1 Introduction
For this lab, it used LOS model V-203 electrodynamic shaker as model. The purpose of this model is to allow it to build and simulate as accurately as possible regarding its electrical and mechanical characteristics of the machine and the activity and interaction when it tests. The outcome of this lab is to understand more with new system equations in a bond graph, translate the ordinary differential equation into different equations, and integrate it within Matlab's ode45. In addition, it creates a plot to look into the simulation and understand the dynamic of the shaker

2 Problem and Calculation
The schematic of the electrodynamic shake is shown below in figure 1. The left side of the machine is the electrodynamic part and the right side is the mechanically dynamic.

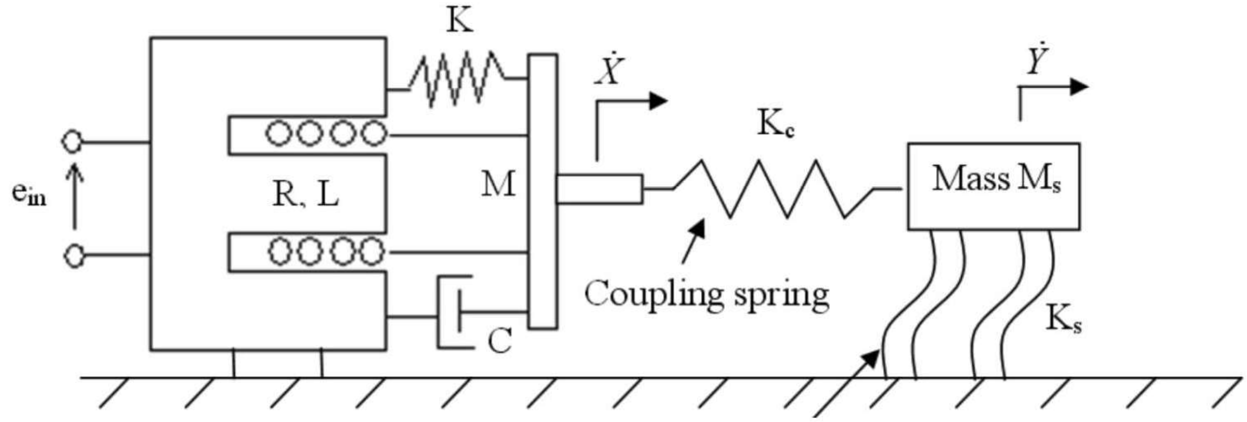


Figure 1: Schematic of LOS Model V-203

The bond graph of the shake is given in the lab where it is shown in figure 2.

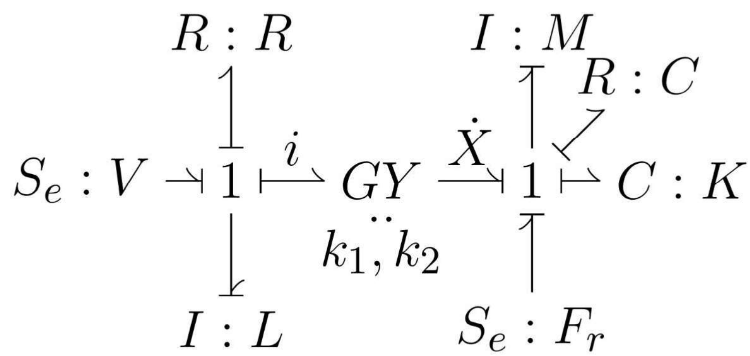


Figure 2: Bond Graph of All Important Functional Element

2.1 The Pertinent Equations Equation

$$M\frac{dX^2}{dt^2} + C\frac{dX}{dt} + KX = k_2 i + Fr$$

$$Ri + L\frac{di}{dt} = e_{in} - k_1\frac{dX}{dt}$$

$k_1, k_2$are constants that represent the $e = k_1\frac{dX}{dt}$and $k_2 i = F$

2.2 Parameter

- Mass of The Test Pieces
  - $M_s = 0.7$ kg
- Shaker Table Mass
  - $M = 178.04 \times 10^{-6}\ \frac{lb\ sec^2}{in}$
- Shaker Internal Stiffness
  - $K = 16.54\ \frac{lb}{in}$
- Shaker Internal Damping

- $C = 39.09 \times 10^{-3} \frac{lb\ sec}{in}$

- Shaker Coil Resistance
  - $R = 1.6\ \Omega$
- Shaker Coil Inductance
  - $L = 764\ \mu H$
- Shaker Voltage-Velocity Proportionality Coefficient
  - $K_1 = 95.10 \times 10^{-3} \frac{Volt}{IPS}$
- Maximum and Minimum Shaker Stroke
  - $X_{lim} = \pm 100$ mm
- Maximum and Minimum Shaker Acceleration
  - $\ddot{X} = \pm 100$mm
- Maximum and Minimum Shaker force
  - $F_{lim} = \pm 4.41$ lb

2.3 System Equation

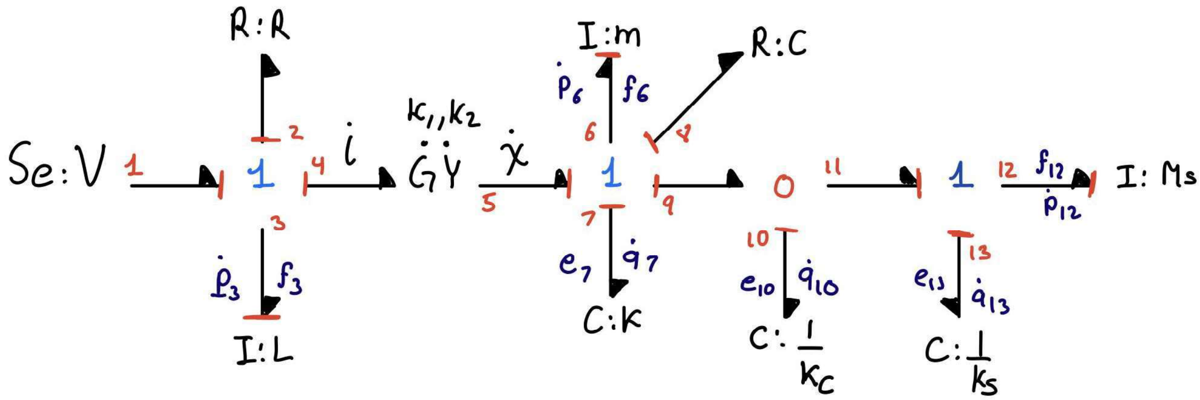


Figure 3: Bond Graph of All Important Functional Element an Test Element

2.4 State Derivative Equation

Input Variable: $S_e = e_{in}$

$$\dot{P}_L = e_{in} - R\frac{P_L}{L} - k_1\frac{P_M}{M}$$
$$\dot{P}_M = k_2\frac{P_L}{L} - Kq_X - C\frac{P_M}{M} - K_c q_c$$
$$\dot{q}_x = \frac{P_M}{M}$$
$$\dot{q}_C = \frac{P_M}{M} - \frac{P_s}{M_s}$$
$$\dot{P}_s = K_c q_c - K_s q_s$$
$$\dot{q}_s = \frac{P_s}{M_s}$$

The derivation of these equations are from the appendix below.

2.5 Input Function

The maximum voltage (e_{in}) is a constant input for a system in the simulation and all initial conditions are zero. By taking the pertinent equations above, it produces new equation for initial condition which shown below

$$0 = k_2 i + F_r$$

$$Ri = e_{in}$$

The maximum force that acts on the mechanical side of the gyrator is 4.4 lbf. Therefore, it can be calculated e_{in} as following:

$$k_2 i - 4.4 = 0 \Rightarrow i = 4.4/k_2 = 4.4/0.8416$$

$$I = 5.228\ \mathrm{A}$$

$$e_{in} = 1.6\ (5.228) =$$

$$\mathbf{e_{in} = 8.365\ V}$$

2.6 Time Parameters

To calculate the time duration, it firstly needs to find the natural frequency for the test pieces where it set to be $f_n = 20$ Hz, and the natural frequency for the load table based on the mass (M) and the internal spring of the shaker (K). Since all frequency are in hertz, it also converts into radian per second. The conversion calculation is shown below.

- Natural Frequency of the Test Piece: $f_n = 20,\ w_n = f_n * 2\pi = 125.67$
- Natural Frequency of the Coupling Spring and the Test Object
  - $w_{n1} = \sqrt{\frac{Kc}{Ms}} = 0$ rad/s , $f_{n1} = \frac{w_n}{2\pi} = 0\ Hz$
- Natural Frequency of Shaker
  - $w_{n2} = \sqrt{\frac{K}{M}} = 304.6961$ rad/s , $f_{n2} = \frac{w_{n2}}{2\pi} = 48.4939\ Hz$
- Natural Frequency the Coupling Spring and the Load table of mass
  - $w_{n3} = \sqrt{\frac{Kc}{M}} = 0$ rad/s , $f_{n3} = \frac{w_{n3}}{2\pi} = 0\ Hz$

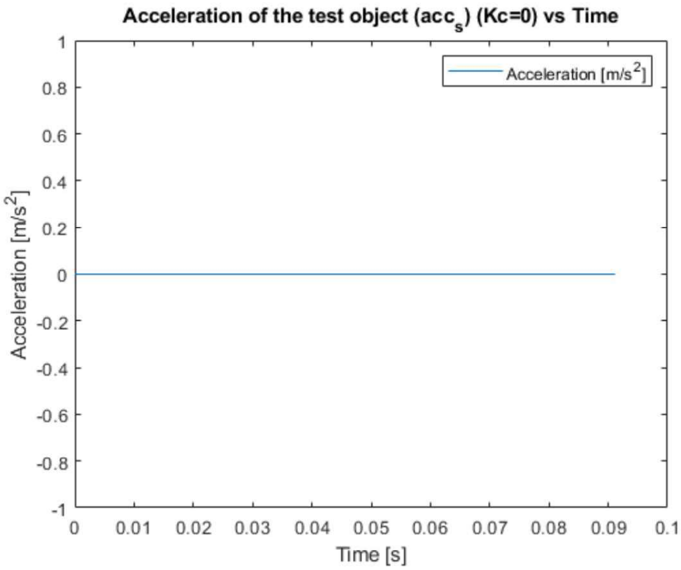
Next, it is to find the period of 1 oscillation in second where it chose the minimum time where the equation as follow:

Minimum Time (T_min) = 1 over the Maximum of all natural frequency

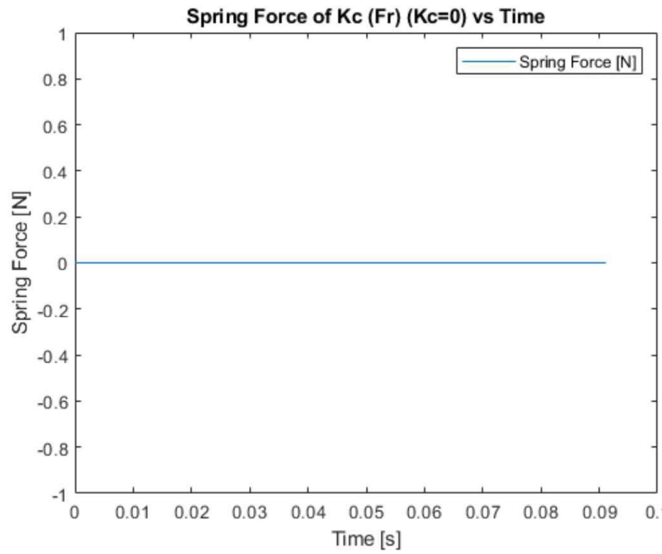
T_min = 0.0206 s

Then, it finds the duration of simulation of using period, and the shortest period of oscillation or time constant of the system. For this simulation, it picked to be at least 100 step per oscillation. After that it define the number of time constants and number of points in total simulation.

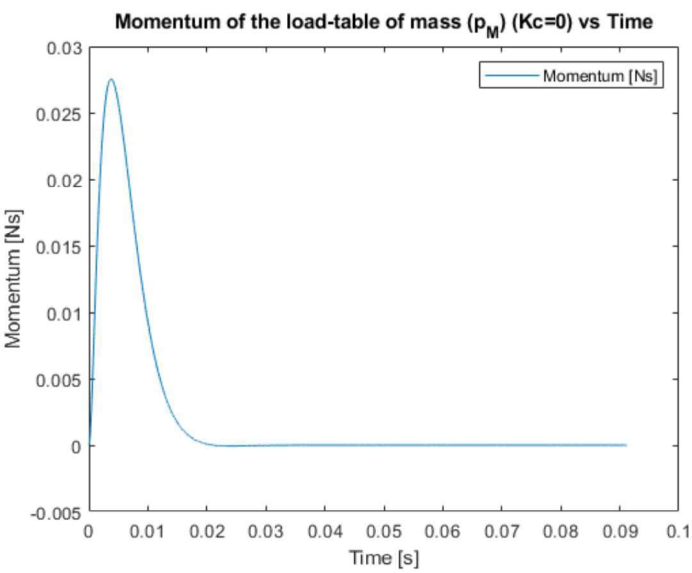
3 Result From Simulation



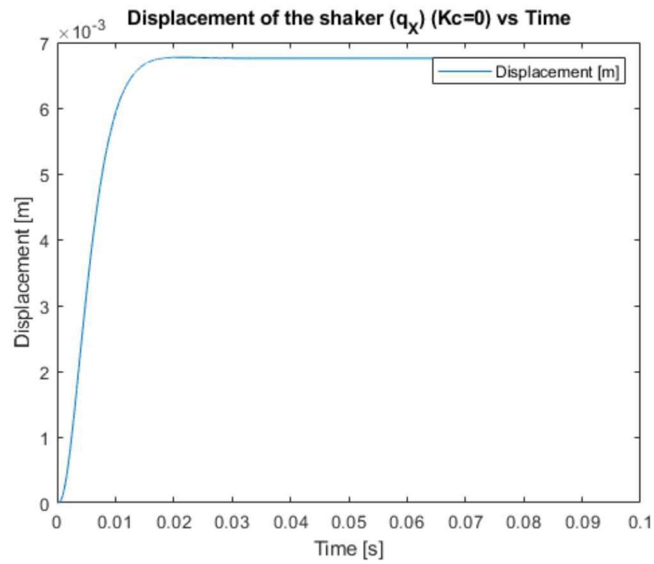
4a/ Acceleration of test object vs time graph



4b/ Spring force vs time graph

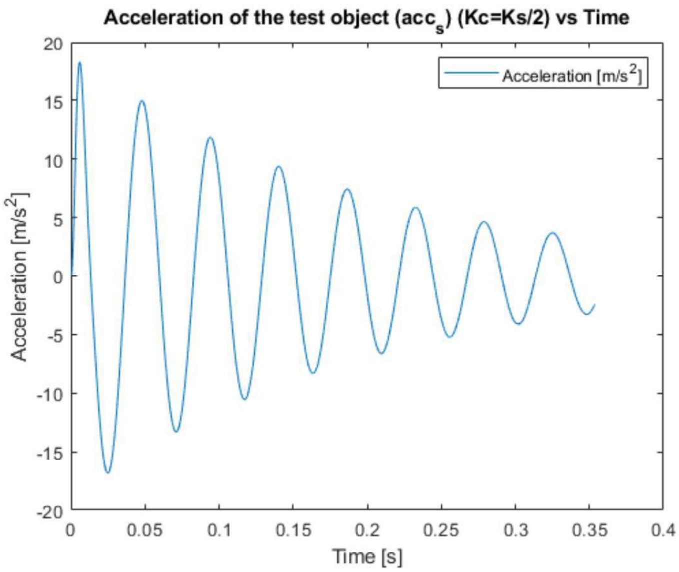


4c/ Momentum of load table mass vs time graph

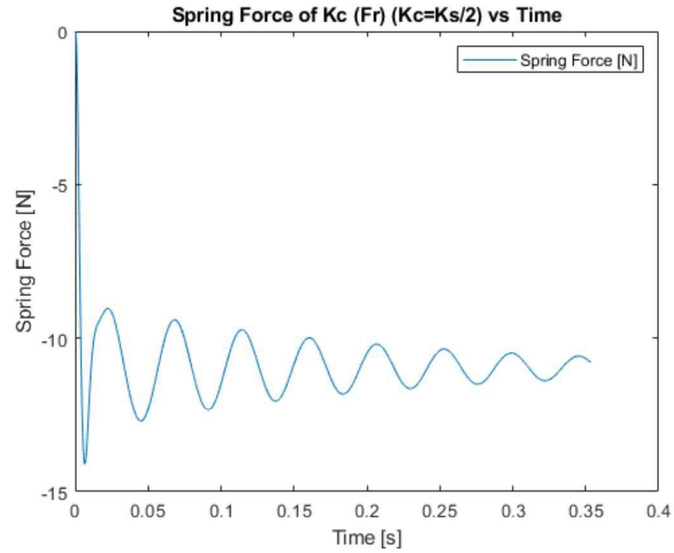


4d/ Displacement of shaker vs time graph

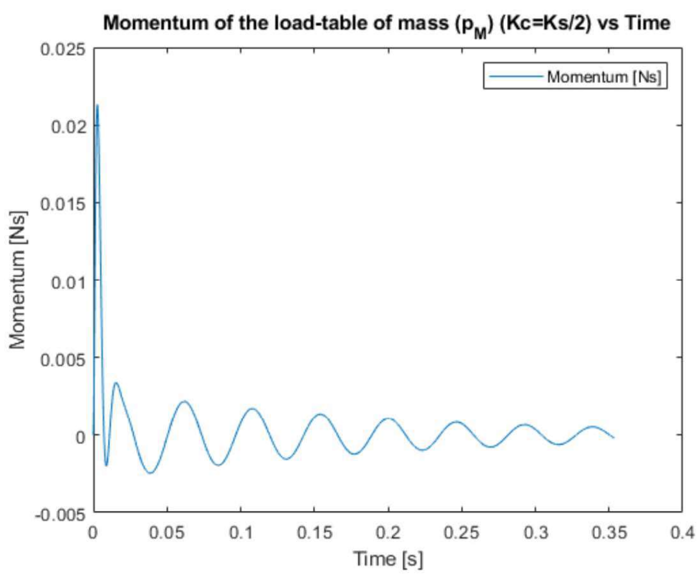
Fig 4 a-d: Electrodynamic shaker system when $K_C = 0$



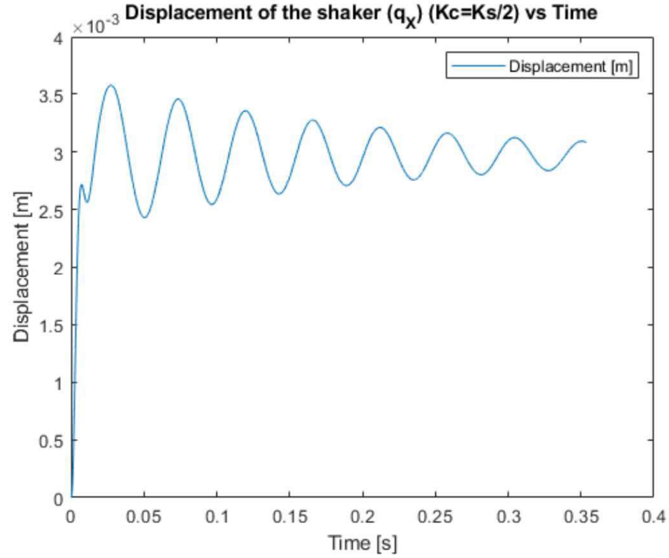
5a/ Acceleration of test object vs time graph



5b/ Spring force vs time graph

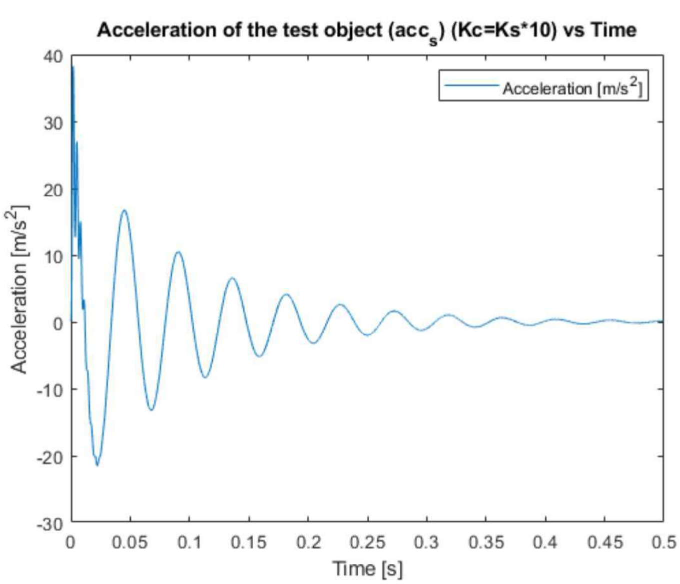


5c/ Momentum of load-table mass vs time graph

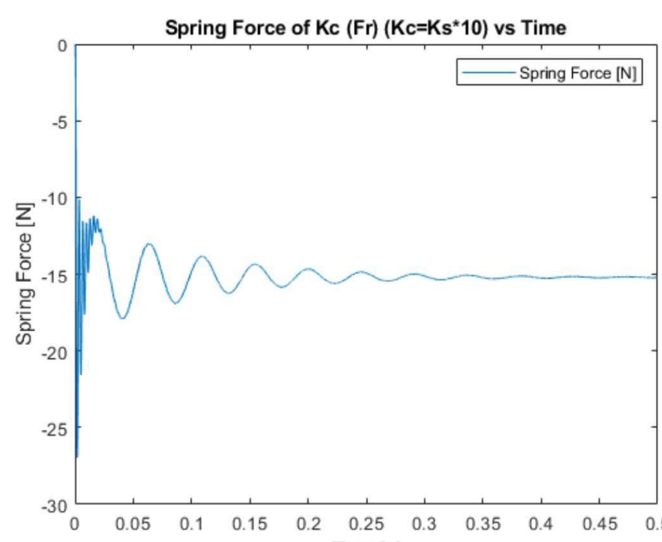


5d/ Displacement of shaker vs time graph

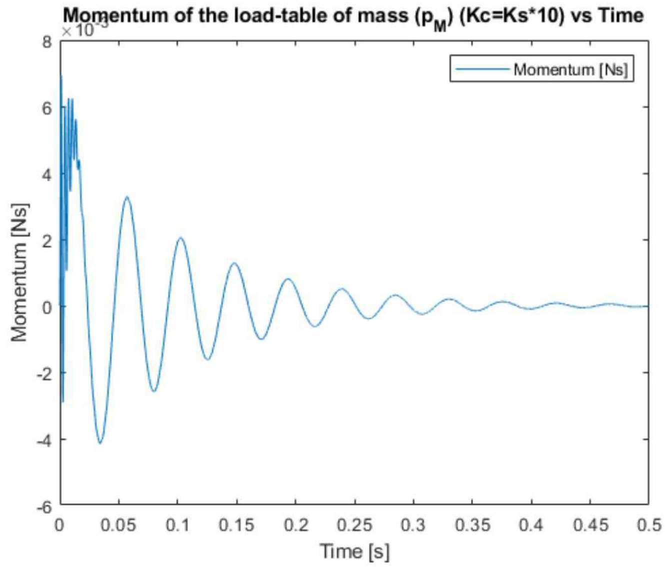
Fig 5 a-d: Electrodynamic shaker system when $K_C = 0.5K_S$



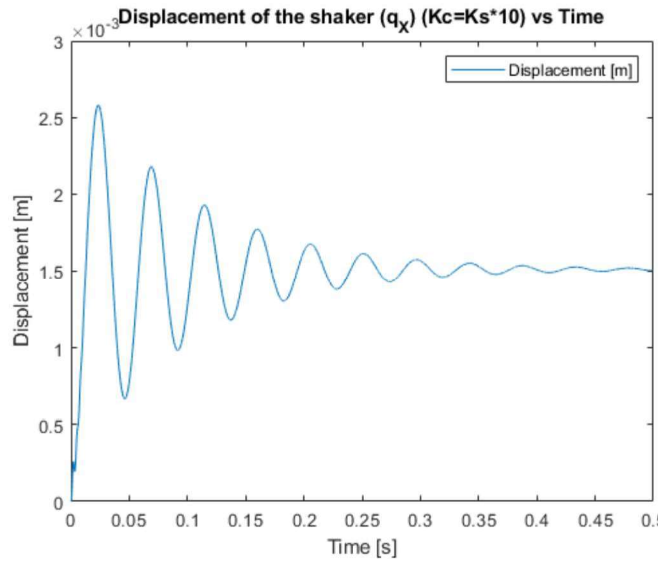
6a/ Acceleration of test object vs time graph



6b/ Spring force vs time graph



6c/ Momentum of load-table mass vs time graph



6d/ Displacement of shaker vs time graph

Fig 6 a-d: Electrodynamic shaker system when $K_C = 10K_S$

Analysis

The 4 variables that we are interested in are:

+ Acceleration of test object (acc_S)
+ Spring force (F_R)
+ Momentum of load table mass (P_M)
+ Displacement of the shaker (q_x)

These 4 variables are tested with coupling spring of different spring constant K_C

+ Condition 1: zero coupling spring constant $K_C = 0$
+ Condition 2: coupling spring constant is half of support spring constant $K_C = 0.5K_S$
+ Condition 3: coupling spring constant is ten times of the support spring constant $K_C = 10K_S$

The first variable we are interested in is the acceleration of the test object. We decide to observe the test object's acceleration versus time since we want to know how the test object accelerates at different Kc values. The graph would tell us whether increasing and decreasing the coupling spring constant will enhance the performance of the system. Looking at figure 4a, 5a, 6a, it can be concluded that when Kc increase, the test object's acceleration will oscillate at higher amplitude and higher frequency. For instance, when Kc = 0 the test object's acceleration is zero, but when Kc = 10Ks, the test object acceleration peak amplitude is 38 m/s^2. When Kc = 0, the test object's acceleration will oscillate sinusoidally and eventually reach zero steady state. We learn that adding the coupling spring will disturb the steadiness of the test object's acceleration.

The second variable we are interested in is the spring force of the coupling spring. We want to graph the coupling force overtime to learn how much force was exerted on the coupling spring when the coupling spring constant is increased from 0 to 10 K_S. Looking at figure 4b, 5b, 6b, when K_C increases, the spring force will oscillate at a higher frequency. The sudden increase of frequency happened during the first 0.02 s when comparing the model of higher K_C with the model with smaller K_C.

The third variable we are interested in is the momentum of the load table mass. We want to model the momentum of load table mass over time to learn how much mass is moving and how fast the mass is moving. Looking at figure 4c, 5c, 6c, when K_C increases, the load table mass will oscillate at lower amplitude and higher frequency. For instance, when $K_C = 0$ the peak amplitude of momentum is 0.027 Ns, but when $K_C = 10\ K_S$, the peak amplitude of momentum is 0.006 Ns. From this behavior, we learn that decreasing coupling spring constant K_C will increase the peak amplitude of momentum of the load table mass.

The fourth variable we are interested in is the displacement of the shaker. We decide to observe the shaker displacement over time to learn how the coupling spring constant affects the displacement of the shaker. Looking at figure 4d, 5d, 6d, when K_C increases, the shaker displacement will oscillate at lower amplitude and higher frequency. For instance, when $K_C = 0$ the peak displacement is 0.007 m but when $K_C = 10\ K_S$, the peak shaker displacement is 0.0026 m. From this, we learn that the coupling spring constant relate inversely proportional to the shaker displacement.

Overall, when $K_C = 0$, the object's acceleration and the spring constant are both zero. The momentum of the load table mass only vibrates one oscillation then reaches zero steady state. The shaker displacement also oscillates one time and stays at its steady-state value as time moves on. In the second and third condition when $K_C > 0$. We observe that the graphs of 4 variables oscillate sinusoidally and eventually reach zero steady state. The only difference between the second and third condition is that the curve with higher K_C value oscillates at higher frequency than the curve of lower K_C value. The variance of the coupling spring constant affects the performance of the electrodynamic shaker.

4/ Contributions

Team members were Vinh Nguyen and Borum Long. Both members worked together to ensure the lab report was completed effectively and on time. The report was done by both members. Borum Long developed the simulations code including the input, output, right hand side, and main function. Vinh Nguyen debugged the code and wrote the analysis part of the report including explanation of the results and the plots from Matlab. Both members also worked on the remaining write up parts of the lab including the introduction, bond graph, state variable, system parameters and time parameters. Overall the workload was evenly distributed for both persons.

5/ Code

eval_rhs_with_input_fun.m – Evaluate the right hand side

```
function xdot = eval_rhs_with_input_fun(t, x, w, c)

% Eval_rhs_with_input_fun - Return the state derivatives of the states.
% Syntax: xdot = eval_rhs_with_input_fun(t, x, w, c)
```

```
% Inputs:
% t - Scalar value of time, size 1x1.
% x - State vector at time t, size mx1 where m is the number of states.
% c - Constant parameter structure
% w -  Anonymous function, w(t, x, c), that returns the input vector at time t, size 1x1
% Outputs: xdot - state derivatives.

% unpack all of the state derivative vector
q_X = x(1); % displacement of the shaker
p_M = x(2); % momentum of the load-table of mass
p_L = x(3); % momentum of the shaker
q_C = x(4); % displacement of the load-table of mass
q_s = x(5); % displacement of the test object
p_s = x(6); % momentum of the test object

%evaluate the input function
r = w(t,x,c)

% unpack input
e_in = r(1)

%unpack the inputs into useful variable names
Ms = c.Ms; % mass of the test piece in kg
M = c.M; % shaker table mass in kg
K = c.K; % shaker internal stiffness in N/m
C = c.C; % shaker internal damping in Ns/m
R = c.R; % shaker coil resistance in ohms
L = c.L; % shaker coil inductance in H
k1 = c.k1; % shaker voltage-velocity proportionality coefficient in Vs/m
k2 = c.k2; % shaker force-current proportionality coefficient in N/A
X_lim = c.X_lim; % maximum and minimum shaker stroke in m
a_lim = c.a_lim; % maximum shaker acceleration in m/s^2
F_lim = c.F_lim; % maximum shaker force in N
Kc = c.Kc; % coupling spring internal stiffness in N/m
Ks = c.Ks; % test object internal stiffness in N/m

% calculate the state derivatives
q_Xdot= (1/M)*p_M;
p_Mdot= -Kc*q_C+ (k2/L)*p_L-K*q_X-C*(p_M/M);
p_Ldot= e_in -(k1/M)*p_M - (R/L)*p_L;
q_Cdot= (p_M/M)-(p_s/Ms);
q_sdot= p_s/Ms;
p_sdot= Kc*q_C - Ks*q_s;

% pack the state derivatives into an mx1 vector
xdot = [q_Xdot;p_Mdot;p_Ldot;q_Cdot;q_sdot;p_sdot];

end
```

eval_shaker_input.m – Evaluate the input for the shaker

```
function r = eval_shaker_input(t, x, c)

%eval_shaker_input - Returns the input vector at any given time.
%Syntax: r = eval_shaker_input(t, x, par)
%Inputs:
```

```
%t - A scalar value of time, size 1x1.
%x - State vector at time t, size mx1 where m is the number of states.
%c - Constant parameter vector.
%Outputs: r - Input vector at time t, size ox1 where o is the number of inputs.

% unpack the parameters into useful variable names
Ms = c.Ms; % mass of the test piece in kg
M = c.M; % shaker table mass in kg
K = c.K; % shaker internal stiffness in N/m
C = c.C; % shaker internal damping in Ns/m
R = c.R; % shaker coil resistance in ohms
L = c.L; % shaker coil inductance in H
k1 = c.k1; % shaker voltage-velocity proportionality coefficient in Vs/m
k2 = c.k2; % shaker force-current proportionality coefficient in N/A
X_lim = c.X_lim; % maximum and minimum shaker stroke in m
a_lim = c.a_lim; % maximum shaker acceleration in m/s^2
F_lim = c.F_lim; % maximum shaker force in N
Kc = c.Kc; % coupling spring internal stiffness in N/m
Ks = c.Ks; % test object internal stiffness in N/m
e_in = 8.365; % e_in = R*F_lim/k2
r = [e_in];

end
```

eval_output_with_state_derivatives_accelerations.m – Evaluate the output derivative

```
function z = eval_output_with_state_derivatives_accelerations(t, xdot, x, w, c)

%eval_output_with_state_derivatives_accelerations - Returns the output vector at the specified time.
%Syntax: z = eval_output_with_state_derivatives_accelerations(t, xdot, x, r, c)
%Inputs:
%t - Scalar value of time, size 1x1.
%xdot - State derivative vector as time t, size mx1 where m is the number of states
%x - State vector at time t, size mx1 where m is the number of states.
%w -  Anonymous function, w(t, x, c), that returns the input vector at time t, size 1x1
%c - Constant parameter structure
%Outputs:z - Output vector at time t, size qx1.

%unpack all of the state derivative vector
q_Xdot = xdot(1);
p_Mdot = xdot(2);
p_Ldot = xdot(3);
q_Cdot = xdot(4);
q_sdot = xdot(5);
p_sdot = xdot(6);

% unpack all of the state vectors
q_X= x(1); % displacement of the shaker
p_M= x(2); % momentum of the load-table of mass p_L= x(3); â€‹% momentum of the shaker
q_C= x(4); % displacement of the load-table of mass q_s= x(5); â€‹% displacement of the test object
p_s= x(6); % momentum of the test object

% evaluate the input function
r = w(t, x, c);
% unpack the inputs into useful variable names
e_in = r(1);

% unpack all of the parameter vectors
Ms = c.Ms; % mass of the test piece in kg
M = c.M; % shaker table mass in kg
K = c.K; % shaker internal stiffness in N/m
```

```
C = c.C; % shaker internal damping in Ns/m
R = c.R; % shaker coil resistance in ohms
L = c.L; % shaker coil inductance in H
k1 = c.k1; % shaker voltage-velocity proportionality coefficient in Vs/m k2 = par(8); â€‹% shaker force-current proportionality coefficient in N/A X_lim = par(9); â€‹% maximum and minimum shaker stroke in m
a_lim = c.a_lim; % maximum shaker acceleration in m/s^2
F_lim = c.F_lim; % maximum shaker force in N
Kc = c.Kc; % coupling spring internal stiffness in N/m
Ks = c.Ks; % test object internal stiffness in N/m

% calculate acceleartions of load table of mast and test object
acc_M = p_Mdot/M;
acc_s = p_sdot/Ms;

% pack the result into a 1x1 vector
z = [acc_M; acc_s];

end
```

eval_output_force.m – Evaluate the output

```
function Y = eval_output_force(t, x, w, c)

%eval_output_force - Returns the output vector at the specificed time
%Syntax: y = eval_output_force(t,x,r,c)
%t: scalar value of time, size 1x1
%x: state vector at time t, size Ox1 where m is the number of state
%c: constant parameter vector
%w -  Anonymous function, w(t, x, c), that returns the input vector at time t, size 1x1
%Output: Y: output vector at time t, size qx1 where q is the number of outputs.

%unpack the parameters into useful varible names
Ms = c.Ms; % mass of the test piece in kg
M = c.M; % shaker table mass in kg
K = c.K; % shaker internal stiffness in N/m
C = c.C; % shaker internal damping in Ns/m
R = c.R; % shaker coil resistance in ohms
L = c.L; % shaker coil inductance in H
k1 = c.k1; % shaker voltage-velocity proportionality coefficient in Vs/m
k2 = c.k2 % shaker force-current proportionality coefficent in N/A
X_lim = c.X_lim; % maximum and minimum shaker stroke in m
a_lim = c.a_lim; % maximum shaker acceleration in m/s^2
F_lim = c.F_lim; % maximum shaker force in N
Kc = c.Kc; % coupling spring internal stiffness in N/m
Ks = c.Ks; % test object internal stiffness in N/m
q_C= x(4);

% calculate the force of the load-table of mass (Fr)
Fr = - Kc * q_C;
Y = [Fr];

end
```

main.m – Main function

```
% main function

clc
clear
c.Ms = 0.7; % mass of the test piece in kg
```

```
c.M = 0.0312; % shaker table mass in kg
c.K = 2896.6; % shaker internal stiffness in N/m
c.C = 6.8457; % shaker internal damping in Ns/m
c.R = 1.6; % shaker coil resistance in ohms
c.L = 764E-6; % shaker coil inductance in H
c.k1 = 3.744; % shaker voltage-velocity proportionality coefficient in Vs/m
c.k2 = 3.7436;% shaker force-current proportionality coefficient in N/A
c.X_lim = 0.1;% maximum and minimum shaker stroke in m
c.a_lim = 1324.35; % maximum shaker acceleration in m/s^2
c.F_lim = 19.572; % maximum shaker force in N
c.Fn = 20; % natural frequency of the test object in Hz
c.Wn = 2*pi*c.Fn; % natural frequency of the test object in Hz
c.Ks = (c.Wn.^2)*c.Ms; % test object internal stiffness in N/m
c.Kc = 0; % coupling spring internal stiffness in N/m
x0 = [0; 0; 0; 0; 0; 0];

% Time steps: determining the time span.
c.Fn = 20; % natural frequency of the test object in Hz
c.Wn = c.Fn*2*pi;

c.Wn1 = sqrt(c.Kc/c.Ms); % natural frequency of the coupling spring and the test object in Hz
c.Fn1= c.Wn1 / (2* pi);

c.Wn2 = sqrt(c.K/c.M); % natural frequency of the shaker in Hz
c.Fn2= c.Wn2 / (2* pi);

c.Wn3 = sqrt(c.Kc/c.M); % natural frequency the coupling spring and the load-table of mass in Hz
c.Fn3= c.Wn3 / (2* pi);

T_min = 1/max([c.Fn,c.Fn1,c.Fn2, c.Fn3]); % period of 1 oscillation in s (choose the larger frequency)

%Defining the Duration of Simulation Using Periods
T1 = c.M/c.C;
T2 = c.L/c.R;
T_min1 = min([T1, T2])
Tperiod = min([T_min,T_min1]) % the shortest period of oscillation or the shortest time constant of the system

% at least 100 steps per oscillation;
dt = Tperiod/100;

%Define the number of time constants and number of points in total
%simulation
T_max1 = max([T1,T2]);

Tfinal = 20*T_max1% 20 oscillations
num_steps = round(Tfinal / dt) +1
ts = linspace(0, Tfinal, num_steps);

% time span
f_anon = @(t, x) eval_rhs_with_input_fun(t, x, @eval_shaker_input, c);
% Anonymous function
[ts, xs] = ode45(f_anon, ts, x0);
% integrate the dynamics equations
for i=1:length(ts)
% calculate xdot at the given time
    xdot = eval_rhs_with_input_fun(ts(i), xs(i, :), @eval_shaker_input, c);
end
% Calculate the force of the load-table of mass (Fr)
Ys = zeros(length(ts), 1); % create a matrix to store the values, nxq
for i=1:length(ts)
% the r input isn't used, so nan can be set as a placeholder
Ys(i, :) = eval_output_force(ts(i), xs(i, :), @eval_shaker_input, c);
```

```
end
% Calculate the acceleration of the load-table of mass (acc_M) and the acceleration of the test object (acc_s)
zs = zeros(length(ts), 2); â€‹% place to store outputs
for i=1:length(ts)
% calculate the outputs that depend on xdot and store them
    xdot = eval_rhs_with_input_fun(ts(i), xs(i, :), @eval_shaker_input, c);
    zs(i, :) = eval_output_with_state_derivatives_accelerations(ts(i), xdot, xs(i, :), @eval_shaker_input, c);
end

% plot each state
figure()
plot(ts, xs(:, 1))
title('Displacement of the shaker (q_X) vs Time')
legend('Displacement [m]')
ylabel('Displacement [m]')
xlabel('Time [s]')
figure()
plot(ts, xs(:, 4))
title('Displacement of the load-table of mass (q_C) vs Time')
legend('Displacement [m]')
ylabel('Displacement [m]')
xlabel('Time [s]')
figure()
plot(ts, xs(:, 5))
title('Displacement of the test object (q_s) vs Time')
legend('Displacement [m]')
ylabel('Displacement [m]')
xlabel('Time [s]')
figure()
plot(ts, xs(:, 3))
title('Momentum of the shaker (p_L) vs Time')
legend('Momentum [Ns]')
ylabel('Momentum [Ns]')
xlabel('Time [s]')
figure()
plot(ts, xs(:, 2))
title('Momentum of the load-table of mass (p_M) vs Time')
legend('Momentum [Ns]')
ylabel('Momentum [Ns]')
xlabel('Time [s]')
figure()
plot(ts, xs(:, 6))
title('Momentum of the test object (p_s) vs Time')
legend('Momentum [Ns]')
ylabel('Momentum [Ns]')
xlabel('Time [s]')
figure()
plot(ts, Ys(:,1))
title('Spring Force of Kc (Fr) vs Time')
legend('Spring Force [N]')
ylabel('Spring Force [N]')
xlabel('Time [s]')
figure()
plot(ts, zs(:,1))
title('Acceleration of the load-table of mass (acc_M) vs Time')
legend('Acceleration [m/s^2]')
ylabel('Acceleration [m/s^2]')
xlabel('Time [s]')
figure()
plot(ts, zs(:,2))
title('Acceleration of the test object (acc_s) vs Time')
legend('Acceleration [m/s^2]')
```