Table of Contents

transferGraph.m

This script compares the simulated open and closed loop responses to the experimental open and closed loop responses. The information was then used to create a transfer function which was used to tune the PI controller.

```
timeNow = 0;
timePast = 0;
positionNow = 0;
positionPast = 0;
velocity = zeros(1, 372);
A = table2array(SEEDTRANSFER);
B = A(:,1)./1000;


for i = 1:372
    timeNow = B(i,1);
    positionNow = A(i,2);
    velocity(i) = (positionNow - positionPast)/(timeNow - timePast);

    timePast = timeNow;
    positionPast = positionNow;
end
%velocity = rmmissing(velocity);
%x = linspace(0,300,300);
%y = velocity;
%p = polyfit(x,y,2);

%x1 = linspace(0,300,300);
%y1 = polyval(p,x1);
%figure
%plot(x,y,'o')
%hold on
%plot(x1,y1)
%plot(velocity)
%hold off
```

A Plot of the results

The control design for this stage was very trial and error based. Basically we would upload new code, see how it worked on the physical model, then recalibrate things we needed to change. We only used the simulations to set the base values so that we could have a relative starting place. The motor constants that

we found with this set up was as follows. For out P, 0.289202862842511; for our I, 10.8369142120677. It was about a 150 ms rise time. The experiment that we ran was basically just turning the motor on for two seconds and letting it output the angular position as well as the time to determine the speed and such.

```
xGUY = B(:,1);
yGUY = A(:,2);

%figure
%plot(xGUY, yGUY);
%title('Position')


%figure
%plot(xGUY, velocity);
%title('Close Loop')
% 4.12 milli-somethings for k
% 1400 - 980 = 420 ms
```

Tuning_SCRIPT.m

This script compares the simulated close and open loops with the experimental versions

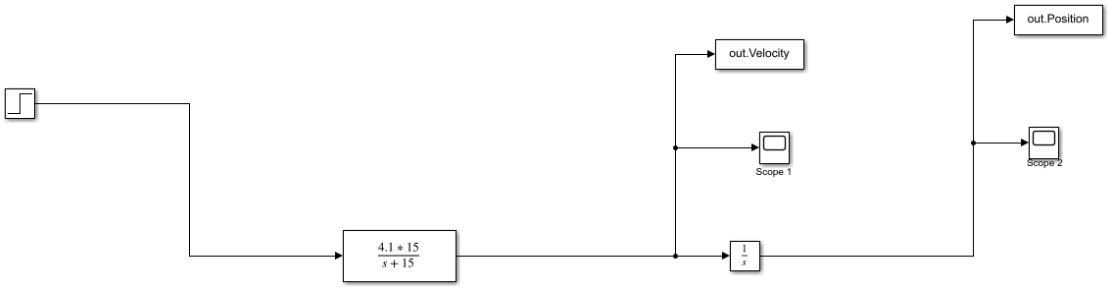
required file: tuning.slx

Run a Simulation

This simulation applies a step input into a transfer function and the motor model in order to compare their open and closed loop responses.

open the block diagram so it appears in the documentation when published. Make sure the block diagram is closed before running the publish function

```
open_system('Tuning')
%
% run the simulation
%
out=sim('Tuning');
```



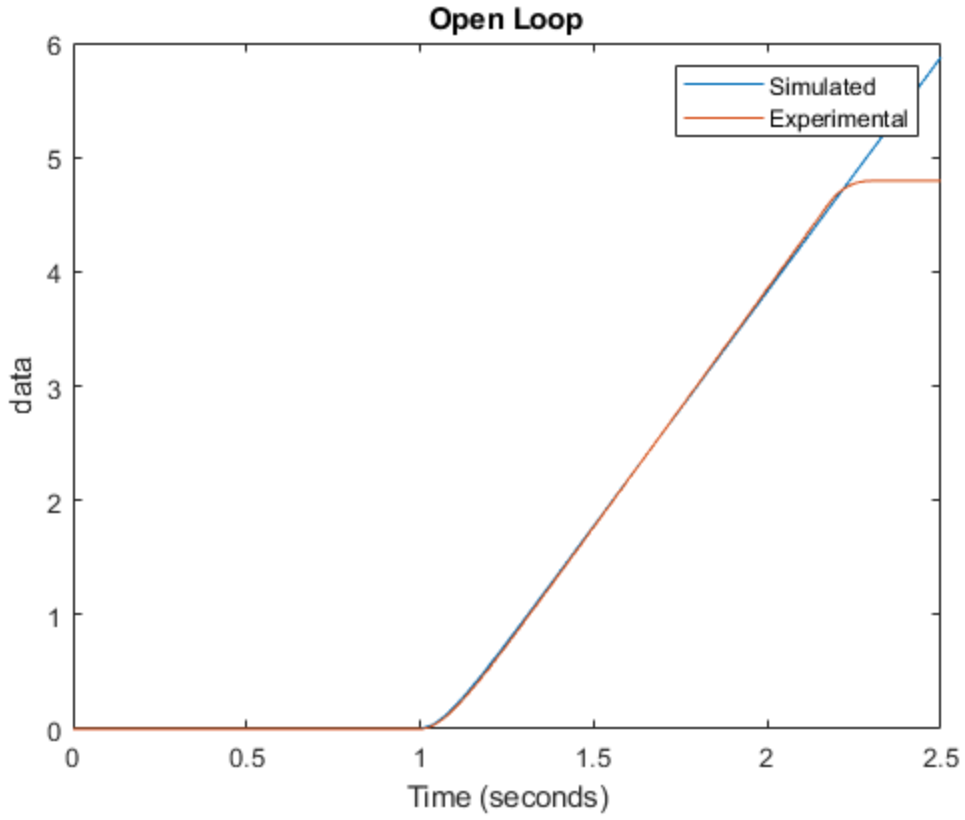
A Plot of the results

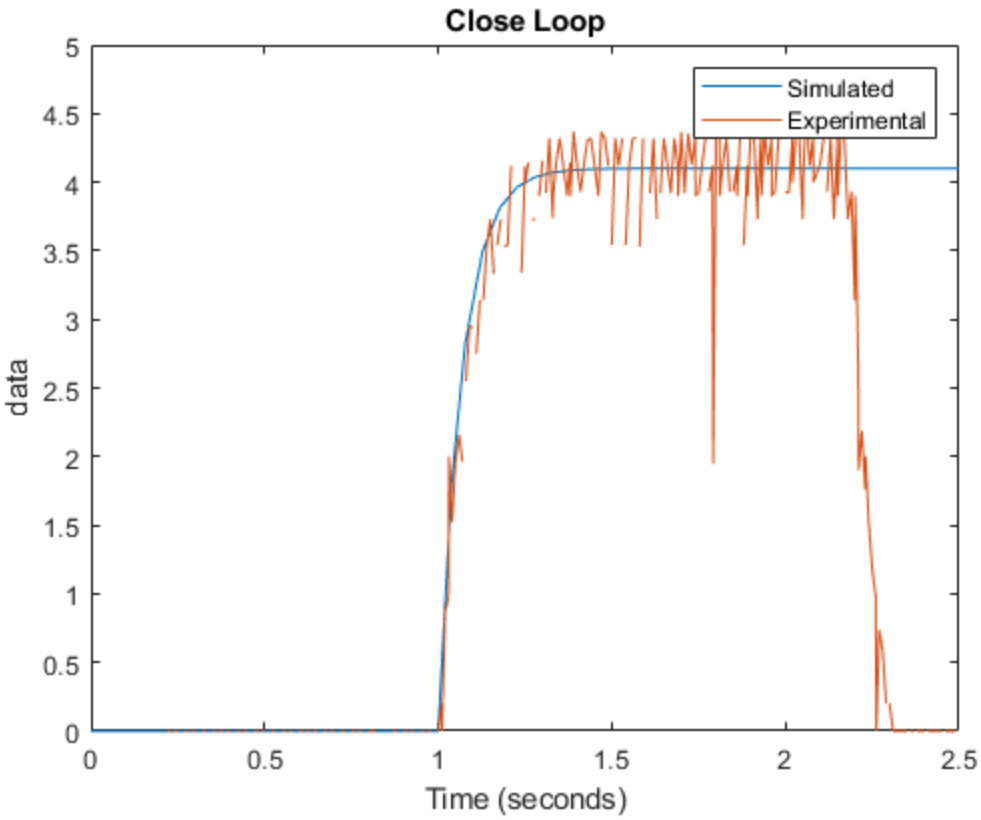
Since the transfer function is tuned to the model, the graph of the responses are overlayed as well as they can be considering the jagged nature of the closed loop reponse.

```
figure
plot(out.Position)
hold on
plot(xGUY, yGUY);
hold off
legend('Simulated','Experimental')
title('Open Loop')

figure
plot(out.Velocity)
hold on
plot(xGUY, velocity)
hold off
legend('Simulated','Experimental')
title('Close Loop')

% Getting the simulated model tuned to the experiement allows us to
 build a
% trasnfer function to test the PI controller against.
```





Published with MATLAB® R2021a