Actividad 5: Movimiento armónico simple: Péndulo

Martin Alejandro Paredes Sosa

Marzo, 2016

1. Introducción

La matemática de un péndulo simple es, en general, compleja. Hacer suposiciones que simplifican la descripción nos permite resolver analíticamente las ecuaciones de movimiento para las oscilaciones de ángulos pequeños.
El péndulo simple, es una idealización se un péndulo real, pero en un sistema aislado donde se asume:

- La cuerda tiene una masa despreciable, es rígida y se mantiene tensa.
- El péndulo se maneja como una masa puntual.
- El movimiento es en dos dimensiones trazando un arco.
- No pierde energía por fricción o resistencia al aire.
- El campo gravitacional es uniforme.
- El soporte no se mueve.

La ecuación diferencial que representa el movimiento de un péndulo simple es:

$$\frac{d^2\theta}{dt^2} + \frac{g}{\ell}\sin\theta = 0 \tag{1}$$

donde g es la aceleración de la gravedad, ℓ es la longitud del péndulo y θ es el angulo de desplazamiento [1].

2. Función *integrate.odeint*

La ecuación diferencial del péndulo no tiene una solución analítica. Para poder resolverla se utilizan herramientas computacionales. En nuestro caso, para resolverla se utiliza la función *integrate.odeint* de la librería *scipy* de Python [2].

Esta función utiliza los siguientes parámetros:

- func: *callable(y, t0, ...)*. Computa la derivada de y en t0.
- y0: *arreglo*. Las condiciones iniciales en y. Puede ser un vector.
- t:*arreglo*. Secuencia de puntos para los cuales se resolverá y. El valor inicial debe ser el primer valor se la secuencia.
- args:*tuple, opcional*. Argumentos extra para la función.

3. Ejercicio y Resultados

Esta actividad consistió en realizar un código en python que nos permitiera resolver la ecuación de movimiento de un péndulo simple. Se hizo uso de la librería *scipy.integrate* haciendo uso de la función *odeint*. Para lograr esto se definieron los parámetros de amortiguamiento y de la ecuación (1). Se corrieron varias simulaciones variando las condiciones del péndulo.

Aquí se tiene el código que se utilizo.

```
import numpy as np
import matplotlib.pyplot as plt
from scipy.integrate import odeint
#Definiendo las cosntantes de la Ec. Diferencial
g = 9.81
l = 1.0
b = 0.0  #Pendulo simple por lo tanto sin friccion
c = g/l
angulo= 0
Vangular= 1.0

#Definicion de la ecuacion diderencial del pendulo
def p (y, t, b, c):
    theta, omega = y
    dy_dt = [omega,-b*omega -c*np.sin(theta)]
    return dy_dt

#Definiendo condiciones iniciales
y0 = [angulo, Vangular]
t = np.linspace(0,10,501) #Para generar la sollucion

#Resolver la Ec. Diferencial
sol = odeint(p, y0, t, args=(b,c))

#Graficacion de la solucion.
plt.plot(t, sol[:, 0], 'b', label='theta(t)')
plt.plot(t, sol[:, 1], 'g', label='omega(t)')
plt.legend(loc='best')
plt.xlabel('Tiempo (s)')
plt.grid()
plt.show()
```

Listing 1: Programa para la solución de la ODE.

Los diferentes escenarios que se obtuvieron fueron los siguientes:

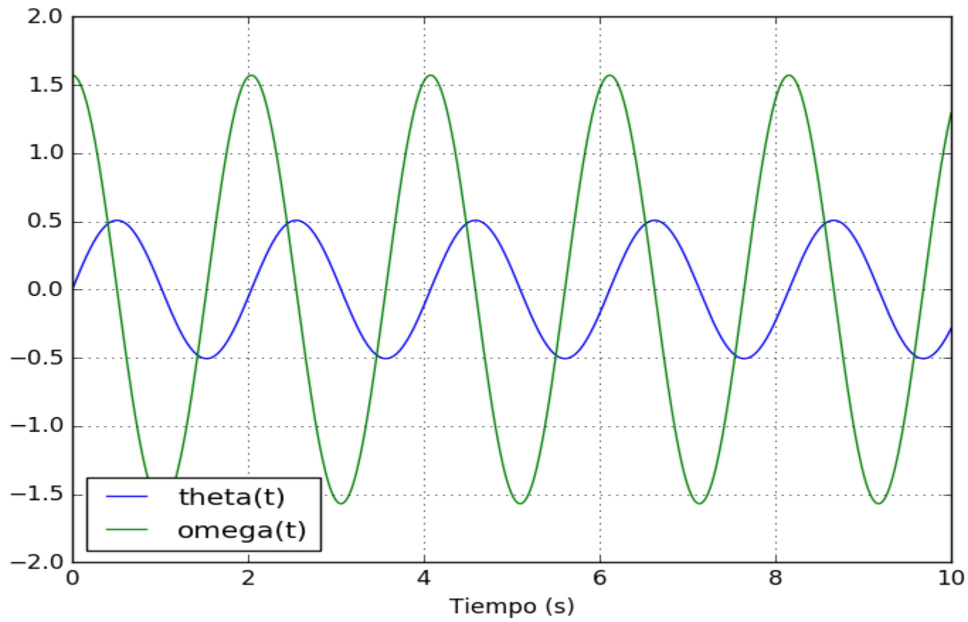

Figura 1: Longitud 1 m en la tierra con $\theta_{=}0$, $\omega_o = \pi/2$ y $b = 0$

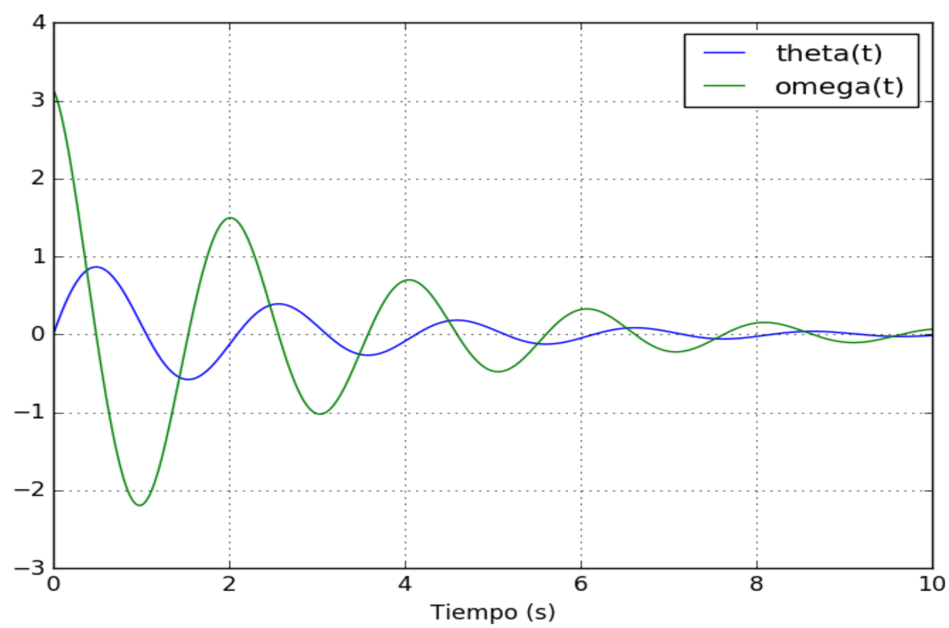

Figura 2: Longitud 1 m en la tierra con $\theta_{=}0$, $\omega_o = \pi$ y $b = 0.75$

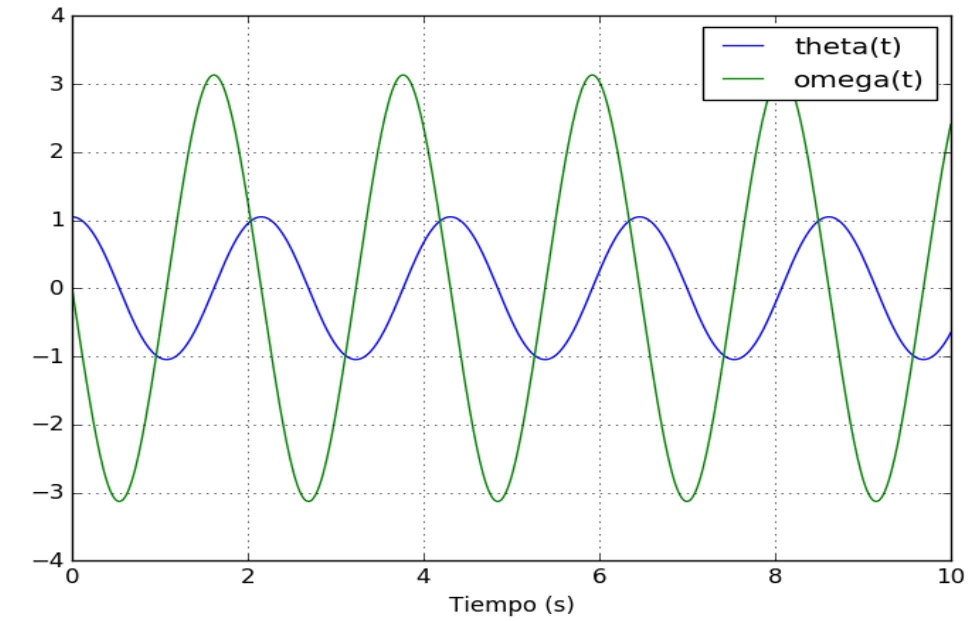

Figura 3: Longitud 1 m en la tierra con $\theta_= \pi/3$, $\omega_o = 0$ y $b = 0$

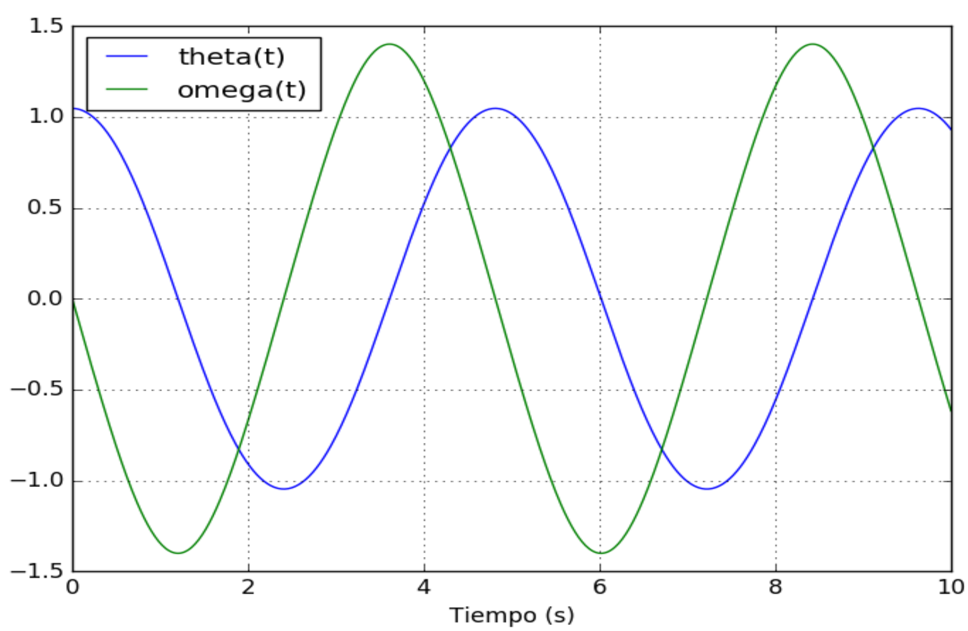

Figura 4: Longitud 5 m en la tierra con $\theta_= \pi/3$, $\omega_o = 0$ y $b = 0$

Los resultados obtenidos son coherentes con las condiciones iniciales.

Referencias

[1] Wikipedia,(2016) *Pendulum (mathematics)*. Recuperado de https://en.wikipedia.org/wiki/Pendulum_%28mathematics%29

[2] Scipy.org (2016) *Integration and ODEs*. Recuperado de http://docs.scipy.org/doc/scipy/reference/generated/scipy.integrate.odeint.html #scipy.integrate.odeint

[3] Lizárraga, C. (2016) *Actividad 5 (2016-1)*. Recuperado de http://computacional1.pbworks.com/w/page/105233358/Actividad%205%20 (2016-1)