Ross Smyth

2/26/20

Dynamic Systems

Project 2

In this report a spring-mass-damper system modeled with the block-based modeling software Simulink and using the state-space method of solving dynamic systems will be compared with previously validated solutions that were solved using the inverse Laplace method of solving the differential equations. Six different situations will be analyzed and compared with varying initial conditions and externals forces between the systems. Below in Figure 1 the Simulink model that was used for the systems is shown. On the right is the actual equation of motion represented in this system, and on the left is the logic to choose the external force, or lack of, to apply to the system. This program was created by hand and then ran with a MATLAB program to automate the process of acquiring the data and plotting it, and this MATLAB program is shown in Append 2.

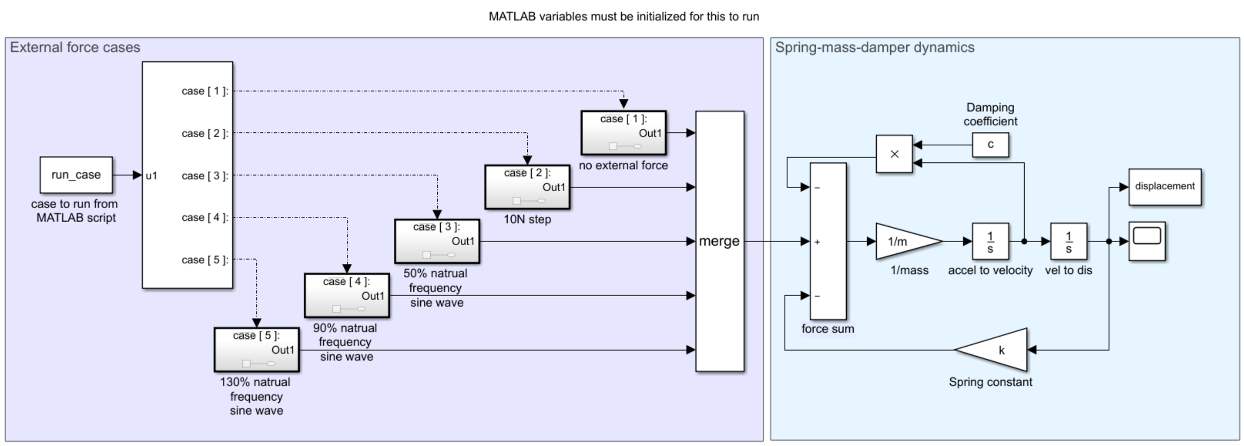


Figure 1: Simulink model of mass-spring-damper system

Table 1: System constants throughout this report

Parameter	Value	Units
Mass	23	kg
Spring constant	19990	N/m

This system can be modeled with the equation of motion defined below in **Error! Reference source not found.**, which can then be transformed with the Laplace transform to simplify the equation into the transfer function shown in **Error! Reference source not found.**. This equation of motion was defined by the free-body diagram that can be seen in Appendix 1. Below this equation of motion is the state-space representation of this equation of motion in *Equation 2*, which did not change throughout each analysis just the values of the variables did. The output equation for solving directly for the displacement and velocity is shown below in *Equation 3*. These equations were solved in MATLAB as well in the program shown in Appendix 2.

$$\sum F = M\ddot{x} = F_e - c\dot{x} - kx$$

Equation 1: Equation of motion from free-body diagram of spring-mass-damper system

$$\begin{bmatrix} \dot{P}_1 \\ \dot{P}_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ -\frac{k}{M} & \frac{c}{M} \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{M} \end{bmatrix} F_e$$

Equation 2: State-space model of mass-spring-damper system

$$\begin{bmatrix} x \\ \dot{x} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} P_1 \\ P_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \end{bmatrix} F_e$$

Equation 3: State-space model output equation

Where:

$$x = Displacement$$
$$F_e = External\ force$$
$$M = Mass\ of\ block$$
$$c = Damping\ coefficient$$
$$k = Spring\ coefficient$$
$$\omega_n = Natrual\ frequency = \sqrt{\frac{k}{m}}$$
$$\zeta = Damping\ ratio = \frac{c}{2M\omega_n}$$

First the undamped spring and mass system were modeled with an initial displacement of 10 millimeters. The validated closed form solution to this system is shown below in Equation *4*. Damping ratio in this case is zero, and thus it is expected to have oscillation that doesn't change with time. The three solutions are shown below in Figure *2*, and while it is hard to tell the difference between the Simulink and closed-form solutions, they are all overlaid onto each other and match up as best as the temporal resolution of the data allows to inspect to. Using a method in which the zero-crosses of the function are counted the frequency of the function can be measured from the data. Using this the Simulink and state-space data are determined to be within 2% error of the calculated natural frequency.

Equation 4: Closed form solution of undamped spring and mass system

$$x(t) = x_0 \cos(\omega_n t) = 0.01 \cos(29.48\ t)$$

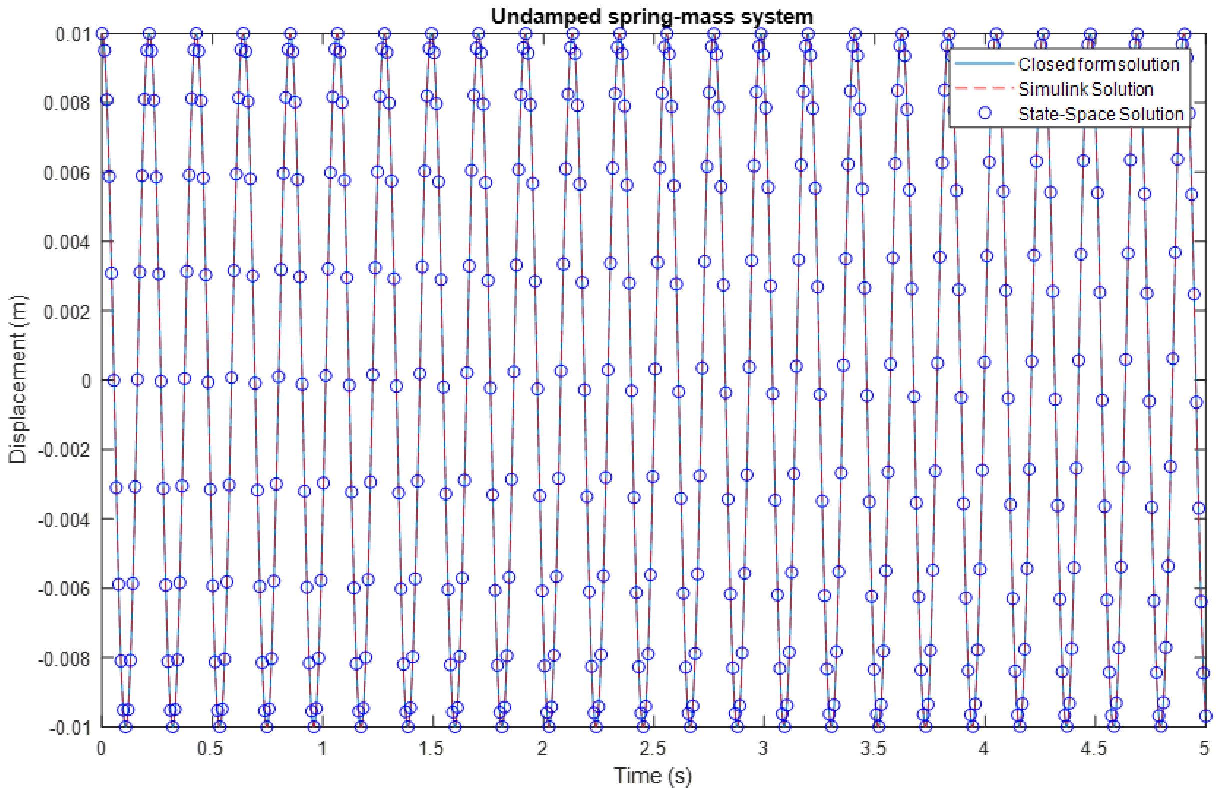


Figure 2: Undamped spring-mass system plot with the closed form in solid blue, the Simulink solution in dashed orange, and the State-space solution represented by circles.

The third part of the assignment was to analyze a mass-spring-damper system with 5% damping and an initial displacement of 10 millimeters. Using the equation described above as the definition of the damping ratio and setting the damping ratio to 0.05, the damping coefficient was found to be 67.8 N/m/s for 5% damping of this system. The Simulink, state-space, and closed-form data are plotted below in Figure *3*, and as they are so close in output to each other again it is hard to tell the difference between each of them. By inspection of the calculated time constant on the plot, the time constant of the system can be assessed to be at least similar to the data's time constant, and by using the same frequency measuring technique described above, the damped natural frequency of the Simulink and state-space models are within 2% of the calculated damped natural frequency.

Equation 5: Closed form solution of 5% damped system with initial displacement of 10 millimeters

$$x(t) = e^{-0.05\omega_n t}\, 0.01\, (\cos(\omega_n t) + \frac{0.05\ *\ 0.01}{\sqrt{1 - 0.05}} \sin(\omega_n t)$$

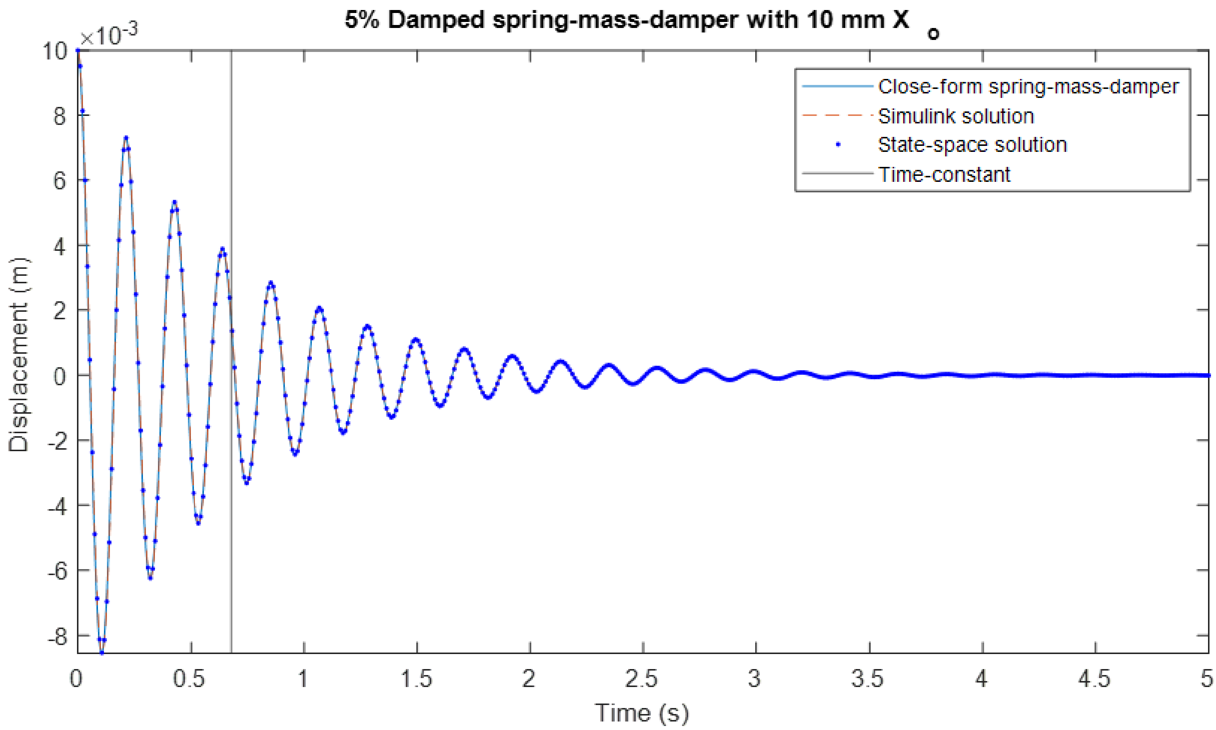


Figure 3: Closed-form solution, Simulink solution, and state-space solution of a 5% damped, 10 millimeter initial displacement mass-spring-damper system plotted on same graph, with the time constant demarcated with a vertical line.

The fourth system analyzed for this report was a step of 10 Newtons to the 5% damped mass-spring-damper system. As before the three functions (see Figure 4) match up exactly as far as it can be ascertained. The new steady state for the system with a 10 Newton step is approximately 0.5 millimeters, and both can be seen to settle to this value. If the mean of the data is taken to find the actual steady-state value being settled toward, the value is calculated to be 5 millimeters within two-tenths of a percent error. As previously, the zero-count technique shows that the frequency is also within a few percent of the expected value. The close-form solution is too long to write here, but it is enclosed in the source code of the MATLAB program in Appendix 2.

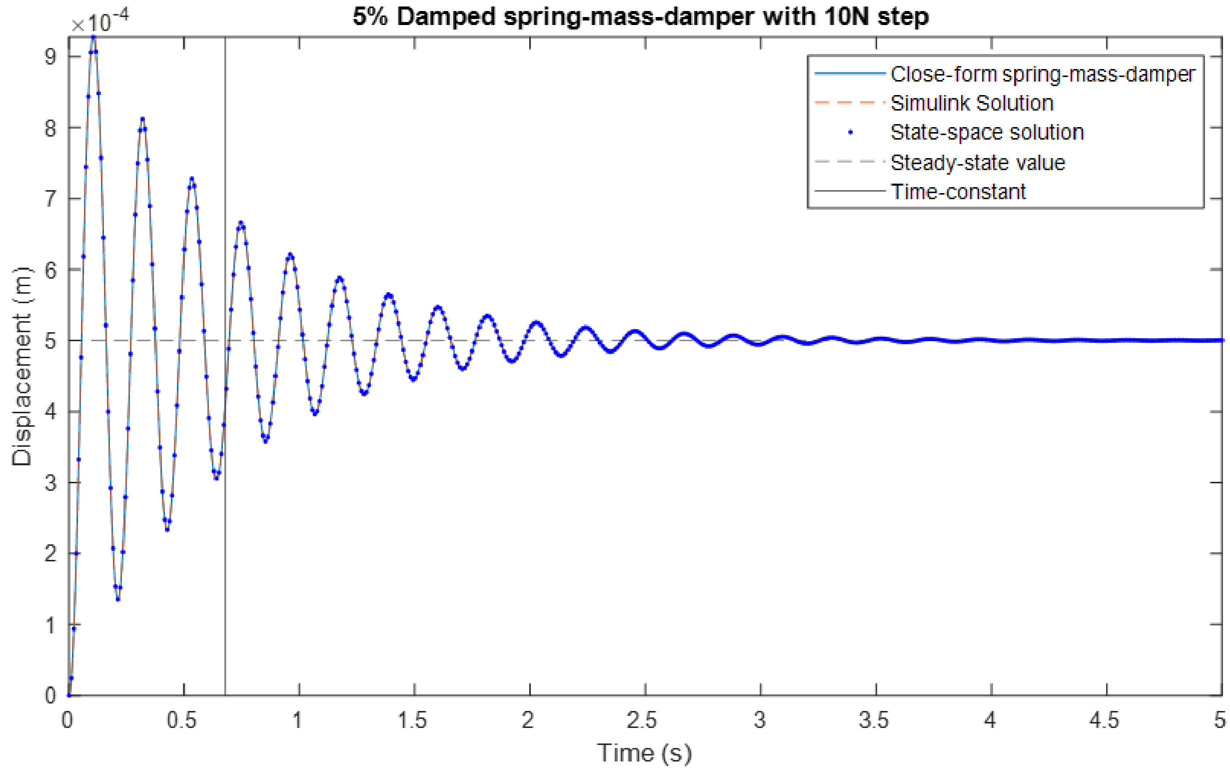


Figure 4: Closed-form, Simulink, and state-space solutions plotted with the vertical line representing the calculated time constant and the horizontal dotted line representing the calculated steady-states value.

The final systems analyzed were three systems that had a sinusoidal force input with an amplitude of 5 Newtons, but with varying frequencies of the force applied to observe the beats between the force and the natural frequency. As with the previous systems analyzed, all the systems matched up between the closed-form, Simulink, and state-space solutions. In this case as the frequency also changes slightly with time, the final 5 zero-crosses are used in this case instead of a mean over all the zero-crosses for confirming the frequency. The frequencies calculated from the data are tabulated below.

External force frequency - percentage of natural frequency (frequency)	**Output Frequency (rad/s)**
100% (Natural frequency for reference)	29.48
50% (14.74)	14.75
90% (26.53)	26.40
130% (38.32)	38.31

Below the plot of all three systems are shown, with each system being plotted with the closed form, Simulink, and state-space solution data plotted all on the same plot. As discovered in a previous report, the steady-state frequency of the three systems decay to the input frequency after the startup transients of the systems dissipate, which is also found to be the case with the Simulink and state-space models.

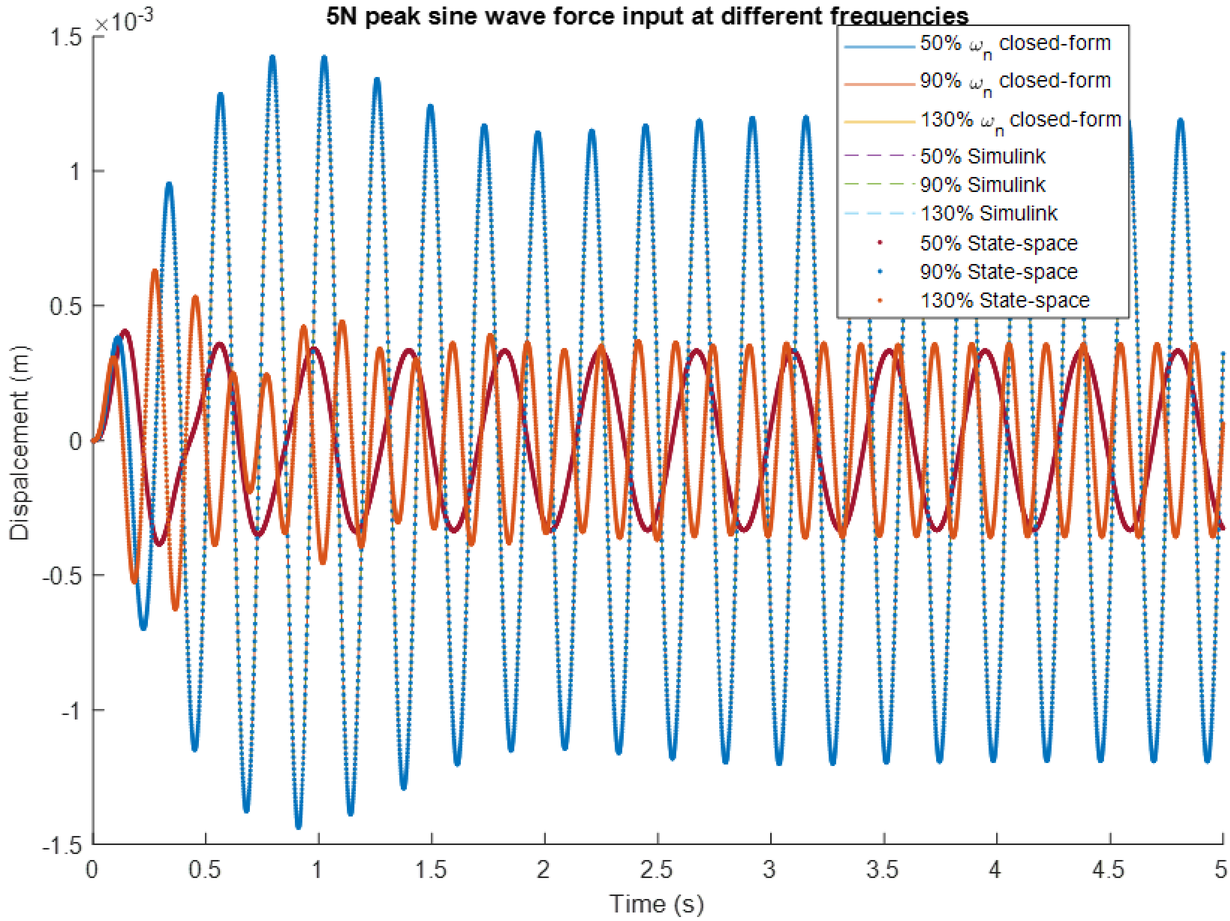
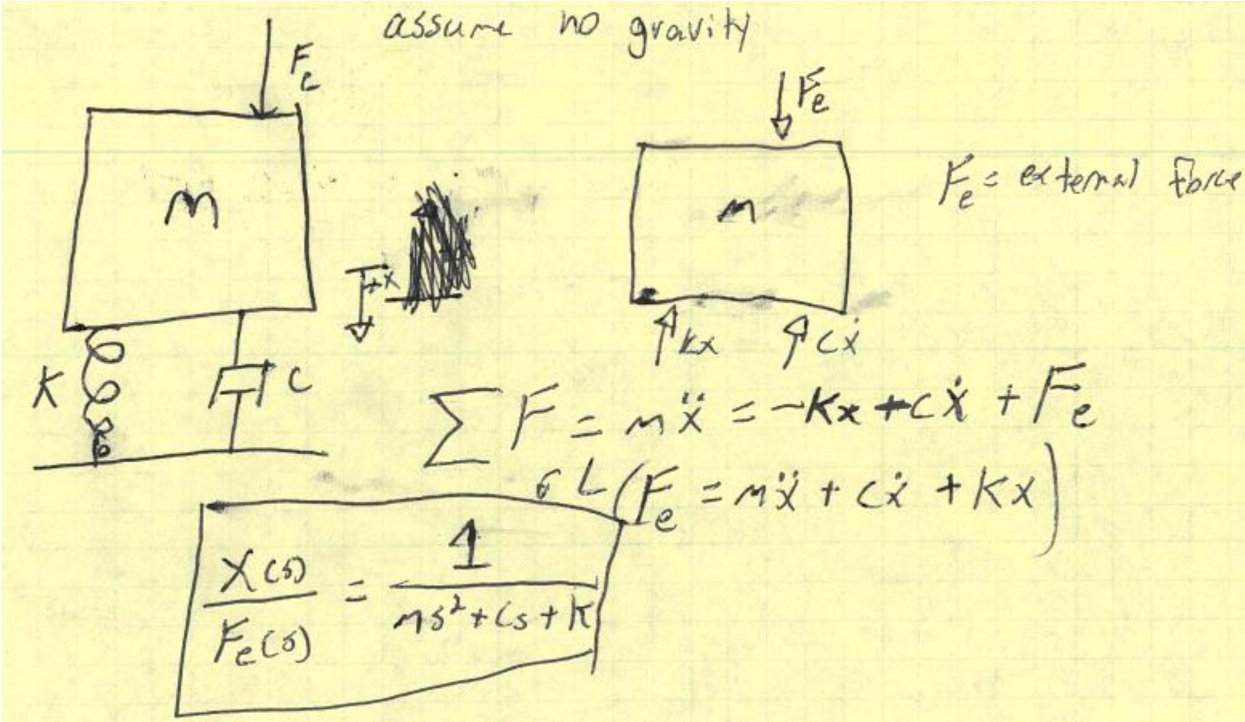


Figure 5: Three mass-spring-damper systems plotted with the closed-form, Simulink, and state-space data overlaid on each other due to their near-perfect matches to each other.

Appendix 1 – Free-body diagram and transfer function



Appendix 2 – MATLAB Code

```
clear, clc, close all
% Ross Smyth
% 2/26/2019
% Dynamic System Project 2

k     = 19990; % N/m
m     = 23; % kg
nat_f = sqrt(k/m); % rad/s
syms t s
zero_cross = @(v) find(v(:).*circshift(v(:), [-1 0]) <= 0);
```

2. No damping

```
x0       = 0.01; % Simulink initial condition
c        = 0; % for simulink
run_case = 1; % simulink external force case

sim2 = sim('springmassdamper.slx');

ss2               = ss([0, 1; -k/m, -c/m], [0; 1/m], [1, 0; 0, 1], [0; 0]);
[ss2_out, ss2_t] = initial(ss2, [x0; 0], 5);
ss2_dis           = ss2_out(:, 1);
```

```
fun2 = 0.01 * cos(nat_f * t);

figure
fplot(fun2, [0, 5])
title("Undamped spring-mass system")
xlabel("Time (s)")
ylabel("Displacement (m)")
hold on
plot(sim2.tout, sim2.displacement, 'r--', ss2_t, ss2_dis, 'bo')
legend('Closed form solution', 'Simulink Solution', 'State-Space Solution')
hold off

mf2 = 2 * pi /(2 * mean(diff( sim2.tout(zero_cross(sim2.displacement)))));
```

3. 5% Damping

```
c     = 2 * m * nat_f * 0.05;
nat_d = sqrt(1 - 0.05^2) * nat_f;
time3  = 1 / (nat_f * 0.05);

sim3 = sim('springmassdamper.slx');

ss3               = ss([0, 1; -k/m, -c/m], [0; 1/m], [1, 0; 0, 1], [0; 0]);
[ss4_out, ss3_t] = initial(ss3, [x0; 0], 5);
ss3_dis          = ss4_out(:, 1);

fun3 = exp(-0.05 * nat_f * t)* 0.01 * (cos(nat_d * t) + 0.05 * 0.01 / sqrt(1-0.05) * sin(nat_f *
t));

figure
fplot(fun3, [0, 5])

hold on
plot(sim3.tout, sim3.displacement, '--', ss3_t, ss3_dis, 'b.')
xline(time3)
hold off
title("5% Damped spring-mass-damper with 10 mm X_o")
xlabel("Time (s)")
ylabel("Displacement (m)")
legend('Close-form spring-mass-damper', 'Simulink solution', 'State-space solution', 'Time-
constant')

mf3 = 2 * pi /(2 * mean(diff( sim3.tout(zero_cross(sim3.displacement)))));
```

4 10N Step

```
run_case = 2;
x0       = 0;

fun4  = 10 / (m * nat_f^2) * (1 - 1 / sqrt(1 - 0.05^2) * exp(-0.05 * nat_f * t) * sin((nat_f *
```

```
sqrt(1 - 0.05^2) * t + atan(sqrt(1 - 0.05^2) / 0.05))));
time4 = 1/(0.05 * nat_f);

ss4_step         = stepDataOptions('StepAmplitude', 10);
[ss4_out, ss4_t] = step(ss3, 5, ss4_step);
ss4_dis          = ss4_out(:, 1);

sim4 = sim('springmassdamper.slx');

figure
fplot(fun4, [0, 5])

hold on
plot(sim4.tout, sim4.displacement, '--', ss4_t, ss4_dis, 'b.')
yline(10/k,'--');
xline(time4);
hold off
title("5% Damped spring-mass-damper with 10N step")
xlabel("Time (s)")
ylabel("Displacement (m)")
legend('Close-form spring-mass-damper', 'Simulink Solution', 'State-space solution', 'Steady-
state value', 'Time-constant')

mf4      = 2 * pi /(2 * mean(diff( sim4.tout(zero_cross(sim4.displacement)))));
err4_ss  = (mean(ss4_dis) - 5e-4) / 5e-4 * 100;
err4_sim = (mean(sim4.displacement) - 5e-4) / 5e-4 * 100;
```

5 5N sine wave, 50% natrual frequency

```
run_case = 3;

f5 = nat_f * 0.5;

sim5 = sim('springmassdamper.slx');

ss5_t   = 0:0.001:5;
ss5_out = lsim(ss3, 5 * sin(f5 * ss5_t), ss5_t);
ss5_dis          = ss5_out(:, 1);

fun5 = (-5/225887).*(cos((9995/46).^(1/2).*t)+(-
15).*sin((9995/46).^(1/2).*t))+(5/225887).*399.^(-1/2).*exp(1).^((-
1/2).*(1999/230).^(1/2).*t).*(399.^(1/2).*cos((1/2).*(797601/230).^(1/2).*t)+(-
149).*sin((1/2).*(797601/230).^(1/2).*t));

mf5 = diff(sim5.tout(zero_cross(sim5.displacement)));
mf5 = 2 * pi /(2 * mean(mf5(end-5)));
```

6 5N sine wave, 90% natrual frequency

```
run_case = 4;
```

```
f6 = nat_f * 0.9;

sim6 = sim('springmassdamper.slx');

ss5_t   = 0:0.001:5;
ss6_out = lsim(ss3, 5 * sin(f6 * ss5_t), ss5_t);
ss6_dis          = ss6_out(:, 1);


fun6 = (-25/441779).*(9.*cos(9.*(1999/230).^(1/2).*t)+(-
19).*sin(9.*(1999/230).^(1/2).*t))+(75/441779).*(3/133).^(1/2).*exp(1).^((-
1/2).*(1999/230).^(1/2).*t).*(399.^(1/2).*cos((1/2).*(797601/230).^(1/2).*t)+(-
37).*sin((1/2).*(797601/230).^(1/2).*t));

mf6 = diff(sim6.tout(zero_cross(sim6.displacement)));
mf6 = 2 * pi /(2 * mean(mf6(end-5)));
```

7 5N sine wave, 130% natural frequency

```
run_case = 5;

f7 = 1.3 * nat_f;

sim7 = sim('springmassdamper.slx');

ss5_t   = 0:0.001:5;
ss7_out = lsim(ss3, 5 * sin(f7 * ss5_t), ss5_t);
ss7_dis = ss7_out(:, 1);

fun7 = (-
5/985507).*(13.*cos(13.*(1999/230).^(1/2).*t)+69.*sin(13.*(1999/230).^(1/2).*t))+(65/985507).*399
.^(-1/2).*exp(1).^((-
1/2).*(1999/230).^(1/2).*t).*(399.^(1/2).*cos((1/2).*(797601/230).^(1/2).*t)+139.*sin((1/2).*(797
601/230).^(1/2).*t));

mf7 = diff(sim7.tout(zero_cross(sim7.displacement)));
mf7 = 2 * pi /(2 * mean(mf7(end-5)));

figure
hold on
fplot(fun5, [0, 5])
fplot(fun6, [0, 5])
fplot(fun7, [0, 5])

plot(sim5.tout, sim5.displacement, '--', sim6.tout, sim6.displacement, '--', sim7.tout,
sim7.displacement, '--')
plot(ss5_t, ss5_dis, '.', ss5_t, ss6_dis, '.', ss5_t, ss7_dis, '.')

legend('50% \omega_n closed-form', '90% \omega_n closed-form', '130% \omega_n closed-form', '50%
Simulink', '90% Simulink', '130% Simulink', '50% State-space', '90% State-space', '130% State-
space')
hold off
```

```
xlabel('Time (s)')
ylabel('Dispalcement (m)')
title('5N peak sine wave force input at different frequencies')
```