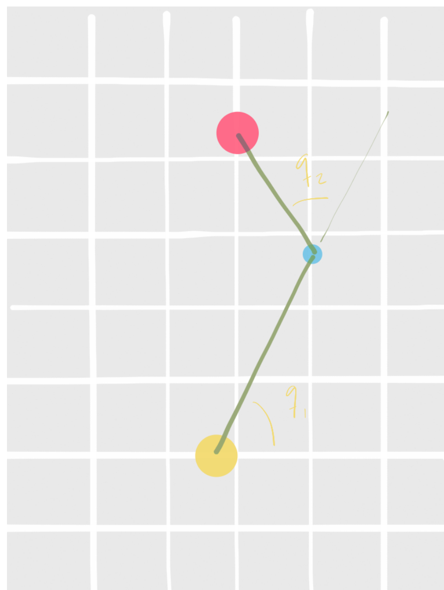
Pendubot

Roberto Cadena Vega

24 de abril de 2015

Examen Final

Modelado de pendubot



Variables de estado, posiciones y velocidades

$$q = \begin{pmatrix} q_1 \\ q_2 \end{pmatrix} \implies \dot{q} = \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \end{pmatrix}$$

Las posiciones de las juntas estan dadas por:

$$\begin{aligned} \begin{pmatrix} x_1 & y_1 \end{pmatrix} &= \begin{pmatrix} l_1 \cos(q_1) & l_1 \sin(q_1) \end{pmatrix} \\ \begin{pmatrix} x_2 & y_2 \end{pmatrix} &= \begin{pmatrix} l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) & l_1 \sin(q_1) + l_2 \sin(q_1 + q_2) \end{pmatrix} \end{aligned}$$

y por lo tanto, las velocidades estarán dadas por:

$$\begin{aligned} \begin{pmatrix} \dot{x}_1 & \dot{y}_1 \end{pmatrix} &= \begin{pmatrix} -l_1 \sin(q_1)\dot{q}_1 & l_1 \cos(q_1)\dot{q}_1 \end{pmatrix} \\ \begin{pmatrix} \dot{x}_2 & \dot{y}_2 \end{pmatrix} &= \begin{pmatrix} -l_1 \sin(q_1)\dot{q}_1 - l_2 \sin(q_1 + q_2)(\dot{q}_1 + \dot{q}_2) & l_1 \cos(q_1)\dot{q}_1 + l_2 \cos(q_1 + q_2)(\dot{q}_1 + \dot{q}_2) \end{pmatrix} \end{aligned}$$

Energía cinética

Considerando que la velocidad v_i es la norma de estas componentes, $v_i = \sqrt{\dot{x}_i^2 + \dot{y}_i^2}$, la energía cinética del sistema esta dada dada por:

$$\begin{aligned} K &= K_1 + K_2 \\ &= \frac{1}{2} m_1 v_1^2 + \frac{1}{2} m_2 v_2^2 \end{aligned}$$

con v_1^2 dada por:

$$\begin{aligned} v_1^2 &= \dot{x}_1^2 + \dot{y}_1^2 \\ &= l_1^2 \sin^2(q_1)\dot{q}_1^2 + l_1^2 \cos^2(q_1)\dot{q}_1^2 \\ &= l_1^2 \dot{q}_1^2 \end{aligned}$$

y v_2^2:

$$\begin{aligned} v_2^2 &= \dot{x}_2^2 + \dot{y}_2^2 \\ &= \left(-l_1 \sin(q_1)\dot{q}_1 - l_2 \sin(q_1+q_2)(\dot{q}_1+\dot{q}_2)\right)^2 + \left(l_1 \cos(q_1)\dot{q}_1 + l_2 \cos(q_1+q_2)(\dot{q}_1+\dot{q}_2)\right)^2 \\ &= l_1^2\dot{q}_1^2 + l_2^2(\dot{q}_1+\dot{q}_2)^2 + 2\left[l_1 \sin(q_1)\dot{q}_1 \cdot l_2 \sin(q_1+q_2)(\dot{q}_1+\dot{q}_2) + l_1 \cos(q_1)\dot{q}_1 \cdot l_2 \cos(q_1+q_2)(\dot{q}_1+\dot{q}_2)\right] \\ &= l_1^2\dot{q}_1^2 + l_2^2(\dot{q}_1+\dot{q}_2)^2 + 2l_1 l_2\left[\sin(q_1)\sin(q_1+q_2) + \cos(q_1)\cos(q_1+q_2)\right]\dot{q}_1(\dot{q}_1+\dot{q}_2) \\ &= l_1^2\dot{q}_1^2 + l_2^2(\dot{q}_1+\dot{q}_2)^2 + 2l_1 l_2 \cos(q_2)\dot{q}_1(\dot{q}_1+\dot{q}_2) \end{aligned}$$

por lo que las energías cinéticas de cada junta quedan:

$$\begin{aligned} K_1 &= \frac{1}{2}m_1 v_1^2 = \frac{1}{2}m_1 l_1^2 \dot{q}_1^2 \\ K_2 &= \frac{1}{2}m_2 v_2^2 = \frac{1}{2}m_2\left[l_1^2\dot{q}_1^2 + l_2^2(\dot{q}_1+\dot{q}_2)^2 + 2l_1 l_2 \cos(q_2)\dot{q}_1(\dot{q}_1+\dot{q}_2)\right] \\ &= \frac{1}{2}m_2\left[l_1^2\dot{q}_1^2 + l_2^2\dot{q}_1^2 + l_2^2\dot{q}_2^2 + 2l_2^2\dot{q}_1\dot{q}_2 + 2l_1 l_2\cos(q_2)\dot{q}_1^2 + 2l_1 l_2 \cos(q_2)\dot{q}_1\dot{q}_2\right] \\ &= \frac{1}{2}m_2\left[\left(l_1^2 + l_2^2 + 2l_1 l_2\cos(q_2)\right)\dot{q}_1^2 + l_2^2\dot{q}_2^2 + 2\left(l_2^2 + l_1 l_2\cos(q_2)\right)\dot{q}_1\dot{q}_2\right] \end{aligned}$$

por lo que la energía cinética total, será:

$$\begin{aligned} K &= K_1 + K_2 \\ &= \frac{1}{2}m_1 l_1^2\dot{q}_1^2 + \frac{1}{2}m_2\left[\left(l_1^2 + l_2^2 + 2l_1 l_2\cos(q_2)\right)\dot{q}_1^2 + l_2^2\dot{q}_2^2 + 2\left(l_2^2 + l_1 l_2\cos(q_2)\right)\dot{q}_1\dot{q}_2\right] \\ &= \frac{1}{2}\left\{m_1 l_1^2\dot{q}_1^2 + m_2\left[\left(l_1^2 + l_2^2 + 2l_1 l_2\cos(q_2)\right)\dot{q}_1^2 + l_2^2\dot{q}_2^2 + 2\left(l_2^2 + l_1 l_2\cos(q_2)\right)\dot{q}_1\dot{q}_2\right]\right\} \\ &= \frac{1}{2}\left\{\left[m_1 l_1^2 + m_2\left(l_1^2 + l_2^2 + 2l_1 l_2\cos(q_2)\right)\right]\dot{q}_1^2 + m_2 l_2^2\dot{q}_2^2 + 2m_2\left(l_2^2 + l_1 l_2\cos(q_2)\right)\dot{q}_1\dot{q}_2\right\} \\ &= \frac{1}{2}\left\{\left[(m_1+m_2)l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2\cos(q_2)\right]\dot{q}_1^2 + m_2 l_2^2\dot{q}_2^2 + 2m_2\left(l_2^2 + l_1 l_2\cos(q_2)\right)\dot{q}_1\dot{q}_2\right\} \end{aligned}$$

La cual, al ver las relaciones con las variables de posición generalizadas del pendulo, podemos escribir como:

$$K = \frac{1}{2}\dot{q}^T M(q)\dot{q}$$

en donde la matriz de inercia $M(q)$ queda de la forma:

$$M(q) = \begin{pmatrix} (m_1+m_2)l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2\cos(q_2) & m_2 l_2^2 + m_2 l_1 l_2\cos(q_2) \\ m_2 l_2^2 + m_2 l_1 l_2 \cos(q_2) & m_2 l_2^2 \end{pmatrix}$$

en donde podemos introducir las constantes μ_i para simplificar la notación:

$$\begin{aligned} \mu_1 &= (m_1+m_2)l_1^2 \\ \mu_2 &= m_2 l_2^2 \\ \mu_3 &= m_2 l_1 l_2 \end{aligned}$$

con lo que la expresión para la energía cinética queda:

$$K = \frac{1}{2}\dot{q}^T \begin{pmatrix} \mu_1 + \mu_2 + 2\mu_3\cos(q_2) & \mu_2 + \mu_3\cos(q_2) \\ \mu_2 + \mu_3\cos(q_2) & \mu_2 \end{pmatrix}\dot{q}$$

Energía potencial

La energía potencial del sistema esta dada por:

$$\begin{aligned}
U &= U_1 + U_2 \\
&= m_1 g h_1 + m_2 g h_2 \\
&= m_1 g y_1 + m_2 g y_2 \\
&= m_1 g l_1 \sin(q_1) + m_2 g \left[l_1 \sin(q_1) + l_2 \sin(q_1 + q_2) \right] \\
&= (m_1 + m_2)\, g l_1 \sin(q_1) + m_2 g l_2 \sin(q_1 + q_2) \\
&= g \left[(m_1 + m_2)\, l_1 \sin(q_1) + m_2 l_2 \sin(q_1 + q_2) \right]
\end{aligned}$$

Lagrangiano

El Lagrangiano del sistema, esta dado por la expresión:

$$L = K - U$$

por lo que solo queda sumar las dos expresiones y obtener las condiciones de optimalidad para la energía del sistema por medio de la ecuación de Euler-Lagrange

Euler-Lagrange

$$\frac{d}{dt}\left(\frac{\partial L}{\partial \dot{q}}\right) - \frac{\partial L}{\partial q} = 0$$

Si empezamos derivando con respecto a $\dot{q}$, tendremos:

$$\frac{\partial L}{\partial \dot{q}} = \frac{\partial K}{\partial \dot{q}} - \frac{\partial U}{\partial \dot{q}}$$

y obteniendo cada uno de estos tenemos que:

$$\begin{aligned}
\frac{\partial K}{\partial \dot{q}} &= K_{\dot{q}} = \frac{\partial}{\partial \dot{q}} \left\{ \frac{1}{2} \dot{q}^T M(q) \dot{q} \right\} = M(q)\dot{q} \\
\frac{\partial U}{\partial \dot{q}} &= U_{\dot{q}} = 0
\end{aligned}$$

y la derivada con respecto al tiempo de estos terminos será:

$$\begin{aligned}
\frac{dK_{\dot{q}}}{dt} &= M(q)\ddot{q} + \dot{M}(q,\dot{q})\dot{q} \\
\frac{dU_{\dot{q}}}{dt} &= 0
\end{aligned}$$

en donde:

$$\begin{aligned}
\dot{M}(q,\dot{q}) &= \frac{d}{dt} \begin{pmatrix} \mu_1 + \mu_2 + 2\mu_3 \cos(q_2) & \mu_2 + \mu_3 \cos(q_2) \\ \mu_2 + \mu_3 \cos(q_2) & \mu_2 \end{pmatrix} \\
&= \begin{pmatrix} -2\mu_3 \sin(q_2)\dot{q}_2 & -\mu_3 \sin(q_2)\dot{q}_2 \\ -\mu_3 \sin(q_2)\dot{q}_2 & 0 \end{pmatrix} \\
&= -\mu_3 \sin(q_2) \begin{pmatrix} 2\dot{q}_2 & \dot{q}_2 \\ \dot{q}_2 & 0 \end{pmatrix}
\end{aligned}$$

Ahora, derivando con respecto a q, tenemos:

$$\frac{\partial L}{\partial q} = \frac{\partial K}{\partial q} - \frac{\partial U}{\partial q}$$

en donde:

$$
\begin{aligned}
\frac{\partial K}{\partial q} = K_q = \begin{pmatrix} \frac{\partial K}{\partial q_1} \\ \frac{\partial K}{\partial q_2} \end{pmatrix} \\
&= \frac{1}{2} \begin{pmatrix} \frac{\partial}{\partial q_1} \left\{ \left[(m_1+m_2)l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 \cos(q_2) \right] \dot{q}_1^2 + m_2 l_2^2 \dot{q}_2^2 + 2m_2 \left(l_2^2 + l_1 l_2 \cos(q_2) \right) \dot{q}_1 \dot{q}_2 \right\} \\ \frac{\partial}{\partial q_2} \left\{ \left[(m_1+m_2)l_1^2 + m_2 l_2^2 + 2m_2 l_1 l_2 \cos(q_2) \right] \dot{q}_1^2 + m_2 l_2^2 \dot{q}_2^2 + 2m_2 \left(l_2^2 + l_1 l_2 \cos(q_2) \right) \dot{q}_1 \dot{q}_2 \right\} \end{pmatrix} \\
&= \frac{1}{2} \begin{pmatrix} 0 \\ -2m_2 l_1 l_2 \sin(q_2) \dot{q}_1^2 - 2m_2 l_1 l_2 \sin(q_2) \dot{q}_1 \dot{q}_2 \end{pmatrix} \\
&= -\mu_3 \sin(q_2) \begin{pmatrix} 0 \\ \dot{q}_1^2 + \dot{q}_1 \dot{q}_2 \end{pmatrix} = -\mu_3 \sin(q_2) \begin{pmatrix} \dot{q}_2 & -\dot{q}_1 \\ \dot{q}_1 + \dot{q}_2 & 0 \end{pmatrix} \dot{q}
\end{aligned}
$$

y para la energía potencial:

$$
\begin{aligned}
\frac{\partial U}{\partial q} = U_q = \begin{pmatrix} \frac{\partial U}{\partial q_1} \\ \frac{\partial U}{\partial q_2} \end{pmatrix} \\
&= g \begin{pmatrix} \frac{\partial}{\partial q_1} \left[(m_1+m_2) l_1 \sin(q_1) + m_2 l_2 \sin(q_1+q_2) \right] \\ \frac{\partial}{\partial q_2} \left[(m_1+m_2) l_1 \sin(q_1) + m_2 l_2 \sin(q_1+q_2) \right] \end{pmatrix} \\
&= g \begin{pmatrix} (m_1+m_2) l_1 \cos(q_1) + m_2 l_2 \cos(q_1+q_2) \\ m_2 l_2 \cos(q_1+q_2) \end{pmatrix}
\end{aligned}
$$

o bien:

$$
U_q = g \begin{pmatrix} \mu_4 \cos(q_1) + \mu_5 \cos(q_1+q_2) \\ \mu_5 \cos(q_1+q_2) \end{pmatrix}
$$

con $\mu_4 = (m_1+m_2) l_1$ y $\mu_5 = m_2 l_2$, por lo que la ecuación de Euler-Lagrange queda:

$$
\begin{aligned}
\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} = 0 \\
\frac{dK_{\dot{q}}}{dt} - \left(K_q - U_q \right) = 0 \\
M(q)\ddot{q} + \dot{M}(q,\dot{q})\dot{q} - K_q + U_q = 0
\end{aligned}
$$

en donde recordando que K_q tiene la forma $F(q,\dot{q})\dot{q}$, podemos juntar esta matriz con $\dot{M}(q,\dot{q})$ y obtener una expresión de la forma:

$$
M(q)\ddot{q} + C(q,\dot{q})\dot{q} + U_q = 0
$$

en donde:

$$
\begin{aligned}
M(q) &= \begin{pmatrix} \mu_1 + \mu_2 + 2\mu_3 \cos(q_2) & \mu_2 + \mu_3 \cos(q_2) \\ \mu_2 + \mu_3 \cos(q_2) & \mu_2 \end{pmatrix} \\
C(q,\dot{q}) &= \dot{M}(q,\dot{q}) - F(q,\dot{q}) = -\mu_3 \sin(q_2) \begin{pmatrix} \dot{q}_2 & \dot{q}_1 + \dot{q}_2 \\ -\dot{q}_1 & 0 \end{pmatrix} \\
U_q &= g \begin{pmatrix} \mu_4 \cos(q_1) + \mu_5 \cos(q_1+q_2) \\ \mu_5 \cos(q_1+q_2) \end{pmatrix}
\end{aligned}
$$

con

$$
\begin{aligned}
\mu_1 &= (m_1+m_2) l_1^2 \\
\mu_2 &= m_2 l_2^2 \\
\mu_3 &= m_2 l_1 l_2 \\
\mu_4 &= (m_1+m_2) l_1 \\
\mu_5 &= m_2 l_2
\end{aligned}
$$

sin embargo este modelo no considera las señales de control, por lo que procederemos a analizar el sistema.

Señales de control

Nuestra señal de control escalar se deberá enlazar al actuador del pendubot, el cual esta asociado al primer grado de libertad q_1, por lo que finalmente tenemos la ecuación:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + U_q = Gu$$

con:

$$G = \begin{pmatrix} 1 \\ 0 \end{pmatrix}$$

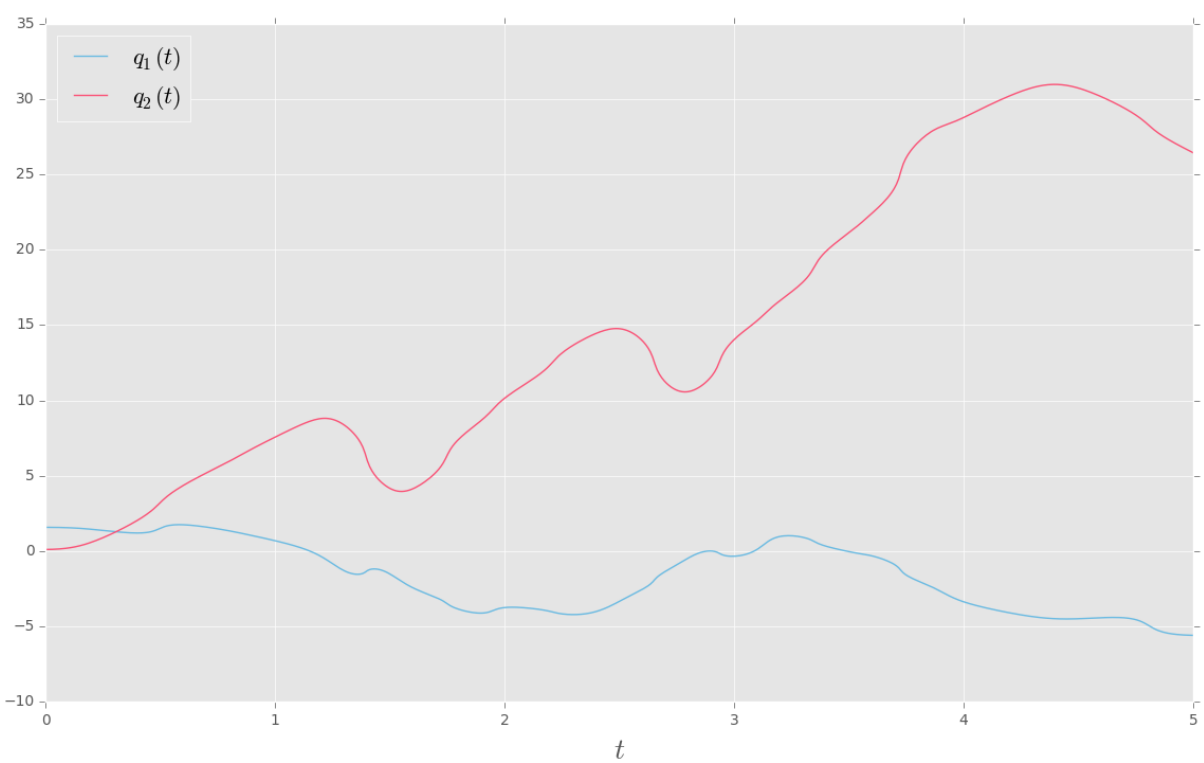
```
In [2]: def f(estado, tiempo, K):
            '''Modelo matematico pendubot
            Esta función f es tal que ẋ = f(x, t), es decir, dado el estado del sistema "x"
            y el tiempo "t", devuelve la derivada en el tiempo "t". Para el caso del pendubot,
            es invariante en el tiempo, por lo que es equivalente a ẋ = f(x).

            Ejemplo
            -------
            >>> x = [τ/4, 0.1, 0, 0]
            >>> t = 0
            >>> f(x, t)
            array([  0.        ,   0.        ,  -4.73091048,  17.41001448])
            '''

            from numpy import matrix, zeros, cos, sin, pi

            # Constantes fisicas del sistema
            m1 = 0.2 # kg
            m2 = 0.6 # kg
            l1 = 0.6 # m
            l2 = 0.3 # m
            g = 9.81 # m/s^2
            τ = 2*pi

            # Variables auxiliares del sistema
            μ1 = (m1 + m2)*l1**2
            μ2 = m2*l2**2
            μ3 = m2*l1*l2
            μ4 = (m1 + m2)*l1
            μ5 = m2*l2

            # Estado agregado
            q1, q2, q_1, q_2 = estado

            # Estado del sistema y derivada
            q = matrix([[q1], [q2]])
            q_ = matrix([[q_1], [q_2]])

            # Matrix de inercia generalizada
            M = matrix([[μ1 + μ2 + 2*μ3*cos(q2), μ2 + μ3*cos(q2)],
                        [μ2 + μ3*cos(q2), μ2]])

            # Matriz de Coriolis
            C = -μ3*sin(q2)*matrix([[ q_2, q_1 + q_2],
                                     [-q_1, 0]])

            # Vector de gravedad
            U = g*matrix([[μ4*cos(q1) + μ5*cos(q1 + q2)],
                          [μ5*cos(q1 + q2)]])
```

```
    # Vector de señales de control
    G = matrix([[1], [0]])

    ξ = matrix(estado).T - matrix([[τ/4], [0], [0], [0]])

    # Ley de control
    u = -(matrix(K)*ξ).tolist()[0][0]

    # Aceleracion del sistema
    q̈ = M.I*(G*u - C*q_ - U)

    # Incializamos la derivada del estado agregado
    ẏ = zeros(4)

    # Se asignan los valores a la derivada del estado agregado
    ẏ[0] = q_1
    ẏ[1] = q_2
    ẏ[2] = q̈[0]
    ẏ[3] = q̈[1]

    # Se devuelve la derivada
    return ẏ
```

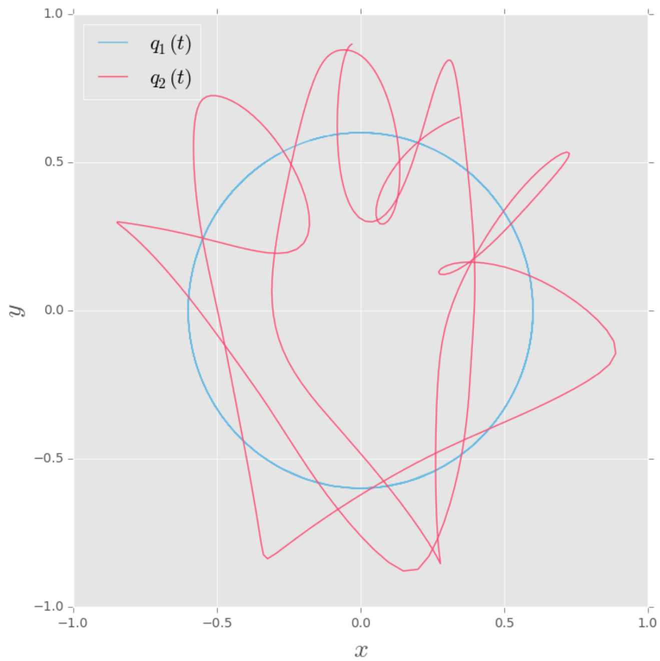
```
In [4]: from scipy.integrate import odeint
        from numpy import pi, linspace, sin, cos, array
        τ = 2*pi
```

```
In [5]: ts = linspace(0, 5, 500)
        estado_inicial = [τ/4, 0.1, 0, 0]

        f_libre = lambda x, t: f(x, t, [0, 0, 0, 0])

        x = odeint(f_libre, estado_inicial, ts)
```

```
In [7]: graficatemporal("./imagenes/trayectoriaspendubotlibre.png", x, ts, [r'$q_1(t)$', r'$q_2(t)$'])
```



```
In [9]: graficaxy("./imagenes/xypendubotlibre.png", x, [r'$q_1(t)$', r'$q_2(t)$'])
```



Linealización del modelo del pendubot

Estado agregado

Dado el modelo de este sistema no lineal que hemos obtenido, podemos linealizar alrededor de los puntos de equilibrio, pero primero tenemos que obtener las ecuaciones del sistema agregado, de tal manera que:

$$\dot{x} = f(x, u)$$

por lo que primero convertimos el modelo representado por una ecuación diferencial de segundo orden, por uno representado por una ecuación diferencial ordinaria de primer orden. Recordemos el modelo obtenido:

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + U_q = Gu \implies \ddot{q} = M^{-1}(q) \left[Gu - C(q, \dot{q})\dot{q} - U_q \right]$$

Si definimos el estado x como:

$$x = \begin{pmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{pmatrix} = \begin{pmatrix} q_1 \\ q_2 \\ \dot{q}_1 \\ \dot{q}_2 \end{pmatrix}$$

significa que tenemos que encontrar $f(x, u)$ tal que:

$$\dot{x} = \begin{pmatrix} \dot{x}_1 \\ \dot{x}_2 \\ \dot{x}_3 \\ \dot{x}_4 \end{pmatrix} = \begin{pmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \ddot{q}_1 \\ \ddot{q}_2 \end{pmatrix} = \begin{pmatrix} f_1(x, u) \\ f_2(x, u) \\ f_3(x, u) \\ f_4(x, u) \end{pmatrix} = f(x, u)$$

en donde f_1 y f_2, trivialmente son:

$$f_1(x, u) = x_3 = \dot{q}_1$$
$$f_2(x, u) = x_4 = \dot{q}_2$$

y f_3 y f_4 estan dadas por el modelo obtenido:

$$\begin{pmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{pmatrix} = \ddot{q} = M^{-1}(q) \left[Gu - C(q, \dot{q})\dot{q} - U_q \right] = \begin{pmatrix} f_3(x, u) \\ f_4(x, u) \end{pmatrix}$$

por lo que ahora debemos obtener expresiones escalares para la dinámica del pendubot obtenida.

Ecuaciones escalares de la dinámica del pendubot

Empezamos calculando $M^{-1}(q)$:

$$\begin{aligned} M^{-1}(q) &= \frac{1}{\det\left(M(q)\right)} \operatorname{adj}\left(M(q)\right) \\ &= \frac{1}{\mu_1\mu_2 - \mu_3^2\cos^2(q_2)} \begin{pmatrix} \mu_2 & -\mu_2 - \mu_3\cos(q_2) \\ -\mu_2 - \mu_3\cos(q_2) & \mu_1 + \mu_2 + 2\mu_3\cos(q_2) \end{pmatrix} \end{aligned}$$

y las ecuaciones que queremos obtener:

$$\begin{pmatrix} f_3(x,u) \\ f_4(x,u) \end{pmatrix} = M^{-1}(q)Gu - M^{-1}(q)C(q,\dot{q})\dot{q} - M^{-1}(q)U_q$$

tienen terminos:

$$\begin{aligned} M^{-1}(q)Gu &= \frac{1}{\mu_1\mu_2 - \mu_3^2\cos^2(q_2)} \begin{pmatrix} \mu_2 & -\mu_2 - \mu_3\cos(q_2) \\ -\mu_2 - \mu_3\cos(q_2) & \mu_1 + \mu_2 + 2\mu_3\cos(q_2) \end{pmatrix} \begin{pmatrix} 1 \\ 0 \end{pmatrix} u \\ &= \frac{1}{\mu_1\mu_2 - \mu_3^2\cos^2(q_2)} \begin{pmatrix} \mu_2 \\ -\mu_2 - \mu_3\cos(q_2) \end{pmatrix} u \\ -M^{-1}(q)C(q,\dot{q})\dot{q} &= \frac{\mu_3\sin(q_2)}{\mu_1\mu_2 - \mu_3^2\cos^2(q_2)} \begin{pmatrix} \mu_2 & -\mu_2 - \mu_3\cos(q_2) \\ -\mu_2 - \mu_3\cos(q_2) & \mu_1 + \mu_2 + 2\mu_3\cos(q_2) \end{pmatrix} \begin{pmatrix} \dot{q}_2 & \dot{q}_1 + \dot{q}_2 \\ -\dot{q}_1 & 0 \end{pmatrix} \dot{q} \\ &= \frac{\mu_3\sin(q_2)}{\mu_1\mu_2 - \mu_3^2\cos^2(q_2)} \begin{pmatrix} \mu_2\left(\dot{q}_1 + \dot{q}_2\right)^2 + \mu_3\cos(q_2)\dot{q}_1^2 \\ -\mu_1\dot{q}_1^2 - \mu_2\left(\dot{q}_1 + \dot{q}_2\right)^2 - \mu_3\cos(q_2)\left[\left(\dot{q}_1 + \dot{q}_2\right)^2 + \dot{q}_1^2\right] \end{pmatrix} \end{aligned}$$

$$\begin{aligned} -M^{-1}(q)U_q &= -\frac{g}{\mu_1\mu_2 - \mu_3^2\cos^2(q_2)} \begin{pmatrix} \mu_2 & -\mu_2 - \mu_3\cos(q_2) \\ -\mu_2 - \mu_3\cos(q_2) & \mu_1 + \mu_2 + 2\mu_3\cos(q_2) \end{pmatrix} \begin{pmatrix} \mu_4\cos(q_1) + \mu_5\cos(q_1 + q_2) \\ \mu_5\cos(q_1 + q_2) \end{pmatrix} \\ &= -\frac{g}{\mu_1\mu_2 - \mu_3^2\cos^2(q_2)} \begin{pmatrix} \mu_2\mu_4\cos(q_1) - \mu_3\mu_5\cos(q_2)\cos(q_1 + q_2) \\ -\mu_2\mu_4\cos(q_1) - \mu_3\mu_4\cos(q_1)\cos(q_2) + \mu_1\mu_5\cos(q_1 + q_2) + \mu_3\mu_5\cos(q_2)\cos(q_1 + q_2) \end{pmatrix} \end{aligned}$$

Por lo que al juntar todos los terminos, y hacer un poco de algebra, tendremos que:

$$\begin{aligned} f_3(x,u) &= \frac{\mu_2 u + \mu_3\mu_5\sin(q_2)\left[l_2\left(\dot{q}_1 + \dot{q}_2\right)^2 + l_1\cos(q_2)\dot{q}_1^2\right] - g\mu_3\left[\mu_4\cos(q_1) + \mu_5\cos(q_2)\cos(q_1 + q_2)\right]}{\mu_1\mu_2 - \mu_3^2\cos^2(q_2)} \\ f_4(x,u) &= \frac{-\mu_5\left(l_2 + l_1\cos(q_2)\right)u - \mu_3\sin(q_2)\left\{m_1 l_1^2\dot{q}_1^2 + \mu_5\left[\left(l_2 + l_1\cos(q_2)\right)\left(\dot{q}_1 + \dot{q}_2\right)^2 + l_1\cos(q_2)\dot{q}_1^2\right]\right\}}{\mu_1\mu_2 - \mu_3^2\cos^2(q_2)} \\ &\quad + \frac{g\mu_3\left[\left(m_1 + m_2\right)\left(l_2\cos(q_1) + l_1\sin(q_1)\sin(q_2)\right) - \mu_5\cos(q_2)\cos(q_1 + q_2)\right]}{\mu_1\mu_2 - \mu_3^2\cos^2(q_2)} \end{aligned}$$

Linealización del sistema no lineal

Dado este sistema no lineal, podemos obtener una linealización alrededor de algun punto de equilibrio al considerar la expansión de Taylor de la dinámica del sistema

$$f(x,u) = f(\bar{x},\bar{u}) + \left.\frac{\partial f}{\partial x}\right|_{x=\bar{x},u=\bar{u}} (x - \bar{x}) + \left.\frac{\partial f}{\partial u}\right|_{x=\bar{x},u=\bar{u}} (u - \bar{u}) + o(x,u)$$

en donde $f(\bar{x},\bar{u}) = 0$ y $o(x,u)$ son terminos de orden superior, por lo que al tener en cuenta variaciones pequeñas de nuestra diámica, podemos despreciarlas y obtener:

$$\dot{\xi} = A\xi + Bu$$

en donde $\xi = x - \bar{x}$ es el estado de nuestra linealización, y A y B estan dados por:

$$\begin{aligned} A &= \left.\frac{\partial f}{\partial x}\right|_{x=\bar{x},u=\bar{u}} \\ B &= \left.\frac{\partial f}{\partial u}\right|_{x=\bar{x},u=\bar{u}} \end{aligned}$$

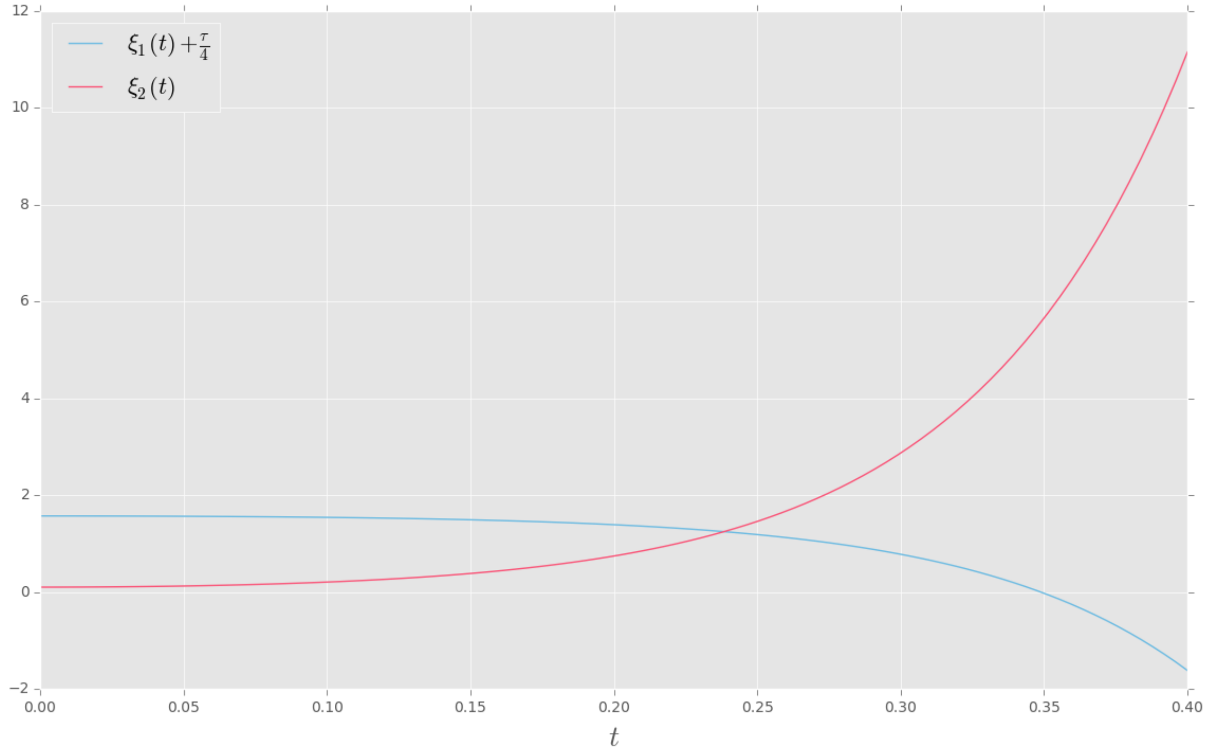
Calculando estas derivadas, tenemos que nuestra linealización queda:

$$A = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \\ \frac{g}{l_1} & -\frac{g m_2}{l_1 m_1} & 0 & 0 \\ -\frac{g}{l_1} & \frac{g}{m_1}\left(\frac{m_1}{l_2} + \frac{m_2}{l_2} + \frac{m_2}{l_1}\right) & 0 & 0 \end{pmatrix}$$

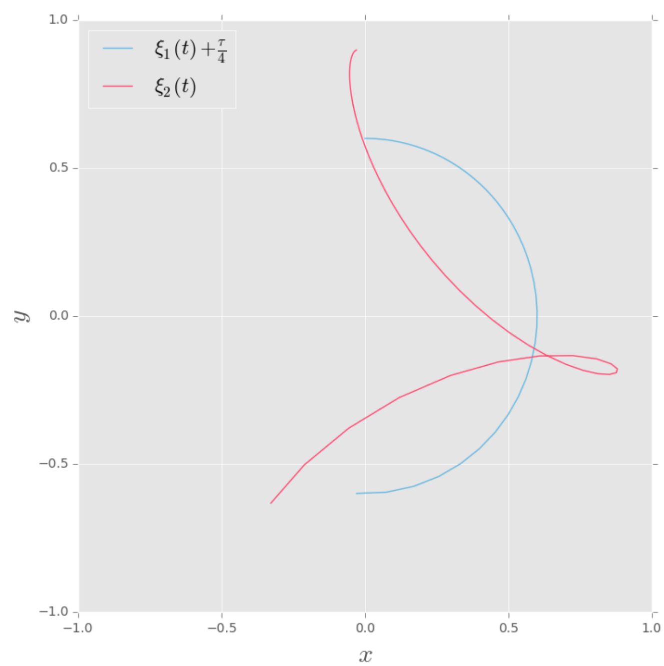
$$B = \begin{pmatrix} 0 \\ 0 \\ \frac{1}{m_1 l_1^2} \\ -\frac{l_1 + l_2}{m_1 l_1^2 l_2} \end{pmatrix}$$

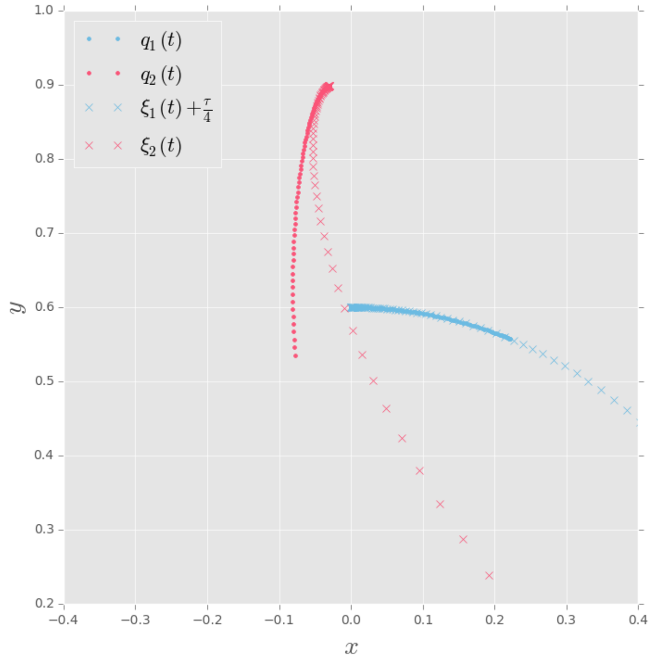
```
In [12]: def f_lineal(x, t, K):
             '''Modelo matematico lineal pendubot
             Esta función f_lineal es tal que ẋ = f(x, t), es decir, dado el estado del sistema "x"
             y el tiempo "t", devuelve la derivada en el tiempo "t". Para el caso del pendubot,
             es invariante en el tiempo, por lo que es equivalente a ẋ = f(x).

             Ejemplo
             -------
             >>> x = [τ/4, 0.1, 0, 0]
             >>> t = 0
             >>> f(x, t)
             [0.0, 0.0, -4.905000000000001, 17.985]
             '''

             from numpy import matrix, zeros, cos, sin

             # Constantes fisicas del sistema
             m1 = 0.2 # kg
             m2 = 0.6 # kg
             l1 = 0.6 # m
             l2 = 0.3 # m
             g = 9.81 # m/s^2

             # Matriz sistema linealizado
             A = matrix([[0, 0, 1, 0],
                         [0, 0, 0, 1],
                         [g/l1, -g*m2/(l1*m1), 0, 0],
                         [-g/l1, g/m1*(m1/l2 + m2/l2 + m2/l1), 0, 0]])

             # Vector de entrada
             B = matrix([[0],[0],[1/(m1*l1**2)],[-(l1 + l2)/(m1*l1**2*l2)]])

             # Ley de control lineal
             u = -matrix(K)*matrix(x).T

             # Dinámica del sistema linealizado
             ẋ = A*matrix(x).T + B*u

             # Se devuelve la derivada
             return (ẋ.T).tolist()[0]

In [13]: ts = linspace(0, 0.4, 100)
         estado_inicial_lineal = [0.0, 0.1, 0.0, 0.0]

         f_lineal_libre = lambda x, t: f_lineal(x, t, [0, 0, 0, 0])

         ζs = odeint(f_lineal_libre, estado_inicial_lineal, ts)
         xl = array([ζ + array([τ/4, 0, 0, 0]) for ζ in ζs])
```

In [14]: graficatemporal("./imagenes/trayectoriaspendubotlineallibre.png", xl, ts, [r'$\xi_1(t) + \frac{\tau}{4}



In [15]: graficaxy("./imagenes/xypendubotlineallibre.png", xl, [r'$\xi_1(t) + \frac{\tau}{4}$', r'$\xi_2(t)$'])





Aqui podemos ver como el sistema linealizado diverge rapidamente del sistema original, por lo que podemos asumir que este modelo linealizado es valido para variaciones muy pequeñas de q_1 y q_2.

Diseño de un control óptimo

Para el diseño de la ley de control óptima que minimice la funcional:

$$J(u) = \int_0^\infty \left(x^T(t) Q x(t) + u^2(t) \right) dt$$

con Q de la forma:

$$Q = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 10 & 0 \\ 0 & 0 & 0 & 10 \end{pmatrix}$$

tal que la ley de control óptima sea:

$$u^*(t) = -K\xi^*(t)$$

Para calcular la ganancia K asociada a esta función de costo, utilizaremos el sistema lineal encontrado.

```
In [18]: from control import lqr
         from numpy import matrix
```

```
In [19]: m1 = 0.2 # kg
         m2 = 0.6 # kg
         l1 = 0.6 # m
         l2 = 0.3 # m
         g = 9.81 # m/s^2

         A = matrix([[0, 0, 1, 0],
                     [0, 0, 0, 1],
                     [g/l1, -g*m2/(l1*m1), 0, 0],
                     [-g/l1, g/m1*(m1/l2 + m2/l2 + m2/l1), 0, 0]])

         B = matrix([[0],
                     [0],
```

```
                [1/(m1*l1**2)],
                [-(l1+l2)/(m1*l1**2*l2)]])

        Q = matrix([[1, 0, 0, 0],
                    [0, 1, 0, 0],
                    [0, 0, 10, 0],
                    [0, 0, 0, 10]])

        R = matrix([[1]])
```

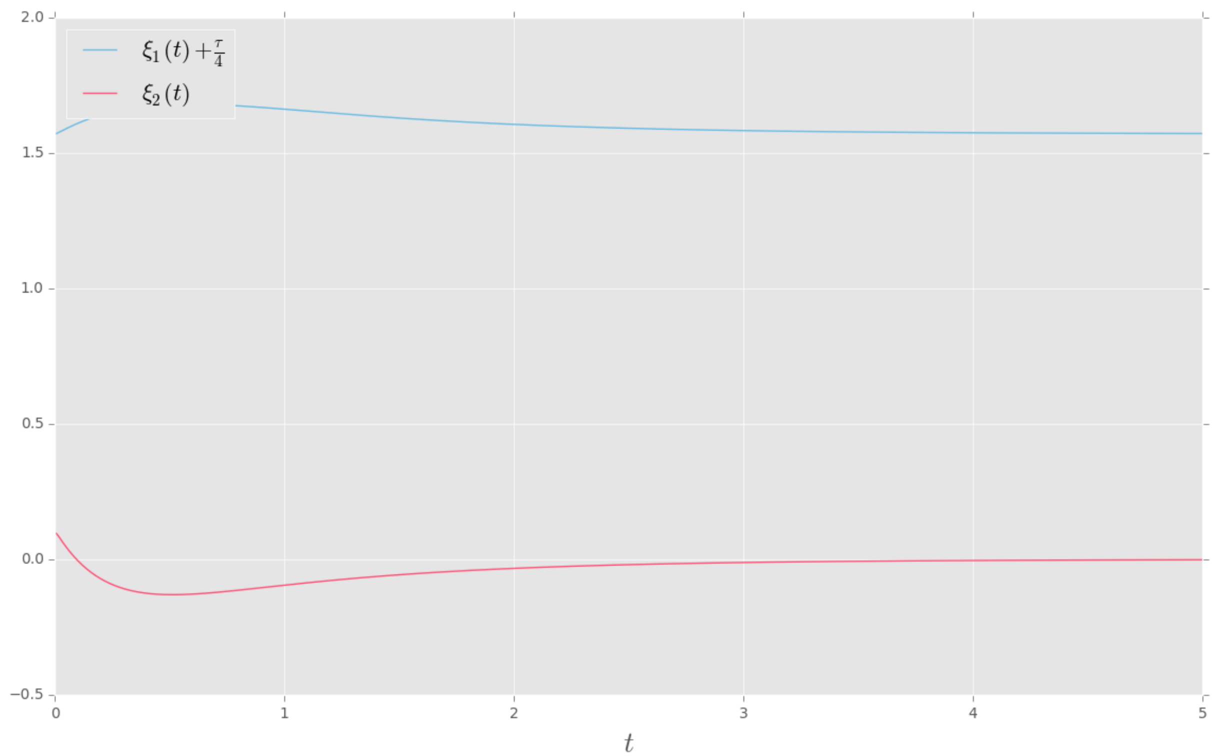
```
In [20]: K, S, E = lqr(A, B, Q, R)
         K
```

```
Out[20]: array([[-51.600978  , -51.39319408, -16.05940057,  -8.92417942]])
```

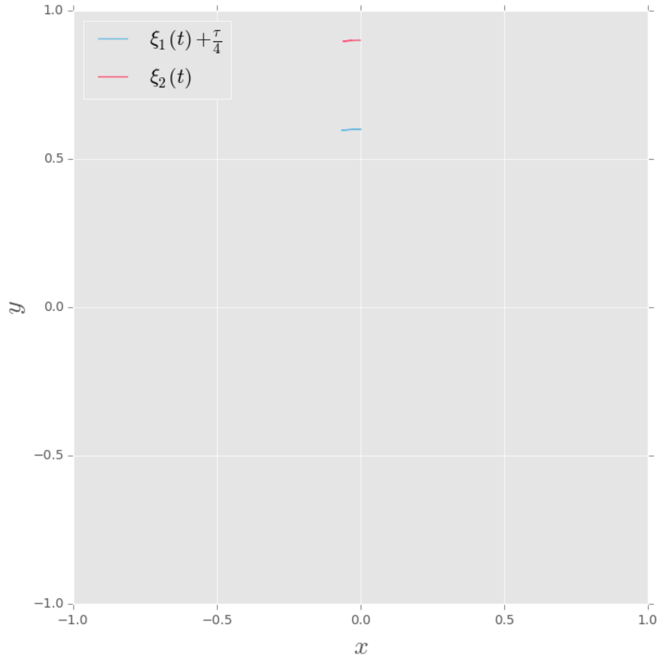
Una vez que hemos obtenido las ganancias de realimentacion, tan solo tenemos que incluirlas en la simulación del sistema para ver su desempeño.

```
In [21]: ts = linspace(0, 5, 500)
         estado_inicial_lineal = [0.0, 0.1, 0.0, 0.0]

         f_lineal_optima = lambda x, t: f_lineal(x, t, K)

         ζs = odeint(f_lineal_optima, estado_inicial_lineal, ts)
         xl = array([ζ + array([τ/4, 0, 0, 0]) for ζ in ζs])
```

```
In [22]: graficatemporal("./imagenes/trayectoriaspendubotlinealoptimo.png", xl, ts, [r'$\xi_1(t) + \frac{\tau}{4
```




```
In [23]: graficaxy("./imagenes/xypendubotlinealoptimo.png", xl, [r'$\xi_1(t) + \frac{\tau}{4}$', r'$\xi_2(t)$'])
```

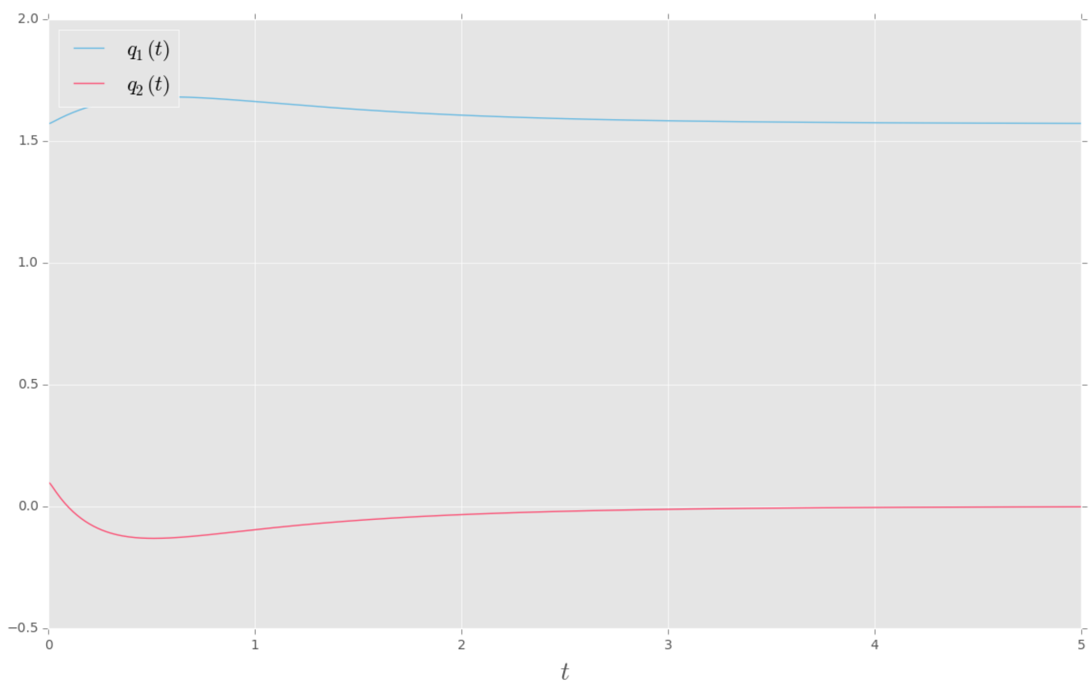


Ahora sabemos que este control estabiliza al sistema linealizado, pero en realidad queremos saber como se desempeña con el sistema no lineal.

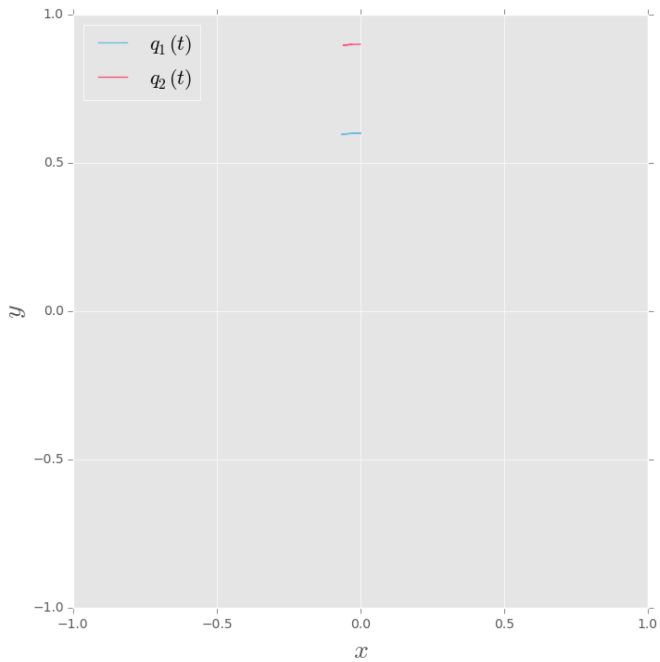
Podemos realimentar el sistema original para ver su desempeño.

```
In [24]: ts = linspace(0, 5, 500)
         estado_inicial = [τ/4, 0.1, 0, 0]

         f_optima = lambda x, t: f(x, t, K)

         x = odeint(f_optima, estado_inicial, ts)
```

```
In [25]: graficatemporal("./imagenes/trayectoriaspendubotoptimo.png", x, ts, [r'$q_1(t)$', r'$q_2(t)$'])
```



```
In [26]: graficaxy("./imagenes/xypendubotoptimo.png", x, [r'$q_1(t)$', r'$q_2(t)$'])
```



Si deseas compartir este Notebook de IPython, o ver las animaciones y el codigo completo, utiliza la siguiente dirección: http://bit.ly/1H1bR6Q, o bien el siguiente código QR: