CHITTAGONG UNIVERSITY OF ENGINEERING AND TECHNOLOGY
DEPARTMENT OF ELECTRONICS & TELECOMMUNICATION ENGINEERING
CHITTAGONG-4349, BANGLADESH.

COURSE NO.: ETE 212

Experiment No. 1

Representation and Conversion of Transfer Function and State-space functions using MATLAB

PRELAB WORK:

- **Read this laboratory manual carefully before coming to the laboratory class, so that you know what is required.**
- Try to follow the lecture notes of ETE 211.
- Familiarize yourself with relevant MATLAB functions and codes necessary for this experiment.
- **Do not bring any prepared MATLAB code in the lab with any portable device.**
- **DONOT** copy others blindly!!!
- **Submit your lab report before the roll call.**

THEORY WITH EXAMPLES:

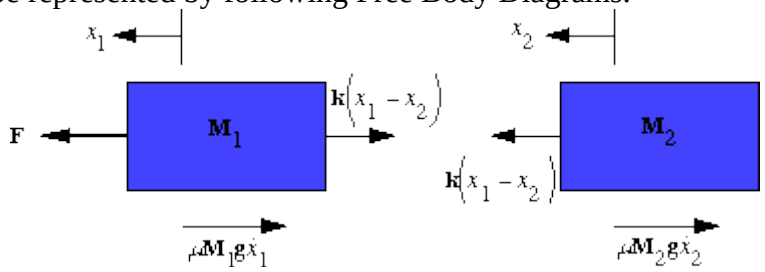
A Simple Train system

In this example, we will consider a toy train consisting of an engine and a car. Assuming that the train only travels in one direction, we want to apply control to the train so that it has a smooth start-up and stop, along with a constant-speed ride.

The mass of the engine and the car will be represented by M1 and M2, respectively. The two are held together by a spring, which has the stiffness coefficient of k. F represents the force applied by the engine, and the Greek letter, mu (which will also be represented by the letter u), represents the coefficient of rolling friction.

Free body diagram and Newton's law

The system can be represented by following Free Body Diagrams.



From Newton's law, you know that the sum of forces acting on a mass equals the mass times its acceleration. In this case, the forces acting on M1 are the spring, the friction and the force applied by the engine. The forces acting on M2 are the spring and the friction. In the vertical direction, the gravitational force is canceled by the normal force applied by the ground, so that there will be no acceleration in the vertical direction. The equations of motion in the horizontal direction are the following:

$$\mathbf{M}_1\ddot{x}_1 = \mathbf{F} - \mathbf{k}(x_1 - x_2) - \mu\mathbf{M}_1\mathbf{g}\dot{x}_1$$

$$\mathbf{M}_2\ddot{x}_2 = \mathbf{k}(x_1 - x_2) - \mu\mathbf{M}_2\mathbf{g}\dot{x}_2$$

State-variable and output equations

This set of system equations can now be manipulated into state-variable form. The state variable are the positions, X1 and X2, and the velocities, V1 and V2; the input is F. The state variable equations will look like the following:

$$\dot{\mathrm{x}}_1 = \mathrm{v}_1$$

$$\dot{\mathrm{v}}_1 = -\mathrm{k}/\mathrm{M}_1\,\mathrm{x}_1 - \mu\mathrm{g}\mathrm{v}_1 + \mathrm{k}/\mathrm{M}_1\,\mathrm{x}_2 + \mathrm{F}/\mathrm{M}_1$$

$$\dot{\mathrm{x}}_2 = \mathrm{v}_2$$

$$\dot{\mathrm{v}}_2 = \mathrm{k}/\mathrm{M}_2\,\mathrm{x}_1 - \mathrm{k}/\mathrm{M}_2\,\mathrm{x}_2 - \mu\mathrm{g}\mathrm{v}_2$$

Let the output of the system be the velocity of the engine. Then the output equation will be:

$$\mathrm{y} = \mathrm{v}_1$$

1. Transfer function

To find the transfer function of the system, we first take the Laplace transforms of the differential equations.

$$\mathbf{M}_1\mathbf{s}^2\mathbf{X}_1(\mathbf{s}) = \mathbf{F}(\mathbf{s}) - \mathbf{k}(\mathbf{X}_1(\mathbf{s}) - \mathbf{X}_2(\mathbf{s})) - \mu\mathbf{M}_1\mathbf{g}\mathbf{s}\mathbf{X}_1(\mathbf{s})$$

$$\mathbf{M}_2\mathbf{s}^2\mathbf{X}_2(\mathbf{s}) = \mathbf{k}(\mathbf{X}_1(\mathbf{s}) - \mathbf{X}_2(\mathbf{s})) - \mu\mathbf{M}_2\mathbf{g}\mathbf{s}\mathbf{X}_2(\mathbf{s})$$

The output is Y(s) = V2(s) = s X2(s). The variable X1 should be algebraically eliminated to leave an expression for Y(s)/F(s). **When finding the transfer function, zero initial conditions must be assumed.** The transfer function should look like the one shown below.

$$\frac{Y(s)}{F(s)} = \frac{M_2 s^2 + M_2 \mu g s + 1}{M_1 M_2 s^3 + (2 M_1 M_2 \mu g) s^2 + (M_1 k + M_1 M_2 (\mu g)^2 + M_2 k) s + k\mu g (M_1 + M_2)}$$

2. State-space

Another method to solve the problem is to use the state-space form. Four matrices A, B, C, and D characterize the system behavior, and will be used to solve the problem. The state-space form which is found from the state-variable and the output equations is shown below.

$$\begin{bmatrix} x_1 \\ v_1 \\ x_2 \\ v_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -k/M_1 & -\mu g & k/M_1 & 0 \\ 0 & 0 & 0 & 1 \\ k/M_2 & 0 & -k/M_2 & -\mu g \end{bmatrix} \begin{bmatrix} x_1 \\ v_1 \\ x_2 \\ v_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1/M_1 \\ 0 \\ 0 \end{bmatrix} [F]$$

$$y = \begin{bmatrix} 0 & 1 & 0 & 0 \end{bmatrix} \begin{bmatrix} x_1 \\ v_1 \\ x_2 \\ v_2 \end{bmatrix} + [0][F]$$

MATLAB representation

Now we will show you how to enter the equations derived above into an m-file for MATLAB. Since MATLAB cannot manipulate symbolic variables, let's assign numerical values to each of the variables. Let

M1 = 1 kg
M2 = 0.5 kg
k = 1 N/m
F= 1 N
u = 0.002 sec/m
g = 9.8 m/s^2

Create an new m-file and enter the following commands.

```
M1=1;
M2=0.5;
k=1;
F=1;
u=0.002;
g=9.8;
```

Now you have one of two choices: 1) Use the transfer function, or 2) Use the state-space form to solve the problem. If you choose to use the transfer function, add the following commands onto the end of the m-file which you have just created.

```
num = [M2 M2*u*g 1];
den = [M1*M2 2*M1*M2*u*g M1*k+M1*M2*u*u*g*g+M2*k M1*k*u*g+M2*k*u*g];
train = tf(num, den)
```

If you choose to use the state-space form, add the following commands at the end of the m-file, instead of num and den matrices shown above.

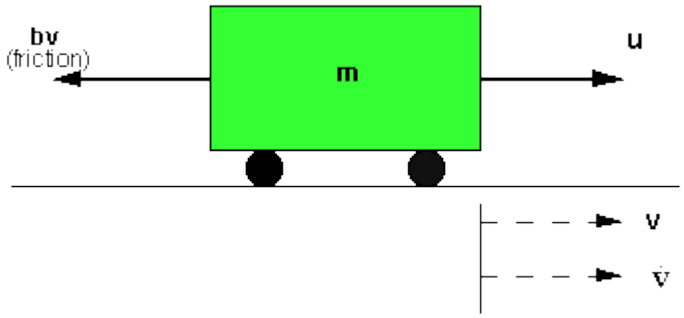
```
A = [0, 1, 0, 0;
     -k/M1, -u*g, k/M1, 0;
```

```
     0, 0, 0, 1;
     k/M2, 0, -k/M2, -u*g];
B = [ 0;
      1/M1;
      0;
      0];
C = [0, 1, 0, 0];
D = [0];
train = ss(A,B,C,D)
```

A Real-time Example: Modeling a Cruise Control System

Physical setup and system equations

The model of the cruise control system is relatively simple. If the inertia of the wheels is neglected, and it is assumed that friction (which is proportional to the car's speed) is what is opposing the motion of the car, then the problem is reduced to the simple mass and damper system shown below.



Using Newton's law, the modeling equations for this system become:

$$m\dot{v} + bv = u \quad (1)$$

$$y = v$$

where u is the force from the engine. For this example, let's assume that

m = 1000kg
b = 50Nsec/m
u = 500N

Design requirements

The next step in modeling this system is to come up with some design criteria. When the engine gives a 500 Newton force, the car will reach a maximum velocity of 10 m/s (22 mph). An automobile should be able to accelerate up to that speed in less than 5 seconds. Since this is only a cruise control system, a 10% overshoot on the velocity will not do much damage. A 2% steady-state error is also acceptable for the same reason.

Keeping the above in mind, we have proposed the following design criteria for this problem:

Rise time < 5 sec
Overshoot < 10%
Steady state error < 2%

MATLAB representation

1. Transfer Function

To find the transfer function of the above system, we need to take the Laplace transform of the modeling equations (1). **When finding the transfer function, zero initial conditions must be assumed.** Laplace transforms of the two equations are shown below.

$$msV(s) + bV(s) = U(s)$$
$$Y(s) = V(s)$$

Since our output is the velocity, let's substitute V(s) in terms of Y(s)

$$msY(s) + bY(s) = U(s)$$

The transfer function of the system becomes

$$\frac{Y(s)}{U(s)} = \frac{1}{ms + b}$$

To solve this problem using MATLAB, copy the following commands into an new m-file:

```
m=1000;
b=50;
u=500;
num = [1];
den = [m b];
cruise = tf(num, den);
```

These commands will later be used to find the open-loop response of the system to a step input. But before getting into that, let's take a look at the state-space representation.

2. State-Space

We can rewrite the first-order modeling equation (1) as the state-space model.

$$[\dot{v}] = \left[-b/m\right][v] + \left[1/m\right][u]$$
$$y = [1][v]$$

To use MATLAB to solve this problem, create an new m-file and copy the following commands:

```
m = 1000;
b = 50;
u = 500;
A = [-b/m];
B = [1/m];
C = [1];
```

```
D = 0;
cruise = ss(A,B,C,D);
```

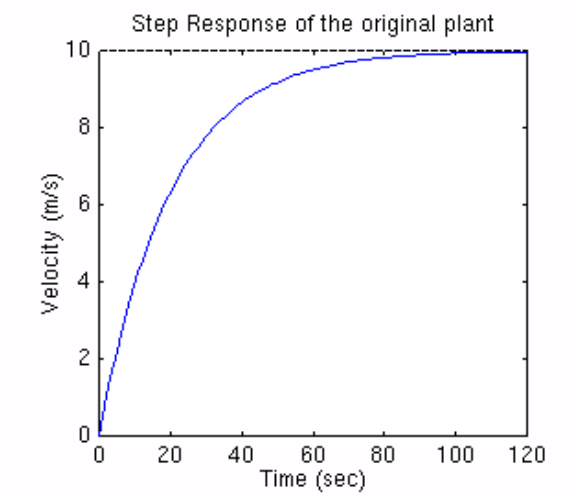
Note: It is possible to convert from the state-space representation to the transfer function or vice versa using MATLAB. To learn more about the conversion, explore MATLAB help files.

Open-loop response

Now let's see how the open-loop system responds to a step input. Add the following command to the end of your m-file and run it in the MATLAB command window:

```
step(u*cruise)
```

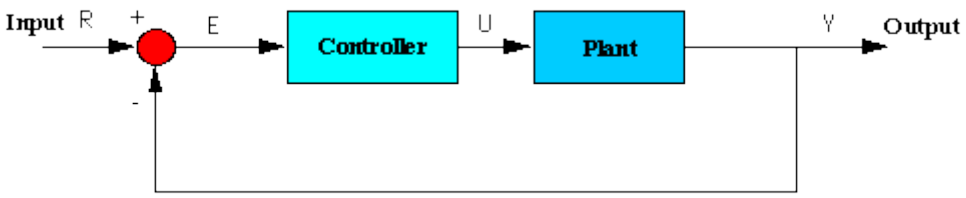
You should get the following plot:



From the plot, we see that the vehicle takes more than 100 seconds to reach the steady-state speed of 10 m/s. This does not satisfy our rise time criterion of less than 5 seconds.

Closed-loop transfer function

To solve this problem, a unity feedback controller will be added to improve the system performance. The figure shown below is the block diagram of a typical unity feedback system.

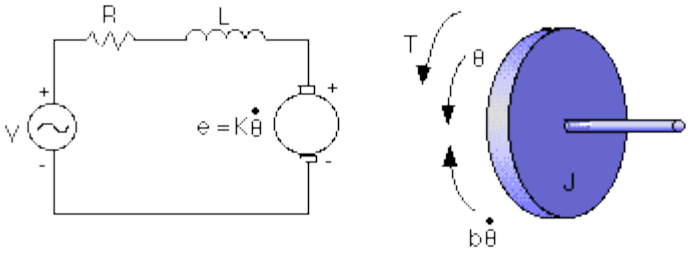


The transfer function in the plant is the transfer function derived above {Y(s)/U(s)=1/ms+b}. The controller will to be designed to satisfy all design criteria.

Another Real-time Example: DC Motor Speed Modeling

Physical setup and system equations

A common actuator in control systems is the DC motor. It directly provides rotary motion and, coupled with wheels or drums and cables, can provide transitional motion. The electric circuit of the armature and the free body diagram of the rotor are shown in the following figure:



For this example, we will assume the following values for the physical parameters.

moment of inertia of the rotor (J) = 0.01 kg.m^2/s^2
damping ratio of the mechanical system (b) = 0.1 Nms
electromotive force constant (K=Ke=Kt) = 0.01 Nm/Amp
electric resistance (R) = 1 ohm
electric inductance (L) = 0.5 H
input (V): Source Voltage
output (theta): position of shaft
The rotor and shaft are assumed to be rigid

The motor torque, **T**, is related to the armature current, **i**, by a constant factor **Kt**. The back emf, **e**, is related to the rotational velocity by the following equations:

$$T = K_t i$$

$$e = K_e \dot{\theta}$$

In SI units (which we will use), **Kt** (armature constant) is equal to **Ke** (motor constant).
From the figure above we can write the following equations based on Newton's law combined with Kirchhoff's law:

$$J\ddot{\theta} + b\dot{\theta} = Ki$$

$$L\frac{di}{dt} + Ri = V - K\dot{\theta}$$

1. Transfer Function

Using Laplace Transforms, the above modeling equations can be expressed in terms of s.

$$s(Js + b)\Theta(s) = KI(s)$$

$$(Ls + R)I(s) = V - Ks\Theta(s)$$

By eliminating I(s) we can get the following open-loop transfer function, where the rotational speed is the output and the voltage is the input.

$$\frac{\dot{\theta}}{V} = \frac{K}{(Js + b)(Ls + R) + K^2}$$

2. State-Space

In the state-space form, the equations above can be expressed by choosing the rotational speed and electric current as the state variable and the voltage as an input. The output is chosen to be the rotational speed.

$$\frac{d}{dt}\begin{bmatrix} \dot{\theta} \\ i \end{bmatrix} = \begin{bmatrix} -\frac{b}{J} & \frac{K}{J} \\ -\frac{K}{L} & -\frac{R}{L} \end{bmatrix}\begin{bmatrix} \dot{\theta} \\ i \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{L} \end{bmatrix}V$$

$$\dot{\theta} = \begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} \dot{\theta} \\ i \end{bmatrix}$$

Design requirements

First, our uncompensated motor can only rotate at 0.1 rad/sec with an input voltage of 1 Volt (this will be demonstrated later when the open-loop response is simulated). Since the most basic requirement of a motor is that it should rotate at the desired speed, the steady-state error of the motor speed should be less than 1%. The other performance requirement is that the motor must accelerate to its steady-state speed as soon as it turns on. In this case, we want it to have a settling time of 2 seconds. Since a speed faster than the reference may damage the equipment, we want to have an overshoot of less than 5%.
If we simulate the reference input (r) by an unit step input, then the motor speed output should have:

Settling time less than 2 seconds
Overshoot less than 5%
Steady-state error less than 1%

MATLAB representation and open-loop response

1. Transfer Function

We can represent the above transfer function into MATLAB by defining the numerator and denominator matrices as follows:

$$num = K$$
$$den = (Js + b)(Ls + R) + K^2$$

Create a new m-file and enter the following commands:

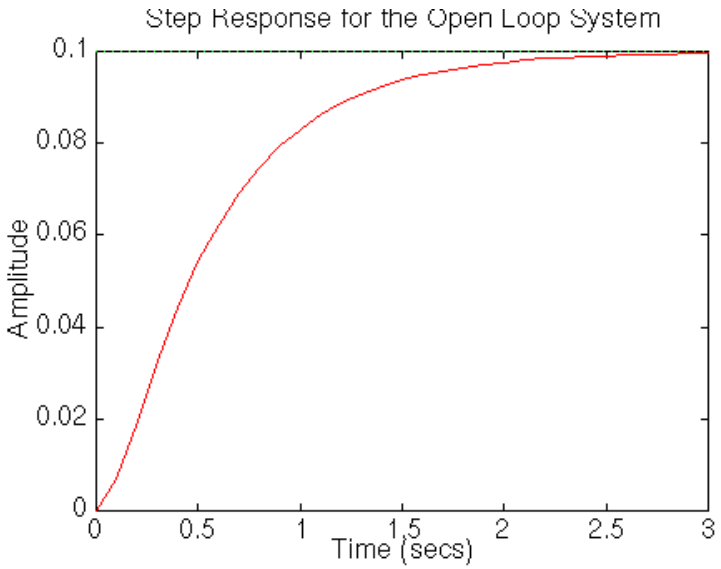
```
J = 0.01;
b = 0.1;
K = 0.01;
```

```
R = 1;
L = 0.5;
num = K;
den = [(J*L) ((J*R) + (L*b)) ((b*R) + K^2)];
motor = tf(num, den);
```

Now let's see how the original open-loop system performs. Add the following commands onto the end of the m-file and run it in the MATLAB command window:

```
step(motor,0:0.1:3);
title('Step Response for the Open Loop System');
```

You should get the following plot:



From the plot we see that when 1 volt is applied to the system, the motor can only achieve a maximum speed of 0.1 rad/sec, ten times smaller than our desired speed. Also, it takes the motor 3 seconds to reach its steady-state speed; this does not satisfy our 2 seconds settling time criterion.

2. State-Space

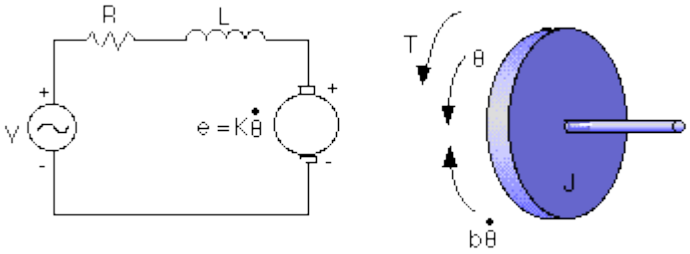
We can also represent the system using the state-space equations. Try the following commands in a new m-file.

```
J=0.01;
b=0.1;
K=0.01;
R=1;
L=0.5;
A = [-b/J, K/J
      -K/L, -R/L];
B = [0;
   1/L];
C = [1, 0];
D = 0;
motor_ss = ss(A, B, C, D);
step(motor_ss)
```

Run this m-file in the MATLAB command window, and you should get the same output as the one shown above.

Home Task

A common actuator in control systems is the DC motor. It directly provides rotary motion and, coupled with wheels or drums and cables, can provide transitional motion. The electric circuit of the armature and the free body diagram of the rotor are shown in the following figure:



For this task, we will assume the following values for the physical parameters.

moment of inertia of the rotor (J) = (3.2284 + <your ID>)E-6 kg.m^2/s^2
damping ratio of the mechanical system (b) = (3.5077 + <your ID>)E-6 Nms
electromotive force constant (K=Ke=Kt) = (0.0274 + 0.<your ID>) Nm/Amp
electric resistance (R) = 4 ohm
electric inductance (L) = (2.75 + <your ID>)E-6 H
input (V): Source Voltage
output (theta): position of shaft
The rotor and shaft are assumed to be rigid

The motor torque, **T**, is related to the armature current, **i**, by a constant factor **Kt**. The back emf, **e**, is related to the rotational velocity by the following equations:

$$T = K_t i$$

$$e = K_e \dot{\theta}$$

In SI units (which we will use), **Kt** (armature constant) is equal to **Ke** (motor constant).
From the figure above we can write the following equations based on Newton's law combined with Kirchhoff's law:

$$J\ddot{\theta} + b\dot{\theta} = Ki$$

$$L\frac{di}{dt} + Ri = V - K\dot{\theta}$$

1) Express the above equations in transfer function. Where, the rotating speed is the output and the voltage is an input.
2) Express the above equations in state-space form.
3) Simulate the reference input (R) by a unit step input, then the motor speed output should have:
   Settling time less than 40 milliseconds
   Overshoot less than 16%
   No steady-state error due to a disturbance