KATHMANDU UNIVERSITY **School of Engineering**
Department of Electrical & Electronics Engineering

CONTROL ENGINEERING LABORATORY EXERCISE-I

1. Introduction to MATLAB

1.1 Eigen values

If ***A*** is an n x n matrix, the n numbers λ that satisfy ***Ax*** = λ***x*** are the eigen-values of ***A***. They are found using ***eig(A),*** which returns the eigen-values in a column vector. Eigen-values and eigen-vectors can be obtained with a double assignment statement ***[X, D] = eig(A)***. The diagonal elements of ***D*** are the eigen-values and the columns of ***X*** are the corresponding eigen-vectors such that ***AX = XD***.

Example-1.1.1

Find the eigen values and eigen vectors of the matrix a given by

$$A = \begin{bmatrix} 0 & 1 & -1 \\ -6 & -11 & 6 \\ -6 & -11 & 5 \end{bmatrix}$$

```
>> A = [ 0 1 -1; -6 -11 6; -6 -11 5];
>> [X,D]=eig(A)

X =                                      D =
   0.7071   -0.2182   -0.0921              -1.0000         0         0
   0.0000   -0.4364   -0.5523                    0   -2.0000         0
   0.7071   -0.8729   -0.8285                    0         0   -3.0000
```

1.2 Complex Numbers

Example-1.2.1

In the circuit shown in figure 1.2.1, determine the node voltages V_1 and V_2 and the power delivered by each source.

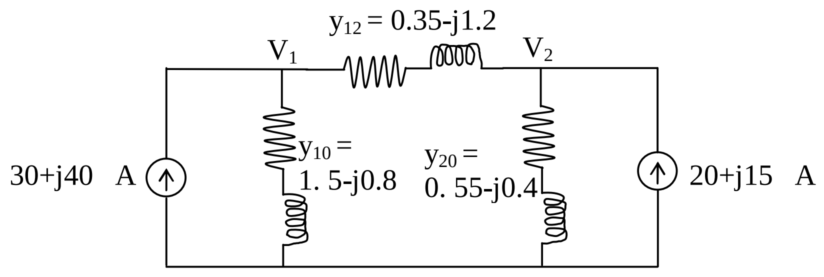


Figure-1.2.1

Krichhoff's current law results in the following matrix node equation.

$$\begin{bmatrix} 1.5 - j2.0 & -0.35 + j1.2 \\ -0.35 + j1.2 & 0.9 - j1.6 \end{bmatrix} \begin{bmatrix} V_1 \\ V_2 \end{bmatrix} = \begin{bmatrix} 30 + j40 \\ 20 + j15 \end{bmatrix}$$

And the complex power of each source is given by $S = VI^*$. The following program, is written to yield solutions to V_1, V_2 and S.

```
>> j=sqrt(-1);
>> I=[30+j*40; 20+j*15];
>> Y=[1.5-j*2 -0.35+j*1.2; -0.35+j*1.2 0.9-j*1.6];
>> V=inv(Y)*I
V =
   3.5902 +35.0928i
   6.0155 +36.2212i
>> S=V.*conj(I)
S =
  1.0e+003 *
                        1.5114 + 0.9092i
                         0.6636 + 0.6342i
```

1.3 Roots of polynomial

Example-1.3.1

Find the roots of the following polynomial.

$$s^6 + 9s^5 + 31.25s^4 + 61.25s^3 + 67.75s^2 + 14.75s + 15$$

The roots are found using **roots.**

```
>> p=[1 9 31.25 61.25 67.75 14.75 15];
>> r=roots(p)
r =
  -4.0000
  -3.0000
  -1.0000 + 2.0000i
  -1.0000 - 2.0000i
   0.0000 + 0.5000i
   0.0000 - 0.5000i
```

Example-1.3.2

The roots of a polynomial are -1, -2 and -3±j4. Determine the polynomial equation.

Complex numbers may be entered suing function I or j. The roots are then entered in a column vector. The polynomial equation is obtained using **poly** as follows

```
>> i=sqrt(-1);
>> r=[-1 -2 -3+4*i -3-4*i];
>> p=poly(r)
p =
    1       9    45    87    50
```

1.4 Transfer Function

Example- 1.4.1

```
>> num=[1 4];
>> den=[1 2 10];
>> sys=tf(num,den)

sys =
               s + 4
           --------------
          s^2 + 2 s + 10
```

Example- 1.4.2

```
>> num=[1 11 30 0];
>> den=[1 9 45 87 50];
>> [z,p,k]=tf2zp(num,den)

z =                                  p =
        0                              -3.0000 + 4.0000i
   -6.0000                             -3.0000 - 4.0000i          k =
   -5.0000                             -2.0000                        1
                                       -1.0000
```

Example-1.4.3

```
>> z=[-6;-5;0];
>> k=1;
>> i=sqrt(-1);
>> p=[-3+4*i;-3-4*i;-2;-1];
>> [num,den]=zp2tf(z,p,k)

num =                                          den =
    0     1    11    30     0                      1       9    45    87    50
```

1.5 State-Space Representation

Example-1.5.1

```
>> num=[1 7 2]; den=[1 9 26 24];
>> [A B C D]=tf2ss(num,den)
```

```
A =                       B =                       C =                       D =
    -9   -26   -24              1                       1     7     2               0
     1     0     0              0
     0     1     0              0
```

Example-1.5.2

A system is described by the following state-space equations

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{bmatrix} = \begin{bmatrix}0 & 1 & 0\\ 0 & 1 & 1\\ -1 & -2 & -3\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_3\end{bmatrix} + \begin{bmatrix}10\\ 0\\ 0\end{bmatrix}u$$

$$y = \begin{bmatrix}1 & 0 & 0\end{bmatrix}x$$

Find the transfer function $G(s) = \frac{Y(s)}{X(s)}$.

```
>> A=[0 1 0; 0 0 1; -1 -2 -3];
>> B=[10; 0; 0];
>> C=[1 0 0]; D=[0];
>> [num,den]=ss2tf(A,B,C,D);
>> sys=tf(num,den)

Transfer function:
 10 s^2 + 30 s + 20
---------------------
s^3 + 3 s^2 + 2 s + 1
```

2. Mathematical Modeling

2.1 Modeling a system

Example-2.1.1

Consider the simple mechanical system as shown in the figure-2.1.1 below. Three forces influence the motion of the mass, namely, the applied force, the frictional force, and the spring force.

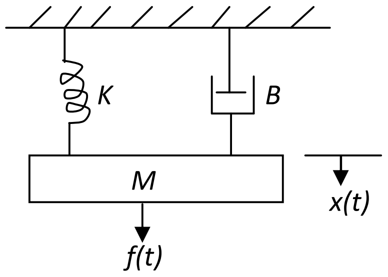


Figure-2.1.1

Applying Newton's law of motion, the force equation of the system is

$$M\frac{d^2x}{dt^2} + B\frac{dx}{dt} + Kx = f(t)$$

Let $x_1 = x$ and $x_2 = \frac{dx}{dt}$, then

$$\frac{dx_1}{dt} = x_2$$

$$\frac{dx_2}{dt} = \frac{1}{M}[f(t) - Bx_2 - Kx_1]$$

With the system initially at rest, a force of 25 Newton is applied at time t = 0. Assume that the mass M = 1 Kg, frictional coefficient B = 5 N/m/sec., and the spring constant K = 25 N/m.

The above equations are defined in an M-file **mechsys.m** as follows:

```
function  xdot=mechsys(t,x);
F=25;
M=1; B=5; K=25;
xdot=[x(2);1/M*(F-B*x(2)-K*x(1))];
```

Save the above instructions as **mechsys.m** and execute the following instructions:

```
>> tspan=[0,3];
>> x0=[0,0];
>> [t,x]=ode23('mechsys',tspan,x0);
>> subplot(2,1,1),plot(t,x)
>> title('Time response of mechanical translational system')
>> xlabel('Time-sec')
>> text(2.5,1.2,'displacement')
>> text(2,.2,'velocity')
```

The result of the simulation is as shown in figure 2.1.2:

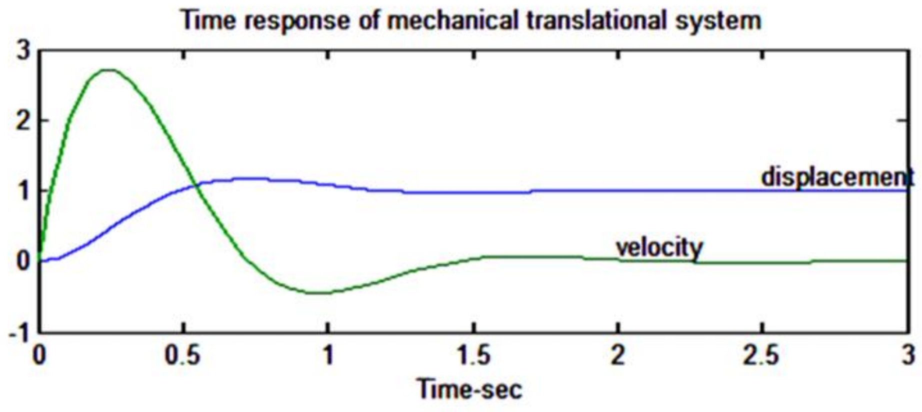


Figure-2.1.2

Example-2.1.2

The circuit elements in the figure 2.1.3 are R = 1.4Ω, L = 2H, and C = 0.32F, the initial inductor current is zero, and the initial capacitor voltage is 0.5 volts. A step voltage of 1 volt is applied at time t = 0. Determine i(t) and $v_c(t)$ over the range 0 < t < 15 sec. Also, obtain a plot of current versus capacitor voltage.

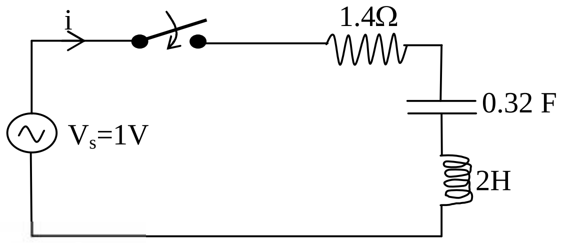


Figure-2.1.3

Applying KVL

$$Ri + L\frac{di}{dt} + v_c = V_s$$

and $$i = C\frac{dv_c}{dt}$$

Let $$x_1 = v_c$$

and $$x_2 = i$$

then $$\dot{x}_1 = \frac{1}{C}x_2$$

and $$\dot{x}_2 = \frac{1}{L}(V_s - x_1 - Rx_2)$$

The above equations are defined in an M-file electsys.m as follows:

```
function  xdot=electsys(t,x);
V=1;
R=1.4; L=2; C=0.32;
xdot=[x(2)/C;1/L*(V-x(1)-R*x(2))];
```

Save the above instructions as **electsys.m** and execute the following instructions:

```
x0=[0.5,0];
tspan=[0,15];
[t,x]=ode23('electsys',tspan,x0);
subplot(2,1,1),plot(t,x)
title('Time response of an RLC series circuit')
xlabel('Time-sec.')
text(8,1.15,'capacitor voltage')
text(8,.1,'current')
```

Result of simulation is as shown in figure 2.1.4:

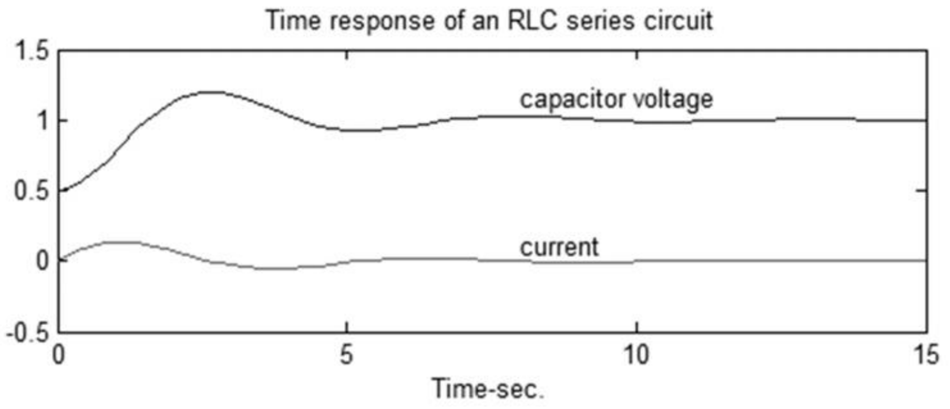


Figure-2.1.4

2.2 System Modeling using MATLAB-simulink

2.2.1 Free body diagram and system equation

Example-2.2.1

Consider the first-order model of the motion of a car. Assume the car to be travelling on a flat road. The horizontal forces acting on the car can be represented as shown in the figure 2.2.1.

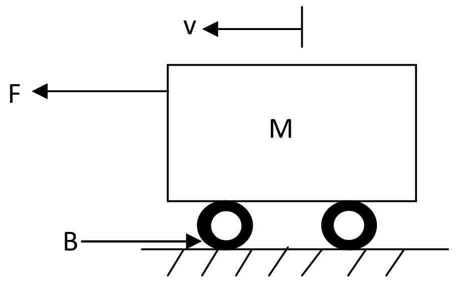


Figure-2.2.1

In figure-2.2.1

- v is the horizontal velocity of the car (units of m/s).
- F is the force created by the car's engine to propel it forward (units of N).
- b is the damping coefficient for the car, which is dependent on wind resistance, wheel friction, etc. (units of N*s/m).
- M is the mass of the car (units of kg).

The differential equation representing the system is

$$M\frac{dv}{dt} = F - bv$$

Assume that:
M = 1000 kg and b = 40 N*sec/m

This system will be modeled in Simulink by using the system equation as above. This equation indicates that the car's acceleration (dv/dt) is equal to the sum of the forces acting on the car (F-bv) divided by the car's mass,

$$\frac{dv}{dt} = \frac{F - bv}{M} = \frac{F - 40v}{1000}$$

To model this equation, insert a Sum block, two Gain blocks and an integrator block into a new model window. Change the parameters of the blocks as per the requirement. Connect the blocks with lines as shown in the figure 2.2.2.

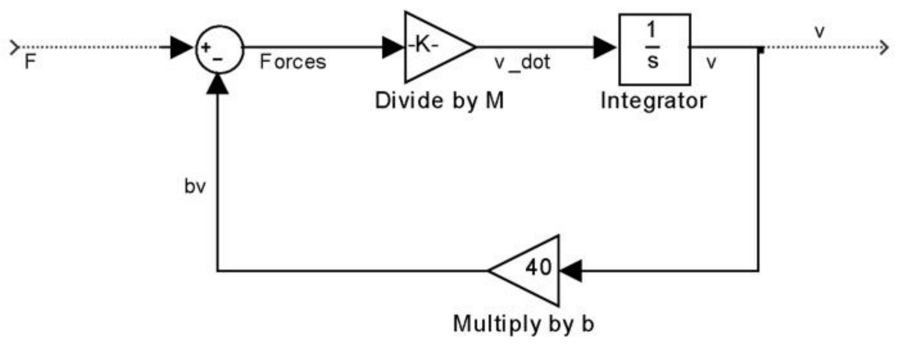


Figure-2.2.2

System Response to Step Input
To be simulate the system the applied input F is to be specified. Assume that the car is initially at rest, and that the engine applies a step input of F = 400 N at t = 0. This is approximately equivalent to the car's driver quickly pushing down and holding the gas pedal in a steady position starting from a stoplight. Insert a Step block from the Sources subfolder into the model, and also add a Scope block from the Sinks subfolder to monitor the system's velocity, v.

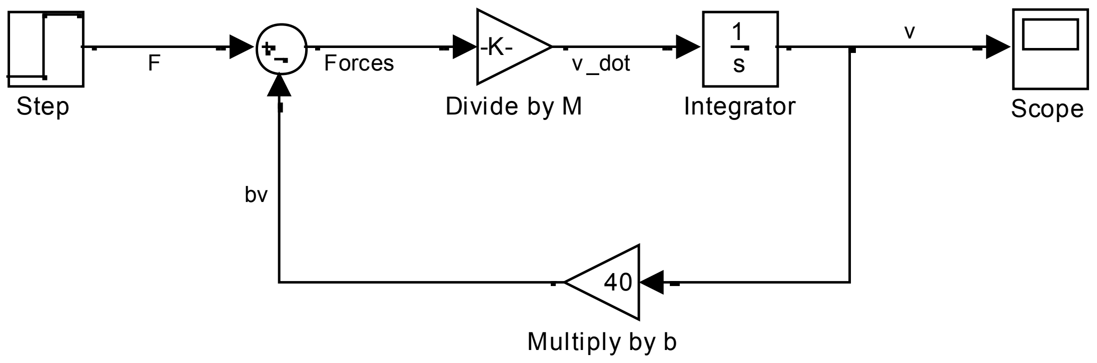
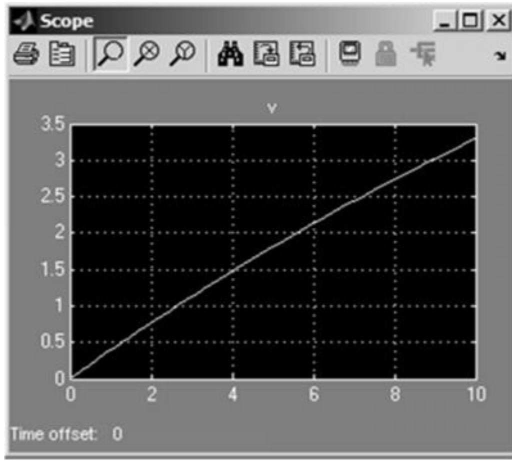


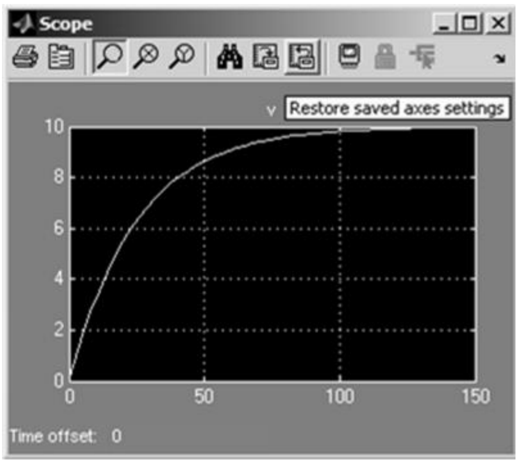
Figure-2.2.3

The Step block must be modified to correctly represent the system. Double-click on it, and change the Step Time to 0 and the Final Value to 400. The Initial Value can be left as 0, since the F step input starts from 0 at t = 0. The Sample Time should remain 0 so that the Step block's input is monitored continuously during simulation.

Next, run a simulation of the system (by clicking the "Start/Pause Simulation" button or selecting **Simulation**, **Start**). Once the simulation has finished, double-click on the Scope block to view the velocity response to the step input. Clicking on the "Autoscale" button (looks like a pair of binoculars) in the Scope window will produce the graph as shown in figure-2.2.4-(a).



(a)



(b)

Figure-2.2.4

This graph of figure-2.2.4-(a) does not appear to show the velocity approaching a steady-state value, as expected for the first-order response to a step input. This result is due to the settling time of the system being greater than the 10 seconds the simulation was run. To observe the system reaching steady-state, click **Simulation, Parameters** in the model window, and change the Stop Time to 150 seconds. Now, re-running the simulation will result in the velocity graph as shown in figure-2.2.4(b).

From this graph, we observe that the system has a steady-state velocity of about 10 m/s, and a time constant of about 25 seconds. Let's check these results with our original equation. For a step input of F = 400 N, the system equation is:

$$1000\frac{dv}{dt} + 40v = 400$$

Setting dv/dt = 0 gives a steady-state velocity of 10 m/s, a result which agrees with the velocity graph above.To find the time constant of the system the characteristic equation as shown below can be used. $1000s + 40 = 0$

Solving this gives the characteristic root, s = -0.04, and thus the time constant is indeed 25 seconds (τ = -1/s), as in the above the graph.

2.2.2 SIMULINK diagram from State-space model

Example-2.2.2

Use the state-space model to simulate the state and output equations described as below:

$$\begin{bmatrix}\dot{x}_1\\ \dot{x}_2\\ \dot{x}_3\end{bmatrix} = \begin{bmatrix}0 & 1 & 0\\ 0 & 0 & 1\\ -5 & -4 & -2\end{bmatrix}\begin{bmatrix}x_1\\ x_2\\ x_2\end{bmatrix} + \begin{bmatrix}0\\ 0\\ 5\end{bmatrix}u(t)$$

And $y = [1 \quad 0 \quad 0]x$

The State-Space model provides a dialog box where the A, B, C and D matrices can be entered in *MATLAB* matrix notation, or by variables defined in Workspace. A *SIMULINK* diagram using the State-Space model is constructed as shown in figure-2.2.5.

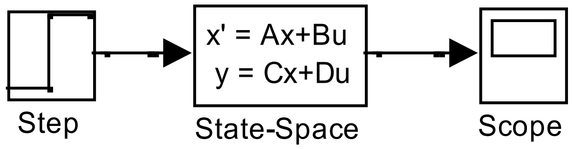


Figure-2.2.5

The output response of the system will be as presented in figure-2.2.6

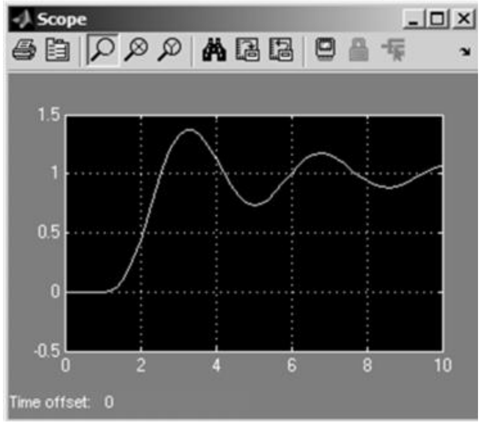


Figure-2.2.6

2.2.3 State-space model from SIMULINK diagram

SIMULINK provides the **linmod**, and **dlinmod** functions to extract linear models from the block diagram model in the form of the state-space matrices A, B, C, and D. The input and outputs of the SIMULINK diagram must be defined using **Inport** and **Outport** blocks in place of the **Source** and **Sink** blocks.

Example-2.2.3

Obtain the state-space model for the system represented by the block diagram shown in figure-2.2.7

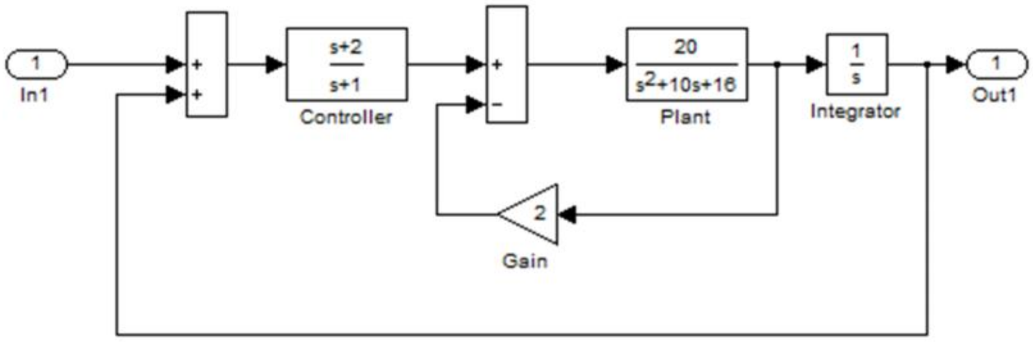


Figure-2.2.7

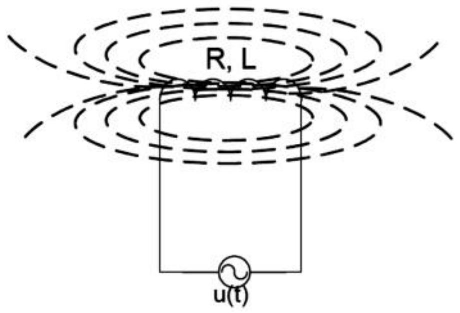
The model is saved with a filename **ex1**. Run the simulation and to extract the linear model of this *SIMULINK* system, in the Command Window, enter the command

```
                    [A,B,C,D]=linmod('ex1')
```
The result is
```
  A =                                B =                        C =                       D =
         0     0     0    20               0                          1     0     0     0            0
         1    -1     0     0               1
         1     1   -10   -56               1
         0     0     1     0               0
```

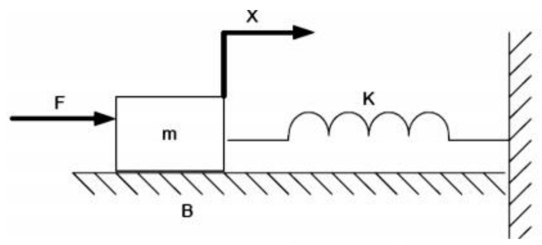
In order to obtains the transfer function of the system from the state-space model, we use the command
```
                    [num,den]=ss2tf(A,B,C,D)
```
The result is
```
num =
         0   -0.0000   -0.0000   20.0000   40.0000
den =
         1.0000   11.0000   66.0000   36.0000  -40.0000
```

Exercise:

1. For the systems shown below draw the block diagram using SIMULINK see the output in the scope with respect to input



Consider u(t) a unit step input and i(t) the output; assume L = 0.01 and R = 0.01 and zero initial conditions.



Consider F(t) a step input and x(t) the output; assume m = 2Kg, K= 32 and B = 2 N-s/m and zero initial conditions.

2. For the system defined by the equation

$$2\frac{d^3y}{dt^3} + 4\frac{d^2y}{dt^2} + 8\frac{dy}{dt} + 10y = 10u(t)$$

Draw the SIMULINK block diagram and plot the output response y(t) with respect to u(t).