MECH-7672 Flight Dynamics and Control of UAVs Course Project

Atilla Saadat – 104411786

University of Windsor

Department of Mechanical, Automotive, and Materials Engineering

Table of Contents

List of Figures

List of Tables

Abstract

The Quanser helicopter system is a complex 3 degrees of freedom dynamic system used in training students in the design of system controllers. Various methods for system dynamics control is explored, such as Linear Quadratic Regulator (LQR), Linear Quadratic Servo (LQ Servo), Linear Quadratic Estimation (LQE), and Dynamics Output Feedback (DOFB), to compute, tune, and analyze the system response for each controller. The resulting methods are then imported a Simulink model of the Quanser system, simulating the response of the system from desired inputs and tuned control parameters. The results showed that from the calculated tuning parameters, the Quanser system is optimally controlled to the given test scenarios, minimizing system response time and transient steady-stat-e error.

Introduction

Designing a complex controller to properly control an advanced aerial vehicle has perplexed control engineers for the past few decades. The aim for a comprehensive controller is to control the actuation of a system through user inputs whilst compensating for possible unmodeled perturbations, sensor noise, and any other potential error causing the current state of the system to deviate from the desired. This report will explore the preliminary design, tuning, and analysis of various control methods on the Quanser helicopter. Such methods include the Linear Quadratic Regulator (LQR), Linear Quadratic Servo (LQ Servo), Linear Quadratic Estimation (LQE), and Dynamics Output Feedback (DOFB). The Qaunser helicopter has 3 degrees of freedom: roll, pitch, and travel. The Quanser helicopter is actuated via two rotor speeds: V_{cyc} , the electric voltage resulting in differential change in the two rotor speeds, and Vcoll, the electric voltage controlling the collective speed of the two propellers (Jonathan P. How & Frazzoli, 2010). The outputs of the system are three angles: roll φ, pitch θ, and travel ψ. For this analysis, the travel output is omitted from the analysis to simplify the solution space.

Theory

Several well-known control techniques exist to solve a multi-input multi-output (MIMO) controls problem. Ideally, for a given system, an optimal controller is used that best utilizes theoretical notions of control law to solve for the system outputs. One such method is the LQR method, a controller design that provides feedback gain. This Full-State Feedback Controller allows for the system to ready a desired steady state value using the LQR method and scaling factor, $\bar{N}$, to facilitate a zero error at steady-state (How & Frazzoli, n.d.-a). Another variation of the LQR method used is named the LQ Servo method. This controller does not use the scaling as it is generally sensitive to system parameters being accurately known. Instead, the LQR method is modified to "ensure that zero steady state error is robustly achieved in response to constant reference commands, done by augmenting integrators to the system output and then including a penalty on the integrated output in LQ cost" (How & Frazzoli, n.d.-b). LQE methods are also utilized on the problem analysis, which creates an "estimate" of the dynamic system states based on the measured output of the system (How & Frazzoli, n.d.-c). Once the LQE and LQ Servo methods are developed, it is advantageous to have a combined estimator and regulator method for system control, called a DOFB controller. For all the utilized methods, the designer selected tuning parameters are adjusted to exhibit an optimal output results, chosen through a compromise between reaching steady-state at the fastest time and minimizing the overall error between the desired output states and current system states (How & Frazzoli, n.d.-d). When proper tuned parameters have been selected, the parameters are then utilized in the comprehensive Simulink model provided by Dr. Rahim to analyze the overall system performance to input conditions for the roll and pitch control outputs.

Problem Definition

To properly simulate the dynamic effects of the Quanser system, the system's nonlinear equations of motion must be defined. The given equations below are derived for the system as such (Jonathan P. How & Frazzoli, 2010):

$$I_{xx}\ddot{\phi} = \tau_{cyc} l_h - mgl_{\phi} sin\phi - L_p\dot{\phi} - I_r\omega_{rotor}(\dot{\theta}\cos\phi + \psi sin\phi) \tag{1}$$

$$I_{yy}\ddot{\theta} = \tau_{coll} l_{boom} cos\phi - Mgl_{\theta}\sin(\theta + \theta_{rest}) - Dl_{boom} sin\gamma + I_r\omega_{rotor}\dot{\phi} - M_q\dot{\phi} - M_q\dot{\theta} \tag{2}$$

$$I_{zz}\ddot{\psi} = \tau_{coll} l_{boom} sin\phi - Dl_{boom} cos\gamma \tag{3}$$

$$D = K_D\dot{\psi} \tag{4}$$

$$\tau_{coll} = K_{\tau}\omega_{coll} - K_v\dot{\psi} \tag{5}$$

$$\tau_{cyc} = K_{\tau}\omega_{cyc} - K_v\dot{\psi} \tag{6}$$

$$\dot{\omega}_{cyc} + 6\omega_{cyc} = 780V_{cyc} \tag{7}$$

$$I_{xx}\ddot{\phi} = \tau_{cyc}l_h - mgl_\phi sin\phi - L_p\dot{\phi} - I_r\omega_{rotor}(\dot{\theta}\cos\phi + \psi sin\phi) \tag{1}$$

$$\dot{\omega}_{coll} + 6\omega_{coll} = 540V_{coll} \tag{8}$$

The terms in Equation 1 (in order of left to right) represent the systems inertial roll angular acceleration, torque from rotor length due to the cyclic thrust, torque due to gravitational acceleration, roll damping effects (causing lag in the roll angle movement), and rotor inertia effects. The terms in Equation 2 (in order of left to right) represent the inertial pitch angular acceleration, torque from rotor length due to the collective thrust, torque due to gravitational acceleration, induced drag effects from the rotor aerodynamics, rotor inertia effects, and pitch damping effects (causing lag in the roll angle movement). The terms in Equation 3 (in order of left to right) represent the systems inertial travel angular acceleration, torque due to the collective thrust, and induced drag effects from the rotor aerodynamics. All terms are relative to their operating reference and rotational planes.

The following equations are also provided to complete the analysis for controller design (Jonathan P. How & Frazzoli, 2010):

$$I_{zz} = 0.93\ Nms^2 \tag{9}$$

$$K_D = 0 \tag{10}$$

$$\omega_{coll_0} = \frac{Mgl_\theta \sin(\theta_{rest})}{K_\tau l_{boom}} \tag{11}$$

$$K_v = 0.0125\omega_{coll_0}K_\tau l_{boom} \tag{12}$$

The system parameters and constants are also defined in Table 1 (Jonathan P. How & Frazzoli, 2010)

Table 1: Physical Parameters

Parameter	**Value**	**Units**	**Description**
m	1.15	kg	Mass of rotor assembly
M	3.57	kg	Mass of the whole setup
l_{boom}	0.66	m	Length from pivot point to heli body
l_{ϕ}	0.004	m	Length of the pendulum for roll axis
l_{θ}	0.014	m	Length of the pendulum for pitch axis
l_h	0.177	m	Length from pivot point to the rotor
I_{xx}	0.036	Nms^2	Moment of inertia about the x-axis
I_{yy}	0.93	Nms^2	Moment of inertia about y-axis
K_{τ}	4.25×10^{-3}	Ns	Coefficient of thrust
θ_{rest}	-25	$deg.$	Theta rest value
g	9.81	$\frac{m}{s^2}$	Acceleration due to gravity
L_p	[0.02, 0.2]	Nms	Roll damping coefficient
M_q	[0.1, 0.9]	Nms	Pitch damping coefficient

The following system must be linearized and analyzed using the method mentioned in the Theory section: LQR, LQ Servo, LQE, DOFB, and the Simulink model.

Methodology

To linearize the system equations presented in the Problem Definition, the various state-space system derivations have been developed to compute the overall state-space representation of the system, fully describing the Quanser system dynamics (neglecting rotor inertia). Using the equations in Appendix 1, the resulting state space matrix is:

$$\dot{x}(t)=\begin{bmatrix}\dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\\ \ddot{\phi}\\ \ddot{\theta}\\ \ddot{\psi}\\ \dot{\omega}_{cyc}\\ \dot{\omega}_{col}\end{bmatrix}=\begin{bmatrix}0 & 0 & 0 & 1 & 0 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0\\ \frac{-mgl_{\phi}}{I_{xx}} & 0 & 0 & \frac{-L_p}{I_{xx}} & 0 & \frac{-K_v l_h}{I_{xx}} & \frac{K_\tau l_h}{I_{xx}} & 0\\ 0 & \frac{-Mgl_{\theta}\cos(\theta_{rest})}{I_{yy}} & 0 & 0 & \frac{-M_q}{I_{yy}} & \frac{-K_v l_{boom}}{I_{yy}} & 0 & \frac{K_\tau l_{boom}}{I_{yy}}\\ \frac{K_\tau l_{boom}\omega_{coll_0}}{I_{zz}} & 0 & 0 & 0 & 0 & \frac{-K_D l_{boom}}{I_{zz}} & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & -6 & 0\\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -6\end{bmatrix}\begin{bmatrix}\phi\\ \theta\\ \psi\\ \dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\\ \omega_{cyc}\\ \omega_{coll}\end{bmatrix}$$

$$+\begin{bmatrix}0 & 0\\ 0 & 0\\ 0 & 0\\ 0 & 0\\ 0 & 0\\ 0 & 0\\ 780 & 0\\ 0 & 540\end{bmatrix}\begin{bmatrix}V_{cyc}\\ V_{coll}\end{bmatrix}$$

$$y(t)=\begin{bmatrix}1 & 0 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0\\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}\phi\\ \theta\\ \psi\\ \dot{\phi}\\ \dot{\theta}\\ \dot{\psi}\\ \omega_{cyc}\\ \omega_{coll}\end{bmatrix}+Du(t)$$

$$D=0$$

Various MATLAB scripts have been developed to design and solve the prescribed controllers. Appendix 2 details the stability, controllability, observability, stabilizability, and detectability of the system, and creates the LQR roll controller for response and V_{cyc} analysis. Appendix 3 creates the LQ Servo pitch controller for response and V_{coll} analysis. Appendix 3 designs the LQE estimator using a LQ Servo controller for both roll and pitch response analysis. Appendix 4 designs the DOFB roll and controller for system response analysis. Appendix 5. Designs a similar DOFB controller for pitch and roll, intended for the defined parameters in the MATLAB workspace to be passed onto the Simulink model for the Quanser system. For all the mentioned appendices, the system and controller tuning variables are either manually altered to reach the desired state by the designer or are automatically iterated to visually choose parameters based on a select few parameter domains.

Results and Discussions

The results from Appendix 1 shows that the system and state-space representation have all negative real eigenvalues, and zero uncontrollable and unobservable system states. This overall means that the total defined system is stable, controllable, observable, stabilizable, and detectable. Appendix 2, the designed LQR system response and V_{cyc} timeseries are displayed in Figure 1 and 2 respectively. The chosen tuned parameters are: $\rho_e = 0.02$

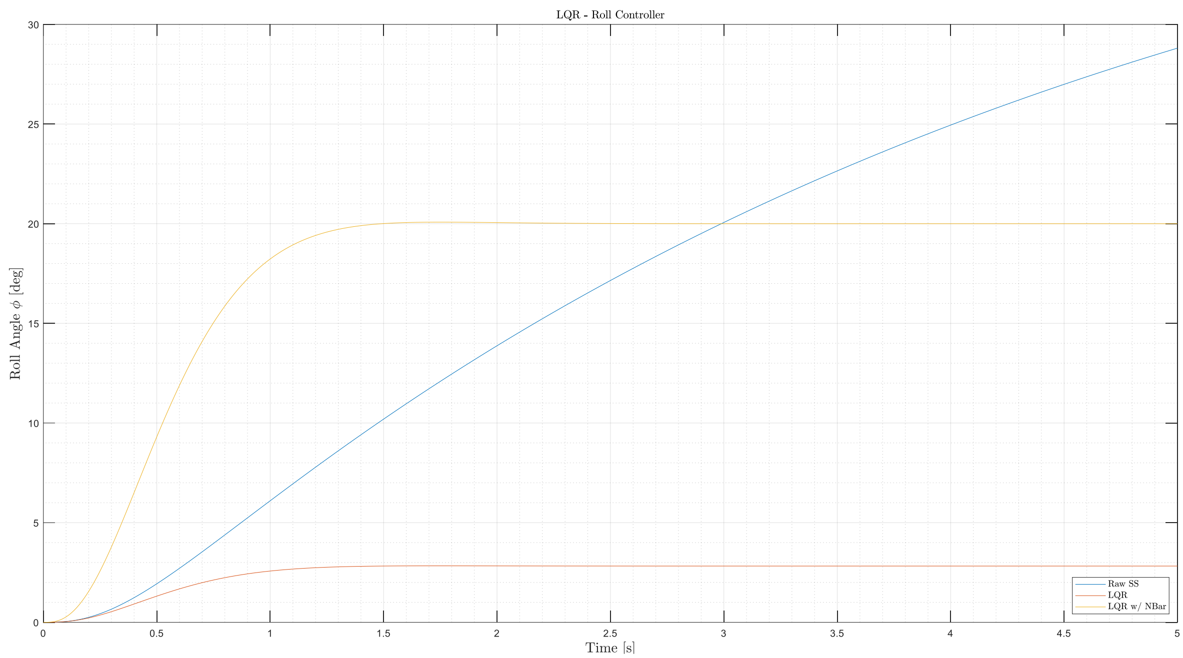


Figure 1: LQR System Response - 20deg roll step

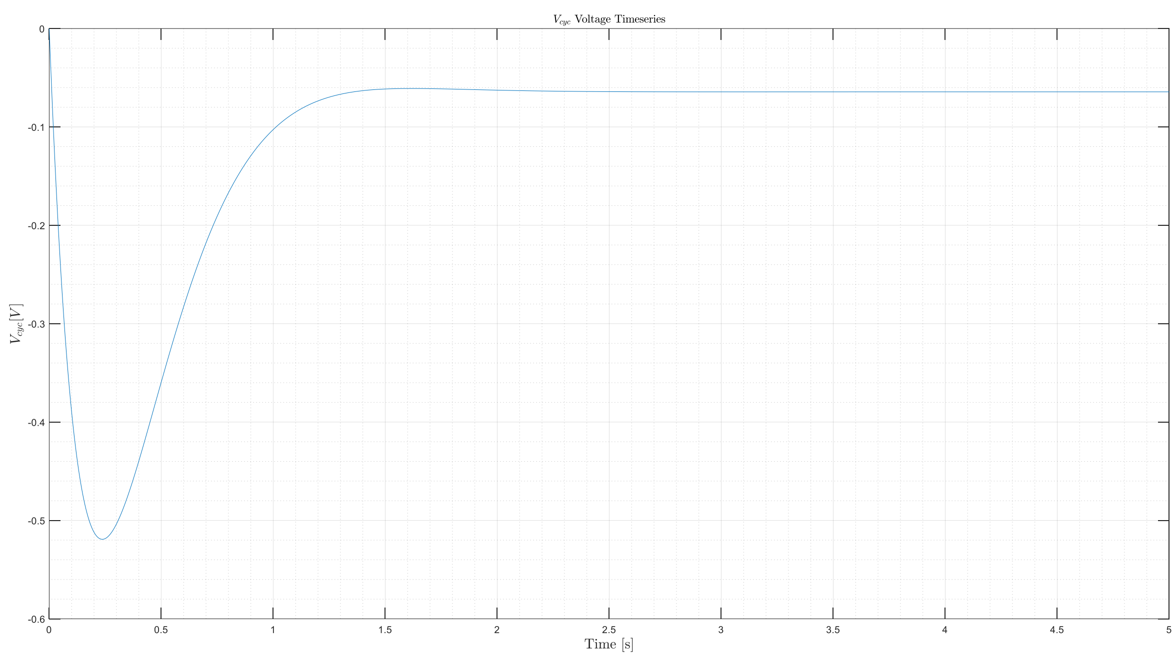


Figure 2: LQR V_{cyc} Timeseries

The system stabilizes at the steady-state roll angle of 20° at approximately 1.5 seconds and the V_{cyc} does not exit the $\pm 5V$ range.

Appendix 3 explores the design of the LQ Servo pitch controller, resulting in the system response and V_{coll} timeseries displayed in Figure 3 and 4 respectively.

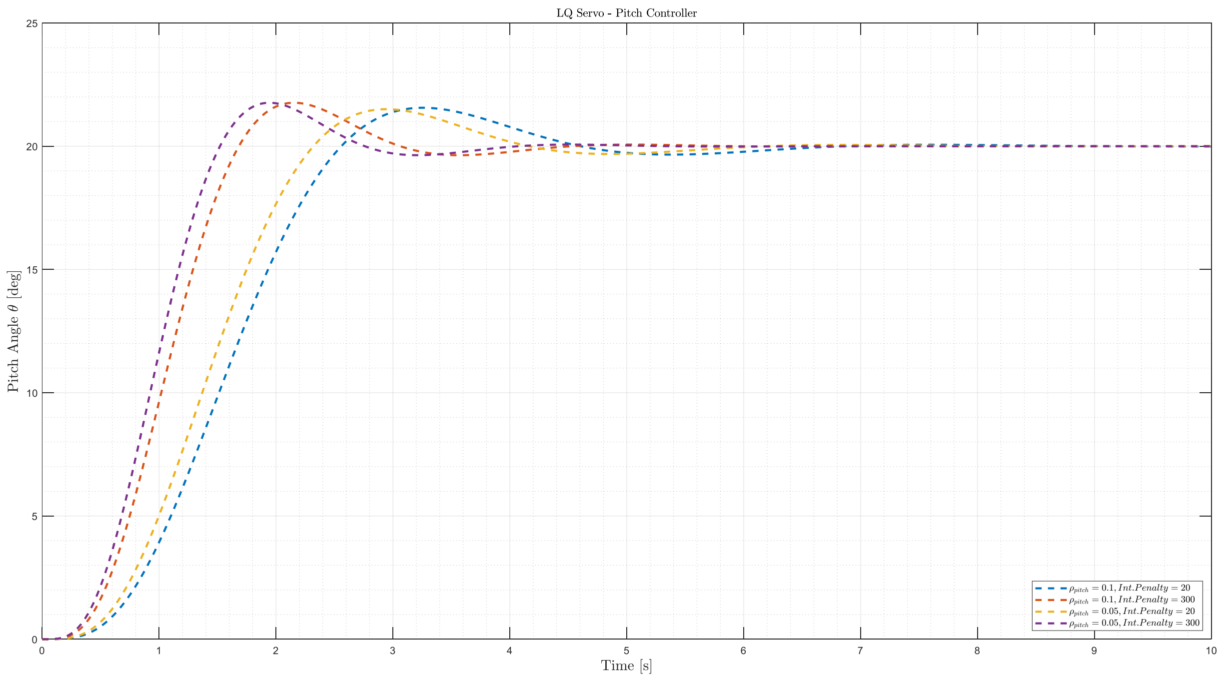


Figure 3: LQ Servo Pitch Controller - 20deg pitch step

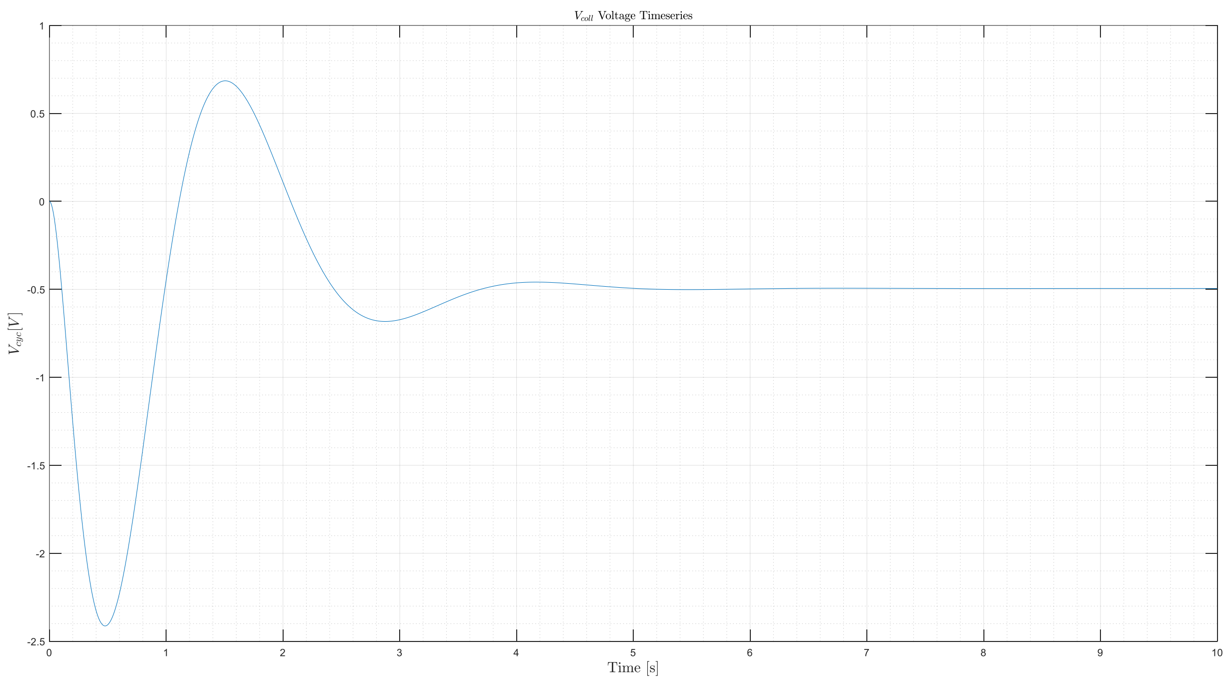


Figure 4: LQ Servo V_{coll} Timeseries

Based on the response from Figure 1, the optimal values for the controlled parameters are: $\rho_e =$ 0.05 and an integration penalty value of 300.

Appendix 4 designs the LQE for the roll and pitch response of the system. No controllers were used in this process, explaining why the angle response does not reach a steady state value. Figure 5, 6, 7, and 8 display the LQE roll angle response estimation, LQE roll angle system states estimation error, LQE pitch angle response estimation, and LQE pitch angle system states estimation error, respectively.

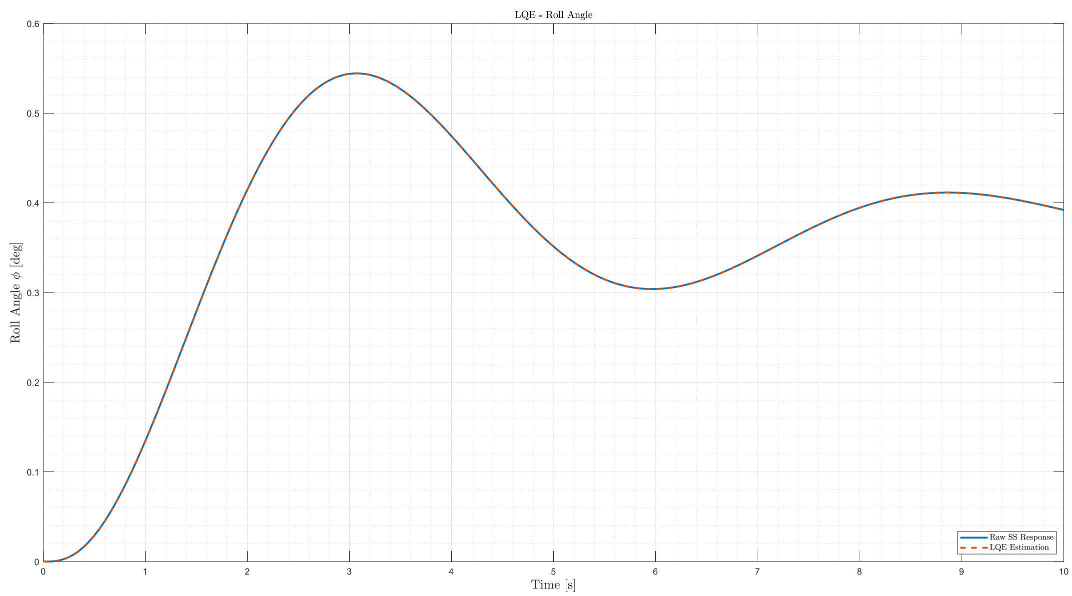


Figure 5: LQE Roll Angle Estimation

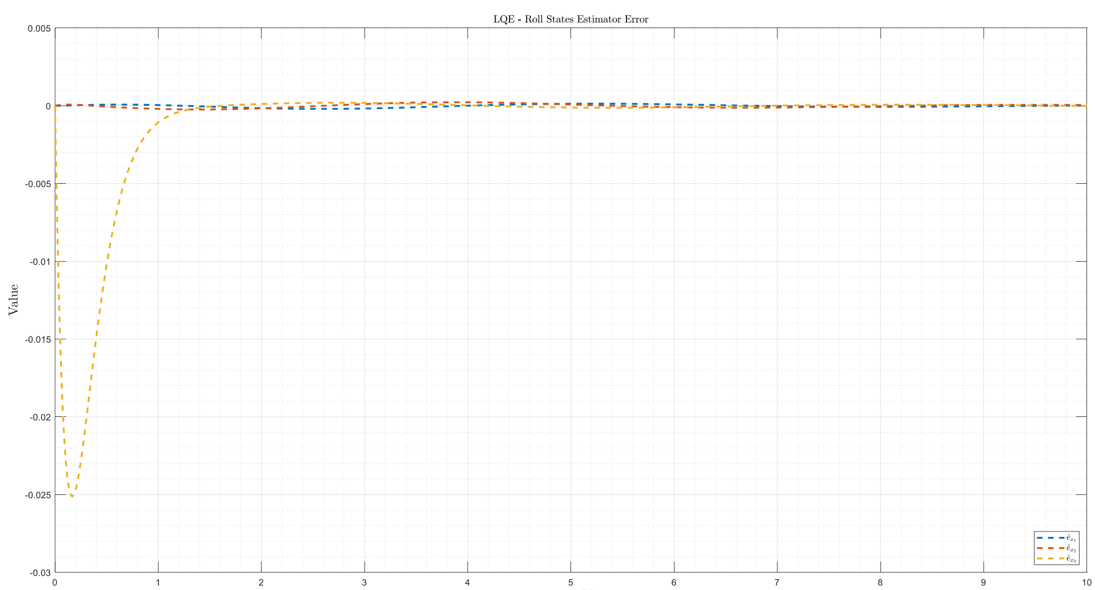


Figure 6: LQE Roll States Estimation Error

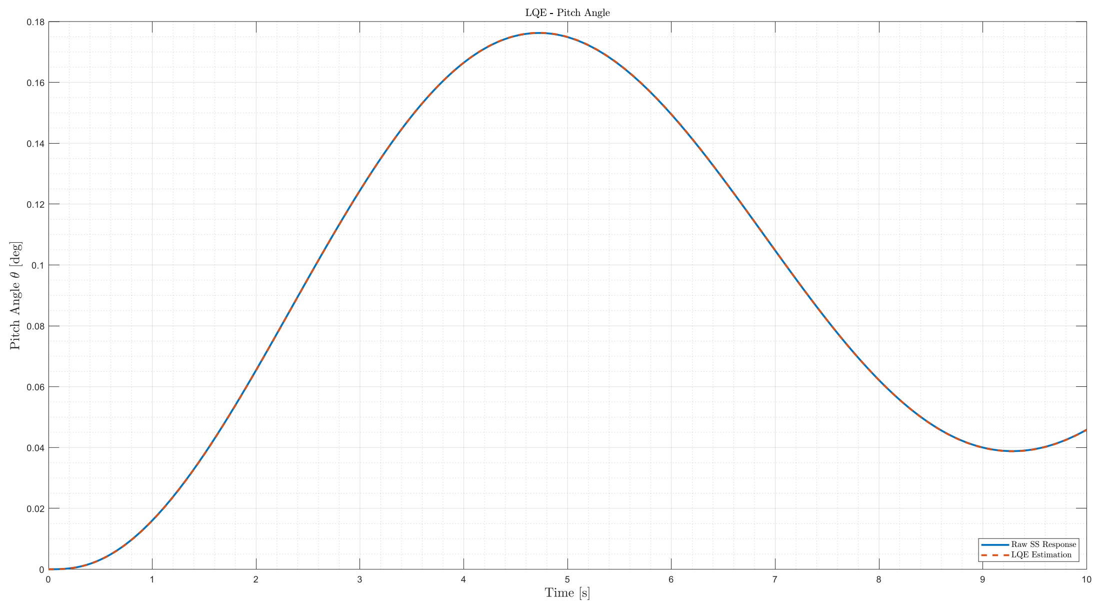


Figure 7: LQE Pitch Angle Estimation

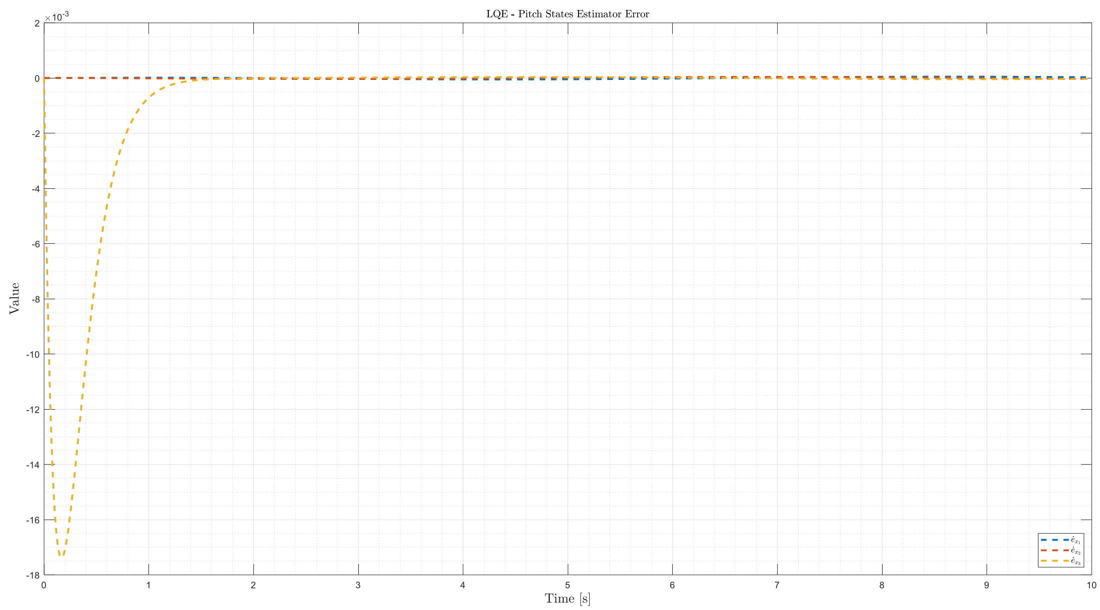


Figure 8: LQE Pitch States Estimation Error

The above figure show that there is low estimation error for the chosen tuning parameter, $R_{vv} = 0.4$ for both roll and pitch estimation. The LQE (and this the tuning parameters) was also analyzed for the "worst" case damping ratios, $L_p = 0.02$ and $M_q = 0.1$.

The DOFB controllers, designed in Appendix 5, was designed using a combination of the previous methodologies and respective theoretical derivations (derived in the references in the

Theory section). The DOFB controller is tuned using the four parameters from the previous estimators and regulators: ρ_e, R_{ww}, R_{vv}, and integration penalty. The results from the DOFB controllers can be simplified to have the same values for tuning parameters. From the cyan dashed line, the optimal system parameters for both the roll and pitch DOFB controllers are $\rho_e = 0.01, R_{ww} = 1, R_{vv} = 0.4$, and integration penalty of 300 respectively.

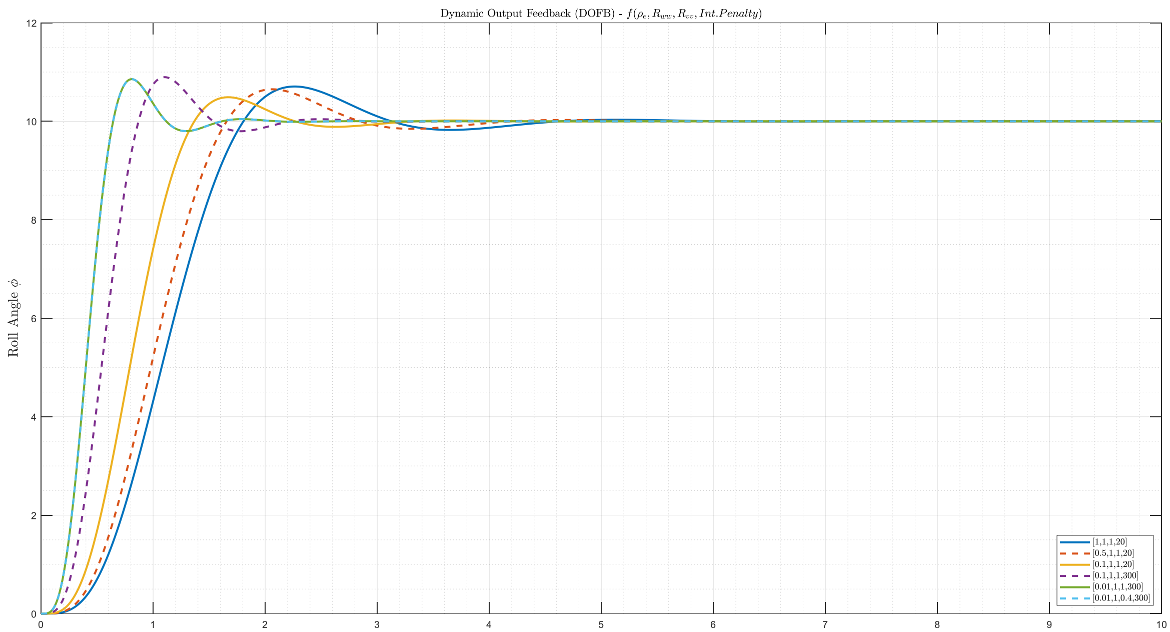


Figure 9: DOFB Roll Angle Response w/ Parameter Tuning

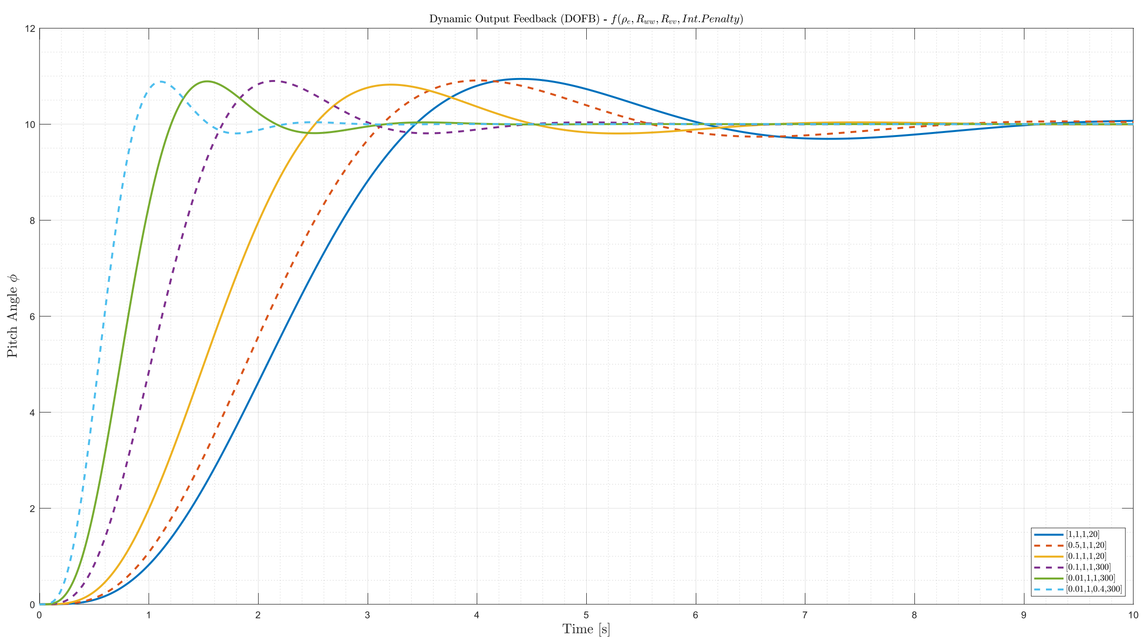


Figure 10: DOFB Pitch Angle Response w/ Parameter Tuning

Once the DOFB controller were designed, the MATLAB code in Appendix 5 was developed to create similar workspace variables for the Quanser Simulink model to execute. The Simulink Model was executed using various input parameters. The model parameters had to be re-tuned to get an optimal response from the controller in the Quanser model. The following test cases can be found in the appendices listed in Table 2.

Table 2: Test Condition Results Reference

Test Conditions	Appendix #
$\Delta\phi = +15°, \phi_0 = 0, \theta = 0°$	7
$\Delta\phi = -15°, \phi_0 = 0, \theta = 0°$	8
$\Delta\phi = +15°, \phi_0 = 0, \theta = 10°$	9
$\Delta\phi = -15°, \phi_0 = 0, \theta = 10°$	10
$\Delta\theta = +15°, \theta_0 = -5, \phi = 0°$	11
$\Delta\theta = +15°, \theta_0 = -5, \phi = 20°$	12

From the results in the appendices listed in Table 2, it can be shown that the chosen tuning parameters in Table 3 resulted in optimal performance of the Quanser system in all test conditions.

Table 3: Optimal Performance Parameters

Parameter	Roll Controller Value	Pitch Controller Value
ρ_e	0.1	0.15
R_{vv}	0.01	0.2
R_{ww}	1	10
Integration Penalty	100	20

Conclusion

From the comprehensive analysis conducted in the Results and Discussion section, it can be demonstrated that the Quanser system is able to be controlled through the design of several methods of controllers and estimators. Using the various design methods (LQR,LQ Servo, LQE, and DOFB), the Quanser system is proven to show an optimal response, given the selected tuning parameters, stabilizing after an acceptable amount of time at steady state, all without exceeding our design constraints set out in the Problem Definition. Through the design and validation of these methods, it can easily be seen why various controls problems are solved using these robust controller designs to complex nonlinear dynamics problems. The methods explored in the report can also be improved even further using alternative methods that aim to reduce overall estimation and steady-state transient error.

References

How, J. P., & Frazzoli, E. (n.d.-a). *Topic #11*. MIT. https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-30-feedback-control-systems-fall-2010/lecture-notes/MIT16_30F10_lec11.pdf

How, J. P., & Frazzoli, E. (n.d.-b). *Topic #13*. MIT. https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-30-feedback-control-systems-fall-2010/lecture-notes/MIT16_30F10_lec13.pdf

How, J. P., & Frazzoli, E. (n.d.-c). *Topic #14*. MIT. https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-30-feedback-control-systems-fall-2010/lecture-notes/MIT16_30F10_lec14.pdf

How, J. P., & Frazzoli, E. (n.d.-d). *Topic #15*. MIT. https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-30-feedback-control-systems-fall-2010/lecture-notes/MIT16_30F10_lec15.pdf

Jonathan P. How, & Frazzoli, E. (2010). *16.30/31 Lab #1*. MIT. https://ocw.mit.edu/courses/aeronautics-and-astronautics/16-30-feedback-control-systems-fall-2010/assignments/MIT16_30F10_lab01.pdf

APPENDICES

APPENDIX 1 – State Space Representation Derivations

$$\ddot{\phi} = \frac{1}{I_{xx}}\left[K_\tau l_h \omega_{cyc} - K_v l_h \dot{\psi} - mgl_\phi \sin(\phi) - L_p\dot{\phi} - I_r\omega_{rotor}\cos(\phi)\,\dot{\theta} - I_r\omega_{rotor}\sin(\phi)\,\dot{\psi}\right]$$

$$\ddot{\theta} = \frac{1}{I_{yy}}\left[K_\tau l_{boom}\cos(\phi)\,\omega_{coll} - K_v l_{boom}\cos(\phi)\,\dot{\psi} - Mgl_{theta}\sin(\theta+\theta_{rest}) - K_D l_{boom}\sin(\gamma)\,\dot{\psi} + I_r\omega_{rotor}\dot{\phi} - M_q\dot{\theta}\right]$$

$$\ddot{\psi} = \frac{1}{I_{zz}}\left[K_\tau l_{boom}\sin(\phi)\omega_{coll} - K_v l_{boom}\sin(\phi)\,\dot{\psi} - K_D l_{boom}\cos(\gamma)\,\dot{\psi}\right]$$

$$\dot{\omega}_{cyc} = -6\omega_{cyc} + 780V_{cyc}$$

$$\dot{\omega}_{coll} = -6\omega_{coll} + 540V_{cyc}$$

$$\begin{array}{ll} x_1 = \phi & x_2 = \dot{x}_1 = \dot{\phi} \\ x_3 = \theta & x_4 = \dot{x}_3 = \dot{\theta} \\ x_5 = \psi & x_6 = \dot{x}_5 = \dot{\psi} \\ x_7 = \omega_{cyc} & x_8 = \dot{x}_7 = \dot{\omega}_{cyc} \\ x_9 = \omega_{coll} & x_{10} = \dot{x}_9 = \dot{\omega}_{coll} \end{array}$$

Steady Hover Condition:

$$x_{1_0} = x_{2_0} = x_{3_0} = x_4 = x_5 = x_6 = 0$$

$$\dot{\omega}_{cyc_0} = \dot{\omega}_{coll_0} = \gamma_0 = 0$$

$$\dot{x}_1 = x_2$$

$$\dot{x}_2 = \frac{1}{I_{xx}}\left[K_\tau l_h x_7 - K_v l_h x_6 - mgl_\phi\sin(x_1) - L_p x_2 - I_r\omega_{rotor}\cos(x_1)\,x_4 - I_r\omega_{rotor}\sin(x_1)\,x_6\right]$$

$$\dot{x}_3 = x_4$$

$$\dot{x}_4 = \frac{1}{I_{yy}}\left[K_\tau l_{boom}\cos(x_1)\,x_9 - K_v l_{boom}\cos(x_1)\,x_6 - Mgl_\theta\sin(x_3+\theta_{rest}) - K_D l_{boom}\sin(\gamma)\,x_6 + I_r\omega_{rotor}x_2 - M_q x_4\right]$$

$$\dot{x}_5 = x_6$$

$$\dot{x}_6 = \frac{1}{I_{zz}}\left[K_\tau l_{boom}\sin(x_1)x_9 - K_v l_{boom}\sin(x_1)\,x_6 - K_D l_{boom}\cos(\gamma)\,x_6\right]$$

$$A = \begin{bmatrix} \frac{\partial f_1}{\partial x_1} & \cdots & \frac{\partial f_1}{\partial x_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial x_1} & \cdots & \frac{\partial f_n}{\partial x_n} \end{bmatrix}$$

$$A_{\frac{\partial \dot{x}_2}{\partial x_1}} = \frac{1}{I_{xx}}\left[-mgl_\phi \cancel{\cos(x_{1_0})}^{1} + I_r\omega_{rotor}\cancel{\sin(x_{1_0})}^{0}\, x_4 - I_r\omega_{rotor}\cos(x_{1_0})\,\cancel{x_6}^{0}\right] = \frac{-mgl_\phi}{I_{xx}}$$

$$A_{\frac{\partial \dot{x}_2}{\partial x_2}} = \frac{-L_p}{I_{xx}}$$

$$A_{\frac{\partial \dot{x}_2}{\partial x_4}} = \frac{-I_r\omega_{rotor}\cancel{\cos(x_{1_0})}^{1}}{I_{xx}} = \frac{-I_r\omega_{rotor}}{I_{xx}}$$

$$A_{\frac{\partial \dot{x}_2}{\partial x_6}} = \frac{-K_v l_h - I_r\omega_{rotor}\cancel{\sin(x_{1_0})}^{0}}{I_{xx}} = \frac{-K_v l_h}{I_{xx}}$$

$$A_{\frac{\partial \dot{x}_2}{\partial x_7}} = \frac{K_\tau l_h}{I_{xx}}$$

$$A_{\frac{\partial \dot{x}_4}{\partial x_2}} = \frac{I_r\omega_{rotor}}{I_{yy}}$$

$$A_{\frac{\partial \dot{x}_4}{\partial x_3}} = \frac{-Mgl_\theta\cos(\cancel{x_{3_0}}^{0} + \theta_{rest})}{I_{yy}} = \frac{-Mgl_\theta\cos(\theta_{rest})}{I_{yy}}$$

$$A_{\frac{\partial \dot{x}_4}{\partial x_4}} = \frac{K_\tau l_{boom}\cancel{\sin(x_{1_0})}^{0} - M_q}{I_{yy}} = \frac{-M_q}{I_{yy}}$$

$$A_{\frac{\partial \dot{x}_4}{\partial x_6}} = \frac{-K_v l_{boom}\cancel{\cos(x_{1_0})}^{1} - K_D l_{boom}\cancel{\sin(\gamma_0)}^{0}}{I_{yy}} = \frac{-K_v l_{boom}}{I_{yy}}$$

$$A_{\frac{\partial \dot{x}_4}{\partial x_9}} = \frac{K_\tau l_{boom}\cancel{\cos(x_{1_0})}^{1}}{I_{yy}} = \frac{K_\tau l_{boom}}{I_{yy}}$$

$$A_{\frac{\partial \dot{x}_6}{\partial x_1}} = \frac{K_\tau l_{boom}\cancel{\cos(x_{1_0})}^{1}\, x_{9_0} - K_v l_{boom}\cancel{\cos(x_{1_0})}^{1}\,\cancel{x_{6_0}}^{0}}{I_{zz}} = \frac{K_\tau l_{boom}\omega_{coll_0}}{I_{zz}}$$

$$A_{\frac{\partial \dot{x}_6}{\partial x_6}} = \frac{-K_v l_{boom}\cancel{\sin(x_{1_0})}^{0} - K_D l_{boom}\cancel{\cos(\gamma_0)}^{1}}{I_{zz}} = \frac{-K_D l_{boom}}{I_{zz}}$$

sssss

APPENDIX 2 – q4.m

```
%MECH-4672 Flight Dynamics and Control of UAVs - Course Project
%Atilla Saadat - 104411786
clc;clear;close all;

m = 1.15;
M = 3.57;
l_boom = 0.66;
l_psi = 0.004;
l_theta = 0.014;
l_h = 0.177;
I_xx = 0.036;
I_yy = 0.93;
K_t = 4.25e-3;
theta_rest = deg2rad(-25);
g = 9.81;
L_p = mean([0.02,0.2]);
M_q = mean([0.1,0.9]);

I_zz = 0.93;
K_D = 0;
omega_coll_0 = M*g*l_theta*sin(theta_rest)/(K_t*l_boom);
K_v = 0.0125*omega_coll_0*K_t*l_boom;

A = [0 0 0 1 0 0 0 0;
    0 0 0 0 1 0 0 0;
    0 0 0 0 0 1 0 0;
    -m*g*l_psi/I_xx 0 0 -L_p/I_xx 0 -K_v*l_h/I_xx K_t*l_h/I_xx 0;
    0 -M*g*l_theta*cos(theta_rest)/I_yy 0 0 -M_q/I_yy -K_v*l_boom/I_yy 0
K_t*l_boom/I_yy;
    K_t*l_boom*omega_coll_0/I_zz 0 0 0 0 -K_D*l_boom/I_zz 0 0;
    0 0 0 0 0 0 -6 0;
    0 0 0 0 0 0 0 -6;
    ];

Aroll = [
    0 1 0;
    -m*g*l_psi/I_xx -L_p/I_xx K_t*l_h/I_xx;
    0 0 -6
    ];

Apitch = [
    0 1 0;
    -M*g*l_theta*cos(theta_rest)/I_yy -M_q/I_yy K_t*l_boom/I_yy;
    0 0 -6
    ];

Atravel = [0 1; -K_D*l_boom/I_zz 0;];

B = [0   0;
     0   0;
     0   0;
```

```
     0   0;
     0   0;
     0   0;
     780 0;
     0 540];

Broll = [0;0;780];
Bpitch = [0;0;540];
Btravel = [0;K_t*l_boom*omega_coll_0/I_zz];

C = [1 0 0 0 0 0 0 0;
      0 1 0 0 0 0 0 0;
      0 0 1 0 0 0 0 0];
Croll = [1 0 0];
Cpitch = [1 0 0];

D = 0;

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

%determine stability
stab = eig(Aroll);
disp(stab);
%determine observability
roll_Ob = obsv(Aroll,Croll);
roll_unob = length(Aroll)-rank(roll_Ob);

%determine controllability
roll_Co = ctrb(Aroll,Broll);
roll_unco = length(Aroll) - rank(roll_Co);

pitch_Ob = obsv(Apitch,Cpitch);
pitch_unob = length(Apitch)-rank(pitch_Ob);

%determine controllability
pitch_Co = ctrb(Apitch,Bpitch);
pitch_unco = length(Apitch) - rank(pitch_Co);
%%

Qroll = 1*(Croll'*Croll);
Rroll = 0.02*eye(size(Broll,2));

[Kroll,Proll,lambda_cl_roll] = lqr(Aroll,Broll,Qroll,Rroll);
ss_orig = ss(Aroll,Broll,Croll,D);
ss_cl = ss((Aroll-Broll*Kroll),Broll,Croll,D);

Nbar = -inv(Croll*(Aroll-Broll*Kroll)^-1*Broll);
ss_clNbar = ss((Aroll-Broll*Kroll),Broll*Nbar,Croll,D);
%knAroll = (Aroll-Broll*Kroll);
%knBroll = Broll*Nbar;
%knAroll = Aroll
%knBroll = Broll
```

```
dt = 0.001; %define step size for simulation time
maxT = 5.0;
t = 0:dt:maxT; %define simulation time domain
u = deg2rad(20)*ones(size(t,2),1); %define step input wih amplitude 1.6 for all time
steps

[y1,t1,x1] = lsim(ss_orig,u,t);
[y2,t2,x2] = lsim(ss_cl,u,t);
[y3,t3,x3] = lsim(ss_clNbar,u,t);

plot(t1,rad2deg(y1),t2,rad2deg(y2),t3,rad2deg(y3));
%plot(t1,rad2deg(y1),t2,rad2deg(y2),tlqe_roll,rad2deg(ylqe_roll));
xlim([0 maxT]);
title('LQR - Roll Controller','Interpreter','latex');
legend({'Raw SS','LQR','LQR w/ NBar'},'Location','southeast','Interpreter','latex');
xlabel('Time [s]','FontSize',14,'FontWeight','bold','Interpreter','latex');
ylabel('Roll Angle
$\phi$ [deg]','FontSize',14,'FontWeight','bold','Interpreter','latex');
grid on
grid minor

figure
uRoll = -Kroll.*x3;
Vcyc_roll = uRoll(:,3);
plot(t1,Vcyc_roll);
title('$V_{cyc}$ Voltage Timeseries','Interpreter','latex');
xlabel('Time [s]','FontSize',14,'FontWeight','bold','Interpreter','latex');
ylabel('$V_{cyc} [V]$','FontSize',14,'FontWeight','bold','Interpreter','latex');
grid on
grid minor
```

APPENDIX 3 – q5.m

```
%MECH-4672 Flight Dynamics and Control of UAVs - Course Project
%Atilla Saadat - 104411786
clc;clear;close all;

m = 1.15;
M = 3.57;
l_boom = 0.66;
l_psi = 0.004;
l_theta = 0.014;
l_h = 0.177;
I_xx = 0.036;
I_yy = 0.93;
K_t = 4.25e-3;
theta_rest = deg2rad(-25);
g = 9.81;
L_p = mean([0.02,0.2]);
M_q = mean([0.1,0.9]);
```

```
I_zz = 0.93;
K_D = 0;
omega_coll_0 = M*g*l_theta*sin(theta_rest)/(K_t*l_boom);
K_v = 0.0125*omega_coll_0*K_t*l_boom;

A = [0 0 0 1 0 0 0 0;
    0 0 0 0 1 0 0 0;
    0 0 0 0 0 1 0 0;
    -m*g*l_psi/I_xx 0 0 -L_p/I_xx 0 -K_v*l_h/I_xx K_t*l_h/I_xx 0;
    0 -M*g*l_theta*cos(theta_rest)/I_yy 0 0 -M_q/I_yy -K_v*l_boom/I_yy 0
K_t*l_boom/I_yy;
    K_t*l_boom*omega_coll_0/I_zz 0 0 0 0 -K_D*l_boom/I_zz 0 0;
    0 0 0 0 0 0 -6 0;
    0 0 0 0 0 0 0 -6;
    ];

Aroll = [
    0 1 0;
    -m*g*l_psi/I_xx -L_p/I_xx K_t*l_h/I_xx;
    0 0 -6
    ];

Apitch = [
    0 1 0;
    -M*g*l_theta*cos(theta_rest)/I_yy -M_q/I_yy K_t*l_boom/I_yy;
    0 0 -6
    ];

Atravel = [0 1; -K_D*l_boom/I_zz 0;];

B = [0   0;
     0   0;
     0   0;
     0   0;
     0   0;
     0   0;
     780 0;
     0 540];

Broll = [0;0;780];
Bpitch = [0;0;540];
Btravel = [0;K_t*l_boom*omega_coll_0/I_zz];

C = [1 0 0 0 0 0 0 0;
      0 1 0 0 0 0 0 0;
      0 0 1 0 0 0 0 0];
Croll = [1 0 0];
Cpitch = [1 0 0];

D = 0;

%% LQ Servo - Pitch
```

```
%A2pitch = horzcat([Apitch; -Cpitch],zeros(4,1));
A2pitch = [Apitch zeros(3,1); -Cpitch zeros(1,1)];
B2pitch = [Bpitch; 0];
C2pitch = [Cpitch 0];

intP = 300; %integration penalty

Rrho_pitch = 0.2;

dt = 0.001; %define step size for simulation time
maxT = 10.0;
t = 0:dt:maxT; %define simulation time domain
u = deg2rad(20)*ones(size(t,2),1); %define step input wih amplitude 1.6 for all time
steps

[y1,t1,x1,Kpitch] = lq_servo(0.1,20,Apitch,Bpitch,Cpitch,D,u,t);
[y2,t2,x2,Kpitch] = lq_servo(0.1,300,Apitch,Bpitch,Cpitch,D,u,t);
[y3,t3,x3,Kpitch] = lq_servo(0.05,20,Apitch,Bpitch,Cpitch,D,u,t);
[y4,t4,x4,Kpitch] = lq_servo(0.05,300,Apitch,Bpitch,Cpitch,D,u,t);

figure
plot(t1,[rad2deg(y1) rad2deg(y2) rad2deg(y3) rad2deg(y4)],'--','LineWidth',2);
xlim([0 maxT]);
title('LQ Servo - Pitch Controller','Interpreter','latex');
legend({'$\rho_{pitch} = 0.1, Int. Penalty = 20$',...
'$\rho_{pitch} = 0.1, Int. Penalty = 300$',...
'$\rho_{pitch} = 0.05, Int. Penalty = 20$',...
'$\rho_{pitch} = 0.05, Int. Penalty = 300$'},...
'Location','southeast','Interpreter','latex');
xlabel('Time [s]','FontSize',14,'Interpreter','latex');
ylabel('Pitch Angle $\theta$ [deg]','FontSize',14,'Interpreter','latex');
grid on;
grid minor

figure
uRoll = -Kpitch.*x4;
Vcoll_pitch = uRoll(:,3);
plot(t1,Vcoll_pitch);
title('$V_{coll}$ Voltage Timeseries','Interpreter','latex');
xlabel('Time [s]','FontSize',14,'FontWeight','bold','Interpreter','latex');
ylabel('$V_{cyc} [V]$','FontSize',14,'FontWeight','bold','Interpreter','latex');
grid on
grid minor

function [y,t,x,Kbar] = lq_servo(Rrho,intP,Apitch,Bpitch,Cpitch,D,u,t)

    Aaug = [Apitch zeros(3,1); -Cpitch zeros(1,1)];
    Baug = [Bpitch; 0];
    Caug = [Cpitch 0];

    QQ = Cpitch'*Cpitch;
    Qpitch = [QQ zeros(3,1); zeros(1,3) intP];
```

```
    RRpitch = Rrho*eye(size(Bpitch,2));

    Kbar = lqr(Aaug,Baug,Qpitch,RRpitch);
    lqs_sys = ss(Aaug-Baug*Kbar,[0 0 0 1]',Caug,D);
    [y,t,x] = lsim(lqs_sys,u,t);
end
```

APPENDIX 4 – q6.m

```
clc;clear;close all;

m = 1.15;
M = 3.57;
l_boom = 0.66;
l_psi = 0.004;
l_theta = 0.014;
l_h = 0.177;
I_xx = 0.036;
I_yy = 0.93;
K_t = 4.25e-3;
theta_rest = deg2rad(-25);
g = 9.81;
%L_p = mean([0.02,0.2]);
L_p = 0.02;
%M_q = mean([0.1,0.9]);
M_q = 0.1;

I_zz = 0.93;
K_D = 0;
omega_coll_0 = M*g*l_theta*sin(theta_rest)/(K_t*l_boom);
K_v = 0.0125*omega_coll_0*K_t*l_boom;

A = [0 0 0 1 0 0 0 0;
    0 0 0 0 1 0 0 0;
    0 0 0 0 0 1 0 0;
    -m*g*l_psi/I_xx 0 0 -L_p/I_xx 0 -K_v*l_h/I_xx K_t*l_h/I_xx 0;
    0 -M*g*l_theta*cos(theta_rest)/I_yy 0 0 -M_q/I_yy -K_v*l_boom/I_yy 0
K_t*l_boom/I_yy;
    K_t*l_boom*omega_coll_0/I_zz 0 0 0 0 -K_D*l_boom/I_zz 0 0;
    0 0 0 0 0 0 -6 0;
    0 0 0 0 0 0 0 -6;
    ];

Aroll = [
    0 1 0;
    -m*g*l_psi/I_xx -L_p/I_xx K_t*l_h/I_xx;
    0 0 -6
    ];

Apitch = [
    0 1 0;
```

```
    -M*g*l_theta*cos(theta_rest)/I_yy -M_q/I_yy K_t*l_boom/I_yy;
    0 0 -6
    ];

Atravel = [0 1; -K_D*l_boom/I_zz 0;];

B = [0   0;
     0   0;
     0   0;
     0   0;
     0   0;
     0   0;
     780 0;
     0 540];

Broll = [0;0;780];
Bpitch = [0;0;540];
Btravel = [0;K_t*l_boom*omega_coll_0/I_zz];

C = [1 0 0 0 0 0 0 0;
      0 1 0 0 0 0 0 0;
      0 0 1 0 0 0 0 0];
Croll = [1 0 0];
Cpitch = [1 0 0];

D = 0;

dt = 0.001; %define step size for simulation time
maxT = 10.0;
t = 0:dt:maxT; %define simulation time domain
u = deg2rad(10)*ones(size(t,2),1); %define step input wih amplitude 1.6 for all time
steps

ss_roll = ss(Aroll,Broll,Croll,D);
ss_pitch = ss(Apitch,Bpitch,Cpitch,D);
[yroll,troll,xroll] = lsim(ss_roll,u,t);
[ypitch,tpitch,xpitch] = lsim(ss_pitch,u,t);
%%
rho_rvv_roll = 0.4;
rho_rww_roll = 0.1;
Rww_roll = rho_rww_roll*eye(size(Broll,2));
Rvv_roll = rho_rvv_roll*eye(size(Broll,2));
Lroll = lqr(Aroll',Croll',Broll*Rww_roll*Broll',Rvv_roll)';
xhat = [0;0;0];
XXhat = xhat;
y_norm = [0];
y_hat = [0];
for i=1:size(troll)-1
    uu = u(i);
    y = Croll*(xroll(i,:))'+D*uu;
    y_norm = [y_norm y];
    yhat = Croll*xhat+D*uu;
    y_hat = [y_hat yhat];
```

```
    xhat = (Aroll*xhat+Broll*uu+Lroll*(y-yhat)')*dt + xhat;
    XXhat = [XXhat,xhat];
end
XXhat = XXhat';
figure
plot(troll,yroll,troll,y_hat,'--','LineWidth',2);
title('LQE - Roll Angle','Interpreter','latex');
legend({'Raw SS Response','LQE
Estimation'},'Location','southeast','Interpreter','latex');
xlabel('Time [s]','FontSize',14,'FontWeight','bold','Interpreter','latex');
ylabel('Roll Angle
$\phi$ [deg]','FontSize',14,'FontWeight','bold','Interpreter','latex');
grid on
grid minor
figure
plot(troll,xroll-XXhat,'--','LineWidth',2);
title('LQE - Roll States Estimator Error','Interpreter','latex');
roll_legend = {'$\dot{e}_{x_1}$','$\dot{e}_{x_2}$','$\dot{e}_{x_3}$'};
legend(roll_legend,'Location','southeast','Interpreter','latex');
xlabel('Time [s]','FontSize',14,'FontWeight','bold','Interpreter','latex');
ylabel('Value','FontSize',14,'FontWeight','bold','Interpreter','latex');
grid on
grid minor
%%
rho_rvv_pitch = 0.4;
rho_rww_pitch = 1;
Rww_pitch = rho_rww_pitch*eye(size(Bpitch,2));
Rvv_pitch = rho_rvv_pitch*eye(size(Bpitch,2));
Lpitch = lqr(Apitch',Cpitch',Bpitch*Rww_pitch*Bpitch',Rvv_pitch)';
xhat = [0;0;0];
XXhat = xhat;
y_norm = [0];
y_hat = [0];
for i=1:size(tpitch)-1
    uu = u(i);
    y = Cpitch*(xpitch(i,:))'+D*uu;
    y_norm = [y_norm y];
    yhat = Cpitch*xhat+D*uu;
    y_hat = [y_hat yhat];
    xhat = (Apitch*xhat+Bpitch*uu+Lpitch*(y-yhat)')*dt + xhat;
    XXhat = [XXhat,xhat];
end
XXhat = XXhat';
figure
plot(tpitch,ypitch,tpitch,y_hat,'--','LineWidth',2);
title('LQE - Pitch Angle','Interpreter','latex');
legend({'Raw SS Response','LQE
Estimation'},'Location','southeast','Interpreter','latex');
xlabel('Time [s]','FontSize',14,'FontWeight','bold','Interpreter','latex');
ylabel('Pitch Angle
$\theta$ [deg]','FontSize',14,'FontWeight','bold','Interpreter','latex');
grid on
grid minor
figure
plot(troll,xpitch-XXhat,'--','LineWidth',2);
```

```
title('LQE - Pitch States Estimator Error','Interpreter','latex');
roll_legend = {'$\dot{e}_{x_1}$','$\dot{e}_{x_2}$','$\dot{e}_{x_3}$'};
legend(roll_legend,'Location','southeast','Interpreter','latex');
xlabel('Time [s]','FontSize',14,'FontWeight','bold','Interpreter','latex');
ylabel('Value','FontSize',14,'FontWeight','bold','Interpreter','latex');
grid on
grid minor
```

APPENDIX 5 – q7.m

```
clc;clear;close all;

m = 1.15;
M = 3.57;
l_boom = 0.66;
l_psi = 0.004;
l_theta = 0.014;
l_h = 0.177;
I_xx = 0.036;
I_yy = 0.93;
K_t = 4.25e-3;
theta_rest = deg2rad(-25);
g = 9.81;
L_p = mean([0.02,0.2]);
L_p = 0.02;
M_q = mean([0.1,0.9]);
M_q = 0.1;

I_zz = 0.93;
K_D = 0;
omega_coll_0 = M*g*l_theta*sin(theta_rest)/(K_t*l_boom);
K_v = 0.0125*omega_coll_0*K_t*l_boom;

A = [0 0 0 1 0 0 0 0;
    0 0 0 0 1 0 0 0;
    0 0 0 0 0 1 0 0;
    -m*g*l_psi/I_xx 0 0 -L_p/I_xx 0 -K_v*l_h/I_xx K_t*l_h/I_xx 0;
    0 -M*g*l_theta*cos(theta_rest)/I_yy 0 0 -M_q/I_yy -K_v*l_boom/I_yy 0
K_t*l_boom/I_yy;
    K_t*l_boom*omega_coll_0/I_zz 0 0 0 0 -K_D*l_boom/I_zz 0 0;
    0 0 0 0 0 0 -6 0;
    0 0 0 0 0 0 0 -6;
    ];

Aroll = [
    0 1 0;
    -m*g*l_psi/I_xx -L_p/I_xx K_t*l_h/I_xx;
    0 0 -6
    ];

Apitch = [
```

```
    0 1 0;
    -M*g*l_theta*cos(theta_rest)/I_yy -M_q/I_yy K_t*l_boom/I_yy;
    0 0 -6
    ];

Atravel = [0 1; -K_D*l_boom/I_zz 0;];

B = [0   0;
     0   0;
     0   0;
     0   0;
     0   0;
     0   0;
     780 0;
     0 540];

Broll = [0;0;780];
Bpitch = [0;0;540];
Btravel = [0;K_t*l_boom*omega_coll_0/I_zz];

C = [1 0 0 0 0 0 0 0;
      0 1 0 0 0 0 0 0;
      0 0 1 0 0 0 0 0];
Croll = [1 0 0];
Cpitch = [1 0 0];

D = 0;

dt = 0.001; %define step size for simulation time
maxT = 10.0;
t = 0:dt:maxT; %define simulation time domain
u = 10*ones(size(t,2),1); %define step input wih amplitude 1.6 for all time steps

%%

%Rrho,rho_rww,rho_rvv,intP
[yroll1,troll1,xroll1,Kroll1,Lroll1] = lq_servo(1,1,1,20,Aroll,Broll,Croll,D,u,t);
[yroll2,troll2,xroll2,Kroll2,Lroll2] = lq_servo(0.5,1,1,20,Aroll,Broll,Croll,D,u,t);
[yroll3,troll3,xroll3,Kroll3,Lroll3] = lq_servo(0.1,1,1,20,Aroll,Broll,Croll,D,u,t);
[yroll4,troll4,xroll4,Kroll4,Lroll4] = lq_servo(0.1,1,1,300,Aroll,Broll,Croll,D,u,t);
[yroll5,troll5,xroll5,Kroll5,Lroll5] =
lq_servo(0.01,1,1,300,Aroll,Broll,Croll,D,u,t);
[yroll6,troll6,xroll6,Kroll6,Lroll6] =
lq_servo(0.01,1,0.4,300,Aroll,Broll,Croll,D,u,t);

plot(troll1,yroll1,'LineWidth',2);
hold on;
plot(troll2,yroll2,'--','LineWidth',2);
plot(troll3,yroll3,'LineWidth',2);
plot(troll4,yroll4,'--','LineWidth',2);
plot(troll5,yroll5,'LineWidth',2);
plot(troll6,yroll6,'--','LineWidth',2);
hold off;
```

```
title('Dynamic Output Feedback (DOFB) - $f(\rho_e,R_{ww},R_{vv},Int.
Penalty)$','Interpreter','latex');
legend({'[1,1,1,20]',...
    '[0.5,1,1,20]',...
    '[0.1,1,1,20]',...
    '[0.1,1,1,300]',...
    '[0.01,1,1,300]',...
    '[0.01,1,0.4,300]'},...
    'Location','southeast','Interpreter','latex');
xlabel('Time [s]','FontSize',14,'FontWeight','bold','Interpreter','latex');
ylabel('Roll Angle $\phi$','FontSize',14,'FontWeight','bold','Interpreter','latex');
grid on;
grid minor;


[ypitch1,tpitch1,xpitch1,Kpitch1,Lpitch1] =
lq_servo(1,1,1,20,Apitch,Bpitch,Cpitch,D,u,t);
[ypitch2,tpitch2,xpitch2,Kpitch2,Lpitch2] =
lq_servo(0.5,1,1,20,Apitch,Bpitch,Cpitch,D,u,t);
[ypitch3,tpitch3,xpitch3,Kpitch3,Lpitch3] =
lq_servo(0.1,1,1,20,Apitch,Bpitch,Cpitch,D,u,t);
[ypitch4,tpitch4,xpitch4,Kpitch4,Lpitch4] =
lq_servo(0.1,1,1,300,Apitch,Bpitch,Cpitch,D,u,t);
[ypitch5,tpitch5,xpitch5,Kpitch5,Lpitch5] =
lq_servo(0.01,1,1,300,Apitch,Bpitch,Cpitch,D,u,t);
[ypitch6,tpitch6,xpitch6,Kpitch6,Lpitch6] =
lq_servo(0.001,1,0.4,300,Apitch,Bpitch,Cpitch,D,u,t);

figure
plot(tpitch1,ypitch1,'LineWidth',2);
hold on;
plot(tpitch2,ypitch2,'--','LineWidth',2);
plot(tpitch3,ypitch3,'LineWidth',2);
plot(tpitch4,ypitch4,'--','LineWidth',2);
plot(tpitch5,ypitch5,'LineWidth',2);
plot(tpitch6,ypitch6,'--','LineWidth',2);
hold off;

title('Dynamic Output Feedback (DOFB) - $f(\rho_e,R_{ww},R_{vv},Int.
Penalty)$','Interpreter','latex');
legend({'[1,1,1,20]',...
    '[0.5,1,1,20]',...
    '[0.1,1,1,20]',...
    '[0.1,1,1,300]',...
    '[0.01,1,1,300]',...
    '[0.01,1,0.4,300]'},...
    'Location','southeast','Interpreter','latex');
xlabel('Time [s]','FontSize',14,'FontWeight','bold','Interpreter','latex');
ylabel('Pitch Angle $\phi$','FontSize',14,'FontWeight','bold','Interpreter','latex');
grid on;
grid minor;

function [y,t,x,Kbar,Lbar] = lq_servo(Rrho,rho_rww,rho_rvv,intP,Aq,Bq,Cq,D,u,t)
```

```
    Aaug = [Aq zeros(3,1); -Cq zeros(1,1)];
    Baug = [Bq; 0];

    Qaug = [Cq'*Cq zeros(3,1); zeros(1,3) intP];
    Raug = Rrho*eye(size(Bq,2));
    Kbar = lqr(Aaug,Baug,Qaug,Raug);

    Rww = rho_rww*eye(size(Bq,2));
    Rvv = rho_rvv*eye(size(Bq,2));
    Lbar = lqr(Aq',Cq',Bq*Rww*Bq',Rvv)';
    %compensators
    ac1=[Aq-Lbar*Cq-Bq*Kbar(1:size(Aq,1)) -Bq*Kbar(end);zeros(3,1)' 0];
    Br=[Lbar*0;1];

    acl = [Aq -Bq*Kbar; [Lbar;-1]*Cq ac1];
    bcl = [Bq*0;Br];
    ccl = [Cq Cq*0 0];
    ss_dofb = ss(acl,bcl,ccl,D);
    [y,t,x] = lsim(ss_dofb,u,t);
end
```

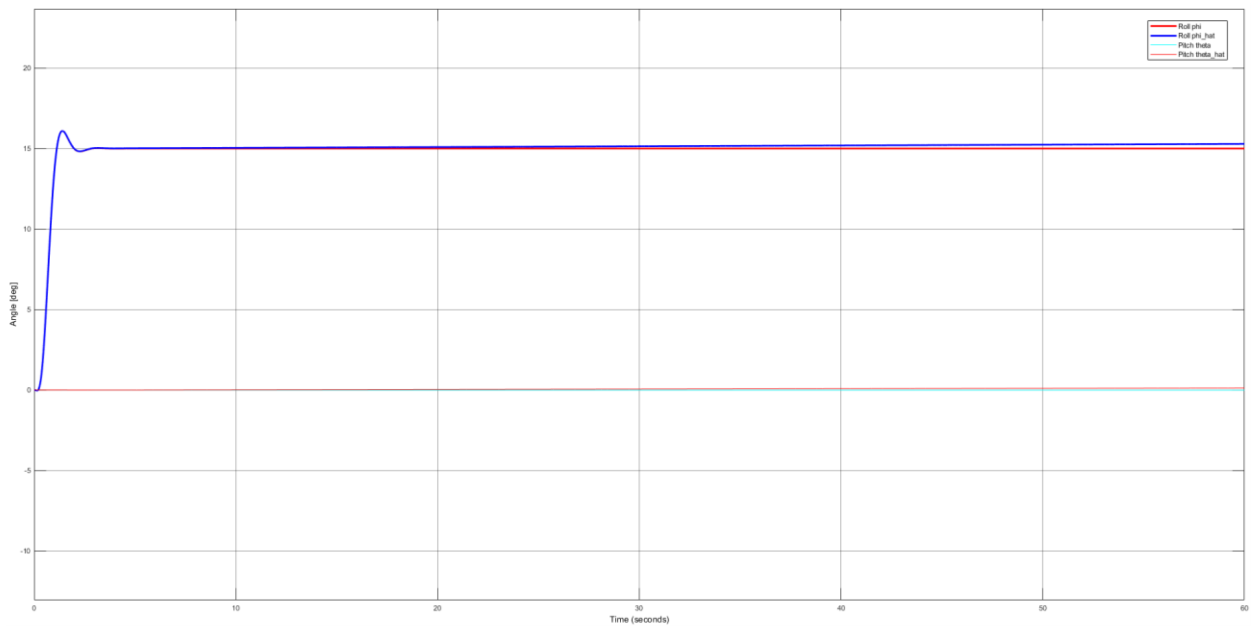
APPENDIX 6 – q9.m

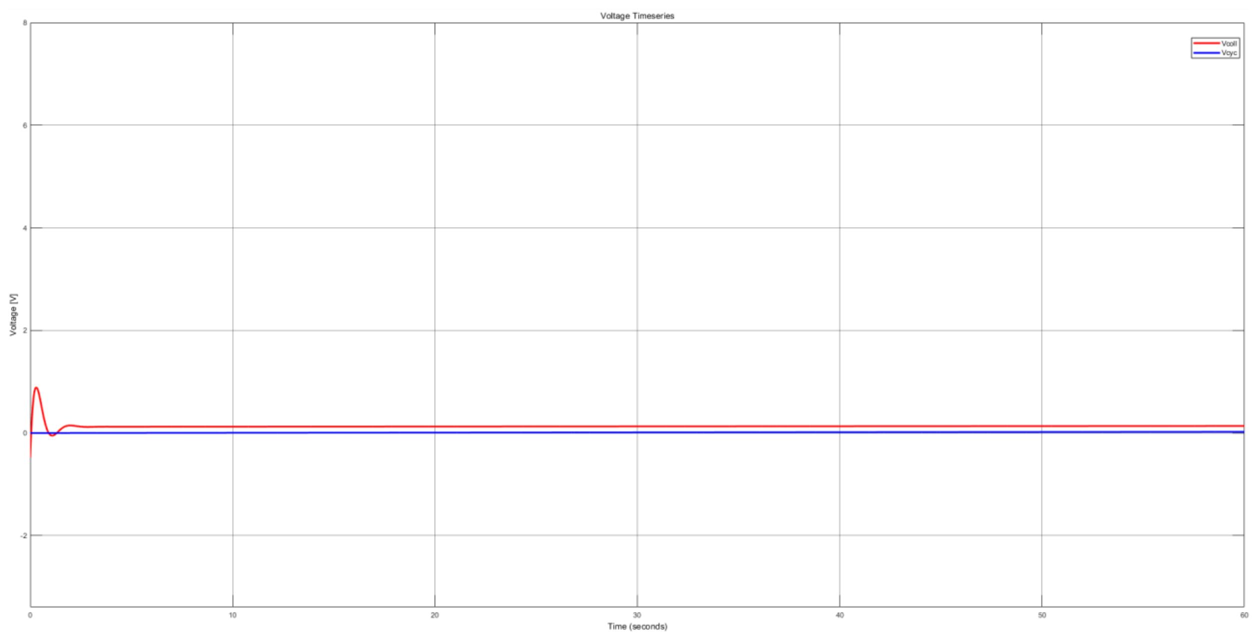
```
clc;clear;close all;

m = 1.15;
M = 3.57;
l_boom = 0.66;
l_psi = 0.004;
l_theta = 0.014;
l_h = 0.177;
I_xx = 0.036;
I_yy = 0.93;
K_t = 4.25e-3;
theta_rest = deg2rad(-25);
g = 9.81;
L_p = mean([0.02,0.2]);
M_q = mean([0.1,0.9]);

I_zz = 0.93;
K_D = 0;
omega_coll_0 = M*g*l_theta*sin(theta_rest)/(K_t*l_boom);
K_v = 0.0125*omega_coll_0*K_t*l_boom;

A = [0 0 0 1 0 0 0 0;
    0 0 0 0 1 0 0 0;
    0 0 0 0 0 1 0 0;
    -m*g*l_psi/I_xx 0 0 -L_p/I_xx 0 -K_v*l_h/I_xx K_t*l_h/I_xx 0;
    0 -M*g*l_theta*cos(theta_rest)/I_yy 0 0 -M_q/I_yy -K_v*l_boom/I_yy 0
K_t*l_boom/I_yy;
    K_t*l_boom*omega_coll_0/I_zz 0 0 0 0 -K_D*l_boom/I_zz 0 0;
```

```
    0 0 0 0 0 0 -6 0;
    0 0 0 0 0 0 0 -6;
    ];

Aroll = [
    0 1 0;
    -m*g*l_psi/I_xx -L_p/I_xx K_t*l_h/I_xx;
    0 0 -6
    ];

Apitch = [
    0 1 0;
    -M*g*l_theta*cos(theta_rest)/I_yy -M_q/I_yy K_t*l_boom/I_yy;
    0 0 -6
    ];

Atravel = [0 1; -K_D*l_boom/I_zz 0;];

B = [0   0;
     0   0;
     0   0;
     0   0;
     0   0;
     0   0;
     780 0;
     0 540];

Broll = [0;0;780];
Bpitch = [0;0;540];
Btravel = [0;K_t*l_boom*omega_coll_0/I_zz];

C = [1 0 0 0 0 0 0 0;
      0 1 0 0 0 0 0 0;
      0 0 1 0 0 0 0 0];
Croll = [1 0 0];
Cpitch = [1 0 0];

D = 0;

dt = 0.001; %define step size for simulation time
maxT = 10.0;
t = 0:dt:maxT; %define simulation time domain
u = deg2rad(10)*ones(size(t,2),1); %define step input wih amplitude 1.6 for all time
steps

%%

%Rrho,rho_rww,rho_rvv,intP
[yroll,troll,xroll,Kroll,Lroll,Rroll] =
lq_servo(0.1,1,0.01,100,Aroll,Broll,Croll,D,u,t);
KIroll = Kroll(4);
Kroll = Kroll(1:3);
Rroll = [Rroll; 0; 0];
```

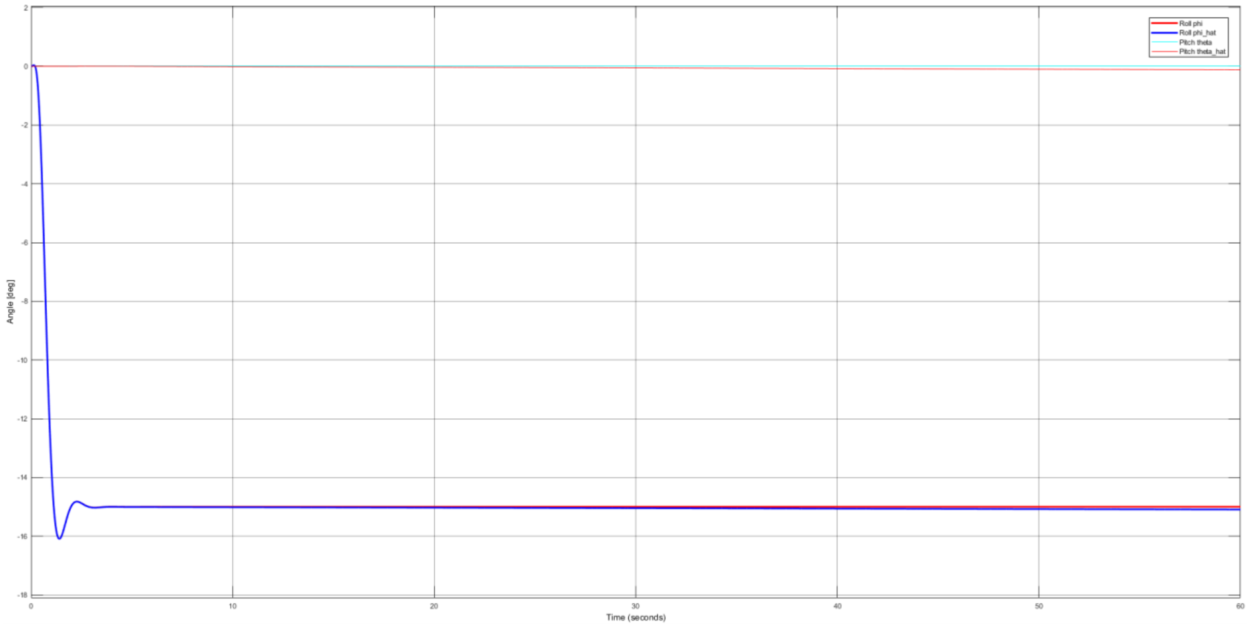
```
[ypitch,tpitch,xpitch,Kpitch,Lpitch,Rpitch] =
lq_servo(0.15,10,0.1,20,Apitch,Bpitch,Cpitch,D,u,t);
KIpitch = Kpitch(4);
Kpitch = Kpitch(1:3);
Rpitch = [Rpitch; 0; 0];

function [y,t,x,Kbar,Lbar,Raug] = lq_servo(Rrho,rho_rww,rho_rvv,intP,Aq,Bq,Cq,D,u,t)
    Aaug = [Aq zeros(3,1); -Cq zeros(1,1)];
    Baug = [Bq; 0];

    Qaug = [Cq'*Cq zeros(3,1); zeros(1,3) intP];
    Raug = Rrho*eye(size(Bq,2));
    Kbar = lqr(Aaug,Baug,Qaug,Raug);

    Rww = rho_rww*eye(size(Bq,2));
    Rvv = rho_rvv*eye(size(Bq,2));
    Lbar = lqr(Aq',Cq',Bq*Rww*Bq',Rvv)';
    %compensators
    ac1=[Aq-Lbar*Cq-Bq*Kbar(1:size(Aq,1)) -Bq*Kbar(end);zeros(3,1)' 0];
    Br=[Lbar*0;1];

    acl = [Aq -Bq*Kbar; [Lbar;-1]*Cq ac1];
    bcl = [Bq*0;Br];
    ccl = [Cq Cq*0 0];
    ss_dofb = ss(acl,bcl,ccl,D);
    [y,t,x] = lsim(ss_dofb,u,t);
end
```

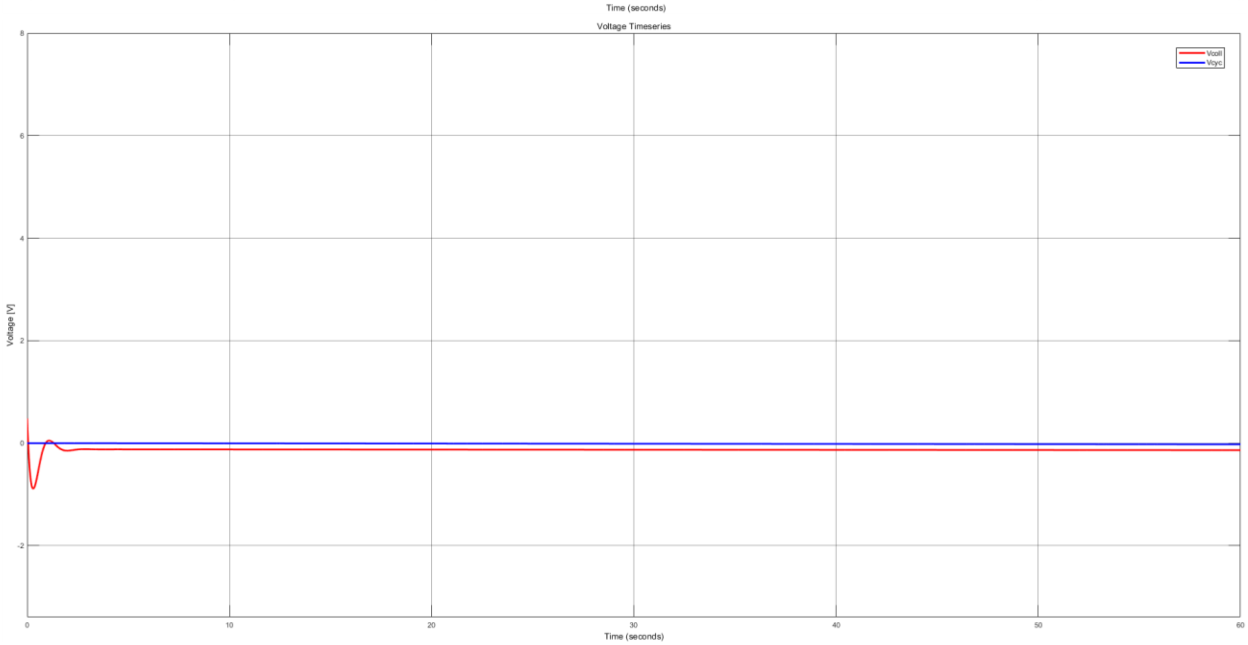
APPENDIX 7


Roll phi
Roll phi_hat
Pitch theta
Pitch theta_hat
Angle [deg]
Time (seconds)

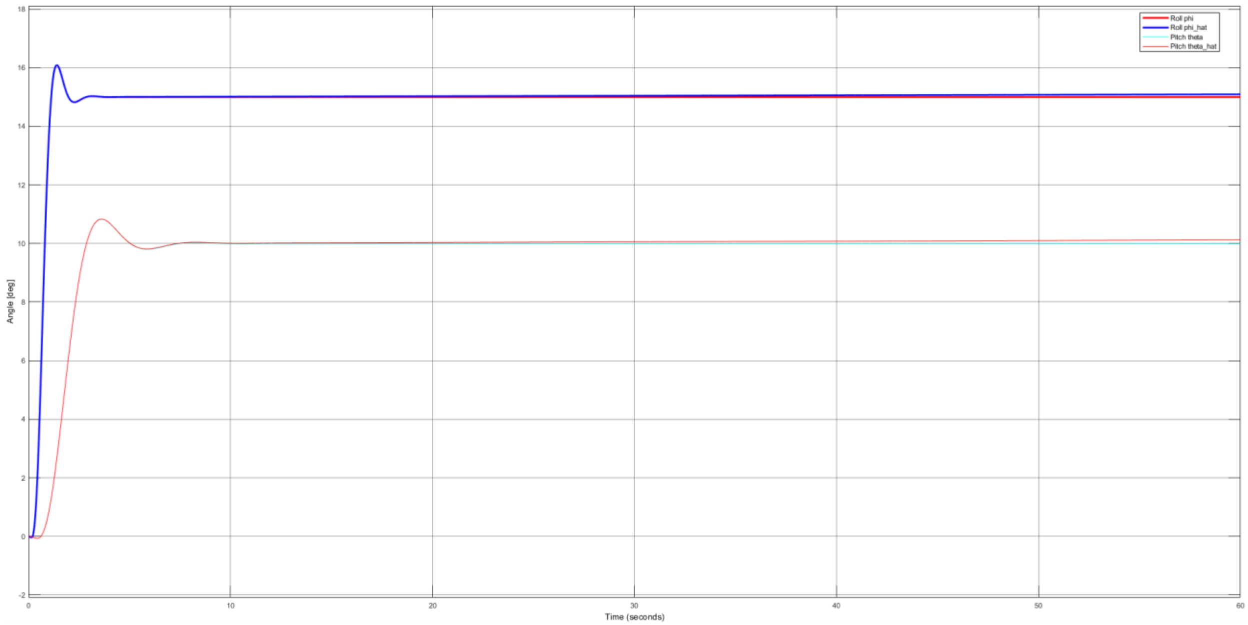

Voltage Timeseries
Vcoll
Vcyc
Voltage [V]
Time (seconds)

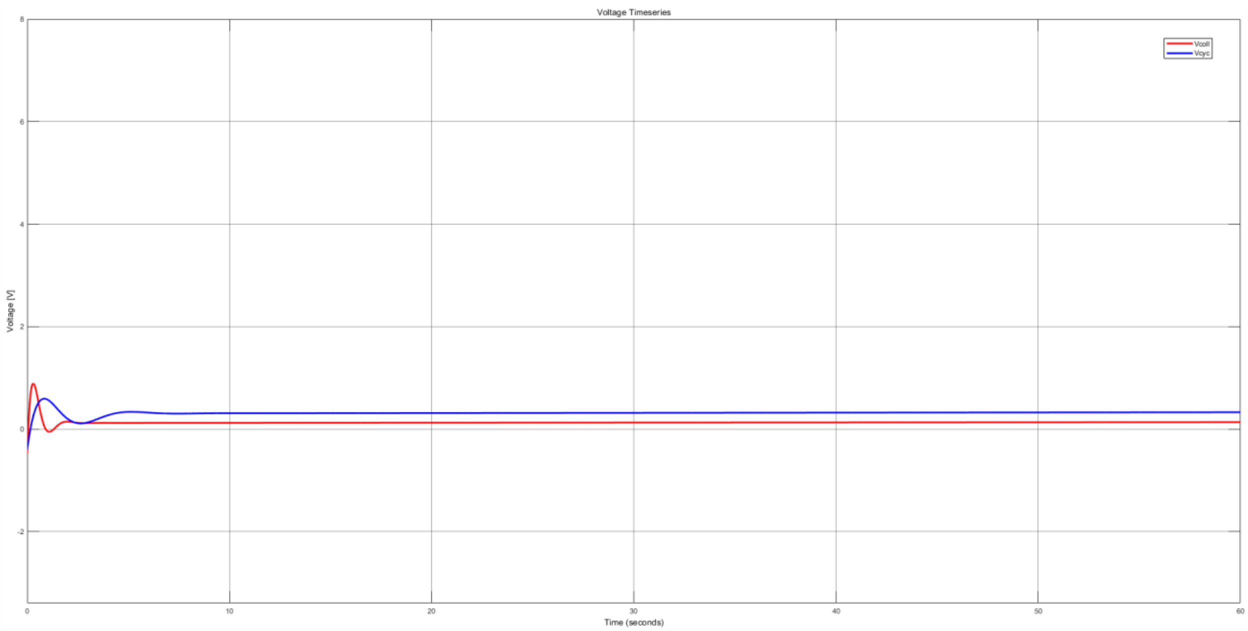
APPENDIX 8


Roll phi
Roll phi_hat
Pitch theta
Pitch theta_hat
Angle [deg]
Time (seconds)

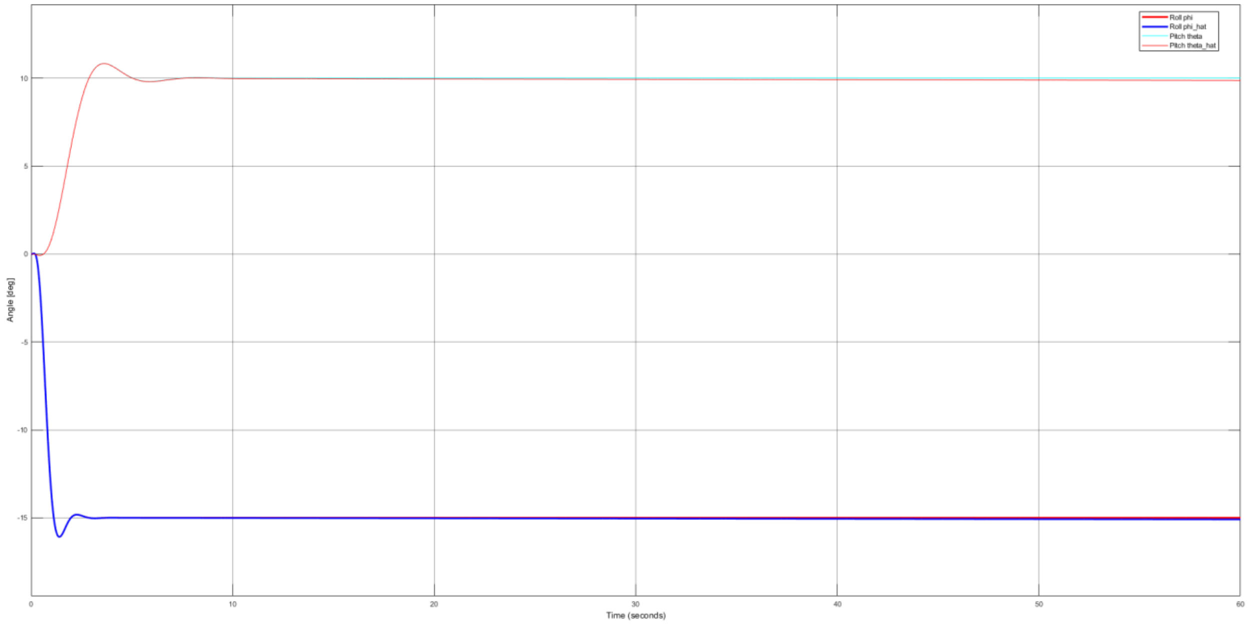

Voltage Timeseries
Vcoll
Vcyc
Voltage [V]
Time (seconds)

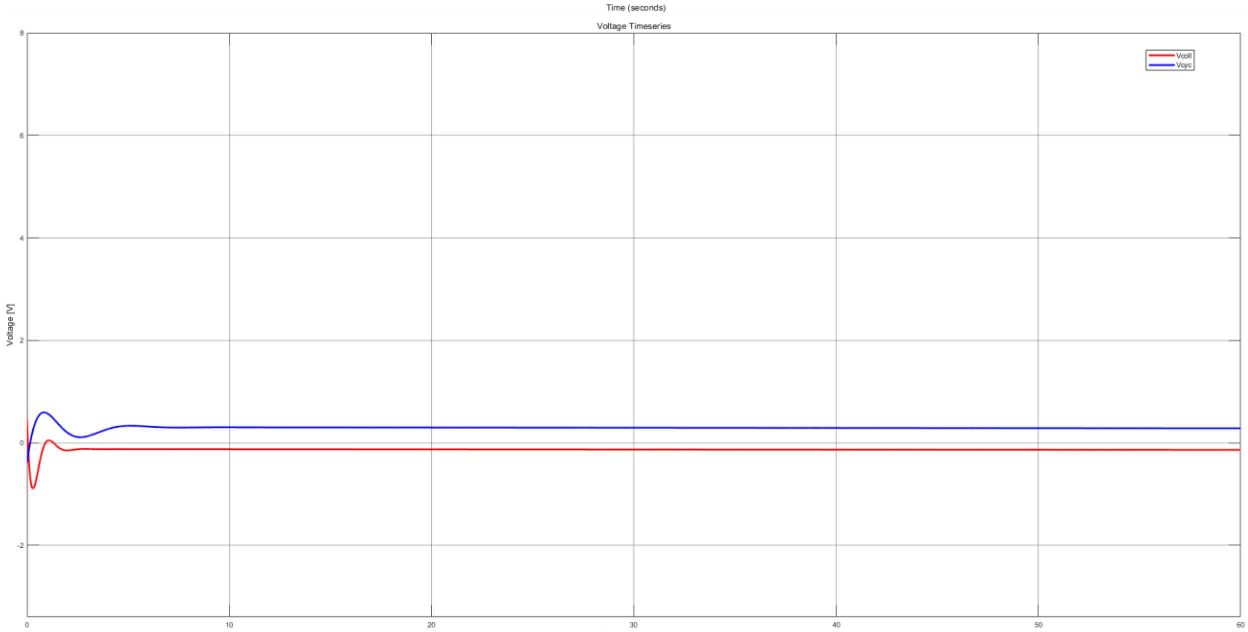
APPENDIX 9


Roll phi
Roll phi_hat
Pitch theta
Pitch theta_hat
Angle [deg]
Time (seconds)

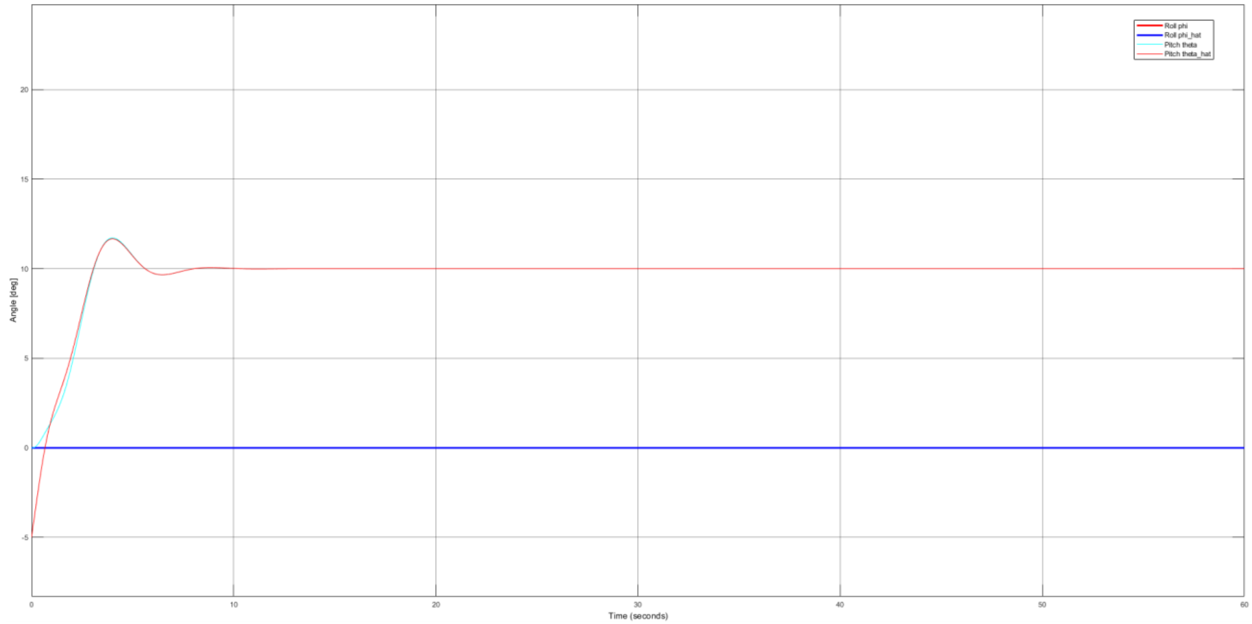

Voltage Timeseries
Vcoll
Vcyc
Voltage [V]
Time (seconds)

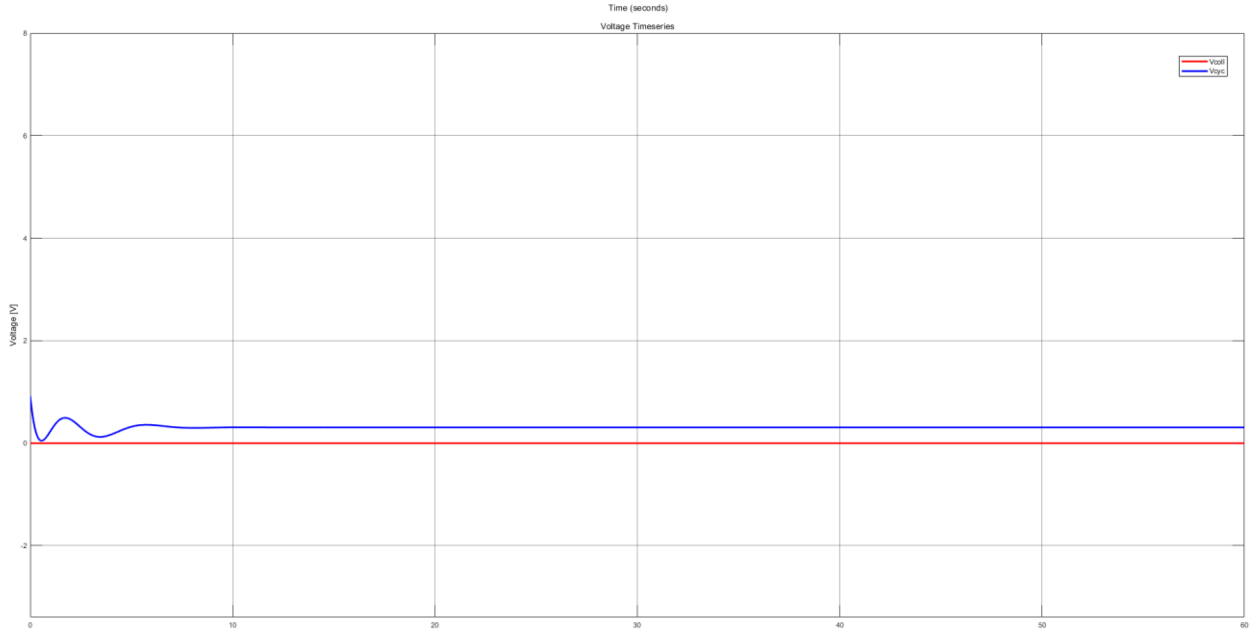
APPENDIX 10


Roll phi
Roll phi_hat
Pitch theta
Pitch theta_hat
Angle [deg]
Time (seconds)
10
5
0
-5
-10
-15
0
10
20
30
40
50
60

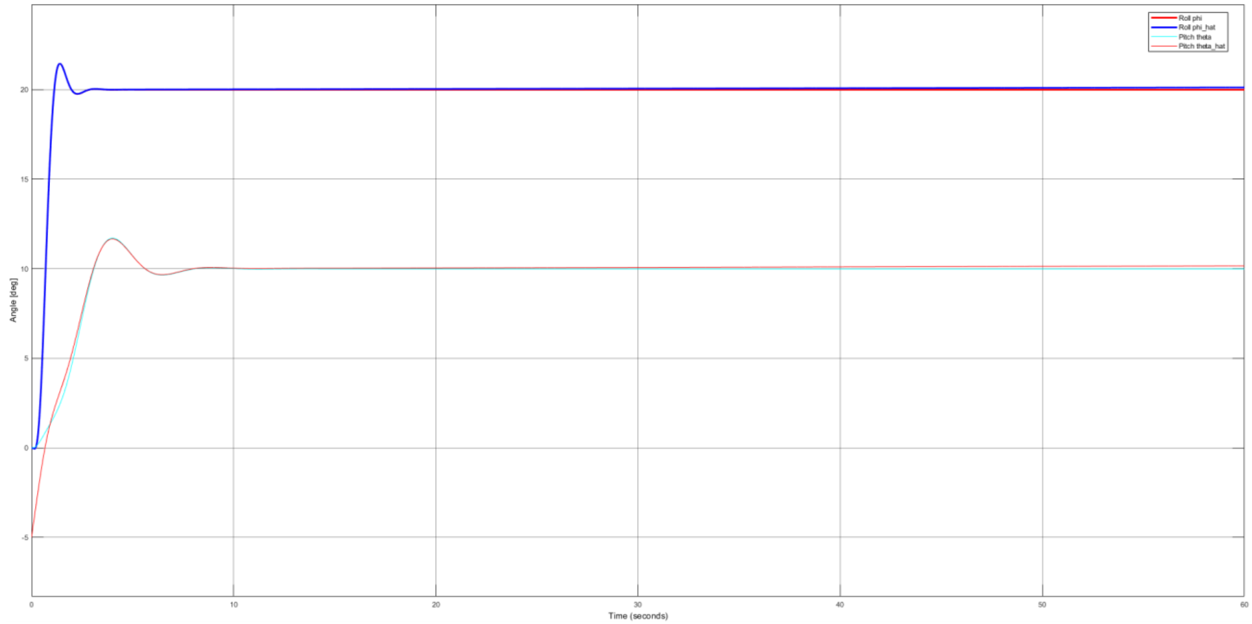

Voltage Timeseries
Vcoll
Vcyc
Voltage [V]
Time (seconds)
8
6
4
2
0
-2
0
10
20
30
40
50
60

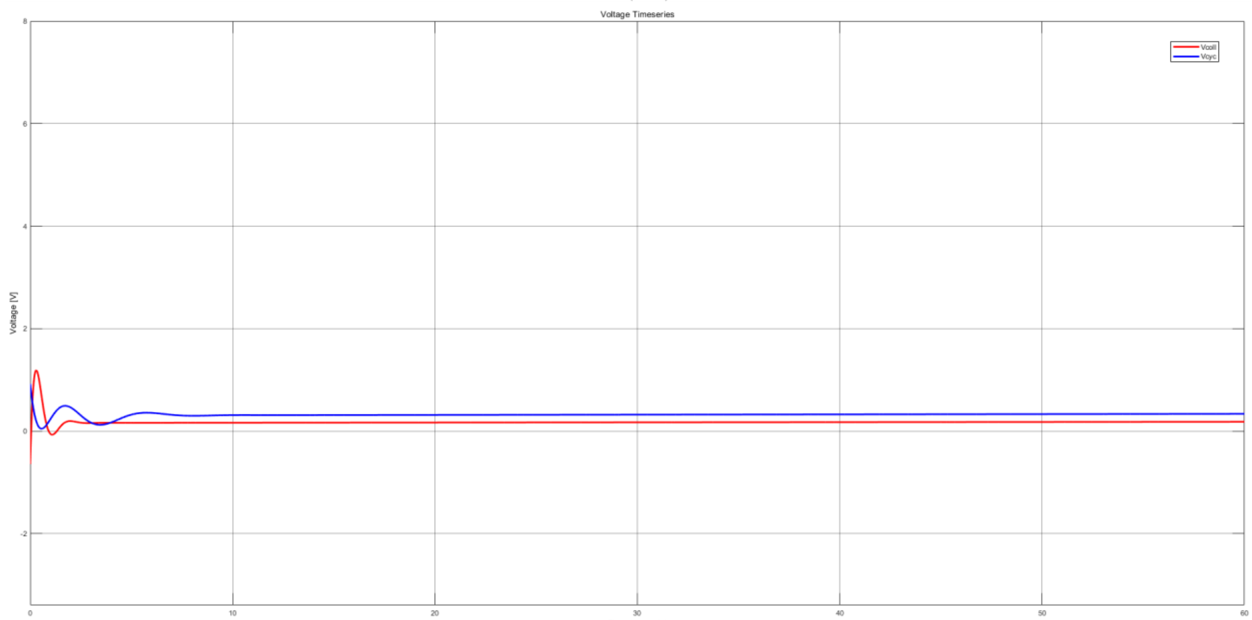
APPENDIX 11


Roll phi
Roll phi_hat
Pitch theta
Pitch theta_hat
Angle [deg]
Time (seconds)


Voltage Timeseries
Vcoll
Vcyc
Voltage [V]
Time (seconds)

APPENDIX 12


Roll phi
Roll phi_hat
Pitch theta
Pitch theta_hat
Angle [deg]
Time (seconds)


Voltage Timeseries
Vcoll
Vcyc
Voltage [V]
Time (seconds)