BRADLEY University

Laboratory Notebook

Smart Control of 2-DoF Helicopters

Glenn Janiak and Kenneth Vonckx

gjaniak@mail.bradley.edu, kvonckx@mail.bradley.edu

Beginning August 30, 2018

Contents

Thursday, August 30, 2018

1 Meeting Minutes

GJ
In today's meeting, Dr. Miah went over introductory materials. Dr. Miah, Kenneth, and I set up a weekly meeting time on Friday at 12:00 to 12:45 pm. We also set up designated lab times on Tuesday and Thursday at 9:00 to 11:00 am and 1:30 to 2:50 pm. Kenneth and I were also shown where will be our work station in the robotics lab (Jobst 254). We were informed by Dr. Miah on how file names should be formatted and how to use Latex for our online lab notebook and presentation slides. We were also informed that we will be using IPE for creating figures. We were instructed, by Dr. Miah, to create a common email that Kenneth and I both have access to read last year's final report and to create a presentation about the content of the report to be presented on Tuesday, September 4th.

Tuesday, September 4, 2018

1 Meeting Minutes

GJ
Met in robotics lab to present our review PowerPoint and to start on our lab work. We went to Dr. Miah's office to present our review. After the presentation, we talked about the tasks that need to be done. The current assignments are to read teaching materials from Quanser and research papers provided by Dr. Miah. We also talked with Mr. Mattus about us doing the project and what we needed for our lab station. We were also given general lab tasks that we will need to accomplish. They are to use LQR to control one helicopter, use ADP to control one helicopter, use LQR to control two helicopters, and use ADP to control two helicopters. After talking with Mr. Mattus, we stored small electronics in cabinet and had to go to class. We went back to the lab hoping that setup of lab space was completed; however, it was not so could not do any implementation testing with Quanser AERO. Kenneth and I decided to leave the lab and work on the reading materials for next lab date.

Thursday, September 6, 2018

1 Meeting Minutes

GJ
Spent the lab time working with the Quick Start Guide and the Integration Workbook to familiarize ourselves with the Quanser AERO. Completed both the Quick Start Guide and the Integration Workbook. Because of class time had to leave before talking with Dr. Miah about LQR overview. Scheduled to talk with Dr. Miah about LQR design during weekly.

2 Quick Start

GJ
Tested both Quanser AERO Helicopter's using the Quick Start Guide. They both worked as expected according to the guide.

3 Integration Workbook

GJ
From the integration workbook we learned how to set up a basic Simulink model to control the AERO. Starting by adding the HIL Initialize block and setting the board model to "quanser_aero_usb". Then under the QUARC tab we "set default options" which set the mode for external use. After this, we then built, connected, and run the initialization to make sure the setup was correct. After stopping the run, we then added a HIL Write Analog block and a constant block to send a constant voltage to motor #0. Added a HIL Read Other, Gain, and Display blocks to show the RPM of the motor. We calculated a gain of 60/2048 to convert the counts/sec to RPM. After taking RPM data based on voltage data, we deciphered the relationship equation to be about Equation 1.

$$RPM = 220 * V \tag{1}$$

KV
In order to control the helicopter through Simulink, first add an HIL Intialize Block by going to "View", "Quanser", and "Data Aquisition". Check the USB connection which should be set to "quanser_aero_usbo". Also "Set Default Options" for Quanser which changes simulink from simulation to lab mode. An "HIL Write Analog" block is required to output to the motor. To run the program: Build, Connect, then Run.
We noticed that for the pitch motor, a positive value applies a positive voltage which spins the pitch motor counter-clockwise from the side opposite of the motor. The thrust points down towards the motor. The yaw motor is the opposite in that a positive value will spins the motor clockwise from the side opposite of the motor which creates thrust away from the motor.
An "HIL Read Output" block allows us to read the number of counts per second and when mutiplied by a gain value gives us RPM. In order to determine our gain value, we needed to find the resolution of the encoder. This was found by looking in the Quanser Aero User Manual [1] to be a value of 2048 for the pitch encoder in quadrature as shown in. Figure 1.

Symbol	Description	Value
DC Motor		
V_{nom}	Nominal input voltage	18.0 V
τ_{nom}	Nominal torque	22.0 mN-m
ω_{nom}	Nominal speed	3050 RPM
I_{nom}	Nominal current	0.540 A
R_m	Terminal resistance	8.4 Ω
k_t	Torque constant	0.042 N-m/A
k_m	Motor back-emf constant	0.042 V/(rad/s)
J_m	Rotor inertia	4.0×10^{-6} kg-m^2
L_m	Rotor inductance	1.16 mH
Aero Body		
M_b	Mass of body	1.075 kg
D_m	Center of mass	-7.59 mm
J_p	Pitch inertia	2.15×10^{-2} kg-m^2
J_y	Yaw inertia	2.37×10^{-2} kg-m^2
D_t	Thrust displacement	15.8 cm
Motor and Pitch Encoders		
	Encoder line count	512 lines/rev
	Encoder line count in quadrature	2048 lines/rev
	Encoder resolution (in quadrature)	0.176 deg/count
Yaw Encoder		
	Encoder line count	1024 lines/rev
	Encoder line count in quadrature	4096 lines/rev
	Encoder resolution (in quadrature)	0.088 deg/count
Amplifier		
	Amplifier type	PWM
	Peak Current	2 A
	Continuous Current	0.5 A
	Output voltage range (recommended)	$\pm$18 V
	Output voltage range (maximum)	$\pm$24 V

Figure 1: *Quanser Aero System Parameters*

Equation 1 was given to us by the Integration lab workbook [2], however we took data from the encoder based on an applied voltage to support Equation 1 as shown in Table 1

RPM	V	RPM/V
130	1	130
430	2	215
900	4	225
1800	8	225
3600	16	225

Table 1: *Integration Lab Data*

Friday, September 7, 2018

1 Meeting Minutes

GJ
Had our weekly meeting with Dr. Miah. He introduced us to the LQR algorithm. Kenneth and I were assigned to read the three supplied research paper, find two more, and write a one paragraph summary of each paper. We were also assigned to reproduce the LQR MATLAB code that Dr. Miah will supply us with.

2 Research Paper Summaries

GJ
These are the summaries to the research papers we had to summarize.

2.1 Decentralized discrete-time neural control for a Quanser 2-DOF helicopter

KV
This research paper [3] discusses the use of Neural Networks in controlling helicopter dynamics due to their ability to learn and adapt to non-linear systems. Kalman Filters are suggested as one of the training methods for a Neural Network. The Kalman Filter estimates the states of the system which are used to determine the weights of the Neural Network. In the case of the Quanser Aero, the states are the pitch angle, yaw angle, pitch velocity, and yaw velocity.
The paper then goes into detail about the matamatical expressions for higher-order neural networks (HONNs) and Extended Kalman filters (EKFs). The Quanser Aero system is also discussed. The author comes up with a method for training a HONN with an EKF to train the system and ensure that it is bounded and then simulates the results.

2.2 4DOF quadcopter: development, modeling and control

KV
This article [4] starts by explaining the significance of control systems in helicopter and quadcopter platforms to allows them to be autonomous. The author goes into detail about

modeling the kinematics and dynamics of UAVs (unmanned aerial vehicle). This includes the roll, pitch, and yaw movements capable of the vehicle and the forces that act against it. A stationary prototype is then constructed to move in 4 degrees of freedom. A gyroscope is used to measure the orientation of the prototype. Some Internal Measurement Units (IMUs) also contain accelerometers, magnetometers, and barometers. A Python program was used to calibrate the sensors and Arduiuno was used as the microcontroller. Brushless motors have several advantages over brushed including fast response, constant torque, and impressive speeds. PID, LQR, Sliding Mode, and Integrative Sliding Mode with LQR Controllers are tested.
The PID controller is tested both with and without the derivative term. Without the derivative term, the output is less noisy. The LQR method is much more clean. The Sliding Window method has similar results. The author also tries using trajectory tracking, which we will also be using in this lab.

2.3 2-DOF helicopter controlling by pole-placements

GJ
In the article [5], their objective was to test the pole placement control technique of a 2-DOF helicopter. It starts off by explaining the mathematical modeling of the system and figuring out the state-space system equations. The authors then talk about their design of the helicopter parameters, from what the body of the helicopter is made of, to the safety designs put in place. The next part was the controller design. They start by considering the state-space system equations and the desire to get a null steady-state error, and with these things in mind they come up with equation 19 in [5]. The pole placement design is used to come up with the values for the K_{amp} matrix based of the desired closed-loop poles. They got the values by using the MATLAB command place. After the design discussion, [5] talks about the simulation results.

2.4 Data-driven Adaptive Optimal Output-feedback Control of a 2-DOF Helicopter

KV
This article [6] studies Quanser's 2 DOF helicopter and the potential to apply an ADP approach to it. A model for the system and an algebraic Riccatio equation (ARE) is solved for by policy iteration (PI) and Output-feedback ADP. They then create a Robust Controller design for the ADP algorithm first by creating a continuous time model and then convert it to discrete time by periodically sampling. When simulating, $Q = \mathrm{diag}\{200, 150\}$, $R = I_2$ is used. Their simulations had good results.

2.5 Fuzzy Control with Estimated Variable Sampling Period for Non-Linear Networked Control Systems: 2-DOF Helicopter as Case Study

GJ
The article, [7], uses a fuzzy controller on non-linear networked control systems. The system uses variable sampling periods which including time delays and packet loss. After going into more detail about the variable sampling period, [7] then goes into describing the fuzzy model mathematically. After the model is described, the fuzzy controller was proposed and described. This included stating and proving a theorem. After finishing the fuzzy controller it was tested, in a case study, on a 2-DOF helicopter using Ethernet. The case study involved a multi-input-multi-output (MIMO), non-linear, open-loop unstable, and time varying system. The last thing that [7] talks about is the results of the case study.

Tuesday, September 11, 2018

1 Meeting Minutes

GJ
We started today by implementing the LQR algorithm provided by Quanser. We have been assigned to run the algorithm using square, sine, and constant inputs. We had to rework the figures to uphold professional level of quality.
KV
Our Lab Director Mr. Christopher Mattus has replaced the high efficiency 2 blade propellers with the standard 10 blade propellers on the new Quanser Aero (Aero #2) which we will be using to implement the LQR algorithm. He also noticed the Aero #1 had vibrations when running. After investigation, he noticed that a spacer had been placed between the frame and grill of the propeller to prevent the blades from colliding with the grill. He removed these and also noticed that the propellers appear to be 3D printed vs. the propellers of Aero #2 appear to be cast. This could be due to a change in Quanser's manufacturing process as the Aero platform became more popular and a need arose to produce the propellers more efficiently. The 3D printed propellers may also be the culprit as to why Aero #1 is unbalanced.

2 LQR Using USB

GJ
We did section 4 of the Aero 2 DOF Laboratory Guide [8]. This section covers the LQR algorithm in Simulink. The simulation figures of the square signal can be seen in Figure 2, Figure 3, Figure 4, and Figure 5. After confirming that the LQR works, we decided to change the Simulink to be able to control what pitch and yaw we wanted. We exchanged the signal generator blocks with constant 1 blocks and changed the value of the amplitude gains to control what angle we wanted. Figure 1 is the Simulink model that we used with the extra constant blocks for specific pitch and yaw control. We created the figure template using MATLAB. Next meeting will will repeat the signal figures to cover sine wave and constant inputs for LQR, then will work on the LQG algorithm for square, sine, and constant inputs.

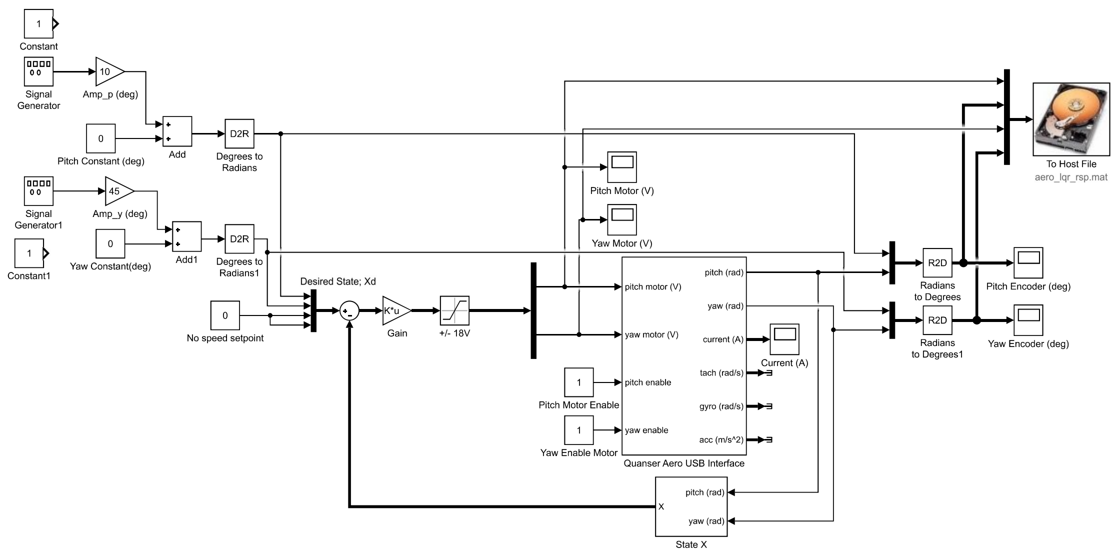

Figure 1: *LQR USB Simulink Model*

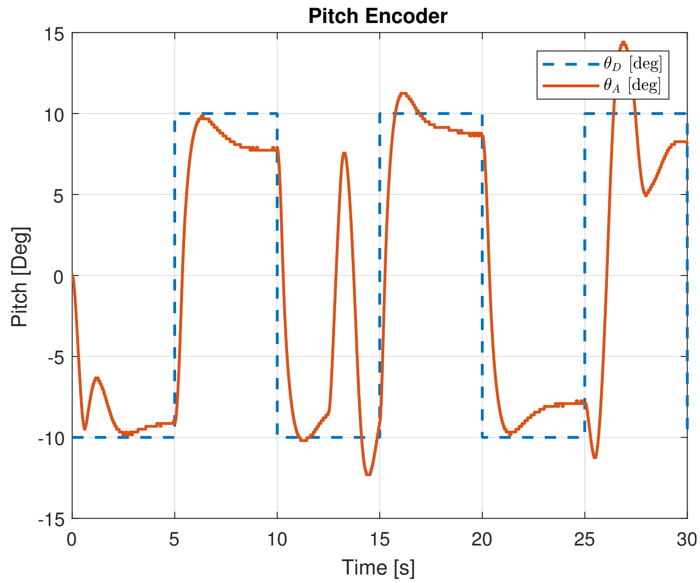

Figure 2: *LQR USB Pitch Encoder w/ Square Wave*

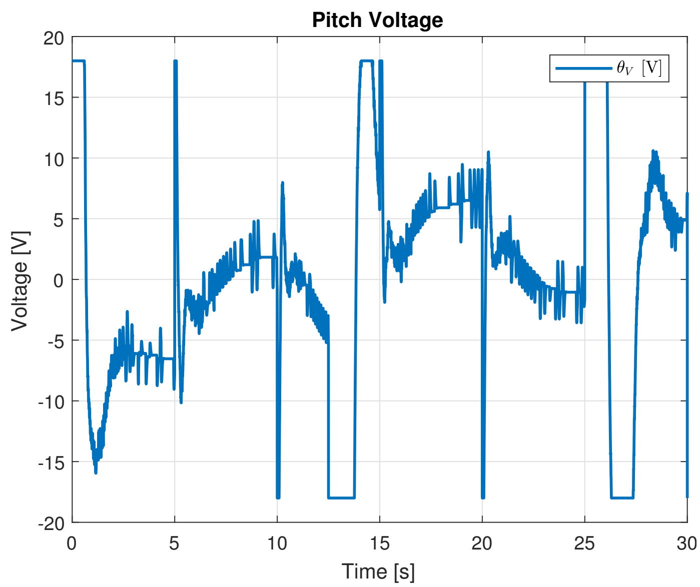

Figure 3: *LQR USB Pitch Motor w/ Square Wave*

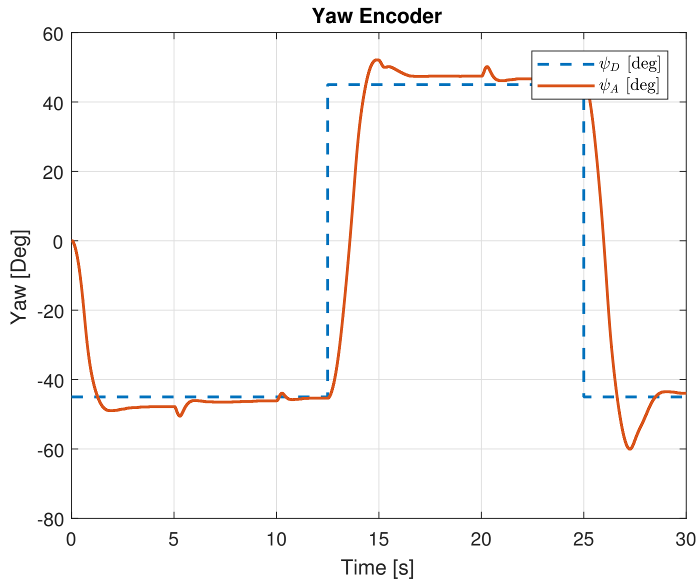

Figure 4: *LQR USB Yaw Encoder w/ Square Wave*

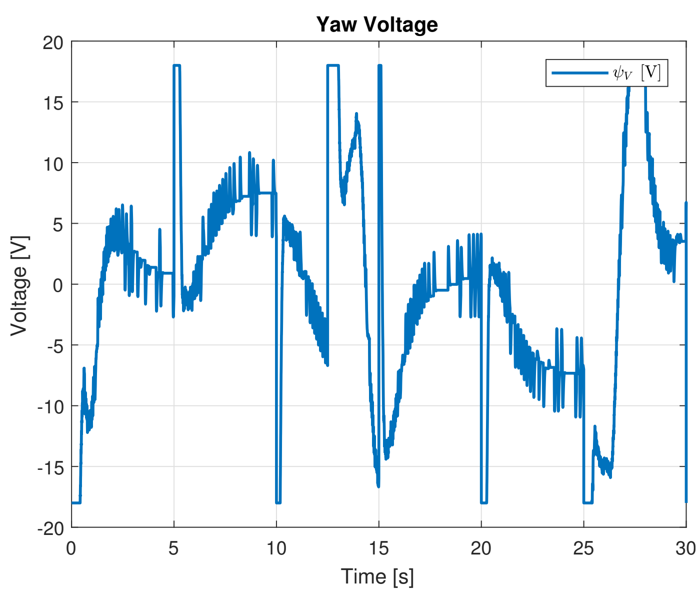

Figure 5: *LQR USB Yaw Motor w/ Square Wave*

Thursday, September 13, 2018

1 Meeting Minutes

GJ
We are tasked with finishing up the LQR testing using USB connection. We will be repeating what we did last lab day except by replacing the input square wave with a sine wave and a constant input. After that we will be testing the LQG algorithm in the same manner as the LQR.

2 LQR Using USB Cont.

GJ
We started today by opening up the LQR Simulink we were working on last lab day and switched the signal generator from square wave to sine wave. The outputs can be seen in Figure 1, Figure 2, Figure 3, and Figure 4. After we saved the figures using MATLAB we went back to the Simulink and change the input from a sine wave to a constant block of value 1. The simulation outputs can be seen in Figure 5, Figure 6, Figure 7, and Figure 8.

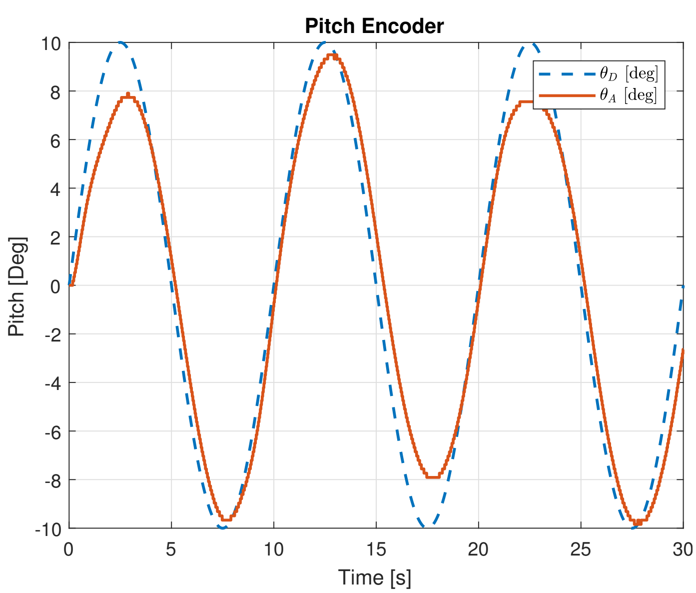

Figure 1: *LQR USB Pitch Encoder w/ Sine Wave*

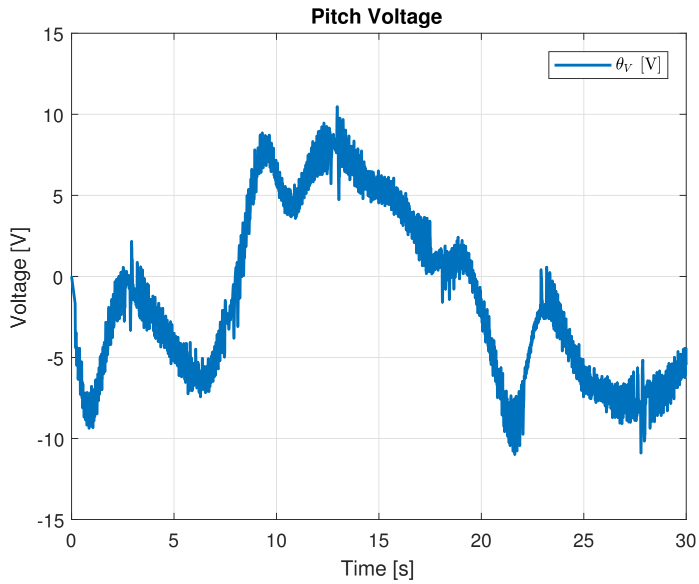

Figure 2: *LQR USB Pitch Motor w/ Sine Wave*

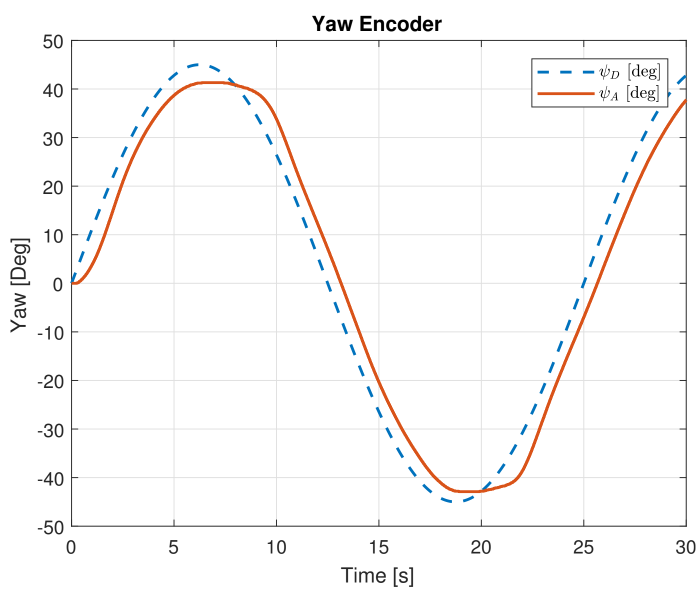

Figure 3: *LQR USB Yaw Encoder w/ Sine Wave*

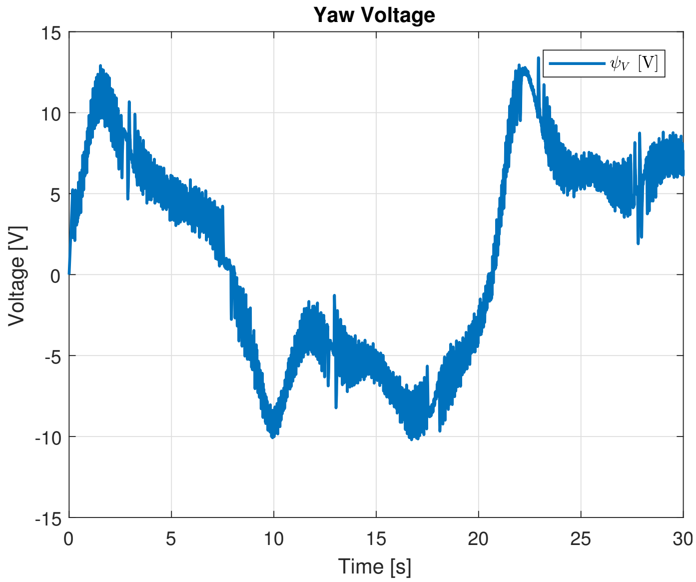

Figure 4: *LQR USB Yaw Motor w/ Sine Wave*

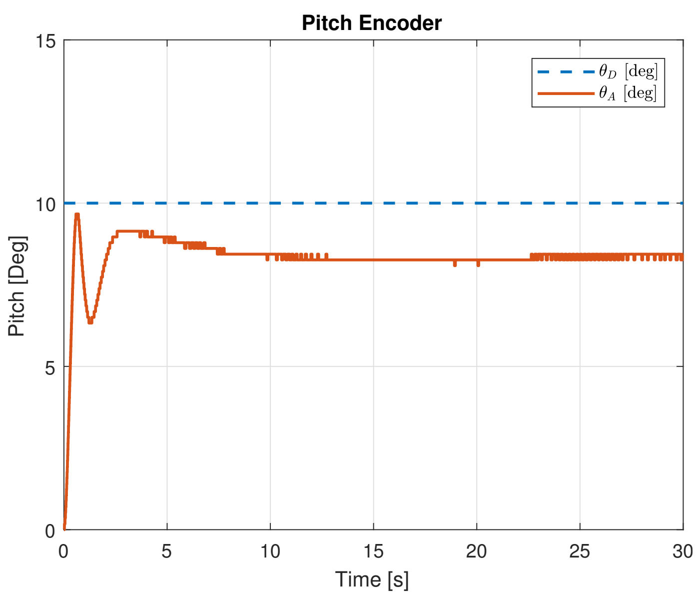

Figure 5: *LQR USB Pitch Encoder w/ Constant Input*

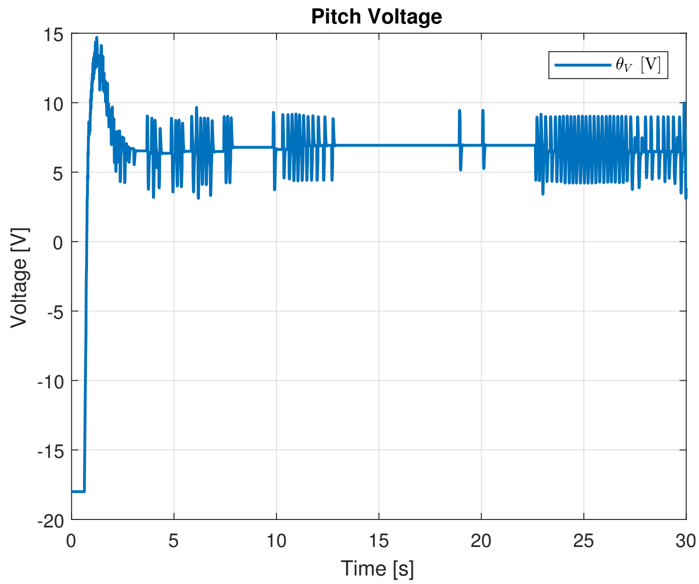

Figure 6: *LQR USB Pitch Motor w/ Constant Input*

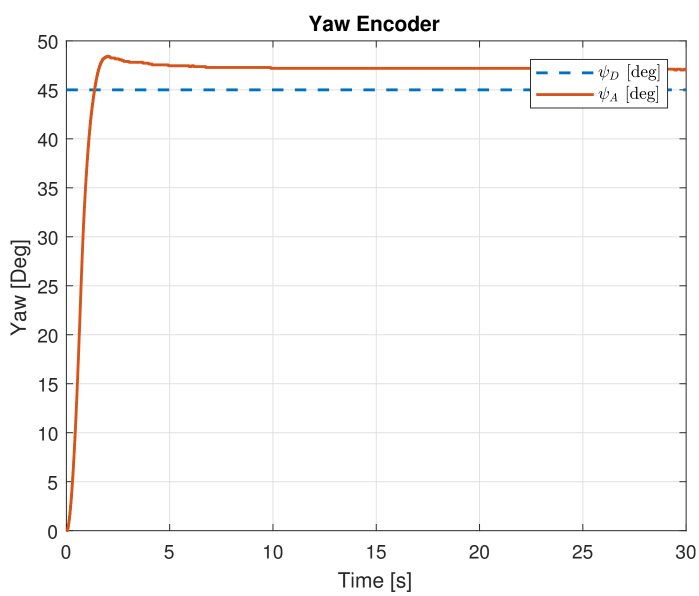

Figure 7: *LQR USB Yaw Encoder w/ Constant Input*

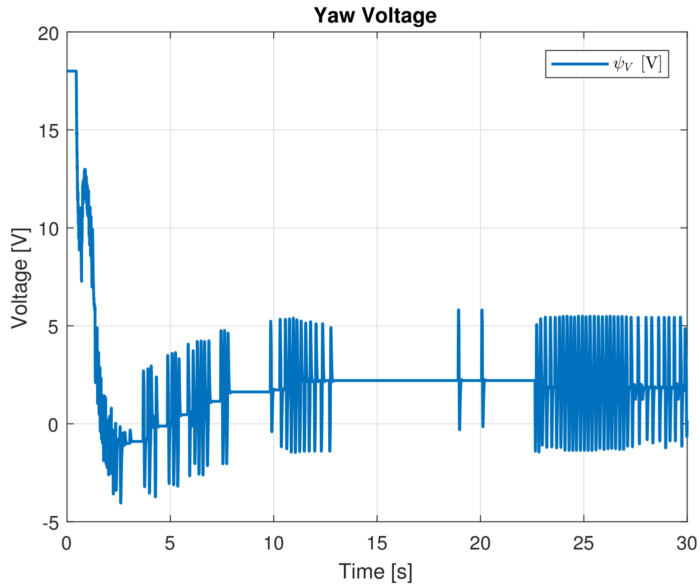

Figure 8: *LQR USB Yaw Motor w/ Constant Input*

3 LQG Using USB

KV
We repeated the test of the Quanser Aero for constant, square, and sinusoidal inputs using a LQG algorithm following section 5 of the Lab Guide[8]. The block diagram for the LQG algorithm made in Simulink is shown in Figure 9. This method required us to lock the yaw axis and put the platform in half-quadcopter mode.
The outputs for the square wave output is shown in Figure 10, Figure 11, Figure 12, and Figure 13. The LQG algorithm did not preform nearly as well as LQR for a square wave input and the over/undershoot was much greater. We also notice an unusual vibration which could be due to the system being unbalanced because the yaw motor is disabled. If the yaw axis is not locked, the yaw of the system still changes.
The outputs for the sine wave output is shown in Figure 14, Figure 15, Figure 16, and Figure 17. The LQG algorithm also struggled with a sine input and was roughly on average 5 degrees farther from the desired position than LQR. The output voltage also saturated much more often.
The outputs for the constant voltage is shown in Figure 18, Figure 19, Figure 20, and Figure 21. The LQG algorithm begins to approach the constant value set to 10 degrees but then falls. Strangely, the output voltage oscillates positively and negatively. We would expect small positive oscillations to achieve steady state, however this behavior leads us to believe that there are unstable elements to the LQR algorithm.

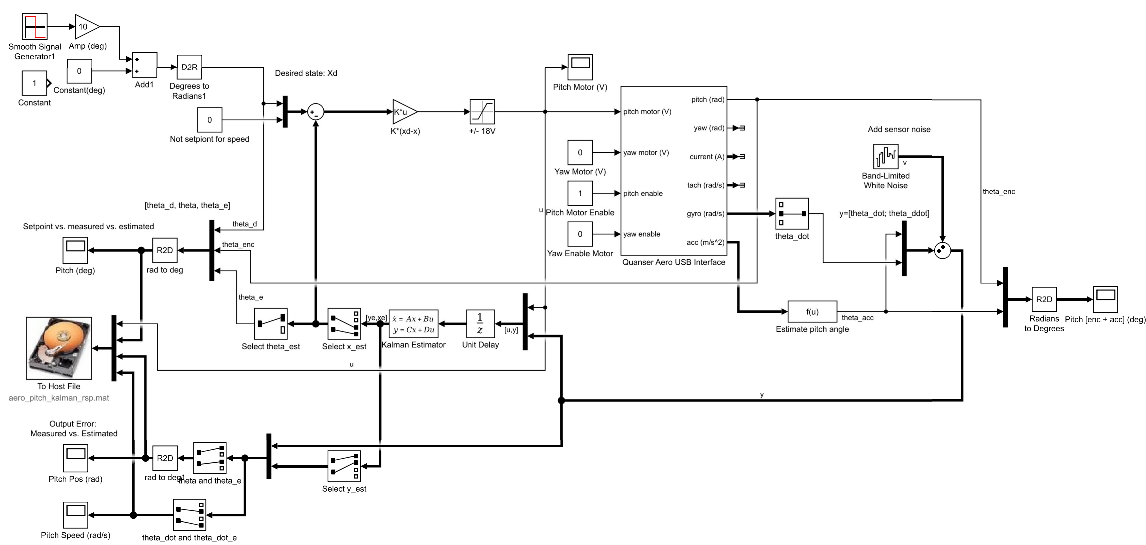

Figure 9: *LQG USB Simulink Model*

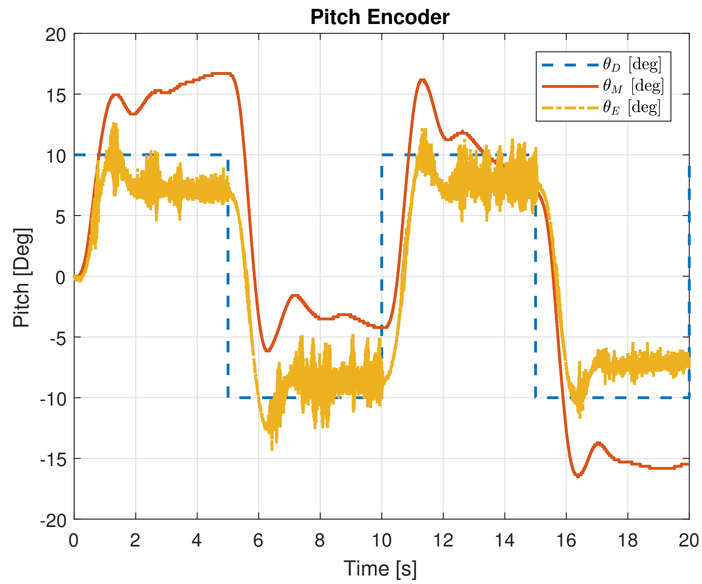

Figure 10: *LQG USB Pitch Encoder [deg] w/ Square Wave*

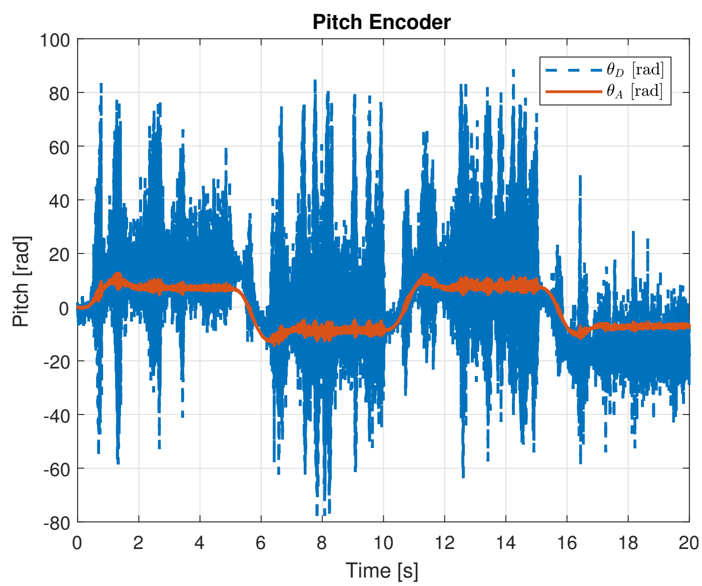

Figure 11: *LQG USB Pitch Encoder [rad] w/ Square Wave*

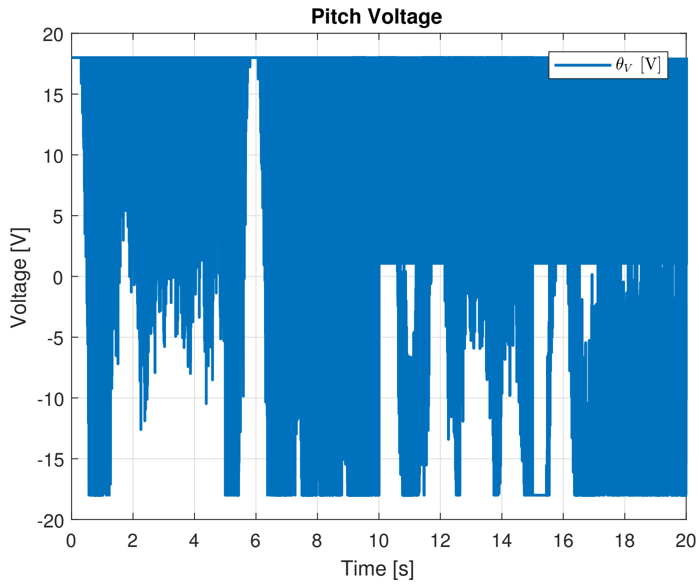

Figure 12: *LQG USB Pitch Voltage w/ Square Wave*

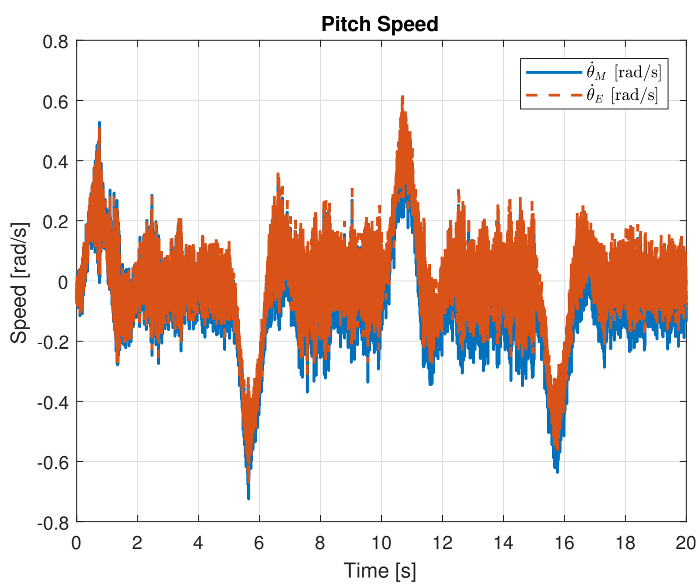

Figure 13: *LQG USB Pitch Speed w/ Square Wave*

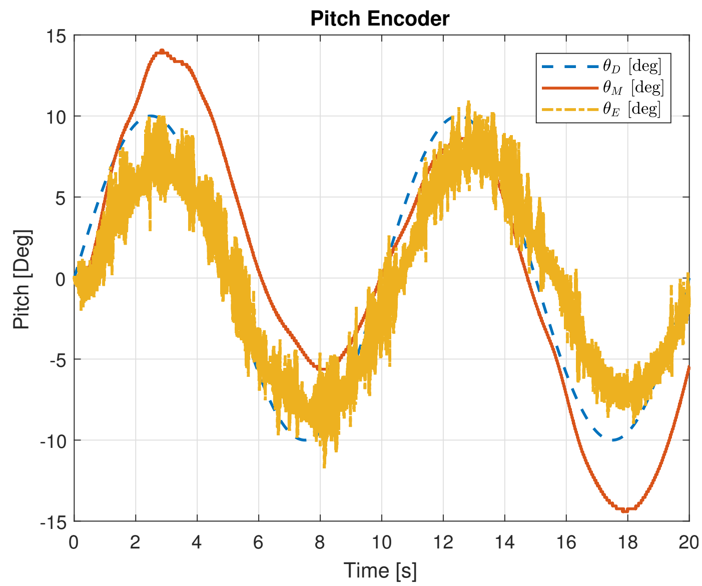

Figure 14: *LQG USB Pitch Encoder [deg] w/ Sine Wave*

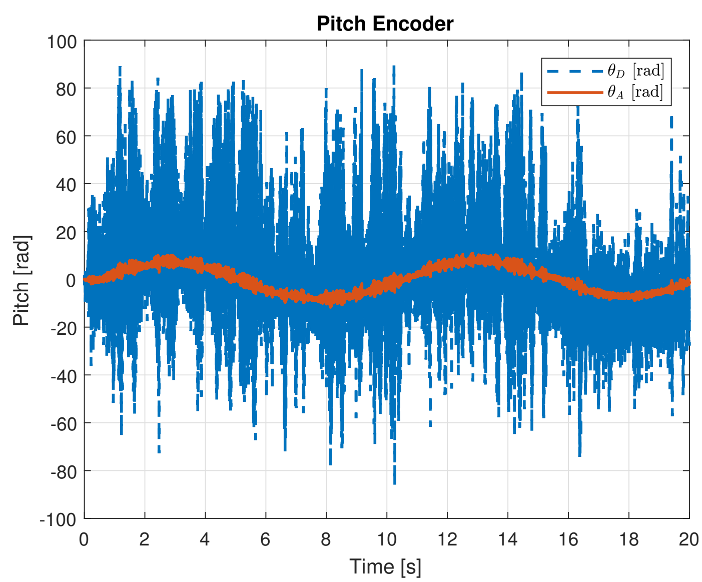

Figure 15: *LQG USB Pitch Encoder [rad] w/ Sine Wave*

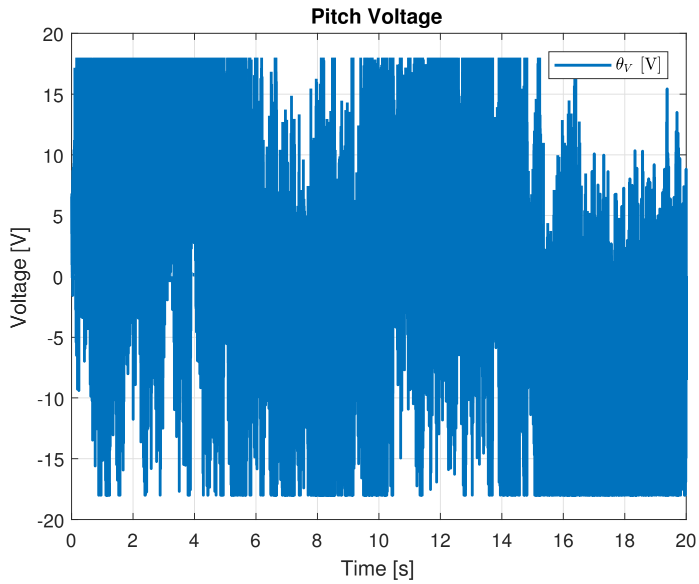

Figure 16: *LQG USB Pitch Voltage w/ Sine Wave*

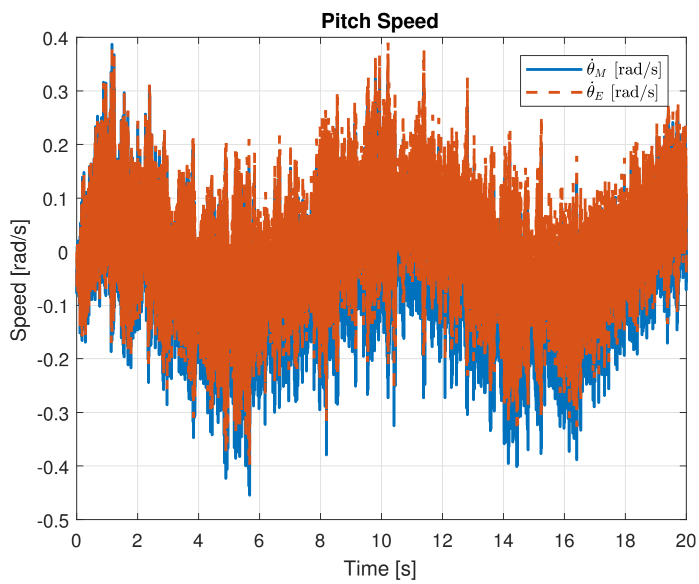

Figure 17: *LQG USB Pitch Speed w/ Sine Wave*

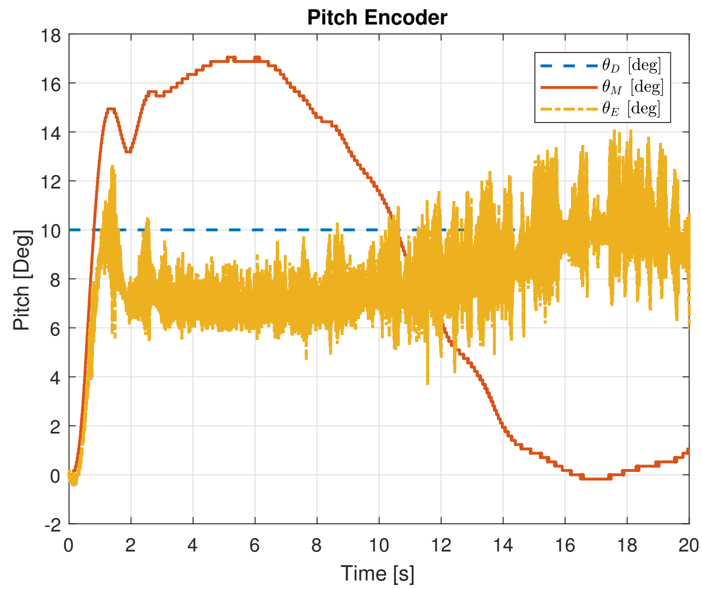

Figure 18: *LQG USB Pitch Encoder [deg] w/ Constant Input*

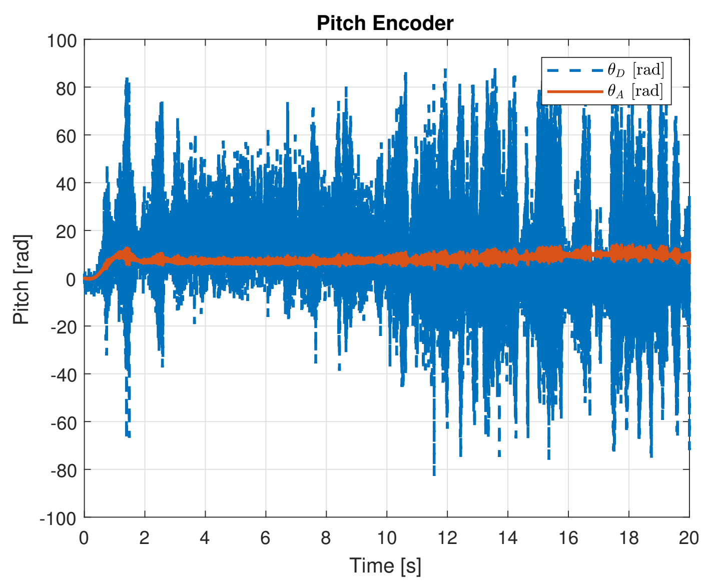

Figure 19: *LQG USB Pitch Encoder [rad] w/ Constant Input*

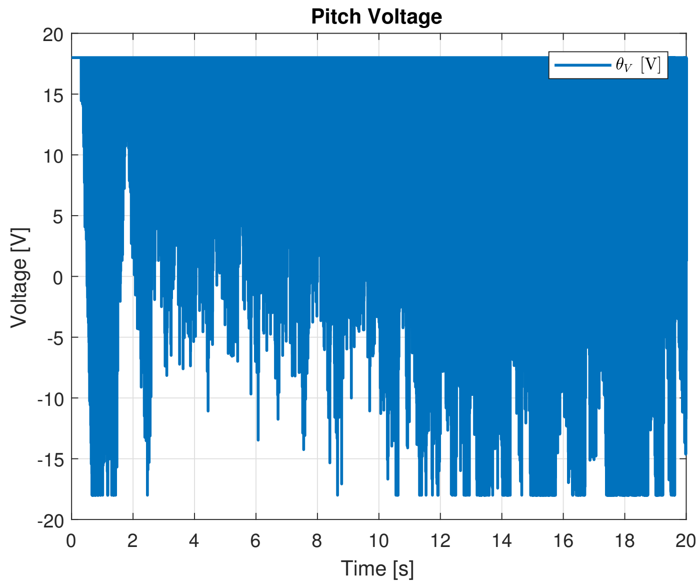

Figure 20: *LQG USB Pitch Voltage w/ Constant Input*

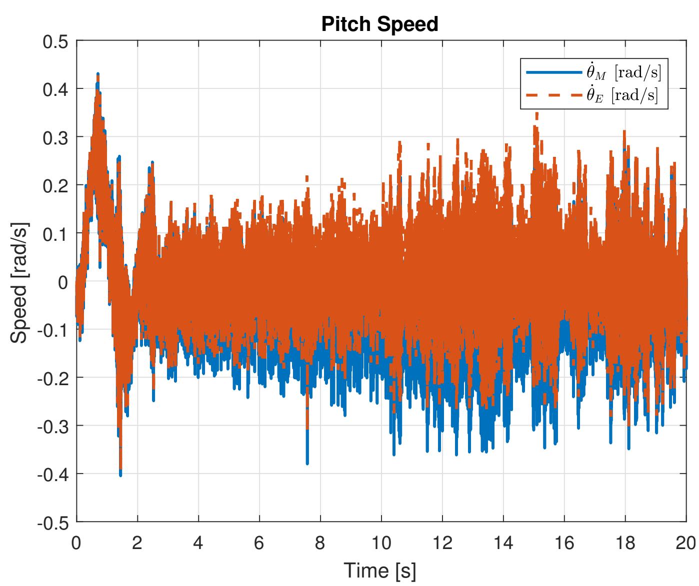

Figure 21: *LQG USB Pitch Speed w/ Constant Input*

Friday, September 14, 2018

1 Meeting Minutes

GJ
For our weekly meeting with Dr. Miah, we started off by talking about communication with the Quanser representatives, Roberto Chan and Arian Panah. We have to email them about the extra Qflex2 Embedded panel for Aero #2, new 10 blade propellers for Aero #1, and documentation for the Embedded panel. We also talked about Ken and my tasks to be completed during our next lab day. We have to make graphs that include the desired angle and actual angles for both LQR and LQG algoritms for constant, square, and sine inputs, and for θ, ψ, $\dot{\theta}$, and $\dot{\psi}$. After doing this, we are then to repeat what we did using the LQR and LQG except instead of using the USB panel, we are to use the Embedded panel with the Raspberry Pi 3.

Tuesday, September 18, 2018

1 Meeting Minutes

GJ
Began today with the intention to finish the LQR and LQG implementation on the USB panel. Received a reply from Arian over the weekend, saying that the Qflex2 Embedded panel does not come with the Quanser Aero anymore; however, due to our support of Quanser we will be receiving free propellers and Qflex2 Embedded panel.

2 Combination Graphs

GJ
We added blocks to be able to save the speed information on the LQR Simulink Model. We are having trouble adding Yaw control to the LQG Simulink model. Sent an email to Arian hoping for help in what needs to be done. Asking for new documentation, physical noise while running LQG, and help with combining yaw to LQG model.

Thursday, September 20, 2018

1 Meeting Minutes

GJ
Began today looking for the yaw controlled half quadcopter Simulink and MATLAB designs, so as to compare them to our nonfunctional pitch and yaw controlled LQG design. We were told about the yaw controlled lab guide by Michel Levis from Quanser. However, the associated MATLAB code and Simulink files were not in the Quanser materials on the Google Drive. Luckily, Michel supplied us with links to all the files where we found the desired MATLAB and Simulink files for quadcopter yaw motion.

2 Lab Work

KV
Instead of attempting to consolidate certain elements in our block diagram and representing the gain for both pitch and yaw using one matrix, we separated the pitch and yaw to have their own error estimating Kalmen Filter and gain values. The model is now able to successfully compile, however the yaw position is inaccurately reporting the position and is displaying erratic maneuvering. Next week, we plan to see if is the state-space model for the yaw is causing problems and if we can get the yaw position to stay fixed at a given constant value.

Friday, September 21, 2018

1 Meeting Minutes

KV
Today we have decided to continue with implementing LQR on the Raspberry PI and Android smart-phone for the next couple of lab days. This is because the way we have implemented LQG will not work in the way in which we can have a fair comparison with LQR. We will continue to research LQG outside of lab for pitch and yaw coupling.
Dr. Miah has agreed to install Latex and IPE on the lab computers as we being to create more formal documents and figures and well as Putty so we can connect to the Raspberry PI.

Tuesday, September 25, 2018

1 Meeting Minutes

GJ
Today, we will start using the Raspberry PI 3 as an interface with the AERO. First, we will learn how to connect the Raspberry PI 3. Next, we will download two support packages that last years group used for Raspberry PI connection for MATLAB and Simulink. After that, we plan on formatting the LQR algorithm so we can test it on the AERO using the Raspberry PI 3.

2 Raspberry PI 3

KV
We created a serial connection to the Raspberry PI using a serial to USB adapter. We then used Putty to connect to COM Port 3 on Desktop #2. The speed was set to 115200 (baud), 8 data bits, 1 stop bit, no parity, and no flow control. Once connected, we logged into the PI. We were able to view last years files, however no c files were present. We created a new directory where all of our files for this project will be kept.

3 Support Packages

GJ
We need to have two support packages to be installed for this part of the project and a third package to be installed for Android use. We sent an email to Mr. Mattus, so these packages can be installed. They are the Raspberry PI Support Package for MATLAB, Raspberry PI Support Package for Simulink, and the Android Support Package for Simulink.

Thursday, September 27, 2018

1 Meeting Minutes

GJ
For today we had to focus on completing the draft of the problem statement for ECE497 deliverable.

Monday, October 1, 2018

1 Meeting Minutes

GJ
Had a meeting with Dr. Miah. We talked about materials we need for Raspberry PI connections. We sent an email to Mr. Mattus about a wireless adapter for the lab PC's, setting up the PC's for this wireless connection, and to question the support packages that were added to MATLAB 2017A instead of 2018A. Also we were informed that we will have to write a conference paper covering LQR and LQG.

Tuesday, October 2, 2018

1 Meeting Minutes

GJ
Received an email fro Mr. Mattus stating that Quarc is compatible with 2017A, not 2018A, so Dr. Miah stated we can continue while using 2017A of MATLAB. Mr. Mattus also informed us that he does have USB WiFi adapters that we can use but for the PC configurations we will have to tell him what we will be using the wireless connection for. We met up with Mr. Mattus to discuss the USB adapters. After meeting with Mr. Mattus, he told us to find the Mac address for the wireless network card inside the Raspberry 3. This address is: b8:27:eb:bd:92:c6. We tried to test the raspberry Pi 3 without the local network, but were unable to achieve this. We need the local network which Mr. Mattus will configure the PC's for.

Thursday, October 4, 2018

1 Meeting Minutes

GJ
We came into the lab today and the WiFi adapters were installed on the PC's. Today we plan on learning and configuring the Raspberry PI 3 to be able to connect with the PC through a wireless network. Also, for later reference the IP address of the Raspberry PI we are using is 169.254.0.2.

2 Raspberry Pi Testing

KV
We started our interface to the Raspberry Pi by using a serial connection. First, we edited a file on the raspberry file located in /etc/wpa_supplicant/wpa_supplicant.conf using the nano editor. We had to uncomment the network details for ECE-Robots-1. The previous group was using a network with an ssid name of 'QuanserAero' which we had to comment. After that, we were able to verify that the Raspberry Pi was connected to ECE-Robots-1 network as the wireless network card was given an IP address. We then connected to the Raspberry Pi using SSH and Putty to verify both PC #2 and the Raspberry Pi we connected to the network. We tested a sample program using MATLAB to turn on one of the user LEDs. In order to do this, we created a raspi object that takes the IP address, the user name, and password. Our IP address for the Raspberry Pi was 192.168.1.20. It is still unknown if this IP is static or if we will have to determine it each time the Raspberry Pi connects to the network. We set the variable rpi equal to this object. We were then able set a variable equal to the user LEDs: rpi.AvailableLEDs{1}. We could now turn on and off the LEDs: writeLED(rpi, led, 1); and writeLED(rpi, led, 0);. We also ran a simple command to display current files in a directory using: system(rpi, 'ls -al /home/pi'). To test the Raspberry Pi with simulink, we ran a simple test program to turn on and off the LED indefinitely. To kill the process, we use the command ps -A to find the name of the process. The run the command kill -9 'THE PROCESS NUMBER' on the Raspberry Pi.

3 QFLEX 2 Embedded

KV
Table 1 shows information about the connections on the QFlex 2 Embedded Panel for the Quanser Aero. We will use 6 of the 7 pins to connect to the Raspberry Pi.

Wire	Color	Function
1	White	VCC (1.8V-5V)
2	Yellow	MOSI
3	Blue	MISO
4	Green	CLK
5	Gray	QCLK (Not Used)
6	Purple	CS (Digital Output Line)
7	Red	GND

Table 1: *Embedded Wiring*

Figure 1 shows the Pin Layout for the Raspberry Pi. Pin 1 (white wire) on the embedded panel connects to +5V on the Pi. Pin 2 (yellow wire) connects to GPIO 10 on the Pi. Pin 3 (blue wire) connects to GPIO 9 on the Pi. Pin 4 (green wire) connects to GPIO 11 on the Pi. Pin 6 (purple wire) connects to GPIO 8 on the Pi. Pin 7 (Red wire) connects to ground on the Pi.

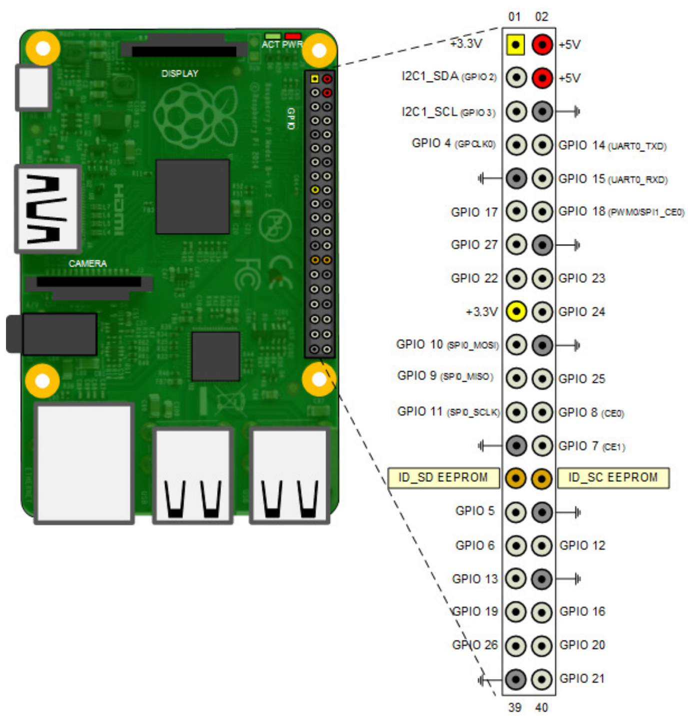

Figure 1: *Raspberry Pi 3 Pin Layout*

Thursday, October 5, 2018

1 Meeting Minutes

KV
Today we discussed with Dr. Miah how we: 1) created an SSH connection between the Raspberry Pi and Desktop 2, 2) tested a MATLAB program on the Pi, and 3)tested a Simulink program on the Pi. Next week, we plan to use SPI communication to to allow the Raspberry Pi to interface with the Quanser Aero. Last year's Implementation files have been made available to us to use as a reference. We also discussed the theory behind the LQG algorithm in greater detail.

Thursday, October 11, 2018

1 Meeting Minutes

GJ
The task for today is to retrieve images of the IP information for both of the PC's and the Raspberry PI. Read through last years SPI documentation and code. Add dots to the Problem Statement Diagram and create a more detailed diagram.

2 IP Images

GJ
Figure 1 is the screenshot of the IP information for PC #1, the wireless network is not configured for it yet. Figure 2 is the screenshot of the IP information for PC #2. The wireless network is configured and is the PC we have been working with so far. Figure 3 is the screenshot of the IP information for the Raspberry Pi 3 that we are using. Interestingly we are able to use the IP addresses for both the eth0 and wlan0 network cards to connect to the Raspberry Pi wirelessly through Putty.

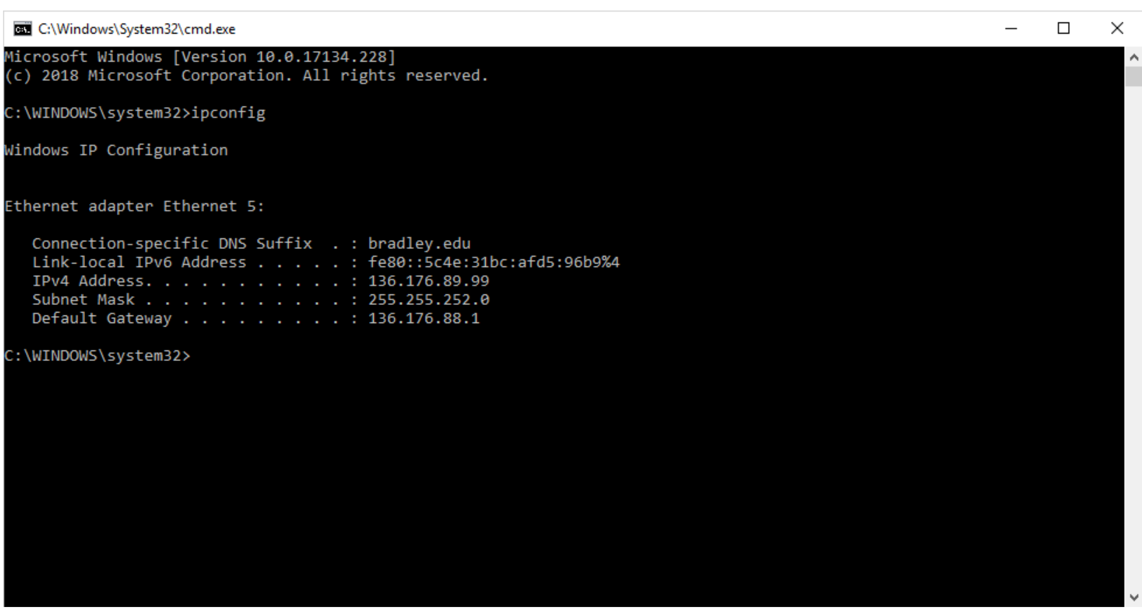

Figure 1: *Command Window for IP Information on PC1*

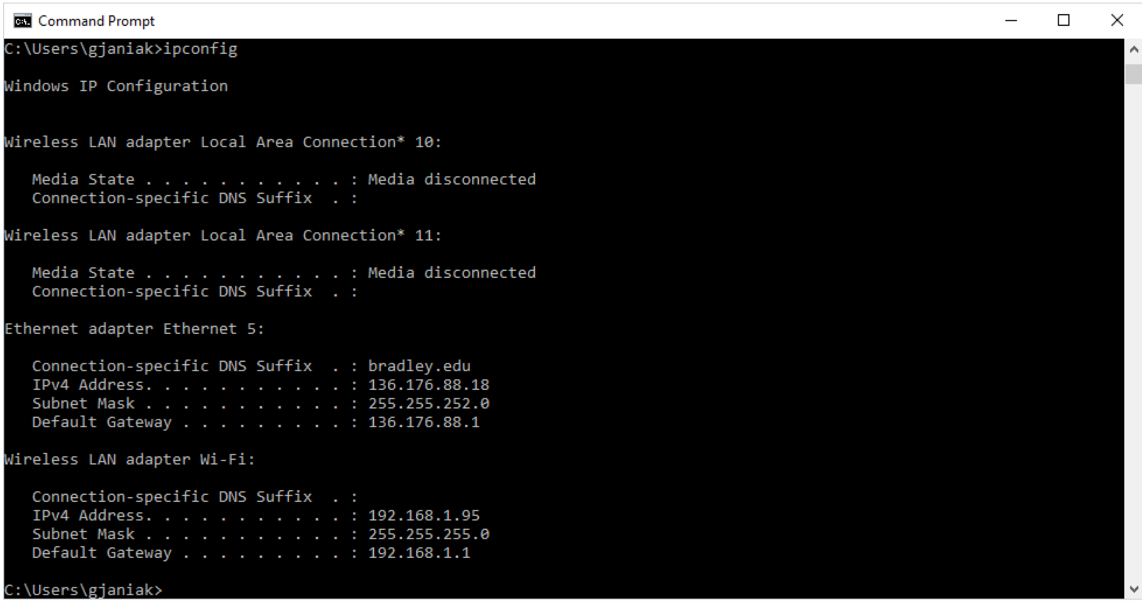

Figure 2: *Command Window for IP Information on PC2*

pi@raspberrypi-Tm3yvLpjge: ~

```
login as: pi
pi@169.254.0.2's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
Last login: Fri Apr 14 20:22:55 2017 from 192.168.1.95
pi@raspberrypi-Tm3yvLpjge:~ $ ifconfig -a
eth0      Link encap:Ethernet  HWaddr b8:27:eb:e8:c7:93
          inet addr:169.254.0.2  Bcast:169.254.255.255  Mask:255.255.0.0
          inet6 addr: fe80::d927:b380:fb64:9a76/64 Scope:Link
          UP BROADCAST MULTICAST  MTU:1500  Metric:1
          RX packets:0 errors:0 dropped:0 overruns:0 frame:0
          TX packets:0 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:0 (0.0 B)  TX bytes:0 (0.0 B)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:150 errors:0 dropped:0 overruns:0 frame:0
          TX packets:150 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:15514 (15.1 KiB)  TX bytes:15514 (15.1 KiB)

wlan0     Link encap:Ethernet  HWaddr b8:27:eb:bd:92:c6
          inet addr:192.168.1.20  Bcast:192.168.1.255  Mask:255.255.255.0
          inet6 addr: fe80::cd23:b6c6:9138:983c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:583 errors:0 dropped:497 overruns:0 frame:0
          TX packets:98 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:93424 (91.2 KiB)  TX bytes:16838 (16.4 KiB)

pi@raspberrypi-Tm3yvLpjge:~ $
```

Figure 3: *IP Information for Raspberry PI 2*

3 SPI Communication

KV

Pins 8, 10, and 11 and set to outputs on the Pi using MATLAB. Pin 9 is set as an input. We ran the following initialization code to set the pins to the correct mode for SPI communication. We also st the SPI period. Last years group documentation noted that we have to disable SPI in order to set the pin configuration

```
clear
clc
close all

rpi = raspi('192.168.1.20','pi','raspberry');

disableSPI(rpi);

% Configure pins on Raspberry Pi for the SPI comunication
% MOSI Output
configurePin(rpi, 10,'DigitalOutput');
% MISO Input
configurePin(rpi, 9,'DigitalInput');
% SPI Clock Output
configurePin(rpi, 11,'DigitalOutput');
% SS Output
configurePin(rpi, 8,'DigitalOutput');
% SPI output clock period 2us (500kHz)
spiPeriod = 0.0001;

enableSPI(mypi);

clearvars —except rpi spiPeriod
```

To run a Simulink model that utilizes SPI on the Pi, use the deploy to hardware button. You can then kill the program by searching for it's process number. You can also rerun the program buy navigating to the directory it is located in using the terminal. Then executing it: sudo ./FILE_NAME.FILE_EXTENSION. To stop the program using this method, hit Control + C. We are having difficulties where the Aero still applies the last voltage it was instructed to apply even after the program is terminated. The 'Run' button in Simulink also does not work as we expected as we encounter an error about the digital pin being busy.

Friday, October 12, 2018

1 Meeting Minutes

KV
Today we gave Dr. Miah a summary of our weekly progress. We shared with him some of our issues and he suggested that we contact Quanser.

Tuesday, October 16, 2018

1 Meeting Minutes

GJ
Started today with the intent to fix our basic recreation of the Simulink model created by last years group. Ken worked on setting up the second Raspberry Pi 3 for use on Aero #1. Starting Implementing LQR via the Raspberry PI.

2 Basic Simulink

GJ
We were getting an error dealing with the parameters of the Simulink model. I rechecked all of the blocks, and I found that the Data Memory blocks from Andrew's Simulink had the initial values of the signal attributes set as an array with eight values. After changing this, we still had an error with the configuration parameters. So I opened the configuration parameters for both Andrew's and our Simulinks and checked every parameter setting one by one. Some of these configurations were different so I changed mine to match Andrew's. After making these changes, we were able to build our basic Simulink model to the Raspberry Pi 3 #2 and ran on the Quanser Aero #2. In Figure 1, the top Simulink Model of the Basic testing using the Raspberry Pi 3 is shown. In Figure 2, the Simulink Model for the SPI communication between the Raspberry Pi 3 and the Aero is shown.

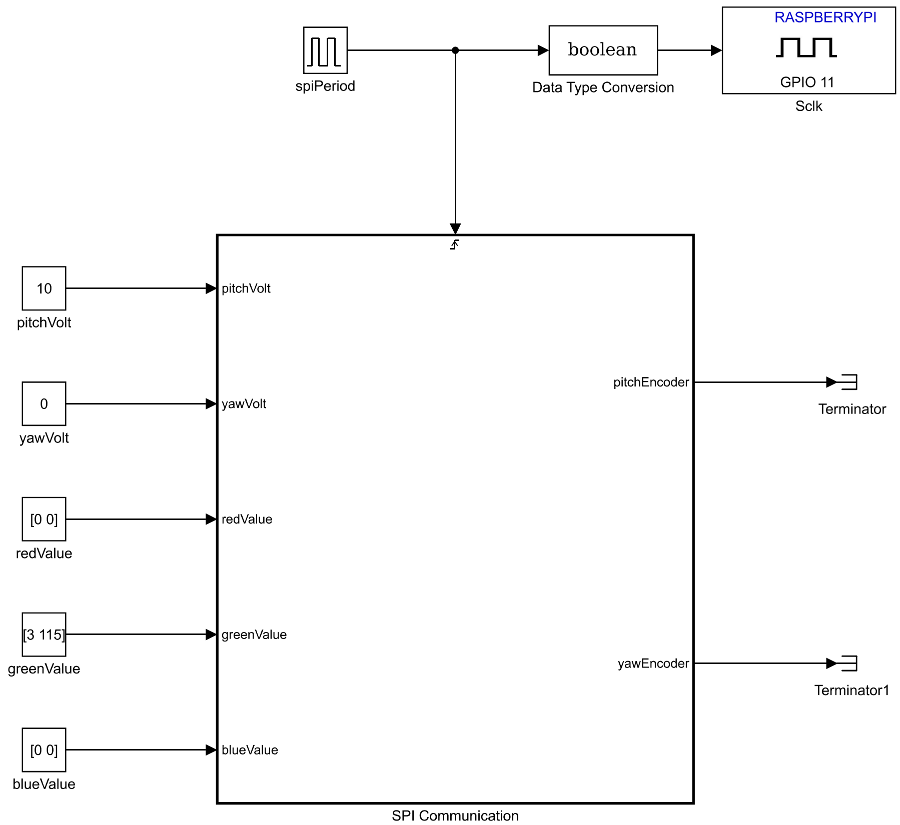

Figure 1: *Basic Top Simulink Model*

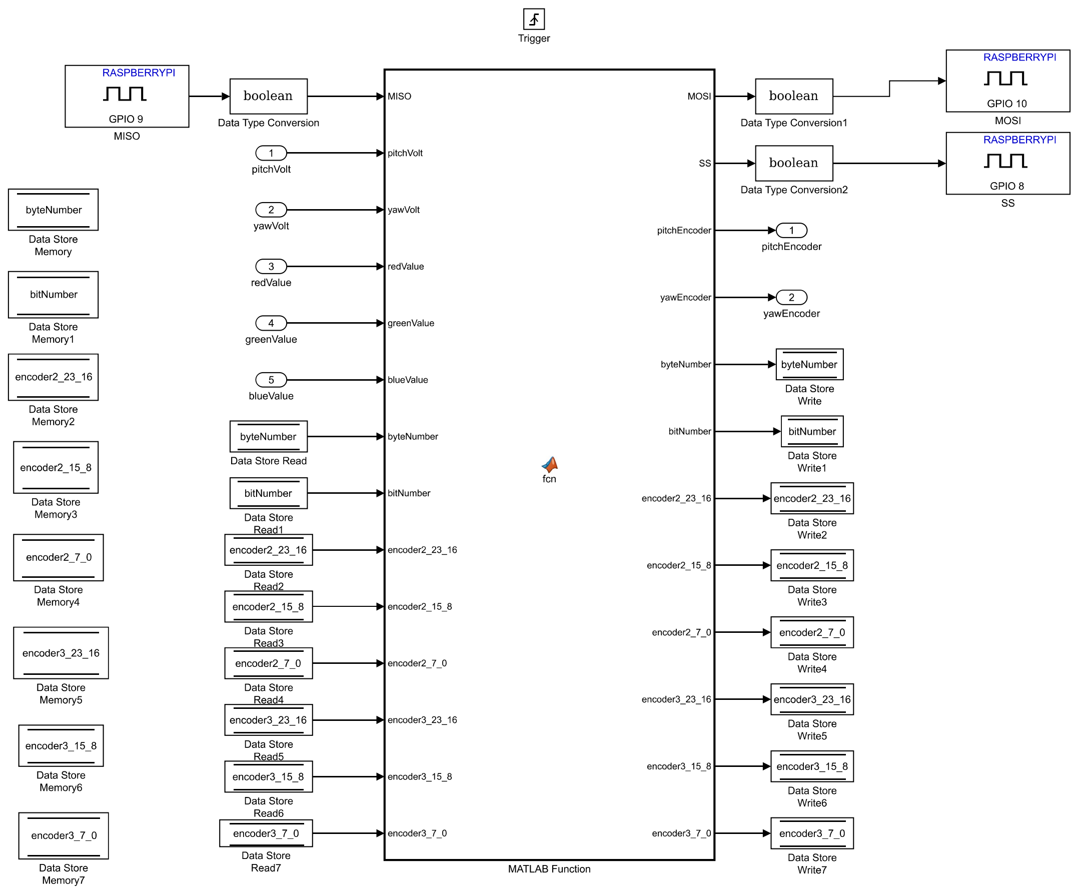

Figure 2: *SPI Communication Simulink Model*

3 LQR Via Raspberry Pi

GJ
We included the LQR control algorithm to our basic Raspberry Communication Simulink model. We then added a unit delay and a conversion constant to the pitch and yaw encoder outputs from the SPI communication block. It builds to the Pi just fine; however, with the testing we did, the Aero move to a certain position instead of moving a set degree variation from its starting position. More testing of this will be done on Thursday.

4 Setting up Second Raspberry Pi

KV
In order to work more efficiently, we will be setting up a second Raspberry Pi so that

both members of our team can work at the same time. Instead of installing the operating system from scratch, we have decided to copy the disk image from the SD card of the Raspberry Pi we have been working on onto the new one. This will give us immediate access to the files we are currently working on as well as save all of our settings. We tried using Win32 Disk Imager, however encountered CRC errors. We hope that this does not indicate that part of our file system is corrupt. We will be contacting Dr. Malinowski who teaches a Linux class for some help with this issue.

Thursday, October 18, 2018

1 Meeting Minutes

GJ
Today we will attempt to continue experimenting and configuring the LQR implementation.

2 LQR Via Raspberry Pi cont.

GJ
After some testing, we have deciphered that when the Pi is power ed up and connected to the Aero the initial position is taken. Then when our current Simulink is run the desired angle changes are applied to that position. The issue right now is changed the desired angle. We cannot reload the Simulink model with changed desired angles because of an error saying the digital I/O pins are busy. We are still trying to fix this error.

Friday, October 19, 2018

1 Cloning Linux Distribution

KV
We met with Dr. Malinowski to discuss the best method to clone the image of the file system. He showed us an easy way using a Linux virtual machine and the "gparted" program which is a partition editing program for Linux. Using this, we mounted the drives in to Virtual Box and then unmounted them so we could read and write to them. We then simply copied the file systems over from one SD card to the other. Since the new SD card was 32GB and the old was only 8GB, we increased the size of the Linux partition. There was also warnings about possibly corrupt files. Future students who work on this Pi should also be made aware and place extra emphasis on backing up files. We will also be more careful when shutting down the Pi, removing power, and removing the SD card.

Tuesday, October 23, 2018

1 Meeting Minutes

GJ
There were tours in the lab so we were moved to our second helicopter for "show and tell." We had to find the IP address of our Raspberry Pi 3 #1 which is '192.168.1.77'.
KV
Figure 1 shows the IP Information for Raspberry PI 1 which we had set up and tested Wednesday. We are not sure if it was assigned a static IP address of if DHCP was used as we do not have authorization to connect to the wireless router (default gateway: 192.168.1.1). Also now that we have wireless capability, we have recorded the PI information with regards to the wireless network interface card as seen in Figure 1.

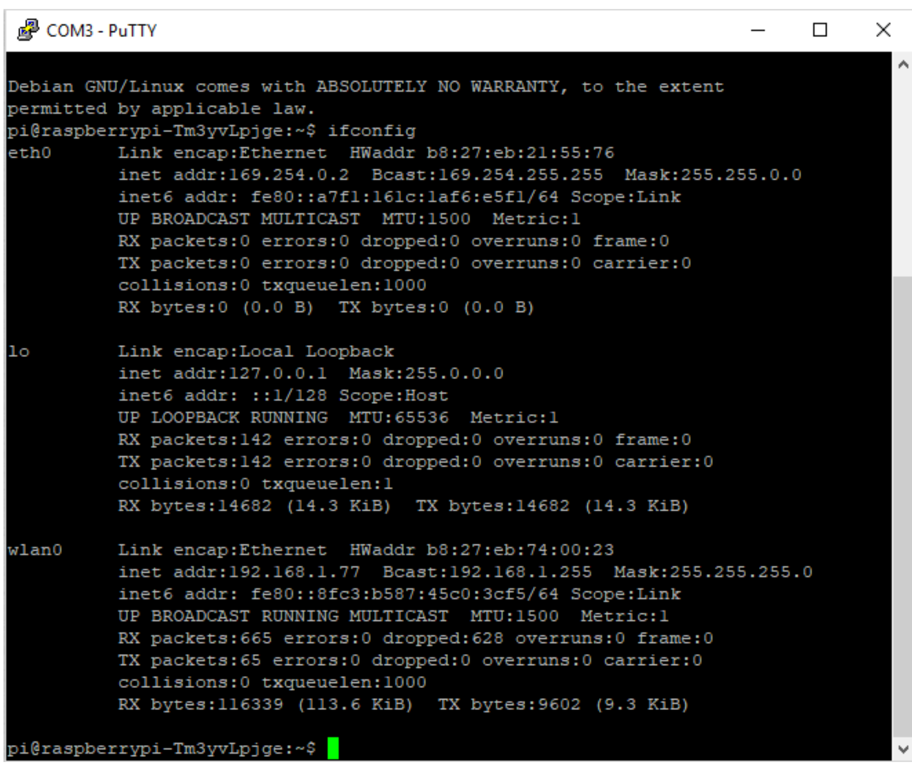

Figure 1: *IP Information for Raspberry PI 1*

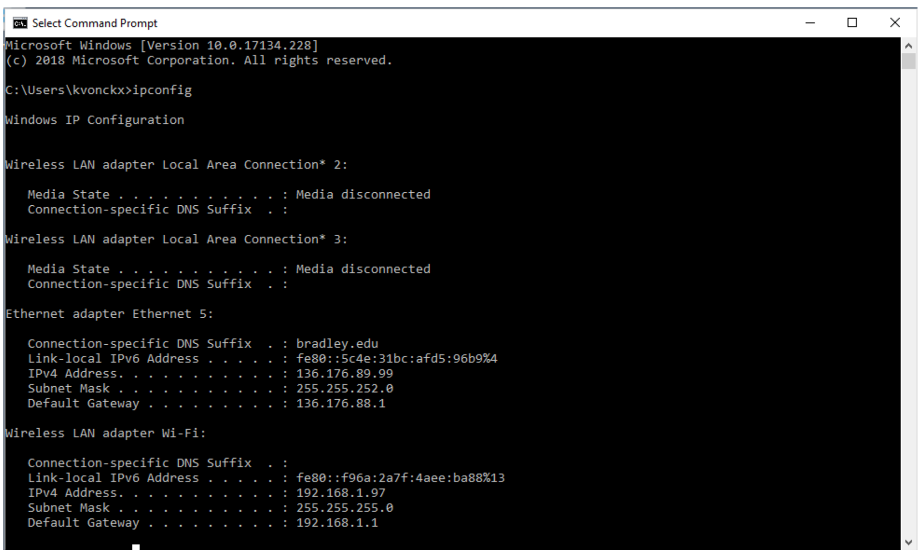

Figure 2: *IP Information for PC 1 with Wireless*

Thursday, October 25, 2018

1 Meeting Minutes

GJ
Today, I had the plan of working on the Simulink Android application, while Ken worked on finishing up the problem statement and started putting the website together for ECE497 credentials. However, there was another tour group, of undecided engineering majors, so we were coaxed into presenting our current worked again which is good practice for talking about our project in front of groups.

2 Android LQR

GJ
Started to incorporate a mobile device to be able to control the Aero. We plan on using Ken's phone which has an IP address of '192.168.1.70'. We created a Simulink for the Android application and modified the SPI_LQR Simulink to be able to communicate with Ken's phone. Figure 1 is the Simulink model of our SPI with the ability to communicate with an Android device. Figure 2 is the Simulink Model for the Android Application. We tried to deploy Figure 2 to Ken's phone but received an error. We believe this is from not having the Samsung Driver installed. We emailed Mr. Mattus about getting this driver installed.

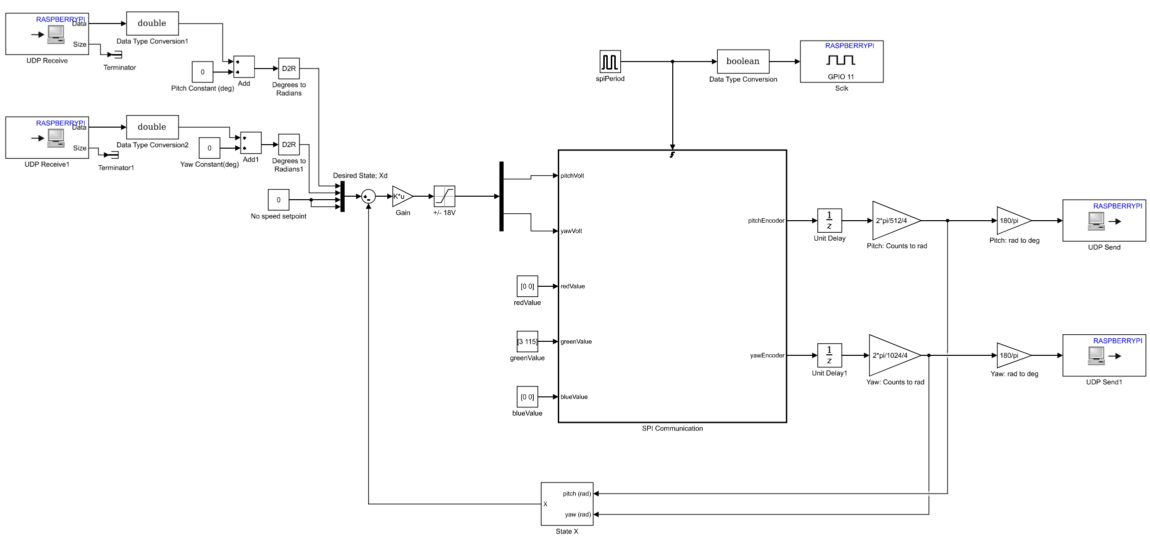

Figure 1: *Simulink Model of LQR with Android Compatibility*

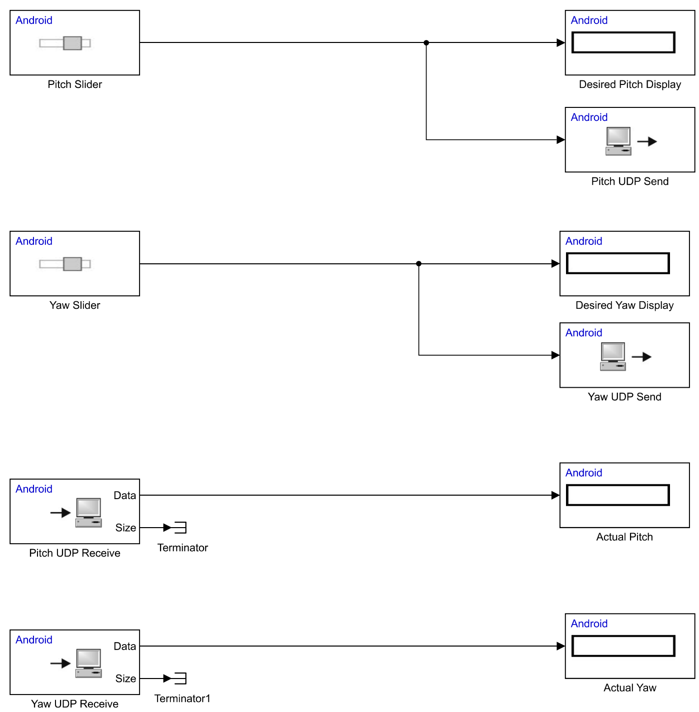

Figure 2: *Simulink Model of Phone App*

Friday, October 26, 2018

1 Meeting Minutes

GJ
For our weekly meeting we gave Dr. Miah an update of where we are with the project. He also gave some advise on the proposal and website.

Tuesday, October 30, 2018

1 Meeting Minutes

GJ
Today we plan on working on downloading the Android app to one of our phones.

2 Android App

GJ
We tried to build a test application to both Ken's and my phones; however, we received a build error dealing with MATLAB being unable to acquire online files for the building. We believe this is a firewall issue from some research we did. We have emailed Mr. Mattus to ask for his help on the issue. From Mr. Mattus's recommendation we tried to run the test app from MATLAB2018; however, we encountered a new issue of needing Android Studio. We have emailed Mr. Mattus so he may install this software.

Thursday, November 1, 2018

1 Meeting Minutes

GJ
Came into the lab with the intention to try downloading the test application with Android Studio installed; however, Mr. Mattus has not gotten to installing it yet. We sent another email to him to remind him then proceeded to work on our proposal draft so the time does not go to waste.

Friday, November 2, 2018

1 Meeting Minutes

GJ
We had our weekly meeting with Dr. Miah. He informed us about a conference paper he wishes to write with us by November 26th. He also shared with us a presentation template for the proposal presentation draft that is due November 13th. We tried getting a hold of Mr. Mattus to get past some administrative blocks that we keep encountering while trying to test an android test application.

Tuesday, November 6, 2018

1 Meeting Minutes

GJ
We discovered that Android Studio removed one of their files with their most recent update and MATLAB2018a requires that file when building android applications. We fixed this by installing an older version of Android Studio NDK bundle from December 2017 and coping the missing folder into the file path that MATLAB was calling. This fixed this issue and the test app was downloaded to an android phone. Figure 1 shows a screen shot of the test application on the android. After we completed this we decided to try and control both helicopters with one android device. After changing the UDP send IP addresses on the LQR Simulink model with the android compatibility we downloaded it to the Raspberry Pi to test the App on the Aero. It worked, we are able to control the Aero with LQR using an android device as a user interface. Figure 2 is the screen shot of the android interfacing app running.

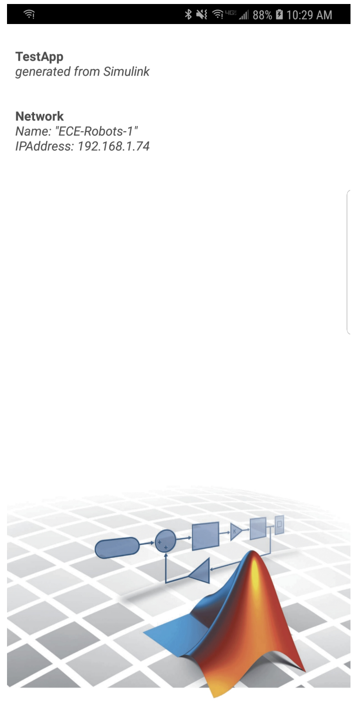

Figure 1: *Screen shot of Test App Running*

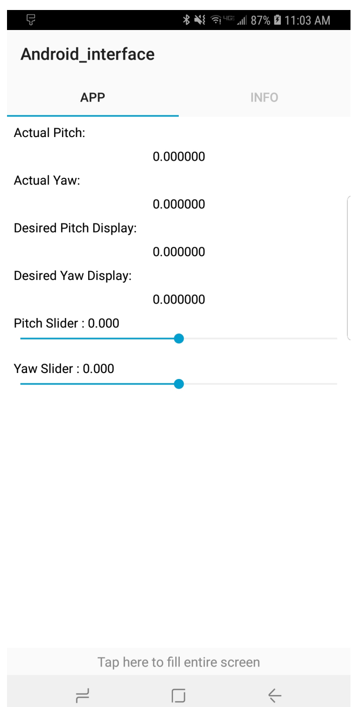

Figure 2: *Screen shot of Android Interface App Running*

Thursday, November 8, 2018

1 Meeting Minutes

GJ
For today we decided to reattempt downloading an app that can control both helicopters to an android device. Warning, it took 50 minutes to build the application to the phone.

2 Two Helicopters Using LQR

GJ
We were able to implement the LQR control algorithm on both helicopters at the same time. Figure 1 shows the application running that controls both of the helicopters.
KV
To accomplish this, we are using User Datagram Protocol (UDP) to send information from the phone to the Raspberry Pi's. Each variable that is being sent and received is being send along a different port ranging from 26000 to 26007 as shown in Table 1. The IP addresses of each device is hard-coded in to the program. The IP address for Glenn's phone is 192.168.1.74.

Port #	Sending Device IP	Receiving Device IP	Function
26000	192.168.1.74	192.168.1.20	Aero 2 Desired Pitch
26001	192.168.1.74	192.168.1.20	Aero 2 Desired Yaw
26002	192.168.1.20	192.168.1.74	Aero 2 Actual Pitch
26003	192.168.1.20	192.168.1.74	Aero 2 Actual Yaw
26004	192.168.1.77	192.168.1.74	Aero 1 Actual Pitch
26005	192.168.1.77	192.168.1.74	Aero 1 Actual Yaw
26006	192.168.1.74	192.168.1.77	Aero 1 Desired Pitch
26007	192.168.1.74	192.168.1.77	Aero 1 Desired Yaw

Table 1: *Port Number Configuration*

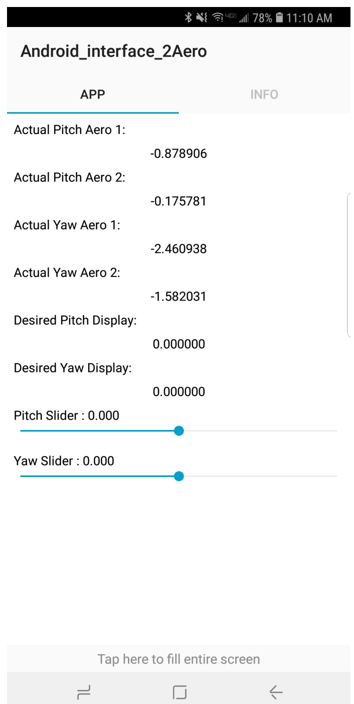

Figure 1: *Screen shot of Android Interface App Running*

Friday, November 9, 2018

1 Meeting Minutes

GJ
For this meeting with Dr. Miah we first talked about the proposal presentation. We also signed up to present it on November 29 at 9:25 AM to 9:50 AM. We also talked about the conference paper that Dr. Miah would like to be completed by November 13th. We also talked a little bit about implementing and simulating LQG.

Tuesday, November 13, 2018

1 Meeting Minutes

We came into the lab with the intent of reloading the two Aero App onto the android since the IP address of one of our PI's changed. We also want to experiment and test saving data to MATLAB over wireless connection.

2 Storing Data Wirelessly

KV
We are getting the following error message when tring to save position data on the raspberry pi from our experimentation.

```
getFile(rpi,'/home/pi/JaVoAero2018/Android\_LQR\_Aero1\_DC.mat')

*** Using a default buffer of size 1024 for logging \\variable tout

*** Log variable rt\_tout has wrapped 682 times
    using a circular buffer of size 1024
*** To avoid wrapping, explicitly specify a
    buffer size of 698841 in your Simulink model
    by adding OPTS="-DDEFAULT\_BUFFER\_SIZE=698841"
    as an argument to the ConfigSet MakeCommand
    parameter
** created Android\_LQR\_Aero1\_DC.mat **
```

Thursday, November 15, 2018

1 Meeting Minutes

GJ
Came into the lab with the intent of completing storing the position data to be plotted on MATLAB. We also made a video of both helicopters running with a screen recording of the app running to be spliced side by side for easy presentation video. We were able to collect the data of one signal; however, we are still having trouble collecting all of the data at once.

Friday, November 16, 2018

1 Meeting Minutes

GJ
Talked about finishing up the conference paper.

2 Data Collection

GJ
We completed figuring out how to collect data and plot it into MATLAB. We used a scope block with sampling time set to 0.01 seconds, limit data points box checked, and log data to workspace checked and storing it as a structure with time.

Thursday, November 29, 2018

1 Proposal Presentation

KV
Today we presented our project to a number of the departments faculty members. We we answered questions such as why we decided to use an embedded system and our control platform.

Friday, November 30, 2018

1 Meeting Minutes

GJ
For our meeting today we went over our proposal presentation feedback that we presented on Thursday, November 29. We were also assigned to read documents on ADP that are on the Google Drive.

Tuesday, December 4, 2018

1 Meeting Minutes

Wednesday, January 23, 2019

1 Meeting Minutes

KV
Weekly meetings with Dr. Miah will now be held on Mondays between 2 & 3 pm. We will be working in the lab from 1030 am to 1 pm on Tuesdays and Thursdays. Over break, we found that our first conference paper was rejected. We will now be submitting it to ISIE with a deadline of February 15th. Our second conference paper will be submitted to IEEE IROS and with a deadline of February 18th. Dr. Miah has also done some work as far as simulating LQG in MATLAB and Simulink.

Thursday, January 24, 2019

1 Lab Work

KV
Today we will ensure we are still able to connect to our raspberry pi's and run one of previous programs. We attempted to run one of Dr. Miah's simulations using LQG, however encountered a matrix dimension mismatch error.

Monday, January 28, 2019

1 Meeting Minutes

KV
Dr. Miah assisted us in debugging the code we were working with last week. The matix for Bw was of the wrong dimensions and show be nx1. We have also set a deadline to have our LQR paper done by Febuary 8th.

Tuesday, January 29, 2019

1 Lab Work

KV
Today, we are working on developing MATLAB code that sets up of state space model for LQG. We have also started to create the block diagram that uses the values calculated in the MATLAB code. However, we are encountering an error where our u is outputting a 2x2 matrix and the output of $\mathbf{K} \times \hat{\mathbf{x}}$ is a 2x1. In order to view some of Dr. Miah's simulink files, we require MATLAB 2018b. We have emailed Mr. Mattus and asked him to install that version for us.

Thursday, January 31, 2019

1 Lab Work

KV
Mr. Mattus has installed MATLAB 2018b on our lab computers. Now we can view Dr. Miah's block diagrams to get a better understanding of LQG.
First, we solved our matrix dimension problem by removing the initial conditions for the derivatives of pitch and yaw. Now we are encountering a problem where the kalman filter is returning an error that y is in discrete-time and u is in continuous time. We the replaced the integrator block with a discrete-time integrator. As a result, we had to add two delays in the feedback for the kalman filter. Ken learned in controls 2 that Euler's forward integration can cause some problems with stability, so we will set the integrator to use the backward rule. We tried to implement this on the Aero, and even though it successfully built, the Aero did not move.
We then tried to convert Dr. Miah's model to discrete-time in the same way we did to our own model. However, the voltage is oscillating between the positive and negative limits

Monday, February 4, 2019

1 Meeting Minutes

KV
Dr. Miah gave us the following tips on how to fix out discrete-time problem with LQG:

1. Follow LQR and add a low-pass filter
2. Start with Quanser's LQG

The following are some notes we took during our discussion on the changes that need to be made to paper 1:

- Add problem statement
- Include prices (single board computers are $35)
- Dr. Miah will make corrections to Alg. 1
- Explain teleoperation (ex. why is a remote control car not a teleoperation system)
- Explain how user knows initial configuration: "For the moment, assume the initial configuration is known to the user and could be measured using a digital compass, however this is out of the scope of this paper."
- Explain the impact of communication delay on the system: "Given that sampling time used, the communication delay has a negligible impact on the system."
- Further discuss modularity: "...applied in multiple homogeneous helicopters and single board computers because it will be able to control the fleet regardless of their internal architecture."
- Edit images to make the font bigger
- Find new image without labels on raspberry pi
- Use the following command to print a higher DPI block diagram: print -s (modelName) -dpdf -R600 modelName.pdf
- Smart logic is out of the scope of this paper

2 Conference Paper 2

KV
Ken has started writing conference paper 2 and has outlined it as follows:

1. Introduction
2. Background
    a) Literature Review
    b) LQR
    c) LQG
3. Modeling a 2-DOF Helicopter
4. Experimental Results
5. Conclusion and Summary

Tuesday, February 4, 2019

1 Lab Work

KV
Today we worked on making changes to some of the suggestions given to us by FLAIRS in paper 1. We have made it so that most all over the images are easier to read. Ken will make a note about why our system is tolerant to lost packets and continue to work on the other paper out of lab. Glenn will finish the corrections to this paper

Thursday, February 6, 2019

1 Lab Work

KV
Today, we have tried to recreate our LQR for the raspberry pi, however it is exhibiting similar behavior to when we attempted LQG.

Monday, February 11, 2019

1 Meeting Minutes

KV
Since we are still having trouble with our LQG, Dr. Miah suggested that we compare the voltages from his simulation to what we are getting in our implementation. He also suggested that we start to recreate LQR from the USB level. Glenn has be assigned to read conference paper 1 line by line to ensure it is ready for submission.

Tuesday, February 12, 2019

1 Lab Work

KV
We have recreated our LQR model in Simulink using the PC and a USB connection using an additional PI controller to reduce steady-state error. The actual position converges on the reference position. We are still having a problem when we run a similar code on the Raspberry Pi. We have gone over every block in the model and have a strong understanding of how it operates, however all we can do is speculate on where the problem is coming from. I have tried to run the data collection program, however, it will not run and I am unsure where the problem lies.

Tuesday, February 12, 2019

1 Lab Work

KV

We have recreated our LQR model in Simulink using the PC and a USB connection using an additional PI controller to reduce steady-state error. The actual position converges on the reference position. We are still having a problem when we run a similar code on the Raspberry Pi. We have gone over every block in the model and have a strong understanding of how it operates, however all we can do is speculate on where the problem is coming from. I have tried to run the data collection program, however, it will not run and I am unsure where the problem lies.

Thursday, February 14, 2019

1 Lab Work

KV
Today we implemented LQG using a USB connection. When running the model, we noticed significant error. After running the simulation for a longer period of time and examining the voltages, we realised that the error was being reduced, however, the rise time was drastically increased. This is most likely the effect of adding an additional PI controller. To fix this we are attempting play with the Q and R matrices that will adjust some of the performance specifications. By increasing R, the voltage will decrease and the time to converge will decrease.

Friday, February 15, 2019

1 Lab Work

KV
I started to experiment with changing Ki, Qe, and Re. I believe that more Ki in the system will help the controller to follow the reference signal better. Since LQR preforms better than LQG, I believe that the Qe and Re matrices need tweaking as well. I plan to bring this up to Dr. Miah during our meeting on Monday. I have also included the plots in this document for system that includes a PI controller.
The following is the results with a PI controller introduced into LQR using a USB connection to the desktop computer:
The following my latest attempt with LQG.
Q_{hat} = diag([.1 .1 1 1 10 .1]);
R_{hat} = 0.01*eye(m,m);
K = K_{hat}(1:m,1:n);
Ki = -2.5*K_{hat}(1:m,n+1:n+p);
Qe =diag(ones(1,n))*10;
Re =.01*diag(ones(size(C,1)));

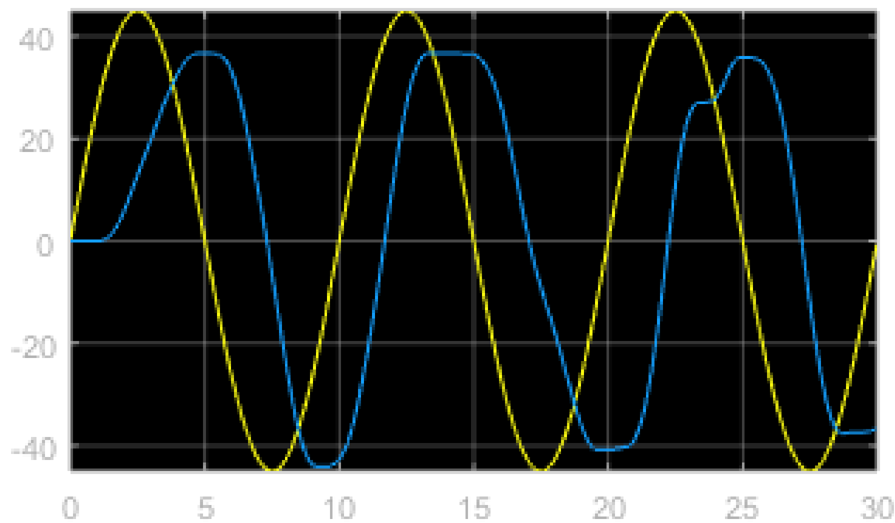

Figure 1: *LQR Pitch with PI controller*

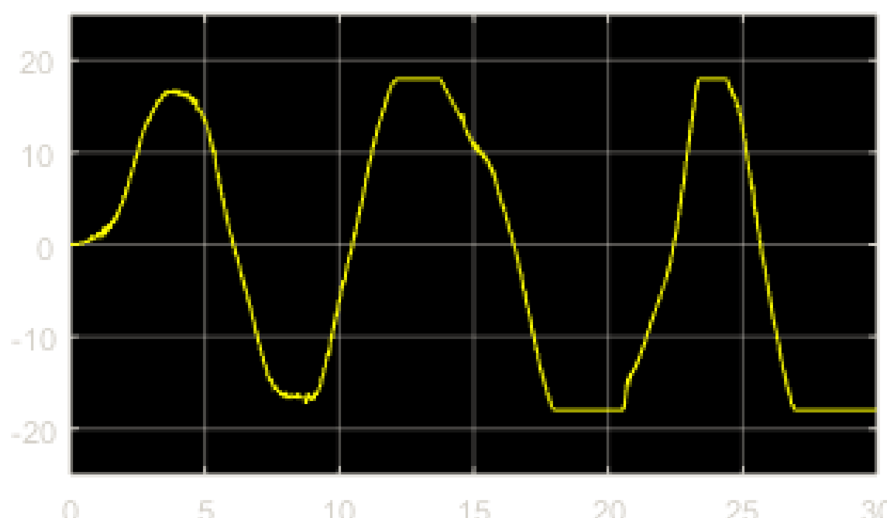

Figure 2: *LQR Pitch Voltage with PI controller*

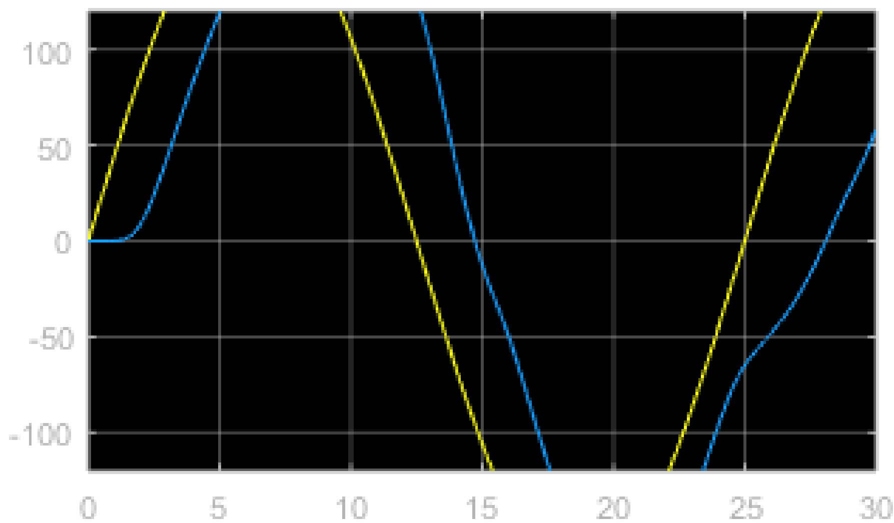

Figure 3: *LQR Yaw with PI controller*

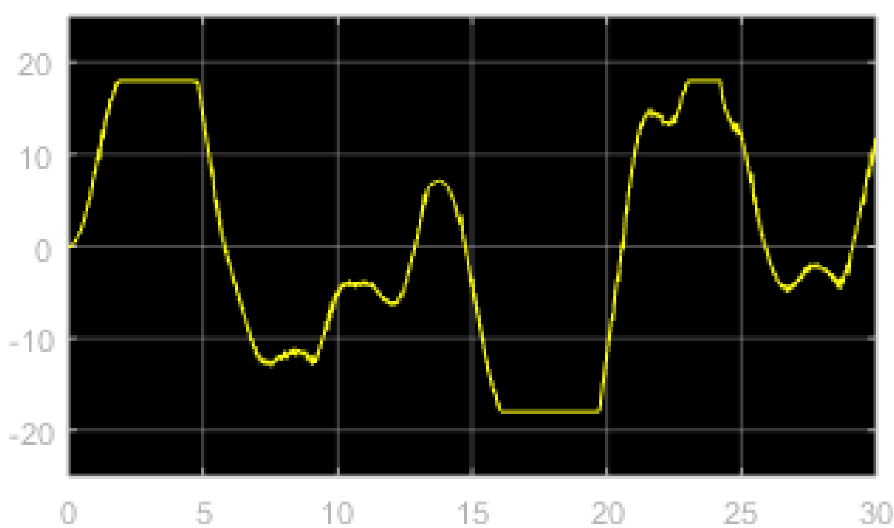

Figure 4: *LQR Yaw Voltage with PI controller*

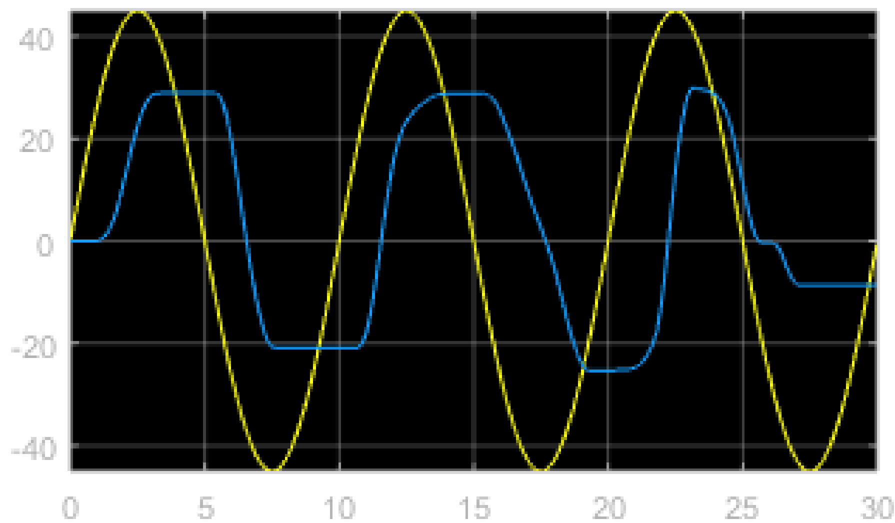

Figure 5: *LQG Pitch with PI controller*

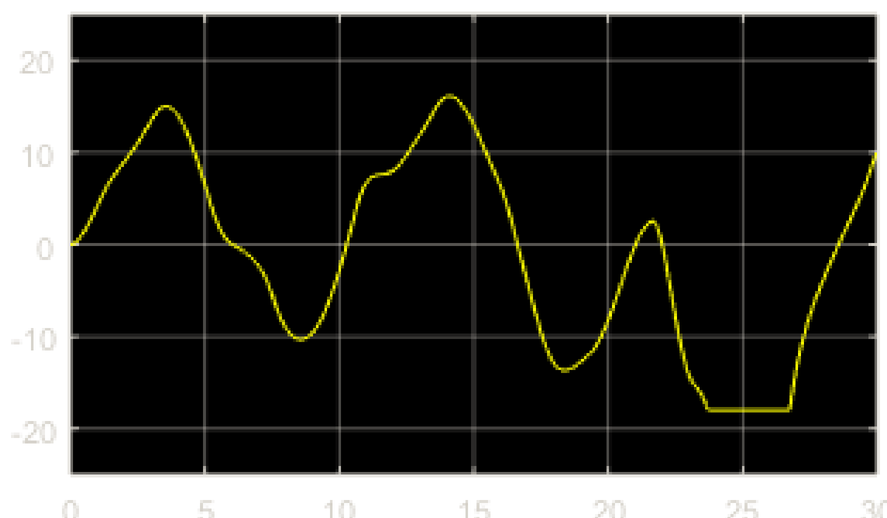

Figure 6: *LQG Pitch Voltage with PI controller*

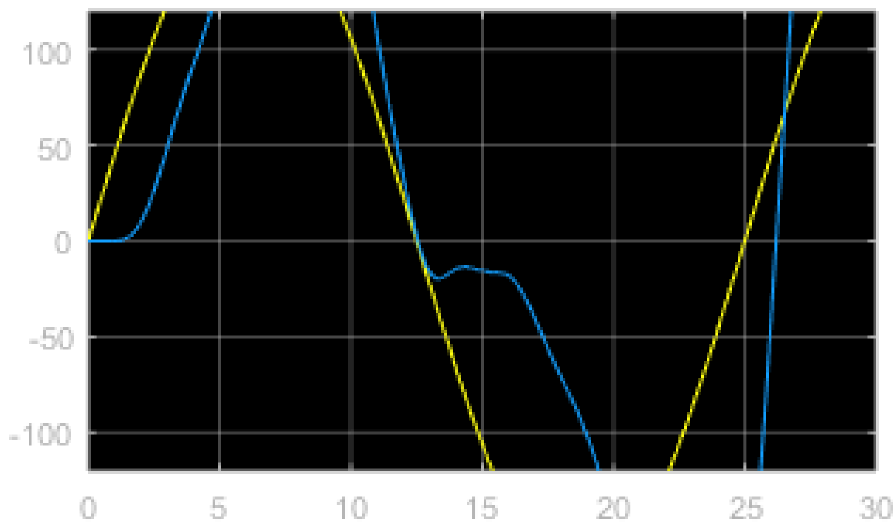

Figure 7: *LQG Yaw with PI controller*

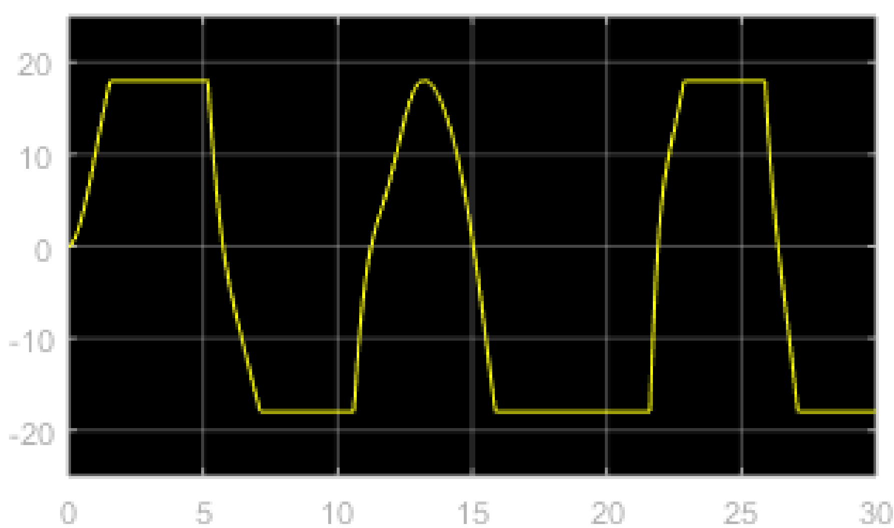

Figure 8: *LQG Yaw Voltage with PI controller*

Monday, February 18, 2019

1 Meeting Minutes

KV
In our meeting, we discussed some of the results from implementing the LQR/LQG with a PI controller. Dr, Miah gave us a few suggestions.

- set Qe and Re to $10^{-}3$
- set Q = [1 1 1 1 10 10]
- do not multiply Ki by a constant
- we can try to increase K
- create new gain matrix to multiply yhat by

One of the main differences we noticed between Quanser's model and our own was the placement of the gain matrix. Quanser has it placed before u, the PI model has it as part of the state feedback.

Tuesday, February 19, 2019

1 Lab Work

KV
After viewing some of Quanser's LQR results, we noticed that their model has better performance over the model with the PI controller.

We will also be meeting with Dr. Miah today at 5pm.

Questions for Miah:
Why is Ki not a diagonal matrix?
Why is Ki negative?
Looking at Quanser LQG, why are they using theta dot and theta dot dot? Should the state space model in the kalman filter be different from the ones used in LQR.

We tried increasing the gain K, however this made the performance worse. We also noticed a correlation between Rhat and rise-time (phase). As Rhat decreases, the rise-time improves. We tried creating a new gain block called Kout which multiplies a value to the estimated output to make the system think that the system is farther away (gain less than 1) or closer, (gain greater than 1). To do this, we made it a diagonal matrix so that the gain does not effect the other states.
The following is LQR:
Q_{hat} = diag([1 1 1 1 10 10]);
R_{hat} = 0.001*eye(m,m);
Kout=([.5, 1])

The following is for LQG:
Q_{hat} = diag([1 1 1 1 10 10]);
R_{hat} = 0.001*eye(m,m);
Qe =0.001*diag(ones(1,n));
Re =0.001*diag(ones(size(C,1)));
Kout=([.2, 1])

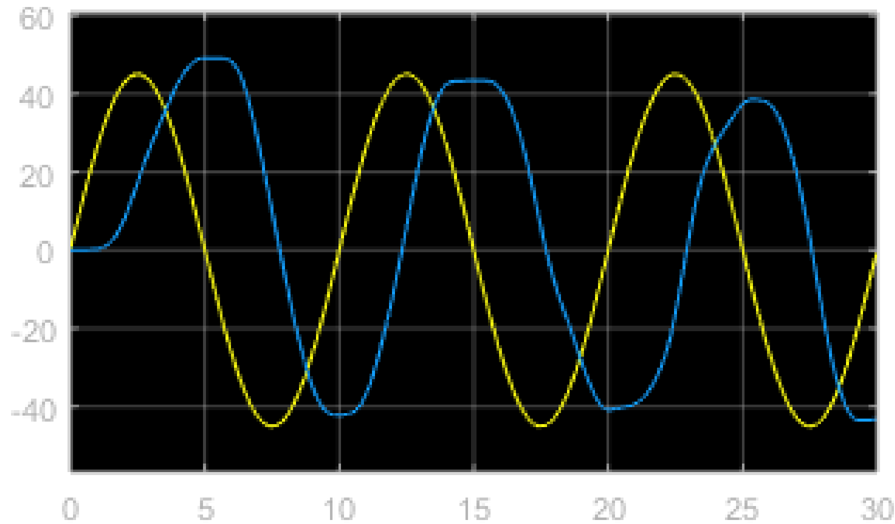

Figure 1: *LQR Pitch with PI controller mk2*

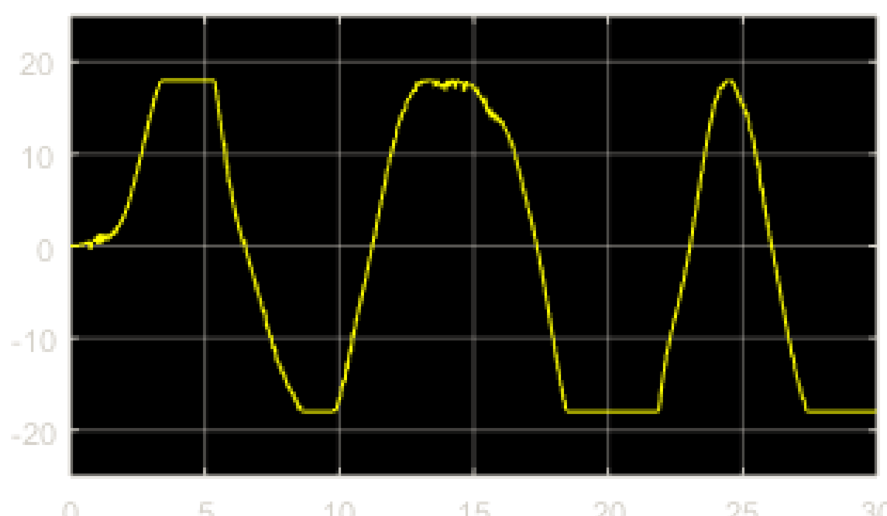

Figure 2: *LQR Pitch Voltage with PI controller mk2*

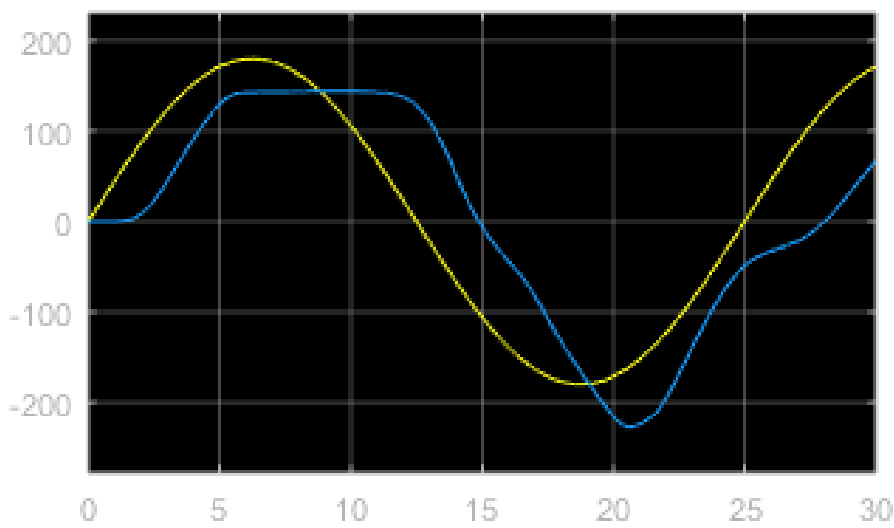

Figure 3: *LQR Yaw with PI controller mk2*

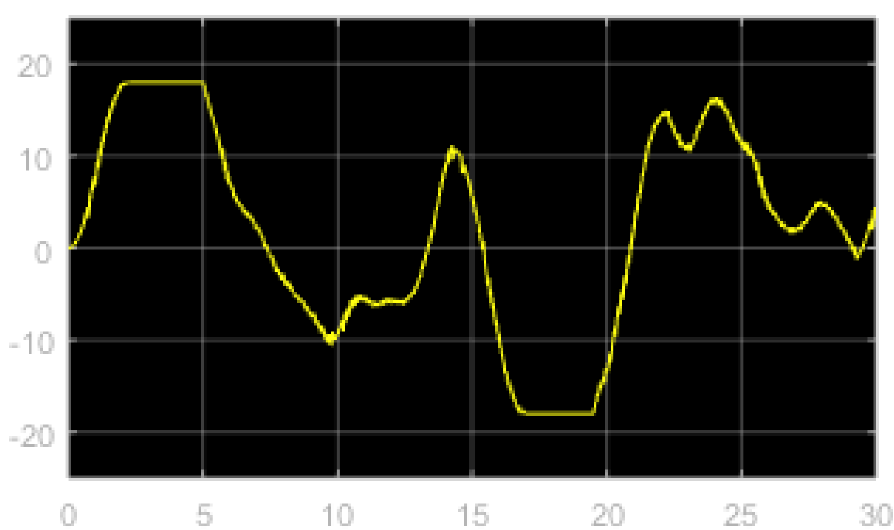

Figure 4: *LQR Yaw Voltage with PI controller mk2*

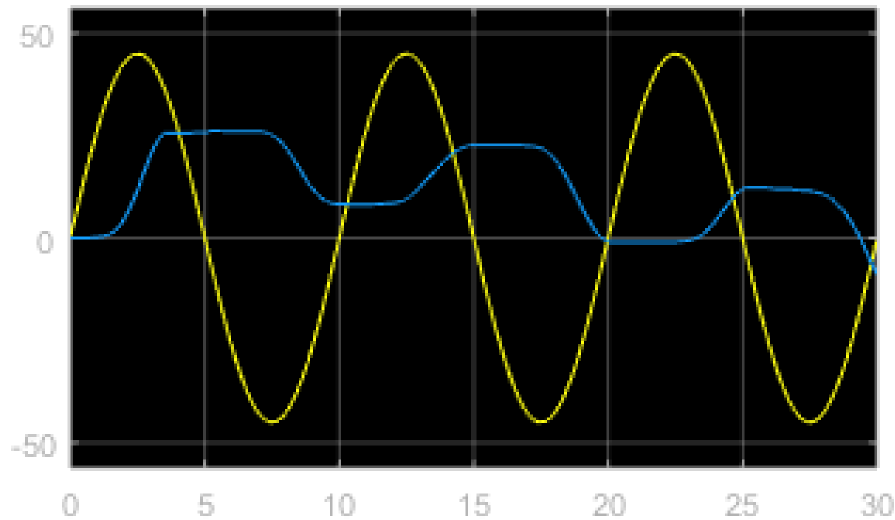

Figure 5: *LQG Pitch with PI controller mk2*

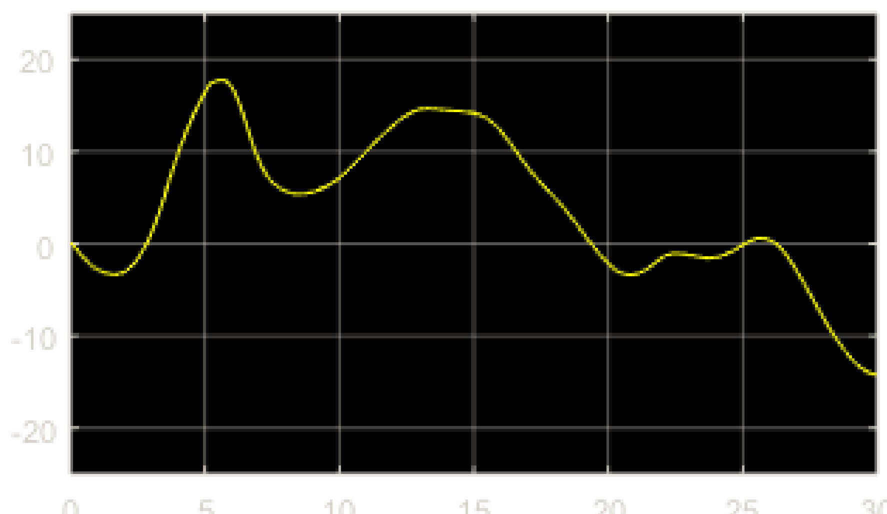

Figure 6: *LQG Pitch Voltage with PI controller mk2*

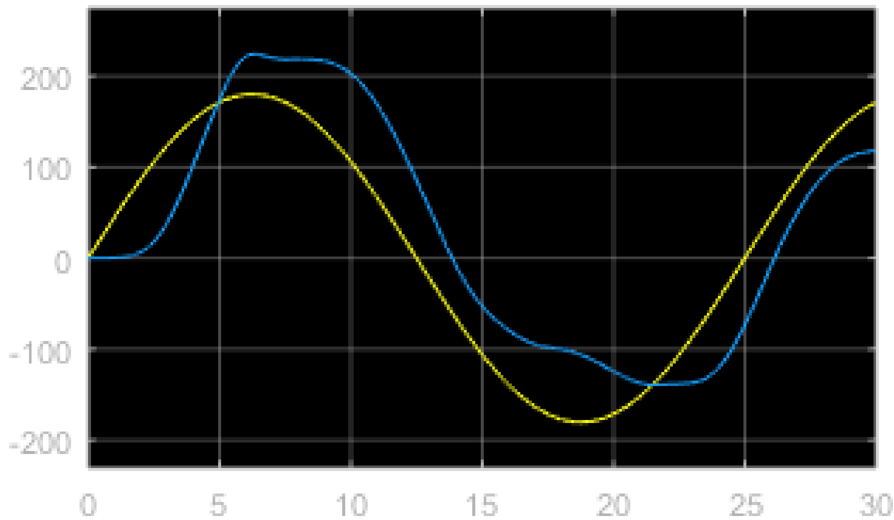

Figure 7: *LQG Yaw with PI controller mk2*

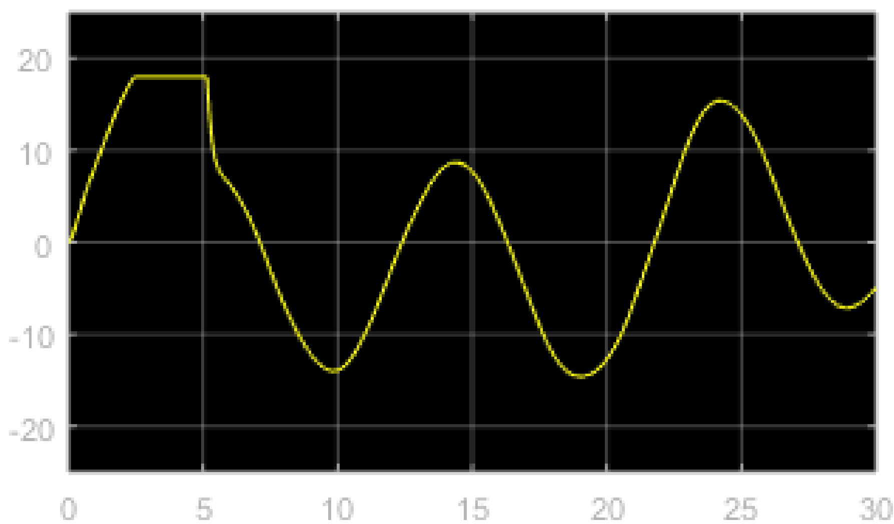

Figure 8: *LQG Yaw Voltage with PI controller mk2*

2 Meeting Minutes

KV
Dr. Miah gave us two new suggestions for our block diagram. The first one resembles the LQR algorithm developed by Quanser. The second one changes the equation that determines the input to the plant. Originally it was:

$$u(t) = -Kx(t) + K_I v(t) \tag{1}$$

This is changed to:

$$u(t) = -K(x_{\text{ref}}(t) - x(t)) + K_I v(t) \tag{2}$$

because the error of the output was being feedback into the system but not the error of the states.

Thursday, February 21, 2019

1 Lab Work

KV
Today, we will being testing the two new block diagrams discussed on Tuesday. The first one entail removing the feedback for the output and to only calculate the error between the desire and actual states. This is shown in Figure 1.
The Q matrix was set to diag([1 1 1 1 10 10]) and the R matrix to 0.0001*eye(2).
The results were much imporved over the PI controller version and very comparable to our LQR from last semester.

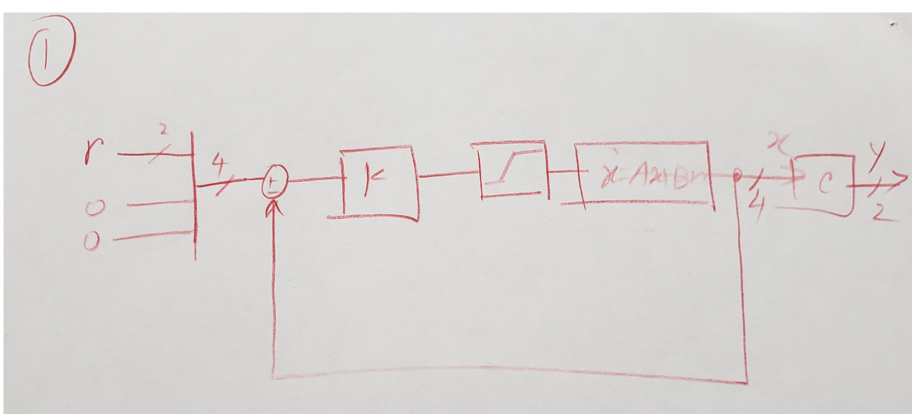

Figure 1: *LQR Block Diagram Version 2*

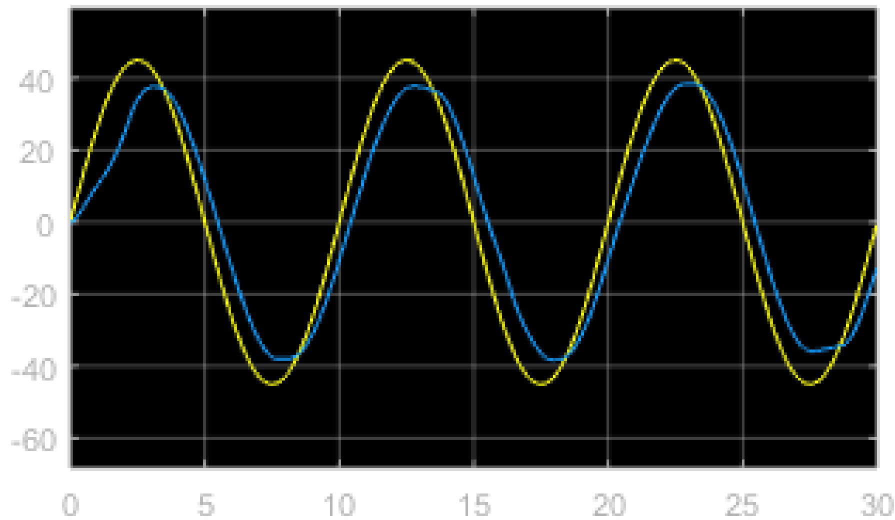

Figure 2: *LQR Pitch Version 2*

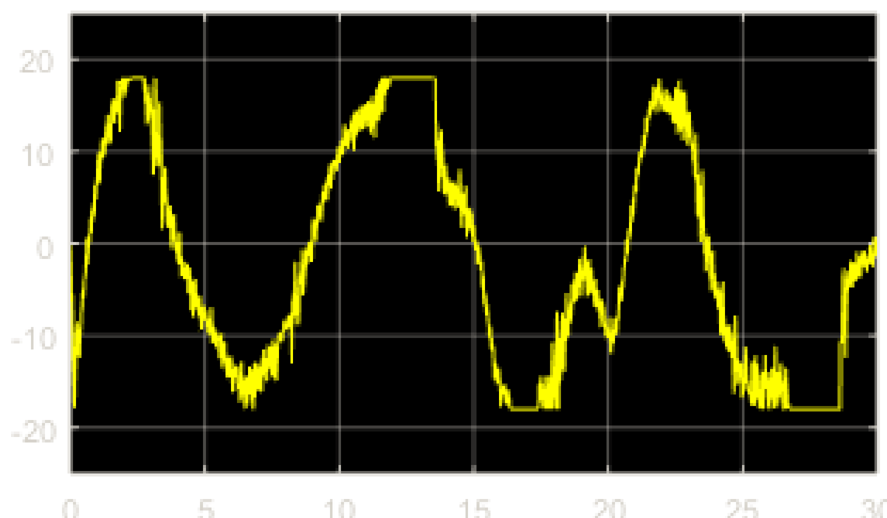

Figure 3: *LQR Pitch Voltage Version 2*

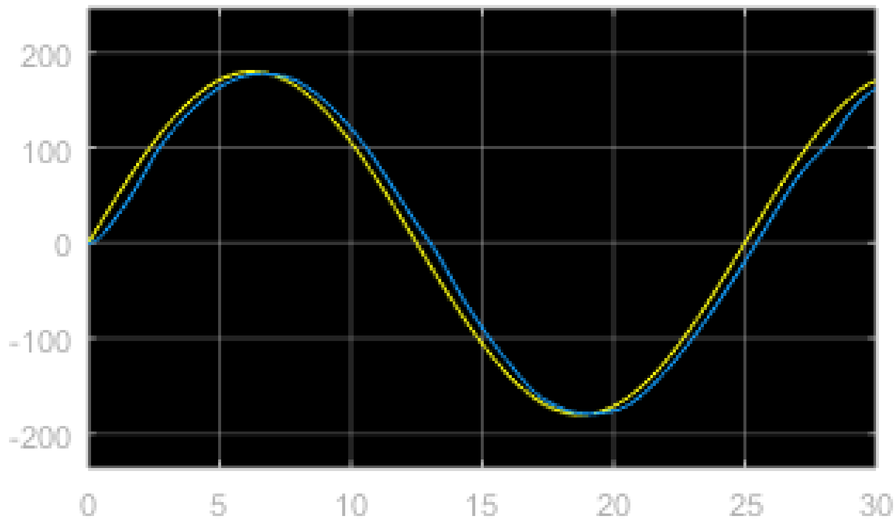

Figure 4: *LQR Yaw Version 2*

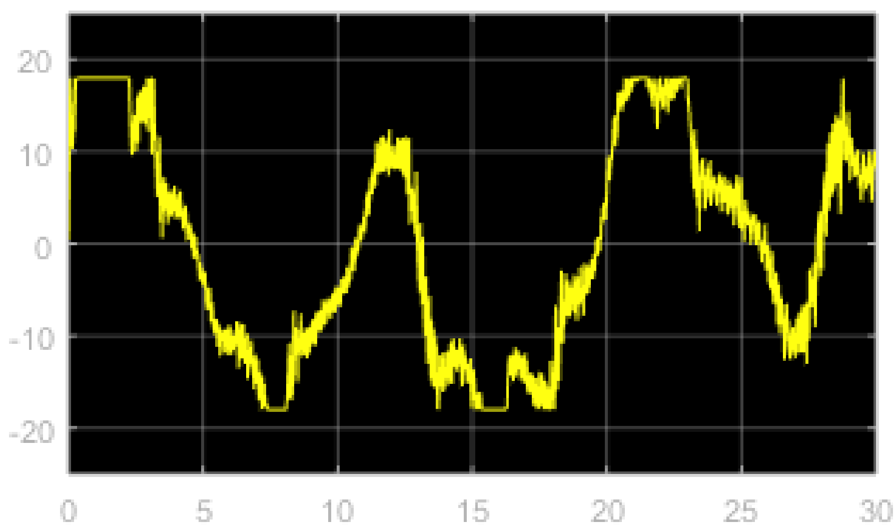

Figure 5: *LQR Yaw Voltage Version 2*

Figure 6 shows the second model proposed by Dr. Miah. This one retains the PI controller, however calculates the error between the state variable befor feeding it back into the system.
We kept the same Q and R values as in LQR.
Personally, I feel as though this model has less error than the previous version as well as Quanser's model. In out meeting next week, I will ask Dr. Miah if we should preform a RSME analysis to test the performance or if we should verify that the model will continue to work with LQG.

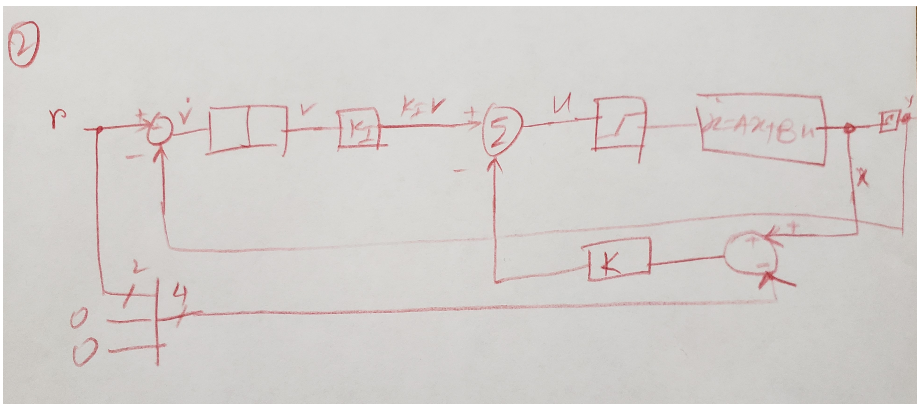

Figure 6: *LQR Block Diagram Version 3*

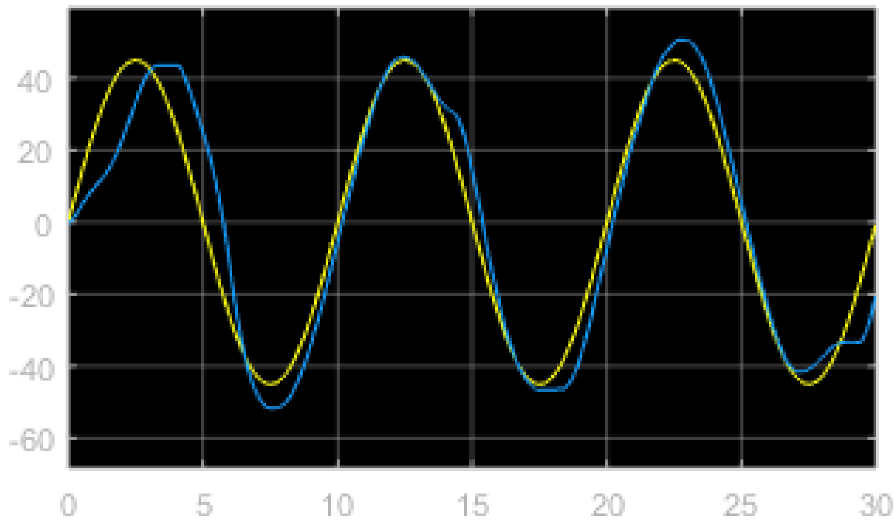

Figure 7: *LQR Pitch Version 3*

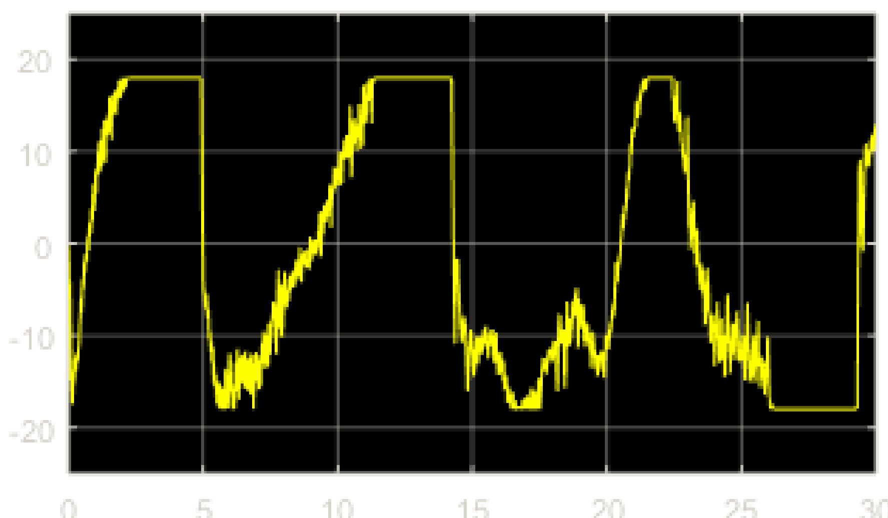

Figure 8: *LQR Pitch Voltage Version 3*

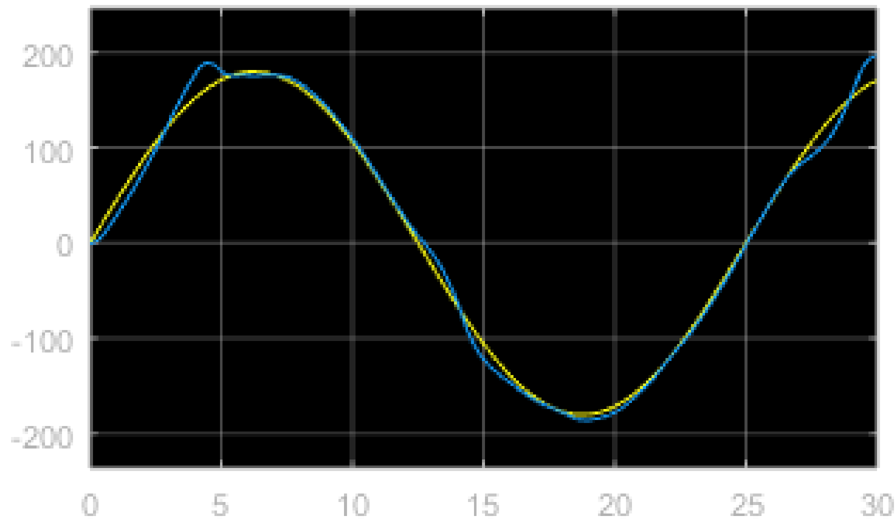

Figure 9: *LQR Yaw Version 3*

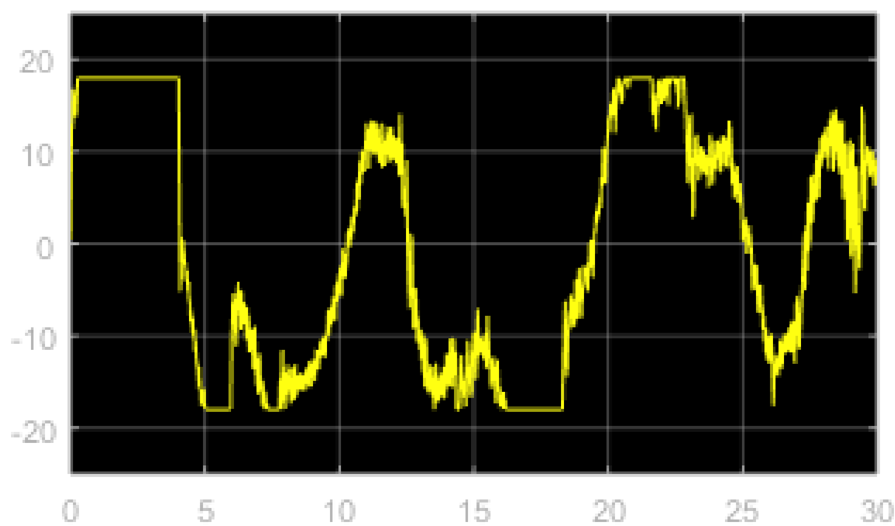

Figure 10: *LQR Yaw Voltage Version 3*

Monday, February 25, 2019

1 Meeting Minutes

KV
In our meeting today, we shared our results with Dr. Miah of the new block diagrams. He agreed that the second diagram discussed in the last journal entry preforms better and would like us to move forward with comparing the performance between it and Quanser. To do this, we will be calculating the RMSE as shown in equation 1.

$$RMSE = \sqrt{\frac{1}{N}\sum_{k=0}^{N-1}(\theta^d - \theta)^2} \tag{1}$$

The RMSE calculates the area of the error between the reference and actual signals.

Dr. Miah also asked us to think about ways in which we can make our algorithm smart. He instructed Glenn to do some research on what it means for something to be smart and asked me to work on determining the RMSE values as well as create the new LQG version.

Tuesday, February 26, 2019

1 Lab Work

KV
Today, Glenn and I worked on calculating the RMSE for LQR and created better plots of our results as shown on the following figures. We also calculated the performance gain of our new algorithm versus Quanser's using the equation in Dr. Maih's research paper. The pitch preformed 33% better and the yaw preformed 71% better. However, we also noticed that the new plot we generated using Quanser's LQR did not match the data we generated at the beginning of the semester. We might be using a different helicopter and this could be skewing our results.

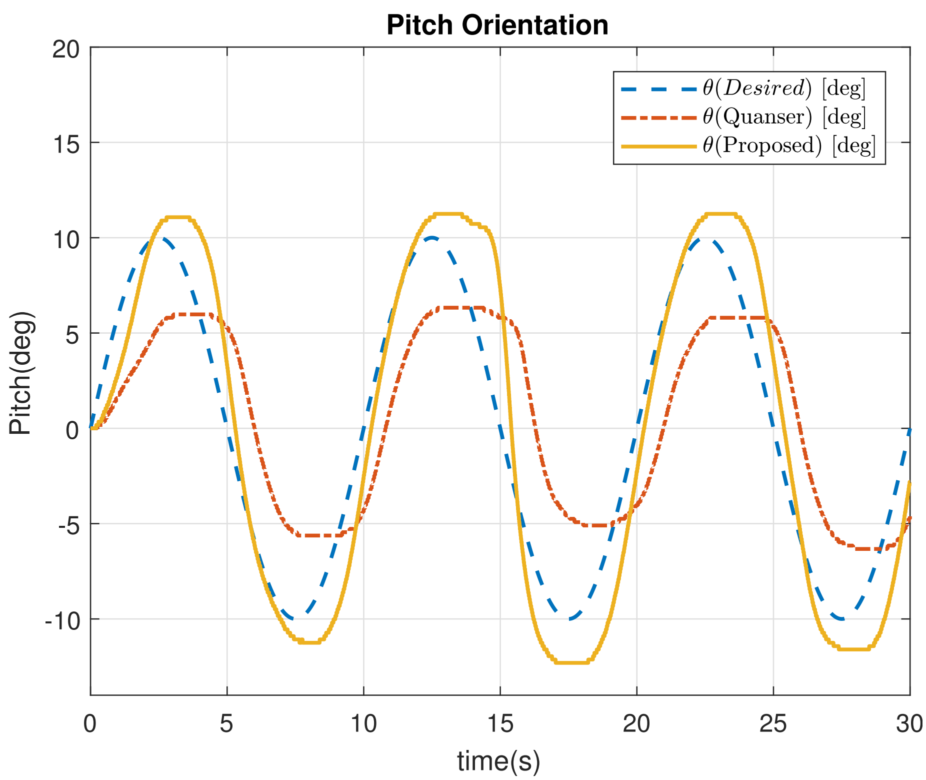

Figure 1: *Pitch LQR RMSE*

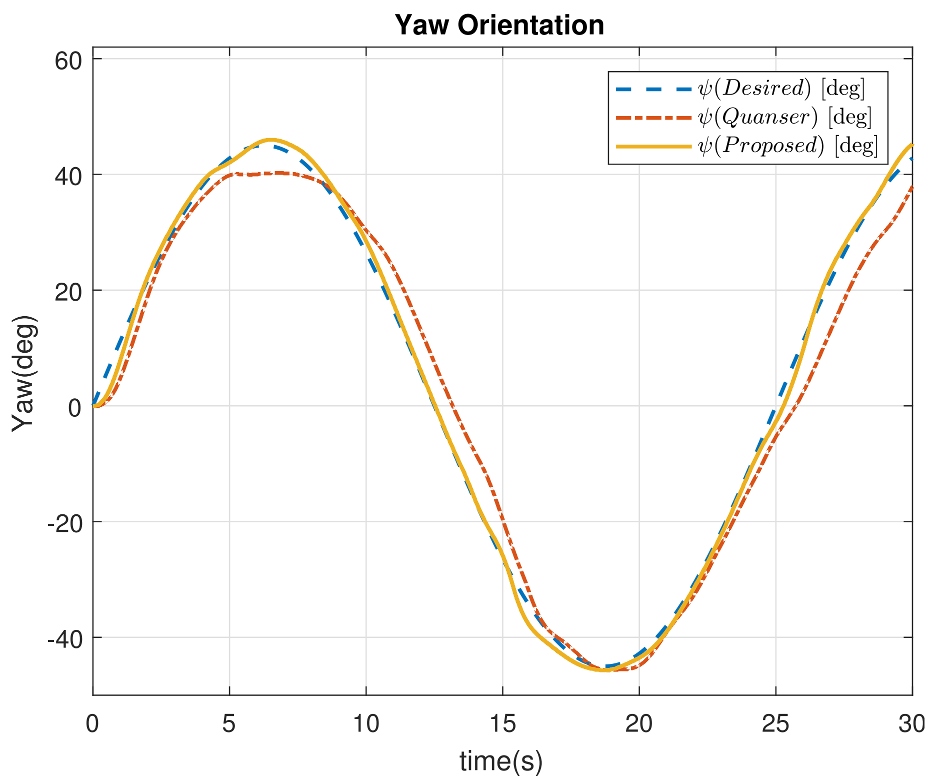

Figure 2: *Yaw LQR RMSE*

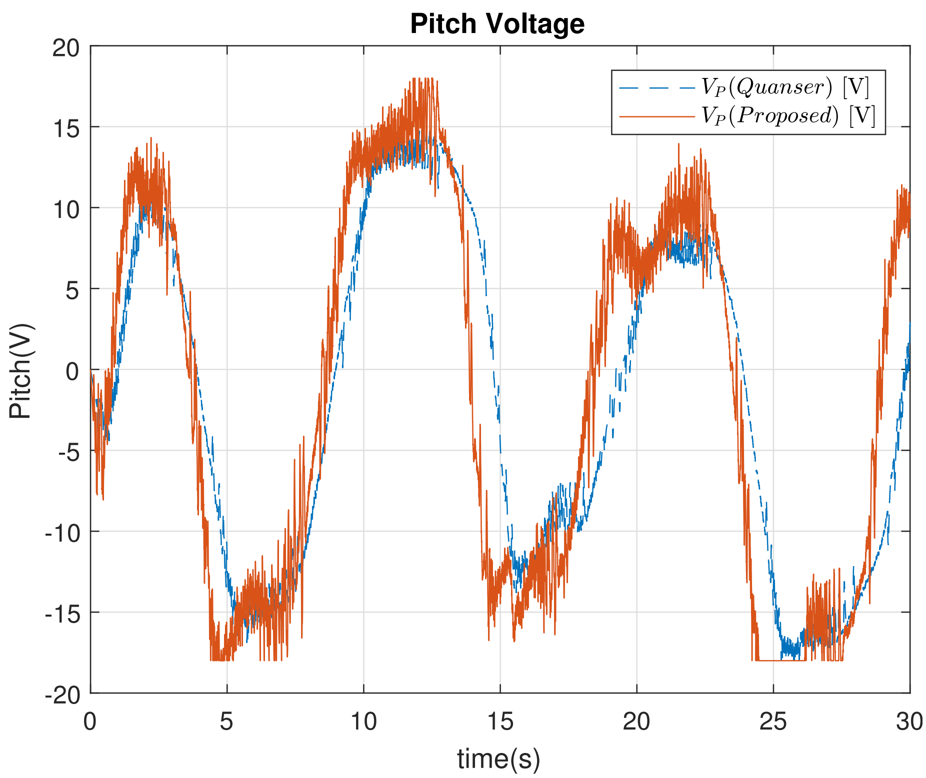

Figure 3: *Pitch Voltage LQR RMSE*

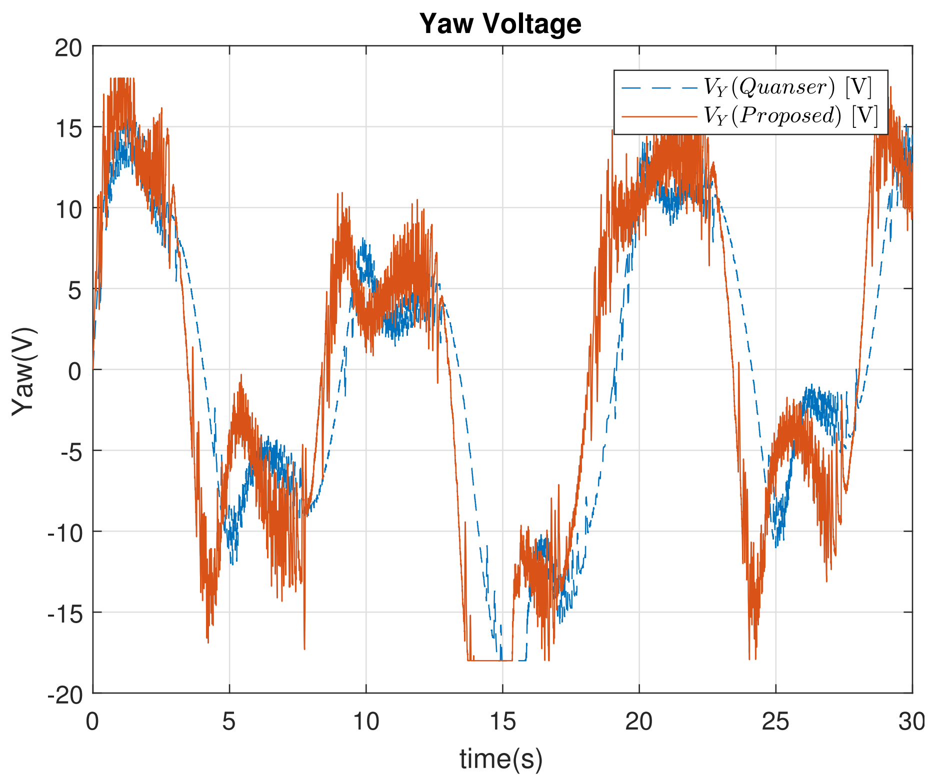

Figure 4: *Yaw Voltage LQR RMSE*

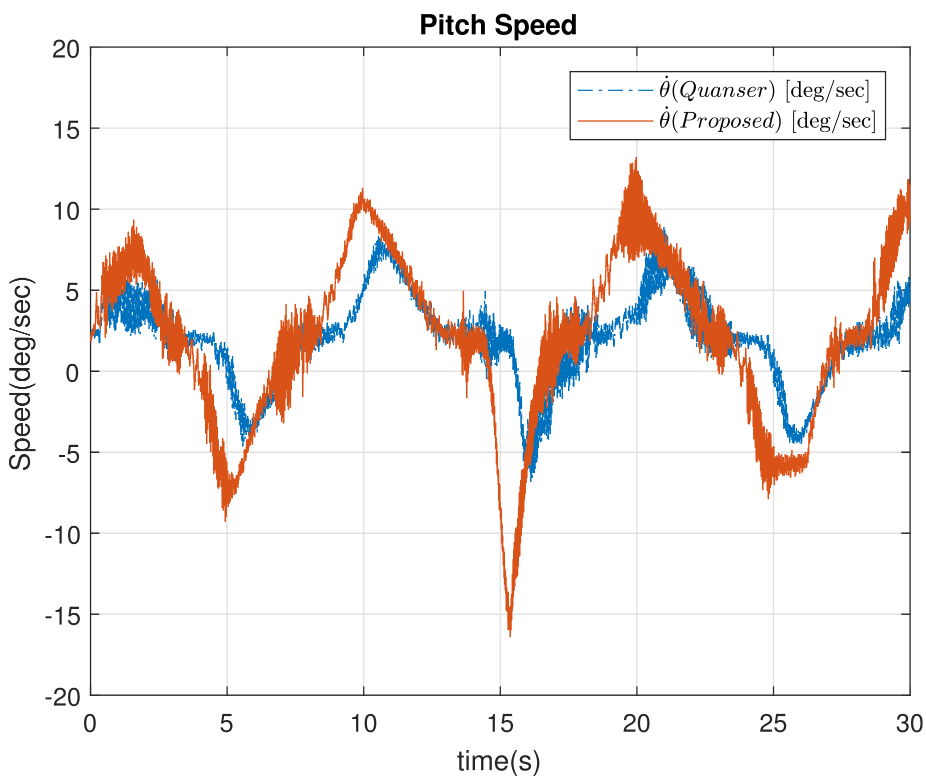

Figure 5: *Pitch Speed LQR RMSE*

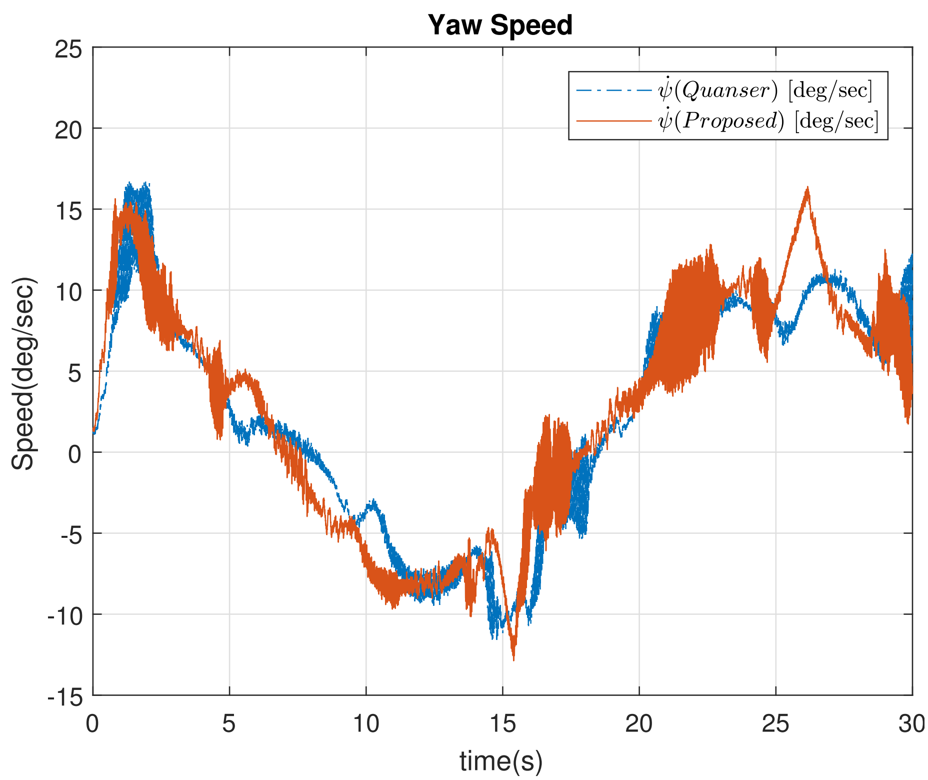

Figure 6: *Yaw Speed LQR RMSE*

Thursday, February 28, 2019

1 Lab Work

KV
Today, we plan on generating the same data for LQG as we did for LQR on Tuesday. After we have done that, we will then re-implement this on the Raspberry Pi. First, we are going to calculate the RMSE for a square wave and constant input signals.

When testing our new algorithm using a square wave, our pitch preformed 76% worse than Quanser and the yaw preformed 61% worse than Quanser. (To see these figures, go to the Google Drive, codeInProgress, RMSE, Graphing_Code).

When testing our new algorithm using a constant signal, our pitch preformed 40% than Quanser and the yaw preformed 31% worse than Quanser. (Once again, see drive for images).

When trying the LQG model with a sine wave, there was voltage applied to the motors, however not enough to move the helicopter. It could be that I made a mistake when creating the diagram. We will show it to Dr. Miah on Monday.

Now we will continue to implement our new LQR model on the Raspberry Pi. We are still encountering the same problem we had a couple weeks ago. 1 theory is that there are more calculations that need to be run than the sample time allows for on the Raspberry Pi. We tried changing the solver step size to the SPI period, but this did not work. We tried adding a scope to see what was going on but this stopped the model from running. When trying to run it in putty, we got an error saying: "error occurred getting packet header. error occurred in rt_PktServerWork."

Monday, March 4, 2019

1 Meeting Minutes

KV
In our meeting, we discused with Dr. Miah some of our difficulties with different input signals for LQR, problems with LQG, and problems with the Raspberry Pi.

To correct our problems with LQR, Dr. Miah suggested that we first try to change the values of the Q matrix and then multiply K by a constant greater than 1 and K_i between 0 and 1.

For LQG, Dr. Miah said we do not need to compare our LQG with Quanser's LQG.

When we get to the Machine learning part of this project, we will be talking with a research associate in Ottowa who specializes in reinforcement learning.

I conducted some research on Smart Devices, however my membership status dose not allow me to view some of them. Hopefully Dr. Miah will be able to help us gain access to some of them. I have also created a new figure which illustrates a way we could make our algorithm 'smarter'

- The main screen on the app allows a user to view helicopters that are available or in use by other users
- This information is relayed to the user by a server. The server is the only device the smartphone interacts with.
- The user can log into an available helicopter from the main screen.
- Once logged in to the helicopter, the user can change its orientation.
- These commands are sent to the server and the server sends the reference signal to the appropriate controller (Raspberry Pi).
- The controller applies a respective voltage and returns the actual orientation to the server.
- The server sends the actual orientation to the user's mobile device

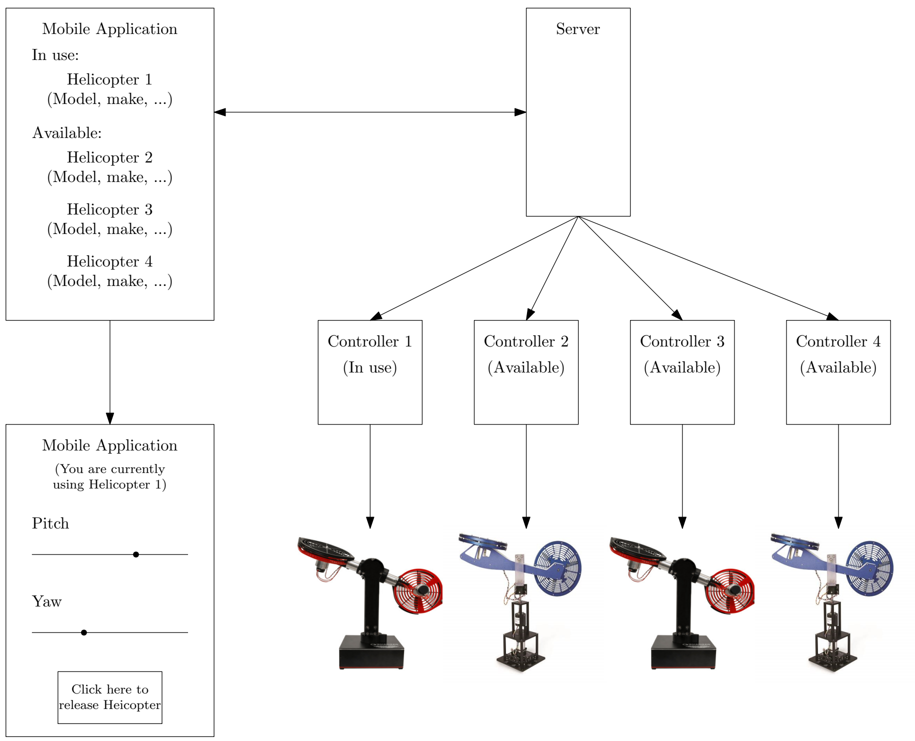

Figure 1: *Smart Algorithm*

Tuesday, March 5, 2019

1 Lab Work

KV
Today we tried removing the integral gain in our LQR model for the Raspberry Pi. Unexpected behavior is still experience on the Pi. We believe this could have something to do with our sampling times because we have a different sampling time for our SPI communication. However, this method worked fine with Quanser's model. The only difference is the value of K and its location in the model.

Thursday, March 7, 2019

1 Lab Work

KV
First, we tried to make changes to the parameters to the basic model (Quanser) which utilizes our gain values by reducing khat and using Quanser's parameters. This did not make any significant changes in the performance.

We then tried to figure out what was wrong with our LQG algorithm. It appears that not enough voltage is being applied to the helicopter to make any significant changes in orientation for a sine input. A constant input have much greater error than LQR. We have printed out figures of the results and plan to discuss them with Dr. Miah on Monday.

We then tried to get the data collection to work. We followed MATLAB documentation and procedure in our lab notebook, however even though the model builds, it does not run on the Raspberry PI. When we try to remove the scope and return the model to its orginal state, the model will still not run as if the file directory is somehow corrupted.

Monday, March 11, 2019

1 Meeting Minutes

In our meeting today, we discussed some possible solutions to correct some of the performance issues experienced in the USB versions of LQR and LQG.

- Improve LQR for constant input and square wave:
  1. Reduce first 2 Q values to 0.1
  2. Reduce K (0.95, 0.9, 0.8, 0.5)
- Fix LQG performance:
  1. Test by removing kalman filter
  2. Make noise a function
     - ζ (zeta-input noise) $= \sqrt{Q_e}$
     - ξ (xi-output noise) $= \sqrt{R_e}$
  3. Change Q_e and R_e
     - $Q_e = 0$
     - $Q_e = 10^{-9} * randn(n, 1)$
     - $R_e = 10^{-4} * randn(2, 1)$
     - NOTE: Q_e can be zero but R_e cannot be zero

Tuesday, March 12, 2019

1 Lab Work

KV
Today we began by trying to change the first two values of Q to 0.1. This made no noticeable difference so we decreased it even more to 0.001. The overshoot and settling time was still the same. We then decided to keep the values of Q at 0.1 and begin changing the proportional gain, K. We decreased it in intervals of 0.05 and noticed our overshoot was becoming worse. Hoping to counter this effect, we began to increase K in intervals of 0.05. As we did this, we noticed the oscillations in the voltage increased, however, this did increase our performance to a point until the voltage continually saturated on both ends while oscillating around the desired orientation. The constant multiple by K we found to work the best was 1.5, however, this consumes much more power. We then decided to put the K value back to its orginal by removing the constant multiplier. We started to increase Q and noticed some improvement by increase in the first two values of Q to 10. We ran our RMSE code again with the new values for K and Q so now the performance for our pitch is now 56.7% better vs. 40.2% and the yaw is now -10% worse vs. -31% worse. We still didn't like these results so we continued to play with the K and Q values. We made Q = diag[50 50 1 1 10 .5] and K = 1.35*K. Now our pitch preforms 62.4% better and is comparable to Quanser's yaw. By changing the last values in Q, the yaw overshoot was reduced, however it too longer to converge. Now that we have gotten the constant signal to work properly, we will go back and check our sine signal performance using the same parameters. Pitch performance increased from 31.1% to 64.6% and the yaw performance decreased slightly from 71.2% to 63.2%. We will now test our square wave signal with the new parameters. We observed a much better performance this time, only 10.8% worse vs. 76% for pitch and -4.2% worse vs. 61% for yaw. There is still room for improvement and I will try to find better parameters after I am finished with class and Glenn is in lab.

I made a slight change to the last value in Q which improved the performance of the square wave to be 7% worse than Quanser in terms of pitch and 3% worse for yaw.

Thursday, March 14, 2019

1 Lab Work

KV

Monday, March 25, 2019

1 Meeting Minutes - LQG

KV

Today, we discussed some of the issues we are having with LQR and LQG. We discussed two options to correct LQG. If these do not work, we will move on to ADP and work on the other algorithms outside of our scheduled lab time.

- Using PI controller on Raspberry Pi (both LQR and LQG)
- LQG estimated state from Kalman filter are diverging (only θ)
  1. replace θ with $\hat{\theta}$ using a demultiplexer
  2. WrapToPi(θ)

2 Meeting Minutes - ADP Overview

KV

Tuesday, March 26, 2019

1 Lab Work

KV
We tested some of the suggestions for LQG but realised that the other estimated states were not what they should be causing the similar problem. This means that the problem is with the kalman filter itself and not just the one diverging state variable.

We tried to run last year's USB version of ADP, however their were some compatibility issues with different versions of MATLAB regarding data-logging so we will completely remake the model. We are going to try to understand how their code work first before proceeding.

Thursday, March 28, 2019

1 Lab Work - ADP USB

KV
Today, we are beginning to remake last year's model of ADP. We also have some questions on some of the code that we will ask Dr. Miah on Monday. We are collecting figures and RMSE data for ADP. We did notice a considerable amount of steady state error, similar to the result we had when we just using a P controller for LQR and LQG.

2 Lab Work - ADP Raspberry Pi

KV
Took longer for gain to converge than USB

3 Lab Work - ADP Android

KV
Much more oscillation that USB. For some reason, we are not reading the actual position. next week, we will reinstall the andoid app on Glenn's phone.

4 Table of Completion

The following table show what we have completed this far.

	LQR(P)	LQR(PI)	LQG(P)	LQG(PI)	ADP(P)	ADP(PI)
Simulation	X	X		X		
USB	X	X			X	
Ras. Pi	X				X	
Android	X				X	

Table 1: *Current Progress*

When we meet With Dr. Miah on Monday, I will ask him which data he would like us to use on our poster board for the scholarship expo. We also saw a IAB poster and were curious on the difference between the two. He will also answer our questions regarding the ADP MATLAB code.

Monday, April 1, 2019

1 Meeting Minutes

KV
Today, we discussed:

1. How the ADP code works
2. What will be include in the Poster
   - LQR(P) simulation, USB, Ras. Pi, Android
   - LQR(PI) simulation, USB
   - ADP(P) simulation, USB, Ras. Pi
3. Difference between Expo poster and IAB poster
   - Same poster, different verbal presentations
   - Expo - Competition (higher level)
   - IAB - Presenting to alumni (more technical)
4. Draft for Final report

2 Meeting Minutes - ADP Code Overview

KV
While going over the code, I notice that nbar (number of samples in inner loop) was set to 100 which was inconsistent in how it should be calculated (T/Tau).

Tuesday, April 2, 2019

1 Lab Work

KV

Today we fixed the error in nbar and reinstalled the app on Glenn's phone because of a significant time delay to report the actual orientation. This did not solve our problem. I thought that part of the reason could be that since the router is not connected to the internet, the PC may be sending it requests that it could not fulfill. Using wireshark, I was able to observe some of the communication in which the PC was continuously sending requests to web pages open the the web browser. We have asked Mr. Mattus to update Android Studio to see if this fixes the problem.

Thursday, April 4, 2019

1 Lab Work

KV

Tuesday, May 7, 2019

1 Writing Report

GJ
Today was dedicated to writing the final report and going through all of our deliverables to correct any file paths for figures and sources.

Bibliography

[1] Quanser Inc., *Quanser Aero User Manual* . Markham, Ontario, 2016. 7

[2] Quanser Inc., *Aero Integration Workbook* . Markham, Ontario, 2016 8

[3] M. Hernandez-Gonzalez, A.Y. Alanis, E.A. Hernandez-Vargas, *Decentralized discrete-time neural control for a Quanser 2-DOF helicopter* . 2012 9

[4] F. Dos Santos Barbosa, *4DOF quadcopter: development, modeling and control* . 2017 9

[5] G.G. Neto, F.S. Barbosa, B.A. Angélico, *2-DOF helicopter controlling by pole-placements* . 10

[6] Weinan Gao and Zhong-Ping Jiang *Data-driven Adaptive Optimal Output-feedback Control of a 2-DOF Helicopter* . 10

[7] P. Méndez-Monroy, H. Benítez-Pérez, *Fuzzy Control with Estimated Variable Sampling Period for Non-Linear Networked Control Systems: 2-DOF Helicopter as Case Study* . Transactions of the Institute of Measurement & Control 34, no. 7 (October 15, 2012): 802–14. 11

[8] Quanser Inc., *Aero 2 DOF Laboratory Guide*. pgs. 26-39 12, 20