

Nicolas Petit

---

# Commande prédictive

---

Notes de cours  
Option Procédé Environnement

<http://cas.ensmp.fr/~petit/>

ÉCOLE CENTRALE PARIS  
ANNÉE SCOLAIRE 2005-2006



# Table des matières

<b>1</b>	<b>Théorie générale de la commande prédictive</b>	<b>1</b>
1.1	Principe . . . . .	1
1.2	Commande optimale . . . . .	1
1.3	Bouclage et stabilité . . . . .	2
1.3.1	Optimisation à horizon fini et bouclage . . . . .	3
1.3.2	Le problème des temps de calcul . . . . .	3
<b>2</b>	<b>Commande prédictive linéaire</b>	<b>4</b>
2.1	Commande prédictive en représentation discrète . . . . .	4
2.2	Calcul d'une cible atteignable . . . . .	5
2.3	Réécriture et résolution . . . . .	6
2.4	Implémentation . . . . .	7
2.5	Extensions possibles . . . . .	7
<b>3</b>	<b>Systèmes FIR</b>	<b>8</b>
3.1	Notations et définitions . . . . .	8
3.2	Prévisions . . . . .	9
3.2.1	Prévisions à partir de la réponse indicielle . . . . .	9
3.2.2	Prévisions à partir de la réponse à l'échelon . . . . .	10
<b>4</b>	<b>Applications</b>	<b>11</b>
4.1	Exemple 1 : réacteur exothermique . . . . .	11
4.1.1	Le problème des transitoires avec PID ou MPC? . . . . .	12
4.2	Exemple 2 : colonne à distiller binaire . . . . .	14
<b>5</b>	<b>Conclusion</b>	<b>16</b>
<b>A</b>	<b>Fonctions de Lyapunov</b>	<b>17</b>
<b>B</b>	<b>Méthode de résolution de la commande optimale</b>	<b>17</b>



## Introduction

La philosophie de la commande prédictive (MPC pour Model Predictive Control) se résume à “utiliser le modèle pour prédire le comportement du système et choisir la décision la meilleure au sens d’un certain coût tout en respectant les contraintes”.

Les termes de coût et de contraintes sont familiers à ceux qui connaissent l’optimisation. Dans notre cas, il s’agit d’un cas particulier s’adressant à l’optimisation des systèmes dynamiques : la commande optimale. L’idée la commande prédictive se trouve déjà entre les lignes de l’ouvrage fondateur de la commande optimale de Bellman en 1957. L’étude de la stabilité d’une telle loi de commande en boucle fermée remonte quand à elle à Kalman en 1960 qui note que “l’optimalité n’implique pas la stabilité”.

Historiquement, cette idée ne sera mise en œuvre industriellement qu’avec Richalet en 1976 dans le logiciel IDCOM (identification, commande). Le formalisme choisi permet de considérer les systèmes linéaires comme des filtres à réponse impulsionnelle finie (FIR pour Finite Impulse Response). Les coûts considérés sont quadratiques, la partie estimation repose sur une approche moindres carrés. On peut spécifier des contraintes affines sur les entrées et les sorties.

En 1980 apparaît DMC (Dynamic Matrix Control) de Cutler et Ramaker. DMC reprend de nombreuses idées de IDCOM, les systèmes sont représentés par leur réponses à l’échelon.

En 1988 il est possible grâce à SMOC (Shell Multivariable Optimizing Control) de considérer les systèmes sous forme d’état. La partie estimation est assurée par un filtre de Kalman. Outre ces outils plus modernes de l’automatique, SMOC intègre la notion de contraintes dures et molles et permet de les ordonner. C’est en pratique une possibilité intéressante.

Le but du présent document est de présenter les éléments constitutifs fondamentaux d’un outil de commande prédictive ainsi que son utilisation pratique.

## 1 Théorie générale de la commande prédictive

### 1.1 Principe

La commande prédictive c’est la résolution répétée à chaque pas de temps (noté ici  $\delta$  et spécifié par l’utilisateur) d’un problème de commande optimale : “comment aller de l’état actuel à un objectif de manière optimale en satisfaisant des contraintes”.

Pour cela il faut connaître à chaque itération l’état du système et utiliser un outil de résolution numérique.

### 1.2 Commande optimale

Considérons dans un premier temps un problème dépourvu de contraintes. Les équations représentant le système sont en toute généralité

$$(1) \quad \begin{cases} \dot{x} = f(x, u) \\ x(0) = x_0 \\ t \in [0, +\infty[ \end{cases}$$

où  $f$  est une fonction Lipchitz (hypothèse générale qui permet d’assurer l’existence des trajectoires). La solution générale de ces équations en fonction de  $u(\cdot)$  est

$$x^u(t, x_0) = x_0 + \int_0^t f(x^u(\tau, x_0), u(\tau)) d\tau.$$

Considérons que le coût à optimiser est

$$J_\infty(u(\cdot), x_0) = \int_0^\infty q(x^u(\tau, x_0), u(\tau)) d\tau$$

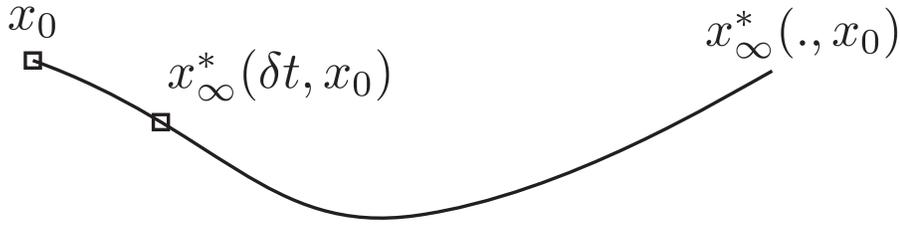


FIG. 1 – Trajectoire optimale

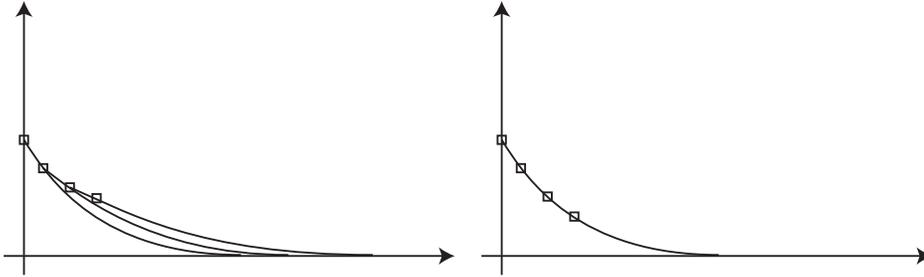


FIG. 2 – Application du principe d'optimalité de Bellman.

où

$$(2) \quad \begin{cases} q(0, 0) = 0, q \in C^2 \\ q(x, u) \geq c_q(\|x\|^2 + \|u\|^2), c_q > 0 \\ u \mapsto q(x, u) \text{ convexe pour tout } x. \end{cases}$$

Notons le coût optimal atteint  $J_\infty^*(x_0) = \min_{u(\cdot)} J_\infty(u(\cdot), x_0)$ .

### 1.3 Bouclage et stabilité

En chaque point  $x_0$  il faut résoudre le problème précédent de commande optimale. Si en tout point  $x_0$  on appliquait au système  $u^*(t, x_0, \infty)$  (la commande est donc un bouclage) alors on aurait stabilité du système. Nous allons prouver ce résultat.

Raisonnons de manière infinitésimale (voir Figure 1). Partant de  $x_0$  à l'instant 0 appliquons pendant  $\delta t \ll 1$  la commande  $u^*(t, x_0, \infty)$ . Nous parcourons une portion de la courbe  $x_\infty^*(., x_0)$  et parvenons alors en  $(\delta t, x_\infty^*(\delta t, x_0))$ . A partir de ce point on décide de résoudre à nouveau le problème de commande optimale qui s'écrit maintenant  $\min_{u(\cdot)} J_\infty(u(\cdot), x_\infty(\delta t, x_0))$ . Que trouve-t-on comme solution ?

Le principe d'optimalité de Bellman nous indique la solution. Il stipule en effet que *“une suite de décisions est optimale si, quels que soient l'état et l'instant considéré sur la trajectoire associée, les décisions ultérieures sont optimales pour le sous-problème ayant cet état et cet instant comme conditions initiales”*.

Autrement dit, on trouve comme solution la fin de la trajectoire  $x_\infty^*(., x_0)$ . On a l'égalité  $x_\infty^*(t, x_0) = x_\infty^*(t - \delta t, x_\infty^*(\delta t, x_0))$  pour tout  $t \in [\delta t, +\infty[$ . Il est donc possible de décomposer le coût optimal comme

$$J_\infty^*(x_0) = J_\infty^*(x_\infty^*(\delta t, x_0)) + \int_0^{\delta t} q(x_\infty^*(\tau, x_0), u_\infty^*(\tau, x_0)) d\tau$$

en passant à la limite  $\delta t \rightarrow 0$  on trouve

$$\frac{d}{dx} J_{\infty}^* \cdot f(x_0, u_{\infty}^*(0, x_0)) = -q(x_0, u_{\infty}^*(0, x_0))$$

soit

$$\frac{dJ_{\infty}^*}{dt} = -q(x_0, u_{\infty}^*(0, x_0)) < 0.$$

Nous allons maintenant utiliser  $J_{\infty}^*$  comme fonction de Lyapunov du système. En effet la fonction  $x \mapsto J_{\infty}^*(x) > 0$  sauf en 0 où elle vaut 0 et sa dérivée temporelle  $\frac{dJ_{\infty}^*}{dt} \leq 0$ . Cette propriété sert à montrer que c'est donc une fonction de Lyapunov (voir [6]). Par le théorème de Lyapunov, 0 est un point stable du système (voir Annexe pour des compléments).

Autrement dit, l'utilisation comme bouclage de la commande minimisant  $J_{\infty}$  garantit la stabilité. C'est un intéressant résultat théorique mais qui en pratique n'est pas utilisable, car on ne sait pas en général calculer la commande minimisant  $J_{\infty}$ .

Toutes les méthodes numériques habituelles (voir références en annexe) reposent sur une représentation approximante du problème combinées à des méthodes d'intégration numérique type Runge-Kutta pour l'évaluation des intégrales qui permettent d'évaluer uniquement des intégrales sur un intervalle de longueur finie.

Dans ce qui suit nous allons désormais nous intéresser à des problèmes qu'on sait traiter numériquement. Nous sommes donc contraints d'approximer  $J_{\infty}$  par

$$J(u(\cdot), x_0, T) = \int_0^T q(x^u(\tau, x_0), u(\tau)) d\tau + V(x^u(T, x_0)).$$

Notons alors  $J^*(x_0, T) = \min_{u(\cdot)} J(u(\cdot), x_0, T)$  et  $u^*(t, x_0, T) = \arg \min_{u(\cdot)} J(u(\cdot), x_0, T)$ , l'optimum de  $J(u(\cdot), x_0, T)$  et la commande le réalisant.

### 1.3.1 Optimisation à horizon fini et bouclage

On définit la commande prédictive comme l'algorithme suivant

1. résoudre  $\min J(u(\cdot), x_0, T)$ , dont la solution est  $u^*(\cdot, x_0, T)$
2. appliquer  $u^*(\tau, x_0, T)$  pour  $\tau \in [0, \delta]$ ,  $0 \leq \delta \leq T$
3. répéter en remplaçant  $x_0$  par  $x(\delta)$ .

Malheureusement cette fois ci le principe d'optimalité de Bellman ne s'applique pas. En effet entre deux itérations les problèmes à résoudre ne sont pas exactement comparables. Ainsi à la première itérations on minimise  $\int_0^T$  alors qu'à la deuxième c'est  $\int_{\delta}^{T+\delta}$ . Le deuxième problème n'est pas un "sous-problème" du premier. C'était le cas en horizon infini puisque  $\int_{\delta}^{T+\delta}$  était bien un sous problème de  $\int_0^{\infty}$ .

En réalité, il est possible par un choix adéquat du coût terminal  $V$  de rendre stable la commande prédictive. Par exemple la condition  $\dot{V} + q \leq 0$  permet de montrer que  $V$  est une fonction de Lyapunov du système.

### 1.3.2 Le problème des temps de calcul

Un autre problème qui n'est pas traité par la théorie de la commande prédictive mais qui a une importance pratique est celui de l'influence des temps de calculs.

A un instant donné, on calcule la commande optimale partant du point actuel, ou plutôt on commence à calculer cette commande. En réalité elle ne sera disponible qu'une fois le calcul terminé. Entre temps on pourra utiliser l'optimum précédent. Mais cette commande n'est pas adaptée au problème. Si les temps de calculs (qui sont en outre variables) ne sont pas négligeables par rapport à  $\delta$  temps d'application, on risque d'engendrer une instabilité dans le schéma de commande prédictive. Cette situation est résumée sur la figure 3.

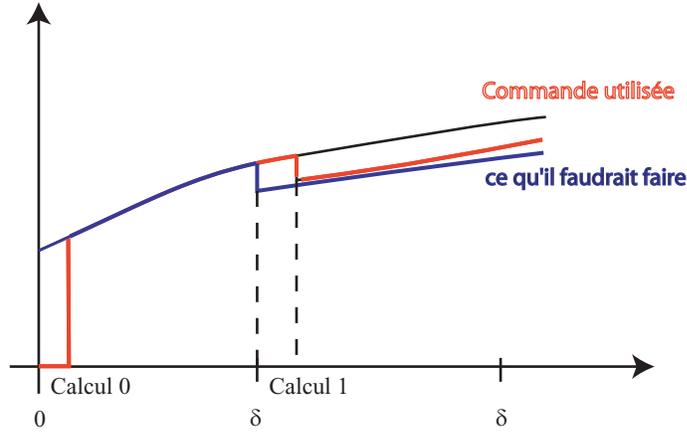


FIG. 3 – Le problème des temps de calculs.

Il faut donc utiliser pour  $\delta$  un temps qui soit grand devant les temps de calculs et court devant les constantes de temps du système. Ces deux contraintes se contredisent dans le cas des systèmes rapides, ceci explique le grand succès de la commande prédictive dans le domaine du génie des procédés et son succès plus que modeste dans le domaine de l'aéronautique (ceci est en train de changer avec l'arrivée de calculateurs embarqués réellement puissants).

## 2 Commande prédictive linéaire

Un cas particulier très important en pratique est celui des systèmes linéaires avec coût quadratique. Ces systèmes sont fréquents dans les applications car ils représentent le comportement au voisinage d'un point de fonctionnement donné d'un système non-linéaire (voir cours d'Automatique de Tronc Commun).

On peut représenter un système linéaire à commander sous les deux formes suivantes

$$(3) \quad \begin{cases} \dot{x} = Ax + Bu \\ y = Cx \end{cases}$$

dite "forme d'état" où sous la forme discrète suivante qui aura dans la suite notre faveur car elle se prête bien au traitement informatique temps réel de la commande prédictive

$$(4) \quad \begin{cases} x_{j+1} = Ax_j + Bu_j \\ y_j = Cx_j. \end{cases}$$

En outre on considère des contraintes affines :

$$Du_j \leq d, Hx_j \leq h$$

où  $D$  et  $H$  sont des matrices à coefficients constants et  $d$  et  $h$  des vecteurs à composantes positives.

### 2.1 Commande prédictive en représentation discrète

On peut définir la "commande prédictive linéaire" (MPC par opposition à la GMPC pour Generalized MPC) comme la loi de rétro-action  $u_j = \rho(x_j)$  qui minimize

$$\phi = \frac{1}{2} \sum_{j=0}^{+\infty} ((y_j - \bar{y})^T Q (y_j - \bar{y}) + (u_j - \bar{u})^T R (u_j - \bar{u}) + \Delta u_j^T S \Delta u_j)$$

avec  $\Delta u_j = u_j - u_{j-1}$ ,  $Q, R, S$  matrices symétriques définies positives.

On appellera  $(\bar{y}, \bar{u})$  les *cibles économiques*.  $\bar{y}$  représente un régime de fonctionnement vu de l'extérieur (sortie) idéal, alors que  $\bar{u}$  est une des recettes pour y arriver (il y en a en général plusieurs). En toute généralité,  $\bar{y}$  et  $\bar{u}$  peuvent varier dans le temps. Par exemple dans un procédé type batch de polymérisation,  $\bar{y}$  pourra représenter un profil thermique dans le temps, qui conditionnera la formation des chaînes et confèrera au polymère produit certaines propriétés rhéologiques.

## 2.2 Calcul d'une cible atteignable

La première chose à faire est de traduire l'objectif économique en un point de fonctionnement  $(x_s, y_s, u_s)$  où  $x_s$  est un point stationnaire. On veut décider autour de quel point stationnaire on va travailler. Ainsi on saura comment évolue le système via son équation d'état  $x_{j+1} = Ax_j + Bu_j$ , ce qu'on ne pourrait pas faire via la seule équation de sortie  $y_j = Cx_j$  qui n'est qu'une simple projection. La donnée de  $(y_s, u_s)$  est insuffisante. Idéalement  $y_s = \bar{y}$ ,  $u_s = \bar{u}$  mais ce n'est pas toujours possible.

Pour trouver un tel point nous allons utiliser une méthode de pénalisation exacte en minimisant le coût suivant

$$\min_{(x_s, u_s, \eta)} \frac{1}{2} \eta^T Q_s \eta + (u_s - \bar{u}) R_s (u_s - \bar{u}) + q_s^T \eta$$

où  $Q_s, R_s$  sont des matrices symétriques définies positives, sous les contraintes

$$x_s = Ax_s + Bu_s, -\eta \leq \bar{y} - Cx_s \leq \eta, \eta \geq 0, Du_s \leq d, Hx_s \leq h$$

où  $\eta$  est une variable de relaxation.

### La méthode de pénalisation exacte

L'intérêt de la méthode de pénalisation exacte est que s'il existe une solution de  $Cx_s = \bar{y}$  (qui est bien l'équation qu'on a relaxée en introduisant  $\eta$ ) alors, pour  $q_s$  suffisamment grand ce sera effectivement la solution du problème de minimisation précédent.

Plus précisément, la théorie traite le problème suivant

$$\min f_0(x), x \in \mathbb{R}^n, \text{ sous les contraintes } i = 1 \dots q : f_i(x) \leq 0, i = q + 1 \dots m : f_i(x) = 0.$$

Les contraintes définissent un certain sous-ensemble de  $\mathbb{R}^n$  noté  $\Omega$ . On introduit alors  $P_\alpha(x) = f_0(x) + \alpha \beta(x)$  où  $\beta$  est en quelque sorte une "fonction indicatrice",

$$\begin{aligned} \beta : \mathbb{R}^n &\rightarrow \mathbb{R} \\ \beta(x) &\geq 0, \forall x \in \mathbb{R}^n \\ \beta(x) &= 0 \text{ équivaut à } x \in \Omega. \end{aligned}$$

Les deux exemples suivants de telles fonctions sont intéressants :

$$\begin{aligned} \beta_1(x) &= \sum_{i=1}^q \max(0, f_i(x))^2 + \sum_{i=q+1}^m f_i(x)^2 \\ \beta_2(x) &= \sum_{i=1}^q \max(0, f_i(x)) + \sum_{i=q+1}^m |f_i(x)|. \end{aligned}$$

Considérons ensuite une suite strictement croissante  $(\alpha_i) \in \mathbb{R}^+$ ,  $\alpha_i \rightarrow +\infty$ . Notons  $(x_k)$  la suite des solutions des problèmes de  $\min_{x \in \mathbb{R}^n} P_{\alpha_k}$ . Cette suite a les propriétés suivantes

$$\begin{aligned} P_{\alpha_k}(x_k) &\leq P_{\alpha_{k+1}}(x_{k+1}) \\ \beta(x_k) &\geq \beta(x_{k+1}) \\ f(x_k) &\leq f(x_{k+1}). \end{aligned}$$

Autrement dit on augmente la satisfaction des contraintes alors qu'on augmente le coût du problème contraint.

Le résultat suivant donne alors une méthode pour calculer une solution du problème contraint

**Proposition.** *Supposons  $\arg \min(P_{\alpha_k}(x)) \neq \emptyset$  et que  $\Omega \neq \emptyset$  alors les points d'accumulations de  $(x_k)$  sont les solutions de  $P$ .*

Fort de cette méthode, on a déterminé un point stationnaire de fonctionnement  $(x_s, u_s)$  dont la projection  $Cx_s$  est proche de (et même souvent égale à) l'objectif économique  $\bar{y}$ . Cette technique de pénalisation exacte est souvent rencontrée en pratique pour traiter des contraintes non-linéaires générales dont on ne sait pas si elles possèdent un point admissible.

## 2.3 Réécriture et résolution

Plaçons nous maintenant en écart par rapport à ce point de fonctionnement que nous venons de déterminer. Posons pour cela

$$\begin{aligned} w_j &= x_j - x_s \\ v_j &= u_j - u_s \\ z_j &= y_j - Cx_s. \end{aligned}$$

Ces quantités vérifient  $w_{k+1} = Aw_k + Bv_k$ ,  $z_k = Cw_k$ . Nous cherchons maintenant à résoudre le problème suivant

$$\min_{w_k, v_k} \phi(x_j) = \sum_{k=0}^{+\infty} (w_k^T Q z_k + v_k^T R v_k + \Delta v_k^T S \Delta v_k)$$

sous les contraintes

$$\begin{aligned} w_0 &= x_j - x_s \\ w_{k+1} &= Aw_k + Bv_k \\ Dv_k &\leq d + Du_s \\ Hw_k &\leq h + Hx_s. \end{aligned}$$

Ceci n'est pas un problème LQR (linear quadratic regulator) classique en raison de la présence de contraintes inégalités. On ne sait pas en trouver une solution analytique comme on sait le faire par une équation de Riccati en l'absence de telles contraintes (voir Annexe).

En outre ce problème possède un nombre infini de variables  $w_0, w_1, \dots$ ,  $v_0, v_1, \dots$  par rapport auxquelles il nous faut optimiser. En vue de la résolution numérique qui va suivre il nous faut donc approximer ce problème par un problème en horizon fini avec coût terminal, avec un nombre fini d'inconnues  $w_0, \dots, w_{N-1}$ ,  $v_0, \dots, v_{N-1}$ .

$$\min_{w_k, v_k} \bar{\phi}(x_j) = \sum_{k=0}^{N-1} (w_k^T Q z_k + v_k^T R v_k + \Delta v_k^T S \Delta v_k) + w_N^T \Pi w_N.$$

Le problème <sup>1</sup>

$$\min_{w_k, v_k} \bar{\phi}(x_j) = \sum_{k=0}^{N-1} (w_k^T Q z_k + v_k^T R v_k + \Delta v_k^T S \Delta v_k) + w_N^T \Pi w_N$$

<sup>1</sup>On pourra en pratique utiliser comme matrice de coût terminal  $\Pi$  la valeur initiale de la solution de l'équation de Riccati discrète correspondant au problème sans contraintes sur  $[N-1, \infty[$ . En effet sous l'hypothèse que les contraintes ne sont pas actives sur l'intervalle  $[N-1, \infty[$  on sait que la queue de la série vaut exactement  $w_N^T \Pi w_N$ . Ce sera donc une approximation intéressante pour notre cas.

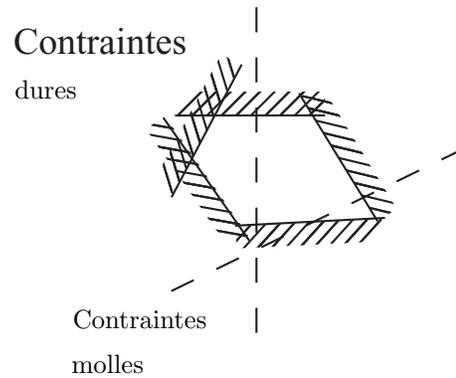


FIG. 4 – Contraintes dures et molles

sous les contraintes

$$\begin{aligned} w_0 &= x_j - x_s \\ w_{k+1} &= Aw_k + Bv_k \\ Dv_k &\leq d + Du_s \\ Hw_k &\leq h + Hx_s \end{aligned}$$

est un problème à coût quadratique sous contraintes affines. C'est un problème de *programmation quadratique* de la forme

$$\min_x x^T M_1 x + M_2^T x$$

sous la contrainte  $M_3^T x \leq M_4$  pour lequel de nombreux algorithmes de résolution sont disponibles.

Les contraintes affines définissent un polytope qui peut être vide. Dans ce cas le problème n'a pas de solution. Si le polytope n'est pas vide, on a unicité de la solution en raison de la convexité du coût et des contraintes.

En pratique on attend de l'algorithme de commande prédictive qu'il donne toujours une réponse, quitte à ce qu'elle viole certaines contraintes. A cet effet on définit des priorités entre les contraintes :

- en premier lieu des *contraintes dures*, dont on s'assurera au préalable qu'elles admettent toujours des points les vérifiant.
- ensuite des *contraintes molles* qu'il est intéressant de vérifier exactement. Ce sont des contraintes qu'on relaxera par la méthode précédente de pénalisation exacte.

En conclusion, les contraintes dures seront toujours satisfaites par la solution fournie par l'algorithme de commande prédictive alors que les contraintes molles seront satisfaites si c'est possible.

## 2.4 Implémentation

En résumé un algorithme de commande prédictive a besoin à chaque itération de

- l'état du système (pour l'écriture de la première contrainte) : qui lui sera fourni par un estimateur calculant une valeur approchée de  $x_s$  à partir des mesures
  - une traduction des objectifs économiques en point de fonctionnement
  - un ordonnancement des contraintes permettant de toujours obtenir une solution acceptable
- Une implémentation réaliste d'une MPC est donc telle que représentée sur la figure 5.

## 2.5 Extensions possibles

Aujourd'hui on envisage d'utiliser de plus en plus le modèle de connaissance (cad issu des équations de la physique) dans le schéma de commande prédictive.

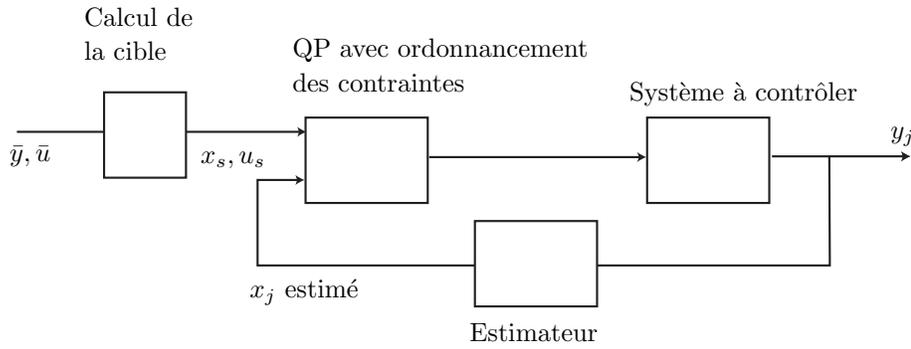


FIG. 5 – Une implémentation de la MPC



FIG. 6 –

- passage d'un point à un autre possédant des approximations linéaires (modèles) très différentes. On pourra pour cela mettre à jour le modèle grâce à un estimateur.
- utilisation du modèle de connaissance nonlinéaire pour l'estimateur
- utilisation d'un algorithme de résolution NLP (nonlinear programming) au lieu d'un QP. Le principal problème réside dans la non-convexité du problème d'optimisation. En pratique on risque de ne pas trouver l'optimum global s'il existe mais un minimum local.

### 3 Systèmes à réponse impulsionnelle finie FIR

#### 3.1 Notations et définitions

On représente un système par le schéma suivant :

Dans un premier on se restreint au cas des systèmes à une seule entrée et à une seule sortie. On échantillonne le signal de sortie  $y(t)$  à intervalles réguliers en notant  $T$  la période d'échantillonnage. On note la suite des échantillons successifs de la sortie

$$\{y(0), y(1), \dots, y(k), \dots\}$$

et de même pour la commande

$$\{u(0), u(1), \dots, u(k), \dots\}.$$

Le signal de commande est défini par le bloqueur d'ordre zéro

$$u(t) = u(k), kT < t \leq (k+1)T$$

On s'intéressera dans la suite à deux signaux d'entrée particuliers :

- l'impulsion

$$\bar{v}(0) = \{1, 0, \dots, 0, \dots\}$$

- l'échelon

$$\bar{v}(0) = \{1, 1, \dots, 1, \dots\}.$$

### Réponse impulsionnelle

Soit  $\bar{y}(0) = \{h_0, h_1, \dots, h_n, h_{n+1}, \dots\}$  la réponse du système à l'impulsion. On appelle système à réponse impulsionnelle finie (FIR) un système tel que  $h_0 = 0$  et  $h_k = 0$  pour  $k > n$  où  $n$  est un entier donné. On dit qu'un tel système s'établit à 0 après  $n$  échantillons. On note  $H_1 = [h_1, h_2, \dots, h_n]^T$  la matrice des coefficients du système, elle le définit entièrement par les deux principes suivants :

**Principe de décalage** L'entrée décalée  $\bar{u}(0) = \{0, 1, 0, \dots\}$  produit la sortie décalée  $\bar{y}(0) = \{0, 0, h_1, h_2, \dots\}$ .

**Principe de superposition** L'entrée  $\bar{u}(0) = \{u(0), u(1), u(2), \dots\}$  produit la sortie  $\bar{y}(0) = \{0, y(1), y(2), \dots\}$  telle que  $y(k) = \sum_{i=1}^n h_i u(k-i)$ . Ainsi pour calculer la valeur de la sortie  $y(k)$  il faut garder en mémoire les  $n$  valeurs de l'entrée  $u(k-1), \dots, u(k-n)$ .

Cette formule provient de la superposition  $\bar{u}(0) = u(0)\{1, 0, \dots, 0, \dots\} + u(1)\{0, 1, 0, \dots, 0, \dots\} + \dots$  qui fournit par linéarité  $\bar{y}(0) = u(0)\{0, h_1, \dots, h_n, h_{n+1}, \dots\} + u(1)\{0, 0, h_1, \dots, h_n, h_{n+1}, \dots\} + \dots$

### Réponse à un échelon

Considérons un système FIR. L'entrée  $\bar{v}(0) = \{1, 1, \dots, 1, \dots\}$  fournit la réponse

$$\bar{y}(0) = \{0, s_1, \dots, s_n, s_{n+1}, \dots\} = \{0, h_1, h_1 + h_2, h_1 + h_2 + h_3, \dots\}.$$

Il est intéressant en pratique de définir la réponse du système par ces coefficients  $S = [s_1, s_2, \dots, s_n]^T$ . En notant  $\Delta u(i) = u(i) - u(i-1)$  on peut écrire la réponse du système à l'échelon  $\text{bary}(0) = \{0, y(1), y(2), \dots\}$ , avec

$$y(k) = s_n u(k-n) + \sum_{i=1}^{n-1} s_i \Delta u(k-i).$$

On constate également que pour calculer  $y(k)$  il faut garder en mémoire les  $n$  dernières valeurs de l'entrée.

Enfin les relations de passage entre  $S$  et  $\xi H$  sont  $s_k = \sum_{i=1}^k h_i$ ,  $h_k = s_k - s_{k-1}$ .

### État d'un système FIR

On appelle l'état d'un système FIR la matrice

$$\bar{Y}(k) = [\bar{y}_0(k), \bar{y}_1(k), \dots, \bar{y}_{n-1}(k)]^T$$

où  $\bar{y}_i(k) = y(k+i)$  calculée pour  $u(k+j) = 0$  pour tout  $j \geq 0$ . En d'autres termes  $\bar{y}_i(k)$  est la sortie du système au temps  $k+i$  lorsque l'entrée vaut 0 à partir du temps  $k$ .

## 3.2 Prévisions

### 3.2.1 Prévisions à partir de la réponse indicielle

L'état du système représente la prévision de l'évolution future du système si les valeurs futures de l'entrée sont nulles. Il est de dimension finie  $n$  car le système s'établit naturellement à 0 après  $n$  échantillons. Si la commande future est non nulle, on peut prévoir les valeurs futures de la sortie par les formules qui vont suivre.

$$\begin{aligned} y(k+1) &= \bar{y}_1(k) + h_1 u(k) \\ y(k+2) &= \bar{y}_2(k) + h_1 u(k+1) + h_2 u(k) \\ &\dots \end{aligned}$$

En d'autres termes

$$\begin{bmatrix} y(k+1) \\ y(k+2) \\ \vdots \\ y(k+p) \end{bmatrix} = \begin{bmatrix} \tilde{y}_1(k) \\ y_2(k) \\ \vdots \\ y_p(k) \end{bmatrix} + \begin{bmatrix} h_1 \\ h_2 \\ \vdots \\ h_p \end{bmatrix} u(k) + \begin{bmatrix} 0 \\ h_1 \\ \vdots \\ h_{p-1} \end{bmatrix} u(k+1) + \dots + \begin{bmatrix} 0 \\ 0 \\ \vdots \\ h_1 \end{bmatrix} u(k+p-1)$$

où on voit l'état et la contribution des commandes futures.

Il est possible enfin de calculer de manière récursive les prévisions par la formule

$$\bar{Y}(k+1) = M^H \bar{Y}(k) + H u(k)$$

où  $M^H$  est la matrice nilpotente

$$M^H = \begin{pmatrix} 0 & 1 & & 0 \\ & \ddots & \ddots & \\ & & \ddots & 1 \\ 0 & & & 0 \end{pmatrix}.$$

### 3.2.2 Prévisions à partir de la réponse à l'échelon

Afin d'utiliser les grandeurs  $\Delta u(i)$  définie précédemment, on définit un autre état  $\tilde{Y}(k) = [\tilde{y}_0(k), \tilde{y}_1(k), \dots, \tilde{y}_{n-1}(k)]^T$  où  $\tilde{y}_i(k) = y(k+i)$  où  $\Delta u(k+j) = 0$  pour tout  $j \geq 0$ . En d'autres termes  $\tilde{y}_i(k)$  est la sortie du système au temps  $k+i$  lorsque l'entrée reste inchangée à partir de l'instant  $k$ .

Comme précédemment il est possible d'établir une formule de récurrence pour la prévision de cet état :

$$\tilde{Y}(k+1) = M^S \tilde{Y}(k) + S \Delta u(k)$$

où

$$M^S = \begin{pmatrix} 0 & 1 & & 0 \\ & \ddots & \ddots & \\ & & \ddots & 0 \\ 0 & & & 1 \end{pmatrix}.$$

Notons enfin qu'il est possible de généraliser les concepts d'état, de réponses impulsionnelles et à l'échelon, et des formules de prédiction au cas des systèmes ayant un nombre quelconque  $n_u$  d'entrées et  $n_y$  de sorties.

Ainsi on notera

$$S = [S_1^T, S_2^T, \dots, S_n^T]^T$$

la collection des matrices  $S_i = (s_{l,m,i})$  où l'élément  $s_{l,m,i}$  représente le coefficient  $i$  de la réponse de la sortie  $l$  à un échelon sur l'entrée  $m$ . L'état du système sera noté

$$\tilde{Y}(k) = [\tilde{y}_0(k), \tilde{y}_1(k), \dots, \tilde{y}_{n-1}(k)]^T$$

où chaque élément est de dimension  $n_y$ .

On établit alors la formule de récurrence

$$\tilde{Y}(k+1) = M \tilde{Y}(k) + S \Delta u(k)$$

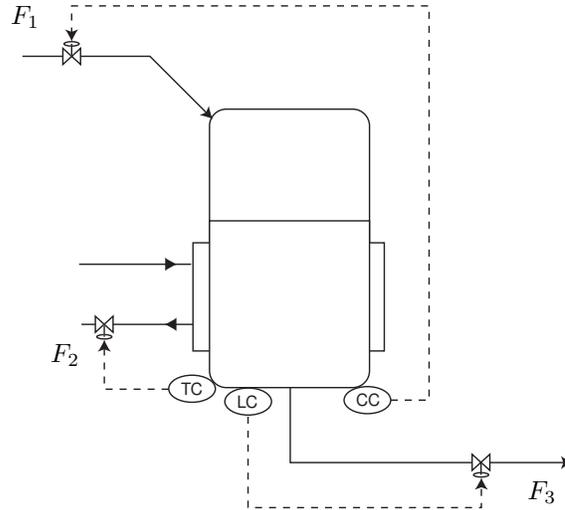


FIG. 7 – Réacteur exothermique avec commandes monovariables

où

$$M = \begin{pmatrix} 0 & I & 0 \\ & \ddots & \ddots \\ 0 & & 0 & I \\ & & & I \end{pmatrix}$$

est une matrice de taille  $(n_y n) \times (n_y n)$ .

## 4 Applications de la MPC en génie des procédés

Aujourd'hui on compte plus de 2000 applications de la MPC dans le monde. Parmi celles-ci seules 100 sont de nature non-linéaire (c'est à dire du type présenté en extension à la section 2.5). Le reste relève donc de la méthodologie présentée dans ce cours.

Nous allons considérer deux exemples caractéristiques d'applications de la commande prédictive sur des cas pratiques.

### 4.1 Exemple 1 : réacteur exothermique

Considérons un réacteur parfaitement agité où se produit la réaction exothermique  $A \rightarrow B$ . Le produit  $A$  est introduit en continu par la vanne  $F_1$ , l'exothermicité de la réaction est évacuée par les échanges avec la double enveloppe dont le débit de circulation est donné par la vanne  $F_2$ . Le mélange réactionnel est soutiré en continu par la vanne  $F_3$ . Les équations différentielles régissant l'évolution de ce système sont de la forme (équations bilans)

$$\begin{aligned} \frac{d}{dt}(VC_A) &= F_1 C_{A0} - F_3 C_A - k e^{-E_a/(RT)} C_A V \\ \rho C_p \frac{d}{dt}(VT) &= \rho(F_1 C_{P0} T_0 - F_3 C_p T) + A e^{-E_a/(RT)} C_A V (-\Delta H) - \alpha F_2 (T - T_c) \\ \frac{d}{dt}V &= F_1 - F_2. \end{aligned}$$

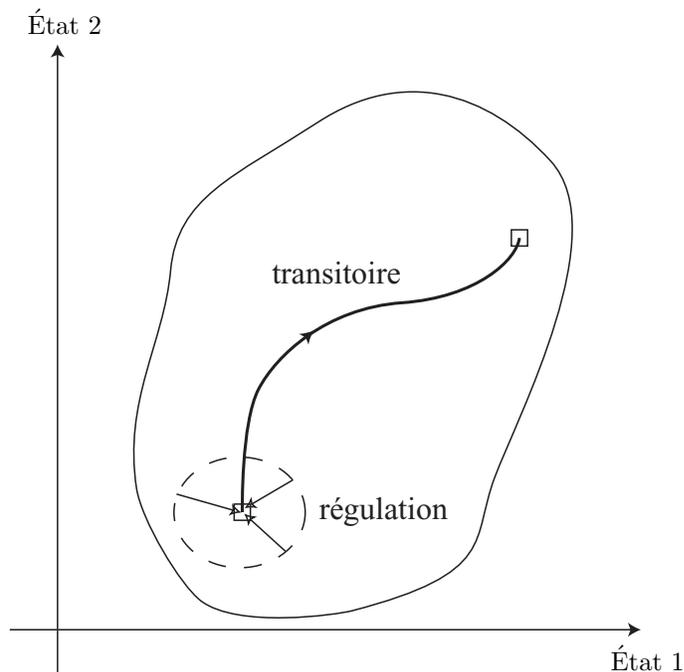


FIG. 8 – Le problème de la régulation et des transitoires

L'état du système est  $(C_A, T, V)$ , respectivement la concentration en produit  $A$ , la température et le volume du milieu réactionnel.  $k, C_p, C_{A0}, C_{P0}, T_c$  et  $E_a$  sont des constantes.  $C_P$  est supposée constante.

On souhaite réguler les trois grandeurs  $(C_A, T, V)$  en utilisant les trois vannes  $F_1, F_2, F_3$ . Dans le cadre de la commande monovariante, on a vu comment équiper cette unité de régulateur PID en asservissant  $C_A$  par la vanne  $F_1$ ,  $T$  par la vanne  $F_2$  et  $V$  par la vanne  $F_3$ . Un tel système de régulation fonctionne très bien.

En revanche il est malaisé de répondre aux questions suivantes :

- comment effectuer des changements de régime (autrement appelées des transitoires) de manière optimale ?
- comment gérer des contraintes sur les actionneurs ?

#### 4.1.1 Le problème des transitoires avec PID ou MPC ?

Faire passer le système d'un point à un autre n'est pas un problème facile quand les équations de la dynamiques sont non-linéaires. En présence de contraintes, la région accessible dans l'espace des états n'est pas forcément convexe. On peut se retrouver dans les différents cas de figure représentés ci-dessous pour lesquels les simples PID monovariants ont des performances variables.

Chaque PID tente de faire converger la grandeur qu'il est supposé asservir vers la consigne qu'on lui a assignée.

1. Dans ce cas les PID apportent une réponse satisfaisante. La commande prédictive n'apporte comme amélioration que l'optimalité du chemin parcouru. C'est un gain souvent marginal comparé à la différence de complexité de la commande prédictive par rapport aux simples PID.
2. En ajoutant un "override" sur les PID, c'est à dire un limiteur sur la valeur de sortie des PID (complété par un dispositif anti-windup), on obtient un comportement relativement

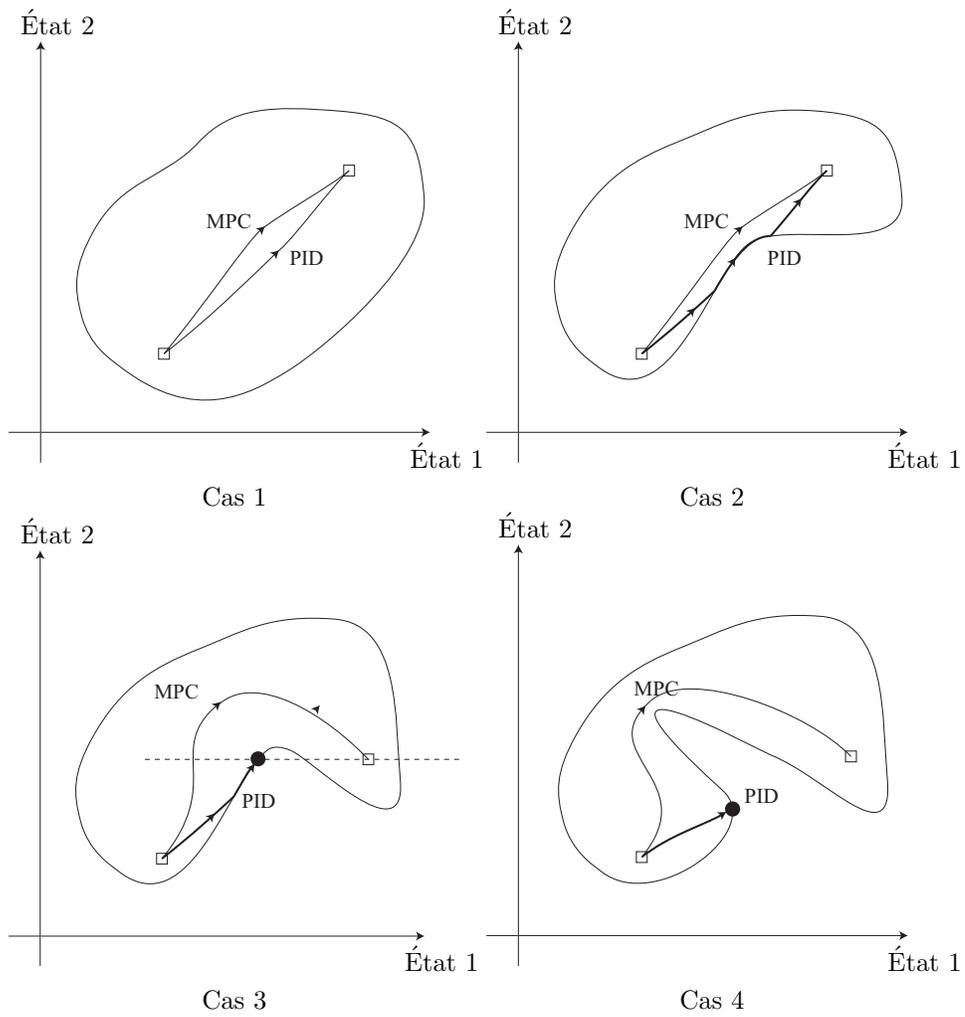


FIG. 9 – Comparaisons PID contre MPC

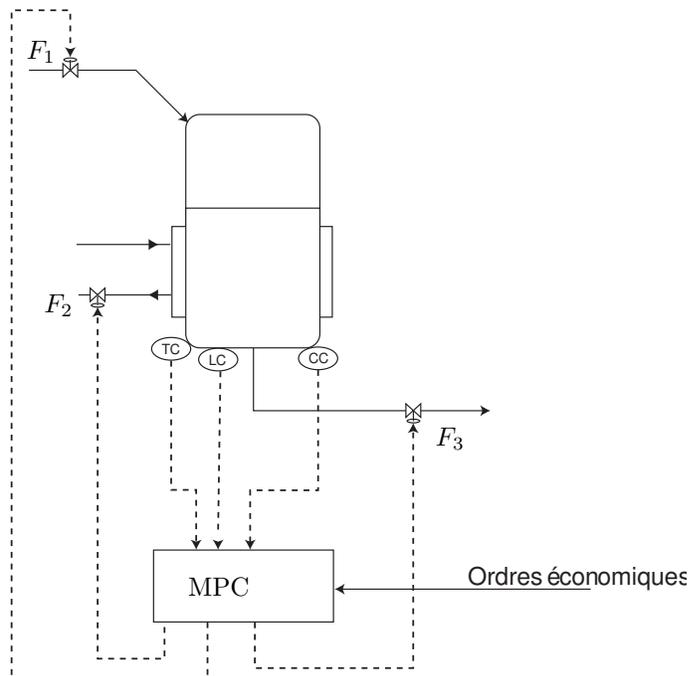


FIG. 10 – Réacteur exothermique avec MPC

satisfaisant tout en respectant les contraintes. En revanche il est probable que le chemin suivi est cette fois relativement coûteux comparé à l'optimum calculé par la MPC.

3. Cette fois il n'y a pas moyen de faire fonctionner les PID monovariables. Le PID sur l'état 2, empêche le système d'avancer au delà du point  $C$ . Les PID sont incapables de trouver le chemin calculé par le MPC.
4. Dans cette situation "extrême" les PID s'avèrent particulièrement inefficaces. En centralisant les décisions, c'est à dire en planifiant une trajectoire pour  $x_1$  et  $x_2$  de manière coordonnée, on trouve le bon chemin.

Notre réacteur chimique est un système dont l'état est de dimension 3. L'analyse graphique précédente est donc plus délicate mais reste valide dans son principe. Un régulateur efficace dans le cas de notre réacteur chimique centralisera les mesures de température, volume et concentration et calculera de manière centralisée les trois commandes  $F_1$ ,  $F_2$ ,  $F_3$ .

Il apparaît donc que la centralisation des décisions n'a que des avantages. Pourtant, certaines règles de sécurité indiquent qu'il n'est pas souhaitable de jouer sur tous ces deux degrés de liberté. Ainsi les erreurs de modèle, peuvent conduire à des méprises sur le respect des contraintes (essayer d'atteindre un volume négatif...). Le cas d'une perte de connexion entre le calculateur centralisé et ses capteurs, ou même d'un bug informatique, entraîne la défaillance de tous les régulateurs en même temps. Aussi préserve-t-on de simples mais efficaces contrôleurs PID monovariables sur les variables essentielles pour la sécurité : niveaux, certaines températures lorsqu'on s'approche des limites de l'emballement, etc... On peut aussi prendre certaines marges de sécurité sur les contraintes admissibles par la MPC.

## 4.2 Exemple 2 : colonne à distiller binaire

Considérons une colonne à distiller binaire. Étant donné un certain débit de charge  $F$  de pureté  $z_F$ , on veut réguler les quatre états  $V_D, x_D, V_B, x_B$ , respectivement le volume et la pureté en

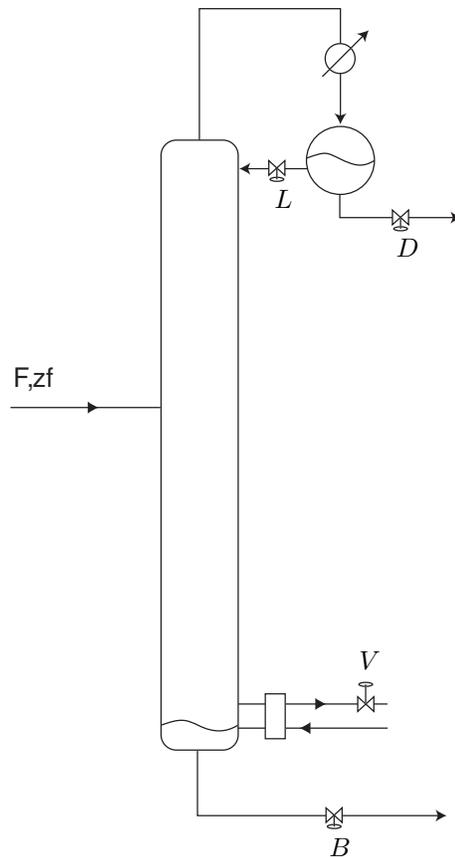


FIG. 11 – Colonne à distiller binaire

tête, le volume et la pureté en fond de la colonne. On dispose de quatre commande  $L, V, D, B$ , respectivement le débit de recycle, le rebouillage, le débit de distillat et de production en fond.

On peut choisir quatre stratégies de commande pour ce système à quatre entrées et quatre sorties en utilisant successivement de moins en moins de PID et de plus en plus de MPC.

1. le “tout PID”. Par exemple on pourra asservir  $x_D$  avec  $L$ ,  $x_B$  avec  $V$ ,  $V_B$  avec  $B$ ,  $V_D$  avec  $D$  (c’est la commande en  $(L, V)$ ). C’est une commande efficace tant qu’on est pas en haute pureté, on pourra la remplacer le cas échéant par une commande en ratio (voir cours PE).
2. On peut choisir de laisser les deux niveaux asservis par des PID agissant respectivement sur  $B$  pour  $V_B$  et sur  $D$  pour  $V_D$ . En revanche on laisse la MPC agir de manière concertée sur  $L$  et  $V$  pour réguler  $x_D$  et  $x_B$ . On est donc assuré que quelles que soient les décisions de la MPC, les niveaux seront régulés, ce qui est important pour éviter l’engorgement et les surpressions. On gagne en gestion de transitoires.
3. En reprenant la structure précédente On peut autoriser la MPC à mettre à jour les points de consignes des régulations de niveaux assurées par les PID. Les chemins calculés par la MPC seront encore plus optimaux dans le sens où la MPC dispose de degrés de liberté supplémentaires par rapport auxquels optimiser. En cas de défaillance de système MPC, les régulateurs de niveaux peuvent continuer à fonctionner en utilisant le dernier point de consigne calculé.
4. On peut utiliser le MPC sur les quatres entrées pour réguler les quatres sorties. C’est optimal au niveau des performances mais risqué au niveau sécurité. En outre il faut s’attendre à des

excursions importantes des niveaux si on ne les fait pas apparaître par des pondérations importantes dans le coût à optimiser (la MPC se réservant le droit de jouer comme bon lui semble sur ces degrés de libertés).

En pratique la solution retenue est souvent la solution 2 ou 3. La solution 1 est généralement considérée comme trop sous-optimale et est réservée aux colonnes dont la qualité de séparation n'est pas cruciale pour l'unité de production. La solution 4 n'est pas souhaitable car jugée trop extrême, même si c'est théoriquement l'optimum.

## 5 Conclusion

Les principales caractéristiques de la MPC sont

- Son caractère multi-variable
- Qu'elle fournit des réponses optimales au sens d'un coût précisé par l'utilisateur
- Qu'elle permet de gérer des contraintes

C'est une technique de commande utile en transitoire. Elle est déjà très utilisée dans l'industrie et reste un thème de recherche actif dont la théorisation est plus récente que les applications.

## Bibliographie

Pour les références historiques en particuliers sur les travaux de Bellman et l'étude de la stabilité par Kalman, on se reportera à [3] et à [8].

Pour des détails concernant la mise en oeuvre concertée de régulateurs PID avec la MPC, on pourra consulter [9].

Enfin pour des considérations sur les avantages et inconvénients de la logique centralisée de commande, on pourra lire [2].

Tous les détails sur l'implémentation en Matlab se trouvent dans [4].

## Références

- [1] F. Bonnans. The shooting algorithm for optimal control problems : A review of some theoretical and numerical aspects. Notes de cours, <http://www-rocq.inria.fr/sydoco/cours/tutorial.html>, INRIA, 2002.
- [2] H. Huang and J. B. Riggs. Comparison of PI and MPC for control of a gas recovery unit. *Journal of Process Control*, 12 :163–173, 2002.
- [3] D. Q. Mayne, J. B. Rawlings, C. V. Rao, and P. O. M. Scokaert. Constrained model predictive control : stability and optimality. *Automatica*, 36 :789–814, 2000.
- [4] M. Morari and N. L. Ricker. Model predictive toolbox. User's guide, <http://www.mathworks.com>, The Math Works Inc., 1998.
- [5] L. S. Pontryagin, V. G. Boltyanskii, R. V. Gamkrelidze, and E. F. Mishchenko. *The Mathematical Theory of Optimal Processes*. Interscience Publishers, 1962.
- [6] L. Praly. Fonctions de Lyapunov et stabilisation de point d'équilibre. Polycoché de cours, <http://cas.ensmp.fr/~praly/publications/synthese-lyapunov.pdf>, DEA ATS, 2002.
- [7] J. B. Rawlings. Tutorial overview of model predictive control. *IEEE Control Systems Magazine*, June :38–52, 2000.
- [8] J. Richalet. *Pratique de la commande prédictive*. Adersa, Hermès, 1993.
- [9] L. O. Santos, P. A. F. N. A. Alfonso, J. A. A. M. Castro, and N. M. C. Oliveira. On-line implementation of nonlinear mpc : an experimental case study. *Control Engineering Practice*, 9 :847–857, 2001.

- [10] H. J. Sussmann and J. C. Willems. 300 years of optimal control : from the brachystochrone to the maximum principle. *IEEE Control Systems Magazine*, pages 32–44, 1997. disponible sur <http://www.math.rutgers.edu/~sussmann>.
- [11] O. von Stryk and R. Bulirsch. Direct and indirect methods for trajectory optimization. *Annals of Operations Research*, 37 :357–373, 1992.

## A Fonctions de Lyapunov

On appelle fonction de Lyapunov pour le système  $\dot{x} = f(x, u)$  au voisinage d'un point d'équilibre  $(\bar{x}, \bar{u})$ , une fonction  $V$  définie au voisinage de ce point telle que

- $V(x) > 0$  sauf au point d'équilibre où elle est nulle
- $V$  est une fonction et sa dérivée sont continues et vérifient

$$\dot{V} \leq 0.$$

L'existence d'une fonction de Lyapunov autour d'un point d'équilibre permet directement d'établir la stabilité de ce point. En pratique on aimera que la fonction de Lyapunov possède des propriétés supplémentaires : on demandera qu'elle soit propre, c'est à dire que  $V(x) \rightarrow +\infty$  lorsque  $\|x\| \rightarrow \infty$ , et en outre que sa décroissance soit rapide, si possible exponentielle. On se reportera à [6] pour des énoncés complets et leurs utilisations générale en théorie du contrôle.

## B Méthode de résolution de la commande optimale

Le calcul effectif des trajectoires optimales d'un système nonlinéaire est un problème difficile. Il repose sur la résolution des équations d'Euler-Lagrange et du problème aux deux bouts associé.

Pour comprendre la méthode de résolution de tels problèmes, revenons pour l'instant aux problèmes d'optimisation sous contraintes en dimension finie (nombre fini de variables et de contraintes). L'idée de Lagrange est d'introduire des variables additionnelles au même nombre que les contraintes (ces variables sont appelées multiplicateurs de Lagrange). On adjoint au coût à minimiser les contraintes, on résoud ensuite le problème de la minimisation sans contraintes de ce coût adjoint et on en déduit une solution du problème contraint.

Étant donné le problème de la minimisation  $\min L(x, u)$ ,  $x \in \mathbb{R}^n$ ,  $u \in \mathbb{R}^m$ ,  $n \geq 1$ ,  $m \geq 1$ , sous la contrainte  $f(x, u) = 0$  où  $f : \mathbb{R}^{n+m} \rightarrow \mathbb{R}^n$  est telle que  $f_x(x, u)$  soit une matrice inversible, on définit le coût adjoint par

$$H(x, u, \lambda) = L(x, u) + \lambda^T f(x, u)$$

avec  $\lambda \in \mathbb{R}^n$ . En résolvant les conditions d'extrémalités  $H_x = 0$ ,  $H_u = 0$ ,  $H_\lambda = 0$ , on obtient un extrémum pour la fonction  $L(x, u)$  sous la contrainte  $f(x, u) = 0$ . Les équations à résoudre prennent la forme

$$\begin{aligned} L_x(x, u) + \lambda^T f_x(x, u) &= 0, \quad (n \text{ équations}) \\ L_u(x, u) + \lambda^T f_u(x, u) &= 0, \quad (m \text{ équations}) \\ f(x, u) &= 0, \quad (n \text{ équations}) \end{aligned}$$

On se trouve bien avec  $2n + m$  équations portant sur  $2n + m$  inconnues.

En commande optimale, l'équation de la dynamique  $\dot{x} = f(x, u)$  doit être vérifiée en tout point de l'intervalle  $[t_0, t_f]$ . On peut l'interpréter comme une "infinité" de contraintes ponctuelles. On utilise alors une "infinité" de multiplicateurs de Lagrange représentés par la fonction  $\lambda : t \in [T_0, t_f] \mapsto \mathbb{R}^n$ .

Les conditions d'extrémalité sont alors issues du calcul des variations. Dans le cas (classique en pratique) du problème de la minimisation de (on trouvera tous les détails dans [7])

$$J(x, u) = \phi(x(t_f), t_f) + \int_{t_0}^{t_f} L(x(t), u(t)) dt$$

sous les contraintes  $\dot{x} = f(x, u)$ ,  $x(t_0)$  étant donné, les conditions d'extrémalité prennent la forme suivante

$$\begin{aligned} \dot{\lambda}^T &= -L_x(x, u) - \lambda^T f_x(x, u), \quad (n \text{ équations différentielles}) \\ \lambda^T(t_f) &= \frac{\partial \phi(x(t_f), t_f)}{\partial x(t_f)} \quad \text{condition finales pour } \lambda \\ 0 &= L_u(x, u) + \lambda^T f_u(x, u), \quad (m \text{ équations}) \\ \dot{x} &= f(x, u) \quad (n \text{ équations différentielles}) \\ x(t_0) &\text{ donné : conditions initiales données pour } x \end{aligned}$$

La résolution pratique de ces équations consiste en la résolution d'un système d'équations différentielles dit "aux deux bouts", c'est à dire dont les conditions initiales sont données en partie en  $t_0$  (pour  $x$ ) et en  $t_f$  (pour  $\lambda$ ). On pourra à ce sujet consulter ([11, 1]) contenant des développements quant aux techniques numériques appropriées.

Dans la pratique de la commande optimale, on est souvent amenés à imposer des contraintes sur la commande. Ceci correspond aux limitations physiques des actionneurs dont on doit tenir compte, typiquement  $u_{min} \leq u(t) \leq u_{max}$ , pour  $t \in [t_0, t_f]$ , où  $u_{min}$  et  $u_{max}$  sont deux constantes. La théorie du maximum de Pontryagine, voir [5, 10], amène les conditions d'extrémalité dans ce cas comportant des contraintes inégalités (qu'on ne peut se contenter de simplement adjoindre comme précédemment pour les contraintes égalités).