Determining Angle from Accelerometer and Gyro Data Using a Complimentary Filter

Objectives:

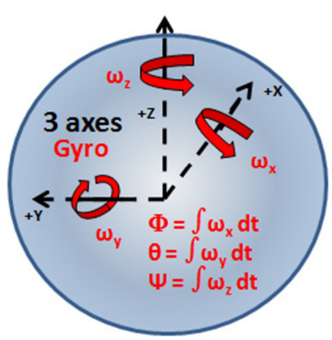
- Compute the board angle using gyroscope data
- Compute the angle using accelerometer data
- Create a "Complimentary Filter" utilizing weighted input from both sensors
- Apply the complimentary filter tool to a position controller

Background:

In previous labs you have used data from the gyroscope to track the rotation of the MinSeg board. However, you may have noticed that the reading could "drift" even while the board was lying flat on a table. In this lab, we will explore the combined use of multiple sensors in order to fight "drift" and add accuracy in measuring additional Degrees of Freedom (DOF).

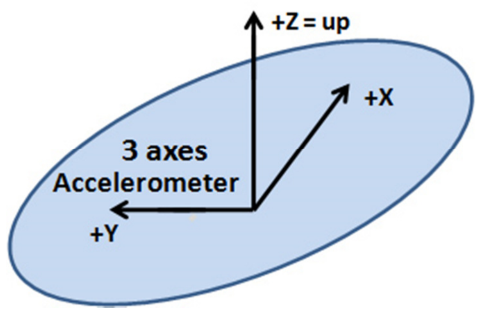
Gyroscopes

Gyroscopes utilize the principle of angular momentum to accurately measure rotation about an axis. Thus, multiple gyroscopes can be used to measure the angular rotation of an object in space. However, gyroscopes cannot inherently tell which was is "up" in the world, and techniques for computing position may result in "drift".



Accelerometers

Accelerometers measure the stresses caused by the movement of masses during linear acceleration or deceleration. Utilizing an array of these elements in three axes, we can effectively track linear movement in a 3D space. Because of the constant downward acceleration of Earth's gravitational field, accelerometers can also generally discern which was is "down" in the world. They cannot, however, accurately measure rotation.



Using both sensors in one application can leverage the strengths of both sensors, while eliminating the drawbacks. One real world example of the use both sensors is the Nintendo® Wii Remote. After its launch in 2006, many users observed that the "WiiMote" was great for sensing up waggle, but couldn't accurately track more complex motions including rotation. This was because the WiiMote only relied on a 3-axis accelerometer for motion tracking. In 2009, Nintendo released a "Wii Remote Plus" add-on that finally enabled "True 1:1 motion tracking" in 6 DOF. The core of this add on was a gyroscopic sensor.

TL;DR: Accelerometers are great for measuring linear movement, and Gyroscopes for rotational movement. Why not both?

Part 1: Computing Position Angle with Gyroscope Data

First, create a new program for angular measurement using the gyroscopic data.

Create the Simulink diagram below.

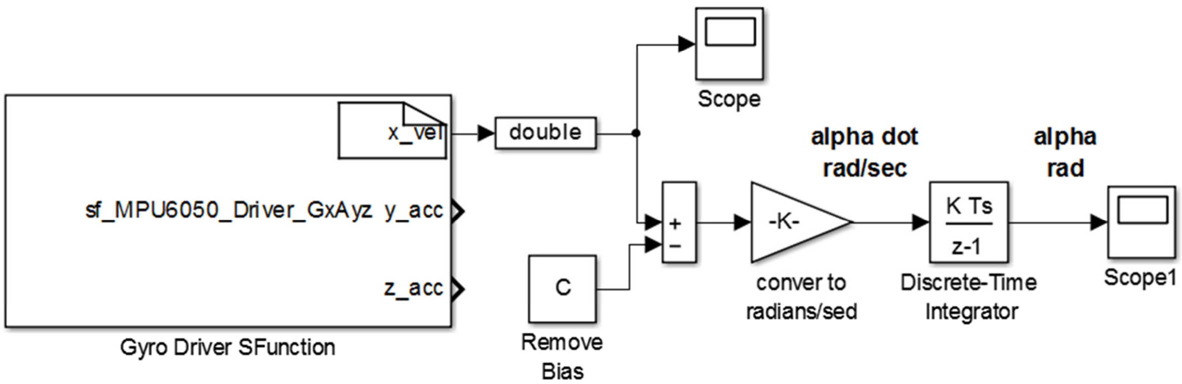


Figure 1: Angle output using gyroscope data

Above, you can see that the input from the Gyro Driver is being integrated in order to produce the final Scope1 output. This also means that if the hardware does not read exactly zero while at rest, the small values will cause a steady increase or decrease in the reading. This is known as “Gyro drift”. To combat this problem, a Bias term -C- is removed before integration. You must manually “calibrate” the system by experimentally finding a value to remove this drift.

Checkpoint 1: Demonstrate a functioning system that accounts for drift. Rotate the board 90° around the x(??) axis, and observe the result in Scope1.

Part 2: Computing Board Angle with Accelerometer Data

In this exercise, we will try computing the board angle utilizing the accelerometer data. Because the board can sense what is "up" (or rather down), the accelerometer measurement essentially has a way to compare error against the real world.

Data from 2 acceleometers (Yacc and Z_{acc}) are used to calculate the angular position. The downward acceleration from the earth will always be fairly constant: if the board measures a Z-component of 9.81m/s^2 but nothing in the Y-component, it knows it is lying flat. If the Z-acceleration decreases, but the Y accelerometer measures some portion of this downward, we can calculate the angle using an arctangent block.

Create the Simulink diagram below.

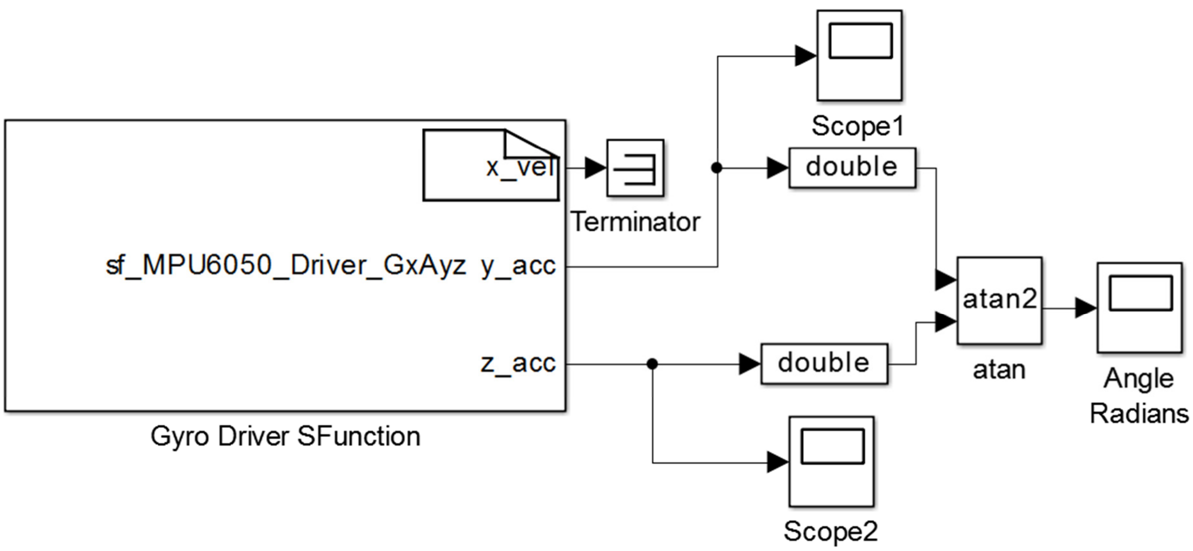


Figure 2: Components of a complimentary filter (detailed view)

Checkpoint 2: Run the system. Rotate the board 90°, and observe the result in the scope.

Discussion Questions:

- Do the measurements seem valid for all angles through a rotation?
- How does the noise of the signal compare to that of the gyroscope function?
- If the accelerometers are being used to track the known gravitational acceleration, what happens to measurements when the board is accelerated in a linear direction?

Part 3: Creating the Complimentary Filter

Now, the previous two measurement circuits can be combined into one measurement system. However, we must weight the input from the two sensors so that we still get a sensical answer: if we simply add two measurements of 80°, would the result be valid? We will thus multiply the sensor readings by fractions which add up to one; this is why our circuit is called a Complimentary Filter.

Create the Simulink diagrams below.

Note: The “GoTo” blocks act just like wire, but transport data from point to point in order to maintain some sanity in the diagrams.

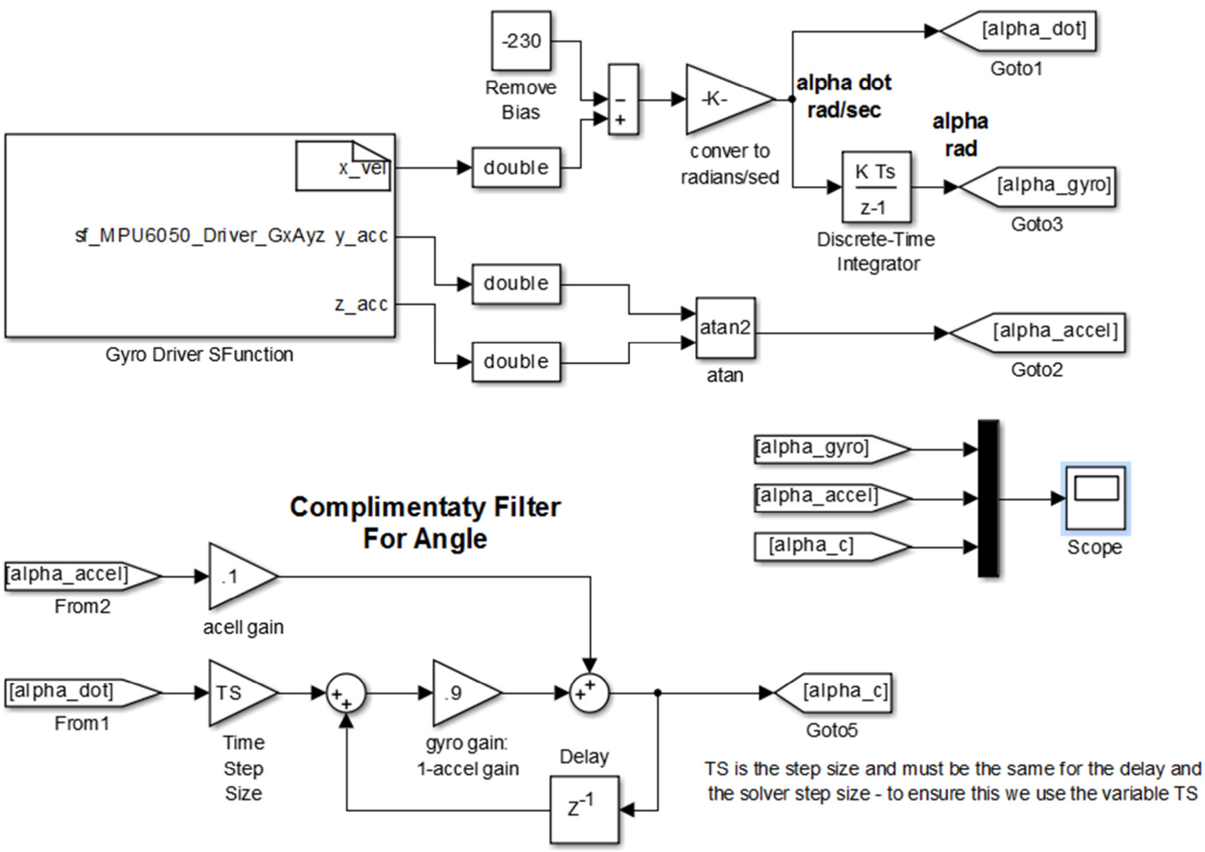


Figure 3: Components of a complimentary filter (detailed view)

You may have noticed that the accelerometer data was somewhat noisy or sensitive to other forces. Because of this, we use a much smaller percentage of the accellerometer data than the gyroscope. Also, the gyroscope only needs a small correction to compensate for drift accumulation.

Now, select the components of the detailed diagram above, then right click and create "Create subsystem from selection". This will create a block that can be easily inserted into code for other applications.

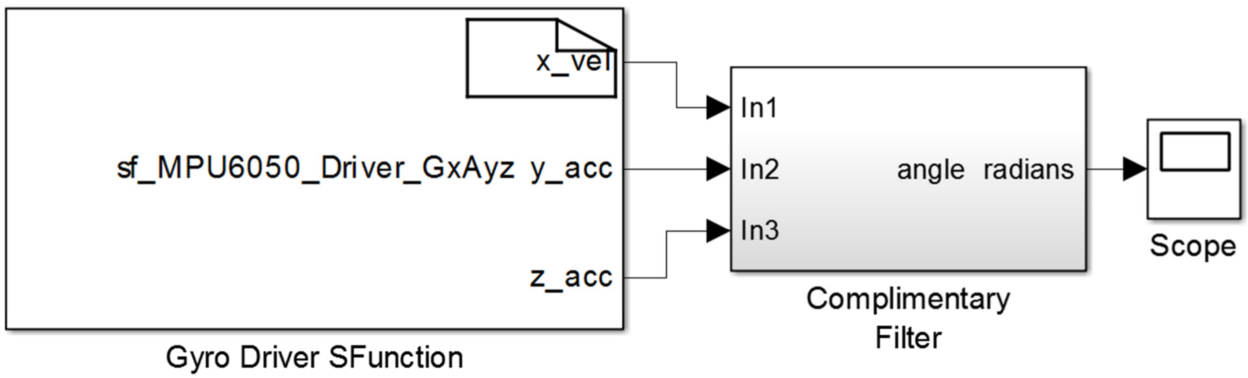


Figure 4: Complimentary filter with subsystem block

Checkpoint 3: Run the system. Rotate the board in several directions, and observe the output.

Discussion Questions:

- What happens to the readings as you adjust the accelerometer and gyroscope gains? What is the system like if their values are reversed (0.9 and 0.1 resp.)?

Part 4: Using the Complimentary Filter for Position Control

Now, we will revisit a position control diagram and apply the Complimentary Filter you have just created. In this case, we will be measuring position, and then outputting this data to a PID controller and motor driver. What do you think the resulting system will accomplish?

Create the Simulink diagram for Closed Loop Position Control below:

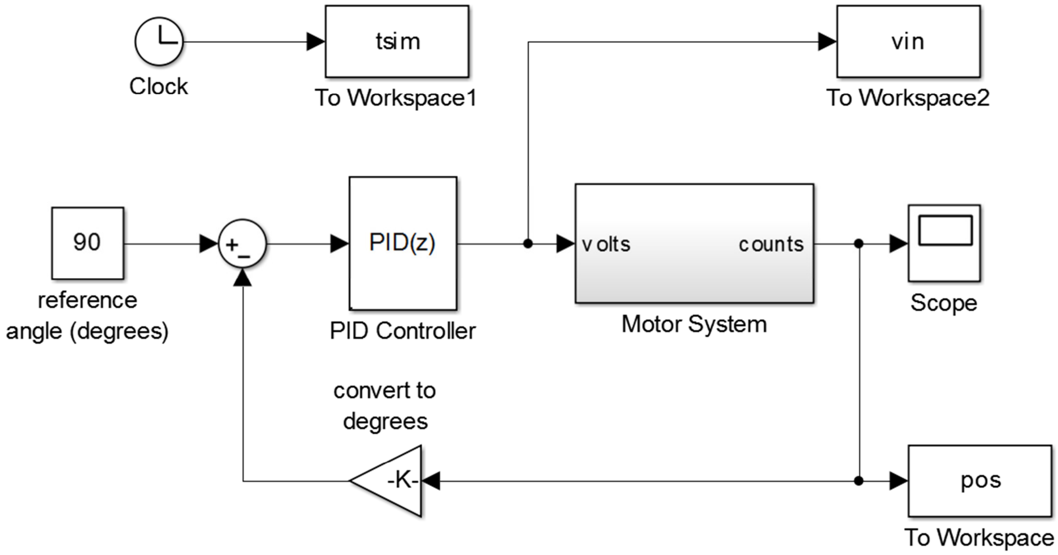


Figure 5: Closed Loop Position Control Diagram

Be sure to choose the correct feedback value K for converting from encoder counts to degrees.

Now, insert the complimentary filter you have created to complete the system below.

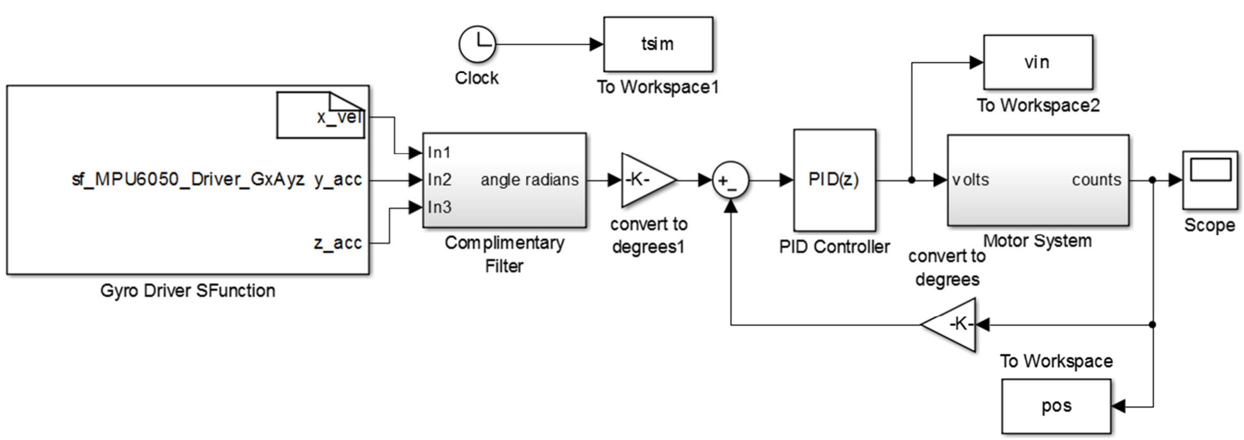


Figure 6: Final System. Closed Loop Position Control utilizing a Complimentary Filter

For this exercise, Proportional control will be sufficient. Double click the PID controller block, and enter a P value of 0.1. The (I) and (D) terms remain 0.

Checkpoint 4: Run the system. Rotate the board in several directions, and observe the resulting wheel behavior. Try sticking a post-it note or other marker on a wheel and test the system.

Discussion Questions:

- You can try modifying the Accel/Gyro ratio to observe the difference in behavior.
- This is also an excellent system to test out PID control. Note the system behavior as you change input values P, I, or D in the controller block.

References:

MIT site: http://web.mit.edu/scolton/www/filter.pdf

https://community.freescale.com/community/the-embedded-beat/blog/2012/03/29/degrees-of-freedom-vs-axes

Ars Technica: http://arstechnica.com/gaming/2008/08/wii-motion-sensor/