Nonlinear and Fault-tolerant Control Techniques for a Quadrotor Unmanned Aerial Vehicle

Tong Li

A Thesis

in

The Department

of

Mechanical and Industrial Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science (Mechanical Engineering) at

Concordia University

Montreal, Quebec, Canada

2011

CONCORDIA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

This is to certify that the Thesis prepared,

By: **Tong Li**

Entitled: **"Nonlinear and Fault-tolerant Control Techniques for a Quadrotor Unmanned Aerial Vehicle"**

and submitted in partial fulfillment of the requirements for the Degree of

Master of Applied Science (Mechanical Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

______________________________ Chair
Dr. S. Rakheja

______________________________ Examiner
Dr. W.F. Xie

______________________________ Examiner
Dr. Z. Tian External
Concordia Institute for Information Systems Engineering

______________________________ Co-Supervisor
Dr. B. Gordon

______________________________ Co-Supervisor
Dr. Y. Zhang

Approved by: ______________________________
Dr. A.K.W. Ahmed, MASc Program Director
Department of Mechanical and Industrial Engineering

____
Dean Robin Drew
Faculty of Engineering & Computer Science

Date: 06-15-2011

ABSTRACT

Nonlinear and Fault-tolerant Control Techniques for a Quadrotor Unmanned Aerial Vehicle

Tong Li

Unmanned Aerial Vehicles (UAVs) have become more and more popular, and how to control them has become crucial. Although there are many different control methods that can be applied to the control of UAVs, nonlinear control techniques are more practical since the nonlinear features of most UAVs. In this thesis, as the first main contribution, three widely used nonlinear control techniques including Feedback Linearization Control (FLC), Sliding Mode Control (SMC), and Backstepping Control (BSC) are discussed, investigated, and designed in details and flight-tested on a unique quadrotor UAV (Qball-X4) test-bed available at the Networked Autonomous Vehicles (NAV) Lab in Concordia University. Each of these three control algorithms has its own features. The advantages and disadvantages are revealed through both simulation and experimental tests. Sliding mode control is well known for its capability of handling uncertainty, and is expected to be a robust controller on Qball-X4 UAV. Feedback linearization control and backstepping control are considered a bit weaker than sliding mode control. A comparison of these three controllers is carried out in both theoretical analysis and experimental results under same fault-free flight conditions. Testing results and comparison show the different features of different control methods, and provide a view on how to choose controller under a specific condition. Besides, safety and reliability of UAVs have been and will always be a critical issue in the aviation industry. Fault-Tolerant Control (FTC) has played an extremely important role towards UAVs'

safety and reliability and the safety of group people if an unexpected crash occurred due to faults/damages of UAVs. Therefore, FTC has been a very active and quickly growing research and development field for UAVs and other safety-critical systems. Based on the use of sliding mode control technique, referred to as Fault-Tolerant SMC (FT-SMC) have been investigated, implemented, flight-tested and compared in the Qball-X4 test-bed and also simulation environment in both passive and active framework of FTC in the presence of different actuator faults/damages, as the second main contribution of this thesis work.

Acknowledgements

I would like to take this opportunity to express the great gratitude of mine to my co-supervisor Dr. Brandon Gordon, who has been a good advisor not only in academic field, but also in life. With his help, I did not only learn new skills or knowledge, but the way of how to gain some skills and knowledge, which is the most important factor in scientific field and engineering. When I made mistakes or got lost, his guidance and advices could always lead me out of the problem and lead me to the success. The achievements I have for the thesis are the strong reasons of his kind support.

I also would like to thank Dr. Youmin Zhang, the other co-supervisor, for his great help. I am very much grateful of what I have learnt from him. If it is not for him, I might not be able to achieve my success.

The colleagues of mine, Ms. Qingli Zhou, Dr. Abbas Chanseddine, Mr. Iman Sadeghzadeh, Mr. Fei Yang, Mr. Sivaram Wijenddra, Mr. Farid Sharifi, and Mr. Mastafa Mirzaei are the people I need to give my thanks to as well. They have given me a lot of information for the progress of completing my thesis and degree.

My family has played another essential role while I was doing research and degree study. My parents have been one hundred percent supported, and they trusted I could finish my work. Thank them for being so supportive.

At last but not least, I need to thank my girlfriend Yu Qin. She never stops encouraging me and trusting me no matter what situation I was in, which makes me push myself towards the success.

In dedication to Mom, Dad and those who have helped me

Table of Contents

List of Figures

List of Tables

Nomenclature

FL	Feedback Linearization Control
SMC	Sliding Mode Control
BSC	Backstepping Control
FTC	Fault-Tolerant Control
FTCS	Fault-Tolerant Control System
PFTC	Passive Fault-Tolerant Control
AFTC	Active Fault-Tolerant Control
IOL	Input-Output Linearization
ISL	Input-State Linearization
SISO	Single-Input and Single-Output system
MIMO	Multiple-Input and Multiple-Output system
IC	Initial Condition
DOF	Degree of Freedom
x	State element
y	Output element
u	Control input element
$f(x)$	System model function
$g(x)$	Control input coefficient function
$h(x)$	Output function
$\alpha(x)$	Virtual system model function
$\beta(x)$	Virtual input coefficient function

v	Virtual control input
$L_f h$	Lie derivative
x_d	Desired state element
y_d	Desired output element
x_d, y_d	Desired global coordinates
e	Tracking error
$\boldsymbol{G}(\boldsymbol{x})$	Matrix of control input coefficients
$\boldsymbol{F}(\boldsymbol{x})$	Matrix of system model functions
k_f	Control gains of feedback linearization
$\boldsymbol{K}_f$	Matrix of control gains of feedback linearization
$\boldsymbol{V}$	Matrix of virtual inputs
$\tilde{x}$	State error
s	Sliding surface
λ, η	Strictly positive numbers of sliding surface
ϕ	Upper boundary of sliding surface
ξ	Boundary parameter
$V(x)$	Lyapunov function
$\hat{f}(x)$	Approximation of $f(x)$
$\hat{g}(x)$	Approximation of $g(x)$
$\hat{u}$	Approximation of u
$\mathrm{sgn}(s)$	Sign function of sliding surface

k_s	Sliding mode control gain
$\bar{f}(x)$	Upper limit function of the difference between approximation and actual functions
$\hat{\boldsymbol{U}}$	Matrix of approximations of control inputs $\hat{u}$
$\boldsymbol{U}$	Matrix of control inputs u
$\boldsymbol{K}_s$	Matrix of control gains k_s of sliding mode control
$\boldsymbol{S}$	Matrix of sliding surface s
$\phi(x)$	Virtual control input
ζ, z	Difference between virtual input and actual input
u'	Transformed control input
α	Control gains of backstepping control
F_1, F_2, F_3, F_4	Four forces generated by each rotor
x, y, z	Coordinates in earth frame
x_q, y_q, z_q	Coordinates in Qball-X4 body frame
θ, ϕ, ψ	Pitch, Roll, Yaw angles
$R_x(\theta), R_y(\phi), R_z(\psi)$	Euler rotation matrices about *x,y,z* axes
$\boldsymbol{R}$	Overall rotation matrix of x,y,z
$\boldsymbol{F}$	Thrust vector generated by rotors
ω	Actuator angular velocity
K_a	Actuator transfer function gain
$\boldsymbol{W}$	PWM input vectors

F_{xq}, F_{yq}, F_{zq}	Forces along *x, y, z* axes
m	Qball-X4 mass
a	Acceleration
f	Friction
G, g	Gravity
d_x, d_y, d_z	Drag coefficients for *x, y, z* axes
d_θ, d_ϕ, d_ψ	Drag coefficients for θ, ϕ, ψ attitude
τ	Torque
F_r	Centripetal force
r	Length between the center and the desired point on the rigid body
J	Moment of Inertial in earth frame
V_c	Overall velocity with translation and rotation motions
v_{cm}	Velocity of center mass
$\boldsymbol{M}$	Translational momentum in earth frame
$\boldsymbol{M}_q$	Translational momentum in body frame
$\boldsymbol{H}$	Angular momentum in earth frame
$\boldsymbol{H}_q$	Angular momentum in body frame
$\boldsymbol{\omega}_q$	Angular velocities of Qball-X4 in body frame
$\boldsymbol{J}_q$	Inertia matrix of *x, y, z* axes in body frame
J_{xx}, J_{yy}, J_{zz}	Inertia about x_q, y_q, z_q body frame axes
J_{xy}, J_{yz}, J_{zx}	Inertia between two axes

J_x, J_y, J_z	Abbreviation of inertia J_{xx}, J_{yy}, J_{zz}
k	Elements of the body frame inertia matrix
L	Lever length of x_q, y_q axes
c	Lever Length of z_q axis
$\boldsymbol{\tau}_q$	Torques in body fixed frame of x_q, y_q, z_q axes
Ω	Disturbance of gyroscopic effects
J_r	Rotor inertia
p	Identified system parameters
u_1, u_2, u_3, u_4	Four control inputs
e_x, e_y, e_z	Tracking errors of x,y,z axes
e_θ, e_ϕ, e_ψ	Tracking errors of θ, ϕ, ψ
u_x	Virtual input for position x
u_y	Virtual input for position y
s_x, s_y, s_z	Sliding surface of x,y,z axes
s_θ, s_ϕ, s_ψ	Sliding surface of θ, ϕ, ψ
$\boldsymbol{\Gamma}$	Mapping matrix
$\boldsymbol{A}, \boldsymbol{B}$	System model coefficients
$\boldsymbol{U}_f$	Control inputs of fault tolerant control

1. Introduction

1.1. Motivation

Due to the recent advances in sensing, communication, computing, and control technologies, unmanned vehicles have become vitally important in the engineering applications and our life. Among many other types of unmanned systems, there are two kinds of most widely investigated and developed unmanned vehicles, UAVs (Unmanned Aerial Vehicles) and UGVs (Unmanned Ground Vehicles). UGVs can be used as ground monitoring robots, and also as a replacement of human force. However there are certain limitations. Since UGVs can only be used on the ground, in some difficult terrain conditions, ground vehicles cannot reach the desired location.

Compared to UGVs, UAVs have greater capabilities. Aerial vehicles can be used to perform a large amount of tasks, such as monitoring forest fires and volcanic activities. They can also support military surveillance and air pollution control etc. There are different types of UAVs: fixed-wing airplanes, conventional helicopters and quadrotor helicopters. Fixed-wing airplanes require special runways to take off from. Both regular helicopters and quadrotors can overcome this flaw and are more flexible. Between these two types, quadrotor helicopters have four rotors more than regular ones, which means that they are more convenient and simpler to be built and to fly, and can possibly take more payload than the conventional helicopters. Quadrotors have received much more attention and interest, because of their special features and advantages. This is one of main motivations for the thesis to use a quadrotor helicopter UAV (Qball-X4) for testing developed nonlinear controllers under normal (fault-free) and fault-tolerant controllers under fault condition s of the UAV.

The Qball-X4 quadrotor helicopter will be discussed in the thesis in details later. It has six-degree of freedom (6DOF), and four-force inputs to four rotors respectively. With all the coupled states, autonomous control could be tricky on occasion. In the following sections, three different nonlinear control algorithms, feedback linearization control, sliding mode control, backstepping control, as well as concept of fault-tolerant control will be reviewed before further discussions.

1.2. Literature Review

In this section, existing feedback linearization control, sliding mode control, backstepping control, and fault-tolerant control algorithms will be reviewed and other commonly used control methods will be discussed for a purpose of comparison.

1.2.1. Feedback Linearization Control

Feedback Linearization Control (FLC) is one of the most commonly used nonlinear control approaches and can be explained as *linearization of a nonlinear system through feedback*. Unlike the state feedback control, FLC can be applied directly to a nonlinear system without linear approximation. This approach transforms the states and the dynamics of the nonlinear system into linear ones. Therefore, after such a transformation, many linear control algorithms can also be used to make the control problem simpler.

A standard feedback linearization control is developed in [2] for tracking task. Since FLC requires invertible matrices, a dynamic extension has been introduced to handle the noninvertible matrices in both [2] and [3]. Kimm *et al* suggest another solution called generalized inverse based on least-square technique that can be used to deal with noninvertible or nonsquare matrices [4]. A robust feedback linearization based on Sobolev norm is developed in [5]. Mokhtari *et al* [5] combine both state feedback and feedback linearization together to transform the nonlinearity of the quadrotor dynamics for inner controller, and an improved H-infinity optimal controller (GH_∞) is applied for outer controller to achieve a desired trajectory tracking performance. Similarly in [6], the overall controller of quadrotor is separated into two loops, which are the inner loop and the outer loop. The difference between the controllers suggested in [5] and [6] is that the one in [6] is using only the feedback linearization control algorithm for both the inner loop (pitch-roll-yaw-z) controller, and the outer loop (*x-y-z-yaw*) controller. The desired trajectory will be given to outer loop controller, *x*, *y*, and *z*, and then desired pitch, roll, and yaw angles can be found by calculations through position control (outer loop controller). A similar procedure is developed in [7]. In previous references, the feedback linearization control is realized in different ways, different combinations, and sometimes with a high price too, which is caused by differentiating equations to find the control inputs. In references [8] and [9], there is a solution by combining the feedback linearization with the sliding mode observer. This combination can effectively reduce the order of derivatives to a lower level and also the number of sensors by adding an estimation of the sliding mode into the overall controller. All the papers introduced so far are focused on quadrotor helicopter. For a regular small-scale helicopter, feedback linearization control is also possible for implementation. Reference [10] has proved that a full nonlinear system of a small-scaled helicopter can be feedback linearized. Feedback linearization is popular as well in other areas. Oriolo *et al* [12] propose an implementation of FLC in wheeled mobile robots tracking task in [10]. From [12], a good trajectory tracking performance of PUMA 560 robot manipulator is achieved by using discontinuous feedback linearization rather than a PID control, which makes the controller more suitable for an electrically driven high speed robot manipulator. Fuzzy control is a powerful tool for handling system uncertainties and noises, and feedback linearization needs an inversion of the system. When the system and environment are uncertain, feedback linearization control alone might not be suitable enough as the controller due to its sensitivity to modeling errors, uncertainties, and noises, and thus reference [13] provides a possible solution by the combination of these two methods. In reference [14], a popular pendulum problem is solved by an input-output feedback linearization cascade controller.

1.2.2. Sliding Mode Control

Sliding Mode Control (SMC) is another advanced nonlinear control technique, with also strong robust abilities as the main feature of such a controller compared with the previous FLC algorithms. Sliding mode control has a sliding surface, which shows how the system converges. By adding a sign function, complexity can be reduced to a minimum so as to increase the stability of control system. For a rather complicated model with some uncertain parameters or dynamics, using controllers such as feedback linearization control which requires a precise model, will be inappropriate and inaccurate. Hence, the sliding mode control is chosen instead. SMC shows a strong capability of dealing with modeling errors, system uncertainties and external disturbances, as long as the sliding condition is satisfied.

In paper [2], an adaptive sliding mode control is proposed. Combining both sliding mode and adaptation law, the controller performs very well against system uncertainties and disturbances. Reference [8] shows how the performance can be improved after adding sliding mode control. In [15], a sliding mode controller has been developed to demonstrate its stability. In [16], Guisser and Medromi present both an observer and a controller that all use sliding mode control algorithms. By observing the unmeasured parameters, pitch, roll and velocities, the controller of *x-y-z-yaw* can be designed. This paper presents a successful improvement in reducing the number of sensors, as well increasing asymptotic stability. Bouadi *et al* [17] and Mokhtari *et al* [18] provide a similar idea to overcome uncertain parameters and external disturbances. In [19], Mokhtari *et al* present a three-way combination: using GH_{∞} for outer loop controller (*x-y-z-yaw*), feedback linearization for inner loop controller (pitch-roll-yaw), and sliding mode for observer. This work uses the advantages of each control method to optimize the overall performance under any circumstances. Reference [20] presents an altitude control using sliding mode to stabilize *x-y-z* directions, as well as pitch-roll-yaw angles. Chattering is always a big problem in SMC, authors from reference [21] have provided an alternative exponentially decaying function to replace the sign function to eliminate chattering as much as possible. For quadrotor UAV, there are four rotors in the system. If any one of these four rotors has degradation/malfunction, the entire system will be seriously affected. Niu *et al* present a design using SMC to handle the situation [22]. Time delay is another factor that can cause damage in system. In [23], with free weighting matrices approach and adaption law added in the controller, SMC has been proven effective in the presence of time delay. In other fields, power quality and stability are very crucial. For UAVs, if the power is unstable, the vehicles will crash. Reference [24] has shown by using a SMC the performance of voltage balancing and regulation are well achieved, and the response to the transience has become fast. Sliding mode control is also effective in formation control, and reference [25] has proved the capability of SMC in a general formation of autonomous vehicles.

1.2.3. Backstepping Control

Back-stepping Control (BSC) is a relatively new nonlinear control technique developed since 1990s based on Lyapunov function, which allows us to choose which system nonlinearity needs to be cancelled and which can be kept. In comparison, feedback linearization cancels all the nonlinearity at the same time, thus barely leaves us any choices for faster system response. However with an appropriate Lyapunov function chosen, and the necessary system nonlinearity kept, a relatively faster convergence can be realized by using BSC concept. When the conditions are met, backstepping is a good choice. As the name implies, the principle of the backstepping algorithm is that a designed controller starts to control the furthest state from the actual control input, and then approaches the input one step at a time. Finally, with all the steps together the overall control input is attained and named backstepping control.

In [26], a backstepping controller has been used as a baseline controller, which supports a followed sliding mode control for controlling an indoor micro quadrotor. By combining these two controllers, it can be shown that backstepping control has a strong capability in stabilizing system by a good Lyapunov function, which is presented in [27] as well. Reference [28] presents a view that underactuated system of quadrotor can be changed into different subsystems, underactuated systems, fully-actuated systems and single propeller systems. By adding seven Lyapunov functions into three subsystems, the altitude *x*-*y*-*z* and attitude pitch-roll-yaw can be controlled at the same time. In [29], authors explained and analyzed in details of designing a backstepping control based on Lagrange form, and also estimated the aerodynamic components by introducing two neural nets. Reference [30] introduces a combination of control algorithm with backstepping and PID. Papers [31] and [32] apply a backstepping controller on a quadrotor using a vision feedback for the *x*-*y*-*z* position tracking. In case there are some unmeasured states and without use of any observers, implementation of a backstepping control will be difficult. Reference [33] presents an alternative way to handle the situation. By adding two extended Kalman filters as an estimation method, authors successfully develop a backstepping controller to overcome the drawback of unmeasured states or system parameters. In [34], a sliding mode based integral filter is used to enhance the backstepping control, and the result shows that the backstepping controller has become more robust. Reference [35] presents a relatively standard procedure of designing a backstepping control for an autonomous helicopter. In [36], a new way of designing PID controller has been introduced. By adding the backstepping structure and combining H_∞ optimal control algorithm, the conventional PID control gains can then be solved by Riccati equations and reduced into two parameters for a helicopter hovering problem. The model uncertainties and the external disturbances can be solved by the enhanced controller. In both references [37] and [38], Saber and Aneke provide some solutions based on backstepping methodology to solve the tracking problems of underactuated mechanical systems.

1.2.4. Fault-Tolerant Control

The time of travelling to different places could be much shorter than before, due to the advanced aviation technology. However, if the system fails, the consequences also could be catastrophic. System faults occur rarely, but unpredictably and mostly suddenly. Therefore, Fault Tolerant Control (FTC) has become more important than ever.

A recent comprehensive overview on FTC is presented in [39] which classifies FTC strategies as Passive Fault Tolerant Control (PFTC), reconfigurable or Active Fault Tolerant Control (AFTC) which makes use of the information from the Fault Detection and Diagnosis (FDD) during operation of the FTC system (FTCS). Safety, reliability and reconfigurability analysis are also included in the paper to make a link for the currently individual research works between control engineering and safety engineering. Some key points in FTCS were also summarized in an early review paper [40] for summarizing control design methods developed up to 1997. Zhang [41] summarized a fault modeling method in FTCS for three different situations on sensor faults, actuator faults, and system dynamic faults. Fekih *et al* [42] presented a passive fault-tolerant control methodology using sliding surface and Lyapunov function to eliminate the pre-specified faults for the model of an F-18 aircraft. The results show that the design is effective. Reference [43] presented an integrated design procedure for fault detection, diagnosis, and reconfigurable control. A two-stage adaptive Kalman filter is used in fault detection and diagnosis scheme. The reconfigurable feedback and feedforward controllers are developed in details as well. Milhim *et al* [44] designed a gain scheduling based PID controller for FTC of a quadrotor UAV under simulation environment. A backstepping control based fault-tolerant control systems is developed for UAV system in [45], and in

[46], by combining the idea of adaptive algorithm, the backstepping control has been reformed into an adaptive backstepping control with more robustness. In [47], a sliding mode based fault-tolerant control has been designed for a civilian fixed-wing aircraft, Boeing 747. The elevator failure is simulated and the simulation results show that the performance of the controller is good. Alwi and Edwards [48] proposed another method using sliding mode scheme with control allocation for fault-tolerant control of B747. With on-line control allocation, an active fault-tolerant control has been successfully designed and simulated using sliding mode control.

1.3. Thesis Contribution and Organization

In this thesis, the first goal is to design and implement three nonlinear controllers based on three different strategies: feedback linearization, sliding mode, and backstepping controls, and to test and evaluate the three algorithms in the real Qball-X4 quadrotor UAV test-bed available at Concordia University. The second goal is to develop and test a passive fault-tolerant control and an active fault-tolerant control strategy based on the developed sliding mode control technique for handling actuator faults and propeller damages in the UAV test-bed. To achieve the above goals, all these controllers are investigated and developed in details. Simulations are used to test if all the theoretically designed controllers function properly under different conditions, and experiments are the final proof of how they behave. Hence, each controller will be focused on practical usage, which means unnecessary assumptions will have to be reduced to the minimum in order to have a more realistic situation. Before experimental implementation of the controllers, Qball-X4 model has been experimentally identified and tested. The results of both simulations and experiments will provide a detailed insight on how to control a quadrotor helicopter.

Thesis organization is outlined as following:

Chapter 1 is about the motivation and literature review. Chapter 2 is regarding to all the background knowledge and theories of all three controllers, in the preparation of the later simulations as well as the experiments. Chapter 3 is modelling of the Qball-X4 UAV. Before all the simulations and experimental tests are carried out, a good and precise model is always needed, especially in this thesis experiments are needed for further testing controllers. Therefore, a detailed discussion of the Qball-X4 UAV model will be carried out in Chapter 3. Chapter 4 is by the background theories of three controllers and the model dynamics equations, practical implementations on specific Qball-X4 system will be demonstrated in both mathematical derivations and numerical simulations. Experiments will be used as a strong proof to show how the performance of the designed controllers is and how close the simulations are to the reality. Chapter 5 will introduce the fault-tolerant control concept, and based on the predesigned sliding mode controller, a passive fault-tolerant control and an active fault-tolerant control have been designed and implemented respectively. Both simulation and experimental testing results will show how the control systems behave and if the designs are suitable for the Qball-X4 system. Chapter 6 concludes all the work that has been done, and summarizes the possible improvements as the future work.

1.4. Summary

This chapter has reviewed the works that have been done on quadrotor using FLC, SMC, BSC or FTCS. All the studies have shown these three controllers are effective and have good control performance. However, most of them are achieved in simulation environment, lack of practical proof on how well the controllers can behave in reality. Therefore, this thesis will redesign all theses controllers, FLC, SMC, BSC, and FTCS to control the quadrotor, Qball, and also implement the controllers in real environment to show the effectiveness of the control systems in practise as the final goal.

2. Background Material

All the background theories will be introduced in this chapter. Detailed mathematical procedure, stability discussion, block diagrams, and some examples to demonstrate the implementation of theories into practical controllers will be presented and well illustrated. This chapter is served as a detailed technical preparation for Chapter 4 of the thesis.

2.1. Feedback Linearization Control

Feedback Linearization Control (FLC) is a nonlinear control technique that can cancel the system nonlinearity and transform a nonlinear system into a linear system, and then many control algorithms for linear systems can be applied to the system controller design. By doing such a transformation, nonlinear control problem can be simplified to a linear problem.

An example of the general form of a single-input single-output nonlinear system is shown below [48]:

$$\begin{aligned} &\qquad\quad g(x)u \\ &y = h(x) \end{aligned} \tag{2-1}$$

where $x(t) \in R^n$ is the system state, $u(t) \in R^m$ is the control input, $y(t) \in R^p$ is the system output, $f(x)$ and $g(x)$ are model functions in R^n.

Assuming that all the states are available for measurement, and the control input u can be formed in the following format:

$$u = \alpha(x) + \beta(x)v \tag{2-2}$$

where v is a new control variable， $\alpha(x)$ and $\beta(x)$ are virtual system functions.

In order to design a FLC, a link between the desired output y and the control input u is needed to be found. The Lie derivative ($L_f h$) is introduced into the theory as follows [48]:

$$y^0 = h(x) \tag{2-3}$$

$$y^1 = L_f h(x) \tag{2-4}$$

$$\vdots$$

$$y^{\rho-1} = L_f^{\rho-1} h(x) \tag{2-5}$$

$$y^{\rho} = L_f^{\rho} h(x) + L_g L_f^{\rho-1} h(x) u \tag{2-6}$$

In equation (2-6), by setting control input u as:

$$u = \frac{v - L_f^{\rho} h(x)}{L_g L_f^{\rho-1} h(x)} = \frac{1}{L_g L_f^{\rho-1} h(x)} (-L_f^{\rho} h(x) + v) \tag{2-7}$$

A simple linear relation is achieved:

$$y^{\rho} = v \tag{2-8}$$

Functions $\alpha(x)$ and $\beta(x)$ will be obtained in the following equations [48]:

$$\alpha(x) = \frac{L_f^{\rho} h(x)}{L_g L_f^{\rho-1} h(x)} \tag{2-9}$$

$$\beta(x) = \frac{1}{L_g L_f^{\rho-1} h(x)} \tag{2-10}$$

Therefore, a controller can be designed with the equation (2-7) as control input. A diagram is shown as following:

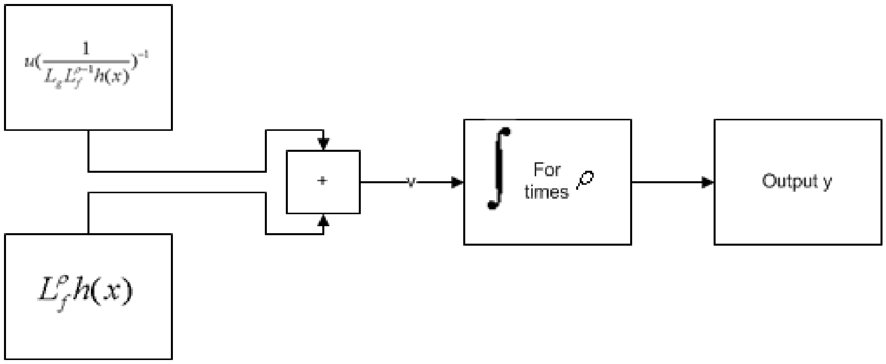

Fig. 2-1. Relation between input *v* and output *y*

The stability of a control system is always a big concern, and in a FLC, external dynamics can be observed directly when designing an input-output linearization, but internal dynamics needs to be investigated carefully. With a certain control input, if output can be maintained at zero, then it is called zero-dynamics. When zero-dynamics is satisfied, which means a zero input *u* will make a zero output *y* at all times, the system is considered stable.

The general concept of FLC has been discussed above, and what follows are some tracking tasks to show how a FLC behaviour has been taken into consideration. There are normally two situations when solving a tracking problem. One is when the system has only one input, and the other is when the system has more than one input.

For a single input system, after a desired trajectory y_d is defined, and the tracking error is defined as $e = y_d - y$, where y_d is the new target output to be controlled. Assuming after *n* times derivatives with respect to output *y*, the input *u* appears. Defining a control gains matrix $\boldsymbol{K}_f$,

$$v = \boldsymbol{K}_f e = e^{(n)} - k_{fn-1} e^{(n-1)} - \cdots \tag{2-11}$$

and having (2-11) to be substituted into equation (2-7) to have the tracking control input u can be found as shown below [48]:

$$
\begin{aligned}
u &= \frac{1}{L_g L_f^{n-1} e}(-L_f^n e + v) = \frac{1}{L_g L_f^{n-1} e}(-L_f^n e + e^{(n)} - k_{fn-1} e^{(n-1)} - \cdots \\
&= G(x)^{-1}(-F(x) + v)
\end{aligned}
\tag{2-12}
$$

where k_f is the control gain to make the system converge.

For multiple inputs system, similar to the idea from single input but with a few changes, the general form will then be changed to [49]:

$$
y_i^{n_i} = L_f^{n_i} h_i(x) + \sum_{j=1}^{m} L_{g_j} L_f^{n_i - 1} h_i(x) u_j \tag{2-13}
$$

The term $\sum_{j=1}^{m} L_{g_j} L_f^{n_i-1} h_i(x) u_j$ can be translated into a matrix format, and the equation (2-13) can then be rewritten as:

$$
\begin{bmatrix} y_1^{n_1} \\ \vdots \\ y_i^{n_i} \end{bmatrix}
\begin{bmatrix} L_f^{n_1} h_1(x) \\ \vdots \\ L_f^{n_i} h_i(x) \end{bmatrix}
\begin{bmatrix} L_{g_1} L_f^{n_1-1} h_1(x) & \cdots & & {}^{-1} h_1(x) \\ \vdots & \vdots & \vdots & \vdots \\ L_{g_1} L_f^{n_i-1} h_i(x) & \cdots & & {}^{-1} h_i(x) \end{bmatrix}
\begin{bmatrix} u_1 \\ \vdots \\ u_j \end{bmatrix}
\tag{2-14}
$$

The control input $u_1 \cdots$ can be written as:

$$
\begin{bmatrix} u_1 \\ \vdots \\ u_j \end{bmatrix}
\begin{bmatrix} L_{g_1} L_f^{n_1-1} h_1(x) & \cdots & & {}^{-1} h_1(x) \\ \vdots & \vdots & \vdots & \vdots \\ L_{g_1} L_f^{n_i-1} h_i(x) & \cdots & & {}^{-1} h_i(x) \end{bmatrix}^{-1}
\begin{bmatrix} y_1^{n_1} - L_f^{n_1} h_1(x) \\ \vdots \\ y_i^{n_i} - L_f^{n_i} h_i(x) \end{bmatrix}
\tag{2-15}
$$

Once the control inputs are found, by letting tracking error as $e_i = y_{id} - y_i$ and

$$
v_i = \boldsymbol{K}_{\boldsymbol{fi}} e_i = e_i^{(n)} - k_{fni-1} e_i^{(n-1)} - \cdots \tag{2-16}
$$

the tracking control can be easily written as:

$$\begin{bmatrix} u_1 \\ \vdots \\ u_j \end{bmatrix} \begin{bmatrix} L_{g_1}L_f^{n_1-1}h_1(x) & \cdots & & {}^{-1}h_1(x) \\ \vdots & \vdots & \vdots & \vdots \\ L_{g_1}L_f^{n_i-1}h_i(x) & \cdots & & {}^{-1}h_i(x) \end{bmatrix}^{-1} \begin{bmatrix} v_1 - L_f^{n_1}h_1(x) \\ \vdots \\ v_i - L_f^{n_i}h_i(x) \end{bmatrix} \tag{2-17}$$
$$= \boldsymbol{G}(\boldsymbol{X})^{-1}[\boldsymbol{V} - \boldsymbol{F}(\boldsymbol{X})]$$

where $\boldsymbol{G}(\boldsymbol{X}), \boldsymbol{F}(\boldsymbol{X}), \boldsymbol{V}$ are all in the format of matrices.

Here is a simple example using feedback linearization control.

Consider a system described as [48]:

$$\begin{aligned} &\dot{} \quad \psi \\ &\dot{} \quad \psi \\ &\dot{} \\ &y_1 = x_1 \\ &y_2 = x_2 \end{aligned} \tag{2-18}$$

and rewrite it into matrix form as:

$$\begin{bmatrix} \dot{} & \psi & 0 \\ \dot{} & \psi & 0 \\ \dot{} & & 1 \end{bmatrix} \begin{bmatrix} u_1 \\ u_2 \end{bmatrix} \quad \begin{bmatrix} y_1 \\ y_2 \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \end{bmatrix} \tag{2-19}$$

Following the FLC design procedure discussed above, input $\boldsymbol{u} = \begin{bmatrix} u_1 & u_2 \end{bmatrix}^T$ needs to be shown in the output $\boldsymbol{y} = \begin{bmatrix} y_1 & y_2 \end{bmatrix}^T$. By taking derivatives on output vector $\boldsymbol{y}$, equation (2-19) can be derived as:

$$\begin{aligned} &\dot{} \quad \dot{} \quad \psi \\ &\dot{} \quad \dot{} \quad \psi \end{aligned} \tag{2-20}$$

From (2-20), input u_1 appears, but still missing input u_2. Therefore, a second derivative is needed to have the u_2 to appear.

$$\begin{aligned} &\ddot{\ } \quad \dot{\ } \quad \dot{\ } \quad \cos\psi \\ &\ddot{\ } \quad \dot{\ } \quad \dot{\ } \quad {}_1\sin\psi \end{aligned} \tag{2-21}$$

Now, all the inputs have been shown and rewritten as:

$$\ddot{\boldsymbol{y}} \quad \begin{bmatrix} \dot{\ } \quad \psi & u_1\cos\psi \\ \cos\psi & -u_1\sin\psi \end{bmatrix} \begin{bmatrix} \dot{\imath} \\ u_2 \end{bmatrix} \tag{2-22}$$

There is one more input $\dot{\imath}$ than original system, and it can be treated simply as a derivative of u_1. In practice, an integrator can be used to turn $\dot{\imath}$ back to u_1.

If $\ddot{\boldsymbol{y}} \quad \boldsymbol{v}$, then equation (2-22) will be transformed into the following:

$$\boldsymbol{v} = \begin{bmatrix} \sin\psi & u_1\cos\psi \\ \cos\psi & -u_1\sin\psi \end{bmatrix} \boldsymbol{u} \tag{2-23}$$

According to (2-23) the control inputs can be derived as:

$$\boldsymbol{u} = \begin{bmatrix} \sin\psi & u_1\cos\psi \\ \cos\psi & -u_1\sin\psi \end{bmatrix}^{-1} \boldsymbol{v} \tag{2-24}$$

If $u_1 = u_2 = 0$, the output $y_1 = x_1 = 0,\ y_2 = x_2 = 0$, which is satisfied with zero-dynamics. The system is then stable.

To design a controller for this system, v and k need to be defined as:

$$\begin{aligned} v_1 &= (\ddot{\ } \quad \ddot{\ } \quad \dot{\ } \quad \dot{\ } \quad y_{1d} - y_1) \\ v_2 &= (\ddot{\ } \quad \ddot{\ } \quad \dot{\ } \quad \dot{\ } \quad y_{2d} - y_2) \end{aligned} \tag{2-25}$$

where the control gains are $k_{f1} = 8,\ \ k_{f2} = 10,\ \ k_{f3} = 8,\ \ k_{f4} = 10$. The simulation results are shown in Fig. 2-2 and Fig. 2-3.

All the initial condition for all the states and inputs are set to 0 s.

The tracking task is to track the reference trajectory $y_{1d} = 3,\ y_{2d} = 5$.

Simulation results show the desired tracking has been achieved successfully.

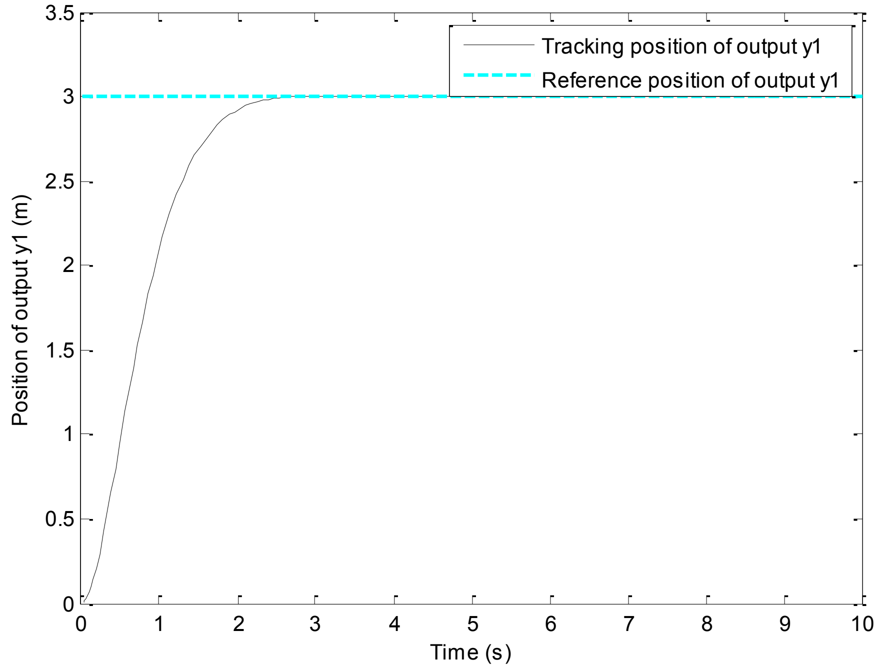

Fig. 2-2. Tracking of output y_1

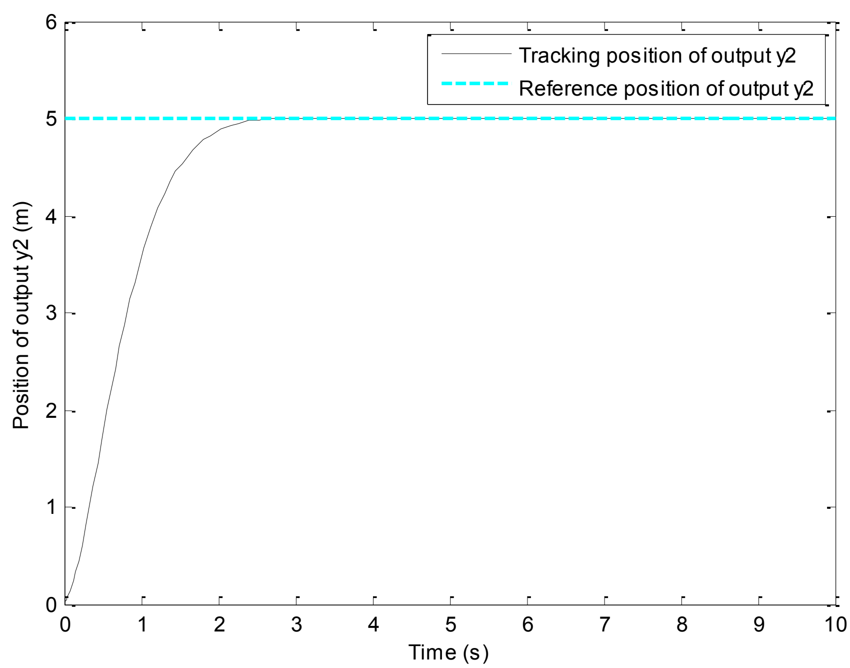

Fig. 2-3. Tracking of output y_2

2.2. Sliding Mode Control

Among robust nonlinear control algorithms, Sliding Mode Control (SMC) is a popular control technique. Sliding mode control has a sliding surface which can provide the stability of the controller and the system. In a non-ideal model, uncertainties can always cause problems for designing a controller. This control technique provides a switching control, which can handle the system uncertainties very well by limiting the amplitude of signals with constraints.

In the same form as for feedback linearization control, a single input single output system is described as:

$$\begin{aligned} &\qquad\qquad g(x)u \\ &y = h(x) \end{aligned}$$

Sliding surface is the most important component in the system, since it will determine the stability of the controller and the control inputs.

In a real system, $f(x)$ and $g(x)$ may have some uncertainties, and the goal of sliding mode control is to control the uncertainties and set a boundary on any uncertain parameters. To do this, an error tracking system needs to be defined to measure the difference between desired value, x_d, and actual value, x. For such a purpose, the tracking error is defined as $\tilde{}$, and a sliding surface is then defined as [49]:

$$s(e;\ t) = (\frac{d}{dt} + \lambda)^{n-1} \tilde{} \tag{2-26}$$

For instance, if the order of the system is $n = 2$, sliding surface can be extended as $s =$, where λ is defined as a positive value. The order of the system can be reduced by 1.

If there is a bound on surface vector $\boldsymbol{s}$, there will be a bound on tracking error vector $\tilde{\boldsymbol{e}}$ as follows [50]

$$\forall t \ge t_0,\ |\boldsymbol{s}| \le \phi \Rightarrow |\tilde{\boldsymbol{e}} \quad {}^{i}\xi,\ i = 0,1,\ldots \tag{2-27}$$

where $\xi = \phi / \lambda^{n-1}$.

Lyapunov stability is a powerful tool that can be used to test a system's stability. If a Lyapunov function chosen as $V(s) = \frac{1}{2}s^2$ is a positive definite function, then to stabilize the system, $\dot{V}$ needs to be a negative semi-definite or negative definite function.

$$\dot{V} \quad \frac{1}{2}\frac{d}{dt}s^2 = s\dot{}$$

By choosing the following condition [49]:

$$\frac{1}{2}\frac{d}{dt}s^2 = s\dot{} \tag{2-28}$$

where η can take only positive values, surface s can be kept at zero, and sliding condition is then defined.

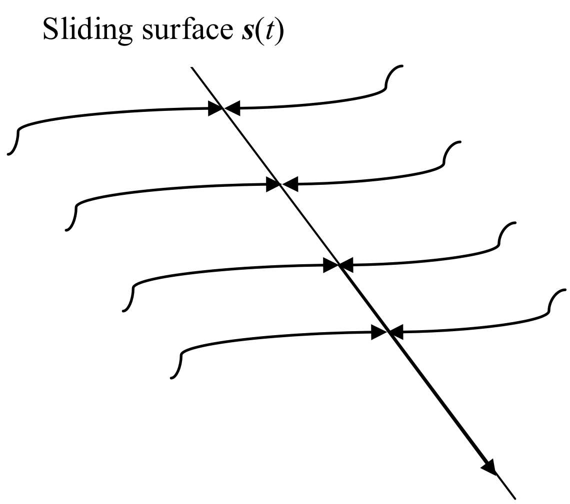

Fig. 2-4. Sliding mode condition

When various system states reach the sliding surface, the system is considered stable. Therefore, when system is stably controlled, the tracking error e will become zero, and by (2-26), sliding surface $s(t)$ can easily be proved as a function of the form.

$$s(\tag{2-29}$$

Once the control system has reached the sliding surface, an equivalent dynamics can be derived based on Filippov's construction. The dynamics of the sliding surface can be written as [49]:

$$\tag{2-30}$$

By solving the equation (2-30), a control input can be found, which is defined as an equivalent control input, $\hat{u}$. For instance, if we have a single input system similar to the form in (2-26), but in second order of the following format:

$$+ g(x)u \tag{2-31}$$

The sliding surface should be chosen as:

$$s = (\frac{d}{dt} + \lambda)^{n-1} \tag{2-32}$$

By taking the first derivative of the surface s in equation (2-32) and combining it with equation (2-30) to form Filippov's construction, the following equations would be satisfied.

$$\tag{2-33}$$

$$\tag{2-34}$$

Solving equation (2-34), an approximated control input $\hat{u}$ can be easily obtained as follows:

$$\hat{u} = g(x)^{-1}(-\hat{f}(x) + \tag{2-35}$$

where $\hat{f}(x)$ is an approximation of $f(x)$. The control input of the system can then be achieved [49].

$$u = \hat{u} + k_s \operatorname{sgn}(s) \tag{2-36}$$

where

$$\begin{cases} \operatorname{sgn}(s) = -1 & \text{if} \quad s < 0 \\ \operatorname{sgn}(s) = 0 & \text{if} \quad s = 0 \\ \operatorname{sgn}(s) = 1 & \text{if} \quad s > 0 \end{cases}$$

By defining a function as:

$$|\hat{f}(x) - f(x)| \le \bar{f}(x) \tag{2-37}$$

and combining it with equation (2-28), the sliding condition can then be derived as:

$$\frac{1}{2}\frac{d}{dt}s^2 = s\dot{} \qquad - f(x))s - k_s |s| \le -\eta |s| \tag{2-38}$$

where $k_s = \bar{f}(x) + \eta$. Thus, sliding condition is satisfied, and the system is considered stable.

The above example is used as a demonstration for a single-input nonlinear system. In a multiple-input system, the general form needs to be rewritten in the following form [49].

$$x_i^{(n_i)} = f_i(x) + \sum_{j=1}^{m} g_{ij}(x) u_j \qquad i = 1, \ldots \qquad \ldots \tag{2-39}$$

Equation (2-39) can be rearranged into matrix format as the following:

$$\begin{bmatrix} x_1^{n_1} \\ \vdots \\ x_i^{n_i} \end{bmatrix} \begin{bmatrix} f_1(x) \\ \vdots \\ f_i(x) \end{bmatrix} \begin{bmatrix} g_{11}(x) & \cdots & &) \\ \vdots & \vdots & \vdots & \\ g_{i1}(x) & \cdots & &) \end{bmatrix} \begin{bmatrix} u_1 \\ \vdots \\ u_j \end{bmatrix} \tag{2-40}$$

The sliding mode surface will then be changed accordingly as follows:

$$s_i = (\frac{d}{dt} + \lambda_i)^{n_i - 1} \tilde{} \tag{2-41}$$

and also the sliding condition will then be [49]

$$\frac{1}{2}\frac{d}{dt}s_i^2 = s_i \dot{} \qquad (2\text{-}42)$$

Taking $n = 2,$ the sliding surface will then take the form of the equation (2-41).

$$s_i = (\qquad (2\text{-}43)$$

$$= \qquad \sum_{j=1} \qquad (2\text{-}44)$$

Following the procedure of a single input system, $\hat{u}$ can be written as a matrix:

$$\begin{bmatrix} \hat{u}_1 \\ \vdots \\ \hat{u}_j \end{bmatrix} \begin{bmatrix} g_{11}(x) & \cdots &) \\ \vdots & \vdots & \vdots \\ g_{i1}(x) & \cdots &) \end{bmatrix}^{-1} \begin{bmatrix} -\hat{f}_1(x) + \\ \vdots \\ -\hat{f}_i(x) + \end{bmatrix} \qquad (2\text{-}45)$$

Hence, overall control input u can be achieved as:

$$\boldsymbol{U} = \hat{\boldsymbol{U}} + \boldsymbol{K}_s \operatorname{sgn}(\boldsymbol{S}) \qquad (2\text{-}46)$$

where $\boldsymbol{U}$, $\hat{\boldsymbol{U}}$, $\boldsymbol{K}_s$, $\boldsymbol{S}$ are all matrices representing overall control inputs, approximated control inputs, control gains, and sliding surfaces, respectively, with control gains greater than or equal to zero.

Note: In switching control, a sign function is used to generate two different outputs, +1 and -1 in the controller as the switch. This function can improve the robustness by constraining control signals to the sliding surface. However, this function can also cause a phenomenon known as *discontinuity* or *chattering*, where signals jump up and down across the surface like series of pulses. In practice, chattering can sometimes be intolerable, therefore a saturation function can then be used instead of the sign function. The saturation function will force the signals to go smoothly in the boundary layer to eliminate chattering.

An example is used here to illustrate in details how to construct a sliding mode controller.

Consider a common pendulum system [50]

$$\begin{bmatrix} \dot{} \\ \dot{} \end{bmatrix} \quad \begin{bmatrix} x_2 \\ \sin x_1 - b x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ c \end{bmatrix} u \tag{2-47}$$

with $\frac{1}{2} \le a \le 2 \quad 0 \le b \le 3 \quad \frac{1}{2} \le c \le \frac{3}{2}$, where these three parameters are treated as uncertainties.

This system has only one input, which makes it an underactuated nonlinear system. In order to stabilize both states x_1, x_2, if x_1 is chosen as the output, $y = x_1$, then the sliding surface has to be chosen according to equation (2-29).

$$s = \tag{2-48}$$

where , $e_{x1} = x_{1d} - x_1$. The derivative of surface s is:

$$= \tag{2-49}$$

According to equation (2-30), by letting , the approximated $\hat{u}$ is then given by

$$\hat{u} = \frac{1}{\hat{c}}(\tag{2-50}$$

Then, the control input is calculated by equation (2-36) as following:

$$u = \hat{u} + k_s \operatorname{sgn}(s) \tag{2-51}$$

Now the stability of this designed control input should be discussed. By sliding condition (2-28), it can be easily proven that the following inequality is satisfied.

$$s\dot{}$$

$$\leq -(-\eta_1 \sin x_1 - \eta_2 x_2 - \eta_3 (\dot{} \qquad s| \leq -\eta |s| \tag{2-52}$$

where $-\eta_1 \sin x_1 - \eta_2 x_2 - \eta_3 (\dot{} \qquad \eta$. The sliding condition is satisfied, and the designed controller is stable.

A simulation is taken to further demonstrate how the above controller works and how the performance is. With gains $\lambda = 5$, $k_s = 15$, and uncertain parameters are $a = \dfrac{3}{2}$, $b = 2$, $c = 1$, the results are shown in Fig. 2-5.

All the initial conditions of input and states are set to 0 s, and the tracking requires to follow the reference $y = 4$.

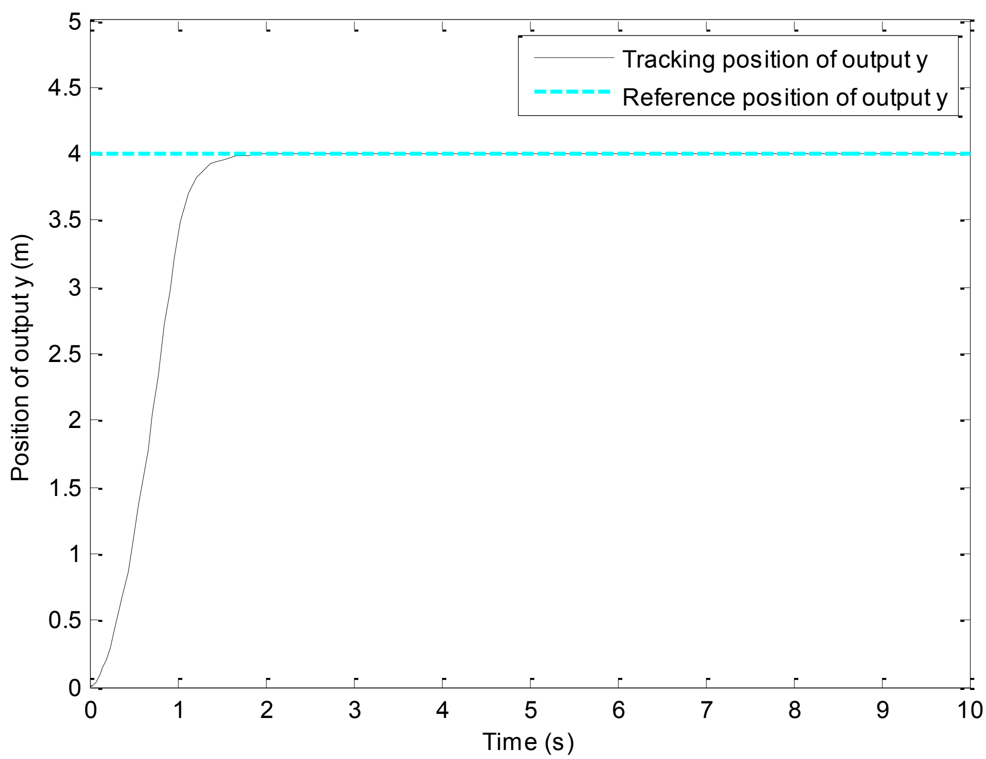

Fig. 2-5. Tracking of output y

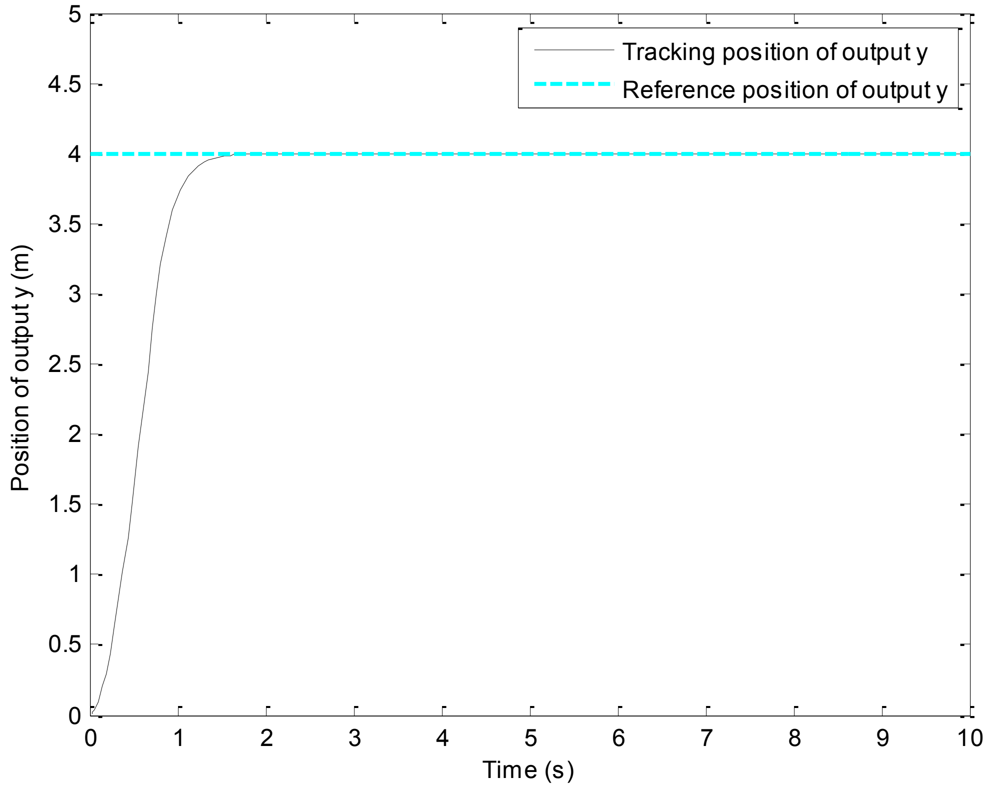

Fig. 2-6. Tracking of output y

For the same reference trajectory, using different sets of system coefficients do not affect the tracking performance, therefore the parameter uncertainties can be handled very well by sliding mode control.

2.3. Backstepping Control

Backstepping Control (BSC) is the third nonlinear control algorithm investigated in this research. As it is evident from the name, the algorithm is going backward through the process, starting from the furthest state and going back step-by-step to the control input. When the procedure reaches the control input, the overall controller of the system becomes available.

Since backstepping control is a direct Lyapunov-based method, which requires to find the appropriate Lyapunov candidate. Searching for the possible candidate will not only make sure that the needed control input can be correctly produced, but it will also show how the chosen Lyapunov function will determine the stability of the overall system.

A relatively simple system is considered as following [50]:

$$\begin{aligned} &\qquad + g(x_1)x_2 \\ &\qquad x_2) + g'(x_1, x_2)u \end{aligned} \tag{2-53}$$

To design a stable controller for the system, based on backstepping scheme, must be stabilized first, before . By assuming $x_2 = \phi(x_1)$ is a stable control input for $+ g(x_1)x_2$, following transformation can be made:

$$u = g'(x_1, x_2)^{-1}(u' - f'(x_1, x_2)) \tag{2-54}$$

By substituting equation (2-54) into equation (2-53), a simpler relation can be found.

$$\tag{2-55}$$

The equation (2-53) can then be simplified. Assuming a Lyapunov function $V_1(x_1)$ is satisfied by the inequality , and defining and rearranging the equation (2-53), the following can be easily shown [51]:

$$\begin{aligned} &\qquad + g(x_1)(\zeta + \phi(x_1)) = f(x_1) + g(x_1)\zeta + g(x_1)\phi(x_1) \\ &\dot{\zeta} \end{aligned} \tag{2-56}$$

where $\zeta = x_2 - \phi(x_1)$ and $x_2 = \zeta + \phi(x_1)$. From Fig. 2-7, the following equation can be derived [51].

$$\dot{\zeta} \qquad \frac{}{dx_1}(f(x_1)+g(x_1)x_2) = \frac{d\phi}{dx_1}(f(x_1)+g(x_1)\zeta+g(x_1)\phi(x_1)) \tag{2-57}$$

By assuming another Lyapunov function $V_2(x)$ as [51]:

$$V_2(x) = V_1(x_1) + \frac{1}{2}\zeta^2 \tag{2-58}$$

$$\dot{V} \tag{2-59}$$

and substituting equations (2-56), (2-57) and (2-58) into (2-59) to satisfy the inequality, two possibilities can occur.

Case 1: If $f'(x_1,x_2)=0, g'(x_1,x_2)=1$, the system control input can be represented as [50]:

$$u' = \frac{d\phi}{dx_1}(f(x_1)+g(x_1)x_2) - \frac{dV_1}{dx_1}g(x_1) - \alpha(x_2-\phi(x_1)) \tag{2-60}$$

Case 2: If $f'(x_1,x_2) \neq 0, g'(x_1,x_2) \neq 1$, the system control input can be simply attained [50].

$$u = g'(x_1,x_2)^{-1}\left(\frac{d\phi}{dx_1}(f(x_1)+g(x_1)x_2) - \frac{dV_1}{dx_1}g(x_1) - \alpha(x_2-\phi(x_1)) - f'(x_1,x_2)\right) \tag{2-61}$$

where the control gain is $\alpha > 0$.

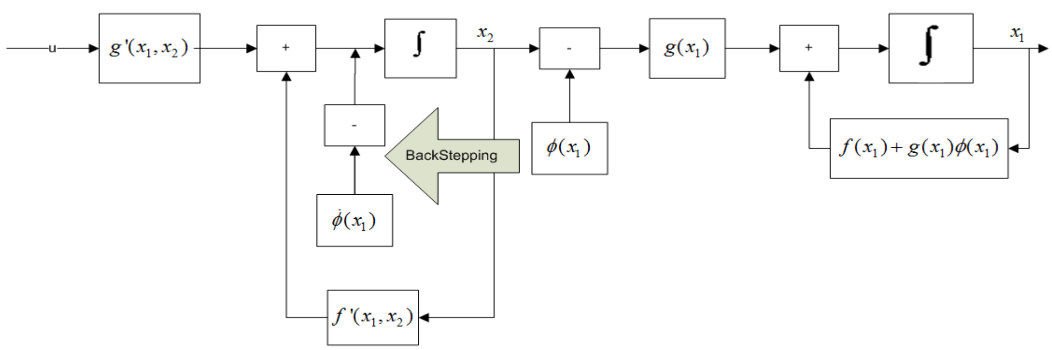

Fig. 2-7. Backstepping scheme

The procedure of designing a backstepping controller is further explored by the following process.

Consider the same system as in form (2-53). By first picking a Lyapunov candidate as $V_1(x_1) = \frac{1}{2}x_1^2$, and taking the first-order derivate on $V_1(x_1)$, it can be easily proven that

$$\dot{V} \quad \cdot \quad x_1) + g(x_1)x_2) = x_1(f(x_1) + g(x_1)\phi(x_1)) \tag{2-62}$$

To guarantee system is asymptotically stable, $\dot{V}$ needs to be a negative definite function. This can be achieved by

$$\phi(x_1) = g(x_1)^{-1}(-f(x_1) - \alpha_1 x_1) \tag{2-63}$$

where $\alpha_1 > 0$. By substituting equation (2-63) into (2-62), a negative definite Lyapunov function can be acquired, where $\phi(x_1)$ is the virtual control input for the first equation in the form as (2-52).

$$\begin{aligned} \dot{V} \quad \cdot \quad & x_1) + g(x_1)x_2) = x_1(f(x_1) + g(x_1)\phi(x_1)) \\ &= x_1(f(x_1) + g(x_1)g(x_1)^{-1}(-f(x_1) - \alpha_1 x_1))) = -\alpha_1 x_1^2 \le 0 \end{aligned} \tag{2-64}$$

Now the second equation with the actual control input needs to be stabilized. Again, another Lyapunov candidate is chosen as $V_2(x) = \frac{1}{2}x_1^2 + \frac{1}{2}z^2$, with $z = x_2 - \phi(x_1)$.

$$\begin{aligned} \dot{V} \quad \cdot \quad \cdot \\ &= x_1(f(x_1) + g(x_1)x_2) + z(f'(x_1, x_2) + g'(x_1, x_2)u - \dot{\varsigma} \\ &= x_1(f(x_1) + g(x_1)\phi(x_1) + g(x_1)z) + z(f'(x_1, x_2) + g'(x_1, x_2)u - \dot{\varsigma} \\ &= x_1(f(x_1) + g(x_1)\phi(x_1)) + z(f'(x_1, x_2) + g'(x_1, x_2)u - \dot{\varsigma} \quad (x_1)) \\ &= x_1(f(x_1) + g(x_1)\phi(x_1)) + z(f'(x_1, x_2) + g'(x_1, x_2)u - \frac{d\phi(x_1)}{dx_1} + \frac{dV_1(x_1)}{dx_1}g(x_1)) \end{aligned} \tag{2-65}$$

Similarly, $\dot{V}$ needs to be a negative definite function. Letting

$$u = g'(x_1,x_2)^{-1}(-f'(x_1,x_2) + \varsigma \qquad - \frac{dV_1(x_1)}{dx_1}g(x_1))$$
$$= g'(x_1,x_2)^{-1}(-f'(x_1,x_2) + \frac{d\phi(x_1)}{dx_1} - \alpha_2(x_2 - \phi(x_1)) - \frac{dV_1(x_1)}{dx_1}g(x_1))) \qquad (2\text{-}66)$$

where $\alpha_2 > 0$, a negative definite function can be deduced as:

$$\dot{V}$$
$$= x_1(f(x_1) + g(x_1)\phi(x_1)) + z(f'(x_1,x_2) + g'(x_1,x_2)u - \frac{d\phi(x_1)}{dx_1} + \frac{dV_1(x_1)}{dx_1}g(x_1))$$
$$= -\alpha_1 x_1^2 + z(f'(x_1,x_2) + [-f'(x_1,x_2) + \frac{d\phi(x_1)}{dx_1} - \alpha_2 z - \frac{dV_1(x_1)}{dx_1}g(x_1)] - \frac{d\phi(x_1)}{dx_1} + \frac{dV_1(x_1)}{dx_1} \qquad (2\text{-}67)$$
$$= -\alpha_1 x_1^2 - \alpha_2 z^2 \le 0$$

Therefore, the overall system control input u is determined by equation (2-66).

If $f'(x_1,x_2) = 0$, and $g'(x_1,x_2) = 1$, the input then becomes

$$u' = \frac{d\phi(x_1)}{dx_1} - \alpha_2 z - \frac{dV_1(x_1)}{dx_1}g(x_1) \qquad (2\text{-}68)$$

Similar to feedback linearization and sliding mode control, a simple example demonstrates more clearly in designing the above BSC.

Consider a system described as [52]:

$$y = x_1 \qquad (2\text{-}69)$$

To solve a tracking problem, the tracking error can be defined as following:

$$e_1 = x_{1d} - x_1 \qquad (2\text{-}70)$$

Now the problem becomes to stabilize e_1 instead of state x_1, a negative definite Lyapunov function needs to be defined as:

$$V_1(e_1) = \frac{1}{2}e_1^2 \qquad (2\text{-}71)$$

By taking the first-order derivative of $V_1(e_1)$, equation (2-72) can be acquired.

$$\dot{V} \quad)) \tag{2-72}$$

where $x_2 = \phi(x_1)$, and by letting

$$\phi(x_1) = \quad \alpha_1 e_1 \tag{2-73}$$

it can be proven that

$$\dot{V} \quad x_1)) = -\alpha_1 e_1^2 \leq 0 \tag{2-74}$$

Similarly, by defining another tracking error

$$\begin{aligned} e_2 &= x_2 - \phi(x_1) = x_2 - \quad \alpha_1 e_1 \\ &= \end{aligned} \tag{2-75}$$

$$\alpha_1 e_1 \tag{2-76}$$

$$V_2(e) = \frac{1}{2}e_1^2 + \frac{1}{2}e_2^2 \tag{2-77}$$

the derivative of $V_2(e)$ is

$$\begin{aligned} \dot{V} & \quad dx_1 \\ &= -e_1 e_2 - \alpha_1 e_1^2 + e_2\left(u - \frac{d\phi(x_1)}{dx_1}\right) \end{aligned} \tag{2-78}$$

Then, control input u can be chosen as:

$$u = e_1 + \frac{d\phi(x_1)}{dx_1} - \alpha_2 e_2 = e_1 + \tag{2-79}$$

Therefore, $\dot{V}$ can be found as:

$$\begin{aligned} \dot{V} & \quad -\alpha_1 e_1^2 + e_2\left(u - \frac{d\phi(x_1)}{dx_1}\right) \\ &= -\alpha_1 e_1^2 - \alpha_2 e_2^2 \leq 0 \end{aligned} \tag{2-80}$$

Equation (2-80) shows that $\dot{V}$ is negative definite.

Fig. 2-8 - Fig. 2-9 show the simulation results with the control gains $\alpha_1 = 40$ and $\alpha_2 = 20$. Similarly to feedback linearization and sliding mode control, all the initial conditions are set to 0, and tracking task requires to follow the reference $y = 4$.

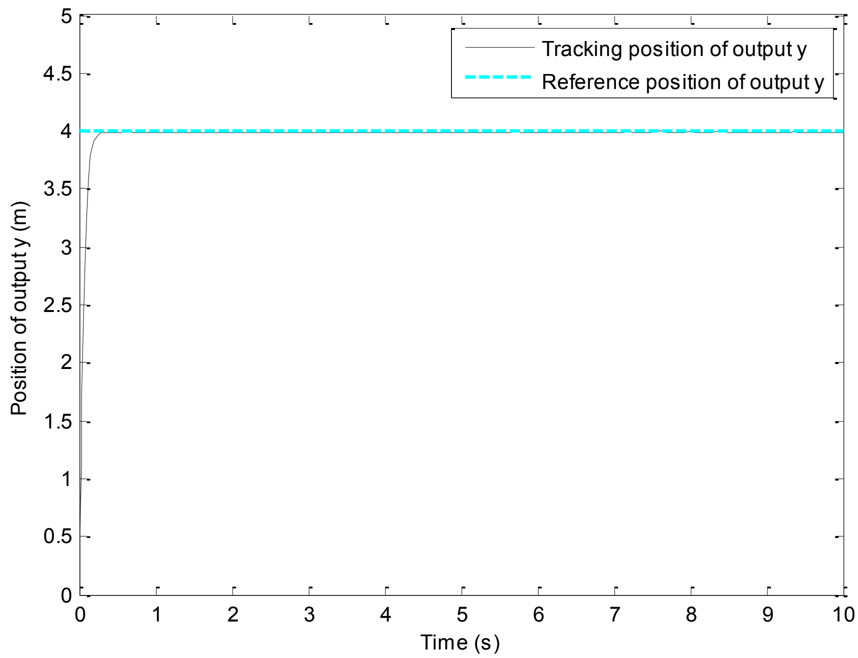

Fig. 2-8. Tracking of output y

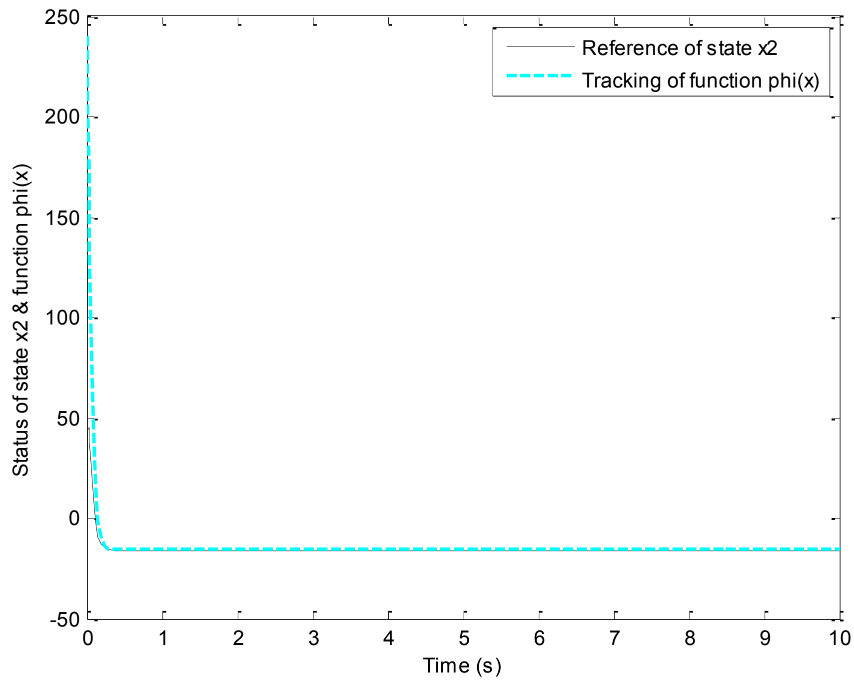

Fig. 2-9. Relation between state x_2 and function $\phi(x_1)$

2.4. Summary

The background material of three nonlinear control techniques including FLC, SMC, and BSC have been introduced and studied in this chapter. The following Qball-X4 flight control systems will be designed and implemented by the principle that has been set up in this chapter. The experimental testing results will show if the theoretical designs are appropriate or not.

3. Modelling and Identification of the Qball-X4 System

A mathematical model is always a crucial groundwork before any further control designing task. If an accurate model is available, then a controller can be designed as close as possible to the real application, and to handle the practical problems very well. If not, the controller will have to be designed based on some unknown dynamics or parameters, which may cause a major difference from theoretical simulations to practical implementation. This chapter introduces the mathematical modelling of Qball-X4 system and unknown parameters identification of the Qball-X4 system. By system analysis, a dynamic model can be derived, and by experiments, some unknown parameters can be attained to have the model as accurate as possible.

3.1. Experimental Setup

In this section, all the experimental equipments used for later parameter identification and real implementation are introduced.

The name of Quadrotor helicopter in the thesis is Qball-X4, because of the ball-shape protection cage surrounding the quadrotor. Four propellers are lined up orthogonally as shown in Fig. 3-1. The black box at the center is the control device that sends control signals to control the attitude of Qball-X4 during the flight, to generate different pulses to each rotor for pitch, roll, and yaw commands with the control algorithm implemented in software format in the on-board Gumstix single-chip micro-computer (control device).

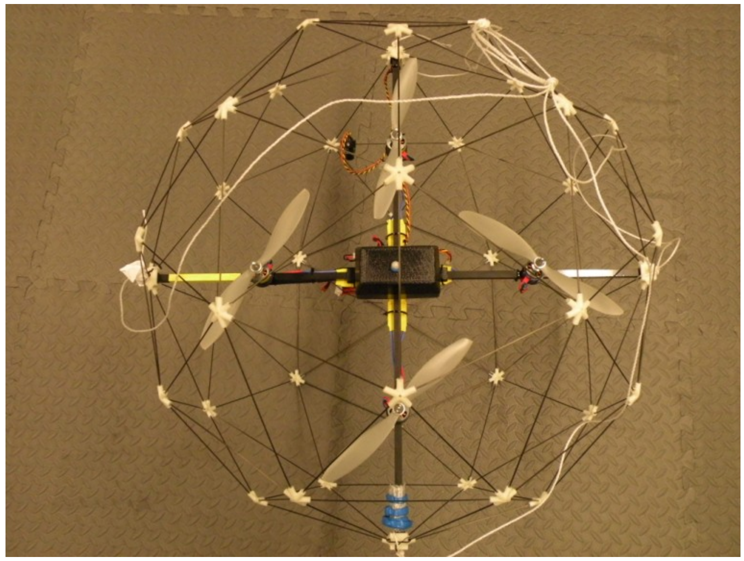

Fig. 3-1. The Qball-X4 structure

Inside the black box, there is a data acquisition board named HiQ and a Gumstix micro-computer. During the flight, all the data from sensors and ground station are collected through the HiQ board.

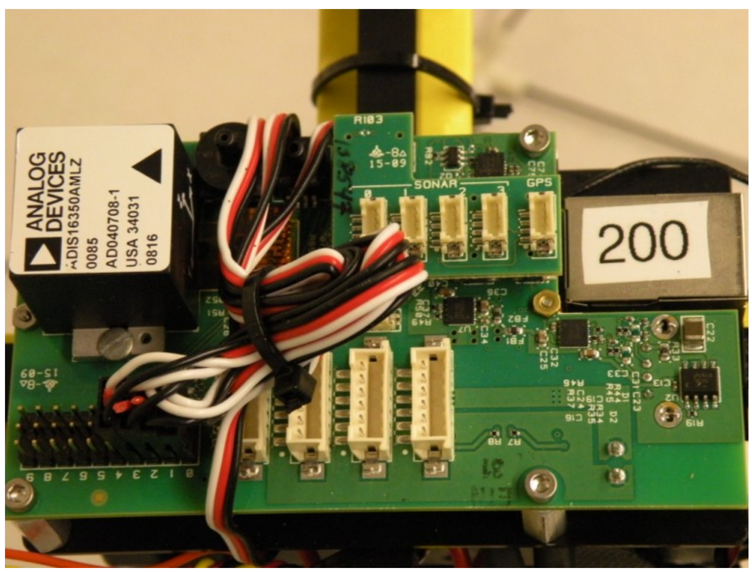

Fig. 3-2. The HiQ board with Gumstix and sensors

The Gumstix is a single-chip micro-computer which provides an embedded development platform. In Qball-X4 system, the Gumstix computer has a Linux operating system with a control software, QuaRC, installed, and is acting as a central processor that processes all the raw data collected from sensors and data received from the ground station. Once the data has been processed, it will be sent to drive the rotors. The communication between Qball-X4 and the ground station is established by wireless connection. As Fig. 3-2 shows, an analog device includes gyroscope, accelerometer, and magnetometer which can measure the angular velocity of *x,y,z* axes, acceleration of *x,y,z* and also the magnetic field. There is another sensor, sonar, available for height measurement.

The power source of the system is two 3-cell 2500mAH LiPo batteries, which can provide a continuous supply for about 15 minutes, and batteries are strapped at the bottom of the black box. The capacity of batteries can be measured from the HiQ board.

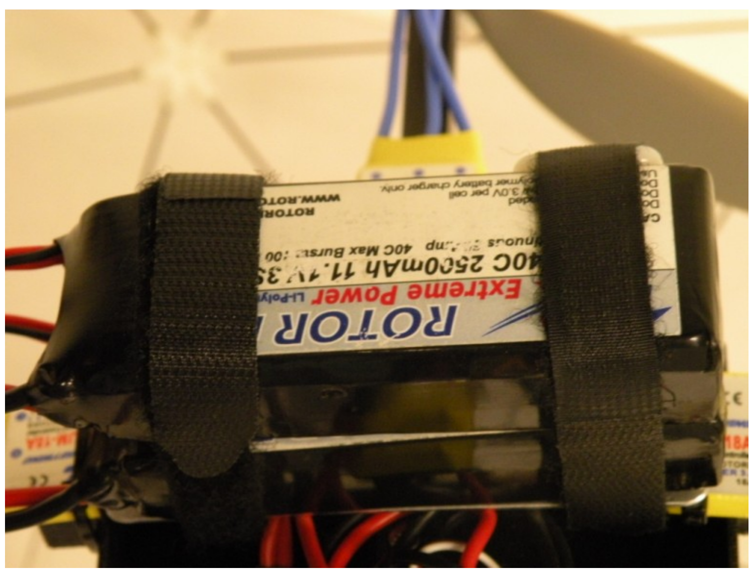

Fig. 3-3. Batteries and installation

For the system, not only are the inertial sensors on HiQ board used, but also vision sensors are in use. Hence, the location of Qball-X4 can be indicated by the feedback from a set of high-precision cameras as shown in Fig. 3-4. The direct global positions can be easily attained, and a direct position *x-y-z* control becomes possible as well. If necessary, the controller of Qball-X4 system can then be separated into two independent parts, attitude controller and altitude controller.

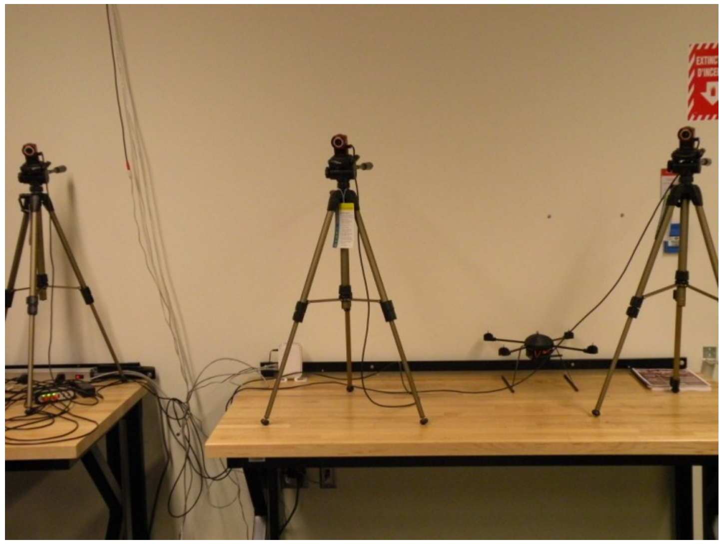

Fig. 3-4. Cameras for vision feedback

A joystick is used for safety reason, in case the Qball-X4 loses control during flight. The joystick can be used to cut Qball-X4 power by moving the left lever down to a zero position. This action will force Qball-X4 to land.

Fig. 3-5. The joystick

A single computer is used as the ground station. The control software installed is named QuaRC, the same as installed in Gumstix computer. QuaRC is a programming tool based on Matlab/Simulink, and is used as a main developing, designing, and implementing platform in this thesis.

The ground station computer has two separate programs designed in QuaRC. One program is the server which connects to joystick and cameras to receive the real-time feedback of safety signals and global coordinates. The other program is the client, which contains all the other feedbacks from sensors, and the main controllers of the system. Once the output of the server confirms that Qball-X4 is within range, the client can be started. First of all, the server starts to run to make sure Qball-X4 is within range. Secondly, the client starts to connect to the system for the sensors feedback, and readies the controllers. Thirdly, once the joystick is released from zero, all four rotors will be started by the commands given from the controllers. Qball-X4 will then start to follow the desired path. All the communications use TCP/IP (Transmission Control Protocol/Internet Protocol) protocol through wireless connections.

3.2. Dynamic Model

The groundwork of a controller designing process is always based on a mathematical model of the system to be controlled. In this thesis, a dynamic model is needed because forces generated by four propellers are the main move that the quadrotor flies and these propellers need to be controlled in appropriate ways for different flight modes and flight conditions. Fig.3-6 shows the attitude movements of the Qball-X4 [54].

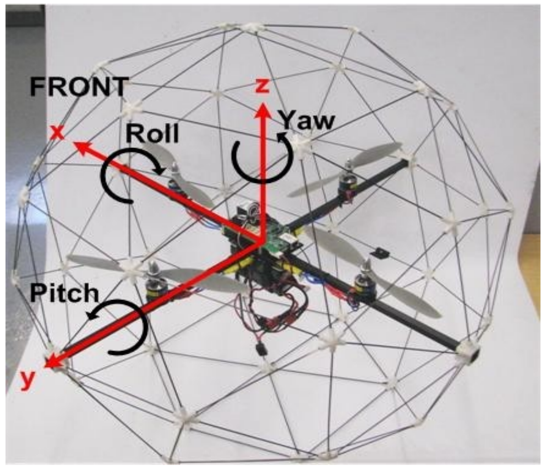

Fig.3-6. The Qball-X4 motions

Fig. 3-7 shows more clearly on the relation between movements and forces. Positive direction of pitch, roll and yaw angles have been presented as well.

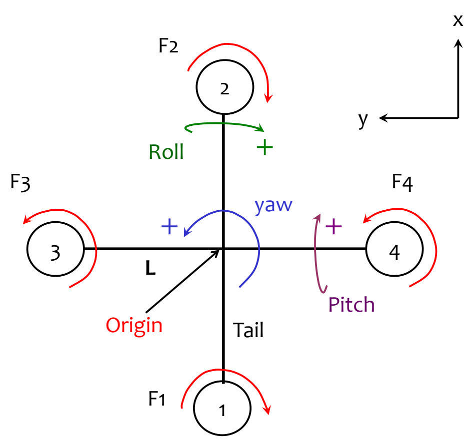

Fig. 3-7. Qball-X4 attitude definitions

Qball-X4 is a rigid body, and two sets of frames have been used to formulate the system dynamic equations. One frame is the body-fixed frame in which the origin is located at the center of the mass of Qball-X4 as shown in Fig. 3-7. The other frame is the earth-fixed frame, also known as global frame, in which the origin can be chosen as desired. The coordinates, x_q, y_q, z_q, are defined in body frame, and x, y, z are defined in earth frame.

Qball-X4 can be considered as a local frame rotating and translating in the global coordination. Euler rotation and translation matrix has been introduced here to generate the general transformation. In three dimensional axes *x, y, z*, there are three different rotation matrices [55]. The rotation matrix for *x* axis can be written as:

$$\boldsymbol{R}_x(\theta) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{bmatrix} \tag{3-1}$$

Similarly for y and z axes,

$$\boldsymbol{R}_y(\phi) = \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix} \tag{3-2}$$

$$\boldsymbol{R}_z(\psi) = \begin{bmatrix} \cos\psi & -\sin\psi & 0 \\ \sin\psi & \cos\psi & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{3-3}$$

where θ is the pitch angle along x axis, ϕ is the roll angle along y axis, and ψ is the yaw angle about z axis.

The general rotation matrix of all three axes can be written as:

$$\boldsymbol{R} = \boldsymbol{R}_z\boldsymbol{R}_y\boldsymbol{R}_x = \left[\begin{matrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\phi - \cos\phi\sin\psi & \cos\psi\sin\theta\cos\phi + \sin \\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\phi + \cos\phi\cos\psi & \sin\psi\sin\theta\cos\phi - \sin \\ -\sin\theta & \sin\phi\cos\theta & \cos\theta\cos\phi \end{matrix} \right. \tag{3-4}$$

The velocity transformation from earth frame to body frame is:

$$\left[\begin{matrix} \cdot \\ \cdot \\ \cdot \end{matrix} \quad \boldsymbol{R} \quad \begin{matrix} \cdot \\ \cdot \\ \cdot \end{matrix} \right. \tag{3-5}$$

where x, y, z and x_q, y_q, z_q are positions of earth frame and body frame respectively.

$$\sin\phi\sin\psi)\ddot{}$$
$$\sin\phi\cos\psi)\ddot{} \qquad (3\text{-}6)$$

Each rotor has a PWM input and the relation between input and output is described as [54]:

$$\boldsymbol{F} = K_a \frac{\omega}{s+\omega}\boldsymbol{W} \tag{3-7}$$

where $\boldsymbol{F}$ is the thrust vector generated by rotor, $\boldsymbol{W}$ is the PWM input vector, ω is the actuator angular velocity, and K_a is the gain.

In the body frame, all the four forces generated by four rotors are along z axis, which is in the form of the following.

$$\begin{bmatrix} F_{xq} \\ F_{yq} \\ F_{zq} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ F_1 + F_2 + F_3 + F_4 \end{bmatrix} \tag{3-8}$$

where F_i is the force generated by each rotor, and F_{iq} is the force along each axis. Using the rotation matrix (3-4), the forces in earth frame can be found as:

$$\begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = \boldsymbol{R} \begin{bmatrix} F_{xq} \\ F_{yq} \\ F_{zq} \end{bmatrix} \tag{3-9}$$

In the extension of the above equation, forces based on earth frame can be generalized.

$$\begin{aligned} \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} &= \begin{bmatrix} \cos\psi\cos\theta & \cos\psi\sin\theta\sin\phi - \cos\phi\sin\psi & \cos\psi\sin\theta\cos\phi + \sin\phi\sin\psi \\ \sin\psi\cos\theta & \sin\psi\sin\theta\sin\phi + \cos\phi\cos\psi & \sin\psi\sin\theta\cos\phi - \sin\phi\cos\psi \\ -\sin\theta & \sin\phi\cos\theta & \cos\theta\cos\phi \end{bmatrix} \begin{bmatrix} \cdot \\ \cdot \\ \cdot \end{bmatrix} \qquad (3\text{-}10) \\ &= \begin{bmatrix} \cos\psi\sin\theta\cos\phi + \sin\phi\sin\psi \\ \sin\psi\sin\theta\cos\phi - \sin\phi\cos\psi \\ \cos\theta\cos\phi \end{bmatrix} F_{zq} \end{aligned}$$

By Newton's second law for motion, $F = ma$, and taking friction factor f into consideration, the acceleration of each axis in earth frame can be extracted as $a = \frac{F - f}{m}$.

$$\begin{bmatrix} \ddot{} & F_x - f_x \\ \ddot{} & F_y - f_y \\ \ddot{} & F_z - G - f_z \end{bmatrix} = \frac{F_{zq}}{m} \begin{bmatrix} \cos\psi \sin\theta \cos\phi + \sin\phi \sin\psi \\ \sin\psi \sin\theta \cos\phi - \sin\phi \cos\psi \\ \cos\theta \cos\phi \end{bmatrix} - \frac{1}{m} \begin{bmatrix} f_x \\ f_y \\ G + f_z \end{bmatrix} \tag{3-11}$$

where m is the mass of Qball-X4, and $G = mg$ is the gravitational field. The drag forces f_x, f_y, and f_z are defined according to aerodynamics [55] as: $f_x = d_x \dot{}$, $f_y = d_y \dot{}$, $f_z = d_z \dot{}$,

Positions, velocities and accelerations are the altitude of Qball-X4, which are caused by the change of the attitude pitch, roll, and yaw angles. Attitude is determined directly from the force generated by each rotor. For instance, from Fig. 3-7, if forces F_1 and F_2 change, the torque of x axis in body frame will be changed by the difference $F_1 - F_2$, so as to the change of pitch angle, θ. Similarly, roll angle, ϕ will be changed by the difference $F_3 - F_4$, and yaw angle, ψ will be changed by $F_1 + F_2 - F_3 - F_4$.

Newton's second law for rotation is

$$\tau = F_\tau r = mr^2 \dot{\iota} \tag{3-12}$$

where τ is the torque, F_τ is the centripetal force, r is the length between the center and the desired point on the rigid body, and ω is the angular velocity. By definition of moment of inertia,

$$J = mr^2 \tag{3-13}$$

and the combination of translation and rotation motions of the desired point is

$$V_c = v_{cm} + \omega \times r \tag{3-14}$$

where v_{cm} is the linear velocity of the center of mass, and ω is the angular velocity. A translational momentum M of the Qball-X4 rigid body can be written as [56]:

$$\begin{aligned} \boldsymbol{M} &= \dot{\boldsymbol{H}} \quad \dot{\boldsymbol{H}}_{\boldsymbol{q}} \quad \boldsymbol{\omega_q} \quad \boldsymbol{H} \\ &= \boldsymbol{J_q}\dot{\boldsymbol{\omega}}_{\boldsymbol{q}} \quad \boldsymbol{\omega_q} \quad \boldsymbol{H} \end{aligned} \tag{3-15}$$

and in terms of $\boldsymbol{M}_q$ in body-fixed frame

$$\dot{\boldsymbol{\omega}}_{\boldsymbol{q}} \quad \boldsymbol{J}_{\boldsymbol{q}}^{-1} \; \boldsymbol{M}_q - \boldsymbol{\omega_q} \times \boldsymbol{H}_q) = \boldsymbol{J}_{\boldsymbol{q}}^{-1}[\boldsymbol{M}_q - \boldsymbol{\omega_q} \times (\boldsymbol{J_q}\boldsymbol{\omega_q})] \tag{3-16}$$

where $\boldsymbol{H}$ is the matrix of the angular momentums in earth-frame, $\boldsymbol{H}_q$ is the matrix of the same momentums in body-frame, $\boldsymbol{\omega_q}$ contains all the angular velocities of body-frame, and $\boldsymbol{J}_q$ is the inertia matrix about axes x_q, y_q, z_q of body-frame as [56]:

$$\boldsymbol{J_q} = \begin{bmatrix} J_{xx} & -J_{xy} & -J_{xz} \\ -J_{xy} & J_{yy} & -J_{yz} \\ -J_{xz} & -J_{yz} & J_{zz} \end{bmatrix} \tag{3-17}$$

$$\boldsymbol{J}_{\boldsymbol{q}}^{-1} = \frac{1}{|\boldsymbol{J_q}|}\begin{bmatrix} k_1 & k_2 & k_3 \\ k_2 & k_4 & k_5 \\ k_3 & k_5 & k_6 \end{bmatrix} \tag{3-18}$$

with

$$\begin{aligned} &k_1 = (J_{yy}J_{zz} - J_{yz}^2) \qquad k_2 = (J_{yz}J_{zx} + J_{xy}J_{zz}) \qquad k_3 = (J_{xy}J_{yz} + J_{zx}J_{yy}) \\ &k_4 = (J_{zz}J_{xx} - J_{zx}^2) \qquad k_5 = (J_{xy}J_{zx} + J_{yz}J_{xx}) \qquad k_6 = (J_{xx}J_{yy} - J_{xy}^2) \\ &|\boldsymbol{J_q}| = J_{xx}J_{yy}J_{zz} - 2J_{xy}J_{yz}J_{zx} - J_{xx}J_{yz}^2 - J_{yy}J_{zx}^2 - J_{zz}J_{xy}^2 \\ &\boldsymbol{\omega_q} = \begin{bmatrix} \dot{\theta} \\ \dot{\phi} \\ \dot{\psi} \end{bmatrix} \end{aligned} \tag{3-19}$$

Based on the principal axes theory, the inertia matrix can be reduced into a simple diagonal matrix.

$$\boldsymbol{J_q} = \begin{bmatrix} J_x & 0 & 0 \\ 0 & J_y & 0 \\ 0 & 0 & J_z \end{bmatrix}$$

$$\boldsymbol{J_q^{-1}} = \begin{bmatrix} \frac{1}{J_x} & 0 & 0 \\ 0 & \frac{1}{J_y} & 0 \\ 0 & 0 & \frac{1}{J_z} \end{bmatrix} \tag{3-20}$$

where J_x, J_y, J_z are inertias about x, y, z axes.

It is known that the torque on a body is equal to the rate of change of the same body's angular momentum. For Qball-X4 system, the body-fixed momentum $\boldsymbol{M_q}$ is defined by using length of lever L and c in x_q, y_q, z_q axes with body-fixed torque $\boldsymbol{\tau_q}$ as:

$$\boldsymbol{M_q} = \dot{\boldsymbol{H}}_q \quad \boldsymbol{\tau_q} \quad \begin{bmatrix} l(F_1 - F_2) \\ l(F_3 - F_4) \\ c(F_1 + F_2 - F_3 - F_4) \end{bmatrix} \tag{3-21}$$

Thus, from equation (3-16), the following can be obtained:

$$\begin{bmatrix} J_x\ddot{\ } & \dot{\ }\ \dot{\ } & l(F_1 - F_2) \\ J_y\ddot{\ } & \dot{\ }\ \dot{\ } & l(F_3 - F_4) \\ J_z\ddot{\ } & (J_x - J_y)\dot{\ }\ \dot{\ } & F_1 + F_2 - F_3 - F_4) \end{bmatrix} \tag{3-22}$$

Due to the gyroscopic effects [57] on four rotors, two more terms need to be added into equation (3-22) as $-J_r\dot{\ }$, $J_r\dot{\ }$ respectively for $\dot{\ }$ and $\dot{\ }$, with Ω defined as a disturbance $\Omega = \Omega_1 + \Omega_2 - \Omega_3 - \Omega_4$. The angular velocity for each rotor is Ω_i and J_r is the moment of inertia of each rotor.

Including drag forces as frictions, the Qball-X4 attitude dynamics is then written as:

$$\begin{bmatrix} J_x \ddot{} \\ J_y \ddot{} \\ J_z \ddot{} \end{bmatrix} \quad (J_x - J_y)\dot{}\dot{} \quad F_1 + F_2 - F_3 - F_4) \quad \begin{bmatrix} \\ \\ 0 \end{bmatrix} \quad \begin{bmatrix} \\ \\ f_\psi \end{bmatrix} \tag{3-23}$$

where $f_\theta = d_\theta \dot{\theta}$, $f_\phi = d_\phi \dot{\phi}$, and $f_\psi = d_\psi \dot{\psi}$ are drag forces with d_i as the drag coefficient for both altitude and attitude.

The overall system is described by combining equations (3-11) and (3-23) as follows:

$$\begin{aligned} &\ddot{} \quad \frac{}{m} \\ &\ddot{} \quad \frac{}{m} \\ &\ddot{} \quad \frac{}{m} \\ &\ddot{} \quad \frac{}{J_x} \quad \frac{}{J_x} \quad \frac{}{J_x} \\ &\ddot{} \quad \frac{-J_x)\dot{}\dot{}}{J_y} \quad \frac{F_4)}{J_y} + \frac{J_r \dot{\theta} \Omega}{J_y} - d_\phi \dot{} \\ &\ddot{} \quad \frac{}{J_z} \quad \frac{}{J_z} \end{aligned} \tag{3-24}$$

3.3. Parameter Identification

In the previous section, a mathematical model has been discussed and developed. However, how accurate the system parameters are needs to be determined. Table 3-1 and Table 3-2 have listed all the theoretical parameters.

Table 3-1 Inertia parameters

Parameter	J_x	J_y	J_z
Value	0.03(kg. m^2)	0.03(kg. m^2)	0.04(kg. m^2)

Table 3-2 System parameters

Parameter	K_a	ω	m	L
Value	120(N)	15(rad/sec)	1.422(kg)	0.2(m)

In practise, due to the error of measurements and noise, theoretical parameters sometimes are different from those in real application. Hence, with a set of inaccurate parameters, the designed controllers might only work in simulations. In order to have a better controller and a better experimental result, parameter identification procedure is then used. Through a series of experiments, by knowing the inputs and the outputs, parameters can be identified, and the Qball-X4 model equations from theories can also be verified if it is applicable in practise.

Before identifying all the needed parameters, the model equations have to be changed into the following format:

[illegible] (3-25)

All the angular velocities of [illegible], [illegible], and [illegible] can be given from the sensors measurements, and the inputs of $(F_1 - F_2)$, $(F_3 - F_4)$, and $(F_1 + F_2 - F_3 - F_4)$ are the output forces from each actuator. By actuator dynamics equation (3-7), it can be known the force is controlled by PWM wave, which is preset to the desired values. In expansion, the above equation (3-25) can be rewritten in more details as:

$$\ddot{} \qquad \frac{}{s+\omega} \qquad \frac{}{s+\omega}$$

$$\ddot{} \qquad \frac{}{s+\omega} \qquad \frac{}{s+\omega} \tag{3-26}$$

$$\ddot{} \qquad \frac{}{s+\omega} \qquad \frac{}{s+\omega} \qquad \frac{}{s+\omega} \qquad \frac{}{s+\omega}$$

so that

$$\ddot{} \qquad \qquad \tag{3-27}$$

where $p_a = K_a \dfrac{\omega}{s+\omega}$ is the actuator coefficient, and a new set of parameters has been chosen as $p_2 = p_2 p_a$, $p_6 = p_6 p_a$, $p_{10} = p_{10} p_a$

The angular accelerations $\ddot{}$, $\ddot{}$, and $\ddot{}$ are calculated by the definition of derivative

$$\ddot{} = \frac{}{dt} \tag{3-28}$$

where dt is a small number to provide enough precision.

For Qball-X4 operating at a low speed, the following sections ignore all the drag forces.

3.3.1. Pitch Identification

For pitch angle, once the input $(F_1 - F_2)$, and output $\ddot{}$ are known, the equation from (3-25) can be rewritten as:

$$\begin{bmatrix}\ddot{\theta}\end{bmatrix} \quad p_2 \quad p_3\Big]\begin{bmatrix}\dot{\varsigma}\\ F_1 - F_2 \\ -\dot{\varsigma}\end{bmatrix} \tag{3-29}$$

$$\begin{bmatrix}p_1 & p_2 & p_3\end{bmatrix} = \begin{bmatrix}\ddot{\theta}\end{bmatrix}\begin{bmatrix}\dot{\varsigma}\\ F_2 \\ -\dot{\varsigma}\end{bmatrix} \tag{3-30}$$

Each experiment has thousands of values for each parameter, thus pseudo inverse approach is needed to calculate the inverse of non-square matrices.

$$\begin{bmatrix}p_{11} & p_{12} & p_{13}\\ \vdots & \ddots & \vdots \\ p_{i1} & p_{i2} & p_{i3}\end{bmatrix} \quad \boldsymbol{\theta}_n \begin{bmatrix}\dot{\varsigma}\\ -F_2)_n \\ -\dot{\varsigma}\end{bmatrix} \tag{3-31}$$

where $n = 1, \ldots$.

The input signals have been given to maximize the changes of the output, so that a close enough approximation of the practical model can be achieved. The range of input is PWM waveform from 0.055 to 0.1 to drive the motor to rotate. In this thesis, the result of $F_1 - F_2$ is the input of the actuators, which has been set to square wave with a magnitude from -0.02 to 0.02 for the first initial condition (IC1). The second initial condition (IC2) and the third initial condition (IC3) have been set from -0.01 to 0.01 and -0.015 to 0.015 respectively as shown in the following tables.

Table 3-3 Estimated parameters of attitude pitch for IC1

Set No.	IC1		
Times	Parameters		
	p_1	p_2	p_3
1	-3.0699	3.1029	2.8679
2	-0.3252	2.4674	-4.7796
3	0.1762	4.3268	-7.2080
4	0.1482	6.6427	-8.9897

Average error bounds	-0.7677 ± 2.3022	4.1349 ± 2.5078	-4.5274 ± 4.4623

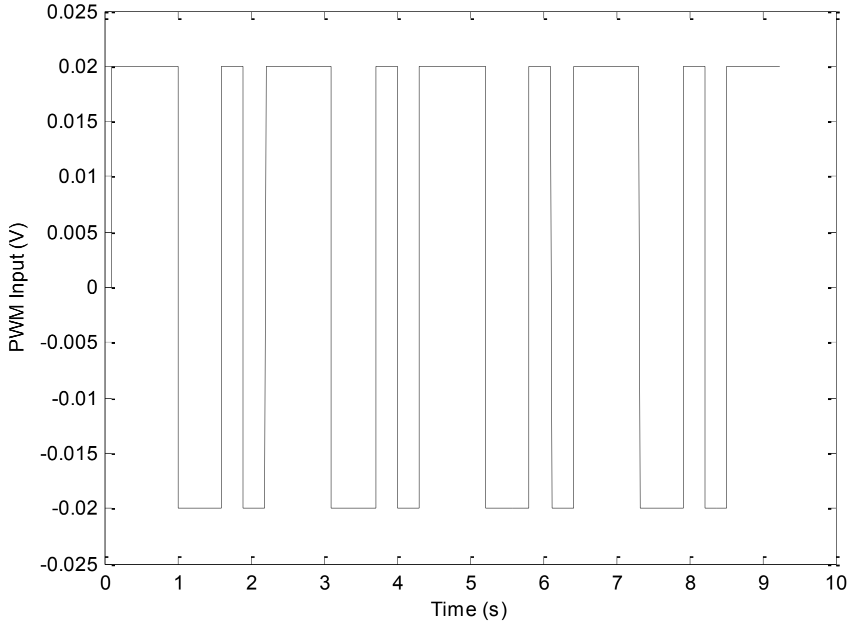

Fig. 3-8. PWM input for pitch of IC1

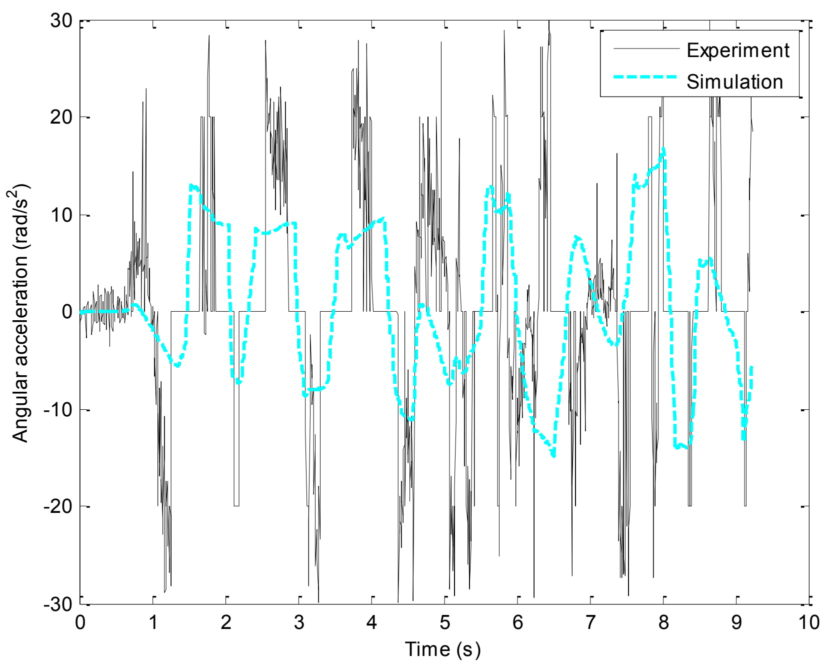

Fig. 3-9. Result of initial condition 1

Table 3-4 Estimated parameters of attitude pitch for IC2

Set No.	IC2		
Times	Parameters		
	p_1	p_2	p_3
1	-0.0531	-0.2917	-4.1946
2	-1.6056	5.4627	2.7320
3	1.4168	1.7824	-1.6919
4	-0.3608	4.2282	-5.0817
Average	-0.1507	2.7954	-2.0591
error bounds	$\pm$ 1.5675	$\pm$ 2.6673	$\pm$ 4.7911

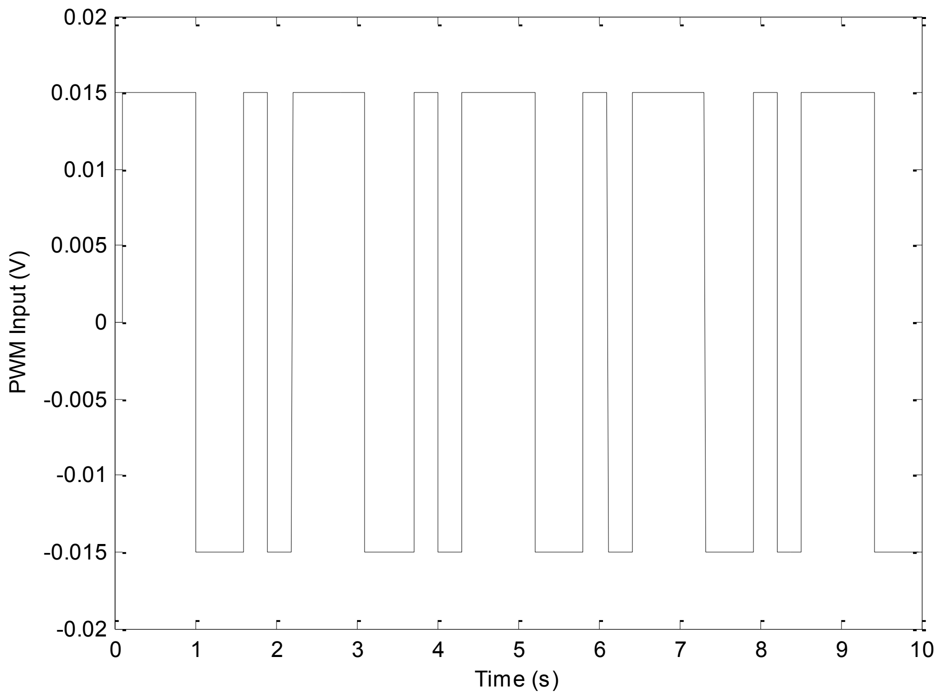

Fig. 3-10. PWM input for pitch of IC2

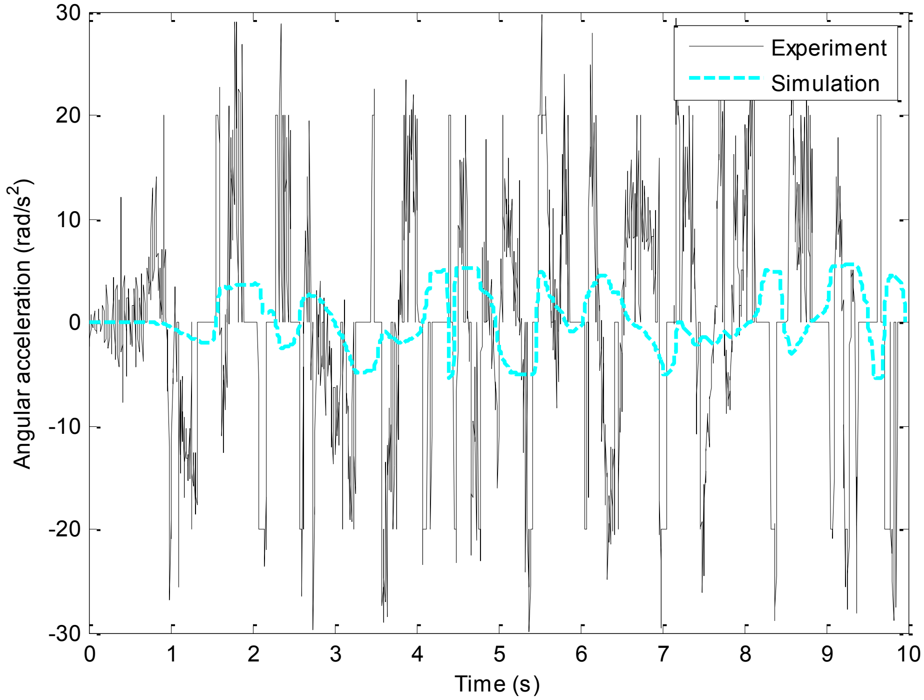

Fig. 3-11. Result of initial condition 2

Table 3-5 Estimated parameters of attitude pitch for IC3

Set No.	IC3		
Times	Parameters		
	p_1	p_2	p_3
1	-0.7962	-2.1213	-4.8125
2	0.1648	4.0321	-3.0568
3	1.0136	0.5105	-0.7622
4	-0.7999	2.2020	-1.2323
Average	-0.1044	1.1558	-2.4659
error bounds	$\pm$ 1.1180	$\pm$ 3.2771	$\pm$ 2.3466

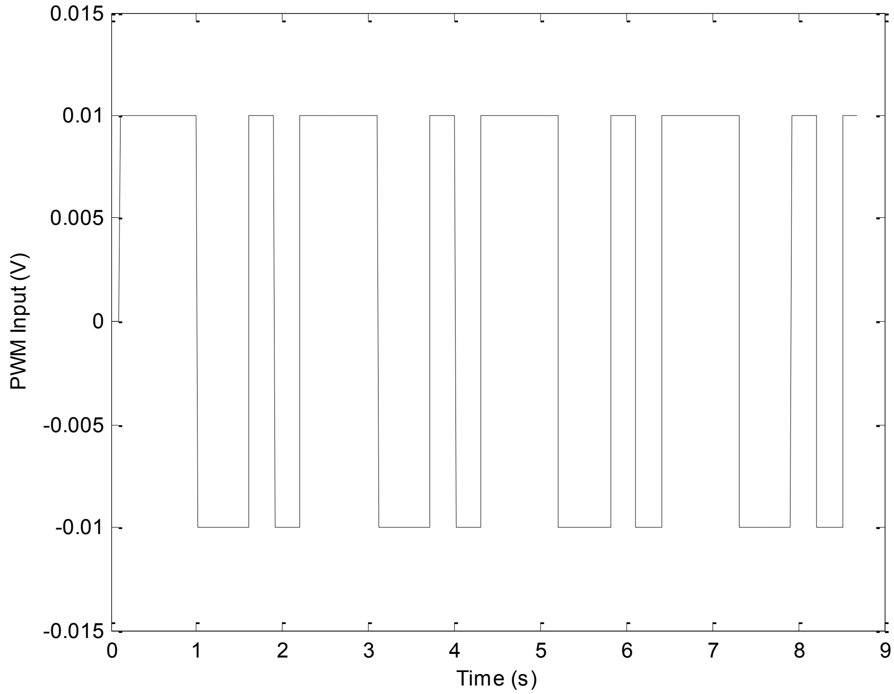

Fig. 3-12. PWM input for pitch of IC3

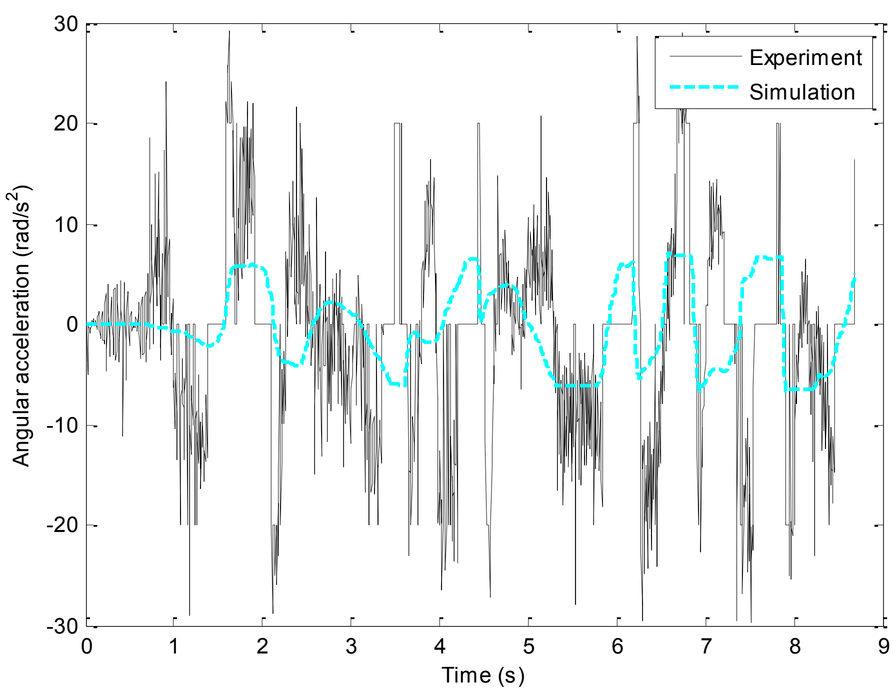

Fig. 3-13. Result of initial condition 3

3.3.2. Roll Identification

Similarly, for roll angle, the equations are written as:

$$\begin{bmatrix}\ddot{\phi}\end{bmatrix} \quad p_6 \quad p_7 \end{bmatrix}\begin{bmatrix} \dot{\iota} \\ F_3 - F_4 \\ \dot{\iota} \end{bmatrix} \tag{3-32}$$

$$\begin{bmatrix} p_{15} & p_{16} & p_{17} \\ \vdots & \ddots & \vdots \\ p_{i5} & p_{i6} & p_{i7} \end{bmatrix} \quad \phi_n \quad \begin{bmatrix} \dot{\iota} \\ -F_4)_n \\ \dot{\iota} \end{bmatrix} \tag{3-33}$$

where $n = 1, \ldots$.

The inputs of roll identification are changed to $F_3 - F_4$, with a square wave magnitudes from -0.02 to 0.02 for IC1, from -0.01 to 0.01 for IC2 and from -0.015 to 0.015 for IC3 as the same as in previous section.

Table 3-6 Estimated parameters of attitude roll for IC1

Set No.	IC1		
Times	Parameters		
	p_5	p_6	p_7
1	0.4846	1.0983	1.5755
2	-0.5849	2.2949	-1.0558
3	0.3840	1.0421	7.4659
4	0.6892	2.0951	0.2552
Average	0.2432	1.6326	2.0602
error bounds	± 0.8281	± 0.6623	± 5.4057

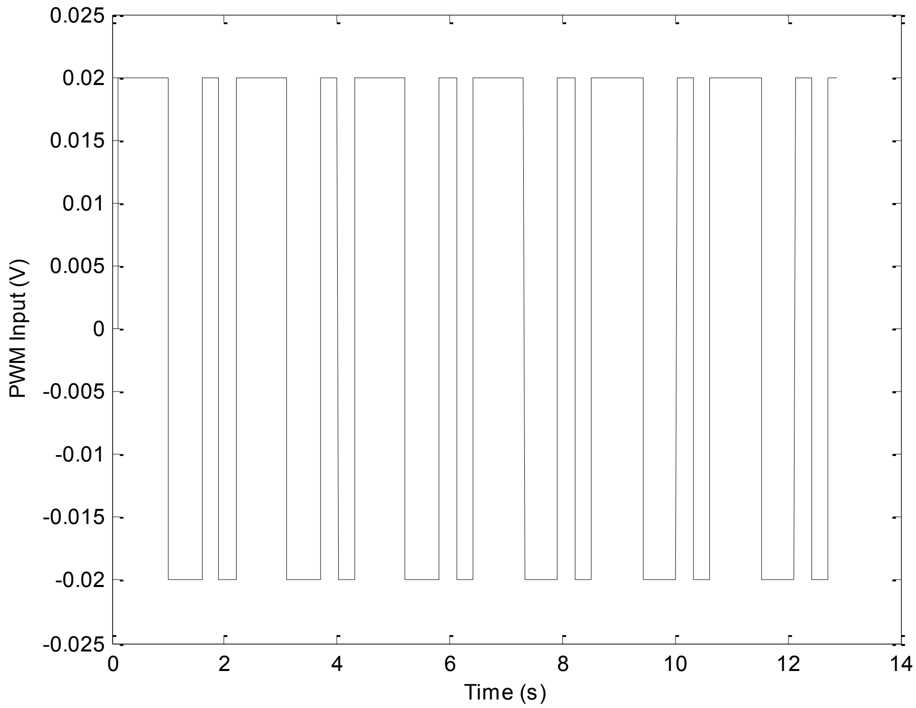

Fig. 3-14. PWM input for roll of IC1

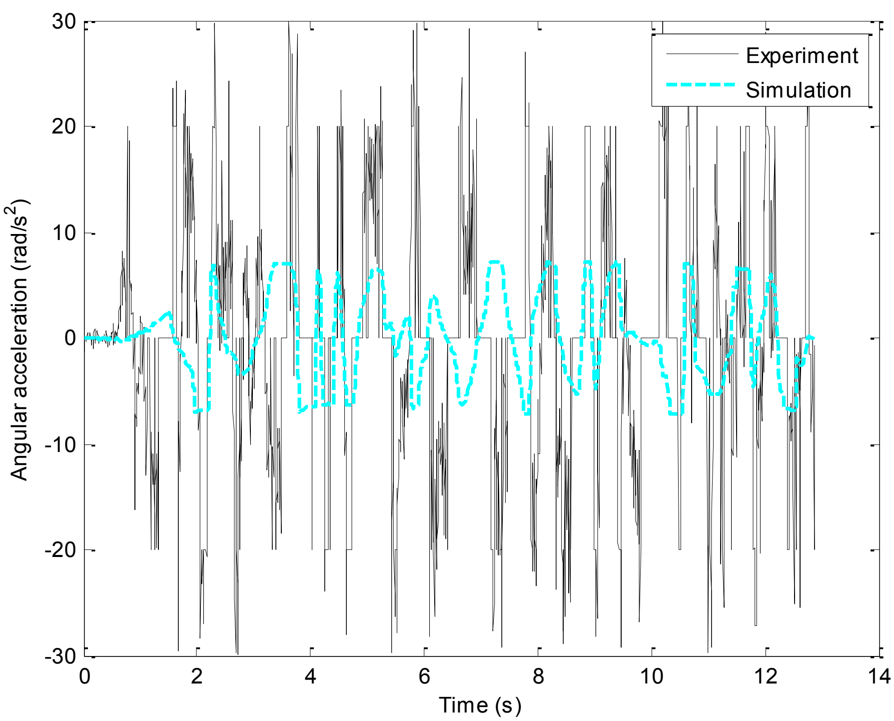

Fig. 3-15. Result of initial condition 1

Table 3-7 Estimated parameters of attitude roll for IC2

Set No.	IC2		
Times	Parameters		
	p_5	p_6	p_7
1	-1.2261	1.3172	-1.1282
2	-1.4324	5.4702	-0.5925
3	-0.1349	2.3934	-0.9432
4	0.9382	1.6203	-0.6028
Average	-0.4638	2.7003	-0.8167
error bounds	$\pm$ 1.4020	$\pm$ 2.7699	$\pm$ 0.3115

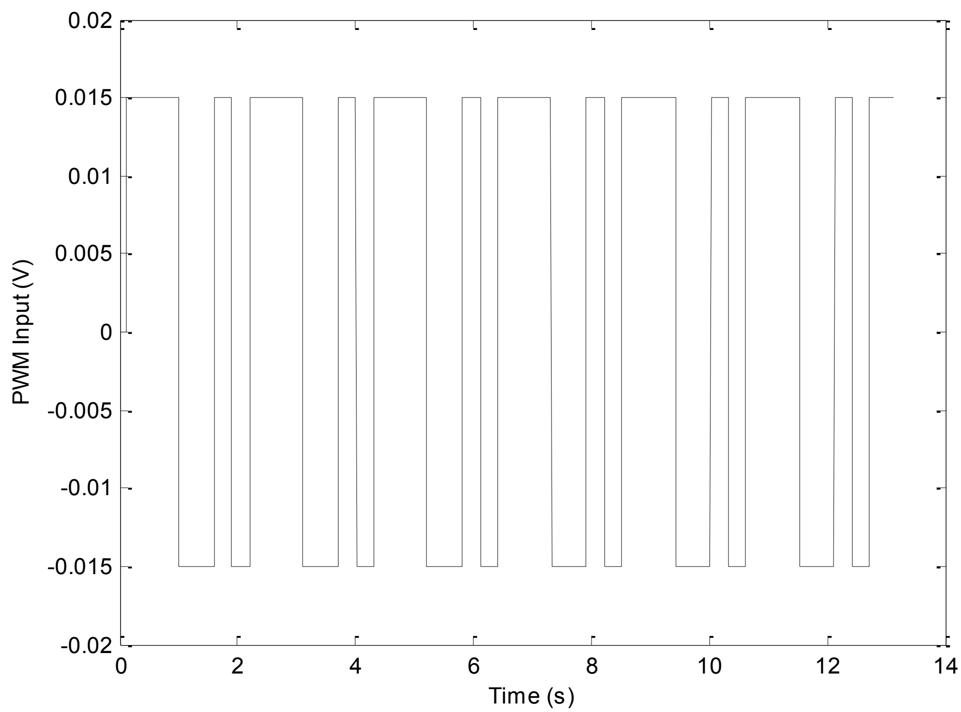

Fig. 3-16. PWM input for roll of IC2

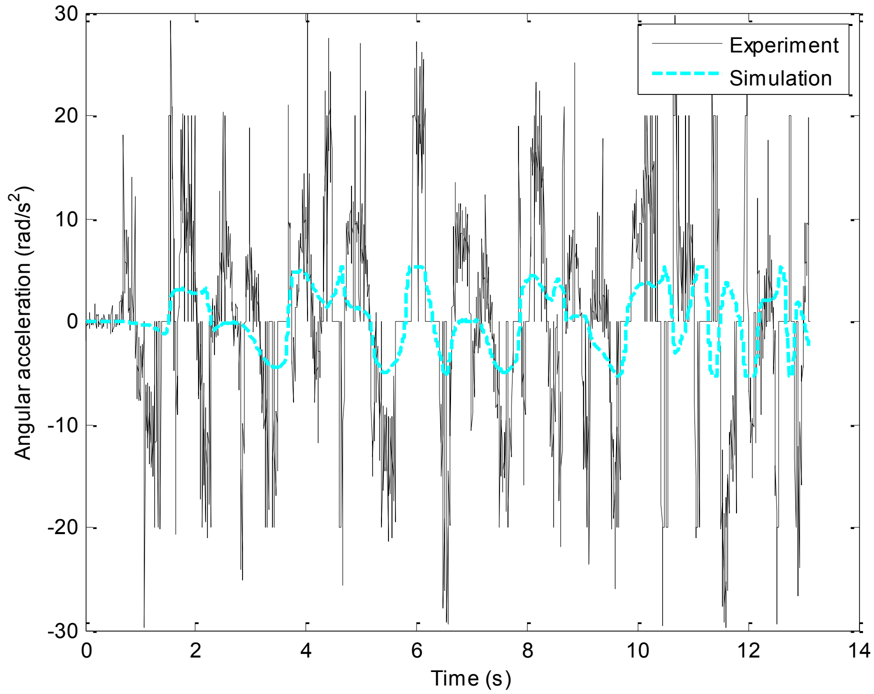

Fig. 3-17. Result of initial condition 2

Table 3-8 Estimated parameters of attitude roll for IC3

Set No.	IC3		
Times	Parameters		
	p_5	p_6	p_7
1	-0.7034	2.2475	-1.4136
2	1.8087	2.3508	0.5491
3	0.9004	2.2209	-0.3518
4	2.7565	0.0436	0.6769
Average	1.1906	1.7157	-0.1349
error bounds	± 1.8940	± 1.6721	± 1.2787

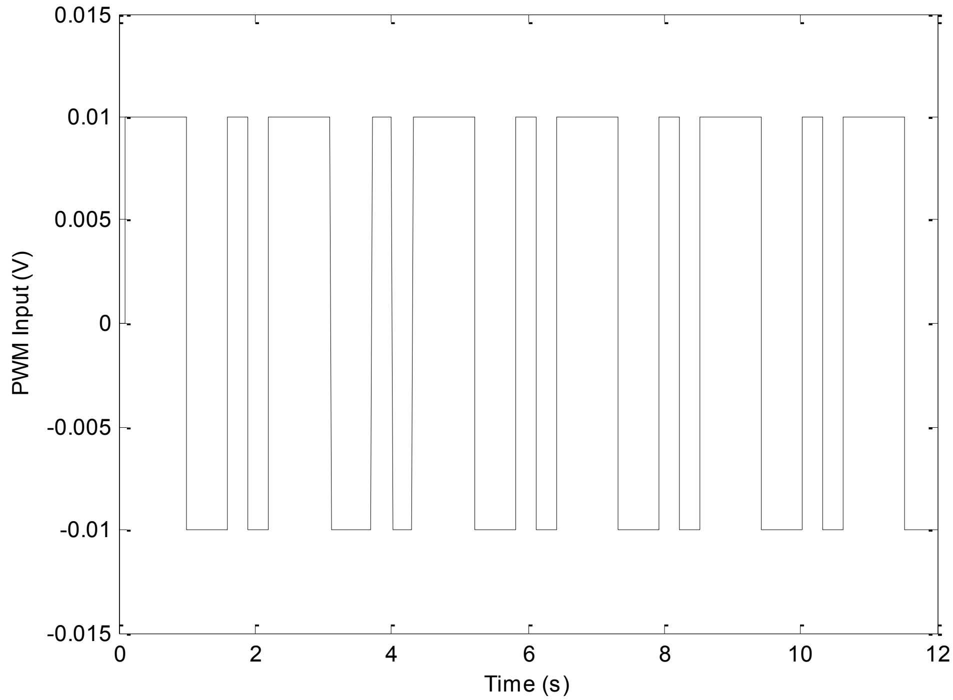

Fig. 3-18. PWM input for roll of IC3

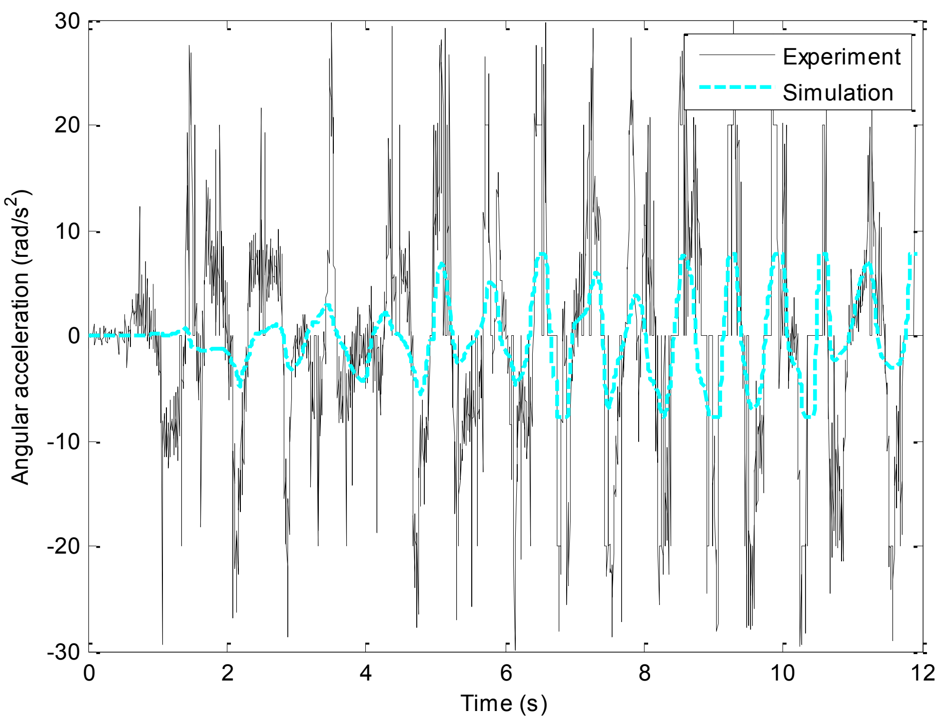

Fig. 3-19. Result of initial condition 3

3.3.3. Yaw Identification

For yaw angle, the equations are as follows:

$$[\ddot{\psi} \qquad p_{10}]\begin{bmatrix} \dot{\iota} \\ F_1+F_2-F_3-F_4 \end{bmatrix} \tag{3-34}$$

$$\begin{bmatrix} p_{19} & p_{110} \\ \vdots & \vdots \\ p_{i9} & p_{i10} \end{bmatrix} \qquad \psi_n \begin{bmatrix} \dot{\iota} \\ (F_1+F_2-F_3-F_4)_n \end{bmatrix}^{-1} \tag{3-35}$$

The input of yaw is different from the previous two sections in the form of $F_1+F_2-F_3-F_4$. The parameters can be identified through four sets of experiments listed below.

Table 3-9 Estimated parameters of attitude yaw for IC1

Set No.	IC1	
Times	Parameters	
	p_9	p_{10}
1	-0.3036	37.3767
2	7.7218	44.9528
3	10.7151	41.5903
4	10.0678	44.6962
Average	7.0503	42.1540
error bounds	$\pm$ 7.3539	$\pm$ 4.7773

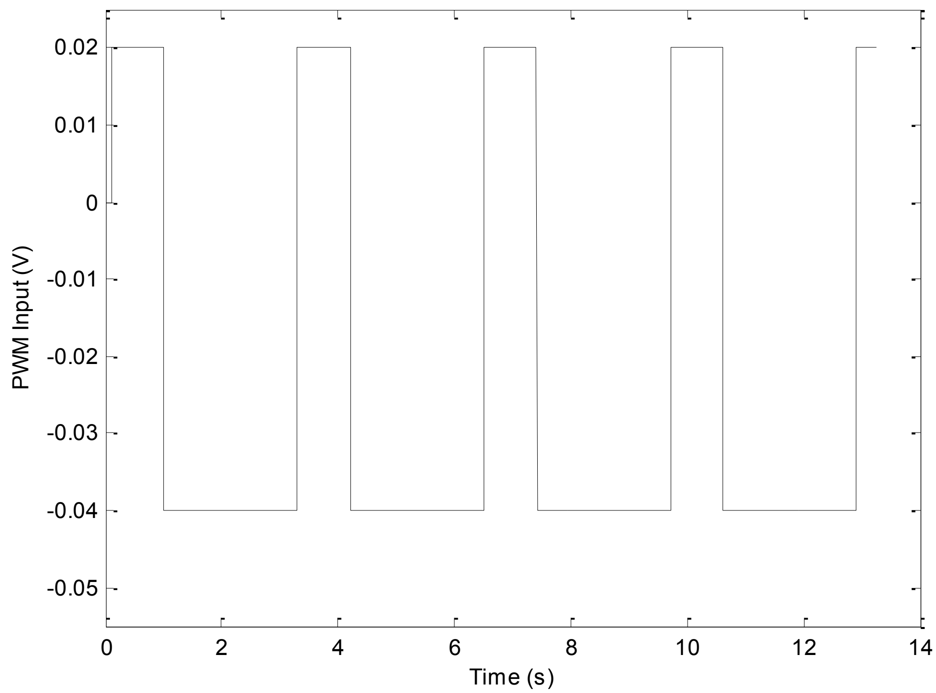

Fig. 3-20. PWM input for yaw of IC1

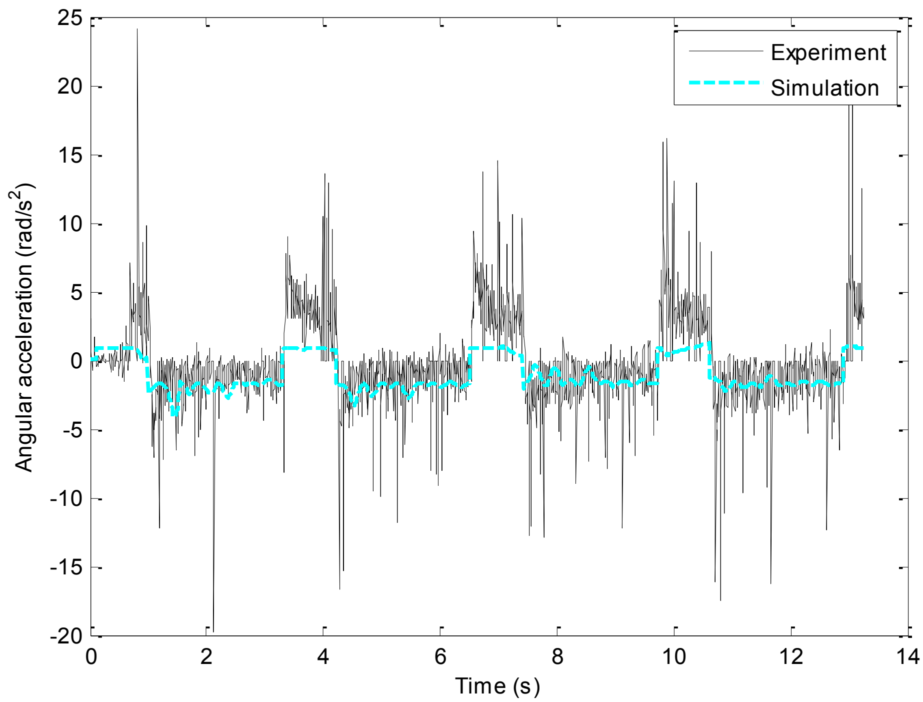

Fig. 3-21. Result of initial condition 1

Table 3-10 Estimated parameters of attitude yaw for IC2

Set No.	IC2	
Times	Parameters	
	p_9	p_{10}
1	-8.2324	51.2026
2	5.6769	54.8508
3	7.7716	74.1427
4	2.3841	81.0267
Average	1.9001	65.3056
error bounds	± 10.1325	± 15.7211

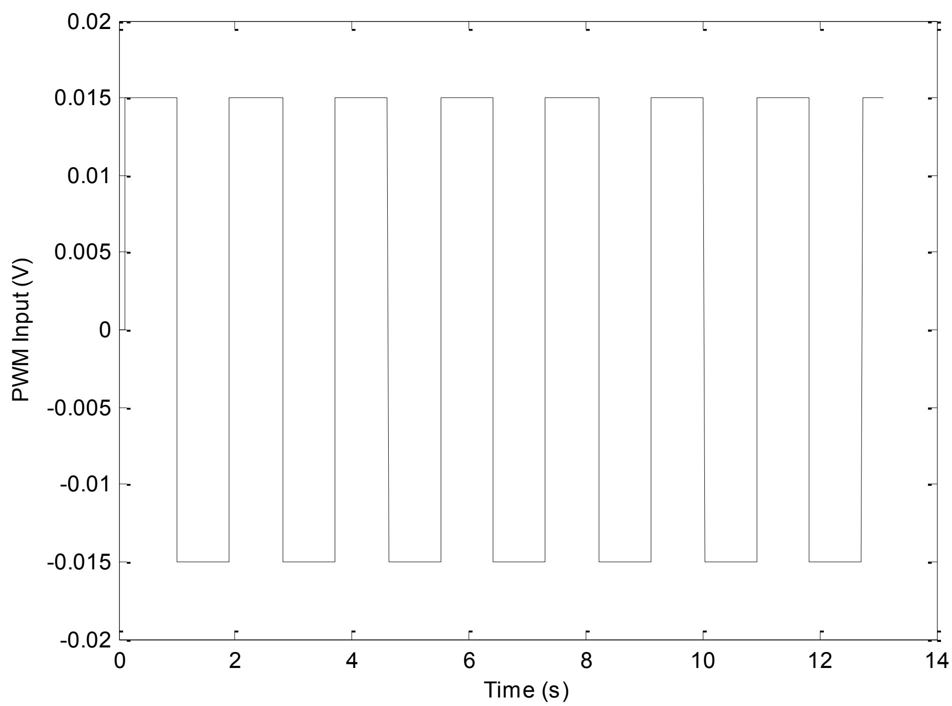

Fig. 3-22. PWM input for yaw of IC2

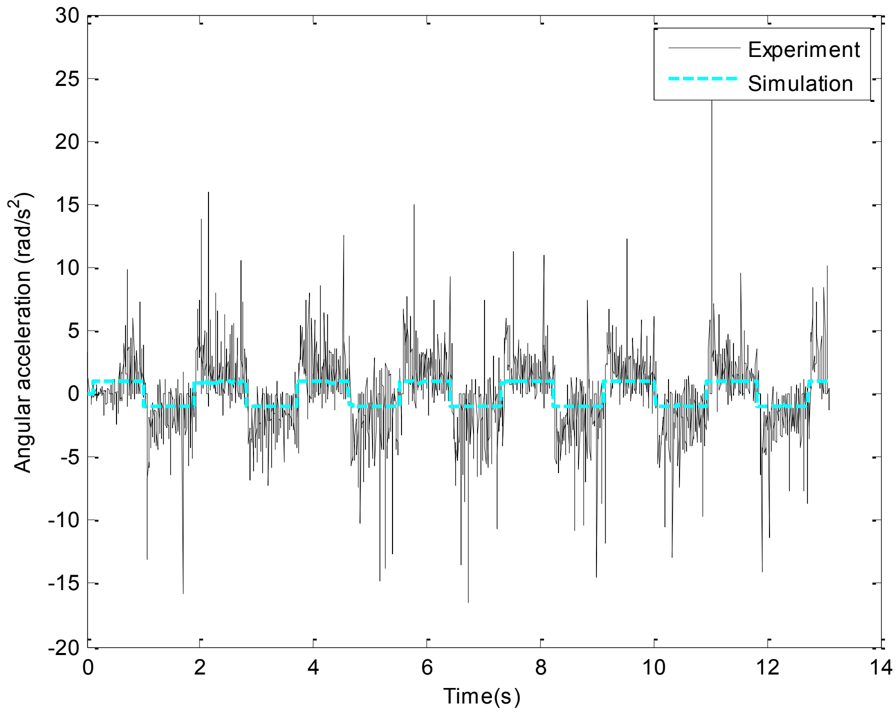

Fig. 3-23. Result of initial condition 2

Table 3-11 Estimated parameters of attitude yaw for IC3

Set No.	IC3	
Times	Parameters	
	p_9	p_{10}
1	1.2428	91.1459
2	4.0248	71.9005
3	6.4178	75.6902
4	10.4587	79.1548
Average	5.5360	79.4728
error bounds	$\pm$ 4.9227	$\pm$ 11.6731

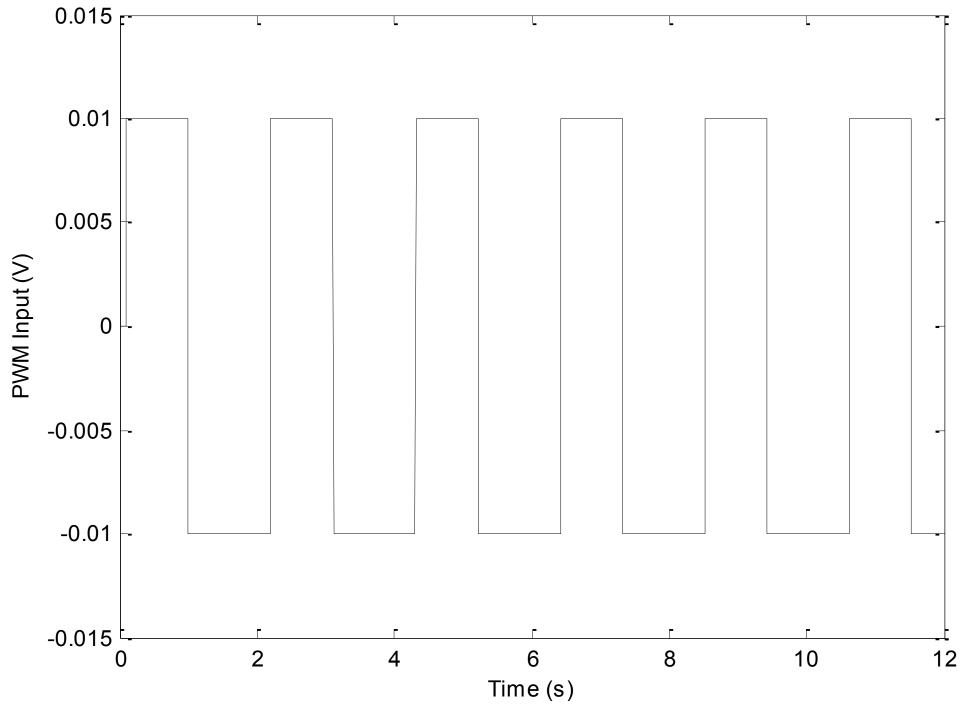

Fig. 3-24. PWM input for yaw of IC3

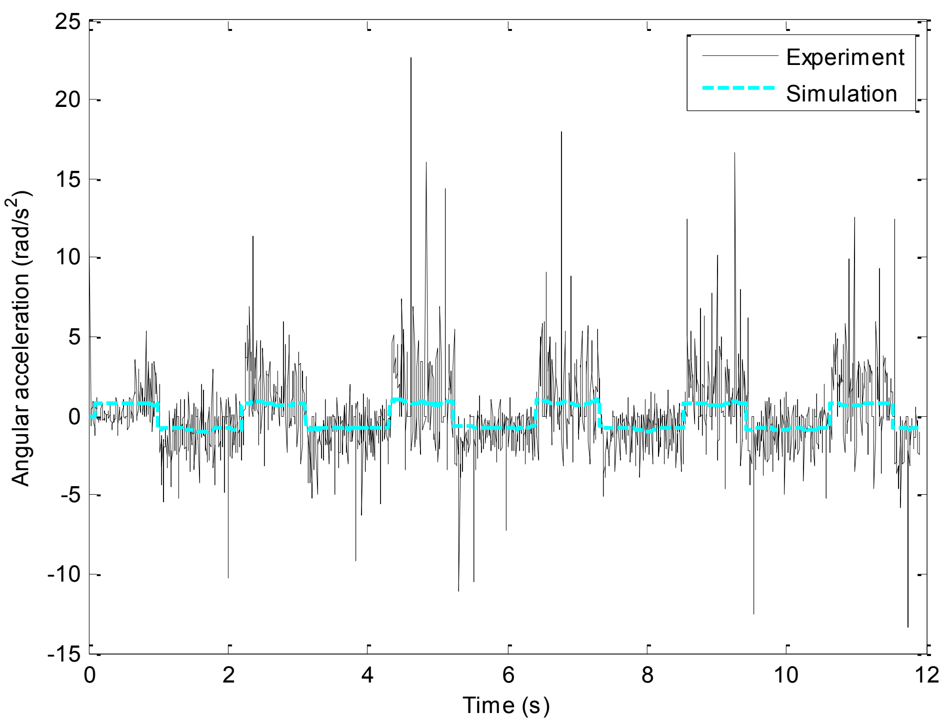

Fig. 3-25. Result of initial condition 3

3.4. Summary

The Qball-X4 dynamics have been derived and identified. The practical controller and the future experimental flight tests will be designed and conducted based on the model equations and system parameters developed in this chapter. Since the thesis targets to do experimental test on the real Qball-X4 UAV test-bed available at the Networked Autonomous Vehicles (NAV) Lab of Concordia University, it is very crucial to have correct and precise mathematical model of the system. If the model dynamics and system parameters are close enough to the reality, the better testing results can be obtained. Therefore, this chapter has laid the ground work for the later experimental flight tests.

4. Nonlinear Control of the Qball-X4 System

In Chapter 2, all the background theories and procedures needed for the controller design of the Qball-X4 system have been explained and illustrated in details. In this chapter, practical design and implementation will be carried out according to what have been discussed before for feedback linearization control, sliding mode control and backstepping control, respectively. As verification, simulations will not be the only approach, and experimental flight tests on the Qball-X4 system will be another strong proof and comparison of the performance of the designed controllers.

From Chapter 3, the system parameters are identified and the theoretical model dynamics have been proven effective. However, due to the limitations of experimental equipment, the identified parameters only have the overall system information. For instance, instead of identifying J_x, J_y, and J_z respectively, the identification procedure can only calculate the parameter p_i as the combination of the individual inertia. Therefore, the identified parameter p_i is not used directly in the design process, but used in the practical implementation, especially for the disturbance between attitude pitch and roll.

The Qball-X4 system is an underactuated system, but all the inputs are fully controllable. In practical systems, not all the states have direct feedback, due to the limit of available sensors. However, the missing states can be either calculated or estimated.

4.1. Feedback Linearization Control

4.1.1. Controller Design

Based on the design procedure of multiple inputs system (2-13)–(2-17) with tracking errors, both position (x, y, z) and attitude (θ, ϕ, ψ) controllers can be designed.

The cancellation of system nonlinearity is achieved through the matrix inversion $G(x)^{-1}$, which requires that the matrix $G(x)$ has to be invertible. Taking system model (3-24) into consideration, it can be seen there are only four inputs u_1, u_2, u_3, u_4, which make only four states can be controlled. To control the positions and attitude of *x-y-z-yaw*, the model equations can be represented as follows:

$$J_x \quad J_x \quad J_y \quad J_y \quad J_z \tag{4-1}$$

where $u_1 = \dfrac{F_1 + F_2 + F_3 + F_4}{m}$, $u_2 = \dfrac{F_1 - F_2}{J_x}$, $u_3 = \dfrac{F_3 - F_4}{J_y}$, and $u_4 = \dfrac{F_1 + F_2 - F_3 - F_4}{J_z}$.

Regroup into the format of $\boldsymbol{F}(\boldsymbol{X}), \boldsymbol{G}(\boldsymbol{X})$, one obtains

$$\begin{aligned} &\ddot{\boldsymbol{X}} \quad \boldsymbol{F} \; \dot{\boldsymbol{X}} \quad \boldsymbol{G} \; \boldsymbol{X})\boldsymbol{U} \\ &\boldsymbol{Y} = \boldsymbol{H}(\boldsymbol{X}) \end{aligned} \tag{4-2}$$

where

$$\ddot{\boldsymbol{X}} \quad \boldsymbol{F}\ \dot{\boldsymbol{X}} \quad \begin{bmatrix} -d_x \\ -d_y \\ -d_z \\ -d_\psi \end{bmatrix} \qquad \boldsymbol{H}(\boldsymbol{X}) = \begin{bmatrix} x \\ y \\ z \\ \psi \end{bmatrix}$$

$$\boldsymbol{G}(\boldsymbol{X}) = \begin{bmatrix} \cos\psi\sin\theta\cos\phi + \sin\phi\sin\psi & 0 & 0 & 0 \\ \sin\psi\sin\theta\cos\phi - \sin\phi\cos\psi & 0 & 0 & 0 \\ \cos\theta\cos\phi & 0 & 0 & 0 \\ 0 & 0 & 0 & c \end{bmatrix} \qquad \boldsymbol{U} = \begin{bmatrix} u_1 \\ 0 \\ 0 \\ u_4 \end{bmatrix}$$

Due to the fact of non-invertible matrix of $\boldsymbol{G}$, a dynamic extension is used to reform $\boldsymbol{G}$. More derivatives will be taken, and constant c will become to 0; therefore, yaw has to be taken to the attitude control of pitch-roll-yaw.

There is another factor that could have an influence on the performance of the controller, which is noise caused by more derivatives. To minimize the noise sensitivity as much as possible but also keep the derivatives, an assumption of $\psi = 0$ is taken into consideration and as well for the sliding mode control and backstepping control. By doing this, the original equations are reduced, and so is the sensitivity of noise. Besides, the assumption of $\psi = 0$ is practically possible, due to the independency of control input u_4.

Four derivatives are taken on $\boldsymbol{Y} = \boldsymbol{H}(\boldsymbol{X})$ to form the new $\boldsymbol{F}(\boldsymbol{X}), \boldsymbol{G}(\boldsymbol{X})$ as shown:

(4-3)

$$x^{(4)} = \ddot{} \quad \dot{} \quad \dot{} \quad \dot{}$$
$$+u_1 \ddot{} \quad \dot{}\,\dot{} \quad \dot{} \quad \ddot{} \quad \dddot{d}_x$$
$$y^{(4)} = -\ddot{} \quad \dot{} \quad \dddot{} \tag{4-4}$$
$$z^{(4)} = \ddot{} \quad \cos\phi - 2\dot{} \quad \dot{} \quad \dot{}$$
$$-u_1 \ddot{} \quad \dot{}\,\dot{} \quad \dot{} \quad \ddot{} \quad \dddot{d}_z$$

Regroup the above equation into the format of $\boldsymbol{F}(\boldsymbol{X}), \boldsymbol{G}(\boldsymbol{X})$ as indicated in equation (4-2) and place $\ddot{} \ \ddot{}$ into the equations as following:

$$\begin{aligned} \boldsymbol{X}^{(4)} &= \boldsymbol{F'}(\dot{\boldsymbol{X}} \quad \boldsymbol{G} \quad \boldsymbol{X})\boldsymbol{U}' \\ \boldsymbol{Y} &= \boldsymbol{H}(\boldsymbol{X}) \end{aligned} \tag{4-5}$$

where using $s\theta - \sin\theta$, $s\phi - \sin\phi$, $c\theta - \cos\theta$, $c\phi - \cos\phi$, the elements of matrix $\boldsymbol{F'}(\dot{\boldsymbol{X}}$ are given as:

$$\boldsymbol{F}'(\dot{\boldsymbol{X}} \quad \begin{bmatrix} F_1'(\dot{} \\ \dot{} \\ F_3'(\dot{} \end{bmatrix}$$

$$F_1'(\dot{} \quad \dot{} \quad \dot{} \quad J_x \quad J_x \quad \dot{}$$
$$-2u_1\dot{}\,\dot{} \quad \dot{} \quad J_y \quad J_y \quad \dot{} \quad \dddot{}$$

$$F_2'(\dot{} \quad \dot{} \quad J_y \quad J_y \quad \dddot{}$$

$$F_3'(\dot{} \quad \dot{} \quad \dot{} \quad J_x \quad J_x \quad \dot{}$$
$$+2u_1\dot{}\,\dot{} \quad \dot{} \quad J_y \quad J_y \quad \dot{} \quad \dddot{}$$

$$\boldsymbol{G}'(\boldsymbol{X}) = \begin{bmatrix} \sin\theta\cos\phi & lu_1\cos\theta\cos\phi & -lu_1\sin\theta\sin\phi \\ -\sin\phi & 0 & -lu_1\cos\phi \\ \cos\theta\cos\phi & -lu_1\sin\theta\cos\phi & -lu_1\cos\theta\sin\phi \end{bmatrix} \qquad \boldsymbol{U}' = \begin{bmatrix} \ddot{} \\ u_2 \\ u_3 \end{bmatrix}$$

For a tracking task, tracking errors of *x-y-z* are defined as the following:

$$\begin{bmatrix} e_x^{(4)} - k_{fx1}e_x^{(3)} - k_{fx2}\ddot{} \quad \dot{} \\ e_y^{(4)} - k_{fy1}e_y^{(3)} - k_{fy2}\ddot{} \quad \dot{} \\ e_z^{(4)} - k_{fz1}e_z^{(3)} - k_{fz2}\ddot{} \quad \dot{} \end{bmatrix} = 0 \tag{4-6}$$

where $e_x = x_d - x;\ e_y = y_d - y;\ e_z = z_d - z.$

The overall system controller for altitude *x-y-z* is then designed as:

$$\boldsymbol{Y}^{(4)} = \begin{bmatrix} x^{(4)} \\ y^{(4)} \\ z^{(4)} \end{bmatrix} = \boldsymbol{V} = \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix} = \begin{bmatrix} x_d^{(4)} - k_{fx1}e_x^{(3)} - k_{fx2}\ddot{} \quad \dot{} \\ y_d^{(4)} - k_{fy1}e_y^{(3)} - k_{fy2}\ddot{} \quad \dot{} \\ z_d^{(4)} - k_{fz1}e_z^{(3)} - k_{fz2}\ddot{} \quad \dot{} \end{bmatrix} \tag{4-7}$$

$$\begin{bmatrix} \ddot{} \\ u_2 \\ u_3 \end{bmatrix} = \boldsymbol{G}'(\boldsymbol{X})^{(-1)} * (-\boldsymbol{F}'(\dot{\boldsymbol{X}} \quad \boldsymbol{V} \tag{4-8}$$

where $k_{fxi}, k_{fyi}, k_{fzi},\ i = 1, \ldots$ are control gains.

Two simple integrations can get u_1 from $\ddot{}$. To control the atitude of pitch-roll-yaw, the model equations are giveb by:

$$\begin{aligned} &\ddot{} \quad J_x \quad J_x \\ &\ddot{} \quad J_y \quad J_y \\ &\ddot{} \quad J_z \end{aligned} \tag{4-9}$$

Similar to the procedure of the *x-y-z* controller, regroup the above equations into the format of $\boldsymbol{F}(\boldsymbol{X}), \boldsymbol{G}(\boldsymbol{X})$ to obtain following matrix-vector format:

$$\ddot{\boldsymbol{X}} \quad \boldsymbol{F} \ \dot{\boldsymbol{X}} \quad \boldsymbol{G} \ \boldsymbol{X})\boldsymbol{U}$$
$$\boldsymbol{Y} = \boldsymbol{H}(\boldsymbol{X})$$

where

$$\ddot{\boldsymbol{X}} \quad \begin{bmatrix} \ddot{\iota} \\ \\ \ddot{\iota} \end{bmatrix} \qquad \boldsymbol{F} \ \dot{\boldsymbol{X}} \quad \begin{bmatrix} \dfrac{(J_y - J_z)\dot{\iota}\dot{\ }}{J_x} - \dfrac{\quad}{J_x} - a_\theta \dot{\iota} \\ \dfrac{\quad}{J_y} \qquad \dfrac{\quad}{J_y} \\ \dfrac{(J_x - J_y)\dot{\iota}\dot{\ }}{J_z} - a_\psi \dot{\iota} \end{bmatrix} \qquad \boldsymbol{H} \ \boldsymbol{X} \quad \begin{bmatrix} \theta \\ \\ \psi \end{bmatrix}$$

$$\boldsymbol{G}(\boldsymbol{X}) = \begin{bmatrix} l & 0 & 0 \\ 0 & l & 0 \\ 0 & 0 & c \end{bmatrix} \qquad \boldsymbol{U} = \begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix}$$

Since the matrix $\boldsymbol{G}$ is already invertible, no extension is needed. The tracking errors for θ, ϕ, ψ are defined as the same as x, y, z.

$$\ddot{\boldsymbol{Y}} \qquad \boldsymbol{V} \qquad \begin{bmatrix} \ddot{\iota} & \ddot{\ } & \dot{\ } \\ & \ddot{\ } & \dot{\ } \\ \ddot{\iota} & \ddot{\ } & \dot{\ } \end{bmatrix} \tag{4-10}$$

where $k_{f\theta i}, k_{f\phi i}, k_{f\psi i}$, $i = 1,2$ are control gains, and $e_\theta = \theta_d - \theta$, $e_\phi = \phi_d - \phi$, $e_\psi = \psi_d - \psi$.

The overall system controller for attitude pitch-roll-yaw is then designed as:

$$\begin{bmatrix} u_2 \\ u_3 \\ u_4 \end{bmatrix} = \boldsymbol{G}^{(-1)}(-\boldsymbol{F}(\dot{\boldsymbol{X}} \quad \boldsymbol{V} \tag{4-11}$$

For either position controller or attitude controller, when control inputs are set to zeros, the outputs become zeros. Zero dynamics applies to both controllers, and the system is stable.

4.1.2. Simulations

In this section, the performance of designed controller will be tested, by giving a desired path in the form of coordinates to *x-y-z*. The position controller or attitude of the Qball-X4 will be shown to prove the stability and tracking performance of the controller.

Ignoring all the drag forces for $x, y, z, \theta, \phi, \psi$ and setting $\psi_d = 0$, and without disturbance, the results are shown in details for both *x-y-z* coordinates and pitch, roll and yaw angles. The control inputs are shown as well in both Voltage (u) and Newton (F).

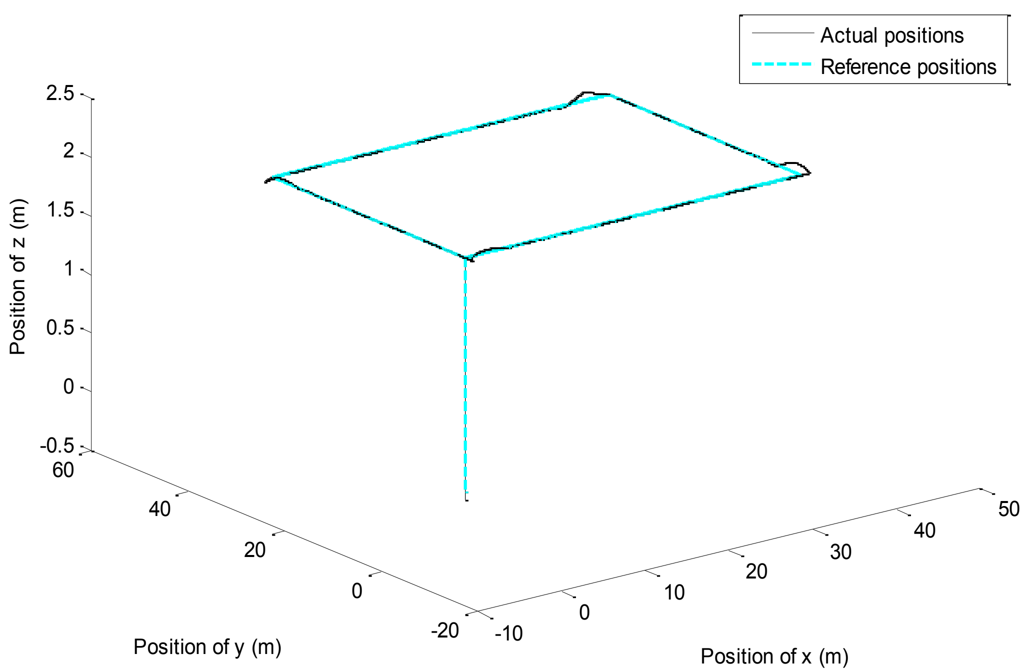

Fig. 4-1. 3-dimensional path tracking

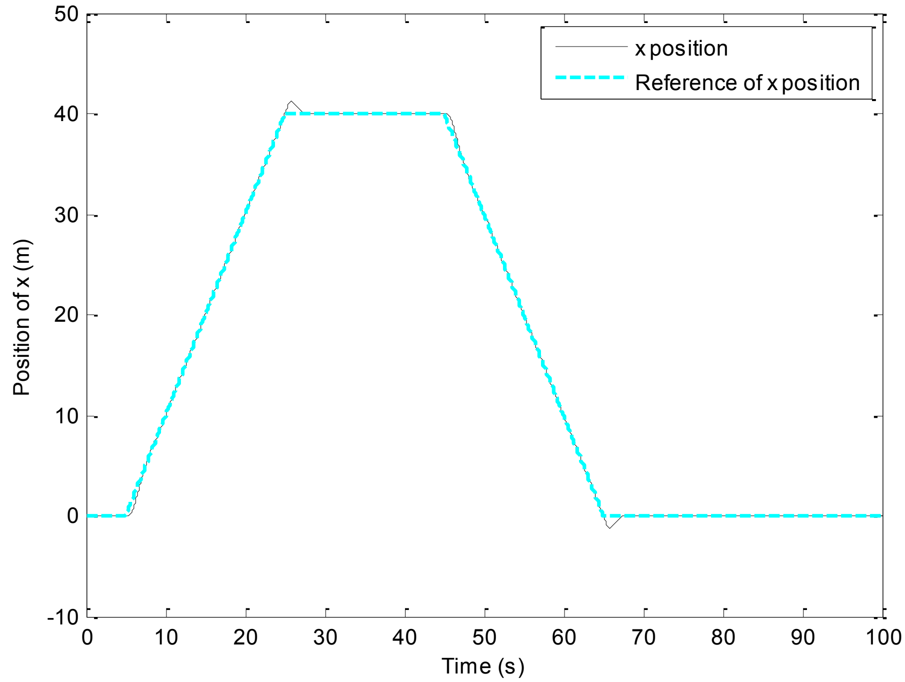

Fig. 4-2. Position tracking in *x* axis

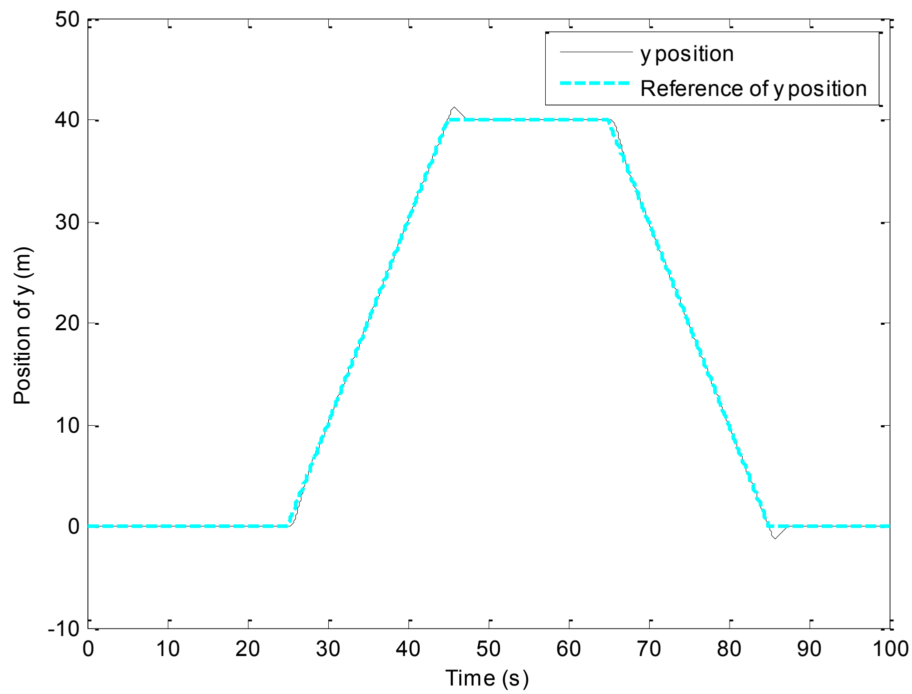

Fig. 4-3. Position tracking in *y* axis

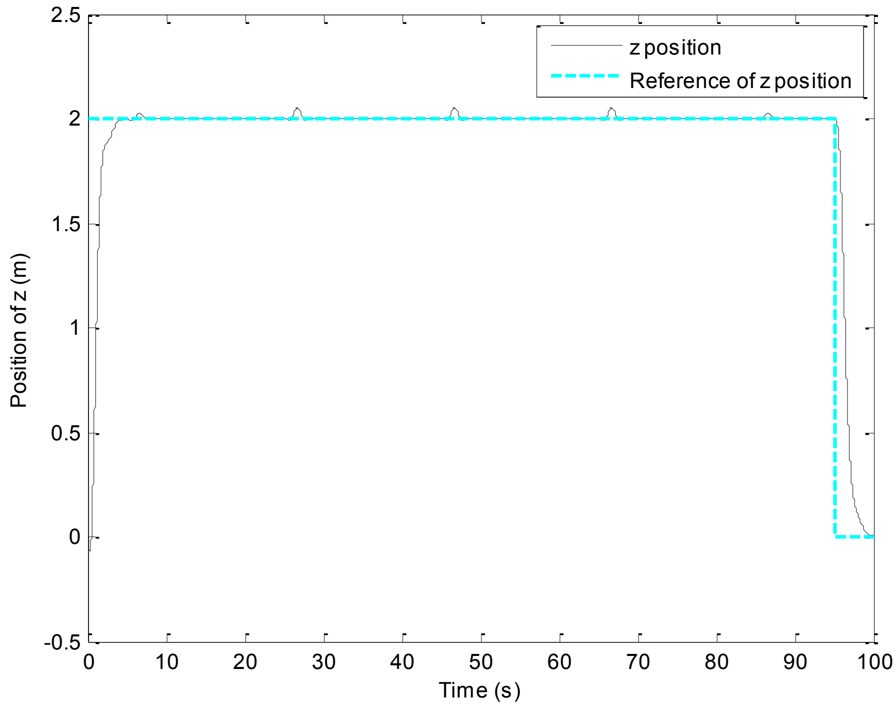

Fig. 4-4. Position tracking in *z* axis

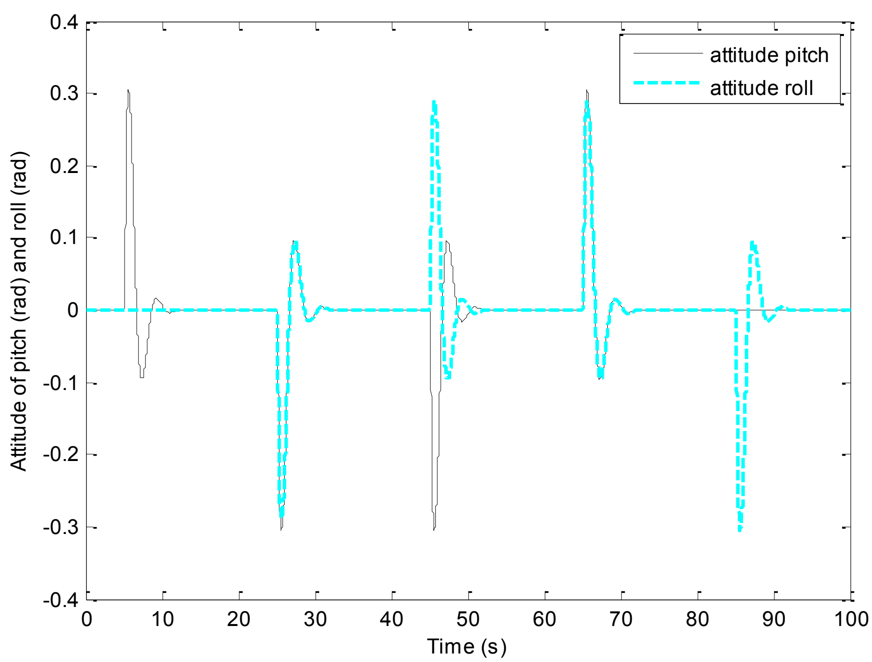

Fig. 4-5. Pitch and roll angles

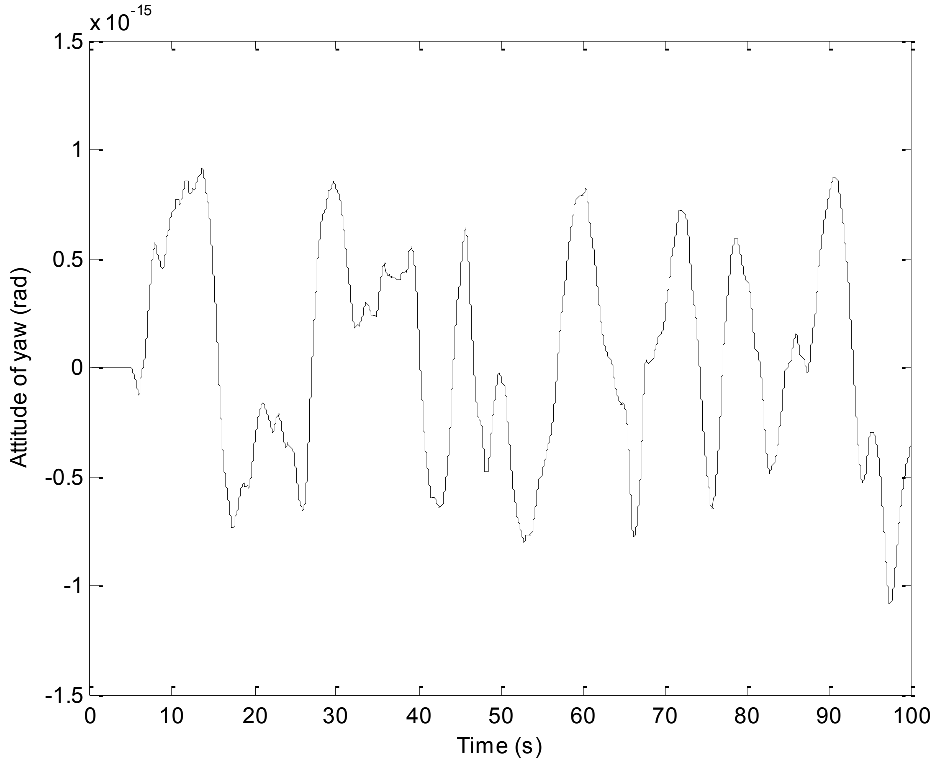

Fig. 4-6. Attitude of yaw angle

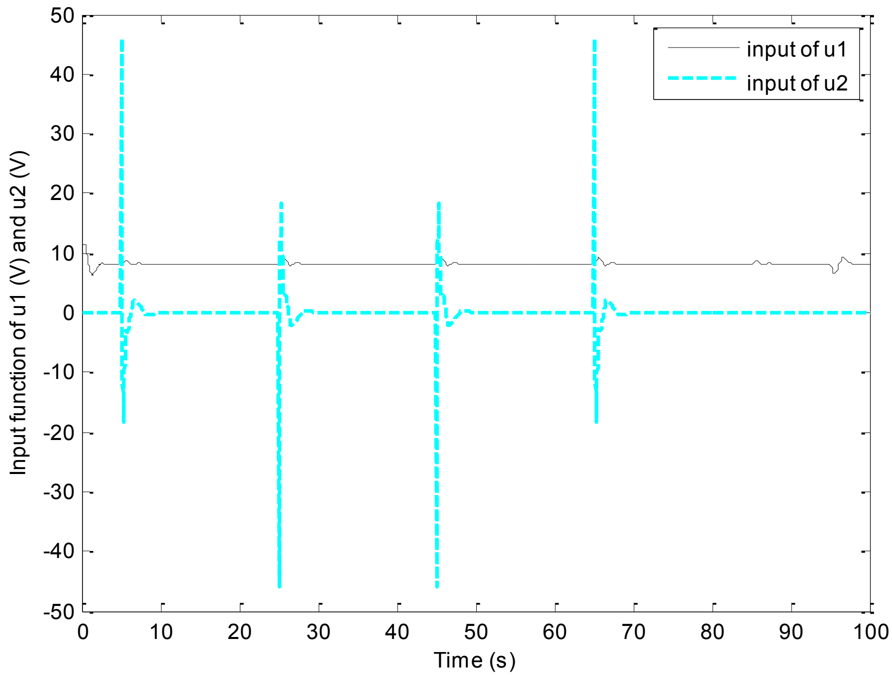

Fig. 4-7. Control inputs of u_1 and u_2

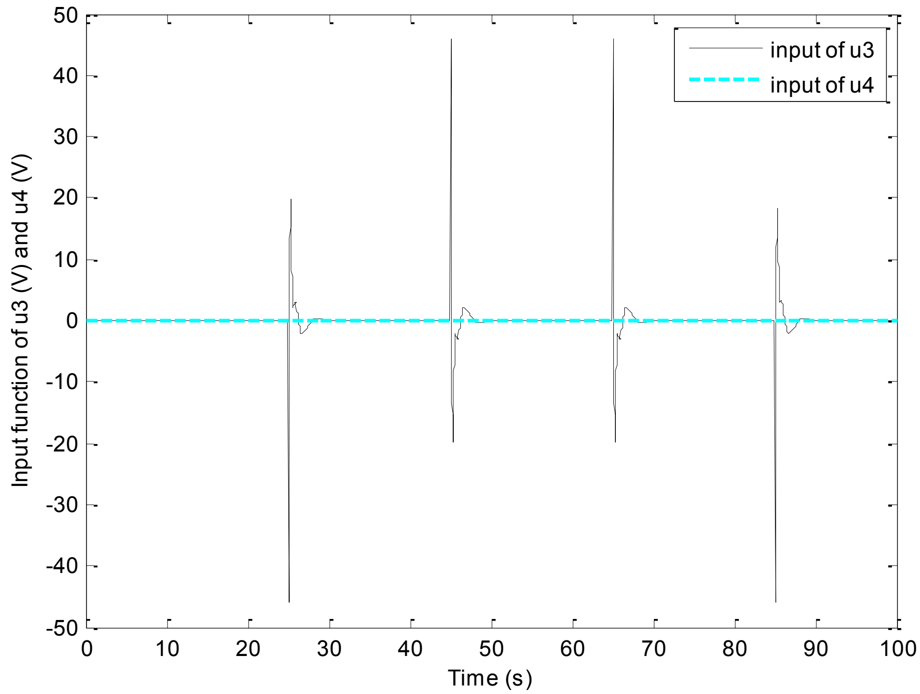

Fig. 4-8. Control inputs of u_3 and u_4

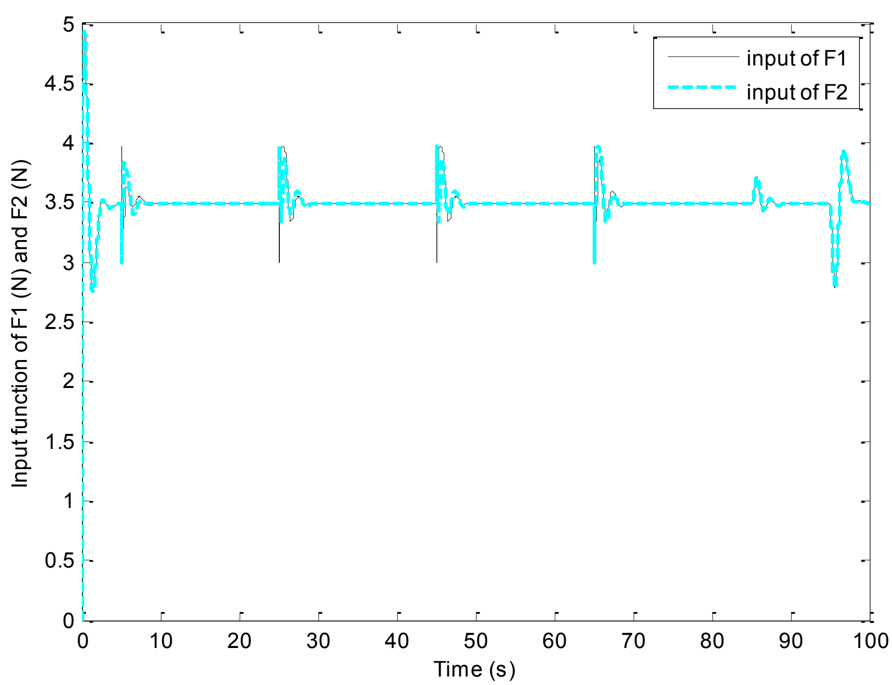

Fig. 4-9. Propellers forces F_1 and F_2

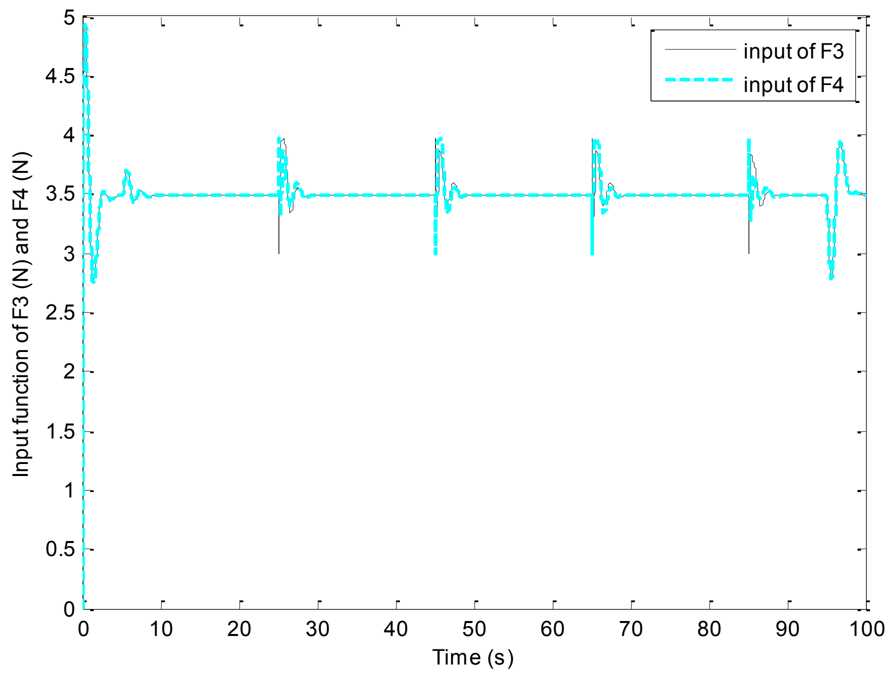

Fig. 4-10. Propellers forces F_3 and F_4

The results show the controller behaves properly. Overall system is stable and the desired path has been tracked. To test the robustness of the controller, disturbances and noises need to be added. Hence, drag forces and gyroscopic effect Ω are added randomly, and so are the sensor noises. With the same controller, another set of simulation results are presented in Fig. 4-11 to Fig. 4-20 as follows.

The tracking performance is deteriorated than the previous case due to the effects of highly coupled matrix $\boldsymbol{G}'(\boldsymbol{X})\boldsymbol{U}'$, extra disturbances and noises. However, the controller can still be able to stabilize the system and to follow the same desired trajectory.

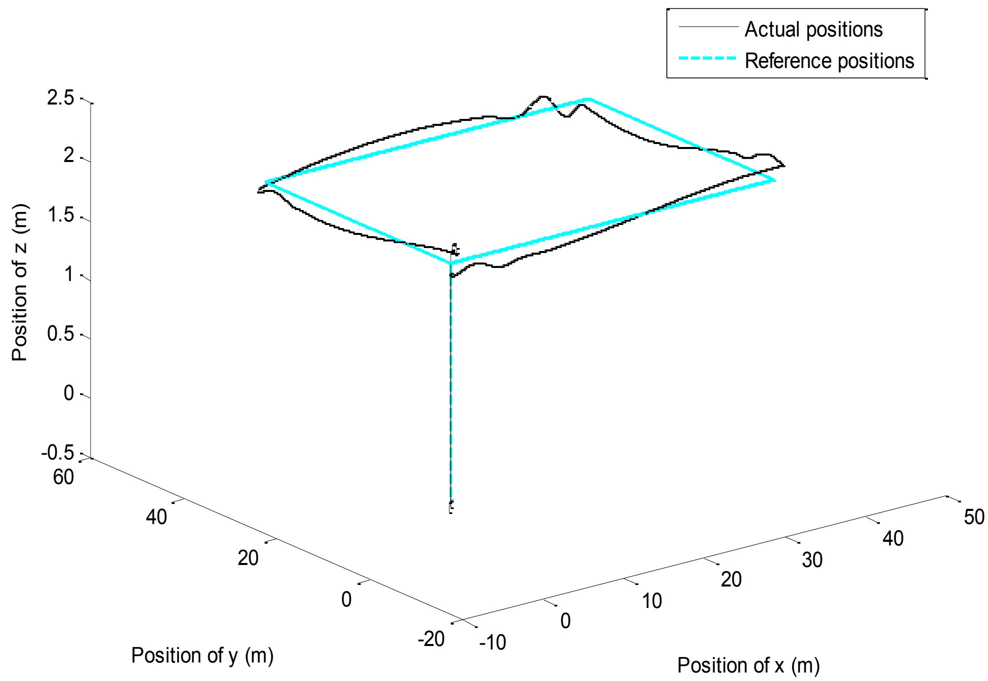

Fig. 4-11. 3-dimensional path tracking

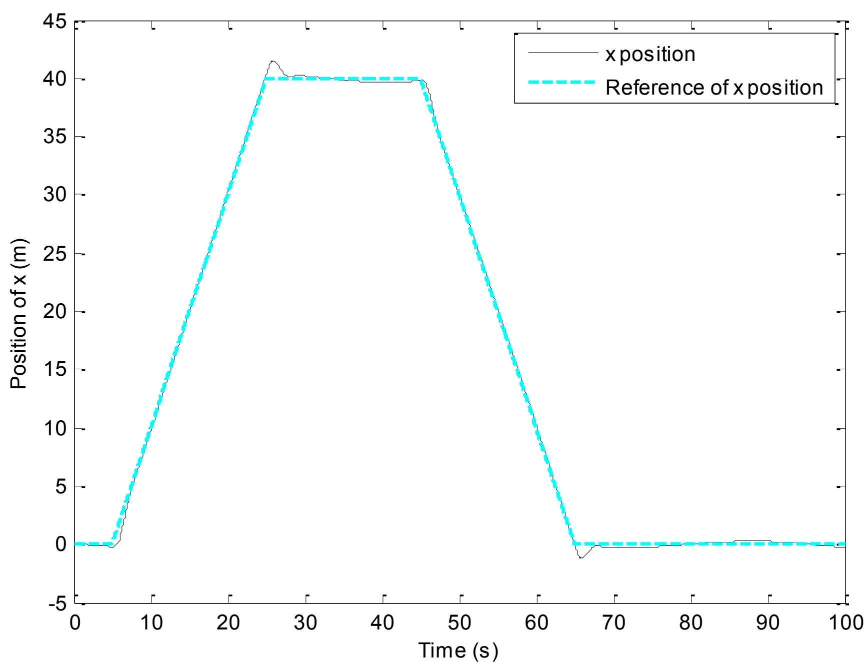

Fig. 4-12. Position tracking in *x* axis

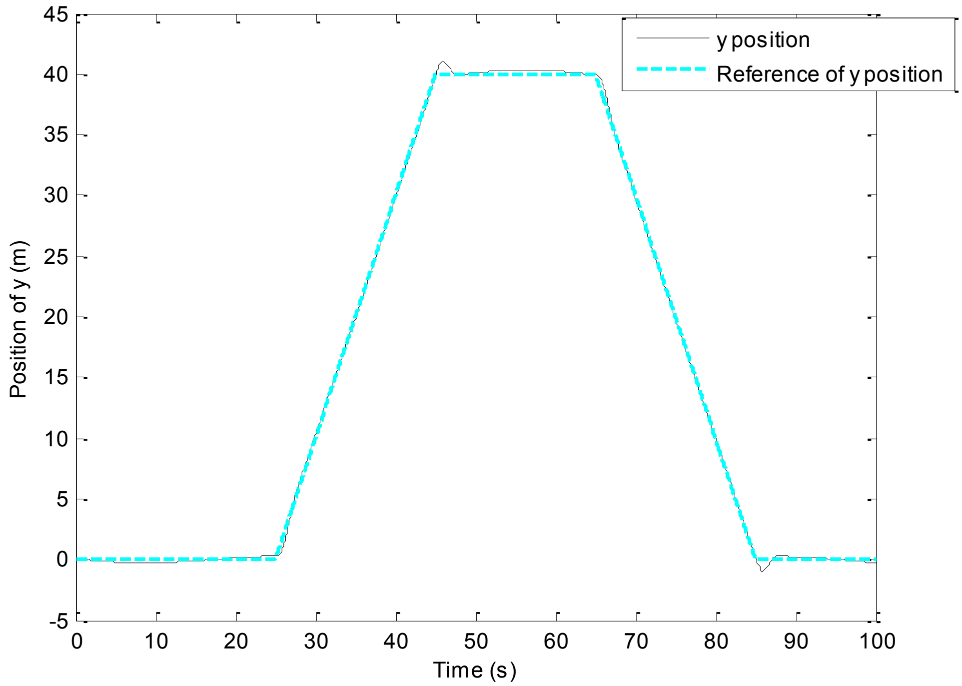

Fig. 4-13. Position tracking in *y* axis

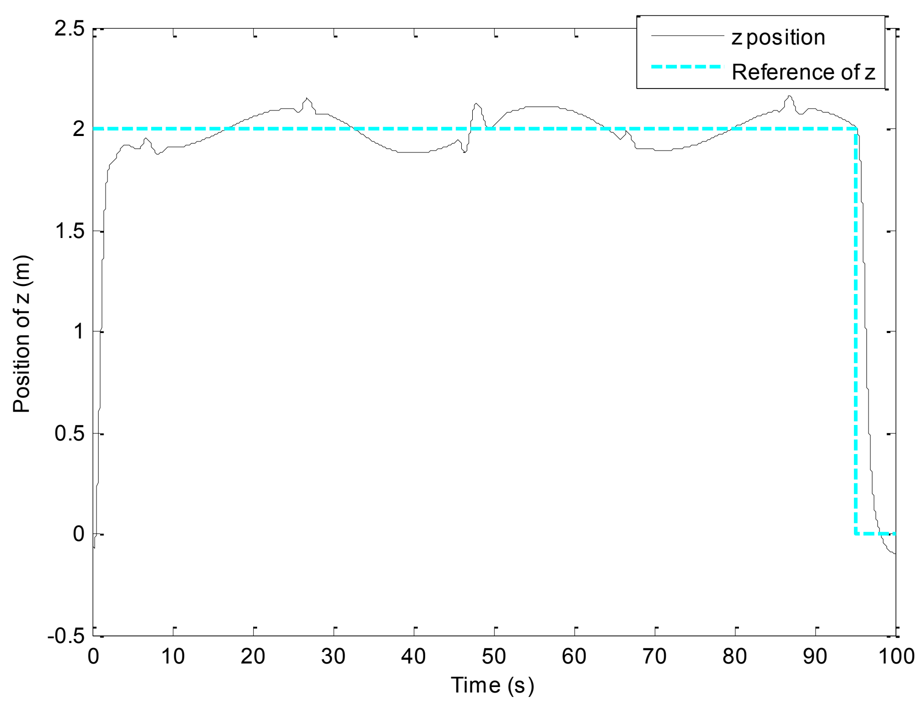

Fig. 4-14. Position tracking in *z* axis

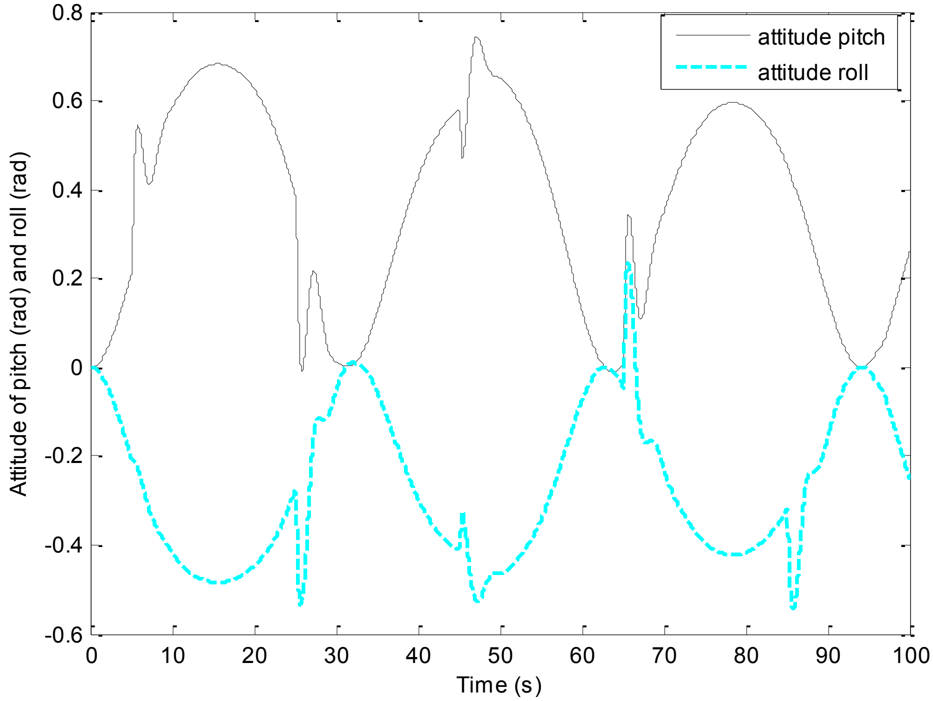

Fig. 4-15. Attitude of pitch and roll angles

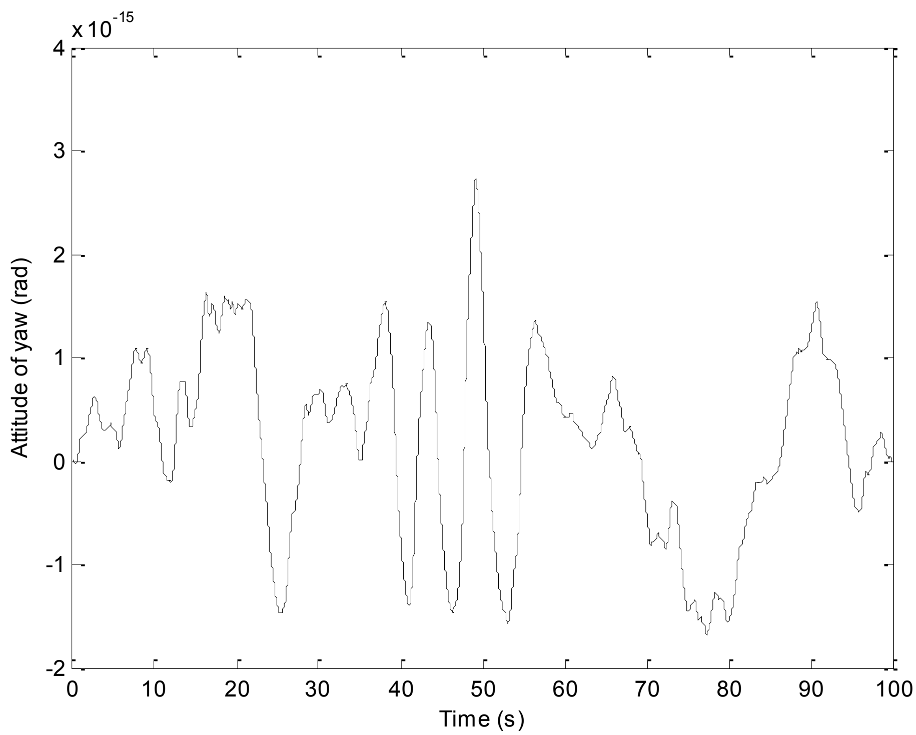

Fig. 4-16. Attitude of yaw angle

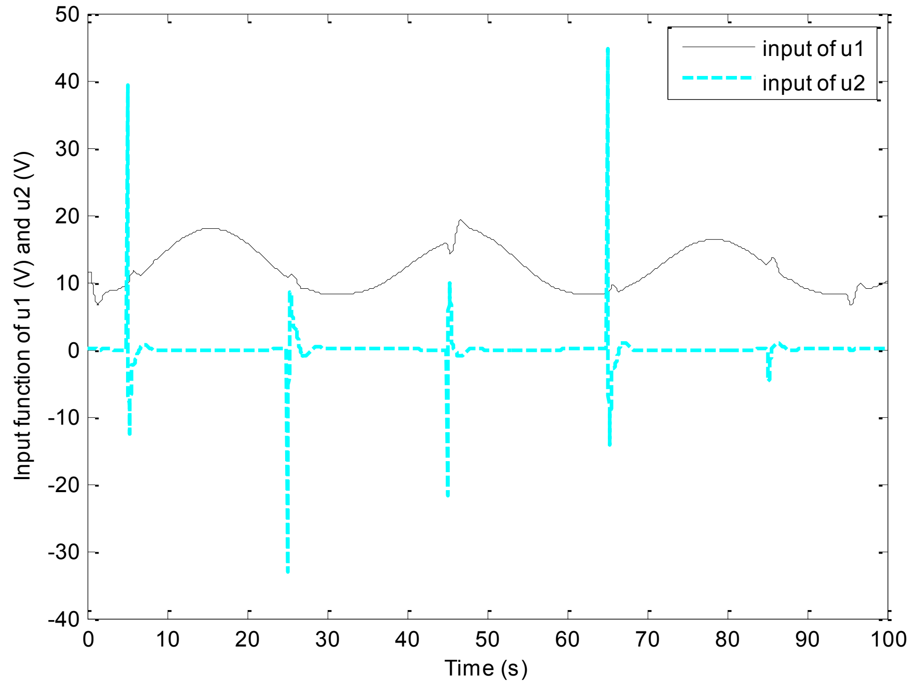

Fig. 4-17. Control inputs of u_1 and u_2

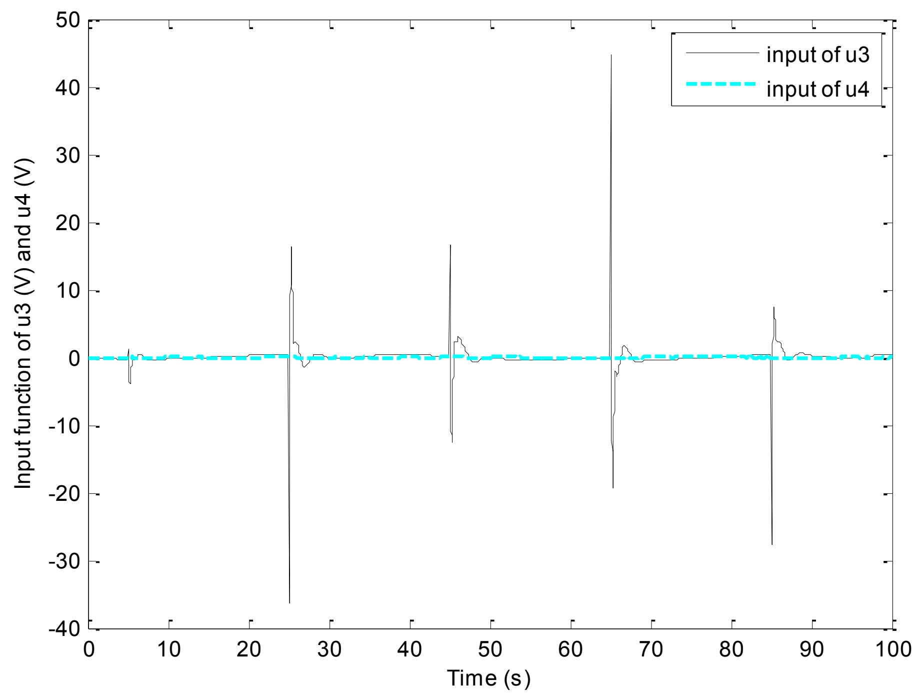

Fig. 4-18. Control inputs of u_3 and u_4

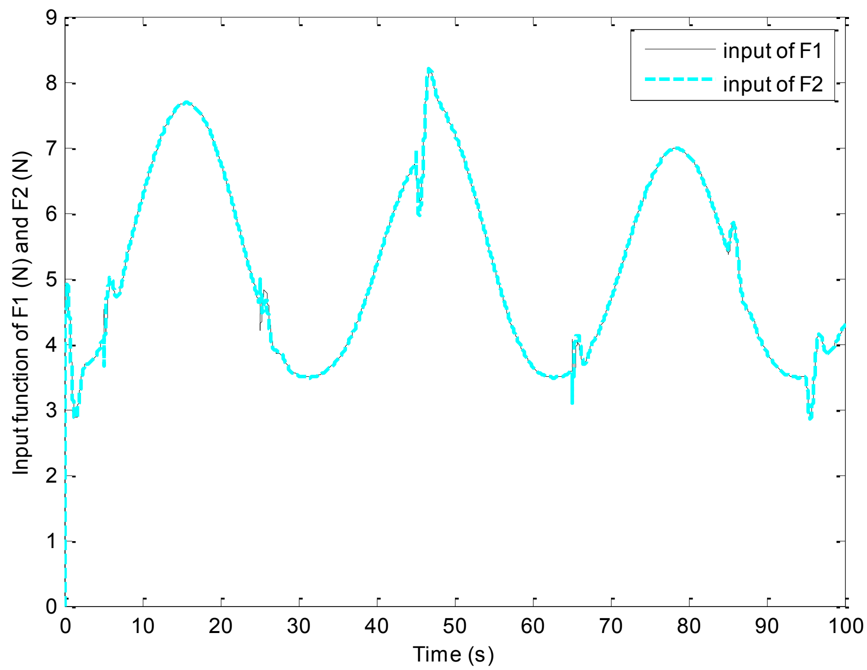

Fig. 4-19. Propellers forces F_1 and F_2

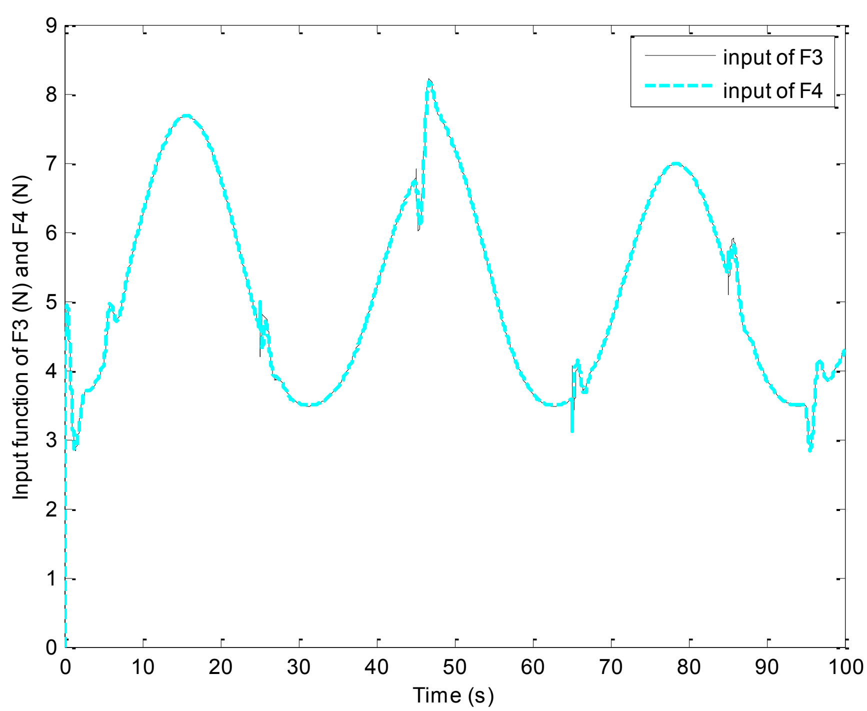

Fig. 4-20. Propellers forces F_3 and F_4

4.2. Sliding Mode Control

4.2.1. Controller Design

Through equations (2-39)-(2-46) in Section 2.2, position (x, y, z) and attitude (θ, ϕ, ψ) controllers can be realized using sliding mode technique for the Qball-X4.

By equation (3-24), the system is in second-order, and only six states as [are not adequate for feedback. Therefore, an expansion of the states has been taken into consideration as:

$$x_2, x_3, x_4, x_5, x_6, x_7, x_8, x_9, x_{10}, x_{11}, x_{12}]^T \tag{4-12}$$

In details, it can be written as:

$$\dot{X} \qquad \begin{bmatrix} x_2 \\ (\cos\phi\sin\theta\cos\psi+\sin\phi\sin\psi)-d_x x_2 \\ x_4 \\ \cos\phi\sin\theta\sin\psi-\sin\phi\cos\psi)-d_y x_4 \\ x_6 \\ u_1(\cos\phi\cos\theta)-g-d_z x_6 \\ x_8 \\ \frac{J_y-J_z}{J_x}x_{10}x_{12}+u_2 l-\frac{J_r x_{10}\Omega}{J_x}-d_\theta x_8 \\ x_{10} \\ \frac{J_z-J_x}{J_y}x_8 x_{12}+u_3 l+\frac{J_r x_8\Omega}{J_y}-d_\phi x_{10} \\ x_{12} \\ \frac{J_x-J_y}{J_z}x_8 x_{10}+cu_4-d_\psi x_{12} \end{bmatrix} \tag{4-13}$$

where $u_1 = \dfrac{F_1 + F_2 + F_3 + F_4}{m}$, $u_2 = \dfrac{F_1 - F_2}{J_x}$, $u_3 = \dfrac{F_3 - F_4}{J_y}$, $u_4 = \dfrac{F_1 + F_2 - F_3 - F_4}{J_z}$.

$u_x = \cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi$ as a virtual input, and as well a virtual input of $u_y = \cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi$.

In order to follow the desired path, a tracking error needs to be defined as mentioned in Chapter 2, i.e., $e_i = x_d - x_i$, where $x_i \in R^n$. Instead of choosing equation (2-26) as the sliding surface, an integration of tracking error component has been introduced into the surface. Therefore, a faster convergence and a smoother tracking trajectory would be achieved by the following equations:

$$s_i(t) = \qquad k_{pi} \int e_i \tag{4-14}$$

To stabilize the controller, sliding condition has to be satisfied, which is

for $i = x, y, z, \phi, \theta, \psi$. By the principles (2-29) and (2-30), the sliding mode controller can then be derived.

From (4-14), one can obtain

$$\tag{4-15}$$

a tracking error component $k_{pi}e_i$ is used to obtain a faster convergence and a better stability. A function is also needed to be chosen instead of the original .

Then, the approximation of control input u_x is given by:

$$\hat{u}_x = \frac{1}{u_1}[\qquad + 2k_{p1}e_x] \tag{4-16}$$

where $\hat{u}_x$ is a virtual control input approximation with u_1 as a constant.

Then, the control input is derived as:

$$u_x = \hat{u}_x + k_{s1} sign(s_x) \tag{4-17}$$

From equations (2-37) and (2-38), (4-16) can be written as:

$$\hat{u}_x = \frac{1}{u_1}[\ \cdots\ 2k_{p1}e_x] \tag{4-18}$$

Then, sliding condition of (2-28) is satisfied.

$$\begin{aligned}\frac{1}{2}\frac{d}{dt}s^2 &= s\cdots \\ &= s\{\cdots + 2k_{p1}e_x] + k_{s1} sign(s_x)] \\ &\quad + f(x) + \lambda_1(\cdots) \\ &= (f(x) - \hat{f}(x) - k_{p1}e_x)s - k_{s1}|s| \le -\eta |s|\end{aligned} \tag{4-19}$$

where $|f(x) - \hat{f}(x) - k_{p1}e_x| \le \bar{f}(x)$ and $k_{s1} = \bar{f}(x) + \eta$. The rest of control inputs as $u_y, u_z, u_\theta, u_\phi, u_\psi$ are all followed the same stablilization rules of equations (4-18) and (4-19).

Following the similar procedure, sliding condition for *y*-position controller can be obtained based on the controller structure given in equation (4-14).

$$\cdots \tag{4-20}$$

with $\cdots_y$, and

$$\hat{u}_y = \frac{1}{u_1}[\ \cdots + 2k_{p2}e_y] \tag{4-21}$$

so that the virtual control input u_y is,

$$u_y = \hat{u}_y + k_{s2} sign(s_y) \tag{4-22}$$

Similarly, for z, θ, ϕ, ψ, one can obtain following conditions:

$$\cdots J_x \quad J_x \quad J_y \quad J_y \quad J_z \quad + k_{p6} e_{\psi} \tag{4-23}$$

all the approximations are written as the following:

$$\begin{aligned}
\hat{u}_1 &= \frac{1}{\cos\phi\cos\theta}[\ \cdots \\
\hat{u}_2 &= \frac{1}{l}(\ \cdots J_x \quad J_x \\
\hat{u}_3 &= \frac{1}{l}(\ \cdots J_y \quad J_y \\
\hat{u}_4 &= \frac{1}{c}(\ \cdots J_z \quad + 2k_{p6} e_{\psi})
\end{aligned} \tag{4-24}$$

so that, the final control inputs are represented as:

$$u_i = \hat{u}_i + k_{si} sign(s_i) \tag{4-25}$$

where λ_i, k_{si}, and k_{pi} are all positive gains.

4.2.2. Simulations

By using the same desired path as given to FLC, the performance of position and attitude control of the Qball-X4 system will be tested to show the stability and tracking performance of the designed SMC.

Ignoring all the drag forces for $x, y, z, \theta, \phi, \psi$ and maintaining desired ψ_d angle at zero at all times, simulations without disturbance are shown for position in *x,y,z*, altitude in pitch, roll, and yaw, and the control inputs in both Voltage (u) and Newton (F).

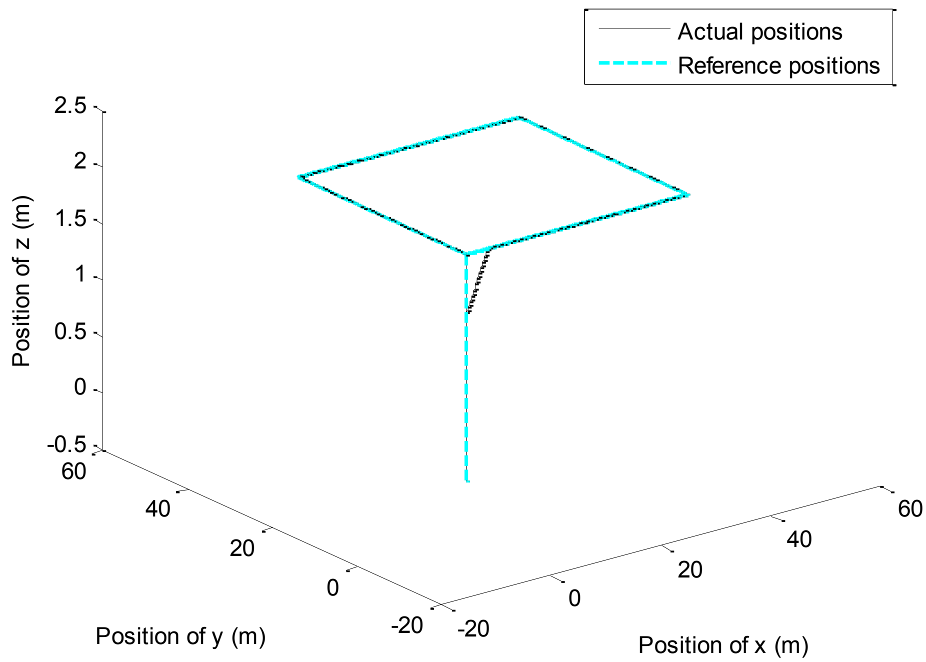

Fig. 4-21. 3-dimensional path tracking

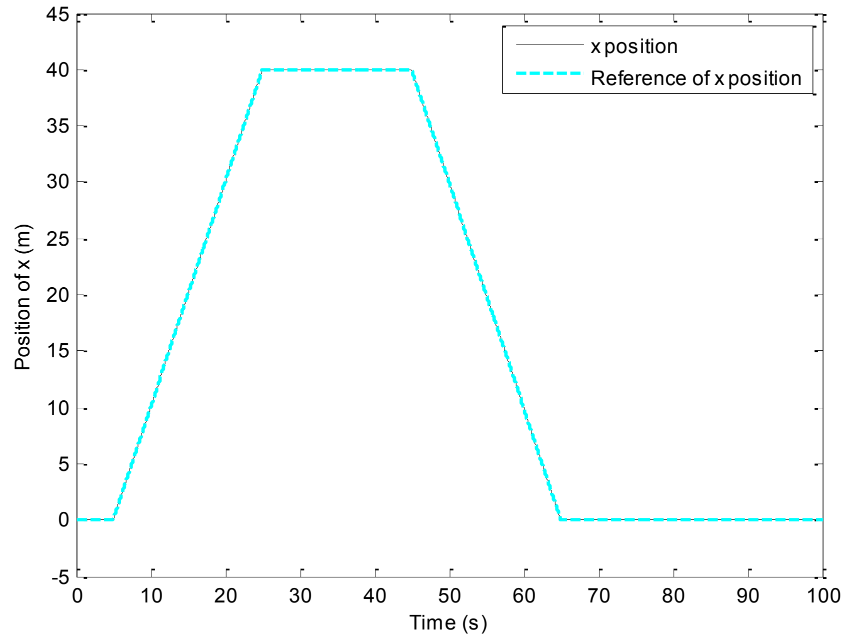

Fig. 4-22. Position tracking in *x* direction

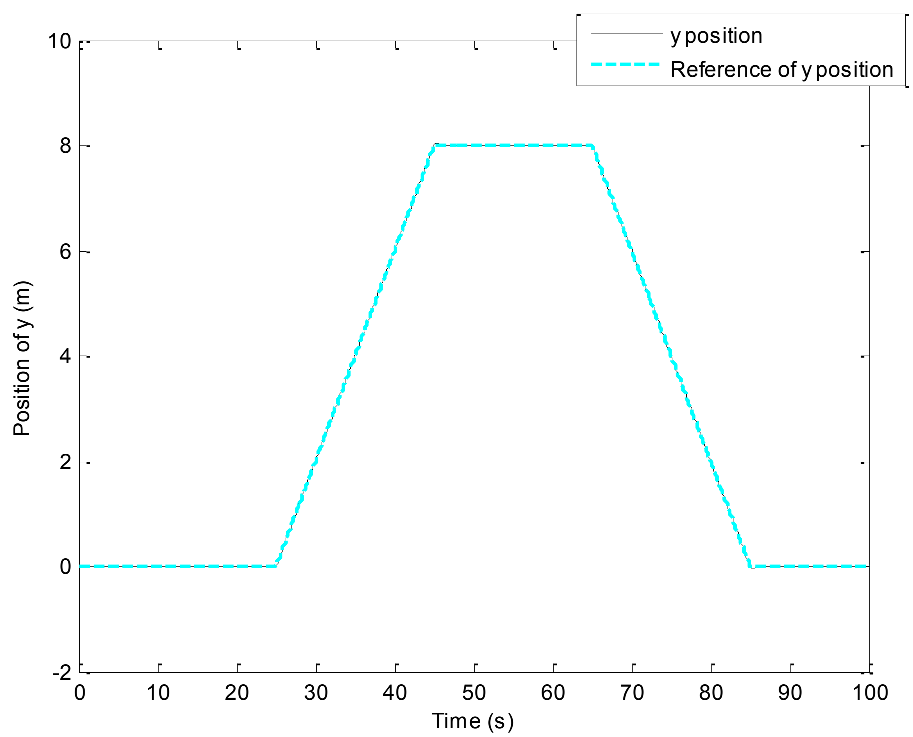

Fig. 4-23. Position tracking in *y* direction

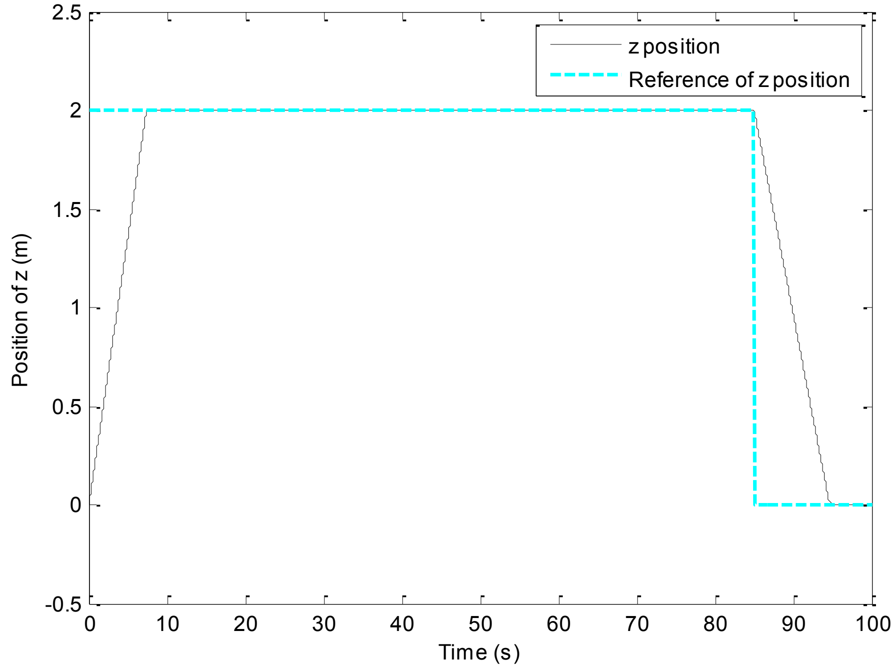

Fig. 4-24. Position tracking in *z* direction

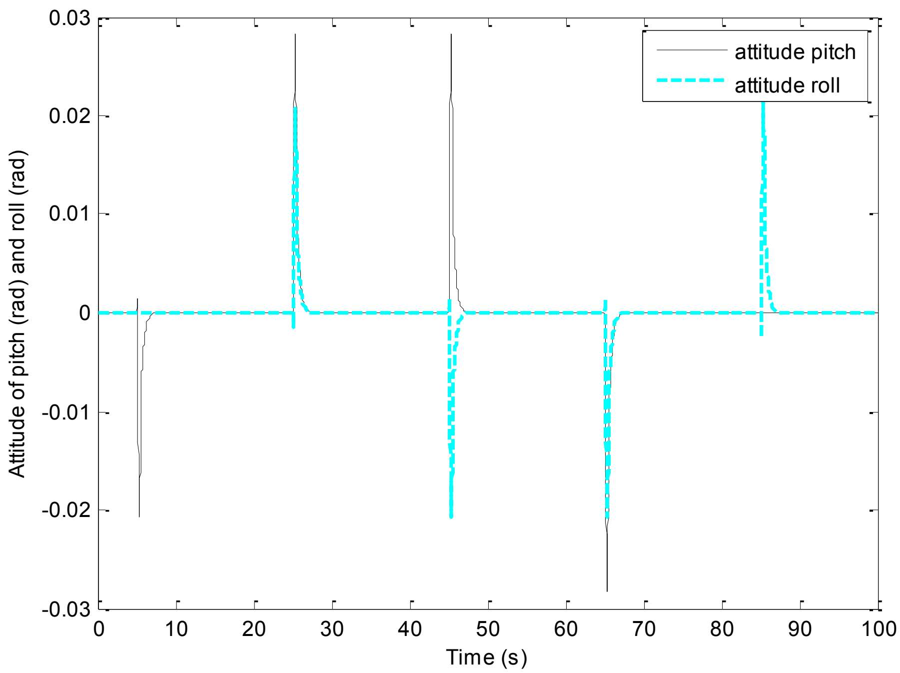

Fig. 4-25. Attitude of pitch and roll angles

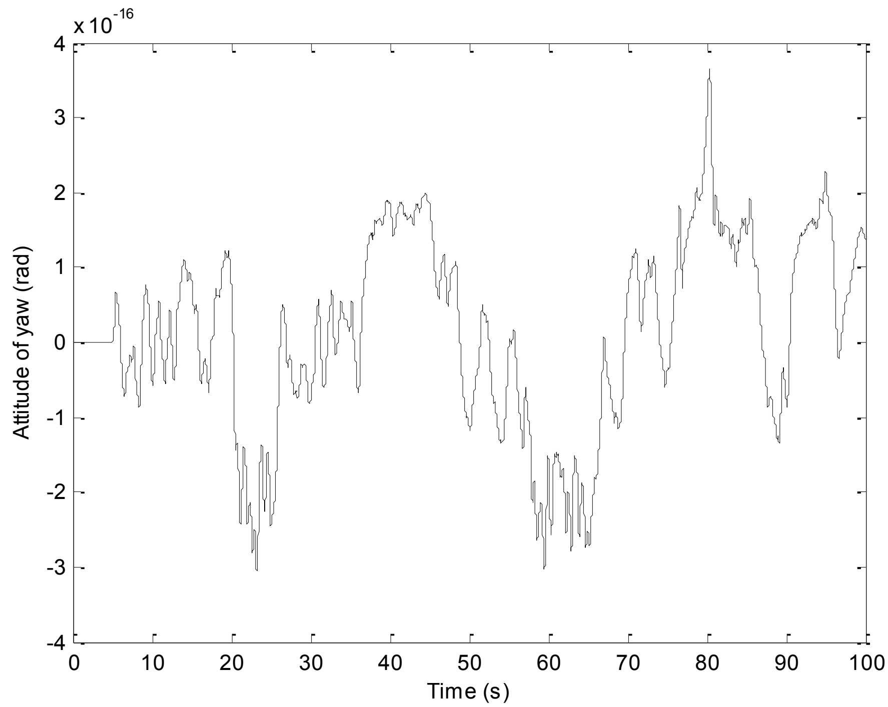

Fig. 4-26. Attitude of yaw angle

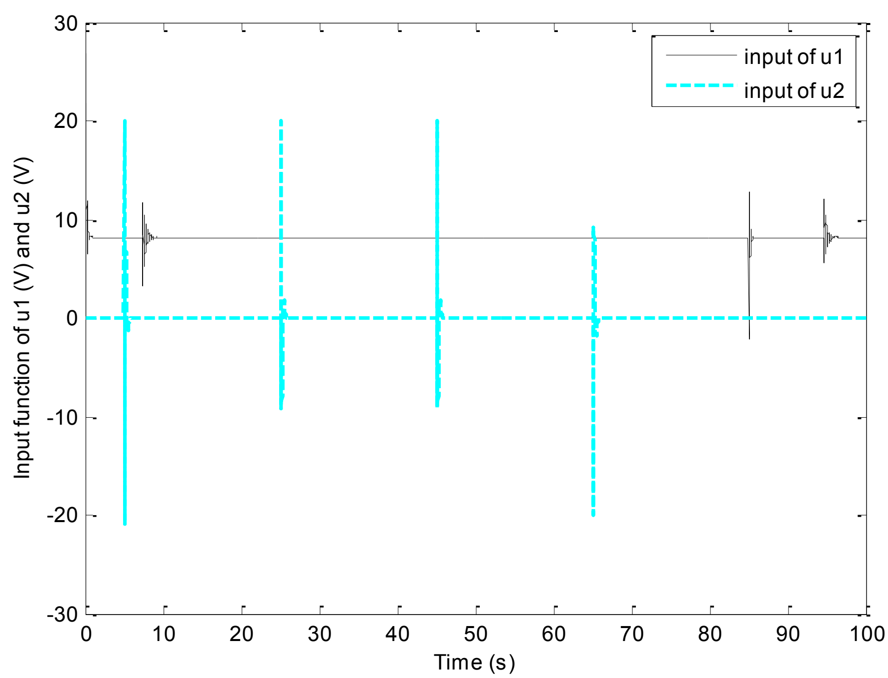

Fig. 4-27. Control inputs of u_1 and u_2

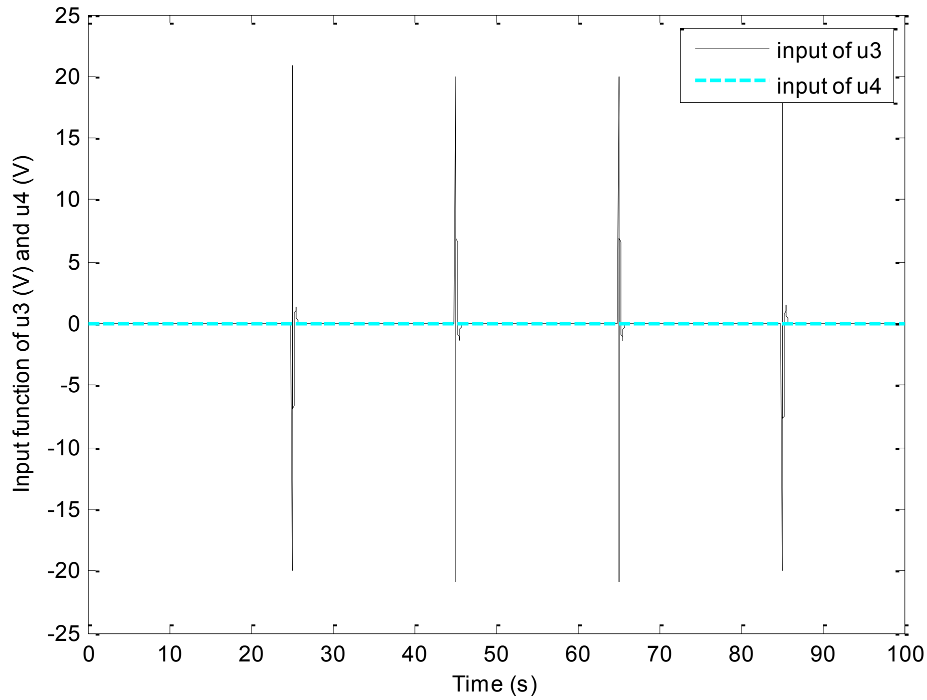

Fig. 4-28. Control inputs of u_3 and u_4

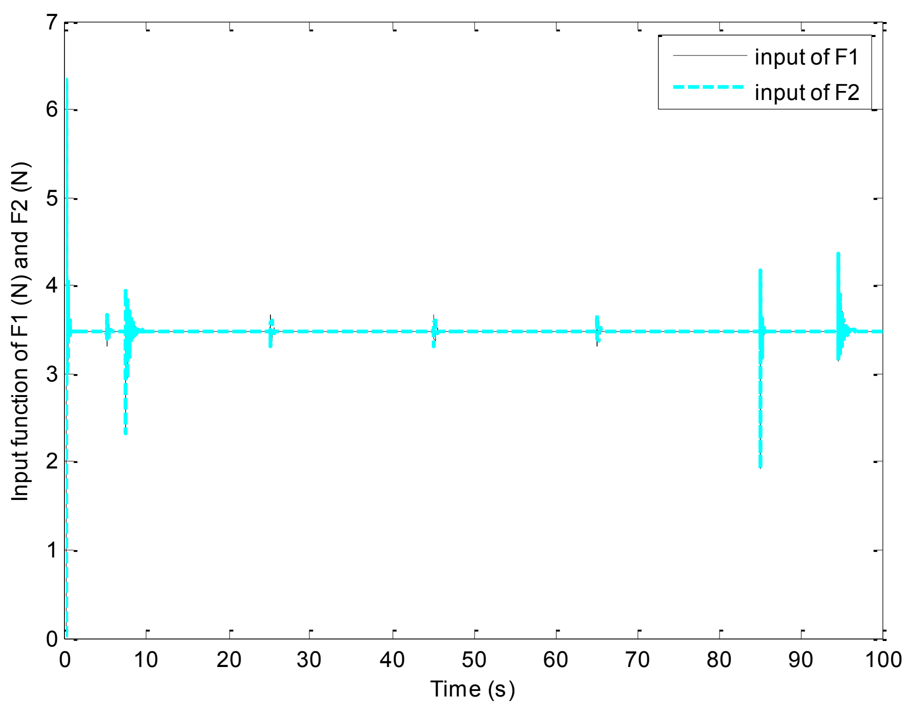

Fig. 4-29. Propellers forces F_1 and F_2

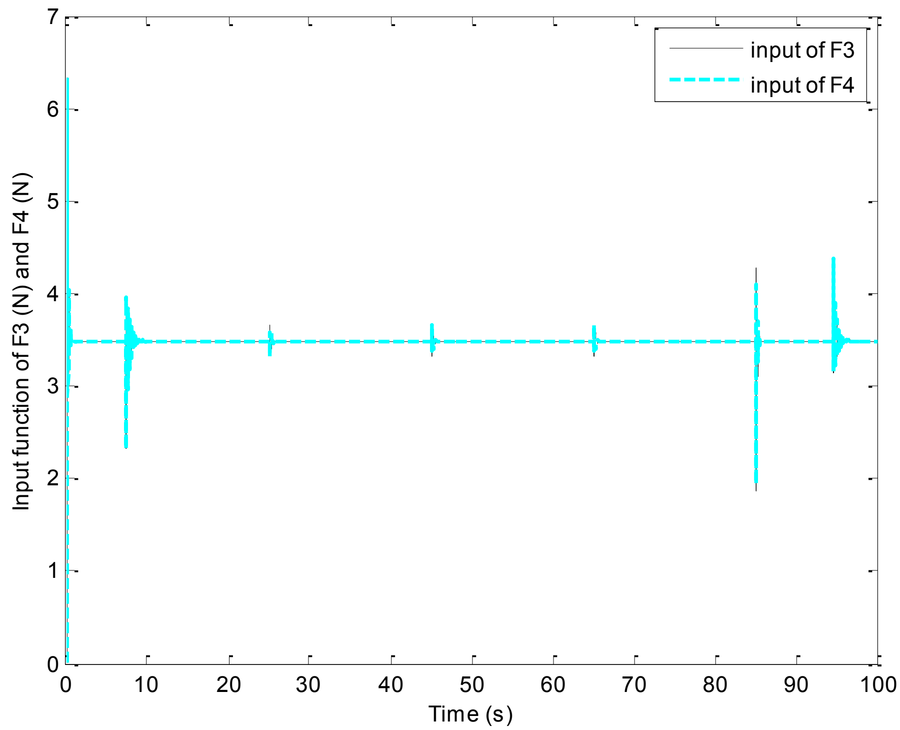

Fig. 4-30. Propellers forces F_3 and F_4

Figures 4-21 to 4-30 have shown that excellent tracking performance has been achieved. Without any coupled matrix as in the FLC, the change of one control input of SMC will not affect the other inputs. All the control inputs are maintained within a relatively small range, and the trajectory is tracked smoothly.

For robustness, drag forces, sensors noises and disturbance Ω are added randomly. Using the same controller, another set of simulations have been carried out and the results are shown below, as the same sequence as previously. The tracking performance of the sliding mode controller is expected to be deteriorated, however the overall system should still be under control, as it can be seen from Figures 4-31 to 4-40. The augmented sliding surface has ensured the robustness of the control system.

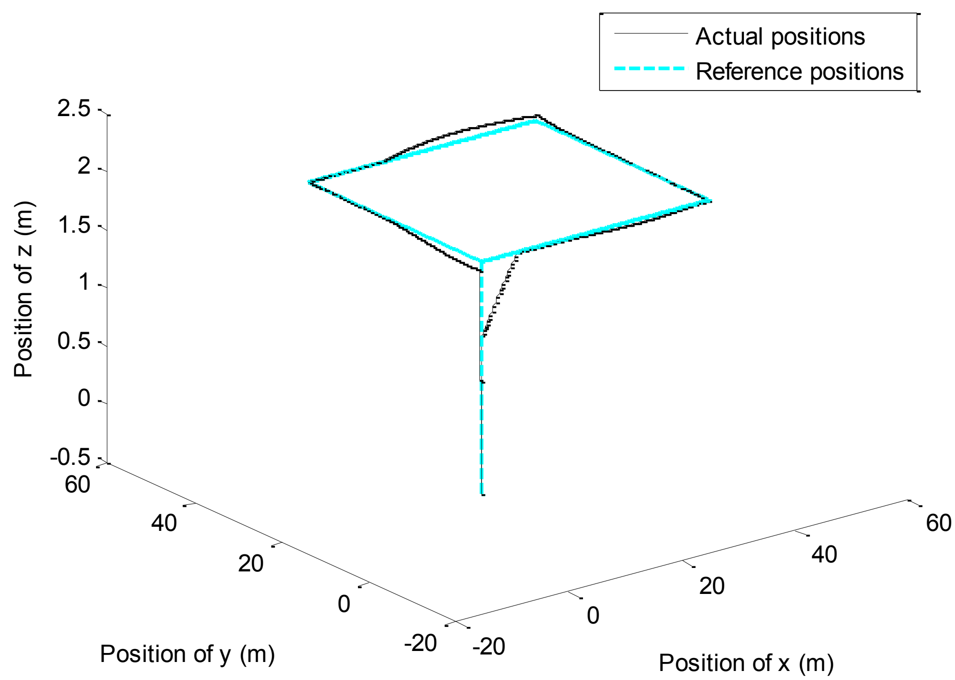

Fig. 4-31. 3-dimensional path tracking

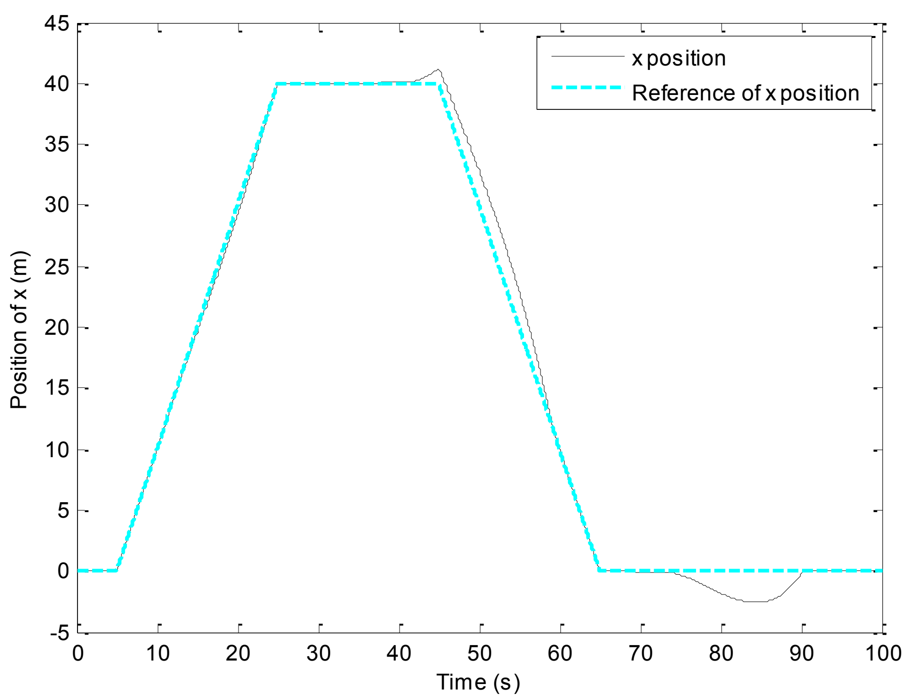

Fig. 4-32. Position tracking in *x* direction

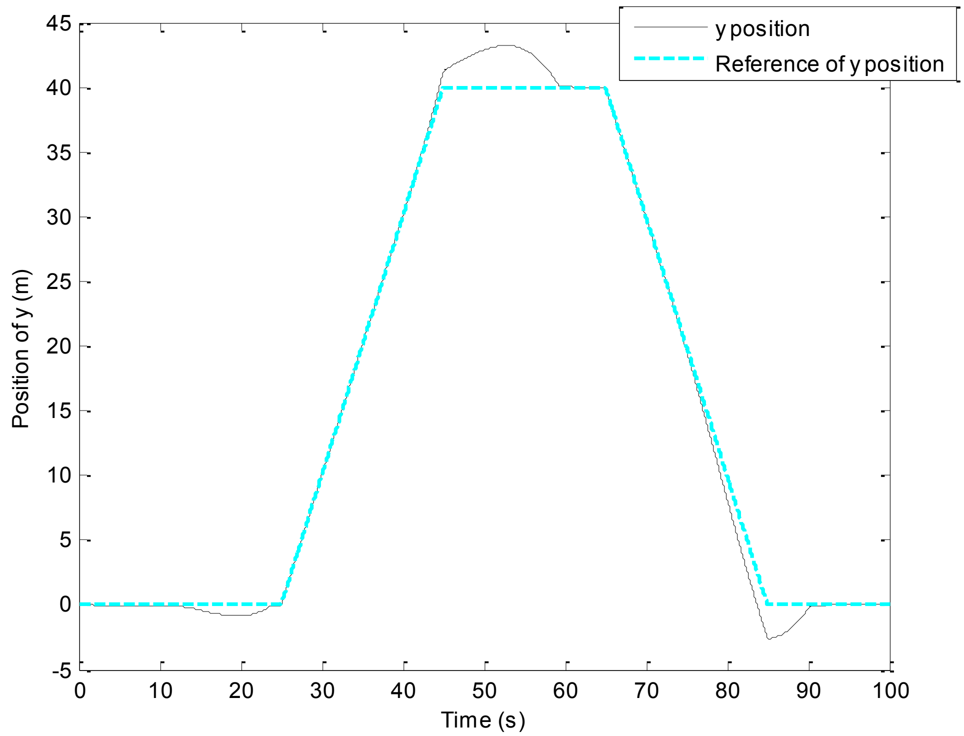

Fig. 4-33. Position tracking in y direction

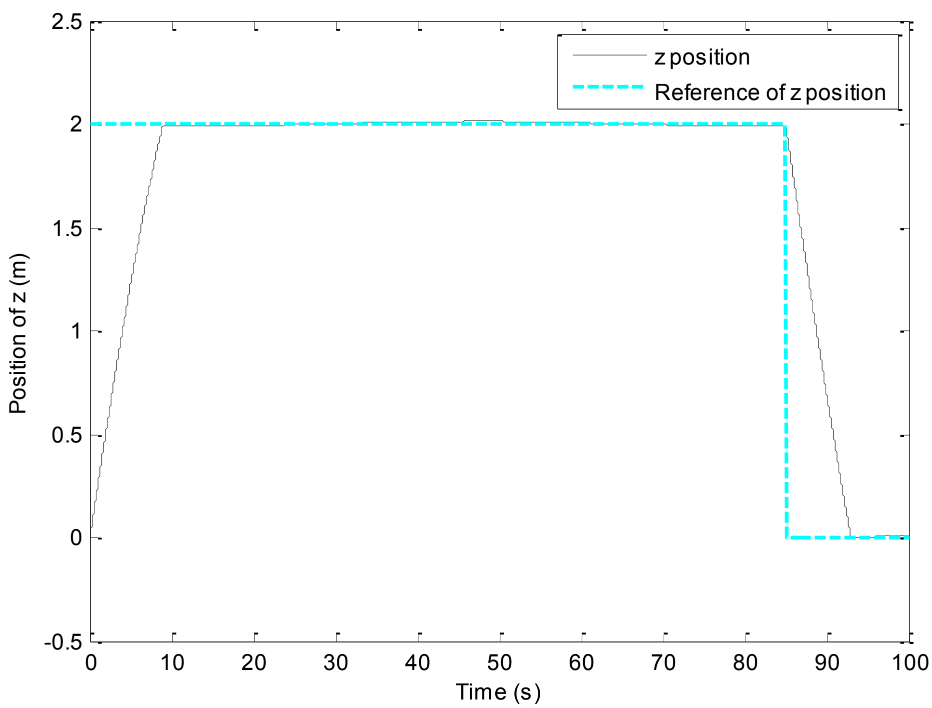

Fig. 4-34. Position tracking in z direction

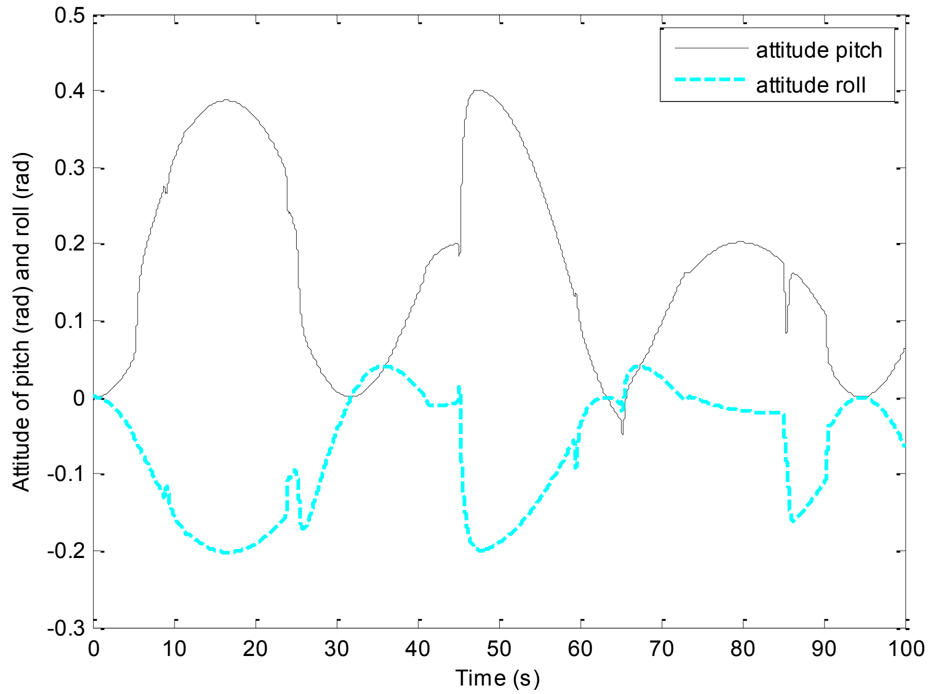

Fig. 4-35. Attitude of pitch and roll angles

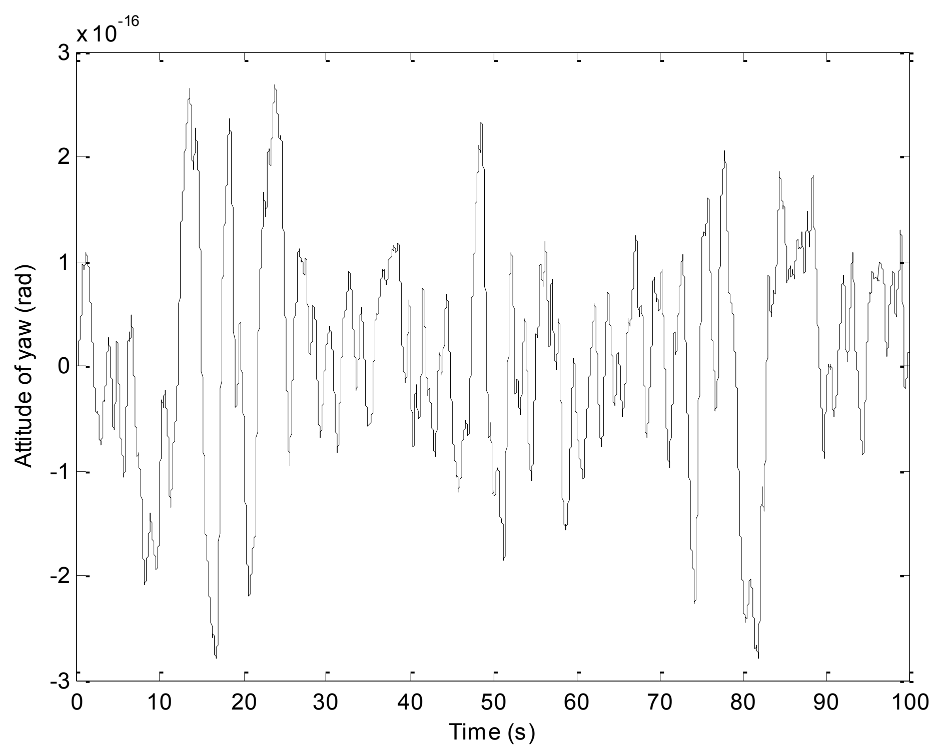

Fig. 4-36. Attitude of yaw angle

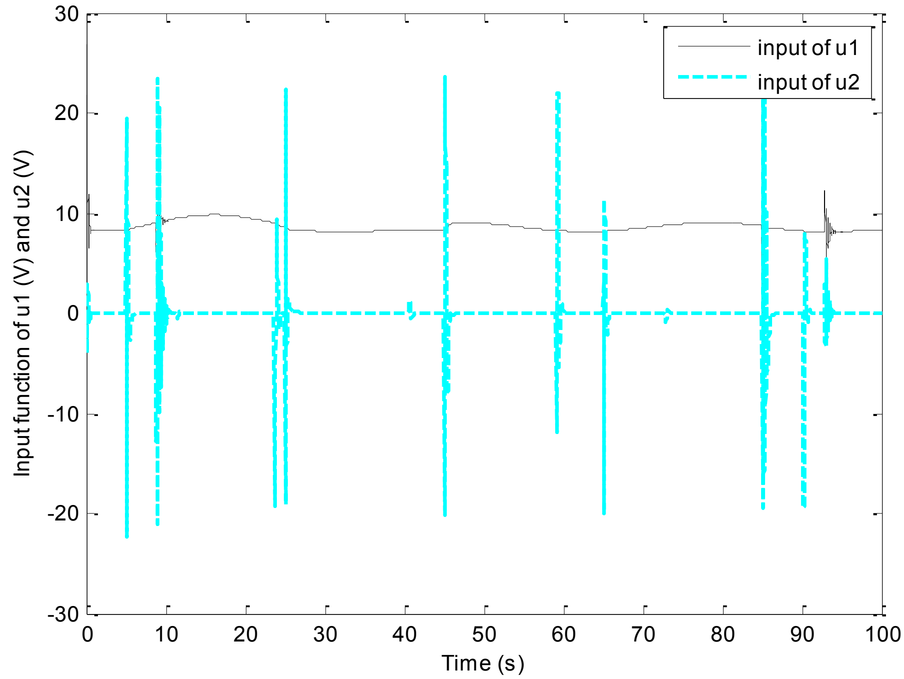

Fig. 4-37. Control inputs of u_1 and u_2

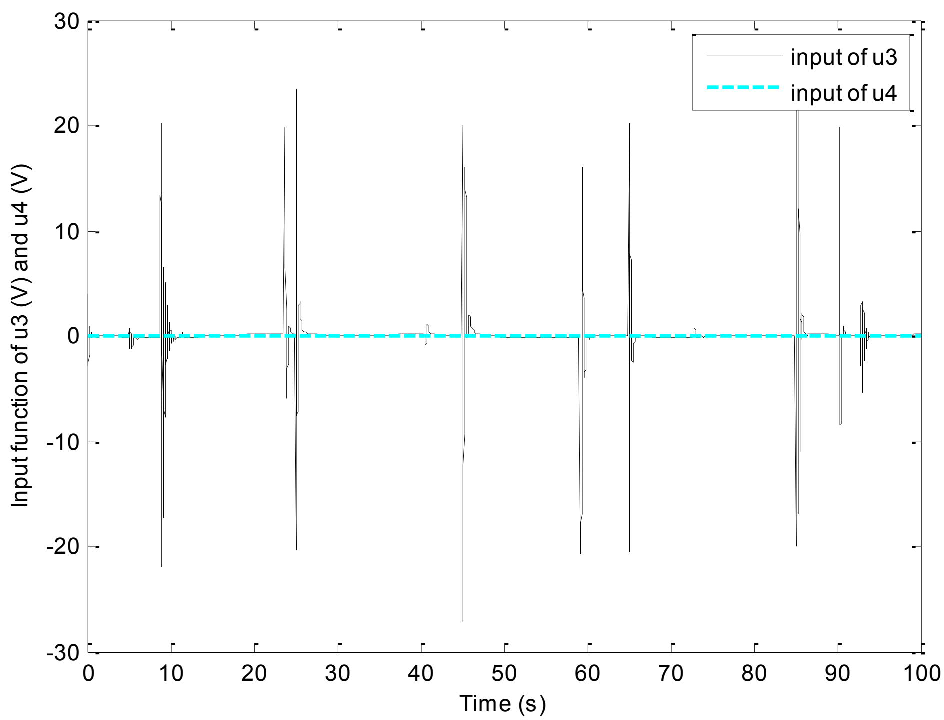

Fig. 4-38. Control inputs of u_3 and u_4

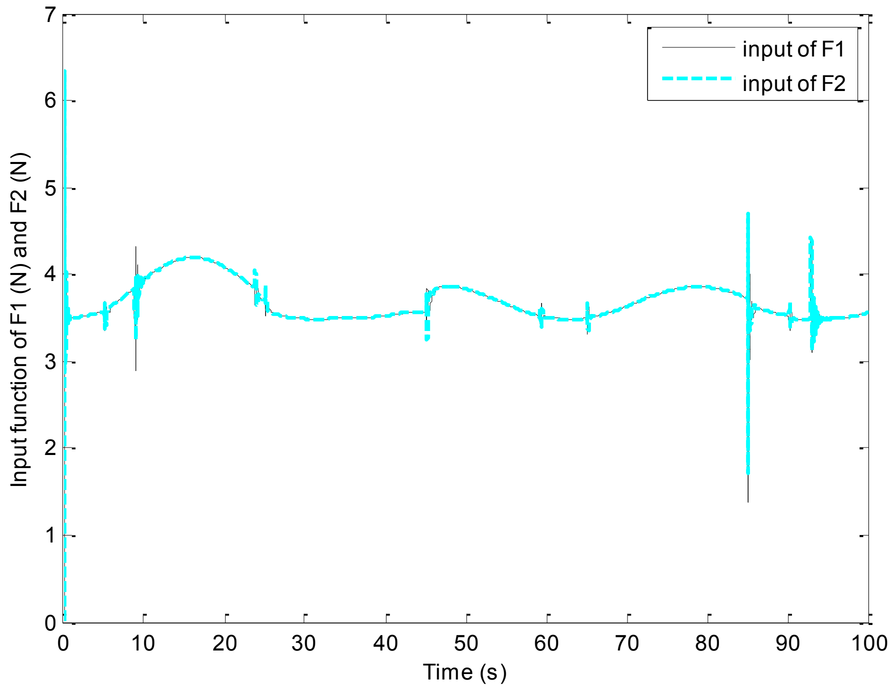

Fig. 4-39. Propellers forces F_1 and F_2

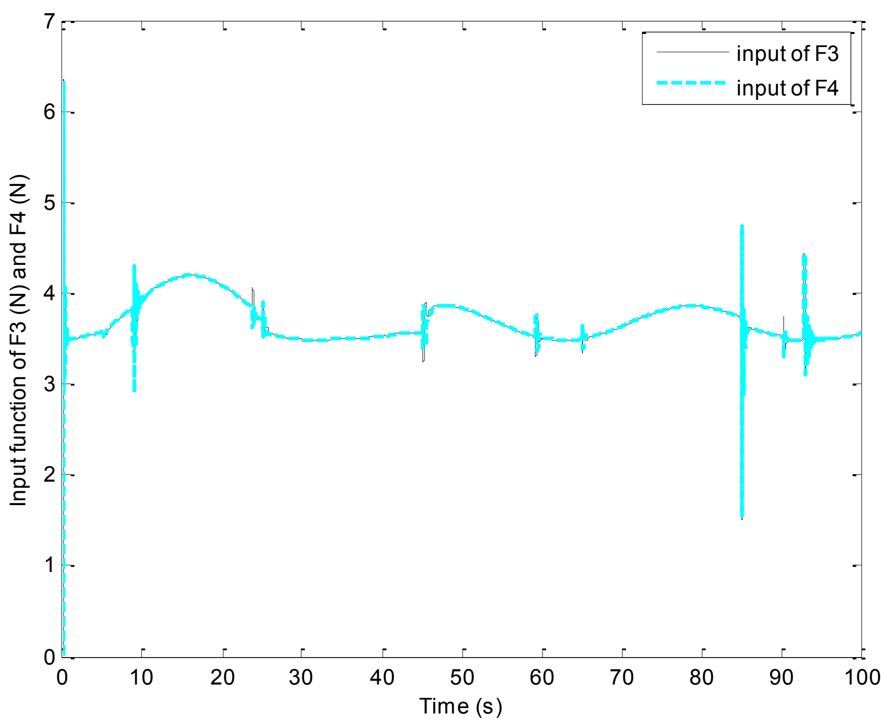

Fig. 4-40. Propellers forces F_3 and F_4

4.3. Backstepping Control

4.3.1. Controller Design

Similar to both feedback linearization control and sliding mode control, by equations (2-62)–(2-66), a multiple-input multiple-output (MIMO) controller on both position and attitude control can be implemented.

Based on the principles of backstepping and the model of the Qball-X4, equation (3-24) has not enough states for a back stepping control. Hence, state expansion (4-13) is used.

For position x, the state equations are represented by:

$$\phi\sin\theta\cos\psi + \sin\phi\sin\psi) - d_x x_2 \tag{4-26}$$

x_1 needs to be stabilized first and then x_2. Defining a tracking error to change the system into a tracking task, $e_{x1} = x_d - x$ is used to track the first state x_1 with the desired value. Lyapunov function is then chosen as:

$$V(e_{x1}) = \frac{1}{2}e_{x1}^2 \tag{4-27}$$

then,

(4-28)

where x_2 is the virtual control input.

Defining

$$\phi(x) = x_2 = \tag{4-29}$$

the above equation (4-28) can be rewritten as:

$$\dot{V} \quad (\quad \alpha_1 e_{x1}^2 \le 0 \tag{4-30}$$

Therefore, e_{x1} has been stabilized. Defining a second tracking error as $e_{x2} = x_2 - \phi(x)$, Lyapunov function needs to be augmented.

$$V(e_{x1}, e_{x2}) = \frac{1}{2}(e_{x1}{}^2 + e_{x2}{}^2) \tag{4-31}$$

Similarly, from $e_{x2} = x_2 - \phi(x)$, it can be derived as follows:

$$\begin{aligned} e_{x2} &= x_2 - \phi(x) = x_2 - \\ & \quad -\alpha_1 e_{x1} \end{aligned} \tag{4-32}$$

$$\begin{aligned} \dot{V} \quad &= e_{x1} \\ &= -e_{x1}e_{x2} - \alpha_1 e_{x1}^2 + e_{x2}(u_1 u_x - d_x x_2 - \\ &= -e_{x1}e_{x2} - \alpha_1 e_{x1}^2 + e_{x2}(u_1 u_x - d_x x_2 - \quad e_{x2} + \alpha_1 e_{x1})) \end{aligned} \tag{4-33}$$

In order to have a negative Lyapunov function, virtual control input u_x has been chosen as:

$$u_x = \frac{1}{u_1}(e_{x1} + \quad e_{x2} + \alpha_1 e_{x1}) + d_x x_2 - \alpha_2 e_{x2}) \tag{4-34}$$

Replace equation (4-34) into (4-33), the chosen Lyapunov function can be proven as a negative function and the two states x_1, x_2 are stable.

$$\begin{aligned} \dot{V} \quad &= e_{x1} \\ &= -e_{x1}e_{x2} - \alpha_1 e_{x1}^2 + e_{x2}(u_1 u_x - d_x x_2 - \\ &= -e_{x1}e_{x2} - \alpha_1 e_{x1}^2 + e_{x2}(u_1 \frac{1}{u_1}(e_{x1} + \quad e_{x2} + \alpha_1 e_{x1}) + d_x x_2 - \alpha_2 e_{x2}) \\ & \quad - d_x x_2 - \quad e_{x2} + \alpha_1 e_{x1})) \\ &= -\alpha_1 e_{x1}^2 - \alpha_2 e_{x2}^2 \le 0 \end{aligned} \tag{4-35}$$

Similar procedure for position y with the states x_3, x_4. First of all, define a tracking error $e_{y1} = y_d - y,$ and a Lyapunov function as $V(e_{y1}) = \frac{1}{2}e_{y1}^2$. Then the following can be attained easily:

$$\dot{V} \qquad {}_1(\qquad \cdot \qquad \cdot \qquad \cdot \tag{4-36}$$

where $\phi(y) = x_4 = \dot{}$ is the virtual control input. By choosing an augmented Lyapunov function

$$V(e_{y1}, e_{y2}) = \frac{1}{2}(e_{y1}{}^2 + e_{y2}{}^2) \tag{4-37}$$

where $e_{y2} = x_4 - \phi(y)$, and $\dot{}$ $-\alpha_3 e_{y1}$, so that

$$\begin{aligned}
\dot{V} \quad &= e_{y1}\dot{} \qquad \cdot \qquad \cdot \\
&= -e_{y1}e_{y2} - \alpha_3 e_{y1}^2 + e_{y2}(u_1 u_y - d_y x_4 - \ddot{} \qquad \cdot \\
&= -e_{y1}e_{y2} - \alpha_3 e_{y1}^2 + e_{y2}(u_1 u_y - d_y x_4 - \ddot{} \qquad e_{y2} + \alpha_3 e_{y1})) \\
&= -e_{y1}e_{y2} - \alpha_3 e_{y1}^2 + e_{y2}(u_1 \frac{1}{u_1}(e_{y1} + \ddot{} \qquad e_{y2} + \alpha_3 e_{y1}) + d_y x_4 - \alpha_4 e_{y2}) \\
&\quad - d_y x_4 - \ddot{} \qquad e_{y2} + \alpha_3 e_{y1})) \\
&= -\alpha_3 e_{y1}^2 - \alpha_4 e_{y2}^2 \\
&\le 0
\end{aligned} \tag{4-38}$$

with the virtual control input u_y as:

$$u_y = \frac{1}{u_1}(e_{y1} + \ddot{} \qquad e_{y2} + \alpha_3 e_{y1}) + d_y x_4 - \alpha_4 e_{y2}) \tag{4-39}$$

For the rest of states z, θ, ϕ, ψ, defining the tracking errors as:

$$\begin{aligned}
e_{z1} &= z_d - z \qquad & e_{\theta 1} &= \theta_d - \theta \\
e_{\phi 1} &= \phi_d - \phi \qquad & e_{\psi 1} &= \psi_d - \psi
\end{aligned} \tag{4-40}$$

and Lyapunov functions as:

$$
\begin{aligned}
V(e_{z1}) &= \frac{1}{2}e_{z1}^2 \\
V(e_{\theta 1}) &= \frac{1}{2}e_{\theta 1}^2 \\
V(e_{\phi 1}) &= \frac{1}{2}e_{\phi 1}^2 \\
V(e_{\psi 1}) &= \frac{1}{2}e_{\psi 1}^2
\end{aligned}
\qquad (4\text{-}41)
$$

all the virtual control inputs need to be chosen as:

$$
\begin{aligned}
\phi(z) &= x_6 = \qquad & \phi(\theta) &= x_8 = \\
\phi(\phi) &= x_{10} = \qquad & \phi(\psi) &= x_{12} =
\end{aligned}
\qquad (4\text{-}42)
$$

Augmented Lyapunov functions are

$$
\begin{aligned}
V(e_{z1}, e_{z2}) &= \frac{1}{2}(e_{z1}{}^2 + e_{z2}{}^2) \qquad && = e_{z1} \\
V(e_{\theta 1}, e_{\theta 2}) &= \frac{1}{2}(e_{\theta 1}{}^2 + e_{\theta 2}{}^2) \qquad && = e_{\theta 1} \\
V(e_{\phi 1}, e_{\phi 2}) &= \frac{1}{2}(e_{\phi 1}{}^2 + e_{\phi 2}{}^2) \qquad && = e_{\phi 1} \\
V(e_{\psi 1}, e_{\psi 2}) &= \frac{1}{2}(e_{\psi 1}{}^2 + e_{\psi 2}{}^2) \qquad && = e_{\psi 1}\;{}_{\psi 1} + e_{\psi 2}
\end{aligned}
\qquad (4\text{-}43)
$$

where $e_{z2} = x_6 - \phi(z)$, $e_{\theta 2} = x_8 - \phi(\theta)$, $e_{\phi 2} = x_{10} - \phi(\phi)$, and $e_{\psi 2} = x_{12} - \phi(\psi)$, then the control inputs are:

$$
\begin{aligned}
u_1 &= \frac{1}{\cos\phi\cos\theta}(e_{z1} + \quad {}_{z2} + \alpha_5 e_{z1}) + g + d_z x_6 - \alpha_6 e_{z2}) \\
u_2 &= \frac{1}{l}(e_{\theta 1} + \quad {}_{\theta 2} + \alpha_7 e_{\theta 1}) - \frac{J_y - J_z}{J_x} x_{10} x_{12} + \frac{J_r x_{10} \Omega}{J_x} + d_\theta x_8 - \alpha_8 e_{\theta 2}) \\
u_3 &= \frac{1}{l}(e_{\phi 1} + \quad {}_{\phi 2} + \alpha_9 e_{\phi 1}) - \frac{J_z - J_x}{J_y} x_8 x_{12} - \frac{J_r x_8 \Omega}{J_y} + d_\phi x_{10} - \alpha_{10} e_{\phi 2}) \\
u_4 &= \frac{1}{c}(e_{\psi 1} + \quad e_{\psi 2} + \alpha_{11} e_{\psi 1}) - \frac{J_x - J_y}{J_z} x_8 x_{10} + d_\psi x_{12} - \alpha_{12} e_{\psi 2})
\end{aligned}
\qquad (4\text{-}44)
$$

where $\alpha_i \;\; i = 1, \ldots$ are all the positive control gains.

4.3.2. Simulations

As similar as previous sections, setting all the drag forces and desired yaw angle to zeros, simulations without disturbance ($\Omega = 0$) or noise are shown as follows. The behaviours of positions, attitudes, and control inputs can be seen respectively in Figures 4-41 to 4-50.

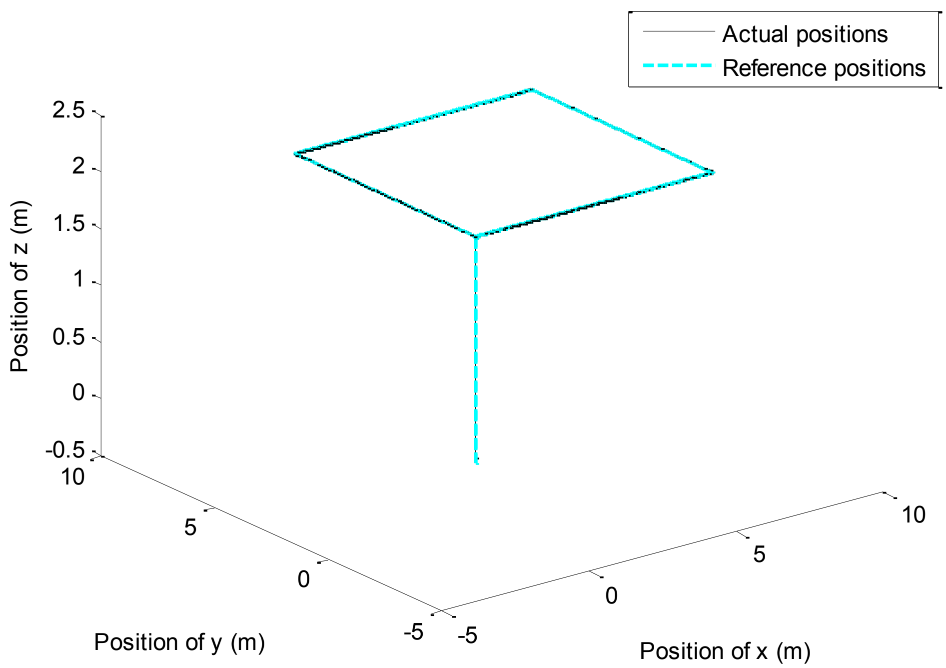

Fig. 4-41. 3-dimensional path tracking

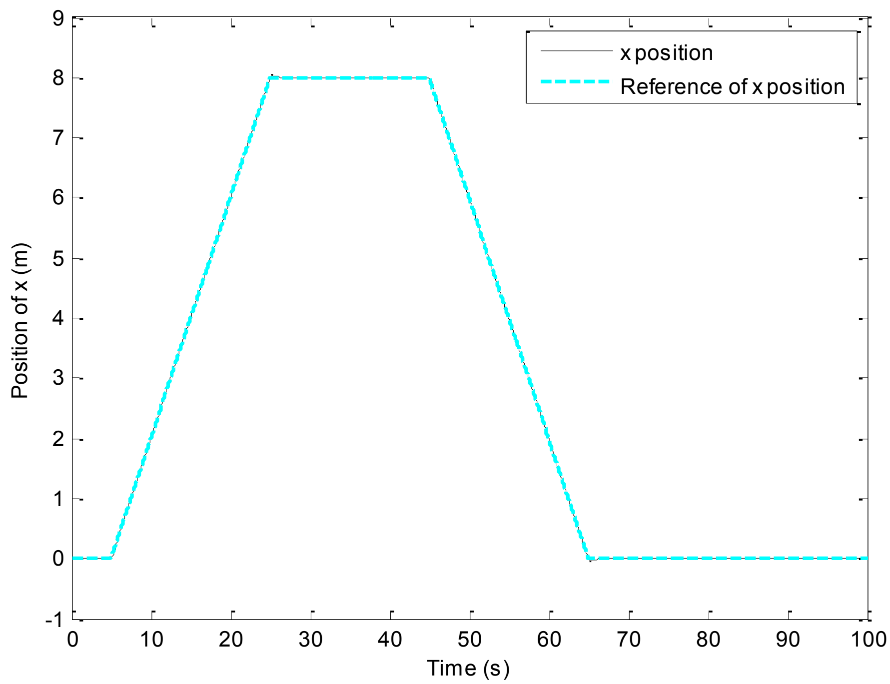

Fig. 4-42. Position tracking in *x* direction

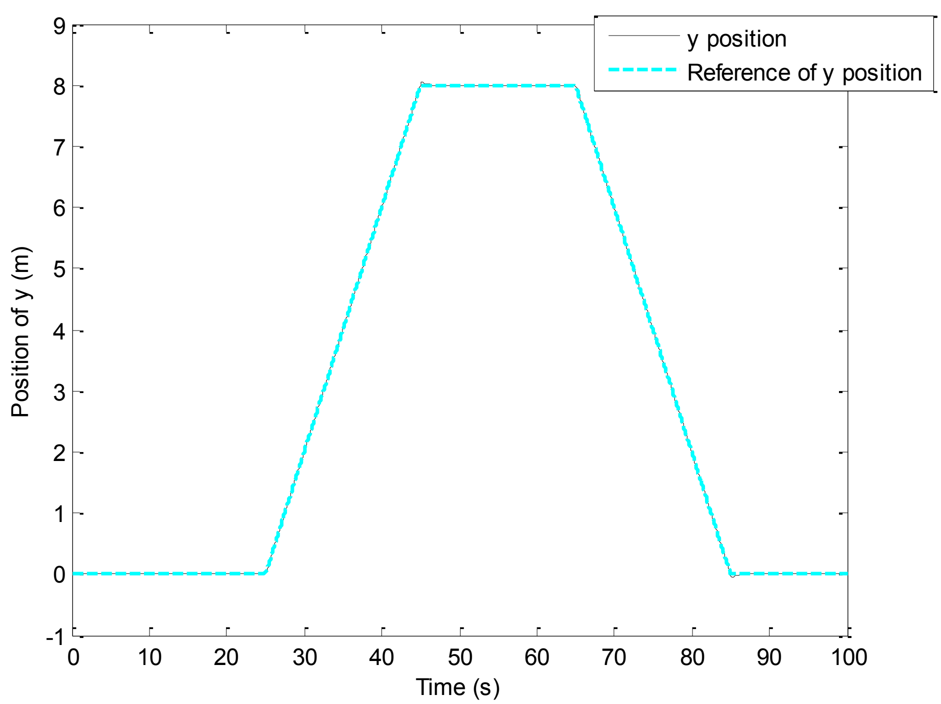

Fig. 4-43. Position tracking in *y* direction

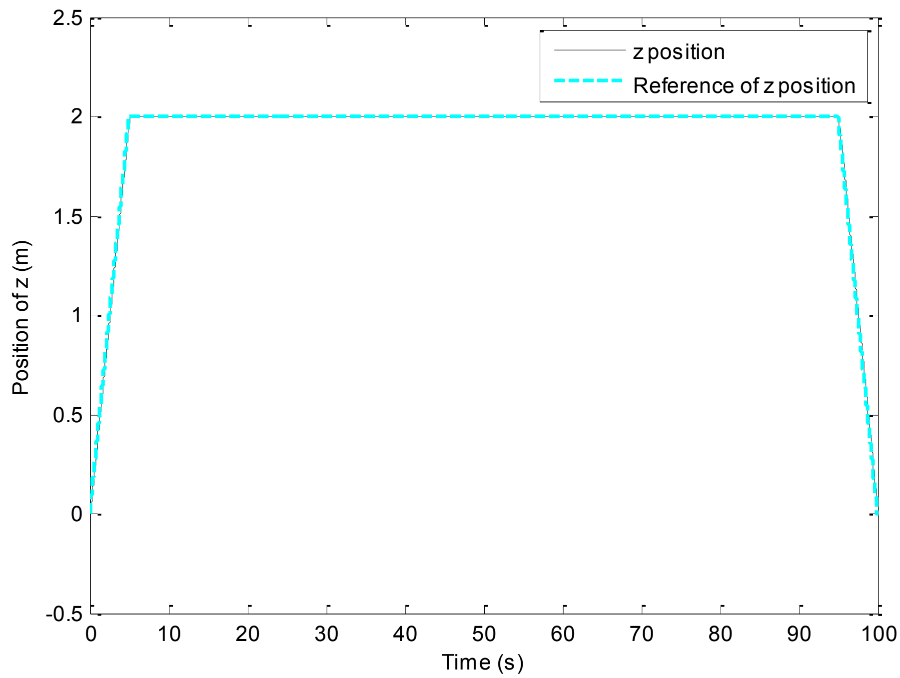

Fig. 4-44. Position tracking in z direction

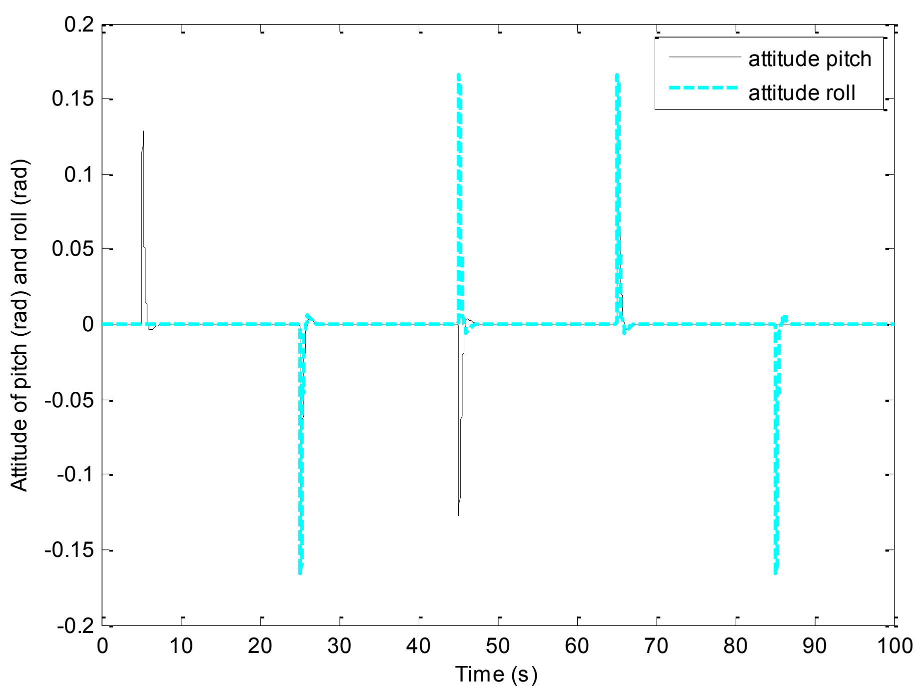

Fig. 4-45. Attitude of pitch and roll angles

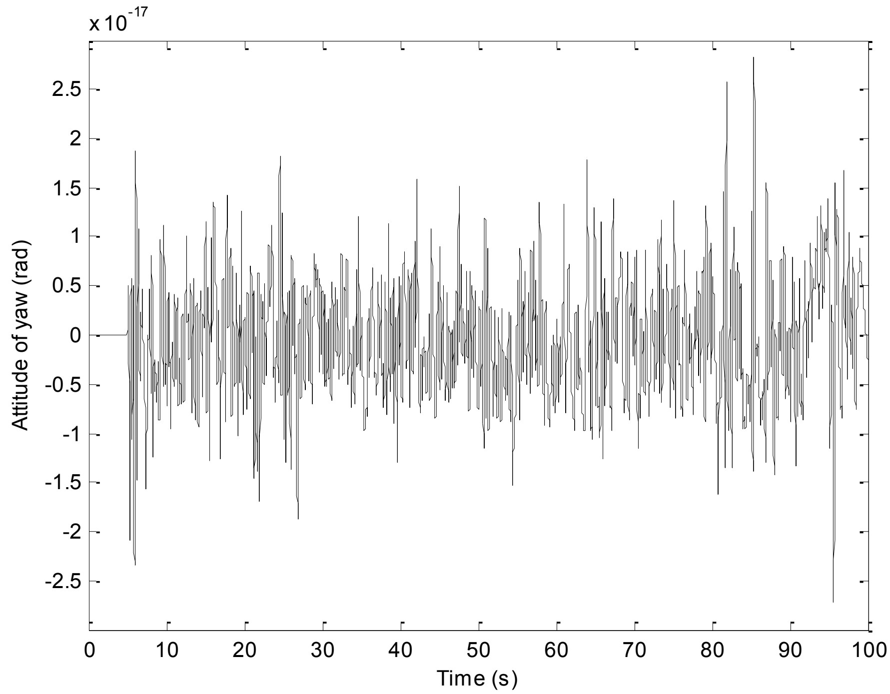

Fig. 4-46. Attitude of yaw angle

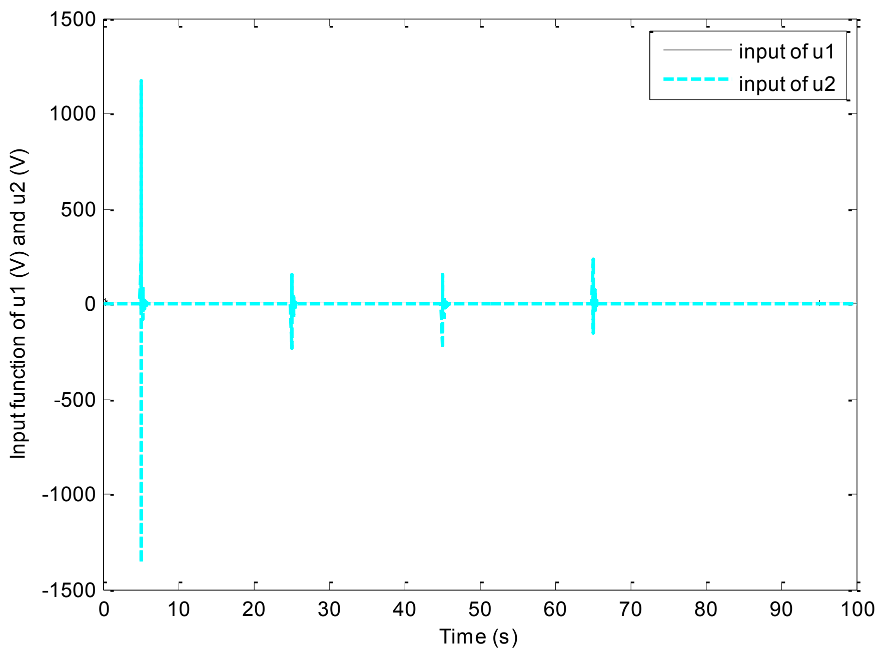

Fig. 4-47. Control inputs of u_1 and u_2

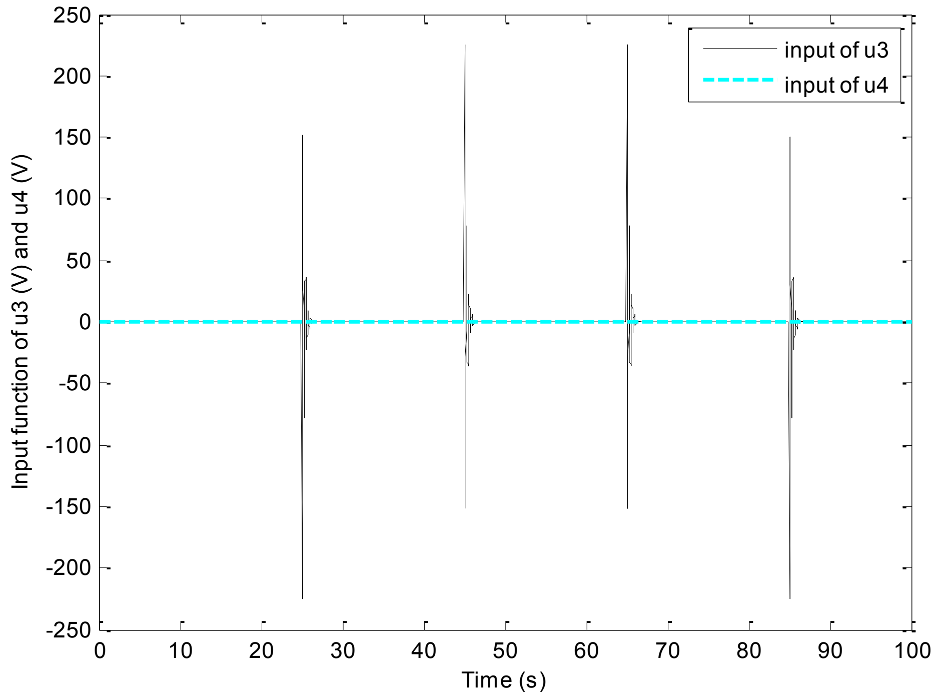

Fig. 4-48. Control inputs of u_3 and u_4

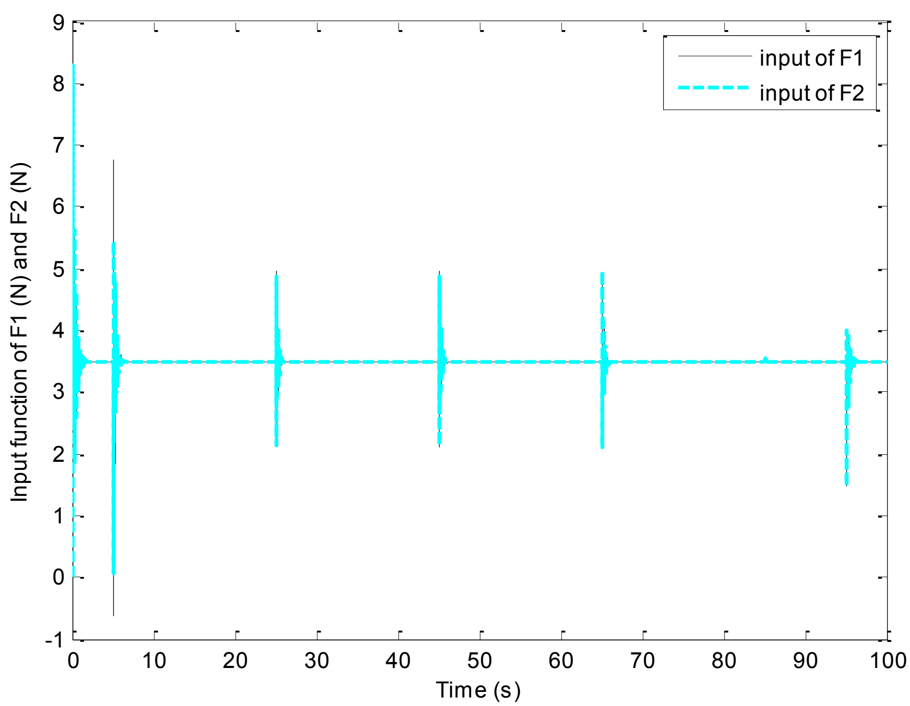

Fig. 4-49. Propellers forces F_1 and F_2

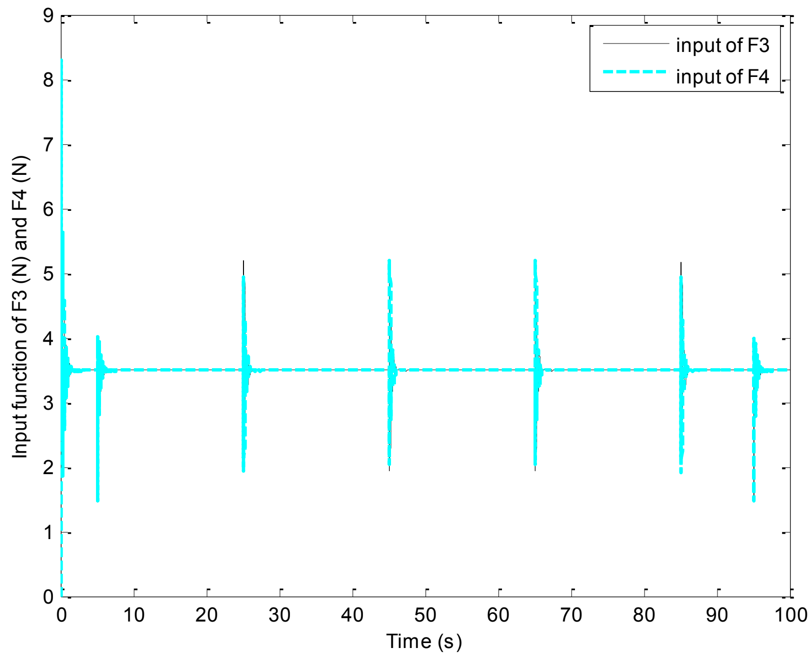

Fig. 4-50. Propellers forces F_3 and F_4

The same controller is used to test the robustness of the controller, while noise and disturbance are added. Simulation results have been shown in Figures 4-51 to 4-60.

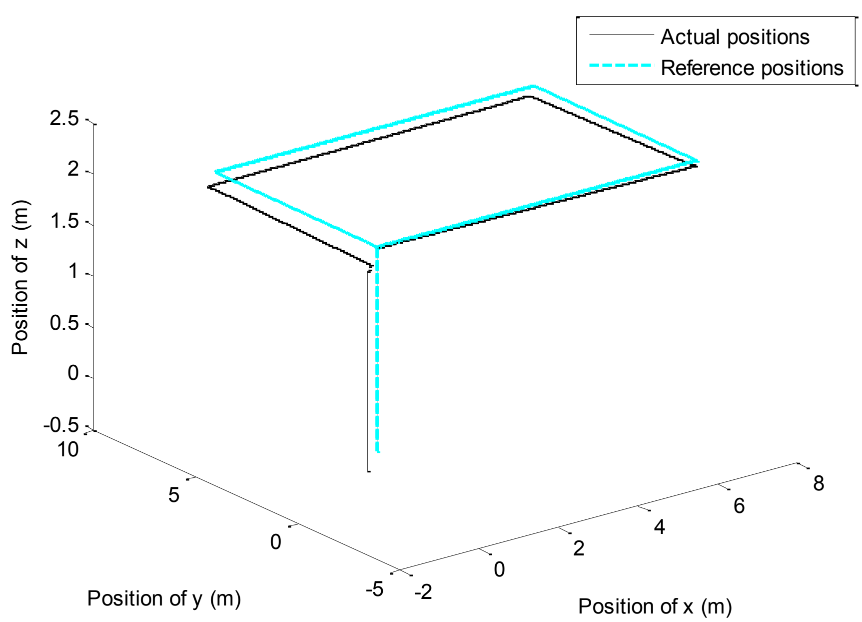

Fig. 4-51. 3-dimensional path tracking

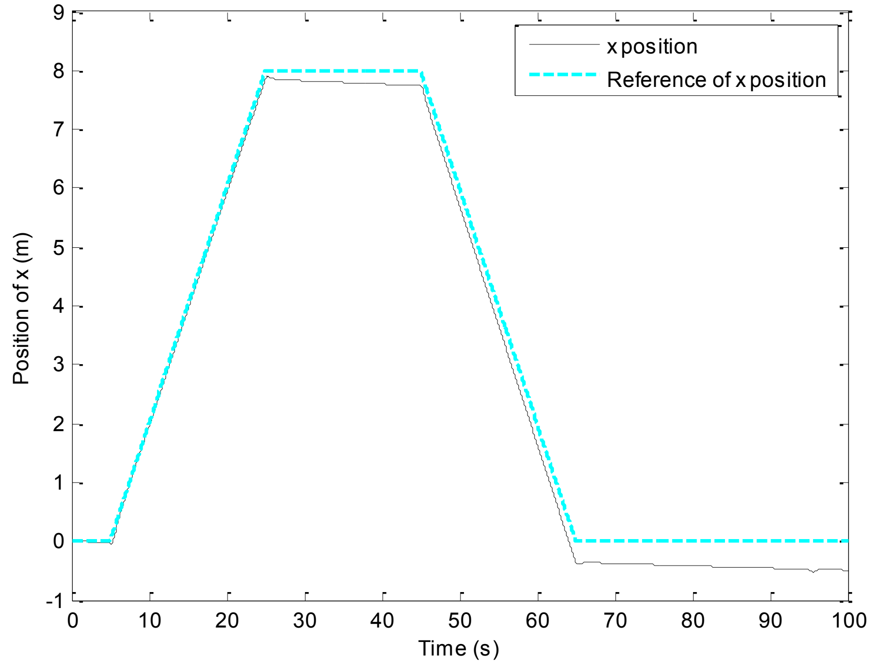

Fig. 4-52. Position tracking in x direction

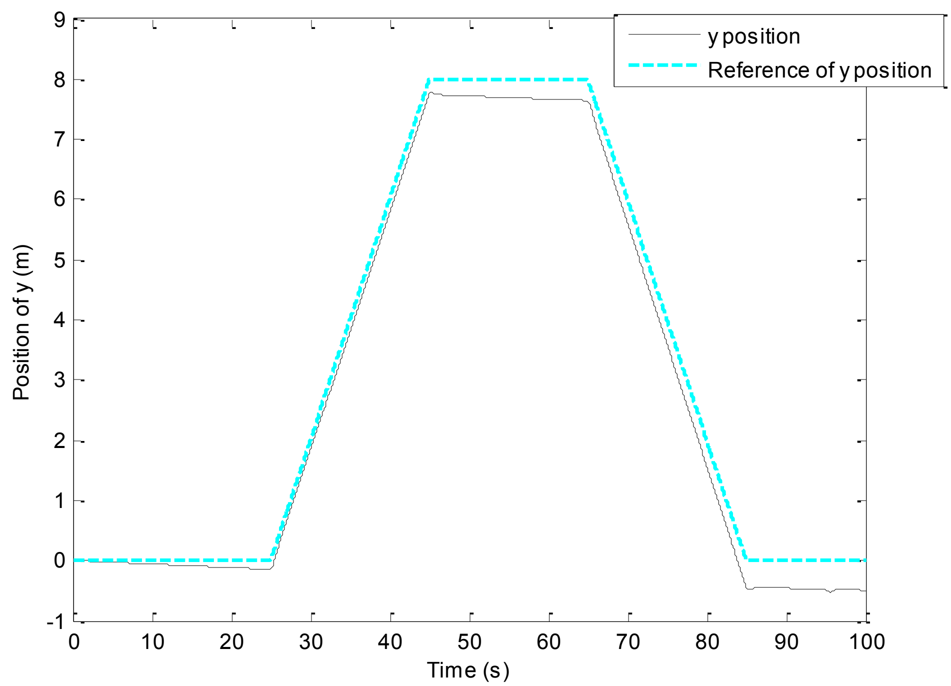

Fig. 4-53. Position tracking in y direction

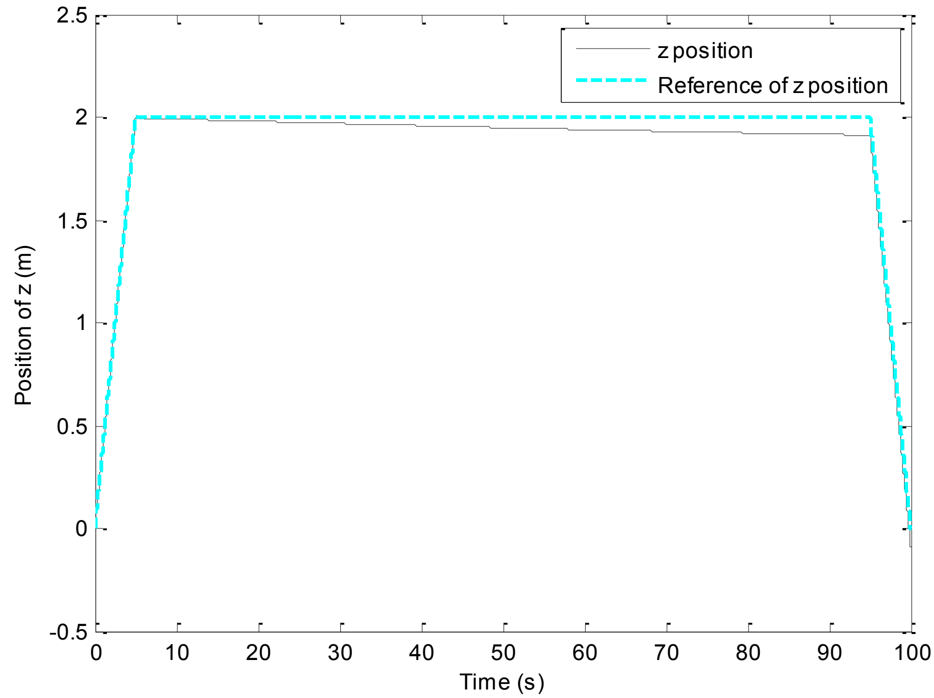

Fig. 4-54. Position tracking in *z* direction

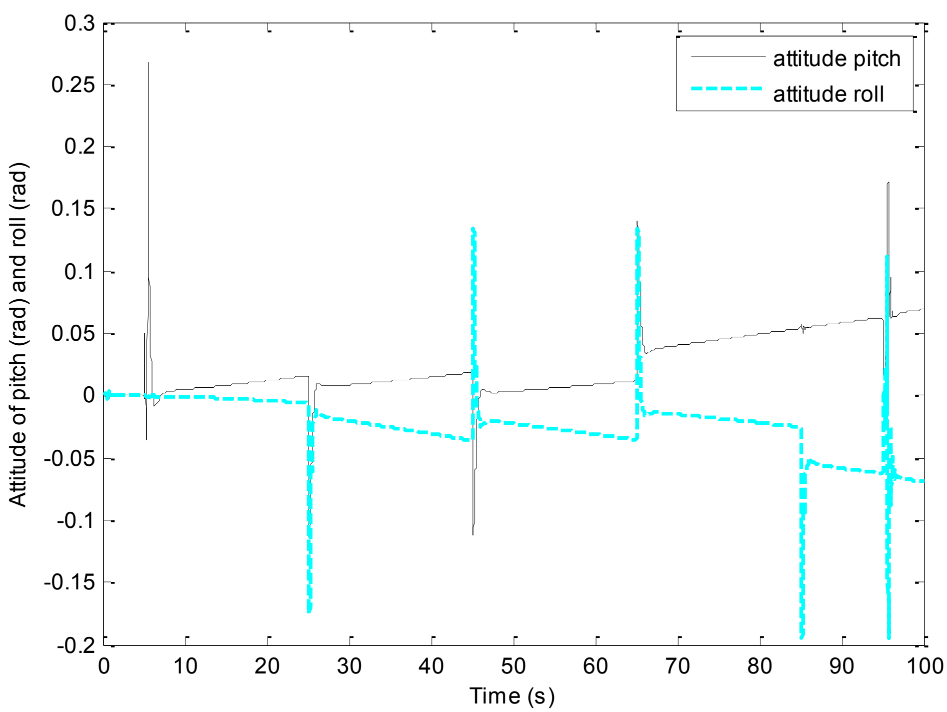

Fig. 4-55. Attitude of pitch and roll angles

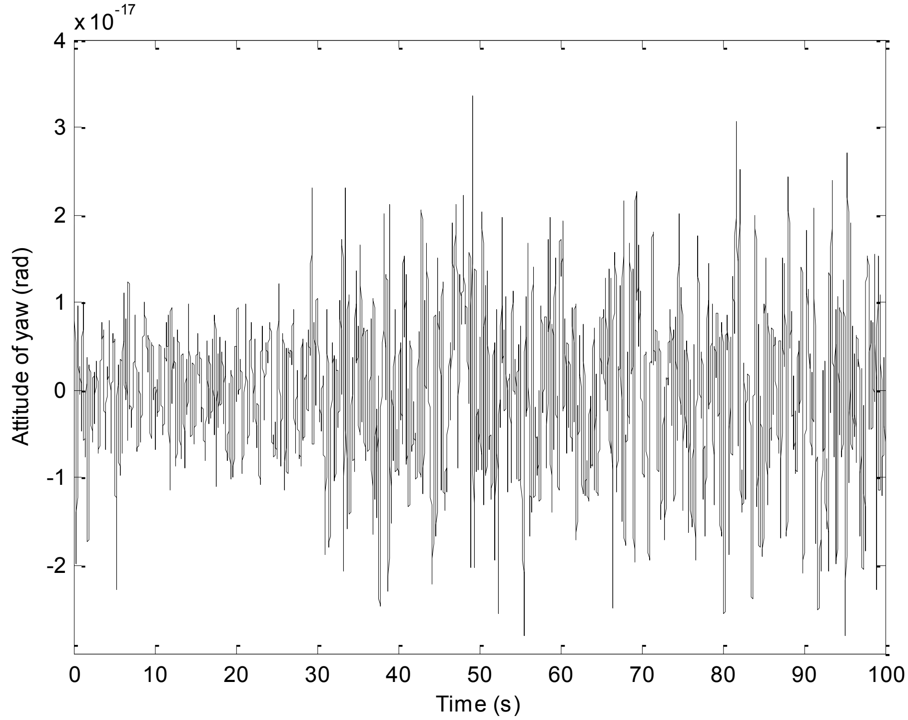

Fig. 4-56. Attitude of yaw angle

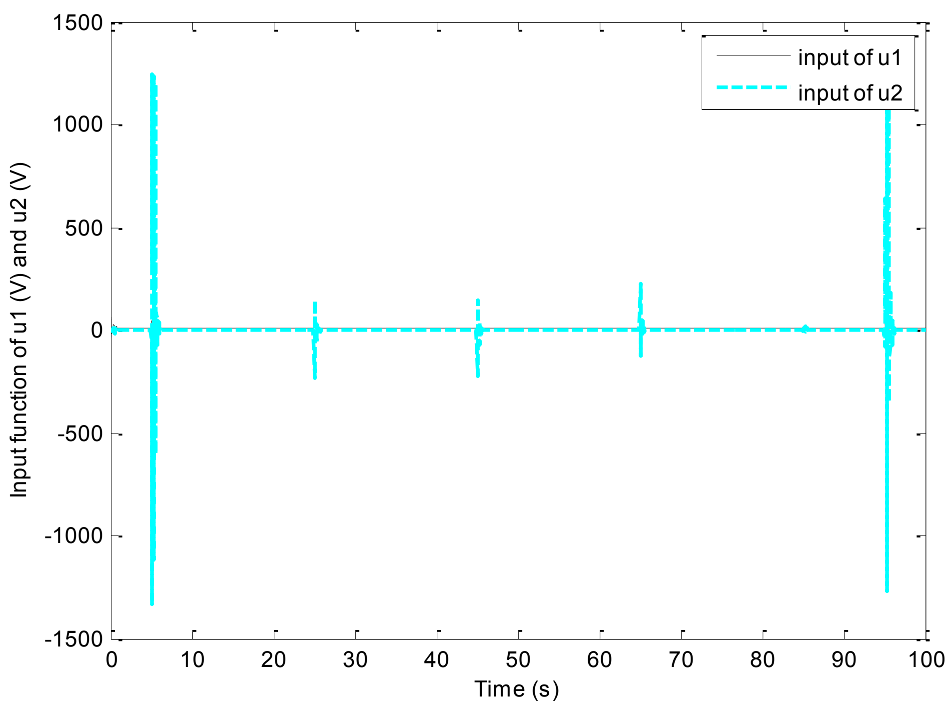

Fig. 4-57. Control inputs of u_1 and u_2

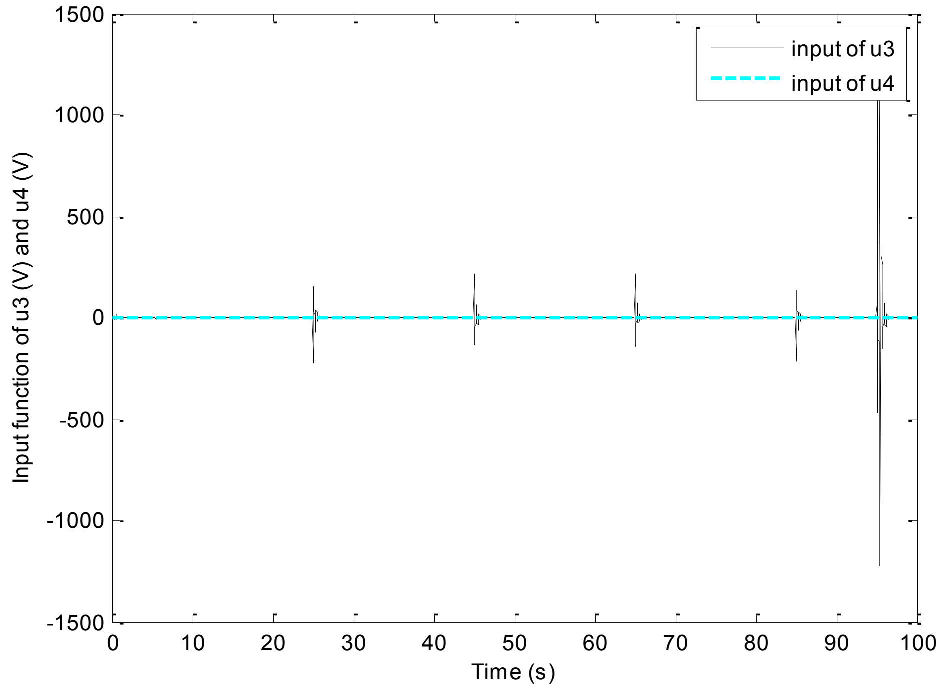

Fig. 4-58. Control inputs of u_3 and u_4

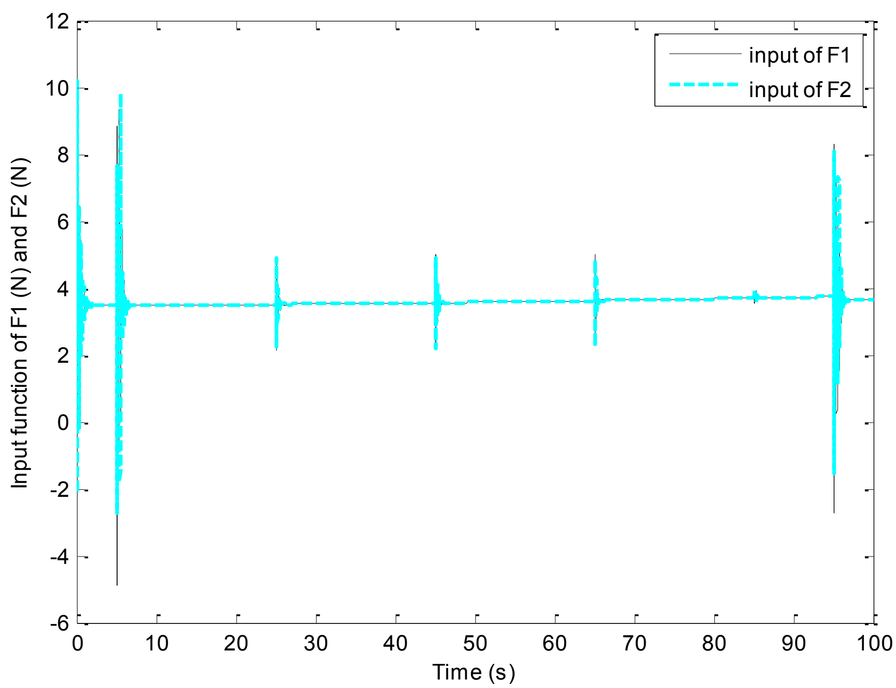

Fig. 4-59. Propellers forces F_1 and F_2

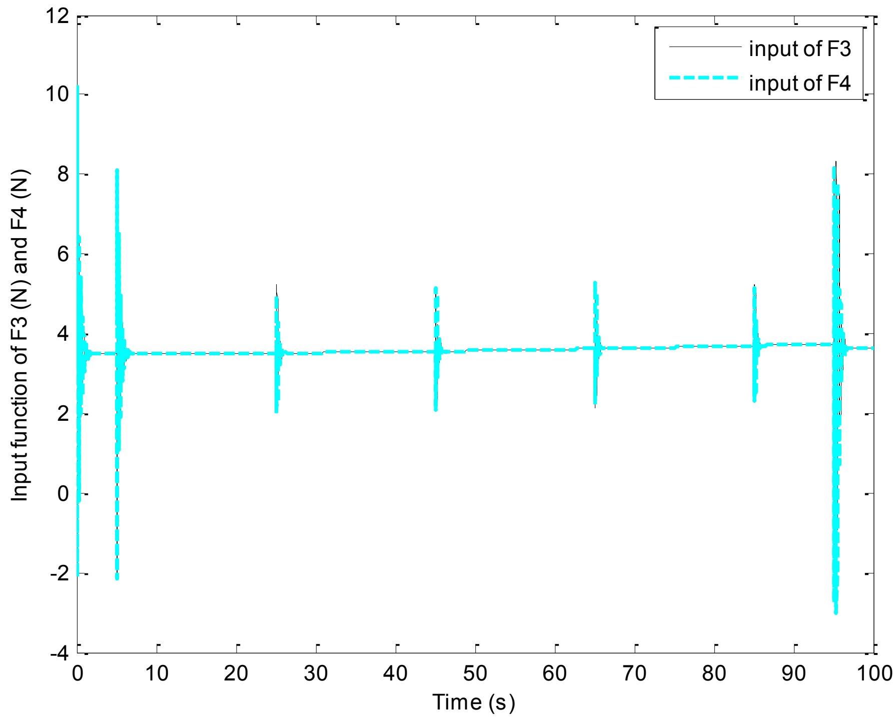

Fig. 4-60. Propellers forces F_3 and F_4

From all the results shown above, it can be learnt that backstepping controller behaves properly, but differently from feedback linearization control and sliding mode control. If the control inputs need to be in a certain range, backstepping control needs a trajectory with a slow speed. Since BSC tends to generate a larger control input to achieve fast convergence, tracking performance becomes deteriorated when the reference trajectory has some critical points (derivatives undefined) or changes quickly. After noises being added into the system, the control capability has been worsen, but the control effort is still being made by the backstepping controller.

4.4. Experimental Testing Results

The experimental tests are carried out in the Networked Autonomous Vehicles (NAV) Lab at the Concordia University. The experimental setup includes six cameras playing as the GPS system, a joystick as the safety control, and a desktop as the ground station as mentioned in Chapter 3. The six cameras are mounted on the lab ceiling to have a better 3-dimensional position feedback of the Qball-X4 UAV. The sensors, gyroscope, accelerometer, and magnetometer installed on the Qball-X4 system send back the status of vehicle during real time flight. When all the necessary states of Qball-X4 are received for the controller on ground station through TCP/IP wireless connection, the control inputs will be generated from the ground station and sent to the Qball-X4 system. The process is then complete.

4.4.1. Feedback Linearization Control

Figures 4-61 to 4-70 show the performance of feedback linearization controller in the real flight tests. The tracking task is still to follow a square as in the simulations. However, due to the size of the lab, the desired square has been reduced to $1.5\times1.5m^2$.

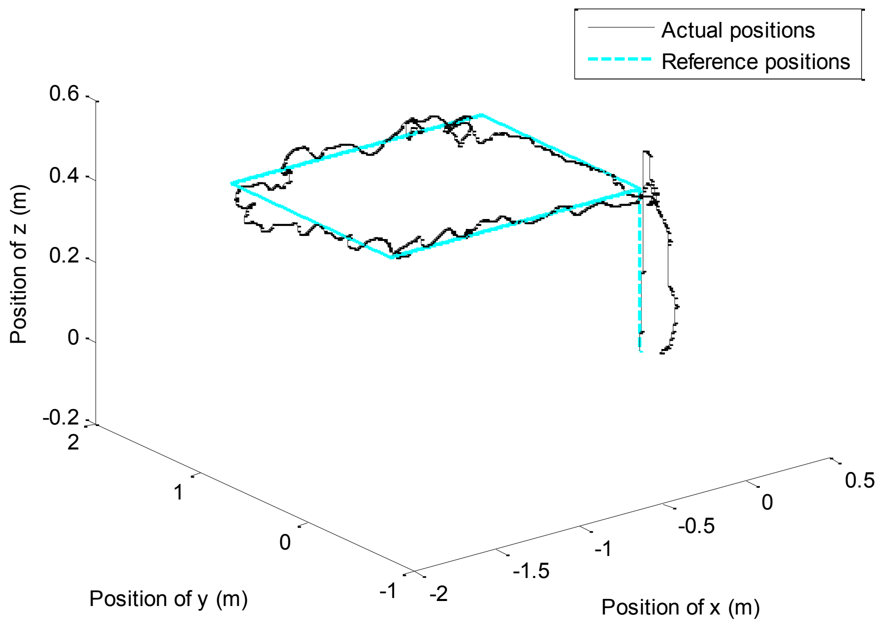

Fig. 4-61. 3-dimensional path tracking

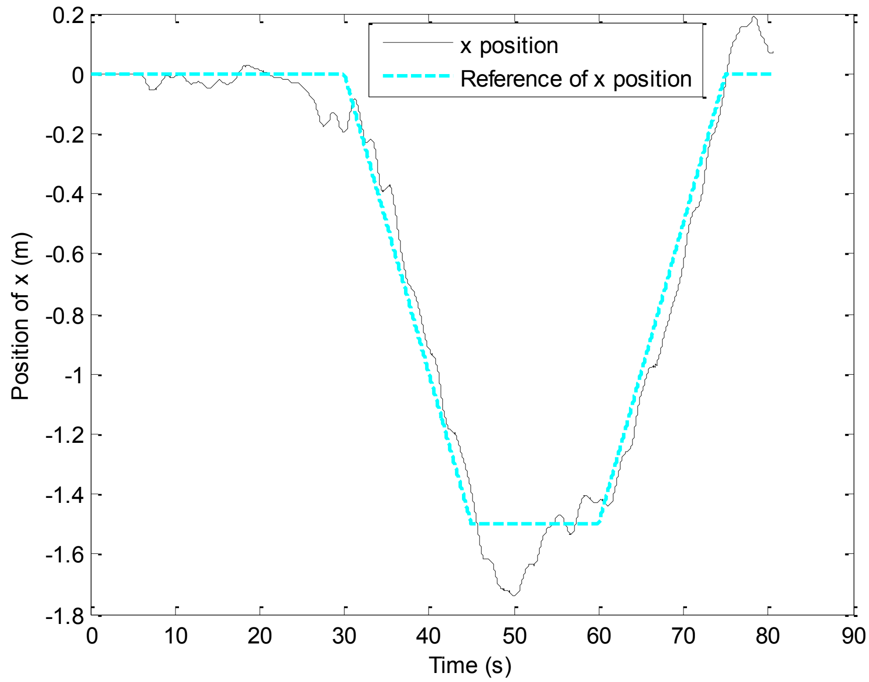

Fig. 4-62. Position tracking in *x* direction

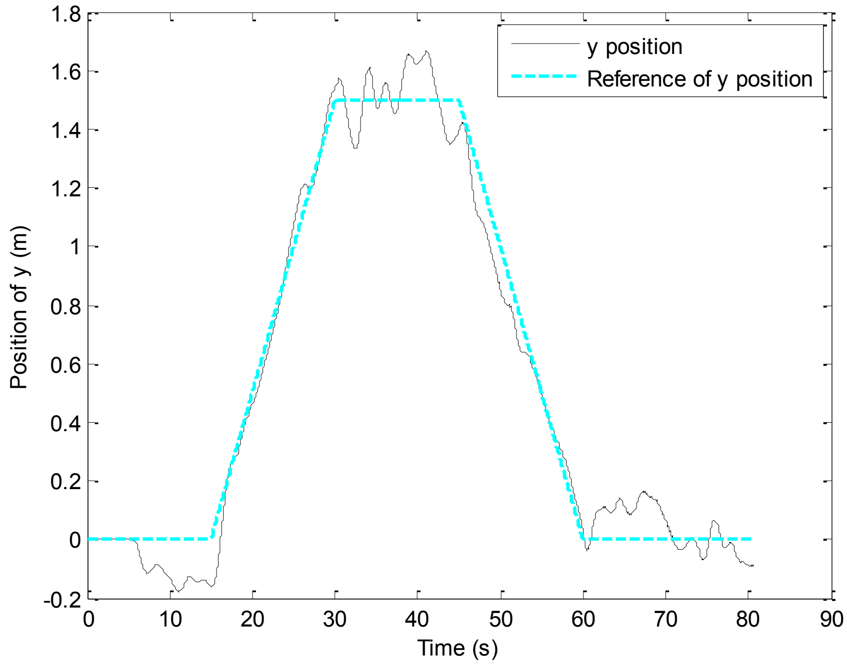

Fig. 4-63. Position tracking in *y* direction

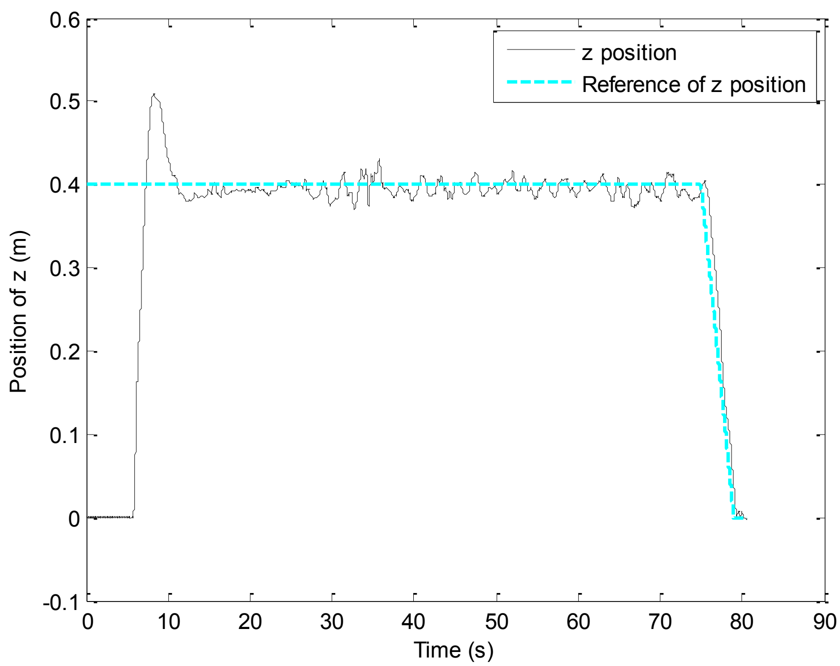

Fig. 4-64. Position tracking in *z* direction

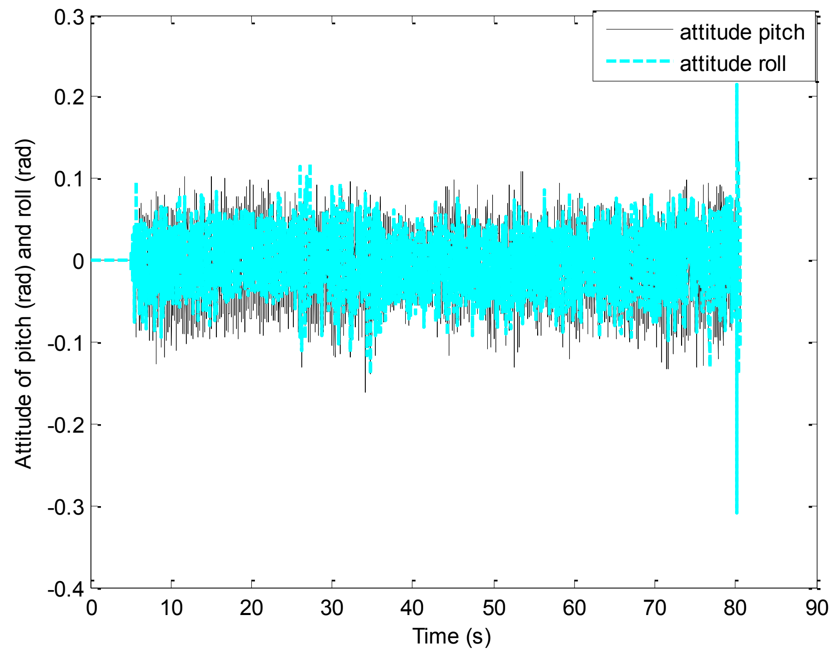

Fig. 4-65. Attitude of pitch and roll angles

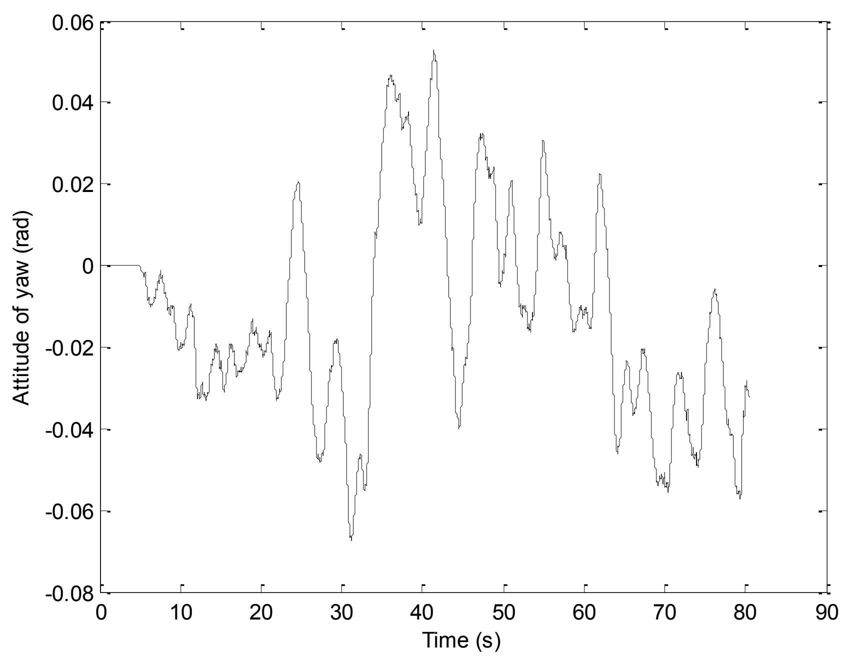

Fig. 4-66. Attitude of yaw angle

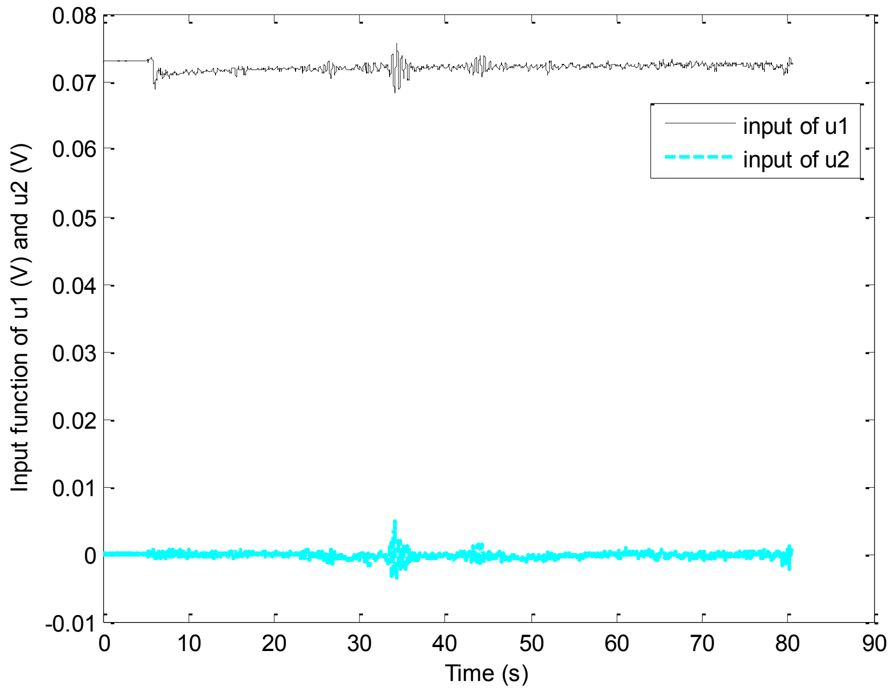

Fig. 4-67. Control inputs of u_1 and u_2

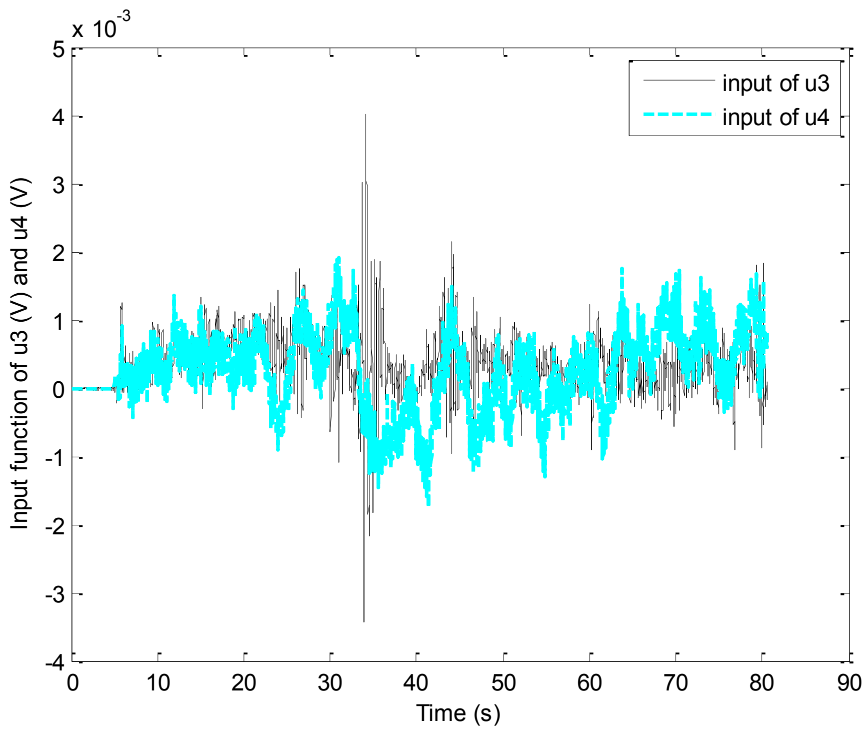

Fig. 4-68. Control inputs of u_3 and u_4

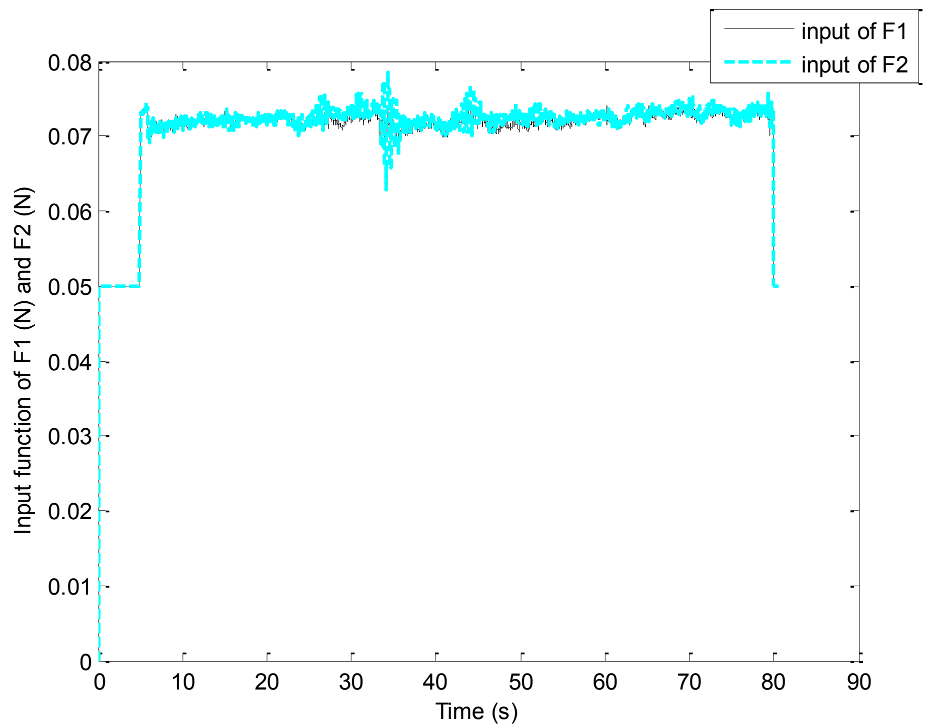

Fig. 4-69. Propellers forces F_1 and F_2

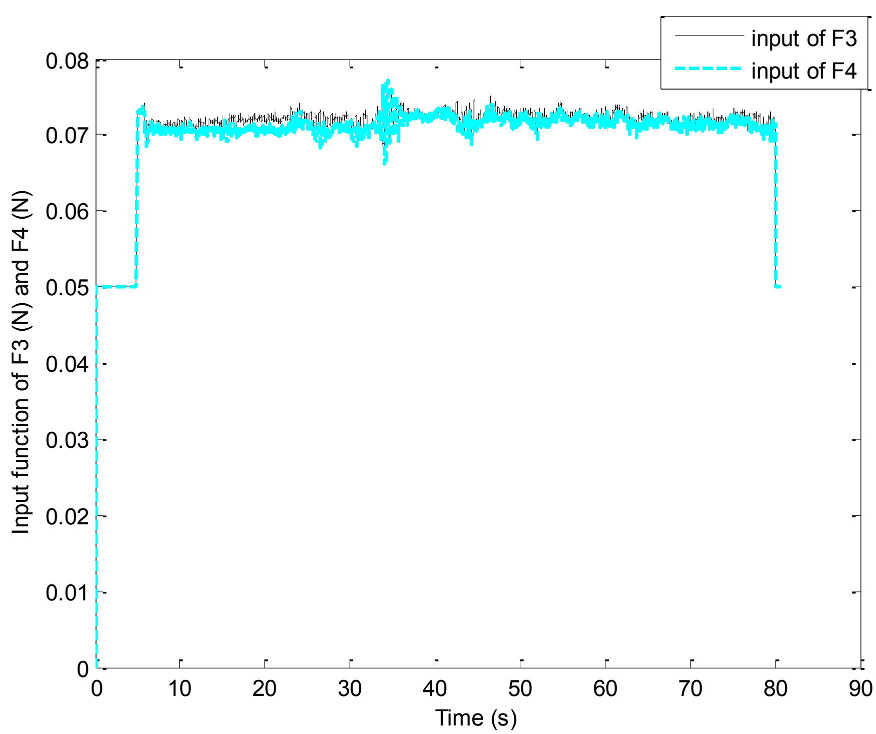

Fig. 4-70. Propellers forces F_3 and F_4

From the above figures shown, it can be seen that the feedback linearization controller controlled the Qball-X4 to finish the trajectory tracking successfully. However, as expected, due to the highly coupled matrix $\boldsymbol{G}'(\boldsymbol{X})\boldsymbol{U}'$, the change of any control input will lead to the rest of the control inputs change. Then, the corresponding attitudes and positions will change accordingly. Therefore, FLC kept trying to stabilize the Qball during the whole flight test, and this is the reason why the performance of the tracking task seems very jumpy.

4.4.2. Sliding Mode Control

For the same desired square trajectory of $1.5\times1.5m^2$, the SMC has also been implemented and fully tested. The results are listed in Figures 4-71 to 4-80 to demonstrate the performance of the control system and for the comparison with the other controllers.

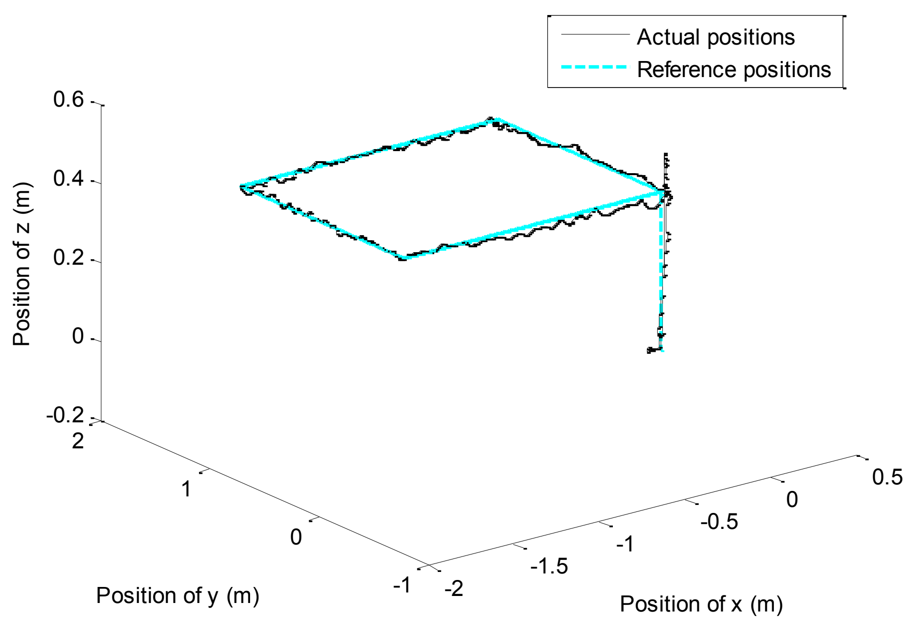

Fig. 4-71. 3-dimensional path tracking

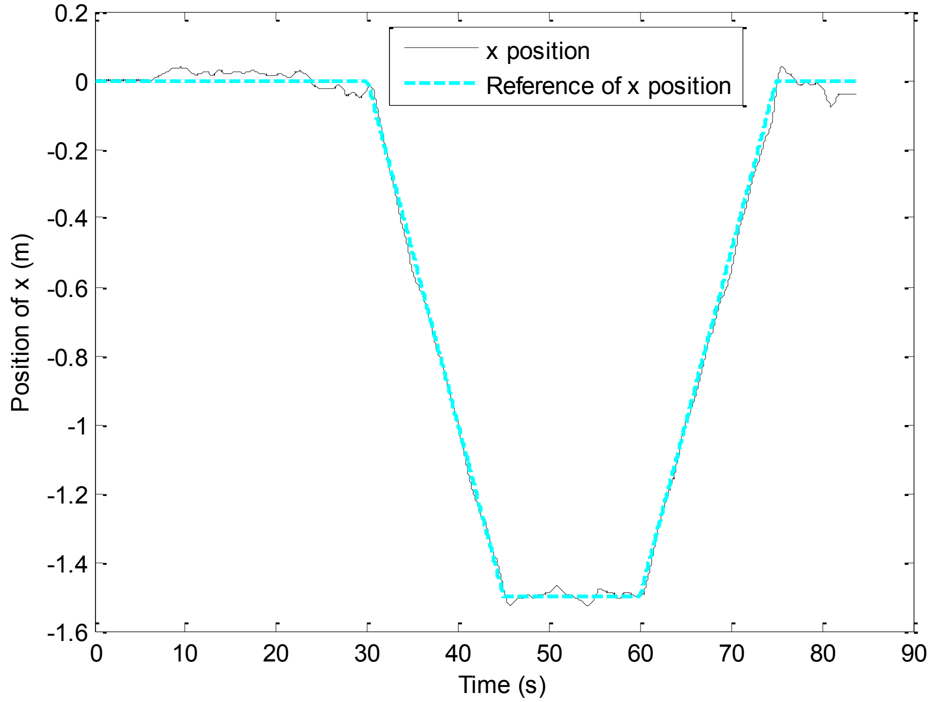

Fig. 4-72. Position tracking in x direction

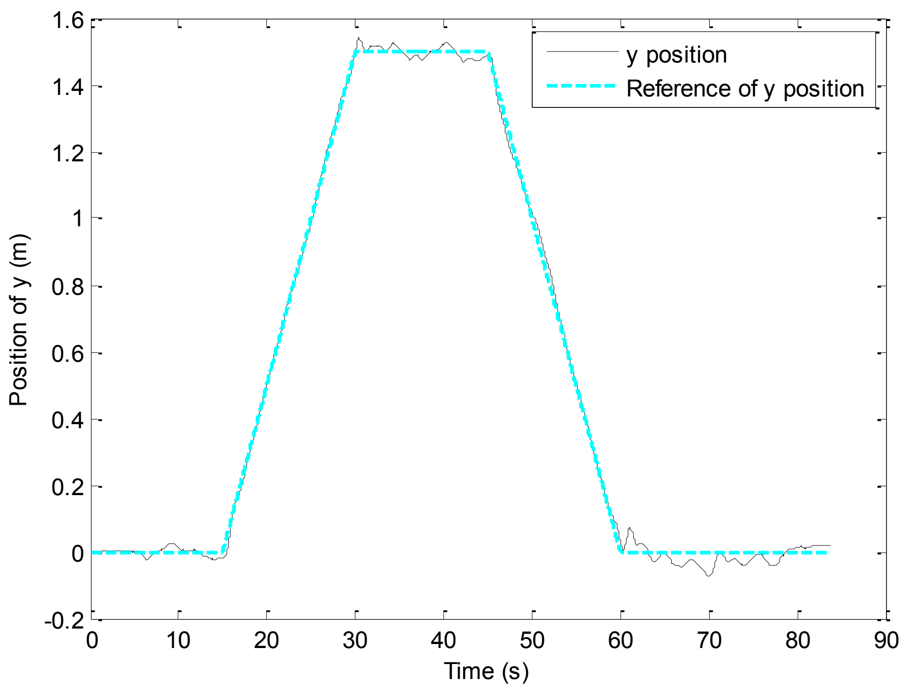

Fig. 4-73. Position tracking in y direction

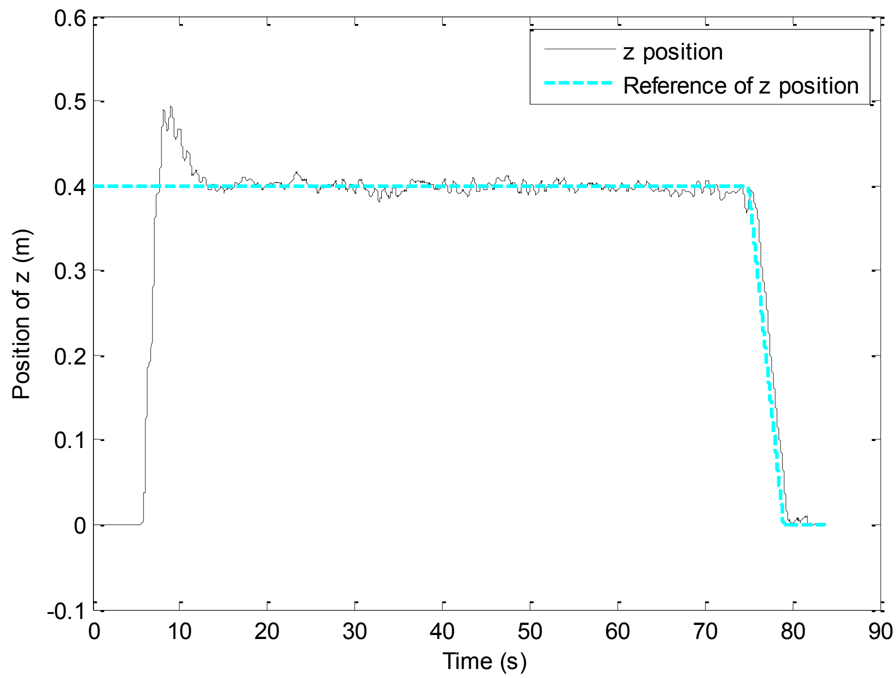

Fig. 4-74. Position tracking in *z* direction

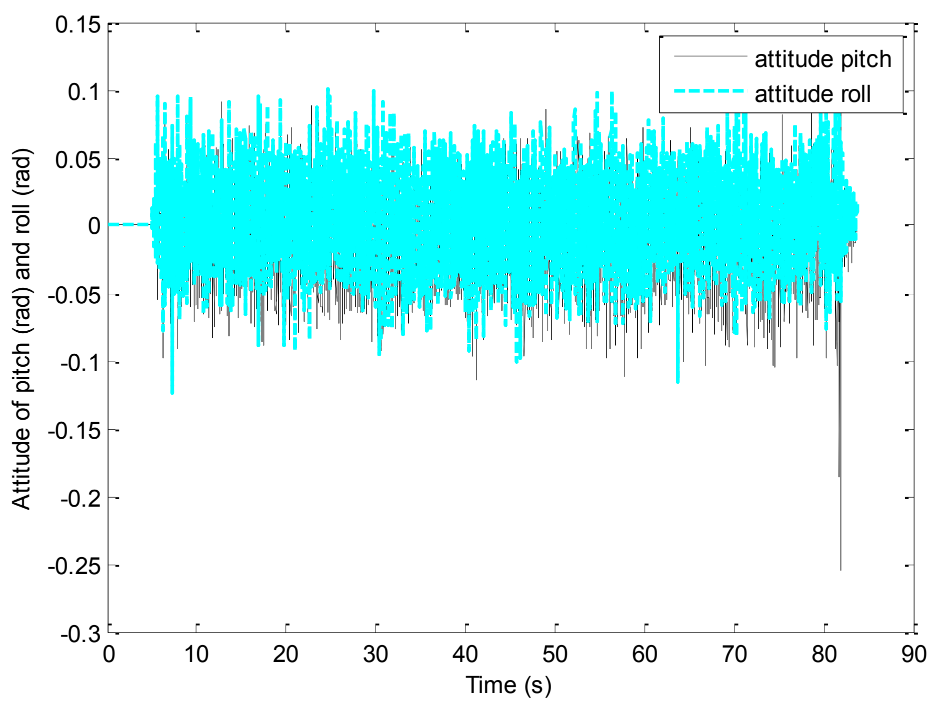

Fig. 4-75. Attitude of pitch and roll angles

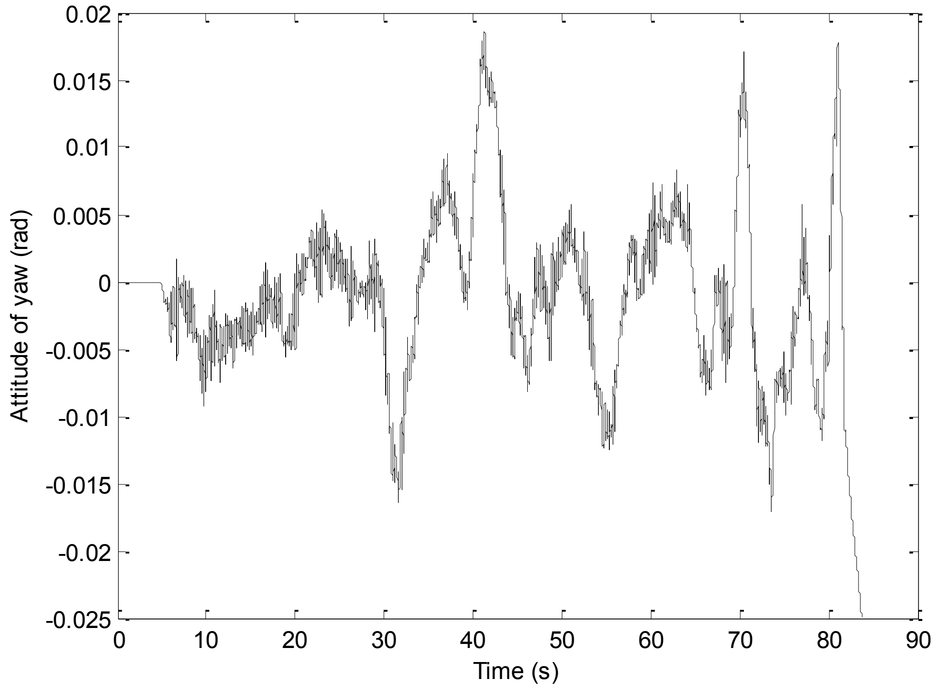

Fig. 4-76. Attitude of yaw angle

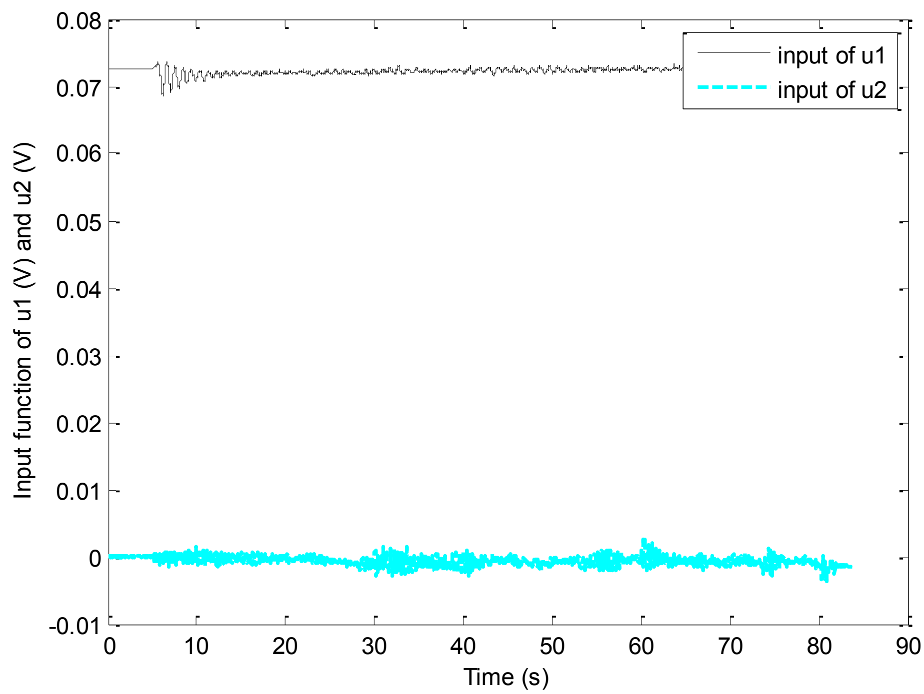

Fig. 4-77. Control inputs of u_1 and u_2

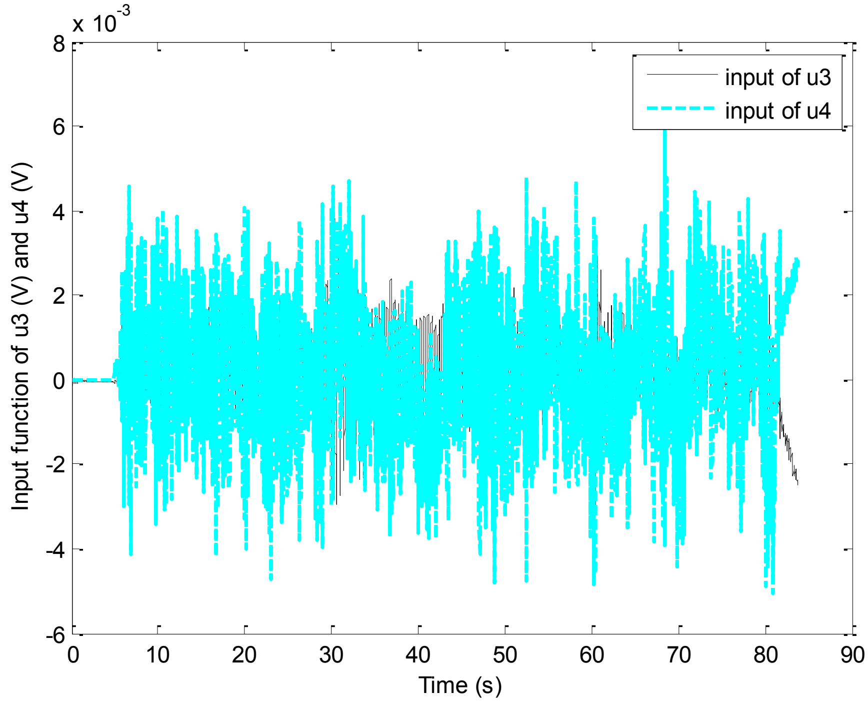

Fig. 4-78. Control inputs of u_3 and u_4

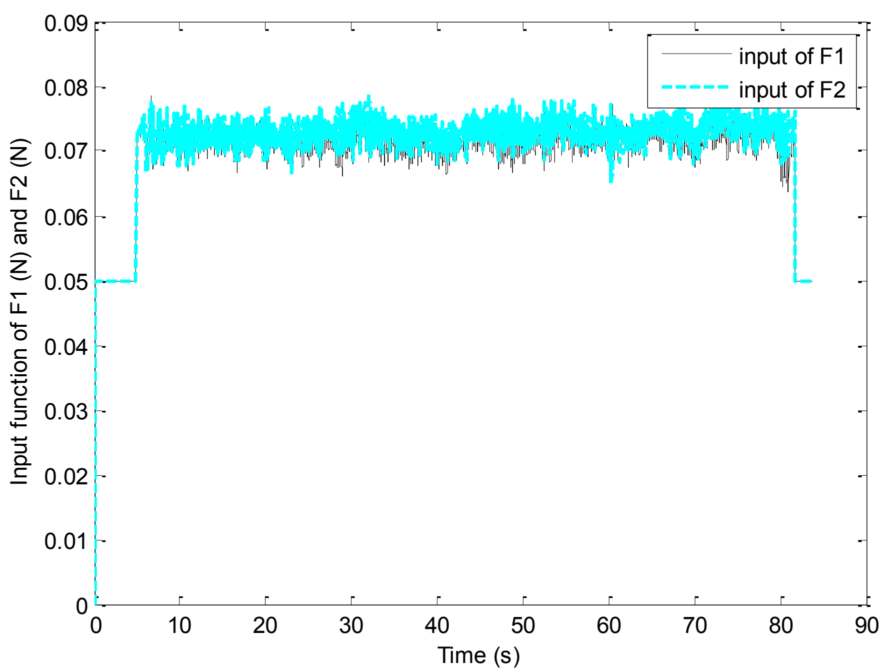

Fig. 4-79. Propellers forces F_1 and F_2

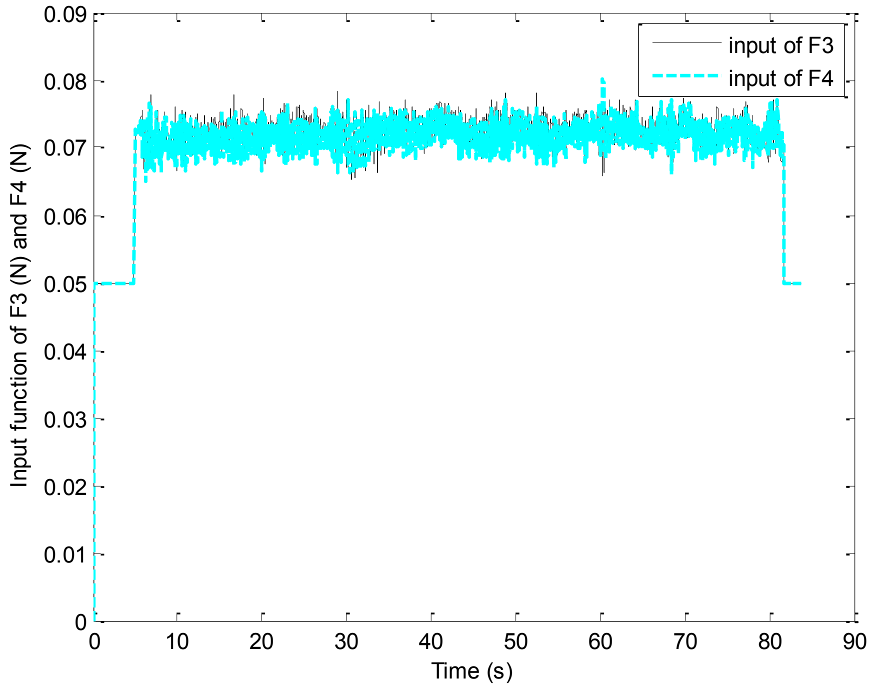

Fig. 4-80. Propellers forces F_3 and F_4

From the results, the sliding mode controller has been proven a very robust controller. By adding the augmented sliding surface, the desired trajectory has been tracked almost perfectly, and the task is very well accomplished. From Figures 4-71 to 4-74, it can be seen that the integration component in the sliding surface does not only increase the stability of the control system, but also smoothens the tracking trajectories.

4.4.3. Backstepping Control

Based on the exactly same condition and desired trajectory, backstepping control has been implemented and tested as well, which are showed in Figures 4-81 to 4-90. The results show that the tracking task is well accomplished and reveal the differences between the behaviours of backstepping controller and that of feedback linearization controller and sliding mode controller.

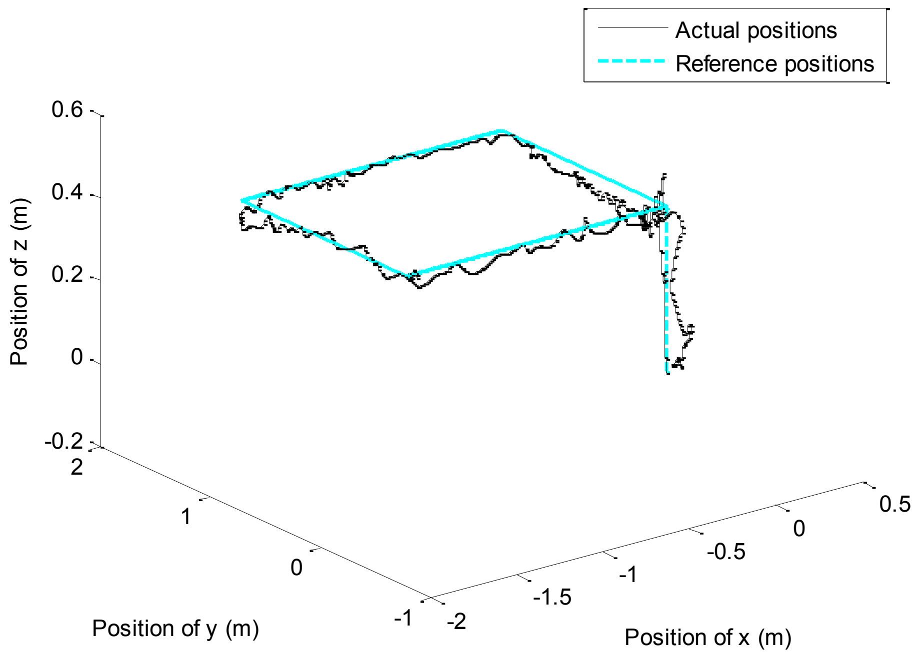

Fig. 4-81. 3-dimensional path tracking

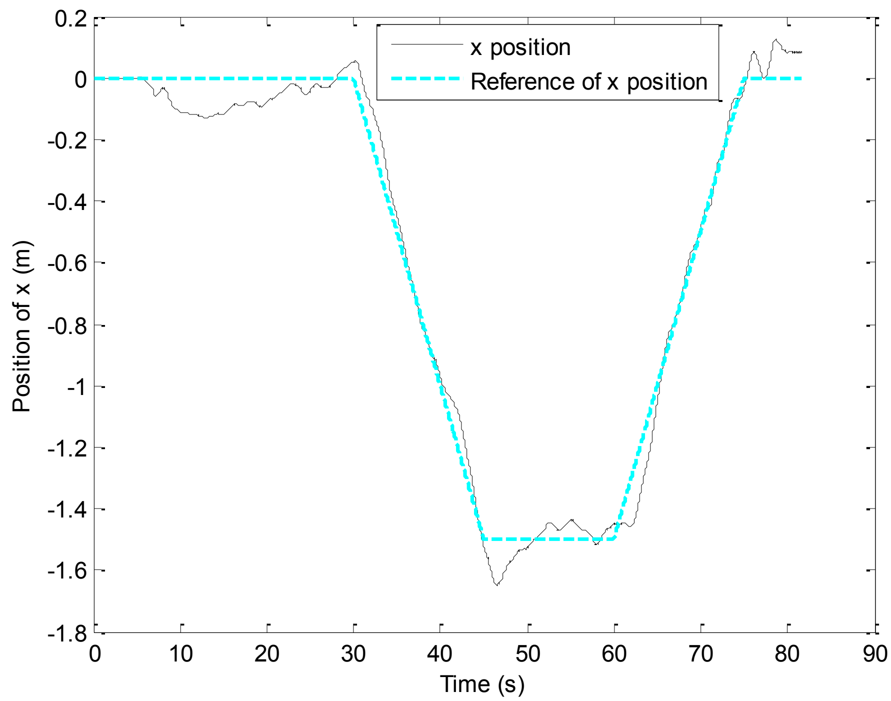

Fig. 4-82. Position tracking in *x* direction

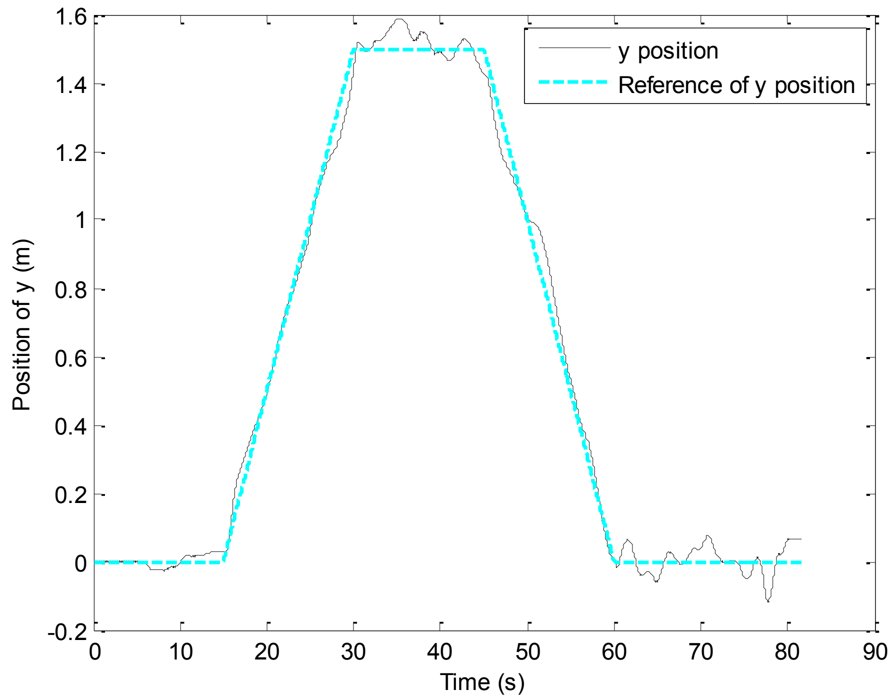

Fig. 4-83. Position tracking in y direction

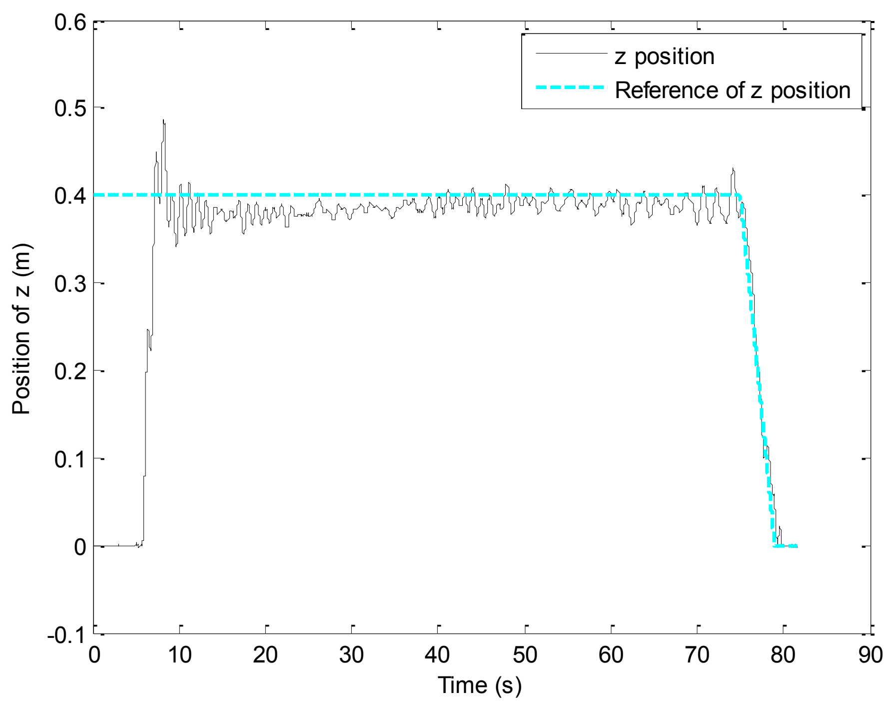

Fig. 4-84. Position tracking in z direction

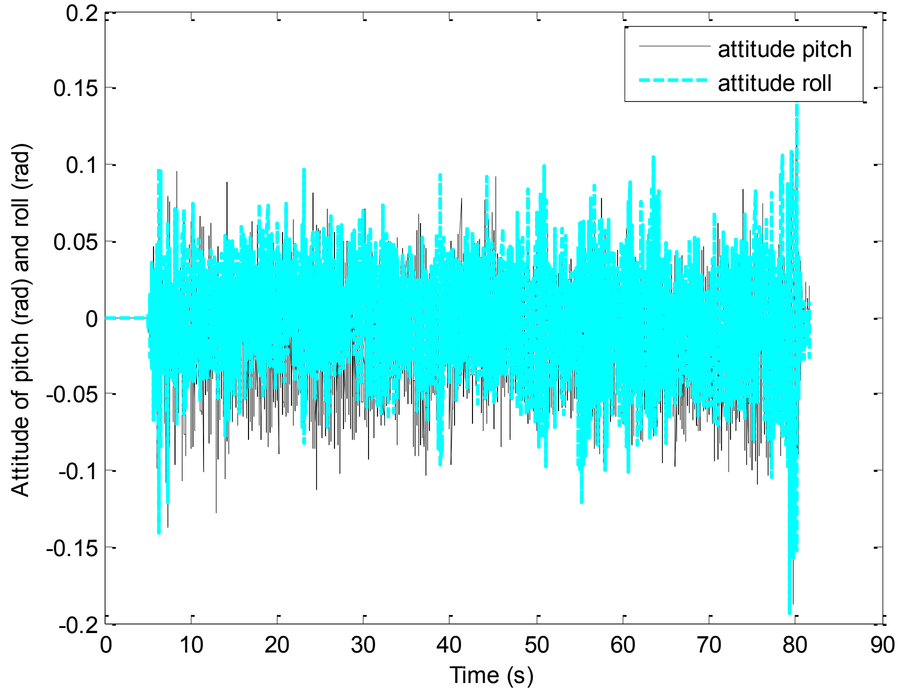

Fig. 4-85. Attitude of pitch and roll angles

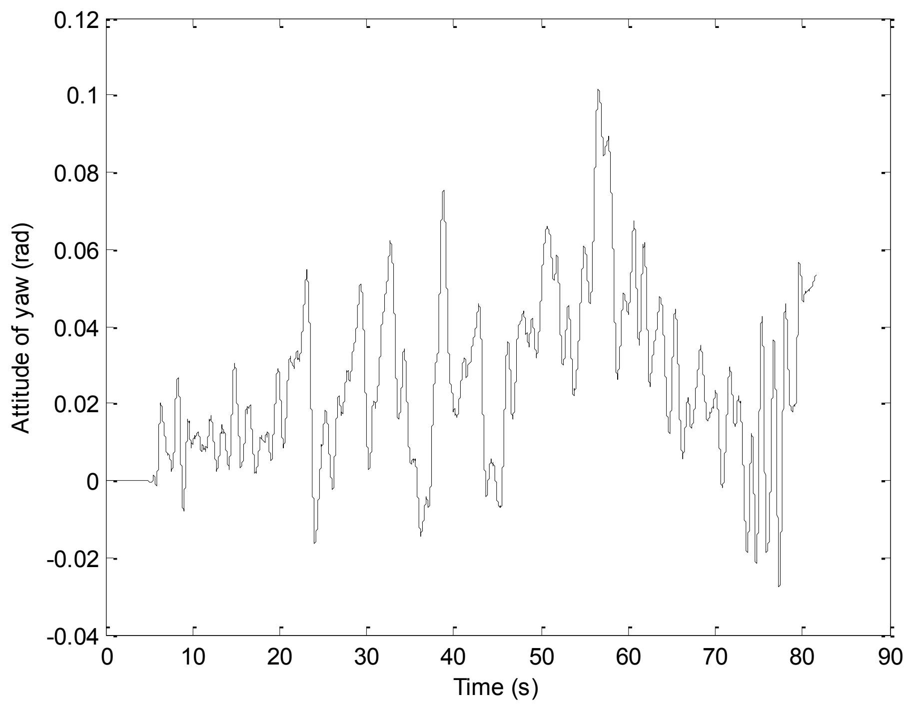

Fig. 4-86. Attitude of yaw angle

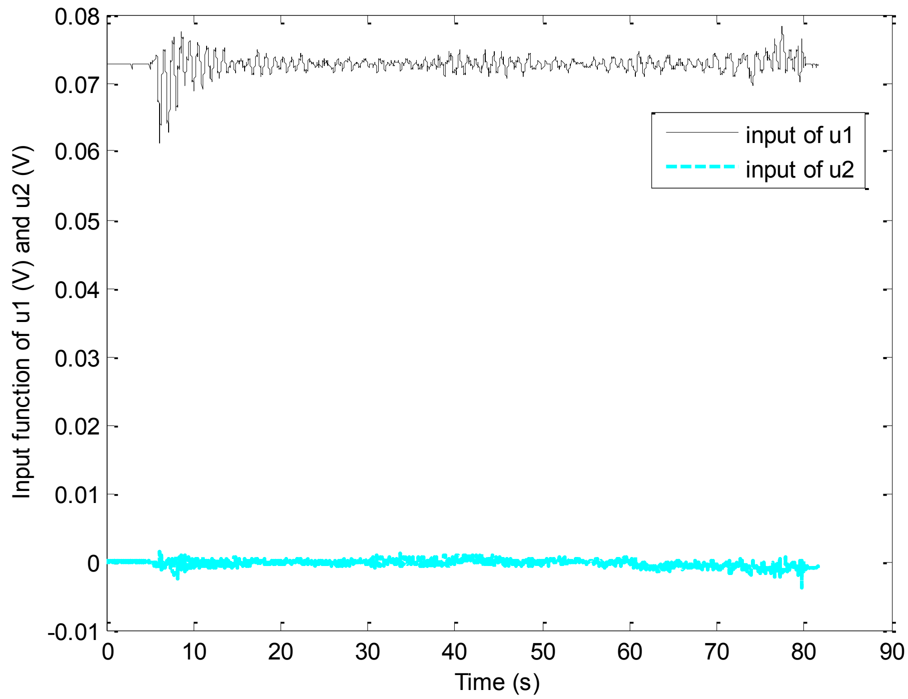

Fig. 4-87. Control inputs of u_1 and u_2

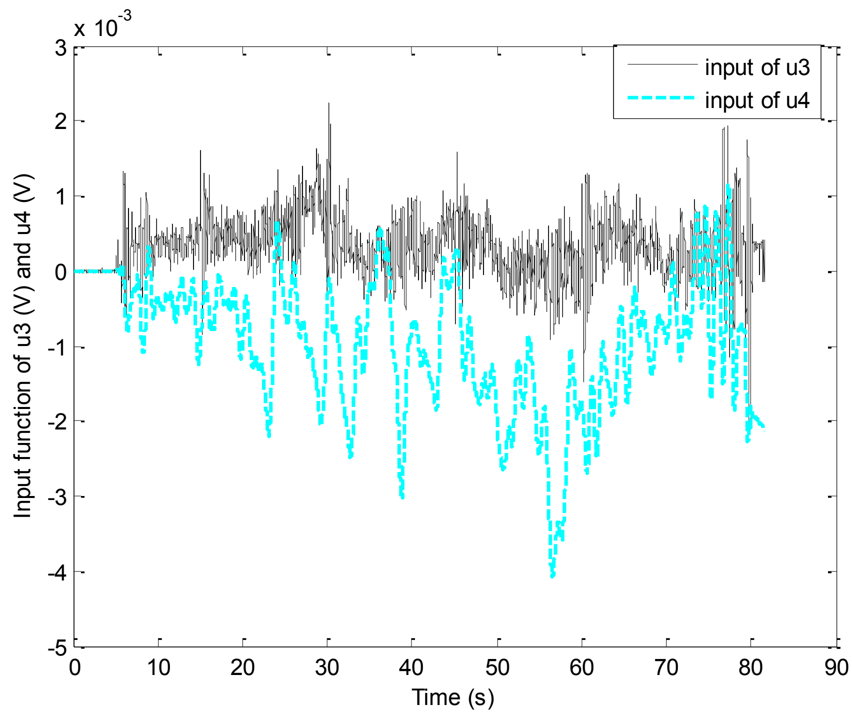

Fig. 4-88. Control inputs of u_3 and u_4

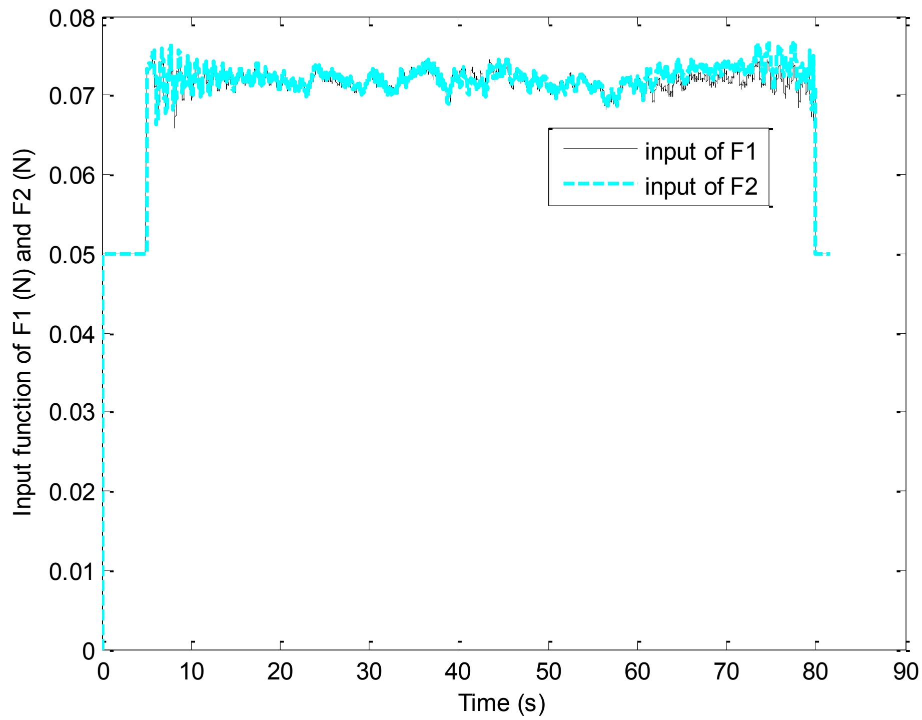

Fig. 4-89. Propellers forces F_1 and F_2

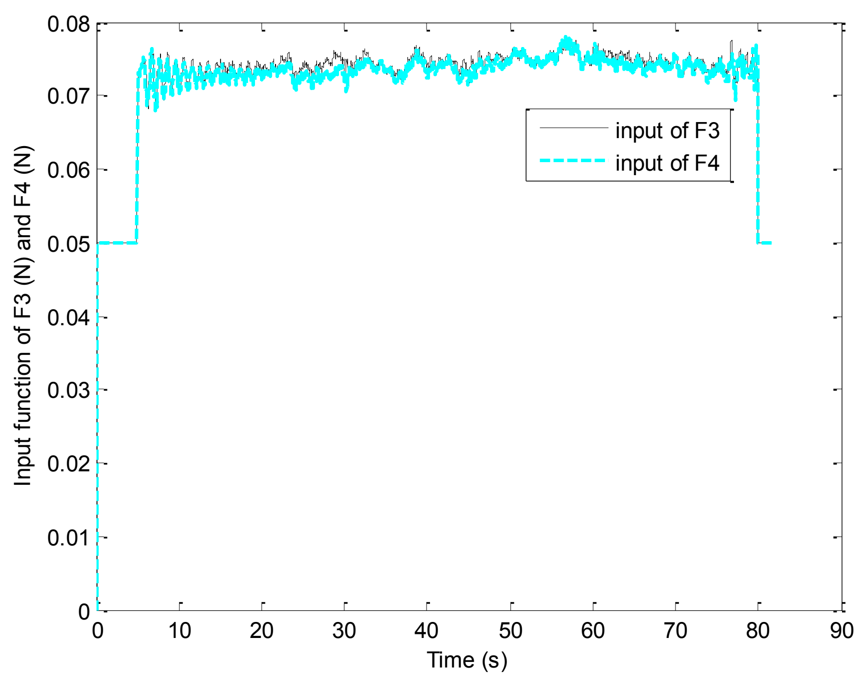

Fig. 4-90. Propellers forces F_3 and F_4

In this thesis, the backstepping controller is designed to have some robustness by decoupling the control inputs. The results have proven that for the same Qball-X4 system, the backstepping technique is more stable than the feedback linearization technique, and less stable than the sliding mode technique, as shown from Fig. 4-81 to Fig. 4-84 of the actual trajectory of the system.

4.5. Comparison of the Three Controllers

From the previous three sections, simulation results have shown all three controllers are tested successfully under both noiseless and noisy conditions. The performance of each controller varies from others, due to its own feature. The differences will be discussed in the following.

When there is no noise added in all three control systems, through Fig. 4-1 to Fig. 4-4, feedback linearization controller has shown that the tracking task is achieved. However, at each turn of the square trajectory, FLC has a small curve and delay to follow the desired path on *x, y*, and *z* axes. From equation (4-5), it can be seen that all the control inputs except for u_4 are coupled in the matrix $\boldsymbol{G}'(\boldsymbol{X})\boldsymbol{U}'$. This means that if any control input changes, it will cause the changes of the rest inputs, and then changes of the positions. This is the reason why in *z* axis height position is changed every time *x* or *y* position changes. Unlike FLC, SMC and BSC decouple the matrix $\boldsymbol{G}'(\boldsymbol{X})\boldsymbol{U}'$. All four inputs u_1, u_2, u_3, u_4 have been separated into individual control from equations (4-44). Hence, any one changes will not cause the changes of others, which also can be seen in Fig. 4-21 to Fig. 4-24 and Fig. 4-41 to Fig. 4-44 that positions maintain stable on the desired path. The difference between SMC and BSC is that if the start point of desired path is far from the start point of controller or the desired speed is too fast, backstepping control will generate a huge control input to track the path as shown in Fig. 4-27 to Fig. 4-30 and Fig. 4-47 to Fig. 4-50, due to the square term of $\alpha(e+\alpha e)$ in equation (4-44). This is why for the same length of time, BSC can only track a shorter square trajectory than both FLC and SMC. Based on the testing results, it may show SMC is the best control algorithm in the application to the Qball-X4 test-bed; however, it shows in Fig. 4-24 that maintaining control inputs at all times may cause a delay in tracking.

When there is noise added into the system, the robustness of controllers can be shown clearly. For FLC, Fig. 4-11 to Fig. 4-20 show the performance of tracking has become affected by noise. Positions of x, y, and z can no longer be stabilized and because of the control inputs matrix coupling issue, height z position is even worse. The attitude parameters of pitch and roll are as well unstable compared to no noise condition, for following the desired path is adjusted by the changes of pitch and roll attitude. For BSC, although the overall performance has been deteriorated as shown in Fig. 4-51 to Fig. 4-60, all the positions and the attitudes are still being controlled to maintain a certain steady path by the decoupled control inputs. The only difference here is there is always an error between the reference path and the actual path. At last but not least, SMC shows a strong capability of dealing with noise as shown in Fig. 4-31 to Fig. 4-40. The actual tracking path is still very close to the reference. The attitude of pitch and roll become a bit unstable, because the controller tries to overcome the influence imposed on Qball-X4. Trying to adjust pitch and roll attitudes at all times is the effort of tracking the reference path. Switch function effect from equation (4-25) has secured once again for relatively stable control inputs and robustness.

The simulation results of each control system have been discussed and compared above. The sliding mode controller has been proven the best controller among the three nonlinear control techniques investigated in this thesis. However, the simulations are still based on theoretical assumptions. The actual applications are much more persuasive on comparing these three different nonlinear control algorithms. From the three sets of experimental testing results, it can be seen clearly that the sliding mode control technique is truly the most robust and high performance control algorithm investigated in this thesis. From all the figures of control inputs u_1, u_2, u_3, u_4 and F_1, F_2, F_3, F_4 of three experiments, the values are within the same range which makes all three controllers succeed in completing the tracking task. From the figures of 3-dimensional path tracking and tracking of positions *x, y*, and *z*, it can be seen that the tracking error of sliding mode controller is the smallest. Thus, the desired trajectory and the actual trajectory of the Qball-X4 system are extremely close to each other by SMC. For backstepping controller, the tracking errors become larger. The overshoot happened more often than with SMC, especially at the turning points on each axis. The backstepping controller responds to the change of the situation a bit slower. The feedback linearization controller has the worst performance of tracking. Due to the highly coupled control inputs, the controller had to adjust the inputs all the time. A small disturbance on one direction could lead to the changes of all the directions and control inputs. Therefore, from the figures of feedback linearization control experiments, the system is barely reached to steady-state and hardly followed the desired path, especially for *x* and *y* axes. The tracking errors are also the largest. Some numeric comparisons are listed in Table 4-1 to 4-3 to show the different behaviours of these three controllers. By the numbers from the experiments, it can be showed that the sliding mode controller is the best, and then backstepping controller, and the last is feedback linearization controller.

Table 4-1 The comparison of position *x*

Time		5 – 15 (seconds)	15 – 30 (seconds)	30 – 45 (seconds)	45 – 60 (seconds)	60 – 75 (seconds)	Overall
Ref	Mean	0	0	-0.7500	-1.500	-0.7500	-0.5375
FLC	Mean	-0.0252	-0.0485	-0.6755	-1.5670	-0.8386	-0.5768
	Variance	-0.000295	0.0041	0.1644	0.0115	0.1800	0.4086
SMC	Mean	0.0196	-0.000783	-0.7581	-1.4975	-0.7772	-0.5433
	Variance	0.000131	0.000630	0.1857	0.000184	0.1784	0.3834
BSC	Mean	-0.0823	-0.0462	-0.6946	-1.5085	-0.7742	-0.5602
	Variance	0.0020	0.0017	0.2029	0.0034	0.2280	0.3851

Table 4-2 The comparison of position *y*

Time		5 – 15 (seconds)	15 – 30 (seconds)	30 – 45 (seconds)	45 – 60 (seconds)	60 – 75 (seconds)	Overall
Ref	Mean	0	0.7500	1.5000	0.7500	0	0.5375
FLC	Mean	-0.1112	0.7372	1.5132	0.6984	0.6984	0.5444
	Variance	0.0023	0.2293	0.0100	0.1414	0.0048	0.3958
SMC	Mean	-0.0020	0.7474	1.4994	0.7566	0.7566	0.5345
	Variance	0.000193	0.1937	0.000336	0.1797	0.000897	0.3871
BSC	Mean	0.0023	0.7397	1.5164	0.7629	0.7629	0.5562
	Variance	0.000381	0.1538	0.0014	0.1682	0.0011	0.3805

Table 4-3 The comparison of position *z*

Time		5 – 15 (seconds)	15 – 30 (seconds)	30 – 45 (seconds)	45 – 60 (seconds)	60 – 75 (seconds)	Overall
Ref	Mean	0.4000	0.4000	0.4000	0.4000	0.4000	0.3679
FLC	Mean	0.3588	0.3933	0.3958	0.3970	0.3932	0.3509
	Variance	0.0196	0.0000376	0.000141	0.0000605	0.0000972	0.0152
SMC	Mean	0.3599	0.4015	0.3969	0.4001	0.3945	0.3408
	Variance	0.0202	0.0000245	0.0000416	0.0000266	0.0000414	0.0191
BSC	Mean	0.3333	0.3790	0.3878	0.3915	0.3890	0.3361
	Variance	0.0155	0.0000599	0.0000529	0.0000719	0.000164	0.0157

To further test these three controllers, the speed of Qball-X4 has been increased to show the upper limits of controllers' responding time and capabilities of handling more disturbances. The results have been shown below, and the performance of SMC is still proven to be the best.

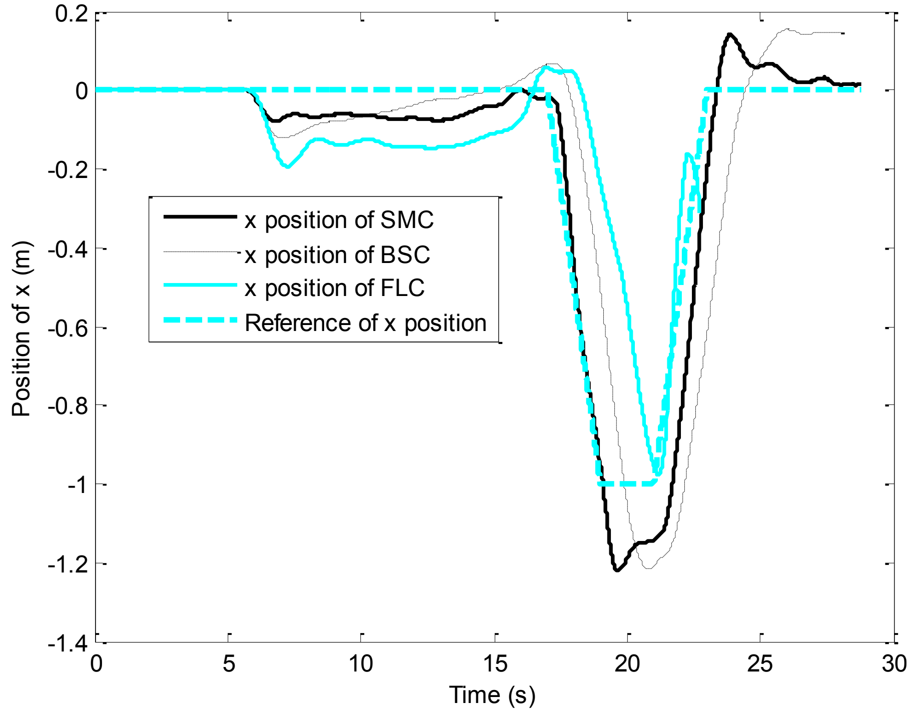

Fig. 4-91. Position tracking in x direction

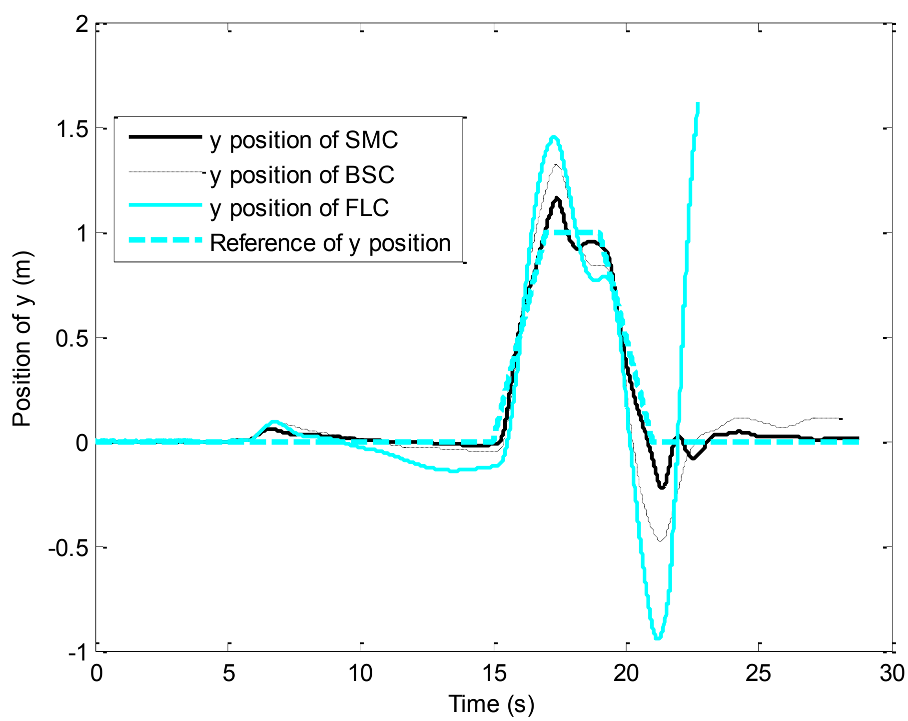

Fig. 4-92. Position tracking in y direction

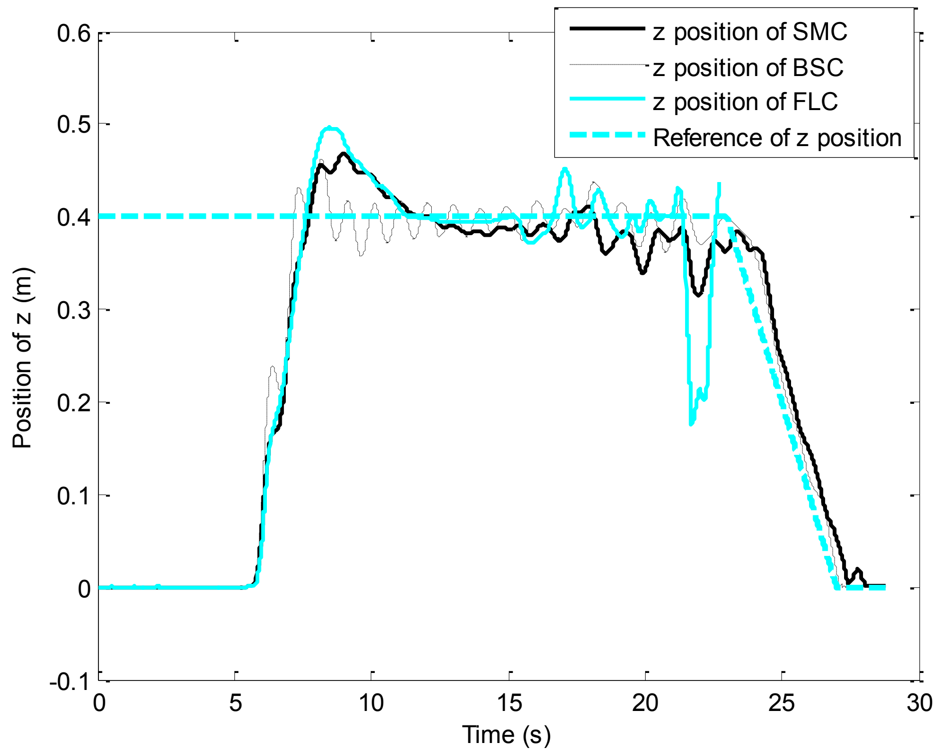

Fig. 4-93. Position tracking in *z* direction

There is a LQR controller implemented in the Qball-X4 system as the baseline controller. Due to some limitations of the indoor testing environment, Qball-X4 model dynamics become linear on occasion. Therefore, a linear control algorithm LQR can be implemented for this UAV under the linear condition. The tracking performance of the original LQR controller is shown as following. From the position tracking Figures 4-94 to 4-96, when the Qball-X4 system works in a linear situation, the controller behaviour is similar to the behaviour of BSC.

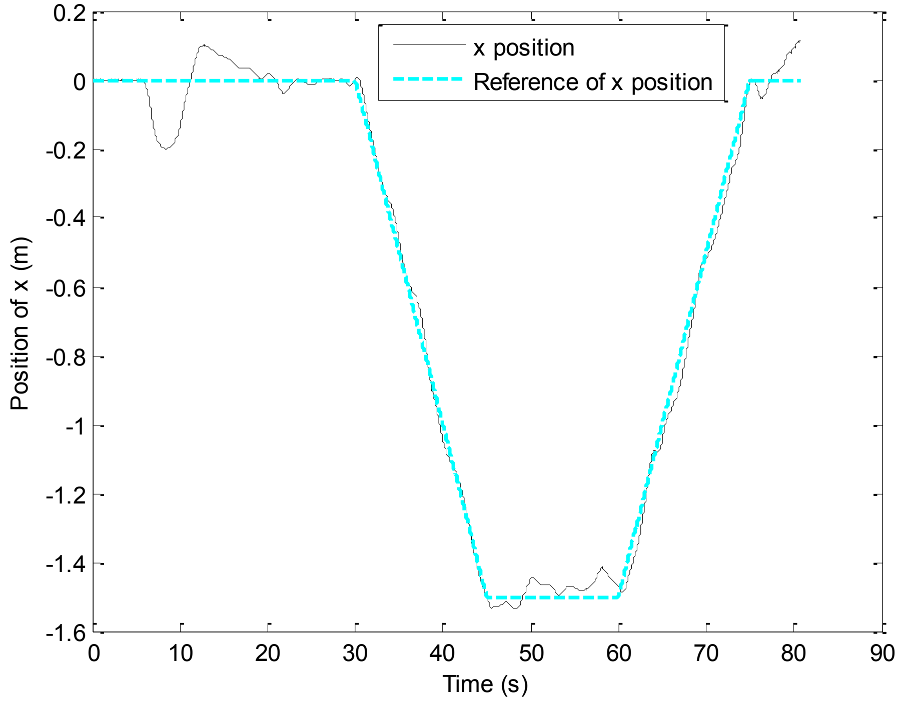

Fig. 4-94. Position tracking in *x* direction

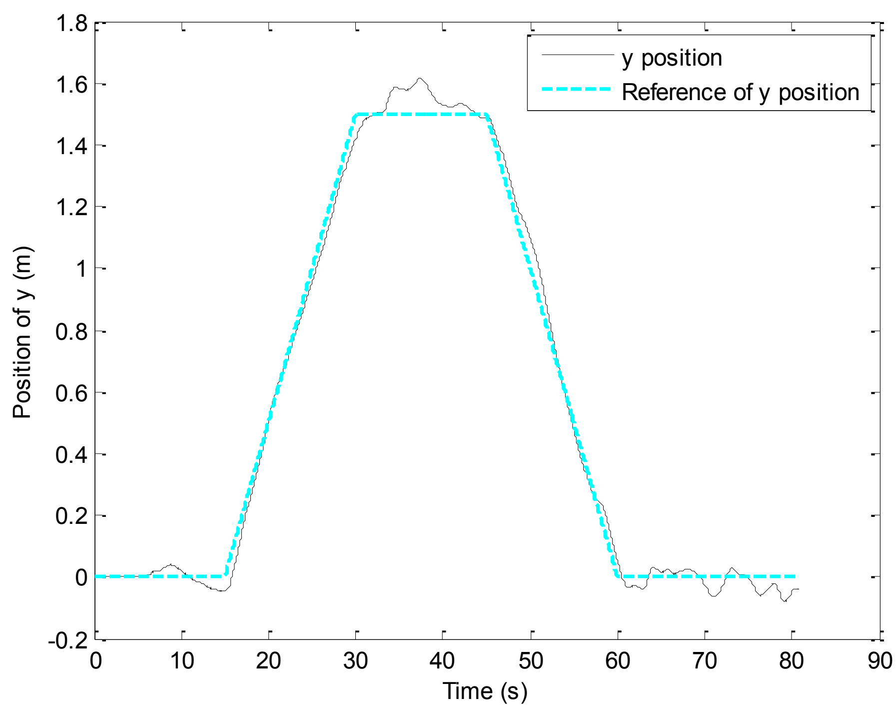

Fig. 4-95. Position tracking in *y* direction

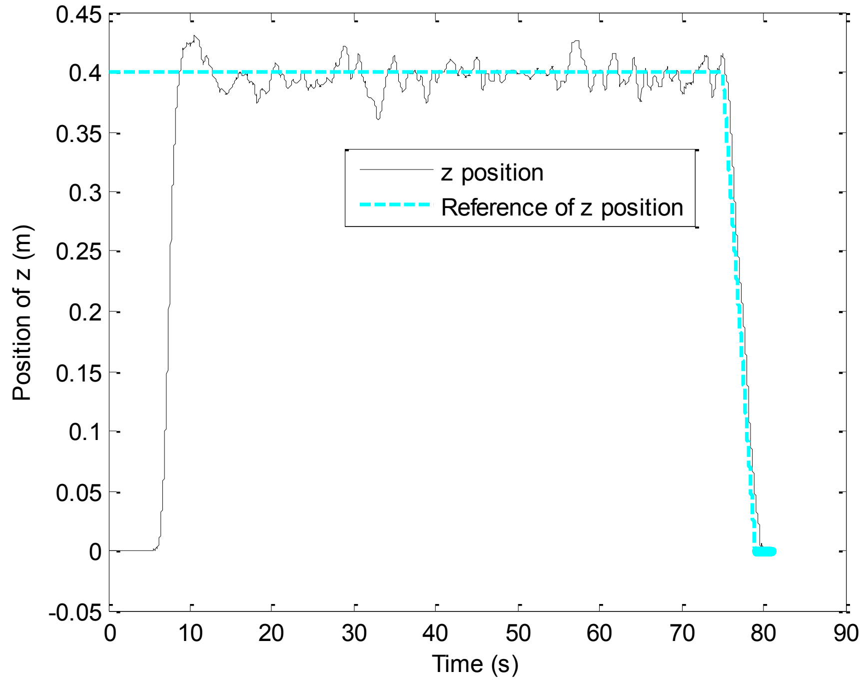

Fig. 4-96. Position tracking in *z* direction

In one words, all the control algorithms discussed in the thesis have different behaviours, but attitude yaw has been controlled very well by these three controllers at all times. From all the simulation results and experimental testing results, it can be learnt that SMC has been proven as a very practical control algorithm in dealing with noise and uncertainties. Backstepping control can be used if decoupling is needed. Feedback linearization control is easy to use but may come with a price, such as an enormous number of matrix calculations and instability caused by coupled control inputs, and sensitivity to modeling errors, uncertainties and noises.

4.6. Summary

The three popular nonlinear control algorithms have been designed and tested successfully in the Qball-X4 UAV test-bed. For the flight tests, all three controllers have been focused on attitude control. Since once all the control inputs for attitudes (pitch, roll, and yaw) have been controlled properly, the trajectory tracking can then be realised easily. By the theoretical and experimental analysis and comparison, FLC has been proven to have the worst performance on the Qball-X4 system, and SMC has the best performance for tracking task. Based on the testing result, investigation of this chapter has also shown the best possible candidate for the FTCS, the SMC. Therefore, in next chapter, SMC has been selected for Fault-Tolerant Control (FTC) of the Qball-X4 in the presence of actuator faults or propeller damages during flight.

5. Fault-Tolerant Control of the Qball-X4 System

5.1. Overview

Modern technologies have realized many different devices and systems. For instance, cars and planes are becoming more and more important than ever in our daily life. A safe and reliable control system is then desired in these applications, since the consequences of faults occurrence can lead to the loss of lives. Building a fault free system is not realistic, therefore it is necessary to design a control system that can tolerate the faults. By adding a fault-tolerant controller into the system, the reliability, availability and maintainability of the system will be improved.

There are three different fault scenarios generally considered: actuator faults, sensor faults, and component faults.

Actuator faults are those faults when the system loses partial or total control function due to actuator malfunctions. For example, if one of the aircraft engines is malfunctioning, the whole actuation from the actuators of the system will be reduced no matter what control input is applied for. The system will become unbalanced may loss control.

Sensor faults are those faults when the sensors do not give the correct measurements. This can be caused by connection of wires, or the noise from the environment.

Component faults are those faults when the faults that associated with system components other than actuators or sensors. This is caused often by plant itself, such as system coefficients.

For fault-tolerant control, there are two different types of control strategies. One is Passive Fault Tolerant Control (PFTC), and the other is Active Fault Tolerant Control (AFTC) [1].

Passive fault-tolerant control needs a fixed controller that can be used for normal and all possible fault cases to minimize the worst case performance. The system diagram is shown below.

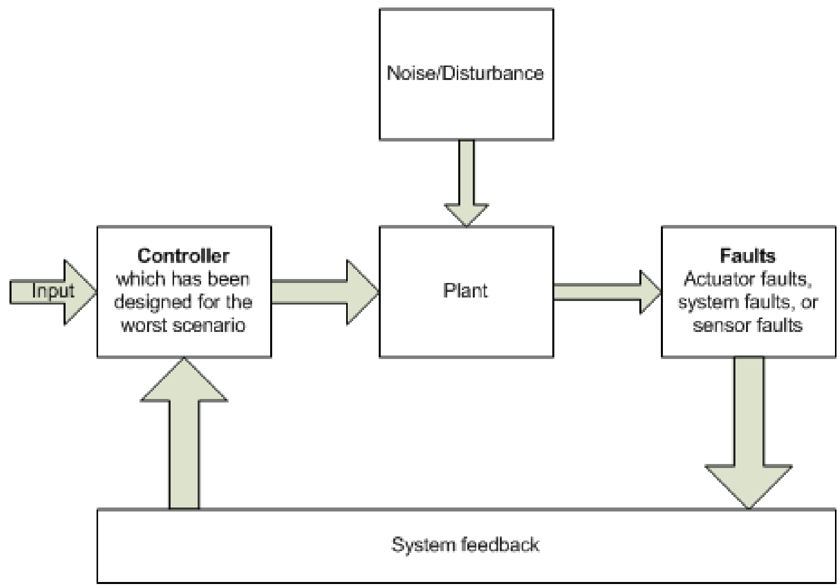

Fig. 5-1. A PFTC system diagram

Active fault-tolerant control needs a controller reconfiguration mechanism, and a Fault Detection and Diagnosis (FDD) component. The controller that can be used in AFTC has to have the reconfigurable capability. Since the controller can be reconfigured when the faults occur, AFTC has a stronger capability than PFTC. The system diagram is shown below.

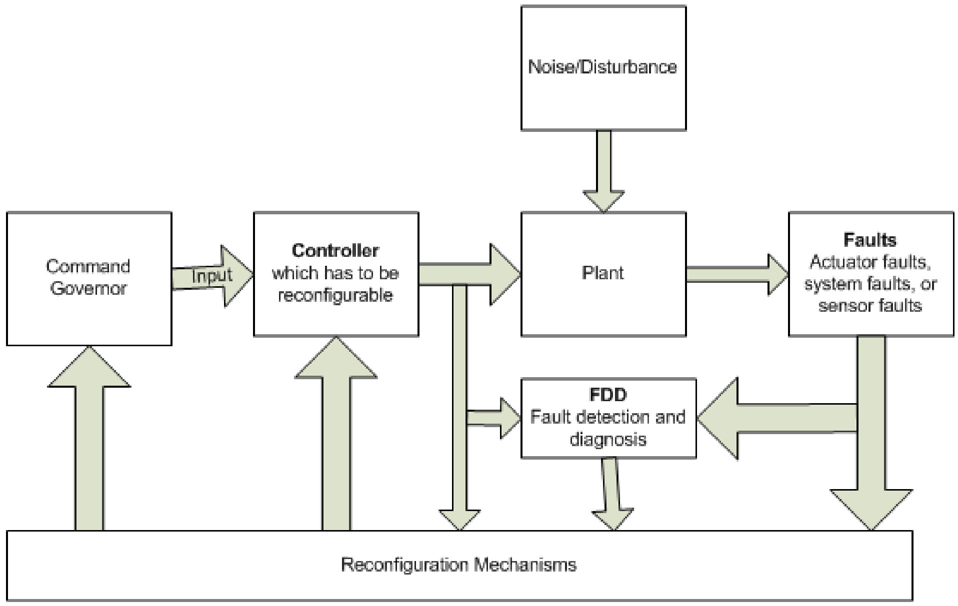

Fig. 5-2. An AFTC system diagram

5.2. Sliding Mode-based Fault-Tolerant Control

A fault-tolerant control is a special type of control techniques that can handle the faulty situations, hence a robust control algorithm is needed to ensure the reliability of the fault-tolerant control system. As introduced and tested in the previous chapters, sliding mode control is a robust control methodology and provides the best performance among three nonlinear control techniques in the application to the Qball-X4 system under normal flight conditions. Due to its unique design of sliding surface, SMC can be used to deal with uncertainties, which also makes it a strong candidate for fault-tolerant control. In the following section, a sliding mode-based fault-tolerant control will be developed.

5.2.1. Passive Fault-Tolerant Control for Qball-X4 System

From Chapter 4, the sliding mode controller has already been designed, which has also included an augmented algorithm by adding an integration component into the sliding surface. In other words, an extra proportional control can be achieved by taking the first derivative on the integration, as equation (4-15) indicates. Therefore, by using this idea, equations (4-24) and (4-25) have shown a very robust sliding mode controller. However, equations (4-24) and (4-25) are derived without the consideration of faulty situations. In order to handle the faults which are mainly the actuator faults in this research, a few changes need to be done to the previous designed controller in Chapter 4. A trade off needs to be added into the control system, in order to balance the performance between faulty situation and fault-free situation, using only one controller, therefore named as passive fault-tolerant controller or reliable controller. In other words, the only controller for both situations will be weaker than it is for each different individual situation. Therefore, all the control gains have to be reset. The following equations have shown the controller for PFTC with the new gains.

$$
\begin{aligned}
\hat{u}_x &= \frac{1}{u_1}[\ \cdots\ -2k_{p1}e_x] \\
\hat{u}_y &= \frac{1}{u_1}[\ \cdots\ -2k_{p2}e_y] \\
\hat{u}_1 &= \frac{1}{\mathrm{c}\phi\mathrm{c}\theta}[\ \cdots \\
\hat{u}_2 &= \frac{1}{l}(\ \cdots\ J_x \ \cdots\ J_x \ \cdots \\
\hat{u}_3 &= \frac{1}{l}(\ \cdots\ \frac{J_x}{J_y}x_8x_{12} - \frac{J_r x_8 \Omega}{J_y} + d_\phi x_{10} + \lambda_{p5}(\ \cdots\ )
\end{aligned}
\tag{5-1}
$$

so that the final control inputs are obtained by:

$$u_{pi} = \hat{u}_{pi} + k_{psi} sign(s_i) \tag{5-2}$$

where λ_{pi}, k_{psi}, and k_{pi} are all positive gains.

A new saturation function is needed to eliminate the nonlinearity caused by the occurrence of a fault, and also to achieve a relatively fast convergence of the system. Rewrite the equation above into the following format:

$$\begin{cases} u_{pi} = \hat{u}_{pi} + k_{psi}(-\delta) & if \quad s_i \leq -\delta \\ u_{pi} = \hat{u}_{pi} + k_{psi} s_i & if \quad -\delta < s_i < \delta \\ u_{pi} = \hat{u}_{pi} + k_{psi}(\delta) & if \quad s_i \geq \delta \end{cases} \tag{5-3}$$

where δ is the boundary of the saturation, and is set small enough.

The simulation is carried out under the situation of 15% loss of control effectiveness in the fourth propeller. The results shown in Figs. 5-3 to 5-12 have proved the PFTC controller has the ability of handling the actuator fault of 15% force loss. The task is still to track a square trajectory as what has been done in previous sections. After the occurrence of the fault, the original trajectory tracking needs to be maintained without any degradation to demonstrate the capability of the sliding mode-based passive fault-tolerant controller. The following figures have shown the performance of the tracking task is good. The desired path along all three axes *x*, *y*, and *z* has been well followed. The fault occurred at 20 seconds for all the following tests, including simulations and experiments for both PFTC and AFTC.

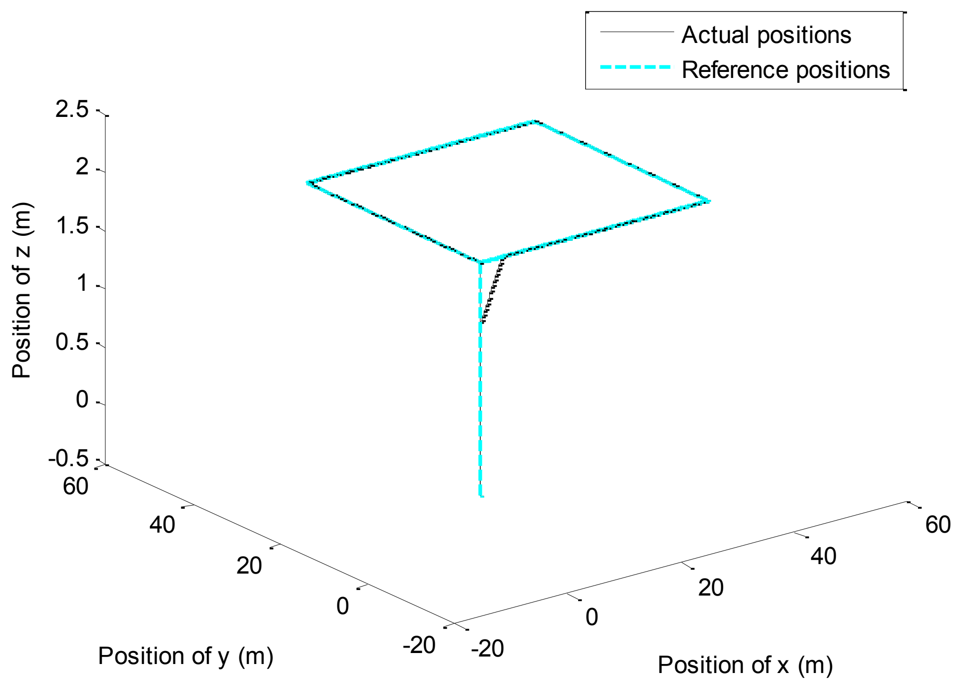

Fig. 5-3. 3-dimensional path tracking

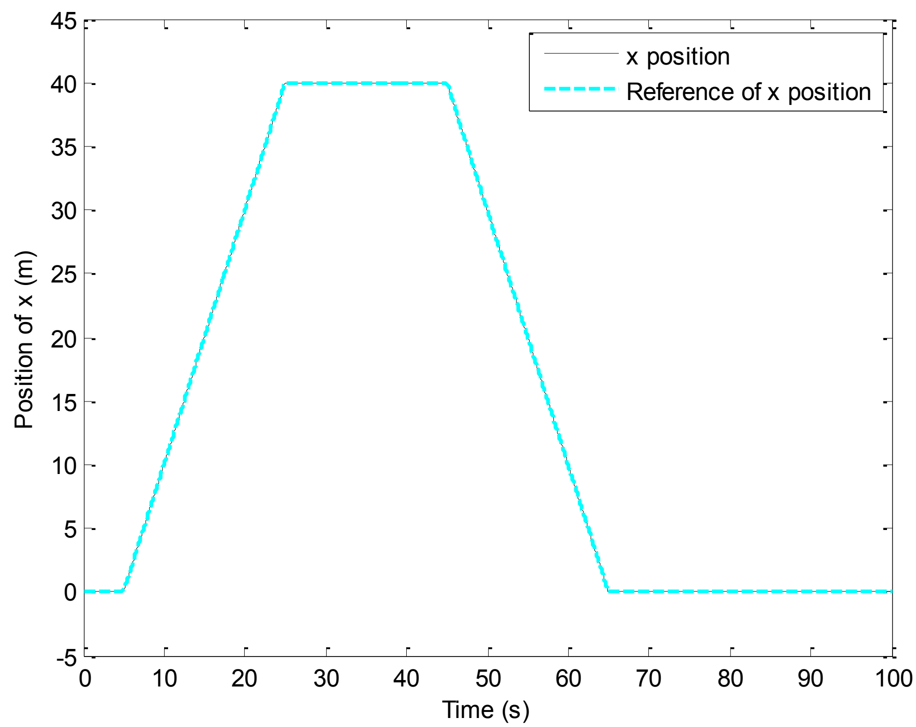

Fig. 5-4. Position tracking in *x* direction

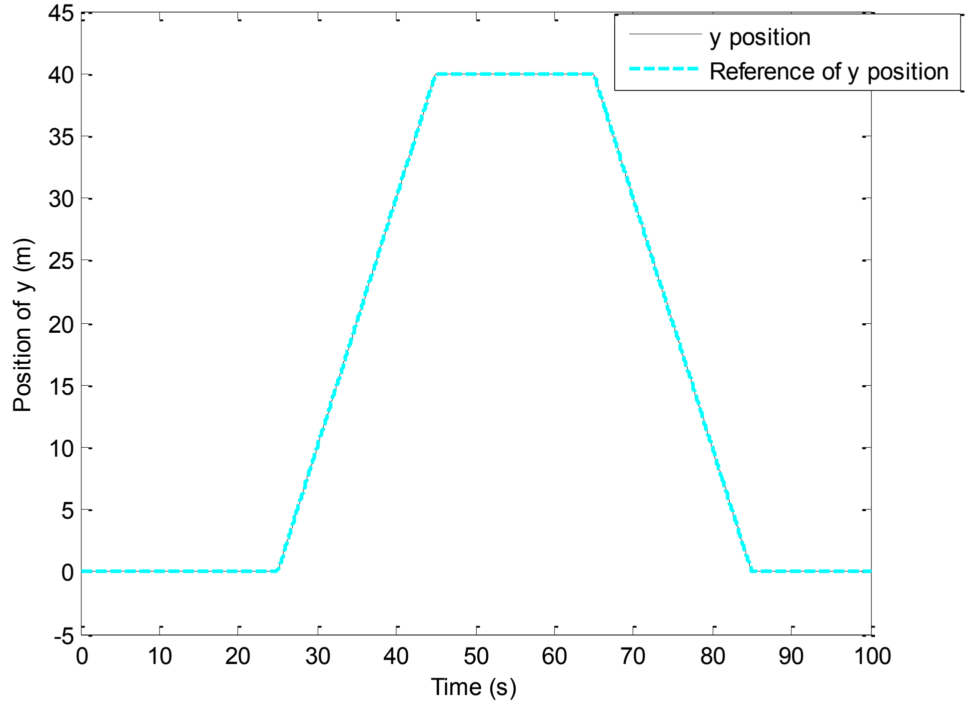

Fig. 5-5. Position tracking in *y* direction

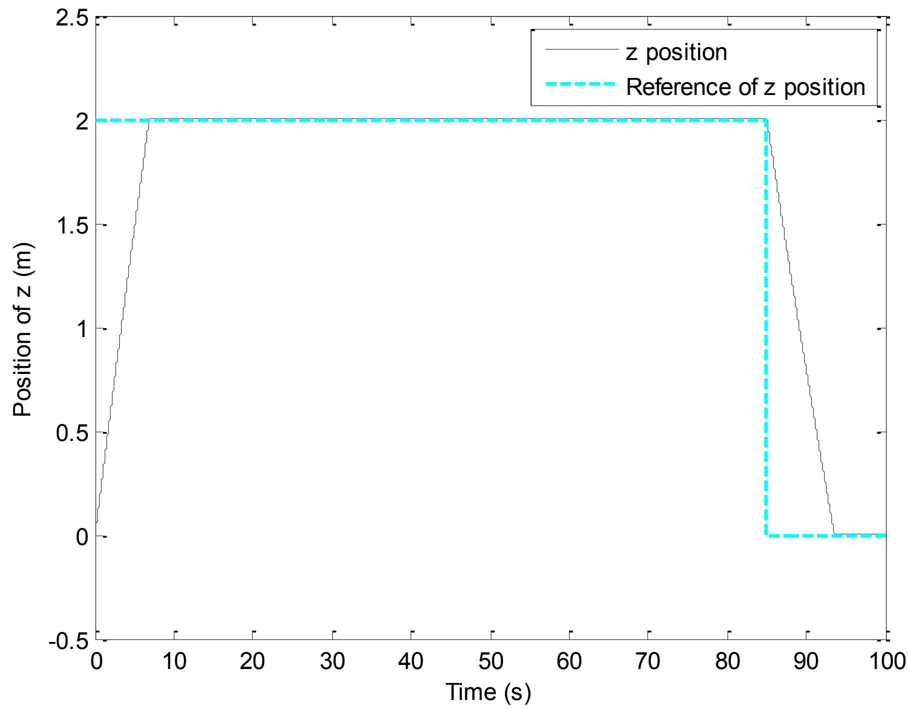

Fig. 5-6. Position tracking in *z* direction

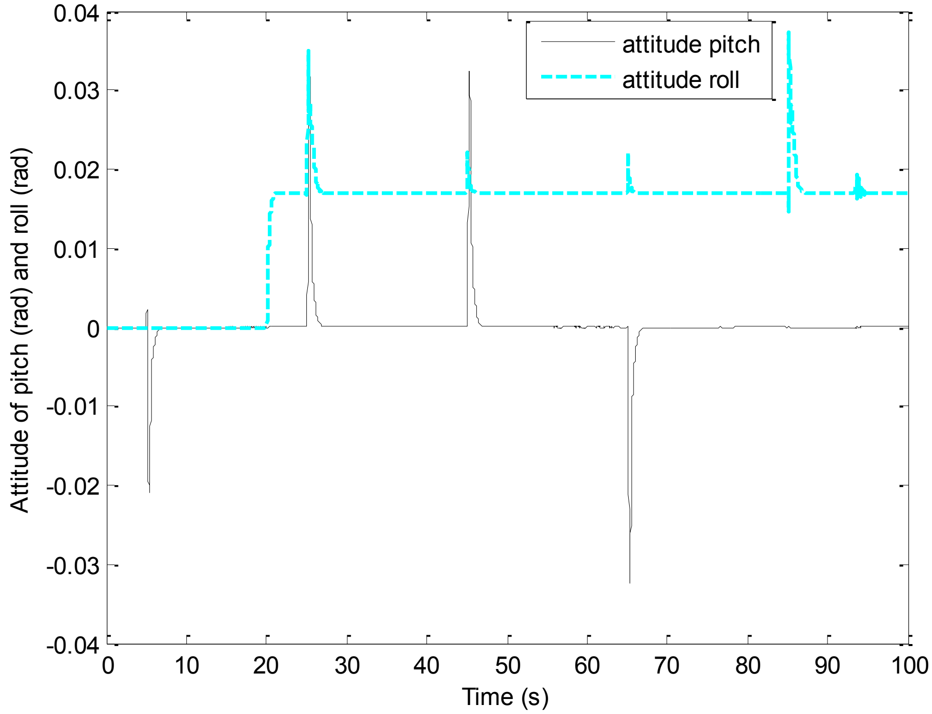

Fig. 5-7. Attitude of pitch and roll angles

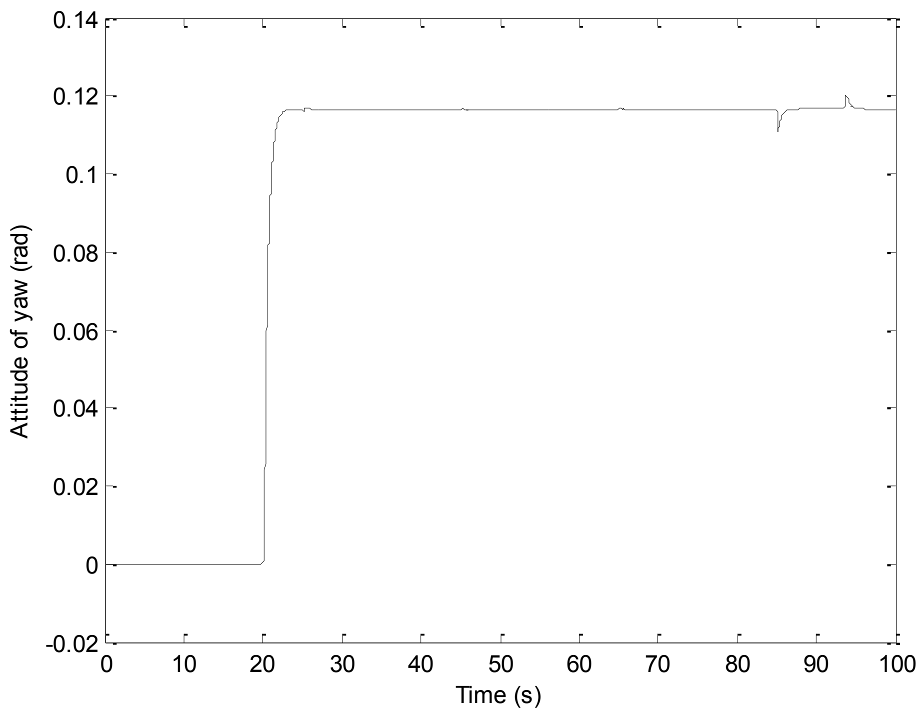

Fig. 5-8. Attitude of yaw angle

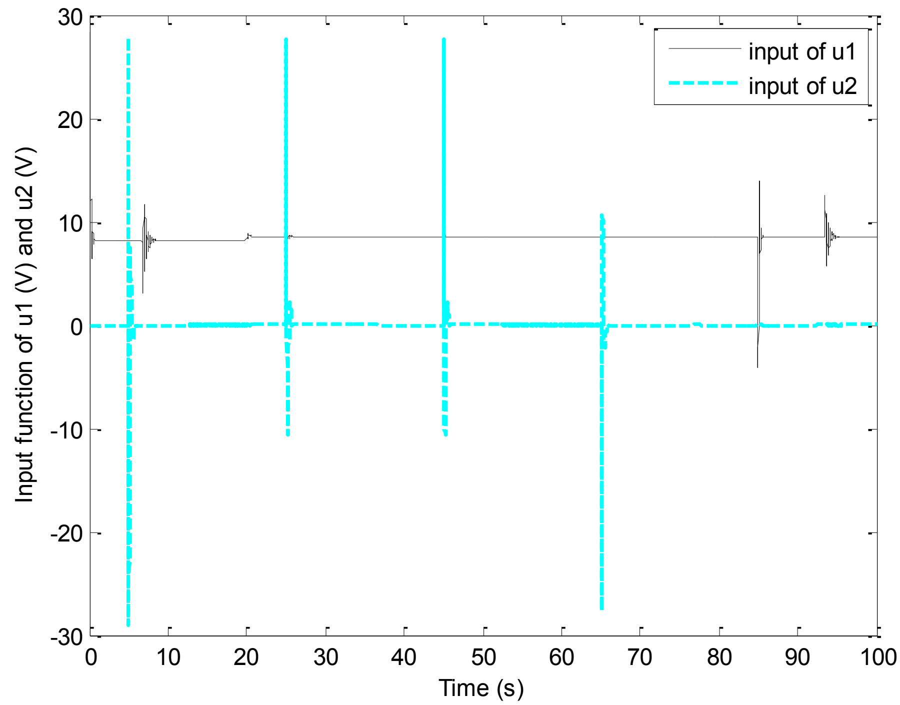

Fig. 5-9. Control inputs of u_1 and u_2

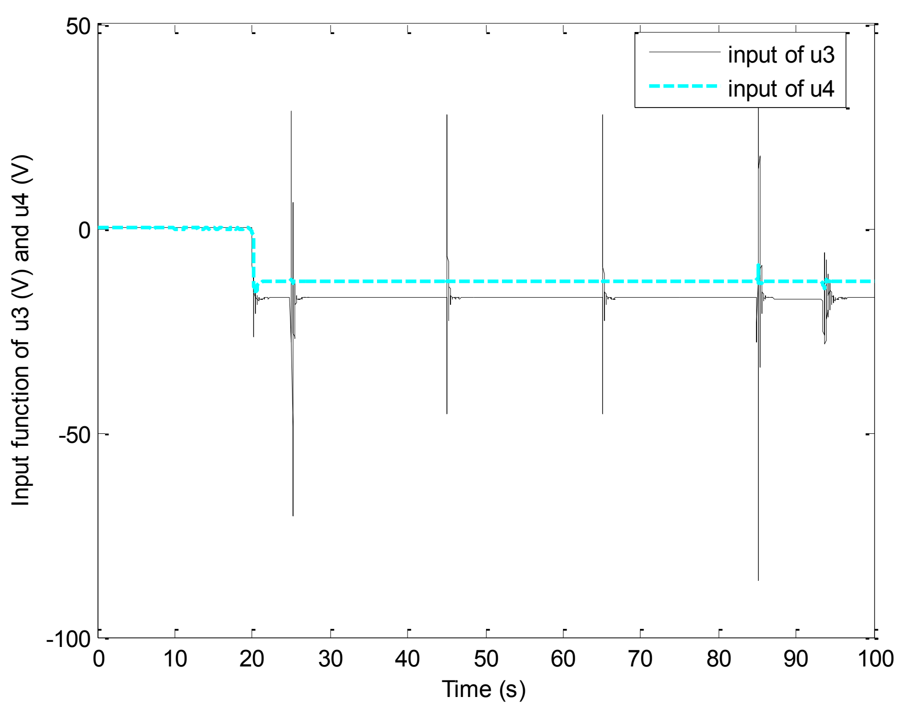

Fig. 5-10. Control inputs of u_3 and u_4

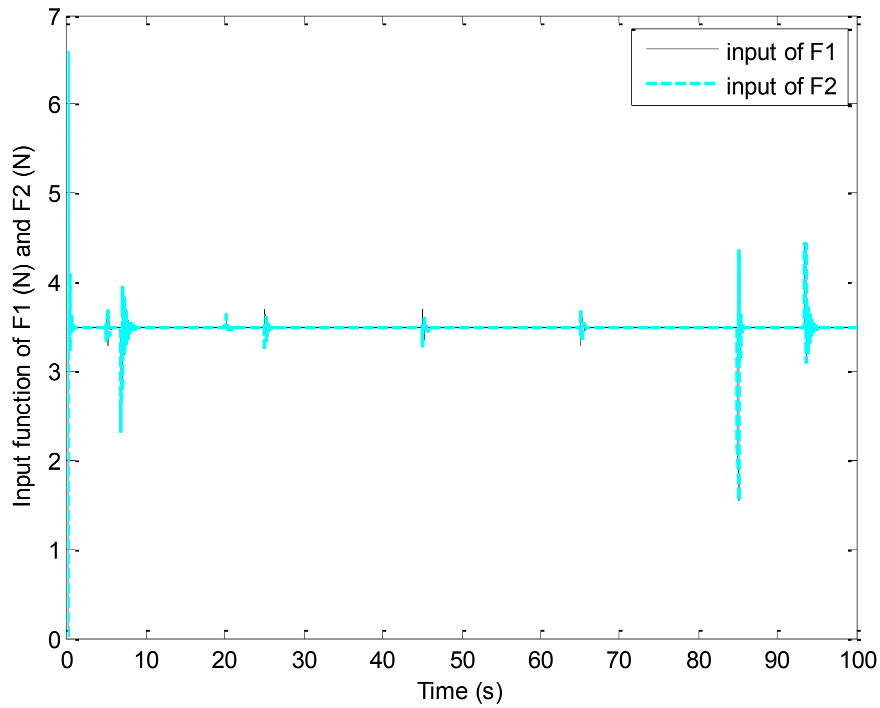

Fig. 5-11. Propellers forces F_1 and F_2

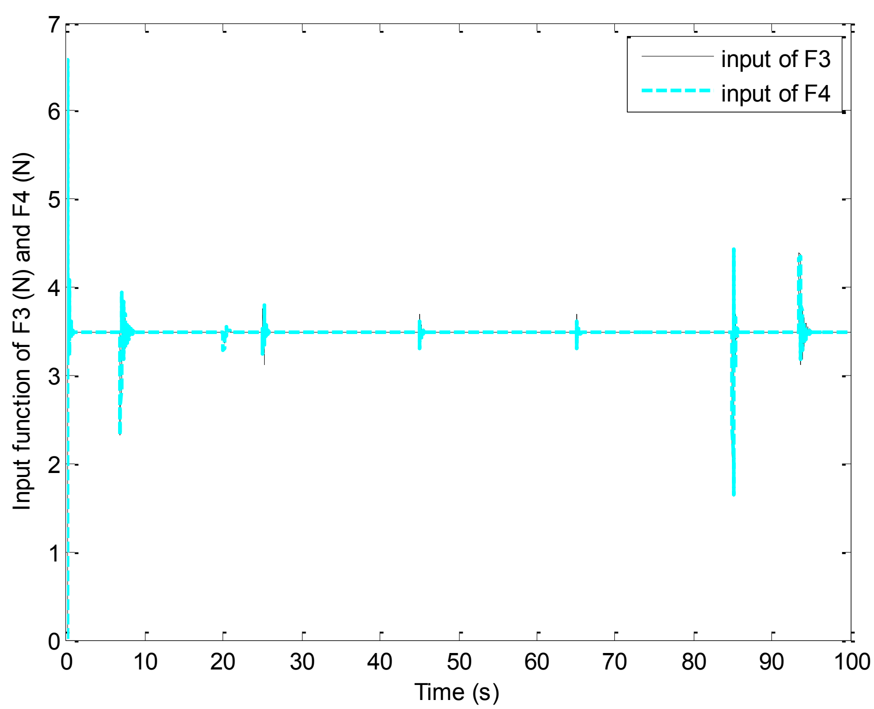

Fig. 5-12. Propellers forces F_3 and F_4

5.2.2. Active Fault-Tolerant Control for the Qball-X4 System

In this section, an AFTC is designed based on SMC technique with the presence of faults, and only actuator faults are considered in the design procedure.

When the actuator faults occur, the system model will be changed. The state equation can be expressed as [48]:

$$\dot{} \quad \boldsymbol{A} \quad t) + \boldsymbol{B}u_i(t) - \boldsymbol{B}k_i(t)u_i(t) \tag{5-4}$$

and

$$\boldsymbol{U}(t) = \begin{bmatrix} u_1(t) & \cdots & \\ \vdots & \ddots & \vdots \\ 0 & \cdots & \end{bmatrix} \qquad \boldsymbol{K} \qquad \begin{matrix} & \cdots & \\ \vdots & \ddots & \vdots \\ & \cdots & \end{matrix} \tag{5-5}$$

where $\boldsymbol{A} \in R^n$, $\boldsymbol{B} \in R^m$, $u_i(t) \in \boldsymbol{U}(t)$, and $k_i(t) \in \boldsymbol{K}(t)$ is the effectiveness gain, with $0 \le k_i(t) \le 1$. If $k_i(t) = 0$, the i^{th} actuator is functioning perfectly, and if $k_i(t) = 1$, the i^{th} actuator has failed completely.

For the same state equation, the above equation can be rearranged as:

$$\dot{} \quad \boldsymbol{A} \quad t) + \boldsymbol{B}(1 - k_i(t))u_i(t) \tag{5-6}$$

Due to the highly coupling feature of the quadrotor system, the control input $u_i(t)$ is related with multiple actuator inputs $F_i(t)$. Therefore, the effectiveness gain needs to be multiplied with the actuator inputs. From equation (4-13), the following equations can be defined as follows:

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ u_4 \end{bmatrix} = \begin{bmatrix} \frac{1}{m} & \frac{1}{m} & \frac{1}{m} & \frac{1}{m} \\ \frac{1}{J_x} & -\frac{1}{J_x} & 0 & 0 \\ 0 & 0 & \frac{1}{J_y} & -\frac{1}{J_y} \\ \frac{1}{J_z} & \frac{1}{J_z} & -\frac{1}{J_z} & -\frac{1}{J_z} \end{bmatrix} \begin{bmatrix} F_1 \\ F_2 \\ F_3 \\ F_4 \end{bmatrix} \tag{5-7}$$

To generalize the above equation, it can be defined as:

$$\boldsymbol{U} = \boldsymbol{\Gamma F} \tag{5-8}$$

where $\boldsymbol{\Gamma}$ is the mapping matrix, $\boldsymbol{U}$ is the control inputs, and $\boldsymbol{F}$ is the actuator inputs.

Since the actuator failure applies directly on $\boldsymbol{F}$, the following relation is satisfied.

$$\boldsymbol{F}_f(t) = \boldsymbol{K}(t)\boldsymbol{F}(t) \tag{5-9}$$

From equation , the new states equation can be obtained as following:

$$\dot{} \qquad \boldsymbol{A} \quad t) + \boldsymbol{B}u_i(t) - \boldsymbol{B}u_{fi}(t) \tag{5-10}$$

where $u_{fi}(t)$ is the control input with fault.

Further, the following equations can be derived:

$$\dot{} \qquad \boldsymbol{A} \quad t) + \boldsymbol{B}(u_i(t) - u_{fi}(t)) \tag{5-11}$$

In general,

$$\dot{\boldsymbol{X}} \qquad \boldsymbol{AX}(t) + \boldsymbol{B}(\boldsymbol{U}(t) - \boldsymbol{\Gamma F}_f(t)) \tag{5-12}$$

$$\dot{\boldsymbol{X}} \qquad \boldsymbol{AX}(t) + \boldsymbol{B}(\boldsymbol{U}(t) - \boldsymbol{\Gamma}(\boldsymbol{K}(t)(\boldsymbol{\Gamma}^{-1}\boldsymbol{U}_f(t)))) \tag{5-13}$$

where $\boldsymbol{U}(t)$ indicates the new control inputs of the quadrotor.

With the redefined states after the actuator faults occurrence as follows:

$$\dot{\boldsymbol{X}} = \begin{bmatrix} x_{f1} & x_{f2} & x_{f3} & x_{f4} & x_{f5} & x_{f6} & x_{f7} & x_{f8} \end{bmatrix}^T \tag{5-14}$$

the state equations from (4-13) can also be rearranged into the following format:

$$\dot{\boldsymbol{X}} = \begin{bmatrix} x_{f2} \\ -g - d_z x_{f2} + (\cos\phi\cos\theta)(u_1 - u_{f1}) \\ x_{f4} \\ \dfrac{-J_z}{J_x} x_{f6} x_{f8} - \dfrac{J_r x_{f6}\Omega}{J_x} - d_\theta x_{f4} + l(u_2 - u_{f2}) \\ x_{f6} \\ \dfrac{-J_x}{J_y} x_{f4} x_{f8} + \dfrac{J_r x_{f4}\Omega}{J_y} - d_\phi x_{f6} + l(u_3 - u_{f3}) \\ x_{f8} \\ \dfrac{J_x - J_y}{J_z} x_{f4} x_{f6} - d_\psi x_{f8} + c(u_4 - u_{f4}) \end{bmatrix} \tag{5-15}$$

with

$$\boldsymbol{K}(t) = \begin{bmatrix} k_1(t) & 0 & 0 & 0 \\ 0 & k_2(t) & 0 & 0 \\ 0 & 0 & k_3(t) & 0 \\ 0 & 0 & 0 & k_4(t) \end{bmatrix} \tag{5-16}$$

where $0 \le k_i(t) \le 1$.

From equation (4-14), the derivative of sliding surface has been changed to

(5-17)

In expansion,

$$\boldsymbol{\Gamma} \quad \boldsymbol{K} \quad \boldsymbol{\Gamma} \quad \boldsymbol{U} \tag{5-18}$$

where x_i is the reference input.

Then, following the same procedure, if the fourth actuator failed, the approximation of control input $\hat{u}_{f3}$ can be derived from equations and .

$$\hat{u}_{f3} = \frac{2J_y}{l(k_4 J_y - J_y)}(- \quad \frac{-mk_4)}{4J_y}u_{f1} - \frac{l(k_4 J_z - J_z)}{4J_y}u_{f4} + \frac{J_z - J_x}{J_y}x_{f4}x_{f8} + \frac{J_r x_{f4}\Omega}{J_y} - d_\phi x_{f6} - \lambda_{a5}(\quad e_\phi) \tag{5-19}$$

The sliding condition can be easily proven as

$$\frac{1}{2}\frac{d}{dt}s^2 = s\dot{} \tag{5-20}$$

Therefore, the control system is stable.

In general, the overall control inputs can be expressed as:

$$u_{fi} = \hat{u}_{fi} + k_{asi} sign(s_{fi}) \tag{5-21}$$

where λ_{ai}, k_{asi}, and k_{api} are all positive gains of the changed sliding mode based fault-tolerant control.

The simulated tracking task is the same as in passive fault-tolerant control, and with the same fault scenario. Since the system is in active mode, there are two separate controllers in use for normal condition and fault condition separately. The first controller is from equation (4-24) used before the occurrence of fault, and the second one is from equation used after the occurrence of fault on the fourth propeller. In the combination of two different controllers, the overall performance can be improved, since each controller will handle only one situation. The fault is the same as in PFTC about 15% force loss of the fourth actuator. The results have shown the performance of the tracking is excellent, and it can be seen that the controller responds to the fault more accurate from Fig. 5-19 to Fig. 5-22.

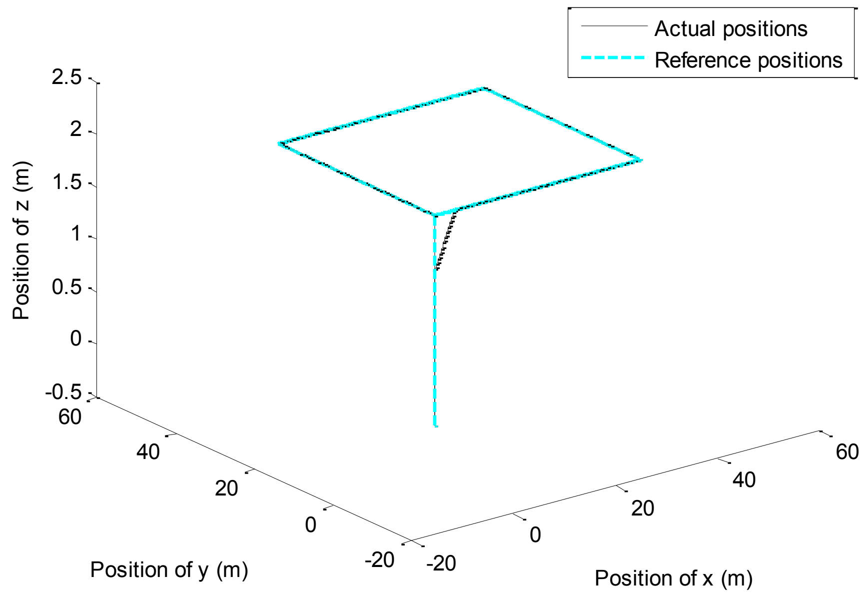

Fig. 5-13. 3-dimensional path tracking

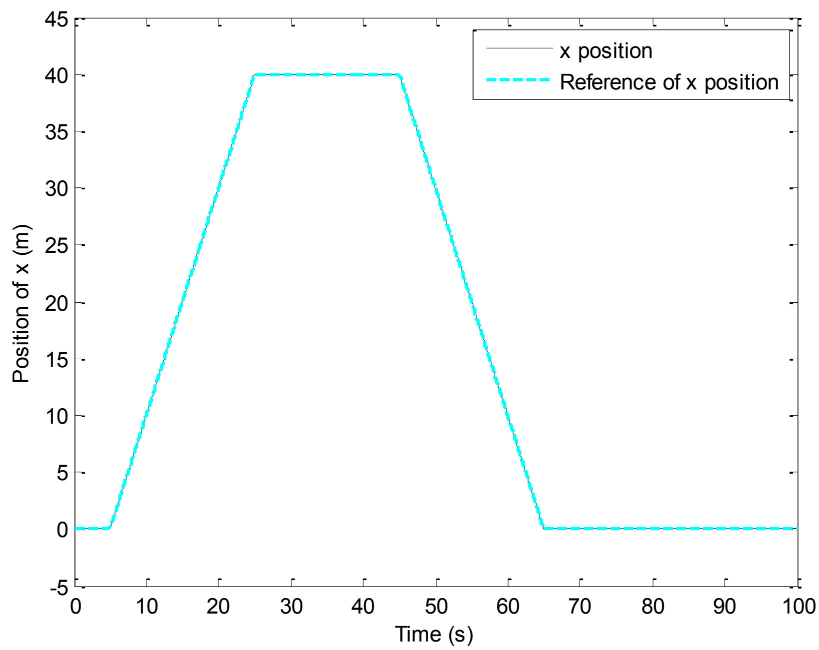

Fig. 5-14. Position tracking in *x* direction

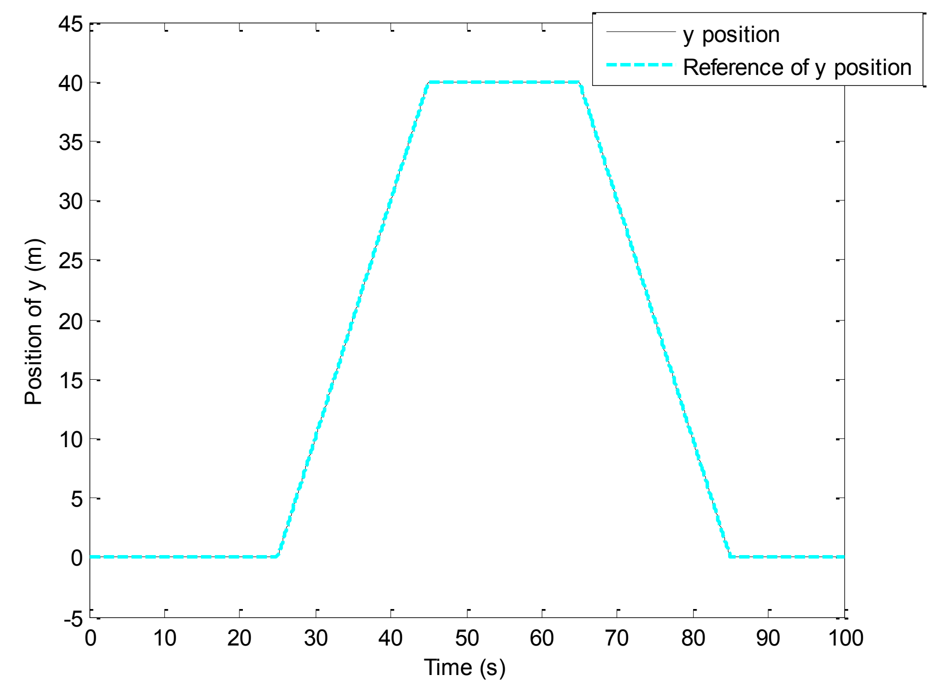

Fig. 5-15. Position tracking in *y* direction

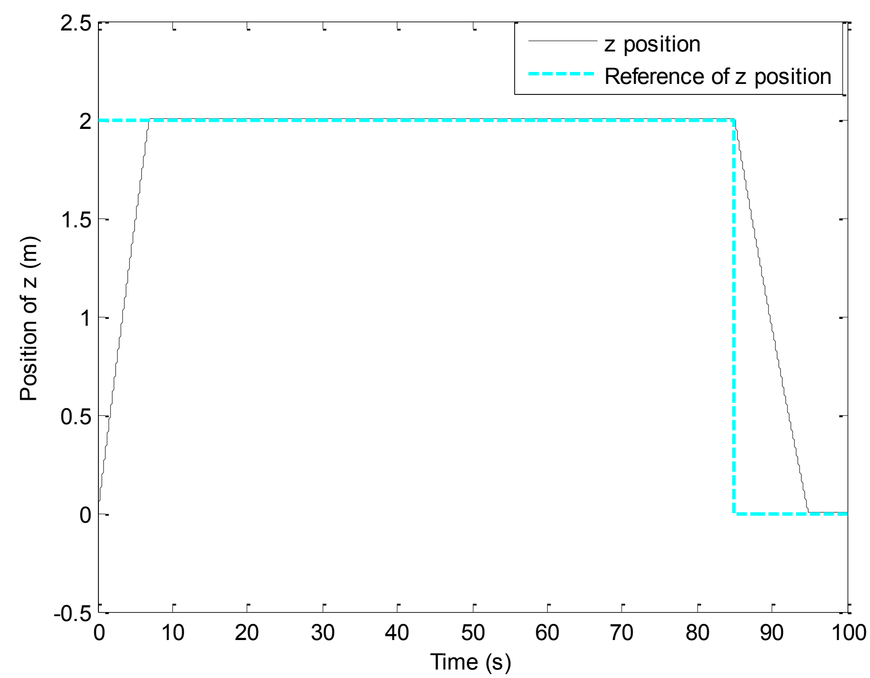

Fig. 5-16. Position tracking in *z* direction

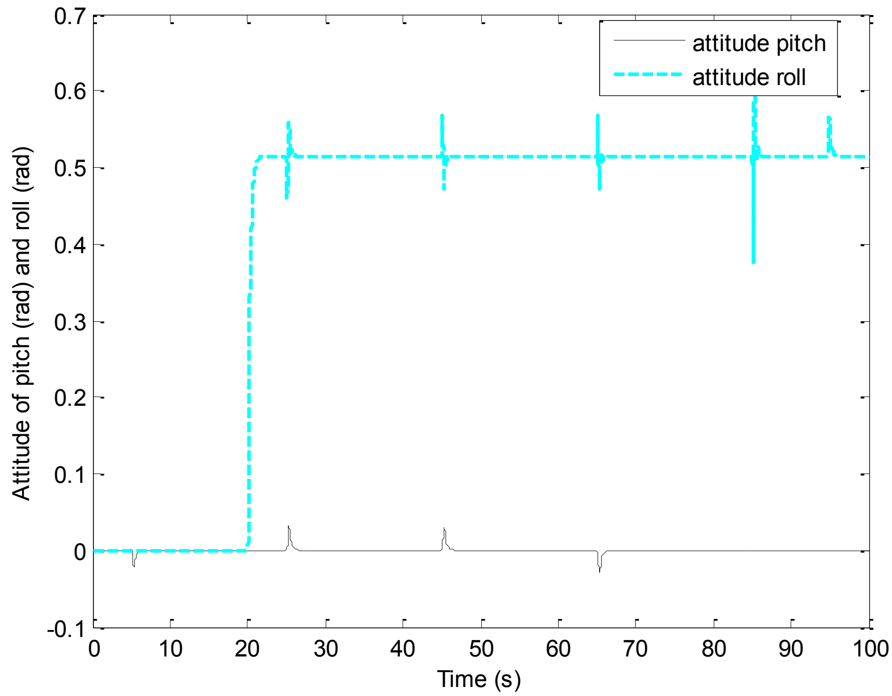

Fig. 5-17. Attitude of pitch and roll angles

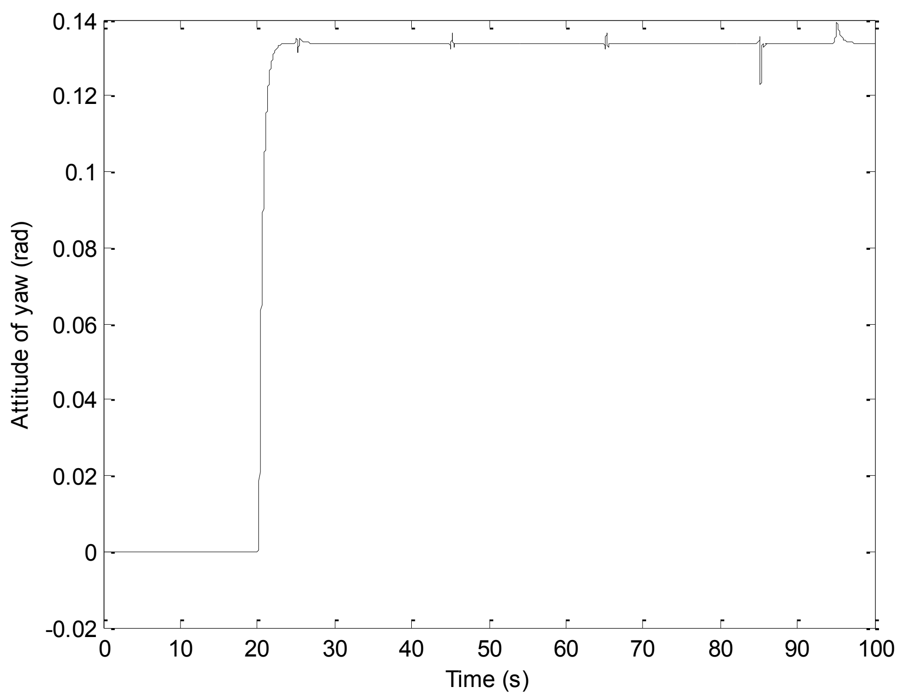

Fig. 5-18. Attitude of yaw angle

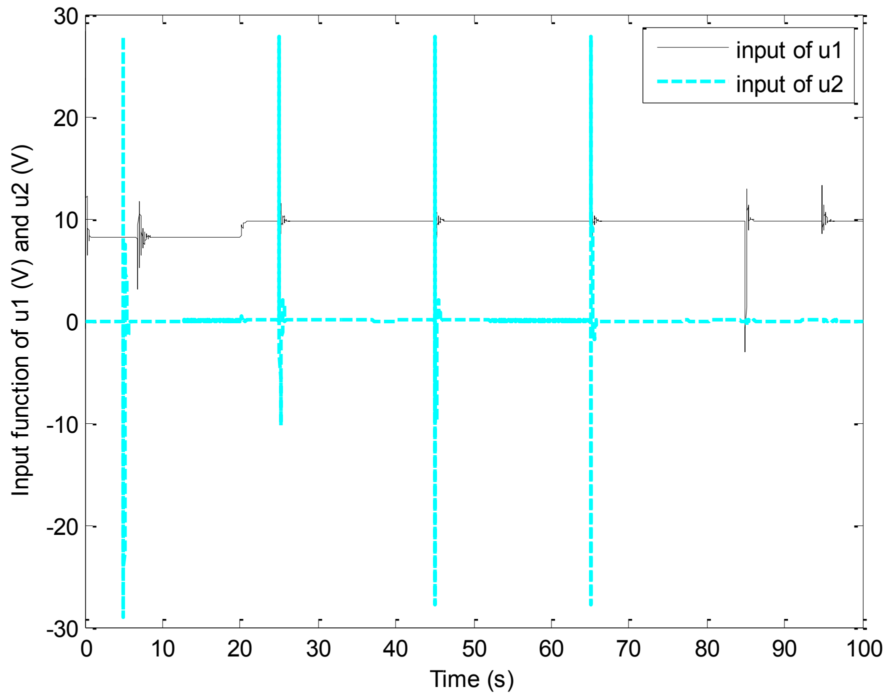

Fig. 5-19. Control inputs of u_1 and u_2

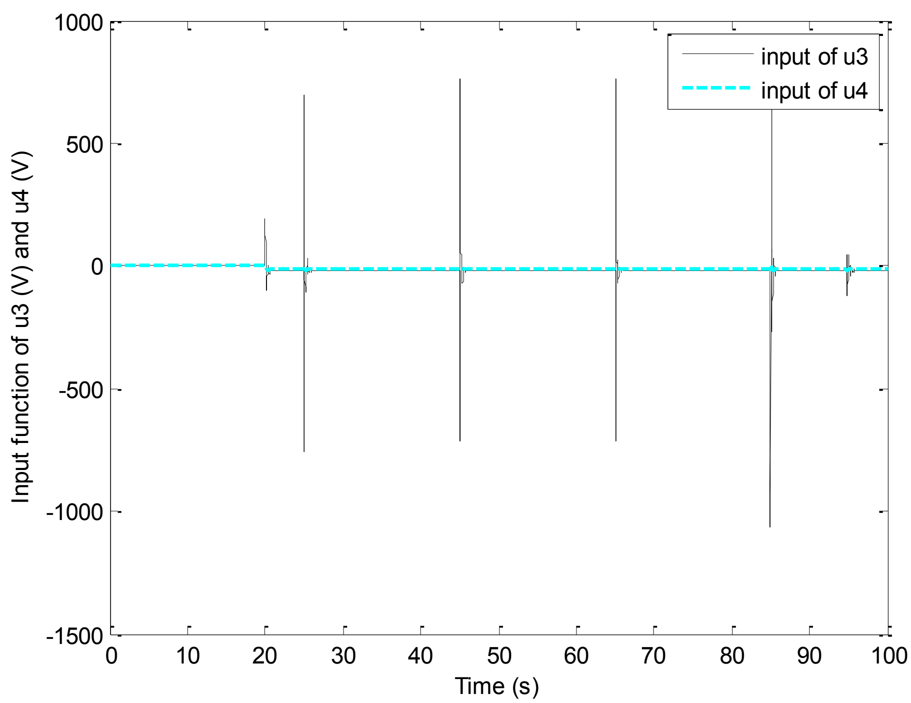

Fig. 5-20. Control inputs of u_3 and u_4

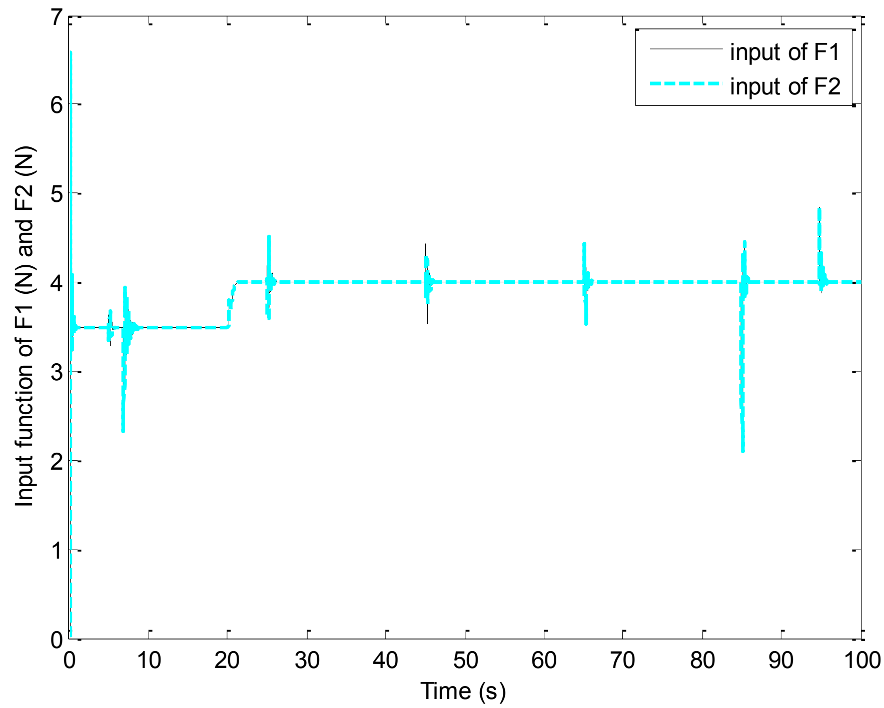

Fig. 5-21. Propellers forces F_1 and F_2

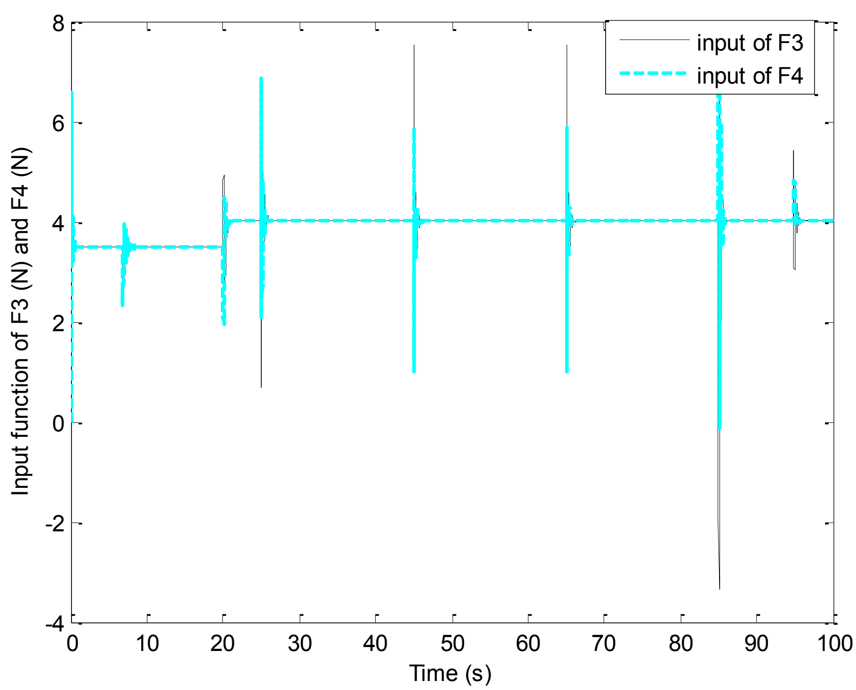

Fig. 5-22. Propellers forces F_3 and F_4

5.3. Experimental Testing Results

The following experiments are done in the same indoor environment and by the same equipments mentioned in Chapter 3. The only difference is that an extra device is used for generating fault scenario during real-time flight to be used for testing fault-tolerant control strategy. The device is shown below to break the fourth propeller blade during the flight. Thus, the fault can be generated. The behaviours of PFTC and AFTC can be tested based on this test bed.

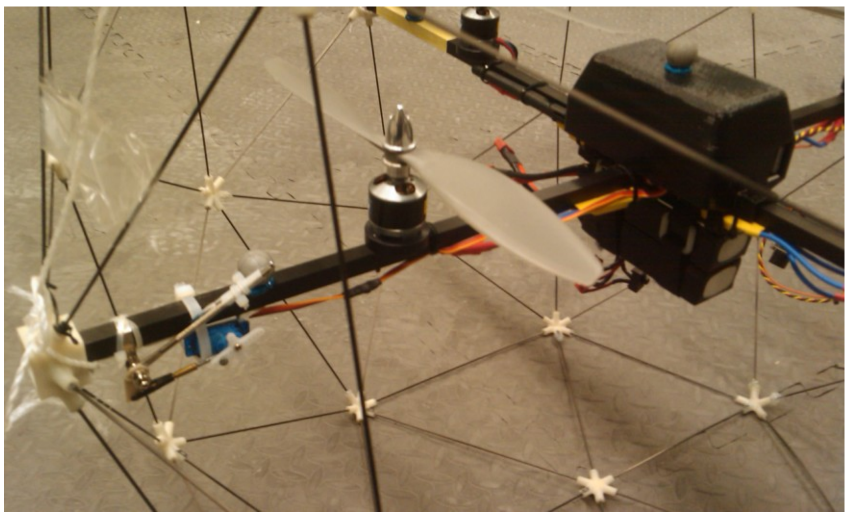

Fig. 5-23. The mechanism to injecting damaged propeller during flight

5.3.1. Passive Fault-Tolerant Control

Using the same experimental setup, the PFTC controller can be implemented and fully tested. The following figures show the performance of the Qball-X4 using the sliding mode-based PFTC controller. The task is to maintain the original tracking of the square $1.5 \times 1.5 m^2$ after the fault occurrence. The results have shown the test is successful, except for a small disturbance caused by the communication delay of the camera system in Fig. 5-27, around 60 seconds. The robustness of PFTC is a bit weak after the fault occurred, since there is only one controller in effect for all the situations. A trade off has been made, thus the performance is worse in any scenarios.

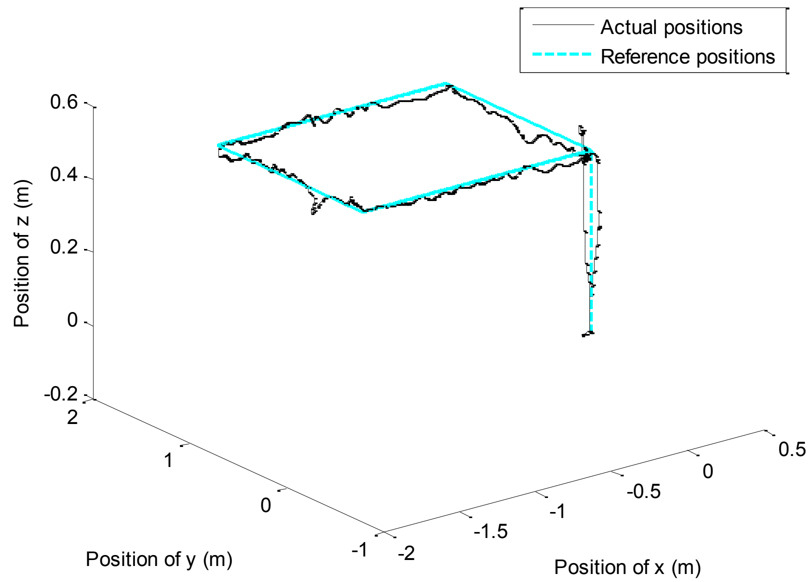

Fig. 5-24. 3-dimensional path tracking

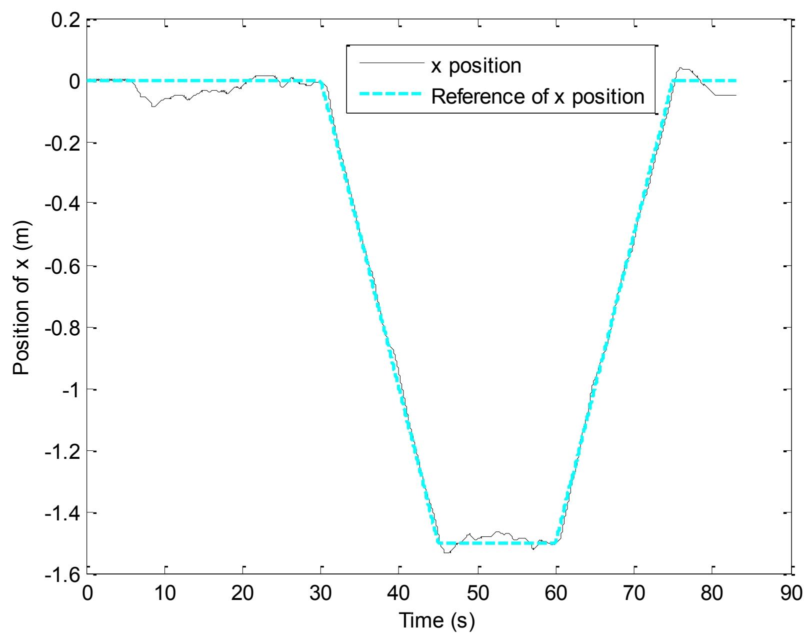

Fig. 5-25. Position tracking in *x* direction

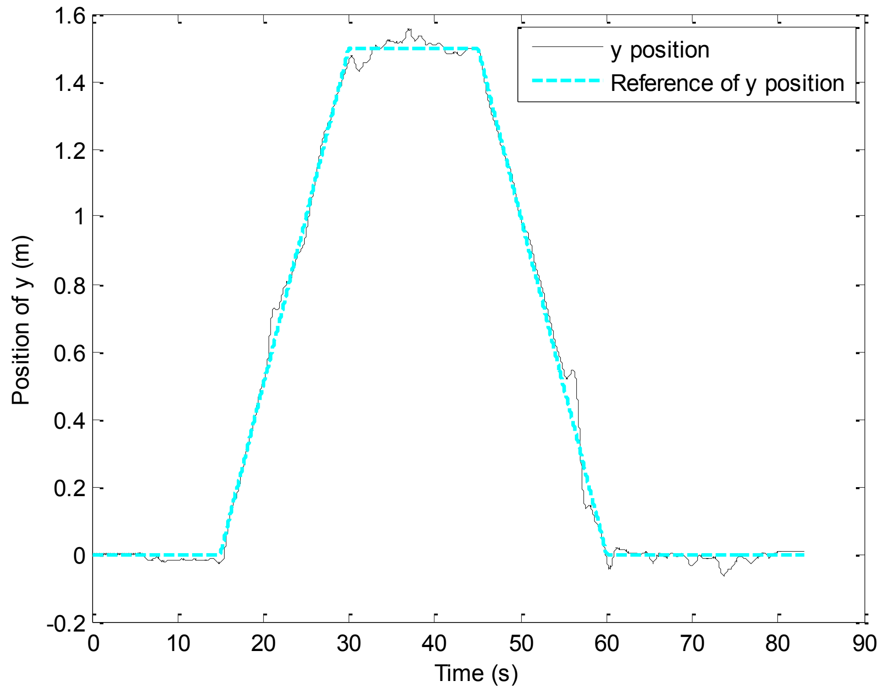

Fig. 5-26. Position tracking in *y* direction

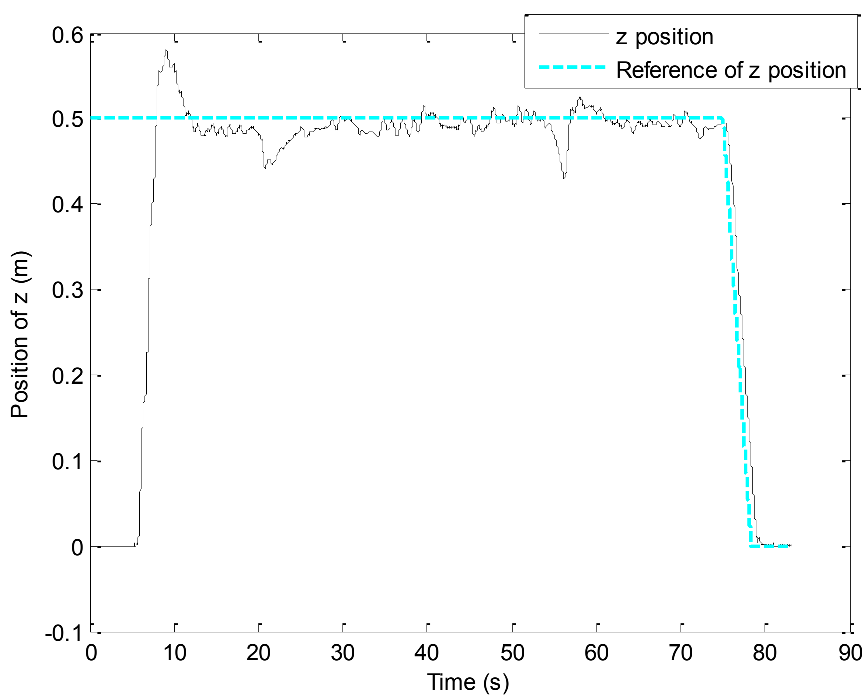

Fig. 5-27. Position tracking in *z* direction

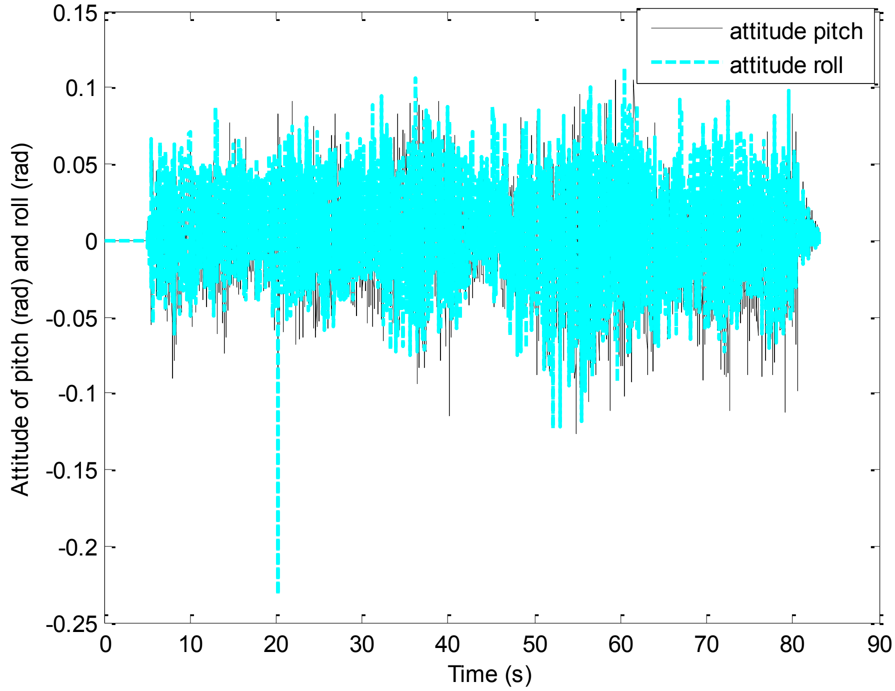

Fig. 5-28. Attitude of pitch and roll angles

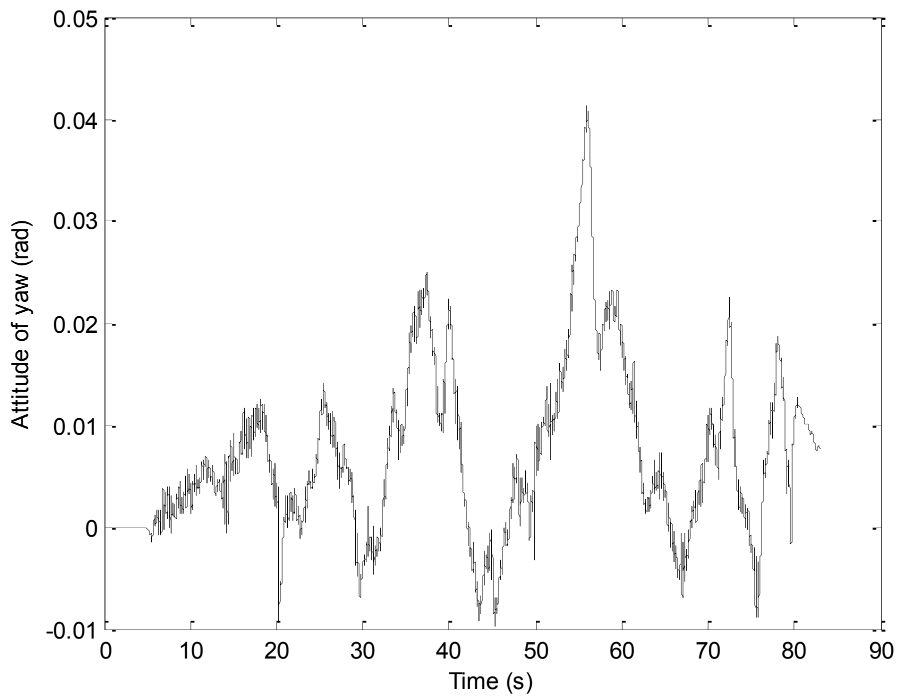

Fig. 5-29. Attitude of yaw angle

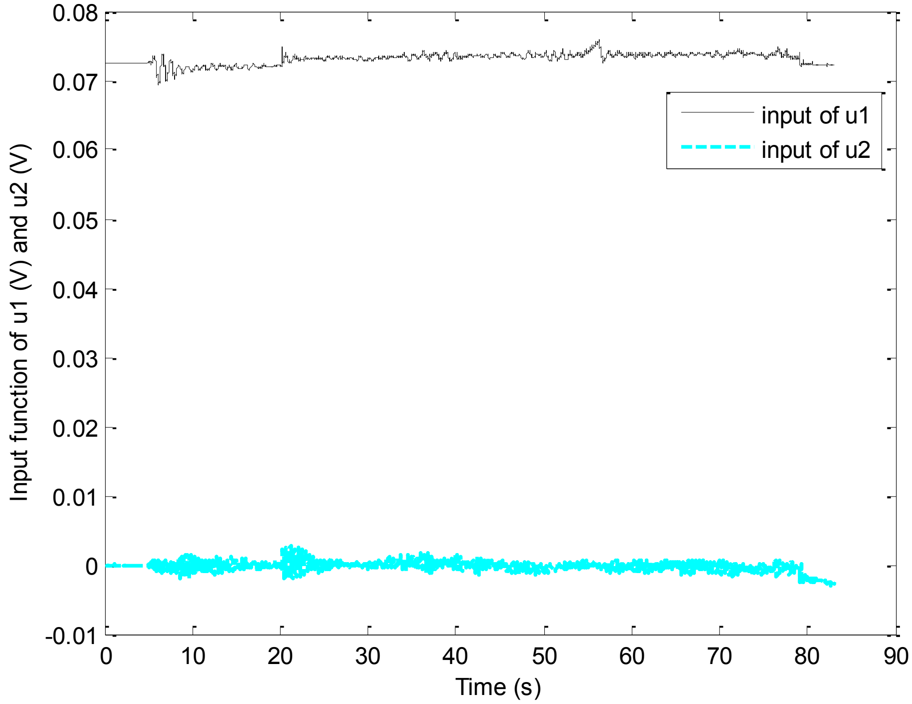

Fig. 5-30. Control inputs of u_1 and u_2

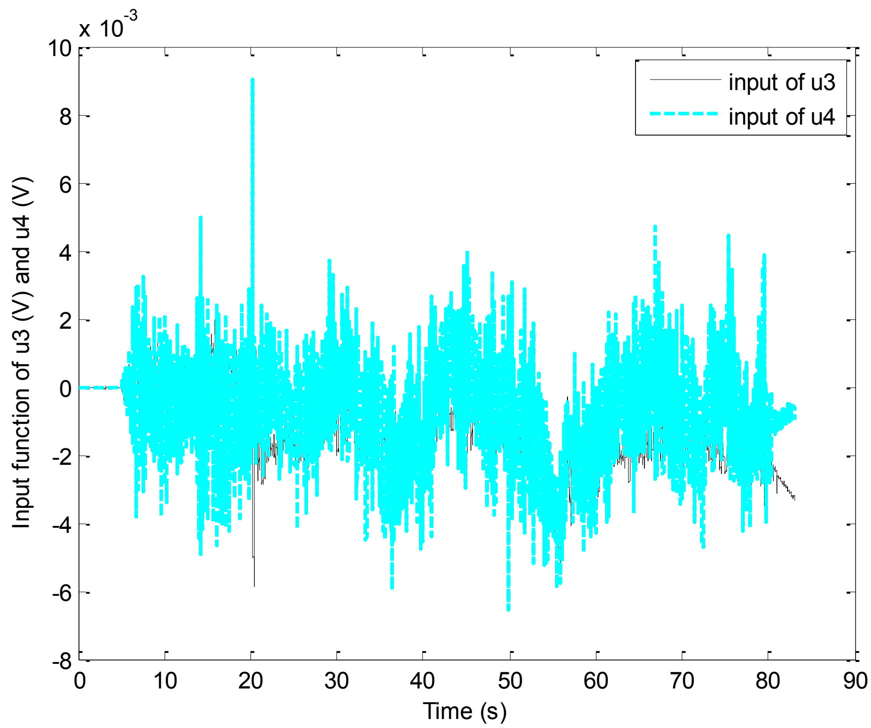

Fig. 5-31. Control inputs of u_3 and u_4

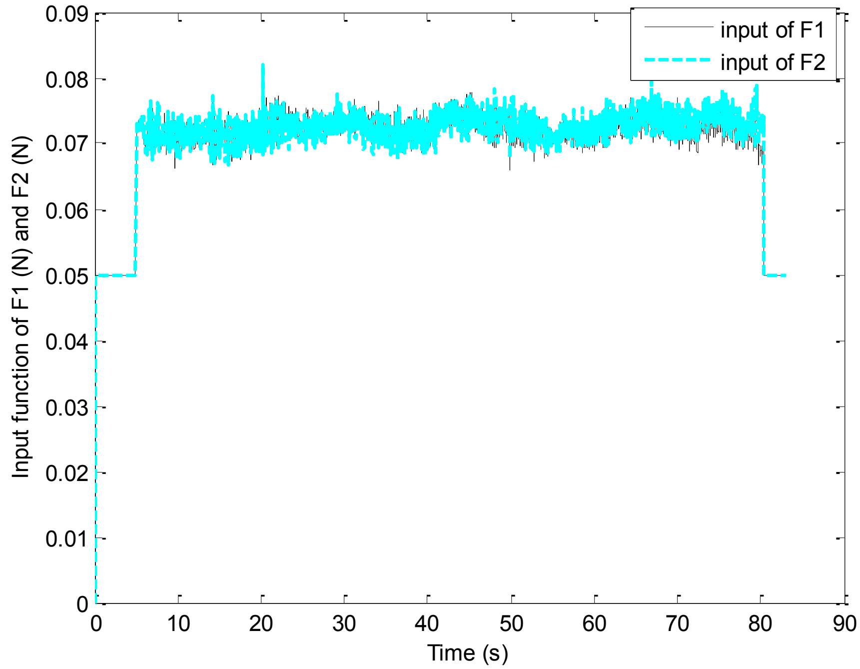

Fig. 5-32. Propellers forces F_1 and F_2

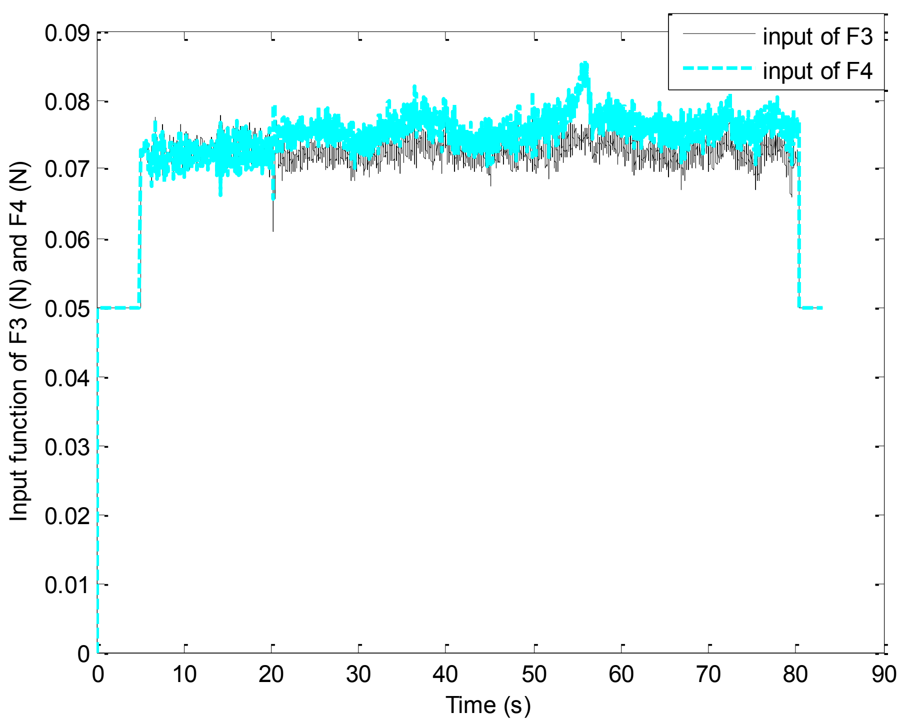

Fig. 5-33. Propellers forces F_3 and F_4

5.3.2. Active Fault-Tolerant Control

The following results show that the active fault-tolerant control system has been tested successfully. The robustness of the AFTC has been proven stronger from the comparison of Fig. 5-27 and Fig. 5-37 in altitude tracking. In the same environment, with a new designed controller handling the faulty situations, no control trade off needs to be made. Therefore, active control is more robust than passive control on dealing with faults.

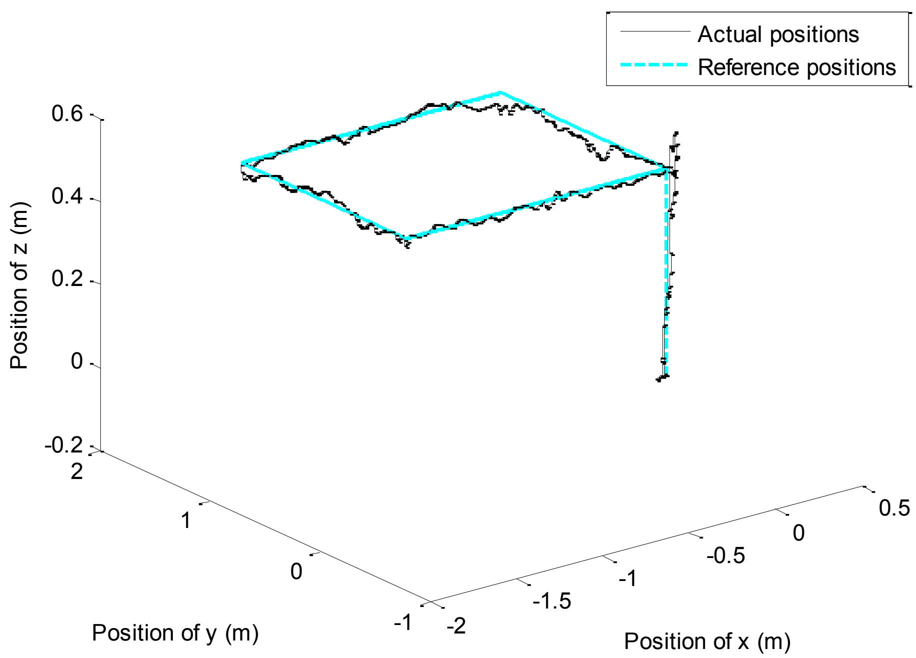

Fig. 5-34. 3-dimensional path tracking

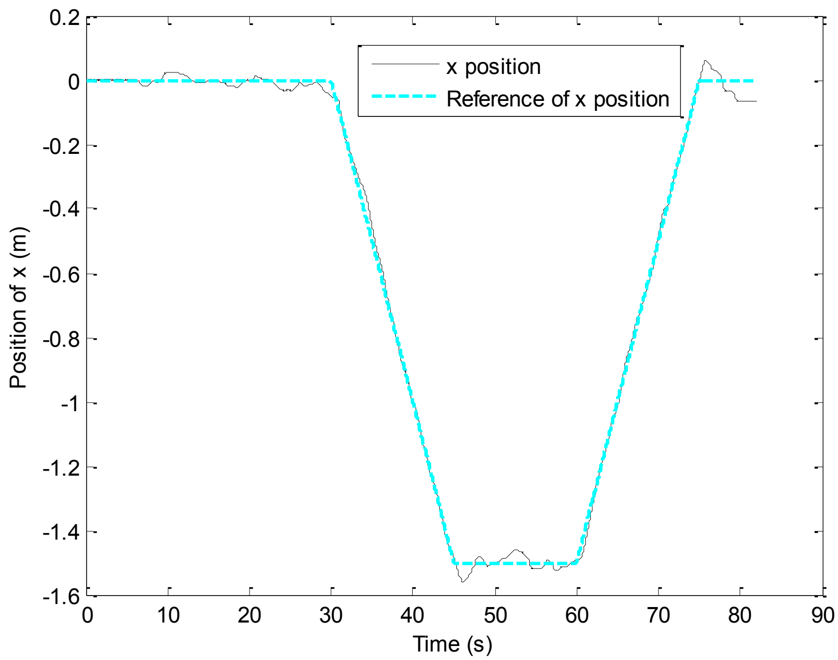

Fig. 5-35. Position tracking in *x* direction

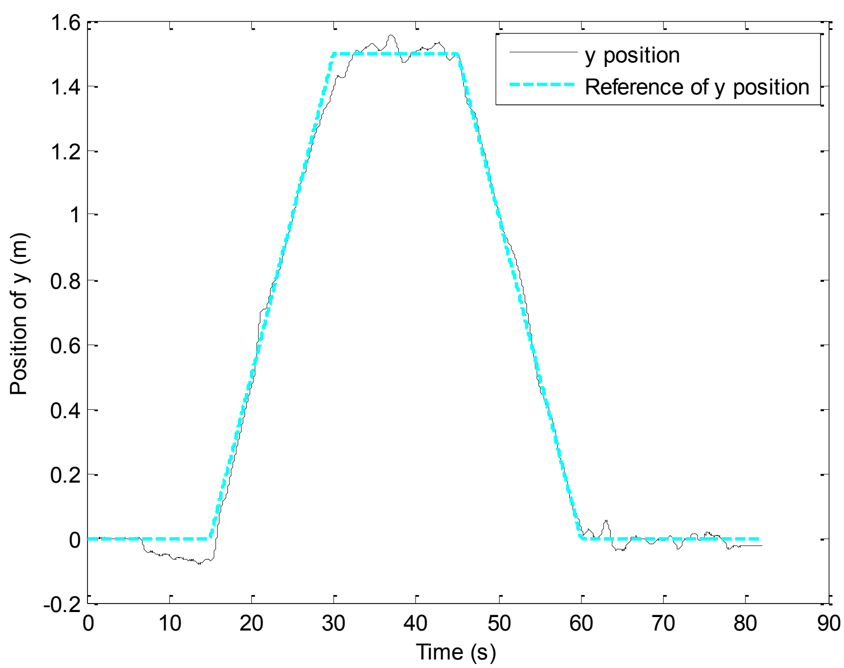

Fig. 5-36. Position tracking in *y* direction

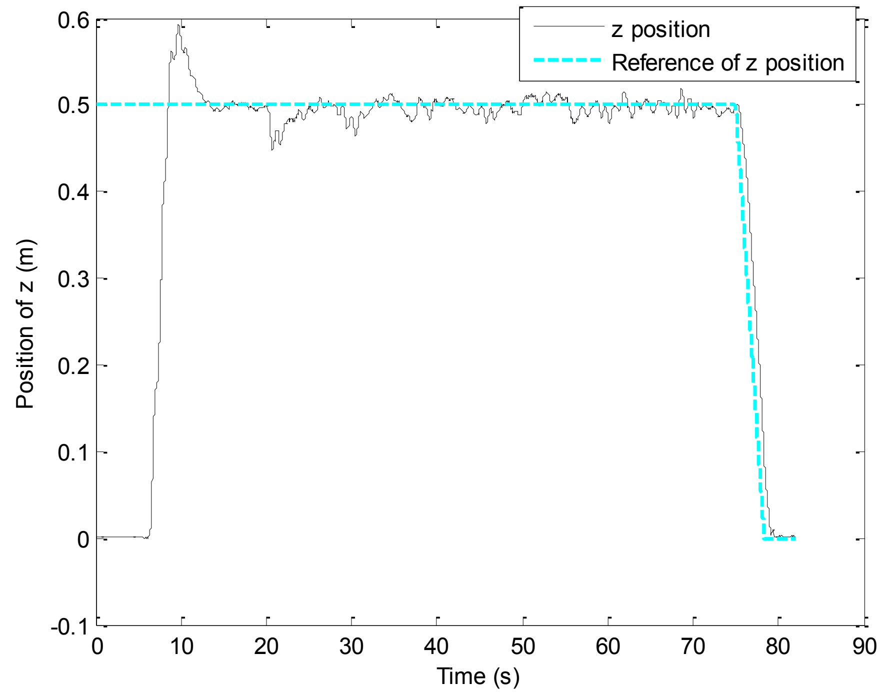

Fig. 5-37. Position tracking in *z* direction

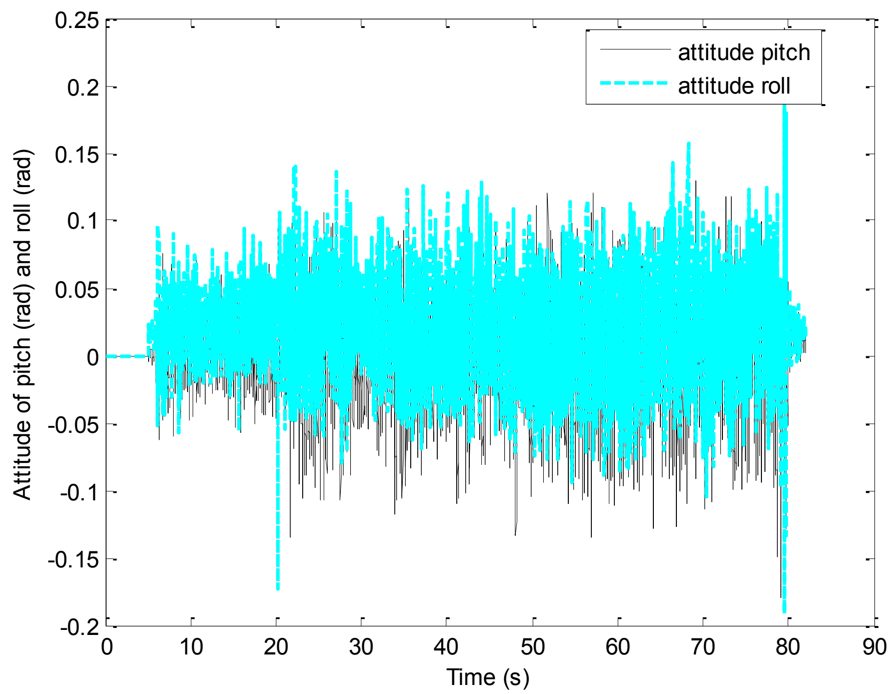

Fig. 5-38. Attitude of pitch and roll angles

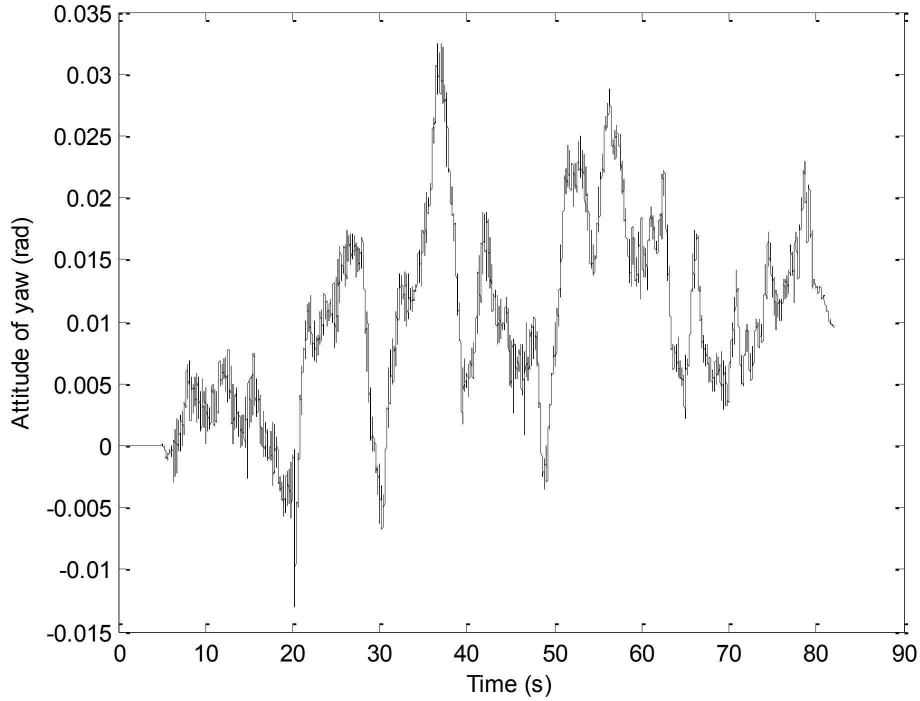

Fig. 5-39. Attitude of yaw angle

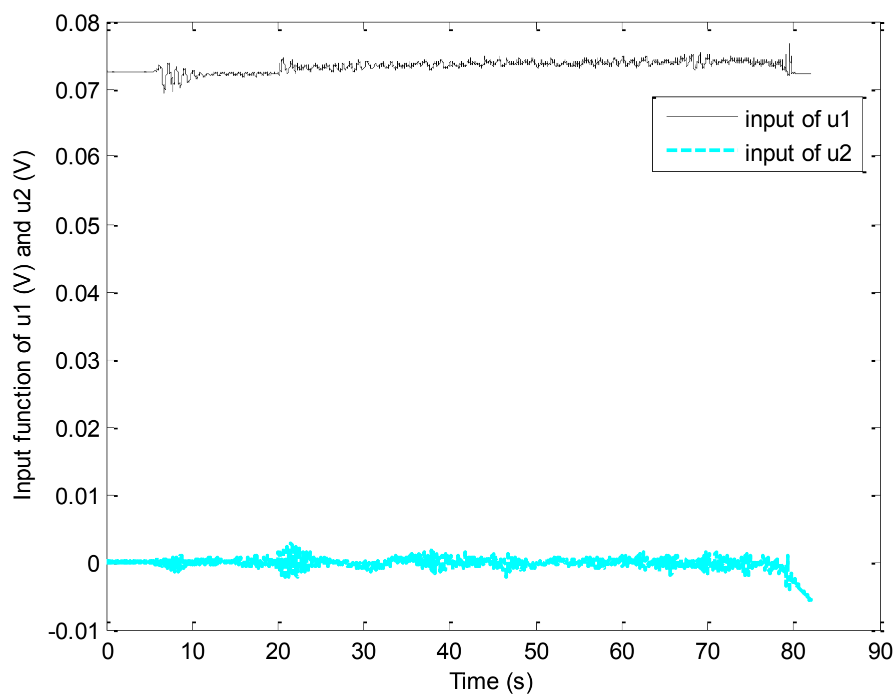

Fig. 5-40. Control inputs of u_1 and u_2

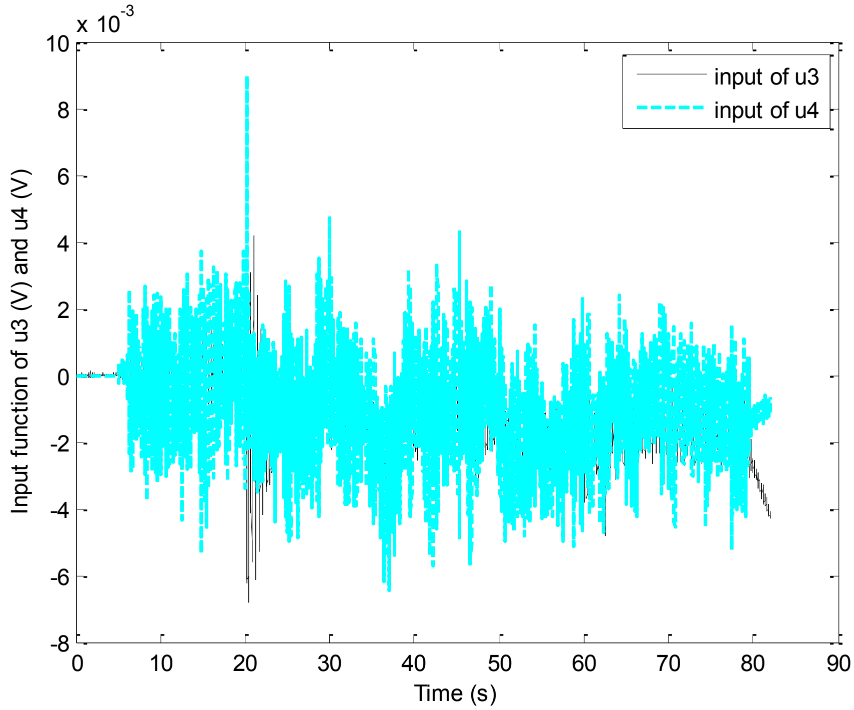

Fig. 5-41. Control inputs of u_3 and u_4

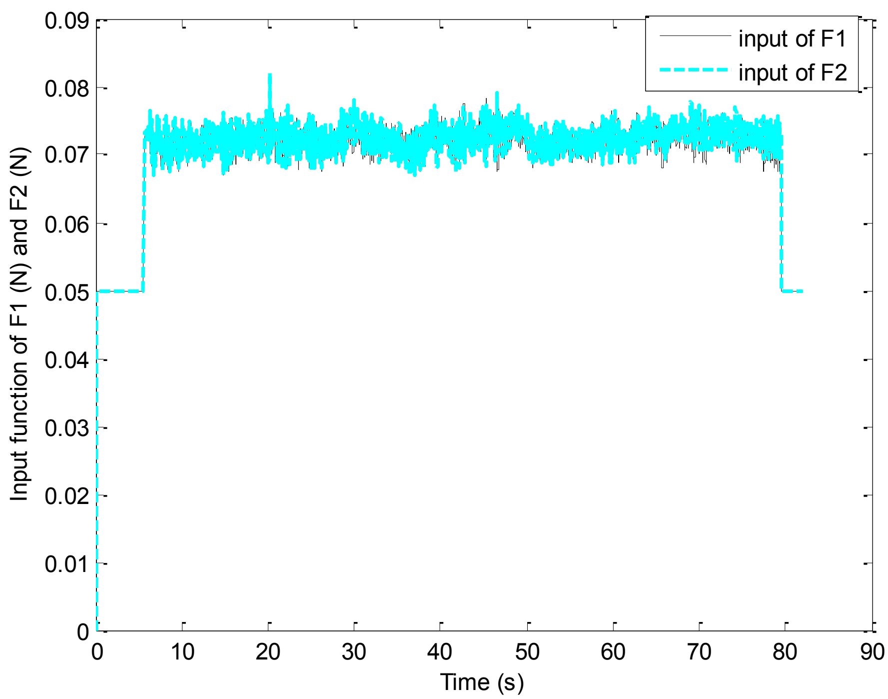

Fig. 5-42. Propellers forces F_1 and F_2

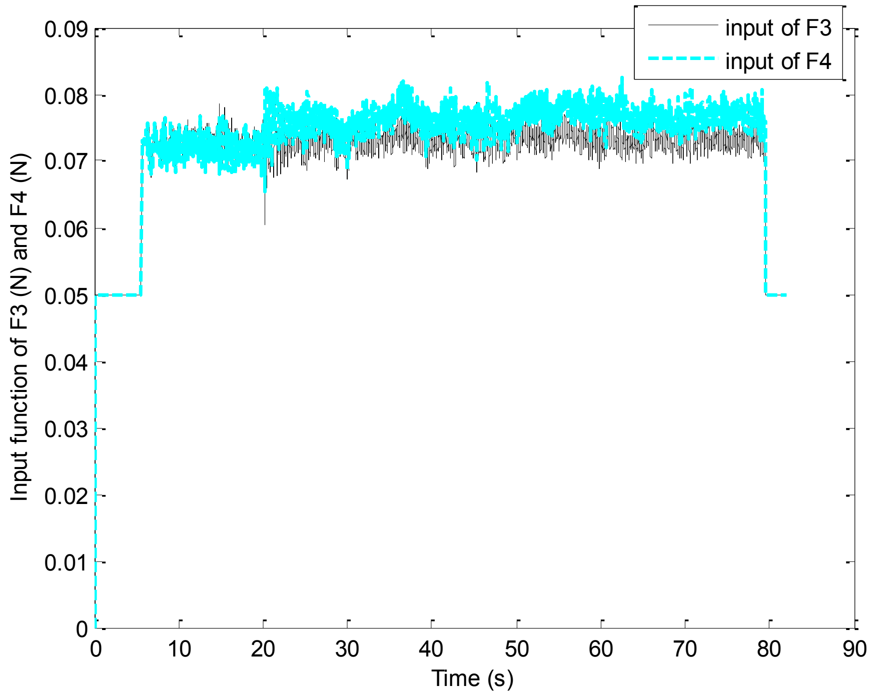

Fig. 5-43. Propellers forces F_3 and F_4

5.3.3. Comparison

The experimental flight testing results have shown in the previous section, and from the figures, it can be learnt that both passive fault-tolerant control and active fault-tolerant control worked properly on the Qball-X4 system. However, if there is extra disturbance other than the faults, passive fault-tolerant control system is more vulnerable to be affected than active fault-tolerant control system. Some numeric comparisons listed in Table 5-1 to 5-3 are used to further demonstrate the difference between these two techniques.

Table 5-1. The comparison of position *x*

Time		5 – 15 (seconds)	15 – 30 (seconds)	30 – 45 (seconds)	45 – 60 (seconds)	60 – 75 (seconds)	Overall
Ref	Mean	0	0	-0.7500	-1.500	-0.7500	-0.5484
PFTC	Mean	-0.0505	-0.0135	-0.7275	-1.4939	-0.7609	-0.5418
	Variance	0.000496	0.000328	0.1878	0.000306	0.1836	0.3702
AFTC	Mean	0.0026	-0.0128	-0.7393	-1.5035	-0.7518	-0.5510
	Variance	0.000172	0.000183	0.1951	0.000531	0.1939	0.3843

Table 5-2. The comparison of position *y*

Time		5 – 15 (seconds)	15 – 30 (seconds)	30 – 45 (seconds)	45 – 60 (seconds)	60 – 75 (seconds)	Overall
Ref	Mean	0	0.7500	1.5000	0.7500	0	0.5484
PFTC	Mean	-0.0157	0.7538	1.4981	0.7631	0.7631	0.5401
	Variance	0.0000428	0.1877	0.000722	0.1851	0.000358	0.3890
AFTC	Mean	-0.0486	0.7238	1.4993	0.7638	0.7638	0.5384
	Variance	0.000621	0.1911	0.0011	0.1895	0.000387	0.3936

Table 5-3. The comparison of position *z*

Time		5 – 15 (seconds)	15 – 30 (seconds)	30 – 45 (seconds)	45 – 60 (seconds)	60 – 75 (seconds)	Overall
Ref	Mean	0.5000	0.5000	0.5000	0.5000	0.5000	0.4672
PFTC	Mean	0.4221	0.4802	0.4926	0.4953	0.4942	0.4190
	Variance	0.0316	0.000189	0.0000657	0.000351	0.0000451	0.0288
AFTC	Mean	0.3971	0.4903	0.4962	0.4979	0.4970	0.4252
	Variance	0.0424	0.000162	0.0000943	0.0000858	0.0000546	0.0286

5.4. Summary

Based on what have been achieved in Chapter 4, SMC has been chosen as the best candidate for fault-tolerant control of the Qball-X4 UAV test-bed. Using the same structure as designed in Chapter 4, with all the redesigned control gains and saturation function, PFTC has been tested successfully in the experiments. For AFTC, by eliminating the lost force from actuator, a new control structure has been designed. With the new controller, the tracking performance of AFTC has been shown excellent. The experimental figures and numerical tables show AFTC is more robust than PFTC on handling extra disturbances.

6. Conclusions and Future Work

In this thesis, Feedback Linearization Control (FLC), Sliding Mode Control (SMC), and Backstepping Control (BSC) have been discussed in details from basic theories to designs with real applications to the Qball-X4 UAV. They have been investigated thoroughly to develop three different controllers that can be used on the Qball-X4 system and fully tested under different flight conditions. The goal is to design a practical controller, thus there is only one assumption as attitude yaw is zero. This assumption is practically possible and has been tested both in simulations and experiments. The results show all three controllers work properly and all can deal with some noisy conditions. SMC is the most robust controller and provides the best tracking performance as expected. FLC and BSC behave equally in general. A comparison has shown each control algorithm has its own advantages and disadvantages, which can be used as a future reference when designing another controller in practice. The model parameters have been identified as well, which can be used as another reference in experiments.

Based on the simulation and experimental results of passive fault-tolerant control and active fault-tolerant control strategies using SMC, it can be seen that the designs of both passive fault-tolerant control system and active fault-tolerant control system has been proven appropriate for Qball-X4 system. Both controllers worked properly. According to the theories, passive fault-tolerant control has a trade off for both faulty situation and fault-free situation. Hence, the performance of PTFC is supposed to be imperfect under each situation, which has been proven by the experiment results. If there is an extra disturbance, the control system will be affected easily. However, active fault-tolerant control can solve this problem by adding two separate controllers in the system under normal and fault flight conditions respectively. Once the fault occurs, the control system will switch to the controller that is designed to compensate the effects due to faults. Thus, the performance of the control system under both fault-free and fault situations can be optimized to the maximum.

Future work will be trying to improve the robustness of each controller in the thesis. A combination of different control methods can be taken into consideration, such as sliding mode control with backstepping control. This can maximally eliminate the disadvantages of each controller working alone. Also, for active fault-tolerant control, a relatively precise fault detection and diagnosis scheme should be considered to be added into the overall control system.

References

[2] D. Lee, H. J. Kim, and S. Sastry, "Feedback linearization vs adaptive sliding mode control for a quadrotor helicopter," *International Journal of Control, Automation, and Systems,* pp. 419-528, December 2008.

[3] S. A. Al-Hiddabi, "Quadrotor control using feedback linearization with dynamic extension," in *Proceeding of the 6th International Symposium on Mechatronics and its Applications (ISMA09)*, March 24-26, 2009.

[4] D. H. Kimm, J. H. Oh, "Tracking control of a two-wheeled mobile robot using input-output linearization," *Control Engineering Practice*, vol. 7, pp. 369-373, September 1998.

[5] A. Mokhtari, A. Benallegue, and B. Daachi, "Robust feedback linearization and GH_{∞} control for a quadrotor unmanned aerial vehicle," *Journal of Electrical Engineering*, vol. 57, no. 1, pp. 20-27, 2006.

[6] H. Voos, and B. Nourghassemi, "Nonlinear control of stabilized flight and landing for quadrotor UAVs," in *Workshop on Advanced Control and Diagnosis*, Mobile Robotics Lab, University of Applied Sciences, Germany, 2009.

[7] H. Voos, "Nonlinear control of a quadrotor micro-UAV using feedback-linearization," in *Proceedings of the 2009 IEEE International Conference on Mechatronics*, pp 1-6, April 2009.

[8] A. Benallegue, A. Mokhtari, and L. Fridman "High-order sliding-mode observer for a quadrotor UAV," *International Journal of Robust and Nonlinear Control*, vol. 18, no. 4-5, pp 427-440, May 2007.

[9] A. Mokhtari, A. Benallegue, and Y. Orlov, "Exact linearization and sliding mode observer for a quadrotor unmanned aerial vehicle," *International Journal of Robotics and Automation*, vol. 21, pp. 39-49, January 2006.

[10] B. Song, Y. Liu, and C. Fan, "Feedback linearization of the nonlinear model of a small-scale helicopter," *IET - Control Theory and Applications*, vol. 8, no. 3, pp. 301-308, January 2010.

[11] G. Oriolo, A. D. Luca, and M. Vendittelli, "WMR control via dynamic feedback linearization: design, implementation, and experimental validation," *IEEE*

Transactions on Control Systems Technology, vol. 10, no. 6, pp. 835-852, November 2002.

[12] A. Izadbakhsh, M. M. Fateh, and M. A. Sadrnia, "Discontinuous feedback linearization of an electrically driven fast robot manipulator," *World Academy of Science, Engineering and Technology*, 29, pp. 217-222, 2007.

[13] R. Boukezzoula, S. Galichet, and L. Foulloy, "Fuzzy feedback linearizing controller and its equivalence with the fuzzy nonlinear internal model control structure," *Int. J. Appl. Math. Comput. Sci*, vol. 17, no. 2, pp. 233-248, 2007.

[14] I. Hassanzadeh, S. Mobayen, and A. Harifi, "Input-output feedback linearization cascade controller using genetic algorithm for rotary inverted pendulum system," *American Journal of Applied Sciences 5*, vol. 10, pp. 1322-1328, 2008.

[15] H. Bouadi, M. Bouchoucha, and M. Tadjine, "Sliding mode control based on backstepping approach for an UAV type-quadrotor," *International Journal of Applied Mathematics and Computer Sciences,* vol. 4, no. 1, pp. 12-17, 2007.

[16] M. Guisser, and H. Medromi, "A high gain observer and sliding mode controller for an autonomous quadrotor helicopter," *International Journal of Intelligent Control and Systems*, vol. 14, no. 3, pp. 204-212, September 2009.

[17] H. Bouadi and M. Tadjine, "Nonlinear observer design and sliding mode control of four rotors helicopter," *International Journal of Mathematical, Physical, and Engineering Sciences,* vol.1, no.2, pp. 115-120, 2007.

[18] A. Mokhtari, A. Benallegue, and A. Belaidi, "Polynomial linear quadratic Gaussian and sliding mode observer for a quadrotor unmanned aerial vehicle," *Journal of Robotics and Mechatronics*, vol. 17, no. 4, pp. 483-495, March 2008.

[19] A. Mokhtari, A. Benallegue, and B. Daachi, "Robust inner outer controller and sliding mode observer for a quadrotor UAV," *ICGST-ARAS Journal*, vol. 6, pp. 17-26, April, 2007.

[20] M. Efe, "Robust low altitude behaviour control of a quadrotor rotorcraft through sliding modes," in *Proceedings of the 15th Mediterranean Conference on Control & Automation*, pp. 1-6, July, 2007.

[21] A. R. Husain, M. N. Ahmad, and A. H. M. Yatim, "Chattering-free sliding mode control for an active magnetic bearing system," *World Academy of Science,*

Engineering and Technology, vol.39, pp.385-391, 2008.

[22] Y. Niu, and X. Wang, "Sliding mode control design for uncertain delay systems with partial actuator degradation," *International Journal of Systems Science*, vol. 40, no. 4, pp. 403-409, April 2009.

[23] Y. Xia, Z. Zhu, C. Li, H. Yang, and Q. Zhu, "Robust adaptive sliding mode control for uncertain discrete-time systems with time delay," *Journal of the Franklin Institute,* vol. 347, pp. 339-357, October 2009.

[24] T. Chatchanayuenyong, "Power quality improvement using a sliding mode control of a series active filter," *American Journal of Applied Sciences 5*, vol. 8, pp. 1029-1033, 2008.

[25] F. Fahimi, "Sliding mode formation control for under-actuated autonomous surface vehicles," in *Proceedings of the 2006 American Control Conference*, pp. 4255-4260, June 14-16 2006.

[26] S. Bouabdallah, and R. Siegwart, "Backstepping and sliding-mode techniques applied to an indoor micro quadrotor," in *Proceedings of the 2005 IEEE International Conference on Robotics and Automation*, pp. 2259-2264, April 2005.

[27] L. Pollini, and A. Metrangolo, "Simulation and robust backstepping control of a quadrotor aircraft," in *AIAA Modeling and Simulation Technologies Conference and Exhibit*, AIAA 2008-6363, August 18-21, 2008.

[28] T. Madani and A. Benallegue, "Backstepping control for a quadrotor helicopter," in *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 3255-3260, October 9-15, 2006.

[29] A. Das, F. Lewis, and K. Subbarao, "Backstepping approach for controlling a quadrotor using lagrange form dynamics," *J. of Intell Robot Systems*, 56, pp. 127-151, March 2009.

[30] A. A. Mian, and B. Wang, "Modeling and backstepping-based nonlinear control strategy for a 6 DOF quadrotor helicopter," *Chinese Journal of Aeronautics*, vol. 21, pp. 261-268, March 2008.

[31] E. Altug, J. P. Ostrowski, and C. J. Taylor, "Control of a quadrotor helicopter using dual camera visual feedback," *The International Journal of Robotics Research*, vol.24, no.5, pp. 329-341, May 2005.

[32] E. Altug, J. P. Ostrowski, and R. Mahony, "Control of a quadrotor helicopter using visual feedback," in *Proceedings of the 2002 IEEE International Conference on Robotics & Automation*, pp. 72-77, May 2002.

[33] A. Soumelidis, P. Gaspar, G. Regula, and B. Lantos, "*Control of an experimental mini quad-rotor UAV*," Technical Report, Computer and Automation Research Institute of the Hungarian Academy of Sciences, pp. 1-6, June, 2000.

[34] C. Li, W. Jing, and C. Gao, "Adaptive backstepping-based flight control system using integral filters," *Aerospace Science and Technology,* vol. 13, no. 2-3, pp. 105-113, 2009.

[35] E. Frazzoli, M. A. Dahleh, and E. Feron, "Trajectory tracking control design for autonomous helicopters using a backstepping algorithm," in *Proceedings of the American Control Conference*, pp. 4102-4107, June 2000.

[36] W. A. Pradana, E. Joelianto, A. Budiyono, and W. Adiprawita, "Robust MIMO H_∞ integral-backstepping PID controller for hovering control of unmanned model helicopter," *Journal of Aerospace Engineering*, pp. 1-29, August 2010.

[37] R. Saber, "*Nonlinear control of underactuated mechanical systems with application to robotics and aerospace vehicles*," PhD Thesis, Department of Electrical Engineering and Computer Science, MIT, pp. 1-307, February 2001.

[38] N. P. I Aneke, "*Control of underactuated mechanical systems*," Thesis, Technische Universiteit Eindhoven, pp. 1-176, April 2003.

[39] Y. M. Zhang, and J. Jiang, "Bibliographical review on reconfigurable fault-tolerant control systems," *Annual Reviews in Control*, vol. 32, pp. 229-252, March 2008

[40] R. J. Patton, "Fault-tolerant control systems," in *IFAC Symposium on Fault Detection Supervision and Safety for Technical Processes*, pp. 1033-1054, 1997

[41] Y. M. Zhang, "*Fault diagnosis and fault tolerant control systems*," Lecture Notes, pp. 20-33, September 2010.

[42] A. Fekih, and P. Pilla, "A passive fault tolerant control strategy for the uncertain MIMO aircraft model F-18," in *Thirty-Ninth Southeastern Symposium on System Theory,* pp. 320-325, 2007.

[43] Y. M. Zhang, and J. Jiang, "Active fault-tolerant control system against partial actuator failures," *IEE Proc-Control Theory & Applications*, vol. 149, no. 1, pp.

95-104, January 2002.

[44] A. B. Milhim, and Y. M. Zhang, "Gain scheduling based PID controller for fault tolerant control of a quad-rotor UAV," in *AIAA Infotech@Aerospace 2010*, AIAA 2010-3530, pp. 1-13, April 2010.

[45] X. B. Zhang, Y. M. Zhang, C.-Y. Su, and Y. Feng, "Fault tolerant control for quadrotor UAV via backstepping approach", in *Proc. of the 48th AIAA Aerospace Sciences Meeting*, AIAA. 2010-947, Jan. 2010.

[46] X. Zhang and Y. M. Zhang, "Fault tolerant control for quad-rotor UAV by employing adaptive backstepping approach", in *2010 AIAA Guidance, Navigation, and Control Conference*, AIAA 2010-8052, Aug. 2010

[47] H. Alwi, and C. Edwards, "Fault tolerant control of a civil aircraft using a sliding mode based scheme," in *Proceeding of the 44th IEEE Conference on Decision and Control and the European Control Conference*, pp. 1011-1016, December 2005.

[48] H. Alwi, and C. Edwards, "Fault tolerant control using sliding modes with on-line allocation," *Automatica,* vol 44, no. 7, pp. 1859-1866, July 2008

[49] J. K. Hedrick and A. Girard, "*Control of nonlinear dynamic Systems: theory and applications*," Chapter 8, pp.133-160, 2005.

[50] J.J. Slotine and W.Li, "*Applied nonlinear control*," Chapter 6-7, Prentice Hall, pp. 207-307, 1990.

[51] H. K. Khalil, "*Nonlinear systems*," Chapter 13, Prentice Hall, pp. 577-650, 1996.

[52] C. Manzie, and M. Good, "*Advanced control and automation, Sontags formula backstepping control*," Lecture Notes, University of Melbourne, Australia, 2005.

[53] O. harkegard, "Backstepping, from simple designs to take-off," In *Internal Seminar of Control & Communication Linkopings University,* pp. 1-11, January 27, 2005.

[54] Quanser, "*Quanser Qball-X4*," User Manual, pp. 1-27, 2010.

[55] J. J. Craig, "*Introduction to robotics: mechanics and control*," Chapter 2, Addison Wesley Longman, pp. 19-59, August 2004.

[56] J. D. Anderson, Jr, "*Fundamentals of aerodynamics*," Chapter 2, McGraw-Hill, pp. 95-181, 2007.

[57] B. L. Stevens, and F. Lewis, "*Aircraft control and simulation*," Chapter 1, John

Wiley & Sons, pp. 1-59, 2003.

[58] M. J. Stepaniak, "*A quadrotor sensor platform*," PhD Thesis, the Russ College of Engineering and Technology of Ohio University, pp. 57-69, November 2008.

[59] P. Pounds, R. Mahony, and P. Corke, "*Modelling and control of a quad-rotor robot*," *Australian Robotics & Automation Association Inc*, pp. 1-10, May 2006.

[60] G. M. Hoffmann, H. Huang, S. L. Waslander, and C. J. Tomlin, "Quadrotor helicopter flight dynamics and control: theory and experiment," in *AIAA Guidance, Navigation and Control Conference and Exhibit*, AIAA 2007-6461, pp. 1-20, August 20-23, 2007.

[61] M. J. Stepaniak, F. V. Graas, and M. U. Haag, "Design of an electric propulsion system for a quadrotor unmanned aerial vehicle," *Journal of Aircraft*, vol. 46, no. 3, pp. 1050-1058, June 2009.