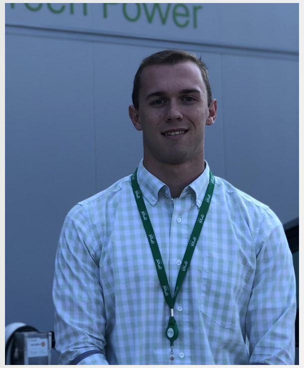
GyroGlove

Final Project Review

Senior Design Project ‘21
Team #21

University of Massachusetts Amherst BE REVOLUTIONARY™

Meet the team



Bradley Spillert
CSE

Jacob Moynihan
EE

Son Pham
EE

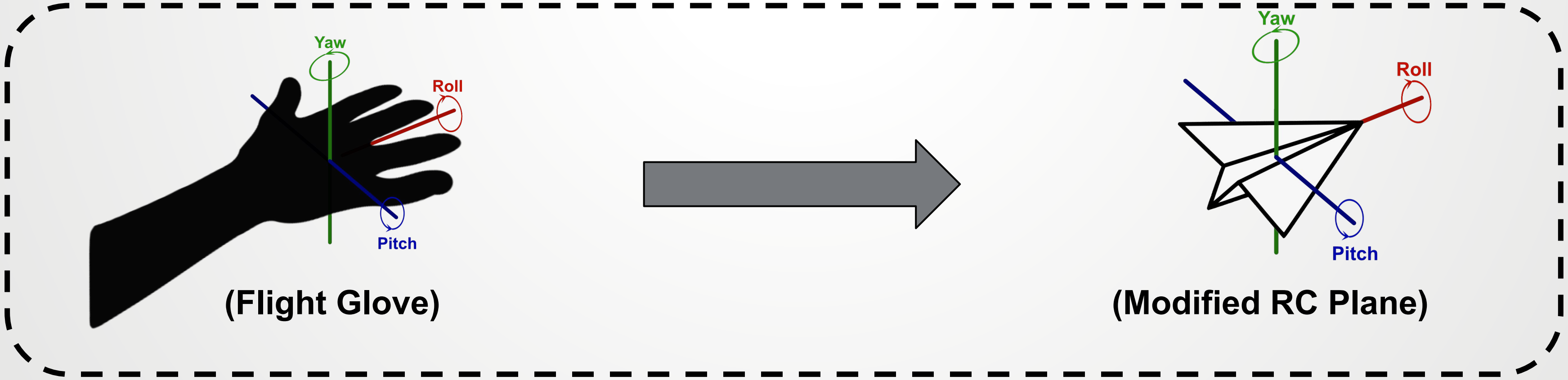
Professor Do-Hoon Kwon
Team Advisor

Problem Statement

Having a reliable flight control system is paramount for the success of any manned aircraft. While the current convention of using control mechanisms akin to analog sticks and steering wheels has proven to be timelessly effective, such mechanisms can often lack one valuable prospect: The immersion factor.

Problem Statement (cont'd)

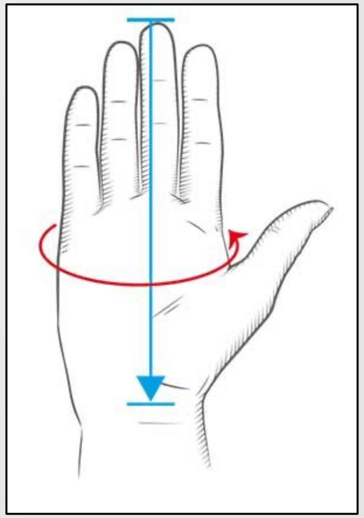
This is where GyroGlove comes into play. GyroGlove is a modern alternative to the classical stick-lever-wheel approach to flight control, ultimately allowing the pilot to control the aircraft simply by rotating their hand (orientation control) and curling their fingers (thrust control). These systems work elegantly in tandem to provide an immersive sensation of being "one with the aircraft", so to speak.



GyroGlove System

GyroGlove - System Specifications

- Flight glove battery life exceeds RC plane battery life by at least 100%
- Aircraft rotation stays within ±5.0° of IMU rotation (±7.5° for yaw axis)
- IMU capable of detecting rotation between ±85.0° along each axis
- Flex sensor capable of generating dynamic output over 90.0° flex range
- Functional range of at least 100 meters
- Control latency of 20 milliseconds or less within the functional range
- Operating frequency of 2.4GHz
- Compliance with all FCC standards and regulations



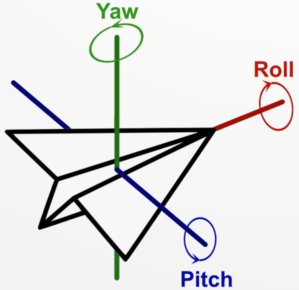
Subsystem Overview

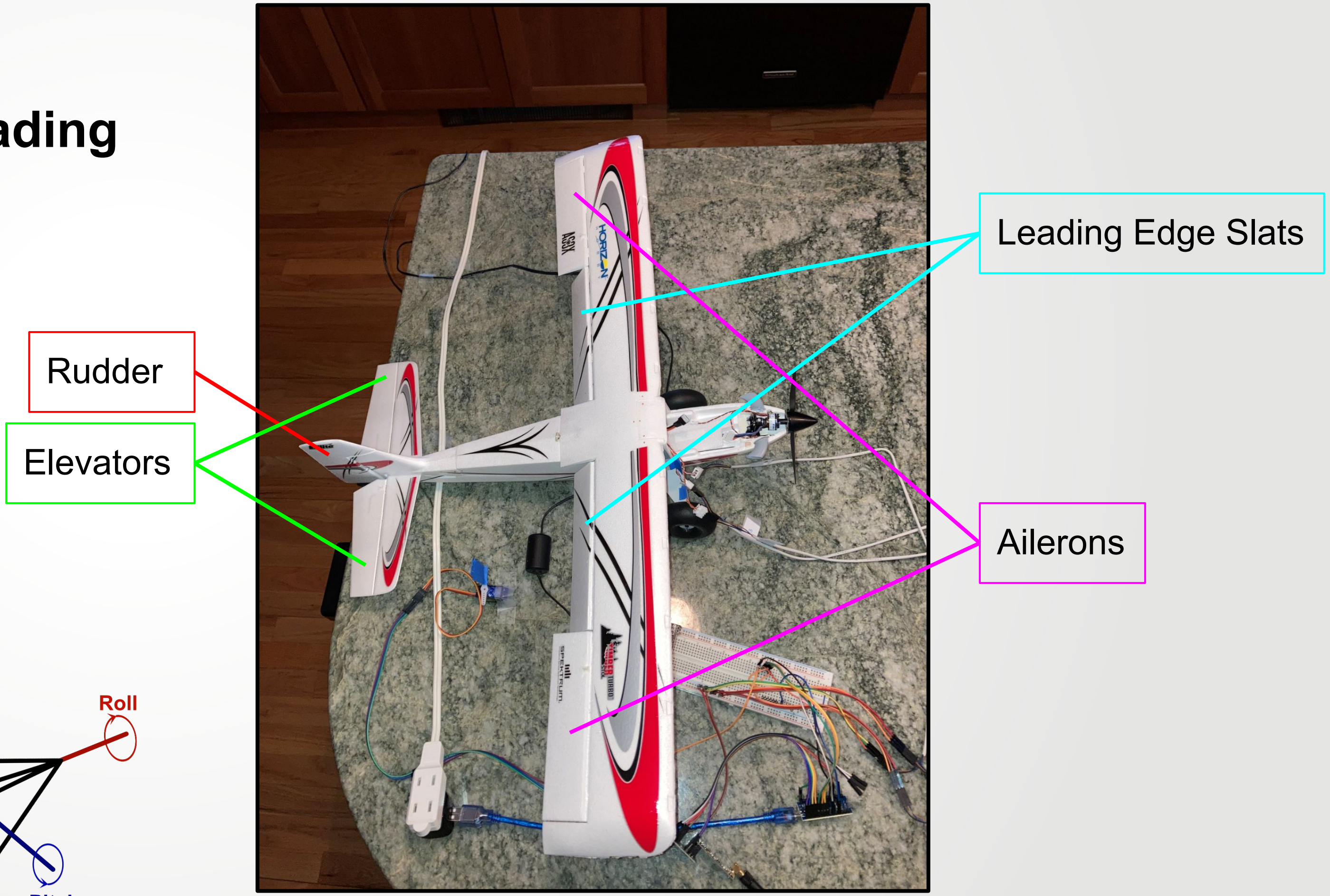
1. **Flight Glove: Captures/interprets movement of the pilot's hand and encodes data for TX to RC plane**
   1.1. Rotational movement capturing: Using hand-mounted IMU
   1.2. Thrust control capturing: Using 3" flex sensor mounted along middle finger
2. **RC Plane: Receives encoded actuation data from Flight Glove and animates existing RC plane actuation hardware**
   2.1. Pitch/yaw/roll servo groups: Control the RC plane's rotational orientation, as defined by glove IMU movement
   2.2. Engine/ESC: Controls the RC plane thrust, as defined by glove flex sensor tension

Axes of Movement & Actuation Hardware

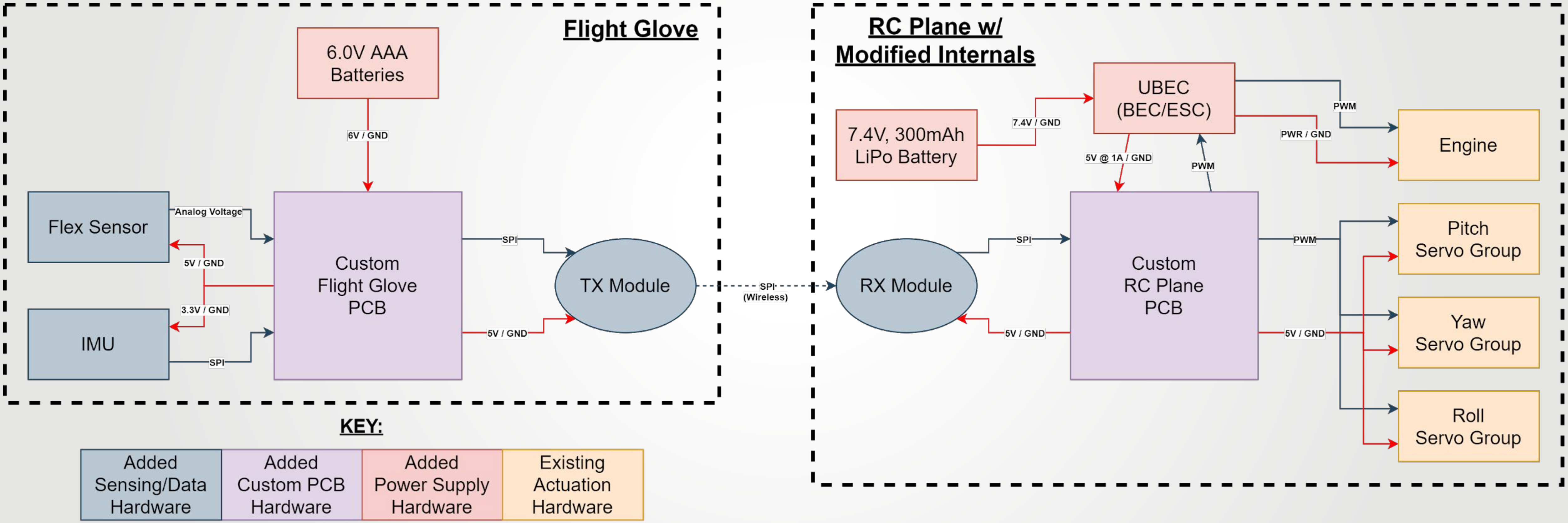
- **Pitch (controlled by elevators, leading edge slats)**
  - Actuation: 2x servos
- **Roll (controlled by ailerons)**
  - Actuation: 2x servos
- **Yaw (controlled by rudder)**
  - Actuation: 1x servo

** # of servos is unique to our particular demo aircraft. **

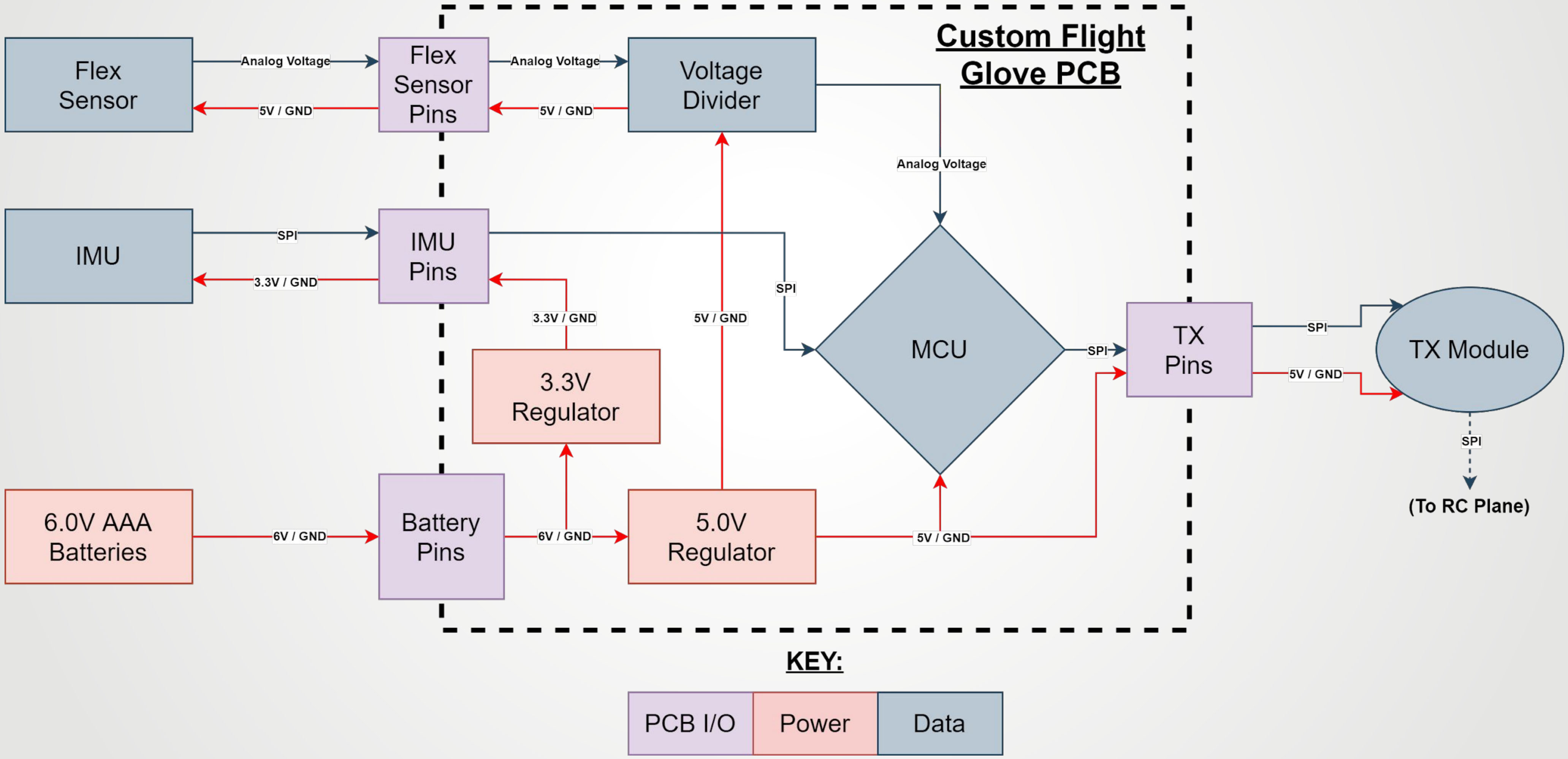




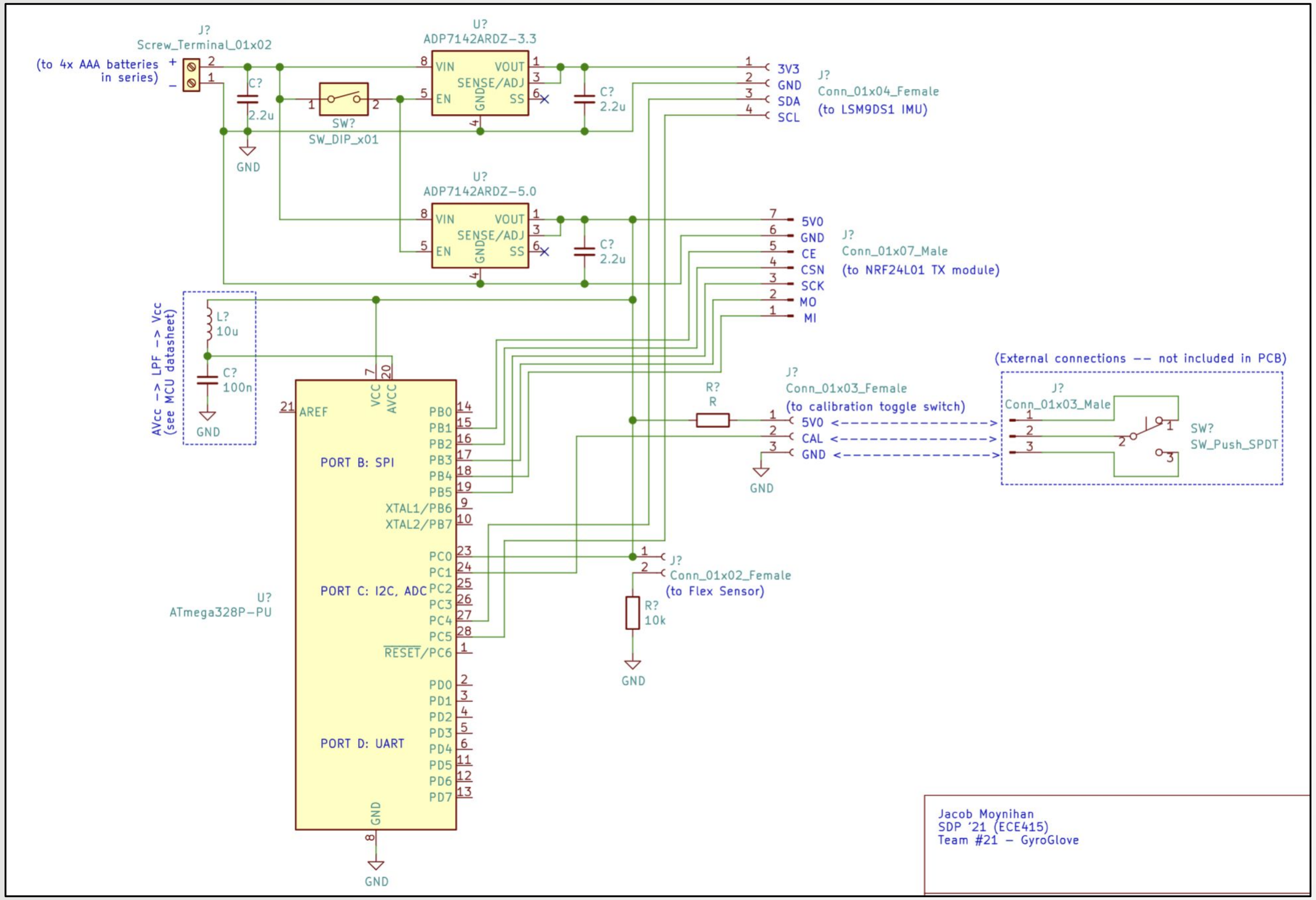
Hardware Block Diagram (Overview)

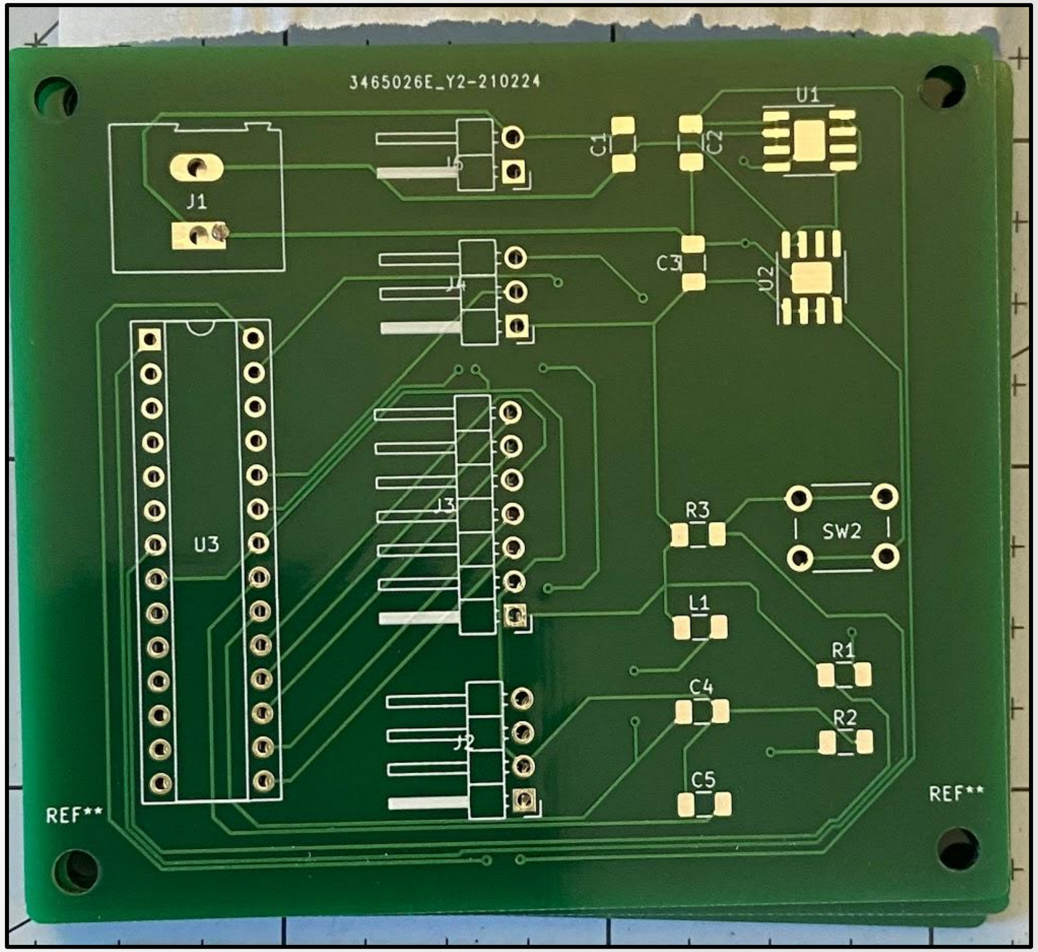


Hardware Block Diagram (Flight Glove)

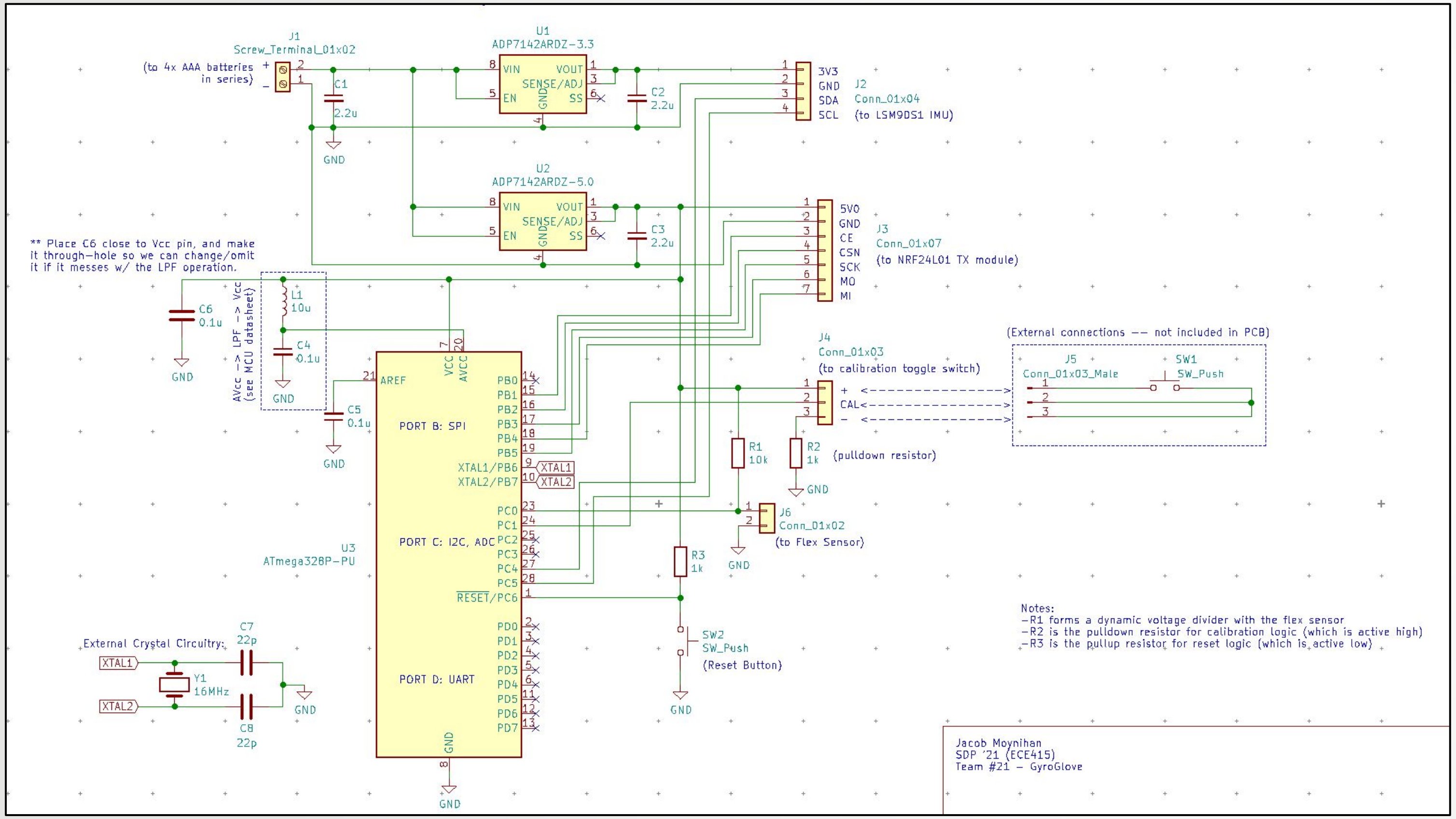


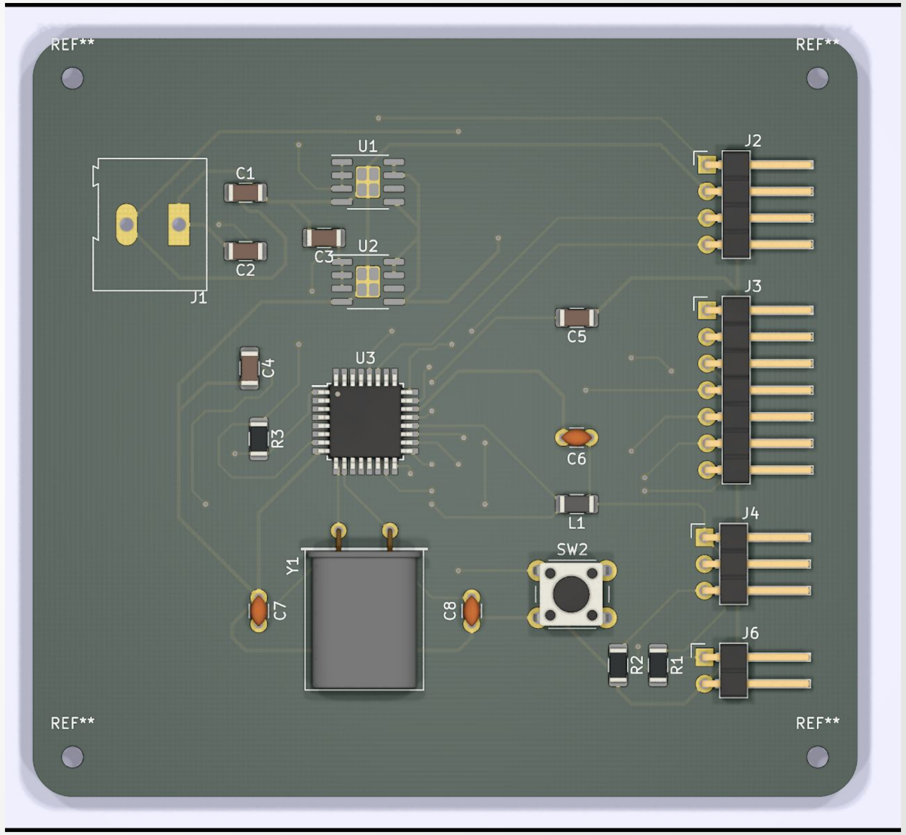
Custom TX PCB (Glove) - 1st Iteration



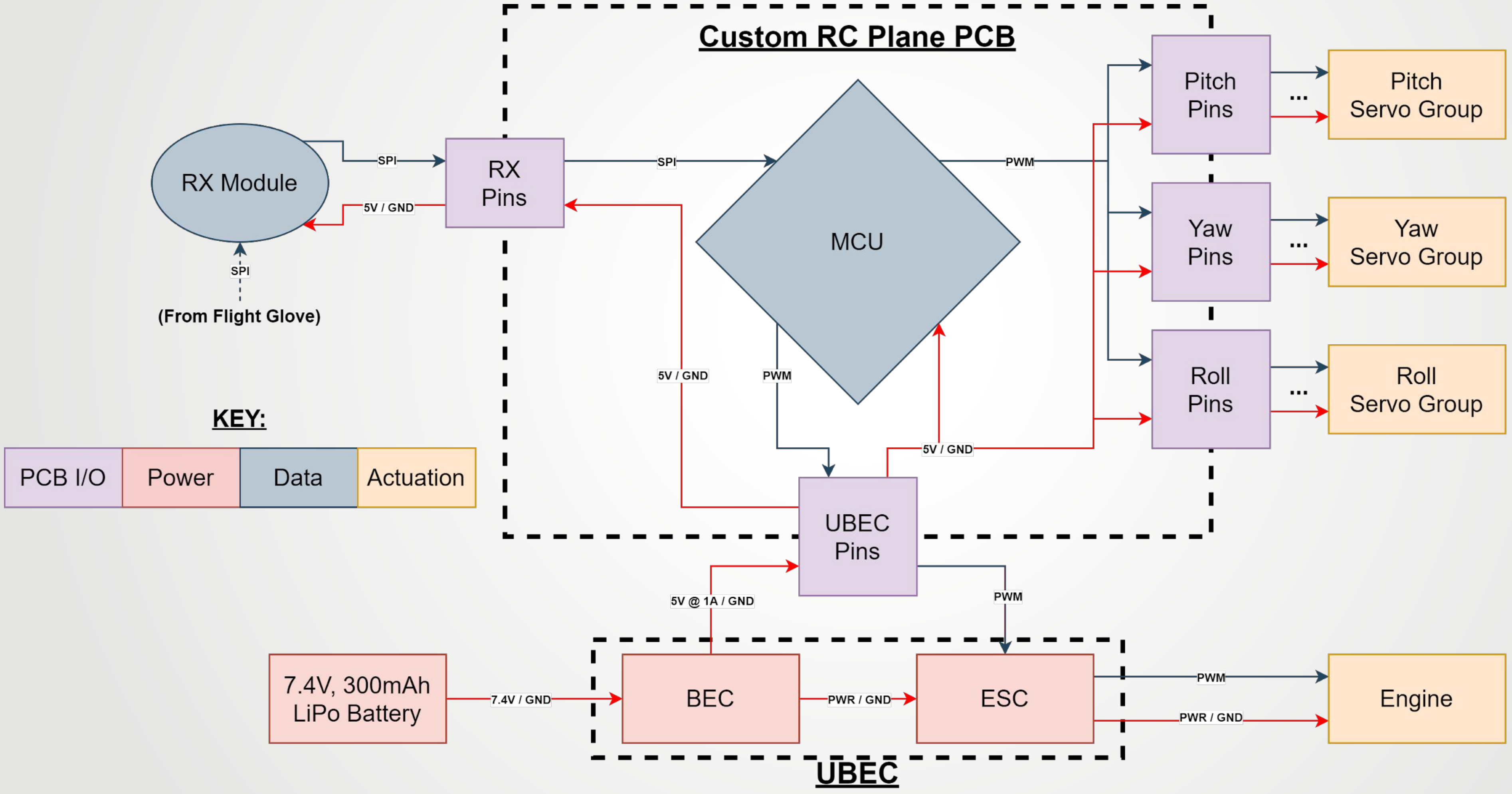


Custom TX PCB (Glove) - 2nd Iteration

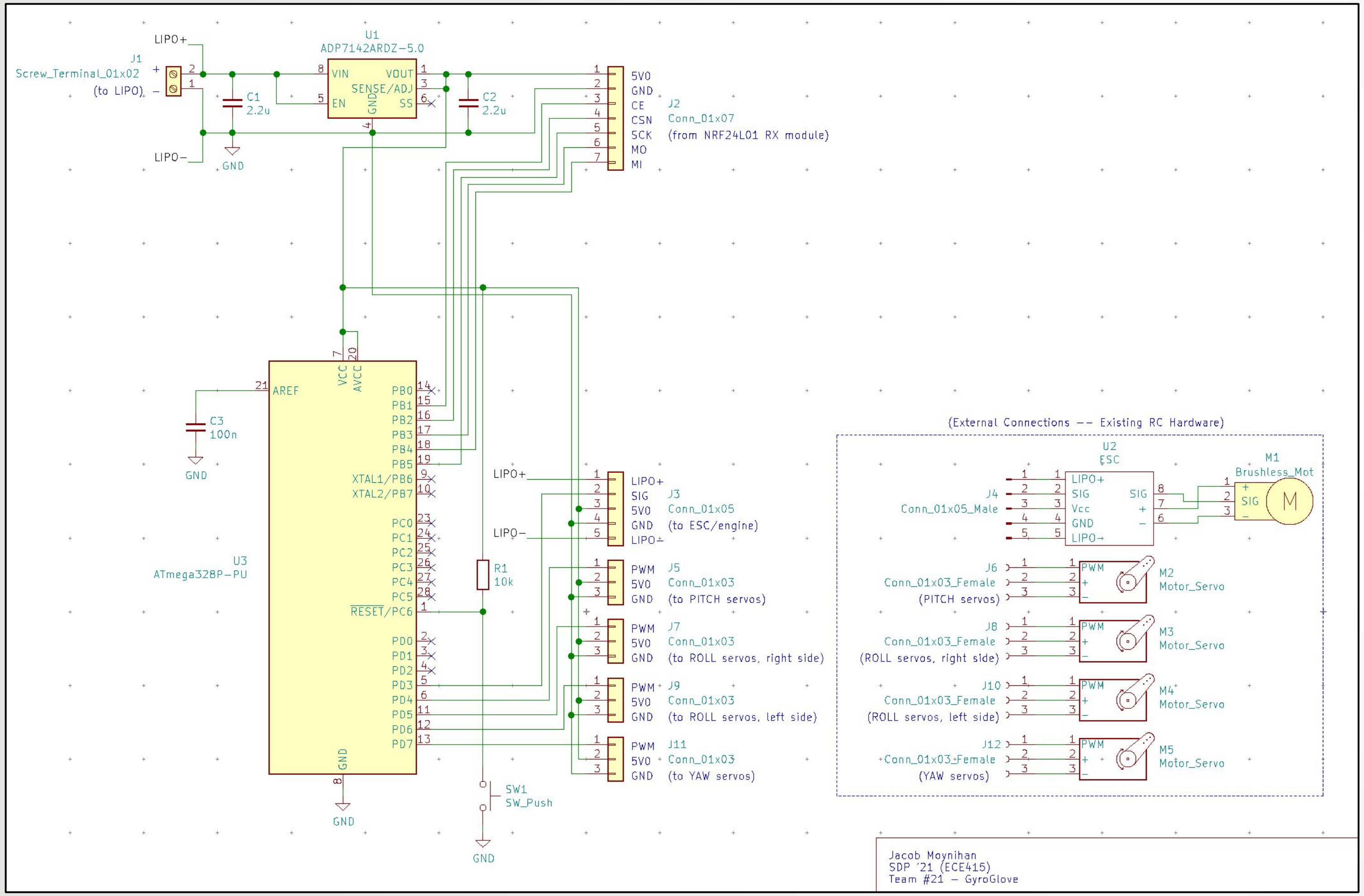


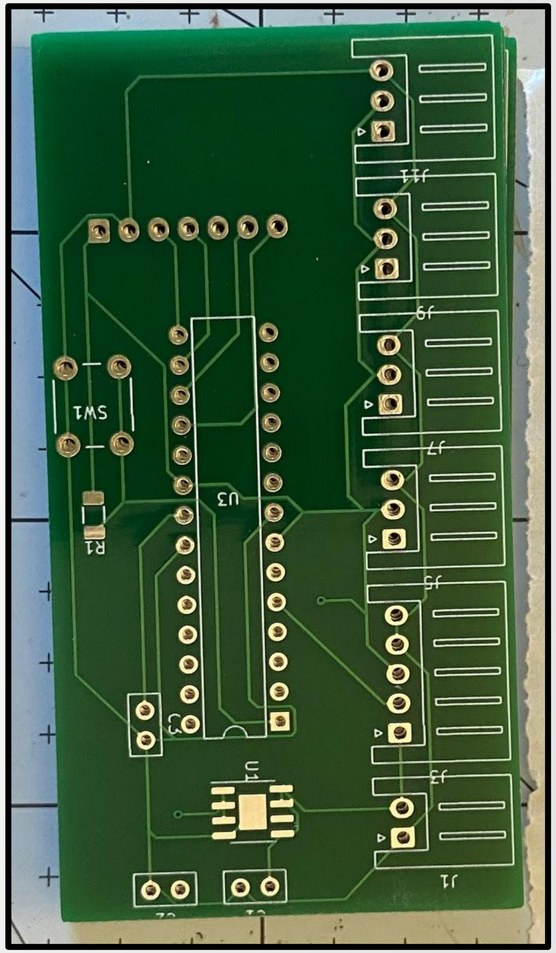


Hardware Block Diagram (Modified RC plane)



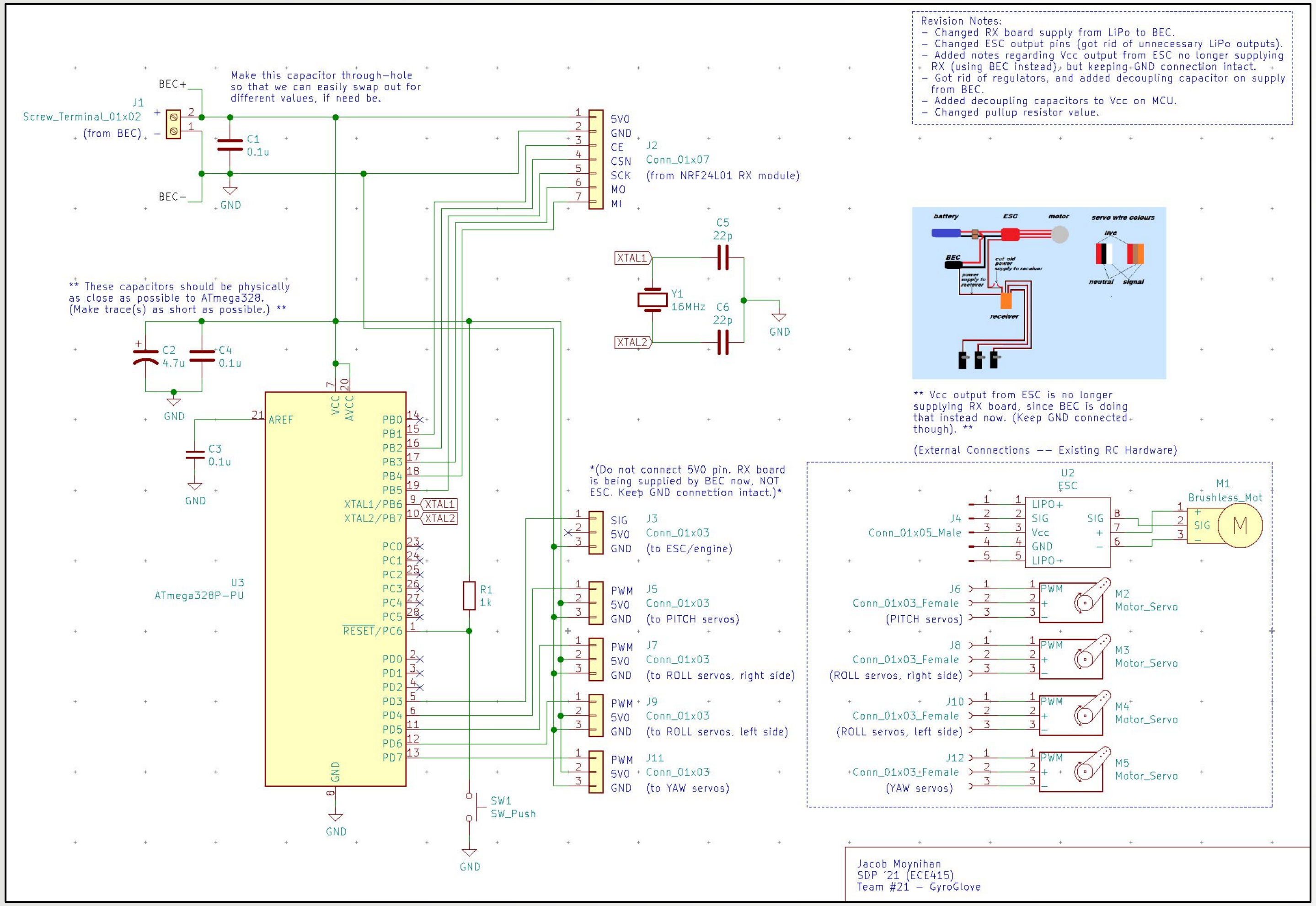
Custom RX PCB (RC plane) - 1st Iteration





University *of* Massachusetts Amherst BE REVOLUTIONARY

Custom RX PCB (RC plane) - 2nd Iteration

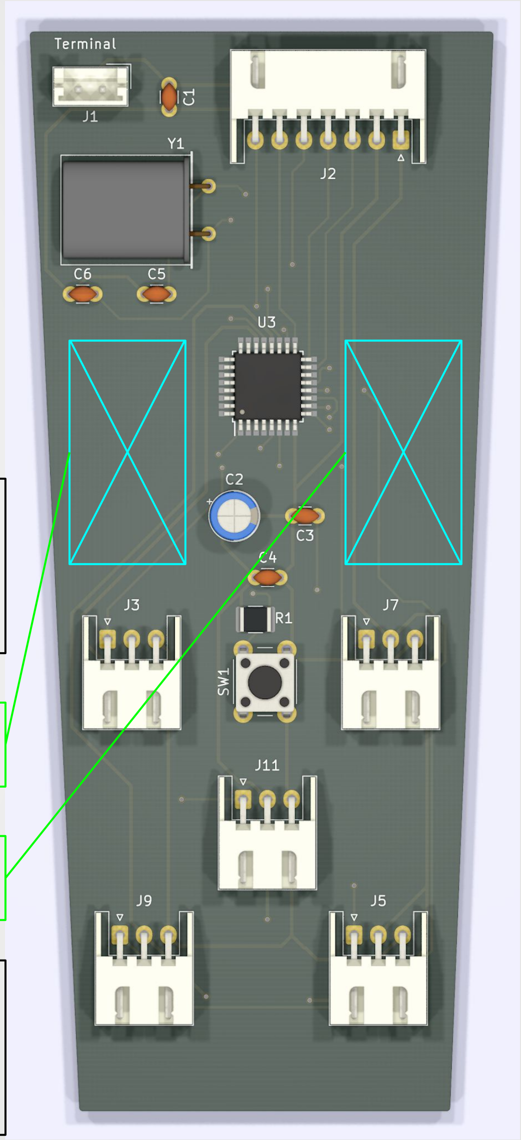


Shape of the RX PCB conforms to RC plane interior.

Elevator Servo (mounted)

Yaw Servo (mounted)

**Both mounted servos actuate their respective control surfaces via mechanical cables.



LiPo Power Division & Delivery

- **We're using a "universal battery elimination circuit" (UBEC), which contains both a BEC and an ESC in one module/package**
  - The BEC (battery elimination circuit) allows us to siphon a safe amount of power from the LiPo battery to the PCB (5.0V @ 1-2A), while still allowing sufficient power through to the ESC/engine
  - The ESC (electronic speed controller) allows us to control the main engine by means of a low-power PWM signal sent from the MCU

**UBEC will be equipped with discharge protection.

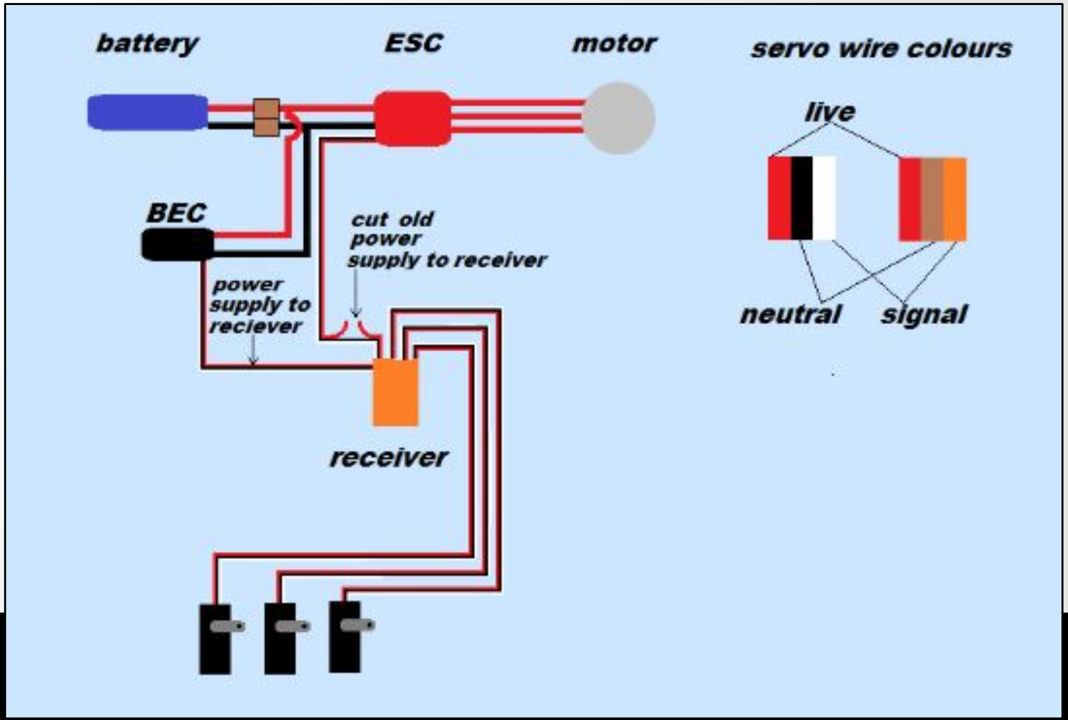
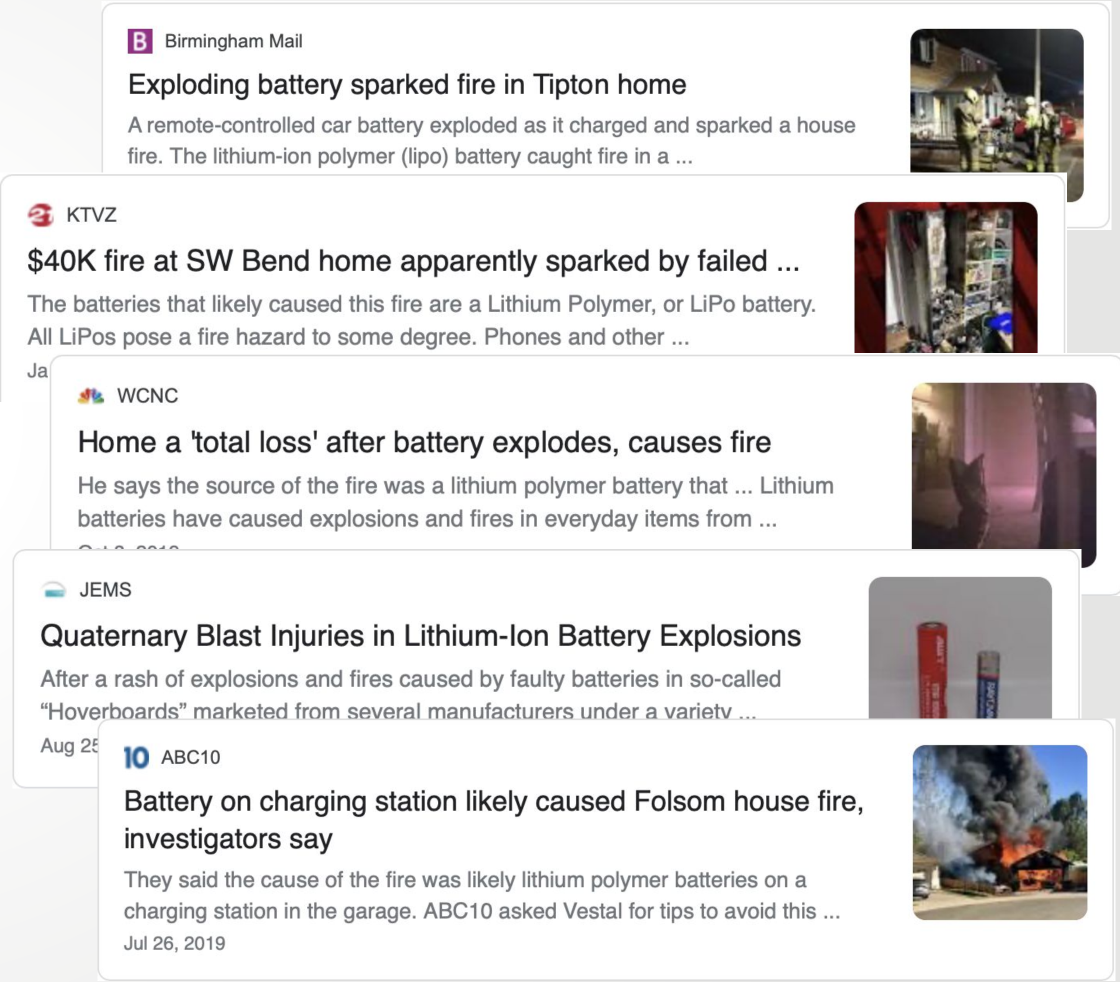


Image: *http://www.rcplanes4beginners.co.uk/bec-wiring-diagram.php*

LiPo Power Division & Delivery

- **Cells must be properly balanced and balanced charged**
  - No difference > 5-10mV
- **Charge and store in fireproof container**
  - Glass, metal bowl/box
  - Never leave unattended while charging
- **Never let cells get under/over 2.9V / 4.2V**
  - Nominal voltage 3.2-4.2V
- **Store with cells between 3.6V - 4.8V**
  - Never leave fully charged for more than 1-3 days
- **Never puncture**
  - dispose of old LiPo batteries properly
- **These are the the most important of many rules**



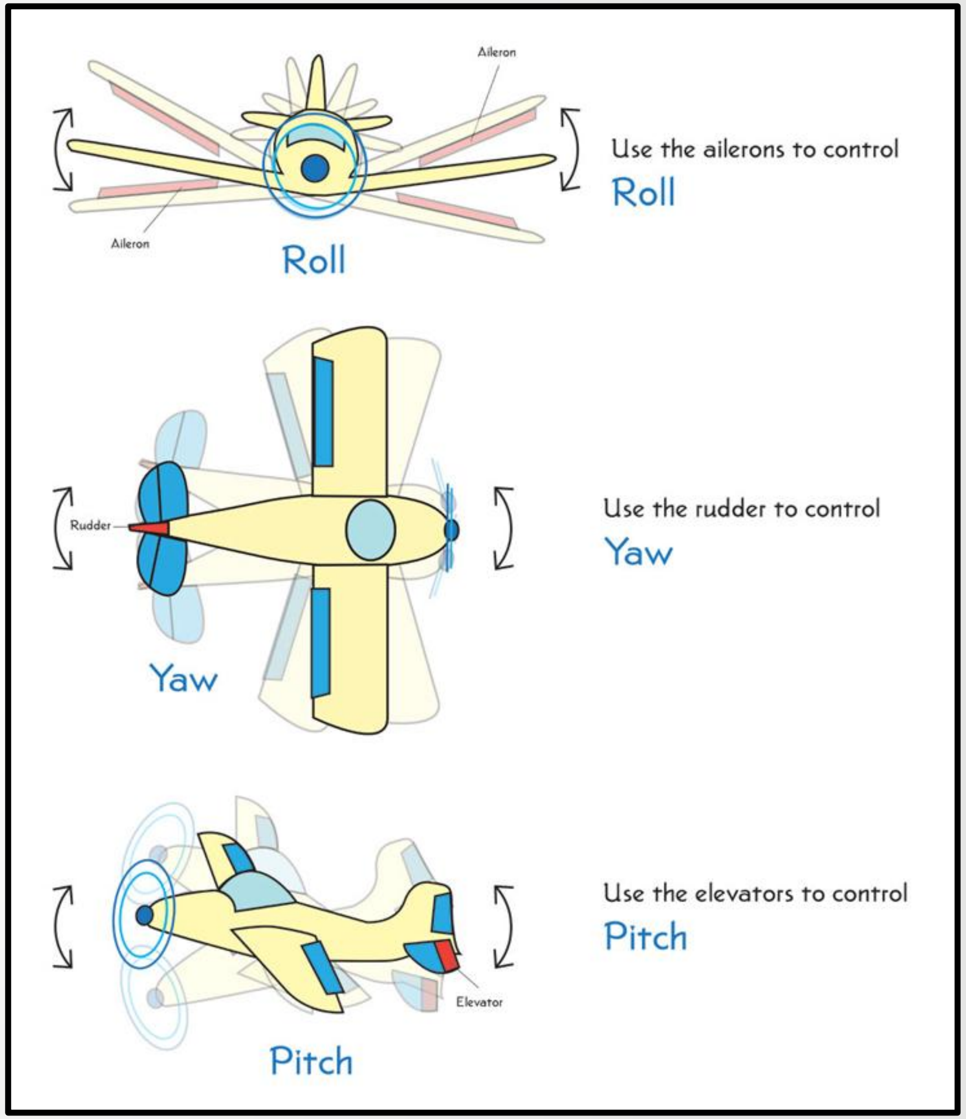
Accelerometer vs. Gyroscope

Gyroscope measures the rate of angular change (of the hand), therefore it must be continuously integrated

- **Used for Yaw**

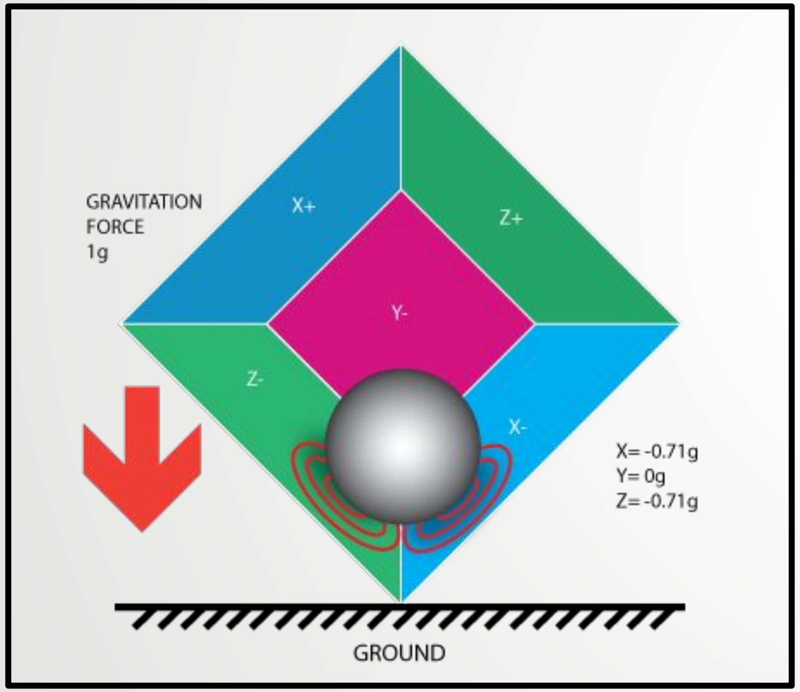
Accelerometer measures the acceleration vector on an axis, therefore it does not require any integration

- **Used for Roll (X) and Pitch (Z)**

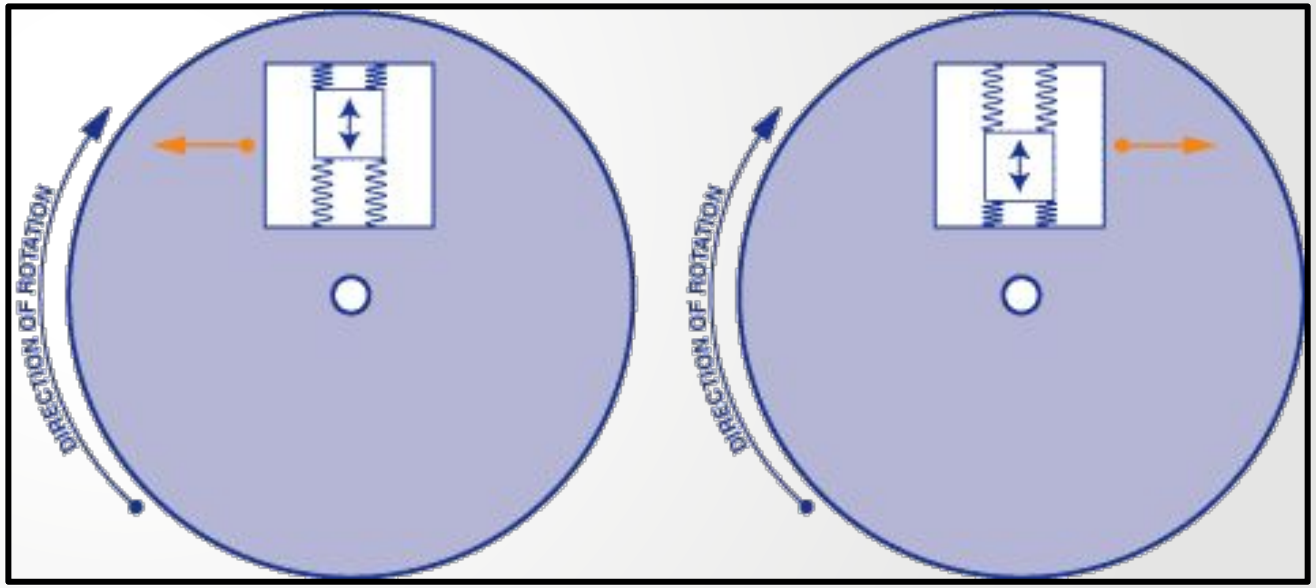


Accelerometer vs. Gyroscope

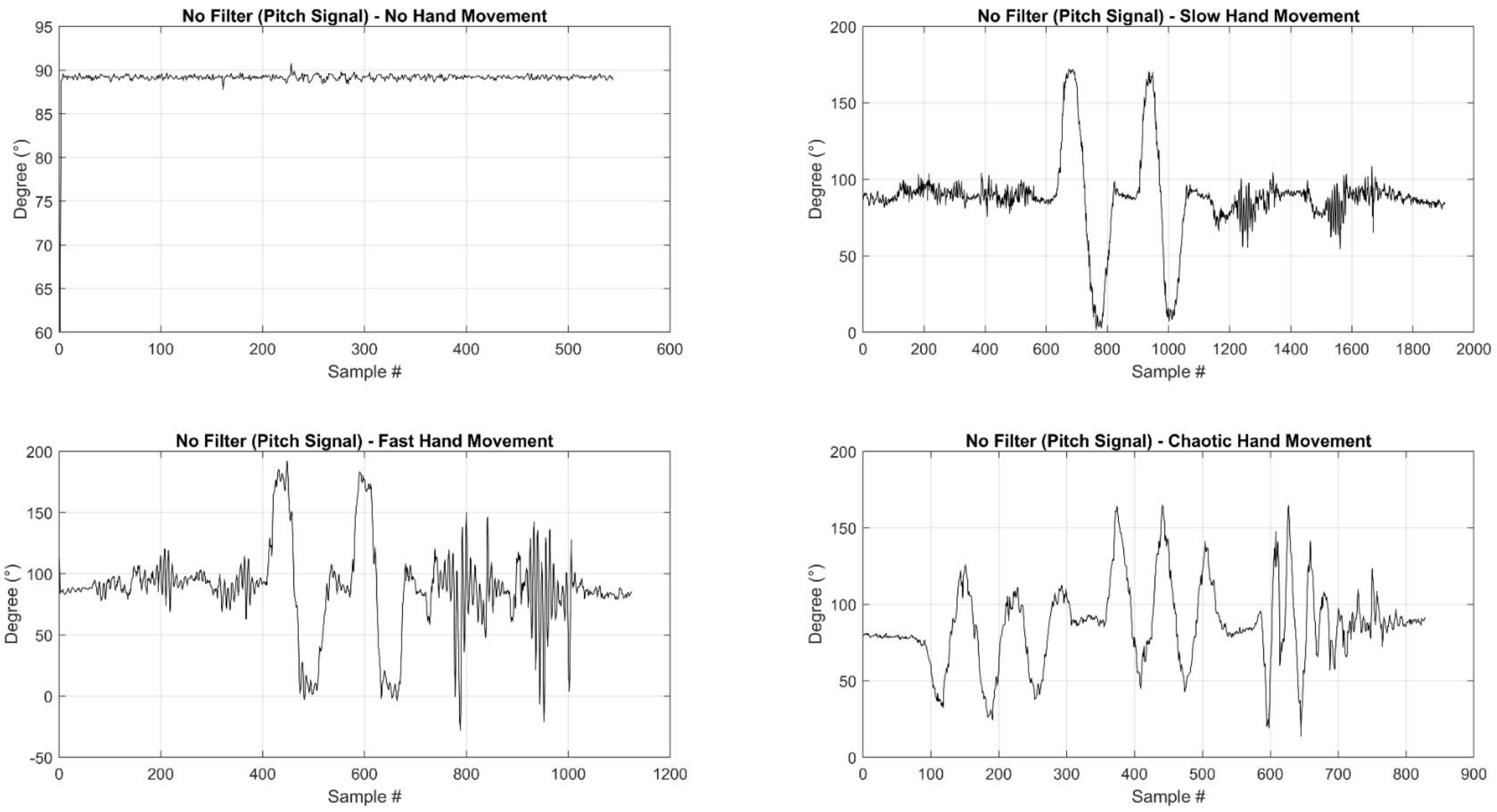
Accelerometer



Gyroscope



Signal Filtering - Raw Data

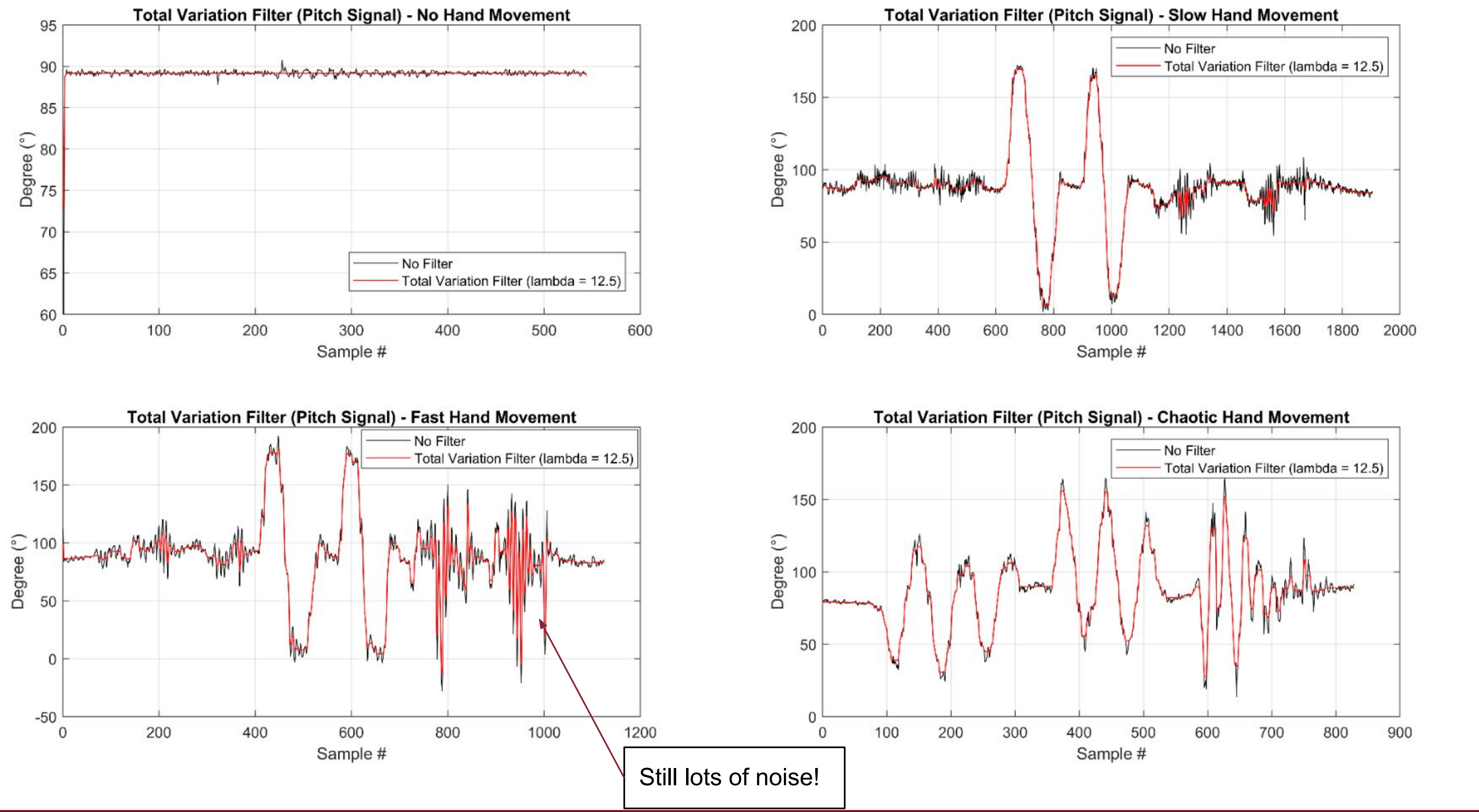


Speculation Criteria:
- Eliminate noise as much as possible + create a curved shape with user input angle
- Fast respond to abrupt change
- Coding requirements: easy and not much space

Sampling Frequency: 100Hz

University of Massachusetts Amherst BE REVOLUTIONARY

Signal Filtering - Total Variation Filter



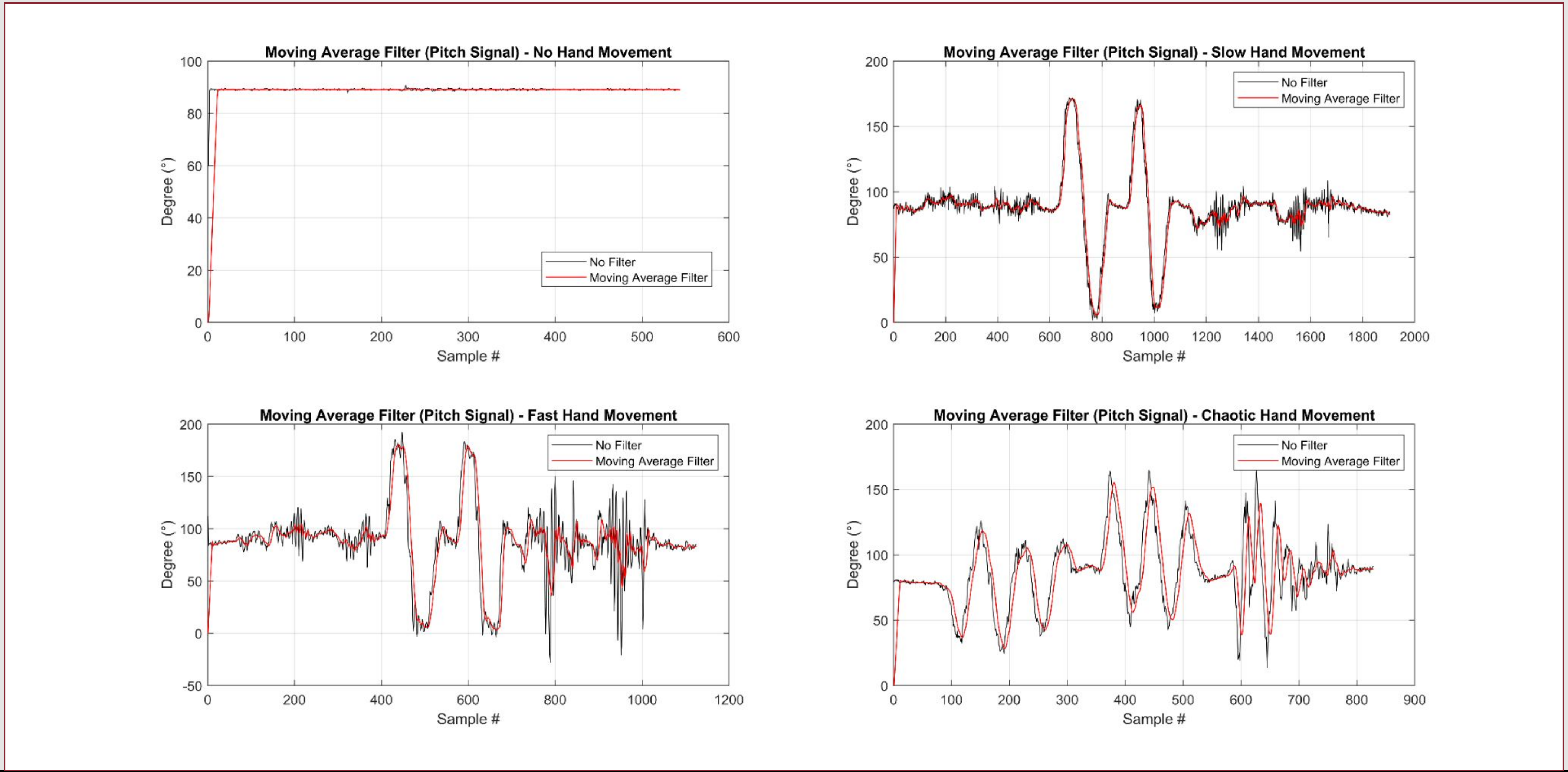
Optimization Problem:

- Solving the Cost Function J(x) using Iterative clipping algorithm

$$J(\mathbf{x}) = \|\mathbf{y} - \mathbf{x}\|_2^2 + \lambda \|\mathbf{D}\mathbf{x}\|_1.$$

Note: ‘Dx’ measures how much the signal changes between values (Total Variation).
Reference for clipping solution:
https://eeweb.engineering.nyu.edu/iselesni/lecture_notes/TV_filtering.pdf

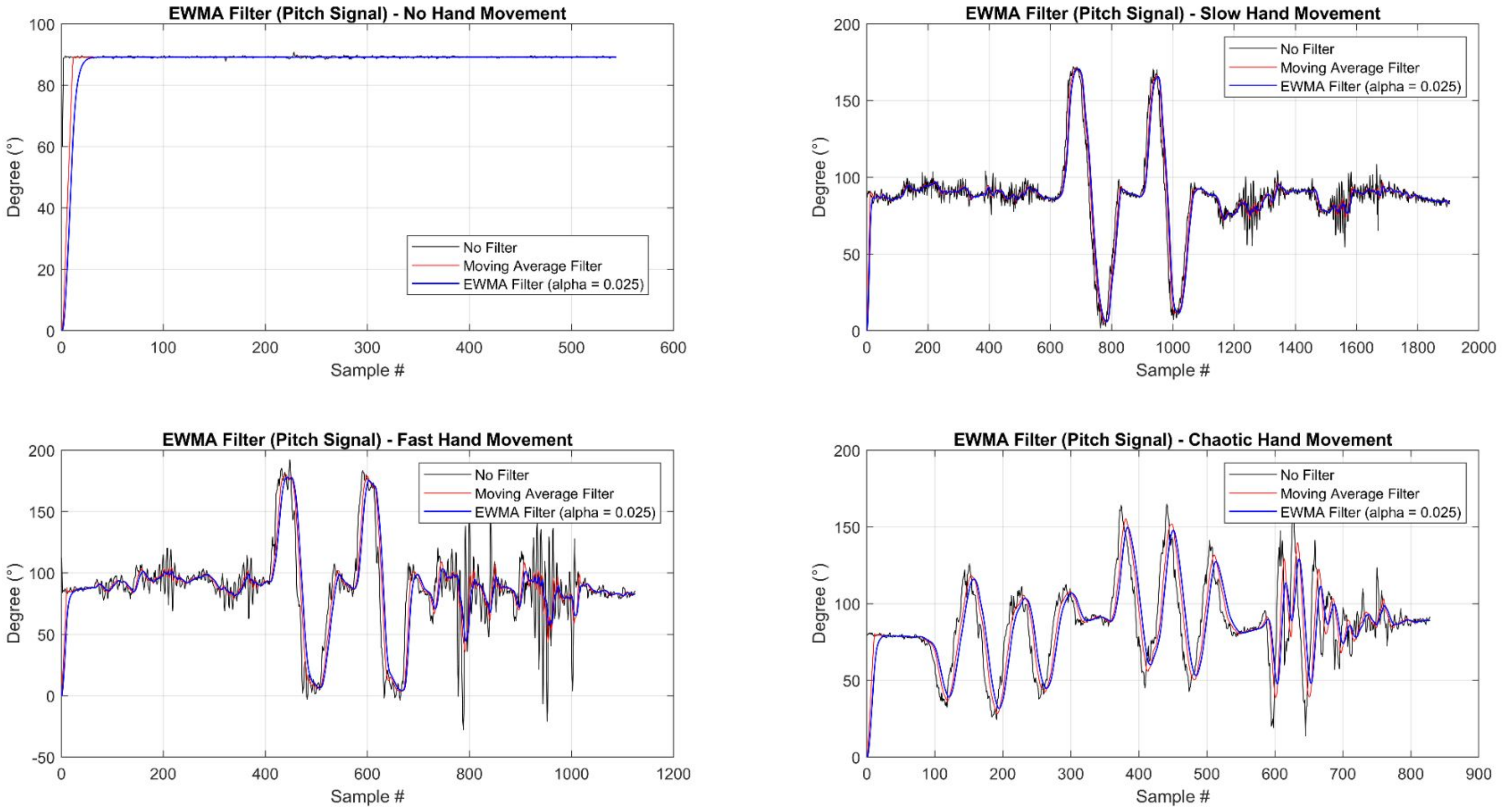
Signal Filtering - Moving Average Filter



Window Size: 10 samples

Note: The window size matters because it applies a latency that can significantly change our system perception of turning angles.

Signal Filtering - Exponentially Weighted Moving Average (EWMA) Filter



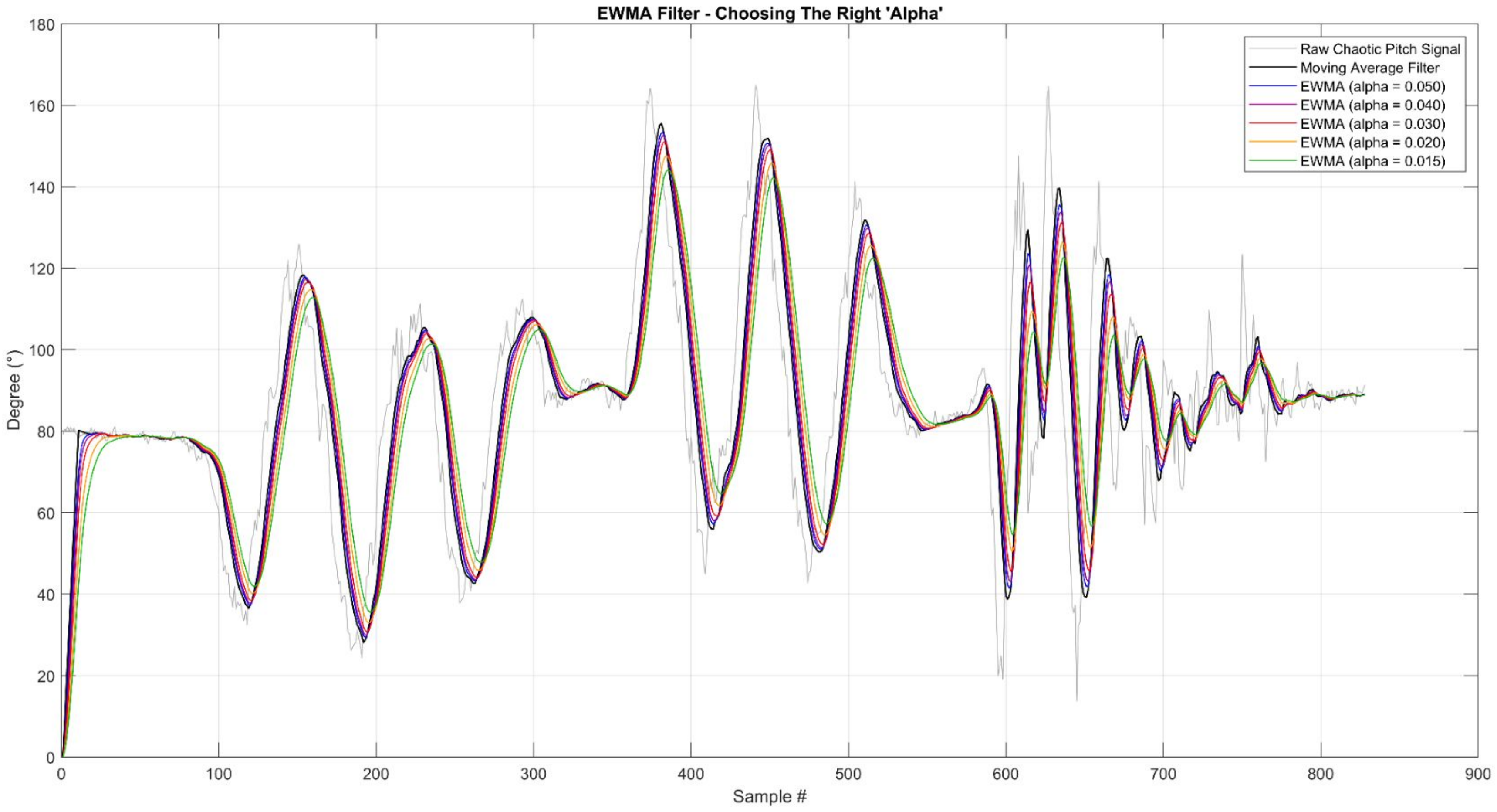
University of Massachusetts Amherst BE REVOLUTIONARY

EWMA Equation

$$S_t = \alpha\left[Y_t + (1-\alpha)Y_{t-1} + (1-\alpha)^2 Y_{t-2} + \cdots \right.$$
$$\left. \cdots + (1-\alpha)^k Y_{t-k}\right] + (1-\alpha)^{k+1} S_{t-(k+1)}$$

Parameters (for window size ‘k’):
Y: is the value at time ‘t’ (our measurement)
S: is the value of EWA at time ‘t’
alpha: weight value (we set it to 0.025)

Signal Filtering - Exponentially Weighted Moving Average (EWMA) Filter



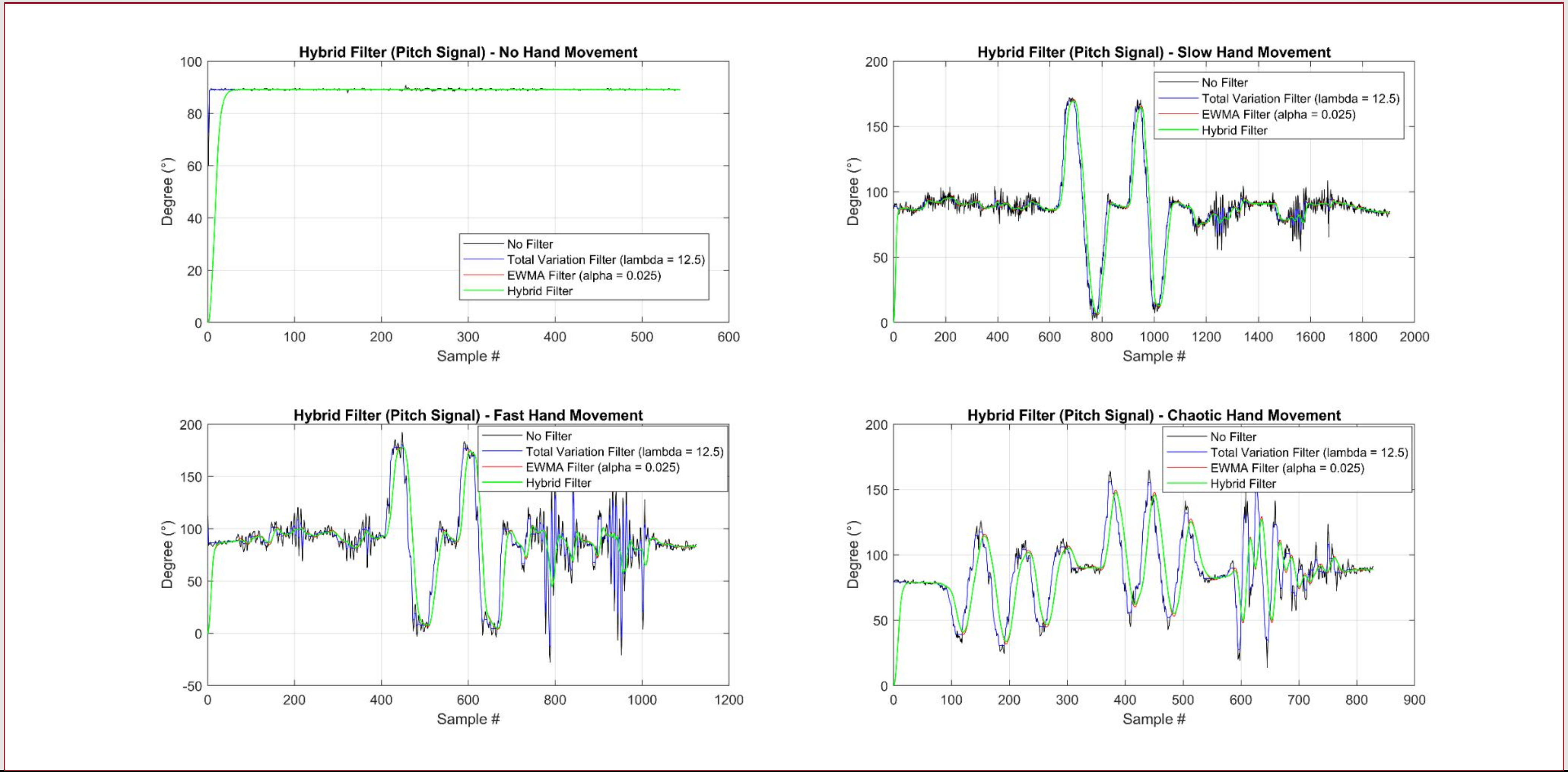
University of Massachusetts Amherst BE REVOLUTIONARY

EWMA Equation

$$S_t = \alpha\left[Y_t + (1-\alpha)Y_{t-1} + (1-\alpha)^2 Y_{t-2} + \cdots + (1-\alpha)^k Y_{t-k}\right] + (1-\alpha)^{k+1} S_{t-(k+1)}$$

Parameters (for window size ‘k’):
Y: is the value at time ‘t’ (our measurement)
S: is the value of EWA at time ‘t’
alpha: weight value (we set it to 0.025)

Signal Filtering - Hybrid Filter (not implemented)



Note: Hybrid filter refers to using an EWMA filter on the output of a total variation filter. One can see that the hybrid filter (green) just barely outperforms the EWMA filter (red) in regions with sensor jitter. More comparison plots in Appendix A.

Signal Filtering - Comparison

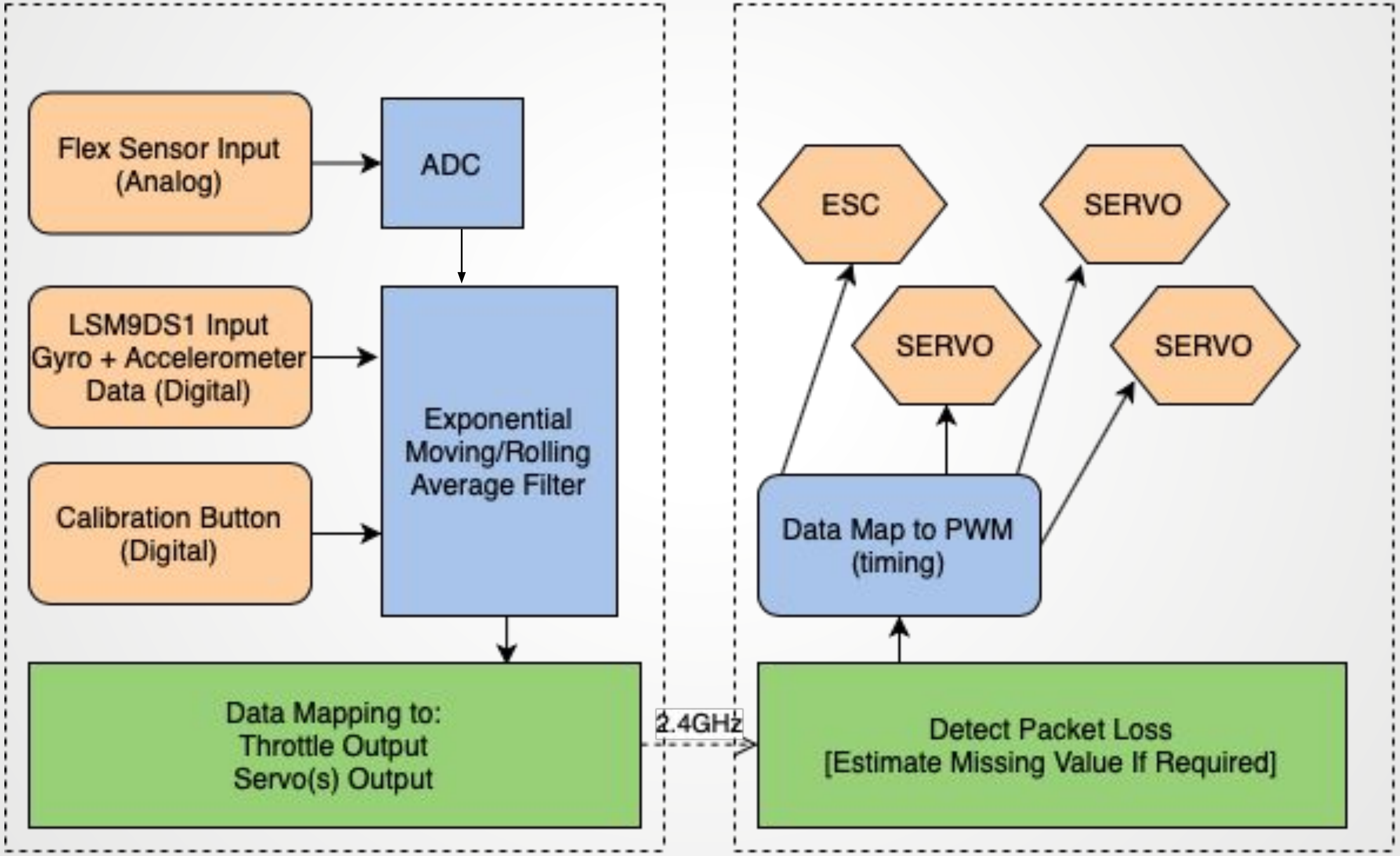
- **Total Variation**
  - extremely good at preserving & restoring flat signal regions → good for sensor jitter
  - effectively adds **zero** latency to signal
  - highly configurable
  - clips noise, but does a poor job of removing it
  - complicated to understand & code
- **Moving Average**
  - easy to understand & code
  - does a good job of filtering out noise
  - minimally configurable
  - requires larger window size to filter out some sensor jitter → higher latency
- **EWMA**
  - does a good job of filtering out noise
  - slightly better than moving average filter at handling sensor jitter for a given window size
  - moderately configurable
  - complicated to understand & code

We ended up using EWMA!

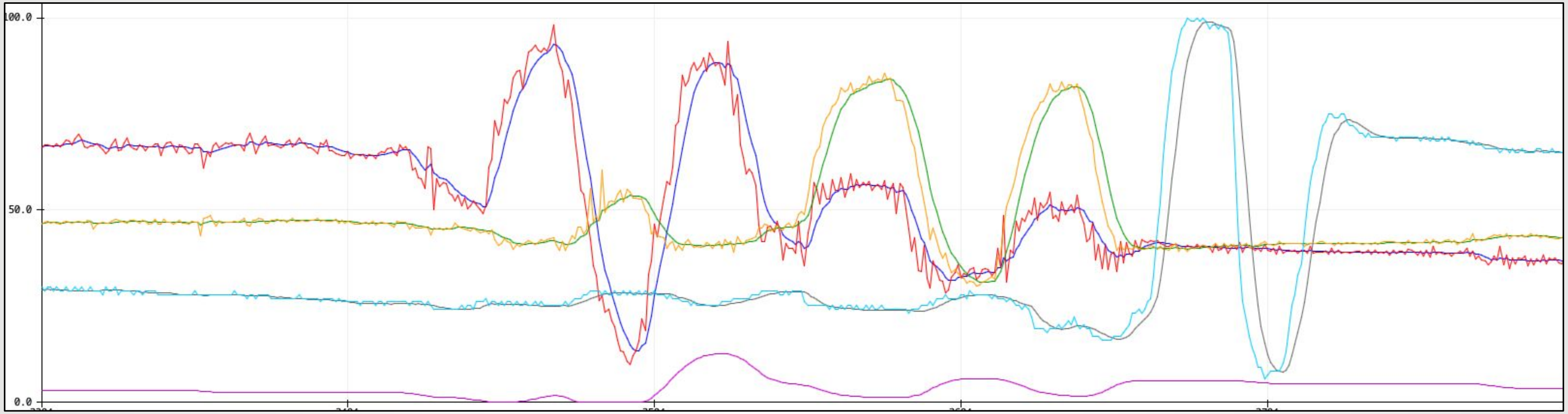
University of Massachusetts Amherst BE REVOLUTIONARY

Note: Sensor jitter differs from noise, in that sensor jitter is characterized by lower frequency oscillations -- which decay over time -- that are usually caused by mechanical properties of the sensor (eg. signal bouncing).

Software Block Diagram



Filter [In System] - EWMA



ROLL Filtered/Raw

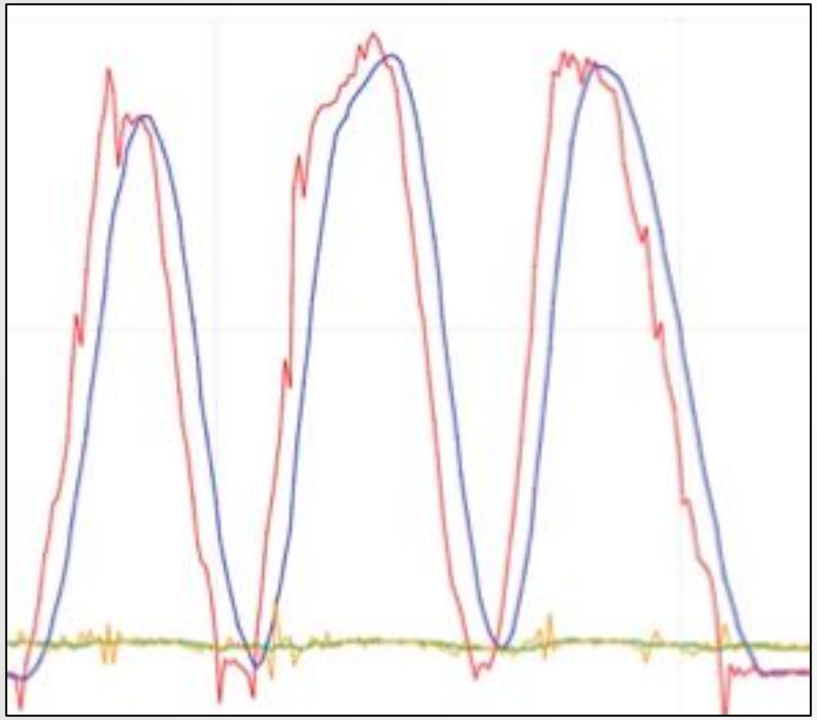
PITCH Filtered/Raw

Stress Filtered/Raw

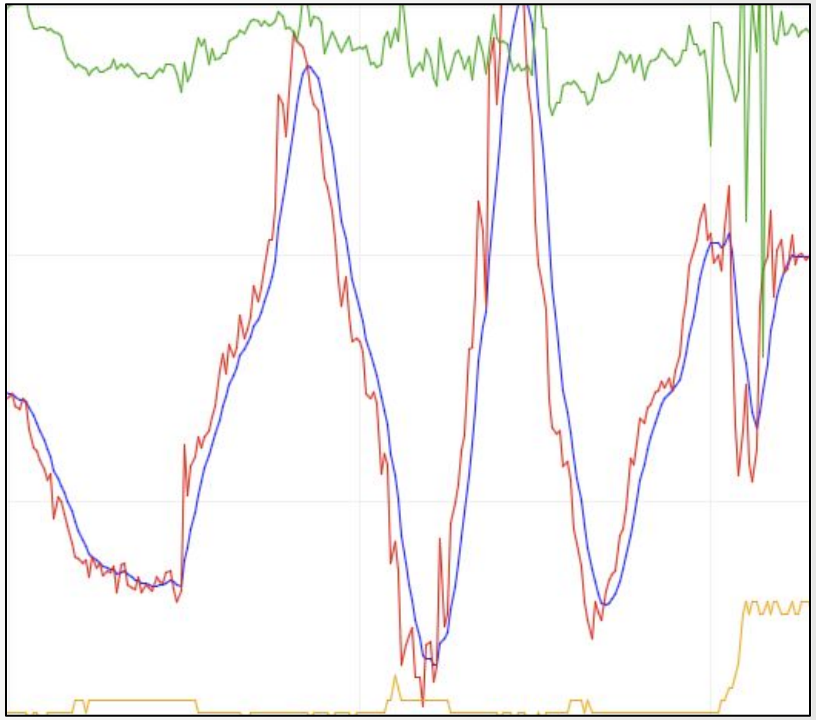
YAW Filtered ← Gyro Integrated after EWMA

Filter [in system] - EWMA

Rolling average used in MDR



EWMA in final prototype



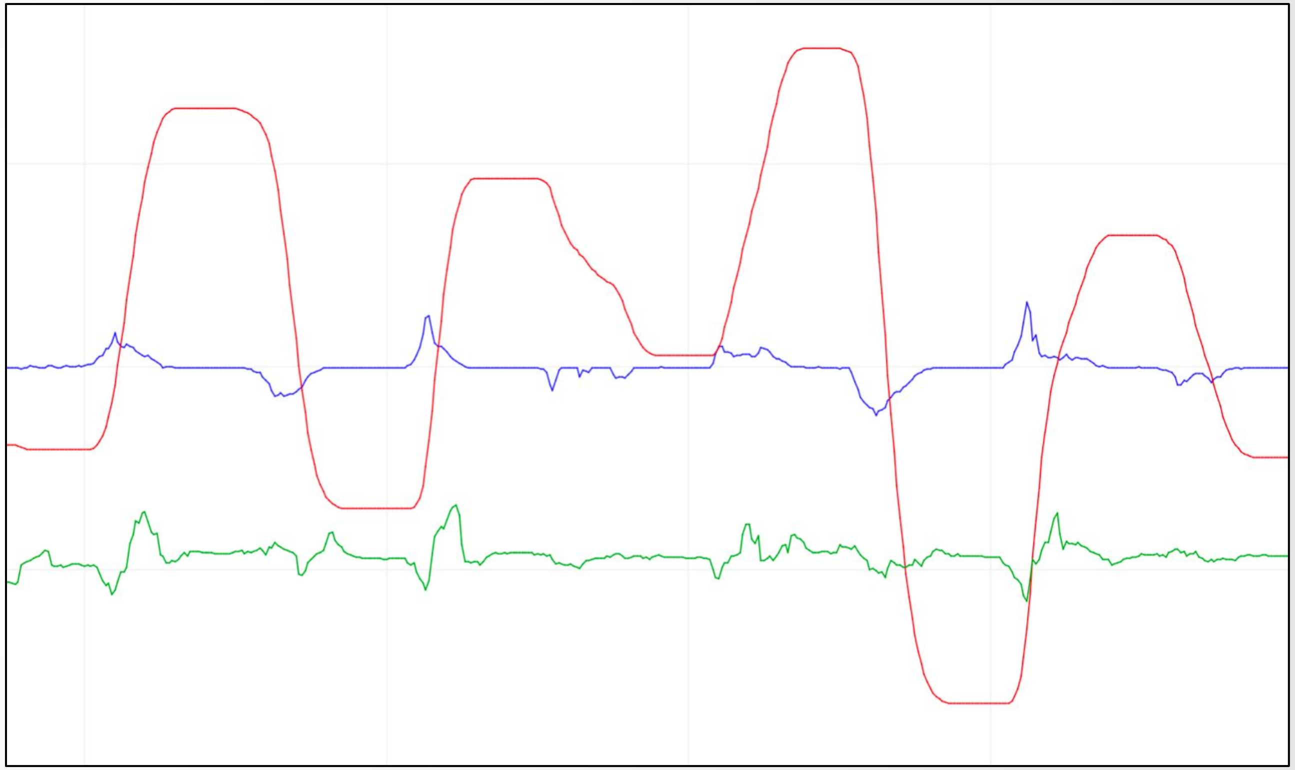
Yaw Integration

Current Rotation [Angle]

EWMA Filtered Gyro Data

Raw Gyro Data

Amplitude is irrelevant, scaling happens on the receiving side (for control mapping)



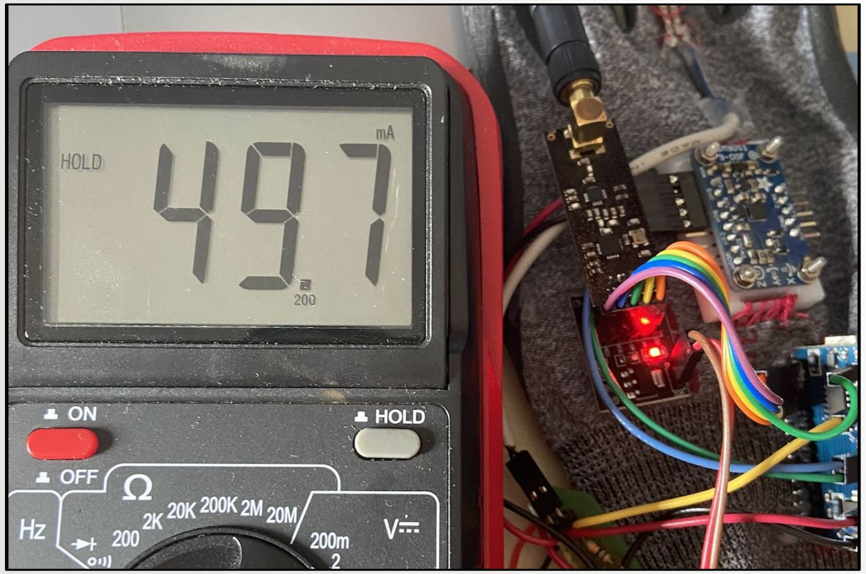
Power Consumption

Glove

4x 1.5V @ 450mAh = 6V @ 450mAh

Typically 80% effective capacity

80% * 450mAh / 49.8mA = ~ 7 hours battery life



Plane

2x 3.7V @300mAh = 7.4V @ 300mAh [7.4V nominal]

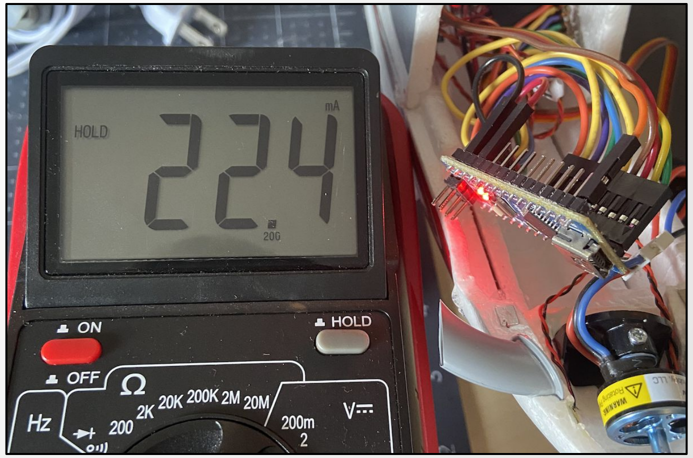
Typically 80% effective capacity

Comparable 3400kV motors run 7.4V @ 3A continuous

Moving servos + electronics only use 22.4mA

Estimating the motor+ESC

80% * 300mAh / 3022mA = ~ 4.8 minutes flight time



List of Hardware/Software (To Date)

Hardware

- Arduino Nano (x2)
- NRF24L01+PA+LNA RF Module (x2)
- LSM9DS1 IMU Board
- 3” Flex Sensor
- LED (placeholder for engine)
- 200 ohm resistor (for LED)
- Nitrile ESD Glove
- Breadboard Power Supply Module
- JST-XH connectors + Pin Headers
- E-flite RC Airplane UMX Turbo Timber
- Voltage Regulator
- Resistors + capacitors (THT + SMD)

Software

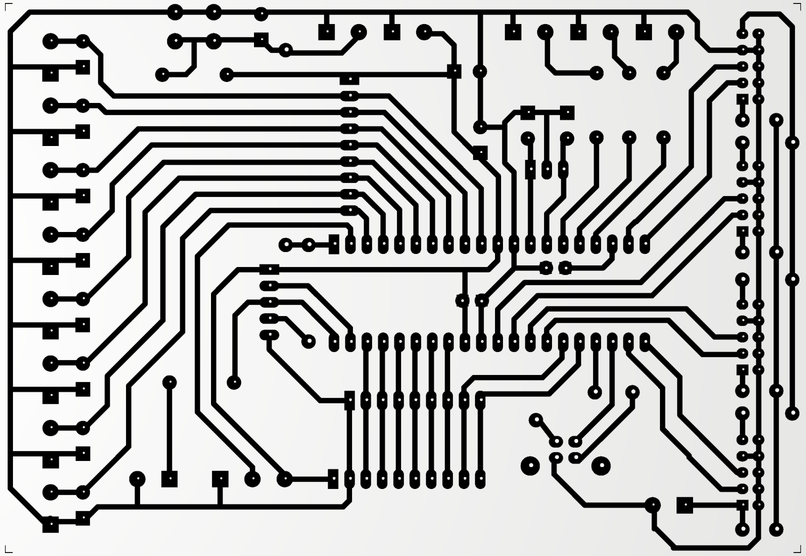
- Matlab: Filtering code, and sample analysis [current]
- Arduino Nano : final prototype code, complete system integration [current]

Rough Cost Estimate (To Date)

- MDR expenses: $66.19
- Demo aircraft: $139.99
- LiPo Battery (2-pack): $17.68
- LiPo Balance Charger: $13.99
- 1st Iteration PCBs: $65.80*
- 2nd Iteration PCBs: $75.30*
- Connector/Crimping Kit: $36.99
- FDR expenses: $104.06

Total: ~ $520

Over budget : $20



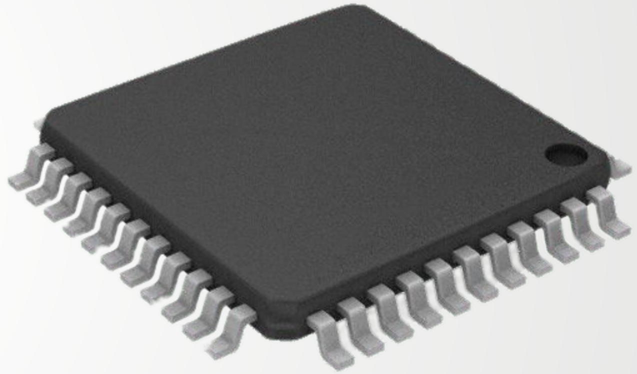
** Each iteration includes 5x TX PCBs, 5x RX PCBs, 1x TX Stencil, 1x RX Stencil

FPR Cost

The cost includes our 2nd PCB iteration that we had before CDR

- **2nd Iteration PCB : $ 75.3**
- **PCB parts (ATmega, crystal oscillation, etc) : $ 46.09**
- **UBEC : $ 21.99**
- **Mounted Servo : $ 35.98**

Total for our FPR (include the PCB iteration) : $ 179.36



** Each iteration includes 5x TX PCBs, 5x RX PCBs, 1x TX Stencil, 1x RX Stencil

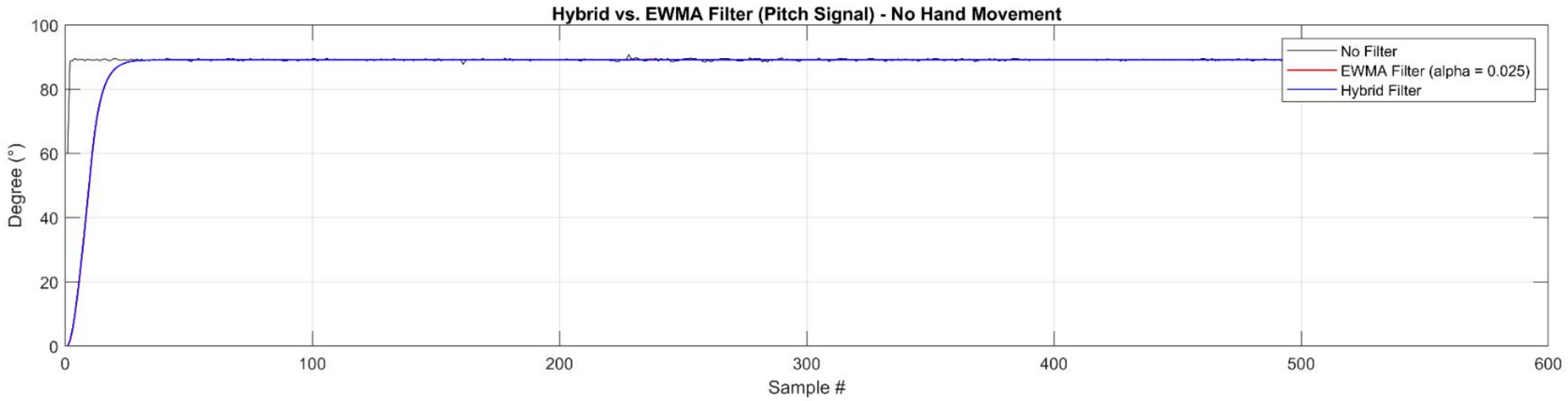
FPR Plan (as intended at CDR)

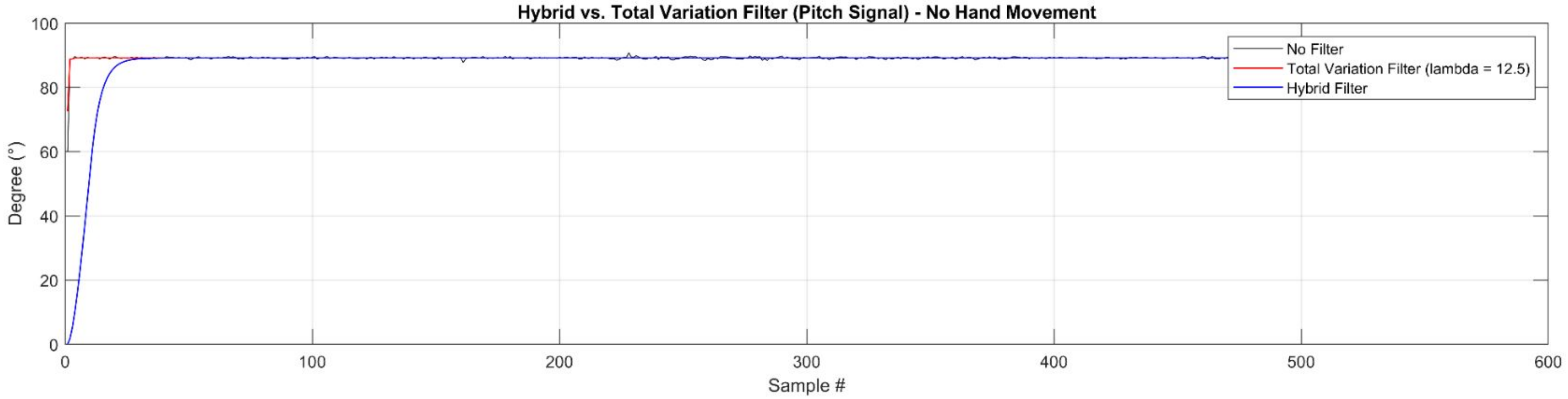
- **Replace Arduinos/breadboards/excess wiring with custom PCBAs**
- **Ensure TX PCBA is powered by 4x AAA batteries**
- **Ensure RX PCBA is powered by LiPo battery**
- **Optimize plane's center of gravity (time permitting)**
  - This can mostly be done with the heavy LiPo positioning
- **Calibrate servo movements on the plane**

Demo

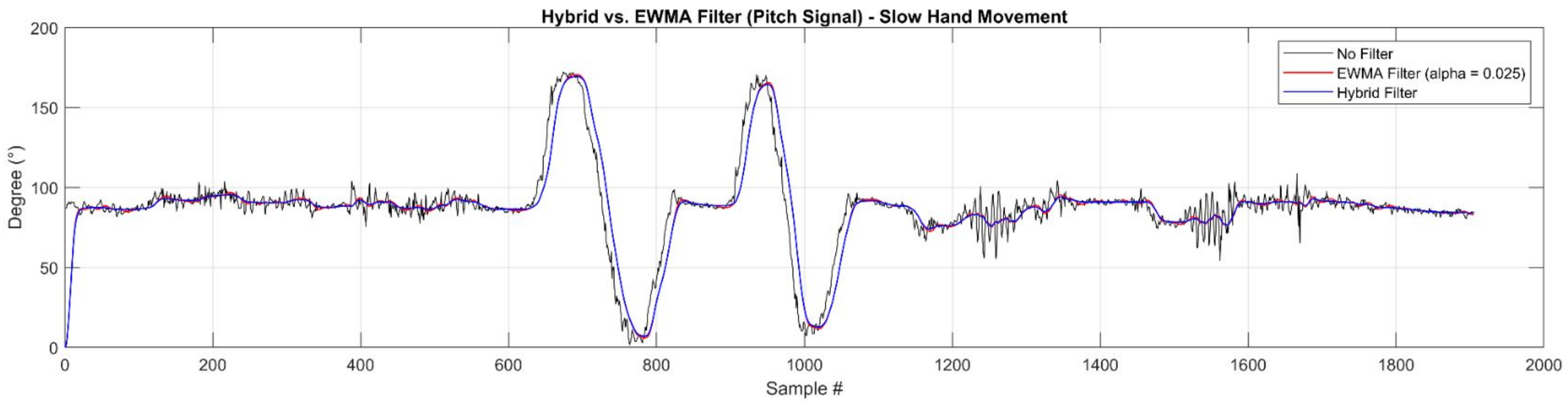
https://drive.google.com/file/d/1q0e19uwaKC0JeYH_FRM7mHS-xi7w3Sao/view?usp=sharing

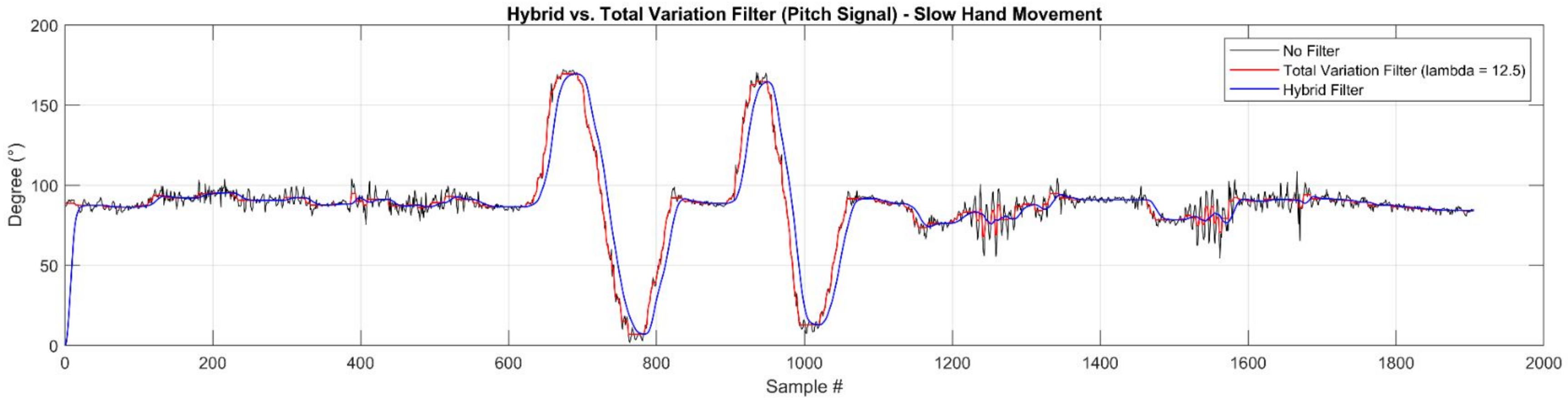
Appendix A - Hybrid Filter (no user input)



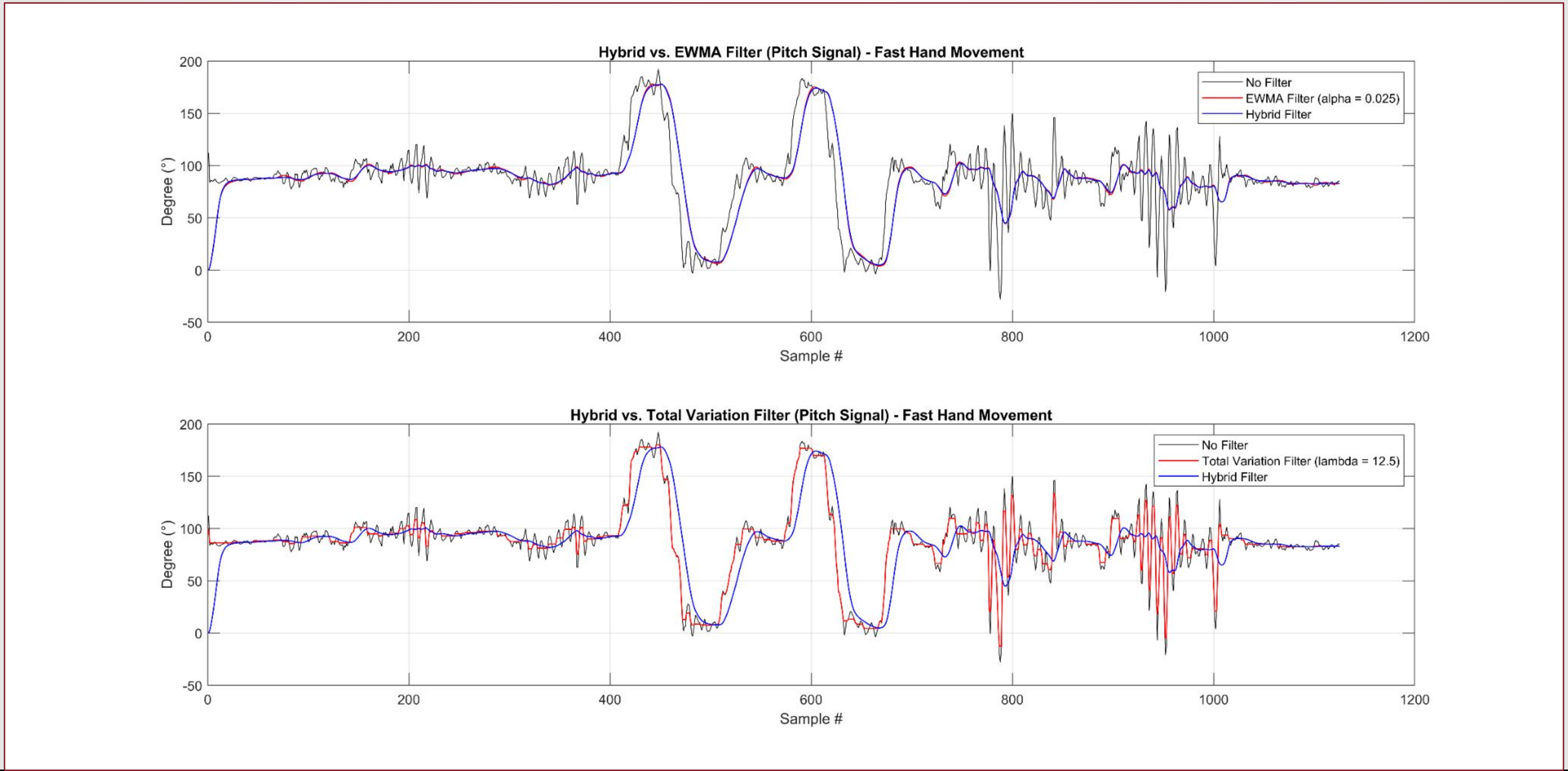


Appendix A - Hybrid Filter (slow user input)





Appendix A - Hybrid Filter (fast user input)



Appendix A - Hybrid Filter (chaotic user input)