**Lab 5: Reading Data from a Gyroscope/Accelerometer**

650:361 Introduction to Mechatronics

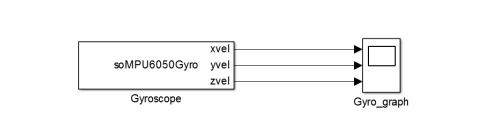
Team Members: Shivani Topiwala, Nancy Contreras, and Pamela Pajarillo

**Introduction**

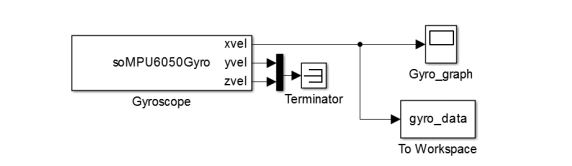
For this lab, we will use MATLAB with Simulink and a gyroscope. A gyroscope is a device that measures angular velocity. Our goal is to analyze the data produced by the gyroscope and determine the conversion constant used to determine the angle.

**Part 1: Reading the Gyroscope**

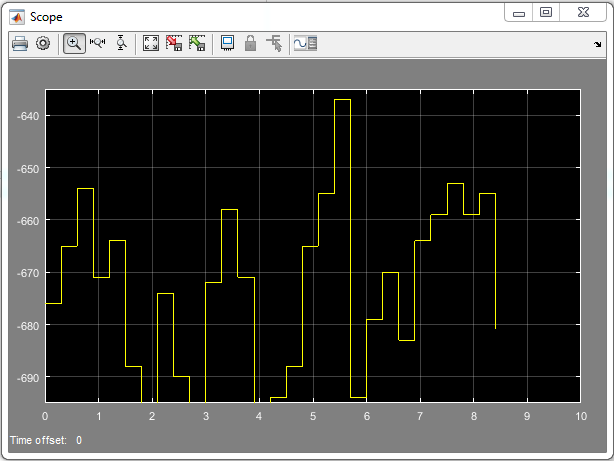
In Simulink, following the diagram provided, we were able to see that there are three outputs, which were the x velocity, the y velocity, and the z velocity. We connected to External Mode, which allows bi-directional communication to and from the Arduino to the computer.

**Figure 1.** Simulink diagram that Produces Three Outputs (x-velocity, y-velocity, z-velocity)

After connecting the Arduino, we changed the fixed step constant from auto to 0.3 and the fixed step time to infinity. On our screen, we can see the three output graphs, which displayed the velocity in the x, y and z direction. We verified that the output was accurate by physically moving the Arduino, which caused the amplitude for the sinusoidal function to increase. Using the Simulink diagram shown in Figure 2, we terminated the y and z output because we are only interested in the velocity in the x direction.

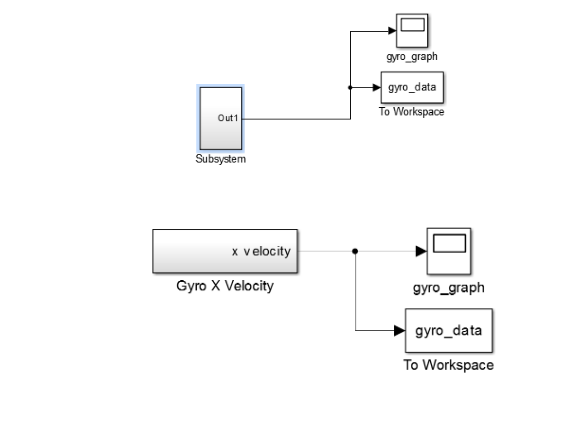


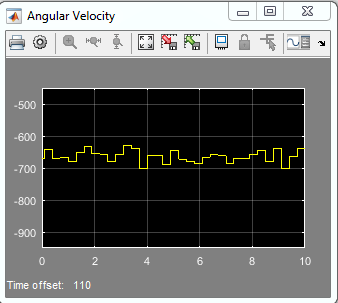
**Figure 2 .** Simulink Diagram to View Velocity in the X Direction and to Collect Data



**Figure 3.** Plot of X-Velocity from Gyro

We created a subsystem and was able to get just one output. We then changed the step size constant to 0.3 and the time to infinity, and also changed the workspace output from ‘gyro\_data’ to ‘array’ and then ran the diagram in Simulink. When we ran ‘gyro\_space’ in MATLAB, we were able to see the x velocity data over a period of time.

**Figure 4.** Simulink Diagram with a Subsystem, Producing One Output, X Velocity

Q1: If the system is at rest – what is the value of the gyro reading? How much is it fluctuating? Use the data written to the workspace to compute the average gyro reading when the sensor is at rest for several seconds. 

**Figure 5.** Angular Velocity Reading When the System is At Rest

Even though the system is at rest, the value of the gyroscope reading was fluctuating. According the angular velocity graph, the value of the gyroscope was fluctuating around -650. Using Matlab, we were able to find the average of the gyroscope data by using the ‘mean’ command. The average of our data was -654.

val(:,:,1) =

-676

val(:,:,2) =

-665

val(:,:,3) =

-654

val(:,:,4) =

-671

val(:,:,5) =

-664

val(:,:,6) =

-688

val(:,:,7) =

-697

val(:,:,8) =

-674

val(:,:,9) =

-690

val(:,:,10) =

-698

val(:,:,11) =

-672

val(:,:,12) =

-658

val(:,:,13) =

-671

val(:,:,14) =

-699

val(:,:,15) =

-694

val(:,:,16) =

-688

val(:,:,17) =

-665

val(:,:,18) =

-655

val(:,:,19) =

-637

val(:,:,20) =

-694

val(:,:,21) =

-679

val(:,:,22) =

-670

val(:,:,23) =

-683

val(:,:,24) =

-664

val(:,:,25) =

-659

val(:,:,26) =

-653

val(:,:,27) =

-659

val(:,:,28) =

-655

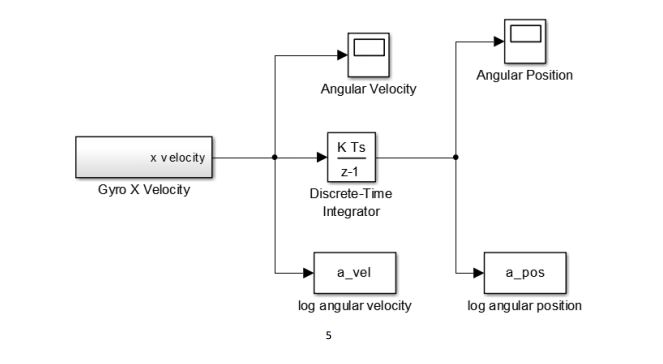
val(:,:,29) =

-681

**Figure 6.** Gyro data collected in MATLAB

**Part 2: Computing Angle from a Gyro**

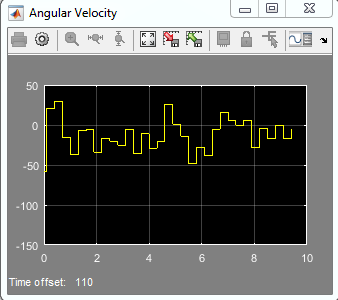
We created the Simulink diagram below in order to to investigate the how the gyroscope provides data and how we can manipulate the data provided to find over values. After creating the diagram, we changed the sample time of the Discrete-time integrator to -1.



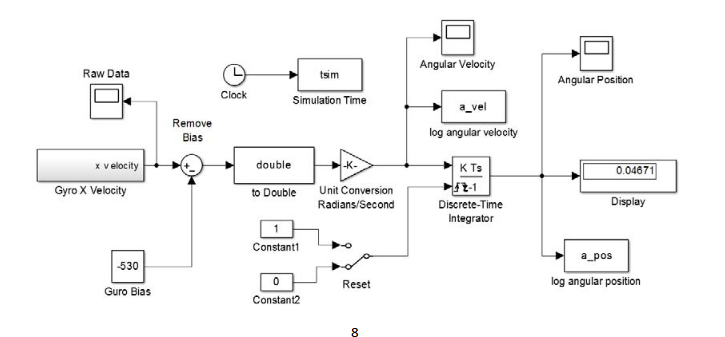
**Figure 7.** Simulink Model to Obtain the Angle By Integrating Velocity

Q2: What do you notice about the angular position and velocity when the sensor is sitting still? Does this make sense?

When we initially ran the Simulink model, the angular would fluctuate. In order to eliminate this, we added a component called ‘remove bias’, which would subtract a constant from the angular velocity function. We used our average velocity value -654, which shifted our angular velocity function down to around zero. The angular velocity would still fluctuate, which did not make sense because the Arduino was not moving. So to further address the issue, we created a new diagram shown below.

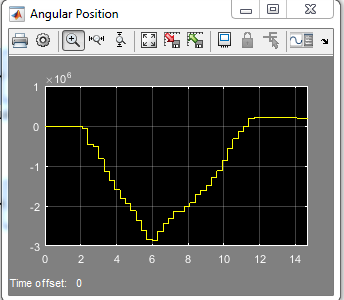
**Figure 8.** Graph after Bias was Removed (Angular Velocity Now Fluctuates Around Zero)

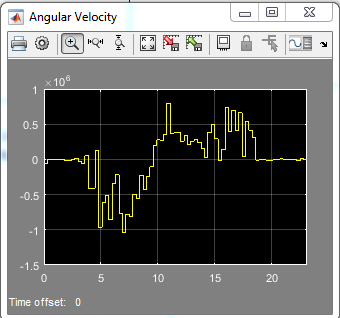
The previous Simulink diagram brought down the fluctuation down to zero, but our goal is for the angular velocity to be completely zero when the Arduino is at rest. In this new model, we introduce a switch that is able to convert the angle in degree per second to radian per second. Our initial conversion constant K was 1, which brought our angular velocity to zero when the Arduino was 90 degrees or normal to the table.



**Figure 9:** Simulink Diagram to Determine Unit Conversion and to Remove Any Fluctuation

Q3: What is the proper unit conversion factor to obtain the results in radians/sec and radians? Create a plot showing both the angular position and angular velocity versus time as the sensor moves from 0 to 90 degrees and back to 0. The units for this plot should be radians, radians/second and seconds. Clearly label this graph.

**Figure 10.** Angular Position vs. Time Graph Sensor Moves from 0 to 90 Degrees and Back to 0

**Figure 11.** Angular Velocity vs. Time When Sensor Moves from 0 to 90 Degrees and Back to 0

The proper unit conversion factor to obtain the results in radians/sec and radians is . In order to calculate the position factor, divide by the position obtained from the plot. To calculate velocity factor divide by time times average velocity. The average velocity was calculated by using the ‘mean’ command in MATLAB.

Calculations

Position Factor =

Velocity Factor =

|  |  |
| --- | --- |
| Position | -5.5 \* 10^5 |
| Time | 6 seconds |
| Average Velocity | -5610 |
| Position Factor | -3.14 \* 10^-5 rad/s |
| Velocity Factor | -4.66 \* 10^-5 rad/s |

**Figure 12.** Data Collected from the Graph and Calculations of Position Factor and Velocity Factor

**Conclusion**

In this lab, we learned how to obtain data from a gyroscope, which measures angular rate. Using MATLAB and Simulink, we analyzed the data produced by the gyroscope and determine the conversion constant to use to determine the angle.