ME 410 – Week 2

Christopher Luey

(a) Text Description (400 words max)

This week, we implemented a complementary filter to improve pitch and roll estimation using both accelerometer and gyroscope data from a BMI088 IMU. We added this logic to `update_filter()` using a weighted sum: low-pass filtered accelerometer angles and high-pass filtered integrated gyro rates. The filter weights were tuned with `A = 0.02`, as recommended. We verified smooth tracking and reduced drift, especially during brief rotations.

We also integrated joystick control via shared memory using the `udp_rx.cpp` interface. We defined a `Joystick` struct and mapped shared memory to read real-time input. Within the main loop, we checked for safety triggers: button "B" to terminate the program, excessive gyro rates (>300°/s), pitch or roll angles exceeding ±45°, and communication timeouts (>0.35 s). These were encapsulated in a `safety_check()` function and controlled program flow using a `run_program` flag.

For calibration, we refined `calibrate_imu()` by resetting accumulator variables before averaging 1000 samples. We moved pitch and roll computation inside `read_imu()` to streamline processing. We also resolved two's complement errors from last week by validating sign conversion on raw 16-bit IMU readings.

For Milestone 1, we printed six continuous outputs: complementary filter output (pitch/roll), gyro-integrated pitch/roll, and accelerometer-derived pitch/roll. We gently rotated the quadcopter and saved console output to a file for plotting in Excel. These plots confirmed the filter's ability to fuse high-frequency gyro and low-frequency accelerometer data into stable orientation estimates.

(b) Task Assessment

1. **What went well:**
    - Complementary filter implementation showed smooth, accurate attitude tracking
    - Joystick integration and shared memory access worked seamlessly.
    - All safety checks were triggered and logged correctly.

2. **What did not go well, why:**
   Gyroscope initially showed no drift, which was misleading; further testing revealed small integration errors.
    - Small bugs in time_elapsed tracking and incorrect filter sign caused unexpected drift in orientation estimates.
    - Initial complementary filter output was unstable until calibration offsets and filter constants were properly tuned.

    - TX/RX communication between joystick and Pi occasionally dropped, causing unexpected joystick timeouts.
    - Applying calibration offsets led to changes in pitch/roll magnitude, requiring re-validation of the angle computation.

3. **What will you change for next class:**
    - Modularize data logging and plotting into a scripts
    - Improve robustness by checking all IMU reads for I2C communication errors.
    - Introduce constants or config header to easily tune filter parameters.

(c) Team Member Effort Report
Jason – 50%:

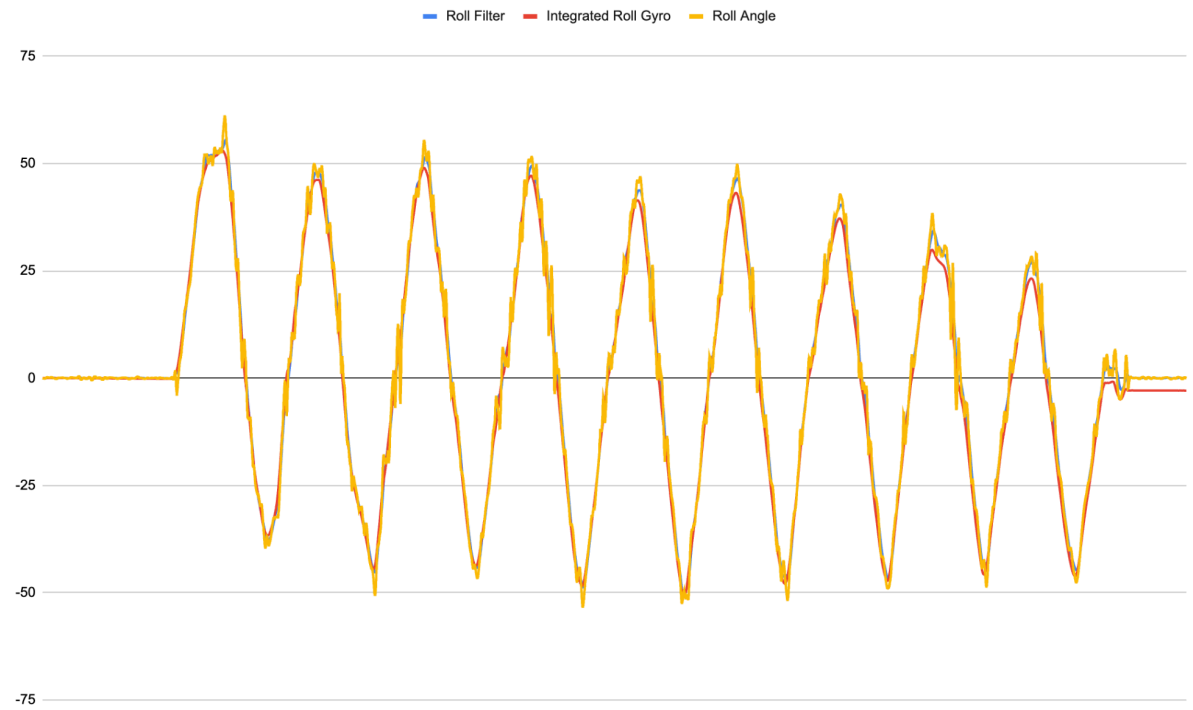
- Integrated joystick control and shared memory access
- Developed and tested safety checks, including sequence timeout

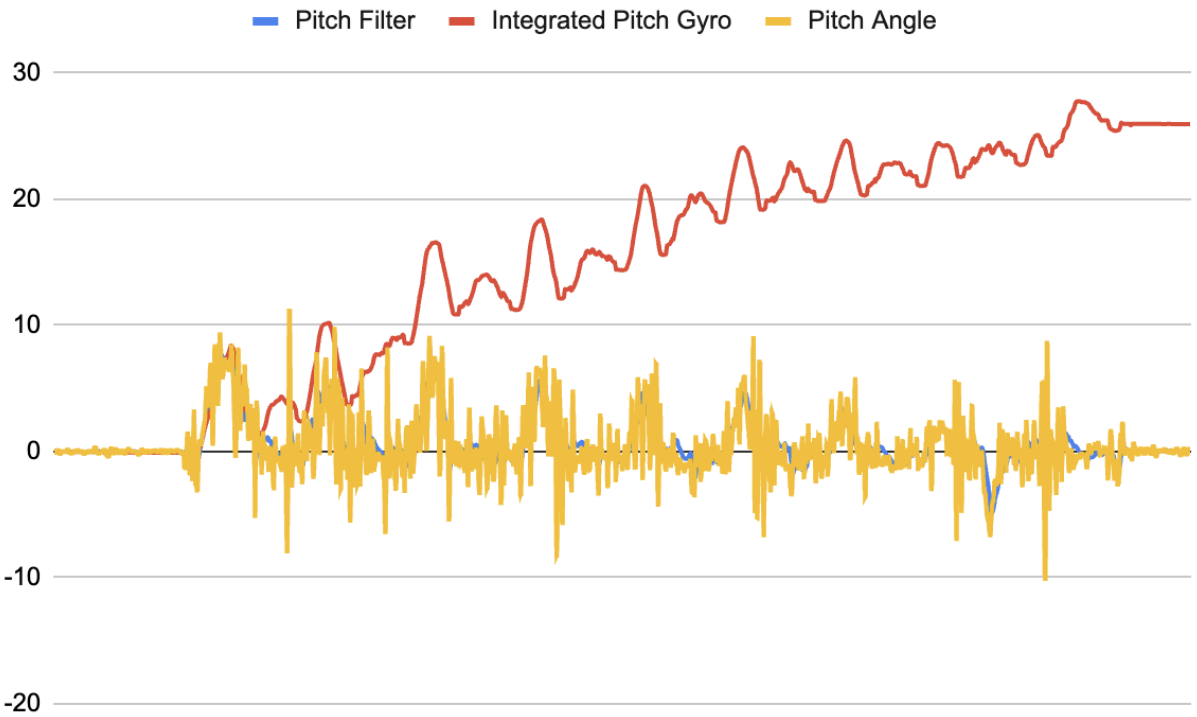
Christopher – 50%:

- Implemented and tuned complementary filter
- Created Excel plots for Milestone 1
- Debugged and corrected IMU reading logic and calibration

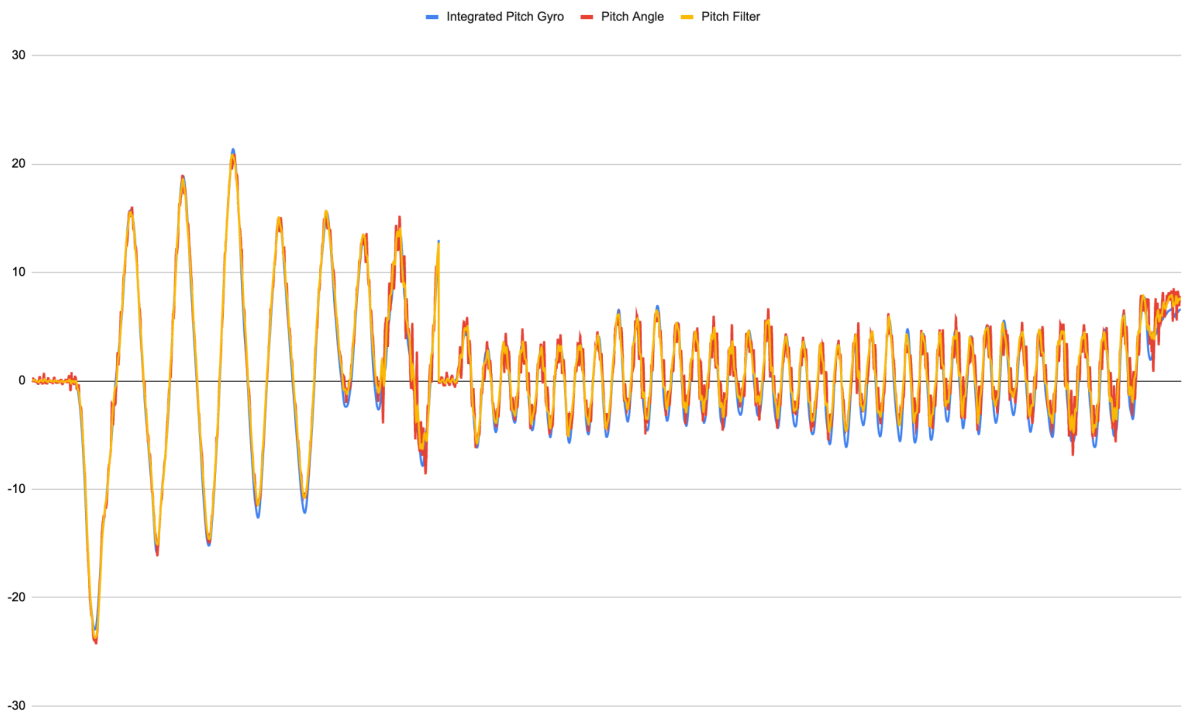
Both members contributed equally and iteratively tested on hardware.

Roll Test


Roll Filter
Integrated Roll Gyro
Roll Angle
75
50
25
0
-25
-50
-75


Pitch Filter
Integrated Pitch Gyro
Pitch Angle
30
20
10
0
-10
-20

Pitch Test


Integrated Pitch Gyro
Pitch Angle
Pitch Filter
30
20
10
0
-10
-20
-30