Contents

Code that reads the position(angle) of an encoder to get a transfer function that we can tune

```
Data = csvread('arduinoData.csv',1,0);
ThetaDot = Data (:,1)
time = Data (:,2)


%%Variables to tune to get a Model Transfer Function
K = 8;
sig = 50;
```

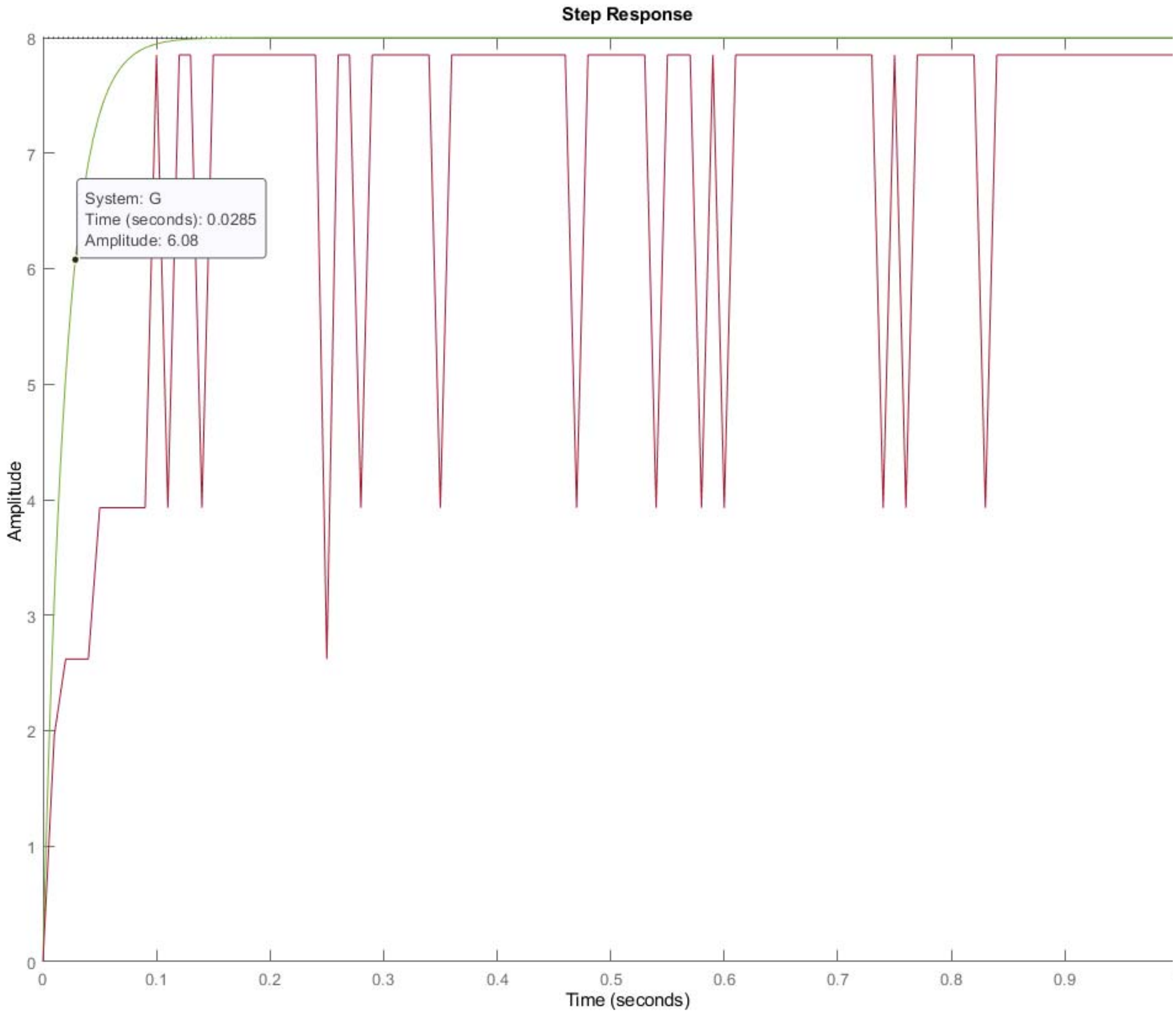
```
ThetaDot =

         0
    1.9600
    2.6200
    2.6200
    2.6200
    3.9300
    3.9300
    3.9300
    3.9300
    3.9300
    7.8500
    3.9300
    7.8500
    7.8500
    3.9300
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500
    2.6200
    7.8500
    7.8500
    3.9300
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500
    3.9300
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500
    3.9300
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500
    3.9300
    7.8500
    7.8500
    7.8500
    3.9300
    7.8500
    3.9300
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500
```

```
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500
    3.9300
    7.8500
    3.9300
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500
    3.9300
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500
    7.8500


time =

         0
    0.0100
    0.0200
    0.0300
    0.0400
    0.0500
    0.0600
    0.0700
    0.0800
    0.0900
    0.1000
    0.1100
    0.1200
    0.1300
    0.1400
    0.1500
    0.1600
    0.1700
    0.1800
    0.1900
    0.2000
    0.2100
    0.2200
    0.2300
    0.2400
    0.2500
    0.2600
    0.2700
    0.2800
    0.2900
    0.3000
    0.3100
    0.3200
    0.3300
    0.3400
    0.3500
    0.3600
    0.3700
    0.3800
    0.3900
    0.4000
    0.4100
    0.4200
    0.4300
    0.4400
    0.4500
    0.4600
    0.4700
    0.4800
    0.4900
```

```
    0.5000
    0.5100
    0.5200
    0.5300
    0.5400
    0.5500
    0.5600
    0.5700
    0.5800
    0.5900
    0.6000
    0.6100
    0.6200
    0.6300
    0.6400
    0.6500
    0.6600
    0.6700
    0.6800
    0.6900
    0.7000
    0.7100
    0.7200
    0.7300
    0.7400
    0.7500
    0.7600
    0.7700
    0.7800
    0.7900
    0.8000
    0.8100
    0.8200
    0.8300
    0.8400
    0.8500
    0.8600
    0.8700
    0.8800
    0.8900
    0.9000
    0.9100
    0.9200
    0.9300
    0.9400
    0.9500
    0.9600
    0.9700
    0.9800
    0.9900
    1.0000
    1.0100
    1.0200
    1.0300
```

Variables calling matlab Function

```
s = tf('s');
figure(1);
hold on;
plot(time, ThetaDot);
```

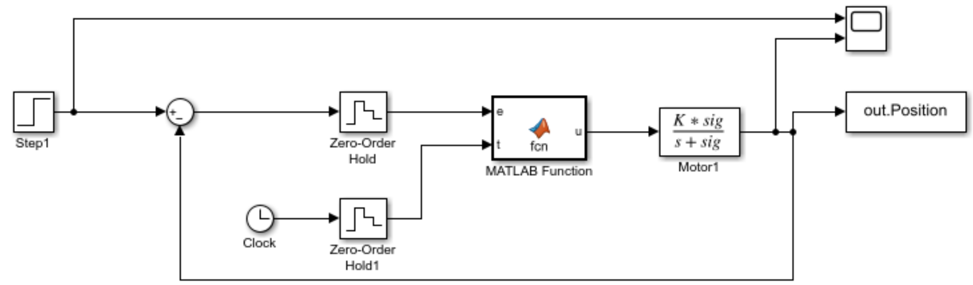


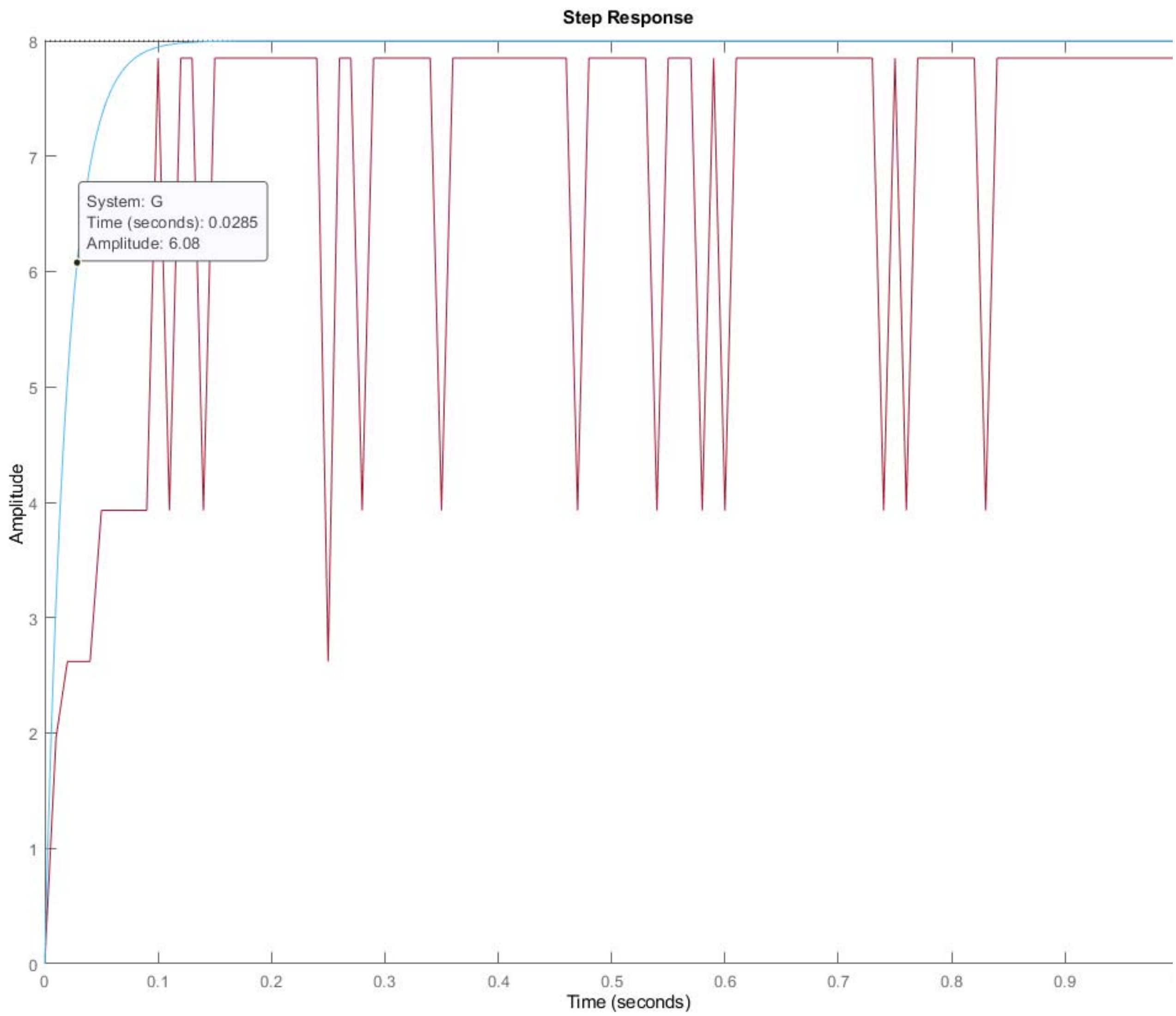
Transfer Function

```
G = K * (sig)/(s+sig)
step(G,1.03);
stepinfo(G)
hold off;

open_system("motorsimulation4p6");
%
% run the simulation
%
out=sim("motorsimulation4p6");
```

```
G =

   400
  ------
  s + 50

Continuous-time transfer function.


ans =

  struct with fields:

        RiseTime: 0.0439
    SettlingTime: 0.0782
     SettlingMin: 7.2360
```

```
     SettlingMax: 7.9998
       Overshoot: 0
      Undershoot: 0
            Peak: 7.9998
        PeakTime: 0.2109

ans =
     6

ans =
     6

ans =
     6

ans =
     6

ans =
     6

ans =
     6

ans =
     6

ans =
     6

ans =
     6

ans =
     6

ans =
     6
```





A Plot of the results: Velocity and Postion

We see that the model transfer function is most similar to the transfer function of the motor with inertial load model when sigma is tuned to 50.0 and K is tuned to about 8. This graphs show the position and velocity outputs of the motor transfer function. A PI controller is used to get the step response to have a rise time of .5 ms and overshoot less than 10%.

Published with MATLAB® R2021a