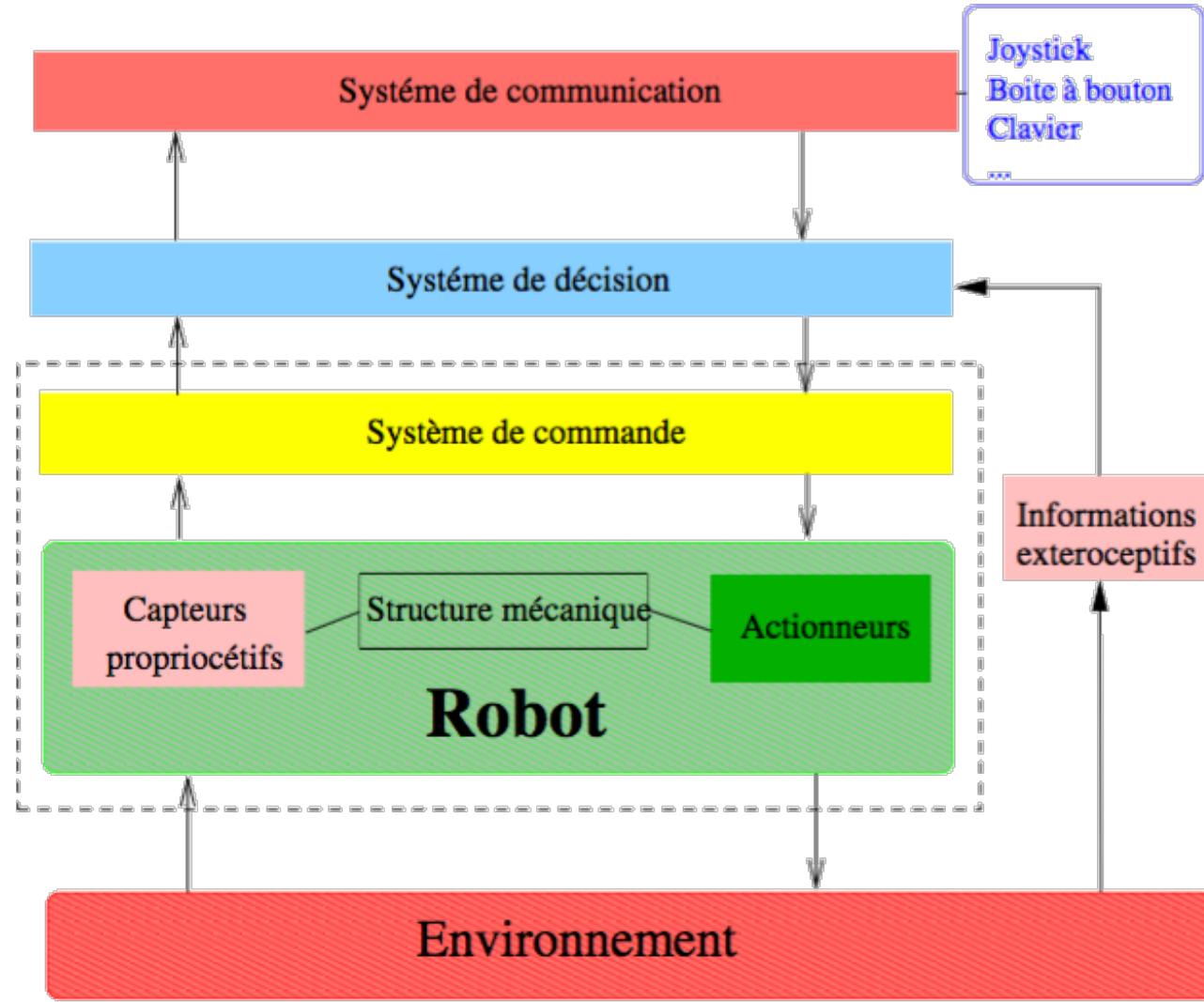


Filière Informatique
9ème Semestre – Option Robotique

UE I9ROBOT-A / Module AU321
Contrôle-Commande de Robots

Un robot, comment ça marche ?

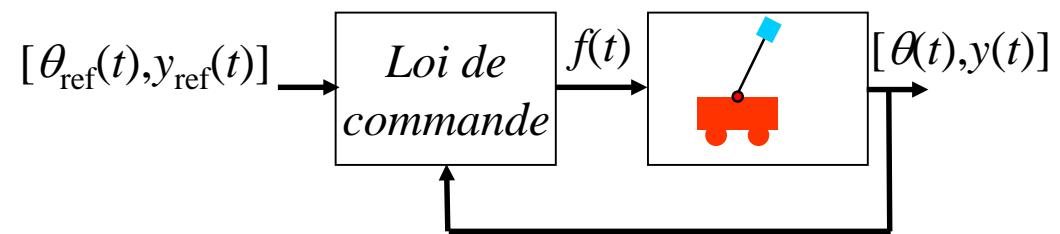
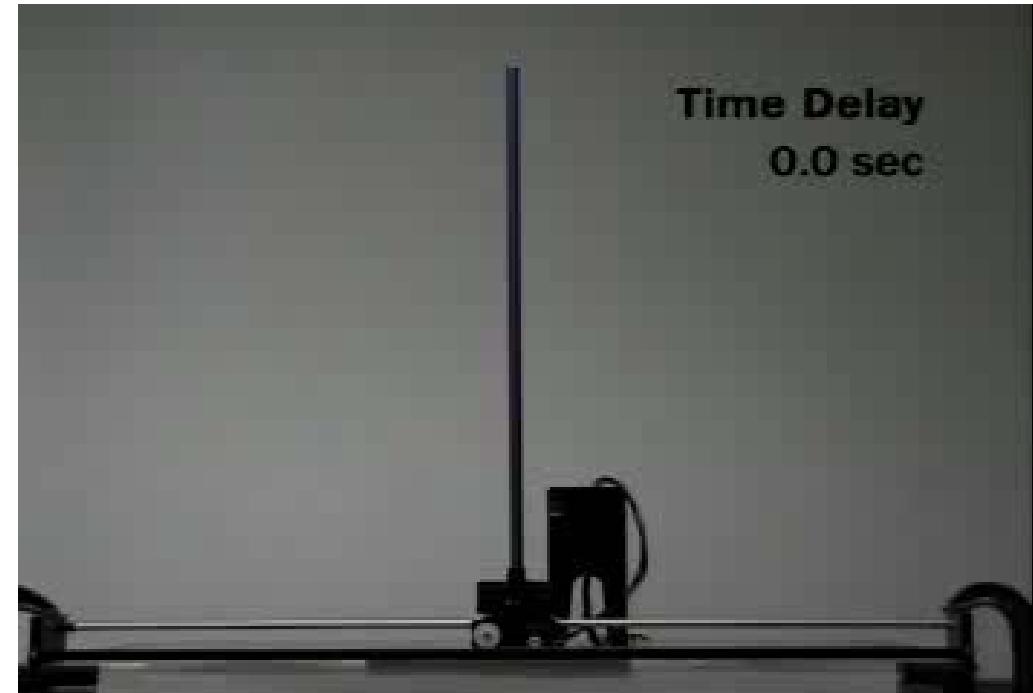
Atlas (2013) Boston Dynamics



Objectifs d'une loi de commande

Loi de commande : génération d'un *signal de commande* pour :

- Stabiliser l'état d'un système s'il est naturellement instable
- Asservir l'évolution de la (ou des) *sortie(s)* du système aux variations d'un *signal* (ou de signaux) *de référence*
- Générer des performances conformes à un cahier des charges
- Atténuer l'influence du milieu ambiant (*perturbations*)
- Assurer des performances indépendantes de l'état réel du système (*variations, incertitude*)



Du système à commander au système de commande en 3 étapes

- 1 **Modélisation** (*modèles de connaissance – géométrie, principes physiques, etc.*)
ou *Identification* (*modèle de comportement – mesures, données*)
du système à commander → modèles
- 2 **Synthèse** (ou design) d'une loi de commande en fonction
des modèles et du cahier des charges → loi de commande mathématique
- 3 **Implantation** physique de la loi de commande → sous forme mécanique,
hydraulique, pneumatique, électronique analogique ou/et numérique,
algorithmique dans un calculateur

Modèles géométriques

- Modèle Géométrique Direct (MGD)

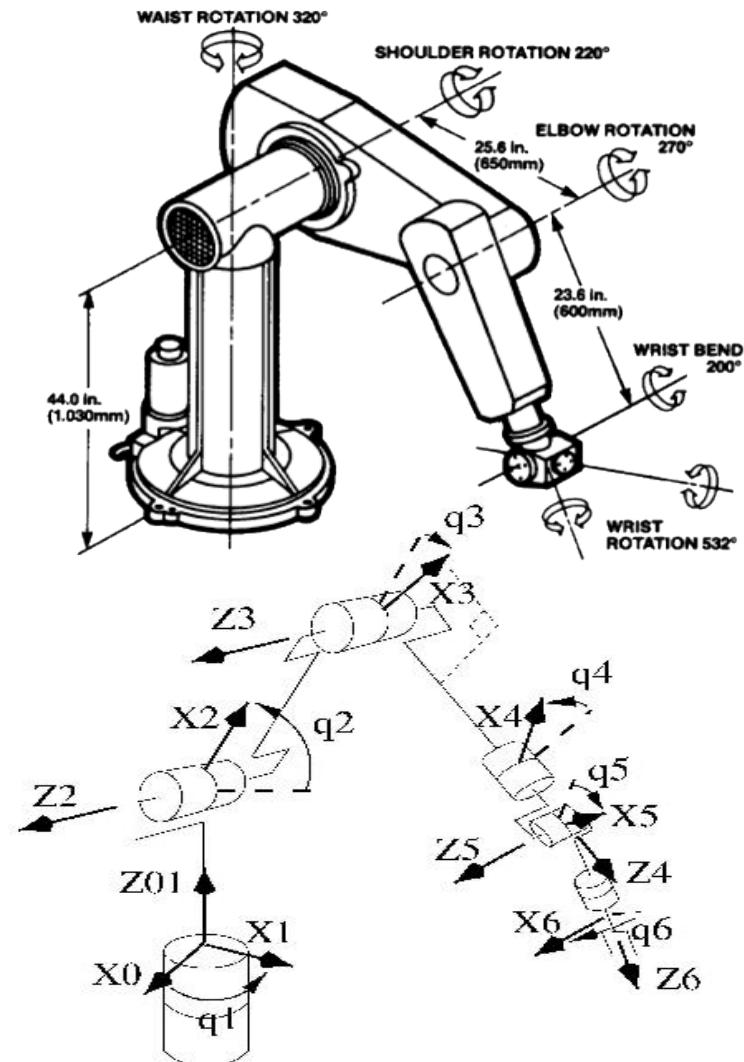
$$x = f(q)$$

x coordonnées opérationnelles
 q coordonnées articulaires



- Modèle Géométrique Inverse (MGI)

$$q = f^{-1}(x)$$



Robot manipulateur PUMA 560

Modèles Cinématiques (ou Différentiels)

- Modèle Cinématique Direct (MCD)

$$\frac{dx}{dt} = J \frac{dq}{dt} \quad \xrightarrow{\dot{q}} \text{MCD} \quad \xrightarrow{\dot{x}}$$

J : matrice Jacobienne du robot

avec $J_{ij} = \frac{\partial f_i(q)}{\partial q_j}$

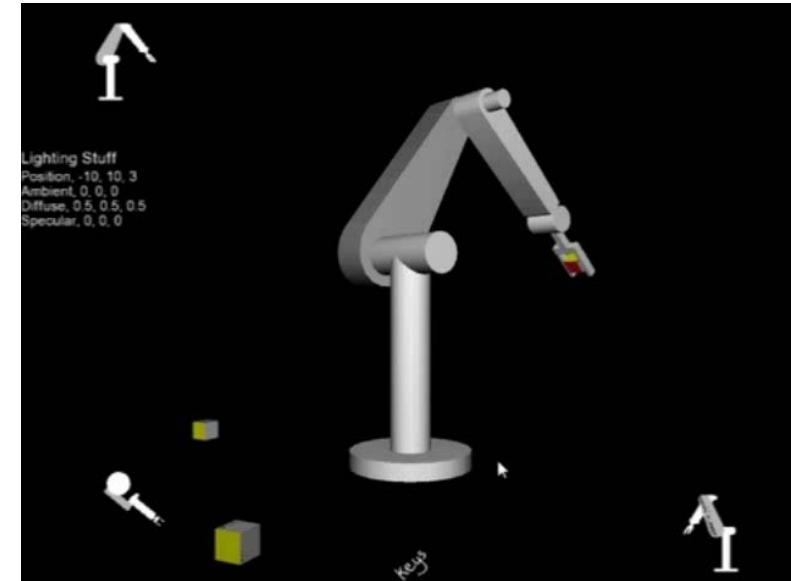
$$\Gamma^T dq = f_e^T dx \Leftrightarrow \Gamma = J^T f_e$$

Γ : force(/couple) produite aux actionneurs
 f_e : force exercée sur l'environnement

- Modèle Cinématique Direct du 2^{ème} ordre (MCD2)

$$\frac{d^2 x}{dt^2} = \frac{dJ}{dt} \frac{dq}{dt} + J \frac{d^2 q}{dt^2}$$

$$\xrightarrow{\begin{matrix} q(t) \\ J(t) \end{matrix}} \text{MCD2} \xrightarrow{\ddot{x}}$$



Modèles Cinématiques (suite)

- Modèle Cinématique Inverse (MCI)

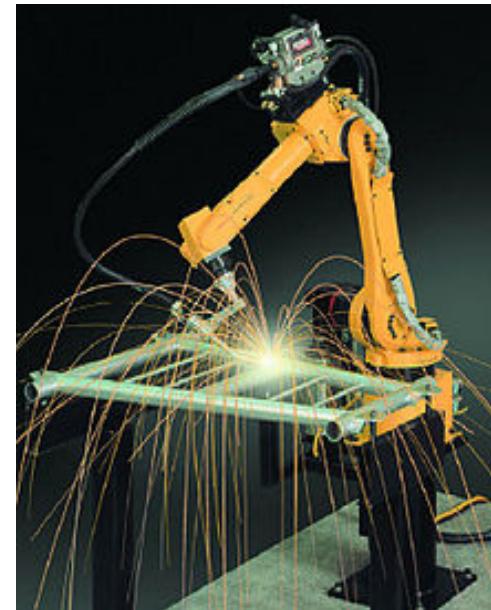
$$\frac{dq}{dt} = J^{-1} \frac{dx}{dt}$$



Si J non unique et inversible, J^{-1} est remplacé par une pseudo inverse de J

Robots redondants :

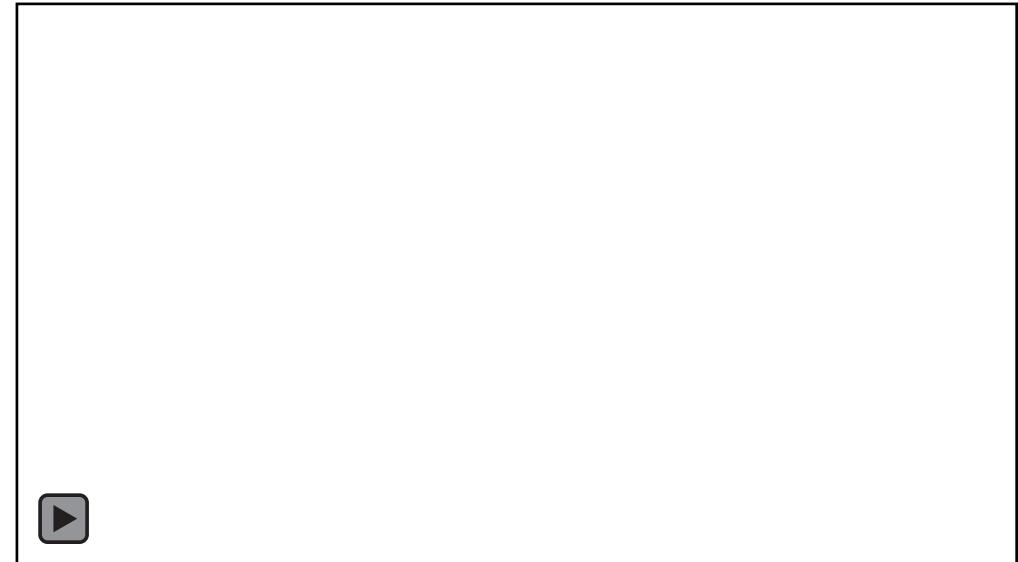
plusieurs q possible pour un x donné



Modèles Dynamiques

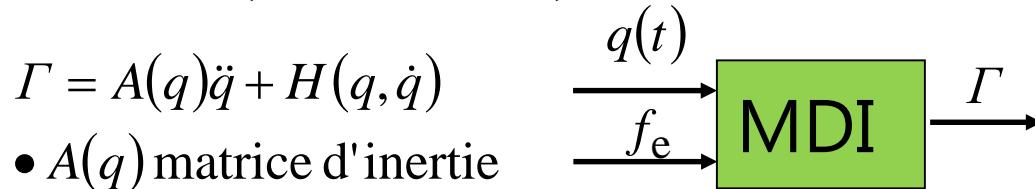
- Modèle Dynamique Direct (MDD)

$$\frac{d^2q}{dt^2} = g\left(q, \frac{dq}{dt}, \Gamma, f_e\right)$$



- Modèle Dynamique Inverse (MDI)

$$\Gamma = h\left(q, \frac{dq}{dt}, \frac{d^2q}{dt^2}, f_e\right)$$

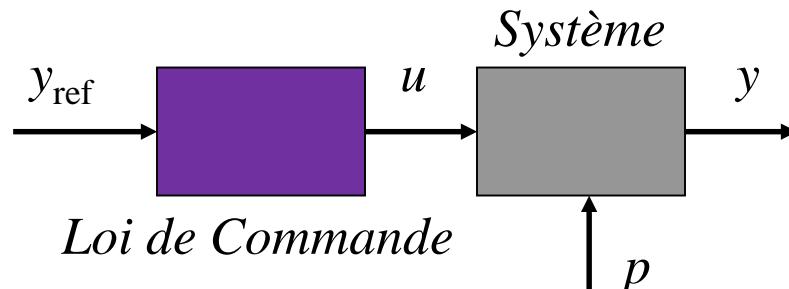


Agir sur un système ... quand on est confiant : la commande en chaîne directe

En fonction de son comportement (modèles) et de son environnement connu, appliquer à un système le signal de commande (l'entrée) nécessaire pour que sa sortie atteigne une valeur désirée.

On parle aussi de

- *open-loop control*
- commande par *précompensation*
- commande de type *feed-forward* (alimenter par l'avant)

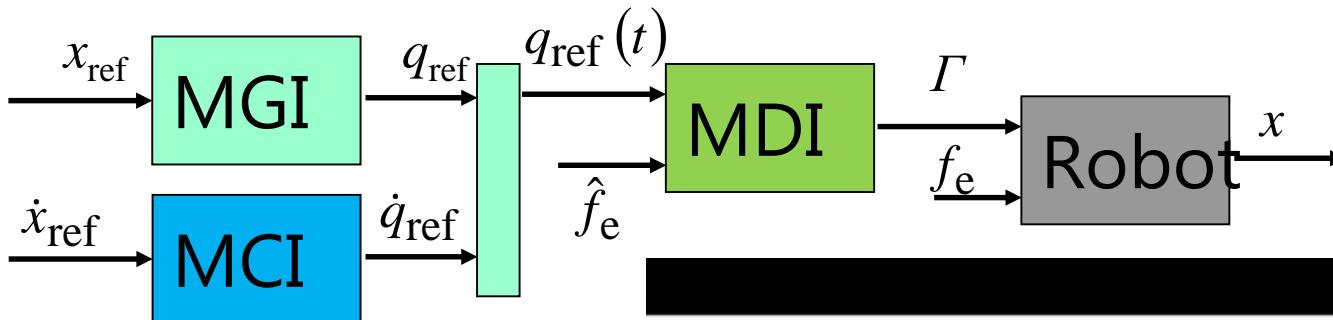


y_{ref}	: signal de référence
u	: signal de commande
p	: signal de perturbation
y	: signal de sortie

On considère ici que le système va réagir comme on l'imagine, c'est-à-dire obéir à un comportement nominal.

Commande de la marche d'un robot bipède

Modélisation aussi fine que possible et détermination directe des commandes à appliquer aux actionneurs (ou des consignes à appliquer aux servo-moteurs)



Ca peut fonctionner ...
mais au prix d'une
mécanique très précise,
lentement et sans garanti.

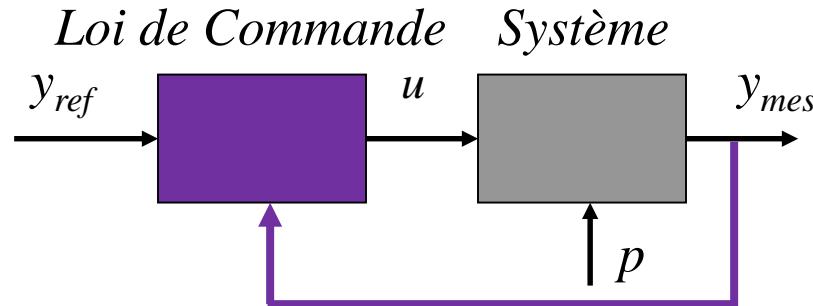


Drexel Autonomous Systems Lab

(Ré)Agir sur un système : la commande en boucle fermée (closed-loop)

Vérification de l'effet du signal de commande appliquée au système (comparaison des valeurs désirée et mesurée de la sortie à asservir) pour éventuellement en modifier la valeur.

On parle aussi de commande par *rétroaction* (*feedback* en anglais) ou de commande par contre-réaction.



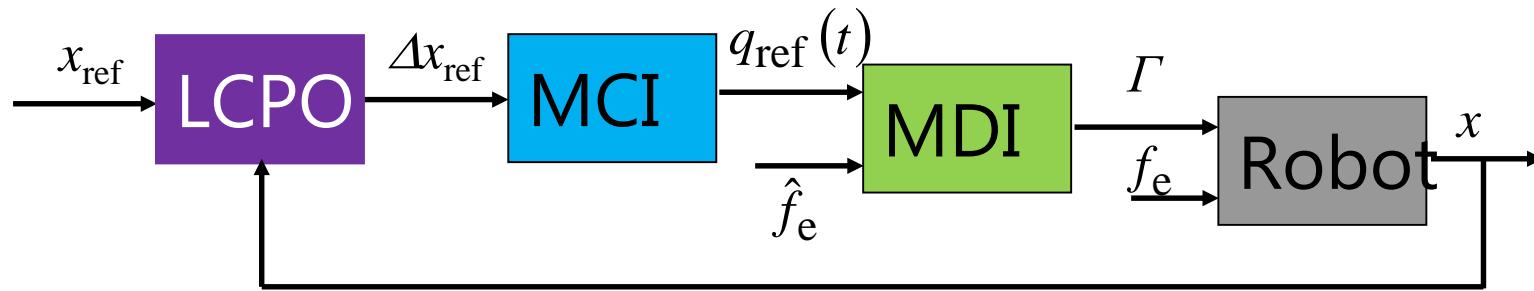
- y_{ref} : signal de référence
- u : signal de commande
- p : signal de perturbation
- y_{mes} : signal de sortie mesurée

- + Rejet de l'effet des perturbations
- + Atténuation de l'effet d'une connaissance imparfaite du système
- + Stabilise la sortie d'un système instable
- Nécessité des organes de mesure (capteurs)

Remarque : c'est par opposition que la commande en chaîne directe est souvent appelée *commande en boucle ouverte*.

Comparaison Closed-loop / Open-loop

Modélisation aussi fine que possible et détermination directe des commandes à appliquer aux actionneurs (ou des consignes à appliquer aux servo-moteurs)

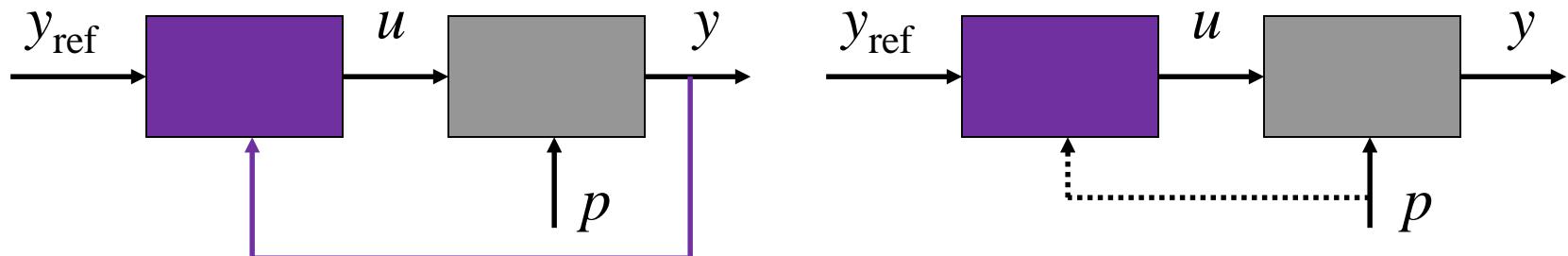


La contre-réaction
permet de « robustifier »
le fonctionnement en le
rendant moins sensible
aux incertitudes.

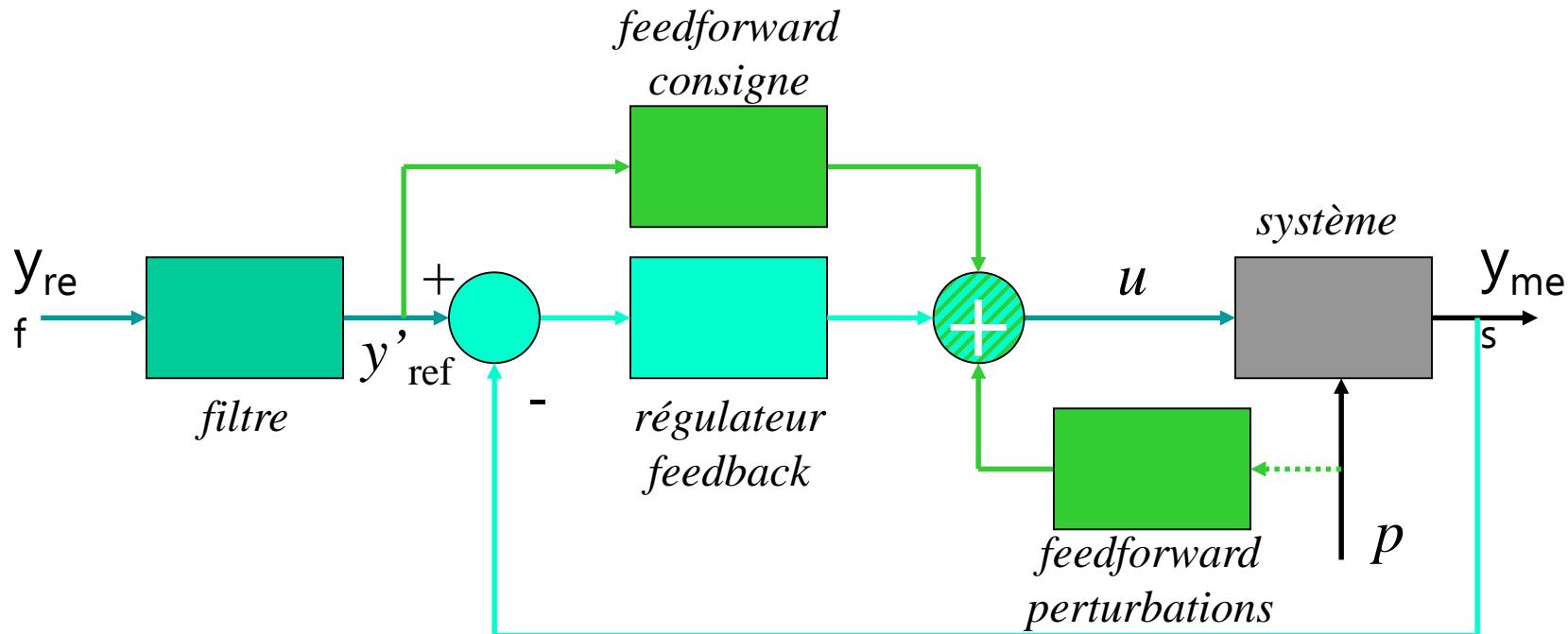


Comparaison FeedBack/FeedForward

FeedBack	FeedForward
<ul style="list-style-type: none"> + stabilise un système instable + rejette des perturbations mal connues + rejette l'effet des incertitudes - nécessite des capteurs - peut déstabiliser un système stable - nécessite une commande temps réel 	<ul style="list-style-type: none"> + nécessite moins de capteurs + commande calculable hors-ligne - le système et les perturbations doivent être parfaitement connues - ne peut pas stabiliser un système instable



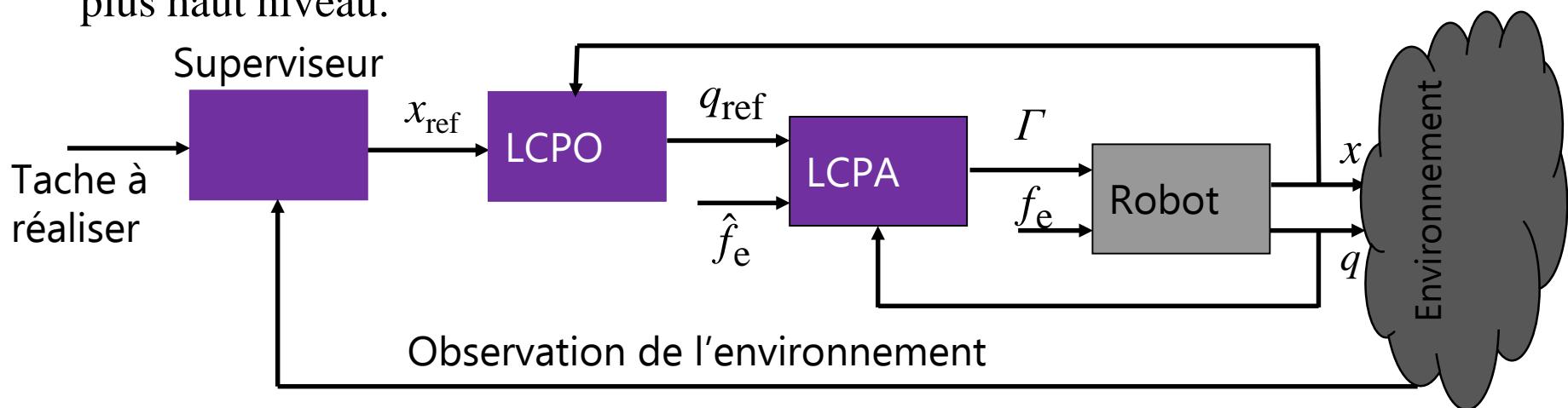
Commande par FeedBack & FeedForward



Structure de commande qui possède tous les avantages d'une commande feedback, plus la possibilité d'anticiper sur la réponse à la commande et sur l'effet des perturbations (quand elles sont mesurables).

Commande bas niveau / Commande haut niveau

Les robots sont souvent équipés d'actionneurs asservis localement (servo-moteurs, etc.) et recevant leurs ordres (signaux de références) de boucles de régulation de plus haut niveau.



La commande du robot est ici hiérarchisée :

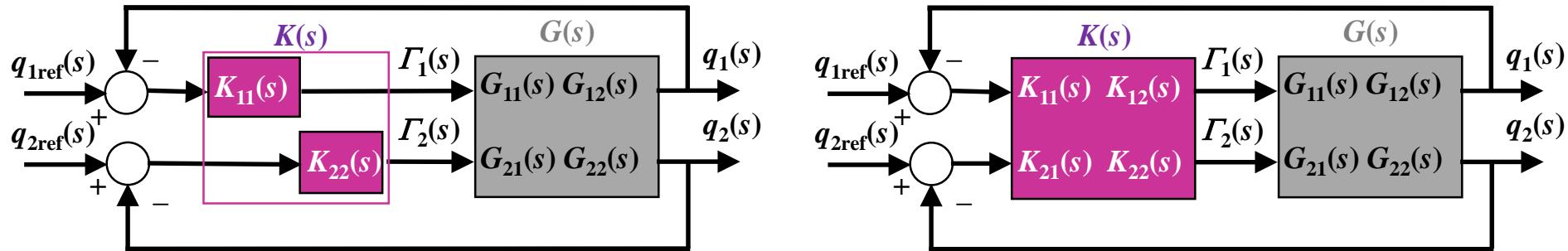
- boucle locale de bas niveau (inner loop)
- boucle de niveau intermédiaire (middle loop)
- boucle de haut niveau (outer loop)

Précision ↘
Bande passante ↘

Il existe même des boucles bas niveau dans les boucles bas niveau ...

Commande décentralisée/ Commande centralisée

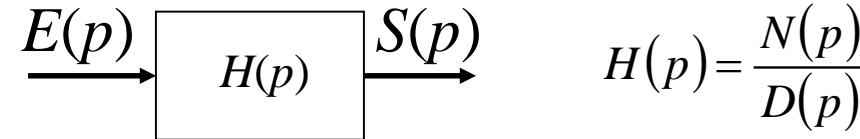
La commande bas niveau est souvent décentralisée et résulte d'une synthèse monovariable (ou **SISO**) , les termes de couplage du système étant supposés ne pas perturber l'asservissement de chacune des variables articulaires.



Pour un robot fortement couplé, une commande centralisée et donc une synthèse multivariable (ou **MIMO**) est nécessaire.

Pour un robot modérément couplé, les termes de couplage peuvent être pris en compte lors de la synthèse du régulateur décentralisé.

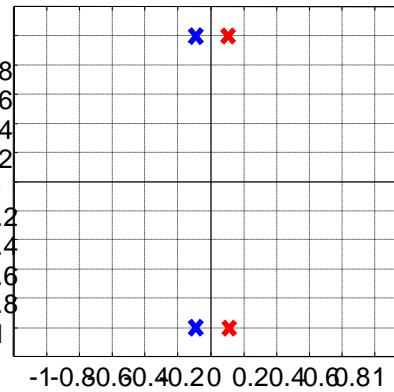
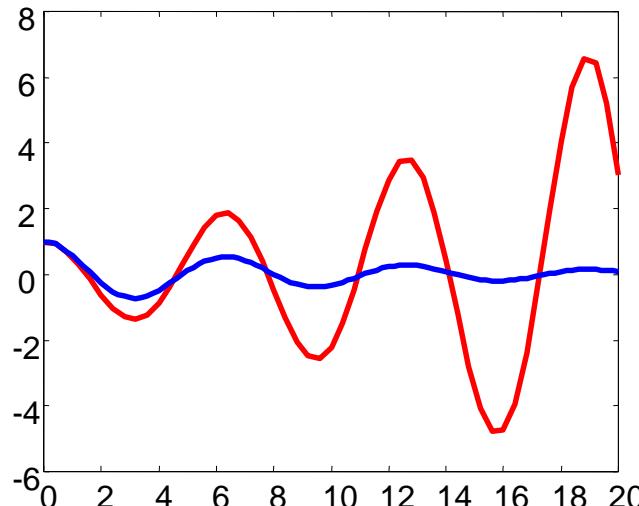
Stabilité des systèmes



Un système est stable si son régime libre ($e(t) = 0$) est amorti.

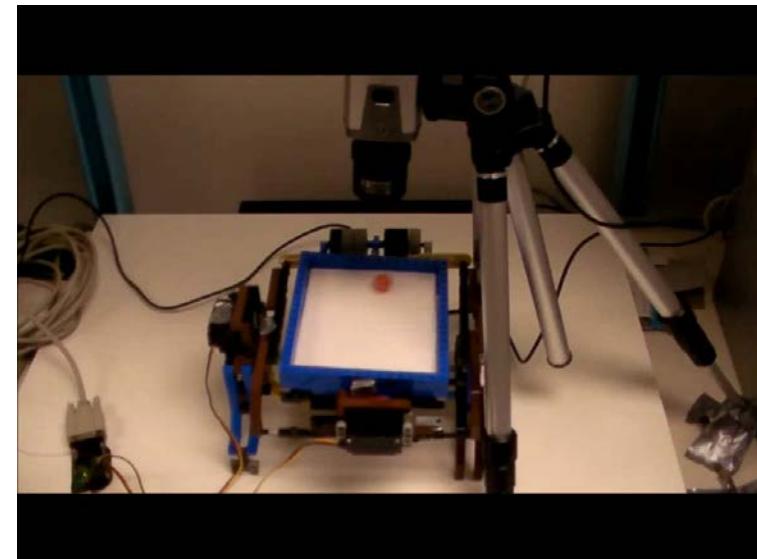
La condition nécessaire et suffisante de stabilité est que les racines de l'équation caractéristique $D(p) = 0$ soient à partie réelle négative.

Ces racines (qui rendent $H(p)$ infini) sont appelées *pôles du système*.

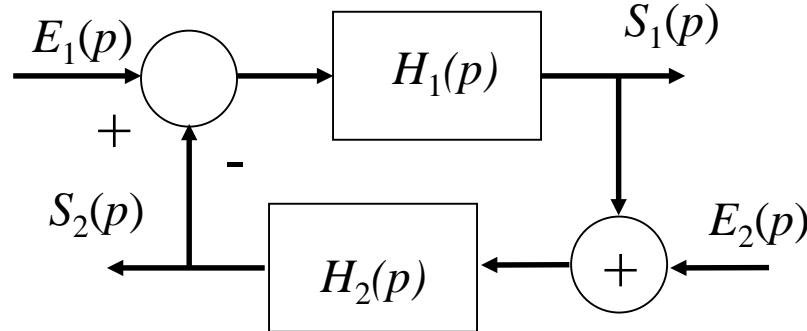


$$p_{1,2} = -0,1 \pm j$$

$$p_{1,2} = 0,1 \pm j$$



Stabilité des systèmes bouclés



$$H(p) = \frac{S_1(p)}{E_1(p)} = \frac{H_1(p)}{1 + H_1(p)H_2(p)}$$

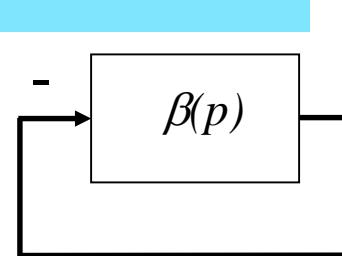
Les pôles de $H(p)$ sont les solutions de l'équation caractéristique $1 + H_1(p)H_2(p) = 0$, ou de $H_1(p)H_2(p) = -1$.

Les pôles de $H(p)$ résultent donc de la comparaison du transfert $H_1(p)H_2(p)$ à -1 .

A travers le terme $1 + H_1(p)H_2(p)$, le transfert $H_1(p)H_2(p)$ apparaît dans toutes les fonctions de transfert relative à ce système bouclé. Il est appelé ***transfert en boucle ouverte***.

Critère graphique

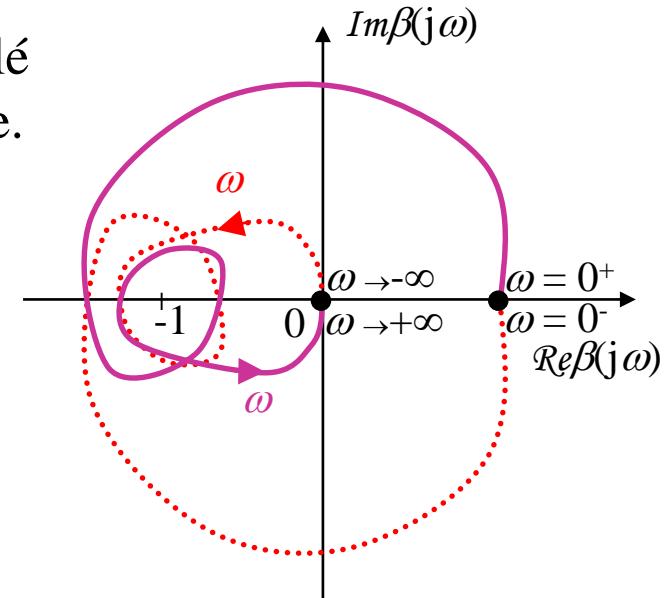
Critère de Nyquist (application du théorème de Cauchy)



Permet de conclure sur la stabilité du système bouclé à partir de l'étude de son transfert en boucle ouverte.

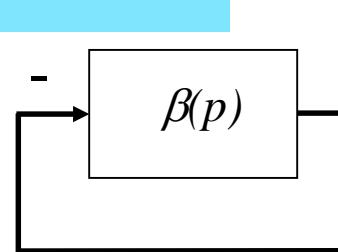
Enoncé : Un système est stable en boucle fermée si sa réponse fréquentielle en boucle ouverte, parcourue de $\omega \rightarrow -\infty$ à $\omega \rightarrow +\infty$, effectue autour du point -1 dans le sens trigonométrique, un nombre de tour égal au nombre de pôles à partie réelle strictement positive que possède la boucle ouverte.

Signalons que $\beta(-j\omega)$ est le symétrique de $\beta(j\omega)$ avec $\omega \in [0, -\infty[$.



-1 entouré 4 fois → boucle fermée stable si boucle ouverte avec 4 pôles à partie réelle strictement positive

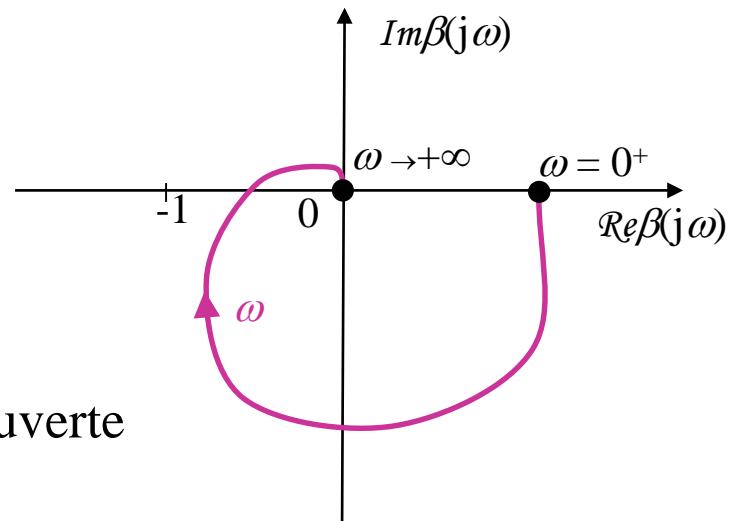
Marges de stabilité



La stabilité en boucle fermée se jugeant sur la distance entre le lieu de Nyquist en boucle ouverte et le point -1 , on est amené à mesurer des marges de stabilité :

- *marge de phase*
- *marge de gain*
- *marge de retard*
- *marge de module*

Elles correspondent aux plus petites quantités dont il serait nécessaire de modifier la boucle ouverte pour déstabiliser le système en boucle fermée.



Grandes marges de stabilité → grand degré de stabilité du système bouclé.

Marge de phase

La marge de phase M_Φ est la valeur minimale du déphasage caractérisant Δ (déphaseur pur) qui déstabilisera le système bouclé de boucle ouverte β :

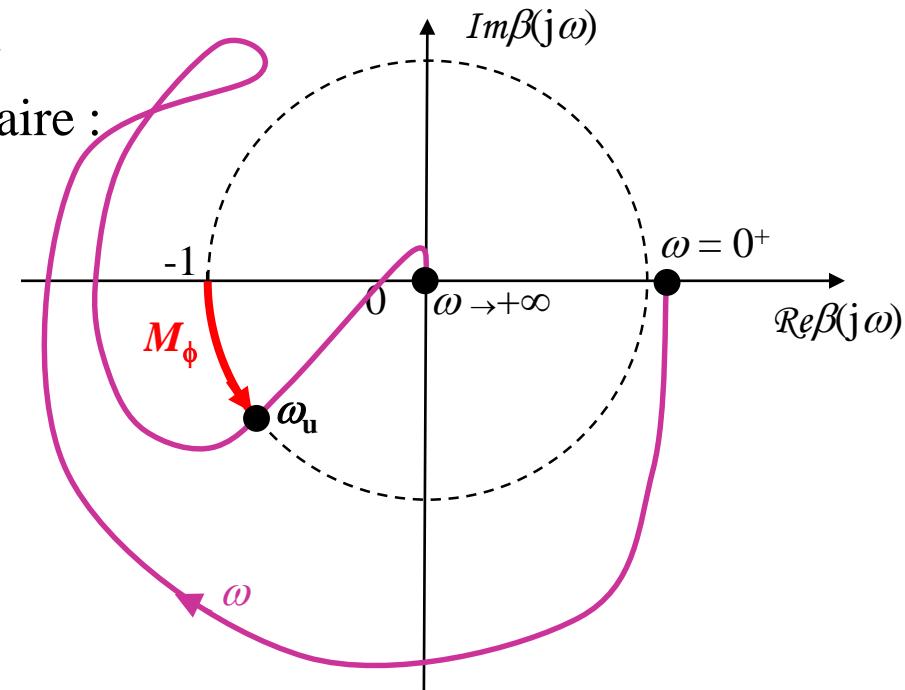
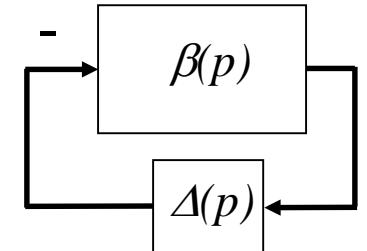
$$\Delta(j\omega)\beta(j\omega) = -1 \quad \text{avec} \quad \arg\Delta(j\omega) = -M_\Phi$$

Elle se mesure donc à une pulsation ω_u pour laquelle le gain de la boucle ouverte est unitaire :

$$|\beta(j\omega_u)| = 1$$

et est ici définie par :

$$M_\Phi = 180^\circ + \arg \beta(j\omega_u)$$



Marge de gain

La marge de gain M_G est la valeur minimale du gain caractérisant Δ (gain pur) qui déstabilisera le système bouclé de boucle ouverte β :

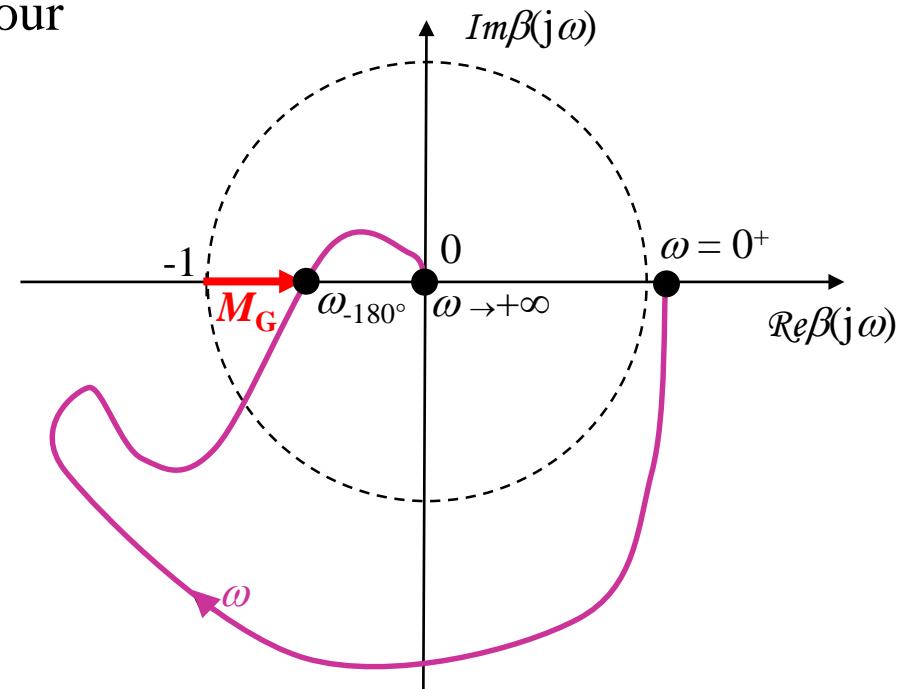
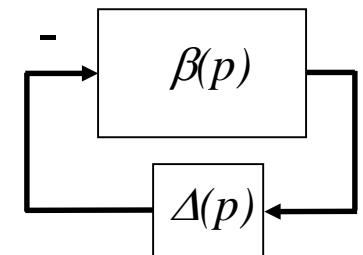
$$\Delta(j\omega)\beta(j\omega) = -1 \quad \text{avec} \quad |\Delta(j\omega)| = M_G$$

Elle se mesure donc à une pulsation ω_{-180° pour laquelle pour laquelle la phase de la boucle ouverte est de -180° :

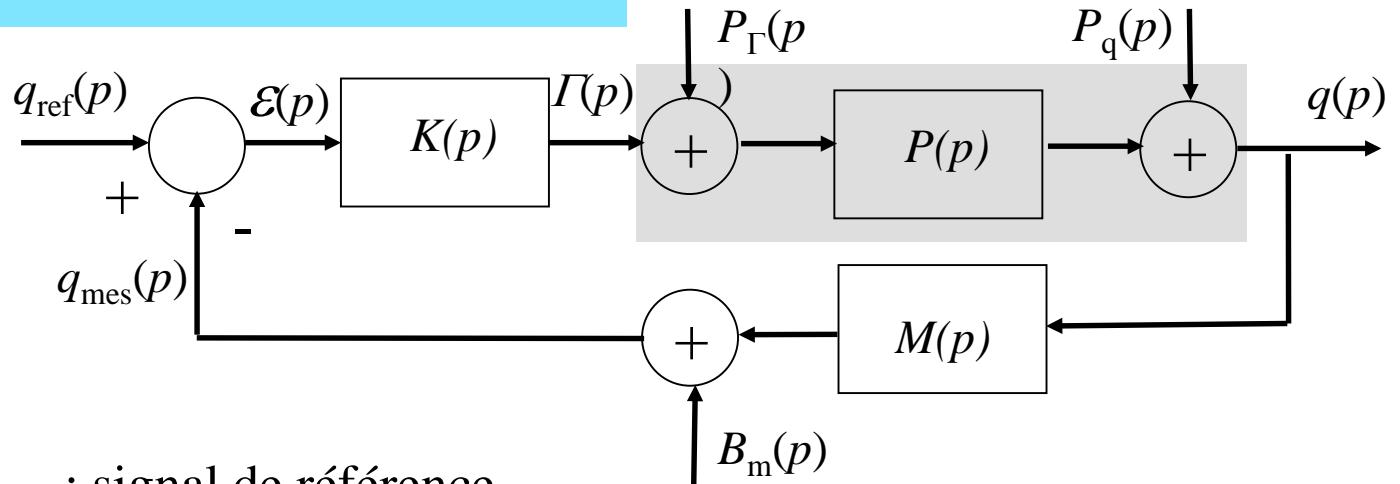
$$\arg \beta(j\omega_{-180^\circ}) = -180^\circ$$

et est ici définie (en dB) par :

$$M_G = -20 \log |\beta(j\omega_{-180^\circ})|$$



La boucle de commande



q_{ref} : signal de référence

\mathcal{E} : signal d'erreur

Γ : signal de commande

P_Γ : perturbation d'entrée

P_q : perturbation de sortie

q : sortie du procédé

B_m : bruit de mesure

q_{mes} : signal de mesure

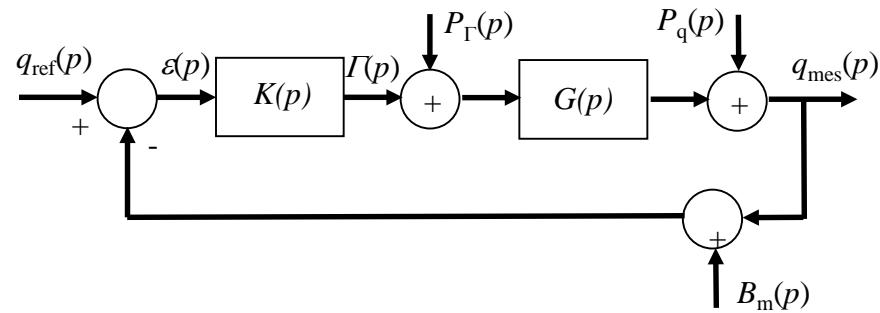
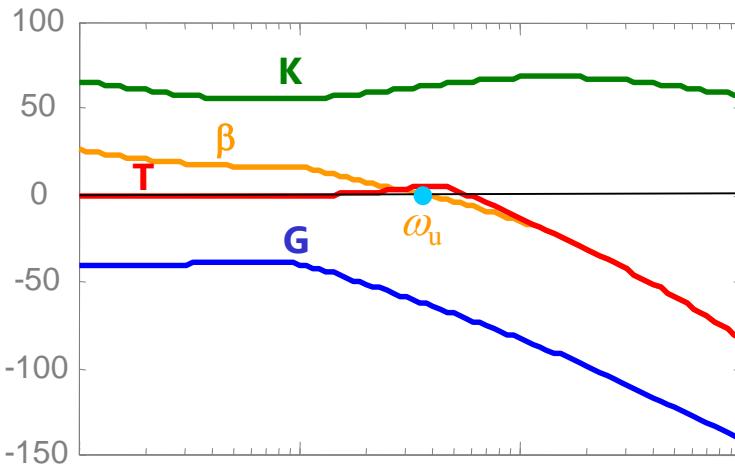
$K(p)$: transfert du régulateur

$P(p)$: modèle du système physique

$M(p)$: modèle de l'organe de mesure

Le régulateur $K(p)$ doit permettre d'asservir avec précision et rapidité le signal de sortie q (en fait sa mesure q_{mes}) sur sa valeur de référence q_{ref} , en rejetant l'effet des perturbations P_Γ et P_q .

Allure de la fonction du transfert en boucle fermée $q_{\text{mes}}/q_{\text{ref}}$



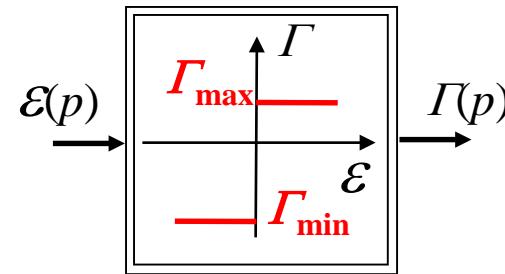
$$T(p) = \frac{q_{\text{mes}}(p)}{q_{\text{ref}}(p)} = -\frac{q_{\text{mes}}(p)}{B_m(p)} = -\frac{\Gamma(p)}{P_\Gamma(p)} = \frac{K(p)G(p)}{1 + K(p)G(p)}$$

- $\omega \ll \omega_u \rightarrow KG = \beta \gg 1 \rightarrow T \approx 1$
- $\omega \gg \omega_u \rightarrow KG = \beta \ll 1 \rightarrow T \approx \beta$

Les variations du signal de référence et du bruit de mesure sont transmises en basse fréquence. La pulsation au gain unité ω_u de β est donc du même ordre de grandeur que la bande passante de la boucle fermée.

Commande en tout-ou-rien (TOR/On-Off)

Intuitivement, la loi de commande la plus simple consiste à appliquer le maximum de commande quand la sortie mesurée est plus petite que la consigne et le minimum sinon.



Avantages :

- temps de réponse minimums et bonne précision si amortissement suffisant
- simplicité de mise en œuvre

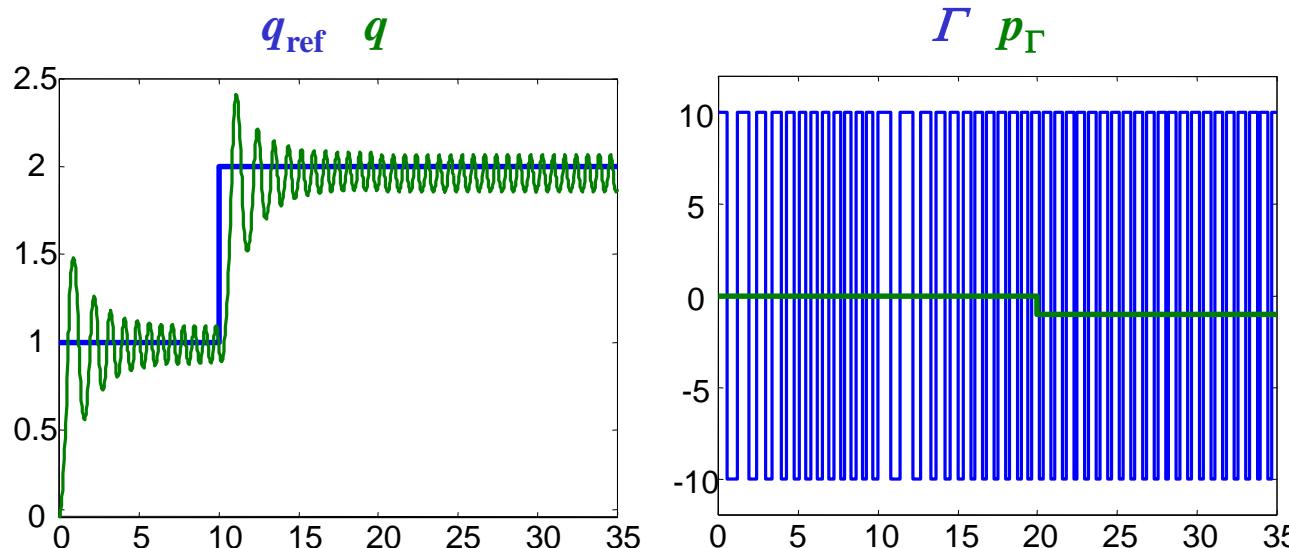
Inconvénients :

- observation fréquente d'oscillations non amorties
- dépense énergétique importante
- variations brusques du signal de commande

Commande T.O.R - Application

Le système commandé (P et M) modélisé par $G(p) = \frac{0.75}{(1 + p + p^2)(0.01p + 1)}$ subit une perturbation sur son entrée.

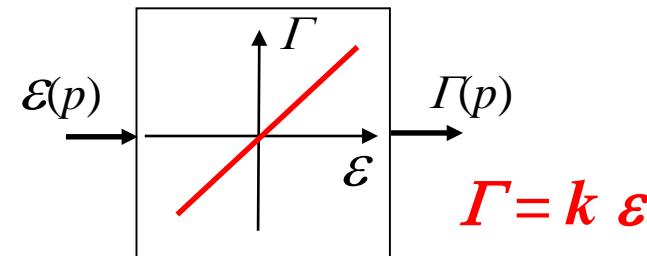
La commande TOR est définie par $\Gamma_{\max} = 10$ et $\Gamma_{\min} = -10$.



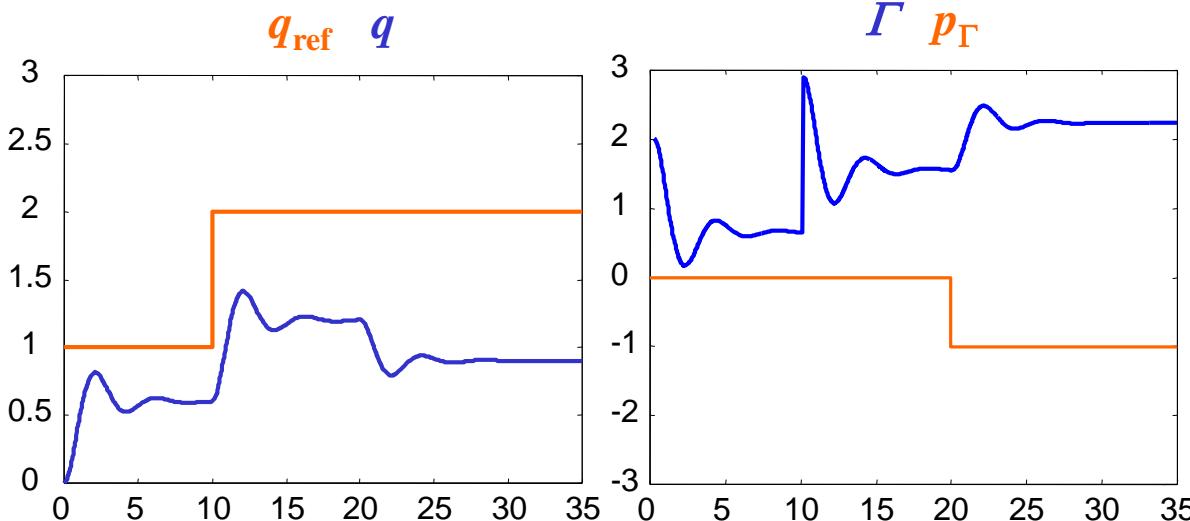
Le phénomène d'oscillation observé est appelé "pompage".

Commande proportionnelle (régulateur P)

Une loi de commande plus évoluée consiste à générer une commande proportionnelle à la différence entre le signal de référence et la sortie mesurée.

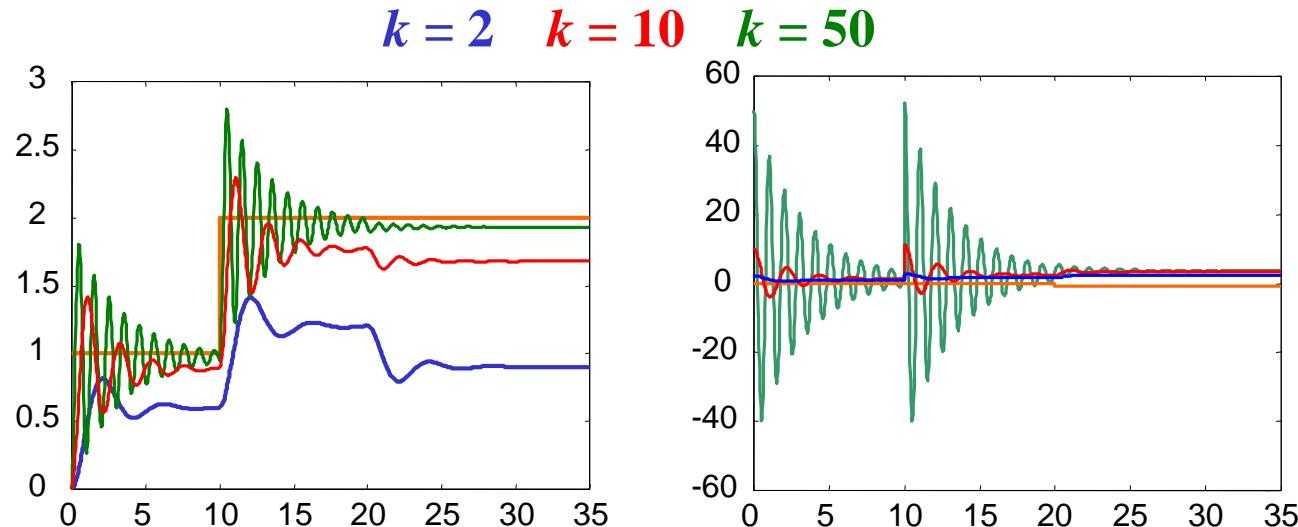


Le procédé précédent est commandé avec un gain $k = 2$



On observe ici un suivi de consigne et un rejet de perturbation imparfaits.

Commande proportionnelle - Dilemme stabilité/précision



Quand k augmente :

- + l'erreur statique ε_∞ tend vers 0
- + le gain statique T_0 tend vers 1
- + le temps de montée t_m diminue
- le premier dépassement D_1 augmente
- l'amortissement ζ diminue
- le temps de réponse $t_{5\%}$ augmente
- le niveau de commande augmente

Précision ↗

Degré de stabilité ↘

Commande à action intégrale (régulateur PI)

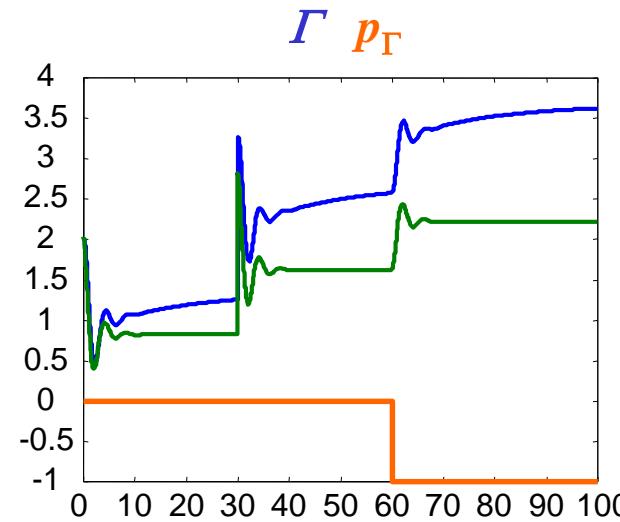
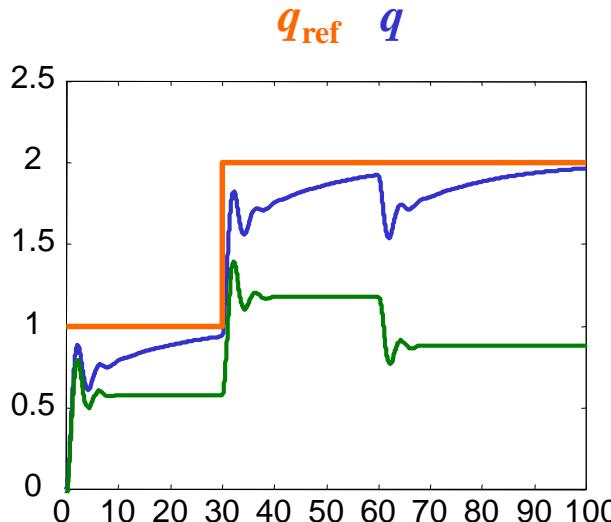
L'augmentation du gain de boucle ouverte en basse fréquence peut être obtenu avec un filtre du type :

$$\frac{1 + p/\omega_i}{p/\omega_i}$$

Son gain tend vers l'infini en basse fréquence, et tend vers 1 au delà de ω_i .

Le filtre est donc transparent au voisinage de la pulsation au gain unité en boucle ouverte ω_u si $\omega_i < \omega_u$.

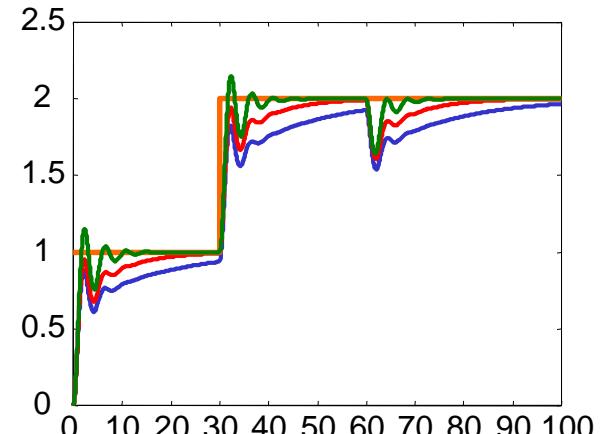
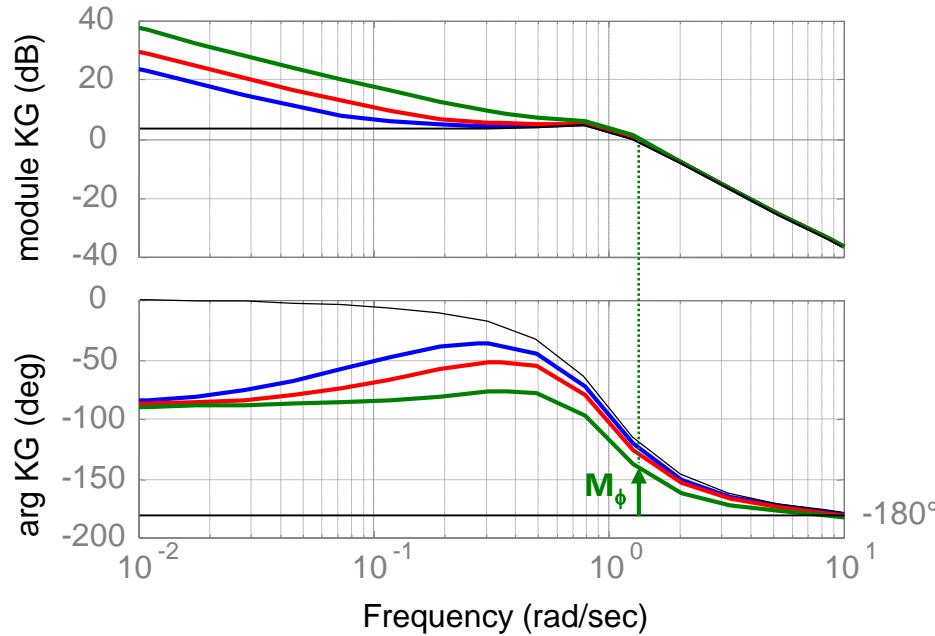
$$K(p) = 2 \quad K_I(p) = 2 \frac{1 + p/0.1}{p/0.1}$$



La précision parfaite n'est ici obtenue qu'après un temps relativement long.

Commande à action intégrale - Accélération de la convergence, diminution du degré de stabilité

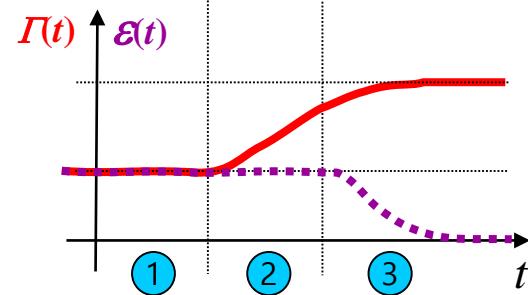
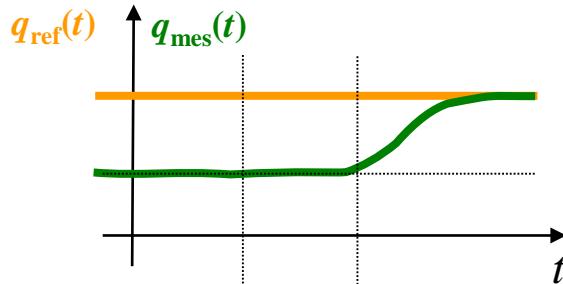
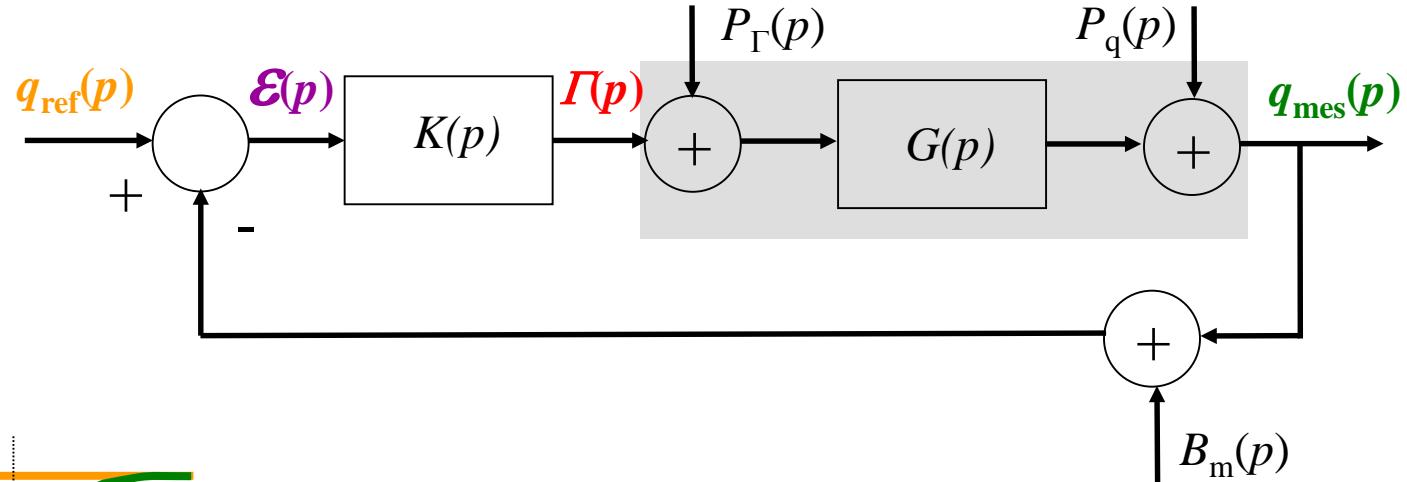
$$K(p) = 2 \quad \omega_i = 0.1 \text{ rad/s} \quad \omega_i = 0.2 \text{ rad/s} \quad \omega_i = 0.5 \text{ rad/s}$$



Quand ω_i augmente :

- + augmentation du gain de boucle ouverte en basse fréquence
- diminution de la marge de phase M_ϕ et donc du degré de stabilité

Commande à action intégrale "expliquée avec les mains"



Phase 1 : commande proportionnelle
 Phase 2 : passage en commande à action intégrale
 Phase 3 : déblocage de q_{mes} et stabilisation
 Le régulateur force de plus en plus tant que nécessaire

$$\Gamma_{kT} = \Gamma_{(k-1)T} + \alpha_I \mathcal{E}_{kT}$$

Si α_I est trop grand, Γ_{kT} risque d'être trop grand et générer des oscillations.

Commande à action dérivée (régulateur PD)

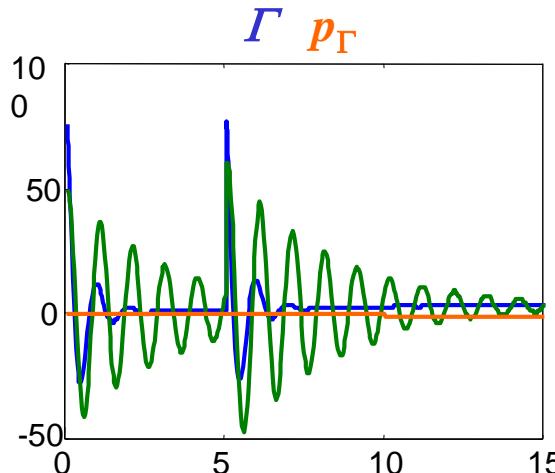
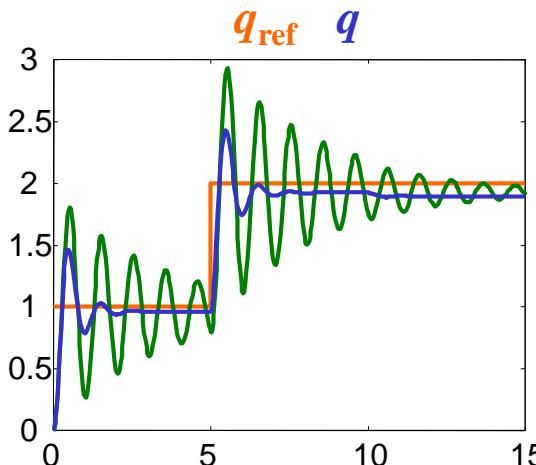
L'augmentation de la marge de phase peut être obtenue avec un filtre du type :

$$\sqrt{\frac{\omega_1}{\omega_2}} \frac{1 + p/\omega_1}{1 + p/\omega_2}$$

Ce filtre génère une avance de phase maximale à $\sqrt{\omega_1\omega_2}$.

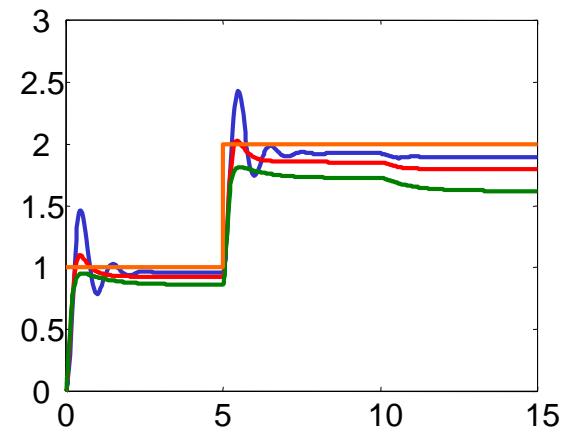
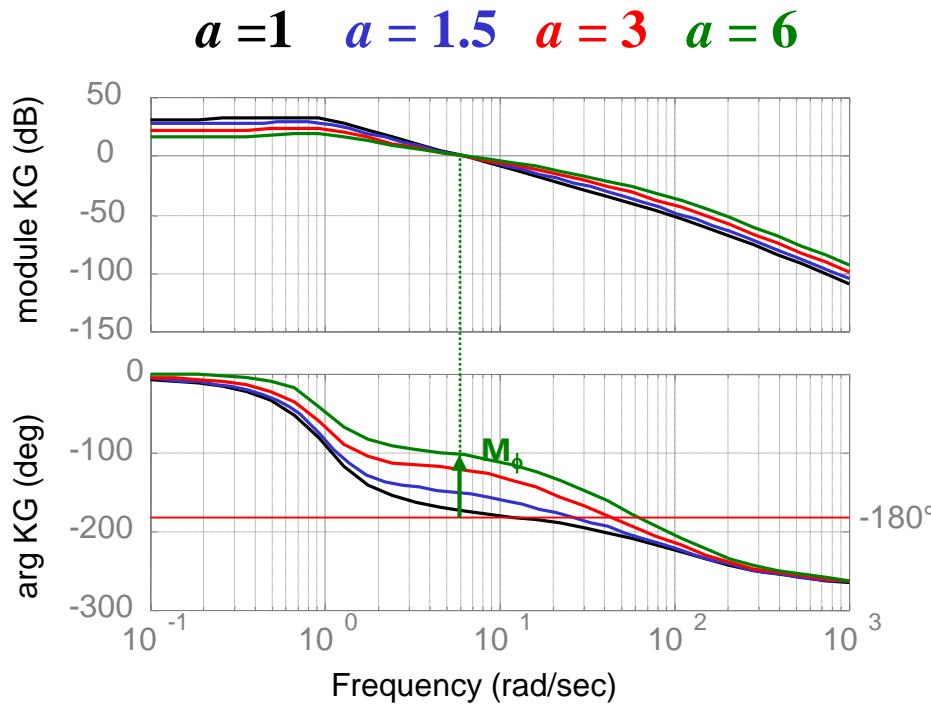
Le gain $\sqrt{\omega_1/\omega_2}$ inférieur à 1 dégrade légèrement la précision mais rend le filtre transparent en gain à $\sqrt{\omega_1\omega_2}$.

$$K(p) = 50 \quad K_D(p) = 50/1.5 \frac{1 + p/4}{1 + p/9}$$



Le premier dépassement a été diminué mais reste important.

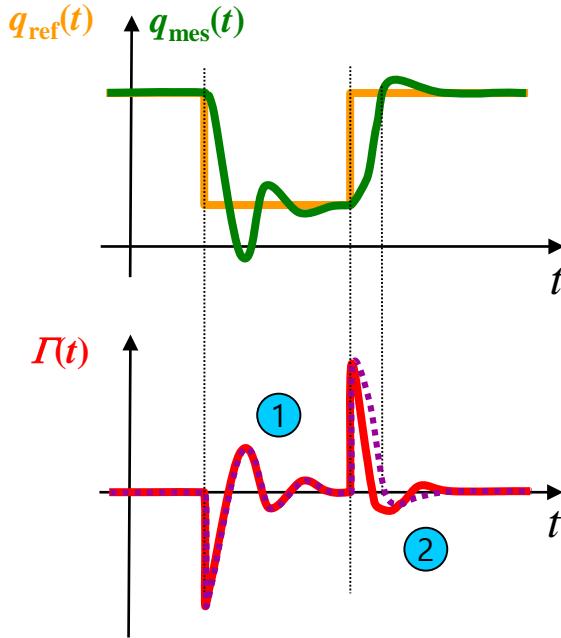
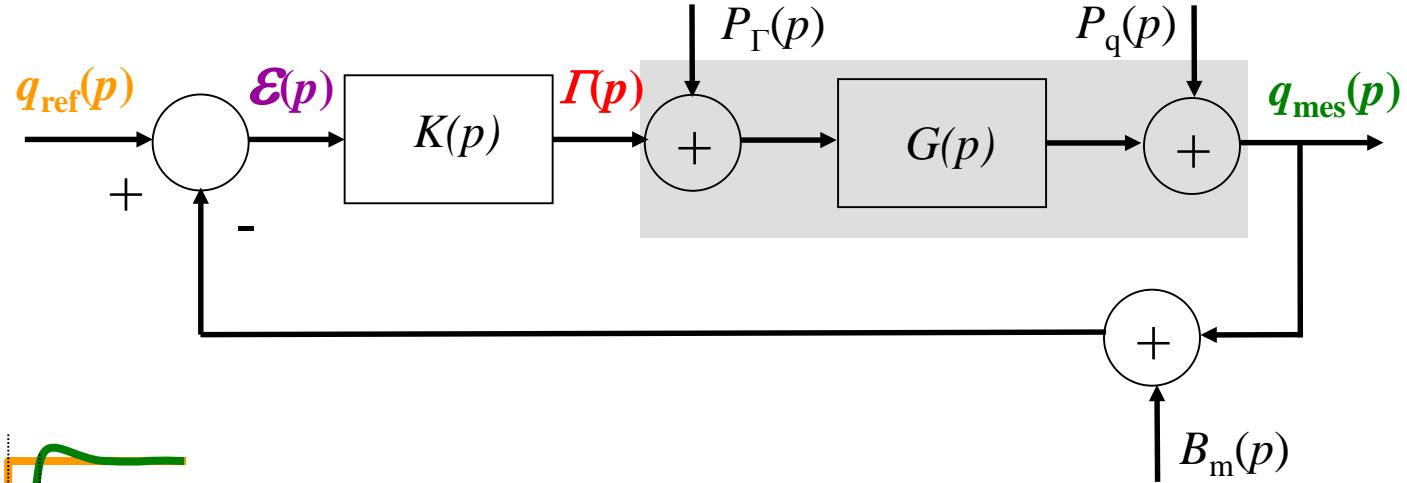
Commande à action dérivée - augmentation de l'avance de phase, diminution des performances



Quand a augmente :

- + augmentation de la marge de phase M_ϕ
 - diminution du gain de boucle ouverte en basse fréquence
 - augmentation du gain de boucle ouverte en haute fréquence
- ⇒ apporter que le minimum de phase nécessaire.

Commande à action dérivée "expliquée avec les mains"



Phase 1 : commande proportionnelle

Phase 2 : ajout d'une action dérivée
anticipation de u pour contrer l'inertie de G

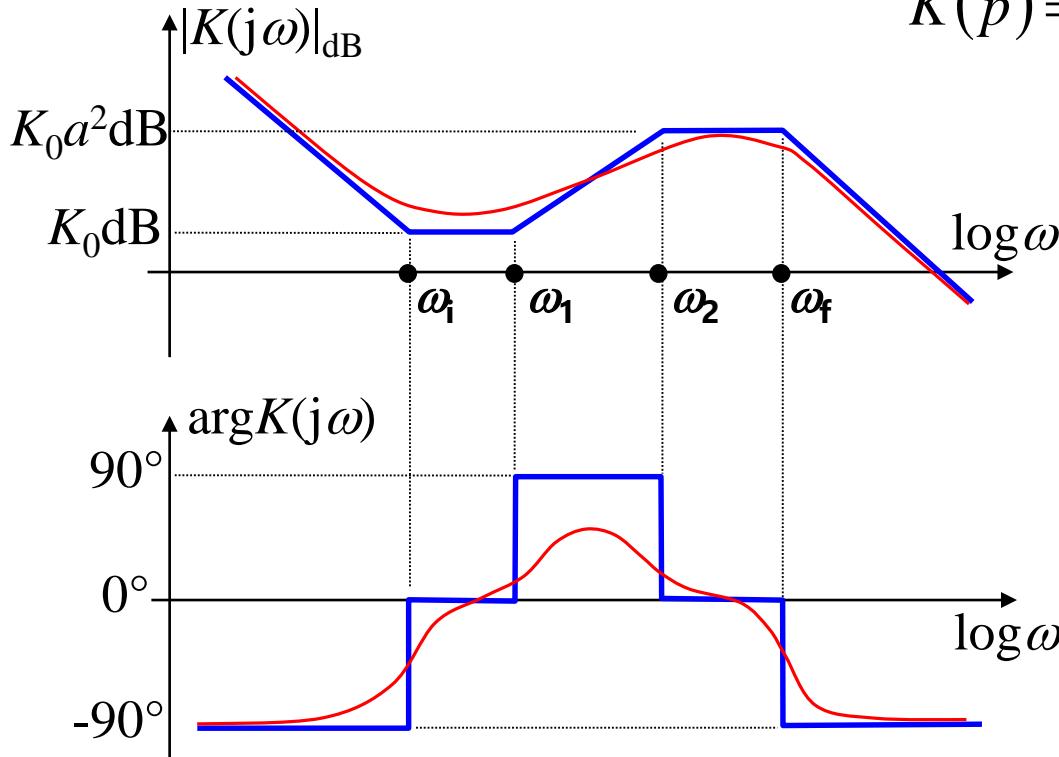
Le régulateur tient compte de la dérivée de \mathcal{E}

$$\Gamma_{kT} = \alpha_P \mathcal{E}_{kT} + \alpha_D (\mathcal{E}_{kT} - \mathcal{E}_{(k-1)T})$$

Si α_D est trop grand, u_{kT} risque d'être trop sensible au bruit de mesure.

Commande de type PID(F) (Proportionnelle, Intégrale, Dérivée)

Permet de conjuguer les avantages de chacune des actions prises séparément.
A ces 3 actions, une action de filtrage peut être ajoutée pour assurer un gain décroissant du régulateur en haute fréquence.

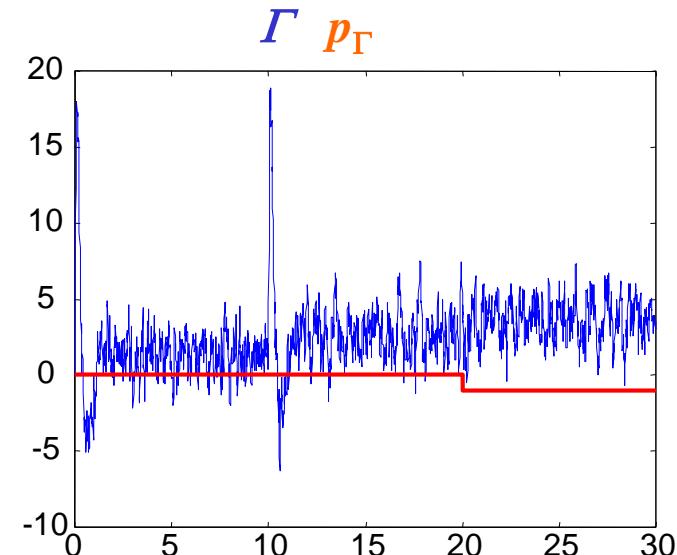
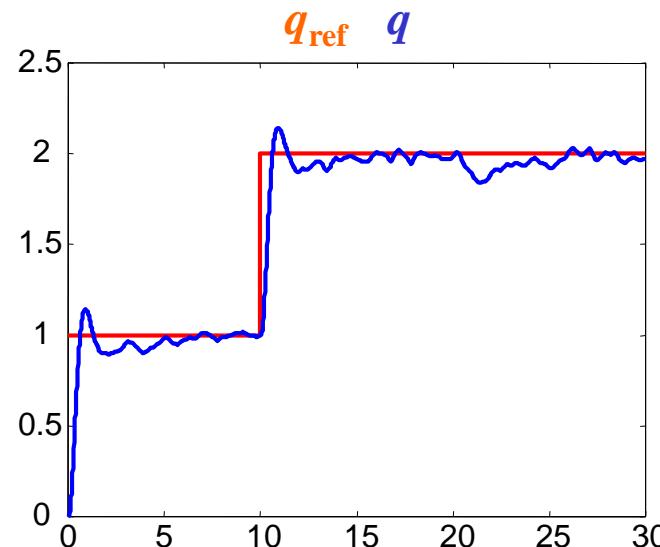


$$K(p) = K_0 \frac{1 + p/\omega_i}{p/\omega_i} \frac{1 + p/\omega_1}{1 + p/\omega_2} \frac{1}{1 + p/\omega_f}$$

P \Rightarrow rapidité
I \Rightarrow précision
D \Rightarrow amortissement
F \Rightarrow filtrage

- $\omega \in]0, \omega_i]$: effet intégrateur
- $\omega \in [\omega_i, \omega_1]$: effet proportionnel
- $\omega \in [\omega_1, \omega_2]$: effet dérivateur
- $\omega \in [\omega_2, \omega_f]$: effet amplificateur
- $\omega \in [\omega_f, \infty[$: effet filtre passe-bas

Commande de type PID - Application



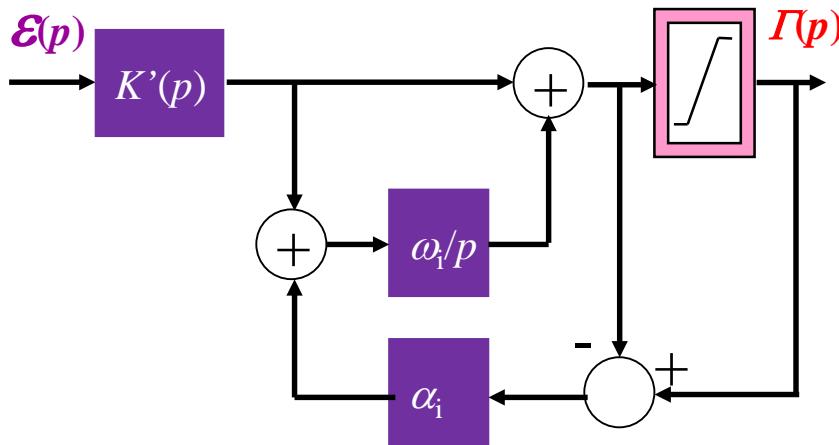
L'erreur statique est nulle, le bruit de commande de l'ordre de ± 2 et les transitoires sont bien amortis.

Une variation unitaire du signal de référence génère une variation de la commande de l'ordre de 20. Si valeur maxi de commande, variations supérieures de la sortie désirée à filtreravant d'être appliquées en signal de référence.

Commande de type PID - Implantation

- ❖ Introduction d'un système de désaturation (*anti-windup*)

$$K(p) = K'(p) \left(1 + \frac{\omega_i}{p} \right)$$



- ❖ Numérisation en utilisant $p \approx \frac{1 - z^{-1}}{T_e}$

$K(p) \rightarrow K(z^{-1})$ et les *équations récurrentes* correspondantes.

```

void BRAM_PIDControl(unsigned int sensor_val)
{
    int motor_res,err_func;
    float KP,KI,KD,cont_res;

    // Get the Error Function
    err_func=sensor_val - TARGET_VAL;

    // Divide all the PID parameters for decimal value
    KP=Kp * 0.1;
    KI=Ki * 0.01;
    KD=Kd * 0.01;

    // Calculate the Motor Response using PID Control Equation
    cont_res=(float)(prev_res + KP * (err_func - prev_err_1) + KI * (err_func + prev_err_1)/2.0
                    + KD * (err_func - 2.0 * prev_err_1 + prev_err_2));

    // Now we have to Limit the control response to the Maximum of our motor PWM Motor Value
    motor_res=(int)cont_res;

    if (motor_res > MaxSpeed)
        motor_res = MaxSpeed;
    if (motor_res < -MaxSpeed)
        motor_res = -MaxSpeed;

    // Save the Motor Response and Error Function Result
    prev_res=motor_res;
    prev_err_2=prev_err_1;
    prev_err_1=err_func;

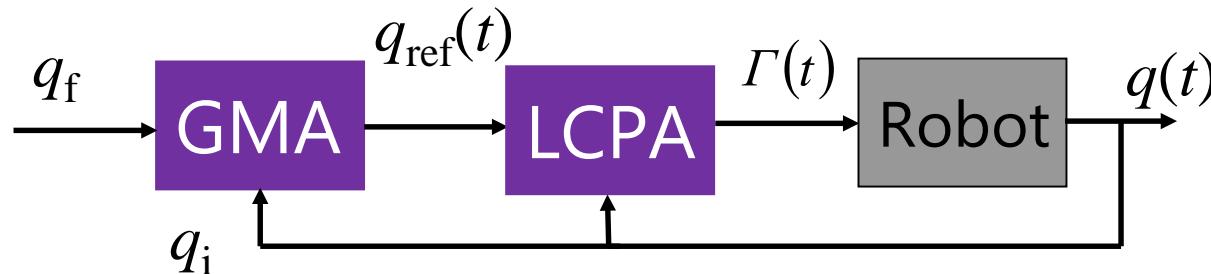
    // Display Motor Response Value in Debugging Mode
#if BRAM_DEBUG
    ansi_cm(4,1);
    printf("Motor Response Factor: %d    ",motor_res);
#endif

    // Negative result mean BRAM is on the right, so we need
    // Positive result mean BRAM is on the left, so we need
    if (motor_res < 0)
        BRAM_DriveMotor(MaxSpeed + motor_res,MaxSpeed); // Left full speed, Right less speed
    else
        BRAM_DriveMotor(MaxSpeed,MaxSpeed - motor_res); // Left full speed, Right less speed
}
  
```

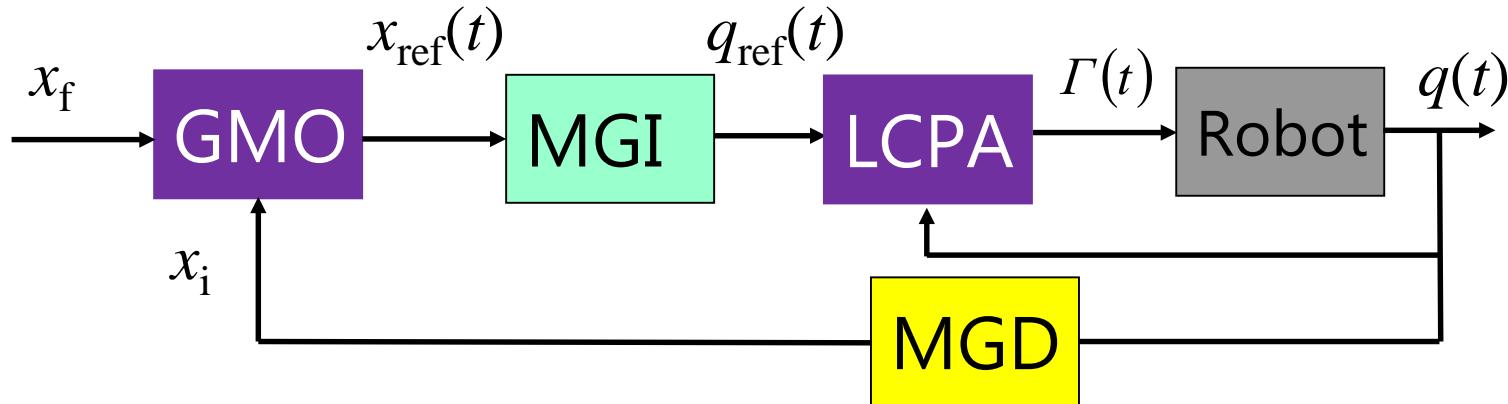


Génération de mouvement

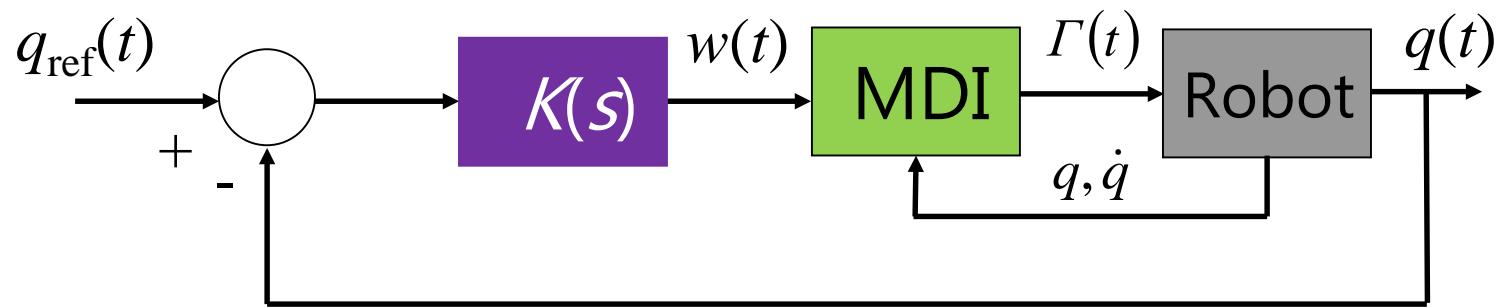
- ❖ Travail dans l'espace articulaire



- ❖ Travail dans l'espace opérationnel (sans mesure de x)



Commande par découplage non linéaire (Computed Torque)



Sachant que $\Gamma = A(q)\ddot{q} + H(q, \dot{q})$ avec $\Gamma = \hat{A}(q)w + \hat{H}(q, \dot{q})$
 on obtient naturellement $w \approx \ddot{q}$ ou $q \approx \int \int w$

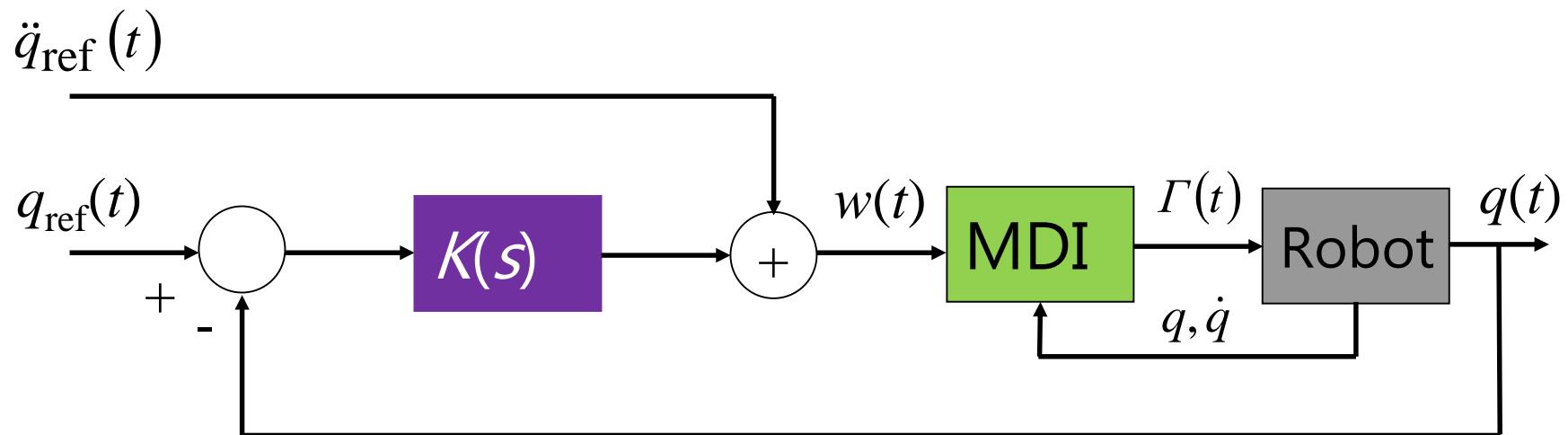
Aussi compliqué soit le robot, le régulateur (feedback K) doit alors asservir un nouveau système dont le comportement est celui d'un double intégrateur.

... bien évidemment si $\hat{A}(q) \approx A(q)$ et $\hat{H}(q, \dot{q}) \approx H(q, \dot{q})$

Si q_{ref} et \dot{q}_{ref} utilisés dans le MDI, la commande est dite dynamique prédictive.

Commande par découplage et feedforward

Si \ddot{q}_{ref} est spécifié ou calculable, il est utilisable comme entrée supplémentaire de commande (feedforward).



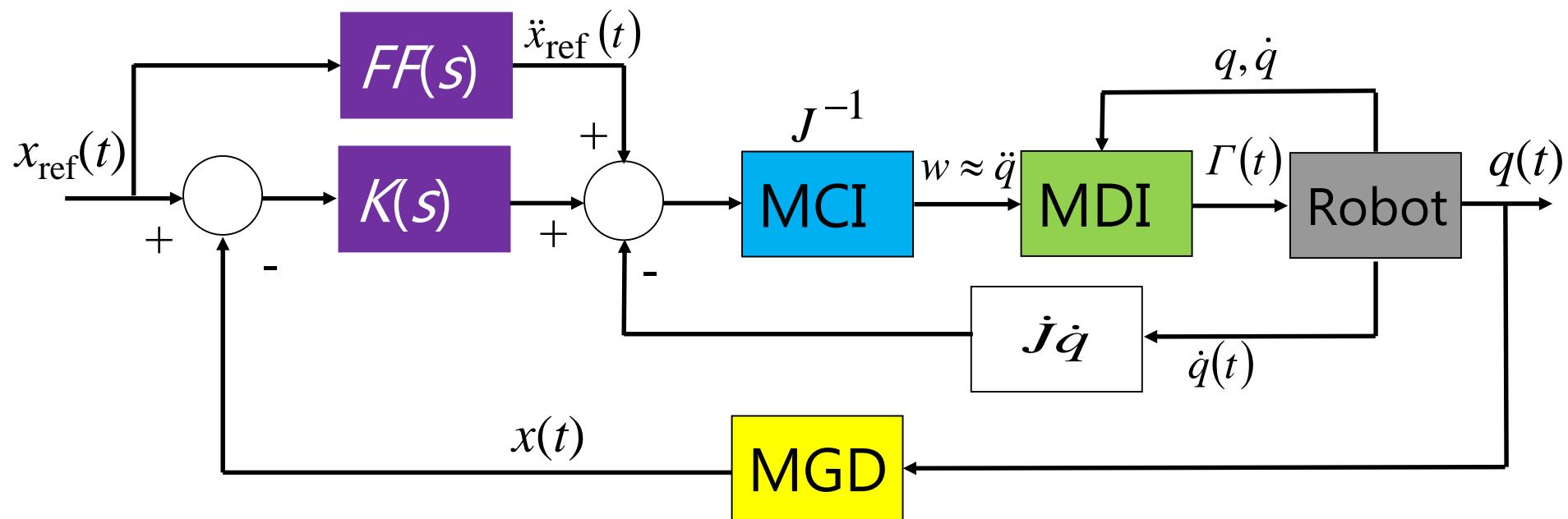
Commande dynamique dans l'espace opérationnel

L'idée est d'utiliser l'équation du Modèle Cinématique Direct de 2ème ordre :

$$\ddot{x} = J\dot{q} + J\ddot{q}$$

```

graph LR
    A[q(t)/J(t)] --> B[MCD2]
    B -- "x double dot" --> C
  
```



Positionnement et interaction avec l'environnement

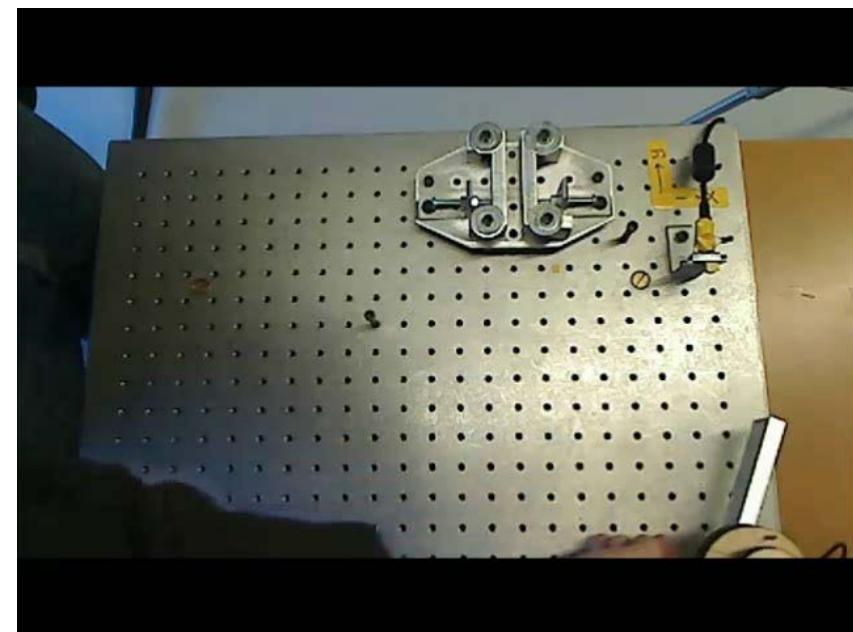
Le point terminal n'est pas libre si en contact avec l'environnement \Leftrightarrow augmentation des efforts quand l'erreur de position tente de se réduire :

- risque de rebond
- risque d'instabilité
- risque de détérioration du robot et/ou de l'environnement

Idées d'une solution :

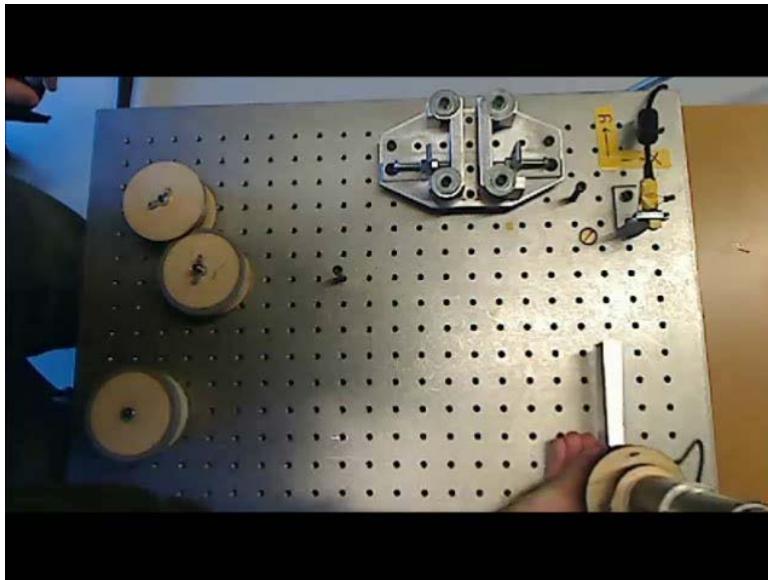
- mesurer *parfaitement* la position du robot
- mesurer *parfaitement* la position de l'environnement
- disposer d'une *parfait* précision de positionnement

Excessivement cher à mettre en œuvre.

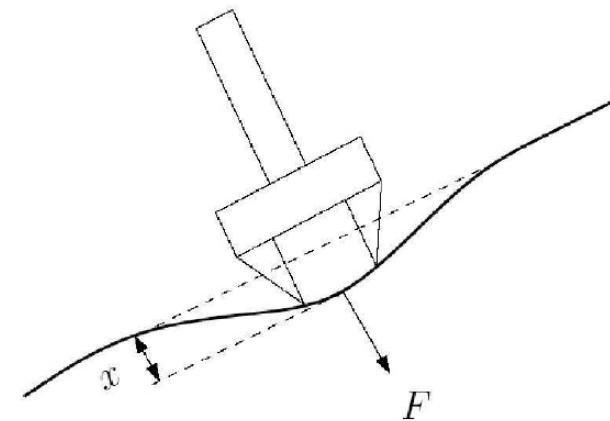


Commande en force

Un ensemble de solutions consiste à prendre implicitement ou explicitement en compte l'effort d'interaction entre l'organe terminal du robot et son environnement.



Commande en force pure

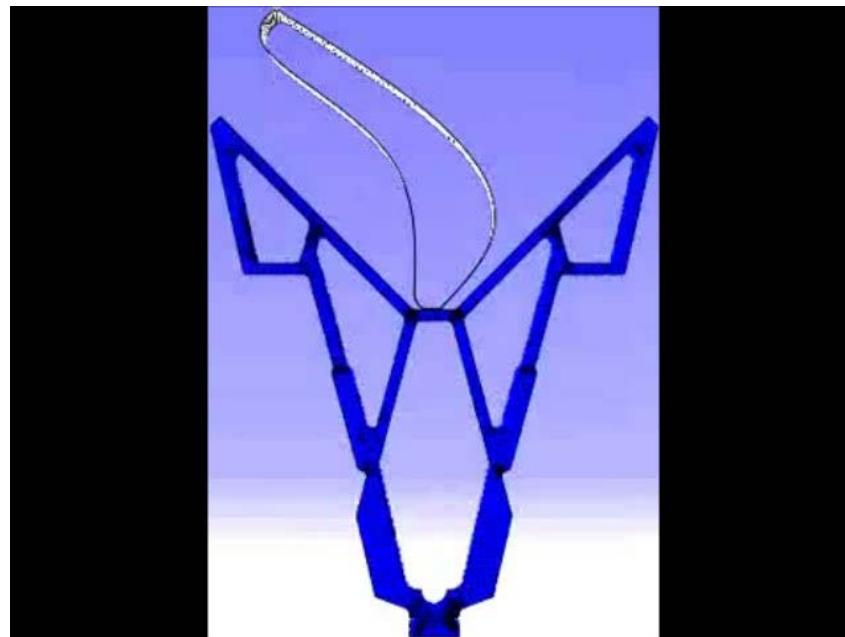


$$F = k \cdot x$$

Généralement connue pour être l'inverse de la raideur, on définit la **compliance** (de l'environnement et/ou du robot) comme la notion associant position et maîtrise de l'effort.

Compliance passive

Commande en force sans mesure de l'effort qui utilise un dispositif déformable entre le robot rigide et l'environnement.



Ne permet pas un positionnement précis du point terminal du robot.

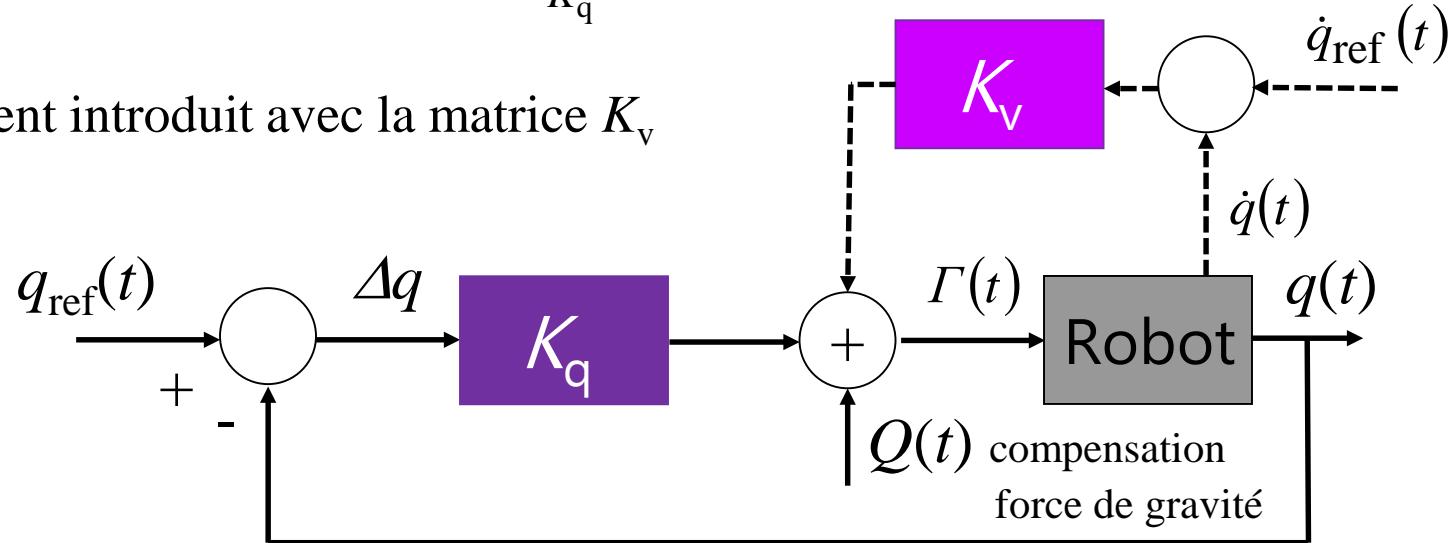
Compliance active (sans mesure de l'effort)

Les forces/couples sont définis par $F = K_x \Delta x$, K_x ayant des gains forts pour les directions à réguler en position, et faibles pour les directions où l'effort est à respecter.

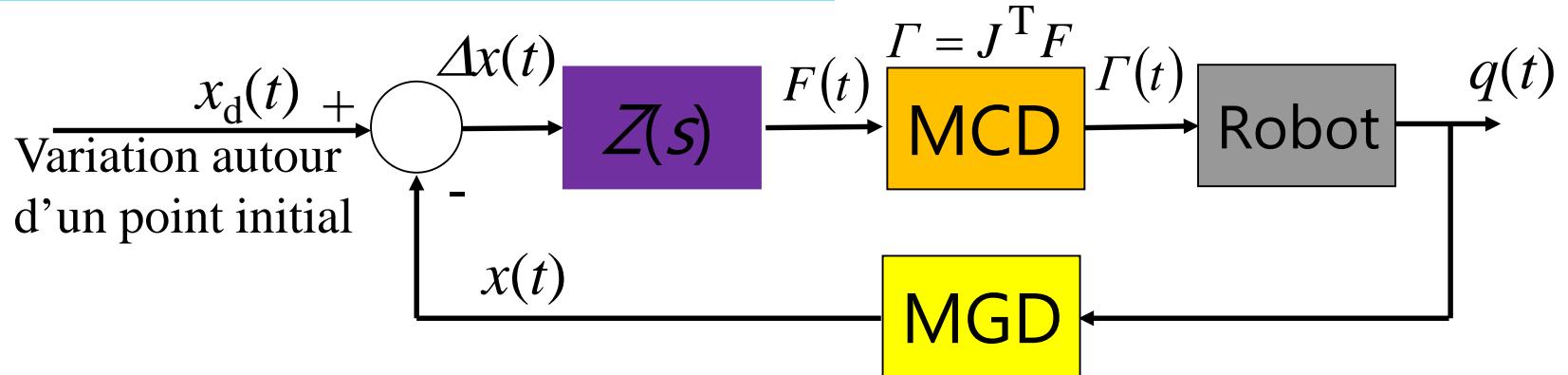
Sachant que $\Delta x = J\Delta q$ et $\Gamma = J^T F$, les couples à appliquer aux actionneurs sont :

$$\Gamma = \underbrace{J^T K_x J}_{K_q} \Delta q$$

Amortissement introduit avec la matrice K_v



Commande en impédance (sans mesure de l'effort)



L'impédance est définie par : $Z(s) = \frac{F(s)}{\Delta x(s)}$

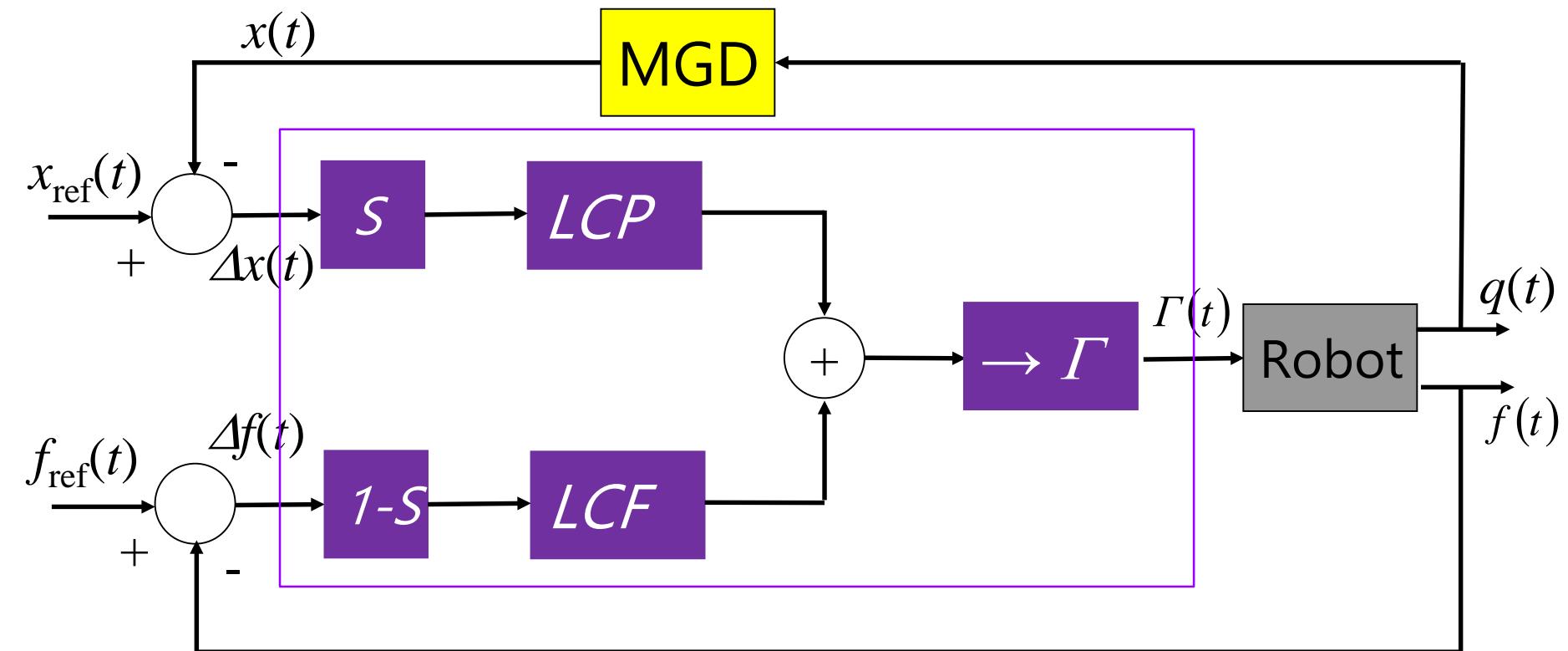
Si $F = A\Delta\ddot{x} + B\Delta\dot{x} + K\Delta x$

alors $Z(s) = As^2 + Bs + K$

**TESTS PERFORMED ON A
LINEAR AXIS OF A SCARA
ROBOT**

BALL SCREW 5 mm/rev
6 Axis Force Sensor Feedback

Commande hybride (avec mesure de l'effort)

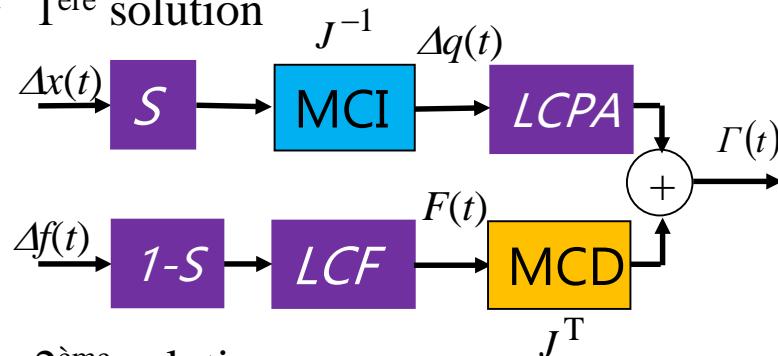


$S_{jj} = 1$ si régulation de position sur l'axe j

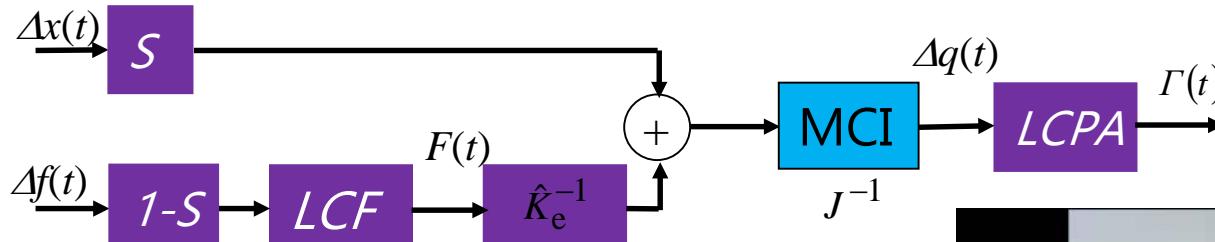
$S_{jj} = 0$ si régulation d'effort sur l'axe j

Commandes hybrides (3 solutions)

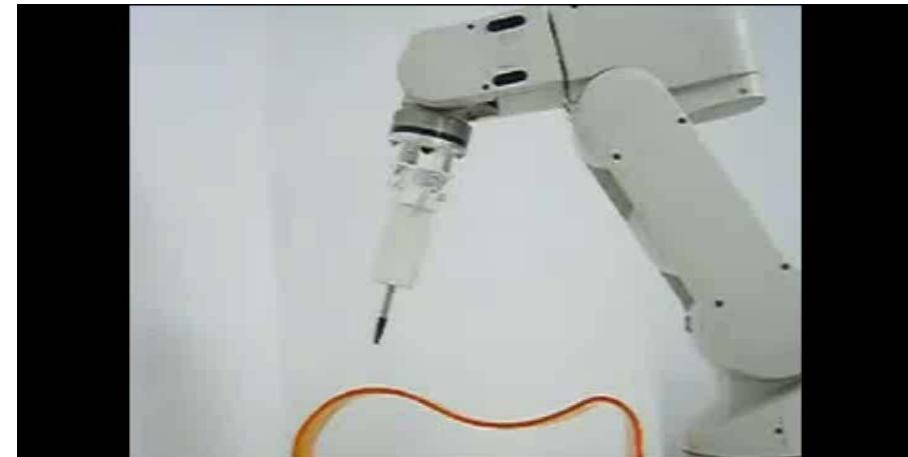
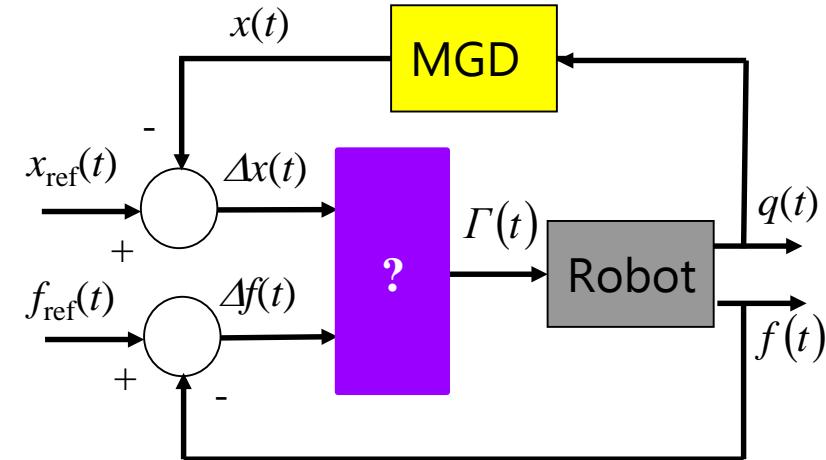
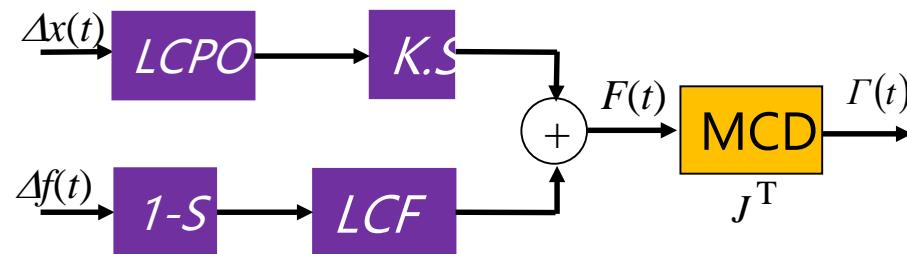
❖ 1^{ère} solution



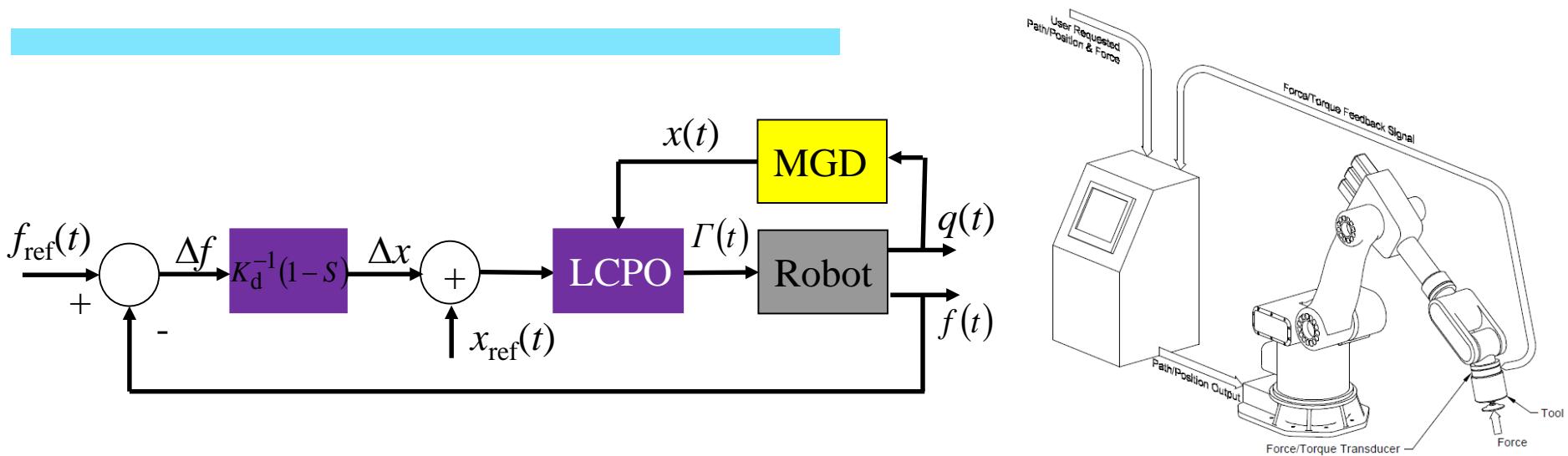
❖ 2^{ème} solution



❖ 3^{ème} solution



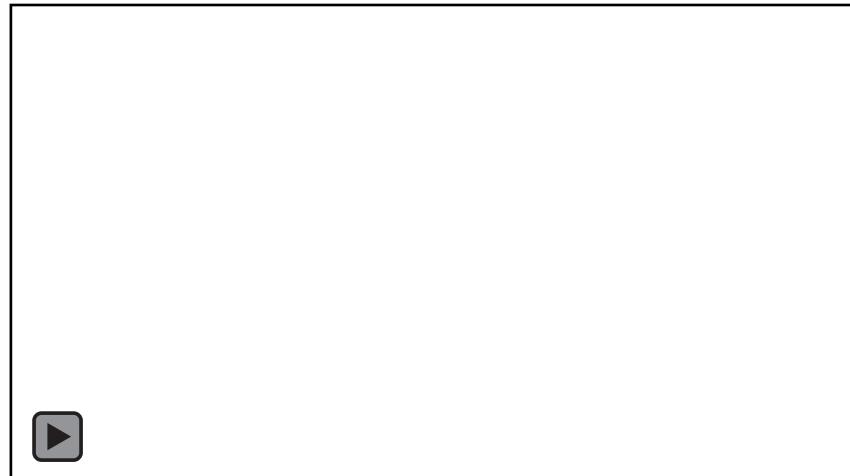
Commande hybride externe (boucles imbriquées)



K_d : raideur désirée ($\Delta f / \Delta x$) suivant les différents axes.

La différence d'effort constatée modifie la consigne de position x_{ref}

$$x \rightarrow x_{ref} + K_d^{-1} \Delta f \text{ soit en effet } K_d \rightarrow \frac{f - f_{ref}}{x - x_{ref}}$$



Corobots et robots compliant

- Prennent en compte les efforts exercés sur/par l'environnement
- Assistent l'humain

