*2A - Automatique*

Chapter 7

# Control Science (AUT)

## State-Space Approaches

October 2019

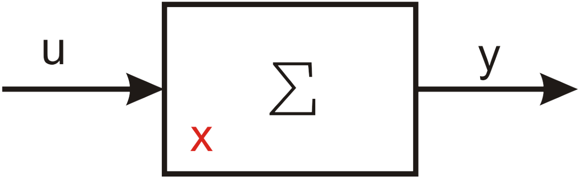
Romain Bourdais
CentraleSupélec
romain.bourdais@centralesupelec.fr

## Course outline

- Reminder on State-Space models
- From TF to SS
- Controllability and Observability
- Pole placement by state-feedback control
- Observer design using pole placement

# Outline

# Context
## Linear systems




- $x$ : state : integrates the system's past history

### Linear systems (Time Varying)

$$\left\{ \begin{array}{rcl} \dot{x}(t) & = & A(t)x(t) + B(t)u(t) \\ y(t) & = & C(t)x(t) + D(t)u(t) \end{array} \right.$$

- $A$ : state matrix, size $n \times n$
- $B$ : input matrix, size $n \times m$
- $C$ : output matrix, size $p \times n$
- $D$ : direct transfer matrix , size $p \times m$

## Linear systems (Time Varying)

$$\left\{ \begin{array}{rcl} \dot{x}(t) & = & A(t)x(t) + B(t)u(t) \\ y(t) & = & C(t)x(t) + D(t)u(t) \end{array} \right.$$

- $A$ : state matrix, size $n \times n$
- $B$ : input matrix, size $n \times m$
- $C$ : output matrix, size $p \times n$
- $D$ : direct transfer matrix , size $p \times m$

## The equivalent block diagram

# From nonlinear system to linear model
## Linearization around equilibrium points !

- Given the nonlinear differential equation (with $f$ locally Lipschitz [1]) :

$$\dot{x} = f(x, u)$$

- $(x_e, u_e)$ is an equilibrium point : $f(x_e, u_e) = 0$
- We can define the linearized model around this equilibrium point $\tilde{x} = x - x_e$, $\tilde{u} = u - u_e$

$$\dot{\tilde{x}} = \tilde{A}\tilde{x} + \tilde{B}\tilde{u}$$

with $\tilde{A} = \left.\frac{\partial f}{\partial x}(x, u)\right|_{x=x_e, u=u_e}$ and $\tilde{B} = \left.\frac{\partial f}{\partial u}(x, u)\right|_{x=x_e, u=u_e}$

- $\tilde{A}$ is known as the Jacobian matrix

1. for fun, try $\dot{x} = \sqrt{x}$

# Solution of the differential state equation
## Reminder of « Modeling course »

$$\dot{x}(t) = A(t)x(t) + B(t)u(t), \quad x(t_0) = x_0, \quad u(t) \text{ inconnu}$$

- Homogeneous part $\dot{x}(t) = A(t)x(t)$.
  - $A(t)$ continuous : there exists a solution
  $$x(t) = \Phi(t, t_0)x(t_0)$$
  - $\Phi(t, t_0)$ : transition matrix

### Transition matrix $\Phi(t, t_0)$

- defined by :
$$\begin{cases} \dot{\Phi}(t, t_0) & = & A(t)\Phi(t, t_0) \\ \Phi(t_0, t_0) & = & I_n \end{cases}$$
- Its properties :
$$\begin{cases} \Phi(t_2, t_0) & = & \Phi(t_2, t_1)\Phi(t_1, t_0) \\ \Phi^{-1}(t_1, t_0) & = & \Phi(t_0, t_1) \end{cases}$$

## Solution of the differential state equation

### Reminder of « Modeling course »

$$\dot{x}(t) = A(t)x(t) + B(t)u(t), \quad x(t_0) = x_0, \quad u(t) \text{ inconnu}$$

- Non homogeneous part. We are looking for a particular solution :

$$x_p(t) = \Phi(t, t_0)S(t), \quad S(t_0) = 0$$

- Finally :

$$S(t) = \int_{t_0}^{t} \Phi(t_0, \tau)B(\tau)u(\tau)\mathrm{d}\tau$$

# Solution of the differential state equation
## The LTI case

- In the general case, determining $\Phi$ is not possible analytically, but ...
- For LTO systems :

$$\Phi(t, t_0) = e^{A(t-t_0)}$$

- Then the solution becomes :

$$\boxed{x(t) = e^{A(t-t_0)}x_0 + \int_{t_0}^{t} e^{A(t-\tau)}Bu(\tau)d\tau} \tag{1}$$

# State-space model
## Non uniqueness of the state model

- A transfer function is unique, but not the state space representation

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

- What is the transfer ?

# State-space model
## Non uniqueness of the state model

- A transfer function is unique, but not the state space representation

$$\left\{\begin{array}{lcl} \dot{x}(t) & = & Ax(t) + Bu(t) \\ y(t) & = & Cx(t) \end{array}\right.$$

- We proceed to the change of basis : $x = Tx_1$
- What is the state-space model in the new basis ?
- What is the transfer function ?

# From continous to discrete

- Let's start with the continuous system :

$$\begin{cases} \dot{x}(t) & = & Ax(t) + Bu(t) \\ y(t) & = & Cx(t) + Du(t) \end{cases}$$

- Using ADC/DAC with a sampling time $T_e$

$$\begin{cases} x_{k+1} & = & A_d x_k + B_d u_k \\ y_k & = & C_d x_k + D_d u_k \end{cases}$$

- What is the relation between $A_d, B_d, C_d, D_d$ and $A, B, C, D, Te$?

# System under state-space representation
## A remark on stability

$$\begin{cases} \dot{x}(t) & = & Ax(t) + Bu(t) \\ y(t) & = & Cx(t) \end{cases}$$

- The stability is given by $A$

### Necessary and sufficient condition for asymptotic stabilty

The system is asymptotically stable if and only if all eigenvalues of $A$ have a strictly negative real part.

# Outline

1. Introduction
2. Reminder on State-Space models
3. **From Transfer Function to State-Space model**
4. Controllability and Observability
5. Pole placement by state feedback
6. Observer design
7. Conclusion

# From Transfer Function to State-Space model

## Controller form

- Let us consider the system described by the transfer :

$$\frac{Y}{U}(p) = \frac{b_0 + b_1 p + \ldots + b_{n-1} p^{n-1}}{a_0 + a_1 p + \ldots + a_{n-1} p^{n-1} + p^n}$$

- denominateur degree : $n$ This is the system order !
- $x \in \mathbb{R}^n$

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 1 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & 0 \\ 0 & 0 & 0 & \ldots & 0 & 1 \\ -a_0 & -a_1 & -a_2 & \ldots & \ldots & -a_{n-1} \end{pmatrix}, \quad B = \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix}$$

$$C = \begin{pmatrix} b_0 & b_1 & b_2 & \ldots & b_{n-1} \end{pmatrix}, \quad D = 0$$

# From Transfer Function to State-Space model

## Controller form : remark on the link between poles and eigenvalues

$$A = \begin{pmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \\ -a_0 & -a_1 & -a_2 & \dots & \dots & -a_{n-1} \end{pmatrix}$$

- From SS $\rightarrow$ TF

$$\frac{Y}{U} = C\,(pI_n - A)^{-1}\,B$$

- $(pI_n - A)^{-1} = \frac{\star}{\det(pI_n - A)}$
- What is $\det(pI_n - A)$ ?

- We get the denominator of the transfer function back !

# From Transfer Function to State-Space model
## Observer form

- Let us consider the system described by the transfer :

$$\frac{Y}{U}(p) = \frac{b_0 + b_1 p + \ldots + b_{n-1} p^{n-1}}{a_0 + a_1 p + \ldots + a_{n-1} p^{n-1} + p^n}$$

- denominator degree : $n$ This is the system order !
- $x \in \mathbb{R}^n$

$$A = \begin{pmatrix} 0 & 0 & 0 & \ldots & 0 & -a_0 \\ 1 & 0 & 0 & \ldots & 0 & -a_1 \\ 0 & 1 & 0 & \ldots & 0 & -a_2 \\ \vdots & \vdots & \vdots & \vdots & \vdots & -a_i \\ 0 & 0 & 0 & \ldots & 1 & -a_{n-1} \end{pmatrix}, \quad B = \begin{pmatrix} b_0 \\ b_1 \\ b_2 \\ \vdots \\ b_{n-1} \end{pmatrix}$$

$$C = \begin{pmatrix} 0 & 0 & 0 & \ldots & 0 & 1 \end{pmatrix}, \quad D = 0$$

# From Transfer Function to State-Space model
## Modal form

- Let us consider the system described by the transfer :

$$\frac{Y}{U}(p) = \frac{b_0 + b_1 p + \ldots + b_{n-1} p^{n-1}}{a_0 + a_1 p + \ldots + a_{n-1} p^{n-1} + p^n}$$

- We proceed to the partial fraction decomposition

$$\frac{Y}{U}(p) = \sum \frac{\alpha_i}{p - p_i}$$

- We can proceed element by element

- Remark : It is also working for more complex elements

# From Transfer Function to State-Space model
## Your turn to play

- Let us consider the system descibed by the following transfer function :

$$\frac{Y}{U}(p) = \frac{(p+2)}{p(p+1)(p+3)}$$

# Outline

# Controllability
## Definition and theorem





### Controllable system

- A system is said to be controllable if for any couple $(x_0, x_f)$, there exists a control action $u(t)$ that brings the system from $x_0$ to $x_f$ in a finite time.
- It is partially controllable if not !

### Controllability matrix

For the $n$-order linear system

$$\left\{ \begin{array}{lcl} \dot{x}(t) & = & Ax(t) + Bu(t) \\ y(t) & = & Cx(t) \end{array} \right.$$

The controllability matrix is defined by :

$$\mathcal{C} = \left( B, AB, A^2B, \ldots, A^{n-1}B \right)$$

### Necessary and sufficient condition for controllability

$$\mathrm{rank}(\mathcal{C}) = n$$

# Controllability

## Controller Canonical decomposition





- $\mathrm{rang}(\mathcal{C}) = n_1 < n$ : partially controllable system
- There exists a change of basis :

$$\bar{x} = Tx = \begin{pmatrix} T_1 \\ T_2 \end{pmatrix} x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

- with $x_1 \in \mathbb{R}^{n_1}$ that leads to :

$$\begin{cases} \dot{\bar{x}} & = & \bar{A}\bar{x} + \bar{B}u \\ y & = & \bar{C}\bar{x} \end{cases}$$

with :

$$\bar{A} = TAT^{-1} = \begin{pmatrix} \bar{A}_{11} & \bar{A}_{12} \\ 0 & \bar{A}_{22} \end{pmatrix}, \quad \bar{B} = TB = \begin{pmatrix} \bar{B}_1 \\ 0 \end{pmatrix}, \quad \bar{C} = CT = (\bar{C}_1 \ \bar{C}_2)$$

- The controllable modes are the eigenvalues of $\bar{A}_{11}$

# Controllability
## Looking for the change of basis



Romain Bourdais

CentraleSupélec



- Let's start with the controllability matrix

$$\mathcal{C} = \left(S_{n_1}, S_{n_1}K\right)$$

- Let's build $S_{n-n_1}$ so that $S = \left(S_{n_1}, S_{n-n_1}\right)$ is inversible
- Let us denote $T^{-1} = S$
- We determine $T = \begin{pmatrix} T_1 \\ T_2 \end{pmatrix}$ : This is the expected change of basis !
- We get $T_1 S_{n_1} = I_{n_1}$ et $T_2 S_{n_1} = 0$
- Then we have

$$T\mathcal{C} = \begin{pmatrix} T_1 S_{n_1} & T_1 S_{n_1} K \\ T_2 S_{n_1} & T_2 S_{n_1} K \end{pmatrix} = \begin{pmatrix} I_r & K \\ 0 & 0 \end{pmatrix}$$

# Controllability

## Looking for the change of basis - your turn to play

- Let us consider the following system :

$$\begin{cases} \dot{x}_1(t) & = & -x_1(t) + 2x_2(t) + u(t) \\ \dot{x}_2(t) & = & x_1(t) - 2x_2(t) - u(t) \\ y(t) & = & x_1(t) \end{cases}$$

- Rewrite the system into a state-space representation
- Proceed to the controllability canonical decomposition

# Controllability

## Looking for the change of basis - Is Matlab Bruce Wayne ?

- Let us consider the following system :

$$\begin{cases} \dot{x}_1(t) & = & -x_1(t) + 2x_2(t) + u(t) \\ \dot{x}_2(t) & = & x_1(t) - 2x_2(t) - u(t) \\ y(t) & = & x_1(t) \end{cases}$$

### The code

- The system : the function `ss`
  - `A=[-1 2 ; 1 -2];`
  - `B=[1 ; -1];`
  - `C=[1 ; 0];`
  - `D=0;`
  - `sys=ss(A,B,C,D)`
- Controllability matrix : the function `ctrb`
  - `ControllabilityMatrix=ctrb(A,B)`
  - The rank `rank(ctrb(A,B))`
- Controllability canonical decomposition : the function `ctrbf`
  - `[ABAR,BBAR,CBAR,T,K] = ctrbf(A,B,C)`
  - See the help ! `help ctrbf`

# Controllability

## Looking for the change of basis - Is Matlab Bruce Wayne ?

```
[ABAR,BBAR,CBAR,T,K] = ctrbf(A,B,C) returns a decomposition
into the controllable/uncontrollable subspaces.

[ABAR,BBAR,CBAR,T,K] = ctrbf(A,B,C,TOL) uses tolerance TOL.

If Co=CTRB(A,B) has rank r <= n = SIZE(A,1), then there is a
similarity transformation T such that

   Abar = T * A * T' ,  Bbar = T * B ,  Cbar = C * T'

and the transformed system has the form

         | Anc    0 |           | 0 |
   Abar =  ----------  ,  Bbar =  ---  ,  Cbar = [Cnc| Cc].
         | A21   Ac |           |Bc |
                                            -1          -1
where (Ac,Bc) is controllable, and Cc(sI-Ac)Bc = C(sI-A)B.
```

- Be careful : $x_c$ and $x_{uc}$ are not arranged in the same order
- Matlab gives us ITS change of basis

# Controllability

## Looking for the change of basis - Is Matlab Bruce Wayne ?

```
>>  [ABAR,BBAR,CBAR,T,K] = ctrbf(A,B,C)

ABAR =

         0         0
    1.0000   -3.0000


BBAR =

         0
    1.4142


CBAR =

    0.7071    0.7071


T =

    0.7071    0.7071
    0.7071   -0.7071


K =

     1     0
```

- Matlab provides a result . . .
- But we lose any possible physical interpretation of ITS state

# Observability
## Definition and theorem


Romain Bourdais

CentraleSupélec



### Definition

- A system is said to be observable if from input and output observations over any finite time interval, we can determine the initial state $x_0$.

### Observability matrix

The observability matrix is defined by :

$$\mathcal{O} = \begin{pmatrix} C \\ CA \\ \vdots \\ CA^{n-1} \end{pmatrix}$$

### Necessary and sufficient condition for observability

$$\mathrm{rank}(\mathcal{O}) = n$$

- It's the dual of the controllability : $(A, B) \rightarrow (A^T, C^T)$

# Observability

## Observability canonical decomposition

- $\mathrm{rang}(\mathcal{O}) = n_1 < n$ : partially observable system
- There exists a change of basis :

$$\bar{x} = Tx = \begin{pmatrix} T_1 \\ T_2 \end{pmatrix} x = \begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$$

- with $x_1 \in \mathbb{R}^{n_1}$ that leads to :

$$\begin{cases} \dot{\bar{x}} &= \bar{A}\bar{x} + \bar{B}u \\ y &= \bar{C}\bar{x} \end{cases}$$

with :

$$\bar{A} = TAT^{-1} = \begin{pmatrix} \bar{A}_{11} & 0 \\ \bar{A}_{21} & \bar{A}_{22} \end{pmatrix}, \quad \bar{B} = TB = \begin{pmatrix} \bar{B}_1 \\ \bar{B}_2 \end{pmatrix}, \quad \bar{C} = CT = \begin{pmatrix} \bar{C}_1 & 0 \end{pmatrix}$$

- The observable modes are the eigenvalues of $\bar{A}_{11}$

### Using Matlab

- `obsv, obsvf`

# Minimal realization of the system
## We only keep things that link input to output !



For a $n$-order linear system :

$$\begin{cases} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{cases}$$

- We proceed to the OCD then to the CCD
- We get the following form (The Kalman form) :

$$\begin{cases} \dot{\tilde{x}}(t) &= \left(\begin{array}{c|c|c|c} A_{11,c} & A_{12,c} & 0 & 0 \\ \hline 0 & A_{11,nc} & 0 & 0 \\ \hline \star & \star & \star & \star \\ \hline 0 & \star & 0 & \star \end{array}\right) \begin{pmatrix} x_{o,c} \\ x_{o,nc} \\ x_{no,c} \\ x_{no,nc} \end{pmatrix} + \left(\begin{array}{c} B_{1,c} \\ \hline 0 \\ \hline B_{2,c} \\ \hline 0 \end{array}\right) u \\ y &= \left(\begin{array}{c|c|c|c} C_{1,c} & C_{1,nc} & 0 & 0 \end{array}\right) x \end{cases}$$

# Minimal realization of the system

## Reduced transfer function

- Let us consider a system described by $(A, B, C)$.
- After getting the Kalman form, its transfer function is :

$$H(q) = C_{1,c}\left(pI - A_{11,c}\right)^{-1} B_{1,c}$$

- We only keep the observable and controllable part !

### Minimal realization

$(A, B, C)$ is a minimal realization if the pair $(A, B)$ is controllable and if the pair $(A, C)$ is observable.

# Some exercices

## Just to think a little

We consider these 4 systems, whose transfer functions are as follows :

| | |
|---|---|
| $H_1(p) = \frac{Y}{U}(p) = \frac{K}{p(1+\tau p)}$ | $H_2(p) = \frac{Y}{U}(p) = \frac{a_0+a_1 p}{b_0+b_1 p+p^2}$ |
| $H_3(p) = \frac{Y}{U}(p) = \frac{1+Tp}{1+aTp}$ | $H_4(p) = \frac{Y}{U}(p) = K(1+Tp)$ |

- For each system, determine both controllable form and observable form
- For each form, check the controllability and observability
- Discuss the conclusions according to the different parameters

## Outline

Romain Bourdais

CentraleSupélec

# Problem statement
## The SISO case : one input, one output


### LTI SISO system

$$\begin{cases} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{cases}$$

- $A$ : size $n \times n$
- $B$ : size $n \times 1$
- $C$ : size $1 \times n$

### What we want

- Design a control law :

$$u = ly^r - Kx$$

- $y^r$ : reference
- $l \in \mathbb{R}$ and $K \in \mathbb{R}^{1\times n}$

# Problem statement
## What we have

### Before correction : system analysis

$$\begin{cases} \dot{x}(t) &= Ax(t) + Bu(t) \\ y(t) &= Cx(t) \end{cases}$$

- The system poles are the eigenvalues of $A$
- The characteristic polynomial of $A$ is :

$$\mathcal{P}_A(\lambda) = a_0 + a_1\lambda + \ldots + a_{n-1}\lambda^{n-1} + \lambda^n$$

- The transfer function is :

$$\frac{Y}{U} = C\left(pI_n - A\right)^{-1} B$$

- Assumption : the system is controllable !

# Problem statement
## What we want :

### After correction (closed-loop behavior) :

$$\begin{cases} \dot{x}(t) & = & (A - BK)x(t) + Bly^r(t) \\ y(t) & = & Cx(t) \end{cases}$$

- The poles of the closed-loop system are the eigenvalues of $(A - BK)$
- The characteristic polynomial of $(A - BK)$ is :

$$\mathcal{P}_{A-BK}(\lambda) = \beta_0 + \beta_1\lambda + \ldots + \beta_{n-1}\lambda^{n-1} + \lambda^n$$

- The transfer function is :

$$\frac{Y}{Y^r} = C\left(pI_n - (A - BK)\right)^{-1} Bl$$

### 1st observation

- We want to impose the closed-loop behavior
- As a consequence, all the coefficients $\beta_i$ are known !

# Problem statement
## What we are looking for

- We want a control law with the following expression :

$$u = ly^r - Kx$$

- We need to determine $K$ and $l$

### 1st step

- The coefficients $\beta_i$ are given (by assumption)
- We need to determine $K$ so that the poles of the closed-loop system are the expected ones.
- We will have a system of equations to solve. The solution will be unique if the system is controllable (SISO case)

### 2nd step

- The matrix $K$ is determined
- Adjustment of the gain $l$ so that the static gain of the closed-loop system is 1

# One way to solve - by hand

## Step 1.0

$$\frac{Y}{U}(p) = \frac{b_0 + b_1 p + \ldots + b_{n-1}p^{n-1}}{a_0 + a_1 p + \ldots + a_{n-1}p^{n-1} + p^n}$$

- Choice of a state-space representation (for example controller form) :

$$\dot{x} = \begin{pmatrix} 0 & 1 & 0 & 0 & \ldots & 0 \\ 0 & 0 & 1 & 0 & \ldots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & 0 \\ 0 & 0 & 0 & \ldots & 0 & 1 \\ -a_0 & -a_1 & -a_2 & \ldots & \ldots & -a_{n-1} \end{pmatrix} x + \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} u$$

$$y = \begin{pmatrix} b_0 & b_1 & b_2 & \ldots & b_{n-1} \end{pmatrix} x$$

# One way to solve - by hand
## Step 1.1

- $A - BK$ is :

- Then the characteristic polynomial of $A - BK$ is :

$$\mathcal{P}_{A-BK}(\lambda) = (a_0 + k_0) + (a_1 + k_1)\lambda + \ldots + (a_{n-1} + k_{n-1})\lambda^{n-1} + \lambda^n$$

# One way to solve - by hand
## Step 1.2

- Then the characteristic polynomial of $A - BK$ is :

$$\mathcal{P}_{A-BK}(\lambda) = (a_0 + k_0) + (a_1 + k_1)\lambda + \ldots + (a_{n-1} + k_{n-1})\lambda^{n-1} + \lambda^n$$

- That we need to make equals to the expected one :

$$\mathcal{P}_{A-BK}(\lambda) = \beta_0 + \beta_1\lambda + \ldots + \beta_{n-1}\lambda^{n-1} + \lambda^n$$

- We can proceed to a term-by-term identification :

$$k_i = \beta_i - a_i$$

# One way to solve - by hand
## Short Break : A picture before Step 2

- The state-space model is now :

$$\dot{x} = \begin{pmatrix} 0 & 1 & 0 & 0 & \dots & 0 \\ 0 & 0 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots & 0 \\ 0 & 0 & 0 & \dots & 0 & 1 \\ -\beta_0 & -\beta_1 & -\beta_2 & \dots & \dots & -\beta_{n-1} \end{pmatrix} x + \begin{pmatrix} 0 \\ 0 \\ \vdots \\ 0 \\ 1 \end{pmatrix} l y^r$$

$$y = \begin{pmatrix} b_0 & b_1 & b_2 & \dots & b_{n-1} \end{pmatrix} x$$

- The transfer function is :

$$\frac{Y}{Y^r}(p) = l \frac{b_0 + b_1 p + \dots + b_{n-1} p^{n-1}}{\beta_0 + \beta_1 p + \dots + \beta_{n-1} p^{n-1} + p^n}$$

### Important Consequence

- A state-feedback does not modify the zeros of the system !

# One way to solve - by hand
## Step 2 : adjustment of the gain $l$

### 1st method

- The matrix $K$ is determined ; the closed-loop transfer function becomes :

$$\frac{Y}{Y^r}(p) = l \frac{b_0 + b_1 p + \ldots + b_{n-1} p^{n-1}}{\beta_0 + \beta_1 p + \ldots + \beta_{n-1} p^{n-1} + p^n}$$

- As we want a static gain equals to 1, (the CL system is AS, we can get it posing $p = 0$)

$$l = \frac{\beta_0}{b_0}$$

### Alternative method

- The matrix $K$ is determined ; the closed-loop transfer function becomes :

$$\frac{Y}{Y^r} = C\left(pI_n - (A - BK)\right)^{-1} Bl$$

- As we want a static gain equals to 1, (the CL system is AS, we can get it posing $p = 0$)

$$l = \frac{-1}{C(A - BK)^{-1} B}$$

# Pole placement by state feedback
## The control diagram

# State feedback
## Where are the difficulties ?

### How to get access to the state ?

- We need to implement a state obserer !
- Wait for the next course !

### What is the impact of a model error ?

- We may add an integral action !
- Not done here . . .

### How to choose the expected poles ?

- If the system is controllable, can we do what we want ?
- What about the margins ?

# Pole placement
## The big diagram

- We need to define $n$ poles ! All of the them have to be stable !
- Dominant pole method : we define the two slowest poles and we add faster ones
- We can choose some poles equals to « stable »zeros ! (artificial simplification)
- The unstable poles of the open loop can be chosen symmetrically. (Optimal control learning - guaranteed phase margin)

# Pole placement
## Your turn to play

### The system

$$H(p) = \frac{Y}{U}(p) = \frac{2}{p^2 - 0.18p + 2.4}$$

- We suppose that the output and the output derivative are measured.

### Specifications 1

- We want the system to behave like a first-order system, with a pole $p = -0.3$;

### Specifications 2

- We want the system to behave like a system with a double pole $p = -0.3$;

# Pole placement
## Your turn to play

# Pole placement
## Matlab my friend

### A not so good use of my friend

- The system
  - `p=tf('p')`
  - `sys=2/(p*p-0.18*p+2.4)`
  - `ss(sys)` : Be careful, Matlab can kill my state !

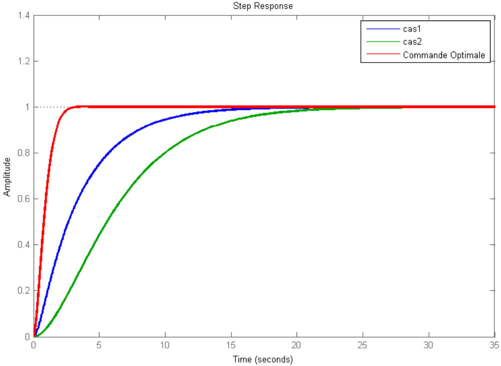
```
>> ss(sys)

ans =

  a =
             x1     x2
     x1    0.18   -1.2
     x2       2      0

  b =
          u1
     x1    1
     x2    0

  c =
          x1   x2
     y1    0    1
```

# Pole placement
## Matlab my friend

### The call to a friend

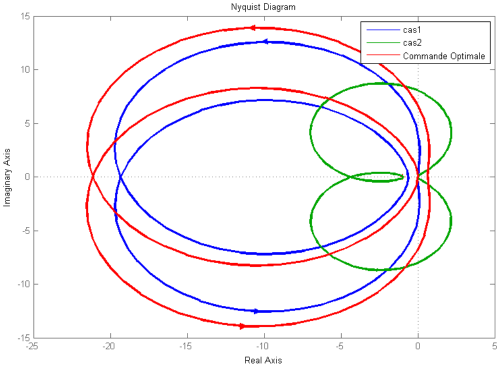
- The system
  - `A=[0 1 ; -2.4 0.18], B=[0 ; 1], C=[2 0], D=[0]`
  - `sys=ss(A,B,C,D)`
- Specifications
  - We have two poles to place : the first one : `p1=-0.3`
  - The second one, much more faster : `p2=-3`
  - Pole placement function : `K=place(A,B,[p1,p2])`
  - Static gain adjustment : `l=-1/(C*(A-B*K)`$^{-1}$`*B)`
- Closed-loop system definition :
  - `Abf=A-B*K,Bbf=B*l,Cbf=C,Dbf=D`
  - `sysbf=ss(Abf,Bbf,Cbf,Dbf)`

# Case 1 results
## Temporal behavior




- It looks nice
- But what about the margins ?

# Case 1 results
## Nyquist



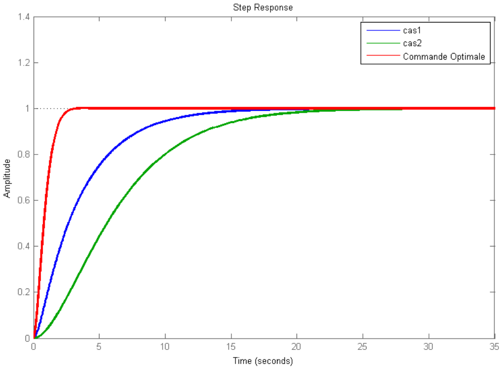

- The Good, The Bad and . . .

# Pole placement
## Case 2 - with Matlab

### The good use of Matlab

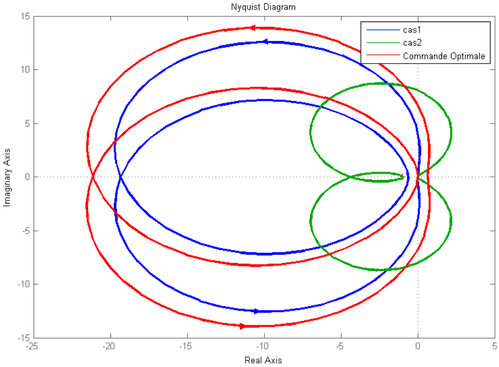
- The system
  - `A=[0 1 ; -2.4 0.18], B=[0 ; 1], C=[2 0], D=[0]`
  - `sys=ss(A,B,C,D)`
- Specifications 2 :
  - We have two poles to place : `p1=-0.3, p2=-0.3`
  - Pole placement function : `K2=acker(A,B,[p1,p1])`
  - The function `place` does not work if there are multiple poles.
  - Static gain adjustment : `l2=-1/(C*(A-B*K2)`$^{-1}$`*B)`
- Closed-loop system definition :
  - `Abf2=A-B*K2,Bbf2=B*l2,Cbf2=C,Dbf2=D`
  - `sysbf2=ss(Abf2,Bbf2,Cbf2,Dbf2)`

# Case 2 - The results
## Temporal behavior



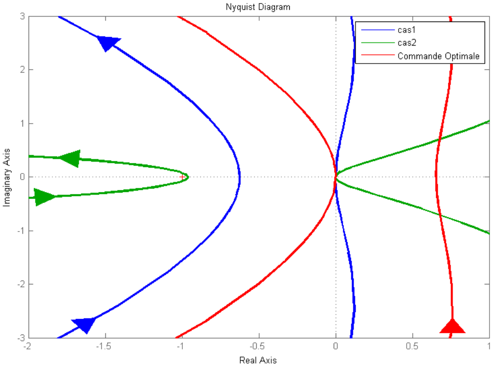

- It seems to work !
- What about the robustness ?

# Case 2 - The results
## Nyquist




- Est-on dans le vrai ?

# Case 2 - The results
## Nyquist




- Pole placement : no guarantee for the margins ! This need to be checked a posteriori !

# Outline

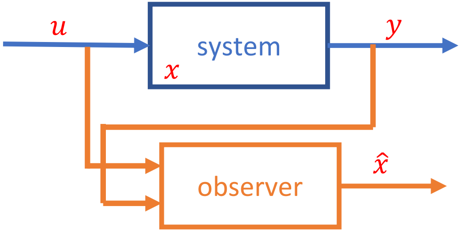
# Observer design
## Problem statement

### LTI SISO system

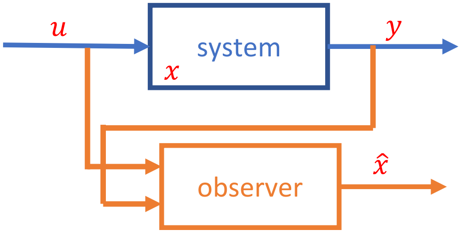
$$\begin{cases} \dot{x}(t) & = & Ax(t) + Bu(t) \\ y(t) & = & Cx(t) \end{cases}$$

- $A$ : size $n \times n$
- $B$ : size $n \times 1$
- $C$ : size $1 \times n$

- Can we have an estimation of $x$ ?



# Observer design
## Problem statement

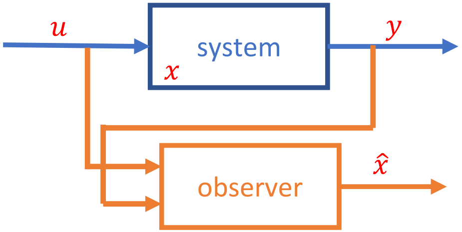



### The observer system

An observer is a system that has :

- Two inputs : the input $u$ of the system to observe AND its output $y$
- One output : the estimated state $\hat{x}$ of the system to observe

# Observer design
## One proposition




- We can use the output equation as a way to estimate the error

**The observer equation**

$$\dot{\hat{x}} = A\hat{x} + Bu + L(y - C\hat{x})$$

- How to tune $L$ so that the error of estimation asymptotically goes to 0 ?

# Observer design
## Tuning of $L$

- Let us define the error of estimation : $\varepsilon = \hat{x} - x$
- What is the dynamic equation of the error $\dot{\varepsilon}$ ?

$$\dot{\varepsilon} = (A - LC)\,\varepsilon$$

- We have to design $L$ so that this system is asymptotically stable !

# Observer design
## Tuning of $L$

$$\dot{\varepsilon} = (A - LC)\,\varepsilon$$

- We have to design $L$ so that this system is asymptotically stable !
- We can do the same as for state feedback : pole placement !

### The first question : is it possible ?

- Yes, if $(A, C)$ is an observable pair !

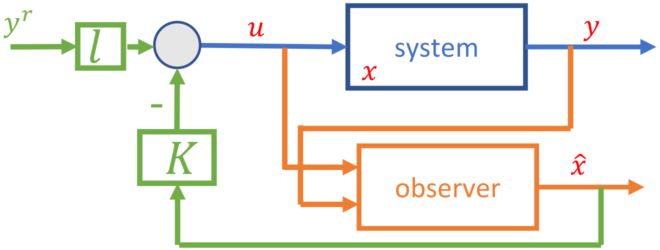
### The second question : how to choose the poles of the observer ?

- Its dynamic should be more faster than the dynamic of the closed-loop system (2 or 3 times)
- But not too fast (think of numerical implementation)

- As a remark : we can see the « dual »notion of controllability !

# Estimated state-feedback control

## The only way to implement a state feedback control !




### The real control law

$$u = -K\hat{x} + ly^r$$

- What are the consequences ?

# Estimated state-feedback control
## Separation principle

- Let us define an augmented state $X = \begin{pmatrix} x \\ \varepsilon \end{pmatrix}$
- What is the dynamical equation of this augmented state $\dot{X}$ ?

- The global closed-loop system has two dynamics : the one of the state-feedback AND the one of the observer

# Estimated state-feedback control

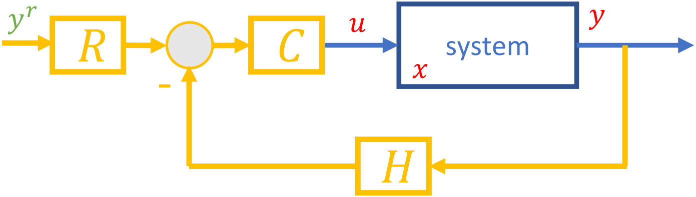
## Separation principle : what about the transfer function ?

- The global closed-loop system has two dynamics : the one of the state-feedback AND the one of the observer
- What about the controllability of the augmented system ?

- The transfer function between the reference and the output is the same as the one we get without the observer design !

# Estimated state-feedback control
## How to investigate on the stability margins ?

- How can we go back to a structure to analyse the margins ?




- What are $R$, $C$ and $H$ ? (Let us denote with $G$ the transfer of the system)

# Estimated state-feedback control

## How to investigate on the stability margins ?

# Outline

# Conclusion
## Expected skills

**Skills**

- Knowing the notion of state (See Modeling course)
- From TF to SS, and conversely
- Controllability and CCD
- Observability and OCD

# Conclusion
## Expected skills (2)

### Feedback by pole placement

- We can only act one the controllable part !
- Two step procedure : $K$ then $l$
- Two Matlab functions : `place` and `acker`

### Pole placement implementation

- Choice of the poles
- Observer design by pole placement
- Separation Principle
- Margins calculation (a posteriori)