ASYMPTOTIC STABILIZATION

👤 Sébastien Boisgérault

CONTROL ENGINEERING WITH PYTHON

- 📖 Documents (GitHub)
- © License CC BY 4.0
- 🏦 Mines ParisTech, PSL University

SYMBOLS

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

🐍 IMPORTS

```
from numpy import *
from numpy.linalg import *
from numpy.testing import *
from scipy.integrate import *
from scipy.linalg import *
from matplotlib.pyplot import *
```

🧭 ASYMPTOTIC STABILIZATION

When the system

$$\dot{x} = Ax, \; x \in \mathbb{R}^n$$

is not asymptotically stable at the origin, maybe there are some inputs $u \in \mathbb{R}^m$ such that

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

that we can use to stabilize asymptotically the system?

🏷️ LINEAR FEEDBACK

We search for u as a **linear feedback**:

$$u(t) = -Kx(t)$$

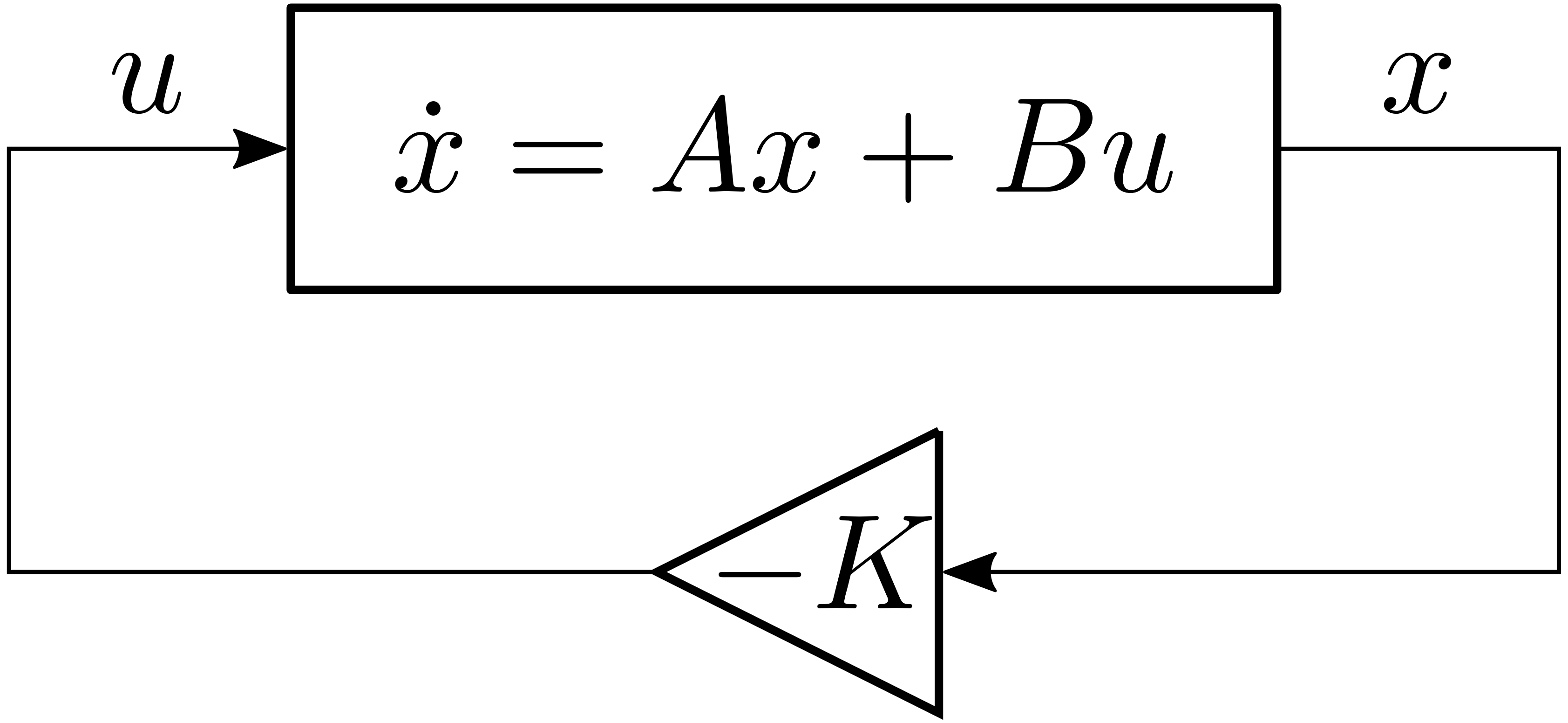
for some $K \in \mathbb{R}^{m \times n}$.

📝 NOTE

In this scheme

- ⚠️ The full system state $x(t)$ **must be measured**.
- 🏷️ This information is then **fed back** into the system.
- 🏷️ The feedback **closes the loop**.

🏷️ CLOSED-LOOP DIAGRAM



💎 CLOSED-LOOP DYNAMICS

When

$$\begin{aligned}\dot{x} &= Ax + Bu \\ u &= -Kx\end{aligned}$$

the state $x \in \mathbb{R}^n$ evolves according to:

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

💎 REMINDER

The closed-loop system is asymptotically stable iff every eigenvalue of the matrix

$$A - BK$$

is in the open left-hand plane.

🏷️ SPECTRUM AS A MULTISET

Multisets remember the multiplicity of their elements.
It's convenient to describe the spectrum of matrices:

$$A := \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 2 \end{bmatrix} \Rightarrow \sigma(A) = \{1, 1, 2\}$$

$$0 \notin \sigma(A),\ 1 \in \sigma(A),\ 1 \in^2 \sigma(A),\ 1 \notin^3 \sigma(A)$$

$$2 \in \sigma(A),\ 2 \notin^2 \sigma(A)$$

💎 POLE ASSIGNMENT

ASSUMPTIONS

- The system

$$\dot{x} = Ax + Bu, \; x \in \mathbb{R}^n, \; u \in \mathbb{R}^p$$

  is controllable.

- Λ is a symmetric multiset of n complex numbers:

$$\Lambda = \{\lambda_1, \dots, \lambda_n\} \subset \mathbb{C} \; \text{ and } \; \lambda \in^k \Lambda \, \Rightarrow \, \overline{\lambda} \in^k \Lambda.$$

💎 POLE ASSIGNMENT

CONCLUSION

$\Rightarrow$ There is a matrix $K \in \mathbb{R}^{n \times m}$ such that

$$\sigma(A - BK) = \Lambda.$$

🔍 POLE ASSIGNMENT

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)

🐍 💻

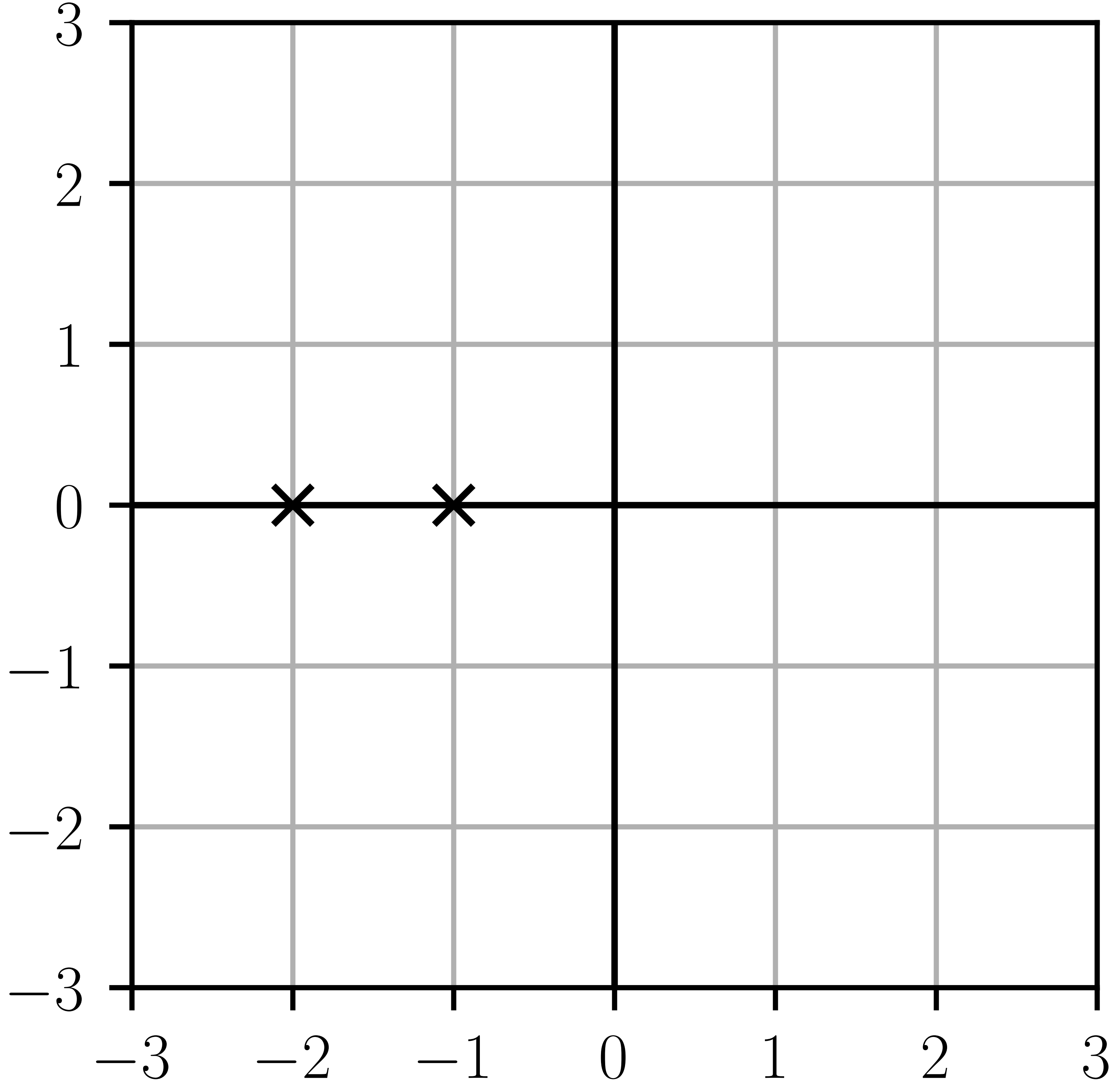
```
from scipy.signal import place_poles
A = array([[0, 1], [0, 0]])
B = array([[0], [1]])
poles = [-1, -2]
K = place_poles(A, B, poles).gain_matrix
```

```python
assert_almost_equal(K, [[2.0, 3.0]])
eigenvalues, _ = eig(A - B @ K)
assert_almost_equal(eigenvalues, [-1, -2])
```

🐍 📊 SPECTRUM

```
figure()
x = [real(s) for s in eigenvalues]
y = [imag(s) for s in eigenvalues]
plot(x, y, "kx")
xticks([-3, -2,-1, 0,1, 2,3])
yticks([-3, -2,-1, 0,1, 2,3])
plot([0, 0], [-3, 3], "k")
plot([-3, 3], [0, 0], "k")
title("Spectrum of $A-BK$"); grid(True)
```

Spectrum of $A - BK$



⚠️ LIMITATIONS

⛔ The `place_poles` function rejects eigenvalues whose multiplicity is higher than the rank of B.

In the previous example, $\operatorname{rank} B = 1$, so

- ❌ `poles = [-1, -1]` won't work.
- ✔️ `poles = [-1, -2]` will.

POLE ASSIGNMENT

Consider the system with dynamics

$$\begin{aligned}\dot{x}_1 &= x_1 - x_2 + u \\ \dot{x}_2 &= -x_1 + x_2 + u\end{aligned}$$

We apply the control law

$$u = -k_1 x_1 - k_2 x_2.$$

1. 🧮

Can we assign the poles of the closed-loop system freely by a suitable choice of k_1 and k_2?

2. 🧠

Explain this result.

🔓 POLE ASSIGNMENT

1. 🔓

$$
\begin{aligned}
A - BK &= \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix} - \begin{bmatrix} 1 \\ 1 \end{bmatrix} \begin{bmatrix} k_1 & k_2 \end{bmatrix} \\
&= \begin{bmatrix} 1 - k_1 & -1 - k_2 \\ -1 - k_1 & 1 - k_2 \end{bmatrix}
\end{aligned}
$$

$$
\begin{aligned}
\det A - BK &= \det\left(\begin{bmatrix} s-1+k_1 & 1+k_2 \\ 1+k_1 & s-1+k_2 \end{bmatrix}\right) \\
&= (s-1+k_1)(s-1+k_2) - (1+k_1)(1+k \\
&= s^2 + (k_1+k_2)s - 2(k_1+k_2)
\end{aligned}
$$

$$
\begin{aligned}
\sigma(A - BK) &= \{\lambda_1, \lambda_2\} \\
&= \{\lambda \in \mathbb{C} \mid s^2 + (k_1 + k_2)s - 2(k_1 + k_2) = 0
\end{aligned}
$$

Since the characteristic polynomial is also

$$(s - \lambda_1)(s - \lambda_2)$$

we get

$$k_1 + k_2 = -\lambda_1 - \lambda_2, \; -2(k_1 + k_2) = \lambda_1 \lambda_2$$

Thus we have

$$\lambda_1 \lambda_2 = 2(\lambda_1 + \lambda_2) \Rightarrow \lambda_2 = \frac{2\lambda_1}{\lambda_1 - 2}$$

and both poles cannot be assigned freely; for example if we select $\lambda_1 = 1$, we end up with $\lambda_2 = -2$.

2. 🔓

We have not checked the assumptions of 💎 Pole Assignment yet.

The commandability matrix is

$$[B, AB] = \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix}$$

whose rank is $1 < 2$.

Since the system is not controllable, pole assignment may fail and it does here.

PENDULUM

Consider the pendulum with dynamics:

$$m\ell^2\ddot{\theta} + b\dot{\theta} + mg\ell\sin\theta = u$$

Numerical Values:

$$m = 1.0,\ \ell = 1.0,\ b = 0.1,\ g = 9.81$$

1. 🧮

Compute the linearized dynamics of the system around the equilibrium $\theta = \pi$ and $\dot{\theta} = 0$ ($u = 0$).

2. 💻

Design a control law

$$u = -k_1(\theta - \pi) - k_2\dot{\theta}$$

such that the closed-loop linear system is asymptotically stable, with a time constant equal to 10 sec.

3. 💻 🧮

Simulate this control law on the nonlinear systems when $\theta(0) = 0.9\pi$ and $\dot{\theta}(0) = 0$.

🔓 PENDULUM

1. 🔓

Let $\Delta\theta = \theta - \pi, \omega = \dot{\theta}, \Delta\omega = \omega, \Delta u = u.$

We notice that

$$\begin{aligned} \sin\theta &= \sin(\pi + \Delta\theta) \\ &= -\sin\Delta\theta \\ &\approx -\Delta\theta \end{aligned}$$

The system dynamics can be approximated around $(\theta, \omega) = (\pi, 0)$ by

$$(d/dt)\Delta\theta = \Delta\omega$$

and

$$m\ell^2(d/dt)\Delta\omega + b\Delta\omega - mg\ell\Delta\theta = \Delta u.$$

or in standard form

$$\frac{d}{dt}\begin{bmatrix}\Delta\theta \\ \Delta\omega\end{bmatrix} = \begin{bmatrix}0 & 1 \\ g/\ell & -b/(m\ell^2)\end{bmatrix}\begin{bmatrix}\Delta\theta \\ \Delta\omega\end{bmatrix} + \begin{bmatrix}0 \\ 1/(m\ell^2)\end{bmatrix}\Delta u$$

2. 🔓

```
m = 1.0
l = 1.0
b = 0.1
g = 9.81
```

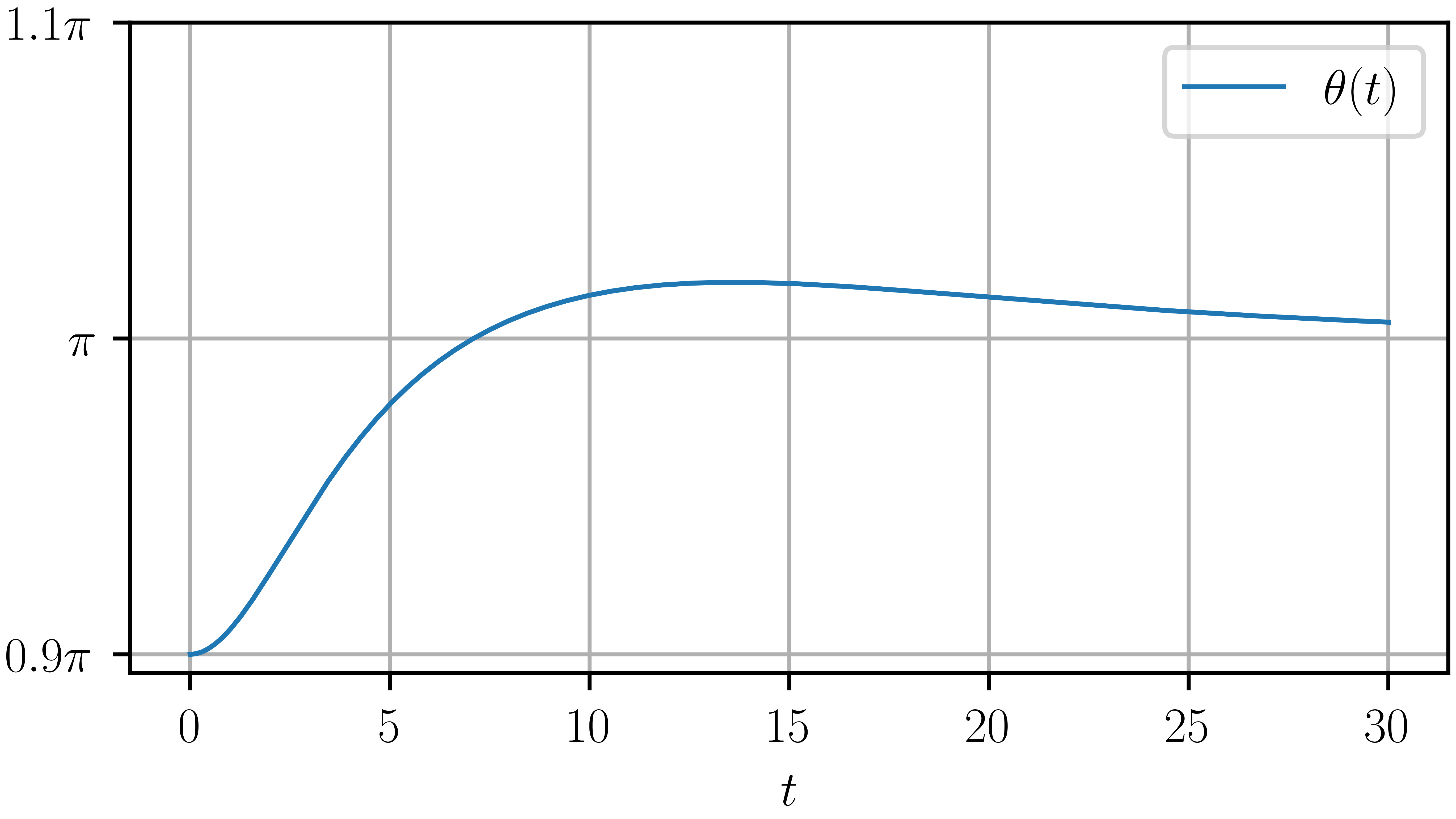
```
A = array([[  0,                 1],
           [g/l , - b/(m*l*l)]])
B = array([[          0],
           [1/(m*l*l)]])
t1, t2 = 10.0, 5.0
poles = [-1/t1, -1/t2]
K = place_poles(A, B, poles).gain_matrix
```

3. 🔓

```python
def fun(t, theta_omega):
    theta, omega = theta_omega
    Δtheta, Δomega = theta - pi, omega
    Δu = - K @ [Δtheta, Δomega]
    u = Δu[0]  # Δu has a (1,) shape
    dtheta = omega
    domega = - (g/l)*sin(theta) - b/(m*l*l)*omega \
                 + 1.0/(m*l*l)*u
    return array([dtheta, domega])
```

```
t_span = [0.0, 30.0]
y0 = [0.9*pi, 0.0]
r = solve_ivp(fun, t_span, y0, dense_output=True)
t = linspace(t_span[0], t_span[-1], 1000)
thetat, omega_t = r["sol"](t)
```

```python
figure()
plot(t, thetat, label=r"$\theta(t)$")
xlabel("$t$")
yticks([0.9*pi, pi, 1.1*pi],
       [r"$0.9\pi$", r"$\pi$", r"$1.1\pi$"])
grid(True); legend()
```



1.1π
π
0.9π
0
5
10
15
20
25
30
t
$\theta(t)$

DOUBLE SPRING

Consider the dynamics:

$$\begin{aligned} m_1 \ddot{x}_1 &= -k_1 x_1 - k_2(x_1 - x_2) - b_1 \dot{x}_1 \\ m_2 \ddot{x}_2 &= -k_2(x_2 - x_1) - b_2 \dot{x}_2 + u \end{aligned}$$

Numerical values:

$$m_1 = m_2 = 1,\ k_1 = 1, k_2 = 100,\ b_1 = 2,\ b_2 = 20$$

1. 🧮 💻

Compute the poles of the system.

Is the origin asymptotically stable?

2. 💻

Use a linear feedback to:

- kill the oscillatory behavior of the solutions,
- “speed up” the dynamics.

🔓 DOUBLE SPRING SYSTEM

1. 🔓

Let $v_1 = \dot{x}_1, v_2 = \dot{x}_2$. With the state (x_1, v_1, x_2, v_2):

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ -(k_1 + k_2)/m_1 & -b_1/m_1 & k_2/m_1 & 0 \\ 0 & 0 & 0 & 1 \\ k_2/m_2 & 0 & -k_2/m_2 & -b_2/m_2 \end{bmatrix}$$

$$B = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1/m_2 \end{bmatrix}$$

```
m1 = m2 = 1
k1 = 1; k2 = 100
b1 = 2; b2 = 20
```

```
A = array([
  [          0,      1,      0,      0],
  [-(k1+k2)/m1, -b1/m1,  k2/m1,      0],
  [          0,      0,      0,      1],
  [      k2/m2,      0, -k2/m2, -b2/m2]
])
B = array([[0.0], [0.0], [0.0], [1/m2]])
```

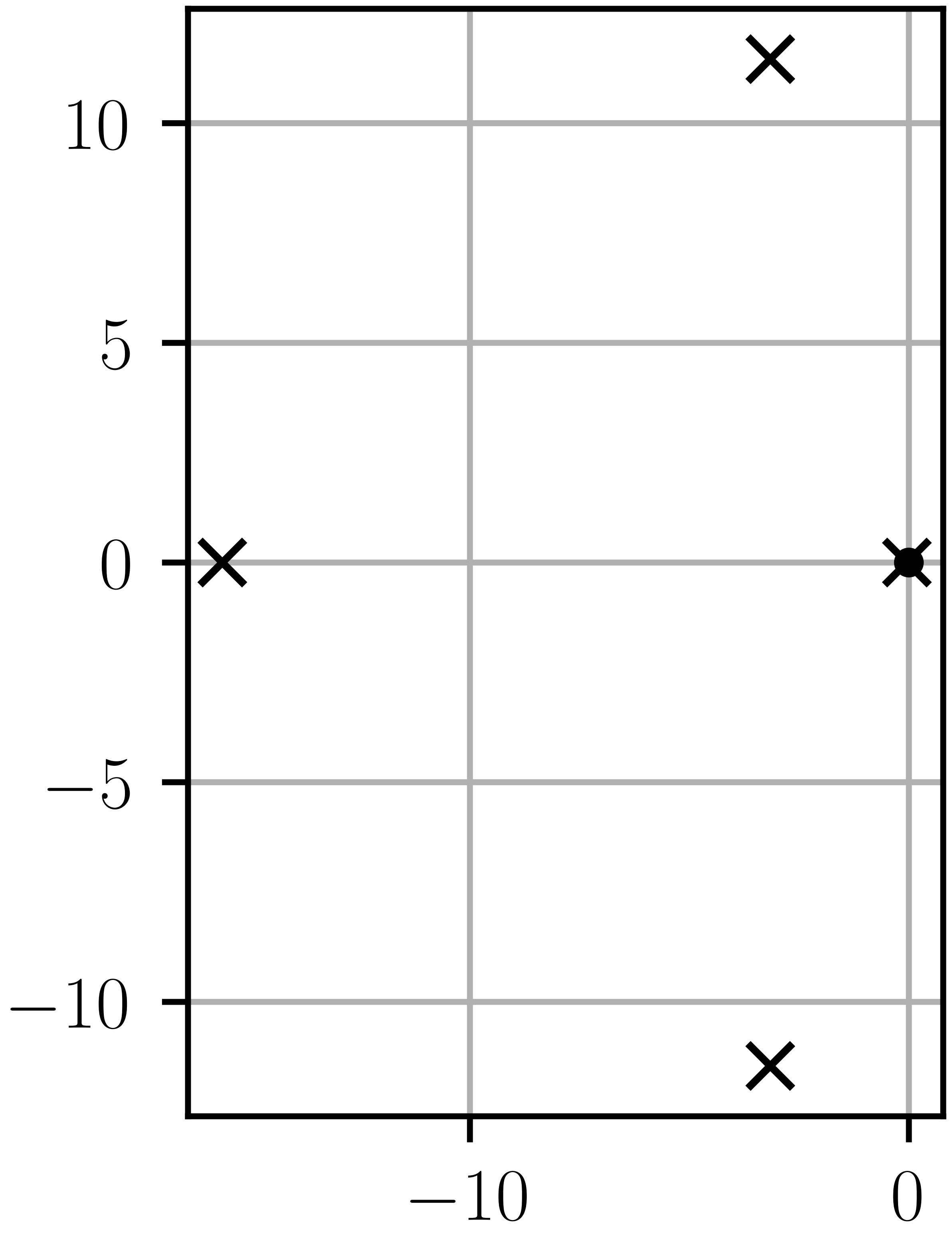
```python
eigenvalues, _ = eig(A)
```

```python
>>> eigenvalues
array([-15.64029062 +0.j          ,
        -3.15722141+11.45767938j,
        -3.15722141-11.45767938j,
        -0.04526657 +0.j          ])
```

Since all eigenvalues have a negative real part, the double-spring system is asymptotically stable.

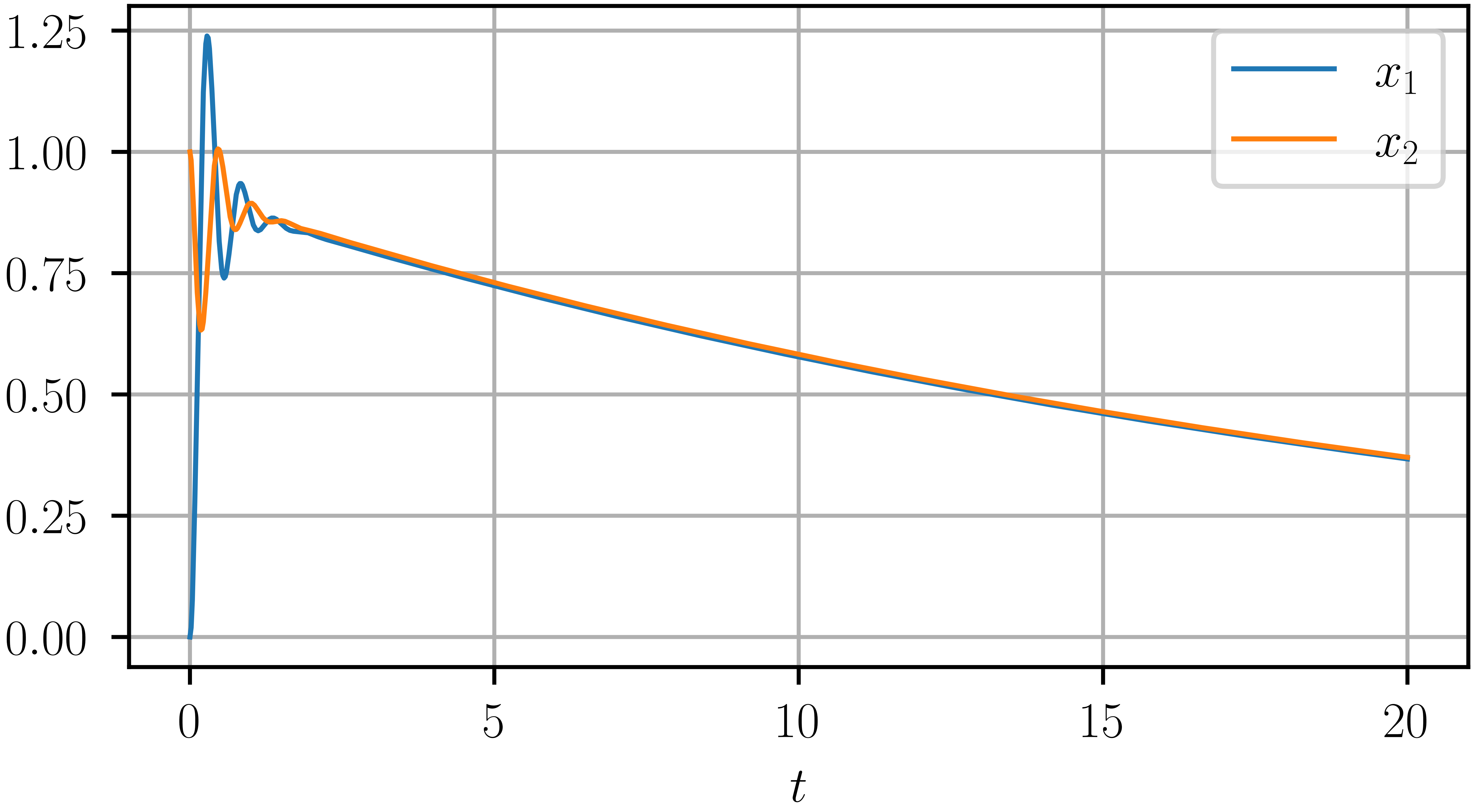
```
figure()
x = [real(s) for s in eigenvalues]
y = [imag(s) for s in eigenvalues]
plot(x, y, "kx")
plot(0.0, 0.0, "k.")
gca().set_aspect(1.0)
title("Spectrum of $A$"); grid(True)
```

Spectrum of A



```python
y0 = [0.0, 0.0, 1.0, 0.0]
t = linspace(0.0, 20.0, 1000)
yt = array([expm(A * t_) for t_ in t]) @ y0
x1t, x2t = yt[:, 0], yt[:, 2]
```

```python
figure()
plot(t, x1t, label="$x_1$")
plot(t, x2t, label="$x_2$")
xlabel("$t$")
grid(True); legend()
```


1.25
1.00
0.75
0.50
0.25
0.00
0
5
10
15
20
t
x_1
x_2

2. 🔓

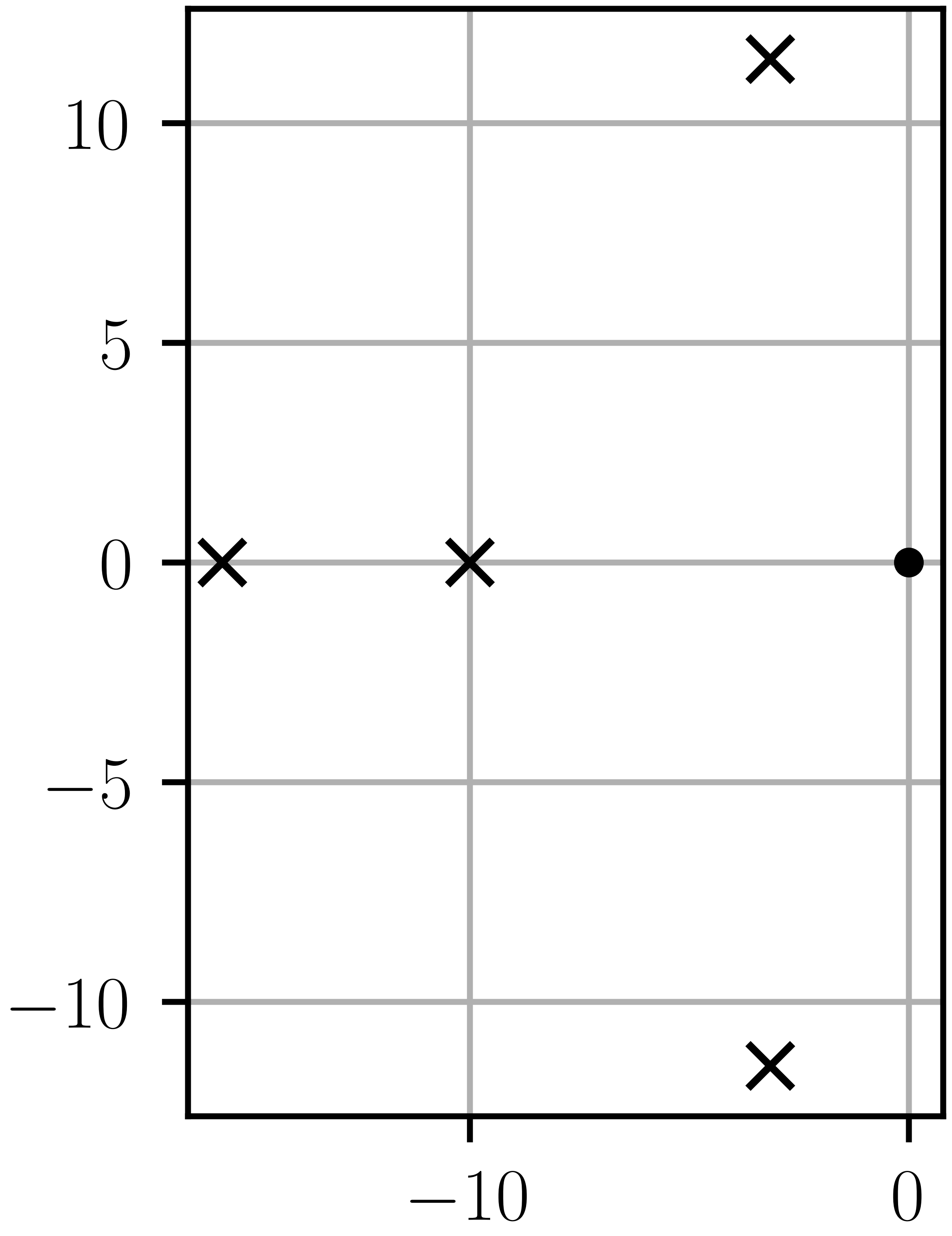
```
eigenvalues[3] = - 1 / 0.1
K = place_poles(A, B, eigenvalues).gain_matrix
print(repr(eig(A - B @ K)[0]))
```

```python
eigenvalues, _ = eig(A - B @ K)
```

```
>>> eigenvalues
array([-15.64029062 +0.j        ,
        -3.15722141+11.45767938j,
        -3.15722141-11.45767938j,
        -1.         +0.j        ])
```

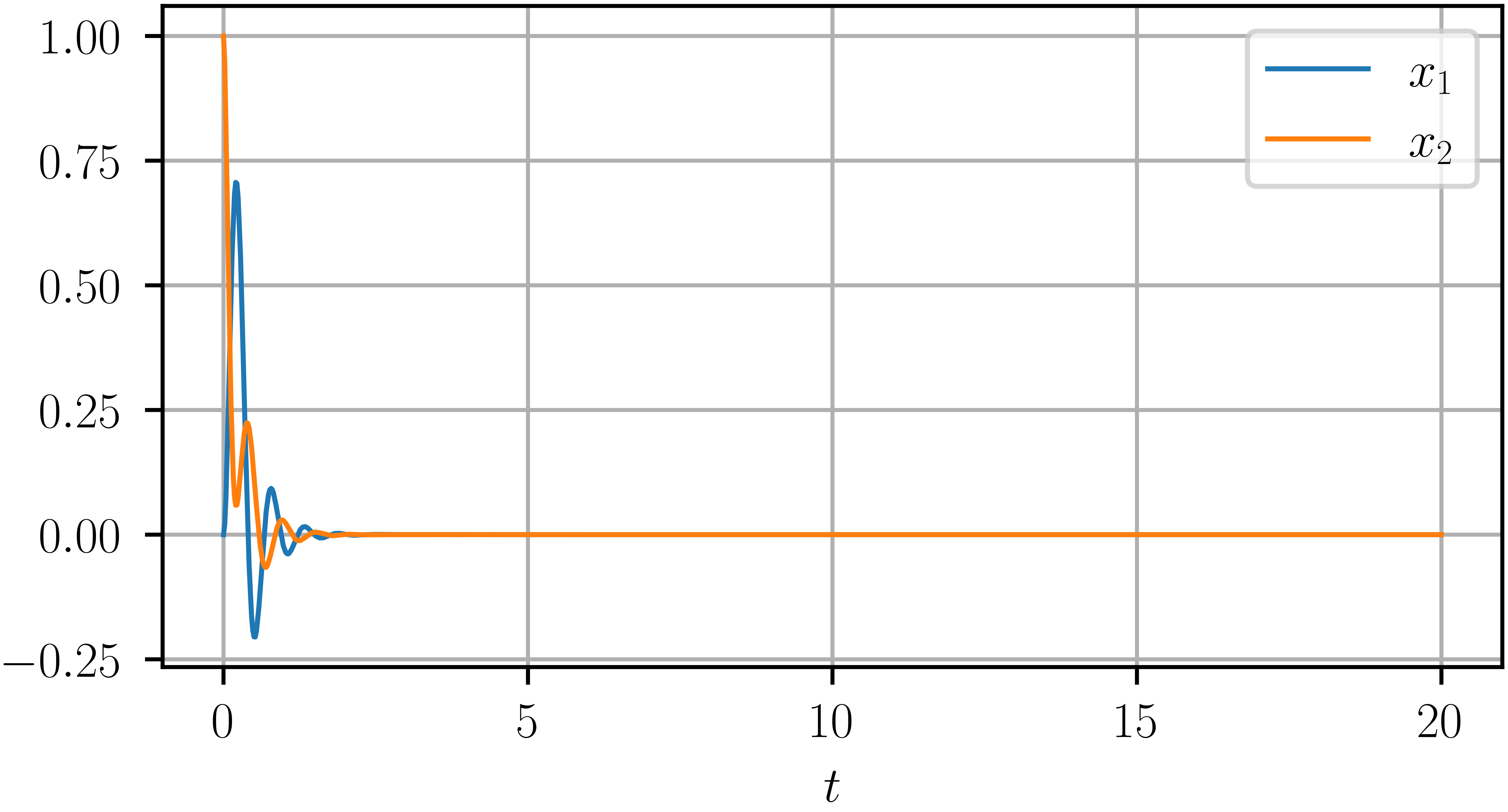
```
figure()
x = [real(s) for s in eigenvalues]
y = [imag(s) for s in eigenvalues]
plot(x, y, "kx")
plot(0.0, 0.0, "k.")
gca().set_aspect(1.0)
title("Spectrum of $A - B K$"); grid(True)
```

Spectrum of $A - BK$



```python
y0 = [0.0, 0.0, 1.0, 0.0]
t = linspace(0.0, 20.0, 1000)
yt = array([expm((A-B@K) * t_) for t_ in t]) @ y0
x1t, x2t = yt[:, 0], yt[:, 2]
```

```python
figure()
plot(t, x1t, label="$x_1$")
plot(t, x2t, label="$x_2$")
xlabel("$t$")
grid(True); legend()
```


1.00
0.75
0.50
0.25
0.00
−0.25
0
5
10
15
20
t
x_1
x_2

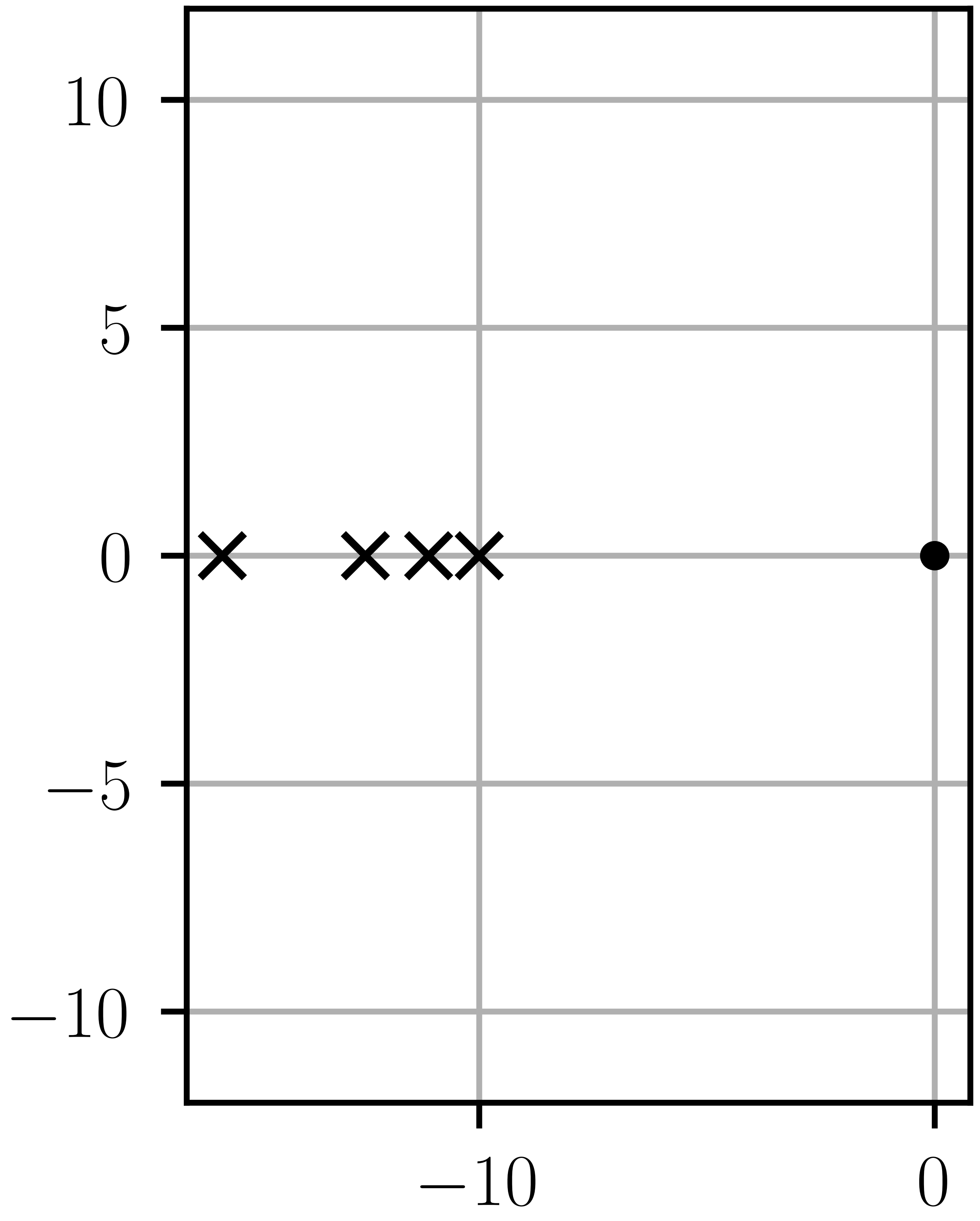
```python
eigenvalues[0] = - 1 / 0.09
eigenvalues[1] = - 1 / 0.08
K = place_poles(A, B, eigenvalues).gain_matrix
print(repr(eig(A - B @ K)[0]))
```

```python
eigenvalues, _ = eig(A - B @ K)
```

```
>>> eigenvalues
array([-15.64029062+0.j,
       -12.5       +0.j,
       -11.11111111+0.j,
       -10.        +0.j])
```

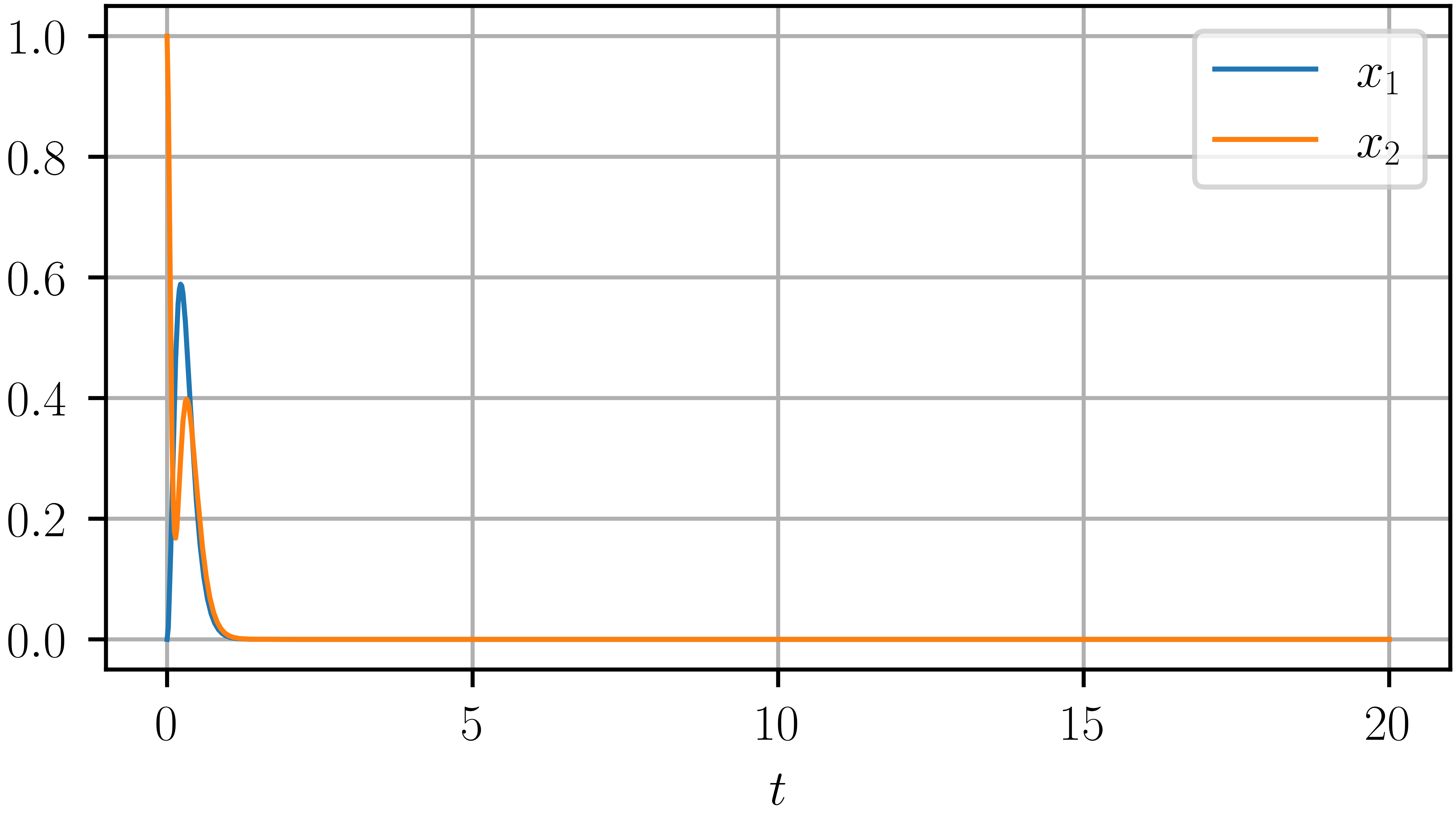
```
figure()
x = [real(s) for s in eigenvalues]
y = [imag(s) for s in eigenvalues]
plot(x, y, "kx")
plot(0.0, 0.0, "k.")
ylim(-12, 12)
gca().set_aspect(1.0)
title("Spectrum of $A - B K$"); grid(True)
```

Spectrum of $A - BK$



```python
y0 = [0.0, 0.0, 1.0, 0.0]
t = linspace(0.0, 20.0, 1000)
yt = array([expm((A-B@K) * t_) for t_ in t]) @ y0
x1t, x2t = yt[:, 0], yt[:, 2]
```

```python
figure()
plot(t, x1t, label="$x_1$")
plot(t, x2t, label="$x_2$")
xlabel("$t$")
grid(True); legend()
```



1.0
0.8
0.6
0.4
0.2
0.0
0
5
10
15
20
t
x_1
x_2