Coursework 1

Data-driven Methods for Engineers

MECH0107
MEng Mechanical Engineering
Date: 6/03/2025
Candidate Number: NBMX0

Contents

I acknowledge the use of ChatGPT 4.0 (OpenAI, https://chat.openai.com). AI tools were used in an assistive role, including proofreading written content, receiving feedback on code, and explaining error messages encountered during development. Additionally, ChatGPT was used to check grammar and spelling to ensure clarity and correctness in the written sections of this report.

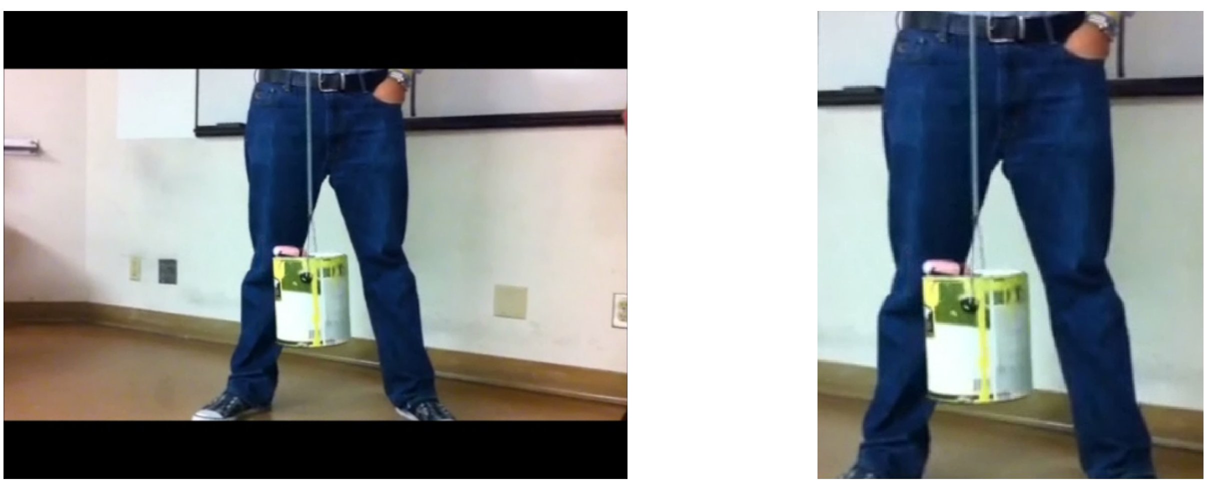
1 Introduction and Problem Statement

The system under study is a spring pendulum (elastic pendulum) which is a hybrid between a simple pendulum, which undergoes rotational motion in the x-z plane due to gravitational forces, and a spring-mass system, which exhibits harmonic oscillation in the z-direction (vertical) due to elastic forces governed by Hooke's Law. Unlike a simple pendulum this system is non-linear, exhibiting coupled oscillations. The main forces at play are elastic restoring force, $F = -kx$, where k is the spring constant and x is the displacement and a gravitational force acting along the z-axis, contributing to pendulum-like oscillations.

The dataset consists of three different camera views (cam1, cam2, and cam3). In order to analyse this problem it will be necessary to post process the dataset to make it suitable for image analysis. Afterwards a tracking algorithm will be implemented in order to track the pendulum within the videos and covert each frame into a set of coordinates. The expected outcomes of this analysis include the identification of the key motion modes, the reduction of the dataset to its most important degrees of freedom, and an understanding of whether the motion can be approximated by a low-rank system. This methodology provides a data-driven approach to modelling the hybrid system while uncovering fundamental physical behaviours.

2 Methodologies

Firstly the datasets, stored in MATLAB's .mat format, are loaded into the python file. Since the video frames are rather large while the area covered by the pendulum is only a small part of them, a cropping function is applied to focus on the region of interest. After each cropping operation a video is generated and reviewed to ensure all pendulum motion is still captured within the new datasets.



(a) Original Frame

(b) Cropped Frame

Figure 1: Comparison of the original and cropped frames

2.1 Image Processing

The next step is to convert every frame to grayscale to remove the colour dependencies within the dataset. Next, histogram equalization is applied to enhance contrast by stretching the intensity distribution, making important features more distinguishable against the background. This technique is particularly useful when dealing with uneven lighting conditions, as it balances the brightness levels across the image. Gaussian blurring further refines the preprocessing by reducing high-frequency noise and smoothing out minor variations that could otherwise introduce errors in object detection. By applying these steps, the tracking algorithm focuses on the dominant motion patterns without being misled by insignificant pixel-level fluctuations or background noise [5].

(a) Grayscale Frame (b) Histogram Equalized Frame (c) Gaussian Blurred Frame

Figure 2: Comparison of the processed frames

2.2 Template Matching

Once the frames are processed, template matching is performed using cv2.matchTemplate(), where a predefined template, extracted from an chosen frame, is searched within each subsequent frame to determine the most likely position of the object. Once the template is detected, the algorithm computes the position of the pendulum as the centre of the detected template [3].

(a) Full Frame (b) Extracted Template Region (c) Tracked Template

Figure 3: Visualization of the template selection process

The two main parameters that need to be optimised for the algorithm to work are the template coordinates as demonstrated in 3 and the match threshold which is the minimum acceptable match score after which the algorithm registers the new position of the pendulum. After initial manual parameter changes an observation was made that the match threshold had to be below 0.5 in order to reliably find the pendulum in the subsequent frames. At the same time this increased the frequency of false matches in regions far away from the pendulum in certain frames. In order to combat that issue a new parameter of max distance moved was added. This parameter specifies the maximum distance the detected pendulum centre can travel between the subsequent frames, minimising the amount of false positives in regions far away from the true pendulum positions.

2.3 Algorithm tuning

The tuning of the algorithm was done manually in the following steps:

1. A random frame is chosen as a template and then the template region is manually positioned using trial and error.
2. The match threshold is kept at 0.65 as a start and the max distance travelled is kept as 100 to effectively disable it
3. A video showing the tracked centre of the pendulum overlayed on the video sample is reviewed to check the algorithm accuracy
4. Match threshold is decreased until the algorithm continuously tracks the can with some amount of false positives
5. Max distance travelled is decreased to decrease the amount of false positives.
6. Tracking video is reviewed periodically to check if the algorithm completes the tracking successfully. If after multiple tweaks the accuracy is not sufficient, a different frame is chosen and the process is repeated.

Once the parameters of the algorithm have been optimised for each of the datasets. A matrix where each row corresponds to the x and y coordinate of the pendulum within each frame was exported. All positions were given in terms of the pixels within the video frames used.

2.4 Data Extraction

Based on the available data a decision was made to analyse the dataset only based on two variables where x is the horizontal position of the pendulum and y is the vertical position of the pendulum. This decision was made for a variety of reasons. Firstly, there is no accurate information as to the angles between the cameras as well as their 3-dimensional positions within the space where the experiment was performed. In addition to that the angles with which the different videos have been recorded and the distance between the cameras and the pendulum varies. This means that the magnitudes in pixels cannot be easily converted into a 3 dimensional Cartesian space without introducing additional errors.

The differences in camera perspective can also introduce parallax error reinforcing the notion that the movements of pixels cannot be scaled to a shared metric space. Lastly, the third video is recorded in a tilted manner meaning that the pendulum does not move vertically in it. It is possible to rotate it to be approximately vertical. However, this would be forcing an assumed solution onto the results which was deemed unfitting for this coursework as its main purpose is to extract insights based on quantitative methods and therefore the video was only rotated by 90 degrees.

2.5 Dimensionality Reduction Methods

Two dimensionality reduction methods were determined to be of interest in the analysis of the dataset:

- Singular Value Decomposition (SVD) – Decomposes the dataset into singular values to assess its rank and determine if the motion can be approximated using a small number of basis functions. This aids in constructing a reduced-order model that captures the system's essential dynamics while filtering out minor variations [4].
- Principal Component (PCA) – Identifies dominant motion patterns by projecting the dataset onto a lower-dimensional space, reducing redundancy while preserving key oscillatory modes. This allows us to determine the primary directions of motion and understand whether the system exhibits vertical, horizontal, or coupled oscillations [4].

2.6 Singular Value Decomposition (SVD)

Singular Value Decomposition (SVD) was selected as the primary dimensionality reduction method for this analysis due to its ability to provide a direct and optimal low-rank approximation of the dataset. Unlike Principal Component Analysis (PCA), which relies on computing the covariance matrix, SVD decomposes the motion data directly, making it numerically stable and well-suited for identifying dominant oscillatory behaviours. This is particularly important for the spring pendulum system, where the motion consists of both harmonic vertical oscillations, governed by the spring force, and pendulum-like rotational motion, influenced by gravity. By extracting the most significant singular modes, SVD enables the construction of a reduced-order model that retains only the essential degrees of freedom governing the system's non-linear dynamics [6]. This approach ensures that the dominant motion patterns are preserved while minimizing the influence of noise and minor variations, resulting in a compact yet accurate representation of the system.

Given three datasets capturing the (x, y) positions of a tracked object across n frames, we construct a data matrix $X \in \mathbb{R}^{6\times n}$, where each row corresponds to a spatial coordinate from a different camera:

$$X = \begin{bmatrix} x_1(1) & x_1(2) & \dots & x_1(n) \\ y_1(1) & y_1(2) & \dots & y_1(n) \\ x_2(1) & x_2(2) & \dots & x_2(n) \\ y_2(1) & y_2(2) & \dots & y_2(n) \\ x_3(1) & x_3(2) & \dots & x_3(n) \\ y_3(1) & y_3(2) & \dots & y_3(n) \end{bmatrix} \tag{1}$$

Each column represents a snapshot in time, and each row corresponds to a different motion component.

The SVD of the data matrix X is given by:

$$X = U\Sigma V^T \tag{2}$$

where: - $U \in \mathbb{R}^{6\times 6}$ contains the left singular vectors, representing dominant spatial modes of the motion. - $\Sigma \in \mathbb{R}^{6\times n}$ is a diagonal matrix containing singular values $\sigma_1, \sigma_2, ..., \sigma_{\min(6,n)}$ in descending order, indicating the strength of each mode. - $V \in \mathbb{R}^{n \times n}$ contains the right singular vectors.

Since the singular values are ordered $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_{\min(6,n)} \geq 0$, the rank of X is determined by the number of nonzero singular values. A low-rank approximation of X can be obtained by selecting only the first k dominant singular values, forming the reduced decomposition:

$$X_k = U_k \Sigma_k V_k^T \tag{3}$$

where U_k, Σ_k, and V_k contain only the first k singular vectors and singular values. This approximation retains the most significant motion patterns while filtering out noise and redundant information.

To compute the SVD efficiently, we consider the self-consistent eigenvalue problems:

$$X^T X V = V \Sigma^2 \tag{4}$$

$$X X^T U = U \Sigma^2 \tag{5}$$

Solving these equations provides the eigenvectors U and V, along with singular values in Σ, which quantify the importance of each mode.

The first few singular values capture the most significant variations in the motion data, enabling a compact representation of the oscillatory behavior. By analyzing the first few singular vectors, we can determine whether the motion is primarily vertical, horizontal, or coupled. This decomposition facilitates the construction of a reduced-order model that effectively describes the system's fundamental dynamics.

3 Results

Once the tracking algorithm is optimised and executed we can show the plots of the positioning of the pendulum at different video frames as evident in Figure (4)

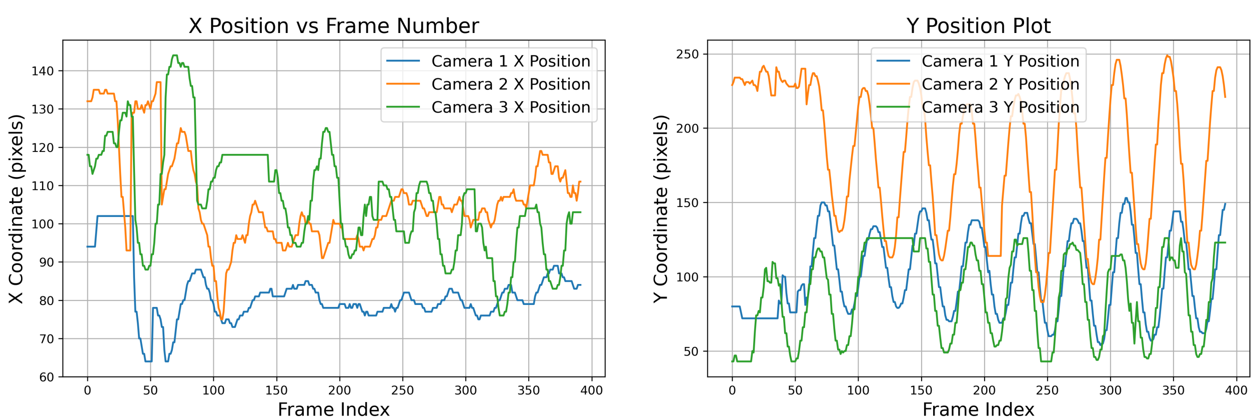


Figure 4: X and Y Position Plots for all cameras

Now, the SVD algorithm can be applied to the dataset. Where each of the modes and their contributions can be seen in Figure (5).

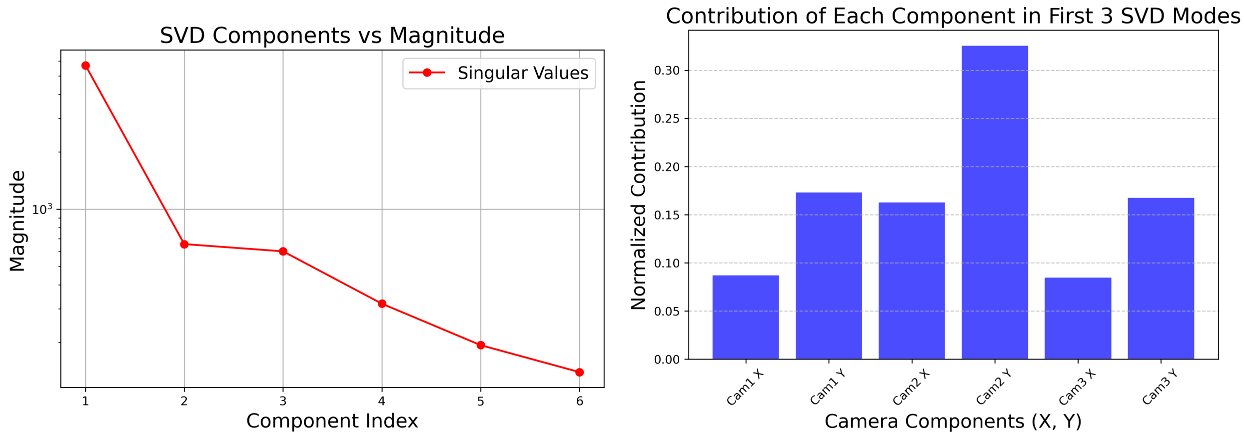


Figure 5: SVD components and their magnitudes

From Figure (5) it can be seen that the first 3 modes contribute to the bulk of the motion in the system. This can be taken a step further in order to analyse the contributions from specific cameras and axes data towards the first 3 modes. Based on this we can see that out of the 4 largest contributors, 3 of them come from the vertical movement (y direction). This shows that the most important modes of the movement occur in the vertical direction. The created modes were then used to reconstruct the datasets and compare them with the original ones.

4 Discussion

A reduced model for each of the cameras can be created. Attempts were made to normalise the dataset because the different angles affect the recorded displacement of the pendulum. However, this has affected the accuracy of the model therefore a decision was made to keep the data as is and make use of the flexible nature of the SVD method in terms of the dataset magnitudes. The reconstructed datasets with the use of different mode numbers, can be seen in Figure (6).

In the left column we can see the X positions recorded by all 3 cameras reconstructed with the different amount of singular values represented by k. The specific number of modes plotted was chosen in a manner to show the lowest number that approximates the dataset reasonably well and then show the differences between that and reconstructions with singular values. It can be observed that the black line in the X position datasets is quite irregular and even using when using 4 singular values, the reconstruction does not reflect the original dataset. This is most likely caused by the fact that the X direction is the one with the least regularity, the movements in that direction are also substantially lower in magnitude which means that any noise due to tracking inaccuracies affects them to a greater extent. On the other hand, the right column which represents the Y position plots shows something different. In all cases 4 modes are enough to reconstruct the motion with a good decree of accuracy, in case of Camera 2, 3 modes are enough to reconstruct the motion almost entirely with very minor deviations.

What is more important is that in all cases 1 mode is enough o correctly reconstruct the period of the vertical oscillations. This points towards the conclusion that the first mode (singular value) is compromised mainly from vertical motion showing that in the recorded datasets, the vertical motion dominates the physical

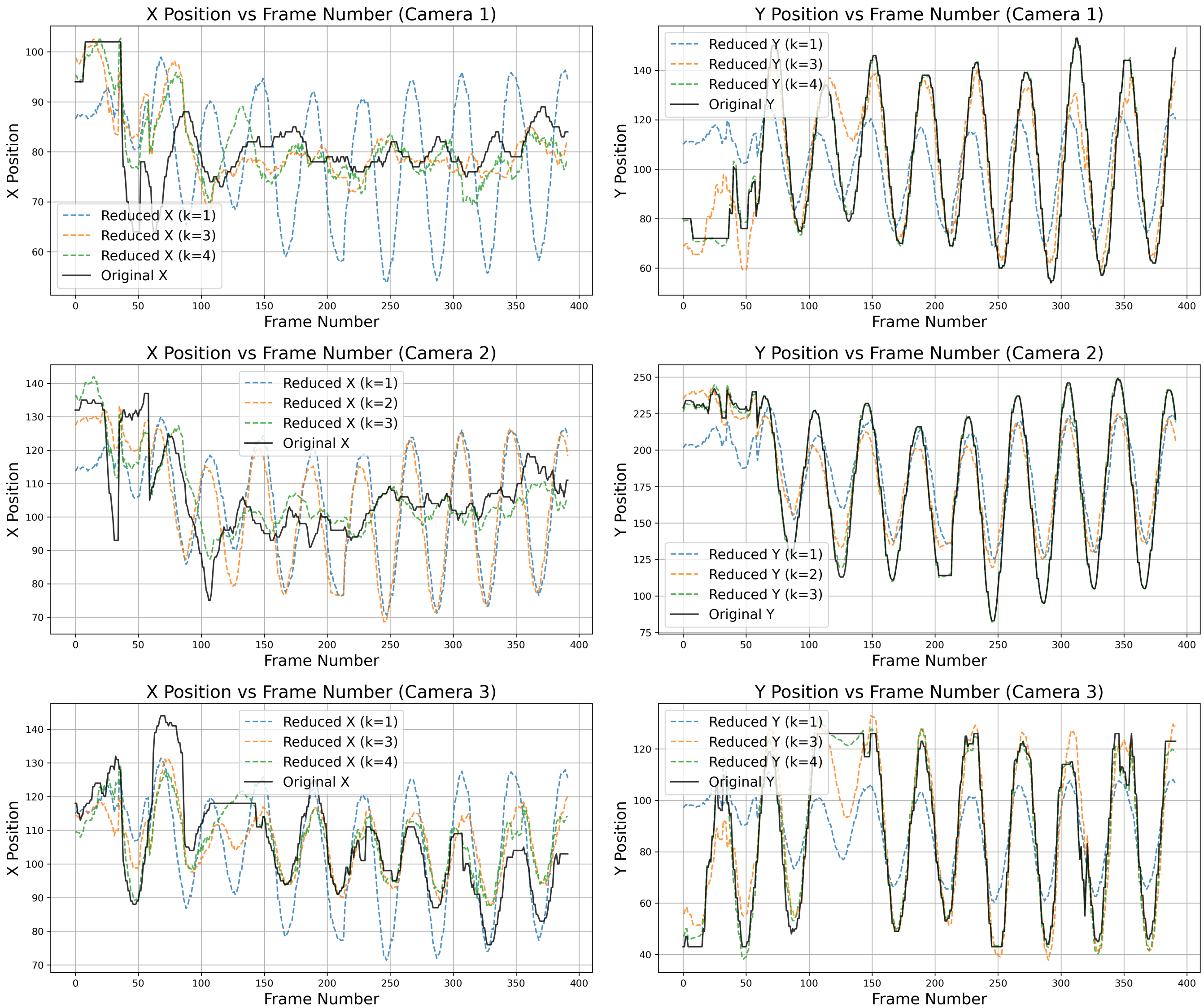


Figure 6: Comparison of X and Y positions vs frame number for Cameras 1, 2, and 3 with different k values.

system. On the other it is important to keep in mind that the pendulum in the videos are predominantly initially displaced in the vertical direction. The horizontal motion seems to be unintentional and its very possible that if the pendulums were initially also displaced sideways, the X position graph would also show an oscillating graph that can be reconstructed fully with 2-3 singular values.

Last thing to focus on is the X position graph for the camera 3. On the first look it seems like it is exhibiting oscillatory behaviour in the x direction, mentioned in the paragraph above. However, a distinction needs to be made as the 3rd dataset is the one where the camera is slightly tilted, hence the oscillation observed here is most likely due to the rotation of the camera introducing the vertical oscillation into what the code defines as the horizontal direction.

In result, this behaviour only reinforces the notion that the vertical oscillation occurring due to Hooke's law and the spring constant of the system is what determines the bulk of the motion in the system. To further reinforce this notion we can revisit the contribution of different cameras towards the modes. Figure (7) shows that the Y movement in the camera 2 and camera 1 are by far the largest contributors to the first

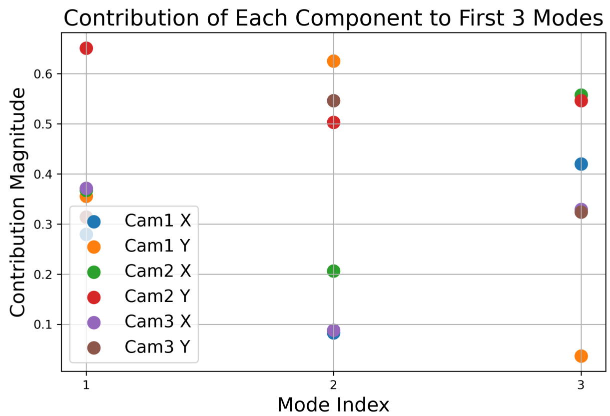


Figure 7: Dataset contribution to the 3 largest modes

two modes of motion with the second one being fully dominated by the 3 Y position datasets. The camera 3 x dataset is a large contributor to the first mode. As mentioned before this is most likely caused by the fact that the vertical movements of the pendulum are partially included in the X position dataset due to the camera being tilted counter-clockwise.

5 Conclusion

In this report, 3 video datasets were used to develop a tracking algorithm in order to extract positional data of a pendulum within them. Afterwards, this data was analysed using graphical methods to establish whether the data was extracted correctly and video material was analysed in order to asses whether the performance of the tracking algorithm was sufficient.

In the next step, Singular Value Decomposition was applied in order to analyse the dominant modes within the system and reconstruct it with a smaller number of singular values to asses how well they describe the dataset as a whole. Based on the conducted analysis it can be determined that the vertical movement (Y direction) dominates the 3 datasets as well as the modes created using the SVD. This makes the vertical oscillatory motion in the datasets the dominant component within the system and it shows that the pendulum motion in this case is a small component of the interactions within.

That being said, the horizontal movement (x direction) within the dataset from the third camera was determined as one of the larger contributors to the 3 main modes after the SVD was performed. Based on the conducted analysis this is most likely caused by the fact that the camera feed in the dataset 3 was not recorded perpendicular or parallel to either of the axis and rotating it by 90 degrees still caused its vertical motion to seep into the horizontal movement, during tracking, causing this horizontal component to appear more important for the main modes of motion then it actually is.

In conclusion, the report has successfully used a template based tracking algorithm in combination with the Singular Value Decomposition data-driven method to extract key insights from the movement of a spring pendulum without the necessity of having any previous knowledge of the physics that take place, making this a great example of extracting data-driven insights from complicated systems.

References

[1] Scipy documentation — scipy v1.8.1 manual.

[2] NumPy . Numpy documentation.

[3] OpenCV . Opencv: Opencv modules.

[4] Lama Hamdeh. MECH0107 - Lecture 2: Dimensionality Reduction. University College London, 2024.

[5] Bhavik Jikadara. Image processing using opencv, 2025.

[6] Andrew Lounsbury. Singular value decomposition, 2018.

[7] Matplotlib. Matplotlib: Python plotting — matplotlib 3.1.1 documentation, 2012.