Clase 5: Forma diagonal y de Jordan

Introducción

En la clase teórica vimos que podemos representar un mismo sistema con infinitas descripciones en variables de estado, y una de las formas de pasar de una a otra forma es a través de cambios de base.

En definitiva, el cambio de base nos da como herramienta una matriz inversible que nos permite convertir un vector de estado x a otro nuevo $\hat{x}$. Usamos:

$$\hat{x} = Tx$$

para transformar a la base de $\hat{x}$, y:

$$x = T^{-1}\hat{x}$$

en el sentido inverso.

Partiendo del sistema en variables de estado:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ y &= Cx + Du \end{aligned}$$

Podemos obtener la ecuación de estado para el sistema en la nueva base:

$$\begin{aligned} \dot{\hat{x}} &= T\dot{x} = T\left(Ax + Bu\right) \\ \dot{\hat{x}} &= T\left(AT^{-1}\hat{x} + Bu\right) \\ \dot{\hat{x}} &= \underbrace{TAT^{-1}}_{\hat{A}}\hat{x} + \underbrace{TB}_{\hat{B}}u \end{aligned}$$

y la ecuación de salida:

$$y = \underbrace{CT^{-1}}_{\hat{C}}\hat{x} + Du$$

En resumen, en la nueva base queda:

$$\begin{aligned} \dot{\hat{x}} &= \hat{A}\hat{x} + \hat{B}u \\ y &= \hat{C}\hat{x} + Du \end{aligned}$$

En donde tiene sentido que D no cambie. **Para pensar:** ¿Por qué?

En la clase anterior buscaron una T que les permitía pasar de un modelo con variables de estado de fase, al otro modelo con variables de estado asociadas a los elementos acumuladores.

Nos preguntamos entonces a qué otro tipo de modelo en variable de estado podemos llegar mediante una transformación de este tipo, que tenga una forma simple y conveniente.

Forma diagonal

Introducción intuitiva

Dada una matriz cuadrada $A \in \mathcal{R}^{n\times n}$ definimos autovalor λ_i asociado al autovector v_i a aquel tal que:

$$Av_i = \lambda_i v_i$$

Esto implica que no produce un cambio de dirección del vector v_i sino que cambia su módulo y quizás su sentido, para autovalores reales. ¿Cómo aplica esto intuitivamente para sistemas en variables de estado?

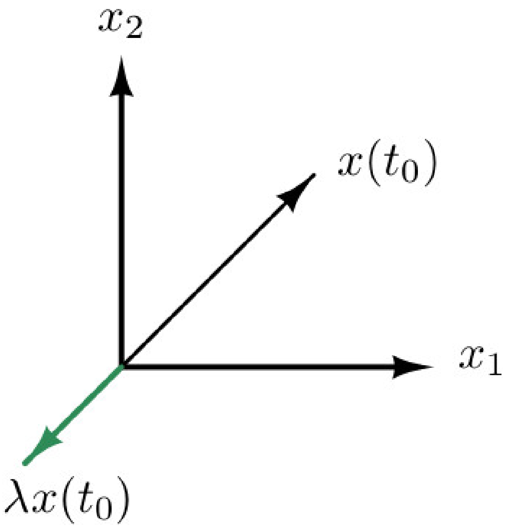
Supongamos que tenemos un sistema sin entrada, de dos variables de estado $x = \begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$:

$$\dot{x}(t) = Ax(t)$$

Y sabemos que la condición inicial $x(t_0)$ es autovector de A. Eso significa que:

$$\dot{x}(t_0) = Ax(t_0) = \lambda x(t_0)$$

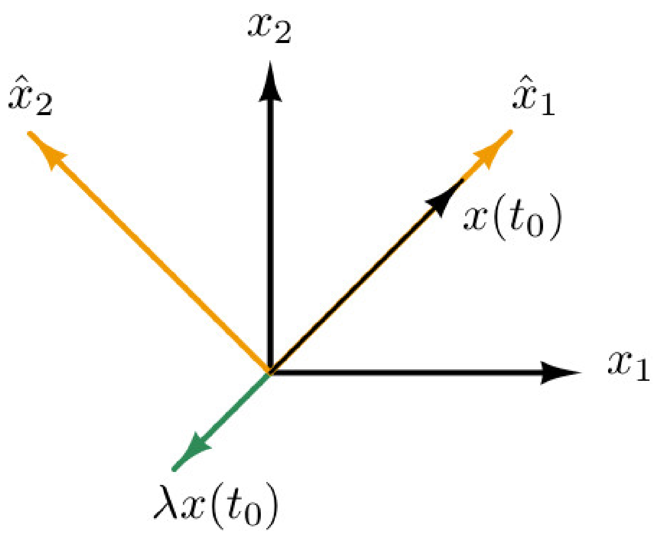
Gráficamente:



En un instante, ¿hacia dónde se desplazará el sistema si la condición inicial es autovector?

Vemos que x(t) tendería a evolucionar dentro de la recta definida por $x(t_0)$ porque la derivada en ese punto mantendría la dirección. El sentido dependerá del signo de λ, pero no nos preocupemos ahora por eso.

Una forma de ver cómo podría simplificarse la respuesta del sistema es si pensamos en definir la dirección del autovector de A, como elemento de la base del vector de estado nuevo $\hat{x}$. Gráficamente sería:



Desplazamiento sobre el autoespacio en nueva base

Esta nueva definición de base del sistema hace que la respuesta a la condición inicial $x(t_0)$ quede confinada sobre la variable de estado $\hat{x}_1$ y no aparezca componente de la misma en $\hat{x}_2$.

Base de autovectores

Para el caso que la matriz A tenga n autovalores distintos, tenemos n autovectores distintos, es decir:

$$Av_i = \lambda_i v_i,\ i = 1, \ldots, n$$

Y $\{v_i\}$, $i = 1, \ldots, n$ es base de autovectores. Como vimos en la clase teórica:

$$x = \sum_{i=1}^{n} \alpha_i v_i = \underbrace{\left[\begin{array}{c|c|c|c} v_1 & v_2 & \cdots & v_n \end{array}\right]}_{M} \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_n \end{bmatrix} = M\hat{x}$$

Donde M es la matriz modal. Veamos cómo nos deja el sistema en la variable $\hat{x}$:

$$\begin{aligned} \dot{\hat{x}} &= M^{-1}AM\hat{x} + M^{-1}Bu \\ y &= CM\hat{x} + Du \end{aligned}$$

¿Qué forma tiene $M^{-1}AM$? A continuación lo descubrimos:

$$AM = A\left[\begin{array}{c|c|c|c} v_1 & v_2 & \cdots & v_n \end{array}\right] = \left[\begin{array}{c|c|c|c} Av_1 & Av_2 & \cdots & Av_n \end{array}\right]$$

Por ser autovectores:

$$AM = \left[\begin{array}{c|c|c|c} \lambda_1 v_1 & \lambda_2 v_2 & \cdots & \lambda_n v_n \end{array}\right]$$

Se puede descomponer en el producto:

$$AM = \left[\begin{array}{c|c|c|c} v_1 & v_2 & \cdots & v_n \end{array}\right] \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix} = M\Lambda$$

Multiplicando por izquierda por M^{-1} se obtiene:

$$\hat{A} = M^{-1}AM = \Lambda = \begin{bmatrix} \lambda_1 & 0 & \cdots & 0 \\ 0 & \lambda_2 & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \cdots & \lambda_n \end{bmatrix}$$

Nos da una matriz de estado diagonal. ¿Qué implicancias tiene sobre la ecuación de estado?

Si $u = 0$:

$$\dot{\hat{x}} = \Lambda \hat{x}$$

es un sistema de ecuaciones diferenciales de primer orden con sus variables desacopladas:

$$\begin{aligned} \dot{\hat{x_1}} &= \lambda_1 \hat{x_1} \\ \dot{\hat{x_2}} &= \lambda_2 \hat{x_2} \\ &\vdots \\ \dot{\hat{x_n}} &= \lambda_n \hat{x_n} \end{aligned}$$

Con soluciones, para $\hat{x}(t_0) = \begin{bmatrix} \alpha_{10} & \alpha_{20} & \cdots & \alpha_{n0} \end{bmatrix}^T$:

$$\begin{aligned} \hat{x_1}(t) &= \alpha_{10} e^{\lambda_1 t} \\ \hat{x_2}(t) &= \alpha_{20} e^{\lambda_2 t} \\ &\vdots \\ \hat{x_n}(t) &= \alpha_{n0} e^{\lambda_n t} \end{aligned}$$

para $t \geq t_0$.

Decimos que los modos del sistema quedan desacoplados y que la salida finalmente queda como combinación lineal de dichos modos:

$$y = \hat{C}\hat{x} + Du$$

A continuación un ejemplo numérico de la guía.

Ejemplo:

Ejercicio 2

$$A_1 = \begin{bmatrix} 1 & 4 & 10 \\ 0 & 2 & 0 \\ 0 & 0 & 3 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, D = 0$$

a) Hallar la forma de Jordan

Aclaración: Podemos llamar forma de Jordan de manera general incluyendo el caso en que la matriz es diagonalizable.

El procedimiento es:

1. Escribir el polinomio característico $p(\lambda) = det(A - \lambda I)$.
2. Hallar las raíces de $p(\lambda)$, que son los autovalores.
3. Buscar en el espacio nulo de $A - \lambda_i I$ los autovectores v_i asociados a cada λ_i. Estos son los vectores que cumplen:
$$(A - \lambda_i I)v_i = 0$$
4. Construir la matriz M con el conjunto de autovectores v_i hallados como columnas.

Como este es un caso simple, lo resolvemos directo con Python para ver cómo usar las herramientas disponibles:

```
# Ejemplo con sympy
!pip install control
import sympy as symb
import control as ctrl
```

```
Requirement already satisfied: control in /usr/local/lib/python3.7/dist-packages (0.9.0)
Requirement already satisfied: scipy in /usr/local/lib/python3.7/dist-packages (from control) (1.4.1)
Requirement already satisfied: matplotlib in /usr/local/lib/python3.7/dist-packages (from control) (3.2.2)
Requirement already satisfied: numpy in /usr/local/lib/python3.7/dist-packages (from control) (1.19.5)
Requirement already satisfied: cycler>=0.10 in /usr/local/lib/python3.7/dist-packages (from matplotlib->contro
Requirement already satisfied: python-dateutil>=2.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib
Requirement already satisfied: pyparsing!=2.0.4,!=2.1.2,!=2.1.6,>=2.0.1 in /usr/local/lib/python3.7/dist-packa
Requirement already satisfied: kiwisolver>=1.0.1 in /usr/local/lib/python3.7/dist-packages (from matplotlib->c
Requirement already satisfied: six in /usr/local/lib/python3.7/dist-packages (from cycler>=0.10->matplotlib->c
```

```
A = symb.Matrix([[1, 4, 10], [0, 2, 0], [0, 0 ,3]])
B = symb.Matrix([[1], [1], [1]])
C = symb.Matrix([[1, 0, 0]])
D = symb.Matrix([0])
print('\n A =')
symb.pprint(A)

print('\n Método manual:')
l = symb.symbols('l')
p = A.charpoly(l)
print('\n Polinomio característico p(l)=')
symb.pprint(symb.factor(p))

n = A.rows
I = symb.eye(n)
# Manualmente:
ecuacion = A-p.root(0)*I
v1 = ecuacion.nullspace()
M = symb.Matrix(v1)

ecuacion = A-p.root(1)*I
v2 = ecuacion.nullspace()
M = M.row_join(symb.Matrix(v2))

ecuacion = A-p.root(2)*I
v3 = ecuacion.nullspace()
M = M.row_join(symb.Matrix(v3))

print('\n M =')
symb.pprint(M)

A_hat = M.inv()*A*M
B_hat = M.inv()*B
C_hat = C*M
print('\n A_hat =')
symb.pprint(A_hat)
print('\n B_hat =')
symb.pprint(B_hat)
print('\n C_hat =')
symb.pprint(C_hat)

# Se pueden usar directo para definir el sistema
# Dependiendo de la versión, puede que no se necesiten los .tolist()
sistema_hat = ctrl.ss(A_hat.tolist(), B_hat.tolist(), C_hat.tolist(), D.tolist())
```

```
 A =
⎡1  4  10⎤
⎢        ⎥
⎢0  2  0 ⎥
⎢        ⎥
⎣0  0  3 ⎦

 Método manual:

 Polinomio característico p(l)=
PurePoly(l**3 - 6*l**2 + 11*l - 6, l, domain='ZZ')
```

```
 M =
⎡1  4  5⎤
⎢       ⎥
⎢0  1  0⎥
⎢       ⎥
⎣0  0  1⎦

 A_hat =
⎡1  0  0⎤
⎢       ⎥
⎢0  2  0⎥
⎢       ⎥
⎣0  0  3⎦

 B_hat =
⎡-8⎤
⎢  ⎥
⎢1 ⎥
⎢  ⎥
⎣1 ⎦

 C_hat =
[1  4  5]
```

```
print('\n Otras funciones:')
autovals = A.eigenvals()
print('\n Autovalores: \n')
symb.pprint(autovals)
autovects = A.eigenvects()
print('\n Autovectores: \n')
symb.pprint(autovects)

print('\n Método directo:')
M, D = A.diagonalize()
print('\n M =')
symb.pprint(M)
print('\n D =')
symb.pprint(D)
```

```
--NORMAL--

 Otras funciones:

 Autovalores:

{1: 1, 2: 1, 3: 1}

 Autovectores:

⎡⎛        ⎡⎡1⎤⎤⎞  ⎛        ⎡⎡4⎤⎤⎞  ⎛        ⎡⎡5⎤⎤⎞⎤
⎢⎜        ⎢⎢ ⎥⎥⎟  ⎜        ⎢⎢ ⎥⎥⎟  ⎜        ⎢⎢ ⎥⎥⎟⎥
⎢⎜1, 1, ⎢⎢0⎥⎥⎟, ⎜2, 1, ⎢⎢1⎥⎥⎟, ⎜3, 1, ⎢⎢0⎥⎥⎟⎥
⎢⎜        ⎢⎢ ⎥⎥⎟  ⎜        ⎢⎢ ⎥⎥⎟  ⎜        ⎢⎢ ⎥⎥⎟⎥
⎣⎝        ⎣⎣0⎦⎦⎠  ⎝        ⎣⎣0⎦⎦⎠  ⎝        ⎣⎣1⎦⎦⎠⎦

 Método directo:

 M =
⎡1  4  5⎤
⎢       ⎥
⎢0  1  0⎥
⎢       ⎥
⎣0  0  1⎦

 D =
⎡1  0  0⎤
⎢       ⎥
⎢0  2  0⎥
⎢       ⎥
⎣0  0  3⎦
```

```
# Ejemplo directo con numpy
import numpy as np

A = np.matrix([[1, 4, 10], [0, 2, 0], [0, 0 ,3]])
B = symb.Matrix([[1], [1], [1]])
C = symb.Matrix([[1, 0, 0]])
D = symb.Matrix([0])
print('\n A =', A)
```

```
autoval, M = np.linalg.eig(A)
D = np.diag(autoval)
print('\n D =', D)
print('\n M =', M)

# Verificamos:

A_hat = np.linalg.inv(M)*A*M
B_hat = np.linalg.inv(M)*B
C_hat = C*M
print('\n A_hat =', A_hat)
print('\n B_hat =', B_hat)
print('\n C_hat =', C_hat)
```

```
A = [[ 1  4 10]
[ 0  2  0]
[ 0  0  3]]

D = [[1. 0. 0.]
[0. 2. 0.]
[0. 0. 3.]]

M = [[1.         0.9701425  0.98058068]
[0.         0.24253563 0.        ]
[0.         0.         0.19611614]]

A_hat = [[1.00000000e+00 0.00000000e+00 4.11536054e-16]
[0.00000000e+00 2.00000000e+00 0.00000000e+00]
[0.00000000e+00 0.00000000e+00 3.00000000e+00]]

B_hat = Matrix([[-8.00000000000000], [4.12310562561766], [5.09901951359278]])

C_hat = Matrix([[1.00000000000000, 0.970142500145332, 0.980580675690920]])
```

Notar que a pesar de hallar la misma $\hat{A}$ con ambos métodos, la matriz M hallada es distinta y por lo tanto las matrices $\hat{B}$ y $\hat{C}$ cambian.

Forma de Jordan

Pero no siempre una matriz cuadrada es diagonalizable. ¿Cuáles son las condiciones para que lo sea? Repasá el video de la clase teórica:

https://youtu.be/6rW7UNVlfc4?t=598

En un caso más general podría darse que para algún autovalor λ_j la multiplicidad algebraica m_j sea mayor que la multiplicidad geométrica g_j. Supongamos por ejemplo que $m_j = 3$, o sea, es un autovalor triple y $g_j = 1$. Es el caso en que solamente vamos a poder extraer un autovector normal o regular asociado a λ_j.

No vamos a poder llegar a una forma diagonal, porque ahora tenemos:

$$Av_j = \lambda_j v_j$$

y

$$(A - \lambda_j I)v_j = 0$$

solo nos da una única posibilidad de v_j.

Vamos a necesitar 2 vectores más para formar la base $\{v_i\}$, $i = 1, \ldots, n$. La forma más simple para obtener esos dos vectores es a partir de las condiciones:

$$\begin{aligned} Aw_1 &= \lambda_j w_1 + v_j \\ Aw_2 &= \lambda_j w_2 + w_1 \end{aligned}$$

Y definiendo w_1 y w_2 para que $\{v_1, \ldots, v_j, w_1, w_2, \ldots v_n\}$ sea LI.

Con esas ecuaciones y las condiciones, despejamos posibles valores y armamos la matriz M:

$$M = \left[\begin{array}{c|c|c|c|c|c|c|c} v_1 & v_2 & \cdots & v_j & w_1 & w_2 & \cdots & v_n \end{array}\right]$$

Con ella, nos preguntamos qué forma tomará $M^{-1}AM$. Desarrollemos:

$$AM = A\left[\begin{array}{c|c|c|c|c|c|c|c} v_1 & v_2 & \cdots & v_j & w_1 & w_2 & \cdots & v_n \end{array}\right] = \left[\begin{array}{c|c|c|c|c|c|c|c} Av_1 & Av_2 & \cdots & Av_j & Aw_1 & Aw_2 & \cdots & Av_n \end{array}\right]$$

Aplicando las condiciones de autovectores y autovectores generalizados en cada caso:

$$AM = \left[\begin{array}{c|c|c|c|c|c|c|c} \lambda_1 v_1 & \lambda_2 v_2 & \cdots & \lambda_j v_j & \lambda_j w_1 + v_j & \lambda_j w_2 + w_1 & \cdots & \lambda_n v_n \end{array}\right]$$

Si consideramos el producto de matrices, se puede factorizar en:

$$AM = \left[\begin{array}{c|c|c|c|c|c|c|c} v_1 & v_2 & \cdots & v_j & w_1 & w_2 & \cdots & v_n \end{array}\right] \underbrace{\begin{bmatrix} \lambda_1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & \lambda_2 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ \vdots & 0 & \ddots & & & & & \vdots \\ \vdots & \vdots & & \lambda_j & 1 & 0 & & \vdots \\ \vdots & \vdots & & 0 & \lambda_j & 1 & & \vdots \\ \vdots & \vdots & & 0 & 0 & \lambda_j & & \vdots \\ \vdots & \vdots & & & & & \ddots & 0 \\ 0 & 0 & \cdots & \cdots & \cdots & \cdots & 0 & \lambda_n \end{bmatrix}}_{J} = MJ$$

En resumen:

$$\hat{A} = M^{-1}AM = J = \begin{bmatrix} \lambda_1 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & 0 \\ 0 & \lambda_2 & 0 & \cdots & \cdots & \cdots & \cdots & 0 \\ \vdots & 0 & \ddots & & & & & \vdots \\ \vdots & \vdots & & \lambda_j & 1 & 0 & & \vdots \\ \vdots & \vdots & & 0 & \lambda_j & 1 & & \vdots \\ \vdots & \vdots & & 0 & 0 & \lambda_j & & \vdots \\ \vdots & \vdots & & & & & \ddots & 0 \\ 0 & 0 & \cdots & \cdots & \cdots & \cdots & 0 & \lambda_n \end{bmatrix}$$

En donde queda una matriz J con lo que se llaman bloques de Jordan. En este caso cada autovalor distinto tiene su propio bloque de Jordan y además, los autovectores asociados a λ_j forman un bloque de 3x3.

En este ejemplo la ecuación de estado tiene algunas variables acopladas, pero de la manera más simple posible. Ya vimos la forma que tenía el sistema con las variables de estado de fase, con la supradiagonal con unos, y con esa experiencia tenemos una idea de cómo queda el diagrama de simulación para esos bloques.

Ejemplo:

Ejercicio 2

$$A_3 = \begin{bmatrix} 0 & 4 & 3 \\ 0 & 20 & 16 \\ 0 & -25 & -20 \end{bmatrix}$$

$$B = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, D = 0$$

a) Hallar la forma de Jordan

Polinomio característico:

$$det(\lambda I - A) = \det\left(\begin{bmatrix} \lambda & -4 & -3 \\ 0 & \lambda - 20 & -16 \\ 0 & -25 & \lambda + 20 \end{bmatrix}\right) = \lambda\left[(\lambda - 20)(\lambda - 20) + 25 \cdot 16\right] = \lambda^3$$

Da autovalor único $\lambda_1 = 0$ con multiplicidad algebraica $m_1 = 3$.

El autoespacio asociado el núcleo o espacio nulo de A:

$$(A-\lambda I)v = Av = \begin{bmatrix} 0 & 4 & 3 \\ 0 & 20 & 16 \\ 0 & -25 & -20 \end{bmatrix} \begin{bmatrix} v_{11} \\ v_{21} \\ v_{31} \end{bmatrix} = 0$$

De esta expresión sale que:

- v_{11} puede tomar cualquier valor.
- $v_{21} = v_{31} = 0$.

Eso deja:

$$v_1 = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}^T$$

como autovector normal o regular. Y espacio nulo de dimensión 1, es decir multiplicidad algebraica $g_1 = 1$. Necesitamos generar dos autovectores generalizados a partir de v_1. El primero:

$$Aw_1 = \lambda w_1 + v_1 = v_1$$

Ahora hay que resolver:

$$\begin{bmatrix} 0 & 4 & 3 \\ 0 & 20 & 16 \\ 0 & -25 & -20 \end{bmatrix} \begin{bmatrix} w_{11} \\ w_{21} \\ w_{31} \end{bmatrix} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$$

Ejercicio:

Verificar que se puede elegir:

$$w_1 = \begin{bmatrix} 0 & 4 & -5 \end{bmatrix}^T$$

Luego, a partir de w_1 se genera w_2:

$$Aw_2 = \lambda w_2 + w_1 = w_1$$

$$\begin{bmatrix} 0 & 4 & 3 \\ 0 & 20 & 16 \\ 0 & -25 & -20 \end{bmatrix} \begin{bmatrix} w_{12} \\ w_{22} \\ w_{32} \end{bmatrix} = \begin{bmatrix} 0 \\ 4 \\ -5 \end{bmatrix}$$

Ejercicio:

Verificar que se puede elegir:

$$w_2 = \begin{bmatrix} 0 & -3 & 4 \end{bmatrix}^T$$

Queda conformada entonces la matriz:

$$M = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 4 & -3 \\ 0 & -5 & -4 \end{bmatrix}$$

Y con ella el sistema en base de autovectores queda en la forma de Jordan:

$$\hat{A} = M^{-1}AM = J = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}$$

Con

$$\hat{B} = M^{-1}B = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 4 & 3 \\ 0 & 5 & 4 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 \\ 7 \\ 9 \end{bmatrix}$$

y

$$\hat{C} = CM = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$$

Para pensar: ¿Es única la matriz M hallada?

```
# Ejemplo con sympy
import sympy as symb
import control as ctrl

A = symb.Matrix([[0, 4, 3], [0, 20, 16], [0, -25 ,-20]])
B = symb.Matrix([[1], [1], [1]])
C = symb.Matrix([[1, 0, 0]])
D = symb.Matrix([0])
print('\n A =', A)

n = A.rows
```

```
I = symb.eye(n)

print('\n Método directo:')

M, _ = A.jordan_form()

print('\n M:', M)

A_hat = M.inv()*A*M
B_hat = M.inv()*B
C_hat = C*M
print('\n A_hat =', A_hat)
print('\n B_hat =', B_hat)
print('\n C_hat =', C_hat)
```

```
A = Matrix([[0, 4, 3], [0, 20, 16], [0, -25, -20]])

Método directo:

M: Matrix([[5, 4, 0], [0, 20, 1], [0, -25, 0]])

A_hat = Matrix([[0, 1, 0], [0, 0, 1], [0, 0, 0]])

B_hat = Matrix([[29/125], [-1/25], [9/5]])

C_hat = Matrix([[5, 4, 0]])
```

```
print('\n Como es de esperar ambas descripciones presentan la misma función transferencia:')

s = symb.symbols('s')
H = C_hat*(s*I-A_hat).inv()*B_hat+D
H.simplify()
print('\n H(s) = ')
symb.pprint(H)

H = C*(s*I-A).inv()*B+D
H.simplify()
print('\n H(s) = ')
symb.pprint(H)
```

```
Como es de esperar ambas descripciones presentan la misma función transferencia:

H(s) =
⎡ 2          ⎤
⎢s  + 7⋅s + 9⎥
⎢────────────⎥
⎢      3     ⎥
⎣     s      ⎦

H(s) =
⎡ 2          ⎤
⎢s  + 7⋅s + 9⎥
⎢────────────⎥
⎢      3     ⎥
⎣     s      ⎦
```

Ejercicios sugeridos para seguir:

1) Resolver el Ejercicio 1, matriz A_2 con el método analítico paso por paso, pero usando la computadora. Es decir, sin usar las funciones de Jordan directos. Verificar a partir de M la forma de la matriz $J = M^{-1}A_2M$.

2) Para los dos ejercicios resueltos en este notebook (Ejercicio 2, A_1 y A_3) completar los ítems:

```
b) Dibujar los diagramas de simulación en la forma original y en la de Jordan

c) Hallar H(s). Verifique por simulación.
```

Pista para el ejercicio 8:

Para llegar a la transformación propuesta en el ejercicio 7, hay que analizar lo que hace la siguiente transformación:

$$T = \begin{bmatrix} 1/2 & -j/2 \\ 1/2 & j/2 \end{bmatrix}$$

a una matriz:

$$D = \begin{bmatrix} \alpha + j\beta & 0 \\ 0 & \alpha - j\beta \end{bmatrix}$$

Primero calculamos:

$$T^{-1} = \begin{bmatrix} 1 & 1 \\ j & -j \end{bmatrix}$$

Al aplicar esta transformación sobre D nos da:

$$\begin{aligned} T^{-1}DT &= \begin{bmatrix} 1 & 1 \\ j & -j \end{bmatrix} \begin{bmatrix} \alpha + j\beta & 0 \\ 0 & \alpha - j\beta \end{bmatrix} \begin{bmatrix} 1/2 & -j/2 \\ 1/2 & j/2 \end{bmatrix} \\ &= \begin{bmatrix} \alpha + j\beta & \alpha - j\beta \\ -\beta + j\alpha & -\beta - j\alpha \end{bmatrix} \begin{bmatrix} 1/2 & -j/2 \\ 1/2 & j/2 \end{bmatrix} \\ &= \begin{bmatrix} \alpha & \beta \\ -\beta & \alpha \end{bmatrix} \end{aligned}$$

Que nos indica que la transformación de semejanza propuesta nos lleva a una forma con coeficientes reales a la matriz D.

Luego, si D sale de diagonalizar la matriz A se llega a la transformación directa:

$$T^{-1}DT = T^{-1}M^{-1}AMT = (MT)^{-1}AMT$$

con:

$$MT = \left[\begin{array}{c|c} u + jv & u - jv \end{array} \right] \begin{bmatrix} 1/2 & -j/2 \\ 1/2 & j/2 \end{bmatrix}$$

$$= \left[\begin{array}{c|c} \frac{1}{2}(u + jv) - \frac{1}{2}(u - jv) & \frac{-j}{2}(u + jv) + \frac{j}{2}(u - jv) \end{array} \right]$$

$$= \left[\begin{array}{c|c} u & v \end{array} \right]$$

✓ 0 s se ejecutó 17:26