Quadrotor Hovering Stability and Control

Mohamed Helal, Simon Lam
Department of Aerospace Engineering
Ryerson University
Toronto, ON, M5B 2K3, Canada

Abstract – Today, Quadrotors are widely used in different applications in both military and civil applications. First responders, photography, videography and research are some of the typical applications offered by these vehicles. In the last few years, extensive research regarding stabilizing, controlling and operating quadrotors was conducted. Quadrotors also known as quadcopters are multi-rotor unmanned aerial vehicles (UAVs) which uses four propellers to control their full pose (position and orientation). In this paper, multiple linear control techniques were applied to achieve quadrotor hovering stability and control. The first control approach was designing a traditional PID controller using SISO Approach and Auto-tuning approach. Linear Quadratic Regulator was the second controller designed to achieve stability. Two more controllers were designed to track and reject disturbance. Feed-forward tracking control was designed for constant tracking and Servomechanism control was designed for both constant and sinusoids tracking. Simulated results of all the controllers with the linearized system model and the nonlinear system model were presented and discussed.

Keywords PID, Linear Quadratic Regulator, Feed-forward tracking, Robust Servomechanism Control, Attitude

I. INTRODUCTION

Quadrotor also known as quadcopter is a mutlirotor unmanned Ariel vehicle (UAV) which has the capability of achieving vertical take-off and landing (VTOL), hovering in addition to executing extremely agile manoeuvres with relatively high speeds compared to traditional aircrafts and UAVs.

Quadrotors are composed of four propellers placed in cross configuration. Two of the propellers turn in a clockwise fashion while the other two turn in anti-clockwise fashion as shown in Figure 1. Full control of the quadrotor attitude, altitude and position could be achieved by varying the rotors rotational speeds with respect to each other. Vertical motion is attained by changing the rotational speed of the four propellers together. For executing a rotation around the pitch axis, the rotational speed of the first rotor increases and the speed of the third motor decreases or vice versa for the opposite rotation direction. For rotating around the roll axis, the rotational speed of the second rotor increases and the speed of the fourth motor decreases or vice versa for the opposite rotation direction. Changing the speeds of rotors 1 and 3 simultaneously with respect to rotors 2 and 4 and vice versa controls the rotation of the quadrotor around its yaw axis.

Stabilizing and control of quadrotors proposed a tough challenge that was investigated by many researchers in the past few years. The challenge is due to the complex nonlinear behaviour of the quadrotor dynamics. It is also a great tool for developing and testing new control approaches and techniques. Numerous control techniques were applied and developed starting form attitude control, altitude control, trajectory control and ending with voice and vision control. All the control techniques that were used and developed could be divided into two main categories. The first category is the linear control while the second category is the nonlinear control. Linear control techniques such as PID control, State feedback control and LQR control were presented extensively in multiple papers such as in Li [1], Bolandi et al. [2], Yacef et al. [3] and Argentim et al. [4]. Nonlinear quadrotor control such as feedback linearization technique, sliding mode, back-stepping and adaptive control was also investigated by Bouadi et al. [5], Al-Younes[6], Lee et al. [7] and Orsag et al. [8] who developed a hybrid control of quadrotors.

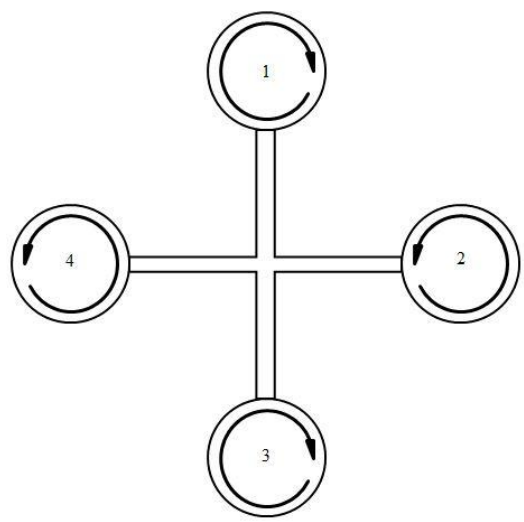


Figure 1: A Top View Diagram of a Quadrotor

The primary objective of this paper is to investigating multiple techniques for achieving quadrotor hovering stability and control. Hovering stabilization and control was achieved using linear control techniques applied to a linearized system dynamics of the quadrotor. The controllers designed were also simulated with the quadrotor's nonlinear system dynamics for evaluating the effectiveness of each controller. The control techniques presented in this paper are PID, LQR, Feed-forward tracking control and Robust Servomechanism control.

This paper is mainly divided into nine main sections along with the introduction. The second section is completely dedicated for providing the main theory, background and assumptions made for deriving a quadrotor model. This section includes setting the main frames, introduces the basics for constructing a full kinematic as well as Dynamic model of the quadrotor and defining the control input parameters in addition to model linearization. The third section provides a quick analysis of the quadrotor linearized model such as checking controllability, stabilizability, observability, detectability and the transmission zeros of the system.

The first method of control using a PID controller is presented in the fourth section. Two main approaches were investigated for tuning the PID controller; first is the auto-tuning method, second is the SISO approach method. Hovering stabilization of the quadrotor was also achieved and presented in the fifth section using the Linear Quadratic Regulator approach. Two methods for tracking control and disturbance rejection were presented in the sixth and the seventh sections of this paper. The first tracking approach is the implementation of the feed-forward tracking control while the second tracking approach is the Robust Servomechanism control approach. The feed-forward approach was applied for a constant input tracking unlike the servomechanism control which was not only applied for a constant input tracking but also applied for a sinusoid input tracking. Finally, simulation results of the linearized system dynamics as well as the nonlinear system dynamics were presented, discussed and concluded. Some of the possible future work was also included in the final section of the conclusion.

II. MODELLING

The dynamic behavior of the quadrotor is totally defined by a total of 12 parameters which could be divided into two sections. The first section is identifying the position and linear velocity of the center of gravity of the quadrotor which is given by 6 parameters, 3 parameters for position and 3 parameters for linear velocity $[X\ Y\ Z\ U\ V\ W]^T$. The second section is identifying the attitude and the angular velocity of the quadrotor which is defined by a total of 6 parameters, 3 parameters for orientation and 3 parameters for angular velocity $[\phi\ \theta\ \Psi\ P\ Q\ R]^T$.

Position vector: $\xi = [x\ y\ z]^T$

Euler angles: $= [\phi\ \theta\ \Psi]^T$

The following assumptions were applied for the simplification of the quadrotor model:

1. Quadrotor body and propellers are considered to be rigid bodies.
2. The quadrotor is symmetrical.
3. The quadrotor's center of mass is at the center of the body.
4. Thrust and drag moment from a propeller are proportional to the propeller speed.
5. The translational drag is ignored.
6. The ground effect is ignored.

The quadrotor model was divided into two parts. The first part is the kinematics model and the relation between the quadrotor frame and the inertial frame (Figure 2) as well as the velocities relation. The second part is the Dynamics model which was achieved using Newton-Euler formulation.

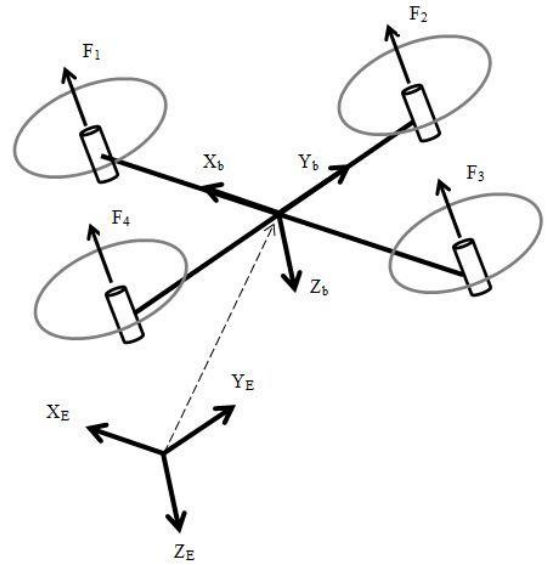


Figure 2: Quadrotor Coordinate Frames Diagram

Rotation between frames could be defined using Euler angles as following:

Using Euler angles ZYX

$$R_x = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\phi) & -\sin(\phi) \\ 0 & \sin(\phi) & \cos(\phi) \end{bmatrix}$$

$$R_y = \begin{bmatrix} \cos(\theta) & 0 & \sin(\theta) \\ 0 & 1 & 0 \\ -\sin(\theta) & 0 & \cos(\theta) \end{bmatrix}$$

$$R_z = \begin{bmatrix} \cos(\Psi) & -\sin(\Psi) & 0 \\ \sin(\Psi) & \cos(\Psi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$R_{zyx} = R_z.R_y.R_x$$
$$= \begin{bmatrix} c\Psi c\theta & c\Psi s\theta s\phi - c\phi s\Psi & s\Psi s\phi + c\Psi c\phi s\theta \\ c\Psi s\phi & c\Psi c\phi + s\Psi s\theta s\phi & c\phi s\Psi s\theta - c\Psi s\theta \\ -s\theta & c\theta s\phi & c\theta c\phi \end{bmatrix}$$

Where, c = cos and s = sin

The quadrotor frame – Earth frame velocities relation is given by the following matrix,

$$M = \begin{bmatrix} -\sin(\theta) & 0 & 1 \\ \cos(\theta)\sin(\phi) & \cos(\phi) & 0 \\ \cos(\theta)\cos(\phi) & -\sin(\phi) & 0 \end{bmatrix}$$

The previous relations summarize the kinematic model of the quadrotor and its relation to the earth's inertial frame. The next step is deriving the dynamics model using the following,

Newton Euler Approach

$$\begin{bmatrix} F \\ \tau \end{bmatrix} = \begin{bmatrix} mI_{3x3} & 0 \\ 0 & I \end{bmatrix} \begin{bmatrix} \dot{V} \\ \dot{\omega} \end{bmatrix} + \begin{bmatrix} \omega \times mV \\ \omega \times I\omega \end{bmatrix}$$

Where,

Gravitational force: $f_g = mg$

Single rotor's thrust: $f_i = \frac{1}{2}\rho C_T \omega_i^2 = b\omega_i^2$

Single rotor's drag: $D_i = \frac{1}{2}\rho C_D \omega_i^2 = d\omega_i^2$

Control inputs

$$U_1 = b(\omega_1^2 + \omega_2^2 + \omega_3^2 + \omega_4^2)$$
$$U_2 = bl(\omega_4^2 - \omega_2^2)$$
$$U_3 = bl(\omega_1^2 - \omega_3^2)$$
$$U_4 = d(\omega_1^2 + \omega_3^2 - \omega_2^2 - \omega_4^2)$$

Using the Newton-Euler Approach and the previous definitions, the complete quadrotor model could be determined as following,

$$\ddot{X} = (s\Psi s\phi + c\Psi c\phi s\theta)\frac{U_1}{m}$$
$$\ddot{Y} = (c\phi s\Psi s\theta - c\Psi s\theta)\frac{U_1}{m}$$
$$\ddot{Z} = -(c\theta c\phi)\frac{U_1}{m} + g$$
$$\ddot{\phi} = \dot{\theta}\dot{\Psi}\frac{I_{yy} - I_{zz}}{I_{xx}} - \frac{J_r}{I_{xx}}\Omega_r\dot{\theta} + \frac{U_2}{I_{xx}}$$
$$\ddot{\theta} = \dot{\phi}\dot{\Psi}\frac{I_{zz} - I_{XX}}{I_{yy}} + \frac{J_r}{I_{yy}}\Omega_r\dot{\phi} + \frac{U_3}{I_{yy}}$$
$$\ddot{\Psi} = \dot{\theta}\dot{\phi}\frac{I_{xx} - I_{yy}}{I_{zz}} + \frac{U_4}{I_{zz}}$$

Where,

Jr is the propeller's moment of inertia

$\Omega_r = \omega_1 - \omega_2 + \omega_3 - \omega_4$

In this paper we are mainly concerned by the quadrotor's hovering stability and control. So, we will consider the attitude as well as the altitude dynamics equations above. The states of this system is given by,

$$X = [Z\ \dot{Z}\ \phi\ \dot{\phi}\ \theta\ \dot{\theta}\ \Psi\ \dot{\Psi}]^T$$

Model Linearization

$$\dot{X} = \begin{bmatrix} f_1(x,u) \\ \cdot \\ \cdot \\ \cdot \\ f_n(x,u) \end{bmatrix}$$

Using Taylor series expansion to linearize the nonlinear equations

$$A = \left.\begin{bmatrix} \frac{df_1}{dx_1} & \cdots & \frac{df_1}{dx_n} \\ \vdots & \ddots & \vdots \\ \frac{df_n}{dx_1} & \cdots & \frac{df_n}{dx_n} \end{bmatrix}\right|_{\substack{x=xeq \\ u=ueq}}$$

$$B = \left.\begin{bmatrix} \frac{df_1}{du_1} & \cdots & \frac{df_1}{du_n} \\ \vdots & \ddots & \vdots \\ \frac{df_n}{du_1} & \cdots & \frac{df_n}{du_n} \end{bmatrix}\right|_{\substack{x=xeq \\ u=ueq}}$$

The linearized model is now expressed by the following equations,

$$\dot{X} = AX + BU$$
$$Y = CX + DU$$

At equilibrium the following is true,

$$\ddot{\phi} = 0, \ddot{\theta} = 0, \ddot{\Psi} = 0, \dot{\phi} = 0, \dot{\theta} = 0, \dot{\Psi} = 0, \phi = 0, \theta = 0,$$
$$\Psi = 0, \Omega_r = 0, w_1 = w_2 = w_3 = w_4 = 201.85\ rad/s$$

Table 1: Quadrotor Parameters

Parameter	Name	Value
l	Quadrotor arm length	0.232 m
b	Rotor thrust coefficient	3.13 x 10^{-5} N.s^2
d	Rotor drag coefficient	7.5 x 10^{-7} m.s^2
m	Total quadrotor mass	0.52 kg
I_{xx}	Moment of inertia about X axis	6.228 x 10^{-3} kg.m^2
I_{yy}	Moment of inertia about Y axis	6.228 x 10^{-3} kg.m^2
I_{zz}	Moment of inertia about Z axis	1.121 x 10^{-2} kg.m^2
J_r	Rotor Inertia	6 x 10^{-5} kg.m^2
Ω_{max}	Maximum rotor speed	279 rad/s

III. ANALYSIS

For applying the following techniques there are three main conditions that have to be satisfied. The first conditions the system has to be controllable/ stabilizable. The second condition the system has to be observable/ detectable. The third condition the transmission zeros of the system should not overlap the tracking or disturbance poles.

Controllability matrix

$$rank(Q) = rank([B\ AB\ A^2B\ \ldots\ A^{n-1}B])$$

The rank of the controllability matrix is 8 (full rank), which means that the system is controllable hence, stabilizable.

Observability matrix

$$rank(N) = rank\left(\begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix}\right)$$

The rank of the observability matrix is 8 (full rank), which means that the system is observable hence, detectable.

This system also doesn't have any transmission zeros at the origin or anywhere else.

The system's eigenvalues are all zero at the origin which means it is marginally stable (unstable including the nonlinearities).

IV. PID CONTROL

The first control approach used is the traditional PID controller design which is given by the following,

$$U_{PID} = K_p e(t) + K_i \int_0^t e(\tau) d\tau + K_d \frac{de(t)}{dt}$$

Using SISO Approach the following gains were determined,

Table 2: PID Gains using SISO Approach

	K_d	K_i	K_p
Altitude	-0.655	-0.006388	-0.37566
Pitch	7.475×10^{-3}	0	0.004397
Roll	7.475×10^{-3}	0	0.004397
Yaw	0.01382	0.001347	0.00792

Another method was also implemented using the auto-tuning approach. The results were compared and discussed below in the results section.

V. LQR CONTROL

The second control approach is the Linear Quadratic Regulator (LQR) approach, where the following function has to be minimized,

$$J = \int_0^\infty (X^T Q X + U^T R U)\, dt$$

$$U = -KX$$

Where,

$K = R^{-1}B^TP$

Optimal cost is $X_o^T P X_o$

And P is the solution to the Algebraic Riccati Equation

$$A^TP + PA - PBR^{-1}B^TP + Q = 0$$

VI. FEEDFORWARD TRACKING CONTROLLER

This controller is used for constant tracking where the controller is given by the following formula,

$$g = [K\ I]\begin{bmatrix} A & B \\ C & D \end{bmatrix}^{-1}\begin{bmatrix} 0 \\ I \end{bmatrix}$$

VII. ROBUST SERVOMECHANISM CONTROL

The last controller designed for tracking and disturbance rejection is the servomechanism control.

The conditions required for designing a servo-controller mechanism are:

1. The system matrices (A, B) have to be stabilizable.
2. The system matrices(C, A) have to be detectable.
3. The tracking and disturbance poles are not the same as (A, B, C, D) transmission zeros.
4. The control inputs are greater or equal to the number of outputs.

Using the tracking and disturbance poles to construct the following equations, (r = number of outputs)

$$\mathcal{C} = \begin{bmatrix} 0 & 1 & \cdots & 0 \\ \vdots & & \ddots & \vdots \\ & & & 1 \\ -m_p & -m_{p-1} & \cdots & -m_1 \end{bmatrix} \qquad \mathfrak{B} = \begin{bmatrix} 0 \\ 0 \\ \vdots \\ 1 \end{bmatrix}$$

$$\mathcal{D} = [1\ 0\ 0\ \dots\ ]$$

$$\mathcal{C}^* = blkdiag(\mathcal{C}_1, \dots, \mathcal{C}_r)$$

$$\mathfrak{B}^* = blkdiag(\mathfrak{B}_1, \dots, \mathfrak{B}_r)$$

$$\mathcal{D}^* = blkdiag(\mathcal{D}_1, \dots, \mathcal{D}_r)$$

The augmented system is given by the following equation

$$\begin{bmatrix} \dot{X} \\ \dot{\eta} \end{bmatrix} = \begin{bmatrix} A & 0 \\ \mathfrak{B}^*\mathcal{C} & \mathcal{C}^* \end{bmatrix}\begin{bmatrix} X \\ \eta \end{bmatrix} + \begin{bmatrix} B \\ \mathfrak{B}^*D \end{bmatrix} U + \begin{bmatrix} E & 0 \\ \mathfrak{B}^*F & -\mathfrak{B}^* \end{bmatrix}\begin{bmatrix} \omega \\ y_{ref} \end{bmatrix}$$

$$U = -K_x X - K_\eta \eta$$

$$A_{cl} = \begin{bmatrix} A & 0 \\ \mathfrak{B}^*\mathcal{C} & \mathcal{C}^* \end{bmatrix} \quad B_{cl} = \begin{bmatrix} B \\ \mathfrak{B}^*D \end{bmatrix}$$

Where, $(A_{cl} - B_{cl}K)$ is stable. And $K = [K_x\ K_\eta]$

VIII. RESULTS AND DISCUSSION

The following results are divided into two sections. The first section is the simulated response of the designed controllers with the linearized quadrotor model. The second section is the simulated response of the same controllers with the quadrotor nonlinear dynamics.

1. Linear System

PID Auto-tuning method

The stabilizing response of the PID auto-tuning method on the four states (Altitude, Pitch, Roll and Yaw) of the linear model is shown in the plots below.

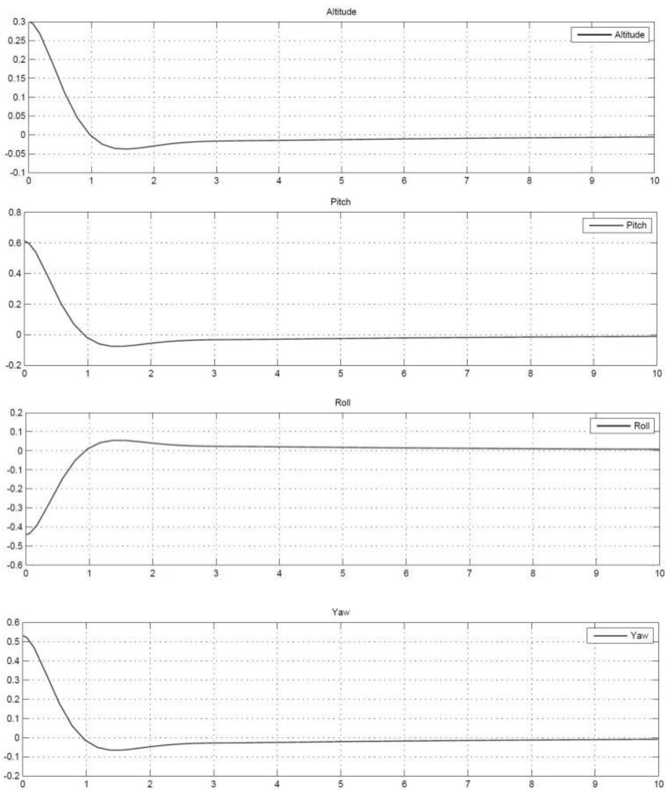


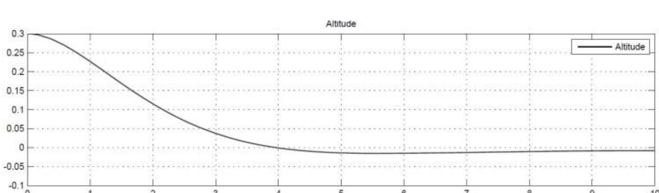
Figure 3: Linear system Auto-tuning PID Stabilization

The previous simulation results were generated by setting the system initial conditions to,

- Altitude (Z) = 0.3m
- Pitch, Roll and Yaw (ϕ, θ, ψ) = $[30\ -25\ 30]^T$ deg.
  = $[0.61\ -0.44\ 0.53]^T$ rad.

PID SISO Approach

The following is the linear system response to the designed SISO Approach PID controller. The response of the system is as twice as slow as the previous method but with almost no undershoot. It was generated using the same initial conditions mentioned above.



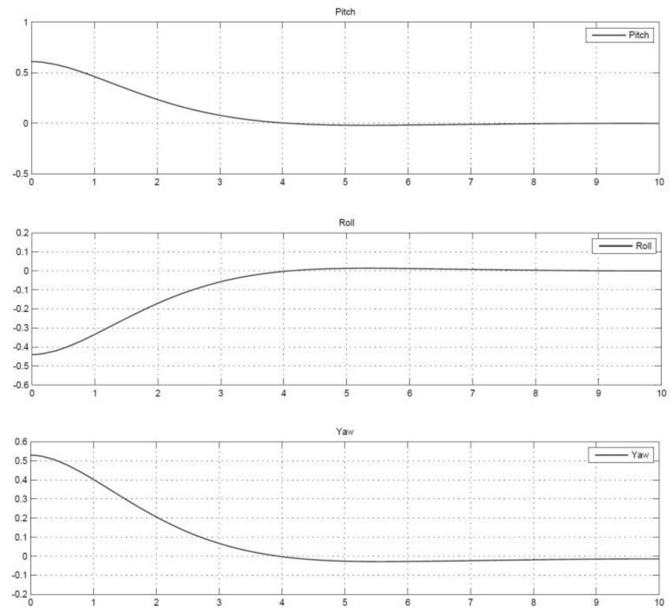


Figure 4: Linear system SISO Approach PID Stabilization

LQR Control

The following response is the linear quadratic regulator (LQR) effect on stabilizing the linear system. It is noticed that the response time is slightly better in for the altitude but very fast for the attitude stabilization. The initial conditions for this simulation is Z = 0.5m and the same attitude as mentioned before.

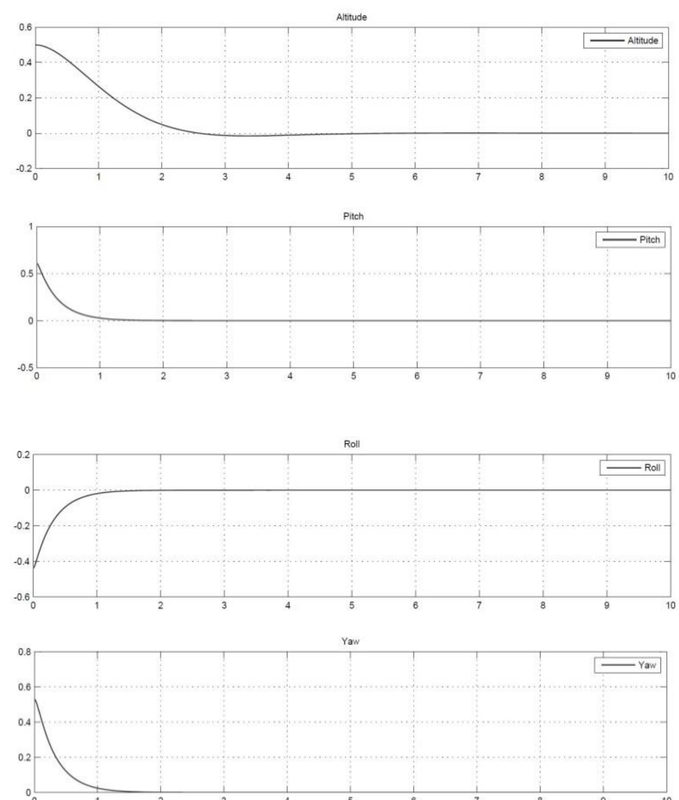


Figure 5: Linear System Stabilization using LQR

Feed-forward Control

The following is the response of the linear system with the feed-forward tracking control to a constant input to the system. The initial conditions of the system was set to zero in this case and the constant inputs were set to the Z = 0.6, ϕ = 0.5 rad, θ = -0.3 rad and ψ = 0.2 rad.

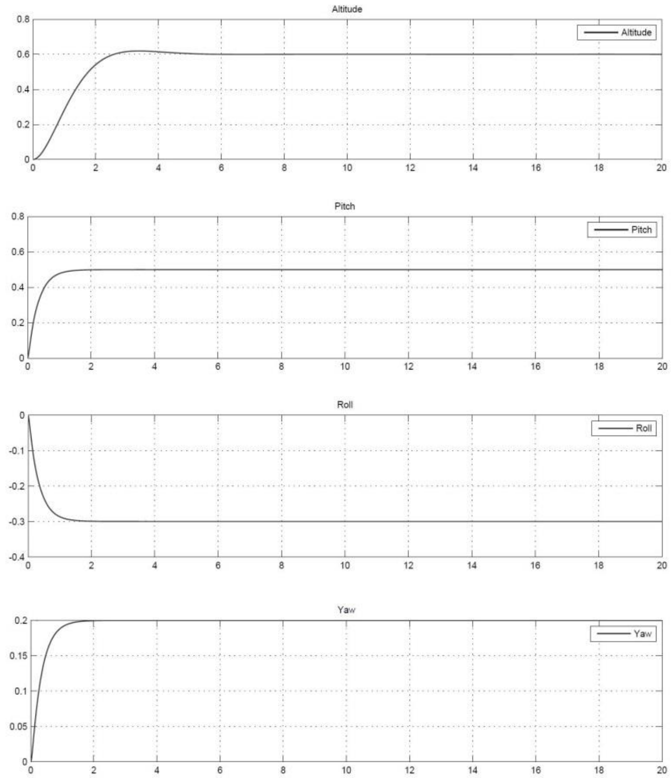
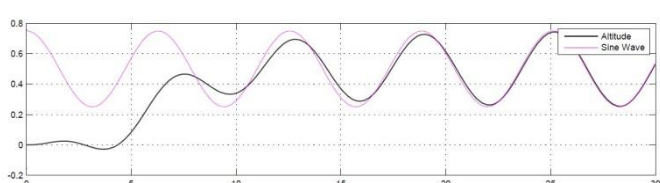


Figure 6: Linear System Response with Feed-forward Tracking Control

As shown in the plots, the system has great response and tracking characteristics. The altitude response is about 3 seconds while the attitude response is about 1 second.

Servomechanism Control

The following plots are the linear system responses with a servomechanism control for constant and sinusoid tracking. The altitude input is a sine wave with an amplitude of 0.25 and constant bias of 0.5. The input to the attitude angles are constants of value ϕ = 0.2 rad, θ = -0.3 rad and ψ = 0.25 rad. In this case the sine wave input was only applied to the altitude but it could be applied to any attitude angle as well or to all of them.



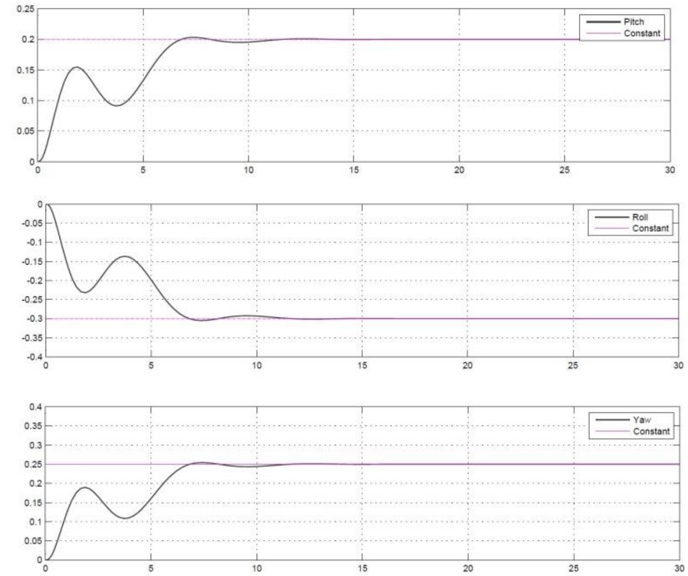


Figure 78: Linear System Response with Servomechanism Control

2. Nonlinear system

In this subsection of the results and discussion section, all the previous controllers stabilizing and tracking effects were simulated with the nonlinear system dynamics. The effect of nonlinearities of the system could be well noticed from the following responses. It is well noticed that all the controllers could stabilize the nonlinear quadrotor model but it is also noticed that the system responses were greatly affected by the system nonlinearities as shown in the following plots.

PID Auto-tuning

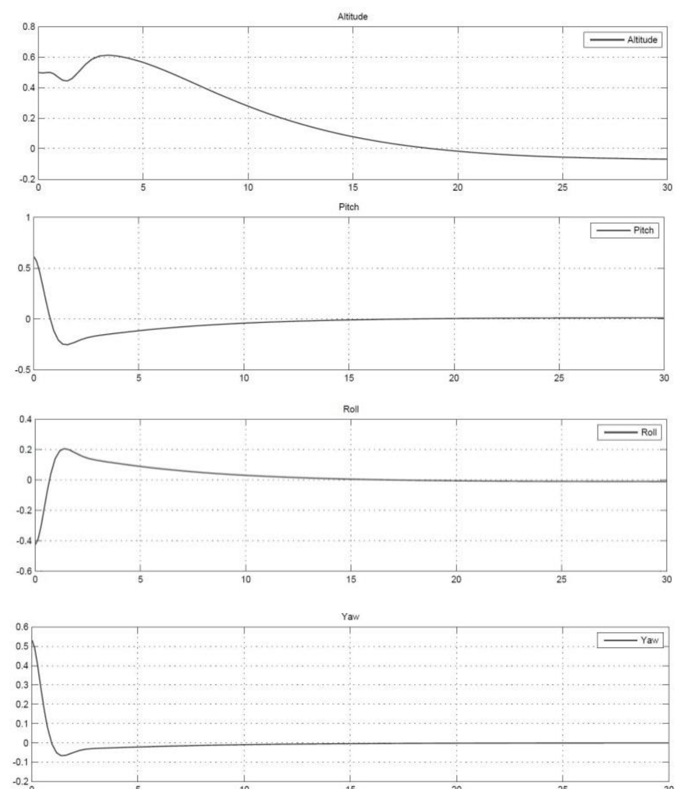


Figure 9: Nonlinear system Auto-tuning PID Stabilization

PID SISO Approach

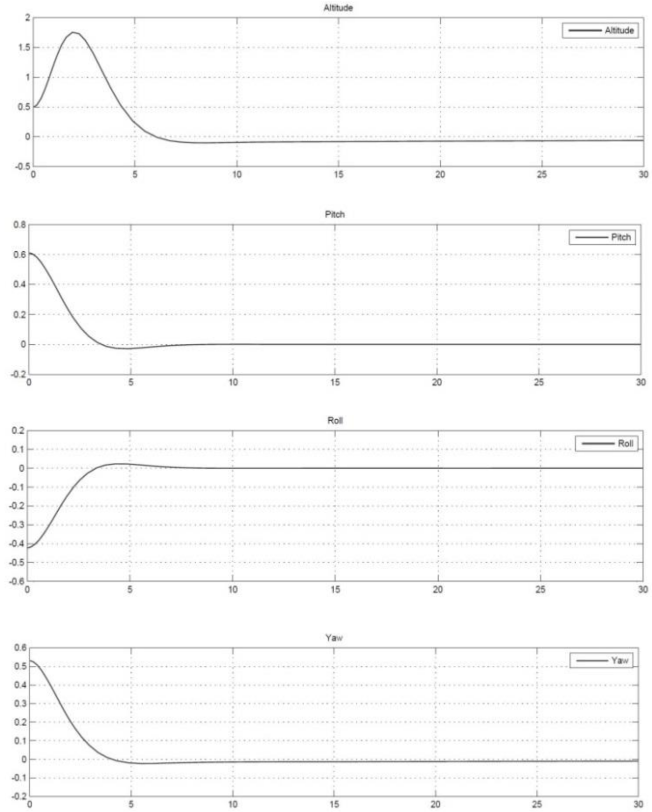


Figure 10: Nonlinear system SISO Approach PID Stabilization

Feed-forward Control

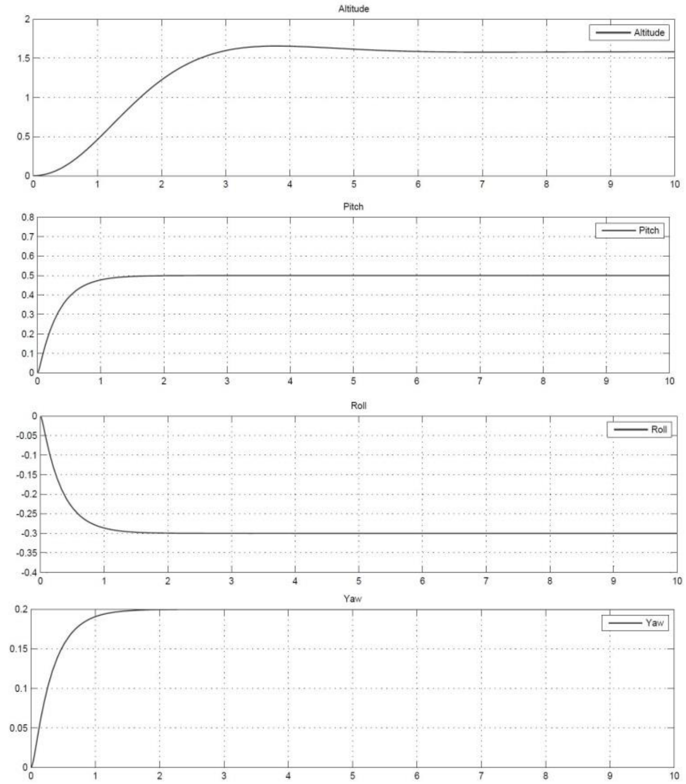


Figure 12: Nonlinear System Response with Feed-forward Tracking Control

LQR Control

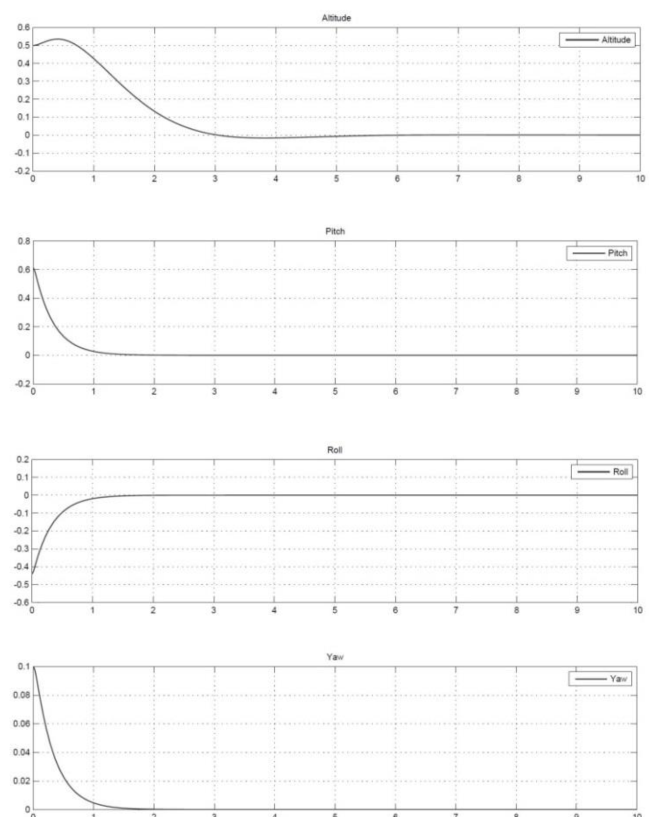


Figure 11: Nonlinear System Stabilization using LQR

Servomechanism Control

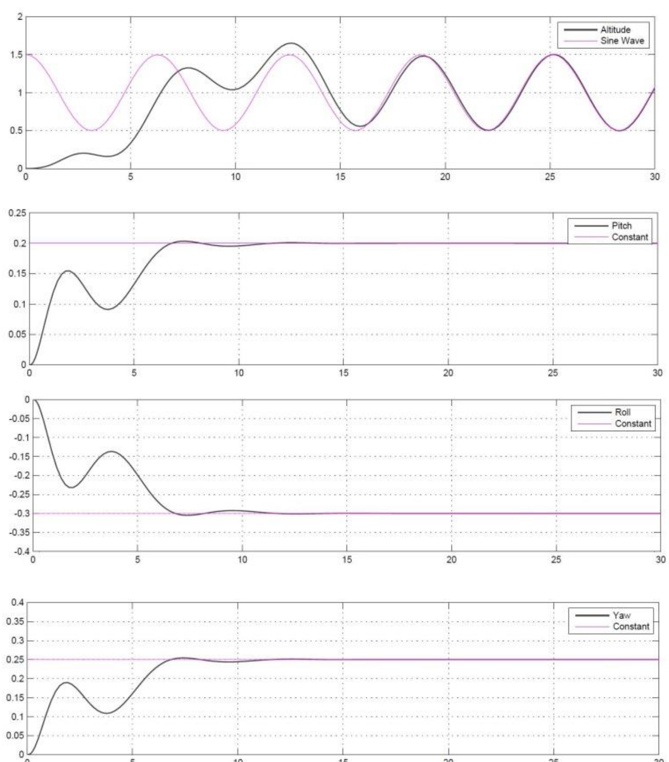


Figure 13: Nonlinear System Response with Servomechanism Control

IX. CONCLUSION

In this paper, a complete model analysis and hovering control of a quadrotor was presented. Multiple linear control techniques were implemented and simulated on both the linearized quadrotor model and the nonlinear quadrotor dynamics. The control techniques used in this paper started by applying a PID controller which was tuned using auto-tune method as well as SISO Approach. The second presented control technique is the Linear Quadratic Regulator (LQR) approach. Two more approaches were presented specifically for tracking and disturbance rejection. Feed-forward tracking control approach was used for constant tracking while Robust Servomechanism control was used for both constant and sinusoid tracking. All the simulation results were simulated for the linearized model and compared to the response of the nonlinear dynamics of the quadrotor. All the methods were successfully capable of stabilizing the quadrotor having various yet good response characteristics.

The future work suggested could be summarized as following:

First, the quadrotor dynamics model has to be refined by including the motor and rotor dynamics in the total model. Second, all the controller parameters have to be fine-tuned for achieving the best possible responses within the quadrotor capabilities. Third, an adequate filter has to be designed for clearing the measured signals noises. Fourth, a full state observer as well as reduced state observer has to be designed and imbedded in the simulation. Finally, experimental verification of the designed controllers, filters and observers has to be conducted.

References

[1] Li, J., & Li, Y. (2011). Dynamic Analysis and PID Control for a Quadrotor. IEEE, 573-578.

[2] Bolandi, H., Rezaei, M., Mohsenipour, R., Nemati, H., & Smailzadeh, S. (2013). Attitude Control of a Quadrotor with Optimized PID Controller. Intelligent Control Automation, 335-342.

[3] Yacef, F., Boudjedir, H., Khebbache, H., & Bouhali, O. (2012). From PID to State Feedback Attitude Atabilization of a Quadrotor UAV. International Journal of Information Technology, Control and Automation.

[4] Argentim, L. M., Rezende, W. C., Santos, P. E., & Aguiar, R. A. (2013). PID, LQR and LQR-PID on a Quadcopter Platform. IEEE.

[5] Bouadi, H., Cunha, S., Drouin, A., & Mora-Camino, F. (2011). Adaptive Sliding Mode Control for Quadrotor Attitude Stabilization and Altitude Tracking . IEEE, 449-455.

[6] Al-Younes, Y. M., Al-Jarrah, M. A., & Jhemi, A. A. (2010). Linear vs. Nonlinear Control Techniques for a Quadrotor Vehicle. ISMA.

[7] Lee , K. U., Kim, H. S., Park, J. B., & Choi, Y. H. (2012). Hovering Control of a Quadrotor. International Conference on Control, Automation and Systems, 162-167.

[8] Orsag, M., & Bogdan, S. (2009). Hybrid Control of Quadrotor. IEEE, 1239-1244.

[9] Ilhan, I., & Karakose, M. (2013). Type-2 Fuzzy Based Quadrotor Control Approach. IEEE.

Extra Derivations

$$A = \left.\begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & \dfrac{\cos\theta\sin\phi\, U_1}{m} & \dfrac{\sin\theta\cos\phi\, U_1}{m} & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & a_1 & a_2 \\ 0 & 0 & 0 & 0 & 0 & a_3 & 0 & a_4 \\ 0 & 0 & 0 & 0 & 0 & a_5 & a_6 & 0 \end{bmatrix}\right|_{\substack{x=xeq \\ u=ueq}}$$

$$B = \left.\begin{bmatrix} 0 & 0 & 0 & 0 \\ \dfrac{-\cos\theta\cos\phi}{m} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & \dfrac{1}{I_{xx}} & 0 & 0 \\ 0 & 0 & \dfrac{1}{I_{yy}} & 0 \\ 0 & 0 & 0 & \dfrac{1}{I_{zz}} \end{bmatrix}\right|_{\substack{x=xeq \\ u=ueq}}$$

Where,

$$a_1 = \dot{\Psi}\frac{I_{yy} - I_{zz}}{I_{xx}} - \frac{J_r}{I_{xx}}\Omega_r$$

$$a_2 = \dot{\theta}\frac{I_{yy} - I_{zz}}{I_{xx}}$$

$$a_3 = \dot{\Psi}\frac{I_{zz} - I_{XX}}{I_{yy}} + \frac{J_r}{I_{yy}}\Omega_r$$

$$a_4 = \dot{\phi}\frac{I_{zz} - I_{XX}}{I_{yy}}$$

$$a_5 = \dot{\theta}\frac{I_{xx} - I_{yy}}{I_{zz}}$$

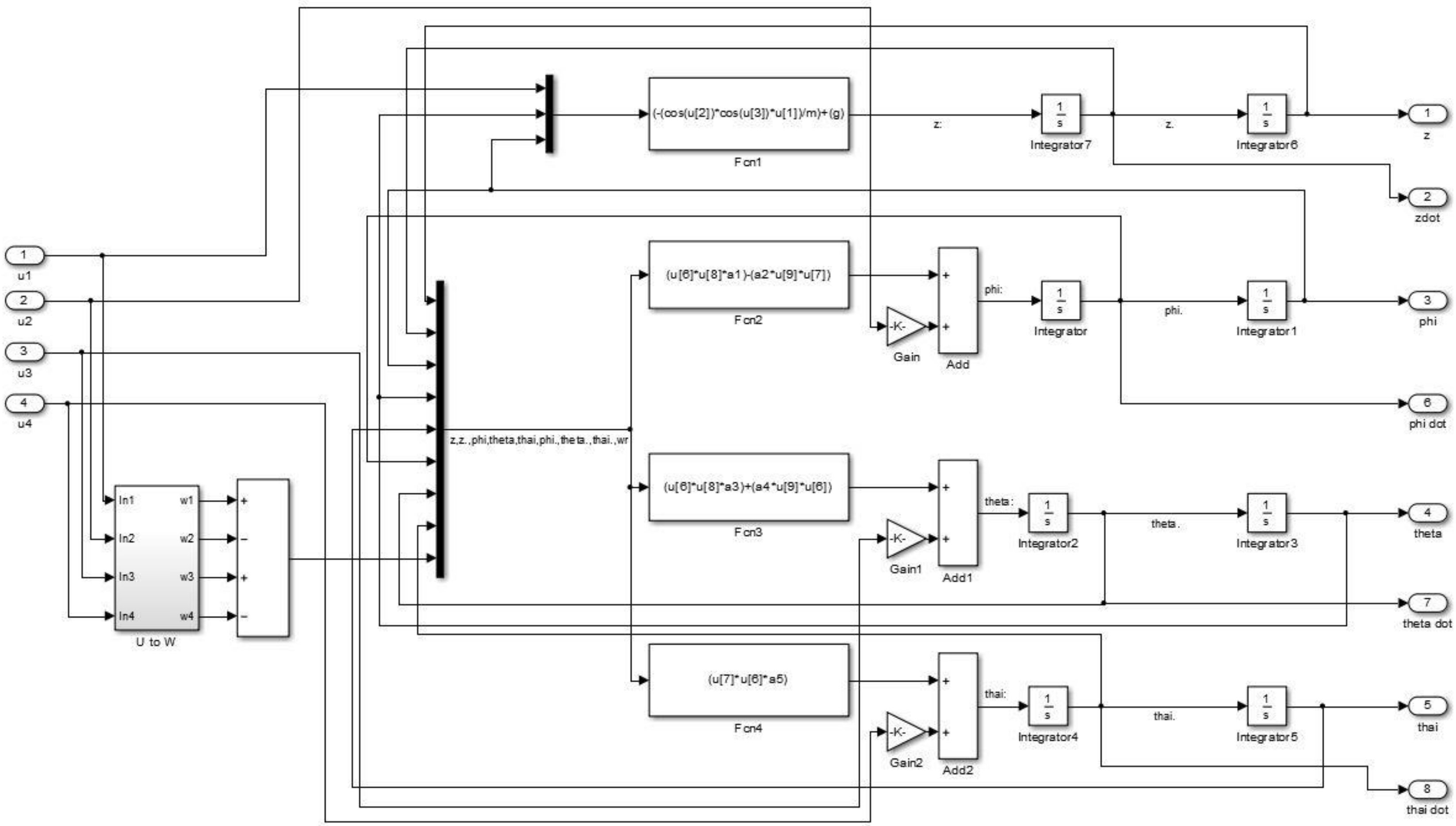
$$a_6 = \dot{\phi}\frac{I_{xx} - I_{yy}}{I_{zz}}$$

The linearized Model

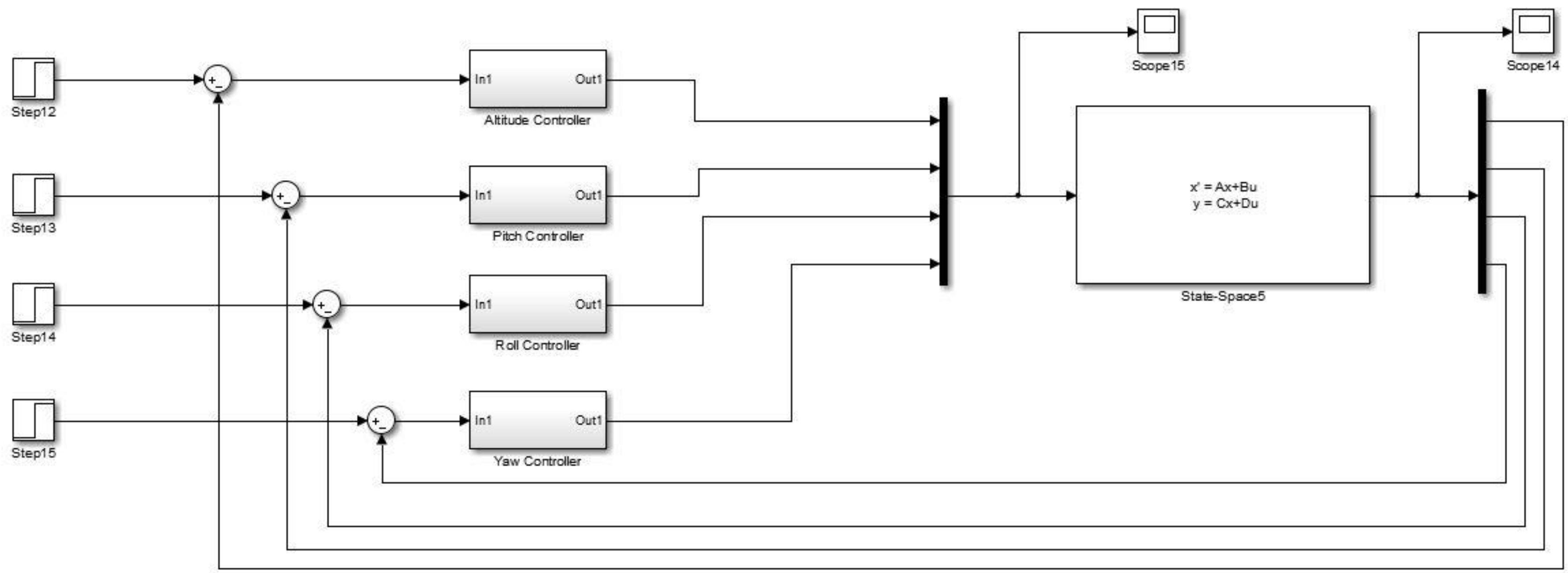
$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} B = \begin{bmatrix} 0 & 0 & 0 & 0 \\ -1.923 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 160.565 & 0 & 0 \\ 0 & 0 & 160.565 & 0 \\ 0 & 0 & 0 & 89.206 \end{bmatrix} \boldsymbol{C} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

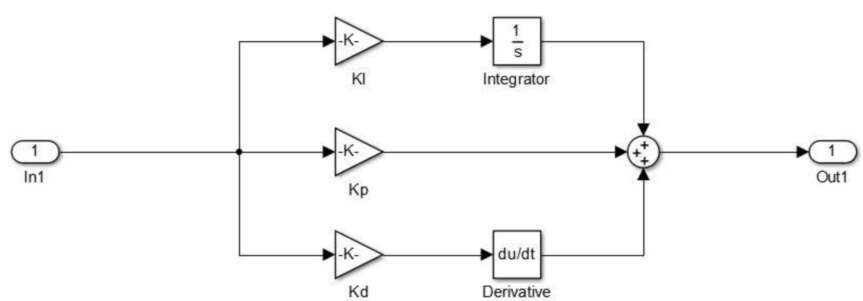
$$\boldsymbol{D} = [\boldsymbol{0}]$$

Simulink Nonlinear Quadrotor Dynamics

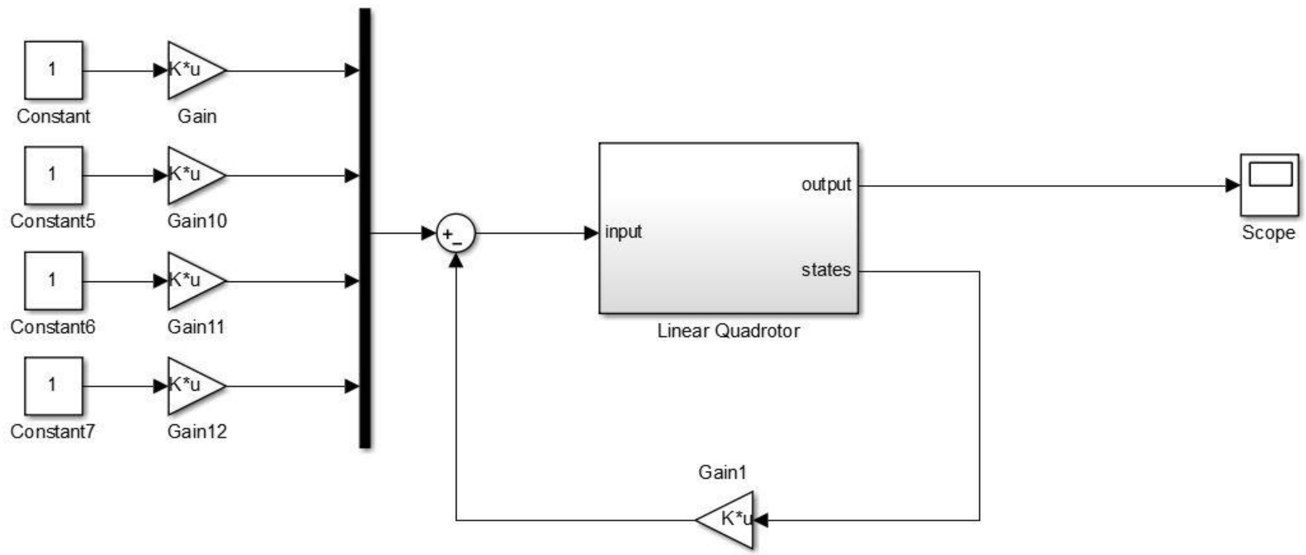


Simulink PID Control and PID Controller

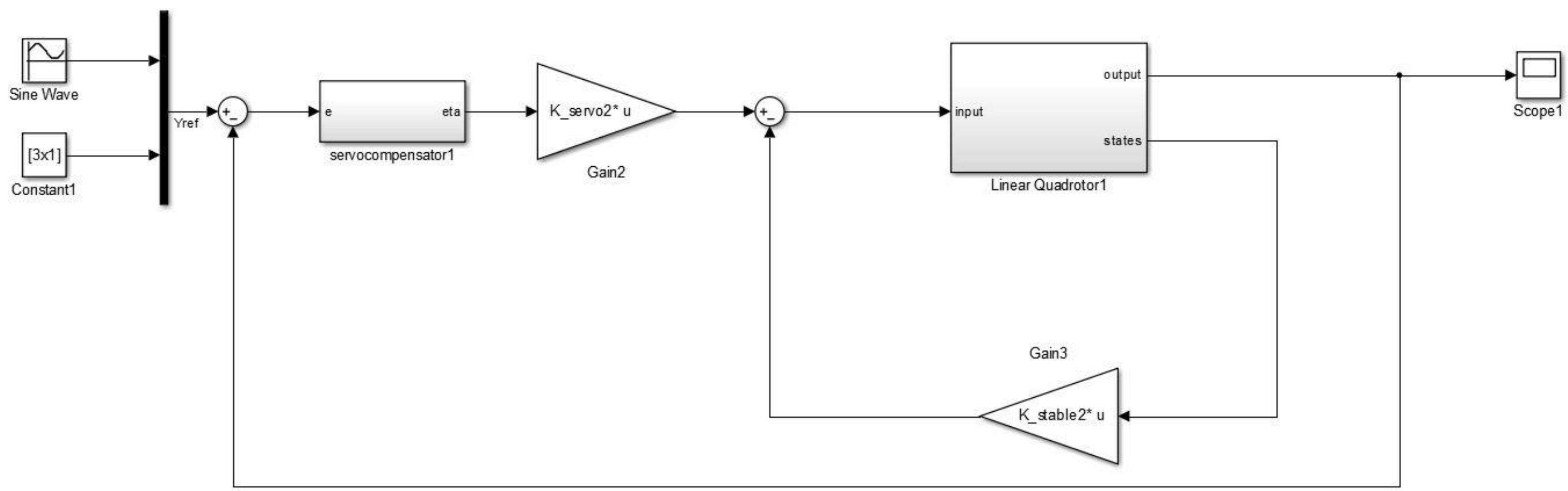




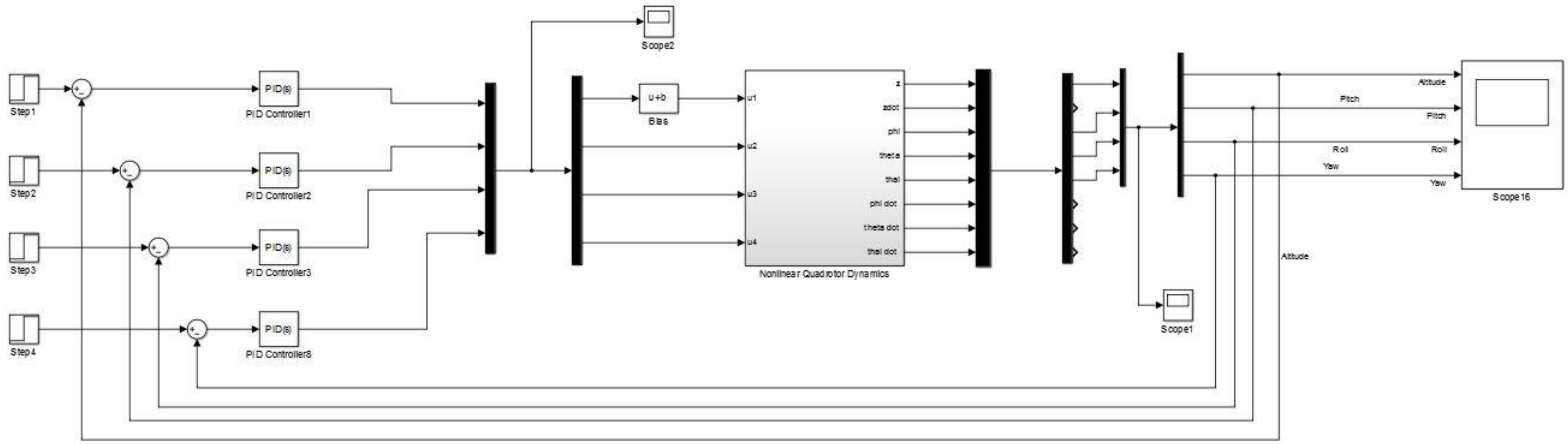
Feed-forward Tracking Control



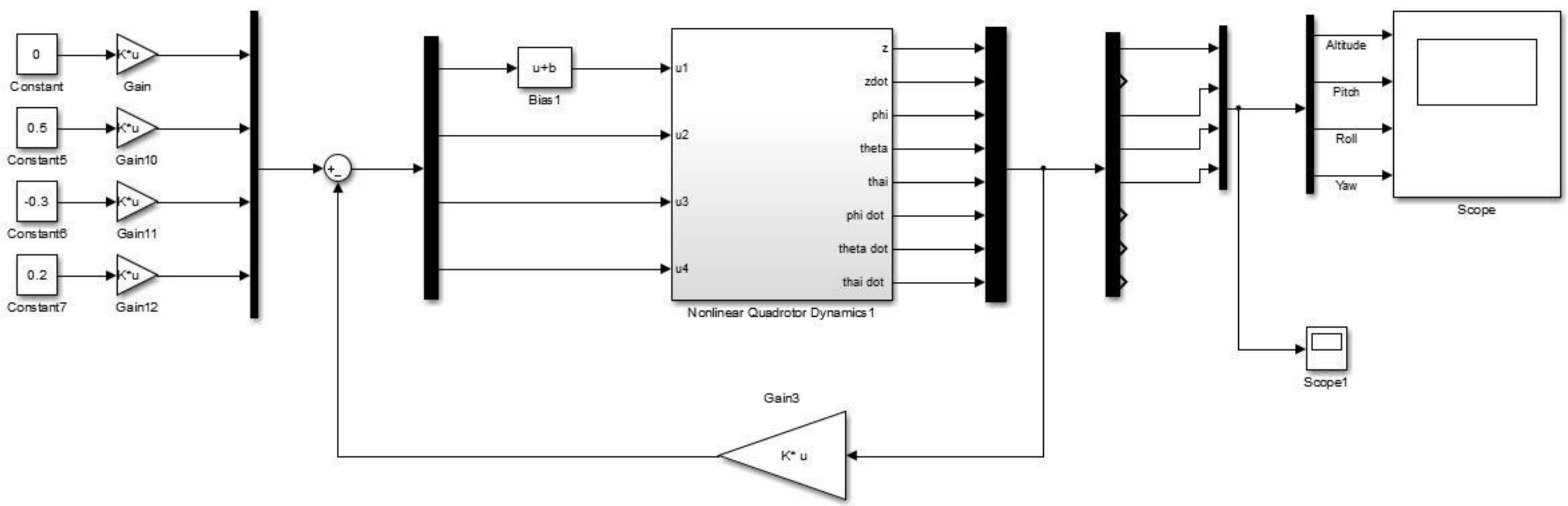
Robust Servomechanism Control



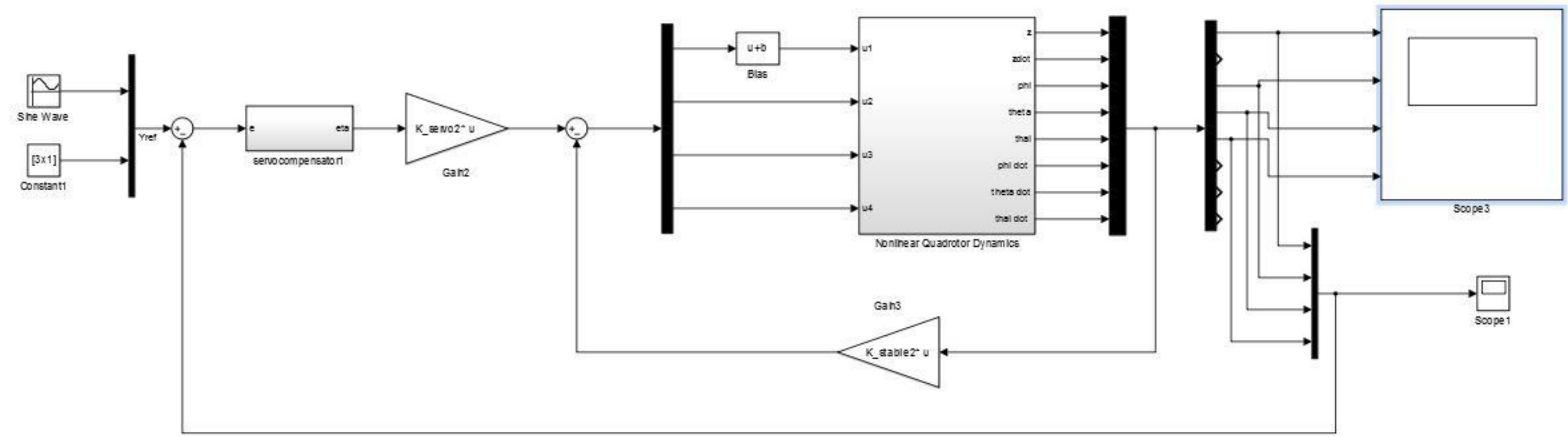
PID Control for Nonlinear System



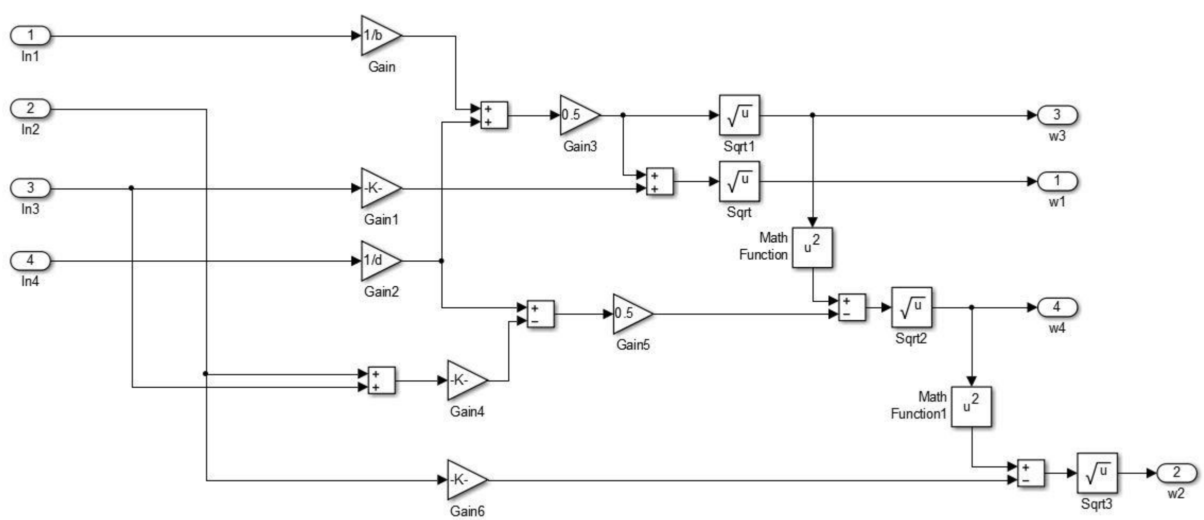
Feed-forward Tracking Control for Nonlinear System



Robust Servomechanism Control for Nonlinear System



Conversion from Control Inputs to Motor Speeds



Matlab Code

```
% -----------------------------
% Mohamed Helal
% MASC W2014
% Advanced Control Systems
% Project-Quadrotor Parameters
%------------------------------

l = 0.232;                    % quadrotor arm length
b = 3.13*10^-5;               % rotor thrust coefficient
d = 7.5*10^-7;                % rotor drag coefficient
m = 0.52;                     % quadrotor mass
Ixx = 6.228*10^-3;            % moment of inertia about x axis
Iyy = 6.228*10^-3;            % moment of inertia about y axis
Izz = 1.121*10^-2;            % moment of inertia about z axis
Jr = 6*10^-5;                 % Rotor inertia
wmax = 279;                   % maximum rotor speed

g = 9.81;                     % gravitational acceleration

a1 = (Iyy-Izz)/Ixx;
a2 = (Jr)/Ixx;
a3 = (Izz-Ixx)/Iyy;
a4 = (Jr)/Iyy;
a5 = (Ixx-Iyy)/Izz;

U1_initial = m*g;             % input 1 at initial condition

% Linearized system
%--------------------
A = zeros(8,8);
A(1,2) = 1;
A(3,6) = 1;
A(4,7) = 1;
A(5,8) = 1;

B = zeros(8,4);
B(2,1) = -1/m;
B(6,2) = 1/Ixx;
B(7,3) = 1/Iyy;
B(8,4) = 1/Izz;

C = [1 0 0 0 0 0 0 0;
     0 0 1 0 0 0 0 0;
     0 0 0 1 0 0 0 0;
     0 0 0 0 1 0 0 0]

D = zeros(4,4);

% Analysis
%------------

tzero(A,B,C,D)

ctrb_r = rank(ctrb(A,B))
obsv_r = rank(obsv(A,C))

% LQR Controller Design
%----------------------
```

```
Q = diag([1,0.1,1,1,1,0.1,0.1,0.1]);
R = eye(4);

K = lqr(A,B,Q,R);


% converting state space to transfer function
%--------------------------------------------

sys = ss(A,B,C,D)

TF = tf(sys)

Alt = TF(1,1)
Pitch = TF(2,2)
Roll = TF(3,3)
Yaw = TF(4,4)


% Decoupling state space
%-----------------------

%Altitdue
%---------
A1 = [0 1;0 0]
B1 = [0 ;-1.9231]
C1 = [1 0]
D1 = [0]

%Roll
%----

A2 = [0 1;0 0]
B2 = [0 ;160.5652]
C2 = [1 0]
D2 = [0]

%Pitch
%------

A3 = [0 1;0 0]
B3 = [0 ;160.5652]
C3 = [1 0]
D3 = [0]

% Yaw
%----

A4 = [0 1;0 0]
B4 = [0 ;89.2061]
C4 = [1 0]
D4 = [0]




% servomechanism tracking control
%---------------------------------
```

```
%track constant
%--------------

C_star = blkdiag(0,0,0,0)
B_star = blkdiag(1,1,1,1)
D_star = blkdiag(1,1,1,1)


Acl = [A zeros(8,4);B_star*C C_star]
Bcl = [B; B_star*D]

Q_servo = 0.01*eye(12);
R_servo = eye(4);

K_servo_total = lqr(Acl,Bcl,Q_servo,R_servo)

K_servo  = K_servo_total(1:4,9:12)
K_stable = K_servo_total(1:4,1:8)

%track constant and sine wave
%----------------------------
CC = [0 1 0;0 0 1;0 -1 0];
BB = [0 0 1]';
DD = [1 0 0];

C_star2 = blkdiag(CC,CC,CC,CC)
B_star2 = blkdiag(BB,BB,BB,BB)
D_star2 = blkdiag(DD,DD,DD,DD)

Acl2 = [A zeros(8,12);B_star2*C C_star2]
Bcl2 = [B; B_star2*D]

Q_servo2 = 0.01*eye(20);
R_servo2 = eye(4);

K_servo_total2 = lqr(Acl2,Bcl2,Q_servo2,R_servo2)

K_servo2  = K_servo_total2(1:4,9:20)
K_stable2 = K_servo_total2(1:4,1:8)

% separating Gains
%-----------------

K_alt = K(1,1:2)
K_pitch = [K(2,3) K(2,6)]
K_roll =[K(3,4) K(3,7)]
K_yaw = [K(4,5) K(4,8)]

% feedforward tracking controller
%--------------------------------




g1 =[K_alt 1]*inv([A1 B1;C1 D1])*[0;0;1]
g2 =[K_pitch 1]*inv([A2 B2;C2 D2])*[0;0;1]
g3 =[K_roll 1]*inv([A3 B3;C3 D3])*[0;0;1]
g4 =[K_yaw 1]*inv([A4 B4;C4 D4])*[0;0;1]
```