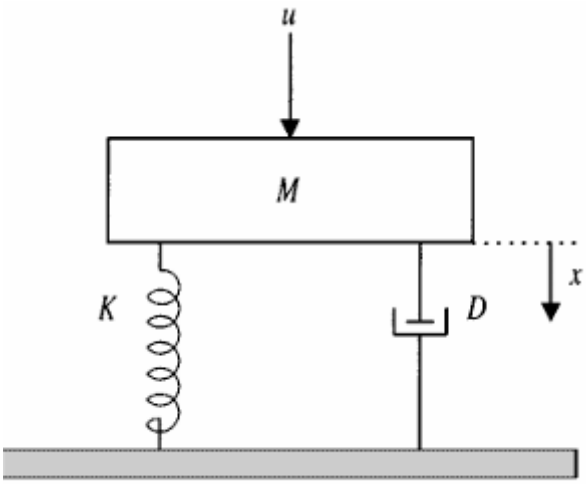
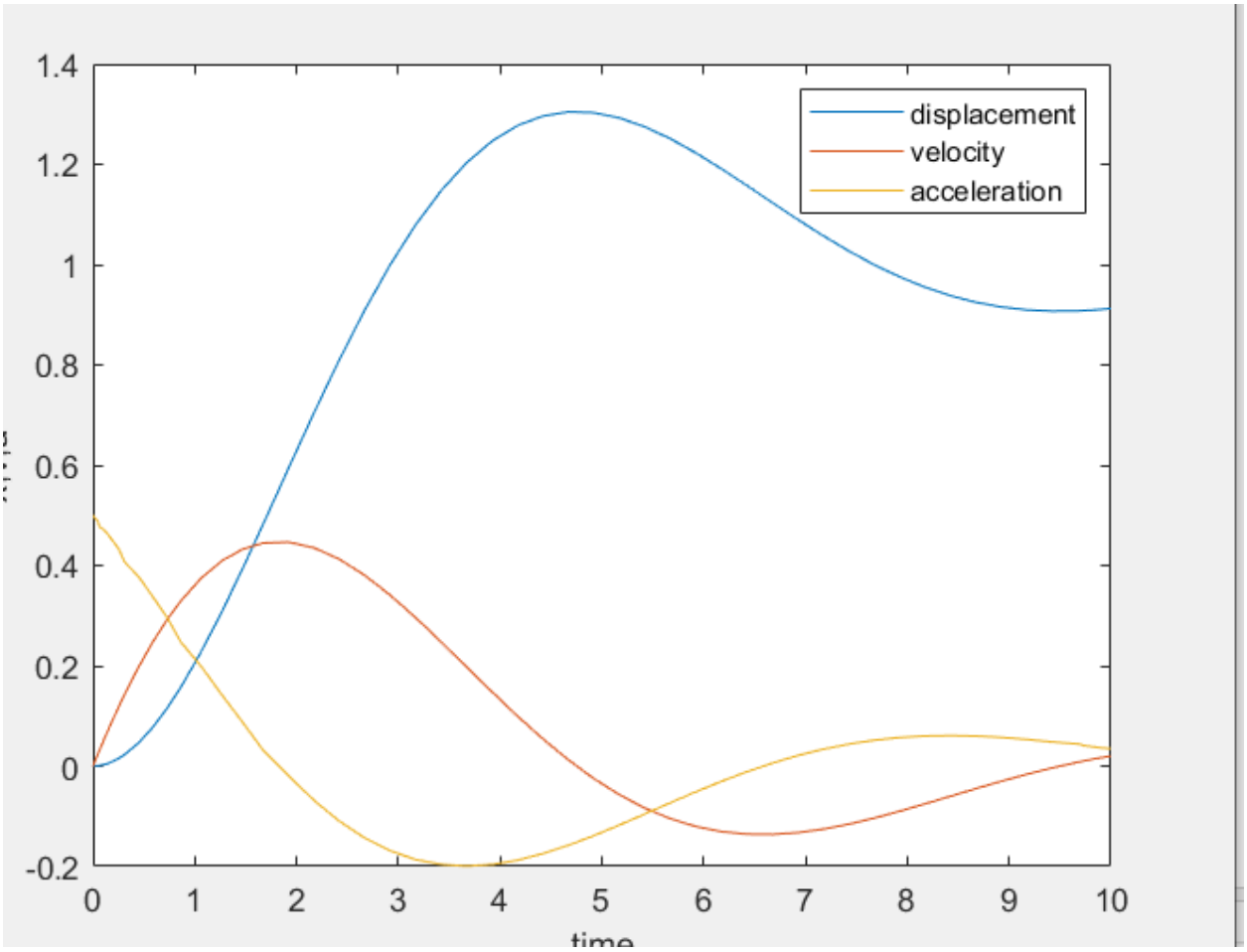
EXAMPLE_1



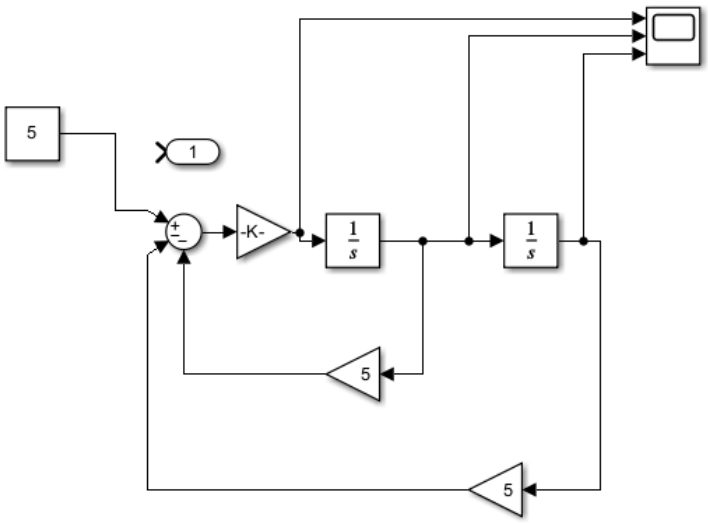
----------<(M File)>----------

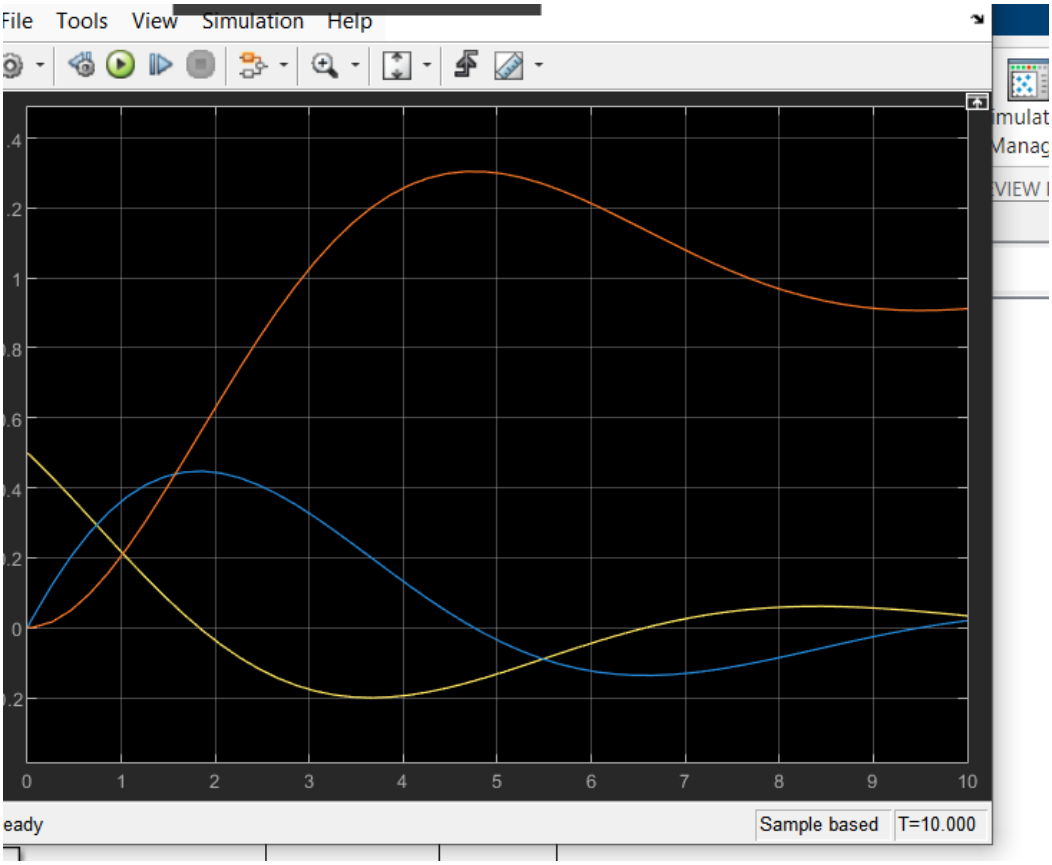
```
X0=[0;0];%initial condiitons are zero
TR=[0 10];%time response RANGE
%t=0:0.1:50;
[t,x]=ode45(@func1,TR,X0);
Displacement=x(:,1)
Velocity=x(:,2)
plot(t, Displacement)
hold on
plot(t,Velocity)
acceleration = gradient(Velocity,t);
hold on
plot(t,[0;acceleration])
ylabel('x,v,a')
xlabel('time')
legend("displacement","velocity","acceleration");

%state variables give the following function
function dx = func1(t,x)
   M=10;B=5;K=5;F=5;
   dx(1)=x(2)%for x dot
   dx(2)=(F-B*x(2)-K*x(1))/M    %for x dot dot
   dx = dx';
```

end

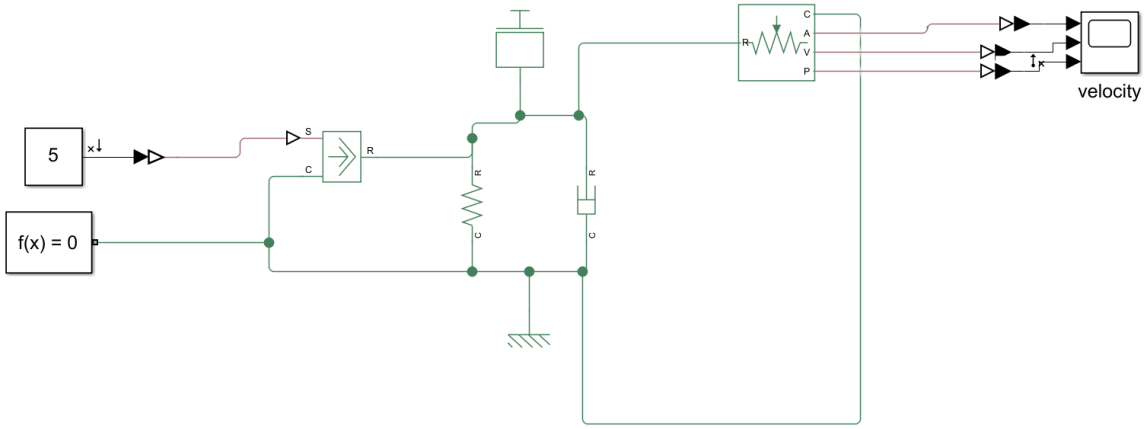

displacement
velocity
acceleration
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
1
2
3
4
5
6
7
8
9
10
time

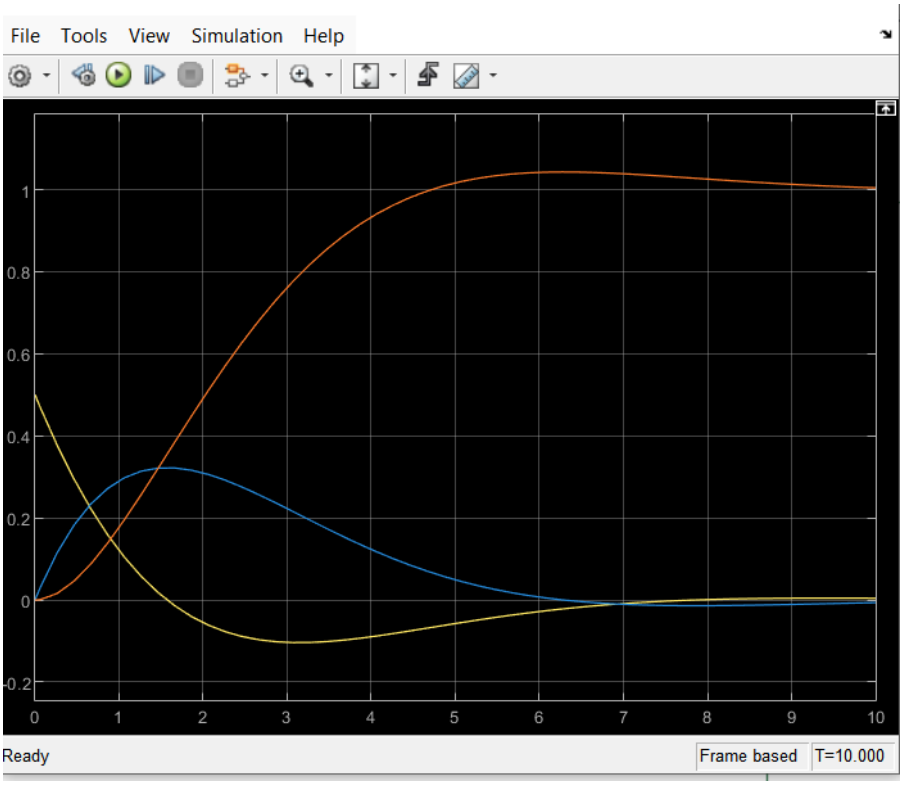
----------<(Simulink)>----------

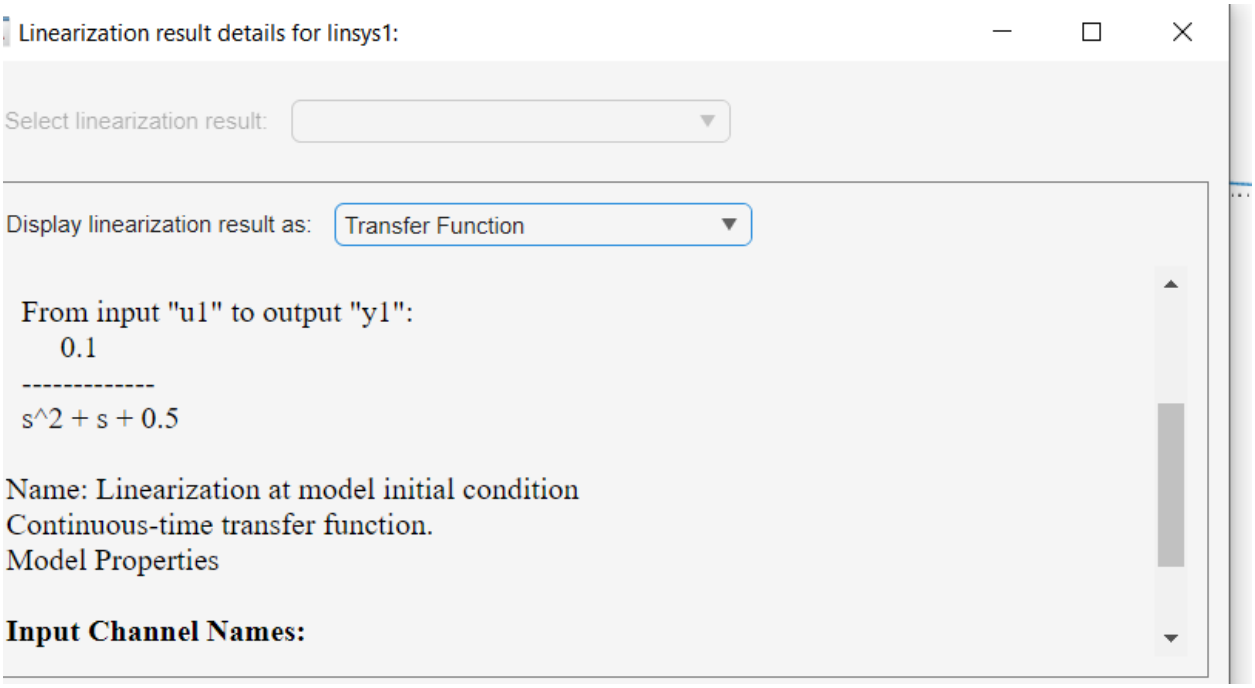

5
1
-K-
1
s
1
s
5
5


File Tools View Simulation Help
.4
.2
1
.8
.6
.4
.2
0
.2
0 1 2 3 4 5 6 7 8 9 10
eady
Sample based
T=10.000

----------<(SIMSCAPE)>----------

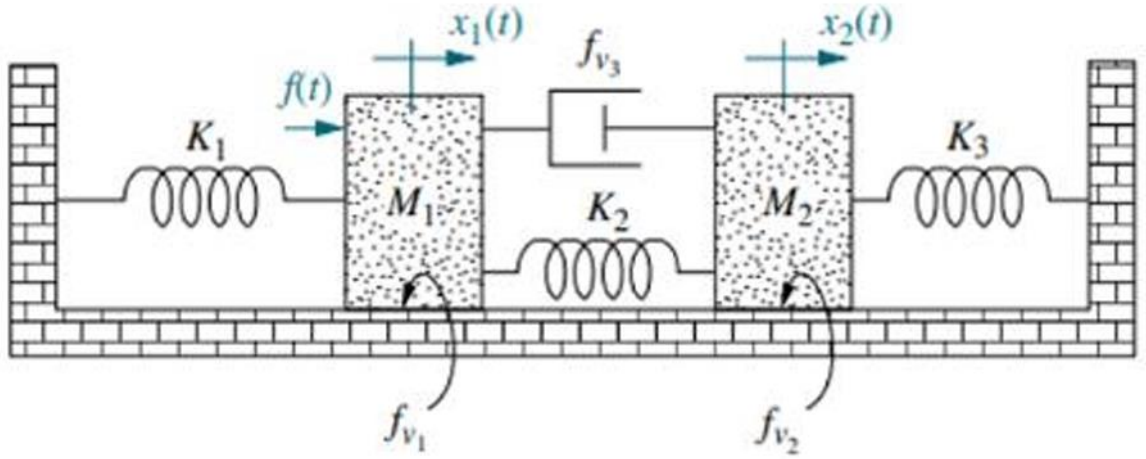

5
f(x) = 0
velocity


File Tools View Simulation Help
1
0.8
0.6
0.4
0.2
0
-0.2
0 1 2 3 4 5 6 7 8 9 10
Ready
Frame based
T=10.000


Linearization result details for linsys1:
Select linearization result:
Display linearization result as:
Transfer Function
From input "u1" to output "y1":
0.1
---
s^2 + s + 0.5
Name: Linearization at model initial condition
Continuous-time transfer function.
Model Properties
Input Channel Names:

EXAMPLE_2

Example#02 (Dual Mass Spring(s) Damper System)

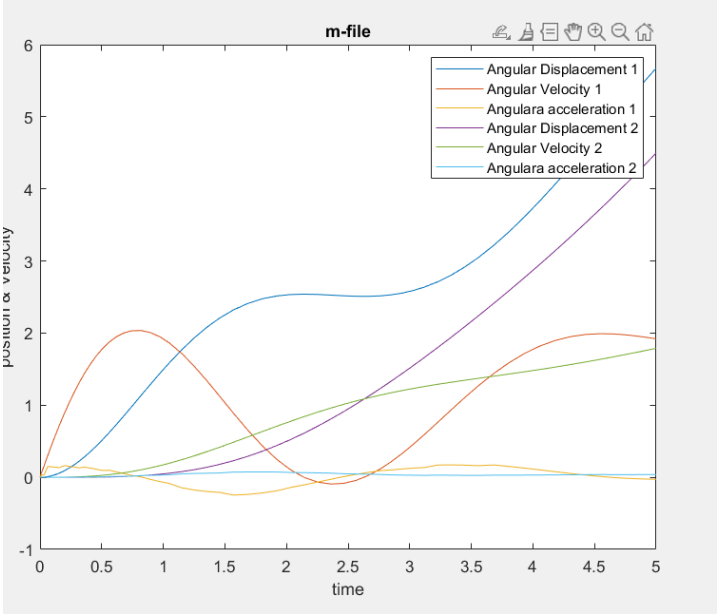


Simulink Results:

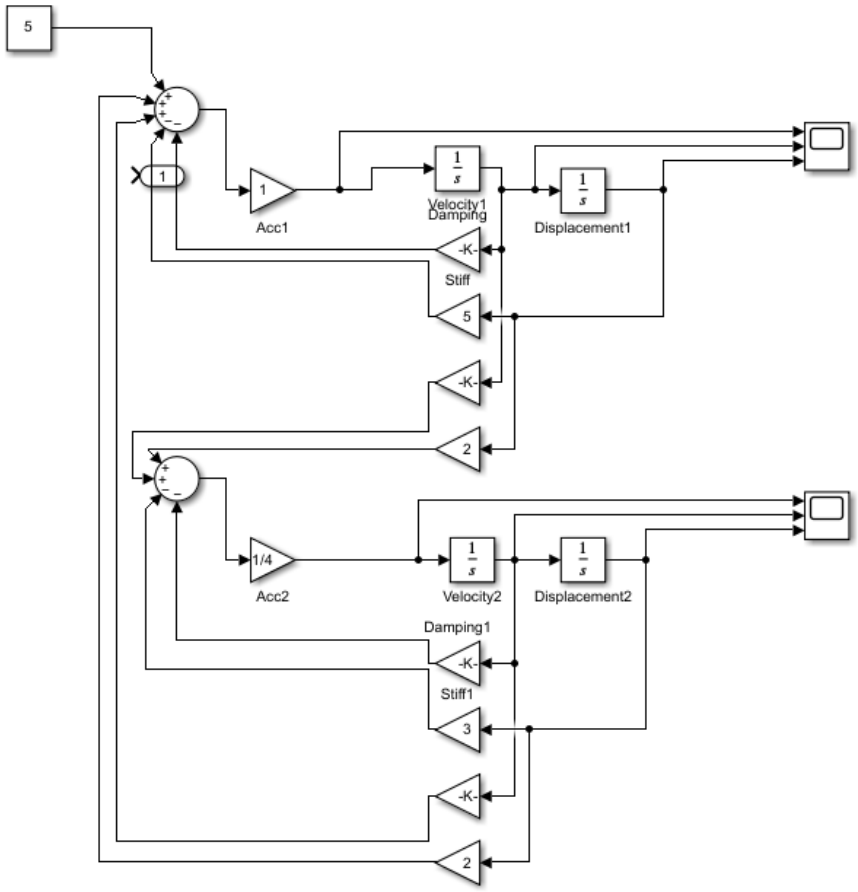
----------<(M File)>----------

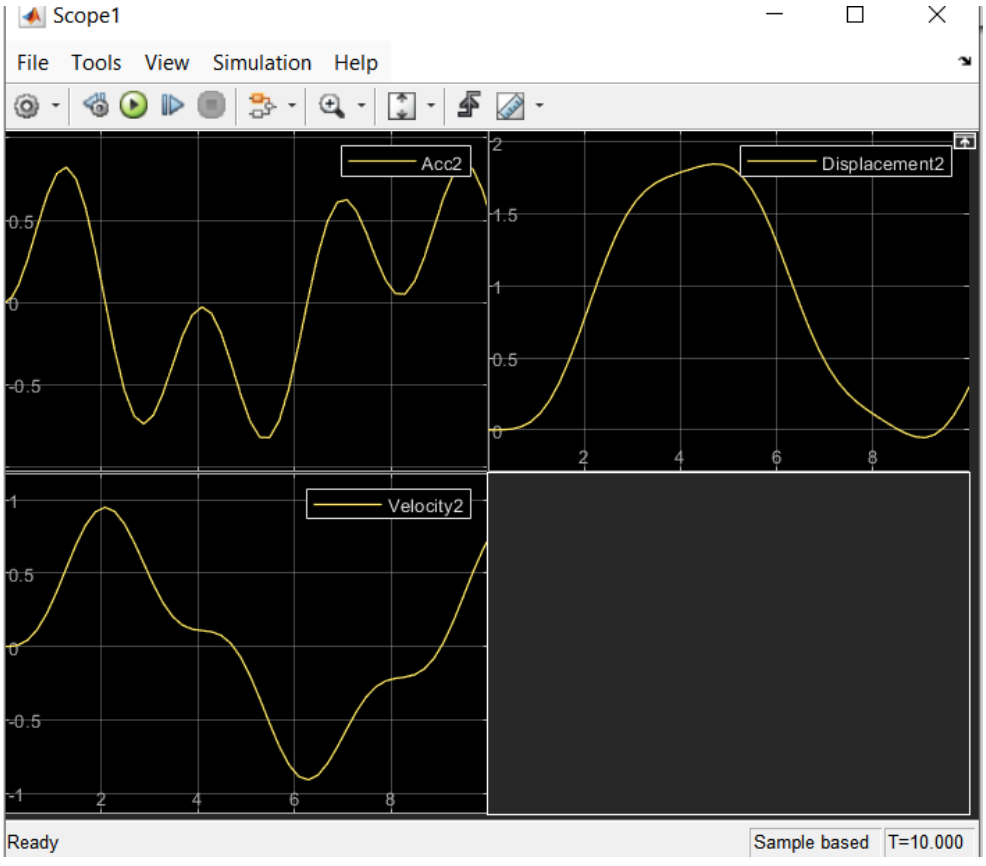
```
% For constant tourque 5Nm
clc
clear
TR = [0 5]; % time RANGE
X0 = [0;0;0;0];%initial conditions
[t,z] = ode45(@func1, TR, X0);%calling thr ide solver to solve by function
%storing given array as vectors
theta1 = z(:, 1);
AngVel1 = z(:, 2);
theta2 = z(:, 3);
AngVel2 = z(:, 4);

%plotting the angular displacements and velocities
acc1 = diff(AngVel1);
acc2= diff(AngVel2);
plot(t,theta1,t,AngVel1,t,[0;acc1],t,theta2,t,AngVel2,t,[0;acc2]);
xlabel('time')
legend('Angular Displacement 1','Angular Velocity 1','Angulara acceleration 1','Angular Displacement 2','Angular Velocity 2','Angulara acceleration 2')
```

```
ylabel('position & Velocity')

title("m-file")

%function containing the differentialequations

function dx = func1(~, x)

% Values of Coefficients

J1=1; J2=10; D1=0.9; D2=0.02; k=3;T=5;


% State Equations

dx(1) = x(2);

dx(3) = x(4);

dx(2) = (T-D1*x(2)-k*x(1)+k*x(3))/J1;

dx(4) = (-k*x(3)-D2*x(4)+k*x(1))/J2;

dx = dx';

end
```

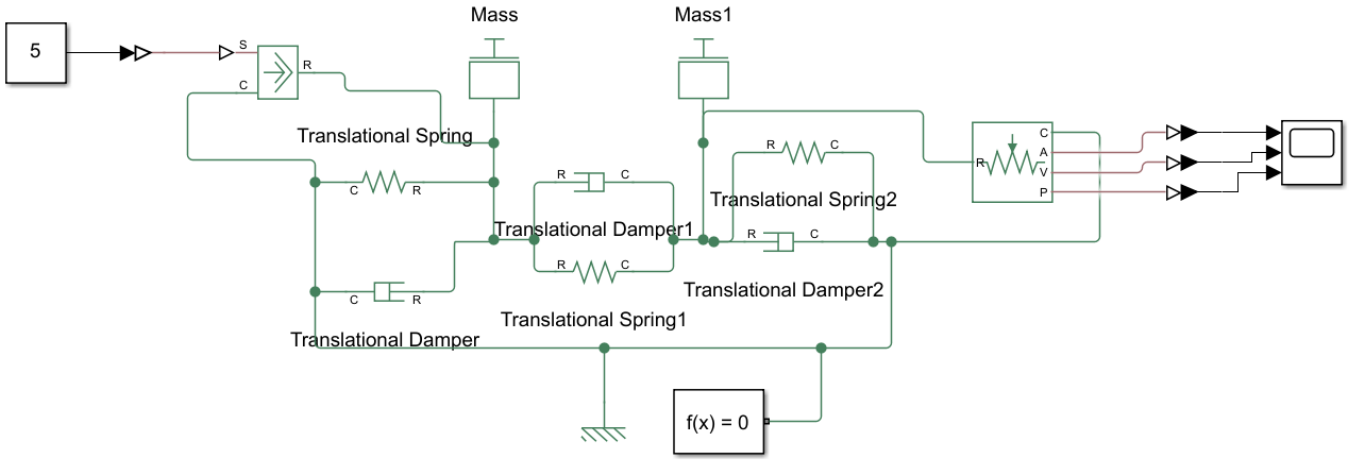


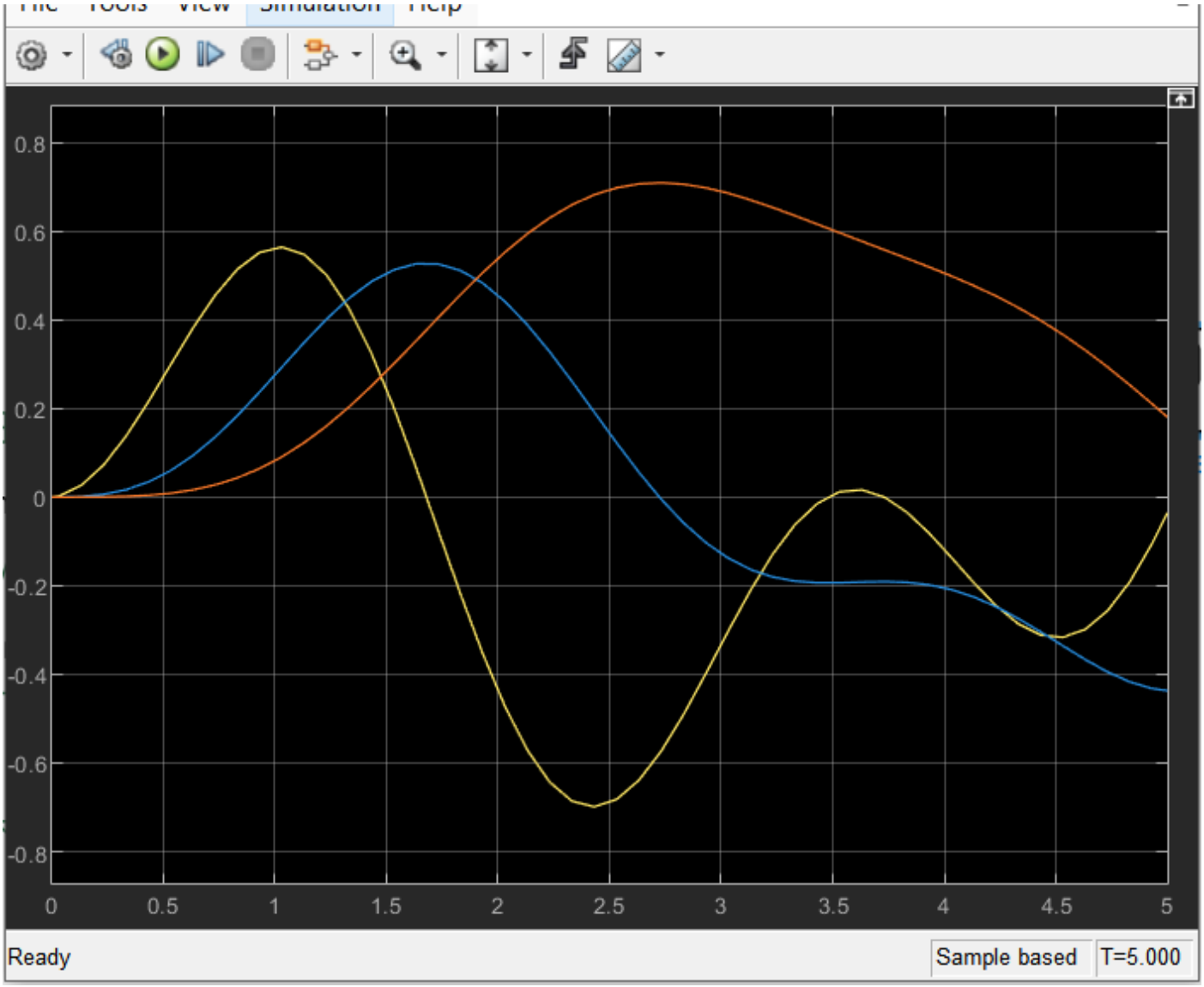
----------<(Simulink)>----------

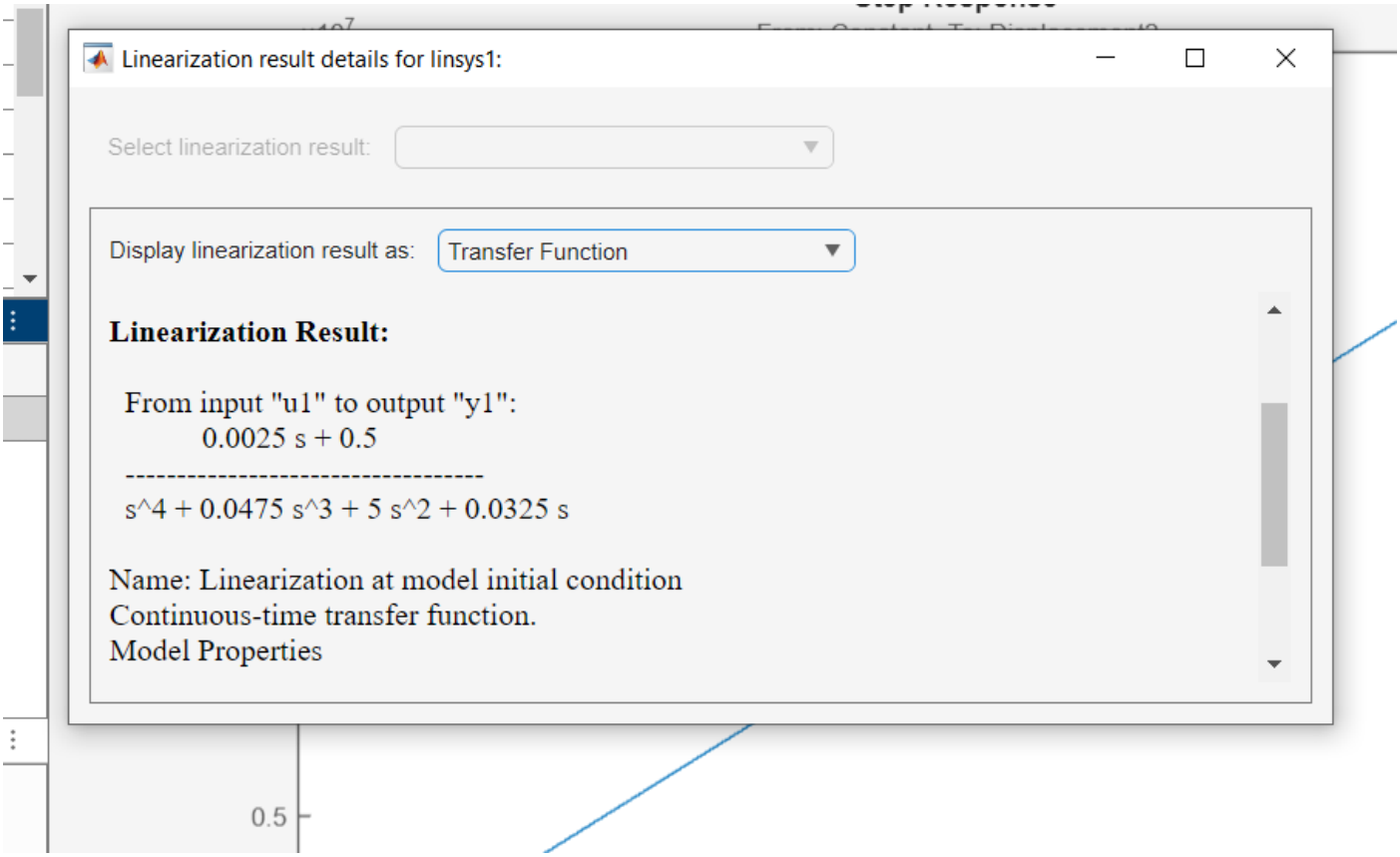




----------<(SIMSCAPE)>----------

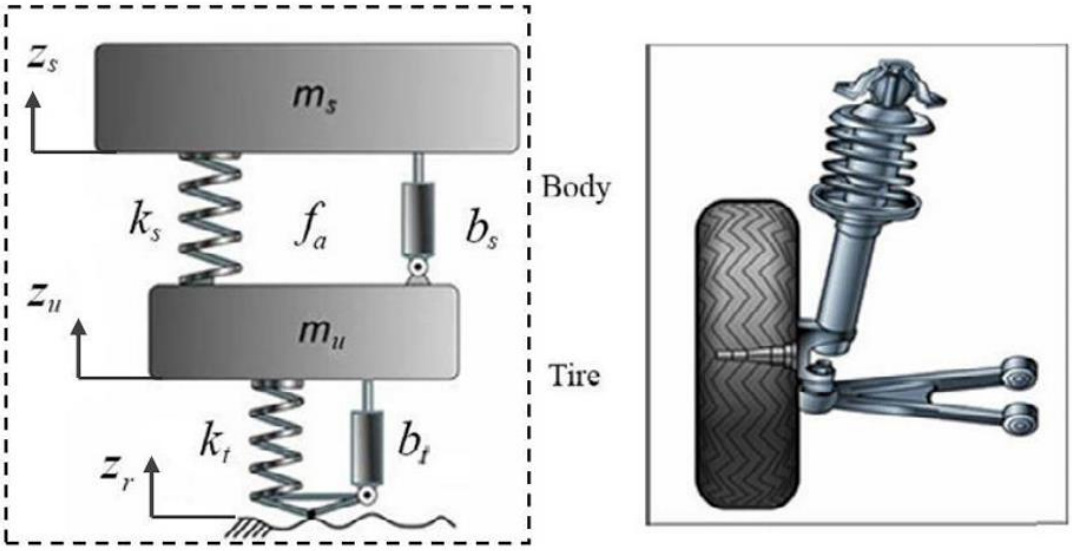

5
Mass
Mass1
Translational Spring
Translational Damper1
Translational Spring2
Translational Damper2
Translational Spring1
Translational Damper
f(x) = 0


0.8
0.6
0.4
0.2
0
-0.2
-0.4
-0.6
-0.8
0
0.5
1
1.5
2
2.5
3
3.5
4
4.5
5
Ready
Sample based
T=5.000


Linearization result details for linsys1:
Select linearization result:
Display linearization result as:
Transfer Function
Linearization Result:
From input "u1" to output "y1":
0.0025 s + 0.5
s^4 + 0.0475 s^3 + 5 s^2 + 0.0325 s
Name: Linearization at model initial condition
Continuous-time transfer function.
Model Properties
0.5

EXAMPLE_3

Example#05 (Quarter Car Model)

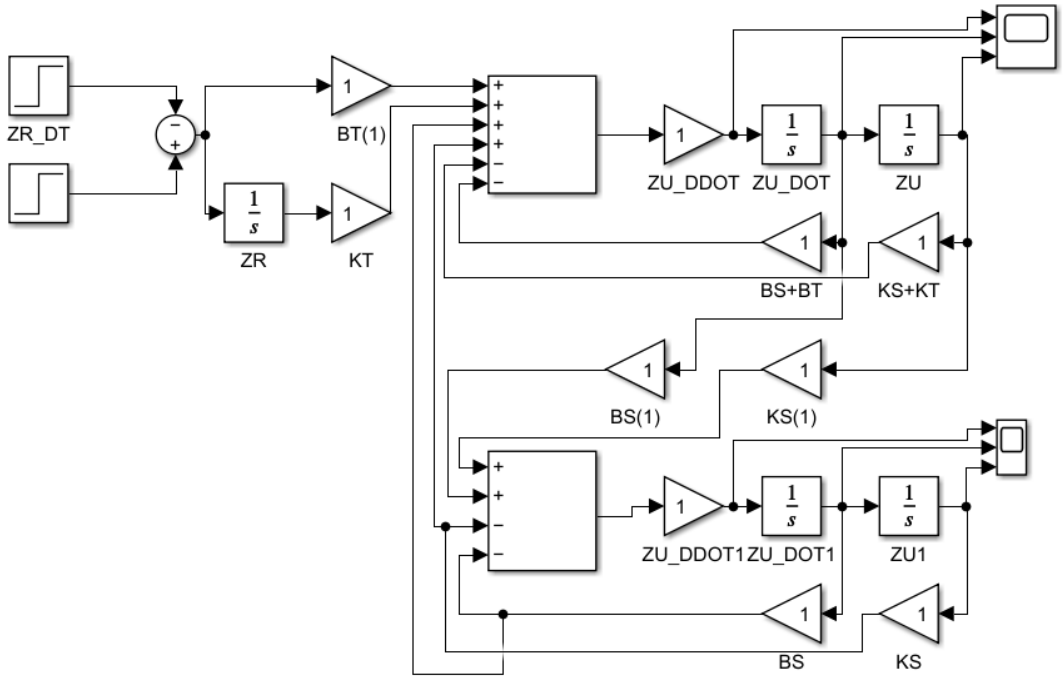


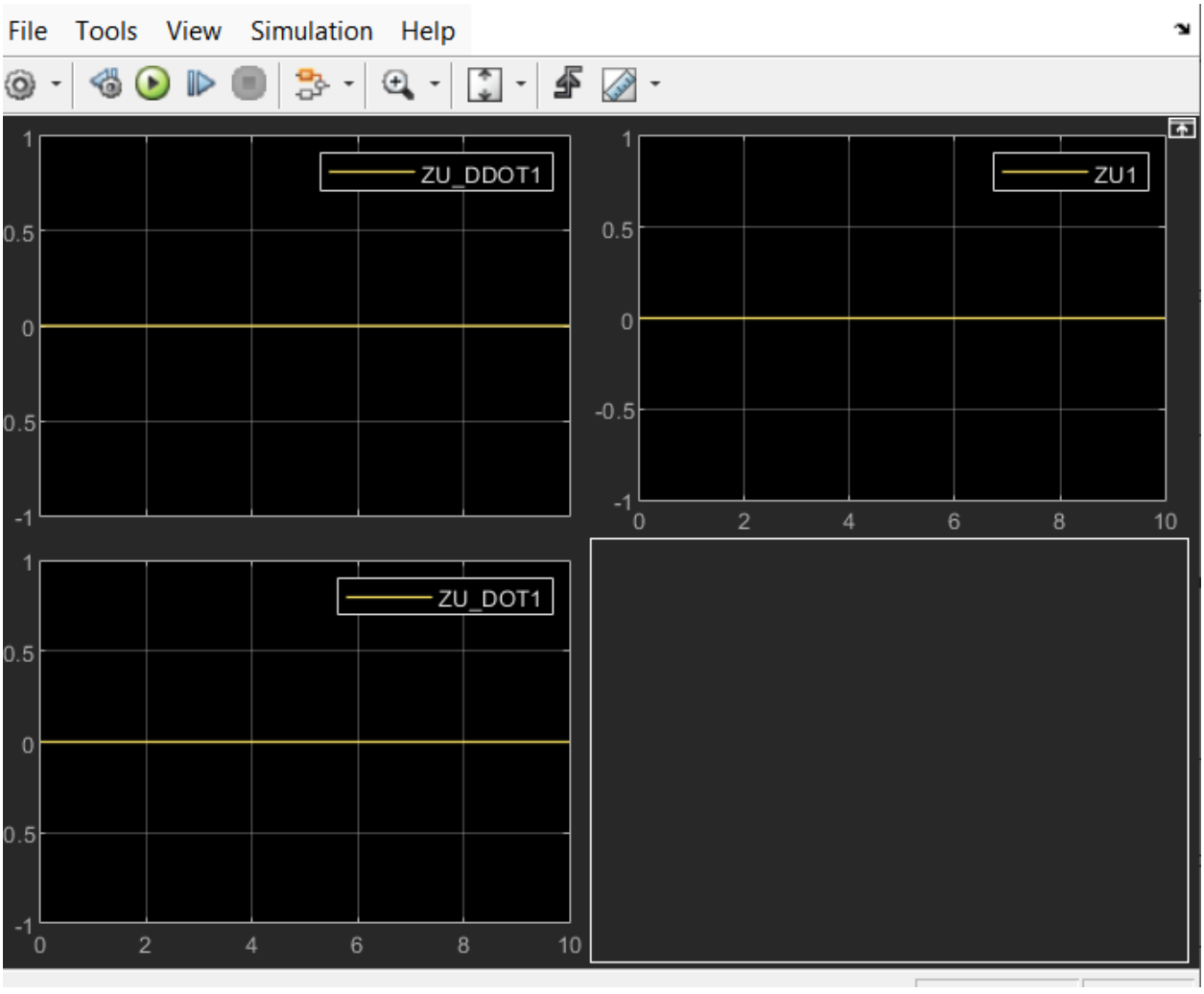
SIMSCAPE MODEL

----------<(M File)>----------

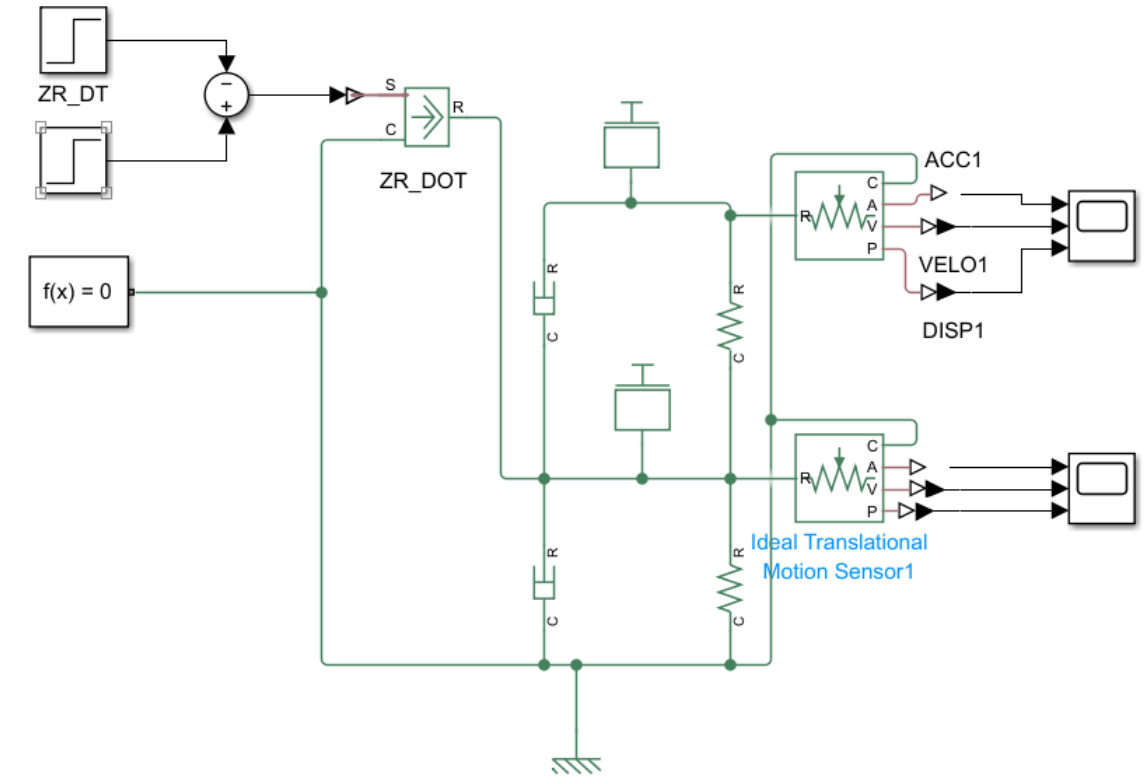
```
syms s zrdot

A=[(mu*s^2+(bs+bt)*s+ks+kt) , -(ks+bs*s) ; -(ks+bs*s) , (ms*s^2+ks+bs*s)];

B=[(kt/s+bt)*zrdot;0];

C=A\B;

zs=C(1);

G=zs/zrdot;

G=collect(G,s);

[num,den]=numden(G);

num=sym2poly(num);den=sym2poly(den);

num=num/den(1); %dividing by den(1) means dividing by the leading coefficient of denominator i.e., 20 in this case. Done to match the outputs.

den=den/den(1);

G=tf(num,den)
```

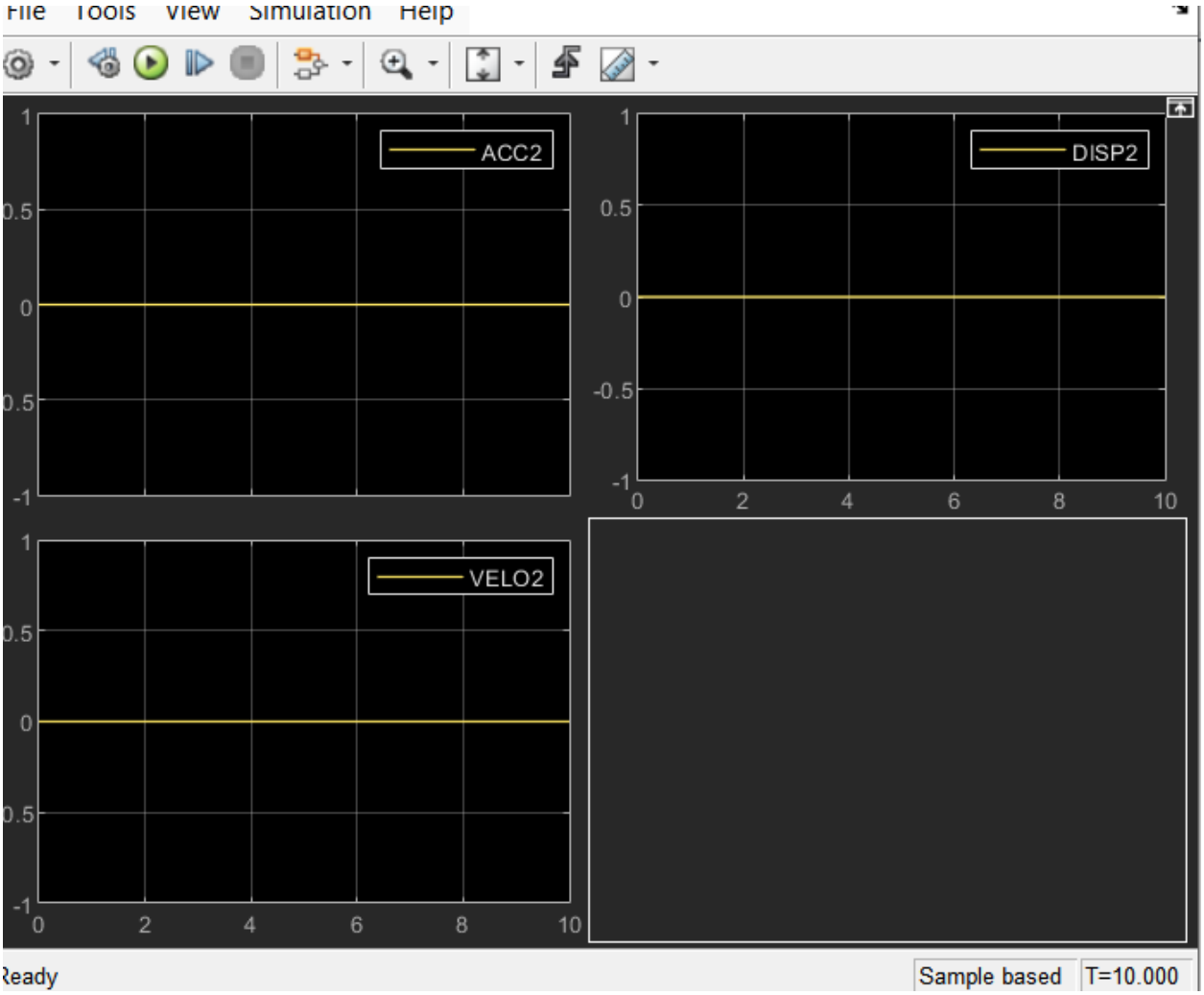
----------<(Simulink)>----------


ZR_DT
BT(1)
ZR
KT
ZU_DDOT
ZU_DOT
ZU
BS+BT
KS+KT
BS(1)
KS(1)
ZU_DDOT1
ZU_DOT1
ZU1
BS
KS


File
Tools
View
Simulation
Help
ZU_DDOT1
ZU1
ZU_DOT1

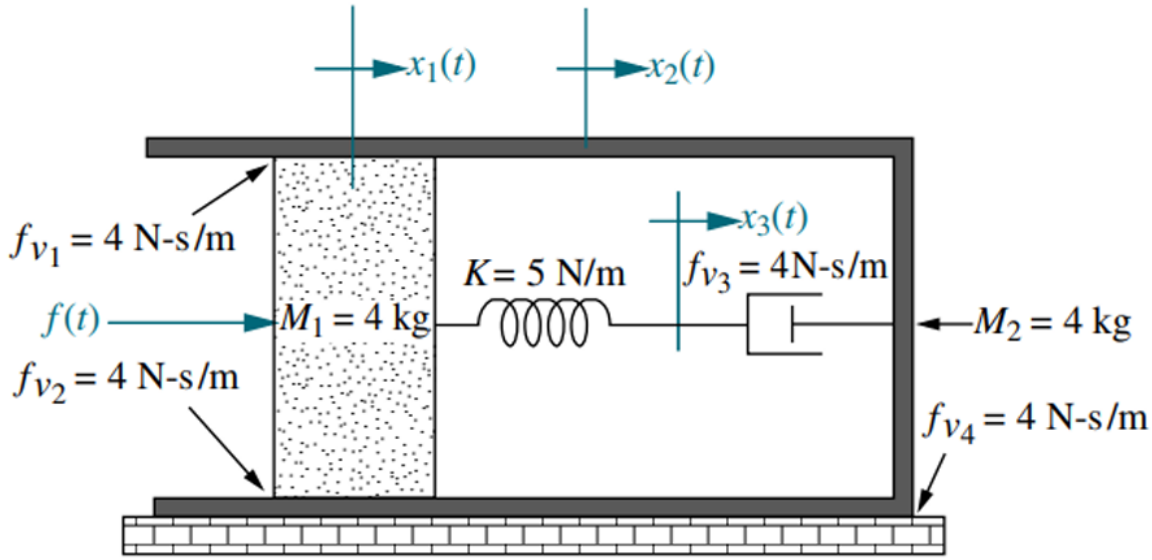
----------<(SIMSCAPE)>----------


ZR_DT
S
C
R
ZR_DOT
f(x) = 0
ACC1
VELO1
DISP1
C
A
V
P
R
C
Ideal Translational
Motion Sensor1


File Tools View Simulation Help
ACC2
DISP2
VELO2
1
0.5
0
-0.5
-1
0
2
4
6
8
10
Ready
Sample based
T=10.000

EXAMPLE_4

Example#06 (Framed System: Translational Piston Spring System)

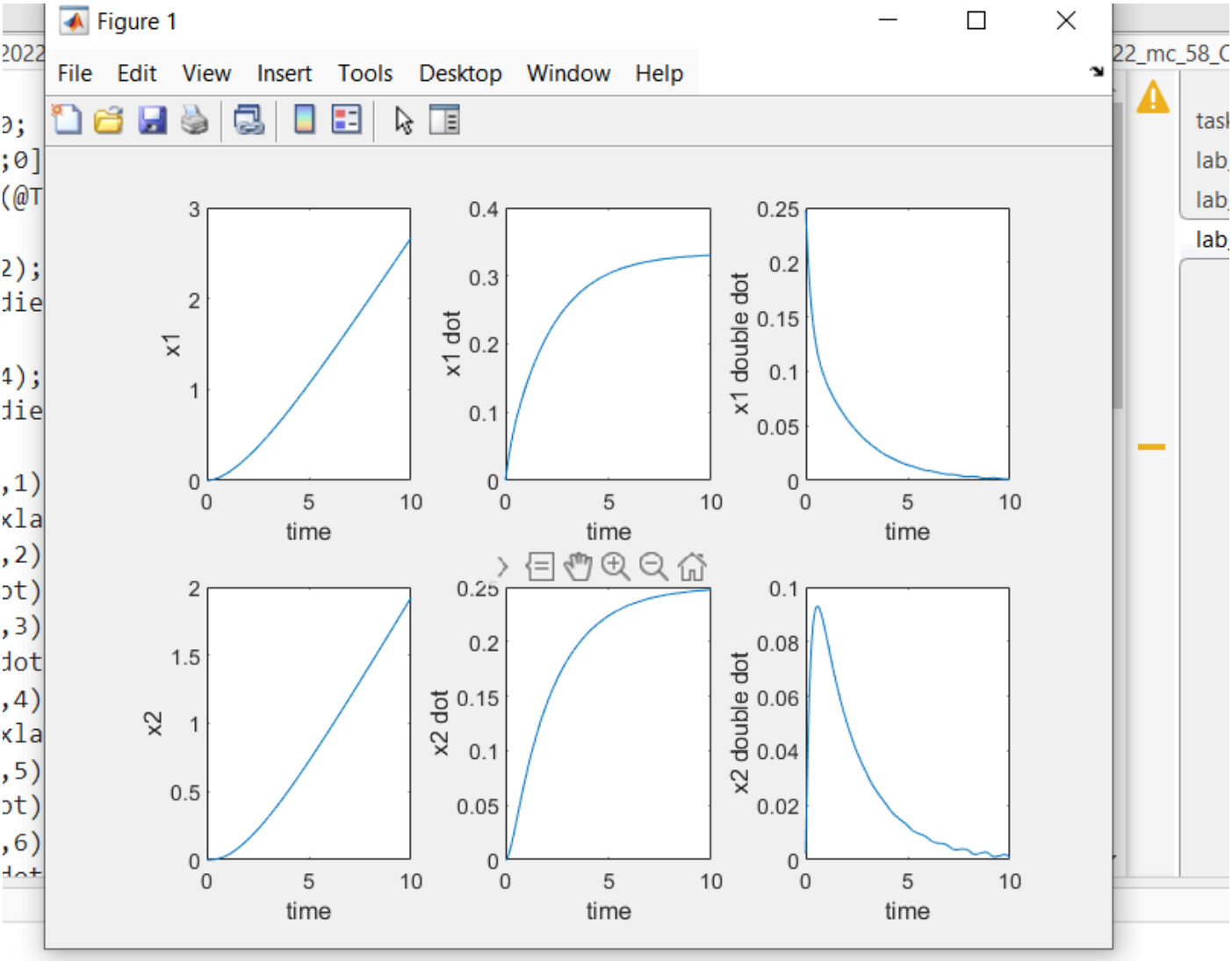


Simulink Model

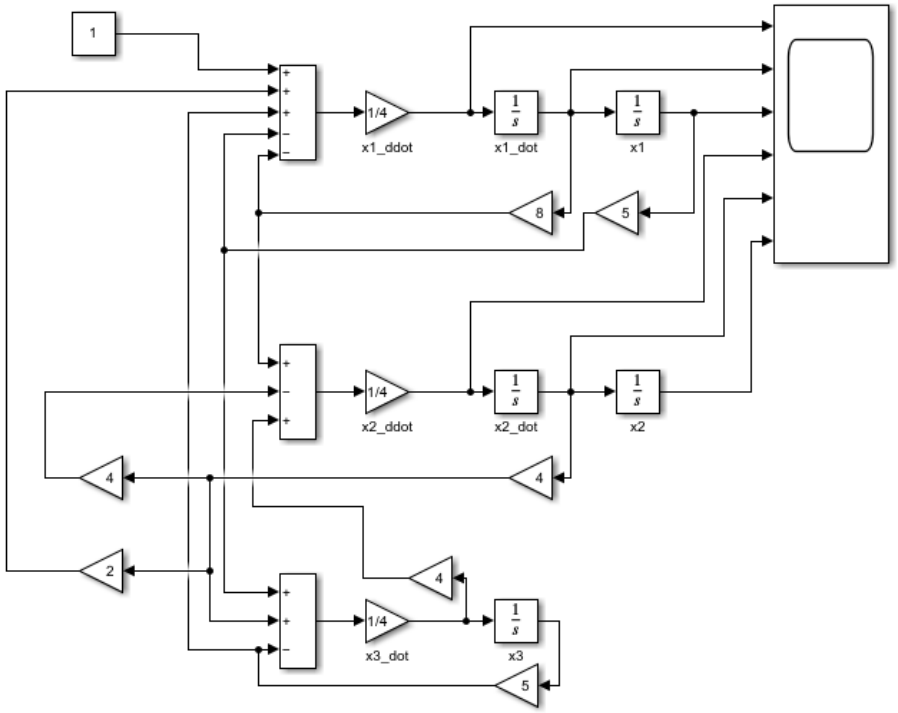
----------<(M File)>----------

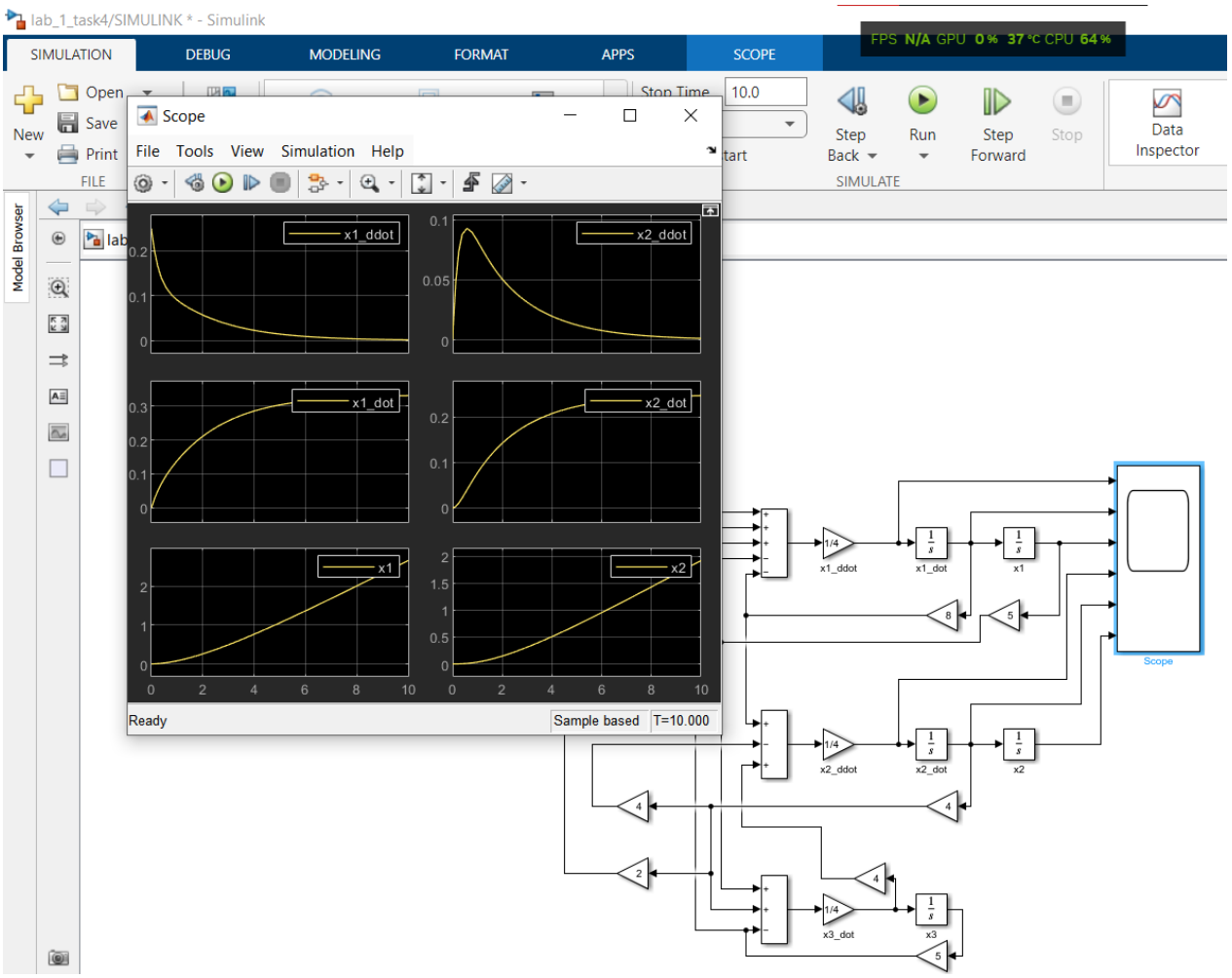
```
clc;
TR=0:0.01:10;
x0=[0;0;0;0;0];
[t,x]=ode45(@Task2Fun,TR,x0);
x1=x(:,1);
x1_dot=x(:,2);
x1_ddot=gradient(x1_dot)./gradient(t);
x2=x(:,3);
x2_dot=x(:,4);
x2_ddot=gradient(x2_dot)./gradient(t);
subplot(2,3,1);
plot(t,x1);xlabel('time');ylabel('x1');
subplot(2,3,2);
plot(t,x1_dot);xlabel('time');ylabel('x1 dot');
subplot(2,3,3);
plot(t,x1_ddot);xlabel('time');ylabel('x1 double dot')
subplot(2,3,4);
plot(t,x2);xlabel('time');ylabel('x2');
```

```
subplot(2,3,5);
plot(t,x2_dot);xlabel('time');ylabel('x2 dot');
subplot(2,3,6);
plot(t,x2_ddot);xlabel('time');ylabel('x2 double dot')
%%%%%%%%%%%%%%%%%%%%
function dy=Task2Fun(t,y)
   f=1;
   dy(1)=y(2);
   dy(2)=1/4*(f - 8*y(2) - 5*y(1) + 8*y(4) + 5*y(5));
   dy(3)=y(4);
   dy(5)=1/4*(5*y(1) + 4*y(4) - 5*y(5));
   dy(4)=1/4*(8*y(2) + 4*dy(5) - 16*y(4));
   dy=dy';
end
```

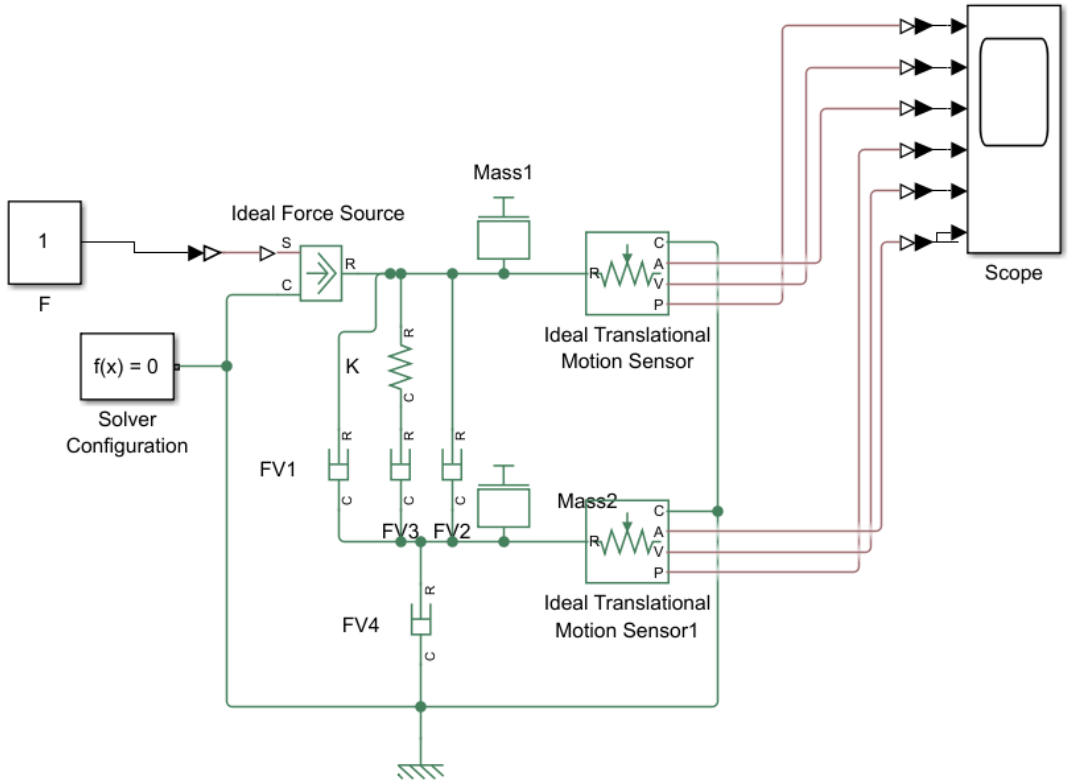


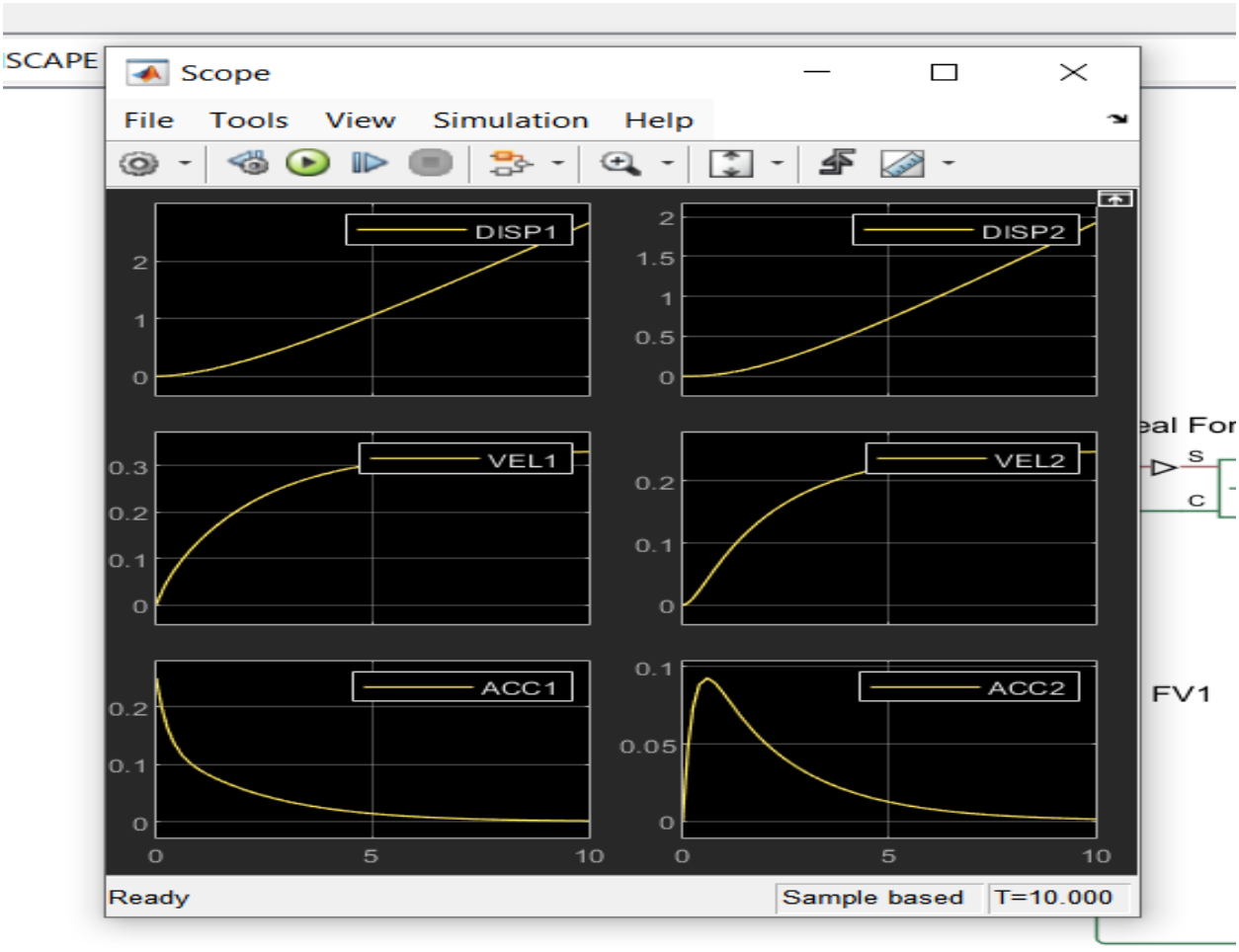
----------<(Simulink)>----------

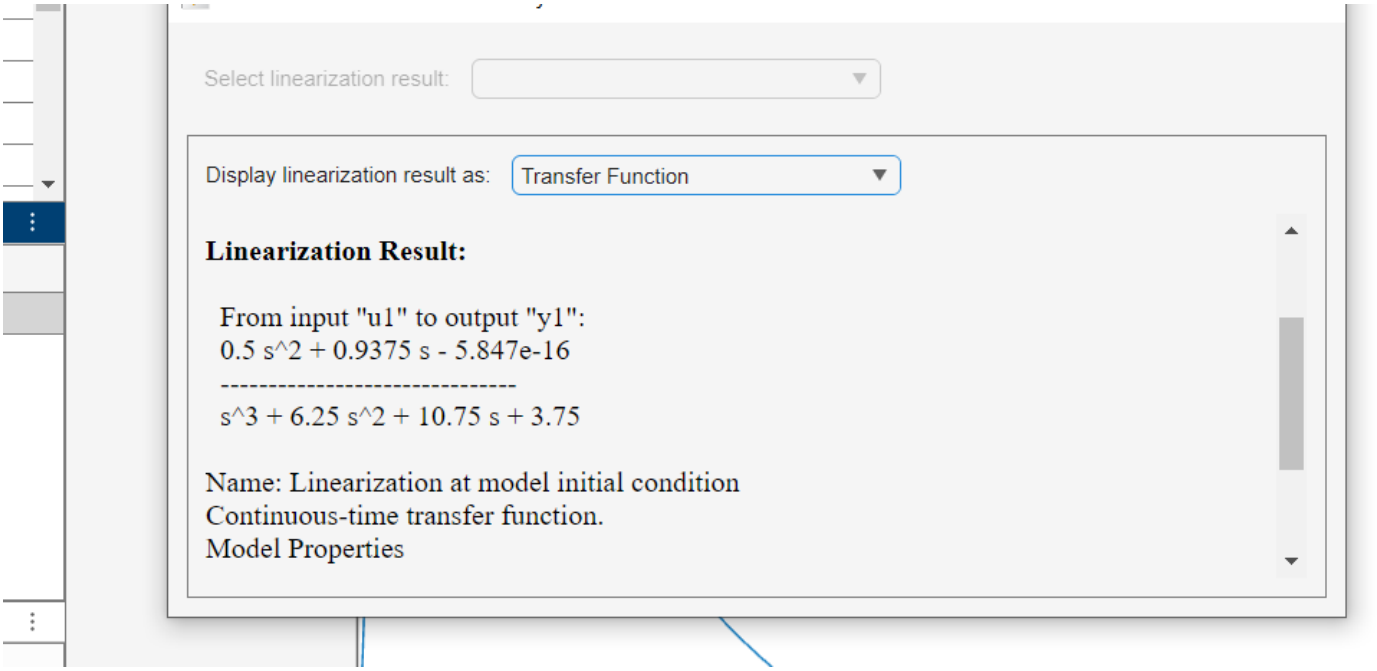

1
1/4
x1_ddot
x1_dot
x1
8
5
1/4
x2_ddot
x2_dot
x2
4
4
2
4
1/4
x3_dot
x3
5


lab_1_task4/SIMULINK * - Simulink
SIMULATION
DEBUG
MODELING
FORMAT
APPS
SCOPE
FPS N/A GPU 0% 37°C CPU 64%
Open
Save
Print
New
FILE
Stop Time
10.0
Step Back
Run
Step Forward
Stop
Data Inspector
SIMULATE
Scope
File Tools View Simulation Help
x1_ddot
x2_ddot
x1_dot
x2_dot
x1
x2
Ready
Sample based
T=10.000
Model Browser
Scope

----------<(SIMSCAPE)>----------

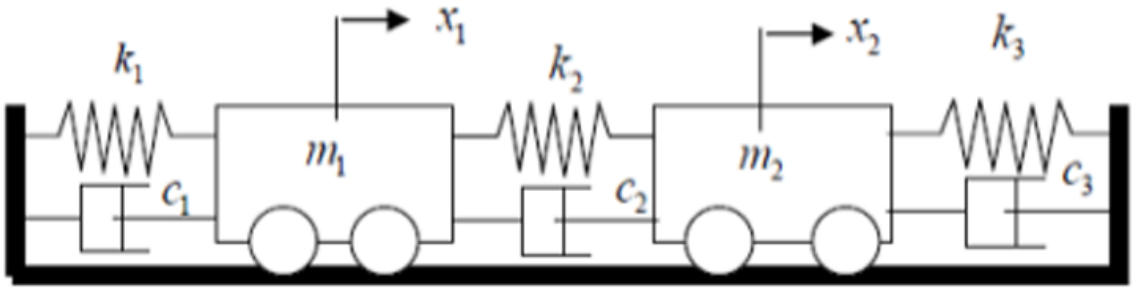

1
F
Ideal Force Source
S
C
R
Mass1
C
A
V
P
Ideal Translational
Motion Sensor
Scope
f(x) = 0
Solver
Configuration
K
FV1
FV3
FV2
Mass2
Ideal Translational
Motion Sensor1
FV4


SCAPE
Scope
File
Tools
View
Simulation
Help
DISP1
DISP2
VEL1
VEL2
ACC1
ACC2
Ready
Sample based
T=10.000
FV1


Select linearization result:
Display linearization result as:
Transfer Function
Linearization Result:
From input "u1" to output "y1":
0.5 s^2 + 0.9375 s - 5.847e-16
---
s^3 + 6.25 s^2 + 10.75 s + 3.75
Name: Linearization at model initial condition
Continuous-time transfer function.
Model Properties

Example_5

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.**



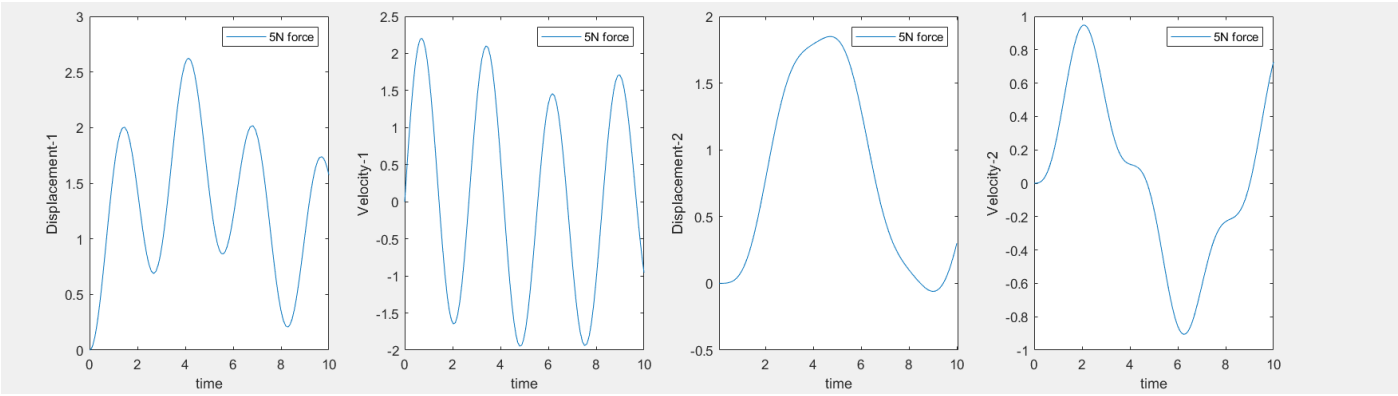
----------<(M File)>----------

```
function dy=Task4Fun(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
```

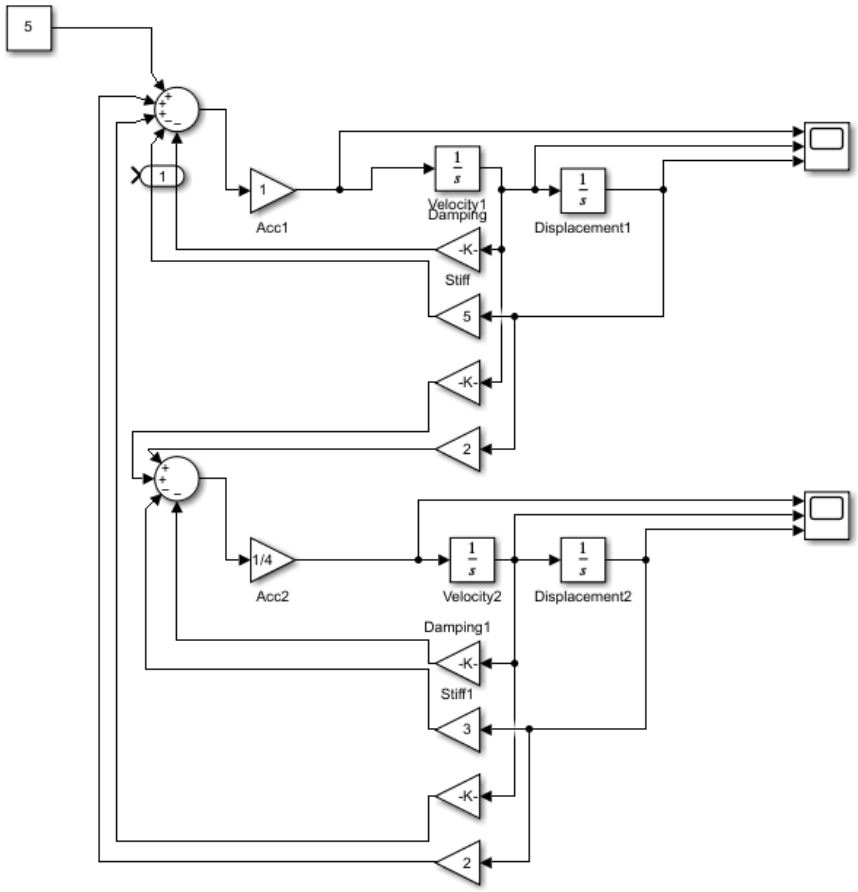
```
clc;
query=input('Do you want to analyze the system at single Force (5N) or on a range of forces? (Single/Range) >> ','s');
TR = [0 10];
```

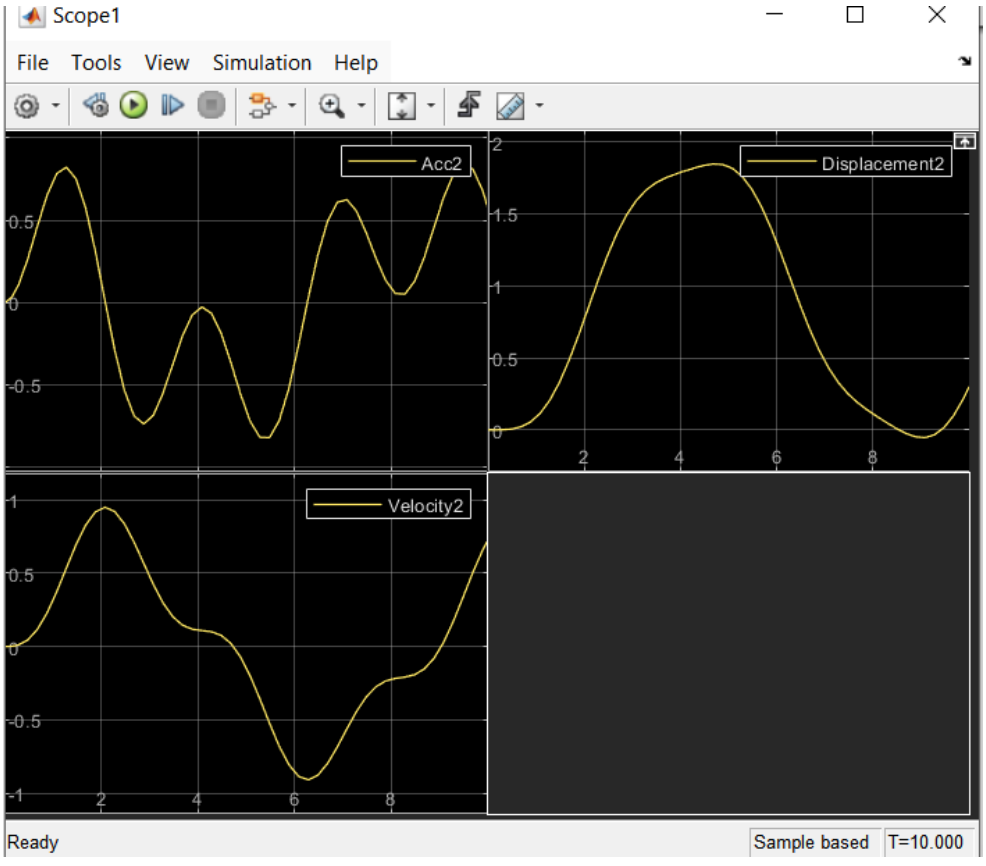
```
X0 = [0;0;0;0];
if query=="Single" || query=="single"
  range=[5,5];
  inc=1;
elseif query=="Range" || query=="range"
  range=input('Please enter a start and an end value for the force in the format [start,end] >> ');
  inc=input('Please enter an increment value >> ');
end

 for F=range(1):inc:range(2)
  [t,y]=ode45(@(t,y) Task4Fun(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
```

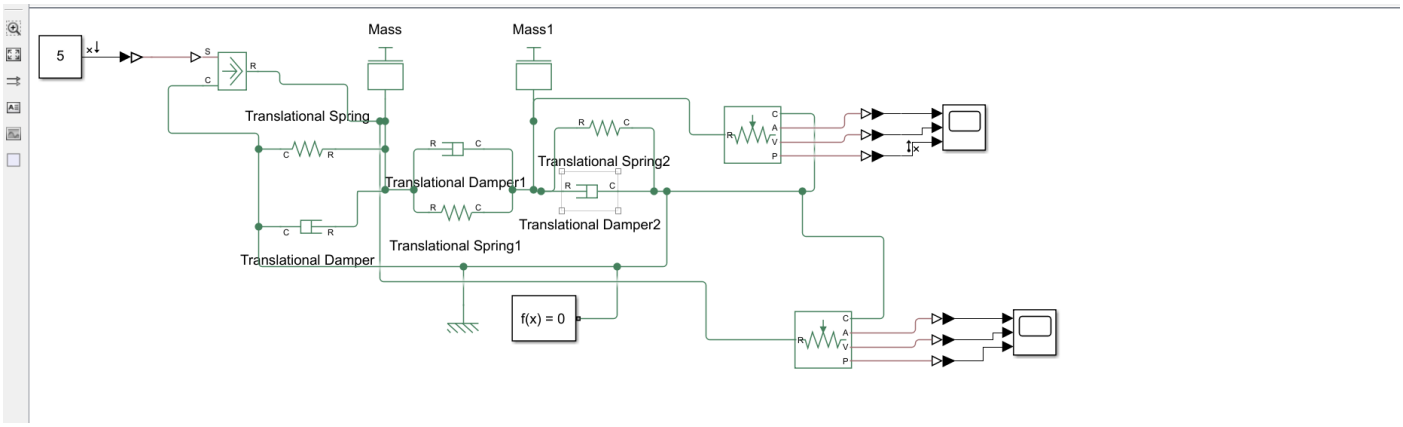


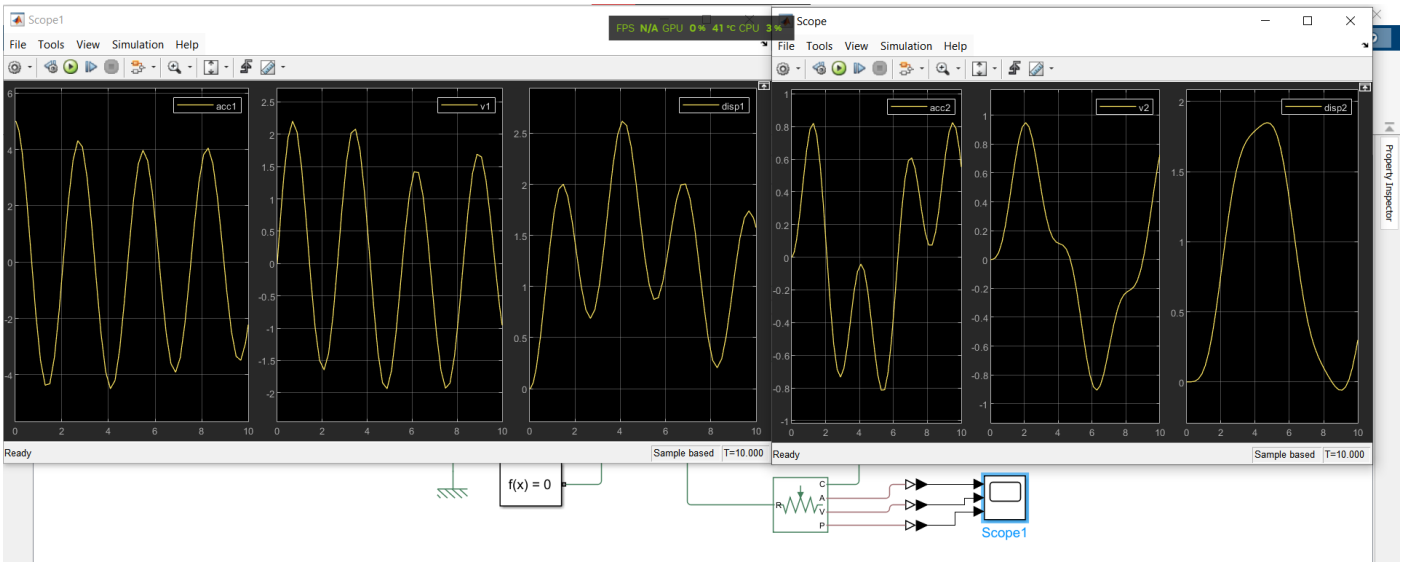
----------<(Simulink)>----------

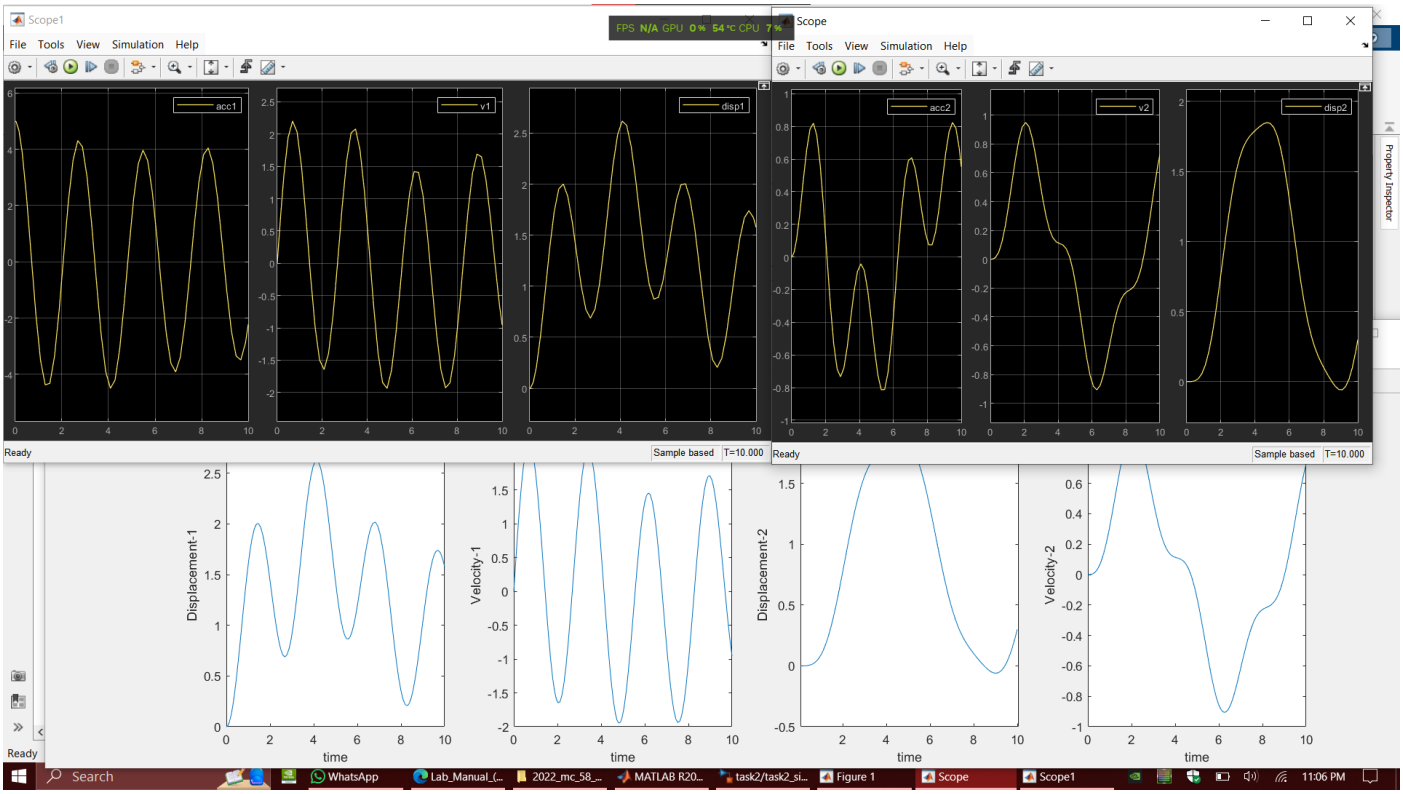




----------<(SIMSCAPE)>----------


5
Mass
Mass1
Translational Spring
Translational Damper1
Translational Spring2
Translational Damper2
Translational Spring1
Translational Damper
f(x) = 0


Scope1
File Tools View Simulation Help
acc1
v1
disp1
Ready
Sample based
T=10.000
Scope
File Tools View Simulation Help
acc2
v2
disp2
Ready
Sample based
T=10.000
Property Inspector
f(x) = 0
Scope1


Scope1
File Tools View Simulation Help
acc1
v1
disp1
Ready
Sample based
T=10.000
Scope
File Tools View Simulation Help
acc2
v2
disp2
Ready
Sample based
T=10.000
Property Inspector
Displacement-1
Velocity-1
Displacement-2
Velocity-2
time
Ready
Search
WhatsApp
Lab_Manual_(...
2022_mc_58_...
MATLAB R20...
task2/task2_si...
Figure 1
Scope
Scope1
11:06 PM

Model Linearizer - task2

LINEAR ANALYSIS | ESTIMATION | PLOTS AND RESULTS | STEP PLOT 1

FPS 3 GPU 0% 38°C CPU 5%

Load Session | Save Session | Preferences — FILE

Analysis I/Os: Model I/Os
Operating Point: Model Initial Condition
Parameter Variations: None — SETUP

Result Viewer | Linearization Advisor | More Options — OPTIONS

Step Plot 1 | Step | Bode — LINEARIZE

MATLAB Workspace

Name	Value
F	5
i	4
inc	1
out	1×1 SimulationOutput
query	'single'
range	[5,5]
t	141×1 double

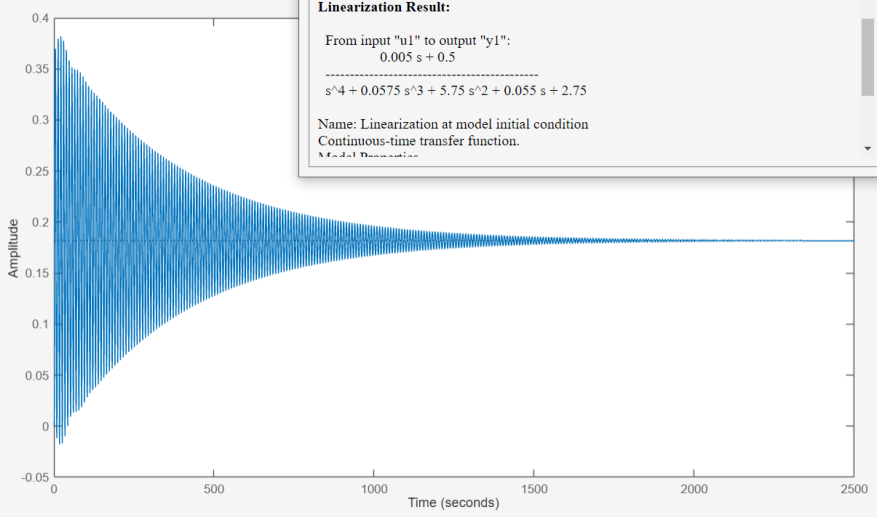
Linear Analysis Workspace

Name	Value
linsys1	1×1 ss

Variable Preview

```
Linearization at model initial condition:
State-space model with 1 outputs, 1 inputs, and 4
states.
```

Step Plot 1



Linearization result details for linsys1:

Select linearization result:

Display linearization result as: Transfer Function

Size: 1 inputs, 1 outputs, 4 states

Linearization Result:

From input "u1" to output "y1":

$$\frac{0.005\ s + 0.5}{s^4 + 0.0575\ s^3 + 5.75\ s^2 + 0.055\ s + 2.75}$$

Name: Linearization at model initial condition
Continuous-time transfer function.
Model Properties

The linearization result "linsys1" is created in the Linear Analysis Workspace.

Search | WhatsApp | Lab_Manual_(1-9)[2].p... | 2022_mc_58_CS_lab_1... | MATLAB R2024a | 2022_mc_58_Control_s... | 11:11 PM