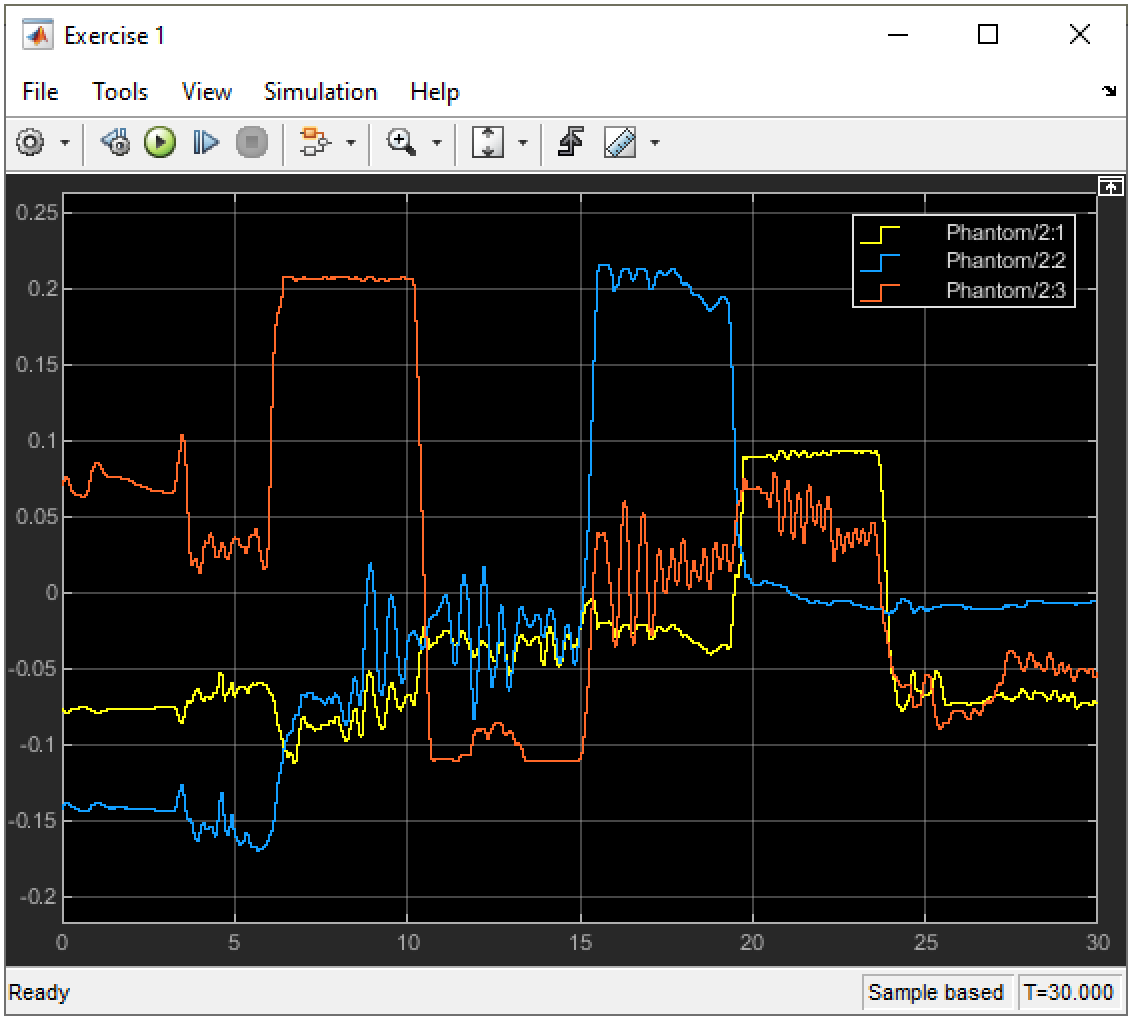
Exercise 1



X → yellow

Y → Blue

Z → Red

Max positive x = 0.09

Max positive y = 0.22

Max positive z = 0.21

Max negative x = -0.11

Max negative y = -0.17

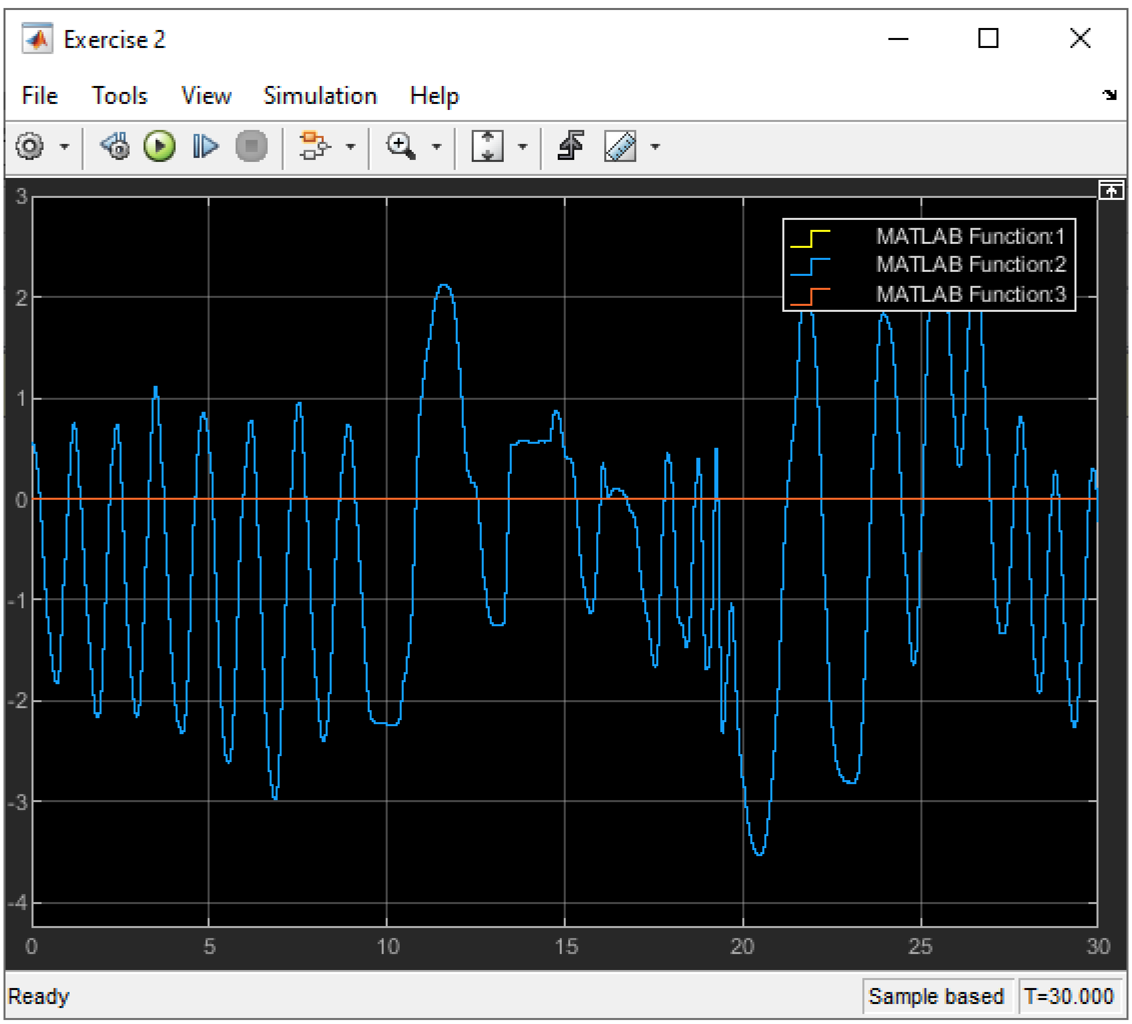
Max negative z = -0.11

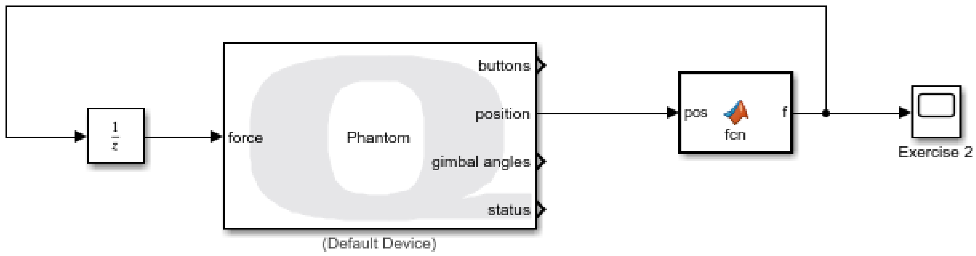
Exercise 2

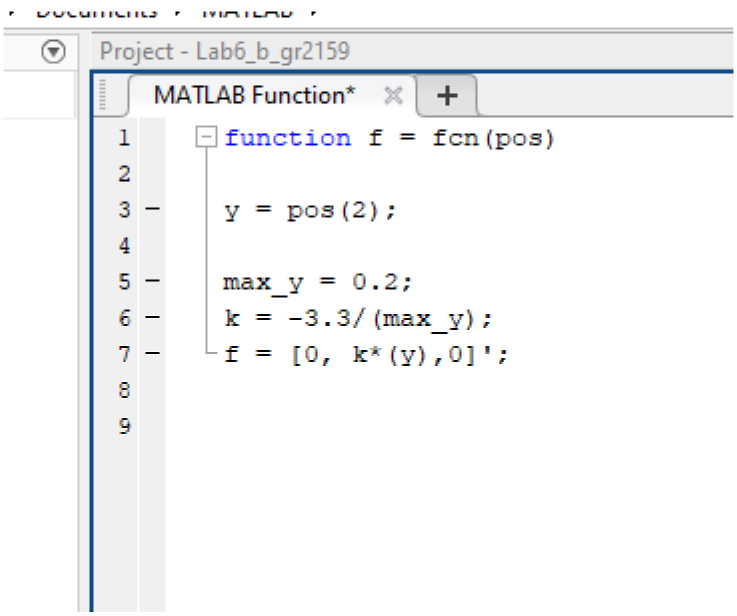
a) What did you choose as y_0?

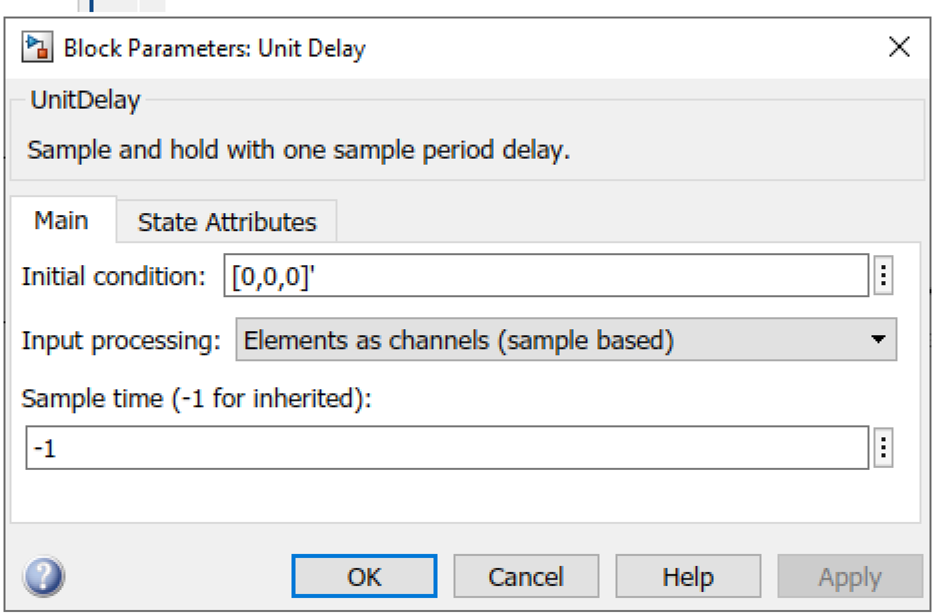
I took $y_0 = 0m$ because I want to measure the restoring force that brings the spring back to equilibrium.

b) Provide a snippet of your model and your code (if used) in the report. Plot the forces.




1
z
force
Phantom
buttons
position
gimbal angles
status
(Default Device)
pos
fcn
f
Exercise 2


Project - Lab6_b_gr2159
MATLAB Function*
function f = fcn(pos)
y = pos(2);
max_y = 0.2;
k = -3.3/(max_y);
f = [0, k*(y),0]';


Block Parameters: Unit Delay
UnitDelay
Sample and hold with one sample period delay.
Main
State Attributes
Initial condition:
[0,0,0]'
Input processing:
Elements as channels (sample based)
Sample time (-1 for inherited):
-1
OK
Cancel
Help
Apply

c) Why is the unit delay necessary? Why do you need to make sure the input is 3x1?

The output position of the Phantom Omni block is split between 3 components x, y, and z and is represented by a 3x1 vector. We want the effect of the spring only in the y-direction.

The unit delay is necessary because we are feeding in the same force that was calculated in the previous time unit. This means that initially, we will have a “chicken and egg” problem where we do not have a force to feed into the Phantom Omni because we don’t have the force calculated using the virtual spring and position. The unit delay skips the first second and makes it so that initially, we do have some value for force and position.

Exercise 3: “Black Hole”

Using your recorded positions from Exercise 1, create a “black hole” at the center of your workspace modelled as a restoring spring force (not a real gravitational pull).

a) What are the implications of using the actual equation for gravity?

If we use Newton’s Gravitational law, we will have to assign a mass, *m*, to position pos which is given by $\sqrt{(x-x_{center})^2+(y-y_{center})^2+(z-z_{center})^2}$ and mass, *M*, to pos_{center} which is given by $\sqrt{x_{center}{}^2+y_{center}{}^2+z_{center}{}^2}$

b) What is the position (x, y, z) of your black hole?

$$(\mathrm{x}, \mathrm{y}, \mathrm{z}) = (0{,}0{,}0)$$

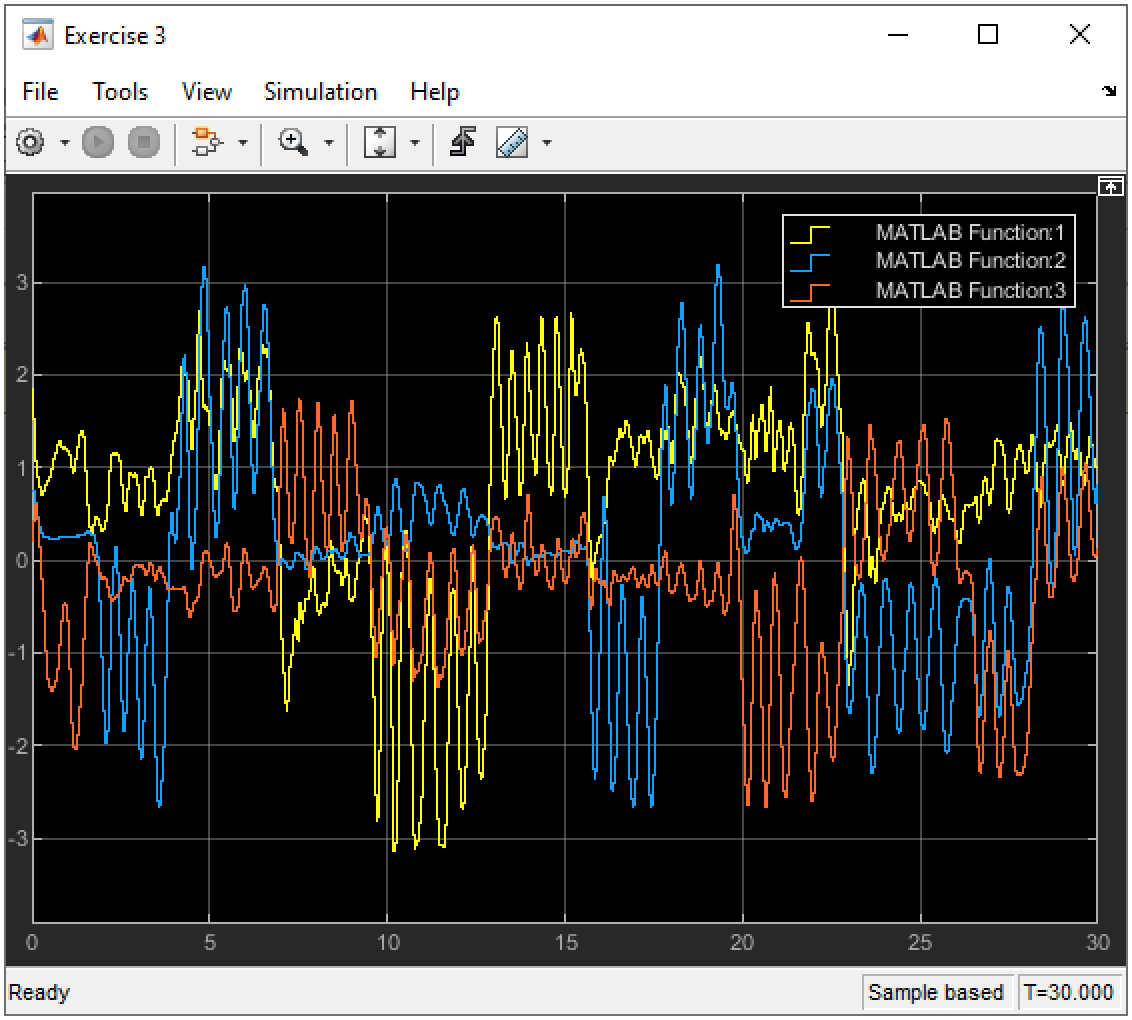
c) What spring constants did you utilize for each direction? Why?

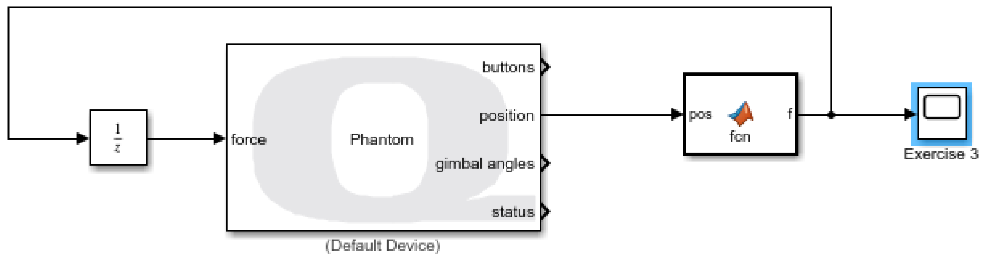
Since we have three directions in which k has an effect, we have k as a vector of three components, where

```
max_x = 0.09;
max_y = 0.2;
max_z = 0.19;

k_y = -3.3/(max_y);
k_x = -3.3/(max_x);
k_z = -3.3/(max_z);
```

Because we want the spring constant k to provide restoring force until the maximum display in each of the 3 directions.

d) Provide snippet of your Simulink model and MATLAB function code (if used) in the report. Plot the forces.

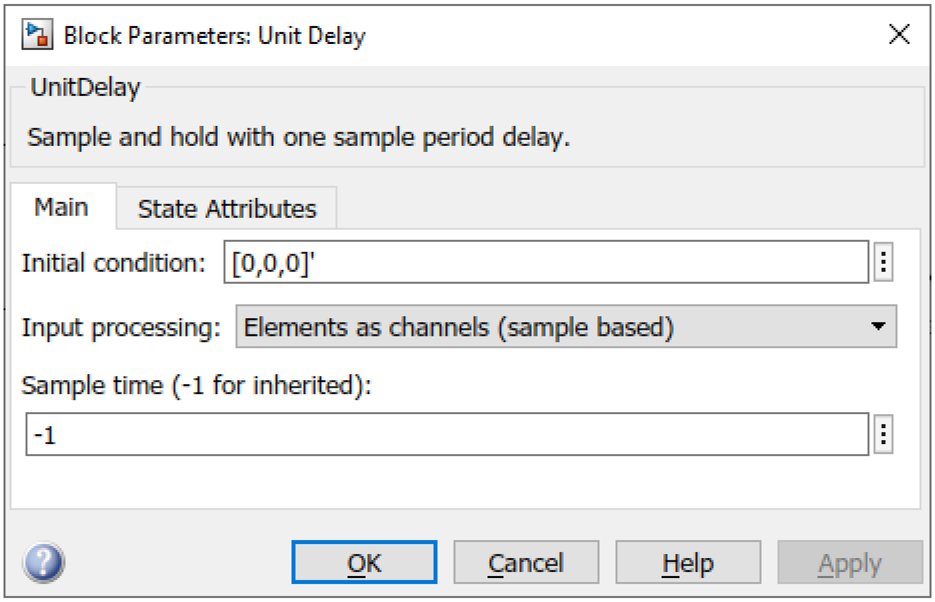




Project - Lab6_b_gr2159

MATLAB Function

```matlab
function f = fcn(pos)


x = pos(1);
y = pos(2);
z = pos(3);

max_x = 0.09;
max_y = 0.2;
max_z = 0.19;

k_y = -3.3/(max_y);
k_x = -3.3/(max_x);
k_z = -3.3/(max_z);

f = [k_x*(x), k_y*(y), k_z*(z)]';


```



Exercise 4: Virtual Wall (+y direction only)

Create a virtual wall in the +y-direction of your workspace. Remember the equation for a wall in one direction:

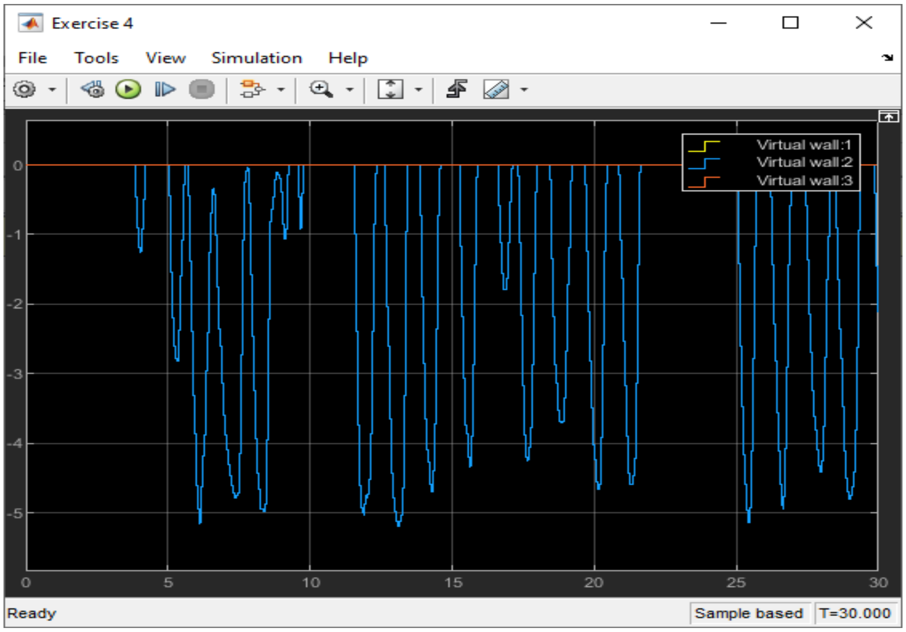
a) What did you choose as ‘y_w’?

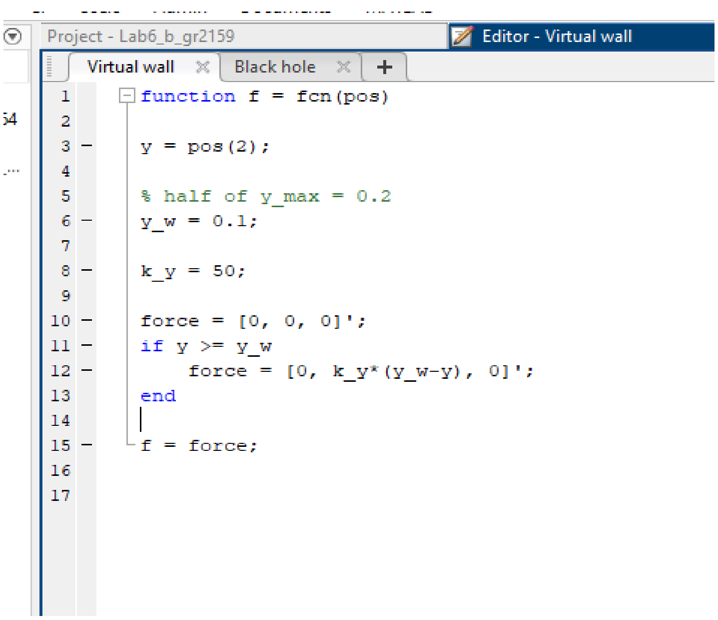
$y_w = 0.1$ which is the half of $y_{max} = 0.2$

b) What value of ‘K’ works best?

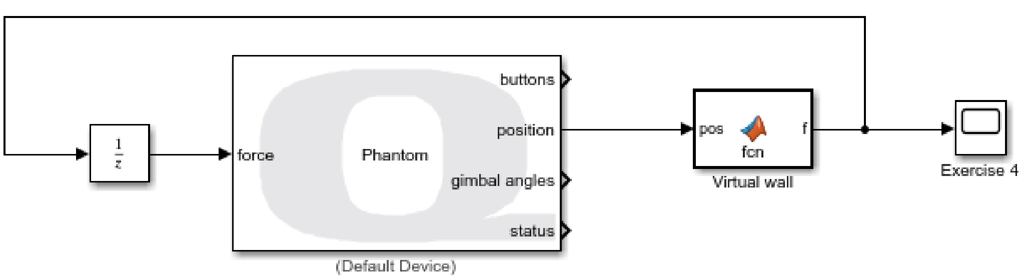
After experimenting with different values, I realized that k=50N/m. This value of k is a high enough spring constant such that we feel a wall and feel maximal restoring force.

c) Provide a snippet of your Simulink Model and MATLAB function block (if used).x






```
function f = fcn(pos)

y = pos(2);

% half of y_max = 0.2
y_w = 0.1;

k_y = 50;

force = [0, 0, 0]';
if y >= y_w
    force = [0, k_y*(y_w-y), 0]';
end

f = force;
```



Exercise 5: Virtual Cube

Create a virtual cube centered at [0,0,0]. This is done by creating a wall in all 6 directions (requires 6 if-else statements at most, can be accomplished with 3). Be mindful about the logic of all your if-else statements. (*to accomplish in 3 if-else, you need to use the function sign() and abs()).

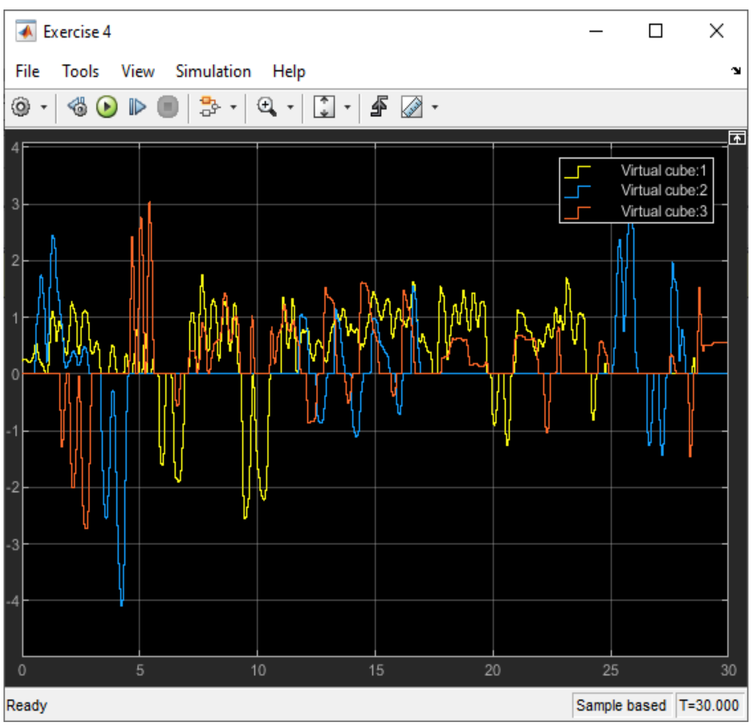
a) What $|x_w|$, $|y_w|$, and $|z_w|$ did you choose for the wall positions?

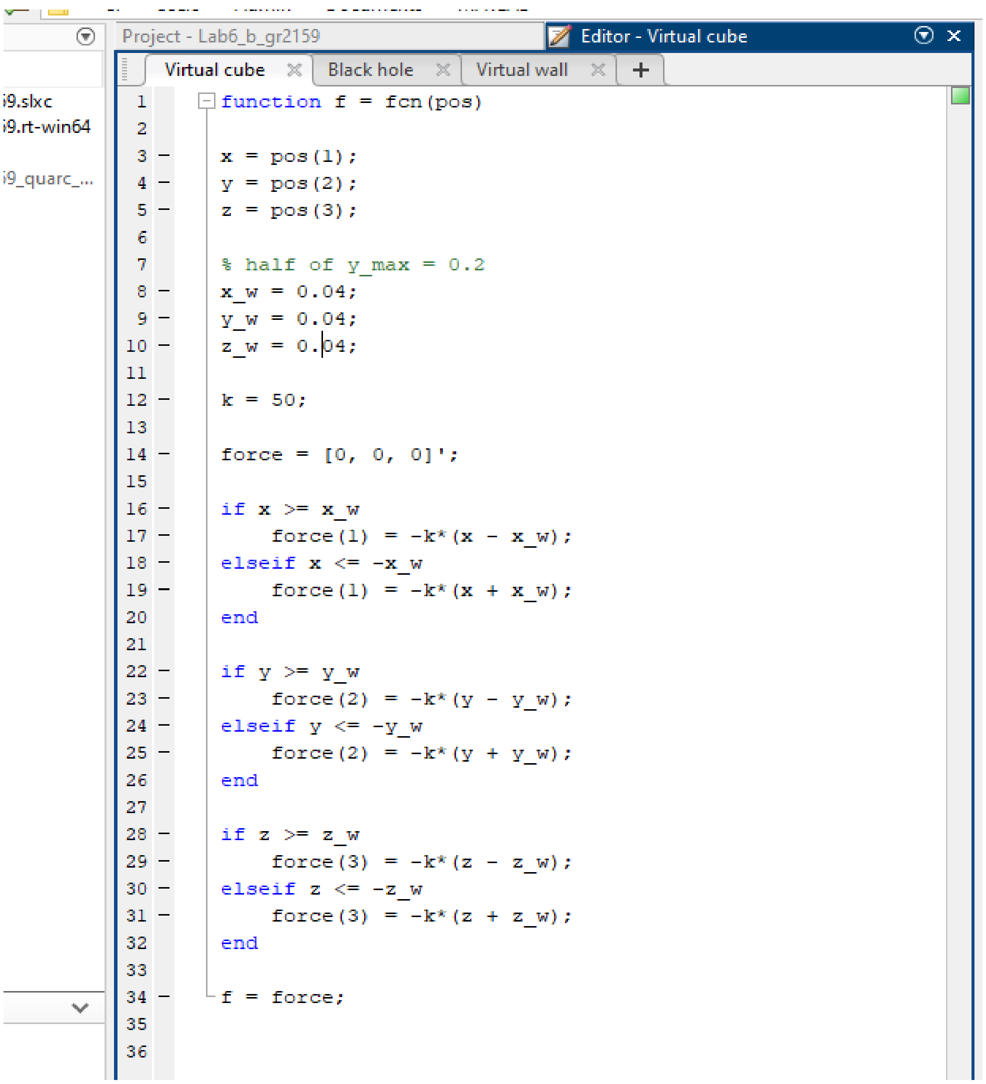
```
x_w = 0.04;
y_w = 0.04;
z_w = 0.04;
```

b) What ‘K’ value(s) did you use?

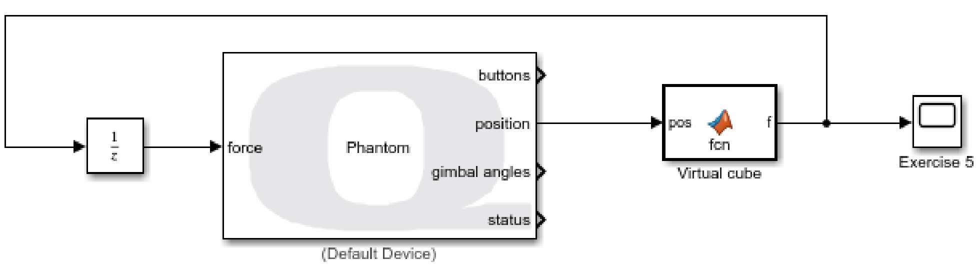
After experimenting with different values, I realized that k=50N/m. This value of k is a high enough spring constant such that we feel a wall and feel maximal restoring force.

c) Provide a snippet of your Simulink Model and MATLAB function block (if used).






```
function f = fcn(pos)

x = pos(1);
y = pos(2);
z = pos(3);

% half of y_max = 0.2
x_w = 0.04;
y_w = 0.04;
z_w = 0.04;

k = 50;

force = [0, 0, 0]';

if x >= x_w
    force(1) = -k*(x - x_w);
elseif x <= -x_w
    force(1) = -k*(x + x_w);
end

if y >= y_w
    force(2) = -k*(y - y_w);
elseif y <= -y_w
    force(2) = -k*(y + y_w);
end

if z >= z_w
    force(3) = -k*(z - z_w);
elseif z <= -z_w
    force(3) = -k*(z + z_w);
end

f = force;
```



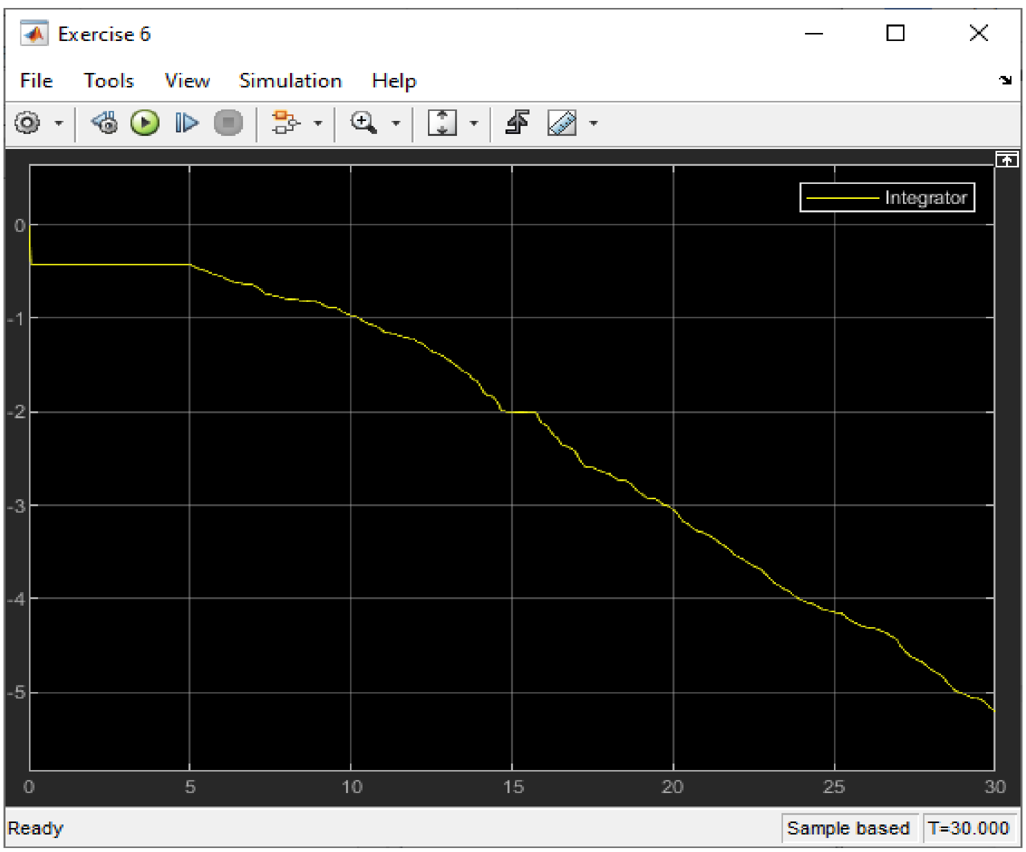
Exercise 6: Virtual Damping (positive damping)

Create a virtual damper in all directions. Utilize the 'Second-Order Low-Pass Filter' that is a part of the QUARC library to output the velocity. 'x' is the input, 'y' is the filtered 'x' and 'yd' is the filtered derivative of 'x'. Double click the block; keep the damping ratio as 1 and change the cut-off frequency to 50 rad/sec. Recall the formula for a damper is: F = bv. However, in this case, make and add a negative gain after the unit delay. This is done to model the two-channel teleoperation where force feedback is subtracted and to observe the energy accumulation from the environment.

a) What did you choose as your 'b' value(s)?

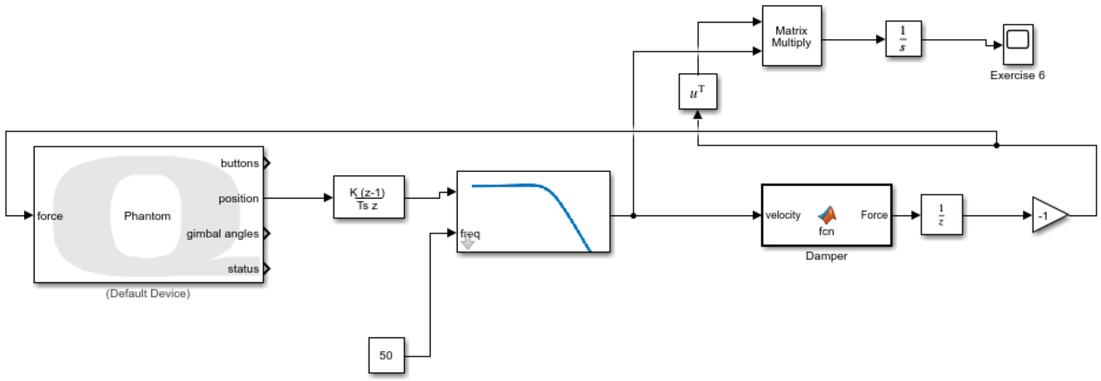
b= 2

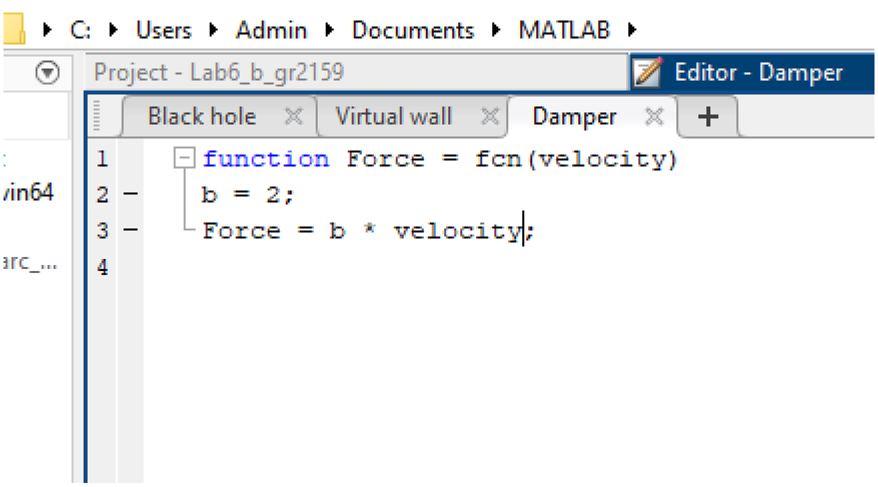
b) Take the product of the force and velocity and integrate it. Plot the energy.



c) Positive damping is like pure resistive therapy. Is this passive? Why or why not? Provide a snippet of your Simulink Model and MATLAB function block (if used).

Yes, according to the graph shown below, we do see that the energy of the system is reducing over a period of time therefore the system is passive.





Exercise 7: Virtual Damping (negative damping)

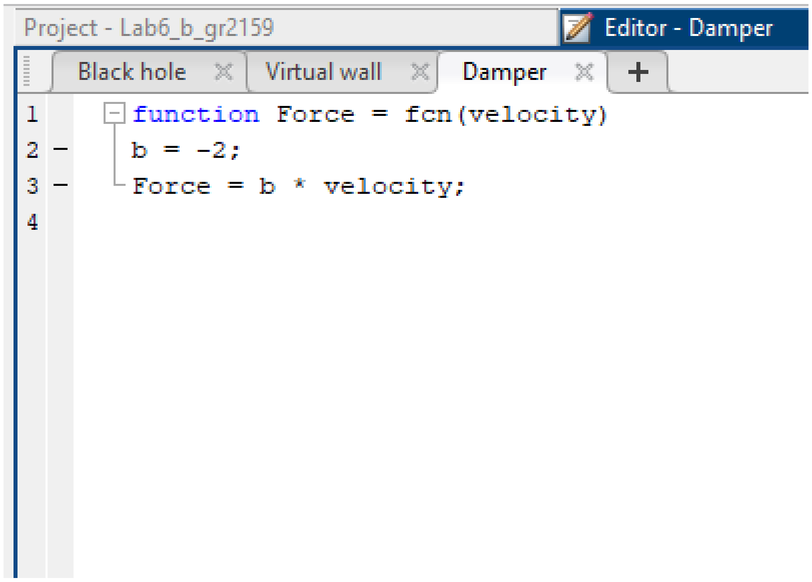
Change your ‘b’ value(s) from the previous exercise from positive to negative.

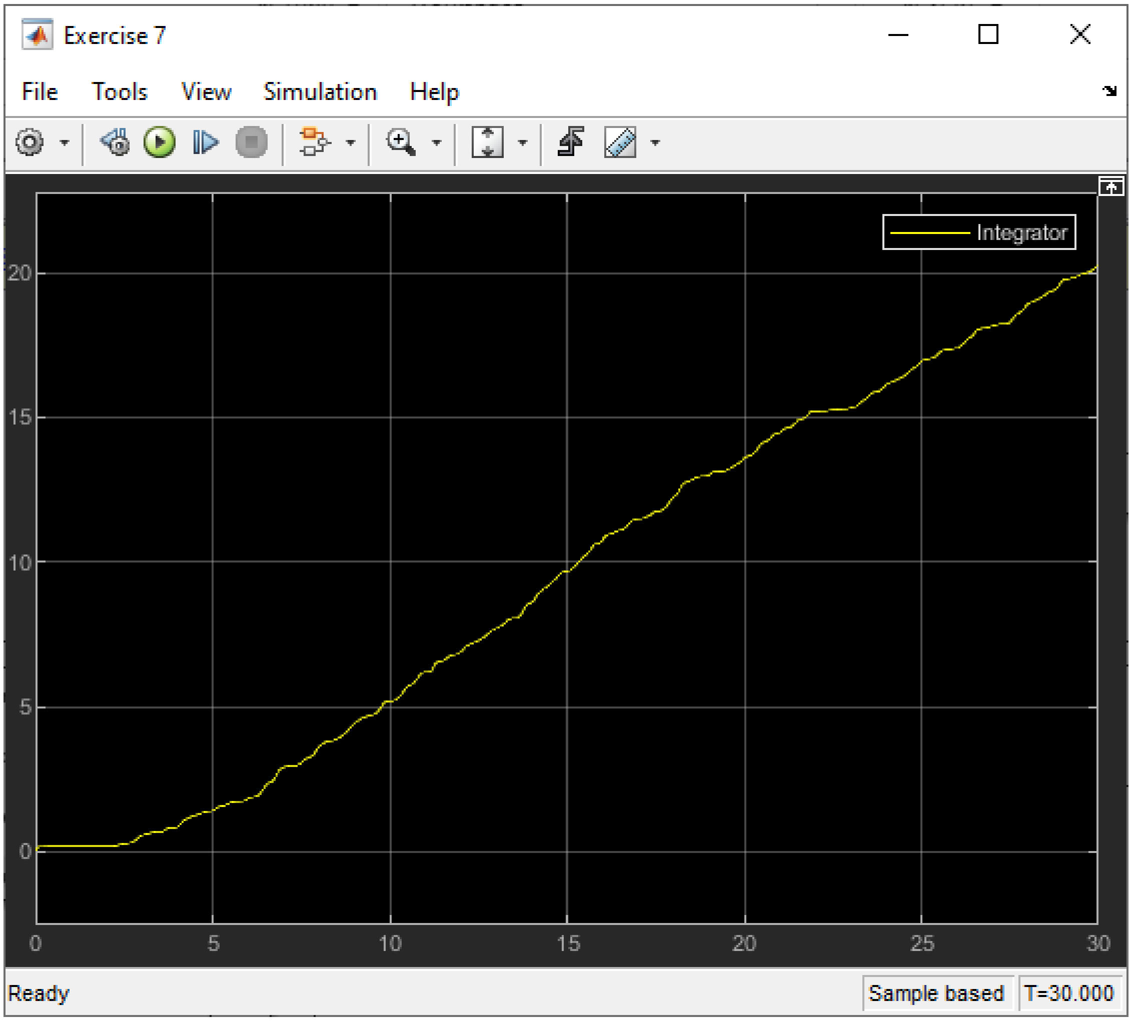
a) Move the handle very slowly in the intended directions. What do you notice?

We face instability as the motion in either x, y, or z directions is magnified by the Phantom Omni.

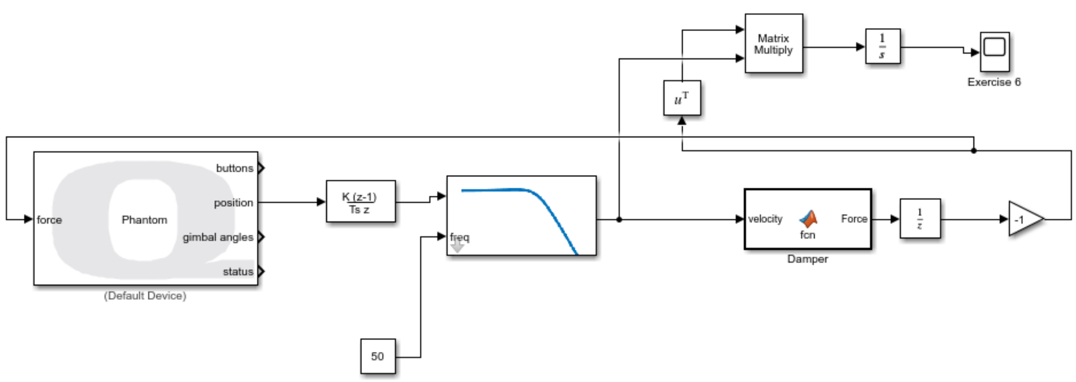
b) Negative damping is like pure assistive therapy. Explain why this is the case. Take the product of the force and velocity and integrate it. Plot the energy.

Negative damping causes the patient to receive experience motion in the direction of their own movement, which is why it is useful in assistance. However, this motion is not passive so it keeps increasing in energy uncontrollably and becomes unstable.





Exercise 7
File Tools View Simulation Help
Integrator
Ready
Sample based
T=30.000



Matrix Multiply
Exercise 6
buttons
position
gimbal angles
status
force
Phantom
(Default Device)
freq
50
velocity
fcn
Force
Damper
-1