Spring Break Project

Table of Contents

Daniel Lopez Villa - - dlopezvilla99@gmail.com - - GM9 DC Motor Characterization

Importing and Organizing Data

```
clear; clc; close all;

totalConfigs = 5; %total amount of configurations

%pre-allocate Data cell array and average data cell array
configData = cell(totalConfigs,1);

%index through txt files to transfer to the configData cell array
for i=1:totalConfigs
        config = num2str(i);
        fileName = "encoderTest_" + config + ".txt";
        configData{i} = abs(importdata(fileName));
end
```

Calculating Velocity

```
%establish necessary conversion values
gear_ratio = 143;
rotation_per_tick = 1/84;
time_micros = 1;%1e6;
conv = gear_ratio*rotation_per_tick*60; %conversion: encoderValue/s -
> rpm

%pre-allocate velocity and time cell arrays
velocity = cell(totalConfigs,1);
timeVector = cell(totalConfigs,1);
displacement = cell(totalConfigs,1);

for i=1:totalConfigs
    encoderCount = configData{i}(:,2); %Encoder tick vector
    displacement{i} = 2*pi*encoderCount(1:end-1)*conv/60;
    timeVector{i} = configData{i}(1:end-1,1)/time_micros; %time vector
 [s]
    velocity{i} = diff(encoderCount(1:end-1))./
diff(timeVector{i})*conv;
```

```
end
```

Filtering

```
%pre-allocate filteredVelocity cell array
filteredVelocity = cell(totalConfigs,1);
for i=1:totalConfigs
    filteredVelocity{i} = zeros(1,length(timeVector{i}));
end

%compute filtered velocity using Vfi = alpha*Vi + (1-alpha)*Vf(i-1)
for i=1:totalConfigs
    for j=2:length(timeVector{i})-1
    tau = .05;
    delta_t = timeVector{i}(j)-timeVector{i}(j-1);
    alpha = delta_t/(delta_t + tau);
    filteredVelocity{i}(j) = alpha*velocity{i}(j) + (1-
alpha)*filteredVelocity{i}(j-1);
    end
end
avgFvel = (filteredVelocity{1} + filteredVelocity{2} +
 filteredVelocity{3}...
    + filteredVelocity{4} + filteredVelocity{5})/5;

avgTermVel = mean(avgFvel(end-30:end));
MeasuredV = zeros(1,totalConfigs)'; %average measured Terminal
 Velocity

for i=1:totalConfigs
        MeasuredV(i) = mean(filteredVelocity{i}(end-30:end));
end
```

Calculating Torque

```
V = 7.4;%Voltage
wNoLoad = avgTermVel*(2*pi/60);
stallCurrent = 0.8;% Stall current
R = V/stallCurrent;
Kv = V/wNoLoad;
Tstall = Kv*(V/R); %stallTorque
Kt = Tstall/wNoLoad;
%calculate index where velocity is within 95% of steady state
wss1 = wNoLoad;
indexVector1 = find(avgFvel >.95*wss1);
interval1 = indexVector1(1);
x = timeVector{1}(1:interval1);
y = displacement{1}(1:interval1);
```

Line Fitting

```
%line fit no flywheel
fo= fitoptions('Method','NonlinearLeastSquares',... % analysis type
```

```
    'StartPoint',[0, .02]); % guesses for variables a and b

ft = fittype('wss1*(x-a+b*(exp(-(x-a)/
b)-1))','problem','wss1','options',fo);
[curve,gof] = fit(timeVector{1}(1:interval1),displacement{1}
(1:interval1),ft,'problem',wss1);
t01 = curve.a; % time offset (s)
tau1 = curve.b; %system time constant (s)
correlation  = gof.rsquare;
```

Calculating RMS

```
%precalculate calibrated fitted line
rdispfit1 = zeros(1,interval1);

rms1=0;
for j=1:interval1
   rdispfit1(j)=wss1*(timeVector{1}(j)-t01+tau1*(exp(-(timeVector{1}
(j)-t01)/tau1)-1));
   error=displacement{1}(j)-rdispfit1(j);
   rms1=rms1+error^2;
end
rms1=sqrt(rms1/interval1); % calculated rms error

I1=tau1*Tstall/wNoLoad; % calculate effective mass inertia

Frame = ["Motor Shaft";"Output Shaft"];
gearBox = [143;143];
freeSpeedRAD = [wNoLoad;wNoLoad/(143)];
freeSpeedRPM = [avgTermVel;avgTermVel/143];
motorVConstant = [Kv ; V/(wNoLoad/143)];
motorTConstant = [Kt;(Tstall/wNoLoad)/143];
stallTorque = [Tstall;(V/(wNoLoad/143))*(V/R)];
Inertia = [I1;tau1*stallTorque(2)/freeSpeedRAD(2)];

%display measured motor specs
Specs1 = table(Frame,stallTorque,motorVConstant,motorTConstant)
Specs2 = table(Frame,freeSpeedRAD,freeSpeedRPM,Inertia)
```

Specs1 =

2×4 table

Frame	*stallTorque*	*motorVConstant*	*motorTConstant*
"Motor Shaft"	*0.0036021*	*0.0045027*	*2.1918e-06*
"Output Shaft"	*0.51511*	*0.64388*	*1.5327e-08*

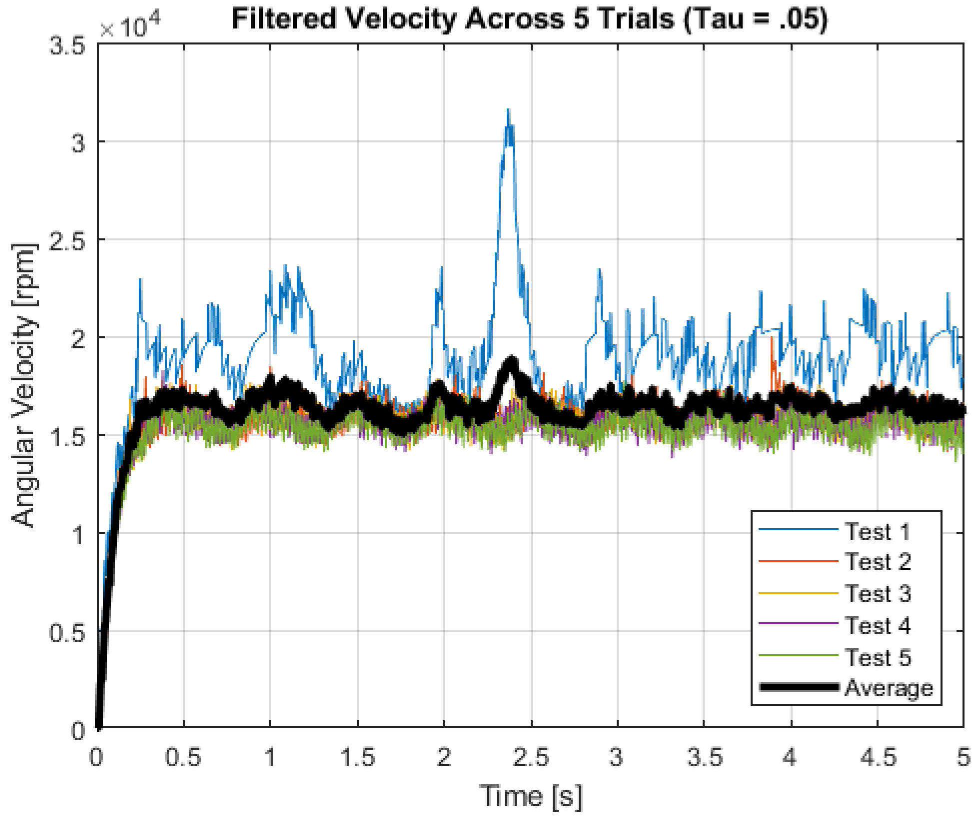
Specs2 =

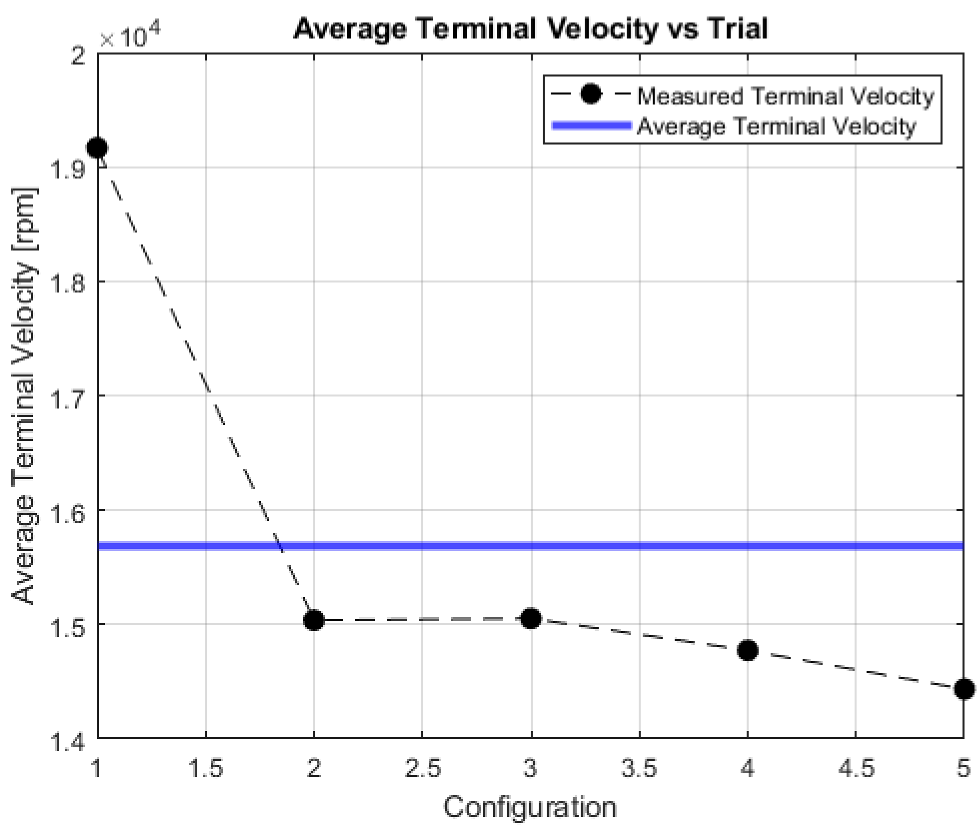
2×4 table

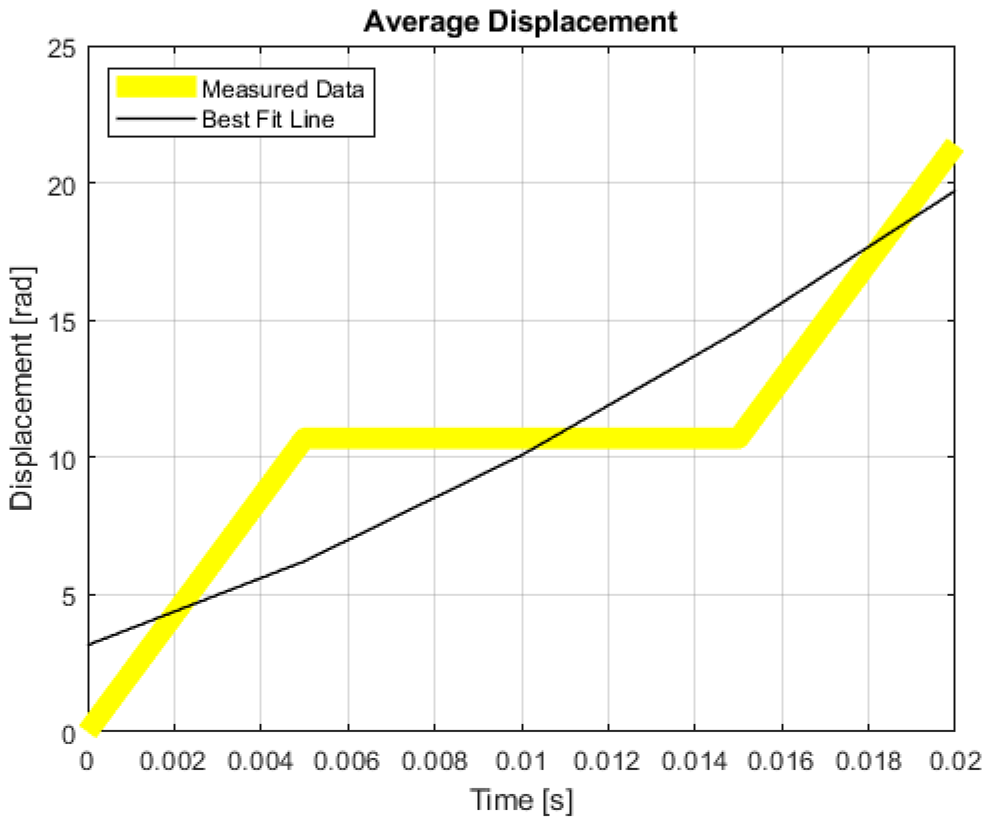
Frame	freeSpeedRAD	freeSpeedRPM	Inertia
"Motor Shaft"	1643.5	15694	6.5085e-08
"Output Shaft"	11.493	109.75	0.0013309

Plotting

```
%filtered velocity plot
figure(1)
for i=1:totalConfigs
    plot(timeVector{i}(1:end-1), filteredVelocity{i}(1:end-1));
    hold on
end
plot(timeVector{1}(1:end-1), avgFvel(1:end-1),'k','Linewidth',3);
hold on
grid on

legend({'Test 1','Test 2','Test 3','Test 4','Test
 5','Average'},'Location','southeast');
ylabel('Angular Velocity [rpm]');
xlabel('Time [s]');
title('Filtered Velocity Across 5 Trials (Tau = .05)');

%terminal velocity plot
figure(2)
plot((1:totalConfigs),MeasuredV,'--k.','MarkerSize',28);
hold on
yline(avgTermVel,'b','Linewidth',3);
grid on

ylabel('Average Terminal Velocity [rpm]');
xlabel('Configuration');
title('Average Terminal Velocity vs Trial');
legend('Measured Terminal Velocity','Average Terminal Velocity')

%displacement plot
figure(3)
plot(timeVector{1}(1:interval1),displacement{1}
(1:interval1),'y','Linewidth', 8);
hold on
plot(timeVector{1}(1:interval1),rdispfit1,'k','Linewidth', 1);
hold on
grid on
xlabel("Time [s]")
ylabel("Displacement [rad]")
title("Average Displacement")
legend('Measured Data','Best Fit Line','location','northwest');
```


Filtered Velocity Across 5 Trials (Tau = .05)
×10^4
Angular Velocity [rpm]
Time [s]
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
Test 1
Test 2
Test 3
Test 4
Test 5
Average


Average Terminal Velocity vs Trial
×10^4
Average Terminal Velocity [rpm]
Configuration
1.4
1.5
1.6
1.7
1.8
1.9
2
1
1.5
2
2.5
3
3.5
4
4.5
5
Measured Terminal Velocity
Average Terminal Velocity



Published with MATLAB® R2019b