openLoopResponse.m

This script takes reads in time and angular velocity values from arduino code that runs a motor. The values are ploted against a simulink simulation in an open loop configuration to determine an approximate 1st order transfer function. The graph shows that the motor behavior matches the transfer funtion

required file: miniProject_openLoop.slxc, 4.6.ino

Contents

Read in Data from Arduino

port must be set to the communication port used by the Arduino you can find the port by going to 'tools -> Port' in the Arduion application. For a PC, it will be something like COM6

```
port='COM3';
obj = serial(port, 'BaudRate', 115200);
obj.terminator = char(10);
fopen(obj)
%
% do a read to get Ready! from Arduino
%
dummy = fgets(obj);
%
% Read and display some data
%
%for i=1:5,
%    data = fgets(obj);
%    disp(data)
%end;
%
% Read data after sending command to Arduino
%
disp('Starting Counting Event in Arduino')
fprintf(obj,'%s\n','S'); % send start signal to Arduino
data=[];
k=0

% Read Data from Arduino
data = fgets(obj);
% Display what you got
disp(data)
while (~strncmp(data,'Finished',8)) % Until Arduino signals that it is done
    k=k+1;
    % change string data to cell array using tab delimiter
    dataarray = strsplit(data,char(9));
    % save data converting strings to numbers
    Time(k) = eval(dataarray{1});
    velocity(k) = eval(dataarray{2});
    % Read Data from Arduino
    data = fgets(obj);
```

```
    % Display what you got
    disp(data)
end;
fclose(obj)
```

```
Starting Counting Event in Arduino

k =

     0

1000    0.00

1005    0.39

1010    1.18

1015    1.96

1020    2.36

1025    2.36

1030    3.14

1035    3.53

1040    3.53

1045    4.32

1050    4.32

1055    4.32

1060    5.11

1065    5.11

1070    3.93

1075    5.50

1080    5.50

1085    5.50

1090    5.89

1095    6.28

1100    5.89

1105    5.89

1110    5.11

1115    6.28
```

1120	6.28
1125	6.68
1130	6.28
1135	6.68
1140	6.28
1145	6.68
1150	6.68
1155	5.11
1160	6.68
1165	7.07
1170	6.68
1175	6.68
1180	7.07
1185	6.68
1190	7.07
1195	5.50
1200	6.68
1205	7.07
1210	6.68
1215	7.07
1220	6.68
1225	7.07
1230	6.68
1235	7.07
1240	5.50
1245	7.07
1250	6.68
1255	7.07
1260	7.07
1265	6.68
1270	7.46

1275	6.68
1280	5.50
1285	7.07
1290	7.07
1295	7.07
1300	7.07
1305	7.07
1310	7.07
1315	6.68
1320	7.07
1325	5.89
1330	7.07
1335	6.68
1340	7.07
1345	7.07
1350	7.07
1355	7.07
1360	7.07
1366	7.07
1371	7.07
1376	7.07
1381	6.68
1386	7.07
1391	7.07
1396	7.07
1401	7.07
1406	7.07
1411	5.50
1416	7.07
1421	7.07

1426 7.07

1431 6.68

1436 7.07

1441 7.07

1446 7.07

1451 5.50

1456 7.07

1461 7.07

1466 7.07

1471 7.07

1476 6.68

1481 7.07

1486 7.07

1491 7.07

1496 5.50

1501 7.07

1506 7.07

1511 7.07

1516 7.07

1521 7.07

1526 6.68

1531 7.07

1536 5.50

1541 7.07

1546 7.07

1551 7.07

1556 7.07

1561 7.07

1566 7.07

1571 6.68

1576 7.07

1581 5.89

1586 7.07

1591 6.68

1596 7.07

1601 7.07

1606 7.07

1611 7.07

1616 7.07

1622 6.68

1627 7.07

1632 7.07

1637 7.07

1642 7.07

1647 6.68

1652 7.07

1657 7.07

1662 7.07

1667 5.50

1672 7.07

1677 6.68

1682 7.07

1687 7.07

1692 7.07

1697 7.07

1702 6.68

1707 5.89

1712 7.07

1717 7.07

1722 6.68

1727 7.07

1732	7.07
1737	7.07
1742	7.07
1747	7.07
1752	5.50
1757	7.07
1762	7.07
1767	7.07
1772	7.07
1777	7.07
1782	6.68
1787	7.07
1792	5.50
1797	7.07
1802	7.07
1807	7.07
1812	7.07
1817	7.07
1822	7.07
1827	6.68
1832	7.07
1837	5.89
1842	6.68
1847	7.07
1852	7.07
1857	7.07
1862	7.07
1867	6.68
1872	7.46
1878	6.68
1883	7.07

```
1888    7.07

1893    7.07

1898    7.07

1903    7.07

1908    7.07

1913    6.68

1918    7.07

1923    5.50

1928    7.46

1933    6.68

1938    7.07

1943    7.07

1948    7.07

1953    7.07

1958    7.07

1963    5.50

1968    7.07

1973    6.68

1978    7.07

1983    7.07

1988    7.07

1993    7.07

1998    7.07

Finished
```

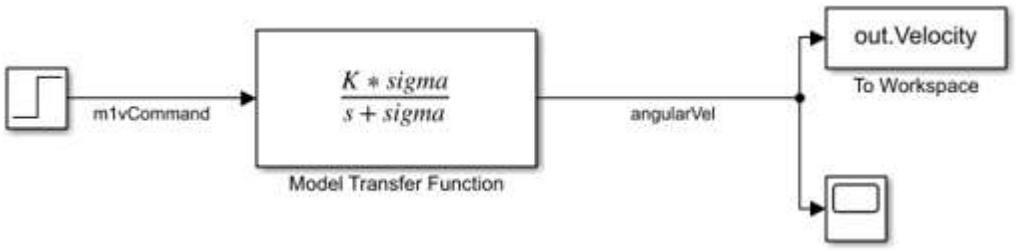
Set Up Transfer Function

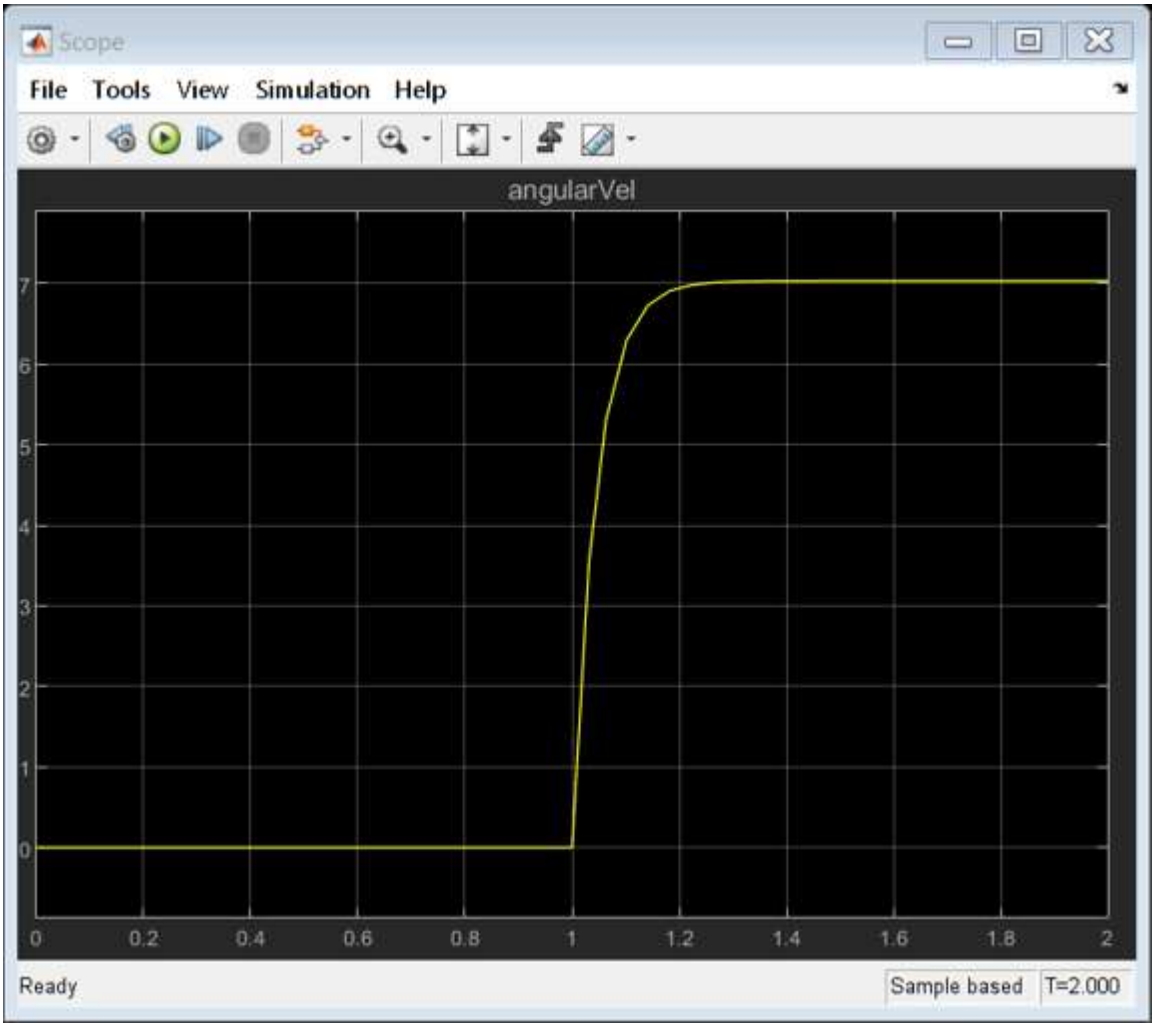
```
%Tansfer function values
K = 1.9;
sigma = 22.22;

%Final voltage
V = 3.7;
```

Run Simulink Simulation

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

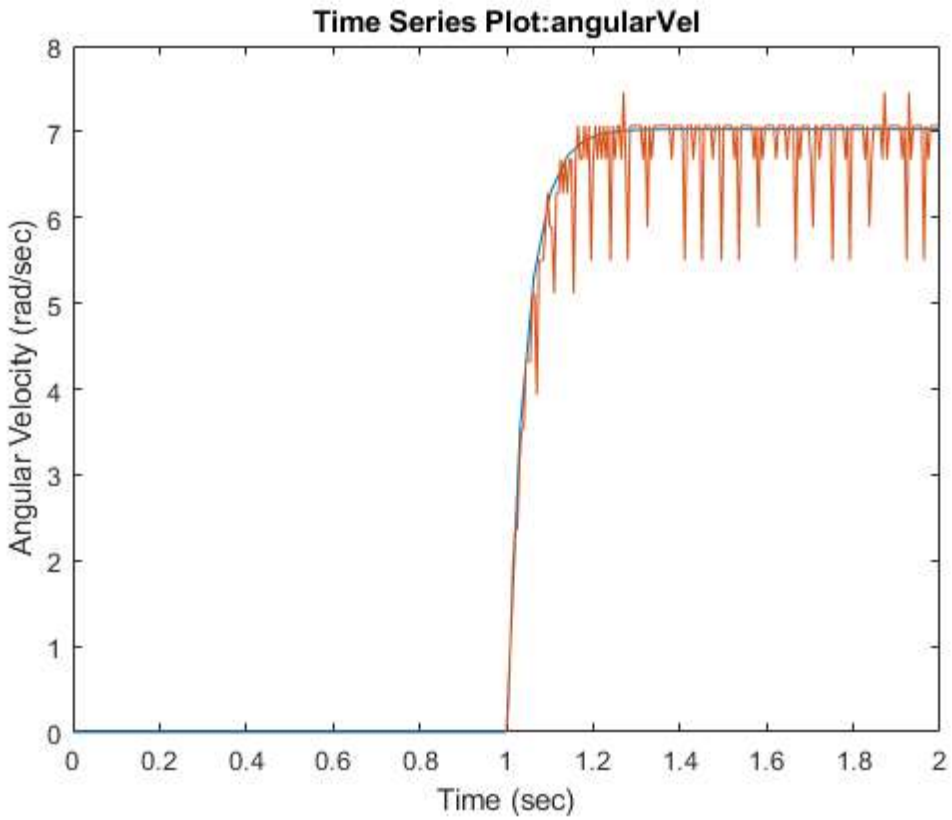




A Plot of the results

```
figure
plot(out.Velocity)
hold on
plot(Time / 1000,velocity)
hold off
xlabel('Time (sec)')
```

```
ylabel('Angular Velocity (rad/sec)')
%
% Save results in a .mat file to use later
%
save somedata.mat Time velocity
```



Published with MATLAB® R2021b