

**NEURAL NETWORK ADAPTIVE
CONTROL OF UNMANNED VEHICLE
WITH SLUNG LOAD**

BY

GHUFRAN AHMED

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

SYSTEMS ENGINEERING

APRIL 2013

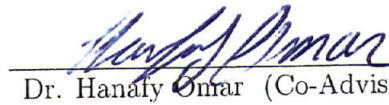
KING FAHD UNIVERSITY OF PETROLEUM & MINERALS
DHAHRAN 31261, SAUDI ARABIA

DEANSHIP OF GRADUATE STUDIES

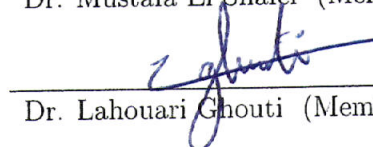
This thesis, written by **GHUFRAN AHMED** under the direction of his thesis adviser and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN SYSTEMS ENGINEERING**.

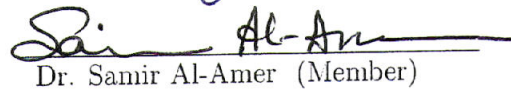
Thesis Committee



Dr. Sami El ferik (Advisor)

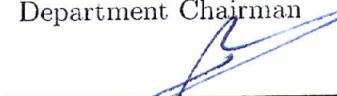

Dr. Hanafy Onrar (Co-Advisor)



Dr. Mustafa El Shafei (Member)


Dr. Lahouari Ghouti (Member)


Dr. Samir Al-Amer (Member)


Dr. Fouad M Al Sunni
Department Chairman


Dr. Salam A. Zummo
Dean of Graduate Studies


Date



©Ghufran Ahmed
2013

Dedicated to my beloved parents, brothers and sister

ACKNOWLEDGMENTS

All praise is to Almighty Allah and his beloved messenger Muhammad(SAWS).

I am grateful to King Fahd University of Petroleum and Minerals for providing a great environment for education and research.

I express my gratitude to my thesis advisor Dr. Sami El Ferik for his continuous support, patience and encouragement. He stood by me in all times and was the greatest support I had during my tenure in the university and most importantly during my thesis. I would also like to thank my co-advisor Dr. Hanafy Omar for

his support and cooperation in my thesis work. Also, I thank my thesis committee Dr. Moustafa El Shafei, Dr. Lahouari Ghouti and Dr. Samir Al Amer for their cooperation, time and valuable comments. I also extend my gratitude to Dr. Fouad Al Sunni, the Chairman of the Systems Engineering

Department for his support throughout my tenure at KFUPM.

I would like to acknowledge my parents for their everlasting love, trust and faith in me and for providing me the finest things I ever needed. I could never have pursued my higher education without their encouragement and support. My brothers who have always loved and supported me in all forms of life, their love gives me immense strength to keep moving ahead in all forms of life.

Lastly I would like to thank all my friends and colleagues back at home and at KFUPM whose presence and discussions were the biggest support during times of loneliness and despair. Things would not have been better if not for their continuous support.

TABLE OF CONTENTS

LIST OF TABLES	vii
LIST OF FIGURES	viii
LIST OF ABBREVIATIONS	xii
ABSTRACT (ENGLISH)	xiii
ABSTRACT (ARABIC)	xiv
CHAPTER 1 INTRODUCTION	1
1.1 Problem Formulation and Contribution	4
1.2 Thesis Organization	6
CHAPTER 2 LITERATURE REVIEW	7
2.1 UAV Modeling and Control	7
2.2 Slung Load Modeling and Control	11
2.3 Fault-Tolerant System	13
CHAPTER 3 MODELING OF THE UNMANNED VEHICLE WITH SLUNG LOAD	16
3.1 Model of the Unamnned Aerial Vehicle	16
3.2 Slung Load Model	25
3.3 Load Contribution to the Unmanned Aerial Vehicle	29
3.4 Overall Model of the Unamnned Aerial Vehicle with Slung Load .	30

CHAPTER 4 CONTROLLER DESIGN	32
4.1 Feedback Linearization Controller	33
4.1.1 Inner Loop Control	33
4.1.2 Outer Control Loop	38
4.2 Neural Network based Feedback Linearization	38
4.3 Time Delayed Feedback Control	47
4.4 Fault Tolerant Control	49
CHAPTER 5 RESULTS AND DISCUSSION	51
CHAPTER 6 CONCLUSIONS	91
6.1 Future Work	92
CHAPTER 7 APPENDIX	93
REFERENCES	95
VITAE	100

LIST OF TABLES

3.1	Parameters of the UAV	25
3.2	Parameters of the Slung Load	29

LIST OF FIGURES

3.1	UAV Quad-rotor.	17
3.2	UAV Quad-rotor with Slung Load.	26
3.3	Slung Load Model.	26
4.1	Two Loop Controller Structure	36
4.2	Two layer NN	40
4.3	NN based FLC	41
4.4	Flow Chart of Conversion of NN estimate to the Input	46
4.5	FLC with Anti-Swing Controller	48
4.6	NNFLC with Anti-Swing Controller	48
5.1	3D view of the tracking.	51
5.2	XY view of the tracking.	52
5.3	Angle ϕ	52
5.4	Angle θ	53
5.5	Angle ψ	53
5.6	Position x.	54
5.7	Position y.	54
5.8	Position z.	54
5.9	3D view of the tracking.	55
5.10	XY view of the tracking.	55
5.11	Angle ϕ	56
5.12	Angle θ	56
5.13	Angle ψ	56

5.14	Position x.	57
5.15	Position y.	57
5.16	Position z.	57
5.17	3D view of the tracking.	58
5.18	XY view of the tracking.	59
5.19	Angle ϕ	59
5.20	Angle θ	60
5.21	Angle ψ	60
5.22	Position x.	60
5.23	Position y.	61
5.24	Position z.	61
5.25	3D view of the tracking.	62
5.26	XY view of the tracking.	62
5.27	Angle ϕ	62
5.28	Angle θ	63
5.29	Angle ψ	63
5.30	Position x.	63
5.31	Position y.	64
5.32	Position z.	64
5.33	3D view of the tracking.	65
5.34	XY view of the tracking.	65
5.35	Angle ϕ	65
5.36	Angle θ	66
5.37	Angle ψ	66
5.38	Position x.	66
5.39	Position y.	67
5.40	Position z.	67
5.41	3D view of the tracking.	68
5.42	XY view of the tracking.	68
5.43	Load Swing Angle ϕ_L	69

5.44	Load Swing Angle θ_L .	69
5.45	Force F_x .	69
5.46	Force F_y .	70
5.47	Force F_z .	70
5.48	Angle ϕ .	71
5.49	Angle θ .	71
5.50	Angle ψ .	71
5.51	Position x .	72
5.52	Position y .	72
5.53	Position z .	73
5.54	3D view of the tracking.	73
5.55	XY view of the tracking.	74
5.56	Load Angle ϕ_L .	74
5.57	Load Angle θ_L .	75
5.58	Angle ϕ .	75
5.59	Angle θ .	75
5.60	Angle ψ .	76
5.61	Position x .	76
5.62	Position y .	76
5.63	Position z .	77
5.64	3D view of the tracking.	77
5.65	XY view of the tracking.	78
5.66	Load Angle ϕ_L .	78
5.67	Load Angle θ_L .	78
5.68	Angle ϕ .	79
5.69	Angle θ .	79
5.70	Angle ψ .	80
5.71	Position x .	80
5.72	Position y .	81
5.73	Position z .	81

5.74	Force F_x	82
5.75	Force F_y	82
5.76	Force F_z	82
5.77	XY view of the tracking.	83
5.78	3D view of the tracking.	83
5.79	Load Angle ϕ_L	83
5.80	Load Angle θ_L	84
5.81	Angle ψ	84
5.82	Position x	84
5.83	Position y	85
5.84	Position z	85
5.85	Position x	86
5.86	Position y	86
5.87	Force F_x	86
5.88	Force F_y	87
5.89	Load Angle ϕ_L	87
5.90	Load Angle θ_L	88
5.91	3D View of the trajectory	88
5.92	XY View of the trajectory	88
5.93	Damping Effect	89
5.94	3D View of the trajectory	90
5.95	XY View of the trajectory	90

LIST OF ABBREVIATIONS

ANN	Artificial Neural Network
DOF	Degree of Freedom
FLC	Feedback Linearization Controller
NN	Neural Network
NN-FLC	Neural Network based Feedback Linearization Controller
PD	Proportional Derivative
UAV	Unmanned Aerial Vehicle
VTOL	Vertical Take-Off and Landing

THESIS ABSTRACT

NAME: Ghufran Ahmed

TITLE OF STUDY: Neural Network Adaptive Control of Unmanned Vehicle
with Slung Load

MAJOR FIELD: Systems Engineering

DATE OF DEGREE: April 2013

Load transportation using Unmanned Aerial Vehicles (UAV) is very beneficial but research in this area has been not conducted extensively. The main challenge in this is that the model for the vehicle and the load together is not yet established and available. The UAV being an under-actuated system poses another challenge in the designing of the controller. The other main concern is the control of the load swing during transportation. All these make it a very interesting topic to be researched and developed. This work aims to present a model for the UAV with a load and then also presents a neural network based feedback linearization controller which helps in trajectory tracking of the UAV and a time-delayed feedback controller to stabilize the load swing angles, which work together making the overall system stable.

ملخص الرسالة

الأسم : غفران أحمد

عنوان الرسالة : تصميم منظومة تحكم تكيفي بأستخدام الشبكات العصبية وذلك للتحكم في استقرارية طيارة بدون طيار تحمل على متنها حمل متأرجح.

التخصص : هندسة نظم التحكم والقياس

تاريخ الدرجة العلمية: ابريل 2013

عمليات نقل اي حمولة باستخدام طائرات بدون طيار (روبوت) (UAV) مفيد جدا ولكن لا توجد هناك الكثير البحوث في هذا المجال على نطاق واسع. ان التحدي الرئيسي في تصميم نظام كهذا هو أن هذا النموذج للطيارة والحمل معا لم يتم اقتراحها من قبل وحتى الان فهذا النموذج لايزال غير متاح. اعتبار نظام هذا النوع من الطيارات من الانظام التي تعمل تحت تأثير الدفع فهو بدوره ايضا يضيف تحدي اخر للتصميم منظومة تحكم. وايضا تصميم نظام تحكم لحمل متأرجح يضيف تحدي كبير. كل ما ذكر سابقا يشكل جدير بأن ينال اهتمام الباحثين لعمل بحوث تطويرية في هذا المجال.

هذ البحث يركز على عمل نموذج رياضي لهذا النوع من الروبوتات مع اعتبار وجود حمل متأرجح خلال عملية النقل وتصميم منظومة تحكم و بناؤها باستخدام الشبكات العصبية والتغذية الراجعة العكسية، حيث يقوم نظام التحكم هذا بالمحافظة على استقرارية الطيارة وسيرها في نفس مسارها وضا يحتوي هذا النظام على متحكمة اخرى الهدف منها مراقبة استقرارية الحمل خلال عملية النقل. تعمل هاتان المتحكمتان معا لجعل النظام الكلي مستقرا.

CHAPTER 1

INTRODUCTION

Unmanned Aerial Vehicles (UAVs) are either operated remotely or are autonomous aircrafts. These vehicles are being utilized in many applications such as: security, management of natural risks of environment, intervention in hostile environments, management of ground installations, agriculture and military. Usage of UAVs makes it possible to gather information in dangerous environments without posing risk to flight crews and humans. For its use in the above mentioned applications, many different structures, shapes and sizes of vehicles were developed and utilized. The most important feature that the vehicle is supposed to have is Vertical Take-Off and Landing (VTOL). This VTOL capability makes it suitable for applications where space is a problem and indeed the missions and tasks now-a-days do not give the benefit of having it, so this makes it superior to other non-VTOL vehicles. This capability is present in only a few vehicles like modern apache helicopters, unmanned helicopters and quad-rotor, octorotors, tilt-rotor, co-axial rotor vehicles. The main disadvantage

of VTOL helicopters are the increased complexity and maintenance that comes with the intricate linkages, cyclic control of the main rotor blade pitch, collective control of the main blade pitch, and anti-torque control of the pitch of the tail rotor blades. Also, weighing the benefits and the drawbacks of all the vehicles, the choice of a UAV narrows down to the Quad-rotor.

The quad-rotor is considered an effective alternative to the high cost and complexity of standard rotorcraft. The quad-rotor UAV became popular in the last few decades, yet a lot of study on the quad-rotor UAV is being carried out since the beginning of the last century. Etienne Oehmichen first tried six designs for quad-rotor in the 1920s. Then, the second design had four rotors and eight propellers, all driven by a single engine. The rotorcraft exhibited a considerable degree of stability and controllability at that time. A thousand test flights were performed in the middle of 1920s and it was able to remain flying for several minutes in 1923. In 1922, George de Bothezat built a helicopter with four rotors, under the sponsorship of the United States army. The helicopter had four rotors mounted at the end of X-shape of 18 meters beams and weighed 1700 kg. About 100 flights were executed in 1923 and five meters was the highest altitude it ever reached. Another design for the quad-rotor was Convertawings Model A quad-rotor in 1956. Employing four rotors to create differential thrust, the craft is able to hover and move without the complex system of linkages and blade elements present on standard single rotor vehicles. From then on there have been

designs and experiments to reduce the size, cost, weight, inclusion of aerodynamic parts and many other factors. Now the size has significantly reduced and we can even see minute quad-rotors, varying from large to palm sizes. The quadrotor is classified as an underactuated system. This is due to the fact that only four actuators (rotors) are used to control all six degrees of freedom (DOF). The four actuators directly impact z-axis translation (altitude) and rotation about each of the three principal axes. The other two DOF are translation along the x and y-axis. These two remaining DOF are coupled, meaning they depend directly on the overall orientation of the vehicle (the other four DOF). Additive quadrotor benefits are swift maneuverability and increased payload.

The control strategy for the quad-rotor is one of the most important designs as it determines the operation and performance of the overall mission it is performing. The quad-rotor being an under-actuated vehicle makes the control design challenging and interesting. A lot of research has been done in the area of stabilization of quad-rotor angles and trajectory tracking. Different control methods have been developed and designed for stabilization and tracking. These have been illustrated in the literature review.

After selecting the UAV, the focus is now shifted upon load transportation. Helicopters are used for carrying loads in many applications even today. The load is slung to the helicopter and is transported. After considering the advantages of

quad-rotors over helicopters, a strong inspiration of transporting the load using a quad-rotor is developed, and this forms as a motivation for this thesis. So to deal with this the quad-rotor is to be modeled with the slung load. The main challenge will be that the load which is slung oscillates while the quad-rotor is in operation. This load swing can cause damage to the load by making the system unstable. It also limits the speed of the quad-rotor and at higher speeds it makes the quadrotor unstable. To deal with this a controller capable of stabilizing the quad-rotor and a controller for controlling the load angle have to be present. As the model is not available, it also has to be developed. The augmentation of the load model with the UAV has to be first done and then controllers have to be designed.

1.1 Problem Formulation and Contribution

After doing a literature review it was found out that there are a lot of open research areas in the UAV field. One of the most basic is to design a controller for the underactuated UAV. Being an underactuated system the controller design is a very challenging one. The normal controllers work in normal operating conditions, but in case of failures these do not perform well. So, fault-tolerant control is another area which needs to be worked upon by researchers. There are many types of faults that can occur, some of them are actuator faults, sensor faults and faults in propellers due to collisions or other damages. Fault detection and Identification (FDI) is also another area in which research has to be concentrated. Designing

switching controllers that switch depending on normal and fault conditions are to yet to be looked upon deeply as it is a very important area because faults are bound to occur at some point or the other in flight operations. Transporting load using UAV's is another good research field. Controllers for controlling the load swing angle while transportation of the load also need a lot of research to be done.

Moving from a single UAV to a group or fleet of UAV's, the open area of research is controlling a fleet of unmanned vehicles and also multiple vehicles carrying loads. The leader-follower problem for a fleet of UAV's and also UAV's carrying load can be looked into. Fault-tolerant systems while transportation of load using multiple UAVs are also open to research.

The model for the UAV with the slung load is not available, so it has to be developed and then control techniques have to be designed for it. To develop the overall model the UAV model has to be augmented with the load model and then controllers have to be designed. The aim of the work is to design a controller that will help in transferring the load using an UAV in the minimum possible time while keeping the load swing angle as small as possible and stabilize it to zero. So, the main contribution of this thesis is a model for the slung load UAV system and then a stable controller which allows for safe load transportation. The use of NN will make the controller model free and any UAV model can be used with this controller. So, the contribution of the work can be summarized as

- * Model of a UAV with a Slung Load.
- * Designing a tracking controller for an underactuated UAV using FLC.
- * Designing a tracking controller for the UAV using NN based FLC, where the NN estimates the dynamics of the UAV which are considered to be unknown.
- * Anti-swing controller to stabilize the load swings to zero.
- * A fault-tolerant controller for the UAV in case of a rotor fault.

1.2 Thesis Organization

The thesis is organized as follows. Chapter 2 deals with the literature review. In Chapter 3, the modeling of the UAV with the slung load is done, where in the UAV and slung load are modeled separately and then augmented together. In Chapter 4, the controller design is presented, in Chapter 5 the results are presented and discussed and Chapter 6 concludes the thesis work and presents a few future extensions.

CHAPTER 2

LITERATURE REVIEW

The literature review done is divided into sections for better understanding. The first section is dedicated to literature related to the modeling and control of quad-rotor UAV. In the second section, literature related to Slung Load model and control is presented and in the third section Fault-tolerant systems for UAV are presented.

2.1 UAV Modeling and Control

Castillo and Dzul et al. (2004) proposed a nested saturation based controller for the dynamic quadrotor. The Lagrangian model was used and the global stability of the system is proved. The proposed controller is also tested experimentally.

Tayebi and McGilvray (2006) proposed a quaternion based controller for attitude stabilization. The controller is a PD2 feedback controller and also is

effective in disturbance rejection. Also, a PD controller is proposed which is independent of the model and it is without the compensation for the coriolis and gyroscopic torque. This controller also provides global asymptotic stability.

Watanabe and Yoshihata, et al. (2007) proposed an image based servo controller in which a stationary camera obtains the images of the quadrotor and these are compared to the reference and a PID controller then stabilizes the quadrotor to the reference position. Image noises and model uncertainty are avoided to a great extent and thus making the controller effective.

Bouabdallah and Becker et al. (2007) proposed the autonomous quadrotor flight controllers for autonomous take-off and landing and tracking. Different control schemes were used. PID and LQR were designed but these showed poor performance in case of disturbances. So, backstepping and integral backstepping were designed to make the controller more robust. Also, an obstacle avoidance controller was also designed. These were tested and the global stability was proved.

Tayebi (2008) proposed a quaternion based dynamic output feedback control scheme without the need of velocity measurements for attitude tracking. The controller guarantees global asymptotical stability. An auxiliary unit quaternion having a similar structure as the unit quaternion is introduced and the input to

this is related to the vector part of the quaternion error under an appropriate unit quaternion based feedback.

Bourquardez and Mahony et al. (2009) proposed an image based visual servo controller for regulation of the quadrotor. A spherical image moment with an inertial goal vector is used to generate an image error. Different controllers were developed and their properties such as stability and transient behavior were given. The controller were also experimentally tested.

Das and Subbarao et al. (2009) proposed a controller based on dynamic inversion. Also the zero dynamics which do not allow good tracking are compensated with a robust term to provide better tracking. This makes the controller effective and also this robust term deals with external disturbances, however any state disturbances cause deterioration in performance of the controller.

Das and Lewis et al. (2009) proposed a backstepping based controller for quadrotor control. The backstepping control is not implemented on the state space form but on the Lagrangian form of the model. Neural networks are also implemented to estimate the aerodynamic effects and disturbances.

Pounds and Mahony et al. (2010) presented a model for large quadrotors. The rotor dynamics were included in the model to make it more effective. Also, rotor

damping and blade flapping were considered in this. A simple PID controller was proposed which was capable of stabilizing the quadrotor.

Raffo, G. V., M. G. Ortega, et al. (2010) proposed a integral model predictive based nonlinear H infinity controller for the quadrotor. The MPC controller is implemented for trajectory tracking and the H infinity controller is designed for attitude stabilization. The controller is effective against disturbances as it uses the integral action. It is also very robust to model uncertainties.

Dierks and Jagannathan (2010) proposed a controller based on Neural networks and output feedback. The dynamics which may not be known are handled by the learning neural network. An observer based on neural network was also designed to estimate the velocities which may not be measurable. Semiglobal uniformly ultimately bounded stability is guaranteed in presence of external disturbances while the separation principle was relaxed.

Nicol and Macnab, et al. (2011) proposed an adaptive control based on Cerebellar Model Arithmetic Computer (CMAC) which are nonlinear approximators and the weights of these are updated in case of increase in payloads. The adaptive parameters handle the robustness for disturbances. The controller was tested against LQR and Linear in parameter controller and was proved to better in performance.

Derafa and Benallegue et al. (2012) proposed a second order sliding mode controller based on the super twisting algorithm. This ensures better robustness and less chattering. The controller is effective in stabilizing the attitude angles and also tracking while dealing with disturbances.

2.2 Slung Load Modeling and Control

The most early analysis for the helicopter with slung loads was conducted by Wolkovitch and Johnston (1965). Lagrange equations were used to develop the single cable dynamic model. Briczinski and Karas (1971) conducted many simulations of a helicopter and external load in real time with a pilot in the loop. The rotor downwash effects and the aerodynamics of the load were included in the model.

Abzug (1970) extended the developed model to include the case of two tandem cables. Instead of using Lagrange equations, the model was developed using Newton-Euler equations of motion for small perturbations and it were separated into longitudinal and lateral sets. Also, the aerodynamic forces on the cables and the load were neglected.

Asseo and Whitbeck (1973) used linearized equations of motion of the

helicopter, winch, cable and load for variable suspension geometry and then used them in conjunction with modern control theory to design different control systems for many kinds of suspensions.

Cera and Farmer (1974) investigated the possibility of stabilizing external loads by attaching controllable fins to the cargo. In a simple linear model representing the yawing and the pendulous oscillations of the slung-load system, an assumption was that the helicopter motion was unaffected by the load. Raz et al. (1989) studied the use of an active aerodynamic load stabilization system for a helicopter slung-load system.

Omar and Nayfeh (2005) presented a time delayed feedback controller for stabilizing the load swing angle for gantry cranes. The time delayed feedback controller works in coordination with a PD controller for the gantry movement. The controller was experimentally tested as well. Also, the effect of friction was taken care of by estimating it and then compensating for it.

Omar (2009) proposed a anti-swing controller for a helicopter at hover condition. The anti-swing controller is based on time delayed feedback of the load swing angles. The anti-swing controller gives some displacements which are added to helicopter trajectory in lateral and longitudinal directions.

Palunko et al (2012) developed a model for a UAV with Slung Load. The authors here implement a feedback linearization controller which is made adaptive. To control the load swings, a dynamic programming algorithm was used. This dynamic programming generates a swing free trajectory. The change in center of gravity occurs in cases when the load is not suspended from the center of gravity. This is also taken into account in the controller presented.

2.3 Fault-Tolerant System

Berbra et al. (2008) proposed a multi observer switching strategy for fault tolerant control of a quadrotor. To estimate the attitude, a bank of observers is employed and each of the observers is sensitive to faults in all sensors but one. Based on a decided criterion the attitude estimation that is insensitive to the faulty sensor is selected.

Alves et al. (2009) proposed a hybrid configuration for control which uses a sliding mode robust controller and an artificial neural network as an adaptive controller. The advantage of this is that both the robustness and performance under uncertainties and faults is maintained. These two controllers are together implemented and the stability of this is proved by Lyapunov stability analysis. Faults are introduced in the actuator of the system and the fault is seen to be affecting the pitch angle of the quadrotor.

Izadi et al. (2010) proposed a data-driven fault tolerant model predictive control which does not require the post fault model to switch or reconfigure the control system, thereby saving critical post fault time. Due to the MPC characteristics it identifies the fault and calculates the control every step and so when a fault occurs it calculates it and saves time after a fault. Different cases are tested in simulation which includes partial failure of a rotor, complete failure of one rotor and simultaneous partial failures of two rotors. In case of a partial fault to one rotor, the altitude of the quadrotor is not much affected, but the yaw and pitch angles are affected which are controlled using FT-MPC. In case of a complete failure of a rotor it was reported that the yaw and pitch angle cannot be stabilized. In case of simultaneous faults as well the control strategy performed very well. Fault detection could be added to the MPC as well but it resulted in a slight decrease in performance.

Sharifi and Mirzaei, et al. (2010) proposed a fault detection and control technique for faults occurring in actuators of the quadrotor. The faults can be detected using a state estimator and in case of faultless operation disturbance rejection is achieved. Fault tolerant control is achieved by the reconfiguration of the sliding mode control whenever a fault is detected.

Freddi et al. (2011) proposed a double control loop architecture using

feedback linearization to control a quadrotor in case of complete failure of one rotor. The inner controller controls the attitude angles and the altitude and the task of the outer loop is to control the position in x and y directions by modifying the attitude angles. The quadrotor keeps spinning in the yaw direction and all other attitude angles and altitude positions are stabilized and tracked accordingly. Controllability of the yaw angle is forfeited and this is seen in the simulation results. The controller shows good tracking in all the desired outputs and the quadrotor can remain flying even when one of its rotors has failed.

Freddi et al. (2012) presented a nonlinear observer (Thau) for sensor fault detection in unmanned vehicles. Certain assumptions are made for the UAV model and these assumptions allow the observer to be implemented. These assumptions are very basic and almost all the UAVs satisfy it. The fault detection is of two components; a residual generation module and a residual evaluation module. Residual generation is being done by nonlinear model based approach, in which a Thau observer is used. The faults introduced are of incipient (ramp) type. The simulation results show good results and the lateral motion of the quadrotor is affected the most in case of roll sensor faults whereas it is least affected when faults occur on pitch sensor.

CHAPTER 3

MODELING OF THE UNMANNED VEHICLE WITH SLUNG LOAD

The UAV model and the Slung Load model can be developed separately and then augmented by adding the interactive forces and moments to make it as a single system overall. So, first the UAV model will be developed.

3.1 Model of the Unamnned Aerial Vehicle

The model of the UAV was developed using the model obtained from Freddi et al (2011), to which the gyroscopic torque is added. There are four motors M_1 , M_2 , M_3 and M_4 connected on the ends of a cross structure. The propellers are connected to the motors through gears. To keep the propellers rigid, fixed

pitch blades are used and thus the flexibility of the blades is negligible. The axes of rotation are fixed and parallel. The quadrotor body is rigid and also the design is assumed to be symmetrical. The free stream air velocity is assumed to be zero. Figure 3.1 depicts the structure of the quadrotor being modeled.

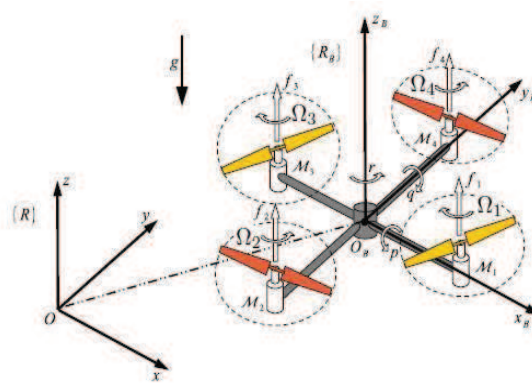


Figure 3.1: UAV Quad-rotor.

The front and rear propellers (P_1 and P_3) turn in the clockwise direction with angular velocities Ω_1 and Ω_3 and the left and right propellers (P_2 and P_4) turn in the anti-clockwise direction with angular velocities Ω_2 and Ω_4 . This relationship of opposite pairs eliminates the requirement of a tail rotor which is required in a standard helicopter. The quadrotor resembles the helicopter to a great extent and with slight modifications, the modeling and control techniques can be implemented to a helicopter as well. The thrust generated by each motor is always upwards which is the lifting force and the throttle input is given by the sum of the thrust of all rotors.

The roll movement in positive direction is attained by incrementing the speed of the left propeller and decrementing the speed of the right propeller and vice versa

for the negative direction of roll movement. This difference in the speeds produces a torque which turns the quadrotor. The pitch movement in positive direction is attained by incrementing the speed of the rear propeller and decrementing the speed of the front propeller and vice versa for the negative direction of pitch movement. The yaw movement in positive direction is attained by incrementing the speed of the front and rear propellers and decrementing the speed of the left and right propellers and vice versa for the negative direction. The yaw movement does not cause a change in the thrust forces so that the altitude changes so, even at a fixed altitude the yaw angle changes to any desired value.

To develop a model of the quadrotor, two frames are used. The first frame is the earth frame which is represented by $\{R\}(O, x, y, z)$ which is the inertial frame. The second frame is a body fixed frame $\{R_B\}(O_B, x_B, y_B, z_B)$ in which O_B is the origin and it represents the center of mass of the quadrotor.

The position of the quadrotor body frame with respect to the earth frame is denoted by the vector $\xi = [x \ y \ z]^T$ and the orientation in angular position of the body frame with respect to the earth frame is denoted by $\eta = [\phi \ \theta \ \psi]^T$ which represent the roll, pitch and yaw. The translational and rotational movement of the quadrotor with respect to the earth inertial frame can be described by using the combined vector of ξ and η given by

$$q = [\xi^T \ \eta^T]^T = [x \ y \ z \ \phi \ \theta \ \psi]^T \quad (3.1)$$

The forces that act on the quadrotor are analyzed and the quadrotor dynamics are derived by it. The thrust force, the weight force and the drag terms are all included to ensure better modeling and control. The drag terms are supposed to obey the Stoke's Law there-by the rotational drag is proportional to angular velocity and the translational drag is proportional to the linear velocity. The matrix \mathbf{I} and \mathbf{W}_η are given by

$$\mathbf{I} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}, \mathbf{W}_\eta = \begin{bmatrix} 1 & 0 & -\sin\theta \\ 0 & \cos\phi & \sin\phi\cos\theta \\ 0 & -\sin\phi & \cos\phi\cos\theta \end{bmatrix}$$

These are the inertia matrix and the transformation matrix. The matrix \mathbf{W}_η^{-1} will be

$$\mathbf{W}_\eta^{-1} = \begin{bmatrix} 1 & \sin\phi\tan\theta & \cos\phi\tan\theta \\ 0 & \cos\phi & -\sin\phi \\ 0 & \frac{\sin\phi}{\cos\theta} & \frac{\cos\phi}{\cos\theta} \end{bmatrix}$$

The matrix M_η is given by

$$\mathbf{M}_\eta = \mathbf{W}_\eta^T \mathbf{I} \mathbf{W}_\eta \quad (3.2)$$

$$\mathbf{M}_\eta = \begin{bmatrix} I_{xx} & 0 & -I_{xx}S\theta \\ 0 & I_{yy}C^2\phi + I_{zz}S^2\phi & (I_{yy} - I_{zz})C\phi S\phi C\theta \\ -I_{xx}S\theta & (I_{yy} - I_{zz})C\phi S\phi C\theta & I_{xx}S^2\theta + I_{yy}S^2\phi C^2\theta + I_{zz}C^2\phi C^2\theta \end{bmatrix}$$

where $S_{(\cdot)}$ and $C_{(\cdot)}$ symbolize $\sin(\cdot)$ and $\cos(\cdot)$ respectively. The Euler-Lagrangian equation for the generalized co-ordinate vector q is given by

$$\mathbf{L}(q, \dot{q}) = T_{trans} + T_{rot} - U = 1/2m\dot{\xi}^T\dot{\xi} + 1/2\dot{\eta}^T\mathbf{M}_\eta\dot{\eta} - mgz \quad (3.3)$$

This equation does not contain cross terms in the kinetic energy combining $\dot{\xi}$ and $\dot{\eta}$, therefore the Euler-Lagrangian equation with respect to the translational motion is given by

$$\frac{d}{dt} \frac{\partial \mathbf{L}}{\partial \dot{\xi}} - \frac{\partial \mathbf{L}}{\partial \xi} = \mathbf{F}_\xi \quad (3.4)$$

The Euler-Lagrangian with respect to the rotational motion is given by

$$\frac{d}{dt} \frac{\partial \mathbf{L}}{\partial \dot{\eta}} - \frac{\partial \mathbf{L}}{\partial \eta} = \tau_\eta \quad (3.5)$$

F_ξ and τ_η are the forces and torques respectively, acting on the body. The drag is represented by the matrices D_η and D_ξ where, k_t is the translational drag and k_r is the rotational drag.

$$D_\eta = k_r I = \begin{bmatrix} k_r & 0 & 0 \\ 0 & k_r & 0 \\ 0 & 0 & k_r \end{bmatrix} \quad (3.6)$$

$$D_\xi = k_t I = \begin{bmatrix} k_t & 0 & 0 \\ 0 & k_t & 0 \\ 0 & 0 & k_t \end{bmatrix} \quad (3.7)$$

We define a matrix for simplification in expression.

$$A_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} \quad (3.8)$$

Simplifying the Euler-Lagrangian equations for translational and rotational motion

$$\frac{d}{dt} \frac{\partial \mathbf{L}}{\partial \dot{\xi}} - \frac{\partial \mathbf{L}}{\partial \xi} = \mathbf{F}_\xi$$

$$\frac{d}{dt} \left(\frac{1}{2} m (2\dot{\xi}) \right) - (mg \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}) = R_{BI} u \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} - D_\xi \dot{\xi}$$

The translational dynamics are written as

$$M_\xi \ddot{\xi} + G_\xi + D_\xi \dot{\xi} = R_{BI} u A_1 \quad (3.9)$$

where

$$M_{\xi} = \begin{bmatrix} m & 0 & 0 \\ 0 & m & 0 \\ 0 & 0 & m \end{bmatrix}$$

$$G_{\xi} = \begin{bmatrix} 0 \\ 0 \\ mg \end{bmatrix}$$

For the rotational dynamics

$$\frac{d}{dt} \frac{\partial \mathbf{L}}{\partial \dot{\eta}} - \frac{\partial \mathbf{L}}{\partial \eta} = \tau_{\eta}$$

The above equation can be written in terms of $\ddot{\eta}$ and $\dot{\eta}$ as

$$M_{\eta} \ddot{\eta} + C_{\eta}(\eta, \dot{\eta}) \dot{\eta} + D_{\eta} \dot{\eta} = \tau_{\eta} \quad (3.10)$$

where M_{η} is given by

$$M_{\eta} = \begin{bmatrix} I_{xx} & 0 & -I_{xx} S \theta \\ 0 & I_{yy} C^2 \phi + I_{zz} S^2 \phi & (I_{yy} - I_{zz}) C \phi S \phi C \theta \\ -I_{xx} S \theta & (I_{yy} - I_{zz}) C \phi S \phi C \theta & I_{xx} S^2 \theta + I_{yy} S^2 \phi C^2 \theta + I_{zz} C^2 \phi C^2 \theta \end{bmatrix}$$

and $C_\eta(\eta, \dot{\eta})$ is given by

$$C_\eta(\eta, \dot{\eta}) = \begin{bmatrix} c_{11} & c_{12} & c_{13} \\ c_{21} & c_{22} & c_{23} \\ c_{31} & c_{32} & c_{33} \end{bmatrix}$$

The terms c_{ij} for $i, j = 1, 2, 3$ are given in the Appendix. The equations Eq 3.9 and Eq 3.10 represent the translational and rotational dynamics of the UAV and can also be combined or used separately. If they are combined the total dynamics of the UAV is expressed as

$$M\ddot{q} + C\dot{q} + D\dot{q} + G = \tau \quad (3.11)$$

The matrices M , C , D and G represent the dynamics of the UAV. The matrix M is called the inertia matrix. C is the matrix containing the centrifugal and coriolis terms. D represents the drag terms and G is the gravitational matrix.

The dynamics can also be expressed in the state space format as

$$\dot{\phi} = p + q \sin\phi \tan\theta + r \cos\phi \tan\theta \quad (3.12)$$

$$\dot{\theta} = q \cos\phi - r \sin\phi \quad (3.13)$$

$$\dot{\psi} = \frac{[q \sin\phi + r \cos\phi]}{\cos\theta} \quad (3.14)$$

$$\dot{p} = \frac{[-q r (I_{zz} - I_{yy}) - k_r p - I_R q \Omega + \tau_p]}{I_{xx}} \quad (3.15)$$

$$\dot{q} = \frac{[-p r (I_{xx} - I_{zz}) - k_r q + I_R p \Omega + \tau_q]}{I_{yy}} \quad (3.16)$$

$$\dot{r} = \frac{[-p q (I_{yy} - I_{xx}) - k_r r + \tau_r]}{I_{zz}} \quad (3.17)$$

$$\dot{x} = v_x \quad (3.18)$$

$$\dot{y} = v_y \quad (3.19)$$

$$\dot{z} = v_z \quad (3.20)$$

$$\dot{v}_x = \frac{[(\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi)u - k_t v_x]}{m} \quad (3.21)$$

$$\dot{v}_y = \frac{[(\cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi)u - k_t v_y]}{m} \quad (3.22)$$

$$\dot{v}_z = \frac{[(\cos\phi \cos\theta)u - k_t v_z - mg]}{m} \quad (3.23)$$

This developed model is used as a basis to design the controller. Here, p , q and r are the angular velocities. This form is usually termed as the control affine form. This is also a form to represent certain class of non linear systems. The parameters of the UAV model are selected as

UAV Model Parameters		
<i>Mass of the UAV</i>	m	2 Kg
<i>Length of the arm</i>	l	0.205 m
<i>Inertia coefficient in x</i>	I_{xx}	1.25 Kg m^2
<i>Inertia coefficient in y</i>	I_{yy}	1.25 Kg m^2
<i>Inertia coefficient in z</i>	I_{zz}	2.5 Kg m^2
<i>Rotor Inertia</i>	I_r	$3.36e^{-3} \text{ Kg m}^2$
<i>Translational Drag</i>	k_t	0.95
<i>Rotational Drag</i>	k_r	0.105
<i>Gravity</i>	g	9.81 m/s^2

Table 3.1: Parameters of the UAV

3.2 Slung Load Model

The load model is obtained from Omar (2012), which was also implemented in Masoud et al.(2002) and Omar and Nayfeh (2005) , in which the slung load was integrated with the helicopter. The external load is modeled as a point mass that behaves like a spherical pendulum suspended from a single point. The only aerodynamic force is a quasi-steady drag force in the direction of the local airflow; no aerodynamic forces act on the cable. The cable is assumed to be inelastic and with no mass.

The geometry and the relevant coordinate systems are shown in Figure 3.2 and Figure 3.3. The unit vectors i_H, j_H, k_H of the hook coordinate system always remain parallel to those of the body axis system. The position of the load is described by the two angles θ_L and ϕ_L , where ϕ_L is the load angle in the xz plane and θ_L is the load oscillation angle out of the xz plane. L is the length of the cable and m_L is the mass of the load.

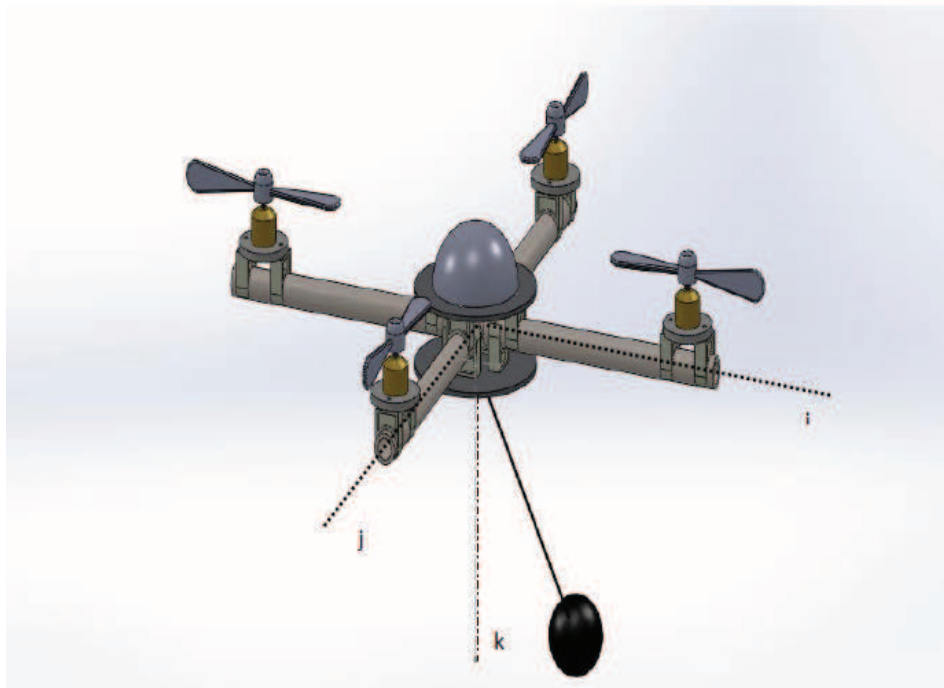


Figure 3.2: UAV Quad-rotor with Slung Load.

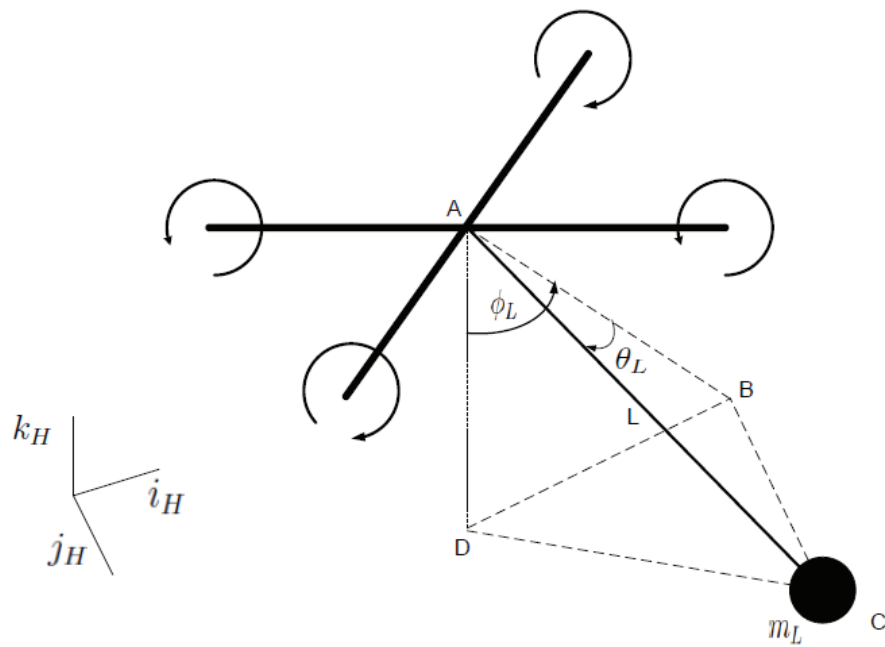


Figure 3.3: Slung Load Model.

In the Figure 3.3, the angle θ_L is represented by $\angle BAC$ in the triangle ΔABC and the angle ϕ_L is represented by $\angle BAD$ in the triangle ΔABD . AC is the length of the cable L . According to ΔBAD

$$\cos(\phi_L) = \frac{AD}{AB} \quad \sin(\phi_L) = \frac{DB}{AB}$$

According to ΔABC ,

$$\cos(\theta_L) = \frac{AB}{AC} \quad \sin(\theta_L) = \frac{BC}{AC}$$

The positions are given by AD , DB and BC and therefore, the position vector of the load with respect to the suspension point is R_L , which is given by

$$R_L = L\cos(\theta_L)\sin(\phi_L)i_H + L\sin(\theta_L)j_H + L\cos(\theta_L)\cos(\phi_L)k_H \quad (3.24)$$

The position vector of the hook with respect to the UAV center of gravity is R_H , which is given by

$$R_H = x_H i_H + y_H j_H + z_H k_H \quad (3.25)$$

The absolute velocity V_L of the load is given by

$$V_L = V_{cg} + \dot{R} + \Omega_L \times R \quad (3.26)$$

where V_{cg} is the absolute velocity of the center of mass of the UAV. $R = R_L + R_H$ is the position vector of the load with respect to the center of mass of the UAV and $\Omega_L = pi_H + qj_H + rk_H$ is the angular velocity of the UAV. The absolute acceleration a_L of the load is

$$a_L = \dot{V}_L + \Omega_L \times V_L \quad (3.27)$$

The unit vector in the direction of the gravity force is given by

$$K_g = -\sin(\theta)i_H + \sin(\phi)\cos(\theta)j_H + \cos(\phi)\cos(\theta)k_H \quad (3.28)$$

Besides the gravity, there is an aerodynamic force applied on the point mass load and it is assumed in the direction of the load velocity and is given by

$$D = \frac{1}{2}\rho S_L |V_L| V_L \quad (3.29)$$

where S_L is the equivalent flat plate area of the suspended load and ρ is the density of air.

The equations of motion of the load are written by enforcing moment equilibrium about the suspension point that is in matrix form

$$R_L \times (-m_L a_L + m_L g k_g + D) = 0 \quad (3.30)$$

The above equation gives three scalar equations of second order, only the equations in the x and y directions are retained which represent the equations of motion of the load. The parameters of the slung load model are given in the table below.

Slung Load Model Parameters		
<i>Mass of the Load</i>	m_L	1 Kg
<i>Length of the cable</i>	L	2 m
<i>Density of air</i>	ρ	1.2 Kg/m ⁻³
<i>Flat Plate area</i>	SL	0.8 m ²

Table 3.2: Parameters of the Slung Load

3.3 Load Contribution to the Unmanned Aerial Vehicle

The suspended load introduces additional terms in the rigid body force and moment equations of motion of the UAV. These terms are of inertia and aerodynamic origin. The force that the load exerts on the UAV is given by

$$F_{VL} = -m_L a_L + D + m_L g K_g \quad (3.31)$$

The additional moment M_{VL} is given by

$$M_{VL} = R_H \times F_{VL} \quad (3.32)$$

Both F_{VL} and M_{VL} are functions of load and UAV states and are

$$F_{VL} = \begin{bmatrix} F_x & F_y & F_z \end{bmatrix}^T \quad (3.33)$$

$$M_{VL} = \begin{bmatrix} M_x & M_y & M_z \end{bmatrix}^T \quad (3.34)$$

These forces are additive forces which act on the UAV. The forces act in the x , y and z directions and the moments act on ϕ , θ and ψ .

3.4 Overall Model of the Unmanned Aerial Vehicle with Slung Load

The load model and the UAV model have to be coupled together and then used for control purposes. It is as follows

$$M\ddot{q} + C\dot{q} + D\dot{q} + G = \tau + \tau_L \quad (3.35)$$

where τ_L is the force exerted by the load on the UAV and is given by

$$\tau_L = \begin{bmatrix} F_{VL} \\ M_{VL} \end{bmatrix} \quad (3.36)$$

As this being a Lagrangian system, there are a few basic properties which it satisfies. These are

The inertia matrix M is symmetric, positive definite and bounded.

The coriolis vector $C\dot{q}$ is quadratic in \dot{q} and the matrix C is bounded.

The matrix $\dot{M} - 2C$ is skew symmetric, such that for any vector x the term $x^T(\dot{M} - 2C)x = 0$. This is a statement of the fact that the fictitious forces do no work.

The gravity force is also bounded.

CHAPTER 4

CONTROLLER DESIGN

The non-linear model has to be controlled using a non-linear controller or linearized and then a linear controller be implemented. Linearization using Jacobian Linearization is not beneficial as the controller developed using that linearized model will be stable only at a particular operating point and will perform poorly at other operating points. Considering this Jacobian linearization is not done rather a non-linear controller is preferred. There are many non-linear controllers available but Feedback Linearization Controller (FLC) is selected because of the primary reason that it is not complicated and reduces to simple PD or PID control which can be implemented practically. FLC also makes the system linear which is much better when compared to Jacobian. Other controllers like Sliding mode and backstepping et cannot be reduced to the PD or PID, thus making it difficult to implement them practically. The details of basic feedback linearization can be found in H. Khalil (2002) and Slotine and Li (1991).

4.1 Feedback Linearization Controller

The main aim of the control structure designed here is to perform vertical take off and hover and then land. To achieve hover condition the attitude angles and the position have to be stabilized. The outputs will be $(x, y, z, \phi, \theta, \psi)$. However selecting all these six outputs will make the system under-actuated as there are only four inputs. There is also coupling problem here, therefore, to overcome this inconvenience the output vector is selected as (ϕ, θ, ψ, z) . In addition, the drag and gyroscopic terms cannot be neglected and are all considered in order to achieve complete stability. So taking into account all these terms a new control technique is designed. The control technique involves double loop architecture where-in the inner loop is responsible to control the attitude angles and the altitude and the outer loop is responsible for providing the inner loop with the desired angle values. Therefore, the inner loop has to be much faster than the outer loop in order to achieve stability. These four states are selected as outputs and feedback linearization technique is used.

4.1.1 Inner Loop Control

For the inner control loop consider a subsystem of the state variables $(\phi, \theta, \psi, p, q, r, z, \dot{z})$. The dynamics of these can be expressed as

$$\dot{\bar{x}} = f(\bar{x}) + h(\bar{x})\mathbf{u}_h \quad (4.1)$$

The dynamics of the selected outputs are

$$\dot{x}_a = \begin{bmatrix} \dot{\phi} \\ \dot{\theta} \\ \dot{\psi} \\ \dot{z} \end{bmatrix} = \begin{bmatrix} p + q \sin\phi \tan\theta + r \cos\phi \tan\theta \\ q \cos\phi - r \sin\phi \\ \frac{1}{\cos\theta} [q \sin\phi + r \cos\phi] \\ v_z \end{bmatrix} \quad (4.2)$$

It can be seen that the dynamic of the selected state variables is independent of \mathbf{u}_h and this is very helpful in computing the second derivative. Denoting this as

$$\hat{f}(\bar{x}) = \begin{bmatrix} p + q \sin\phi \tan\theta + r \cos\phi \tan\theta \\ q \cos\phi - r \sin\phi \\ \frac{1}{\cos\theta} [q \sin\phi + r \cos\phi] \\ \dot{z} \end{bmatrix} \quad (4.3)$$

Computing the second derivative of the outputs

$$\ddot{x}_a = \frac{d\hat{f}(\bar{x})}{dt} = \frac{\partial \hat{f}(\bar{x})}{\partial \bar{x}} \dot{\bar{x}} = \frac{\partial \hat{f}(\bar{x})}{\partial \bar{x}} f(\bar{x}) + \frac{\partial \hat{f}(\bar{x})}{\partial \bar{x}} h(\bar{x}) \mathbf{u}_h \quad (4.4)$$

Now the Jacobian matrix is defined as

$$J(\bar{x}) = \frac{\partial \hat{f}(\bar{x})}{\partial \bar{x}} \quad (4.5)$$

Substituting in the previous equation gives

$$\ddot{x}_a = J(\bar{x})f(\bar{x}) + J(\bar{x})h(\bar{x})\mathbf{u}_h \quad (4.6)$$

Let ϕ_d , θ_d , ψ_d and z_d are the desired outputs and denoted by x_{ad} . Then the error is defined as

$$e_\phi = \phi - \phi_d \quad (4.7)$$

$$e_\theta = \theta - \theta_d \quad (4.8)$$

$$e_\psi = \psi - \psi_d \quad (4.9)$$

$$e_z = z - z_d \quad (4.10)$$

The control inputs can be written as

$$\begin{bmatrix} \tau_p^* \\ \tau_q^* \\ \tau_r^* \\ u^* \end{bmatrix} = -(J(\bar{x})h(\bar{x}))^{-1}J(\bar{x})f(\bar{x}) - (J(\bar{x})h(\bar{x}))^{-1} \begin{bmatrix} Kd_i\dot{e}_{x_a} + Kp_i e_{x_a} - \ddot{x}_{ad} \end{bmatrix} \quad (4.11)$$

where Kp_i and Kd_i are positive, so that the error dynamics can be written as a stable second order dynamics. The desired values ψ_d and z_d are selected as per the choice of any specific mission. But the desired values ϕ_d and θ_d are given to the inner control loop by an outer loop so that the position x and y are also stabilized.

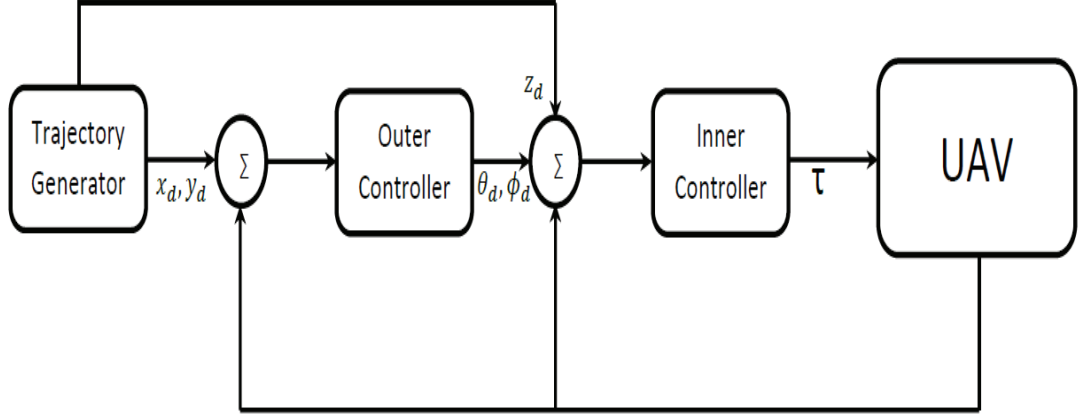


Figure 4.1: Two Loop Controller Structure

To estimate the gyroscopic torque term the propeller speeds are to be estimated. They are given by this relation of the inputs with the propeller speeds.

$$\begin{bmatrix} \tau_p \\ \tau_q \\ \tau_r \\ u \end{bmatrix} = \begin{bmatrix} 0 & -l & 0 & l \\ l & 0 & -l & 0 \\ d & -d & d & -d \\ 1 & 1 & 1 & 1 \end{bmatrix} \begin{bmatrix} f_1^2 \\ f_2^2 \\ f_3^2 \\ f_4^2 \end{bmatrix} \quad (4.12)$$

The propeller speeds are then estimated by inverting the matrix and taking a square root. After that Ω is estimated as

$$\Omega = \Omega_1 - \Omega_2 + \Omega_3 - \Omega_4 \quad (4.13)$$

This helps in estimating the gyroscopic torque and thus, the controller stabilizes the quadrotor with these torques acting on it.

Another way of writing this is that the dynamic equations of selected outputs x_a be written as

$$M_{x_a}\ddot{x}_a + C_{x_a}\dot{x}_a + D_{x_a}\dot{x}_a + G = B_{x_a}\tau_{x_a} \quad (4.14)$$

where

$$B_{x_a} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & \cos(\phi)\cos(\theta) \end{bmatrix} \quad (4.15)$$

pre-multiplying both sides by the inverse of the input matrix B_{x_a} the equation now becomes

$$M_b\ddot{x}_a + C_b\dot{x}_a + D_b\dot{x}_a + G_b = \tau_{x_a} \quad (4.16)$$

where $M_b = B_{x_a}^{-1}M_{x_a}$, $C_b = B_{x_a}^{-1}C_{x_a}$, $D_b = B_{x_a}^{-1}D_{x_a}$ and $G_b = B_{x_a}^{-1}G$. Now, the filtered tracking error is defined as

$$\gamma = \dot{e}_{x_a} + \Lambda e_{x_a} \quad (4.17)$$

where Λ is a positive definite constant matrix. Then the FLC is given by

$$\tau_{x_a} = K_v\gamma + M_b(\ddot{x}_a + \Lambda\dot{e}_{x_a}) + C_b(\dot{x}_a + \Lambda e_{x_a}) + D_b\dot{x}_a + G_b \quad (4.18)$$

where K_v is a positive diagonal design matrix. As long as the filtered error γ is bounded, the error remains bounded.

4.1.2 Outer Control Loop

The motion of the quadrotor in the horizontal direction is due to the horizontal components of the thrust forces. The roll and pitch angles are important for the horizontal components of the thrust forces and therefore, to reach a desired position for x and y , desired values of ϕ_d and θ_d are to be generated by the outer loop. Also, the inner loop has to be much faster than the outer loop for the overall control structure to be stable. Considering a subsystem of the states (x, y, \dot{x}, \dot{y}) and let ϕ_d and θ_d be small angles, then it can be derived as

$$\begin{bmatrix} \phi_d \\ \theta_d \end{bmatrix} = -\frac{m}{u} \begin{bmatrix} S_\psi & -C_\psi \\ C_\psi & S_\psi \end{bmatrix} \times \begin{bmatrix} -\frac{k_t}{m}\dot{x} + Kd_o e_{\dot{x}} + Kp_o e_x - \ddot{x}_d \\ -\frac{k_t}{m}\dot{y} + Kd_o e_{\dot{y}} + Kp_o e_y - \ddot{y}_d \end{bmatrix} \quad (4.19)$$

where Kd_o and Kp_o are both positive constants and give a stable second order error dynamics for the horizontal displacement. These desired values of ϕ_d and θ_d are the desired inputs to the inner controller. Therefore the inner controller stabilizes the quadrotor at a desired altitude and yaw angle and then the outer loop stabilizes it to a desired location in the x and y directions.

4.2 Neural Network based Feedback Linearization

Artificial Intelligence is digging its roots deeper into the theories of control. In open-loop applications it is being used for classification and pattern recognition

and in the closed-loop applications the function approximation property is the most important feature of this. A two layer NN is selected for this purpose in this work. The advantage of this two layered NN is that it does not require any preselection of a basis set. Also, the restriction of linearity in parameters is overcome by using this. The first layer weights allow the NN to train its own basis set for the system non-linearities. These weights are tuned by different algorithms, but the algorithm selected in this work is backpropagation with modified tuning algorithm. No preliminary offline training is required in this type of NN and also, the problem of net weight initialization is not a matter of concern in this approach. The initial weight updates are selected as zero. While the weights are trained on-line in real time, the PD tracking loop carries out its tracking keeping the error small. When the NN gets trained the tracking error reduces. The modification in the tuning algorithm is done to improve robustness to disturbances and estimation errors.

The NN is given in Figure 4.2 (F.Lewis,1999) where $\sigma(\cdot)$ is the activation function and it can be selected as any one of hard limit, linear threshold, hyperbolic tangent, radial basis function etc. and V , W are the weights of the respective layers. The activation function is selected depending upon different applications. The purpose of the activation function is to model the behaviour of the cell where there is no output below a certain value of the argument of $\sigma(\cdot)$ and the output takes a specified magnitude above that value. A general class of monotonously, non-decreasing functions taking on bounded values at $+\infty$ and $-\infty$ is known as

the sigmoid function. The sigmoid function here is

$$\sigma(x) = \frac{1}{1 + e^{-x}}$$

It is possible to construct NN with multiple hidden layers but, the computational load increases with the number of hidden layers and the number of neurons. A simple NN with two layers can form the most the complex decisions for classifications. Thus, a two layer NN is sufficient to perform all the estimations in many applications. The use of additional layers will only complicate and increase the computational load, which is undesired. So, a two layer NN is selected to increase the speed of computation.

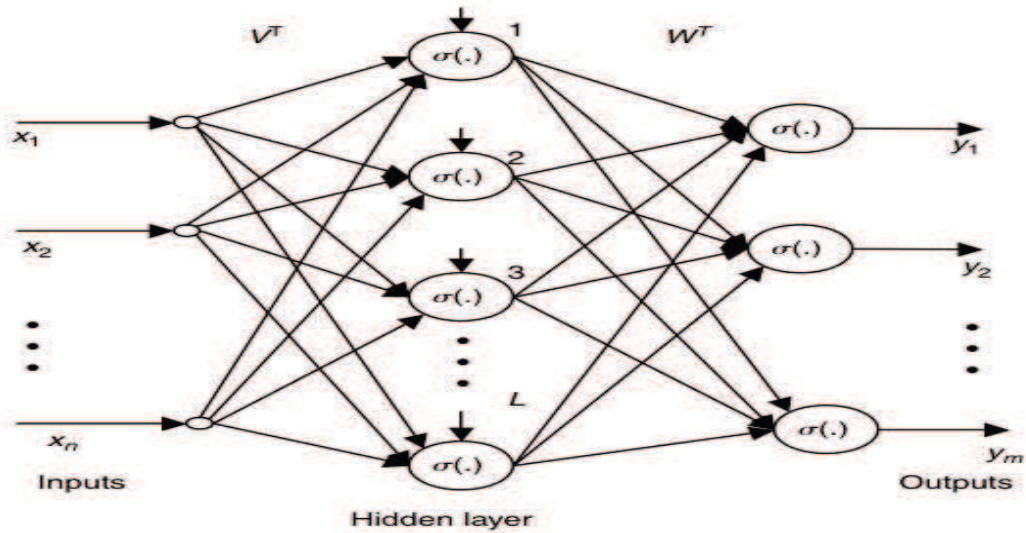


Figure 4.2: Two layer NN

The control structure with the NN is given in Figure 4.3

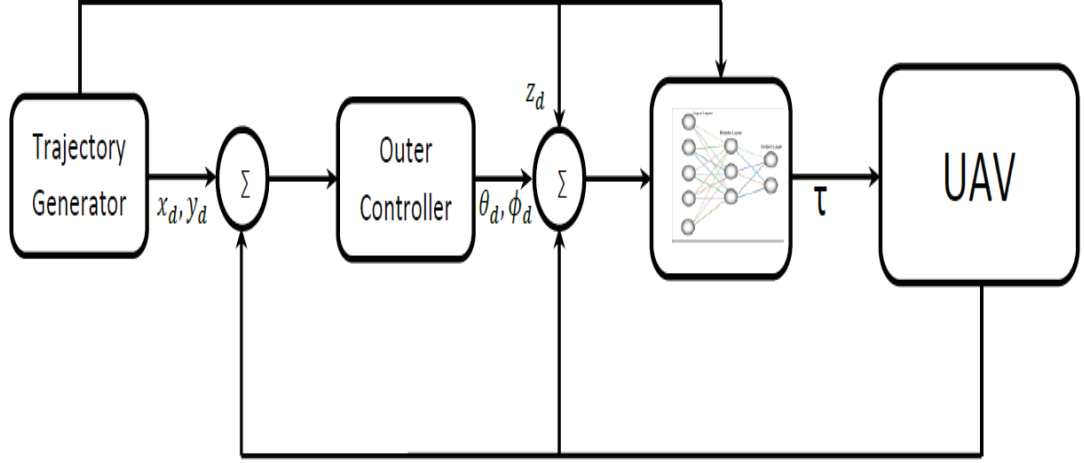


Figure 4.3: NN based FLC

The UAV inner loop can be described by this equation.

$$M_b \ddot{x}_a + C_b \dot{x}_a + D_b \dot{x}_a + G_b = \tau \quad (4.20)$$

Now, the tracking error and the filtered tracking error are defined as

$$e_{x_a} = x_{a_d} - x_a \gamma = \dot{e}_{x_a} + \Lambda e_{x_a}$$

Then the dynamics in terms of filtered error are expressed as

$$M_b \dot{\gamma} = -C_b \gamma + f(x) - \tau \quad (4.21)$$

where the unknown nonlinear UAV dynamics are defined as

$$f(x) = M_b(\ddot{x}_{a_d} + \Lambda \dot{e}_{x_a}) + C_b(\dot{x}_{a_d} + \Lambda e_{x_a}) + D_b \dot{x}_a + G_b \quad (4.22)$$

The vector x may here in this section be defined as

$$x = \left[e_{x_a}^T \quad \dot{e}_{x_a}^T \quad x_{a_d}^T \quad \dot{x}_{a_d}^T \quad \ddot{x}_{a_d}^T \right]^T \quad (4.23)$$

Now, according to the universal approximation property of NN, there is a two layer NN such that,

$$f(x) = W^T \sigma(V^T x) + \epsilon \quad (4.24)$$

with the approximation error bounded on a compact set by

$$\|\epsilon\| < \epsilon_N \quad (4.25)$$

with the ϵ a known bound. W and V are the ideal weights which provide good approximation to $f(x)$ and they are unknown. All that is needed is that they exist and there is also no restriction that they have to be unique. Defining the matrix with the neural network weights as

$$Z = \begin{bmatrix} W & 0 \\ 0 & V \end{bmatrix} \quad (4.26)$$

The number of hidden layer neurons is selected by running trials till good per-

formance is achieved. In the simulation the number of neurons were selected as 10.

Assumption 4.1 *On any compact subset of \mathfrak{R}^n the ideal NN weights are bounded so that*

$$\|Z\| \leq Z_B \quad (4.27)$$

where Z_B is known. Now, let a NN estimate of $f(x)$ be given by

$$\hat{f}(x) = \hat{W}^T \sigma(\hat{V}^T x) \quad (4.28)$$

with \hat{W} and \hat{V} the actual values of the NN weights given by the tuning algorithm. These are estimates of the ideal weights and the weight deviation or weight estimation error is defined as

$$\begin{aligned} \tilde{V} &= V - \hat{V} \\ \tilde{W} &= W - \hat{W} \\ \tilde{Z} &= Z - \hat{Z} \end{aligned} \quad (4.29)$$

The linearity in parameters restriction is overcome by providing tuning algorithms which appear in a non-linear fashion. The hidden layer output error for a given x is given by

$$\tilde{\sigma} = \sigma - \hat{\sigma} = \sigma(V^T x) - \sigma(\hat{V}^T x) \quad (4.30)$$

The Taylor series expansion of $\sigma(x)$ for a given x may be written as

$$\sigma(V^T x) = \sigma(\hat{V}^T x) + \acute{\sigma}(\hat{V}^T x)\tilde{V}^T x + O(\tilde{V}^T x)^2 \quad (4.31)$$

with

$$\acute{\sigma}(\hat{z}) = \left. \frac{d\sigma(z)}{dz} \right|_{z=\hat{z}} \quad (4.32)$$

$$\tilde{\sigma} = \acute{\sigma}\tilde{V}^T x + O(\tilde{V}^T x)^2 \quad (4.33)$$

where $O(z)^2$ denotes terms of second order. This equation is very important as the non-linear term in \tilde{V} is replaced by a linear term and higher order terms. This allows to derive the tuning algorithm for \tilde{V} . The control input now is selected as

$$\tau = \hat{W}^T \sigma(\hat{V}^T x) + K_v \gamma \quad (4.34)$$

K_v is the feedforward gain which is selected by the designer. The NN weight tuning algorithms are given by (F.Lewis,1999)

$$\dot{\hat{W}} = F \hat{\sigma} \gamma^T \quad (4.35)$$

$$\dot{\hat{V}} = G x (\acute{\sigma}^T \hat{W} \gamma)^T \quad (4.36)$$

The design parameters F,G are positive definite matrices.

Theorem 4.1 (Backpropagation Weight Tuning) *Let the desired trajectory*

be bounded by a known bound. The NN weights are also bounded by a known assumption and with the weight tuning algorithms provided in the above equations with any constant matrices $F = F^T > 0$, $G = G^T > 0$. Then the filtered error $\gamma(t)$ and the NN weight estimates \hat{V}, \hat{W} are uniformly ultimately bounded. Moreover the tracking error may be kept very small by proper selection of K_v .

Proof. Let the NN approximation property hold for the function $f(x)$ with a given accuracy ϵ_N for all x in the compact set $S_x \equiv x \mid \|x\| < b_x$ with $b_x > x_d$ and x_d be the bound on the desired trajectory. Defining the Lyapunov function for the stability proof as

$$L(\gamma, \tilde{W}, \tilde{V}) = \frac{1}{2}\gamma^T M_x \gamma + \frac{1}{2}\text{tr}\{\tilde{W}^T F^{-1} \tilde{W}\} + \frac{1}{2}\text{tr}\{\tilde{V}^T G^{-1} \tilde{V}\} \quad (4.37)$$

Differentiating it gives the following, without any abuse of notations, for simplicity

$$\dot{L} = \gamma^T M \dot{\gamma} + \frac{1}{2}\gamma^T \dot{M} \gamma + \text{tr}\{\tilde{W}^T F^{-1} \dot{\tilde{W}}\} + \text{tr}\{\tilde{V}^T G^{-1} \dot{\tilde{V}}\} \quad (4.38)$$

using the closed loop error system as

$$M \dot{\gamma} = -(K_v + C)\gamma + \tilde{W}^T \hat{\sigma} + \hat{W}^T \hat{\sigma} \tilde{V}^T x \quad (4.39)$$

then the equation now changes to

$$\dot{L} = -\gamma^T K_v \gamma + \gamma^T (\dot{M} - 2C) \gamma + \text{tr}\{\tilde{W}^T (F^{-1} \dot{\tilde{W}} + \hat{\sigma} \gamma^T)\} + \text{tr}\{\tilde{V}^T (G^{-1} \dot{\tilde{V}} + x \gamma^T \hat{W}^T \hat{\sigma})\} \quad (4.40)$$

Since, $\dot{M} - 2C$ is skew symmetric, the term vanishes and also substituting the weight update equations, it simplifies to

$$\dot{L} = -\gamma^T K_v \gamma \quad (4.41)$$

Thus, the Lyapunov function is either always zero or negative, as K_v is always selected as positive definite. (F.Lewis,1999) **I**

A flow chart is presented, which explains the conversion of the NN estimates to the actual angular velocity of the rotor.

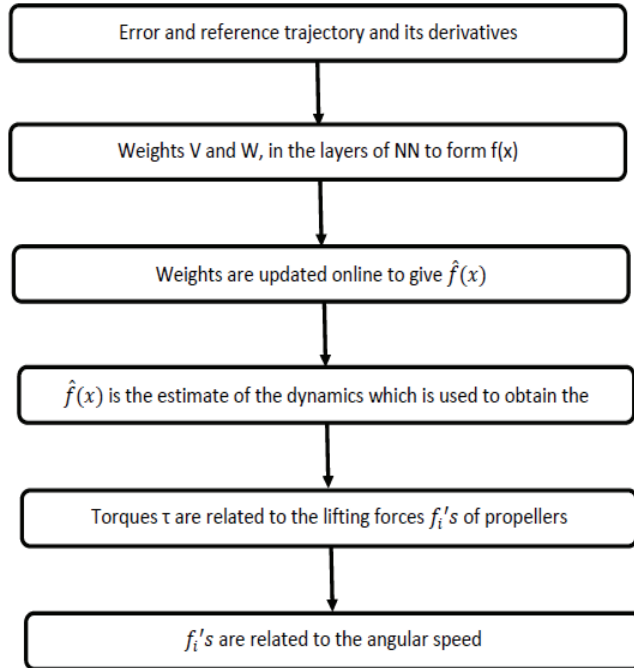


Figure 4.4: Flow Chart of Conversion of NN estimate to the Input

4.3 Time Delayed Feedback Control

Time-delayed feedback control or time-delay autosynchronization constructs a control value from the difference of the present state of a given system to its delayed value, i.e., $s(t) - s(t - \tau_{cor})$. By proper selection of the time delay τ_{cor} , the control value vanishes if the state to be stabilized converges. Thus, this method is noninvasive. This feedback scheme is easy to implement in an experimental setup and numerical calculation. It is capable of stabilizing fixed points as well as periodic orbits even if the dynamics are very fast. Also from a mathematical perspective it is an appealing instrument as the corresponding equations fall in the class of delay differential equations.

Very minimum knowledge of the system is required. The only quantity of the system that needs to be known is the period T of the periodic orbit which determines the choice of the time delay. Instead of time-delayed feedback, it is tempting to use proportional feedback, where the control is given by the difference of the current state to the desired state, which turns out to be a complicated process and is numerically exhaustive. Time-delayed feedback has been successfully employed in the context of chaos control.

The time-delayed feedback gives a correction trajectory which adds to the reference trajectory. This correction trajectory is given by the equation

$$x_{cor} = K_{xc} L \phi_L(t - \tau_{xc}) \quad (4.42)$$

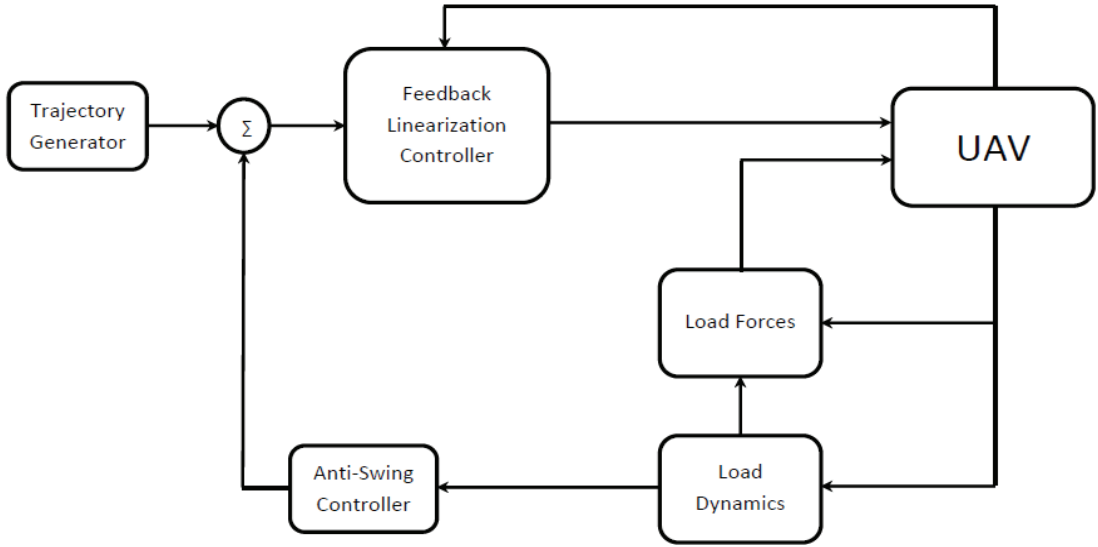


Figure 4.5: FLC with Anti-Swing Controller

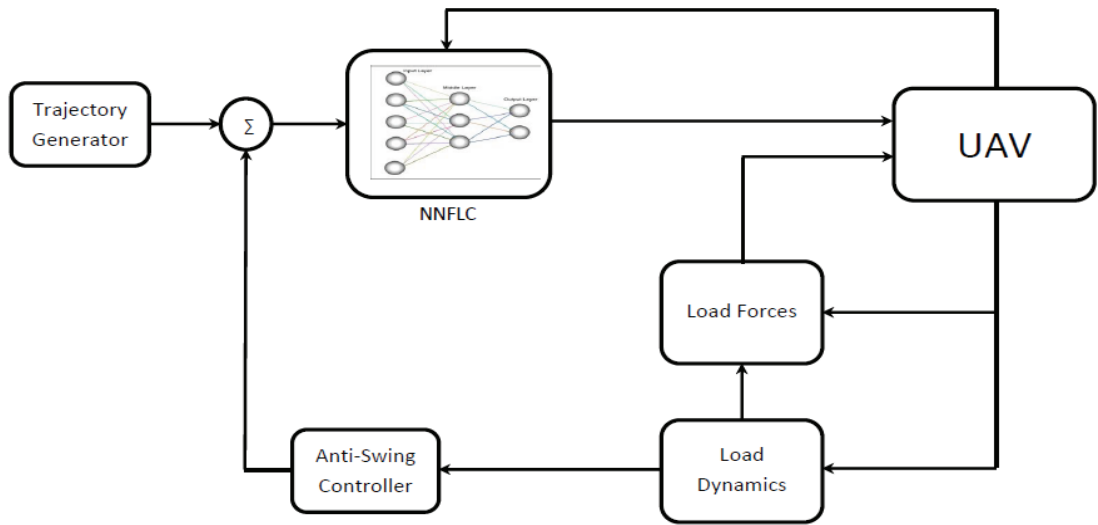


Figure 4.6: NNFLC with Anti-Swing Controller

$$y_{cor} = K_{yc}L\theta_L(t - \tau_{yc}) \quad (4.43)$$

These two trajectories act as anti-trajectories to the load angles and these are added to the reference tracking trajectories of the UAV. This helps in stabilizing the load angles. The parameters are K_{xc} , K_{yc} , τ_{xc} and τ_{yc} . The feedback gain and the time delay are selected accordingly to stabilize the angles as fast as pos-

sible. The optimal values of these can however be selected by using optimization algorithms, which may be included in future work. The values of the gains are selected as $K_{xc}=K_{yc}=0.4 * L$ and the time delay is selected as $\tau_{xc}=\tau_{yc}=0.32 * T_L$, and T_L is given by $T_L=\frac{2\pi}{\sqrt{\frac{g}{L}}}$ and it is the period of oscillation of the suspended load.

4.4 Fault Tolerant Control

A case of failure of one rotor is considered. An assumption is made that a Fault Detection and Isolation (FDI) module is present, which detects and isolates the faults whenever they occur. The details of this FDI module are out of the scope of this work and not illustrated in this work. Now, suppose rotor 2 fails and there is no lifting force provided by it, then the quadrotor is left with only three rotors to provide the thrust and movement. The rotor on the same axis which is rotor 4 is controlled till the pitch angle becomes zero. This leaves the quadrotor with only two rotors spinning in one direction and thus making the quadrotor rotate about its vertical axis or in the yaw angle. This is acceptable because the other attitude angles and the altitude are stabilized. The outputs will be (x, y, z, ϕ, θ) . The angle ψ is the yaw angle and it is compromised in the control strategy so it will not be considered as the output. The control technique involves double loop architecture where-in the inner loop is responsible to control the attitude angles ϕ and θ and the altitude z and the outer loop is responsible for providing the inner loop with the desired angle

values ϕ_d and θ_d . Therefore, the inner loop has to much faster than the outer loop in order to achieve stability. Three states $(\phi \theta z)$ are selected as outputs and feedback linearization is used. $(\cdot)_f$ is written for fault condition. The input vector is

$$\mathbf{u}_f = \begin{bmatrix} u \\ \tau_q \\ \tau_r \end{bmatrix} \quad (4.44)$$

The above developed procedure for normal operation is used except that the angle ψ is not controlled and the input τ_p is calculated using the following equation

$$\tau_p = \frac{l}{2} \left(u - \frac{\tau_r}{d} \right) \quad (4.45)$$

CHAPTER 5

RESULTS AND DISCUSSION

To test the performance of the proposed controllers two different reference trajectories are selected. The first one is a square shaped trajectory in which the UAV vertically takes off to a desired level and then performs a trajectory in the shape of a square and also along the diagonal of the square as seen in Figure 5.1, where the 3D plot of the trajectories is plotted. The UAV is first simulated separately with FLC controller and the following results are obtained.

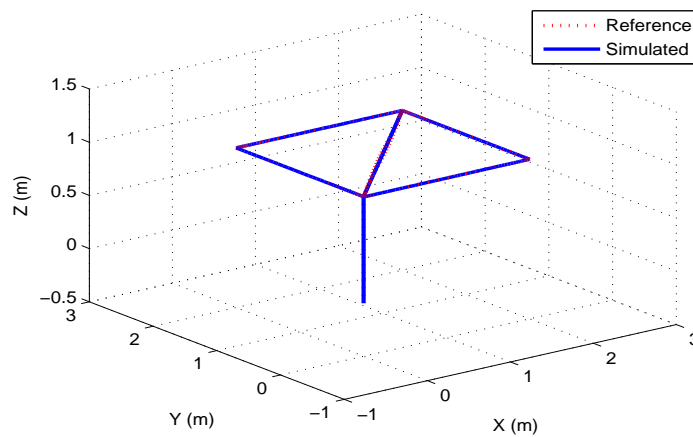


Figure 5.1: 3D view of the tracking.

The XY view of the trajectory is also plotted in Figure 5.2 to show the tracking in X and Y directions.

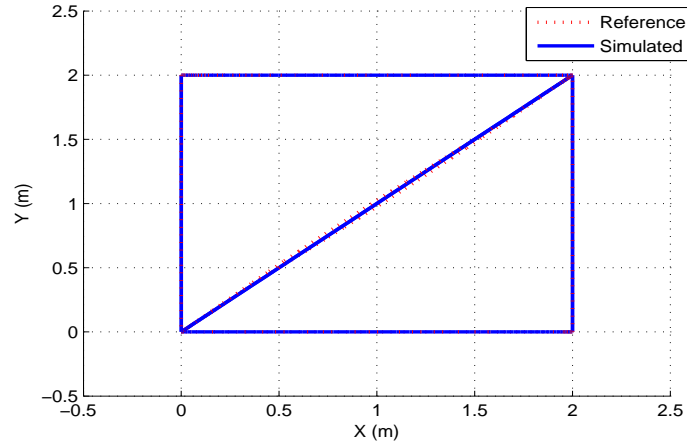


Figure 5.2: XY view of the tracking.

Also, the attitude angles are plotted separately in Figure (5.3-5.5). The desired angles ϕ_d and θ_d are generated by the outer loop and so, no reference is given for them by the designer. The angle ψ is selected to be zero and remains as such as seen in Figure 5.5.

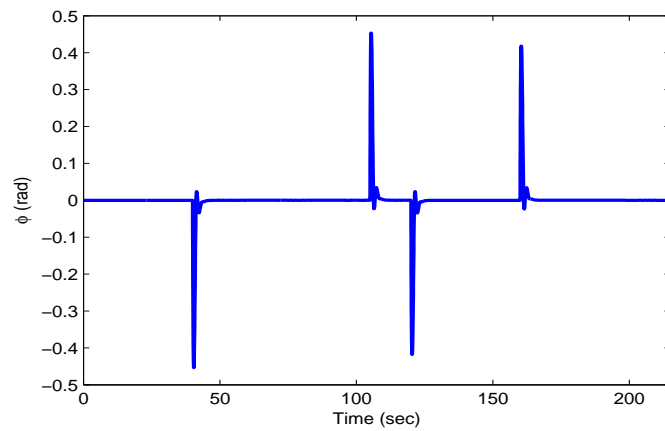


Figure 5.3: Angle ϕ .

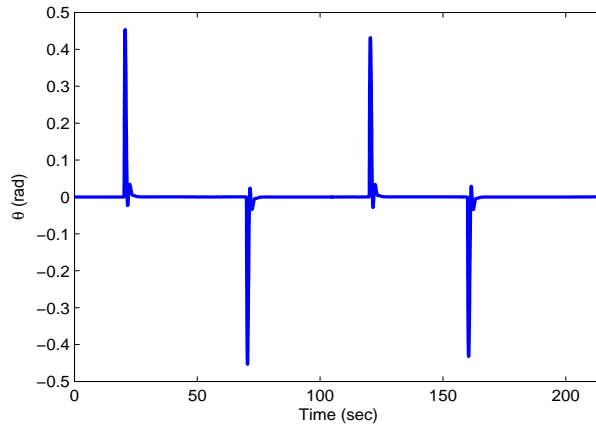


Figure 5.4: Angle θ .

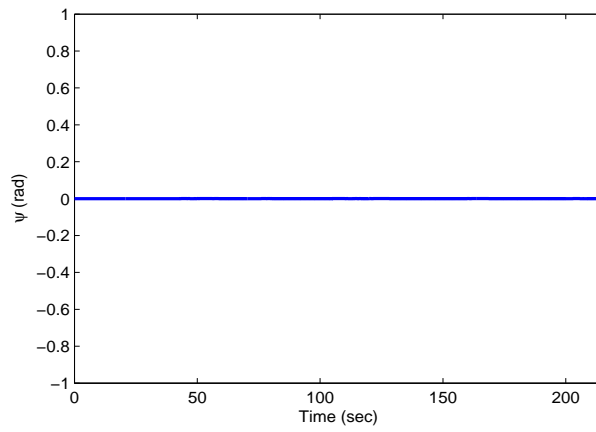


Figure 5.5: Angle ψ .

The translational motion represented in X,Y and Z co-ordinates is also plotted separately in Figure (5.6-5.8).

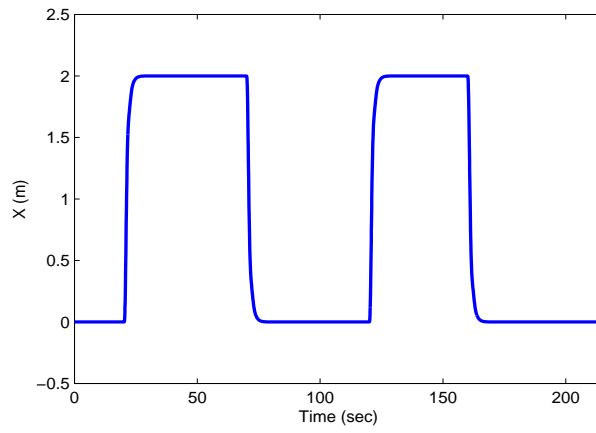


Figure 5.6: Position x .

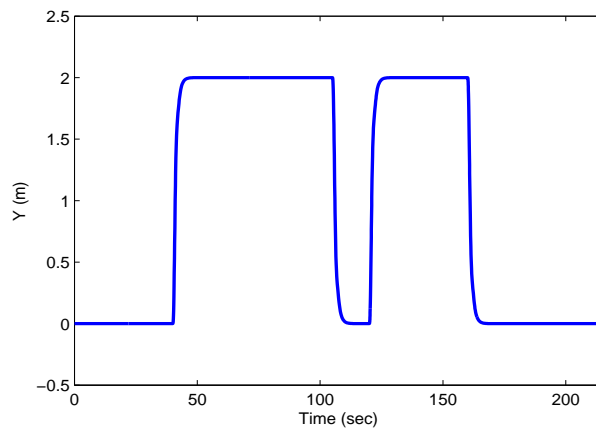


Figure 5.7: Position y .

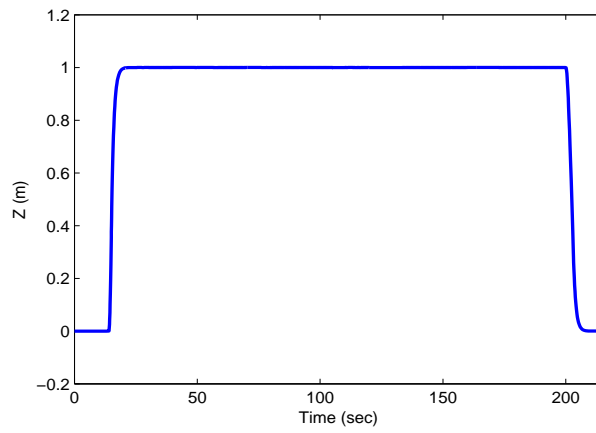


Figure 5.8: Position z .

The second reference trajectory is selected as a eight shaped trajectory where in the UAV takes off vertically and then follows a eight shaped path as seen in Figure 5.9.

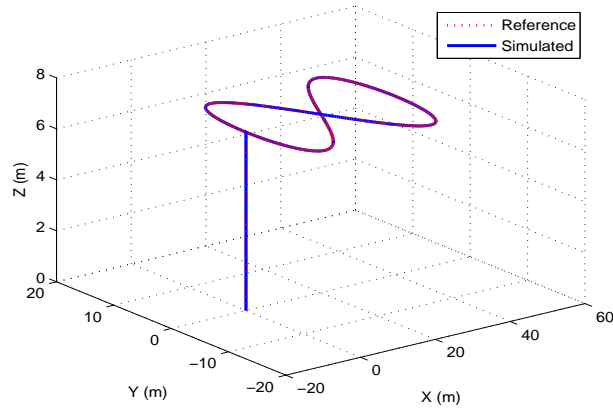


Figure 5.9: 3D view of the tracking.

The XY view of the eight shaped trajectory is also plotted in Figure 5.10. Also, the attitude angles are plotted separately in Figure (5.11-5.13). The desired

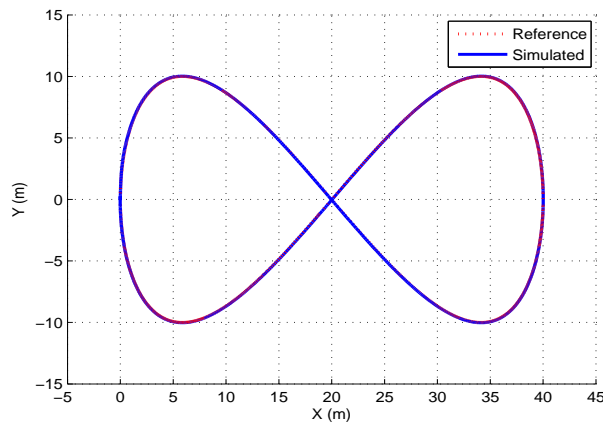


Figure 5.10: XY view of the tracking.

angles ϕ_d and θ_d are generated by the outer loop and so, no reference is given for them by the designer. The angle ψ is selected to be zero and remains as such as seen in Figure 5.13. The translational motion represented in X,Y and Z

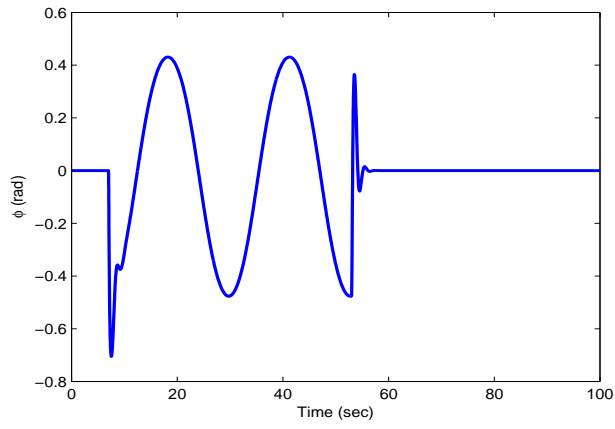


Figure 5.11: Angle ϕ .

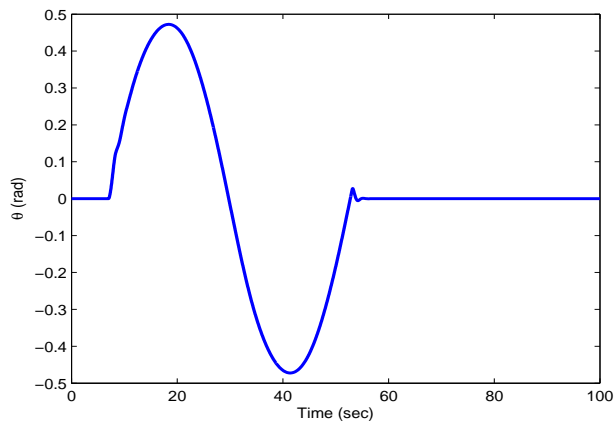


Figure 5.12: Angle θ .

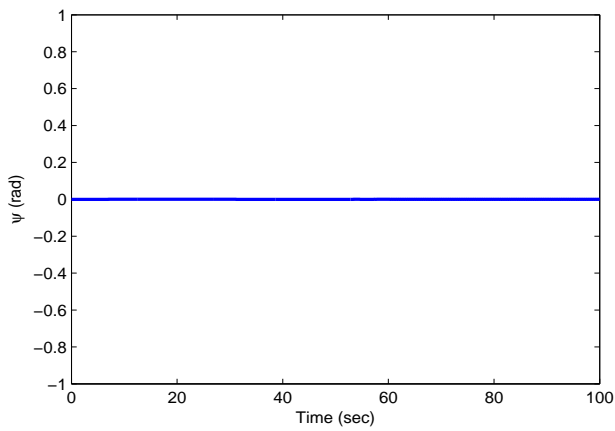


Figure 5.13: Angle ψ .

co-ordinates is also plotted separately in Figure (5.14-5.16).

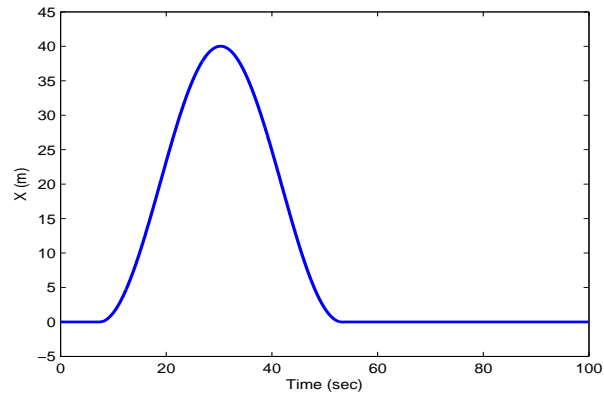


Figure 5.14: Position x.

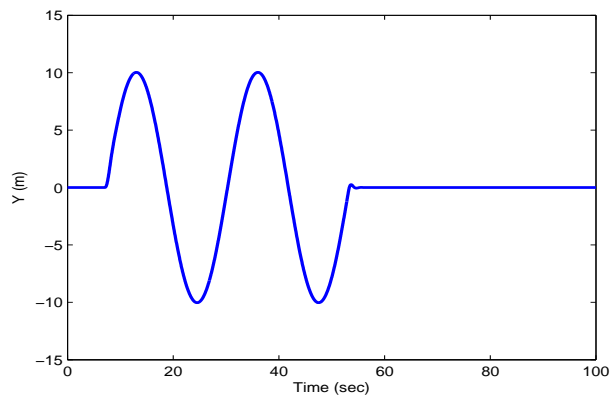


Figure 5.15: Position y.

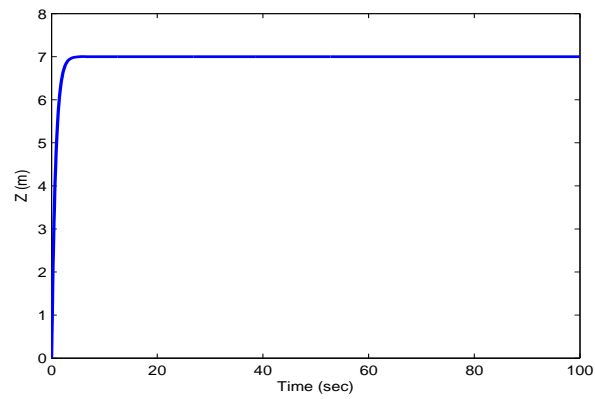


Figure 5.16: Position z.

After the normal operating UAV the fault condition of it is simulated. The UAV with a fault in the rotor is simulated. The path is selected as the square shape where even after the fault the UAV performs good tracking and then lands safely for repair without any further damage and protecting the expensive parts of it. The initial conditions are set to any position supposed to be the position when the fault is detected and the fault tolerant controller then starts functioning. The initial condition here is assumed to be (5,9,10) for the x,y and z position. From here the UAV is first stabilized to a point (10,0,9). Then it follows the path to (10,10,9). From here it moves to (0,10,9) and then it reaches to (0,0,9). From this point it is made to land safely at (0,0,0). The 3D view of the trajectory is plotted in Figure 5.17, where it can be seen that the UAV starts from a fault position and then tries to follow the desired trajectory, but as one of its rotor has a fault and is not working, the UAV is spinning in one direction and tries to track the reference in that condition. The XY view of it can be seen in Fig. 5.18. The attitude angles

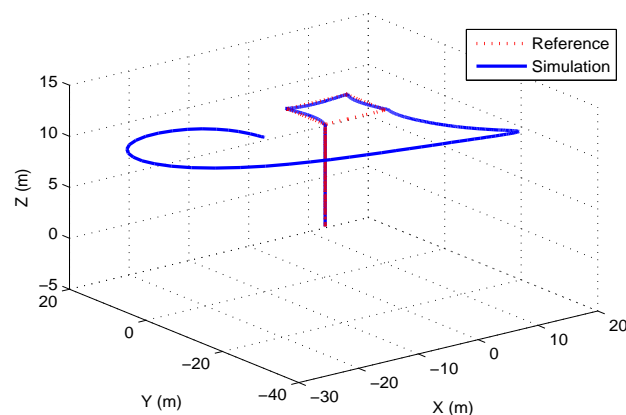


Figure 5.17: 3D view of the tracking.

and the X,Y and Z positions are plotted as well in Figure (5.19-5.24).

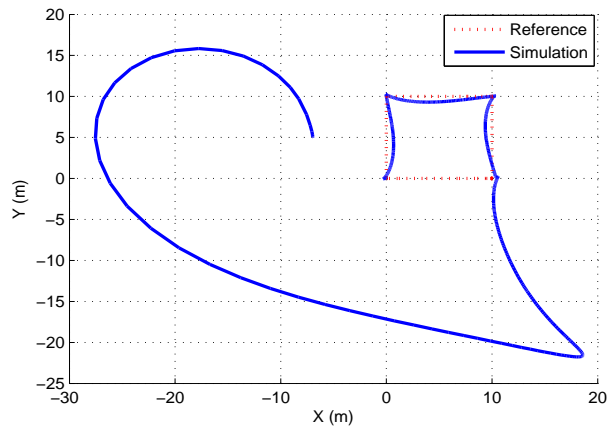


Figure 5.18: XY view of the tracking.

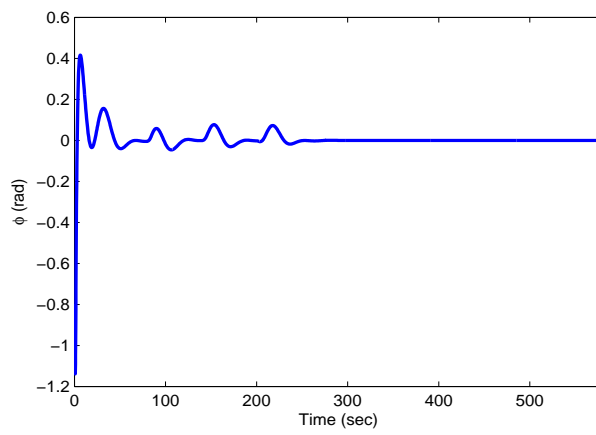


Figure 5.19: Angle ϕ .

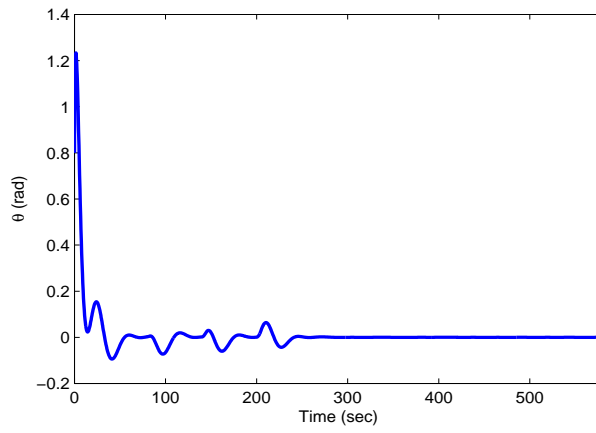


Figure 5.20: Angle θ .

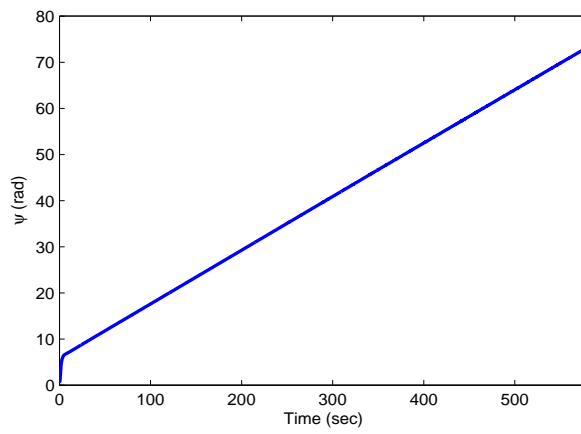


Figure 5.21: Angle ψ .

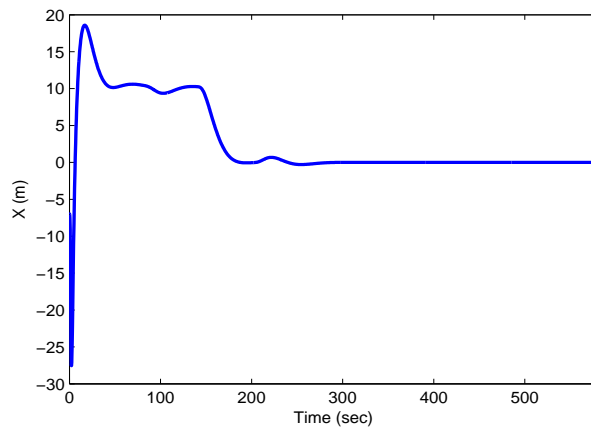


Figure 5.22: Position x .

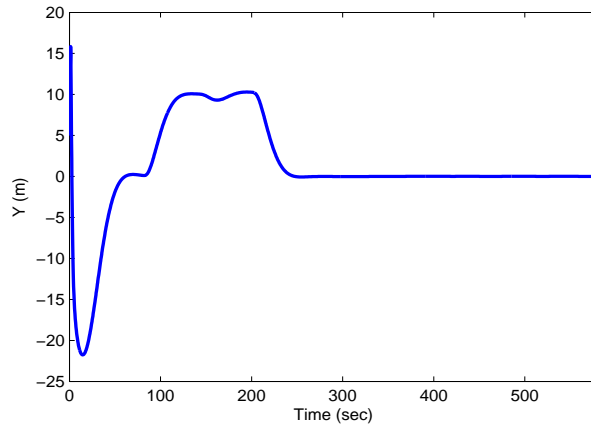


Figure 5.23: Position y .

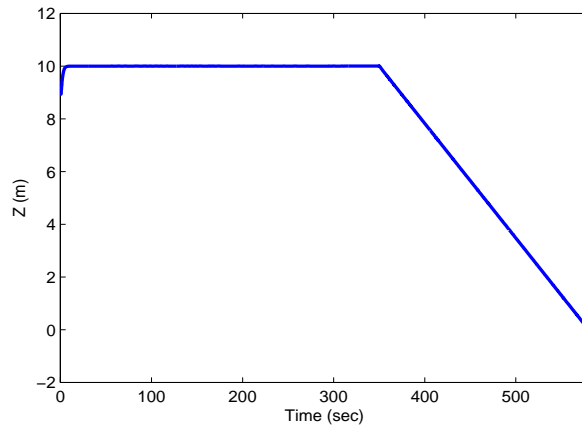


Figure 5.24: Position z .

The NN-FLC is implemented on the UAV. Here too, two different trajectories of square shaped and eight shaped are selected to validate the proposed controller. This also shows good tracking performance with the NN estimating the unknown dynamics of the UAV. The 3D plot of the trajectories is shown in Figure 5.25. The XY view of the trajectory is plotted in Fig. 5.26. The attitude angles and position are separately plotted in Figure (5.27-5.32).

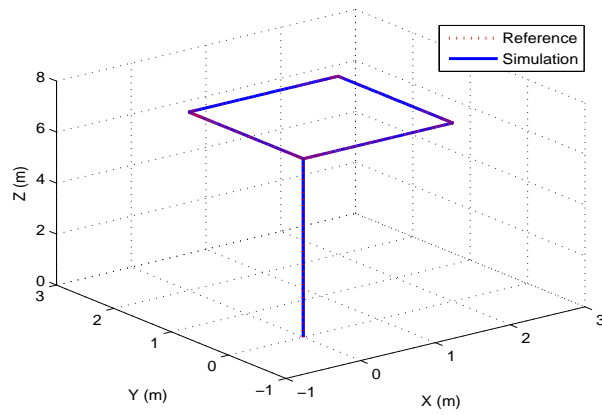


Figure 5.25: 3D view of the tracking.

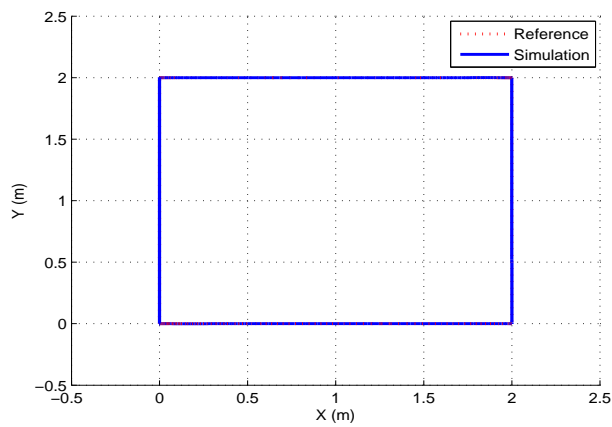


Figure 5.26: XY view of the tracking.

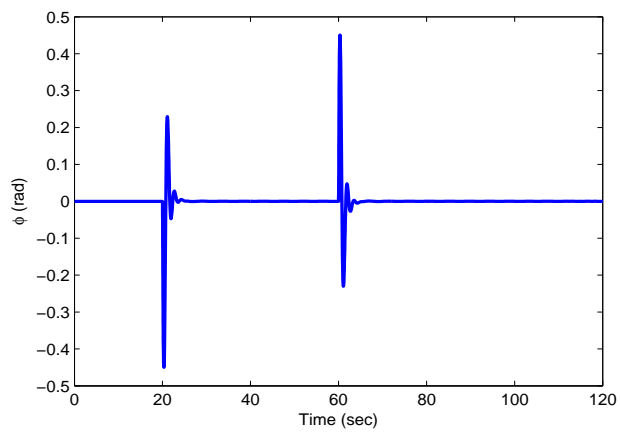


Figure 5.27: Angle ϕ .

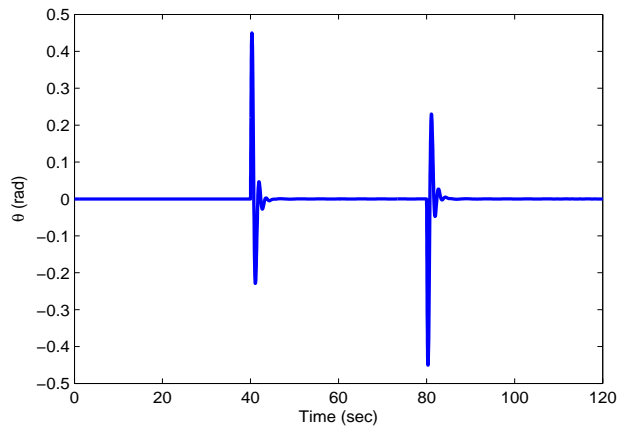


Figure 5.28: Angle θ .

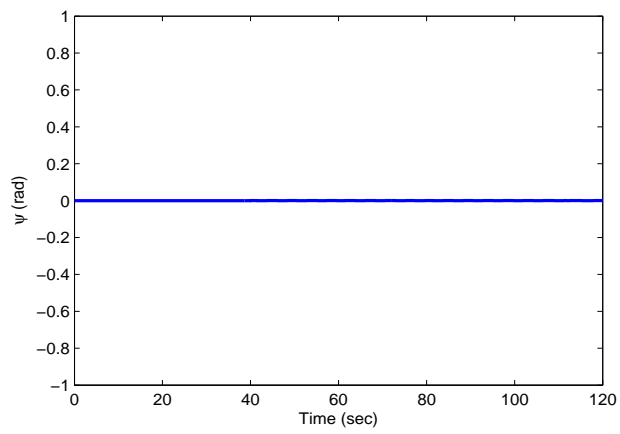


Figure 5.29: Angle ψ .

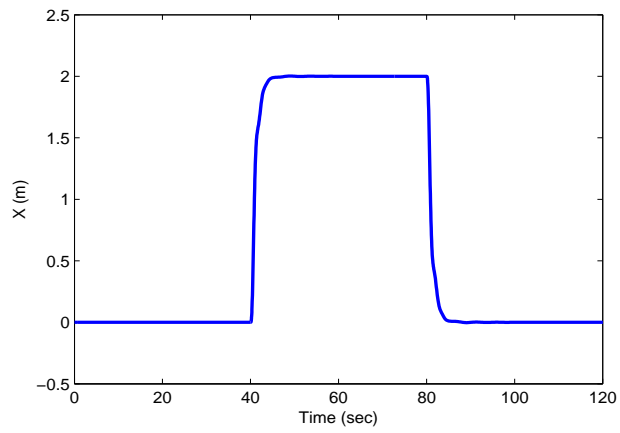


Figure 5.30: Position x .

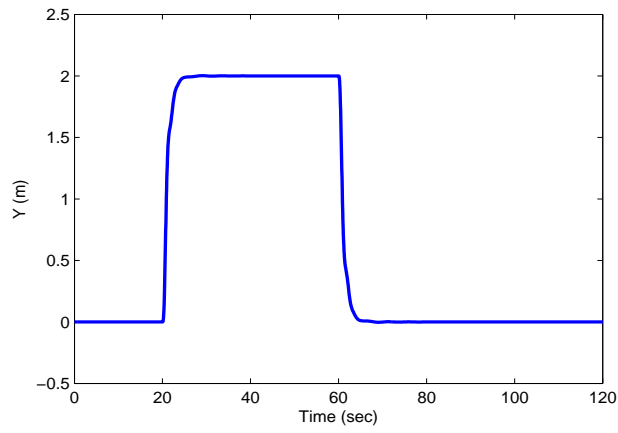


Figure 5.31: Position y.

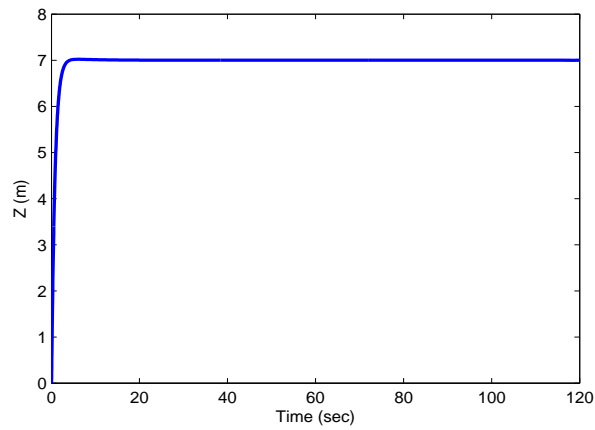


Figure 5.32: Position z.

The trajectory is now selected as eight shaped and the 3D trajectory of it is plotted in Figure 5.33. The XY view of the trajectory is plotted in Fig. 5.34. The attitude angles and position are separately plotted in Figure (5.35-5.40).

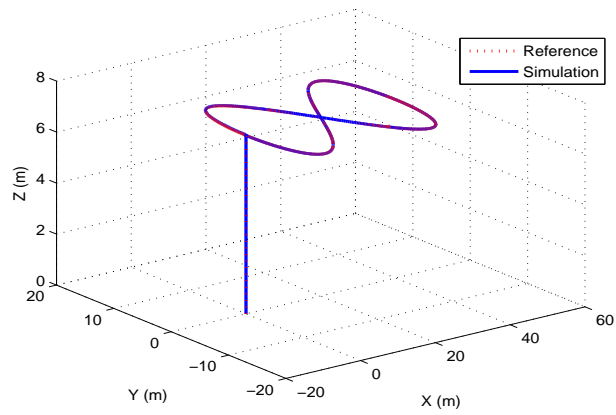


Figure 5.33: 3D view of the tracking.

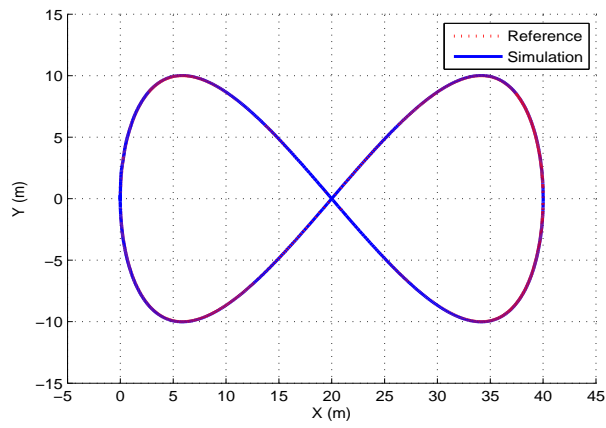


Figure 5.34: XY view of the tracking.

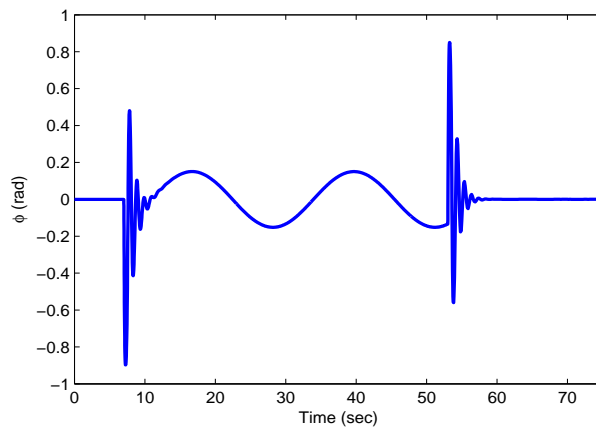


Figure 5.35: Angle ϕ .

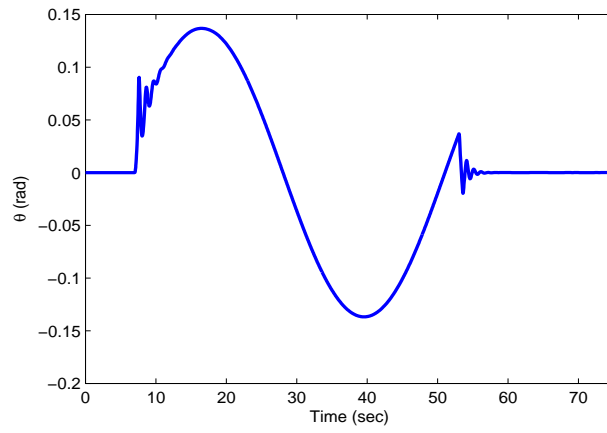


Figure 5.36: Angle θ .

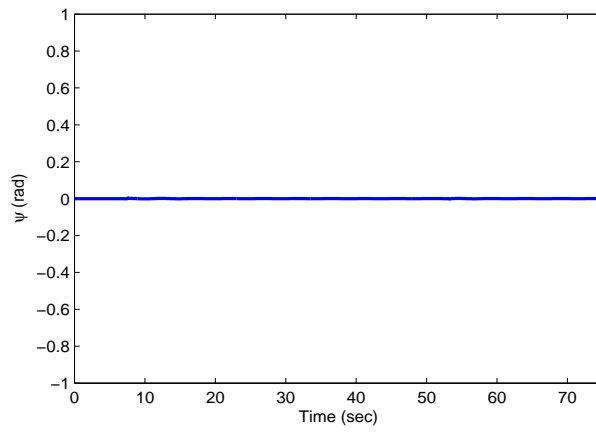


Figure 5.37: Angle ψ .

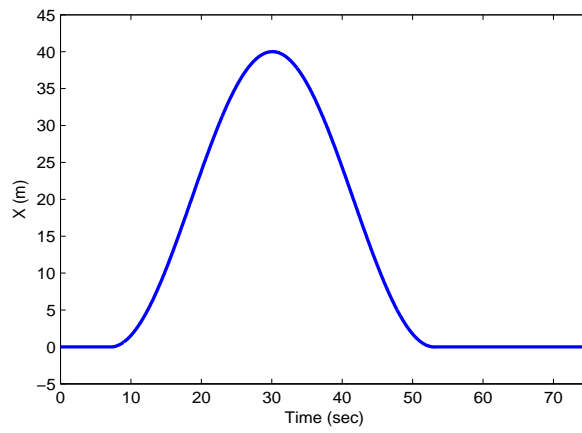


Figure 5.38: Position x .

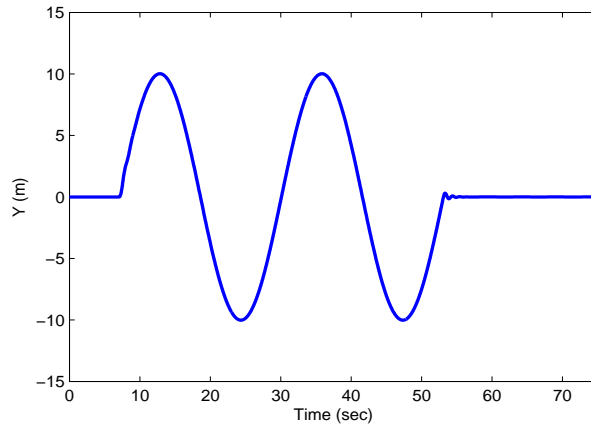


Figure 5.39: Position y.

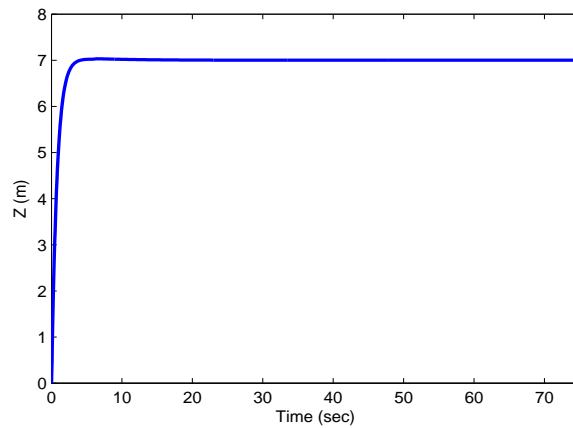


Figure 5.40: Position z.

The slung load is now augmented to the UAV model and then without implementing the controller, the model is simulated to see the effect of the load on the UAV. Two different scenarios are considered in which, the UAV with the load moves in straight paths in the first scenario and in the second scenario it moves in curved paths which are represented by the eight shaped trajectory. In Figure 5.41, the 3D view of the trajectory is plotted and in Figure 5.42, the XY view is plotted which represents the first scenario.

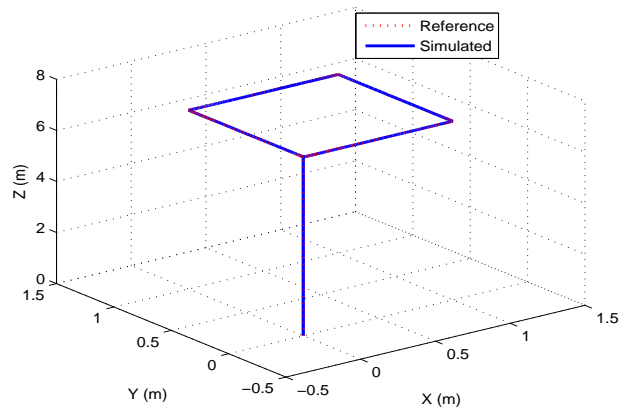


Figure 5.41: 3D view of the tracking.

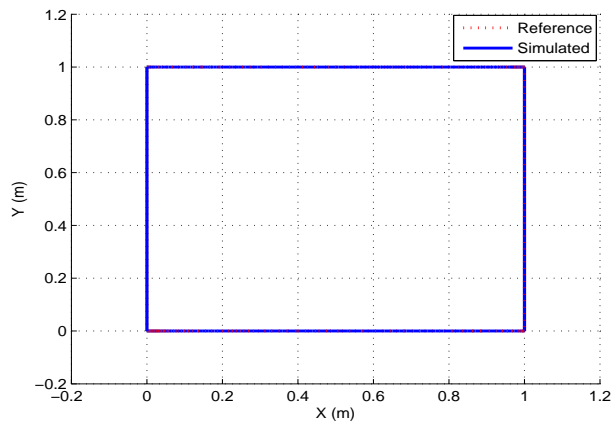


Figure 5.42: XY view of the tracking.

The load swing angles are plotted in Figure 5.43 and Figure 5.44, in which it can be seen that the load swings with constant amplitude as the UAV moves. Whenever, the UAV moves the load follows the UAV and when the UAV stops the load swings. This is undesirable and has to be controlled as it can cause damage to the load and also make the system unstable.

The forces are plotted in Figure (5.45-5.47).

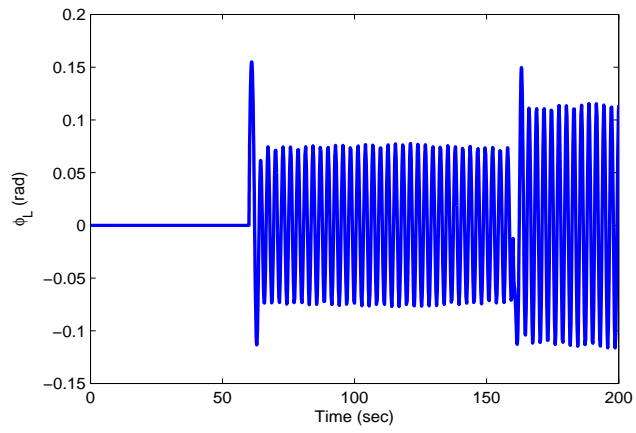


Figure 5.43: Load Swing Angle ϕ_L .

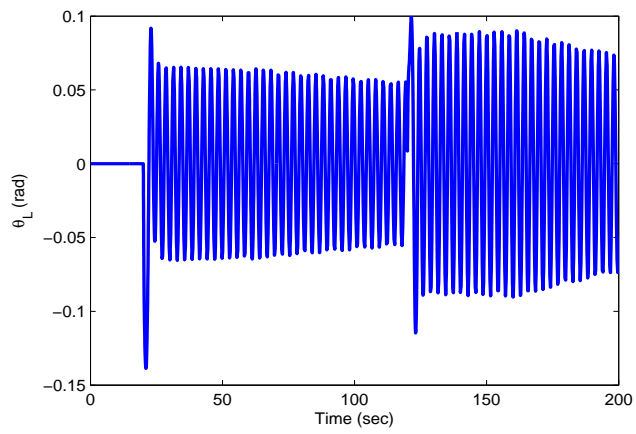


Figure 5.44: Load Swing Angle θ_L .

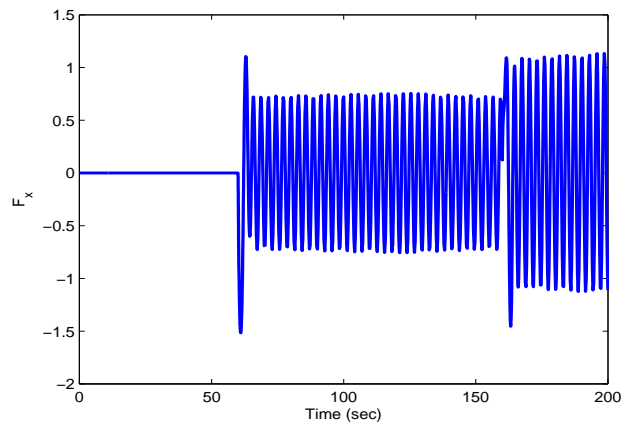


Figure 5.45: Force F_x

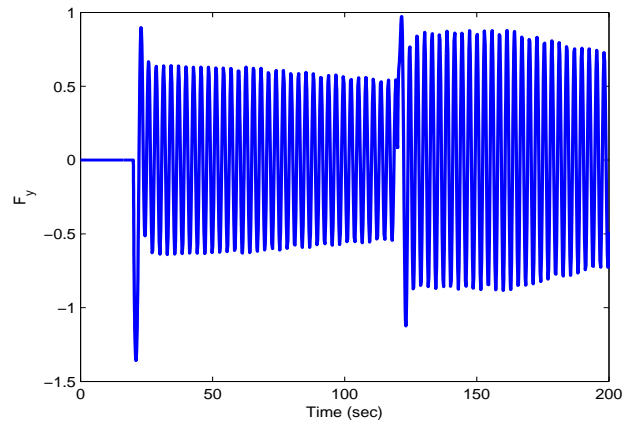


Figure 5.46: Force F_y .

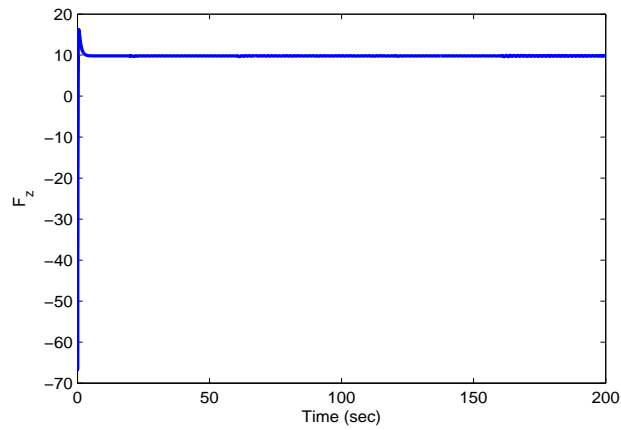


Figure 5.47: Force F_z .

The attitude angles of the UAV are plotted in Figure (5.48-5.50) and the translational motion is plotted separately in Fig. (5.51-5.53).

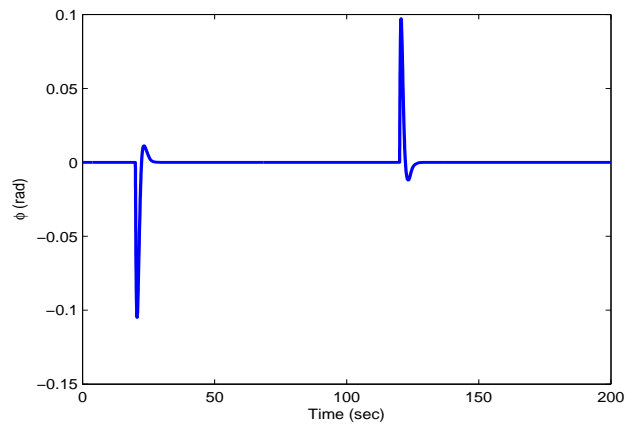


Figure 5.48: Angle ϕ .

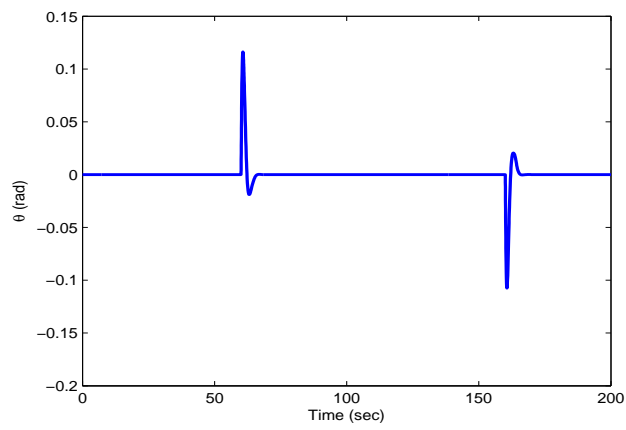


Figure 5.49: Angle θ .

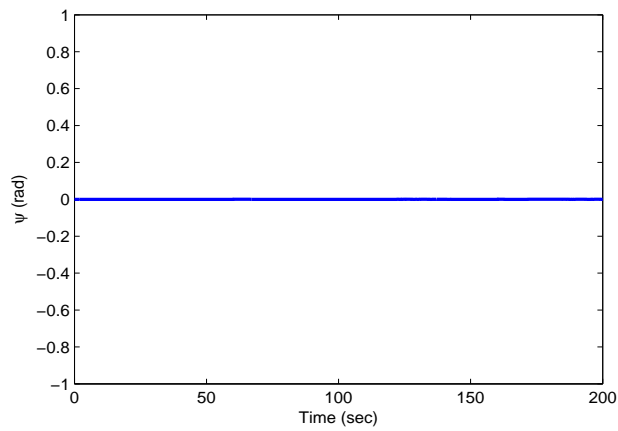


Figure 5.50: Angle ψ .

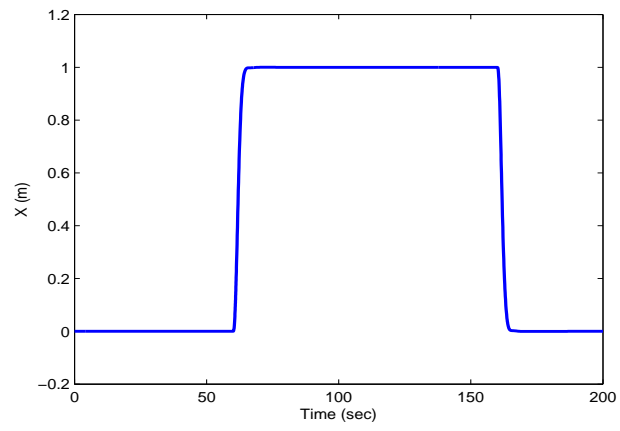


Figure 5.51: Position x.

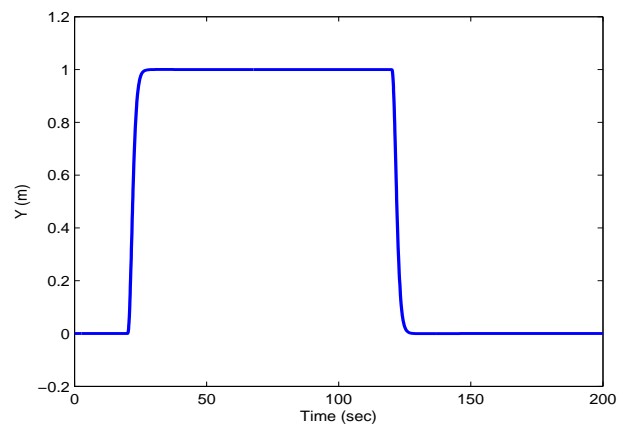


Figure 5.52: Position y.

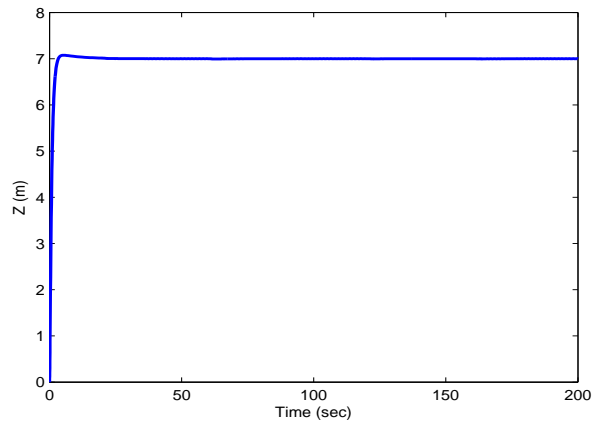


Figure 5.53: Position z .

The second scenario is now considered, the path followed is changed to '8' shape and the system without the delayed feedback controller is simulated. The trajectory in 3D is plotted in Figure 5.54. The XY view of the trajectory is plotted in Figure 5.55.

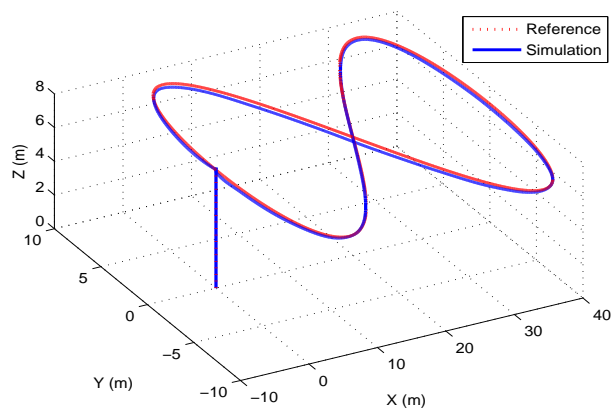


Figure 5.54: 3D view of the tracking.

The load swing angles are plotted in Figure 5.56 and Figure 5.57 and it can be seen that the load oscillates continuously and it need to be stablized, if not it can damage the load which is undesirable.

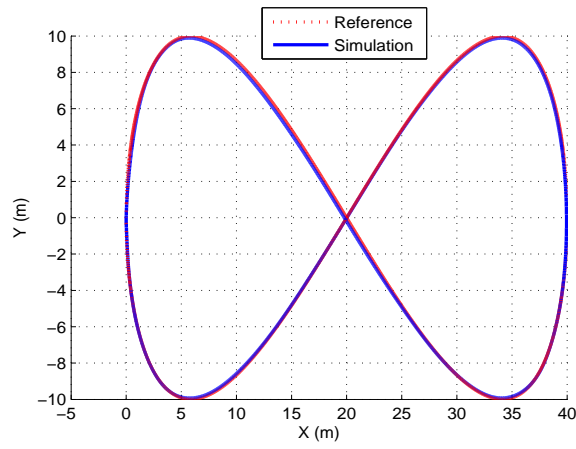


Figure 5.55: XY view of the tracking.

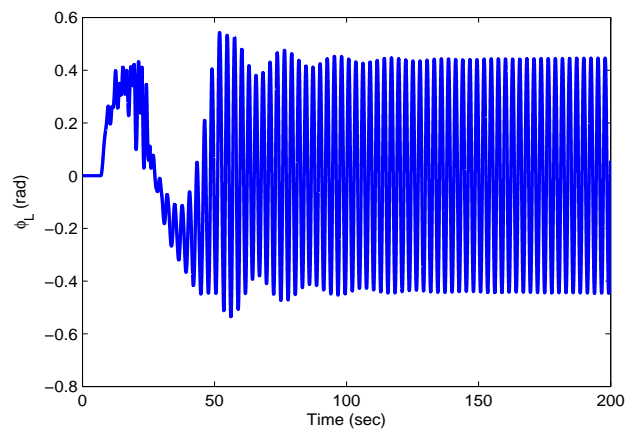


Figure 5.56: Load Angle ϕ_L .

The attitude angles and the translational motion of the UAV are plotted separately as well in Figure (5.58-5.63).

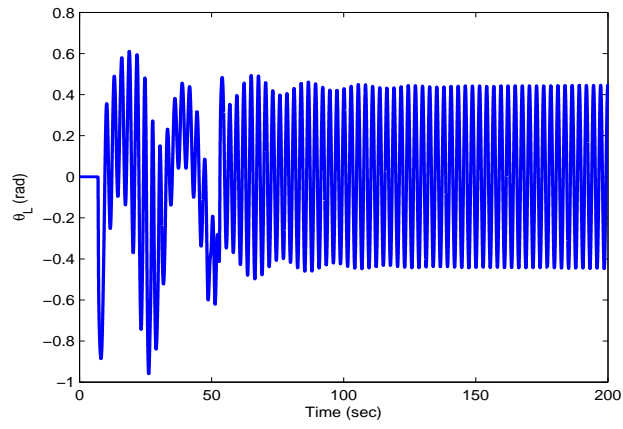


Figure 5.57: Load Angle θ_L .

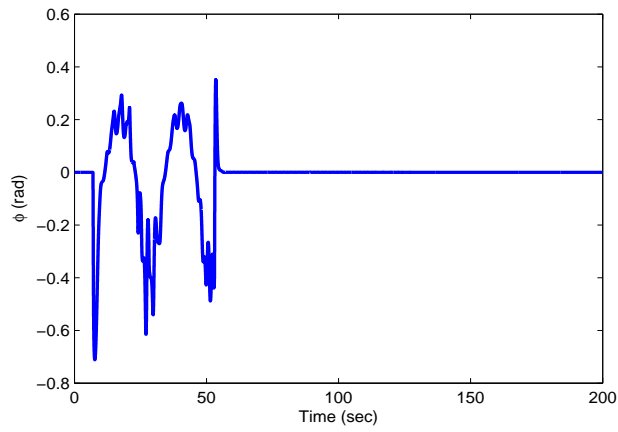


Figure 5.58: Angle ϕ .

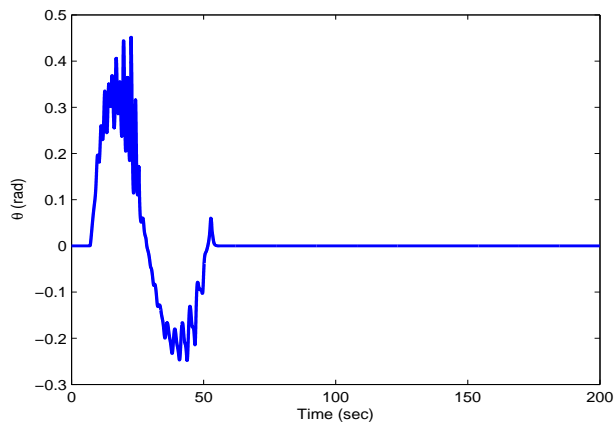


Figure 5.59: Angle θ .

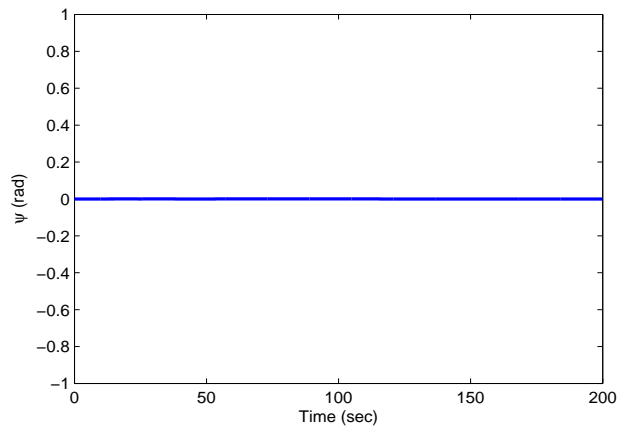


Figure 5.60: Angle ψ .

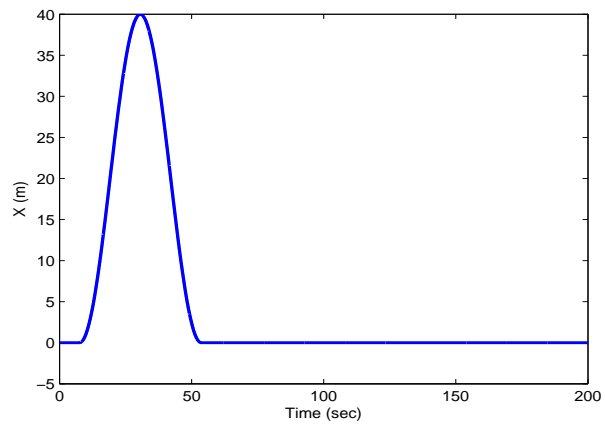


Figure 5.61: Position x .

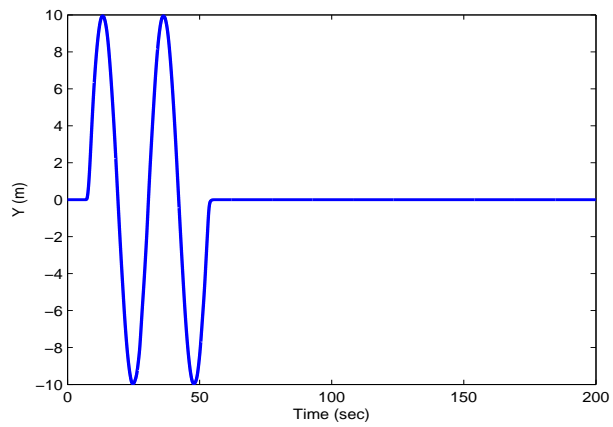


Figure 5.62: Position y .

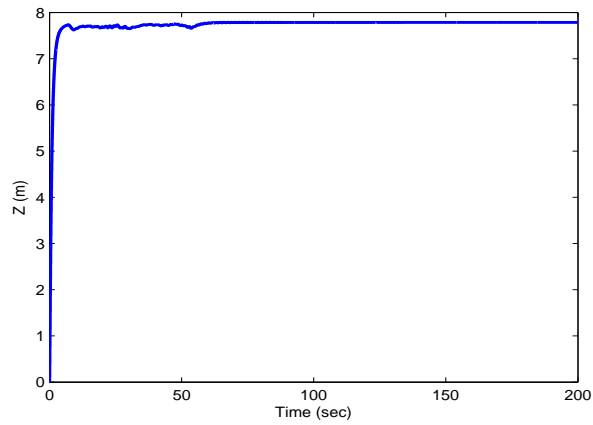


Figure 5.63: Position z .

The time-delayed feedback controller which is the anti-swing controller is now implemented to stabilize the load swing angles to zero and the simulation is carried out. In this case as well both the scenarios are considered. The following results are for the square shaped trajectory. The 3D view of the trajectory is plotted in Figure 5.64. The XY view of the trajectory is plotted in Figure 5.65

The load angles are plotted in Figure and Figure and it can be seen from these

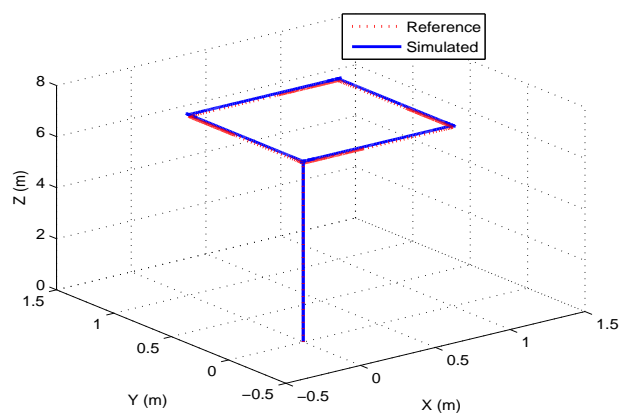


Figure 5.64: 3D view of the tracking.

figures that the load angles are stabilized to zero by the anti-trajectory controller.

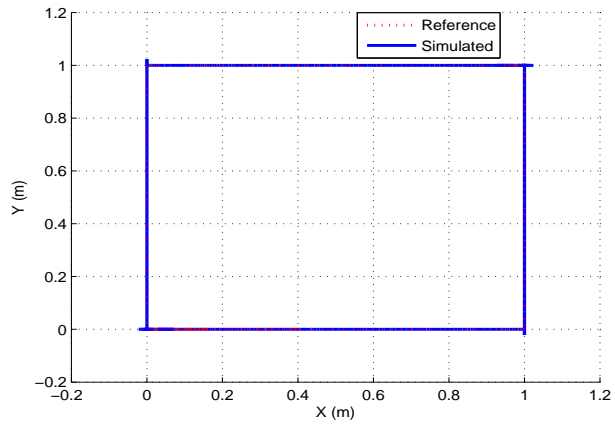


Figure 5.65: XY view of the tracking.

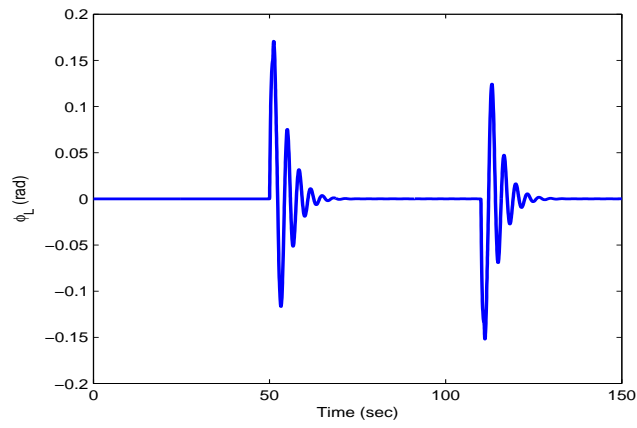


Figure 5.66: Load Angle ϕ_L .

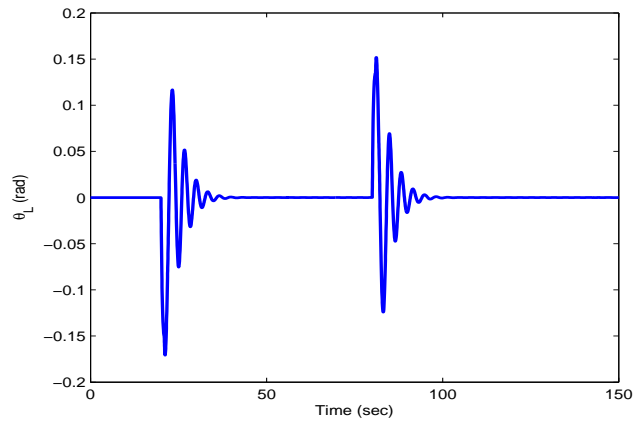


Figure 5.67: Load Angle θ_L .

The attitude angles and the translational motion of the UAV are plotted separately in Figure (5.68-5.73).

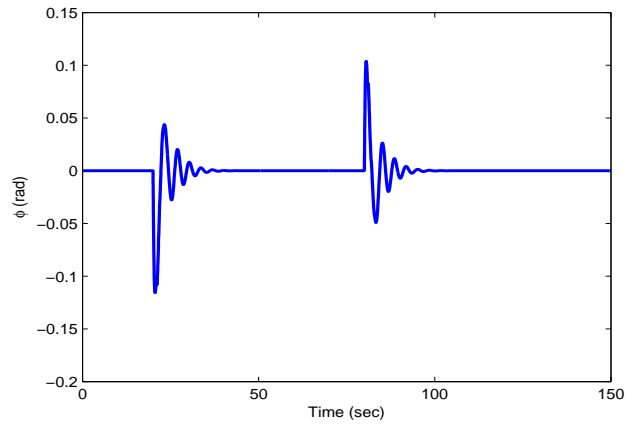


Figure 5.68: Angle ϕ .

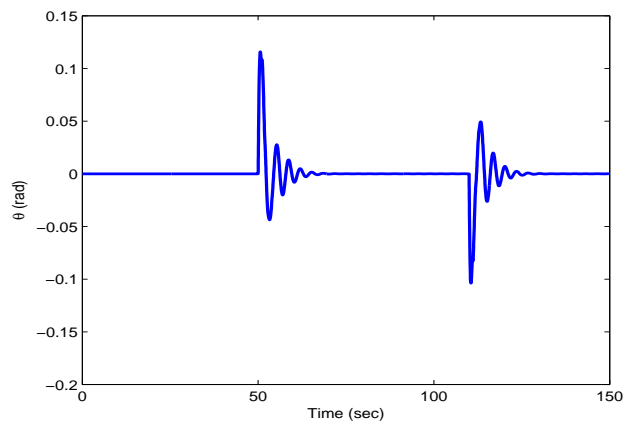


Figure 5.69: Angle θ .

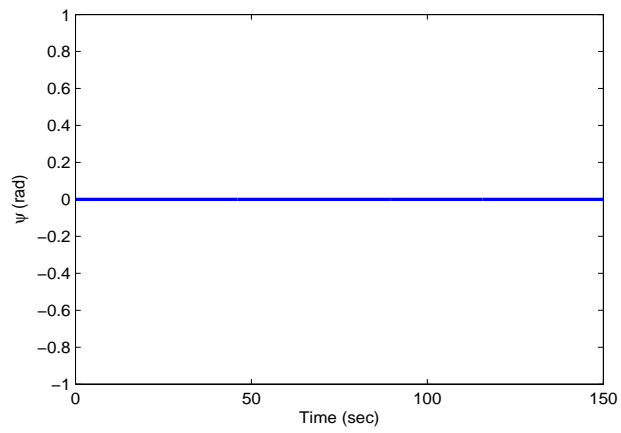


Figure 5.70: Angle ψ .

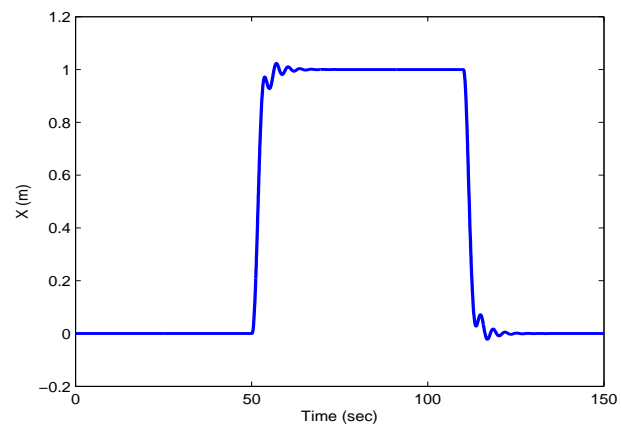


Figure 5.71: Position x .

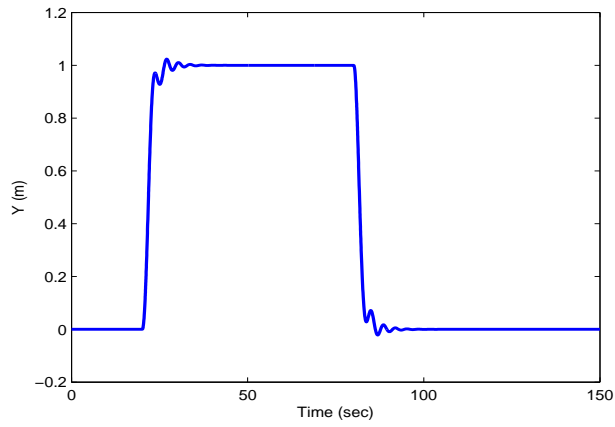


Figure 5.72: Position y.

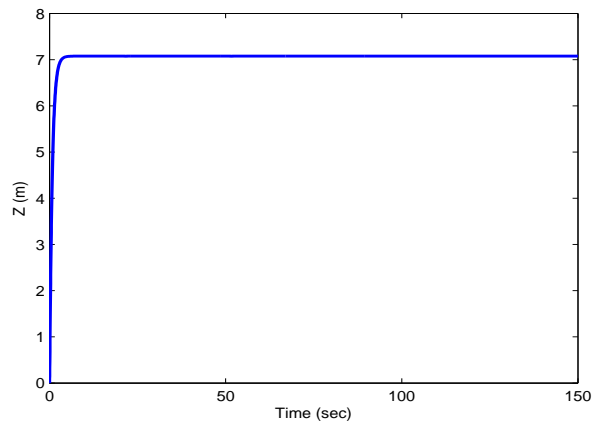


Figure 5.73: Position z.

The forces are plotted in Figure (5.74-5.76) and it can be seen that these forces are also suppressed and reduced to zero.

In the '8' shape trajectory as well the delayed feedback controller is added and simulated. the following results are obtained. The load angles are suppressed and are stabilized to zero by the anti-swing controller as seen in the Figure 5.79 and Figure 5.80 The angle ψ remains zero as it is not required.

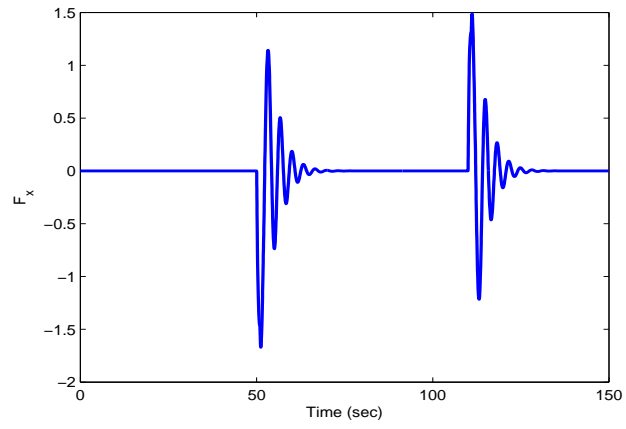


Figure 5.74: Force F_x

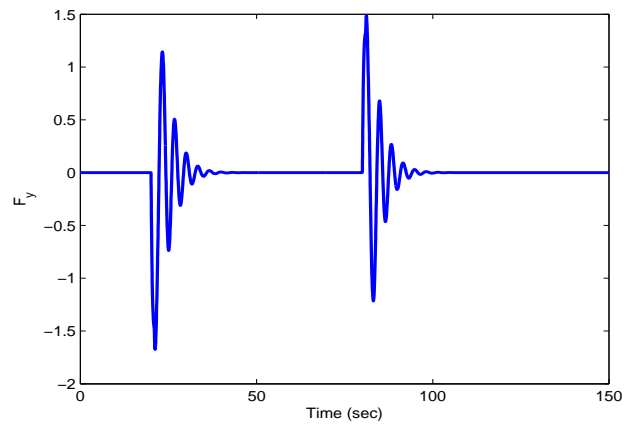


Figure 5.75: Force F_y .

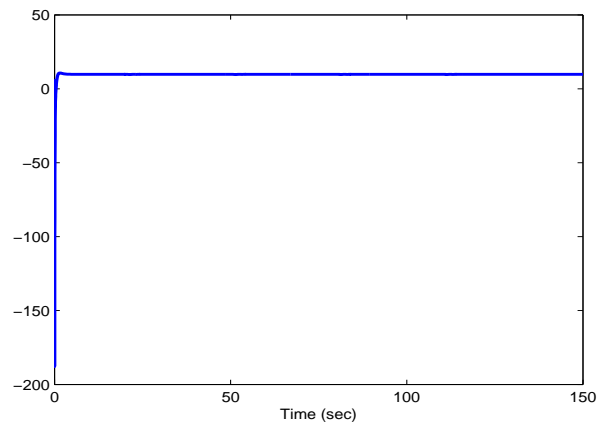


Figure 5.76: Force F_z .

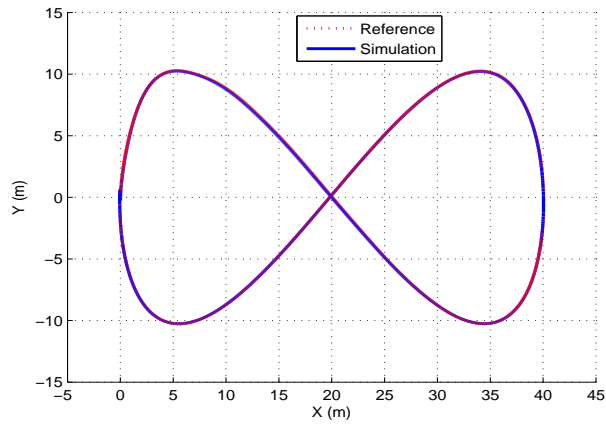


Figure 5.77: XY view of the tracking.

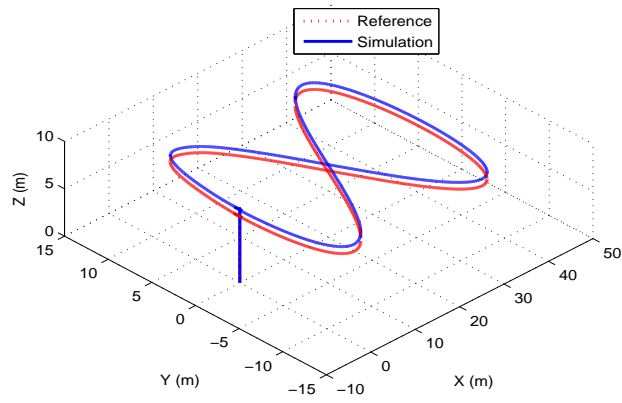


Figure 5.78: 3D view of the tracking.

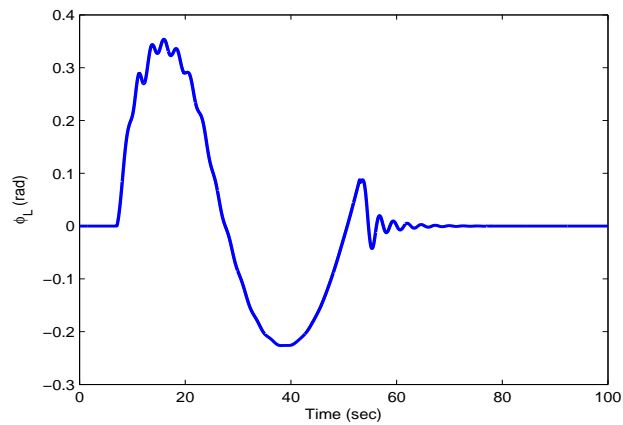


Figure 5.79: Load Angle ϕ_L .

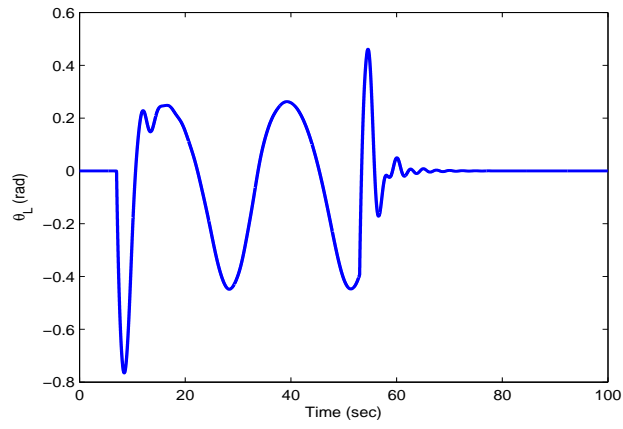


Figure 5.80: Load Angle θ_L .

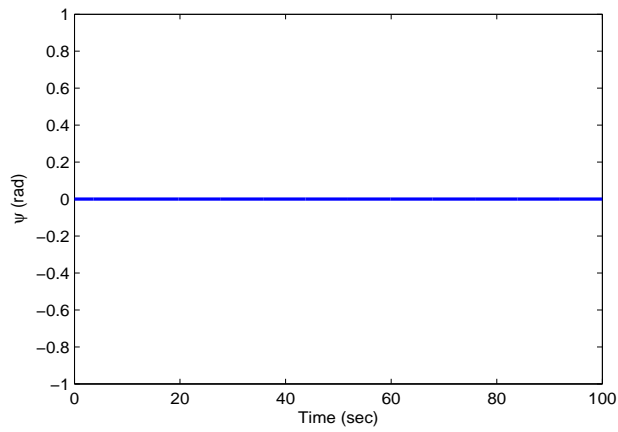


Figure 5.81: Angle ψ .

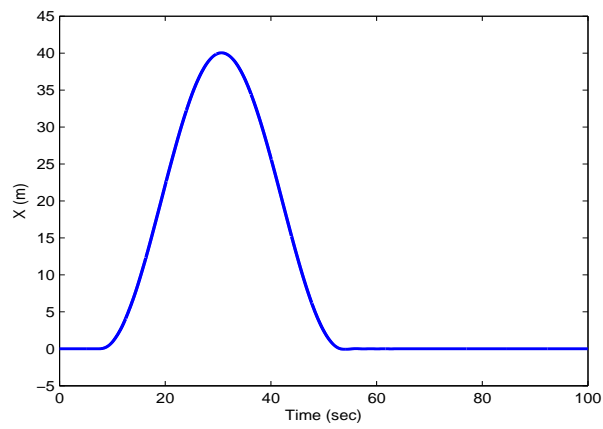


Figure 5.82: Position x .

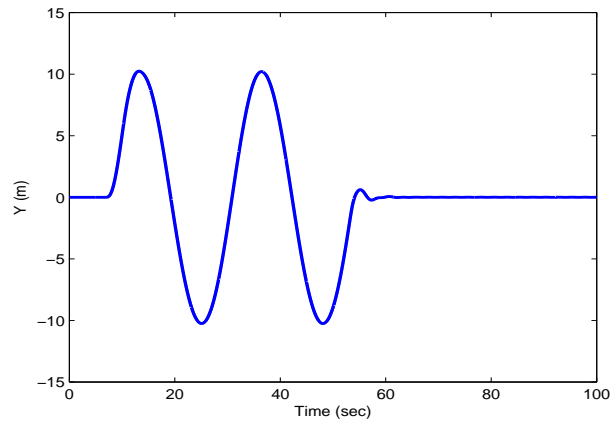


Figure 5.83: Position y .

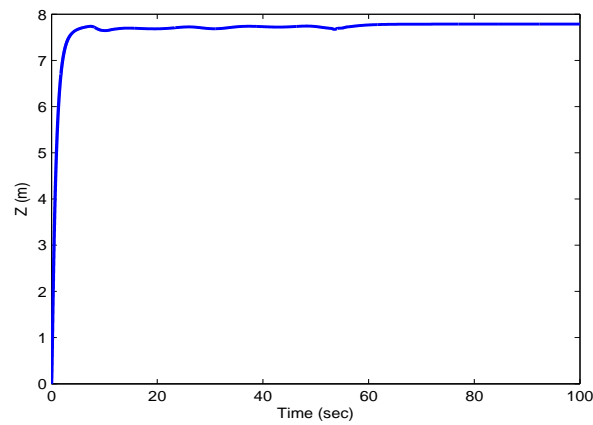


Figure 5.84: Position z .

The comparison of NN-FLC and the FLC is shown below where it is seen that the NN is very effective in estimating the dynamics of the system which are assumed to be unknown and it gives the desired tracking and performance. In the figures Figure 5.85 and Figure 5.86, the x and y positions are compared to prove that the NN estimates the dynamics very well.

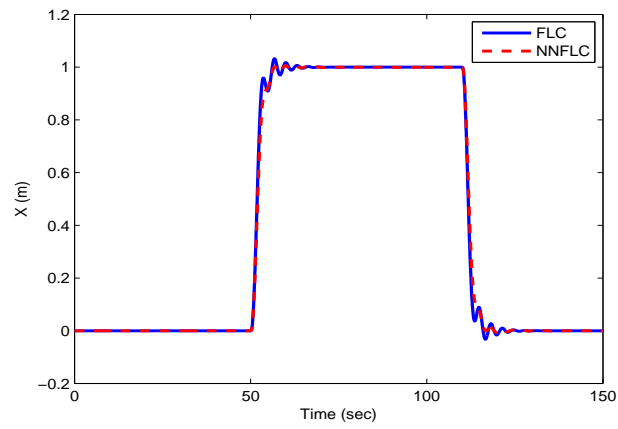


Figure 5.85: Position x

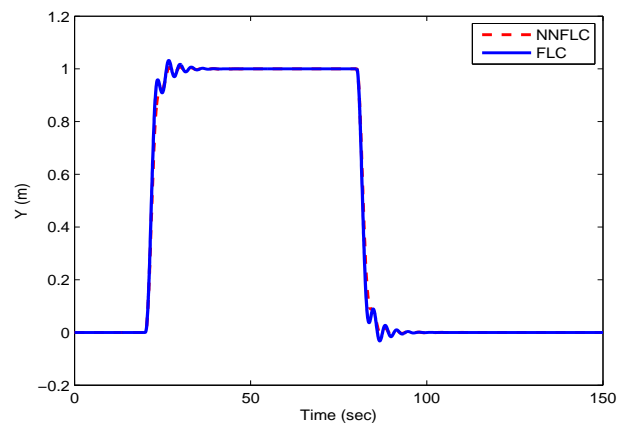


Figure 5.86: Position y

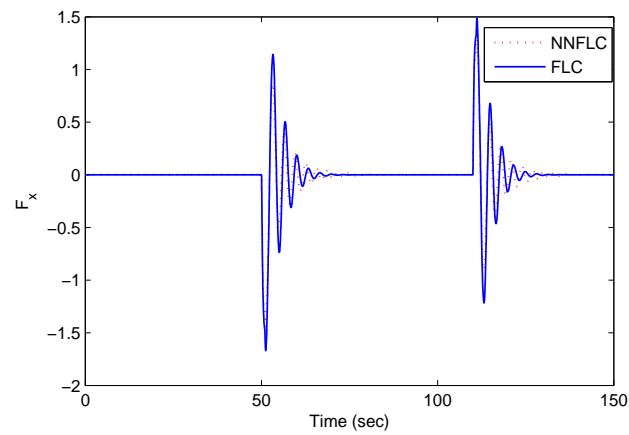


Figure 5.87: Force F_x

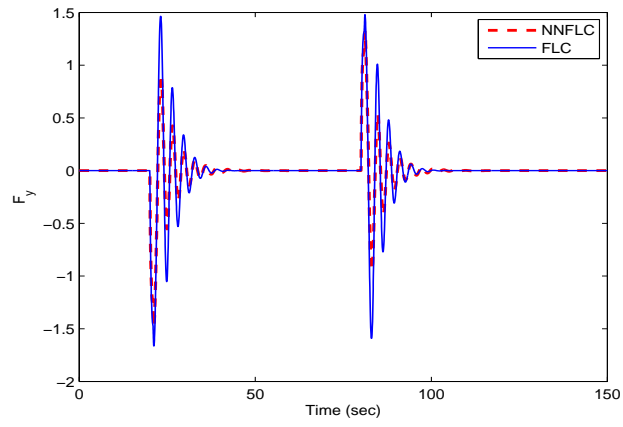


Figure 5.88: Force F_y

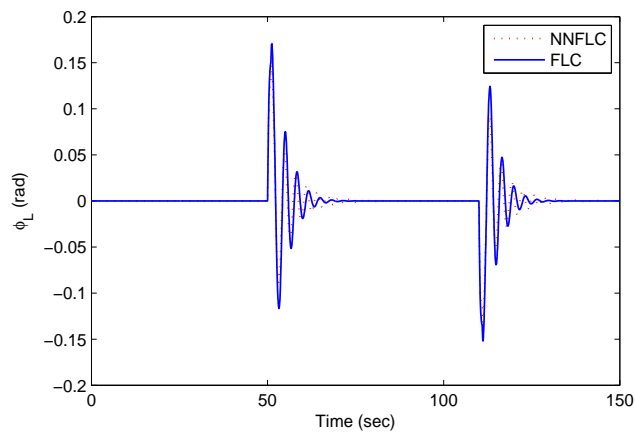


Figure 5.89: Load Angle ϕ_L

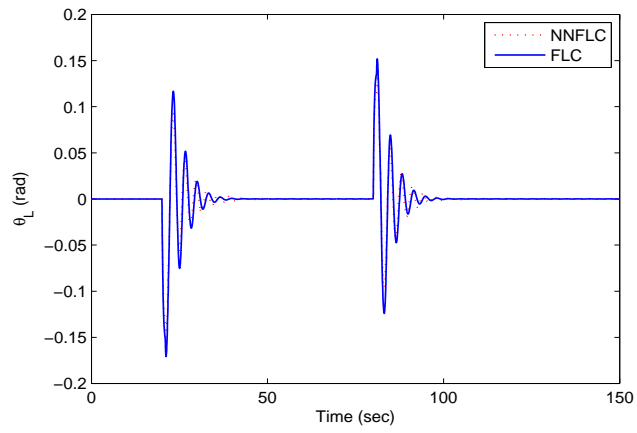


Figure 5.90: Load Angle θ_L

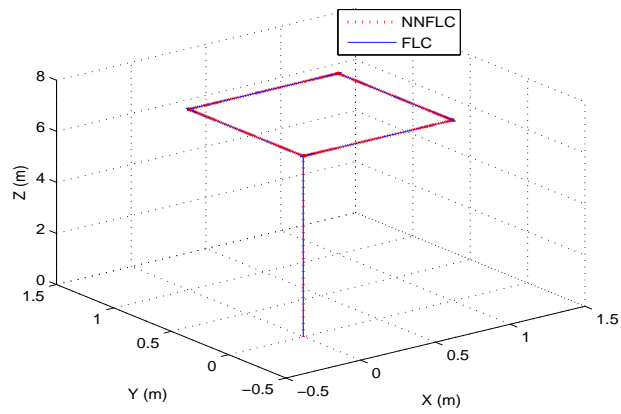


Figure 5.91: 3D View of the trajectory

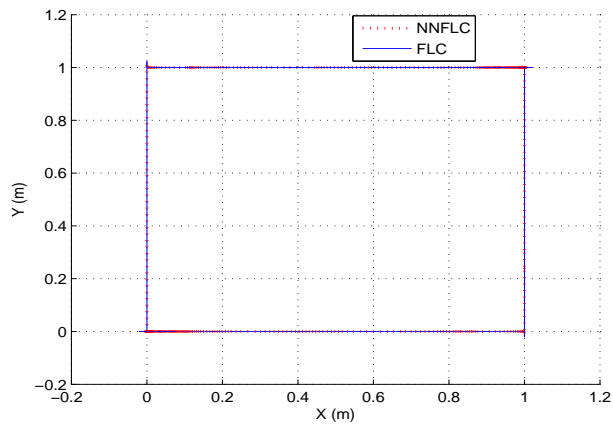


Figure 5.92: XY View of the trajectory

The damping term in the slung load which is represented by the term D_L is due to the resistance of air in the load. The effect of the damping can be seen in the Figure 5.93, where it can be noticed that if damping is present the load swing damps a little faster than without the damping. Also, the FLC is not able to adapt

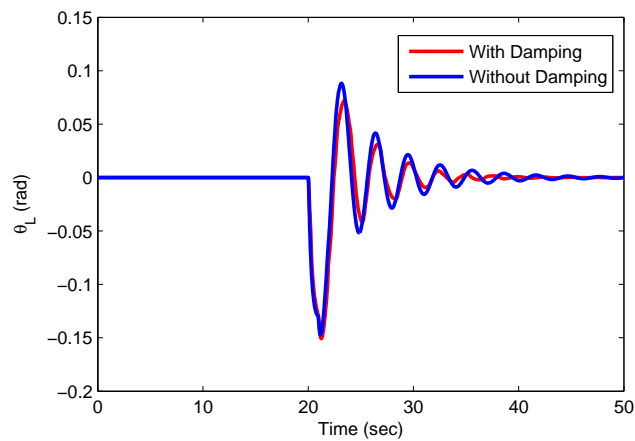


Figure 5.93: Damping Effect

to change in parameters of the model. Whereas, any change in the parameters in the inner loop will be adaptive while using the NNFLC and the controller adapts to the change in parameters and gives the desired performance. The normal FLC is not able to handle the change in parameters and the quadrotor crashes as seen in Figure 5.94 and Figure 5.95.

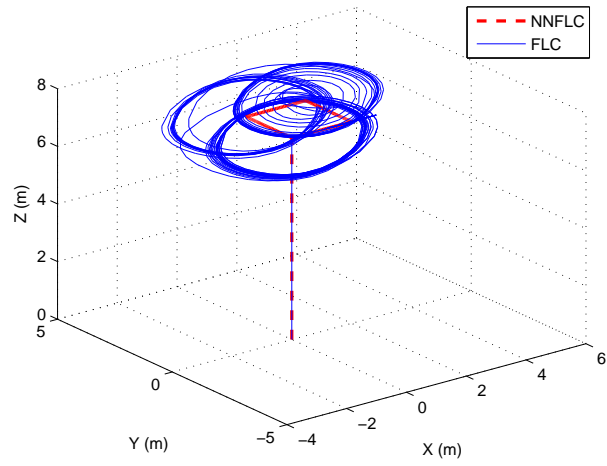


Figure 5.94: 3D View of the trajectory

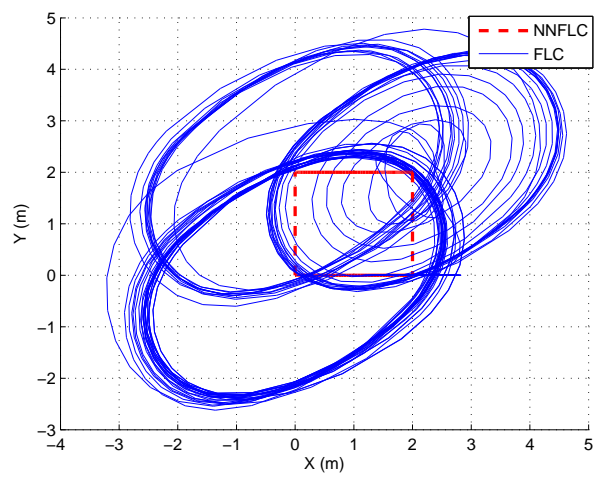


Figure 5.95: XY View of the trajectory

CHAPTER 6

CONCLUSIONS

A model for UAV with load is developed as a model is not yet completely established in the literature. A controller to control the UAV and the load swing angles was designed by developing them separately and then augmenting them together. Feedback Linearization controller was selected to control the UAV. As the feedback linearization controller has a few restrictions, neural networks are employed to overcome these restrictions. The neural network based feedback linearization controller to control the UAV works very well and approximates the dynamics of the model which are considered to be unknown. The time-delayed feedback controller stabilizes the load swing angle to zero and can be practically implemented as it has been implemented in cases of gantry tower loads, helicopters and applications in chaos theory. The damping in the air if included further helps in stabilizing the load swing.

6.1 Future Work

The effect of disturbance can be modeled and the controller can be tested against disturbances. The gains and time delay in the anti-swing controller can be obtained by using an optimization technique and thus, optimal gains can be obtained more accurately and easily. Load transportation using many cables can be modeled and controlled, as the model is not yet fully established. Also, control techniques for load hauling using multiple UAVs working in co-operation can be considered.

CHAPTER 7

APPENDIX

The components of matrix C are written as c_{ij} for $i, j = 1, 2, 3$ and are given by

$$c_{11} = 0$$

$$c_{12} = (I_{yy} - I_{zz})(\dot{\theta}C\phi S\phi + \dot{\psi}S^2\phi C\theta) + (I_{zz} - I_{yy})\dot{\psi}C^2\phi C\theta - I_{xx}\dot{\psi}C\theta$$

$$c_{13} = (I_{zz} - I_{yy})\dot{\psi}C\phi S\phi C^2\theta$$

$$c_{21} = (I_{zz} - I_{yy})(\dot{\theta}C\phi S\phi + \dot{\psi}S^2\phi C\theta) + (I_{yy} - I_{zz})\dot{\psi}C^2\phi C\theta + I_{xx}\dot{\psi}C\theta$$

$$c_{22} = (I_{zz} - I_{yy})\dot{\phi}C\phi S\phi$$

$$c_{23} = -I_{xx}\dot{\psi}S\theta C\theta + I_{yy}\dot{\psi}S^2\phi C\theta S\theta + I_{zz}\dot{\psi}C^2\phi S\theta C\theta$$

$$c_{31} = (I_{yy} - I_{zz})\dot{\psi}C\phi S\phi C^2\theta - I_{xx}\dot{\theta}C\theta$$

$$c_{32} = (I_{zz} - I_{yy})(\dot{\theta}C\phi S\phi S\theta + \dot{\psi}S^2\phi C\theta) + (I_{yy} - I_{zz})\dot{\phi}C^2\phi C\theta + \\ I_{xx}\dot{\psi}S\theta C\theta - I_{yy}\dot{\psi}S^2\phi S\theta C\theta - I_{zz}\dot{\psi}C^2\phi S\theta C\theta$$

$$c_{33} = (I_{yy} - I_{zz})\dot{\phi}C\phi S\phi C^2\theta - I_{yy}\dot{\theta}S^2\phi S\theta C\theta - I_{zz}\dot{\theta}C^2\phi S\theta C\theta + I_{xx}\dot{\theta}C\theta S\theta$$

REFERENCES

- [1] M. Abzug. Dynamics and control of helicopters with two cable sling loads. *AIAA 2nd Aircraft Design and Operations Meeting*, (American Institute of Aeronautics and Astronautics), 1970.
- [2] MAO Jr. Alves, E Nobrega, and T. Yoneyama. Adaptive neural control for a tolerant fault system. In *7th IFAC Symposium on Fault Detection, Supervision and Safety of Technical Processes.*, pages 137–142.
- [3] S. Asseo and R. Whitbeck. Control requirements for sling-load stabilization in heavy lift helicopters. *Journal of the American Helicopter Society*, Vol. 18:pp. 23–31, 1973.
- [4] C. Berbra, S. Leseq, and J. J. Martinez. A multi-observer switching strategy for fault-tolerant control of a quadrotor helicopter. In *Control and Automation, 2008 16th Mediterranean Conference on*, pages 1094–1099.
- [5] S. Bouabdallah, M. Becker, and R. Siegwart. Autonomous miniature flying robots: coming soon! - research, development, and results. *Robotics and Automation Magazine, IEEE*, 14(3):88–98, 2007.

- [6] O. Bourquardez, R. Mahony, N. Guenard, F. Chaumette, T. Hamel, and L. Eck. Image-based visual servo control of the translation kinematics of a quadrotor aerial vehicle. *Robotics, IEEE Transactions on*, 25(3):743–749, 2009.
- [7] S. Briczinski and G. Karas. Criteria for externally suspended helicopter loads. Technical report, 1971.
- [8] P. Castillo, A. Dzul, and R. Lozano. Real-time stabilization and tracking of a four-rotor mini rotorcraft. *Control Systems Technology, IEEE Transactions on*, 12(4):510–516, 2004.
- [9] J. Cera and S.W. J. Farmer. A method of automatically stabilizing helicopter sling loads. Technical report, 1974.
- [10] A. Das, K. Subbarao, and F. Lewis. Dynamic inversion with zero-dynamics stabilisation for quadrotor control. *Control Theory and Applications, IET*, 3(3):303–314, 2009.
- [11] Abhijit Das, Frank Lewis, and Kamesh Subbarao. Backstepping approach for controlling a quadrotor using lagrange form dynamics. *Journal of Intelligent and Robotic Systems*, 56(1):127–151, 2009.
- [12] L. Derafa, A. Benallegue, and L. Fridman. Super twisting control algorithm for the attitude tracking of a four rotors uav. *Journal of the Franklin Institute*, 349(2):685–699, 2012.

- [13] T. Dierks and S. Jagannathan. Output feedback control of a quadrotor uav using neural networks. *Neural Networks, IEEE Transactions on*, 21(1):50–66, 2010.
- [14] Alessandro Freddi, Alexander Lanzon, and Sauro Longhi. A feedback linearization approach to fault tolerance in quadrotor vehicles. In *Proceedings of The 2011 IFAC World Congress, Milan, Italy*, 2011.
- [15] Alessandro Freddi, Sauro Longhi, and Andrea Monteriu. A diagnostic thau observer for a class of unmanned vehicles. *Journal of Intelligent and Robotic Systems*, pages 1–13, 2012.
- [16] H. A. Izadi, B. W. Gordon, and Y. Zhang. A data-driven fault tolerant model predictive control with fault identification. In *Control and Fault-Tolerant Systems (SysTol), 2010 Conference on*, pages 732–737.
- [17] H.K. Khalil. *Nonlinear Systems*. Prentice Hall, 2002.
- [18] F.L. Lewis, S. Jagannathan, and A. Yeşildirek. *Neural Network Control Of Robot Manipulators And Non-Linear Systems*. The Taylor & Francis systems and control book series. Taylor & Francis Group, 1999.
- [19] ZiyadN. Masoud and AliH. Nayfeh. Sway reduction on container cranes using delayed feedback controller. *Nonlinear Dynamics*, 34:347–358, 2003.
- [20] C. Nicol, C. J. B. Macnab, and A. Ramirez-Serrano. Robust adaptive control of a quadrotor helicopter. *Mechatronics*, 21(6):927–938, 2011.

- [21] H. M. Omar. New fuzzy-based anti-swing controller for helicopter slung-load system near hover. In *Computational Intelligence in Robotics and Automation (CIRA), 2009 IEEE International Symposium on*, pages 474–479.
- [22] H. M. Omar. Anti-swing controller based on time-delayed feedback for helicopter slung load system near hover. *Journal of Aerospace and Technology Managment*, Vol. 4 No. 3:297–305, 2012.
- [23] H. M. Omar and A. H. Nayfeh. Anti-swing control of gantry and tower cranes using fuzzy and time-delayed feedback with friction compensation. *Shock and Vibration*, 12(2):73–89, 2005.
- [24] I. Palunko, P. Cruz, and R. Fierro. Agile load transportation : Safe and efficient load manipulation with aerial robots. *Robotics Automation Magazine, IEEE*, 19(3):69 –79, Sept. 2012.
- [25] P. Pounds, R. Mahony, and P. Corke. Modelling and control of a large quadrotor robot. *Control Engineering Practice*, 18(7):691–699, 2010.
- [26] Guilherme V. Raffo, Manuel G. Ortega, and Francisco R. Rubio. An integral predictive/nonlinear control structure for a quadrotor helicopter. *Automatica*, 46(1):29–39, 2010.
- [27] et al Raz R. Active aerodynamic stabilization of a helicopter/sling-load system. *AIAA Journal of Aircraft*,, Vol. 26:pp. 822–828., 1989.

- [28] F. Sharifi, M. Mirzaei, B. W. Gordon, and Zhang Youmin. Fault tolerant control of a quadrotor uav using sliding mode control. In *Control and Fault-Tolerant Systems (SysTol), 2010 Conference on*, pages 239–244.
- [29] J.J.E. Slotine and W.A. LI. *Applied Nonlinear Control*. Prentice Hall, 1991.
- [30] A. Tayebi. Unit quaternion-based output feedback for the attitude tracking problem. *Automatic Control, IEEE Transactions on*, 53(6):1516–1520, 2008.
- [31] A. Tayebi and S. McGilvray. Attitude stabilization of a vtol quadrotor aircraft. *Control Systems Technology, IEEE Transactions on*, 14(3):562–571, 2006.
- [32] K. Watanabe, Y. Yoshihata, Y. Iwatani, and K. Hashimoto. Image-based visual pid control of a micro helicopter using a stationary camera. In *SICE, 2007 Annual Conference*, pages 3001–3006.
- [33] J. Wolkovitch and D. Johnston. Automatic control considerations for helicopter and vtol aircraft with and without sling loads. Technical report, 1965.
- [34] A. Yesildirek and F.L. Lewis. Feedback linearization using neural networks. *Automatica*, 31(11):1659 – 1664, 1995.

Vitae

Name: Ghufran Ahmed

Nationality: Indian

Date of Birth: 23 February 1987

Email: *ghufran.ahmed@hotmail.com*

Permenant Address: H No: 5-9-848/1, Gunfoundry

Hyderabad, 500001, India.

Academic Background: Master of Science, Automation and Control

King Fahd University of Petroleum and Minerals, Dhahran

Kingdom of Saudi Arabia.