

République Algérienne Démocratique et populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud Mammeri de Tizi-Ouzou



Faculté de Génie Electrique et Informatique
Département Automatique

MEMOIRE DE MAGISTER

En Automatique

Option : Automatique des systèmes continus et productique

présenté par

IDIRI Ghania

Thème :

Commande prédictive des systèmes non linéaires dynamiques

Mémoire soutenu le 06 /07/ 2011 devant le jury d'examen composé de :

Président :	DJENNOUNE Saïd	Professeur à l'UMMTO
Rapporteur :	HAMMOUCHE Kamal	Maître de conférences A à l'UMMTO
Examineur :	OUANES Mohand	Maître de conférences A à l'UMMTO
Examineur :	MANSOURI Rachid	Maître de conférences A à l'UMMTO
Examineur :	DJAMAH Tounsia	Maître de conférences B à l'UMMTO

Sommaire

Introduction Générale	01
Chapitre 1. Généralités sur la Commande Prédictive	05
1.1 Introduction	05
1.2 Principe de la commande prédictive	06
1.3 Eléments d'une commande prédictive	08
1.3.1 Modèle du système	08
1.3.2 Prédiction	10
1.3.3 Critère de performance (fonction objectif)	10
1.3.4 Contraintes	12
1.3.5 Loi de commande	12
1.4 Algorithme DMC	13
1.4.1 Modèle de prédiction	13
1.4.2 Synthèse d'un correcteur DMC	16
1.4.3 Evaluation des performances de l'algorithme DMC	16
1.4.4 Etude de l'influence des paramètres de réglage d'une commande prédictive	19
1.5 Stabilité de la commande prédictive	22
1.6 Etat de l'art sur la commande prédictive	24
1.7 Conclusion	25

Chapitre 2. Optimisation Globale dans \mathfrak{R}	26
2.1 Introduction	26
2.2 Formulation mathématique d'un problème d'optimisation	27
2.3 Classification des problèmes d'optimisation dans \mathfrak{R}	28
2.4 Conditions pour un minimum	29
2.5 Méthodes de résolution des problèmes d'optimisation	30
2.6 Classifications des méthodes d'optimisation	32
2.7 Intérêt des méthodes globales	34
2.8 Méthodes d'optimisation globales et commande prédictive	37
2.9 Conclusion	38
Chapitre 3. Méthodes d'optimisation globale : Alienor et Essaim de Particules	40
3.1 Introduction	40
3.2 Méthode d'Alienor	41
3.3 Méthode d'essaim de particules	45
3.3.1 Principe de la méthode d'essaim de particules	46
3.3.2 Algorithme de la méthode d'essaim de particules	47
3.4 Application à l'identification d'un système non linéaire	50
3.4.1 Méthode du modèle	52
3.4.2 Modèle du système non linéaire à identifier	54
3.5 Conclusion	56
Chapitre 4. Commande Prédictive Non linéaire Basée sur l'Optimisation Globale	58
4.1 Introduction	58
4.2 Etat de l'art de la commande prédictive non linéaire	59
4.3 Commande prédictive non linéaire	61
4.3.1 Commande prédictive basée sur la méthode d'Alienor	63
4.3.2 Commande prédictive basée sur la méthode d'essaim de particules	66
4.4 Conclusion	69
Conclusion Générale	74

Liste des figures

1.1 Principe de la commande prédictive	07
1.2 Stratégie de commande prédictive	08
1.3 Réponse indicielle du système	17
1.4 Poursuite et rejet de perturbation	18
1.5 Evolution de la commande	18
1.6 Influence de l'horizon de commande	20
1.7 Influence de l'horizon de prédiction	21
2.1 Evolution de la fonction objectif du problème d'optimisation (2.24)	36
2.2 Evolution de la fonction objectif du problème d'optimisation (2.25)	36
2.3 Fonction objectif et son enveloppe convexe	37
3.1 Exploration globale du plan paramétrée par α (spirale d'Archimède)	42
3.2 Schéma de principe du déplacement d'une particule	47
3.3 Principe de la méthode du modèle	53
4.1 Structure générale de la commande prédictive non linéaire	62
4.2 Méthode d'Alienor : cas sans contrainte sur la variable de commande $u(k)$	70
4.3 Méthode d'Alienor : cas avec contrainte sur la variable de commande $u(k)$ ($ u(k) \leq 1,25$)	71
4.4 Méthode d'essaim de particules : cas sans contrainte sur la variable de commande $u(k)$	72
4.5 Méthode d'essaim de particules : cas avec contrainte sur la variable de commande $u(k)$	73

Liste des tableaux

1.1 Version commercialisés des correcteurs prédictifs	23
2.1 Classification des problèmes d'optimisation	28
4.1 Temps moyen pour le calcul de la commande	68

Introduction Générale

La commande prédictive constitue actuellement l'approche de commande la plus indiquée pour la commande des systèmes complexes dans le milieu industriel. L'intérêt est que des spécifications de fonctionnement ainsi que des contraintes d'exploitation (sécurité de fonctionnement des biens, des personnels et de l'environnement, qualité des produits), inévitables pour la plupart des systèmes, peuvent être conjointement traitées dans l'élaboration de la loi de commande (Flaus, 1994 ; Camacho et Bordons, 2008 ; Findeisen et al. 2007). De plus, la mise en œuvre d'une commande prédictive est relativement simple et il en va de même pour les outils théoriques nécessaires à leur étude, ce qui justifie sa popularité en utilisation industrielle (Henson, 1998). Aussi, l'évolution de l'informatique sur le plan matériel et logiciel a propulsé la commande prédictive pour occuper une place prépondérante dans les milieux académique et industriel (Martinsen, 2002).

La commande prédictive est une commande basée sur le modèle, par conséquent avoir de bonnes performances est lié directement à la précision et la complexité du modèle utilisé dans l'algorithme de commande pour la prédiction des sorties futures du système (Huang, 2008; Magni et al. 2009). Sur ce plan, l'existence des calculateurs numériques avec des vitesses de traitement vertigineuses, et des capacités de stockage énormes a largement contribué au succès de la commande prédictive.

Dans une stratégie de commande prédictive, les objectifs de commande sont spécifiés par un critère à minimiser et des contraintes à imposer sur les variables d'état, de commandes et de sorties (Flaus, 1994 ; Huang et Kadali, 2008 ; Camacho et Bordons, 2008). Par conséquent, la commande à appliquer, à chaque instant d'échantillonnage, est obtenue en résolvant un

problème d'optimisation avec contraintes en un temps inférieur à la période d'échantillonnage (Cannon, 2004 ; Tatjewski, 2007).

La nature du problème d'optimisation dépend généralement du type du modèle utilisé pour la prédiction puisque le critère est généralement quadratique et exprime des objectifs de poursuite et de minimisation d'énergie (Huang et Kadali, 2008 ; Camacho et Bordons, 2008). Lorsque le modèle est linéaire, la solution analytique du problème d'optimisation existe et simple à calculer et les performances seront meilleures puisque l'optimum global est atteint (Cutler, 1979 ; Clarke, 1987 ; Flaus, 1994 ; Huang et Kadali, 2008).

Par contre l'utilisation d'un modèle non linéaire, bien que les prédictions seront précises, mais la nature du problème d'optimisation est non linéaire dont la complexité est lié directement à la précision du modèle. Avoir un problème d'optimisation non linéaire, à résoudre à chaque période d'échantillonnage, constitue une tâche ardue et empêche l'application de la commande prédictive à des systèmes non linéaires (Cannon, 2004 ; Findeisen, 2007 ; Magni et al., 2009).

En effet, dans une stratégie de commande prédictive, les performances seront meilleures si l'optimum global du critère est atteint avec une très grande précision, et la convergence est assurée en un temps inférieur à la période d'échantillonnage (Long et al., 2006). Par conséquent, l'algorithme d'optimisation doit posséder les propriétés de convergence et de rapidité (Cannon, 2004).

La théorie de l'optimisation offre une panoplie de méthodes d'optimisation locales et globales (Horst et Pardalos, 1995). Dans une commande prédictive, pour avoir de bonnes performances, les méthodes d'optimisation intéressantes sont les méthodes globales, c'est-à-dire les méthodes permettant de localiser l'optimum global du critère contrairement aux méthodes locales qui localise l'optimum local (Cannon, 2004 ; Long et al., 2006). Les méthodes d'optimisation globale, proposées dans la littérature, sont scindées en trois classes (Berthiau et Siarry, 2001): les méthodes exactes pour lesquelles la convergence vers l'optimum globale est assurée au prix d'un temps de calcul important, les méthodes heuristiques caractérisées par une convergence asymptotique et rapide, et les méthodes hybrides qui combinent des algorithmes exactes et heuristiques pour fusionner leurs avantages et avoir des algorithmes rapides.

Parmi les méthodes d'optimisation globale qui ont prouvé leurs efficacités, on retrouve la méthode d'Alienor (Cherruault, 1991; Ziadi et al. 2001 ; Cherruault and Mora, 2005), qui est une méthode exacte, et la méthode d'essaim de particules qui est une méthode heuristique (Van Den Bergh, 2002 ; Poli et al. 2007). La méthode d'Alienor permet de ramener, à l'aide

d'une transformation réductrice, un problème d'optimisation à plusieurs variables de décision à un problème d'optimisation à une seule variable de décision ce qui permet de localiser facilement l'optimum globale et avec une précision souhaitée (Cherruault and Mora, 2005). La méthode d'essaim de particules est caractérisée par un algorithme simple à comprendre et à programmer. Le principe de la méthode repose sur un ensemble de particules qui représente un ensemble de solutions admissibles au problème d'optimisation (Poli, et al. 2007). Les particules sont initialement disposées de manière aléatoire et vont se déplacer dans le domaine de solutions. Chaque particule dispose d'une mémoire concernant sa meilleure solution visitée ainsi que la capacité de communiquer avec les particules de son voisinage. A partir de ces informations, la particule va suivre une tendance faite, d'une part, de sa volonté à retourner vers sa solution optimale, et d'autre part, de son mimétisme par rapport aux solutions trouvées dans son voisinage. A partir des solutions initiales, l'ensemble des particules va converger vers l'optimum global.

Dans ce mémoire, on s'intéresse à étudier l'apport des méthodes d'optimisation globale dans une stratégie de commande prédictive d'un système non linéaire. L'objectif du travail consiste à appliquer les méthodes d'optimisation globale d'Alienor et d'essaim de particules pour la résolution du problème d'optimisation non linéaire d'une commande prédictive non linéaire, puis de comparer leurs performances. L'étude se limite au système non linéaire caractérisé par une dynamique lente, et à la poursuite de consigne en présence des contraintes, de type boîtes, sur la variable de commande. Notons que la commande des systèmes rapides constitue un domaine de recherche ouvert, c'est pourquoi la commande prédictive a fait ses preuves dans le cas des systèmes caractérisés par une très grande inertie, par exemple les procédés chimiques et thermiques.

L'organisation de ce mémoire est la suivante :

Le **premier chapitre** est consacré à la commande prédictive dans lequel nous allons présenter le principe d'un algorithme de commande prédictive, ses différents éléments, et son réglage, puis pour illustrer le principe d'un algorithme prédictif, nous abordons de manière détaillée l'algorithme *Dynamic Matrix Control* (DMC) et on présente un exemple d'application avec des résultats de simulation pour montrer l'influence des paramètres de réglage sur les performances obtenues. Le chapitre se termine par un état de l'art sur la commande prédictive.

Dans le **deuxième chapitre**, on s'intéresse à l'optimisation globale dans l'ensemble des nombres réels. Le chapitre commence par des notions de base de la théorie de l'optimisation.

Dans la première partie, après avoir introduit la formulation mathématique d'un problème d'optimisation, nous présentons la classification des problèmes d'optimisation suivie des conditions pour un minimum. Dans la deuxième partie, on présente les différentes classes de méthodes d'optimisation, et on indique l'intérêt des méthodes globales. Dans la fin du chapitre, on mentionne le lien fort existant entre l'optimisation globale et la commande prédictive non linéaire.

Le **troisième chapitre** est réservé aux méthodes d'optimisation globale. Dans ce chapitre, on présente particulièrement la méthode exacte d'Alienor et la méthode heuristique d'essaim de particules. Pour illustrer et comparer ces deux méthodes, une application concernant l'identification des paramètres d'un système non linéaire en utilisant la méthode du modèle est présentée.

Au **quatrième chapitre**, après avoir présenté un état de l'art sur la commande prédictive non linéaire, les méthodes d'Alienor et d'essaim de particules seront adoptées dans une stratégie de commande prédictive d'un système non linéaire monovarié. Les spécifications exigées imposent d'assurer la poursuite de consigne en présence des contraintes, du type boîtes, sur la variable de commande. L'objectif principal est d'évaluer l'apport de chaque méthode d'optimisation, puis de comparer leurs performances en présence et en absence de la contrainte sur la variable de commande.

La fin du mémoire est réservée à la conclusion sur l'ensemble des applications réalisées et aux perspectives de continuité.

Chapitre 1

Généralités sur la Commande Prédictive

1.1 Introduction

Ces dernières années la commande prédictive est largement utilisée dans le domaine industriel et a été appliquée avec succès pour différentes applications. Le terme *commande prédictive* ne désigne pas une stratégie de commande spécifique mais un ensemble d'algorithmes qui utilisent explicitement le modèle du système dans un problème d'optimisation, à résoudre, pour déterminer la commande à appliquer.

Le modèle du système est essentiellement utilisé pour deux tâches (Huang, et al. 2008) : la *prédiction* du comportement dynamique future du système et le calcul de l'action correctrice appropriée pour assurer la poursuite de la consigne imposée.

La commande prédictive basée sur le modèle présente des avantages qui ont fait d'elle une approche de commande attractive. Parmi ces avantages, on peut citer (Flaus 1994 ; Huang et al. 2008) :

- elle permet de respecter les différentes contraintes sur les états, les commandes et les sorties,

- elle permet d’assurer la poursuite pour certaines consignes tout en maintenant d’autres dans des couloirs bien spécifiés,
- elle évite les variations excessives pour les variables de commandes,
- elle peut s’appliquer à des systèmes avec ou sans retard,
- son réglage est aisé et son principe est intuitif,
- elle est assez robuste aux erreurs de modèle.

L’intérêt de la commande prédictive est multiple :

- comparés aux techniques classiques, elle est extrêmement performante si le procédé réagit avec un certain retard ou si les changements de consignes sont connus à l’avance,
- comparés aux techniques plus sophistiquées réalisant la minimisation d’un critère à un pas, les techniques prédictives sont nettement robustes, en particulier lorsque la structure du modèle utilisée pour décrire le procédé est mal connue.
- elle peut être utilisée en conservant un seul paramètre libre pour le réglage.

Ce chapitre est consacré à la commande prédictive basée sur le modèle où il sera présenté le principe de la commande prédictive, la commande prédictive linéaire et non linéaire et un état de l’art sur les techniques de commande prédictives.

1.2 Principe de la commande prédictive

La commande prédictive est basée sur le principe de la Figure 1.1, elle est décrite par la stratégie suivante :

1. A chaque instant d’échantillonnage k , les sorties futures du système sont prédites sur un horizon de temps N_p , appelé horizon de prédiction, qui est relativement long par rapport à la vitesse d’évolution du procédé. Les prédictions $\hat{y}(k+j/k)$, pour $j=1,\dots,N_p$, sont réalisées en utilisant le modèle du système et dépendent non seulement du passé du système (les commandes et les sorties avant l’instant k), mais aussi des commandes futures $u(k+i)$, pour $i=0,\dots,N_p-1$, à déterminer et appliquer au système. Par conséquent, le modèle du système fait partie de l’algorithme de commande. Ainsi la prédiction ne va pas dépendre uniquement des sorties précédentes mais aussi de l’évolution envisagée dans le temps futur pour la variable de commande. Notons que plusieurs évolutions sont possibles pour la variable de commande.

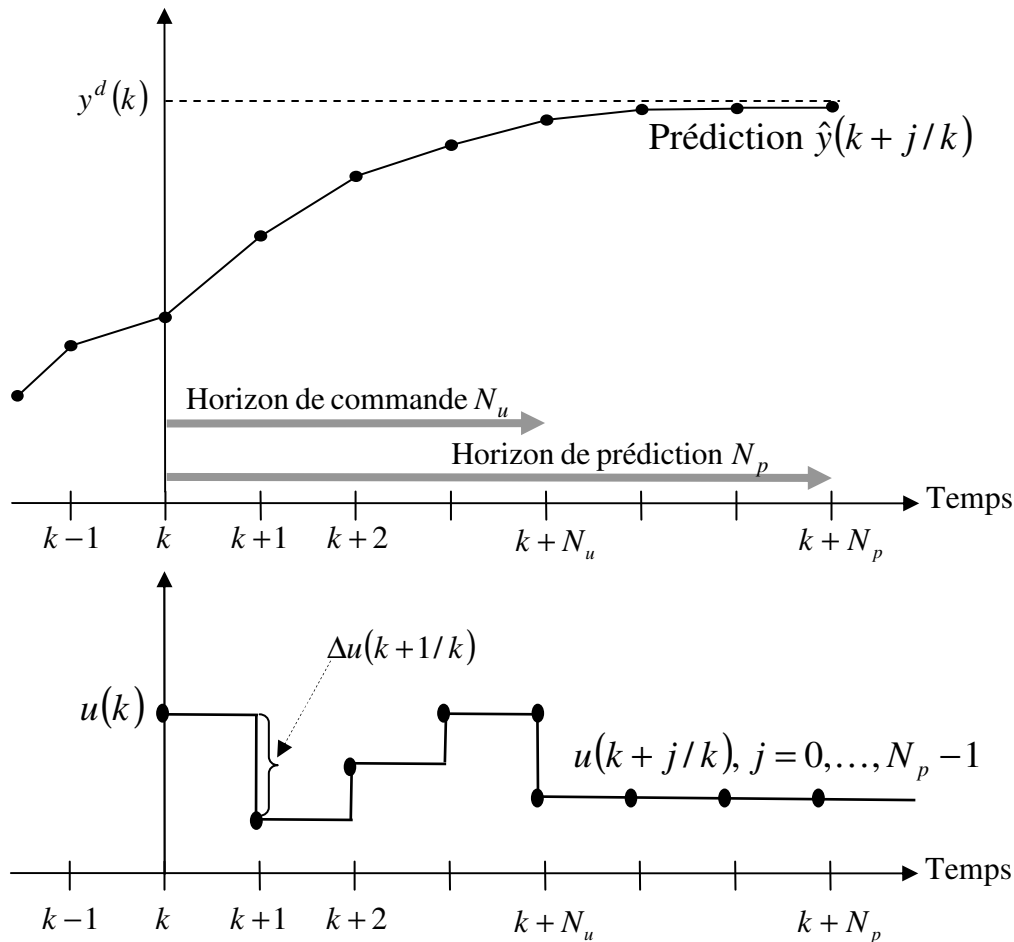


Figure 1.1 Principe de la commande prédictive.

2. La séquence de commande est déterminée en minimisant un critère de performances qui permet d'assurer la poursuite de la consigne désirée. Le critère est une fonction quadratique des erreurs entre les sorties prédites et la trajectoire de référence. L'effort de commande est généralement inclus dans le critère à minimiser. Ainsi, une solution explicite peut être facilement obtenue dans le cas où le critère est quadratique et le modèle du système ainsi que les contraintes sont linéaires, sinon une méthode d'optimisation numérique doit être appliquée. Notons que pour la structure de la loi de commande, certaines hypothèses peuvent être considérées, par exemple la commande reste constante après un certain instant d'échantillonnage $k \leq N_p$.
3. La solution déterminée par optimisation sera ensuite appliquée au système réel, mais seule sa valeur à l'instant présent k est réellement utilisée. A l'instant suivant $k+1$, la procédure complète est répétée. Ceci permet d'obtenir une valeur réactualisée pour la

commande, en fonction des mesures les plus récentes en utilisant le concept de l'horizon glissant.

1.3 Eléments d'une commande prédictive

Tous les algorithmes de la commande prédictive possèdent les mêmes éléments (Figure 1.2), et différentes options peuvent être considérées pour chaque élément, ce qui donne une multitude d'algorithmes. Ces éléments sont :

1. le modèle du système (pour la prédiction),
2. le critère de performances et
3. l'algorithme d'optimisation (pour déterminer la séquence de commande).

1.3.1 Modèle du système

La pierre angulaire d'une stratégie de commande prédictive est le modèle du système. Le modèle du système comprend généralement deux parties : le modèle du système et le modèle de perturbation. Le premier représente les relations entrée-sortie du système et le deuxième est souvent utilisé pour représenter les perturbations ou simplement pour approximer les erreurs de modélisation. Il existe différentes formes pour le modèle utilisé dans une commande prédictive, mais ils doivent être toujours de nature discrète puisque la commande prédictive est une commande numérique. Les formes couramment utilisées sont du type entrée-sortie, elles sont présentées ci-après.

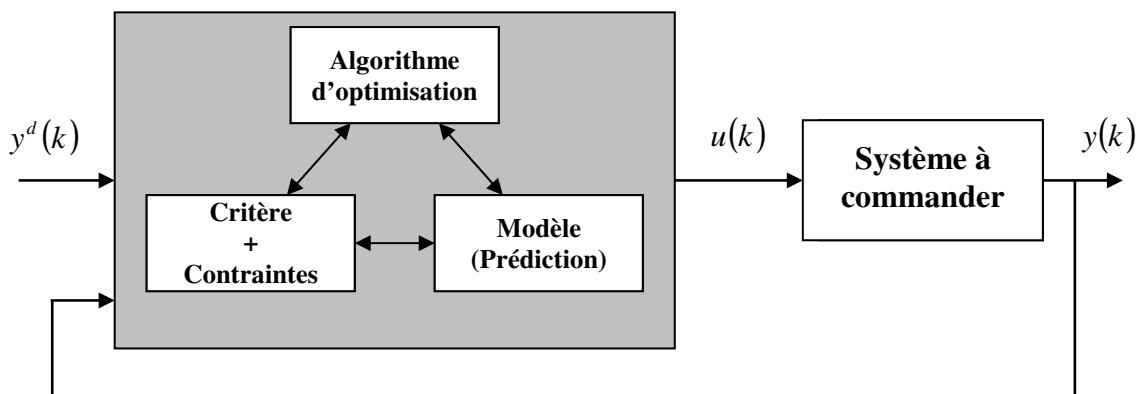


Figure 1.2 Stratégie de commande prédictive.

- Réponse impulsionnelle

Dans ce cas, la réponse du système est :

$$y(k) = \sum_{i=1}^{+\infty} r(i)u(k-i) \quad (1.1)$$

où $r(i)$ représentent les coefficients de la réponse impulsionnelle (les valeurs de la sortie, aux instants d'échantillonnage, lorsque l'entrée est une impulsion de Dirac). Souvent cette somme est tronquée et seules les N_s valeurs sont considérées puisque à partir de l'instant $N_s + 1$ la sortie est nulle. $N_s \times T_e$ représente le temps de réponse du système.

- Réponse indicielle

Dans ce cas, la réponse du système est :

$$y(k) = \sum_{i=1}^{+\infty} s(i) \Delta u(k-i) \quad (1.2)$$

où $s(i)$ représentent les coefficients de la réponse indicielle pour une entrée échelon unité, et $\Delta u(k) = u(k) - u(k-1)$.

Pour un système stable, les coefficients $s(i)$ sont constants après l'instant N_s . $N_s \times T_e$ représente le temps de réponse du système.

Comme les coefficients de la réponse impulsionnelle peuvent être considérés comme la différence entre deux coefficients successifs de la réponse indicielle, alors les relations suivantes sont vérifiées :

$$r(i) = s(i) - s(i-1) \quad (1.3)$$

$$s(i) = \sum_{j=1}^{+\infty} r(j) \quad (1.4)$$

- Equation aux différences

Dans ce cas, la sortie du système est :

$$y(k) = F(y(k-1), \dots, y(k-n_y), u(k-1), \dots, u(k-n_u)) \quad (1.5)$$

où n_y et n_u sont des entiers naturels. La fonction $F(\cdot)$ peut être linéaire ou non linéaire selon la nature du système.

Remarque 1.1

D'autres formes du modèle peuvent être aussi utilisées parmi lesquels on peut citer la fonction de transfert discrète, le modèle d'état discret, modèle de Volterra, modèle neuronal, modèle flou. Dans ce mémoire, l'étude est limitée aux modèles présentés dans cette sous-section.

1.3.2 Prédiction

Dans la commande prédictive, un modèle du système n'est pas utilisé pour la conception de la loi de commande, mais il est utilisé pour la prédiction des sorties futures du système. Ces prédictions seront utilisées par la suite pour la détermination de la séquence de la variable de commande en résolvant un problème d'optimisation.

Comme exemple, considérons le modèle suivant :

$$y(k) = -a y(k-1) + b \Delta u(k-1) \quad (1.6)$$

Les deux premières prédictions futures de la sortie sont données comme suit :

$$\hat{y}(k+1/k) = -a y(k) + b \Delta u(k) \quad (1.7)$$

$$\hat{y}(k+2/k) = -a \hat{y}(k+1/k) + b \Delta u(k+1)$$

$$\hat{y}(k+2/k) = -a(-a y(k) + b \Delta u(k)) + b \Delta u(k+1)$$

$$\hat{y}(k+2/k) = a^2 y(k) - a b \Delta u(k) + b \Delta u(k+1) \quad (1.8)$$

Les autres prédictions sont déterminées de la même manière en utilisant le modèle (1.6).

Sous forme matricielle, les deux premières prédictions sont données comme suit :

$$\begin{bmatrix} \hat{y}(k+1/k) \\ \hat{y}(k+2/k) \end{bmatrix} = \begin{bmatrix} -a \\ a^2 \end{bmatrix} y(k) + \begin{bmatrix} b & 0 \\ -ab & b \end{bmatrix} \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \end{bmatrix} \quad (1.9)$$

Les prédictions (1.9) sont composées de deux termes. Le second terme du côté droit de l'équation (1.9) dépend des entrées futures. Par contre le premier terme dépend seulement des sorties précédentes ou également des entrées précédentes selon le système. Ainsi, les prédictions peuvent être décomposées en général en deux parties :

$$\hat{y}(k+j/k) = \hat{y}_{\text{libre}}(k+j) + \hat{y}_{\text{forcée}}(k+j) \quad (1.10)$$

La réponse libre, $\hat{y}_{\text{libre}}(k+j)$, correspond à la prédiction de la sortie lorsque la commande est maintenue constante à sa valeur actuelle $u(k)$ le long de l'horizon de prédiction N_p . Par contre la réponse forcée, $\hat{y}_{\text{forcée}}(k+j)$, correspond à la prédiction de la sortie due aux actions futures $u(k+j)$, $j=1, \dots, N_p$.

1.3.3 Critère de performance (fonction objectif)

Pour déterminer la loi de commande, les algorithmes de la commande prédictive utilisent différentes formes pour le critère. Généralement, le but recherché est d'assurer la poursuite de

la consigne désirée $y^d(k+j)$ avec un minimum d'effort $u(k+j-1)$. Ainsi, le critère utilisé le plus souvent prend la forme quadratique car différentiable :

$$J = \sum_{j=N_r}^{N_p} [y^d(k+j) - \hat{y}(k+j/k)]^T Q_j [y^d(k+j) - \hat{y}(k+j/k)] + \sum_{j=0}^{N_u-1} u(k+j-1)^T R_j u(k+j-1) \quad (1.11)$$

où les matrices Q_j et R_j sont symétriques et définies positives. Leur choix caractérise l'importance relative que nous souhaitons donner aux diverses composantes du critère à minimiser. Elles peuvent aussi être constantes. Les paramètres N_r , N_p et N_u sont les paramètres de réglage du correcteur prédictif et représentent respectivement l'instant du début de la prédiction, l'horizon de prédiction maximal et l'horizon de commande. L'horizon de prédiction N_p représente la longueur de l'intervalle de temps futur sur lequel on cherche à minimiser le critère. L'horizon de commande N_u représente le nombre de variations de la commande que l'on autorise dans le futur avant qu'elle ne soit maintenue à une valeur constante.

Pour les algorithmes qui utilisent un modèle de convolution tronqué, on utilise un autre paramètre N_s qui représente la longueur de la réponse indicielle ou impulsionnelle.

Il faut aussi choisir convenablement la période d'échantillonnage T_e puisque le correcteur prédictif est de type discret. Cependant ce problème n'est pas spécifique à la commande prédictive, mais à toute commande numérique. Le seul aspect par lequel le choix de la période d'échantillonnage peut être lié de façon étroite à la synthèse du correcteur prédictif est celui de la longueur N_s du modèle de convolution. Si on définit le temps de réponse t_r comme le temps à partir duquel la réponse du système a atteint 99% de la valeur finale, on doit vérifier :

$$N_s \times T_e = t_r \quad (1.12)$$

pour tenir compte de la dynamique du système. Dans le cas où la période d'échantillonnage T_e a été choisie petite (cas d'un système rapide), la longueur du modèle de convolution peut devenir excessive et entraîner des calculs importants.

L'influence de ces différents paramètres sur les performances du système sera abordée dans le cas de l'algorithme *Dynamic Matrix Control* (DMC), qui fera l'objectif de la Section 1.4.4, et sera illustré par un exemple d'application. Notons que les conclusions concernant l'influence des différents paramètres pourront facilement être transposées pour un autre algorithme même dans le cas de la commande prédictive non linéaire.

1.3.4 Contraintes

Les contraintes caractérisent en général les limitations physiques sur la commande, sur l'état ou sur la sortie du système. Elles sont introduites pour éviter des changements brusques pour la commande. Pour certains systèmes, des contraintes sur les sorties (prédites) doivent être respectées pour des raisons économiques ou de sécurité.

Généralement, les contraintes sont instantanées et s'expriment par des inégalités de la forme :

$$u_{\min} \leq u(k) \leq u_{\max} \quad (1.13)$$

$$y_{\min} \leq \hat{y}(k + j/k) \leq y_{\max}, N_r \leq j \leq N_p, \forall k \quad (1.14)$$

Dans ce mémoire, on considère que les contraintes sur la commande de type (1.13).

1.3.5 Loi de commande

Les actions de commande, $u(k+i)$, sont calculées en minimisant le critère (1.11). Ainsi, on commence d'abord par la détermination des prédictions en utilisant le modèle du système (voir l'exemple de la sous-section 1.3.2). Ces dernières seront forcément fonction des actions de commande $u(k+i)$. Ainsi, en substituant ces prédictions dans le critère J à minimiser, donnée par la relation (1.11), et en résolvant le système d'équations algébriques obtenu en imposant le gradient ∇J par rapport aux actions de commande $u(k+i)$ égale à zéro, c'est-à-dire :

$$\nabla_{(u(k), u(k+1), \dots, u(k+N_u))} J(u(k), u(k+1), \dots, u(k+N_u)) = \begin{bmatrix} \frac{\partial J(u(k), u(k+1), \dots, u(k+N_u))}{\partial u(k)} \\ \frac{\partial J(u(k), u(k+1), \dots, u(k+N_u))}{\partial u(k+1)} \\ \vdots \\ \frac{\partial J(u(k), u(k+1), \dots, u(k+N_u))}{\partial u(k+N_u)} \end{bmatrix} = 0 \quad (1.15)$$

on détermine les actions de commandes $u^*(k+i)$ optimales. Rappelons que d'après le principe de la commande prédictive seul l'action de commande $u^*(k)$ sera appliquée au système, et la procédure sera répétée à l'instant d'échantillonnage suivant.

Annuler le gradient du critère J est une procédure d'optimisation. Pour le système d'équations algébriques (1.15) une solution analytique peut être obtenue, dans le cas contraire une méthode numérique (itérative) doit être utilisée dont le temps de convergence ne doit pas

dépasser une période d'échantillonnage pour que l'action optimale $u^*(k)$ soit disponible à l'instant k .

Lorsqu'une méthode numérique est utilisée, l'obtention de la solution du problème d'optimisation suivant :

$$\min_{u(k), u(k+1), \dots, u(k+N_u)} J(u(k), u(k+1), \dots, u(k+N_u)) \quad (1.16)$$

n'est pas nécessairement garantie surtout si le problème d'optimisation est non linéaire, et la difficulté s'accroît davantage en présence des contraintes sur la commande ou/et sur la sortie commandée. De plus, assurer la convergence en un temps inférieur à T_e représente une autre contrainte à prendre en considération, et la plupart des méthodes numériques peuvent être piégées dans un minimum local pour (1.16) au lieu de minimum global, d'où la nécessité d'utiliser des algorithmes d'optimisation globale rapides.

Notons que la solution analytique peut être obtenue seulement dans le cas du modèle linéaire avec contraintes linéaires, par exemple dans le cas de l'algorithme *Dynamic Matrix Control* (cet algorithme sera étudié dans la section suivante). Pour les autres cas, la solution numérique est inéluctable, par conséquent le choix de la méthode d'optimisation joue un rôle du premier plan dans une stratégie de commande prédictive. Le chapitre 2 s'inscrit dans cette optique, il sera consacré à l'optimisation globale.

Dans la section suivante, nous présenterons un des algorithmes de la commande prédictive pour illustrer le principe de cette dernière. Il s'agit de l'algorithme DMC (*Dynamic Matrix Control*).

1.4 Algorithme DMC

L'algorithme DMC a été initialement développé par Cutler et Ramaker (1979) de la société Shell Oil Co vers la fin des années 70. Cet algorithme est largement accepté et utilisé par les industriels, en particulier dans le secteur des hydrocarbures et de la pétrochimie. Dans cette section, nous présentons une version simplifiée de cet algorithme dans le cas où le modèle est fourni sous forme de réponse à un échelon. Cette formulation est généralement la plus choisie car elle permet la compréhension intuitive du fonctionnement de la commande prédictive. Néanmoins des développements équivalents peuvent être conduits pour le modèle de réponse impulsionnelle où fonction de transfert conduisant respectivement au *Model Algorithmic Control* (MAC) et au *Generalized Predictive Control* (GPC).

1.4.1 Modèle de prédiction

Le modèle utilisé pour la prédiction dans l'algorithme de commande prédictive DMC est le modèle réponse indicielle. Considérons la réponse indicielle du système

$$y(k) = \sum_{i=1}^{+\infty} s_i \Delta u(k-i) \quad (1.17)$$

qu'on peut mettre sous la forme suivante :

$$y(k) = \sum_{i=1}^{N_s} s_i \Delta u(k-i) + \sum_{i=N_s+1}^{+\infty} s_i \Delta u(k-i) \quad (1.18)$$

$$y(k) = \underbrace{\sum_{i=1}^{N_s} s_i \Delta u(k-i)}_{\text{Réponse indicielle finie}} + Z(k) \quad (1.19)$$

Ainsi, on remarque bien que la réponse indicielle infinie (1.18) du système fait intervenir une réponse indicielle finie qu'on peut exploiter pour l'étape de prédiction, c'est-à-dire :

$$\hat{y}(k+j/k) = \sum_{i=1}^{N_s} s_i \Delta u(k+j-i) + Z(k+j) \quad (1.20)$$

Comme cela a été indiqué précédemment, la réponse du système prédite par le modèle peut être divisée en deux parties : libre et forcée. Ainsi, l'équation (1.20) peut être décomposée en deux termes : le premier terme contient les actions de commande futures ($\Delta u(k), \Delta u(k+1), \dots$), et le second terme contient les actions de commande passées ($\Delta u(k-1), \Delta u(k-2), \dots$), ce qui donne

$$\hat{y}(k+j/k) = \sum_{i=1}^j s_i \Delta u(k+j-i) + \sum_{i=j+1}^{N_s} s_i \Delta u(k+j-i) + Z(k+j) \quad (1.21)$$

Cette dernière équation peut être arrangée pour faire apparaître les deux termes de la sortie (forcée et libre) comme suit :

$$\hat{y}(k+j/k) = \sum_{i=1}^j s_i \Delta u(k+j-i) + y^*(k+j) \quad (1.22)$$

où le terme

$$y^*(k+j) = \sum_{i=j+1}^{N_s} s_i \Delta u(k+j-i) + Z(k+j) \quad (1.23)$$

représente le terme libre :

$$\hat{y}_{\text{libre}}(k+j) = y^*(k+j) \quad (1.24)$$

Le terme restant représente le terme forcé qui est l'évolution de la réponse du système à partir de l'instant k dû à l'application des actions de commande futures. Ainsi,

$$\hat{y}_{\text{forcée}}(k+j) = \sum_{i=1}^j s_i \Delta u(k+j-i) \quad (1.25)$$

Le critère considéré dans un algorithme DMC consiste à minimiser la différence entre la consigne désirée et la sortie prédite par le modèle le long de l'horizon de prédiction N_p . Les sorties prédites sont données comme suit :

$$\begin{aligned} \hat{y}(k+1/k) &= y^*(k+1) + s_1 \Delta u(k) \\ \hat{y}(k+2/k) &= y^*(k+2) + s_2 \Delta u(k) + s_1 \Delta u(k+1) \\ \hat{y}(k+3/k) &= y^*(k+3) + s_3 \Delta u(k) + s_2 \Delta u(k+1) + s_1 \Delta u(k+2) \\ &\dots \\ \hat{y}(k+N_u/k) &= y^*(k+N_u) + s_{N_u} \Delta u(k) + s_{N_u-1} \Delta u(k+1) + \dots + s_1 \Delta u(k+N_u-1) \\ &\vdots \\ \hat{y}(k+N_p/k) &= y^*(k+N_p) + s_{N_p} \Delta u(k) + s_{N_p-1} \Delta u(k+1) + \dots + s_1 \Delta u(k+N_p-1) \end{aligned} \quad (1.26)$$

Notons que $\Delta u(k+N_u) = \Delta u(k+N_u+1) = \dots = \Delta u(k+N_p-1) = 0$ car l'action de commande est maintenue constante à partir de l'instant $k = N_u$.

Les prédictions (1.26) peuvent être écrites sous la forme matricielle comme suit :

$$\hat{y} = y^* + M \Delta u \quad (1.27)$$

Avec :

$$\hat{y} = \begin{bmatrix} \hat{y}(k+1/k) \\ \hat{y}(k+2/k) \\ \vdots \\ \hat{y}(k+N_p/k) \end{bmatrix}, y^* = \begin{bmatrix} y^*(k+1) \\ y^*(k+2) \\ \vdots \\ y^*(k+N_p) \end{bmatrix}, \Delta u = \begin{bmatrix} \Delta u(k) \\ \Delta u(k+1) \\ \vdots \\ \Delta u(k+N_u-1) \end{bmatrix}$$

et la matrice

$$M = \begin{bmatrix} s_1 & 0 & 0 & \dots & 0 \\ s_2 & s_1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ s_{N_u} & s_{N_u-1} & s_{N_u-2} & \dots & s_1 \\ \vdots & \vdots & \vdots & \vdots & \vdots \\ s_{N_p} & s_{N_p-1} & s_{N_p-2} & \dots & s_{N_p-N_u+1} \end{bmatrix}$$

est appelée matrice dynamique (*Dynamic Matrix*).

1.4.2 Synthèse d'un correcteur DMC

Rappelons que l'objectif consiste à poursuivre une trajectoire, le long de l'horizon de prédiction, donnée comme suit :

$$y^d = \begin{bmatrix} y^d(k+1) \\ y^d(k+2) \\ \vdots \\ y^d(k+N_p) \end{bmatrix}$$

Ainsi, pour déterminer la séquence des commandes à appliquer, il suffit de résoudre, à chaque instant k , le problème d'optimisation suivant :

$$\begin{aligned} \min_{\Delta u} J(\Delta u) &= (y^d - \hat{y})^T Q (y^d - \hat{y}) \\ \text{sujet à :} & \end{aligned} \quad (1.28)$$

$$\hat{y} = y^* + M \Delta u$$

En substituant \hat{y} dans le critère J , il vient :

$$\min_{\Delta u} J(\Delta u) = (y^d - y^* - M \Delta u)^T Q (y^d - y^* - M \Delta u) \quad (1.29)$$

La solution optimale est obtenue comme suit :

$$\nabla_{\Delta u} J(\Delta u) = -2M^T Q^T (y^d - y^* - M \Delta u) = 0 \quad (1.30)$$

Ce qui donne

$$\Delta u = (M^T Q^T M)^{-1} M^T Q^T (y^d - y^*) \quad (1.31)$$

1.4.3 Evaluation des performances de l'algorithme DMC

Pour illustrer et évaluer les performances de l'algorithme DMC, nous proposons d'étudier la commande d'un système de deuxième ordre avec un retard important. Considérons le procédé décrit par la fonction de transfert :

$$G(s) = \frac{1}{(1+7s)(1+15s)} e^{-8s} \quad (1.32)$$

La réponse indicielle de ce système est donnée par la Figure 1.3. D'après cette dernière, nous pouvons prendre la longueur N_s du modèle de convolution supérieure à 80. Dans notre cas on a pris le double, c'est-à-dire $N_s = 160$. Pour déterminer la matrice dynamique M , cette

réponse est échantillonnée avec une période d'échantillonnage égale à 1s. Pour les paramètres de réglage de l'algorithme DMC, on considère que la prédiction commence après 8 périodes d'échantillonnage ($N_r = 8$), l'horizon de commande $N_u = 2$, et l'horizon de prédiction $N_p = 30$. Pour le critère à minimiser, la matrice de pondération $Q = I$.

Pour le problème de poursuite, la consigne désirée y^d est supposée constante et égale à 60. Pour le test de régulation (rejet de perturbation), on considère une perturbation d bornée qui affecte la sortie du système avec une dynamique d'un système du premier ordre c'est-à-dire :

$$G_d(s) = \frac{1}{0.2s + 1} \quad (1.33)$$

Par conséquent la sortie du système, en boucle ouverte est :

$$y(s) = G(s)u(s) + G_d(s)d(s) \quad (1.34)$$

Pour le test concernant le rejet de perturbation, on a appliqué une perturbation d'amplitude $d = 0,1$ à l'instant 50 s.

Les Figures 1.4, et 1.5 présentent les résultats de simulation obtenus. On constate que l'algorithme DMC assure la poursuite de la consigne et rejette parfaitement l'effet de la perturbation (Fig. 1.4). L'évolution de la commande (Fig. 1.5) est physiquement acceptable.

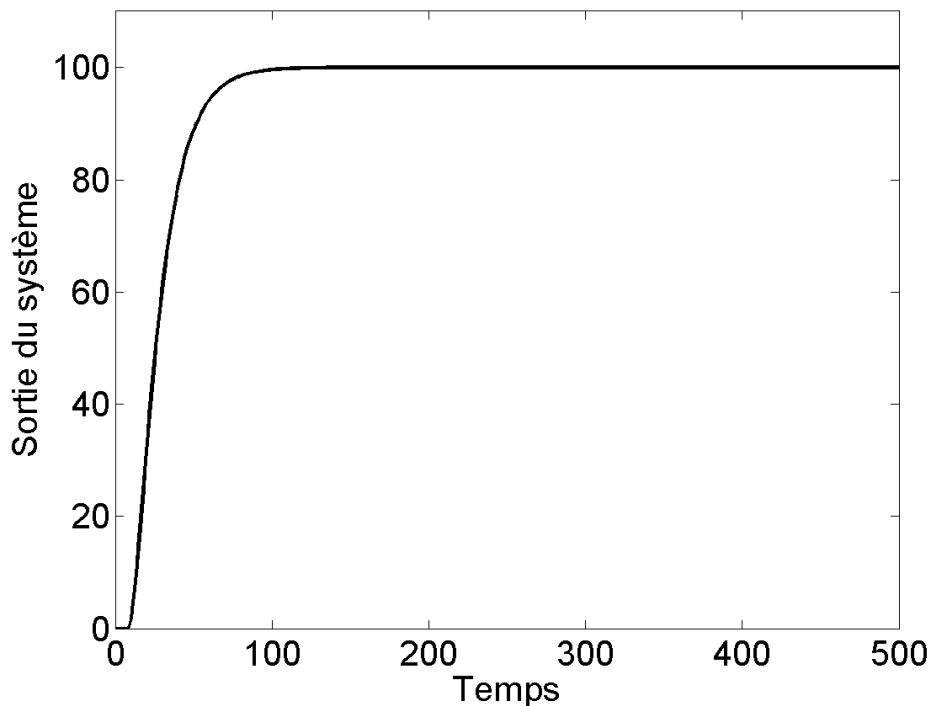


Figure 1.3 Réponse indicielle du système.

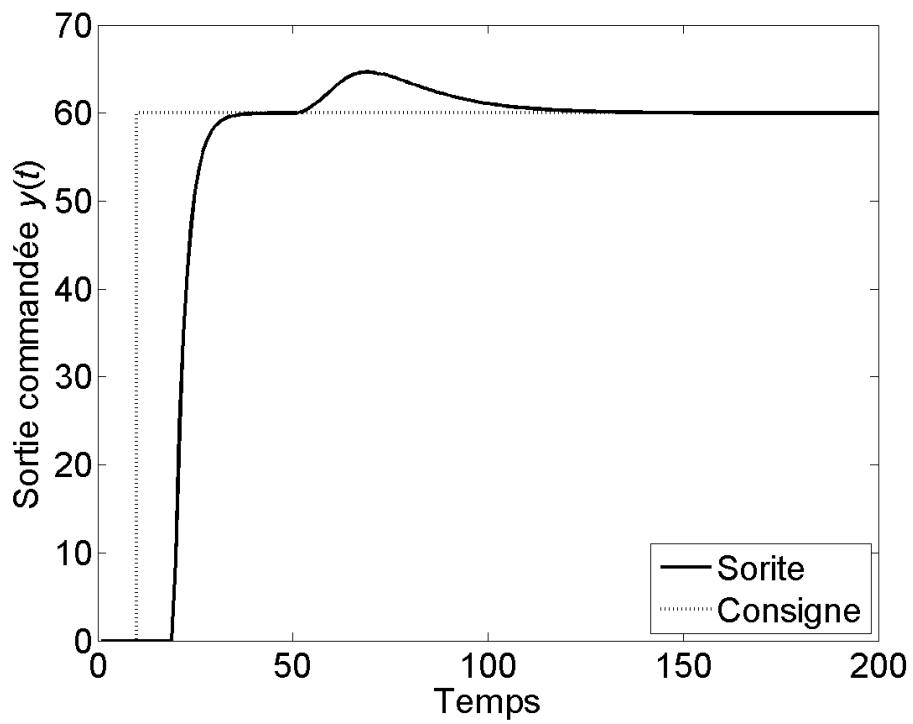


Figure 1.4 Poursuite et rejet de perturbation.

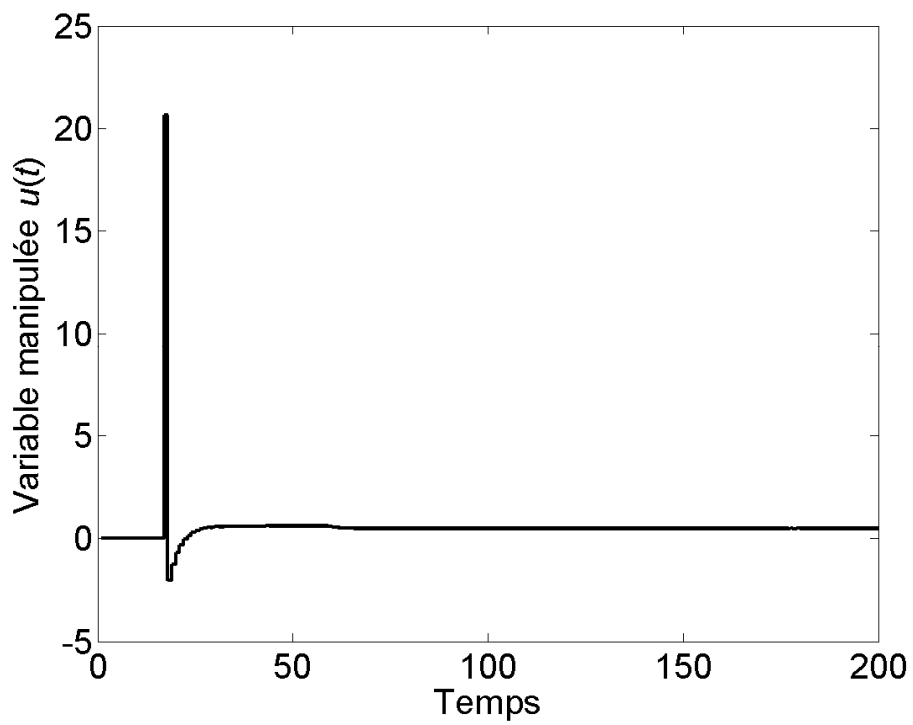


Figure 1.5 Evolution de la commande.

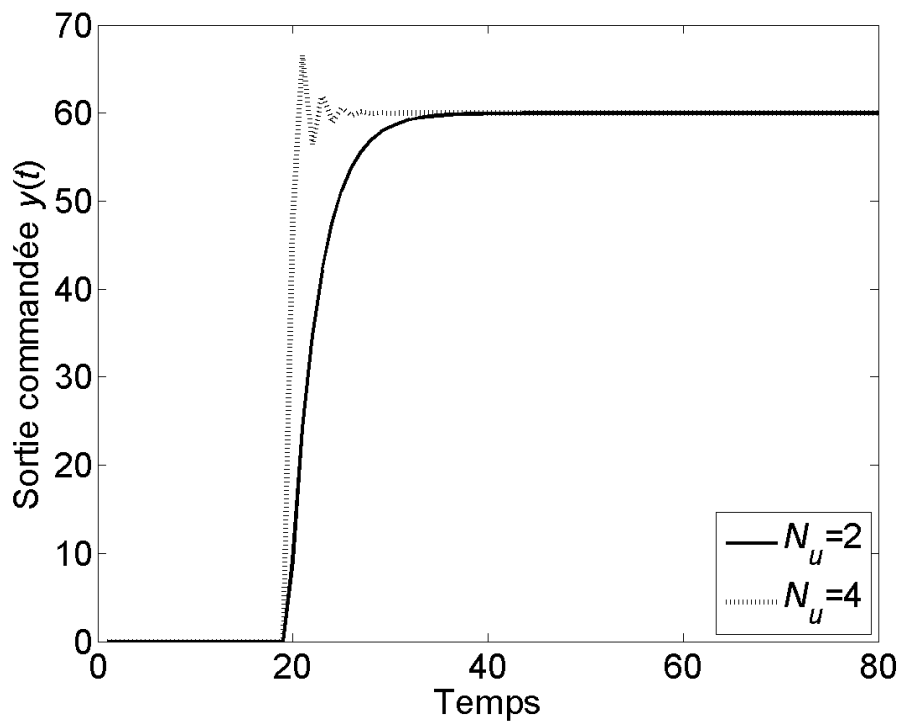
1.4.4 Etude de l'influence des paramètres de réglage d'une commande prédictive

En plus de la période d'échantillonnage, une commande prédictive est caractérisée par les paramètres de réglage suivant :

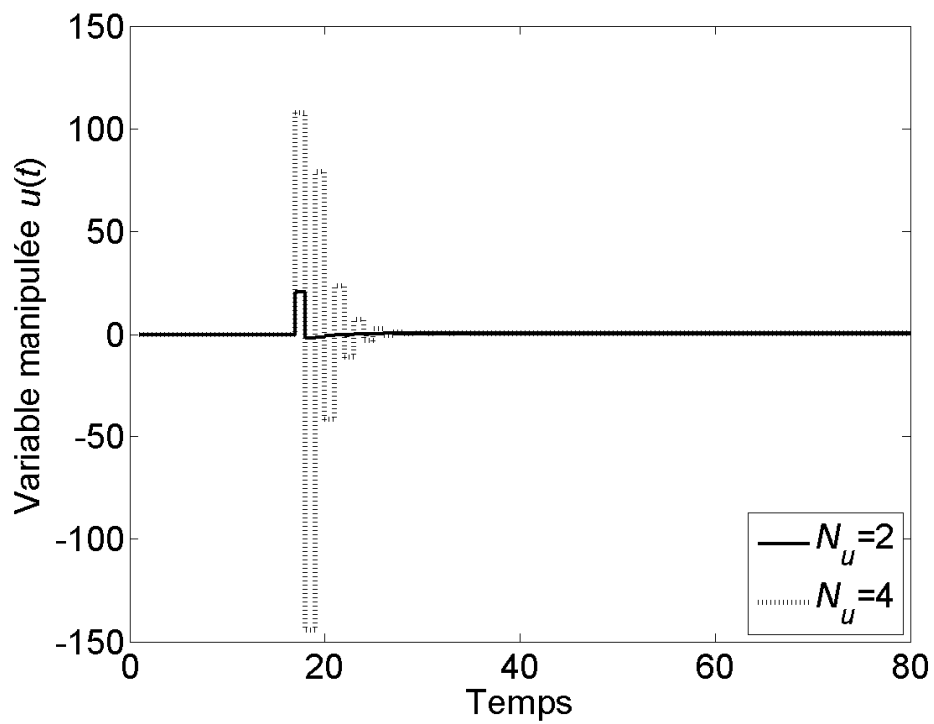
- Instant de *début de prédiction* N_r : ce paramètre est pris égale au retard du système τ . Si le système ne présente pas de retard, on prend $N_r = 1$.
- *horizon du système* N_s : il est déterminé selon la relation (1.12). Des valeurs entre 20 et 70 sont des recommandations typiques.
- *horizon de commande* N_u : il représente le nombre d'actions futures calculées par optimisation du critère pour réduire les erreurs de prédictions. La valeur de ce paramètre influe sur le temps de calcul de la solution optimale. L'augmentation de ce paramètre permet d'accélérer le système, mais en contre partie on va avoir des actions plus énergiques. Comme le principe de commande prédictive consiste à appliquer seulement la première commande, alors il est inutile de prendre une valeur supérieure à 5.
- *horizon de prédiction* N_p : il représente le nombre de prédictions utilisées dans le critère à optimiser. La valeur minimale doit être égale à $\tau + 1$ où τ désigne le temps de retard du système à commander. L'augmentation de N_p revient à considérer plus de prédictions, par conséquent la sortie sera bien amortie et les commandes seront de faibles amplitudes, mais leur traitement nécessite un temps de calcul important.

Ainsi, pour le réglage des paramètres d'une commande prédictive, on s'intéresse en particulier à l'ajustement de N_u et N_p . Pour illustrer l'influence de ces deux paramètres, nous proposons de reprendre l'exemple étudié précédemment. Pour montrer, l'influence de chaque paramètre, on fixe un des paramètres puis on considère deux valeurs pour l'autre paramètre. La perturbation d est considérée nulle.

La Figure 1.6 montre que l'augmentation de N_u , ($N_p = 30$), donne des réponses rapides (Fig. 1.6a), mais au prix d'une sollicitation plus importante de la commande (Fig. 1.6b). Pour l'influence de N_p , ($N_u = 2$), d'après la Figure 1.7, on note l'effet stabilisant pour la sortie (Fig. 1.7a), avec une commande moins énergique (Fig. 1.7b).

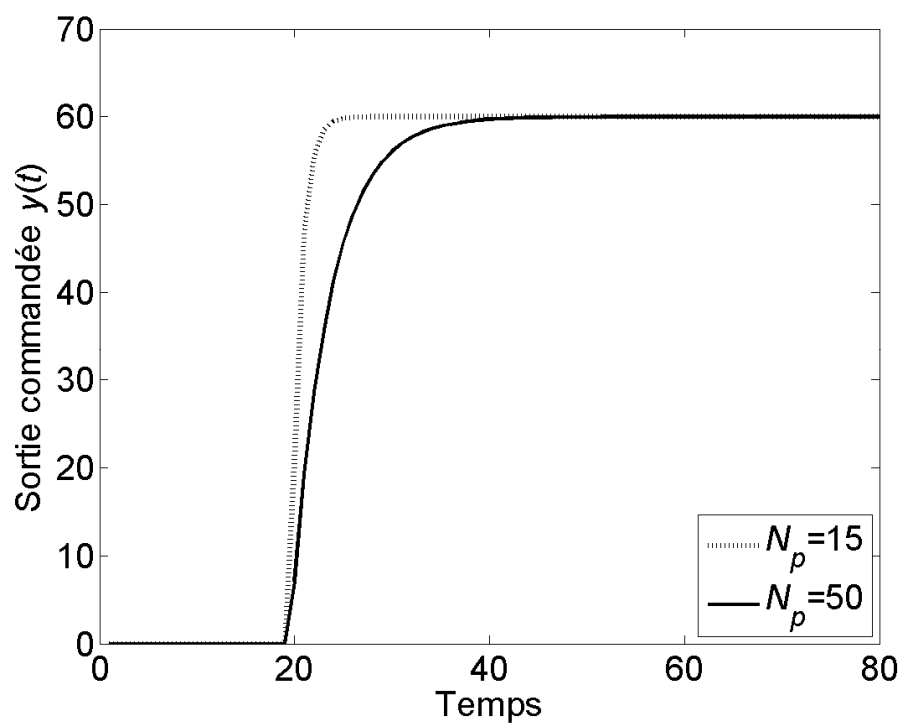


a) Evolution de la sortie.

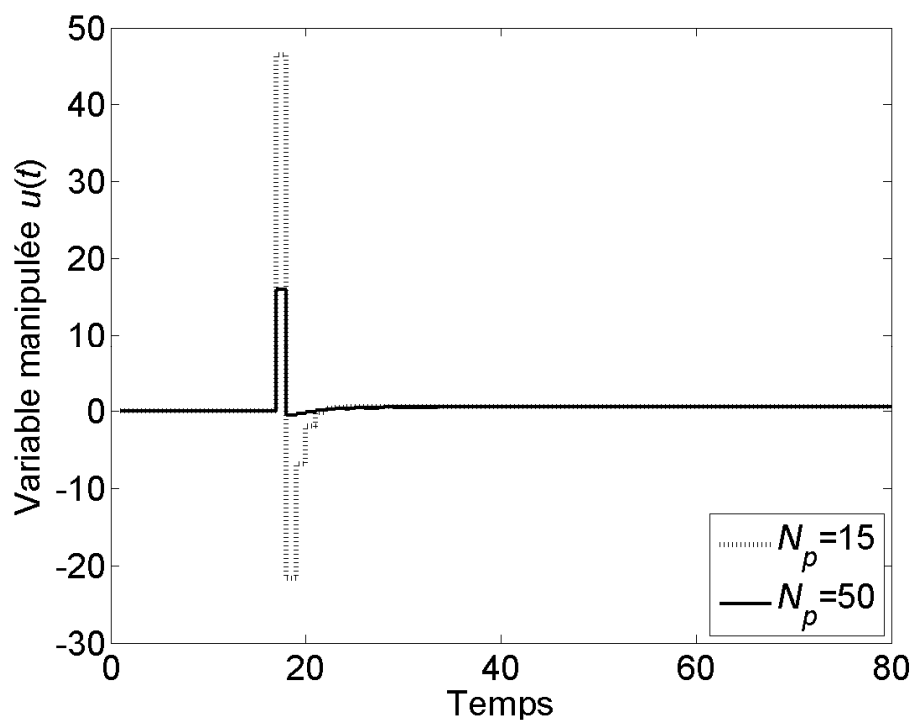


b) Evolution de la commande.

Figure 1.6 Influence de l'horizon de commande.



a) Evolution de la sortie.



b) Evolution de la commande.

Figure 1.7 Influence de l'horizon de prédiction.

1.5 Stabilité de la commande prédictive

La commande prédictive est basée sur le modèle pour calculer des prédictions de la sortie qui seront utilisées par la suite dans l'algorithme d'optimisation, souvent itératif, pour déterminer la séquence des commandes optimale à appliquer. Ainsi, à chaque instant d'échantillonnage, un problème d'optimisation est résolu en temps réel. Par conséquent, la stabilité de la commande prédictive est liée directement, à la stabilité du modèle utilisé (Lee et Park, 1991) pour l'étape de prédiction donc du système à commander, et à la convergence de l'algorithme d'optimisation. Pour assurer la stabilité en boucle fermée, le système en boucle ouverte doit être stable pour que les prédictions le seront aussi, et l'algorithme d'optimisation doit garantir une convergence même locale. Pour la commande des systèmes instables ou intégrateurs, un correcteur de premier niveau doit être synthétiser au premier lieu pour stabiliser le système (Lee et Park, 1991 ; Mayne et al., 2000 ; Richalet et al., 2005), puis on utilise la commande prédictive pour assurer certaines performances, spécifiées par un critère à optimiser, et respecter bien sûr des contraintes de fonctionnement du système, et de sécurité.

1.6 Etat de l'art sur la commande prédictive

La commande prédictive est largement utilisée dans des domaines très variés de l'industrie tels que la pétrochimie, en particulier le génie des procédés et l'automobile (Kouvaritakis et Cannon, 2001 ; Findeisen et al., 2007 ; Huang et Kadali, 2008 ; Magni et al., 2009) où elle a démontré ses preuves. Elle fait partie des stratégies de commande qui utilise explicitement le modèle du système à commander en s'acquittant de deux tâches essentielles : la prédiction explicite du comportement dynamique du système et le calcul de la commande appropriée pour assurer la poursuite de la trajectoire de consigne en minimisant un certain critère de performance (Flaus, 1994 ; Huang et Kadali, 2008).

Le principe de la commande prédictive présenté reste le même, les différentes méthodes proposées dans la littérature diffèrent surtout par la structure choisie pour représenter le modèle de système, c'est-à-dire pour la prédiction. Pour le critère, en général, c'est le critère quadratique qui est retenu. Dans le cas de la commande prédictive non linéaire, en plus de la structure de modèle considérée, la différence entre les techniques de commande réside aussi dans la technique d'optimisation utilisée pour la détermination de la séquence des actions de commande.

Les premiers travaux sur la commande prédictive remontent à la fin des années 70 et ont été initiés par Richalet (1976) en proposant le *Model Algorithmic Control* (MAC). Puis De Keyser et Van Cauwenberghe (1979) introduisent le *Extended Prediction Self Adaptive*

Control (EPSAC) qui est une commande auto-adaptative prédictive étendue développée . L'idée consiste à utiliser un signal de commande constant pour tout l'horizon de prédiction, et qui est appliqué dès le début du calcul de la commande qui optimise le critère. Cutler et Ramaker (1979) ont développé le célèbre algorithme *Dynamic Matrix Control* (DMC), basé sur le modèle de convolution (réponse indicielle et impulsionnelle), présenté dans ce chapitre. Cet algorithme est le plus indiqué pour illustrer le principe d'une commande prédictive. Le DMC est l'algorithme qui connaît une large utilisation, dans le domaine industriel, surtout pour les systèmes linéaires et pour les systèmes non linéaires lorsqu'une linéarisation autour d'un point de fonctionnement est possible. Ydstie (1984) propose l'*Extended Horizon Adaptive Control* (EHAC) qui est une commande adaptative à horizon étendu. L'idée fondamentale consiste à calculer à chaque instant, la séquence des signaux de commande pour essayer de maintenir la sortie future la plus proche possible de la consigne pour un horizon de temps plus grand que le retard présent sur le système. Clarke (1987) développe le *Generalized Predictive Control* (GPC) qui détrône le DMC et qui devient la méthode la plus populaire. Comparativement au DMC, cet algorithme utilise une adaptation en ligne du modèle utilisé (modèle) pour réaliser les prédictions. Le GPC est très proche de l'ESPAC et de l'EHAC. On peut trouver une synthèse sur ces méthodes et de leurs caractéristiques les plus importantes dans De Keyser et al. (1988) et dans Clarke et Mohtadi (1989).

Le succès de ces différents algorithmes, en particulier le DMC et le GPC, a suscité graduellement un grand intérêt pour la commande prédictive, pas seulement au milieu académique mais aussi au milieu industriel, depuis les années 80. Cet engouement a débouché sur d'autres méthodologies, partageant le même principe, qui sont apparues dans la littérature spécialisée de la commande dont certaines sont actuellement commercialisées par certaines firmes (voir le Tableau 1.1).

Algorithme	Firme
DMC (Dynamic Matrix Control)	ASPEN Tech
Aspen Target	
ADMC (Adaptive DMC)	CTC
IDCOM (Identification and Command)	Adersa
HIECON (Hierarchical Constraint Control)	
PFC (Predictive Functional Control)	
RMPCT (Robust Multivariable Predictive Control)	Honeywell
SMCA (Setpoint Multivariable Control Architecture)	Stepoint Inc.
IDCOM-M (Identification and Command Multivariable)	
APCS (Adaptive Predictive Control System)	SCAP Europa
SMOC (Shell Multivariable Optimising Controller)	Shell
Connoisseur (Control and Identification package)	Invensys
MVC (Multivariate Control)	Continental Controls Inc.
NOVA-NLC (NOVA Nonlinear Controller)	DOT Products
Process Perfecter	Pavilion Technologies

Tableau 1.1 Version commercialisés des correcteurs prédictifs.

Les années 90, ont marqué une vraie explosion dans le nombre des applications de la commande prédictive principalement en Etats-Unis, en Japon et en Europe. Par exemple, pour l'industrie des processus chimiques et le domaine de la robotique, plusieurs algorithmes ont été appliqués avec succès (De Keyser, 1988). Ces applications ont été accompagnés d'une forte activité de recherche (Camacho et Bordons, 1998).

Comme la plupart des algorithmes utilisent des modèles entrée-sortie, la commande prédictive a été formulée dans le contexte de la représentation en variables d'état (Morari, 1994). L'objectif visé est de faire usage des théorèmes et résultats existant dans la théorie d'espace d'état, mais aussi facilite l'extension de la théorie MPC à des cas plus complexes.

Étant donné que la détermination de la séquence de commande demande plus d'efforts de calcul et exige des algorithmes de programmation non linéaire (algorithme d'optimisation) rapides et convergent, alors les travaux de recherche se sont focalisés, dans les années 2000, sur le développement et l'étude des algorithmes permettant d'obtenir une solution, le plus souvent locale, pour le problème d'optimisation à résoudre à chaque période d'échantillonnage (Ramirez et Camacho, 2001 ; Bemporad et al. 2002 ; Cannon, 2004, Magni et al., 2009). Ainsi, présenter un état de l'art concernant cette période, revient à esquisser les

différents algorithmes et stratégies d'optimisation développées pour accélérer la convergence des algorithmes d'optimisation et l'obtention de la solution globale, en particulier pour le cas de la commande prédictive non linéaire. Le travail présenté dans ce mémoire s'inscrit dans cette optique dont l'objectif consiste à appliquer certains algorithmes de l'optimisation globale réputés robustes dans une stratégie de commande prédictive non linéaire. Ainsi, une synthèse sur les différents algorithmes d'optimisation et approches utilisées dans une stratégie de commande prédictive fera l'objet du chapitre 4 dédié à la commande prédictive non linéaire.

1.7 Conclusion

Ce chapitre a été consacré à des généralités sur la commande prédictive et l'état de l'art concernant cette approche de commande. Ainsi, après avoir présenté le principe de la commande prédictive, nous avons détaillé les différents éléments d'une commande prédictive, et les paramètres de réglage d'une commande prédictive, en l'occurrence l'instant de début de prédiction, l'horizon de commande, l'horizon de prédiction, et l'horizon du système. Pour illustrer le principe de la commande prédictive, l'algorithme *Dynamic Matrix Control* (DMC) est présenté d'une manière détaillée, et appliquée à un système de second ordre avec retard. Des résultats de simulation ont été présentés pour démontrer les performances du DMC et pour illustrer l'influence de deux paramètres de réglage importants, en l'occurrence les horizons de commande et de prédiction. Les conclusions tirées coïncident avec celles rapportées dans la littérature.

La fin du chapitre constitue une synthèse sur les différents travaux réalisés dans le cadre de la commande prédictive des systèmes. Cette synthèse retrace les contributions importantes qui ont marqué le développement de la commande prédictive.

Dans une commande prédictive, l'obtention de la solution optimale passe par la résolution d'un problème d'optimisation souvent non linéaire, par conséquent cette étape d'optimisation joue un rôle du premier plan dans une stratégie de commande prédictive. En effet, le succès d'une stratégie de commande prédictive est lié directement à l'algorithme d'optimisation utilisé. Ainsi, l'examen de la littérature dédiée à la commande prédictive, en particulier la commande prédictive non linéaire montre que les efforts de recherche sont focalisés, ces dernières années, au développement d'algorithmes d'optimisation rapides qui convergent vers l'optimum globale en un temps inférieur à une période d'échantillonnage, qui constituent actuellement un vrai challenge pour les mathématiciens appliqués et en particulier aux automaticiens.

Dans ce travail, des algorithmes d'optimisation globale seront adoptés dans la stratégie de commande prédictive non linéaire. Ainsi, le chapitre suivant est consacré à l'optimisation globale différentiable et non différentiable des fonctions mathématiques où les différentes notions utilisées le long du travail seront exposées.

Chapitre 2

Optimisation Globale dans \mathcal{R}

2.1 Introduction

Dans une stratégie de commande prédictive, un problème d'optimisation doit être résolu à chaque instant d'échantillonnage pour déterminer la commande à appliquer au système. L'étape de résolution du problème d'optimisation est primordiale dans une commande prédictive, car les performances attendues dépendent de la solution du problème d'optimisation, elles sont meilleures si l'optimum global du critère est atteint. A cette contrainte s'ajoute celle relative au temps de résolution qui doit être inférieur à une période d'échantillonnage. La nature du problème d'optimisation à résoudre dépend de la nature du critère, du système, et des contraintes. Hormis le cas où le système et les contraintes sont linéaires et le critère est quadratique pour lequel la solution analytique est disponible (cas du *Dynamic Matrix Control* présenté au chapitre 1), les autres cas sont souvent des problèmes d'optimisation non linéaires pour lesquels la localisation de l'optimum global est très difficile.

Résoudre un problème d'optimisation revient à localiser l'optimum global du critère en présence ou en absence de contraintes. Le domaine de l'optimisation constitue un domaine de

recherche immense. Ces dernières années, avec l'évolution de l'informatique, une attention particulière a été accordée à l'optimisation globale dont l'objectif est de développer des algorithmes sophistiqués et rapides permettant de localiser l'optimum global d'une fonction mathématique et d'échapper ainsi au problème rencontré avec les méthodes de l'optimisation locale qui sont souvent piégées dans un minimum local.

Ce chapitre est consacré à des généralités sur l'optimisation globale dans l'ensemble des nombres réels. L'objectif consiste à présenter les différentes notions utilisées, le long de ce mémoire, relatives aux mathématiques répertoriées sous le vocable optimisation globale. Ainsi, après la formulation mathématique du problème d'optimisation, on donne une classification des problèmes d'optimisation. Puis à l'aide d'un exemple, on illustre la nécessité et le besoin de développer des méthodes d'optimisation globale, suivi de la classification des méthodes d'optimisation globale développées dans la littérature. La fin du chapitre est réservée à la relation étroite entre l'optimisation globale et la commande prédictive non linéaire.

2.2 Formulation mathématique d'un problème d'optimisation

Considérons une fonction scalaire de plusieurs variables x_1, x_2, \dots, x_n , appelées variables de décision, notée $f(x)$, et appelée fonction objectif ou critère. Le vecteur de variables de décision, noté $x = (x_1, x_2, \dots, x_n)^T$, doit appartenir à un domaine donné $D \in \mathfrak{R}^n$. Ce dernier est défini par des relations de contraintes du type égalité

$$g_i(x) = 0, i = 1, \dots, p \quad (2.1)$$

et/ou inégalité

$$h_j(x) \leq 0, j = 1, \dots, q \quad (2.2)$$

L'objectif de l'optimisation globale est de rechercher parmi les $x \in D$, un x particulier qui est appelé minimum absolu ou global (n'est pas obligatoirement unique), noté x^* , tel que :

$$f(x^*) \leq f(x), \forall x \in D \quad (2.3)$$

Nous présentons un problème d'optimisation mathématiquement sous la forme suivante :

$$\begin{aligned} & \min_x f(x) \\ \text{Sujet à :} & \\ & g_i(x) = 0, i = 1, \dots, p \\ & h_j(x) \leq 0, j = 1, \dots, q \end{aligned} \quad (2.4)$$

ou encore

$$\begin{aligned} & \arg \min_x f(x) \\ \text{Sujet à :} & \\ & g_i(x) = 0, i = 1, \dots, p \\ & h_j(x) \leq 0, j = 1, \dots, q \end{aligned} \tag{2.5}$$

L'argument *argmin* identifie la valeur des variables de décision qui atteignent le minimum, c'est-à-dire

$$x^* = \arg \min_x f(x) \tag{2.6}$$

Alors que l'opérateur *min* identifie la valeur correspondante de la fonction objectif ou critère, c'est-à-dire

$$f(x^*) = \min_x f(x) \tag{2.7}$$

Remarque 2.1

Notons que, moyennant le changement de fonctions $f(x)$ en $-f(x)$, on peut passer d'une formulation minimiser à une formulation maximiser et vice versa : minimiser $f(x)$ revient à maximiser $-f(x)$, c'est-à-dire :

$$\min_x f(x) = -\max_x [-f(x)] \tag{2.8}$$

Dans l'étude présentée dans ce mémoire, on considère les différents résultats et théories relatifs à la formulation « minimiser ».

2.3 Classification des problèmes d'optimisation dans \mathfrak{R}

Suivant la nature de la fonction objectif ou du critère et des contraintes qui définissent le domaine des solutions admissibles D , le problème d'optimisation correspondant porte des noms divers. On distingue plus particulièrement les cas rassemblés dans le Tableau 2.1.

		Fonction objectif		
		Linéaire	Quadratique (convexe)	Non linéaire
Contraintes	Linéaire	Programmation linéaire	Programmation Quadratique	Programmation Non linéaire
	Quadratique (convexe)	Programmation Non linéaire	Programmation Convexe	Programmation Non linéaire
	Non linéaire	Programmation Non linéaire	Programmation Non linéaire	Programmation Non linéaire

Tableau 2.1 Classification des problèmes d'optimisation

- Programmation linéaire

Il s'agit d'une classe de problèmes d'optimisation où la fonction objectif ou le critère est linéaire

$$f(x) = C^T x \quad (2.9)$$

(C est un vecteur des coefficients) et où l'ensemble des contraintes est un polyèdre convexe fermé (représenté sous la forme $Ax \leq b$ par exemple, avec A et b sont respectivement une matrice et un vecteur constants).

- Programmation quadratique

Il s'agit d'un problème d'optimisation où la fonction objectif est quadratique convexe, c'est-à-dire

$$f(x) = \frac{1}{2} x^T A x + C^T x \quad (2.10)$$

avec A est une matrice symétrique semi-définie positive et l'ensemble de contrainte est toujours un polyèdre convexe fermé comme dans le cas de la programmation linéaire.

- Programmation convexe

Ici la fonction objectif est convexe, et l'ensemble admissible D est convexe. Le cas de la programmation quadratique est un cas particulier de la programmation convexe.

Un domaine D est convexe si :

$$\forall (x_1, x_2) \in D \text{ et } 0 \leq \alpha \leq 1 \Rightarrow x = \alpha x_1 + (1 - \alpha) x_2 \in D \quad (2.11)$$

Une fonction $f(x)$ est convexe dans un domaine convexe D si :

$$\forall (x_1, x_2) \in D \text{ et } 0 \leq \alpha \leq 1 \Rightarrow f(\alpha x_1 + (1 - \alpha) x_2) \leq \alpha f(x_1) + (1 - \alpha) f(x_2) \quad (2.12)$$

- Programmation non linéaire

Toutes les données du problème d'optimisation sont des fonctions différentiables (continûment différentiables), c'est-à-dire la fonction objectif est différentiable et les contraintes sont supposées différentiables.

2.4 Conditions pour un minimum

Dans la suite du travail présenté dans ce mémoire, on considère seulement un cas particulier de contraintes du type inégalités, il s'agit des contraintes « de boîtes » données comme suit :

$$x_i^{\min} \leq x_i \leq x_i^{\max} \quad (2.13)$$

où x_i^{\min} et x_i^{\max} sont des composantes de deux vecteurs x^{\min} et x^{\max} , de dimension n , données. Ces deux vecteurs définissent un domaine admissible D hyperrectangulaire.

Deux hypothèses classiquement rencontrées pour le minimum. La première hypothèse concerne l'hyperrectangulaire $D \in \mathfrak{R}^2$ supposé convexe et compact. La deuxième est relative à la fonction objectif ou critère, à minimiser, supposée continue et possède des dérivées partielles premières et secondes, c'est-à-dire

$$\frac{\partial f(x)}{\partial x_i} \text{ et } \frac{\partial^2 f(x)}{\partial x_i^2}, i = 1, \dots, n \quad (2.14)$$

qui sont continues pour tout x .

Sous ces deux hypothèses, les conditions nécessaires et suffisantes pour que x^* soit un minimum local ou global de $f(x)$ sont :

- *Condition de stationnarité* (relative au gradient de la fonction objectif)

$$\nabla_x f(x^*) = 0 \quad (2.15)$$

- *Condition pour un minimum* (relative au Hessian de la fonction objectif)

$$\nabla_x^2 f(x^*) = \left[\frac{\partial^2 f(x)}{\partial x_i \partial x_j} \Big|_{x=x^*} ; i, j = 1, \dots, n \right] > 0 \quad (2.16)$$

La condition (2.16) signifie que le Hessian est une matrice définie positive. Les deux conditions (2.15) et (2.16) reviennent à supposer que la fonction objectif $f(x)$ est strictement convexe dans un voisinage de x^* . Notons que dans le cas d'une fonction convexe, vérifiant la condition (2.16), la stationnarité à elle seule constitue une condition nécessaire et suffisante d'optimalité globale.

Horst et Pardalos (1995, chapitre 1) rassemble des résultats théoriques récents relatifs à des problèmes d'optimisation globale dotés d'une structure particulière.

2.5 Méthodes de résolution des problèmes d'optimisation

L'obtention de la solution du problème d'optimisation, revient à résoudre le système d'équations algébriques (2.15), par conséquent la difficulté est liée directement à la nature de la fonction objectif $f(x)$. Ainsi, la solution peut être obtenue analytiquement dans le cas où la résolution du système d'équations (2.15) est facile, dans le cas contraire, on procède par des méthodes numériques. Pratiquement, les fonctions objectifs à minimiser sont fortement non linéaire, par conséquent résoudre analytiquement le système algébrique (2.15) est quasiment

impossible d'où la nécessité d'utiliser les méthodes numériques. Ce besoin a donné naissance à un domaine de recherche, faisant partie des mathématiques appliquées, appelé *optimisation numérique* dont les efforts sont centralisés sur le développement des algorithmes numériques qui permettent de localiser l'optimum d'une fonction en un nombre d'itérations faible.

La programmation linéaire (voir la section 2.3) constitue un des rares cas particuliers pour lequel un algorithme itératif, appelé algorithme de simplexe, a été développé. Cet algorithme converge en un nombre d'itérations fini, et n'impliquent que des calculs assez élémentaires. Dans le cas d'une programmation quadratique convexe (voir la section 2.3), les algorithmes développés sont efficaces et convergent, en général, en une seule itération (par exemple la méthode de Newton). Dans le cas de la programmation quadratique (voir la section 2.3), la solution analytique est toujours disponible et facile à déterminer (cet avantage a été exploité dans le cas de la commande prédictive pour développer l'algorithme *Dynamic Matrix Control* présentée au chapitre 1, section 1.4).

Comme nous l'avons déjà signalé, une solution explicite ne peut pas être en général obtenue à partir des conditions théoriques (2.15). On va donc utiliser des méthodes itératives, dont le principe général consiste à calculer à partir d'une valeur initiale $x^{(0)}$, la suite des valeurs

$$x^{(1)}, x^{(2)}, \dots, x^{(k)}, \dots \quad (2.17)$$

Si l'on choisit les points successifs de façon que :

$$f(x^{(0)}) > f(x^{(1)}) > \dots > f(x^{(k)}) \quad (2.18)$$

comme cette suite numérique est bornée inférieurement par $f(x^*)$ elle convergera. Si le minimum global x^* est unique et s'il n'y a pas d'autre minimum local, la méthode converge alors vers le point x^* cherché. Dans le cas contraire seul l'un des minimums locaux est atteint. Ce dernier cas justifie aussi, le besoin à développer des algorithmes d'optimisation globale.

Puisqu'un algorithme numérique ne peut pas fournir mieux qu'une réponse approchée, dans le cas de contraintes boîtes, un élément x d'un des ensembles suivants sera considéré comme solution du problème (Berthiau et Siarry, 2001), x^* est un minimum global et ε est un nombre positif quelconque :

$$A_x(\varepsilon) = \{x \in D; \|x - x^*\| \leq \varepsilon\} \quad (2.19)$$

$$A_f(\varepsilon) = \{x \in D; \|f(x) - f(x^*)\| \leq \varepsilon\} \quad (2.20)$$

2.6 Classification des méthodes d'optimisation

Pour l'optimisation continue, on sépare sommairement le cas linéaire (qui relève notamment de la programmation linéaire pour lequel la solution est facile à obtenir) du cas non linéaire. Pour certains problèmes qui vérifient la propriété de convexité, la solution peut être obtenue en utilisant une méthode locale qui exploite, ou non, les gradients de la fonction objectif. Néanmoins, la plupart des problèmes d'optimisation sont classés difficiles, car le nombre de minima locaux est très élevé, alors le recours à une méthode globale s'impose.

Pour le traitement des contraintes, on a recours à la méthode des multiplicateurs de Lagrange dans le cas des contraintes du type égalités, et la méthode de Kuhn-Tucker ou la méthode de fonctions d'écarts dans le cas de contraintes inégalités. Notons que la méthode de fonctions de pénalisation constitue une approche élégante pour traiter les deux cas. Cette méthode permet de ramener un problème d'optimisation posé avec contraintes à un problème sans contraintes, puis d'appliquer les méthodes d'optimisation numériques.

En utilisant la méthode de fonction de pénalisation, pour le problème d'optimisation (2.4), posé avec contraintes égalités et inégalités, on obtient :

$$\min_x L(x) = f(x) + \sum_{i=1}^p \alpha_i [g_i(x)]^2 + \sum_{j=1}^q \beta_j [h_j(x)]^2 \quad (2.21)$$

où les α_i et β_j représentent les paramètres de pénalisation, positifs, associés respectivement aux contraintes $g_i(x)$ et $h_j(x)$. Les valeurs de ces paramètres sont en général considérés constantes durant la résolution du problème d'optimisation, c'est-à-dire :

$$\alpha_i = \alpha, i = 1, \dots, p \text{ et } \beta_j = \beta, j = 1, \dots, q \quad (2.22)$$

Dans la relation (2.21), la fonction $h_j(x)$ est définie comme suit :

$$h_j(x) = \begin{cases} h_j(x) & \text{si } h_j(x) > 0 \\ 0 & \text{si } h_j(x) \leq 0 \end{cases} \quad (2.23)$$

Dans la littérature, on distingue trois classes pour les méthodes d'optimisation numériques : les méthodes exactes (appelées encore classiques ou déterministes), les méthodes heuristiques (appelées encore méthodes stochastiques), et les méthodes hybrides (coopération entre une méthode exacte et une méthode heuristique).

- Méthodes exactes

Ces méthodes requièrent des propriétés mathématiques restrictives de la fonction objectif à optimiser telle que la continuité, la différentiabilité, et la convexité. Ces méthodes permettent

de localiser la solution avec une très grande précision au prix, généralement, d'un nombre d'itérations important, c'est-à-dire nécessite un temps de calcul énorme parfois. Comme exemples de méthodes, on peut citer les méthodes du gradient et de Newton, et leurs variantes. Néanmoins, en pratique, il est très difficile de savoir si la fonction objectif satisfait ou non de telles propriétés mathématiques. De plus, la plupart des fonctions sont multimodales (plusieurs maximums et minimums), discontinues et non dérivables.

Le principe de ces méthodes peut être explicité par l'algorithme suivant :

Etape 1. Initialisation $k = 0$, $x^{(0)}$ et calcul de $f(x^{(0)})$.

Etape 2. Pour $k = k + 1$, on calcule la variation de $x^{(k)}$ donnée par le vecteur suivant :

$$\Delta x^{(k)} = \begin{bmatrix} \Delta x_1^{(k)} \\ \Delta x_2^{(k)} \\ \vdots \\ \Delta x_n^{(k)} \end{bmatrix}$$

en utilisant une procédure appropriée selon la méthode utilisée (par exemple la méthode du gradient, méthode du gradient conjugué ou la méthode de Newton).

Etape 3. Calcul $x^{(k)} = x^{(k-1)} + \Delta x^{(k)}$.

Etape 4. Calcul de $f(x^{(k)})$ et son gradient noté $\nabla_x f(x^{(k)})$, et vérification de la convergence en utilisant un critère approprié (par exemple la norme du gradient de la fonction à optimiser doit être inférieur à une valeur ε , c'est-à-dire ($\|\nabla_x f(x^{(k)})\| < \varepsilon$). Si la convergence est assurée on passe à l'étape 5, sinon on revient à l'étape 2.

Etape 5. La solution est $x^* = x^{(k)}$.

Par exemple dans le cas de la méthode de Newton, la variation de $x^{(k)}$, notée $\Delta x^{(k)}$, est donnée comme suit :

$$\Delta x^{(k)} = -[\nabla_x^2 (f(x^{(k)}))]^{-1} \nabla_x (f(x^{(k)})).$$

- Méthodes heuristiques

Ces dernières années, on marque l'arrivée d'une nouvelle classe de méthodes, nommées méthodes heuristiques ou stochastiques qui constituent une alternative très intéressante aux méthodes exactes. Les méthodes heuristiques sont un ensemble d'algorithmes d'optimisation

visant à résoudre des problèmes d'optimisation difficile. Ces méthodes vont chercher un optimum de façon aléatoire (au hasard). En général, elles procèdent par voisinages successifs. De façon plus précise : une solution étant obtenue à l'itération k , à l'itération $k + 1$ on cherche au hasard une solution meilleure dans un voisinage précédent. Les itérations successives doivent permettre de passer d'une solution de mauvaise qualité à la solution optimale. L'algorithme s'arrête après avoir atteint un critère d'arrêt, consistant généralement en l'atteinte du temps d'exécution imparti ou en une précision demandée.

Parmi les heuristiques, on retrouve les heuristiques de *voisinage*, qui font progresser une seule solution à la fois (par exemple la méthode de recuit simulé, la recherche tabou), et les heuristiques *distribuées*, qui manipulent en parallèle toute une population de solutions (par exemples les algorithmes génétiques, les essaims de particules). L'inconvénient majeur de ces méthodes est qu'on ne peut garantir leur convergence vers la solution globale que d'une manière asymptotique, c'est-à-dire elles permettant de localiser un $\tilde{x} \in A_\epsilon(x)$ (voisinage de l'optimum global). Bien que ces méthodes puissent être adaptées à tout type de problème d'optimisation, mais elles sont souvent moins puissantes que les méthodes exactes sur certains types de problèmes. Elles ne garantissent pas non plus la découverte de l'optimum global en un temps fini. Cependant, un grand nombre de problèmes réels n'est pas optimisable efficacement par des approches purement mathématiques, les heuristiques peuvent alors être utilisées avec profit par exemple dans le cas où l'expression analytique de la fonction objectif à optimiser est absente ou cette dernière est non différentiable.

- Méthodes hybrides

Les méthodes hybrides profitent des avantages des méthodes exactes et heuristiques. Le principe de ces méthodes consiste à coopérer une méthode heuristique et une méthode exacte pour localiser l'optimum global recherché avec une précision déterminée. En général, la recherche de l'optimum global commence par l'utilisation d'une heuristique qui localise un \tilde{x} appartenant au voisinage de l'optimum globale, c'est-à-dire $A_\epsilon(x)$, puis on passe le relais à une méthode d'optimisation exacte qui permet de localiser l'optimum globale \hat{x} avec une bonne précision.

2.7 Intérêt des méthodes globales

Pour représenter avec précision les phénomènes et les grandeurs physiques, on utilise souvent des modèles ou des fonctions de nature non linéaire. Ces modèles non linéaires

constituent des clés pertinentes pour réussir toute étude scientifique quantitative, en particulier les problèmes pratiques formulés sous forme de problèmes d'optimisation. Néanmoins, la difficulté vient de la manipulation des non linéarités.

En optimisation, la présence des non linéarités demande plus d'efforts de calcul, et une difficulté pour localiser l'optimum global parmi les extremums possibles pour la fonction objectif. Le problème d'optimisation devient plus complexe avec la présence de contraintes et selon le degré de leurs non linéarités. Aussi, le nombre d'extremums devient important avec l'augmentation des variables de décision. Pour illustrer cette difficulté, considérons le problème d'optimisation, à une seule variable de décision, suivant :

$$\begin{aligned} \min_x \cos(x) \sin(x^2 - x) \\ \text{Sujet à :} \quad 0 \leq x \leq 10 \end{aligned} \tag{2.24}$$

L'évolution de la fonction objectif pour ce problème est donnée par la Figure 2.1. On constate que la fonction admet plusieurs minimums. L'objectif de l'optimisation est de localiser le minimum global en un nombre fini d'itérations et avec une très grande précision. La complexité du problème d'optimisation augmente exponentiellement avec l'augmentation du nombre de variables de décision ou des contraintes. Pour illustrer ce point, considérons le problème d'optimisation obtenu simplement en considérant une fonction objectif définie comme la somme de deux fonctions mathématiques de même forme que celle du problème d'optimisation (2.24), c'est-à-dire :

$$\begin{aligned} \min_x \cos(x_1) \sin(x_1^2 - x_1) + \cos(x_2) \sin(x_2^2 - x_2) \\ \text{Sujet à :} \quad 0 \leq x_1 \leq 10 \\ \quad \quad \quad 0 \leq x_2 \leq 10 \end{aligned} \tag{2.25}$$

La Figure 2.2, montre que le nombre des minimums a augmenté avec l'augmentation de variables de décision.

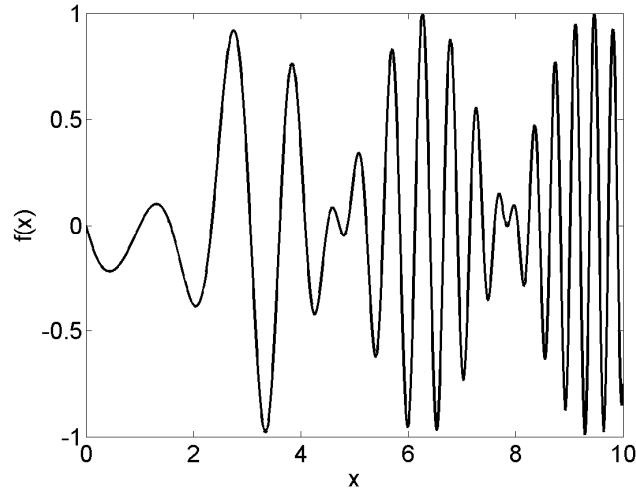


Figure 2.1 Evolution de la fonction objectif du problème d'optimisation (2.24).

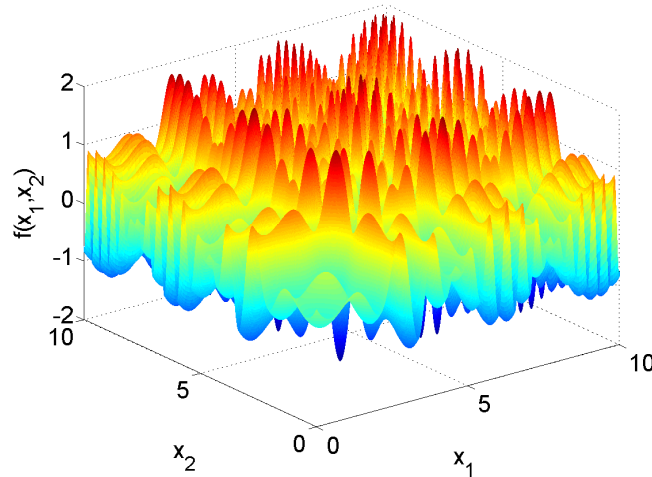


Figure 2.2 Evolution de la fonction objectif du problème d'optimisation (2.25).

La question qui se pose y a-t-il des tests permettant de filtrer des minimums globaux parmi les minima locaux ? Mis à part le cas où la fonction objectif est convexe dans lequel il n'y a pas lieu de faire de distinguo, les conditions nécessaires et/ou suffisantes de minimisation globale ne sont accessibles que pour certaines classes de problèmes d'optimisation spécialement structurés. Néanmoins, reconnaître un minimum global de la fonction objectif parmi les points critiques (les points qui vérifient le système d'équations algébriques (2.5)) de la fonction objectif est possible grâce à la notion de l'enveloppe convexe de la fonction objectif. L'enveloppe convexe notée, $\text{conv } f(x)$ est définie comme la plus grande fonction convexe minorant $f(x)$ sur $D \in \mathfrak{R}^n$. Ainsi, ce qui manque pour un point critique \tilde{x} pour être un minimum global de $f(x)$ est exactement la propriété suivante (Hiriart-Urruty, 1996) :

$$\text{conv } f(\tilde{x}) = f(\tilde{x}) \quad (2.26)$$

Evidemment, cette propriété n'est pas facile à tester puisqu'il est difficile habituellement de déterminer $\text{conv } f(x)$ exactement surtout pour une fonction à plusieurs variables ou non différentiables. De plus, l'obtention de cette enveloppe demande plus d'efforts. La Figure 2.3 illustre la notion de l'enveloppe convexe.

Ces difficultés rencontrées pour localiser le minimum global a incité les mathématiciens, en particulier les mathématiciens appliqués, à développer des méthodes numériques permettant de localiser directement le minimum globale d'une fonction non linéaire à plusieurs variables. Ce qui a donné naissance à un domaine de recherche très actif appelé « optimisation globale ».

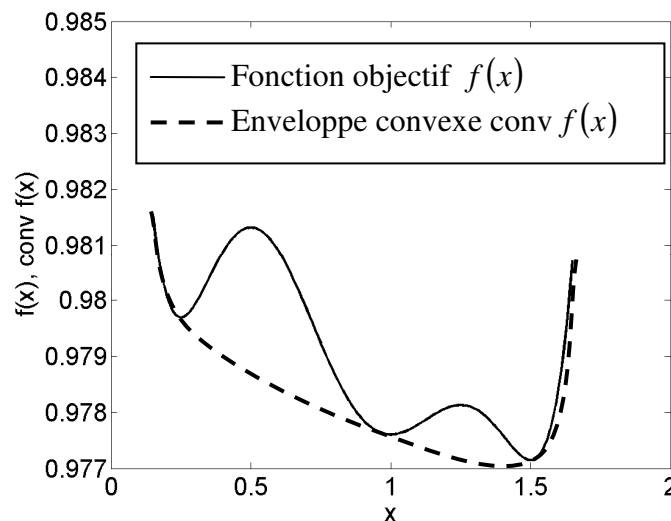


Figure 2.3 Fonction objectif et son enveloppe convexe.

2.8 Méthodes d'optimisation globales et commande prédictive

Dans un problème de commande prédictive un problème d'optimisation doit être résolu à chaque instant d'échantillonnage. La nature du problème d'optimisation dépend de la nature du système, du critère (fonction objectif) et les contraintes considérées. Généralement, le critère considéré prend une forme quadratique incluant des objectifs de poursuite et d'énergie minimale. Bien que le critère de performances est quadratique, ce qui représente une forme de fonction objectif intéressante pour localiser la solution globale, néanmoins la non linéarité du problème d'optimisation vient de la nature du modèle utilisée pour la prédiction. Dans le cas d'un modèle linéaire, une solution analytique du problème d'optimisation peut être obtenu, c'est le cas de l'algorithme *Dynamic Matrix Control*, présenté dans la section 1.4 du chapitre

1, par conséquent vu la capacité des calculateurs numériques disponibles, l'implémentation ne pose pas de problème puisque la solution peut être calculée rapidement. Par contre, si le modèle est non linéaire, les prédictions le seront aussi, par conséquent le critère prend une forme non linéaire dont la complexité dépend directement des non linéarités du système. Ainsi, le problème d'optimisation est non linéaire, donc l'obtention de la solution à chaque instant d'échantillonnage est une tâche très délicate, et nécessite des méthodes numériques qui convergent en un nombre fini d'itérations et rapidement vers la solution optimale.

Comme la quasi-totalité des systèmes sont de nature non linéaire, pour lesquels l'hypothèse de linéarité ne présente aucun intérêt, alors l'utilisation d'un modèle non linéaire pour prédire les sorties du système à commander est inévitable. De plus, utiliser un modèle non linéaire va dans le sens d'accroître la précision de la prédiction ce qui permet d'améliorer davantage les performances du système en boucle fermée. Néanmoins, la prédiction non linéaire complexifie l'application de la commande prédictive à des systèmes non linéaires puisque le problème d'optimisation à résoudre est de nature fortement non linéaire dont la difficulté de localiser l'optimum globale, en un nombre fini d'itérations ou en un temps inférieur à la période d'échantillonnage, se présente avec acuité. L'application d'une solution locale dans une stratégie de commande prédictive n'est pas dramatique, mais en appliquant une solution globale les performances seront meilleures, puisque minimiser mieux le critère implique une bonne poursuite de consigne, un bon rejet de perturbation, et un minimum d'énergie (objectifs souvent visés par l'utilisation d'un critère quadratique). Par conséquent, l'utilisation des méthodes d'optimisation globale dans une commande prédictive non linéaire constitue une solution intéressante qui permettra d'élargir son champ d'application.

Dans ce chapitre, nous avons précisé uniquement la liaison directe entre l'optimisation et la commande prédictive non linéaire, et l'intérêt de l'utilisation des méthodes d'optimisation globales. Dans le chapitre 4, on présentera une synthèse sur les différentes approches d'optimisation globale non linéaire proposées et utilisées dans le cadre d'une commande prédictive.

2.9 Conclusion

Ce chapitre a été consacré à des généralités sur l'optimisation de fonctions mathématiques, en particulier l'optimisation globale. L'objectif consiste à présenter les éléments essentiels utilisés le long de ce mémoire, et de montrer l'intérêt des méthodes de l'optimisation globale, en particulier dans une stratégie de commande prédictive non linéaire. Quoique plusieurs méthodes d'optimisation globale ont été développées dans la littérature, mais en pratique il est

très difficile, voire impossible, de préconiser une telle méthode ou une autre pour un problème d'optimisation donné. En outre, l'examen de la littérature, montre que la théorie n'est pas d'un grand secours, puisque les théorèmes de convergence sont souvent inexistantes, ou applicables sous des hypothèses restrictives. De plus, le réglage optimal des paramètres de la méthode préconisé par la théorie est souvent inapplicable en pratique surtout dans le cas des heuristiques. Aussi, les comparaisons entre les différentes méthodes disponibles abordées dans la littérature se limitent à des problèmes de tests idéalisés. Ainsi, le choix d'une méthode d'optimisation globale fait appel souvent à l'expérience.

En consultant la littérature dédiée à l'optimisation globale, deux méthodes d'optimisation globale très intéressantes ont attiré notre attention, elles s'agissent de la méthode d'Alienor qui fait partie des méthodes exactes, et la méthode des essaims de particules qui est une méthode heuristique. Ces méthodes sont caractérisées par des algorithmes simples à comprendre et à programmer, et nécessitent peu de paramètres de réglages dont leurs choix est souvent dictées par des théories mathématiques poussées, en particulier pour la méthode d'Alienor. Ces méthodes constituent des variantes intéressantes pour réussir une commande prédictive non linéaire. Les principes de ces deux méthodes seront exposés dans le chapitre suivant, et pour évaluer leur capacité à localiser l'optimum globale, ces méthodes seront appliquées pour l'identification des paramètres d'un système non linéaire.

Chapitre 3

Méthodes d'optimisation globale : Alienor et Essaim de Particules

3.1 Introduction

Dans le chapitre précédent, on a montré l'intérêt des méthodes d'optimisation globale, en particulier dans une stratégie de commande prédictive non linéaire. Le monde industriel affronte une compétition mondiale exacerbée : gagner par exemple ne serait-ce qu'un pour cent sur l'énergie consommée par un procédé représente un gain de compétitivité très important. Pour ce faire, les efforts se sont focalisés sur le développement de méthodes et d'algorithmes d'optimisation souples et rapides qui permettant de localiser avec succès l'optimum global tout en respectant les contraintes imposées par le problème. Naturellement, l'approche numérique a été favorisée dû à l'augmentation considérable de la puissance des

ordinateurs qui a largement contribué à répondre à l'impératif de temps de calcul, et la complexité des problèmes d'optimisation (fortement non linéaires).

La littérature sur l'optimisation décrit pour l'essentiel des méthodes locales qui permettent l'obtention de minima locaux qui peuvent être certes intéressantes dans une étude préalable du phénomène mais qu'il faudra de toute façon améliorer pour optimiser au mieux le problème formulé ou modélisé. Quant aux méthodes globales existantes elles sont la plupart du temps des améliorations des méthodes locales que l'on tente de débloquent d'un minimum local pour aller vers un meilleur minimum, espérant qu'in fine on obtiendra un minimum global.

Parmi ces méthodes, on retrouve la méthode d'Alienor développée par l'équipe de recherche d'Yves Cherruault dans les années 80, dont le principe est simple et l'efficacité n'est plus à démontrer. Ces dernières années, les méthodes heuristiques ont aussi démontré leur efficacité dans la localisation d'un optimum globale, et commence à s'imposer comme des méthodes alternatives intéressantes aux méthodes exactes qui exigent une certaine régularité des fonctions à minimiser (continuité et différentiabilité) qui ne sont pas nécessairement vérifiées. Parmi ces méthodes, on retrouve la méthode d'essaim de particules qui bénéficie d'une large utilisation pour la résolution des problèmes d'optimisation réputés difficiles. Cette méthode présente plusieurs avantages par rapport aux autres heuristiques (Hammouche et al., 2010).

Dans ce chapitre, on présentera les principes des méthodes d'optimisation globale d'Alienor et d'essaim de particules. Pour illustrer leur efficacité, les méthodes seront utilisées pour l'identification des paramètres d'un système non linéaire.

3.2 Méthode d'Alienor

La méthode Alienor, proposée par Cherruault (2005), repose sur une suite de transformations réductrices qui permet de ramener toute fonction de plusieurs variables à une fonction d'une seule variable. On peut alors utiliser, pour résoudre le problème multivariable, les méthodes puissantes habituellement mises en œuvre pour le cas unidimensionnel.

Soient deux variables réelles, x_1 , x_2 . Nous allons d'abord passer en coordonnées polaires

$$x_1 = r \cos(\alpha), x_2 = r \sin(\alpha), \alpha \geq 0 \quad (3.1)$$

Ce faisant nous obtenons deux nouvelles variables r et α que nous allons relier grâce à la spirale d'Archimède d'équation

$$r = a \alpha \quad (3.2)$$

où a est un paramètre fixé, destiné à tendre vers 0. Ainsi, les relations (3.1) deviennent

$$x_1 = a \alpha \cos(\alpha), x_2 = a \alpha \sin(\alpha) \quad (3.3)$$

Par conséquent, elles permettent d'exprimer x_1 et x_2 à l'aide d'une unique variable $\alpha \geq 0$.

La Figure 3.1 représente l'exploration globale du plan (fonction objectif à deux variables) paramétrée par le paramètre unique α .

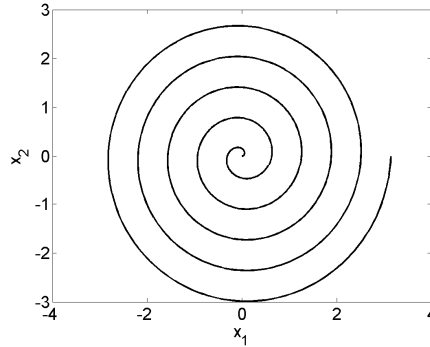


Figure 3.1 Exploration globale du plan paramétrée par α (spirale d'Archimède).

On peut généraliser cette transformation à n variables. Il suffit de relier les variables deux à deux. Considérons, par exemple, trois variables x_1, x_2, x_3 . Nous allons d'abord relier x_1 et x_2 à l'aide de α_1 , on obtient :

$$x_1 = a \alpha_1 \cos(\alpha_1), x_2 = a \alpha_1 \sin(\alpha_1) \quad (3.4)$$

Puis on relie les deux variables qui restent α_1 et x_3 à l'aide de α pour obtenir :

$$\alpha_1 = a \alpha \cos(\alpha), x_3 = a \alpha \sin(\alpha) \quad (3.5)$$

Il est alors clair que x_1, x_2, x_3 s'expriment à l'aide de α . En effet, on a :

$$\begin{aligned} x_1 &= a^2 \alpha \cos(\alpha) \cos(a \alpha \cos(\alpha)) \\ x_2 &= a^2 \alpha \cos(\alpha) \sin(a \alpha \cos(\alpha)) \\ x_3 &= a \alpha \sin(\alpha) \end{aligned} \quad (3.6)$$

On voit que, d'une façon générale, on aboutit à des relations

$$x_i = h_i(\alpha), i = 1, \dots, n; \quad \alpha \geq 0 \quad (3.7)$$

avec des les fonctions h_i faisant intervenir les fonctions trigonométriques en *sinus* et *cosinus*.

Par conséquent, ces fonctions sont infiniment différentiables.

La méthode d'Alienor peut être exploitée pour la résolution des problèmes d'optimisation de la manière suivante :

Soit à résoudre le problème d'optimisation suivant :

$$\min_{x_1, \dots, x_n} f(x_1, \dots, x_n) \quad (3.8)$$

où f est une fonction non linéaire continue.

La transformation (3.7) permet de remplacer la fonction $f(x_1, \dots, x_n)$ par la nouvelle fonction $f^*(\alpha)$ donnée comme suit :

$$f^*(\alpha) = f(h_1(\alpha), \dots, h_n(\alpha)) \quad (3.9)$$

qui est une seule fonction d'une seule variable α .

Le problème d'optimisation (3.8) est alors remplacé par le problème suivant :

$$\min_{\alpha \geq 0} f^*(\alpha) \quad (3.10)$$

qui est un problème de minimisation à une seule variable que l'on peut résoudre simplement à condition de savoir définir l'intervalle $[0, \alpha_{\max}]$ sur lequel on va rechercher le (ou les) minima globaux de $f^*(\alpha)$.

Le succès de la méthode d'Alienor de base présentée a poussé les chercheurs à développer d'autres transformations réductrices pour simplifier les calculs. Ainsi, plutôt que de réduire deux à deux les variables par la méthode d'Alienor de base, des transformations réductrices ont été proposées dont l'idée de base consiste à exprimer toutes les variables en fonction de la variable α en une seule étape. Ceci impose de choisir correctement les fonctions h_i de la transformation réductrice (3.7). Par conséquent, les efforts se sont focalisés sur ce point dont l'objectif est de développer des transformations réductrices simples qui demande moins de calculs, et surtout de bien approximer le problème d'optimisation à n variables.

Parmi les transformations réductrices proposées dans la littérature, on retrouve les transformations suivantes :

- Transformation 1 :

$$x_i = \beta \alpha \cos(\rho_i \alpha) \quad (3.11)$$

où les paramètres ρ_i forment une suite croissante et vérifient la condition suivante :

$$\frac{\rho_i}{\rho_{i+1}} < 1 \text{ et } \beta > 0 \quad (3.12)$$

- Transformation 2 :

$$x_i = \beta \cos(m^i \pi \alpha) \quad (3.13)$$

avec $m > 1$.

- Transformation 3 :

$$x_i = \beta \cos(\rho_i \alpha) \quad (3.14)$$

Les paramètres $\rho_1, \dots, \rho_{n-1}$ sont choisis proches les uns des autres, tout en constituant une suite lentement croissante, par exemple

$$\rho_{i+1} = \rho_i + \varepsilon \quad (3.15)$$

avec $\varepsilon > 0$ et choisi très petit. Le paramètre ρ_n est choisi comme suit :

$$\rho_n = \pi \sqrt{n-1} \frac{\rho_{n-1}}{\beta} \quad (3.16)$$

- Transformation 4 :

$$x_i = \beta \cos(\omega_i \alpha + \varphi_i) \quad (3.17)$$

où les $\omega_i > 0$ forment une suite croissante. φ_i est une suite lentement croissante dont les termes sont proches l'un des autres, par exemple :

$$\varphi_{i+1} = \varphi_i + \varepsilon \quad (3.18)$$

avec $\varepsilon > 0$ et choisi très petit.

Notons que la dernière transformation réductrice (3.17), permet une exploration globale du plan paramétré par α en assurant une très bonne précision par rapport aux autres.

Aussi, d'autres variantes ont été proposées pour lier r et α afin d'améliorer la précision de l'approximation. Parmi ces relations, on peut citer les suivantes (Bendiab et Cherruault, 1995) :

$$\begin{aligned} r &= e^{-\beta \alpha}, \\ r &= e^{-\beta \alpha^2 + b \alpha}, b > 0. \end{aligned} \quad (3.19)$$

Pour illustrer la détermination de la fonction à une seule variable $f^*(\alpha)$, considérons le problème d'optimisation (2.25). Pour ce faire, on propose d'utiliser la transformation 4 donnée par la relation (3.17) et de modifier les contraintes du problème d'optimisation (2.25), qui sont de type boîtes, de manière à avoir $-10 \leq x_1, x_2 \leq 10$. Ainsi, pour respecter ces contraintes, il suffit de prendre $\beta = 10$, $\omega_1 = 100$, $\omega_2 = 101$, $\varphi_1 = 1$, et $\varepsilon = 0.00001$

$$\begin{aligned} x_1 &= 10 \cos(100 \alpha + 1) \\ x_2 &= 10 \cos(101 \alpha + 1.00001) \end{aligned} \quad (3.20)$$

Ce qui donne la fonction suivante

$$\begin{aligned}
f^*(\alpha) = & \cos(10 \cos(100 \alpha + 1)) \sin\left(\left(10 \cos(100 \alpha + 1)\right)^2 - 10 \cos(100 \alpha + 1)\right) \\
& + \cos(10 \cos(101 \alpha + 1.00001)) + \sin\left(\left(10 \cos(101 \alpha + 1.00001)\right)^2 - 10 \cos(101 \alpha + 1.00001)\right)
\end{aligned} \tag{3.21}$$

En résolvant l'équation algébrique suivante :

$$\nabla_{\alpha} f^*(\alpha) = 0 \tag{3.22}$$

on détermine facilement le minimum global α^* et à partir de la transformation (3.20), on détermine la solution optimale globale comme suit :

$$\begin{aligned}
x_1^* &= 10 \cos(100 \alpha^* + 1) \\
x_2^* &= 10 \cos(101 \alpha^* + 1.00001)
\end{aligned} \tag{3.23}$$

Après avoir présenté la méthode d'Alienor qui est une méthode exacte, dans la section suivante, on présentera une méthode d'optimisation globale heuristique, en l'occurrence la méthode d'essaim de particules.

3.3 Méthode d'essaim de particules

L'optimisation par essaim de particules ou *Particle Swarm Optimisation* (PSO) en anglais fait partie des méthodes heuristiques, elle est basée sur la reproduction d'un comportement social. Elle a été développée en 1995 par Eberhart et Kennedy (1995) suite aux travaux de recherche sur la simulation de vols groupés d'oiseaux et de bancs de poissons réalisés par Reynold (1987) et Heppner et Grenander (1990). Les résultats obtenus ont démontré la capacité d'un groupe en mouvement à maintenir une distance optimale entre eux et à suivre un mouvement global par rapport aux mouvements locaux de leur voisinage. Un autre résultat important concerne l'importance du mimétisme dans la compétition qui oppose les particules à la recherche de la nourriture. Ces dernières sont dispersées aléatoirement dans un espace de recherche, et lorsqu'une particule localise une source de nourriture, les autres particules vont alors chercher à le reproduire.

Ce comportement social basé sur l'analyse de l'environnement et du voisinage représente réellement une méthode d'optimisation par observation des tendances des particules voisines. Le principe impose à chaque particule à chercher à optimiser ses chances en suivant une tendance qu'elle modère par ses propres vécus.

3.3.1 Principe de la méthode d'essaim de particules

Un essaim est disposé de façon aléatoire et homogène dans l'espace de recherche et chaque particule possède la capacité de se déplacer avec une vitesse aléatoire. Ainsi, à chaque pas de temps, chaque particule :

- évalue la qualité de sa position et garde en mémoire sa meilleure performance, c'est-à-dire la meilleure position atteinte jusqu'ici (elle peut être la position courante) et sa qualité (la valeur de la fonction à optimiser en cette position).
- interroge un certain nombre de particules pour obtenir de chacune d'entre elles sa propre meilleure performance.
- choisit la meilleure des meilleures performances dont elle a connaissance, puis adapte sa vitesse en fonction de cette information et de ses propres données et se déplace en conséquence.

Une fois la particule ayant une meilleure performance est localisée, la modification de la vitesse est une simple combinaison linéaire de trois tendances, à l'aide des coefficients de confiance :

- la tendance *aventureuse*, consistant à continuer selon la vitesse actuelle,
- la tendance *conservatrice*, ramenant plus ou moins vers la meilleure position déjà trouvée,
- la tendance *panurgienne*, orientant approximativement vers la meilleure informatrice.

L'utilisation des termes *plus ou moins* ou *approximativement* implique que le hasard joue un rôle, grâce à une modification aléatoire limitée des coefficients de confiance, ce qui favorise l'exploration de l'espace de recherche. Le principe de la méthode d'essaim de particules est résumé par la Figure 3.2. Pour réaliser son prochain mouvement, chaque particule combine trois tendances : suivre sa vitesse propre, revenir vers sa meilleure performance, aller vers la meilleure performance de ses informatrices.

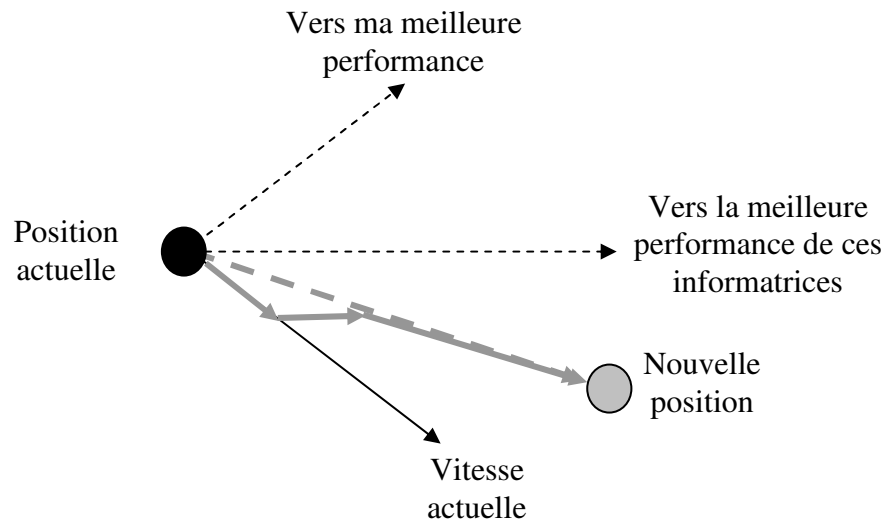


Figure 3.2 Schéma de principe du déplacement d'une particule.

3.3.2 Algorithme de la méthode d'essaim de particules

Pour illustrer la méthode d'optimisation par essaim de particules, on considère le problème de minimisation d'une fonction à plusieurs variables avec des contraintes boîtes :

$$\begin{aligned} & \min_x f(x) \\ \text{sujet à : } & x^{\min} \leq x \leq x^{\max} \end{aligned} \quad (3.24)$$

On note la taille de l'essaim (nombre de particules) par n_p et la position d'une particule j et sa vitesse, à l'itération k , respectivement par $x^j(k)$ et $v^j(k)$. Pour réduire le nombre d'évaluations de la fonction objectif, nécessaires pour déterminer la solution, on doit considérer une petite taille pour l'essaim. Généralement, une taille de 20 à 30 particules est suffisante.

Les étapes de l'algorithme de la méthode d'essaim de particules sont :

1. choix de la taille de l'essaim n_p ,
2. génération de façon aléatoire une population initiale $x^1(0), x^2(0), \dots, x^{n_p}(0)$. Les vecteurs $x^j(0) (j=1, \dots, n_p)$ correspondent aux particules. Par conséquent chaque particule est de dimension égale à n (dimension du vecteur de variables de décision).
3. évaluation de la fonction objectif pour les différentes particules de la population initiale en calculant $f(x^1(0)), f(x^2(0)), \dots, f(x^{n_p}(0))$.
4. Choix des vitesses initiales. Généralement, on considère pour la population initiale $v^j(0) = 0, j=1, \dots, n_p$ et prendre $k = 1$.

5. Détermination, pour chaque particule j , la meilleure valeur de $x^j(k)$, notée $P_{m,j}$, pour laquelle la fonction objectif est minimale pour toutes les itérations précédentes. Puis déterminer la meilleure des meilleures notée P_m comme suit :

$$P_m = \min\{P_{m,j}, j = 1, \dots, n_p\} \quad (3.25)$$

6. Détermination de la nouvelle vitesse pour chaque particule j comme suit :

$$v^j(k) = v^j(k-1) + c_1 r_1 [P_{m,j} - x^j(k-1)] + c_2 r_2 [P_m - x^j(k-1)]; j = 1, \dots, n_p, \quad (3.26)$$

Les paramètres c_1 et c_2 sont généralement égaux à 2, et les paramètres r_1 et r_2 sont des nombres aléatoirement choisis entre 0 et 1.

7. Détermination de la nouvelle position pour chaque particule j comme suit :

$$x^j(k) = x^j(k-1) + v^j(k); j = 1, \dots, n_p \quad (3.27)$$

et évaluation de la fonction objectif pour les nouvelles particules, c'est-à-dire

$$f(x^1(k)), f(x^2(k)), \dots, f(x^{n_p}(k))$$

8. Vérifier la convergence de la solution courante. La convergence de la solution signifie que toutes les particules convergent vers le même ensemble de valeurs. Dans le cas contraire, on pose $k = k + 1$ et reprend à partir de l'étape 5. ■

Pour rester dans un espace de recherche fini donné, alors on ajoute un mécanisme pour éviter qu'une particule ne sorte de cet espace. Le plus fréquent est le confinement d'intervalle. Par exemple pour les contraintes boîtes, on a $x_i^{\min} \leq x_i \leq x_i^{\max}$. Par conséquent, si une position x_i , calculée selon les équations de mouvement, sort de l'intervalle $[x_i^{\min}, x_i^{\max}]$ on lui attribue en fait la valeur du point frontière le plus proche. Pratiquement, cela revient à remplacer l'équation de la position (3.27) par

$$x^j(k) = \min(\max(x^j(k) = x^j(k-1) + v^j, x_{\min}), x_{\max}) \quad (3.28)$$

Pour la vitesse, elle est modifiée soit en remplaçant la composante qui pose problème par son opposé, souvent pondérée par un coefficient inférieur à 1, soit, tout simplement, en l'annulant.

Notons aussi que des améliorations ont été proposées pour améliorer la convergence de la méthode d'essaim de particules et d'adapter la méthode pour des problèmes d'optimisation avec contraintes et multi-objectifs (Banks et al., 2007 ; 2008). Ainsi, plusieurs formules ont été proposées pour le calcul des nouvelles positions des particules, c'est-à-dire la relation (3.26). Chaque méthode introduit un certain nombre de paramètres avec des réglages appropriées (Banks et al., 2007).

Parmi les formules suggérées pour calculer les vitesses des particules, on peut citer les suivantes (Banks et al., 2007). :

Formule 1.

$$v^j(k) = c_0 v^j(k-1) + c_1 r_1 [P_{m,j} - x^j(k-1)] + c_2 r_2 [P_m - x^j(k-1)]; j = 1, \dots, n_p,$$

où c_0 est un paramètre à choisir.

Formule 2.

$$v^j(k) = \frac{2}{\left|2 - c - \sqrt{c^2 - 4c}\right|} (v^j(k-1) + c_1 r_1 [P_{m,j} - x^j(k-1)] + c_2 r_2 [P_m - x^j(k-1)]); j = 1, \dots, n_p,$$

avec $c = c_1 + c_2 > 4$.

Pour illustrer la méthode d'essaim de particules, on considère le problème d'optimisation à une seule variable de décision ($n = 1$) suivant :

$$\begin{aligned} \max_x f(x) &= -x^2 + 2x + 11 \\ \text{Sujet à: } & -2 \leq x \leq 2 \end{aligned}$$

Pour rechercher la solution, on prend un essaim de particules de taille $n_p = 4$. Pour la population initiale, on choisit :

$$x^1(0) = -1.5, \quad x^2(0) = 0.0, \quad x^3(0) = 0.5, \quad x^4(0) = 1.25$$

avec des vitesses initiales nulles

$$v^j(0) = 0, \quad j = 1, 2, 3, 4.$$

Première itération :

L'évaluation de la fonction objectif pour les différentes positions initiales donne :

$$\begin{aligned} f(x^1(0)) &= f(-1.5) = 5.75 \\ f(x^2(0)) &= f(0.0) = 11.0 \\ f(x^3(0)) &= f(0.5) = 11.75 \\ f(x^4(0)) &= f(1.25) = 11.9375 \end{aligned}$$

alors

$$P_{m,1} = -1.5, \quad P_{m,2} = 0.0, \quad P_{m,3} = 0.5, \quad P_{m,4} = 1.25 \quad \text{et} \quad P_m = 1.25.$$

Les nouvelles vitesses des particules calculées par la relation (3.26), avec $c_1 = c_2 = 1$, $r_1 = 0.3294$ et $r_2 = 0.9542$ sont :

$$v^1(1) = 0 + 0.3294(-1.5 + 1.5) + 0.9542(1.25 + 1.5) = 2.6241$$

$$v^2(1) = 0 + 0.3294(0.0 - 0.0) + 0.9542(1.25 - 0.0) = 1.1927$$

$$v^3(1) = 0 + 0.3294(0.5 - 0.5) + 0.9542(1.25 - 0.5) = 0.7156$$

$$v^4(1) = 0 + 0.3294(1.25 - 1.25) + 0.9542(1.25 - 1.5) = 0.0$$

et la relation (3.27) donne les nouvelles positions suivantes :

$$x^1(1) = -1.5 + 2.6241 = 1.1241$$

$$x^2(1) = 0.0 + 1.1927 = 1.1927$$

$$x^3(1) = 0.5 + 0.7156 = 1.2156$$

$$x^4(1) = 1.25 + 0.0 = 1.25$$

L'évaluation de la fonction objectif pour les nouvelles positions des particules donne

$$f(x^1(1)) = 11.9846$$

$$f(x^2(1)) = 11.9629$$

$$f(x^3(1)) = 11.9535$$

$$f(x^4(1)) = 11.9375$$

On constate que les particules convergent presque vers la même valeur, c'est-à-dire vers le voisinage de la solution du problème qui est $f(x^*) = 12$ et $x^* = 1$. Dans le cas contraire, on doit continuer les itérations.

Après avoir présenté les principes des méthodes d'optimisation globale, en l'occurrence la méthode d'Alienor et méthode d'essaim de particules, nous allons utiliser dans la section suivante ces deux méthodes pour l'identification d'un système non linéaire en utilisant la méthode du modèle.

3.4 Application à l'identification d'un système non linéaire

La modélisation mathématique d'un système dynamique est fréquemment utilisée pour l'étude de son comportement dynamique, vis-à-vis de différentes sollicitations, pour l'analyse des propriétés fondamentales (Söderström et Stoica, 1989 ; Ljung, 1990 ; Nelles, 2001 ; Raol et al., 2004 ; Ogunfunmi, 2007) du système (stabilité, commandabilité et observabilité), pour la conception de la loi de commande (Gevers, 1996 ; Soroush, 1998 ; Unbehauen et Rae, 1998) et pour la prédiction (commande prédictive) (Flaus, 1994). La modélisation consiste à décrire sous forme d'équations mathématiques les relations liant les différentes variables caractéristiques du système (entrées, sorties, états et perturbations) (Söderström et Stoica, 1989 ; Ljung, 1990 ; Nelles, 2001 ; Raol et al., 2004 ; Ogunfunmi, 2007). Le modèle peut être

décrit sous forme d'équations différentielles ordinaires, de fonction de transfert (modèle de comportement) ou de représentation d'état (modèle de connaissance) (Flaus, 1994).

Pour l'obtention d'un modèle mathématique deux approches sont possibles (Söderström et Stoica, 1989 ; Ljung, 1990). La première approche, appelée modélisation mathématique, consiste à écrire toutes les lois physiques régissant le fonctionnement du système (relations mathématiques entre les variables du système) en considérant un certain nombre d'hypothèses simplificatrices acceptables (Flaus, 1994). La deuxième approche, appelée modélisation expérimentale, consiste à déterminer un modèle de comportement (fonction de transfert) à partir des mesures entrées-sorties (Söderström et Stoica, 1989 ; Ljung, 1990 ; Nelles, 2001 ; Raol et al., 2004 ; Ogunfunmi, 2007). Les paramètres du modèle sont alors déterminés par des techniques d'identification.

L'identification est une technique expérimentale qui s'appuie sur l'utilisation des procédures et des algorithmes manipulant les mesures expérimentales (Gevers, 1996 ; Dochain, 2003 ; Faber et Wozny, 2007 ; Ljung, 2008 ; Schwaab et al., 2008) et qui a pour but d'ajuster les paramètres du modèle de telle sorte à ce que le comportement du modèle soit identique à celui du système. Les différentes méthodes d'identification existantes peuvent être scindées en trois classes (Flaus, 1994). La première classe regroupe les méthodes graphiques (par exemple la méthode de Stretj), la seconde les méthodes non récursives (comme la méthode des moindres carrés et la méthode du modèle) et la troisième les méthodes récursives (par exemple la méthode des moindres carrés récursifs).

La méthode du modèle reste l'une des méthodes d'identification la plus efficace (Söderström et Stoica, 1989 ; Ljung, 1990 ; Nelles, 2001 ; Raol et al., 2004 ; Ogunfunmi, 2007). Son principe consiste à proposer, en se référant aux résultats expérimentaux, une structure (fonction de transfert) pour le modèle puis de formuler un problème d'optimisation dont la fonction objectif consiste à minimiser l'écart entre les mesures expérimentales et celles prédites par le modèle (Flaus, 1994). Les paramètres du modèle représentent les variables de décision dans ce cas (Söderström et Stoica, 1989 ; Ljung, 1990 ; Nelles, 2001 ; Raol et al., 2004 ; Ogunfunmi, 2007). L'algorithme des moindres carrés constitue un exemple de la méthode du modèle, il est très utilisé dans le cas où le modèle est linéaire par rapport aux paramètres à identifier (Flaus, 1994 ; Gevers, 1996). Dans le cas général, des méthodes exactes ou heuristiques sont adoptées pour la résolution du problème d'optimisation formulé (Unbehauen et Rae, 1998 ; Ljung, 2004). Celles-ci sont généralement des méthodes exactes qui peuvent échouer dans un minimum local (Ikonen and K. Najim, 2002). Bien que les méthodes heuristiques constituent une alternative intéressante (Ursema et Vadstrup, 2004),

pour une meilleure identification (localisation du voisinage d'un minimum global), mais leur réglage est très délicat et le temps de convergence peut être important pour certaines applications. Les méthodes heuristiques utilisées sont, généralement, les algorithmes génétiques (Nyarko et Scitovski, 2004 ; Patelli and Ferariu, 2009) et l'essaim de Particules (Schwaab et al., 2008 ; Sakhivel et al. 2010)

Afin de surmonter ce problème, nous proposons dans cette section d'adopter les deux méthodes d'optimisation globale présentées dans ce chapitre pour l'identification globale des paramètres d'un système dynamique.

3.4.1 Méthode du modèle

Le principe de la méthode du modèle est donné par la Figure 3.3 (Flaus, 1994). L'idée consiste à minimiser l'écart observé entre les mesures expérimentales $y(k)$ et celles prédites par le modèle $\hat{y}(k)$ en utilisant un algorithme d'optimisation qui adapte les paramètres du modèle d'une manière à avoir l'erreur $e(k) = y(k) - \hat{y}(k)$ proche de zéro pour toute valeur de k qui désigne l'instant d'échantillonnage. Dans ce cas le modèle et le système physique sont soumis à une même entrée $u(k)$ supposée suffisamment excitante pour toutes les dynamiques du système (par exemple un signal SBPA) (Söderström et Stoica, 1989 ; Ljung, 1990 ; Nelles, 2001; Raol et al., 2004 ; Ogunfunmi, 2007). Par conséquent, pour l'identification des paramètres, on considère la minimisation d'une fonction objectif (critère) qui dépend explicitement de l'erreur $e(k)$. Généralement, cette fonction objectif prend la forme suivante :

$$J = \sum_{k=0}^{+\infty} f(e(k)) = \sum_{k=0}^{+\infty} f(y(k) - \hat{y}(k)) \quad (3.29)$$

où f est une fonction non linéaire qui mesure l'écart entre les deux sorties mesurée et prédite, i.e. $y(k)$ et $\hat{y}(k)$.

La sortie prédite $\hat{y}(k)$ du modèle, dans le cas discret, est donnée comme suit :

$$\hat{y}(k) = g(x(k), \theta) \quad (3.30)$$

avec

$$x(k) = [u(k), \dots, u(k - r_u), y(k), \dots, y(k - r_y)] \quad (3.31)$$

où $u(k)$ et $y(k)$ représentent respectivement la commande et la sortie du système. $x(k)$ représente le vecteur de régression. r_u et r_y représentent respectivement les retards tolérés pour

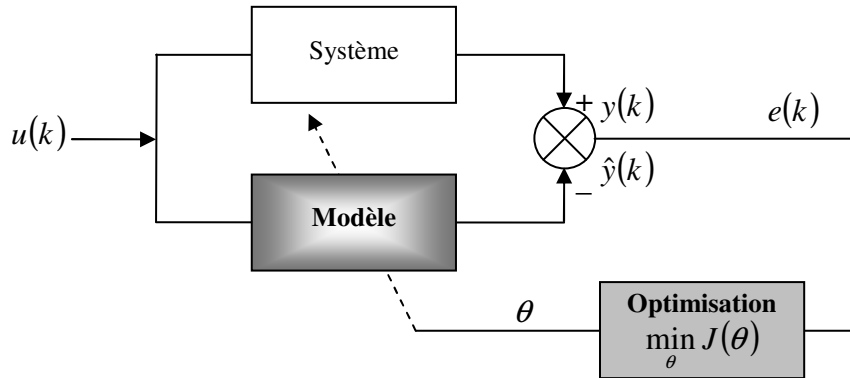


Figure 3.3 Principe de la méthode du modèle.

l'entrée et la sortie du système et θ représente le vecteur des paramètres du modèle à identifier. Ce vecteur est de dimension n .

Ainsi, en remplaçant $\hat{y}(k)$ par son expression (3.30) dans la fonction objectif (3.29), il vient :

$$J = \sum_{k=0}^{+\infty} f(y(k) - g(x(k), \theta)) \quad (3.32)$$

Comme les mesures $y(k)$ et les commandes $u(k)$ sont disponibles, il est clair que le critère J dépend explicitement du vecteur des paramètres θ à identifier. Par conséquent,

$$J(\theta) = \sum_{k=0}^{+\infty} f(y(k) - g(x(k), \theta)) \quad (3.33)$$

Le problème d'identification des paramètres, regroupés dans le vecteur désigné par θ , se ramène à la résolution du problème d'optimisation globale suivant :

$$\min_{\theta} J(\theta) = \min_{\theta} \sum_{k=0}^{+\infty} f(y(k) - g(x(k), \theta)) \quad (3.34)$$

La complexité du problème d'optimisation dépend de la fonction f considérée. Généralement, cette fonction prend la forme quadratique telle que :

$$f(e(k)) = e^2(k) = (y(k) - \hat{y}(k))^2 \quad (3.35)$$

d'où le critère à minimiser est

$$J(\theta) = \sum_{k=0}^{+\infty} (y(k) - g(x(k), \theta))^2 \quad (3.36)$$

Ce problème d'optimisation peut être résolu par plusieurs méthodes qui sont généralement itératives. Cependant, ces méthodes numériques peuvent tomber dans un minimum local et

deviennent très délicates lorsque le nombre de variables est très élevé. A fin d'y remédier à tous ces problèmes, nous proposons de chercher le minimum du critère (3.36) par les deux méthodes d'optimisation globale présentées dans ce chapitre.

3.4.2 Modèle du système non linéaire à identifier

Le modèle du système non linéaire à identifier est donné sous forme d'une équation de récurrence (modèle entrée-sortie discret). Pour générer les mesures, on procède par simulation. On propose de fixer d'abord des valeurs pour les paramètres du modèle, puis de faire une simulation sur un horizon de temps déterminé pour récupérer les valeurs de la sortie correspondante à une entrée convenablement choisie (Söderström et Stoica, 1989 ; Ljung, 1990 ; Flaus, 1994 ; Nelles, 2001; Raol et al., 2004 ; Ogunfunmi, 2007). Les valeurs obtenues pour la sortie seront considérées comme les mesures expérimentales et seront utilisées pour l'identification.

Le modèle du système non linéaire à identifier est donné comme suit :

$$y(k+2) = a y(k+1) + b y(k)u(k+1) + cu(k+1) \quad (3.37)$$

Pour obtenir les mesures expérimentales, les valeurs suivantes $a = 0,9, b = -0,7$ et $c = 0,5$ ont été fixées pour les paramètres du modèle. Ainsi, le vecteur de paramètres à identifier est

$$\theta = (\theta_1, \theta_2, \theta_3)^T = (a, b, c)^T \quad (3.38)$$

Pour l'identification de système, treize mesures ont été considérées en prenant $y(0) = y(1) = 0$ et une entrée $u(k)$ aléatoire.

Pour l'identification, en utilisant la méthode d'Alienor pour la résolution du problème d'optimisation (3.36), on utilise la transformation réductrice (3.17) avec les paramètres suivants $\omega_0 = 100$, $\varphi_0 = 1$, et $\varepsilon = 5 \times 10^{-5}$. Ainsi, en explicitant les variables de décision θ_i , regroupé dans le vecteur θ , en fonction de la seule variable de décision α , à l'aide de la transformation réductrice (3.17), le problème d'identification (3.36) se réduit au problème d'optimisation à une seule variable de décision α suivant :

$$\min_{\alpha} \hat{J}(\theta) = \min_{\alpha} \sum_{k=0}^{+\infty} (y(k) - g(x(k), h(\alpha)))^2 \quad (3.39)$$

où $h(\alpha)$ est une fonction vectorielle donnée comme suit :

$$h(\alpha) = \begin{bmatrix} \theta_1 = h_1(\alpha) \\ \theta_2 = h_2(\alpha) \\ \theta_3 = h_3(\alpha) \end{bmatrix} = \begin{bmatrix} \cos(100\alpha + 1.00000) \\ \cos(101\alpha + 1.00005) \\ \cos(102\alpha + 1.00010) \end{bmatrix} \quad (3.40)$$

Le problème d'optimisation à résoudre est le suivant :

$$\min_{\alpha} \sum_{k=0}^{11} (y(k+2) - \cos(100\alpha + 1)y(k+1) - \cos(101\alpha + 1.00005)y(k)u(k+1) - \cos(102\alpha + 1.00010)u(k+1))^2 \quad (3.41)$$

La solution de problème a été réalisée en utilisant la méthode des évaluations simultanées. Dans cette méthode, on fixe, à priori, le nombre de points équidistants auxquels on évaluera la valeur du critère (3.41) (points supports de l'évaluation). L'application de la méthode conduit à l'optimum global $\alpha^* = 3,4501$. En remplaçant α^* dans les relations (3.40), on obtient les paramètres du modèle suivants :

$$\theta_1^* = a^* = 0,9055; \theta_2^* = b^* = -0,7339; \theta_3^* = c^* = +0,4929.$$

La valeur du critère est :

$$J(\alpha^*) = 4,4547 \times 10^{-7}.$$

Cet exemple montre que la méthode d'Alienor permet d'identifier de manière précise les paramètres du système, puisque les valeurs estimées sont très proches des valeurs réelles.

Pour l'identification en utilisant la méthode d'essai de particules pour la résolution du problème d'optimisation (3.36) exprimé en fonctions des paramètres du modèle comme suit :

$$\min_{\alpha} \sum_{k=0}^{11} (y(k+2) - a y(k+1) - b y(k)u(k+1) - c u(k+1))^2 \quad (3.42)$$

Par conséquent, une particule (position) est donnée comme suit :

$$\theta^j(k) = \begin{bmatrix} a^j(k) \\ b^j(k) \\ c^j(k) \end{bmatrix} \quad (3.43)$$

La population considérée est de taille $n_p = 5$ dont les particules (positions) initiales sont :

$$\theta^1(0) = \begin{bmatrix} 0,2 \\ -1,5 \\ 0,0 \end{bmatrix}, \theta^2(0) = \begin{bmatrix} 0,5 \\ -1,0 \\ 0,1 \end{bmatrix}, \theta^3(0) = \begin{bmatrix} 1,0 \\ -0,6 \\ 0,3 \end{bmatrix}, \theta^4(0) = \begin{bmatrix} 1,25 \\ -0,3 \\ 0,6 \end{bmatrix}, \theta^5(0) = \begin{bmatrix} 1,50 \\ -0,0 \\ 1,80 \end{bmatrix}$$

avec des vitesses initiales nulles

$$v^j(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, j = 1, \dots, 5$$

Le critère d'arrêt utilisé est le nombre d'itération maximum à effectuer fixé dans notre application à 500 itérations. Notons que l'augmentation du nombre d'itération n'améliore pas les résultats. La solution obtenue est :

$$\theta_1^* = a^* = 0,979; \theta_2^* = b^* = -0,61; \theta_3^* = c^* = +0,33.$$

La valeur du critère est :

$$J(\theta^*) = 9.0878 \times 10^{-5}$$

En comparant les résultats de la méthode d'essai de particules à ceux obtenus avec la méthode d'Alienor, on constate que la méthode d'Alienor donne un résultat meilleur par rapport à la méthode d'essai de particules. Ceci est évident, puisque la méthode d'Alienor est une méthode exacte, alors que la méthode d'essai de particules est une méthode heuristique dont la convergence est garantie que de manière asymptotique. C'est pour cette raison qu'une méthode heuristique doit être toujours épaulée par une méthode exacte pour atteindre l'optimum global avec une très grande précision.

Sur le plan temps de calcul, qui constitue un critère de comparaison important, on constate que la méthode d'Alienor est plus performante que la méthode d'essai de particules. En effet les résultats obtenus, en utilisant une machine de fréquences de 1.83 GHz, montre que la méthode d'Alienor mis un temps de 7.120454 s pour converger vers la solution, par contre la méthode d'essai de particules mis un temps de 148.63 s pour atteindre de manière asymptotique la solution.

3.5 Conclusion

Dans ce chapitre, nous avons présenté deux méthodes d'optimisation globale, une exacte et l'autre heuristique. La méthode exacte est la méthode d'Alienor dont le principe consiste à utiliser une transformation réductrice pour ramener le problème d'optimisation initiale à plusieurs variables à un problème d'optimisation à une seule variable dont la résolution est simple, et l'obtention de l'optimum global avec une très grande précision est assurée. La méthode heuristique est la méthode d'essai de particules fondée sur la notion de coopération entre particules qui peuvent être vues comme des êtres vivants aux capacités intellectuelles assez limitées (peu de mémoire et de facultés de raisonnement). Néanmoins, par échange d'informations entre eux fait que, globalement, ils arrivent à résoudre des problèmes

difficiles. Ce comportement a été formalisé sous forme d'un algorithme exploité pour la résolution des problèmes d'optimisation non linéaires difficiles. Cet algorithme est très simple à comprendre, à programmer et à utiliser, et se révèle très efficace.

Pour montrer l'efficacité de ces deux méthodes d'optimisation globale, une application à l'identification des paramètres d'un modèle mathématique a été présentée. L'objectif est d'estimer de manière globale les paramètres d'un modèle d'un système non linéaires à partir des mesures réalisées. L'application a permis de mettre en évidence les propriétés de chaque méthode. Ainsi, les résultats obtenus, montre la convergence exacte de la méthode d'Alienor, et la convergence asymptotique de la méthode d'essai de particules. Mais les deux méthodes s'avèrent très intéressantes pour localiser l'optimum global d'une fonction objectif non linéaire.

Après avoir présenter et démontrer l'efficacité des deux méthodes d'optimisation globale, nous allons expérimenter ces deux méthodes pour la résolution du problème d'optimisation dans une stratégie de commande prédictive d'un système non linéaire. Cette étude fera l'objet du chapitre suivant.

Chapitre 4

Commande Prédicative Non linéaire Basée sur l'Optimisation Globale

4.1 Introduction

La commande prédictive ne désigne pas une stratégie de commande spécifique mais constitue un ensemble d'algorithmes de commande qui utilisent le modèle du système (pour la prédiction) dans un problème d'optimisation qui consiste à minimiser un critère de performance (Huang et Kadali, 2008). La résolution du problème d'optimisation ainsi formulé permet de calculer une séquence de valeurs pour la variable manipulée sur un horizon appelé horizon de commande où seule la première commande sera appliquée au système (Cannon, 2004 ; Findeisen et al., 2007).

Pour la commande des systèmes linéaires, la solution globale est obtenue facilement puisque le problème revient à résoudre un problème de programmation quadratique comme

dans le cas de la méthode DMC (*Dynamic Matrix Control*) (Flaus, 1994 ; Huang et Kadali, 2008). Par contre pour les systèmes non linéaires le problème d'optimisation est généralement non linéaire à cause de la prédiction (modèle) (Henson, 1998 ; Long et al., 2006). Par conséquent, l'optimum globale est très difficile à obtenir (Kouvaritakis et Cannon, 2001; Cannon, 2004 ; Findeisen et al., 2007 ; Magni et al., 2009). Comme la résolution du problème d'optimisation doit être réalisée pendant une durée égale à une période d'échantillonnage, alors la commande prédictive des systèmes non linéaires nécessite des algorithmes d'optimisation globale qui doivent être convergents et rapides (Cannon, 2004 ; Magni et al., 2009).

Bien que la théorie d'optimisation globale offre une panoplie d'algorithmes (Cannon, 2004 ; Findeisen et al., 2007 ; Huang et Kadali, 2008), leur utilisation dans une stratégie de commande prédictive est conditionnée par le temps de convergence qui ne doit pas dépasser une période d'échantillonnage (Cannon, 2004 ; Magni et al., 2009). Par conséquent, leur application à des systèmes rapides et des systèmes fortement non linéaires est limitée (Henson, 1998 ; Cannon, 2004).

Les différentes contributions dans le cadre de la commande prédictive non linéaire se sont focalisées sur le type du modèle utilisé pour la prédiction et le développement d'algorithmes d'optimisation globale rapides.

Dans ce chapitre, les deux méthodes d'optimisation globale, en l'occurrence la méthode d'Alienor et la méthode d'essaim de particules seront utilisées pour la commande prédictive, du système non linéaire identifié dans le chapitre précédent en considérant des contraintes boîtes pour la variable de commande. Avant de rentrer dans le vif du sujet, on commence d'abord par la présentation d'un état de l'art de la commande prédictive non linéaire.

4.2 Etat de l'art de la commande prédictive non linéaire

Dans une commande prédictive, l'étape la plus importante est la résolution du problème d'optimisation qui consiste à minimiser un critère de performances. Le vecteur de variables de décision représente la séquence de la commande déterminée pour un horizon de commande. Néanmoins, seule la première commande sera appliquée au système. Par conséquent à chaque instant d'échantillonnage, la séquence de commande optimale doit être calculée ce qui nécessite l'utilisation des algorithmes d'optimisation très rapides (Cannon, 2004 ; Magni et al., 2009).

Dans toutes les contributions concernant la commande prédictive non linéaire, le principe de commande reste le même, c'est-à-dire un algorithme de commande prédictive est composé de trois étapes essentielles à réaliser, à chaque période d'échantillonnage, qui sont : la prédiction (utilisation d'un modèle mathématique), l'optimisation (utilisation d'une méthode d'optimisation) et application de la commande. Les différentes approches proposées dans la littérature peuvent être scindées en deux groupes, selon le type du modèle utilisé pour la prédiction (Findeisen et al., 2007), et selon la méthode d'optimisation adoptée pour localiser la séquence de commande optimale (Cannon, 2004).

La sélection d'un modèle souhaitable pour la prédiction des sorties futures joue un rôle non négligeable dans une stratégie commande prédictive non linéaire. Les modèles utilisés sont, généralement, du type entrée-sortie et du type d'état. Parmi les modèles entrée-sortie, on retrouve le modèle NARMAX donnée sous forme d'équation aux différences (Nelles, 2001), le modèle de Volterra, le modèle neuronal (réseau de neurones), le modèle flou obtenu par linéarisation autour de plusieurs points de fonctionnement (Tatjewski, 2007). Le choix du modèle est lié directement au temps de calcul, puisque une prédiction des sorties avec précision impose l'utilisation d'un modèle précis donc complexe, ce qui complique les calculs des prédictions. Par conséquent, un compromis doit être fait entre précision et complexité de calculs. Le modèle entrées-sorties représente une variante très intéressante pour le problème de prédiction. Ces modèles donnent une prédiction de qualité acceptable, et l'estimation de leurs paramètres peut être réalisée en ligne.

Une fois le modèle sélectionné, la réussite d'une commande prédictive est directement liée à l'algorithme d'optimisation utilisé. Les exigences imposent que l'algorithme d'optimisation doit être convergent, à défaut vers un minimum local, pour assurer la stabilité en boucle fermée (Mayne et al., 2000), et en tenant de la contrainte du temps, cet algorithme doit être rapide, c'est-à-dire la séquence de commande optimale doit être calculée et disponible en un temps inférieur à la période d'échantillonnage. Magni et al. (2009) présente une synthèse sur les différentes approches d'optimisation utilisées dans le cadre de la commande prédictive non linéaire et les différentes applications.

Parmi les méthodes d'optimisation utilisées on retrouve les méthodes directes dont le principe consiste à approcher le problème d'optimisation non linéaire par une séquence de problèmes d'optimisation quadratique (Berthiau et Siarry, 2001). Chaque problème d'optimisation convexe peut être résolu par la suite par une méthode numérique (méthode de Powell, méthode de Newton et ses variantes, et méthode du point intérieur). Quoique la solution puisse être facilement obtenue, l'inconvénient est lié au temps de calcul nécessaire

pour localiser la solution optimale. Ainsi, la séquence des problèmes d'optimisation convexe doit être résolu pendant un temps égale à une période d'échantillonnage, ce qui n'est pas aisé à faire surtout pour les systèmes caractérisés par une dynamique rapide, et en présence de contraintes. De plus la convergence à l'optimum global n'est pas aussi garantie. Pour surmonter ces difficultés, au lieu de l'approche séquentielle adoptée pour la résolution des problèmes d'optimisation convexe, des méthodes faisant partie de l'approche simultanée ont été proposées pour résoudre ces problèmes d'optimisation.

Une autre classe de méthodes proposées pour la résolution d'un problème de commande prédictive non linéaire s'inspirent de la commande optimale. Ainsi, le problème de commande prédictive sur un horizon de commande est vu comme un problème de commande optimale à horizon fini (égale à l'horizon de commande) qu'on doit résoudre en utilisant le principe du minimum ou la programmation dynamique.

L'examen de la littérature concernant la commande prédictive non linéaire, montre que l'application des méthodes exactes d'optimisation globale, et les méthodes heuristiques dans une stratégie de commande prédictive non linéaire est un axe au stade de balbutiement. Dans ce chapitre, on s'intéresse à l'utilisation des méthodes d'Alienor et d'essaim de particules dans une stratégie de commande prédictive d'un système non linéaire.

4.3 Commande prédictive non linéaire

La commande prédictive est une commande optimale calculée sur un horizon glissant dont l'objectif de poursuite est formulé par un problème d'optimisation non linéaire (Henson, 1998 ; Tatjewski, 2007). Ainsi, un critère de performance mesurant l'écart entre la consigne désirée ou la trajectoire de référence et la sortie commandée du système, sur un horizon de prédiction fini, doit être minimisé (Huang et Kadali, 2008). Généralement, ce critère prend la forme quadratique, mais comme le système est non linéaire, alors on obtient un problème d'optimisation non linéaire (Cannon, 2004). Le principe de la commande prédictive non linéaire est résumé par la Figure 4.1 (Tatjewski, 2007).

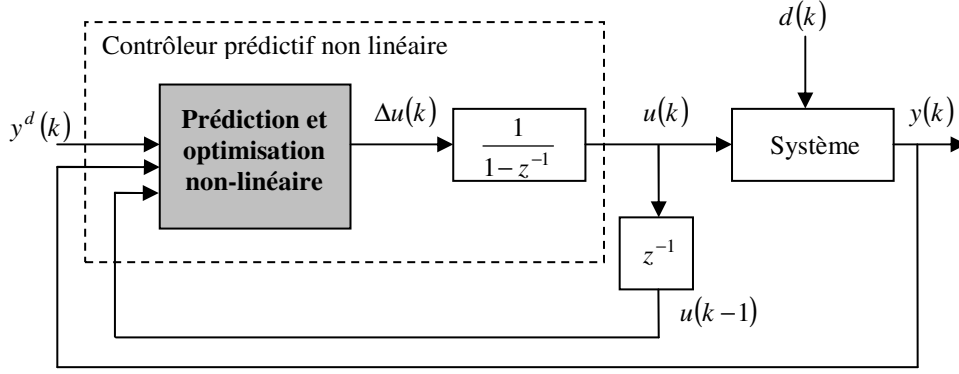


Figure 4.1. Structure générale de la commande prédictive non linéaire.

La commande prédictive fait partie de la famille des commandes basée sur le modèle (Flaus, 1994 ; Findeisen et al., 2007 ; Huang et Kadali, 2008). Ce dernier est utilisé pour la prédiction du comportement future de la sortie du système pour anticiper et prendre la bonne décision (commande). Pour le cas des systèmes non linéaires, plusieurs formes de modèles peuvent être considérées : modèle entrée-sortie (équation aux différences), modèle de Volterra et modèle réseau de neurone (Kouvaritakis et Cannon, 2001 ; Findeisen et al., 2007).

Dans ce mémoire, on considère le modèle entrée-sortie donné par l'équation générale :

$$y(k) = \Phi(y(k-1), \dots, y(k-n_y), u(k-1), \dots, u(k-n_u), d(k-1), \dots, d(k-n_d)) \quad (4.1)$$

où y est la sortie du système, u est la variable de commande, d est la perturbation et Φ est une fonction non linéaire.

Le critère à minimiser englobe généralement deux objectifs : la poursuite et l'énergie minimale (Flaus, 1994 ; Cannon, 2004 ; Huang et Kadali, 2008).

$$J(U) = \underbrace{\sum_{j=0}^{N_p} [y^d(k+j) - \hat{y}(k+j/k)]^T Q_j [y^d(k+j) - \hat{y}(k+j/k)]}_{\text{Poursuite}} + \underbrace{\sum_{j=0}^{N_u-1} [\Delta u(k+j)]^T R_j [\Delta u(k+j)]}_{\text{Energie minimale}} \quad (4.2)$$

N_p et N_u représentent respectivement les horizons de prédiction et de commande. y^d représente la consigne ou la trajectoire désirée et \hat{y} représente la sortie prédite par le modèle (4.1). Q et R sont des matrices de pondération qui peuvent être constantes ou variables. $k+j$ représente les instants de prédictions et k est l'instant actuel. $\hat{y}(k+j/k)$ représente la sortie prédite à l'instant d'échantillonnage k pour le future instant $k+j$. U représente le vecteur regroupant la séquence des commandes à appliquer aux instants d'échantillonnage dont le nombre est égale à N_u , i.e. :

$$U = [u(k), u(k+1), \dots, u(k+N_u-1)] \quad (4.3)$$

Notons que seule la commande $u(k)$ sera appliquée au système à l'instant k . Pour déterminer cette dernière, on doit résoudre, en un temps égal à la période d'échantillonnage, le problème d'optimisation non linéaire suivant (Cannon, 2004) :

$$\begin{aligned} \min_U J(U) = & \sum_{j=0}^{N_p} [y^d(k+j) - \hat{y}(k+j/k)]^T Q_j [y^d(k+j) - \hat{y}(k+j/k)] \\ & + \sum_{j=0}^{N_u-1} [\Delta u(k+j)]^T R_j [\Delta u(k+j)] \\ \text{sujet à : } & y(k) = \Phi(y(k-1), \dots, y(k-n_y), u(k-1), \dots, u(k-n_u), d(k-1), \dots, d(k-n_d+1)) \\ & |u(k)| \leq M \end{aligned} \quad (4.4)$$

La solution recherchée revient à localiser l'optimum global pour le critère $J(U)$ en un temps égal à la période d'échantillonnage. Comme le problème d'optimisation (4.4) est généralement fortement non linéaire, localiser l'optimum global en un temps faible est généralement très difficile voire impossible pour certains systèmes dynamiques (Kouvaritakis and Cannon, 2001 ; Cannon, 2004 ; Findeisen et al., 2007 ; Tatjewski, 2007 ; Magni et al., 2009). Plusieurs algorithmes d'optimisation ont été adoptés dans la littérature (Martinsen et al., 2002 ; Cannon, 2004). Une synthèse des différentes algorithmes utilisés pour la commande prédictive des systèmes non linéaires est présentée dans (Cannon, 2004 ; Findeisen et al., 2007 ; Magni et al., 2009). Notons qu'il est difficile de recommander de manière générale un algorithme car chaque algorithme présente ces propres limites et ses conditions d'applicabilité sont difficiles à préciser (Findeisen et al., 2007).

Dans ce qui suit, les deux méthodes d'optimisation globale présentées dans le chapitre précédent seront adoptées pour la commande prédictive du système non linéaire, identifié dans le même chapitre, c'est-à-dire pour la résolution du problème d'optimisation non linéaire (4.4) dont le nombre de variables de décision est égal à l'horizon de commande. L'objectif est d'assurer la poursuite de consigne en respectant une contrainte boîte sur la commande.

4.3.1 Commande prédictive basée sur la méthode d'Alienor

La méthode Alienor est adoptée dans la stratégie de commande prédictive pour la résolution du problème d'optimisation non linéaire (4.4). Le vecteur de variables de décision représente les valeurs de la commande à des instants d'échantillonnage allant de k jusqu'à

$k + N_u - 1$ avec $N_u < N_p$. Pour $k + N_u, \dots, k + N_p$, on maintient la commande constante à une valeur égale à $u(k + N_u - 1)$ (Tatjewski, 2007).

Dans ce cas, le vecteur de variables de décision U , donné par la relation (4.3), est de dimension N_u avec $u(k + N_u - 1) = u(k + N_u) = \dots = u(k + N_p)$. La résolution du problème d'optimisation (4.4) n'est pas facile à obtenir et bute, généralement, aux deux problèmes suivants (Martinsen et al., 2002 ; Cannon, 2004) : la convergence est lente lorsque l'horizon de commande est grand, et la plupart des algorithmes d'optimisation, généralement numériques, peuvent être piégés dans un minimum relatif au lieu d'un minimum global ce qui n'assure pas de bonnes performances. Les algorithmes d'optimisation globale n'échappent pas à la complexité des calculs (Cannon, 2004 ; Findeisen et al., 2007). Afin d'éviter ces problèmes, la méthode d'optimisation globale d'Alienor est adoptée pour la résolution du problème d'optimisation (4.4). Cette méthode permet de réduire davantage les calculs puisque le problème se voit simplifié en le ramenant à un problème d'optimisation à une seule variable (Cherruault et Mora, 2005). Ainsi, il est facile d'établir, qu'à partir du modèle, que les prédictions des sorties $\hat{y}(k + j/k)$ seront fonction des variables de commande $u(k), \dots, u(k + N_u)$. Par conséquent, le critère de performance (4.2) prend la forme suivante :

$$\hat{J}(U) = F(u(k), \dots, u(k + N_u - 1)) \quad (4.5)$$

et le problème revient à rechercher, à chaque instant d'échantillonnage k , la solution du problème d'optimisation équivalent suivant :

$$\min_{u(k), \dots, u(k + N_u)} \hat{J}(U) = F(u(k), \dots, u(k + N_u - 1)) \quad (4.6)$$

C'est justement pour simplifier la résolution qu'on propose d'utiliser la méthode d'Alienor. Dans cette optique, chaque variable de décision $u(i)$ pour $i = k, \dots, k + N_u - 1$ sera définie par une transformation réductrice

$$u(i) = h_{i-k+1}(\alpha) \quad (4.7)$$

et le problème d'optimisation (4.6) prend la forme suivante

$$\min_{\alpha} G(\alpha) = F(h_1(\alpha), \dots, h_{N_u}(\alpha)) \quad (4.8)$$

ce dernier est à une seule variable de décision, en l'occurrence α , dont l'obtention de l'optimum global α^* , à chaque instant d'échantillonnage k , est très simple à réaliser. D'après la relation (4.7), la commande à appliquer à chaque instant d'échantillonnage k est donnée comme suit :

$$u(k) = h_1(\alpha^*) \quad (4.9)$$

On remarque bien que la méthode d'Alienor facilite l'implémentation de la commande prédictive pour un système non linéaire. L'approche proposée sera illustrée pour la commande du système non linéaire, identifié dans le chapitre précédent, suivant :

$$y(k) = 0.9y(k-1) - 0.7y(k-2)u(k-1) + 0.5u(k-1) \quad (4.10)$$

L'objectif est de concevoir un correcteur prédictif pour assurer une poursuite de consigne y^d , supposée constante, ce qui revient à minimiser le critère suivant ($Q=1$ et $R=0$ dans le critère (4.2)) :

$$J = \sum_{j=1}^{N_p} (y^d(k) - \hat{y}(k+j/k))^2 \quad (4.11)$$

Ainsi, le problème d'optimisation à résoudre à chaque instant d'échantillonnage k est formulé comme suit :

$$\min_U J(U) = \sum_{j=1}^{N_p} (y^d(k) - \hat{y}(k+j/k))^2 \quad (4.12)$$

Les prédictions $\hat{y}(k+j/k)$ sont données par le modèle (4.10) :

$$y(k+j/k) = 0.9\hat{y}(k+j-1/k) - 0.7\hat{y}(k+j-2/k)u(k+j-1/k) + 0.5u(k+j-1/k) \quad (4.13)$$

En considérant $N_p = 7$ et $N_u = 4$, le problème d'optimisation s'écrit alors :

$$\min_U \sum_{j=1}^7 (y^d(k) - 0.9\hat{y}(k+j-1/k) + 0.7\hat{y}(k+j-2/k)u(k+j-1/k) - 0.5u(k+j-1/k))^2 \quad (4.14)$$

Pour résoudre le problème d'optimisation (4.14) par la méthode d'Alienor, on considère la transformation réductrice (3.17). Ainsi :

$$u(i) = \beta \cos(\omega_{i-k+1}\alpha + \varphi_{i-k+1}), \quad i = 1, \dots, 4. \quad (4.15)$$

En substituant (4.15) dans les prédictions données par la relation de récurrence (4.13) puis dans le problème d'optimisation (4.14), on obtient le problème d'optimisation à une seule variable α suivant :

$$\min_{\alpha} \sum_{j=1}^7 (y^d - \Psi(\alpha))^2 \quad (4.16)$$

où Ψ est une fonction non linéaire à une seule variable α . Par conséquent, la commande à appliquer à chaque instant k est déduite de (4.15) comme suit :

$$u(k) = \beta \cos(\omega_1 \alpha^* + \varphi_1) \quad (4.17)$$

où α^* est l'optimum global du problème d'optimisation (4.16) calculé à chaque instant d'échantillonnage k . La période d'échantillonnage considérée est de 20 s.

Les paramètres utilisés pour la transformation réductrice (4.15) sont :

$$\beta = 2, \omega_l = 100 + (l - 1), \varphi_l = 1 + 0,0005(l - 1); l = 1, \dots, 4. \quad (4.18)$$

Rappelons que le but recherché par la commande est d'assurer la poursuite d'une consigne constante $y^d = 1$. Les résultats de simulation sont donnés par la Figure 4.2. On constate que la sortie (Fig. 4.2a) poursuit convenablement la consigne désirée avec des variations admissibles pour la commande (Fig. 4.2b).

Maintenant pour le même problème de commande précédent, on considère que la variable de commande $u(k)$ est soumise à la contrainte suivante :

$$|u(k)| \leq 1,25 \quad (4.19)$$

Dans ce cas, il suffit de jouer sur le réglage des paramètres de la transformation réductrice (4.15). Ainsi, pour respecter la contrainte (4.19), il faut prendre

$$\beta = 1,25. \quad (4.20)$$

Les résultats de simulation obtenus sont donnés par la Figure 4.3. On constate que la sortie atteint la consigne imposée après un certain nombre de périodes d'échantillonnage (Fig. 4.3a). Pour la commande, on constate que la contrainte est respectée (Fig. 4.3b) puisque les valeurs de cette dernière ne dépassent pas la limite 1,25. L'examen des Figures 4.2b et 4.3b montre qu'en absence de contrainte, pour atteindre la consigne désirée $y^d = 1$, la commande générée prend des valeurs supérieures à 1,25. Par contre en imposant la contrainte sur la commande et pour atteindre la même consigne $y^d = 1$, la commande générée reste dans le domaine admissible, i.e. $u(k) \leq 1,25, \forall k$. En somme, la méthode d'Alienor permet de prendre aisément les contraintes sur la commande sans transformer le problème d'optimisation par introduction des variables d'écart ou l'utilisation des conditions de Kuhn-Tucker.

4.3.2 Commande prédictive basée sur la méthode d'essaim de particules

Pour résoudre le problème d'optimisation par la méthode d'essaim de particules à chaque itération, nous avons considéré les paramètres de réglage de l'algorithme de la commande prédictive suivants : $N_p = 7$ et $N_u = 4$, le problème d'optimisation (4.14) s'écrit alors :

$$\min_U \sum_{j=1}^7 (y^d(k) - \Gamma(U))^2 \quad (4.21)$$

et une particule (position) est donnée comme suit :

$$U^j = \begin{bmatrix} u^j(k) \\ u^j(k+1) \\ u^j(k+2) \\ u^j(k+3) \end{bmatrix} \quad (4.22)$$

La taille de la population considérée est $n_p = 5$ dont les particules (positions) initiales, $k = 0$, sont choisies

$$U^1(0) = \begin{bmatrix} 0,2 \\ -1,5 \\ 0,0 \\ 0,0 \end{bmatrix}, \quad U^2(0) = \begin{bmatrix} 0,5 \\ -1,0 \\ 0,1 \\ 0,1 \end{bmatrix}, \quad U^3(0) = \begin{bmatrix} 1,0 \\ -0,6 \\ 0,3 \\ 0,3 \end{bmatrix}, \quad U^4(0) = \begin{bmatrix} 1,25 \\ -0,30 \\ 0,60 \\ 0,60 \end{bmatrix}, \quad U^5(0) = \begin{bmatrix} 1,50 \\ 0,00 \\ 1,80 \\ 1,80 \end{bmatrix}$$

avec des vitesses initiales nulles

$$V^j(0) = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad j = 1, \dots, 5$$

A chaque instant d'échantillonnage k , les particules initiales (positions) sont prises égales aux meilleures solutions précédentes, c'est-à-dire,

$$U^j(0) = U^j(20), \quad j = 1, \dots, 5.$$

Et, à chaque instant d'échantillonnage k , les vitesses initiales sont considérées nulles. Comme critère d'arrêt de l'algorithme d'essaim de particules, on utilise un nombre d'itérations maximal égale à 20. Une valeur supérieure n'améliore pas les résultats.

Le but recherché est d'assurer la poursuite de la consigne $y^d = 1$. Pour le cas sans contrainte sur la variable manipulée, les résultats obtenus sont donnés par la Figure 4.4. On constate que, la sortie atteint la consigne désirée (Fig. 4.4a) mais avec un régime transitoire comparativement long par rapport au résultat obtenu par la méthode d'Alienor (Fig. 4.2a). Ceci peut être expliqué par le fait que la méthode d'optimisation globale d'essaim de particules permet de localiser que de manière asymptotique l'optimum globale, ce qui donne un écart un peu important, mais qui décroît à chaque instant d'échantillonnage, pour devenir nul après un certains nombre de périodes d'échantillonnage. Aussi, on constate que l'évolution de la variable de commande est physiquement acceptable (Fig. 4.4b).

Maintenant, nous allons imposer une contrainte sur la variable de commande. Comme la méthode d'essaim de particules est caractérisée par une convergence asymptotique, par

conséquent en présence des contraintes la convergence est très difficile. Ainsi, pour évaluer les performances de la commande prédictive basée sur la méthode d'essaim de particules en présence de la contrainte, on considère la contrainte suivante :

$$|u(k)| \leq 1,5 \quad (4.23)$$

La population initiale est de taille $n_p = 5$, et les composantes des particules initiales doivent évidemment vérifier la contrainte (4.23). Initialement, $k = 0$, nous avons choisi

$$U^1(0) = \begin{bmatrix} 0,2 \\ -1,5 \\ 0,0 \\ 0,0 \end{bmatrix}, \quad U^2(0) = \begin{bmatrix} 0,5 \\ -1,0 \\ 0,1 \\ 0,1 \end{bmatrix}, \quad U^3(0) = \begin{bmatrix} 1,0 \\ -0,6 \\ 0,3 \\ 0,3 \end{bmatrix}, \quad U^4(0) = \begin{bmatrix} 1,25 \\ -0,30 \\ 0,60 \\ 0,60 \end{bmatrix}, \quad U^5(0) = \begin{bmatrix} 1,50 \\ 0,00 \\ 1,50 \\ 1,40 \end{bmatrix}$$

avec des vitesses nulles. Pour l'arrêt de l'algorithme, on fixe le nombre d'itérations maximal à 25. De la même manière, à chaque instant d'échantillonnage k , les particules initiales (positions) sont prises égales aux meilleures solutions précédentes, c'est-à-dire

$$U^j(0) = U^j(25), \quad j = 1, \dots, 5.$$

avec des vitesses initiales nulles.

Pour respecter la contrainte imposée sur la commande, si l'une des composantes de chaque particule ne respecte pas la contrainte (4.23), alors on propose de garder sa valeur précédente.

Les résultats obtenus sont donnés par la Figure 4.5. Concernant l'évolution de la sortie, on peut noter le même constat, comme dans le cas sans contrainte. La sortie atteint sa consigne imposée et la commande reste dans son domaine admissible défini par la contrainte (4.23).

Le dernier point concerne la comparaison des deux méthodes sur le plan de temps de calcul. Dans le cas de la commande prédictive, à chaque période d'échantillonnage on doit résoudre le problème d'optimisation pour déterminer la commande à appliquer au système, alors pour un but de comparaison, on considère le temps de calcul moyen.

	Méthode d'Alienor	Méthode d'essaim de particules
Sans contrainte	6.164 s	13.457 s
Avec contrainte	6.925 s	12.775 s

Tableau 4.1 Temps moyen pour le calcul de la commande.

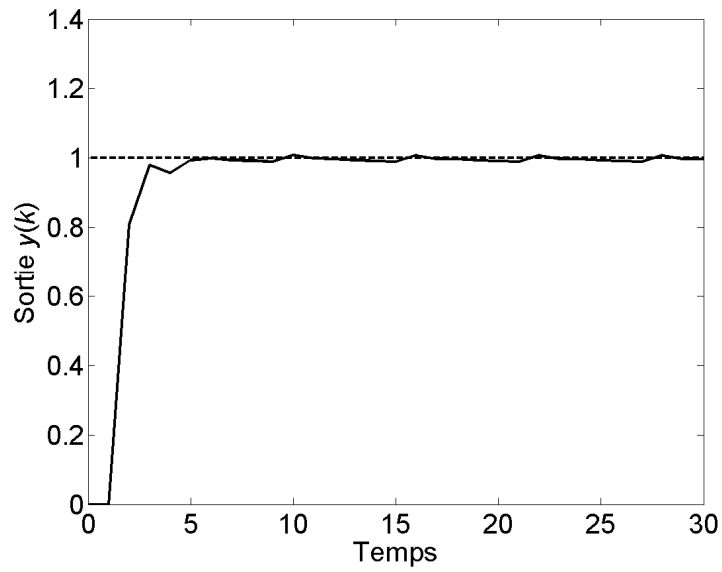
4.4 Conclusion

Dans ce chapitre, nous avons adopté les méthodes d'optimisation globale d'Alienor et d'essaim de particules dans une commande prédictive d'un système non linéaire, pour la résolution du problème d'optimisation. L'objectif traduit par le critère, à minimiser, consiste à assurer la poursuite d'une consigne constante. Les deux cas avec et sans contrainte boîte sur la variable de commande ont été étudiés.

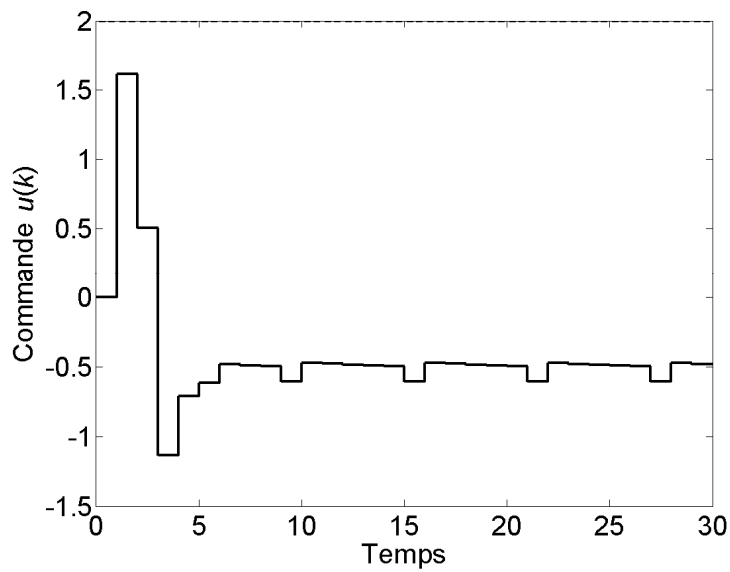
Sans contrainte, les deux méthodes permettent d'assurer la poursuite de consignes avec des régimes transitoires acceptables. Néanmoins la méthode d'Alienor donne un meilleur résultat par rapport à la méthode d'essaim de particules. Ceci est expliqué par le fait que la convergence est exacte dans le cas de la méthode d'Alienor et asymptotique de la méthode d'essaim de particules. Pour les deux méthodes d'optimisation, les commandes générées présentent des fluctuations physiquement acceptables. Dans les deux cas avec et sans contrainte sur la variable de commande, le résultat est nettement meilleur dans le cas de la méthode d'Alienor.

Cet exemple a permis de mettre en évidence, une des propriétés importantes des méthodes exactes d'optimisation globale à savoir la convergence exacte. Pour les méthodes heuristiques quoiqu'elles permettent d'atteindre asymptotiquement l'optimum global mais leur réglage (choix du nombre de particules et de la population initiale) reste une tâche stochastique. Par conséquent, leur application à des problèmes d'optimisation en temps réel nécessite inéluctablement de les aider par une méthode d'optimisation exacte, même locale, c'est-à-dire utiliser des méthodes hybrides, par exemple la méthode d'essaim de particules avec la méthode de Newton. De plus en présence de la contrainte, le réglage devient très difficile et la convergence est assurée mais au prix d'un effort de calcul important.

En résumé, l'exemple étudié montre clairement que la commande prédictive non linéaire basée sur l'optimisation globale donne de bons résultats si la convergence exacte est assurée.

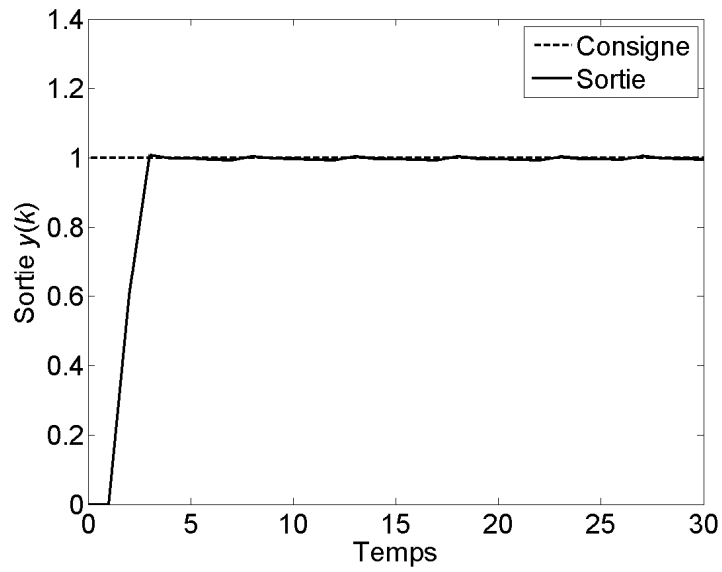


(a) Evolution de la sortie commandée $y(k)$.

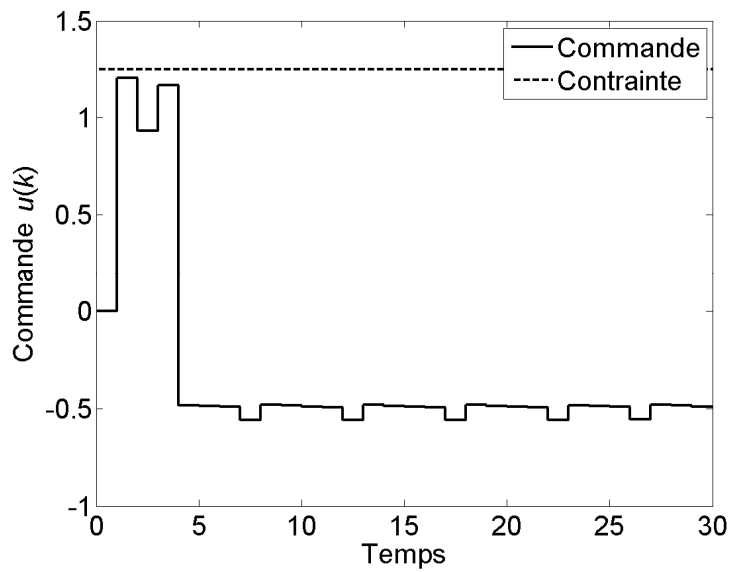


(b) Evolution de la commande $u(k)$.

Figure 4.2 Méthode d'Alienor : cas sans contrainte sur la variable de commande $u(k)$.

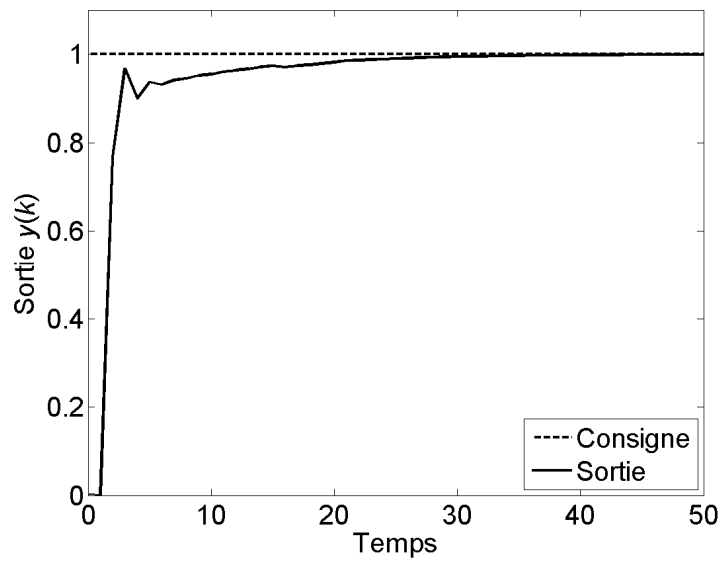


(a) Evolution de la sortie commandée $y(k)$.

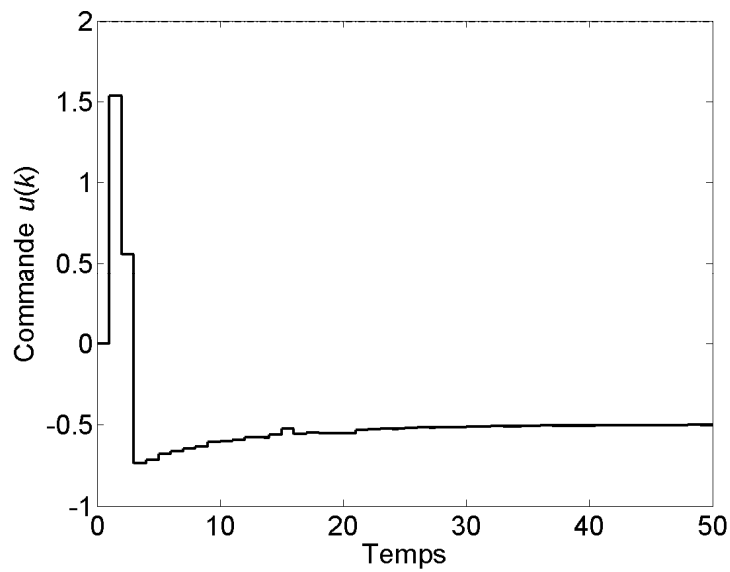


(b) Evolution de la commande $u(k)$.

Figure 4.3 Méthode d'Alienor : cas avec contrainte sur la variable de commande $u(k)$ ($|u(k)| \leq 1,25$).

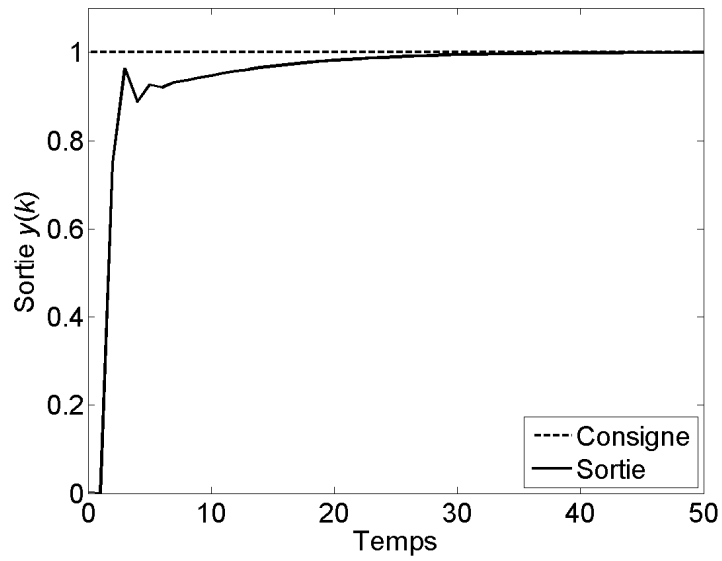


(a) Evolution de la sortie commandée $y(k)$.

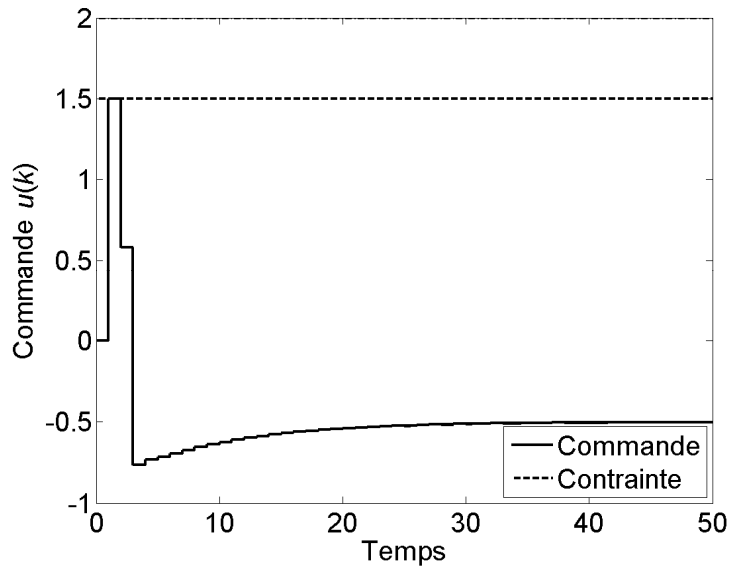


(b) Evolution de la commande $u(k)$.

Figure 4.4 Méthode d'essaim de particules : cas sans contrainte sur la variable de commande $u(k)$.



(a) Evolution de la sortie commandée $y(k)$.



(b) Evolution de la commande $u(k)$.

Figure 4.5 Méthode d'essaim de particules : cas avec contrainte sur la variable de commande $u(k)$.

Conclusion Générale

Le travail réalisé dans ce mémoire s'inscrit dans le cadre de la commande à base d'un modèle. Il porte essentiellement sur la commande prédictive des systèmes dynamiques, en particulier les systèmes monovariables non linéaires. L'objectif du travail consiste à utiliser des techniques d'optimisation globale dans la commande prédictive d'un système non linéaire monovariable, de montrer leurs apports et de comparer leurs performances. Ainsi, les deux méthodes étudiées sont : la méthode exacte d'Alienor, et la méthode heuristique d'essaim de particules.

Après avoir introduit le principe et ses différents éléments d'une commande prédictive, nous avons présenté en détail un des célèbres algorithmes de la commande prédictive qui est le *Dynamic Matrix Control* dont les performances ont été évaluées en considérant un système de second ordre avec un retard important, tout en étudiant l'influence des paramètres de réglages de la commande prédictive. Comme l'étape de l'optimisation joue un rôle du premier plan dans une commande prédictive, nous avons passé en revue les différentes notions d'optimisation utiles le long de notre travail, tout en insistant sur l'optimisation globale des fonctions mathématiques et son importance dans une commande prédictive. Par la suite, deux méthodes d'optimisation globale ont été présentées de manière détaillée, en l'occurrence la méthode exacte d'Alienor et la méthode heuristique d'essaim de particules. Pour mettre en évidence les propriétés des deux méthodes, une application à l'identification des paramètres d'un système non linéaire a été présentée. Ensuite, nous avons adopté les deux méthodes pour la commande prédictive d'un système non linéaire monovariable dont l'objectif est d'assurer une poursuite de consigne tout en respectant des contraintes sur la variable de commande.

Les différentes applications réalisées ont démontré l'efficacité de la commande prédictive et son réglage simple. Concernant la commande prédictive des systèmes non linéaires, nous avons montré que l'utilisation des méthodes d'optimisation globale pour la résolution du problème d'optimisation permet d'améliorer davantage les performances.

Ainsi, les résultats d'identification et de commande prédictive obtenus démontrent que la méthode exacte d'Alienor est très intéressante par rapport à la méthode heuristique d'essaim de particules. Quoique la convergence est assurée pour les deux méthodes, mais elle est globale dans le cas de la méthode d'Alienor alors qu'elle est asymptotique dans le cas de la méthode d'essaim de particules. Atteindre l'optimum global dans le cas de l'identification et de la commande prédictive garantie une très bonne estimation des paramètres et une poursuite parfaite de la consigne.

La méthode d'Alienor bénéficie d'un ensemble de transformations réductrices permettant d'avoir une convergence avec une très grande précision et le réglage de ces différents paramètres est dicté par des théorèmes. Aussi, les contraintes du type boîtes sur les variables de décision peuvent être prises en compte directement dans l'algorithme d'optimisation sans faire appel au Lagrangien ou à la fonction de pénalisation, ce qui réduit énormément les calculs, par conséquent le temps de convergence. Par contre, dans le cas de la méthode d'essaim de particules, outre sa convergence asymptotique, le réglage est souvent fait de manière aléatoire, et le choix de la taille de la population initiale, de ces particules initiales et leurs vitesses constitue une tâche très difficile surtout en présence de la contrainte.

En résumé, les résultats obtenus montre clairement que la méthode d'optimisation globale d'Alienor est bien adaptée à l'implémentation de commande prédictive pour un système non linéaire. Néanmoins, certains points peuvent être explorés dans le cadre de la commande prédictive non linéaire tel que :

- La comparaison entre plusieurs transformations réductrices dans le cas d'une commande prédictive,
- L'application des méthodes hybrides, par exemple hybrider la méthode d'essaim de particules avec la méthode de Quasi-Newton,
- L'application de la commande prédictive à base de la méthode d'Alienor basée sur un modèle d'état et d'étendre l'étude à des systèmes multivariables.
- Prise en compte du rejet de perturbations.

Références

- Banks, A. , Vincent, J. and Anyakoha, C. (2007). A Review of Particle Swarm Optimization. Part I: Background and Development. *Nat Comput*, Vol. 6, pp. 467–484.
- Banks, A. , Vincent, J. and Anyakoha, C. (2008). A Review of Particle Swarm Optimization. Part II: Hybridisation, Combinatorial, Multicriteria and Constrained Optimization, and Indicative Application. *Nat Comput*, Vol. 7, pp. 109–124.
- Bendiab, O. and Cherruault, Y. (1995). A new method for global optimization in two dimensions. *International Journal of Bio-Medical Computing*, vol. 38, pp. 71–73.
- Berthiau, G. et Siarry, P. (2001). Etat de l'Art des Méthodes d'Optimisation Globale. *RAIRO Operations Research*, Vol. 35, pages 329–365.
- Bemporad, A., Morari, M., Dua, V. et Pistikopoulos, E. N. (2002). The explicit linear quadratic regulator for constrained systems. *Automatica*, Vol. 38, pp. 3–20.
- Camacho, E. F. et Bordons, C. (1998). *Model Predictive Control in the Process Industry*. Springer, London.
- Cannon, M. (2004). Efficient Nonlinear Model Predictive Control Algorithms. *Annual Reviews in Control*, vol. 28, pp. 229–237.
- Cherruault, Y. (1991). New Deterministic Methods for Global Optimization and Applications to Biomedicine. *Int. J. of Bio-Medical Computing*, Vol. 27, pp. 215–229.
- Cherruault, Y. and Mora, G. (2005). *Optimisation Globale : Théorie des Courbes- α -Denses*. Economica.
- Clarke, D. W., Mohtadi, C. et Tuffs, P. S. (1987). Generalized Predictive Control. Part I. The Basic Algorithm, *Automatica*, Vol. 23, pp. 137–148.

- Clarke, D. W. et Mohtadi, C. (1989). Properties of Generalized Predictive Control. *Automatica*, Vol. 25, pp. 859–875.
- Clerc, M. and Kennedy, J. (2002). The Particle Swarm: Explosion, Stability, and Convergence in a multidimensional complex space. *IEEE Trans. Evolutionary Computation*, Vol. 6, No. 1, pp. 58–73.
- Cutler, C.R. and Ramaker, B.L. (1979). Dynamic Matrix Control. A Computer Control Algorithm. *AIChE 86th National Meeting*, USA.
- De Keyser, R. M. C. et Van Cauwenberghe, A. (1979). A self-tuning multistep predictor application. *Automatica*, Vol. 15, pp. 167–174.
- De Keyser, R. M. C. (1988). A Gentle Introduction to Model Based Predictive Control. *Proceedings of the International Conference on Control Engineering and Signal Processing*, Peru.
- Dochain, D. (2003). State and Parameter Estimation in Chemical and Biochemical Processes : A Tutorial. *Journal of Process Control*, Vol. 13, pp. 801–818.
- Eberhart, R. C., Shi, Y., and Kennedy, J. (2001). *Swarm Intelligence*. The Morgan Kaufmann Series in Artificial Intelligence. Morgan Kaufmann, San Francisco, CA, USA.
- Faber, R., Li, P. and Wozny, G. (2007). An Optimization Framework for Parameter Estimation of Large-scale Systems. *Chemical Engineering and Processing*, Vol. 46, pp. 1085–1095.
- Fakhfakh, M. (2009). A Novel Alienor-based Heuristic for the Optimal Design of Analog Circuits. *Microelectronics Journal*, Vol. 40, pp. 141–148.
- Findeisen, R., Allgöwer, F. and Biegler, L. T. (2007). Assessment and Future Directions of Nonlinear Model Predictive Control. *Lecture Notes in Control and Information Sciences*, Vol. 358, Springer.
- Flaus, J.-M. (1994). *La Régulation Industrielle : Régulateurs PID, Prédicatifs et Flous*. Hermès.
- Gevers, M. (1996). Identification for Control. *Annual Reviews in Control*, Vol. 20, pp. 95–106.
- Hammouche, K., Diaf, M. and Siarry, P. (2010). A Comparative Study of Various Metaheuristic Techniques Applied to the Multilevel Thresholding Problem. *Engineering Applications of Artificial Intelligence*, Vol. 23, pp. 676–688.
- Henson, M. A. (1998). Nonlinear Model Predictive Control : Current Status and Future Directions. *Computers and Chemical Engineering*, Vol. 23, pp. 187–202.
- Heppner, F. and Grenander, U. (1990). A Stochastic Nonlinear Model for Coordinated Bird Flocks. *AAAS Publication*, Washington, DC.

- Hiriart-Urruty, J.-B. (1996). *L'Optimisation*. Presses Universitaires de France, 1^{ère} Edition.
- Horst, R. et Pardalos, P. M. (1995). *Handbook of Global Optimization*. Kluwer Academic Publishers.
- Huang, B., and Kadali, R. (2008). Dynamic Modeling, Predictive Control and Performance Monitoring : a Data-Driven Subspace Approach. *Lecture Notes in Control and Information Sciences*, Vol. 374, Springer.
- Ikonen, E., Najim, K. (2002). *Advanced Process Identification and Control*. Marcel Dekker, New York, USA.
- Kouvaritakis, B., and Cannon, M. (2001). *Nonlinear Predictive Control : Theory and Practice*. *IEE Control Engineering Series*, Vol. 61, IET.
- Lee, J. K. and Park, S. W. (1991). Model Predictive Control fo Multivariable Unstable Processes with Constraints on Manipulated Variables. *Korean J. of Chem. Eng.*, Vol. 8, No. 4, pp. 195-202.
- Ljung, L. (1990). *System Identification : Theory for the User*. Prentice Hall.
- Ljung, L. (2004). State of the art in linear system identification : Time and Frequency Domain methods. *American Control Conference (ACC)*, June 30-July 2, Boston Massachusetts, USA, pp. 650–660.
- Ljung, L. (2008). Perspectives on System Identification. *17th IFAC World Congress*, July 6-11, Seoul, Korea, pp. 47–59.
- Long, C. E., Polisetty, P. K. and Gatzke, E. P. (2006). Nonlinear Model Predictive Control Using Deterministic Global Optimization. *Journal of Process Control*, Vol. 16, pp. 635–643.
- Magni, L., Raimondo, D. M., and Allgöwer, F. (2009). *Nonlinear Model Predictive Control : Towards New Challenging Applications*. *Lectures Notes in Control and Information Sciences*, Springer.
- Martinsen, F., Biegler, L. T., Foss, B. A. (2002). Application of Optimization Algorithms to Nonlinear MPC. *Proceedings of the 15th Triennial World Congress*, Barcelona, Spain.
- Mayne, D. Q., Rawlings, J. B., Rao, C. V. et Scokaert, P. O. M. (2000). Constrained Model Predictive Control: Stability and Optimality. *Automatica*, Vol. 36, No. 6, pp. 789-814.
- Morari, M. (1994). *Model predictive control: Multivariable control technique of choice in the 1990s?*. Livre : *Advances in Model-Based Predictive Control*. Oxford University Press.
- Nelles, O. (2001). *Nonlinear System Identification : From Classical Approaches to Neural Networks and Fuzzy Models*. Springer.

- Nyarko, E. K., Scitovski, R. (2004). Solving the Parameter Identification Problem of Mathematical Models Using Genetic Algorithms. *Applied Mathematics and Computation*, Vol. 153, pp. 651–658.
- Ogunfunmi, T. (2007). *Adaptive Nonlinear System Identification : the Volterra and Wiener Model Approaches*. Springer.
- Patelli, A., Ferariu, L. (2009). Nonlinear System Identification by Means of Genetic Programming. *Proceedings of the European Control Conference*, Budapest, Hungary, August 23–26, pp. 502–507.
- Poli, R., Kennedy, J., and Blackwell, T. (2007). Particle swarm optimization. *Swarm Intelligence*, Vol. 1, No. 1, pp. 33–57.
- Ramírez, D. R. et Camacho, E. F. (2001). On the Piecewise Linear Nature of Min-Max Model Predictive Control with Bounded Uncertainties. *Proceedings of the 40th Conference on Decision and Control*, Florida USA.
- Raol, J., Girija, G. and Singh, J. (2004). *Modelling and Parameter Estimation of Dynamic Systems*. IEEE Control Engineering Series, Vol. 65.
- Reynolds, W. C. (1987). Flocks, Herds, and Schools : A distributed Behavioral Model. *Computer Graphics*, Vol. 21, No. 4, pp. 25–34.
- Richalet, J., Rault, A., Testud, J. L. et Papon, J. (1976). Algorithmic Control of Industrial Processes. *Proceedings of the 4th IFAC Symposium on Identification and System Parameter Estimation*, Tbilisi, URSS.
- Richalet, J. Lavielle G., et Mallet, J. (2005). *La Commande Prédictive*. Mise en Œuvre et Applications Industrielles. Editions Eyrolles.
- Sakthivel, V. P., Bhuvanewari, R., Subramanian, S. (2010). Multi-objective Parameter Estimation of Induction Motor Using Particle Swarm Optimization. *Engineering Applications of Artificial Intelligence*, Vol. 23, pp. 302–312.
- Schwaab, M., Biscaia, E. C., Monteiro, J. L. and Pinto, J. C. (2008). Nonlinear Parameter Estimation Through Particle Swarm Optimization. *Chemical Engineering Science*, Vol. 63, pp. 1542–1552.
- Söderström, T. and Stoica, P. (1989). *System Identification*. Prentice Hall.
- Soroush, M. (1998). State and Parameter Estimation and their Applications in Process Control. *Computers and Chemical Engineering*, vol. 23, pp. 229–245.
- Tatjewski, P. (2007). *Advanced Control of Industrial Processes : Structures and Algorithms*. Springer.

- Ydstie, B. E. (1984). Extended Horizon Adaptive Control. *Proceedings of the 9th IFAC World Congress*, Budapest, Hungary.
- Unbehauen, H. and Rae, G. P. (1998). A Review of Identification in Continuous Time Systems. *Annual Reviews in Control*, vol. 22, pp. 145–171.
- Ursema, R. K., Vadstrup, P. (2004). Parameter Identification of Induction Motors Using Stochastic Optimization Algorithms. *Applied Soft Computing*, Vol. 4, pp. 49–64.
- Van Den Bergh, F. (2002). *An Analysis of Particle Swarm Optimizers*. PhD thesis, Department of Computer Science, University of Pretoria.
- Ziadi, A., Cherruault, Y. and Mora, G. (2001). Global Optimization : A New Variant of the Alienor Method. *J. Comput. Math. Appl.*, Vol. 41, pp. 63–71.

Résumé

Dans ce mémoire, on s'intéresse à étudier l'apport des méthodes d'optimisation globale dans une stratégie de commande prédictive d'un système non linéaire. L'objectif du travail ;consiste à appliquer les méthodes d'optimisation globale d'Alienor et d'essaim de particules pour la résolution du problème d'optimisation non linéaire d'une commande prédictive non linéaire, puis de comparer leurs performances. L'étude se limite au système non linéaire caractérisé par une dynamique lente, et à la poursuite de consigne en présence des contraintes, de type boîtes, sur la variable de commande. Les deux méthodes sont implémentées en simulation pour la commande prédictive d'un système non linéaire dont la prédiction est réalisée par un modèle du type entrée-sortie. Les deux cas avec et sans contrainte sur la variable de commande sont étudiés.

Mot clés :

Commande prédictive, système non linéaire, optimisation globale, méthode Alienor, méthode d'essaim de particules.