
Control for Embedded Systems

(Contrôle/Commande pour les systèmes embarqués)

Nicolas MARCHAND

GIPSA-Lab (CNRS, Grenoble-INP)

Email: Nicolas.Marchand@grenoble-inp.fr

Web: <http://lag.ensieg.inpg.fr/marchand>

Master ISTRe

S4: Contrôle/Commande, 18h CM + 12h TP



Préambule

La théorie, c'est quand ça ne marche pas mais on sait pourquoi

La pratique, c'est quand ça marche mais on ne sait pas pourquoi

En général, on associe les deux : ça ne marche pas et on ne sait pas pourquoi



Contenu du cours

- **Introduction:**
 - Motivations et spécificité de l'embarqué
 - Quelques exemples de contrôle pour les systèmes embarqués
- **Modélisation:**
 - A base d'équation physique du système
 - Représentation d'état d'un système
 - Linéarisation d'un système non linéaire
- **Analyse:**
 - Transformations: Non unicité, SS vers TF et vice versa, Temps continu et temps discret
 - Stabilité and Robustesse.
 - Propriétés de contrôlabilité et d'observabilité
- **Commande:**
 - PID
 - Retour d'état (Placement de pôle et LQR)
 - Retour de sortie (Placement de pôle, LQG)
- **Implantations:**
 - Problématiques numériques (Echantillonnage, codage des mesures, calculs en nb de bits limités)
 - Incertitudes
 - Saturations des actionneurs



PART I. Introduction

Systemes embarques



Caractéristiques :

- Pas de définition claire
- Majorité des unités de calcul au monde (100 fois plus que tous les PC)
- Quelques points qui reviennent fréquemment :
 - Systèmes produits en grand nombres
 - Forte contrainte de coût
 - Souvent une forte contrainte énergétique
 - Systèmes électroniques/informatiques/(mécaniques)
- Le contrôle dans les systèmes embarqués :
 - Contrôle du système lui-même
 - Contrôle d'un système extérieur

Le rôle de la commande est, soit d'améliorer le comportement en réduisant les coûts (!), soit d'offrir de nouvelles fonctionnalités



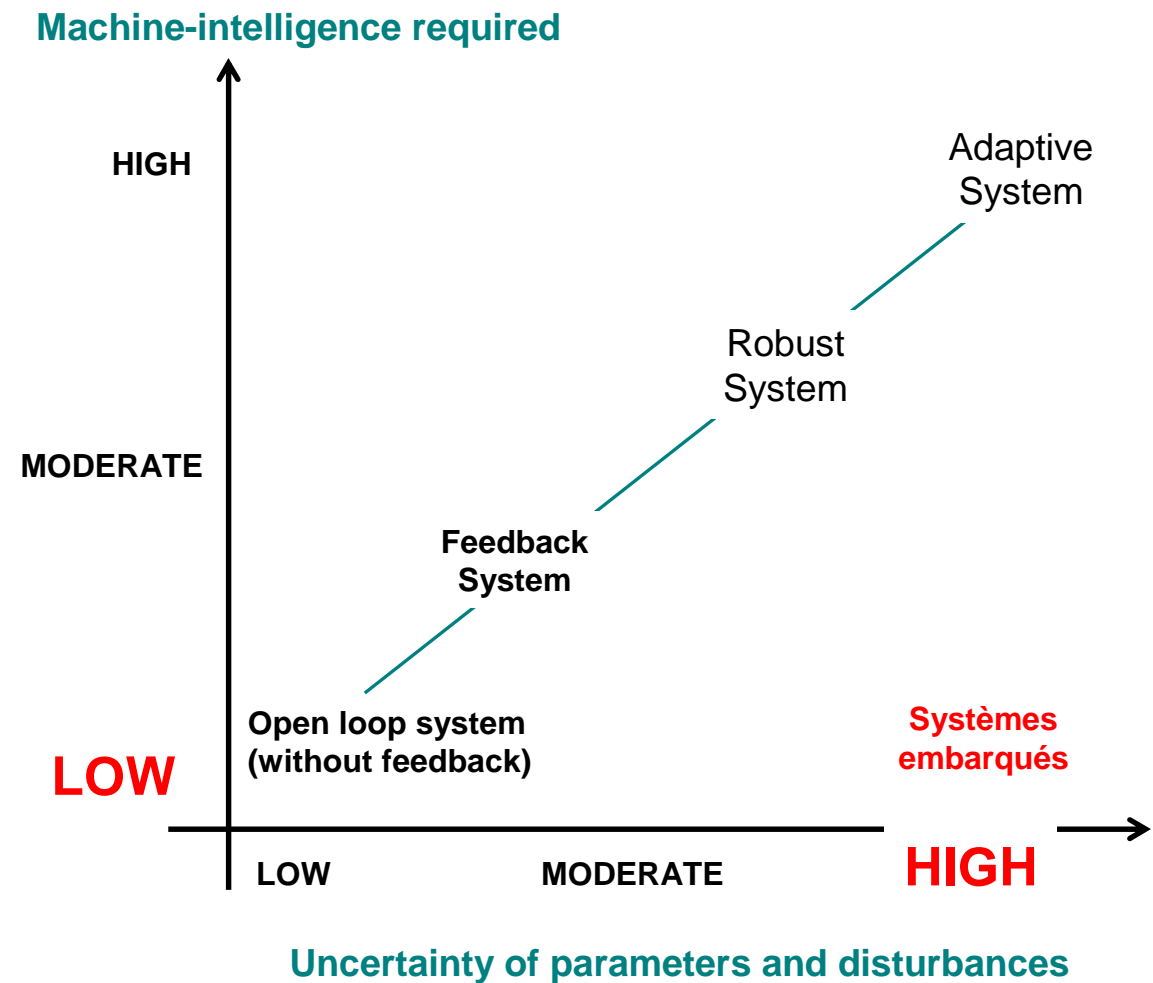
Quelques chiffres

- **Industrie automobile :**
 - 68 millions de voitures/an dans le monde \Rightarrow 1€/unité = 68 000k€ économisé
 - 5 millions/an par les constructeurs français \Rightarrow 1€/unité = 5 000k€ économisé
- **Téléphones portables :**
 - 1,15 milliards de nouveaux téléphone/an \Rightarrow 1ct/unité = 11 500k€ économisé
 - 24,3 millions/an en France \Rightarrow 1ct/unité = 243k€ économisé
 - 200 millions de Nokia 1100 vendus dans le monde \Rightarrow 1ct/unité = 2 000k€ économisé
- **Autres appareils électroniques :**
 - 110 millions d'iPod \Rightarrow 1ct/unité = 1 100k€ économisé
 - 115 millions de PS2 \Rightarrow 1ct/unité = 1 150k€ économisé
- **Robotique :**
 - 2,1 millions de robots dans le monde \Rightarrow 10€/unité = 21 000k€ économisé
- ...

Systemes embarqués

Les soucis :

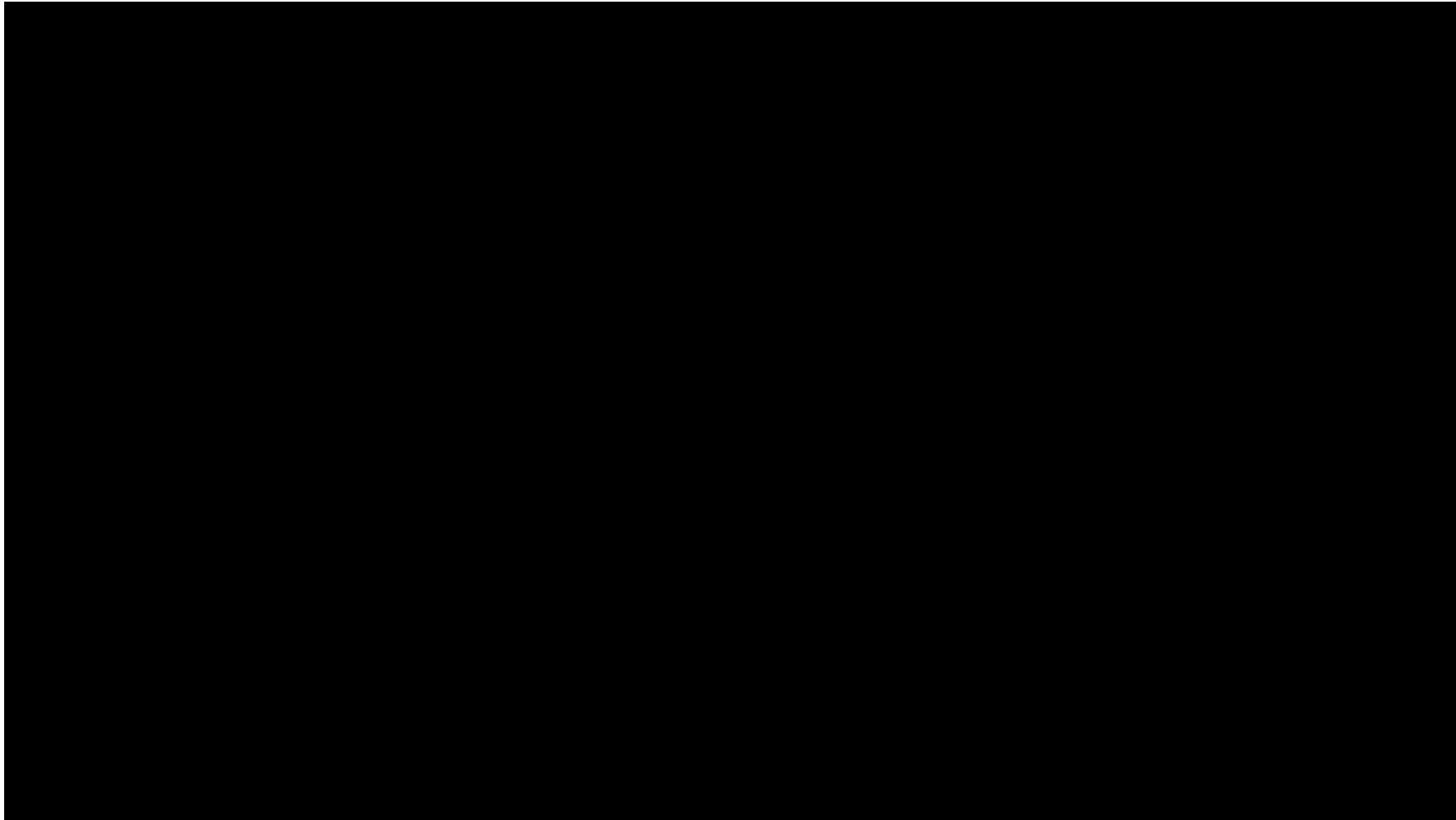
- Proviennent des contraintes de coût :
 - Éléments constitutifs à bas coût
 - Réduction des capacités de traitement de l'information
- Grande variabilité
 - ⇒ Incertitudes de paramètres
 - ⇒ Paramètres variants (batteries)
 - ⇒ Problème de robustesse
- Capacité de calcul limitée
 - ⇒ Nombres de bits limités, problèmes de quantification
 - ⇒ Retards
 - ⇒ Réseaux plus lents ou à capacité limitées





Quelques exemples

ESP : Amélioration des performances





Quelques exemples

Suspension active : Amélioration des performances





Quelques exemples

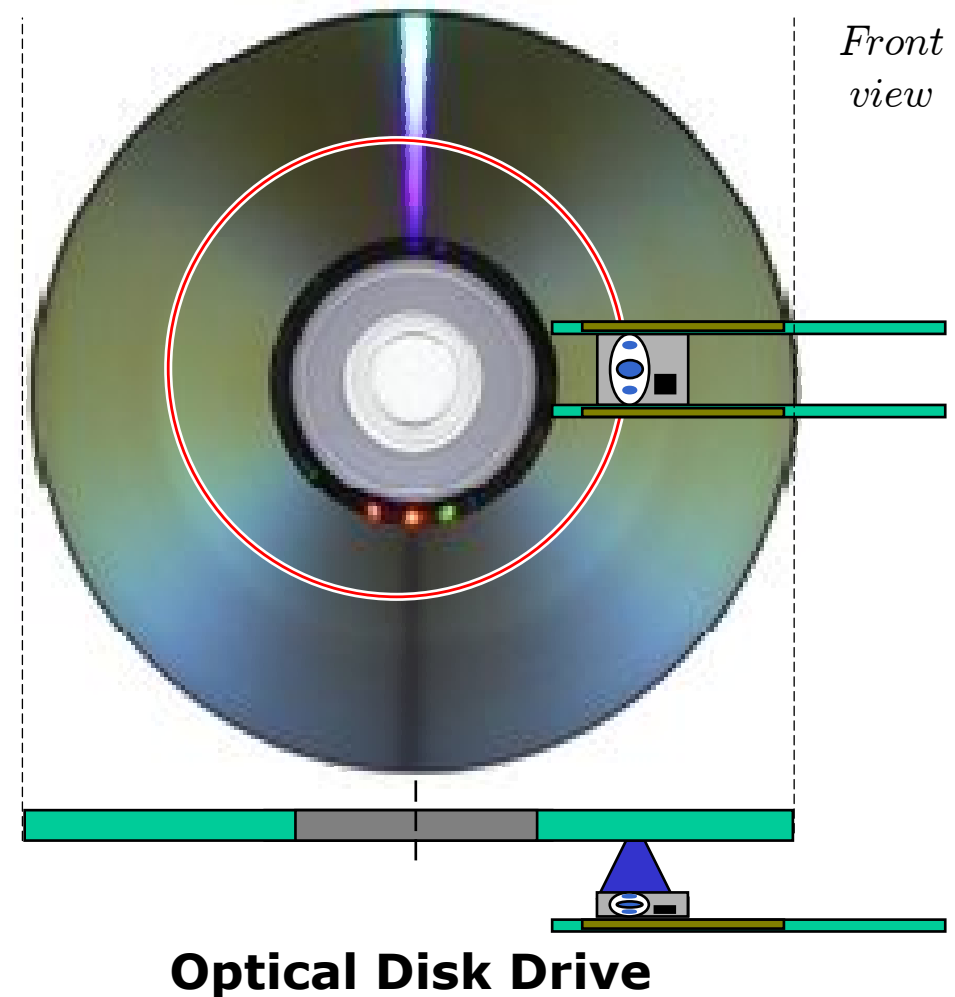
ESP : Amélioration des performances



Quelques exemples

Blue ray : Nécessaire au fonctionnement

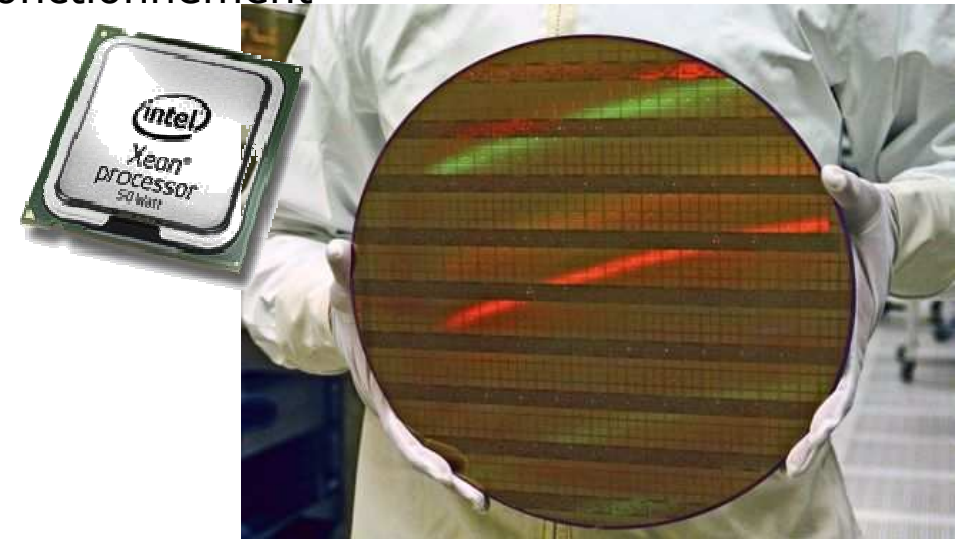
- **Système :**
 - Mvt radial et le focus
 - Mesures limitées à l'erreur de poursuite (capteurs optiques)
- **Perturbations:**
 - Perturbations périodiques
 - Principalement dues à la déformation du disque (**disc tilt, eccentricity,...**)
- **Contraintes fortes :**
 - BD ont les mêmes défauts que les DVD
 - BD nécessitent une erreur de poursuite plus faible pour être lisibles



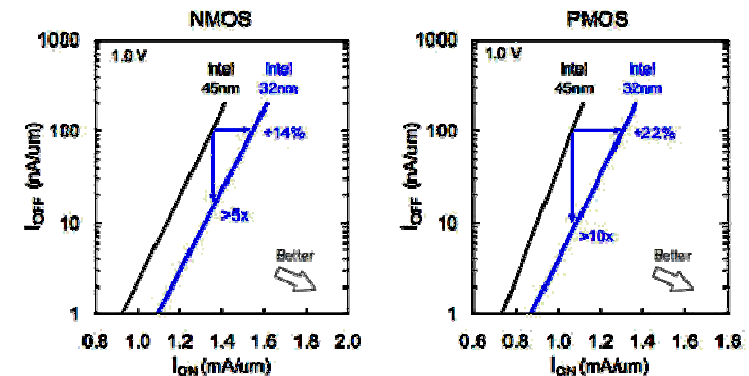
Quelques exemples

Puces submicroniques : Nécessaire au fonctionnement

- **Système :**
 - Très grande variabilité de performance entre les puces sur un wafer
- **Objectif :**
 - Rendre transparent cette inhomogénéité pour l'OS
- **Moyens :**
 - DVFS (dynamic voltage and frequency scaling)
 - Augmenter les tension et fréquence pour rendre transparent
- **Problèmes:**
 - Très grandes incertitudes de modèle
 - Très fortes contraintes d'intégration (parties en hard, parties en soft en local, parties en soft au niveau OS)



32nm Transistor Performance



32nm provides improved performance or reduced leakage
Highest drive current of all reported 32nm technologies

Quelques exemples

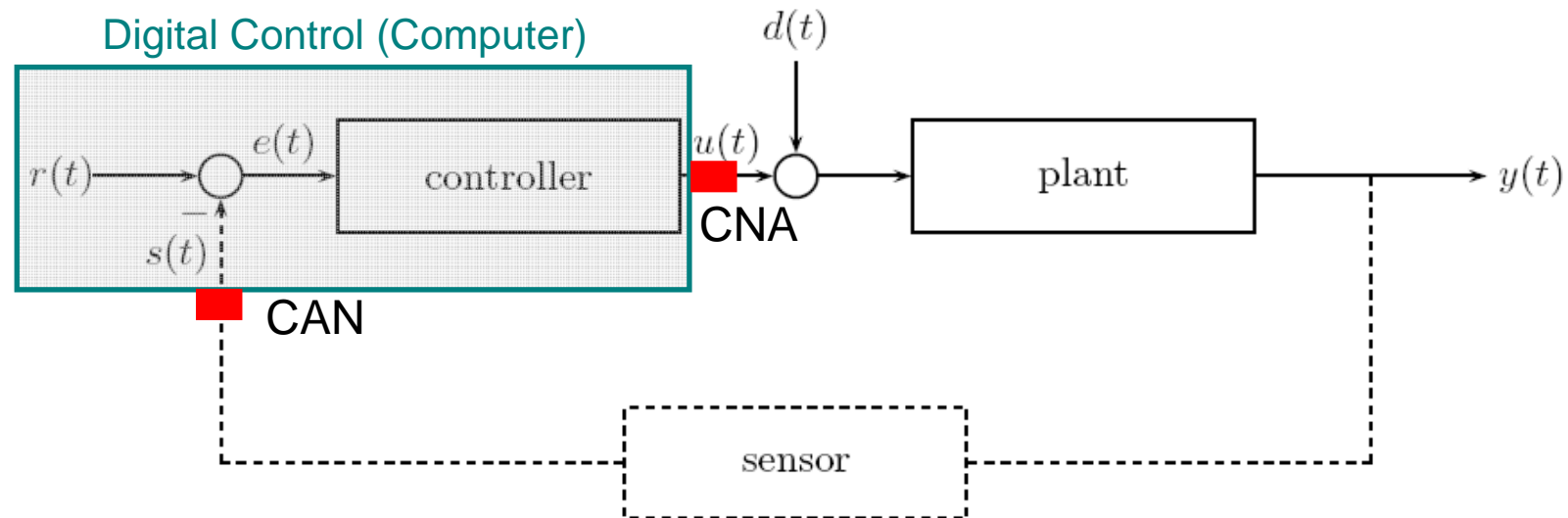
Produits de grande consommation : Gains par rapport à la concurrence

- **Système :**
 - Génération d'eau ou de vapeur
- **Objectif :**
 - Avoir une température et une pression donnée en sortie
- **Moyens :**
 - Commande de pression et de température
- **Contraintes:**
 - Matériel très bas coût
 - Matériel en très grandes séries (plusieurs centaines de milliers chaque jour)
 - 1ct/unité d'économisé \Rightarrow + de 350k€/an



La boucle de commande

- Schéma de base d'une boucle de commande (i.e.: boucle fermée)



$y(t)$: sortie,

$u(t)$: commande,

$r(t)$: référence

$e(t)$: erreur

$s(t)$: mesure

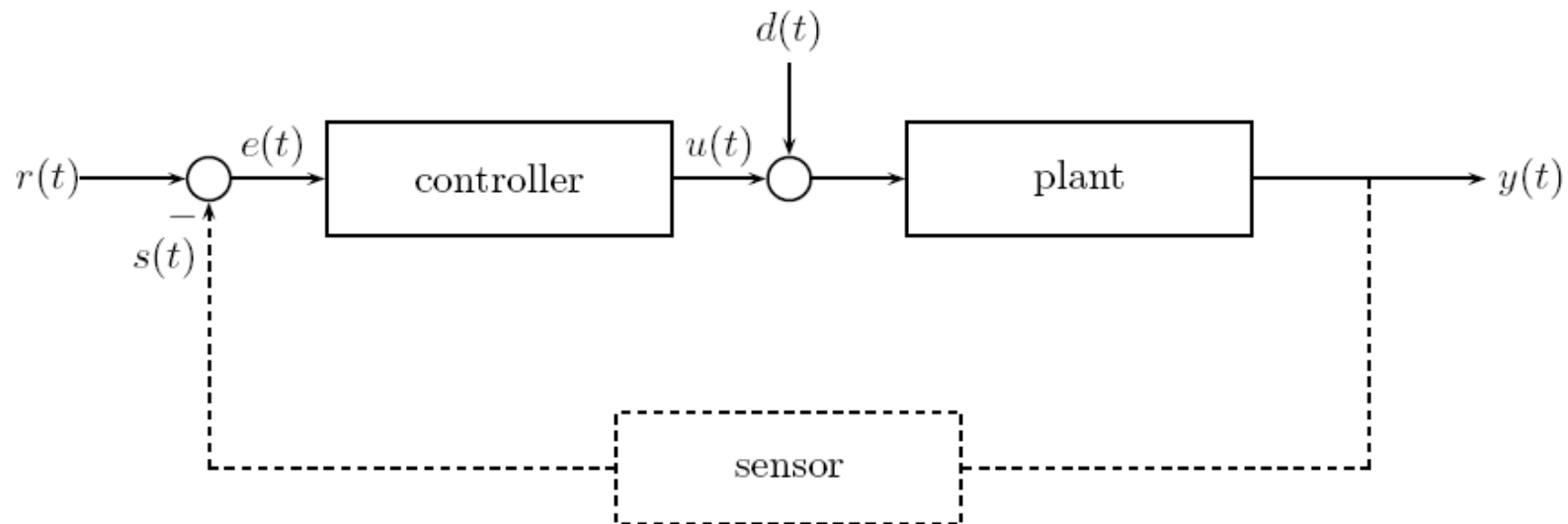
$d(t)$: perturbation

2 manière de suivre une trajectoire de référence:

- Open loop

- Closed loop (Feedback)

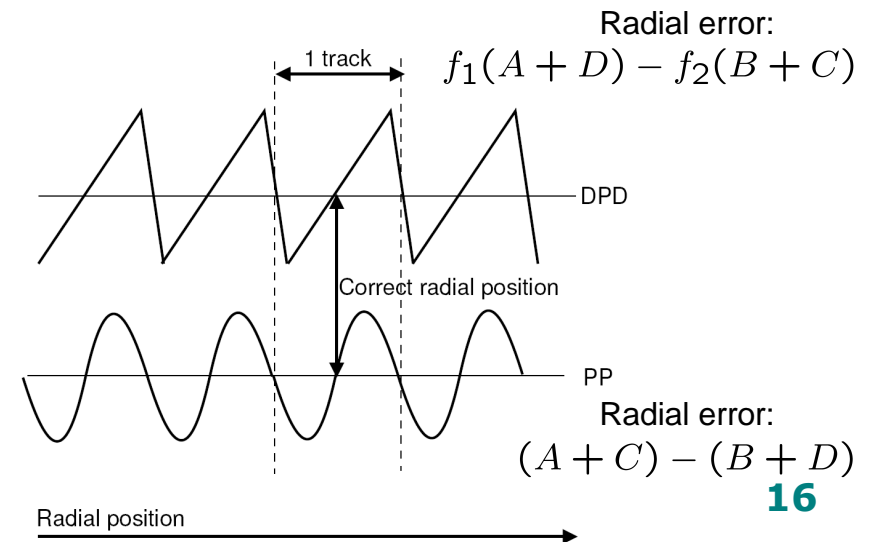
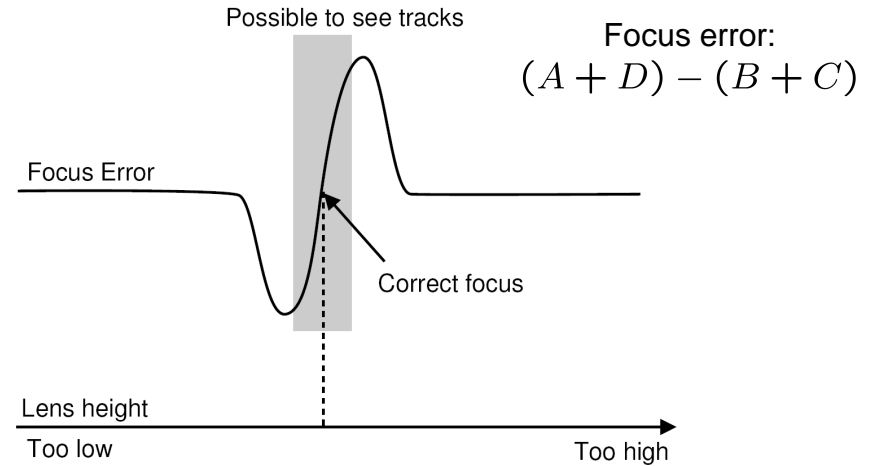
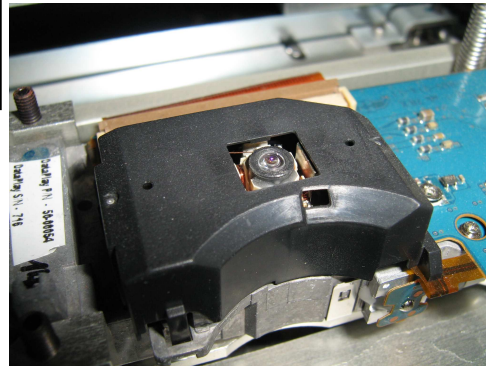
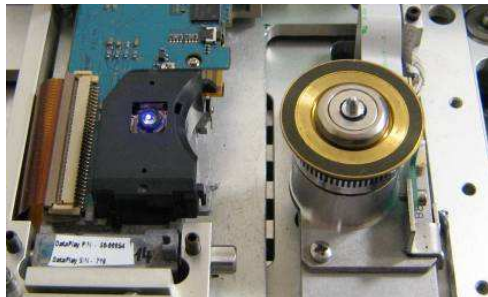
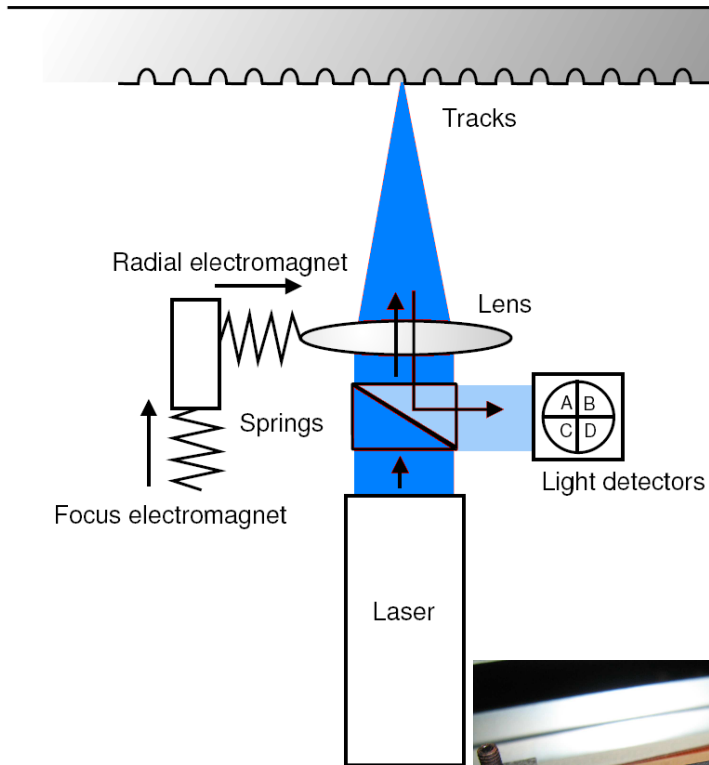
Un peu de vocabulaire



- **Stabilité** : Capacité à maintenir $r-y$ petit - *Asservissement*
- **Performances** : Capacité à réagir de manière adéquate (temps de réponses, dépassements, etc...) - *Régulation*
- **Robustesse** : Capacité à maintenir stabilité et performances en dépit d'incertitudes



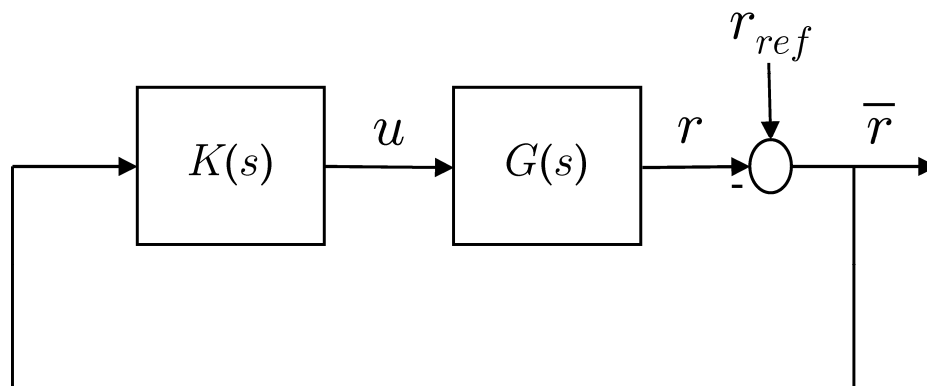
Example: The Blu-ray Disc servo control



Example: Performance specifications

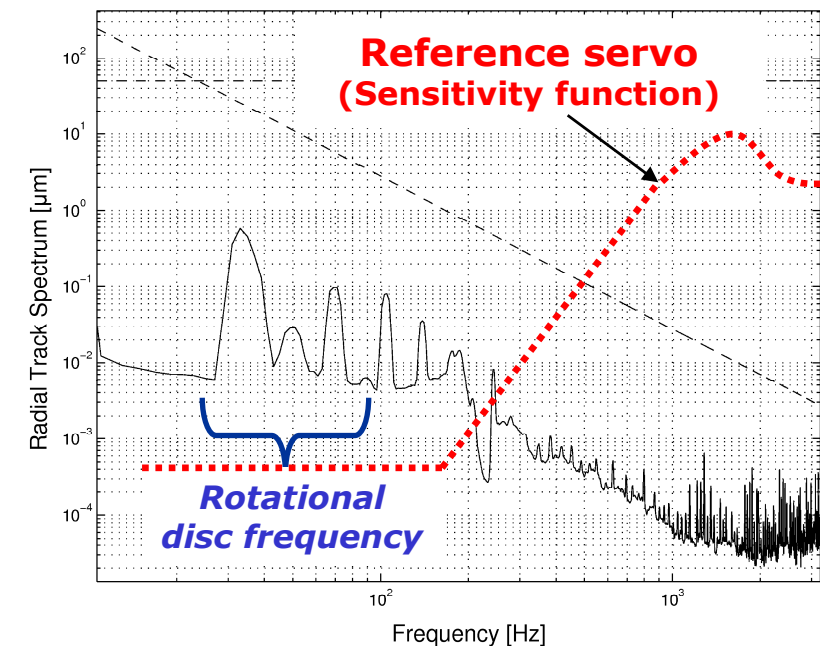
Time-domain specifications

Disturbance	Magnitude	Track.Error
Vertical deviation	$500\mu m$	$0.3\mu m$
Radial runout	$70\mu m$	$0.04\mu m$



- The control problem is to design a controller $K(s)$ such that the performance specifications are met.

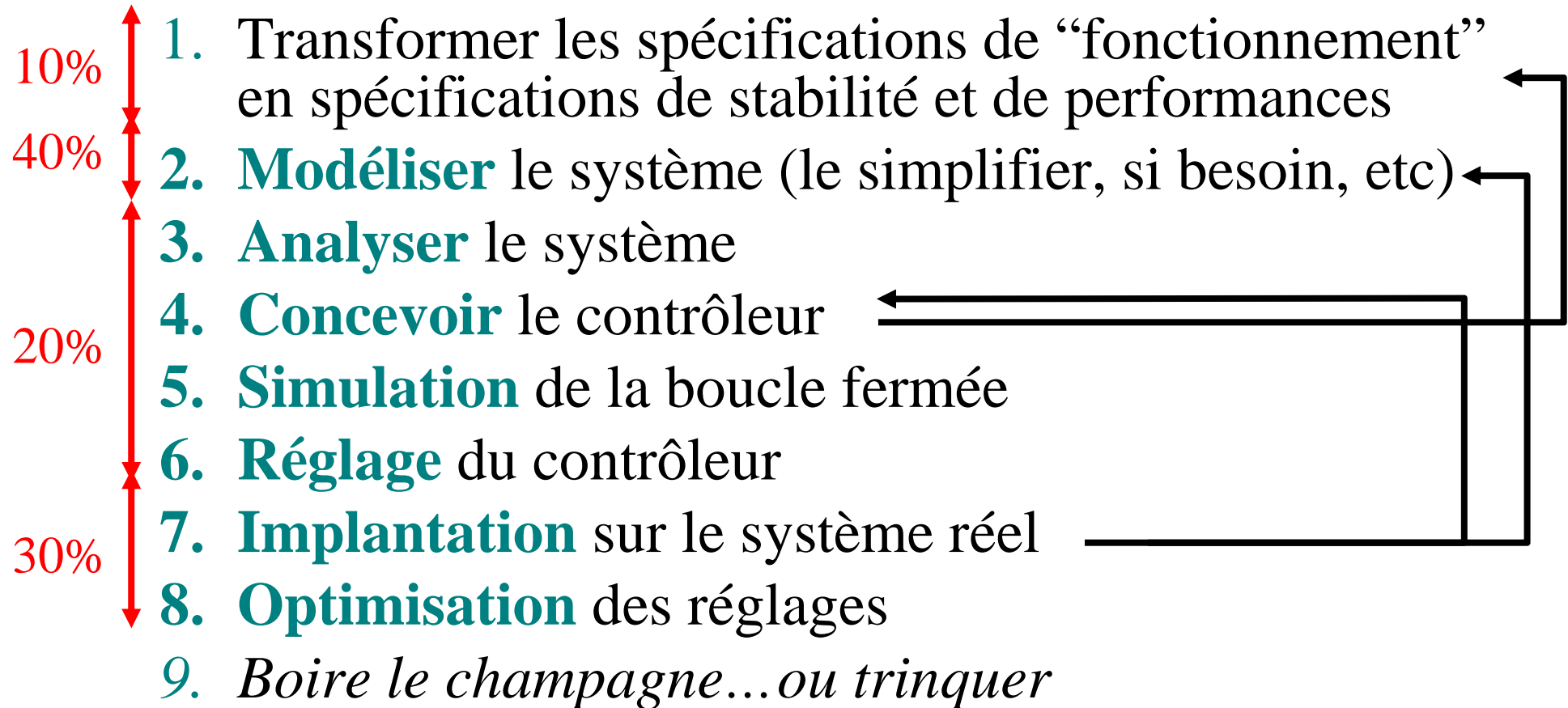
Frequency-domain specifications



- The sensitivity functions of the controlled system should stay below a given reference curve.



Travail de l'automaticien





PART II. Modélisation

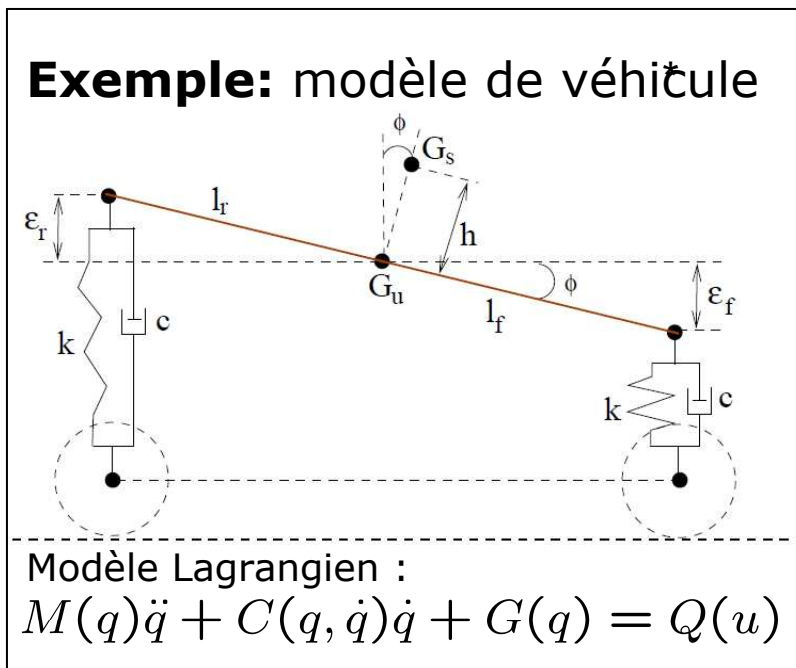
Modélisation "système"

- Modélisation du système
- Modélisation des actionneurs
- Modélisation des capteurs

Modélisation pour la simulation (Modelica, etc.)

VS.

Modélisation pour la commande (Cause-effets)



Monde macroscopique (Lois de Newton)

VS.

Monde micro-nano (interactions, couplages, frottements).

Théorie (Principes physiques)

VS.

Modélisation expérimentale (Identification, validation).



Modélisation - Identification

Modélisation à base d'identification expérimentale :

- Excitation du système par une SBPA (signal binaire pseudo aléatoire, PRBS en anglais) et on mesure la sortie
- Détermination de la fonction de transfert par identification (toolbox matlab)

➔ Modélisation à base de connaissances et identification des paramètres :

Représente le comportement du système en utilisant les équations différentielles ou/et algébriques basées sur la connaissance physique du modèle

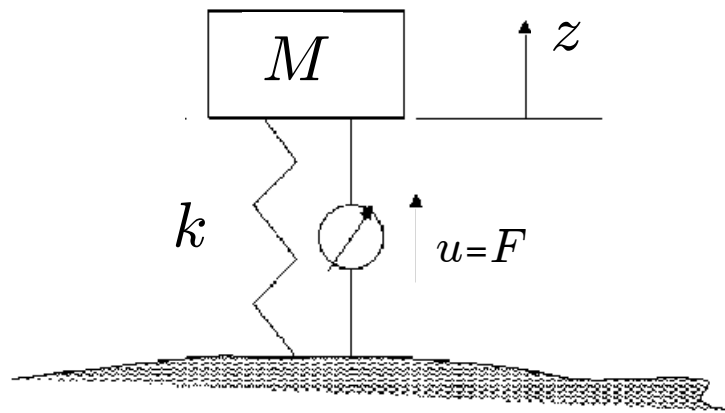
- Equations du système (souvent non linéaires)
- Fixe le point de fonctionnement
- Introduire les variables de déviations et linéarisation du modèle

Modélisation des systèmes linéaires

- Système masse-ressort (Equations de Newton)

$$M\ddot{z} = -kz + u$$

$$y = z$$



où z est la position relative de la masse, M la masse du système, k la raideur du ressort, u la force générée par la suspension active

En utilisant la transformée de Laplace :

$$\frac{dx}{dt} \equiv sX(s)$$

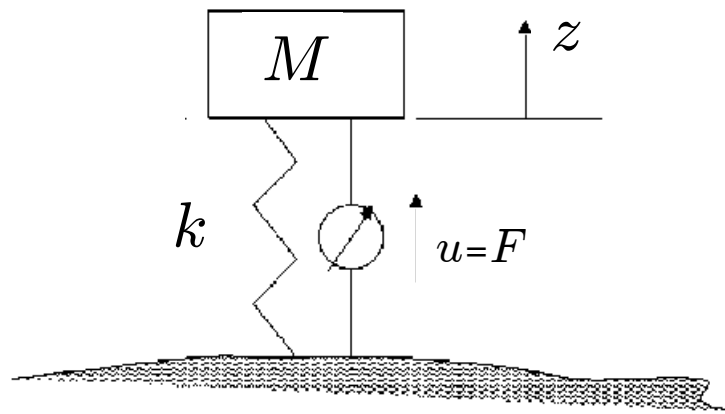
$$Ms^2Y(s) = -kY(s) + U(s)$$

La fonction de transfert est alors :

$$\frac{Y(s)}{U(s)} = \frac{(1/M)}{s^2 + (k/M)}$$

Modélisation des systèmes linéaires

- Système masse-ressort (Equations de Newton)



où z est la position relative de la masse, M la masse du système, k la raideur du ressort, u la force générée par la suspension active

$$M\ddot{z} = -kz + u$$

$$y = z$$

On définit les **variables d'état** : $x = \begin{bmatrix} z \\ \dot{z} \end{bmatrix}$

La **représentation d'état** est donnée par :

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

$$y = Cx + Du$$

Avec :

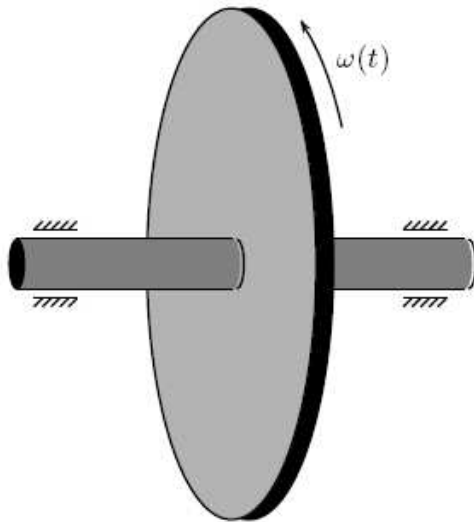
$$A = \begin{bmatrix} 0 & 1 \\ -\frac{k}{M} & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ \frac{1}{M} \end{bmatrix}, C = [1 \ 0], D = 0.$$

Le passage à la fonction de transfert est :

$$Y(s) = [C[sI - A]^{-1}B + D]U(s)$$

Modélisation des systèmes linéaires

- Système tournant (Lois de Newton)



where ω is the angular velocity, J is the inertial moment, d is the coefficient of the viscous dissipation (friction), and T is an external disturbance torque.

$$J\dot{\omega} = -d\omega + T$$

$$y = \omega$$

En utilisant la *transformée de Laplace* :

$$JsY(s) = -dY(s) + T(s)$$

La *fonction de transfert* est alors :

$$\frac{Y(s)}{T(s)} = \frac{(1/J)}{s + (d/J)}$$

Exercice: Trouver la représentation d'état

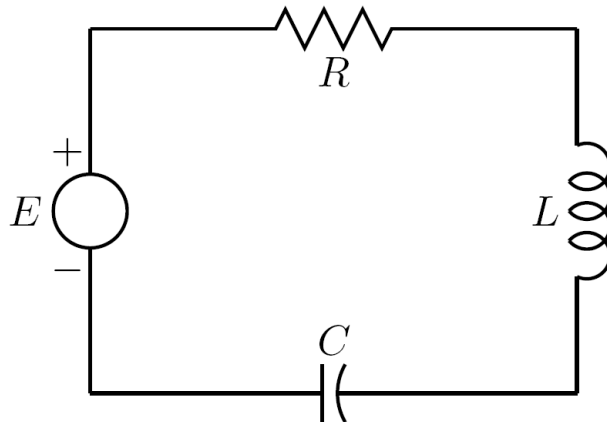
$$\dot{x} = Ax + Bu$$

$$y = Cx + Du$$

avec $A = -\frac{d}{J}$, $B = \frac{1}{J}$, $C = 1$, $D = 0$.

Modélisation des systèmes linéaires

- Circuit RLC (Lois de Kirchhoff)



où i est le courant électrique, R est la résistance, L l'inductance, C est la capacité, E est une source extérieure de tension

$$Ri + L \frac{di}{dt} + \frac{1}{C} \int i dt = E$$

$$y = i$$

En utilisant la *transformée de Laplace* :

$$RY(s) + LsY(s) + \frac{1}{sC}Y(s) = E(s)$$

Alors, la *fonction de transfert* sera :

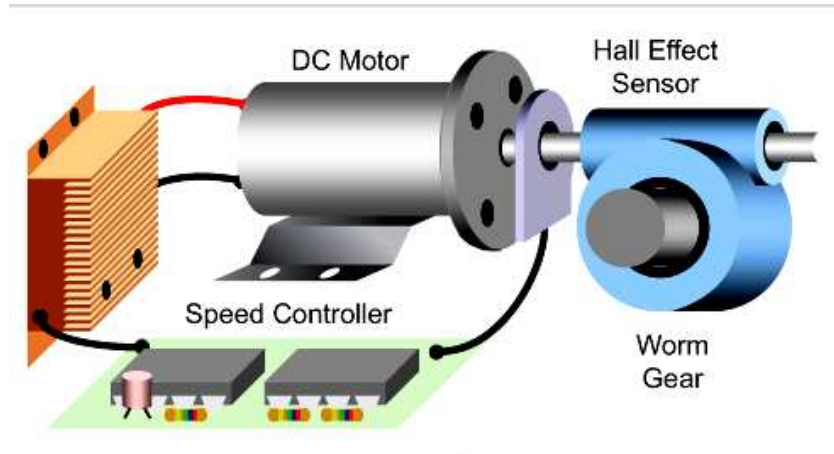
$$\frac{Y(s)}{E(s)} = \frac{(1/L)s}{s^2 + (R/L)s + (1/(LC))}$$

Exercice: Trouver la représentation d'état

$$A = \begin{bmatrix} -\frac{R}{L} & -\frac{1}{CL} \\ 1 & 0 \end{bmatrix}, B = \begin{bmatrix} \frac{1}{L} \\ 0 \end{bmatrix}, C = [1 \ 0], D = 0.$$

Modélisation des systèmes linéaires

- Système Electro-Mécanique



where θ is the angular position of the shaft, I is the electrical current, R is the resistance, L is the inductance, E is an external voltage source. K_e is a constant (EMF). J is the inertial moment, d is the coefficient of the viscous dissipation, and K_t is a constant ($K_t = K_e$).

Equations électriques :

$$RI + L\dot{I} + K_e\dot{\theta} = E$$

Equations mécaniques :

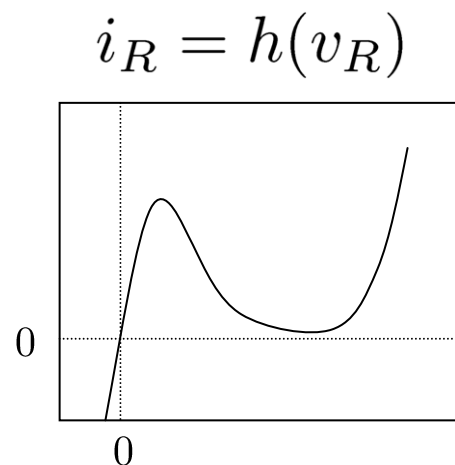
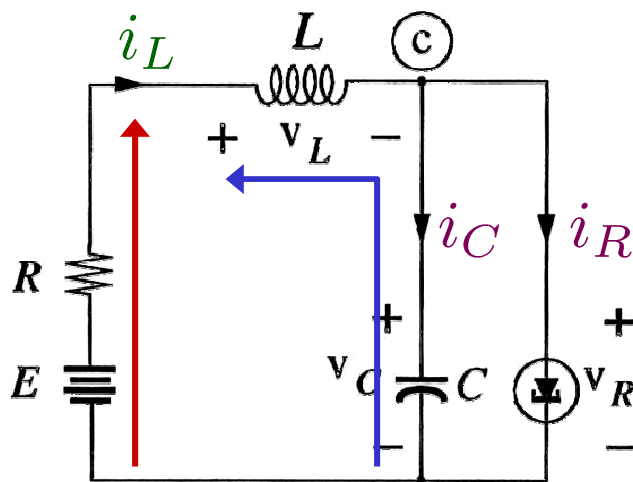
$$J\ddot{\theta} = -d\dot{\theta} + K_t I$$

Sortie :

$$y = \theta$$

Exercice: Trouver la fonction de transfert et la représentation d'état.

Modélisation des systèmes non linéaires



Equations électriques :

$$C \frac{dV_C}{dt} + i_R = i_L$$

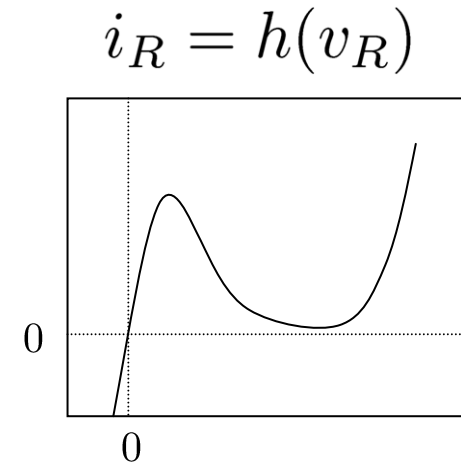
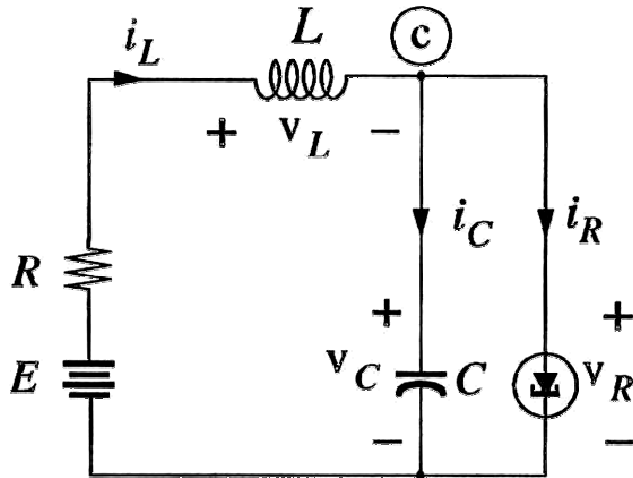
$$E - Ri_L = V_C + L \frac{di_L}{dt}$$

On cherche un modèle de la forme :

$$\begin{aligned} \dot{x}(t) &= f(x(t), u(t)) \\ y(t) &= h(x(t), u(t)) \end{aligned}$$

take: $x_1 = v_C, x_2 = i_L, u = E$

Modélisation des systèmes non linéaires



On obtient le modèle :

$$\dot{x}_1(t) = \frac{1}{C}(-h(x_1) + x_2)$$

$$\dot{x}_2(t) = \frac{1}{L}(-x_1 - Rx_2 + u)$$

with :

$$x_1 = v_C$$

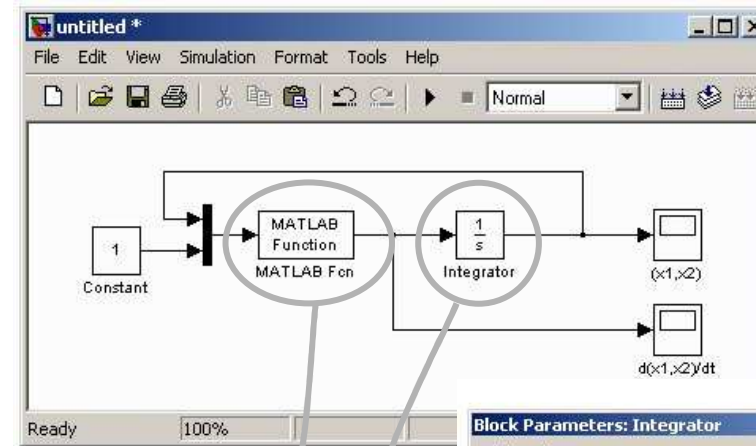
$$x_2 = i_L$$

$$u = E$$



Simulation des systèmes non linéaires

```
C:\MATLAB6p5\work\speed.m
File Edit View Text Debug Breakpoints Web Window Help
Stack Base
1 function dx=speed(x,u)
2 % Parameters
3 R=1500; % (Ohms)
4 L=5e-6; % (H)
5 C=2e-12; % (F)
6 iR=17.76*x(1)-103.79*x(1)^2+229.62*x(1)^3-226.31*x(1)^4+83.72*x(1)^5;
7 dx=[ 1/C*(-iR+x(2));
8     1/L*(-x(1)-R*x(2)+u) ];
9
```



Block Parameters: MATLAB Fcn

MATLAB Fcn
Pass the input values to a MATLAB function for evaluation. The function must return a single value having the dimensions specified by 'Output dimensions' and 'Collapse 2-D results to 1-D'.
Examples: sin, sin(u), foo(u(1), u(2))

Parameters:
MATLAB function:
speed(u(1:2),u(3))

Output dimensions:
2

Output signal type: auto

Collapse 2-D results to 1-D

OK Cancel Help Apply

Block Parameters: Integrator

Integrator
Continuous-time integration of the input signal.

Parameters:
External reset: none
Initial condition source: internal
Initial condition:
[1;3]

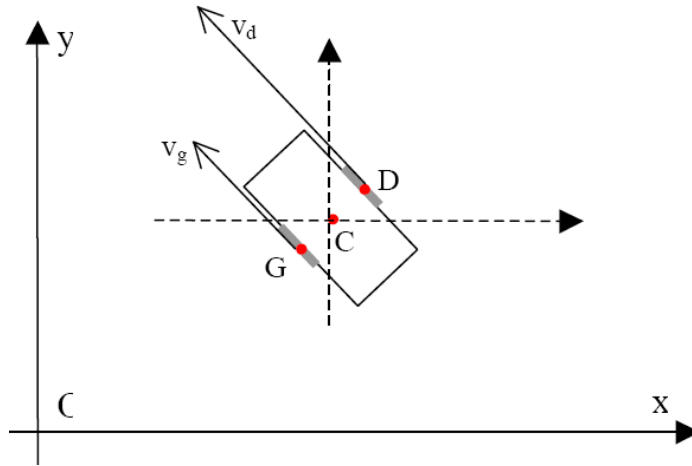
Limit output
Upper saturation limit: inf
Lower saturation limit: -inf

Show saturation port
 Show state port
Absolute tolerance: auto
 Enable zero crossing detection

OK Cancel Help Apply

Modélisation des systèmes non linéaires

- Robot Lego :



Modèle cinématique (vitesses prises comme entrées)

$$\begin{aligned}\dot{x} &= V \cos(\theta) \\ \dot{y} &= V \sin(\theta) \\ \dot{\theta} &= \omega\end{aligned}$$

Modèle dynamique (Couples moteurs comme entrées):

$$\begin{aligned}\dot{x} &= f_1(x, y, \theta, U_d, U_g) \\ \dot{y} &= f_2(x, y, \theta, U_d, U_g) \\ \dot{\theta} &= f_3(x, y, \theta, U_d, U_g)\end{aligned}$$

Linéarisation des systèmes non linéaires

On considère le **système non linéaire** suivant :

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

$$y = h(x, u)$$

La linéarisation en x_0 du système ci-dessus aboutit au **système linéaire** suivant :

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

$$\tilde{y} = C\tilde{x} + D\tilde{u}$$

Où :

$$A = \frac{\delta f}{\delta x}(x, u)|_{x_0, u_0}$$

$$B = \frac{\delta f}{\delta u}(x, u)|_{x_0, u_0}$$

Jacobien de f

$$C = \frac{\delta h}{\delta x}(x, u)|_{x_0, u_0}$$

$$D = \frac{\delta h}{\delta u}(x, u)|_{x_0, u_0}$$

Et :

$$0 = f(x_0, u_0)$$

$$y_0 = h(x_0, u_0)$$

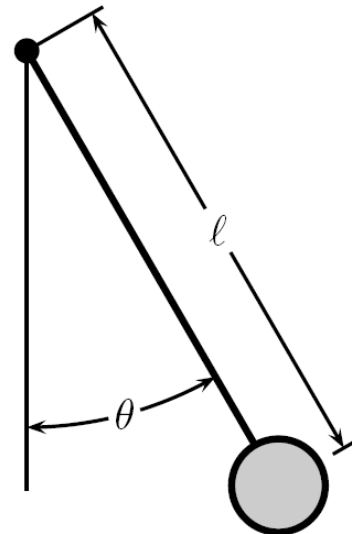
$$\tilde{x} = x - x_0$$

$$\tilde{u} = u - u_0$$

$$\tilde{y} = y - y_0$$

Exemple

- Pendule simple



où θ est l'angle du pendule, m la masse du pendule, l la longueur du pendule, g la force de gravitation et T est un couple externe.

$$ml^2\ddot{\theta} = -mgl \sin \theta + T$$

$$y = \theta$$

Prendre comme variable d'état : $x_1 = \theta$
 $x_2 = \dot{\theta}$

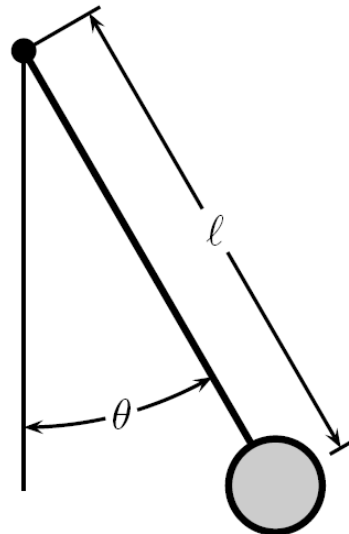
comme entrée : $u = T$

Les équations d'état sont alors :

$$\begin{aligned} \dot{x}_1 &= x_2 \\ \dot{x}_2 &= -\frac{g}{l} \sin(x_1) + \frac{1}{ml^2}u \\ y &= x_1 \end{aligned}$$

Exemple

- Pendule simple



où θ est l'angle du pendule, m la masse du pendule, l la longueur du pendule, g la force de gravitation et $u=T$ comme entrée de commande

La linéarisation du système non linéaire au point d'équilibre $\theta_o=0$ (c.à.d. u_o déduis de $0=f(0,0)$), donne:

pour
$$x = [\theta \ \dot{\theta}]'$$

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

$$y = Cx$$

avec

$$A = \begin{bmatrix} 0 & 1 \\ -\frac{g}{l} & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ \frac{1}{ml^2} \end{bmatrix}, C = [1 \ 0].$$

En utilisant la transformée de Laplace $\frac{dx}{dt} \equiv sX(s)$

La fonction de transfert sera :

$$Y(s) = [C[sI - A]^{-1}B + D]U(s)$$

Exercice:

Trouver le modèle linéaire pour $\theta_o = \pi$ (rad)



PART III. Analyse

Représentations temps continu et discret

Représentation d'état (temps continu)

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

Fonction de transfert (temps continu)

$$Y(s) = \frac{q_n s^n + \dots q_1 s + q_0}{s^n + \dots p_1 s + p_0} U(s)$$

Stable si

$$\Re \leq 0$$

Représentation d'état (temps discret)

$$\begin{aligned}x(k+1) &= A_d x(k) + B_d u(k) \\ y(k) &= C_d x(k) + D_d u(k)\end{aligned}$$

Fonction de transfert (temps discret)

$$Y(z) = \frac{b_n z^n + \dots b_1 z + b_0}{z^n + \dots a_1 z + a_0} U(z)$$

Stable si

$$|\Re| \leq 1$$

La stabilité du système est déterminée par les **Valeurs propres** de la matrice A ou de manière équivalente par les **POLES** (racines du dénominateur).

Les racines du numérateur sont appelés **ZEROS** du système, ils influencent la réponse du système mais pas la stabilité.



Passage temps continu - temps discret méthodes d'approximations

Forward difference (Euler's method)

$$\frac{dx(t)}{dt} \approx \frac{x(t+h) - x(t)}{h} = \frac{q-1}{h} x(t)$$

Backward difference

$$\frac{dx(t)}{dt} \approx \frac{x(t) - x(t-h)}{h} = \frac{1-q^{-1}}{h} x(t)$$

Trapezoidal method (Tustin, bilinear)

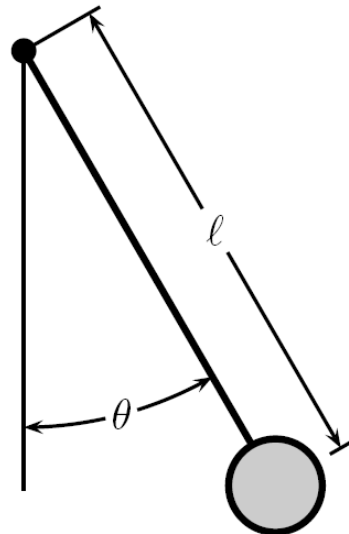
$$\frac{\dot{x}(t+h) + \dot{x}(t)}{2} \approx \frac{x(t+h) - x(t)}{h}$$

$$\longrightarrow \frac{dx(t)}{dt} = \frac{2}{h} \cdot \frac{q-1}{q+1} x(t)$$

méthodes d'approximations

Exemple

- Pendule simple



où θ est l'angle du pendule, m la masse du pendule, l la longueur du pendule, g la force de gravitation et $u=T$ comme entrée de commande

La représentation d'état pour $x = [\theta \ \dot{\theta}]'$

sera

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

$$y = Cx$$

En utilisant l'approximation d'Euler :

$$x(k+1) = A_d x(k) + B_d u(k)$$

$$y(k) = Cx(k)$$

avec :

$$A_d = [I + AT_e]; \quad B_d = [BT_e]; \quad C_d = C; \quad D_d = 0.$$

En utilisant la transformée en Z :

$$z^{-1}x(k) = x(k-1)$$

la fonction de transfert est :

$$Y(z) = [C_d [zI - A_d]^{-1} B_d + D_d] U(z)$$

Change of coordinates

Given a transfer function, there exists an infinity of state space representations (equivalent in terms of input-output behavior). Let

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

the transfer matrix being $G(s) = C(sI_n - A)^{-1}B + D$, and consider the change of variables $x = Tz$ (T being an invertible matrix). Replacing $x = Tz$ in the previous system gives:

$$\begin{aligned} T\dot{z}(t) &= ATz(t) + Bu(t) \\ y(t) &= CTz(t) + Du(t) \end{aligned}$$

Hence

$$\begin{aligned} \dot{z}(t) &= T^{-1}ATz(t) + T^{-1}Bu(t) \\ y(t) &= CTz(t) + Du(t) \end{aligned}$$



Change of coordinates

Defining $\tilde{A} = T^{-1}AT$, $\tilde{B} = T^{-1}B$ and $\tilde{C} = CT$, the transfer function of the previous system is:

$$\begin{aligned}\tilde{G}(s) &= \tilde{C}(sI_n - \tilde{A})^{-1}\tilde{B} + D \\ &= C T (sI_n - T^{-1}AT)^{-1} T^{-1} B + D\end{aligned}$$

Using $I_n = T^{-1}T$, we get

$$\tilde{G}(s) = C T T^{-1} (sI_n - A)^{-1} T T^{-1} B + D = G(s)$$

We have then *infinity* space state representations. The transfer function $G(s) = Y(s)/U(s)$ is exactly the same but the internal states are changed. However the the system properties remains unchanged (stability, controllability, observability...).



Solution of state space equations

The state $x(t)$, solution of $\dot{x}(t) = Ax(t)$, with initial condition $x(0) = x_0$ is given by

$$x(t) = e^{At}x(0)$$

This requires to compute e^{At} . There exist 3 methods to compute e^{At} :

1. Inverse Laplace transform of $(sI_n - A)^{-1}$:
2. Diagonalisation of A
3. Cayley-Hamilton method

Solution of state space equations

The state $x(t)$, solution of system (6), is given by

$$x(t) = \underbrace{e^{At}x(0)}_{\text{free response}} + \underbrace{\int_0^t e^{A(t-\tau)}Bu(\tau)d\tau}_{\text{forced response}}$$

Simulation of state space systems

In Matlab use: `>>sys=ss(A,B,C,D)`
`>>initial(sys,X0)`
`>>step(sys)`

Defining the system
obtaining free response
obtaining forced response



Summarizing

- Experimental and/or theoretical modeling are more suitable for control.
- System dynamics are described by Ordinary Differential Equations (ODE).
- Most of dynamical systems are actually Non-linear system.
- Linearization around an equilibrium point is often necessary.
- Linear control theory allow us to analyze and control the system around the equilibrium point.
- Engineers are often confronted to Electro-mechanical systems.
- Analysis and control of dynamical systems will be explored next.



PART III. Commande



State feedback control

Take the dynamical system:

$$x(k + 1) = Ax(k) + Bu(k)$$

If the system is “*controllable*”, then we may make a **state feedback** as follows:

$$u(k) = -Kx(k) + gr(k)$$

yielding:

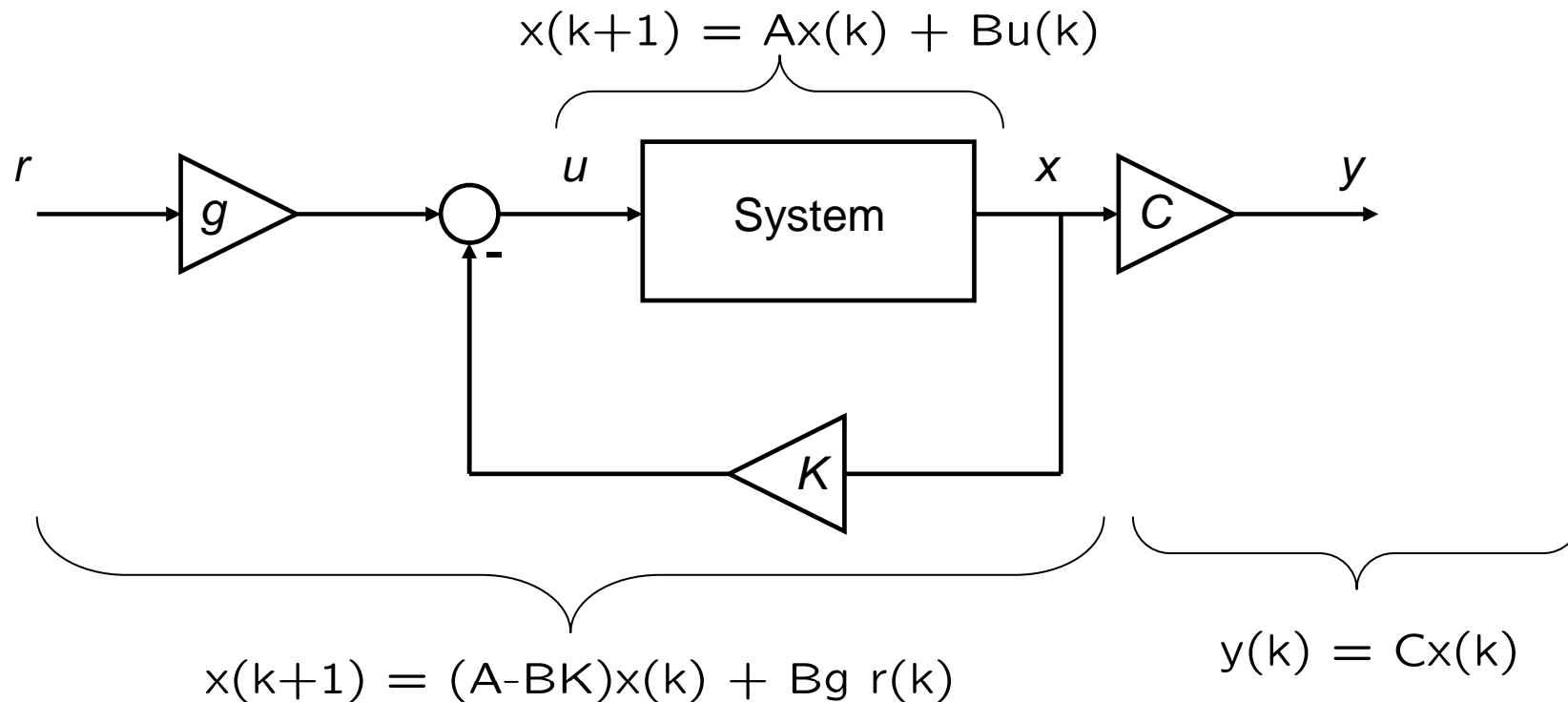
$$x(k + 1) = Ax(k) + B(-Kx(k)) + Bgr(k)$$

After factorization, the closed-loop system dynamics will be:

$$x(k + 1) = (A - BK)x(k) + Bgr(k)$$

The stability of the *closed-loop system* is given by the eigenvalues of matrix $(A - BK)$. For any initial condition $x(0)$, the control guarantees that the system state converges to zero.

State feedback control (*Multi-variable*)



The stability of the controlled system is given by the eigenvalues of matrix $(A-BK)$
 Methods to find a good matrix K : *Pole-placement, Optimal LQR, Loop shaping ...*
 The static gain g is computed to guarantee an **unitary gain** y/r .

Closed-loop Transfer Function

The closed loop system will be:

$$x(k+1) = \overbrace{(A - BK)}^{A_{bf}} x(k) + \overbrace{B}^{B_{bf}} gr(k)$$

$$y(k) = Cx(k)$$

with an equivalent transfer function :

$$Y(z) = C[I_n z - A_{bf}]^{-1} B_{bf} \cdot R(z) \leftarrow \text{Discrete time}$$

$$\star Y(s) = C[I_n s - A_{bf}]^{-1} B_{bf} \cdot R(s) \leftarrow \text{Continuous time}$$

In general, g could be computed as :

$$g = \left(C[I_n - A_{bf}]^{-1} B \right)^{-1} \leftarrow \begin{array}{l} \text{dc gain of the system} \\ \text{before compensation} \\ \text{(discrete time)} \end{array}$$

$$\star g = \left(C[-A_{bf}]^{-1} B \right)^{-1} \leftarrow \begin{array}{l} \text{dc gain of the system} \\ \text{before compensation} \\ \text{(continuous time)} \end{array}$$

to compensate the steady-state gain (i.e. *dcgain*) of the closed-loop system.



State observer

Take the dynamical system:

$$x(k+1) = Ax(k) + Bu(k)$$

If the system is “*observable*”, then we may make an **observer** as follows:

$$\hat{x}(k+1) = A\hat{x}(k) + Bu(k) + L(y - C\hat{x}(k))$$

Remark that inputs of the observer are u and y .

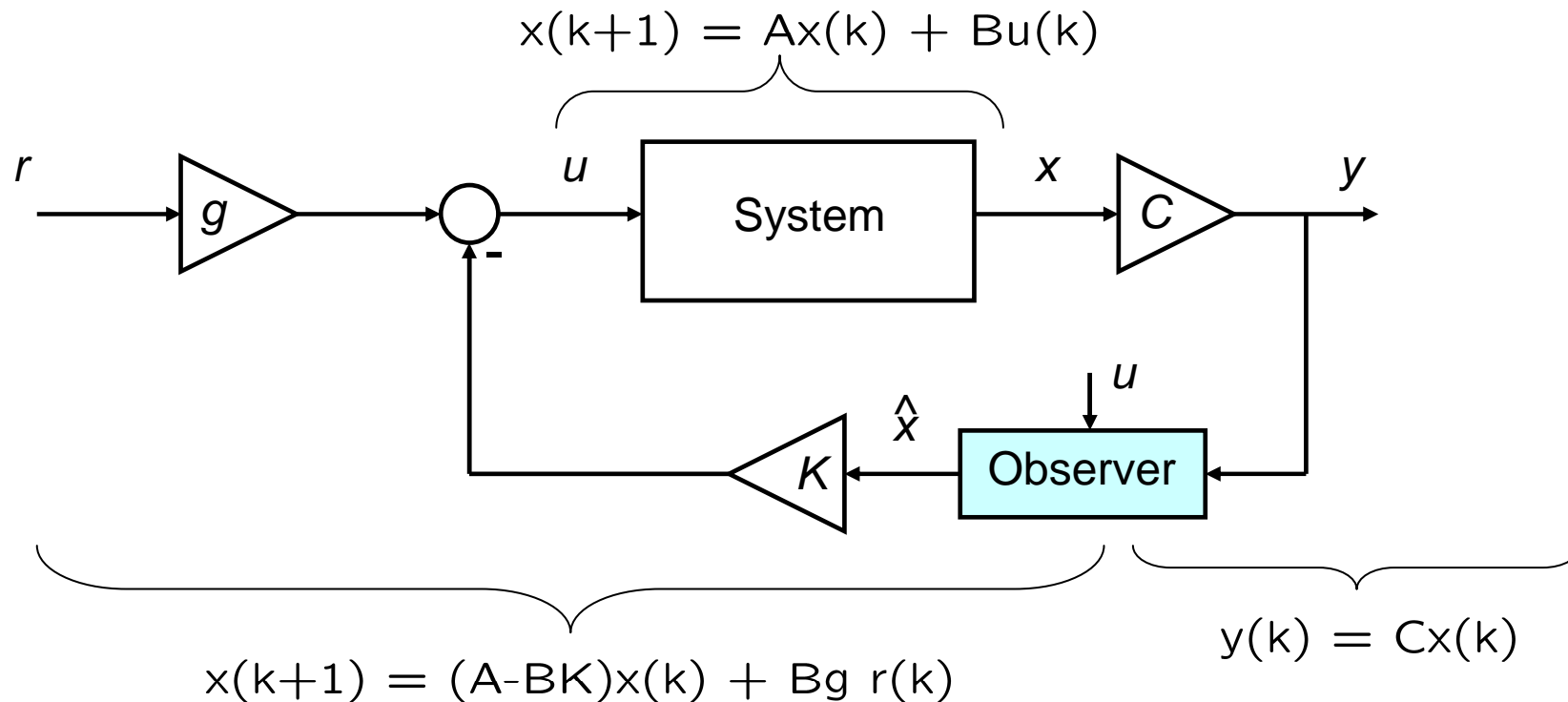
Defining the estimation error as: $e(k) := x(k) - \hat{x}(k)$

The estimation error dynamics is described by the following equation:

$$e(k+1) = (A - LC)e(k)$$

The stability of the *estimation error* is given by the eigenvalues of matrix $(A-LC)$. For any initial condition $e(0)$, the observer guarantees that the estimated state converges to the actual state.

State feedback control (*with Observer*)



The stability of the controlled system is given by the eigenvalues of matrix $(A-BK)$
 The observer guarantees that the *estimated state* converges to the *actual state*.
 Methods to find a good matrix L : *Pole-placement, Kalman filter, ...*

Observer-based state feedback control

Take the system model:

$$x(k+1) = Ax(k) + Bu(k)$$

If the system is “*controllable*”, then we may make an **observer-based state feedback** as follows:

$$u(k) = -K\hat{x}(k) + gr(k)$$

By definition the estimation error is: $e(k) := x(k) - \hat{x}(k)$

Then, we can write the state feedback as:

$$u(k) = -Kx(k) + Ke(k) + gr(k)$$

The closed loop system dynamics will be described as follows:

$$x(k+1) = (A - BK)x(k) + ~~BKe(k)~~ + Bgr(k)$$

The stability of the *closed-loop system* is given by the eigenvalues of matrix $(A - BK)$. For any initial condition $x(0)$, the control guarantees that the system state converges to zero. All that assuming that the observer guarantees $e(k)=0$



To discuss

- The equivalent system: $\dot{[x \ e]} = \bar{A} [x \ e] + \bar{B} u$
- Analysis about TF of the outputs w.r.t. input (disturbances), recall signals norms and system norms. H_2 and H_∞ . The only degree of freedom are the gains K , L and g . Optimisation problem formulation.