

**REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTRE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA  
RECHERCHE SCIENTIFIQUE**

**UNIVERSITE CHEIKH LARBI TEBESSI  
INSTITUT DES SCIENCES EXACTES ET TECHNOLOGIE  
DEPARTEMENT DE GENIE ELECTRIQUE**

**MEMOIRE**  
présenté en vue de l'obtention de diplôme de  
**MAGISTER**  
en électronique option  
Contrôle et automatique

**THEME**

**APPLICATION DES LOIS DE COMMANDE SUR UN  
PENDULE INVERSE**

**Soutenu par :**

**AIZI EL-HADJ**

**Rapporteur : Mr N.GUERFI**

**Maître de conférences**

**U.TEBESSA**

**Devant le jury**

**Président : Mr N.Doghmane**

**Professeur**

**U.ANNABA**

**Examineurs : Mr M.Mameri**

**Phd**

**U.TEBESSA**

**Invité : Mr A.Gattal**

**C.C**

**U.TEBESSA**



# **Remerciments**

*Je tiens à remercier tous qui m'ont aidé de près ou de loin à élaborer ce modeste travail*

*Principalement mon encadreur  
Mr. N.GUERFI*

*Ainsi que tous les enseignants de cycle de maîtrise  
et tous les enseignant au département de  
l'électronique*

**AIZI EL-HADJ**





# *Dédicace*

*Je dédie ce travail à mes parents qui méritent le  
meilleur des respects*

*A mes frères, et à mes soeurs*

*A toute la famille sans exception*

*A mon ami et mon frère Daouadi bouzid*

*A tous mes amis*

***AIZI EL-HADJ***



# SOMMAIRE

Introduction générale.....	1
----------------------------	---

## Chapitre I: La commande prédictive généralisée sans contraintes

I.1 Introduction.....	3
I.2 Stratégie de la commande prédictive.....	4
I.2.1 Principe général.....	5
I.2.2 Modélisation du système.....	5
I.3 Commande prédictive généralisée.....	6
I.3.1 Mise en équation de système.....	7
I.3.2 Prédicteur optimal.....	8
I.3.3 Fonction de coût quadratique.....	8
I.3.4 Prédicteur optimal sous forme matricielle.....	9
I.3.5 Minimisation de la fonction de coût.....	10
I.3.6 Structure <i>RST</i> du régulateur.....	10
I.4 Choix de paramètres de réglage.....	11
I.4.1 Choix de l'horizon minimal de prédiction $N_1$ .....	11
I.4.2 Choix de l'horizon maximal de prédiction $N_2$ .....	12
I.4.3 Choix de l'horizon de prédiction sur la commande $N_u$ .....	12
I.4.4 Choix du facteur de pondération de la commande $\lambda$ .....	12
I.5 Exemple d'application.....	13

## Chapitre II: La commande prédictive généralisée avec contraintes

II.1 Introduction.....	14
II.2 Types de contraintes.....	14
II.2.1 contraintes sur l'amplitude du signal de commande.....	15
II.2.2 Contraintes sur l'incrément de commande.....	15
II.2.3 Contraintes sur la sortie.....	15
II.2.4 Contraintes sur les oscillations permises dans la sortie du système.....	16
II.2.5 Contrainte pour éviter des comportements de phase non minimale.....	16
II.3 La programmation de GPC avec contraintes.....	17
II.3.1 La programmation Dynamique.....	17
II.3.1.1 Formulation du problème.....	17
II.3.1.2 Equation récurrente d'optimalité.....	17
II.3.1.3 Algorithme de la Programmation Dynamique.....	18
II.3.2 La programmation quadratique.....	18
II.3.2.1 Introduction.....	18
II.3.2.2 Formulation du problème.....	18
II.3.2.3 Résolution du problème.....	20
II.4 Autres méthodes d'optimisation.....	20
II.4.1 La méthode de gradient conjugué.....	20
II.4.1.1 Le cas d'une fonction quadratique.....	20
II.4.2 La méthode de FIBONACCI.....	21
II.4.2.1 Le cas monovarié.....	21
II.4.2.2 Le cas multivarié.....	22
II.4.3 La méthode de DICHOTOMIE.....	23

II.4.3.1 Le cas monovariable.....	23
II.4.3.2 Le cas multivariable.....	23
II.5 résultat de simulation.....	24

### Chapitre III: L'application de la GPC sur un pendule inversé

III.1 Introduction.....	26
III.2 Modèle flou de type Takagi-Sugeno.....	26
III.2.1 Le modèle de type Mamdani.....	26
III.2.2 Le modèle de type Takagi-Sugeno.....	27
III.2.2.1 Modèle flou continu.....	27
III.2.2.2 Modèle flou discret.....	28
III.3 La commande d'un pendule inversée par GPC.....	28
III.3.1 La modélisation d'un pendule inversé.....	28
III.3.1.1 Modèle linéarisé du système.....	29
III.3.1.2 Modèle flou T-S.....	31
III.3.2 L'algorithme de GPC sur une représentation d'état.....	34
III.4 Calcul d'observateur flou.....	34
III.4.1 Le principe de fonctionnement d'un observateur.....	35
III.4.2 calcul du gain de l'observateur.....	35
III.5 test et simulation.....	36
III.5.1 Simulation du modèle linéarisé.....	37
III.5.2 Simulation du modèle T-S.....	38

### Chapitre IV: L'application de la commande en mode glissant sur un pendule inversé

IV.1 Introduction.....	41
IV.2 Théorie de la commande par mode de glissement.....	41
IV.2.1 Systèmes à structure variable.....	41
IV.2.1.1 Mode de convergence (MC).....	42
IV.2.1.2 Mode de glissement (MG).....	42
IV.2.1.3 Mode du régime permanent (MRP).....	42
IV.2.3 Différentes structures de la commande par mode de glissement.....	42
IV.2.3.1 Structure par commutation au niveau de l'organe de commande.....	42
IV.2.3.2 Structure par commutation au niveau d'une contre réaction d'état.....	43
IV.2.3.3 Structure par commutation au niveau de l'organe de commande avec commande équivalente.....	43
IV.3 Description du mode glissant.....	44
IV.3.1. Définition du mode glissant.....	44
IV.3.2 Explication du phénomène du glissement.....	44
IV.4 La commande en mode glissant pour les systèmes linéaires.....	45
IV.4.1 Choix de la surface de glissement.....	45
IV.4.2 Conditions de convergence .....	46
IV.4.2.1 La fonction de commutation.....	46
IV.4.2.2 La Fonction de LYAPUNOV.....	46
IV.4.3 Calcul de la commande.....	47
IV.4.3.1 La commande équivalente.....	47
IV.4.3.2 La commande commutant.....	48
IV.4.3.2.a La commande discontinue.....	48
IV.4.3.2.b Commande continue dans une bande de la surface.....	50
IV.4.3.3 Calcul de la commande commutant.....	50

IV.4.4 Le phénomène de réticence ou chattering.....	51
IV.4.5 Lois de commutation par contre-réaction d'état.....	52
IV. 5 La commande en mode glissant pour les systèmes non linéaires.....	53
IV.5.1 Commande discontinue.....	53
IV.5.2 Commande continue.....	56
IV.6 Application de la commande en mode glissant sur le pendule inversé.....	57
IV.6.1 Commande discontinue.....	58
IV.6.2 Commande continue.....	59
Conclusion générale.....	61
Annexe 1: Le modèle TS à partir d'un modèle non linéaire.....	62
Annexe 2: Détermination du gain de glissement.....	63
Références.....	64

## **Introduction générale**

Actuellement, avec les progrès observés dans le domaine des ordinateurs numériques, il est presque impossible de trouver un secteur de l'activité humaine qui ne soit touché par ce phénomène. L'ingénierie de la commande n'a pas échappé à ce mouvement et change rapidement avec le développement conséquent qui se produit dans le domaine numérique et, en particulier, avec la croissance de la capacité de calcul qui est accompagnée avec une réduction de coût.

L'emploi de l'ordinateur dans les processus de commande a permis l'implantation des techniques compliquées et sophistiquées qui permettent le développement de stratégies efficaces tout en restant de coût raisonnable ce qui était pratiquement impossible d'envisager avec les mécanismes et les dispositifs précédents. Des exemples de telles techniques incluent la commande non linéaire, la commande multivariable et la commande robuste.

La commande prédictive basée modèle (*MPC* : Model Predictive Control) connue plus simplement comme commande prédictive, a aussi son origine et son développement à l'emploi de l'ordinateur dans les processus de commande.

La *MPC* est des théories de commande avancée qui a été intensivement étudiée par la communauté des chercheurs, ces dernières décennies. La raison peut être attribuée au fait qu'elle permet d'inclure de manière explicite des restrictions sur les variables du système dans l'étape de calcul de la loi commande ou de conception du contrôleur.

La *MPC* se différencie des autres techniques de commande par le fait que doit être résolu en ligne pour le système en boucle ouverte, un problème d'optimisation à chaque instant d'échantillonnage en tenant compte de l'état réel mesuré du processus. La loi de commande est obtenue en général par le biais de l'optimisation d'un critère qui pénalise l'effort de commande et l'état du système. Du profil de commande obtenu par la solution du problème, on applique au système à commander le premier signal seulement, et le même processus de calculs est répété pour l'instant suivant à partir de la mesure nouvelle. En ce sens on peut également parler de boucle fermée intermittente.

Naturellement, le succès de cette technique dépend du degré de précision du modèle du système à contrôler. Des erreurs de modélisation entraînent une dégradation dans la performance du système commandé et peuvent même causer une instabilité, puisque la prédiction des sorties du système dans les moments de temps futurs, est faite par utilisation du modèle.

S'appuyant sur les avancées de la théorie de la commande robuste, il est possible d'inclure des caractéristiques de robustesse en *MPC* de manière à pouvoir garantir stabilité et performance adéquate quand le modèle du processus est incertain.

On présente dans ce mémoire un des algorithmes de la commande prédictive qui est la commande prédictive généralisée, le premier chapitre contient l'algorithme de base de ce type de

commande. Dans le deuxième chapitre, on présente l'algorithme complet qui prend en compte le traitement des contraintes. Le troisième chapitre contient l'application de la commande prédictive généralisée sur un système non linéaire qui est le pendule inversé.

Enfin, le quatrième chapitre présente une autre technique de commande qui est la commande en mode glissant, on présente la théorie de cette technique sur les systèmes linéaires et non linéaires, leurs avantages, et une application sur le pendule inversé.



## **I.1 Introduction**

La *commande prédictive* ne désigne pas une stratégie de commande spécifique mais un ensemble de méthodes de l'automatique qui utilisent explicitement un modèle du processus à commander, afin d'obtenir le signal de commande par la minimisation d'une fonction de coût. Ces méthodes donnent des correcteurs qui ont pratiquement tous la même structure et qui se basent tous plus ou moins sur les idées suivantes :

- utilisation d'un modèle pour prédire les sorties du procédé à des instants futurs (notion d'*horizon de prédiction*).
- calcul de la séquence de commande qui minimise une fonction de coût dans le futur (notion d'*horizon de commande*).
- à chaque instant d'échantillonnage, l'horizon de prédiction est déplacé vers le futur, et seule la première des commandes calculées est effectivement appliquée au système (notion d'*horizon fuyant*).

De nombreuses applications de la commande prédictive sont utilisées avec succès actuellement, dans l'industrie lourde, et aussi dans des domaines très divers comme les robots manipulateurs, l'anesthésie clinique, l'industrie du ciment...etc. Elle a montré son efficacité, sa flexibilité et son succès même pour des systèmes à faible période d'échantillonnage. Les applications dans le domaine de la conduite des moteurs ont donné de bons résultats en termes de rapidité et de précision.

La commande *MPC* présente un certain nombre d'avantages par rapport aux autres méthodes, parmi lesquels on trouve :

- son principe très intuitif et le réglage relativement facile de ses paramètres la rendent accessible aux personnes avec des connaissances limitées en automatique.
- elle peut être utilisée pour commander une grande variété de processus, ceux avec des dynamiques simples à ceux plus complexes, par exemple les systèmes à grand retards, à phase non minimale ou instables.
- le cas multivariable se traite facilement.
- elle est capable intrinsèquement de compenser les retards et les temps morts.
- le correcteur obtenu est une loi de commande linéaire facile à implémenter et qui requiert peu de temps de calcul.
- le traitement de contraintes sur le système à commander peut être inclus dans la définition du correcteur systématiquement.
- elle est très utile lorsque les consignes ou trajectoires à suivre sont connues à l'avance (ce qui est le cas par exemple dans la commande de robots industriels ou dans certains processus industriels).

Bien sûr, il y a aussi quelques désavantages, l'un d'eux est que, bien que le correcteur obtenu soit simple à programmer, son obtention est beaucoup plus complexe que pour les correcteurs de type *PID* classiques. Si la dynamique du procédé ne change pas, le calcul du correcteur se fait à l'avance, hors-ligne, mais dans le cas d'une commande adaptative, un nombre important de calculs doit être mené à chaque instant d'échantillonnage. Lorsque des contraintes sont considérées, le nombre des calculs nécessaires est encore plus grand. Bien que, avec l'énorme puissance des ordinateurs actuels, ce problème tende à s'estomper, il subsiste lorsque l'application nécessite des fréquences d'échantillonnage très élevées, notamment dans les applications de recherche en asservissement visuel rapide de robots. De plus, dans l'industrie, les systèmes de contrôle et de commande sur lesquels fonctionnent les correcteurs ont habituellement d'autres tâches à accomplir, comme la gestion d'alarmes, l'enregistrement de données, les communications avec les autres systèmes et les opérateurs, ce qui réduit les ressources disponibles pour le correcteur.

L'autre, et le plus grand des inconvénients, est le besoin d'un modèle approprié du système. L'algorithme de calcul du correcteur prédictif se base sur une connaissance à priori de ce modèle, il est donc évident que les bénéfices obtenus avec la commande prédictive soient affectés par les écarts qui peuvent exister entre le processus réel et le modèle utilisé.

## **I.2 Stratégie de la commande prédictive**

La stratégie de la commande prédictive est très similaire à la stratégie utilisée pour la conduite automobile. Le conducteur connaît la trajectoire de référence désirée (le tracé de la route) sur un horizon de commande fini (celui de son champ visuel), et en prenant en compte les caractéristiques de la voiture (modèle mental du comportement du véhicule), il décide quelles actions (accélérer, freiner ou tourner le volant) il faut réaliser afin de suivre la trajectoire désirée. Seule la première action de conduite est exécutée à chaque instant, et la procédure est répétée à nouveau pour les prochaines actions. Notons que lorsqu'on recourt à des schémas de commande classiques, comme les *PID*, les commandes sont calculées à partir des erreurs passées et de l'erreur courante. Si l'on étend l'analogie de la conduite automobile, la technique *PID* serait alors équivalente à conduire une voiture en utilisant seulement ce qui se passe juste devant la voiture (accès à la mesure courante) et le rétroviseur (mesures passées). Mais l'analogie est un peu sévère, car la commande *MPC* a besoin en réalité de beaucoup plus d'informations que le *PID* (la trajectoire à suivre par le système, et un modèle de ce système). Et si un point situé dans le futur sur la trajectoire de référence est utilisé comme consigne pour le *PID*, alors les différences entre les deux stratégies sont moins grandes.

### I.2.1 Principe général

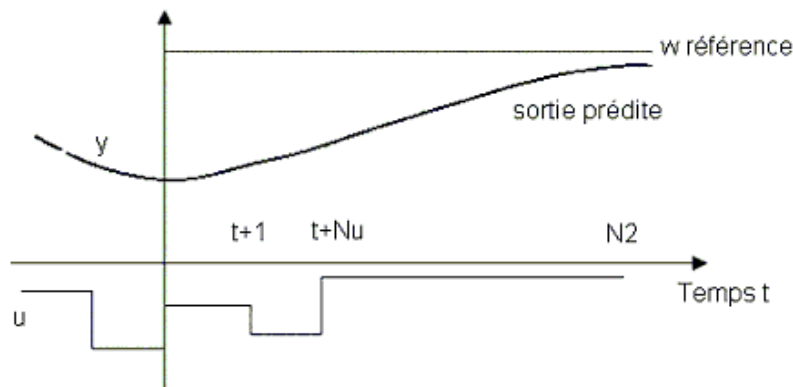
Le principe général de tous les correcteurs de la classe prédictive se caractérise par la stratégie suivante, représentée sur la figure 1.1 :

1. les sorties futures sur un horizon déterminé de taille  $N_2$  (l'horizon de prédiction), sont prédites à chaque instant  $t$  à l'aide du modèle du processus. Ces prédictions sont notées  $y(t+j/t)$ , avec  $j=1:N_2$ , pour indiquer la valeur de la sortie à l'instant  $t+j$  calculée à l'instant  $t$ . Elles dépendent des valeurs connues jusqu'à l'instant  $t$  (entrées et sorties passées) et des commandes futures  $u(t+j/t)$ ,  $j = 0:N_u - 1$ , qui sont celles à envoyer au système et qui doivent être calculées.

2. l'ensemble des commandes futures est calculé en optimisant un critère déterminé pour garder la sortie aussi proche que possible de la trajectoire de référence  $w(t+j)$ .

Ce critère prend généralement la forme d'une fonction quadratique des erreurs entre le signal de sortie prédit et les consignes futures. L'énergie de commande est incluse dans la fonction de coût dans la plupart des cas. Une solution explicite peut être obtenue si le critère est quadratique, le modèle linéaire, et s'il n'y a pas de contraintes, sinon une méthode d'optimisation itérative doit être utilisée. Des hypothèses sont également faites sur la structure de la loi de commande future, comme par exemple qu'elle sera constante à partir d'un instant donné (horizon de commande).

3. La commande  $u(t/t)$  est envoyée au système et les autres valeurs de commande sont oubliées, parce qu'à l'instant d'échantillonnage suivant la mesure  $y(t+1)$  est déjà connue et que l'étape (1) est répétée avec cette nouvelle valeur, toutes les séquences étant mises à jour. Alors on calcule la commande  $u(t+1/t+1)$ , qui est différente de  $u(t+1/t)$  en principe à cause des nouvelles informations prises en compte.



*Figure 1.1: Représentation temporelle du principe de la commande prédictive*

### I.2.2 Modélisation du système

Pour l'implémentation de la stratégie prédictive, la structure de base de la figure 1.2 est mise en oeuvre. Un modèle sert à prédire les futures sorties du système, grâce aux valeurs courantes et

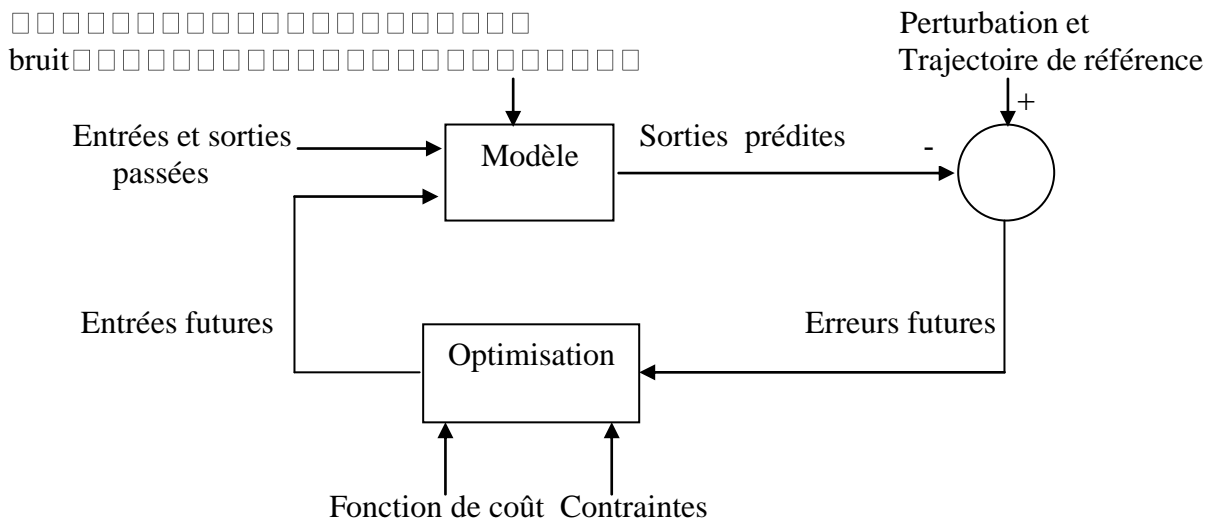
passées de la commande et aux commandes optimales futures. Ces dernières sont calculées par une méthode d'optimisation, qui prend en compte la fonction de coût (qui dépend aussi des consignes futures), et éventuellement des contraintes.

Le modèle du système joue donc un rôle central dans le correcteur. Le modèle choisi doit être capable de rendre compte de la dynamique du processus pour prédire précisément les sorties futures et aussi doit être simple à implémenter et à comprendre.

Tous les algorithmes de commande *MPC* ne diffèrent entre eux que par le modèle utilisé pour représenter le procédé et les bruits, par la fonction de coût à minimiser, et la prise en compte ou non des contraintes.

Les modèles les plus souvent considérés, particulièrement dans l'industrie chimique, sont ceux qui utilisent les réponses impulsionnelles ou indicielles. La réponse impulsionnelle du système est relativement simple à obtenir car elle requiert uniquement la mesure de la sortie lorsque le système est excité par une impulsion. L'autre modèle, qui se base sur la réponse indicielle du système, est obtenu à l'aide des sorties du système lorsque l'entrée excitante est un échelon.

La description du procédé sous forme de fonction de transfert est la plus répandue, car elle permet de décrire un grand nombre de systèmes. On trouve aussi des formulations de la commande prédictive utilisant la représentation d'espace d'état, qui décrit facilement les systèmes multivariables. D'autres modélisations existent encore, qui incluent un modèle du bruit et des perturbations (modèles stochastiques).



*Figure 1.2: Structure de base de la commande prédictive*

### **I.3 Commande prédictive généralisée**

La commande prédictive généralisée (*GPC*) est devenue l'un des algorithmes de commande prédictive les plus répandus. Elle a été très tôt implémentée, avec succès, dans de nombreuses applications industrielles, et a montré de bonnes performances et un certain degré de robustesse.

Elle peut traiter beaucoup de problèmes de commande pour un large étendu de systèmes et met en jeu un nombre raisonnable de paramètres, que l'utilisateur doit spécifier en fonction de sa connaissance a priori du procédé à commander et de ses objectifs de commande.

L'algorithme général de la commande *GPC* est de calculer une séquence de commandes futures de telle façon qu'une fonction de coût à plusieurs composantes soit minimale sur un certain horizon de prédiction. L'indice à optimiser est une fonction quadratique qui mesure la distance entre la sortie prédite du système et une séquence de référence, plus une fonction quadratique qui mesure l'effort de commande.

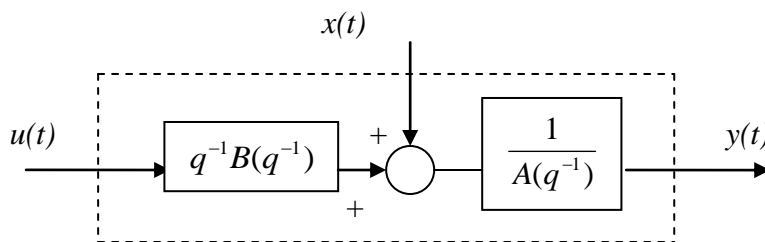
Les spécificités de la commande *GPC* sont l'existence d'une solution optimale analytique (en l'absence de contraintes), le fait qu'elle soit compatible avec des systèmes instables ou à phase non minimale, et aussi la notion d'horizon de commande et d'incrément de commande.

Le *GPC* peut également définir comme étant une version simplifiée du correcteur *LQG* (Linear Quadratic Gaussian). Ils ont en commun la minimisation d'une fonction de coût, cette fonction est minimisée sur un horizon fini fuyant pour la *GPC*, tandis que pour le *LQG*, cette minimisation est faite sur un horizon qui s'étend jusqu'à l'infini.

### **I.3.1 Mise en équation de système**

La loi de commande prédictive n'est pas astreinte à un type de représentation particulier, mais historiquement adopte de préférence une forme de représentation du système de type polynomiale externe entrée-sortie. Le système est alors représenté sous la forme CARIMA (Controlled Auto-Regressive Integrated Moving Average), correspondant à la figure 1.3 dont l'expression est :

$$A(q^{-1})y(t) = B(q^{-1})u(t - T_e d) + \frac{c(q^{-1})}{\Delta(q^{-1})} \xi(t) \quad (1.1)$$



**Figure 1.3: Représentation CARMA du modèle numérique**

Avec :  $y(t)$  sortie du processus  
 $u(t)$  commande appliquée au système  
 $q^{-1}$  opérateur retard

$x(t)$  terme lié aux perturbations, choisi généralement sous la forme  $x(t) = \frac{C(q^{-1})}{\Delta(q^{-1})} \xi(t)$  ,

$\xi(t)$  est un bruit blanc.

Les polynômes  $A(q^{-1})$ ,  $B(q^{-1})$  et  $C(q^{-1})$  sont définis par:

$$\begin{aligned} A(q^{-1}) &= a_0 + a_1 q^{-1} + \dots + a_{na} q^{-na} \\ B(q^{-1}) &= b_0 + b_1 q^{-1} + \dots + b_{nb} q^{-nb} \\ C(q^{-1}) &= 1 + c_1 q^{-1} + \dots + c_{nc} q^{-nc} \end{aligned}$$

Le polynôme  $\Delta(q^{-1})$  est pris égal à l'opérateur de différentiation discrète, que l'on notera  $\delta(q^{-1})$ , soit :

$$\Delta(q^{-1}) = \delta(q^{-1}) = 1 - q^{-1} \quad (1.2)$$

Le but recherché avec l'introduction d'un intégrateur est l'annulation de toute erreur statique vis-à-vis d'une consigne ou d'une perturbation constante.

### **I.3.2 Prédicteur optimal**

La méthodologie prédictive requiert la définition d'un prédicteur optimal à  $j$ -pas qui permette d'anticiper le comportement du processus dans le futur sur un horizon fini. Pour cela, à partir de la forme du modèle d'équation (1.1), on élabore la sortie estimée à l'instant  $t+j$ , connaissant la sortie à l'instant  $t$ . La sortie prédite  $y(t+j)$  est décomposée de façon classique en réponse libre et réponse forcée, incluant une forme polynomiale pour mener à bien la synthèse polynomiale finale, sous la forme :

$$\hat{y}(t+j) = \underbrace{F_j(q^{-1})y(t) + H_j(q^{-1})\Delta u(t-1)}_{\text{Réponse libre}} + \underbrace{G_j(q^{-1})\Delta u(t+j-1) + J_j(q^{-1})\xi(t+j)}_{\text{Réponse forcée}} \quad (1.3)$$

Avec  $G_j$  représentant le futur,  $H_j$ ,  $F_j$  correspondant au présent et au passé,  $J_j$  lié aux perturbations. Dès lors, le premier terme représentant la réponse libre est dû aux incréments de commandes passées, le deuxième terme représentant la réponse forcée correspond aux incréments de commandes futures et présente, et à l'influence des perturbations.

L'équation du modèle CARIMA (équation (1.1)), combinée avec celle du prédicteur (équation (1.3)) fournit le système d'équations ci-dessous :

$$A(q^{-1})\Delta(q^{-1})y(t+j) = B(q^{-1})\Delta u(t+j-1) + \xi(t+j) \quad (1.4)$$

$$[1 - q^{-j} F_j(q^{-1})]y(t+j) = [G_j(q^{-1}) + q^{-j} H_j(q^{-1})]\Delta u(t+j-1) + J_j(q^{-1})\xi(t+j) \quad (1.5)$$

Le prédicteur optimal est défini, après la résolution des équations de Diophantine [1], par la relation :

$$\hat{y}(t+j) = F_j(q^{-1})y(t) + G_j(q^{-1})\Delta u(t+j-1) + H_j(q^{-1})\Delta u(t-1) \quad (1.6)$$

Avec: degré de  $[J_j(q^{-1})] = \text{degré de } [G_j(q^{-1})] = j-1$

degré de  $[F_j(q^{-1})] = \text{degré de } [A(q^{-1})]$

degré de  $[H_j(q^{-1})] = \text{degré de } [B(q^{-1})]$

### **I.3.3 Fonction de coût quadratique**

L'objectif du correcteur *GPC* est de minimiser l'erreur quadratique entre les prédictions de la sortie et les consignes futures, cela signifie qu'il doit réaliser une estimation des prédictions des valeurs de la sortie  $y$  aux instants d'échantillonnage futurs en fonction des valeurs futures de l'entrée  $u$ .

On appelle  $w$  la grandeur de consigne de l'asservissement dont la valeur est supposée connue non seulement à l'instant  $k$  présent, mais également pendant les  $N_2$  instants d'échantillonnages suivants.

Le correcteur *GPC* génère, à chaque instant d'échantillonnage,  $N_u$  incréments de commande  $\Delta u$  de manière à minimiser la fonction de coût suivante :

$$J(u, k) = \sum_{j=N_1}^{N_2} [\hat{y}(k+j) - w(k+j)]^2 + \lambda \sum_{j=1}^{N_u} \Delta u(k+j-1)^2 \quad (1.7)$$

Avec :  $w(t+j)$  □ consigne appliquée à l'instant  $t+j$ ,  $\hat{y}(t+j)$  □ sortie prédite à l'instant  $t+j$ .

$\Delta u(t+j-1)$  incrément de commande à l'instant  $t+j-1$ .

$N_u < N_2$  et  $\Delta u(k+j) = 0, \forall j \geq N_u$ .

$N_1, N_2$  et  $N_u$  sont des entiers strictement positifs et  $\lambda$  □ est un scalaire positif,

- $N_1$  : horizon de prédiction minimal sur la sortie.
- $N_2$  : horizon de prédiction maximal sur la sortie.
- $N_u$  : horizon de prédiction sur la commande.
- $\lambda$  : coefficient de pondération sur la commande.

Ce critère comprend donc un terme quadratique sur l'erreur et l'incrément de la commande. Sa minimisation analytique fournit la séquence des commandes futures dont seule la première sera effectivement appliquée.

L'aspect incrémental du modèle se retrouve dans le critère par l'intermédiaire de  $\Delta u$ . Enfin le coefficient  $\lambda$  permet de donner plus ou moins de poids à la commande par rapport à la sortie, de façon à assurer la convergence lorsque le système de départ présente un risque d'instabilité.

### **I.3.4 Prédicteur optimal sous forme matricielle**

Pour simplifier les notations de prédicteur optimal donné par l'équation (1.7), il est possible d'utiliser une représentation matricielle, posons pour cela :

$$\begin{aligned} if(q^{-1}) &= [F_{N_1}(q^{-1}) \dots F_{N_2}(q^{-1})] & ih(q^{-1}) &= [H_{N_1}(q^{-1}) \dots H_{N_2}(q^{-1})] \\ \hat{u} &= [\Delta u(t) \dots \Delta u(t+N_u-1)] & \hat{y} &= [\hat{y}(t+N_1) \dots \hat{y}(t+N_2)] \end{aligned}$$

$$G = \begin{bmatrix} g_{N_1}^{N_1} & g_{N_1-1}^{N_1} & \dots & \dots \\ g_{N_1+1}^{N_1+1} & g_{N_1}^{N_1+1} & \dots & \dots \\ \dots & \dots & \dots & \dots \\ g_{N_2}^{N_2} & g_{N_2-1}^{N_2} & \dots & g_{N_2-N_u+1}^{N_2} \end{bmatrix}$$

Avec ces notations, le prédicteur optimal à  $j$ -pas peut s'écrire sous forme matricielle :

$$\hat{y} = G\tilde{u} + if(q^{-1})y(t) + ih(q^{-1})\Delta u(t-1) \quad (1.8)$$

Le critère quadratique d'équation (1.6) peut se combiner avec la relation (1.7) pour obtenir l'expression matricielle de ce critère :

$$j = \left[ G\tilde{u} + if(q^{-1})y(t) + ih(q^{-1})\Delta u(t-1) - w \right] \left[ G\tilde{u} + if(q^{-1})y(t) + ih(q^{-1})\Delta u(t-1) - w \right] + \lambda \tilde{u}'\tilde{u} \quad (1.9)$$

### I.3.5 Minimisation de la fonction de coût

La commande optimale s'obtient par minimisation analytique du critère sous forme matricielle,

$$\frac{\partial J}{\partial u} = 0 \quad \Leftrightarrow \quad \tilde{u} = M \left[ w - if(q^{-1})y(t) - ih(q^{-1})\Delta u(t-1) \right] \quad (1.10)$$

Avec les matrices:  $M = QG'$ , de dimension  $N_u \times (N_2 - N_1 + 1)$

$$Q = \left[ G'G + \lambda I_{N_u} \right]^{-1}, \text{ de dimension } N_u \times N_u$$

En commande prédictive, seule la première valeur de la séquence (équation (1.10)) est finalement appliquée au système en accord avec la stratégie de l'horizon fuyant, l'ensemble de la procédure étant effectué de nouveau à la période d'échantillonnage suivante.

$$\Delta u_{opt}(t) = -m_1' \left[ if(q^{-1})y(t) + ih(q^{-1})\Delta u(t-1) - w \right] \quad (1.11)$$

Avec  $m_1$  est la première ligne de la matrice  $M$ .

### I.3.6 Structure RST du régulateur

La structure *RST* polynomiale est introduite afin d'obtenir une relation entre la sortie  $y(t)$ , la commande  $u(t)$  et la consigne  $w(t)$ . À partir de la relation (1.11), il vient :

$$\Delta u_{opt}(t) \left[ 1 + m_1' ih(q^{-1})q^{-1} \right] = -m_1' if(q^{-1})y(t) + m_1' \left[ q^{N_1} \dots \dots q^{N_2} \right] w(t) \quad (1.12)$$

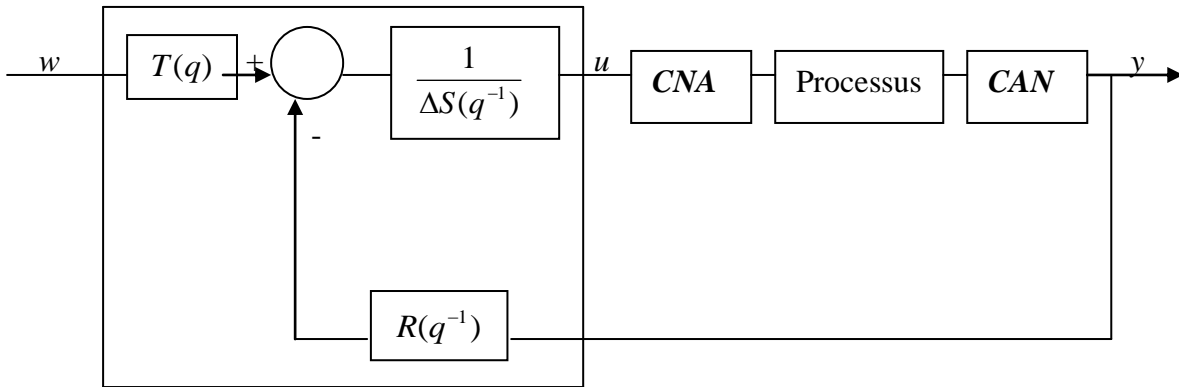
Cette relation doit correspondre d'après la figure 1.4 à l'équation :

$$S(q^{-1})\Delta(q^{-1})u(t) = -R(q^{-1})y(t) + T(q)w(t) \quad (1.13)$$

Ce qui fournit par identification les trois polynômes  $R$ ,  $S$  et  $T$  constituant le régulateur linéaire équivalent :

$$\begin{aligned} S(q^{-1}) &= 1 + m_1' ih(q^{-1})q^{-1} = 1 + S^*(q^{-1})q^{-1} && \text{degré de } [S(q^{-1})] = \text{degré } [B(q^{-1})] \\ R(q^{-1}) &= m_1' if(q^{-1}) && \text{degré de } [R(q^{-1})] = \text{degré de } [A(q^{-1})] \\ T(q) &= m_1' [q^{N_1} \dots \dots q^{N_2}] && \text{degré de } [T(q)] = N_2 \end{aligned}$$





régulateur polynomial  
équivalent

**Figure 1.4:** Structure RST de la commande GPC

L'intérêt de la représentation *RST* (par ailleurs très générale puisque toute loi de commande numérique peut se modéliser ainsi) est que la boucle temps réel s'avère très peu gourmande en temps de calcul, puisque la commande appliquée au système se calcule par une simple équation aux différences. Les trois polynômes *RST*, sont en effet élaborés hors ligne et définis de façon unique dès lors que les quatre paramètres de réglage  $N_1, N_2, N_u, \lambda$  sont choisis. En conséquence, ce type de commande permet la sélection de périodes d'échantillonnage faibles et s'avère bien adapté à la commande de systèmes électromécaniques rapides (machine outil, usinage très grande vitesse, ...).

Un autre intérêt majeur de cette structure *RST* concerne l'étude de la stabilité de la boucle corrigée, et donc la caractérisation de la stabilité de la commande prédictive élaborée, qui est dès lors possible pour un jeu de paramètres du critère fixé. On peut vérifier la position des pôles de la fonction caractéristique en boucle fermée.

## I.4 Choix de paramètres de réglage

La définition du critère quadratique a montré que l'utilisateur doit fixer quatre paramètres de réglage. Ce choix des paramètres s'avère cependant délicat pour une personne non spécialiste, car il n'existe pas de relations empiriques permettant de relier ces paramètres à des 'indicateurs' classiques en Automatique, tels que marges de stabilité où bande passante.

Nous énonçons ci-dessous quelques idées guidant le choix des paramètres de réglage, obtenues à partir de l'étude d'un grand nombre de systèmes.

### I.4.1 Choix de l'horizon minimal de prédiction $N_1$

Le produit  $Te N_1$  ( $Te$  période d'échantillonnage) est choisi égal au retard pur du système. Ainsi, pour un système ne présentant pas de retard ou un retard mal connu ou variable,  $N_1$  est choisi

égal à 1. Cependant, dans le cas d'un système à déphasage non minimal,  $N_1$  est pris égal à la valeur  $N_0$  indiqué sur la figure 1.5.

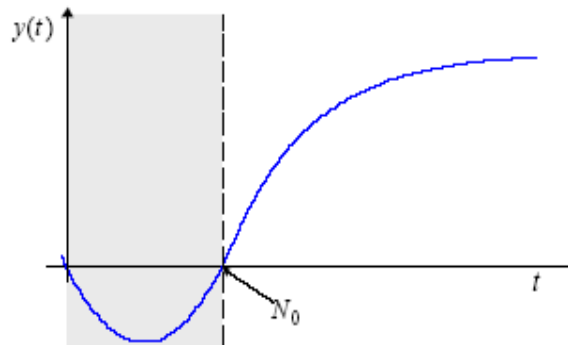


Figure 1.5: Valeur de  $N_1$  pour un système à déphasage non minimal

#### I.4.2 Choix de l'horizon maximal de prédiction $N_2$

$N_2$  est choisi de sorte que le produit  $Te N_2$  soit limité par la valeur du temps de réponse souhaité. En effet augmenter la prédiction au delà du temps de réponse n'apporte aucune information supplémentaire et complexifie la résolution. Par ailleurs, plus  $N_2$  est grand, plus le système corrigé est lent et la correction est mieux. Notons sur le schéma de la figure 1.6 le choix de  $N_2$  dans le cas d'un système à déphasage non minimal.

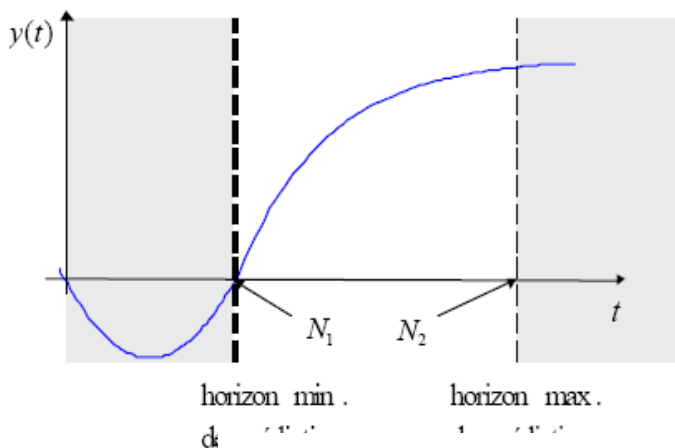


Figure 1.6: Choix de  $N_2$  pour un système à déphasage non minimal

#### I.4.3 Choix de l'horizon de prédiction sur la commande $N_u$

Généralement  $N_u$  est choisi égal au nombre d'états du système, ainsi la valeur  $N_u=1$  est très souvent suffisante pour beaucoup d'applications relativement simples. Dans ce dernier cas, le calcul de la séquence de commandes futures se réduit au simple calcul du scalaire  $\tilde{u}_{opt}$ .

#### I.4.4 Choix du facteur de pondération de la commande $\lambda$

On peut interpréter le facteur de pondération  $\lambda$  comme 'l'équilibre de la balance'. En effet, si  $\lambda=0$ , on minimise uniquement dans le critère quadratique, la différence entre la consigne et la

sortie prédite. Il peut donc en résulter une commande très forte pouvant faire diverger le processus réel. D'autre part, si  $\lambda$  est très élevé, on pondère alors excessivement la commande qui n'est plus assez 'dynamique' pour obtenir le ralliement à la consigne. Dans le cadre monovarié, partant donc de la constatation que plus le gain d'un système est grand, plus la commande doit être pondérée (c'est-à-dire plus  $\lambda$  est important, et vice et versa), il apparaît alors clairement que  $\lambda$  est lié au gain du système. Une relation permettant de déterminer rapidement la valeur de  $\lambda$  apportant au système le maximum de stabilité est donnée par :

$$\lambda_{opt} = trace(G^T G) \tag{1.14}$$

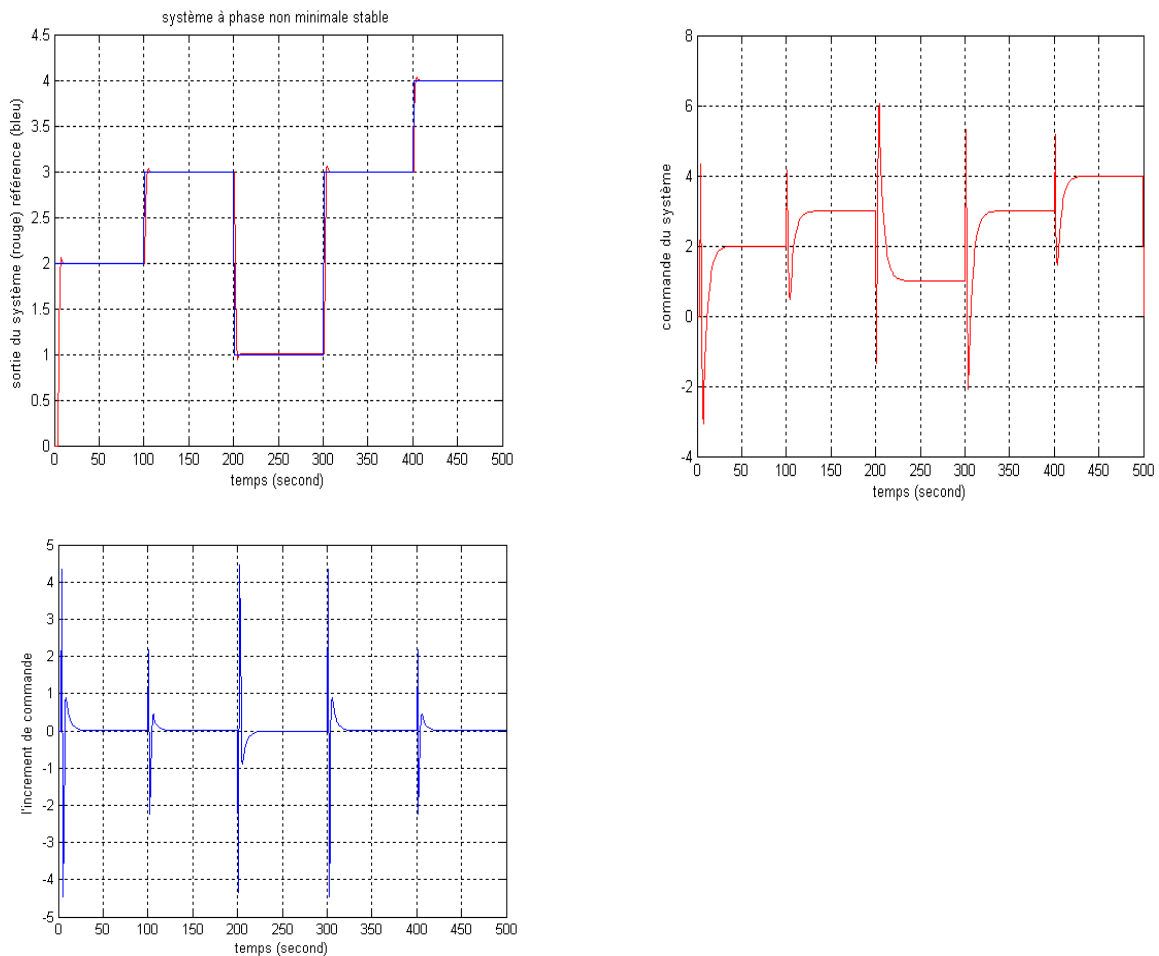
### **I.5 Exemple d'application**

Le système choisi est donné par l'équation en récurrence suivante :

$$(1 - 2.4428q^{-1} + 1.4918q^{-2})Y(q) = q^{-1}(0.2672 - 0.2181q^{-1})U(q)$$

Les paramètres de commande choisis sont :

$$N_1 = 1 ; N_2 = 6 ; N_u = 6 ; \lambda = 0.02$$



**Figure 1.7:** Exemple d'application de la GPC sans contraintes

**Commentaire :**

Le système se stabilise rapidement avec un petit dépassement : la sortie du système et la référence sont presque identiques, on peut dire que la GPC sans contraintes est très utile dans la commande des systèmes représentés par des fonctions de transfert.

## II.1 Introduction

Les contraintes, les plus souvent définies dans le domaine temporel, interviennent sur toutes les parties d'un système de commande, des actionneurs aux capteurs, et en général comme contraintes sur les états du système à commander. Leur présence est source de problèmes parfois complexes, non seulement pour les praticiens de l'Automatique mais aussi pour le milieu académique, car ils constituent le type de non linéarité le plus classique affectant en pratique une loi de commande, avec des implications directes sur la stabilité, les performances et la sûreté de fonctionnement du système en boucle fermée.

Dans beaucoup d'applications pratiques de la commande, le système non linéaire à commander est approximé par un modèle linéaire. De plus, vu la nature non linéaire des actionneurs, les non linéarités sont inhérentes aux boucles de commande en dépit des modèles linéaires. Les non linéarités sont en pratique des contraintes, et parmi ces non linéarités on peut citer la saturation, la zone morte, ...etc. Ces contraintes peuvent se matérialiser dans la pratique par exemple par la limitation physique des actionneurs, comme le cas de la commande des vannes qui sont limitées par l'ouverture et la fermeture complète ainsi que par l'incrément maximal du déplacement dans les deux sens; un autre exemple est la limitation des plages de fonctionnement des capteurs. On peut citer aussi les exigences de sécurité qui prohibent certaines plages de fonctionnement en limitant la sortie du procédé.

Les techniques de la commande prédictive intègrent les contraintes pendant la phase de synthèse et d'implantation du contrôleur, qui nous permet de présenter les contraintes d'une façon directe de sorte que l'algorithme trouve la meilleure solution admissible automatiquement.

## II.2 Types de contraintes

Les contraintes sont des relations qui limitent le domaine de variation des variables. Elles peuvent être linéaires ou non linéaires.

Les contraintes d'égalité sont des équations reliant certaines variables entre elles: les équations du modèle, la composition des matières premières ou des variables externes non contrôlables comme la pression atmosphérique sont d'autres exemples de contraintes d'égalité.

Les contraintes d'inégalité définissent des limites aux valeurs de certaines variables ou de certaines fonctions de variables. Elles sont dues aux caractéristiques des équipements (pression ou température maximum de fonctionnement d'un appareil), des spécifications commerciales (pureté d'un produit).

### II.2.1 contraintes sur l'amplitude du signal de commande

Les restrictions sur l'amplitude du signal de la commande, assez fréquentes en pratique, peuvent s'exprimer au moyen de l'inégalité suivante:

$$u_{\min} \leq u(k) \leq u_{\max} \quad (2.1)$$

Ces contraintes sont à satisfaire sur tout l'horizon d'optimisation :

$$1u_{\min} \leq U(k) \leq 1u_{\max} \quad (2.2)$$

où  $U(k)=[u(k) \ u(k+1) \ \dots \ u(k+N-1)]^T$

$1$  vecteur de dimension  $N_u$  à éléments égaux à 1.

Il convient parfois de représenter cette contrainte en fonction des accroissements successifs  $\Delta U(k)$  de la commande, soit

$$1u_{\min} \leq T\Delta U(k) + 1u(k-1) \leq 1u_{\max} \quad (2.3)$$

Où  $T$  est une matrice  $N_u \times N_u$ , triangulaire inférieure à éléments égaux à 1.

### II.2.2 Contraintes sur l'incrément de commande

Les restrictions sur l'augmentation du signal de commande prennent une forme très simple, et peuvent être exprimées au moyen de l'inégalité:

$$\Delta u_{\min} \leq u(k) - u(k-1) \leq \Delta u_{\max} \quad (2.4)$$

Ou sous la forme vectorielle:

$$1\Delta u_{\min} \leq \Delta U(k) \leq 1\Delta u_{\max} \quad (2.5)$$

### II.2.3 Contraintes sur la sortie

Il est très fréquent de trouver comme spécification désirée dans les processus commandés que leur sortie doit être trouvée dans une fourchette autour d'une trajectoire désirée, ceci se traduit par une inégalité de la forme:

$$y_{\min} \leq y(k) \leq y_{\max} \quad (2.6)$$

Sur l'horizon d'optimisation la contrainte s'écrit :

$$Y(k)_{\min} \leq Y(k) \leq Y(k)_{\max} \quad (2.7)$$

où  $Y(k)_{\min}=[y(k+1)_{\min} \ y(k+2)_{\min} \ \dots \ y(k+N)_{\min}]$

$Y(k)_{\max}=[y(k+1)_{\max} \ y(k+2)_{\max} \ \dots \ y(k+N)_{\max}]$

D'après la relation (1.7) de premier chapitre, et on posant :

$$if(q^{-1})y(k) + ih(q^{-1})\Delta u(k-1) = f(k) \quad (2.8)$$

Une telle contrainte par rapport aux incréments sur la variable de commande peut s'écrire :

$$Y(k)_{\min} \leq G\Delta U(k) + f(k) \leq Y(k)_{\max} \quad (2.9)$$

Où  $f(k)$  est la sortie de régime libre a partir de  $k$ .

Sur les accroissements de la commande, ces diverses contraintes peuvent s'exprimer sous forme condensée comme:

$$R\Delta u \leq c \tag{2.10}$$

Avec:

$$R = \begin{bmatrix} I_{N \times N} \\ -I_{N \times N} \\ T \\ -T \\ G \\ -G \end{bmatrix} \quad c(k) = \begin{bmatrix} 1\Delta u_{\max} \\ -1\Delta u_{\min} \\ 1\Delta u_{\max} - 1u(k-1) \\ -1\Delta u_{\min} + 1u(k-1) \\ 1Y(k)_{\max} - f(k) \\ -1Y(k)_{\min} + f(k) \end{bmatrix}$$

### II.2.4 Contraintes sur les oscillations permises dans la sortie du système

Dans de nombreux processus, les fortes oscillations pour les variables de sortie ne sont pas souhaitables, c'est le cas des manipulateurs, pour lesquels une sur oscillation peut produire des collisions avec d'autres éléments dans l'aire de travail. Il est également très facile de traiter ce type de restrictions dans le contexte prédictif. Toutes les fois qu'un changement de la référence intervient, et qui est maintenue constante sur une période suffisamment longue, les restrictions suivantes sont ajoutées au système de commande [2]:

$$y(k+i) \leq \gamma w(k) \quad \text{pour } i=N_{01} \dots N_{02}$$

Où  $N_{01}$  et  $N_{02}$  définissent l'horizon pendant lequel la sur oscillation peut être observée (ils peuvent toujours être pris égaux à 1 et à  $N_2$  respectivement si cet horizon n'est pas connu) et  $\gamma > 1$  est un facteur généralement près de l'unité qui permet de définir le maximum permis pour les sur oscillations par rapport à la référence. En termes d'augmentation des variables manipulées, il est ainsi possible d'écrire:

$$G\Delta u \leq \gamma w(k) - f \tag{2.11}$$

### II.2.5 Contrainte pour éviter des comportements de phase non minimale [2]

Il existe des processus qui exhibent naturellement un comportement de phase non minimale, cela se produit quand le processus sous l'effet de variation brusque de l'entrée tend à voir, aux premiers moments transitoires, la variable de sortie évoluer dans le sens contraire à sa position finale. Ce comportement peut ne pas être souhaitable dans certains cas, des restrictions de la forme suivante peuvent alors être définies: contraintes de dépassement

$$y(k+j) < y(k) \text{ si } y(k) > w(k)$$

$$y(k+j) > y(k) \text{ si } y(k) < w(k)$$

En termes de variables de commande ces contraintes peuvent être représentées par l'inégalité (choisissant par exemple le cas  $y(k) < w(k)$ ):

$$G\Delta u \geq 1y(k) - f \quad (2.12)$$

## II.3 La programmation de GPC avec contraintes

### II.3.1 La programmation Dynamique

C'est une méthode de résolution des problèmes d'optimisation dynamique (commande optimale) développé par Richard Bellman (Bellman 1957), elle s'inspire directement du principe de Bellman: le principe d'optimalité.

#### Principe d'optimalité:

*Une politique optimale est elle que, quels que soient l'état initial et la décision initiale, les décisions suivantes doivent constituer une politique optimale par rapport à l'état résultant de la première décision [4].*

#### II.3.1.1 Formulation du problème

Considérons un processus discret décrit par l'équation :

$$x(k+1) = f(x(k), u(k), k) \quad (2.13)$$

Où  $x$  est le vecteur d'état,  $u$  est le vecteur de commande, et  $k$  représente l'étape courante.

Supposons de plus que la commande est soumise à chaque étape à la contraintes  $u(k) \in \Omega_k$ .

Le problème est de trouver la séquence de commande  $\{u(0), u(1), \dots, u(N-1)\}$  qui fasse évoluer le processus de l'état  $x(0)$  à l'instant  $k=0$  vers l'état  $x(N)$  à l'instant  $N$  en satisfaisant les contraintes imposées et en minimisant le critère :

$$J = \sum_0^{N-1} l(x(k), u(k), k) \quad (2.14)$$

Où  $l(x(k), u(k), k)$  le revenu élémentaire obtenu en partant de l'état  $x(k)$  à l'instant  $k$  avec la commande  $u(k)$ .

Si l'instant final  $N$  est non fixé le processus est dit à l'horizon libre, si  $N = \infty$  le processus est dit à l'horizon infini, et si  $x(N)$  est non fixé, l'état final est dit libre.

#### II.3.1.2 Equation récurrente d'optimalité

Tout d'abord nous immergeons le problème dans un problème plus vaste qui est la recherche de la séquence de commande minimisant le critère :

$$J(x(k), k) = \sum_{i=k}^{N-1} l(x(i), u(i), i) \quad (2.15)$$

En partant de l'état  $x(k)$ , à l'instant  $k$ , tout en satisfaisant aux diverses contraintes, la solution si elle existe s'exprime sous la forme :

$$J^*(x(k), k) = \underset{u(k) \in \Omega_k, \dots, u(N-1) \in \Omega_{N-1}}{\text{Min}} \left( \sum_{i=k}^{N-1} l(x(i), u(i), i) \right) \quad (2.16)$$



En explicitant la relation entre les solutions et les données entre deux instants consécutifs  $k$  et  $k+1$ , on voit que l'optimum du revenu immédiat  $l(x(k), u(k), k)$  plus le revenu future  $J^*(x(k+1), k+1)$  en partant du nouveau point  $x(k+1)$  est :

$$J^*(x(k), k) = \underset{u_k \in \Omega_k}{\text{Min}} (l(x(k), u(k), k) + J^*(f(x(k), u(k), k), k+1)) \quad (2.17)$$

Qui porte le nom d'équation récurrente d'optimalité.

La résolution de cette équation s'effectue de proche en proche à partir de l'état final.

### II.3.1.3 Algorithme de la Programmation Dynamique

Dans la section précédente nous avons vu comment un problème d'optimisation dynamique pouvait en principe être résolu par programmation dynamique, en particulier nous avons vu que la solution consistait à résoudre l'équation (3.14) de proche en proche.

Nous développons maintenant l'algorithme dit « backward » le plus utilisé pour obtenir cette solution. L'idée de base de l'algorithme backward est de commencer par l'état final et de faire marche arrière en calculant la valeur de la fonctionnelle jusqu'au point initial.

Pour chaque état et chaque étape, on calcule la valeur optimale de la fonctionnelle, et on détermine la décision optimale associée au processus commençant à l'état considéré.

Supposons que l'on soit à l'étape  $N-1$ , état  $x^i(N-1)$ , nous cherchons la transition qui minimise  $l$  :

$$s(x^i(N-1), N-1) = \underset{u}{\text{Min}} l(x^i(N-1), x^j(N)) \quad (2.18)$$

$i=1,2,3,\dots,n ; j=1,2,3,\dots,n.$

Passant à l'étape  $N-2$ , état  $x^i(N-2)$ , on appelle la valeur optimale de la fonctionnelle

$$s(x^i(N-2), N-2) = \underset{u}{\text{Min}} (l(x^i(N-2), x^j(N-1)) + s(x^j(N-1), N-1)) \quad (2.19)$$

## II.3.2 La programmation quadratique

### II.3.2.1 Introduction

Le traitement des contraintes pour la GPC a été introduit par *T.SANG* et *CLARKE* où deux algorithmes ont été développés pour prendre en considération les contraintes sur l'entrée des procédés pour le cas particulier de ( $N_u=2$ ), le premier traite les contraintes sur l'incrément de commande alors que le deuxième traite les contraintes sur la commande.

*CLARKE* et *MOHTADI*, utilisent la programmation quadratique pour résoudre le problème des contraintes d'une façon générale, c'est cette hypothèse qu'on va appliquer pour traiter notre problème, en faisant appel à la fonction, *QuadProg* (quadratic programming) disponible dans le Toolbox de MATLAB.

### II.3.2.2 Formulation du problème

Pour le cas des contraintes affectant les entrées sur la commande on a :

$$u_{min} \leq u(t+j-1) \leq u_{max} \quad (2.20)$$

Avec:  $j=N_l$  à  $N_u$

Après développement de la relation de commande

$$u(t+j-1) = \Delta u(t+j-1) + u(t+j-2) \quad (2.21)$$

Nous arrivons à la nouvelle relation suivante:

$$u(t+j-1) = u(t-1) + \sum_{i=1}^{j-1} \Delta u(t+i-1) \quad (2.22)$$

Après les simplifications en utilisant aussi (2.10) :

$$u_{min} - u(t-1) \leq \sum_{i=1}^{j-1} \Delta u(t+i-1) \leq u_{max} - u(t-1) \quad (2.23)$$

$N_l$  et  $N_u$  sont respectivement l'horizon de contraintes, inférieur et supérieur de contrôle.

L'inégalité (2.20) peut s'écrire comme suit :

$$A \cdot \tilde{U} \leq \tilde{U}_{max} \quad (2.24)$$

$$-A \tilde{U} \leq -\tilde{U}_{min} \quad (2.25)$$

Avec:

$$\tilde{U}_{max} = (\Delta u_{max}, \Delta u_{max}, \dots, \Delta u_{max})^T$$

$$\tilde{U}_{min} = (\Delta u_{min}, \Delta u_{min}, \dots, \Delta u_{min})^T$$

$$\tilde{U} = [\Delta u(t), \Delta u(t+1), \dots, \Delta u(t+N_u)]^T$$

A: est une matrice triangulaire inférieure de dimension ( $N_u \times N_u$ )

$$A = \begin{bmatrix} 1 & 0 & \dots & 0 \\ 1 & 1 & 0 & \dots & 0 \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ 1 & 1 & 1 & \dots & 1 \end{bmatrix}$$

Et pour les contraintes affectant les incréments de commande on a :

$$\Delta u_{min} \leq \Delta u(t+j-1) \leq \Delta u_{max} \quad (2.26)$$

Avec :  $j=N_l, N_u$

Les inégalités (2.26) peuvent s'écrire comme suit :

$$I \cdot U \leq \Delta u_{max} \quad (2.27)$$

$$-I \cdot U \leq -\Delta u_{max} \quad (2.28)$$

Avec:  $I$ : matrice d'identité de dimension ( $N_u \times N_u$ )

$$U_{max} = [\Delta u_{max}, \Delta u_{max}, \dots, \Delta u_{max}]^T \text{ de dimension } (N_u \times 1)$$

$$U_{min} = [\Delta u_{min}, \Delta u_{min}, \dots, \Delta u_{min}]^T \text{ de dimension } (N_u \times 1)$$

$$U = [\Delta u(t), \Delta u(t+1), \dots, \Delta u(t+N_u)]^T \text{ de dimension } (N_u \times 1)$$

### II.3.2.3 Résolution du problème

Le critère d'optimisation donné par l'équation (1.22), peut être réécrit comme suit :

$$J = (G.U + f - W)^T (G.U + f - W) + \lambda U^T U = \frac{1}{2} U^T H U + Q U + f_0 \quad (2.29)$$

Où :  $H = G^T G + \lambda I$

$$Q = 2 (f - W)^T G$$

$$f_0 = - W^T (I - W) = \text{constant}$$

Après que  $H$  et  $Q$  soient calculés, le problème de l'optimisation est résolu facilement, on appelle la fonction QuadProg du Matlab.

## II.4 Autres méthodes d'optimisation

L'algorithme d'optimisation est une partie fondamentale de la stratégie. Le problème d'optimisation peut être formulé de la façon suivante: trouver la combinaison de paramètres qui optimise une quantité donnée (la fonction de coût), pouvant être soumise à quelques limitations (contraintes). Si la fonction de coût est quadratique, son minimum peut être obtenu comme une fonction (linéaire) explicite des entrées et sorties passées et des consignes futures. La taille du problème d'optimisation dépend du nombre de variables et des horizons de prédiction et est en général relativement modeste (calcul d'inverse d'une matrice par exemple). En présence de contraintes par contre (contraintes d'inégalité sur les commandes ou les sorties), la solution s'obtient numériquement à l'aide d'algorithmes beaucoup plus coûteux en temps de calcul.

Il existe beaucoup d'algorithmes d'optimisation dans différentes applications scientifiques. Cependant beaucoup de méthodes ne sont valables que pour certains types de problèmes. Dans ce mémoire, on va présenter trois algorithmes d'optimisation:

### II.4.1 La méthode de gradient conjugué

Cette méthode a été proposée à l'origine pour la minimisation de fonctions quadratiques et convexes, mais en suite, elle a été étendue pour le cas général. De point de vue algorithmique la méthode de gradient conjugué commence comme la méthode de la plus forte pente, mais les directions de descente successives sont construites de façon à ce qu'elles soient mutuellement conjuguées par rapport à la matrice  $A$  pour une fonction objective quadratique et convexe.

#### II.4.1.1 Le cas d'une fonction quadratique

Soit  $f(x) = (1/2)x^t A x + B^t x + C$  une fonction quadratique, où  $A$  est définie positive.

La méthode consiste, à partir d'un point  $x^0$ , à minimiser  $f$  suivant  $n$  directions  $d^0, d^1, \dots, d^{n-1}$  mutuellement conjuguées par rapport à  $A$ , c'est-à-dire vérifiant :

Pour  $1 \leq i < j \leq n$ ,  $(d^i)^t A d^j = 0$ .

Soient  $n$  telles directions :  $d^0, d^1, \dots, d^{n-1}$ .

Ayant déterminé  $x^k$ , le point  $x^{k+1}$  est le point:  $x^{k+1} = x^k + s_k d^k$  où  $s_k$  est choisi de façon à minimiser  $f(x^k + s_k d^k)$ .

On a donc :  $(d^k)^t \cdot \tilde{\nabla} f(x^k + s_k d^k) = 0$  ou encore :  $(d^k)^t \cdot [A(x^k + s_k d^k) + B] = 0$

D'où l'on déduit :

$$s_k = - \frac{(d^k)^t (Ax^k + B)}{(d^k)^t \cdot A \cdot d^k} \quad (2.30)$$

Lemme: Si  $d^0, d^1, \dots, d^{k-1}$  sont mutuellement conjuguées, alors on a pour tout  $i < k$  la relation:  $(d^i)^t \cdot \nabla f(x^k) = 0$ .

Preuve : on a en effet  $(d^i)^t \cdot \nabla f(x^k) = (d^i)^t \cdot (Ax^k + B) = (d^i)^t [A(x^i + \sum_{j=i}^{k-1} s_j d^j) + B]$   
 $= (d^i)^t \cdot (Ax^i + B) + s_i \cdot (d^i)^t \cdot A \cdot d^i$   
 $= 0$  d'après la valeur de  $s_i$  calculée ci-dessus.

Théorème: Le point  $x^n$  est l'optimum de  $f(x)$  sur  $\mathbb{R}^n$ .

Preuve: Les directions  $d^0, d^1, \dots, d^{n-1}$  étant mutuellement conjuguées, elles forment une base de  $\mathbb{R}^n$ . D'après le lemme,  $\nabla f(x^n) = 0$ , ce qui démontre le théorème.

La méthode de Fletcher et Reeves engendre au fur et à mesure les directions  $d^i$ ; on l'explique ci-dessous en posant:  $g^k = \nabla f(x^k) = Ax^k + B$ .

\* Choisir un point de départ  $x^0$ , poser  $d^0 = -g^0$ .

\* Pour  $k$  variant de 0 à  $n$  faire:

- $s_k = \frac{(d^k)^t \cdot g^k}{(d^k)^t \cdot A \cdot d^k}$
- $x^{k+1} = x^k + s_k d^k$
- $b_k = \frac{(g^{k+1})^t \cdot A \cdot d^k}{(d^k)^t \cdot A \cdot d^k}$
- $d^{k+1} = -g^{k+1} + b_k \cdot d^k$

On pourra remarquer l'égalité suivante:  $(d^k)^t \cdot g^k = - \|g^k\|^2$

L'optimum finalement est  $x^{n+1}$ .

Pour la commande prédictive, l'optimisation est faite par l'algorithme précédent, où le variable  $x$  est l'incrément de commande.

## II.4.2 La méthode de FIBONACCI

### II.4.2.1 Le cas monovariabile

Cette méthode cherche le minimum de toute fonction  $f(x)$  unimodale définie sur un intervalle  $D(x_1, x_2)$ . Elle consiste à calculer la valeur de  $f(x)$  en  $N$  points; ces derniers sont choisis de telle manière que, étant évalué la fonction à tout nouveau point, on élimine un sous intervalle (le plus large possible) de l'intervalle courant. La relation de récurrence entre des intervalles successifs est donnée par:

$$\Delta^k = \Delta^{k+1} + \Delta^{k+2} \quad (2.31)$$

Considérant  $\Delta^{N-1}$  la longueur de l'intervalle obtenu après  $N$  évaluations de la fonction ( $N$  étant donné) et définissent les nombres  $F_1, F_2, \dots, F_{N-1}$  par

$$\Delta^k = F_{N-k} \Delta^{N-1} \quad (2.32)$$

On peut écrire: 
$$\frac{\Delta^k}{\Delta^{N-1}} = \frac{\Delta^{k+1}}{\Delta^{N-1}} + \frac{\Delta^{k+2}}{\Delta^{N-1}} \quad (2.33)$$

On constate que les nombres  $F_n$  satisfont la relation récurrente suivante:

$$\begin{cases} F_n = F_{n-1} + F_{n-2} & n=3, 4, \dots, N-1 \\ F_1 = 1 \end{cases}$$

Il faut trouver la suite qui donne la meilleure réduction d'intervalle, et pour ce faire, il faut choisir  $F_2$  de telle manière que  $F_{N-1}$  soit la plus grande possible, alors  $F_2$  doit être aussi la plus grande possible. Sachant qu'on doit avoir  $F_2 \leq 2F_1$ , il convient de choisir la plus grande valeur; soit 2. Alors la séquence des nombres  $F_n$  est complètement définie par ses valeurs initiales  $F_1=1$  et  $F_2=2$ . Cette séquence est appelée la séquence de FIBONACCI.

L'application de cet algorithme est comme suit:

- On définit la précision désirée, soit le dernier intervalle à atteindre:  $\Delta^{N-1}$
- Etant donné l'intervalle initial et final on calcule le nombre d'estimation  $N$  de la fonction à partir de la relation (3.18).
- Les intervalles successifs sont calculés soit par (2.31) soit par (2.32).

#### II.4.2.2 Le cas multivariable

On présente comme cas particulier, l'optimisation d'une fonction quadratique bidimensionnel, ou on cherche à une réduction des intervalles en parallèle (il y a des cas où on fixe l'une des variables et l'optimisation unidimensionnelle sur l'autre).

Soit  $f(x,y)$  une fonction quadratique à deux variables sur un intervalle convexe:

$$D(x,y): \begin{cases} x_1 \leq x \leq x_2 \\ y_1 \leq y \leq y_2 \end{cases}$$

L'algorithme d'optimisation est le suivant:

- la détermination de  $x_3, x_4, y_3, y_4$  comme dans le cas unidimensionnelle
- Pour chaque  $x(x_3, x_4)$ , on calcule :
 
$$g(x_3) = \text{Max} \{f(x_3, y_3), f(x_3, y_4)\}$$

$$g(x_4) = \text{Max} \{f(x_4, y_3), f(x_4, y_4)\}$$
- Utilisant les évaluations précédentes, pour chaque point  $y(y_3, y_4)$  on a:
 
$$h(y_3) = \text{Max} \{f(x_3, y_3), f(x_4, y_3)\}$$

$$h(y_4) = \text{Max} \{f(x_3, y_4), f(x_4, y_4)\}$$

- On compare  $g(x_3)$  avec  $g(x_4)$ , si  $g(x_3) \geq g(x_4)$  alors la nouvelle intervalle est  $[x_1, x_4]$ , si non l'intervalle est  $[x_3, x_2]$ .
- On compare  $h(y_3)$  avec  $h(y_4)$ , si  $h(y_3) \geq h(y_4)$  alors la nouvelle intervalle est  $[y_1, y_4]$ , si non l'intervalle est  $[y_3, y_2]$ .

Ces cinq dernières procédures seront appliquées à tout nouveau intervalle jusqu'à le  $\text{Max}(g(x)) = \text{Max}(h(y)) = \text{Max}(f(x, y))$  sera localisé.

Pour la généralisation de cet algorithme au cas multidimensionnel, la réduction des intervalles se fait en parallèle et de la même manière que pour le cas bidimensionnel.

### **II.4.3 La méthode de DICHOTOMIE**

La méthode de DICHOTOMIE est un algorithme d'itération connu pour optimiser les critères sans calculer les dérivés, son principe est très facile à comprendre ou à appliquer. Cette méthode est utilisée pour trouver le point de zéro d'une fonction sur un intervalle déterminé ou pour la minimisation (maximisation) des critères avec l'apprendre en compte les contraintes.

#### **II.4.3.1 Le cas monovarié**

Soit  $f(x)$  une fonction uni modale définie sur un intervalle  $D(x_1, x_2)$ , l'algorithme est le suivant (cas de minimisation):

- Détermination de la valeur de l'intervalle minimum de séparation entre  $x_1$  et  $x_2$  que l'on accepte.
- On calcule les deux points  $x_3$  et  $x_4$ :

$$X_3 = \frac{x_1 + x_2 - \varepsilon}{2} \quad X_4 = \frac{x_1 + x_2 + \varepsilon}{2}$$

- On calcule  $f(x_3)$  et  $f(x_4)$ .
- Si  $f(x_3) \geq f(x_4)$  alors la nouvelle intervalle est  $[x_3, x_2]$ , si non l'intervalle est  $[x_1, x_4]$ .
- On répète le calcul avec le nouveau intervalle jusqu'à avoir  $x_2 - x_1 \leq \varepsilon$
- La valeur cherchée est :  $x^* = \frac{x_1 + x_2}{2}$

#### **II.4.3.2 Le cas multivarié**

Les étapes de l'optimisation multivarié avec DICOTHOMIE sont les mêmes qu'avec la méthode de FIBONACCI, sauf que  $x_3$  et  $x_4$  sont calculés avec la méthode de DICOTHOMIE unidimensionnelle.

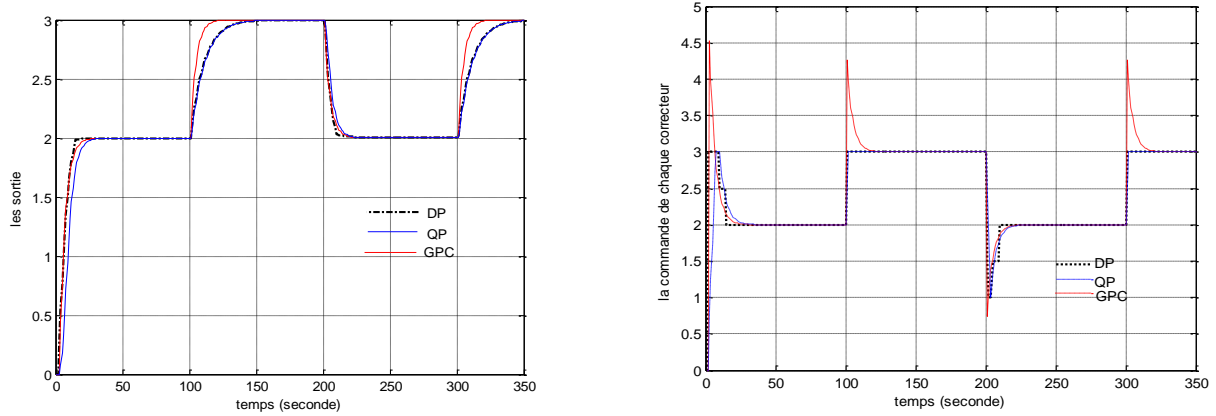
## II.5 résultat de simulation

Le système choisi est donné par l'équation en récurrence suivante :

$$(1 - 1.6333q^{-1} + 0.99q^{-2})Y(q) = q^{-1}(0.1165 - 0.1108q^{-1})U(q)$$

Les paramètres de commande choisis sont :

$$N_1 = 1 ; N_2 = 6 ; N_u = 4 ; \lambda = 0.5 ; du_{max}=0.5 ; du_{min}=0.5 ; u_{max}=3 ; u_{min}=1$$



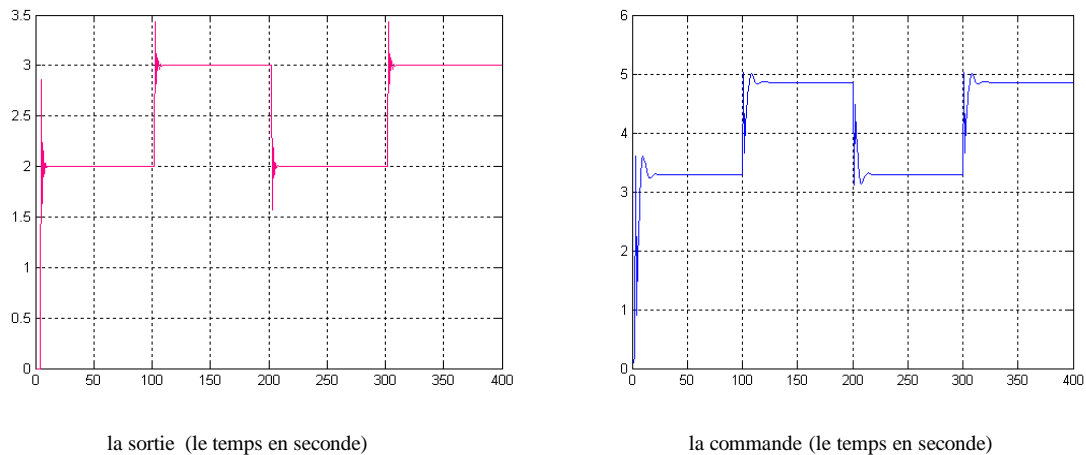
**Figure 2.1:** Résultat de la simulation de DP, QP, GPC

On a choisit un autre système pour tester les autres méthodes d'optimisation, ce système est

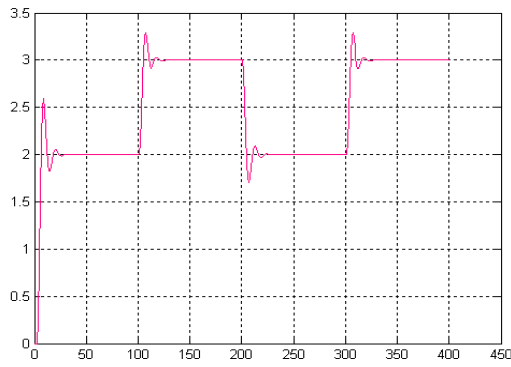
donné par:  $(1 - 0.4428q^{-1} + 0.4918q^{-2})Y(q) = q^{-1}(-0.2672 + 1.2181q^{-1})U(q)$

Les paramètres de commande choisis sont :

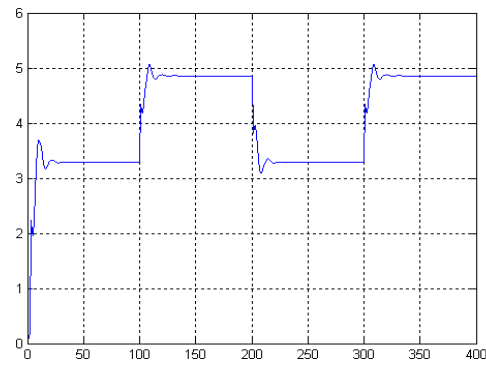
$$N_1 = 1 ; N_2 = 8 ; N_u = 4 ; \lambda = 0.5 ; du_{max}=1 ; du_{min}=0 ; u_{max}=5 ; u_{min}=3$$



**Figure 2.2:** Résultat de la simulation avec la méthode du gradient conjugué en optimisation

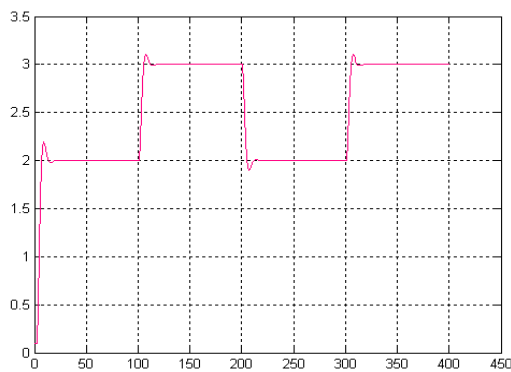


la sortie (le temps en seconde)

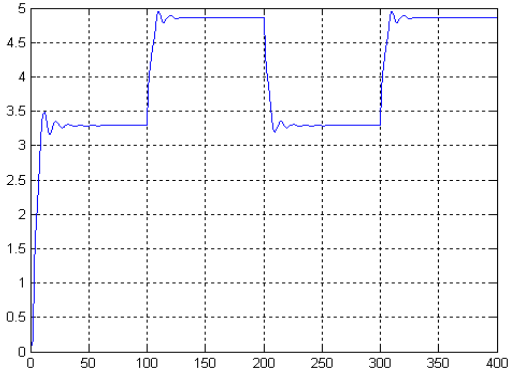


la commande (le temps en seconde)

***Figure 2.3: Résultat de la simulation avec la méthode de Dichotomie en optimisation***



la sortie (le temps en seconde)



la commande (le temps en seconde)

***Figure 2.4: Résultat de la simulation avec la méthode de Fibonacci en optimisation***

**Commentaire :**

Ce qui est évident, la prendre en compte des contraintes dans le calcul de commande limite les plages de variation des entrées et des sorties, ce que dégrade l'efficacité et les performances de la commande. Pour la GPC, et ce que montre la figure 2.1, la dégradation est infiniment petite, on a obtenu des bonnes réponses, malgré que les contraintes soient respectées.

Pour les autres figures 2.2, 2.3, et 2.4, on a appliqué différentes méthodes dans la phase d'optimisation, on peut dire que ces méthodes sont applicables dans la GPC avec des bonnes réponses.



### III.1 Introduction

L'automatique a souvent recours à des cas d'étude particuliers, qui sont représentatifs de grande classe d'application, et dont le caractère spectaculaire est conforme.

De plus avec l'expérience, la connaissance de ces cas est affinée et ils fournissent aujourd'hui une base idéale pour comparer d'une façon valable les avantages et les inconvénients d'approche différente. Le pendule inversé est un de ces cas types.

Le pendule inversé est composé d'une barre mobile en rotation autour d'un axe horizontale solidaire d'un chariot qui peut se déplacer le long d'un rail, ce système est multivariable, non linéaire, et instable en boucle ouverte.

L'étude du pendule inversé a assez d'importance :

- La robotique utilise ce genre de concept.
- Les étudiants en automatique et robotique font des labos avec ce genre de dispositif pour approfondir leurs connaissances dans les domaines suivants : la programmation, l'automatisation, la mécanique....

### III.2 Modèle flou de type Takagi-Sugeno

Identifier un processus, c'est chercher un modèle mathématique, appartenant à une classe de modèle connue, est qui donne une réponse la plus proche possible du système réel.

Parfois, il est difficile d'incorporer les informations disponibles sur le système, dans un modèle paramétrique standard, en particulier lorsqu'on cherche à identifier des modèles non linéaires.

L'utilisation de la logique floue permet l'identification dans ces cas des paramètres qu'il n'était pas possible de prendre en compte avec l'utilisation d'un modèle paramétrique classique.

Le modèle flou est une structure qui utilise des règles de la forme « **if...then ...** » pour établir des relations entre les variables du procédé.

Deux types de modèles flous sont utilisés, le premier est celui de Mamdani et le deuxième est celui de **Takagi-Sugeno**.

#### III.2.1 Le modèle de type Mamdani

Le modèle de **Mamdani** est basé sur des règles linguistiques telle que ' **If** la tension est élevée **Then** la vitesse croit rapidement'.

Ce genre de règles décrit le procédé d'une manière qualitative. Les ensembles flous représentent une interface entre les variables qualitatives dans les règles (exemple : Tension élevée) et les domaines numériques des entrées/sorties du modèle. La conception de ce genre de modèle nécessite une connaissance de priori sur le procédé.

### III.2.2 Le modèle de type Takagi-Sugeno

Dans le modèle **Takagi-Sugeno**, chaque règle représente un modèle linéaire local, par exemple

$$R_i : \text{If } x_1 \text{ is } F_1^i \text{ et } x_2 \text{ is } F_2^i \text{ Then } y = h_i(x)$$

Les fonctions de sortie  $h_i(x)$  peuvent être en principe des fonction arbitraires des entrées, mais d'une manière générale elles sont choisies telles quelles soient une combinaison linéaire des entrées.

$$h_i(x) = a_{i1}x_1 + a_{i2}x_2 + \dots + a_{in}x_n \quad (3.1)$$

Ce type de système est appelé un modèle Takagi-Sugeno linéaire, ou modèle Takagi-Sugeno homogène, ou encore si un terme constant est ajouté, on parle de modèle Takagi-Sugeno affine.

La particularité de modèle T-S est que la logique floue est seulement utilisée dans la partie prémisse des règles. La partie conclusion est décrite par des valeurs numériques.

Les modèles flous T-S sont habituellement représentés sous les formes suivantes :

#### III.2.2.1 modèle flou continu

Règle  $i$  de procédé :

Si  $z_1(t)$  est  $F_1^i$  et  $z_2(t)$  est  $F_2^i$  ... et  $z_p(t)$  est  $F_p^i$

$$\text{Alors } \begin{cases} \dot{x}(t) = A_i x(t) + B_i u(t) \\ y(t) = C_i x(t) \end{cases} \quad i=1, 2, \dots, r \quad (3.2)$$

Chaque équation de sortie est nommée sous modèle, à chaque règle  $R_i$  est attribué un poids  $w_i(z(t))$  qui dépend du degré d'appartenance des variables de prémisses  $z_j(t)$  aux sous ensembles  $F_j^i$ , et du choix du connecteur 'ET' des prémisses.

$$w_i(z(t)) = \prod_{j=1}^p F_j^i(z(t)) \quad (3.3)$$

$i=1, 2, \dots, r$   $w_i(z(t))$  car les fonctions d'appartenance prennent leurs valeurs dans  $[0,1]$ .

Les sorties finales du système flou inférées sont donnée par les équations suivantes :

$$\dot{x}(t) = \frac{\sum_{i=1}^r w_i(z(t)) \{A_i x(t) + B_i u(t)\}}{\sum_{i=1}^r w_i(z(t))} \quad (3.4)$$

$$y(t) = \frac{\sum_{i=1}^r w_i(z(t)) C_i x(t)}{\sum_{i=1}^r w_i(z(t))} \quad (3.5)$$

En posant  $h_i(z(t)) = \frac{w_i(z(t))}{\sum_{i=1}^r w_i(z(t))}$  avec  $\forall t, \sum_{i=1}^r h_i(z(t)) = 1$ , elles peuvent être réécrites :

$$\dot{x}(t) = \sum_{i=1}^r h_i(z(t)) \{A_i x(t) + B_i u(t)\} \quad (3.6)$$

$$y(t) = \sum_{i=1}^r h_i(z(t)) C_i x(t) \quad (3.7)$$

### III.2.2.2 modèle flou discret

Si  $z_1(t)$  est  $F_1^i$  et  $z_2(t)$  est  $F_2^i \dots$  et  $z_p(t)$  est  $F_p^i$

$$\text{Alors } \begin{cases} x(t+1) = A_i x(t) + B_i u(t) \\ y(t) = C_i x(t) \end{cases} \quad i=1,2,\dots,r \quad (3.8)$$

Les sorties finales du modèle flou sont définies par :

$$x(t+1) = \frac{\sum_{i=1}^r w_i(z(t)) \{A_i x(t) + B_i u(t)\}}{\sum_{i=1}^r w_i(z(t))} \quad (3.9)$$

$$y(t) = \frac{\sum_{i=1}^r w_i(z(t)) C_i x(t)}{\sum_{i=1}^r w_i(z(t))} \quad (3.10)$$

Posant  $h_i(z(t)) = \frac{w_i(z(t))}{\sum_{i=1}^r w_i(z(t))}$  avec  $\forall t, \sum_{i=1}^r h_i(z(t)) = 1$ , elles peuvent réécrites :

$$x(t+1) = \sum_{i=1}^r h_i(z(t)) \{A_i x(t) + B_i u(t)\} \quad (3.11)$$

$$y(t) = \sum_{i=1}^r h_i(z(t)) C_i x(t) \quad (3.12)$$

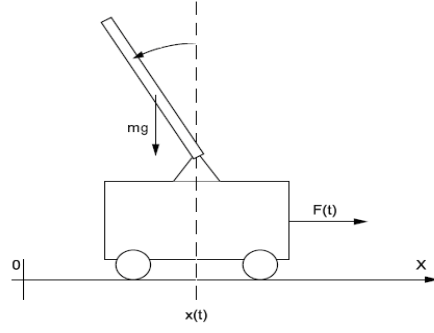
## III.3 La commande d'un pendule inversée par GPC

### III.3.1 La modélisation d'un pendule inversé

La modélisation d'un pendule inversé par la méthode de LAGRANGE est comme suit :

$L = E_c - E_p$   $L$ : est appelé lagrangien

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} + \frac{\partial D}{\partial \dot{q}} = F_q \quad (3.13)$$



**Figure 3.1:** Schéma d'un pendule inversé

$q$ : degrés de liberté,  $x_p$ : la position de chariot

$D$ : énergie dissipée par frottement

$F_q$ : force généralisée dans la direction du degré de liberté

$E_c$ : énergie cinétique

$E_p$ : énergie proportionnelle

L'énergie cinétique de l'ensemble est :

$$E_c = \frac{1}{2} m_p v^2 + \frac{1}{2} J \dot{\theta}^2 + \frac{1}{2} m_c \dot{x}_p^2 \quad (3.14)$$

Où  $v$  représente le vecteur de vitesse du centre de gravité de la barre dont les coordonnées sont :

$$\begin{pmatrix} x_p + \frac{l}{2} \sin \theta \\ \frac{l}{2} \cos \theta \end{pmatrix}, \text{ d'où le vecteur de vitesse : } \begin{pmatrix} \dot{x}_p + \frac{l}{2} \dot{\theta} \cos \theta \\ -\frac{l}{2} \dot{\theta} \sin \theta \end{pmatrix}$$

L'énergie cinétique prend alors l'expression suivante après un petit calcul :

$$E_c = \frac{1}{2} m_p \left( \dot{x}_p^2 + \frac{l^2 \dot{\theta}^2}{4} + \dot{x}_p l \dot{\theta} \cos \theta \right) + \frac{1}{2} J \dot{\theta}^2 + \frac{1}{2} m_c \dot{x}_p^2 \quad (3.15)$$

L'énergie potentielle est :

$$E_p = \frac{m_p g l}{2} \cos \theta \quad (3.16)$$

Donc le lagrangien est :

$$L = \frac{1}{2} m_p \left( \dot{x}_p^2 + \frac{l^2 \dot{\theta}^2}{4} + \dot{x}_p l \dot{\theta} \cos \theta \right) + \frac{1}{2} J \dot{\theta}^2 + \frac{1}{2} m_c \dot{x}_p^2 - \frac{m_p g l}{2} \cos \theta \quad (3.17)$$

Pour le degré de liberté  $q_1(t) = x_p(t)$ , l'équation de Lagrange s'écrit comme suit :

$$(m_p + m_c) \ddot{x}_p + m_p l (\ddot{\theta} \cos \theta - \dot{\theta}^2 \sin \theta) = F(t) \quad (3.18)$$

Et pour le degré de liberté  $q_2(t) = \theta(t)$ , on a :

$$j_T \ddot{\theta} + m_p l_p \ddot{x}_p \cos \theta - m_p g l_p \sin \theta = 0 \quad (3.19)$$

### III.3.1.1 Modèle linéarisé du système

Dans ce modèle, les petites variations autour de l'angle  $\theta=0$  sont considérées, en remplaçant les termes non linéaires en sin et cos par leur développement limité au premier ordre :

$\cos \theta \approx 1$  et  $\sin \theta \approx \theta$

Après simplification, on a :

$$(m_p + m_c)\ddot{x}_p + \frac{1}{2}m_p l\ddot{\theta} = F \quad (3.20)$$

$$\frac{m_p l^2}{3}\ddot{\theta} + \frac{1}{2}m_p l\ddot{x}_p - \frac{1}{2}m_p g l\theta = 0 \quad (3.21)$$

Le système est alors :

$$\ddot{x}_p = -\frac{m_p}{2(m_c + m_p)}l\ddot{\theta} + \frac{1}{m_c + m_p}F \quad (3.22)$$

$$\ddot{\theta} = \frac{3g}{2l}\theta - \frac{3}{2l}\ddot{x}_p \quad (3.23)$$

On sait que l'ensemble du système considéré peut être décrit par les variables d'états, ces variables sont la position, l'angle et la vitesse angulaire de pendule, et la vitesse de chariot, pour cela on définit le vecteur d'état suivant et sa dérivé :

$$X(t) = \begin{pmatrix} \dot{\theta}(t) \\ \dot{x}_p(t) \\ \theta(t) \\ x_p(t) \end{pmatrix} \quad \dot{X}(t) = \begin{pmatrix} \ddot{\theta}(t) \\ \ddot{x}_p(t) \\ \dot{\theta}(t) \\ \dot{x}_p(t) \end{pmatrix}$$

On prend compte que la force  $F$  appliquée sur l'ensemble est bien considérée comme une commande  $U$  du système.

Et d'après les équations (3.12) et (3.8) on obtient :

$$\ddot{\theta} = \frac{6(m_p + m_c)}{(4m_c + m_p)l}g\theta - \frac{6}{(4m_c + m_p)l}U \quad (3.24)$$

En introduisant l'équation (3.14) dans l'équation (3.12) on trouve :

$$\ddot{x}_p = \frac{-3m_p}{4m_c + m_p}g\theta + \frac{4}{4m_c + m_p}U \quad (3.25)$$

Enfin, on obtient la description suivante :

$$\dot{X}(t) = A.X(t) + B.U(t)$$

Avec:  $A = \begin{pmatrix} 0 & 0 & \alpha 1 & 0 \\ 0 & 0 & \beta 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad B = \begin{pmatrix} \alpha 2 \\ \beta 2 \\ 0 \\ 0 \end{pmatrix}$  A et B sont deux matrices

$$\alpha 1 = \frac{6g(m_p + m_c)}{(4m_c + m_p)l} \quad \alpha 2 = \frac{-6}{(4m_c + m_p)l} \quad \beta 1 = \frac{-3m_p}{4m_c + m_p}g \quad \beta 2 = \frac{4}{4m_c + m_p}$$

### III.3.1.2 Modèle flou T-S

Le choix des variables linguistiques floues et les fonctions d'appartenance représentent la structure d'identification, leur conception est basée sur les connaissances acquises par des opérateurs experts, ou par des méthodes systématiques par exemple l'algorithme **Subtractive-Clustering** (qui est un algorithme disponible dans le toolbox de **MATLAB** pour extraire les formes des fonctions d'appartenances, leurs paramètres et le nombre de règles, ainsi les paramètres des sous modèles flous).

Autre méthode, est qui la plus utilisée, est d'utiliser une transformation sur des fonctions à une variable. Cette méthode est résumée par le lemme suivant :

Lemme: pour  $x \in [-b a]$ ,  $a > 0$ ,  $b > 0$ , soit  $f(x)$ ,  $R \rightarrow R$  une fonction bornée, alors il existe toujours deux fonctions  $h_1(x)$ ,  $R \rightarrow R$ ,  $h_2(x)$ ,  $R \rightarrow R$ ,  $\alpha$  et  $\beta$  deux scalaires et vérifiant les propriétés suivantes:  $h_1(x) + h_2(x) = 1$ ,  $h_1(x) \geq 0$ ,  $h_2(x) \geq 0$ , et:  $f(x) = h_1(x) \alpha + h_2(x) \beta$ .

Preuve: elle est évident en considérant que  $f_{min} \leq f(x) \leq f_{max}$  sur l'intervalle considéré, on peut toujours écrit :

$$f(x) = h_1(x) \alpha + h_2(x) \beta \text{ avec } h_1(x) = \frac{f(x) - f_{min}}{f_{max} - f_{min}}, h_2(x) = \frac{f_{max} - f(x)}{f_{max} - f_{min}}, \alpha = f_{max}, \beta = f_{min}$$

Habituellement,  $f(x)$  prend les deux termes les plus utilisés suivantes :  $\cos(x)$ ,  $\frac{\sin(x)}{x}$ .

(Exemple détaillé en Annexe).

Pour le pendule inversé, et d'après les deus relations (3.8) et (3.9), on peut écrire :

$$\ddot{\theta} = \frac{m_p}{2A} gl \sin \theta - \frac{m_p}{2Am_t} lu - \frac{m_p^2}{2Am_t} l^2 \dot{\theta}^2 \sin \theta \cos \theta \quad (3.26)$$

$$\ddot{x}_p = \frac{1}{m_t} u + \frac{m_p}{m_t} l \dot{\theta}^2 \sin \theta - \frac{m_p^2}{2Am_t} l \cos \theta \left[ gl \sin \theta - \frac{l}{m_t} u - \frac{m_p}{m_t} l^2 \dot{\theta}^2 \sin \theta \cos \theta \right] \quad (3.27)$$

$$A = j_T - \frac{m_p^2}{2m_t^2} l^2 \cos^2 \theta \quad (3.28)$$

Ces dernières équations présentent des termes non linéaires :

$\sin \theta$ ,  $\cos \theta$ ,  $\cos \theta \sin \theta$ ,  $\cos^2 \theta$ ,  $\dot{\theta}^2 \sin \theta \cos^2 \theta$ ,  $\dot{\theta}^2 \cos \theta \sin \theta$ ,  $\dot{\theta}^2 \sin \theta$  qui sont tous limités, donc il est possible de construire le modèle flou, il sera composé de  $2^6 = 64$  règles, il est alors impératif de voir la possibilité de réaliser une réduction de ce nombre de règles.

Dans le cas général, il existe plusieurs considérations qu'avec on peut réduire le nombre de règle, et dans le cas d'un pendule inversé :

- $\sin \theta = \cos(\pi/2 + \theta)$ , donc on peut dire que le deux termes  $\sin \theta$ ,  $\cos \theta$  ne sont qu'un seule terme non linéaire.

- Le même chose pour  $\sin \theta \cos \theta$ , puisque:  $\sin \theta \cos \theta = 0.5 \sin 2\theta = 0.5 \cos(\pi/2 + 2\theta)$ .
- Le principe de séparation entre le calcul des gains d'un observateur flou et d'une loi de commande oblige que les variables de promesses n'utilisent que les variables mesurables, ce principe permet l'élimination des termes:  $\dot{\theta}^2 \sin \theta$ ,  $\dot{\theta}^2 \sin \theta \cos^2 \theta$ ,  $\dot{\theta}^2 \cos \theta \sin \theta$ .

**Remarques**

- 1- Pour les deux premiers considérations, il y a seulement changement de variable, ce qui donne un changement des fonctions  $h_1(\theta)$  et  $h_2(\theta)$ , mais les paramètres  $\alpha$  et  $\beta$  sont restés les mêmes.
- 2- Un autre cas par exemple pour le terme  $\cos^2 \theta$ , il est possible d'utiliser la même transformation avec  $\cos \theta$ , puisque et pour  $\theta \in [-\theta_0, \theta_0]$

$$\cos \theta = \frac{\cos \theta - \cos \theta_0}{1 - \cos \theta_0} \cdot 1 + \frac{1 - \cos \theta}{1 - \cos \theta_0} \cos \theta_0, \quad \cos^2 \theta = \frac{\cos^2 \theta - \cos^2 \theta_0}{1 - \cos^2 \theta_0} \cdot 1 + \frac{1 - \cos^2 \theta}{1 - \cos^2 \theta_0} \cos^2 \theta_0$$

$$\left| \frac{\cos^2 \theta - \cos^2 \theta_0}{1 - \cos^2 \theta_0} - \frac{\cos \theta - \cos \theta_0}{1 - \cos \theta_0} \right| = p\%$$

Pour  $\theta = 30^\circ$ ,  $p\% = 1.8\%$ , et pour  $\theta = 50^\circ$ ,  $p\% = 2.4\%$ .

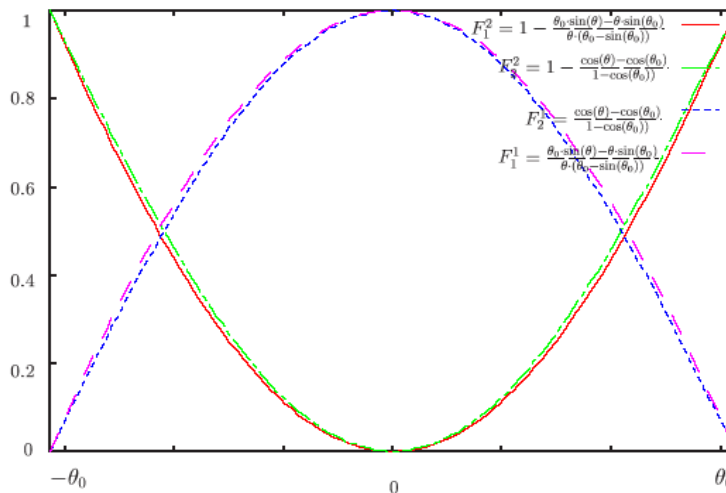
Dans le cas général, on peut éliminer ce terme si le terme (cos) est existante.

Finalement, le modèle flou d'un pendule inversé est composé de  $2^2 = 4$  règles, le vecteur des prémisses est alors  $z(t) = [\theta(t) \quad \cos(\theta(t))]$ ,

et avec  $f(\theta, \theta_0) = \frac{\theta_0 \sin \theta - \theta \sin \theta_0}{\theta(\theta_0 - \sin \theta_0)}$ , les fonctions d'appartenance sont :

$$F_1^1(z_1(t)) = f(\theta(t), \theta_0), \quad F_1^2(z_1(t)) = 1 - f(\theta(t), \theta_0)$$

$$F_2^1(z_2(t)) = f(\cos(\theta(t)), \cos(\theta_0)), \quad F_2^2(z_2(t)) = 1 - f(\cos(\theta(t)), \cos(\theta_0))$$



**Figure 3.2: Fonction d'appartenance des ensembles flous**

Le modèle flou complet est décrit par les règles suivantes :

Règle 1: Si  $\theta$  est  $F_1^1$  et  $\cos \theta$  est  $F_2^1$  alors :

$$\dot{X}(t) = \begin{pmatrix} 0 & 0 & \frac{m_p}{2A'} gl & 0 \\ 0 & 0 & -\frac{m_p^2 l^2 g}{2A' m_t} & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} X(t) + \begin{pmatrix} -\frac{m_p l}{2A' m_t} \\ \frac{1}{m_t} + \frac{m_p^2 l^2}{2A' m_t^2} \\ 0 \\ 0 \end{pmatrix} U(t)$$

Règle 2: Si  $\theta$  est  $F_1^1$  et  $\cos \theta$  est  $F_2^2$  alors :

$$\dot{X}(t) = \begin{pmatrix} 0 & 0 & \frac{m_p gl \sin \theta_0}{2A' \theta_0} & 0 \\ 0 & 0 & -\frac{m_p^2 l^2 g \sin \theta_0}{2A' m_t \theta_0} & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} X(t) + \begin{pmatrix} -\frac{m_p l}{2A' m_t} \\ \frac{1}{m_t} + \frac{m_p^2 l^2}{2A' m_t^2} \\ 0 \\ 0 \end{pmatrix} U(t)$$

Règle 3: Si  $\theta$  est  $F_1^2$  et  $\cos \theta$  est  $F_2^1$  alors :

$$\dot{X}(t) = \begin{pmatrix} 0 & 0 & \frac{m_p}{2A'} gl & 0 \\ 0 & 0 & -\frac{m_p^2 l^2 g}{2A' m_t} \cos \theta_0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} X(t) + \begin{pmatrix} -\frac{m_p l}{2A' m_t} \\ \frac{1}{m_t} + \frac{m_p^2 l^2}{2A' m_t^2} \cos \theta_0 \\ 0 \\ 0 \end{pmatrix} U(t)$$

Règle 4: Si  $\theta$  est  $F_1^2$  et  $\cos \theta$  est  $F_2^2$  alors :

$$\dot{X}(t) = \begin{pmatrix} 0 & 0 & \frac{m_p gl \sin \theta_0}{2A' \theta_0} & 0 \\ 0 & 0 & -\frac{m_p^2 l^2 g \sin \theta_0}{2A' m_t \theta_0} \cos \theta_0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} X(t) + \begin{pmatrix} -\frac{m_p l}{2A' m_t} \\ \frac{1}{m_t} + \frac{m_p^2 l^2}{2A' m_t^2} \cos \theta_0 \\ 0 \\ 0 \end{pmatrix} U(t)$$

Avec:  $A' = j_T - \frac{m_p^2}{2m_t} l^2$ ,  $j_T = \frac{m_p l^2}{3}$



### III.3.2 L'algorithme de GPC sur une représentation d'état

Soit le système décrit par sa représentation d'état discrète suivante:

$$\begin{cases} X(k+1) = AX(k) + BU(k) \\ Y(k) = C(k)X(k) \end{cases} \quad (3.29)$$

Cette représentation peut être réécrite en remplaçant la commande  $U(t)$  par l'incrément de commande  $\Delta U(k)$ , et en ajoutant la commande précédente  $U(k-1)$ .

Pour pouvoir minimiser la critère qui donne la commande optimale, il faut calculer la sortie prédite alors on peut écrit :

$$\begin{aligned} \tilde{Y}(k+1) &= C(AX(k) + B\Delta U(k)) \\ \tilde{Y}(k+2) &= C(A^2X(k) + AB\Delta U(k)) \\ \tilde{Y}(k+3) &= C(A^3X(k) + A^2B\Delta U(k) + AB\Delta U(k+1)) \\ &\dots \end{aligned}$$

On calcule  $N_2$  valeur de  $Y$ , et pour simplifier la représentation, on utilise les vecteurs. La sortie prédite s'écrit sous la forme :

$$Y = F.X(k) + H. \Delta U(k) \quad (3.30)$$

$$\begin{aligned} \text{Avec : } Y &= [\tilde{Y}(k+N_1) \dots \tilde{Y}(k+N_2)] \\ \Delta U(k) &= [\Delta U(k) \dots \Delta U(k+N_u)] \end{aligned}$$

$$F = \begin{bmatrix} C.A \\ C.A^2 \\ \dots \\ C.A^{N_2} \end{bmatrix}, \quad H = \begin{bmatrix} h_{N_1-1, N_1-1} \dots h_{N_1-1, N_1-N_u} \\ \dots \\ h_{N_2-1, N_2-1} \dots h_{N_2-1, N_2-N_u} \end{bmatrix}, \quad h_{i,j} = \begin{cases} C \left[ \prod_i^{i-j+1} A \right] B(i-j) : j > 0 \\ C.B(i-j) : j = 0 \\ 0 : j < 0 \end{cases}$$

On remplace la sortie dans le critère à minimiser par son expression, on dérive et annule la dérivée, la solution optimale sera:

$$\Delta U = (H^T.H + \lambda I)^{-1} H^T (w - F.X(k)) \quad (3.31)$$

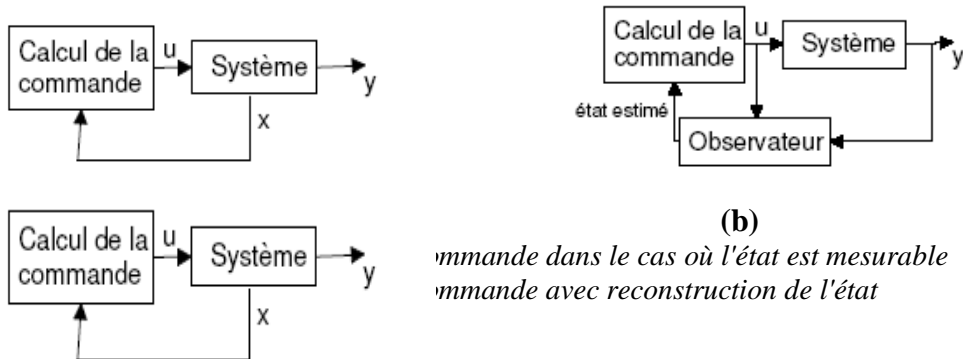
On peut voir que la matrice  $H^T.H + \lambda I$  est symétrique et définit positive quelque soit la valeur de  $\lambda$ , ceci implique que l'inverse existe toujours et la solution optimale est un minimum unique. Le premier élément de  $\Delta U$  est utilisé pour alimenter le système, puis toutes les étapes précédentes sont répétées.

### III.4 calcul d'observateur flou

Une bonne maîtrise d'un procédé passe en général par une bonne information sur ce procédé. Les variables directement mesurées ne couvrant généralement pas la totalité des grandeurs susceptibles de décrire le comportement du procédé (les *états*), on peut se poser le problème de

reconstruction de l'information non directement mesurée au moyen de celle disponible : c'est le rôle de l'*observateur*, ou *estimateur* d'état.

### III.4.1 Le principe de fonctionnement d'un observateur



commande dans le cas où l'état est mesurable  
commande avec reconstruction de l'état

Le procédé étant modélisé comme un système dynamique soumis à l'action de grandeurs externes (*entrées*) faisant varier un ensemble de grandeurs mesurées (*sorties*), l'observateur consiste en un système dynamique auxiliaire dont les entrées sont les entrées/sorties mesurées du procédé, et les sorties sont supposées donner une estimation de son état interne, selon le schéma décrit dans la figure 3.3. De cette façon, l'observateur peut se substituer à un certain nombre de capteurs physiques (il devient alors un *capteur logiciel* ou *capteur virtuel*).

### III.4.2 calcul du gain de l'observateur

On distingue deux types d'observateur, suivant que la mesure est disponible à l'instant précédent l'estimation ou à l'instant de l'estimation. Un observateur prédictif donne une estimation de l'état à l'instant  $k+1$  d'après ce que l'on connaît à l'instant  $k$ , on note  $\hat{X}(k+1)/k$  cette prédiction. Le deuxième type, et qui est utilisé dans la commande prédictive, est l'observateur correcteur, l'estimation de l'état sera alors notée  $\hat{X}(k+1)/k+1$ .

La structure de l'observateur du système décrit par l'équation (3.18) est donnée par :

$$\begin{cases} \hat{X}(k+1) = A\hat{X}(k) + BU(k) + L(Y(k) - \bar{Y}(k)) \\ \hat{Y}(k) = C(k)\hat{X}(k) \end{cases} \quad (3.32)$$

$L$ : appelé gain de l'observateur.

$\bar{Y}(k)$ : représente l'estimation de la sortie à l'instant  $k$ .

Cette structure peut être écrite sous la forme :

$$\hat{X}(k+1)/k+1 = A\hat{X}(k)/k + BU(k) + L(Y(k+1) - \bar{Y}(k+1)) \quad (3.33)$$

Si on suppose connue  $\hat{X}(k+1)/k+1$ , on pose dans ce cas :

$$\bar{Y}(k+1) = C\hat{X}(k+1)/k+1 \quad (3.34)$$

Ce qui conduit à écrire l'observateur, sous la forme :

$$\hat{X}(k+1)/k+1 = [I + LC]^{-1}[A\hat{X}(k)/k + BU(k) + LY(k+1)] \quad (3.35)$$

L'erreur de reconstruction,  $\tilde{X}(k+1)/k+1 = X(k+1) - \hat{X}(k+1)/k+1$  est définie par l'équation récurrente :

$$\tilde{X}(k+1)/k+1 = [I + LC]^{-1}A\tilde{X}(k)/k \quad (3.36)$$

Pour un bon fonctionnement de l'observateur,  $L$  doit être choisie telle que  $[I + LC]^{-1}A$  soit une matrice de Schur.

Pour les modèles linéaires, le gain de l'observateur est trouvé directement par l'instruction **OBS** qui est disponible dans le Toolbox de MATLAB, cet observateur est dit observateur linéaire.

Pour les modèles flous, on peut utiliser la même instruction pour chaque modèle flou comme on peut conditionner le gain de l'observateur avec l'état du système, on calcule l'observateur flou.

### III.5 test et simulation

Les objectifs de la commande d'un tel système (le pendule inversé) sont de réguler l'angle  $\theta$  autour de zéro (point d'équilibre de pendule) et d'asservir la position de chariot à une consigne désirée, et cela se fait pour le modèle linéarisé et le modèle T-S avec le calcul d'un observateur linéaire.

On résume les paramètres de commande, les paramètres du pendule inversé dans le tableau suivant :

Paramètre	Valeur
$m_p$	2 kg
$m_c$	0.25 kg
$l_p=l/2$	0.1 m
$g$	9.8 m/s <sup>2</sup>
$N_1$	1
$N_u$	4
$N_2$	12
$\lambda$	0.05

On choisit comme références 0.3 m pour la position et 0 rad. Les sorties sont présentées par rapport au temps (en seconde).

### III.5.1 Simulation du modèle linéarisé

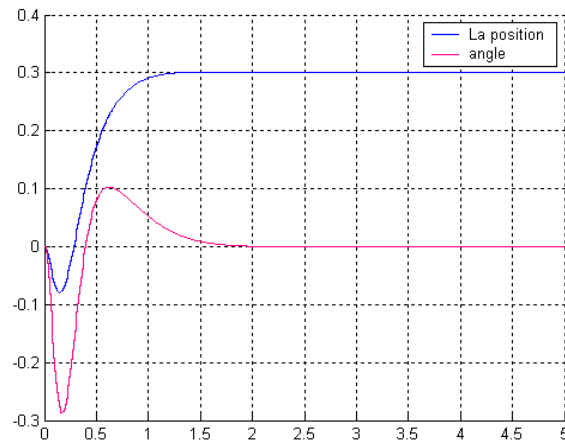


Figure 3.4: Le vecteur de sortie (la position et l'angle du pendule) sans prendre en compte des contraintes

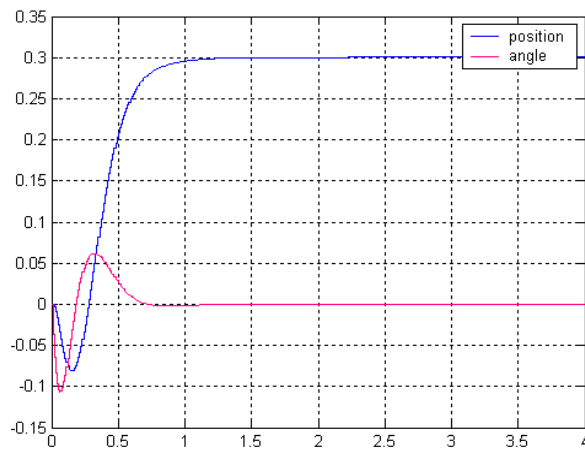


Figure 3.5: Le vecteur de sortie avec contrainte sur l'angle du pendule ( $\theta$  entre -0.2 rad et 0.2 rad)

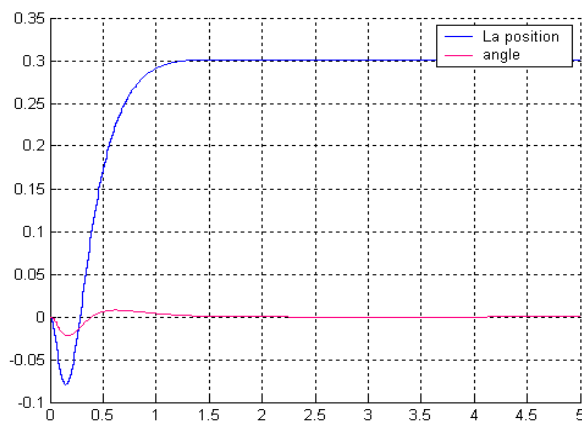
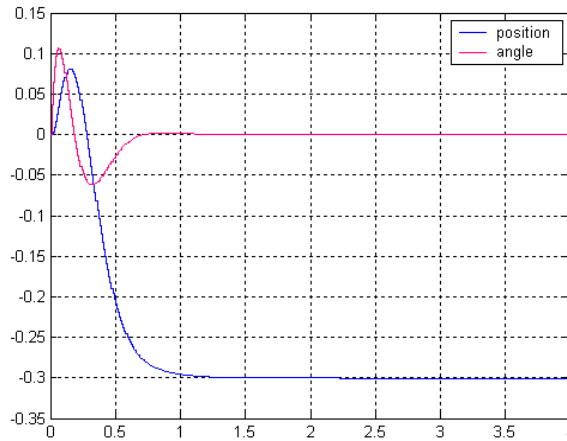


Figure 3.6: Limitation maximum de la plage de la variation de l'angle du pendule ( $\theta$  entre -0.1 rad et 0.1 rad)

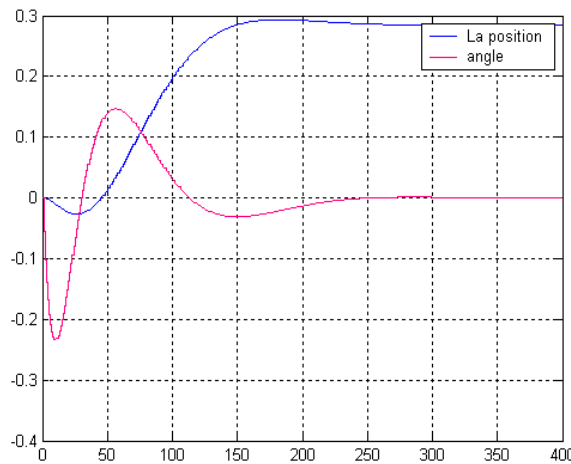


**Figure 3.7:** L'asservissement de chariot avec consigne de position négative et contrainte sur l'angle ( $\theta$  entre  $-0.2$  rad et  $0.2$  rad)

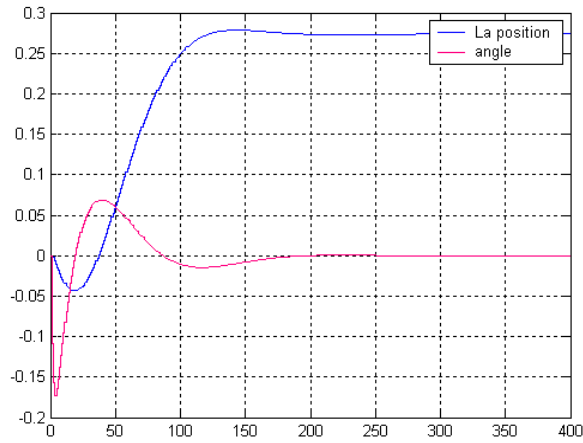
**Commentaire :**

L'erreur de l'observation d'état est considérable dans le régime transitoire, cette erreur tend vers zéro avec le temps jusqu'au régime permanent, ou on obtient une valeur très petite mais non nulle. Les quatre figures précédentes montrent que malgré la présence de l'erreur d'observation, la commande GPC reste toujours utile même pour les systèmes présentés par une représentation d'état.

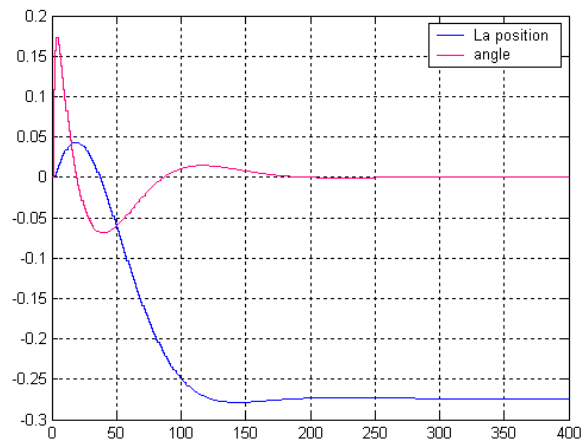
**III.5.2 Simulation du modèle T-S**



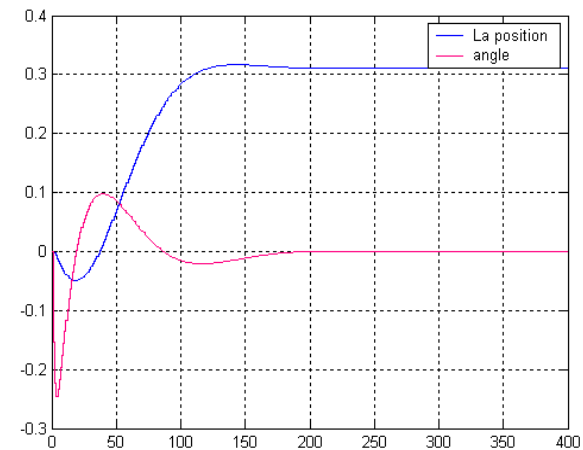
**Figure 3.8:** Le vecteur de sortie (la position et l'angle du pendule) sans prendre en compte les contraintes



**Figure 3.9:** Le vecteur de sortie avec contrainte sur l'angle du pendule ( $\theta$  entre -0.2 rad et 0.2 rad)



**Figure 3.10:** L'asservissement du chariot avec consigne de position négative et contrainte sur l'angle ( $\theta$  entre -0.2 rad et 0.2 rad)



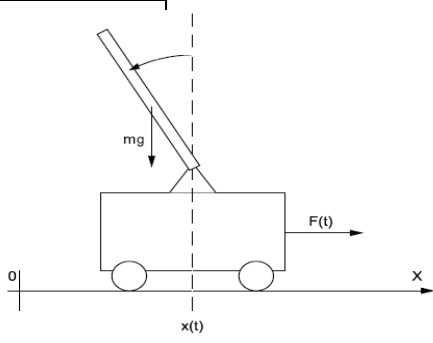
**Figure 3.11:** Le vecteur de sortie sans prendre compte des contraintes avec paramètres d'observateurs choisis

**Commentaire :**

La commande d'un modèle T-S nécessite le calcul d'un observateur flou (l'utilisation de la logique floue pour calculer à chaque exécution d'une seule boucle de commande les nouveaux paramètres de l'observateur), et cette méthode est la plus correcte pour commander ce type de système.

Pour commander le pendule inversé on a utilisé le calcul d'un autre type d'observateur, qui est l'observateur linéaire (pour chaque sous modèle flou, on calcule son propre observateur). Ce type d'observateur est encore utilisé surtout dans les systèmes complexes.

L'erreur d'observation, l'utilisation d'un observateur linéaire, et la linéarisation par logique floue, sont des éléments qui dégradent légèrement la performance de la GPC du pendule inversé.





## IV.1 Introduction

La commande en régime glissant (ou plus exactement la commande à structure variable générant un régime glissant) a essentiellement pour vocation de permettre l'obtention, en boucle fermée, d'une dynamique largement indépendante de celle du processus et surtout de ses variations éventuelles. En ce sens, elle peut être considérée comme appartenant à la classe des commandes robustes.

La commande à structure variable (CSV) est par nature une commande non linéaire. La caractéristique principale des systèmes à structure variable est que leur loi de commande se modifie d'une manière discontinue.

Les commutations de la commande s'effectuent en fonction des variables d'état, utilisées pour créer une surface dite de glissement dont le but est de forcer la dynamique du système de correspondre avec celle définie par l'équation de la surface.

Quand l'état est maintenu sur cette surface, le système est dit en régime glissant, la dynamique du système reste insensible aux variations des paramètres du processus, aux erreurs de modélisation (dans une gamme qui reste plus large par rapport à celle des approches classiques de l'automatique), et à certaines perturbations.

Ce type de commande présente plusieurs avantages tels que robustesse, précision importante, stabilité et simplicité, temps de réponse très faible. Il faut aussi noter sa très grande versatilité dans la mesure où elle permet de traiter les systèmes non linéaires et les systèmes linéaires, les systèmes multivariables et les systèmes monovariables, et aussi les problèmes de poursuite et les problèmes de régulation.

## IV.2 Théorie de la commande par mode de glissement

### IV.2.1 Systèmes à structure variable

Un système à structure variable est un système dont la structure change pendant son fonctionnement. Il est caractérisé par le choix d'une fonction et d'une logique de commutation. Ce choix permet au système de commuter d'une structure à une autre à tout instant. De plus, un tel système peut avoir des nouvelles propriétés qui n'existent pas dans chaque structure.

Dans la commande des systèmes à structure variable par mode glissant, la trajectoire d'état est amenée vers une surface. Puis à l'aide de la loi de commande, elle est obligée de rester au voisinage de cette surface. Cette dernière est appelée mouvement de glissement. La forme d'une commande à structure variable est :

$$\begin{cases} u_i = u_{\max} & \text{si } S_i(x) > 0 \\ u_i = u_{\min} & \text{si } S_i(x) < 0 \end{cases} \quad (4.1)$$

La structure de commande est caractérisée par le signe d'une fonction vectorielle  $S(x)$  appelée fonction de commutation. Cette fonction est choisie comme une fonction linéaire de l'état [15] :

$$S(x) = [S_1(x), S_2(x), \dots, S_m(x)] = Cx \text{ OÙ } C = [c_1, c_2, \dots, c_m].$$

La trajectoire dans le plan de phase est constituée de trois parties distinctes :

#### IV.2.1.1 Mode de convergence (MC)

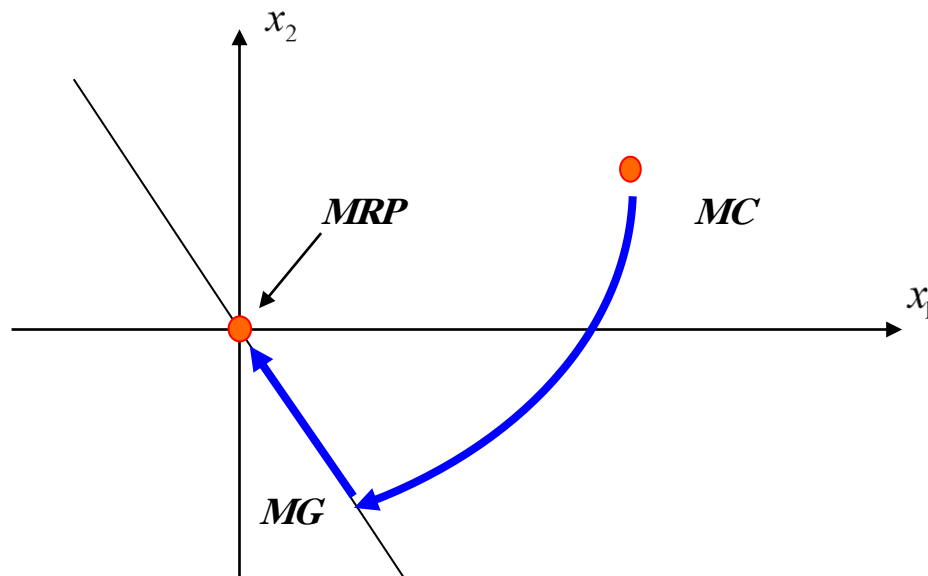
C'est le mode durant lequel la variable à régler se déplace à partir de n'importe quel point initial dans le plan de phase, et tend vers la surface de commutation  $S(x)=0$ . Ce mode est caractérisé par la loi de commande et le critère de convergence [12].

#### IV.2.1.2 Mode de glissement (MG)

C'est le mode durant lequel la variable d'état a atteint la surface de glissement et tend vers l'origine du plan de phase. La dynamique de ce mode est caractérisée par la détermination de la surface de glissement  $S(x)=0$  [12].

#### IV.2.1.3 Mode du régime permanent (MRP)

Ce mode est ajouté pour l'étude de la réponse de système, il est caractérisé par la qualité et les performances de la commande [12].



*Figure 4.1: Différents modes pour la trajectoire dans le plan de phase*

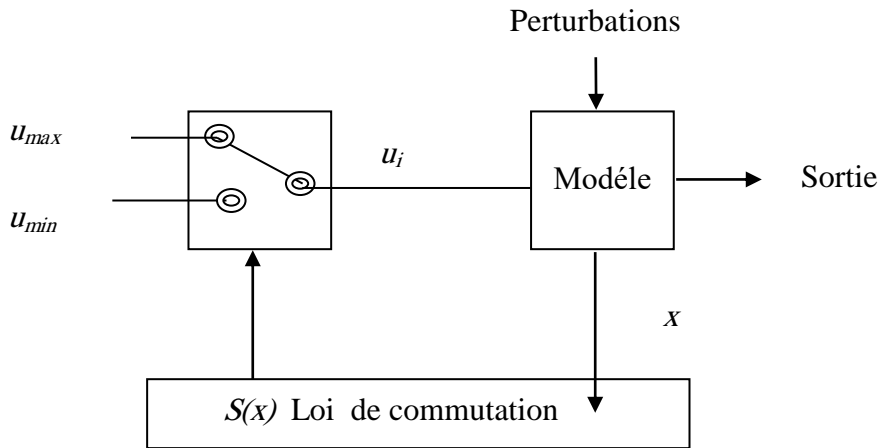
### IV.2.2 Différentes structures de la commande par mode de glissement

Dans les systèmes à structure variable utilisant la commande par mode de glissement, on peut trouver trois configurations de base pour la synthèse des différentes commandes :

#### IV.2.2.1 Structure par commutation au niveau de l'organe de commande

Le schéma d'une structure par commutation au niveau de l'organe de commande est donné par la figure 4.2. Cette structure de commande est la plus classique et la plus utilisée. Elle correspond

au fonctionnement tout ou rien des interrupteurs de puissance associée dans une grande majorité d'application aux variateurs de vitesse.

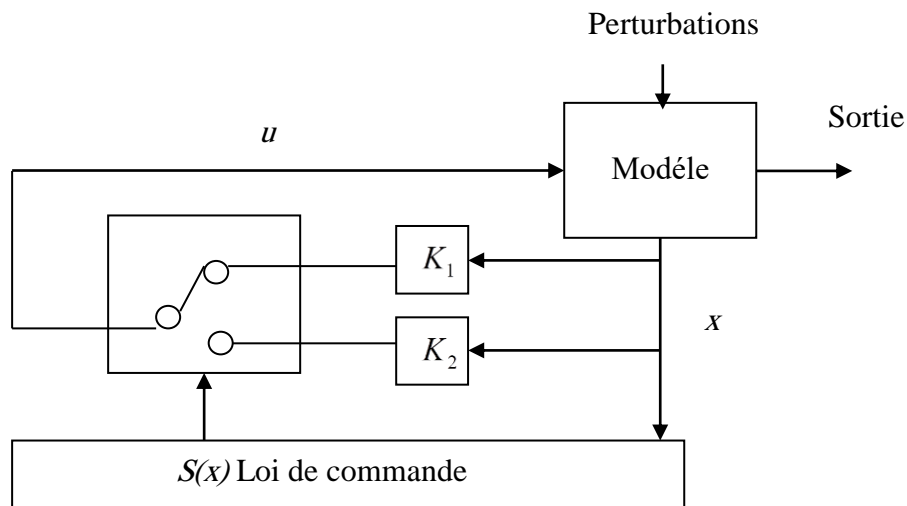


**Figure 4.2:** Structure de régulation par commutation au niveau de l'organe de commande

#### IV.2.2.2 Structure par commutation au niveau d'une contre réaction d'état

C'est la structure la moins exigeante au niveau de la sollicitation de la commande. Un ouvrage a été consacré à ce type de commande. Elle s'appuie sur la commande par contre réaction d'état classique ou le réglage de la dynamique du système est réalisée par les gains de réglage.

La non linéarité provient de la commutation entre les gains donc on crée une commutation au niveau de la dynamique du système [16].



**Figure 4.3:** Structure de régulation par commutation au niveau de la contre-réaction d'état

#### IV.2.2.3 Structure par commutation au niveau de l'organe de commande avec commande équivalente

Une telle structure présente un réel avantage. Elle permet de repositionner l'état futur du système grâce à la commande équivalente qui n'est rien d'autre que la valeur désirée du système en régime permanent. L'organe de commande est beaucoup moins sollicité, mais on est plus

dépendant des variations paramétriques du fait de l'expression de cette commande équivalente [14]

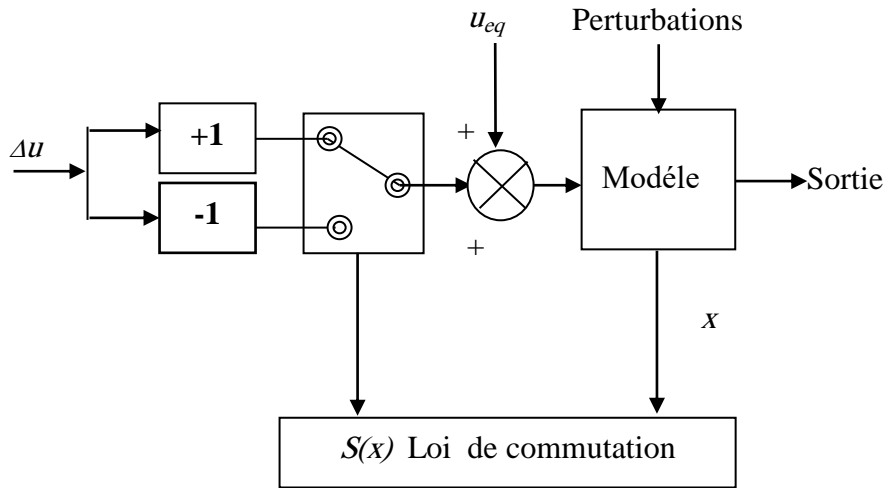


Figure .4.4: Structure de régulation par ajout de la commande équivalente

### IV.3 Description du mode glissant

#### IV.3.1 Définition du mode glissant

La technologie des modes glissants consiste à amener la trajectoire d'état d'un système vers la surface de glissement et de la faire commuter à l'aide d'une logique de commutation appropriée autour de celle-ci jusqu'à l'origine du plan d'état, d'où le phénomène de glissement.

#### IV.3.2 Explication du phénomène du glissement

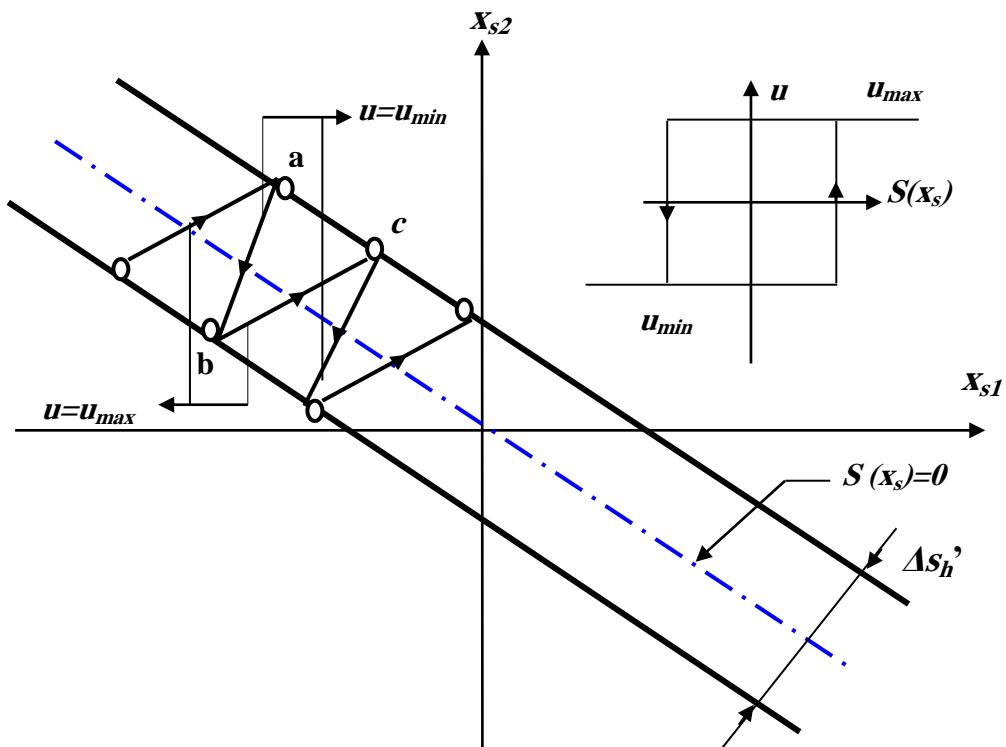


Figure 4.5: Démonstration du mode de glissement

Le mode de glissement existe lorsque les commutations ont lieu continûment entre  $u_{max}$  et  $u_{min}$ .

Pour un système à régler du deuxième ordre avec les deux grandeurs d'état  $x_{s1}$  et  $x_{s2}$ , on admet d'abord un hystérèse sur la loi de commutation  $S(x_s) = 0$  (droite en trait mixte), par conséquent, les commutations ont lieu sur les droites décalées parallèlement de  $\pm \Delta s'_h/2$ .

Une trajectoire avec  $u = u_{max}$  touche au point  $a$  le seuil de basculement inférieur, si avec  $u = u_{min}$  la trajectoire est orientée vers l'intérieur de la zone de l'hystérèse, elle touche au point  $b$  le seuil de basculement supérieur où a lieu une commutation sur  $u = u_{max}$ , si la trajectoire est de nouveau orientée vers l'intérieur, elle touchera le point  $c$  le seuil de basculement inférieure et ainsi de suit. Il y a donc un mouvement continu à l'intérieure de la zone de l'hystérèse pour conduire l'état de système vers l'origine du plan de phase.

#### **IV.4 La commande en mode glissant pour les systèmes linéaires**

Les avantages de la commande par mode glissant sont importantes et multiples, la haut précision, la bonne stabilité, la simplicité, la robustesse,...etc. Ceci lui permet d'être particulièrement adapté pour les systèmes ayant un modèle imprécis [16]. Dans ce cas, la structure de contrôleur comporte deux parties: une partie continue représentant la dynamique de système durant le mode convergence et une autre discontinue représentant la dynamique de système durant le mode de glissant, cette dernière est importante dans la commande non linéaire car elle a pour rôle d'éliminer les effets d'imprécision et des perturbations sur le modèle [17].

La conception de la commande peut être effectuée en trois étapes principales très indépendantes l'une de l'autre [17] :

- Choix de la surface.
- L'établissement des conditions d'existence.
- Détermination de la loi de commande.

##### **IV.4.1 Choix de la surface de glissement**

Pour un système linéaire la surface de glissement est une fonction scalaire telle que la variable à régler glisse sur cette surface est tend vers l'origine du plant de phase.

La forme non linéaire est une fonction de l'erreur sur la variable à régler  $x$ , elle est donné par : [17]

$$S(x) = \left( \frac{d}{dt} + \lambda \right)^{r-1} e(x) \quad (4.2)$$

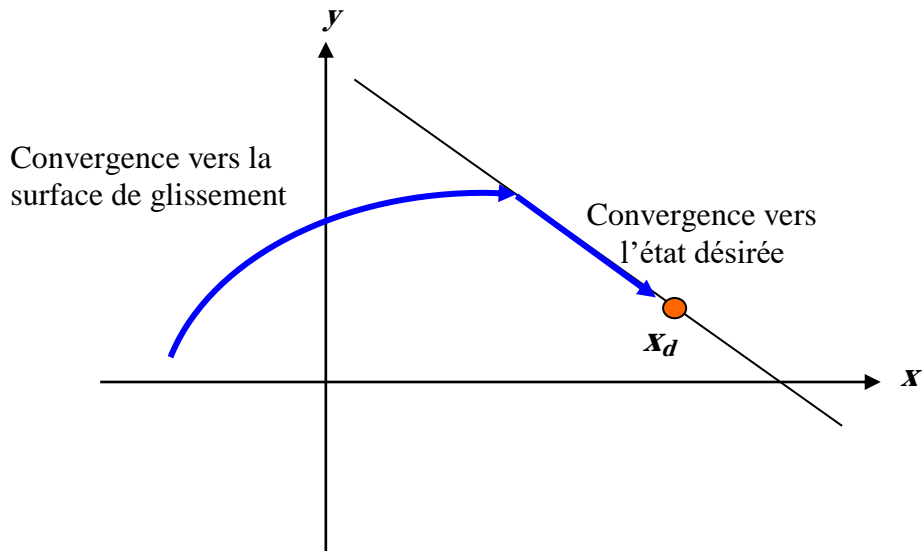
$e(x)$  : est l'écart entre la variable à régler et la référence.

$\lambda$  : est une constante positif.

$r$  : est un degré relatif, il présente le nombre de fois qu'il faut dérivée la surface pour faire apparaître la commande [15], [18].

L'objectif de la commande est maintenir la surface à zéro. Cette dernière est une équation différentielle linéaire dont l'unique solution est  $e(x) = 0$  pour un choix convenable du paramètre,

#### IV.4.2 Conditions de convergence [19]



**Figure 4.6:** Convergence du système glissant

Les conditions de convergence sont des critères qui permettent aux différentes dynamiques de converger vers la surface de glissement est d'y rester indépendamment de la perturbation. Il existe deux considérations pour assurer le mode de convergence :

##### IV.4.2.1 La fonction de commutation

C'est la première condition de convergence, elle est étudiée par EMILIANOV et UTKIN. Il s'agit de donner à la surface une dynamique convergente vers zéro. Elle est donné par :

$$\begin{cases} \dot{S}(x) > 0 & \text{si } S(x) < 0 \\ \dot{S}(x) < 0 & \text{si } S(x) > 0 \end{cases} \quad (4.3)$$

Cette condition peut être formulé comme suit :

$$\dot{S}(x)S(x) < 0 \quad (4.4)$$

##### IV.4.2.2 La Fonction de LYAPUNOV

La fonction de LYAPUNOV est une fonction scalaire positive  $V(x) > 0$  pour les variable d'état du système. La loi de commande doit être décroître cette fonction  $\dot{V}(x) < 0$ . L'idée et de choisir une fonction scalaire  $S(x)$  pour garantir l'attraction de la variable à contrôler vers sa valeur de référence et de concevoir une commande  $u$  tel que le carré de la surface correspond a une fonction de LYAPUNOV. Cette fonction de est défini comme suit [20] :

$$V(x) = \frac{1}{2} S^2(x) \quad (4.5)$$

La dérivée de cette fonction est :

$$\dot{V}(x) = S(x)\dot{S}(x) \quad (4.6)$$

Pour que la fonction  $V(x)$  puisse décroître, il suffit d'assurer que la dérivée soit négative, ceci n'est vérifié que si la condition (4.4) est vérifiée. L'équation (4.5) explique que le carré de la distance entre un point donné du plan de phase et la surface de glissement exprime par  $S^2(x)$  diminue tout le temps, contraignant la trajectoire du système à se diriger vers la surface à partir des deux côtés de cette dernière. Cette condition suppose un régime glissant idéal ou la fréquence de commutation est infinie [20].

### **IV.4.3 Calcul de la commande**

Lorsque le régime glissant est atteint, la dynamique de système est indépendante de la loi de commande qui n'a pour but de maintenir les conditions de glissement (l'attractivité de la surface), c'est pour cette raison que la surface est déterminée indépendamment de la commande. Maintenant, il reste à déterminer la commande nécessaire pour attirer la trajectoire d'état vers la surface et ensuite vers son point d'équilibre.

Cette commande est composée de deux parties, une composante discontinue qui a pour fonction essentielle de maintenir l'évolution sur la surface de glissement en dépit des variations du processus, et une composante continue dite la commande **équivalente**.

$$u = u_{eq} + u_n \quad (4.7)$$

$u_{eq}$  : la commande équivalente, proposée par FILIPOV, elle sert à maintenir **la variable** à contrôler sur la surface de glissement  $S(x) = 0$ , elle est déduite en considérant que la dérivée de surface est nulle  $\dot{S}(x) = 0$ .

$u_n$  : est déterminée pour vérifier la condition de convergence.

#### **IV.4.3.1 La commande équivalente**

Soit un système linéaire décrit par l'équation différentielle suivante :

$$\dot{x}(t) = Ax(t) + Bu(t) \quad (4.8)$$

Un vecteur commande équivalente  $u_{eq}$  se définit comme étant les équations du régime glissant idéal. On exprime la condition pour l'obtention de la commande équivalente comme :

$$\dot{S}(x,t) = \frac{\partial S}{\partial t} = \frac{\partial S}{\partial x} \cdot \frac{\partial x}{\partial t} \quad (4.9)$$

En remplaçant (4.7) et (4.8) dans (4.9) nous trouvons :

$$\dot{S}(x, t) = \frac{\partial S}{\partial x} (Ax + Bu_{eq}) + \frac{\partial S}{\partial x} Bu_n \quad (4.10)$$

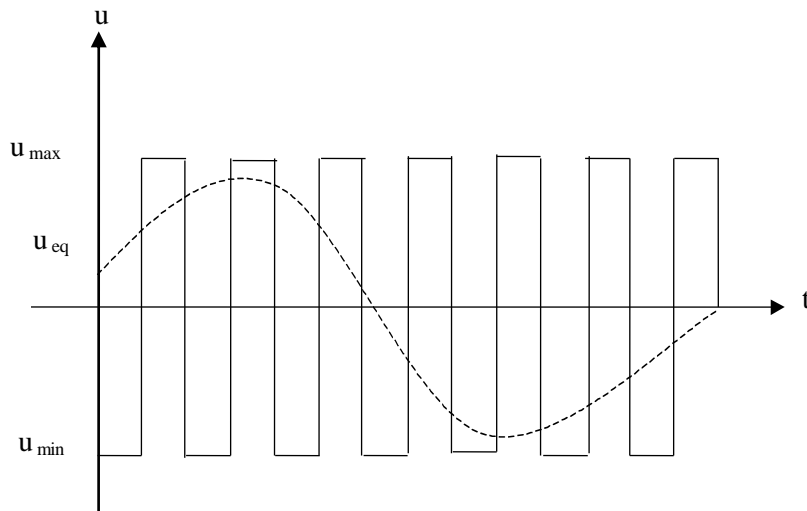
Durant du mode de glissement et le régime permanent, la surface est nulle et par conséquent sa dérivée et la partie discontinue sont aussi nulles, d'où nous déduisons l'expression de la commande équivalente :

$$u_{eq} = - \left( \frac{\partial S}{\partial x} B \right)^{-1} \frac{\partial S}{\partial x} A \quad (4.11)$$

Pour que la commande équivalente puisse prendre une valeur finie, il faut que  $\left( \frac{\partial S}{\partial x} B \neq 0 \right)$ .

La commande équivalente est interprétée physiquement comme étant une fonction continue représentant la valeur moyenne qui prend la commande lors des commutations successives entre  $u_{max}$  et  $u_{min}$  [21], d'où la condition :

$$u_{min} \leq u_{eq} \leq u_{max} \quad (4.12)$$



**Figure 4.7 :** La commande équivalente

#### IV.4.3.2 La commande **commutant**

##### IV.4.3.2.1 La commande discontinue

Plusieurs choix pour la commande discontinue  $u_n$  peuvent être faits. Le plus simple consiste à exprimer la commande discontinue  $u_n = [u_1, u_2, \dots, u_m]$  avec la fonction signe par rapport à  $S = [S_1, S_2, \dots, S_m]$ , elle est donné :

$$sign(S(x, t)) = \begin{cases} +1 & si \ S > 0 \\ 0 & si \ S = 0 \\ -1 & si \ S < 0 \end{cases} \quad (4.13)$$

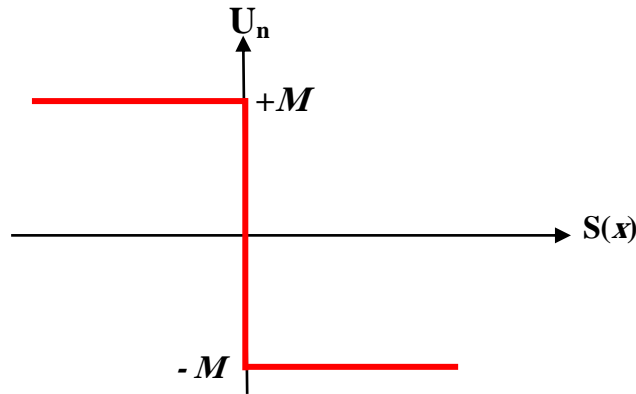


$u_n$  s'exprime donc comme :

$$u_n = -M \text{sign}(S) \quad (4.14)$$

Où  $M$  est un gain positif.

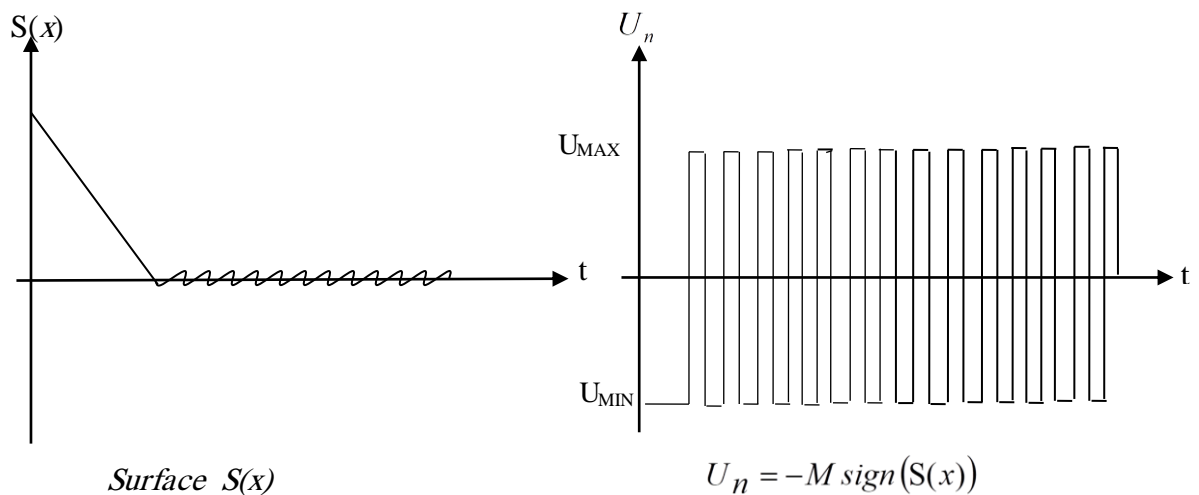
Ce premier choix de la fonction discontinue est représenté sur la figure 4.8



**Figure 4.8:** La commande discontinue

Si le gain  $M$  est très petit, le temps de réponse sera long, si  $M$  est très grand, le temps de réponse sera rapide mais des oscillations indésirables (Chattering) risquent d'apparaître sur les réponses en régime permanent. La figure 4.9 représente un cas typiques de chattering.

(on peut trouver dans des autres documents que  $u_n = M \cdot \text{sign}(S)$ ).



**Figure 4.9:** Surface de glissement et la commande discontinue

Dans le but de réduire les oscillations haute fréquence (indésirables sur les réponses), nous allons présenter quelques solutions classiques qui consiste à imposer une variation de la valeur de la commande  $u_{eq}$  en fonction de la distance entre la variable d'état et la surface de glissement.

Certaines de ces méthodes introduisent commande continue dans une bande de la surface.

**IV.4.3.2.2 Commande continue dans une bande de la surface**

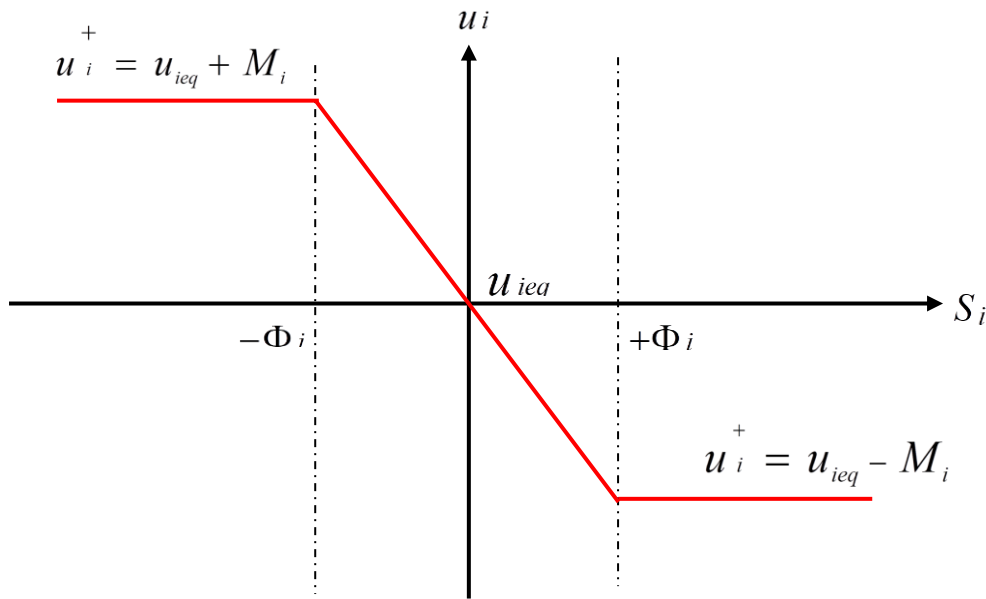
La solution proposée par Slotine [23] consiste à remplacer la loi de commande discontinue par une loi continue en  $x$  à l'extérieur d'une fine bande située au voisinage de la surface de glissement, la discontinuité étant conservée à l'extérieur de cette bande.

On choisit pour la composante haute fréquence [15] :

$$\Delta u_i = -M_i \text{sat}\left(\frac{S_i}{\Phi_i}\right) \tag{4.15}$$

Si  $\left|\frac{S_i}{\Phi_i}\right| \leq 1$  alors  $\text{sat}\left(\frac{S_i}{\Phi_i}\right) = \left(\frac{S_i}{\Phi_i}\right)$  et  $u_i = u_{ieq} - M_i \frac{S_i}{\Phi_i}$  (4.16)

Si  $\left|\frac{S_i}{\Phi_i}\right| > 1$  alors  $\text{sat}\left(\frac{S_i}{\Phi_i}\right) = \text{sign}(S_i)$  et  $u_i = u_{ieq} - M_i \text{sign}(S_i)$  (4.17)



**Figure 4.10:** Loi de commande par bande continue au voisinage de la surface de glissement selon Slotine

**IV.4.3.3 Calcul de la commande commutant**

Durant le mode de convergence et en remplaçant la commande équivalente par son expression dans (4.10) nous trouvons la nouvelle expression de la dérivée de surface.

$$\dot{S}(x) = \frac{\partial S}{\partial x} B u_n \tag{4.18}$$

La condition d'attractivité exprimée par (4.4) devient :

$$S \dot{S} = S(x) \frac{\partial S}{\partial x} B u_n < 0 \tag{4.19}$$

Afin de satisfaire cette condition, le signe de  $u_n$  doit être opposé à celui de  $S(x) \frac{\partial S}{\partial x} B$ .

La forme la plus simple que prend la commande discontinue est celle d'un relais présentée par l'équation (4.14), avec :

$$u = \begin{cases} u_n = u_{\max} & \text{si } S(x) > 0 \\ u_n = u_{\min} & \text{si } S(x) < 0 \end{cases} \quad (4.20)$$

Donc la technique par régime glissant est une loi à structure variable, qui est formée par deux composantes la partie discontinue ou glissante  $u_n$  et la partie continue appelée commande équivalente  $u_{eq}$  :

$$u = u_{eq} - M \cdot \text{sign}(S) \quad (4.21)$$

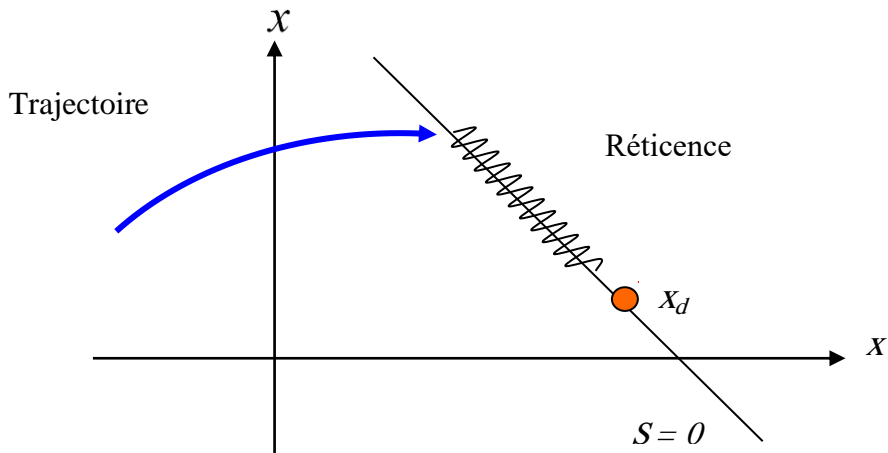
Tant que le régime glissant est assuré [condition (4.4) vérifiée], le système décrit par l'équation (4.8) est robuste, indépendant des variations des paramètres de la partie commandée et des perturbations qui agissent sur elle. Il admet en outre des erreurs de modélisation, tant que ces variations et erreurs maintiennent les conditions de régime glissant qui s'effectue alors au voisinage de la surface de glissement  $S(x) = 0$ .

En présence des imperfections mentionnées et pour se prémunir des oscillations résiduelles (dans le cas d'une commande  $u_n$  exprimée par l'équation (4.19)), la commande effective  $u_n$  peut être constituée par une composante  $u_{eq}$  de basse fréquence (moyenne de  $u$ ), et une composante  $u_n$  de haute fréquence.

Les amplitudes des oscillations sur les réponses du système en régime glissant peuvent être fortement réduites en utilisant la commande décrite par l'équation (4.21) au lieu de celle représentée par l'équation (4.20) et qui n'introduit pas la commande équivalente.

#### **IV.4.4 Le phénomène de réticence ou chattering**

L'oscillation au voisinage de la surface est appelée réticence (Chattering dans la littérature anglo-saxonne) ou broutement. La figure 4.11 montre l'effet de la réticence dans la convergence du système. La réticence n'est pas désirable, car elle induit des dynamiques de haute fréquence du système, augmentant la consommation énergétique.



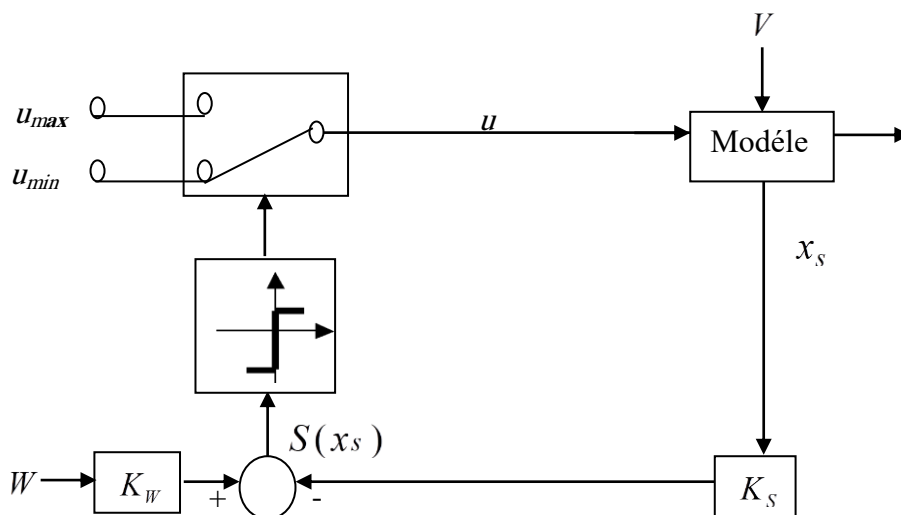
**Figure 4.11:** Phénomène de réticence

La réticence est le principal désavantage des modes glissants de nombreuses études ont été effectuée dans le but de réduire ou d'éliminer ce problème, par exemple les solutions par limitation de la condition de glissement, le terme  $sign(S)$  dans la partie du glissement de la commande est souvent remplacé par un terme à variation plus douce, par exemple la fonction de saturation  $sat(S)$  [22].

#### IV.4.5 Lois de commutation par contre-réaction d'état

Dans le cas de la figure 4.3, la loi de commutation est exprimée par une contre réaction d'état, dont le comportement dynamique du système de réglage peut être représenté de manière générale dans l'espace d'état en faisant appel à l'écriture matricielle. En analogie avec le réglage par contre réaction d'état, la loi de commutation est choisie par la relation :

$$S(x_s) = -K_s x_s + K_w w \quad (4.22)$$



**Figure 4.12:** Configuration avec loi de commutation par contre réaction d'état

Où:  $x_s$  : Vecteur d'état du système à régler de dimension "  $n_s$  ".

$w$  : Grandeur de consigne.

$K_s$  : Vecteur ligne de dimension  $n_s$ , qui contient les coefficients de la contre réaction d'état.

$K_w$  : Coefficient de l'intervention directe de la grandeur de consigne.

Le système à régler est régi par les équations d'état :

$$\begin{cases} \dot{x}_s = A_s x_s + B_s U + B_{sv} V \\ y_s = C_s x_s \end{cases} \quad (4.23)$$

Où  $V$  est la grandeur de perturbation intervenant sur le système à régler.

A cause de la non linéarité par commutation, la relation de la commande est :

$$\begin{cases} U = U_{\max} & \text{Pour } S(x_s) > 0 \\ U = U_{\min} & \text{Pour } S(x_s) < 0 \end{cases} \quad (4.24)$$

Ces relations peuvent être exprimées de manière plus compacte par [15] :

$$\begin{cases} U = \frac{U_{\max} + U_{\min}}{2} + \frac{U_{\max} - U_{\min}}{2} \cdot \text{sign}S(x_s) \\ \text{sign}S(x_s) = +1 & \text{Pour : } S(x_s) > 0 \\ \text{sign}S(x_s) = -1 & \text{Pour : } S(x_s) < 0 \end{cases} \quad (4.25)$$

Pour calculer la commande équivalente, et en régime de glissement, la loi de commutation suit les conditions :

$$\begin{cases} S(x_s) = 0 \\ \dot{S}(x_s) = 0 \end{cases} \quad (4.26)$$

Alors :

$$\dot{S}(x_s) = -k_s \dot{x}_s + k_w \dot{W} = -k_s (A_s x_s + B_s U + B_{sv} V) + k_w \dot{W} = 0 \quad (4.27)$$

La solution de (4.27) est donnée par  $u = u_{eq}$  avec :

$$U_{eq} = \frac{-1}{k_s B_s} k_s (A_s x_s + B_{sv} V) + \frac{1}{k_s B_s} k_w \dot{W} \quad (4.28)$$

Pour que "  $U_{eq}$  " ait une valeur finie, il faut que  $k_s B_s \neq 0$  [23].

## **IV. 5 La commande en mode glissant pour les systèmes non linéaires**

### **IV.5.1 Commande discontinue**

On a présenté la commande en mode glissant pour les systèmes non linéaires directement par un exemple. (Pour la théorie voir [11]).

Considérons le système non linéaire du deuxième ordre représenté par :

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = -x_1 x_2 + u + p(x,t) \\ y = x_1 \end{cases} \quad (4.29)$$

On souhaite pour la sortie  $y$ , une réponse à une entrée nulle du type deuxième ordre de constante de temps équivalente égale à 1/3 de seconde. Comme ce n'est pas possible ici, on choisit un comportement du type premier ordre avec la même constante de temps.

Il en résulte pour l'expression :

$$S = 3x_1 + x_2 \quad (4.30)$$

En utilisant l'équation (4.3), la condition de glissement devient :

$$S\dot{S} = S(3x_2 - x_1 x_2 + u + p(x,t)) < 0 \quad (4.31)$$

La résolution de cette inéquation, par rapport à  $u$ , fournit des solutions assurant l'évolution du système en régime glissant. Le bouclage de l'équation (4.32) assure l'attractivité de la surface  $S$ .

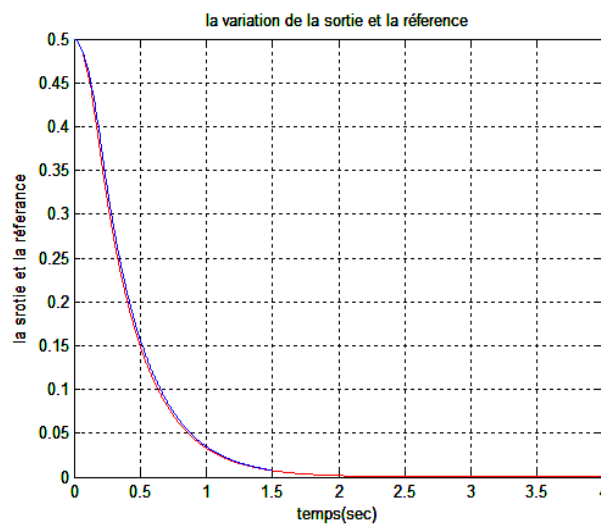
$$u = -3x_2 + x_1 x_2 - M \operatorname{sign}(S) \quad (4.32)$$

La simulation a été réalisée dans le cas où :

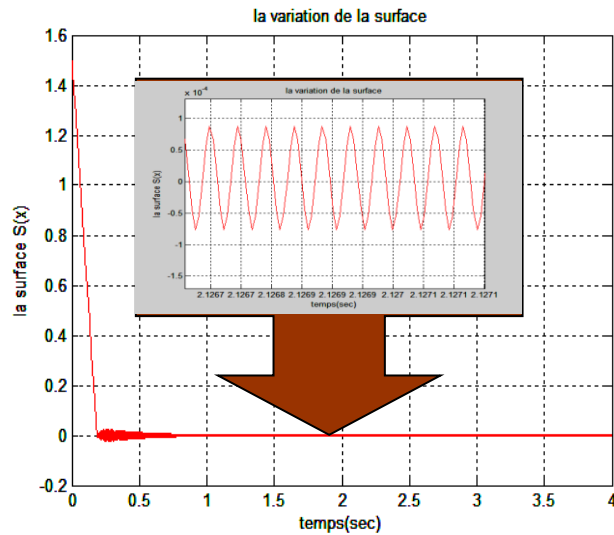
$$p(x,t) = \sin(t) \quad (4.33)$$

Par suite, pour satisfaire la condition d'attractivité de la surface et de robustesse de la commande impose un gain de glissement  $M > 2$ .

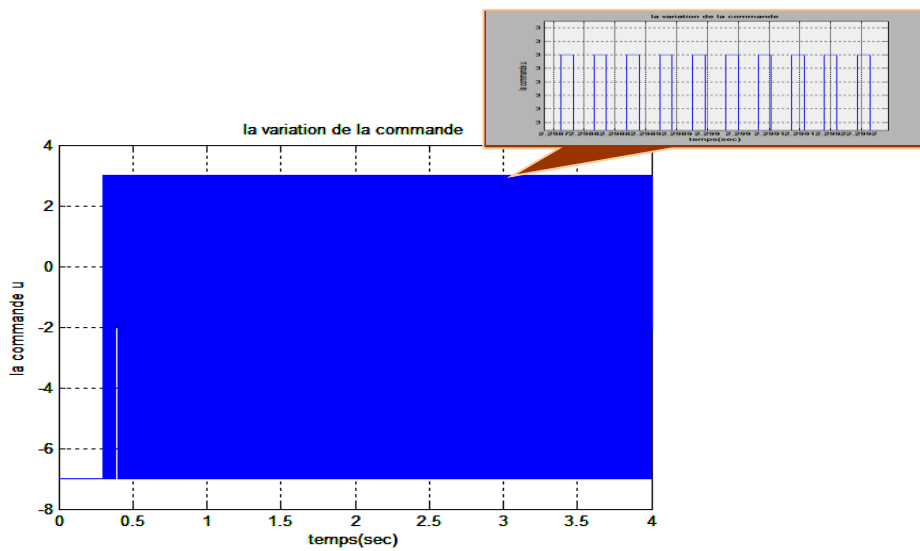
Les figures suivantes présentent les résultats de simulation qui a été faite en 4 secondes pour un gain de glissement  $M=5$ .



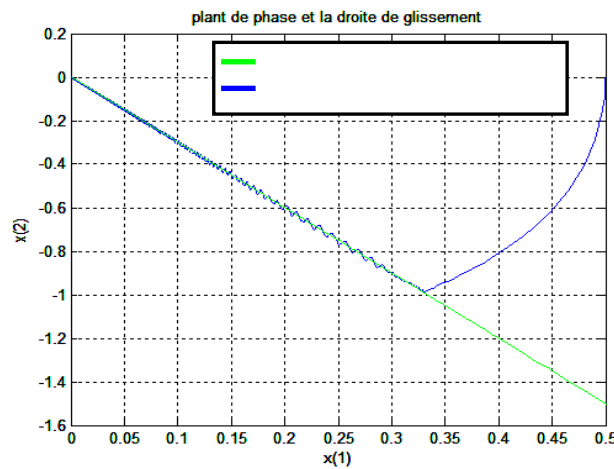
**Figure 4.13:** La variation de la sortie et la référence



**Figure 4.14:** La variation de la surface de glissement



**Figure 4.15:** La variation de la commande discontinue



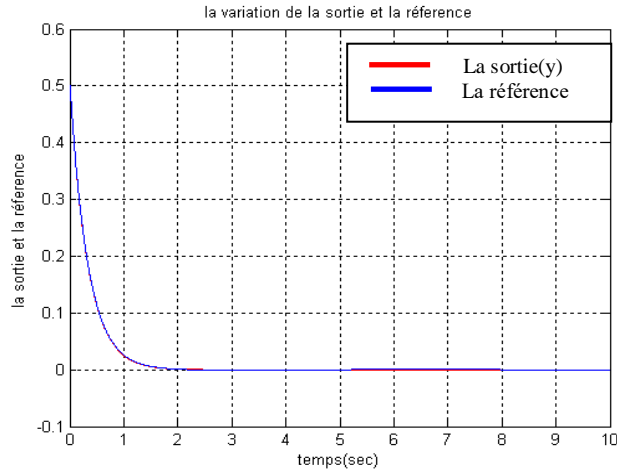
**Figure 4.16:** Plan de phase et droite de glissement

On constate que la sortie réelle  $y$  du processus coïncide avec la sortie désirée  $y_d$  (sortie du système non perturbé). Ce type de loi de commande permet donc d'atteindre l'objectif souhaité.

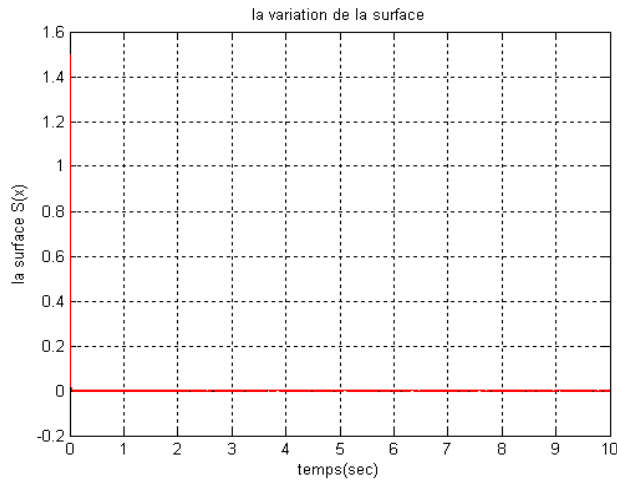
### IV.5.2 Commande continue

En remplaçant la fonction  $sign(S)$  dans (4.32) par la fonction  $sat(\frac{S}{\Phi})$ , la commande sera :

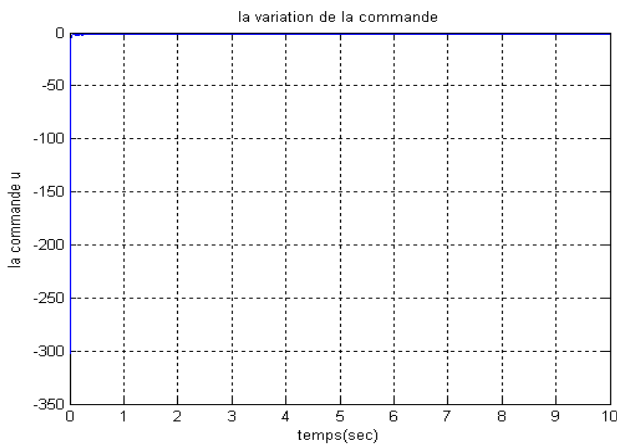
$$u = -3x_1x_2 + x_1x_2 - M \cdot sat(\frac{S}{\Phi}) \quad (4.34)$$



**Figure 4.17:** La variation de la sortie et la référence

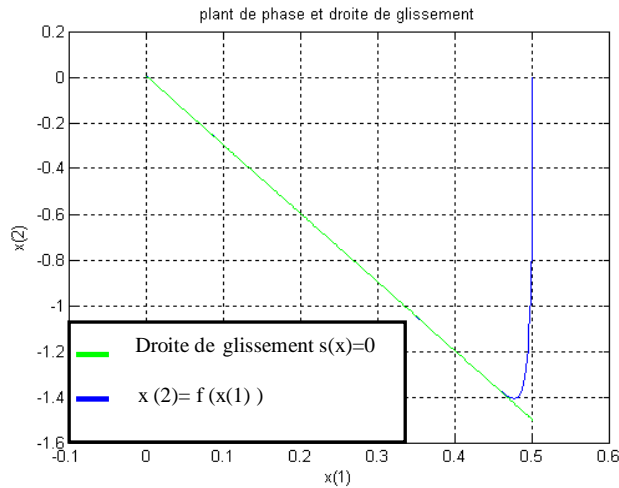


**Figure 4.18:** La variation de la surface de glissement



**Figure 4.19:** variation de la commande continue





**Figure 4.20:** Plant de phase et la droite de glissement

Les résultats de simulation montrent les bonnes performances de poursuite et la convergence de la réponse. Néanmoins on a toujours de grandes et rapides variations dues aux phénomènes de chattering.

#### **IV.6 Application de commande en mode glissant sur le pendule inversé**

La simulation est fait sur le modèle linéaire du pendule inversé présenté dans le chapitre précédent :

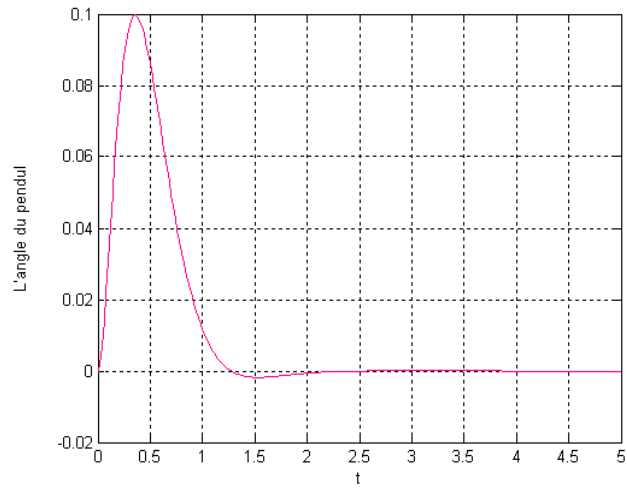
$$\dot{x}(t) = Ax(t) + Bu(t) \quad (4.35)$$

Avec: 
$$A = \begin{pmatrix} 0 & 0 & \alpha 1 & 0 \\ 0 & 0 & \beta 1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{pmatrix} \quad B = \begin{pmatrix} \alpha 2 \\ \beta 2 \\ 0 \\ 0 \end{pmatrix}$$

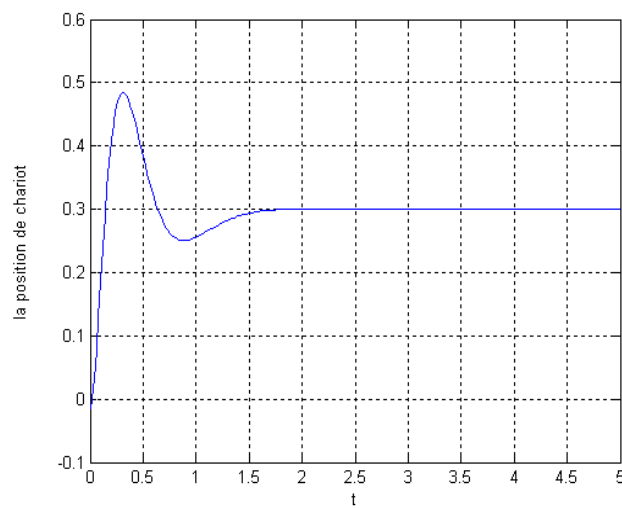
$$\alpha 1 = \frac{6g(m_p + m_c)}{(4m_c + m_p)l} \quad \alpha 2 = \frac{-6}{(4m_c + m_p)l} \quad \beta 1 = \frac{-3m_p}{4m_c + m_p} g \quad \beta 2 = \frac{4}{4m_c + m_p}$$

Pour les deux cas de commande, continue ou discontinue, on a fait la simulation avec gain de glissement égale à 5.

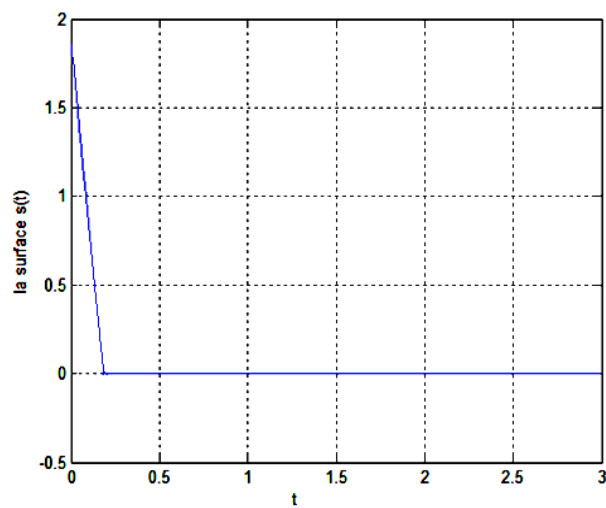
### IV.6.1 Commande discontinue



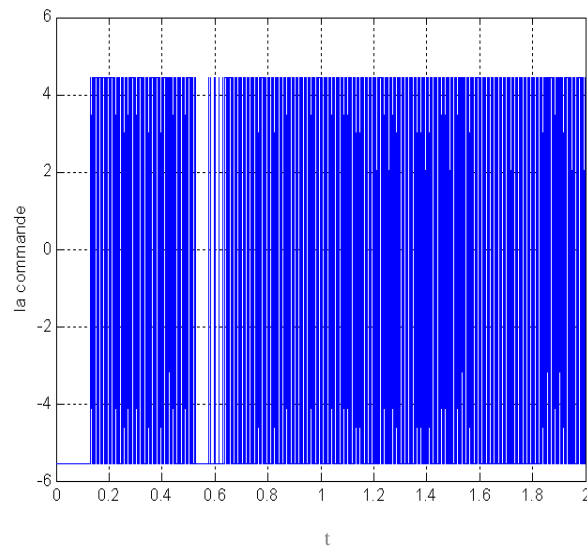
*Figure 4.21: La variation de l'angle du pendule*



*Figure 4.22: La variation de la position du pendule*



*Figure 4.23: La variation de la surface de glissement*

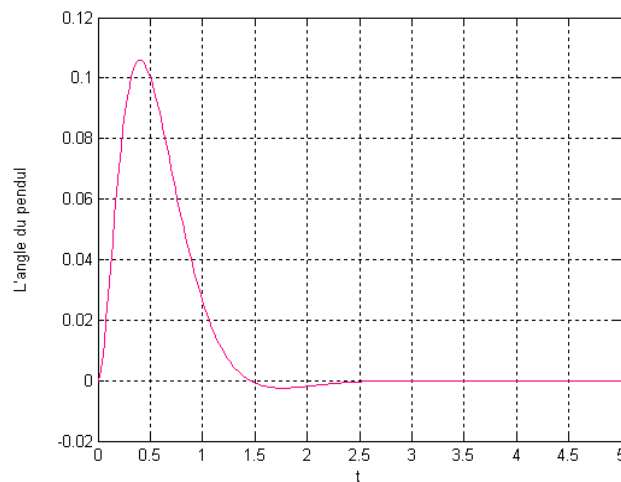


**Figure 4.24:** *La variation de la commande discontinue*

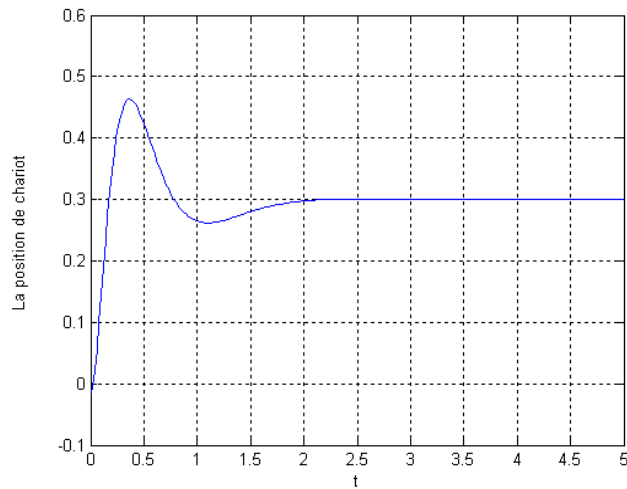
**Commentaire :**

Les résultats de la simulation montrent les bonnes performances de la commande en mode glissant, néanmoins la commande présente de grandes et rapides variations dues au phénomène de chattering. Ce type de commande serait difficile à mettre en œuvre.

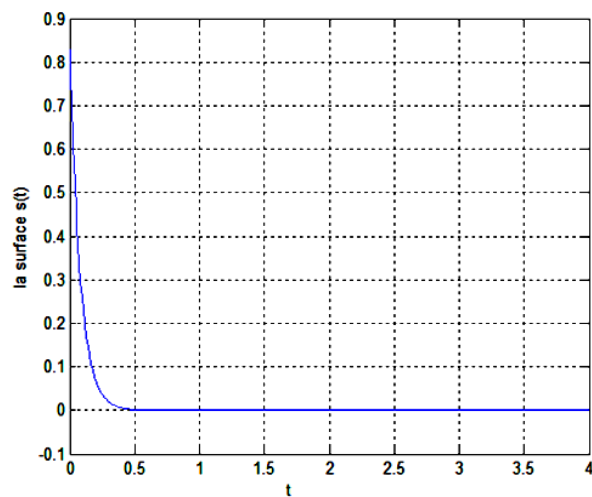
**IV.6.2 Commande continue**



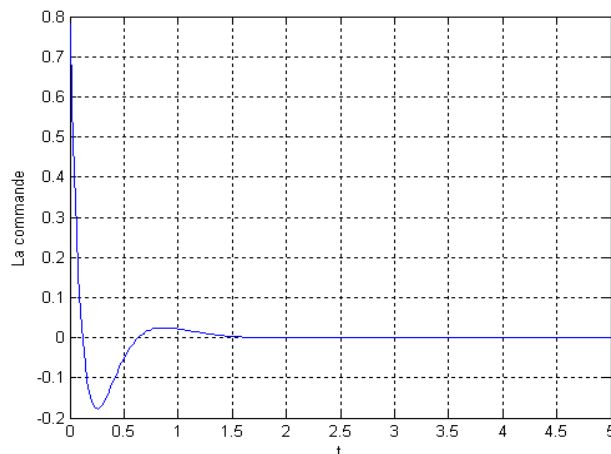
**Figure 4.25:** *La variation de l'angle du pendule*



*Figure 4.26: La variation de la position du pendule*



*Figure 4.27: La variation de la surface de glissement*



*Figure 4.28: La variation de la commande continue*

**Commentaire :**

Le résultat obtenu avec la commande continue presque la même obtenue par la commande discontinue avec l'absence du phénomène de chattering, donc, la commande continue permet de résoudre le problème de broutement et facilite de mise en œuvre ce type de commande.

## **Conclusion**

Le but principal de ce mémoire est de voir la possibilité de l'application de deux types différents de commande, la commande prédictive généralisée et la commande en mode glissant, sur un système non linéaire qui est le pendule inversé.

La commande prédictive prend la grande partie dans ce mémoire, on a présenté dans le premier chapitre l'algorithme de base, le choix des paramètres de commande, et une application de cet algorithme sur un système donné, ou on a vu l'efficacité de ce type de commande lorsque le modèle est une fonction de transfert.

Dans le deuxième chapitre, on a parlé sur deux termes très importantes qui sont les contraintes et l'optimisation. On a présenté les différents types de contraintes, et la résolution de ce problème par deux méthodes: la programmation dynamique et la programmation quadratique. On a utilisé autres méthodes dans la phase d'optimisation du critère. L'application sur un modèle nous permet de dire que la commande prédictive généralisée reste applicable et efficace malgré la présence des contraintes et l'utilisation d'autres méthodes d'optimisation.

Dans le troisième chapitre, on a présenté le modèle linéarisé et le modèle flou de pendule inversé, et on a réécrit l'algorithme de GPC lorsque le modèle est une représentation d'état.

L'application de cet algorithme sur les deux modèles nous permet de juger que la GPC est une meilleure méthode de régulation pour les systèmes linéaires et les systèmes non linéaires.

Dans le quatrième chapitre, on a parlé brièvement sur la théorie et l'algorithme de la commande en mode glissant, on peut dire après l'application qu'elle est aussi une meilleure méthode de régulation.

Le choix d'une méthode de régulation dépend de type de modèle, les contraintes sur le régime transitoire et le régime permanent, et les avantages et les inconvénients d'une méthode par rapport à l'autre. Une méthode choisie est la meilleur seulement pour un système donné et un cahier de charge à prendre en compte.

Finalement, pour faire déplacer le pendule inversé, on a besoin d'un moteur à courant continue, donc et comme perspective, on propose de trouver le modèle de l'ensemble (pendule inversé + moteur), l'application sur le modèle pour régler les paramètres de commande et trouver les sorties, puis l'application sur le système pour confirmer pratiquement les résultats obtenus.

On propose aussi de refaire tout le travail avec le double pendule inversé.

## Annexe 1: Le modèle TS à partir d'un modèle non linéaire

Pour mieux comprendre la méthode utilisée pour avoir le modèle flou, on propose des exemples :

### Exemple 1 :

On considère le modèle non linéaire suivant :

$$\dot{x}(t) = x(t) \cdot \cos(x(t))$$

Pour traiter la non linéarité, il faut prendre en compte la fonction  $f(x) = \cos(x)$  qui évidemment bornée. Pour  $x \in [-x_0, x_0]$ ,  $x_0 < \pi$  on obtient :

$$\cos(x) = \frac{\cos(x) - \cos(x_0)}{1 - \cos(x_0)} \cdot 1 + \frac{1 - \cos(x)}{1 - \cos(x_0)} \cdot \cos(x_0)$$

On aboutit donc au modèle TS à deux règles suivant :

$$\text{Si } x \text{ est } h_1(x) \text{ alors } \dot{x}(t) = x(t) \cdot 1$$

$$\text{Si } x \text{ est } h_2(x) \text{ alors } \dot{x}(t) = x(t) \cdot \cos(x_0)$$

### Exemple 2 :

Le modèle flou n'est pas unique, et il peut être capable d'obtenir des modèles ayant un nombre de règles minimal. Pour illustrer ces propos considérons le modèle non linéaire suivant :

$$\dot{x}(t) = \sin(x(t)) - x^3(t) + x(t) \cos(x(t))$$

Une première façon de faire, de manière brute, est de considérer chaque non linéarité séparément, ce qui revient à devoir transformer :  $\frac{\sin(x)}{x}$ ,  $x^2$  et  $\cos(x)$  et qui aboutit à un modèle à

8 règles. On peut très facilement aboutir à un modèle à 2 règles en considérant directement la

fonction  $f(x) = \frac{\sin(x)}{x} - x^2(t) + \cos(x(t))$ . Par exemple pour  $x \in [-\pi, \pi]$  on a :

$f_{\min} = 1 - \pi^2$  et  $f_{\max} = 2$ , ce que permet d'obtenir un modèle flou à 2 règles suivantes :

$$\text{Si } x \text{ est } h_1(x) \text{ alors } \dot{x}(t) = x(t) \cdot 2$$

$$\text{Si } x \text{ est } h_2(x) \text{ alors } \dot{x}(t) = x(t) \cdot (1 - \pi^2)$$

### Exemple 3 :

Pour deux variables d'état, on considère le modèle non linéaire suivant :

$$\dot{x}_1(t) = x_1^3(t) + \sin(x_1(t))x_2^2(t)$$

$$\dot{x}_2(t) = x_1(t) - x_2(t)$$

Avec des transformations directes sur les non linéarités présentes, il faut utiliser les fonctions  $\sin(x_1(t))$ ,  $x_1^2(t)$  et  $x_2(t)$  ce qui amène un modèle sous la forme TS à 8 règles.

La deuxième équation étant linéaire on a bien entendu  $f_2^1(x(t)) = f_2^2(x(t)) = 1$ .

En utilisant :  $f_1^1(x(t)) = x_1^2$  et  $f_1^2(x(t)) = \sin(x_1(t)).x_2(t)$ , et en utilisant comme variable de prémisses  $z_1(t) = x_1^2(t)$  et  $z_2(t) = \sin(x_1(t)).x_2(t)$  on aboutit à un modèle à 4 règles. Mais dans ce cas, on peut même proposer un modèle à 2 règle, on peut utiliser comme variable de prémisses :

$$z_1(t) = x_1^2(t) + \frac{\sin(x_1(t))}{x_1(t)} x_2^2(t), \text{ ainsi } f_1^2(x(t)) = 0 \text{ et on aboutit au modèle pour } x_1(t) \in [-x_{10}, x_{10}]$$

$$x_2(t) \in [-x_{20}, x_{20}], \text{ et avec } z_{1\min} = \min_{x_1, x_2}(z_1), z_{1\max} = \max_{x_1, x_2}(z_1), h_1(z_1) = \frac{z_1(t) - z_{1\min}}{z_{1\max} - z_{1\min}},$$

$$h_2(z_1) = \frac{z_{1\max} - z_1(t)}{z_{1\max} - z_{1\min}}$$

$$\text{Si } z_1 \text{ est } h_1(z_1) \text{ alors } \dot{x}(t) = \begin{bmatrix} z_{1\max} & 0 \\ 1 & -1 \end{bmatrix} x(t)$$

$$\text{Si } z_1 \text{ est } h_2(z_1) \text{ alors } \dot{x}(t) = \begin{bmatrix} z_{1\min} & 0 \\ 1 & -1 \end{bmatrix} x(t)$$

## **Annexe 2 : Détermination du gain de glissement [25]**

Il existe plusieurs méthodes pour calculer le vecteur  $C^T$  de glissement, on utilise dans cette mémoire la méthode d'ACKERMAN. Cette méthode est résumé comme suit :

- Soit  $[\lambda_1, \lambda_2, \dots, \lambda_{n-1}]$  est choisi comme le spectre désirer du mouvement glissant.
- On trouve l'équation du  $S = C^T x$  de la surface de glissement comme suit :

$$C^T = e^T (A - \lambda_1.I)(A - \lambda_2.I) \dots (A - \lambda_{n-1}.I)$$

$$\text{Avec : } e^T = (0 \ 0 \ 0 \ \dots \ 1) [B \ A.B \ A^2.B \ \dots \ A^{n-1}.B]$$

On utilise le logiciel MATLAB pour obtenir les résultats.

## Références

- [1]: D. Clarke, C. Mohtadi, "Generalized Predictive Control – part I".
- [2]: Ernesto Granado Migliore, "Commande Prédictive à base de programmation semi définie".
- [3]: Gérard Ramond, "Contribution à la commande prédictive généralisée adaptative directe et applications", le grade de docteur en sciences de l'université Paris Xiorsay, septembre 2001.
- [4]: Mémoire fin d'étude, "La Robustification de la loi de commande prédictive généralisée Sous contraintes", pour grade ingénieur, 2005
- [5]: J. Espinosa, J. Vandewalle, "Constrained Predictive Control Using Fuzzy Models", site web: [www.neci.nj.nec.com](http://www.neci.nj.nec.com).
- [6]: J. J. Espinosa, M. L. Hadjili, V. Werts, "Predictive Control Using Fuzzy Model Comparative Study", European Control Conference, Karlshur, Germany, 1999.
- [7]: Easter Tan, Gilles Mourot, D. Maquin, J. Ragot, "Identification of fuzzy models", centre de recherche en Automatique de Nancy -CNRS URA 821, site web: [www.neci.nj.nec.com..](http://www.neci.nj.nec.com..)
- [8]: Jean – Pierre Barrat et Yves Lecluse, "Logique floue technique de l'ingénieur".
- [9]: S. Olaru, "Contribution de l'étude de la commande prédictive sous contraintes par approche géométrique", pour le grade de docteur en sciences de l'université Paris Xiorsay, 2005
- [10]: Help Matlab v. 6.5, "Clustering- Subtractive".
- [11]: Mémoire fin d'étude, "Nouvelles approches de la commande en mode glissant", pour grade ingénieur d'état, 2008.
- [12]: J.J. Slotine, "Sliding controler design for non linear system".
- [13]: D. Arzelier et D. Peaucelle, "Systèmes et asservissements non linéaires".
- [14]: Paul E. Vidal, "Commande non linéaire d'une machine asynchrone à double alimentation", thèse de l'institut national polytechnique de Toulouse, 14 décembre 2004.
- [15]: O. Mahmoud, N. Madani, N.F Benkhoris, F. Boudjema, "Cascade sliding mode control of a field oriented induction machine drive", the European physical journal, April 1999.
- [16]: J.J. Slotine, Weiping Li, "Applied nonlinear control", Prentice-Hall International Inc, USA, 1991.
- [17]: W. Gao, J.C. Hung, "Variable structure control system", IEEE Tras.Indust.Applic.
- [18]: A. Kerboua, "Commandes et observateur par mode de glissement", Thèse de magistère, ENP, 1999.



- [19]: Tomas Salgado Jimenez, "Contribution à la commande d'un robot sous-marin autonome de type torpille", Thèse de doctorat université Montpellier, 21 décembre 2004.
- [20]: V. Utkin, "Variable structure systems with sliding mode", IEEE Trans. Auto. Control.
- [21]: S. Barkati, "commande robuste par retour d'état d'une machine synchrone", Thèse de magistère, ENP 1997.
- [22]: A. Gayed, "Contribution à la commande par mode de glissement d'une machine synchrone à aimants permanents", Thèse de l'Université de Nantes, 1996.
- [23]: V. Avenikov, "Transient Processus in Electrical Power Systems", Edited by V.A stroyer, translated from the Russian by N.YA,Utkin.
- [24]: C. Mira, "Système asservis non linéaires". Hermés, Paris, 1990.
- [25]: J. Ackermann, V.Utkin, "Sliding Mode Control Design Based on Ackermann's Formula",

## **Résumé**

L'automatique a souvent recours à des cas d'étude particuliers, qui sont représentatifs de grande classe d'application, et dont le caractère spectaculaire est conforme.

De plus avec l'expérience, la connaissance de ces cas est affinée et ils fournissent aujourd'hui une base idéale pour comparer d'une façon valable les avantages et les inconvénients d'approche différente. Le pendule inversé est un de ces cas types.

Il existe plusieurs types de commande qu'on peut utiliser pour commander le pendule inversé, mais dans ce mémoire, on a choisi deux types seulement. Le premier type et qui prend une grande partie de l'étude est la commande prédictive généralisée, ce type de commande est devenue l'un des algorithmes de commande prédictive les plus répandus. Elle a été très tôt implémentée, avec succès, dans de nombreuses applications industrielles, et a montré de bonnes performances et un certain degré de robustesse.

Le deuxième type est la commande en mode glissant, ce type de commande peut être considérée comme appartenant à la classe des commandes robustes.

Ce mémoire présente essentiellement la théorie de les deux algorithmes de commande et la possibilité de les appliquer sur le pendule inversé.

## **Abstract**

Automatics often resorts to particular study cases which represent high application classes and whose spectacular character is in accordance.

In addition, with experience, knowing these cases is refined and these cases, nowadays, provide an ideal basis for comparing in a meaningful way the advantages and disadvantages of different approaches. The inverted pendulum is one of those cases.

There are several types of control that can be used to control the inverted pendulum, but in this thesis, we chose two types only.

The first type, which takes a large part of the study, is the generalized predictive control. This type of control has become one of the most common predictive control algorithms. It was early implemented successfully in many industrial applications, and has shown good performance and a degree of robustness.

The second type is the sliding mode; this command can be considered as belonging to the class of robust controls. This thesis present mainly on the theory of the two control algorithms and the possibility to be applied to the inverted pendulum.

