MATLAB- Simulink

Kun-Yen Chiu

Example 3.3 Inverted pendulum control

Step 1. Used MATLAB to transfer the transfer function from state space

System matrix:

$$\mathbf{A} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -mg/M & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & g/l & 0 \end{bmatrix}, \quad \mathbf{B} = \begin{bmatrix} 0 \\ 1/M \\ 0 \\ -1/(Ml) \end{bmatrix}.$$

$$\dot{\mathbf{x}}(t) = \mathbf{A}\mathbf{x}(t) + \mathbf{B}u(t)$$

$$y(t) = \mathbf{C}\mathbf{x}(t) + \mathbf{D}u(t)$$

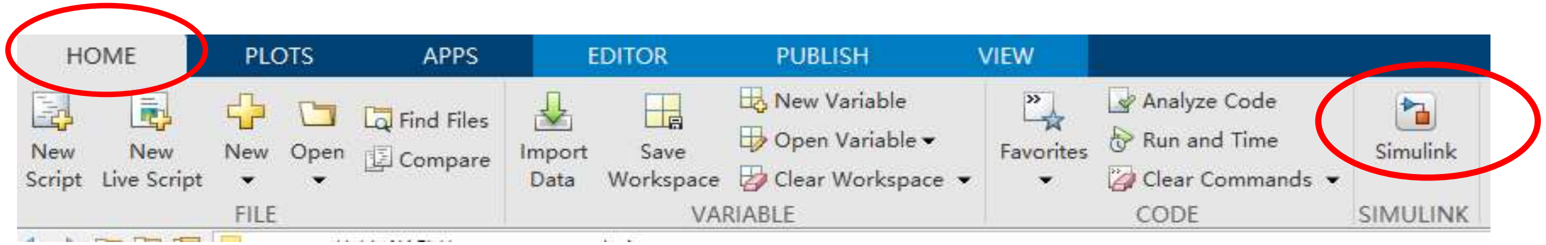
```
%% Problem 1=> calculate transfer function for C=[0 0 1 0];
clc
%inital parameter
g=9.8;
l=0.5;
m=0.01;
M=2;

%system matrix
A=[0 1 0 0 ; 0 0 -m*g/M 0;0 0 0 1; 0 0 g/l 0 ];
B=[0 ;1/M;0;-1/(M*l)];
C=[0 0 1 0];
D=[0];

[num,den] = ss2tf(A,B,C,D)  %ss2tf: state space --> transfer function
```

Command Window

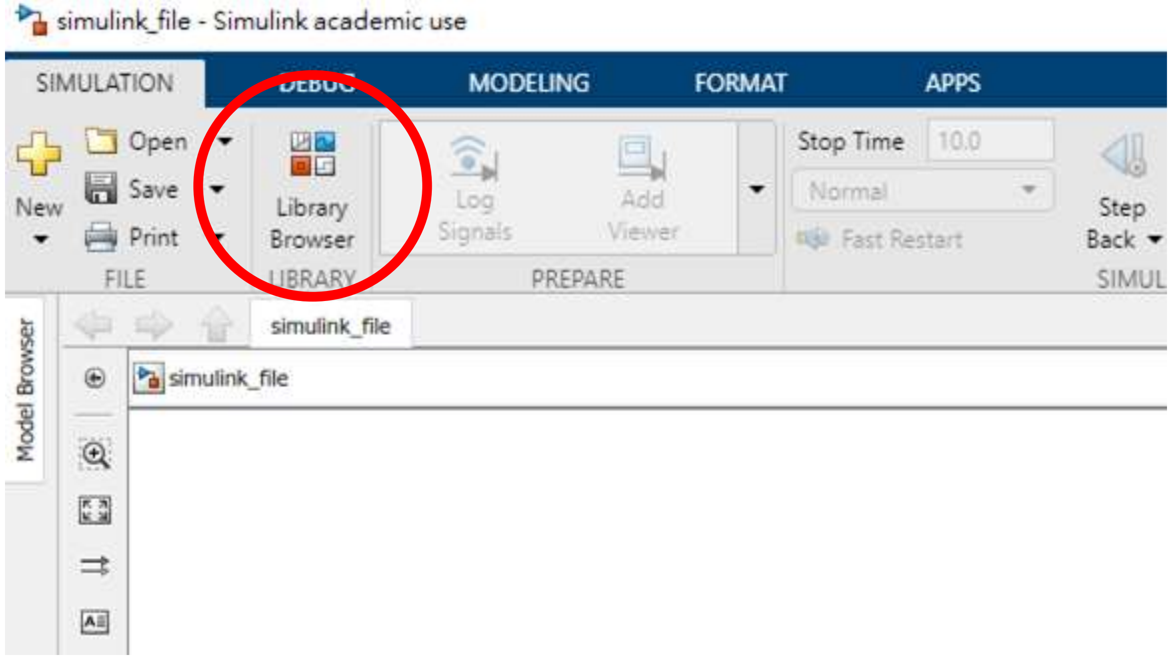
```
num =

         0         0   -1.0000    0.0000    0.0000


den =

    1.0000         0  -19.6000         0         0
```

Step 2. Open Simulink

Home Simulink Blank Model (create new model)save it!!



Once you save the model,
you can search the function blocks from library Browser

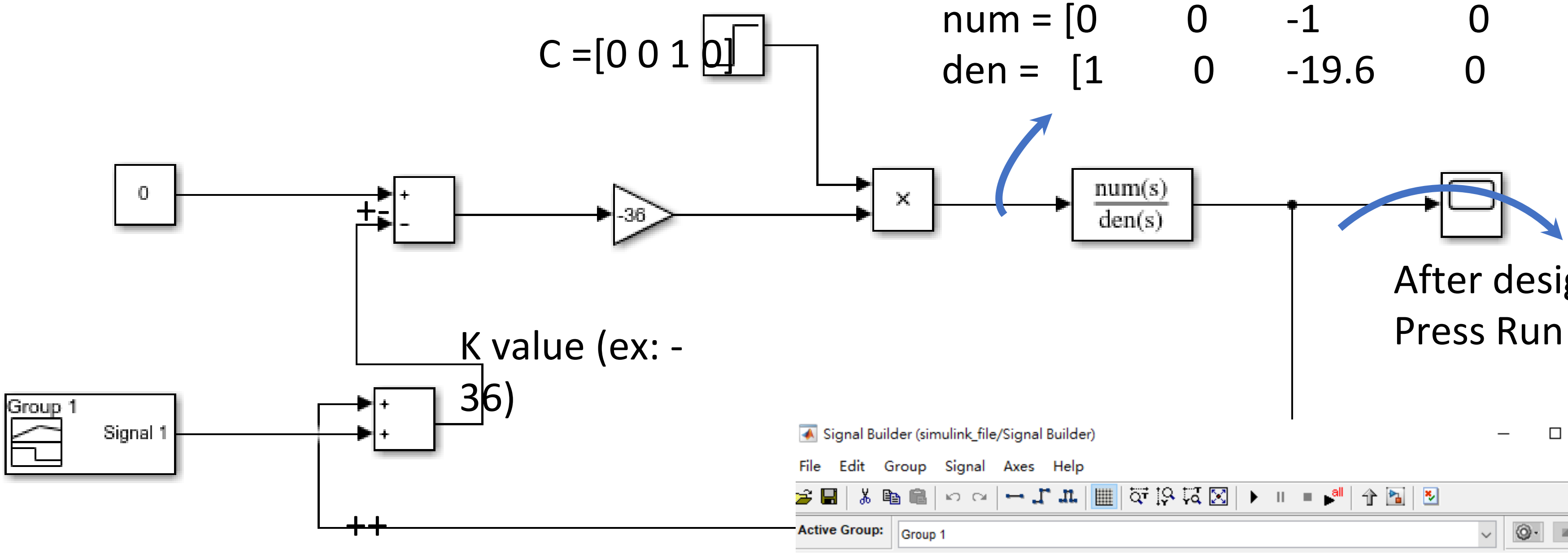


You will need the following function blocks:
Signal Builder
Add
Constant
Product
Gain
Step
Transfer Fcn
Scope (see output result)

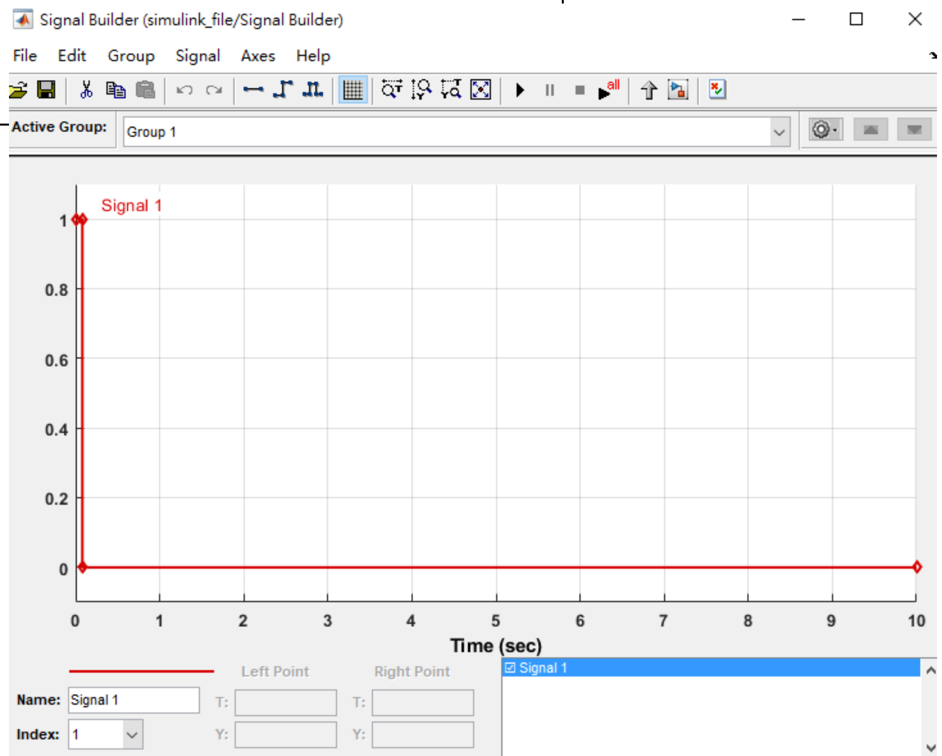
Step 3. Create Block Diagram

num = [0 0 -1 0 0]

den = [1 0 -19.6 0 0]



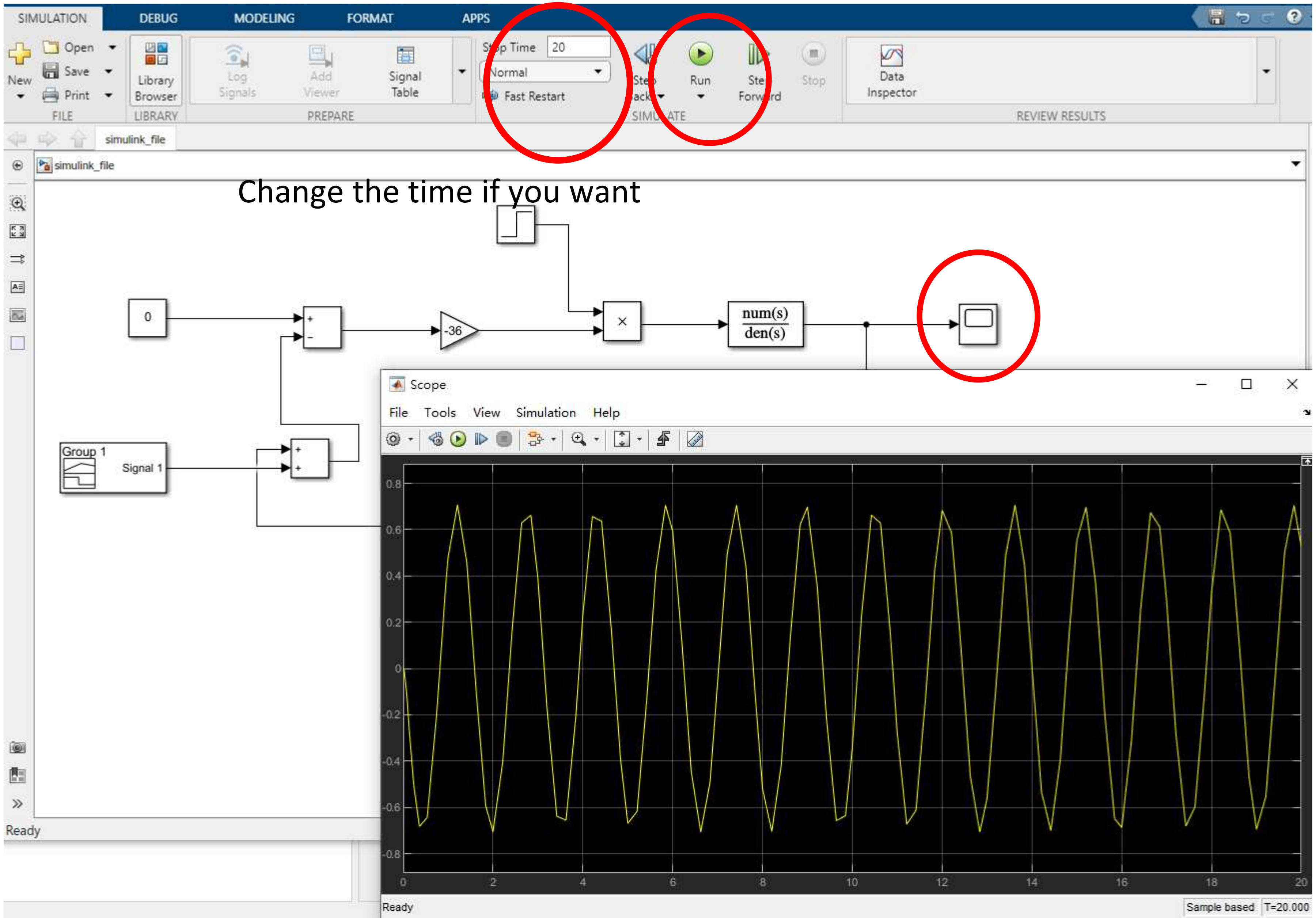
After designing the block diagram,
Press Run and click the scope to see the result



For Signal Builder
You can set
for t= 0~0.1
y=1
else
y=0

to simulate initial condition

Output Result



Problem 2: ODE45 Reference

ODE Solvers: Standard Syntax

- To use standard options and variable time step
  - [t,y]=ode45('myODE',[0,10],[1;0])

ODE integrator:
23, 45, 15s

ODE function

Time range

Initial conditions

- Inputs:
  - ODE function name (or anonymous function). This function takes inputs (t,y), and returns dy/dt
  - Time interval: 2-element vector specifying initial and final time
  - Initial conditions: column vector with an initial condition for each ODE. This is the first input to the ODE function
- Outputs:
  - t contains the time points
  - y contains the corresponding values of the integrated variables.

More info:
https://www.mathworks.com/help/matlab/ref/ode45.html

Problem 2 reference

```
%% Problem 2
clc
 tspan = [0 10];   %time interval from 0 - 10
 iniCon = [0;0;0;0];  %initial condition
 [t, y] = ode45(@sys, tspan, iniCon)
 y1=y(:, 1) % y
 y2=y(:, 2) % y'
 y3=y(:, 3) % angle
 y4=y(:, 4) % angle'
 %%plot(t, xxxxxxxx Plot it Yourself xxxxxxxx)
 pulse= rectangularPulse(0,0.1,t);
function  dx = sys(t, x)
%initial parameter
   g=9.8;
   l=0.5;
   m=0.01;
   M=2;
  pulse = rectangularPulse(0,0.1,t);
  A=[0 1 0 0 ; 0 0 -m*g/M 0;0 0 0 1; 0 0 g/l 0 ];  %system matrix
  B=[0 ;1/M;0;-1/(M*l)];
  k=-100;          % Try different value of K
  C=[0 0 1 0]*x; % here can change the value of C
  Gc=(0-(C+pulse))*k;
  u = Gc*heaviside(t);
   dx = A*x + B*u;
end
```