

M0036/02A

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

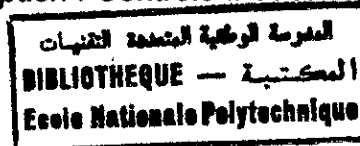
un seul  
ed.

**UNIVERSITE DE BATNA**  
FACULTE DES SCIENCES DE L'INGENIEUR

## Mémoire

Présenté au Département d'Electronique pour l'obtention du diplôme de  
MAGISTER en ELECTRONIQUE

Option : Contrôle Industriel



THEME

# ETUDE DU CHOIX DES PARAMETRES DE REGLAGE DES ALGORITHMES GPC

Par

**Salima DJEBRANI**

*Ingénieur d'Etat en Electronique*

Soutenu le : 30 / 10 / 2002

Devant le jury constitué de :

M <sup>r</sup> . <u>Mohammed BOULEMDEN</u> ,	Prof, Président	Université de Batna
M <sup>r</sup> . <u>Khier BENMAHAMMED</u> ,	Prof, Rapporteur	Université de Sétif
M <sup>r</sup> . <u>Foudil ABDESSEMED</u> ,	C.C, Co-Encadreur	Université de Batna
M <sup>r</sup> . <u>Khaled BELARBI</u> ,	Prof, Examineur	Université de Constantine
M <sup>r</sup> . <u>Noureddine GOLEA</u> ,	M.C, Examineur	Université de Oum El-Bouagui



## **Mémoire**

*Présenté au Département d'Electronique pour l'obtention du diplôme de  
MAGISTER en ELECTRONIQUE  
Option : Contrôle Industriel*

THEME

# **ETUDE DU CHOIX DES PARAMETRES DE REGLAGE DES ALGORITHMES GPC**

*Par : Salima DJEBRANI*



## Remerciements

Mes remerciements sont adressés à Monsieur Khier BENMAHAMMED, *Professeur à l'institut d'électronique de l'université de Sétif*, pour avoir accepté de m'encadrer et d'assurer la direction scientifique de mes travaux. Je le remercie aussi pour sa gentillesse et sa compréhension.

Mes grands remerciements vont particulièrement à Monsieur Foudil ABDESSEMED, *Chargé de cours à l'institut d'électronique de l'université de Batna*, pour avoir participé à l'encadrement et pour ses orientations et ses judicieux conseils tout au long de l'élaboration de mon travail. Je le remercie encore une fois pour avoir été gentil, compréhensif et bon conseiller.

Mes vifs remerciements à Monsieur Mohammed BOULEMDEN, *Professeur à l'institut d'électronique de l'université de Batna*, pour avoir accepté de présider le jury.

De même à Monsieur Khaled BELARBI, *Professeur à l'institut d'électronique de l'université de Constantine*, pour m'avoir honoré par sa présence au jury.

Je tiens à remercier Monsieur Noureddine GOLEA, *Maitre de conférence à l'institut d'électrotechnique du centre universitaire de Oum El-Bouagui*, pour la documentation qu'il m'a fournit et pour l'honneur qu'il me fait en figurant dans la composition du jury.

Je remercie Monsieur Vincent WERTZ, *Professeur à l'université catholique de Louvain (Belgique)*, pour sa contribution.

Je remercie également Monsieur Salim FILALI, *enseignant à l'institut d'électronique de l'université de Constantine*, pour son aide, sa gentillesse et ses éclaircissements.

Mon grand remerciement va à Monsieur Noureddine BOUGUECHAL, *Doyen de la faculté des sciences de l'ingénieur à l'université de Batna*.

Je remercie Monsieur Ahmed LOUCHENE, *chargé de cours à l'institut d'électronique de l'université de Batna*.

Je n'omet pas de remercier enfin M<sup>r</sup> HAMADA pour son aide et sa confiance, M<sup>me</sup> Nabila et M<sup>lle</sup> Kalinka.

Ma profonde reconnaissance va, à mes parents, à ma sœur et à mes deux frères.

Enfin, ma profonde gratitude devrait être exprimée envers tous les enseignants qui ont contribué à ma formation et à tous qui m'ont aidé de près ou de loin.

## Résumé

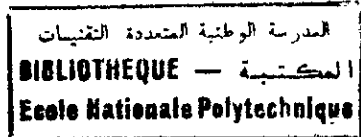
La commande prédictive à base de modèle (*MBPC* : Model Based Predictive Control) a joué un rôle très important dans le domaine de contrôle des processus. Le principe du contrôle prédictif est basé sur la prédiction du comportement futur du processus à contrôler sur un horizon du temps fini. En effet ; le problème est, connaissant la sortie prédite du système, de déterminer la commande à appliquer au système pour que la sortie suit une trajectoire de référence désirée. Le contrôle prédictif généralisé (*GPC* : Generalized Predictive Control) de *Clarke, Mohtadi et Tuffs* est considéré comme étant la plus populaire des méthodes de prédiction, particulièrement pour des processus industriels. La méthode du *GPC* est une procédure de contrôle applicable aussi bien à des systèmes *SISO* qu'à des systèmes *MIMO*. Un modèle de processus est employé pour prédire le comportement futur du système à contrôler à travers un horizon donné de temps. De cette prédiction, une commande est calculée en réduisant au minimum une fonction quadratique de coût. Cependant, les paramètres de réglage du *GPC* (essentiellement l'horizon de prédiction, l'horizon de commande et le facteur de pondération) ont une influence déterminante sur le comportement du système en boucle fermée. Mais, il n'est pas toujours facile de trouver des valeurs optimales pour ces paramètres. C'est pourquoi, on propose dans ce travail d'utiliser un algorithme génétique pour les optimiser. L'application étudiée est l'asservissement d'un chariot avec pendule inversé qui est un système non linéaire et instable. L'objectif final de cette application consiste à asservir la position du chariot tout en maintenant le pendule inversé dans sa position verticale.



## Table des matières

<b>Introduction générale</b>	1
<b>Chapitre 1 : Stratégies de contrôle numérique</b>	4
1.1. Introduction	4
1.2. Forme canonique des contrôleurs numériques	6
1.3. Régulateur <i>P.I.D.</i> numérique	9
1.4. Placement des pôles	12
1.4.1. Structure	12
1.4.2. Régulation (Calcul de $R(q^{-1})$ et $S(q^{-1})$ )	13
1.4.3. Poursuite (Calcul de $T(q^{-1})$ )	14
1.4.4. Conception dans le domaine temporel	16
1.4.5. Prédicteur de placement des pôles	17
1.5. Poursuite et régulation à objectifs indépendants	18
1.5.1. Structure	19
1.5.2. Régulation (calcul de $R(q^{-1})$ et $S(q^{-1})$ )	20
1.5.3. Poursuite (calcul de $T(q^{-1})$ )	21
1.5.4. Conception dans le domaine temporel	22
1.5.5. Prédicteur de régulation et poursuite à objectifs indépendants	23

1.6.	Poursuite et régulation avec pondération de l'entrée	24
1.6.1.	Structure	24
1.6.2.	Calcul du régulateur	25
1.6.3.	Conception dans le domaine temporel	25
1.7.	Poursuite et régulation à variance minimale	26
1.7.1.	Conception directe	27
1.7.2.	Utilisation du théorème de séparation	28
1.7.3.	Rapprochement avec poursuite et régulation à objectifs indépendants	29
1.8.	Poursuite et régulation à variance minimale généralisée	30
1.8.1.	Calcul du régulateur	31
1.9.	Conclusion	31
<b>Chapitre 2 : Le contrôle prédictif généralisé</b>		<b>33</b>
2.1.	Introduction	33
2.2.	Généralisation	34
2.3.	Le contrôle prédictif généralisé	35
2.3.1.	La loi de commande prédictive	36
2.3.2.	Le prédicteur à j-pas	37
2.3.3.	Choix de l'horizon de sortie et de l'horizon de contrôle	42
2.4.	Simulation	43
2.4.1.	Simulation 1	43
2.4.1.1.	Modélisation du système à 2 degré de liberté	44
2.4.1.1.1.	Energie cinétique du système en mouvement	44
2.4.1.1.2.	Energie potentielle du système	45
2.4.1.1.3.	Equation de Lagrange pour le degré de liberté $q(t) = \alpha(t)$	45
2.4.1.1.4.	Equation de Lagrange pour le degré de liberté $q(t) = x(t)$	46
2.4.1.1.5.	Linéarisation du modèle autour du point de fonctionnement	46
2.4.1.2.	Modélisation du système à 2 degré de liberté	47
2.4.1.3.	Test du modèle discrétisé	48
2.4.1.4.	Asservissement de la variable angulaire $\alpha(t)$	52
2.4.1.5.	Asservissement de la position $x(t)$ et de l'angle $\alpha(t)$	59
2.4.2.	Simulation 2	78



2.4.2.1. Exemple 1	78
2.4.2.2. Exemple 2	80
2.5. Conclusion	83
<b>Chapitre 3 : Les algorithmes génétiques</b>	<b>84</b>
3.1. Introduction	84
3.2. Ce qui rend les algorithmes génétiques différents des méthodes classiques	85
3.3. Terminologie et définitions	87
3.4. Principes généraux	87
3.5. Les opérateurs génétiques	91
3.5.1. Reproduction	92
3.5.2. Le crossover	93
3.5.3. La mutation	93
3.6. Les paramètres d'un algorithme génétique	95
3.7. Les étapes d'un algorithme génétique	96
3.8. Critères de convergence des AGs	97
3.9. Le contrôle prédictif généralisé et les algorithmes génétiques	98
3.9.1. Position du problème	98
3.9.2. Simulation 1	101
3.9.3. Simulation 2	108
3.9.4. Simulation 3	109
3.9.5. Simulation 4	118
3.10. Comparaison	120
3.11. Conclusion	124
<b>Conclusion générale</b>	<b>125</b>
<b>Annexe A : Bruit et Modèles Stochastiques</b>	
<b>Annexe B : Forme du Prédicteur dans le cas Déterministe et dans le cas Stochastique</b>	
<b>Annexe C : Solutions Récursives des Equations Diophantine</b>	
<b>Annexe D : Animation Graphique du Système (Chariot avec Pendule Inversé)</b>	

## Introduction générale

La plupart des systèmes en automatique sont continus en nature, tandis que les réalisations récentes d'un contrôleur sont principalement numériques. Ainsi, une conception d'un système de commande commence habituellement avec un système continu et se termine avec un contrôleur à temps discret. Presque chaque technique de contrôle (*P.I.D*, placement des pôles, commande optimale...) a les deux versions, le continu et le discret ; cependant, l'automaticien est libre d'en choisir l'approche. Dans une application donnée, une fois que la technique de commande à employer est décidée, le choix de la stratégie de contrôle demeure le problème majeur.

Considérons une stratégie de commande qui emploie seulement la valeur courante du sortie de système comme une rétroaction et une stratégie différente de commande qui emploie la connaissance du comportement futur de la sortie pour calculer le signal de commande. Parmi ces deux stratégies, quelle est laquelle qui exécute le mieux ? C'est clairement la seconde, car on y tient compte des sorties futures du système, d'où l'appellation "sorties prédites".

La commande prédictive est l'une des plus importantes commandes dans le domaine de recherche ; d'où l'intérêt accru de nombreux auteurs qui lui ont consacré en précurseurs plusieurs études intéressantes qui en font une référence. Cette méthodologie a été introduite pour la première fois par *Richalet*. Plus tard, plusieurs chercheurs ont développé



d'autres algorithmes en se basant sur le même principe. Le principe du contrôle prédictif est basé sur la prédiction du comportement futur du processus à contrôler sur un horizon de temps fini. En effet, le problème est, connaissant la sortie prédite du système, de déterminer la commande à appliquer au système pour que sa sortie suit une trajectoire de référence désirée.

Toutes les méthodes du contrôle prédictif ont deux points en commun :

- ◆ Elles utilisent toutes, un modèle du système pour prédire le comportement à travers un horizon mobile fini.
- ◆ L'entrée de contrôle est calculée de telle façon à minimiser un critère quadratique.

Le contrôle prédictif généralisé (*GPC* : Generalized Predictive Control) de *Clarke, Mohtadi et Tuffs* est considéré comme étant la plus populaire des méthodes de prédiction, particulièrement pour des processus industriels. Il combine la prédiction du comportement futur du procédé avec la commande de rétroaction. Un modèle de processus est employé pour prédire le comportement futur du système à contrôler à travers un horizon donné de temps. De cette prédiction, une commande est calculée en réduisant au minimum une fonction quadratique de coût depuis l'instant courant  $t$  jusqu'à l'instant  $t+N$  où  $N$  est l'horizon de prédiction. Cette méthode généralise la méthode *GMV* (Generalized Minimum Variance), qui n'utilise qu'une seule valeur de la sortie prédite. La commande prédictive généralisée est une méthode robuste, qui permet de contrôler un système instable ou à phase non-minimale et de compenser l'effet du retard du système.

Malgré les avantages du *GPC*, le problème du choix des paramètres de conception se pose aux utilisateurs de cet algorithme. Jusqu'à présent, ils imposent des valeurs aux paramètres de réglage selon des règles empiriques proposées par les concepteurs de l'algorithme. Les paramètres de réglage (essentiellement l'horizon de prédiction, l'horizon de commande et le facteur de pondération) ont une influence déterminante sur le comportement du système en boucle fermée. Mais, il n'est pas toujours facile de trouver des valeurs optimales pour ces paramètres. C'est pourquoi, on propose dans ce travail d'utiliser un algorithme génétique pour les optimiser.

Les algorithmes génétiques sont des techniques de programmation d'inspiration biologique pouvant être appliqués pour résoudre un grand nombre de problèmes d'optimisation et d'apprentissage automatique. Ils sont capables d'obtenir des solutions quasi-optimales pour différents types de problèmes sans connaissance explicite du domaine de travail en manipulant simplement des chaînes de bits et en utilisant des opérations simples : génération des nombres aléatoires, la copie des chaînes et les échanges des morceaux de chaînes.

Ce mémoire est organisé en trois chapitres comme suit : Dans le premier chapitre, on présente quelques méthodes de calcul des régulateurs numériques. Ces méthodes concernent la régulation monovariante avec ou sans les perturbations déterministes ou aléatoires. Le deuxième chapitre sera consacré à la description et à l'analyse théorique de l'algorithme *GPC*. On y trouvera l'analyse détaillée d'un système mécanique non linéaire très réputé dans les tests de simulations, il s'agit en l'occurrence du chariot sur lequel est monté un pendule inversé. L'application de l'algorithme *GPC* sur ce système est envisagé.

Dans le troisième chapitre, on présentera un aperçu sur les algorithmes génétiques. Puis on montre comment un algorithme génétique est combiné avec le contrôle prédictif généralisé pour la détermination des paramètres de réglage optimaux du *GPC* (l'horizon de prédiction, l'horizon de commande et le facteur de pondération). On présente enfin, l'application de cette méthode sur le système de simulation choisi.

# Chapitre 1

## Stratégies de contrôle numérique

### 1.1. Introduction

L'automatique peut être définie comme la science et les techniques de l'automatisation des systèmes. Le système, s'il est isolé, n'est pas complètement séparé du monde : ce qui lui est extérieur peut agir par l'intermédiaire des entrées du système qui lui-même agit à l'extérieur par ses sorties. On appellera *sorties* les grandeurs caractéristiques ou variables auxquelles on s'intéresse directement et que l'on mesure. Pour les *entrées*, on a deux types : les premières seront appelées les actions, tandis que les secondes seront des perturbations [3]. Le but de l'automatique est de réaliser la commande du système considéré, c'est-à-dire, de faire varier les entrées pour que les sorties aient des valeurs fixées à l'avance, tout en satisfaisant certaines propriétés (la stabilité, le rejet des bruits aléatoires, la minimisation de l'effet des bruits aléatoires et le suivi des trajectoires).

Le correcteur (régulateur) élabore le signal de commande. Donc pour concevoir et ajuster un "bon" régulateur, il faut :

- ◆ Spécifier les performances désirées pour la boucle de commande-régulation.
- ◆ Connaître le modèle dynamique du procédé qui décrit la relation entre les variations de la commande et celles de la sortie.
- ◆ Déterminer la structure et calculer un régulateur permettant d'atteindre les performances désirées pour le modèle de procédé considéré [1][2].

Dans cette partie, on s'intéresse à la conception des contrôleurs numériques pour les systèmes *SISO* décrits par des modèles discrets sous la forme *entrée/sortie*. Le contrôleur aura aussi la forme *d'E/S* et constitue 3 polynômes (structure à trois branches *R-S-T*) faisant la relation entre la commande  $u(t)$ , la sortie  $y(t)$  et la trajectoire désirée  $y^*(t)$ . Dans plusieurs cas, la conception peut être faite en utilisant l'approche polynomiale qui correspond à la conception dans le domaine fréquentiel. Cependant, la même conception peut être obtenue par la synthèse dans le domaine temporel. L'approche temporelle a l'avantage, d'une part d'éclaircir la façon dont se fait la régulation dans le cas des systèmes avec retard, en mettant en évidence l'existence d'un prédicteur à l'intérieur du régulateur, et d'autre part, elle permet de mieux comprendre le lien avec la synthèse des régulateurs dans un environnement stochastique. Comparant l'objectif du contrôle dans le domaine temporel associé par diverses stratégies (la stratégie du placement des pôles, la poursuite et régulation à variance minimale généralisée, le contrôle prédictif généralisé...), on peut classifier ces stratégies et d'autres en deux catégories [2] :

❖ **Le contrôle prédictif d'un pas en avant, "one step ahead predictive control"**

Dans ces stratégies, on calcule la prédiction de la sortie à  $t+d+1$  ( $d$  le retard entier du procédé) nommé  $\hat{y}(t+d+1)$  en fonction de  $u(t)$ ,  $u(t-1)$ ,  $u(t-2), \dots$ ,  $y(t)$ ,  $y(t-1), \dots$  et on calcule  $u(t)$  tel que l'objectif du contrôle est satisfait.

❖ **Le contrôle prédictif à longue portée, "long range predictive control"**

Dans ces stratégies, l'objectif du contrôle est exprimé en terme des valeurs futures de la sortie sur un certain horizon et de la séquence des valeurs futures de la commande.

On a besoin de calculer  $\hat{y}(t+d+1)$ ,  $\hat{y}(t+d+2), \dots, \hat{y}(t+d+j)$

$$\hat{y}(t+d+1) = f_1 \{ y(t), y(t-1), \dots, u(t), u(t-1), \dots \}$$

$$\hat{y}(t+d+j) = f_j \{ y(t), y(t-1), \dots, u(t), u(t-1), \dots \} + g_j \{ u(t+1), \dots, u(t+j-1) \}$$

Pour satisfaire l'objectif du contrôle, la séquence des valeurs de la commande  $u(t)$ ,  $u(t+1), \dots, u(t+j-1)$  est calculée ; mais, seulement la première ( $u(t)$ ) est appliquée au procédé.

Toutes les stratégies du contrôle ont la même structure. L'utilisation d'une ou autre stratégie correspond à faire la différence entre les valeurs des paramètres du contrôleur pour le même modèle du procédé utilisé.

## 1.2. Forme canonique des contrôleurs numériques

Le procédé à contrôler est décrit par :

$$A(q^{-1})y(t) = q^{-d}B(q^{-1})u(t) + A(q^{-1})v(t) \quad 1.1$$

où  $y(t)$  la sortie,  $u(t)$  l'entrée,  $v(t)$  perturbation,  $d$  un retard pur multiple entier de la période d'échantillonnage et  $q^{-1}$  opérateur de retard.

On utilise les notations suivantes :

$$A(q^{-1}) = 1 + a_1q^{-1} + \dots + a_{n_A}q^{-n_A} = 1 + q^{-1}A^*(q^{-1}) \quad 1.2$$

$$B(q^{-1}) = b_1q^{-1} + \dots + b_{n_B}q^{-n_B} = q^{-1}B^*(q^{-1}) \quad 1.3$$

$$\text{avec } A^*(q^{-1}) = a_1 + a_2q^{-1} + \dots + a_{n_A}q^{-n_A+1} \quad 1.4$$

$$B^*(q^{-1}) = b_1 + b_2q^{-1} + \dots + b_{n_B}q^{-n_B+1} \quad 1.5$$

La forme générale (canonique) d'un contrôleur numérique est donnée par :

$$S(q^{-1})u(t) + R(q^{-1})y(t) = T(q^{-1})y^*(t+d+1) \quad 1.6$$

$y^*(t+d+1)$  est la trajectoire de poursuite désirée donnée avec  $d+1$  pas en avant. Cette trajectoire peut être mémorisée ou engendrée à chaque changement de consigne à l'aide d'un modèle de référence.

$$y^*(t+d+1) = \frac{B_m(q^{-1})}{A_m(q^{-1})}r(t) \quad 1.7$$

(telles que  $r(t)$  est la consigne et  $B_m(q^{-1}) = b_{m0} + b_{m1}q^{-1} + \dots$ ).

La figure (1.1) représente la boucle de contrôle avec le contrôleur numérique  $R$ - $S$ - $T$ .

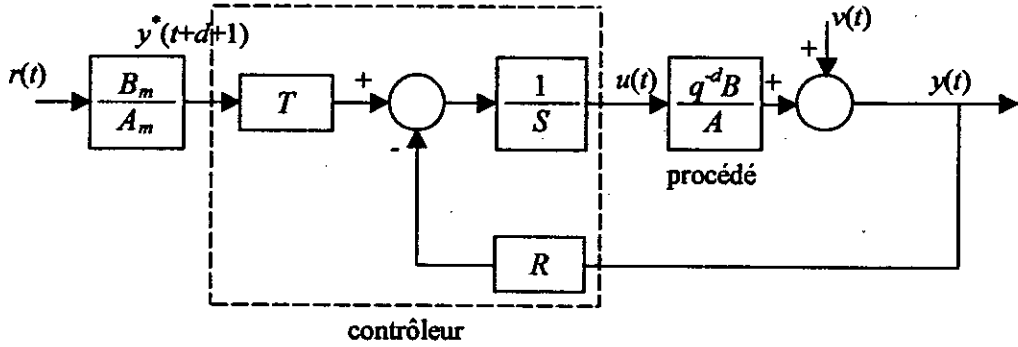


Figure 1.1. Boucle de contrôle avec le contrôleur numérique  $R-S-T$

Le contrôleur de l'équation (1.6) est nommé un contrôleur  $R-S-T$ . L'objectif de régulation est assuré par la partie  $R-S$  et de poursuite par le polynôme  $T$ .

En utilisant les équations (1.1) et (1.6), on peut définir les opérateurs de transfert qui définissent la relation entre d'un côté la trajectoire désirée et la perturbation et de l'autre côté la sortie et l'entrée du procédé [2].

Les pôles en boucle fermée sont définis par :

$$P(q^{-1}) = A(q^{-1})S(q^{-1}) + q^{-d-1}B^*(q^{-1})R(q^{-1}) \quad 1.8$$

et en éliminant  $u(t)$  entre les équations (1.1) et (1.6), on obtient :

$$y(t) = H_{CL}(q^{-1})y^*(t+d+1) + S_{yp}(q^{-1})v(t) \quad 1.9$$

où

$$H_{CL}(q^{-1}) = \frac{q^{-d-1}B^*(q^{-1})T(q^{-1})}{P(q^{-1})} \quad 1.10$$

et

$$S_{yp}(q^{-1}) = \frac{A(q^{-1})S(q^{-1})}{P(q^{-1})} \quad 1.11$$

$S_{yp}(q^{-1})$  est la fonction de sensibilité *perturbation-sortie*. Par la même manière on obtient l'expression de  $u(t)$ .

$$u(t) = H_U(q^{-1})y^*(t+d+1) + S_{up}(q^{-1})v(t) \quad 1.12$$

où

$$H_U(q^{-1}) = \frac{A(q^{-1})T(q^{-1})}{P(q^{-1})} \quad 1.13$$

$$\text{et} \quad S_{up}(q^{-1}) = -\frac{A(q^{-1})R(q^{-1})}{P(q^{-1})} \quad 1.14$$

$S_{up}(q^{-1})$  est la fonction de sensibilité *perturbation-entrée*.

Le raisonnement dans le domaine fréquentiel (représentation par fonction de transfert) indique clairement que les pôles en boucle fermée auront un rôle d'atteindre les spécifications de régulation alors que dans le domaine temporel, il se traduit par la rejection d'une erreur de sortie. De l'équation (1.9), pour  $v(t) \equiv 0$ , on obtient :

$$P(q^{-1})y(t) - B^*(q^{-1})T(q^{-1})y^*(t) = 0 \quad 1.15$$

et de l'équation (1.12), pour  $v(t) \equiv 0$ , on obtient :

$$P(q^{-1})u(t) - A(q^{-1})T(q^{-1})y^*(t+d+1) = 0 \quad 1.16$$

Pour une rejection parfaite de la perturbation, les structures générales de  $R(q^{-1})$  et  $S(q^{-1})$  sont :

$$S(q^{-1}) = S'(q^{-1})H_S(q^{-1}) \quad 1.17$$

$$R(q^{-1}) = R'(q^{-1})H_R(q^{-1}) \quad 1.18$$

où  $H_S(q^{-1})$  et  $H_R(q^{-1})$  sont des polynômes introduits dans le contrôleur pour atteindre certaines performances.  $S'(q^{-1})$  et  $R'(q^{-1})$  sont les solutions de l'équation :

$$P(q^{-1}) = A(q^{-1})S'(q^{-1})H_S(q^{-1}) + q^{-d-1}B^*(q^{-1})R'(q^{-1})H_R(q^{-1}) \quad 1.19$$

La conception du contrôleur *R-S-T* peut être faite dans le domaine fréquentiel ou dans le domaine temporel. Dans le domaine fréquentiel, on parle des pôles désirés ou des fonctions de transfert désirées en boucle fermée, alors que dans le domaine temporel, on s'intéresse à forcer quelques performances à zéro ou à minimiser un critère quadratique.

Les méthodes de calcul des régulateurs numériques qui seront présentées dans ce chapitre sont :

- ◆ *P.I.D.* numérique.
- ◆ Placement des pôles en boucle fermée.
- ◆ Poursuite et régulation à objectifs indépendants.

- ◆ Poursuite et régulation avec pondération de l'entrée.
- ◆ Poursuite et régulation à variance minimale.
- ◆ Poursuite et régulation à variance minimale généralisée.

### 1.3. Régulateur *P.I.D.* numérique

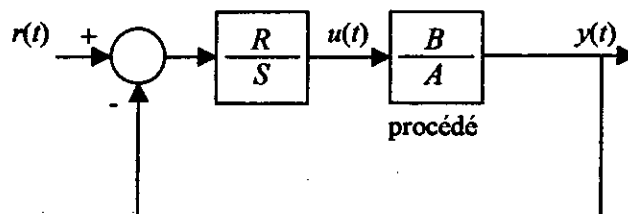
La régulation type *P.I.D.* est la plus ancienne technique de régulation et la plus répandue à l'échelle industrielle en raison de sa simplicité de mise en œuvre et de ses performances acceptables. La version de base du régulateur *P.I.D.* numérique résulte de la discrétisation du régulateur *P.I.D.* continu à actions *P.I* et *D* indépendantes. La méthode de calcul des régulateurs *P.I.D.* numériques (deux versions seront présentées) ne s'applique qu'aux procédés modélisables par une fonction de transfert continue de degré maximum égal à deux, avec ou sans retard pur. Dans le cas de l'existence du retard pur, il doit être inférieur à la période d'échantillonnage [1].

Le calcul des paramètres du régulateur *P.I.D.* numérique comporte plusieurs étapes :

- Détermination du modèle échantillonné du procédé discrétisé.
- Spécification des performances.
- Calcul des paramètres du régulateur (les coefficients des polynômes  $R(q^{-1})$  et  $S(q^{-1})$ ).

La discrétisation du *P.I.D.* classique conduit à un régulateur numérique ayant une structure canonique à 3 branches (*R-S-T*) avec  $T(q^{-1}) = R(q^{-1})$  (*P.I.D.* 1).

Le schéma bloc correspondant est donné dans la figure (1.2).



**Figure 1.2. Schéma bloc équivalent d'une boucle de régulation numérique utilisant le régulateur numérique *P.I.D.* 1**



La fonction de transfert en boucle fermée reliant la consigne  $r(t)$  et la sortie  $y(t)$  est :

$$H_{BF}(q^{-1}) = \frac{B(q^{-1})R(q^{-1})}{A(q^{-1})S(q^{-1}) + B(q^{-1})R(q^{-1})} = \frac{B(q^{-1})R(q^{-1})}{P(q^{-1})} \quad 1.20$$

où  $P(q^{-1})$  définit les pôles en boucle fermée (directement reliés aux performances souhaitées en régulation). Le produit  $B(q^{-1})R(q^{-1})$  définit les zéros en boucle fermée. Le régulateur *P.I.D.*1 introduit des zéros supplémentaires définis par  $R(q^{-1})$  qui vont dépendre de  $A(q^{-1})$ ,  $B(q^{-1})$  et  $P(q^{-1})$  et donc ne pourront pas être spécifiés a priori.

La spécification des performances désirées du système en boucle fermée se fait par le choix de  $P(q^{-1})$ .

$$\text{On choisit } P(q^{-1}) = 1 + p_1q^{-1} + p_2q^{-2} \quad 1.21$$

La méthode pour définir  $p_1$  et  $p_2$  consiste à considérer un modèle continu du 2<sup>ème</sup> ordre permettant d'obtenir un temps de montée ( $t_M$ ) et un dépassement maximal (*DM*). Ceci permet de déterminer les paramètres  $\omega_0$  et  $\xi$  du système du 2<sup>ème</sup> ordre, en s'assurant que la période d'échantillonnage  $T_e$  et la pulsation propre  $\omega_0$  vérifient la condition :

$$0.25 \leq \omega_0 T_e \leq 1.5 \quad ; \quad 0.7 \leq \xi \leq 1$$

Puis on calcule le modèle discrétisé avec bloqueur d'ordre zéro, le dénominateur de la fonction de transfert échantillonnée ainsi obtenu représentera le polynôme  $P(q^{-1})$ .

Pour calculer les paramètres du régulateur numérique, il faut résoudre l'équation suivante :

$$P(q^{-1}) = A(q^{-1})S(q^{-1}) + B(q^{-1})R(q^{-1}) \quad 1.22$$

pour  $P(q^{-1})$  donné par l'équation (1.21),  $A(q^{-1})$  et  $B(q^{-1})$  donnés par les équations suivantes :

$$A(q^{-1}) = 1 + a_1q^{-1} + a_2q^{-2} \quad 1.23$$

$$B(q^{-1}) = b_1q^{-1} + b_2q^{-2} \quad 1.24$$

Les structures  $R(q^{-1})$  et  $S(q^{-1})$  sont données respectivement par les équations :

$$R(q^{-1}) = r_0 + r_1q^{-1} + r_2q^{-2} \quad 1.25$$

$$S(q^{-1}) = (1 - q^{-1})(1 + s_1q^{-1}) \quad 1.26$$

puis, on calcule les valeurs des paramètres du *P.I.D.* numérique par l'équation matricielle suivante :

$$\begin{bmatrix} r_0 \\ r_1 \\ r_2 \\ s_1 \end{bmatrix} = \begin{bmatrix} b_1 & 0 & 0 & 1 \\ b_2 & b_1 & 0 & a_1 - 1 \\ 0 & b_2 & b_1 & a_2 - a_1 \\ 0 & 0 & b_2 & -a_2 \end{bmatrix}^{-1} \begin{bmatrix} p_1 + 1 - a_1 \\ p_2 + a_1 - a_2 \\ a_2 \\ 0 \end{bmatrix} \quad 1.27$$

Il existe une autre structure de *P.I.D.* numérique (*P.I.D.2*) qui n'introduit pas des zéros supplémentaires dans la fonction de transfert échantillonnée en boucle fermée. La fonction de transfert désirée en boucle fermée sera de la forme :

$$H_{BF}(q^{-1}) = \frac{P(1)B(q^{-1})}{B(1)P(q^{-1})} \quad 1.28$$

où  $B(q^{-1})$  sont les zéros du procédé qui resteront inchangés.  $P(q^{-1})$  définit les pôles désirés en boucle fermée et le terme  $P(1)/B(1)$  est introduit pour assurer un gain unitaire entre la consigne et la sortie.

Le régulateur aura la structure générale :

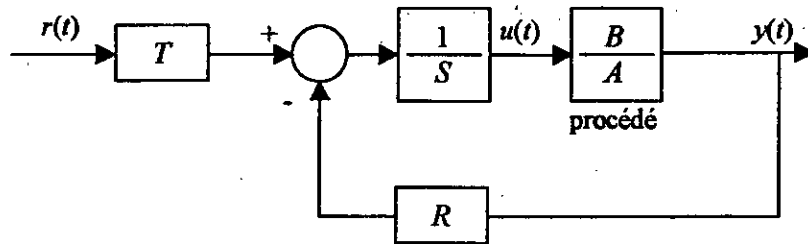
$$S(q^{-1})u(t) + R(q^{-1})y(t) = T(q^{-1})r(t) \quad 1.29$$

La fonction de transfert en boucle fermée sera alors :

$$H_{BF}(q^{-1}) = \frac{T(q^{-1})B(q^{-1})}{A(q^{-1})S(q^{-1}) + B(q^{-1})R(q^{-1})} = \frac{P(1)B(q^{-1})}{B(1)P(q^{-1})} \quad 1.30$$

Comme pour le *P.I.D.1*, les coefficients des polynômes  $S(q^{-1})$  et  $R(q^{-1})$  seront obtenus par la résolution de l'équation (1.22). Il résulte alors de l'équation (1.30)

$$T(q^{-1}) = \frac{P(1)}{B(1)} = \frac{B(1)R(1)}{B(1)} = R(1) \quad 1.31$$



**Figure 1.3. Schéma bloc équivalent d'une boucle de régulation numérique utilisant le régulateur numérique *P.I.D.2***

En conclusion le régulateur numérique *P.I.D.2* a les mêmes polynômes  $S(q^{-1})$  et  $R(q^{-1})$  que le régulateur numérique *P.I.D.1*, la seule différence étant que maintenant  $T(q^{-1}) = R(1)$  au lieu de  $R(q^{-1})$ , ce qui préserve le gain unitaire du système en boucle fermée sans pour autant introduire l'effet des zéros de  $R(q^{-1})$ .

## 1.4. Placement des pôles

Le calcul des paramètres du régulateur numérique *P.I.D.* (en particulier le *P.I.D.2*) est un cas particulier de la stratégie de "placement des pôles" [1].

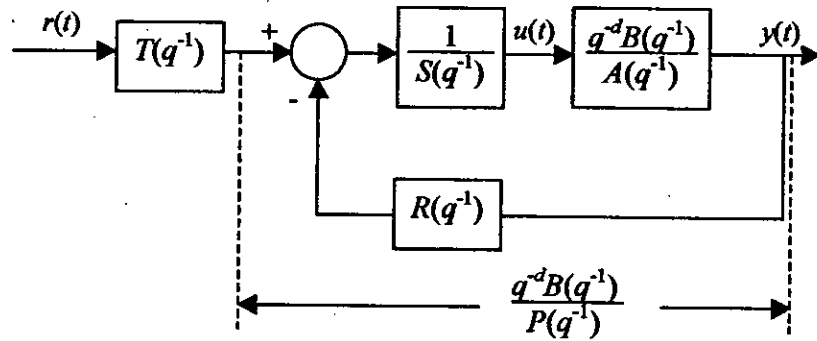
La stratégie de placement des pôles est appliquée aux modèles des procédés de la forme (1.1), elle permet de calculer un régulateur *R-S-T* pour des systèmes stables ou instables.

- ◆ Sans restriction sur les degrés des polynômes  $A(q^{-1})$ ,  $B(q^{-1})$  et la valeur du retard de procédé  $d$ .
- ◆ Les degrés  $n_A$ ,  $n_B$  et le retard  $d$  et les coefficients de  $A(q^{-1})$  et  $B(q^{-1})$  sont connus.
- ◆ Sans restriction sur les zéros du procédé (stables ou instables).
- ◆  $A(q^{-1})$  et  $B(q^{-1})$  n'ont pas des facteurs communs.

### 1.4.1. Structure

La structure du système en boucle fermée est donnée dans la figure suivante :

$$(v(t) \equiv 0)$$

Figure 1.4. Placement des pôles avec régulateur  $R$ - $S$ - $T$ 

La fonction de transfert en boucle fermée est donnée par :

$$H_{BF}(q^{-1}) = \frac{q^{-d}T(q^{-1})B(q^{-1})}{A(q^{-1})S(q^{-1}) + q^{-d}B(q^{-1})R(q^{-1})} = \frac{q^{-d}T(q^{-1})B(q^{-1})}{P(q^{-1})} \quad 1.32$$

où

$$P(q^{-1}) = A(q^{-1})S(q^{-1}) + q^{-d}B(q^{-1})R(q^{-1}) \quad 1.33$$

$$= 1 + p_1q^{-1} + p_2q^{-2} + \dots$$

définit les pôles en boucle fermée et respectivement le comportement en régulation. Et les polynômes  $S(q^{-1})$  et  $R(q^{-1})$  sont donnés par les équations (1.17) et (1.18).

### 1.4.2. Régulation (Calcul de $R(q^{-1})$ et $S(q^{-1})$ )

La boucle fermée est définie par :

- ◆ Les pôles désirés en boucle fermée.
- ◆ Le choix des parties  $H_R(q^{-1})$  et  $H_S(q^{-1})$ .

Les pôles en boucle fermée désirés sont choisis sous la forme  $P(q^{-1}) = P_D(q^{-1})P_F(q^{-1})$ . Où  $P_D(q^{-1})$  est choisi comme un polynôme de second ordre résultat de la discrétisation d'un système de second ordre continu ayant une fréquence  $\omega_0$  et un facteur d'amortissement  $\xi$  (ou des systèmes de premier ordre en cascade). Le rôle de  $P_F(q^{-1})$  est d'introduire l'effet d'un filtre pour certaines fréquences et d'améliorer la robustesse du contrôleur.

$H_R(q^{-1})$  et  $H_S(q^{-1})$  sont choisis en relation avec les performances nominales désirées. La nécessité d'avoir une erreur statique nulle pour une consigne ou une

perturbation en échelon implique la présence d'un intégrateur numérique dans la voie directe, (par exemple  $H_S(q^{-1}) = 1 - q^{-1}$ ).

Et pour des raisons de robustesse numérique (calcul de  $R(q^{-1})$  et  $S(q^{-1})$ ), un filtre  $H_R(q^{-1})$  est introduit en cascade avec  $R(q^{-1})$  [2].

Les polynômes  $S(q^{-1})$  et  $R(q^{-1})$  sont donnés par :

$$S(q^{-1}) = S'(q^{-1})H_S(q^{-1}) \quad 1.34$$

$$R(q^{-1}) = R'(q^{-1})H_R(q^{-1}) \quad 1.35$$

avec  $S'(q^{-1})$  et  $R'(q^{-1})$  sont les solutions de l'équation suivante :

$$P(q^{-1}) = A(q^{-1})S'(q^{-1})H_S(q^{-1}) + q^{-d-1}B^*(q^{-1})R'(q^{-1})H_R(q^{-1}) \quad 1.36$$

### 1.4.3. Poursuite (Calcul de $T(q^{-1})$ )

Dans le cas idéal, lors d'un changement de consigne, on souhaite faire suivre à la sortie du procédé  $y(t)$  une trajectoire  $y^*(t)$ . Cette trajectoire peut être mémorisée ou engendrée à chaque changement de consigne à l'aide d'un modèle de référence.

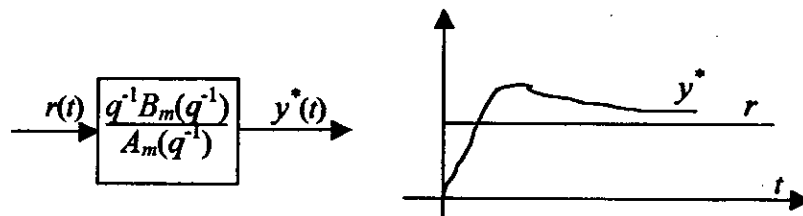


Figure 1.5. Génération de la trajectoire désirée  $y^*(t)$

La fonction de transfert du modèle de référence :

$$H_m(q^{-1}) = \frac{q^{-1}B_m(q^{-1})}{A_m(q^{-1})} \quad 1.37$$

elle se détermine à partir de performances souhaitées (temps de réponse, dépassement) en passant par exemple par un modèle continu normalisé du 2<sup>ème</sup> ordre (paramètres  $\omega_0$ ,  $\xi$ ), et

à partir de la fonction de transfert continue et la période d'échantillonnage  $T_e$ , on obtient par discrétisation la fonction de transfert échantillonnée du modèle de référence.

La fonction de transfert échantillonnée du modèle de référence sera de la forme :

$$\frac{q^{-1}B_m(q^{-1})}{A_m(q^{-1})} = \frac{q^{-1}(b_{m0} + b_{m1}q^{-1})}{1 + a_{m1}q^{-1} + a_{m2}q^{-2}} \quad 1.38$$

C'est cette fonction de transfert que le régulateur doit réaliser entre la consigne  $r(t)$  et la sortie  $y(t)$ . Néanmoins, dans le cas du "placement des pôles" ceci n'est pas possible pour les raisons suivantes :

- On conserve les zéros du procédé (polynôme  $B(q^{-1})$ ).
- Le retard  $d$  ne peut être compensé.

L'objectif sera alors d'approcher la trajectoire de référence :

$$y^*(t) = \frac{q^{-d-1}B_m(q^{-1})}{A_m(q^{-1})} r(t) \quad 1.39$$

Pour ceci on engendre d'abord  $y^*(t+d+1)$  à partir de  $r(t)$  :

$$y^*(t+d+1) = \frac{B_m(q^{-1})}{A_m(q^{-1})} r(t) \quad 1.40$$

La fonction de transfert en boucle fermée entre  $y(t)$  et  $y^*(t+d+1)$  est :

$$H_{CL}(q^{-1}) = \frac{q^{-d-1}T(q^{-1})B^*(q^{-1})}{P_D(q^{-1})P_F(q^{-1})} \quad 1.41$$

et on choisit  $T(q^{-1})$  pour assurer :

- Un gain statique unitaire entre  $y^*$  et  $y$ .
- La compensation de la dynamique de régulation  $P(q^{-1})$  (car la dynamique de régulation est en général différente de la dynamique de poursuite  $A_m(q^{-1})$ ).

Ceci conduit à choisir :

$$T(q^{-1}) = \beta P(q^{-1}) \quad 1.42$$

$$\text{avec } \beta = \begin{cases} \frac{1}{B^*(1)} & \text{si } B^*(1) \neq 0 \\ 1 & \text{si } B^*(1) = 0 \end{cases} \quad 1.43$$

Le schéma complet pour le placement des pôles est donné par la figure (1.6).

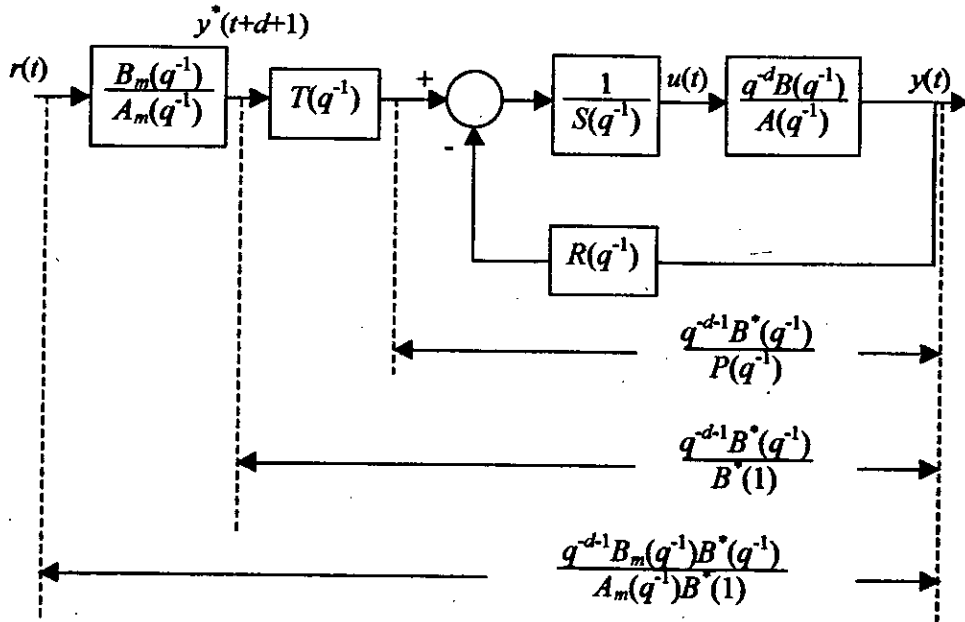


Figure 1.6. Placement des pôles – poursuite et régulation

### 1.4.4. Conception dans le domaine temporel

Les mêmes résultats sont obtenus si la conception commence dans le domaine temporel [2]. Dans le cas du placement des pôles, on a d'après l'équation (1.15) :

$$P(q^{-1})e_y(t+d+1) = 0; \quad t \geq 0 \tag{1.44}$$

$$\text{où } e_y(t+d+1) = y(t+d+1) - \beta B^*(q^{-1})y^*(t+d+1) \tag{1.45}$$

$$\text{avec } T(q^{-1}) = \beta P(q^{-1}) \tag{1.46}$$

et lorsqu'on remplace l'expression de  $T(q^{-1})$  dans l'équation (1.16), on obtient :

$$P(q^{-1})[u(t) - \beta A(q^{-1})y^*(t+d+1)] = P(q^{-1})e_u(t) = 0 \tag{1.47}$$

$$\text{où } e_u(t) = u(t) - \beta A(q^{-1})y^*(t+d+1); \quad t > 0 \tag{1.48}$$

Donc on peut écrire :

$$P(q^{-1})e_y(t) = P(q^{-1})e_u(t) = 0 \tag{1.49}$$

La loi de contrôle donnée par l'équation (1.6) devient :

$$S(q^{-1})u(t) = -R(q^{-1})y(t) + \left[ A(q^{-1})S(q^{-1}) + q^{-d-1}B^*(q^{-1})R(q^{-1}) \right] \beta y^*(t+d+1) \quad 1.50$$

en d'autre terme :

$$S(q^{-1})u(t) = -R(q^{-1}) \left[ y(t) - \beta B^*(q^{-1})y^*(t) \right] + \beta S(q^{-1})A(q^{-1})y^*(t+d+1) \quad 1.51$$

donc

$$u(t) = -\frac{R(q^{-1})}{S(q^{-1})}e_y(t) + \beta A(q^{-1})y^*(t+d+1) \quad 1.52$$

et on peut écrire aussi :

$$S(q^{-1})e_u(t) + R(q^{-1})e_y(t) = 0 \quad 1.53$$

La figure suivante montre le schéma équivalent du placement des pôles.

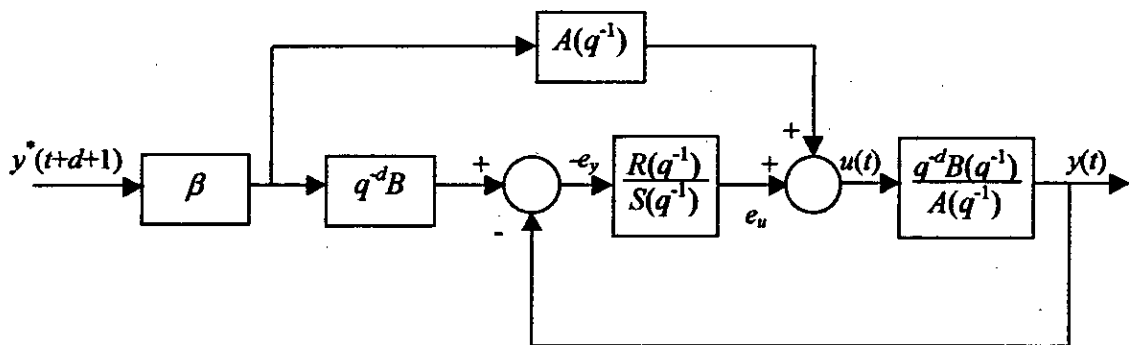


Figure 1.7. Schéma équivalent du placement des pôles

#### 1.4.5. Prédicteur de placement des pôles

Si on considère le modèle du procédé de l'équation (1.1) avec :

$$P(q^{-1}) = P_D(q^{-1})P_F(q^{-1}) \quad 1.54$$

Le contrôleur aura la forme :

$$S(q^{-1})u(t) = -R(q^{-1})y(t) + T(q^{-1})r(t) \quad 1.55$$

où  $S(q^{-1})$  et  $R(q^{-1})$  sont les solutions de :

$$P(q^{-1}) = P_D(q^{-1})P_F(q^{-1}) = A(q^{-1})S(q^{-1}) + q^{-d-1}B^*(q^{-1})R(q^{-1}) \quad 1.56$$



$$\text{et } T(q^{-1}) = \beta P(q^{-1}) = \frac{1}{B^*(1)} P_D(q^{-1}) P_F(q^{-1}) \quad 1.57$$

Le contrôleur de l'équation (1.55) est équivalent à :

$$P_F(q^{-1}) S_0(q^{-1}) u(t) = -R_0(q^{-1}) P_F(q^{-1}) \hat{y}(t+d/t) + P_F(q^{-1}) T_0(q^{-1}) r(t) \quad 1.58$$

$$P_F(q^{-1}) \hat{y}(t+d/t) = F(q^{-1}) y(t) + B(q^{-1}) E(q^{-1}) u(t) \quad 1.59$$

où  $R_0(q^{-1})$  et  $S_0(q^{-1})$  sont les solutions de :

$$P_D(q^{-1}) = A(q^{-1}) S_0(q^{-1}) + B(q^{-1}) R_0(q^{-1}) \quad 1.60$$

$F(q^{-1})$  et  $E(q^{-1})$  sont les solutions de :

$$P_F(q^{-1}) = A(q^{-1}) E(q^{-1}) + q^{-d} F(q^{-1}) \quad 1.61$$

$$\text{et } T_0(q^{-1}) = \frac{1}{B^*(1)} P_D(q^{-1}) \quad 1.62$$

Le schéma équivalent du placement des pôles (forme du prédicteur) est donné dans la figure (1.8).

## 1.5. Poursuite et régulation à objectifs indépendants

Cette méthode de calcul des régulateurs numériques permet d'obtenir le comportement désiré en poursuite (changement de consigne) indépendamment du comportement désiré en régulation (rejet d'une perturbation).

A la différence de la méthode du "placement des pôles", cette méthode conduit à la simplification des zéros du modèle échantillonné du procédé, ce qui permet de réaliser sans approximation les performances imposées en poursuite et en régulation.

Cette stratégie peut être considérée comme un cas particulier du placement des pôles. La seule différence est que  $B^*(q^{-1})$  a tous les zéros à l'intérieur du cercle unitaire (à cause de la simplification des zéros, cette méthode ne s'applique qu'aux modèles échantillonnés de procédé ayant des zéros stables), et  $B^*(q^{-1})$  et  $A(q^{-1})$  peuvent avoir des facteurs communs [1][2].

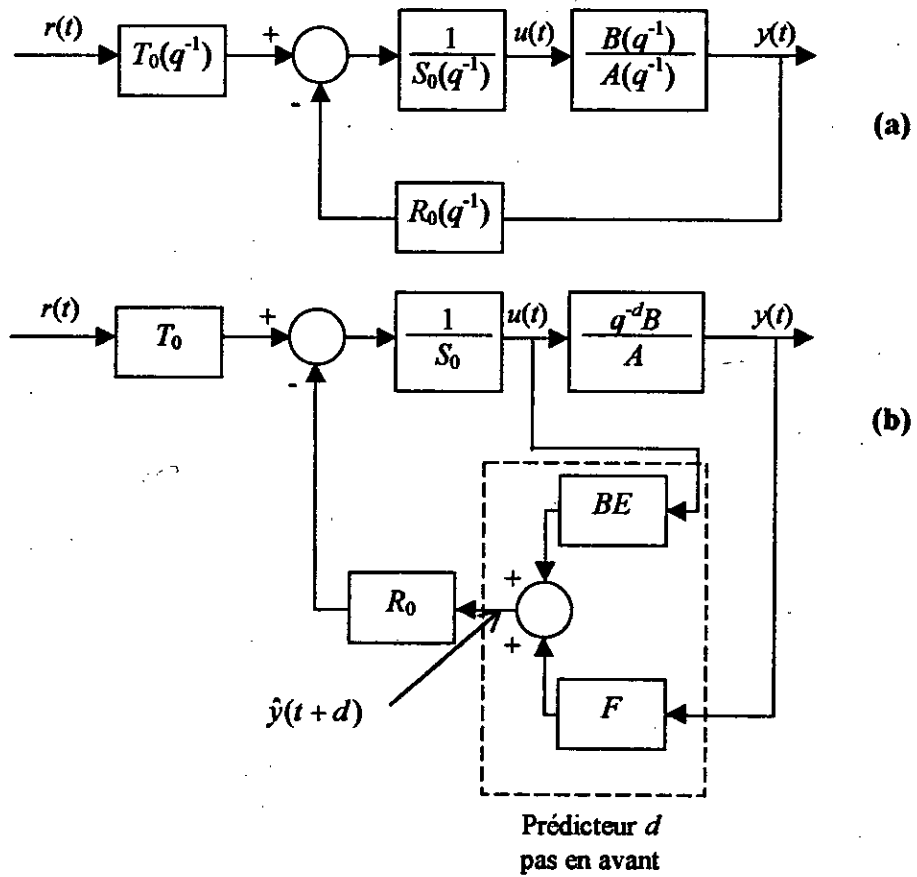


Figure 1.3. Schéma équivalent du placement des pôles (forme du prédicteur)  
 a) le cas  $d = 0$  et  $P_F(q^{-1}) = 1$     b) le cas général

### 1.5.1. Structure

La stratégie en boucle fermée est représentée dans la figure suivante :

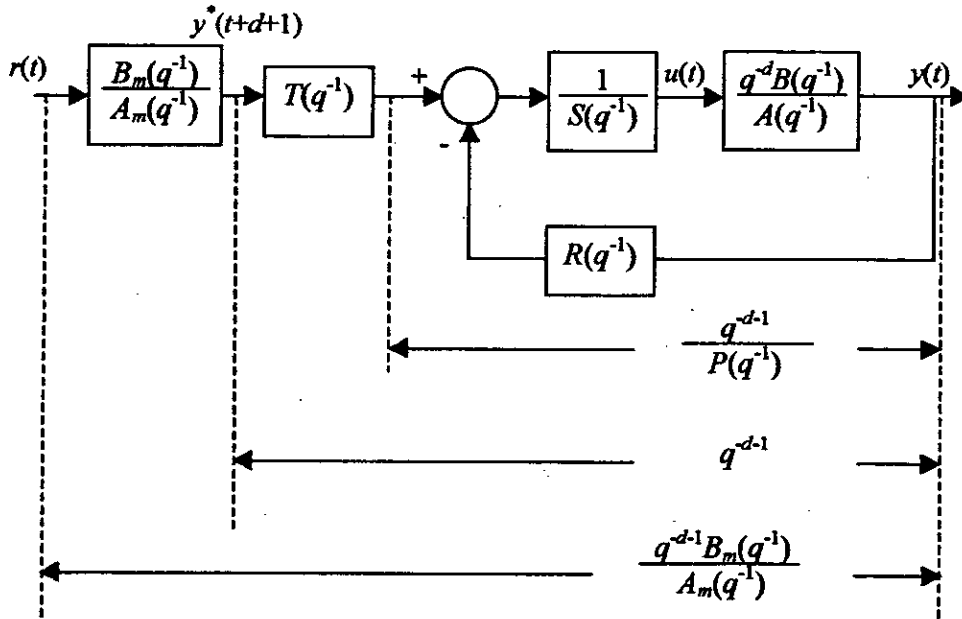


Figure 1.9. Poursuite et régulation à objectifs indépendants

Le calcul de  $R(q^{-1})$ ,  $S(q^{-1})$  et  $T(q^{-1})$  se fera en deux temps. Dans un premier temps à l'aide de  $R(q^{-1})$  et  $S(q^{-1})$ , on placera les pôles en boucle fermée aux valeurs désirées, spécifiées par un polynôme  $P(q^{-1})$  (objectifs en régulation) et on simplifiera les zéros du modèle échantillonné du procédé. Dans un deuxième temps, on déterminera le précompensateur  $T(q^{-1})$  pour obtenir en poursuite que la sortie du procédé suit la trajectoire de référence  $y^*$ .

### 1.5.2. Régulation (calcul de $R(q^{-1})$ et $S(q^{-1})$ )

La fonction de transfert en boucle fermée, sans le précompensateur  $T(q^{-1})$  est :

$$H'_{CL}(q^{-1}) = \frac{q^{-d-1} B^*(q^{-1})}{A(q^{-1})S(q^{-1}) + q^{-d-1} B^*(q^{-1})R(q^{-1})} \quad 1.63$$

On souhaite que les pôles soient ceux définis par  $P(q^{-1})$  et que les zéros du procédé soient simplifiés (pour pouvoir obtenir une poursuite parfaite).

$$H'_{CL}(q^{-1}) = \frac{q^{-d-1}}{P(q^{-1})} = \frac{q^{-d-1} B^*(q^{-1})}{B^*(q^{-1})P(q^{-1})} \quad 1.64$$

De l'équation (1.64), il résulte que les pôles en boucle fermée doivent en fait contenir en plus les zéros du modèle du procédé.

$P(q^{-1})$  étant spécifié, il résulte de l'équation (1.64) que pour calculer  $R(q^{-1})$  et  $S(q^{-1})$ , il faut résoudre l'équation suivante :

$$A(q^{-1})S(q^{-1}) + q^{-d-1}B^*(q^{-1})R(q^{-1}) = B^*(q^{-1})P(q^{-1}) \quad 1.65$$

Pour compenser  $B^*(q^{-1})$ ,  $S(q^{-1})$  devra avoir  $B^*(q^{-1})$  comme facteur commun.

$$S(q^{-1}) = s_0 + s_1q^{-1} + \dots + s_{n_s}q^{-n_s} = B^*(q^{-1})S'(q^{-1}) \quad 1.66$$

En introduisant l'expression de  $S(q^{-1})$  dans l'équation (1.65) et après simplification par  $B^*(q^{-1})$  on obtient :

$$A(q^{-1})S'(q^{-1}) + q^{-d-1}R(q^{-1}) = P(q^{-1}) \quad 1.67$$

Et comme pour le cas de placement des pôles, pour avoir une erreur statique nulle à une entrée ou perturbation échelon, la voie directe doit contenir un intégrateur numérique. Donc, on prend :

$$S(q^{-1}) = S'(q^{-1})H_S(q^{-1}) \quad 1.68$$

avec  $H_S(q^{-1}) = B^*(q^{-1})H'_S(q^{-1}) \quad 1.69$

on prend par exemple :  $H'_S(q^{-1}) = 1 - q^{-1}$

et aussi  $R(q^{-1}) = R'(q^{-1})H_R(q^{-1}) \quad 1.70$

avec  $S'(q^{-1})$  et  $R'(q^{-1})$  sont les solutions de l'équation suivante :

$$P(q^{-1}) = A(q^{-1})S'(q^{-1})H'_S(q^{-1}) + q^{-d-1}R'(q^{-1})H_R(q^{-1}) \quad 1.71$$

### 1.5.3. Poursuite (calcul de $T(q^{-1})$ )

Le précompensateur  $T(q^{-1})$  est calculé pour assurer entre  $r(t)$  et  $y(t)$  une fonction de transfert :

$$H_{BF}(q^{-1}) = \frac{q^{-d-1}B_m(q^{-1})}{A_m(q^{-1})} = \frac{q^{-d-1}B_m(q^{-1})T(q^{-1})}{A_m(q^{-1})P(q^{-1})} \quad 1.72$$

de l'équation (1.72), il résulte :  $T(q^{-1}) = P(q^{-1}) \quad 1.73$

L'entrée de  $T(q^{-1})$  étant constituée par la prédiction de la trajectoire  $y^*(t)$  obtenue par le passage de  $r(t)$  à travers un modèle de poursuite.

$$y^*(t+d+1) = \frac{B_m(q^{-1})}{A_m(q^{-1})} r(t) \quad 1.74$$

L'équation de régulation sera donnée par :

$$S(q^{-1})u(t) + R(q^{-1})y(t) = P(q^{-1})y^*(t+d+1) \quad 1.75$$

elle peut aussi être réécrite sous la forme :

$$u(t) = \frac{-R(q^{-1})y(t) + P(q^{-1})y^*(t+d+1)}{S(q^{-1})} \quad 1.76$$

#### 1.5.4. Conception dans le domaine temporel

Le problème de régulation et poursuite à objectifs indépendants peut être formulé comme suit :

$$u(t) = f_u \{y(t), y(t-1), \dots, u(t-1), u(t-2), \dots\}$$

$$\text{tel que} \quad P(q^{-1})[y(t+d+1) - y^*(t+d+1)] = 0 \quad ; t > 0 \quad 1.77$$

ou minimiser le critère :

$$J(t+d+1) = \left\{ P(q^{-1})[y(t+d+1) - y^*(t+d+1)] \right\}^2 \quad 1.78$$

##### Etape 1

On calcule à  $d+1$  pas en avant la prédiction filtrée de  $y(t)$ .

$$\begin{aligned} P(q^{-1})y(t+d+1) &= f \{y(t), y(t-1), \dots, u(t), u(t-1), \dots\} \\ &= F(q^{-1})y(t) + B^*(q^{-1})E(q^{-1})u(t) \end{aligned} \quad 1.79$$

où  $F(q^{-1})$  et  $E(q^{-1})$  sont les solutions de :

$$P(q^{-1}) = A(q^{-1})E(q^{-1}) + q^{-d-1}F(q^{-1}) \quad 1.80$$

##### Etape 2

On remplace  $P(q^{-1})y(t+d+1)$  donné par l'équation (1.79) dans l'équation (1.77), on obtient :

$$F(q^{-1})y(t) + B^*(q^{-1})E(q^{-1})u(t) - P(q^{-1})y^*(t+d+1) = 0 \quad 1.81$$

on obtient donc le contrôleur  $R$ - $S$ - $T$

$$S(q^{-1})u(t) = T(q^{-1})y^*(t+d+1) - R(q^{-1})y(t) \quad 1.82$$

$$\text{avec} \quad R(q^{-1}) = F(q^{-1}); \quad S(q^{-1}) = B^*(q^{-1})E(q^{-1}); \quad T(q^{-1}) = P(q^{-1}) \quad 1.83$$

### 1.5.5. Prédicteur du régulation et poursuite à objectifs indépendants

Si on considère le modèle du procédé de l'équation (1.1) avec  $P(q^{-1}) = P_D(q^{-1})P_F(q^{-1})$ .

Le contrôleur aura la forme :

$$S(q^{-1})u(t) = P(q^{-1})y^*(t+d+1) - R(q^{-1})y(t) \quad 1.84$$

$$\text{avec} \quad S(q^{-1}) = B^*(q^{-1})S'(q^{-1}) \quad 1.85$$

où  $S'(q^{-1})$  et  $R(q^{-1})$  sont les solutions de :

$$A(q^{-1})S'(q^{-1}) + q^{-d-1}R(q^{-1}) = P(q^{-1}) = P_D(q^{-1})P_F(q^{-1}) \quad 1.86$$

Le contrôleur de l'équation (1.84) est équivalent à :

$$P_F(q^{-1})S_0(q^{-1})u(t) = -R_0(q^{-1})P_F(q^{-1})\hat{y}(t+d/t) + P_F(q^{-1})P_D(q^{-1})y^*(t+d+1) \quad 1.87$$

$$P_F(q^{-1})\hat{y}(t+d/t) = F(q^{-1})y(t) + B(q^{-1})E(q^{-1})u(t) \quad 1.88$$

$$\text{où} \quad S_0(q^{-1}) = S_0'(q^{-1})B^*(q^{-1}) \quad 1.89$$

$R_0(q^{-1})$  et  $S_0'(q^{-1})$  sont les solutions de :

$$P_D(q^{-1}) = A(q^{-1})S_0'(q^{-1}) + q^{-1}R_0(q^{-1}) \quad 1.90$$

$F(q^{-1})$  et  $E(q^{-1})$  sont les solutions de :

$$P_F(q^{-1}) = A(q^{-1})E(q^{-1}) + q^{-d}F(q^{-1}) \quad 1.91$$

## 1.6. Poursuite et régulation avec pondération de l'entrée

Il s'agit de l'extension de la méthode de poursuite et régulation à objectifs indépendants aux modèles échantillonnés de procédés ayant un polynôme  $B(q^{-1})$  avec des zéros stables ou instables.

Cette extension s'obtient par une pondération de l'énergie de commande résultant du calcul par la méthode "poursuite et régulation à objectifs indépendants". L'effet de cette pondération se traduit par la modification du polynôme  $S(q^{-1})$  du régulateur qui a comme la non-simplification des zéros du procédé mais aussi une modification, par rapport aux spécifications, des pôles en boucle fermée.

Dans le cas où le modèle échantillonné du procédé a des zéros instables, cette méthode s'applique d'une façon satisfaisante si les pôles du procédé sont stables [1][2].

### 1.6.1. Structure

La structure du système en boucle fermée est représentée par la figure suivante :

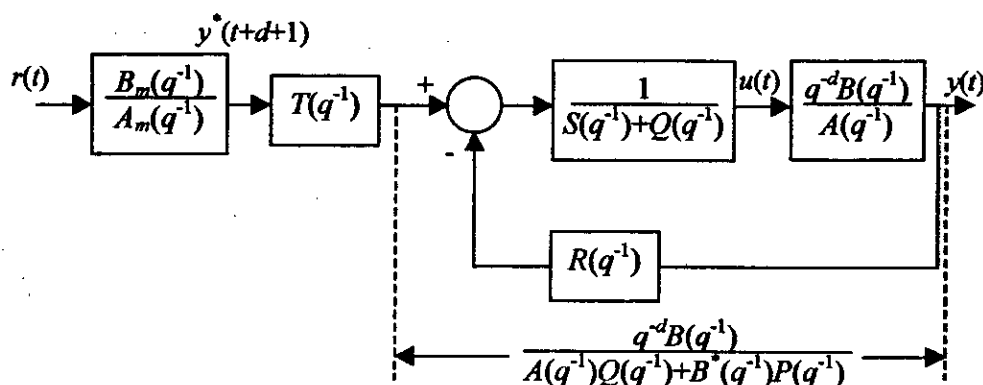


Figure 1.10. Poursuite et régulation avec pondération de l'entrée

Le polynôme  $S(q^{-1})$  qui intervient dans la méthode de "poursuite et régulation à objectifs indépendants" est remplacé par  $S(q^{-1}) + Q(q^{-1})$ .

Le polynôme de pondération des variations de l'entrée a la structure :

$$Q(q^{-1}) = \frac{\lambda(1 - q^{-1})}{1 + \alpha q^{-1}} \quad \lambda \geq 0 \quad 1.92$$

L'introduction du polynôme de pondération  $Q(q^{-1})$  modifiera les pôles désirés en boucle fermée spécifiés par le polynôme  $P(q^{-1})$ . Il faut donc s'assurer que le système en

boucle fermée reste stable et suffisamment amorti pour chaque valeur de  $\lambda$  (c'est à dire proche des performances spécifiées).

### 1.6.2. Calcul du régulateur

Le calcul du régulateur se fait en deux étapes :

#### Etape 1

On calcule  $R(q^{-1})$ ,  $S(q^{-1})$  pour  $Q(q^{-1}) = 0$  par la méthode de poursuite et régulation à objectifs indépendants même si  $B(q^{-1})$  a des zéros instables.

#### Etape 2

On introduit le polynôme  $Q(q^{-1})$  donné par l'équation (1.92) avec  $\lambda > 0$ .

La fonction de transfert en boucle fermée sans le précompensateur  $T(q^{-1})$  devient :

$$H'_{CL}(q^{-1}) = \frac{q^{-d} B(q^{-1})}{A(q^{-1})[S(q^{-1}) + Q(q^{-1})] + q^{-d-1} B^*(q^{-1})R(q^{-1})} \quad 1.93$$

$$= \frac{q^{-d} B(q^{-1})}{A(q^{-1})Q(q^{-1}) + B^*(q^{-1})P(q^{-1})}$$

$$\text{car} \quad B^*(q^{-1})P(q^{-1}) = A(q^{-1})S(q^{-1}) + q^{-d-1} B^*(q^{-1})R(q^{-1}) \quad 1.94$$

L'équation de la loi de commande est :

$$S(q^{-1})u(t) + R(q^{-1})y(t) + Q(q^{-1})u(t) = T(q^{-1})y^*(t+d+1) \quad 1.95$$

$$\text{avec} \quad T(q^{-1}) = P(q^{-1}) \quad 1.96$$

$$\text{donc} \quad u(t) = \frac{-R(q^{-1})y(t) + P(q^{-1})y^*(t+d+1)}{S(q^{-1}) + Q(q^{-1})} \quad 1.97$$

### 1.6.3. Conception dans le domaine temporel

Plusieurs critères sont utilisés pour faire la conception et obtenir la même loi de commande.

$$\text{Donc} \quad u(t) = f_u \{ y(t), y(t-1), \dots, u(t-1), u(t-2) \dots \}$$



tel que :

$$1) \quad \varepsilon_0(t+d+1) = P(q^{-1})\{y(t+d+1) - y^*(t+d+1)\} + \lambda Q(q^{-1})u(t) = 0 \quad 1.98$$

$$2) \quad J_1(t+d+1) = \left\{ P(q^{-1})\left(y(t+d+1) - y^*(t+d+1)\right) + \lambda Q(q^{-1})u(t) \right\}^2 \quad 1.99$$

est minimisé

$$3) \quad J_2(t+d+1) = \left\{ P(q^{-1})\left(y(t+d+1) - y^*(t+d+1)\right) \right\}^2 + b_1 \lambda \left\{ Q(q^{-1})u(t) \right\}^2 \quad 1.100$$

est minimisé

on prend comme forme typique  $Q(q^{-1}) = 1 - q^{-1}$ .

Même résultat, si on prend le critère :

$$J(t+d+1) = \left\{ P(q^{-1})\left(y(t+d+1) - y^*(t+d+1)\right)^2 + b_1 Q(q^{-1})\left(u(t)\right)^2 \right\} \quad 1.101$$

$$\text{avec } Q(q^{-1}) = \frac{\lambda(1 - q^{-1})}{1 + \alpha q^{-1}}$$

c'est à dire, minimiser la somme des carrés de l'écart filtré par  $P(q^{-1})$  et des carrés de l'entrée pondérée par  $b_1 Q(q^{-1})$ .

Pour avoir des meilleures performances de poursuite indépendamment de la valeur de  $\lambda$ , on choisit :

$$T(q^{-1}) = \beta \left( \lambda Q(q^{-1})A(q^{-1}) + B^*(q^{-1})P(q^{-1}) \right) \quad 1.102$$

$$\text{avec } Q(q^{-1}) = 1 - q^{-1} \quad \text{et} \quad \beta = \frac{1}{B^*(1)}$$

## 1.7. Poursuite et régulation à variance minimale

Cette stratégie concerne le calcul d'un régulateur optimal assurant la variance minimale de la variable réglée autour de la consigne dans le cas où le procédé est sujet à des perturbations aléatoires.

L'objectif de la Poursuite et régulation à variance minimale est de réduire la variance (l'écart type) de la sortie réglée autour de la valeur de la consigne soit quand elle est constante (régulation à variance minimale) soit quand elle est variable (poursuite à variance minimale) [1].

L'objectif de la variance minimale est de calculer  $u(t)$  qui minimise le critère :

$$E\left\{\left(y(t) - y^*(t)\right)^2\right\} \quad 1.103$$

$y(t) - y^*(t)$  représente l'écart par rapport à la valeur souhaitée à l'instant  $t$ .

Pour pouvoir résoudre ce problème, il faudra introduire des modèles de perturbation en plus de modèles de procédé. La structure retenue est le modèle *A.R.M.A.X* (voir annexe A).

$$A(q^{-1})y(t) = q^{-d}B(q^{-1})u(t) + C(q^{-1})e(t) \quad 1.104$$

$$\text{ou bien } A(q^{-1})y(t+d+1) = B^*(q^{-1})u(t) + C(q^{-1})e(t+d+1) \quad 1.105$$

$e(t)$  est une séquence de variables aléatoires gaussiennes indépendantes de valeur moyenne nulle et variance  $\sigma^2$ .

Les hypothèses données pour "la régulation et poursuite à objectifs indépendants" sont données pour la variance minimale, la principale hypothèse est que  $B^*(q^{-1})$  a tous les pôles à l'intérieur du cercle unitaire.

L'objectif est de calculer pour le modèle du procédé avec perturbation la commande  $u(t)$  :

$$u(t) = f_u\{y(t), y(t-1), \dots, u(t-1), u(t-2), \dots\}$$

qui minimise le critère :

$$\min_{u(t)} J(t+d+1) = E\left\{\left(y(t+d+1) - y^*(t+d+1)\right)^2\right\} \quad 1.106$$

Deux approches équivalentes peuvent être utilisées :

- 1) Minimisation directe du critère de l'équation (1.106).
- 2) Utilisation du théorème de séparation.

### 1.7.1. Conception directe

On utilise l'équation polynomiale :

$$C(q^{-1}) = A(q^{-1})E(q^{-1}) + q^{-d-1}F(q^{-1}) \quad 1.107$$

$$\text{où } E(q^{-1}) = 1 + e_1 q^{-1} + \dots + e_d q^{-d} \quad 1.108$$

$$F(q^{-1}) = f_0 + f_1 q^{-1} + \dots + f_{n_A-1} q^{-n_A+1} \quad 1.109$$

on multiplie l'équation (1.107) par le terme  $y(t+d+1)$ , on obtient :

$$C(q^{-1})y(t+d+1) = A(q^{-1})E(q^{-1})y(t+d+1) + q^{-d-1}F(q^{-1})y(t+d+1) \quad 1.110$$

après, on remplace le terme  $y(t+d+1)A(q^{-1})$  donné par l'équation (1.105) dans l'équation (1.110).

$$y(t+d+1) = \frac{F(q^{-1})}{C(q^{-1})}y(t) + \frac{B^*(q^{-1})E(q^{-1})}{C(q^{-1})}u(t) + E(q^{-1})e(t+d+1) \quad 1.111$$

puis on calcule  $u(t)$  qui minimise le critère (1.106), on obtient :

$$u(t) = \frac{-R(q^{-1})y(t) + T(q^{-1})y^*(t+d+1)}{S(q^{-1})} \quad 1.112$$

avec

$$S(q^{-1}) = B^*(q^{-1})E(q^{-1}) \quad 1.113$$

$$R(q^{-1}) = F(q^{-1}) \quad 1.114$$

$$T(q^{-1}) = C(q^{-1}) \quad 1.115$$

$E(q^{-1})$  et  $F(q^{-1})$  sont les solutions de (1.107).

## 1.7.2. Utilisation du théorème de séparation

La conception du contrôleur est faite en deux étapes :

### Etape 1

Calcul du prédicteur optimal  $\hat{y}(t+d+1/t)$  qui minimise le critère :

$$E\left\{\left(y(t+d+1) - \hat{y}(t+d+1)\right)^2\right\} \quad 1.116$$

$$\hat{y}(t+d+1) = \hat{y}(t+d+1/t) = \frac{F(q^{-1})}{C(q^{-1})}y(t) + \frac{B^*(q^{-1})E(q^{-1})}{C(q^{-1})}u(t) \quad 1.117$$

### Etape 2

Calcul de  $u(t)$  tel que  $\hat{y}(t+d+1)$  minimise le critère :

$$\min_{u(t)} J(t+d+1) = \left(\hat{y}(t+d+1) - y^*(t+d+1)\right)^2 \quad 1.118$$

$$\text{ou bien} \quad \hat{y}(t+d+1) - y^*(t+d+1) = 0 \quad 1.119$$

qui donne immédiatement l'équation :

$$S(q^{-1})u(t) = -R(q^{-1})y(t) + T(q^{-1})y^*(t+d+1) \quad 1.120$$

avec

$$S(q^{-1}) = B^*(q^{-1})E(q^{-1}) ; R(q^{-1}) = F(q^{-1}) ; T(q^{-1}) = C(q^{-1})$$

### 1.7.3. Rapprochement avec poursuite et régulation à objectifs indépendants

Si on compare le contrôleur donné par :

$$S(q^{-1})u(t) = -R(q^{-1})y(t) + T(q^{-1})y^*(t+d+1)$$

$$S(q^{-1}) = B^*(q^{-1})E(q^{-1}) ; R(q^{-1}) = F(q^{-1}) ; T(q^{-1}) = C(q^{-1})$$

avec le contrôleur pour la poursuite et régulation à objectifs indépendants

$$S(q^{-1})u(t) = -R(q^{-1})y(t) + T(q^{-1})y^*(t+d+1)$$

$$S(q^{-1}) = B^*(q^{-1})E(q^{-1}) ; R(q^{-1}) = F(q^{-1}) ; T(q^{-1}) = P(q^{-1})$$

On remarque que la loi de commande à variance minimale est la même que celle pour la poursuite et régulation à objectifs indépendants dans le cas déterministe si on choisit  $P(q^{-1}) = C(q^{-1})$

Donc en général, et en tenant compte de la similarité avec la méthode déterministe de "poursuite et régulation à objectifs indépendants", le calcul du régulateur sera donc identique en choisissant :  $P(q^{-1}) = C(q^{-1})$ . Ceci est résumé dans la figure (1.11).

Si on remplace  $u(t)$  donné par l'équation (1.112) dans l'équation (1.105), on trouve

$$y(t+d+1) - y^*(t+d+1) = E(q^{-1})e(t+d+1) \quad 1.121$$

on observe que l'erreur est un processus *MA* d'ordre  $d$ . Et pour  $d = 0$ , l'erreur est un bruit blanc (voir Annexe A).

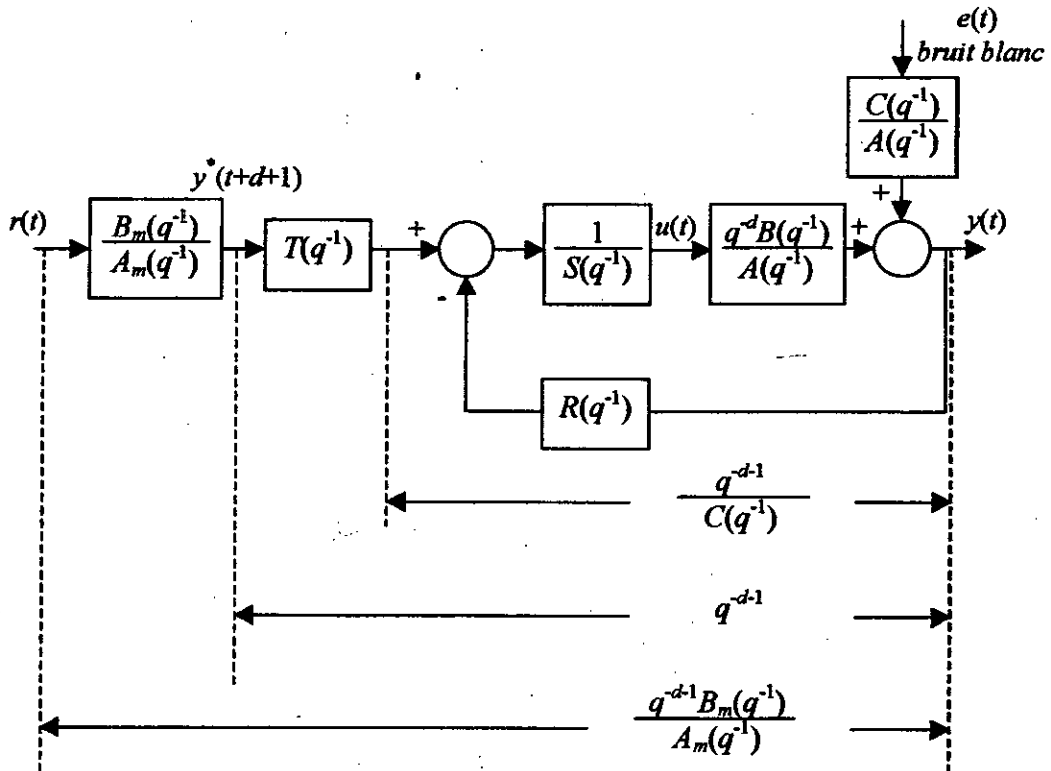


Figure 1.11. Poursuite et régulation à variance minimal

## 1.8. Poursuite et régulation à variance minimale généralisée

La stratégie de poursuite et régulation à variance minimale ne s'applique qu'aux procédés ayant un modèle échantillonné avec des zéros stables.

La stratégie de poursuite et régulation à variance minimale généralisée est une extension qui permet d'étendre l'algorithme de commande à variance minimale aux procédés ayant des modèles échantillonnés avec des zéros instables [1].

Cette méthode permet de calculer une commande  $u(t)$  qui minimise le critère suivant :

$$E \left\{ \left( y(t+d+1) - y^*(t+d+1) + \frac{Q(q^{-1})}{C(q^{-1})} u(t) \right)^2 \right\} \quad 1.122$$

$$\text{où } Q(q^{-1}) = \frac{\lambda(1-q^{-1})}{1+\alpha q^{-1}} = \frac{Q_n(q^{-1})}{Q_d(q^{-1})} \quad 1.123$$

### 1.8.1. Calcul du régulateur

On utilise l'équation :

$$A(q^{-1})S(q^{-1}) + q^{-d-1}B^*(q^{-1})R(q^{-1}) = B^*(q^{-1})C(q^{-1}) \quad 1.124$$

Pour calculer  $R(q^{-1})$  et  $S(q^{-1})$  dans le cas de poursuite et régulation à variance minimale. Et pour minimiser le critère  $E\left\{\left(y(t+d+1) - y^*(t+d+1)\right)^2\right\}$ , on applique la

commande 
$$u(t) = \frac{-R(q^{-1})y(t) + C(q^{-1})y^*(t+d+1)}{S(q^{-1})}$$
.

Pour cette méthode, on applique la commande suivante :

$$u(t) = \frac{-R(q^{-1})y(t) + C(q^{-1})y^*(t+d+1)}{S(q^{-1}) + Q(q^{-1})} \quad 1.125$$

qui minimise le critère  $E\left\{\left(y(t+d+1) - y^*(t+d+1) + \frac{Q(q^{-1})}{C(q^{-1})}u(t)\right)^2\right\}$

La structure du régulateur est la même que dans le cas de la poursuite et régulation avec pondération de l'entrée, en choisissant  $P(q^{-1}) = C(q^{-1})$ .

## 1.9. Conclusion

Dans ce chapitre, plusieurs méthodes de calcul de régulateurs numériques ont été présentées.

Tous les régulateurs numériques ont une structure canonique à trois branches ( $R$ - $S$ - $T$ ), correspondant à une loi de commande de la forme :

$$S(q^{-1})u(t) + R(q^{-1})y(t) = T(q^{-1})y^*(t+d+1)$$

qui peut être exprimé sous la forme :

$$u(t) = \frac{-R(q^{-1})y(t) + T(q^{-1})y^*(t+d+1)}{S(q^{-1})}$$

Le calcul du régulateur s'effectue en général en 2 étapes :

1. Calcul des polynômes  $S(q^{-1})$  et  $R(q^{-1})$  pour obtenir les performances désirées en régulation.

2. Calcul du polynôme  $T(q^{-1})$  pour approcher (ou atteindre parfaitement) les performances désirées en poursuite.

Pour des procédés caractérisés par des modèles d'ordre réduit ( $\leq 2$ ) et des retards petits (ou nuls), on peut utiliser les régulateurs *P.I.D.* numériques obtenus par la discrétisation des régulateurs *P.I.D.* analogiques. Les régulateurs *P.I.D.* numériques sont calculés directement à partir de la connaissance du modèle discret du procédé à réguler. Deux structures de régulateurs *P.I.D.* numériques ont été examinées. Ils diffèrent entre eux uniquement par le choix du polynôme  $T(q^{-1})$ . Il est recommandé d'utiliser  $T(q^{-1}) = R(1)$  (correspond aux régulateurs numériques *P.I.D.2*), qui offre de meilleures performances en poursuite.

La stratégie de commande *poursuite et régulation à objectifs indépendants* permet d'atteindre parfaitement les performances désirées en poursuite et en régulation, mais, pour qu'elle soit applicable, la fonction de transfert échantillonnée du modèle discret du procédé doit avoir des zéros stables.

La stratégie de commande *poursuite et régulation avec pondération de l'entrée* est une extension de la stratégie de *poursuite et régulation à objectifs indépendants* pour les cas où la fonction de transfert du modèle du procédé a des zéros instables.

La stratégie de commande *placement des pôles* s'applique aux procédés ayant des modèles discrets avec des zéros aussi bien stables qu'instables.

La stratégie de commande *poursuite et régulation à variance minimale* concerne le calcul d'un régulateur optimal assurant la variance minimale de la variable réglée autour de la consigne dans le cas où le procédé est sujet à des perturbations aléatoires, mais, la fonction de transfert échantillonnée du modèle discret du procédé doit avoir des zéros stables.

La stratégie de *poursuite et régulation à variance minimale généralisée* est une extension qui permet d'étendre l'algorithme de commande à variance minimale aux procédés ayant des modèles échantillonnés avec des zéros instables.

## Chapitre 2

# Le contrôle prédictif généralisé

### 2.1. Introduction

La commande prédictive est l'un des plus importants domaines de recherche. Le principe du contrôle prédictif est basé sur la prédiction du comportement futur du processus à contrôler. En effet, le problème se pose comme suit : Connaissant la sortie prédite du système, quelle commande à appliquer au système pour que sa sortie suit une trajectoire de référence ?

Les stratégies de contrôle présentées dans le chapitre 1 appartiennent à la classe "one step ahead predictive control", dans le sens qu'on s'intéresse seulement aux sorties du procédé  $d+1$  pas en avant, où  $d$  est un retard entier [1][2].

Le contrôle prédictif généralisé est considéré comme étant le plus populaire des méthodes de prédiction, il appartient à la classe " *long range predictive control* ", dans le sens qu'on s'intéresse aux sorties et aux entrées du procédé sur un certain horizon dans le futur au-delà de  $d+1$  [2].

Le *GPC* est une technique du contrôle prédictif à horizon fuyant. La commande est calculée de telle façon à minimiser un critère quadratique enveloppant les entrées et les sorties dans le futur. On calcule une séquence d'un signal de contrôle dans le futur mais on applique seulement la première valeur calculée. Puis, la procédure de l'optimisation est recommencée au prochain pas.



## 2.2. Généralisation

Le contrôle prédictif généralisé avec une formulation appropriée d'un critère quadratique peut être interprété comme généralisation de plusieurs stratégies de contrôle "one step ahead" [2].

Si on considère le modèle du procédé :

$$A(q^{-1})y(t+d+1) = B^*(q^{-1})u(t) + v(t+d+1) \quad 2.1$$

où  $v(t)$  est une perturbation (déterministe ou stochastique).

On considère le critère quadratique :

$$J(t, h_i, h_p, h_c) = E \left\{ \sum_{j=h_i}^{h_p} [P(q^{-1})y(t+j) - P(q^{-1})y^*(t+j)]^2 + \lambda [Q(q^{-1})u(t+j-h_i)]^2 \right\} \quad 2.2$$

avec :  $h_i$  : l'horizon initial.

$h_p$  : l'horizon de prédiction.

$\lambda \geq 0$  et  $P(q^{-1})$ ,  $Q(q^{-1})$  sont des opérateurs de transfert.

L'objectif du contrôle est de minimiser le critère (2.2) en prenant en compte :

$$U^T(t+h_c-1) = \{u(t), u(t+1), \dots, u(t+h_c-1)\}$$

sujet aux contraintes  $Q(q^{-1})u(t+i) = 0$  ;  $h_c \leq i \leq h_p$ , avec  $h_c$  : l'horizon de commande.

La forme du critère (2.2) permet d'établir un rapport avec les stratégies de contrôle "one step ahead".

1. Pour  $h_p = h_i = d+1$ ,  $h_c = 1$ ,  $\lambda = 0$

On obtient la poursuite et régulation à objectifs indépendants.

2. Pour  $h_p = h_i = d+1$ ,  $h_c = 1$ ,  $\lambda > 0$ ,  $Q(q^{-1}) \neq 0$

On obtient la poursuite et régulation avec pondération de l'entrée.

3. Pour  $h_p = h_i = d+1$ ,  $h_c = 1$ ,  $\lambda = 0$ ,  $P(q^{-1}) = 1$ ,  $v(t) = C(q^{-1})e(t)$

On obtient la poursuite et régulation à variance minimale. Cette méthode ne peut pas manipuler le cas des systèmes à phase non-minimale, elle a comme conséquence les efforts excessifs de la commande.

4. Pour  $h_p = h_i = d+1$ ,  $h_c = 1$ ,  $\lambda > 0$ ,  $P(q^{-1}) = 1$ ,  $Q(q^{-1}) = \frac{1-q^{-1}}{C(q^{-1})}$

On obtient la poursuite et régulation à variance minimale généralisée. En ajoutant dans cette méthode une pondération de l'énergie de commande, les

Les systèmes à phase non-minimale peuvent être traités et des efforts excessifs de la commande peuvent être évités. Cette méthode est toujours une méthode d'un seul pas en avant, dans le sens qu'elle réduit au minimum une fonction quadratique d'une seule valeur de la sortie au temps  $d+1$ , où  $d$  le retard du processus. Cette méthode est robuste contre les suppositions de l'ordre du modèle mais, peut s'exécuter mal si le temps mort du procédé est variable ou inconnu.

La méthode de placement des pôles citée dans le chapitre 1, dans laquelle l'équation Diophantine est résolue numériquement comme un pas liant l'identification du procédé et le calcul du contrôleur, est sensible aux changements de l'ordre du système.

Le *GPC* est conçu pour parer à ces problèmes dans un seul algorithme. C'est une méthode robuste, qui permet de contrôler un système instable ou à phase non-minimale et de compenser l'effet du retard du système. Elle est capable de contrôler un processus avec des paramètres variables, temps mort variable et un ordre du modèle variable.

### 2.3. Le contrôle prédictif généralisé

Plusieurs algorithmes du *GPC* ont été développés par plusieurs auteurs [11][12][13][15]. L'idée de base est de combiner les prédictions du comportement futur du système avec un retour de contrôle.

La commande prédictive suit trois étapes de base [15] :

- 1) **La prédiction de sortie** : le modèle du système est employé pour prédire le comportement futur de la sortie en se basant sur les données disponibles au temps courant  $t$ .
- 2) **Calcul de la commande** : la commande est calculée de sorte que la sortie prédite soit aussi proche que possible de la sortie désirée.
- 3) **Fermeture de la boucle de rétroaction (feedback)** : la boucle est fermée par l'application de la commande proposée, calculée dans l'étape précédente, au système.

Les trois étapes ci-dessus sont effectuées à chaque instant, une approche connue sous le nom "*receding horizon strategy*".

### 2.3.1. La loi de commande prédictive

Pour pouvoir commander un procédé, il faut avoir un modèle de base pour la synthèse de la loi de commande [11][12].

Une représentation linéaire est donnée par l'expression suivante :

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + x(t) \quad 2.3$$

où  $u(t)$  est l'entrée de commande,  $y(t)$  la variable mesurée de la sortie,  $A(q^{-1})$  et  $B(q^{-1})$  sont des polynômes en  $q^{-1}$ .

$$A(q^{-1}) = 1 + a_1q^{-1} + \dots + a_{n_a}q^{-n_a} \quad 2.4$$

$$B(q^{-1}) = b_0 + b_1q^{-1} + \dots + b_{n_b}q^{-n_b} \quad 2.5$$

$n_a$  et  $n_b$  étant des nombres entiers représentant les degrés respectivement de  $A(q^{-1})$  et  $B(q^{-1})$ ,  $q^{-1}$  est l'opérateur de retard, tel que :  $q^{-1}y(t) = y(t-1)$

$x(t)$  est un terme relatif à une perturbation survenant à un instant  $t$  quelconque.

$$\text{Dans le cas du GPC, } x(t) = C(q^{-1})e(t) / \Delta \quad 2.6$$

En remplaçant (2.6) dans (2.3), on aura :

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + C(q^{-1})e(t) / \Delta \quad 2.7$$

$$\text{avec } C(q^{-1}) = 1 + c_1q^{-1} + \dots + c_{n_c}q^{-n_c} \quad 2.8$$

( $n_c$  est le degré de  $C(q^{-1})$ ),  $\Delta = 1 - q^{-1}$  et  $C(q^{-1})$  est un polynôme strictement stable,  $e(t)$  représente un bruit blanc gaussien de moyenne nulle. (voir annexe A)

Le modèle décrit par l'équation (2.7) représentatif du système à contrôler est appelé *CARIMA* (Controlled-Auto-Regressive-Integrated-Moving-Average), ce modèle constitue le modèle de base de la méthode *GPC* à partir du quel sera dérivée l'expression de la loi de commande.

La méthode du *GPC* consiste à calculer les prédictions de la sortie pour tout l'intervalle  $[N_1, N_2]$ , ensuite à utiliser ces prédictions pour calculer la commande  $u(t)$  qui minimisera un critère  $J$ .

La fonction coût à minimiser est :

$$J(N_1, N_2) = E \left\{ \sum_{j=N_1}^{N_2} [y(t+j) - w(t+j)]^2 + \sum_{j=1}^{N_2} \lambda(j) [\Delta u(t+j-1)]^2 \right\} \quad 2.9$$

où  $N_1$  l'horizon minimal de prédiction,  $N_2$  l'horizon maximal de prédiction. Le signal  $w(t)$  est le signal de référence que le système doit suivre,  $\lambda$  est le coefficient de pondération du signal de commande. La fonction coût est formée de deux termes, le premier représente l'erreur quadratique entre la sortie du système et la référence et le deuxième terme est une pénalité sur la commande, le paramètre  $\lambda$  permet de maintenir l'énergie de commande dans des intervalles raisonnables.

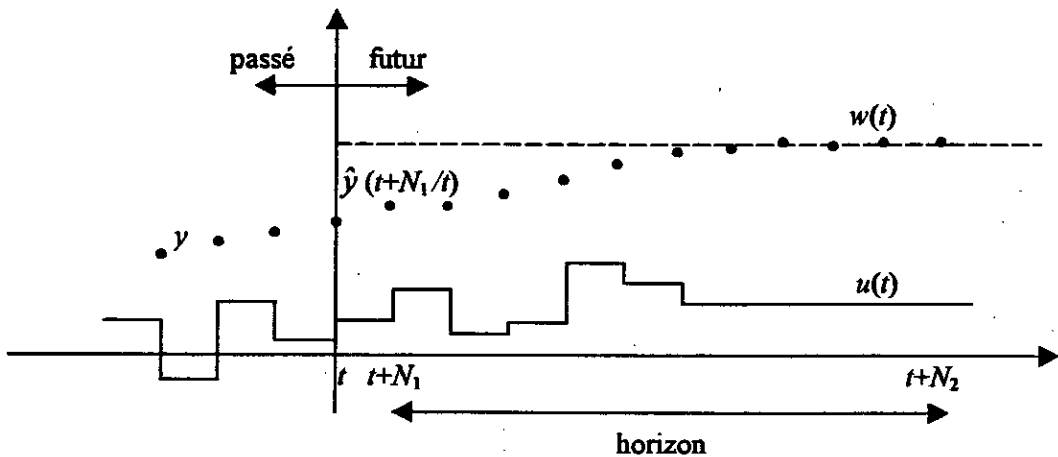


Figure 2.1.

### 2.3.2. Le prédicteur à j-pas

La technique *GPC* repose sur la prédiction à  $j$  pas de la variable de sortie du système. Pour déduire les prédictions de la sortie nous utilisons les équations Diophantine.

On choisit  $C(q^{-1}) = 1$  (le cas général est détaillé dans les paragraphes suivants), l'équation (2.7) s'écrit :

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + e(t)/\Delta \quad 2.10$$

pour dériver le prédicteur  $j$ -pas en avant, on considère l'identité :

$$1 = E_j(q^{-1})A(q^{-1})\Delta(q^{-1}) + q^{-j}F_j(q^{-1}) \quad 2.11$$

on multiplie l'équation (2.10) par le terme  $E_j(q^{-1})\Delta(q^{-1})q^j$ , on obtient la sortie  $y(t+j)$ .

$$y(t+j) = E_j(q^{-1})B(q^{-1})\Delta(q^{-1})u(t+j-1) + F_j(q^{-1})y(t) + E_j(q^{-1})e(t+j) \quad 2.12$$

avec  $\deg(E_j(q^{-1})) = j-1$

Dans l'équation (2.12), le dernier terme contient une information indépendante des signaux mesurables à l'instant  $t$ . De ce fait la prédiction  $\hat{y}(t+j)$  à l'instant  $t$  est obtenue en remplaçant le dernier terme par zéro.

$$\hat{y}(t+j) = G_j(q^{-1})\Delta(q^{-1})u(t+j-1) + F_j(q^{-1})y(t) \quad 2.13$$

$$\text{avec } G_j(q^{-1}) = E_j(q^{-1})B(q^{-1}) \quad 2.14$$

Pour la simplicité, on prend  $N_1 = 1$ ,  $N_2 = N$  et  $\lambda(j) = \lambda$ .

De l'équation (2.12), on obtient :

$$\begin{aligned} y(t+1) &= G_1(q^{-1})\Delta u(t) + F_1(q^{-1})y(t) + E_1(q^{-1})e(t+1) \\ y(t+2) &= G_2(q^{-1})\Delta u(t+1) + F_2(q^{-1})y(t) + E_2(q^{-1})e(t+2) \\ &\vdots \\ y(t+N) &= G_N(q^{-1})\Delta u(t+N-1) + F_N(q^{-1})y(t) + E_N(q^{-1})e(t+N) \end{aligned} \quad 2.15$$

$y(t+j)$  contient des valeurs connues à l'instant  $t$  et aussi des futures entrées de contrôle et sans oublier le terme  $e(t+j)$ .

Donc on calcule le prédicteur  $f(t+j)$  en se basant seulement sur les données disponibles à l'instant  $t$ , on obtient :

$$\begin{aligned} f(t+1) &= [G_1(q^{-1}) - g_{10}] \Delta u(t) + F_1(q^{-1})y(t) \\ f(t+2) &= q[G_2(q^{-1}) - q^{-1}g_{21} - g_{20}] \Delta u(t) + F_2(q^{-1})y(t) \end{aligned} \quad 2.16$$

$$\text{avec } G_i(q^{-1}) = g_{i0} + g_{i1}q^{-1} + \dots \quad 2.17$$

A partir de (2.16), on peut écrire :

$$\hat{Y} = G\tilde{U} + F \quad 2.18$$

où  $\hat{Y}$ ,  $\tilde{U}$  et  $F$  sont des vecteurs de dimension  $(N \times 1)$

$$\begin{aligned}
 \hat{Y} &= [\hat{y}(t+1), \hat{y}(t+2), \dots, \hat{y}(t+N)]^T \\
 \tilde{U} &= [\Delta u(t), \Delta u(t+1), \dots, \Delta u(t+N-1)]^T \\
 F &= [f(t+1), f(t+2), \dots, f(t+N)]^T
 \end{aligned}
 \tag{2.19}$$

La matrice  $G$  est triangulaire inférieure ( $N \times N$ )

$$G = \begin{bmatrix} g_0 & 0 & \dots & \dots & 0 \\ g_1 & g_0 & \dots & \dots & 0 \\ \cdot & \cdot & \dots & \dots & \cdot \\ \cdot & \cdot & \dots & \dots & \cdot \\ g_{N-1} & g_{N-2} & \dots & \dots & g_0 \end{bmatrix}
 \tag{2.20}$$

avec  $g_{ij} = g_j$  pour  $j = 0, 1, 2, \dots < i$

Le critère de l'équation (2.9) peut être écrit sous forme matricielle comme suit :

$$J_1 = \left\{ (G\tilde{U} + F - W)^T (G\tilde{U} + F - W) + \lambda \tilde{U}^T \tilde{U} \right\}
 \tag{2.21}$$

avec :  $W = [w(t+1), w(t+2), \dots, w(t+N)]^T$

$J_1$  est minimal si  $\frac{\delta J_1}{\delta \tilde{U}} = 0$ ,  $\tilde{U}$  est donné par :

$$\tilde{U} = (G^T G + \lambda I)^{-1} G^T (W - F)
 \tag{2.22}$$

Cette dernière équation fournit les incréments futurs de contrôle pour les instants  $t$  à  $t+N-1$ , en se basant sur les informations disponibles à l'instant  $t$ .

Seul l'incrément  $\Delta u(t)$ , le premier élément de  $\tilde{U}$ , sera appliqué au système et la commande se présente alors sous la forme :

$$u(t) = u(t-1) + \tilde{g}^T (W - F)
 \tag{2.23}$$

avec  $\tilde{g}^T$  est la première ligne de  $(G^T G + \lambda I)^{-1} G^T$  et l'action intégrante est fournie pour avoir une erreur nulle.

La dimension de la matrice  $G$  est ( $N \times N$ ). Dans le cas non-adaptatif, on a besoin de l'inversion une seule fois, mais dans la version d'auto-réglage la charge computationnelle d'inversion à chaque pas serait excessive. De plus, si une mauvaise valeur du temps mort est supposée,  $G^T G$  est singulière et par conséquent une valeur non nulle de  $\lambda$  serait exigée.

Dans le cas précédent, on a pris l'horizon de commande  $N_u$  égal à l'horizon maximal  $N_2$ . Donc, pour réduire le fardeau computationnel, on suppose qu'après un intervalle  $N_u < N_2$  les incréments futurs de contrôle sont supposés nuls, c'est à dire :

$$\Delta u(t+j-1) = 0 \text{ pour } j > N_u$$

Le critère de l'équation (2.9) devient :

$$J(N_1, N_2, N_u) = E \left\{ \sum_{j=N_1}^{N_2} [y(t+j) - w(t+j)]^2 + \sum_{j=1}^{N_u} \lambda(j) [\Delta u(t+j-1)]^2 \right\} \quad 2.24$$

et le prédicteur prendra la forme :

$$\hat{Y} = G_1 \tilde{U} + F \quad 2.25$$

avec

$$G_1 = \begin{bmatrix} g_0 & 0 & \dots & \dots & 0 \\ g_1 & g_0 & \dots & \dots & 0 \\ \cdot & \cdot & \dots & \dots & \cdot \\ \cdot & \cdot & \dots & \dots & \cdot \\ g_{N-1} & g_{N-2} & \dots & \dots & g_{N-N_u} \end{bmatrix} \quad 2.26$$

avec  $N_1 = 1$ ,  $N_2 = N$ ,  $\lambda(j) = \lambda$  et la matrice  $G_1$  est de dimension  $(N \times N_u)$ .

Le vecteur  $\tilde{U}$  qui minimisera le critère  $J$  donné par l'équation (2.24) ou le critère  $J_1$  donné par cette équation :

$$J_1 = \left\{ (G_1 \tilde{U} + F - W)^T (G_1 \tilde{U} + F - W) + \lambda \tilde{U}^T \tilde{U} \right\} \quad 2.27$$

est donné par la formule suivante :

$$\tilde{U} = (G_1^T G_1 + \lambda I)^{-1} G_1^T (W - F) \quad 2.28$$

L'équation (2.28) produit les incréments futurs de contrôle des instants  $t$  à  $t+N_u-1$ , en se basant toujours sur l'information disponible à l'instant  $t$ . Dans le cas général, on prend l'équation (2.7) avec  $C(q^{-1}) \neq 1$ , le prédicteur à  $j$ -pas est obtenu en séparant les informations passées et futures en utilisant deux équations Diophantine.

Pour calculer la prédiction de sortie de  $j$ -pas en avant, nous avons l'équation suivante :

$$C(q^{-1}) = E_j(q^{-1})A(q^{-1})\Delta(q^{-1}) + q^{-j}F_j(q^{-1}) \quad 2.29$$

où chaque variable est un polynôme en  $q^{-1}$  et  $\deg(E_j(q^{-1})) = j-1$

On multiplie l'équation (2.7) par le terme  $E_j(q^{-1})\Delta(q^{-1})q^j$ , on obtient la sortie  $y(t+j)$ .

$$y(t+j) = \frac{E_j(q^{-1})B(q^{-1})}{C(q^{-1})}\Delta u(t+j-1) + \frac{F_j(q^{-1})}{C(q^{-1})}y(t) + E_j(q^{-1})e(t+j) \quad 2.30$$

où le dernier terme contient une information indépendante des signaux mesurables à l'instant  $t$ . De ce fait la prédiction  $\hat{y}(t+j)$  à l'instant  $t$  est obtenue en remplaçant le dernier terme par zéro, produisant ainsi :

$$\hat{y}(t+j) = \frac{E_j(q^{-1})B(q^{-1})}{C(q^{-1})}\Delta u(t+j-1) + \frac{F_j(q^{-1})}{C(q^{-1})}y(t) \quad 2.31$$

Dans cette expression  $\hat{y}(t+j)$  est une fonction des valeurs connues du signal à l'instant  $t$  et aussi des futures entrées de contrôle qui sont à calculer, alors nous utilisons la seconde équation Diophantine pour différencier entre les valeurs passées et futures de contrôle, elle est donnée par :

$$G_j(q^{-1}) = G'_j(q^{-1})C(q^{-1}) + q^{-j}\Gamma_j(q^{-1}) \quad 2.32$$

$$\text{avec } G_j(q^{-1}) = E_j(q^{-1})B(q^{-1})$$

où  $E_j(q^{-1})$ ,  $F_j(q^{-1})$ ,  $G'_j(q^{-1})$  et  $\Gamma_j(q^{-1})$  sont à calculer en résolvant les deux équations Diophantine (2.29) et (2.32). (voir annexe C)

Ainsi l'expression de la prédiction est donnée par :

$$\hat{y}(t+j) = G'_j(q^{-1})\Delta u(t+j-1) + \frac{\Gamma_j(q^{-1})}{C(q^{-1})}\Delta u(t-1) + \frac{F_j(q^{-1})}{C(q^{-1})}y(t) \quad 2.33$$

sous une autre forme :

$$\hat{y}(t+j) = G'_j(q^{-1})\Delta u(t+j-1) + f(t+j) \quad 2.34$$

$$\text{avec } f(t+j) = \frac{\Gamma_j(q^{-1})}{C(q^{-1})}\Delta u(t-1) + \frac{F_j(q^{-1})}{C(q^{-1})}y(t) \quad 2.35$$

Le premier terme de (2.33) est une fonction des futurs incréments de commande à optimiser, le deuxième terme contient les informations passées.



Soient :

- Le vecteur  $F$  composé des prédictions de la réponse libre :

$$F = [f(t+1), f(t+2), \dots, f(t+N_2)]^T \quad 2.36$$

- Le vecteur des incréments futurs de contrôle  $\tilde{U}$  :

$$\tilde{U} = [\Delta u(t), \Delta u(t+1), \dots, \Delta u(t+N_u-1)]^T \quad 2.37$$

- Et enfin, le vecteur des sorties prédites du système sous contrôle :

$$\hat{Y} = [\hat{y}(t+1), \hat{y}(t+2), \dots, \hat{y}(t+N_2)]^T \quad 2.38$$

De l'équation de prédiction (2.34) la relation d'entrée-sortie des prédictions du système est donnée par :

$$\hat{Y} = G_1 \tilde{U} + F \quad 2.39$$

De la relation (2.39) la matrice  $G_1$  est de dimension  $(N_2 \times N_u)$  avec  $N_u \leq N_2$ .

Si  $N_u \leq N_2$ , on impose une contrainte supplémentaire qui caractérise la commande GPC, cette contrainte est que la commande ne change plus au-delà de l'horizon de commande.

De l'équation (2.39), la fonction de coût peut s'écrire :

$$J_1 = \{(G_1 \tilde{U} + F - W)^T (G_1 \tilde{U} + F - W) + \lambda \tilde{U}^T \tilde{U}\} \quad 2.40$$

et le vecteur  $\tilde{U}$  est donné par cette équation :

$$\tilde{U} = (G_1^T G_1 + \lambda I)^{-1} G_1^T (W - F) \quad 2.41$$

### 2.3.3. Choix de l'horizon de sortie et de l'horizon de contrôle

Une variété des modèles des procédés (stable, instable, à phase minimale, à phase non minimale, à temps mort variable...) a amenée les chercheurs à penser comment l'horizon initial  $N_1$ , l'horizon de prédiction  $N_2$  et l'horizon de commande  $N_u$  devraient être sélectionnés le mieux [11][12].

**Règles :****•  $N_1$  : l'horizon minimal de sortie**

Si le temps mort  $d$  du procédé est connu exactement,  $N_1$  prend cette valeur. Mais quand  $d$  est non connu ou variable  $N_1$  prend la valeur 1.

**•  $N_2$  : l'horizon maximal de sortie**

Une valeur plus grande de  $N_2$  est suggérée correspond au temps de montée du procédé.

**•  $N_u$  : l'horizon de commande**

C'est un paramètre de conception important. Pour un procédé stable en boucle ouverte, une valeur de  $N_u = 1$  donne un contrôle généralement acceptable, l'augmentation de  $N_u$  rend le contrôle et la réponse de la sortie correspondante plus actives jusqu'à ce qu'une étape soit atteinte où toute augmentation supplémentaire de  $N_u$  n'a aucune influence.

Pour un système instable, le bon contrôle est réalisé quand  $N_u$  est au moins égal au nombre des pôles instables.

Pour les hautes performances, une valeur plus grande de  $N_u$  est souhaitable.

## 2.4. Simulation

### 2.4.1. Simulation 1

L'objectif final de cette application (chariot avec pendule inversé) consiste à asservir la position du chariot tout en maintenant le pendule inversé dans sa position verticale. Dans un premier temps, on s'imposera uniquement de maintenir le bras du pendule inversé en position verticale ce qui revient à ne considérer que le degré de liberté selon  $\theta$ . Puis dans un second temps, on considère l'asservissement complet en prenant en compte les deux degrés de liberté,  $x$  et  $\theta$ .

avec :

- Le chariot est de masse  $M$ , la barre a une petite masse  $m$ .
- La longueur de la barre suspendue de section uniforme est  $l$ .
- Le chariot se déplace selon l'axe  $x$  et la barre suspendue pivote d'un angle  $\theta$  par rapport à la verticale de son point d'attache au chariot.
- La force appliquée au chariot selon l'axe  $x$  est notée  $U(t)$ .
- Le système est supposé sans frottement.

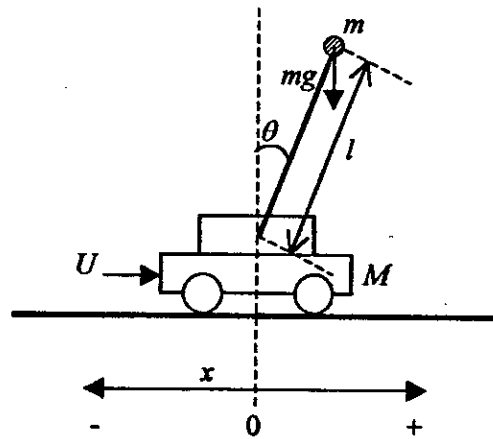


Figure 2.2.

### 2.4.1.1. Modélisation du système à 2 degré de liberté

On utilise pour cela la méthode de Lagrange définie par les équations suivantes :

$$\begin{cases} L = E_c - E_p \\ \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{q}} \right) - \frac{\partial L}{\partial q} + \frac{\partial D}{\partial \dot{q}} = U_q \end{cases} \quad 2.42$$

avec :

- ◆  $q$  : vecteur d'état, tel que  $q = [x(t), \theta(t)]^T$
- ◆  $D$  : énergie dissipée par frottements.
- ◆  $U_q$  : force généralisée dans la direction du degré de liberté  $q$ .
- ◆  $E_c$  : énergie cinétique.
- ◆  $E_p$  : énergie potentielle.

#### 2.4.1.1.1. Energie cinétique du système en mouvement

L'énergie cinétique de l'ensemble chariot plus pendule est exprimée par l'équation,

$$E_c = \frac{1}{2} M \dot{x}^2 + \frac{1}{2} m v^2 \quad 2.43$$

où  $v$  représente le vecteur vitesse du centre de gravité de la masse  $m$ .

Les coordonnées du centre de gravité de la masse  $m$  sont :

En abscisse :

$$x + l \sin \theta \quad 2.44$$

En ordonnée :

$$l \cos \theta \quad 2.45$$

On exprime à partir de ces coordonnées les expressions des vitesses, selon x :

$$v_x = \dot{x} + l\dot{\theta} \cos \theta \quad 2.46$$

selon y :

$$v_y = -l\dot{\theta} \sin \theta \quad 2.47$$

soit :

$$v^2 = v_x^2 + v_y^2 = \dot{x}^2 + l^2 \dot{\theta}^2 + 2xl\dot{\theta} \cos \theta \quad 2.48$$

L'énergie cinétique prend alors l'expression :

$$E_c = \frac{1}{2} M \dot{x}^2 + \frac{1}{2} m (\dot{x}^2 + l^2 \dot{\theta}^2 + 2xl\dot{\theta} \cos \theta) \quad 2.49$$

#### 2.4.1.1.2. Energie potentielle du système

L'énergie potentielle du centre de gravité de la masse  $m$ ,

$$E_p = mgl \cos \theta \quad 2.50$$

#### 2.4.1.1.3. Equation de Lagrange pour le degré de liberté $q(t) = \theta(t)$

Le Lagrangien  $L$  du système est la différence entre son énergie cinétique et son énergie potentielle.

$$L = \frac{1}{2} M \dot{x}^2 + \frac{1}{2} m (\dot{x}^2 + l^2 \dot{\theta}^2 + 2xl\dot{\theta} \cos \theta) - mgl \cos \theta \quad 2.51$$

La dérivée partielle selon  $\dot{\theta}$  de cette expression s'écrit,

$$\frac{\partial L}{\partial \dot{\theta}} = ml^2 \dot{\theta} + mlx \cos \theta \quad 2.52$$

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}} \right) = ml^2 \ddot{\theta} + ml\dot{x} \cos \theta - ml\dot{x} \dot{\theta} \sin \theta \quad 2.53$$

$$\frac{\partial L}{\partial \theta} = -m\dot{x}\dot{\theta} \sin \theta + mgl \sin \theta \quad 2.54$$

On obtient comme première équation de Lagrange :

$$ml^2\ddot{\theta} + ml(\ddot{x} \cos \theta - g \sin \theta) = 0 \quad 2.55$$

#### 2.4.1.1.4. Equation de Lagrange pour le degré de liberté $q(t) = x(t)$

La dérivée partielle du Lagrangien suivant  $x$  s'écrit,

$$\frac{\partial L}{\partial \dot{x}} = m\dot{x} + M\dot{x} + ml\dot{\theta} \cos \theta \quad 2.56$$

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{x}} \right) = (M + m)\ddot{x} + ml\ddot{\theta} \cos \theta - ml\dot{\theta}^2 \sin \theta \quad 2.57$$

$$\frac{\partial L}{\partial x} = 0 \quad 2.58$$

Soit, comme seconde équation de Lagrange :

$$(M + m)\ddot{x} + ml(\ddot{\theta} \cos \theta - \dot{\theta}^2 \sin \theta) = U(t) \quad 2.59$$

#### 2.4.1.1.5. Linéarisation du modèle autour du point de fonctionnement

Si on se limite aux petites variations de  $\theta$  autour du point de fonctionnement  $\theta_0 = 0$ , correspondant à la position verticale de la barre, on peut écrire :

$$\begin{cases} \cos \theta \approx 1 \\ \sin \theta \approx \theta \end{cases} \quad 2.60$$

On peut également simplifier l'expression,

$$\ddot{\theta} \cos \theta - \dot{\theta}^2 \sin \theta \quad 2.61$$

en l'écrivant ainsi ;

$$\frac{d}{d\theta} (\dot{\theta} \cos \theta) \approx \ddot{\theta} \quad 2.62$$

on obtient le système ainsi linéarisé :

$$\begin{cases} (M+m)\ddot{x} + ml\ddot{\theta} = U(t) \\ ml^2\ddot{\theta} + ml(\ddot{x} - g\theta) = 0 \end{cases} \quad 2.63$$

### 2.4.1.2. Modélisation du système à 2 degré de liberté

On obtient, à partir de la première équation du système précédent :

$$\ddot{\theta}(t) = \frac{1}{ml} [U(t) - (M+m)\ddot{x}] \quad 2.64$$

Après report dans la seconde équation du système, on obtient :

$$\ddot{x}(t) = \frac{1}{M} U(t) - \frac{mg}{M} \theta(t) \quad 2.65$$

et :

$$\ddot{\theta}(t) = \frac{-1}{Ml} U(t) + \frac{(M+m)g}{Ml} \theta(t) \quad 2.66$$

Le système se résume à 2 équations différentielles du second degré, le vecteur d'état du système est choisi comme suit :

$$X = [x \quad \dot{x} \quad \theta \quad \dot{\theta}] \quad 2.67$$

La modélisation d'état est alors représentée par les équations suivantes :

$$\begin{cases} \dot{X} = AX + Bu \\ Y = CX \end{cases} \quad 2.68$$

$$\begin{bmatrix} \dot{x} \\ \ddot{x} \\ \dot{\theta} \\ \ddot{\theta} \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{mg}{M} & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & \frac{(M+m)g}{Ml} & 0 \end{bmatrix} \times \begin{bmatrix} x \\ \dot{x} \\ \theta \\ \dot{\theta} \end{bmatrix} + \begin{bmatrix} 0 \\ \frac{1}{M} \\ 0 \\ -\frac{1}{Ml} \end{bmatrix} \times u(t) \quad 2.69$$

### Données numériques

$$M = 2\text{Kg} ; m = 0.1\text{Kg} ; l = 0.5\text{m} ; g \approx 10\text{m/s}^2$$

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & -0.5 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 21 & 0 \end{bmatrix} \quad B = \begin{bmatrix} 0 \\ 0.5 \\ 0 \\ -1 \end{bmatrix} \quad 2.70$$

Dans un premier temps, nous nous efforçons d'asservir la barre en position verticale indépendamment de la position en  $x$ .

### 2.4.1.3. Test du modèle discrétisé

Pour tester le système, on va appliquer une commande positive discrète (une impulsion de force  $u(t)$  est choisie d'amplitude  $0.1N$  et de durée  $0.1s$ ), donc il nous faut par conséquent discrétiser le modèle donné par l'équation (2.69) avec les matrices  $A$  et  $B$  données par l'équation (2.70) et la matrice  $C$  égale à :

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad 2.71$$

Une période d'échantillonnage de  $0.01s$  est utilisée.

Les sorties  $x$ ,  $\dot{x}$ ,  $\theta$  et  $\dot{\theta}$  sont représentées par les figures suivantes :

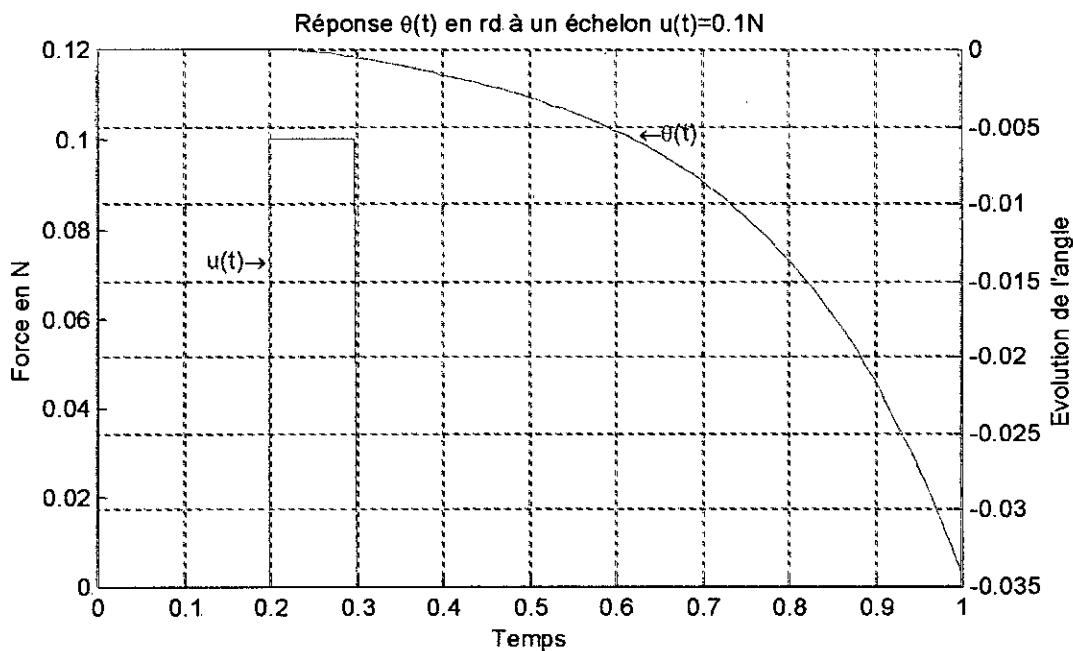


Figure 2.3. Réponse  $\theta(t)$  en rd à un échelon  $u(t) = 0.1N$

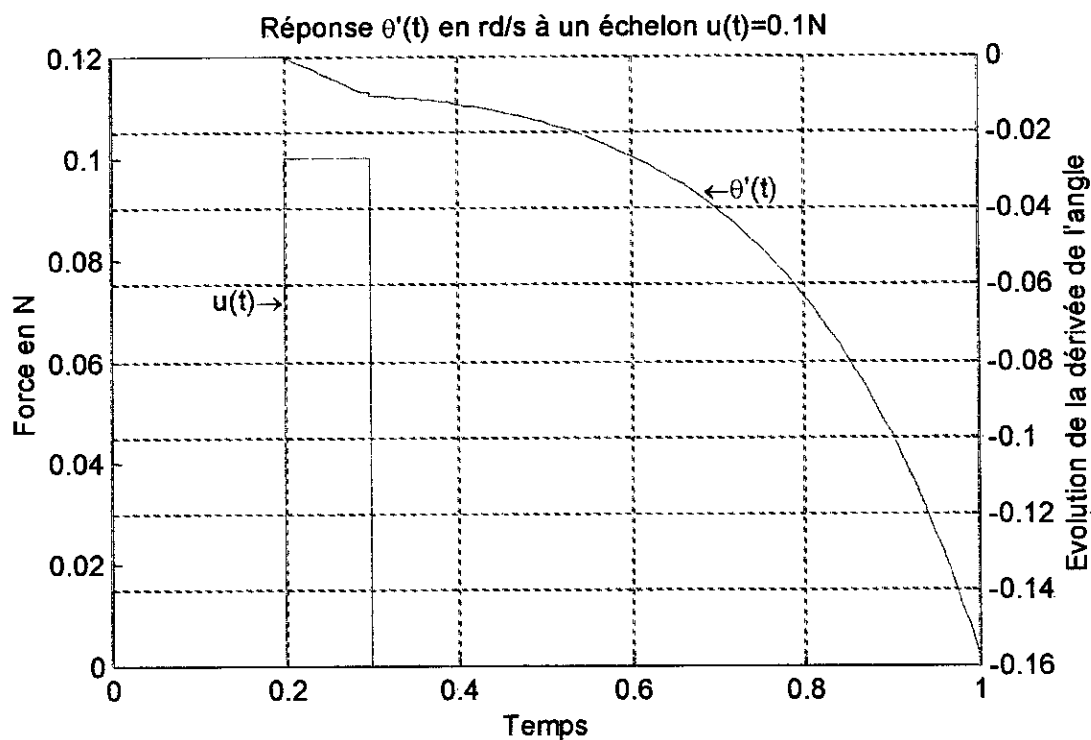


Figure 2.4. Réponse  $\theta'(t)$  en rd/s à un échelon  $u(t) = 0.1N$

Sous l'action d'une force positive, la barre bascule avec un angle  $\theta(t)$  négatif et une vitesse de module croissant.

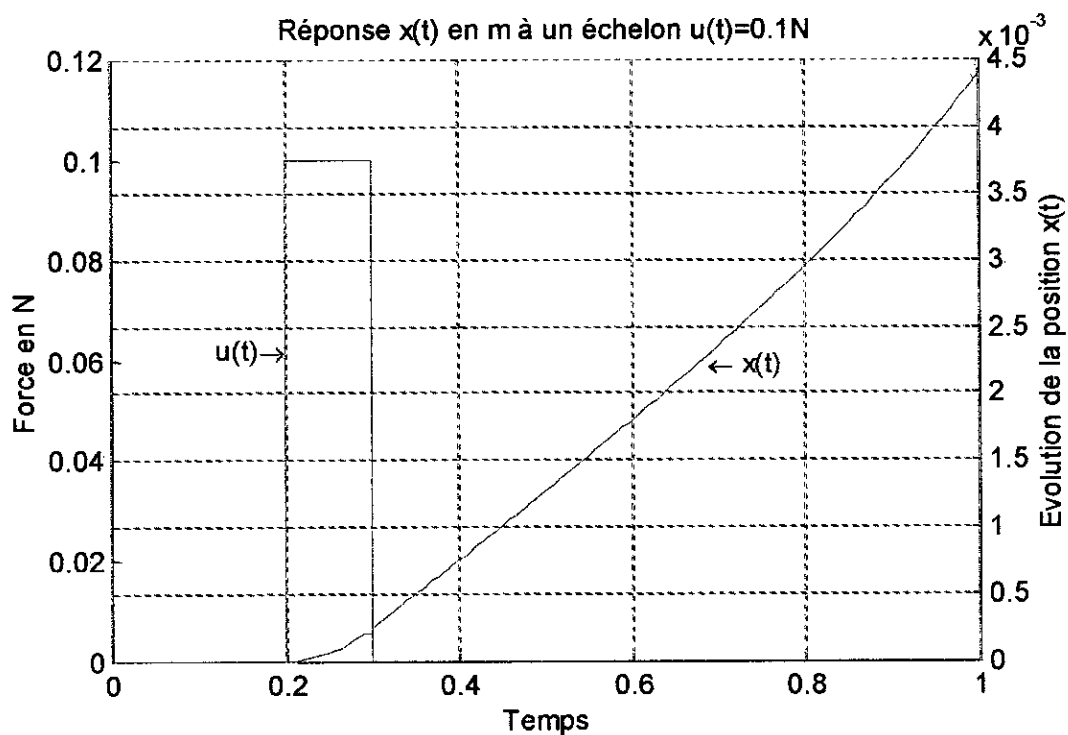
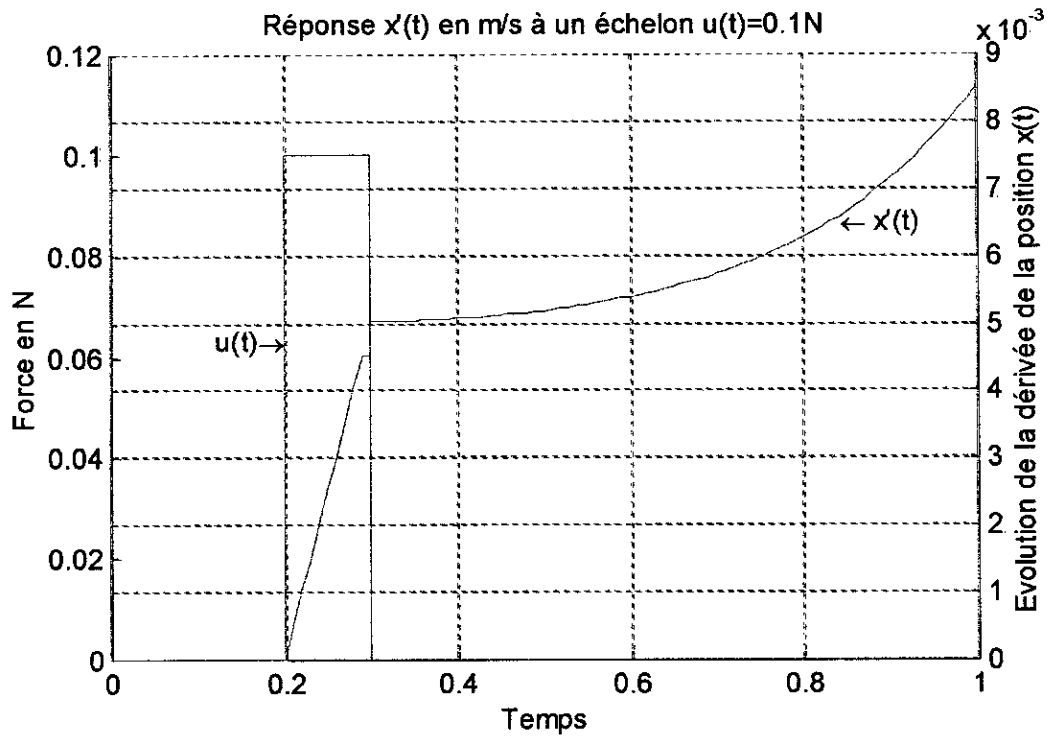


Figure 2.5. Réponse  $x(t)$  en m à un échelon  $u(t) = 0.1N$





**Figure 2.6.** Réponse  $x'(t)$  en  $m/s$  à un échelon  $u(t) = 0.1N$

La position évolue paraboliquement lors de l'action d'une force. A l'annulation de celle-ci le chariot continue sa progression car le système est supposé sans frottements, mais également sous l'action du basculement de la barre.

On peut également vérifier la réponse du processus à un vecteur de conditions initiales données.

Soit  $X_0 = \begin{bmatrix} 0 \\ 0 \\ 0.1 \\ 0 \end{bmatrix}$  ce qui correspond à une inclinaison initiale de la barre de  $0.1rd$ , le chariot

se trouvant en  $x = 0$  et les vitesses du chariot et de la barre étant nulles.

Les figures suivantes représentent l'évolution des sorties après avoir imposé le vecteur des conditions initiales.

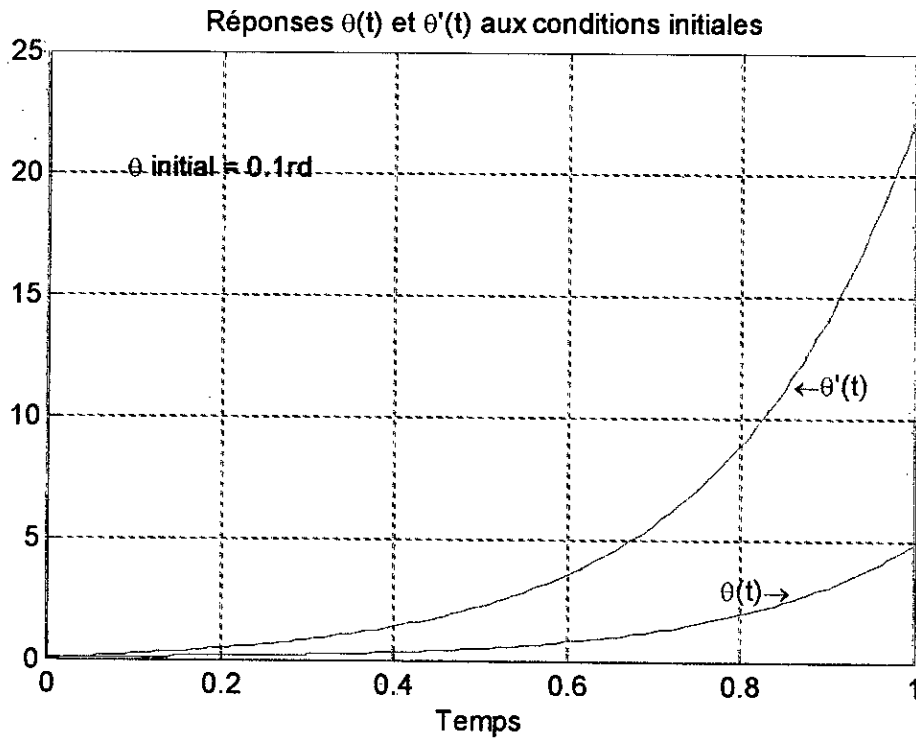


Figure 2.7. Réponses  $\theta(t)$  et  $\theta'(t)$  aux conditions initiales

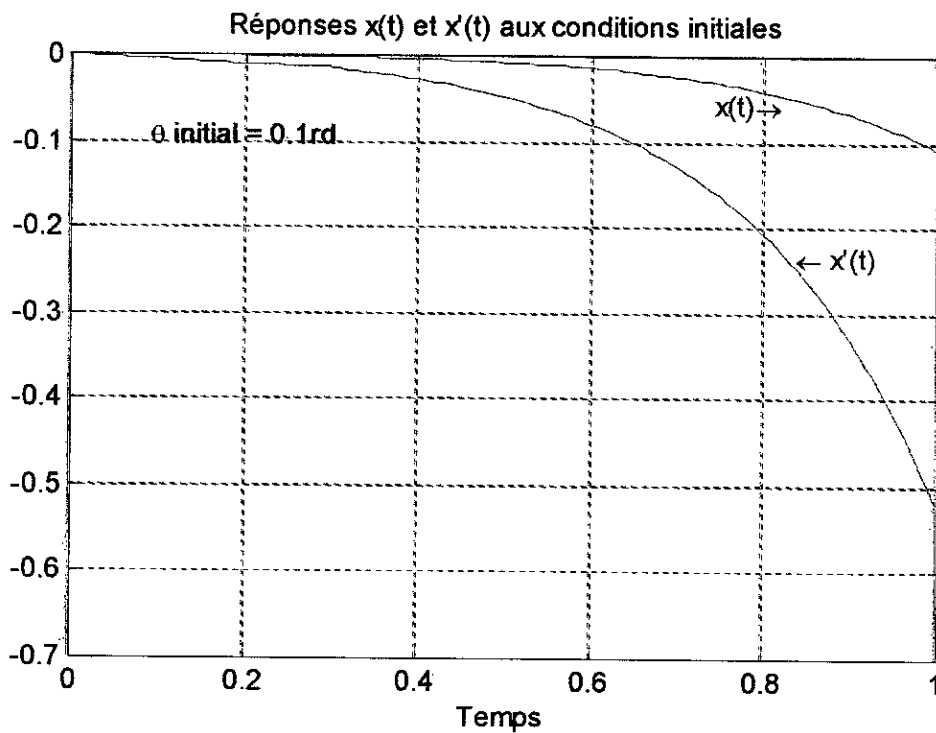


Figure 2.8. Réponses  $x(t)$  et  $x'(t)$  aux conditions initiales

La force  $u(t)$  est nulle, la position initiale  $\theta = 0.1rd$  entraîne la barre vers le bas et fait partir le chariot vers la gauche.

#### 2.4.1.4. Asservissement de la variable angulaire $\theta(t)$

On cherche à asservir la position angulaire  $\theta(t)$  à zéro de façon à maintenir la barre verticale indépendamment de la position du chariot en  $x$ .

Pour ceci, un contrôle prédictif généralisé est utilisé.

On utilise les mêmes matrices  $A$  et  $B$  données par l'équation (2.70) avec la matrice  $C$  égale à :

$$C = [0 \quad 0 \quad 1 \quad 0] \quad 2.72$$

Il s'agit d'une commande discrète, il nous faut par conséquent discrétiser le modèle avec une période d'échantillonnage égale à 0.1s et déterminer la fonction de transfert donnée par la forme suivante :

$$H(q^{-1}) = \frac{B(q^{-1})}{A(q^{-1})} \quad 2.73$$

La représentation linéaire est donnée par l'expression suivante :

$$A(q^{-1})y(t) = B(q^{-1})u(t-1) + C(q^{-1})e(t)/\Delta \quad 2.74$$

où

$$A(q^{-1}) = 1 + a_1q^{-1} + a_2q^{-2} + a_3q^{-3} + a_4q^{-4} \quad 2.75$$

$$B(q^{-1}) = b_0 + b_1q^{-1} + b_2q^{-2} + b_3q^{-3} \quad 2.76$$

D'un autre côté on a pris  $C(q^{-1}) = 1$ , l'horizon de prédiction maximal  $N_2 = 4$ , l'horizon de commande  $N_u = 1$  et l'horizon initial  $N_1 = 1$ . Le calcul des prédicteurs nous permet de déduire la matrice  $G_1$  donnée par l'équation (2.26) et le vecteur  $F$  et aussi calculer les incréments de l'action de contrôle  $\tilde{U}$  donnée par (2.28). Nous ne prenons que le premier élément  $\Delta u(t)$ .

Dans un premier temps la consigne (une consigne angulaire  $\theta_c$ ) est nulle ; on vérifie la réponse du processus asservi aux conditions initiales :

- Inclinaison de la barre de 0.1rd.
- Vitesse de la barre de 0.2rd/s.

$$X_0 = \begin{bmatrix} 0 \\ 0 \\ 0.1 \\ 0.2 \end{bmatrix}$$

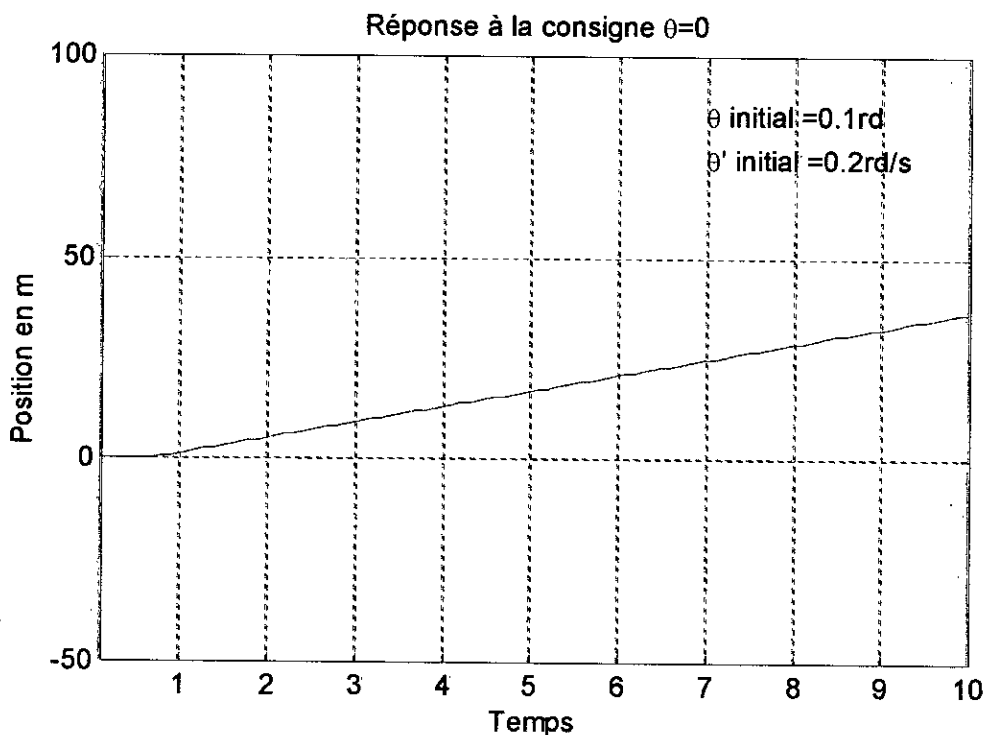


Figure 2.9. Réponse  $x(t)$  à la consigne  $\theta = 0$

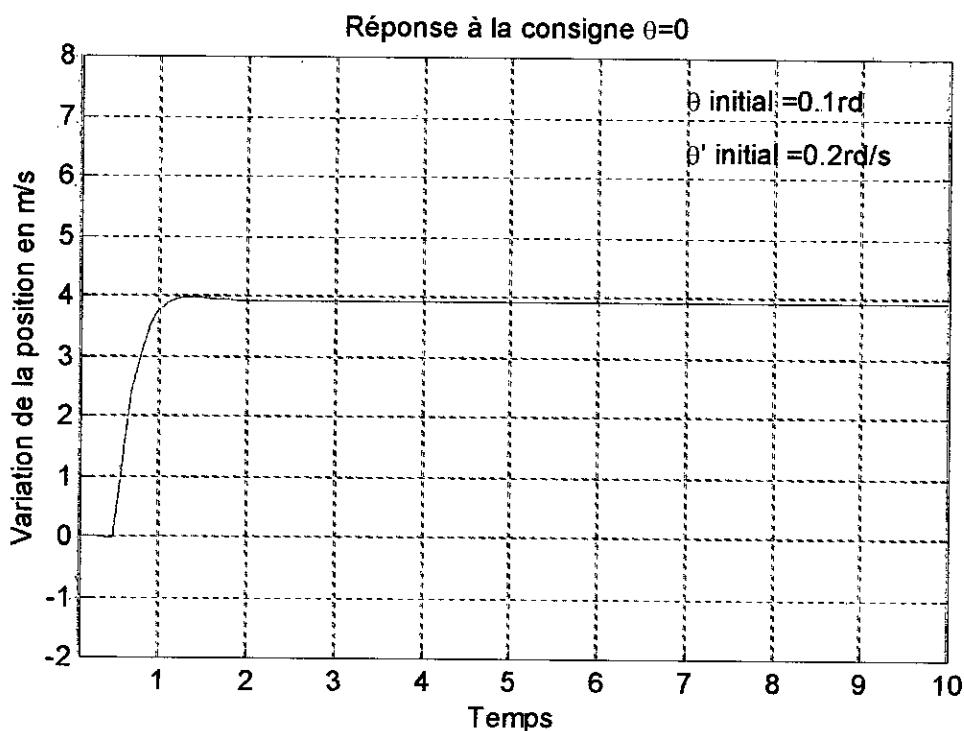
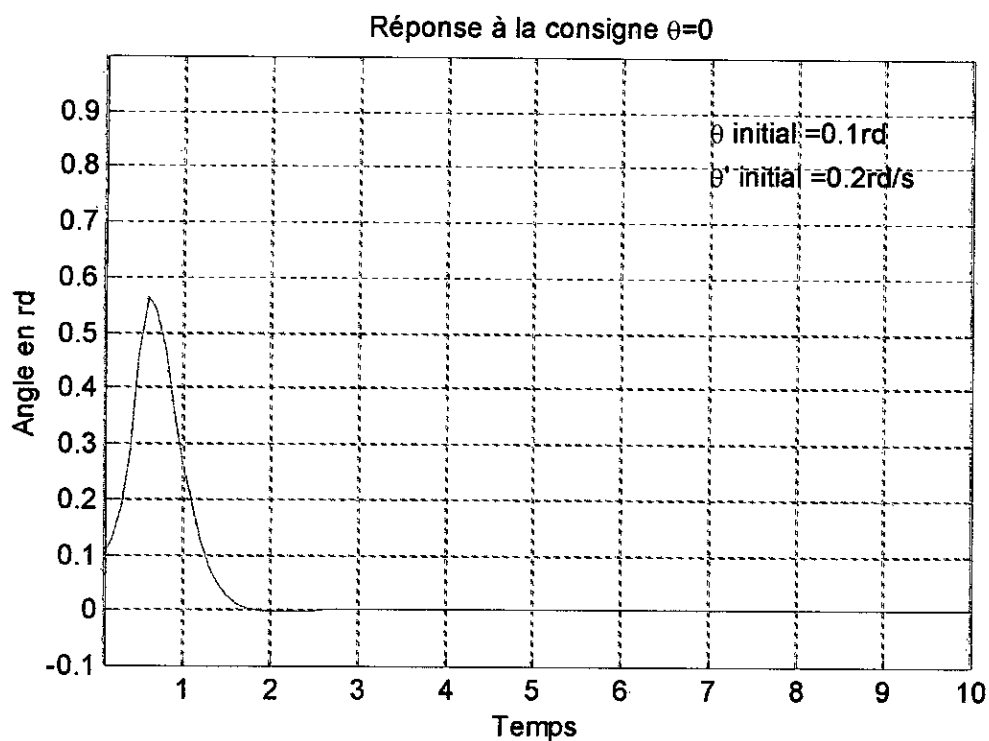
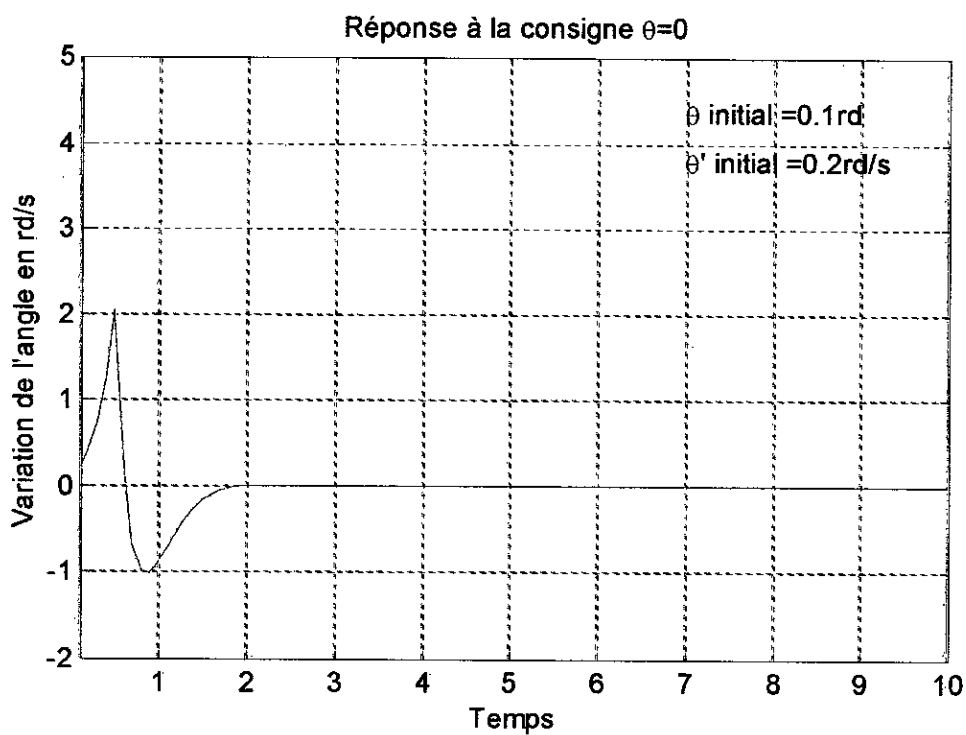
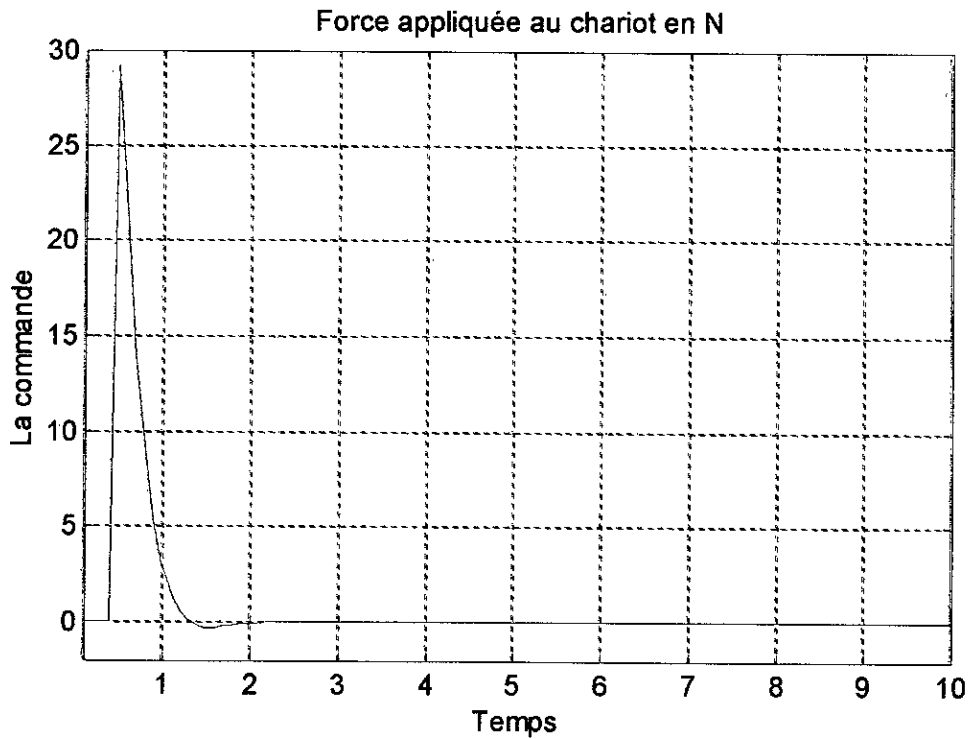


Figure 2.10. Réponse  $x'(t)$  à la consigne  $\theta = 0$

Figure 2.11. Réponse  $\theta(t)$  à la consigne  $\theta = 0$ Figure 2.12. Réponse  $\theta'(t)$  à la consigne  $\theta = 0$



**Figure 2.13. Force appliquée au chariot en  $N$**

La barre initialement inclinée de  $0.1rd$  avec une vitesse de  $0.2rd/s$ , est ramenée à la verticale : pour cela le chariot a dû se déplacer sur la droite. On constate qu'une fois la barre est en position verticale, le chariot continue à évoluer même en l'absence de force  $u(t)$ , ceci étant toujours consécutif au fait que le système est considéré sans pertes. On vérifie à présent la réponse du processus en poursuite en imposant un signal carré d'amplitude  $0.05rd$  comme consigne.

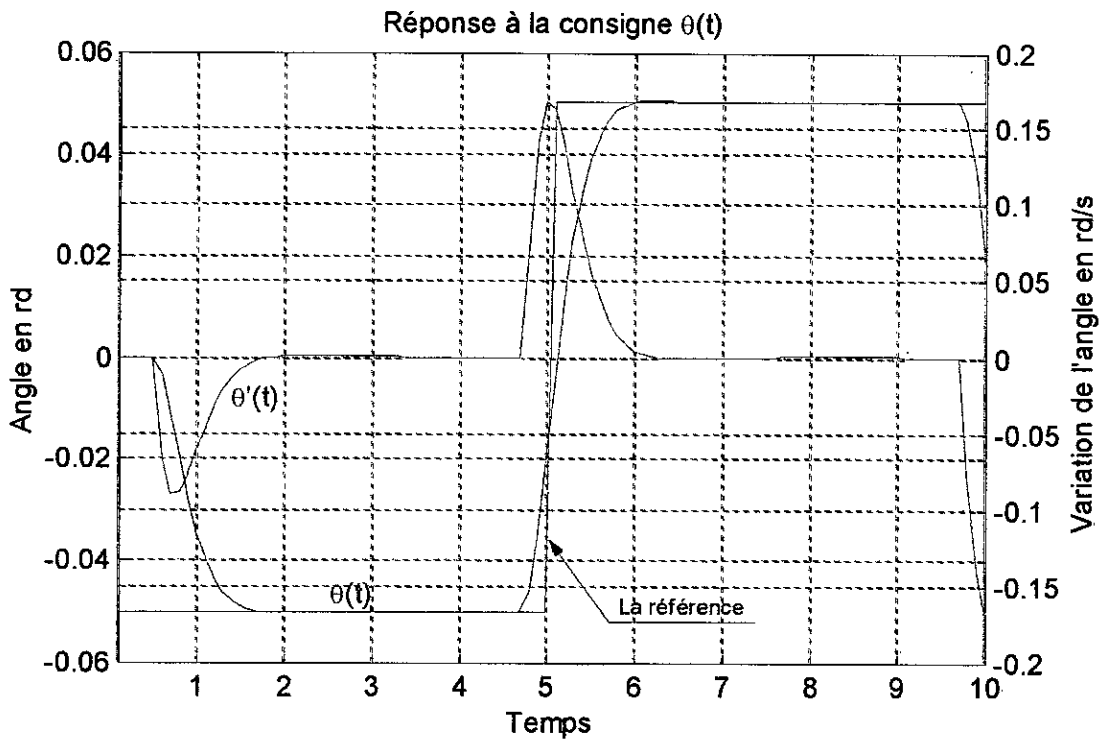


Figure 2.14.

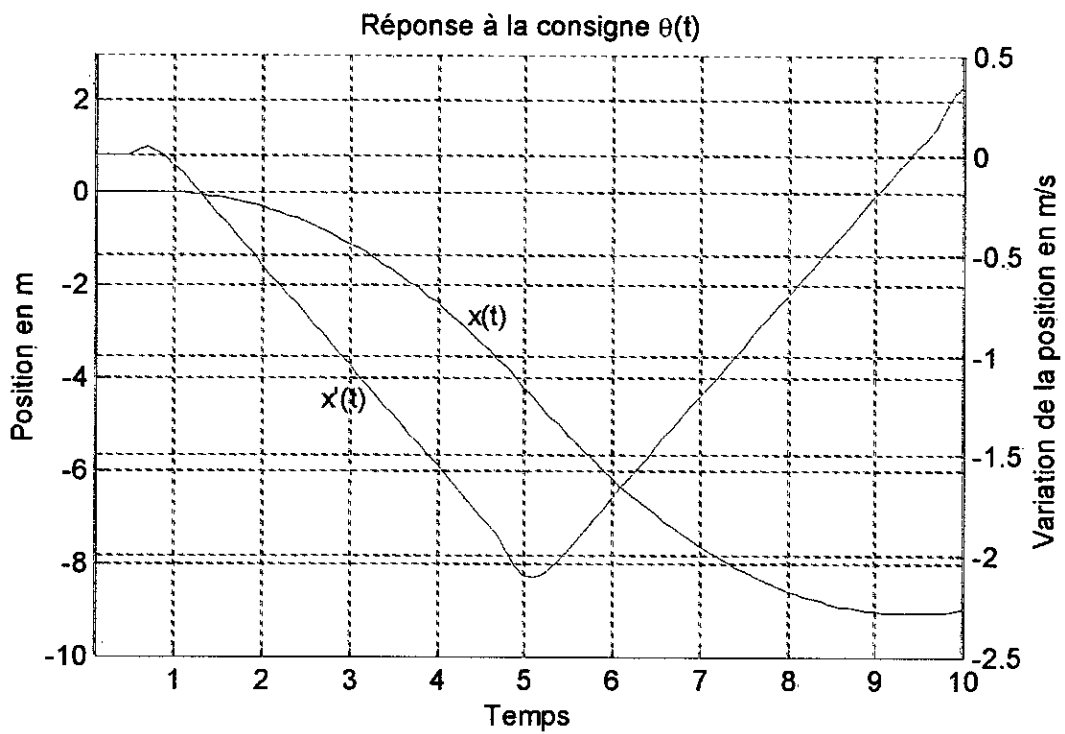
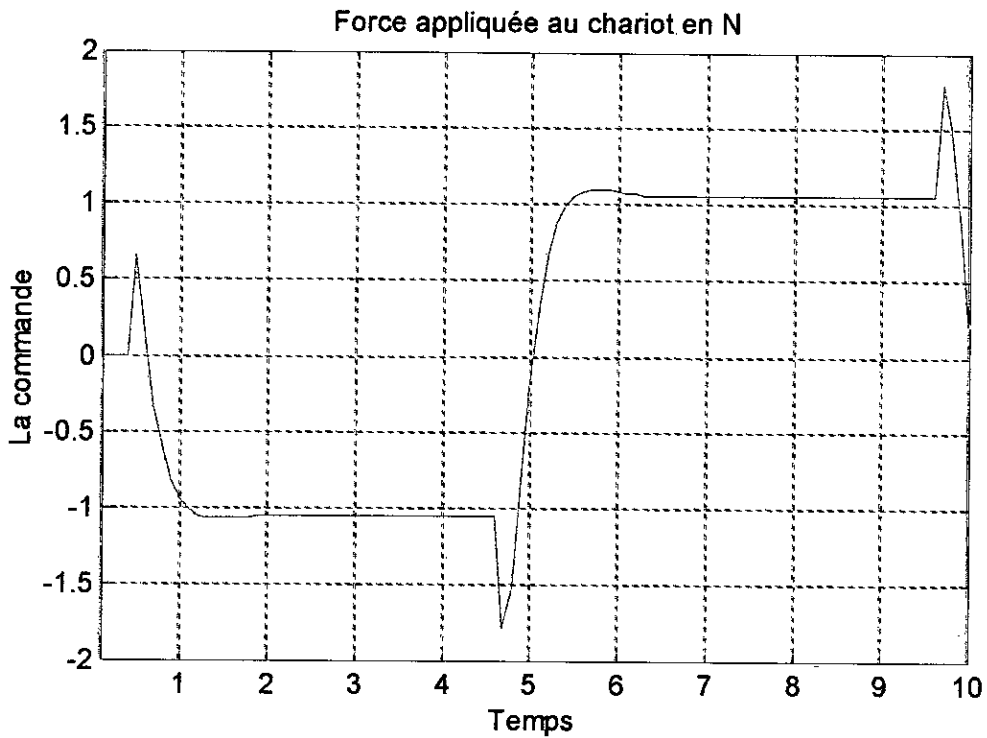


Figure 2.15.

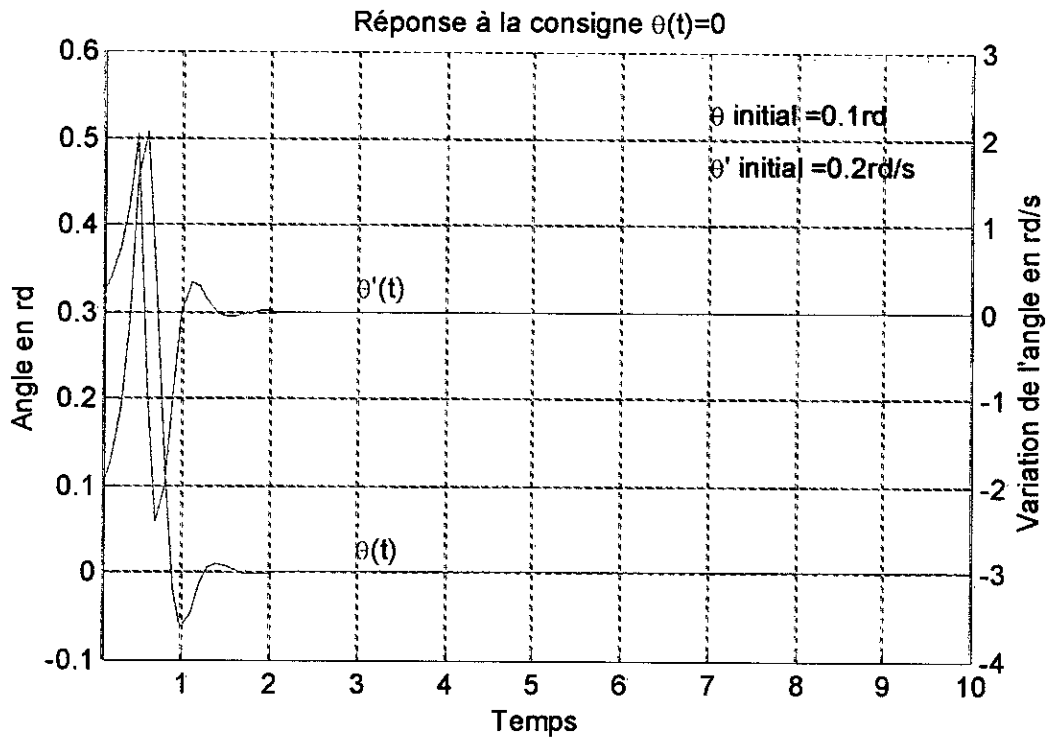
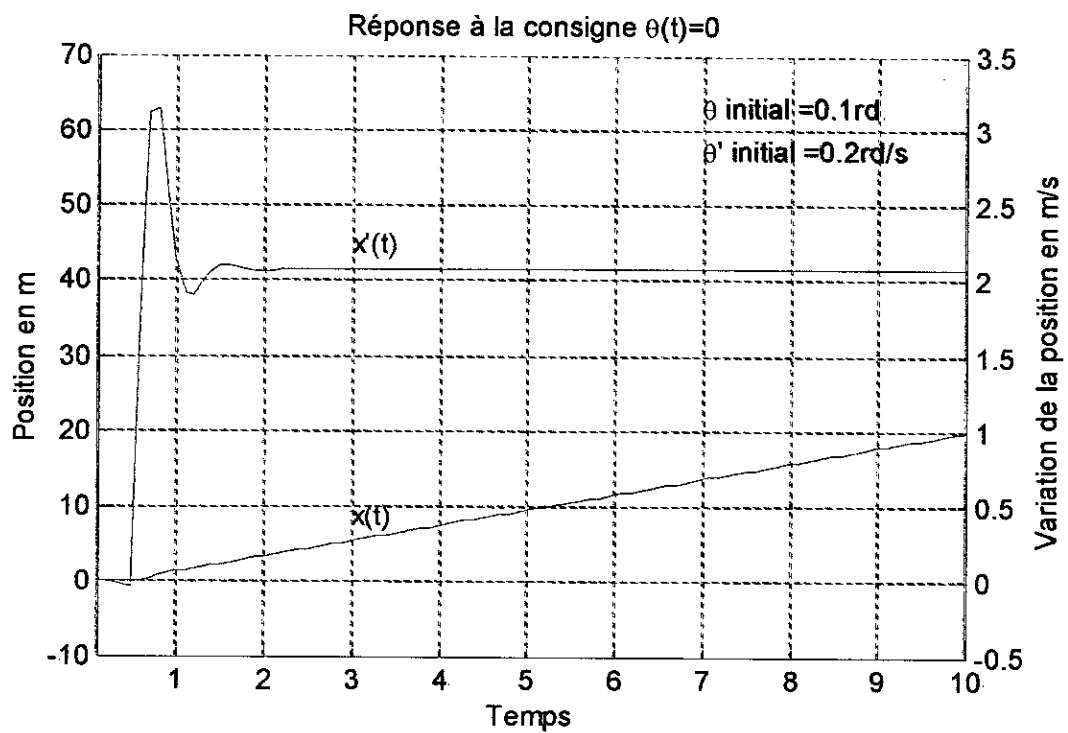


**Figure 2.16. Force appliquée au chariot en  $N$**

Le système suit la consigne avec une erreur statique non nulle.

Puisque le système discret est instable en boucle ouverte, une valeur plus grande de l'horizon de commande est souhaitable. On prend  $N_u = N_2 = 4$  et le coefficient de pondération du signal de commande  $\lambda = 0.0001$ , les résultats obtenus pour une consigne nulle sont montrés dans les figures suivantes.



Figure 2.17. Réponses  $\theta(t)$  et  $\theta'(t)$  à la consigne  $\theta = 0$ Figure 2.18. Réponses  $x(t)$  et  $x'(t)$  à la consigne  $\theta = 0$

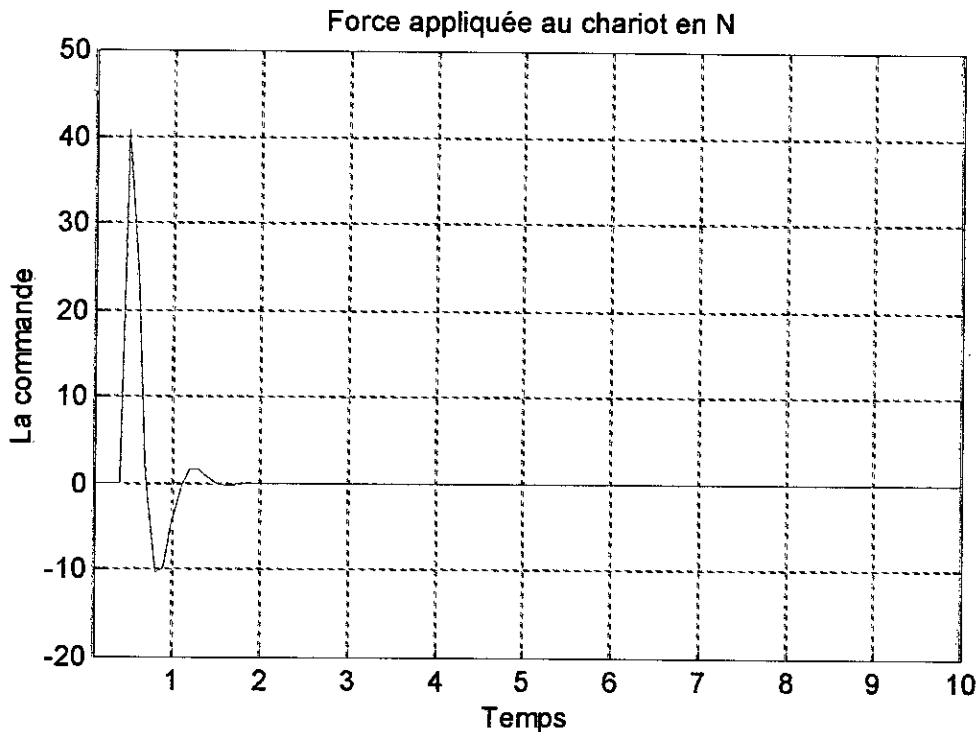


Figure 2.19. Force appliquée au chariot en  $N$

On remarque que la barre initialement inclinée de  $0.1rd$  avec une vitesse de  $0.2rd/s$  et  $N_u = N_2 = 4$ , est ramenée à la verticale en 1.5 secondes environ.

#### 2.4.1.5. Asservissement de la position $x(t)$ et de l'angle $\theta(t)$

Il s'agit à présent d'asservir la position du chariot en  $x$  tout en veillant à maintenir la barre suspendue en position verticale.

Au début, avant d'utiliser l'algorithme *GPC*, on va faire la régulation du système autour d'une consigne nulle avec la commande optimale.

La commande optimale dans l'espace d'état est basée sur la minimisation d'un critère quadratique, le plus général étant :

$$J = \frac{1}{2} X(N)^T H X(N) + \frac{1}{2} \sum_{t=0}^{N-1} [X(t)^T Q X(t) + u(t)^T R u(t)] \quad 2.77$$

$N$  désigne l'instant auquel on atteint l'objectif qui est l'annulation de l'état dans le cas d'une régulation autour d'une consigne nulle.

$Q$  et  $H$  sont des matrices de pondération, symétriques. La matrice  $H$  ou terme de pondération terminal permet de pénaliser plus ou moins l'écart final par rapport à la cible cherchée.

$Q$  est une matrice de pondération de l'état intermédiaire,  $R$  est une matrice de pondération des signaux de commande.

Le problème consiste à minimiser le critère  $J$  de l'équation (2.77) sous la contrainte suivante :

$$X(t+1) = AX(t) + Bu(t) \quad 2.78$$

La minimisation du critère quadratique entre les étapes  $(N-1)$  et  $N$  sous la contrainte précédente permet d'obtenir :

$$\begin{aligned} u(N-1) &= -[R + B^T P(N)B]^{-1} B^T P(N)AX(N-1) \\ &= -F(N-1)X(N-1) \\ P(N-1) &= A^T \left[ P(N) - P(N)B[R + B^T P(N)B]^{-1} B^T P(N) \right] A + Q \end{aligned} \quad 2.79$$

(équation de *Riccati*)

On utilise les mêmes matrices  $A$  et  $B$  données par l'équation (2.70) avec la matrice  $C$  égale à :

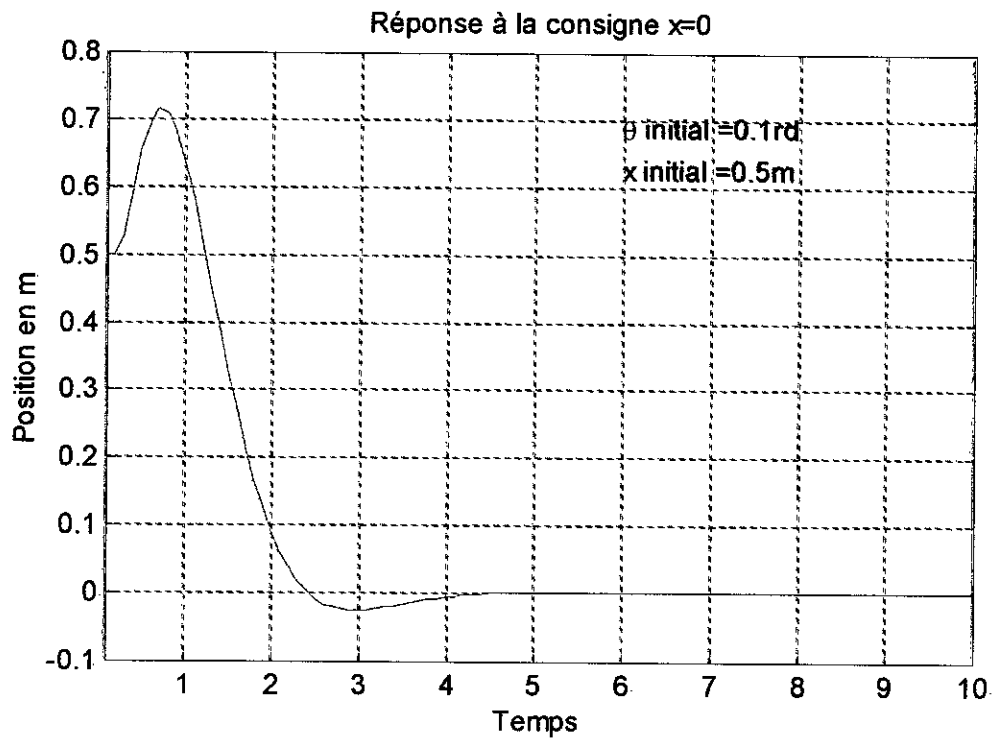
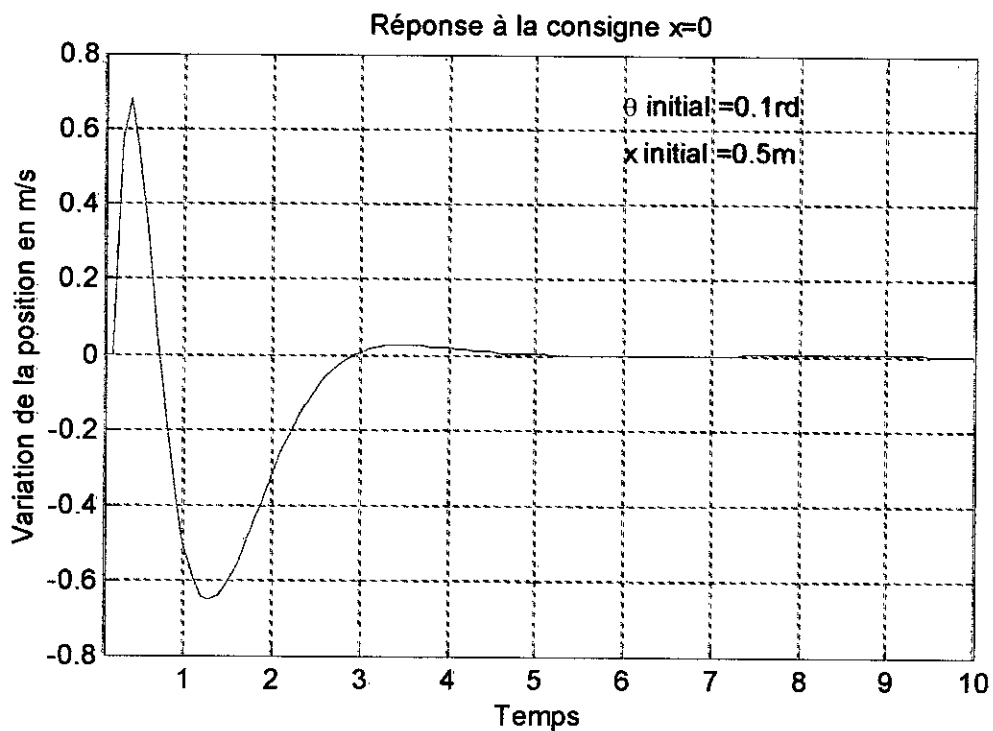
$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad 2.80$$

Les matrices de pondération sont données par l'équation suivante :

$$Q = H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \quad R = 0.01 \quad 2.81$$

Dans un premier temps la consigne (une consigne de position  $x_c$ ) est nulle et l'on vérifie la réponse du processus à des conditions initiales données par le vecteur suivant :

$$X_0 = \begin{bmatrix} 0.5 \\ 0 \\ 0.1 \\ 0 \end{bmatrix}$$

Figure 2.20. Réponse  $x(t)$  à la consigne  $x = 0$ Figure 2.21. Réponse  $x'(t)$  à la consigne  $x = 0$

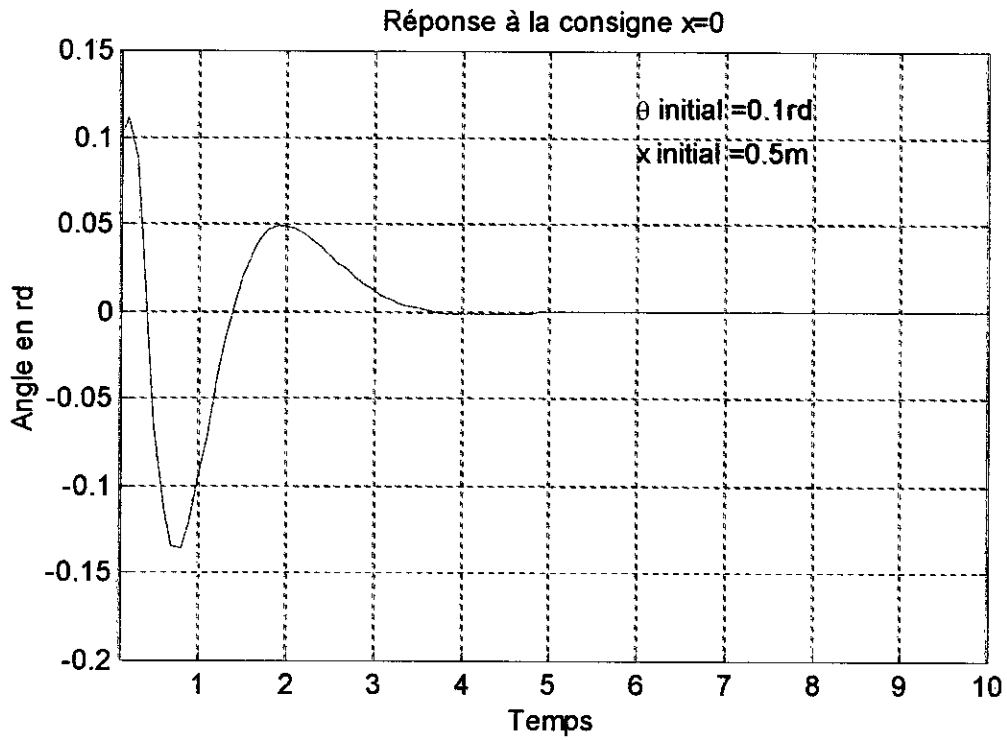


Figure 2.22. Réponse  $\theta(t)$  à la consigne  $x = 0$

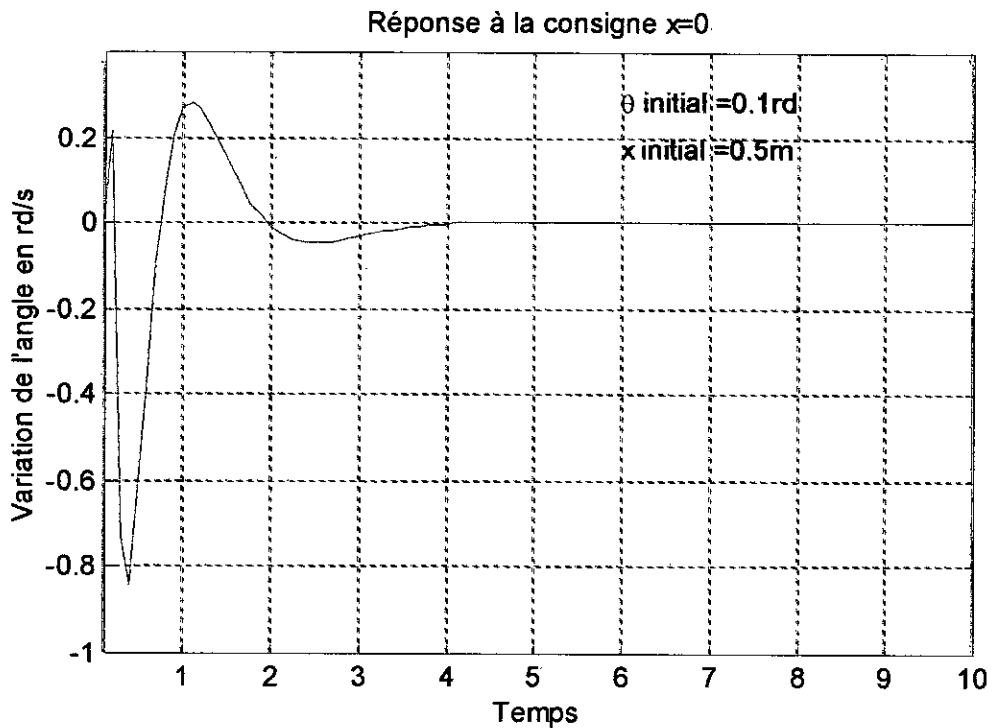
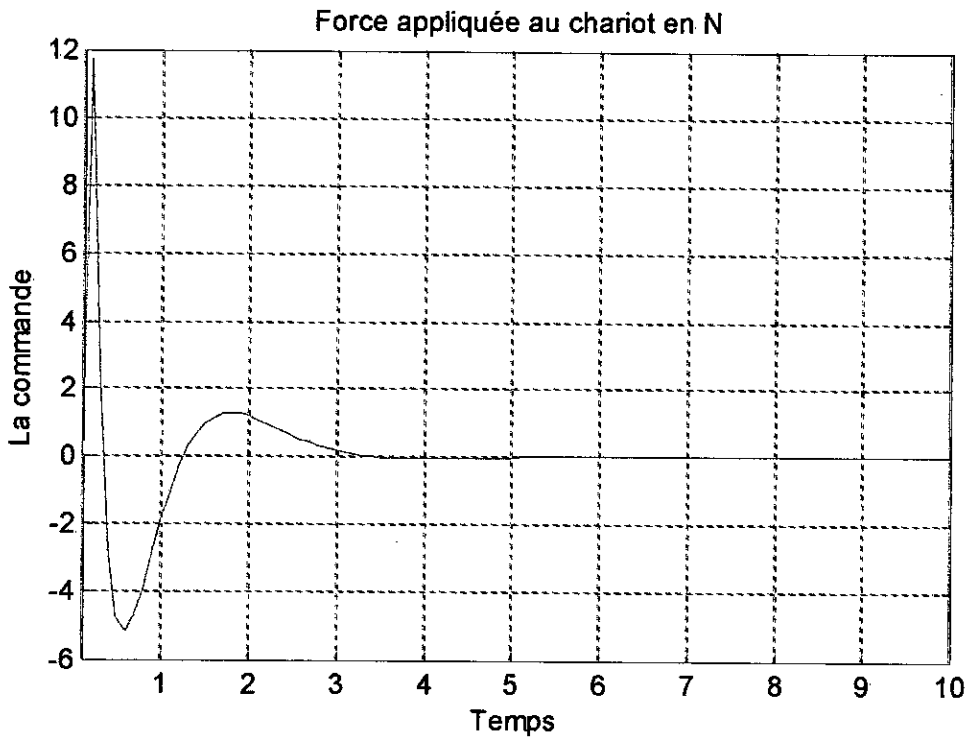


Figure 2.23. Réponse  $\theta'(t)$  à la consigne  $x = 0$



**Figure 2.24. Force appliquée au chariot en  $N$**

Toujours pour le même objectif, c'est à dire l'asservissement de la position du chariot  $x$  tout en veillant à maintenir la barre suspendue en position verticale, mais cette fois-ci en utilisant le contrôle prédictif généralisé (*GPC*) avec les mêmes matrices  $A$  et  $B$  données par l'équation (2.70) et la matrice  $C$  donnée par l'équation (2.80). Donc, il s'agit d'asservir deux sorties ( $x$  et  $\theta$ ) en même temps.

Après discrétisation du modèle avec une période d'échantillonnage égale à  $0.1s$ , on détermine pour chaque sortie sa propre représentation linéaire.

Pour la position  $x$  (sortie  $y_1$ ), on obtient :

$$A(q^{-1})y_1(t) = B1(q^{-1})u(t-1) + C(q^{-1})e(t) / \Delta \quad 2.82$$

et pour l'angle  $\theta$  (sortie  $y_2$ ), on obtient :

$$A(q^{-1})y_2(t) = B2(q^{-1})u(t-1) + C(q^{-1})e(t) / \Delta \quad 2.83$$

donc, on a deux fonctions de transferts.

$$H1(q^{-1}) = \frac{B1(q^{-1})}{A(q^{-1})} \quad 2.84$$

$$H2(q^{-1}) = \frac{B2(q^{-1})}{A(q^{-1})} \quad 2.85$$

où

$$A(q^{-1}) = 1 + a_1q^{-1} + a_2q^{-2} + a_3q^{-3} + a_4q^{-4} \quad 2.86$$

$$B1(q^{-1}) = b_{10} + b_{11}q^{-1} + b_{12}q^{-2} + b_{13}q^{-3} \quad 2.87$$

$$B2(q^{-1}) = b_{20} + b_{21}q^{-1} + b_{22}q^{-2} + b_{23}q^{-3} \quad 2.88$$

D'un autre coté, on a pris  $C(q^{-1}) = 1$ , l'horizon initial  $N_1 = 4$  (4 est le nombre des états du système), l'horizon de prédiction maximal  $N_2 = 7$ , l'horizon de commande  $N_u = 4$  et le coefficient de pondération du signal de commande  $\lambda = 0.0001$ . Pour chaque représentation, le calcul des prédicteurs nous permet de déduire la matrice  $G_1$  et le vecteur  $F$ , et aussi calculer les incréments de l'action de contrôle  $\tilde{U}$ .

Pour la première représentation, on calcule la matrice  $G_{11}$  et le vecteur  $F_1$ . Et pour la deuxième représentation, on calcule la matrice  $G_{12}$  et le vecteur  $F_2$ .

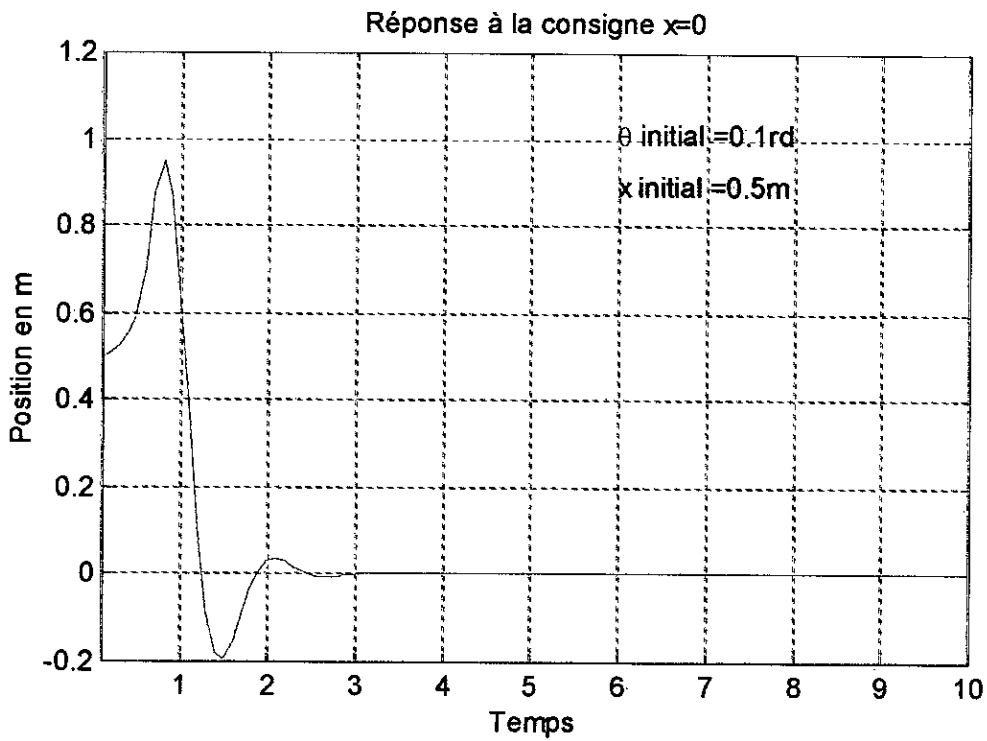
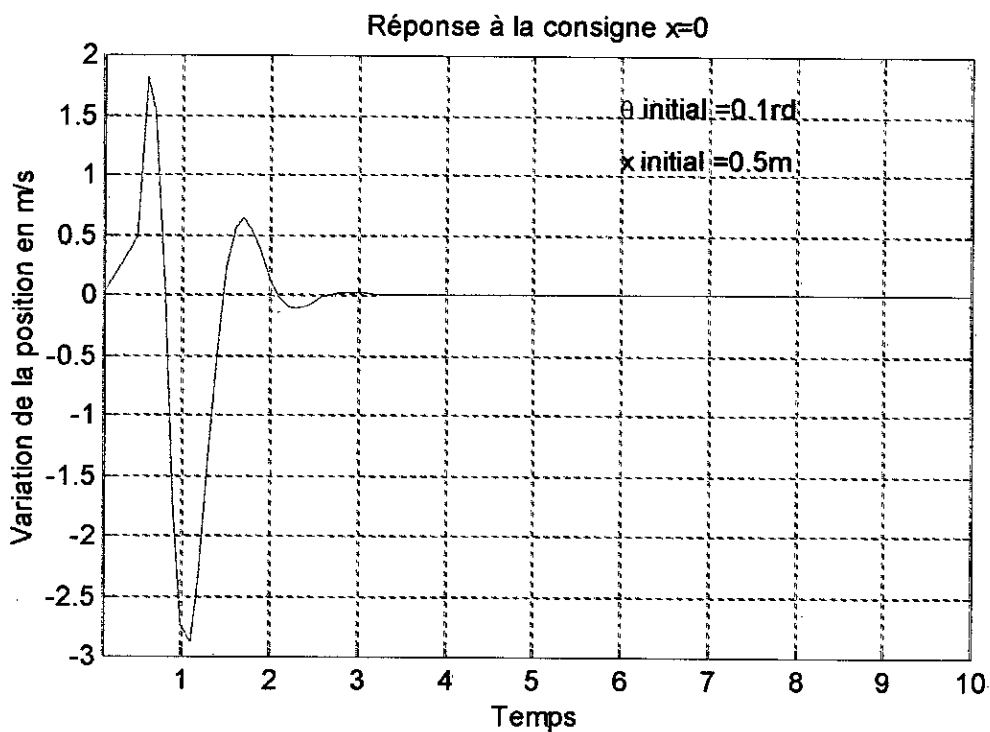
Le prédicteur a la forme :

$$\begin{aligned} \hat{Y} &= G_1 \tilde{U} + F \\ \Rightarrow \hat{Y} &= \begin{bmatrix} \hat{Y}_1 \\ \hat{Y}_2 \end{bmatrix} = \begin{bmatrix} G_{11} \\ G_{12} \end{bmatrix} \tilde{U} + \begin{bmatrix} F_1 \\ F_2 \end{bmatrix} \end{aligned} \quad 2.89$$

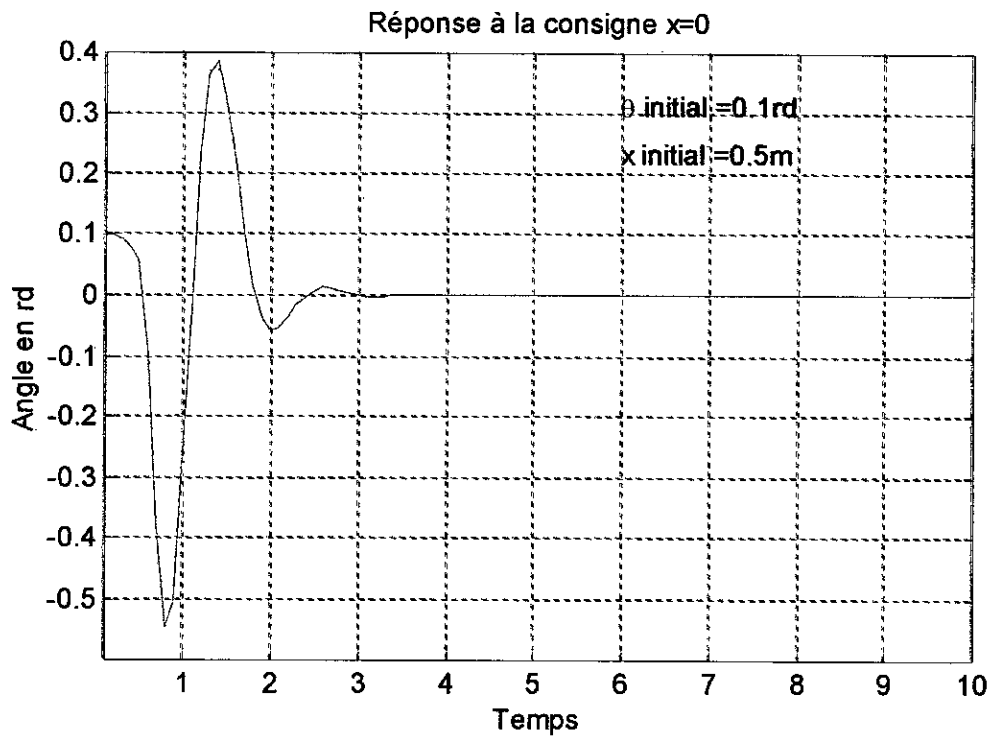
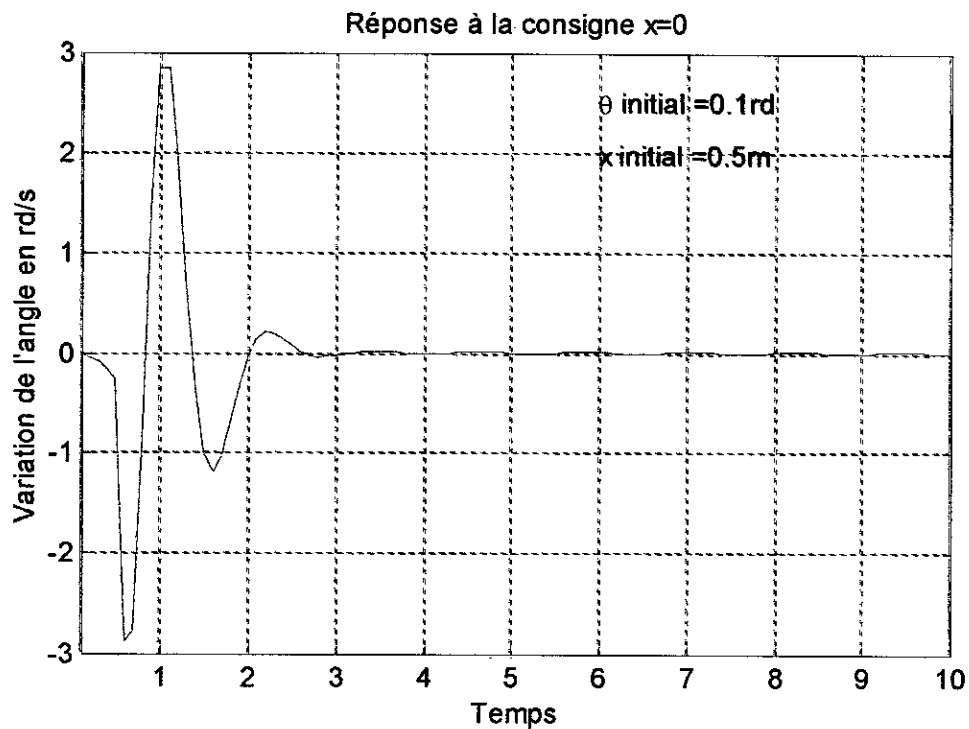
où  $\hat{Y}_1$  est le vecteur des sorties prédites de la sortie  $y_1$  et  $\hat{Y}_2$  est le vecteur des sorties prédites de la sortie  $y_2$ .

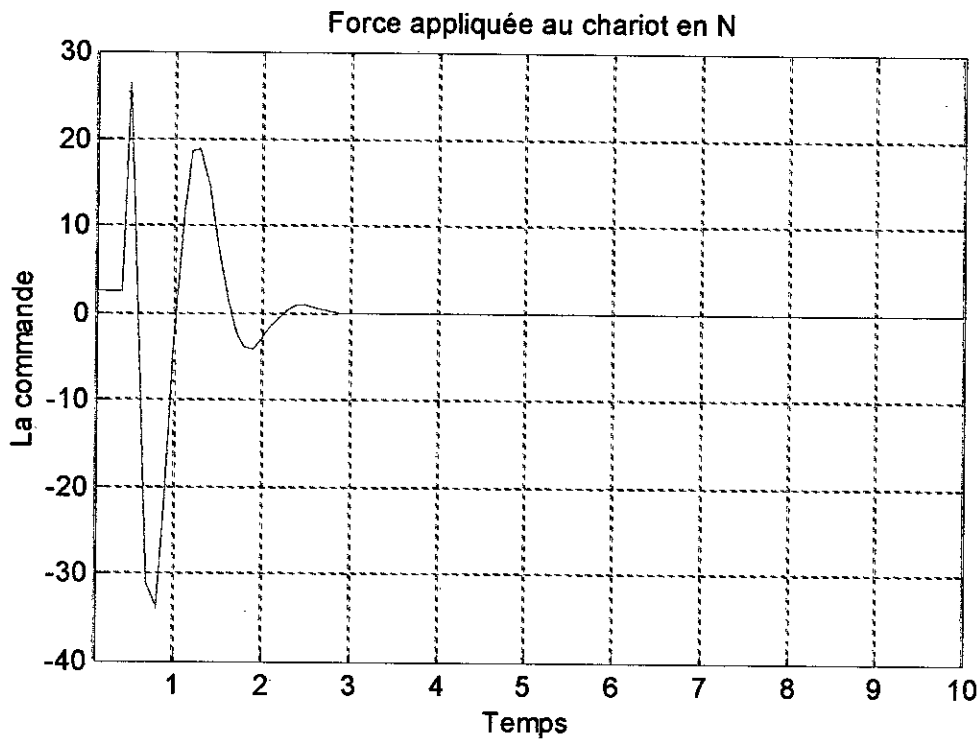
Les résultats obtenus à des conditions initiales données par le vecteur  $X_0 = \begin{bmatrix} 0.5 \\ 0 \\ 0.1 \\ 0 \end{bmatrix}$ ,

sont représentés par les figures suivantes :

Figure 2.25. Réponse  $x(t)$  à la consigne  $x = 0$ Figure 2.26. Réponse  $x'(t)$  à la consigne  $x = 0$



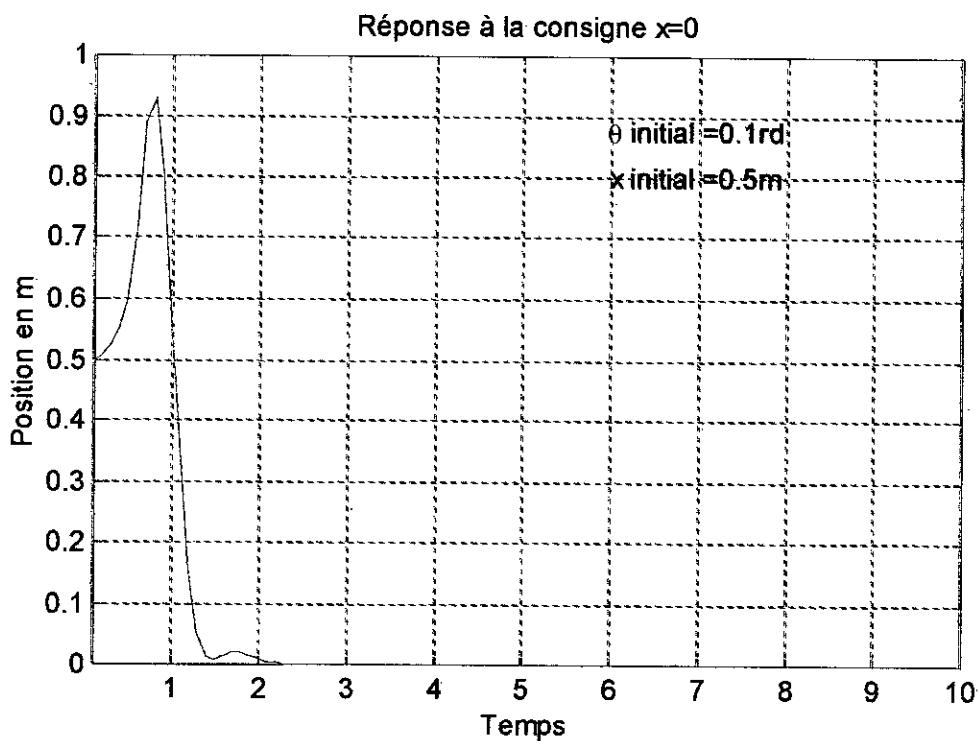
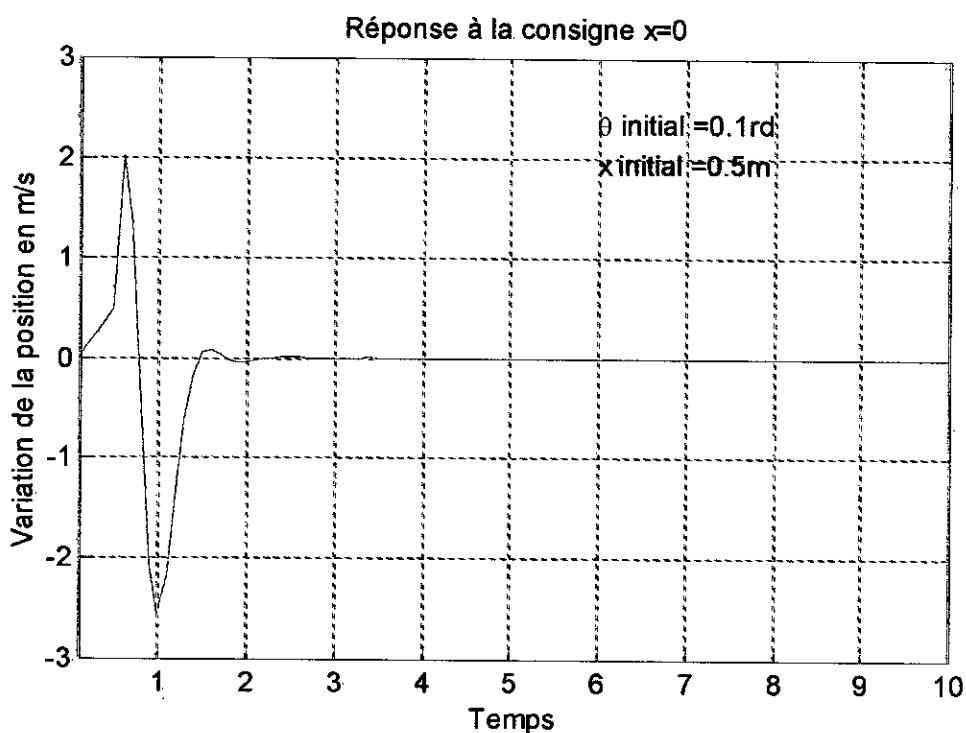
Figure 2.27. Réponse  $\theta(t)$  à la consigne  $x = 0$ Figure 2.28. Réponse  $\theta'(t)$  à la consigne  $x = 0$

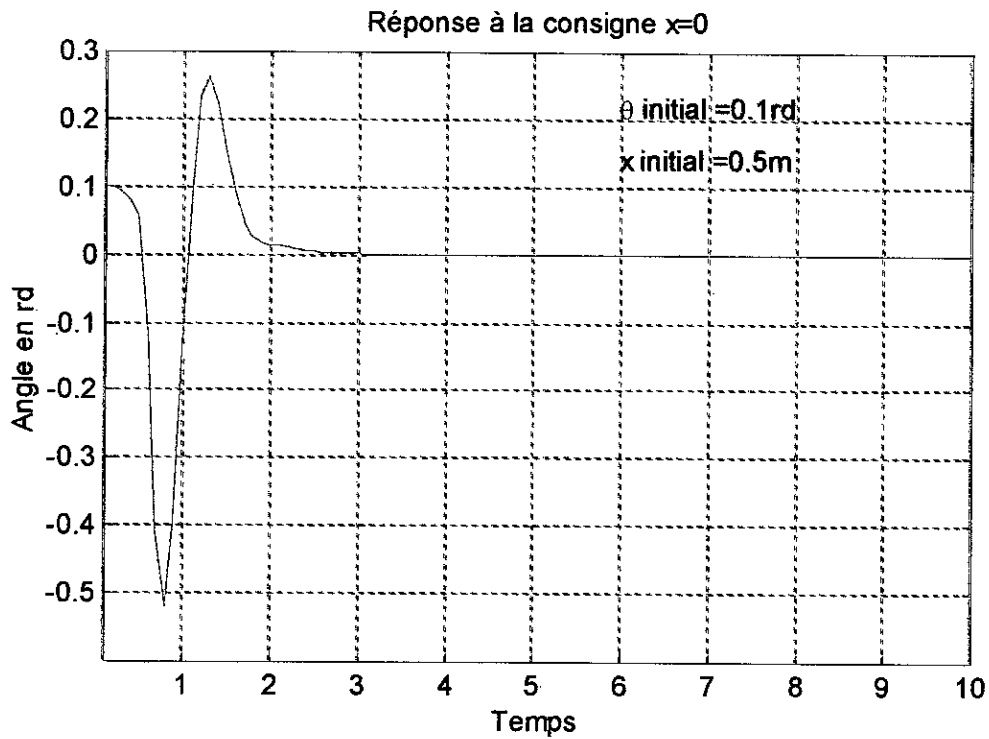
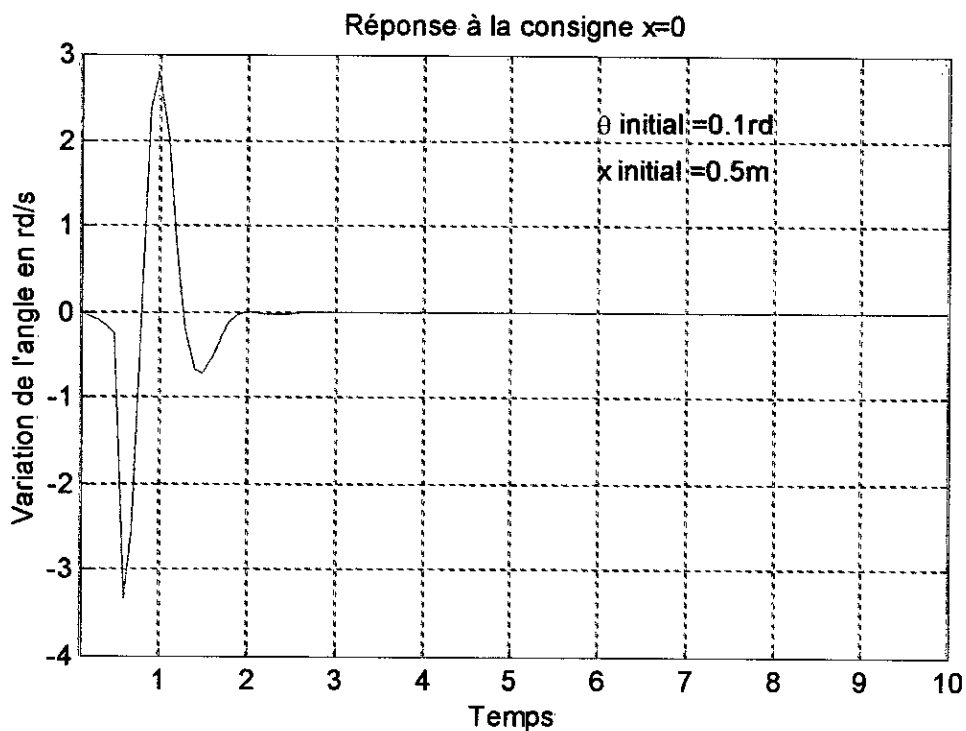


**Figure 2.29. Force appliquée au chariot en  $N$**

On constate que le contrôleur applique un effort brutal afin d'inverser l'angle  $\theta$ , ce qui éloigne un peu plus le chariot de la position à atteindre. La commande peut ensuite amener simultanément le chariot et la barre dans les positions souhaitées.

On prend maintenant l'horizon initial  $N_1 = 5$ , l'horizon de prédiction maximal  $N_2 = 9$ , l'horizon de commande  $N_u = 5$  et le coefficient de pondération du signal de commande  $\lambda = 0.0001$ , on obtient les résultats suivants :

Figure 2.30. Réponse  $x(t)$  à la consigne  $x = 0$ Figure 2.31. Réponse  $x'(t)$  à la consigne  $x = 0$

Figure 2.32. Réponse  $\theta(t)$  à la consigne  $x = 0$ Figure 2.33. Réponse  $\theta'(t)$  à la consigne  $x = 0$

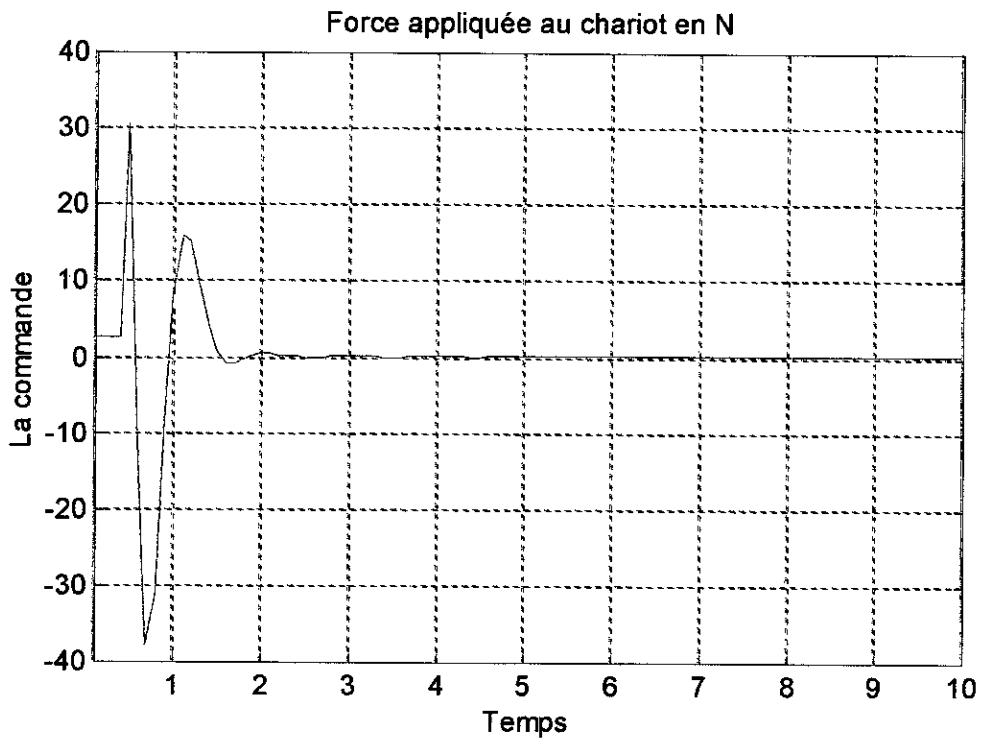


Figure 2.34. Force appliquée au chariot en  $N$

On vérifie à présent la réponse du processus en poursuite en appliquant alternativement une consigne de position de  $\pm 1m$  ( $N_1 = 5$ ,  $N_2 = 9$ ,  $N_u = 5$  et  $\lambda = 0.0001$ )

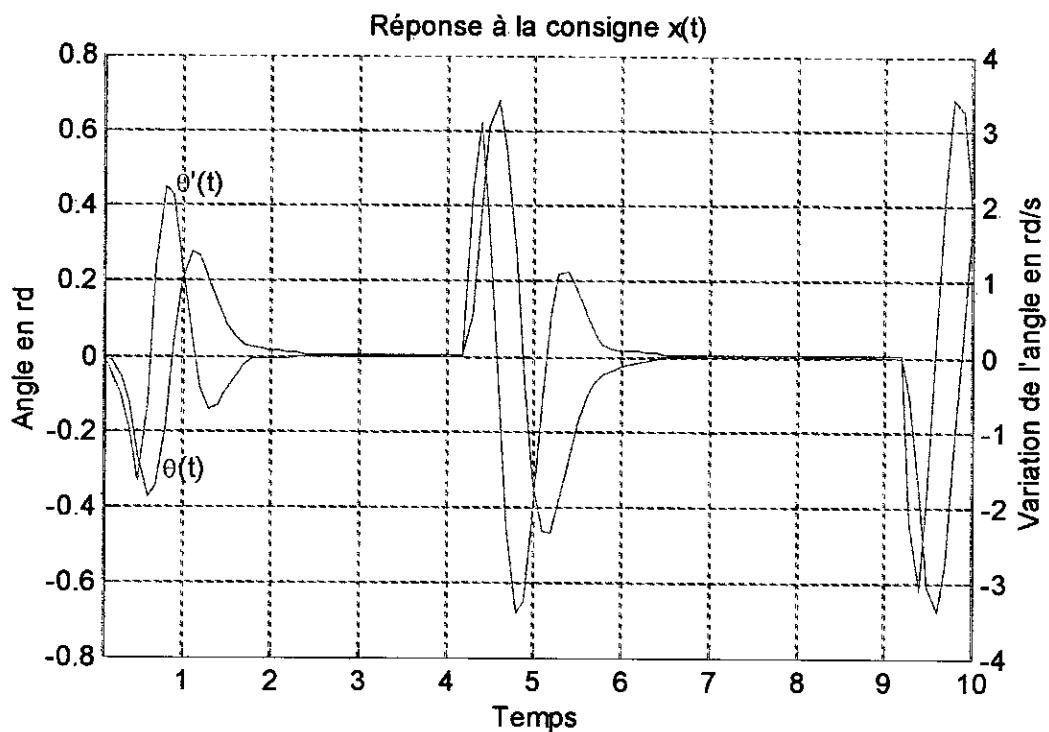


Figure 2.35.

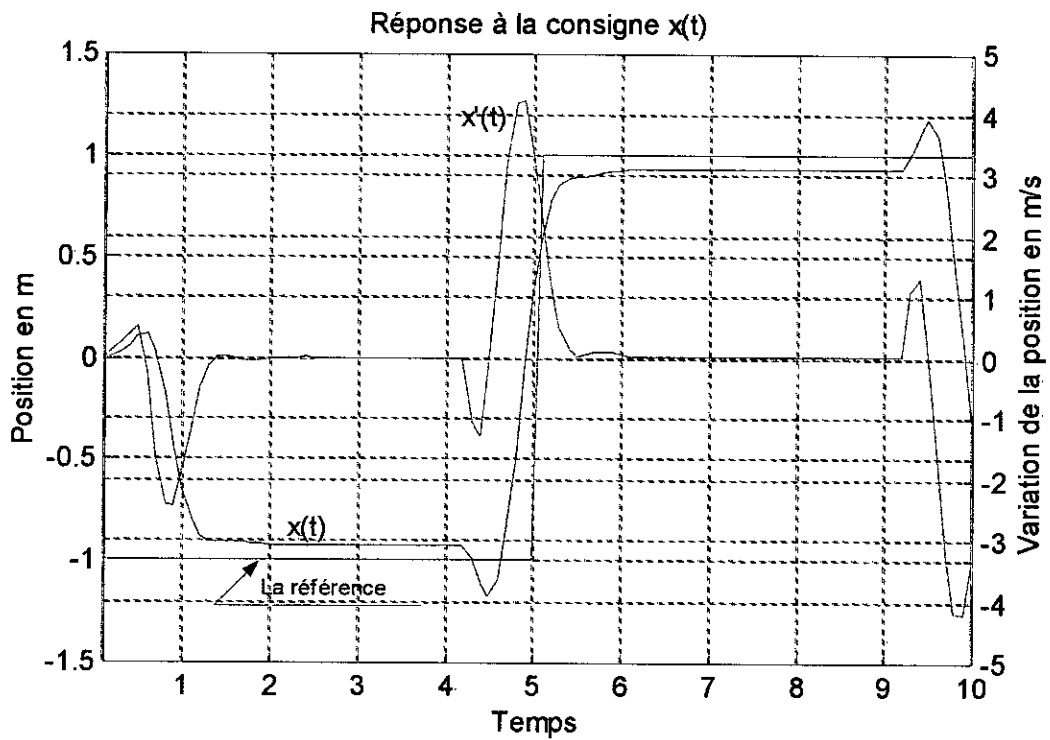
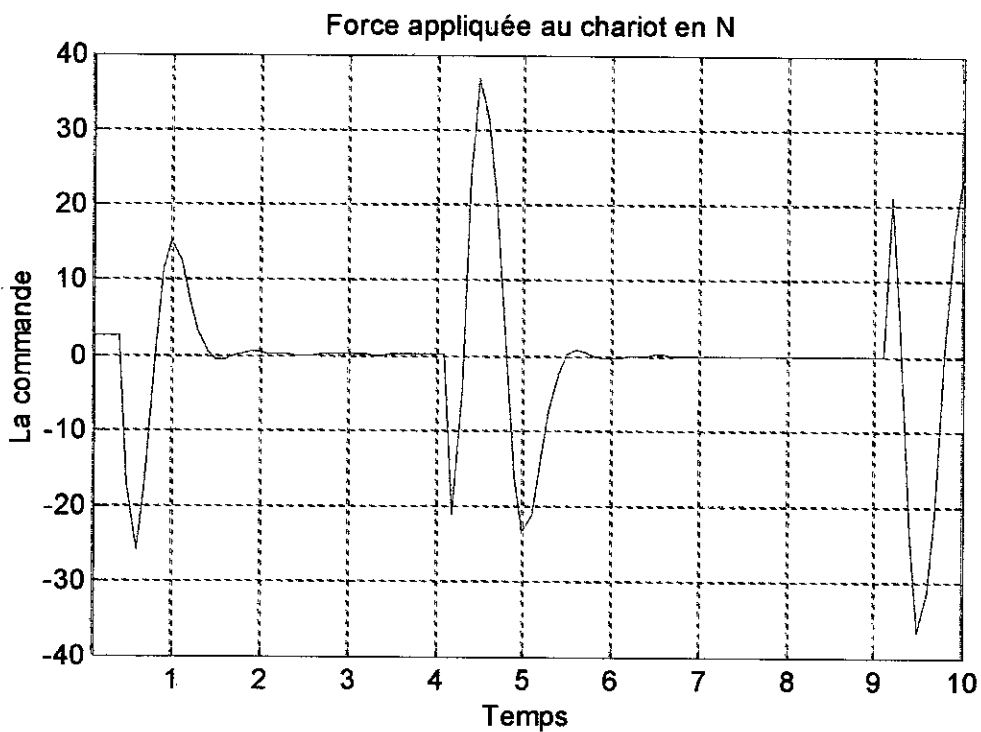


Figure 2.36.

Figure 2.37. Force appliquée au chariot en  $N$

On remarque, à chaque nouvelle consigne de position, l'impulsion de force appliquée dans le but d'incliner la barre dans le sens de déplacement du chariot.

Il s'agit enfin de faire la relation entre le *GPC* et la conception dans l'espace d'état.

N'importe quel contrôleur peut être implanté en application dans le cadre de l'espace d'état par des transformations appropriées. Donc, on peut faire une relation entre le *GPC* et l'espace d'état si la commande minimise une fonction quadratique de *Riccati* qui est peut être utilisée comme fonction de coût.

On considère le système suivant :

$$A(q^{-1})\Delta(q^{-1})y(t) = B(q^{-1})\Delta(q^{-1})u(t-1) \quad 2.90$$

avec  $\Delta(q^{-1}) = 1 - q^{-1}$

dans ce cas, l'inclusion des perturbations n'est pas nécessaire.

Le système est représenté dans l'espace d'état par les équations suivantes :

$$\begin{cases} X(t+1) = AX(t) + B\Delta u(t) \\ Y(t) = CX(t) \end{cases} \quad 2.91$$

où  $A$  est la matrice d'état exprimée sous forme canonique d'observabilité. Le vecteur  $B$  est déterminé selon les paramètres du polynôme  $B(q^{-1})$ .

on définit :

$$\tilde{A}(q^{-1}) = A(q^{-1})\Delta(q^{-1}) = 1 + \tilde{a}_1 q^{-1} + \tilde{a}_2 q^{-2} + \dots + \tilde{a}_n q^{-n} \quad 2.92$$

$$A = \begin{bmatrix} -\tilde{a}_1 & 1 & 0 & \dots & 0 \\ -\tilde{a}_2 & 0 & 1 & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & 1 \\ -\tilde{a}_n & \cdot & \cdot & \cdot & 0 \end{bmatrix} \quad B = [b_0 \quad b_1 \quad \dots \quad b_{n-1}]^T$$

$$\text{où } \tilde{a}_i = 0 \quad \text{pour } i > \deg(\tilde{A}(q^{-1})) \quad 2.93$$

$$\text{et } b_i = 0 \quad \text{pour } i > \deg(B(q^{-1}))$$

pour ce modèle, le critère du *GPC* à minimiser est donné par l'équation suivante :

$$J = X(t+N_2)^T H X(t+N_2) + \sum_{i=t}^{N_2+t-1} [X(i)^T Q X(i) + \lambda(i)\Delta u(i)^2] \quad 2.94$$

$N_2$  désigne l'horizon de prédiction maximal.

$Q$  et  $H$  sont des matrices de pondération, symétriques. La matrice  $H$  ou terme de pondération terminal permet de pénaliser plus ou moins l'écart final par rapport à la cible cherchée.

$Q$  est une matrice de pondération de l'état intermédiaire,  $\lambda$  est un facteur de pondération des signaux de commande.

Le problème consiste à minimiser le critère  $J$  de l'équation (2.94) sous la contrainte suivante :

$$X(t+1) = AX(t) + B\Delta u(t) \quad 2.95$$

Le contrôle qui minimise la fonction de coût de l'équation (2.94) est donné par :

$$\begin{aligned} \Delta u(t) &= -K^T X(t) \\ K^T &= [\lambda(t) + B^T P(t)B]^{-1} B^T P(t)A \end{aligned} \quad 2.96$$

Donc, la minimisation du critère quadratique sous la contrainte précédente permet de résoudre les équations suivantes :

$$\begin{aligned} P^*(i) &= P(i+1) - P(i+1)B[\lambda(i) + B^T P(i+1)B]^{-1} B^T P(i+1) \\ P(i) &= Q + A^T P^*(i)A \end{aligned} \quad 2.97$$

### théorème

Un système en boucle fermée est équivalent à un contrôleur stable (state-dead-beat)

Si

1. Le système  $(A, B, C)$  est observable et contrôlable et
2.  $N_1 = n, N_2 \geq 2n-1, N_u = n, \lambda = 0$  ; où  $n$  est le nombre des états du procédé.

Commençons par  $N_2 = 2n-1$ , le coût à minimiser est donné par l'équation suivante :

$$J = \sum_{i=t}^{N_2+t} X(i)^T Q(i)X(i) \quad 2.98$$

avec  $Q(i) = 0$  pour  $i < t+N_1$  et  $Q(i) = C^T C$  pour  $i \geq t+N_1$ .

le gain de *Kalman* est calculé en utilisant les itérations de l'équation de *Riccati* donnée par (2.97).



pour les premières  $(n-1)$  itérations,  $Q = C^T C$  et pour les  $(n)$  itérations suivantes  $Q = 0$ . Avec ces itérations, on obtient à la fin et à l'instant  $t$  un gain de Kalman unique avec  $P(t+N_2) = C^T C$ .

Du point de vue de la prédiction, le calcul de  $\Delta u(t)$  est équivalent de résoudre l'équation suivante :

$$\begin{bmatrix} g_{n-1} & g_{n-2} & \dots & g_0 \\ g_n & g_{n-1} & \dots & g_1 \\ \cdot & \cdot & \dots & \cdot \\ \cdot & \cdot & \dots & \cdot \\ g_{2n-2} & g_{2n-1} & \dots & g_{n-1} \end{bmatrix} \times \begin{bmatrix} \Delta u(t) \\ \Delta u(t+1) \\ \cdot \\ \cdot \\ \Delta u(t+n-1) \end{bmatrix} = \begin{bmatrix} w(t+n) - f(t+n) \\ w(t+n+1) - f(t+n+1) \\ \cdot \\ \cdot \\ w(t+2n-1) - f(t+2n-1) \end{bmatrix} \quad 2.99$$

Clairement, si la matrice  $G_1$  est de rang maximal,  $\Delta u(t)$  est unique.

Mais la matrice  $G_1$  donnée par l'équation (2.26) est remplacée maintenant par la formule suivante :

$$G_1 = \begin{bmatrix} C \\ CA \\ \cdot \\ \cdot \\ CA^{n-1} \end{bmatrix} \begin{bmatrix} A^{n-1}B & A^{n-2}B & \dots & B \end{bmatrix} \quad 2.100$$

par la supposition (1),  $\Delta u(t)$  minimisant le coût est unique parce que  $G_1$  est du rang maximal. Dans notre cas, le système est représenté par les équations suivantes :

$$\begin{cases} \dot{X} = AX + Bu \\ Y = CX \end{cases} \quad 2.101$$

le système  $(A, B, C)$  est observable et contrôlable. Les matrices  $A$  et  $B$  sont données par l'équation (2.70) et la matrice  $C$  est donnée par l'équation suivante :

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \quad 2.102$$

(le modèle est discrétisé avec une période d'échantillonnage égale à 0.1s)

on a pris  $N_2 = 7$ ,  $N_u = 4$ , l'horizon initial  $N_1 = 4$  et le coefficient de pondération du signal de commande  $\lambda = 0.0001$ .

Il s'agit d'asservir deux sorties (la position  $x$  et l'angle  $\theta$ ) en même temps. Donc, la matrice  $G_1$  a la forme suivante :

$$G_1 = \begin{bmatrix} C_1 \\ C_1 A \\ \vdots \\ C_1 A^{n-1} \\ C_2 \\ C_2 A \\ \vdots \\ C_2 A^{n-1} \end{bmatrix} \begin{bmatrix} A^{n-1} B & A^{n-2} B & \dots & B \end{bmatrix} \quad 2.103$$

avec  $C_1$  est la première ligne de la matrice  $C$  et  $C_2$  étant la deuxième ligne.

On a pris  $n = 4$  (le nombre des états du système) et la commande est tel que :

$$u(t) = u(t-1) + \tilde{g}^T (W - F) \quad 2.104$$

avec  $\tilde{g}^T$  est la première ligne de  $(G_1^T G_1 + \lambda I)^{-1} G_1^T$ .

Les résultats obtenus par cette méthode sont représentés par les figures suivantes :

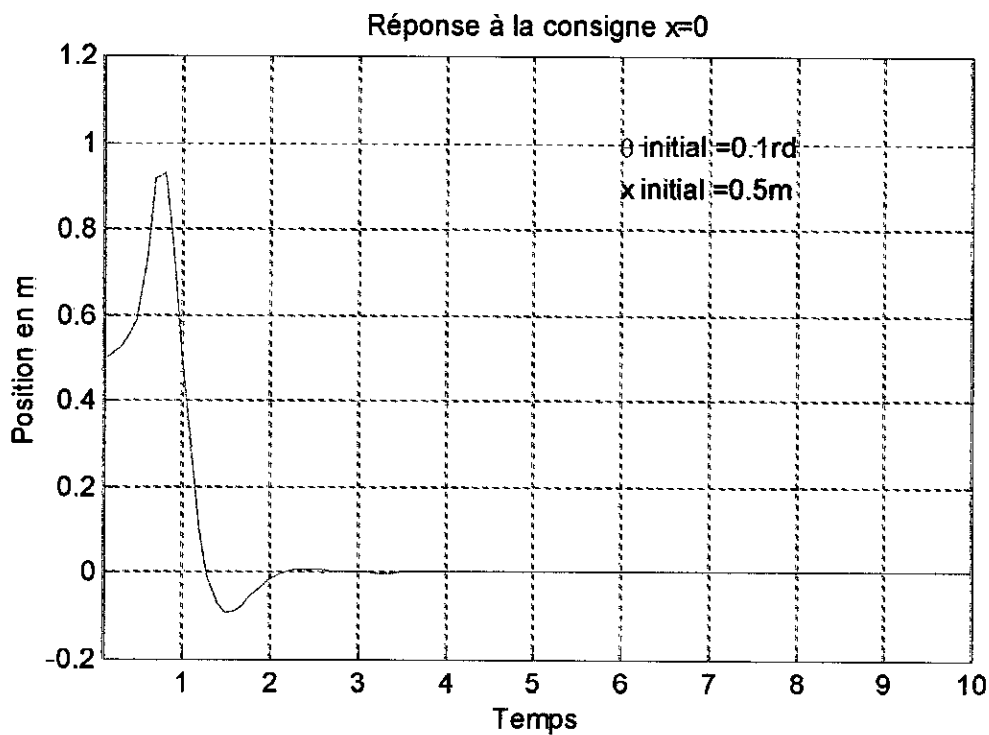
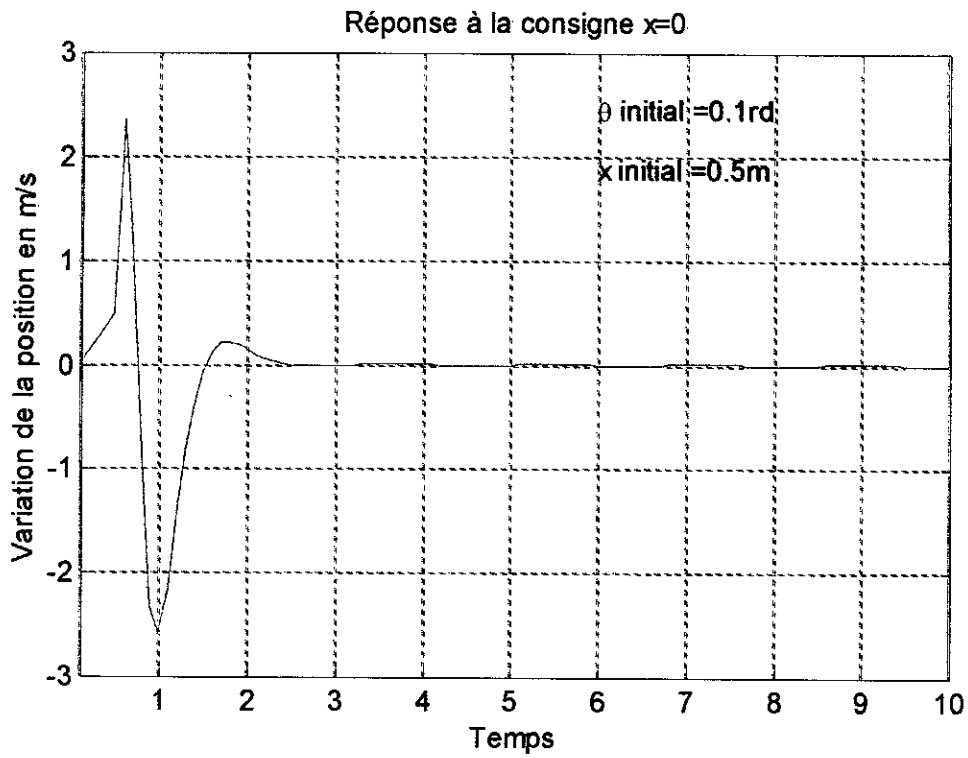
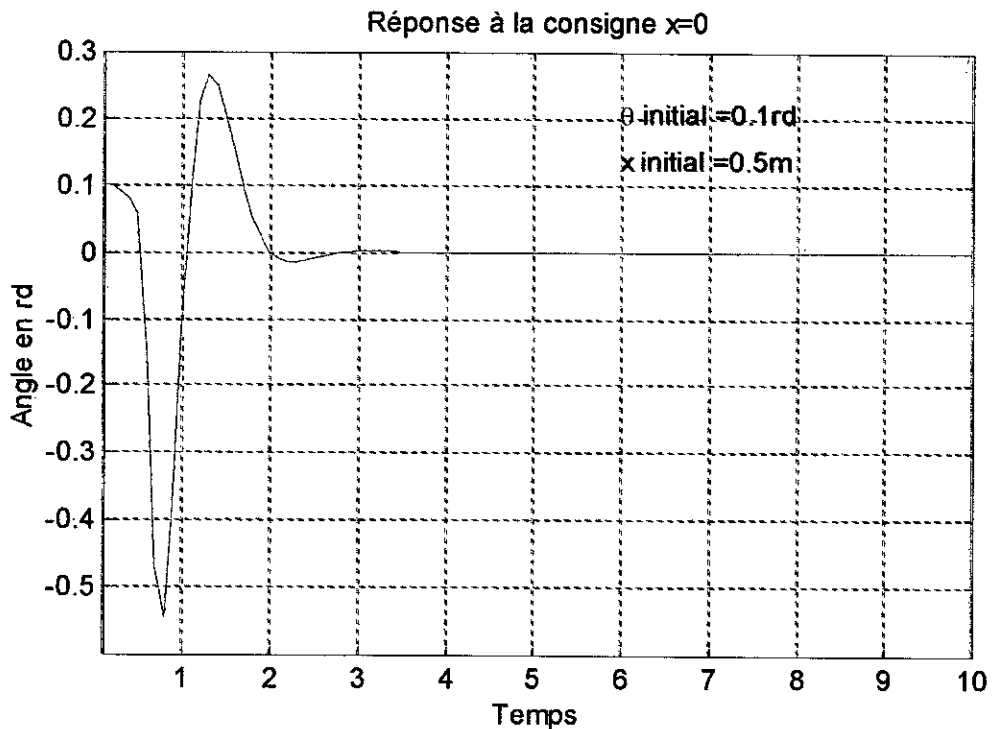


Figure 2.38. Réponse  $x(t)$  à la consigne  $x = 0$

Figure 2.39. Réponse  $x'(t)$  à la consigne  $x = 0$ Figure 2.40. Réponse  $\theta(t)$  à la consigne  $x = 0$

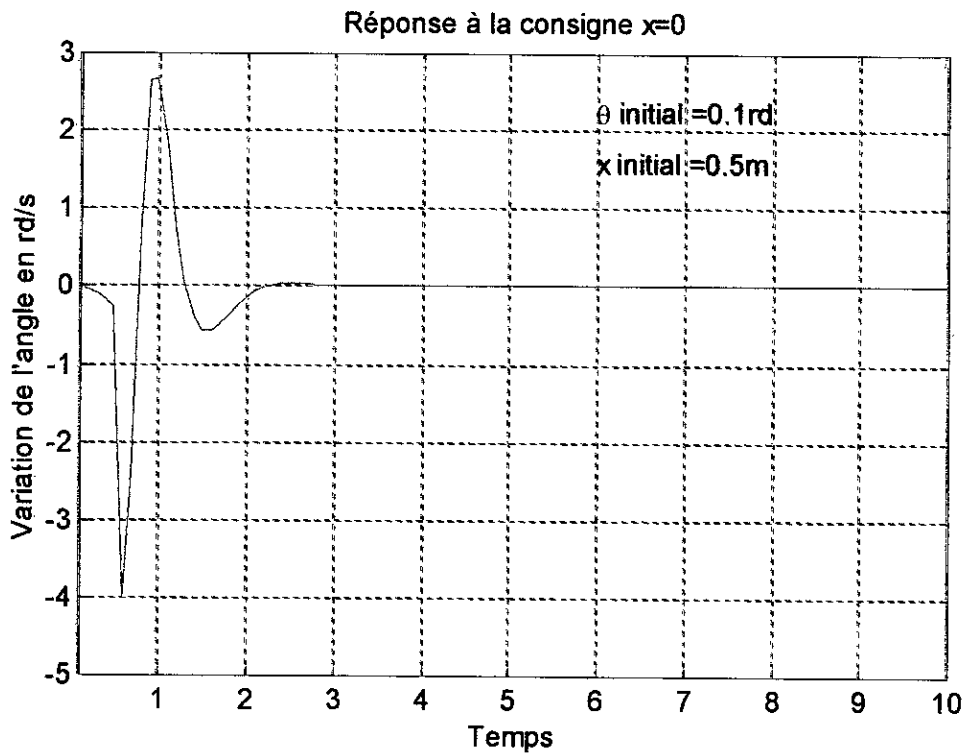


Figure 2.41. Réponse  $\theta'(t)$  à la consigne  $x = 0$

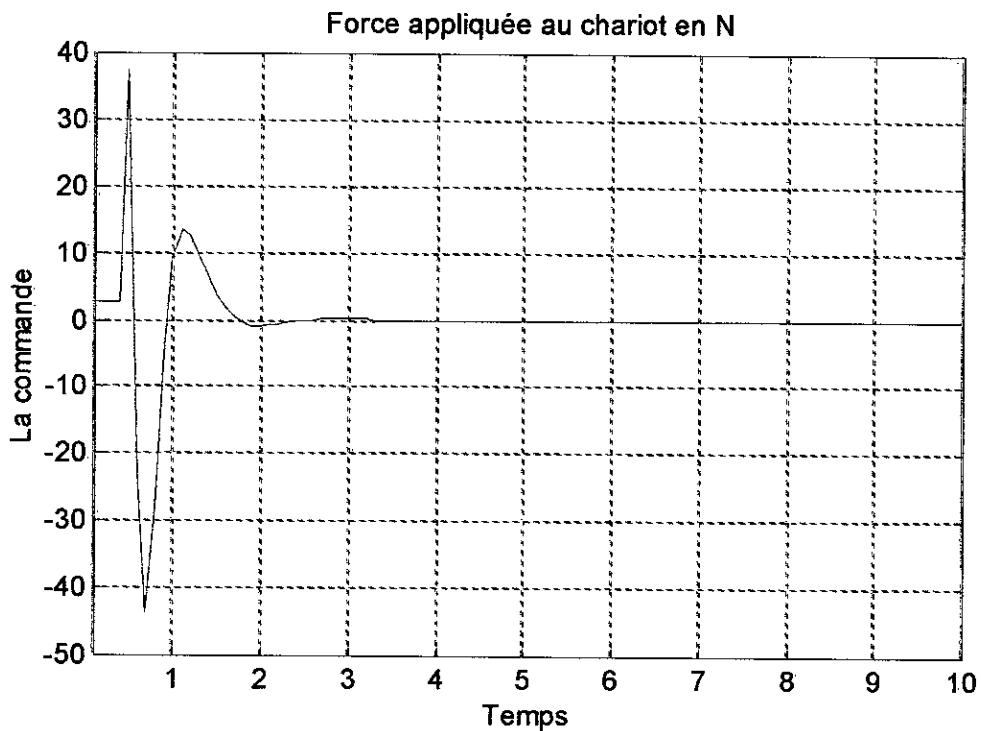


Figure 2.42. Force appliquée au chariot en N

On peut conclure, à partir des résultats obtenus, que le contrôle prédictif généralisé est le meilleur pour l'asservissement de  $x$  et  $\theta$ . Mais le problème rencontré est comment choisir les paramètres du *GPC*, surtout les deux horizons  $N_2$  et  $N_u$ , l'horizon de prédiction maximal et l'horizon de commande.

## 2.4.2. Simulation 2

Dans cette simulation, on va montrer l'influence des horizons de prédiction maximal  $N_2$  et de commande  $N_u$  sur la réponse d'un système quelconque.

### 2.4.2.1. Exemple 1

Le premier exemple à traiter est un simple processus de premier ordre à phase non-minimale d'écrit par :

$$\begin{aligned} (1 + a_1 q^{-1})y(t) &= (b_0 + b_1 q^{-1})u(t-1) \\ (1 - 0.9q^{-1})y(t) &= (1 + 2q^{-1})u(t-1) \end{aligned} \quad 2.105$$

on a pris  $N_1 = 1$  et  $N_2$  varie entre 1 et 3.

Les résultats obtenus sont montrés par les figures suivantes :

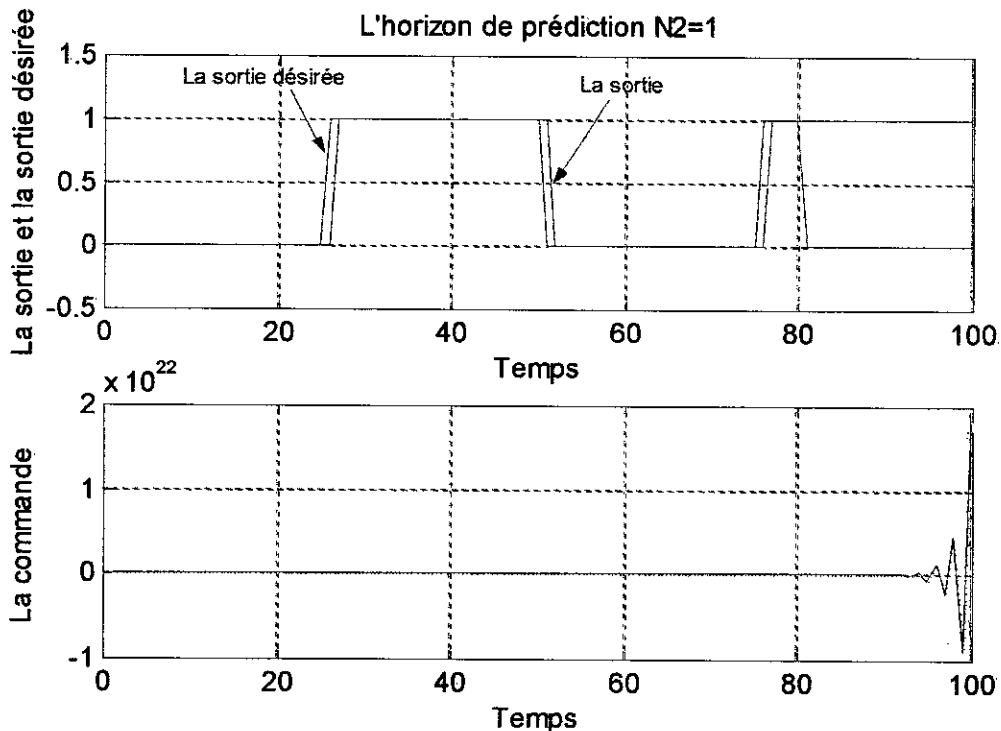


Figure 2.43. Résultat du contrôle prédictif pour  $N_2 = 1$

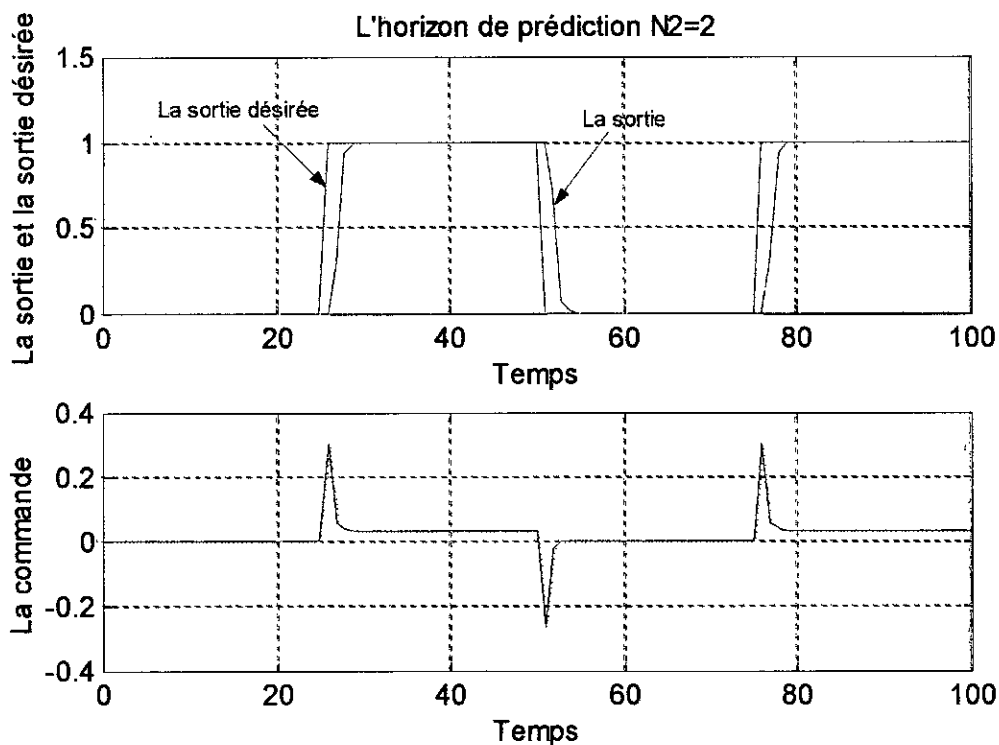


Figure 2.44. Résultat du contrôle prédictif pour  $N_2 = 2$

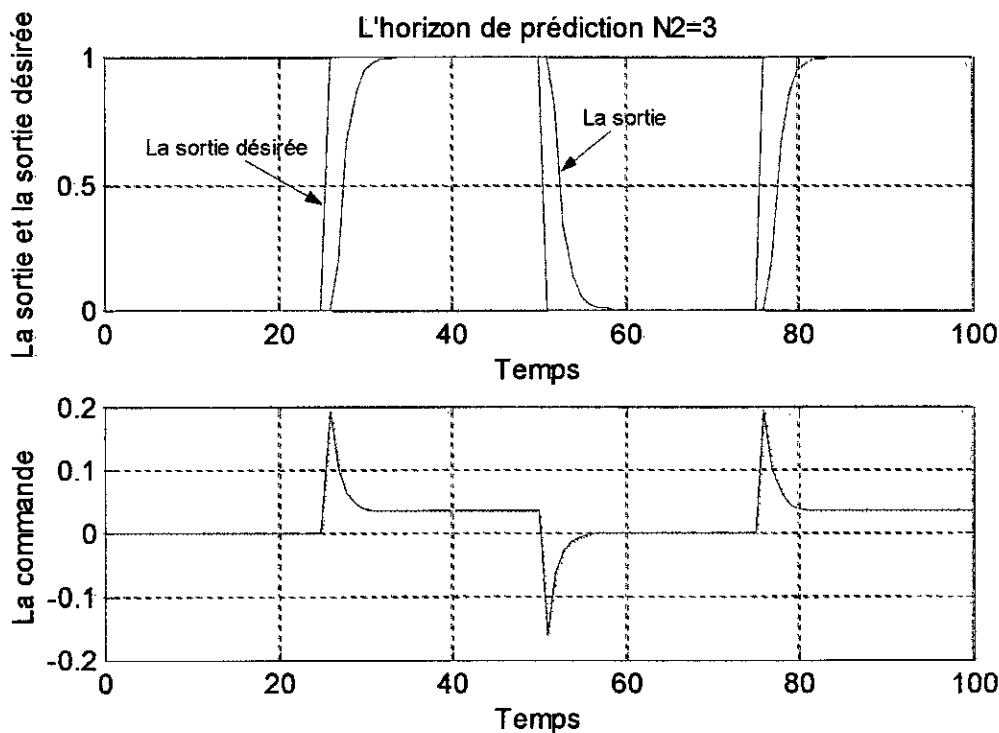


Figure 2.45. Résultat du contrôle prédictif pour  $N_2 = 3$

Pour éviter la divergence de la sortie, il faut que  $N_2$  dépasse le degré du polynôme  $B(q^{-1})$ .

### 2.4.2.2. Exemple 2

Le deuxième exemple à traiter est un procédé double oscillateur représenté par la fonction de transfert suivante :

$$(1 + s^2)(1 + 1.5s^2)y(t) = u(t) \quad 2.106$$

dans cet exemple, on considère une sortie auxiliaire donnée par :

$$\psi(t) = P(q^{-1})y(t) \quad \text{où} \quad P(q^{-1}) = \frac{P_n(q^{-1})}{P_d(q^{-1})}$$

$$P_n(q^{-1}) = (1 - 0.5q^{-1})^2 \quad 2.107$$

$$P_d(q^{-1}) = 0.2778(1 - 0.1q^{-1})$$

La fonction de coût à minimiser est :

$$J(N_1, N_2, N_u) = E \left\{ \sum_{j=N_1}^{N_2} [\psi(t+j) - w(t+j)]^2 + \sum_{j=1}^{N_u} \lambda(j) [\Delta u(t+j-1)]^2 \right\} \quad 2.108$$

on a pris  $N_1 = 1$ ,  $N_2 = 10$  et  $N_u$  prend les valeurs 1, 2, 4 et 6.

Les résultats obtenus sont montrés par les figures suivantes :

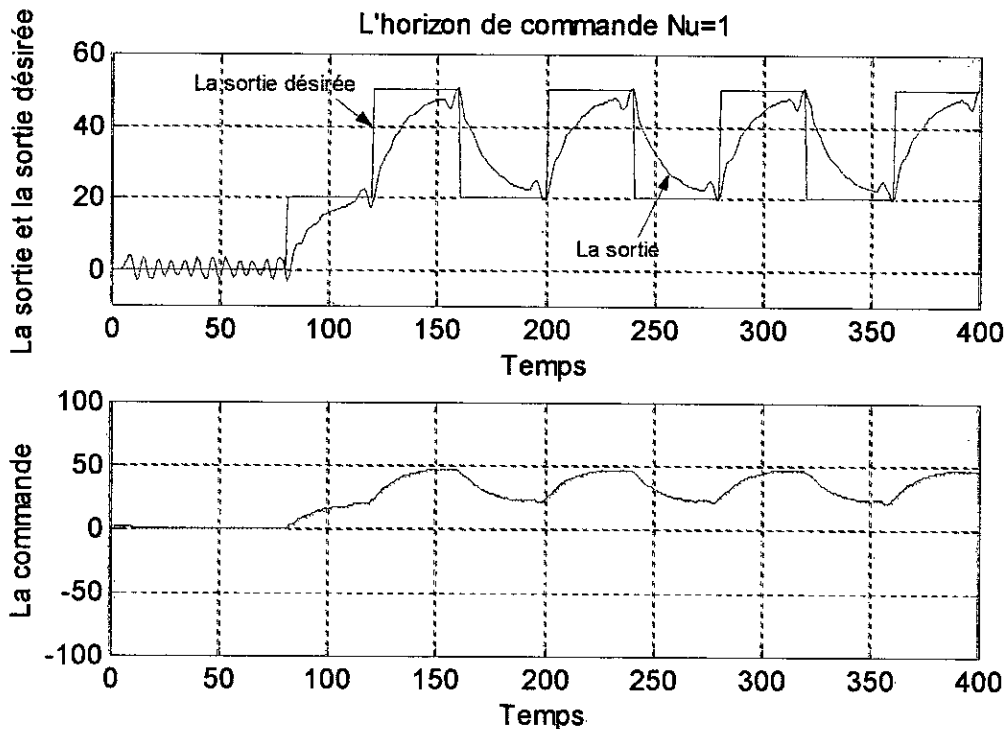


Figure 2.46. Résultat du contrôle prédictif pour  $N_u = 1$

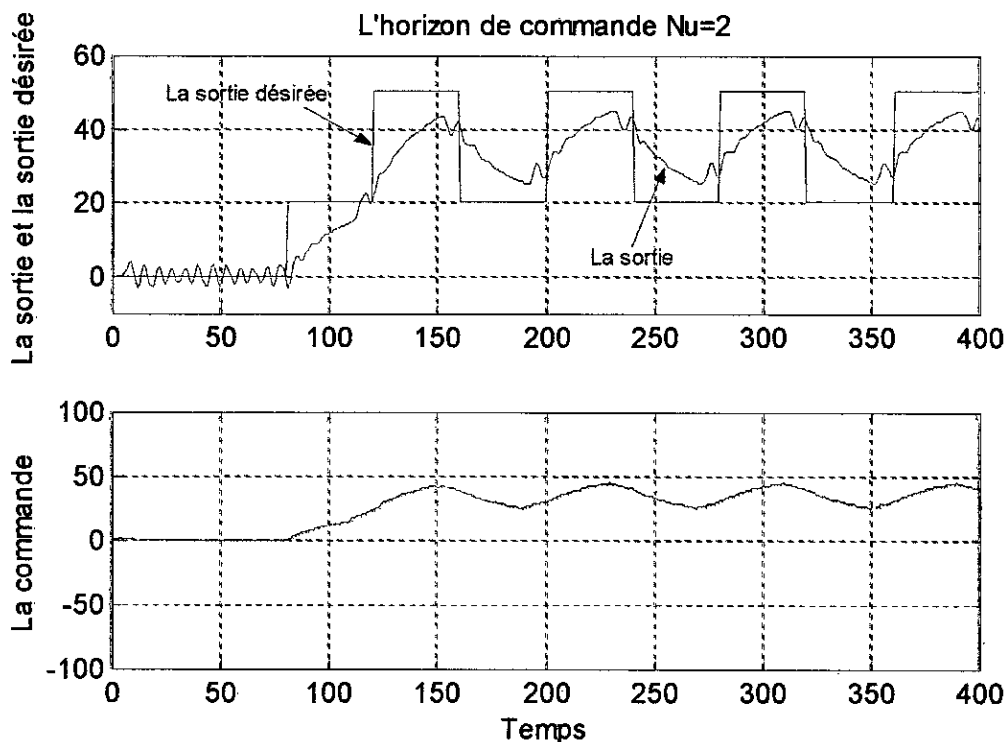


Figure 2.47. Résultat du contrôle prédictif pour  $N_u = 2$

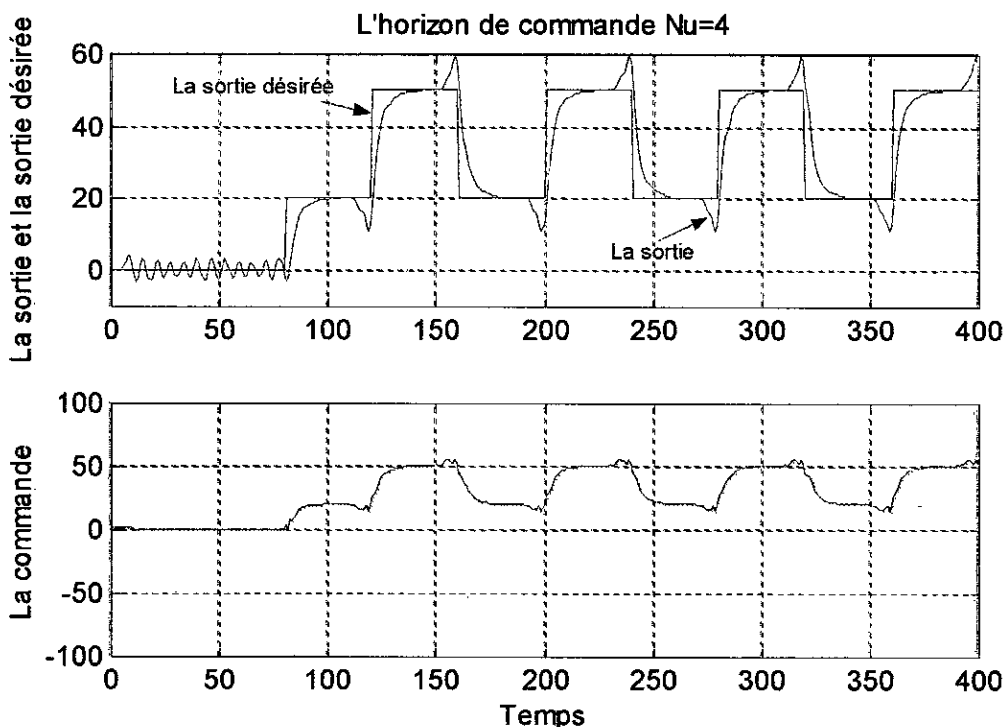


Figure 2.48. Résultat du contrôle prédictif pour  $N_u = 4$



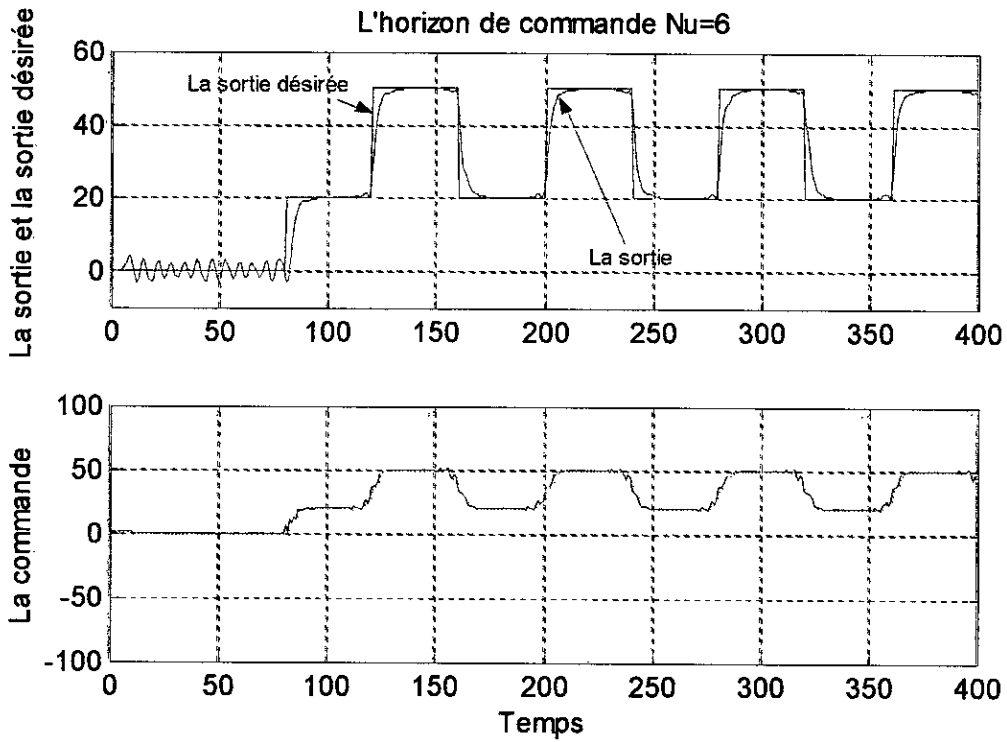


Figure 2.49. Résultat du contrôle prédictif pour  $N_u = 6$

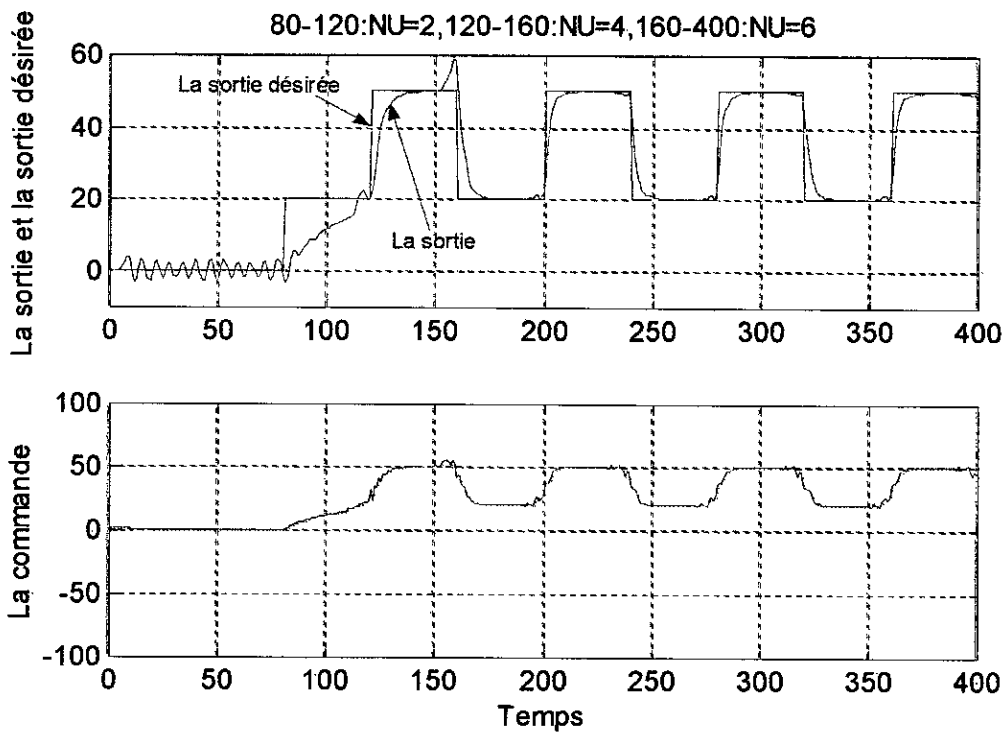


Figure 2.50.

le bon contrôle est réalisé quand  $N_u$  est au moins égal au nombre des pôles du système. Donc il faut choisir  $N_u > 4$  (le nombre des pôles).

## 2.5. Conclusion

Le contrôle prédictif généralisé *GPC* dépend de l'intégration de cinq idées : la supposition du modèle *CARIMA*, l'utilisation d'une prédiction à longue portée sur un horizon fini, la récursion des équations Diophantine, la considération d'une pondération des incréments futurs dans la fonction de coût et le choix de l'horizon de contrôle. Et si l'on veut tirer tous les avantages, le *GPC* s'appuie sur ces principes : modèle interne, trajectoire de référence et structuration de la variable manipulée.

Dans ce chapitre, on a utilisé le *GPC* comme contrôleur d'un système non linéaire qui est le chariot avec pendule inversé. Dans un premier temps, on a imposé uniquement de maintenir le bras du pendule inversé en position verticale (on n'a considéré que le degré de liberté selon  $\theta$ ). Puis dans un second temps, on a réalisé l'asservissement complet (on a pris en compte les deux degrés de liberté,  $x$  et  $\theta$ ) c'est à dire l'asservissement de la position du chariot tout en maintenant le pendule inversé dans sa position verticale. Et pour réaliser ce deuxième but, on a utilisé aussi la commande optimale pour faire la comparaison entre ces deux méthodes.

Après plusieurs tests de simulation, on a remarqué l'efficacité de ce contrôleur pour trouver les meilleures performances (rapidité, minimisation de la fonction coût). Mais, le problème rencontré est le choix des paramètres de réglage, essentiellement l'horizon de prédiction, l'horizon de commande et le facteur de pondération.

# Chapitre 3

## Les algorithmes génétiques

### 3.1. Introduction

Les algorithmes génétiques sont des techniques de recherche stochastique dont les fondements théoriques ont été définis par *John Holland* [6][22].

Les recherches de *Holland*, ses collègues et ses étudiants, avaient deux objectifs principaux :

- Mettre en évidence et expliquer rigoureusement les processus d'adaptation des systèmes naturels.
- Concevoir des systèmes artificiels (en l'occurrence des logiciels) qui possèdent les propriétés importantes des systèmes naturels [6].

Les algorithmes génétiques sont des algorithmes d'exploration fondés sur les mécanismes de la sélection naturelle et de la génétique. Ils utilisent à la fois les principes de la survie des structures les mieux adaptées, et les échanges d'information pseudo-aléatoires, pour former un algorithme d'exploration qui possède certaines des caractéristiques de l'exploration humaine. A chaque génération, un nouvel ensemble de créatures artificielles (des chaînes de caractères) est créé en utilisant des parties des meilleurs éléments de la génération précédente, ainsi que des parties innovatrices, à l'occasion [6].

De façon très intuitive, on identifie le problème à un environnement donné et les solutions à des individus évoluant dans cet environnement. A chaque génération, on ne retient que les individus les mieux adaptés à cet environnement. Au bout d'un certain nombre de générations, les individus restants sont particulièrement adaptés à l'environnement donné. On obtient donc des solutions très proches de la solution idéale du problème.

Les algorithmes génétiques utilisent un mécanisme de sélection naturelle. Ainsi, les AGs ne se basent pas sur un *individu*, mais sur une *population* d'individus qui vont évoluer de génération en génération pour obtenir un résultat se rapprochant de la solution optimale. Les meilleurs individus d'une *génération* vont créer une nouvelle génération plus adaptée au problème. Bien qu'utilisant le hasard, les AGs ne sont pas purement aléatoires car on suit des lois probabilistes. Les AGs exploitent efficacement l'information obtenue précédemment pour spéculer sur la position de nouveaux points à explorer, avec l'espoir d'améliorer la performance.

De plus, les AGs ne sont pas fondamentalement limités par des hypothèses contraignantes sur le domaine d'exploration (hypothèses concernant la continuité, l'existence de dérivée, l'unimodalité...) [6].

Dans la pratique, un grand nombre de fonctions à optimiser ne sont pas dérivables et souvent même pas continues. Le monde réel à explorer est envahi de discontinuités, d'espaces de recherche bruités et multimodaux, ce qui le rend bien moins adapté au calcul. Il est évident que les méthodes soumises aux contraintes de continuité et de dérivabilité ne sont adaptées qu'à une classe de problèmes très limitée. Elles ne sont pas suffisamment robustes.

### 3.2. Ce qui rend les algorithmes génétiques différents des méthodes classiques

Un algorithme génétique recherche le ou les *extrema* d'une fonction définie sur un espace de données. Pour qu'il puisse surpasser les autres méthodes classiques, un algorithme génétique doit être fondamentalement différent.

Il est en fait différent selon quatre axes principaux :

- Un algorithme génétique utilise un codage des paramètres, et non les paramètres eux mêmes.

- Un algorithme génétique travaille sur une population de points, au lieu d'un point unique.
- Un algorithme génétique n'utilise que les valeurs de la fonction étudiée, pas sa dérivée, ou une autre connaissance auxiliaire.
- Un algorithme génétique utilise des règles de transition probabilistes, et non déterministes [6].

Les algorithmes génétiques nécessitent le codage de l'ensemble des paramètres d'origine du problème d'optimisation en une chaîne de caractères et de longueur finie.

Dans beaucoup de méthodes d'optimisation, on se déplace avec précaution d'un point unique à un autre dans l'espace de recherche, en utilisant une règle de transition pour déterminer le nouveau point. Cette méthode qui fonctionne point par point est dangereuse parce qu'elle a une forte tendance à trouver de faux pics dans des espaces multimodaux (multipics). En revanche, les algorithmes génétiques utilisent simultanément un ensemble de points (une population de chaînes), en escaladant plusieurs pics en parallèle ; ainsi, la probabilité de trouver un faux sommet est réduite par rapport à d'autres méthodes qui explorent point par point.

Les méthodes d'exploration nécessitent beaucoup d'informations auxiliaires pour bien fonctionner. Par exemple, les méthodes de *gradient* ont besoin des dérivées (calculées analytiquement ou numériquement) afin de pouvoir escalader le pic local. En revanche, les algorithmes génétiques n'ont aucun besoin de cette information auxiliaire : les AGs sont aveugles. Pour effectuer une recherche performante de structures de plus en plus intéressantes, ils n'ont besoin que des valeurs de la fonction à optimiser associées à chaque chaîne. Cette caractéristique fait des AGs une méthode plus générale que beaucoup de procédures d'exploration.

Contrairement à beaucoup de méthodes, les AGs utilisent des règles de transition probabilistes afin de guider leur exploration : pour les personnes habituées aux méthodes déterministes, cela semble bizarre, mais l'utilisation de probabilités ne signifie pas que la méthode n'est qu'une exploration aléatoire, cela n'a rien à avoir avec un simple tirage à pile ou face. Les algorithmes génétiques utilisent des choix aléatoires comme des outils

pour guider l'exploration à travers les régions de l'espace de recherche, avec une amélioration probable.

Les quatre différences : l'utilisation d'un codage, le travail sur une population, l'indifférence à l'information auxiliaire et les opérateurs aléatoires contribuent à la robustesse des algorithmes génétiques.

### 3.3. Terminologie et définitions

L'*environnement* ou *espace de recherche* définit l'ensemble des configurations possibles des paramètres de la fonction à optimiser. Un *individu* représente un point de cet espace, la *population* désignant un groupe d'individus.

On qualifie la fonction à optimiser de fonction *objectif*, fonction *coût* (pour les problèmes de minimisation), fonction *profit* (pour les problèmes de maximisation), fonction d'*évaluation*, fonction d'*adaptation*, voire fonction de forme ( *fitness* ). Elle mesure la *performance* des individus dans leur environnement et caractérise le relief de l'espace de recherche.

Le *chromosome* est une chaîne d'allèles qui contient les caractéristiques génétiques d'un individu et code les paramètres de sa fonction d'adaptation. Le *gène* ou *allèle* est la partie élémentaire du chromosome. Il représente un trait de caractère ou une fonction particulière. Si le chromosome est codé en binaire, un gène sera symbolisé par un bit.

Donc, les gènes sont des caractères dont leurs valeurs sont appelées allèles. Ils sont juxtaposés d'une façon linéaire pour former ce qu'on appelle le *génotype* ou le *chromosome*. Le *phénotype* est l'expression physique du génotype, à partir de laquelle une fonction de performance est calculée pour évaluer le génotype correspondant [21].

### 3.4. Principes généraux

Pour utiliser un algorithme génétique, on doit disposer des cinq éléments suivants:

#### 1. Un principe de codage de l'élément de population :

Cette étape associe à chacun des points de l'espace d'état une structure de données. Elle se place généralement après une phase de modélisation mathématique du problème traité. La qualité du codage des données conditionne le succès des algorithmes génétiques.

Selon *Goldberg*, l'utilisateur doit choisir le plus petit alphabet qui permette une expression naturelle du problème (principe des alphabets minimaux). C'est pourquoi, l'alphabet binaire  $\{0, 1\}$  est particulièrement bien adapté à la représentation des paramètres.

Pour chaque paramètre  $x_i$  situé dans l'intervalle  $[x_{imin}, x_{imax}]$ , on associe une chaîne binaire  $b_0 b_1 \dots b_{l_{x_i}-1}$  définie sur  $l_{x_i}$  bits. A cette chaîne correspond une valeur entière naturelle,

$$v(x_i) = \sum_{j=0}^{l_{x_i}-1} 2^{l_{x_i}-j-1} b_j \quad 3.1$$

Le paramètre réel  $x_i$  de l'espace de recherche relatif à  $v(x_i)$  est obtenu par interpolation linéaire :

$$x_i = x_{imin} + \frac{x_{imax} - x_{imin}}{2^{l_{x_i}} - 1} v(x_i) \quad 3.2$$

La précision des paramètres a une valeur  $\varepsilon_i$  correspondant à l'écart entre deux configurations réelles adjacentes.

$$\varepsilon_i = \frac{x_{imax} - x_{imin}}{2^{l_{x_i}} - 1} \quad 3.3$$

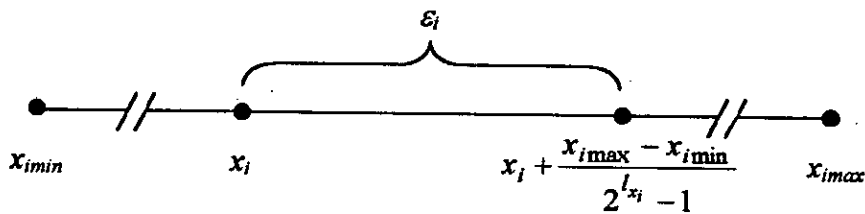


Figure 3.1. Précision maximale de la méthode de codage binaire

La longueur  $l_{x_i}$  de la sous chaîne binaire nécessaire pour obtenir une précision  $\varepsilon_i$  pour le paramètre réel  $x_i$  correspond donc au plus petit entier naturel tel que :

$$l_{x_i} \geq \frac{1}{\ln 2} \ln \left( \frac{x_{imax} - x_{imin}}{\varepsilon_i} + 1 \right) \quad 3.4$$

La longueur totale du chromosome est donnée par :

$$l = \sum_{i=1}^n l_{x_i} \quad \text{avec } n \text{ est le nombre total des paramètres} \quad 3.5$$

Donc pour construire un codage multiparamétré, on peut tout simplement utiliser le codage concaténé à borne fixe. Dans un *AG*, la longueur de la chaîne est fixée par l'utilisateur selon le domaine de variation du paramètre codé et la précision demandée. Pour optimiser une fonction à plusieurs variables  $f(x_1, x_2, \dots, x_n)$ , on utilise parfois un codage binaire concaténé.

Soit  $chaîne = (code_1)(code_2)\dots(code_n)$  où  $(code_j)$  est le code du  $j^{ème}$  paramètre dans l'espace de recherche de dimension  $n$ .

### 2. Un mécanisme de génération de la population initiale :

Ce mécanisme doit être capable de produire une population d'individus non homogène qui servira de base pour les générations futures. Le choix de la population initiale est important car il peut rendre plus ou moins rapide la convergence vers l'optimum global. Dans le cas où l'on ne connaît rien du problème à résoudre, il est essentiel que la population initiale soit répartie sur tout le domaine de recherche.

### 3. Une fonction à optimiser :

Pour appliquer les *AGs*, on doit avoir une fonction de coût ou une fonction d'évaluation  $g(.)$  qui est minimisée indirectement ; sa valeur est élaborée en une valeur d'aptitude  $f(.)$  qui doit être maximisée. La fonction à optimiser retourne une valeur appelée *fitness* ou fonction d'évaluation de l'individu. Cette fonction doit être capable d'interpréter les données contenues dans les chromosomes et de décider si la solution résultante est optimale.

En raison de l'analogie avec la théorie de l'évolution (survie des individus les mieux adaptés à leur environnement), l'algorithme génétique est naturellement formulé en terme de maximisation. Etant donnée une fonction  $f$  réelle a une ou plusieurs variables, le problème d'optimisation sur l'espace de recherche  $E$  s'écrit de la manière suivante :

$$\max_{x \in E} f(x) \quad 3.6$$

De plus, la fonction à optimiser par un algorithme génétique doit avoir des valeurs positives sur l'ensemble du domaine  $E$ . Dans le cas contraire, il convient d'ajouter aux valeurs de  $f$  une constante positive  $F_{min}$  conformément à l'équivalence de (3.6) et (3.7).



$$\max_{x \in E} f(x) + F_{\min} \quad 3.7$$

On peut prendre  $F_{\min}$  comme coefficient d'entrée, ou comme la valeur absolue de la plus faible valeur de  $f$  dans la génération actuelle ou dans les  $k$  dernières générations.

Dans de nombreux problèmes, l'objectif est exprimé sous forme de minimisation d'une fonction  $g$ ,

$$\min_{x \in E} g(x) \quad 3.8$$

La dualité entre la minimisation de coût et la maximisation du profit est bien connue. Pour transformer un problème de minimisation en un problème de maximisation, on multiplie tout simplement la fonction de coût par  $-1$ . Avec les algorithmes génétiques, cette opération seule est insuffisante car la fonction ainsi obtenue n'est pas nécessairement positive sur tout son domaine. Il est aisé de montrer par exemple que l'équation (3.8) est équivalente à l'équation (3.9),

$$\begin{aligned} & \max_{x \in E} h(x) \\ \text{avec} \quad & h(x) = G_{\max} - g(x) \end{aligned} \quad 3.9$$

où  $G_{\max} \geq \max_{x \in E} g(x)$  pour éviter que la nouvelle fonction prenne des valeurs négatives sur son domaine de définition. Si  $G_{\max}$  ne peut déterminer rigoureusement, on modifiera la fonction  $h$  suivant,

$$h(x) = \begin{cases} G_{\max} - g(x) & \text{si } G_{\max} \geq g(x) \\ 0 & \text{sinon} \end{cases} \quad 3.10$$

Il y a de nombreuses façons de choisir le coefficient  $G_{\max}$ .  $G_{\max}$  peut être pris comme coefficient d'entrée, ou bien on peut lui affecter la plus grande valeur de  $g$  observée, ou la plus grande valeur de  $g$  dans la population actuelle, ou la plus grande dans les  $k$  dernières générations.

Notons que le choix de la fonction  $h$  n'est pas unique. En effet, toute composition de la fonction  $g$  par une fonction quelconque décroissante et monotone sur le domaine  $E$ , conduisant à un problème de maximisation équivalent à l'équation (3.8). On rencontre notamment dans la littérature la fonction de transformation  $h$  suivante :

$$h(x) = \frac{1}{1 + g(x)} \quad 3.11$$

4. Des opérateurs permettant de diversifier la population au cours des générations et d'explorer l'espace d'état. L'opérateur de croisement recompose les gènes d'individus existant dans la population, l'opérateur de mutation a pour but de garantir l'exploration de l'espace d'état.

5. Des paramètres de dimensionnement :

taille de la population, nombre total de générations ou critère d'arrêt, probabilités d'application des opérateurs de croisement et de mutation.

Le principe général de fonctionnement d'un algorithme génétique est représenté dans le tableau suivant:

<b>Fonctionnement général</b>	
1.	Création de la population initiale.
2.	Evaluation de chacun des chromosomes de la population initiale.
3.	Sélection et regroupement des chromosomes par paires.
4.	Application du croisement et de la mutation à chacune des paires.
5.	Evaluation des nouveaux chromosomes et insertion dans la population suivante.
6.	Si le critère d'arrêt est atteint, l'algorithme génétique s'arrête et rend le meilleur chromosome produit; sinon, l'algorithme retourne à l'étape 3.

**Tableau 3.1. Fonctionnement général d'un algorithme génétique**

### **3.5. Les opérateurs génétiques**

Les mécanismes d'un algorithme génétique de base sont étonnamment simples, et ne mettent en jeu rien de plus compliqué que des copies de chaînes et des échanges de morceaux de chaînes. L'AG génère de nouveaux individus de telle sorte qu'ils soient plus performants que leurs prédécesseurs.

Les opérateurs génétiques définissent la manière dont les chaînes binaires se recombinent et s'agencent pendant la phase de reproduction. Ils permettent ainsi la création de nouvelles chaînes. Les deux plus importants opérateurs sont : le crossover et la mutation

Un algorithme génétique simple composé des opérateurs suivants conduit généralement à des bons résultats pour un grand nombre d'applications :

- 1) Reproduction.
- 2) Crossover.
- 3) Mutation.

Ces outils sont basés sur des processus aléatoires. La reproduction est une opération dans laquelle la probabilité de survie d'un individu est liée à la valeur de la fonction coût qui lui est associée. Le croisement et la mutation permettent d'introduire de manière aléatoire de nouvelles informations au niveau des individus.

### 3.5.1. Reproduction

Les mécanismes de reproduction dans les algorithmes génétiques jouent un rôle très important d'une part pour diriger les recherches vers les meilleurs individus, d'autre part pour maintenir la diversité des génotypes (qualifie la structure du chromosome) dans la population.

Cette opération est telle que la probabilité de survie d'un individu est liée à la valeur de la fonction coût qui lui est associée. Le résultat de ce processus est que les individus qui ont la meilleure *fitness* peuvent avoir une ou plusieurs copies dans la prochaine génération, alors que ceux avec une faible *fitness* peuvent n'avoir aucune.

La reproduction est un procédé dans lequel chaque chaîne est copiée en fonction des valeurs de la fonction à optimiser  $f$  (fonction d'adaptation). Intuitivement, cette fonction  $f$  peut être envisagée comme une mesure de profit, d'utilité ou de qualité, que l'on souhaite maximiser. Copier des chaînes en fonction des valeurs de leur fonction d'adaptation revient à donner aux chaînes dont la valeur est plus grande une probabilité plus élevée de contribuer à la génération suivante, en créant au moins un descendant.

L'opérateur de reproduction peut être mis en œuvre sous forme algorithmique de différentes façons. La méthode de base, appelée *roue de loterie biaisée* attribue à chaque individu  $vi$  une probabilité de survie  $pi$  proportionnelle à son adaptation dans la population :  $pi = \text{adaptation } vi / \text{SOM adaptation } vi$ .

Ou avec une autre écriture :

$$P_{select} = \frac{f_j}{\sum f_j} \quad (f_j \text{ la valeur de la fonction coût associée au } j^{\text{ème}} \text{ individu}).$$

La sélection pouvant être faite en utilisant le rapport  $f_j / f_s$  ( $f_j$  la valeur de la fonction d'aptitude associée avec le  $j^{\text{ème}}$  individu et  $f_s$  la somme des valeurs de cette fonction) pour réaliser une roulette pondérée où chaque individu occupe une surface proportionnelle au rapport précédent. Des tirages aléatoires sur cette roulette donneront les chaînes qui participeront à la prochaine population. De cette façon, les chaînes bien adaptées ont un plus grand nombre de descendants dans les générations suivantes.

### 3.5.2. Le crossover

Le crossover se définit comme suit (figure (3.2)) : soient deux chaînes binaires  $V_1$  et  $V_2$  juxtaposées, le crossover les coupe en un point  $k$ , représentant une position ou un emplacement sur la chaîne, choisi aléatoirement entre 1 et la longueur de la chaîne moins 1 ( $k \in [1, l-1]$ ) et produit deux nouvelles chaînes  $V'_1$  et  $V'_2$  après avoir échangé les parties coupées. Les deux chaînes  $V_1$  et  $V_2$  participant à cette opération sont assimilées aux parents, et les deux chaînes résultantes aux descendants. Il faut également noter qu'une partie seulement de la population participe à cette opération : chaque individu se voit attribuer une même probabilité  $p_c$  de participer à un crossover, et c'est un tirage aléatoire qui détermine sa participation effective à cette opération. Ce type de crossover est le croisement à un point.

On rencontre aussi le croisement en deux points : cette technique est similaire à la précédente, dans ce cas deux positions sont sélectionnées et les sous chaînes entre ces deux positions sont échangées. (voir figure (3.3))

Le croisement aléatoire combine deux chromosomes selon une chaîne binaire aléatoire. A chaque position, les bits correspondants des parents sont échangés si la chaîne aléatoire contient un 1 à cette position. Si le bit aléatoire est 0, il n'y a pas d'échange. (voir figure (3.3))

### 3.5.3. La mutation

La mutation joue un rôle secondaire dans la mise en œuvre des algorithmes génétiques. Elle est cependant nécessaire parce que, bien que la reproduction et le croisement explorent et recombinent efficacement les notions existantes, ils peuvent parfois devenir trop zélés et perdre de la matière génétique potentielle utile (des 1 ou des 0 à des endroits précis). Dans les systèmes génétiques artificiels, l'opérateur de mutation protège contre de telles pertes irréparables. Dans un AG simple, la mutation est la

modification aléatoire occasionnelle (de faible probabilité) de la valeur d'un caractère de la chaîne.

Chaque bit de l'ensemble des chaînes de la population a une probabilité  $p_m$  de subir une mutation à chaque génération. Cet opérateur est appliqué par un tirage aléatoire sur chacun des digits de la population. (figure (3.2))

Pour ne pas altérer les performances de l'AG, *Goldberg* conseille de prendre une fréquence de mutation tous les 1000 bits. Certains chercheurs préconisent de prendre la valeur suivante :

$$p_m = \frac{1}{l} \quad 3.12$$

où  $l$  est la longueur du chromosome. D'autres études ont conduit à une formule empirique qui exprime le taux optimal de mutation en terme de la longueur  $l$  du chromosome et de la taille  $N_{pop}$  de la population,

$$p_m = \frac{1}{N_{pop} \sqrt{l}} \quad 3.13$$

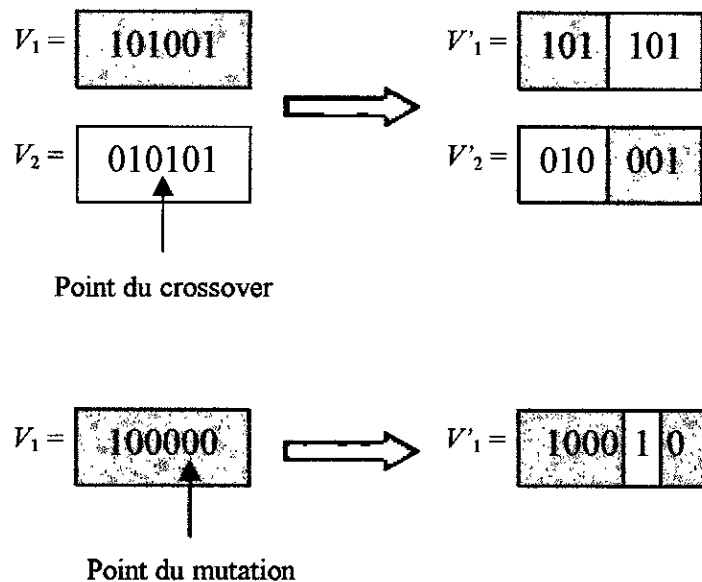


Figure 3.2. Opérateurs génétiques de Crossover et de Mutation

### 3.6. Les paramètres d'un algorithme génétique

Un *AG* est traditionnellement contrôlé par quatre paramètres :

◆ **La fonction d'évaluation :**

qui détermine la probabilité de sélection et de reproduction d'un individu.

◆ **La taille de la population :**

qui détermine le nombre d'individus dont l'*AG* dispose pour produire de bonnes solutions. Une taille de population trop faible peut empêcher l'*AG* de traiter certains individus ou peut encore limiter sa recherche si la fonction à optimiser (*fitness*) possède de nombreux *maxima* locaux.

◆ **La probabilité de mutation :**

que subit chaque allèle lors de la reproduction.

◆ **La probabilité de crossover :**

qui détermine la fréquence à laquelle les hybridations entre individus vont avoir lieu.

Ces deux derniers paramètres sont corrélés. Alors que la mutation permet l'apparition de nouveaux individus par la modification d'un allèle, le crossover diffuse les individus existants lors du renouvellement de la population. Cette balance entre l'exploration par mutation et l'exploitation par crossover doit être soigneusement respectée car un taux de mutation trop élevé entraîne la destruction d'individus avant qu'ils n'aient eu la chance d'être assemblés par crossover afin de former des structures valables. De l'autre côté, si le taux de crossover est excessif, la population sera uniformisée trop rapidement et la population convergera alors prématurément vers un type d'individu probablement sous-optimal.

Parent 1	00110100   1110
Parent 2	01110110   0101

---

Enfant 1	00110100   0101
Enfant 2	01110110   1110

#### Croisement à un point $k = 8$

Parent 1	00   101101   1110
Parent 2	01   110110   0101

---

Enfant 1	00   110110   1110
Enfant 2	01   101101   0101

#### Croisement en deux points

Parent 1	001011011110
Parent 2	011101100101

---

Chaîne aléatoire	110011110010
------------------	--------------

---

Enfant 1	011001101100
Enfant 2	001111010111

#### Croisement aléatoire

Figure 3.3.

### 3.7. Les étapes d'un algorithme génétique

Il est maintenant possible de définir les étapes d'un algorithme génétique. Après constitution d'une population initiale  $P(0)$  par tirage aléatoire, les étapes suivantes : évaluation, sélection et recombinaison (crossover + mutation) sont appliquées de manière itérative sur  $P(t)$  jusqu'à satisfaction de la condition de fin.  $P(t)$  étant la population à la génération  $t$ .

Un algorithme génétique ne nécessite aucune connaissance du problème : on peut représenter celui-ci par une boîte noire comportant des entrées (les variables) et des sorties (les fonctions objectif). L'algorithme ne fait que manipuler les entrées, lire les sorties, manipuler à nouveau les entrées de façon à améliorer les sorties, etc.

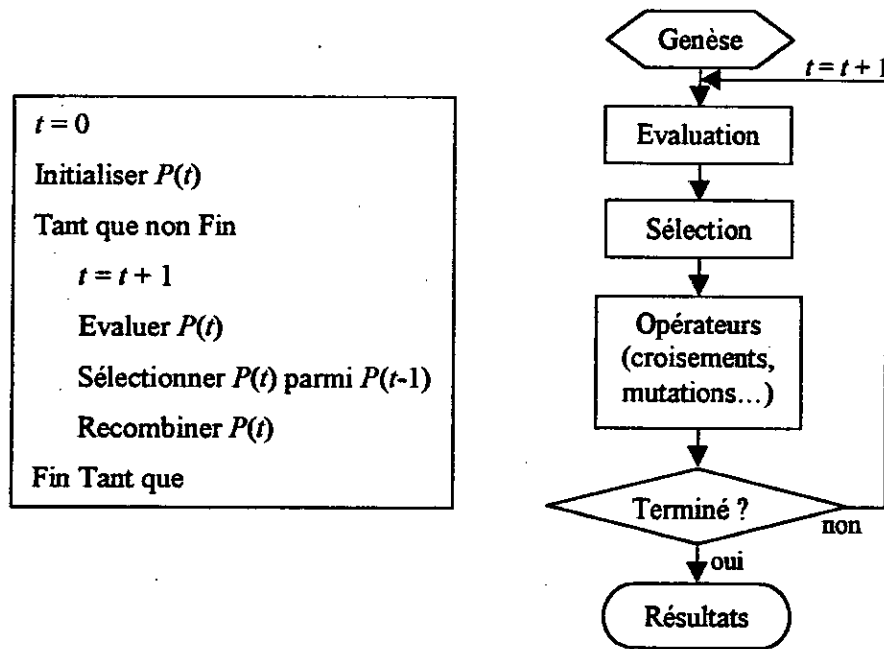


Figure 3.4. Déroulement d'un algorithme génétique

### 3.8. Critères de convergence des AGs

Un aspect important des algorithmes génétiques concerne le choix du critère de convergence. Quand peut-on considérer que l'algorithme a convergé vers l'optimum global ?

Le problème de convergence est généralement éludé en imposant un nombre maximal de générations  $t_{max}$  et en arrêtant la recherche lorsque  $t = t_{max}$ . On estime alors que l'algorithme a convergé et que l'individu de plus forte performance dans la population  $P^{t_{max}}$  correspond à la solution recherchée.

Une méthode plus rigoureuse consiste à supposer que l'algorithme converge vers l'optimum lorsque l'aptitude d'une partie (ou de l'ensemble) de la population se rapproche de celle du meilleur individu. On peut considérer que cet événement se produit à la génération  $t$  pour laquelle,

$$1 - \frac{f_{moy}}{f_{max}} \leq \varepsilon \quad 3.14$$

où  $\varepsilon$  est la précision requise sur la convergence,  $f_{max}$  est l'aptitude du meilleur individu de la population à la génération  $t$ , et  $f_{moy}$  est la moyenne de l'aptitude calculée sur l'ensemble



de la population ou sur une partie correspondant à un pourcentage des représentants les plus performants.

On peut aussi supposer que l'algorithme a convergé lorsque le meilleur individu de la population n'évolue plus.

Ces critères ne sont pas absolument fiables dans la mesure que l'algorithme peut converger et se stabiliser autour d'une solution dans l'attente d'une mutation qui le dirigera vers une autre région plus intéressante. Le choix d'un critère idéal d'arrêt reste donc sans réponse [21].

Du fait des problèmes de convergence et de la nature stochastique de l'exploration génétique, il est habituel d'exécuter plusieurs fois le même algorithme sur le même problème. Un taux de convergence (taux de réussite) rend compte de son efficacité. On peut par exemple définir le taux de convergence comme le rapport du nombre de fois où l'algorithme a convergé vers la solution optimale sur le nombre total d'exécutions.

## 3.9. Le contrôle prédictif généralisé et les algorithmes génétiques

### 3.9.1. Position du problème

Dans le chapitre 2, on a montré la difficulté de choisir les paramètres de réglage pour concevoir le contrôle prédictif généralisé en vue d'avoir les meilleures performances possibles du système commandé. Il s'agit de :

$N_1$  : l'horizon minimal de prédiction.

$N_2$  : l'horizon maximal de prédiction.

$N_u$  : l'horizon de commande.

$\lambda$  : le coefficient de pondération.

Le problème est de trouver des valeurs optimales pour ces paramètres qui vérifient la minimisation de la fonction quadratique de coût définie par le contrôleur *GPC*.

Dans cette partie, on propose un algorithme de sélection des paramètres de conception optimaux d'un contrôleur de type *GPC*. Puisque les algorithmes génétiques sont des outils d'optimisation performants qui permettent de réaliser une exploration globale de l'espace des paramètres et ils sont capables d'obtenir des solutions quasi-optimales, on va les utiliser pour trouver des valeurs optimales pour les paramètres de réglage du *GPC*.

Au début, on s'intéresse à trouver des valeurs optimales pour l'horizon de prédiction maximal  $N_2$  et l'horizon de commande  $N_u$ . Donc, il est nécessaire de définir un critère approprié et de le minimiser par rapport à  $N_2$  et  $N_u$  qui constituent les paramètres principaux du *GPC*.

Le contrôle prédictif généralisé *GPC* minimise un critère quadratique de la forme :

$$J_{GPC} = \sum_{j=N_1}^{N_2} [y(t+j) - w(t+j)]^2 + \sum_{j=1}^{N_u} \lambda [\Delta u(t+j-1)]^2 \quad 3.15$$

pour l'algorithme proposé dans cette section (*GPC* + algorithme génétique), on va prendre un critère équivalent,

$$J_{GPCAG1} = \sum_{t=0}^t \left[ \sum_{j=N_1}^{N_2} [y(t+j) - w(t+j)]^2 + \sum_{j=1}^{N_u} \lambda [\Delta u(t+j-1)]^2 \right] \quad 3.16$$

$$J_{GPCAG2} = J_{GPCAG1} + N_2 + N_u$$

La première partie du critère  $J_{GPCAG2}$  représente la somme des valeurs du critère  $J_{GPC}$  de l'instant  $t = 0$  jusqu'à l'instant courant  $t$ . Les deux derniers termes sont ajoutés afin de maintenir les valeurs de  $N_2$  et  $N_u$  dans des limites raisonnables ; à des performances sensiblement équivalentes, il y a lieu de prendre les horizons  $N_2$  et  $N_u$  les plus faibles.

L'objectif est exprimé sous forme de minimisation d'un critère  $J_{GPCAG2}$ . Le passage d'un problème de minimisation à un problème de maximisation est obtenu par transformation du critère  $J_{GPCAG2}$  (la fonction à minimiser). La fonction de transformation *fitness* est donnée par l'équation suivante :

$$fitness = \frac{1}{J_{GPCAG2}} \quad 3.17$$

L'objectif maintenant est de maximiser la fonction *fitness* donnée par l'équation (3.17) qui déterminera les horizons de prédiction et de commande optimaux.

Le schéma de fonctionnement est représenté par la figure (3.5).

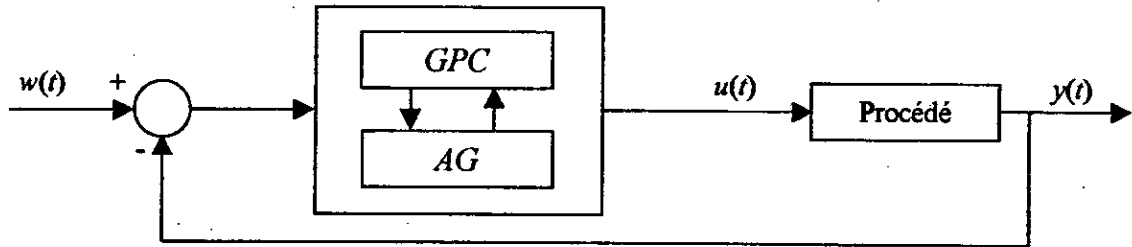


Figure 3.5. Schéma de fonctionnement

Le principe de programmation de l'algorithme proposé est le suivant :

#### Etape 1

A l'instant  $t = 0$ , on génère aléatoirement une population initiale de 20 individus, 20 couples entiers ( $N_2$ ,  $N_u$ ), qui représente la population initiale  $P(0)$ .

Après le décodage des paramètres, on calcule la fonction *fitness* pour chaque individu en évaluant les équations (3.15), (3.16) et (3.17).

#### Etape 2

A l'instant  $t$  et à partir de la population  $P(t-1)$ , on génère la nouvelle population  $P(t)$  en utilisant les opérateurs classiques des algorithmes génétiques : sélection, crossover et mutation. Et pour chaque individu de la nouvelle population  $P(t)$ , il est nécessaire d'évaluer la fonction d'aptitude en utilisant toujours les équations (3.15), (3.16) et (3.17).

A la fin, on retient l'individu ( $N_2^*$ ,  $N_u^*$ ) avec la meilleure *fitness* ( $J_{GPCAG1}$  minimal et  $J_{GPCAG2}$  minimal). Si la condition n'est pas vérifiée, c'est à dire le couple qui maximise la fonction *fitness* ne minimise pas les deux critères  $J_{GPCAG1}$  et  $J_{GPCAG2}$  en même temps, on retient donc le couple qui minimise le critère  $J_{GPCAG1}$ .

Puis une petite comparaison est faite pour le cas où les deux couples retenus aux instants  $t-1$  et  $t$  sont différents : en utilisant le meilleur couple obtenu à l'instant  $t-1$ , on évalue les équations (3.15), (3.16) et (3.17). Si on trouve un résultat mieux que le résultat obtenu avec le couple retenu à l'instant  $t$ , on garde l'ancien couple. Sinon on retient le nouveau couple.

#### Etape 3

Toujours à l'instant  $t$ , on calcule la commande  $u(t)$  qui minimise le critère du contrôleur GPC donné par l'équation (3.15) en utilisant les meilleures valeurs trouvées à l'étape 2. Cette commande est appliquée au système.

#### Etape 4

A l'instant  $t+1$ , on revient à l'étape 2 si le nombre d'itérations  $N$  n'est pas atteint.

On peut traduire ce fonctionnement par un organigramme général présenté par la figure (3.6).

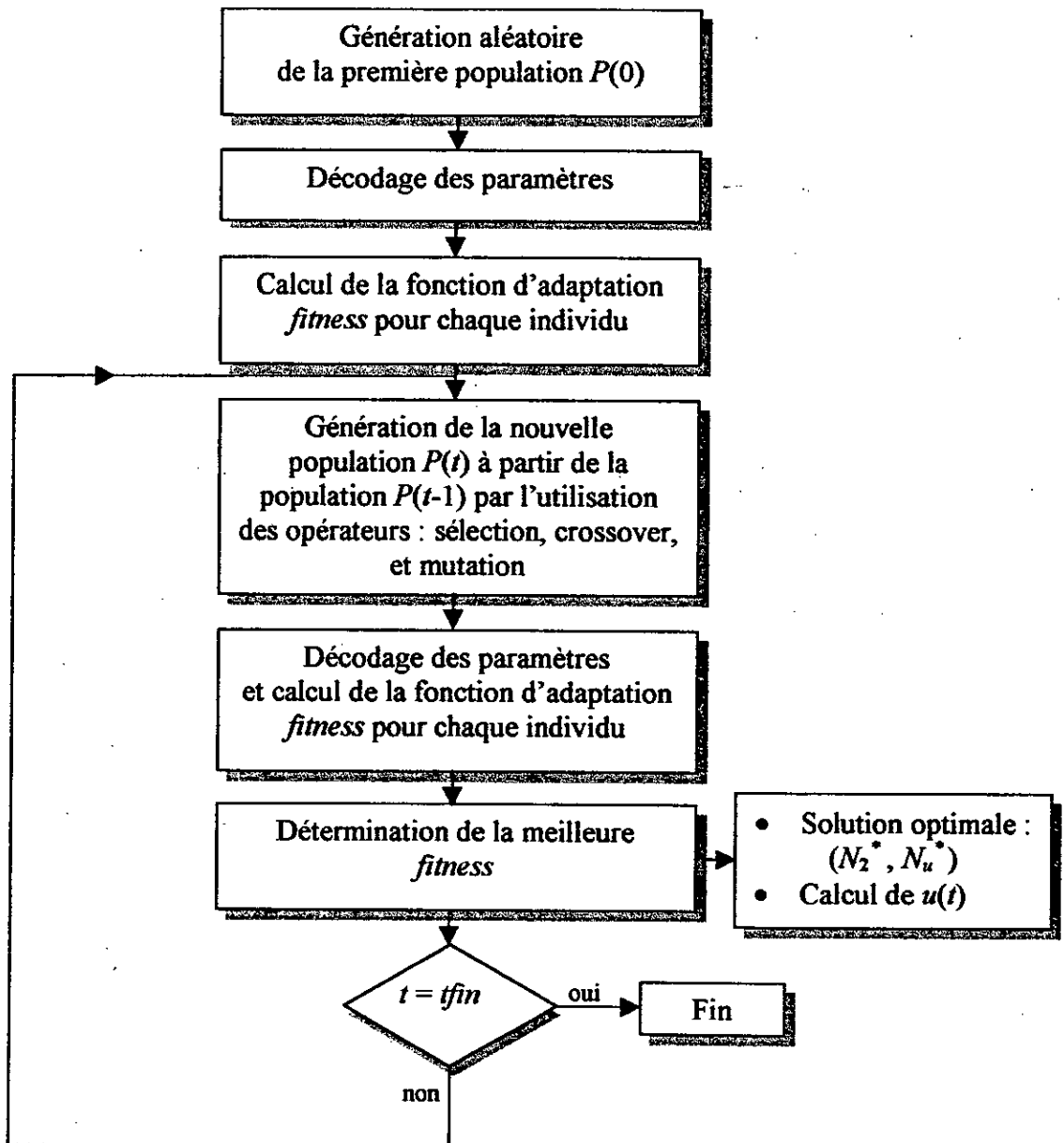


Figure 3.6. L'organigramme de l'algorithme *GPC\_AG*

### 3.9.2. Simulation 1

L'algorithme *GPC\_AG* pour l'estimation des paramètres de conception du contrôleur *GPC* avec l'utilisation des algorithmes génétiques pour l'optimisation de  $N_2$  et  $N_u$  permet à partir d'une population initiale générée aléatoirement, contient un certain nombre de

couples  $(N_2, N_u)$ , de détecter la meilleure *fitness*, le meilleur couple, pour que le contrôleur *GPC* aura les meilleures performances possibles.

Dans cet algorithme, les paramètres utilisés pour le *GPC* ne sont plus fixés par l'utilisateur mais sont les paramètres optimaux trouvés à chaque instant  $t$ .

On prend comme exemple le pendule inversé traité dans le chapitre 2. Donc, il s'agit d'asservir la position du chariot en  $x$  tout en veillant à maintenir la barre suspendue en position verticale. La consigne (une consigne de position  $x_c$ ) est nulle, on va vérifier la réponse du processus à des conditions initiales données par le vecteur suivant :

$$X_0 = [0.5 \ 0 \ 0.1 \ 0]^T$$

Pour vérifier la marche de l'algorithme (*GPC\_AG*), on va faire plusieurs tests sur le système en utilisant seulement le contrôleur *GPC* pour différentes valeurs de  $N_2$  et  $N_u$ , mais en assurant toujours la minimisation du critère (3.15) pour chaque couple  $(N_2, N_u)$ .

La somme des valeurs du critère *GPC* pour chaque couple sur l'horizon de l'exécution (on a pris  $N$  le nombre des échantillons égal à 100) est calculée. Les sommes obtenues pour le cas  $N_1 = 4$  et  $\lambda = 0.0001$  sont groupées dans le tableau (3.2) et celles obtenues pour le cas  $N_1 = 5$  et  $\lambda = 0.0001$  sont groupées dans le tableau (3.3).

➤ **Premier cas :  $N_1 = 4$  et  $\lambda = 0.0001$**

$N_u$	4	5	6	7	8	9	10	11	12	13	14	15	20	25
$N_2$														
5	4.9684	5.0365												
6	4.3495	4.2327	4.2204											
7	4.2800	3.9805	3.9246	3.9226										
8	4.2606	3.8391	3.7759	3.7790	3.7796									
9	4.2761	3.8001	3.7532	3.7600	3.7596	3.7593								
10	4.3262	3.8429	3.8065	3.8020	3.7924	3.7893	3.7891							
11	4.4063	3.9381	3.8965	3.8691	3.8461	3.8388	3.8378	3.8377						
12	4.5069	4.0593	3.9975	3.9413	3.9042	3.8932	3.8917	3.8917	3.8917					
13	4.6187	4.1887	4.0951	4.0079	3.9572	3.9431	3.9414	3.9413	3.9412	3.9412				
14	4.7358	4.3169	4.1831	4.0645	4.0009	3.9839	3.9819	3.9817	3.9816	3.9816	3.9816			
15	4.8561	4.4409	4.2605	4.1109	4.0348	4.0149	4.0124	4.0121	4.0119	4.0118	4.0118	4.0118		
20	5.5696	5.0518	5.0276	4.2824	4.1627	4.1185	4.1097	4.1075	4.1062	4.1054	4.1049	4.1046	4.1045	
25	6.6839	5.8231	5.7717	4.4690	4.5988	4.1912	4.2069	4.2519	4.1190	4.2162	4.1710	4.1134	5.2568	4.1125

Tableau 3.2.

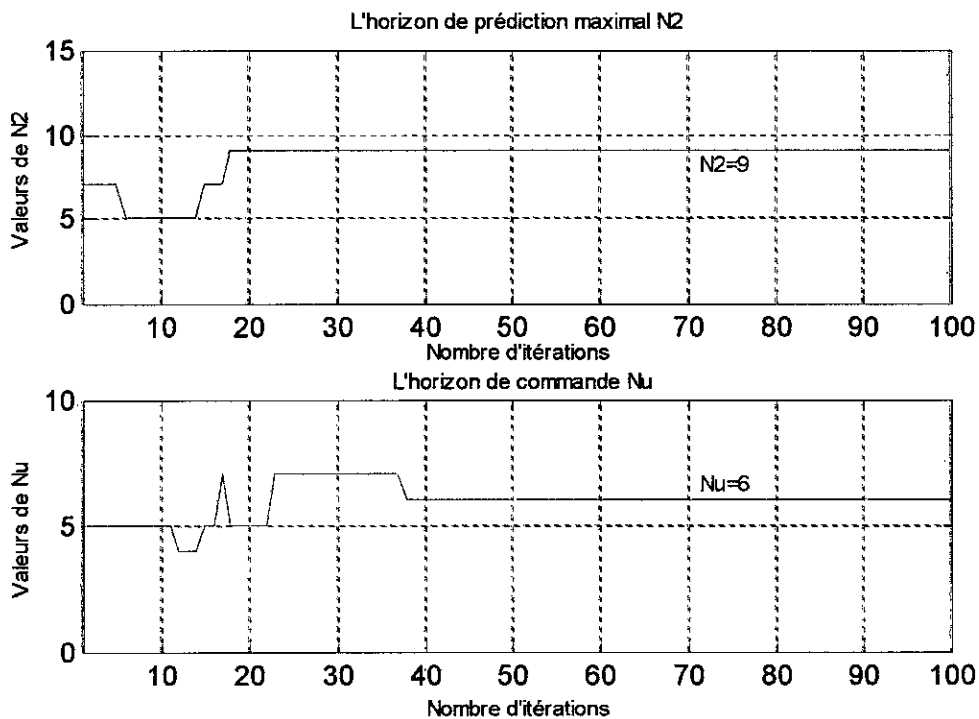
On remarque que la valeur minimale du critère est obtenue avec les valeurs optimales :

$$(N_2^*, N_u^*) = (9, 6)$$

Les paramètres utilisés pour l'algorithme *GPC\_AG* sont :

La taille de la population	:	$popsiz$ = 20	
Probabilité de croisement	:	$pcross$ = 0.6	
Probabilité de mutation	:	$pmutation$ = 0.033	
Le nombre des paramètres	:	$nparam$ = 2	
La longueur d'un paramètre	:	$lparam$ = 5	
La longueur de la chaîne	:	$lchrom$ = 10	
L'horizon de prédiction minimal	:	$N_1$ = 4	
Domaine de l'estimation de $N_2$	:	$[N_1 + 1, 30]$	
Domaine de l'estimation de $N_u$	:	$[N_1, 30]$	avec $N_u \leq N_2$ et $N_u \geq N_1$
Le coefficient de pondération	:	$\lambda$ = 0.0001	

Les résultats obtenus après lancement de l'algorithme *GPC\_AG* sont montrés par la figure suivante :



**Figure 3.7.** L'horizon de prédiction maximal  $N_2$  et l'horizon de commande  $N_u$

➤ **Deuxième cas :  $N_1 = 5$  et  $\lambda = 0.0001$**

$N_u$	5	6	7	8	9	10	11	12	13	14	15	20	25
$N_2$													
6	2.3401	2.3363											
7	2.3580	2.2477	2.2397										
8	2.3246	2.1350	2.1174	2.1173									
9	2.3117	2.1213	2.1148	2.1146	2.1143								
10	2.3581	2.2007	2.1970	2.1851	2.1790	2.1785							
11	2.4527	2.3281	2.3100	2.2764	2.2603	2.2573	2.2572						
12	2.5737	2.4687	2.4235	2.3654	2.3396	2.3345	2.3341	2.3341					
13	2.7035	2.6034	2.5243	2.4417	2.4077	2.4013	2.4008	2.4008	2.4008				
14	2.8326	2.7243	2.6083	2.5025	2.4610	2.4538	2.4532	2.4531	2.4530	2.4530			
15	2.9578	2.8306	2.6766	2.5485	2.5002	2.4920	2.4913	2.4910	2.4909	2.4909	2.4909		
20	3.5836	3.5593	2.9700	2.7848	2.6860	2.6627	2.6591	2.6579	2.6572	2.6569	2.6566	2.6565	
25	4.4050	4.2686	3.2216	3.4692	2.7154	2.9210	2.7388	2.6698	2.7525	2.8311	2.9460	3.1619	4.4222

**Tableau 3.3.**

La valeur minimale du critère est obtenue avec les valeurs optimales :

$$(N_2^*, N_u^*) = (9, 9)$$

Les paramètres utilisés pour l'algorithme *GPC\_AG* sont :

- La taille de la population :  $popsiz = 20$
- Probabilité de croisement :  $pcross = 0.6$
- Probabilité de mutation :  $pmutation = 0.033$
- Le nombre des paramètres :  $nparam = 2$
- La longueur d'un paramètre :  $lparam = 5$
- La longueur de la chaîne :  $lchrom = 10$
- L'horizon de prédiction minimal :  $N_1 = 5$
- Domaine de l'estimation de  $N_2$  :  $[N_1 + 1, 30]$
- Domaine de l'estimation de  $N_u$  :  $[N_1, 30]$  avec  $N_u \leq N_2$  et  $N_u \geq N_1$
- Le coefficient de pondération :  $\lambda = 0.0001$

Les résultats obtenus après lancement de l'algorithme *GPC\_AG* sont montrés par la figure (3.8).

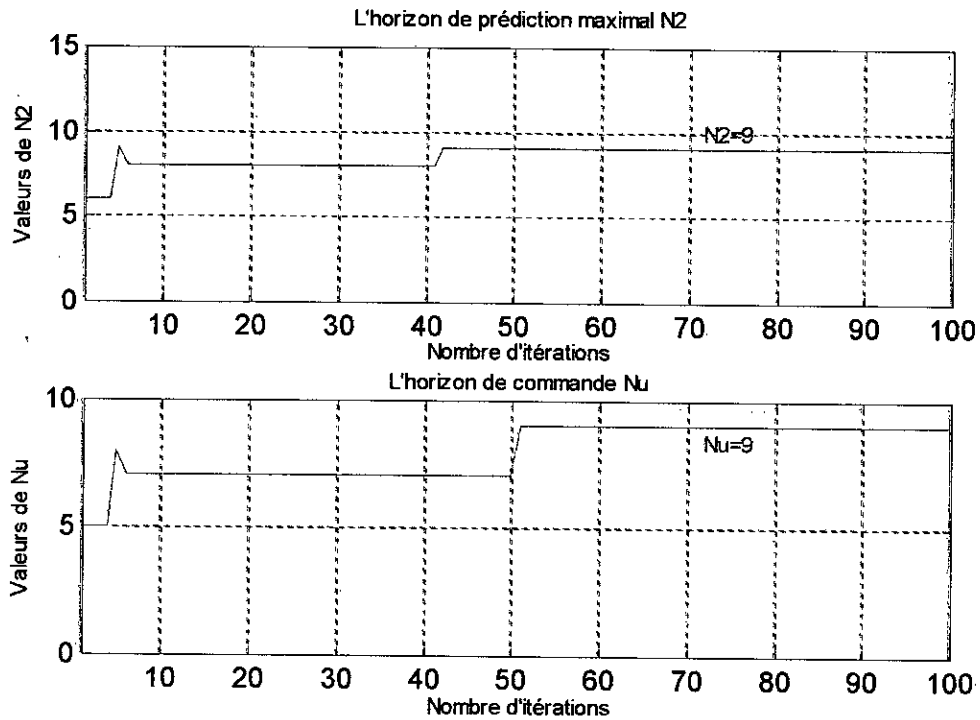


Figure 3.8. L'horizon de prédiction maximal  $N_2$  et l'horizon de commande  $N_u$

La réponse du processus pour les deux cas est montrée par les figures suivantes :

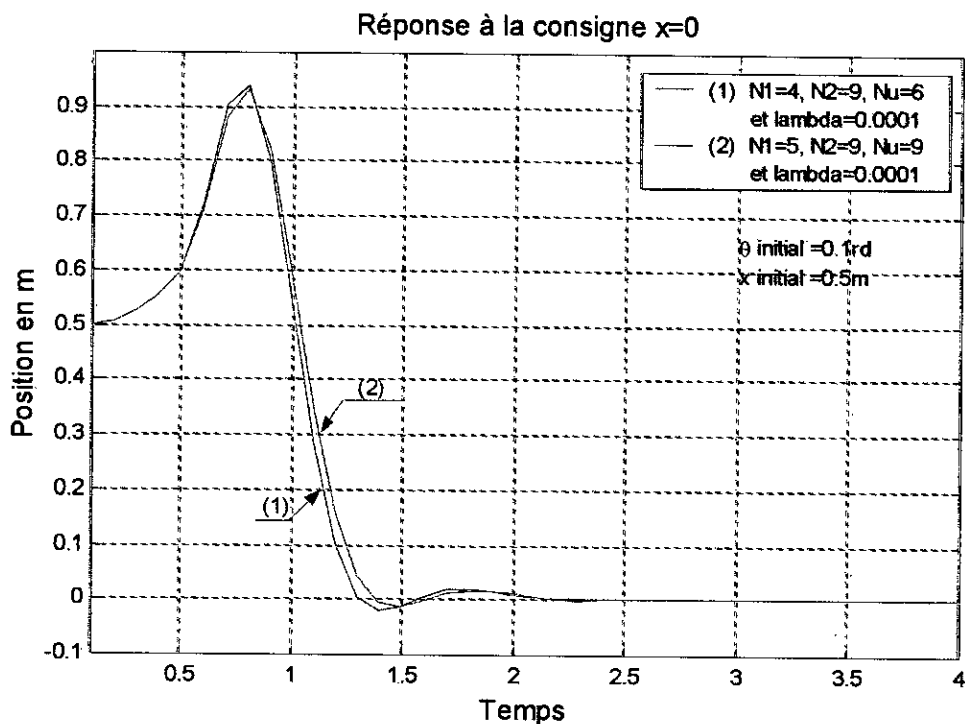


Figure 3.9. Réponse  $x(t)$  à la consigne  $x = 0$



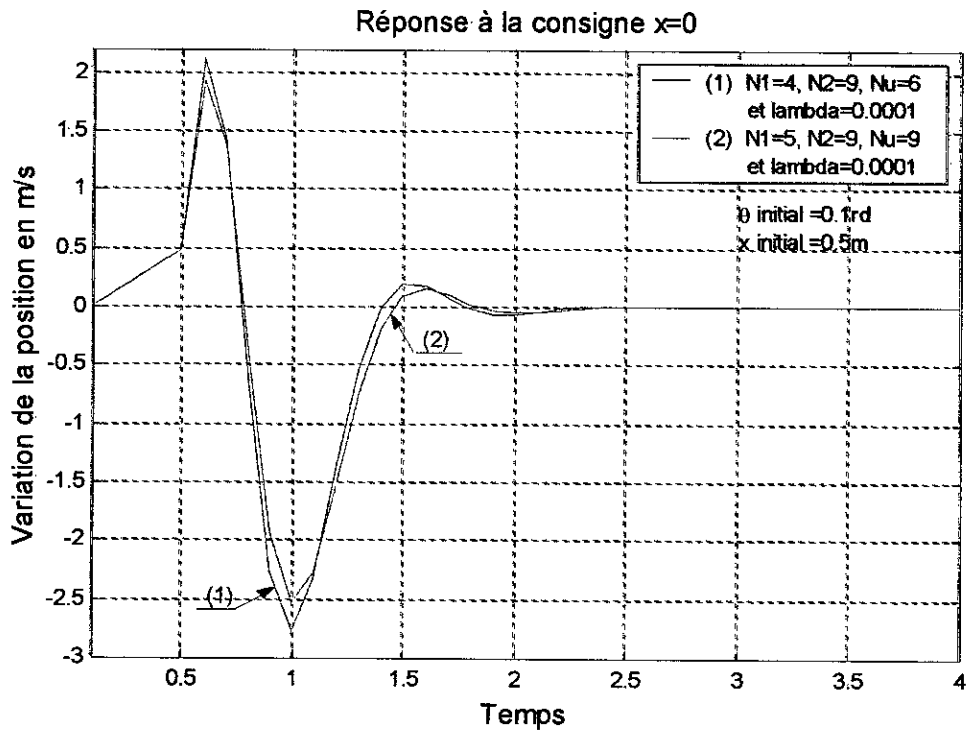


Figure 3.10. Réponse  $x'(t)$  à la consigne  $x = 0$

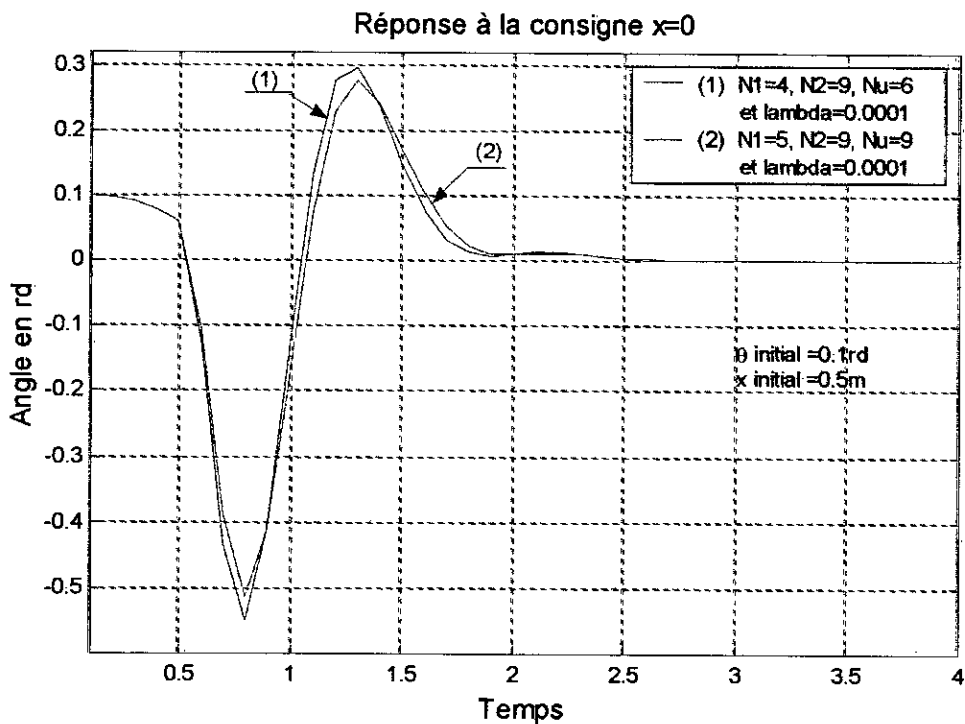


Figure 3.11. Réponse  $\theta(t)$  à la consigne  $x = 0$

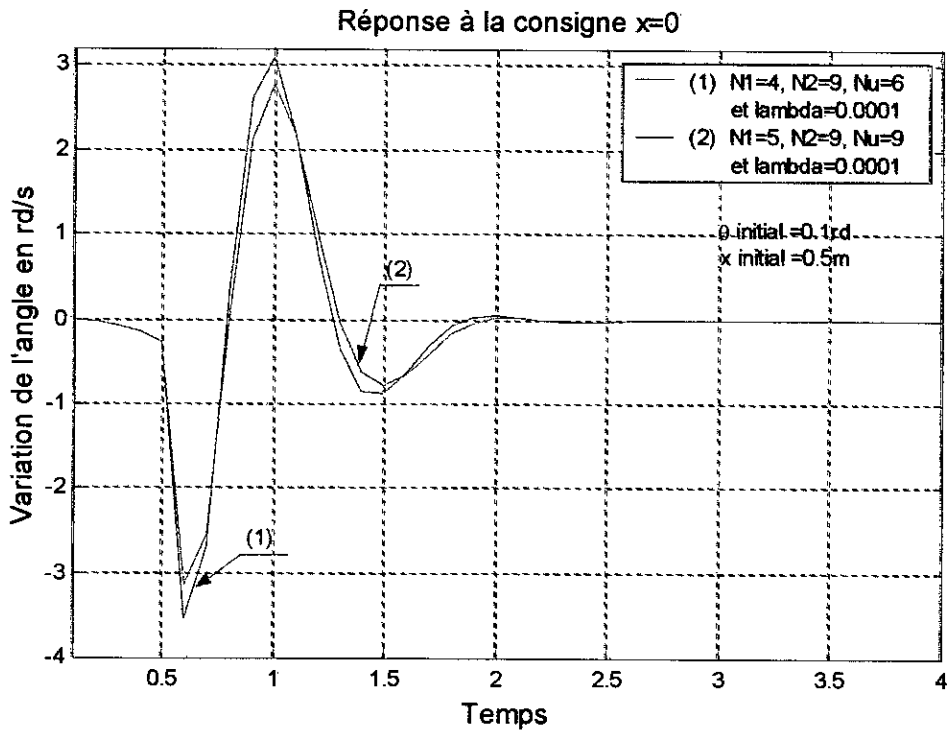


Figure 3.12. Réponse  $\theta'(t)$  à la consigne  $x = 0$

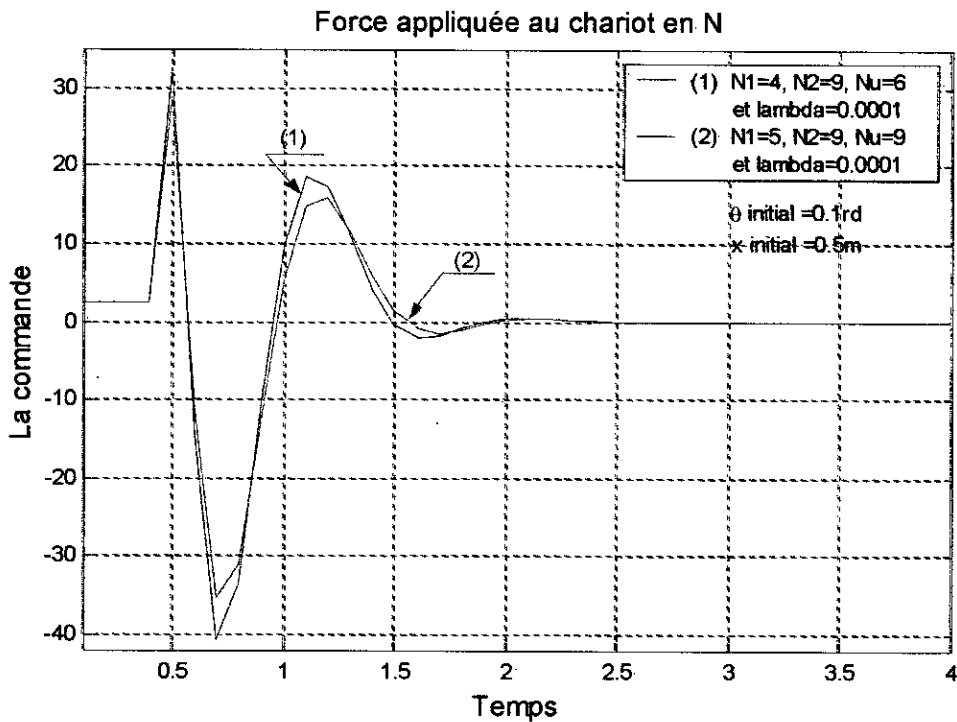


Figure 3.13. Force appliquée au chariot en  $N$

### 3.9.3. Simulation 2

Dans cette simulation, le principe de programmation est le même. Seulement, qu'on cherche maintenant à optimiser les valeurs de l'horizon de prédiction minimal  $N_1$  et l'horizon de commande  $N_u$  avec la valeur de l'horizon de prédiction maximal  $N_2$  égale à 9.

Le critère à minimiser a la forme suivante :

$$J_{GPCAG2} = J_{GPCAG1} + N_u + N_1$$

$$\text{avec } J_{GPCAG1} = \sum_{t=0}^t \left[ \sum_{j=N_1}^{N_2} [y(t+j) - w(t+j)]^2 + \sum_{j=1}^{N_u} \lambda [\Delta u(t+j-1)]^2 \right] \quad 3.18$$

La fonction *fitness* est donnée par l'équation (3.17).

Et pour vérifier aussi la marche de l'algorithme  $GPC\_AG$ , on va faire plusieurs tests sur le système en utilisant seulement le contrôleur  $GPC$  pour différentes valeurs de  $N_u$  et  $N_1$ . Et pour chaque couple  $(N_u, N_1)$ , il faut assurer la minimisation du critère (3.15). La somme des valeurs du critère pour chaque couple sur l'horizon de l'exécution ( $N = 100$ ) est calculée. Les sommes obtenues sont groupées dans le tableau (3.4).

$N_u$	3	4	5	6	7	8	9
$N_1$							
3	7.9037	6.4284	6.1874	6.1873	6.1916	6.1908	6.1906
4		4.2761	3.8001	3.7532	3.7600	3.7596	3.7593
5			2.3117	2.1213	2.1148	2.1146	2.1143
6				1.1975	1.1308	1.1285	1.1285
7					0.6083	0.5862	0.5856
8						0.2867	0.2834

Tableau 3.4.

La valeur minimale du critère est obtenue avec les valeurs optimales :

$$(N_u^*, N_1^*) = (9, 8)$$

Les paramètres utilisés pour ce test sont :

La taille de la population	:	$popsiz = 20$
Probabilité de croisement	:	$pcross = 0.6$
Probabilité de mutation	:	$pmutation = 0.033$
Le nombre des paramètres	:	$nparam = 2$
La longueur d'un paramètre	:	$lparam = 5$

La longueur de la chaîne	:	$lchrom = 10$	
L'horizon de prédiction maximal	:	$N_2 = 9$	
Domaine de l'estimation de $N_1$	:	$[3, 30]$	avec $N_1 < N_2$
Domaine de l'estimation de $N_u$	:	$[3, 30]$	avec $N_u \leq N_2$ et $N_u \geq N_1$
Le coefficient de pondération	:	$\lambda = 0.0001$	

Les résultats obtenus après lancement de l'algorithme  $GPC\_AG$  sont montrés par la figure suivante :

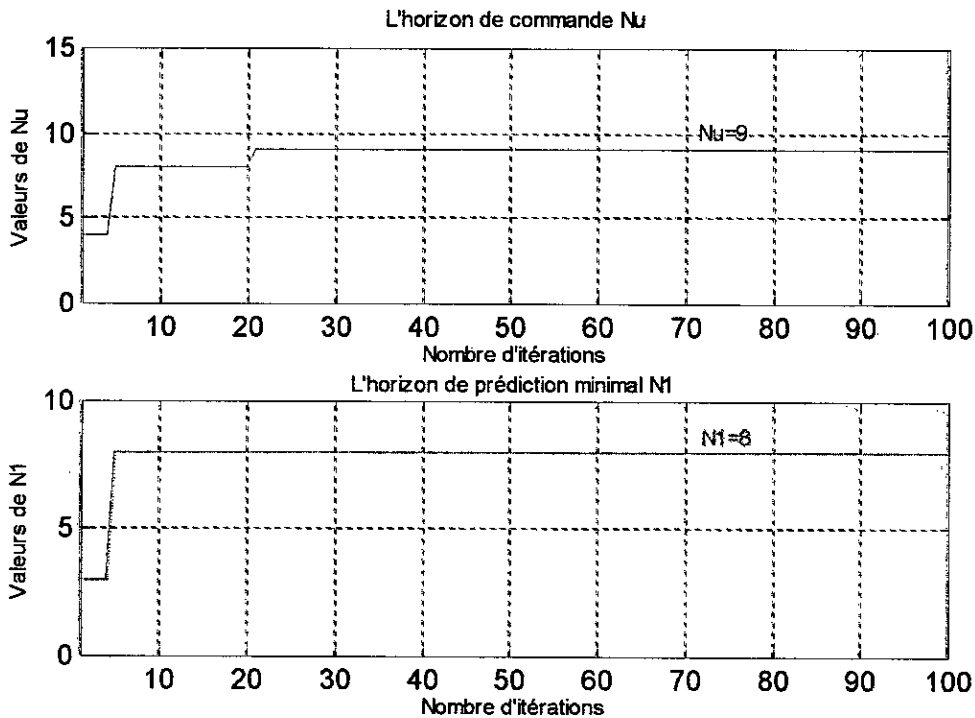


Figure 3.14. L'horizon de commande  $N_u$  et l'horizon de prédiction minimal  $N_1$

### 3.9.4. Simulation 3

On cherche maintenant à optimiser les valeurs de  $N_2$ ,  $N_u$  et  $N_1$ , c'est à dire de trouver le meilleur individu  $(N_2, N_u, N_1)$  qui nous donne les meilleures performances.

Le critère à minimiser maintenant est,

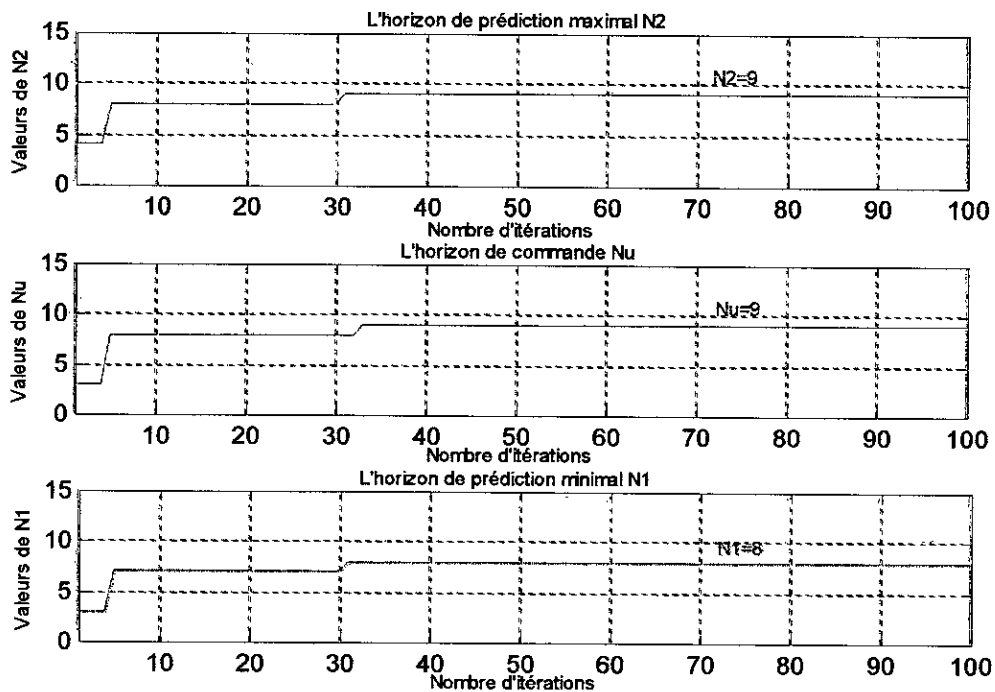
$$J_{GPCAG2} = J_{GPCAG1} + N_2 + N_u + N_1$$

$$\text{avec } J_{GPCAG1} = \sum_{t=0}^t \left[ \sum_{j=N_1}^{N_2} [y(t+j) - w(t+j)]^2 + \sum_{j=1}^{N_u} \lambda [\Delta u(t+j-1)]^2 \right] \quad 3.19$$

Les paramètres utilisés pour ce test sont :

La taille de la population	:	$popsiz$	$= 30$
Probabilité de croisement	:	$pcross$	$= 0.6$
Probabilité de mutation	:	$pmutation$	$= 0.033$
Le nombre des paramètres	:	$nparam$	$= 3$
La longueur d'un paramètre	:	$lparam$	$= 5$
La longueur de la chaîne	:	$lchrom$	$= 15$
Domaine de l'estimation de $N_2$	:	$[4, 30]$	
Domaine de l'estimation de $N_1$	:	$[3, 8]$	avec $N_1 < N_2$
Domaine de l'estimation de $N_u$	:	$[3, 30]$	avec $N_u \leq N_2$ et $N_u \geq N_1$
Le coefficient de pondération	:	$\lambda$	$= 0.0001$

Les résultats obtenus après lancement de l'algorithme  $GPC\_AG$  sont montrés par la figure suivante :



**Figure 3.15. L'horizon de prédiction maximal  $N_2$ , l'horizon de commande  $N_u$  et l'horizon de prédiction minimal  $N_1$**

On peut tester si l'algorithme s'adapte pour donner les meilleures valeurs. Donc, pour  $N_1 = 8$ , les résultats obtenus sont groupés dans le tableau (3.5). Chaque case représente la somme des valeurs de critère  $GPC$  sur l'horizon de l'exécution ( $N = 100$ ) pour chaque couple  $(N_2, N_u)$ .

$N_u$	8	9	10	11	12	13	14	15	20	25
$N_2$										
9	0.2867	0.2834								
10	0.3538	0.3469	0.3468							
11	0.4423	0.4392	0.4385	0.4383						
12	0.0055	0.5471	0.5410	0.5384	0.5381					
13	0.6604	0.6460	0.6313	0.6251	0.6239	0.6239				
14	0.7554	0.7269	0.7037	0.6948	0.6929	0.6927	0.6927			
15	0.8334	0.7894	0.7586	0.7477	0.7456	0.7453	0.7453	0.7453		
20	1.1136	0.9885	0.9115	0.8889	0.8821	0.8811	0.8806	0.8802	0.8800	
25	1.2785	1.1140	1.1127	1.0358	0.9247	1.2263	1.1516	1.3565	1.9371	0.8936

Tableau 3.5.

On remarque que la valeur minimale du critère est obtenue avec les valeurs optimales :

$$(N_2^*, N_u^*, N_1^*) = (9, 9, 8)$$

On remarque aussi que ces valeurs optimales sont les mêmes trouvées après l'application de l'algorithme  $GPC\_AG$ .

La réponse du système est montrée par les figures suivantes :

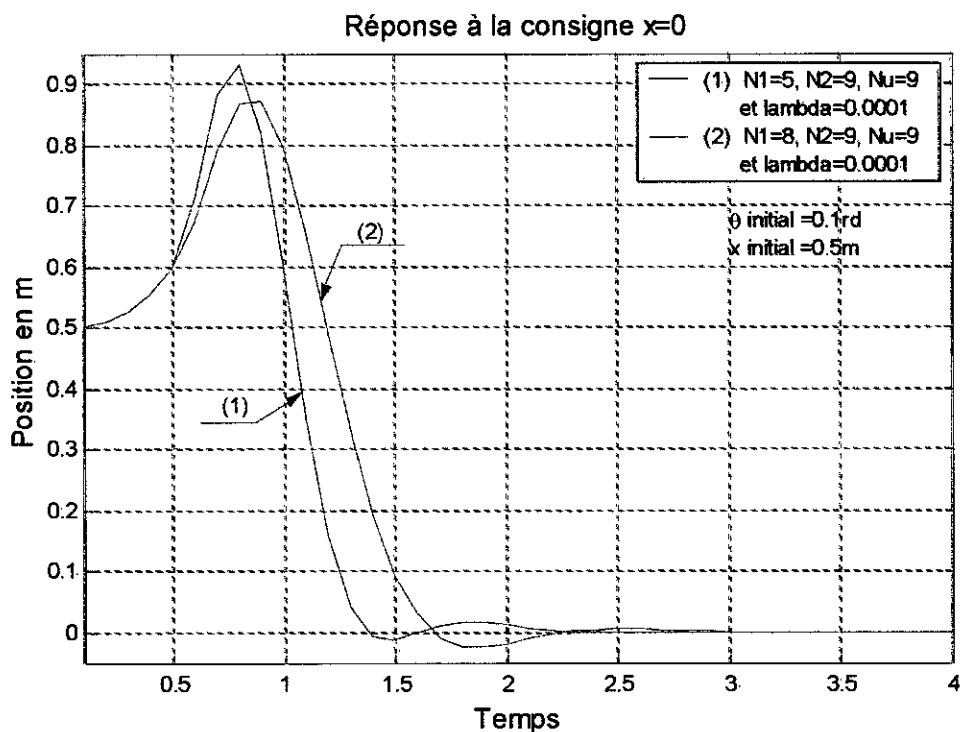


Figure 3.16. Réponse  $x(t)$  à la consigne  $x = 0$

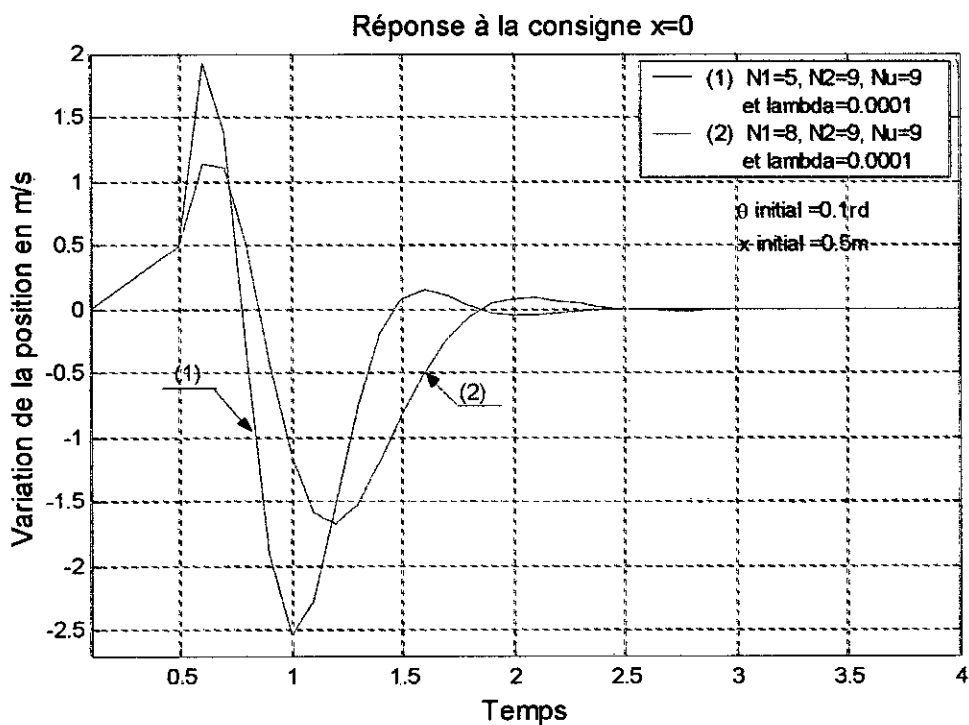


Figure 3.17. Réponse  $x'(t)$  à la consigne  $x = 0$

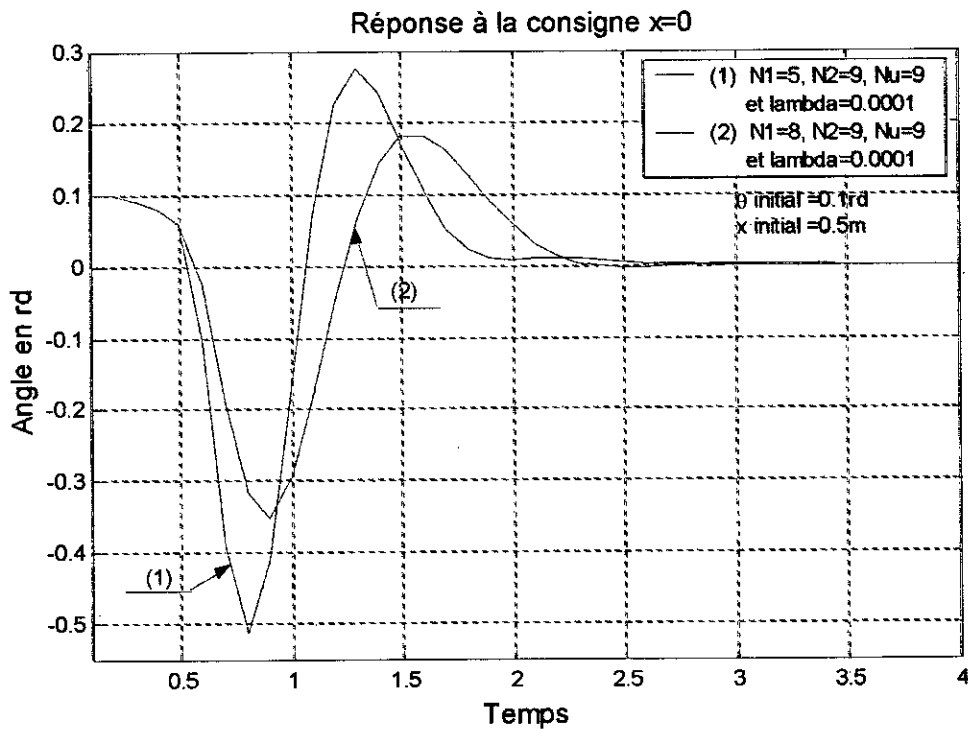


Figure 3.18. Réponse  $\theta(t)$  à la consigne  $x = 0$

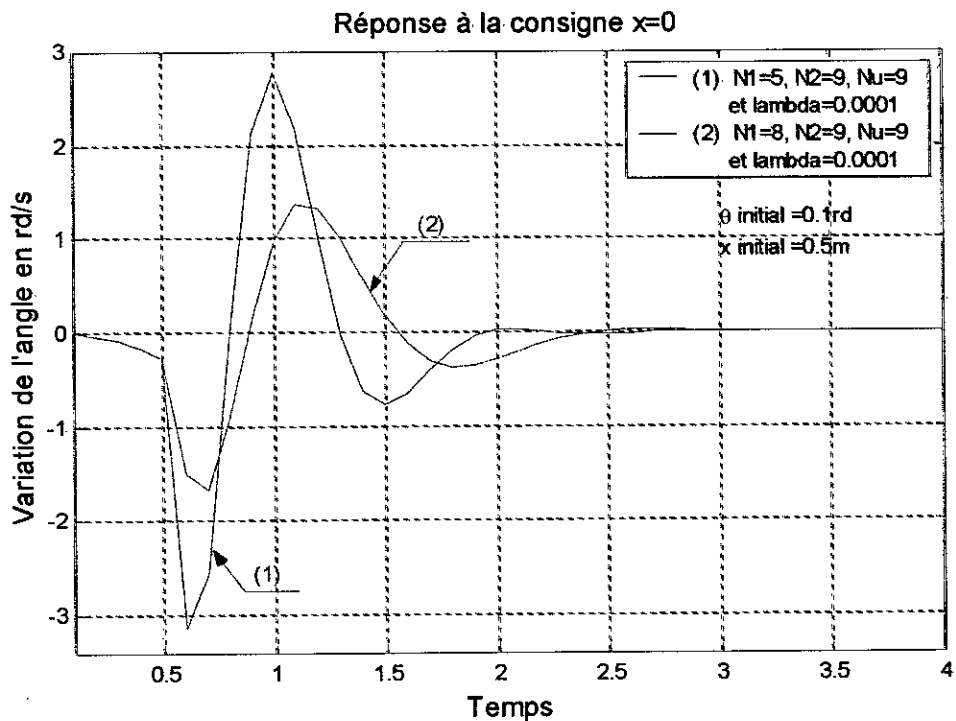
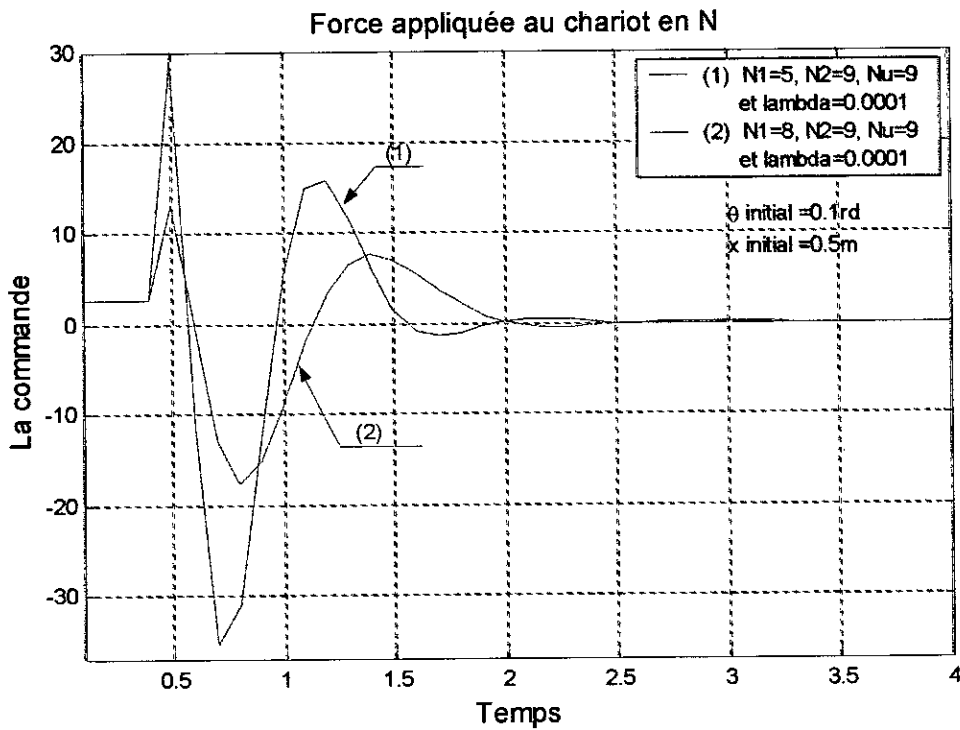


Figure 3.19. Réponse  $\theta'(t)$  à la consigne  $x = 0$





**Figure 3.20.** Force appliquée au chariot en  $N$

On va tester maintenant la réponse du système pour des valeurs grandes de l'horizon de prédiction minimal  $N_1$ . Donc, on applique l'algorithme *GPC<sub>AG</sub>* pour optimiser les valeurs de  $N_2$  et  $N_u$  (voir simulation 1). Les résultats obtenus avec  $N_1 = 11$  sont montrés par la figure (3.21) et avec  $N_1 = 15$  sont montrés par la figure (3.22).

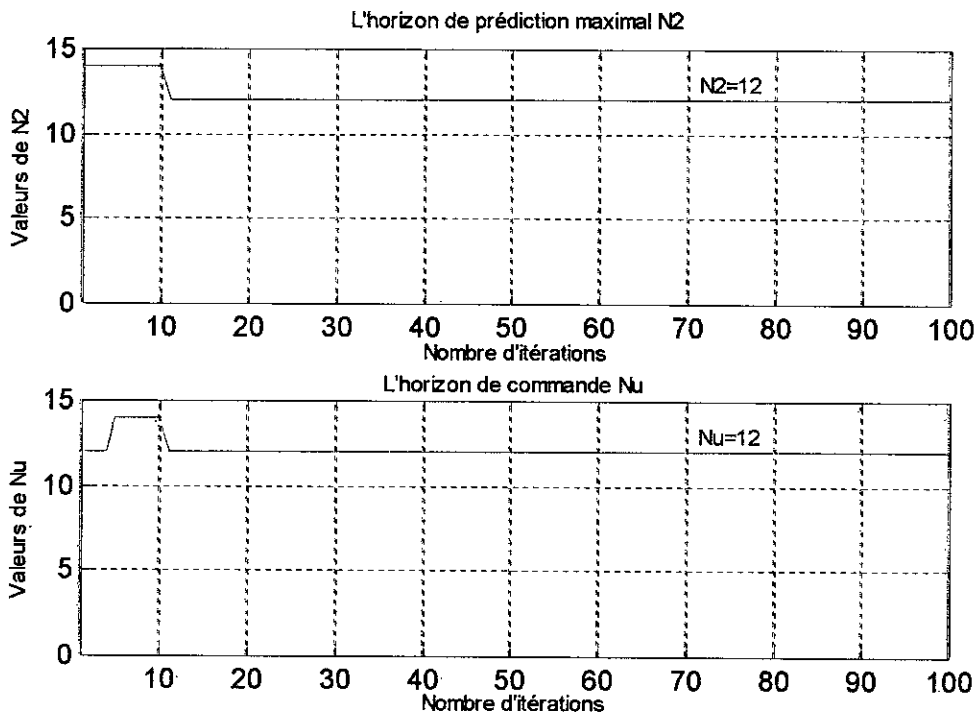


Figure 3.21. L'horizon de prédiction maximal  $N_2$  et l'horizon de commande  $N_u$

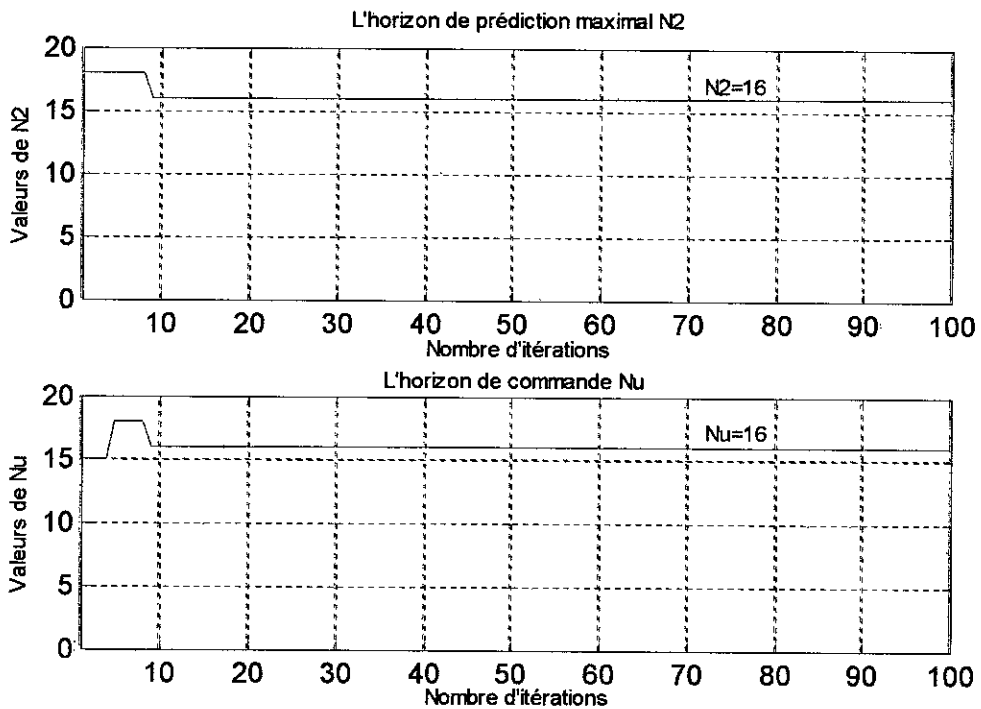


Figure 3.22. L'horizon de prédiction maximal  $N_2$  et l'horizon de commande  $N_u$

La réponse du système est montrée par les figures suivantes :

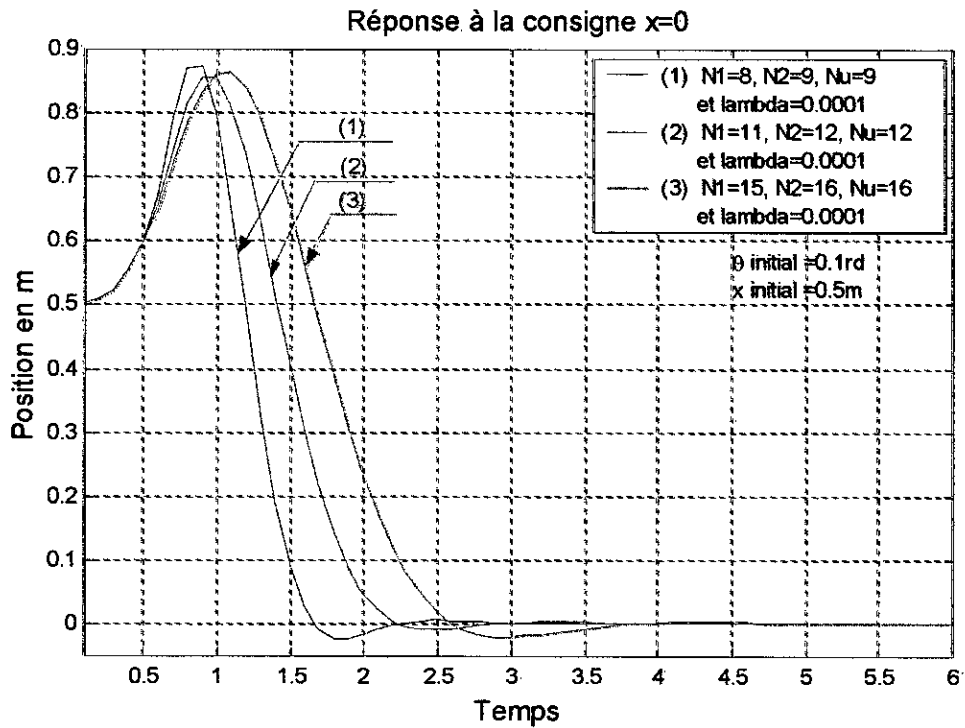


Figure 3.23. Réponse  $x(t)$  à la consigne  $x = 0$

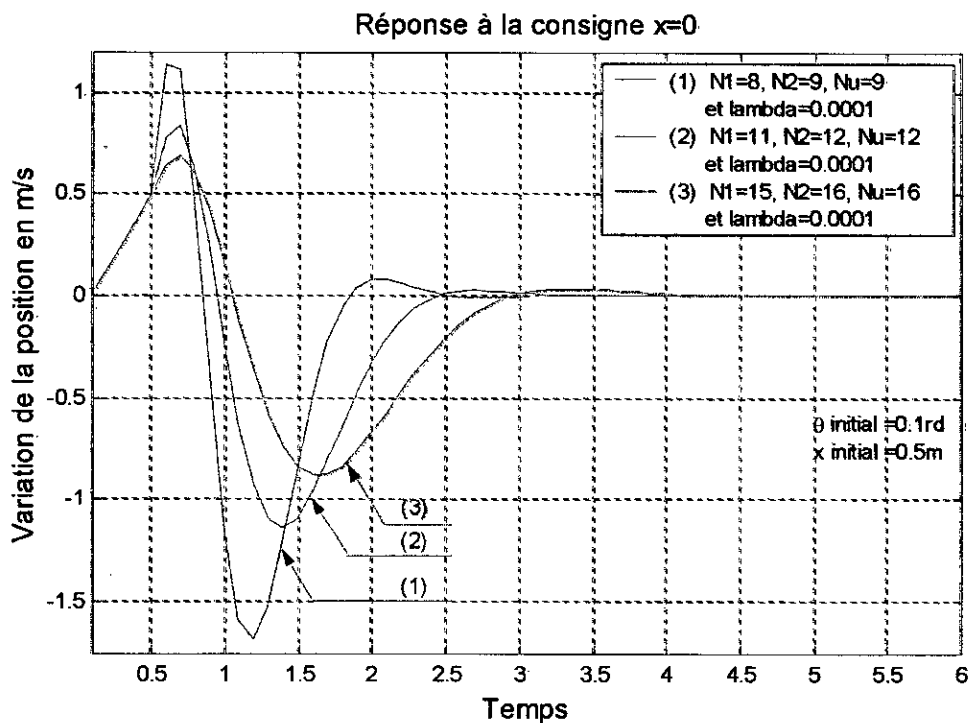


Figure 3.24. Réponse  $x'(t)$  à la consigne  $x = 0$

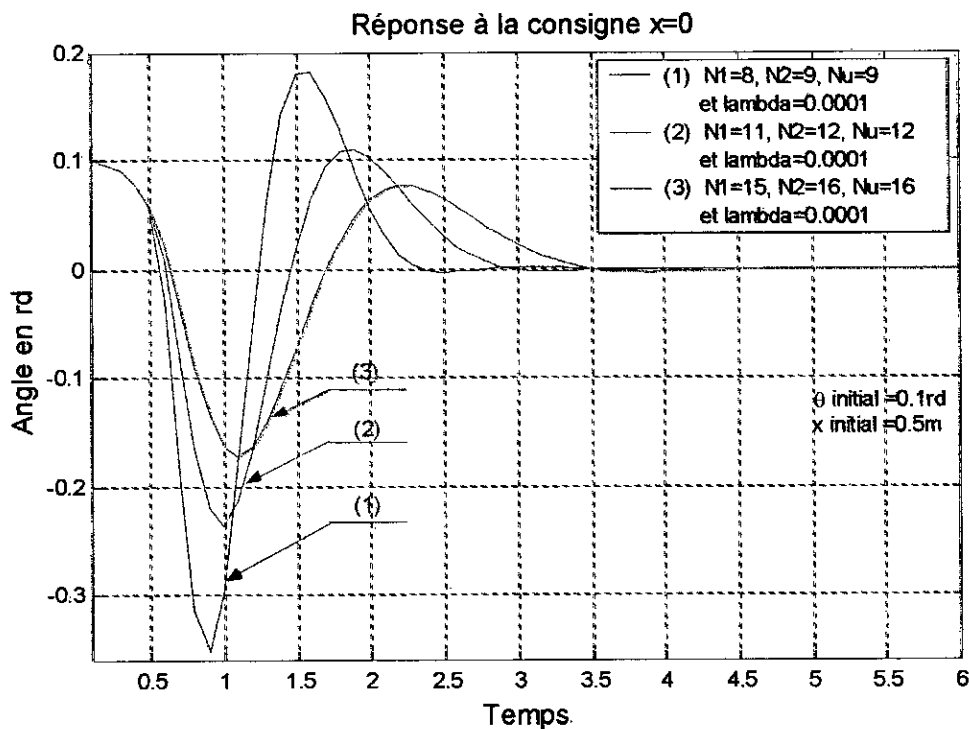


Figure 3.25. Réponse  $\theta(t)$  à la consigne  $x = 0$

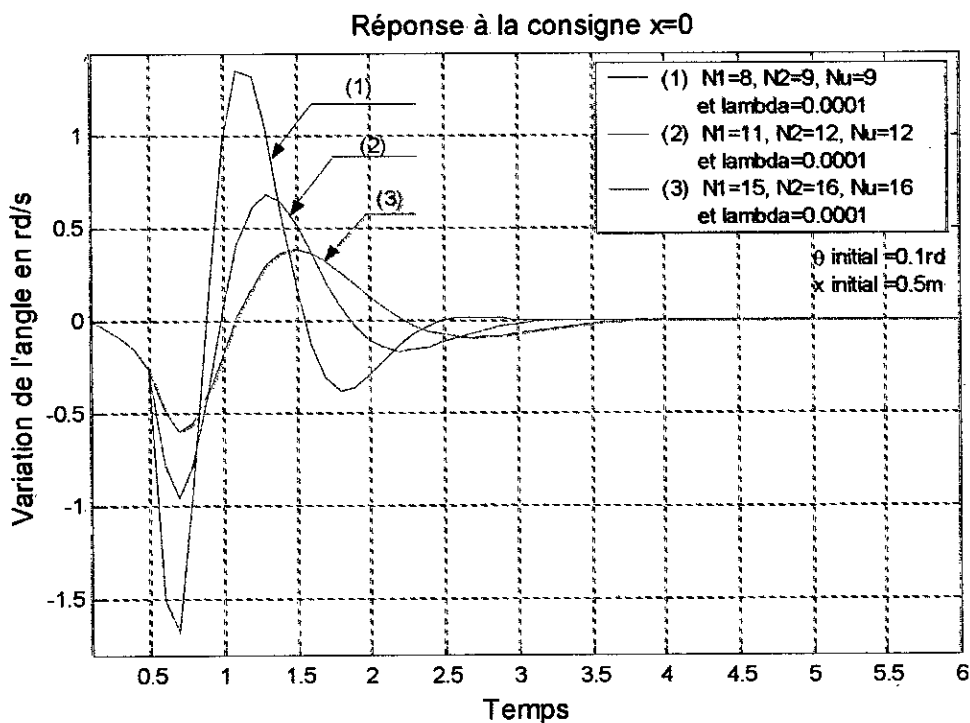
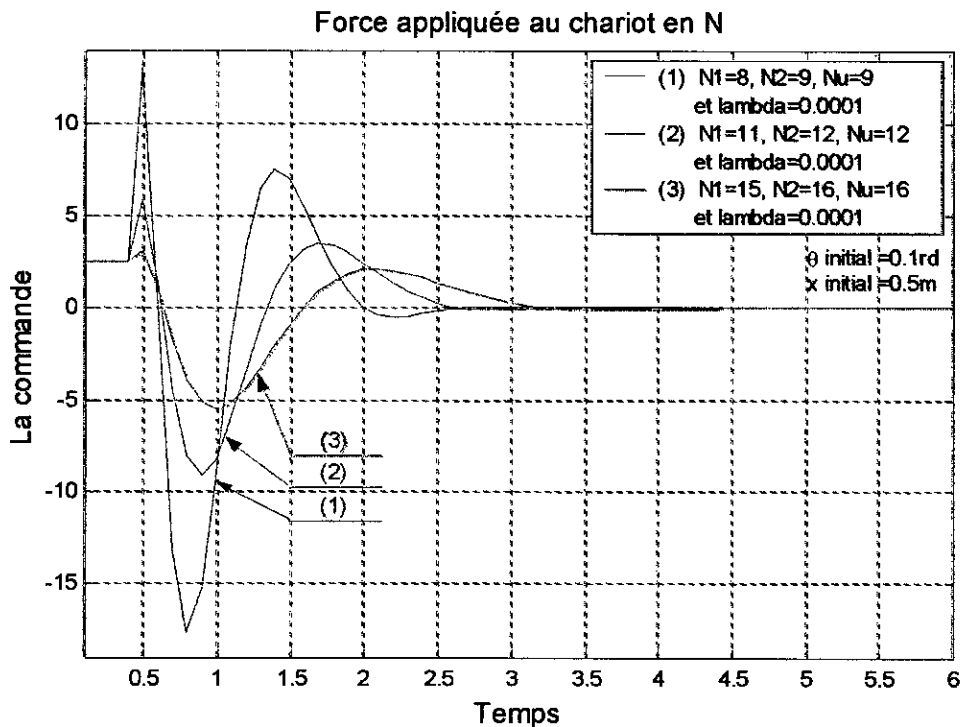


Figure 3.26. Réponse  $\theta'(t)$  à la consigne  $x = 0$



**Figure 3.27. Force appliquée au chariot en N**

On constate que ; avec  $N_1$  plus grand, la réponse du système est plus lente.

### 3.9.5. Simulation 4

Le contrôle prédictif généralisé est basé sur la minimisation d'un critère quadratique. Le critère à minimiser permet d'assurer un compromis entre l'erreur de poursuite et la variation de la commande. Le paramètre  $\lambda$  qui apparaît dans le critère (équation (2.9)) permet de maintenir l'énergie de commande dans des intervalles raisonnables. Le coefficient  $\lambda$  peut être déterminé a posteriori par simulation ; l'utilisateur pourrait choisir la valeur qui donne, soit le meilleur temps de réponse, soit des incréments de commande faibles.

On cherche dans cette simulation à optimiser la valeur de  $\lambda$  en utilisant les algorithmes génétiques.

Le critère à minimiser est donné par l'équation (3.20).

$$J_{GPCAG2} = J_{GPCAG1} + N_u + \lambda$$

$$\text{avec } J_{GPCAG1} = \sum_{t=0}^t \left[ \sum_{j=N_1}^{N_2} [y(t+j) - w(t+j)]^2 + \sum_{j=1}^{N_u} \lambda [\Delta u(t+j-1)]^2 \right] \quad 3.20$$

Les paramètres utilisés pour ce test sont :

La taille de la population	:	$popsiz$	= 30
Probabilité de croisement	:	$pcross$	= 0.6
Probabilité de mutation	:	$pmutation$	= 0.033
Le nombre des paramètres	:	$nparam$	= 2
La longueur d'un paramètre	:	$lparam$	= 5 pour $N_u$
La longueur d'un paramètre	:	$lparam$	= 14 pour $\lambda$
La longueur de la chaîne	:	$lchrom$	= 19
L'horizon de prédiction minimal	:	$N_1$	= 8
L'horizon de prédiction maximal	:	$N_2$	= 9
Domaine de l'estimation de $N_u$	:	$[N_1, N_2]$	
Domaine de l'estimation de $\lambda$	:	$[10^{-7}, 0.01]$	

Les résultats obtenus après lancement de l'algorithme  $GPC\_AG$  sont montrés par la figure (3.28) :

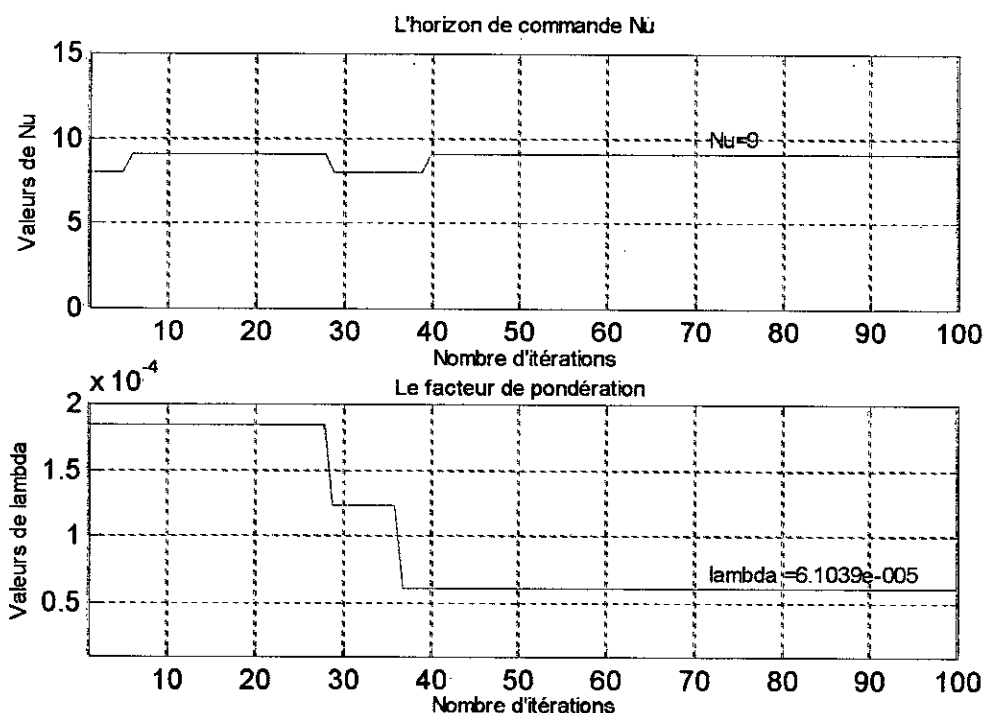


Figure 3.28. L'horizon de commande  $N_u$  et le facteur de pondération  $\lambda$

On constate que la valeur de  $\lambda$  trouvé par l'algorithme génétique est proche de celle choisie arbitrairement pour les simulations précédentes.

### 3.10. Comparaison

Par les figures suivantes, on montre la différence entre la réponse du système en utilisant les paramètres de réglage du *GPC* selon les règles évoquées dans le chapitre 2, et la réponse en utilisant les paramètres trouvés avec les algorithmes génétiques.

Selon *Clarke* [12], les valeurs des paramètres de réglage du *GPC* dans l'espace d'état sont :  $N_1 = n$ ,  $N_u = n$  et  $N_2 \geq 2n-1$  avec  $n$  est le nombre des états.

pour notre cas,  $N_1 = 4$ ,  $N_u = 4$  et  $N_2 \geq 7$ .

Et les paramètres optimaux trouvés après l'application des algorithmes génétiques sont :

$N_1 = 8$ ,  $N_u = 9$  et  $N_2 = 9$ .

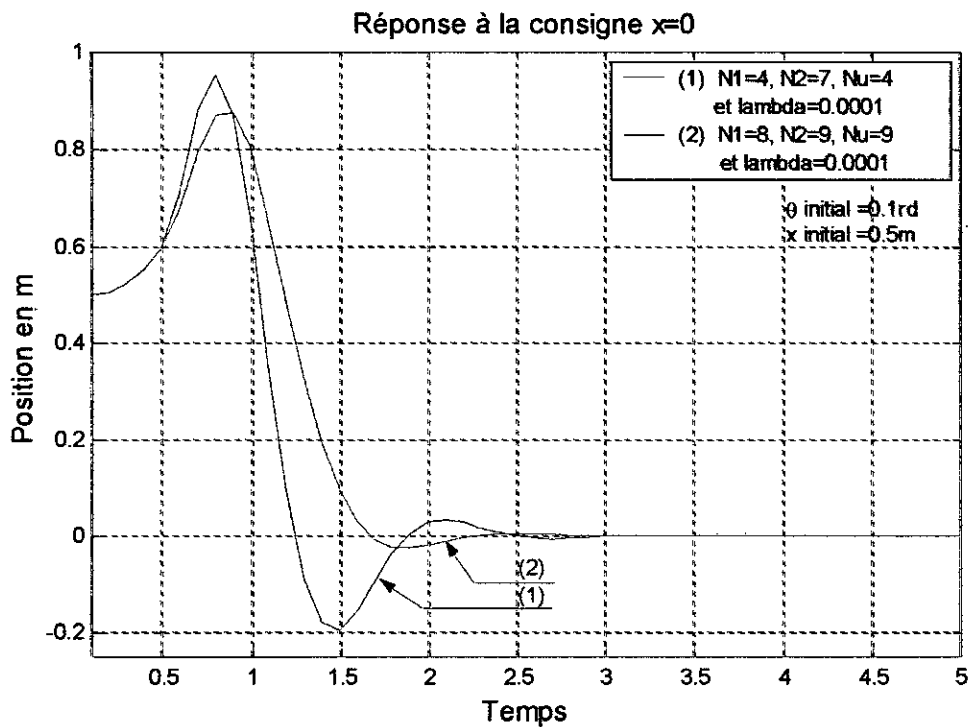


Figure 3.29. Réponse  $x(t)$  à la consigne  $x = 0$

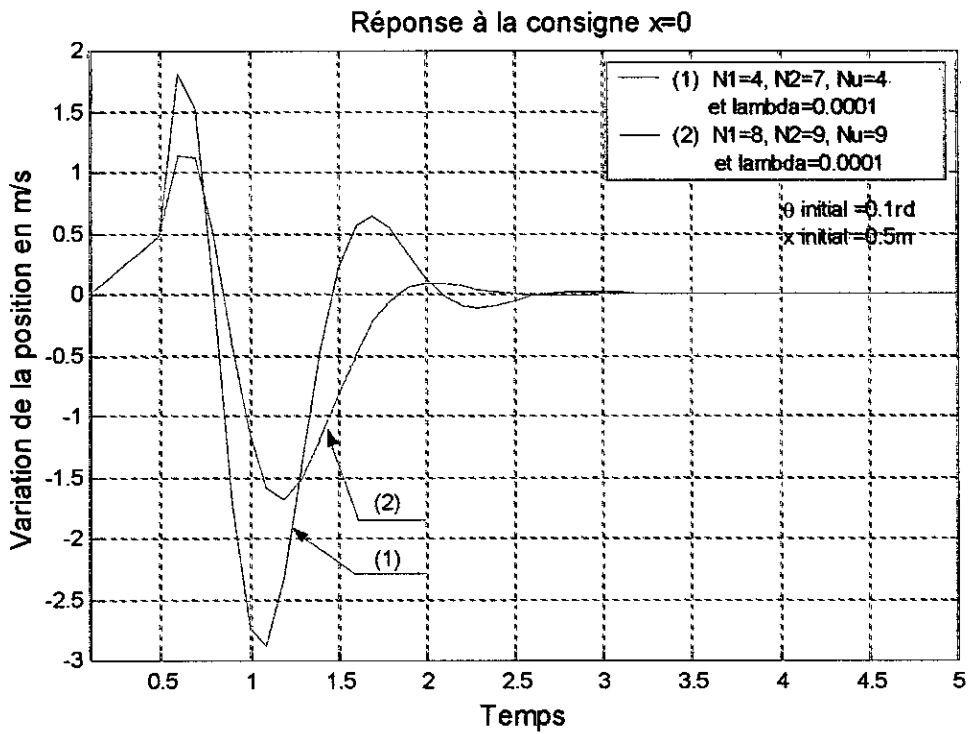


Figure 3.30. Réponse  $x'(t)$  à la consigne  $x = 0$

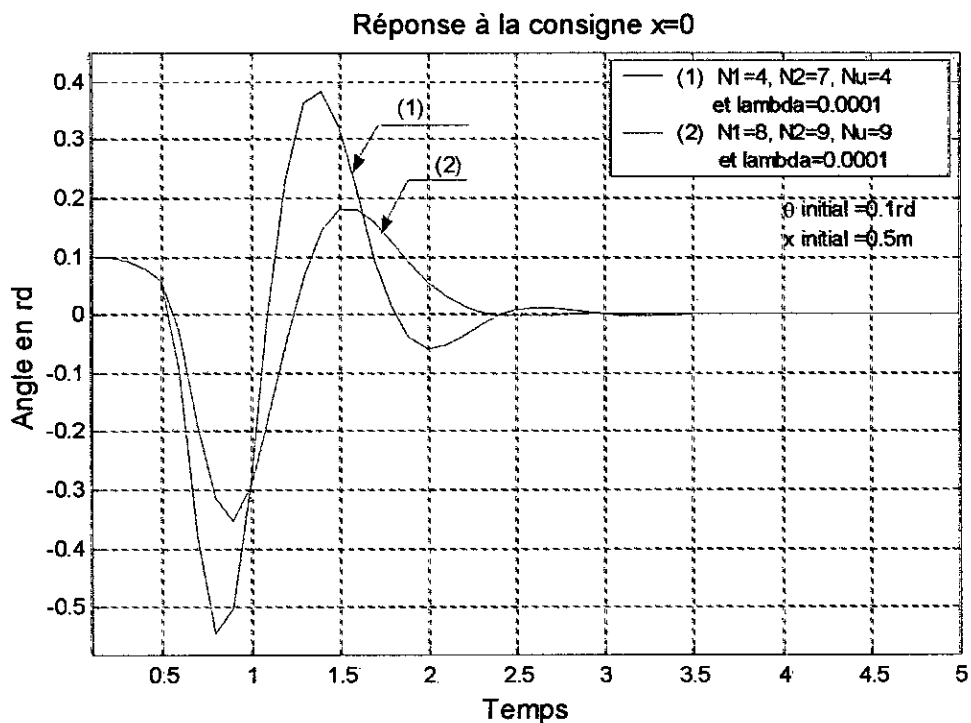
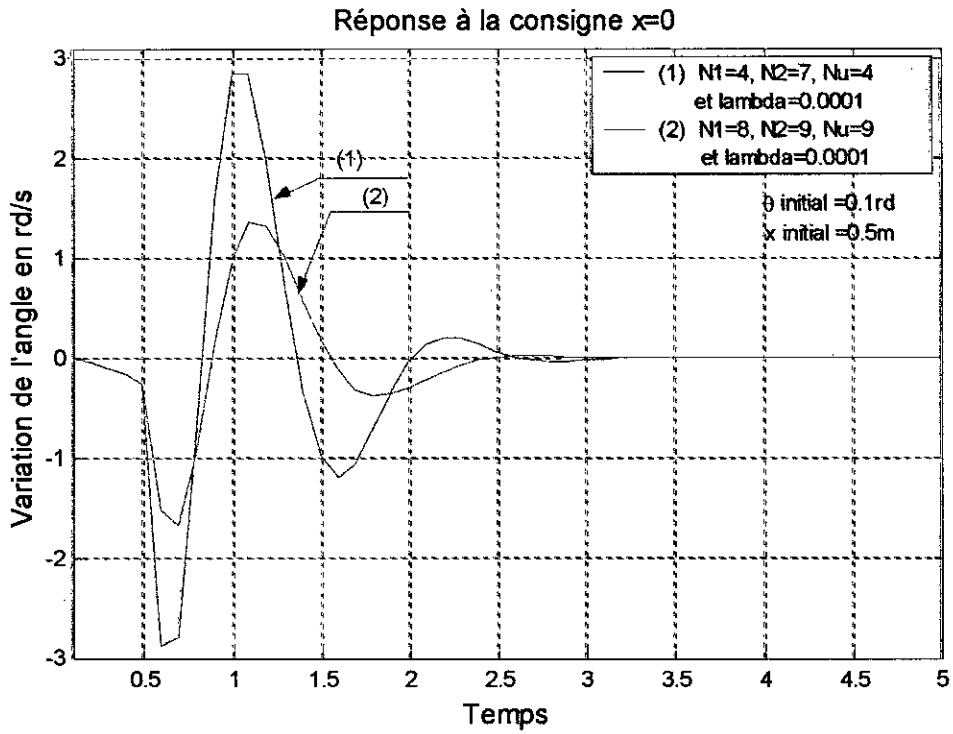
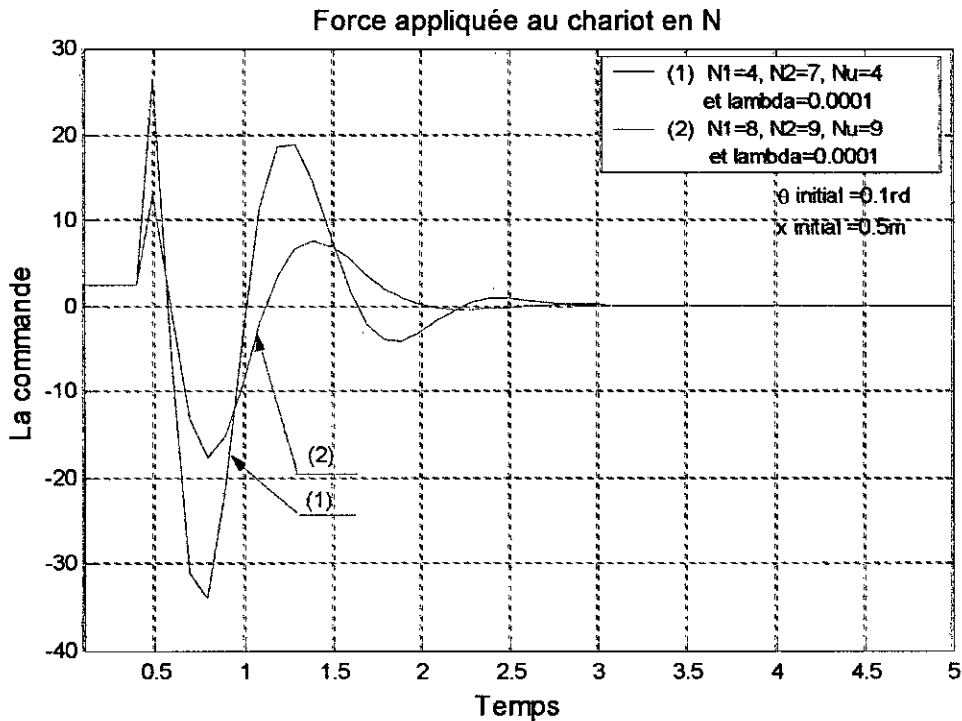


Figure 3.31. Réponse  $\theta(t)$  à la consigne  $x = 0$





**Figure 3.32. Réponse  $\theta'(t)$  à la consigne  $x = 0$**



**Figure 3.33. Force appliquée au chariot en N**

Pour les deux cas suscités, la réponse du processus en poursuite est donnée par les figures suivantes :

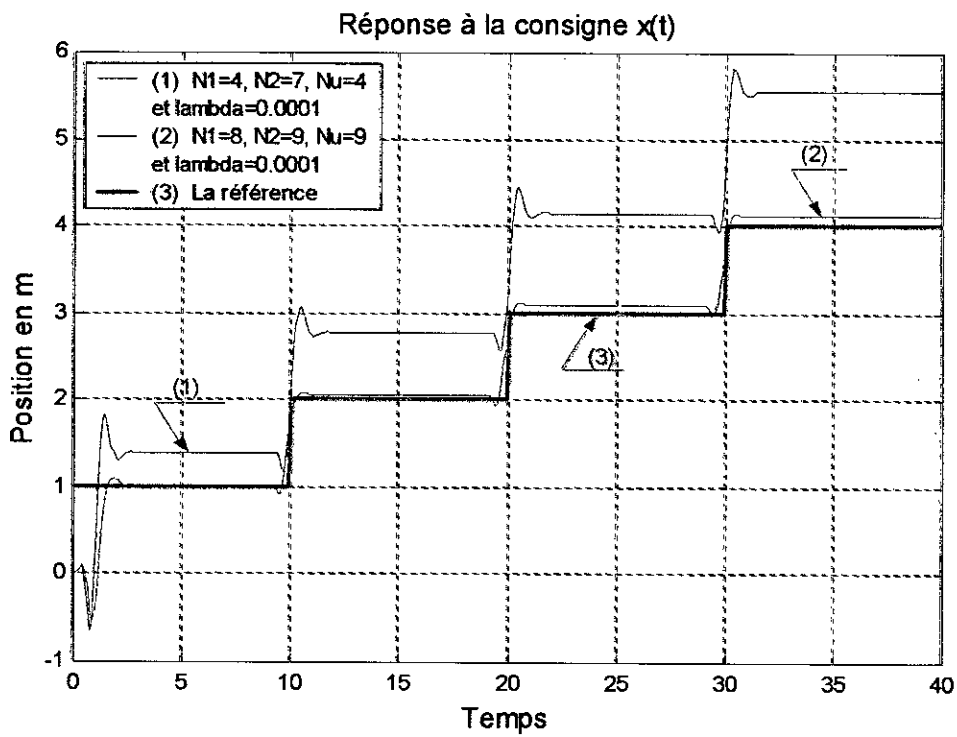


Figure 3.34.

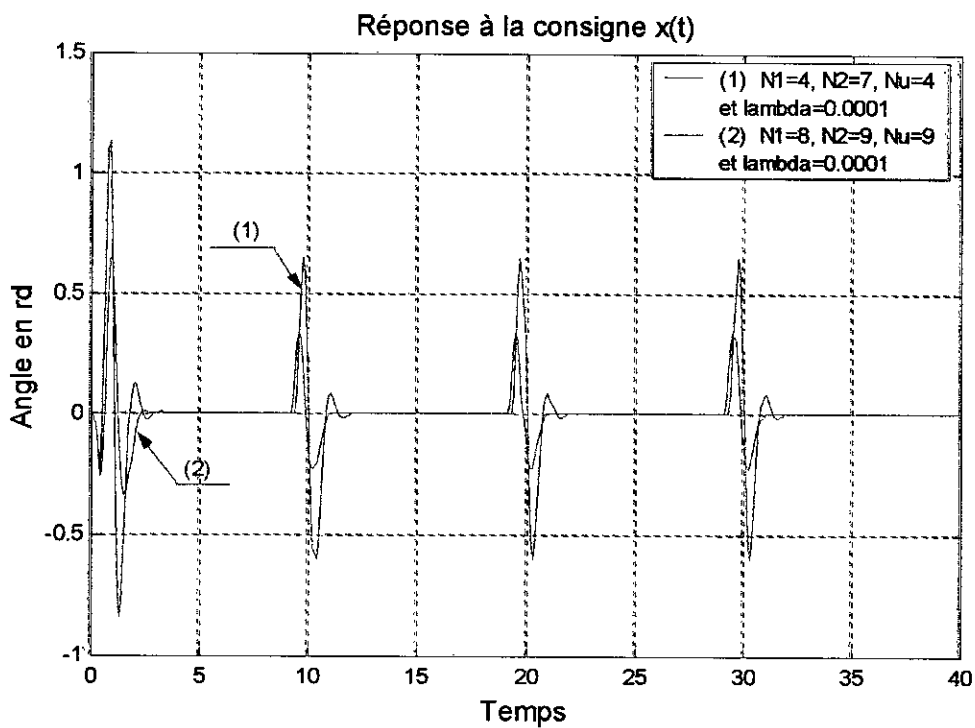


Figure 3.35.

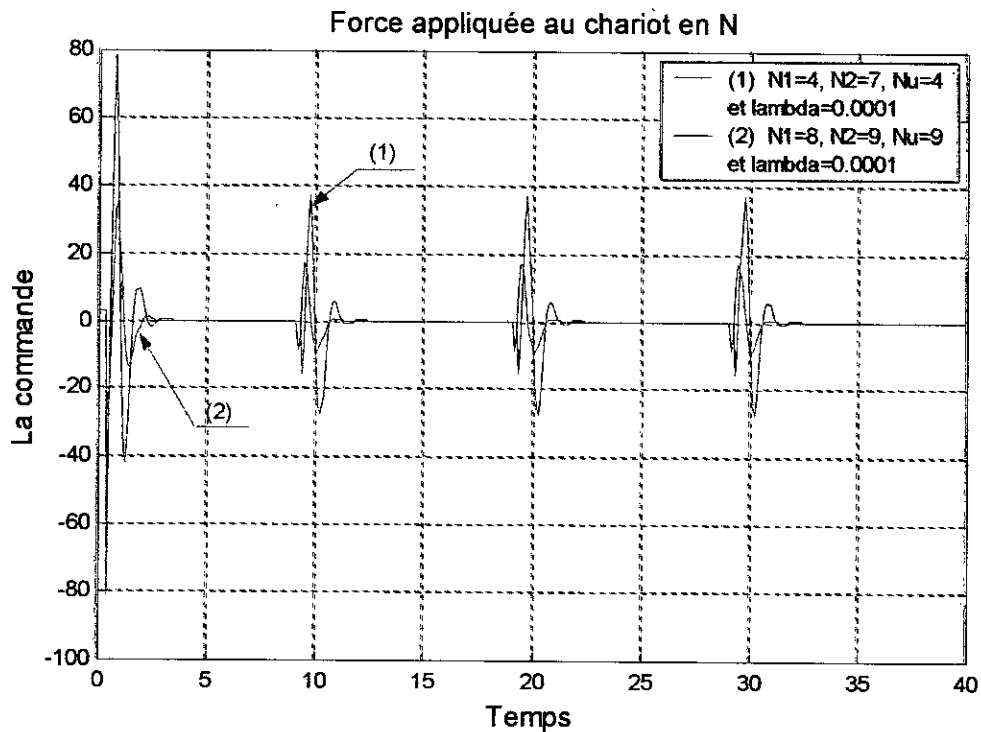


Figure 3.36.

### 3.11. Conclusion

Après les tests de simulation, on a remarqué que l'algorithme *GPC\_AG* proposé pour optimiser les valeurs des paramètres de réglage du contrôleur *GPC* s'adapte pour donner les meilleures valeurs qui minimisent, pour chaque test, le critère  $J_{GPC}$ .

Dans ce chapitre, on a montré comment on a combiné les algorithmes génétiques avec le contrôle prédictif généralisé pour l'optimisation des paramètres de réglage ( $N_1, N_2, N_u$  et  $\lambda$ ).

# Conclusion générale

Dans ce travail, on a démontré l'efficacité du contrôle prédictif généralisé pour la régulation d'un système donné.

La démarche suivie a conduit à une étude détaillée sur plusieurs régulateurs numériques classiques. On a présenté d'abord le calcul des régulateurs *P.I.D.* numériques, ensuite, on a examiné les méthodes suivantes : placement des pôles, poursuite et régulation à objectifs indépendants, poursuite et régulation avec pondération de l'entrée qui permettent de traiter des systèmes de n'importe quel ordre avec ou sans retard pur. Dans la partie finale du premier chapitre, les méthodes de calcul des régulateurs numériques en présence de perturbations aléatoires sont expliquées.

Le contrôle prédictif généralisé *GPC* est présenté dans le deuxième chapitre. Pour l'application de cet algorithme, on a choisit le pendule inversé qui est un système non linéaire où on a cherché d'asservir la position du chariot tout en maintenant le pendule inversé dans sa position verticale. Après la simulation, on a abouti à des résultats acceptables, mais le choix des paramètres de conception du *GPC* a fait l'objet de certaines difficultés.

La stabilité du contrôle prédictif généralisé dépend du choix des paramètres de conception ( $N_1$ ,  $N_2$ ,  $N_u$  et  $\lambda$ ) pour n'importe quel système. Donc, ces paramètres ont une influence déterminante sur le comportement du système. Mais, il n'est pas toujours facile de trouver des valeurs optimales pour ces paramètres.

Les algorithmes génétiques sont des outils d'optimisation performants qui permettent de réaliser une exploration globale de l'espace. Contrairement aux méthodes déterministes classiques, ils ne nécessitent aucun calcul de dérivées et peuvent être appliqués aussi bien à des fonctions continues d'une seule variable, qu'à des fonctions discontinues dépendant d'un grand nombre de paramètres.

Néanmoins, on a combiné le contrôle prédictif généralisé avec la théorie des algorithmes génétiques pour optimiser les paramètres de réglage du *GPC*. Avec les simulations présentées dans le chapitre 3, on a montré comment les meilleures valeurs étaient choisies pour obtenir un *GPC* stable et qui satisfait les performances souhaitées.

La conception des contrôleurs est donc une affaire d'expérience, de bon sens de compromis entre une performance désirée (un système doit répondre rapidement aux variations de la consigne, réduire au minimum une fonction quadratique de coût) et la commande à appliquer.

Enfin, il est à noter que l'inconvénient de la commande appliquée au pendule (le système traité dans ce travail) est son manque de robustesse par rapport à une variation des paramètres (par exemple si la petite masse du pendule est mal estimée). La mise en œuvre d'une commande floue est particulièrement efficace sur cet exemple. Donc, la certitude d'avoir une action de contrôle acceptable reste toujours un problème pour les systèmes non linéaires avec des paramètres changeants. La combinaison d'un contrôle prédictif généralisé avec les techniques nouvelles tel que le flou et les algorithmes génétiques pour l'optimisation des paramètres de réglage du *GPC* fait partie des perspectives d'avenir.

# Annexe A

## Bruit et Modèles Stochastiques

De nombreux procédés sont sujets à des perturbations de type aléatoire. En pratique la plupart des perturbations aléatoires intervenant en automatique peuvent être décrites avec une bonne précision comme un "bruit blanc discret" passé par un filtre.

Le "bruit blanc" joue dans le cas aléatoire le même rôle que l'impulsion de *Dirac* dans le cas déterministe. Il constitue le "signal générateur".

Le bruit blanc gaussien est un processus indépendant dans le temps (séquence de variables aléatoires indépendantes) de valeur moyenne nulle et variance  $\sigma^2$ . Cette séquence sera notée  $[e(t)]$  et sera caractérisée par les paramètres  $(0, \sigma)$  où le premier chiffre indique la valeur moyenne et  $\sigma$  l'écart (racine carrée de la variance).

La valeur moyenne (ou l'espérance) est donnée par :

$$V_M = E\{e(t)\} = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N e(t) = 0 \quad \text{A.1}$$

et la variance sera donnée par :

$$\text{var} = E\{e^2(t)\} = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N e(t)^2 = \sigma^2 \quad \text{A.2}$$

Nous définissons par fonction d'autocorrélation ou covariance  $R(i)$  l'expression :

$$R(i) = E\{e(t)e(t-i)\} = \lim_{N \rightarrow \infty} \frac{1}{N} \sum_{t=1}^N e(t) \times e(t-i) \quad \text{A.3}$$

ou en d'autres mots le produit entre la séquence et la même séquence décalée de  $i$ -pas.

Notons que :

$$R(0) = \text{var} = \sigma^2 \quad \text{A.4}$$

Définissons aussi la covariance (ou autocorrélation) normalisée :

$$RN(i) = \frac{R(i)}{R(0)} \quad (RN(0) = 1) \quad \text{A.5}$$

Le bruit blanc gaussien est une séquence de variables aléatoires indépendantes, la connaissance de  $e(t)$  ne permet pas de prédire une approximation de  $e(t+1)$ ,  $e(t+2)$ .... La meilleure prédiction étant 0.

Cette propriété d'indépendance se traduit par la propriété suivante des autocorrélations : (test d'indépendance).

$$R(i) = RN(i) = 0 \quad (i = 1, 2, 3, \dots, -1, -2, -3, \dots) \quad \text{A.6}$$

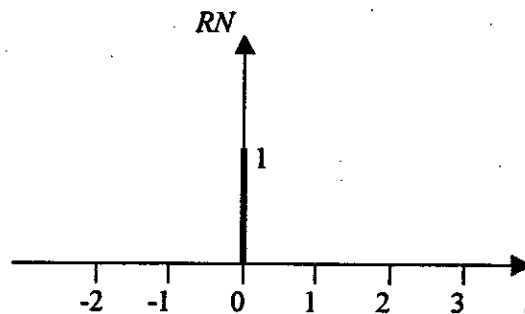


Figure A.1. Autocorrélations normalisées du bruit blanc

## 1. Modèles de perturbations aléatoires

Les modèles qui permettent de représenter les perturbations aléatoires sont dits modèles stochastiques, leur connaissance est équivalente à la connaissance de la perturbation. On examinera quelques types de processus obtenus ainsi.

### 1.1. Processus "Moyenne Mobile Ajustée" (M.A.)

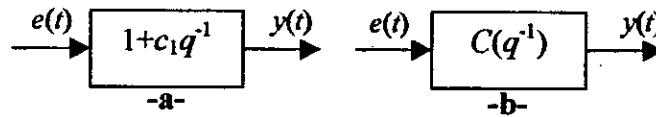
La forme générale d'un processus à Moyenne Mobile Ajustée est :

$$y(t) = e(t) + \sum_{i=1}^{n_c} c_i e(t-i) = C(q^{-1})e(t) \quad \text{A.7}$$

où

$$C(q^{-1}) = 1 + \sum_{i=1}^{n_c} c_i q^{-i} \quad \text{A.8}$$

Ce processus correspond au passage d'un bruit blanc à travers un filtre  $C(q^{-1})$  tel qu'il est représenté dans la figure (A.2).



**Figure A.2. Génération d'un processus aléatoire "Moyenne Mobile"**  
a : premier ordre b : cas général

## 1.2. Processus "Auto-Régressif" (A.R.)

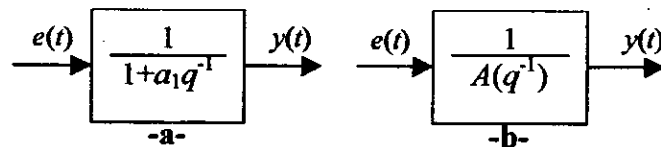
La forme générale de ce processus est donnée par :

$$y(t) = -\sum_{i=1}^{n_A} a_i y(t-i) + e(t) \quad \text{A.9}$$

qui s'écrit encore :  $A(q^{-1})y(t) = e(t) \quad \text{A.10}$

où  $A(q^{-1}) = 1 + \sum_{i=1}^{n_A} a_i q^{-i} \quad \text{A.11}$

est un polynôme avec toutes les racines à l'intérieur du cercle unité, c'est à dire que le processus est stable.



**Figure A.3. Génération d'un processus aléatoire "Auto-Régressif"**  
a : premier ordre b : cas général

## 1.3. Processus Auto-Régressif à Moyenne Ajustée (A.R.M.A.)

Ce processus est obtenu par le passage d'un bruit blanc à travers un filtre avec des pôles et des zéros tel qu'il est illustré dans la figure (A.4).



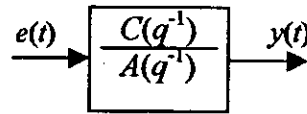


Figure A.4. Génération d'un processus aléatoire *A.R.M.A.*

Ce processus est décrit dans le cas général par :

$$y(t) = -\sum_{i=1}^{n_A} a_i y(t-i) + \sum_{i=1}^{n_C} c_i e(t-i) + e(t) \quad \text{A.12}$$

ou encore sous la forme :

$$A(q^{-1})y(t) = C(q^{-1})e(t) \quad \text{A.13}$$

où  $A(q^{-1})$  est donné par l'équation (A.11) et  $C(q^{-1})$  par l'équation (A.8).

#### 1.4. Le modèle *A.R.M.A.X.*

Il s'agit du modèle utilisé pour représenter ensemble l'effet de la commande et des perturbations sur la sortie du procédé. *A.R.M.A.X.* signifie processus *A.R.M.A.* avec une entrée exogène, qui dans notre cas est  $u(t)$ . La génération des processus *A.R.M.A.X.* est illustrée dans la figure (A.5).

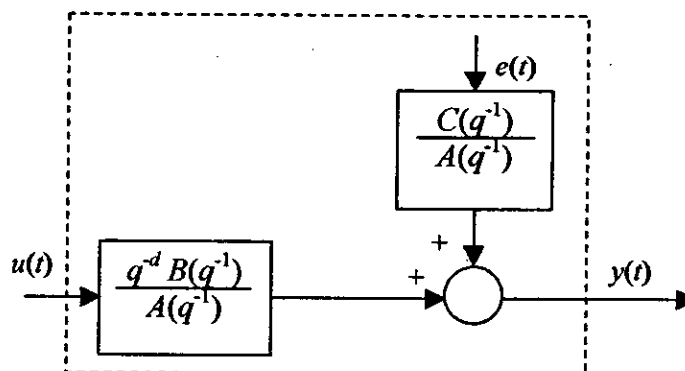


Figure A.5. Modèle *A.R.M.A.X.*

La sortie perturbée du procédé  $y(t)$  s'écrit :

$$y(t) = \frac{q^{-d} B(q^{-1})}{A(q^{-1})} u(t) + \frac{C(q^{-1})}{A(q^{-1})} e(t) \quad \text{A.14}$$

où le premier terme représente l'effet de la commande et le deuxième terme l'effet de la perturbation. En multipliant les deux membres de l'équation (A.14) par  $A(q^{-1})$ , on obtient :

$$A(q^{-1})y(t) = q^{-d}B(q^{-1})u(t) + C(q^{-1})e(t) \quad \text{A.15}$$

qui en tenant compte des expressions de  $A(q^{-1})$ ,  $B(q^{-1})$  et  $C(q^{-1})$  s'écrit encore :

$$y(t) = -\sum_{i=1}^{n_A} a_i y(t-i) + \sum_{i=1}^{n_B} b_i u(t-d-i) + \sum_{i=1}^{n_C} c_i e(t-i) + e(t) \quad \text{A.16}$$

On parle alors de structure *A.R.M.A.X* (Auto Régressive à Moyenne Ajustée et variable exogène).

## Annexe B

# Forme du Prédicteur dans le cas Déterministe et dans le cas Stochastique

### 1. Environnement déterministe

On considère les systèmes invariants dans le temps décrits par les modèles échantillonnés, de la forme :

$$y(t) = -\sum_{i=1}^{n_A} a_i y(t-i) + \sum_{i=1}^{n_B} b_i u(t-d-i) \quad \text{B.1}$$

avec

$t$  : le temps discret normalisé (temps réel divisé par la période d'échantillonnage,  $t = \frac{t}{T_e}$ ).

$y(t)$  : la sortie.

$u(t)$  : l'entrée.

$d$  : un retard pur multiple entier de la période d'échantillonnage.

$a_i, b_i$  : les paramètres du modèle

Le modèle de l'équation (B.1) peut être représenté par des opérateurs de changement, de retard ou d'avance :

$q$  : opérateur d'avance.

$q^{-1}$  : opérateur de retard.

On utilise les mêmes notations comme dans le chapitre 1, on obtient la forme suivante :

$$A(q^{-1})y(t) = q^{-d}B(q^{-1})u(t) \quad \text{B.2}$$

Et, en avance du temps, on obtient la forme prédictive décalée d'un pas :

$$y(t+1) = -A^*(q^{-1})y(t) + q^{-d}B^*(q^{-1})u(t) \quad \text{B.3}$$

### 1.1. Forme du prédicteur

Le problème est de prédire pour le modèle (B.1) les valeurs futures de la sortie au-delà de  $t+1$  basant sur les informations disponibles jusqu'à l'instant  $t$ . Dans le cas déterministe les perturbations n'affectent pas le système dans le futur. L'exemple typique est le contrôle d'un système avec un retard  $d$ , prédire sa sortie  $d+1$  pas en avant ou calculer  $u(t)$  afin d'atteindre certaine valeur de la sortie à  $t+d+1$ .

L'objectif est en général de calculer :

$$\hat{y}(t+j/t) = f\{y(t), y(t-1), \dots, u(t), u(t-1), \dots\} \quad \text{B.4}$$

ou la valeur prédite filtrée :

$$P(q^{-1})\hat{y}(t+j/t) = fP\{y(t), y(t-1), \dots, u(t), u(t-1), \dots\} \quad \text{B.5}$$

$$\text{où } P(q^{-1}) = 1 + p_1q^{-1} + p_2q^{-2} + \dots + p_{n_p}q^{-n_p} \quad \text{B.6}$$

$$n_p \leq n_A + j - 1$$

donc pour  $1 \leq j \leq d+1$ , la valeur prédite filtrée peut être exprimée par

$$P(q^{-1})\hat{y}(t+j) = F_j(q^{-1})y(t) + E_j(q^{-1})B^*(q^{-1})u(t+j-d-1) \quad \text{B.7}$$

$$\text{avec } \deg(P(q^{-1})) \leq n_A + j - 1 \quad \text{B.8}$$

où  $E_j(q^{-1})$  et  $F_j(q^{-1})$  sont les solutions de l'équation polynomiale :

$$P(q^{-1}) = A(q^{-1})E_j(q^{-1}) + q^{-j}F_j(q^{-1}) \quad \text{B.9}$$

$$E_j(q^{-1}) = 1 + e_1q^{-1} + \dots + e_{j-1}q^{-j+1}$$

$$\text{avec : } F_j(q^{-1}) = f_0^j + f_1^j q^{-1} + \dots + f_{n_F}^j q^{-n_F} \quad \text{B.10}$$

$$n_F = \max(n_A - 1, n_p - j)$$

## 2. Environnement stochastique

Dans beaucoup de situations pratiques, le modèle déterministe  $d'E/S$  donné par l'équation (B.1) ne prend pas en compte la présence des perturbations stochastiques. Donc le modèle a besoin d'un accommodement dans la présence de tels perturbations.

$$y(t+1) = -A^*(q^{-1})y(t) + q^{-d}B^*(q^{-1})u(t) + v(t+1) \quad \text{B.11}$$

où  $v(t)$  est le processus stochastique.

Donc, on a besoin de caractériser ces perturbations afin de prédire la conduite du système et le contrôler.

La plupart des perturbations aléatoires intervenant en automatique peuvent être décrites avec une bonne précision comme un "bruit blanc discret" passé par un filtre.

## 2.1. Forme du prédicteur

Dans le contexte stochastique, la prédiction prend sa signification, les valeurs futures de la sortie seront affectées par les perturbations.

Cependant l'information sur le modèle de perturbation sera pris en compte pour la construction du prédicteur.

On considère le modèle *A.R.M.A.X*.

$$y(t+1) = -A^*(q^{-1})y(t) + q^{-d}B^*(q^{-1})u(t) + C(q^{-1})e(t+1) \quad \text{B.12}$$

le prédicteur a la forme :

$$\hat{y}(t+j|t) = \hat{y}(t+j) = f\{y(t), y(t-1), \dots, u(t), u(t-1), \dots\} \quad \text{B.13}$$

$$\text{où } E\left\{[y(t+j) - \hat{y}(t+j)]^2\right\} = \min \quad \text{B.14}$$

Pour un système de l'équation (B.12) ( $C(q^{-1})$  est asymptotiquement stable et  $e(t)$  est un bruit blanc discret), le prédicteur optimal qui minimise le critère (B.14) est donné par :

$$\hat{y}(t+j) = \frac{F_j(q^{-1})}{C(q^{-1})}y(t) + \frac{E_j(q^{-1})B^*(q^{-1})}{C(q^{-1})}u(t+j-d-1) \quad \text{B.15}$$

où  $E_j(q^{-1})$  et  $F_j(q^{-1})$  sont les solutions de l'équation polynomiale :

$$C(q^{-1}) = A(q^{-1})E_j(q^{-1}) + q^{-j}F_j(q^{-1}) \quad \text{B.16}$$

$$E_j(q^{-1}) = 1 + e_1q^{-1} + \dots + e_{j-1}q^{-j+1}$$

$$\text{avec : } F_j(q^{-1}) = f_0^j + f_1^j q^{-1} + \dots + f_{n_F}^j q^{-n_F} \quad \text{B.17}$$

$$n_F = n_A - 1 \quad n_C \leq n_A$$

## Annexe C

### Solutions Récursives des Equations Diophantine

Le calcul de  $\hat{Y}$  donné par :

$$\hat{Y} = [\hat{y}(t+1), \hat{y}(t+2), \dots, \hat{y}(t+N_2)]^T \quad \text{C.1}$$

et la matrice  $G_1$ , exige la forme récursive pour les solutions de :

$$C(q^{-1}) = E_j(q^{-1})A(q^{-1})\Delta(q^{-1}) + q^{-j}F_j(q^{-1}) \quad \text{C.2}$$

$$G_j(q^{-1}) = G'_j(q^{-1})C(q^{-1}) + q^{-j}\Gamma_j(q^{-1}) \quad \text{C.3}$$

$$\text{avec } G_j(q^{-1}) = E_j(q^{-1})B(q^{-1}) \quad \text{C.4}$$

#### 1. Solution récursive de l'équation (C.2)

On considère l'équation polynomiale :

$$C(q^{-1}) = E_j(q^{-1})A(q^{-1})\Delta(q^{-1}) + q^{-j}F_j(q^{-1}) \quad \text{C.5}$$

avec :

$$A(q^{-1}) = 1 + a_1q^{-1} + \dots + a_{n_a}q^{-n_a} \quad \text{C.6}$$

$$B(q^{-1}) = b_0 + b_1q^{-1} + \dots + b_{n_b}q^{-n_b} \quad \text{C.7}$$

La solution récursive consiste à déterminer la solution  $(E_j(q^{-1}), F_j(q^{-1}))$  de la solution  $(E_{j-1}(q^{-1}), F_{j-1}(q^{-1}))$ , ces paires polynomiales représentent respectivement les solutions de :

$$C(q^{-1}) = E_j(q^{-1})\tilde{A}(q^{-1}) + q^{-j}F_j(q^{-1}) \quad \text{C.8}$$

$$C(q^{-1}) = E_{j-1}(q^{-1})\tilde{A}(q^{-1}) + q^{-j+1}F_{j-1}(q^{-1}) \quad \text{C.9}$$

avec :  $\tilde{A}(q^{-1}) = A(q^{-1})\Delta(q^{-1})$

$$E_j(q^{-1}) = e_0^j + e_1^j q^{-1} + \dots + e_{j-1}^j q^{-j+1} \quad \text{C.10}$$

$$E_{j-1}(q^{-1}) = e_0^{j-1} + e_1^{j-1} q^{-1} + \dots + e_{j-2}^{j-1} q^{-j+2} \quad \text{C.11}$$

$$F_j(q^{-1}) = f_0^j + f_1^j q^{-1} + \dots + f_{n_{F_j}}^j q^{-n_{F_j}} \quad \text{C.12}$$

$$F_{j-1}(q^{-1}) = f_0^{j-1} + f_1^{j-1} q^{-1} + \dots + f_{n_{F_{j-1}}}^{j-1} q^{-n_{F_{j-1}}} \quad \text{C.13}$$

Soustrayons l'équation (C.9) de l'équation (C.8), on obtient :

$$\tilde{A}(E_j - E_{j-1}) + q^{-j+1}(q^{-1}F_j - F_{j-1}) = 0 \quad \text{C.14}$$

le degré du polynôme  $E_j - E_{j-1}$  est  $j-1$ . Ce polynôme peut être écrit en deux parties :

$$E_j - E_{j-1} = L + e_{j-1}^j q^{-j+1} \quad \text{C.15}$$

donc  $\tilde{A}L + q^{-j+1}(q^{-1}F_j - F_{j-1} + \tilde{A}e_{j-1}^j) = 0 \quad \text{C.16}$

De l'équation (C.16),  $L = 0$  et  $F_j$  est donné par :

$$F_j(q^{-1}) = q(F_{j-1}(q^{-1}) - e_{j-1}^j \tilde{A}(q^{-1})) \quad \text{C.17}$$

Le premier élément de  $\tilde{A}$  est 1

pour  $j \geq 2$ , on obtient:

$$e_{j-1} = f_0^{j-1} \quad \text{C.18}$$

$$f_i^j = f_{i+1}^{j-1} - \tilde{a}_{i+1} e_{j-1} \quad 0 \leq i \leq n_{F_j} \quad \text{C.19}$$

avec  $E_j(q^{-1}) = e_0 + e_1 q^{-1} + \dots + e_{j-1} q^{-j+1} \quad \text{C.20}$

pour  $j = 1$

$$E_1(q^{-1}) = e_0$$

$$F_1(q^{-1}) = f_0^1 + f_1^1 q^{-1} + \dots + f_{n_{F_1}}^1 q^{-n_{F_1}}$$

on obtient :

$$e_0 = 1 \quad \text{C.21}$$

$$f_i^1 = c_{i+1} - \tilde{a}_{i+1} e_0 \quad 0 \leq i \leq n_{F_1} \quad \text{C.22}$$

## 2. Solution récursive de l'équation (C.3)

On considère l'équation polynomiale :

$$G_j(q^{-1}) = E_j(q^{-1})B(q^{-1}) = G'_j(q^{-1})C(q^{-1}) + q^{-j}\Gamma_j(q^{-1}) \quad \text{C.23}$$

La solution récursive consiste à déterminer la solution  $(G'_j(q^{-1}), \Gamma_j(q^{-1}))$  de la solution  $(G'_{j-1}(q^{-1}), \Gamma_{j-1}(q^{-1}))$ , ces paires polynomiales représentent respectivement les solutions de :

$$G_j(q^{-1}) = G'_j(q^{-1})C(q^{-1}) + q^{-j}\Gamma_j(q^{-1}) \quad \text{C.24}$$

$$G_{j-1}(q^{-1}) = G'_{j-1}(q^{-1})C(q^{-1}) + q^{-j+1}\Gamma_{j-1}(q^{-1}) \quad \text{C.25}$$

avec :

$$G_j(q^{-1}) = E_j(q^{-1})B(q^{-1}) \quad \text{C.26}$$

$$G_{j-1}(q^{-1}) = E_{j-1}(q^{-1})B(q^{-1}) \quad \text{C.27}$$

$$G'_j(q^{-1}) = g'^j_0 + g'^j_1 q^{-1} + \dots + g'^j_{j-1} q^{-j+1} \quad \text{C.28}$$

$$G'_{j-1}(q^{-1}) = g'^{j-1}_0 + g'^{j-1}_1 q^{-1} + \dots + g'^{j-1}_{j-2} q^{-j+2} \quad \text{C.29}$$

$$\Gamma_j(q^{-1}) = \gamma^j_0 + \gamma^j_1 q^{-1} + \dots + \gamma^j_{m_j} q^{-m_j} \quad \text{C.30}$$

$$\Gamma_{j-1}(q^{-1}) = \gamma^{j-1}_0 + \gamma^{j-1}_1 q^{-1} + \dots + \gamma^{j-1}_{m_{j-1}} q^{-m_{j-1}} \quad \text{C.31}$$

Soustrayons l'équation (C.25) de l'équation (C.24), on obtient :

$$C(G'_j - G'_{j-1}) + q^{-j+1}(q^{-1}\Gamma_j - \Gamma_{j-1}) - (G_j - G_{j-1}) = 0 \quad \text{C.32}$$



$$\text{avec : } G'_j - G'_{j-1} = L' + g'^{j-1} q^{-j+1} \quad \text{C.33}$$

$$E_j - E_{j-1} = L + e_{j-1}^j q^{-j+1} \quad \text{C.34}$$

donc l'équation (C.32) devient :

$$C(L' + g'^{j-1} q^{-j+1}) + q^{-j+1} (q^{-1} \Gamma_j - \Gamma_{j-1}) - B(L + e_{j-1}^j q^{-j+1}) = 0 \quad \text{C.35}$$

De l'équation (C.35),  $L = 0$  et  $L' = 0$

$$\text{donc } E_j(q^{-1}) = e_0 + e_1 q^{-1} + \dots + e_{j-1} q^{-j+1} \quad \text{C.36}$$

$$\text{et } G'_j(q^{-1}) = g'_0 + g'_1 q^{-1} + \dots + g'_{j-1} q^{-j+1} \quad \text{C.37}$$

et  $\Gamma_j$  est donné par :

$$\Gamma_j(q^{-1}) = q(\Gamma_{j-1}(q^{-1}) - g'_{j-1} C(q^{-1}) + B(q^{-1}) e_{j-1}) \quad \text{C.38}$$

$$\text{Si } C(q^{-1}) = 1 + c_1 q^{-1} + \dots + c_{n_c} q^{-n_c} \quad \text{C.39}$$

Pour  $j \geq 2$

$$g'_{j-1} = e_{j-1} b_0 + \gamma_0^{j-1} \quad \text{C.40}$$

$$\gamma_i^j = \gamma_{i+1}^{j-1} - g'_{j-1} c_{i+1} + e_{j-1} b_{i+1} \quad 0 \leq i \leq n_{\Gamma_j} \quad \text{C.41}$$

pour  $j = 1$

$$g'_0 = e_0 b_0 = \beta_0 \quad \text{C.42}$$

$$\gamma_i^1 = \beta_{i+1} - g'_0 c_{i+1} \quad \text{C.43}$$

avec  $G_1 = E_1 B = \beta_0 + \beta_1 q^{-1} + \dots$

## Annexe D

### Animation Graphique du Système (Chariot avec Pendule Inversé)

On désire réaliser une animation graphique du chariot avec pendule inversé asservi ; pour cela, on passe comme paramètres d'entrée  $x$  (la position) et  $\theta$  (l'angle) ; soit :  $[x(k), \theta(k)]$ . Les valeurs de ces paramètres sont trouvées après l'application d'un contrôleur de type *GPC*.

Le tracé mobile, chariot et pendule, est schématisé par le graphique ci-dessous (figure (C.1)) pour lequel on se fixe un repère orthonormé avec une étendue  $\pm 50$  en  $x$  et de  $-20$  à  $+100$  en  $y$ .

Les points de passage du tracé mobile sont précisés à l'aide de deux vecteurs,  $Mx$  pour les coordonnées en  $x$  et  $My$  pour les coordonnées en  $y$ . A chaque période d'échantillonnage ce couple de vecteurs est réactualisé.

On représente avec les figures d'animation graphique les résultats obtenus après l'application de l'algorithme *GPC* avec  $N_1 = 8$ ,  $N_2 = 9$ ,  $N_u = 9$  et  $\lambda = 0.0001$ . La consigne (une consigne de position  $x_c$ ) est nulle et les conditions initiales sont données par le vecteur suivant :  $X_0 = [0.5 \ 0 \ 0.1 \ 0]^T$

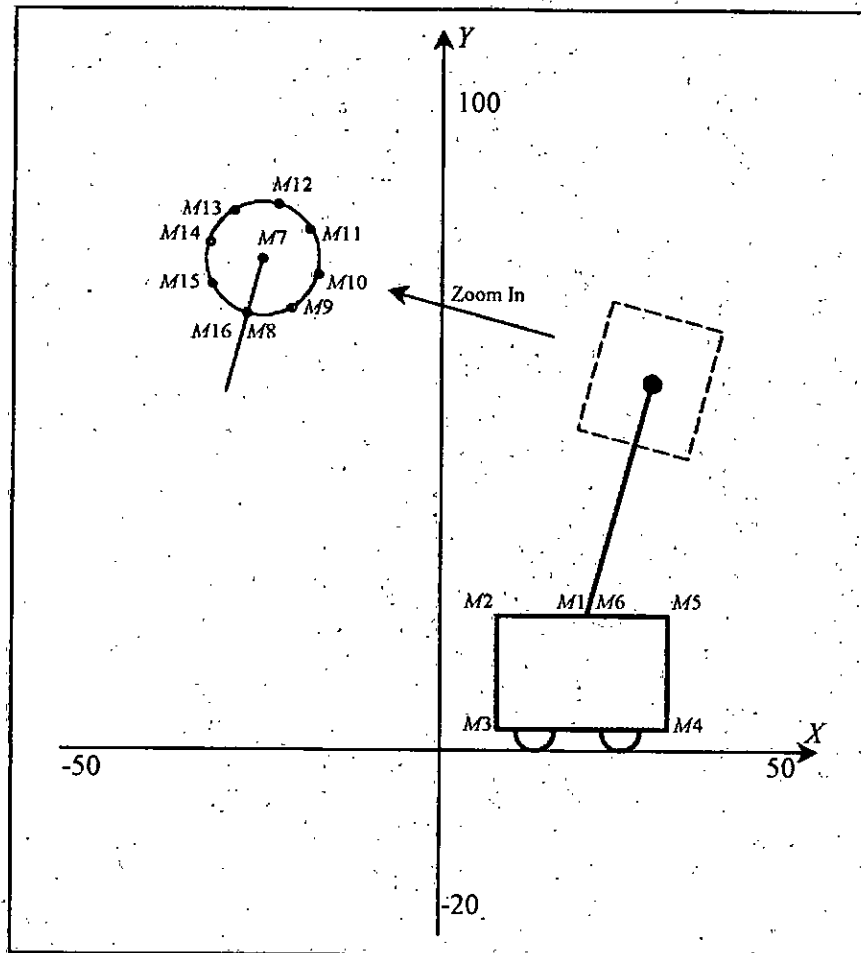


Figure D.1.

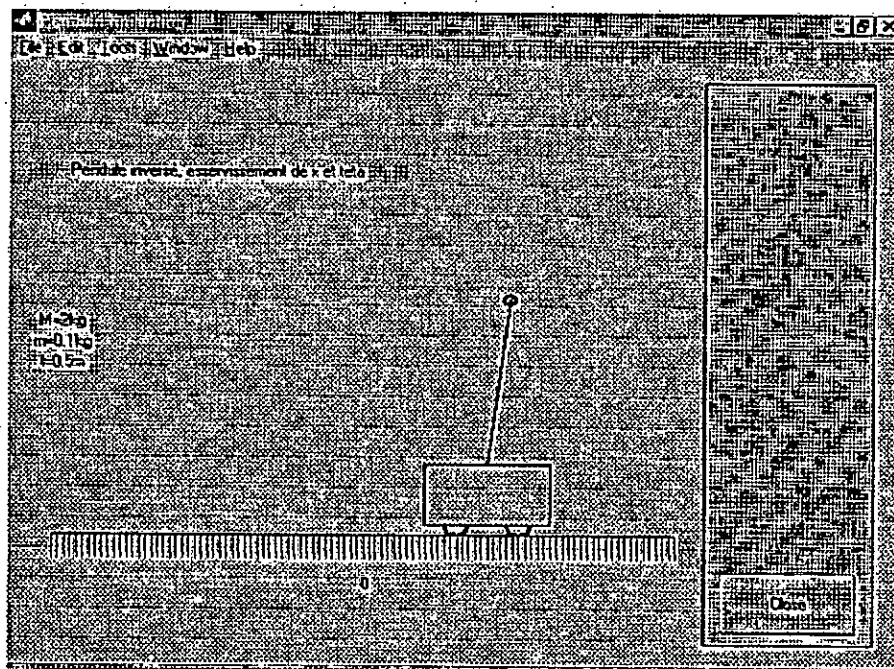


Figure D.2. ( $t = 0.1s$ ,  $x = 0.5m$ ,  $\theta = 0.1rd$ )

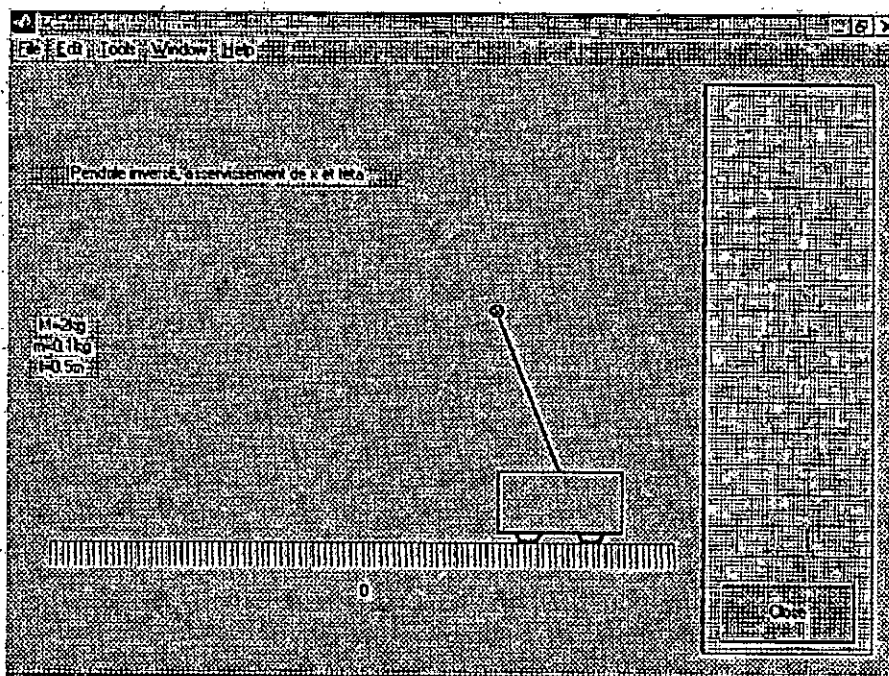


Figure D.3. ( $t = 1s$ ,  $x = 0.7933m$ ,  $\theta = -0.2977rd$ )

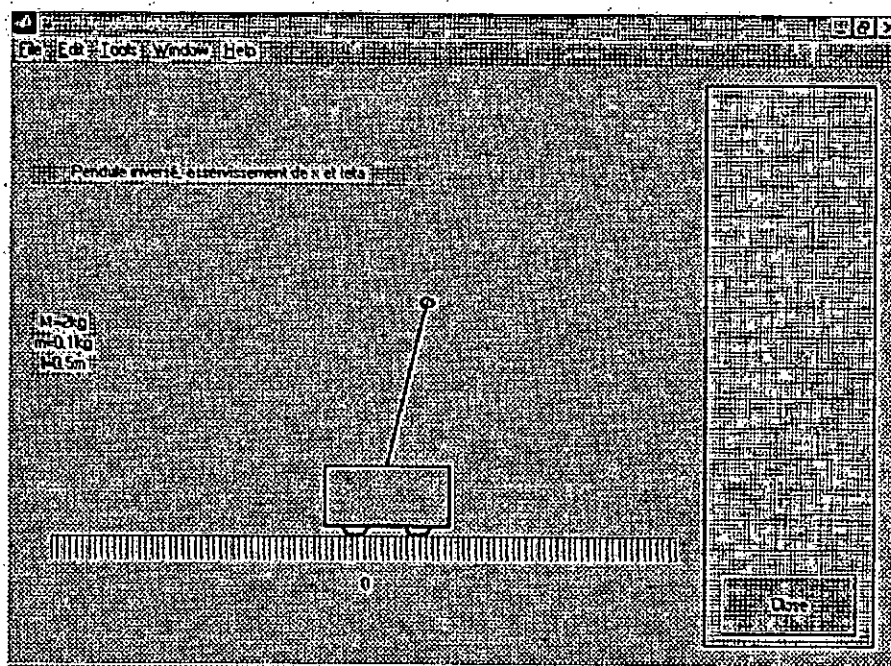


Figure D.4. ( $t = 1.5s$ ,  $x = 0.0942m$ ,  $\theta = 0.1790rd$ )

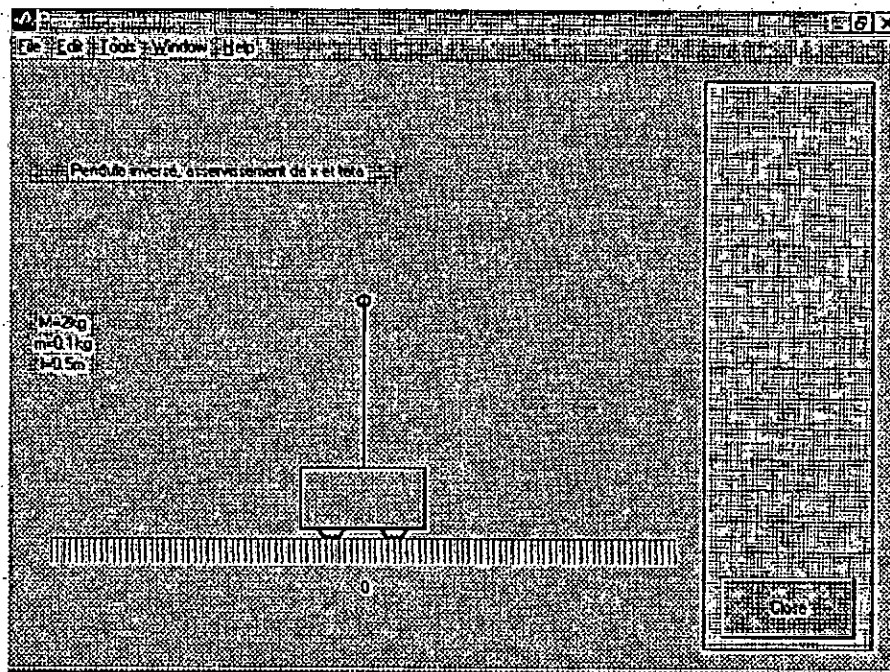


Figure D.5. ( $t = 3s, x = 0m, \theta = 0rd$ )

## Bibliographie

- [1] Ioan Doré LANDAU, "Identification et Commande des systèmes à l'aide des progiciels P.I.M. et PC-REG", HERMES, Paris, 1988.
- [2] I.D. LANDAU, R. LOZANO and M. M'SAAD, "Adaptive Control", Springer-Verlag London Limited, 1998.
- [3] Claude FOULARD, Sylviane GENTIL et Jean-Paul SANDRAZ, "Commande et Régulation par Calculateur Numérique", EYROLLES, Paris, 1982.
- [4] Pierre BORNE, Geneviève DAUPHIN-TANGUY et Jean-Pierre RICHARD, "Commande et Optimisation des processus", TECHNIP, Paris, 1990.
- [5] P. NASLIN, "Théorie de la Commande et Conduite Optimale", DUNOD, Paris, 1969.
- [6] David E. GOLDBERG, "Algorithmes Génétiques", ADDISON-WESLEY, France, 1994.
- [7] I.D. LANDAU et L. DUGARD, "Commande Adaptative, aspects pratiques et théoriques", MASSON, 1986.
- [8] Maurice RIVOIRE et Jean-Louis FERRIER, "Cours d'Automatique, Tome1 : Signaux et systèmes", CHIHAB-EYROLLES, 1994.
- [9] Maurice RIVOIRE et Jean-Louis FERRIER, "Cours d'Automatique, Tome2 : Asservissement-Régulation, Commande analogique", CHIHAB-EYROLLES, 1994.

- [10] Maurice RIVOIRE et Jean-Louis FERRIER, "Cours d'Automatique, Tome3 : Commande par ordinateur, Identification", CHIHAB-EYROLLES, 1994.
- [11] D.W. CLARKE, C. MOHTADI and P.S. TUFFS, "Generalized Predictive Control- Part I. The Basic Algorithm", *Automatica*, Vol.23, No.2, pp.137-148, 1987.
- [12] D.W. CLARKE, C. MOHTADI and P.S. TUFFS, "Generalized Predictive Control- Part II. Extensions and Interpretations", *Automatica*, Vol.23, No.2, pp.149-160, 1987.
- [13] R.GOREZ, V. WERTZ and ZHU Kuan-Yi, "On a Generalized Predictive Control Algorithm", *Systems & Control letters* 9, 369-377, 1987.
- [14] Mohamed Laid HADJILI, Vincent WERTZ and Gérard SCORLETTI, "Fuzzy Model-Based Predictive Control", *IEEE Conference on Decision & Control, Tampa, Florida USA*, 1998.
- [15] HUSEYIN, DEMIRCIOGLU and Ercan KARASU, "Generalized Predictive Control, A Practical Application and Comparison of Discrete and Continuous-Time Versions", *IEEE Control Systems Magazine*, 2000.
- [16] J. DUAN and M. GRIMBLE, "Design of Long-Predictive Control Algorithms for Industrial Applications", *University of Strathclyde, U.K.*
- [17] Carlos BORDONS and Eduardo F. CAMACHO, "A Generalized Predictive Controller for a Wide Class of Industrial Processes", *IEEE Transactions on Control Systems Technology*, vol.6, No.3, 1998.
- [18] MEGHERBI Hassina & BENMAHAMMED Khier, "Designing of Sliding Fuzzy Controller Using Genetic Algorithms" *Technologies avancées*, n.11, 1999.
- [19] Richard SPILLMAN, "Genetic Algorithms", *DR. Dobb's Journal*, 1993.
- [20] Salim FILALI and Vincent WERTZ, "Using Genetic Algorithms to Optimise the Design Parameters of Generalized Predictive Controllers", *Institut d'Electronique Université de Constantine, Algérie. Université catholique de Louvain, CESAME.*
- [21] Bruno SARENI, "Algorithmes Génétiques Standards", *Rapport Interne R-97-01, Centre de Génie Electrique de Lyon, Ecole Centrale de Lyon, CEGELY, France*, 1997.
- [22] HANNOUN Mahdi et AMROUCHE Amar, "GENETICA : Un environnement de développement destiné aux algorithmes génétiques", *Institut National de formation en Informatique, (I.N.I)*, Alger, 1997.
- [23] Pierre Yves GLORENNEC, "Algorithmes d'apprentissage pour les Réseaux Neuro-Flous", 1992.

- [24] R.M.C. De KEYSER and A.R. Van CAUWENBERGHE, "Extended Prediction Self-Adaptive Control", *Automatic Control Laboratory, University of Gent, Belgium*, 1985.
- [25] J. RICHALET, A. RAULT, J.L. TESTUD and J. PAPON, "Model Predictive Heuristic Control: applications to industrial processes", *Automatica, Vol.14, pp.413-428*, 1978.
- [26] D.A LINKENS and H.O. NYONGESA, "Genetic Algorithms for Fuzzy Control, Part 1: Offline system development and application", *IEE Proc. Control Theory Appl, Vol.142, No.3*, 1995.
- [27] C.L. KARR and E.J. GENTRY, "Fuzzy Control of pH Using Genetic Algorithms", *IEEE Transactions on Fuzzy Systems. Vol.1, No.1*, 1993.
- [28] K.J. ASTROM and B.WITTENMARK, "Adaptive Control", ADDISON-WESLEY, France, 1989.
- [29] Philippe COIFFET, "La Robotique, principes et applications", HERMES, Paris, 1992.
- [30] Pierre BORNE, Jean ROZINOER, Jean-Yves DIEULOT et Laurent DUBOIS "Introduction à la Commande Floue", TECHNIP, Paris, 1998.
- [31] Bernadette BOUCHON-MEUNIER, "La Logique Floue et ses applications", ADDISON-WESLEY, France, 1995.
- [32] Jean-Christophe CULIOLI, "Introduction à l'Optimisation", ELLIPSES, 1994.
- [33] F. MEGRY, M. CHEMACHEMA, M. BAHITA, K. BELARBI, "Application de la Programmation Dynamique Floue à la Commande Prédictive Non Linéaire", *Conférence Maghrébine en Génie Electrique, Constantine*, 2001.
- [34] M. CHEMACHEMA, F. MEGRY, M. BAHITA, K. BELARBI, "La Commande Prédictive avec Contraintes: une solution basée sur la méthode de FIBONACCI", *Conférence Maghrébine en Génie Electrique, Constantine*, 2001.
- [35] Salim FILALI, "Etude de l'influence des paramètres de réglage de la commande prédictive généralisée sur la robustesse de l'algorithme", *Thèse en vue de l'obtention du diplôme de Docteur d'état en Electronique, option automatique, Université de Constantine*, 2000.
- [36] Abdellatif M.BENYAHIA, "Commande non linéaire et prédictive, application à la machine asynchrone", *Thèse en vue de l'obtention du diplôme de magister, option contrôle, Université de Batna*, 2001.



- [37] Ibtiem ABDELMALEK, "Identification et commande floues des systèmes non linéaires", *Thèse en vue de l'obtention du diplôme de magister en Electronique, option contrôle industriel, Université de Batna, 1999.*
- [38] Kheireddine CHAFAA, "Etude de poursuite d'une cible", *Thèse en vue de l'obtention du diplôme de magister en Electronique, option contrôle industriel, Université de Batna, 1999.*