Code Appendix

Main File

This file calls other methods for parameter calculation

```
clc
close all
clear all
format compact

[wn, l_eff] = gyroscope_parameters();
[K_motor, tau] = motor_parameters();

[Kp, Ki] = control_parameters(K_motor, tau, wn, l_eff)
```

Gyroscope Test

This file calculates ω_n and l_{eff} for the system.

```
function [wn, l_eff] = gyroscope_parameters(varargin)

% GYROSCOPE TEST!!!
load("mats/gyrotest.mat");

% find exponential points
[pks,locs] = findpeaks(angle_rad, t);

% get natural frequency
% as difference between peaks
mean_period = mean(diff(locs));
wn = 2*pi/mean_period;

% length as a function of frequency
l_eff = 9.81/(wn^2);

if nargin == 1
    % plot
    hold off
    plot(t, angle_rad, 'DisplayName', 'Angle');
    hold on;
    plot(locs, pks, 'o', 'DisplayName', 'Peaks');
    xlabel("Time (s)");
    ylabel("Angle (rad)");
    title("Angle over Time for an Oscillating Rocky");
    legend;
    savefig("figs/gyroscope_params.fig");
    saveas(gcf, "figs/gyroscope_params.png");
end

end
```

Motor Parameters

This file calculated K_{motor} and τ for Rocky.

```
function [K, tau] = motor_parameters(varargin)

% load stepper data
steppy = load("mats/steptest.mat");

% create custom exponential fit
custom_exp_func = @(K, tau, x) K*(1-exp(-x./tau));
custom_exp_fit = fittype(custom_exp_func);

% create an exponential line of best fit
good_xs = [steppy.t(steppy.t < 0.5); steppy.t(steppy.t < 0.5)];
good_ys = [steppy.outputL(steppy.t < 0.5); steppy.outputR(steppy.t < 0.5)];
fit_params = fit(good_xs, good_ys, custom_exp_fit, 'Start', [1,1]);

% extract K and tau
K = fit_params.K / mean(steppy.input);
disp(mean(steppy.input))
tau = fit_params.tau;

if nargin == 1
    % plot
    hold off
    plot(steppy.t(steppy.t < 0.5), steppy.outputL(steppy.t < 0.5), ...
        '*', 'DisplayName', 'Experimental Left Wheel Data')
    hold on
    plot(steppy.t(steppy.t < 0.5), steppy.outputR(steppy.t < 0.5), ...
        '*', 'DisplayName', 'Experimental Right Wheel Data')
    plot(steppy.t(steppy.t < 0.5), ...
        custom_exp_func(fit_params.K, tau, steppy.t(steppy.t < 0.5)), ...
        'DisplayName', 'Best Fit for Both Wheels Data')
    xlabel("Time (s)");
    ylabel("Velocity (m/s)");
    title("Velocity over Time with a Stepper Input");
    legend;
    savefig("figs/motor_params.fig");
    saveas(gcf, "figs/motor_params.png");
end
```

```
end
```