OBSERVERS

Sébastien Boisgérault, Mines ParisTech

PREAMBLE

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

OBSERVABILITY

MOTIVATION

Controling a system generally requires the knowledge of the state $x(t)$, but measuring every state variable may be impossible (or too expensive).

Can we reduce the amount of physical sensors and still be able to compute the state with “virtual” or “software” sensors ?

Control engineers call these software devices **observers**.

First we adress the mathematical feasibility of observers: **observability**.

DEFINITION

The system

$$\left|\begin{array}{lcl} \dot{x} & = & f(x) \\ y & = & g(x) \end{array}\right.$$

is **observable** if the knowledge of $y(t) = g(x(t))$ on some finite time span $[0, \tau]$ determines uniquely the initial condition $x(0)$.

REMARKS

- The knowledge of $x(0)$ determines uniquely $x(t)$ via the system dynamics.
- Later, observers will provide merely **asymptotically exact** estimates $\hat{x}(t)$ of $x(t)$, that satisfy

  $\hat{x}(t) - x(t)$ when $t \to +\infty$.

EXTENSION

The definition of observability may be extended to systems with (known) inputs u:

$$\left|\begin{array}{rcl} \dot{x} & = & f(x,u) \\ y & = & g(x,u) \end{array}\right.$$

In general, the input u may then be selected specifically to generate the appropriate $y(t)$ that allows us to compute $x(0)$.

But for linear systems, the choice of u is irrelevant.

Indeed, if

$$\left|\begin{array}{rcl} \dot{x} & = & Ax + Bu \\ y & = & Cx + Du \end{array}\right.$$

and we can deduce $x(0)$ from $y(t)$ when $u = 0$:

$$y_0(t) = Ce^{At}x(0) \quad \rightarrow \quad x(0)$$

then in the general case, when we measure

$$y_u(t) = Ce^{At}x(0) + (H * u)(t)$$

we can compute

$$y_0(t) = y_u(t) - (H * u)(t)$$

and deduce $x(0)$ at this stage.

👁 OBSERVABILITY / CAR

The position x (in meters) of a car of mass m (in kg) on a straight road is governed by

$$m\ddot{x} = u$$

where u the force (in Newtons) generated by its motor.

- we don’t know where the car is at $t = 0$,
- we don’t know what its initial speed was,
- but we know that the car doesn’t accelerate ($u = 0$).

If we measure the position $y(t) = x(t)$:

- $x(0) = y(0)$ is known,
- $\dot{x}(0) = \dot{y}(0)$ is also computable.

Thus the system is observable.

But what if we measure instead the speed $y(t) = \dot{x}(t)$?

The system dynamics $m\ddot{x}(t) = u(t) = 0$ yields

$$x(t) = x(0) + \dot{x}(0)t$$

thus

$$\dot{x}(t) = \dot{x}(0)$$

and any $x(0)$ is consistent with a measure of a constant speed.

We can’t deduce the position of the car from the measure of its speed; the system is not observable.

KALMAN CRITERION

The system $\dot{x} = Ax,\ y = Cx$ is observable iff:

$$\operatorname{rank} \begin{bmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{bmatrix} = n$$

$[C; \ldots ; CA^{n-1}]$ is the **Kalman observability matrix.**

(";" denotes the column concatenation of matrices)

DUALITY

Since

$$[C; \dots ; CA^{n-1}]^t = [C^t, \dots , (A^t)^{n-1} C^t],$$

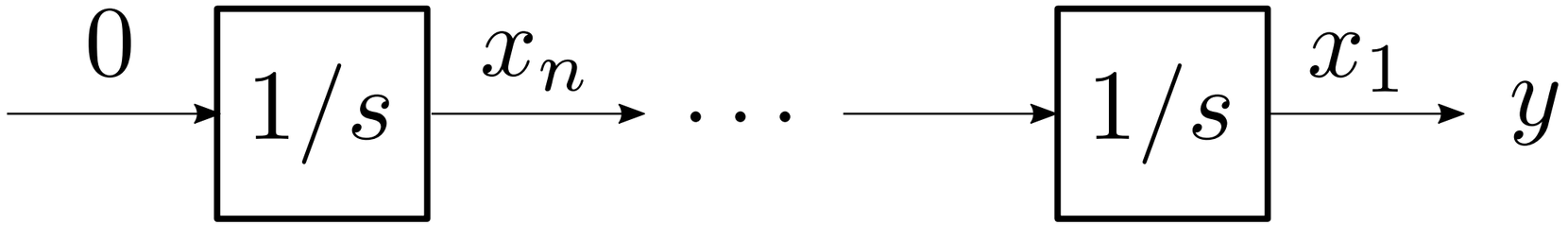
the system $\dot{x} = Ax, \;\; y = Cx$ is observable iff the system $\dot{x} = A^t x + C^t u$ is controllable.

⁉ FULLY MEASURED SYSTEM

Consider $\dot{x} = Ax,\ \ y = Cx$ with $x \in \mathbb{R}^n, y \in \mathbb{R}^p$ and $\operatorname{rank} C = n.$

- [💡, $\mathbf{x^2}$] Is the system observable ?

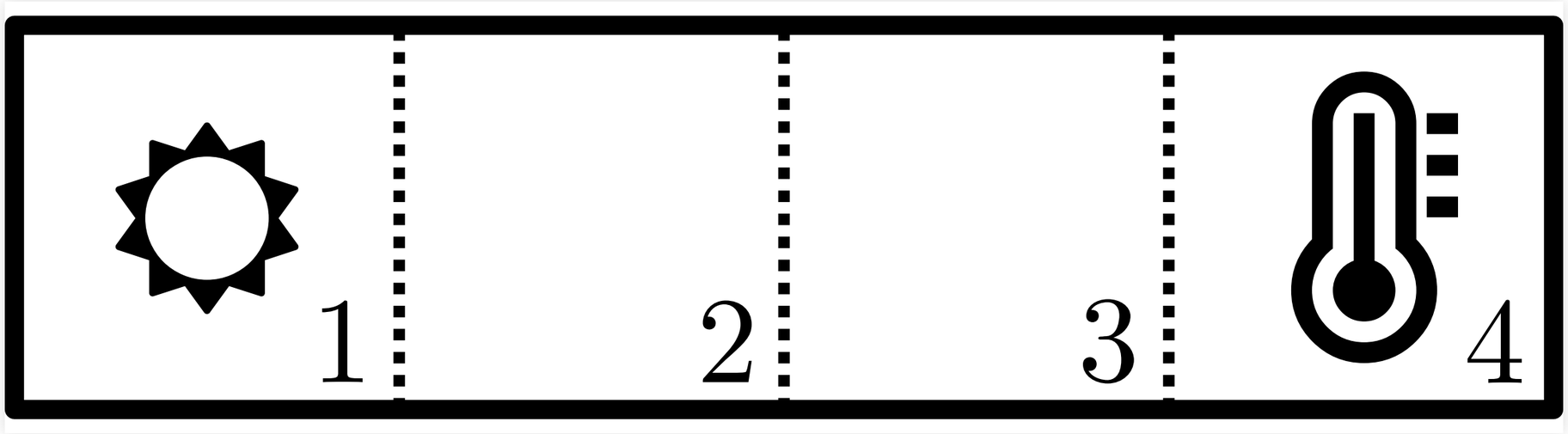
⑦ INTEGRATOR CHAIN



$$\dot{x}_n = 0,\ \dot{x}_{n-1} = x_n,\ \cdots\ , \dot{x}_1 = x_2,\ y = x_1$$

- [💡, $\mathbf{x^2}$] Show that the system is observable.

⑦ HEAT EQUATION



- $dT_1/dt = 0 + (T_2 - T_1)$
- $dT_2/dt = (T_1 - T_2) + (T_3 - T_2)$
- $dT_3/dt = (T_2 - T_3) + (T_4 - T_3)$
- $dT_4/dt = (T_3 - T_4)$
- $y = T_4$

- [💡, x^2] Show that the system is observable.
- [💡, x^2] Is it still true if the four cells are organized as a square and the temperature sensor is in any of the corners ? How many independent sensors do you need to make the system observable and where can you place them?

OBSERVER DESIGN

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

STATE OBSERVER V1

Simulate the system behavior

$$\left| \begin{aligned} \frac{d\hat{x}}{dt} &= A\hat{x} + Bu \\ \hat{y} &= C\hat{x} + Du \end{aligned} \right.$$

and since we don't know better,

$$\hat{x}(0) = 0.$$

STATE ESTIMATE ERROR

Does $\hat{x}(t)$ provide a good asymptotic estimate of $x(t)$?

The dynamics of the **state estimate error** $e = \hat{x} - x$ is

$$\begin{aligned}
\dot{e} &= \frac{d}{dt}(\hat{x} - x) \\
&= \frac{d\hat{x}}{dt} - \dot{x} \\
&= (A\hat{x} + Bu) - (Ax + Bu) \\
&= Ae
\end{aligned}$$

The state estimator error $e(t)$, solution of

$$\dot{e} = Ae$$

doesn't satisfy

$$\lim_{t \to +\infty} e(t) = 0$$

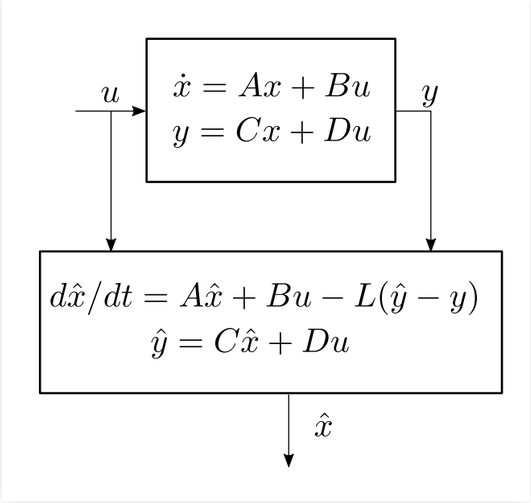
for every value of $e(0) = \hat{x}(0) - x(0)$, **unless the eigenvalues of A are in the open left-hand plane** (i.e. $\dot{x} = Ax$ is asymptotically stable).

STATE OBSERVER V2

Change the observer dynamics to account for differences between $\hat{y}$ and y (both known values):

$$\left| \begin{aligned} \frac{d\hat{x}}{dt} &= A\hat{x} + Bu - L(\hat{y} - y) \\ \hat{y} &= C\hat{x} + Du \end{aligned} \right.$$

for some **observer gain** matrix $L \in \mathbb{R}^{n \times p}$ (to be determined).


$\dot{x} = Ax + Bu$
$y = Cx + Du$
u
y
$d\hat{x}/dt = A\hat{x} + Bu - L(\hat{y} - y)$
$\hat{y} = C\hat{x} + Du$
$\hat{x}$

The new dynamics of $e = \hat{x} - x$ is

$$\begin{aligned}
\dot{e} &= \frac{d}{dt}(\hat{x} - x) \\
&= \frac{d\hat{x}}{dt} - \dot{x} \\
&= (A\hat{x} + Bu - L(C\hat{x} - Cx)) - (Ax + Bu) \\
&= (A - LC)e
\end{aligned}$$

REMINDER

The system $\dot{x} = Ax, \;\; y = Cx$ is observable

$\Longleftrightarrow$

The system $\dot{x} = A^t x + C^t u$ is commandable.

SO WHAT?

In this case, we can perform arbitrary pole assignment:

- for any conjugate set Λ of eigenvalues,
- there is a matrix $K \in \mathbb{R}^{p \times n}$ such that

$$\sigma(A^t - C^t K) = \Lambda$$

Since $\sigma(M) = \sigma(M^t)$ for any square matrix M,

$$\begin{aligned}\sigma(A^t - C^t K) &= \sigma((A - K^t C)^t) \\ &= \sigma(A - K^t C)\end{aligned}$$

OBSERVERS / POLE ASSIGNMENT

Thus, if we set

$$L = K^t$$

we have solved the pole assignment problem **for observers:**

$$\sigma(A - LC) = \Lambda$$

👁 OBSERVER/POLE ASSIGNMENT

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

$$\frac{d}{dt}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \end{bmatrix} u$$

$$y = \begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} x_1 \\ x_2 \end{bmatrix}$$

(in standard form)

```
from scipy.signal import place_poles
A = array([[0, 1], [0, 0]])
C = array([[1, 0]])
poles = [-1, -2]
K = place_poles(A.T, C.T, poles).gain_matrix
L = K.T
assert_almost_equal(K, [[3.0, 2.0]])
```

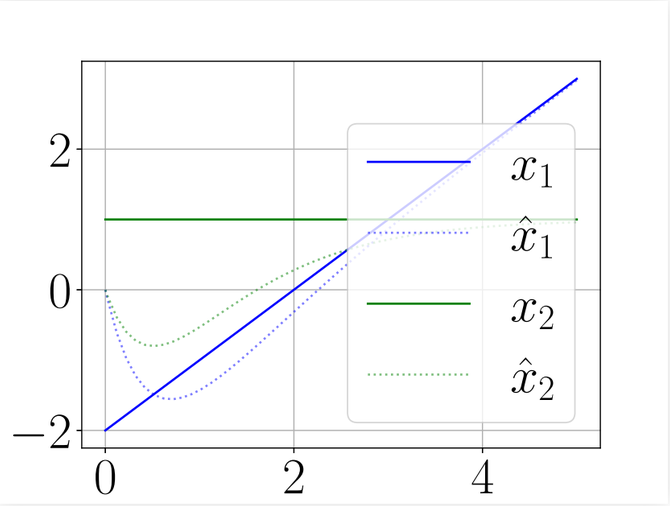
$$\frac{d}{dt}\begin{bmatrix}\hat{x}_1\\ \hat{x}_2\end{bmatrix} = \begin{bmatrix}0 & 1\\ 0 & 0\end{bmatrix}\begin{bmatrix}\hat{x}_1\\ \hat{x}_2\end{bmatrix} + \begin{bmatrix}0\\ 1\end{bmatrix}u - \begin{bmatrix}3\\ 2\end{bmatrix}(\hat{y} - y)$$

$$\hat{y} = \begin{bmatrix}1 & 0\end{bmatrix}\begin{bmatrix}\hat{x}_1\\ \hat{x}_2\end{bmatrix}$$

```
def fun(t, X_Xhat):
    x, x_hat = X_Xhat[0:2], X_Xhat[2:4]
    y, y_hat = C.dot(x), C.dot(x_hat)
    dx = A.dot(x)
    dx_hat = A.dot(x_hat) - L.dot(y_hat - y)
    return r_[dx, dx_hat]
```

```
y0 = [-2.0, 1.0, 0.0, 0.0]
result = solve_ivp(fun=fun, t_span=[0.0, 5.0],
y0=y0, max_step=0.1)
```

```
figure()
t = result["t"]
y = result["y"]
plot(t, y[0], "b", label="$x_1$")
plot(t, y[2], "b:", alpha=0.5,
label=r"$\hat{x}_1$")
plot(t, y[1], "g", label="$x_2$")
plot(t, y[3], "g:", alpha=0.5,
```


2
0
−2
0
2
4
x_1
$\hat{x}_1$
x_2
$\hat{x}_2$

KALMAN FILTERING

SETTING

Consider $\dot{x} = Ax, \;\; y = Cx$ where:

- the state $x(t)$ is unknown ($x(0)$ is unknown),
- only (a noisy version of) $y(t)$ is available.

We want a sensible estimation $\hat{x}(t)$ of $x(t)$.

We now assume the existence of state and output disturbances $v(t)$ and $w(t)$ (deviations from the exact dynamics)

$$\left| \begin{aligned} \dot{x} &= Ax + v \\ y &= Cx + w \end{aligned} \right.$$

Thes disturbances (or “noises”) are unknown; we are searching for the estimate $\hat{x}(t)$ of $x(t)$ that requires the smallest deviation from the exact dynamics to explain the data.

For a known $y(t)$, among all possible trajectories $x(t)$ of the system, find the one that minimizes

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

where:

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

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

HEURISTICS

If it is known that there is a large state disturbance but small output disturbance, it makes sense to reduce the impact of the state disturbance in the composition of J, hence to select a small Q wrt R.

OPTIMAL SOLUTION

Assume that $\dot{x} = Ax, \;\; y = Cx$ is observable.

There is a state estimation $\hat{x}(t)$, given for some $L \in \mathbb{R}^{n \times p}$ as the solution of

$$\left| \begin{array}{c} d\hat{x}/dt = A\hat{x} - L(\hat{y} - y) \\ \hat{y} = C\hat{x} \end{array} \right.$$

The dynamics of the corresponding estimation error $e(t) = \hat{x}(t) - x(t)$ is asymptotically stable.

ALGEBRAIC RICCATI EQUATION

The gain matrix L is given by

$$L = \Sigma C^t R,$$

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

$$\Sigma C^t R C \Sigma - \Sigma A^t - A\Sigma - Q^{-1} = 0.$$

OPTIMAL CONTROL $\leftrightarrow$ FILTER

Solve the Ricatti equation for optimal control with

$$(A, B, Q, R) = (A^t, C^t, Q^{-1}, R^{-1})$$

then

$$\Sigma = \Pi^t \text{ and } L = K^t.$$

KALMAN FILTER

Consider the system

$$\dot{x} = v$$

$$y = x + w$$

If we believe that the state and output perturbation are of the same scale, we may try

$$Q = [1.0], R = [1.0]$$

With $\Sigma = [\sigma]$, the filtering Ricatti equation becomes

$$\sigma^2 - 2\sigma - 1 = 0$$

whose only positive solution is

$$\sigma = \frac{2 + \sqrt{(-2)^2 - 4 \times 1 \times (-1)}}{2} = 1 + \sqrt{2}.$$

With $L = [\ell]$, we end up with

$$\ell = \sigma = 1 + \sqrt{2}.$$

Thus, the optimal filter is

$$d\hat{x}/dt = -(1 + \sqrt{2})(\hat{y} - y)$$
$$\hat{y} = \hat{x}$$

👁 STABILIZATION/KALMAN FILTER

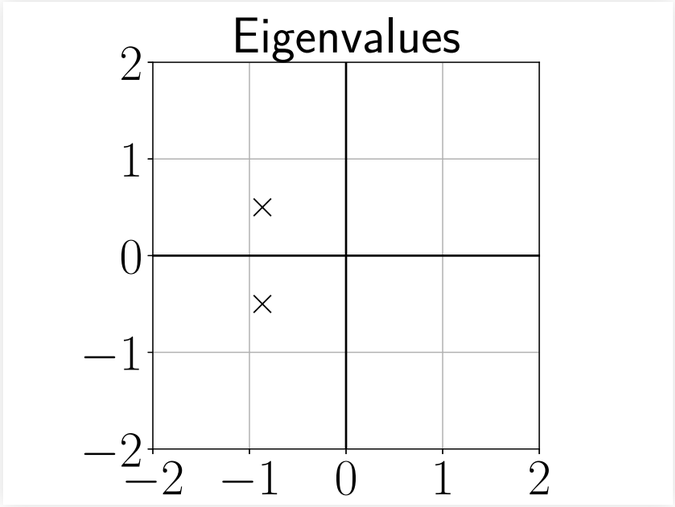
Consider the double integrator $\ddot{x} = 0, y = x.$

$$\frac{d}{dt}\begin{bmatrix} x \\ \dot{x} \end{bmatrix} = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}\begin{bmatrix} x \\ \dot{x} \end{bmatrix} + \begin{bmatrix} v_1 \\ v_2 \end{bmatrix}, y = \begin{bmatrix} 1 & 0 \end{bmatrix}\begin{bmatrix} x \\ \dot{x} \end{bmatrix} + w$$

(in standard form)

```
from scipy.linalg import solve_continuous_are
A = array([[0, 1], [0, 0]])
B = array([[0], [1]])
Q = array([[1, 0], [0, 1]]); R = array([[1]])
Sigma = solve_continuous_are(A.T, C.T, inv(Q),
inv(R))
L = Sigma @ C.T @ R
eigenvalues, _ = eig(A - L @ C)
```

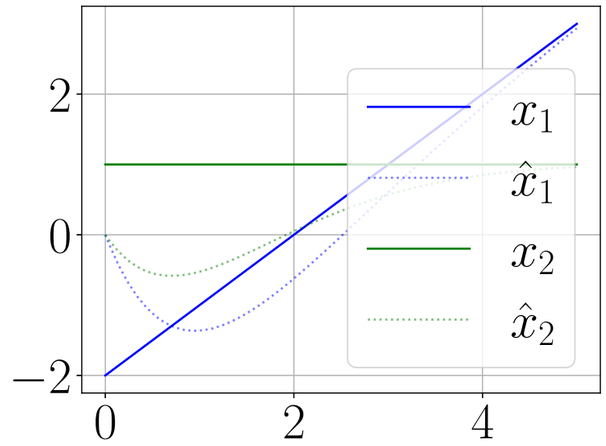
```
figure()
x = [real(s) for s in eigenvalues]
y = [imag(s) for s in eigenvalues]
plot(x, y, "kx", ms=12.0)
xticks([-2, -1, 0, 1, 2])
yticks([-2, -1, 0, 1, 2])
plot([0, 0], [-2, 2], "k")
plot([-2, 2], [0, 0], "k")
```


Eigenvalues
2
1
0
−1
−2
−2
−1
0
1
2

```
def fun(t, X_Xhat):
    x, x_hat = X_Xhat[0:2], X_Xhat[2:4]
    y, y_hat = C.dot(x), C.dot(x_hat)
    dx = A.dot(x)
    dx_hat = A.dot(x_hat) - L.dot(y_hat - y)
    return r_[dx, dx_hat]
```

```
y0 = [-2.0, 1.0, 0.0, 0.0]
result = solve_ivp(fun=fun, t_span=[0.0, 5.0],
y0=y0, max_step=0.1)
```

```
figure()
t = result["t"]
y = result["y"]
plot(t, y[0], "b", label="$x_1$")
plot(t, y[2], "b:", alpha=0.5,
label=r"$\hat{x}_1$")
plot(t, y[1], "g", label="$x_2$")
plot(t, y[3], "g:", alpha=0.5,
```


2
0
−2
0
2
4
x_1
$\hat{x}_1$
x_2
$\hat{x}_2$