

Reporte de la Actividad 9

Daniela Olmos Velderrain
Grupo 3

8 de mayo de 2019

1. Introducción

El objetivo de esta actividad fue resolver numéricamente un sistema de ecuaciones diferenciales ordinarias que describe un sistema masa-resorte acoplado, representado en la siguiente figura:

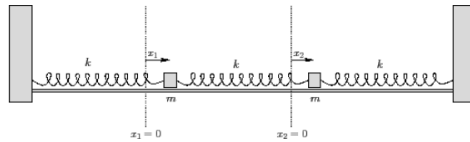


Figura 1: : Sistema masa-resorte con dos grados de libertad.

En el presente reporte se describirá el procedimiento seguido para llegar a la solución del problema planteado, mostrando la teoría detrás del modelo físico, así como las consideraciones para llegar a una ecuación adecuada para trabajar en Python y la gráfica que muestra el resultado obtenido.

2. Desarrollo

2.1. Marco teórico

Un oscilador armónico es aquel sistema que cuando se deja en libertad fuera de su posición de equilibrio, vuelve hacia ella describiendo oscilaciones sinusoidales en torno a dicha posición estable.

Mediante la ley de Hooke, describimos la fuerza ejercida por un resorte sobre una masa como:

$$F = -kx$$

Donde k es la constante elástica del resorte, x es la distancia entre la posición de equilibrio y la masa, la cual denominamos como m .

De acuerdo a la segunda ley de Newton, sabemos que $F = ma = m\ddot{x}$. Reemplazando la fuerza tenemos:

$$m\ddot{x} = -kx$$

Si además tomamos en cuenta la fricción, esta ecuación resulta ser:

$$m\ddot{x} = -kx - b\dot{x}$$

Donde b es el coeficiente de fricción.

El problema presentado en el esquema es un sistema de osciladores acoplados, ya que consta de varios osciladores individuales interconectados entre sí.

2.2. Metodología

Para encontrar la solución numérica del sistema de ecuaciones diferenciales, nos basamos en el algoritmo que aparece en el Cookbook de SciPy. Aquí se apoyan en las bibliotecas *NumPy* y *SciPy* para realizar los cálculos, empleando la función *odeint*, la cual soluciona ecuaciones diferenciales de primer orden.

Sin embargo, el problema abordado aquí es distinto al nuestro, ya que en nuestro caso la configuración de masas y resortes incluye una pared y un resorte extra, como se muestra a continuación:

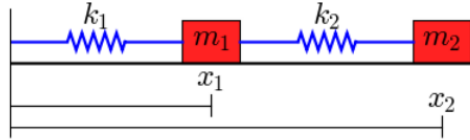


Figura 2: : Sistema masa-resorte solucionado en el Cookbook de SciPy.

Las ecuaciones diferenciales que modelan este sistema son:

$$m_1\ddot{x}_1 + b_1\dot{x}_1 + k_1(x_1 - L_1) - k_2(x_2 - x_1 - L_2) = 0$$

$$m_2\ddot{x}_2 + b_2\dot{x}_2 + k_2(x_2 - x_1 - L_2) = 0$$

Además de las diferencias en el problema físico, el sistema de referencia tomado para las variables también resulta ser distinto, puesto que aquí se incluyen como parámetros las longitudes naturales de los resortes (L_1 y L_2), y las posiciones de las masas x_1 y x_2 se miden a partir de la pared izquierda, mientras que en primer sistema se mide el desplazamiento a partir de la posición de equilibrio para cada masa.

Basándonos en la solución propuesta en las notas de Richard Fitzpatrick, escribimos las ecuaciones de movimiento para las dos masas como:

$$m\ddot{x}_1 = -kx_1 + k(x_2 - x_1)$$

$$m\ddot{x}_2 = -k(x_2 - x_1) + k(x_2)$$

Adaptando estas ecuaciones a nuestro nuevo sistema, tenemos:

$$m_1\ddot{x}_1 = -b_1\dot{x}_1 + k_1(x_1 - L_1) - k_2(x_1 - L_1 + L_2 - x_2) \quad (1)$$

$$m_2\ddot{x}_2 = -b_2\dot{x}_2 - k_3(x_2 - L_2) - k_2(x_2 - L_2 + L_1 - x_1) \quad (2)$$

Este es un par de ecuaciones diferenciales de segundo orden. Para resolverlas mediante la función de *SciPy* debemos convertir esto en un sistema de primer orden, para ello introducimos las siguientes variables:

$$y_1 = \dot{x}_1$$

$$y_2 = \dot{x}_2$$

Reescribimos ahora las ecuaciones de segundo orden, obteniendo un sistema de cuatro ecuaciones de primer orden:

$$\dot{x}_1 = y_1$$

$$\dot{y}_1 = (-b_1y_1 + k_1(x_1 - L_1) - k_2(x_1 - L_1 + L_2 - x_2))/m_1$$

$$\dot{x}_2 = y_2$$

$$\dot{y}_2 = (-b_2y_2 - k_3(x_2 - L_2) - k_2(x_2 - L_2 + L_1 - x_1))/m_2$$

Estas son las ecuaciones que fueron implementadas en Python para ser resueltas mediante la función *odeint*, generándose así un archivo de texto con las posiciones de las masas, las cuales fueron graficadas posteriormente.

2.3. Resultados

Se tomaron las siguientes condiciones iniciales para el sistema:

$$k_1 = k_2 = k_3 = 1$$

$$m_1 = m_2 = 1$$

$$L_1 = L_2 = 1$$

$$b_1 = b_2 = 0$$

$$x_1 = 1, y_1 = x_2 = y_2 = 0$$

Los parámetros para la gráfica fueron `stoptime=30` y `numpoints=750`. El resultado fue el siguiente:

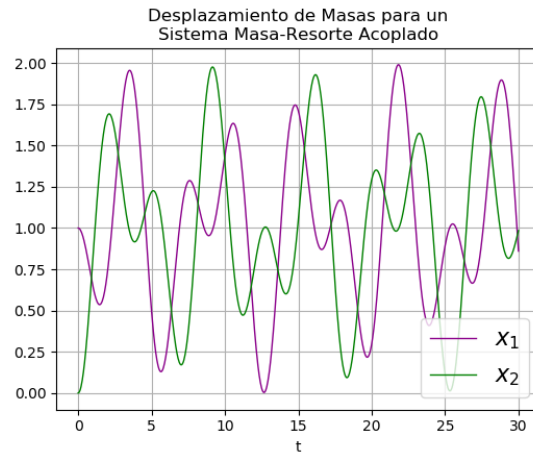


Figura 3: : Sistema masa-resorte acoplado.

3. Conclusiones

Como pudimos apreciar, python puede ser una herramienta muy útil a la hora de modelar diversos sistemas físicos, ya que las librerías que contiene permiten trabajar en problemas matemáticos, como la resolución de ecuaciones diferenciales.

La mayor dificultad que presentó esta actividad fue el encontrar una ecuación adecuada para representar el comportamiento del fenómeno, ya que se debía modelar de acuerdo a un sistema de referencia adecuado para que el programa funcionara de manera correcta.

Bibliografía

- Osciladores acoplados. Recuperado el 7 de mayo de 2019 desde https://es.wikipedia.org/wiki/Oscilador_arm%C3%B3nico
- Osciladores acoplados. Recuperado el 7 de mayo de 2019 desde https://es.wikipedia.org/wiki/Osciladores_acoplados
- Coupled spring-mass system. Recuperado el 7 de mayo de 2019 desde <https://scipy-cookbook.readthedocs.io/items/CoupledSpringMassSystem.html>

- Two Spring-Coupled Masses. Recuperado el 7 de mayo de 2019 desde <https://farside.ph.utexas.edu/teaching/315/Waves/node18.html>