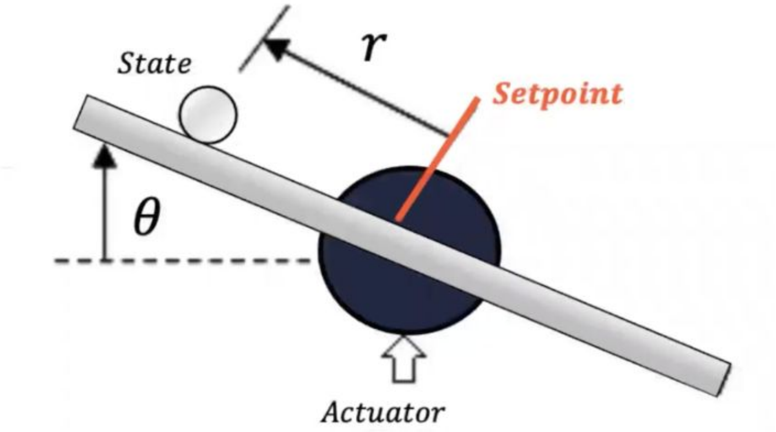
Ball and Beam Balance Control System

Ameer Alwadiya

Monday, April 18th, 2024

Objectives

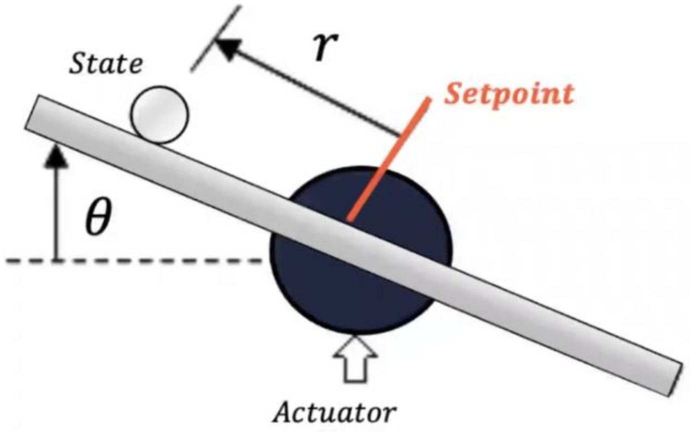
The objective of this project is to stabilise and control the position of a ball on a beam, as well as to design a suitable controller and model for the system.



1 Plant Modeling

$$\begin{bmatrix}\dot{\theta}\\ \ddot{\theta}\\ r\\ \dot{r}\end{bmatrix} = \begin{bmatrix}0 & 1 & 0 & 0\\ 0 & 0 & -\frac{m\cdot g}{\frac{J}{R^2}+m} & 0\\ 0 & 0 & 0 & 1\\ 0 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}\theta\\ \dot{\theta}\\ r\\ \dot{r}\end{bmatrix} + \begin{bmatrix}0\\ 0\\ 0\\ 1\end{bmatrix} u$$

$$r = \begin{bmatrix}1 & 0 & 0 & 0\end{bmatrix}\begin{bmatrix}\theta\\ \dot{\theta}\\ r\\ \dot{r}\end{bmatrix} + \begin{bmatrix}0\end{bmatrix} u$$



θ: The position of the beam.

$\dot{\theta}$: The velocity of the beam.

r: The position of the ball.

$\dot{r}$: The velocity of the ball.

u: The input applied to the system.

m: The mass of the ball.

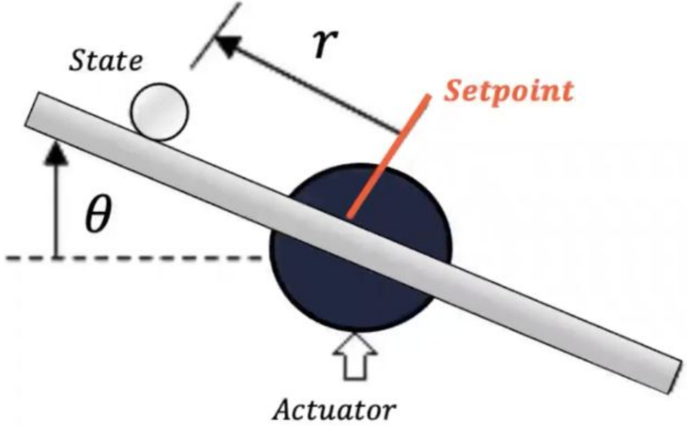
R: The radius of the ball.

g: The acceleration due to gravity.

J: The moment of inertia of the ball.

1 Plant Modeling

Transfer Function
$P(s) = \frac{R(s)}{\Theta(s)} = -\frac{mg}{\left(\frac{J}{R^2}+m\right)}\frac{1}{s^2}\ \left[\frac{m}{rad}\right]$

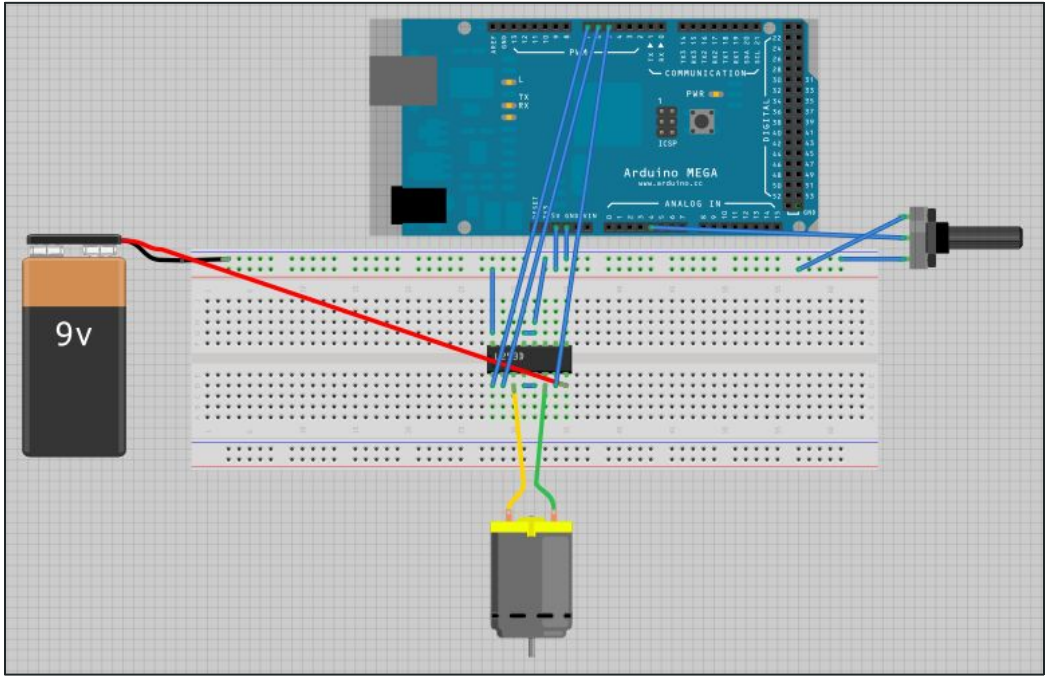


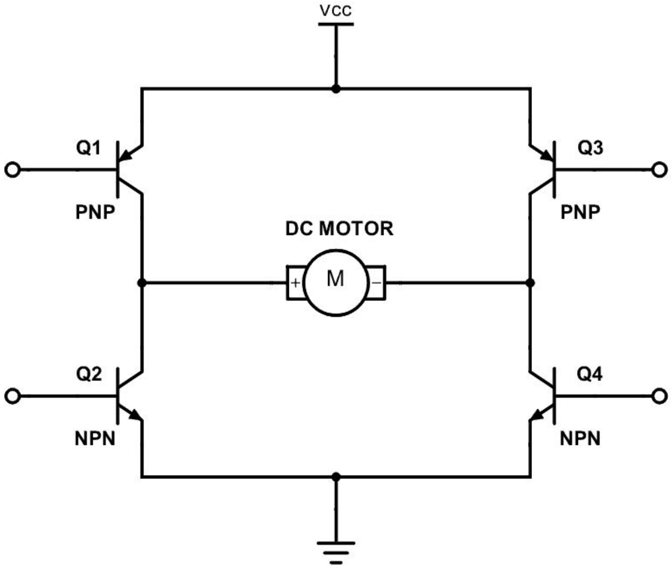
1 Plant Modeling

```
%% plant transfer function
s = tf('s');

% Given parameters
m = 0.15;
R = 0.01;
g = -9.8;
J = 9.99e-6;

% Define the symbolic expression
tf_plant = -(m*g /(J/R^2 + m))*1/(s^2);
```

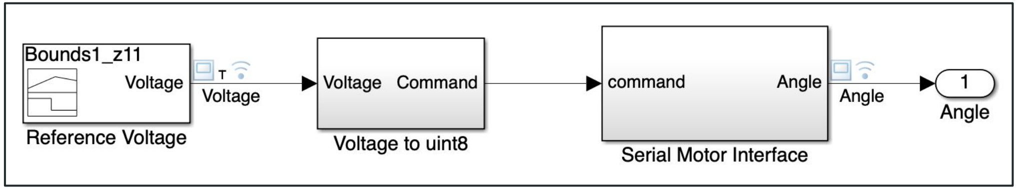
```
%% plant state space
H = -m*g/(J/(R^2)+m);

A = [0 1 0 0
     0 0 H 0
     0 0 0 1
     0 0 0 0];
B = [0;0;0;1];
C = [1 0 0 0];
D = [0];

ss_plant = ss(A,B,C,D);
```

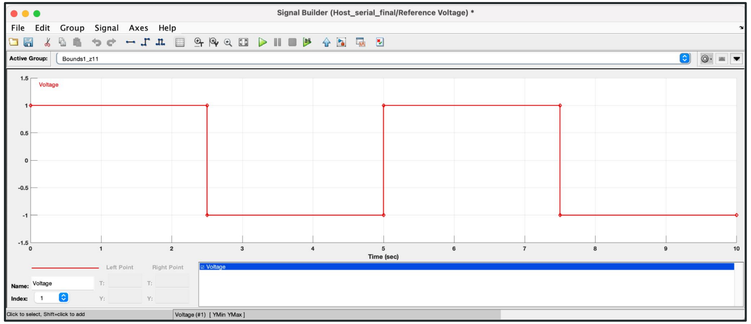
2 DC Motor Modelling

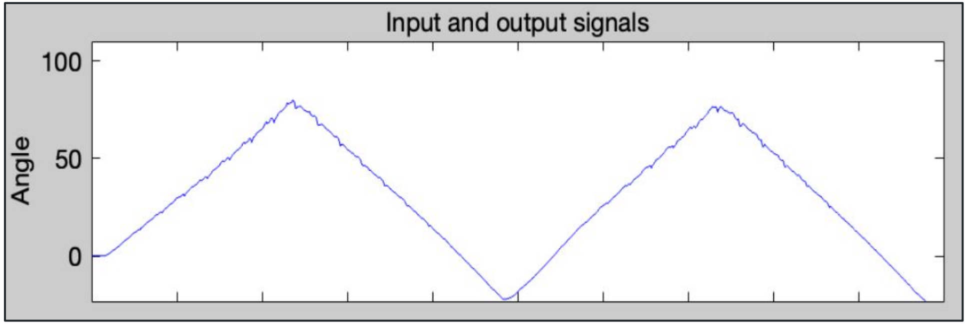




2 DC Motor Modelling

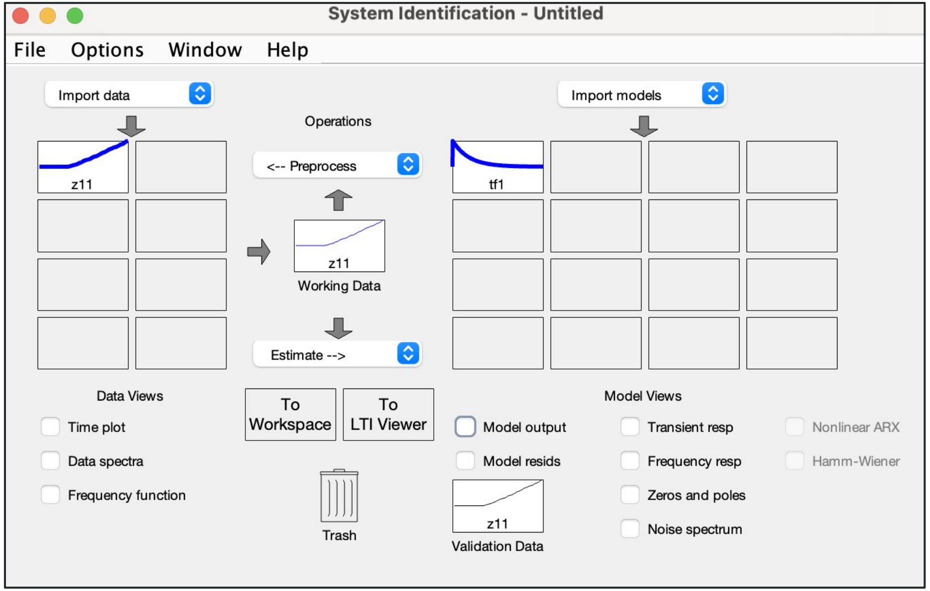


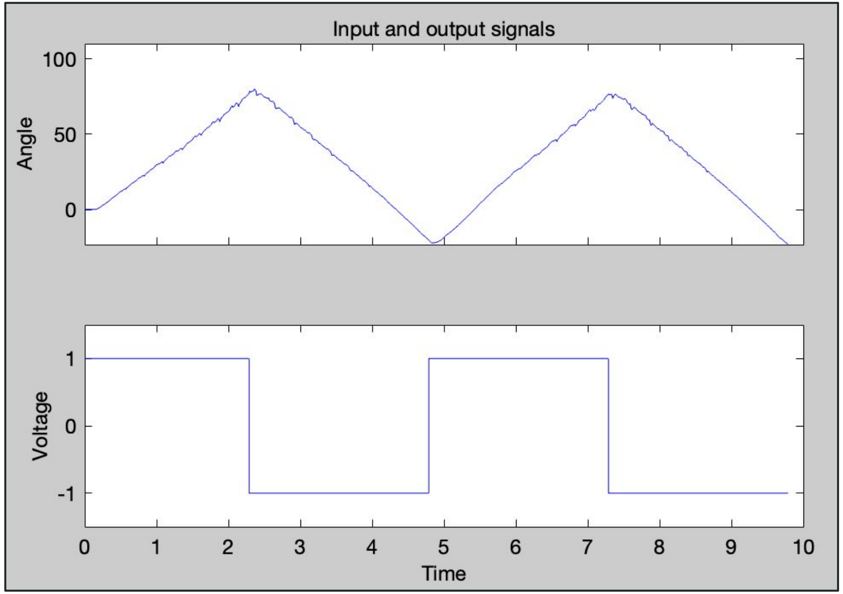




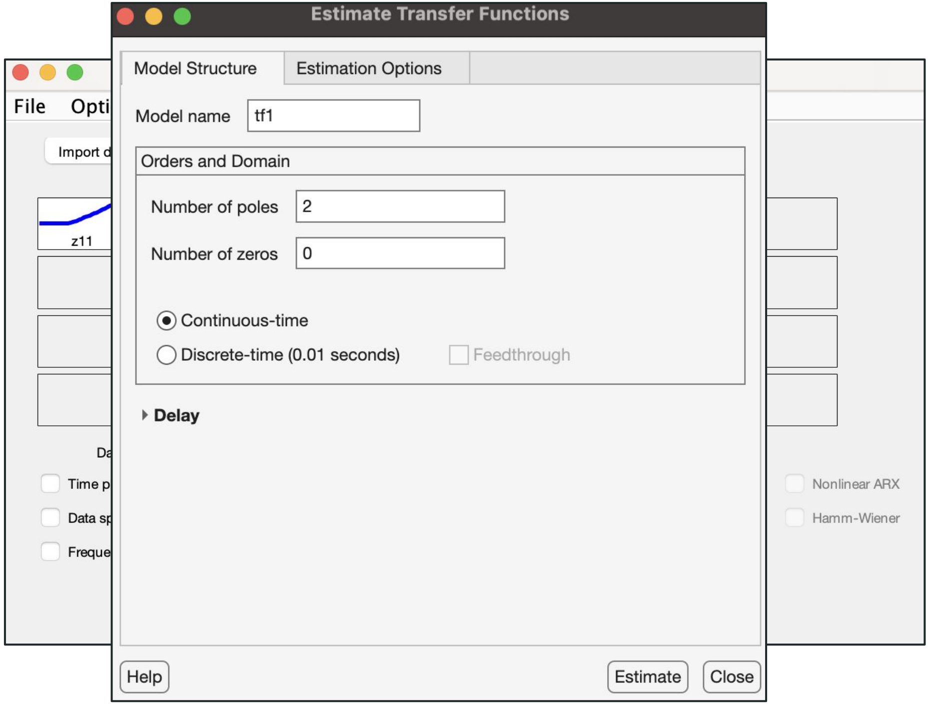
2 DC Motor Modelling

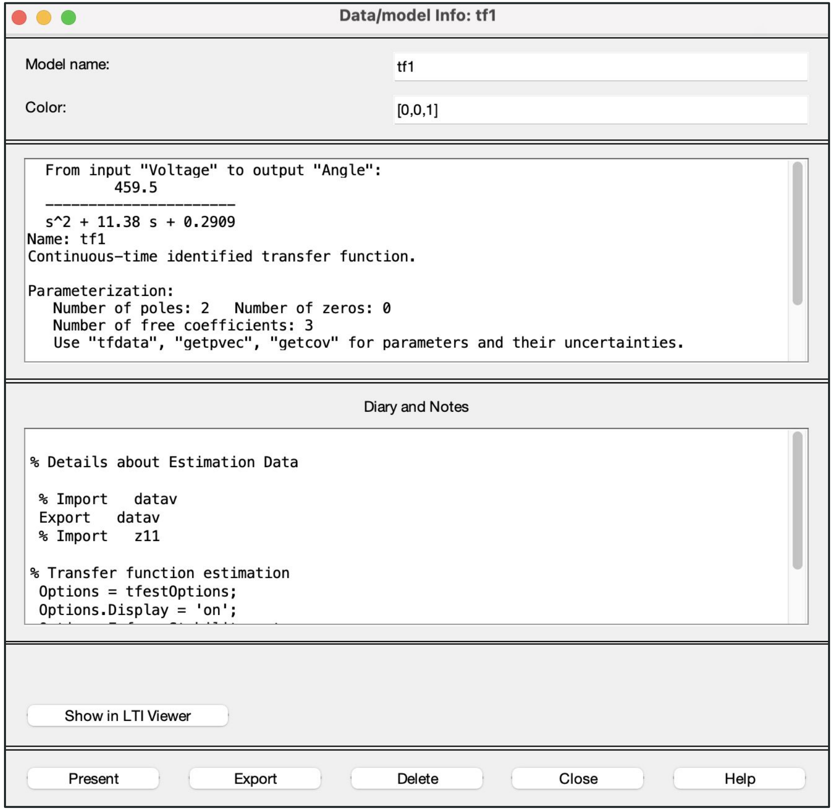
```
ident;
```



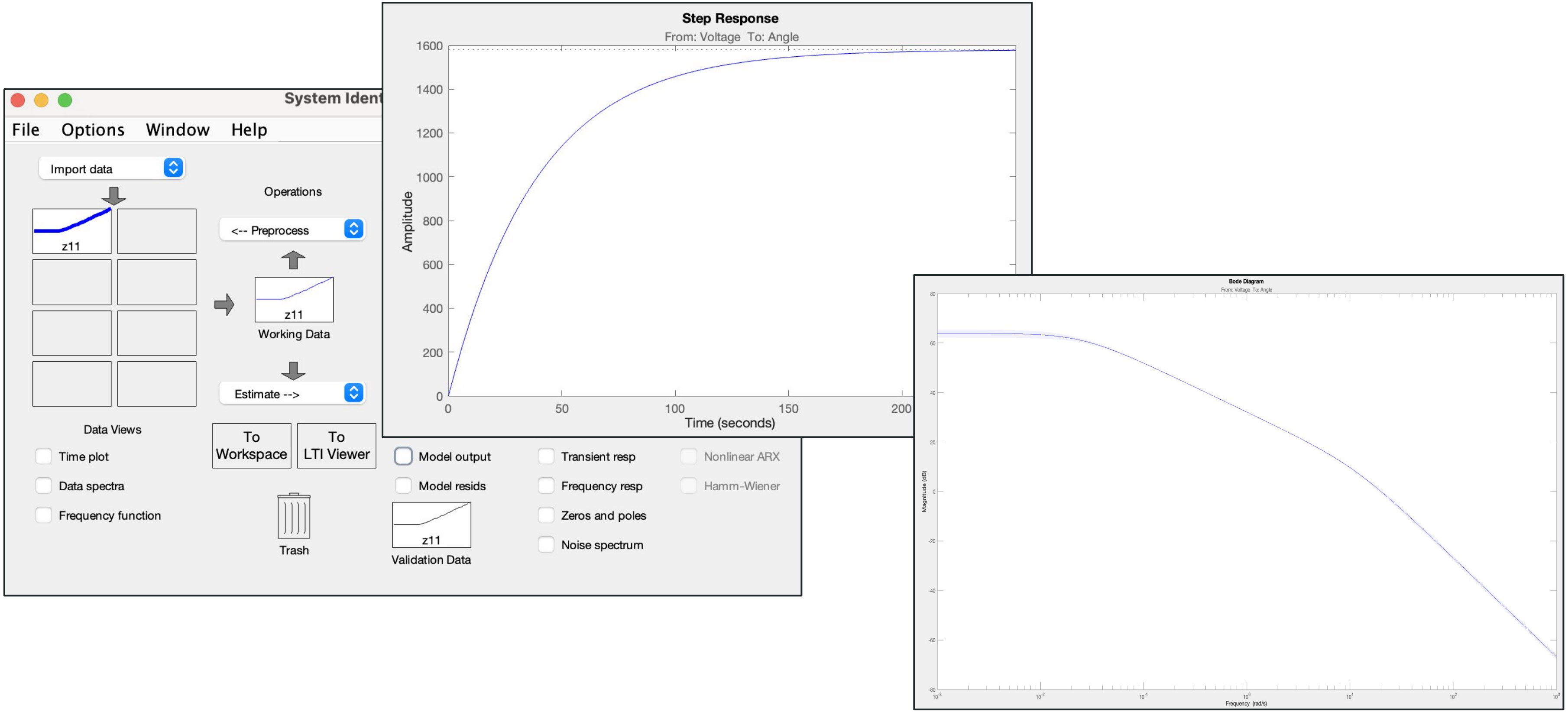


2 DC Motor Modelling

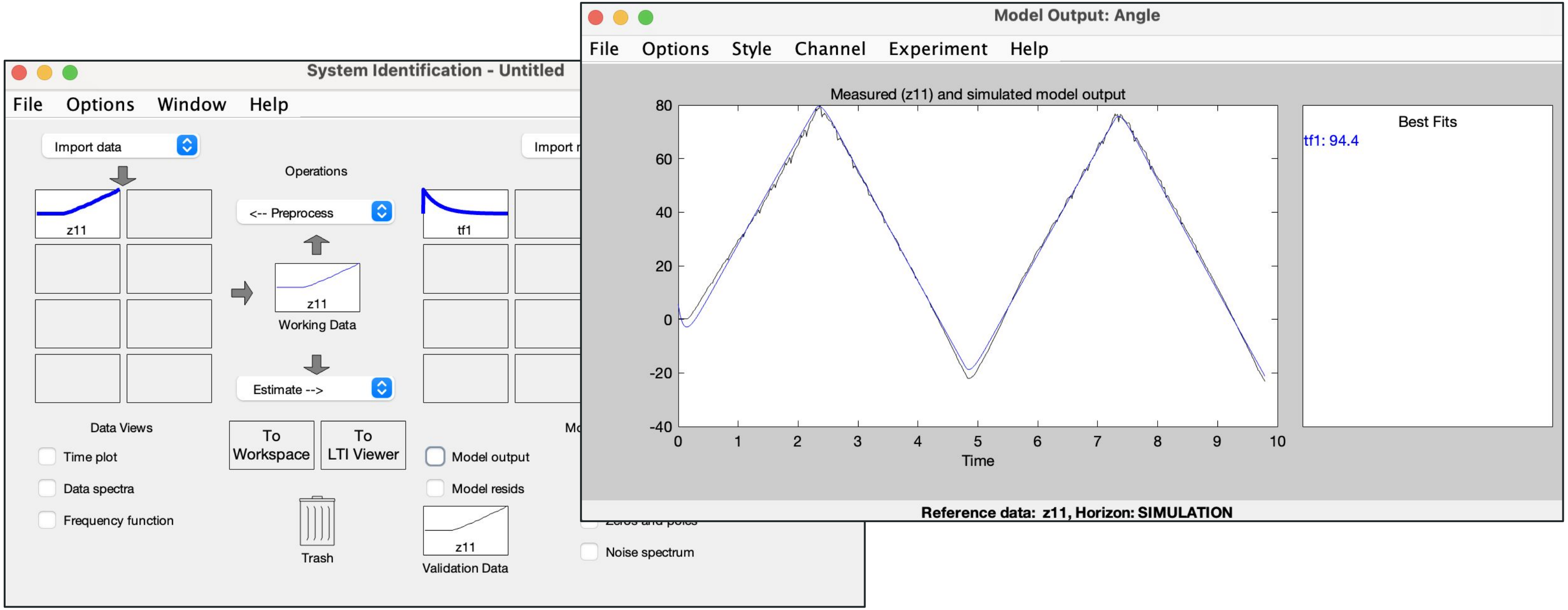




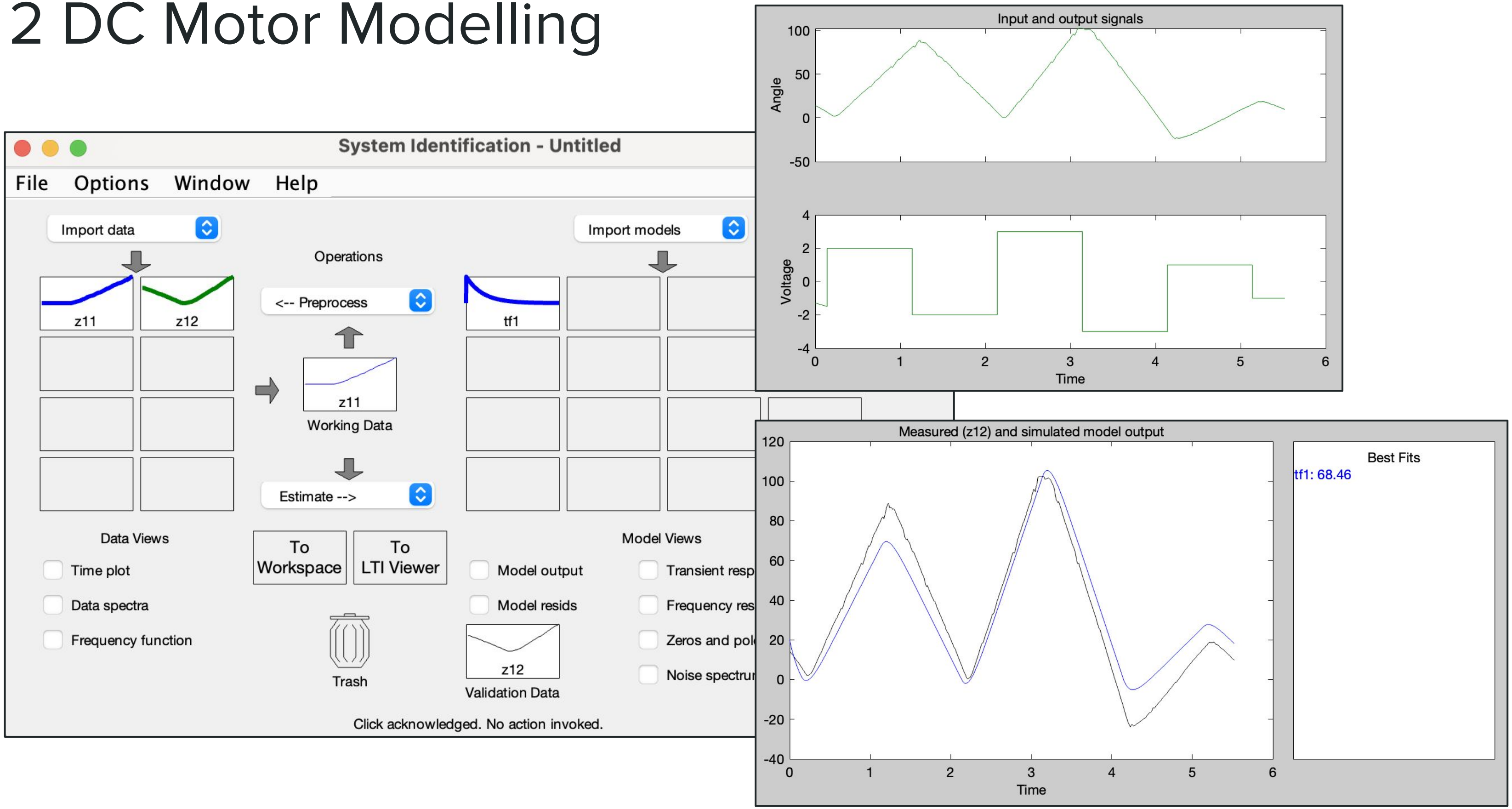
2 DC Motor Modelling



2 DC Motor Modelling



2 DC Motor Modelling

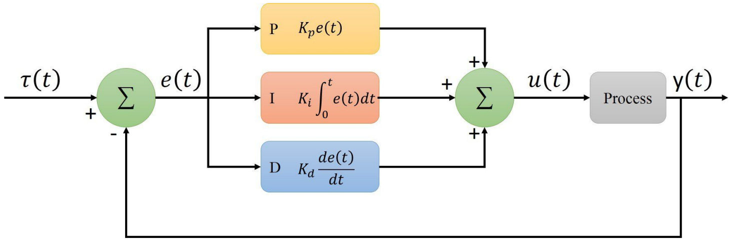


Designing the Controller

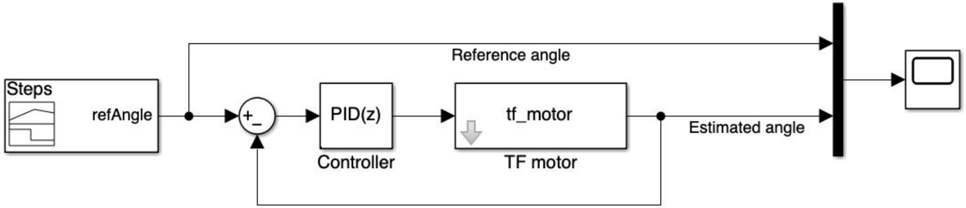
```
C = pidtune(tf_motor, 'pidf', 6);
Cd = c2d(C, 0.01);
```

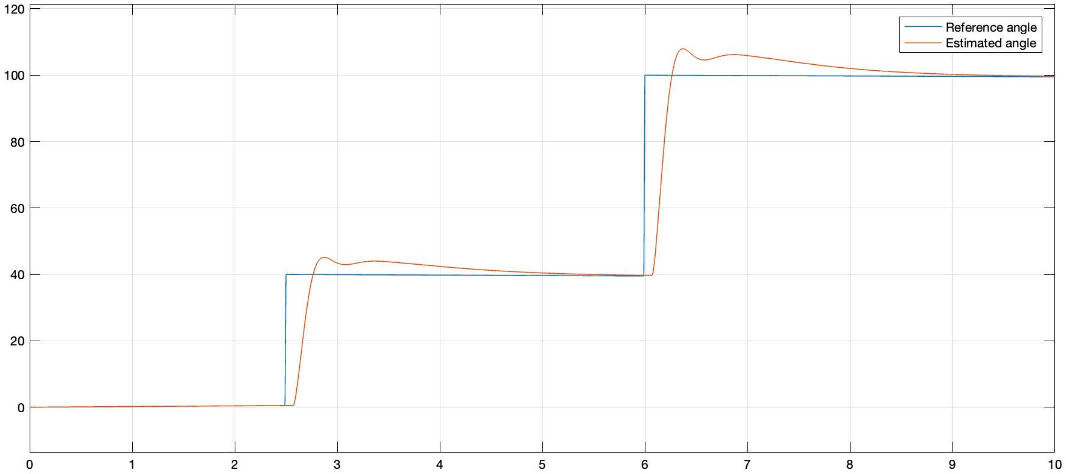
```
C =

                 1             s
  Kp + Ki * --- + Kd * --------
                 s          Tf*s+1

  with Kp = 0.166, Ki = 0.173, Kd = 0.00802, Tf = 0.00146

Continuous-time PIDF controller in parallel form.
```

```
Cd =

                 Ts                1
  Kp + Ki * ------ + Kd * -----------
                z-1           Tf+Ts/(z-1)

  with Kp = 0.166, Ki = 0.173, Kd = 0.0551, Tf = 0.01, Ts = 0.01

Sample time: 0.01 seconds
Discrete-time PIDF controller in parallel form.
```

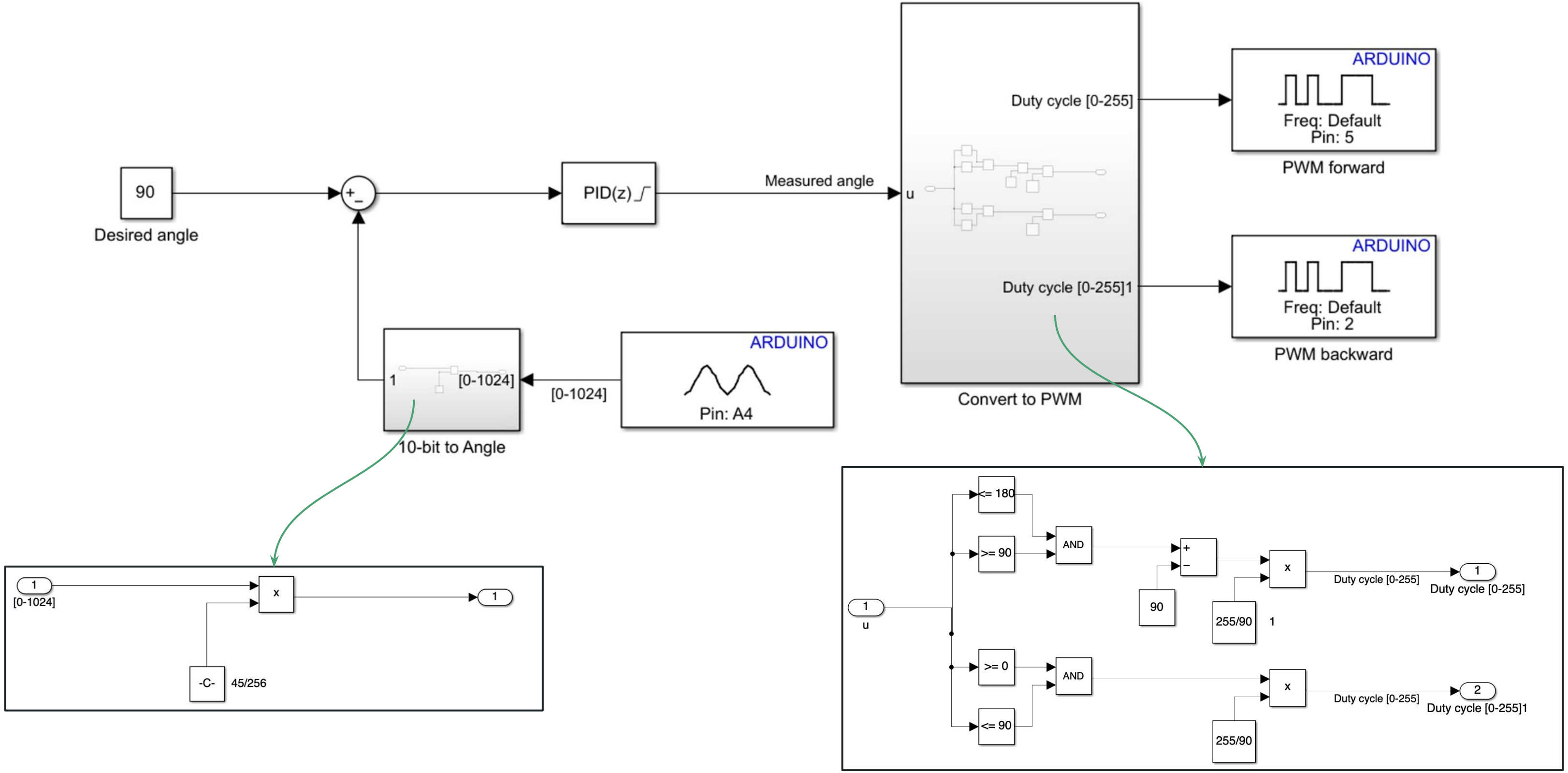


Testing the Controller on Simulink

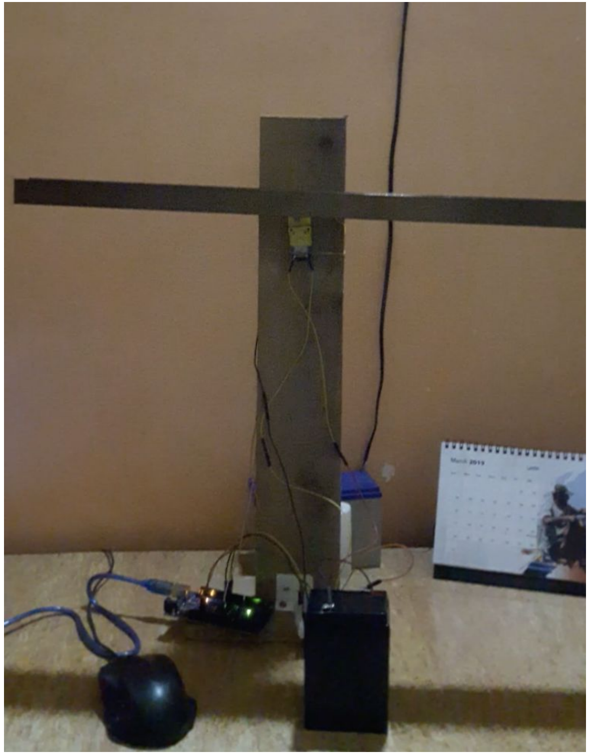


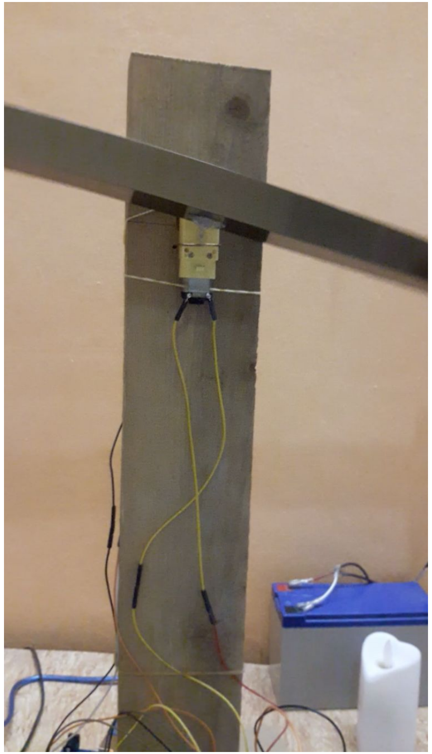


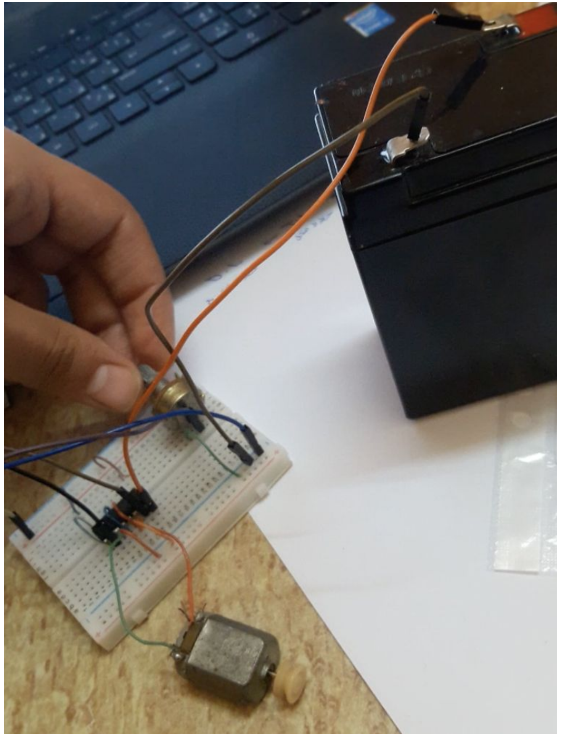
Testing the Controller on Hardware



Testing the Controller on Hardware







Results and Further developments

- Estimate nonlinear dynamics to get rid of oscillating problem.
- Install a second sensor for controlling the position of the ball like ultrasonic sensors because the potentiometer alone is not very accurate in measure the angle.
- Further develop simulink model and integrate with research.
- Implementing methods for predicting ball position under uncertainty using Kalman filter.

Any Question?

Thank You!