Simple Harmonic Motion (SHM) of a mass-spring system

The dataset had recorded time, displacement, velocity, and acceleration.

Also given were: mass of 1kg, spring constant of 1N/m, damping coefficient of 0.1Ns/m, initial displacement 1m, initial velocity 0.

The equation of motion for a harmonic oscillator is given by a 2nd order differential equation:

ma+cv+kx=0

where m is mass

a is acceleration

c is damping coefficient

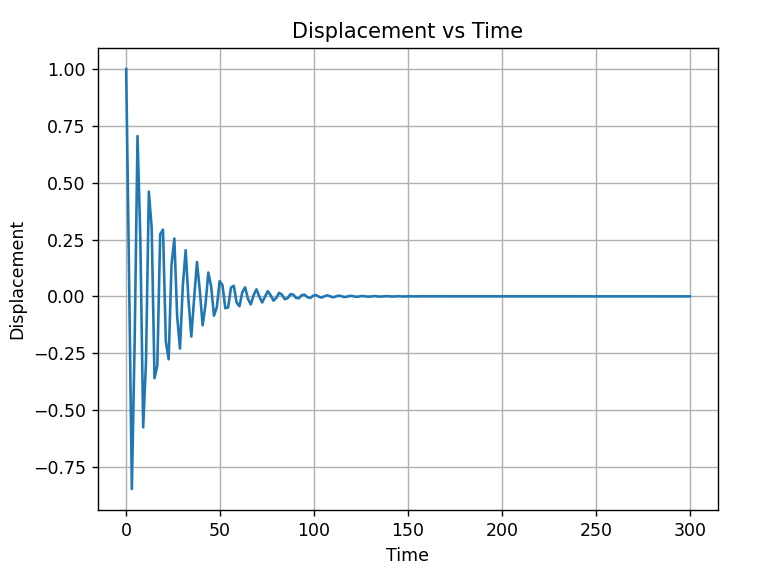
k is spring constant

x is displacement

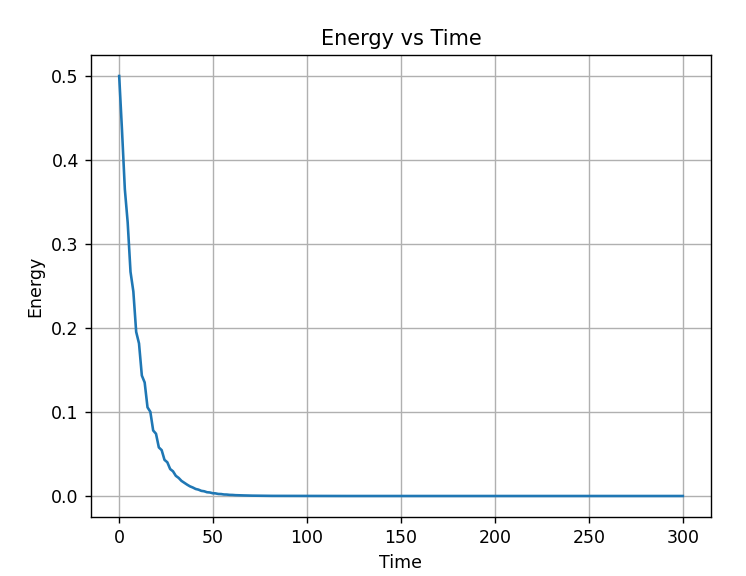
v is velocity

First, I carried out some basic numerical functions and plots.

Read the csv file and stored the values of the 4 variables in arrays.

Plotted time against displacement to visualize the damping effect.

Zero crossings, program counted 95. This number isn’t apparent from the plot, meaning most of the crossings were done throughout the damping process. In the plot, the line appeared horizontal but the mass was still in motion.

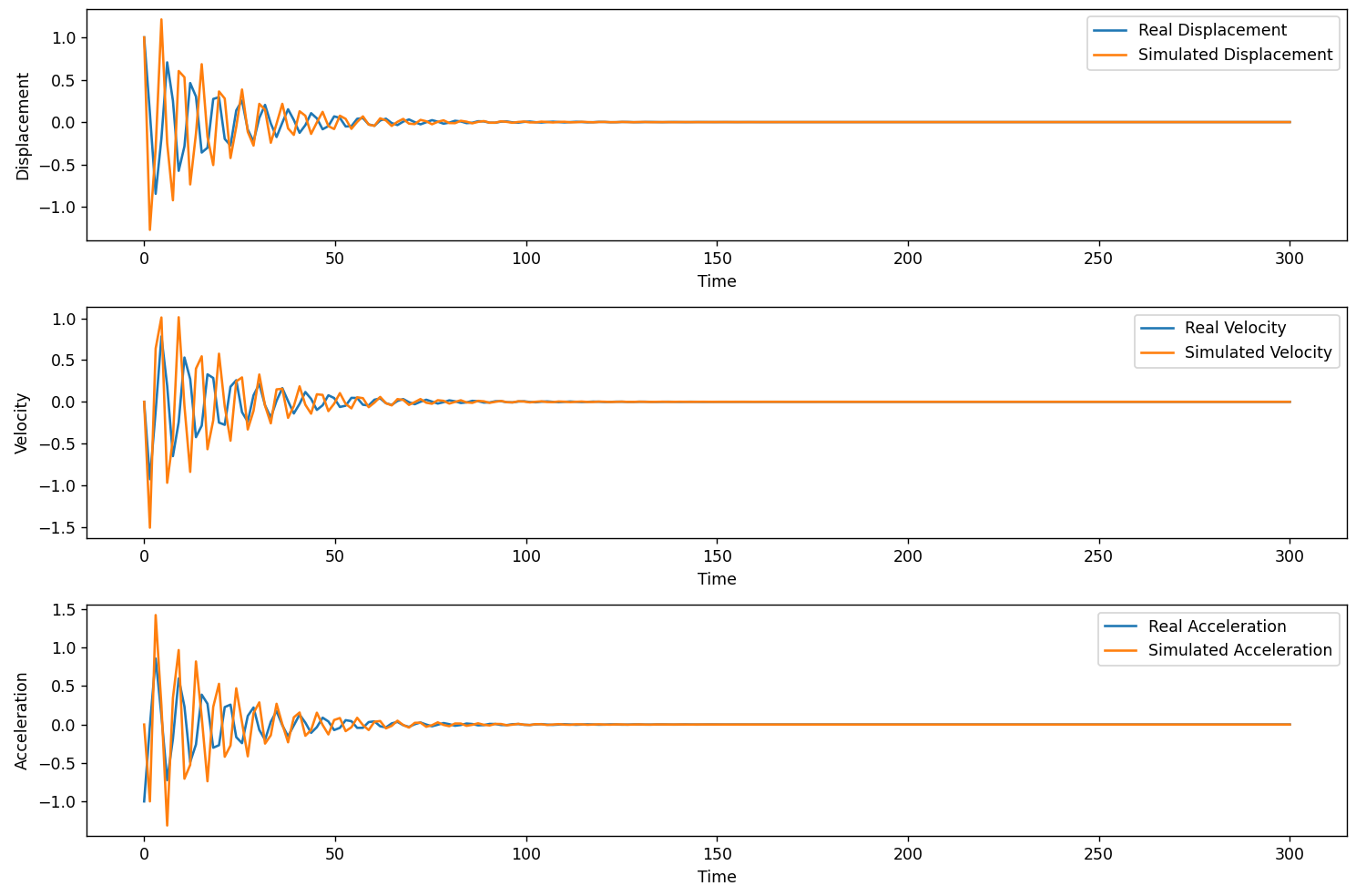
Next, plotted energy against time. For potential energy, the equation used was ½ kx2. For kinetic energy, it was ½ mv2. Then the total energy would be the sum of those two. As expected, the graph showed the energy decreasing exponentially with time.

Verify the dataset by comparing to Euler's method

Euler's method is a numerical technique used to solve ordinary differential equations with a given initial value. It provides an approximate solution by breaking down the problem into small time steps and iteratively calculating the value of the function at each step.

Re-arranged the equation above in terms of acceleration, a= (-cv-kx)/m

Plot the simulated data along with the real data.



The simulation captures the overall damped harmonic motion pattern of the real data. Both exhibit oscillations that gradually decrease in amplitude over time. However, the values are not a perfect match. The simulated data shows higher peaks and less severe damping, especially near the start.

The underlying system might have non-linearities or other complexities not fully captured by the simple harmonic motion model used in the simulation. If the simulation parameters (damping coefficient, natural frequency) are not perfectly estimated, it can lead to deviations from the real data. More sophisticated numerical methods or refined modelling might be necessary for improved accuracy.

Checked accuracy using the sklearn’s Mean Squared Error (MSE) function.

Displacement: 0.0632

Velocity: 0.0481

Acceleration: 0.0450

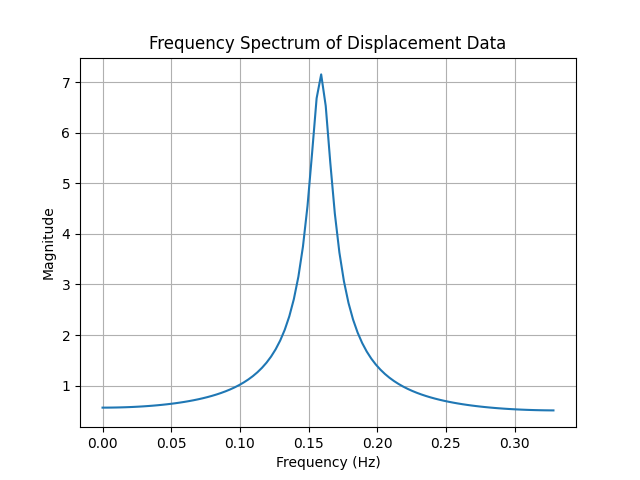
The MSE values are relatively small. This indicates that the squared difference between the simulated and real data points is not very large. In other words, the simulation is reasonably close to the real data.

However, looking at the plot, it is clear the simulation isn’t perfect. Further consideration of the physical system could be useful and other methods of simulation could be investigated.

Signal Processing with Fourier Transformation

To get a frequency spectrum, I applied a Fast Fourier Transform (FFT) to the displacement data with scipy’s fft and fftfreq functions. The Y-axis represents the magnitude of each frequency component in the displace data.

Since the FFT results are symmetric, we only consider the positive frequency components.



It peaks at around 0.16Hz, which corresponds to the dominant frequencies in the displacement data. These frequencies represent the rate at which specific components of the motion are oscillating.

The consistent shape and few peaks indicate the data contained less or no noise.