2 Tutorial 2: Dimensionality Reduction

In this Tutorial, you will use the Principal Component Analysis algorithm (PCA) to analyze the 3D data of the dynamics of a simple pendulum captured from three different orientations, as shown in Fig(1).

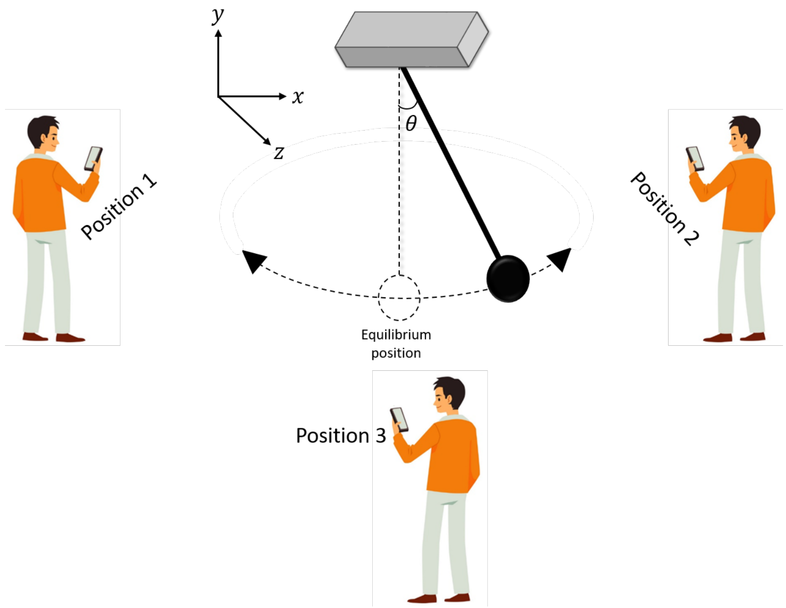


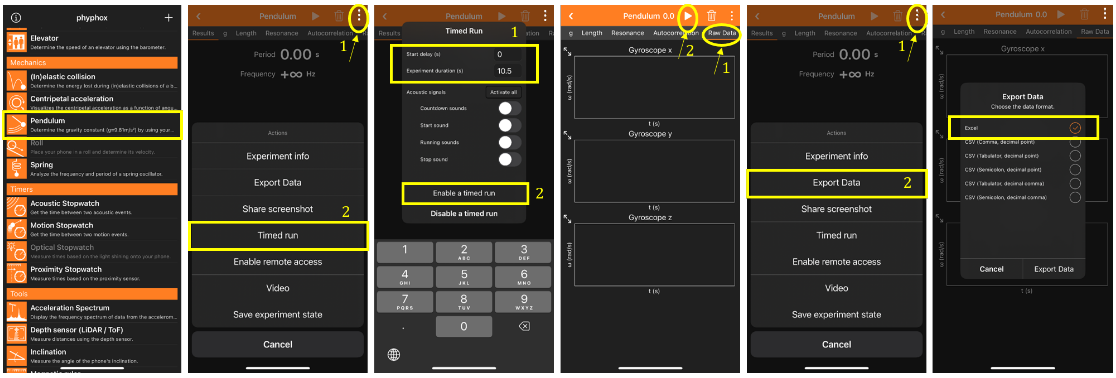
Figure 1: The Pendulum Experiment.

To do that, follow the next steps in order.

1. **Download PhyPhox Software.** There will be no need for this tutorial to have hardware equipment to do the experiment. A simple software on your phone will be enough to capture the necessary data. Phyphox (`https://phyphox.org/`) is a mobile lab that allows you to use the sensors in your phone to do your experiments. For example, detecting the dynamics of a pendulum using the accelerometer. PhyPhox is available for free download on Android and iOS. You can directly install it from Google Play (Android) or the App Store (iOS), and it will only take a couple of minutes.

2. **Experiment Setup**. Once PhyPhox is downloaded to your phone, please open it to start setting up the experiment. The software's main page has several sections containing different experiments, e.g., mechanics, timers, tools, etc. Scroll down to the "Mechanics" section and choose "Pendulum", as shown in Fig(2, a). Press on the three dots on the top right corner of the screen and then choose "Timed run", as shown in Fig(2, b) so that you do not have to

worry about the experiment timing and so that the size of all of your datasets will be the same. As shown in Fig(2, c), enter the start delay value as 0 sec and the experiment duration as 10.5 sec. Once this is done, press the "Enable a timed run". To see the 3D dynamics in action, move to the "Raw Data" tab, as shown in Fig(2, d), which shows you the dynamics of the gyroscope on three axes while the experiment is running.



(a) (b) (c) (d) (e) (f)

Figure 2: Pendulum experiment steps using PhyPhox.

3. **Collect Data** At this point, you are ready to record your data. Before pressing the start button (the right-headed triangle on the top right of the screen), you need to determine the right and left reference points to oscillate your phone between them. Once this is determined, hit start and oscillate your phone right and left until the experiment is stopped.

4. **Export Data** To export the data, press the three dots on the top right of your screen and choose the "Export Data" option, as shown in Fig(2, e). Multiple options will appear to you of which format you would like your data to be exported, as shown in Fig(2, f). For simplicity, choose "Excel" as the exported data format. Having said that, Feel free to choose any other format. If you open the Excel file, you should see that your dataset has four columns, the time, the gyroscope x data, the gyroscope y data, and the gyroscope z data.
NB: The exported Excel file is in its compatibility mode and you should save it as an "Excel worksheet" to be properly imported to Jupyter Notebook later on.

5. **Multiple Readings**. You need to repeat the previous three steps two more times from different orientations so you will end up with three sets of data. For example, if you were facing north in the first experiment, face south and west in the next two experiments.

6. **Analyse Data**. Now that you have the datasets, you can move on to your Jupyter Notebook. Before applying the PCA algorithm, we need to understand the three sets of data and prepare them to be passed on to the algorithm. To do that, the following steps are necessary:

   - Import the three data sets into your Jupyter Notebook.
   - Check the size/shape of each dataset and make sure they have similar sizes, otherwise, you will be unable to do the subsequent calculations.
   - Take a look at the first five rows of each of the three datasets to see the range of features and values you have. You should have four columns (features): Time, x rotational data, y rotational data, and z rotational data.
   - Slice each dataset so that it contains only the rotational data:

$$\mathbf{X}_{\text{EXP}} = \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_n & y_n & z_n \end{bmatrix}$$

   - Visualise the dynamics of each axis for each experiment on a subplot. You should see by eye that the dynamics happen clearly on one axis which refers to the one degree of freedom of the system.

7. **Implement PCA on the Data** Your three datasets are now ready to be passed on to the PCA algorithm and you can start applying its seven main steps.

   - STEP 1: Construct the Data matrix. The first step is to sort out the PCA data matrix so that the rows correspond to the number of trials taken (9 in this case which refers to the rotational data in three axes in three experiments), and the columns refer to the number of readings. to do this, you need to transpose each column of each experiment dataset so that eventually the PCA matrix looks as follows:

$$\mathbf{X}(m = 9 \times n) = \begin{bmatrix} — & \mathbf{x}_a & — \\ — & \mathbf{y}_a & — \\ — & \mathbf{z}_a & — \\ — & \mathbf{x}_b & — \\ — & \mathbf{y}_b & — \\ — & \mathbf{z}_b & — \\ — & \mathbf{x}_c & — \\ — & \mathbf{y}_c & — \\ — & \mathbf{z}_c & — \end{bmatrix}$$

Check the size of the resulting PCA matrix, it should be $9 \times n$. If your matrix has an extra dimension, e.g., $9 \times 1 \times n$, then you need to remove it using the line: `X = X[:,0,:]`.

- STEP 2: Compute the mean of each row, `mean`. Check the size of the mean vector, if you have 9×1, then you can carry on to the next step. Otherwise, if you have (9,), then this would cause a problem and you need to reshape it so it has the correct size using the line:
`mean = np.reshape(mean, (mean.shape[0], 1))`.
- STEP 3: Subtract the mean from the data, $B = X - \texttt{mean}$.
- STEP 4: Construct the covariance matrix C. To do this, you can either calculate the matrix multiplication of $C = B\,B^T$, or you can use the `cov` function from `numpy` library. After each step, you need to check the size of the resulting dataset you just calculated. In this case, make sure the covariance matrix has a size of 9×9.
- STEP 5: Compute the eigenvalues and eigenvectors of the covariance matrix.
- STEP 6: Sort both the eigenvalues and eigenvectors in ascending order and select k of them with the highest variance. Visualize the distribution of the eigenvalues.
- STEP 7: Project the data along the principal components. Visualize the behavior of the first k PCA modes.

8. **Answer the following questions:**

- Starting from Newton's second law of motion $\mathbf{F} = m\mathbf{a}$, derive the differential equation that governs the motion of a simple pendulum and solve it using the small angle approximation. Use Fig(3) for assistance.
- How many degrees of freedom the differential equation gives? And what are the physical quantities that represent them?
- How many degrees of freedom the PCA algorithm can determine for the simple pendulum system?
- Does the number of degrees of freedom match between the analytical and statistical analysis?
- What do you think would happen to the statistical output if noise was added to the system? If you still have time in the tutorial, repeat the experiment by adding noise to the pendulum motion and check the PCA results.

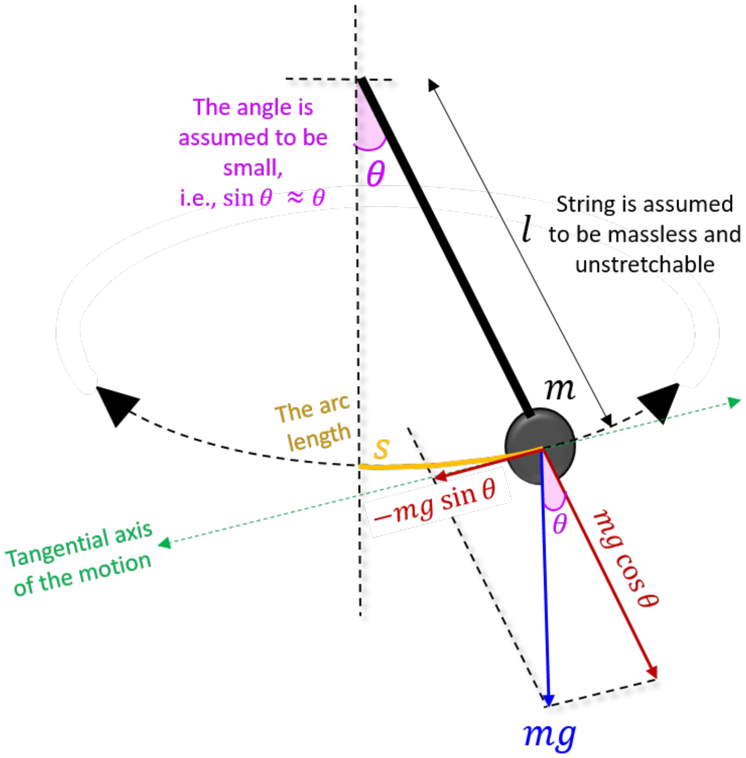


Figure 3: The simple pendulum and the forces acting on it.