OPTIMAL CONTROL

👤 Sébastien Boisgérault

CONTROL ENGINEERING WITH PYTHON

- 📖 Documents (GitHub)
-
- 🏦 Mines ParisTech, PSL University

SYMBOLS

🐍	Code	🔍	Worked Example
📈	Graph	🧩	Exercise
🏷️	Definition	💻	Numerical Method
💎	Theorem	🧮	Analytical Method
📝	Remark	🧠	Theory
ℹ	Information	🗝️	Hint
⚠️	Warning	🔓	Solution

🐍 IMPORTS

```
from numpy import *
from numpy.linalg import *
from matplotlib.pyplot import *
from scipy.integrate import solve_ivp
from scipy.linalg import solve_continuous_are
```

WHY OPTIMAL CONTROL?

Limitations of Pole Assignment

- It is not always obvious what set of poles we should target (especially for large systems),
- We do not control explicitly the trade-off between "speed of convergence" and "intensity of the control" (large input values maybe costly or impossible).

Let

$$\dot{x} = Ax + Bu$$

where

- $A \in \mathbb{R}^{n \times n}, B \in \mathbb{R}^{m \times n}$ and
- $x(0) = x_0 \in \mathbb{R}^n$ is given.

Find $u(t)$ that minimizes

$$J = \int_0^{+\infty} x(t)^t Q x(t) + u(t)^t R u(t) \, dt$$

where:

- $Q \in \mathbb{R}^{n \times n}$ and $R \in \mathbb{R}^{m \times m}$,
- (to be continued …)

- Q and R are **symmetric** ($R^t = R$ and $Q^t = Q$),
- Q and R are **positive definite** (denoted "> 0")

$$x^tQx \geq 0 \text{ and } x^tQx = 0 \text{ iff } x = 0$$

and

$$u^tRu \geq 0 \text{ and } u^tRu = 0 \text{ iff } u = 0.$$

HEURISTICS / SCALAR CASE

If $x \in \mathbb{R}$ and $u \in \mathbb{R}$,

$$J = \int_0^{+\infty} qx(t)^2 + ru(t)^2 \, dt$$

with $q > 0$ and $r > 0$.

When we minimize J:

- Only the relative values of q and r matters.
- Large values of q penalize strongly non-zero states:

  $\Rightarrow$ fast convergence.
- Large values of r penalize strongly non-zero inputs:

  $\Rightarrow$ small input values.

HEURISTICS / VECTOR CASE

If $x \in \mathbb{R}^n$ and $u \in \mathbb{R}^m$ and Q and R are diagonal,

$$Q = \operatorname{diag}(q_1, \cdots, q_n), \; R = \operatorname{diag}(r_1, \cdots, r_m),$$

$$J = \int_0^{+\infty} \sum_i q_i x_i(t)^2 + \sum_j r_j u_j(t)^2 \, dt$$

with $q_i > 0$ and $r_j > 0$.

Thus we can control the cost of each component of x and u independently.

💎 OPTIMAL SOLUTION

Assume that $\dot{x} = Ax + Bu$ is controllable.

- There is an optimal solution; it is a linear feedback

$$u = -Kx$$

- The closed-loop dynamics is asymptotically stable.

💎 ALGEBRAIC RICCATI EQUATION

- The gain matrix K is given by

$$K = R^{-1} B^t \Pi,$$

where $\Pi \in \mathbb{R}^{n \times n}$ is the unique matrix such that $\Pi^t = \Pi, \Pi > 0$ and

$$\Pi B R^{-1} B^t \Pi - \Pi A - A^t \Pi - Q = 0.$$

🔍 OPTIMAL CONTROL

Consider the double integrator $\ddot{x} = u$

$$\frac{d}{dt}\begin{bmatrix} x \\ \dot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

(in standard form)

🐍 PROBLEM DATA

```
A = array([[0, 1], [0, 0]])
B = array([[0], [1]])
Q = array([[1, 0], [0, 1]])
R = array([[1]])
```

🐍 OPTIMAL GAIN

```
Pi = solve_continuous_are(A, B, Q, R)
K = inv(R) @ B.T @ Pi
```

🐍 CLOSED-LOOP BEHAVIOR

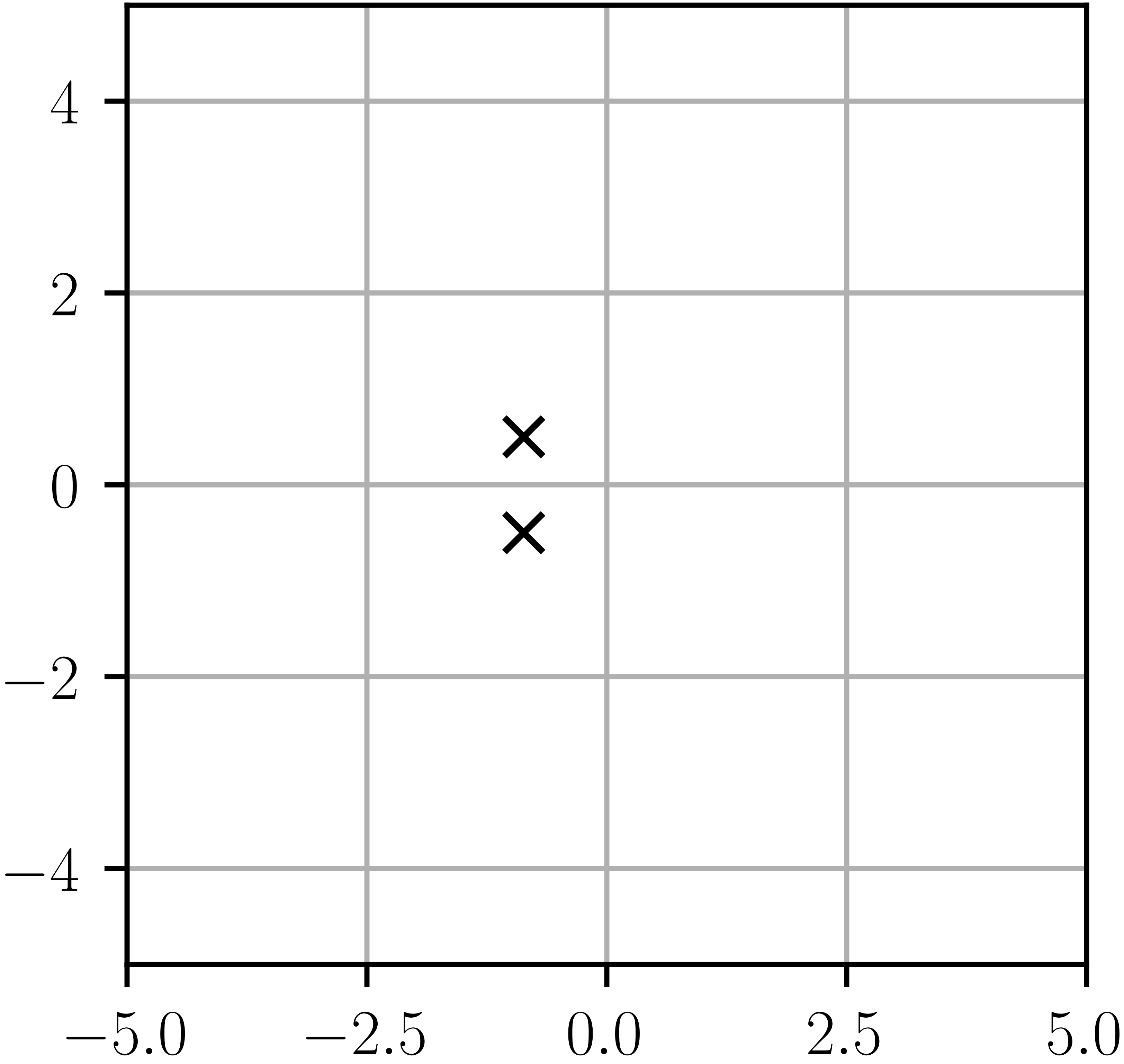
It is stable:

```
eigenvalues, _ = eig(A - B @ K)
assert all([real(s) < 0 for s in eigenvalues])
```

📈 EIGENVALUES LOCATION

```
figure()
x = [real(s) for s in eigenvalues]
y = [imag(s) for s in eigenvalues]
plot(x, y, "kx")
grid(True)
title("Eigenvalues")
axis("square")
axis([-5, 5, -5, 5])
```

Eigenvalues



🐍 SIMULATION

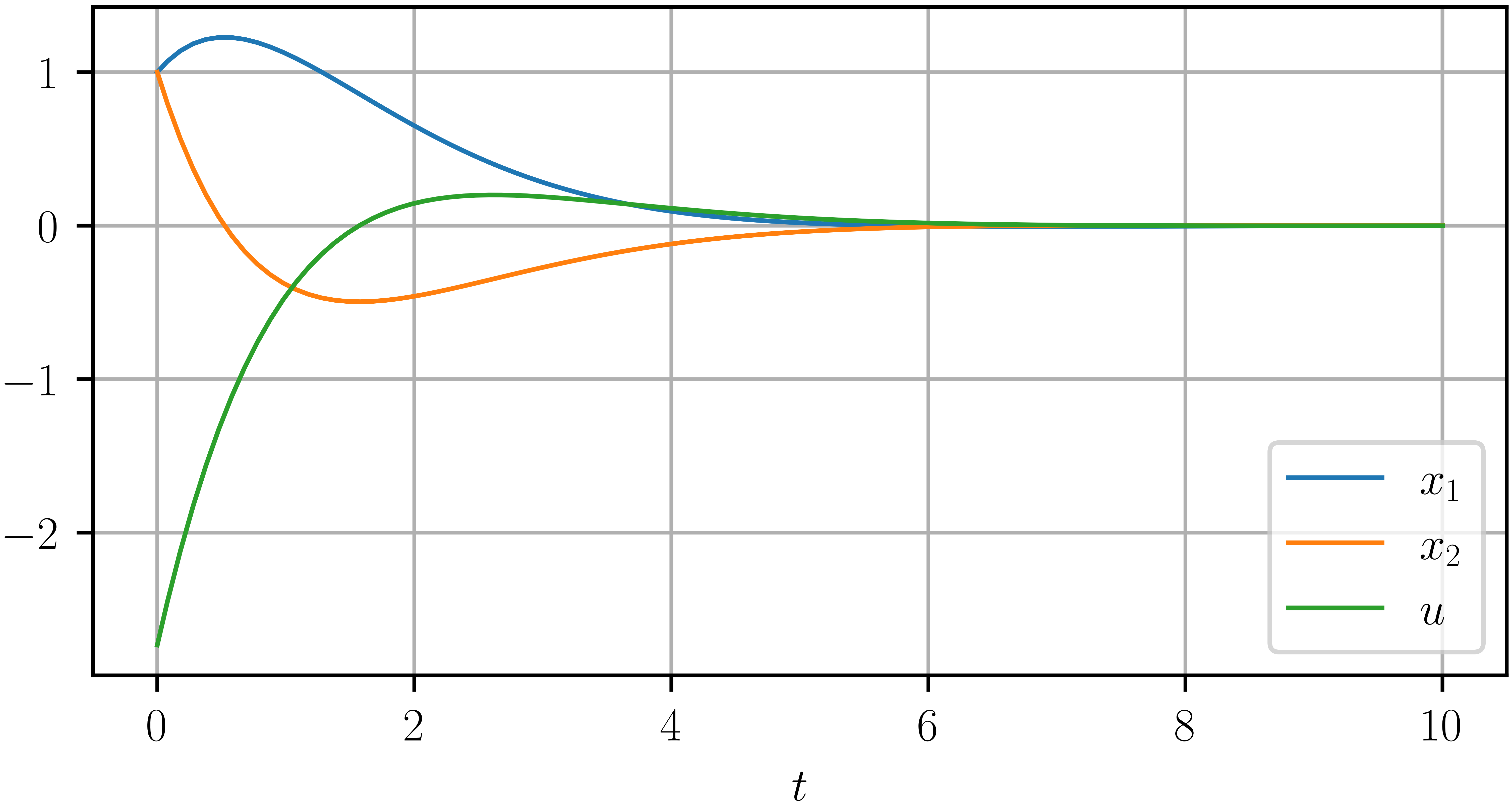
```python
y0 = [1.0, 1.0]
def f(t, x):
    return (A - B @ K) @ x
```

🐍 SIMULATION

```python
result = solve_ivp(
  f, t_span=[0, 10], y0=y0, max_step=0.1
)
t = result["t"]
x1 = result["y"][0]
x2 = result["y"][1]
u = - (K @ result["y"]).flatten() # vect. -> scalar
```

📈 INPUT & STATE EVOLUTION

```
figure()
plot(t, x1, label="$x_1$")
plot(t, x2, label="$x_2$")
plot(t, u, label="$u$")
xlabel("$t$")
grid(True)
legend(loc="lower right")
```


1
0
−1
−2
0
2
4
6
8
10
t
x1
x2
u

🐍 OPTIMAL GAIN

```
Q = array([[10, 0], [0, 10]])
R = array([[1]])
Pi = solve_continuous_are(A, B, Q, R)
K = inv(R) @ B.T @ Pi
```

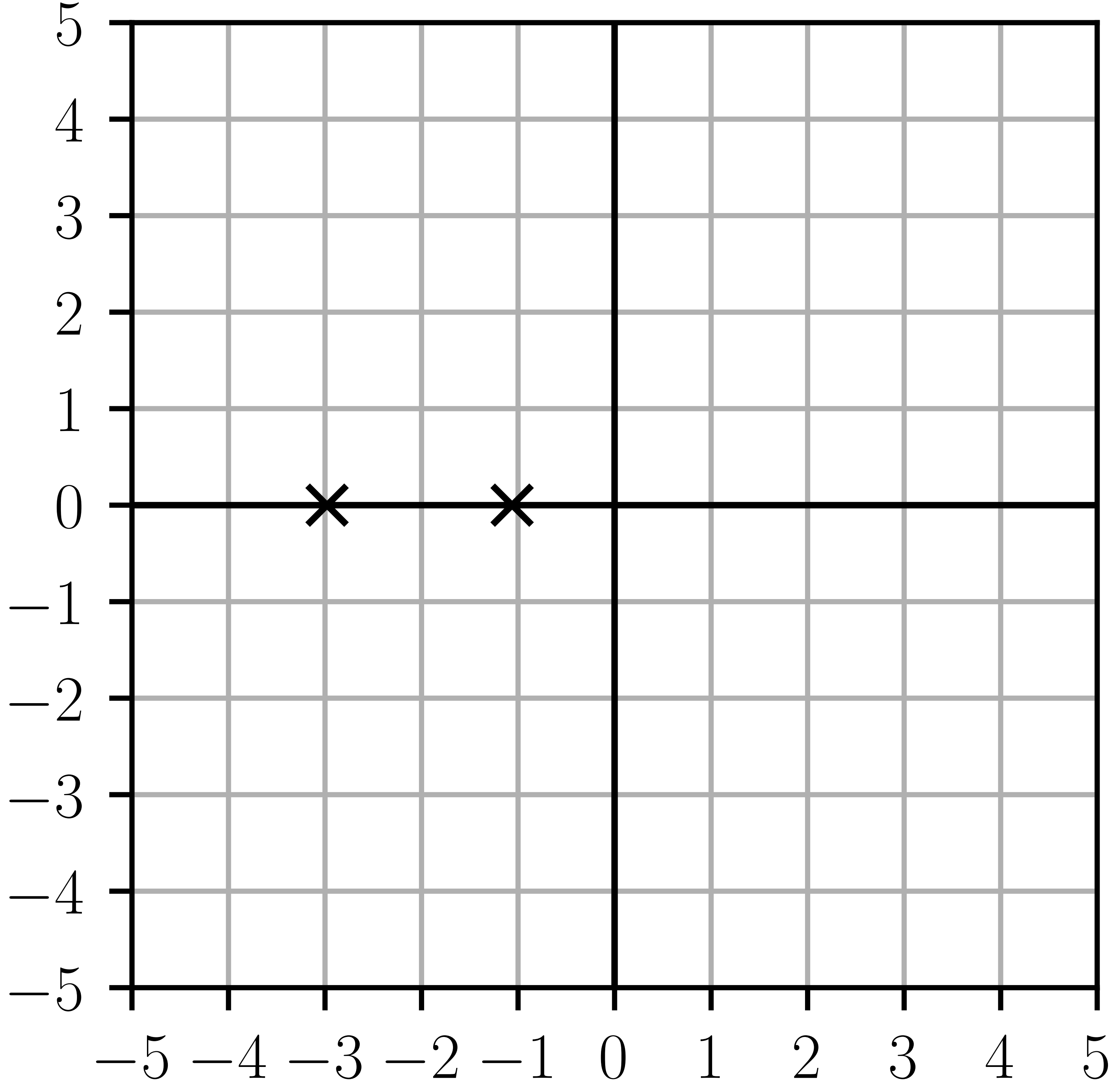
🐍 CLOSED-LOOP ASYMP. STAB.

```python
eigenvalues, _ = eig(A - B @ K)
assert all([real(s) < 0 for s in eigenvalues])
```

📈 EIGENVALUES LOCATION

```
figure()
x = [real(s) for s in eigenvalues]
y = [imag(s) for s in eigenvalues]
plot(x, y, "kx")
```

Eigenvalues

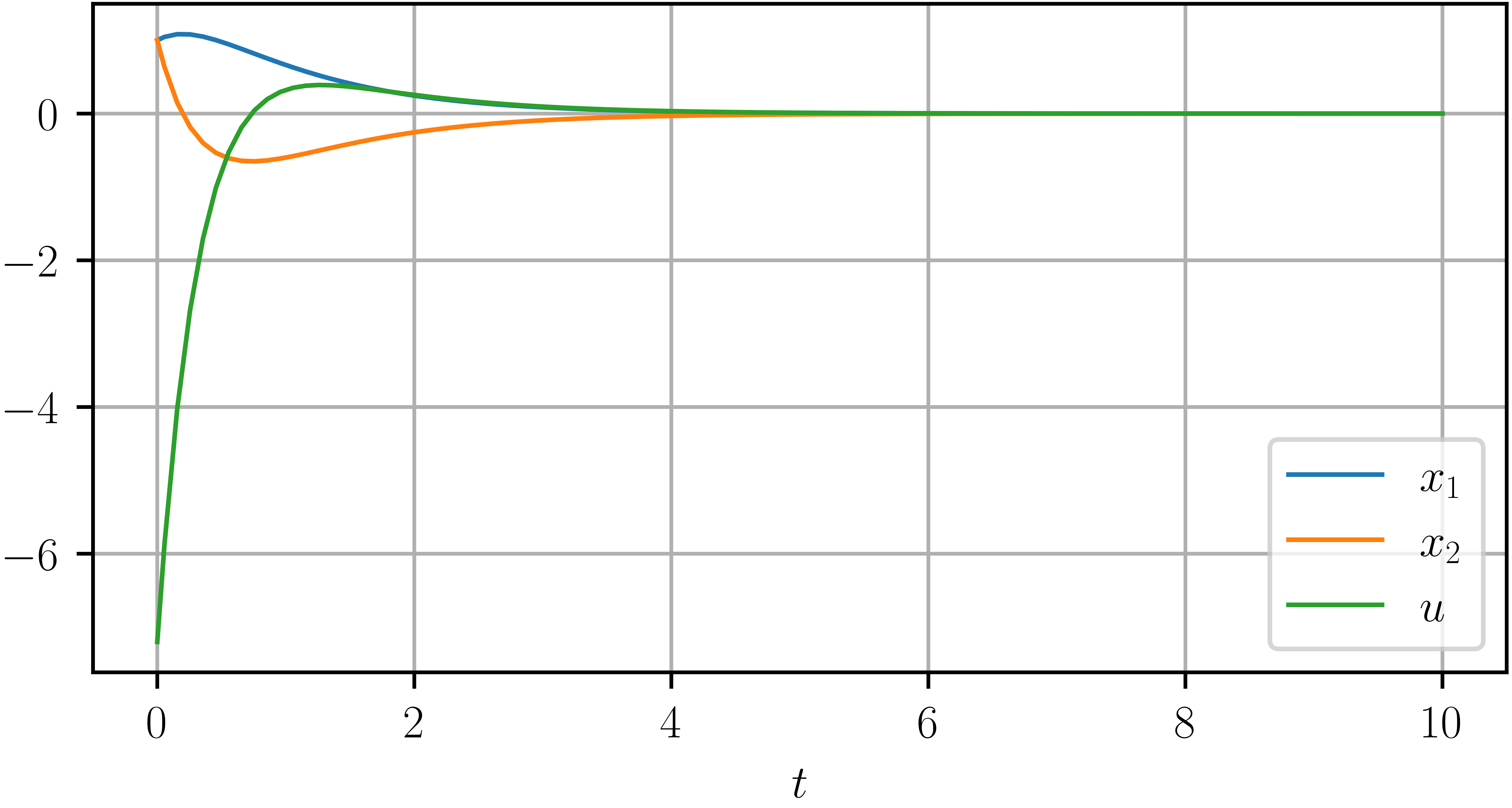


🐍 SIMULATION

```
result = solve_ivp(
  f, t_span=[0, 10], y0=y0, max_step=0.1
)
t = result["t"]
x1 = result["y"][0]
x2 = result["y"][1]
u = - (K @ result["y"]).flatten() # vect. -> scalar
```

📈 INPUT & STATE EVOLUTION

```
figure()
plot(t, x1, label="$x_1$")
plot(t, x2, label="$x_2$")
plot(t, u, label="$u$")
xlabel("$t$")
grid(True)
legend(loc="lower right")
```


0
−2
−4
−6
0
2
4
6
8
10
t
x1
x2
u

🐍 OPTIMAL GAIN

```python
Q = array([[1, 0], [0, 1]])
R = array([[10]])
Pi = solve_continuous_are(A, B, Q, R)
K = inv(R) @ B.T @ Pi
```

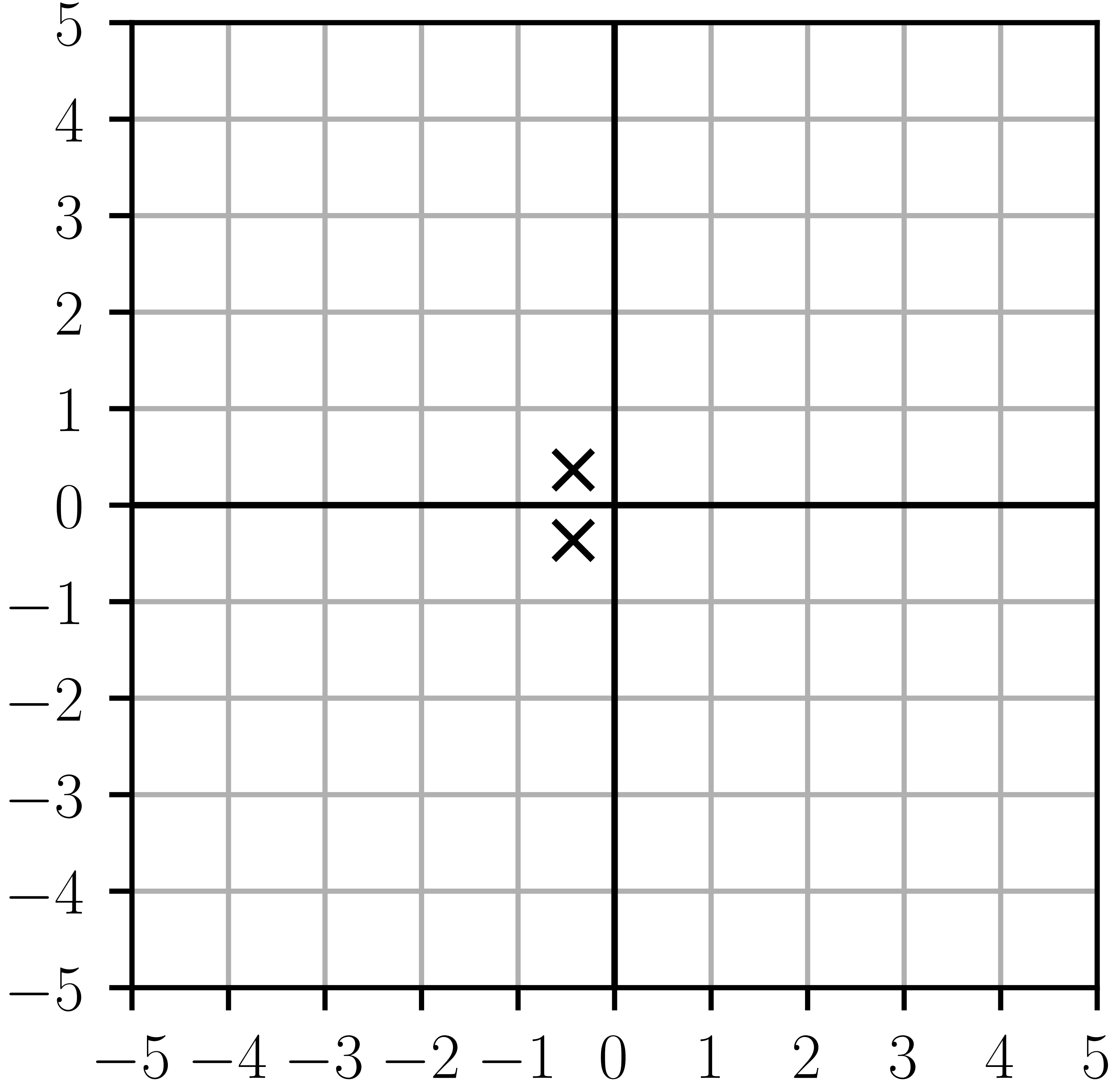
🐍 CLOSED-LOOP ASYMP. STAB.

```python
eigenvalues, _ = eig(A - B @ K)
assert all([real(s) < 0 for s in eigenvalues])
```

📈 EIGENVALUES LOCATION

```
figure()
x = [real(s) for s in eigenvalues]
y = [imag(s) for s in eigenvalues]
plot(x, y, "kx")
```

Eigenvalues

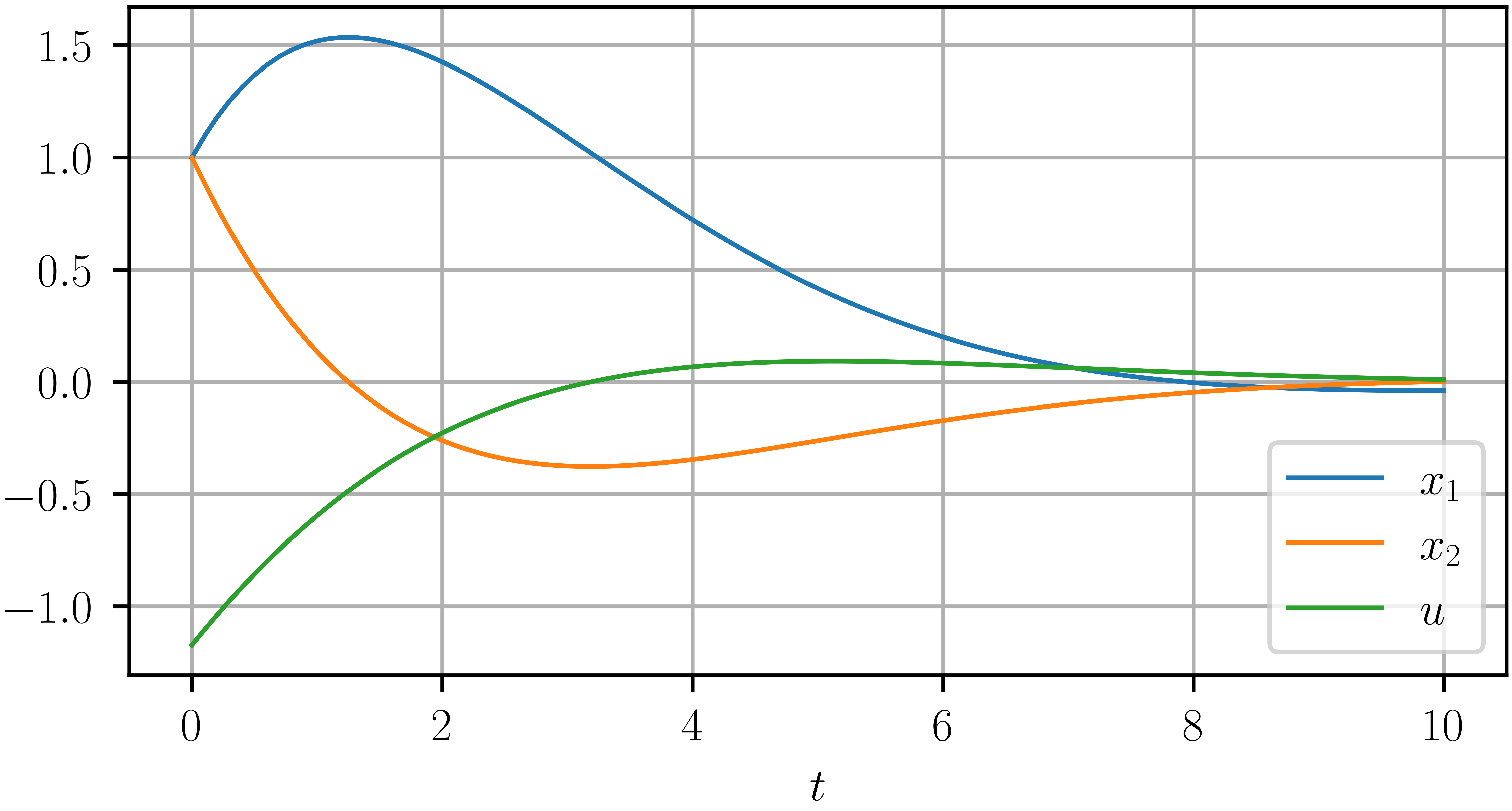


🐍 SIMULATION

```
result = solve_ivp(
  f, t_span=[0, 10], y0=y0, max_step=0.1
)
t = result["t"]
x1 = result["y"][0]
x2 = result["y"][1]
u = - (K @ result["y"]).flatten() # vect. -> scalar
```

📈 INPUT & STATE EVOLUTION

```
figure()
plot(t, x1, label="$x_1$")
plot(t, x2, label="$x_2$")
plot(t, u, label="$u$")
xlabel("$t$")
grid(True)
legend(loc="lower right")
```


1.5
1.0
0.5
0.0
−0.5
−1.0
0
2
4
6
8
10
t
x_1
x_2
u

🧩 OPTIMAL VALUE

Consider the controllable dynamics

$$\dot{x} = Ax + Bu$$

and $u(t)$ the control that minimizes

$$J = \int_0^{+\infty} x(t)^t Q x(t) + u(t)^t R u(t) \, dt.$$

1 🧮.

Let

$$j(x, u) := x^t Q x + u^t R u.$$

Show that

$$j(x(t), u(t)) = -\frac{d}{dt} x(t)^t \Pi x(t)$$

2. 🧮

What is the value of J?

🔓 OPTIMAL VALUE

1. 🔓

We know that $u = -Kx$ where $K = R^{-1}B^t\Pi$ and Π is a symmetric solution of

$$\Pi B R^{-1} B^t \Pi - \Pi A - A^t \Pi - Q = 0.$$

Since R is symmetric,

$$\Pi B R^{-1} B^t \Pi = \Pi B (R^{-1})^t R R^{-1} B^t \Pi = K^t R K$$

and thus

$$\Pi A + A^t \Pi = K^t R K - Q.$$

Since $\dot{x} = (A - BK)x$,

$$
\begin{aligned}
\frac{d}{dt} x^t \Pi x &= x^t (\Pi(A - BK) + (A - BK)^t \Pi) x \\
&= x^t (\Pi A + A^t \Pi - \Pi B K - (BK)^t \Pi) x \\
&= x^t (K^t R K - Q - K^t R K - K^t R K) x \\
&= x^t (-Q - K^t R K) x^t \\
&= -x^t Q x - u^t R u \\
&= -j(x, u).
\end{aligned}
$$

2. 🔓

Since the system is controllable, the optimal control makes the origin of the closed-loop system asymptotically stable. Consequently, $x(t) \to 0$ when $t \to +\infty$. Hence,

$$
\begin{aligned}
J &= \int_0^{+\infty} j(x, u)\, dt \\
&= - \int_0^{+\infty} \frac{d}{dt} x^t \Pi x \, dt \\
&= - \left[x^t \Pi x \right]_0^{+\infty}
\end{aligned}
$$