Linear Control Systems

Laboratory Experiment 2: Mathematical Modeling of Physical Systems

Objectives: The objective of this exercise is to grasp the important role mathematical models of physical systems in the design and analysis of control systems. We will learn how MATLAB helps in solving such models.

List of Equipment/Software

Following equipment/software is required:

- MATLAB

Category Soft-Experiment

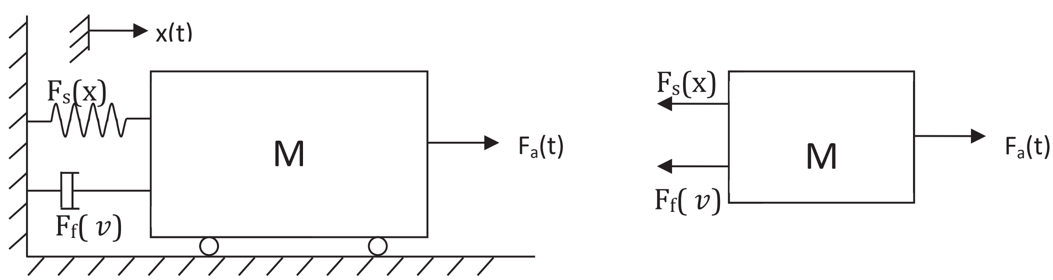
Deliverables

A complete lab report including the following:

- Summarized learning outcomes.
- MATLAB scripts and their results for all the assignments and exercises should be properly reported.

Mass-Spring System Model

Consider the following Mass-Spring system shown in the figure. Where $F_s(x)$ is the spring force, $F_f(\dot{x})$ is the friction coefficient, x(t) is the displacement and $F_a(t)$ is the applied force:



Where

$$a = \frac{dv(t)}{dt} = \frac{d^2x(t)}{dt^2} \text{ is the acceleration,}$$
$$v = \frac{dx(t)}{dt} \text{ is the speed,}$$

and

$$x(t) \text{ is the displacement.}$$

According to the laws of physics

$$Ma + F_f(v) + F_s(x) = F_a(t) \quad (1)$$

In the case where:

$$F_f(v) = Bv = B\frac{dx(t)}{dt}$$

$$F_s(x) = Kx(t)$$

The differential equation for the above Mass-Spring system can then be written as follows

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

B is called the friction coefficient and K is called the spring constant.

The linear differential equation of second order (2) describes the relationship between the displacement and the applied force. The differential equation can then be used to study the time behavior of x(t) under various changes of the applied force. In reality, the spring force and/or the friction force can have a more complicated expression or could be represented by a graph or data table. For instance, a nonlinear spring can be designed (see figure 4.2) such that

$$F_s(x) = Kx^r(t) \quad \text{Where } r > 1.$$

Figure 4.2: MAG nonlinear spring (www.tokyo-model.com.hk/ecshop/goods.php?id=2241)
In such case, (1) becomes

$$M\frac{d^2x(t)}{dt^2} + B\frac{dx(t)}{dt} + Kx^r(t) = F_a(t) \quad (3)$$

Equation (3) represents another possible model that describes the dynamic behavior of the mass-damper system under external force. Model (2) is said to be a linear model whereas (3) is said to be nonlinear. To decide if a system is linear or nonlinear two properties have to be verified homogeneity and superposition.

Assignment: use homogeneity and superposition properties to show that model (1) is linear whereas model (3) is nonlinear.

Solving the differential equation using MATLAB:
The objectives behind modeling the mass-damper system can be many and may include

- Understanding the dynamics of such system

- Studying the effect of each parameter on the system such as mass M, the friction coefficient B, and the elastic characteristic Fs(x).
- Designing a new component such as damper or spring.
- Reproducing a problem in order to suggest a solution.

The solution of the difference equations (1), (2), or (3) leads to finding x(t) subject to certain initial conditions.

MATLAB can help solve linear or nonlinear ordinary differential equations (ODE). To show how you can solve ODE using MATLAB we will proceed in two steps. We first see how can we solve first order ODE and second how can we solve equation (2) or (3).

Speed Cruise Control example:

Assume the spring force $F_s(x) = 0$ which means that K=0. Equation (2) becomes

$$M\frac{d^2x(t)}{dt^2} + B\frac{dx(t)}{dt} = F_a(t) \tag{4}$$

Or

$$M\frac{dv(t)}{dt} + Bv = F_a(t) \tag{5}$$

Equation (5) is a first order linear ODE.

Using MATLAB solver ode45 we can write do the following:

1_ create a MATLAB-function cruise_speed.m

```
function dvdt=cruise_speed(t, v)

%flow rate

M=750; %(Kg)

B=30; %( Nsec/m)

Fa=300; %N

% dv/dt=Fa/M-B/M v

dvdt=Fa/M-B/M*v;
```

2_ create a new MATLAB m-file and write

```
v0= 0; %(initial speed)

[t,v]=ode45('cruise_speed', [0 125],v0);

plot(t,v); grid on;

title('cruise speed time response to a constant traction force Fa(t) ')
```

There are many other MATLAB ODE solvers such as ode23, ode45, ode113, ode15s, etc… The function dsolve will result in a symbolic solution. Do 'doc dsolve' to know more. In MATLAB write

```
>>dsolve('Dv=Fa/M-B/M*v', 'v(0)=0')
```

Note that using MATLAB ODE solvers are able to solve linear or nonlinear ODE's. We will see in part II of this experiment another approach to solve a linear ODE differently. Higher order systems can also be solved similarly.

Mass-Spring System Example:

Assume the spring force $F_s(x) = Kx^r(t)$. The mass-spring damper is now equivalent to

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

The second order differential equation has to be decomposed in a set of first order differential equations as follows

Variables	New variable	Differential equation
x(t)	X_1	$\frac{dX_1}{dt} = X_2$
dx(t)/dt	X_2	$\frac{dX_2}{dt} = -\frac{B}{M}X_2 - \frac{K}{M}{X_1}^r(t) + \frac{F_a(t)}{M}$

In vector form, let $X = \begin{bmatrix} X_1 \\ X_2 \end{bmatrix}$; $\frac{dX}{dt} = \begin{bmatrix} \frac{dX_1}{dt} \\ \frac{dX_2}{dt} \end{bmatrix}$ then the system can be written as

$$\frac{dX}{dt} = \begin{bmatrix} X_2 \\ -\frac{B}{M}X_2 - \frac{K}{M}{X_1}^r(t) + \frac{F_a(t)}{M} \end{bmatrix}$$

The ode45 solver can be now be used:

1_ create a MATLAB-function mass_spring.m

```
Function dXdt=mass_spring(t, X)

%flow rate

M=750; %(Kg)

B=30; %( Nsec/m)

Fa=300; %N

K=15; %(N/m)

r=1;

% dX/dt
```

```
dXdt(1,1)=X(2);
dXdt(2,1)=-B/M*X(2)-K/M*X(1)^r+Fa/M;
```

2_ in MATLAB write

```
>> X0=[0; 0]; %(initial speed and position)
>> options = odeset('RelTol',[1e-4 1e-4],'AbsTol',[1e-5 1e-5],'Stats','on');
>>[t,X]=ode45('mass_spring', [0 200],X0);
```

Exercise 1

1. Plot the position and the speed in separate graphs.
2. Change the value of r to 2 and 3.
3. Superpose the results and compare with the linear case r=1 and plot all three cases in the same plot window. Please use different figures for velocity and displacement.

Exercise 2

Consider the mechanical system depicted in the figure. The input is given by $f(t)$, and the output is given by $y(t)$. Determine the differential equation governing the system and using MATLAB, write a m-file and plot the system response such that forcing function f(t)=1. Let $m = 10$, $k = 1$ and $b = 0.5$. Show that the peak amplitude of the output is about 1.8.