

Mapeo de coordenadas del sistema de un péndulo doble acoplado a un péndulo simple

Exposición final.

Proyecto de Física Computación
Facultad de ciencias físico-matemáticas
Jesús Alexis López Luque

30/05/2022

Método Implementado:

Para facilitar la exposición de dicho sistema se explicaran los sub sistemas por separado. Para esto iniciaremos con el Péndulo Simple.

Aprovecharemos la mecánica de Lagrange (cálculo variacional) para el tratamiento de los sistemas por comodidad de las coordenadas gen.

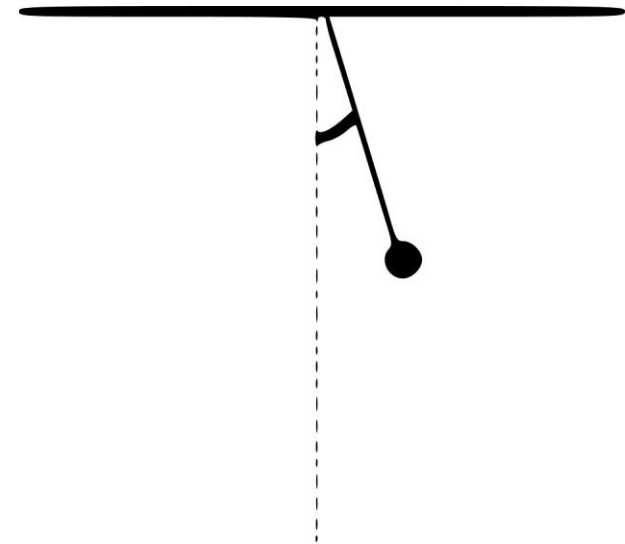
$$\mathcal{L} = T - U = \frac{1}{2}m(\dot{x}^2 + \dot{y}^2) - mgy$$

$$\mathcal{L} = \frac{1}{2}mL^2\dot{\theta}_3^2 + mgL\cos(\theta_3)$$

$$\frac{d}{dt}\left(\frac{\partial}{\partial \dot{\theta}_3}\mathcal{L}\right) - \frac{\partial}{\partial \theta_3}\mathcal{L} = 0$$

$$mL^2\ddot{\theta}_3 + mgl\dot{\theta}_3\sin(\theta_3) = 0$$

$$\ddot{\theta}_3 + \frac{g}{L}\sin(\theta_3) = 0$$



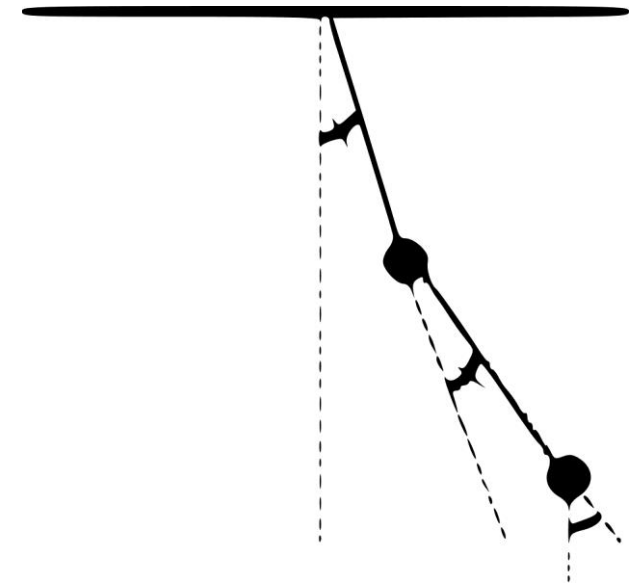
Ahora, este mismo cálculo se realiza para el péndulo doble, el cual ya presenta un poco mayor de dificultad al momento de solucionarlo.

$$T = \frac{1}{2}(m_1 l_1^2 \dot{\theta}_1^2 + m_2(l_1^2 \dot{\theta}_1^2 + l_2^2 \dot{\theta}_2^2 + 2l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \cos(\theta_1 - \theta_2)) \quad (6)$$

$$(m_1 + m_2)l_1^2 \ddot{\theta}_1 + m_2 l_1 l_2 \ddot{\theta}_2 \cos(\theta_1 - \theta_2) - \dot{\theta}_2 \sin(\theta_1 - \theta_2)(\dot{\theta}_1 - \dot{\theta}_2) + m_2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2) + (m_1 + m_2)g l_1 \sin(\theta_1) = 0 \quad (8)$$

$$U = -(m_1 g l_1 \cos(\theta_1) + m_2 g(l_1 \cos(\theta_1) - l_2 \cos(\theta_2))) \quad (7)$$

$$m_2 l_2^2 \ddot{\theta}_2 + m_2 l_2 l_1 \ddot{\theta}_1 \cos(\theta_1 - \theta_2) - \dot{\theta}_1 \sin(\theta_1 - \theta_2)(\dot{\theta}_1 - \dot{\theta}_2) - m_2 l_1 l_2 \dot{\theta}_1 \dot{\theta}_2 \sin(\theta_1 - \theta_2) + m_2 g l_2 \sin(\theta_2) = 0 \quad (9)$$



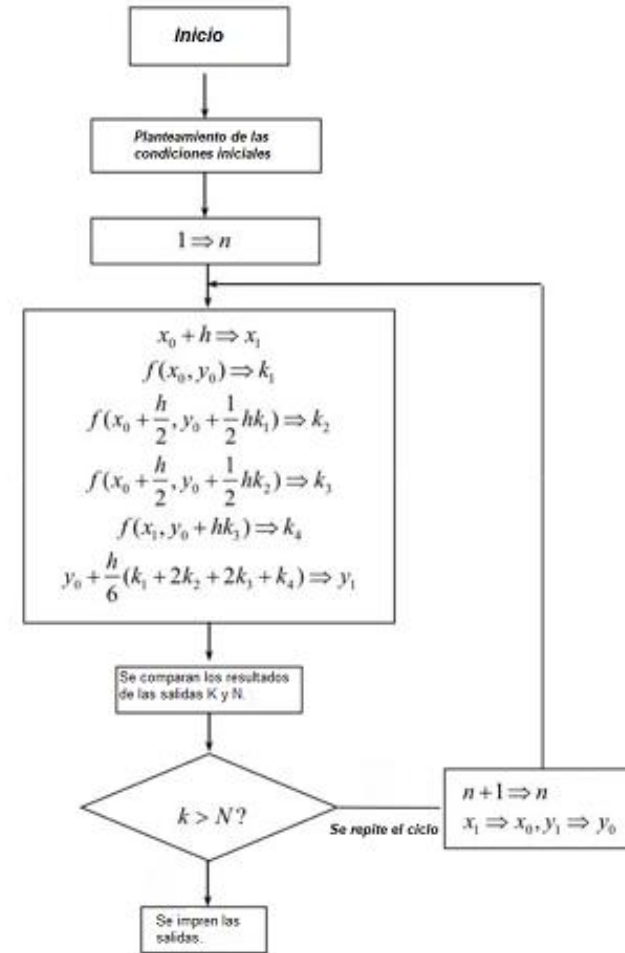
¿Por qué se optó finalmente por este método?

Además de ser un método relativamente sencillo a nivel código, nos da muchas ventajas a la hora de nuestro error, al tener un grado de error de 4to orden nos permite resolver ecuaciones diferenciales de manera exacta o bien, con una muy buena aproximación.

¿Limitantes?

Claro, el poder computación necesario para resolver ecuaciones diferenciales muy complejas presentará un problema, debido al gran numero de iteraciones que podrían necesitarse para el cálculo.

Además, se aprovecha para introducir un segundo método numérico al proyecto.



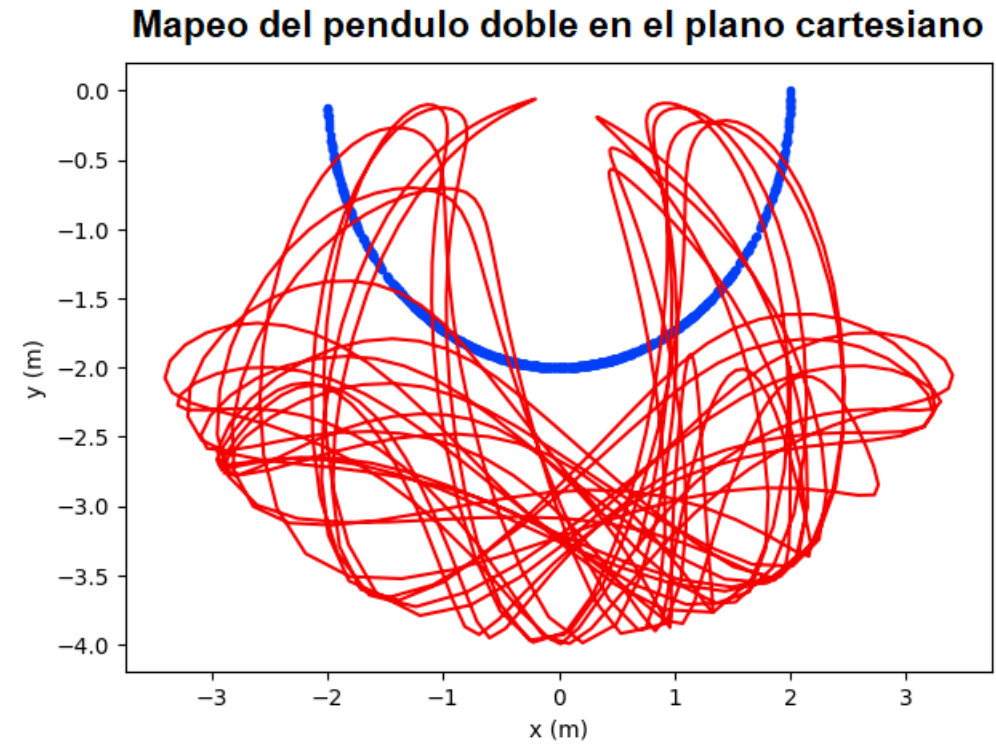
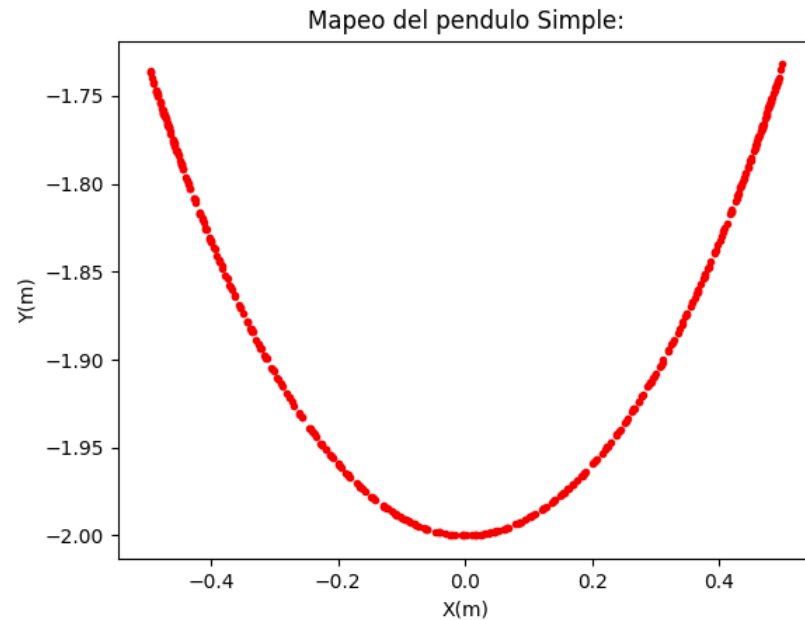
Pero, ¿Cómo solucionamos un sistema de ecuaciones diferenciales de 2do orden? (Caso del péndulo doble).

Para esto es necesario crear un arreglo donde se considere un cambio de variable, este cambio de variable se realiza para reescribir la ecuaciones diferenciales de segundo orden en una de primer orden, las cuales se pueden resolver utilizando integrate.odeint.

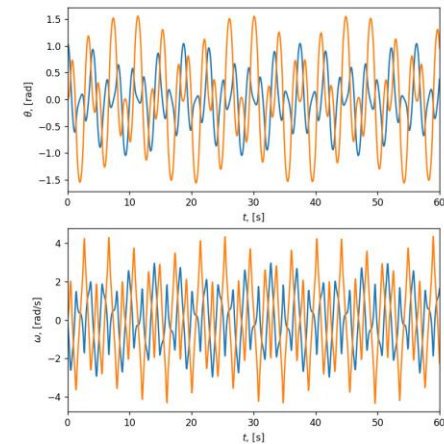
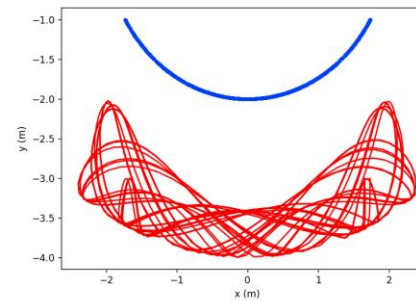
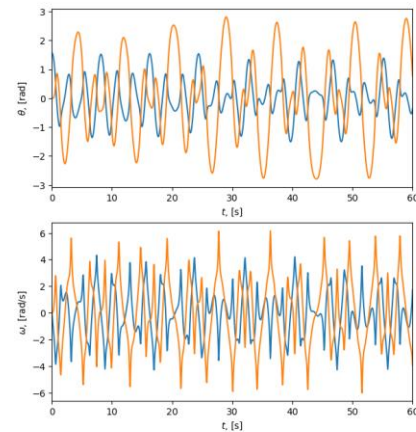
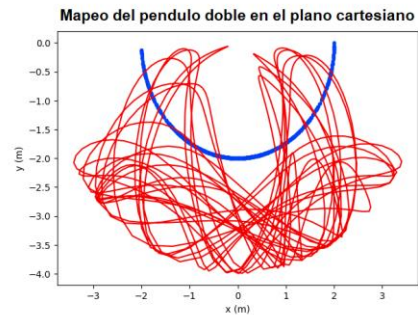
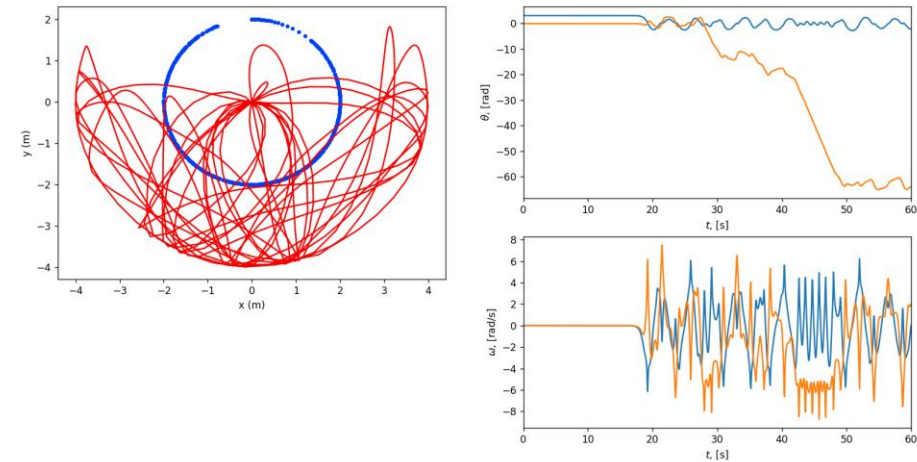
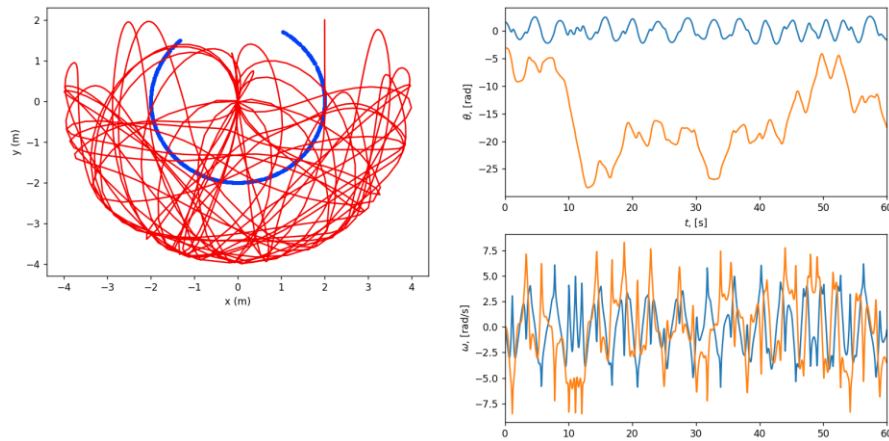
$$\ddot{\theta}_1 = \frac{-g(2m_1 + m_2) \sin \theta_1 - m_2 g \sin(\theta_1 - 2\theta_2) - 2 \sin(\theta_1 - \theta_2) m_2 (\dot{\theta}_2^2 l_2 + \dot{\theta}_1^2 l_1 \cos(\theta_1 - \theta_2))}{l_1 (2m_1 + m_2 - m_2 \cos(2\theta_1 - 2\theta_2))}$$
$$\ddot{\theta}_2 = \frac{2 \sin(\theta_1 - \theta_2) (\dot{\theta}_1^2 l_1 (m_1 + m_2) + g(m_1 + m_2) \cos \theta_1 + \dot{\theta}_2^2 l_2 m_2 \cos(\theta_1 - \theta_2))}{l_2 (2m_1 + m_2 - m_2 \cos(2\theta_1 - 2\theta_2))}$$

```
# Primera ecuacion diferencial
du[0] = u[1]
du[1] = ( m2*g*np.sin(u[2])*c - m2*s*(L1*c*u[1]**2 + L2*u[3]**2) - (m1+m2)*g*np.sin(u[0]) ) / ( L1 *(m1+m2*s**2) )
# Segunda ecuacion diferencial
du[2] = u[3]
du[3] = ((m1+m2)*(L1*u[1]**2*s - g*np.sin(u[2]) + g*np.sin(u[0])*c) + m2*L2*u[3]**2*s*c) / (L2 * (m1 + m2*s**2))
```

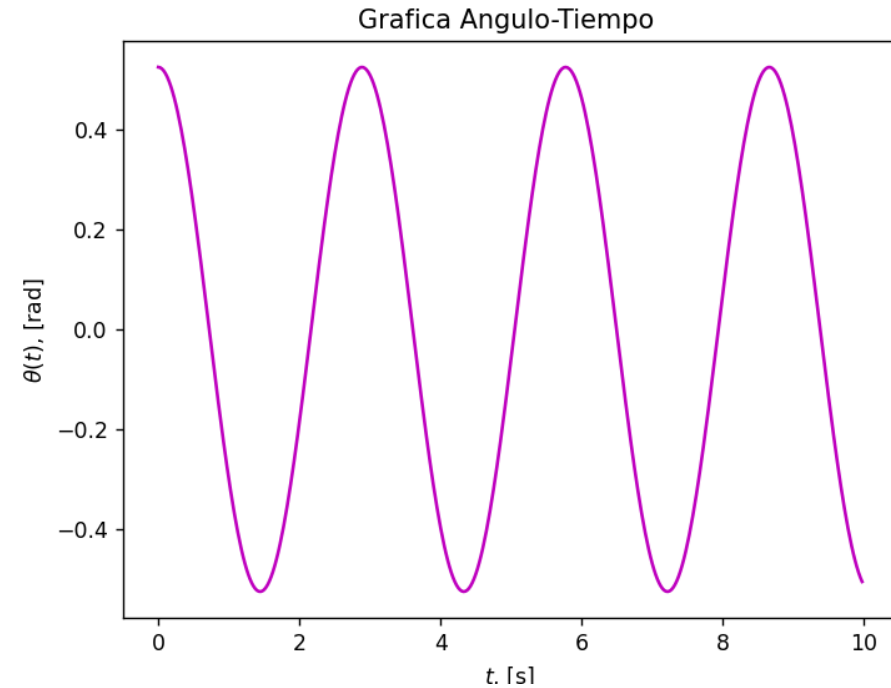
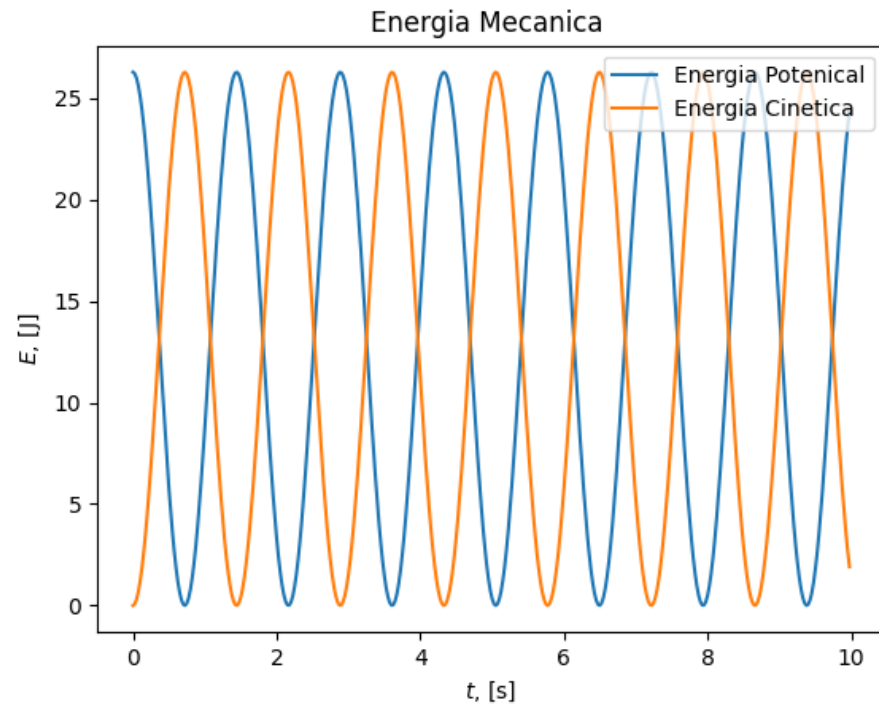
Con lo expuesto con anterioridad se llegó a la soluciones numéricas de las ecuaciones diferenciales de los sistemas, logrando así un mapeo en el plano cartesiano, el cual es el siguiente:



Como curiosidad, podemos observar los siguientes ejemplos del péndulo doble, en el cual nuestras condiciones iniciales cambian radicalmente el comportamiento del sistema.



Se aprovechó al igual que en el péndulo doble y en el péndulo simple, para analizar un poco más su evolución a través del tiempo, por lo cual se generaron las siguientes gráficas:



Mientras que para nuestro sistema completo el sistema de ecuaciones quedó de la siguiente manera:

```
def Pen_Acop(u,t,m1,m2,L1,L2,g, k):          #Definimos el sistema de ecuaciones diferenciales que describen el sistema

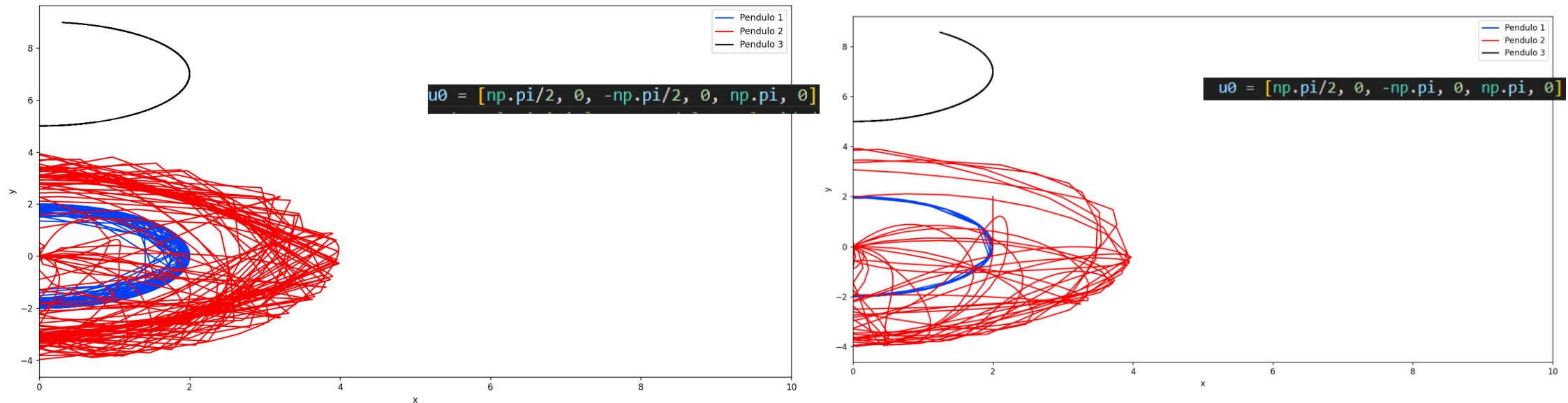
    du = np.zeros(6)
    c = np.cos(u[0]-u[2])  #Coseno de diferencia de Thethas's
    s = np.sin(u[0]-u[2])  #seno de diferencia de Thethas's

    du[0] = u[1]  # d(theta 1)
    du[1] = ( m2*g*np.sin(u[2])*c - m2*s*(L1*c*u[1]**2 + L2*u[3]**2) - (m1+m2)*g*np.sin(u[0]) ) / ( L1 *(m1+m2*s**2) ) +(k/(2*m1*L1))*(np.sin(u[4])-np.sin(u[0])*np.cos(u[0]))
    du[2] = u[3]  # d(theta 2)
    du[3] = ((m1+m2)*(L1*u[1]**2*s - g*np.sin(u[2]) + g*np.sin(u[0])*c) + m2*L2*u[3]**2*s*c) / (L2 * (m1 + m2*s**2))
    du[4] = u[5]  # d(theta 3)
    du[5] = (-g/L1)*np.sin(u[5])-k*L1*(np.sin(u[5])-np.sin(u[1])*np.cos(u[5]))

    return du
```

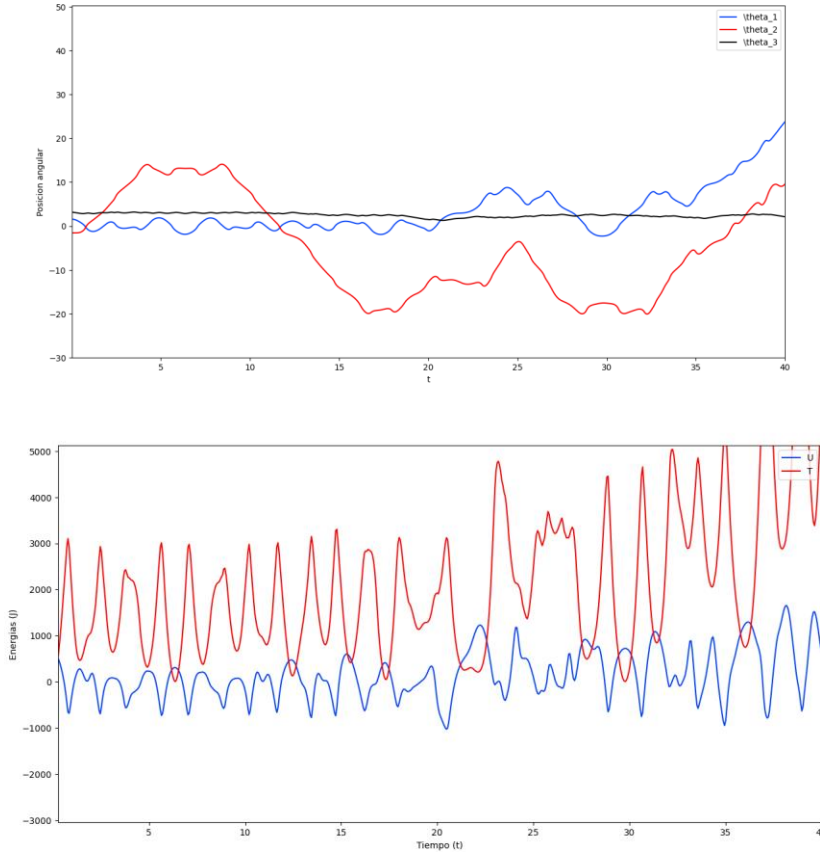
Se procedió a resolver de igual manera que el péndulo doble, utilizando la función `odeint` de `Scipy.integrate`. Obteniendo así los siguientes mapeos (se muestra un par para apreciar el comportamiento caótico del sistema completo).

Se puede apreciar que debido al amortiguamiento presente por el resorte la trayectoria del péndulo simple se ve limitada.

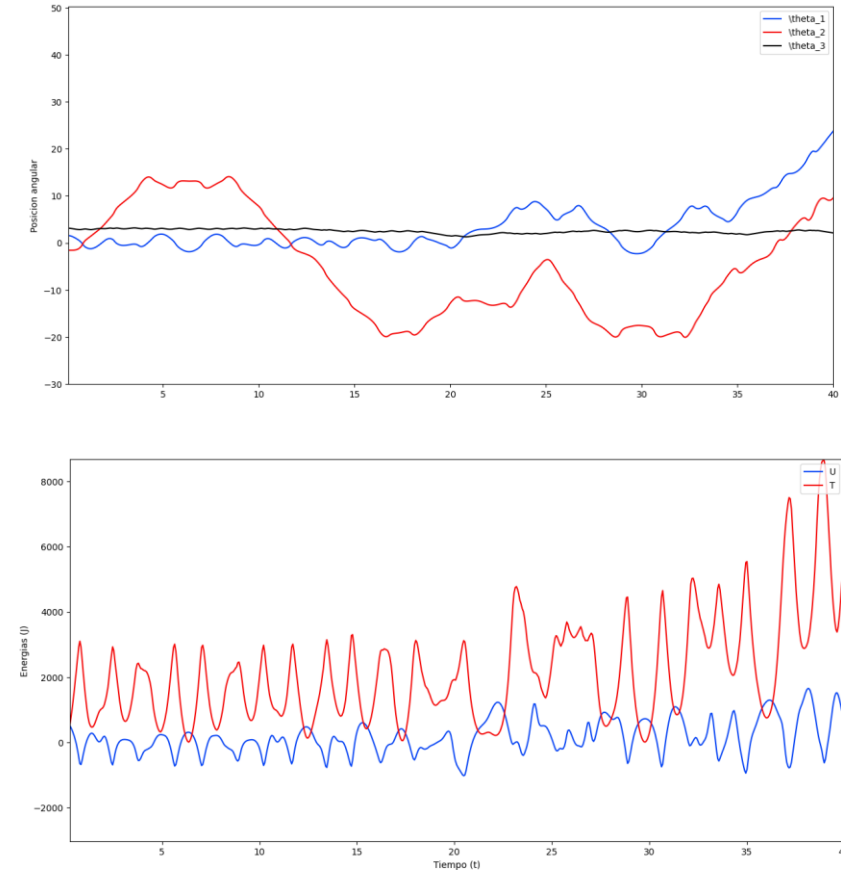


Además de esto, se realizó un análisis de la evolución de las coordenadas a lo largo del tiempo y de las energías en el sistema para ver si el comportamiento era el esperado o si se encontraba alguna irregularidad.

Primeras condiciones.



Segundas condiciones.



Conclusiones.

- Respecto a los métodos números que nos permitieron el análisis de las ecuaciones diferenciales.
- Los resultados obtenidos de los mapeos.
- Análisis energético del sistema y sub-sistema.
- Posibles extensiones del proyecto para un futuro.

Referencias hasta el 25/04/2022:

- Tjemsland, Oyre, Stovng. (08/03/2022). Double Pendulum and Chaotic Motion. Jupyter: https://nbviewer.org/urls/www.numfys.net/media/notebooks/double_pendulum.ipynb#apndx
- Guancho Arevalo Jose Luis. (08/03/2022). Solución Numerica a la Ecuación Diferencial No Lineal que describe las Oscilaciones del Péndulo Simple con Python: REVISTA DE CIENCIAS, Vol. 1, No. 2 de 2012.
- MARAZ, E, & BURGOA, O. (2014). SIMULACION DEL MOVIMIENTO DE UN PÉNDULO DOBLE EN UN MEDIO VISCOSO. Revista Boliviana de Física, 25(25), 27-30. Recuperado en 08 de marzo de 2022, de http://www.scielo.org.bo/scielo.php?script=sci_arttext&pid=S1562-38232014000200005&lng=es&tlng=es.
- Lee S. M. The double-simple pendulum problem. Am. J. Phys. 38 (1970) pp. 536-537
- García, A. (08/03/2022), Las Ecuaciones de Lagrange: <http://www.sc.ehu.es/sbweb/fisica3/dinamica/lagrange/lagrange.html>
- Teoria del Caos. (08/03/2022). En Wikipedia https://es.wikipedia.org/w/index.php?title=Teor%C3%ADa_del_caos&oldid=141076745
- The double pendulum. (z.d.). The double pendulum. Geraadpleegd op 17 juli 2017, van <https://scipython.com/blog/the-double-pendulum/>
- Experimento del método de cálculo (3): método de Runge-Kutta de cuarto orden - programador clic. (s/f). Programmerclick.com. Recuperado el 26 de abril de 2022, de <https://programmerclick.com/article/63021944030/>
- Domínguez, E., Ardila, F., & Bustamante, S. (s/f). System-Solver: una herramienta de código abierto para la modelación de sistemas dinámicos. Org.co. Recuperado el 26 de abril de 2022, de <http://www.scielo.org.co/pdf/iei/v30n3/v30n3a15.pdf>
- Función odeint . (s/f). Programmerclick.com. Recuperado el 26 de abril de 2022, de <https://programmerclick.com/article/74001063480/>
- IMULACION CON CIRCUITOS ELECTRONICOS DE LA PRACTICA DE LABORATORIO DE FÍSICA III: PÉNDULOS ACOPLADOS: Redalyc.org. 2022. [online] Available at: <https://www.redalyc.org/pdf/849/84917316036.pdf> [Accessed 30 May 2022].