
MA4703 Control Óptimo: Teoría y Laboratorio
Informe del Laboratorio 2
Controlabilidad, observabilidad, estabilidad y detectabilidad

Autores: Carlos Antil, Manuel Torres.
Profesor: Héctor Ramírez.
Auxiliar: Javier Madariaga, Pablo Araya.
Fecha del laboratorio: Miércoles 7 de Septiembre, 2022.
Fecha de entrega: Martes 20 de Septiembre, 2022.

Índice general

1. Introducción	2
1. Objetivo	2
2. Preliminares	2
2. Laboratorio	3
1. Parte A: Modelamiendo y simulación	3
1.1. Ejercicio 1	3
1.2. Ejercicio 2	4
2. Parte B: Controlabilidad, observabilidad y estabilidad	7
2.1. Ejercicio 3	7
2.2. Ejercicio 4	12
2.3. Ejercicio 5	13
2.4. Ejercicio 6	14
3. Parte C: Reguladores y estabilizadores	15
3.1. Ejercicio 7	15
3.2. Ejercicio 8	16
3.3. Ejercicio 9	19

Capítulo 1

Introducción

1.- Objetivo

El objetivo de este laboratorio es estudiar la controlabilidad, observabilidad, estabilidad y detectabilidad de un sistema lineal controlado. Para esto, se pide verificar los respectivos criterios de manera directa y usando el paquete *control* de python. También se estudian conceptos relacionados como la *matriz Grammiana* y la *forma canónica de Brunovski*.

2.- Preliminares

Con el fin de que el lector pueda replicar lo presentado en este informe, el preámbulo necesario es el siguiente:

1. Para correr el código adjunto puede ser necesario instalar las librerías *toolbox* y *control*. Para ello basta con correr el siguiente código:

```
1 pip install toolbox
2 pip install control
```

Listing 1.1: Instalación de paquetes

2. Los paquetes a utilizar se importan con el siguiente código:

```
1 import numpy as np
2 from numpy.linalg import eig
3 import sympy as sym
4 import scipy as sp
5 from scipy import integrate
6 from scipy.integrate import solve_ivp
7 import control as cl
8 from control import lqr, place
9 import matplotlib.pyplot as plt
10 import seaborn as sb
11 sb.set_theme(style='darkgrid')
12 sb.set_palette('dark')
```

Listing 1.2: Carga de paquetes

Para complementar los resultados puede revisar el documento *ipynb* con los códigos utilizados, se encuentra adjunto en la entrega con este informe.

Capítulo 2

Laboratorio

A continuación se presenta el problema a desarrollar junto a su análisis y desarrollo, con las respectivas simulaciones realizadas. En algunos casos se adjunta el código utilizado.

1.- Parte A: Modelamiento y simulación

1A.- Ejercicio 1

Un barco carguero debe llevar su cargamento hasta el puerto de Guanzhou ubicado en el sur de China, atravesando el Océano Pacífico. El objetivo de este primer ejercicio es modelar el movimiento del barco como un sistema no-lineal no autónomo:

$$\vec{X}'(t) = F(t, \vec{X}(t)), \quad t \geq 0.$$

Para esto, se seguirán los siguientes supuestos:

1. El par $(x(t), y(t))$ corresponde al vector (abscisa, ordenada) que indica la posición del barco en el océano en un tiempo t .
2. El barco se mueve con rapidez constante igual a V .
3. El mar posee corrientes marinas que dependen solo de la posición del barco $(x(t), y(t))$ y que influyen en su movimiento actuando como una fuerza externa, denotada por $\vec{G}(x, y) = (G_1(x, y), G_2(x, y))^T$.
4. La masa del barco disminuye a medida que este consume combustible. Por simplicidad, se supone que esta disminución de masa es conocida y depende solo del tiempo t , obteniendo así una función denotada $m(t)$.
5. Los efectos del roce o viscosidad son despreciables.
6. La posición del puerto Guangzhou se encuentra en $(0, 0)$.

Se denotará por $\nu(t)$ el ángulo de inclinación del vector de velocidad $\vec{V}(t) = (\dot{x}(t), \dot{y}(t))^T$ del barco, con respecto al eje de las abscisas. Utilizando las relaciones que satisfacen \dot{x} e \dot{y} y escriba un modelo de la forma $\vec{X}'(t) = F(t, \vec{X}(t))$ (para $t \geq 0$) a partir de la segunda ley de Newton

$$\frac{d\vec{P}}{dt} = \vec{F}_{\text{ext}},$$

donde $\vec{P}(t) = m(t)\vec{V}(t)$ es el momentum del barco y $\vec{F}_{\text{ext}}(t)$ corresponde a las fuerzas externas que actúan sobre él (en este caso, las corrientes marinas), donde el lado derecho viene dado (salvo reordenamiento de las variables) por

$$F(t, \vec{X}) = A(t)\vec{X} + \tilde{G}(\vec{X}),$$

con

$$A := \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{\dot{m}(t)}{m(t)} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{\dot{m}(t)}{m(t)} \end{pmatrix}, \quad \text{y} \quad \begin{pmatrix} 0 \\ \frac{G_1(x_1, x_2)}{m(t)} \\ 0 \\ \frac{G_2(x_1, x_2)}{m(t)} \end{pmatrix}.$$

A continuación se identifican las componentes de \vec{X} :

El sistema viene dado por

$$F(t, \vec{X}(t)) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{\dot{m}(t)}{m(t)} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{\dot{m}(t)}{m(t)} \end{pmatrix} \vec{X}(t) + \begin{pmatrix} 0 \\ \frac{G_1(x_1, x_2)}{m(t)} \\ 0 \\ \frac{G_2(x_1, x_2)}{m(t)} \end{pmatrix},$$

en donde por correspondencia:

$$\vec{X}(t) = \begin{pmatrix} x_1(t) \\ \dot{x}_1(t) \\ x_2(t) \\ \dot{x}_2(t) \end{pmatrix}.$$

1B.- Ejercicio 2

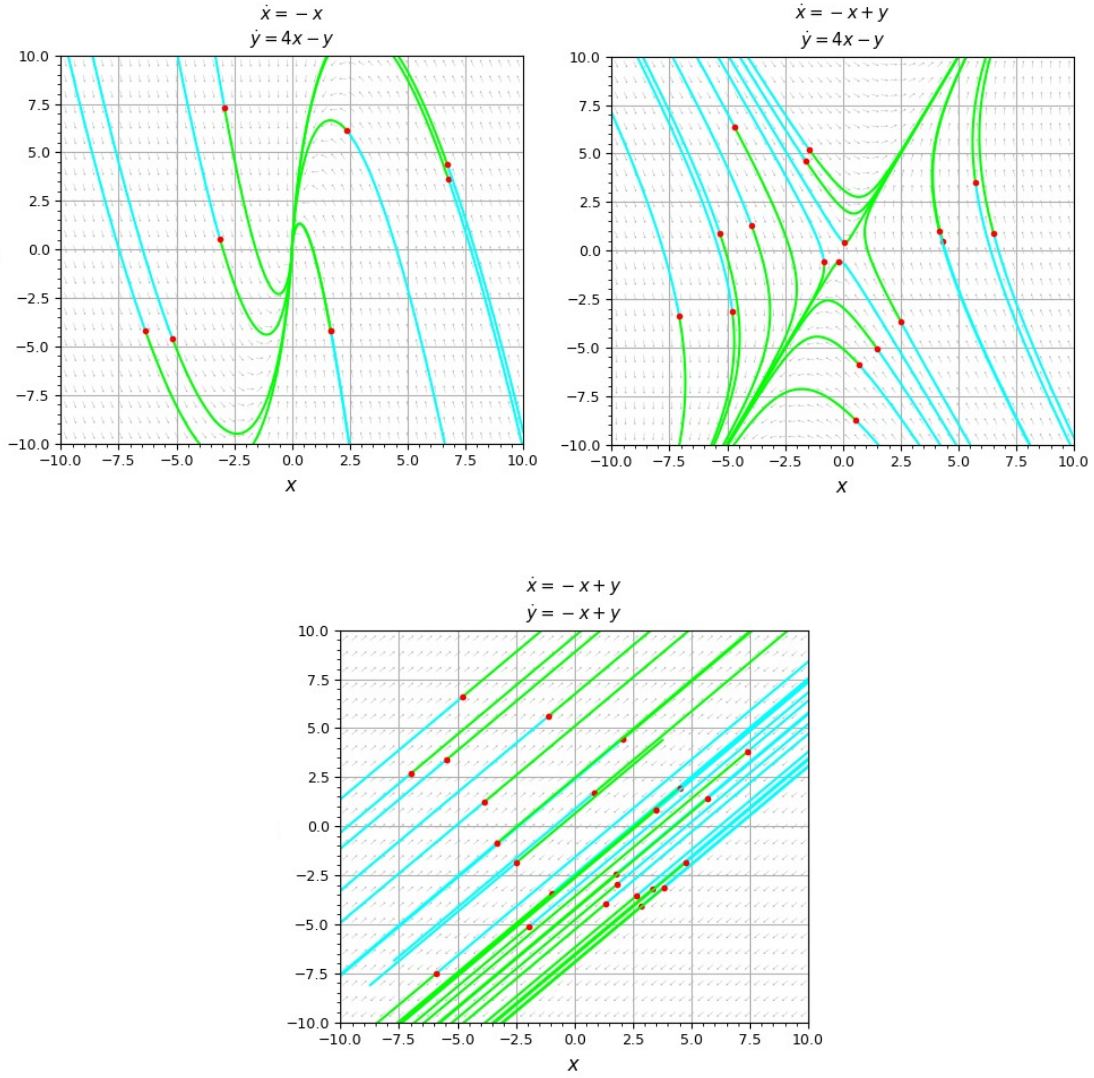
Por simplicidad, en lo que sigue se supondrá que las corrientes marinas vienen dadas por una función G lineal. En este caso las corrientes se ven como giros, tal como se aprecia en la figura adjunta:

1. Utilizando *pyplane* para dibujar los diagramas de fase que representan estas corrientes:

$$\begin{aligned} \dot{x}(t) &= ax(t) + by(t), \\ \dot{y}(t) &= cx(t) + dy(t), \end{aligned}$$

con distintas configuraciones paramétricas. Se reescribe un modelo de la forma $\vec{X}'(t) = F(t, \vec{X}(t))$, $t \geq 0$ pero donde el lado derecho ahora sea una función lineal no homogénea $F(t, \vec{X}) = \tilde{A}(t)\vec{X}$.

Mediante *pyplane* se obtienen los siguientes diagramas de fase que representan corrientes:



Es posible notar que en cualquier caso la corriente diverge, en el tercer gráfico (segunda fila) se observa que la corriente además es estacionaria en el puerto.

A partir de un sistema de la forma

$$(P) \quad F(t, \vec{X}(t)) = \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{\dot{m}(t)}{m(t)} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{\dot{m}(t)}{m(t)} \end{pmatrix} \begin{pmatrix} x_1(t) \\ \dot{x}_1(t) \\ x_2(t) \\ \dot{x}_2(t) \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{G_1(x_1, x_2)}{m(t)} \\ 0 \\ \frac{G_2(x_1, x_2)}{m(t)} \end{pmatrix},$$

Suponiendo que G_1, G_2 son funciones lineales respecto a (x_1, x_2) , luego existen coeficientes $\alpha_1, \alpha_2, \beta_1, \beta_2 \in \mathbb{R}$ tales que

$$\begin{aligned} G_1(x_1, x_2) &= \alpha_1 x_1 + \alpha_2 x_2 \\ G_2(x_1, x_2) &= \beta_1 x_1 + \beta_2 x_2, \end{aligned}$$

entonces el sistema (P) se puede expresar de la forma $F(t, \vec{X}) = \tilde{A}(t)\vec{X}$, en efecto,

$$\begin{aligned}
 F(t, \vec{X}(t)) &= \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{\dot{m}(t)}{m(t)} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{\dot{m}(t)}{m(t)} \end{pmatrix} \begin{pmatrix} x_1(t) \\ \dot{x}_1(t) \\ x_2(t) \\ \dot{x}_2(t) \end{pmatrix} + \begin{pmatrix} 0 \\ \frac{\alpha_1 x_1 + \alpha_2 x_2}{m(t)} \\ 0 \\ \frac{\beta_1 x_1 + \beta_2 x_2}{m(t)} \end{pmatrix} \\
 &= \begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{\dot{m}(t)}{m(t)} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{\dot{m}(t)}{m(t)} \end{pmatrix} \begin{pmatrix} x_1(t) \\ \dot{x}_1(t) \\ x_2(t) \\ \dot{x}_2(t) \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ \frac{\alpha_1}{m(t)} & 0 & \frac{\alpha_2}{m(t)} & 0 \\ 0 & 0 & 0 & 0 \\ \frac{\beta_1}{m(t)} & 0 & \frac{\beta_2}{m(t)} & 0 \end{pmatrix} \begin{pmatrix} x_1(t) \\ \dot{x}_1(t) \\ x_2(t) \\ \dot{x}_2(t) \end{pmatrix} \\
 &= \left(\begin{pmatrix} 0 & 1 & 0 & 0 \\ 0 & -\frac{\dot{m}(t)}{m(t)} & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & -\frac{\dot{m}(t)}{m(t)} \end{pmatrix} + \begin{pmatrix} 0 & 0 & 0 & 0 \\ \frac{\alpha_1}{m(t)} & 0 & \frac{\alpha_2}{m(t)} & 0 \\ 0 & 0 & 0 & 0 \\ \frac{\beta_1}{m(t)} & 0 & \frac{\beta_2}{m(t)} & 0 \end{pmatrix} \right) \begin{pmatrix} x_1(t) \\ \dot{x}_1(t) \\ x_2(t) \\ \dot{x}_2(t) \end{pmatrix} \\
 &= \begin{pmatrix} 0 & 1 & 0 & 0 \\ \frac{\alpha_1}{m(t)} & -\frac{\dot{m}(t)}{m(t)} & \frac{\alpha_2}{m(t)} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{\beta_1}{m(t)} & 0 & \frac{\beta_2}{m(t)} & -\frac{\dot{m}(t)}{m(t)} \end{pmatrix} \begin{pmatrix} x_1(t) \\ \dot{x}_1(t) \\ x_2(t) \\ \dot{x}_2(t) \end{pmatrix},
 \end{aligned}$$

donde

$$\tilde{A} = \begin{pmatrix} 0 & 1 & 0 & 0 \\ \frac{\alpha_1}{m(t)} & -\frac{\dot{m}(t)}{m(t)} & \frac{\alpha_2}{m(t)} & 0 \\ 0 & 0 & 0 & 1 \\ \frac{\beta_1}{m(t)} & 0 & \frac{\beta_2}{m(t)} & -\frac{\dot{m}(t)}{m(t)} \end{pmatrix}.$$

2. Para maniobrar el barco se introduce un motor que produce cambios de aceleración con componentes horizontales y verticales, dados por u y v , respectivamente. A partir del modelo de la parte anterior, se describirá el movimiento del barco como un sistema lineal (no homogéneo) controlado de la forma $\vec{X}'(t) = \tilde{A}(t)\vec{X}(t) + B\vec{U}(t)$, donde $\vec{U}(t) = (u(t), v(t))$.

Considerando un control $\vec{U}(t) = \begin{pmatrix} u(t) \\ v(t) \end{pmatrix}$, luego la matriz B es de la forma

$$B := \begin{pmatrix} 0 & 0 \\ 1 & 0 \\ 0 & 0 \\ 0 & 1 \end{pmatrix},$$

así queda el sistema de la forma

$$\vec{X}'(t) = \tilde{A}(t)\vec{X}(t) + B\vec{U}(t),$$

que es lo deseado.

2.- Parte B: Controlabilidad, observabilidad y estabilidad

En el resto de las preguntas se supondrá que la masa es constante (es decir, $\dot{m}(t) = 0$), así $m(t) = m > 0$ para todo $t \geq 0$. Además se consideraran las siguientes configuraciones de parámetros:

1. $a = -1, b = -2, c = 4, d = -5$.
2. $a = -1, b = 0, c = 4, d = -1$.

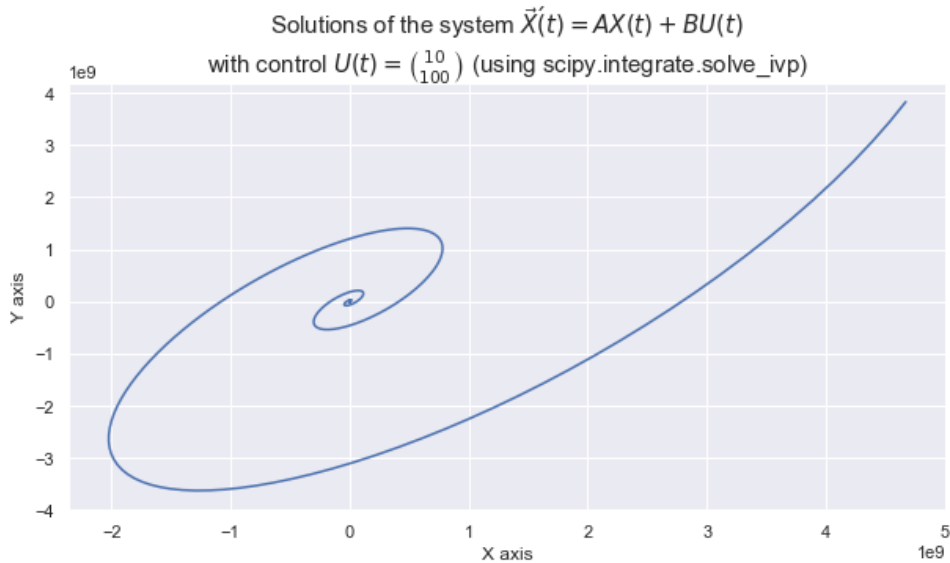
2A.- Ejercicio 3

Utilizando *solve_ivp* se simularán trayectorias del sistema para diferentes controles (constante, sinusoidales, feedbacks, bang-bang, etc) y distintas condiciones iniciales. Se comenta lo observado en cada situación.

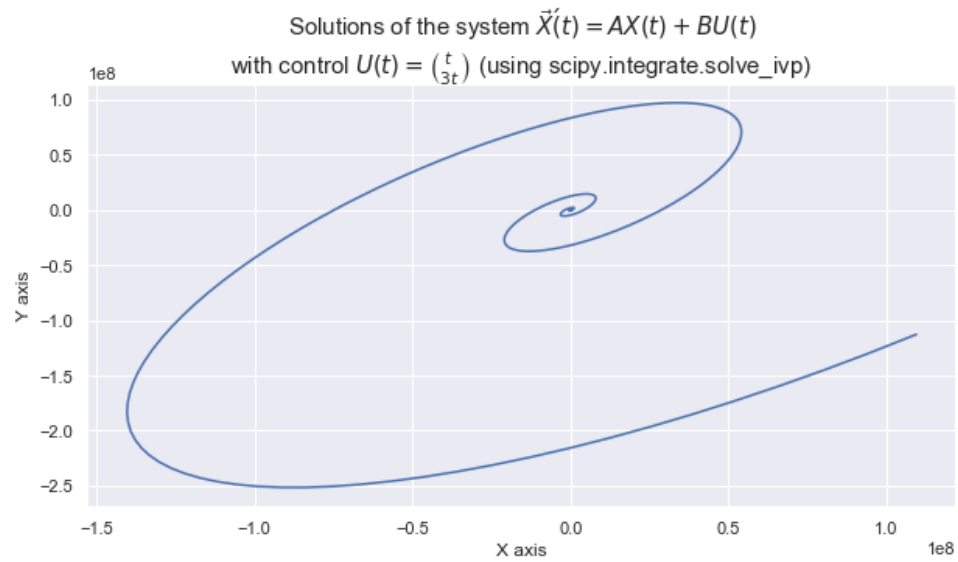
En lo que sigue se considera la masa independiente del tiempo, es decir, tomando $m(t) = 100$ [unids. de masa]. A continuación se estudian las trayectorias en el intervalo de tiempo $[0, T_f]$, con $T_f = 100$ [unids. de tiempo], tomando la condición inicial $\vec{X}(0) = \begin{pmatrix} 1 & 5 & 1 & 5 \end{pmatrix}^T$. Las simulaciones obtenidas son:

1. Usando el primer set de coeficientes: Sean $(a, b, c, d) = (-1, -2, 4, -5)$, luego:

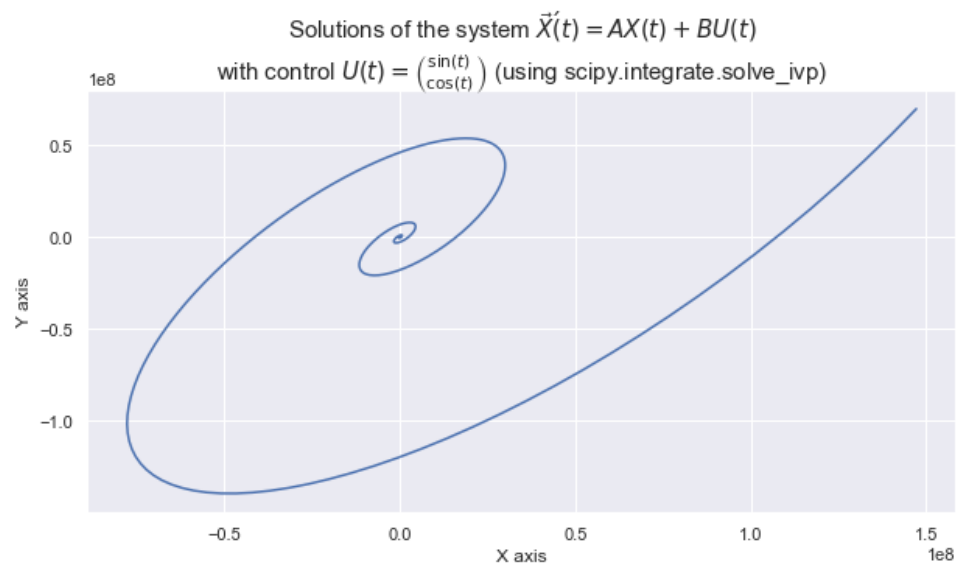
- Considerando el control $U(t) = \begin{pmatrix} 10 \\ 100 \end{pmatrix}$:



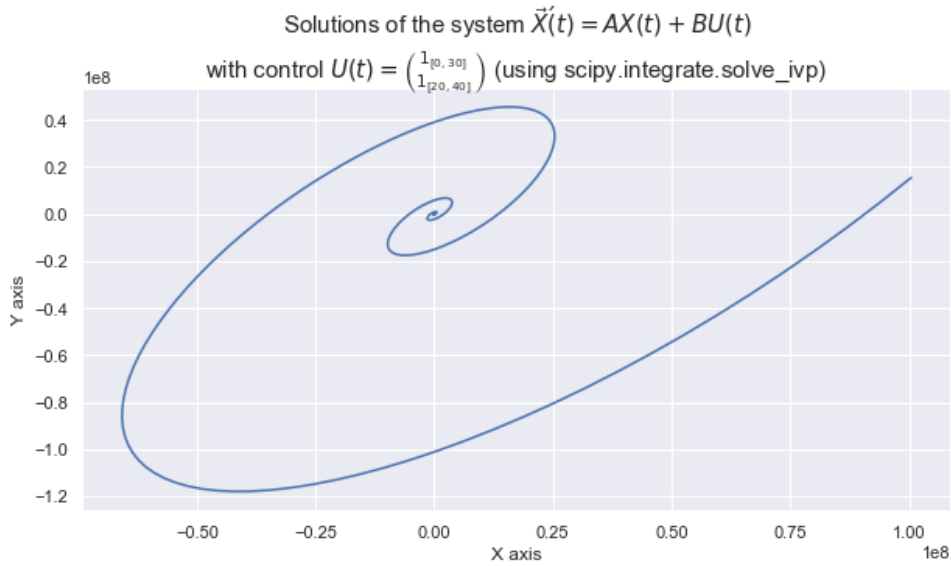
- Considerando el control $U(t) = \begin{pmatrix} t \\ 3t \end{pmatrix}$:



- Considerando el control $\vec{U}(t) = \begin{pmatrix} \sin(t) \\ \cos(t) \end{pmatrix}$:



- Considerando el control $\vec{U}(t) = \begin{pmatrix} 1_{[0,30]} \\ 1_{[20,40]} \end{pmatrix}$:

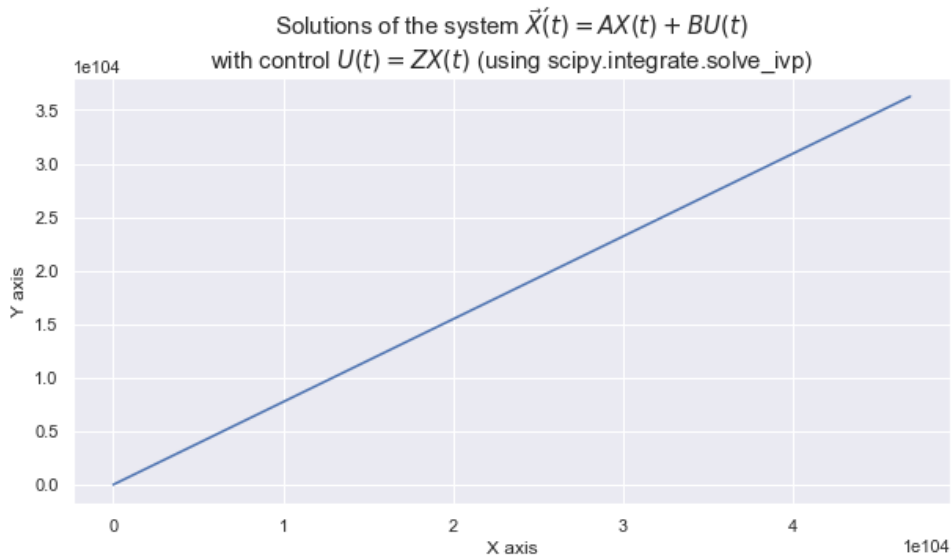


- Considerando el control $U(t) = ZX(t)$, luego se obtiene un sistema simplificado de la forma $\vec{X}'(t) = (A + BZ)X(t)$: Basta notar que al considerar una matriz de la forma $Z := \begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \end{pmatrix}$ luego se tiene que

$$U(t) = \begin{pmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \end{pmatrix} \begin{pmatrix} x(t) \\ \dot{x}(t) \\ y(t) \\ \dot{y}(t) \end{pmatrix} = \begin{pmatrix} z_{11}x(t) + z_{12}\dot{x}(t) + z_{13}y(t) + z_{14}\dot{y}(t) \\ z_{21}x(t) + z_{22}\dot{x}(t) + z_{23}y(t) + z_{24}\dot{y}(t) \end{pmatrix}.$$

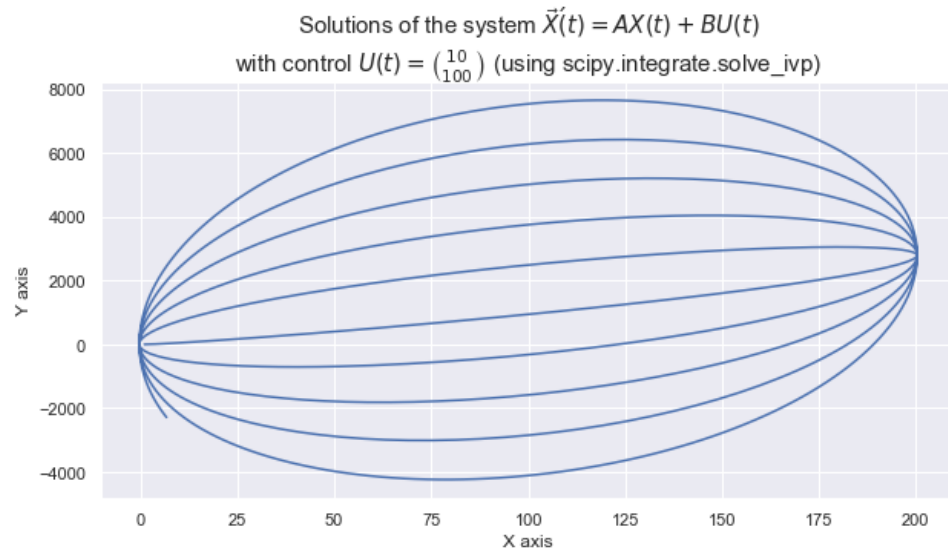
Tomando un caso particular, considerando:

$$U(t) = \begin{pmatrix} x(t) + \dot{x}(t) + y(t) + \dot{y}(t) \\ 5x(t) - \dot{x}(t) + \dot{y}(t) \end{pmatrix}$$

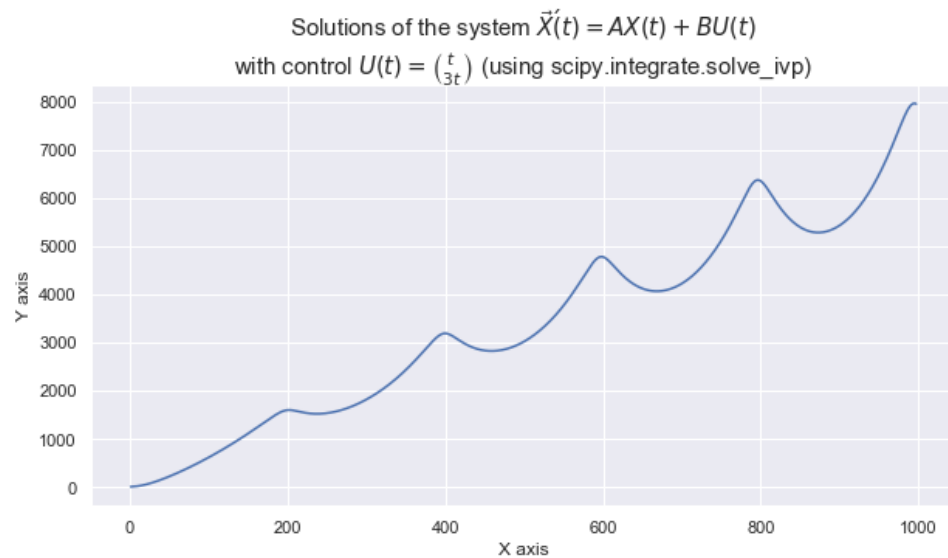


2. Usando el segundo set de coeficientes: Sean $(a, b, c, d) = (-1, 0, 4, -1)$, luego:

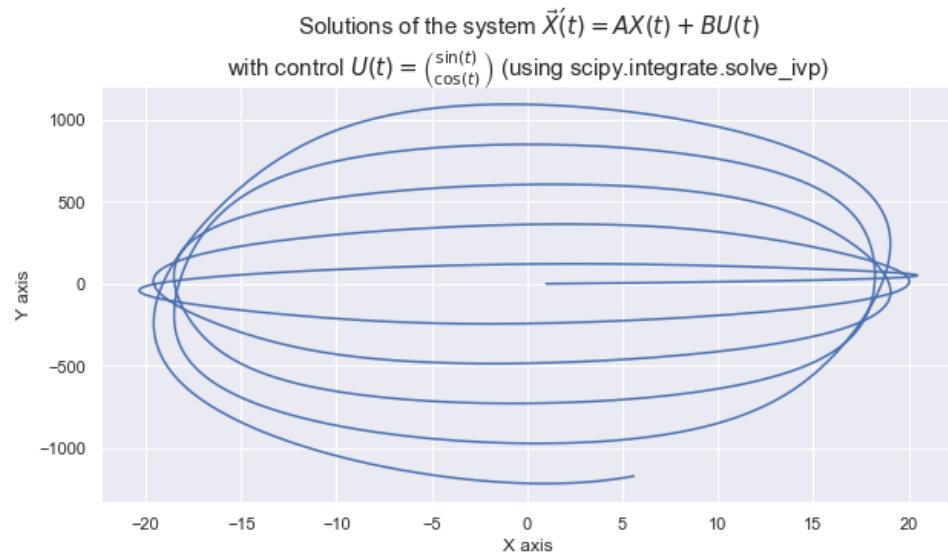
- Considerando el control $U(t) = \begin{pmatrix} 10 \\ 100 \end{pmatrix}$:



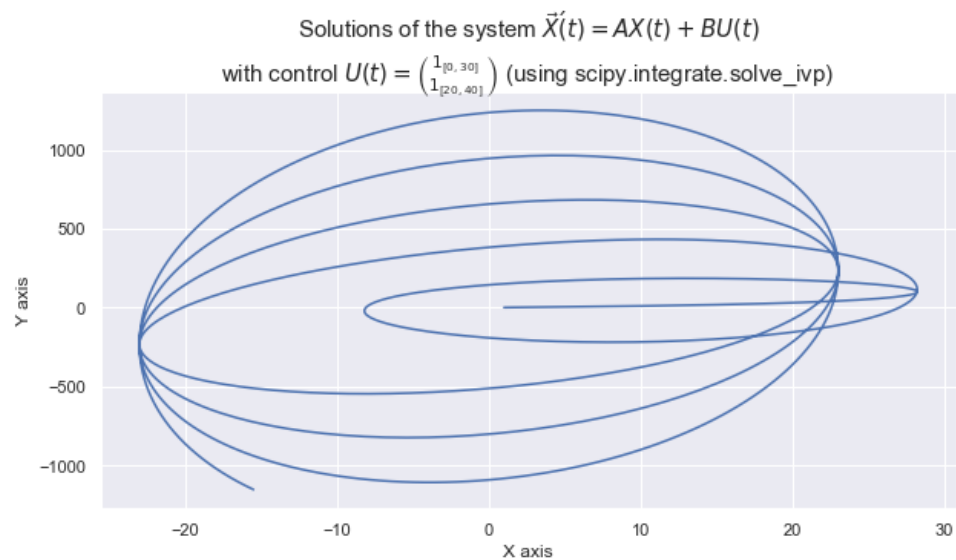
- Considerando el control $U(t) = \begin{pmatrix} t \\ 3t \end{pmatrix}$:



- Considerando el control $U(t) = \begin{pmatrix} \sin(t) \\ \cos(t) \end{pmatrix}$:

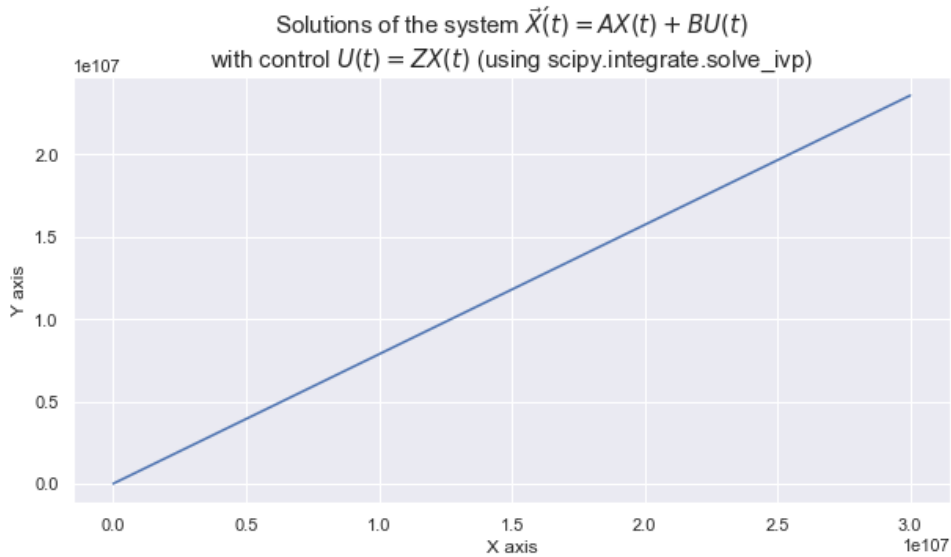


- Considerando el control $U(t) = \begin{pmatrix} 1_{[0,30]} \\ 1_{[20,40]} \end{pmatrix}$:



- Nuevamente, considerando el control *feedback*:

$$U(t) = \begin{pmatrix} x(t) + \dot{x}(t) + y(t) + \dot{y}(t) \\ 5x(t) - \dot{x}(t) + \dot{y}(t) \end{pmatrix}$$



2B.- Ejercicio 4

Utilizando *python* (sin el toolbox de *control*), se calculará la matriz de controlabilidad

$$C = [B, AB, A^2B, \dots, A^{n-1}B],$$

con $A \in \mathbb{R}^{n \times n}$ y $B \in \mathbb{R}^{n \times p}$, para los distintos a, b, c y d dados. Se comparará el resultado con el obtenido usando el comando *ctrb* del toolbox de *control* de *python*. ¿Es el sistema controlable?

Se calcula la matriz de controlabilidad

$$C = [B, AB, A^2B, \dots, A^{n-1}B],$$

utilizando el comando *ctrb* y el script adjunto más adelante, obteniendo en cada caso:

$$C_{\text{lib}} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0,1 & 0,1 \\ 1 & 0 & 0 & 0 & 0,1 & 0,1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0,1 & 0,1 \\ 0 & 1 & 0 & 0 & 0,1 & 0,1 & 0 & 0 \end{pmatrix}$$

y

$$C_{\text{calc}} = \begin{pmatrix} 0 & 0 & 1 & 0 & 0 & 0 & 0,1 & 0,1 \\ 1 & 0 & 0 & 0 & 0,1 & 0,1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0,1 & 0,1 \\ 0 & 1 & 0 & 0 & 0,1 & 0,1 & 0 & 0 \end{pmatrix}$$

respectivamente.

El método utilizado para calcular C_{calc} es:

```
1 def Control(alpha_1, alpha_2, beta_1, beta_2, m):
2     A1=[0,1,0,0]
3     A2=[alpha_1/m, 0, alpha_2 /m, 0]
4     A3=[0,0,0,1]
```

```

5  A4=[beta_1/m, 0, beta_2 /m, 0]
6  A=np.array([A1,A2,A3,A4])
7  B=np.array([[0,0],[1,0],[0,0],[0,1]])
8
9  m0=B
10 m1= np.matmul(A,B)
11 m2=np.matmul(np.matmul(A,A),B)
12 m3=np.matmul(np.matmul(np.matmul(A,A),A),B)
13
14 return [A,B,np.concatenate([m0,m1,m2,m3], axis=1)]

```

Listing 2.1: Método para calcular la matriz de controlabilidad

2C.- Ejercicio 5

Suponiendo ahora que es posible observar la evolución del sistema según el modelo $\tilde{Y} = C\tilde{X}$, con C de la forma

$$C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}.$$

Con lo anterior, el sistema queda de la forma:

$$\begin{cases} \vec{X}' = A\vec{X} + B\vec{U} \\ \vec{Y} = C\vec{X} \end{cases}$$

Utilizando *python* (sin el toolbox de *control*), se calculará la matriz de observabilidad

$$O = [C; CA; CA^2; \dots; CA^{n-1}],$$

con $A \in \mathbb{R}^{n \times n}$ y $C \in \mathbb{R}^{q \times n}$, para los sistemas de los ejercicios precedentes. Se compara el resultado con el obtenido usando el comando *obsv* del toolbox de *control* de *python*. La pregunta a responder es: ¿Es acaso el sistema observable?

Los resultados obtenidos son:

$$O_{\text{calc}} = \begin{pmatrix} 1. & 0. & 0. & 0. \\ 0. & 0. & 1. & 0. \\ 0. & 1. & 0. & 0. \\ 0. & 0. & 0. & 1. \\ 0,1 & 0. & 0,1 & 0. \\ 0,1 & 0. & 0,1 & 0. \\ 0. & 0,1 & 0. & 0,1 \\ 0. & 0,1 & 0. & 0,1 \end{pmatrix}$$

y

$$O_{\text{lib}} = \begin{pmatrix} 1. & 0. & 0. & 0. \\ 0. & 0. & 1. & 0. \\ 0. & 1. & 0. & 0. \\ 0. & 0. & 0. & 1. \\ 0,1 & 0. & 0,1 & 0. \\ 0,1 & 0. & 0,1 & 0. \\ 0. & 0,1 & 0. & 0,1 \\ 0. & 0,1 & 0. & 0,1 \end{pmatrix}.$$

Para calcular O_{calc} se implementó el siguiente código

```

1 def Observar(alpha_1, alpha_2, beta_1, beta_2, m):
2     A1=[0,1,0,0]
3     A2=[alpha_1/m, 0, alpha_2 /m, 0]
4     A3=[0,0,0,1]
5     A4=[beta_1/m, 0, beta_2 /m, 0]
6     A=np.array([A1,A2,A3,A4])
7     C=np.array([[1,0,0,0],[0,0,1,0]])
8
9     m0=C
10    m1= np.matmul(C,A)
11    m2=np.matmul(C,np.matmul(A,A))
12    m3=np.matmul(C,np.matmul(np.matmul(A,A),A))
13
14    return [A,C,np.concatenate([m0,m1,m2,m3])]
```

Listing 2.2: Cálculos del ejercicio 5

2D.- Ejercicio 6

Utilizando el comando *gram* se calcula el *Grammiano* para los sistemas de los ejercicios precedentes.

Para calcular se implementó el siguiente código

```

1 A=control[0]
2 B=control[1]
3 C=observar[1]
4 sys1 = cl.ss(A,B,C,0)
5 Wc = cl.gram(sys1, 'o')
6
7 A=control[0]
8 B=control[1]
9 C=observar[1]
10 sys1 = cl.ss(A,B,C,0)
11 Wc = cl.gram(sys1, 'c')
```

Listing 2.3: Cálculos del ejercicio 6

Se calcula la matriz *Grammiana* asociada al sistema

$$G(t) = \int_0^t X(s)^{-1} B(s) B(s)^T X(s)^{-T} ds.$$

Resulta que el método utilizado en ambos casos arroja **Oops, the system is unstable!**, lo que refiere a que los sistemas no son controlables (véase el código en Jupiter adjunto en html o ipynb).

3.- Parte C: Reguladores y estabilizadores

3A.- Ejercicio 7

Considerando $\vec{U} = -K\vec{X}$ para una matriz K apropiada, el sistema

$$\vec{X}' = A\vec{X} + B\vec{U}$$

se puede escribir como

$$\vec{X}' = (A - BK)\vec{X}.$$

Utilizando el comando *place* para obtener una matriz de ganancia K tal que $A - BK$ sea asintóticamente estable, luego utilizando el comando *lqr* para obtener una matriz de ganancia K tal que $A - BK$ sea estable y se verifica usando el comando *eig* que la matriz $A - BK$ es asintóticamente estable. Se compararán los resultados obtenidos al utilizar ambas rutinas y se presentan simulaciones de las trayectorias obtenidas por estos controles para distintas condiciones iniciales.

Para calcular se implementó el siguiente código

```
1 #Matriz de Ganancia K que sea GAE
2 p=np.array([-0.2,-3,-0.4,-4])
3 K_place=place(A,B,p)
4
5 #Matriz de Ganancia K que sea Estable
6 I=np.array([[1,0,0,0],[0,1,0,0],[0,0,1,0],[0,0,0,1]])
7 R =np.array([[1,0],[0,1]])
8 K_lqr,q,w=lqr(A,B,I,R)
9
10 # Valores propios (informacion)
11 print('Valores propios')
12 print(eig(A-np.matmul(B,K_lqr))[0])
```

Listing 2.4: Cálculos del ejercicio 7

La matriz de ganancia calculada en cada caso es:

$$K_{\text{place}} = \begin{pmatrix} 1,53411834 & 4,20094201 & -0,27197437 & -0,44636925 \\ -0,27197437 & -0,44636925 & 0,86588166 & 3,39905799 \end{pmatrix}$$

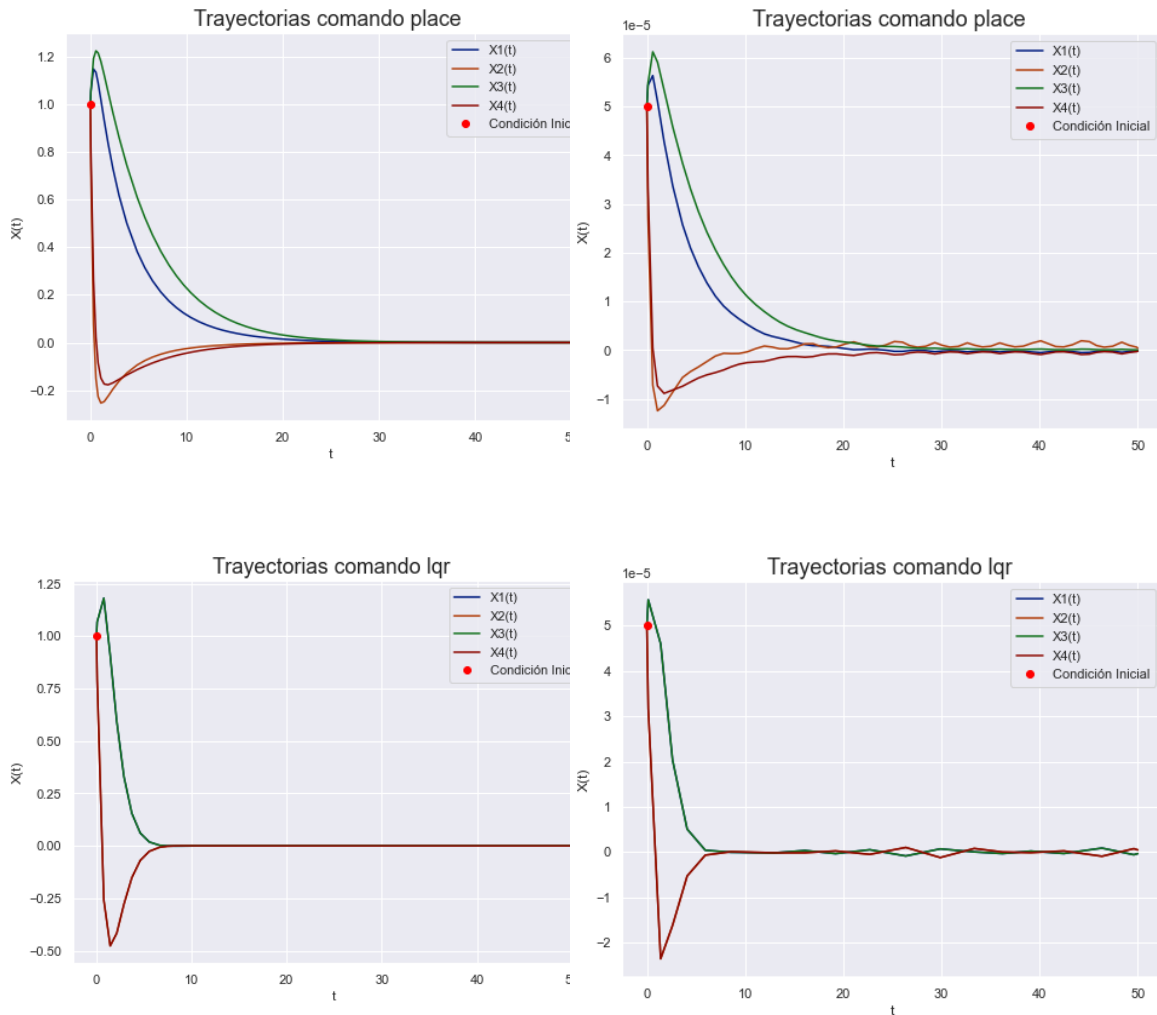
$$K_{\text{lqr}} = \begin{pmatrix} 1,10990195 & 1,79333439 & 0,10990195 & 0,06128358 \\ 0,10990195 & 0,06128358 & 1,10990195 & 1,79333439 \end{pmatrix},$$

y el conjunto de valores propios es:

$$\text{vp} = \left\{ \begin{array}{l} -0,92730898 + 0,39987742i, \\ -0,92730898 - 0,39987742i, \\ -0,8660254 + 0,5i, \\ -0,8660254 - 0,5i \end{array} \right\}.$$

Es posible notar que todos los valores propios obtenidos son complejos, pero con parte real negativa. Por lo tanto, $A - BK$ es una matriz Hurwitz, lo que implica que es asintóticamente estable. Comparando resultados, con el primer comando es necesario darle valores propios con parte real negativa para encontrar una matriz K tal que $A - BK$ sea Hurwitz. Para el segundo comando basta entregarle la matriz A y B con tal de construir una matriz K que cumpla con lo deseado. Por ende, para ambas construcciones sigue que $A - BK$ es asintóticamente estable.

Las simulaciones obtenidas son:



3B.- Ejercicio 8

Usualmente es difícil conocer completamente las variables de estado ya que solo es posible obtener observaciones imprecisas de estas. Por ejemplo, si se controla el barco desde Chile, vía satélite, no es posible saber siempre su velocidad exacta y solo es posible estimarla, por ejemplo, usando dispositivos GPS. Por este motivo, en lo que sigue, se supondrá que solamente se observa la posición del barco (x_1, x_3) pero no su velocidad. Esto lleva a considerar un observador de la forma

$$\dot{\vec{Y}} = C\vec{X}.$$

Se sigue el siguiente esquema:

1. Identificar C e \vec{Y} .
2. Definir el estimador $\hat{\vec{X}}$ como la solución del siguiente sistema:

$$\hat{\vec{X}}'(t) = A\hat{\vec{X}}(t) + BU(t) + L(\vec{Y}(t) - C\hat{\vec{X}}),$$

donde el estado inicial $\hat{\vec{X}}(0) = \hat{\vec{X}}_0$ es conocido y dado en este trabajo, y la matriz L representa la ganancia de la diferencia entre la observación y la estimación del estado.

3. Se simula el estimador obtenido bajo los controles estudiados antes.

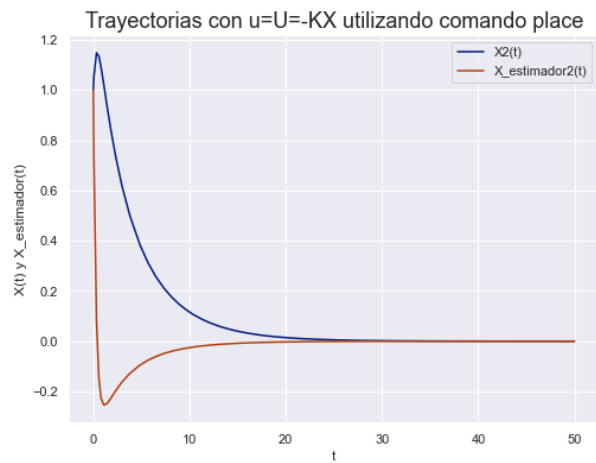
Con la información del problema es posible obtener una observación donde solo se ven x_1 y x_3 , luego la matriz C nos queda:

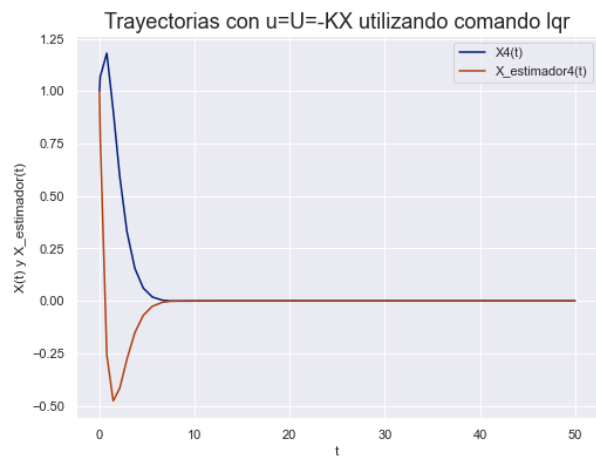
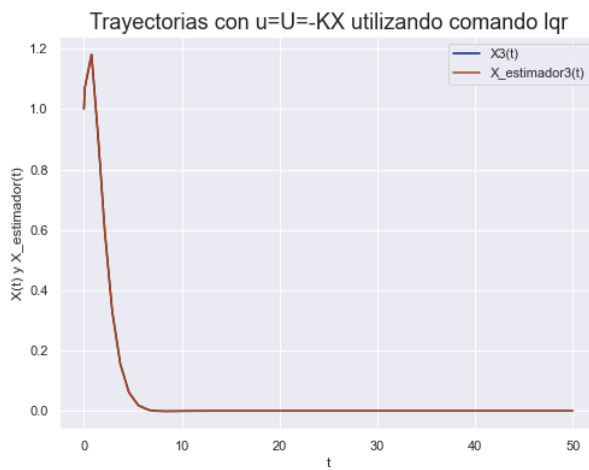
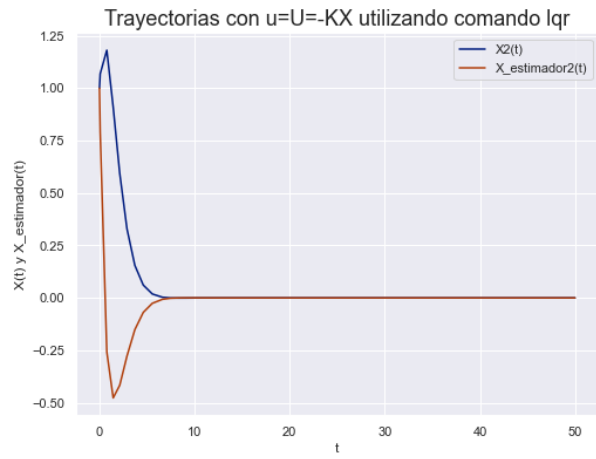
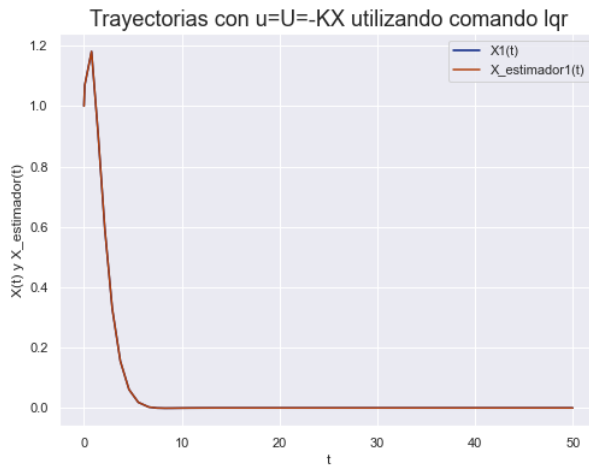
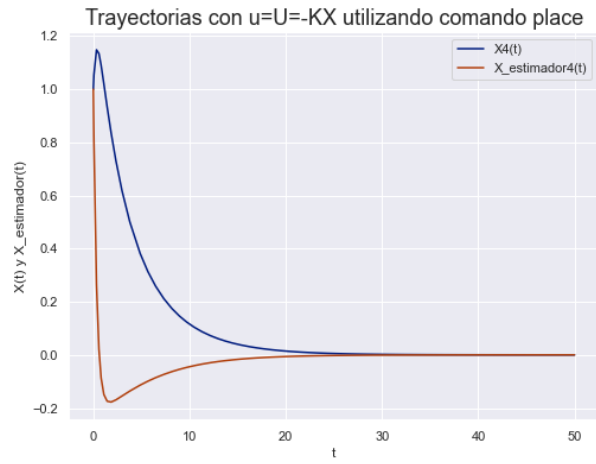
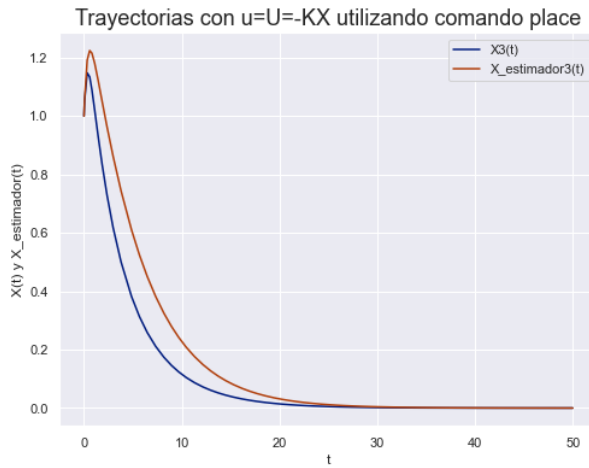
$$C = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix}$$

con $Y = (x_1, x_3)$.

1. Para el comando *place* creamos una matriz L tal que $A - LC$ tiene valores propios negativos, por ende, $A - LC$ es Hurwitz y esto implica que $e(t) \rightarrow 0$, para $t \rightarrow \infty$.
2. Para el comando *lqr* se encuentra una matriz L que tiene valores propios con parte real negativa, por ende, $A - LC$ es Hurwitz y esto implica que $e(t) \rightarrow 0$, para $t \rightarrow \infty$.

A continuación se adjuntan los gráficos obtenidos de las simulaciones utilizando controles del *ejercicio 7*:





Notamos que al agregar un observador podemos estimar a la trayectoria X y al graficar ambas funciones, se puede obtener que $e_X = \hat{X} - X \rightarrow 0$, pues cada $X_i \rightarrow \hat{X}_i$ (algunos en un

tiempo antes que otros), lo que coincide con los cálculos obtenidos en la creación de matrices Hurwitz.

3C.- Ejercicio 9

Utilizando un control regulador feedback lineal, asumiendo que solo se tiene acceso al estimador de \vec{X} , y suponiendo que $\vec{Y} = C\vec{X}$, se deducen las ecuaciones

$$\begin{cases} \hat{\vec{X}}'(t) = A\hat{\vec{X}}(t) + B\vec{U}(t) + L(\vec{Y}(t) - C\hat{\vec{X}}) \\ \vec{e}'(t) = (A - LC)\vec{e}(t), \end{cases}$$

y un sistema de la forma:

$$\begin{pmatrix} \hat{\vec{X}} \\ \vec{e} \end{pmatrix}' = \hat{A}(A, B, C, K, L) \begin{pmatrix} \hat{\vec{X}} \\ \vec{e} \end{pmatrix},$$

se encuentran matrices K y L que hacen a las matrices $A - BK$ y $A - LC$ estables y por lo tanto la matriz $\hat{A}(B, C, K, L)$. Además se simula lo obtenido.

Considerando $U = -KX$, $Y = CX$ y las ecuaciones dadas, se tiene que:

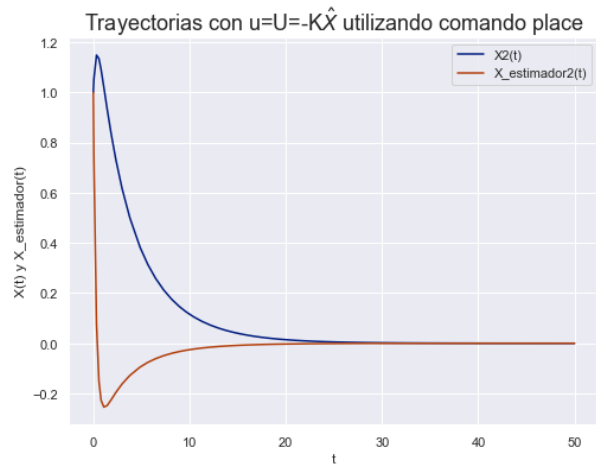
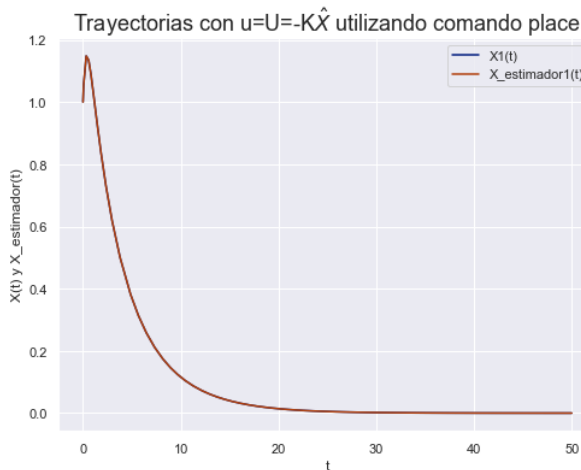
$$\hat{X}' = A\hat{X} - BKX + LC\vec{e},$$

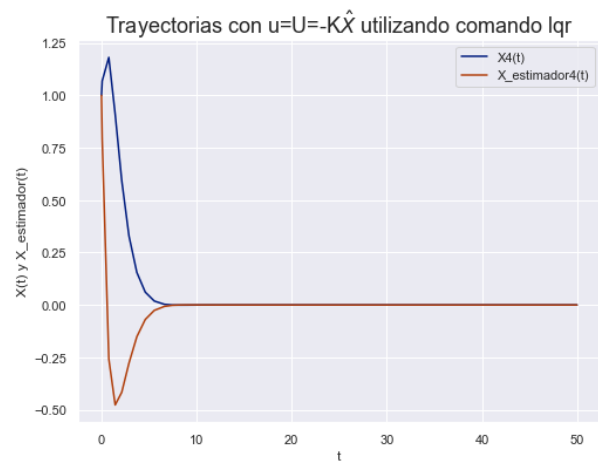
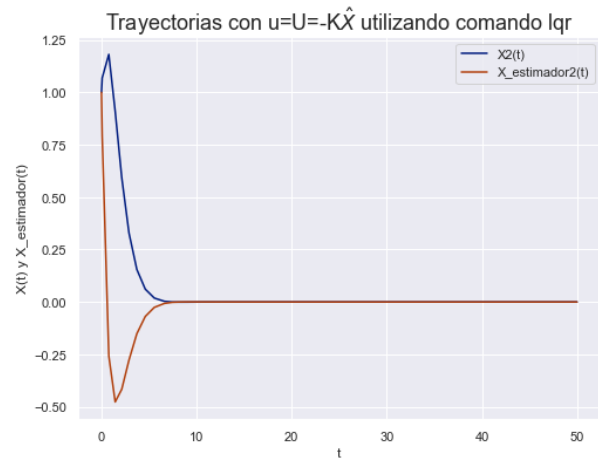
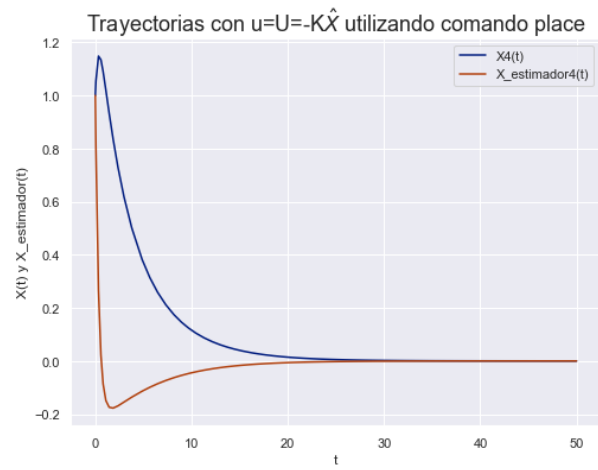
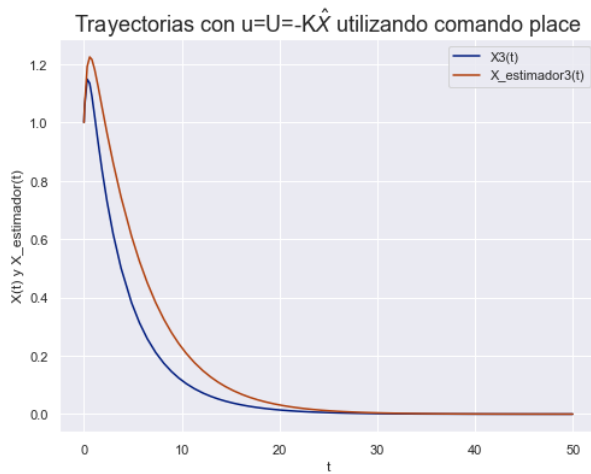
donde $\vec{e} = (X - \hat{X})$ (que es lo conocido). Por lo tanto, sumando $\pm BK\hat{X}$ a la ecuación anterior se obtiene

$$\begin{pmatrix} \hat{X} \\ \vec{e} \end{pmatrix}' = \begin{pmatrix} A - BK & LC - BK \\ 0 & A - LC \end{pmatrix} \begin{pmatrix} \hat{X} \\ \vec{e} \end{pmatrix}$$

La matriz obtenida es asintóticamente estable, ya que sus valores propios dependerán de los valores propios de A_BK y A_LC , cuyas matrices ya son asintóticamente estables.

Las simulaciones obtenidas son:





Es posible notar que para ambos métodos (*place* y *lqr*) las trayectorias convergen al origen. Pues al graficar cada variable X_i en función de su estimación, se puede observar que convergen

al 0, esto es independiente del punto inicial. Dado que cada coordenada converge a 0, entonces la trayectoria X converge al origen. Además, la solución es asintóticamente estable, ya que $e_X = \hat{X} - X$ tiende a 0.