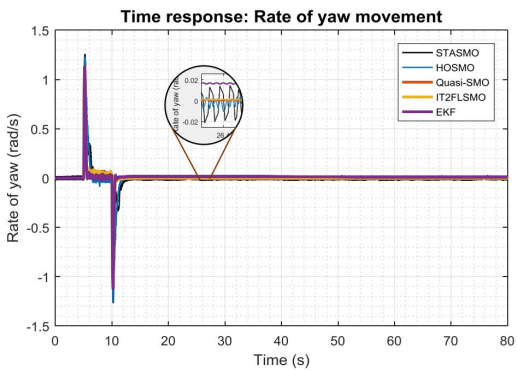

Figure 8.79. Rate of yaw movement

The same noise power as in the previous case with 50% parameters mismatch was in-

troduced into the UAV system in this numerical validation. It is noted in the simulation results presented in Figures 8.69, 8.71, 8.73, 8.75, 8.77, and 8.79, that the effect of the introduced high power noise and high uncertainty was reduced sufficiently by QuasiSMO, IT2FSMO, and EKF so that second order states were generated quite smoothly. In other words, noise disturbance on the output states as well as parameters mismatch did not affect significantly the performance of the observers. In contrast, corrupted output states influenced the performance of STASMO and HOSMO quite significantly, especially in estimating second order states of pitch and roll as shown in Figures 8.71 and 8.75. The noise amplitude on the output states were amplified high enough due to the effect of switching term on the methods. In addition, HOSMO could not deal with parameters mismatch well. This is shown by the high amplitude oscillation of estimated states of x, and y rate when the x state changes.

Although the SNR on the output states of x and y were the same as yaw and z states, that is 37dB where it was smaller than the noise on the pitch and roll states, that is 10dB, the effect of noise received by x and y was greater than yaw and z, as can be seen in Figures 8.69, 8.73, 8.77, and 8.79. This is because x and pitch, y and roll are dependent states which will affect each other.

MSE of estimated states in steady-state condition as presented in Table 8.10 describes another view of observer performance. The smaller the MSE value, it may become one proof that the method has performed well. Similar to previous cases, QuasiSMO, IT2FSMO, and EKF still showed the superiority in estimating second order states. It can be seen in the table that those observers had smaller MSE in every estimated state than those of STASMO and HOSMO. However, generally only QuasiSMO and IT2FSMO produced estimated states with smaller MSE consistently for all second order states.

Table 8.10. Mean squared errors of estimated states

Observer methods	**MSE estimated state of**					
	x rate	**pitch rate**	**y rate**	**roll rate**	**yaw rate**	**z rate**
QuasiSMO	2.5525E-05	0.002094731	2.5978E-05	0.002130726	2.68267E-07	0.000283188
IT2FSMO	3.12687E-05	0.002033382	3.71187E-05	0.002233188	5.25198E-07	0.000316382
STASMO	0.023525204	0.399282357	0.034575754	0.564109729	0.000183453	0.002076536
HOSMO	0.004259111	0.157738573	0.003225206	0.147723186	1.71098E-05	0.001773266
EKF	6.39898E-05	0.001311612	4.73083E-05	0.001348531	0.000176153	0.000259961

In the presence of noise and parameters mismatch, true states tracking errors appeared similar for all observer methods as presented in Figures 8.80 - 8.85. However, noting the computed MSE as presented in Table 8.11, each observer technique showed different performance in tracking true states. Although HOSMO could not estimate second order states as well as QuasiSMO, IT2FSMO and EKF; it showed better performance in tracking true states. It is noticeable in Table 8.11 that HOSMO had smaller MSE than other approaches in tracking the measured states of x, y, yaw and z, while QuasiSMO and IT2FSMO showed better performance in tracking pitch and roll states. However, HOSMO did not perform as well as other approaches in estimating second order states so that the overall control performance would be affected.

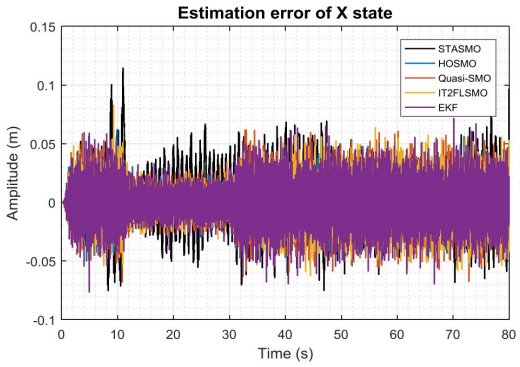

Figure 8.80. Estimation error of x

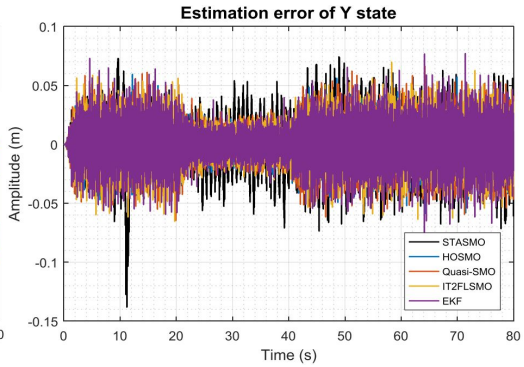

Figure 8.81. Estimation error of y

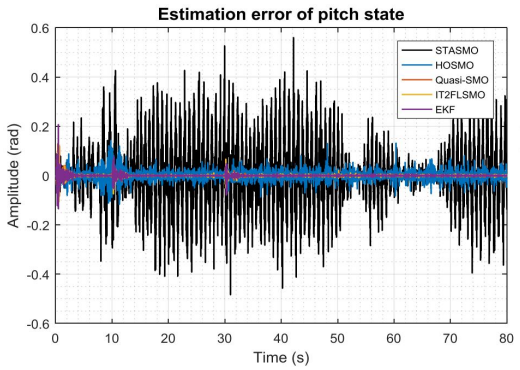

Figure 8.82. Estimation error of pitch

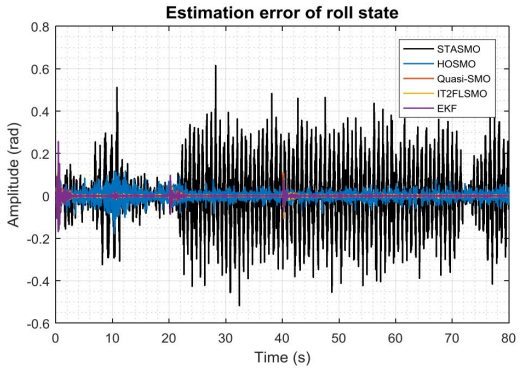

Figure 8.83. Estimation error of roll

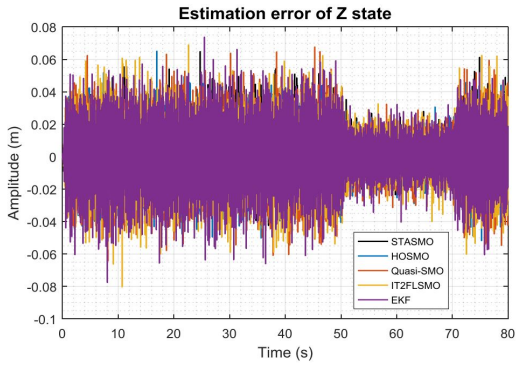

Figure 8.84. Estimation error of z

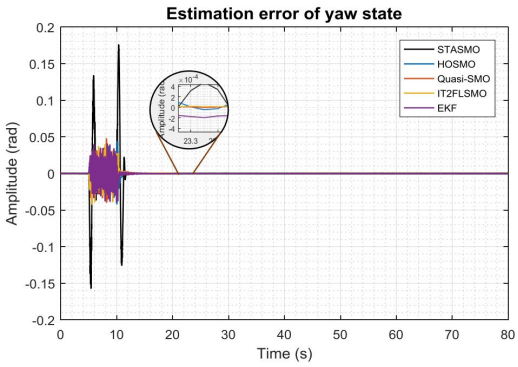

Figure 8.85. Estimation error of yaw

Table 8.11. Mean squared errors of true states tracking

Observer method	MSE true state tracking of					
	x	pitch	y	roll	yaw	z
QuasiSMO	0.000268757	4.18026E-05	0.000275136	3.77515E-05	1.21974E-05	0.000268942
IT2FSMO	0.000282343	2.88577E-05	0.000282501	3.77656E-05	1.13356E-05	0.000280316
STASMO	0.00056134	0.013350086	0.000572546	0.014828432	0.000290956	0.000201345
HOSMO	0.000189241	0.000354916	0.000184425	0.00039083	1.09729E-05	0.000188622
EKF	0.000310058	3.86859E-05	0.000304581	6.30563E-05	1.37512E-05	0.000311477

The presence of noise and parameters mismatch might certainly affect the performance of controller. However, the effect of noise on systems depends on how to organize and select control and observer systems. Note in Figures 8.68, 8.70, 8.72, 8.74, 8.76, and 8.78, that the use of STASMO and HOSMO as states estimator had much effect on the control performance in controlling x, y, pitch, and roll. There were significant overshoot and undershoot in x-axis movement especially when references changed around maximum 40.86% and 21.17% when using HOSMO and STASMO respectively, meanwhile in y-axis moment the overshoots were around 56.38% and 31.24% for HOSMO and STASMO respectively. Furthermore, high amplitude of random fluctuated states because of noise disturbance were noticeable clearly in controlling pitch around $\pm 8.65^{\circ}$ and $\pm 12.04^{\circ}$ using HOSMO and STASMO respectively, while in roll state the fluctuations were about $\pm 10.45^{\circ}$ and $\pm 12.10^{\circ}$ for HOSMO and STASMO respectively.

In contrast, the UAV system behaved in a more stable manner in all movements when QuasiSMO, IT2FSMO, and EKF were employed. In other words, the control system performs more robustly if those observer approaches are applied. However, the effect of noise was still noticeable around the output states and control inputs as seen in Figures 8.86 - 8.89. This is due to the trade-off of choosing parameters for the observers. On the one

hand, increasing or decreasing observer gains will speed up tracking true value, yet may potentially increase sensitivity to measurement noise and vice versa. In this research, the speed of tracking measured states is more considered (without neglecting noise effect) to make the controller more responsive.

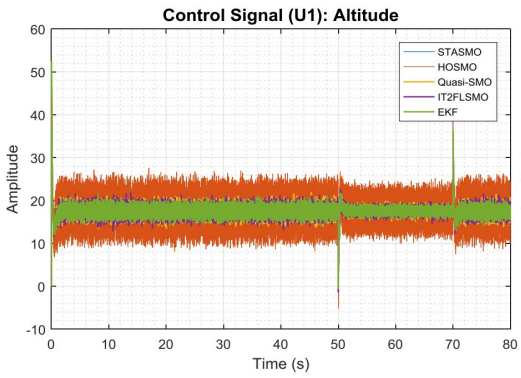

Figure 8.86. Control signal u_1

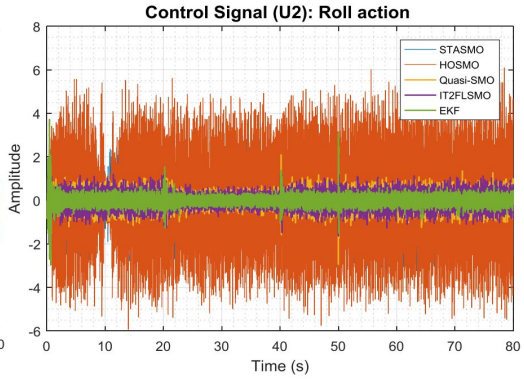

Figure 8.87. Control signal u_2

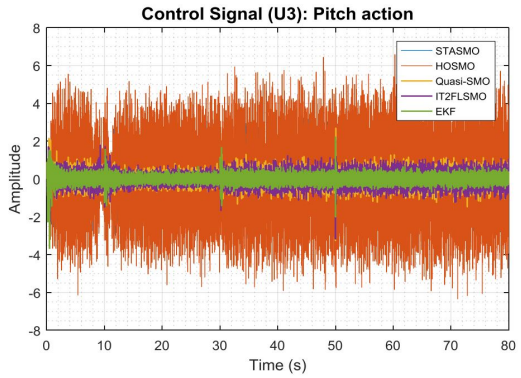

Figure 8.88. Control signal u_3

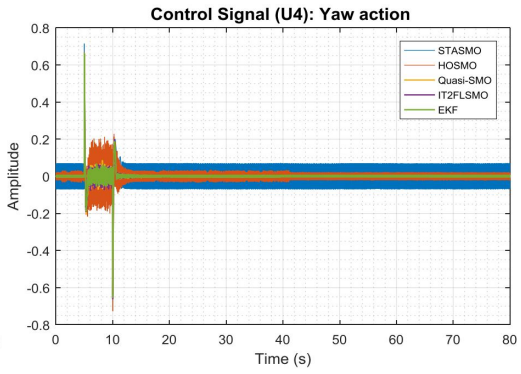

Figure 8.89. Control signal u_4

Steady-states error of output states was taken into account to further evaluate the robustness of selected controller and observers in dealing with noise disturbance and parameters mismatch. It is noted in Table 8.12, that in the presence of such issues, generally steady-states errors were small and the observers influenced the control system to stabilize the rotor-craft and reduce steady-state error quite well. In this case, QuasiSMO observer method showed better performance than the other estimators in improving controller performance by exhibiting smaller steady-state error in x, y, and yaw states computed by MSE method, whereas STASMO improved the controller performance better than others in roll and z states . However, due to noise disturbance and parameters mismatch, using STASMO and HOSMO, the control signals produced were not suitable for real-time embedded systems. In addition, the step responses of the controller was not smooth enough

when STASMO and HOSMO were applied by showing high overshoot and undershoot responses when references changed.

Table 8.12. Mean squared errors of steady-states errors

Observer methods	MSE steady-state error of					
	x	pitch	y	roll	yaw	z
QuasiSMO	0.000151116	0.010144717	0.00015613	0.012517839	2.71788E-08	0.000149155
IT2FSMO	0.000163444	0.010076722	0.00016175	0.012577633	3.45172E-08	0.000156412
STASMO	0.001337583	0.009700899	0.001954941	0.012148891	1.79973E-06	2.98304E-05
HOSMO	0.000654156	0.010983188	0.000375645	0.014017541	7.15961E-08	7.76457E-05
EKF	0.000195049	0.010079484	0.000200092	0.012469069	3.20473E-08	0.000202875

In summary, in the presence of noise and parameter mismatch, STASMC maintained good performance in stabilizing the dynamics of the quadcopter UAV with no response overshoot except for x and y movements, pitch and roll actions when STASMO and HOSMO were used, considered small rise time, free from chattering when QuasiSMO, IT2FSMO and EKF observer methods were employed, and small steady-states errors. In addition, random fluctuated signals on the references still appeared due to noise disturbance. This is due to the selection of observer gain being a trade-off. Reducing or increasing the gain may make observers become insensitive to noise, however, on the other hand it may delay tracking true states. In terms of observer, generally QuasiSMO method still outperformed other methods by showing smooth estimated states, smaller states estimation errors, and faster in tracking true states. Therefore, the observer has influenced and improved the overall performance of selected controller.

8.3 Performance summary

After conducting performance evaluation of the control system and the observer methods with 4 predefined scenarios, it can be summarized that set-point integral super-twisting algorithm of sliding mode control (STASMC) can still maintain the stability of its performance to control the dynamics of quadcopter UAV well in all conditions. Integral term plays an important role in reducing steady-state error of the output states while set-point weighting function plays a role in reducing overshoot due to the integral term. Oscillations on x, y, pitch and roll motions only occur in all scenarios (1-4) when STASMO and

HOSMO observer methods were employed. In terms of observer, actually QuasiSMO and IT2FSMO have similar performance in estimating unmeasured states. However, IT2FSMO has slower computational time than QuasiSMO due to type reduction process in the type-2 fuzzy logic system. Hence, generally QuasiSMO outperformed than others and has contributed to improved control system performance in dealing with all predefined conditions

Chapter 9

Numerical simulations: set-point integral dynamic sliding mode control

9.1 Introduction

This chapter presents numerical simulation results to highlight the performance of set-point integral dynamic sliding mode control (DSMC) with the developed nonlinear full-order state observers, namely:

- quasi - sliding mode observer
- sliding mode-based interval type-2 fuzzy observer
- super-twisting algorithm of sliding mode observer
- higher order sliding mode observer
- extended Kalman filter

Performance comparison between sliding mode-based observers and extended Kalman filter (EKF) is conducted to highlight the robustness of each method. Through numerical validation, the performance of the control system and the performance comparisons of the observer methods are carried out and presented in this chapter.

The overall control and observer system diagram, simulation setup and performance criteria are the same as in Chapter 6. In addition, the same scenerios as in Chapter 6 are introduced to test the performance of the control and observer methods, including:

- without noise and no parameter mismatch

- without noise and with parameter mismatch
- with noise and no parameter mismatch
- with noise and with parameter mismatch

The purpose of this approach is to determine the ability of the controller and observer to deal with several conditions.

The reference frame of the quadcopter movements in this numerical simulation can be seen in Figure 9.1

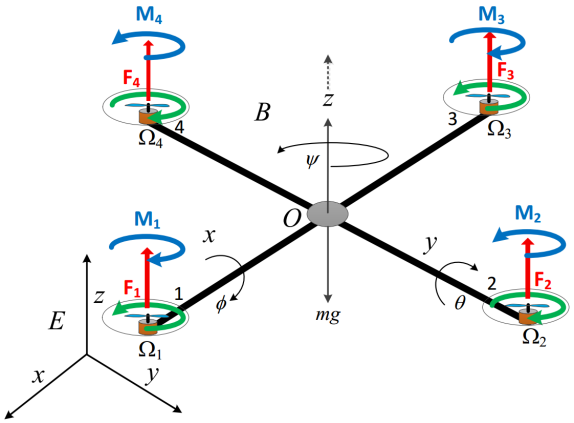

Figure 9.1. Reference frame of quadcopter movements

where x, y, and z represent the movements of the quadcopter in those axes, while ϕ, θ, and ψ represent roll, pitch, and yaw motions of the quadcopter respectively.

9.2 Simulation results

After a set of investigations, suitable values of sliding mode control to stabilized quadcopter UAV were determined as

$$\begin{array}{cccccc}
\lambda_\phi = 10.0 & \lambda_\theta = 10.0 & \lambda_\psi = 15.0 & \lambda_x = 4.0 & \lambda_y = 4.0 & \lambda_z = 30.0 \\
c_\phi = 20.0 & c_\theta = 20.0 & c_\psi = 5.0 & c_x = 0.9 & c_y = 0.9 & c_z = 10.0 \\
\lambda_{i\phi} = 0.01 & \lambda_{i\theta} = 0.01 & \lambda_{i\psi} = 2.86 & \lambda_{ix} = 0.053 & \lambda_{iy} = 0.053 & \lambda_{iz} = 6.22 \\
k_\phi = 1.5 & k_\theta = 1.5 & k_\psi = 1.0 & k_x = 0.1 & k_y = 0.1 & k_z = 2.0 \\
\mu_\phi = 100.0 & \mu_\theta = 100.0 & \mu_\psi = 4.5 & \mu_x = 10,0 & \mu_y = 10.0 & \mu_z = 3.0
\end{array}$$

9.2.1 Without noise and no parameter mismatch

The simulation results obtained in this case are presented in Figures 9.2 – 9.23.

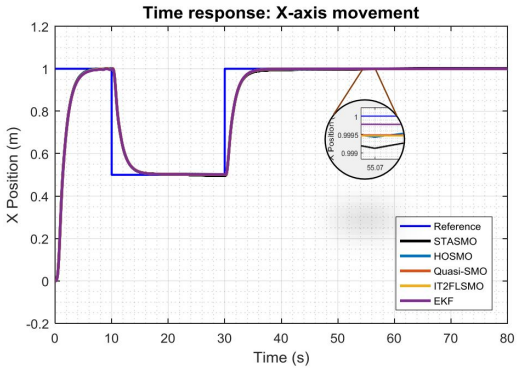

Figure 9.2. x-axis movement

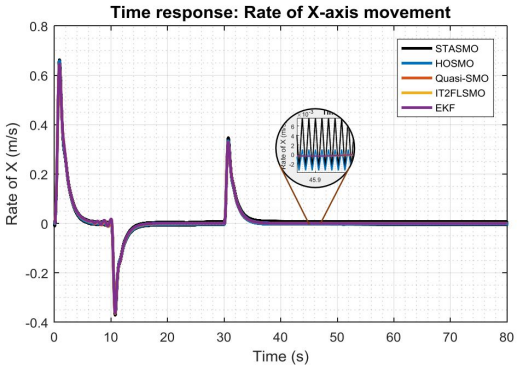

Figure 9.3. Rate of x-axis movement

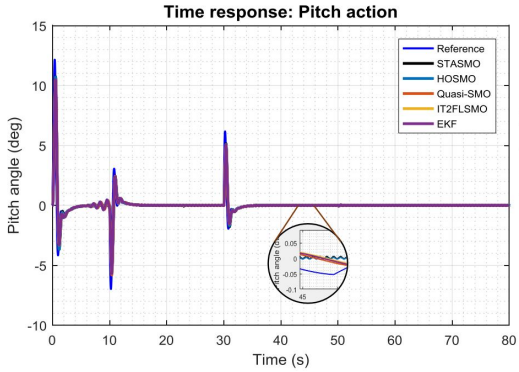

Figure 9.4. Pitch action

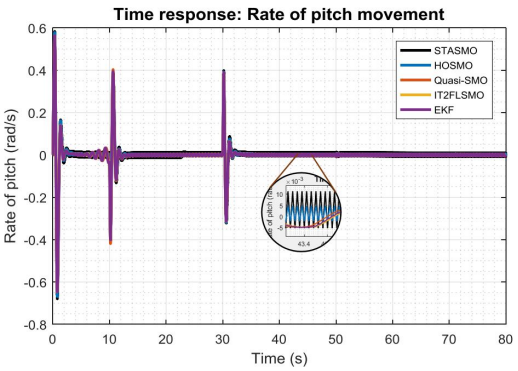

Figure 9.5. Rate of pitch movement

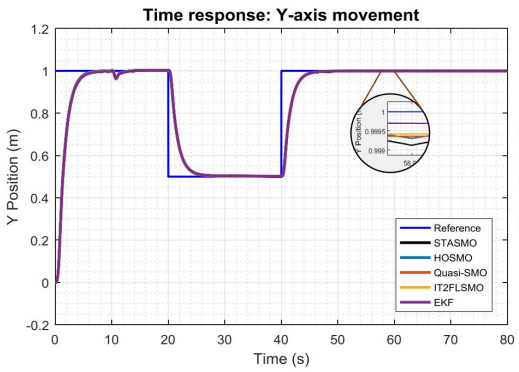

Figure 9.6. y-axis movement

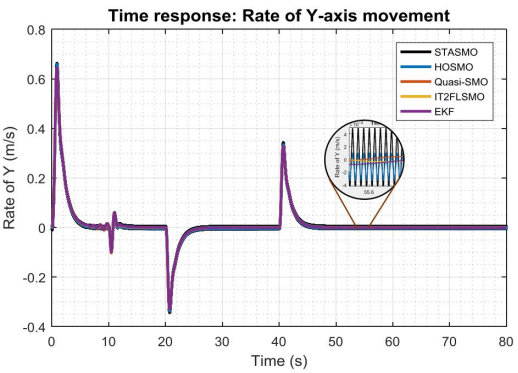

Figure 9.7. Rate of y-axis movement

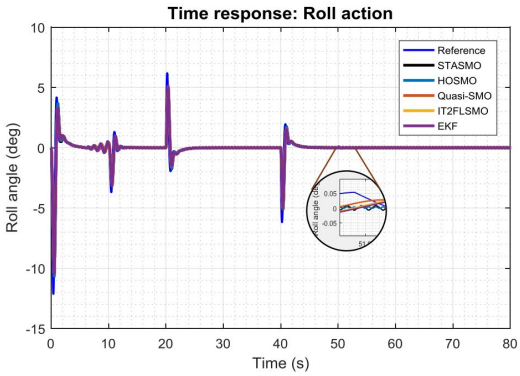

Figure 9.8. Roll action

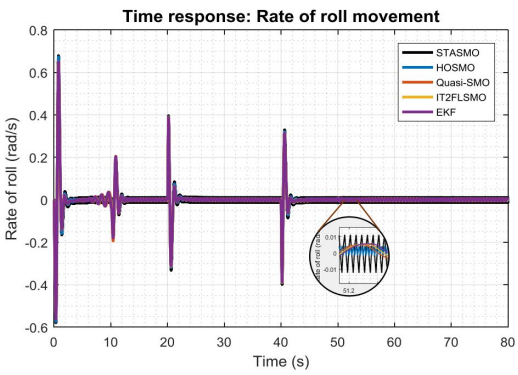

Figure 9.9. Rate of roll movement

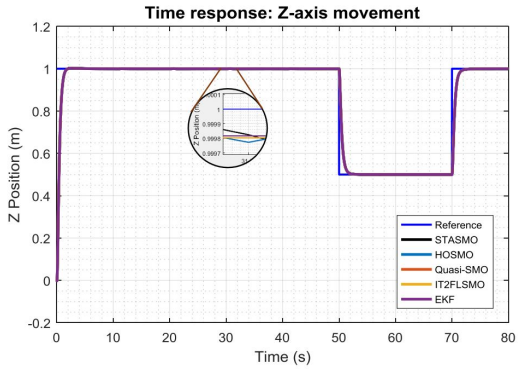

Figure 9.10. z-axis movement

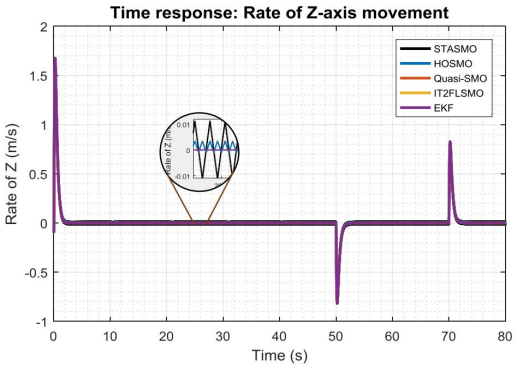

Figure 9.11. Rate of z-axis movement

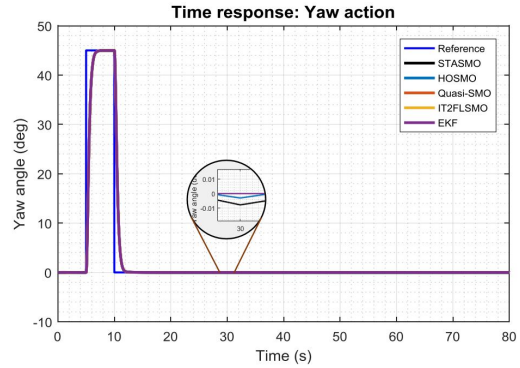

Figure 9.12. Yaw action

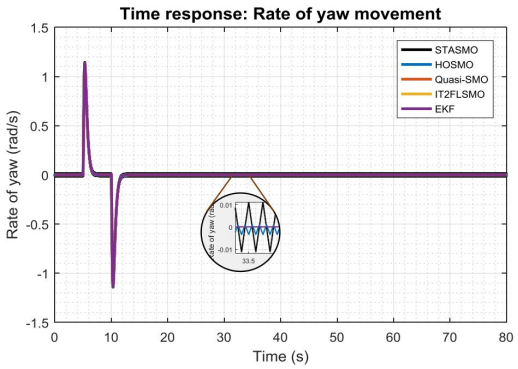

Figure 9.13. Rate of yaw movement

Generally, in terms of observer performance, all observers performed well in estimating unmeasured states of the quadcopter UAV as can be seen in Figures 9.3, 9.5, 9.7, 9.9, 9.11, and 9.13. Significant chattering that is usually seen on second order states of roll and pitch with STASMO and HOSMO was no longer visible clearly as seen in Figures 9.5 and 9.9. However, although the chattering issue has been reduced significantly in STASMO and HOSMO, switching term used in both observers was still the main cause of the slight oscillation on the estimated states.

The MSE of estimated states in steady-state condition as presented in Table 9.1 was

calculated to provide more data for further evaluation of observer performance. From the table, it can be seen that IT2FSMO had smaller MSE in x, pitch and z rate whereas QuasiSMO was better in estimating y and yaw rate states. Furthermore, HOSMO was good in generating estimated state of roll rate. However, the MSE values between QuasiSMO and IT2FSMO were quite close so that both observers were considered to have similar performance in estimating unmeasured states.

Table 9.1. Mean squared states of estimated states

Observer methods	**MSE estimated state of**					
	x rate	**pitch rate**	**y rate**	**roll rate**	**yaw rate**	**z rate**
QuasiSMO	8.98791E-08	1.33916E-05	9.11871E-08	1.34791E-05	8.19673E-10	1.99643E-07
IT2FSMO	8.75774E-08	1.33371E-05	1.08689E-07	1.44075E-05	8.19678E-10	1.97097E-07
STASMO	3.38737E-05	7.40598E-05	3.03218E-05	7.25747E-05	6.28488E-05	6.46195E-05
HOSMO	6.29399E-06	1.36294E-05	6.27793E-06	9.43137E-06	6.61028E-06	6.75966E-06
EKF	9.50554E-08	1.35298E-05	2.97137E-07	1.38534E-05	8.21467E-10	2.02128E-07

Furthermore, in addition to estimating unmeasured states, each observer showed different performance in tracking true states as can be seen in Figures 9.14 - 9.19. QuasiSMO and IT2FSMO showed good performance in tracking true states by showing small estimation errors in steady state condition, whereas EKF performed better than others in states change condition by exhibiting smaller estimation error. MSE was computed to provide better data in measuring the quality of observers. The calculated MSE is presented in Table 9.2. In the condition of without noise and no parameters mismatch, QuasiSMO and IT2FSMO outperformed other methods by showing smaller MSE for pitch, roll, yaw, and z states, whereas EKF was better in x and y than others.

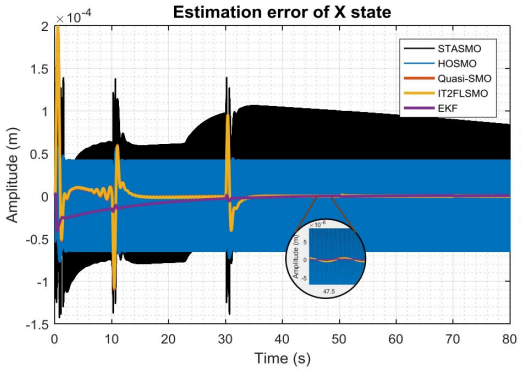

Figure 9.14. Estimation error of x

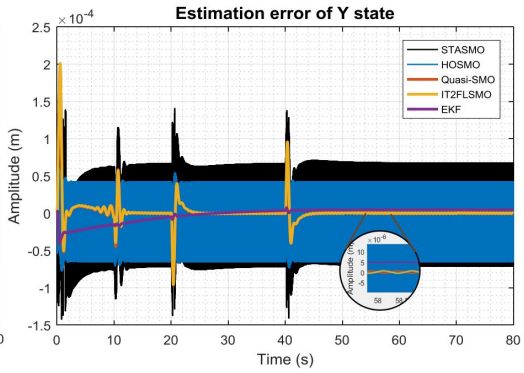

Figure 9.15. Estimation error of y

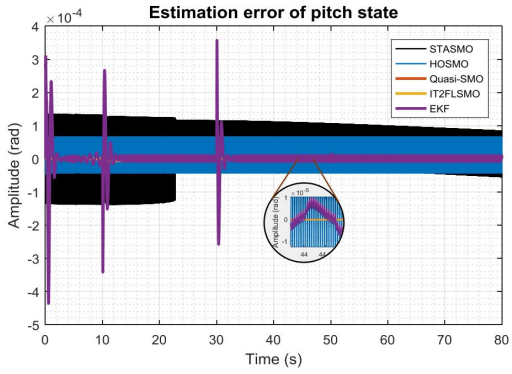

Figure 9.16. Estimation error of pitch

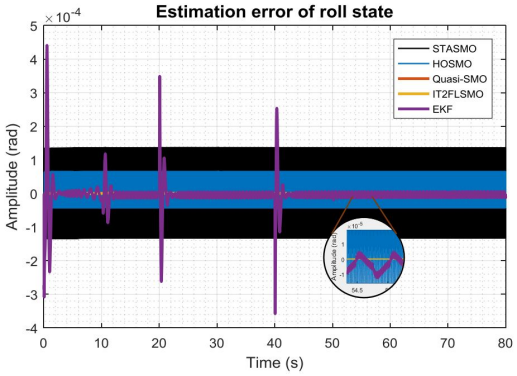

Figure 9.17. Estimation error of roll

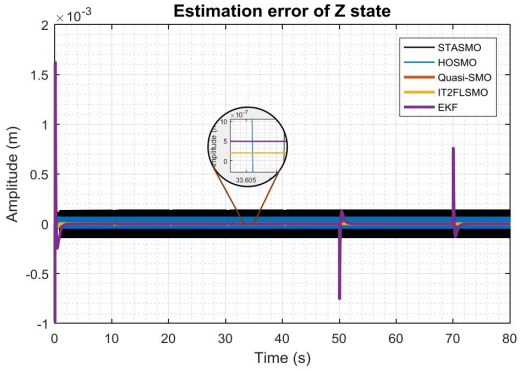

Figure 9.18. Estimation error of z

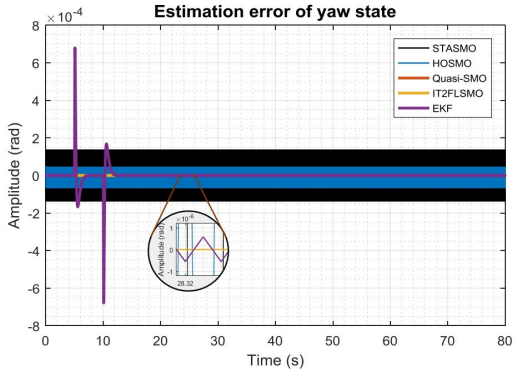

Figure 9.19. Estimation error of yaw

Table 9.2. Mean squared errors of true states tracking

Observer method	MSE true state tracking of					
	x	pitch	y	roll	yaw	z
QuasiSMO	3.11324E-10	2.35233E-12	3.13195E-10	2.44451E-13	6.36673E-15	3.39063E-12
IT2FSMO	3.12907E-10	3.96453E-12	3.15066E-10	7.87942E-14	6.41306E-15	3.40674E-12
STASMO	5.31837E-09	6.70786E-09	4.90389E-09	9.19166E-09	9.21405E-09	9.20483E-09
HOSMO	2.98112E-09	2.97349E-09	2.97376E-09	2.97351E-09	2.97384E-09	2.97571E-09
EKF	7.76156E-11	1.64104E-09	8.68866E-11	1.67363E-09	1.76539E-09	4.30985E-09

In terms of control system, from the simulation results obtained, the issue of time consumption for IT2FSMO was not major problem as the selected controller could stabilize altitude and attitude of the quadcopter UAV smoothly. It can be proved by showing good performance of the controller in tracking references as presented in Figures 9.2 - 9.13. Significant chattering that usually follows the use of STASMO and HOSMO, was no longer visible clearly. This means that DSMC plays a vital role in improving the observers performance in estimating unmeasured states. In addition, the time demand to process the control was faster than previous controllers around 0.1719 ms.

The DSMC showed good performance in controlling the copter with various observers.

It can be seen in the figures that there was no overshoot in responding to step input for all observers methods with similar rise time. However, significant oscillations were seen in the control inputs of roll and pitch actions using STASMO as can be seen in Figures 9.21 and 9.22, whereas QuasiSMO, IT2FSMO and EKF generated smoother control signals for the aircraft.

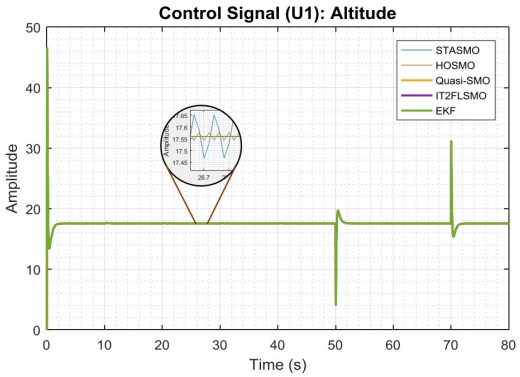

Figure 9.20. Control signal u_1

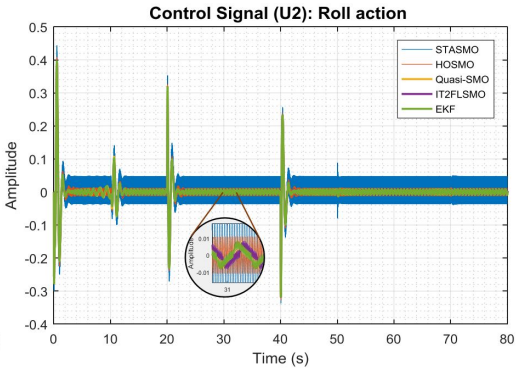

Figure 9.21. Control signal u_2

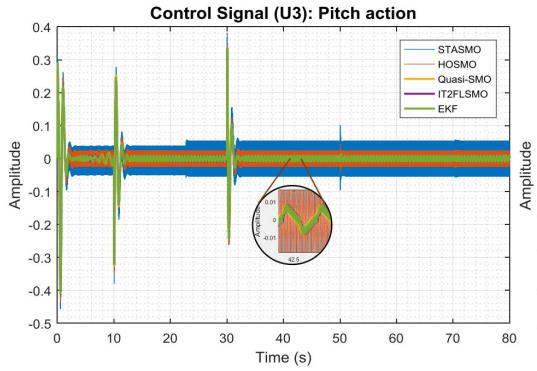

Figure 9.22. Control signal u_3

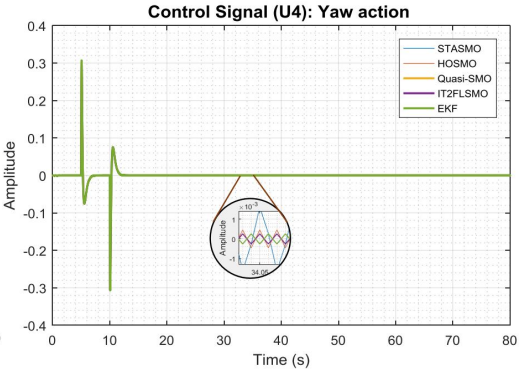

Figure 9.23. Control signal u_4

Steady-state error of output state is another factor which needs to be taken into account for evaluation of performance of selected controller which relates to the use of several observers. Therefore, the MSE was computed as presented in Table 9.3. Generally, all observers showed similar performance in improving controller to reduce steady-states errors in almost all states except in yaw state for which QuasiSMO performed better than others.

Table 9.3. Mean squared errors of steady-states errors

Observer methods	MSE steady-state error of					
	x	pitch	y	roll	yaw	z
QuasiSMO	3.32422E-07	4.04871E-07	3.5677E-07	4.07073E-07	1.88658E-13	1.87105E-08
IT2FSMO	3.43706E-07	3.77827E-07	3.54602E-07	6.28935E-07	1.86451E-13	1.88812E-08
STASMO	2.51545E-06	3.68175E-07	5.99312E-07	3.76505E-07	9.47006E-09	2.87283E-08
HOSMO	3.76194E-07	3.64405E-07	3.53125E-07	3.69027E-07	2.87142E-09	2.15704E-08
EKF	8.713E-08	4.45222E-07	1.06684E-07	6.52378E-07	3.55165E-13	1.91677E-08

In summary, in the condition of no noise and no parameters mismatch, DSMC has shown good performance in controlling the dynamics of the quadcopter UAV with no response overshoot, small rise time, very small chattering, and small steady-states errors. In addition, the controller had faster computational time than other sliding mode-based controllers. However, DSMC is constructed by two sliding surfaces and many parameters need to be tuned which eventually the stability of the system would be difficult to be evaluated. In terms of observer, QuasiSMO method has outperformed other methods by showing smooth estimated states, smaller states estimation errors, and faster in tracking true states. Therefore, the observer has influenced and improved the overall performance of selected controller.

9.2.2 Without noise and with parameter mismatch

In this case, the robustness of DSMC and observer approaches is tested in dealing with high uncertainty of the inertia moment of the rotor-craft system. The responses of the proposed approaches are investigated through simulation works and presented as in Figures 9.24 – 9.45.

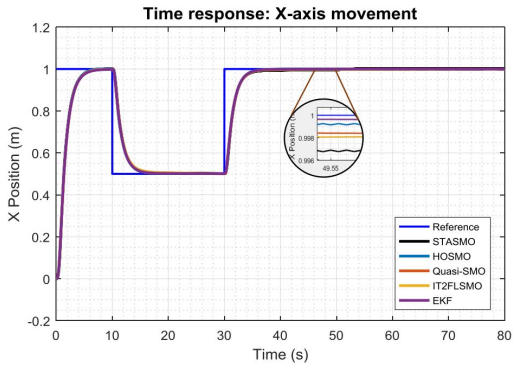

Figure 9.24. x-axis movement

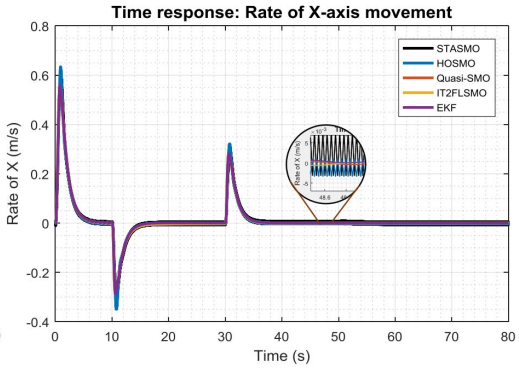

Figure 9.25. Rate of x-axis movement

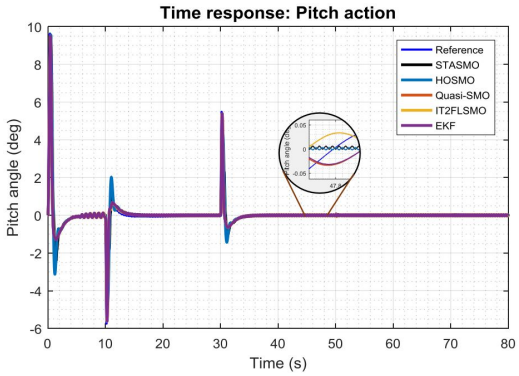

Figure 9.26. Pitch action

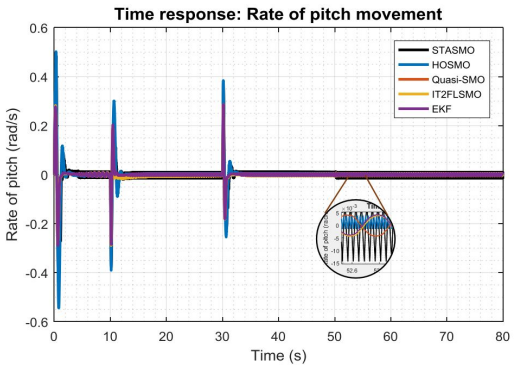

Figure 9.27. Rate of pitch movement

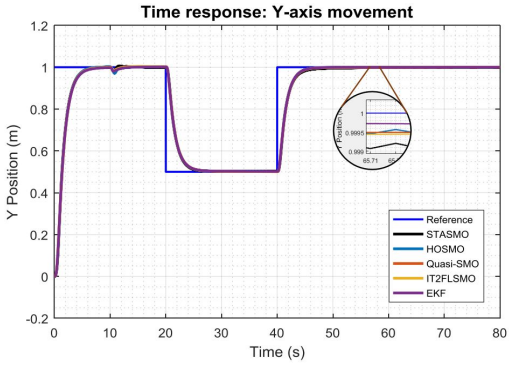

Figure 9.28. y-axis movement

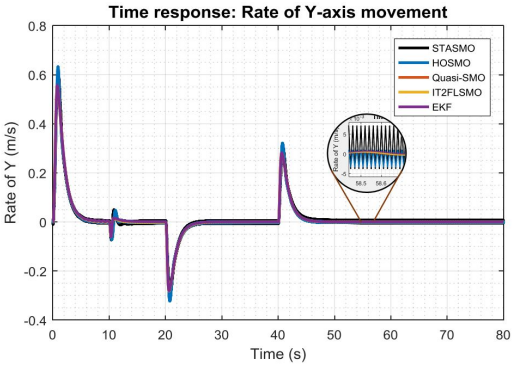

Figure 9.29. Rate of y-axis movement

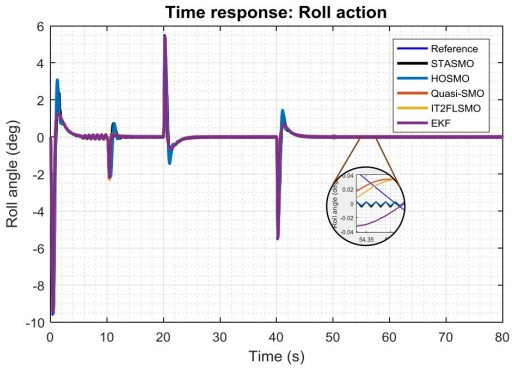

Figure 9.30. Roll action

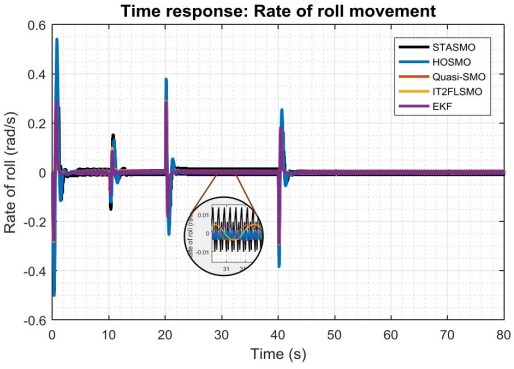

Figure 9.31. Rate of roll movement

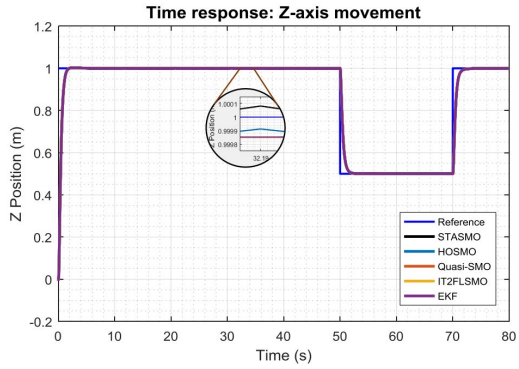

Figure 9.32. z-axis movement

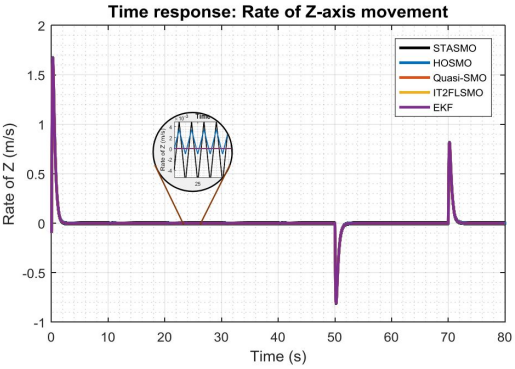

Figure 9.33. Rate of z-axis movement

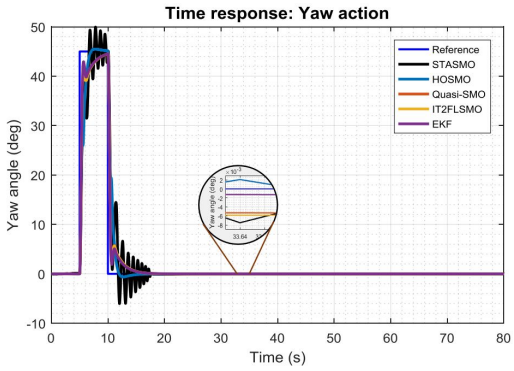

Figure 9.34. Yaw action

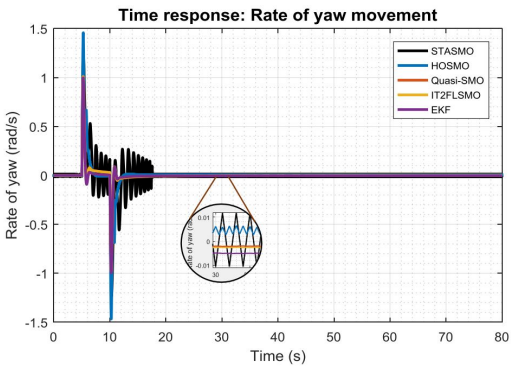

Figure 9.35. Rate of yaw movement

From the simulation results obtained as presented in Figures 9.25, 9.27, 9.29, 9.31, 9.33, and 9.35, it can be seen that generally the proposed control system and observer methods have performed adequately in estimating the unknown internal states of the rotor-craft system. This circumstance has only affected significantly the performance of STASMO in estimating the rate of yaw. Furthermore, oscillation occurred with yaw state changes. However, other observers were more robust than STASMO in dealing with parameters mismatch by demonstrating similar performance as the earlier case (no parameters mismatch).

Observing the calculated MSE as presented in Table 9.4, it can be noticed that QuasiSMO outperformed other methods in estimating unknown states by showing smaller values of MSE in almost all states except pitch and roll rates. In those two states HOSMO performed better than others. However, the MSE values of pitch and roll rates were quite close for almost all observers except STASMO. It can be concluded that the observers had similar performance in case of estimating the pitch and roll rates.

Table 9.4. Mean squared errors of estimated states

Observer methods	**MSE estimated state of**					
	x rate	**pitch rate**	**y rate**	**roll rate**	**yaw rate**	**z rate**
QuasiSMO	7.5777E-08	9.60827E-06	7.994E-08	9.7938E-06	3.41868E-06	1.31521E-07
IT2FSMO	1.01107E-07	9.76078E-06	8.12934E-08	9.8397E-06	4.07618E-06	1.32071E-07
STASMO	3.28073E-05	7.124E-05	3.30127E-05	4.86157E-05	6.30901E-05	3.26054E-05
HOSMO	6.25888E-06	8.52274E-06	6.24759E-06	8.52135E-06	2.35667E-05	6.71885E-06
EKF	1.91321E-07	8.96343E-06	1.66554E-07	9.10002E-06	1.7624E-05	1.33472E-07

Furthermore, in terms of tracking true states, Figures 9.36 - 9.41 showed different responses of observers methods. Generally, all approaches performed well in tracking true states by presenting small estimation errors for all measured states. However, some observers including QuasiSMO, IT2FSMO, and EKF performed better than STASMO and

HOSMO. Fluctuated errors were still noticeable using STASMO and HOSMO methods. QuasiSMO and IT2FSMO showed good performance by showing smaller estimation errors in steady-state condition than EKF, whereas EKF was better than those two in states change condition as seen in Figures 9.36 and 9.37. The overall performance of tracking true states was measured by MSE approach as presented in Table 9.5. From the table it can be seen that QuasiSMO was better than others in tracking the states of y ad z, whereas EKF performed better in tracking x and yaw states. HOSMO showed smallest MSE value of state estimation for the remaining two states: pitch and roll.

Table 9.5. Mean squared errors of true states tracking

Observer method	MSE true state tracking of					
	x	pitch	y	roll	yaw	z
QuasiSMO	1.92511E-10	1.30032E-07	8.75715E-11	1.30045E-07	9.33604E-07	5.62547E-13
IT2FSMO	2.73852E-10	1.31264E-07	9.75583E-11	1.30326E-07	9.38151E-07	5.70332E-13
STASMO	5.0822E-09	2.2965E-06	5.27134E-09	2.02359E-06	0.000707775	4.88E-09
HOSMO	2.97302E-09	1.35226E-08	2.97281E-09	1.60765E-08	2.83895E-06	2.97567E-09
EKF	1.25382E-10	9.59323E-08	1.34388E-10	9.71362E-08	7.15995E-07	4.33622E-09

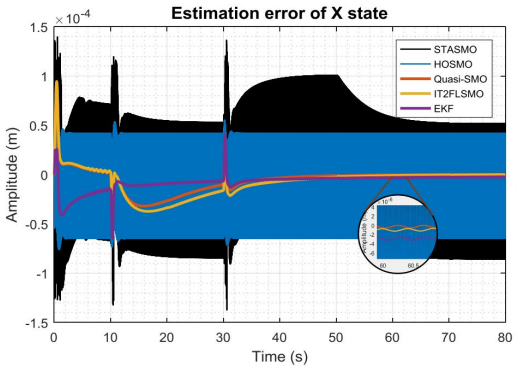

Figure 9.36. Estimation error of x

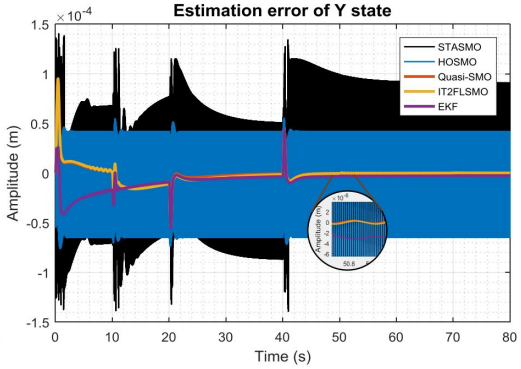

Figure 9.37. Estimation error of y

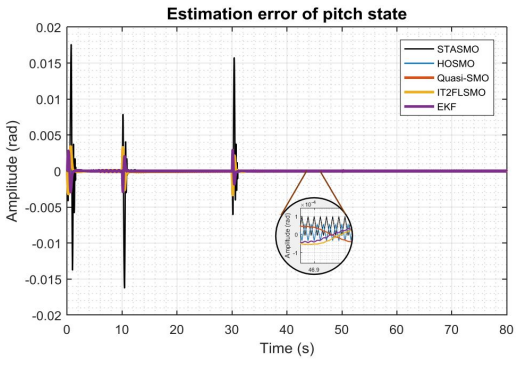

Figure 9.38. Estimation error of pitch

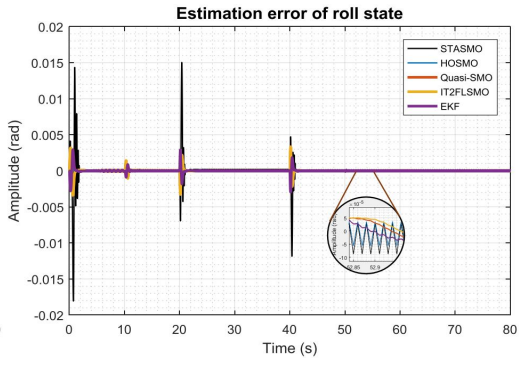

Figure 9.39. Estimation error of roll

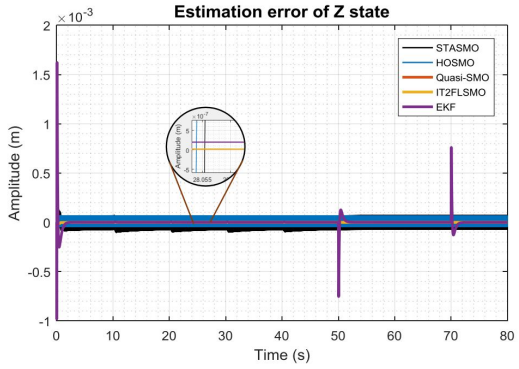

Figure 9.40. Estimation error of z

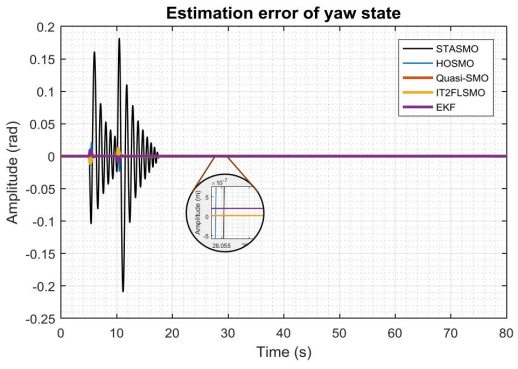

Figure 9.41. Estimation error of yaw

In the use of DSMC, the presence of parameter uncertainty only influenced significantly the control performance in managing yaw state when STASMO was employed, as seen in Figure 9.34. Oscillation occurred when the reference changed in amplitude around $14.5°$. The selected controller performed well in handling the unforeseen circumstance and the effect of chattering phenomenon which was generated by switching term in the control system and some of observer methods including STASMO and HOSMO, by generating smooth responses of step inputs signals except yaw state.

However, high frequency chattering was still clearly visible in the control inputs using all observers methods for u_2 and u_3 control signals especially STASMO as can be seen in Figures 9.42 - 9.45. Meanwhile, smoother control signals were shown in u_1 and u_4 using the observers.

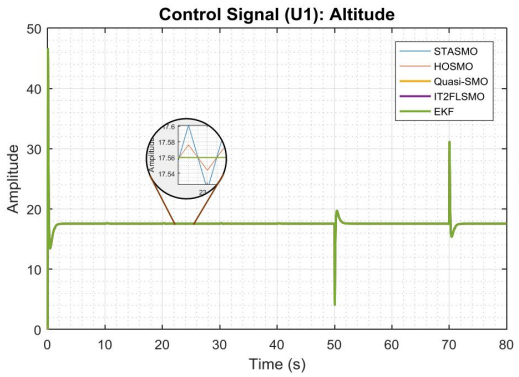

Figure 9.42. Control signal u_1

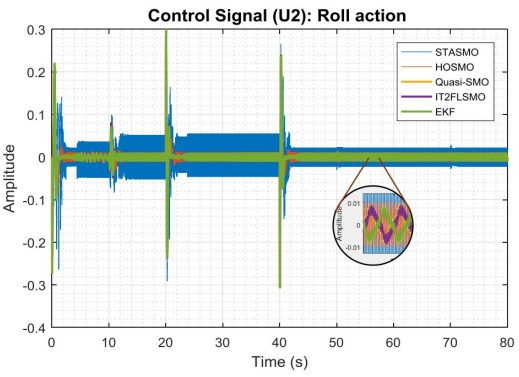

Figure 9.43. Control signal u_2

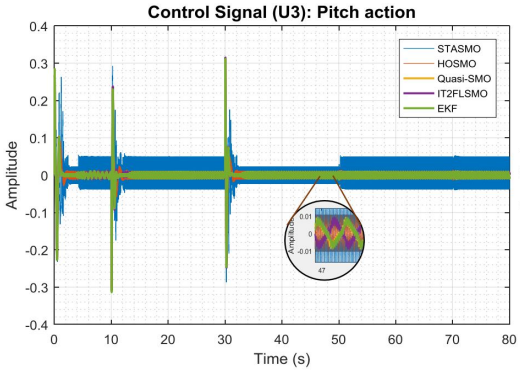

Figure 9.44. Control signal u_3

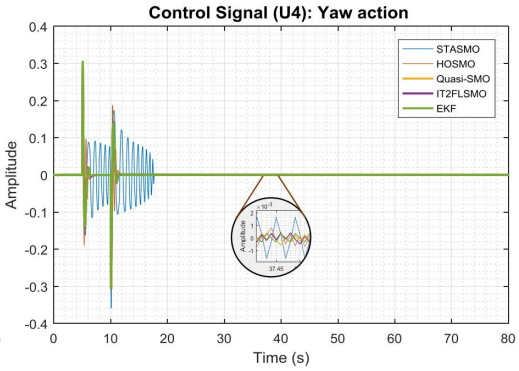

Figure 9.45. Control signal u_4

Further evaluation of control system performance is steady-states errors of output states of the quadcopter UAV. This performance appraisal is to consider the use of several observer techniques. In the case of parameter uncertainty, HOSMO outperformed slightly other observers in enhancing the performance of DSMC as can be seen in Table 9.6. HOSMO improved the performance of selected controller by demonstrating smaller steady-states errors of roll, pitch, and yaw motion, whereas EKF performed better in x and y states. QuasiSMO performed better in enhancing DSMC in z state only.

Table 9.6. Mean squared errors of steady-states errors

Observer methods	**MSE steady-state error of**					
	x	**pitch**	**y**	**roll**	**yaw**	**z**
QuasiSMO	1.64374E-06	3.22723E-07	5.56308E-07	3.28028E-07	1.00702E-08	1.32603E-08
IT2FSMO	2.1632E-06	6.57108E-07	6.4288E-07	4.30785E-07	7.1979E-09	1.32667E-08
STASMO	7.66778E-06	2.59209E-07	2.53076E-06	2.55677E-07	1.46935E-08	4.95726E-08
HOSMO	4.38767E-07	2.47791E-07	4.60433E-07	2.53333E-07	1.47395E-09	1.95162E-08
EKF	1.21241E-07	4.43052E-07	1.57101E-07	4.90834E-07	1.89095E-09	1.36265E-08

In summary, in the presence of parameters mismatch, DSMC still demonstrated good performance to control the dynamics of the quadcopter UAV with no response overshoot except for the use of STASMO in yaw, small rise time using STASMO and HOSMO in yaw movement, small chattering appeared in the use STASMO and HOSMO, and small steady-states errors. In addition, the controller had faster computational time than other sliding mode-based controllers. However, DSMC is constructed by two sliding surfaces and many parameters need to be tuned which stability might be difficult to be evaluated. In terms of observer, QuasiSMO, EKF, and HOSMO method outperformed other methods by demonstrating smooth estimated states, smaller states estimation errors, and faster in tracking true states. Therefore, the observer has influenced and improved the overall

performance of selected controller.

9.2.3 With noise and no parameter mismatch

In the case of corrupted output states with high power white Gaussian noise, the simulation results obtained are presented as in Figures 9.46 – 9.67.

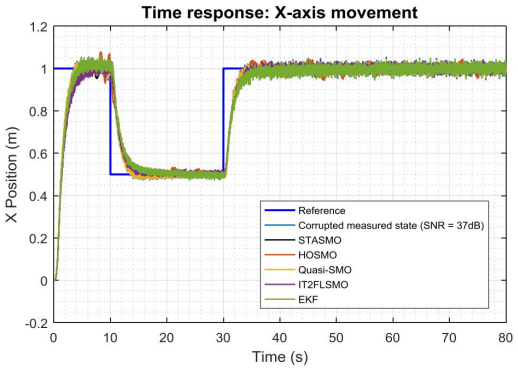

Figure 9.46. x-axis movement

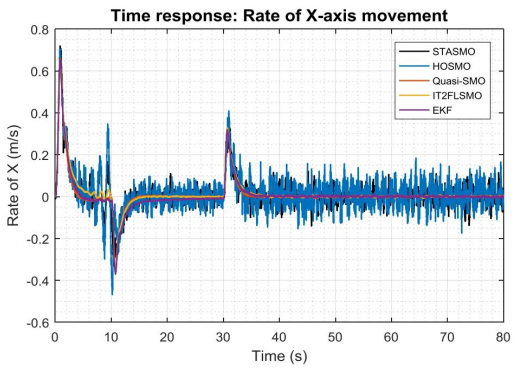

Figure 9.47. Rate of x-axis movement

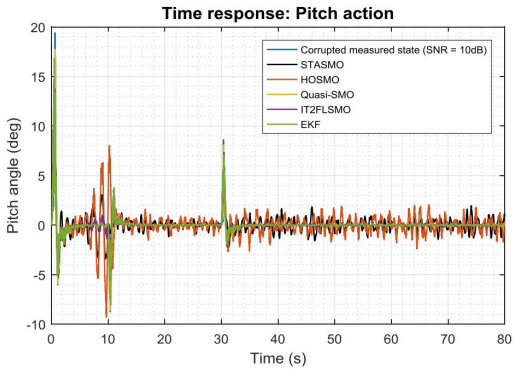

Figure 9.48. Pitch action

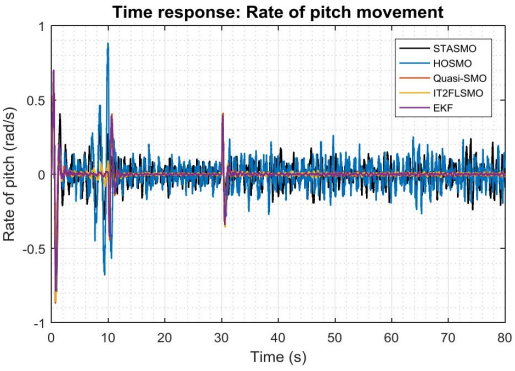

Figure 9.49. Rate of pitch movement

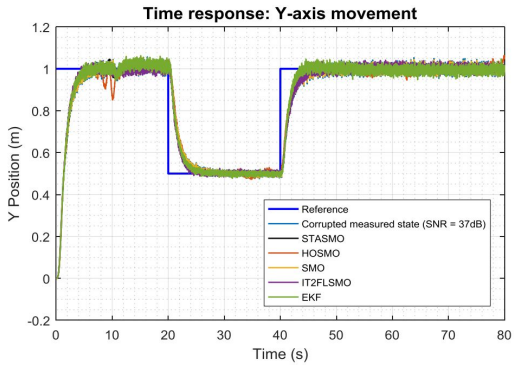

Figure 9.50. y-axis movement

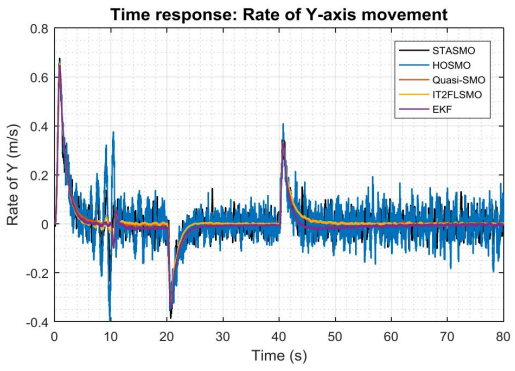

Figure 9.51. Rate of y-axis movement

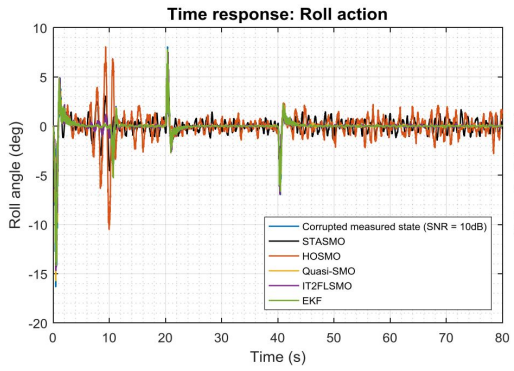

Figure 9.52. Roll action

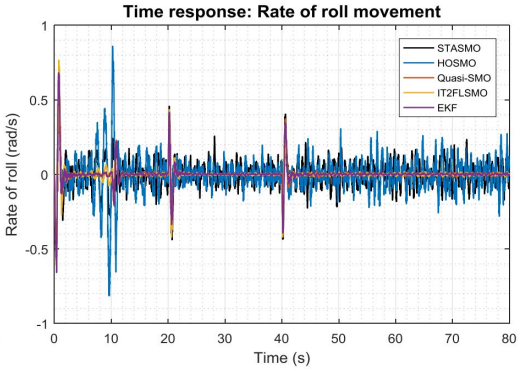

Figure 9.53. Rate of roll movement

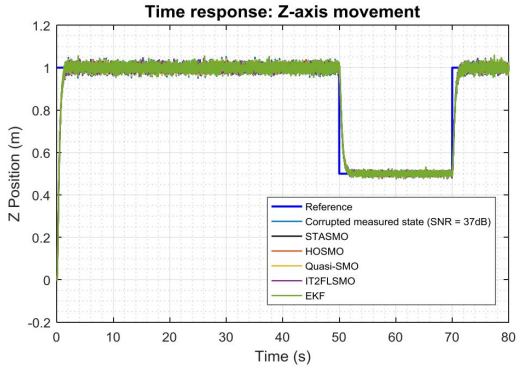

Figure 9.54. z-axis movement

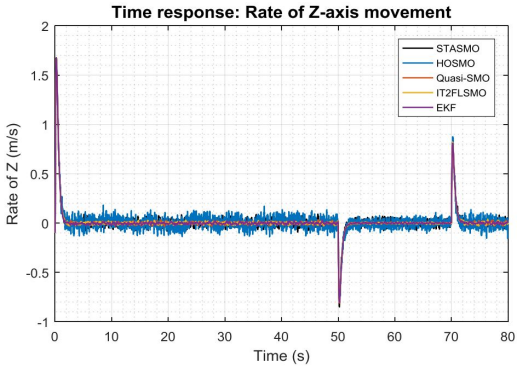

Figure 9.55. Rate of z-axis movement

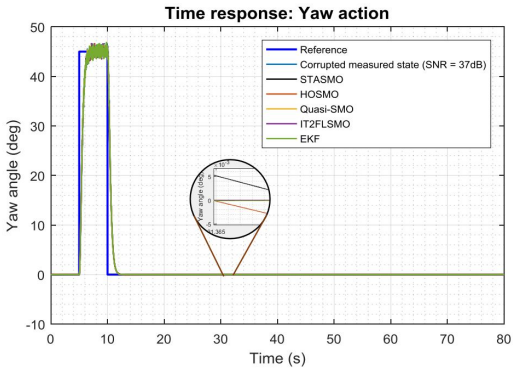

Figure 9.56. Yaw action

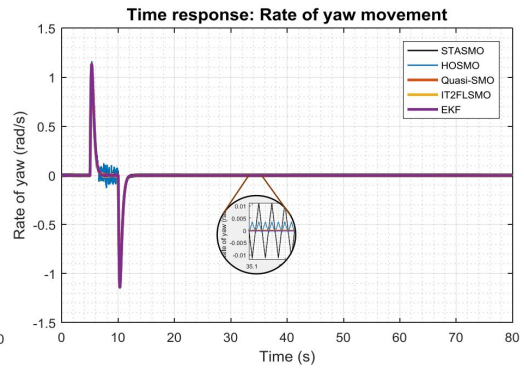

Figure 9.57. Rate of yaw movement

It is noted in the simulation results presented in Figures 9.47, 9.49, 9.51, 9.53, 9.55, and 9.57, that the effect of noise was reduced sufficiently by QuasiSMO, IT2FSMO, and EKF so that second order states were generated quite smoothly. Noise disturbance on the output states did not affect significantly the performance of the observers. In contrast, corrupted output states influenced the performance of STASMO and HOSMO significantly for all second order states as presented in the figures. The noise amplitude on the output states was amplified high enough due to the effect of switching term on the methods.

Although the SNR on the output states x and y were the same as yaw and z states, that

is 37dB where it was smaller than the noise on the pitch and roll states, that is 10dB, the effect of noise received by x and y was greater than yaw and z. This is because x and pitch, y and roll are dependent states which will affect each other.

MSE of estimated states in steady-state condition was calculated for further evaluation of observer performance as presented in Table 9.7. In spite of noise disturbance, the three observers showed high quality performance of estimating unknown states. In addition to smooth estimation results, QuasiSMO, IT2FSMO and EKF generated the second order states with small MSE.

Table 9.7. Mean squared errors of estimated states

Observer methods	**MSE estimated state of**					
	x rate	**pitch rate**	**y rate**	**roll rate**	**yaw rate**	**z rate**
QuasiSMO	5.88653E-06	4.76559E-05	5.51029E-06	4.31481E-05	8.23193E-10	9.53212E-05
IT2FSMO	4.99664E-06	5.0548E-05	5.83385E-06	4.57998E-05	8.24014E-10	8.61567E-05
STASMO	0.001415418	0.005120557	0.001449294	0.005223281	6.28518E-05	0.000887607
HOSMO	0.002681594	0.006468115	0.003286927	0.009069879	6.6102E-06	0.002308482
EKF	6.47614E-06	2.92644E-05	6.82047E-05	2.3897E-05	2.92674E-08	7.40301E-05

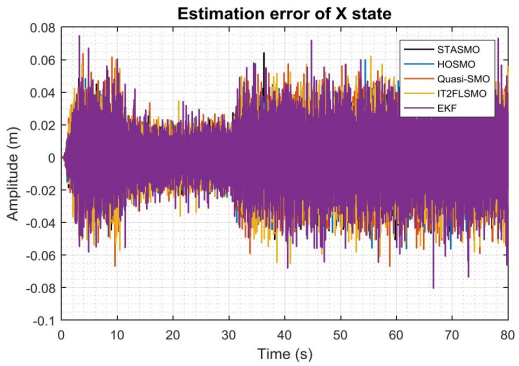

Figure 9.58. Estimation error of x

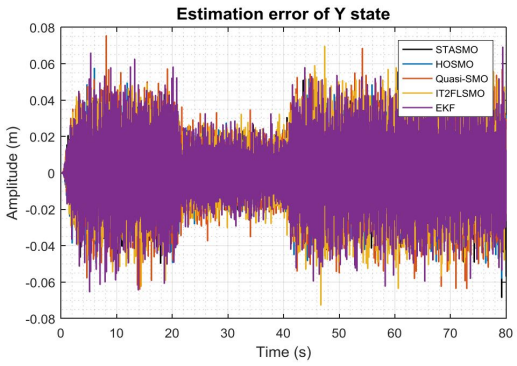

Figure 9.59. Estimation error of y

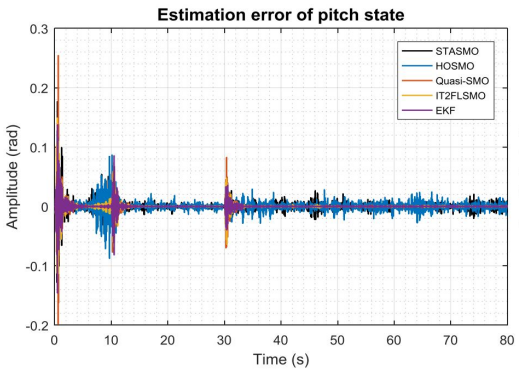

Figure 9.60. Estimation error of pitch

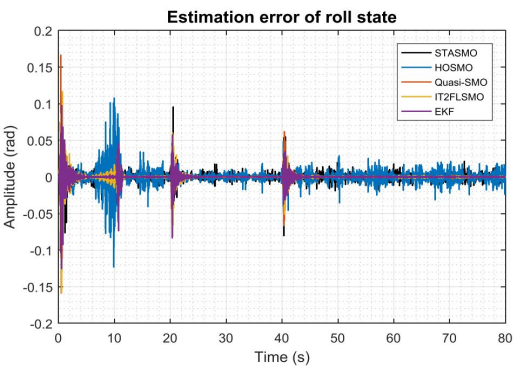

Figure 9.61. Estimation error of roll

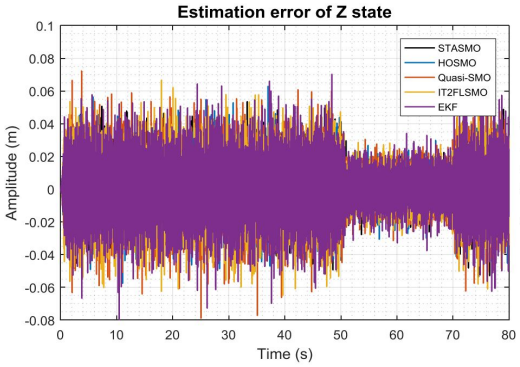

Figure 9.62. Estimation error of z

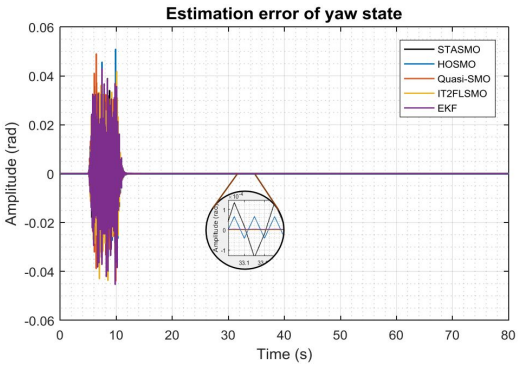

Figure 9.63. Estimation error of yaw

In the presence of noise, true states tracking errors appeared similar for all observer methods as presented in Figures 9.58 - 9.63. However, noting the computed MSE as presented in Table 9.8, each observer technique showed different performance in tracking true states. In this condition, STASMO and HOSMO performed well to track true states even though could not estimate the unknown states as smooth as QuasiSMO, IT2FSMO and EKF. It is noticeable in Table 9.8 that STASMO and HOSMO had smaller MSE than other approaches in tracking the measured states of x, y, yaw and z, while EKF showed better performance in tracking roll and pitch states. However, the ability of STASMO and HOSMO approaches in estimating second order states was not as well as other approaches, and this might affect the overall control system performance.

Table 9.8. Mean squared errors of true states tracking

Observer method	**MSE true state tracking of**					
	x	**pitch**	**y**	**roll**	**yaw**	**z**
QuasiSMO	0.000270799	5.84768E-05	0.000274679	4.04991E-05	1.22929E-05	0.000292735
IT2FSMO	0.000274325	4.67145E-05	0.000277059	3.8046E-05	1.26819E-05	0.000280666
STASMO	0.000179126	4.83635E-05	0.000170273	3.8076E-05	7.94895E-06	0.000179221
HOSMO	0.000184617	6.59907E-05	0.000185941	8.45685E-05	9.0241E-06	0.000187726
EKF	0.000315013	3.51557E-05	0.000306179	2.88234E-05	1.46743E-05	0.00031421

Noise issue is an inevitable fact in real-time applications and will affect the overall performance of controller. However, the effect of noise on systems depends on how to manage and select a proper control and observer system. Note in Figures 9.46, 9.48, 9.50, 9.52, 9.54, and 9.56, that generally all observers influenced the performance of selected controller similarly. Only slight overshoot occurred in x-axis movement around 8.0% and 4.5% when using HOSMO and STASMO respectively, meanwhile in y-axis movement, the overshoots were around 15.0% and 4.5% for HOSMO and STASMO respectively. Higher

overshoots were noticeable clearly in controlling pitch by around 8.02° and -3.4° using HOSMO and STASMO respectively, while in roll state the overshoots were about -10.5° and -4.55° for HOSMO and STASMO respectively.

In contrast, the selected controller performed in a more robust manner in controlling all output states with QuasiSMO, IT2FSMO, and EKF. However, the noise effect was still noticeable clearly around output states and control inputs as seen in Figures 9.46 - 9.56. This is the trade-off of choosing parameters for the observers. on the one hand, increasing or decreasing observer gains will speed up tracking true value, yet may potentially increase sensitivity to measurement noise and vice versa. In this research, the speed of tracking measured states was more considered (without neglecting noise effect) to make the controller more responsive.

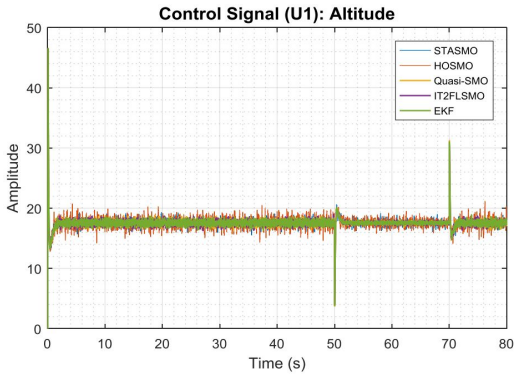

Figure 9.64. Control signal u_1

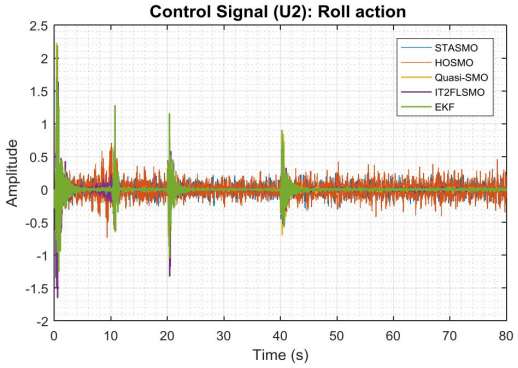

Figure 9.65. Control signal u_2

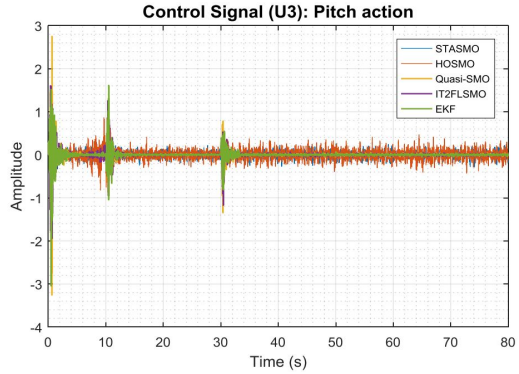

Figure 9.66. Control signal u_3

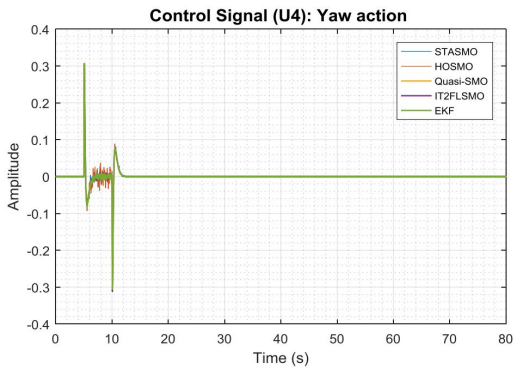

Figure 9.67. Control signal u_4

Steady-states error of output states was considered also to evaluate the robustness of selected controller and observers in dealing with noise disturbance. Note in Table 9.9, that although STASMO had problem in handling noise effect in estimating unmeasured states, the observer influenced control system to reduce steady-state error quite well. STASMO

mostly had smaller steady-state error (except yaw) computed by MSE method than other approaches. Switching term in the algorithms had the main role in keeping observed state tracking measured state all the time. However, due to noise disturbance, the control signals as seen in Figures 9.64 - 9.67 were not suitable for real-time embedded systems as might be harmful for hardware devices.

Table 9.9. Mean squared errors of steady-states errors

Observer methods	MSE steady-state error of					
	x	pitch	y	roll	yaw	z
QuasiSMO	0.000160792	0.000176605	0.000153683	0.00017989	3.74803E-13	0.000166054
IT2FSMO	0.000149721	0.000174484	0.000162635	0.000185667	1.83345E-13	0.000148906
STASMO	7.40011E-05	0.000143316	6.72557E-05	0.000143428	9.50965E-09	3.81508E-05
HOSMO	0.00035611	0.000310714	0.000324423	0.000371131	2.87551E-09	0.000125861
EKF	0.000227426	0.000177123	0.000221934	0.000182748	2.5127E-12	0.00019205

In summary, in the presence of noise, DSMC maintained good performance in controlling the dynamics of the quadcopter UAV with no response overshoot except for x and y movements, pitch and roll actions when applying STASMO and HOSMO, considered small rise time, free from chattering when using QuasiSMO, IT2FSMO and EKF observer method, small steady-states errors, and faster computational time. Furthermore, random fluctuated signals on the references appeared due to noise disturbance. The selection of observer gain was a trade-off. Reducing or increasing the gain may make observers become insensitive to noise, however, on the other hand it may delay tracking true states. As for the observers, in terms of tracking true states and improving controller performance in reducing steady-states error of output states, STASMO outperformed other methods by showing smaller states estimation errors, and small steady-states error in output states. Therefore, the observer has influenced and improved the overall performance of selected controller.

9.2.4 With noise and with parameter mismatch

At this stage of testing the performance of control system and observers, noise and uncertainty were introduced at the same time into the quadcopter UAV system. The behaviours of the rotor-craft system are presented in Figures 9.68 - 9.89.

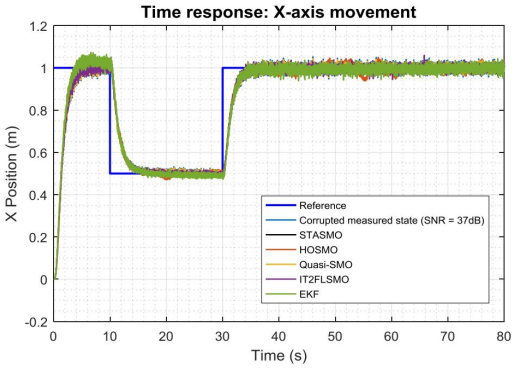

Figure 9.68. x-axis movement

Figure 9.69. Rate of x-axis movement

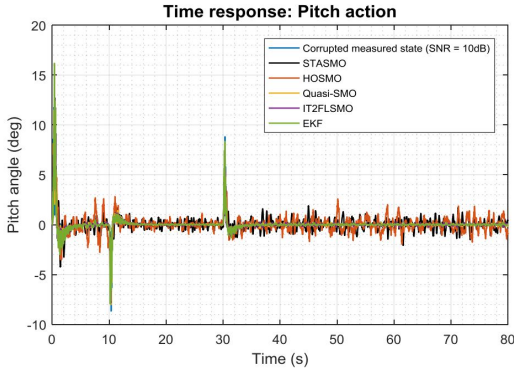

Figure 9.70. Pitch action

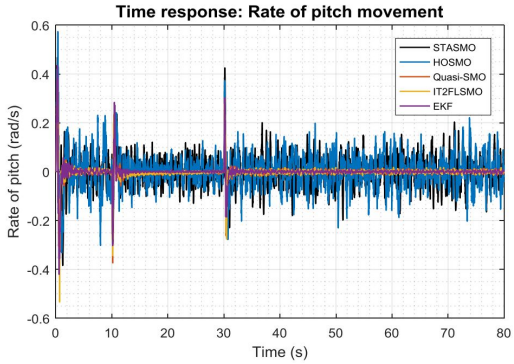

Figure 9.71. Rate of pitch movement

Figure 9.72. y-axis movement

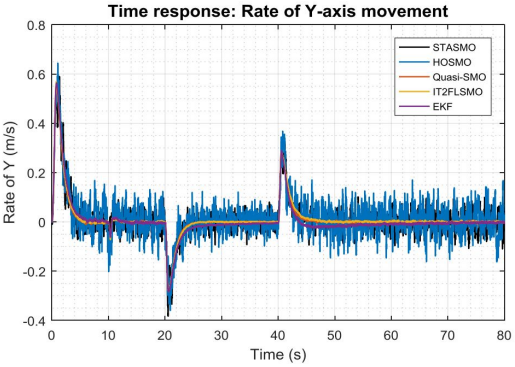

Figure 9.73. Rate of y-axis movement

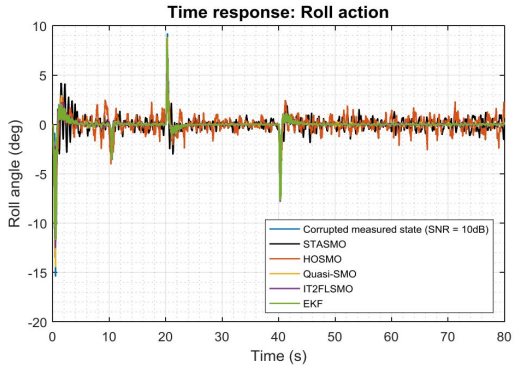

Figure 9.74. Roll action

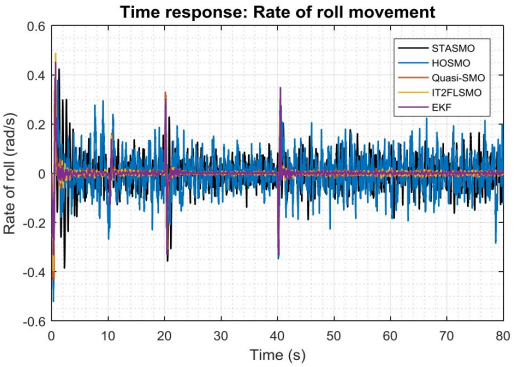

Figure 9.75. Rate of roll movement

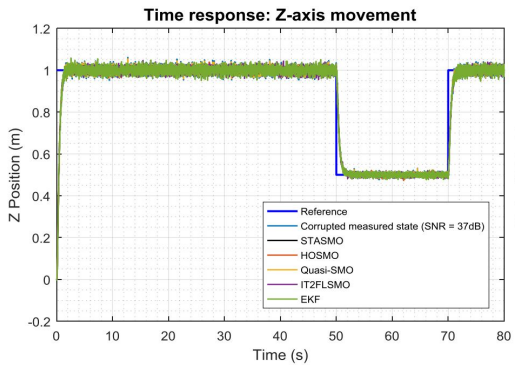

Figure 9.76. z-axis movement

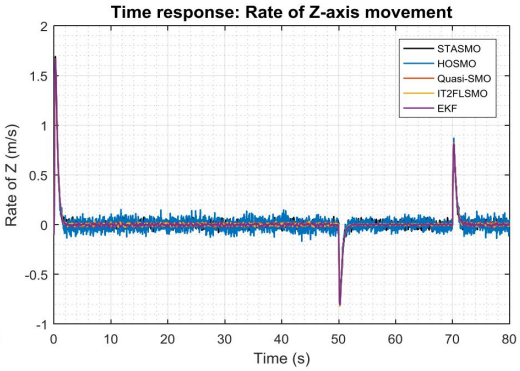

Figure 9.77. Rate of z-axis movement

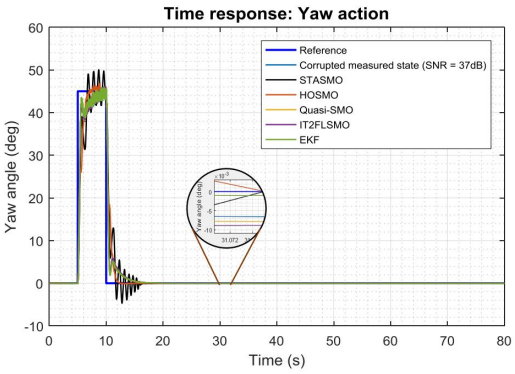

Figure 9.78. Yaw action

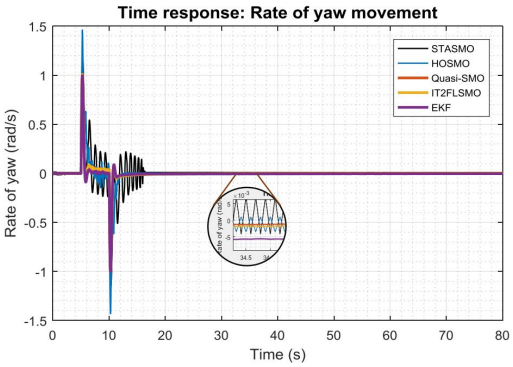

Figure 9.79. Rate of yaw movement

The same noise power was introduced as the previous case with 50% parameters mismatch of inertia moment. It is noted in the simulation results presented in Figures 9.69, 9.71, 9.73, 9.75, 9.77, and 9.79, that the effect of disturbances was reduced adequately by the three observer approaches QuasiSMO, IT2FSMO, and EKF so that second order states were generated quite smoothly. In other words, noise disturbance on the output states as well as parameters mismatch did not affect significantly the performance of the observers. In contrast, corrupted output states influenced the performance of STASMO and HOSMO significantly, particularly in estimating second order states of pitch, roll, and yaw as shown in the figures. The noise amplitude on the output states were amplified high enough due to the effect of switching term on the methods. In addition, STASMO could not deal with parameters mismatch well. It can be noticed that this observer resulted high amplitude oscillation of the estimation of yaw rate.

MSE of estimated states in steady-state condition as presented in Table 9.10 was calculated to evaluate the performance of observers numerically. Noise disturbance and uncertainty are inevitable conditions which will disturb the performances of observers. However,

as noted in the table, the three estimators QuasiSMO, IT2FSMO, and EKF maintained their good performance in estimating unmeasured states. The observers showed good accuracy in predicting the unknown states by showing smaller MSE than STASMO and HOSMO.

Table 9.10. Mean squared errors of estimated states

Observer methods	MSE estimated state of					
	x rate	pitch rate	y rate	roll rate	yaw rate	z rate
QuasiSMO	4.71813E-06	3.22091E-05	5.67621E-06	3.87957E-05	3.42127E-06	8.31377E-05
IT2FSMO	4.78787E-06	3.45078E-05	5.0615E-06	3.24607E-05	4.6615E-06	8.25333E-05
STASMO	0.001408557	0.003987265	0.001360913	0.003837892	3.28468E-05	0.000758379
HOSMO	0.002667377	0.004578211	0.002849846	0.005344991	6.56119E-06	0.001901026
EKF	1.54249E-05	1.54136E-05	8.22583E-05	1.85355E-05	2.43512E-05	8.2426E-05

In the presence of noise and parameter mismatch, as noted in Figures 9.80 - 9.85 all observer methods showed similar true states tracking errors. However, noting the computed MSE as presented in Table 9.11, each observer technique showed different performance in tracking true states. It is noticeable in the table that STASMO and HOSMO showed better performance in tracking true states of x, y, yaw and z by presenting smaller MSE than other approaches, while QuasiSMO and IT2FSMO showed better performance in tracking pitch and roll states. However, the ability of STASMO and HOSMO in estimating second order states was not as well as QuasiSMO, IT2FSMO and EKF so that the overall control performance migh be affected.

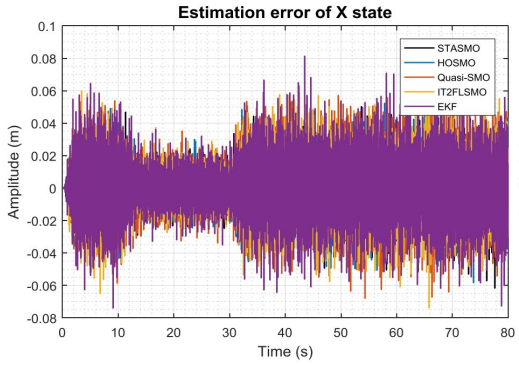

Figure 9.80. Estimation error of x

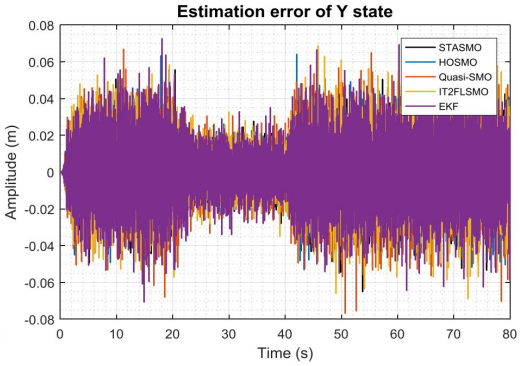

Figure 9.81. Estimation error of y

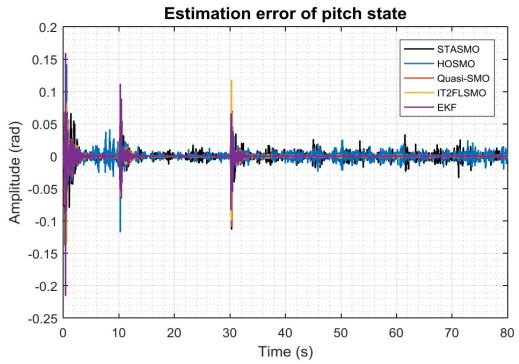

Figure 9.82. Estimation error of pitch

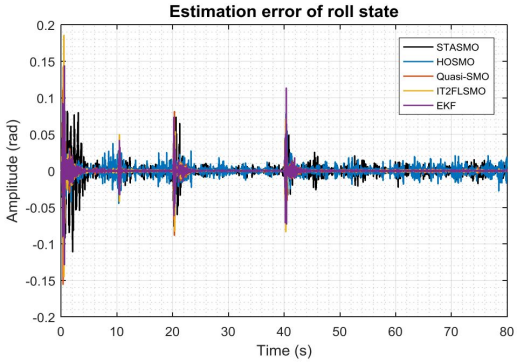

Figure 9.83. Estimation error of roll

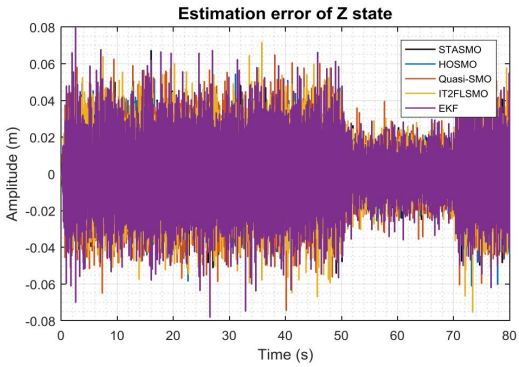

Figure 9.84. Estimation error of z

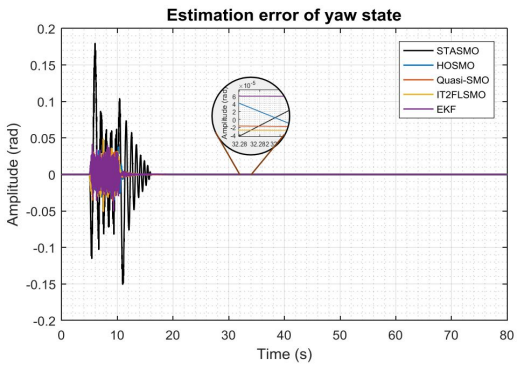

Figure 9.85. Estimation error of yaw

Table 9.11. Mean squared errors of true states tracking

Observer method	MSE true state tracking of					
	x	**pitch**	**y**	**roll**	**yaw**	**z**
QuasiSMO	0.000277102	2.31257E-05	0.000278138	4.0883E-05	1.19431E-05	0.000285373
IT2FSMO	0.000276616	2.86099E-05	0.000271504	3.22704E-05	1.18428E-05	0.000286913
STASMO	0.000172541	7.25988E-05	0.000173697	0.00010442	0.000445377	0.000177637
HOSMO	0.000189739	4.28961E-05	0.000182192	3.85983E-05	1.17318E-05	0.000192498
EKF	0.000324588	3.9194E-05	0.000299838	3.50203E-05	1.37551E-05	0.000325291

The presence of noise and parameter mismatch might affect the performance of controller. However, the effect of those unforeseen circumstances depends on how to manage and select control and observer systems. As noted in Figures 9.68, 9.70, 9.72, 9.74, 9.76, and 9.78, the use of all type of observer approaches did not have much effect on the control performance in stabilizing output states except yaw. Generally, the controller responded to step input reference quite well by tracking references quite smoothly. However, in the case of yaw state, oscillation occurred around $\pm 5.0^{\circ}$ when STASMO was used.

The effect of noise was still noticeable clearly around the output states and control inputs as seen in Figures 9.86 - 9.89. However, if compare with previous controllers, the

effect of noise on the control input was low by showing small amplitude of oscillation in the signals. The effect of noise still occurred due to trade-off of choosing parameters for the observers. On the one hand, increasing or decreasing observer gains will speed up tracking true value, yet may potentially increase sensitivity to measurement noise and vice versa. In this research, the speed of tracking measured states was more considered (without neglecting noise effect) to make the controller more responsive.

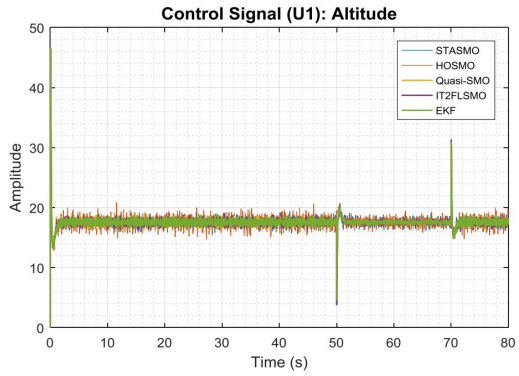

Figure 9.86. Control signal u_1

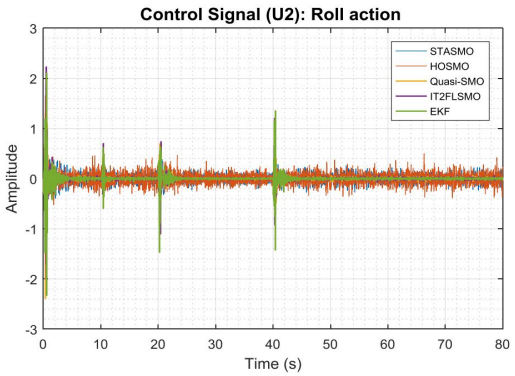

Figure 9.87. Control signal u_2

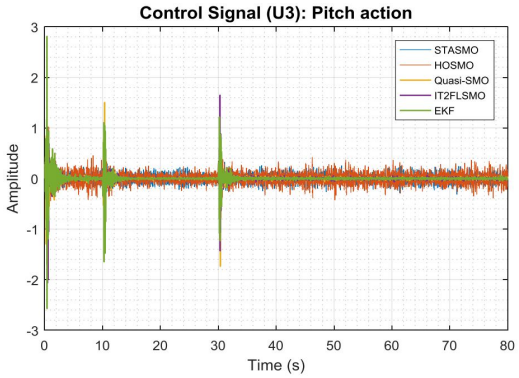

Figure 9.88. Control signal u_3

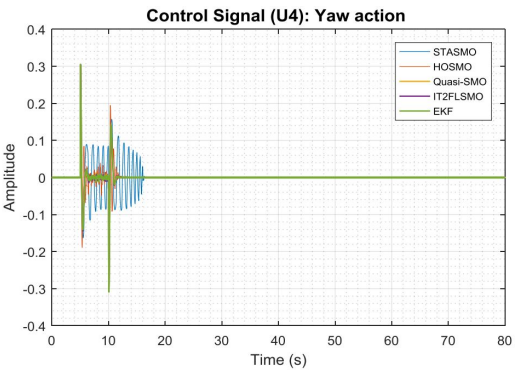

Figure 9.89. Control signal u_4

Steady-states error of output state was also considered to evaluate the robustness of selected controller and observers in dealing with noise disturbance and parameters mismatch. The MSE of the steady-state errors is presented in Table 9.12. Generally, in the presence of such issues steady-states errors were considered small. It means that despite noise and parameters mismatch, the observers influenced control system to stabilize the quadcopter UAV and reduce steady-state error quite well. In this case, STASMO observer method performed better than the other four estimators by showing smaller steady-state error in almost all states.

Table 9.12. Mean squared errors of steady-states errors

Observer methods	MSE steady-state error of					
	x	pitch	y	roll	yaw	z
QuasiSMO	0.000154099	0.00014667	0.000153519	0.000140496	1.00158E-08	0.000154979
IT2FSMO	0.000164337	0.00014533	0.000153464	0.000141111	8.50916E-09	0.00015562
STASMO	7.92722E-05	8.08244E-05	7.20529E-05	7.86021E-05	5.89205E-09	3.46551E-05
HOSMO	0.000412268	0.000275643	0.000420688	0.000302138	2.84122E-09	9.26384E-05
EKF	0.000235152	0.000143496	0.000227582	0.000140625	1.92223E-09	0.000205899

In summary, although the worst case is introduced to the system, DSMC still maintained good performance in controlling the dynamics of the quadcopter UAV with no response overshoot except for yaw action when STASMO was applied, considered small rise time, free from chattering when utilizing QuasiSMO, IT2FSMO and EKF observer methods, small steady-states errors, and faster computational time. However, random fluctuated signals on the estimated true states appeared due to noise disturbance. The selection of observer gain was a trade-off. Reducing or increasing the gain may make observers become insensitive to noise, however, on the other hand it may delay tracking true states. With reference to observers, regarding true state tracking and steady-states errors of output states, STASMO outperformed other methods by showing small MSE values. However, in terms of reducing effect of noise in the estimated states and tracking reference, QuasiSMO, IT2FSMO, EKF performed better than STASMO and HOSMO. It can be seen that smoother estimated states were generated, and no oscillation was noted in tracking reference of yaw state.

9.3 Performance summary

After conducting performance evaluation of the control system and the observer methods with 4 predefined scenarios, it can be summarized that set-point integral dynamic sliding mode control (DSMC) can still maintain the stability of its performance to control the dynamics of quadcopter UAV well in all conditions. Integral term plays an important role in reducing steady-state error of the output states while set-point weighting function plays a role in reducing overshoot due to the integral term. Oscillations are only seen on yaw motion in the condition of parameter mismatch when STASMO observer method was

employed. Furthermore, in terms of observer, actually QuasiSMO and IT2FSMO have similar performance in estimating unmeasured states. However, IT2FSMO has slower computational time than QuasiSMO due to type reduction process in the type-2 fuzzy logic system. Hence, generally QuasiSMO outperformed than others and has contributed to improved control system performance in dealing with all predefined conditions

9.4 Overall summary of numerical simulations

Chapers 6, 7, 8, and 9, presented simulations results of four sliding mode control system approaches and five observer techniques to demonstrate their robustness in dealing with nonlinearity and disturbances due to noise and parameters mismatch. White Gaussian noise with low signal to noise ratio (SNR) and high parameters mismatch has been tested for each combination of control system and observer method. Unmeasured states estimation results, noise rejection, and true states tracking have been taken into account to verify the performance of observer approaches. In addition, computational time contributing to improved performance of controllers has been evaluated to provide additional consideration of performance of observers. The robustness of controller is a very crucial factor in developing and building quadcopter UAVs. Time responses to step input signals including rise time, overshoot, and steady-states errors have been measured to validate the robustness of control systems in several unforeseen conditions. In addition, computational time contributing to improved observer performance has been assessed as additional factor in obtaining appropriate data for comparative assessments.

In terms of performance of observers, it is noted that QuasiSMO, IT2FSMO, and EKF generate smoother estimated states for every condition than STASMO and HOSMO. Generally, in spite of noise free condition, chattering issue still occurs significantly especially in estimating second order states of roll and pitch when STASMO and HOSMO are applied. MSE of estimated states in steady-state conditions has also been computed and evaluated to verify observer performance. In noise-free condition, QuasiSMO has outperformed others by showing smaller MSE in all states. However, in the presence of noise

and parameters mismatch, QuasiSMO had smaller MSE for second order states of x, y, and yaw than other observers; while EKF was better than the other four estimators in estimating roll and pitch rates.

The MSE of true states tracking and noise rejection are taken into account to evaluate the performance of observers. In the ideal condition, including the case of noise-free and no parameters mismatch, QuasiSMO has shown smaller MSE for all states, while STASMO and HOSMO had good performance in the presence of noise and uncertainty by showing small MSE for most states. However, in terms of noise rejection, QuasiSMO, IT2FSMO and EKF have more capability in rejecting noise effect in the estimation process than STASMO and HOSMO by showing small random fluctuation in all estimated states. Actually, reducing or rejecting noise effect and fast tracking are trade-off for observer methods. On the one hand, increasing or decreasing observer gains will speed up tracking true value, yet may potentially increase sensitivity to noise measurement and vice versa. In this research, the speed of tracking measured states is more considered (without neglecting noise effect) to make the controller more responsive.

Each observer has shown different achievement in contributing to improving control system performance. With the use of QuasiSMO, IT2FSMO and EKF, the control systems has performed well by exhibiting no overshoot or oscillation for all output states, whereas STASMO and HOSMO in general made controllers to respond to step input references with overshoot and oscillation. Furthermore, in terms of steady-state errors of output states, MSE has been calculated and compared. For noise free condition, QuasiSMO has performed better than others when the control methods of QuasiSMC, IT2FSMC, and STASMC were utilized. Meanwhile, when DSMC was applied, no observer showed dominating better performance. However, in the presence of noise, STASMO and HOSMO were dominating with better performance by resulting smaller steady-state error of output states especially when DSMC was employed.

Computational time of observer algorithm as measured and presented in Table 6.5 provides an overview of the feasibility of the observer to be applied in real-time. From the table, it is noticeable clearly that QuasiSMO algorithm has smaller time consumption than

others of around 0.04125 ms to execute estimation process, whereas IT2FSMO has been the slowest around 4.02375 ms. Type reduction method in type-2 Fuzzy logic system is one of the issues which leads to this delay.

Control system design plays critical role in developing and building UAV systems including quadcopter-based UAVs. As described in the literature review in Chapter 2, sliding mode control system is believed to have good abilities in dealing with nonlinearity and uncertainty. However, the issue of chattering due to switching term in the control algorithm has become a constraint in implementing this approach. The current research has been conducted to reduce and eliminate this phenomenon without degrading the control performance. Numerical validations of several SMC algorithms including QuasiSMC, IT2FSMC, STASMC, and DSMC have been carried out to verify the performance of such methods.

Chattering which is the main issue with the use of SMCs, is eliminated perfectly in QuasiSMC and IT2FSMC. Replacing discontinuous term with saturation/sigmoid function and rule-based algorithm have been shown as a way to eliminate the phenomenon. However, chattering are still seen clearly in the estimation of roll and pitch rate when STASMO and HOSMO are applied. However, although cannot eliminate the chattering issue as well as QuasiSMC and IT2FSMC, STASMC has performed adequately to reduce the oscillations sufficiently in amplitude. However, the performance of STASMC relies on what observer is employed to estimate second order states. Significant oscillation has been seen when using STASMO and HOSMO, whereas such issue has not been seen when QuasiSMO and IT2FSMO were employed. Finally, the chattering phenomenon has been reduced effectively by employing DSMC as control system. Oscillation used to be generated by STASMO and HOSMO were no longer seen when DSMC was employed. In other words, what ever estimators used, all estimated states have been generated quite smoothly.

Time responses of several SMCs including rise time, overshoot, and steady-state error have been evaluated for purposes of performance comparison. Controller parameters were tuned manually to get optimal performance of control systems. Two issues were considered in evaluating rise times. Firstly, they were measured in the noise-free condition and no

parameters mismatch. Secondly, only four output states were evaluated including x, y, z, and yaw. Meanwhile, overshoot and steady-state errors were assessed for all possible conditions of the system, including no noise and no parameters mismatch, no noise and with parameters mismatch, with noise and no parameters mismatch, and with noise and parameters mismatch; and for all output states, namely x, y, z, roll, pitch, and yaw.

In most output states and various uses of observer, STASMC method responded to step input references with shorter rise time than others followed by QuasiSMC, IT2FSMC, and DSMC. In other words, STASMC outperformed other methods in terms of rise time. However, this control system had larger overshoot and oscillation occurred in all possible conditions of the UAV system when STASMO was employed. In contrast, DSMC approach responded to step input references with no overshoot for various uses of observers, except in the condition of noise disturbance and the use of STASMO and HOSMO approaches. Meanwhile, QuasiSMC and IT2FSMC responded to input references with overshoot for all output states in the condition of noise disturbance when STASMO and HOSMO were applied. However, all control system approaches performed with no response overshoot when QuasiSMO, IT2FSMO and EKF were employed.

Generally, by employing set-point weighting function (ζ), the overshoot due to the use of integral function in sliding mode-based nonlinear control systems, has been reduced significantly especially in the noise-free condition. Although there were some overshoot occurrences on the application of one of the method of control system under very bad noise conditions and very high parameters mismatch, the overshoot was not of very significant amplitude. The overshoot was more due to the very high noise disturbance and parameters mismatch on the system.

The performance of control systems have been evaluated, compared and analysed also by calculating MSE of steady-state error (Ess) for each control system. This was carried out to show which approach perform better in reducing such errors. QuasiSMC has shown its advantages by showing smaller MSE in output states of x, y, and roll; while IT2FSMC, DSMC, and STASMC were better in pitch, yaw, and z respectively. Therefore, based on MSE data presented in Tables 6.6, 6.9, 6.12, 6.15, 7.3, 7.6, 7.9, 7.12, 8.3, 8.6, 8.9, 8.12,

9.3, 9.6, 9.9, and 9.12 in general, QuasiSMC has outperformed other methods followed by STASMC, IT2FSMC, and DSMC.

Time required to process control algorithm plays an important role in designing responsive control system. Therefore, this parameter has been taken into account to obtain the general description whether the method is feasible to be used in real-time application or not. Table 9.13 presents total control loop time for each control system method. From the table it can be seen that IT2FSMC had the longest control loop time by over 3 ms, due to type reduction method in type-2 Fuzzy logic system. Meanwhile, DSMC is the fastest method by lower than 0.3 ms for each cycle of process.

Table 9.13. Total control loop time

Control systems	**Total computational time** (ms)				
	QuasiSMO	IT2FSMO	STASMO	HOSMO	EKF
QuasiSMC	0.6358	4.6767	0.6342	0.6415	0.7076
IT2FSMC	3.6573	7.8940	3.6774	3.6849	3.7387
STASMC	0.6350	4.6662	0.6448	0.6496	0.7034
DSMC	0.1719	4.1576	0.1733	0.1807	0.2202

Lastly, for further robustness testing in dealing with uncertainty, a percentage of increasing parameters mismatch has been conducted subject to noise disturbance. In this assessment, the robustness of the methods are shown by the stability of the system in responding to step input references. The maximum percentage values of parameters uncertainty obtained are presented in the Table 9.14. From the table it can be seen clearly that the use of QuasiSMC and STASMC demonstrate better ability in dealing with uncertainty than other control methods by showing higher percentage of parameters mismatch. Furthermore, with reference to observers, QuasiSMO and EKF have outperformed other estimator approaches in dealing with such uncertainty. Therefore, the combination of such control systems and observer methods can be a good choice for further validation.

Table 9.14. Maximum parameters mismatch subject to noise disturbance

Control systems	**Maximum parameters mismatch of each observer (%)**				
	QuasiSMO	IT2FSMO	STASMO	HOSMO	EKF
QuasiSMC	90.0	90.0	58.0	70.0	90.0
IT2FSMC	89.0	89.0	58.0	71.0	89.0
STASMC	90.7	90.6	60.0	75.0	90.7
DSMC	89.0	89.0	52.0	64.0	90.2

After performing a series of performance assessments for control system and observer

methods and considering the results obtained, it can be summarized that generally, set-point integral quasi-sliding mode control (QuasiSMC) maintains better performance than others in all conditions and various uses of observers. Meanwhile, in addition to having faster processing time than other methods, quasi-sliding mode observer has a promising performance in estimating unmeasured states by maintaining good results in every determined performance categories. Therefore, the combination of such control and observer methods is considered to be evaluated experimentally. However, set-point integral super-twisting algorithm of sliding mode control (STASMC) which is more robust in terms of parameter mismatch than other control methods, is considered to be evaluated in experimental validation for comparison performance in real-time.

Chapter 10

Experimental validations

10.1 Introduction

In Chapter 6 - 9, numerical simulations were presented to validate the performance of several sliding mode-based nonlinear control systems with nonlinear full-order state observer methods in the quadcopter UAV application. The simulations have shown that generally set-point integral quasi-sliding mode control (QuasiSMC) yields better performance than other sliding mode-based nonlinear control systems in all conditions and various uses of observers. However, set-point integral super-twisting algorithm of sliding mode control (STASMC) shows superior performance in dealing with high percentage of parameter mismatch with any combination of observer methods. In terms of observer method, in general, quasi-sliding mode observer (QuasiSMO) demonstrates the ability to estimate unmeasured states and improve the performance of the controllers well. Apart from that, QuasiSMO has faster processing time compared with others so it is more suitable to be used in real-time applications. Therefore, QuasiSMC, STASMC, and QuasiSMO are considered in the experimental validations.

The overall control and observer system is as shown in Figure 10.1. In this work, sliding mode control (SMC) is employed to control the altitude and attitude of the quadcopter. The aim of such controller is to enforce the system dynamics to track desired trajectories, whereas SMO is utilized to estimate unmeasured states of the vehicle which is required for control process.

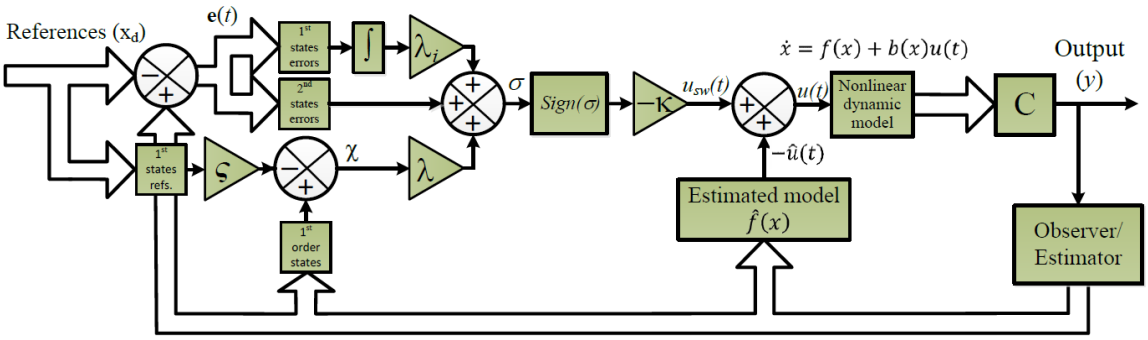

Figure 10.1. Block diagram of the overall control and observer system

10.2 Experimental setup

In this research, myRIO-1900 platform with LABVIEW based-framework, is used to apply the selected controller and observer in real-time application. The quadrotor was set up by integrating different mechanical and electrical components. The quadcopter system which is illustrated in Figure 10.2, consists of the structural frame (Turnigy H.A.L frame), 10x45" propulsion system, controller board (myRIO-1900 equipped with wifi), external sensors (IMU for AHRS), and power system. The same PC (equipped with wifi dongle) as simulation setup was used as ground navigation system.

In this experiment, 9 Degrees of Freedom - GY-86 IMU is utilized to acquire linear accelerations, angular velocity and earth magnetic field. The sensor data is acquired and processed through Arduino Pro Mini 328 - 5V/16MHz. The firmware to generate attitude and heading reference system (AHRS) is obtained in github (Kheowree, 2016). Static accuracy of such attitude data are presented as: roll/pitch $< 1°$; and yaw $< 2°$.

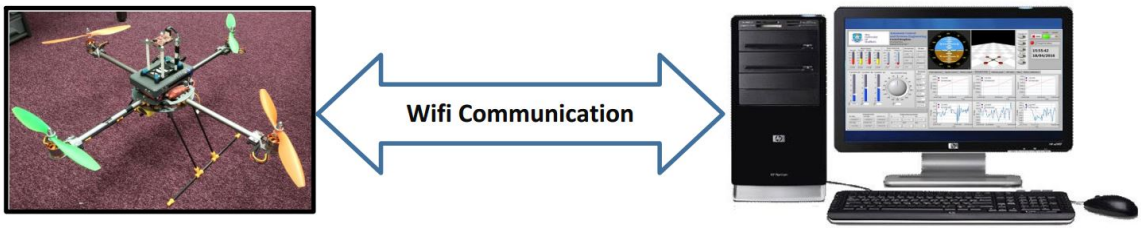

Figure 10.2. Experimental setup of quadcopter system

The mass moment of inertia (Turnigy H.A.L frame), mass, and length of quadcopter arms are real parameters obtained by measurements. The parameters of the plant are as presented in Table 10.1, meanwhile the selected parameter values for controllers are shown in Tables 10.2 and 10.3. The saturation constant (ϵ) for QuasiSMC is 0.1. To estimate

unmeasured states, the selected observer parameters are presented as α_1 = 120, α_1 = 300, K_1 = 0.1, and K_2 = 5. The selection of the controllers and observer parameters is carried out through trial and error (manual tune) until obtaining optimal performance from the two controllers and observer. Furthermore, the set-point weighting values (ζ) for roll (ϕ) are 1 and 0.83, and for pitch (θ) are 1 and 0.82. The selection of the two values of set-point weighting function (ζ) for each state is to demonstrate the effect of the weighting on the performance of the controller. The bandwidth of quadcopter closed-loop system as obtained experimentally was around 12.638 Hz. According to Nyquist sampling criterion, the sampling frequency (f_s) for control and estimation process must be greater than twice of the system's bandwidth. Therefore, the sampling frequancy used in this experiement was around 1 KHz.

Table 10.1. Quadcopter parameters

	Values	
Variabels	**Observer & control models**	**Units**
m	1.79	kg
l	0.29	m
I_x	0.026281674	kgm^2
I_y	0.027484487	kgm^2
I_z	0.045603074	kgm^2
g	9.81	m/s^2

Table 10.2. Controller parameters for QuasiSMC method

QuasiSMC parameters							
Variables	**Values**	**Variables**	**Values**	**Variables**	**Values**	**Variables**	**Values**
λ_ϕ	1.485	λ_θ	1.251	λ_ψ	1.109	λ_z	1
$\lambda_{i\phi}$	5.76	$\lambda_{i\theta}$	5.76	$\lambda_{i\psi}$	0.5	λ_{iz}	0.1
k_ϕ	80.9	k_θ	97.86	k_ψ	60	k_z	50
μ_ϕ	17.5	μ_θ	17.5	μ_ψ	17.5	μ_z	17.5

Table 10.3. Controller parameters for STASMC method

STASMC parameters							
Variables	**Values**	**Variables**	**Values**	**Variables**	**Values**	**Variables**	**Values**
λ_ϕ	1.485	λ_θ	1.251	λ_ψ	1.109	λ_z	1
$\lambda_{i\phi}$	5.76	$\lambda_{i\theta}$	5.76	$\lambda_{i\psi}$	0.5	λ_{iz}	0.1
κ_ϕ	80.9	κ_θ	97.86	κ_ψ	60	κ_z	50
μ_ϕ	17.5	μ_θ	17.5	μ_ψ	17.5	μ_z	17.5

According to the system identification that has been done in Chapter 3 to obtain the information about the capabilities of the motor and propeller used for the designed quadcopter system, the maximum thrust force generated by four motors is around 46.33 N as shown in Table 3.6. This information should be considered when designing control system in real-time applications. Therefore, in the experimental validation, the maximum control

input to be processed in the control system will be limited to the value of $\pm$ 46.33 N or equal to 100% of PWM signal to avoid unexpected behaviour of the quadcopter system.

There are two conditions to validate the performance of the control systems and observer method experimentally,

i. without payload disturbance

ii. with payload disturbance

The purpose of this approach is to know the ability of selected controllers and observer in dealing with such conditions.

10.3 Performance criteria

This chapter presents performance verification of selected control and observer methods in real time applications. Comparative assessment of the two controllers is carried out subject to various set-point weighting values and payload disturbance. Maximum overshoot, rise time, steady-state error, true state estimation error, and the error of second state order estimation in steady-state condition are taken into account to evaluate the performance of controllers and observer. In addition, recovery time to track reference after introducing payload disturbance is assessed to demonstrate the robustness of the control systems and observer method.

In the numerical simulation, the number of states to be controlled was 12 states. In this experiment, however, the number of states to be controlled is reduced to 8 due to the availability of adequate sensors, by excluding: x-movement, rate of x-movement($\dot{x}$), y-movement, and rate of y-movement($\dot{y}$). In addition, due to time constraints, the amount of data taken is only for 4 states, namely roll(ϕ), rate of roll($\dot{\phi}$), pitch(θ) and rate of pitch($\dot{\theta}$). The consideration of selecting such data is because the data is very crucial in the application of quadcopter UAVs. Based on these considerations, the complete experimental validations for 12 states will be performed in future works.

10.4 Experimental results

10.4.1 Roll movements

In the first experiment, the evaluation of selected controllers, including QuasiSMC and STASMC; and selected observer, namely QuasiSMO is carried out with free of payload disturbance to control roll movements of quadcopter UAV. Performance assessment through comparative studies between two controllers is investigated and analysed subject to set-point weighting values. Furthermore, to produce a more valid comparison analysis, data retrieval is done at a voltage level of 11.6 volts.

In the condition of payload free disturbance, the experimental results obtained are shown in Figures 10.3 - 10.14

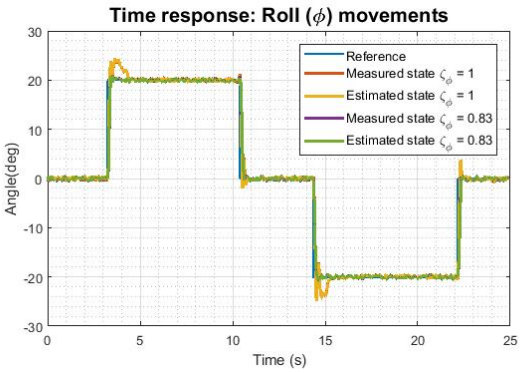

Figure 10.3. Roll movements - QuasiSMC

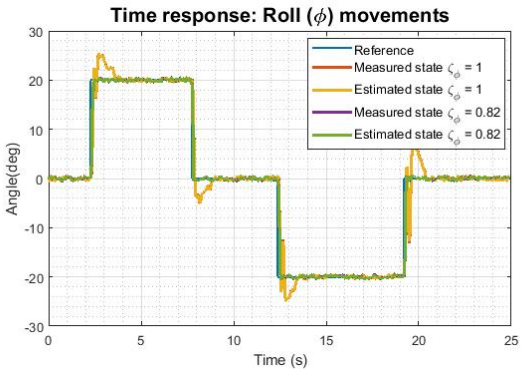

Figure 10.4. Roll movements - STASMC

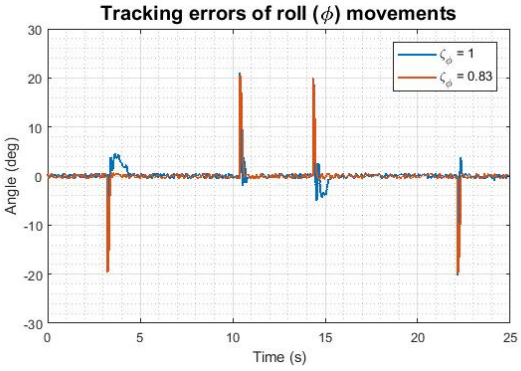

Figure 10.5. Tracking errors - QuasiSMC

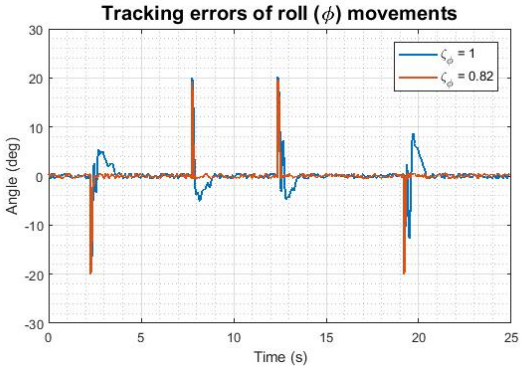

Figure 10.6. Tracking errors - STASMC

As noted, although the reference changes suddenly, in general, the two controllers, namely QuasiSMC and STASMC showed good response to keep the system trajectory to track the reference with fast rise time (Tr) and small tracking errors as seen in Figures 10.3 - 10.6. Moreover, the chattering phenomenon following the issue of sliding mode control

was reduced effectively by employing the two control methods so the overall performance of the system was not affected. The performance of the two controllers also improved after introducing set-point weighting values to the control system. It can be seen in Figures 10.3 - 10.6 and Table 10.4 that the overshoot phenomenon due to the utilization of integral action in the control systems was reduced significantly by reducing the weighting value from 22.65% to 4.115% for QuasiSMC, and from 27.42% to 2.5% for STASMC without affecting rise time (Tr) significantly, and the steady-state error (Ess) remained at similar values for both control systems. Fast rise time is demonstrated by the two controllers as shown in the Table 10.4, around 0.108 seconds ($\zeta = 1$) and 0.152 seconds ($\zeta = 0.83$) for QuasiSMC; and 0.144 seconds ($\zeta = 1$) and 0.132 seconds ($\zeta = 0.83$) for STASMC

Table 10.4. Control systems performances

Control methods	**Overshoot (%)**		**Tr(s)**		**Ess(deg)**	
	ζ=1	**ζ=0.83**	**ζ=1**	**ζ=0.83**	**ζ=1**	**ζ=0.83**
QuasiSMC	22.65	4.115	0.108	0.152	0.225960693	0.229352473
STASMC	27.42	2.5	0.144	0.132	0.247334433	0.204601213

Noting the control signals in Figures 10.7 - 10.10, although numerical simulations results showed that smooth control signals were generated by QuasiSMC, in the practical application, high frequency oscillations were seen in the signals for both control systems. However, the oscillations were small. This phenomenon occurs due to maintaining the system trajectory on the reference for all times.

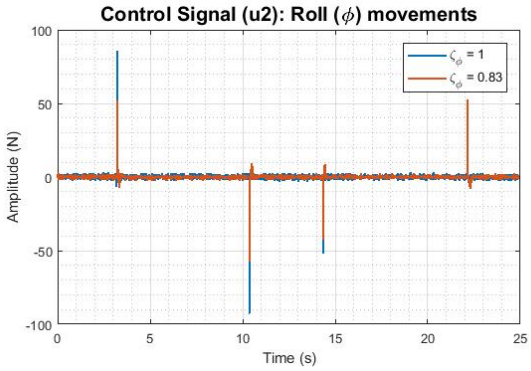

Figure 10.7. Control signals - QuasiSMC

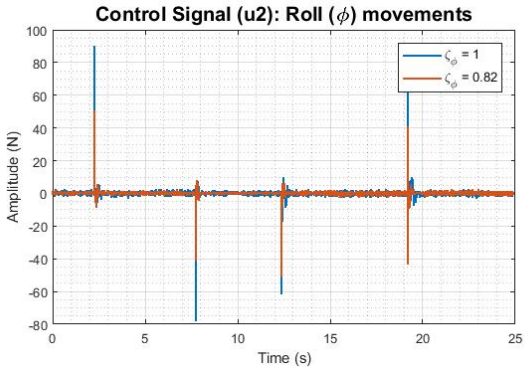

Figure 10.8. Control signals - STASMC

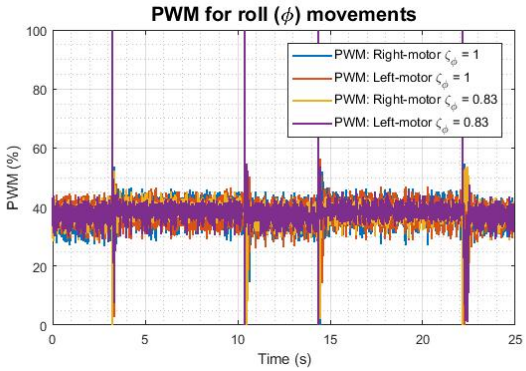

Figure 10.9. PWM signals - QuasiSMC

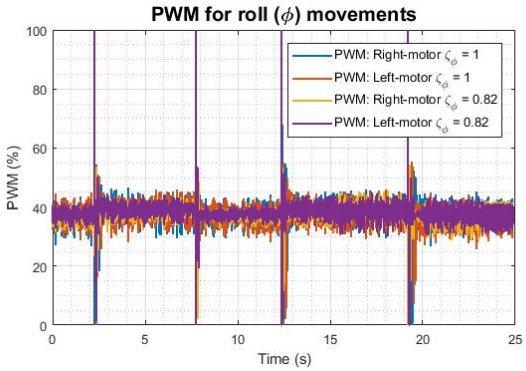

Figure 10.10. PWM signals - STASMC

In most cases, all selected control systems succeeded in dealing with nonlinearity of the system and all unforeseen circumstances. However, observing Table 10.4, the two controllers showed slightly different performances. Although STASMC exhibited higher overshoot when the set-point weighting value was 1, STASMC showed smaller overshoot and steady-state errors compared with QuasiSMC when the weighting value was 0.83.

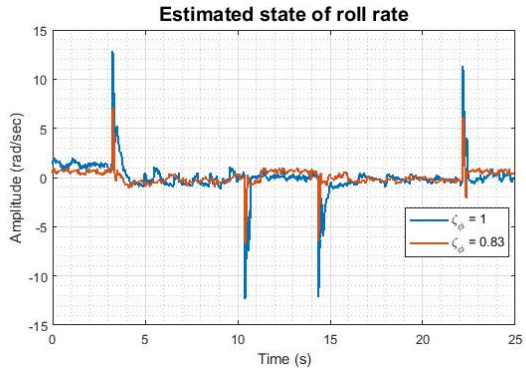

Figure 10.11. Roll rate - QuasiSMC

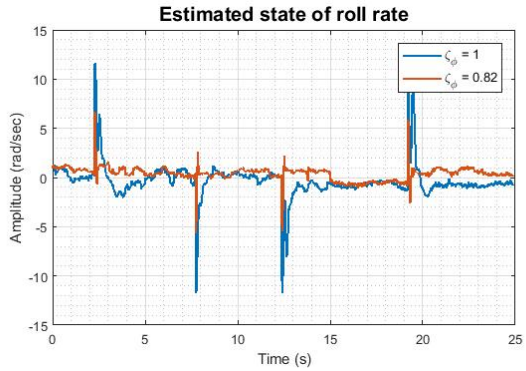

Figure 10.12. Roll rate - STASMC

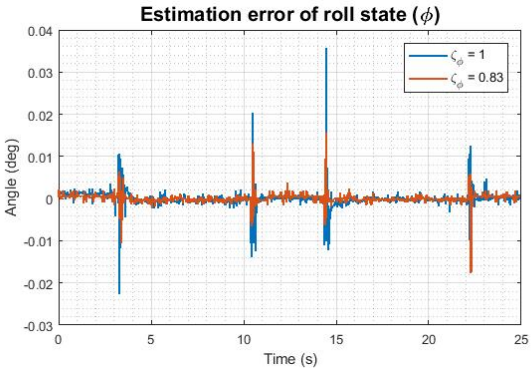

Figure 10.13. Estimation errors - QuasiSMC

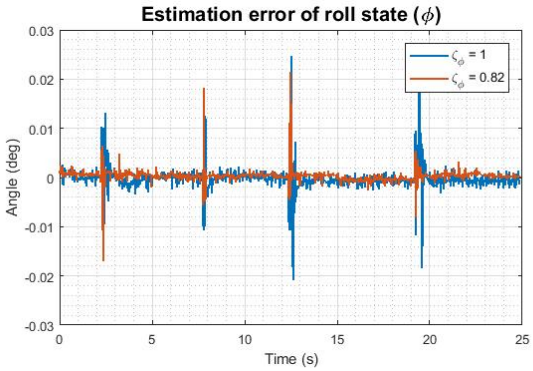

Figure 10.14. Estimation errors - STASMC

In the numerical simulations, QuasiSMO demonstrated good performance in estimating unmeasured states in all predetermined conditions and had a faster processing time compared to other methods. In addition, this observer had better performance in improv-

ing control system performance. In this experimental validations, the observer performed quite well as seen in Figures 10.11 - 10.12. The observer provided proper unmeasured states to be utilized in the control systems. It can be seen in Table 10.5 that mean squared error (MSE) of estimated roll rate (ERR) in steady-state condition was quite small. Furthermore, the performance quality of the observer was evaluated by measuring estimation error (EE) of tracking true state (estimated true state as seen in Figures 10.3, 10.4). As shown in the table, the estimation errors were very small for all the controller methods. It means that the values of estimated states were very close to the actual ones.

Table 10.5. Observer performances

Control methods	**MSE EE**		**MSE ERR**	
	ζ=1	**ζ=0.83**	**ζ=1**	**ζ=0.83**
QuasiSMC	2.81804E-06	9.61817E-07	0.300457406	0.180575645
STASMC	3.51E-06	8.47465E-07	0.399361504	0.54728051

Set-point weighting function also contributed to improving the performance of the observer as seen in the table. Overall, by introducing smaller weighting values to control the system, the performance of the observer improved significantly by showing smaller MSE of estimation error (EE) and MSE of estimated roll rate (ERR).

10.4.2 Roll movements with payload disturbance

The robustness of the proposed control systems and observer method were assessed by providing payload disturbance of 0.082 kg to one of the vehicle arms. The same control parameters were applied to the controllers with a set-point weighted (ζ) value of 0.83. The responses of the methods in dealing with this type of disturbance are presented in Figures 10.15 - 10.20

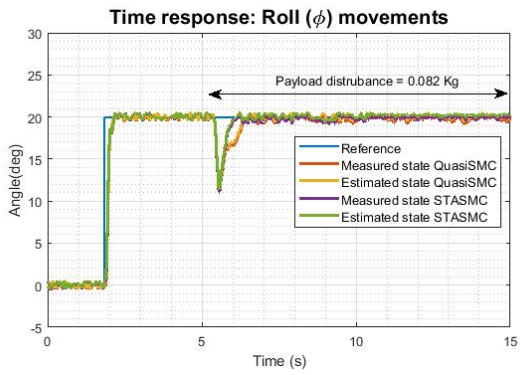

Figure 10.15. Roll movement - payload

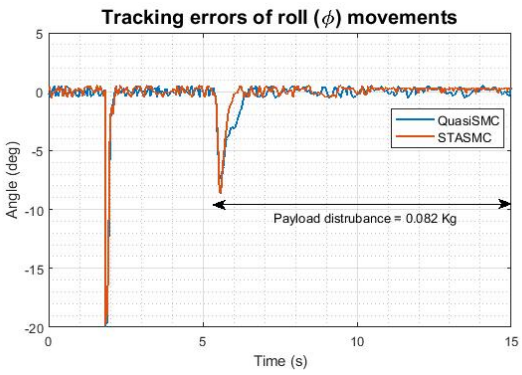

Figure 10.16. Tracking errors - payload

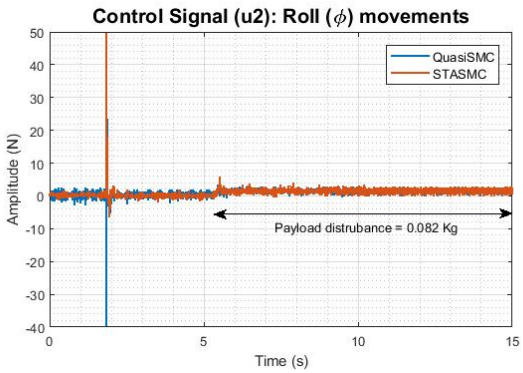

Figure 10.17. Control signals - payload

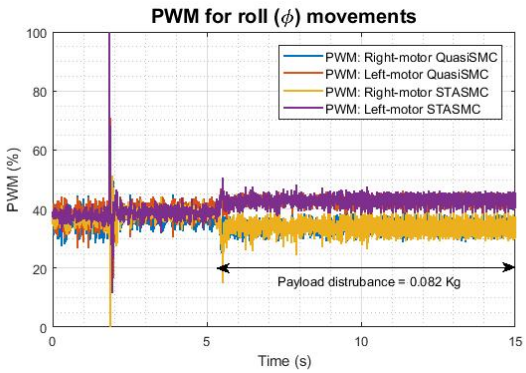

Figure 10.18. PWM signals - payload

From the figures, it can be seen that the controller handled the payload interference. Although it could not withstand a few moments of the load on the arm, the controllers restored the system trajectory to follow back the reference fairly quickly even though there was still a load on the arm as seen in Figures 10.15 and 10.16. The control signals changed slightly larger to maintain the system in tracking the reference as depicted in Figures 10.17 and 10.18.

As noted, STASMC showed better responses than QuasiSMC. Table 10.6 shows that in spite of having larger angle change, STASMC responded to the disturbance faster than QuasiSMC by showing smaller recovery time of about 0.572 seconds to track back the reference. Nevertheless, both controllers kept maintaining small steady-state errors (Ess) and similar values before and after disturbance was given.

Table 10.6. Control systems and observer performances

Performances	**QuasiSMC**	**STASMC**
Angle change (deg)	7.377	8.652
Recovery time(s)	0.912	0.572
Ess (before)	0.176532951	0.234082209
Ess(after)	0.228564955	0.233742812
MSE EE(before)	2.78672E-06	1.23423E-06
MSE EE (After)	2.82324E-05	2.80775E-05
MSE ERR (before)	2.50878194	0.226738191
MSE ERR (After)	41.50188045	41.36048848

The observer still performed well despite a payload disturbance on the rotorcraft. The unknown state of quadcopter UAV was well estimated as depicted in Figure 10.19. Change of roll rate during interruption occurred because the control system was trying to keep the system to track the reference all the time. The most interesting aspect in this experiment was the observer's ability to track the true state very well during the time the disturbance occurred. The performance of the observer to track actual state was measured by MSE of

estimation error (EE). As presented in Table 10.6 and Figure 10.20, the estimation error (EE) for both controllers did not change significantly. It means that the estimated state was still keeping close to the actual one under payload disturbance.

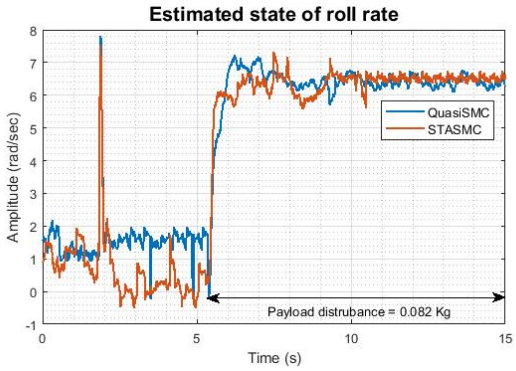

Figure 10.19. Roll rate - payload

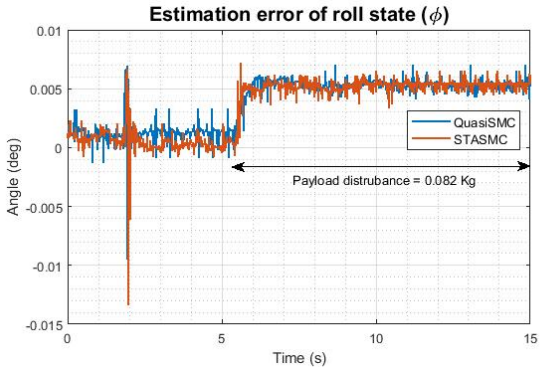

Figure 10.20. Estimation errors - payload

10.4.3 Pitch movements

The performance of selected controllers, including QuasiSMC and STASMC; and selected observer, namely QuasiSMO were also evaluated with free of payload disturbance to control pitch movements of the quadcopter UAV. Performance assessment through comparative studies between two controllers was investigated and analysed also subject to set-point weighting values. The same as previous works, to produce a more valid comparison analysis, data retrieval was done at a voltage level of 11.6 volts.

In the condition of payload free disturbance, the experimental results obtained for pitch movements are presented in Figures 10.21 - 10.32.

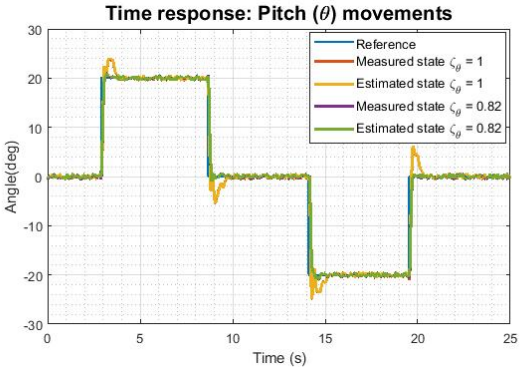

Figure 10.21. Pitch movements - QuasiSMC

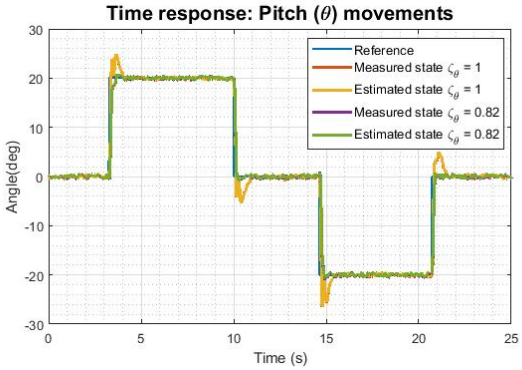

Figure 10.22. Pitch movements - STASMC

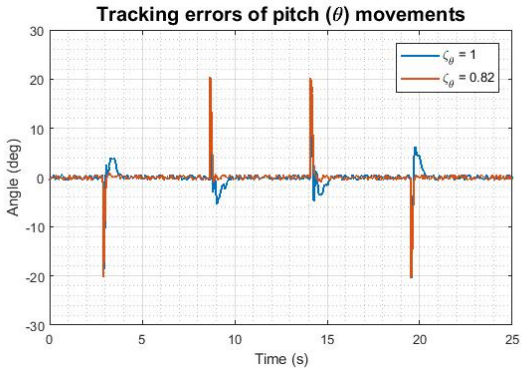

Figure 10.23. Tracking errors - QuasiSMC

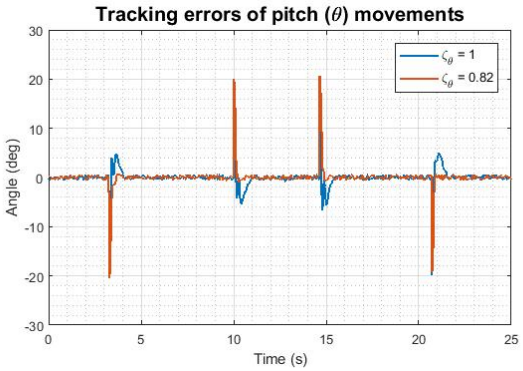

Figure 10.24. Tracking errors - STASMC

Similar with roll movements, in general, both controllers, namely QuasiSMC and STASMC showed good response to keep the system trajectory to track the reference with fast rise time (Tr) and small tracking errors as seen in Figures 10.21 - 10.24 despite of sudden changes of reference. Moreover, the chattering phenomenon following the issue of sliding mode control was reduced effectively by employing the two control methods so the overall performance of the system was not affected. The performance of the controllers was also improved after introducing set-point weighting values to the control systems. It can be seen in Figures 10.21 - 10.24 and Table 10.7 that the overshoot phenomenon due to the utilization of integral action in the control systems was reduced significantly by reducing the weighting value from 20.44% to 4.915% for QuasiSMC, and from 24.53% to 4.155% for STASMC without affecting rise time (Tr) significantly, and the steady-state error (Ess) remained at similar value for both control systems. Fast rise time was achieved by the two controllers as shown in Table 10.7, around 0.148 seconds ($\zeta = 1$) and 0.164 seconds ($\zeta = 0.82$) for QuasiSMC; and 0.128 seconds ($\zeta = 1$) and 0.200 seconds ($\zeta = 0.82$) for STASMC.

Table 10.7. Control systems performances

Control methods	**Overshoot (%)**		**Tr(s)**		**Ess(deg)**	
	ζ=1	**ζ=0.82**	**ζ=1**	**ζ=0.82**	**ζ=1**	**ζ=0.82**
QuasiSMC	20.44	4.915	0.148	0.164	0.223549497	0.214425562
STASMC	24.53	4.155	0.128	0.2	0.176444253	0.213413197

Observing the control signals in Figures 10.25 - 10.28, although the numerical simulations results showed that smooth control signals were generated by QuasiSMC, in the practical application, high frequency oscillations were seen in the signals for both control systems. However, this was small. This phenomenon occurs due to maintaining the system

trajectory on the reference for all times.

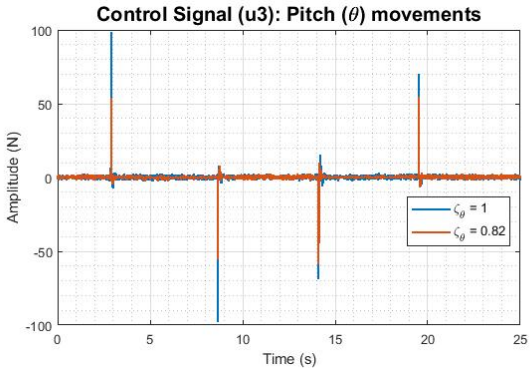

Figure 10.25. Control signals - QuasiSMC

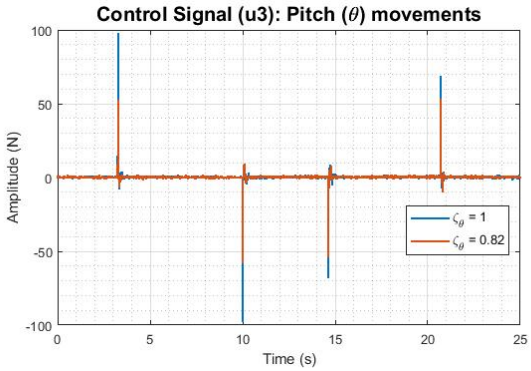

Figure 10.26. Control signals - STASMC

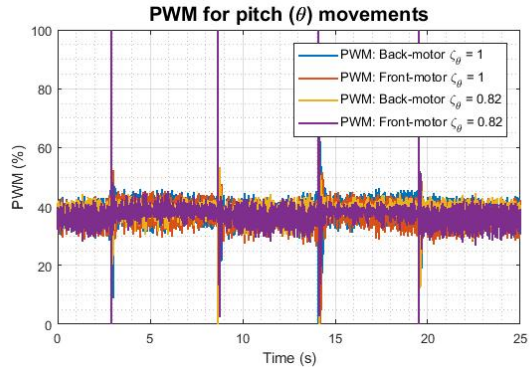

Figure 10.27. PWM signals - QuasiSMC

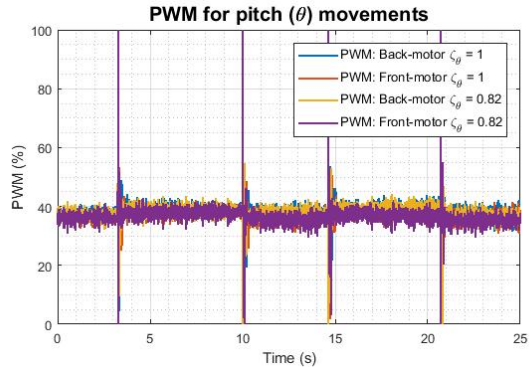

Figure 10.28. PWM signals - STASMC

Generally, both control systems were successful in handling the nonlinearity of the system and all unforeseen circumstances. With reference to Table 10.4, the two controllers showed similar performances in controlling the pitch movements of the vehicle. There were slightly different performance results between QuasiSMC and STASMC, for instant, STASMC had lower overshoot than QuasiSMC when ζ was 0.82, and the controller had lower steady-state error (Ess) for both ζ values, but the differences were not significant. Therefore, in this case, both controllers showed similar performances in controlling the pitch movements.

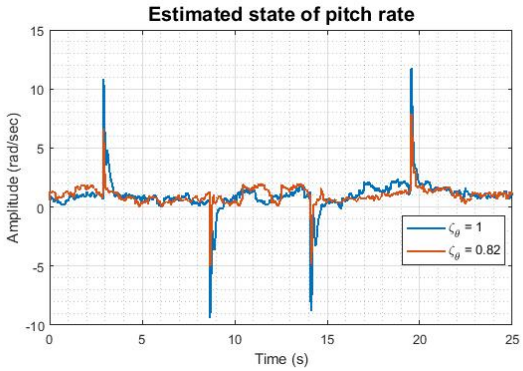

Figure 10.29. Pitch rate - QuasiSMC

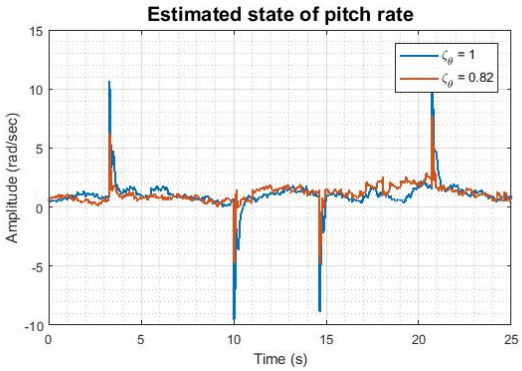

Figure 10.30. Pitch rate - STASMC

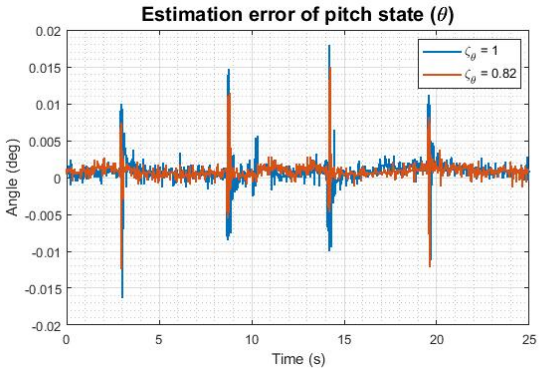

Figure 10.31. Estimation errors - QuasiSMC

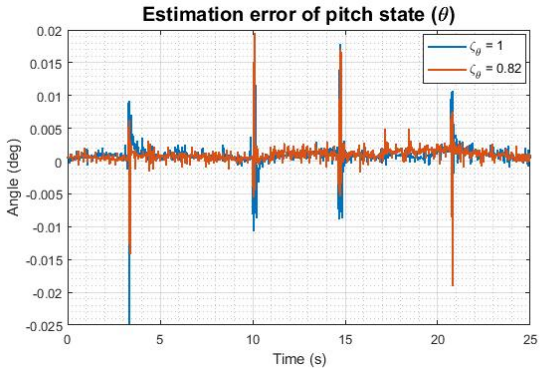

Figure 10.32. Estimation errors - STASMC

In the numerical simulations, QuasiSMO demonstrated good performance in estimating unmeasured states in all predetermined conditions and had a faster processing time compared to other methods. In addition, this observer performed better to improve the performance of control systems. In this experimental validation, the observer has performed quite well as seen in Figures 10.29 - 10.32 to estimate unmeasured states and tracking true states. The observer generated adequate pitch rate state to be utilized in the control systems. It can be seen in Table 10.8 that mean squared error (MSE) of estimated pitch rate (EPR) in steady-state condition was quite small. Furthermore, the performance quality of the observer was evaluated by measuring estimation error (EE) of tracking true states (estimated true state as seen in Figures 10.21, 10.22). As shown in the same table and Figures 10.31 - 10.32, the estimation errors were very small for all the control systems. It means that the values of estimated states were very close to the actual ones.

Table 10.8. Observer performances

Control methods	**MSE EE**		**MSE EPR**	
	ζ**=1**	ζ**=0.82**	ζ**=1**	ζ**=0.82**
QuasiSMC	2.6566E-06	1.56151E-06	0.485602409	0.560899811
STASMC	2.38261E-06	2.08375E-06	1.089531314	0.586225678

Furthermore, set-point weighting function also contributed to improving the performance of the observer as seen in Table 10.8. Overall, by introducing smaller weighting values into the control system, the performance of the observer improved significantly by showing smaller MSE of estimation error (EE) and MSE of estimated pitch rate (EPR).

10.4.4 Pitch movements with payload disturbance

The robustness of the proposed nonlinear control systems and observer method was evaluated by providing payload disturbance of 0.082 kg to one of the vehicle arms. The same control parameters were applied to the controllers with a set-point weighted (ζ) value of 0.82. The responses of the methods in dealing with this type of disturbance are presented in Figures 10.33 - 10.38

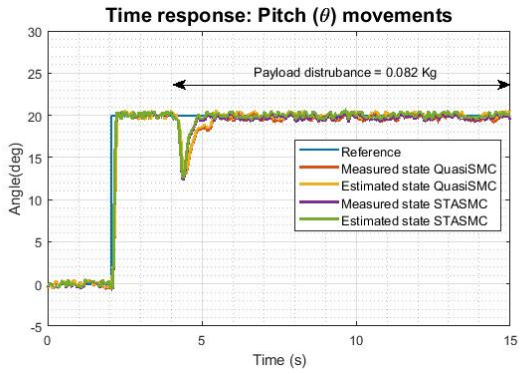

Figure 10.33. Roll movement - payload

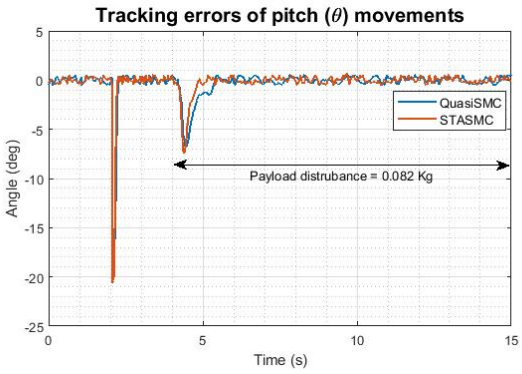

Figure 10.34. Tracking errors - payload

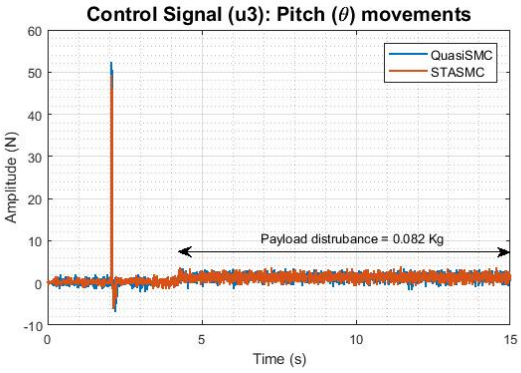

Figure 10.35. Control signals - payload

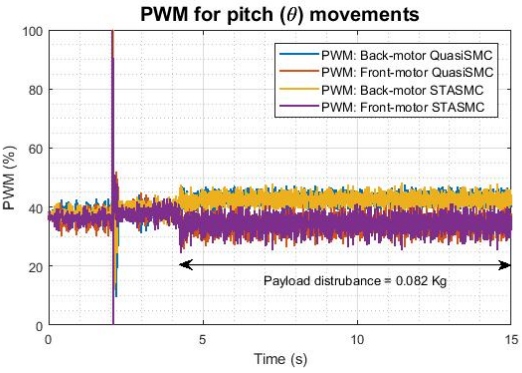

Figure 10.36. PWM signals - payload

From the figures, it can be noticed that the controllers could handle the payload disturbance on one of the quadcopter arms well. Although could not withstand a few moments of the load on the arm, both controllers restored the system trajectory to track back the reference fairly quickly even though there was still a load on the arm as seen in Figures 10.33

and 10.34. The control signals changed slightly larger to maintain the system in tracking the reference as depicted in Figures 10.35 and 10.36.

The STASMC showed better responses than QuasiSMC. Table 10.9 shows that in spite of having larger angle change, STASMC responded to the disturbance faster than QuasiSMC by showing smaller recovery time of about 0.664 seconds to track back the reference, while QuasiSMC had recovery time of about 1.152 seconds. Nevertheless, both controllers kept maintaining small steady-state error (Ess) within similar values before and after disturbance was given.

Table 10.9. Control systems and observer performance

Performances	**QuasiSMC**	**STASMC**
Angle change (deg)	6.854	7.469
Recovery time(s)	1.152	0.664
Ess (before)	0.221238832	0.21779918
Ess(after)	0.240404882	0.219914032
MSE EE(before)	1.36354E-06	1.33526E-06
MSE EE (After)	2.34767E-05	2.05273E-05
MSE EPR (before)	0.219932577	1.092264816
MSE EPR (After)	34.61850702	30.31843526

The observer still performed well despite a payload disturbance on the rotorcraft. The unknown state of the quadcopter UAV was well estimated as depicted in Figure 10.37. Change of pitch rate during interruption occurred because the control system was trying to keep the system to track reference all the time. The most interesting aspect in this experiment was the observer's ability to track the true state very well during the time of the disturbance occurred. The performance of the observer in tracking actual state was measured by MSE of estimation error (EE). As presented in Table 10.9 and Figure 10.38, the estimation error (EE) for both controllers did not change significantly. It means that the estimated state was still kept close to the actual one under payload disturbance.

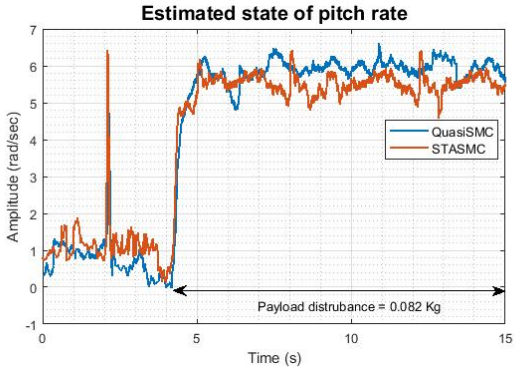

Figure 10.37. Pitch rate - payload

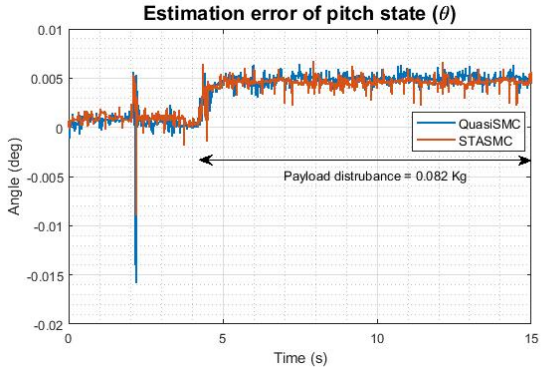

Figure 10.38. Estimation errors - payload

10.5 Summary

Two selected control systems and one observer method, the performances of which were verified in chapter 6 through numerical simulations, have been validated experimentally. Two predetermined conditions have been considered to evaluate the proposed control systems and observer method, including: without payload disturbance and with payload disturbance in controlling the quadcopter UAV.

From the experimental works, it can be summarized that both selected controllers, namely QuasiSMC and STASMC have shown good performance in dealing with the nonlinear and underactuated system as well as unforeseen circumstances, including: possibility of parameters mismatch and payload disturbance. The issue of unavailability of information on system state has been handled very well by QuasiSMO as selected observer in this real-time application. The unforeseen circumstances have not affected much the performance of the observer. Furthermore, the presence of set-point weighting function has played a vital role in improving the performance of the controllers by demonstrating good character in reducing overshoot significantly while maintaining rise time in small values. In addition, generally, by employing this weighting function, the steady-state error (Ess) and estimation errors (EE) were reduced quite well.

In terms of controller performance comparisons, in general, super-twisting algorithm of sliding mode control (STASMC) has shown better ability than quasi-sliding mode control (QuasiSMC) to control some movements of quadcopter UAV especially when a payload disruption to one of the quadcopter arm is applied. In addition, the controller has been able to improve the observer's performance well so that some unmeasured states can be provided adequately and the estimation values of true state is very close to actual values. This fact proves that STASMC has a better ability to handle parameter uncertainties than QuasiSMC.

Chapter 11

Conclusions and recommendations for further work

Methodological approaches and project plan play a critical role in directing the research to achieve the established goals. Some ultimate objectives are fulfilled, but not for some others. Some unfulfilled objectives require more detailed and interdisciplinary research in the future to produce better outcomes. Control engineering, computer science and embedded systems engineering are the main disciplines to be collaborated to resolve various stability, computing and hardware issues in developing this system theoretically and experimentally

11.1 Conclusions

A literature review has a vital role to open up insights and knowledge about what others have done previously. The previous works associated with nonlinear and underactuated systems and its control systems have been investigated and analysed as a foundation in organizing research direction that to be conducted. Since nonlinear and underactuated systems call for complexity of control algorithms to be designed, opportunities to improve the controllers performance on this system will always be widely open. Therefore, the research has been to develop new nonlinear control algorithms with full-order state observers to deal with the issue of nonlinear and underactuated systems. In addition, unforeseen circumstances have become the main focus to be handled by the proposed control systems. As exemplification of the underactuated system for validating the designed control systems is quadcopter UAVs. The selection of this application has been with consideration of model

complexity of this system, so it has been found appropriate to examine the reliability of the proposed controllers.

As has been analysed in the literature review that sliding mode-based nonlinear control system has many advantages over other control systems in dealing with nonlinearity, underactuation, and coupled systems. However, there is a drawback that can degrade the performance of the control system, namely chattering phenomenon. In this work, a new method to tackle the issue has been proposed by employing interval type-2 fuzzy logic system in the control system, namely *sliding mode-based interval type-2 fuzzy control* (IT2FSMC). In addition, *set-point weighting function* and *integral term* have been proposed for use in the control system to improve the overall performance of the system. Furthermore, the *weighting* and *integral terms* have been utilised in several other sliding mode-based control systems, namely: *quasi-sliding mode control* (QuasiSMC), *super-twisting algorithm of sliding mode control* (STASMC), and *dynamic sliding mode control* (DSMC).

In optimal control system-based controller including sliding mode control (SMC), full state feedback plays a crucial role in the control process. Consequently, state observer is required to estimate unmeasured states of the system. Sliding mode observer SMO as a nonlinear observer method outperforms other methods in estimating unmeasured states and has fascinating properties that can overcome the issues of nonlinearity and noise disturbance well. The advantages of such observer have been investigated in the literature review. However, this does not mean there is no shortcoming. Chattering phenomena following the use of discontinuous term in the observer that may lead to performance degradation is the major problem. In this study, *sliding mode-based interval type-2 fuzzy observer* (IT2FSMO) has been proposed to tackle this issue. For performance comparison, several other nonlinear observer approaches namely: *quasi-sliding mode observer* (QuasiSMO), *super-twisting algorithm of sliding mode observer* (STASMO), *high order sliding mode observer* (HOSMO), and *extended Kalman filter* (EKF) have been designed.

The proposed control system and observer method have been evaluated through numerical simulations, and its performances have been compared with other methods, including:

QuasiSMC, STASMC, and DSMC for control systems; QuasiSMO, STASMO, HOSMO, and EKF for observer methods. The methods that have demonstrated the best performance have been selected for further validation through experimentations. From the simulation results obtained, generally all methods have exhibited good performance in controlling the altitude and attitude of the quadcopter UAV in all predetermined conditions. The observers generated the unmeasured states and tracked the true states very well. However, overall QuasiSMC and QuasiSMO have performed better than other methods. Actually, IT2FSMC and IT2FSMO have shown similar performance with QuasiSMC and QuasiSMO, but the type-2 fuzzy logic based-methods has had longer computation time than other methods. Consequently, the methods are not suitable for application to systems that require fast response time such as quadcopter UAVs. The problem of such methods is due to the long iteration process of type reduction algorithm in type-2 fuzzy logic system. However, the proposed method of employing *set-point weighting function* and *integral term* to sliding mode-based control system have performed very well. The methods have improved the performance of control systems significantly in all performance criteria, including reducing overshoot, rise time, and steady-state errors.

Further validation of the selected methods has been carried out in real-time applications. In addition to QuasiSMC, STASMC has also been selected to be validated experimentally for comparison analysis. The reason for choosing STASMC is because it has the ability to deal will parameter mismatch very well. From the experimental results obtained, it can be summarized that *set-point weighting function* plays an important role in reducing overshoot events due to integral term of the controllers while the *integral term* maintains steady-state errors to small value. The robustness of selected controllers has been verified by showing that both control systems could handle payload disturbance applied to the system very well. Furthermore, in term of observer, QuasiSMO has yielded 2^{nd} - order state adequately. The estimation errors of true state tracking have been very small which shows that the estimated states are very close to actual states. The performance of the observer has not been much affected by the disturbance applied to the system and the experimental results have not been much different between before and after the disturbance applied.

In terms of controllers performance comparisons, generally, super-twisting algorithm of sliding mode control (STASMC) has shown better ability than quasi-sliding mode control (QuasiSMC) to control some movements of quadcopter UAV especially when a payload disruption to one of the quadcopter arm is applied. In addition, the controller has been able to improve the observer's performance well so that some unmeasured states can be provided adequately and the estimation values of true state is very close to actual values. This fact proves that STASMC has a better ability in handling parameter uncertainties than QuasiSMC.

11.2 Recommendations for further work

Further research needs to be conducted to fulfill some ultimate objectives and to improve the performance of overall system numerically and experimentally, including but not limited to:

- Conducting more in-depth research to resolve computational time problems in the type reduction process of type-2 fuzzy logic system.
- Designing systematic tuning for control and observer to gain system stability.
- Developing adaptive methods for tuning parameters that can be applied to simulations and real-time applications.
- Validating the performance of the control and observer methods to other nonlinear underactuated systems.
- Developing discrete-time sliding mode control (SMC) and sliding mode observer (SMO) with application to quadcopter UAVs system.
- Validating and comparing the performance of designed discrete-time SMC and SMO with continuous SMC and SMO through numerical simulation.
- Developing an advanced attitude and heading reference system (AHRS) sensor technique by combining IMU sensors with other sensors to produce better AHRS infor-

mation for control process.

- Validating the performance of control system and observer method for complete state of quadcopter UAV experimentally.
- Introducing more disturbance types to the system to validate the robustness of control systems and observer method.
- Conducting interdisciplinary research program involving control engineering, computer science, and embedded systems engineering to develop systems in real applications.

References

Adir, V. G. and Stoica, A. M. (2012). Integral lqr control of a star-shaped octorotor. *INCAS BULLETIN*, 14:3–18.

Alaimo, A., Artale, V., Milazzo, C., Ricciardello, A., and Trefiletti, L. (2013). Mathematical modeling and control of a hexacopter. In *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1043–1050.

Alaimo, A., Artale, V., Milazzo, C. L. R., and Ricciardello, A. (2014). Pid controller applied to hexacopter flight. *Journal of Intelligent & Robotic Systems*, 73(1):261–270.

Alexis, K., Nikolakopoulos, G., and Tzes, A. (2011). Switching model predictive attitude control for a quadrotor helicopter subject to atmospheric disturbances. *Control Engineering Practice*, 19(10):1195–1207.

Alexis, K., Nikolakopoulos, G., and Tzes, A. (2012). Model predictive quadrotor control: attitude, altitude and position experimental studies. *IET Control Theory Applications*, 6(12):1812–1827.

Argentim, L. M., Rezende, W. C., Santos, P. E., and Aguiar, R. A. (2013). Pid, lqr and lqr-pid on a quadcopter platform. In *Informatics, Electronics & Vision (ICIEV), 2013 International Conference on*, pages 1–6. IEEE.

Astolfi, D. and Marconi, L. (2015). A high-gain nonlinear observer with limited gain power. *IEEE Transactions on Automatic Control*, 60(11):3059–3064.

Baik, I.-C., Kim, K.-H., and Youn, M.-J. (2000). Robust nonlinear speed control of pm synchronous motor using boundary layer integral sliding mode control technique. *IEEE Transactions on Control Systems Technology*, 8(1):47–54.

Baklanov, F. Y. and Morozov, V. M. (2013). Stabilization of desired motion of a quadrotor helicopter. *Journal of Computer and Systems Sciences International*, 52(6):955–962.

Belokon', S. A., Zolotukhin, Y. N., Kotov, K. Y., Mal'tsev, A. S., Nesterov, A. A., Pivkin, V. Y., Sobolev, M. A., Filippov, M. N., and Yan, A. P. (2013). Using the kalman filter in the quadrotor vehicle trajectory tracking system. *Optoelectronics, Instrumentation and Data Processing*, 49(6):536–545.

Benallegue, A., Mokhtari, A., and Fridman, L. (2008). High-order sliding-mode observer for a quadrotor uav. *International Journal of Robust and Nonlinear Control*, 18(4-5):427–440.

Bergman, K. and Ekström, J. (2014). *Modeling, Estimation and Attitude Control of an Octorotor Using PID and L1 Adaptive Control Techniques*. Thesis. Linköpings universitet, Sweden.

Bouadi, H. and Tadjine, M. (2007). Nonlinear observer design and sliding mode control of four rotors helicopter. *International Journal of Aerospace and Mechanical Engineering*, 1(7):329–334.

Cabecinhas, D., Naldi, R., Marconi, L., Silvestre, C., and Cunha, R. (2012). Robust takeoff for a quadrotor vehicle. *IEEE Transactions on Robotics*, 28(3):734–742.

Cao, Z., Xiao, Q., Huang, R., and Zhou, M. (2018). Robust neuro-optimal control of underactuated snake robots with experience replay. *IEEE Transactions on Neural Networks and Learning Systems*, 29(1):208–217.

Capello, E., Scola, A., Guglieri, G., and Quagliotti, F. (2012). Mini quadrotor uav: Design and experiment. *Journal of Aerospace Engineering*, 25(4):559–573.

Cen, Z., Noura, H., Susilo, T. B., and Younes, Y. A. (2014). Robust fault diagnosis for quadrotor uavs using adaptive thau observer. *J. Intell. Robotics Syst.*, 73(1-4):573–588.

Chee, K. and Zhong, Z. (2013). Control, navigation and collision avoidance for an unmanned aerial vehicle. *Sensors and Actuators A: Physical*, 190(Supplement C):66 – 76.

Chen, F. and Dunnigan, M. W. (2002). Comparative study of a sliding-mode observer and kalman filters for full state estimation in an induction machine. *IEE Proceedings - Electric Power Applications*, 149(1):53–64.

Choukchou-Braham, A., Cherki, B., and Djemaï, M. (2011). A backstepping procedure for a class of underactuated system with tree structure. In *2011 International Conference on Communications, Computing and Control Applications (CCCA)*, pages 1–6.

DeCarlo, R. A., Zak, S. H., and Matthews, G. P. (1988). Variable structure control of nonlinear multivariable systems: a tutorial. *Proceedings of the IEEE*, 76(3):212–232.

Elamine, L., Meguenni, K., Youssouf, M., and Litim, M. (2013). Nonlinear observer and backstepping control of quadrotor unmanned aerial vehicle. *International Review of Automatic Control (IREACO)*, 6(5).

Elkholy, H. (2014). *Dynamic Modeling and Control of a Quadrotor Using Linear and Nonlinear Approaches*. Thesis. American University in Cairo. School of Engineering Interdisciplinary Program, Egypt.

Escareño, J., Salazar, S., Romero, H., and Lozano, R. (2013). Trajectory control of a quadrotor subject to 2d wind disturbances. *Journal of Intelligent & Robotic Systems*, 70(1):51–63.

Fatan, M., Sefidgari, B. L., and Barenji, A. V. (2013). An adaptive neuro pid for controlling the altitude of quadcopter robot. In *2013 18th International Conference on Methods Models in Automation Robotics (MMAR)*, pages 662–665.

Gonzalez-Hernandez, I., Palacios, F. M., Cruz, S. S., Quesada, E. S. E., and Leal, R. L. (2017). Real-time altitude control for a quadrotor helicopter using a super-twisting controller based on high-order sliding mode observer. *International Journal of Advanced Robotic Systems*, 14(1):1729881416687113.

Grzonka, S., Grisetti, G., and Burgard, W. (2012). A fully autonomous indoor quadrotor. *IEEE Transactions on Robotics*, 28(1):90–100.

Hartman, D. (2014). Quadcopter dynamic modeling and simulation. https://github.com/dch33/quad-sim/tree/master/quadcopter dynamic modeling and simulation.

Hehn, M., Ritz, R., and D'Andrea, R. (2012). Performance benchmarking of quadrotor systems using time-optimal control. *Autonomous Robots*, 33(1):69–88.

Henning, F. C. (1919). Artificial hand. US Patent 1,298,502.

Hermann, R. and Krener, A. J. (1977). Nonlinear controllability and observability. *IEEE Transactions on Automatic Control*, AC-22(5):728–740.

Hoffmann, G. M., Huang, H., Waslander, S. L., and Tomlin, C. J. (2011). Precision flight control for a multi-vehicle quadrotor helicopter testbed. *Control Engineering Practice*, 19(9):1023 – 1036. Special Section: DCDS'09 – The 2nd IFAC Workshop on Dependable Control of Discrete Systems.

Honglei, A., Jie, L., Jian, W., Jianwen, W., and Hongxu, M. (2013). Backstepping-based inverse optimal attitude control of quadrotor. *International Journal of Advanced Robotic Systems*, 10(5):223.

Ibarra, E. and Castillo, P. (2017). Nonlinear super twisting algorithm for uav attitude stabilization. In *Unmanned Aircraft Systems (ICUAS), 2017 International Conference on*, pages 640–645. IEEE.

Jardin, M. R. and Mueller, E. (2009). Optimized measurements of unmanned-air-vehicle mass moment of inertia with a bifilar pendulum. *Journal of Aircraft*, 46(3).

Jeong, S. and Chwa, D. (2018). Coupled multiple sliding-mode control for robust trajectory tracking of hovercraft with external disturbances. *IEEE Transactions on Industrial Electronics*, 65(5):4103–4113.

Kheowree, T. (2016). Quadx 2560gy86 pidauto v5: https://github.com/quadtinnakon.

Leishman, R., Macdonald, J., McLain, T., and Beard, R. (2012). Relative navigation and control of a hexacopter. In *2012 IEEE International Conference on Robotics and Automation*, pages 4937–4942.

Leong, B. T. M., Low, S. M., and Ooi, M. P.-L. (2012). Low-cost microcontroller-based hover control design of a quadcopter. *Procedia Engineering*, 41(Supplement C):458 – 464. International Symposium on Robotics and Intelligent Sensors 2012 (IRIS 2012).

Li, H., Xie, P., and Yan, W. (2017a). Receding horizon formation tracking control of constrained underactuated autonomous underwater vehicles. *IEEE Transactions on Industrial Electronics*, 64(6):5004–5013.

Li, H., Yan, W., and Shi, Y. (2017b). Continuous-time model predictive control of underactuated spacecraft with bounded control torques. *Automatica*, 75:144 – 153.

Liu, H., Bai, Y., Lu, G., Shi, Z., and Zhong, Y. (2014). Robust tracking control of a quadrotor helicopter. *Journal of Intelligent & Robotic Systems*, 75(3):595–608.

Liu, H., Derawi, D., Kim, J., and Zhong, Y. (2013). Robust optimal attitude control of hexarotor robotic vehicles. *Nonlinear Dynamics*, 74(4):1155–1168.

Liu, J. and Wang, X. (2011). *Dynamic Sliding Mode Control*, pages 111–116. Springer Berlin Heidelberg, Berlin, Heidelberg.

López-Martínez, M. (2009). A nonlinear strategy to control unstable underactuated mechanical systems with underactuation ¿ 1 . applications to control augmentations †.

Lu, B., Fang, Y., and Sun, N. (2018). Continuous sliding mode control strategy for a class of nonlinear underactuated systems. *IEEE Transactions on Automatic Control*, PP(99):1–1.

Luque-Vega, L., Castillo-Toledo, B., and Loukianov, A. G. (2012). Robust block second order sliding mode control for a quadrotor. *Journal of the Franklin Institute*, 349(2):719 – 739. Advances in Guidance and Control of Aerospace Vehicles using Sliding Mode Control and Observation Techniques.

Magnussen, ., Ottestad, M., and Hovland, G. (2013). Experimental validation of a quaternion-based attitude estimation with direct input to a quadcopter control system. In *2013 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 480–485.

Magnusson, T. (2014). *Attitude Control of a Hexarotor*. Thesis. Linköpings universitet, Sweden.

Mantz, R., Puleston, P., and Battista, H. D. (1999). Output overshoots in systems with integral action operating in sliding mode. *Automatica*, 35(6):1141 – 1147.

Mendel, J., Hagras, H., Tan, W.-W., Melek, W. W., and Ying, H. (2014). *Introduction To Type-2 Fuzzy Logic Control: Theory and Applications*. Wiley-IEEE Press, 1st edition.

Mobayen, S. (2015). A novel global sliding mode control based on exponential reaching law for a class of underactuated systems with external disturbances. *Journal of Computational and Nonlinear Dynamics*.

Nagaty, A., Saeedi, S., Thibault, C., Seto, M., and Li, H. (2013). Control and navigation framework for quadrotor helicopters. *Journal of Intelligent & Robotic Systems*, 70(1):1–12.

Nikolakopoulos, G. and Alexis, K. (2013). Switching networked attitude control of an unmanned quadrotor. *International Journal of Control, Automation and Systems*, 11(2):389–397.

Olivares-Mendez, M. A., Campoy, P., Mellado-Bataller, I., and Mejias, L. (2012). See-and-avoid quadcopter using fuzzy control optimized by cross-entropy. In *2012 IEEE International Conference on Fuzzy Systems*, pages 1–7.

Ömürlü, V. E., Büyükşahin, U., Artar, R., Kirli, A., and Turgut, M. N. (2013). An experimental stationary quadrotor with variable dof. *Sadhana*, 38(2):247–264.

Ozer, H. O., Hacioglu, Y., and Yagiz, N. (2017). High order sliding mode control with estimation for vehicle active suspensions. *Transactions of the Institute of Measurement and Control*, page 0142331216685394.

Pucci, D., Romano, F., and Nori, F. (2015). Collocated adaptive control of underactuated mechanical systems. *IEEE Transactions on Robotics*, 31(6):1527–1536.

Riachy, S., Orlov, Y., Floquet, T., Santiesteban, R., and Richard, J.-P. (2008). Second-order sliding mode control of underactuated mechanical systems i: Local stabilization with application to an inverted pendulum. *International Journal of Robust and Nonlinear Control*, 18(4-5):529–543.

Ryan, T. and Kim, H. J. (2013). Lmi-based gain synthesis for simple robust quadrotor control. *IEEE Transactions on Automation Science and Engineering*, 10(4):1173–1178.

Sa, I. and Corke, P. (2012). System identification, estimation and control for a cost effective open-source quadcopter. In *2012 IEEE International Conference on Robotics and Automation*, pages 2202–2209.

Salgado, I., Chairez, I., Moreno, J., Fridman, L., and Poznyak, A. (2011). Generalized super-twisting observer for nonlinear systems. *IFAC Proceedings Volumes*, 44(1):14353 – 14358. 18th IFAC World Congress.

Sanca, A. S., Alsina, P. J., and d. J. F. Cerqueira, J. (2010). Dynamic modeling with nonlinear inputs and backstepping control for a hexarotor micro-aerial vehicle. In *2010 Latin American Robotics Symposium and Intelligent Robotics Meeting*, pages 36–42.

Satici, A. C., Poonawala, H., and Spong, M. W. (2013). Robust optimal control of quadrotor uavs. *IEEE Access*, 1:79–93.

Schröder, D., Lenz, U., Beuschel, M., Hangl, F., Frenz, T., Strobl, D., Straub, S., Fischle, K., Rau, M., and Angermann, A. (2000). *Intelligent Observer and Control Design for Nonlinear Systems*. Springer-Verlag, Berlin Heidelberg.

Sebesta, K. D. and Boizot, N. (2014). A real-time adaptive high-gain ekf, applied to a quadcopter inertial navigation system. *IEEE Transactions on Industrial Electronics*, 61(1):495–503.

Shtessel, Y., Edwards, C., Fridman, L., and Levant, A. (2010). *Sliding Mode Control and Observation*. Springer-Verlag, New York.

Slotine, J.-J., Hedrick, J., and Misawa, E. (1986). On sliding observers for nonlinear systems. *Journal of Dynamic Systems, Measurement, and Control*, 109(3):245–252.

Slotine, J.-J. E. and Li, W. (1988). *Applied Nonlinear Control*. Pearson Education, United States.

Su, G. J., Gong, Q. L., and Li, H. Y. (2017). Adaptive sliding-mode control based on decoupled method for a class of underactuated system. In *2017 36th Chinese Control Conference (CCC)*, pages 639–643.

Sun, N., Fang, Y., Chen, H., and Lu, B. (2017a). Amplitude-saturated nonlinear output feedback antiswing control for underactuated cranes with double-pendulum cargo dynamics. *IEEE Transactions on Industrial Electronics*, 64(3):2135–2146.

Sun, N., Wu, Y., Fang, Y., Chen, H., and Lu, B. (2017b). Nonlinear continuous global stabilization control for underactuated rtac systems: Design, analysis, and experimentation. *IEEE/ASME Transactions on Mechatronics*, 22(2):1104–1115.

Tuan, L. A. and Lee, S.-G. (2016). Nonlinear feedback control of underactuated mechanical systems. In Lee, D., Burg, T., and Volos, C., editors, *Nonlinear Systems - Design, Analysis, Estimation and Control*, chapter 10. InTech, Rijeka.

Utkin, V. I. (1977). Variable structure systems with sliding modes: A survey paper. *IEEE Transactions on Automatic Control*, AC-22(2):212–222.

Visioli, A. (2006). *Practical PID Controll*. Springer-Verlag, London.

Wang, H., Zhou, Z., Hao, C., Hu, Z., and Zheng, W. (2017a). Fteso-based finite time control for underactuated system within a bounded input. *Asian Journal of Control*, pages n/a–n/a. asjc.1624.

Wang, X. and Shirinzadeh, B. (2015). Nonlinear augmented observer design and application to quadrotor aircraft. *Nonlinear Dynamics*, 80(3):1463–1481.

Wang, Z., Bao, W., and Li, H. (2017b). Second-order dynamic sliding-mode control for nonminimum phase underactuated hypersonic vehicles. *IEEE Transactions on Industrial Electronics*, 64(4):3105–3112.

Wang, Z., Freidovich, L. B., and Zhang, H. (2017c). Periodic motion planning and control for underactuated mechanical systems. *International Journal of Control*, 0(0):1–13.

Wu, T. S., Karkoub, M., Wang, H., Chen, H. S., and Chen, T. H. (2017). Robust tracking control of mimo underactuated nonlinear systems with dead-zone band and delayed uncertainty using an adaptive fuzzy control. *IEEE Transactions on Fuzzy Systems*, 25(4):905–918.

Xu, R. (2007). *Optimal Sliding Mode Control and Stabilization of Underactuated Systems*. Thesis. The Ohio State University, United States.

Yang, X. and Zheng, X. (2018). Swing up and stabilization control design for an underactuated rotary inverted pendulum system: Theory and experiments. *IEEE Transactions on Industrial Electronics*, PP(99):1–1.

Zain, Z. M., Watanabe, K., Izumi, K., and Nagai, I. (2011). A nonholonomic control method for stabilizing an x4-auv. *Artificial Life and Robotics*, 16(2):202.

Zhang, R., Quan, Q., and Cai, K.-Y. (2011). Attitude control of a quadrotor aircraft subject to a class of time-varying disturbances. *IET control theory & applications*, 5(9):1140–1146.

Zhao, W. and Go, T. H. (2014). Quadcopter formation flight control combining mpc and robust feedback linearization. *Journal of the Franklin Institute*, 351(3):1335 – 1355.

Zheng, E.-H., Xiong, J.-J., and Luo, J.-L. (2014). Second order sliding mode control for a quadrotor uav. *ISA transactions*, 53(4):1350–1356.