Universidad de Guadalajara

centro universitario de ciencias exactas e ingenierías

Tarea 2: SDYC

Luis Ramón Mendoza Escareño

Licenciatura en Física | Sistemas Dinámicos y Caos

15 de febrero del 2022

I. Generar los valores de las constantes

```
#Parte 1
q0=5
v0=10
w=4
m=30

n=1 #Sistema q_0,v_0 (esto se verá porqué
después)
```

II. Generar la función que resuelve el sistema

```
    def giro(q_a,v_a,t_0,t_f,m):
     x_0=np.append(q_a,v_a)
     w_m=[[0,1],[-w**2,0]]
     I=np.eye(n)    #La 'n' es el número de
     partículas

     A=np.kron(w_m,I)

     t=np.linspace(t_0,t_f,m)
        def RHS(t,x):
            return np.matmul(A,x_0)

     x=odeint(RHS,x_0,t)
     return x
```

Entrega los valores de las q's y v's a través del tiempo tomando la información sobre el tiempo y los valores iniciales de q y v (La ω se toma como que ya está definida).

Seguido a esto sólo se llama a la función y se grafica.

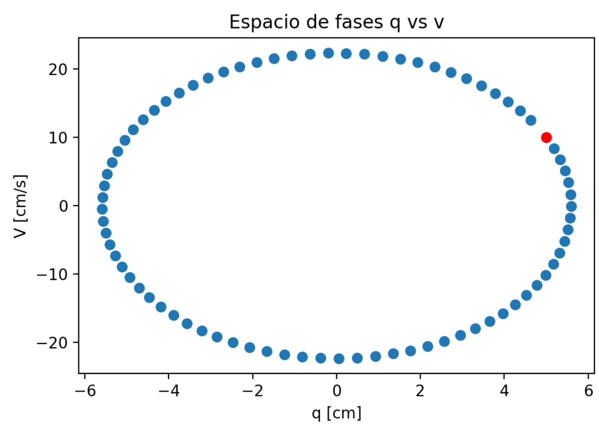

Figura 1: Llamada a la función 'giro' con los parámetros $(q_0, v_0, 0, 1.538, 80)$, el punto rojo es el punto inicial de la partícula.

Para tomar el 98 % de la trayectoria se tomó como que $T = \frac{2\pi}{\omega}$ y se multiplicó por 0.98, dando 1.53s.

III. Generar el círculo de valores iniciales para p y q

```
n=20
m=5
r=6
s=np.linspace(0,2*np.pi,n)
q_ac=r*np.cos(s)+10
v_ac=r*np.sin(s)+10
```

Parametrizando p y v con una variable extra 's', se generan los puntos de un círculo de radio 6 centrado en $(10, 10)$ con 20 puntos.

Acto seguido, sólo se llama a la función con estos puntos nuevos q_{ac} y v_{ac} y se grafica.

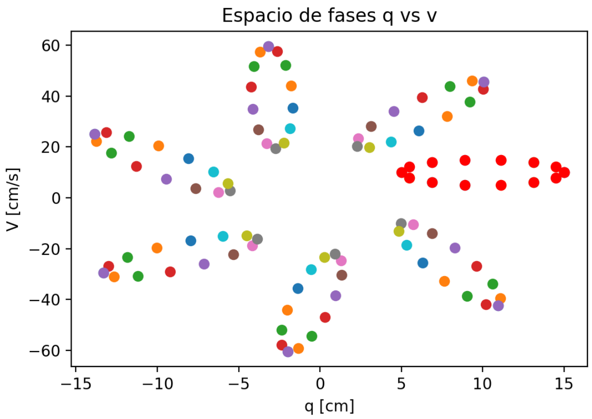

Figura 2: Ploteo de los valores que regresa la función giro($q_{ac}, v_{ac}, 0, 1.4, 7$), el círculo rojo son los valores iniciales.

IV. Sistema de partículas acopladas

Para la última parte tomamos el Hamiltoniano con el factor de acoplamiento $\alpha q_1^2 q_2^2$. Se generaron las ecuaciones de Hamilton y consideramos $\vec{x} = (q_0, q_1, v_0, v_1)$, de esta forma quedaron las ecuaciones diferenciales:

$$\begin{bmatrix} \dot{q_0} \\ \dot{q_1} \\ \dot{v_0} \\ \dot{v_1} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ -\omega_1^2 & \frac{-2\alpha}{m_1} q_2 q_1 & 0 & 0 \\ \frac{-2\alpha}{m_2} q_2 q_1 & -\omega_2^2 & 0 & 0 \end{bmatrix} \begin{bmatrix} q_0 \\ q_1 \\ v_0 \\ v_1 \end{bmatrix}$$

Entonces, realmente sólo se tiene que cambiar en la función anterior el RHS (incluyendo cambiar la matriz w) y generar las nuevas condiciones iniciales, las cuales fueron: $\alpha = 0.3$, $\omega_1 = 6$, $\omega_2 = 8$, $m_1 = 3$, $m_2 = 5$, $q_a = (8, 20)$, $v_a = (10, 15)$, con esto ya sólo se define el RHS_2 de la nueva función:

```
zero_m=np.zeros((4,4)) #Nueva matriz que
arriba llamamos 'w'
def RHS_2(x,t):
  w_m=[[-w1**2, -2*alpha/m1 *x[0]*x[1]],
  [-2*alpha/m2 *x[0]*x[1], -w2**2]]

  zero_m[2:,:2]=w_m
  zero_m[:2,2:]=np.eye(2,2)

  return np.matmul(zero_m,x)
```

Finalmente ya con esto, se llama a la función nueva y se guardan y plotean los resultados.

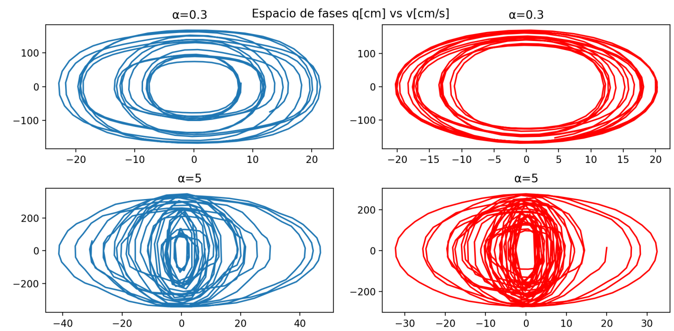

Figura 3: Espacio de fases para los mismas condiciones iniciales en el mismo tiempo, sólo cambiando el valor de α.

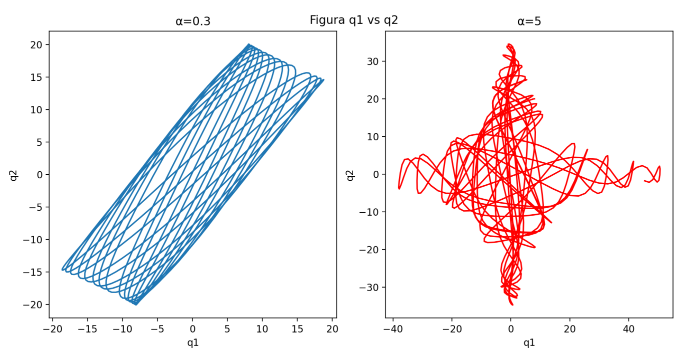

Figura 4: Valores de las posiciones q_1 y q_2 en el tiempo para diferentes valores de α.

De igual manera el link al script está en el google colab.