Project Report

To begin, models for the non-linear spring and dampers were made, using two functions (calculateSpringForce() and caculateDamperForce()). These functions created models based on the equations of $F_{spring} = -k\left(d + \alpha d^{\beta}\right)$ and $F_{damper} = -cv + sign(v)\mu N.$ These functions can be used to model both linear and non-linear springs and dampers by changing the value of the inputs of spring coefficients and damper friction force to zero. These functions were used to create two figures from ranges of $-1 \leq d \leq 1\ m$ and $-1 \leq v \leq 1\ m$/s (Figures 1 and 2 in the appendix).

Then, a model to determine the response of the mass-spring-damper-suspension system versus time is created with varying input and system parameters. To do this, the previous functions are called to calculate the spring and damper forces for each scenario. This function uses a coupled ODE in order to find the vectors of time, position, velocity, and total force for the combination of parameters. For efficiency purposes, the differential equation chosen is a fourth order Runge Kutta method in order to optimize time and the number of calculations. These coupled equations were derived from the formula of $mx" = -k(x - y) - c(x' - y')$ where y' can be found analytically from $y = Ysin(wt)$The coupled equations help find the vectors of time, position and velocity, and are then used to calculate the value of parameters used in our Force equations/functions. With this function a figure is created to compare the linear and non-linear system responses versus time using parameters and solver settings supplied in Tables 1 and 3, respectively (Task 2 Figure A).

Continuing, a model to test convergence of the solution of the mass-spring-damper-system response. This function returns vectors of time, position, velocity, force, step sizes and the approximate errors. This function calculates the system response for multiple step sizes, decreasing the step size until the maximum approximate error of the position of that system response versus the previous calculation is below a certain error tolerance. In order to do this, a loop is created to call calculateSystemResponse() and create a vector of time and position for different step sizes, halving the step size until we achieve the error tolerance we want. Each iteration of the loop compares its time and position vector to the previous iteration's in order to find the approximate error. Since altering the step size alters the time vector, the overlap between the two time vectors needs to be found first.

Then, the positions at those overlapping times is found and compared to find approximate error (displacement errors). Finally, the maximum approximate error of the iteration is found and added to a vector of approximate errors with altering step size. This maximum approximate error is also compared to our error tolerance; if it is below, the loop will terminate, if not, it will iterate again. A figure (Task 3 Figure A) is then created to compare linear and nonlinear displacement errors versus time step. This figure also includes a line for our displacement error tolerance. Another figure (Task 3 Figure B) is also created to compare the linear and nonlinear system response versus time using the converged solution. This figure also includes base motion.

Additionally, two more functions are created: calculateDisplacementAmplitude() and calculateForceAmplitude(). Each of these functions calls our Converged System Response function and retrieves a vector of time and either position or force (depending on what we want to calculate). In order to minimize calculation time, the final converged timesteps previously calculated were used as the initial time steps for these functions. First, the time vector is sorted through to only use values of force or position after the time has reached its steady state (in this case around 130 sec). Then, the displacement amplitude is found by the equation $X = \frac{(x_{max} - x_{min})}{2}$ using the position values after the steady state. The Force Amplitude is calculated in a similar way, applying the equation $F = \frac{F_{x,max} - F_{x,min}}{2}$ to the force values after the steady state.

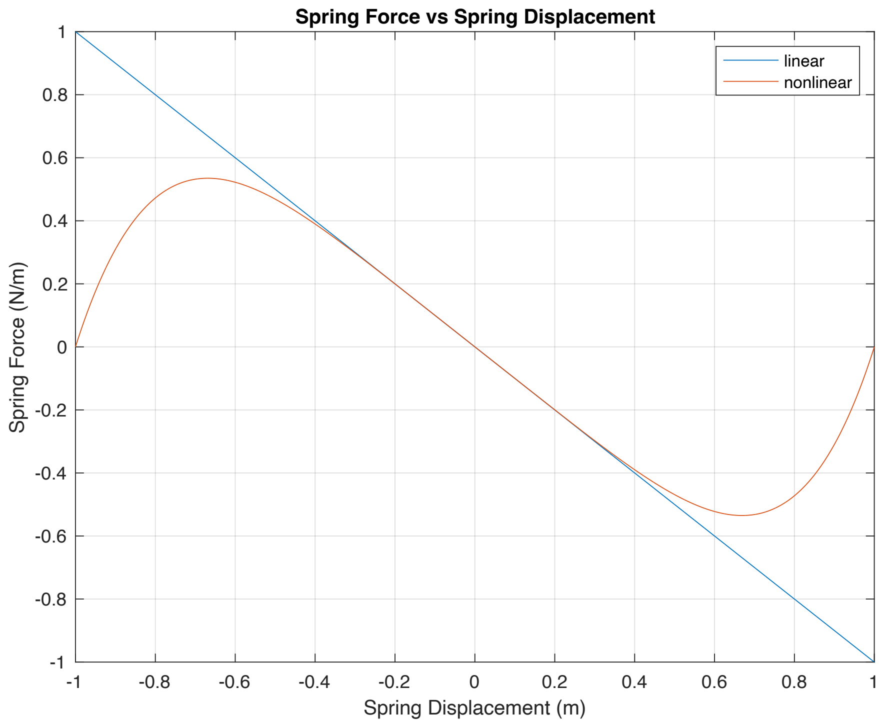
Using these functions, the displacement amplitude and force are found for frequencies from 0.5 rad/s to 5 rad/s, in steps of 0.5. A figure (Task 4 Figure A) is then created to show displacement amplitude versus frequency for both linear and nonlinear systems. This figure also includes the expected displacement amplitude for the linear system found using the analytical solution using formula $X = Y * \sqrt{\frac{k^2+(cw)^2}{(k-mw^2)^2+(cw)^2}}$. Another figure (Task 4 Figure B) is created to show force amplitude versus frequency for both linear and nonlinear systems. This figure also includes the expected force amplitude for the linear system using the analytical solution of formula $F = kY * \left(\frac{w}{w_n}\right)^2 * \sqrt{\frac{k^2+(cw)^2}{(k-mw^2)^2+(cw)^2}}$.

For the next task, the damped natural frequency for linear system, nonlinear system and analytical solutions are found. This is found using the Golden Section Search method, which is an optimization method. The Golden Section Search was chosen because it does not require a derivative and can be given an upper and lower bound to search for a max. Using the plotted figure, the initial bounds are set to 1.25 and 1.75 based on where the Td figure seems to have a max. This returns a damped natural frequencies of 1.4124, 1.4122, and 1.4002 for the analytical, linear, and nonlinear systems respectively. These frequencies are added to Task 4 Figure A. For the linear and non-linear systems, the calculateDisplacementAmplitude() is turned into a function handle to be used as an input for the Golden Section Search.

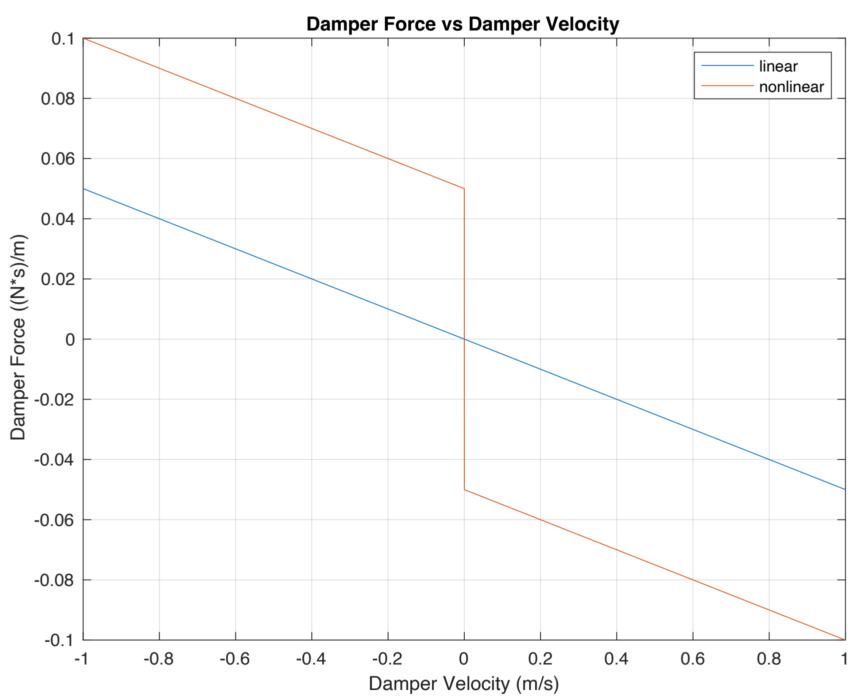
For the final task, we need to find the operating frequency range of the systems. To do this, the False Position method is used since it too does not require a derivative and can be supplied an initial upper and lower bound to guess. This is applied by subtracting the max acceptable value from the formula for our X, Td, F, or Tf calculation to shift the output down. Then we find where the shifted calculation hits zero, which will be where the original calculation would intersect with the max value. The ranges for X and Td in bound are exactly the same, as are Tf and F. These ranges were included in Table 1 of the appendix. The bounds were also included in both figures (Task 4 Figure A and Figure B).

Appendix

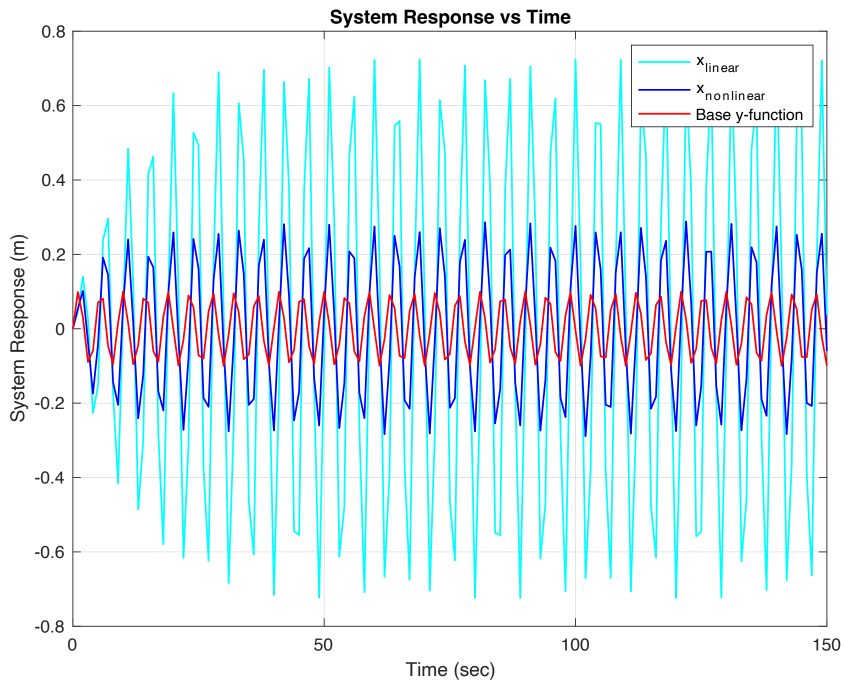
Figures



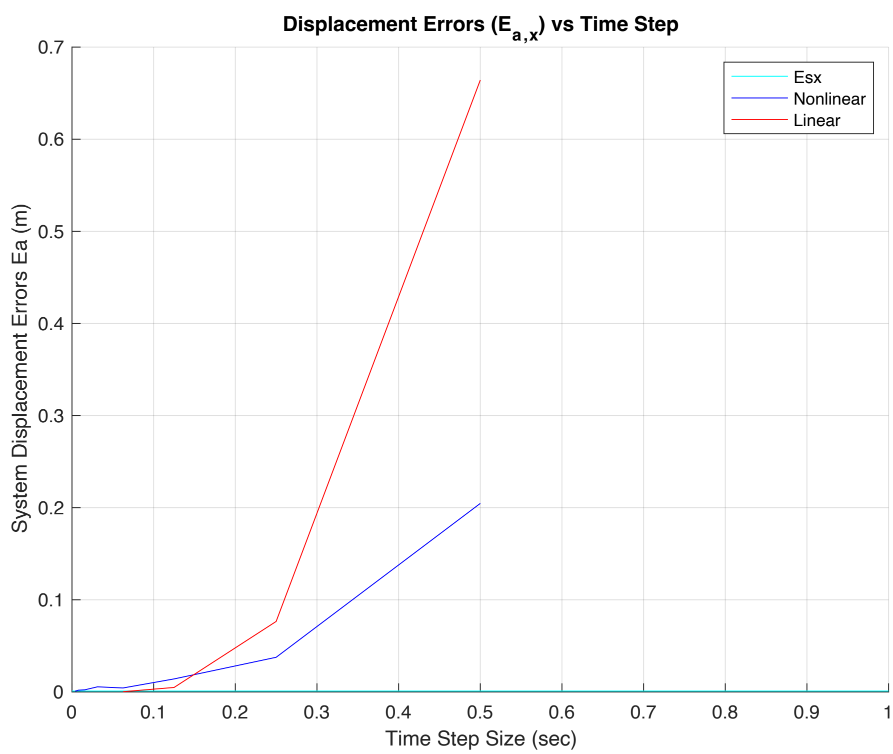
Task 1 Figure A



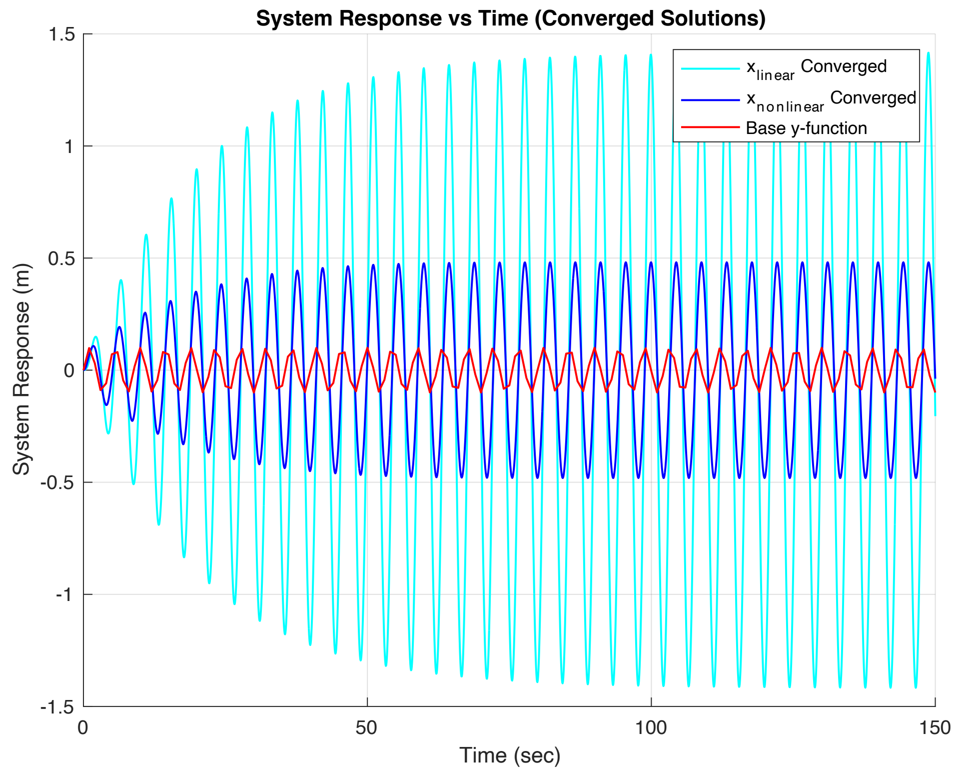
Task 1 Figure B



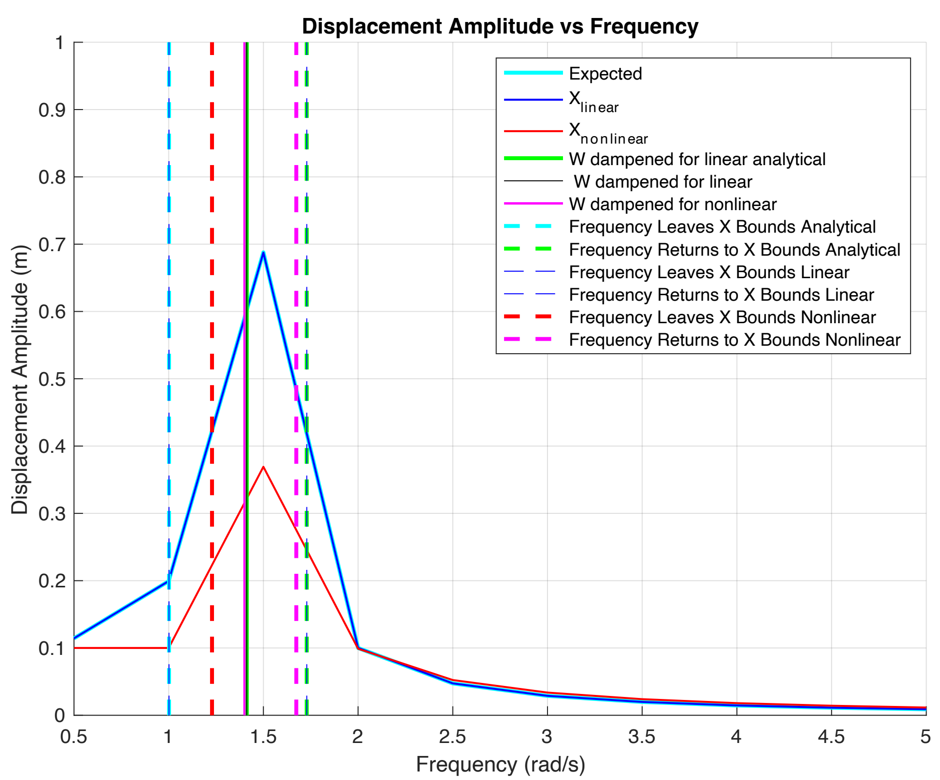
Task 2 Figure A



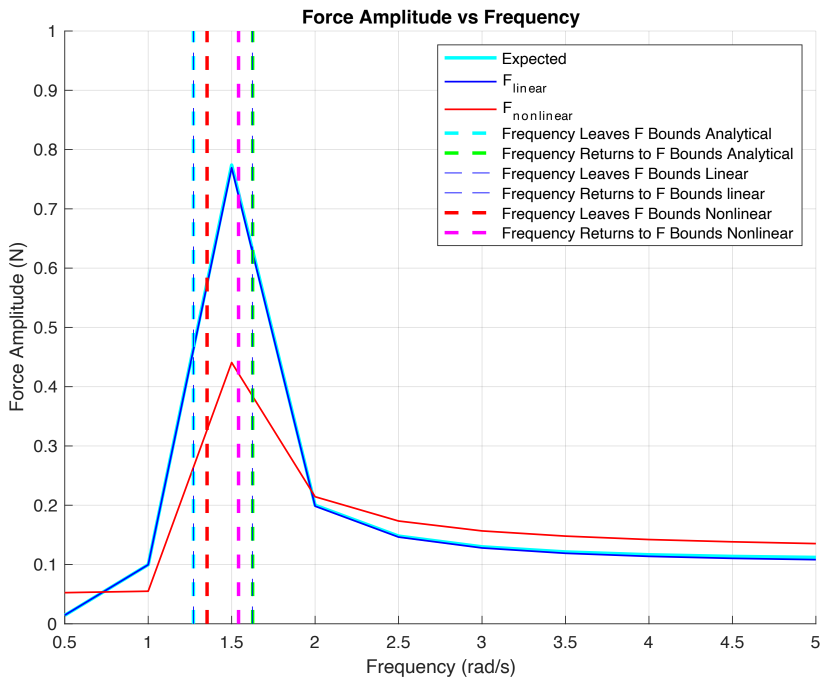
Task 3 Figure A



Task 3 Figure B



Task 4 Figure A (with Extra Credit Tasks 5 & 6)



Task 4 Figure B (with Extra Credit Tasks 5 & 6)

Tables

Table 1: System frequency study summary

System Model	Displacement Operating Range (rad/s)	Force Operating Range (rad/s)	Combined Operating Range (rad/s)	Damped Natural Frequency (rad/s)
Linear	0.5 to 1.0015 and 1.7290 to 5	0.5 to 1.2716 and 1.6241 to 5	0.5 to 1.0015 and 1.7290 to 5	1.4122
Non-linear	0.5 to 1.2288 and 1.6737 to 5	0.5 to 1.3527 and 1.5412 to 5	0.5 to 1.2288 and 1.6737 to 5	1.4002
Analytical	0.5 to 1.0015 and 1.7290 to 5	0.5 to 1.2711 and 1.6241 to 5	0.5 to 1.0015 and 1.7290 to 5	1.4124