Exercise 1

Q1) What is x0? Why did you choose this value to be x0 (is you device calibrated perfectly)?

$x_0 = -0.0410m$. This is the value of x_h at 0 degrees or the equilibrium position calculated using the sensor reading and the linear equation discovered in lab 4.

Q2) Considering your maximum handle position xh (should be around 0.01m) and the maximum force you got previously (if you don't know, try 0.5N), what is minimum spring constant that you need to feel the range of all forces?

In order to find the minimum spring constant model, we need the maximum force $F_{max} = 0.5N$. The maximum handle position is $x_{hmax} = 0.0473m$ which was found in lab 4. In order to feel the range of all forces, we have "k" as the variable given by:

$$k_{max} = \frac{F_{max}}{x_{hmax} - x_0}$$
$$= 5.663\ N/m$$

Rounding it up, k = 6 N/m

Q3) In an ideal spring, the system oscillates back and forth indefinitely. However, due to damping (friction in the axle, motor, etc.), the system is naturally damped and will return to an equilibrium position. Move the handle all the way to one side and release. Achieve an underdamped response by changing the spring constant. Note down the value of the spring constant for later submission. Note: there are a range of values that can give this response.

Using the value k = 6 N/m, the handle was moved to maximum displacement from the equilibrium position and then left, which made the handle oscillate for a short time, like a pendulum and finally come to a stop. The code for making this happen is as follows:

```
// Virtual spring section
 float x0 = -0.0410;
 int k = 6;
 force = -k * (xh-x0);

 tau = rh*rp*force/rs;
```

Exercise 2

Q1) What value of $Festimate$ worked best? At around what distance (xh) did you feel the handle stop? Why do you think that is?

For this question, we use an if-else statement to create a virtual wall. If x_{handle} is larger than $x_{wall} = 0.0025m$, a strong force against the user applying the force can be felt, simulating the situation of a strong counter force perpendicular to the user's if the user pressed hard against a wall. Code for this is as follows

```
float x_wall = 0.0025;

if (xh > x_wall) {
    force = force_estimate * (x_wall - xh);
} else {
  force = 0;
}

tau = rh*rp*force/rs;
```

Exercise 3

Q1) Start with a damping coefficient of 0.05Ns/m. Upload the sketch and vary the speed the handle is moved. If nothing feels different, increase the value and reupload the sketch. Test different values until a reasonable effect of damping is felt. Provide this value in the report.

b = 0.02

```
// Virtual damper section
double lastXh = 0; //last x position of the handle
double vh = 0; //velocity of the handle
double lastVh = 0; //last velocity of the handle
double lastLastVh = 0; //last last velocity of the handle
double dT = 0.0001; //estimated loop time

vh = (((xh - lastXh)/dT) + lastVh + lastLastVh)/3;
lastXh = xh; //update the previous values with the current
lastLastVh = lastVh;
lastVh = vh;

Serial.println(vh, 5);

double b = 0.02; //damping coefficient
force = -b*vh;

tau = rh*rp*force/rs;
```

Exercise 4

Q1) What forces do you think were involved in making the sphere simulation and hollow cube simulation? How do you think the rubbery effect of the sphere was perceived?

For the sphere simulation, normal force from the contact point on the sphere is felt by the user. The force is similar to a force felt when touching a wall but is encountered in a spherical shape which gives the effect of a spherical simulation. The hollow cube simulation is experienced due to the wall effect again by feeling the normal force from five planes (left, right, up, down, front).

Virtual textures like the rubber effect can be synthesized by actuating a friction display with a time-varying periodic signal such as a square or sinusoidal wave. The signal is mapped directly to the driving voltage of an electrostatic display, or modulates the high-frequency carrier signal of an ultrasonic display. Varying the frequency and amplitude of an electrovibrating signal can evoke natural sensations such as rubber.

Exercise 5

Q1) Output the position of the Phantom Touch in the Cartesian Space. The output contains the x,y, and z cartesian coordinates of the end-effector as a column vector. Use a derivative block to get velocity, and another to get acceleration. Set the stop time to 20 seconds. Scope and observe the position, velocity, and acceleration as you move the device. Provide your plots in the report. Is the position, velocity, and acceleration performing as expected? Why or why not? Which direction (left/right, up/down, in/out) is associated with +x/-x, +y/-y, and +z/-z?

The Phantom touch follows a designated 3-axis coordinate system where

+x is left from user's ride

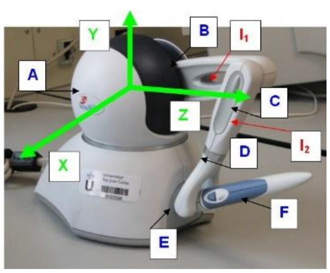
-x is right from user's ride

+y is up from user's ride

-y is down from user's ride

+z is in from user's ride

-z is out from user's ride

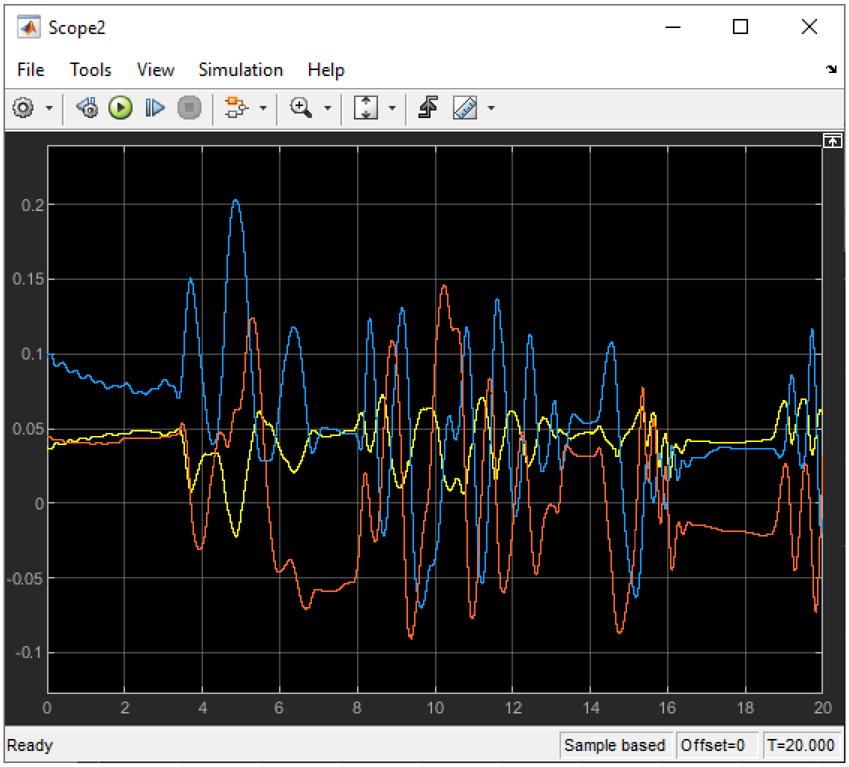


Position graph is performing as expected. Movements in aforementioned directions produces the relevant result in the graphs of x, y, z curves shown below.

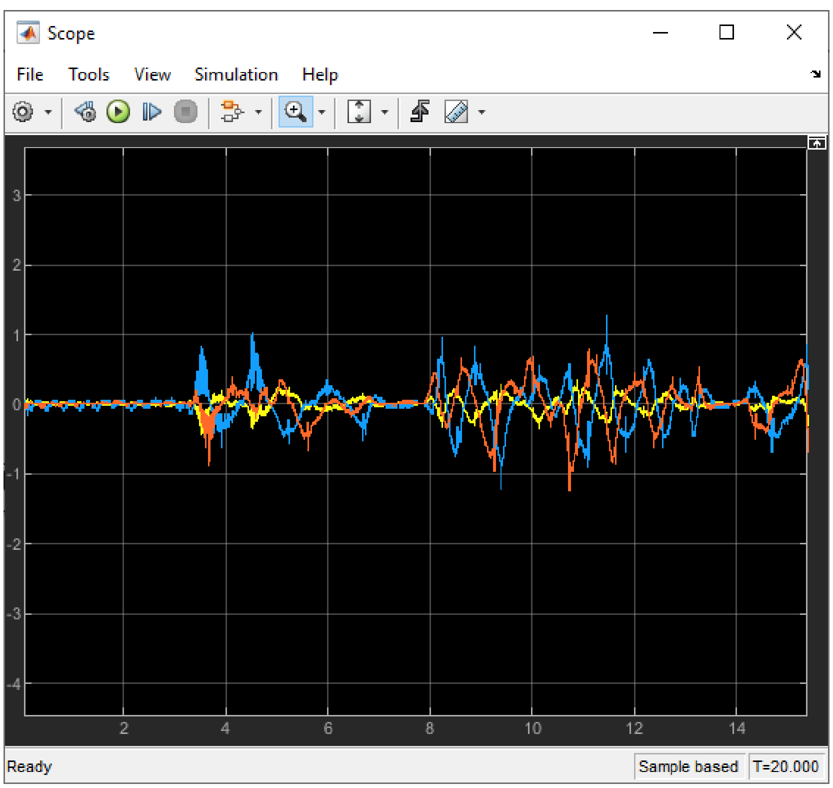
Velocity graph is performing as expected. Changes in position in a particular direction on the position graph maps to changes in velocity in that direction in the velocity graph.

Acceleration graph also seems to be performing as expected but this cannot be confirmed. This is because the graph has a lot of noise. This can be corrected by using a low pass filter in Simulink to remove high frequency waves.

Position


Scope2
File Tools View Simulation Help
0.2
0.15
0.1
0.05
0
-0.05
-0.1
0 2 4 6 8 10 12 14 16 18 20
Ready
Sample based
Offset=0
T=20.000

velocity


Scope
File Tools View Simulation Help
3
2
1
0
-1
-2
-3
-4
2 4 6 8 10 12 14
Ready
Sample based
T=20.000

Acceleration