

Final Project Report

Team 14: The Embree-yos

May 3, 2023

Team Members:

Raphael Orange, Aaditya Parameswaran, Atticus Rex, Abigail Sun

Client:

Dr. Mark Embree, Virginia Tech Department of Mathematics

Executive Summary:

Team 14, “The Embree-yos”, advised by Dr. Mark Embree, sought to design a hands-on experiment for students in CMDA 3605 courses to see eigenvalues in action. The experiment will involve collecting high-speed video of a swinging pendulum with multiple connected masses, converting the video to time series data, processing the time series data to extract the fundamental frequencies, and finally comparing the experimental values to theoretical ones.

Since most CMDA 3605 classes lack real-life demonstrations of eigenvalue analysis, Dr. Mark Embree wanted this project used as an augmentation to CMDA 3605 discussions on eigenvalues. The strategies the team used to accomplish the problem included defining the problem, visualizing the necessary steps to solve it using a flow chart and Ghant charts, using a decision/scoring matrix to select the best solutions, and discussing solutions among the team members and the client. Using these strategies, the team obtained significant results—the team developed a MATLAB program that pre-processes raw video to isolate colored masses, used k-means clustering to convert the footage into low-dimensional time-series data, and implemented the Fast-Fourier Transform algorithm to find the fundamental frequencies of the system. The team also implemented a theoretical model that coincides with the experimental results. The team was successfully able to construct a pendulum apparatus and develop software that consistently and accurately computes experimental eigenvalues from data that match eigenvalues according to existing theory.

Section 1: Problem Statement

The goal of the project was to design an experiment that allows CMDA 3605 students to perform eigenvalue analysis on a physical system. This involved building a pendulum with multiple connected masses, obtaining a camera that could take slow-motion videos of the pendulum in motion, creating a program to convert the raw footage into time series data for each of the masses, and developing software that can apply the Fast Fourier Transform (FFT) algorithm to extract the fundamental frequencies and compare the data to computed theoretical frequencies. As mentioned above, the current curriculum for CMDA 3605 lacks real-life opportunities for students to do hands-on work with math modeling and data analysis.

The objectives that took the highest priority were constructing the pendulum and developing the image processing and K-Means Clustering software for the videos. Construction took priority because it has historically been the bottleneck of conducting similar experiments. Similarly, because the time-series data is the sole input to the Fast Fourier Transform, it was imperative that the motion tracking and K-Means Clustering implementation were as accurate as possible.

In attempting to implement a theoretical model, the team needed to understand the actual physical laws that produced the underlying dynamics. Thanks to Dr. Embree, the team was able to receive a holistic review of the applicable equations and results, as well as a foundation for the implementation of the algorithm that would output the theoretical frequencies.

Eigenvalue analysis can be applied to several different systems and can illustrate important concepts such as how resonant vibrations can cause damage to a structure. Further, the ability of CMDA 3605 students to experience the entire process of collecting, cleaning, processing, and analyzing data will provide valuable insight into what dynamical systems research looks like in a physical setting.

Section 2: Ethical Considerations

The main ethical concern in this project is that the method of finding the eigenvalues is not the type typically used in CMDA 3605 classes. The eigenvalues in the pendulum model are found by solving a second order system of differential equations, and each eigenvalue is complex. The type of eigenvalue analysis typically taught in beginner classes was mostly in discrete time with a brief emphasis on first-order systems of differential equations. The type of eigenvalue analysis conducted in this project seems to fit more with higher-level mathematical modeling, which may be too far a leap for 3605 students. Hence, the model may not be the best example of real-world eigenvalues, as it lacks a directly obvious connection to the topics being taught. This ethical concern with lack of applicability/validity also overlaps with the potential of misuse of the model or data, since students may misuse this model in class and unknowingly conduct incorrect eigenvalue analysis. This, in turn, could negatively affect their understanding of eigenvalues as the class progresses. To prevent this, the team made sure to write out clear steps on how to conduct this certain experiment, so students will have less of a chance of misusing this model.

Section 3: Literature Review

The most prominent source used in this investigation was the lab manual that Dr. Mark Embree developed while he was at Rice University. This lab manual included a comprehensive overview of

convenient theoretical frameworks to describe the pendulum problem, how to produce the modal mass and stiffness matrices, the Euler-Lagrange equations, how to convert the units of the eigenvalues, and what to expect when running the experiment [1]. This, along with meeting with Dr. Embree, himself, proved to be an invaluable resource throughout the project.

Another significant hurdle in this project was understanding Fourier Analysis, and specifically, implementing the discrete Fast Fourier Transform algorithm on the team's experimental data. Therefore, reading and understanding as much as possible about this algorithm, the Nyquist frequency theorem, sampling rates, complex analysis, convolution, and so on, was also quite key in delivering quality results that would also be interpretable. This was covered in Brigham and Morrow et al. [2].

Lastly, the other significantly challenging component of this project was the object tracking algorithm implemented in MATLAB. While detecting the masses might not be particularly difficult, optimizing the algorithm so that it can be efficiently run on tens of thousands of frames is key to the final deliverable. Hence, the team looked at an article by Nicolescu et al. that gave a comprehensive overview of effective object-tracking algorithms using clustering [3]. This was quite useful in narrowing down options for how the team was going to effectively track the motion of the masses.

Section 4: Project Criteria

The team developed the following Target Specifications to quantify the success of the design and results for the most substantial aspects of the project. Find below the criteria for the pendulum:

- (1) *Dimensions* - The entire apparatus should be able to fit on a tabletop. To achieve this, a target specification of a 2' x 2' x 3' structure was prescribed where 3' denotes the maximum height.
- (2) *Adjustability* - Both the length of the strings and the masses should be adjustable. The masses should range between 25-200g and the strings should range from 2'' to 30'' in length between masses.
- (3) *Initial Conditions* - The design should be able to release the masses from various initial conditions. The design of the frame should provide space for such initial conditions. The team should be able to drop four masses from at least four different initial conditions within 10 milliseconds of each other.
- (4) *Consistent Dynamics* - The design should provide consistent results with as little external disturbance as possible. Given the same initial conditions, the pendulum should produce near-identical dynamics, which can be quantified with the L2-norm of the difference in time series data.
- (5) *Low Damping* - To simplify the complexity of the math, zero damping is assumed. Therefore, the pendulum should have a sufficient lack of friction such that it can swing for at least 30 seconds before returning to the neutral position.

Find below the criteria for data collection and analysis:

- (1) *Large Volume of Data* - When the masses of a pendulum are released in a particular manner, the frequency with which they move can be significantly large. Thus, to ensure that the algorithm can accurately determine the frequency of the system, it is required that the camera is able to capture

a significant volume of data. The criterion for this step is that the camera has a frame rate of 240 FPS.

- (2) *Robust* - The camera must be able to recreate the experiment multiple times, and possibly even be used in the future for CMDA 3605 classes. While there is no easy way to quantify this criteria, the will take it into consideration when reviewing various camera options.
- (3) *Conversion of Raw Data to a Time Series* - After the raw data has been collected, the needs a way to convert the video to time series data that can then be inputted into the mathematical model. The target for this criterion is that the team can take all of the frames from the raw footage and convert them with a 1:1 ratio.
 - (a) *Ability to Determine Distance from Pendulum Vertical* - A sub criterion for the conversion of raw data to a time series data is that there should be software that can determine how far the masses are from the pendulum vertical. This will actually be what the time series data is. The target for this step is that for 20 frames with predetermined distances from the vertical, the program should accurately determine the distance with less than 1% error.
 - (b) *Accurate* - Additionally, while a 1:1 ratio is desirable, the program should be able to determine when there is an outlier in the data and remove it from the dataset while informing the user. It is known that a mass cannot move further than its initial displacement so the program should take this into account.

The final step in the experiment is the ability to use the converted time series data to determine the frequency of the system and compare this to the theoretical values of the same system. Thus, the criteria for this step is:

- (1) *Compute Frequencies from Time Series Data*: This is essentially the crux of the project. The team needs to have software that can take time series data as an input, compute the necessary matrices, and output the eigenvalues of the system. While difficult to quantify, the following will provide some numeric way by which the team will be able to determine whether this criterion is accomplished:
 - (a) *Experimental Results Match Theoretical Results*: While the purpose of this experiment is to expose CMDA 3605 students to the entire process from data collection to analysis, it is important that the results be accurate up to a certain degree. The team requires that the frequencies of the experimental and theoretical models match within 1% of each other.

Section 5: Selected Solution

The team ended up selecting an aluminum 8020 frame for the pendulum, braided wire as the string of the pendulum, and center-drilled bronze cylinders as the masses. This was mainly due to ease of construction, budget, and lead time for these parts. Furthermore, the team chose the DJI Action 2 camera for its high 240 Hz frame rate, high resolution, durable body, and generally positive reviews online. The team also decided to integrate a peg board with metal rods into the back of the frame to more easily position the masses in their initial positions. To make the masses stand out against the background, the team decided to paint the pegboard blue and the masses red. This would allow for easier isolation of the pixels that corresponded to the masses from those that correspond to the background.

On the theoretical side, the team largely used the same model from Dr. Embree's lab manual which included some key assumptions. Those were that there was no vertical motion in the masses, and the change in potential energy of the vertical motion was treated as elastic potential energy to preserve the linear dynamics. There was a Taylor series expansion of a radical, and there was no damping or outside force assumed in the model.

The team decided to perform the analysis in MATLAB as this is the accepted programming standard in mathematics and engineering literature. To track the masses, the team used K-Means Clustering with a preprocessing stage that isolated the pixels that matched a certain RGB threshold. To extract the frequencies of vibration, the teams used the Fast Fourier Transform (FFT) as described in the literature review. The theoretical model was implemented as a MATLAB function that could be parameterized by the initial conditions of the pendulum.

Section 6: Results

The first step of the project was the fabrication of the pendulum. The first part of this was to determine what material the frame was to be built of. As described previously, the pendulum was designed with an 8020 aluminum frame designed in Solidworks before construction. The construction is shown below in **Figure 1**.

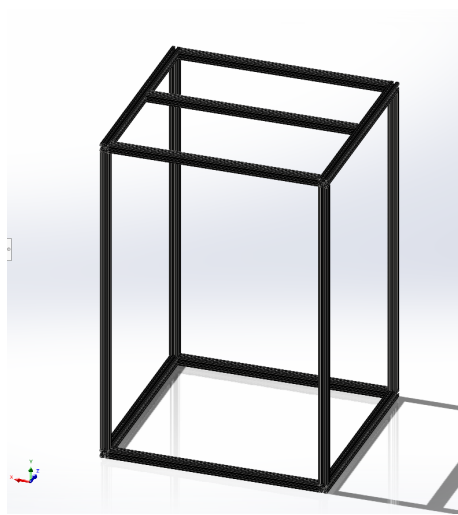


Figure 1: CAD model of pendulum frame.

After this, the team assembled a bill of materials for hardware, masses, string, various tools, and a camera. Once the materials were ordered and the hardware delivered, the team then assembled the 8020 frame, used machinery such as a lathe, tap, and drill press to center-bore the masses, and employed shaft collars with set screws to keep the masses in place on the braided wire. The team did face some difficulty with the stock set screws that came with the shaft collars as they did not protrude far enough into the shaft collar to secure the masses in place. Subsequently, the team had to purchase longer screws with Phillips head attachments for easier tightening and removal. After that, the masses and peg board were painted red and blue, respectively, to easily isolate the masses when processing the raw video footage. The peg board was approximately one foot taller than the back of the pendulum frame and had to be cut down to size with a bandsaw.

Once the frame and masses were drilled, the team had to figure out how to permanently mount the camera. The DJI Action 2 came with a small magnetic mount which the team had to use a hacksaw to cut the bottom off of, drill two holes in this fixture to fit a screw in and stay flush with the camera, and tap some scrap 8020 to make a crossbar. Funnily enough, the team made a cross-bar before figuring out that it had to shoot in portrait mode, not landscape mode, and then had to come up with a quick fix with another piece of scrap 8020 to flip the camera 90 degrees.

Overall, the construction is quite solid—there is no play in any of the joints, the structure is lightweight and robust, the masses are visible, and the set screws hold the masses firmly in place. The camera mount is fully adjustable and the camera is easily removable. **Figure 2** shows the completed pendulum build.



Figure 2: Completed Pendulum Build.

Once the pendulum was constructed, it was time to begin gathering data. The utilized rods that could be inserted into the pegboard to set the starting positions of the masses more accurately than if released with just hands, and because of the proprietary app that pairs with the DJI camera, collecting slow motion video footage and uploading it to MATLAB was seamless.

For this report, the team chose to use video footage of two masses swinging on a pendulum to demonstrate the software used to preprocess the raw footage, perform K-Means clustering to extract the displacement of the masses, apply the Fast Fourier Transform to extract the fundamental frequencies, and finally, output the theoretical frequencies. This was accomplished in one MATLAB interactive live script. The remainder of the results section will detail each pertinent section of the MATLAB Live Script (MLS) and show the respective results.

The first cell in the MLS gives the user the ability to set the starting frame for the video. This can be used to ensure that there are no obstructions (such as hands) to the motion of the masses. As depicted

in **Figure 3**, the user can use the slider and the image of the current frame below it to choose when the video should start.

Turn Slider Until Mass Swinging Begins

```
startIndex = 2421  ;  
pic = vid.read(startIndex);  
figure()  
imshow(pic);
```

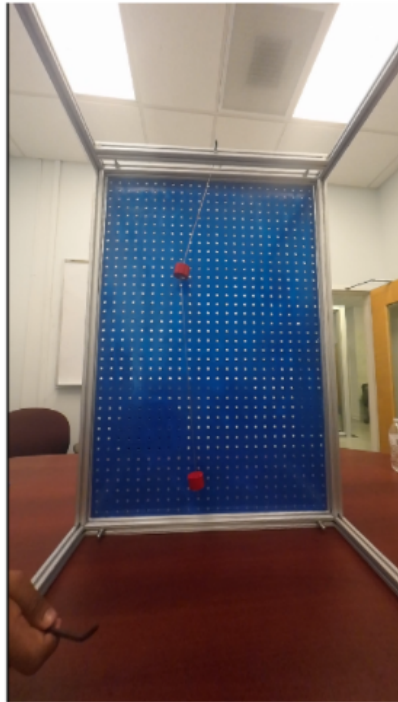


Figure 3: Interactive Trimming Section Cell and Results

The next cell in the script is dedicated to cropping the actual frames of the video so that just the pegboard and the masses are present. This increases the reliability of the clustering algorithm so that no miscellaneous objects in the background are taken into account. This is again accomplished with the use of sliders to adjust the cropping for the left, right, upper, and lower edges of the frame. This section of the script is shown in **Figure 4**.

```

1
% Sliders to adjust the left, right, upper and lower bounds to crop the
% masses
leftBound = 221  ;
rightBound = 881  ;
lowerBound = 1371  ;
upperBound = 551  ;

% Cropping the frame
pic = pic(upperBound:lowerBound, leftBound:rightBound, :);

figure;
imshow(pic);

```



Figure 4: Example video cropped in MATLAB so that just the backing and masses are visible.

Once the video is cropped, the team then employs even more sliders so that the user can adjust the thresholds of each color channel to accurately isolate the masses. Because the team used blue for the pegboard and red for the masses, it was possible to recreate the image as a matrix of 0's and 1's, with 1 being assigned to the pixels that achieved a high red pixel value and a low blue pixel value, and 0 being the rest. **Figure 5** depicts an example of this initial isolation of the masses.

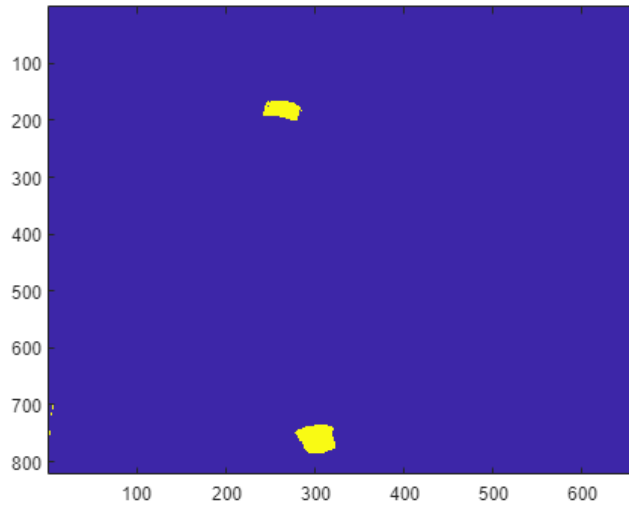


Figure 5: Processed frame of video with the masses isolated.

The aforementioned reduction of the frame to a matrix of 0's and 1's is what is used as input for the K-Means Clustering algorithm. It is important to note that the convergence of K-Means Clustering is crucial for the validity of the results. Hence, the next cell in the MLS runs K-Means Clustering until convergence on the first frame, treating each pixel with a "1" as its unique data point and its row and column index as its location. After the algorithm has converged, a visualization of where the K-Means algorithm places the centroids gets displayed. While the K-Means clustering algorithm usually finds the masses, in rare cases, the clustering may pick up noise and be off-center. This cell provides the user the possibility to run the algorithm multiple times, possibly adjusting some of the RGB and cropping parameters from the above cells, to ensure that the masses have been found. A visualization of the predicted centroids is shown in **Figure 6**.

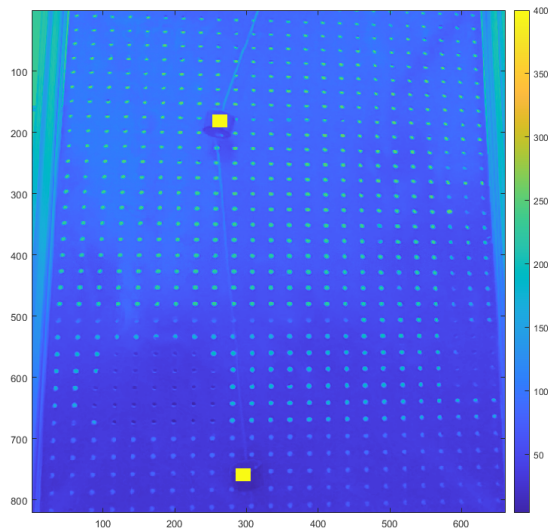


Figure 6: Visual of the K-Means convergence on the first frame to ensure that the following frames are effectively seeded.

Following this step, it is time to run the K-Means Clustering algorithm on all of the frames. While the script has the option to plot the clustering in real-time, which can be useful if there is a significant amount of noise in isolating the masses, it is nearly two orders of magnitude slower. Without plotting, the algorithm, run on a MacBook Air in MATLAB, can consistently and accurately process 3000 frames of slow-motion video in less than a minute. This corresponds to approximately 12.5 seconds of real-time video.

To understand why the improvements that were implemented for the K-Means processing of the video footage worked, it is important to understand how the algorithm itself is implemented. The K-Means algorithm is always seeded with an initial guess for where the user thinks the center of each cluster is. Generally, when run as the default function, the seed might be random values, resulting in the algorithm having to run through more iterations to converge to where the centroids actually are. However, if you seed the function with a guess that is actually quite good, then it will take significantly fewer iterations to converge.

By ensuring that the K-Means Clustering algorithm has indeed converged to masses in the first frame and taking into consideration how little change the pendulum will go through in 1/240th of a second, the output is now a *very* good guess for where the centroids will be in the second frame and means that the algorithm only needs to run one iteration of Lloyd's algorithm before it easily adjusts the mean to account for the change in the masses' motion between frames. The output of this can then be seeded to the next frame and so on, reducing the run time by orders of magnitude. The team has found that this method works quite well for extracting the x-locations of each mass. As the algorithm iterates through each of the frames, the x-locations are stored within a matrix organized by mass and subsequently plotted so that the user can make sure the lines are smooth and the K-Means has not done anything rogue. For the example footage used for this report, the following time-series plots are depicted in **Figure 7**.

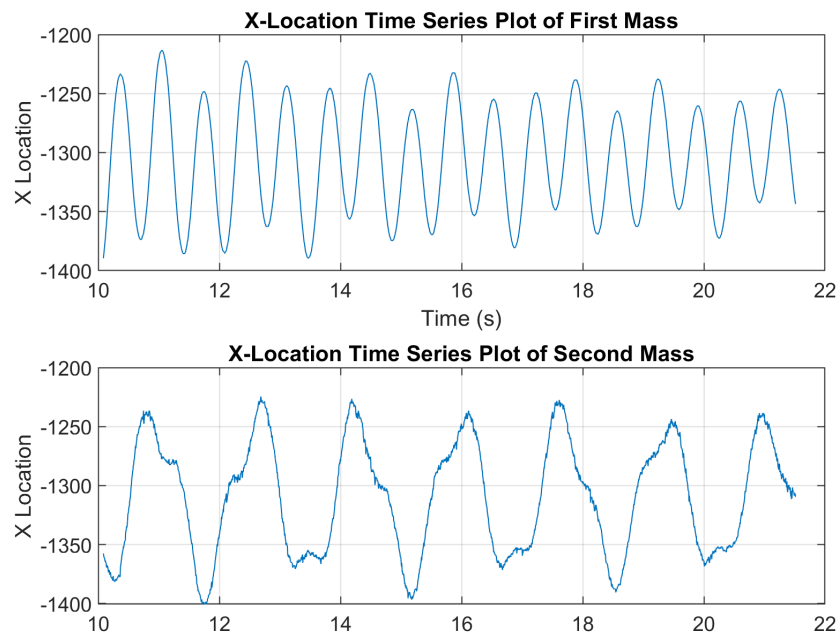


Figure 7: Time-series plots produced from successive K-Means Clustering of the masses.

Usually, the masses closer to the top of the string display more of the higher modes of vibration pristinely. This is because they are under more tension, constrained at both ends, and can sustain higher frequency vibrations for longer before they dampen away. However, the user can examine these plots to see which masses display more diverse vibration characteristics. Once they select which mass to decompose into its fundamental frequencies, it is time for the FFT cell. Using a common variation of the Fast Fourier Transform, the next cell runs the algorithm on the time-series data for the specified mass and outputs the frequency of the system. In this example, the FFT was calculated on the time-series plot of the first mass's movement. This produced the following frequency response plot in **Figure 8**.

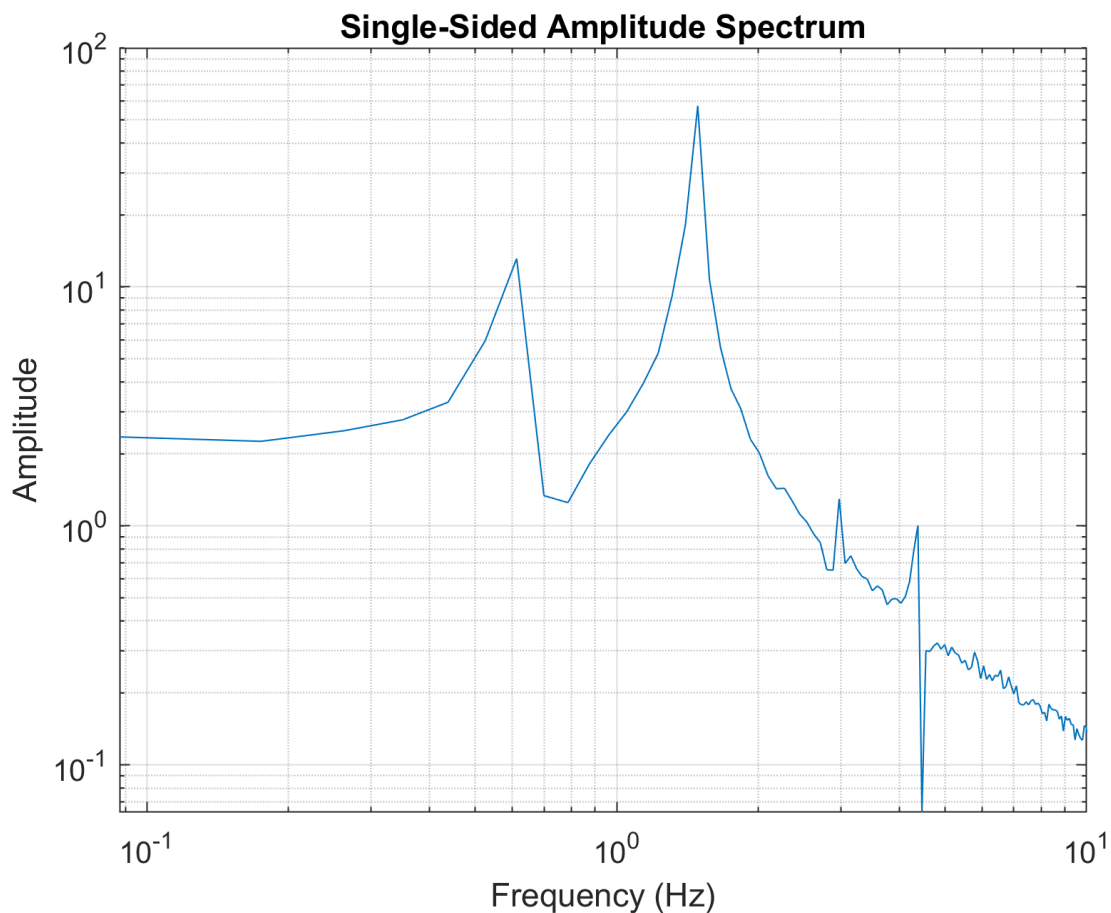


Figure 8: Fourier Transformation of the horizontal motion of the top mass.

As shown in the frequency response plot, there are two distinct peaks, with the second peak being much higher than the first (note the logarithmic scale of the y-axis). This is due to the masses being released at opposite sides of the neutral position, which makes them excite their second mode.

The last cell of the script takes in the masses and lengths between each mass as parameters, calculates the theoretical frequencies according to the Euler-Lagrange equations, and plots these as vertical lines on the frequency response plot for comparison. This is shown in **Figure 9**.

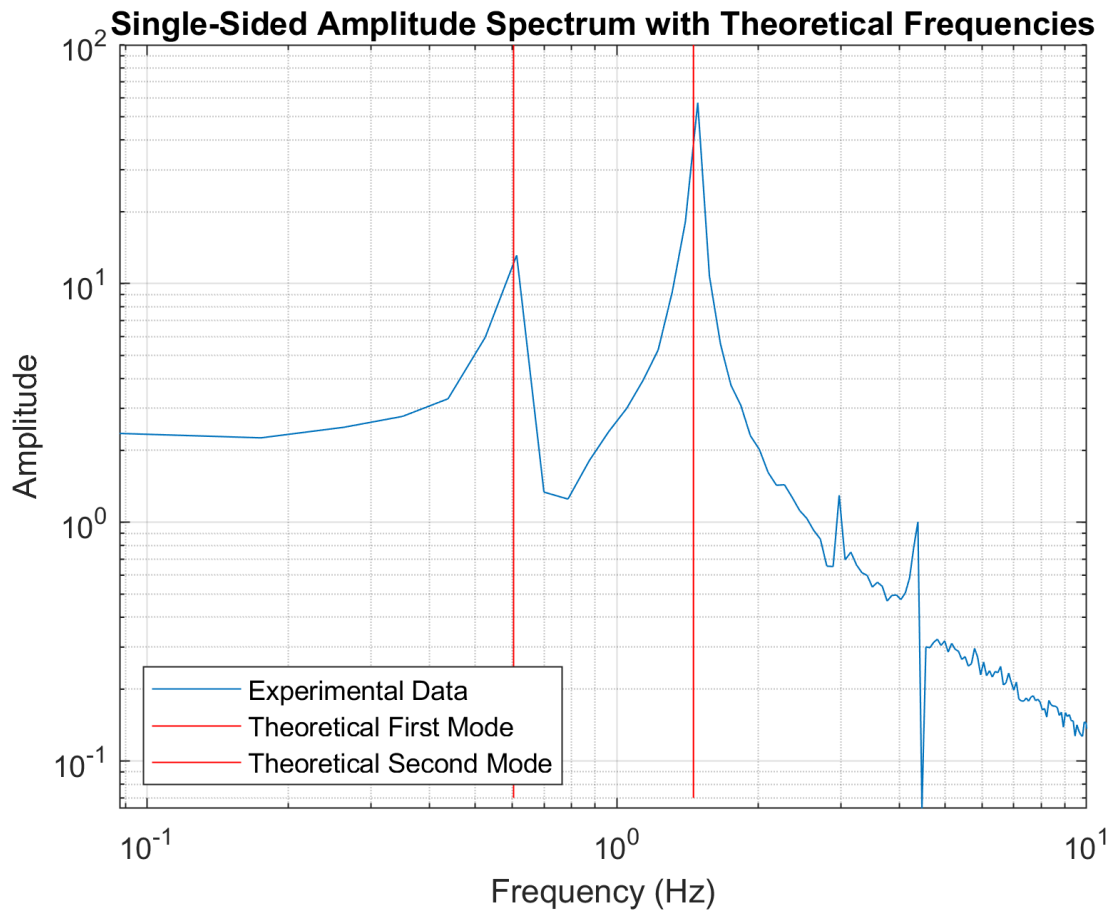


Figure 9: Frequency response plot with theoretical first and second modes indicated.

As shown in **Figure 9**, the theoretical prediction almost exactly matches the spikes in the experimental data, indicating the process is successful.

The theoretical implementation is based on linear algebra, but fundamentally, it originates from the classical physical equation, $F = m \cdot a$. In this case, specifically, the vertical motion of the masses is treated as the compression of a spring with stiffness, K , and the mass and position vectors are matrices because there are multiple masses. Knowing that acceleration is the second derivative of the position vector, the equation to be solved, as a combination of the spring equation and Newton's Second Law is:

$$M \ddot{x} = -Kx$$

Where \ddot{x} is a matrix with the second derivative of the displacement with respect to time (e.g. acceleration) of each of the masses, M is a diagonal mass matrix with each of the actual masses, and K is an equivalent tridiagonal stiffness matrix, which accounts for all of the assumptions stated previously (no

damping, no vertical motion, elastic potential energy instead of gravitational potential, etc.). Isolating \ddot{x} in this system yields the *modal* matrix, Φ , such that:

$$\ddot{x} = \Phi x = -M^{-1}Kx$$

Finally, solving for the imaginary part of the eigenvalues of this modal matrix Φ yields the theoretical natural frequencies. The imaginary part of the eigenvalues will be the square of the true frequencies because the natural unit of eigenvalues is radians per unit of time, so they have to be square-rooted and adjusted to Hz. The MATLAB code for this model creates the mass and stiffness matrices based on the initial conditions and outputs the eigenvalues for the modal matrix.

After having successfully verified experimental data, the team was interested in understanding how different configurations impact the amplitude and location of the peaks in the mode of vibration graph. To illustrate the findings on the effect that the initial conditions have on exciting one mode or another, the team took two videos—one with the masses opposite one another on either side of the neutral axis and one with the masses on the same side of the neutral axis. **Figure 10** depicts the comparison of the frequency responses of the top mass for both of these initial conditions.

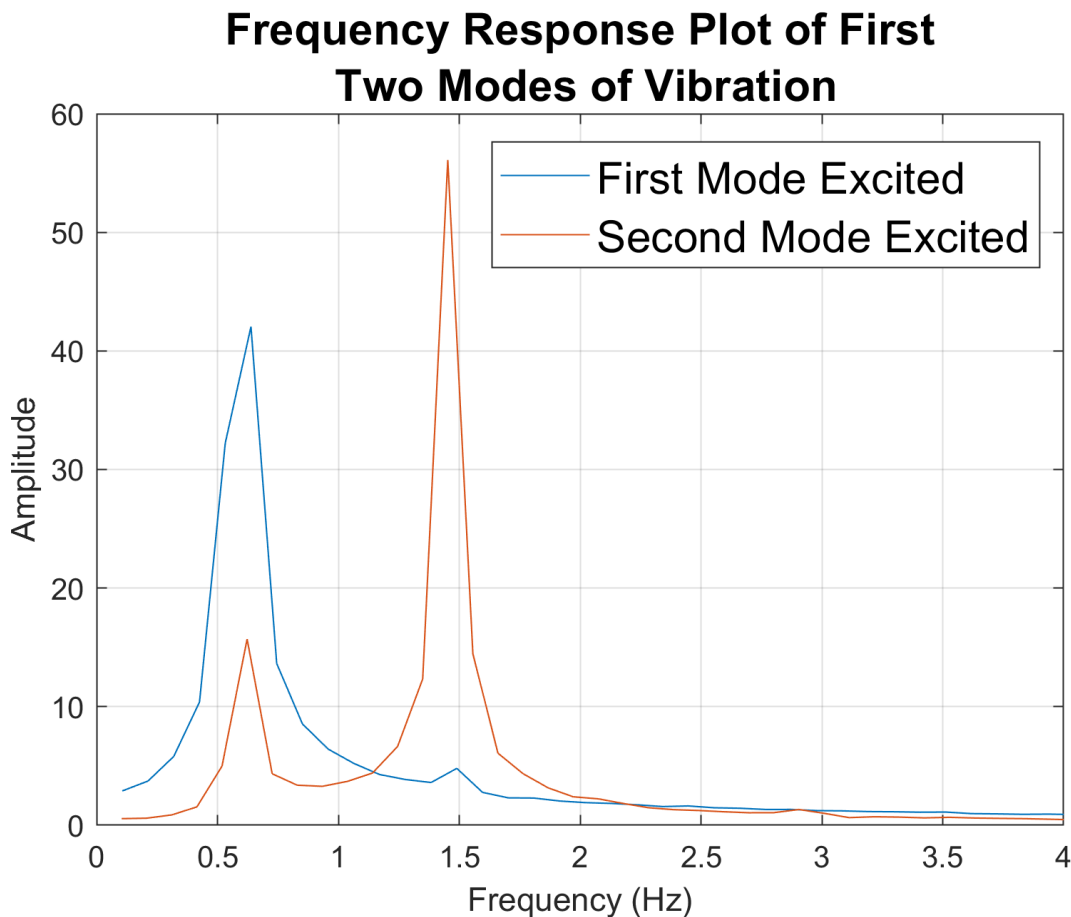


Figure 10: Comparison of First and Second mode frequency response without log scaling

While the location of the natural frequencies does not vary, the extent, or amplitude, to which they are present in the physical motion can change drastically depending on the initial conditions.

Section 7: Limitations on Deliverables

The main assumption the team used was that the pendulum masses were not moving on the y-axis. This was because the pendulum would become a nonlinear system, which is not taught in CMDA 3605 classes. Instead, the masses were treated as fixed on the y-axis, connected by a theoretical bungee cord while oscillating, with the changes in the y-axis being accredited to stretch in the string. Similar to the single mass system, the equations start to fall apart when the initial displacement in the y-axis is too excessive. As one can imagine, if y-axis movement grows, and the team continues to ignore it, eventually the model will be inapplicable to the problem. Thus, it is important that when swinging the pendulum, initial displacements should not be overly large. The system does allow for a good amount of initial displacement, but anything that causes “excessive” y-axis movement should be avoided. As a closing remark, while the pendulum is built of solid aluminum and bolted together, this does not mean that it is indestructible. Caution should be taken in general when initially displacing the system to assure that the pendulum does not break.

As present in the MATLAB live script, there is a bit of discretion that must be taken when choosing the optimal red, green, and blue thresholds to isolate the masses. As evident in the experimental observations, various lighting conditions can manifest themselves as glare in the video footage. This can cause issues such as masses getting split in half, or the masses not being properly isolated from the background. As a built-in safety check, there exists a step in the live script where the user may modify the thresholds, and see what the resulting isolation looks like. They should then, using their discretion, choose the parameters that best highlight the masses, and remove any other disturbances. This will allow the K-Means clustering algorithm to function properly, and thus, give a better estimate of the experimental frequencies.

Finally, similar to the previous section, the K-Means clustering algorithm implementation on the first frame must converge to the proper masses. As implemented in the live script, to speed up the runtime, all K-Means runs on the frames after the first only run one iteration and use the previous K-Means clustering output as its seed. Experimentally, if the first clustering output is correct, then a continuous time series for the displacement of the masses is produced. As a safety check, the team included a section in the live script that outputs the K-Means clustering outcome for the first frame, with the predicted centroids highlighted. It is important that if this is not what is expected, that the parameters be refined and the K-Means clustering algorithm run again. Failure to do so could result in an inaccurate time series which will then return inaccurate theoretical frequencies.

Section 8: Interpretation of Results for Clients

Through verification with a theoretical model and the data collected, the team found the frequencies of the experiment to match, making the project a success. The frequencies do not change much when changing the displacements, but do seem to be affected by the weight of the masses and the number of masses on the pendulum. The pendulum and code will be used to demonstrate real-world applications of eigenvalue analysis for 3605 students, and the results have been delivered to the client.

After presenting these results, the client mentioned how these results could be built upon for next semester's project, such as being able to find a better way to release the masses or how to create exact displacements. Considering the nature of a pendulum with multiple masses, there are many different ways to manipulate the initial displacement of the masses, such as having all the masses oscillate in opposite directions or having them be released from the same side to oscillate uniformly. These changes in the data would give different frequencies and give the operator different types of peaks. There also may be a more efficient way to create the displacement like having exact measurements with pins or creating automatic releases. Finally, there are applications of eigenvalue analysis that involve taking the frequencies and attempting to extract information about the source. Thus, a future project could use the pendulum apparatus and the frequencies of the system to derive a model that can determine the original displacements and masses.

Section 9: Team Roles

Since this is a semester-long project consisting of four different members, it is expected that each of the members equally contributes. With varying skill sets and strengths, each member has made contributions, both technical and non-technical. The list of contributions can be found below:

Raphael Orange:

- *Technical Contribution:* Raphael helped create the code for finding the theoretical frequencies, recorded videos to collect data, and assisted in the construction of the pendulum.
- *Nontechnical Contribution:* Raphael researched the camera, contributed to the Tech Memos, and delivered the demo part of the Tools and Techniques workshop.

Aaditya Parameswaran:

- *Technical Contribution:* Aaditya helped with the construction of the pendulum, developed the initial image processing and k-means clustering code to isolate the masses and output a time series of the displacement of each mass, and aided in refinements.
- *Nontechnical Contribution:* Aaditya researched the masses and shaft collars for the pendulum, helped write the Tech Memos, presented the Midterm Presentation along with Atticus, assisted with setting up and collecting data from the pendulum, and will present part of the Final Presentation.

Atticus Rex:

- *Technical Contribution:* Atticus developed a 3D CAD rendering of the pendulum, constructed the majority of the apparatus, aided in the development of the theoretical model, refined the image processing software, and compiled together all the code for the final MATLAB live script.
- *Nontechnical Contribution:* Atticus helped research the parts for the frame of the pendulum, contributed to the writing of the Tech Memos, presented the Midterm Presentation with Aaditya, assisted in the collection of data from the pendulum, and will present part of the Final Presentation.

Abigail Sun:

- *Technical Contribution:* Abigail painted the masses and backboard for the pendulum apparatus and assisted team members with data collection.
- *Nontechnical Contribution:* Abigail helped research materials for the pendulum, contributed to Tech Memos, worked on and delivered the slides portion of the Tools and Techniques workshop, and will present part of the Final Presentation.

Section 10: Conclusions and Future Work

Throughout the semester, the team has successfully designed and built a pendulum with multiple masses, developed software that can process and extract time-series data from video footage, produced theoretical models that can output the expected frequencies of a pendulum given initial conditions, applied the Fast Fourier Transform algorithm to convert time-series data to fundamental frequencies, and finally, packaged everything together into a comprehensive, and easy to use, MATLAB live script.

While the study of eigenvalue analysis has proven to be useful in understanding the modes of frequency for a swinging pendulum with multiple masses, its applications in frequency analysis are broad. Vibration analysis is essential for ensuring the safety of buildings, vehicles, and aircraft, building robust classification systems, and characterizing the free dynamics of almost any rigid body. These tasks are important, often risk human life, and hence require a great deal of ethical foresight and responsibility.

Others could improve this project by creating a better method for starting displacements, perhaps an automatic style or even a precise manual one. A past experiment had the masses magnetically attached to the wall, and detached with the flip of a switch. More directions include adding viscous or coulomb damping to the model, or even implementing some kind of air resistance simulacrum.

The theoretical model could also be built upon by including and measuring the y-axis during oscillation. While this would make the project non-linear and difficult for 3605 students, this would be a more advanced experiment and could reveal new findings and more data for future capstone students to explore. If the project were to be redone, the team would paint the masses and frame backing with different colors, so they are more visible to larger audiences in larger classrooms. A better color pair would be red for the masses and black or white for the backing, instead of blue. The paint would also be matte instead of gloss to prevent glare.

Section 11: References

- [1] S. Cox, M. Embree, and J. Hokanson, “CAAM 335 · MATRIX ANALYSIS Physical Laboratory,” 2008. Accessed: Feb. 08, 2023. [Online]. Available: <https://personal.math.vt.edu/embree/labman.pdf>
- [2] E. O. Brigham and R. E. Morrow, “The fast fourier transform,” IEEE Spectrum, vol. 4, no. 12, pp. 63–70, 1967.
- [3] A. Ambardekar, M. Nicolescu, and M. Nicolescu, “Object tracking using piecewise feature clustering,” *Computer Science and Engineering*. [Online]. Available: https://www.cse.unr.edu/~ambardek/_includes/FinalManuscript2.pdf. [Accessed: 11-Feb-2023].

Section 12: VT Honor Pledge

“We have neither given nor received unauthorized assistance on this assignment.”

Aaditya Parameswaran, Raphael Orange, Atticus Rex, Abigail Sun