LAB 6: Introduction to ODE Solvers along with its implementation on the translational mechanical system.

1. OBJECTIVES:

- To simulate and analyze the translational mechanical system by applying the translational dynamics.
- To become familiar with solving ordinary differential equations in MATLAB.
- To understand the effect of controllable parameters on the translational mechanical system.

2. INTRODUCTION:

A Differential Equation is an equation with a function and one or more of its derivatives:

$$y = \frac{dy}{dx} + 5x$$

For example an equation with the function "y" and its derivative $\frac{dy}{dx}$. The term ordinary is used in contrast with the term partial differential equation which may be concerning more than one independent variable.

3. SOLVING ODE USING MATLAB:

Before learning how to use ODE solvers, lets discuss how to define a Function in MATLAB.

3.1 Defining a simple function in Matlab

Syntax

`[y1,...,yN] = myfun(x1,...,xM)`

declares a function named `myfun` that accepts inputs `x1,...,xM` and returns outputs `y1,...,yN`.

Example

```
Function File:

function ave = average(x)
    ave = sum(x(:))/numel(x);
end

Script:
z = 1:99;
ave = average(z)
```

3.2. USING ODE SOLVERS IN MATLAB:

Syntax:

```
[t,y] = ode45(odefun,tspan,y0),
```

Input= `odefun` (function name as saved) , `tspan` (time range of simulation), `y0` (initial condition)

where `tspan = [t0 tf]`, integrates the system of differential equations $y'=f(t,y)$ from `t0` to `tf` with initial conditions `y0`. Each row in the solution array `y` corresponds to a value returned in column vector `t`.

Quick Hands-on Practice:

Solve this expression using *ode45* in MATLAB.

$$M\ddot{x} + B\dot{x} = F_a(t)$$

$$=> \ddot{x} = \frac{F_a(t)}{M} - \frac{B\dot{x}}{M} \qquad , \text{ here } \dot{x} = v \;\&\; \ddot{x} = \dot{v}$$

M=750, B=30, F_a=300. Initial conditions are 0. Plot the velocity with respect to time.

<u>Function file:</u>

```
function dvdt=example(t,v)
M=750;B=30;Fa=300;
dvdt=(Fa/M)-(B/M)*v;
end
```

<u>Script file:</u>

```
v0=0;     % initial conditions
TR=[0 200];      % Time range
[t,v]=ode45(@example,TR,v0);
plot(t,v)
```

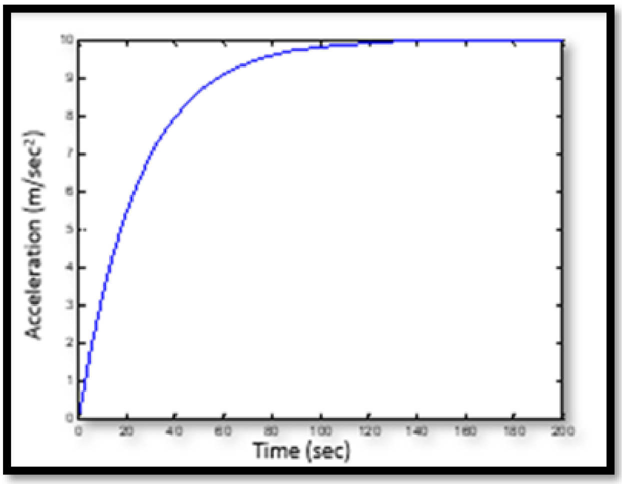
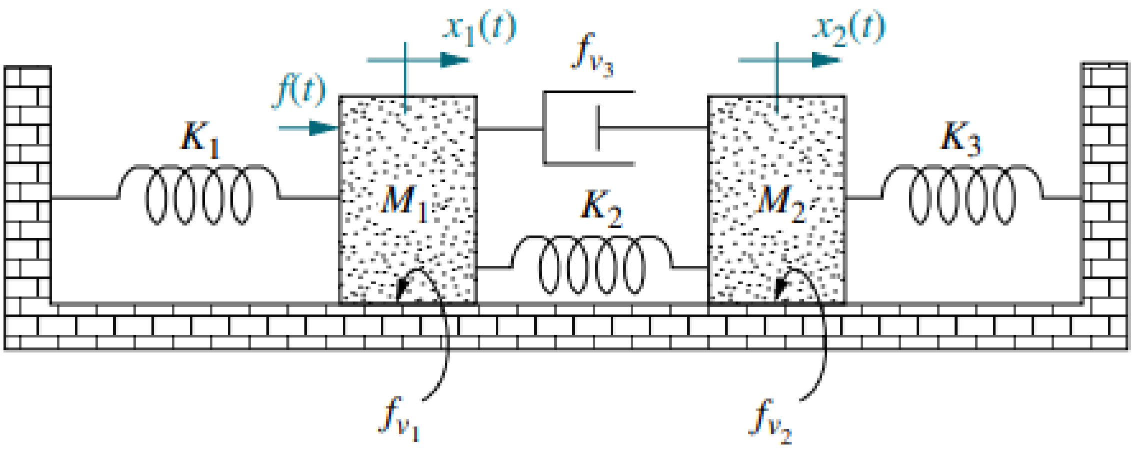


Figure 1: Velocity vs time

Assigned Tasks

Task#01: Write the MATLAB code to analyze the time-varying displacement, linear velocity, and linear acceleration of each inertial element of the given mechanical system. Assume all the values of each discrete element (As discussed in the lab). Furthermore, in this question, you can assume the value of Force as 5 Nm



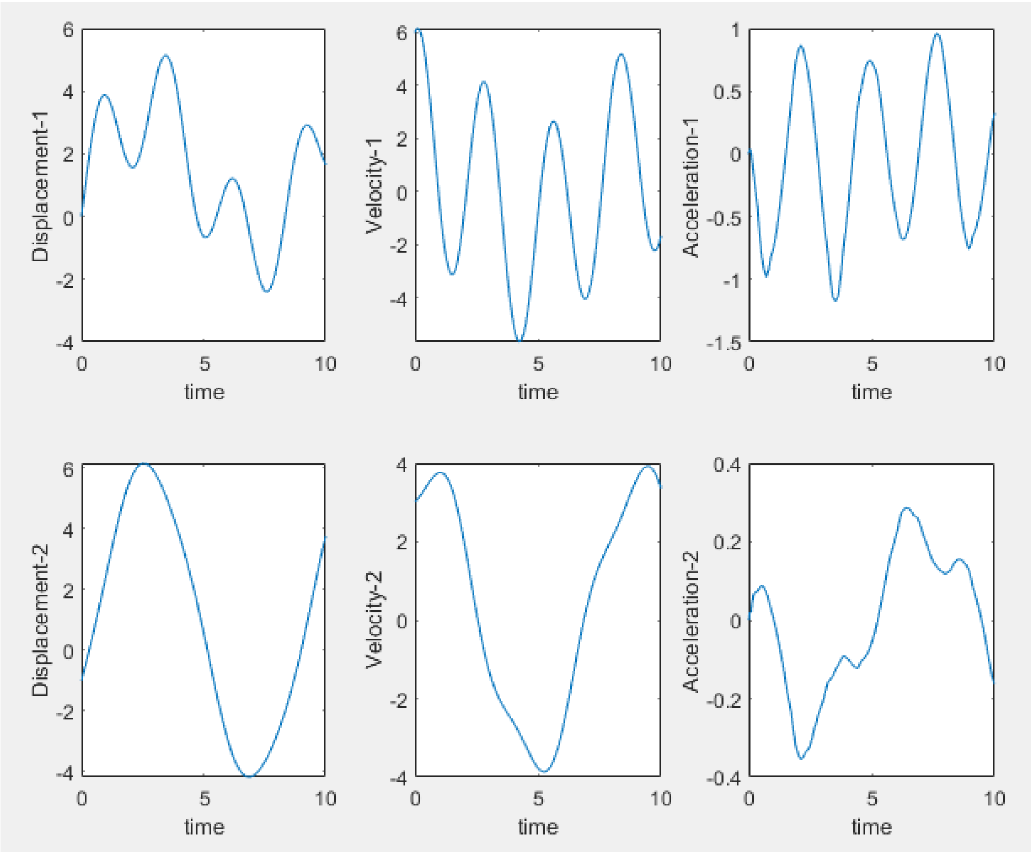
Function File:

```
function dy= fun(t,y)
   f=5;
   m1=1;
   m2=4;
   k1=3;
   k2=2;
   k3=1;
   fv1=0.03;
   fv2=0.02;
   fv3=0.01;
   dy(1)=y(2);
   dy(3)=y(4);
   dy(2)=1/m1*(f-(k1+k2)*y(1)-(fv1+fv3)*y(2)+fv3*y(4)+k2*y(3));
   dy(4)=1/m2*(-(fv2+fv3)*y(4)-(k2+k3)*y(3)+k2*y(1)+fv3*y(2));
   dy=dy';
end
```

Script File:

```
clc;
TR = [0 10];
X0 = [0;6;-1;3];
[t,y]=ode45(@fun,TR,X0);
x1=y(:,1);
v1=y(:,2);
x2=y(:,3);
v2=y(:,4);
a1=gradient(v1);
a2=gradient(v2);
subplot(2,3,1)
plot(t,x1)
xlabel('time')
ylabel('Displacement-1')
subplot(2,3,2)
plot(t,v1)
xlabel('time')
ylabel('Velocity-1')
subplot(2,3,3)
plot(t,a1)
xlabel('time')
ylabel('Acceleration-1')
subplot(2,3,4)
plot(t,x2)
xlabel('time')
ylabel('Displacement-2')
subplot(2,3,5)
plot(t,v2)
xlabel('time')
ylabel('Velocity-2')
subplot(2,3,6)
plot(t,a2)
xlabel('time')
ylabel('Acceleration-2')
```

Output:



Task#02: Consider the simplest translational mechanical system as shown in Figure below and utilize the provided code to implement the following action items:

a) Write the MATLAB code to analyze the time-varying displacement as well as the velocity of each mass by assuming all the values of each discrete element. Furthermore, in this part (a), you can assume the value of force as 5 N (Already done in the most recent lab).

b) Improve the MATLAB code in such a way that instead of applying constant force, the user should be capable of analyzing the output response (i.e., Four graphs) by considering the variant force (1:1:5) which means on each graph, there should be five lines and you can include the legend to represent the correspondence of each line with the value of applied force.

Task 2 (a):

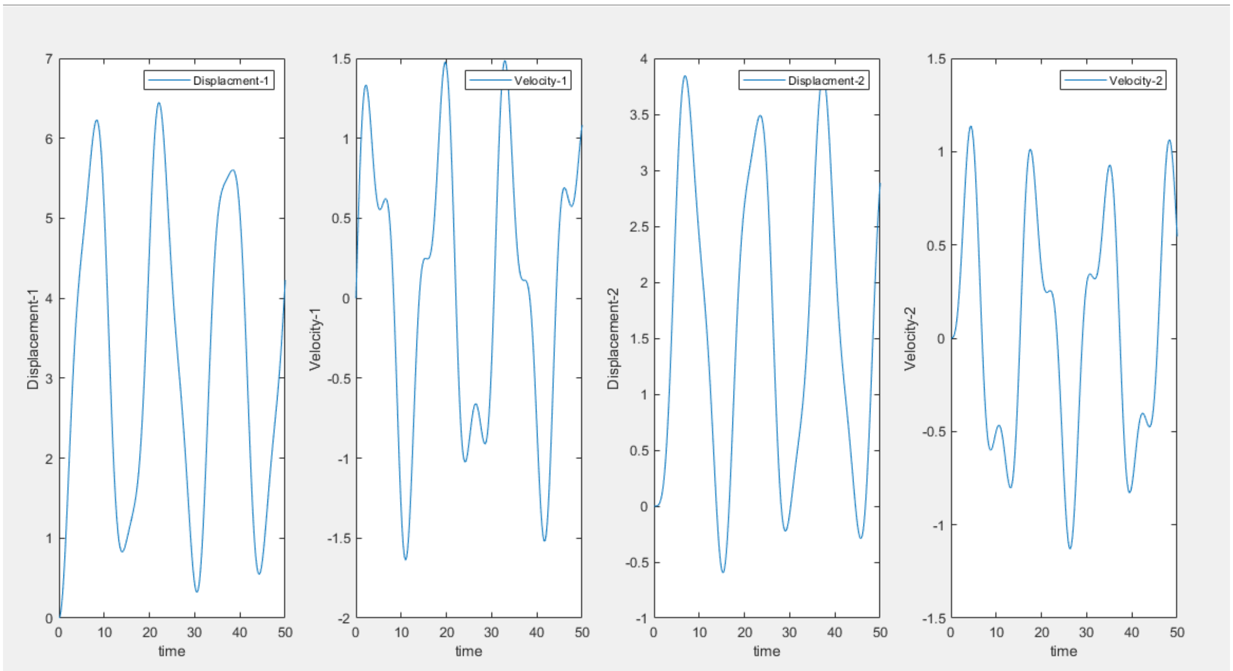
```
clc;
TR = [0 50];
X0 = [0;0;0;0];
[t,y]=ode45(@funs,TR,X0);
x1=y(:,1);
v1=y(:,2);
x2=y(:,3);
v2=y(:,4);
figure;
subplot(1,4,1);
plot(t,x1);
legend('Displacment-1');
xlabel('time');
ylabel('Displacement-1');
subplot(1,4,2);
plot(t,v1);
legend('Velocity-1');
xlabel('time');
ylabel('Velocity-1');
subplot(1,4,3);
plot(t,x2);
legend('Displacment-2');
xlabel('time');
ylabel('Displacement-2');
subplot(1,4,4);
plot(t,v2);
legend('Velocity-2');%
xlabel('time');
ylabel('Velocity-2');
```

Function File:

```
function dy= funs(t,y)
       m1=5;
       m2=10;
       k=3;
       b1=0.03;
       b2=0.02;
       F=5;
       dy(1)=y(2);
       dy(3)=y(4);
       dy(2)=1/m1*(F-b1*y(2)-k*y(1)+k*y(3));
       dy(4)=1/m2*(k*y(1)-b2*y(4)-(2*k*y(3)));

   dy=dy';
end
```

Output:



Task 2 (b):

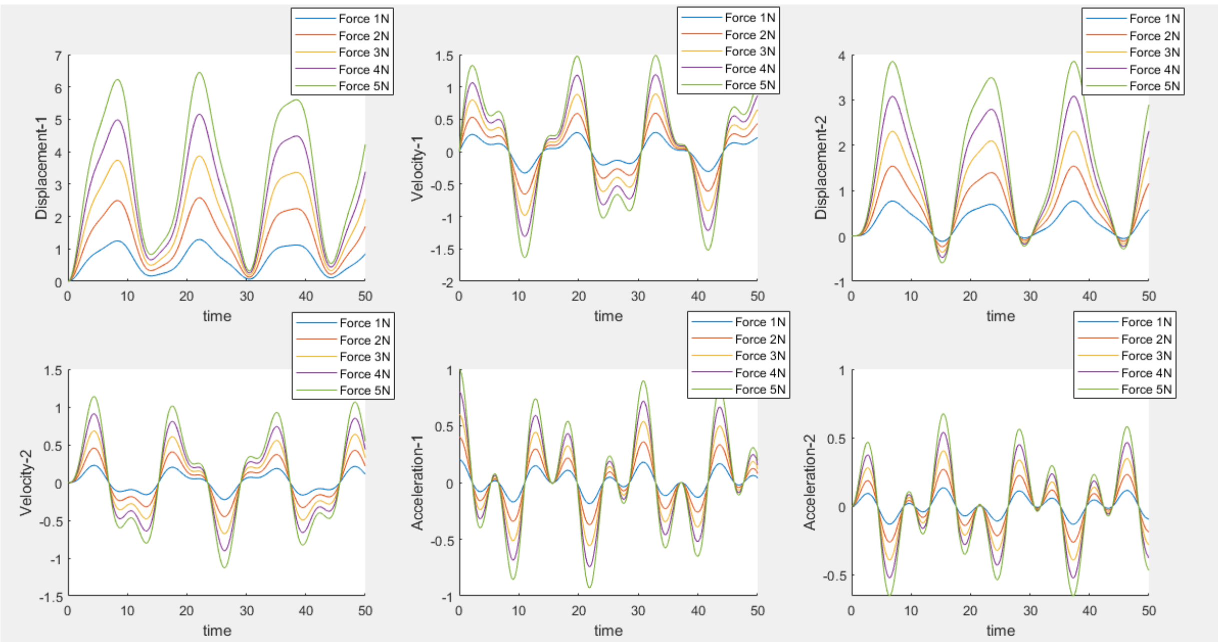
```
clc;
TR = [0 50];
X0 = zeros(1,20);
[t,y]=ode45(@sys,TR,X0);
velocity1=[y(:,2) y(:,6) y(:,10) y(:,14) y(:,18)];
velocity2=[y(:,4) y(:,8) y(:,12) y(:,16) y(:,20)];
figure;
for j=1:4
    subplot(2,3,j)
    for i=1:5
        index=(i-1)*4;
        hold on
        plot(t,y(:,index+j))
    end
    legend('Force 1N', 'Force 2N', 'Force 3N', 'Force 4N', 'Force 5N')%
    xlabel('time')
end
subplot(2,3,1);ylabel('Displacement-1');
subplot(2,3,2);ylabel('Velocity-1');
subplot(2,3,3);ylabel('Displacement-2');
subplot(2,3,4);ylabel('Velocity-2');
subplot(2,3,5);
for i=1:5
    hold on
    plot(t,gradient(velocity1(:,i),t))
end
```

```
legend('Force 1N', 'Force 2N', 'Force 3N', 'Force 4N', 'Force 5N');
xlabel('time');
ylabel('Acceleration-1');
subplot(2,3,6);
for i=1:5
    hold on
    plot(t,gradient(velocity2(:,i),t))
end
legend('Force 1N', 'Force 2N', 'Force 3N', 'Force 4N', 'Force 5N');
xlabel('time');
ylabel('Acceleration-2');
```

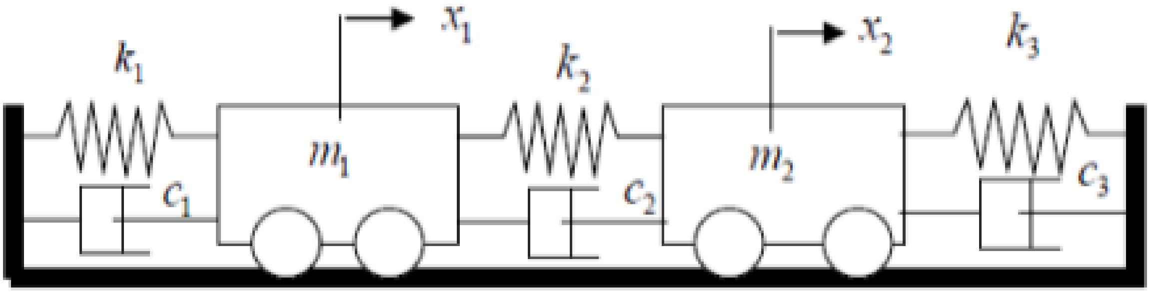
Function File:

```
function dy= sys(t,y)
   for F=1:5
        m1=5;
        m2=10;
        k=3;
        b1=0.03;
        b2=0.02;
        index=(F-1)*4;
        dy(index+1)=y(index+2);
        dy(index+3)=y(index+4);
        dy(index+2)=1/m1*(F-b1*y(index+2)-k*y(index+1)+k*y(index+3));
        dy(index+4)=1/m2*(k*y(index+1)-b2*y(index+4)-(2*k*y(index+3)));
   end
   dy=dy';
end
```

Output:



Task#03: A dual mass cart is moving on a frictionless surface which in turn ignores the influence of the external disturbance as shown in Figure. It is assumed that cart is beheld between a single frame which in turn corresponds the stiffness as well as damping characteristics of the cart. You are required to analyze the displacements and velocities of both masses. **Assume the force is being applied to mass 1. The value of this force will be given by the user. It can either be a single force value (5N) or a range of forces.**



Function file:

```
function dy=Task2Fun(t,y,f)
   m1=1;
   m2=4;
   k1=3;
   k2=2;
   k3=1;
   c1=0.03;
   c2=0.02;
   c3=0.01;
   dy(1)=y(2);
   dy(3)=y(4);
   dy(2)=1/m1*(f-(k1+k2)*y(1)-(c1+c2)*y(2)+c2*y(4)+k2*y(3));
   dy(4)=1/m2*(-(c2+c3)*y(4)-(k2+k3)*y(3)+k2*y(1)+c2*y(2));
   dy=dy';
end
```

Script File:

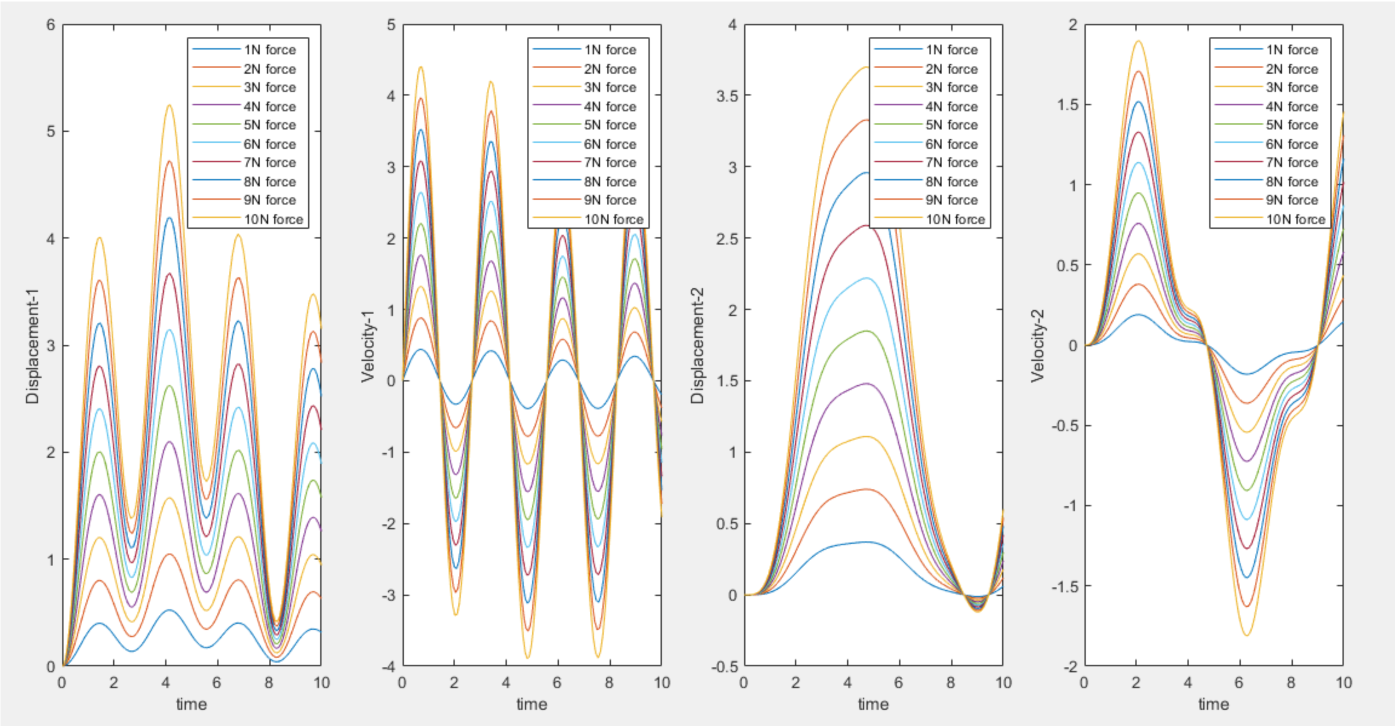
```
clc;
query=input('Do you want to analyze the system at single Force (5N) or on a range of
forces? (Single/Range) >> ','s');
TR = [0 10];
X0 = [0;0;0;0];
if query=="Single" || query=="single"
   range=[5,5];
   inc=1;
elseif query=="Range" || query=="range"
   range=input('Please enter a start and an end value for the force in the format
[start,end] >> ');
   inc=input('Please enter an increment value >> ');
end
```

```
for F=range(1):inc:range(2)
    [t,y]=ode45(@(t,y) Task2Fun(t,y,F),TR,X0);
    x1=y(:,1);
    v1=y(:,2);
    x2=y(:,3);
    v2=y(:,4);
    subplot(1,4,1);
    plot(t,x1);
    hold on;
    xlabel('time');
    ylabel('Displacement-1');
    subplot(1,4,2);
    plot(t,v1);
    hold on;
    xlabel('time');
    ylabel('Velocity-1');
    subplot(1,4,3);
    plot(t,x2);
    hold on;
    xlabel('time');
    ylabel('Displacement-2');
    subplot(1,4,4);
    plot(t,v2);
    hold on;
    xlabel('time');
    ylabel('Velocity-2');
end

text="";
for i=range(1):inc:range(2)
    text(end+1)=sprintf("%dN force",i);
end
text=text(2:end);
for i=1:4
    subplot(1,4,i);legend(text);
end
```

Output:

```
Do you want to analyze the system at single Force (5N) or on a range of forces? (Single/Range) >> Range
Please enter a start and an end value for the force in the format [start,end] >> [1,10]
Please enter an increment value >> 1
>>
```



Displacement-1
Velocity-1
Displacement-2
Velocity-2
time
1N force
2N force
3N force
4N force
5N force
6N force
7N force
8N force
9N force
10N force