TD9_SDZ

December 5, 2021

[45]:
```python
import numpy as np
import control
from control.matlab import lsim
import matplotlib.pyplot as plt
```

Task XII.1. (30 points) For the following systems, place the eigenvalues accordingly, by assuming a state-feedback matrix K. Consider no input and that the feedforward matrix D is zero.

For each of the following systems, build a controller and simulate the system for some time-frame and initial values you consider appropriate.

(a) $\mathbf{A} = \begin{bmatrix} 1 & 0.3 \\ -1 & -0.1 \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} 1 \\ 1 \end{bmatrix}$, $\mathbf{C} = \begin{bmatrix} 1 & 0 \end{bmatrix}$, $\mathbf{x_0} = \begin{bmatrix} 1 \\ 0 \end{bmatrix}$, $\mathbf{p_{1,2}} = -2, -3$

(b) $\mathbf{A} = \begin{bmatrix} 1 & 4 & 1 \\ 1 & 1 & 1 \\ 1 & 2 & -1 \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} 2 \\ 2 \\ 2 \end{bmatrix}$, $\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$, $\mathbf{x_0} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$, $\mathbf{p_{1,2,3}} = -5, -2 \pm i$

(c) Same as previous, but with $\mathbf{p_{1,2,3}} = -2, -3 \pm i$

(d) $\mathbf{A} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 7 & 1 \\ 1 & 3 & -1 \end{bmatrix}$, $\mathbf{B} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}$, $\mathbf{C} = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}$, $\mathbf{x_0} = \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$, $\mathbf{p_{1,2,3}} = -5, 2 \pm i$.

1 Système (a)

$$K^T = \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} B & AB \end{pmatrix}^{-1} \left(A^2 - (p_1 + p_2)A + Ip_1p_2\right)$$

$$AB = \begin{pmatrix} 1.3 \\ -1.1 \end{pmatrix}$$

$$K = \begin{pmatrix} 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 1.3 \\ 1 & -1.1 \end{pmatrix}^{-1} \left(A^2 - (p_1 + p_2)A + Ip_1p_2\right)$$

$$K = \begin{pmatrix} 0 & 1 \end{pmatrix} \frac{-1}{2.4} \begin{pmatrix} -1.1 & -1.3 \\ -1 & 1 \end{pmatrix} \left(A^2 - (p_1 + p_2)A + Ip_1p_2\right)$$

$$K = \begin{pmatrix} 0 & 1 \end{pmatrix} \frac{1}{2.4} \begin{pmatrix} 1.1 & 1.3 \\ 1 & -1 \end{pmatrix} \left(A^2 - (p_1 + p_2)A + Ip_1p_2\right)$$

$$K = \frac{1}{2.4} \begin{pmatrix} 1 & -1 \end{pmatrix} \left(A^2 - (p_1 + p_2 + p_3)A + Ip_1p_2\right)$$

$$K = \frac{1}{2.4} \begin{pmatrix} 1 & -1 \end{pmatrix} \left(\begin{pmatrix} 0.7 & 0.27 \\ -0.9 & -0.29 \end{pmatrix} + 5 \begin{pmatrix} 1 & 0.3 \\ -1 & -0.1 \end{pmatrix} + 6I \right)$$

$$K = \frac{1}{2.4}\begin{pmatrix}1 & -1\end{pmatrix}\begin{pmatrix}11.7 & 1.77 \\ -5.9 & 5.21\end{pmatrix}$$

$$K = \frac{1}{2.4}\begin{pmatrix}1 & -1\end{pmatrix}$$

$$K = \begin{pmatrix}\frac{22}{3} & \frac{-43}{30}\end{pmatrix}$$

[81]:

```
systems = []

# Système (a)
A = np.matrix('[1 0.3;-1 -0.1]')
B = np.matrix('[1;1]')
C = np.matrix('[1 0]')
P = np.array([-2, -3])
#K = 1/2.4*np.matrix('[1 -1]')*(A**2 - (p1+p2)*A + p1*p2*np.eye(2))
K = control.acker(A, B, P)
x0 = np.array([1,0])

systems.append([A, B, C, K, x0, P])

# Système (b)
A = np.matrix('[1 4 1;1 1 1; 1 2 -1]')
B = np.matrix('[2;2;2]')
C = np.matrix('[1 0 0]')
P = np.array([-5, -2+1j, -2-1j])
K = control.acker(A, B, P)
x0 = np.array([1,0,0])
systems.append([A, B, C, K, x0, P])

# Système (c)
A = np.matrix('[1 4 1;1 1 1; 1 2 -1]')
B = np.matrix('[2;2;2]')
C = np.matrix('[1 0 0]')
P = np.array([-2, -3+1j, -3-1j])
K = control.acker(A, B, P)
x0 = np.array([1,0,0])
systems.append([A, B, C, K, x0, P])

# Système (d)
A = np.matrix('[1 0 1; 0 7 1; 1 3 -1]')
B = np.matrix('[1;1;1]')
C = np.matrix('[1 0 0]')
P = np.array([-5, 2+1j, 2-1j])
K = control.acker(A, B, P)
x0 = np.array([1,0,0])
systems.append([A, B, C, K, x0, P])
```

[82]:

```python
plt.figure(figsize=(15,10))
for i, s in enumerate(systems):
    A, B, C, K, x0, P = s
    plt.subplot(220 + i + 1)
    sys_bf = control.StateSpace(A-B*K, B, C, 0)

    Treg = np.max(3/np.abs(P))
    t = np.linspace(0, 3, 1000)
    y, _, X = lsim(sys_bf, T=t, X0=x0)
    plt.plot(t, X)
    plt.legend(['Sortie'])
    plt.title(f"Système {i+1}")
    plt.grid()

plt.show()
```