



University of Cincinnati
College of Engineering and Applied Science

Course Code : **MECH-3080**
Course Title : **System Dynamics & Vibrations**

Date Submitted : **August 5, 2016**

Group Members : **STANG, Matt**
: **BRAIDICH, Alex**
: **ELLMAKER, Spencer**

Instructor : **Dr. Allyn W. Phillips**

ABSTRACT

The System Dynamics and Vibrations final project centered on an electro-mechanical device that served the original purpose of providing an impact emulator for the AURA virtual reality immersive gaming vest. The electro-mechanical device's operation can be modelled as a two degree of freedom system through the interaction of a mass-spring-damper system and a resistor-inductor-capacitor circuit. The goals of this project was to develop the coupled electro-mechanical differential equations, simulate the response of the system for varying initial conditions, plotting the frequency response functions, and developing/plotting the transmissibility to the ground.

Table of Contents

ABSTRACT	2
Table of Contents.....	3
1. Developing the Coupled Differential Equations.....	5
1.1 Development of Electrical Differential Equation	5
1.1.1 Summary of Procedure.....	5
1.1.2 Development of Differential Equation	5
1.1.3 Analysis & Discussion	5
1.2 Development of Mechanical Differential Equation	5
1.2.1 Summary of Procedure.....	5
1.2.2 Development of Differential Equation	6
1.2.3 Analysis & Discussion	6
2. Simulating the Response of the System to Initial Conditions.....	7
2.1 Simulating the Response of the System to Initial Conditions	7
2.1.1 Summary of Procedure.....	7
2.1.2 Simulation of Initial Conditions (Part 1).....	7
2.1.3 Analysis & Discussion (Part 1).....	8
2.1.4 Simulation of Initial Conditions (Part 2).....	8
2.1.5 Analysis & Discussion (Part 2).....	8
2.1.6 Simulation of Initial Conditions (Part 3).....	9
2.1.7 Analysis & Discussion (Part 3).....	9
2.1.8 Simulation of Initial Conditions (Part 4).....	10
2.1.9 Analysis & Discussion (Part 4).....	11
2.1.10 Simulation of Initial Conditions (Part 5).....	12
2.1.11 Analysis & Discussion (Part 5).....	13
3. Developing the Frequency Response Function and Transmissibility.....	13
3.1 Developing and Plotting the Frequency Response Function	13
3.1.1 Summary of Procedure.....	13
3.1.2 Developing of Frequency Response Function.....	13
3.1.3 Plotting of Frequency Response Function.....	14
3.1.4 Analysis and Discussion.....	15
3.2 Developing and Plotting the Transmissibility.....	15

3.2.1 Summary of Procedure.....	15
3.2.2 Developing and Plotting of Transmissibility	15
3.2.3 Analysis and Discussion.....	16
4. “WOW” Factor.....	17
APPENDICES.....	19
A – MATLAB CODE	19

1. Developing the Coupled Differential Equations

1.1 Development of Electrical Differential Equation

1.1.1 Summary of Procedure

For the first portion of this project, the resistor-inductor-capacitor circuit was analyzed in order to develop the differential equation for its behavior. This process was completed using Kirchhoff's Voltage Laws as well as the time domain expressions for impedance.

1.1.2 Development of Differential Equation

The following figure and equations summarize the process that was utilized to obtain the resistor-inductor-capacitor circuit's differential equation for its behavior.

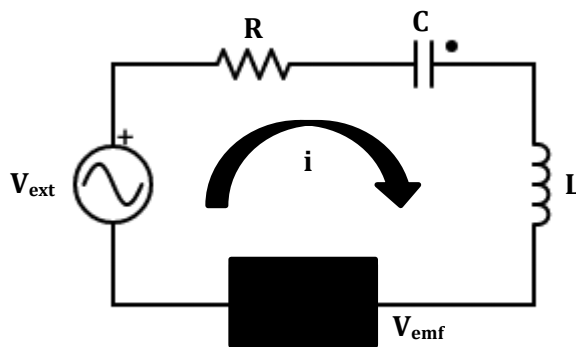


Figure 1. Modelled Circuit

$$V_{emf} = B \dot{x}$$

Circuit Equation (KVL)

$$V_{ext} - Ri - \frac{1}{C} \int idt - L \frac{di}{dt} - B\dot{x} = 0$$

Final Differential Equation

$$L\ddot{q} + B\dot{x} + R\dot{q} + \frac{1}{C}q = V_{ext}$$

1.1.3 Analysis & Discussion

Kirchhoff's Voltage Law was successfully used to obtain the differential equation for the modelled circuit's behavior. It can be noted that a linear velocity term appears in this equation coupled with the magnetic field (B). This term will be shown in the next section to be the connecting term between the electrical and mechanical systems.

1.2 Development of Mechanical Differential Equation

1.2.1 Summary of Procedure

For the second portion of this project, the mass-spring-damper mechanical system was analyzed in order to develop the differential equation for its motion. This process was completed using free body diagram analysis as well as Newton's second law ($\sum F = ma$).

1.2.2 Development of Differential Equation

The following figure and equations summarize the process that was utilized to obtain the mass-spring-damper mechanical system's differential equation for its behavior.

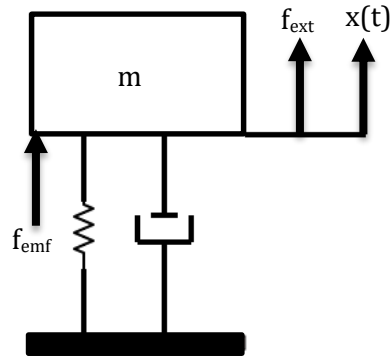


Figure 2. Modelled Mechanical System

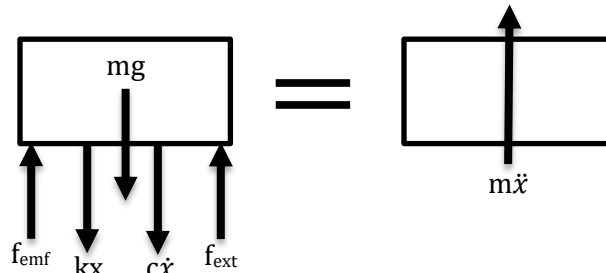


Figure 3. Mechanical System Free Body Diagram

$$f_{emf} = Bi$$

$$+\uparrow \sum F_y = f_{ext} + f_{emf} - c\dot{x} - kx - mg = m\ddot{x}$$

$$-mg + f_{ext} = m\ddot{x} + c\dot{x} + kx - Bi$$

$$m\ddot{x} + c\dot{x} - B\dot{q} + kx = f_{ext} - mg \quad \text{Static Equilibrium}$$

Final Differential Equation

$$m\ddot{x} + c\dot{x} - B\dot{q} + kx = f_{ext}$$

Final Combined Electrical and Mechanical Matrix Equation of Motion

$$\begin{bmatrix} m & 0 \\ 0 & L \end{bmatrix} \begin{Bmatrix} \ddot{x} \\ \ddot{q} \end{Bmatrix} + \begin{bmatrix} c & -B \\ B & R \end{bmatrix} \begin{Bmatrix} \dot{x} \\ \dot{q} \end{Bmatrix} + \begin{bmatrix} k & 0 \\ 0 & 1/C \end{bmatrix} \begin{Bmatrix} x \\ q \end{Bmatrix} = \begin{Bmatrix} f_{ext} \\ V_{ext} \end{Bmatrix}$$

1.2.3 Analysis & Discussion

Newton's second law was successfully used to obtain the differential equation for the modelled mechanical system's behavior. It can be noted that a current term appears in this equation coupled with the magnetic field (B). This confirms the observation that the magnetic field term links the electrical and mechanical systems. When combining the equations of motion, a two degree of freedom matrix equation can be constructed that accurately represents both systems and their interactions.

2. Simulating the Response of the System to Initial Conditions

2.1 Simulating the Response of the System to Initial Conditions

2.1.1 Summary of Procedure

For this portion of the project, the response of the electro-mechanical system was modelled for varying initial conditions and forcing functions. MATLAB was utilized in order to simulate the following responses. General parameters that were applied to all sections of these simulations include:

- $m = 0.025 \text{ kg}$
- $c = 9 \text{ Ns/m}$
- $k = 18 \text{ kN/m}$
- $L = 1.0 \text{ mH}$
- $R = 1 \text{ } \Omega$
- $C = 50 \text{ } \mu\text{F}$
- $B = 6 \text{ W/m}$

2.1.2 Simulation of Initial Conditions (Part 1)

The given initial conditions for the first section of simulation were as follows:

- $x = 0.008 \text{ m}$; $\dot{x} = 0 \text{ m/s}$; $q = 0 \text{ C}$; $\dot{q} = 0 \text{ A}$

The given initial forcing functions for the first section were:

- $f(t) = 0 \text{ N}$; $V(t) = 0 \text{ V}$

The response was plotted over a time range of 0 seconds to 0.05 seconds.

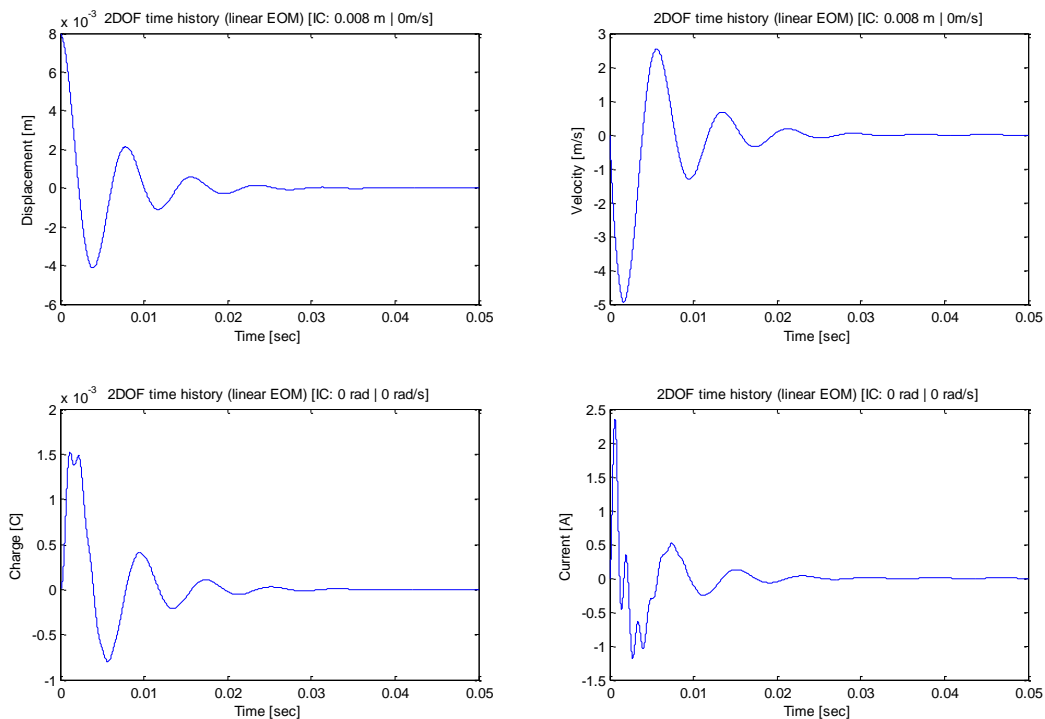


Figure 4. Displacement, Charge, Velocity, Current Response for Part 1

2.1.3 Analysis & Discussion (Part 1)

As seen in Figure 4, the only initial condition was a positive displacement of the mass in the electro-mechanical system. This initial condition was observed to produce a decaying oscillatory motion in both the mechanical and electrical systems that had essentially decayed to zero after 0.03 seconds. This behavior seems logical as the initial displacement of the mass was relatively small (8mm). Also, it can be noted that the displacement and velocity begin out of phase with the charge and current, but after half an oscillation (approximately .005 seconds) appear to move in phase for the remainder of the motion. As a general conclusion, it can be stated that a displacement of 8mm resulted in a 2.25A current.

2.1.4 Simulation of Initial Conditions (Part 2)

The given initial conditions for the second section of simulation were as follows:

- $x = 0 \text{ m}$; $\dot{x} = 0 \text{ m/s}$; $q = 0 \text{ C}$; $\dot{q} = 12.0 \text{ A}$

The given initial forcing functions for the second section were:

- $f(t) = 0 \text{ N}$; $V(t) = 0 \text{ V}$

The response was plotted over a time range of 0 seconds to 0.05 seconds.

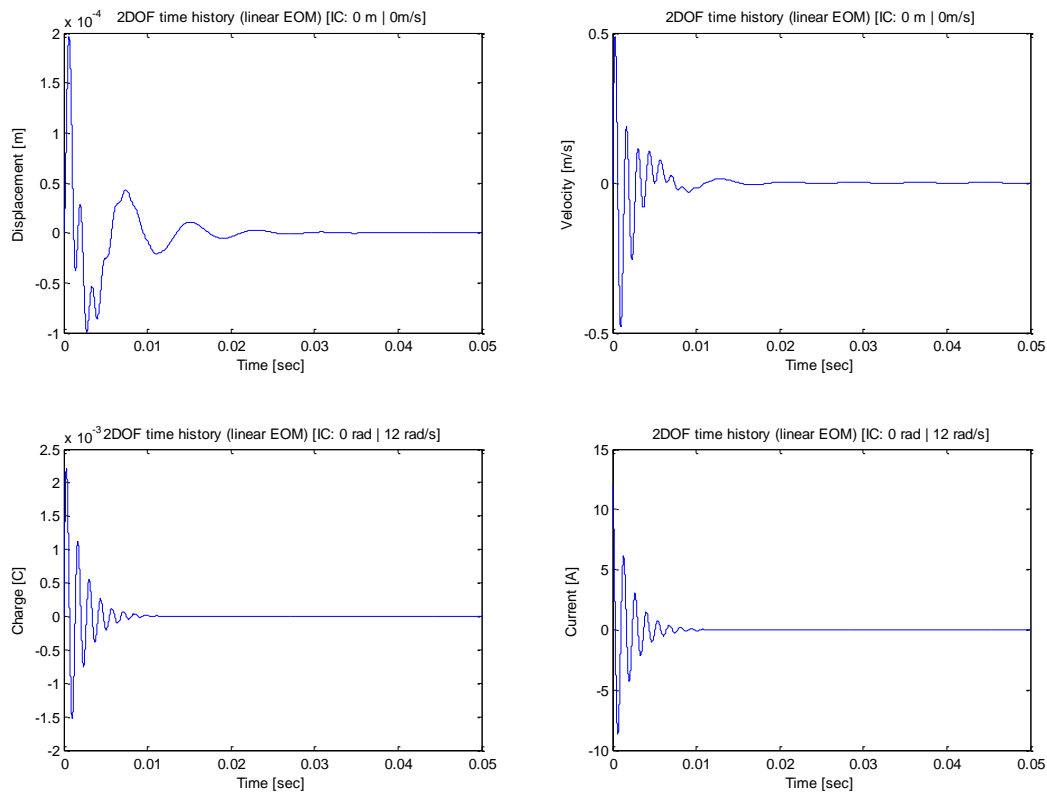


Figure 5. Displacement, Charge, Velocity, Current Response for Part 2

2.1.5 Analysis & Discussion (Part 2)

As seen in Figure 5, the only initial condition was a positive current through the electrical circuit in the electro-mechanical system. This initial condition was observed to produce a decaying oscillatory motion in both the mechanical and electrical systems. In the electrical

circuit, the charge and current had essentially decayed to zero after 0.01 seconds whereas the mechanical system took approximately 0.03 seconds to decay to zero. As a general conclusion, it can be stated that a current of 12A resulted in a 0.2mm displacement. Comparing this result to Part 1, it can also be stated that an initial displacement was considerably more effective at creating a complimentary current than a current was at generating motion in the mass. This is a useful observation when considering how the electro-mechanical device can be most effectively implemented.

2.1.6 Simulation of Initial Conditions (Part 3)

The given initial conditions for the third section of simulation were as follows:

- $x = 0 \text{ m}; \dot{x} = -1 \text{ m/s}; q = 0 \text{ C}; \dot{q} = 5.0 \text{ A}$

The given initial forcing functions for the third section were:

- $f(t) = 0 \text{ N}; V(t) = 0 \text{ V}$

The response was plotted over a time range of 0 seconds to 0.05 seconds.

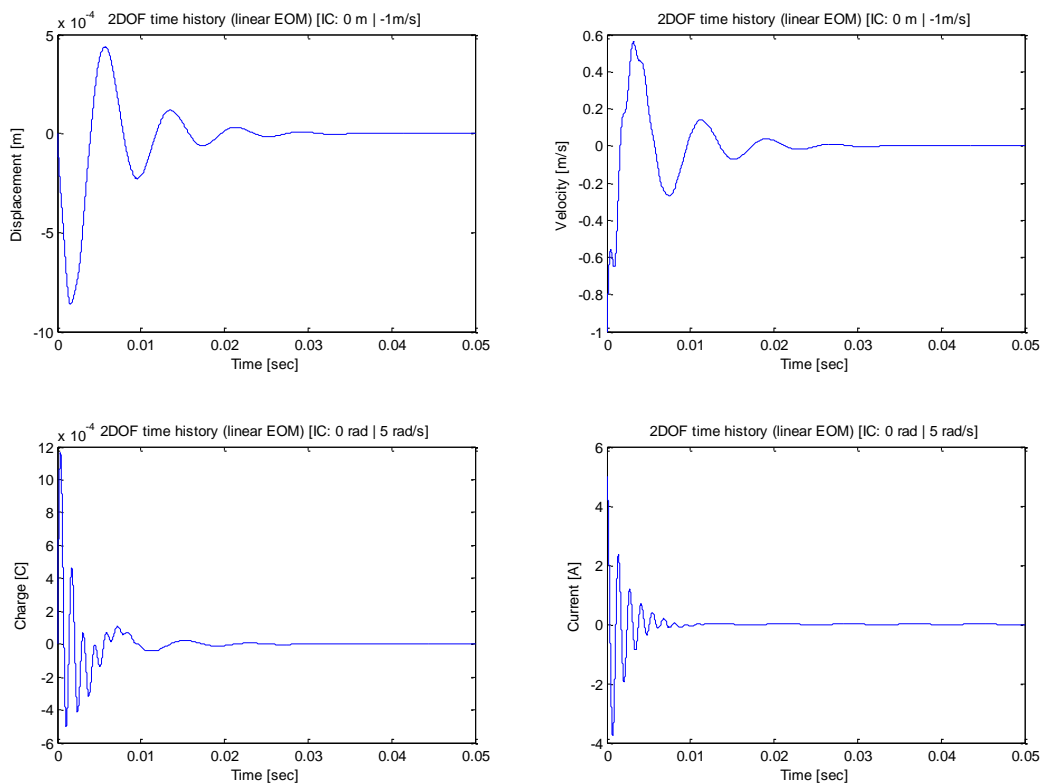


Figure 6. Displacement, Charge, Velocity, Current Response for Part 3

2.1.7 Analysis & Discussion (Part 3)

As seen in Figure 6, the combined initial conditions of velocity and current make it more difficult to isolate the exact impact each had on the system's behavior. However, by comparing this response to the Part 1 (only initial displacement) and 2 (only initial current) some conclusions can begin to be drawn. The combined initial condition was observed to produce a decaying oscillatory motion in both the mechanical and electrical systems. In the electrical circuit, the charge and current had essentially decayed to zero after 0.02 seconds whereas the mechanical system took approximately 0.03 seconds to decay to zero. As a general conclusion, it

can be stated that when current is applied initially (Parts 2 and 3) it can be observed to decay from the system quite rapidly. In addition, an initial velocity was noted to be ineffective at generating a complimentary current in the electrical system as well as does not guarantee a substantial displacement. These conclusions again offer insights into potentially successful applications for this device.

2.1.8 Simulation of Initial Conditions (Part 4)

The given initial conditions for the fourth section of simulation were as follows:

- $x = 0 \text{ m}; \dot{x} = 0 \text{ m/s}; q = 0 \text{ C}; \dot{q} = 0 \text{ A}$

The given initial forcing functions for the fourth section were:

- $f(t) = 0 \text{ N}; V(t) = 9\sin(100\pi t^2) \text{ V}$

The response was plotted over the time range of 0 seconds to 10 seconds and was then further analyzed between the time ranges of 0 seconds to 0.5 seconds and 1.1 seconds to 1.6 seconds.

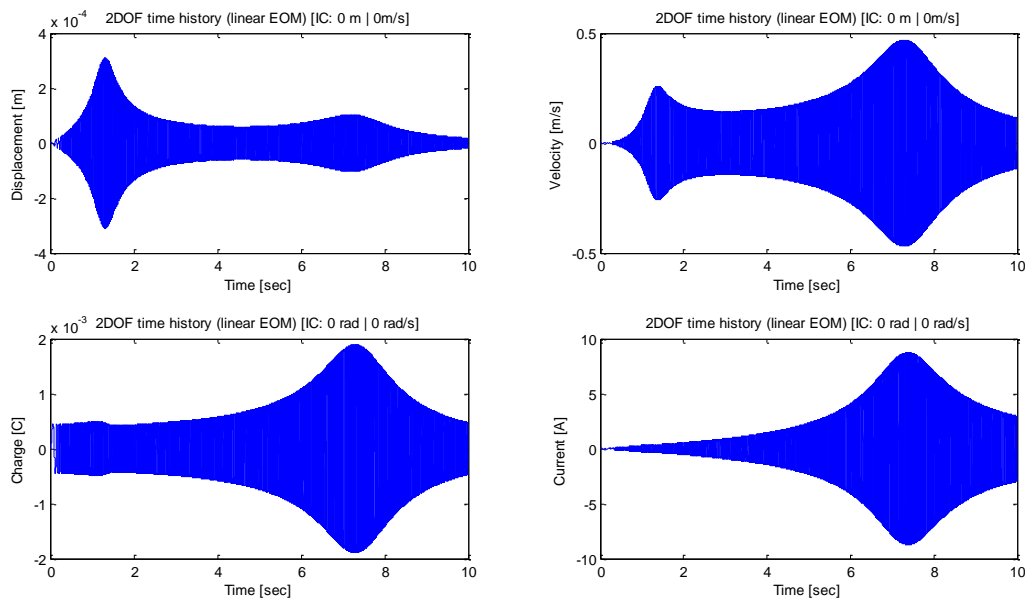


Figure 7. Displacement, Charge, Velocity, Current Response for Part 4

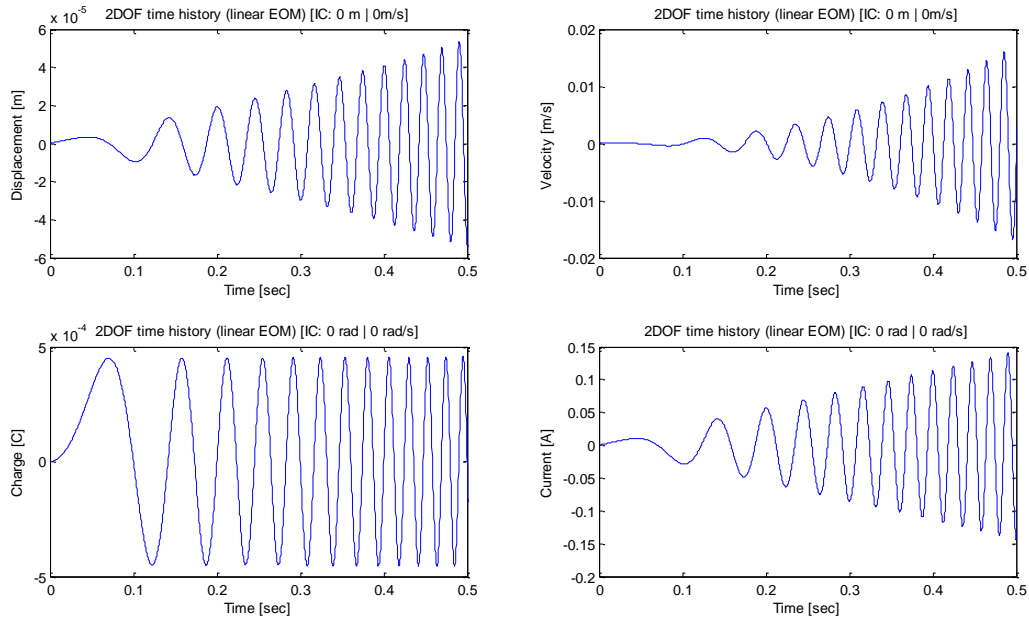


Figure 8. Displacement, Charge, Velocity, Current Response for Part 4 (0-0.5 seconds)

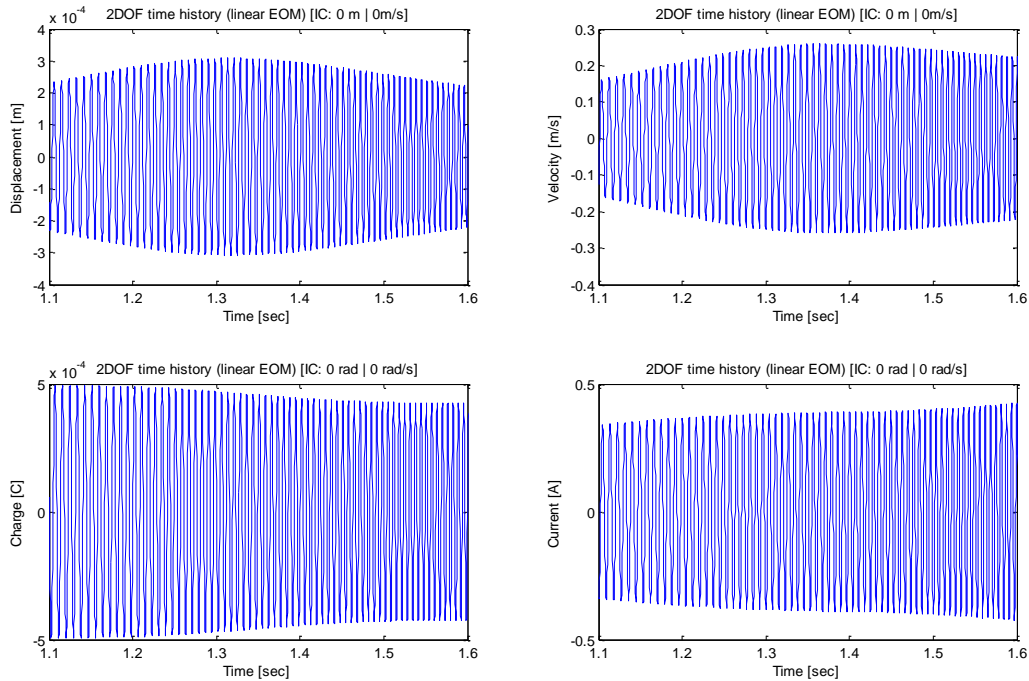


Figure 9. Displacement, Charge, Velocity, Current Response for Part 4 (1.1-1.6 seconds)

2.1.9 Analysis & Discussion (Part 4)

As seen in Figure 7, the only initial condition was a continuous oscillatory voltage through the electrical circuit in the electro-mechanical system. This initial condition was observed to produce a sustained response for displacement, velocity, charge, and current. Comparing Part 4 to the previous three parts, a continuous input voltage was the only initial condition that produced sustained responses. Additionally, Figures 8 and 9 illustrate that the frequency of all

four response components appears to increase over time. This can likely be attributed to the continuous nature of the forcing voltage function. For example, the response at 1.3 seconds is not solely a function of the input voltage at 1.3 seconds, but rather the culmination of all previous responses. This was further highlighted by advice given by the instructor in class to plot the response from 0 seconds regardless of the displayed time frame window.

2.1.10 Simulation of Initial Conditions (Part 5)

The given initial conditions for the fifth section of simulation were as follows:

- $x = 0 \text{ m}$; $\dot{x} = 0 \text{ m/s}$; $q = 0 \text{ C}$; $\dot{q} = 0 \text{ A}$

The given initial forcing functions for the fifth section were:

- $f(t) = 0 \text{ N}$; $V(t) = 12\sin(120\pi t) \text{ V}$ (one half cycle repeating each second)

The response was plotted over a time range of 0 seconds to 10 seconds and was then further analyzed between the time range of 0.98 seconds and 1.08 seconds.

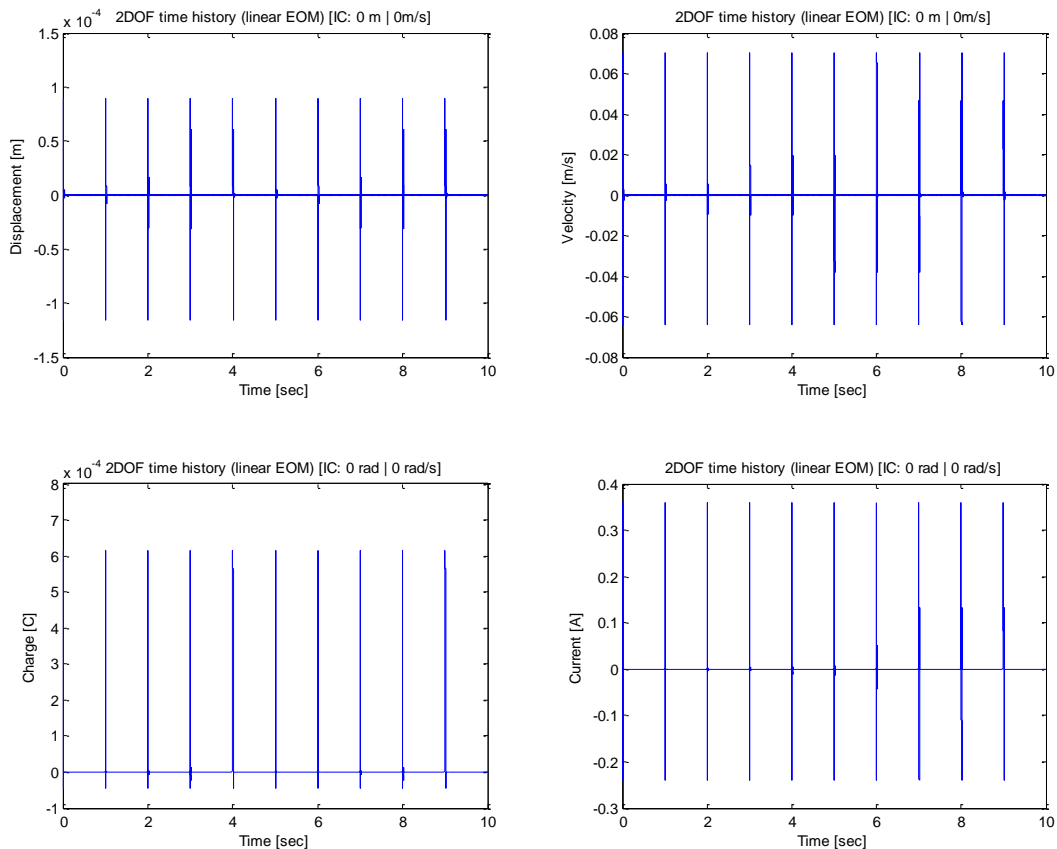


Figure 10. Displacement, Charge, Velocity, Current Response for Part 5

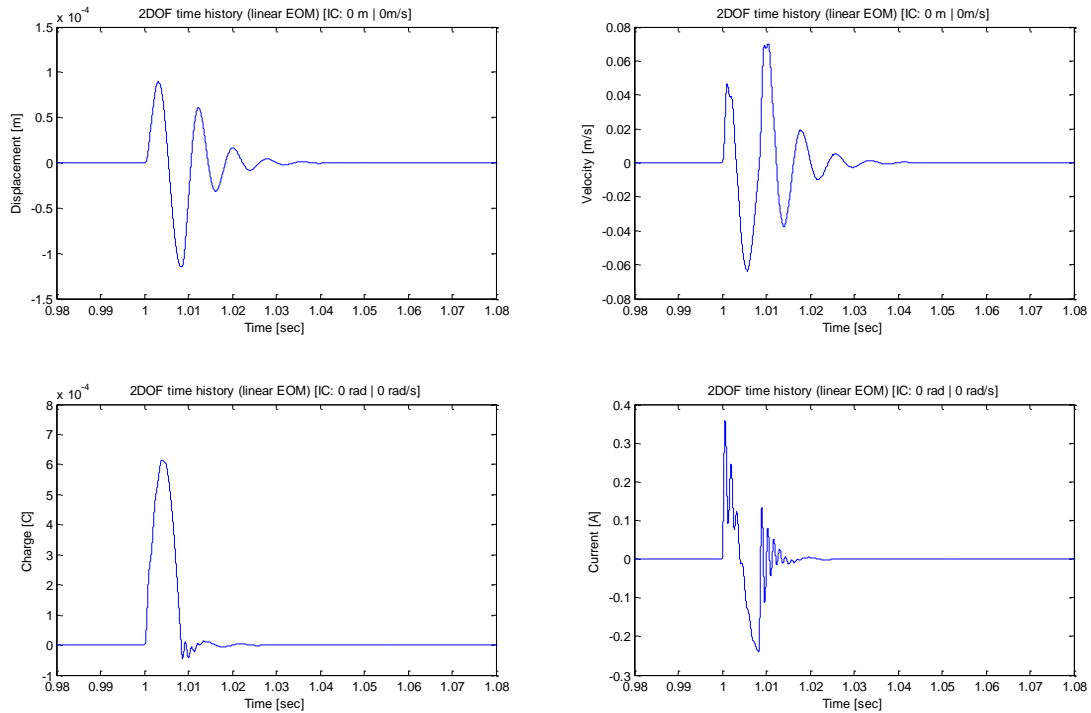


Figure 11. Displacement, Charge, Velocity, Current Response for Part 4 (0.98-1.08 seconds)

2.1.11 Analysis & Discussion (Part 5)

As seen in Figure 11, the initial condition of a repeated pulse does not produce a sustained response similar to the ones observed in Part 4. Each responses decays to zero in approximately 0.03 seconds which is similar to the decay times noted in parts 1 through 3. Therefore, a general conclusion from Part 5 is that a sustained response is unachievable unless a continuous forcing function is applied.

3. Developing the Frequency Response Function and Transmissibility

3.1 Developing and Plotting the Frequency Response Function

3.1.1 Summary of Procedure

For this portion of the project, the multi degree of freedom frequency response function (FRF) was developed using course notes and plotted over a frequency range from 0 Hz to 1000 Hz using MATLAB.

3.1.2 Developing of Frequency Response Function

Final Combined Electrical and Mechanical Matrix Equation of Motion

$$\underbrace{\begin{bmatrix} m & 0 \\ 0 & L \end{bmatrix}}_M \underbrace{\begin{Bmatrix} \ddot{x} \\ \ddot{q} \end{Bmatrix}} + \underbrace{\begin{bmatrix} c & -B \\ B & R \end{bmatrix}}_C \underbrace{\begin{Bmatrix} \dot{x} \\ \dot{q} \end{Bmatrix}} + \underbrace{\begin{bmatrix} k & 0 \\ 0 & 1/C \end{bmatrix}}_K \underbrace{\begin{Bmatrix} x \\ q \end{Bmatrix}} = \underbrace{\begin{Bmatrix} f_{ext} \\ V_{ext} \end{Bmatrix}}$$

Frequency Response Function

$$H(\omega) = [-\omega^2[M] + j\omega[C] + [K]]^{-1}$$

$$H(\omega) = \text{inv}\left(\begin{bmatrix} -\omega^2 m & 0 \\ 0 & -\omega^2 L \end{bmatrix} + \begin{bmatrix} j\omega c & -j\omega B \\ j\omega B & j\omega R \end{bmatrix} + \begin{bmatrix} k & 0 \\ 0 & 1/C \end{bmatrix}\right)$$

$$H(\omega) = \text{inv}\left(\begin{bmatrix} -\omega^2 m + j\omega c + k & -j\omega B \\ j\omega B & -\omega^2 L + j\omega R + 1/C \end{bmatrix}\right)$$

3.1.3 Plotting of Frequency Response Function

To plot this frequency response function over a range of frequencies, the matrix must be evaluated at each frequency resulting in the formation of a three dimensional matrix.

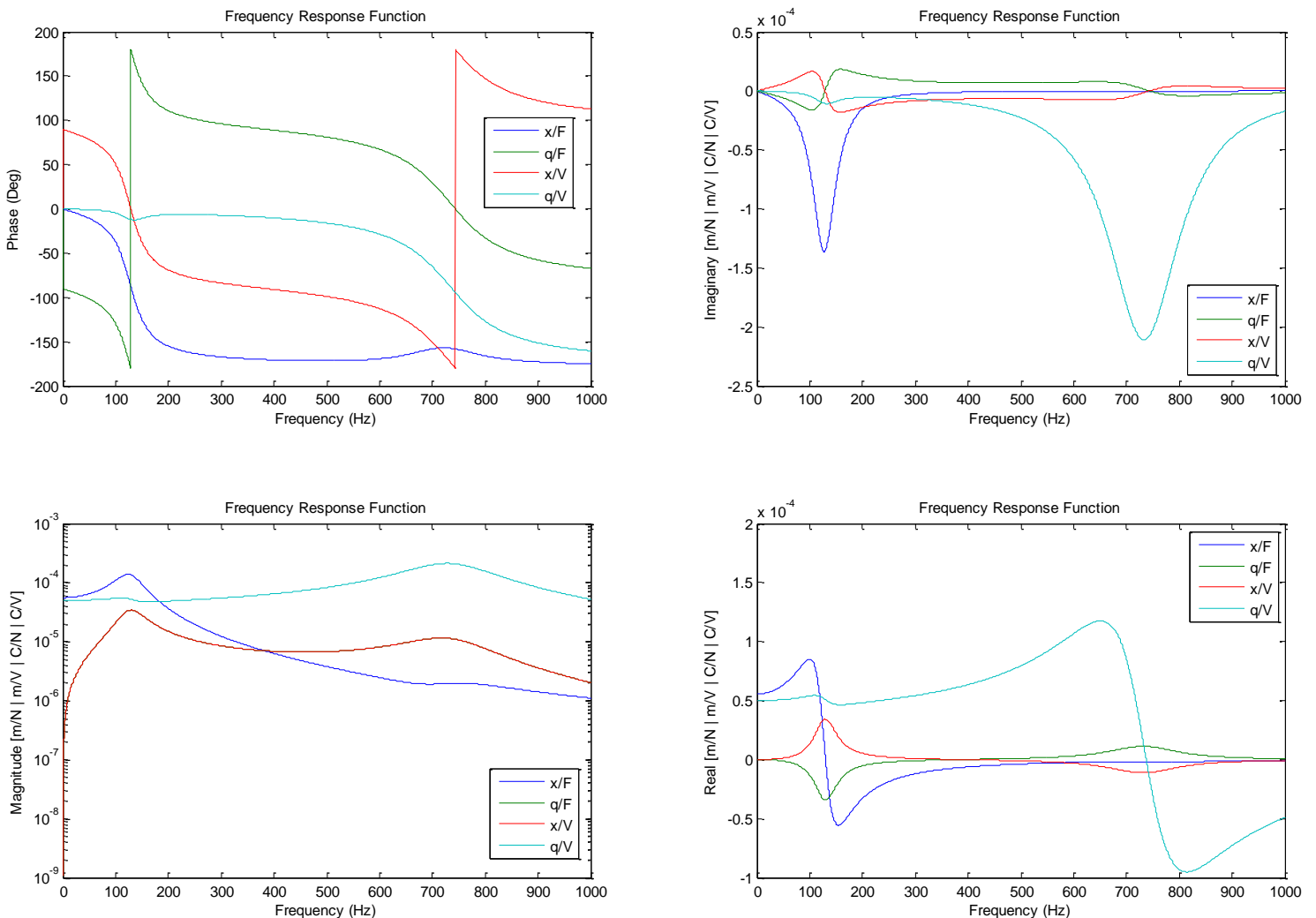


Figure 12. Frequency Response Function

3.1.4 Analysis and Discussion

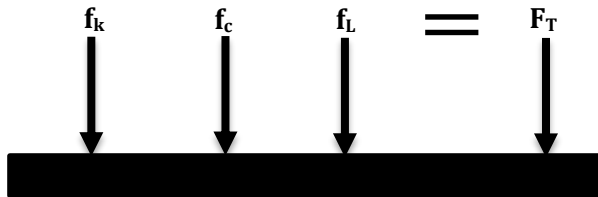
The first general observation from Figure 12 is that the peak magnitudes, specifically for the displacement versus force and the displacement versus voltage curves, align with the relative -90 degree point on the phase portion of the plot. Also, the maximum peaks for the displacement versus force and the displacement versus voltage curves each occur at roughly 150 Hz. As expected from the final combined matrix equation of motion, the magnitudes of the charge versus force and the displacement versus voltage were identical. Finally, 150 Hz (maximum magnitude) corresponds to the maximum real values and zero imaginary values in the plots.

3.2 Developing and Plotting the Transmissibility

3.2.1 Summary of Procedure

For the last portion of the project, the transmissibility to the ground was developed and plotted using MATLAB over a frequency range of 0 Hz to 1000 Hz.

3.2.2 Developing and Plotting of Transmissibility



$$F_T = f_k + f_c + f_L$$

$$F_T = kx + c\dot{x} + B\dot{q}$$

* Note that all forces are expressed in units of Newtons

Transmissibility = Force of transmissibility * Frequency Response Function

$$T = ((j\omega c + k) * "x/V FRF") + (j\omega B * "q/V FRF")$$

Unit Analysis

$$\frac{N}{V} = \left(\left(\frac{1}{s} * \frac{Ns}{m} + \frac{N}{m} \right) * \frac{m}{V} + \left(\frac{1}{s} * \frac{kgm^2}{mAs^2} \right) * \frac{C}{V} \right)$$

$$\frac{N}{V} = \left(\left(\frac{N}{m} + \frac{N}{m} \right) * \frac{m}{V} + \left(\frac{1}{s} * \frac{kgsm}{Cs^2} \right) * \frac{C}{V} \right)$$

$$\frac{N}{V} = \left(\left(\frac{N}{m} + \frac{N}{m} \right) * \frac{m}{V} + \left(\frac{kgm}{Cs^2} \right) * \frac{C}{V} \right)$$

$$\frac{N}{V} = \left(\left(\frac{N}{m} \right) * \frac{m}{V} + \left(\frac{N}{C} \right) * \frac{C}{V} \right) = \frac{N}{V} + \frac{N}{V}$$

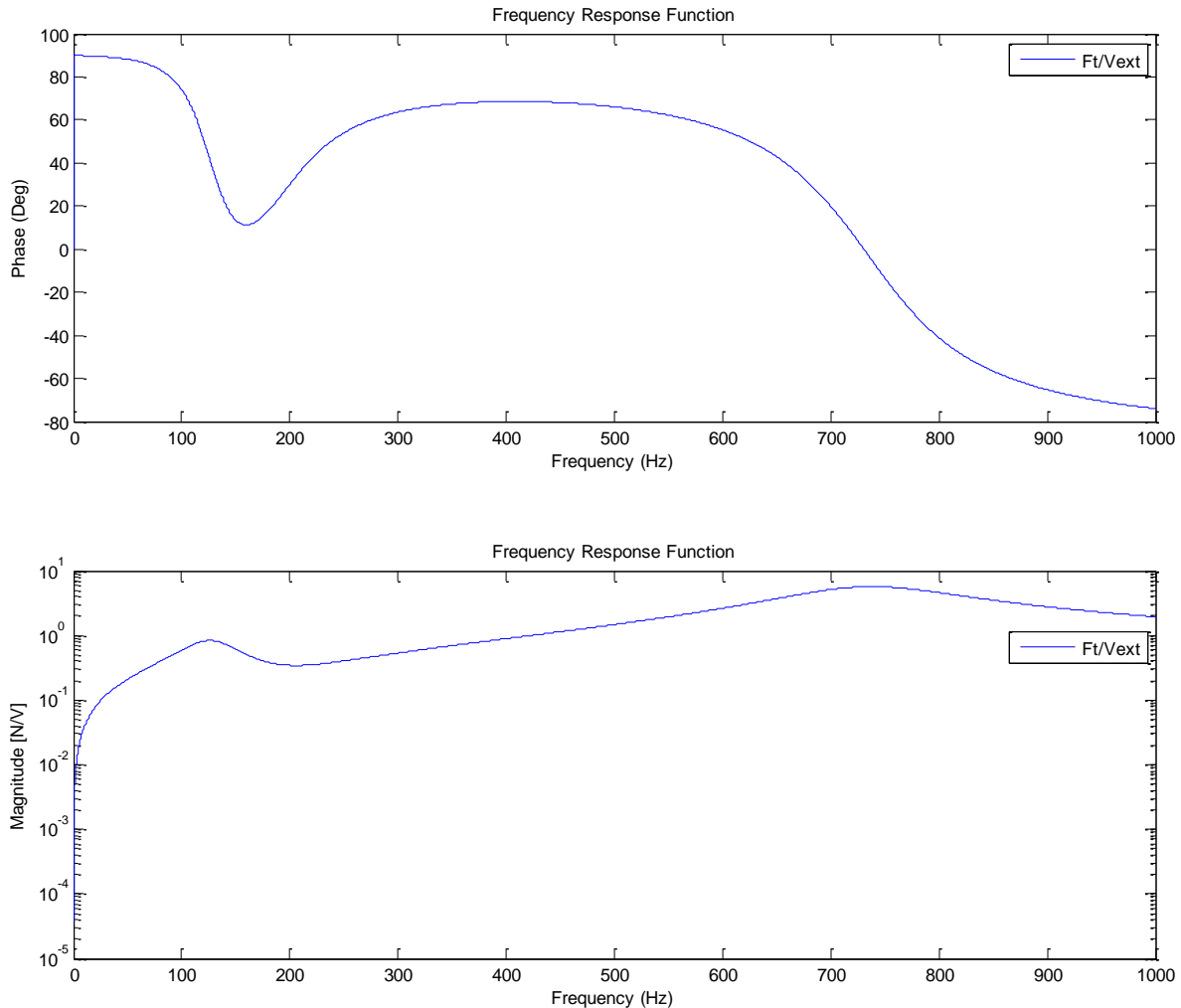


Figure 13. Transmissibility Magnitude and Phase

3.2.3 Analysis and Discussion

Figure 13 shows the relationship between the frequency of the input voltage to the force transmitted to the ground per volt applied. It can be observed that the maximum force transmitted per volt applied occurs at a frequency of approximately 750 Hz. Depending on the intended operation of the electro-mechanical device, this frequency range could either be desired (if a transmitted force is needed – eg the AURA virtual reality immersive gaming vest) or unwanted (if isolation of device is critical – eg see the **WOW** factor).

4. "WOW" Factor

Having completed the analysis for this project, our team is proposing an idea to use this electro-mechanical device in an application that will convert initial displacements into an output current rather than its original function of generating vibrations. This idea was formulated when it was observed in part 2.1 that an initial displacement was more effective at creating a complimentary current than the other way around. The proposal is to use this device under the flow of a waterfall (or similar displacement inducing conditions) in order to generate current for general power applications.

When modelling how this proposed system would behave various portions of this project were considered. As mentioned previously, the observation that displacement is an effective inducer of current served as the inspiration for the concept. Next, 2.1.8 indicated that in order to achieve any form of consistent response that a continuous forcing function was needed. As the goal for this proposal is to generate current, the forcing function focused upon was applied force. Then, the ideal frequency for the forcing function was estimated as 150 Hz using the maximum values from the displacement versus voltage curve on the frequency response function. Finally, the device would likely operate with the intent of minimum transmissibility to the ground as the design is for the electro-mechanical device to absorb as much of the force and displacement as possible.

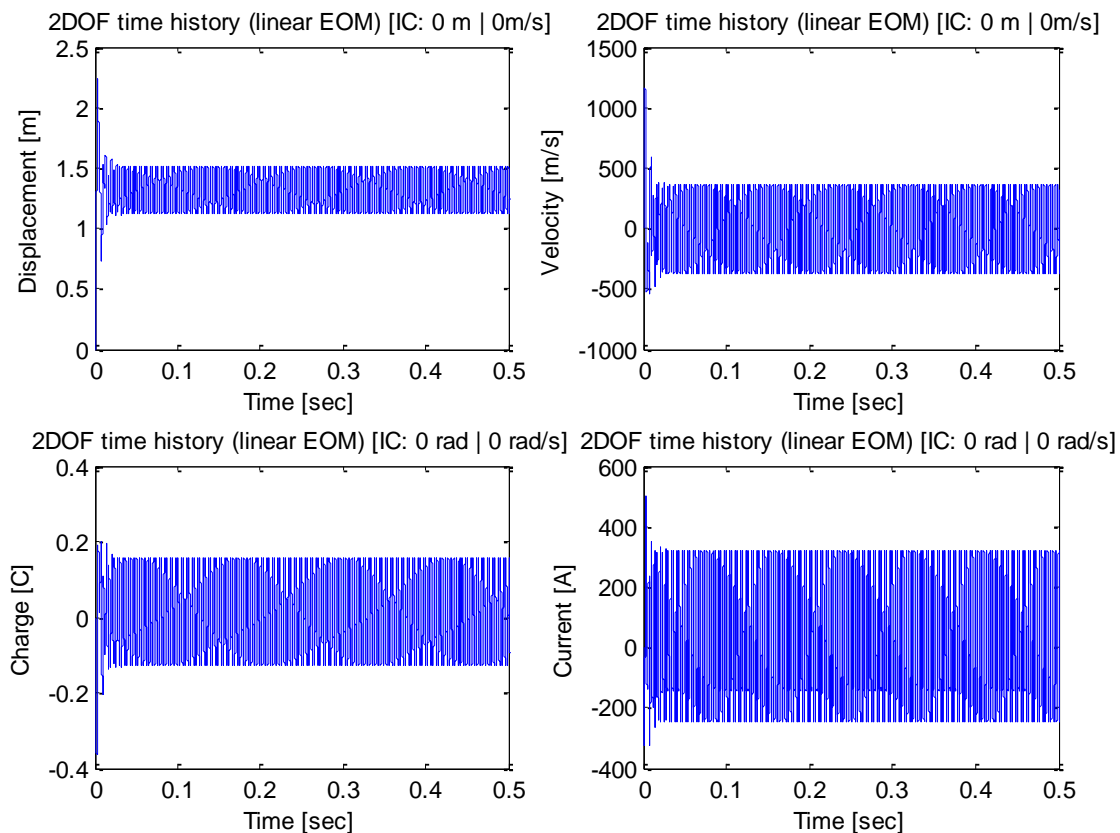


Figure 14. Medium Waterfall Projected Responses

In order to determine the magnitude of the force from the waterfall (used in Figure 14), a waterfall 150 times smaller than Niagra Falls was estimated to have a flow of 1,000 gallons per second (or 37,000 N of force). This was multiplied by the absolute value of the sine function at the ideal frequency mentioned above. Although a true waterfall would not behave as consistently as this mathematical model and current creation could be inhibited by other unknown factors, our team believes this application for the provided electro-mechanical device could feasibly produce current for general power applications.

APPENDICES

A – MATLAB CODE

```
% Part 1
function prob_2dofNL
% Using MATLAB v6 syntax.
%
% Important other related ODE functions: odeset;
global NLflg
% Define time range.
t_start = 0;
t_stop = 0.05;
t0 = [t_start ; t_stop];
% Define initial conditions.
x_vel = 0; % m/s
x_disp = 0.008; % m
q_dot = 0; % A
q_i = 0; % coulomb
y0 = [x_vel ; q_dot ; x_disp; q_i];

NLflg = 0;
options = odeset('RelTol',1e-9,'Refine',12,'InitialStep',1e-9);
[t,y] = ode23(@ndof,t0,y0,options);
subplot(2,2,1), plot(t,y(:,3))
title(['2DOF time history (linear EOM) [IC: ' num2str(x_disp) ' m | '
num2str(x_vel) 'm/s]'])
xlabel('Time [sec]');
ylabel('Displacement [m]');
subplot(2,2,3), plot(t,y(:,4))
title(['2DOF time history (linear EOM) [IC: ' num2str(q_i) ' rad | '
num2str(q_dot) ' rad/s]'])
xlabel('Time [sec]');
ylabel('Charge [C]');

subplot(2,2,2), plot(t,y(:,1))
title(['2DOF time history (linear EOM) [IC: ' num2str(x_disp) ' m | '
num2str(x_vel) 'm/s]'])
xlabel('Time [sec]');
ylabel('Velocity [m/s]');
subplot(2,2,4), plot(t,y(:,2))
title(['2DOF time history (linear EOM) [IC: ' num2str(q_i) ' rad | '
num2str(q_dot) ' rad/s]'])
xlabel('Time [sec]');
ylabel('Current [A]');

% -----
% Function to be integrated
% -----
function ydot = ndof(t,y)
global NLflg
f_ext = 0;

V_ext = 0;
```

```

k = 18000;    % N/m
c = 9;       % Ns/m
B = 6;       % W/m
m = 0.025;   % kg
Cap = 50e-6; % F
R = 1;       % Ohms
L = 1e-3;    % H

xdot = y(1);
qdot = y(2);
x = y(3);
q = y(4);

xdotdot = (f_ext - k*x - c*xdot + B*qdot)/m;
qdotdot = (V_ext - (1/Cap)*q - B*xdot - R*qdot)/L;

ydot = [xdotdot ; qdotdot; xdot; qdot];

% Part 2
function prob_2dofNL
% Using MATLAB v6 syntax.
%
% Important other related ODE functions: odeset;
global NLflg
% Define time range.
t_start = 0;
t_stop = 0.05;
t0 = [t_start ; t_stop];
% Define initial conditions.
x_vel = 0; % m/s
x_disp = 0; % m
q_dot = 12; % A
q_i = 0; % coulomb
y0 = [x_vel ; q_dot ; x_disp; q_i];

NLflg = 0;
options = odeset('RelTol',1e-9,'Refine',12,'InitialStep',1e-9);
[t,y] = ode23(@ndof,t0,y0,options);
subplot(2,2,1), plot(t,y(:,3))
title(['2DOF time history (linear EOM) [IC: ' num2str(x_disp) ' m | '
num2str(x_vel) ' m/s]'])
xlabel('Time [sec]');
ylabel('Displacement [m]');
subplot(2,2,3), plot(t,y(:,4))
title(['2DOF time history (linear EOM) [IC: ' num2str(q_i) ' rad | '
num2str(q_dot) ' rad/s]'])
xlabel('Time [sec]');
ylabel('Charge [C]');

subplot(2,2,2), plot(t,y(:,1))
title(['2DOF time history (linear EOM) [IC: ' num2str(x_disp) ' m | '
num2str(x_vel) ' m/s]'])
xlabel('Time [sec]');
ylabel('Velocity [m/s]');

```

```

subplot(2,2,4), plot(t,y(:,2))
title(['2DOF time history (linear EOM) [IC: ' num2str(q_i) ' rad | '
num2str(q_dot) ' rad/s]'])
xlabel('Time [sec]');
ylabel('Current [A]');

% -----
% Function to be integrated
% -----
function ydot = ndof(t,y)
global NLflg
f_ext = 0;

V_ext = 0;

k = 18000; % N/m
c = 9; % Ns/m
B = 6; % W/m
m = 0.025; % kg
Cap = 50e-6; % F
R = 1; % Ohms
L = 1e-3; % H

xdot = y(1);
qdot = y(2);
x = y(3);
q = y(4);

xdotdot = (f_ext - k*x - c*xdot + B*qdot)/m;
qdotdot = (V_ext - (1/Cap)*q - B*xdot - R*qdot)/L;

ydot = [xdotdot ; qdotdot; xdot; qdot];

% Part 3
function prob_2dofNL
% Using MATLAB v6 syntax.
%
% Important other related ODE functions: odeset;
global NLflg
% Define time range.
t_start = 0;
t_stop = 0.05;
t0 = [t_start ; t_stop];
% Define initial conditions.
x_vel = -1; % m/s
x_disp = 0; % m
q_dot = 5; % A
q_i = 0; % coulomb
y0 = [x_vel ; q_dot ; x_disp; q_i];

NLflg = 0;
options = odeset('RelTol',1e-9,'Refine',12,'InitialStep',1e-9);
[t,y] = ode23(@ndof,t0,y0,options);
subplot(2,2,1), plot(t,y(:,3))

```

```

title(['2DOF time history (linear EOM) [IC: ' num2str(x_disp) ' m | '
num2str(x_vel) 'm/s]'])
xlabel('Time [sec]');
ylabel('Displacement [m]');
subplot(2,2,3), plot(t,y(:,4))
title(['2DOF time history (linear EOM) [IC: ' num2str(q_i) ' rad | '
num2str(q_dot) ' rad/s]'])
xlabel('Time [sec]');
ylabel('Charge [C]');

subplot(2,2,2), plot(t,y(:,1))
title(['2DOF time history (linear EOM) [IC: ' num2str(x_disp) ' m | '
num2str(x_vel) 'm/s]'])
xlabel('Time [sec]');
ylabel('Velocity [m/s]');
subplot(2,2,4), plot(t,y(:,2))
title(['2DOF time history (linear EOM) [IC: ' num2str(q_i) ' rad | '
num2str(q_dot) ' rad/s]'])
xlabel('Time [sec]');
ylabel('Current [A]');

% -----
% Function to be integrated
% -----
function ydot = ndof(t,y)
global NLflg
f_ext = 0;

V_ext = 0;

k = 18000; % N/m
c = 9; % Ns/m
B = 6; % W/m
m = 0.025; % kg
Cap = 50e-6; % F
R = 1; % Ohms
L = 1e-3; % H

xdot = y(1);
qdot = y(2);
x = y(3);
q = y(4);

xdotdot = (f_ext - k*x - c*xdot + B*qdot)/m;
qdotdot = (V_ext - (1/Cap)*q - B*xdot - R*qdot)/L;

ydot = [xdotdot ; qdotdot; xdot; qdot];

% Part 4
function prob_2dofNL
% Using MATLAB v6 syntax.
%
% Important other related ODE functions: odeset;
global NLflg
% Define time range.

```

```

t_start = 0;
t_stop = 10;
t0 = [t_start ; t_stop];
% Define initial conditions.
x_vel = 0; % m/s
x_disp = 0; % m
q_dot = 0; % A
q_i = 0; % coulomb
y0 = [x_vel ; q_dot ; x_disp; q_i];

NLflg = 0;
options = odeset('RelTol',1e-9,'Refine',12,'InitialStep',1e-9);
[t,y] = ode23(@ndof,t0,y0,options);
subplot(2,2,1), plot(t,y(:,3))
title(['2DOF time history (linear EOM) [IC: ' num2str(x_disp) ' m | '
num2str(x_vel) 'm/s]'])
xlabel('Time [sec]');
ylabel('Displacement [m]');
subplot(2,2,3), plot(t,y(:,4))
title(['2DOF time history (linear EOM) [IC: ' num2str(q_i) ' rad | '
num2str(q_dot) ' rad/s]'])
xlabel('Time [sec]');
ylabel('Charge [C]');

subplot(2,2,2), plot(t,y(:,1))
title(['2DOF time history (linear EOM) [IC: ' num2str(x_disp) ' m | '
num2str(x_vel) 'm/s]'])
xlabel('Time [sec]');
ylabel('Velocity [m/s]');
subplot(2,2,4), plot(t,y(:,2))
title(['2DOF time history (linear EOM) [IC: ' num2str(q_i) ' rad | '
num2str(q_dot) ' rad/s]'])
xlabel('Time [sec]');
ylabel('Current [A]');

figure(2)
subplot(2,2,1), plot(t,y(:,3))
title(['2DOF time history (linear EOM) [IC: ' num2str(x_disp) ' m | '
num2str(x_vel) 'm/s]'])
xlim([0 0.5]);
xlabel('Time [sec]');
ylabel('Displacement [m]');
subplot(2,2,3), plot(t,y(:,4))
title(['2DOF time history (linear EOM) [IC: ' num2str(q_i) ' rad | '
num2str(q_dot) ' rad/s]'])
xlim([0 0.5]);
xlabel('Time [sec]');
ylabel('Charge [C]');

subplot(2,2,2), plot(t,y(:,1))
title(['2DOF time history (linear EOM) [IC: ' num2str(x_disp) ' m | '
num2str(x_vel) 'm/s]'])
xlim([0 0.5]);
xlabel('Time [sec]');
ylabel('Velocity [m/s]');

```

```

subplot(2,2,4), plot(t,y(:,2))
title(['2DOF time history (linear EOM) [IC: ' num2str(q_i) ' rad | '
num2str(q_dot) ' rad/s]'])
xlim([0 0.5]);
xlabel('Time [sec]');
ylabel('Current [A]');

figure(3)
subplot(2,2,1), plot(t,y(:,3))
title(['2DOF time history (linear EOM) [IC: ' num2str(x_disp) ' m | '
num2str(x_vel) 'm/s]'])
xlim([1.1 1.6]);
xlabel('Time [sec]');
ylabel('Displacement [m]');
subplot(2,2,3), plot(t,y(:,4))
xlim([1.1 1.6]);
title(['2DOF time history (linear EOM) [IC: ' num2str(q_i) ' rad | '
num2str(q_dot) ' rad/s]'])
xlabel('Time [sec]');
ylabel('Charge [C]');

subplot(2,2,2), plot(t,y(:,1))
title(['2DOF time history (linear EOM) [IC: ' num2str(x_disp) ' m | '
num2str(x_vel) 'm/s]'])
xlim([1.1 1.6]);
xlabel('Time [sec]');
ylabel('Velocity [m/s]');
subplot(2,2,4), plot(t,y(:,2))
title(['2DOF time history (linear EOM) [IC: ' num2str(q_i) ' rad | '
num2str(q_dot) ' rad/s]'])
xlim([1.1 1.6]);
xlabel('Time [sec]');
ylabel('Current [A]');

% -----
% Function to be integrated
% -----
function ydot = ndof(t,y)
global NLflg
f_ext = 0;

V_ext = 9*sin(100*pi*t.^2);

k = 18000; % N/m
c = 9; % Ns/m
B = 6; % W/m
m = 0.025; % kg
Cap = 50e-6; % F
R = 1; % Ohms
L = 1e-3; % H

xdot = y(1);
qdot = y(2);
x = y(3);
q = y(4);

```



```

xdotdot = (f_ext - k*x - c*xdot + B*qdot)/m;
qdotdot = (V_ext - (1/Cap)*q - B*xdot - R*qdot)/L;

ydot = [xdotdot ; qdotdot; xdot; qdot];

% Part 5
function prob_2dofNL
% Using MATLAB v6 syntax.
%
% Important other related ODE functions: odeset;
global NLflg
% Define time range.
t_start = 0;
t_stop = 10;
t0 = [t_start ; t_stop];
% Define initial conditions.
x_vel = 0; % m/s
x_disp = 0; % m
q_dot = 0; % A
q_i = 0; % coulomb
y0 = [x_vel ; q_dot ; x_disp; q_i];

NLflg = 0;
options = odeset('RelTol',1e-9,'Refine',12,'InitialStep',1e-9);
[t,y] = ode45(@ndof,t0,y0,options);
subplot(2,2,1), plot(t,y(:,3))
title(['2DOF time history (linear EOM) [IC: ' num2str(x_disp) ' m | '
num2str(x_vel) 'm/s]'])
xlabel('Time [sec]');
ylabel('Displacement [m]');
subplot(2,2,3), plot(t,y(:,4))
title(['2DOF time history (linear EOM) [IC: ' num2str(q_i) ' rad | '
num2str(q_dot) ' rad/s]'])
xlabel('Time [sec]');
ylabel('Charge [C]');

subplot(2,2,2), plot(t,y(:,1))
title(['2DOF time history (linear EOM) [IC: ' num2str(x_disp) ' m | '
num2str(x_vel) 'm/s]'])
xlabel('Time [sec]');
ylabel('Velocity [m/s]');
subplot(2,2,4), plot(t,y(:,2))
title(['2DOF time history (linear EOM) [IC: ' num2str(q_i) ' rad | '
num2str(q_dot) ' rad/s]'])
xlabel('Time [sec]');
ylabel('Current [A]');

figure(2)
subplot(2,2,1), plot(t,y(:,3))
title(['2DOF time history (linear EOM) [IC: ' num2str(x_disp) ' m | '
num2str(x_vel) 'm/s]'])
xlim([0.98 1.08]);
xlabel('Time [sec]');
ylabel('Displacement [m]');

```

```

subplot(2,2,3), plot(t,y(:,4))
title(['2DOF time history (linear EOM) [IC: ' num2str(q_i) ' rad | '
num2str(q_dot) ' rad/s]'])
xlim([0.98 1.08]);
xlabel('Time [sec]');
ylabel('Charge [C]');

subplot(2,2,2), plot(t,y(:,1))
title(['2DOF time history (linear EOM) [IC: ' num2str(x_disp) ' m | '
num2str(x_vel) 'm/s]'])
xlim([0.98 1.08]);
xlabel('Time [sec]');
ylabel('Velocity [m/s]');
subplot(2,2,4), plot(t,y(:,2))
title(['2DOF time history (linear EOM) [IC: ' num2str(q_i) ' rad | '
num2str(q_dot) ' rad/s]'])
xlim([0.98 1.08]);
xlabel('Time [sec]');
ylabel('Current [A]');

% -----
% Function to be integrated
% -----
function ydot = ndof(t,y)
global NLflg
f_ext = 0;

V_ext = VPulse(t);

k = 18000; % N/m
c = 9; % Ns/m
B = 6; % W/m
m = 0.025; % kg
Cap = 50e-6; % F
R = 1; % Ohms
L = 1e-3; % H

xdot = y(1);
qdot = y(2);
x = y(3);
q = y(4);

xdotdot = (f_ext - k*x - c*xdot + B*qdot)/m;
qdotdot = (V_ext - (1/Cap)*q - B*xdot - R*qdot)/L;

ydot = [xdotdot ; qdotdot; xdot; qdot];

%% Part Ca
clear
clc
hz = 0:.01:1000;
omega = hz*2*pi;

```

```

m = 0.025; % kg
l = 1e-3; % H
c = 9; % Ns/m
b = 6; % W/m
r = 1; % Ohms
k = 18000; % N/m
cap = 50e-6; % F

M = [m 0; 0 l];
C = [c -b; b r];
K = [k 0; 0 1/cap];

h = zeros(2,2,length(omega));
h1 = zeros(1,length(omega));
h2 = zeros(1,length(omega));
h3 = zeros(1,length(omega));
h4 = zeros(1,length(omega));

for n = 1:length(omega)
    h(:, :, n) = inv(-(omega(n).^2)*M + j*omega(n)*C + K);
    h1(1,n) = h(1,1,n);
    h2(1,n) = h(2,1,n);
    h3(1,n) = h(1,2,n);
    h4(1,n) = h(2,2,n);
end

subplot(2,2,3);
semilogy(hz,abs(h1),hz,abs(h2),hz,abs(h3),hz,abs(h4));
xlabel('Frequency (Hz)'); ylabel('Magnitude [m/N | m/V | C/N | C/V]');
title('Frequency Response Function');
legend('x/F', 'q/F', 'x/V', 'q/V');

subplot(2,2,1)
plot(hz,angle(h1)*180/pi,hz,angle(h2)*180/pi,hz,angle(h3)*180/pi,hz,angle(h4)
*180/pi);
xlabel('Frequency (Hz)'); ylabel('Phase (Deg)');
title('Frequency Response Function');
legend('x/F', 'q/F', 'x/V', 'q/V');

subplot(2,2,4);
plot(hz,real(h1),hz,real(h2),hz,real(h3),hz,real(h4));
xlabel('Frequency (Hz)'); ylabel('Real [m/N | m/V | C/N | C/V]');
title('Frequency Response Function');
legend('x/F', 'q/F', 'x/V', 'q/V');

subplot(2,2,2)
plot(hz,imag(h1),hz,imag(h2),hz,imag(h3),hz,imag(h4));
xlabel('Frequency (Hz)'); ylabel('Imaginary [m/N | m/V | C/N | C/V]');
title('Frequency Response Function');
legend('x/F', 'q/F', 'x/V', 'q/V');

%% Part Cb
for n = 1:length(omega)
    T(n) = ((j.*omega(n)*c + k).*h3(n)) + (j.*b*omega(n).*h4(n));
end

```

```
subplot(2,1,2);
semilogy(hz,abs(T));
xlabel('Frequency (Hz)'); ylabel('Magnitude [N/V]');
title('Frequency Response Function');
legend('Ft/Vext');

subplot(2,1,1)
plot(hz,angle(T)*180/pi);
xlabel('Frequency (Hz)'); ylabel('Phase (Deg)');
title('Frequency Response Function');
legend('Ft/Vext');
```