

2017

Multivariable Control of a Rolling Spider Drone

Haifeng Lyu

University of Rhode Island, haifeng@my.uri.edu

Follow this and additional works at: <https://digitalcommons.uri.edu/theses>

Recommended Citation

Lyu, Haifeng, "Multivariable Control of a Rolling Spider Drone" (2017). *Open Access Master's Theses*. Paper 1130.

<https://digitalcommons.uri.edu/theses/1130>

This Thesis is brought to you by the University of Rhode Island. It has been accepted for inclusion in Open Access Master's Theses by an authorized administrator of DigitalCommons@URI. For more information, please contact digitalcommons-group@uri.edu. For permission to reuse copyrighted content, contact the author directly.

MULTIVARIABLE CONTROL OF A ROLLING SPIDER DRONE

BY

HAIFENG LYU

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE

REQUIREMENTS FOR THE DEGREE OF

MASTER OF SCIENCE

IN

ELECTRICAL ENGINEERING

UNIVERSITY OF RHODE ISLAND

2017

MASTER OF SCIENCE THESIS
OF
HAIFENG LYU

APPROVED:

Thesis Committee:

Major Professor Richard J Vaccaro

Resit Sendag

Chengzhi Yuan

Nasser H. Zawia

DEAN OF THE GRADUATE SCHOOL

UNIVERSITY OF RHODE ISLAND

2017

ABSTRACT

The research and application of Unmanned Aerial Vehicles (UAVs) has been a hot topic recently. A UAV is defined as an aircraft which is designed not to carry a human pilot or operated with remote electronic input by the flight controller. In this thesis, the design of a control system for a quadcopter named Rolling Spider Drone is conducted. The thesis work presents the design of two kinds of controllers that can control the drone to keep it balanced and track different kinds of input trajectories. The nonlinear mathematical model for the drone is derived by the Newton-Euler method. The rotational subsystem and translational subsystem are derived to describe the attitude and position motion of drone. Techniques from linear control theory are employed to linearize the highly coupled and nonlinear quadcopter plant around equilibrium points and apply the linear feedback controller to stabilize the system. The controller is a digital tracking system that deploys LQR for system stability design. Fixed gain and adaptive gain scheduled controllers are developed and compared with different LQR weights. Step references and reference trajectories involving significant variation for the yaw angle in the xy-plane and three-dimensional space are tracked in the simulation. The physical implementation and an output feedback controller are considered for future work.

ACKNOWLEDGMENTS

I would like to acknowledge Dr. Richard J. Vaccaro for his guidance and advice in the area of control theory, for his immense knowledge and patience. It is my pleasure to study with him. I would also like to thank my family for their love, support and encouragement.

TABLE OF CONTENTS

ABSTRACT	ii
ACKNOWLEDGMENTS	iii
TABLE OF CONTENTS	iv
LIST OF FIGURES	vii
CHAPTER	
1 Introduction	1
1.1 Objectives and Motivation	1
1.2 Contribution of this Work	2
1.3 Thesis Structure	2
List of References	3
2 System Modeling	4
2.1 Kinematic Model	4
2.2 Dynamic Model	8
2.2.1 Translational Equations of Motion	8
2.2.2 Rotational Equations of Motion	9
2.3 State Space Model	11
2.3.1 State Vector	11
2.3.2 Input Vector	11
2.3.3 Aerodynamic Moment and Force	11
List of References	13

	Page
3 Control System Theory	14
3.1 Linear Control System	14
3.2 Controllability and Observability	15
3.3 Linear Quadratic Regulator	16
3.4 Linearization near an equilibrium point	18
3.5 Lyapunov Stability	19
3.6 Stability Margins For Linear Multivariable Control Systems	20
3.7 Gain-Scheduled Control	25
List of References	26
4 Tracking System Design and Simulation Analysis	27
4.1 Linearization and Controllability of Quadcopter Dynamics	27
4.2 Obtaining a Tracking System by Regulating the Design Model	29
4.2.1 Additional Dynamics Blocks	31
4.2.2 LQR Design of a Linear Tracking System	33
4.3 Design of a Fixed-Gain Tracking System	34
4.4 Gain-Scheduled LQR Control	44
4.4.1 Figure - 8 trajectory tracking	44
4.4.2 3D trajectory tracking	50
List of References	60
5 Conclusions	61
 APPENDIX	
Matlab Code	63

BIBLIOGRAPHY 69

LIST OF FIGURES

Figure		Page
1	Quadcopter Structure Configuration	5
2	Quadcopter Reference Frame and Euler Angles	6
3	Optimal State Feedback Regulator	18
4	Feedback Interconnection of Two Stable Systems	21
5	Perturbation Model with δ_1	21
6	Interconnection of Perturbation and System	22
7	Classical Stability Margins	23
8	Perturbation Model with δ_2	24
9	Digital Tracking System	29
10	Digital Tracking System	30
11	Design Model Structure	30
12	Partition \tilde{K} for the New State Variables	31
13	Redraw the System Structure with Actual Plant Model	31
14	Structure of a State-Space Tracking System	32
15	Extend the Design Model with Additional Dynamics	32
16	Additional Dynamics and Integrators	34
17	Tracking System for the Nonlinear Drone	35
18	Step Reference Tracking X and Y states	36
19	Step Reference Tracking Z and Yaw states	36
20	Plant Input Angular Speed	37
21	Step Reference Tracking X and Y states	38

Figure		Page
22	Step Reference Tracking Z and Yaw states	39
23	Plant Input Angular Speed	39
24	Step Reference Tracking X and Y States	40
25	Step Reference Tracking Z and Yaw States	40
26	Plant Input Angular Speed	41
27	Step Reference Tracking X and Y States	42
28	Step Reference Tracking Z and Yaw States	43
29	Plant Input Angular Speed	43
30	Gain Scheduling Controller Model	46
31	Steady State Figure Eight Tracking xy-plane $\Delta = \pi/3$	47
32	Steady State Figure Eight Tracking xy-plane $\Delta = \pi/12$	47
33	Steady State Figure Eight Tracking xy-plane $\Delta = \pi/30$	48
34	Figure Eight Tracking with Penalties $\Delta = \pi/3$	48
35	Figure Eight Tracking with Penalties $\Delta = \pi/12$	49
36	Figure Eight Tracking with Penalties $\Delta = \pi/30$	49
37	3D Helix Trajectory Tracking	51
38	Plant Input Angular Speed	52
39	Helix Trajectory Tracking xy-plane	52
40	Roll Angle Variation	53
41	Yaw Angle Trajectory Tracking	53
42	Z Direction Trajectory Tracking	54
43	Y Direction Trajectory Tracking	54
44	X Direction Trajectory Tracking	55

Figure		Page
45	3D Lemniscate Trajectory Tracking	56
46	Plant Input Angular Speed	56
47	Lemniscate Trajectory Tracking xy-plane	57
48	Roll Angle Variation	57
49	Yaw Angle Trajectory Tracking	58
50	Z Direction Trajectory Tracking	58
51	Y Direction Trajectory Tracking	59
52	X Direction Trajectory Tracking	59

CHAPTER 1

Introduction

The quadcopter is one of many types of unmanned aerial vehicles (UAVs). Because of its reliability of maneuvering and the ability to be controlled and modeled, the research about quadcopter is becoming a hot topic in the area of robotics control.

1.1 Objectives and Motivation

This thesis work deals with the design of a control system for a specific quadcopter which is called a Rolling Spider Drone [1]. It can take off and land vertically and hover and fly with high maneuverability. Also, one of the most important advantages of the Rolling Spider Drone is that various control algorithms can be downloaded into the Drone's controller via firmware to test the actual flight performance.

The challenge in control system design for a quadcopter is that the mathematical model of the system is highly coupled, nonlinear and multivariable[2]. The quadcopter dynamics contain 6 degrees of freedom, which are 3 rotational degrees, (roll, pitch and yaw), and also translation along the x, y, z axes. Because the number of control inputs are less than the system's degrees of freedom, it is an under-actuated system. The Rolling Spider Drone moves in a three dimensional space. Optimal control theory and gain scheduling strategy will be employed in the design of controller.

Currently, the Control System Laboratory in ECBE Department at the University of Rhode Island does not have any hardware for flight control experiments. The purpose of this thesis is to use the commercially available Rolling Spider

Drone and Matlab/Simulink software to begin the development of a flight control testbed for multivariable control algorithms [3]. The specific goals are as follows:

1. Derive the theoretical state-space model for the drone.
2. Develop linear multivariable control algorithm based on linearized models using optimal control technique [4]. Evaluate the controller performance in simulation.
3. Extend the single controller to a set of controllers designed at a set of equilibrium points and evaluate a gain-scheduled control system for reference trajectories that require the use of multiple linear models.

1.2 Contribution of this Work

There are various control techniques that are used in order to provide stable position and orientation of quadcopter. Reference [5] presents a PID tuned LQR controller with good robustness and easy implementation, however, only simulation is provided, and the actual applicability of the controllers is not proposed. Reference [6] presents a gain scheduling technique to design a controller for nonlinear system by linearizing the system at several equilibrium points. They use LQR controller to verify the controllability and observability. However, this paper did not provide closed-loop control for the trajectory tracking. The design of this system will be based on LQR controller design for linear models and use gain scheduling technique to deal with the nonlinear model of Drone. A digital tracking system [7] will be designed that would be suitable for implementation on the Rolling Spider drone.

1.3 Thesis Structure

This thesis is organized into five Chapters. Chapter 2 shows the mathematical model set up of a quadcopter using the Newton-Euler theorem, gives the details

about the kinematic and dynamic model, and obtains the nonlinear state-space model. Chapter 3 presents the control theory that is employed in the thesis. It covers the linear and nonlinear control theory, stability robustness and margins, linear quadratic regulator and gain-scheduled control. Chapter 4 demonstrates the techniques used to develop a linear feedback controller to regulate the nonlinear model. The controller is verified using the simulation. Chapter 5 shows the conclusions and future work for the thesis.

List of References

- [1] Parrot SA. “Parrot minidrones rolling spider.” June 2008. [Online]. Available: <http://global.parrot.com/au/products/rolling-spider>
- [2] A. Jafar, S. Ahmad, and N. Ahmed, “Mathematical modeling and control law design for 1 dof quadcopter flight dynamics,” in *International Conference on Computing, Electronic and Electrical Engineering (ICE Cube)*, 2016, pp. 79–84.
- [3] Massachusetts Institute of Technology. “Matlab toolbox for parrot rolling spider.” [Online]. Available: <http://karaman.mit.edu/software.html>
- [4] E. Lavretsky and K. A. Wise, *Robust and Adaptive Control with Aerospace Applications*. Springer, Inc, 2013.
- [5] L. M. Argentim, W. C. Rezende, P. E. Santos, and R. A. Aguiar, “PID, LQR and LQR-PID on a Quadcopter Platform,” in *Proceedings IEEE Informatics, Electronics and Vision (ICIEV)*, 2013, pp. 1–6.
- [6] A. Ataka, H. Tnunay, and R. Inovan, “Controllability and observability analysis of the gain scheduling based linearization for uav quadrotor,” in *International Conference on Robotics, Biomimetics, Intelligent Computational Systems (RO-BIONETICS)*, 2013.
- [7] R.J.Vaccaro, *Digital Control A State-Space Approach*. McGraw-Hill, Inc, 1996.

CHAPTER 2

System Modeling

In this chapter the flight dynamics model will be derived. The ordinary quadcopter has a symmetrical design. It consists of four rotors fixed at an equal distance from the body central as shown in Fig.1. Each of the rotors are driven by a DC motor. Propellers 1 and 3 rotate in the same direction and propellers 2 and 4 rotate in the opposite direction to keep the system balance [1].

2.1 Kinematic Model

In order to set up the model of the quadcopter, the reference coordinate frames should be defined. The earth inertial reference frame is with N, E, D axes and A, B, C body frame is with x, y, z axes. As shown in Fig.2, the inertial reference is fixed on a specific place and it uses the $N - E - D$ notation which represents North, East and Downwards respectively. For the body frame, the origin located in the center of the quadcopter body with x-axis pointing towards propeller 1, y-axis pointing towards propeller 2 and the z-axis is pointing to the ground. It is known that the quadcopter is a 6 DOF object. The three Euler angles ϕ, θ, ψ represent the orientation of the quadcopter, where ϕ is roll angle about the x-axis, θ is pitch angle about the y-axis, ψ is yaw angle about z-axis. The Euler-angles are important for the definition of the transformation matrix from different reference frames.

The transformation matrix is formed by a sequence of three plane rotations, which are named $R_{yaw}, R_{pitch}, R_{roll}$. There are two intermediate coordinate systems being defined, the vehicle-1 frame and vehicle-2 frame; to describe the orientation of the quadcopter.

The inertial frame is rotated about its z-axis by the yaw angle ψ to get the

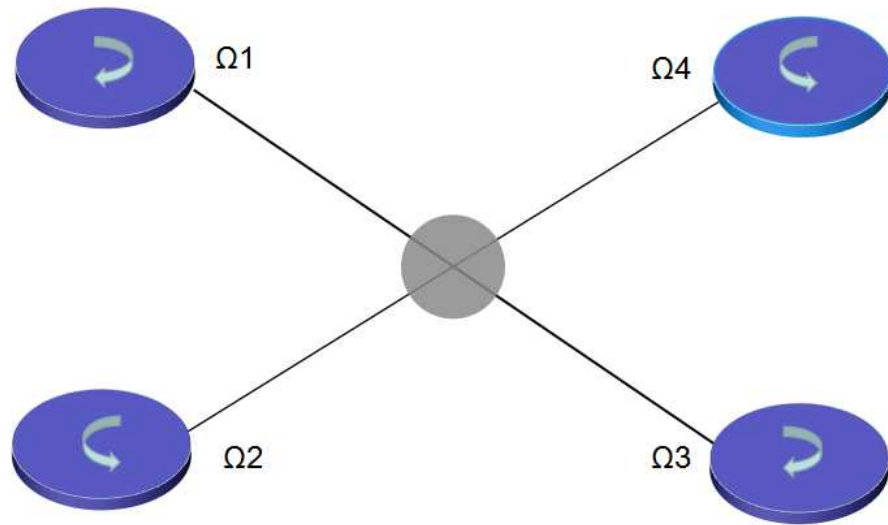


Figure 1. Quadcopter Structure Configuration

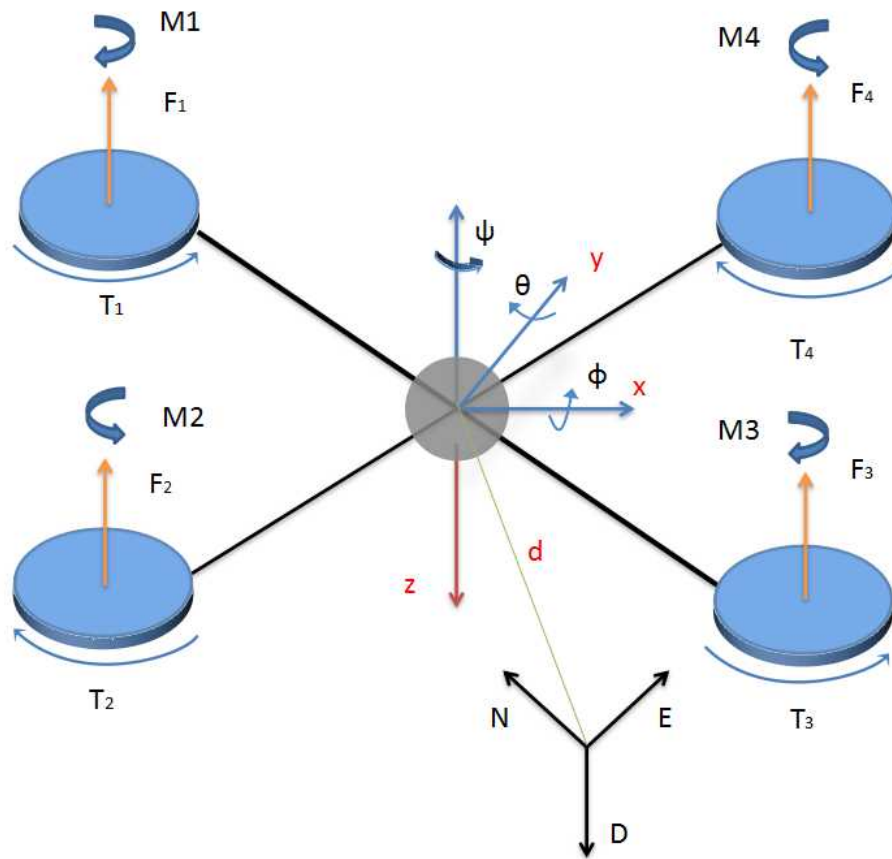


Figure 2. Quadcopter Reference Frame and Euler Angles

vehicle-1 frame. The rotation matrix is

$$R_{yaw} = \begin{bmatrix} \cos(\psi) & \sin(\psi) & 0 \\ -\sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (1)$$

The resulting frame is then rotated by the pitch angle θ around its y-axis to produce the vehicle-2 frame. This rotation matrix is

$$R_{pitch} = \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix}. \quad (2)$$

The last rotation from vehicle-2 to the body frame is

$$R_{roll} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & \sin(\phi) \\ 0 & -\sin(\phi) & \cos(\phi) \end{bmatrix}. \quad (3)$$

Finally, we get the rotation matrix from the NED world inertial frame to the body frame, which is

$$R_{W2B} = R_{roll} \cdot R_{pitch} \cdot R_{yaw}. \quad (4)$$

Multiplying out the three individual rotation matrices yields

$$\begin{aligned} R_{W2B} &= \begin{bmatrix} \cos(\psi) & -\sin(\psi) & 0 \\ \sin(\psi) & \cos(\psi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix} \\ &= \begin{bmatrix} c(\theta)c(\psi) & c(\theta)s(\phi) & -s(\theta) \\ c(\psi)s(\theta)s(\phi) - c(\phi)s(\psi) & c(\phi)c(\psi) + s(\theta)s(\phi)s(\psi) & c(\theta)s(\phi) \\ s(\psi)s(\phi) + c(\psi)c(\phi)s(\theta) & c(\phi)s(\psi)s(\theta) - c(\psi)s(\phi) & c(\theta)c(\phi) \end{bmatrix}, \end{aligned} \quad (5)$$

where c and s denote \cos and \sin respectively. Also the rotation matrix that transforms from the body frame to the inertial world frame, R_{B2W} is obtained by inverting R_{W2B} . The result can be shown to be

$$R_{B2W} = \begin{bmatrix} c(\theta)c(\psi) & c(\psi)s(\theta)s(\phi) - c(\phi)s(\psi) & s(\phi)s(\psi) + c(\phi)c(\psi)s(\theta) \\ c(\theta)s(\psi) & c(\phi)c(\psi) + s(\theta)s(\phi)s(\psi) & c(\phi)s(\theta)s(\psi) - c(\psi)s(\phi) \\ -s(\theta) & c(\theta)s(\phi) & c(\theta)c(\phi) \end{bmatrix}. \quad (6)$$

These rotation matrices will be used to convert the states measured in one frame to another frame. In addition to transforming angular positions, it is also possible to transform angular velocities. There is an inverted Wronskian matrix W^{-1} that can transform the angular body rates $\omega = (p, q, r)^T$ to the Euler rates $\dot{\Theta} = (\dot{\phi}, \dot{\theta}, \dot{\psi})^T$ that are measured in the inertial frame. The transformation is described as follows,

$$\dot{\Theta} = W^{-1} \cdot \omega , \quad (7)$$

where W^{-1} is given by

$$W^{-1} = \begin{bmatrix} c(\theta) & s(\phi)s(\theta) & c(\phi)s(\theta) \\ 0 & c(\phi)c(\theta) & -s(\phi)c(\theta) \\ 0 & s(\phi) & c(\phi) \end{bmatrix} \cdot \frac{1}{c(\theta)} . \quad (8)$$

2.2 Dynamic Model

The motion of the quadcopter can be divided into two subsystems, translational motion (x,y positions and altitude), and rotational motion (roll, pitch and yaw)[2]. Based on the Newton-Euler formalism, the forces and moments acting on the quadcopter will be investigated.

The twelve states variables are as follows: three translational positions, their corresponding velocities and three angular positions and their corresponding velocities. The position vector in an earth-fixed inertial coordinate system is denoted as $\vec{P} = [x, y, z]^T$, its velocity rate vector in the inertial system is $\vec{v} = [\dot{x}, \dot{y}, \dot{z}]^T$. The roll, pitch, yaw angles are denoted as $\vec{\Theta} = [\phi, \theta, \psi]^T$ and their derivatives are body angular rates $\vec{\omega} = [p, q, r]^T$. The position vector of the center of mass in the world frame is denoted by d .

2.2.1 Translational Equations of Motion

Assume that the structure of quadcopter is rigid and symmetrical about the center of gravity of the quadcopter, and the propellers are also rigid. Based on the

Newton's second law,

$$\vec{F}_w = \frac{d}{dt}(m \cdot \vec{v}) \quad (9)$$

where $\vec{F}_w = \vec{G} - \vec{T}$. G is the gravity and T is the thrust generated from rotors.

Note that the \vec{T} is described in the body frame, and must be transformed from body frame to inertial frame. Then equation 9 can be expressed as

$$\dot{v} = \frac{1}{m} \left[\begin{array}{c} 0 \\ 0 \\ mg \end{array} \right] - R_{B2W} \cdot \left[\begin{array}{c} 0 \\ 0 \\ T \end{array} \right] \quad (10)$$

Substituting R_{B2W} into this equation gives an expression for the derivative of velocity rate in the inertial frame.

$$\dot{v} = \frac{1}{m} \left[\begin{array}{c} -T(c(\phi)s(\theta)c(\psi) + s(\phi)s(\psi)) \\ -T(c(\phi)s(\theta)s(\psi) - s(\phi)c(\psi)) \\ mg - T(c(\phi)c(\theta)) \end{array} \right] \quad (11)$$

This vector equation can be written in component form as

$$\begin{aligned} \ddot{x} &= -\frac{1}{m}T(c(\phi)s(\theta)c(\psi) + s(\phi)s(\psi)) \\ \ddot{y} &= -\frac{1}{m}T(c(\phi)s(\theta)s(\psi) - s(\phi)c(\psi)) \\ \ddot{z} &= -\frac{1}{m}Tc(\phi)c(\theta) + g \end{aligned} \quad (12)$$

2.2.2 Rotational Equations of Motion

In the inertial frame, it is known that the moment is defined as the time derivative of the angular momentum, using rigid body rotational law, we have

$$\vec{M}_w = \frac{d\vec{L}}{dt} = \frac{d}{dt}(\vec{I} \cdot \vec{\omega}) \quad (13)$$

where \vec{I} is the body's inertia tensor

$$\vec{I} = \begin{bmatrix} I_{xx} & I_{xy} & I_{xz} \\ I_{yx} & I_{yy} & I_{yz} \\ I_{zx} & I_{zy} & I_{zz} \end{bmatrix} = \begin{bmatrix} I_{xx} & 0 & 0 \\ 0 & I_{yy} & 0 \\ 0 & 0 & I_{zz} \end{bmatrix}. \quad (14)$$

Note that the off-diagonal terms are equal to zero due to the symmetric mass distribution of the quadcopter. In order to make the understanding and calculations more intuitive, it is better to set up the expression in the body frame using Euler's equation,

$$\vec{M}_b = \vec{I} \cdot \dot{\vec{\omega}} + \vec{\omega} \times (\vec{I} \cdot \vec{\omega}) = \begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} \quad (15)$$

where the right-hand side is a vector of applied torques.

$$\vec{\omega} \times (\vec{I} \cdot \vec{\omega}) = \begin{bmatrix} p \\ q \\ r \end{bmatrix} \times \begin{bmatrix} I_{xx} p \\ I_{yy} q \\ I_{zz} r \end{bmatrix} = \begin{bmatrix} (I_{zz} - I_{yy}) qr \\ (I_{yy} - I_{zz}) rp \\ (I_{yy} - I_{xx}) pq \end{bmatrix} \quad (16)$$

substituting this into equation 15 yields

$$\begin{aligned} \tau_x &= I_{xx} \dot{p} + (I_{zz} - I_{yy}) qr \\ \tau_y &= I_{yy} \dot{q} + (I_{xx} - I_{zz}) rp \\ \tau_z &= I_{zz} \dot{r} + (I_{yy} - I_{xx}) pq \end{aligned} \quad (17)$$

These equations can be solved for the derivatives of the angular rates to obtain

$$\begin{aligned} \dot{p} &= (\tau_x + I_{yy}qr - I_{zz}qr)/I_{xx} \\ \dot{q} &= (\tau_y - I_{xx}pr + I_{zz}pr)/I_{yy} \\ \dot{r} &= (\tau_z + I_{xx}pq - I_{yy}pq)/I_{zz} \end{aligned} \quad (18)$$

Using 7 and 8, the derivatives of the Euler angles are obtained by multiplying by W^{-1} to get

$$\begin{aligned} \dot{\phi} &= p + (rc(\phi)s(\theta))/c(\theta) + (qs(\theta)s(\phi))/c(\theta) \\ \dot{\theta} &= qc(\phi) - rs(\phi) \\ \dot{\psi} &= (rc(\phi))/c(\theta) + (qs(\phi))/c(\theta) \end{aligned} \quad (19)$$

2.3 State Space Model

2.3.1 State Vector

The state vector of the quadcopter is defined to be,

$$\begin{aligned} X &= [x_1 \ x_2 \ x_3 \ x_4 \ x_5 \ x_6 \ x_7 x_8 \ x_9 \ x_{10} \ x_{11} \ x_{12}]^T. \\ &= [x \ y \ z \ \phi \ \theta \ \psi \ \dot{x} \ \dot{y} \ \dot{z} \ p \ q \ r]^T. \end{aligned} \tag{20}$$

Note that the translational positions and velocities are defined in the inertial frame while the $p \ q \ r$ are defined in the body frame.

2.3.2 Input Vector

The vector input U consisting of four inputs, is defined as,

$$U = [U_1 \ U_2 \ U_3 \ U_4] = [\omega_1^2 \ \omega_2^2 \ \omega_3^2 \ \omega_4^2]. \tag{21}$$

that is, the inputs to the quadcopter system are the squares of the motor speeds driving the four propellers.

2.3.3 Aerodynamic Moment and Force

As an effect of the propeller rotation, the lift and translational force called the aerodynamic force generated. The direction of the force is always aligned with the body frame z-axis. The thrust force is generated from rotors $i = 1, 2, 3, 4$ is proportional to the square of the motor speed

$$F_i = K_a \omega_i^2. \tag{22}$$

The rotors also generate the aerodynamic moment, which is proportional to the square of the rotor speed,

$$M_i = K_m \omega_i^2. \tag{23}$$

For the low-altitude flight of quadcopters, air density can be considered as a constant number, and K_a and K_m in equation 22 and 23 are constants respectively.

Consider now the moments and thrust acting on the quadcopter. Fig.2 shows the forces and moments acting on the quadcopter: the upwards thrust force F_i and moment M_i come from the rotation of the rotors.

We define l as the lever length of each of the quadrotor's arms, based on the right-hand-rule, as Fig.2 presents, F_2 multiplied by the moment arm l generates a negative moment about the y-axis, while in the same manner, F_4 generates a positive moment. Thus the total moment about the x-axis is,

$$\begin{aligned}
\tau_x &= -F_2l + F_4l \\
&= -(K_a\omega_2^2)l + (K_a\omega_4^2)l \\
&= K_al(\omega_4^2 - \omega_2^2)
\end{aligned} \tag{24}$$

For the moments about the body frame's y-axis, the thrust of rotor 1 generate a positive moment, while the thrust of rotor 3 generates a negative moment about the y-axis, the total moment can be expressed as,

$$\begin{aligned}
\tau_y &= F_1l - F_3l \\
&= (K_a\omega_1^2)l - (K_a\omega_3^2)l \\
&= K_al(\omega_1^2 - \omega_3^2)
\end{aligned} \tag{25}$$

The moment about the body frame's z-axis is caused by the rotors' rotation from aerodynamic moment, while the thrust force has no effect on the moment for z-axis.

$$\begin{aligned}
\tau_z &= M_1 - M_2 + M_3 - M_4 \\
&= (K_m\omega_1^2) - (K_m\omega_2^2) + K_m\omega_3^2 - (K_m\omega_4^2) \\
&= K_m(\omega_1^2 - \omega_2^2 + \omega_3^2 - \omega_4^2)
\end{aligned} \tag{26}$$

The thrust is given by

$$T = -K_a(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2) \quad (27)$$

In the body frame, we define the positive orientation is pointing downwards, so the thrust forces acting on the z axis produced by the rotation of four propellers is in the negative direction, while there are no forces acting on the x and y directions. The non-linear state-space equation $\dot{X} = f(X, U)$ can be written as,

$$\begin{aligned} f_1 = \dot{x} &= x_7 \\ f_2 = \dot{y} &= x_8 \\ f_3 = \dot{z} &= x_9 \\ f_4 = \dot{\phi} &= p + (rc(\phi)s(\theta))/c(\theta) + (qs(\theta)s(\phi))/c(\theta) \\ f_5 = \dot{\theta} &= qc(\phi) - rs(\phi) \\ f_6 = \dot{\psi} &= (rc(\phi))/c(\theta) + (qs(\phi))/c(\theta) \\ f_7 = \ddot{x} &= -\frac{1}{m}T(c(\phi)s(\theta)c(\psi) + s(\phi)s(\psi)) \\ f_8 = \ddot{y} &= -\frac{1}{m}T(c(\phi)s(\theta)s(\psi) - s(\phi)c(\psi)) \\ f_9 = \ddot{z} &= -\frac{1}{m}Tc(\phi)c(\theta) + g \\ f_{10} = \dot{p} &= (\tau_x + I_{yy}qr - I_{zz}qr)/I_{xx} \\ f_{11} = \dot{q} &= (\tau_y - I_{xx}pr + I_{xx}pr)/I_{yy} \\ f_{12} = \dot{r} &= (\tau_z + I_{xx}pq - I_{yy}pq)/I_{zz} \end{aligned} \quad (28)$$

List of References

- [1] K. Patel and J. Varve, "Modeling, simulation and control study for the quadcopter UAV," in *International Conference on Industrial and Information Systems (ICIIS)*, 2014, pp. 1–6.
- [2] H. ElKholy, "Dynamic modeling and control of a quadrotor using linear and nonlinear approaches," Master's thesis, The American University in Cairo, 2014.

CHAPTER 3

Control System Theory

The goal of this chapter is to design a digital tracking system for the quadcopter. Although this plant is a nonlinear system, we can still use results from linear control theory. As such, before we develop the various controllers, we first go over the basic knowledge of linear control theory.

3.1 Linear Control System

A linear time invariant (*LTI*) system may be represented by a state-space model of the form

$$\begin{aligned}\dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) + Du(t),\end{aligned}\tag{29}$$

where $x(t) \in \mathfrak{R}^n$, $y(t) \in \mathfrak{R}^p$, $u(t) \in \mathfrak{R}^m$ are time-dependent vectors, while $A \in R^{n \times n}$, $B \in R^{n \times m}$, $C \in R^{p \times n}$, $D \in R^{p \times m}$ are constant matrices.

Since the quadcopter is a multi-input multi-output (*MIMO*) system which has 12 state variables, 4 inputs and 4 outputs, after linearization, the matrix A is 12×12 , B is 12×4 , C is 4×12 and D is 4×4 . Because we want to use digital control, we let the plant-input signals be generated through D/A converters (zero-order hold) with sampling interval T seconds. Also, the plant outputs and state variables are sampled by A/D converters. The *LTI* continuous-time system in (29), when surrounded by D/A and A/D converters, is equivalent to the following discrete-time state-space model,

$$\begin{aligned}x[k+1] &= \Phi x[k] + \Gamma u[k] \\ y[k] &= Cx[k] + Du[k],\end{aligned}\tag{30}$$

where $x[k] = x(kT)$, $y[k] = y(kT)$, $u(t) = u[k]$ for $kT \leq t < (k+1)T$, $\Phi = e^{AT}$ and $\Gamma = \int e^{A\tau} B d\tau$. T is the sampling interval and k is the time value of the discrete-time sequences.

3.2 Controllability and Observability

Consider a continuous linear system. If it is possible to find an input $u(t)$, $t_0 \leq t \leq t_1$ to drive the system from arbitrary initial state $x(t_0) = x_0$ to an arbitrary final state $x(t_1) = x_f$ with finite time, the system is said to be controllable. The controllability matrix W_c can be defined as

$$W_c = [B \ AB \ A^2B \ \dots \ A^{n-1}B]. \quad (31)$$

The pair (A, B) is controllable if and only if the rank of (A, B) is equal to n , the size of A matrix.

Consider a digital control system $x[k+1] = \Phi x[k] + \Gamma u[k]$. If it is possible to find an input sequence $u[k]$ that takes the system from an arbitrary initial state $x[0]$ to any final state $x[j]$ with finite step j , then the digital system is said to be controllable.

The controllability matrix W_C for the digital system is,

$$W_C = [\Gamma \ \Phi\Gamma \ \Phi^2\Gamma \ \dots \ \Phi^{n-1}\Gamma]. \quad (32)$$

According to [1], the controllability of the *ZOH* equivalent system (Φ, Γ) obtained from a continuous-time system (A, B) can be determined with two criteria. First, the continuous-time system (A, B) must be controllable. Second, if the imaginary part of the poles with largest magnitude β_{max} satisfies

$$T < \frac{\pi}{\beta_{max}} \quad (33)$$

then (Φ, Γ) will be controllable.

The property of observability can be defined for a continuous-time LTI system,

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx, \end{aligned} \tag{34}$$

which is a simplified form of (29). This system is said to be observable, if the initial state x_{t_0} can be calculated using the inputs and outputs where $t_0 \leq t \leq t_1$, with finite time $t_1 - t_0$. We define the observability matrix W_o as

$$W_o = \begin{bmatrix} C \\ AC \\ A^2 C \\ \dots \\ A^{n-1} C \end{bmatrix}. \tag{35}$$

The pair (A, C) is observable if and only if the rank of W_o is equal to n . Two additional properties as follows: (A, B) is to be stabilizable if its uncontrollable poles are stable and (A, C) is to be detectable if its unobservable poles are stable.

Consider a linear discrete-time system $x[k+1] = \Phi x[k]$, $x[0] = x_0$ with output measurements $y[k] = C x[k]$. If we know the information about x_0 , then the system dynamics $x[k+1] = \Phi x[k]$ will give complete knowledge about the state variables at any discrete-time instant. While the initial conditions x_0 is determined if and only if the digital observability matrix W_O (36) has rank n .

$$W_O = \begin{bmatrix} C \\ \Phi C \\ \Phi^2 C \\ \dots \\ \Phi^{n-1} C \end{bmatrix}. \tag{36}$$

3.3 Linear Quadratic Regulator

The desired control system for the quadcopter is a digital tracking system in which reference trajectories are supplied for desired output variables. However, it

will be seen that the design of a tracking system is accomplished by solving an associated regulator design problem. For ease of presentation, we first consider regulator design in continuous time.

The notion of linear quadratic regulator (LQR) comes from the *MIMO* control system design [2]. Since the feedback gain is not unique for multi-input systems, optimal control solutions provide the best possible feedback gains for *MIMO* system. Linear quadratic optimal control can be realized by specifying some kind of performance objective function to be optimized. Consider that we have the continuous linear system,

$$\dot{x} = Ax + Bu \quad (37)$$

where the (A, B) is controllable. We would like to design a state feedback control $u = -Kx$ such that the cost function,

$$J = \int (x^T Qx + u^T Ru) dt, \quad x(t_0) = x_0 \quad (38)$$

is minimized and drives state x to 0. $Q \in \mathfrak{R}^{n \times n}, R \in \mathfrak{R}^{m \times m}$ are positive definite symmetric matrices chosen by the designer.

For the infinite time LQR problems, assuming that (A, B) are constant matrix, (A, B) is stabilizable and $(A, Q^{\frac{1}{2}})$ is detectable. In this case, the solution that minimizes cost function is the feedback control function

$$u = -Kx \quad (39)$$

with the feedback gain matrix

$$K = R^{-1} B^T P. \quad (40)$$

The matrix P is the unique symmetric positive definite is a constant matrix that solves the Control Algebraic Ricatti Equation (CARE)

$$A^T P + PA - PBR^{-1}B^T P + Q = 0. \quad (41)$$

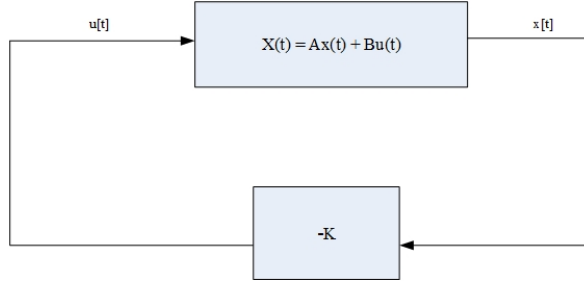


Figure 3. Optimal State Feedback Regulator

then the optimal control input is

$$u = -R^{-1}B^T Px = -Kx \quad (42)$$

where K is called Kalman gain matrix. The “optimal” state feedback regulator can be drawn as Fig.3.

3.4 Linearization near an equilibrium point

In order to use linear state feedback to regulate a nonlinear dynamic model at an equilibrium point, we need to linearize the nonlinear system around this point. According the Lyapunov stability theory, we can guarantee the system to be locally asymptotically stable around the equilibrium point and also we can find the Region of Attraction for the equilibrium point[3]. Suppose we have a nonlinear system,

$$\dot{x} = f(x, u) \quad (43)$$

with an operating point (x_{eq}, u_{eq}) and $f(x, u)$ continuously differentiable in a domain containing the operating points such that $f(x_{eq}, u_{eq}) = 0$. Expand the nonlinear function in a Taylor series about the operating point. Then, for x near x_{eq} and u near u_{eq} we define the actual plant state vector x as the constant equilibrium state vector x_{eq} plus a vector of deviations w ,

$$x = x_{eq} + w. \quad (44)$$

Also the actual plant input vector u is the constant equilibrium plant input vector u_{eq} plus a vector of deviations v ,

$$u = u_{eq} + v. \quad (45)$$

then the behavior of the nonlinear system in (43) can be approximated by a linear state-space model for the deviations as follows,

$$\begin{aligned} \dot{w} &= Aw + Bv \\ z &= Cw, \end{aligned} \quad (46)$$

where z is the deviation from the equilibrium output $z = y - Cx_{eq}$. The nonlinear system looks like a linear system around the equilibrium point where

$$\begin{aligned} A &= \left. \frac{\partial f}{\partial x}(x, u) \right|_{(x,u)=(x_{eq},u_{eq})} \\ B &= \left. \frac{\partial f}{\partial u}(x, u) \right|_{(x,u)=(x_{eq},u_{eq})}. \end{aligned} \quad (47)$$

Assume that the pair (A, B) is controllable, then we can develop a linear feedback control $u = -Kx$ to regulate the closed loop nonlinear system.

$$\dot{x} = f(x, -K(x - w)x + v). \quad (48)$$

3.5 Lyapunov Stability

In the sense of Lyapunov stability, an equilibrium point is to be asymptotically stable if it is both stable and convergent, that means for all ε there exist a $\delta > 0$ satisfy $\|x(0) - x_{eq}\| < \delta \Rightarrow \|x(t) - x_{eq}\| < \varepsilon$ and $\lim_{t \rightarrow \infty} x(t) = x_{eq}$ for all $t \geq 0$. For a particular physical nonlinear system we would deploy a scalar-valued positive definite function which is Lyapunov function to evaluate the stability of an equilibrium point.

Consider a n variables scalar-valued function $V(x)$, which is a positive definite function on some domain D containing x_{eq} with $V(x_{eq}) = 0$. If $\dot{V}(x)$ is negative semi-definite in the domain D then x_{eq} is stable. For the quadcopter system, we note that the origin $x = 0$ is an equilibrium point of the closed-loop system, since closed-loop linearized system $A - BK$ is stable (Hurwitz). It follows that, for the nonlinear system, x_{eq} is asymptotically stable. A set $M \in \mathfrak{R}^n$ (n dimensional state space) is said to be an invariant set if $x(0) \in M \Rightarrow x(t) \in M \forall t \geq 0$. In our control laws, the feedback matrix was computed by the linearized model of the quadcopter. The limitations for this method is that the feedback matrix is useful in a neighborhood of the equilibrium point. As for how large the valid neighborhood is, we will use Region of Attraction (RoA) to define the area. The definition of Region of Attraction can help us find the initial states for which $x(t) \rightarrow x_{eq}$. Given a linearized point, from Lyapunov stability theory, we may be able to obtain a conservative estimate for the RoA from the invariant sets.

3.6 Stability Margins For Linear Multivariable Control Systems

Consider the feedback interconnection of two stable systems as shown Fig.4. According to the small-gain theorem, if

$$\|H_1(s)\|_\infty \|H_2(s)\|_\infty < 1.$$

then the closed-loop system is stable, where the norm is defined as $\|H(s)\|_\infty = \max_{\omega} \bar{\theta}(H(j\omega))$, called the system infinity norm, and $\bar{\theta}(M)$ is the maximum singular value of the matrix M . Consider the following system Fig.5 with multiplicative plant perturbation Δ . The inputs and outputs of Δ are v and w respectively, when $\Delta = 0$, the system is equivalent to the nominal plant model system. Based on the small gain theorem, we can find a model from w to v with r set to zero. As

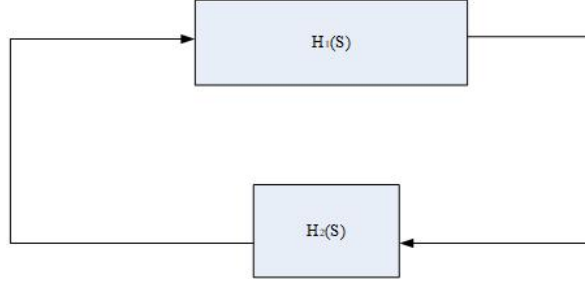


Figure 4. Feedback Interconnection of Two Stable Systems

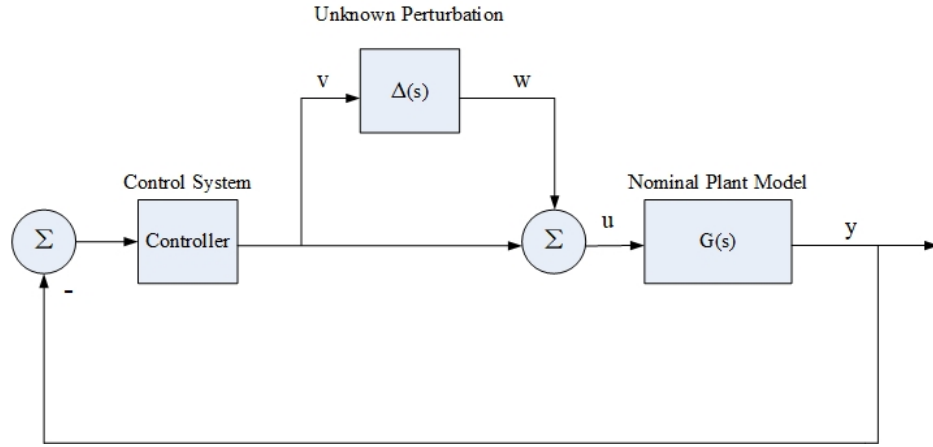


Figure 5. Perturbation Model with δ_1

the Fig.6 shows, the system $H(s)$ will be stable if,

$$\|\Delta(s)\|_{\infty} < \frac{1}{\|H(s)\|_{\infty}}. \quad (49)$$

Then the robustness bound for this closed loop system can be defined by

$$\delta_1 = \frac{1}{\|H(s)\|_{\infty}} \quad (50)$$

where $H(s)$ is the system form w to v which can be calculated using the system infinity norm of (A, B, C, D) . The value of the δ_1 implies how large a perturbation can be made to the plant model before the closed-loop system goes unstable. Now, suppose that we have the perturbation model that is a diagonal matrix of unknown complex numbers,

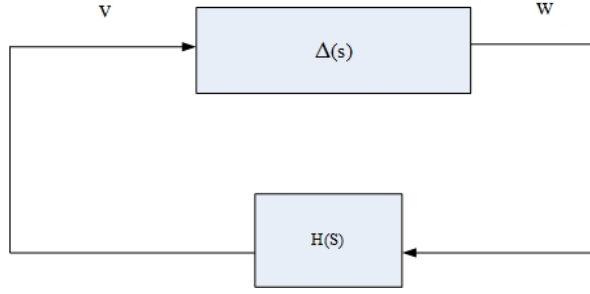


Figure 6. Interconnection of Perturbation and System

$$\Delta(s) = \begin{bmatrix} c_1 & & 0 \\ & \ddots & \\ 0 & & c_m \end{bmatrix} \quad (51)$$

where m is the number of plant inputs, c_i are the complex numbers.

We know that this closed-loop system remains stable for all $\Delta(s)$ with $\|\Delta(s)\|_\infty < \delta_1$. The maximum singular value of a diagonal matrix is simply the largest absolute value of the diagonal elements,

$$\|\Delta(s)\|_\infty = \max_i |c_i|. \quad (52)$$

So we have the conclusion that the control system will be stable if $\max |c_i| < \delta_1$. This analysis allows us to make a connection between the stability robustness bound δ_1 and classical stability margins (gain and phase margins).

Suppose we set up a model for the plant uncertainty with a diagonal matrix of complex numbers, which is the classical perturbation model as Fig.7

where

$$Q = \begin{bmatrix} q_1 & & 0 \\ & \ddots & \\ 0 & & q_m \end{bmatrix}. \quad (53)$$

Assuming δ_1 has been calculated for a given control system and $\Delta(s)$ is assumed to be a complex number c_i , so the stability conditions on corresponding classical gain margins can be found by using $Q = I + \Delta$ or $q_i = 1 + c_i$. If q_i is a real number,

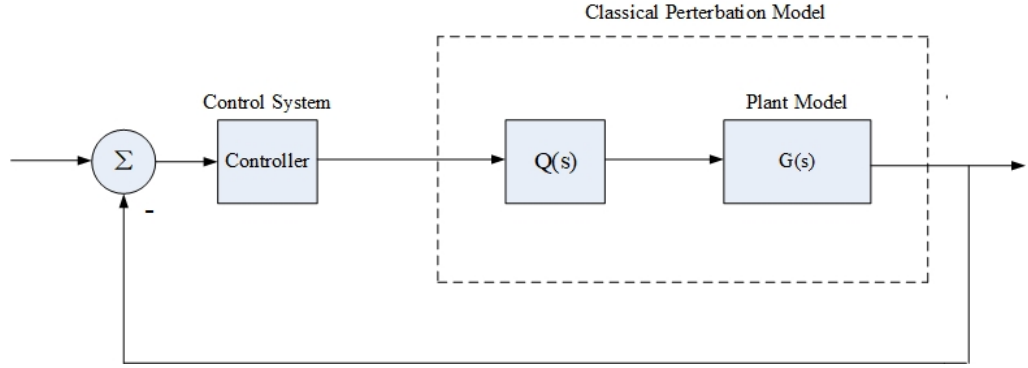


Figure 7. Classical Stability Margins

the system stable condition is

$$\begin{aligned}
 q_{max} &= 1 + \delta_1 \\
 q_{min} &= 1 - \delta_1 \\
 1 - \delta_1 &< q_i < 1 + \delta_1,
 \end{aligned} \tag{54}$$

Then the classical upper gain margin (*UGM*) and lower gain margin (*LGM*) are bounded by

$$\begin{aligned}
 UGM &\geq 20 \log_{10}(1 + \delta_i) \text{ dB} \\
 LGM &\leq 20 \log_{10}(1 - \delta_i) \text{ dB}.
 \end{aligned} \tag{55}$$

If q_i is a complex number, that is $q = e^{-j\phi} = 1 + c_i$, where 1 is the center of the circle and c_i is a disk with the radius δ_1 , we can find the classical phase margin (PM) to be bounded by,

$$\phi_{max} \geq 2 \sin^{-1}\left(\frac{\delta_1}{2}\right). \tag{56}$$

If the system perturbation plant model with following structure Fig.8, the stability robustness bound can be defined as δ_2 . The system will be stable if

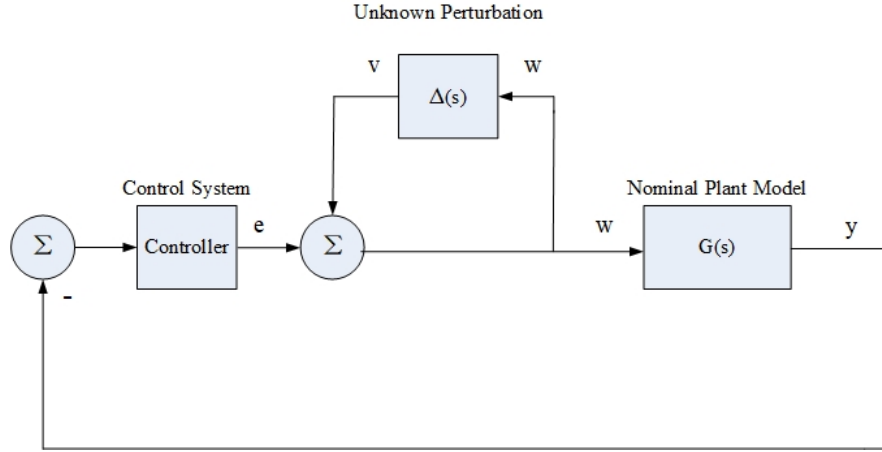


Figure 8. Perturbation Model with δ_2

$\|\Delta(s)\|_\infty < \delta_2$, where δ_2 is the reciprocal of the infinity norm of the system from v to w . The connection between the classical perturbation model with this plant model is, $Q = (I + \Delta)^{-1}$ or $q_i = \frac{1}{(1+c_i)}$. If c_i is a real number then $-\delta_2 < c_i < \delta_2$, so the q_i satisfy $\frac{1}{1+\delta_2} < q_i < \frac{1}{1-\delta_2}$ when q_i is a real number. If q_i is a complex number that is $q = e^{-j\phi} = 1 + C_i$, where 1 is the center of the circle and C_i is a disk with the radius δ_2 , we can find the classical phase margin (PM) is bounded by,

$$\phi_{max} \geq 2\sin^{-1}\left(\frac{\delta_2}{2}\right). \quad (57)$$

Combining the results for δ_1 and δ_2 yields the following relationships between the classical stability margins and the robustness bounds δ_1 and δ_2 .

$$\begin{aligned} UGM &\geq \max\left(1 + \delta_1, \frac{1}{1 - \delta_2}\right) \\ LGM &\leq \min\left(1 - \delta_1, \frac{1}{1 + \delta_2}\right) \\ PM &\geq \max\left(2\sin^{-1}\left(\frac{\delta_1}{2}\right), 2\sin^{-1}\left(\frac{\delta_2}{2}\right)\right). \end{aligned} \quad (58)$$

3.7 Gain-Scheduled Control

In the previous discussion, we implemented the linear controller around the equilibrium point x_{eq} which is a single point. Although we can guarantee the local stability around this point, when a trajectory goes outside of the RoA, the designed controller would not be effective. This situation can be handled when there are a sequence of operating points. In this case we need to design a sequence of different linear controllers corresponding to the operating points in order to maintain the stability.

The result is called a Gain-Scheduled controller. The main idea of the gain scheduling is to linearize a range of operating points which contains the scheduled variables, and then compute the varying linearized dynamics and feedback matrices for the operating points. A series of linear controllers will be combined together to control the nonlinear model [4]. The control precision would be determined by the accuracy selection of scheduled variables. It is usually affected by the computation speed and memory of the hardware for the quadcopter.

Consider the nonlinear system $\dot{x} = f(x, u)$. Let the $x_{eqi}, u_{eqi}, i = 0, 1, \dots$ be a set of equilibrium points and $x_{eq0} = \alpha_\phi$. The value of the control inputs is $u(\alpha_0)$ where satisfies the equation,

$$0 = f(\alpha_0, u(\alpha_0)). \quad (59)$$

We can linearize the nonlinear model about the $(\alpha_0, u(\alpha_0))$ then we have the state matrix and input matrix $A(\alpha_0)$ and $B(\alpha_0)$ where

$$A(\alpha_0) = \frac{\partial f}{\partial x}|_{(x, u) = (\alpha_0, u(\alpha_0))}, B(\alpha_0) = \frac{\partial f}{\partial u}|_{(x, u) = (\alpha_0, u(\alpha_0))}. \quad (60)$$

Assuming that the pair $(A(\alpha_0), B(\alpha_0))$ is controllable, we can now continue to develop a feedback control

$$u_\delta = -K(\alpha_0)(x_\delta - \alpha_0) + u(\alpha_0). \quad (61)$$

such that $A(\alpha_0) - B(\alpha_0)K(\alpha_0)$ is stable. Applying this to the linearized the state space equation results in the closed-loop system,

$$\dot{x}_\delta = (A(\alpha_0) - B(\alpha_0))K(\alpha_0)x_\delta. \quad (62)$$

We know that the Lyapunov stability can guarantee the closed-loop system is stable in the region of attraction of $R\alpha(0) \in D$. When we switch the equilibrium point to α_1 at time t_1 , where $\alpha_1, x(t_1) \in R\alpha(1)$ we can linearize the system around $(\alpha_1, u(\alpha_1))$ to obtain a new controller,

$$u(x; \alpha_1) = -K(\alpha_1)(x - \alpha_1) + u(\alpha_1). \quad (63)$$

to asymptotically stabilize the closed-loop system about $x = \alpha_1$. In this way, we can linearize a sequence of operating points α_i , so long as $\alpha_{k-1}, x(t_k) \in \mathfrak{R}_{\alpha_k} \subset D$ where t_k denotes the time of the k^{th} switch.

List of References

- [1] R.J.Vaccaro, *Digital Control A State-Space Approach*. McGraw-Hill, Inc, 1996.
- [2] E.Lavetsky and K. Wise, *Robust and Adaptive Control with Aerospace Applications*. Springer-Verlag, 2013.
- [3] H. J. Marquez, *Nonlinear Control Systems*. John Wiley and Sons, 2003.
- [4] S. Sawyer, "Gain-scheduled control of a quadcopter UAV," Master's thesis, University of Waterloo, 2015.

CHAPTER 4

Tracking System Design and Simulation Analysis

As discussed, the technique of gain scheduling is well suited for linear parameter-varying systems where the varying parameter can be assigned to be the scheduling variable in the control law [1]. This strategy can be applied to nonlinear systems when the linearized dynamics happen to depend on a parameter of interest to the control engineer. We will use the linearized dynamics of the quadcopter to develop a series of gain scheduled control laws that use the yaw angle as the scheduling variable. We will construct LQR control laws for use with state feedback. Simulations will be conducted for each control law showing tracking trajectories that exhibit the gain-scheduled nature of the controller.

4.1 Linearization and Controllability of Quadcopter Dynamics

We now are interested in linearizing the nonlinear dynamics of the quadcopter about its equilibrium points in order to develop our gain-scheduled controller. To do this, we must first find the equilibrium points of the quadcopter dynamics. Equilibrium points satisfy

$$0 = f(x_{eq}, u_{eq}), \quad (64)$$

where

$$x_{eq} = (x, y, z, 0, 0, \psi, 0, 0, 0, 0, 0)^T. \quad (65)$$

$$u_{eq} = (\omega_1^2, \omega_2^2, \omega_3^2, \omega_4^2)^T. \quad (66)$$

This situation corresponds to the hovering configuration of the quadcopter.

Linearizing the nonlinear dynamics about the steady state values results in

$$\dot{x} = A(x - x_{eq}) + B(u - u_{eq}). \quad (67)$$

where the Jacobian matrices

$$\begin{aligned} A &= \left. \frac{\partial f}{\partial x}(x, u) \right|_{(x,u)=(x_{eq},u_{eq})} \\ B &= \left. \frac{\partial f}{\partial u}(x, u) \right|_{(x,u)=(x_{eq},u_{eq})}. \end{aligned} \quad (68)$$

are given by

$$A = \begin{bmatrix} O_{6 \times 6} & I_{6 \times 6} \\ \Psi_{eq} & O_{6 \times 6} \end{bmatrix}. \quad (69)$$

$$B = \begin{bmatrix} O_{8 \times 4} \\ I_{4 \times 4} \end{bmatrix}. \quad (70)$$

where

$$\Psi_{eq} = \begin{bmatrix} 0 & 0 & 0 & g \cos \psi_{eq} & g \sin \psi_{eq} & 0 \\ 0 & 0 & 0 & g \sin \psi_{eq} & -g \cos \psi_{eq} & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (71)$$

The signals to be tracked are

$$y = (x, y, z, \psi)^T, \quad (72)$$

which can be expressed as

$$y = Cx, \quad (73)$$

where

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}. \quad (74)$$

We denote the controllability matrix of the linearized system as,

$$W_c(A_{eq}, B_{eq}), \quad (75)$$

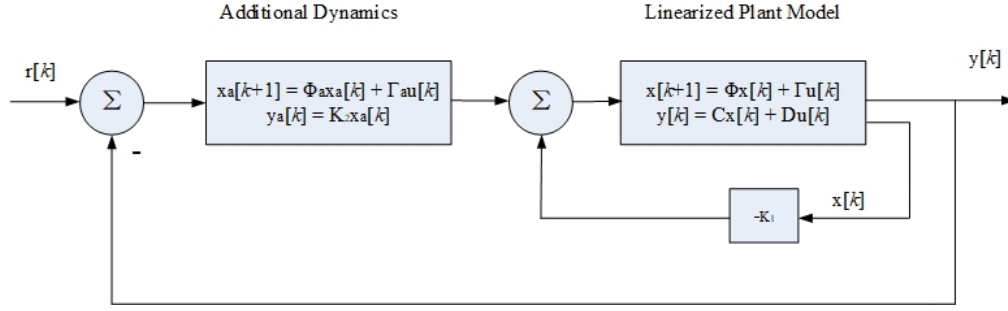


Figure 9. Digital Tracking System

it can be verified that

$$\rho W_c(A_{eq}, B_{eq}) = 12, \forall \psi_{eq} \in \mathfrak{R}. \quad (76)$$

4.2 Obtaining a Tracking System by Regulating the Design Model

The goal of this chapter is to design a tracking system that makes the plant outputs to follow given reference trajectories. Chapter 3 show that a state-feedback gain matrix can be calculated that will regulate the plant (drive all state variables to zero). We now show how to extend the plant to a “design model” such that regulating the design model is equivalent to a tracking system for the plant.

The control algorithm for the quadcopter will be a digital tracking system. As Fig.9, the digital tracking system is comprised of the nonlinear plant, a linear feedback matrix and additional dynamics. Here we derive a continuous-time tracking system for ease of presentation.

Consider the plant as Fig.10 with a constant disturbance vector, w ,

$$\begin{aligned} \dot{x} &= Ax + Bu + w \\ y &= Cx + Du. \end{aligned} \quad (77)$$

Also, let r be a vector of constant commands for the plant outputs. Note that $\dot{r} = 0$ and $\dot{w} = 0$. Since the tracking error $e = y - r$ then we have $\dot{e} = \dot{y} - \dot{r}$ and

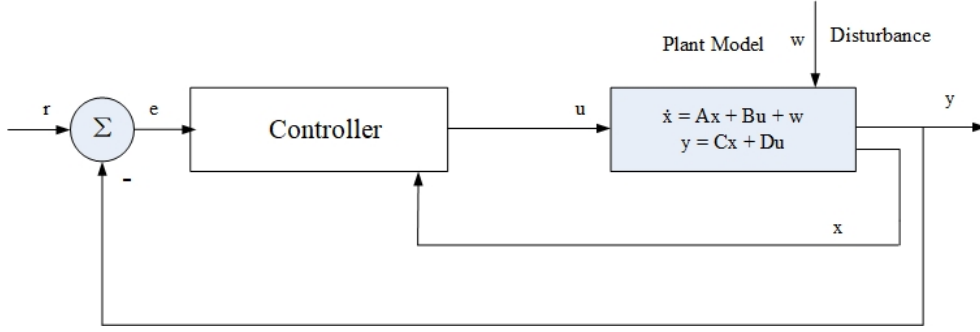


Figure 10. Digital Tracking System

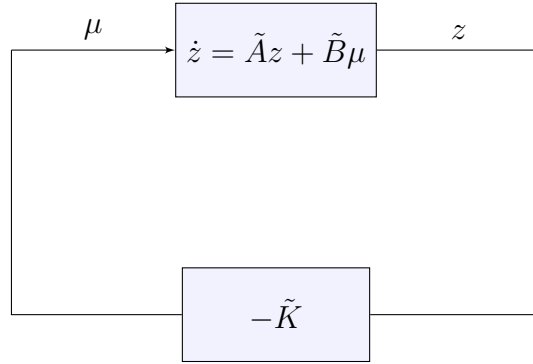


Figure 11. Design Model Structure

$\dot{y} = C\dot{x} + D\dot{u}$. Define two new variables $\xi = \dot{x}$ and $\mu = \dot{u}$. Then $\dot{e} = C\xi + D\mu$ and $\dot{\xi} = \dot{\dot{x}} = A\dot{x} + B\dot{u} = A\xi + B\mu$.

Now define a new state vector z

$$z = \begin{bmatrix} \xi \\ e \end{bmatrix}. \quad (78)$$

then we have

$$\dot{z} = \begin{bmatrix} \dot{\xi} \\ \dot{e} \end{bmatrix} = \begin{bmatrix} A & O \\ C & O \end{bmatrix} \begin{bmatrix} \xi \\ e \end{bmatrix} + \begin{bmatrix} B \\ D \end{bmatrix} \mu. \quad (79)$$

Let

$$\tilde{A} = \begin{bmatrix} A & O \\ C & O \end{bmatrix} \text{ and } \tilde{B} = \begin{bmatrix} B \\ D \end{bmatrix}. \quad (80)$$

then (80) can be written as $\dot{z} = \tilde{A}z + \tilde{B}\mu$. A regulating gain matrix \tilde{K} can be computed for (\tilde{A}, \tilde{B}) using the LQR algorithm, closing the loop with $\mu = -\tilde{K}z$

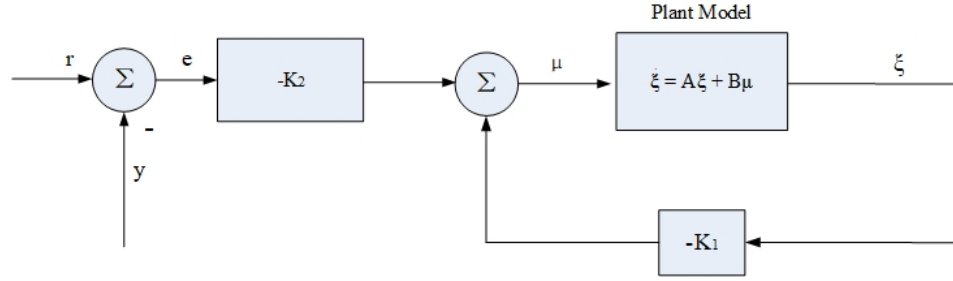


Figure 12. Partition \tilde{K} for the New State Variables

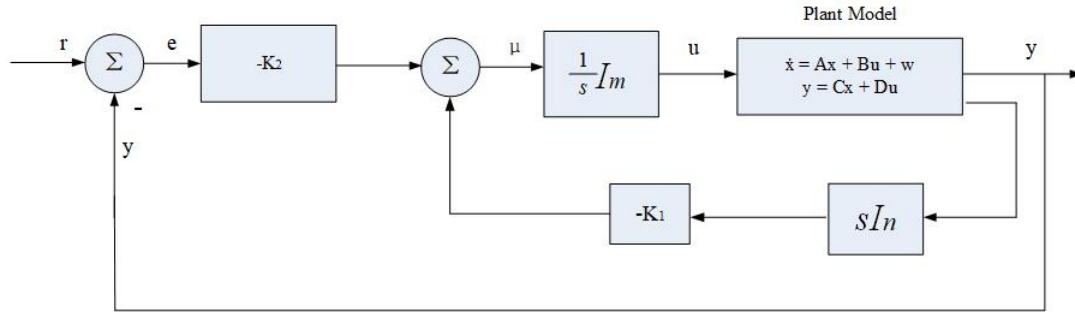


Figure 13. Redraw the System Structure with Actual Plant Model

Fig.11. This regulator drives

$$z = \begin{bmatrix} \xi \\ e \end{bmatrix} \rightarrow \begin{bmatrix} 0 \\ 0 \end{bmatrix} \quad (81)$$

no matter what the disturbance w is. We can partition the \tilde{K} matrix as $\tilde{K} = [K_1 \ K_2]$, so the input $\mu = -K_1\xi - K_2e$. Then the tracking system structure can be obtained from Fig.12 Fig.13 and finally redrawn as Fig.14 and Fig.15 with additional dynamics.

The system will drive y to r and keep zero steady-state error despite of uncertain disturbance w .

4.2.1 Additional Dynamics Blocks

The additional dynamics block consists of m integrators in parallel, whose outputs are multiplied by the gain matrix K_2 . As Fig.16 shows, the input to the additional dynamics are the differences between the measured outputs and the

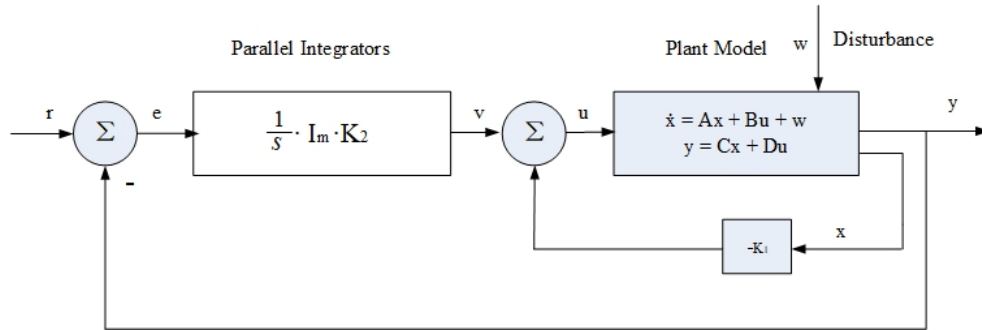


Figure 14. Structure of a State-Space Tracking System

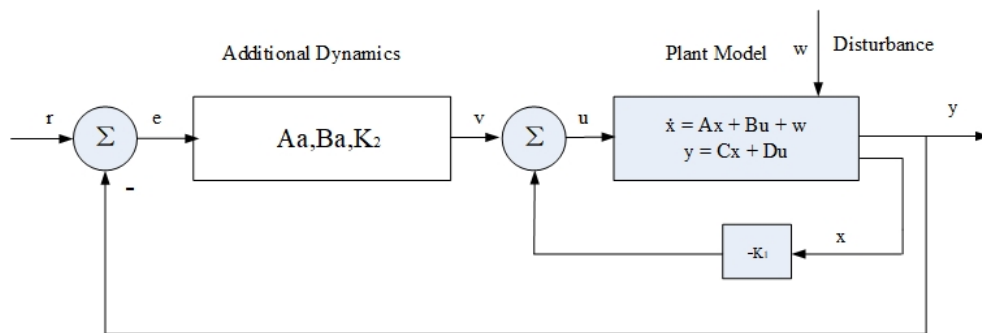


Figure 15. Extend the Design Model with Additional Dynamics

references inputs. We will supply reference input signals for the following four variables: x-position, y-position, z-position (height) and ψ (yaw angle). Thus the additional dynamics block will contain four integrators. Note that the state-space model for parallel integrators is $Aa = O_{m \times m}$ and $Ba = I_{m \times m}$, where m is the number of the plant output variables.

4.2.2 LQR Design of a Linear Tracking System

Given a plant model (A, B, C) with n state variables and m outputs, the procedure developed in the previous subsections is used to calculate the gain matrices K_1 and K_2 that are needed to implement the tracking system shown in Fig.15. The procedure may be summarized as follows:

- (1) Form the matrices \tilde{A} and \tilde{B} for the design model shown in Fig.11.
- (2) Choose LQR weighting matrices Q and R . This choice is discussed below.
- (3) Calculate the LQR gain matrix \tilde{K} using the matlab 'lqr' command.
- (4) Partition the \tilde{K} matrix to obtain K_1 and K_2 . K_1 consists of the first n columns of \tilde{K} and K_2 consists of the last r columns.

The Q and R matrices used in the LQR design are chosen using the following considerations. The Q matrix is chosen to have the following form:

$$Q = \alpha C_d^T C_d, \quad (82)$$

where the $r \times (n + r)$ matrix $C_d = [0 \ M]$. For a four-output plant, $M = \text{diag}(c_1, c_2, c_3, c_4)$. Let α be a varied parameter. c_1, c_2, c_3, c_4 represent the amount of effort applied to tracking the outputs x, y, z, ψ . For the R matrix, define $R = \rho \text{diag}(r_1, r_2, r_3, r_4)$ where (r_1, r_2, r_3, r_4) represent the efforts to implement the plant inputs the angular speed of the 4 rotors. Increasing the value of c_1, c_2, c_3, c_4 will get a good response for the corresponding states, decreasing the value of r_1, r_2, r_3, r_4 will allocate more energy employed by the control inputs. Varying α and ρ will affect the dynamics performance of the system and find a optimal

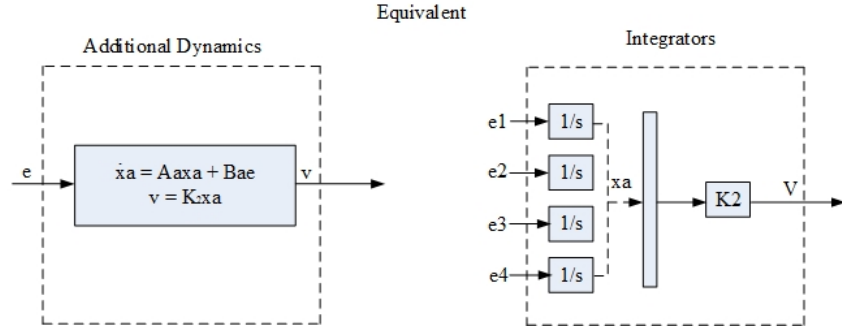


Figure 16. Additional Dynamics and Integrators

$u = -Kx$ where the feedback matrix K is calculated by the optimal Q and R to minimize the cost function J in (38).

4.3 Design of a Fixed-Gain Tracking System

We will begin with developing the fixed-gain controller to track a step input reference. The nonlinear model of the quadcopter is

$$\dot{x} = f(x, u). \quad (83)$$

Let

$$\dot{x}_\delta = Ax_\delta + Bu_\delta. \quad (84)$$

be the linearization about the equilibrium point x_{eq}, u_{eq} , where,

$$x_{eq} = (x, y, z, 0, 0, \psi, 0, 0, 0, 0, 0)^T \quad (85)$$

$$u_{eq} = (\omega_1^2, \omega_2^2, \omega_3^2, \omega_4^2)^T \quad (86)$$

and $x_\delta = x - x_{eq}, u_\delta = u - u_{eq}$.

The linear tracking system will be designed using x_{eq} and u_{eq} . In order to control the nonlinear drone, a linear feedback controller will be used, as shown in Fig.17. Note that the linearized dynamics do not depend on the spatial coordinates

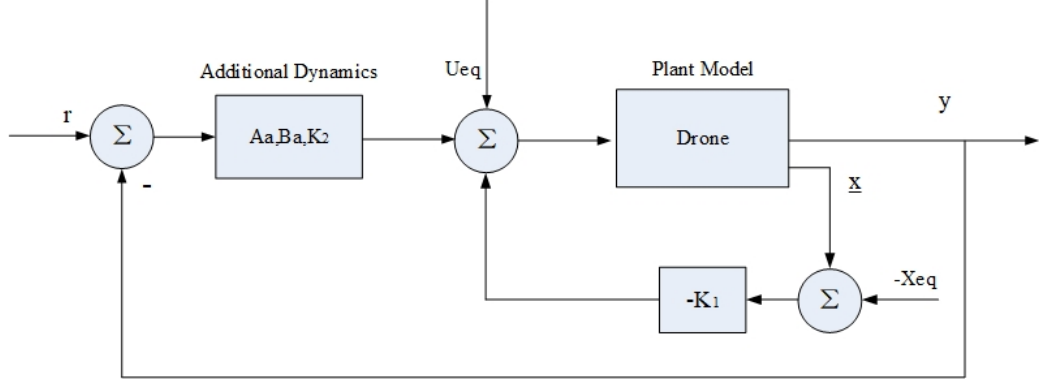


Figure 17. Tracking System for the Nonlinear Drone

(x, y, z) of the steady state. As such, the gain matrix K only depends on the yaw angle configuration ψ of the quadcopter, so we can use this controller to track a step reference,

$$r = (x, y, z, \psi)^T. \quad (87)$$

We will now go through a series of choices for the Q and R weight matrices and evaluate the performance of the resulting tracking systems. Firstly, we linearized the nonlinear dynamics at $x_{eq} = (2, 2, -2, 0, 0, \pi/6, 0, 0, 0, 0, 0, 0)^T$ with the balanced input $u_{eq} = (\omega_1^2, \omega_2^2, \omega_3^2, \omega_4^2)^T$, where $\omega_1 = \omega_2 = \omega_3 = \omega_4 = \frac{mg}{4ka}$, when the quadcopter stay in a constant altitude hover state. Then we got the Jacobian matrices A, B . For the choice of symmetric positive Q and R matrix, let c_1, c_2, c_3, c_4 and u_1, u_2, u_3, u_4 be a constant number 1, that means M and R are identity matrices.

Fig.18, 19 and 20 show the step input position and ψ angle reference tracking performance. The reference inputs to the model are $r = (2, 2, -2, \pi/6)^T$ and the scaling parameters for the Q and R matrices are $\alpha = 10, \rho = 0.01$, thus $Q = \text{diag}(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 100, 100, 100, 100)$ and $R = \text{diag}(0.01, 0.01, 0.01, 0.01)$.

Fig.18, 19 show that the actual position and ψ angle all reach the zero steady

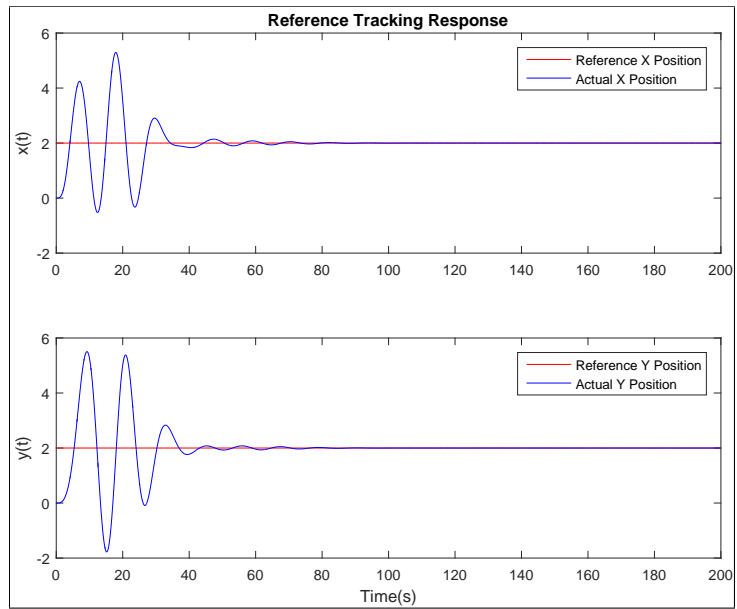


Figure 18. Step Reference Tracking X and Y states

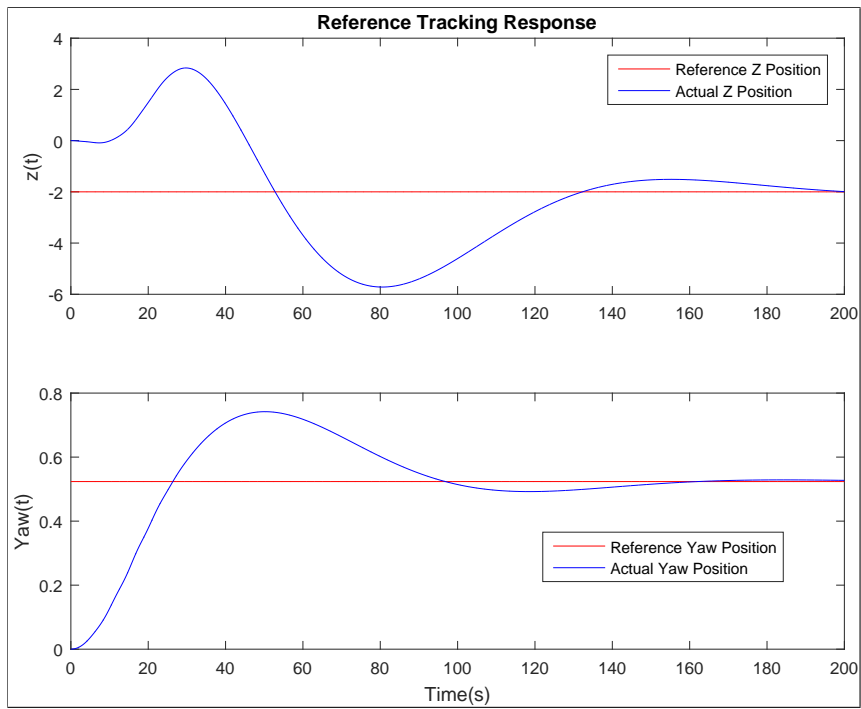


Figure 19. Step Reference Tracking Z and Yaw states

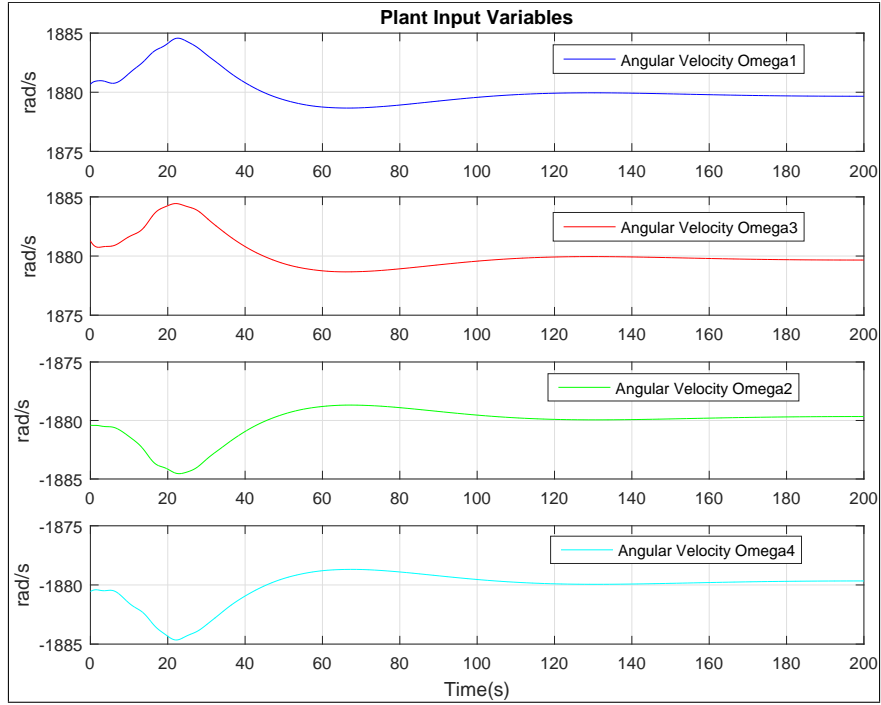


Figure 20. Plant Input Angular Speed

state error as expected, although there is excessive oscillation. The oscillation can be reduced by penalizing the seventh, eighth and ninth state variables, \dot{x} , \dot{y} and \dot{z} . It takes about 80 seconds for x and y states to reach steady state, while z and ψ need more time, about 190 seconds. Increasing the diagonal elements of Q will improve the setting time for the corresponding states, and increasing the element of R will reduce the energy the controller uses to stabilize the plant. Fig.19 shows that the input angular velocity are equal in magnitude but have opposite direction for the opposite pair ω_1, ω_3 and ω_2, ω_4 when the quadcopter take off and hover at a stable state. And the robustness bound values of δ_1 and δ_2 for this control system are 0.5132 and 0.9771 respectively.

It takes more time for the height and yaw angle to reach a steady state. So, we set the c_1, c_2, c_3, c_4 to be (10, 10, 100, 100) giving more effort for z and ψ states control. We will choose some weights to penalize the \dot{x}, \dot{y} and \dot{z} and keep

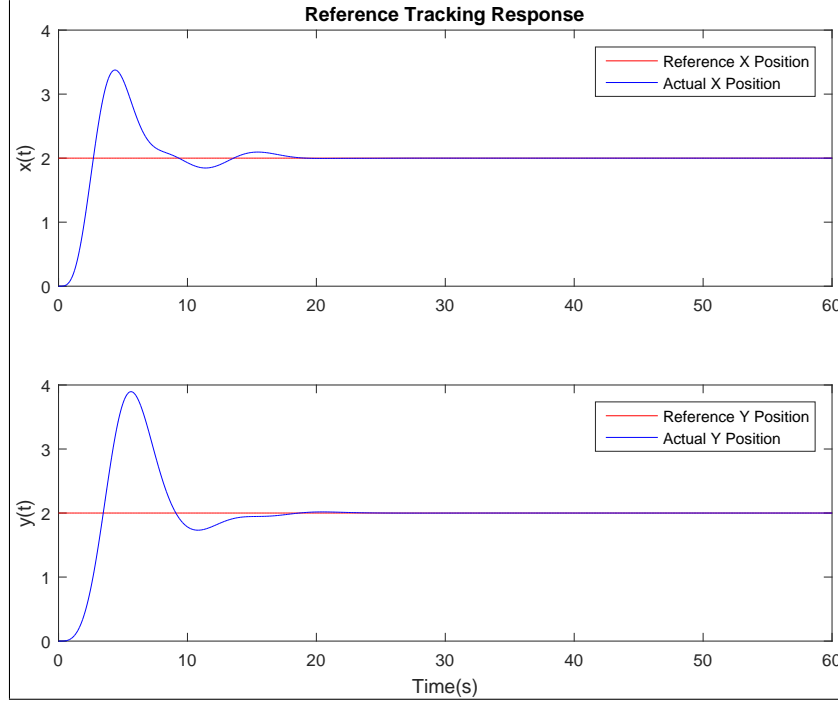


Figure 21. Step Reference Tracking X and Y states

(u_1, u_2, u_3, u_4) , α and ρ as the first test. So the $Q = \text{diag}(0, 0, 0, 0, 0, 0, 1e+5, 1e+5, 1e+3, 0, 0, 0, 1e+4, 1e+4, 1e+6, 1e+6)$ and $R = \text{diag}(0.01, 0.01, 0.01, 0.01)$.

Fig.21 and 22 show that when effort applied to control state is increasing, it takes less time for the states reach a steady state. For x and y states, it needs about 15 seconds, while for z and ψ the time would be around 35 seconds. The overshoot for x, y, z, ψ is much better than previous values. And the values of δ_1 and δ_2 for this control system are 0.5628 and 0.9838 respectively.

Although the overshoot has been improved and settling time is smaller, we want to seek a better transient response with good combination of Q and R . This time we keep the c_1, c_2, c_3, c_4 as $(10, 10, 100, 100)$, (u_1, u_2, u_3, u_4) as one, and $\alpha = 10$, penalty for \dot{x}, \dot{y} and \dot{z} , change $\rho = 0.001$, thus $Q = \text{diag}(0, 0, 0, 0, 0, 0, 1e+6, 1e+6, 1e+4, 0, 0, 0, 1e+4, 1e+4, 1e+6, 1e+6)$ and $R = \text{diag}(0.001, 0.001, 0.001, 0.001)$.

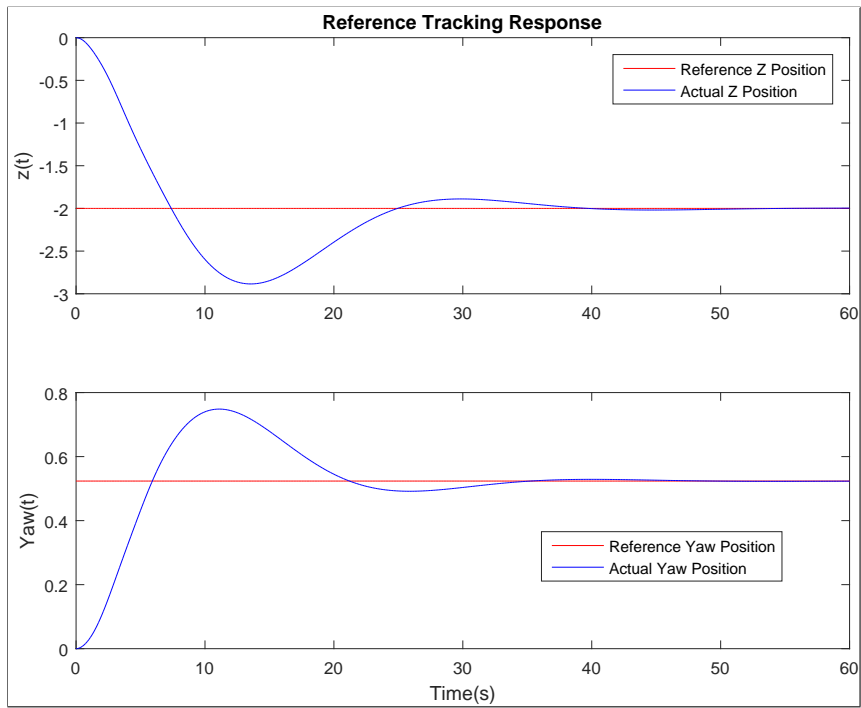


Figure 22. Step Reference Tracking Z and Yaw states

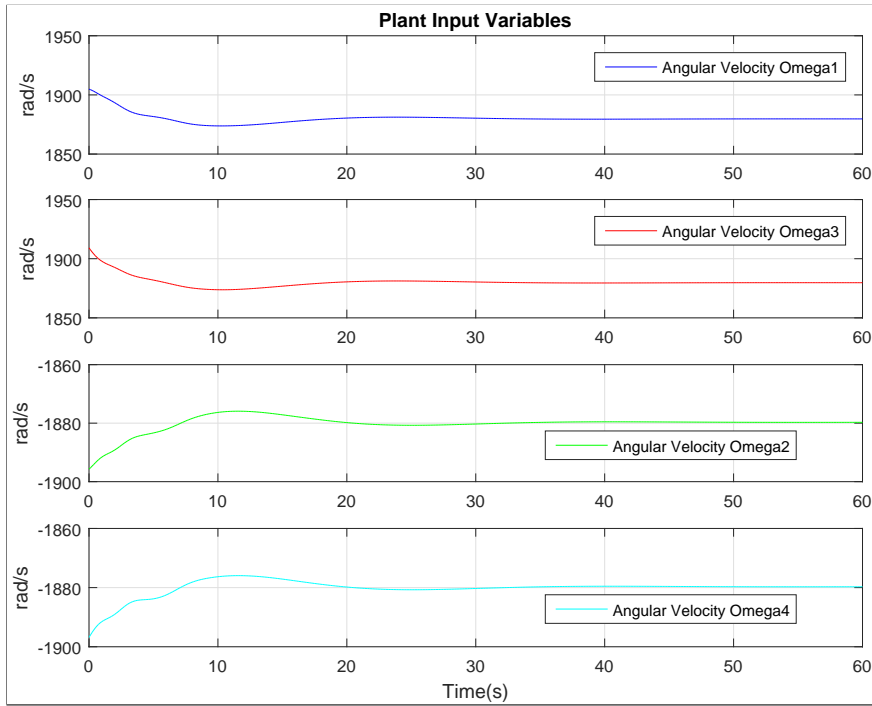


Figure 23. Plant Input Angular Speed

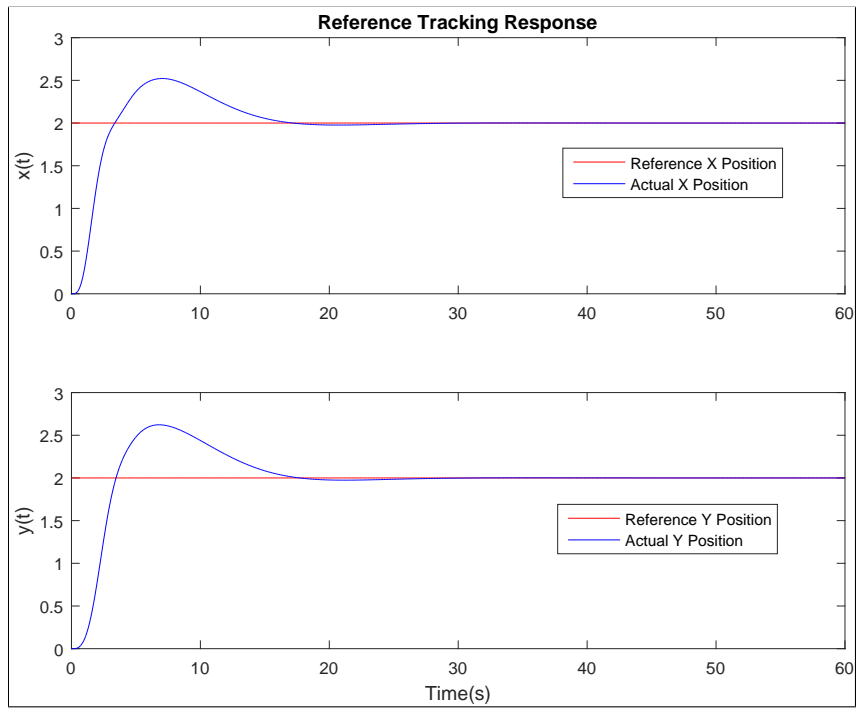


Figure 24. Step Reference Tracking X and Y States

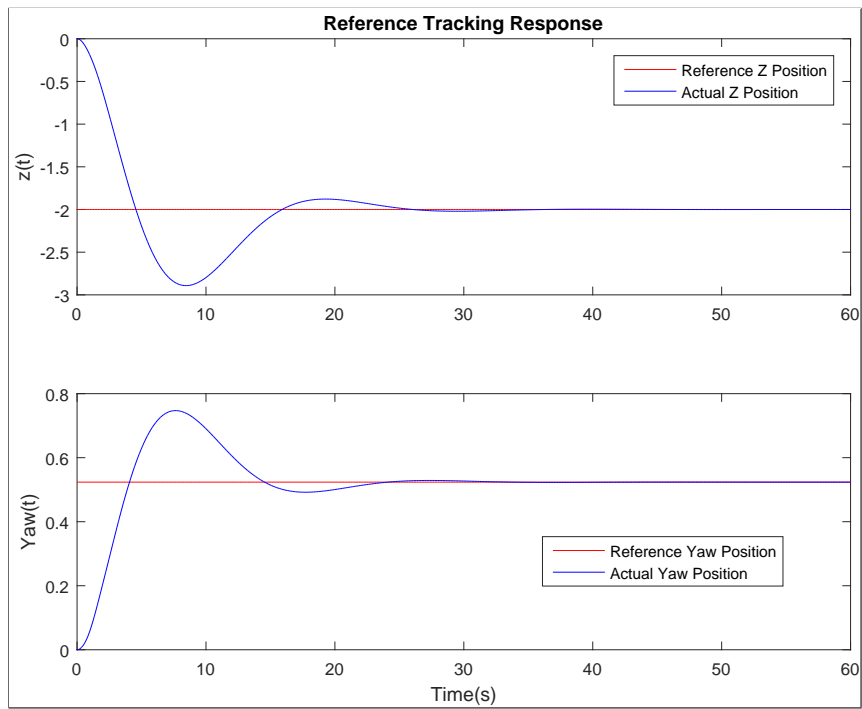


Figure 25. Step Reference Tracking Z and Yaw States

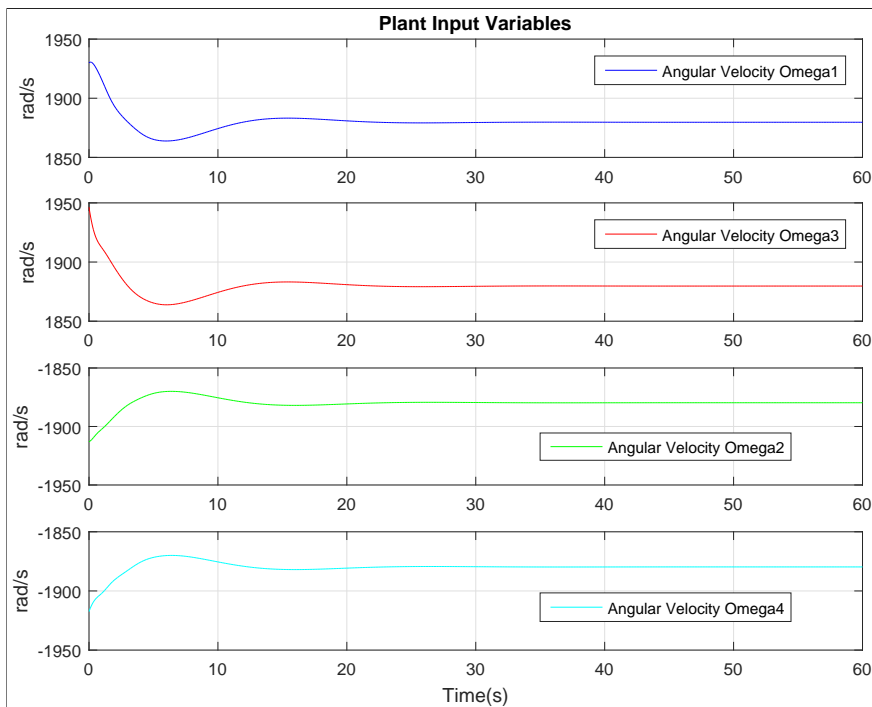


Figure 26. Plant Input Angular Speed

Fig.24, 25 show that it takes more time, about 15 seconds for the states x and y to reach a steady state, while for z and ψ about 25 seconds. So decrease the weights of R will produce a good response and quicker converge to the steady state with little overshoot. And the values of δ_1 and δ_2 for this control system are 0.6062 and 0.9716 respectively.

In the last test we set the c_1, c_2, c_3, c_4 to be (100, 100, 1000, 1000) and keep $\alpha = 10, \rho = 0.001$ (u_1, u_2, u_3, u_4) as 1 and penalties for \dot{x}, \dot{y} and \dot{z} . Thus $Q = \text{diag}(0, 0, 0, 0, 0, 0, 1e + 6, 1e + 6, 1e + 4, 0, 0, 0, 1e + 6, 1e + 6, 1e + 8, 1e + 8)$ and $R = \text{diag}(0.001, 0.001, 0.001, 0.001)$. Compared with the plant input Fig.26 and Fig.29 we can see that when we put bigger penalties on the Q matrix, more effort should be supplied to the command control inputs and the quadcopter would have to take off with a much larger motor speed. This also happens when we put larger penalties (1e10, 1e10, 1e6) on the derivatives of x,y and z positions. That is just the

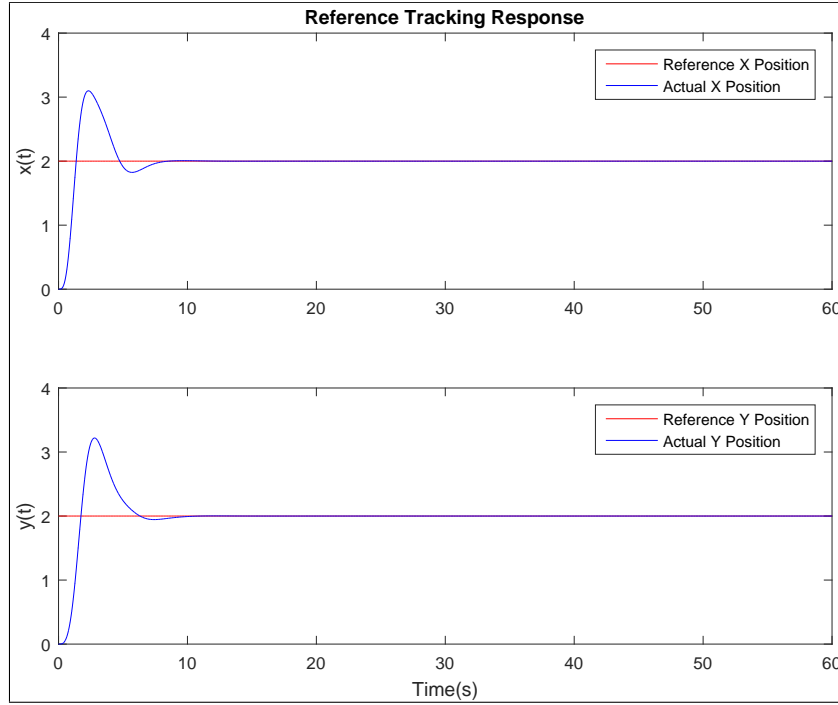


Figure 27. Step Reference Tracking X and Y States

limitation of the LQR regulator, we have to make a trade-off between the outputs transient performance with the power we can supply for the plant inputs.

Fig.27, 28 show that it takes more time, about 8 seconds, for the states x and y z and yaw to reach a steady state. Fig.29 shows that the quadcopter takes about 10 seconds from taking off to a steady state. The transient response performance is good, and the values of δ_1 and δ_2 for this control system are 0.5677 and 0.9669 respectively.

This controller can also be used to track trajectories in which there is little variation in the yaw angle of the quadcopter. However, when the yaw angle undergoes substantially variation, the fixed-gain controller will work not well. To correct this issue, a set of linear tracking gains can be calculated for a given set of equilibrium points. A gain-scheduled tracking system will be developed using the measured yaw angle as a scheduling variable.

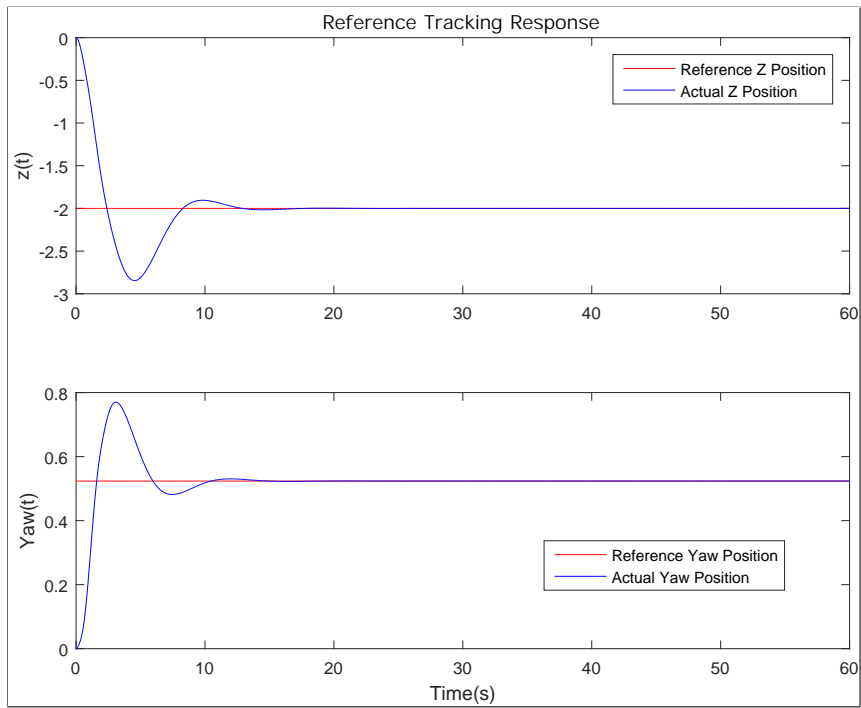


Figure 28. Step Reference Tracking Z and Yaw States

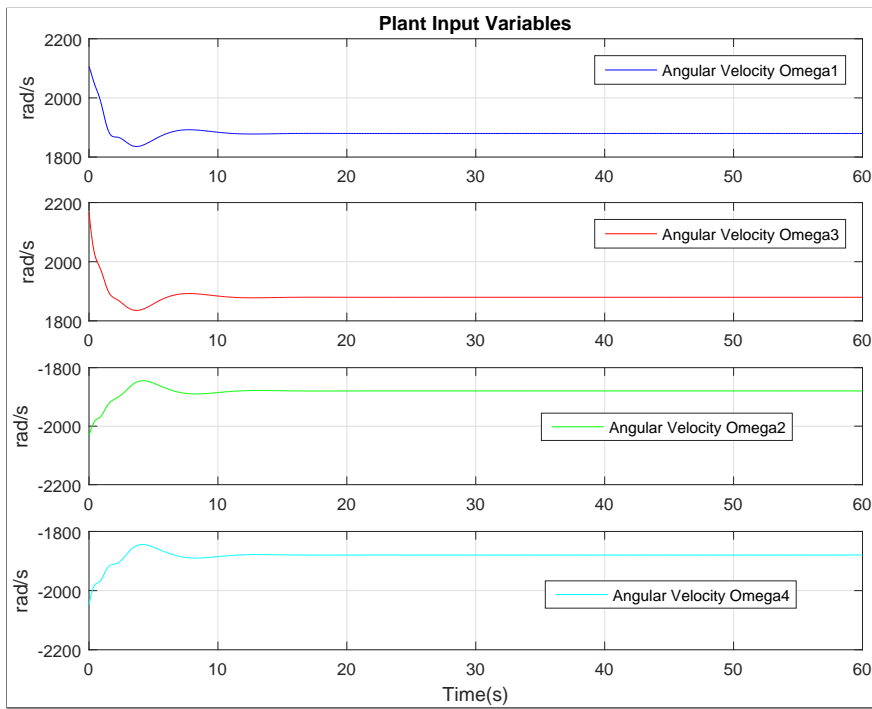


Figure 29. Plant Input Angular Speed

4.4 Gain-Scheduled LQR Control

The fixed gain controller we designed has been evaluated for step reference tracking. Now we will show how the feedback matrices K_1 and K_2 may be obtained by linear interpolation to track references with significant change in the yaw angle.

4.4.1 Figure - 8 trajectory tracking

Tracking a figure eight curve in a 2 dimensional plane. During this test only the design values Q and R in the last group experiment for the step references tracking will be employed. The figure eight references inputs are generated in the commands block. The range of the desired x_m and y_m are defined as

$$x_m(t) = A_x \sin \frac{2t}{T} \quad (88)$$

$$y_m(t) = A_y \sin \frac{t}{T}, \quad (89)$$

where A_x is the amplitude of x_m , A_y is the amplitude of y_m and T is the period of the figure eight. This means that a figure eight can be commanded to remain in an area of $A_x \times A_y$ and one of the period of the figure eight would be completed in $2\pi T$ seconds. The relation between the x_m and y_m with the ψ angle command is given by,

$$\psi_{cmd} = \tan^{-1} \frac{A_y \cos(\frac{t}{T})}{2A_x \cos(\frac{2t}{T})}. \quad (90)$$

Note that the \tan^{-1} is calculated by the *atan2* function to make it change smoothly in all quadrants. The main idea of gain scheduling controller is to design a family of linear controllers around operating points, or regions that should guarantee the robustness, and change according to the variation of scheduled variable ψ angle. The change of controllers usually be implemented with interpolation or by switching from different values of scheduled variable. [2] shows the instability created by switching among controllers, which may come from the transition dynamics that the information is not covered by each switching linear controllers.

So, interpolation methods will be focused in the thesis. The values spacing Δ of interpolation for the ψ angle will be determined first. We can define a $\Psi(t)$ for the real time position of yaw angles and check which interval $\psi(t)$ lies in (Define $\psi(q) \leq \psi(t) < \psi(q+1)$ and $\Delta = \psi(q+1) - \psi(q)$ where q is the index). Then, a sequence of linear feedback matrix could be calculated using equilibrium points with the changeable ψ angle. As Fig.30 shows, the matlab function inside the controller block defines a parameter $c = \frac{\psi(t)-\psi(q)}{\Delta}$. The actual feedback matrix is computed using the parameter c . In this way, the missing information between the transition dynamics could be detected mostly. Also, the modulo operation used to make sure the ψ angles value stays in the range $(0, 2\pi)$. The control precision is usually determined by the spacing Δ radians and limited by the physical hardware. We tried the following values for Δ as $\pi/3$, $\pi/12$ and $\pi/30$. Note that these tests are completed based on the $Q = \text{diag}(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1e+6, 1e+6, 1e+8, 1e+8)$ and $R = \text{diag}(0.001, 0.001, 0.001, 0.001)$.

We can see that as the Δ decreases from $\pi/3$ to $\pi/12$ and finally to $\pi/30$, the tracking performance becomes much better. The actual tracking locus is very sloppy and fuzzy when $\Delta = \pi/3$. When the Δ decreased to $\pi/12$, although with little distortion during the trajectory tracking, the overall performance is good. The improvement from last when $\Delta = \pi/30$ is more noticeable with precise tracking.

For the fixed-gain tracking system, penalties are added to the derivative of x , y and z positions to improve the overshoot. While, for the Figure-8 trajectory tracking, we will put the same weights on the Q matrix with different spacing values of Δ . So the matrices $Q = \text{diag}(0, 0, 0, 0, 0, 0, 1e6, 1e6, 1e4, 0, 0, 0, 1e+6, 1e+6, 1e+8, 1e+8)$ and $R = \text{diag}(0.001, 0.001, 0.001, 0.001)$ for this tracking controller. Also, in the simulations of the gain-scheduled controllers, we will be using the same Q and R matrices for different equilibrium points.

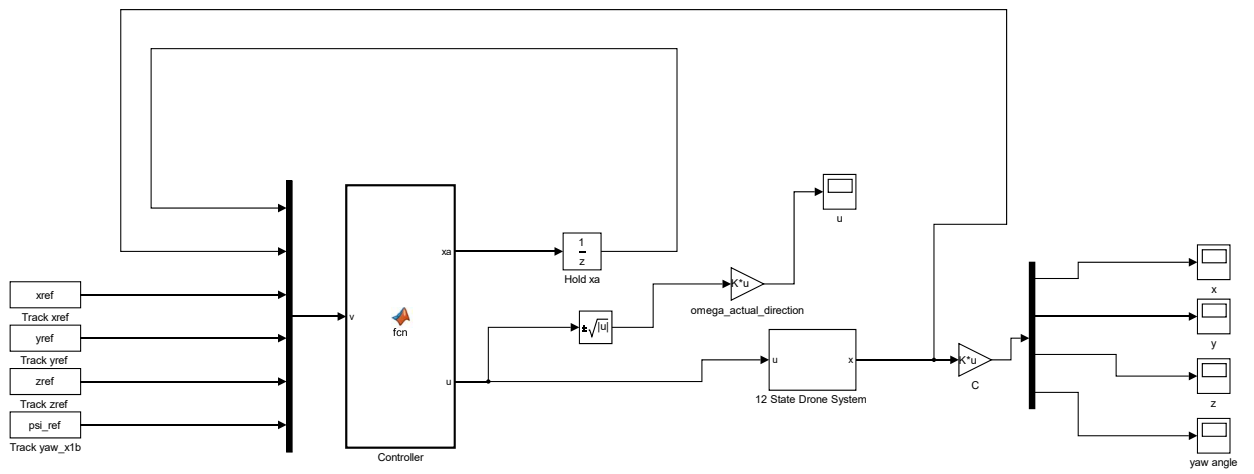


Figure 30. Gain Scheduling Controller Model

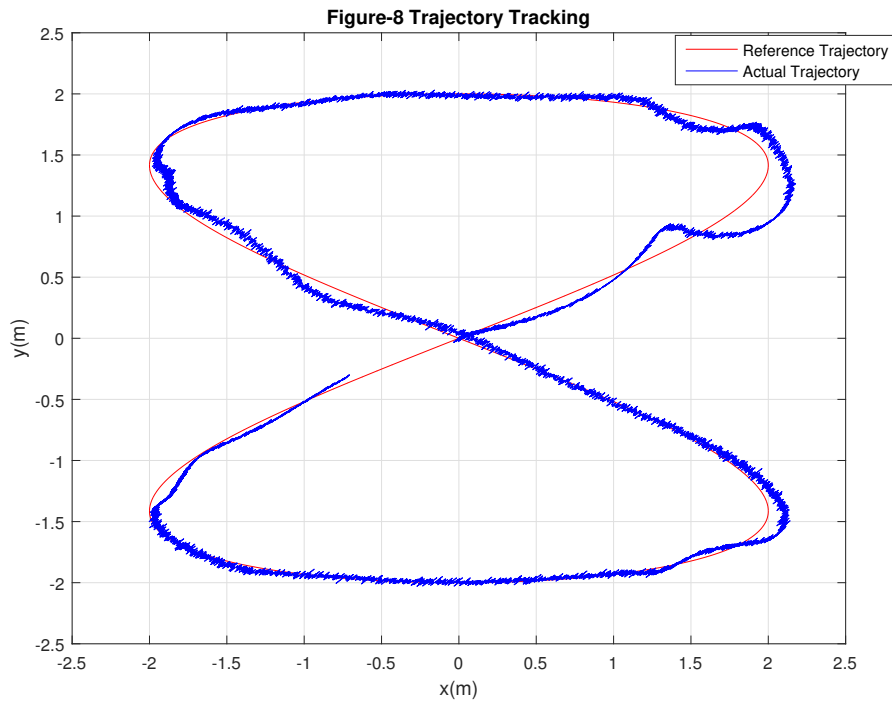


Figure 31. Steady State Figure Eight Tracking xy -plane $\Delta = \pi/3$

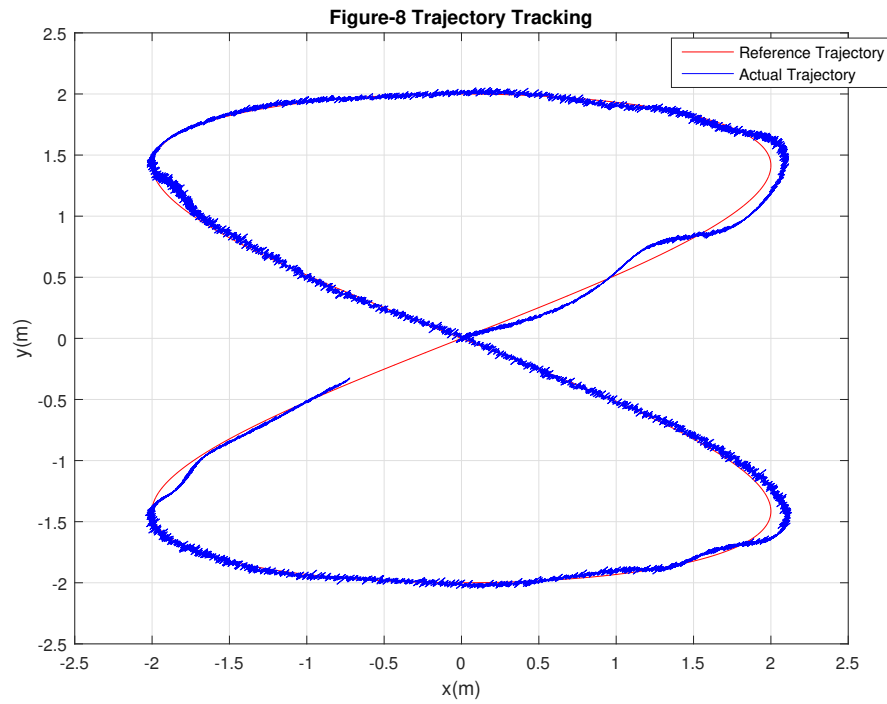


Figure 32. Steady State Figure Eight Tracking xy -plane $\Delta = \pi/12$

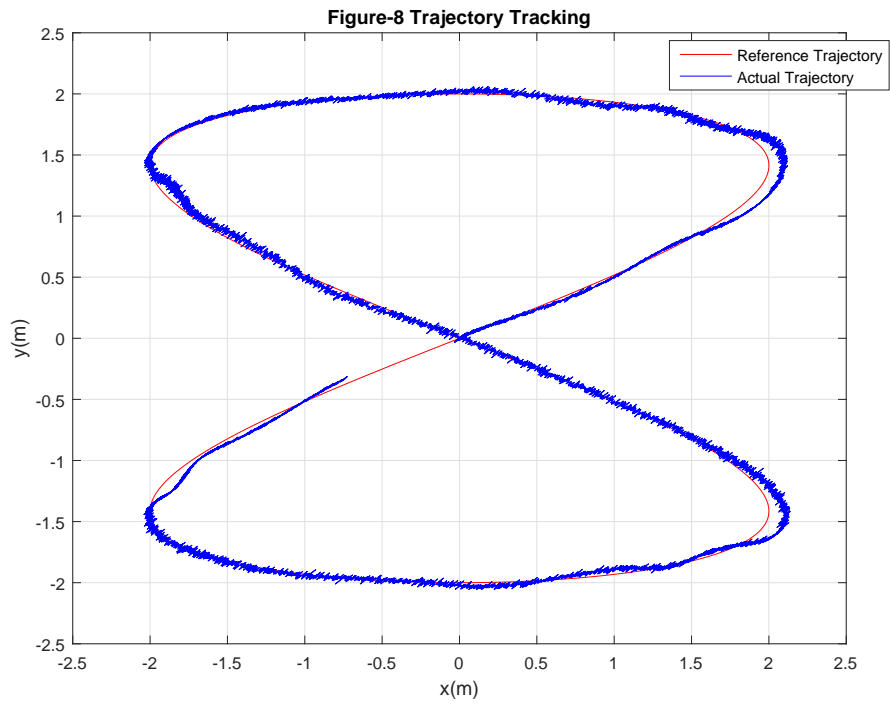


Figure 33. Steady State Figure Eight Tracking xy-plane $\Delta = \pi/30$

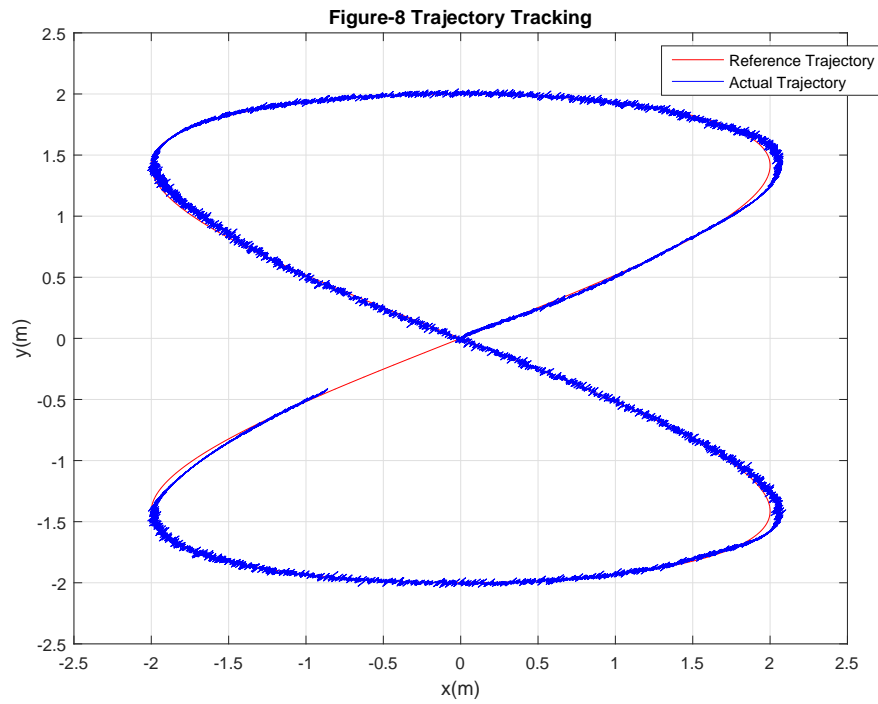


Figure 34. Figure Eight Tracking with Penalties $\Delta = \pi/3$

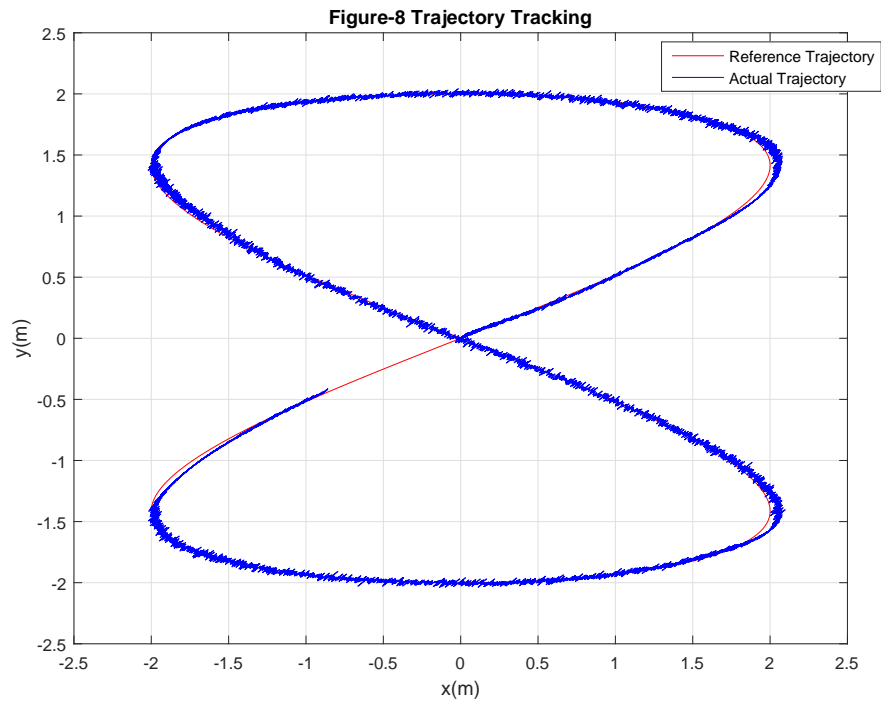


Figure 35. Figure Eight Tracking with Penalties $\Delta = \pi/12$

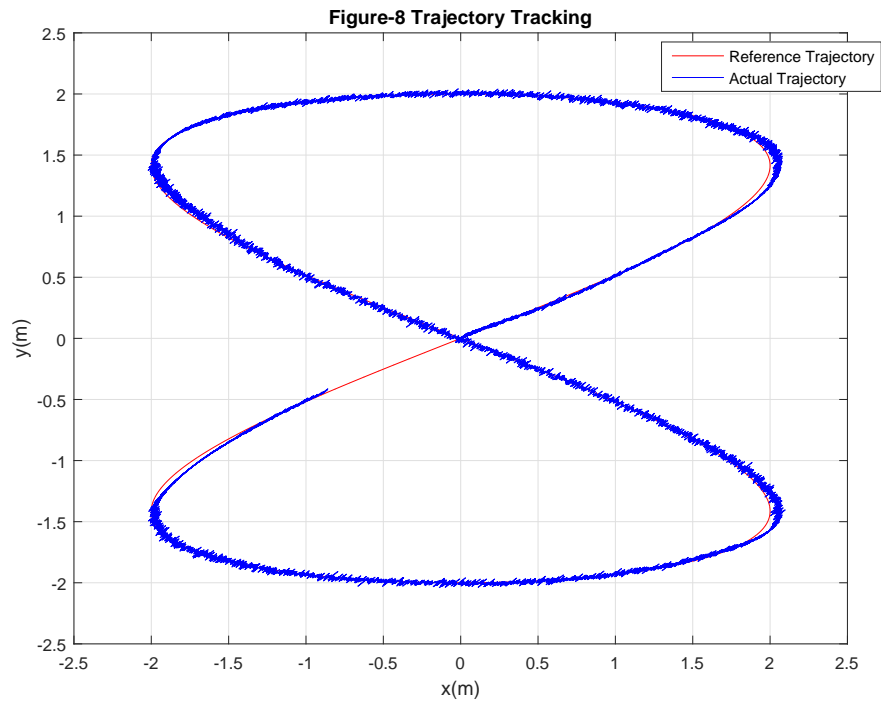


Figure 36. Figure Eight Tracking with Penalties $\Delta = \pi/30$

Fig.34, 35, 36 depict the figure-8 curve tracked with different spacing values Δ for yaw angles. It is clear that there is little change for the tracking performance with values of Δ from $\pi/3$ to $\pi/12$ until to $\pi/30$. Even for choose a bigger spacing value Δ , the quadcopter tracks the reference inputs well.

4.4.2 3D trajectory tracking

We will further conduct a simulation where the quadcopter is to track a rising spiral and lemniscate trajectory in 3-dimensional space to test the effectiveness of the gain-scheduled controller. The helix curves can be parameterized as

$$\begin{aligned}x_m(t) &= A_x \sin \frac{2t}{T} \\y_m(t) &= A_y \cos \frac{2t}{T} \\z_m(t) &= A_z t,\end{aligned}\tag{91}$$

where A_x is the amplitude of x_m , A_y is the amplitude of y_m and T is the period of the figure eight. The ψ angle can be defined as the following equation,

$$\psi_{cmd} = \tan^{-1} \frac{A_y \cos(\frac{t}{T})}{2A_x \cos(\frac{2t}{T})}.\tag{92}$$

We will set $A_x = 15, A_y = 15, A_z = 0.5, t = 0.005, T = 10$ in the simulation. The 3D trajectories tracking is performed using $Q = \text{diag}(0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1e + 6, 1e + 6, 1e + 8, 1e + 8)$ and $R = \text{diag}(0.001, 0.001, 0.001, 0.001)$ without penalties on the derivatives. The quadcopter is manipulated to fly 3 rotations around z axis. The actual trajectory is slightly difference with the references around the start point due to the initial state setting for the quadcopter, this can also be seen in the xy-plane projection and y trajectory position response. At about 5 seconds, the measured trajectory begins to track the references well, although with little deformity. The Fig. 39 xy-plane plot gives a top vision of the behavior for the helix curve. Fig.38 shows

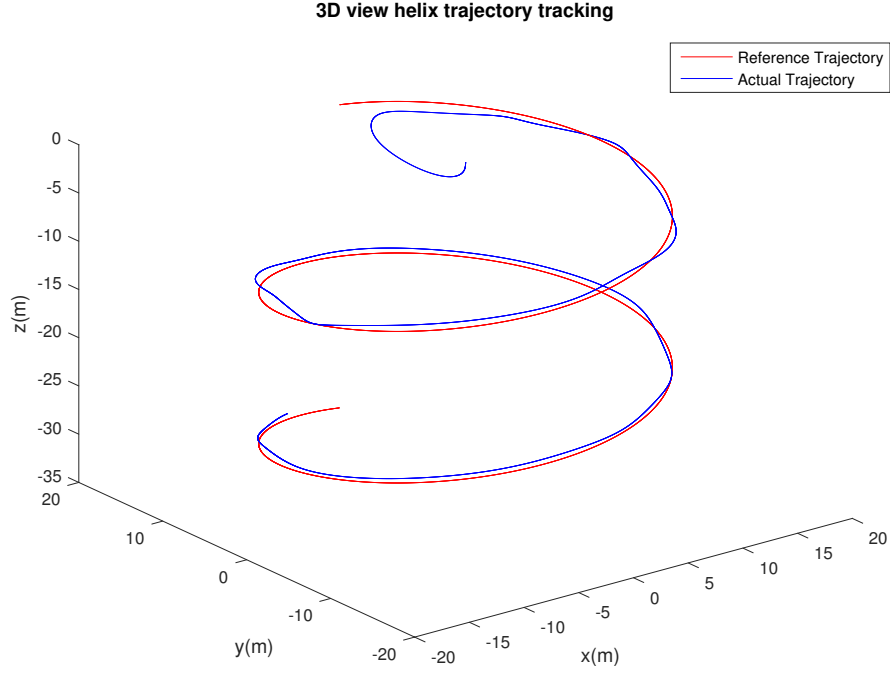


Figure 37. 3D Helix Trajectory Tracking

that the two clockwise spinning and two counterclockwise spinning rotors have the same magnitude but opposite direction of the angular speed. This two pairs rotors make quadcopter rise quickly and balance drag-induced torque about z-axis. Fig.40 shows the roll angle variation is less than $0.8rad$ even for the big *yaw* angle change during the trajectory tracking. Fig.41, 42, 43 and 44 show that the transient response do a good job. The settling time and overshoot are acceptable.

Now we switch to another 3D trajectory, which is a lemniscate trajectory defined as,

$$\begin{aligned}
 x_m(t) &= A_x \sin \frac{2t}{T} \\
 y_m(t) &= A_y \sin \frac{t}{T} \\
 z_m(t) &= A_z t - 5,
 \end{aligned} \tag{93}$$

where A_x is the amplitude of x_m , A_y is the amplitude of y_m and T is the period of the figure eight. The relation between the x_m and y_m with the ψ angle using

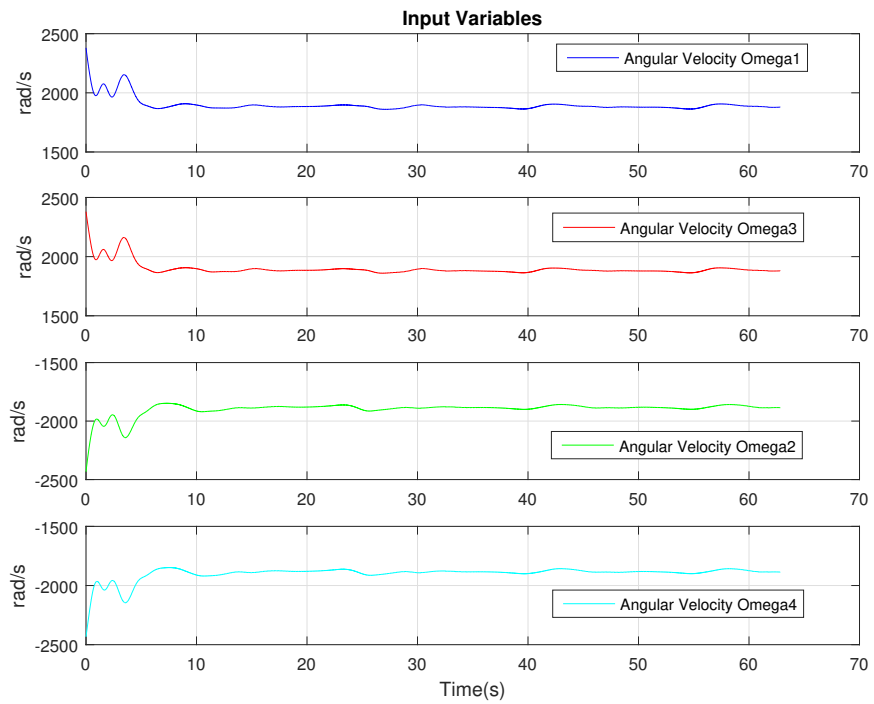


Figure 38. Plant Input Angular Speed

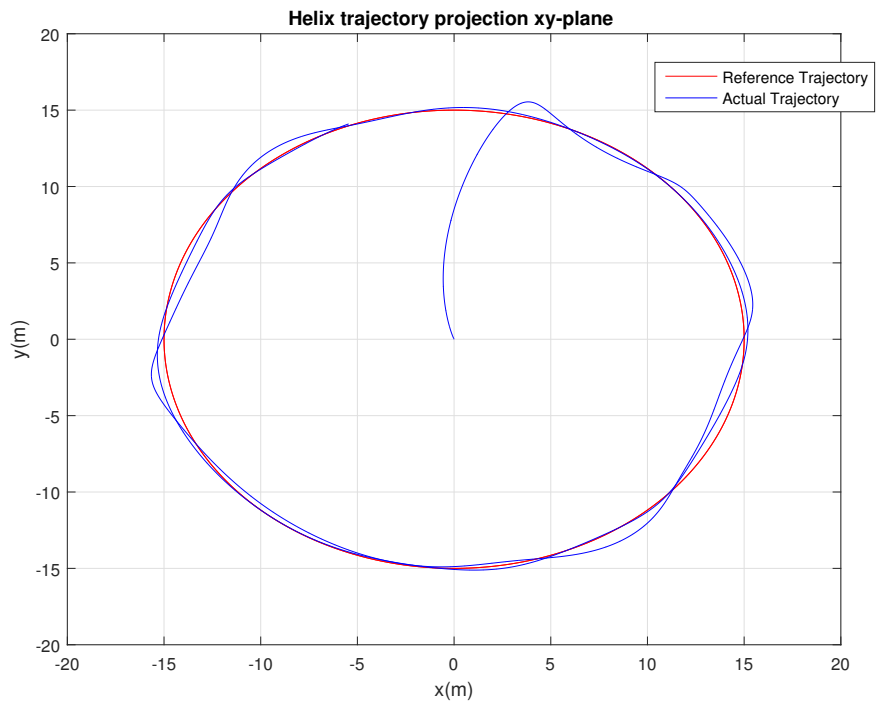


Figure 39. Helix Trajectory Tracking xy-plane

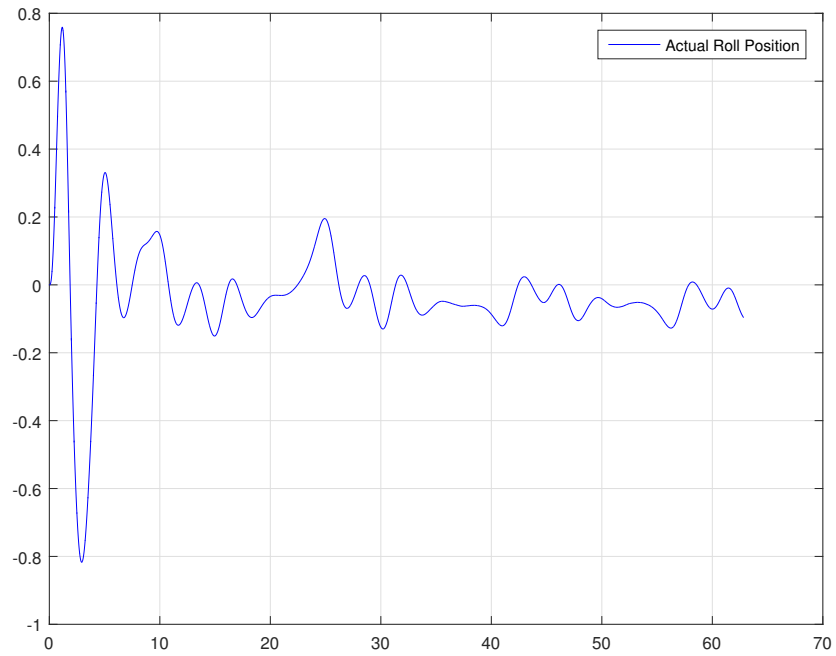


Figure 40. Roll Angle Variation

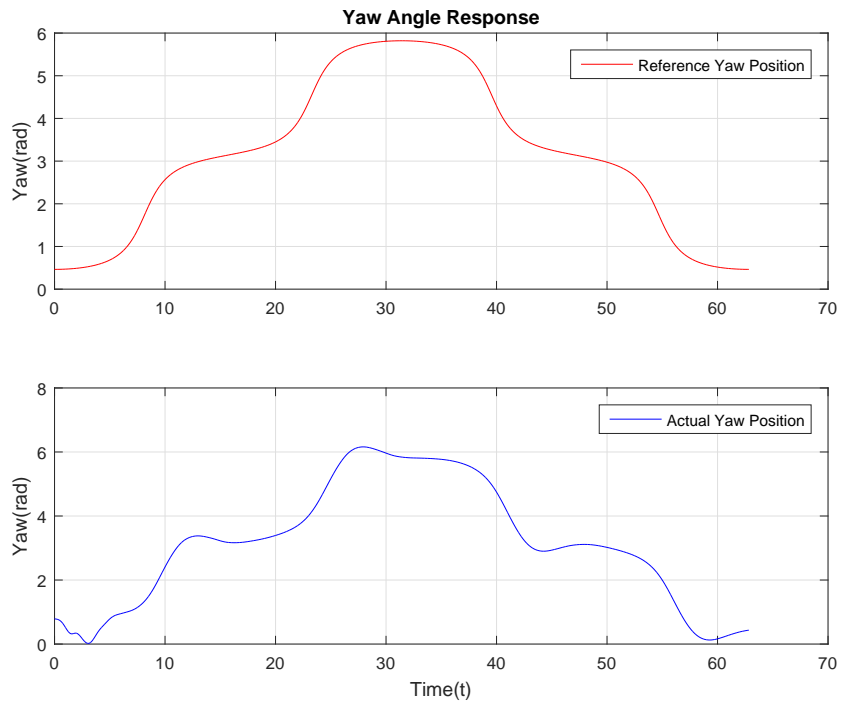


Figure 41. Yaw Angle Trajectory Tracking

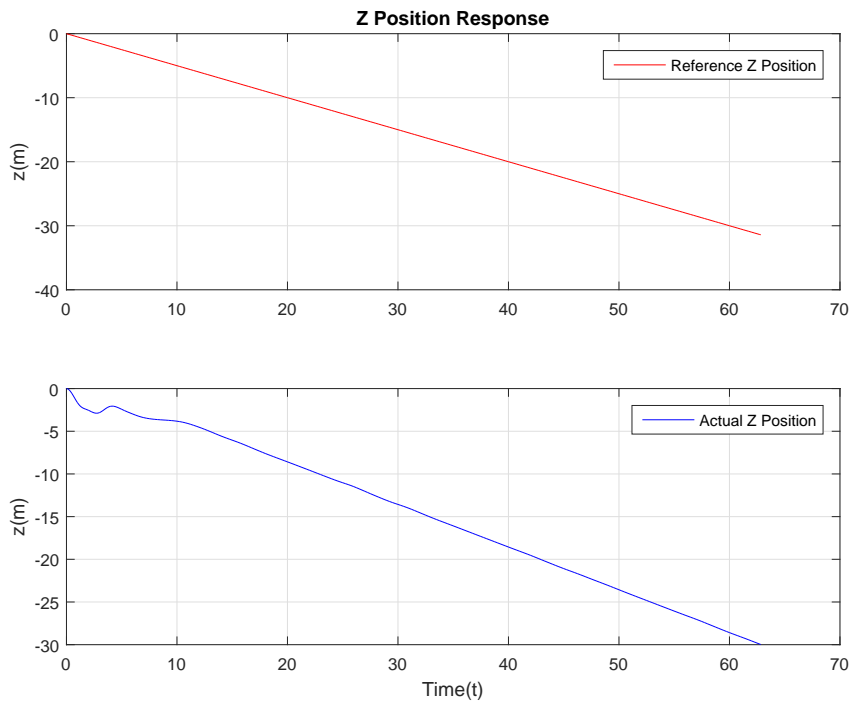


Figure 42. Z Direction Trajectory Tracking

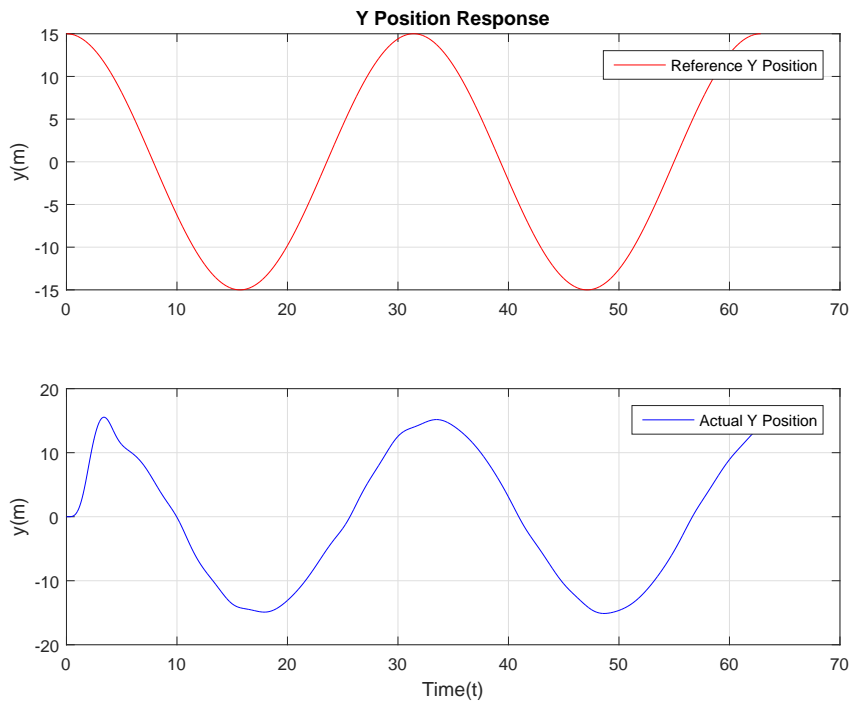


Figure 43. Y Direction Trajectory Tracking

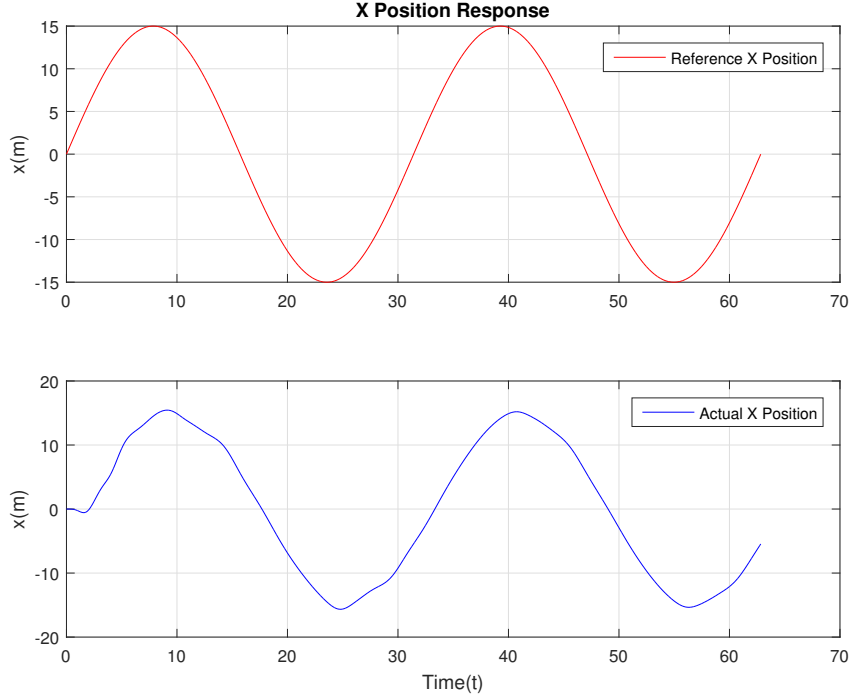


Figure 44. X Direction Trajectory Tracking

the following equation (94). In the simulations, we will use $A_x = 2$, $A_y = 2$, $A_z = 0.5$, $t = 0.005$, $T = 10$.

$$\psi_{cmd} = \tan^{-1} \frac{A_y \cos(\frac{t}{T})}{2A_x \cos(\frac{2t}{T})}. \quad (94)$$

The actual trajectory is slightly different with the references around the start point due to the initial state setting for the quadcopter, this can also be seen in the xy-plane projection and y trajectory position response. To check the ability of the tracking system to “catch up” to the references, the initial states we set for the drone is different from the initial values of the references. Fig.45 shows that the two clockwise spinning and two counterclockwise spinning rotors have the same magnitude but opposite direction of the angular speed. This motion make the quadcopter rise quickly and balance drag-induced torque about z-axis. Fig.48 shows the roll angle variation is less than 0.025 rad even for the big yaw angle change during the trajectory tracking. Fig.49, 50, 51 and 52 show that the

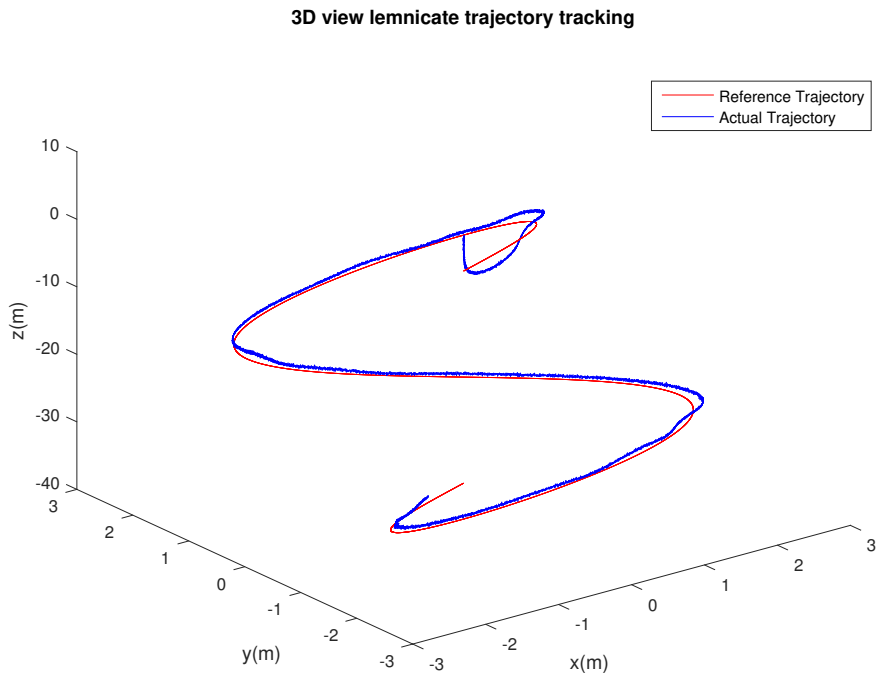


Figure 45. 3D Lemniscate Trajectory Tracking

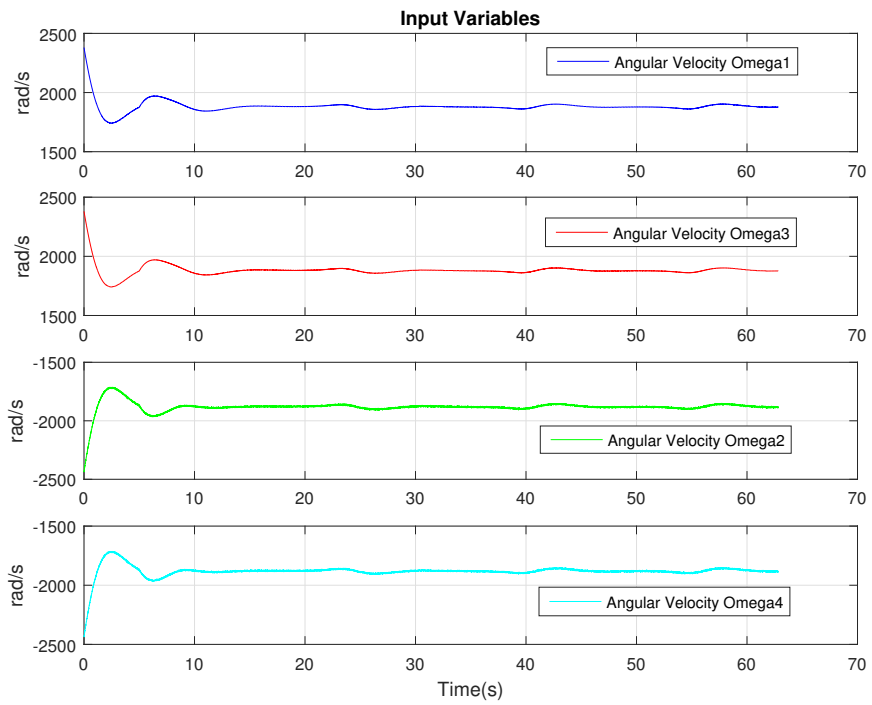


Figure 46. Plant Input Angular Speed

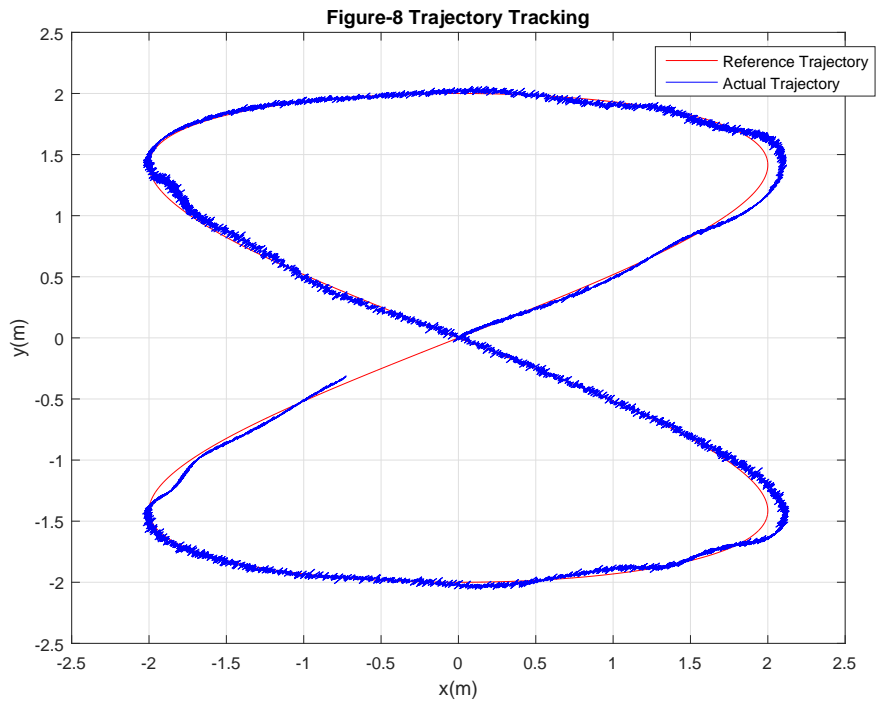


Figure 47. Lemniscate Trajectory Tracking xy-plane

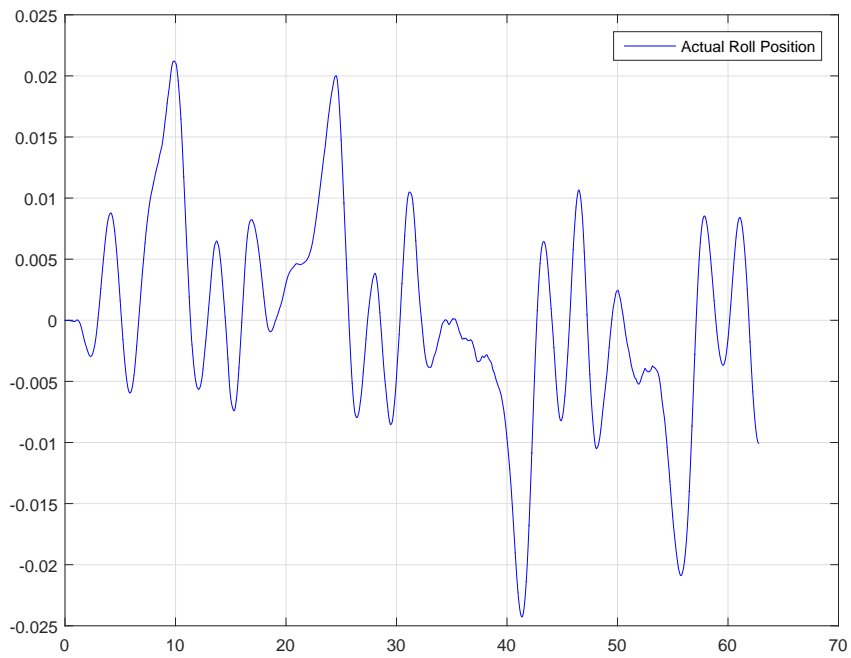


Figure 48. Roll Angle Variation

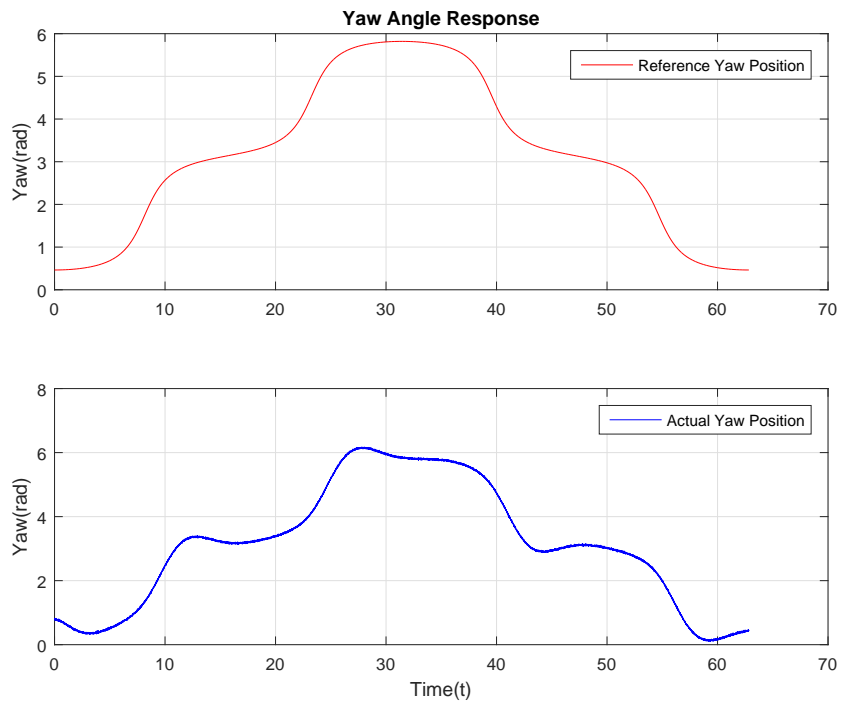


Figure 49. Yaw Angle Trajectory Tracking

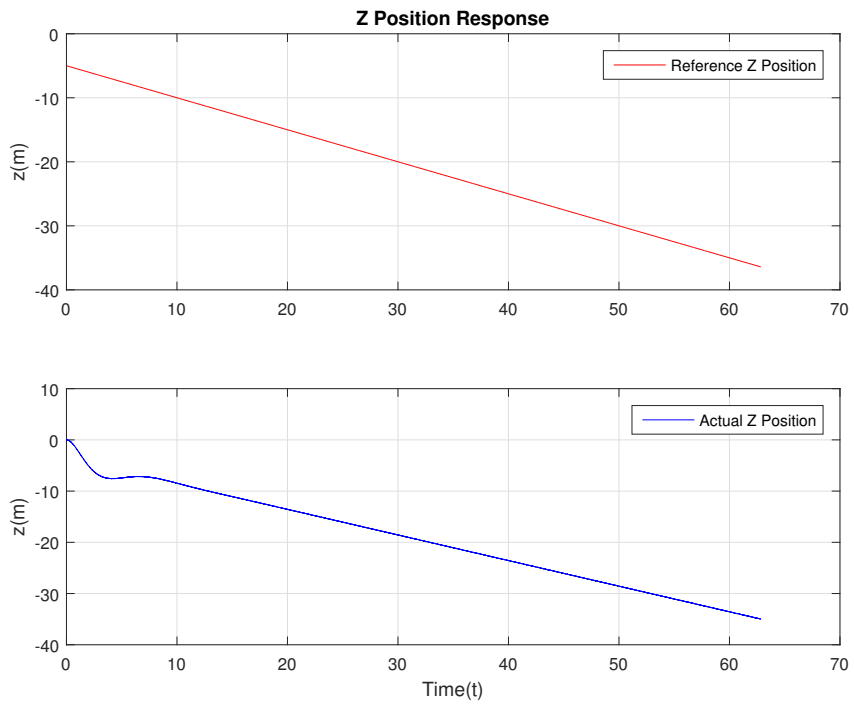


Figure 50. Z Direction Trajectory Tracking

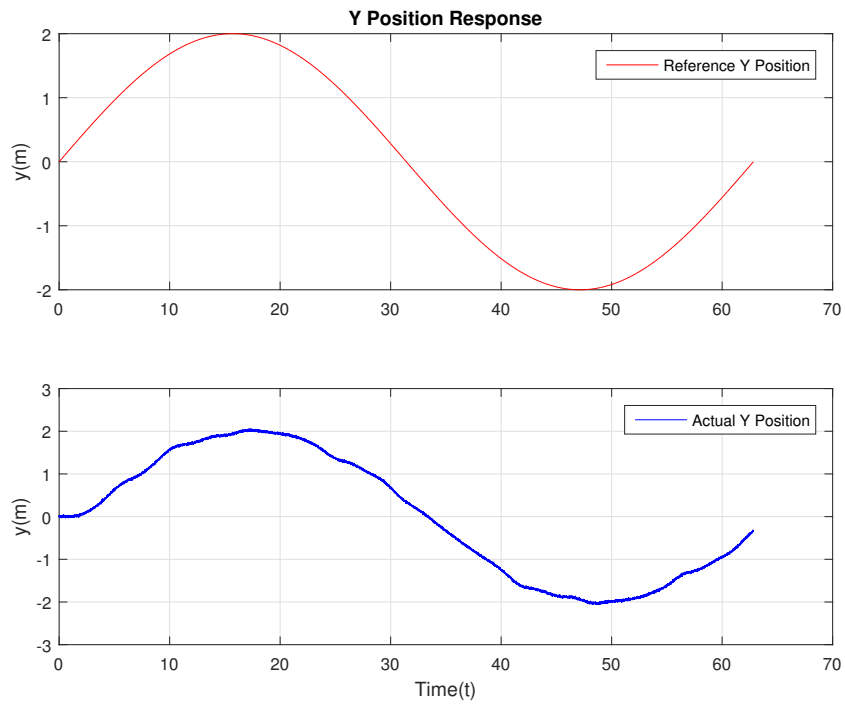


Figure 51. Y Direction Trajectory Tracking

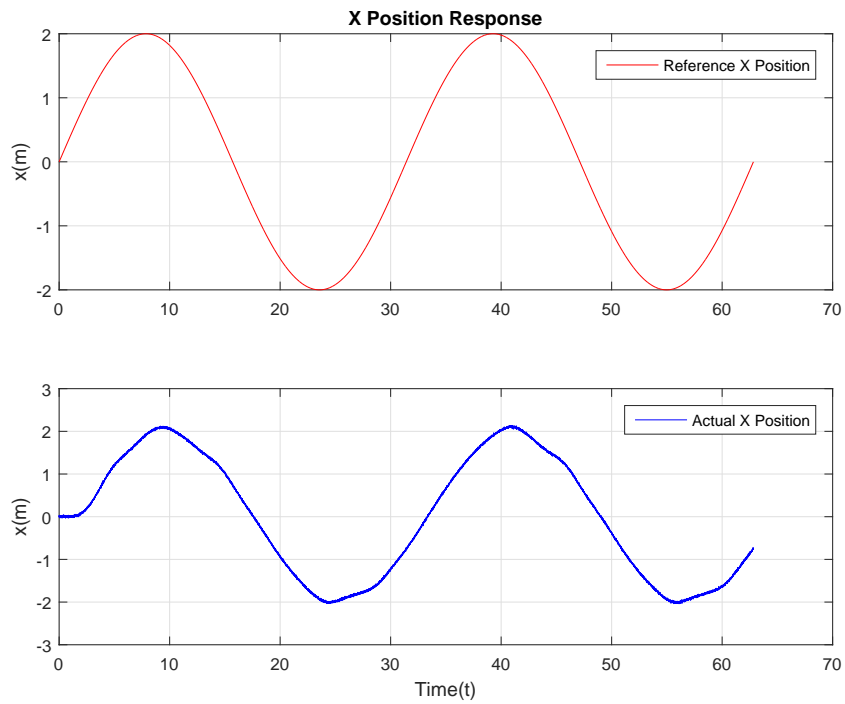


Figure 52. X Direction Trajectory Tracking

transient response for the x, y, z, ψ trajectory. The settling time and overshoot are acceptable.

List of References

- [1] J. S. Shamma and M. Athans, “Analysis of gain scheduled control for nonlinear plants,” in *Proceedings IEEE Transactions on Automatic Control*, August 1990, pp. 898–907.
- [2] D. Liberzon, *Switching in Systems and Control*. Springer, 2003.

CHAPTER 5

Conclusions

The quadcopter is a *MIMO* nonlinear system with four inputs and twelve state variables. Linearizing the nonlinear model using the Jacobian methods, we obtain the state space model around the desired equilibrium points. In the design of digital LQR controller, the altitude, x position, y position and yaw angle are tracked outputs and the twelve state variables are fed into the gain matrix. The weights choice for LQR regulator is tested using different pairs of Q and R matrices. The stability margin and settling time of the linearized system are guaranteed although with little overshoot. Techniques from linear control theory used to design the fixed gain digital tracking system. We extended the LQR regulator to a digital tracking system with the additional dynamics. When tracking references signals involving big variation in the yaw angle, we designed the gain-scheduled controller taking yaw angle as the scheduling variable, the adaptive gain matrices are calculated using the gain-scheduled algorithm. To demonstrate the effectiveness of the algorithm we developed, step references and trajectory tracking simulation we conducted. Also, we used the control laws tracking some trajectories in three dimensional spaces such as helix and lemniscate when there is no dramatic changes in the trajectory of yaw angle. The simulation result show that the controller we developed is able to successfully converge to the stable state and track the desired references.

This thesis provides a basis platform to design the trajectory tracking controllers for the Rolling Spider Drone. Although the control algorithm we developed gave good results, there are many parts of the design can be improved. The state-feedback regulator employed in the design is under the assumption that all 12 state

variables measured. However, in the practice, it is more necessary and smarter to measure some of the variables with the sensors mounted on the Drone. Usually, the Inertial Measurement Unit (IMU) which is a 6 axis-accelerometer-gyroscope equipment and pressure sensor can obtain the acceleration and angular rates in the body-frame and heights in the inertial-frame. Then the measured inertial-frame velocity and x,y positions can be obtained by transformation matrix R_{B2W} (6). The Euler angular rates are given by (7). The output feedback regulator can be designed by measuring the acceleration and angular rates in the body-frame and heights in the inertial-frame, an estimator can be used to obtain all the desired 12 state variables and feed them back to the controllers. Also, the physical experiment will need to performance with the real system Rolling Spider Drone to demonstrate the algorithms devised.

APPENDIX

Matlab Code

```
1 % clear all
2 % Origianl Model Digital
3 load('figure8_1degree.mat');
4
5 % Quadcopter Parameter defination
6 quad.Ct = 0.0107; %Thrust coefficient
7 quad.Cq = quad.Ct*sqrt(quad.Ct/2); %Torque coefficient
8 quad.r = 33/1000; %Rotor radius
9 quad.A = pi*quad.r^2; %Rotor Area
10 quad.rho = 1.184; %Density of air
11 quad.ka = quad.Ct*quad.rho*quad.A*quad.r^2;
12 quad.km = quad.Cq*quad.rho*quad.A*quad.r^3;
13 quad.g = 9.81;
14 quad.ix = 0.0686e-3;
15 quad.iy = 0.092e-3;
16 quad.iz = 0.1366e-3;
17 quad.d = 0.0624; %Distance from rotor to the
    center of Drone
18 quad.m = 0.068; %Mass of Drone
19 T = 0.005; %Sample time
20
21 [phia,gammaa] = c2d(zeros(4),eye(4),T); %Additional Dynamics
22 C = [eye(3,3) zeros(3,9);0 0 0 0 0 1 0 0 0 0 0 0 ];%Output state
    variables
23 D = zeros(4,4);
24
```

```

25 syms x1 x2 x3 x4 x5 x6 x7 x8 x9 x10 x11 x12 u1 u2 u3 u4;
26 % (x y z phi theta psi xdot ydot zdot p q r)
27
28 T_thrust = quad.ka*(u1+u2+u3+u4);
29 tau_x = quad.ka*quad.d*(u4-u2);
30 tau_y = quad.ka*quad.d*(u1-u3);
31 tau_z = quad.km*(u1-u2+u3-u4);
32
33 % Non-linear function
34 f1 = x7;
35 f2 = x8;
36 f3 = x9;
37 f4 = x10+sin(x4)*tan(x5)*x11+cos(x4)*tan(x5)*x12;
38 f5 = cos(x4)*x11+sin(x4)*x12;
39 f6 = (sin(x4)/cos(x5))*x11+(cos(x4)/cos(x5))*x12;
40 f7 = (-1/quad.m)*T_thrust*(cos(x4)*sin(x5)*cos(x6) + sin(x4)*sin(x6))
    ;%x7
41 f8 = (-1/quad.m)*T_thrust*(cos(x4)*sin(x5)*sin(x6)...
42     -sin(x4)*cos(x6));
43 f9 = quad.g - (1/quad.m)*T_thrust*(cos(x4)*cos(x5));
44 f10 = (tau_x+quad.iy*x11*x12-quad.iz*x11*x12)/quad.ix;
45 f11 = (tau_y-quad.ix*x10*x12+quad.iz*x10*x12)/quad.iy;
46 f12 = (tau_z+quad.ix*x10*x11-quad.iy*x10*x11)/quad.iz;
47
48 % Scheduled variables definition
49 yaw_equil = [0:1:360]*(pi/180); % Change yaw angle to radians
50 n = length(yaw_equil);
51 xbar = zeros(n,12);
52 x1bar = yaw_equil;
53
54 % Feedback matrix calculation
55 for i = 1:n

```

```

56 % Calculate the input and equilibrium points
57 % Hover stable state
58 omega = (sqrt((quad.m*quad.g)/(quad.ka*4)));
59 ubar = [omega^2;omega^2;omega^2;omega^2];
60 xbar(i,:) = [0;0;-2;0;0;yaw_equil(i);0;0;0;0;0;0];
61
62 % Jacobian function for the partial derivative of xbar and ubar
63 System.V = [x1, x2, x3, x4, x5, x6, x7, x8, x9, x10, x11, x12];
64 System.A = jacobian([f1,f2,f3,f4,f5,f6,f7,f8,f9,f10,f11,f12],System.V
    );
65 A(:,:,i) = double(subs(System.A,{x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,
    x12,u1,u2,u3,u4},{0,0,-2,0,0,yaw_equil(i),0,0,0,0,0,0,omega^2,
    omega^2,omega^2,omega^2}));
66 System.B = jacobian([f1,f2,f3,f4,f5,f6,f7,f8,f9,f10,f11,f12],[u1,u2,
    u3,u4]);
67 B(:,:,i) = double(subs(System.B,{x1,x2,x3,x4,x5,x6,x7,x8,x9,x10,x11,
    x12,u1,u2,u3,u4},{0,0,-2,0,0,yaw_equil(i),0,0,0,0,0,0,omega^2,
    omega^2,omega^2,omega^2}));
68
69 % Additional Dynamics
70 [m,l] = size(C);
71 [l,p] = size(B(:,:,i));
72 Aa = kron(eye(m),0);
73 Ba = kron(eye(m),1);
74 q = length(Aa);
75 Ad(:,:,i) = [A(:,:,i) zeros(l,q);Ba*C Aa];
76 Bd(:,:,i) = [B(:,:,i);zeros(q,p)];
77
78 % Choose Q and R weights for LQR Matrix
79 alpha = 1;
80 c1 = 1000;
81 c2 = 1000;

```

```

82 c3 = 10000;
83 c4 = 10000;
84 N = [c1 0 0 0;0 c2 0 0;0 0 c3 0;0 0 0 c4];% mapping the output for
      tracking
85 Cd = [zeros(4,12) N];%
86 Q = alpha*(Cd) '*Cd;
87 R = 0.001*eye(p);
88
89 % Digital LQR Feedback Matrix
90 [phid(:,:,i),gammad(:,:,i)] = c2d(Ad(:,:,i),Bd(:,:,i),T);
91 phiD = phid(:,:,i);
92 gammaD = gammad(:,:,i);
93 Kd(:,:,i) = dlqr(phiD,gammaD,Q,R);
94 K11(:,:,i) = Kd(:,1:12,i);
95 K22(:,:,i) = Kd(:,13:16,i);
96
97 % Parameters for Simulink
98 vK1(:,:,i)= K11(:,:,i);
99 vK2(:,:,i)= K22(:,:,i);
100 end

```

```

1 % controller function
2
3
4 function [xa,u] = fcn(v,vK1,vK2,xlbar,xbar,ubar,C,phia,gammaa)
5
6 % Define the dynamic K matrix
7 lpoints=xlbar;
8 N=length(xlbar);
9 inc=xlbar(2)-xlbar(1);
10
11 xa=v(1:4); % Track 1-4 output
12 x=v(5:16); % State Variables 12
13 y=C*x; % Output x,y,z,yaw
14 x_ref=v(17);
15 y_ref=v(18);
16 z_ref=v(19);
17 xlb=v(20); %v(20)=actual yaw position
18 r=[x_ref;y_ref;z_ref;xlb];%v(17:20);
19
20 xlb1=mod(xlb,2*pi);
21 if xlb >= 0 && xlb1 < lpoints(end)
22     n=floor((xlb-lpoints(1))/inc+1);% Divide inc then plus 1
23     alpha=(xlb-lpoints(n))/inc;
24     K1=(1-alpha)*vK1(:, :, n)+alpha*vK1(:, :, n+1);
25     K2=(1-alpha)*vK2(:, :, n)+alpha*vK2(:, :, n+1);
26     xbar_sl = (1-alpha)*xbar(n, :)' + alpha*xbar(n+1, :)' ;
27     % xbar_sl=xbar(n, :)' ;
28 else
29     % n=0;
30     % alpha=0;
31     K1=vK1(:, :, 1);
32     K2=vK2(:, :, 1);

```

```
33     xbar_sl=xbar(1,:);
34 end
35
36 % Controller code
37 e=r-y; % e=r-(y-C*xbar_sl); % r input 1 column
38 %u=K2*xa-K1*x; For Linear
39 u=K2*xa-K1*(x-xbar_sl)+ubar; % non-linear write here
40 xa=phia*xa+gammaa*e;
```

BIBLIOGRAPHY

- Argentim, L. M., Rezende, W. C., Santos, P. E., and Aguiar, R. A., "PID, LQR and LQR-PID on a Quadcopter Platform," in *Proceedings IEEE Informatics, Electronics and Vision (ICIEV)*, 2013, pp. 1–6.
- Ataka, A., Tnunay, H., and Inovan, R., "Controllability and observability analysis of the gain scheduling based linearization for uav quadrotor," in *International Conference on Robotics, Biomimetics, Intelligent Computational Systems (ROBIONETICS)*, 2013.
- Castelan, E. and Cury, J., "A reduced-order framework applied to linear systems with constrained controls," in *Proceedings IEEE Transactions on Automatic Control*, 1996, pp. 249–255.
- E.Lavetsky and Wise, K., *Robust and Adaptive Control with Aerospace Applications*. Springer-Verlag, 2013.
- ElKholy, H., "Dynamic modeling and control of a quadrotor using linear and non-linear approaches," Master's thesis, The American University in Cairo, 2014.
- Jafar, A., Ahmad, S., and Ahmed, N., "Mathematical modeling and control law design for 1 dof quadcopter flight dynamics," in *International Conference on Computing, Electronic and Electrical Engineering (ICE Cube)*, 2016, pp. 79–84.
- Kalman, R. and Bucy, R., "New results in linear filtering and prediction theory," in *Journal of Fluids Engineering*, 1961, pp. 95–108.
- Lavretsky, E. and Wise, K. A., *Robust and Adaptive Control with Aerospace Applications*. Springer, Inc, 2013.
- Liberzon, D., *Switching in Systems and Control*. Springer, 2003.
- Marquez, H. J., *Nonlinear Control Systems*. John Wiley and Sons, 2003.
- Massachusetts Institute of Technology. "Matlab toolbox for parrot rolling spider." [Online]. Available: <http://karaman.mit.edu/software.html>
- Nawawi, S., Ahmad, M., Osman, J., Husain, A., and Abdollah, M., "Controller design for two-wheels inverted pendulum mobile robot using pismc," in *4th Student Conference on Research and Development*, June 2006, pp. 194–199.
- Nilulescu, M., "Controlling a mobile robot along planned trajectories," in *CEAI*, 2005, pp. 18–24.

- Parrot SA. “Parrot minidrones rolling spider.” June 2008. [Online]. Available: <http://global.parrot.com/au/products/rolling-spider>
- Patel, K. and Varve, J., “Modeling, simulation and control study for the quadcopter UAV,” in *International Conference on Industrial and Information Systems (ICIIS)*, 2014, pp. 1–6.
- R.J.Vaccaro, *Digital Control A State-Space Approach*. McGraw-Hill,Inc, 1996.
- Sawyer, S., “Gain-scheduled control of a quadcopter UAV,” Master’s thesis, University of Waterloo, 2015.
- Shamma, J. S. and Athans, M., “Analysis of gain scheduled control for nonlinear plants,” in *Proceedings IEEE Transactions on Automatic Control*, August 1990, pp. 898–907.
- Stevens, B. L., Lewis, F. L., and N.Johnson, E., *Aircraft Control and Simulation: Dynamics, Controls Design, and Autonomous System, 3rd Edition*. Wiley-Blackwell, 2015.