

THESE DE DOCTORAT

Présentée à:

**L'UNIVERSITE MENTOURI DE CONSTANTINE
FACULTE DES SCIENCES
DEPARTEMENT D'ELECTRONIQUE**

Par:

MEGRI Abederrahim Fayçal

Pour l'obtention du diplôme de doctorat es science

Thème:

Programmation dynamique floue appliquée
à la commande des systèmes non linéaires

Soutenue publiquement le 25/10/2008 devant le jury composé de:

Mr S. FILALI	Professeur, Univ. de Constantine	Président
Mr K.BELARBI	Professeur, Univ. de Constantine	Directeur de thèse
Mr D.CHIKOUCHE	Professeur, Univ. de Msila	Examineur
Mr A. BENNIA	Professeur, Univ. de Constantine	Examineur
Mr N.GUERFI	Maître de Conférences, Univ. de Tébessa	Examineur

Publications personnelles

Ce travail de thèse a donné lieu à une publication dans une revue internationale avec comité de lecture et une publication dans une conférence internationale avec comité de lecture .

K.Belarbi and F.Megri, « A Stable-based fuzzy predictive control based on fuzzy dynamic programming», IEEE Transactions on fuzzy systems, Vol.15,pp.764-754, N°4, 2007

F.Megri, M.Chemachema, M.Bahita, K.Belarbi, «Application de la Programmation dynamique floue à la commande predictive non linéaire», Conférence maghrébine sur la génie électrique, **CMGE 2001, Algerie, 5-6 Novembre 200**

Remerciements

Ce travail a été réalisé au laboratoire d'automatique et de robotique du département d'électronique de l'université de Constantine.

J'exprime toute ma reconnaissance à Monsieur BELARBI Khaled, Professeur à l'université de Constantine, pour avoir assuré l'encadrement de ce travail. Je le remercie pour son soutien, son orientation et ses consignes. Son expérience et sa connaissance ont contribué à ma formation scientifique.

Je remercie vivement Monsieur BOUKEZZOULA Réda, maître de conférences au laboratoire LISTIC à l'université de Savoie, pour sa disponibilité et son orientation dans la rédaction de cette thèse.

Je tiens à exprimer mes vifs remerciements à Monsieur FILALI Salim, Professeur à l'université de Constantine, pour avoir présidé le jury de ma soutenance de thèse.

J'exprime ma reconnaissance à Monsieur CHIKOUCHE Djamel, Professeur à l'université de Msila, Monsieur BENNIA Abdelhak, Professeur à l'université de Constantine, ainsi qu'à Monsieur GUERFI Noureddine, Maître de Conférences à l'université de Tebéssa, pour l'intérêt qu'ils ont porté à ce travail et qui ont accepté d'en être les rapporteurs.

Et enfin, cet avant-propos serait incomplet sans des remerciements chaleureux pour ma famille, pour leur soutien, leur patience et leur grande disponibilité pour m'avoir offert leur aide, contribuant ainsi à mon enrichissement personnel.

Megri Fayçal

TABLE DES MATIERES

Introduction.....	4
Chapitre I. Introduction à la commande Prédictive.....	7
I.1. Introduction.....	8
I.2. Principe de la commande prédictive à base de modèles	8
I.2.1. Modèle de prédiction.....	10
I.2.2. Méthodologie d'optimisation.....	11
I.2.3. Méthodologie d'application de la commande.....	13
I.3. Les Contraintes dans la commande prédictive.....	14
I.3.1. Contraintes sur le signal de commande.....	14
I.3.2. Contraintes sur la sortie du système.....	15
I.3.3. Contrainte sur l'état terminal.....	15
I.3.4. Autres types de contraintes.....	15
I.4. La commande prédictive linéaire.....	15
I.4.1. Introduction.....	15
I.4.2. Synthèse et étude de la stabilité de la C.P.L.....	17
I.5. La Commande Prédictive Non-linéaire (C.P.N).....	21
I.5.1. Introduction.....	21
I.5.2. Synthèse et étude de la stabilité de la C.P.N.....	22
I.5.3. Commande prédictive sous optimale.....	28
I.6. Conclusion.....	30
Chapitre II. Commande Prédictive Non Linéaire Floue	31
II.1. Introduction.....	32
II.2. C.P.N.L avec Optimisation Floue (C.P.N.O.F).....	33
II.2.1. Principe de la méthode.....	33
II.2.2. Eléments de base de la prise de décision floue.....	33
II.2.3. Méthodologie d'implantation	38
II.2.4. Généralisation au cas multi-contraintes.....	39
II.2.5. Résolution par la Programmation Dynamique.....	40
II.2.6. Solution par la méthode de Branch & Bound	43
II.2.7. Performances de la C.P.N.O.F.....	48
II.3. Commande Prédictive Non linéaire Floue (C.P.N.F)	49
II.3.1. Principe et implantation de la méthode.....	49
II.4. Conclusion.....	52

Chapitre III. Sur la Stabilité de la Commande Prédictive Non Linéaire Floue.....	53
III.1. Introduction.....	54
III.2. Commande Prédictive Floue Duale (Double).....	54
III.2.1. Formulation générale	54
III.2.2. Hypothèse de l'approche.....	56
III.3. Analyse de stabilité	57
III.3.1. Problème	57
III.3.2. Théorème.....	58
III.3.3. Démonstration.....	59
III.4. Méthodologie d'optimisation.....	61
III.4.1. Algorithme de la méthode.....	64
III.4.2. Calcul de la région terminale.....	65
III.5. Conclusion.....	67
Chapitre IV. Exemples d'applications.....	68
IV.1. Introduction:.....	69
IV.2. Exemples d'applications.....	69
IV.2.1. Exemple1: Système Non linéaire 1er ordre.....	69
IV.2.2. Exemple2: Système Non linéaire 2er ordre.....	72
IV.2.3. Exemple 3: Réacteur Chimique CSTR.....	75
IV.3. Conclusion.....	82
Conclusion Générale.....	83
Bibliographie.....	85
Annexe.....	89

CHAPITRE I:

INTRODUCTION A LA COMMANDE
PREDICTIVE

I.1. Introduction

La commande prédictive repose sur l'utilisation d'un modèle pour prédire le comportement du système à commander et le choix d'une décision la meilleure au sens d'un certain coût tout en respectant un certain nombre de contraintes.

Le concept de la commande prédictive a été introduit au milieu des années soixante dix par plusieurs auteurs notamment pour la première fois par Richalet et al. [38] et Cutler et Ramaker [12]. Dans ces méthodes, la commande est obtenue en minimisant un critère quadratique. Ces techniques de commande ont été appliquées rapidement et avec succès dans le monde industriel en particulier sur les processus industriels chimiques. Dans la continuité des ces travaux, d'autres techniques ont été développées. Nous citons ici la plus populaire, à savoir la commande GPC «Generalized Predictive Control», proposée dans [8][9][10]. Cette méthode s'inspire des principes de la commande GMV «Generalized Minimum Variance»[8] avec l'utilisation d'une fonction de transfert paramétrique comme modèle de prédiction. La formalisation de la commande prédictive ne se limite pas uniquement à des systèmes linéaires mais elle a aussi été donnée dans le contexte non linéaire. De plus, le formalisme d'état permet l'extension de la commande prédictive à des cas plus complexes, tels que les systèmes avec perturbations stochastiques et les systèmes multivariables.

Durant les années 1970 et le début des années 1980, la recherche de résultats de stabilité dans des problèmes de commande prédictive n'avait été considérée profondément. A partir de cette date, apparaissent des travaux exploitant des horizons finis et infinis pour lesquels il est possible de démontrer, sous certaines conditions, la stabilité du système contrôlé par une stratégie MPC. A titre d'exemple, Chen et Shaw [4] ont utilisé la fonction de coût comme une fonction de Lyapunov pour établir la stabilité des systèmes où une contrainte d'égalité terminale est exigée. Ce résultat a été étendu dans le domaine continu par Mayne et Michalska [27]. Dans un contexte non linéaire, Keerthi et Gilbert [18] sont les premiers qui ont utilisé ce principe pour établir la stabilité de la commande prédictive à base de modèles sur les systèmes non linéaire en temps discret avec contraintes. Dans ce cas, une contrainte d'égalité terminale est employée.

Dans ce chapitre nous abordons les outils de représentation, de synthèse et d'analyse nécessaires pour le développement des structures de commande prédictive linéaires étudiées dans cette thèse. Dans la deuxième section, nous donnons le principe de la commande prédictive à base de modèles. Dans la troisième section, les différentes contraintes que peut subir un système commandé sont énoncées. Dans la quatrième section, la synthèse et l'étude de la stabilité de la commande prédictive linéaire est détaillée. L'extension de cette dernière au cas non linéaire est expliquée dans la cinquième section.

I.2. Principe de la commande prédictive à base de modèle

Pour des raisons de simplicité et de clarté, dans un premier temps le principe de la commande prédictive est introduit en absence des contraintes. Par la suite, la problématique de la présence des contraintes et leurs influences sur les performances de la commande prédictive sera discutée.

Le principe de la commande prédictive est illustré sur la Figure I.1 où l'objectif est de minimiser

la distance (l'erreur) entre la sortie du système y et la trajectoire de référence w .

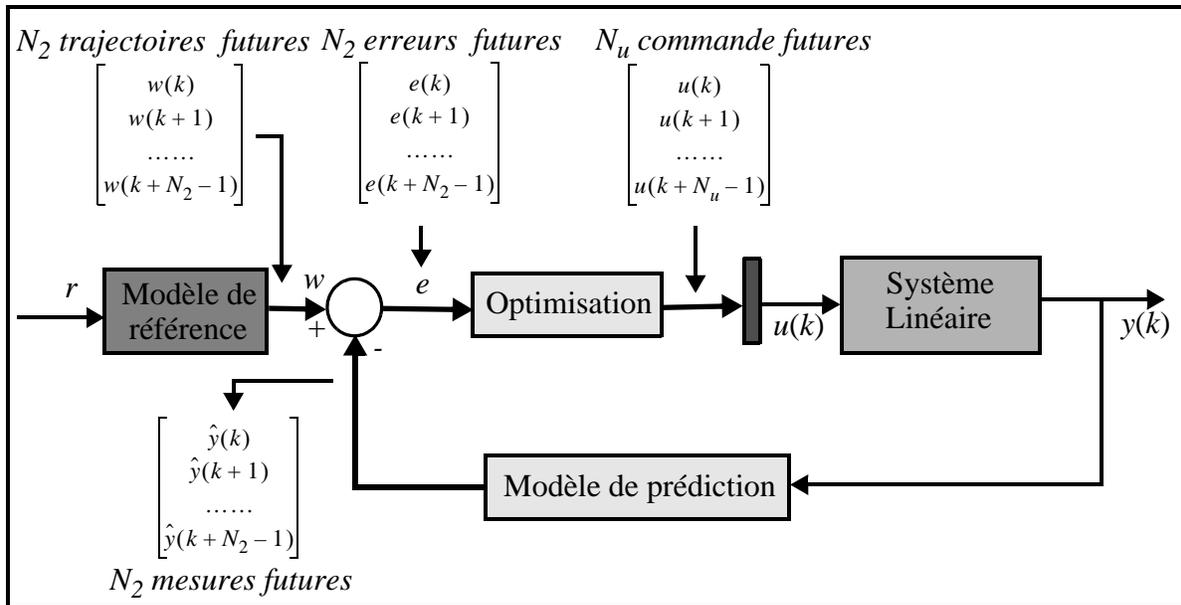


Figure I.1: Schéma classique de la commande prédictive

Etant donné un système (linéaire ou non linéaire) avec une entrée $u(k)$ et une sortie $y(k)$, la philosophie de la commande prédictive repose sur les trois étapes décrites ci-dessous (cf. Figure I.2).

1. L'utilisation d'un modèle du système pour prédire les sorties futures de ce dernier, notées $y(k)$, $y(k+1)$, ..., $y(k+N_2-1)$, où N_2 représente l'horizon de prédiction.
2. Le calcul d'une séquence des futurs échantillons de commande $u(k)$, $u(k+1)$, ..., $u(k+N_u-1)$, où N_u représente l'horizon de commande, en minimisant une certaine fonction d'objectif (fonction du coût) sur cette horizon.
3. L'application de la première commande $u(k)$ issue de la séquence d'optimisation au système.

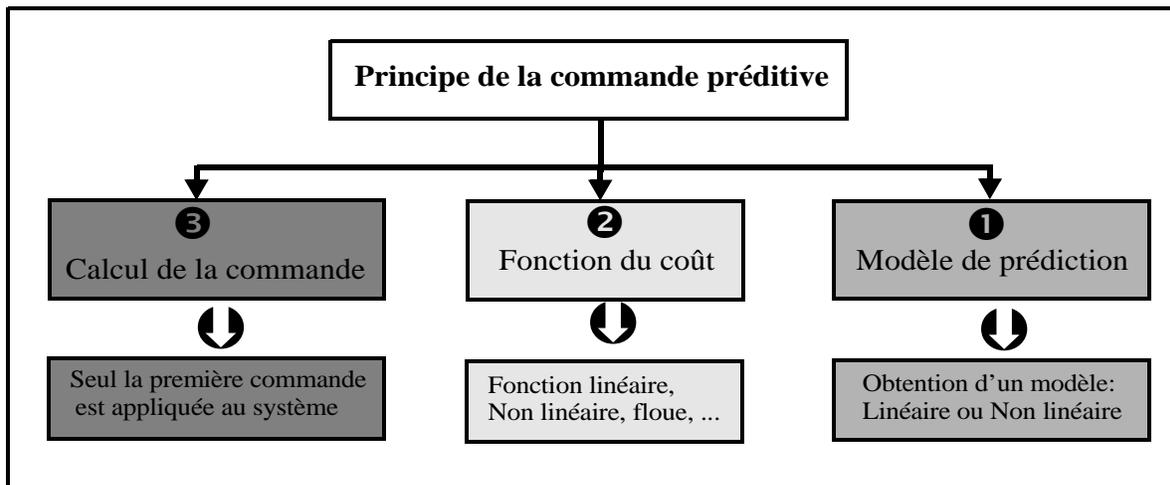


Figure I.2 Principe de la commande prédictive.

I.2.1. Modèle de prédiction

Le modèle du système linéaire ou non linéaire a pour objectif la prédiction de la sortie de ce dernier à des instants futurs (échantillonnés). Cette phase d'obtention du modèle est une étape initiale et primordiale dans la conception d'une stratégie de commande prédictive. En effet, le modèle doit reproduire avec une exactitude suffisante les caractéristiques dynamiques du système sur un horizon de temps fini. En pratique, plusieurs formes de modèles de prédiction ont été proposés. Nous citons ici:

I.2.1.1. Les modèles linéaires

Dans toute la suite, la notation $z(k+j/k)$ fait référence à la valeur de z à l'instant $k+j$ calculée à l'instant k . Supposons qu'un système dynamique linéaire avec une entrée $u(k)$ et une sortie $y(k)$. Le modèle de prédiction de ce système qui a pour sortie $\hat{y}(k)$ peut être représenté par:

- Un modèle à base de réponse impulsionnelle donné par l'expression suivante:

$$\hat{y}(k+j/k) = \sum_{i=1}^{N_2} h_i \cdot u(k+j-i/k) \quad (\text{I.1})$$

où h_i représentent les valeurs de la sortie suite à une excitation du système par une impulsion de Dirac unitaire.

- Un modèle à base de réponse indicielle donné par l'équation suivante sur un horizon de prédiction fini:

$$\hat{y}(k+j/k) = \sum_{i=1}^{N_2} g_i \cdot \Delta u(k+j-i/k) \quad (\text{I.2})$$

où g_i sont les valeurs de la sortie suite à une excitation du système par un échelon unitaire.

- Un modèle à base d'une représentation d'état. Supposons qu'on dispose d'un modèle de référence sous la forme suivante:

$$\begin{cases} x(k+1) = A \cdot x(k) + B \cdot u(k) \\ y(k) = C \cdot x(k) \end{cases} \quad (\text{I.3})$$

où:

u , y et x désignent respectivement l'entrée, la sortie et le vecteur d'état du modèle de référence. A , B et C sont respectivement une matrice ($n \times n$), un vecteur colonne ($n \times 1$) et un vecteur ligne ($1 \times n$) (représentation d'état classique).

Dans ce cas, le modèle de prédiction qui sera utilisé est sous la forme suivante:

$$\hat{y}(k+j/k) = C[A^j x(k) + \sum_{i=1}^j A^{i-1} B u(k+j-i/k)] \quad (\text{I.4})$$

où:

$\hat{y}(k+j/k)$ est la prédiction de la sortie en $k+j$ étant donnée sa connaissance en k .

- Un modèle à base de fonction de transfert. Dans ce cadre, on suppose que le système peut être représenté par un modèle CARMA (Controlled AutoRegressive Moving Average), dont l'expression est donnée par:

$$\hat{y}(k+j/k) = B(z^{-1})/A(z^{-1}) \cdot u(k+j/k) \quad (\text{I.5})$$

avec:

$$A(z^{-1}) = 1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2} + \dots + a_{n_a} \cdot z^{-n_a}$$

$$B(z^{-1}) = 1 + b_1 \cdot z^{-1} + b_2 \cdot z^{-2} + \dots + b_{n_b} \cdot z^{-n_b}$$

où:

z^{-1} : représente l'opérateur de retard,

B, A : sont respectivement des polynômes associés à l'entrée et à la sortie.

I.2.1.2. Les modèles non linéaires

Les modèles non linéaires utilisés sont souvent représentés par des équations aux différences non linéaires, des systèmes flous et/ou des réseaux de neurones. Dans ce contexte, la forme la plus utilisée pour le modèle de prédiction repose sur une représentation d'état sous la forme suivante:

$$\begin{cases} x(k+1) = f(x(k), u(k)) \\ \hat{y}(k) = h(x(k)) \end{cases} \quad (\text{I.6})$$

où:

- f est un vecteur $f = (f_1, f_2, \dots, f_n)$ de fonctions non linéaires de forme analytique, et h est une fonction analytique non-linéaire.
- $x(k)$ est le vecteur d'état du système, $x(k) \in X$ (X représente son domaine de définition).
- $u(k)$ est la variable de commande $u(k) \in U$ (U représente son domaine de définition).

Parfois l'obtention d'un modèle de prédiction sous cette forme demeure un problème difficile ou au moins coûteux. Cette dernière condition peut représenter un obstacle important dans la commande des systèmes non linéaires complexes. Dans ce contexte, en l'absence d'un modèle analytique pertinent, d'autres techniques telles que les systèmes flous et/ou les réseaux de neurones peuvent être utilisés pour représenter la dynamique des systèmes. En effet, les deux catégories des systèmes sont maintenant connues comme des approximateurs universels et peuvent être utilisées pour représenter les systèmes non linéaires et/ou pour mettre en oeuvre des régulateurs non linéaires.

I.2.2. Méthodologie d'optimisation

La méthodologie d'optimisation a pour objectif l'obtention d'une séquence de futures actions de commande en minimisant une certaine fonction d'objectif (fonction du coût).

Pour des systèmes linéaires, la fonction d'objectif est souvent donnée sous une forme d'une association d'une forme quadratique de l'état (ou erreur de suivi) et d'une énergie sur la commande. Autrement dit, l'objectif de l'optimisation est de chercher une séquence de commande capable de stabiliser l'état du système (ou réaliser un suivi de trajectoire avec une erreur minimale) et une énergie minimale sur la commande. Sous une forme mathématique, la fonction à minimiser est souvent donnée par:

$$\sum_{j=1}^{N_2} [\hat{y}(k+j/k) - w(k+j/k)]^2 + \sum_{j=1}^{N_u} \lambda \cdot \Delta u(k+j-1/k)^2 \quad (\text{I.7})$$

λ : est un facteur de pondération et w est la trajectoire de référence.

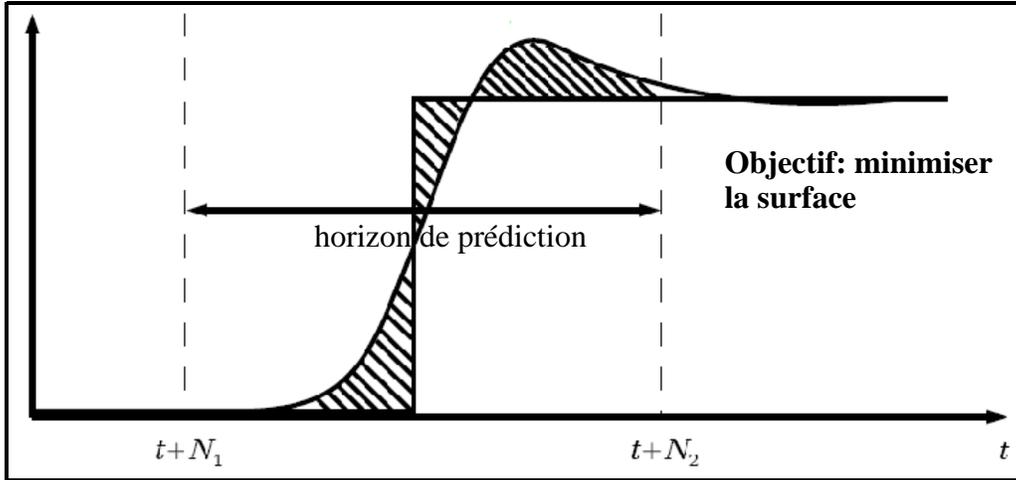


Figure I.3: Principe de l'optimisation.

Dans la littérature, des variantes de cette fonction du coût existent. En effet, d'une part certaines méthodes relaxent la contrainte sur la commande (le terme relatif à l'effort de commande n'est pas pris en compte[43]). D'autre part, des méthodes ajoutent d'autres termes pour prendre en considération le changement de la sortie. Pour des systèmes non linéaires, la forme de la fonction d'objectif est souvent de nature plus complexe. En effet, contrairement au problème d'optimisation linéaire, dans le cas non linéaire on ne dispose pas de techniques universelles capables de présenter des résultats généraux pour le problème d'optimisation. La majorité des méthodes présentées dans la littérature sont restreintes à des formes particulières des fonctions d'objectif. D'une manière générale, un problème d'optimisation $P_N(x)$ dans le cas linéaire ou non linéaire peut s'écrire sous la forme suivante [26]:

$$P_N(x) : \text{Min}_{\Pi} V_N(x, \Pi) \quad (\text{I.8})$$

où:

- $V_N(x, \Pi) = \sum_{j=1}^N L(x(k+j/k), u(k+j/k)) + F(x(k+N/k))$ est la fonction à optimiser.
- $L(x(k+j/k), u(k+j/k))$ est la fonction de coût.
- $F(x(k+N/k))$ est la contrainte sur l'état terminal.
- Π est un vecteur qui indique la séquence future de la commande de N éléments: $\Pi = \{u(k/k), \dots, u(k+N-1/k)\}$.
- $u(k+j/k)$ est la variable de commande à l'instant $k+j$ calculée à l'instant actuel k .
- L'évolution prédite du système à l'instant $k+j+1$ est donnée par l'équation dynamique du modèle de prédiction utilisé.

Pour des raisons de simplicité et de clarté, nous avons supposé que: $N_2 = N_u = N$.

Dans la littérature, différentes méthodologies d'optimisation pour minimiser la fonction d'objectif ont été proposées [3][23] (programmation dynamique, programmation quadratique séquentielle,...).

Pour une implantation efficace de la commande prédictive, une solution analytique au problème d'optimisation doit être disponible. Si ce problème est relativement simple à résoudre pour des systèmes linéaires, il représente un obstacle important dans le cadre des systèmes non linéaires.

En effet, sachant que les séquences de commande optimales et analytiques ne peuvent pas généralement être déterminées, on doit avoir recours à des techniques numériques d'optimisation [33]. Ces techniques présentent nécessairement des difficultés inhérentes aux algorithmes itératifs (valeurs d'initialisation, états de convergence, erreurs d'approximation). De plus, ces techniques travaillent sur un long horizon de commande, très gourmandes en temps de calcul et ne convergeant souvent que vers des optimaux locaux. A titre d'exemple, nous citons les méthodes souvent exploitées dans les travaux de recherche, à savoir la méthode de la programmation quadratique séquentielle (*SQP*) [3] et l'extension de la méthode Simplex proposée par Nelder et Mead dans [14]. Afin de contourner ces difficultés, d'autres méthodes d'optimisation peuvent être exploitées sous l'hypothèse d'une discrétisation complète de l'espace des solutions. Dans ce cadre, les techniques de la programmation dynamique [23] et de Branch and bound [39] peuvent être utilisées. Si ces méthodes ont la possibilité de s'affranchir des problèmes de choix des conditions initiales et permettent la convergence vers un minimum global, leur implantation reste redevable à un temps de calcul assez important. Ces inconvénients peuvent affaiblir l'utilisation potentielle de ces méthodes dans des stratégies de commande prédictive non linéaires. Dans ce cas, beaucoup de méthodologies proposées dans la littérature, sont restreintes à des classes particulières de systèmes où des solutions optimales sont facilement calculables. De plus, nombreuses sont les méthodes décrites dans la littérature, qui quoique bien formalisées mathématiquement, restent très difficiles à implanter pratiquement. En effet, afin de faciliter l'étude des systèmes non linéaires, de nombreuses hypothèses concernant le système et sa stabilité sont souvent posées, bien que difficilement vérifiables.

D'un point de vue pratique, il est important d'insister sur le fait que l'optimisation requiert parfois un calcul rigoureux (problème de calcul dans l'optimisation). Toutefois, l'utilisation d'un modèle simple et explicite ainsi qu'une approche d'optimisation efficace peut rendre ces techniques capables de contrôler des systèmes complexes tels que les systèmes non linéaires, les systèmes à phase non minimale, les systèmes instables en boucle ouverte et les systèmes avec contraintes.

I.2.3. Méthodologie d'application de la commande

Dans un contexte linéaire ou non linéaire, après avoir déterminé la séquence de commande optimale (séquences des actions futures) en optimisant la fonction coût, seul le premier échantillon de commande est réellement appliqué au système et toute la séquence est calculée à chaque pas d'échantillonnage. Cette méthodologie repose sur l'exploitation du principe de l'horizon mobile, c'est-à-dire qu'à chaque période d'échantillonnage, la procédure d'optimisation est répétée avec des nouveaux signaux (entrées et sorties)(cf. Figure I.4).

Il est important d'insister sur le fait que souvent la solution optimale est définie sur l'horizon de commande N_u et non pas sur l'horizon de sortie N_2 . Dans ce cas, en dehors de cet horizon, c'est-à-dire, $N_2 > N_u$, la commande reste constante (la variation de l'incrément de commande est nulle). Il est alors possible d'écrire:

$$u(t + j - 1) = \text{constante}, \text{ pour } j = N_u \dots N_2. \quad (\text{I.9})$$

ou encore:

$$\Delta u(t + j - 1) = 0, \text{ pour } j = N_u \dots N_2. \quad (\text{I.10})$$

Dans ce contexte, pour faciliter l'étude de la stabilité, on pose $N_u = N_2 = N$.

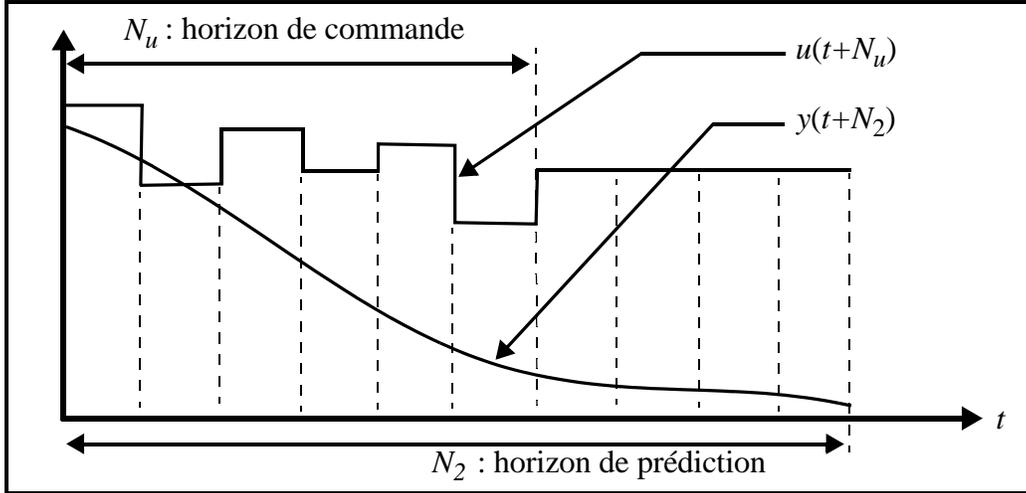


Figure I.4: Principe de la commande prédictive à horizon glissant

I.3. Les Contraintes dans la commande prédictive

Dans toutes les techniques de commande prédictive, les variables d'entrées, d'états et de sorties d'un système sont souvent contraintes par leurs domaines de définitions. Ces contraintes sont de natures diverses: limitation physique des actionneurs, spécification de la qualité d'un produit, exigence de sécurité et domaine de tolérance pour une sortie, ...etc. Par exemple, dans la plupart des réacteurs, la qualité de la production exige qu'une partie des variables ne dépassent pas certaines limites indiquées. En effet, la violation de ces limites peut créer une mauvaise qualité du produit et causer des dégâts sur les équipements.

D'une manière générale, tous les systèmes sont sujets à des contraintes qui doivent être prises en compte dans le problème d'optimisation. Il existe plusieurs types de contraintes, nous avons recensé les plus utilisés dans la commande prédictive.

I.3.1. Contraintes sur le signal de commande

En pratique, le signal de commande doit satisfaire la contrainte du domaine de validité de l'actionneur (son domaine de fonctionnement). Dans ce cas, si on suppose l'intervalle $[u_{min}, u_{max}]$ comme le domaine de validité de l'actionneur, alors la contrainte de commande sur $u(t)$ s'écrit comme suit:

$$u_{min} \leq u(k) \leq u_{max} \quad (I.11)$$

Cette contrainte peut aussi s'exprimer par l'incrément de commande $\Delta u(k) = u(k) - u(k-1)$:

$$\Delta u_{min} \leq \Delta u(k) = u(k) - u(k-1) \leq \Delta u_{max} \quad (I.12)$$

Il est évident que cette contrainte de commande se déploie sur toutes les entrées futures calculées sur un horizon N . Dans ce cas, on peut écrire:

$$\begin{aligned} I \cdot u_{min} \leq \bar{U} \leq I \cdot u_{max} & \quad \text{où:} \quad \bar{U} = [u(k) \dots \dots u(k+N-1)]^T \\ I \cdot \Delta u_{min} \leq \Delta \bar{U} \leq I \cdot \Delta u_{max} & \quad \Delta \bar{U} = [\Delta u(k) \dots \dots \Delta u(k+N-1)]^T \end{aligned} \quad (I.13)$$

avec I vecteur de dimension N avec des éléments égaux à 1.

I.3.2. Contraintes sur la sortie du système

D'une manière générale, il est toujours préférable d'assurer que la sortie d'un système évolue dans une zone de tolérance autour de la trajectoire de référence. Dans ce contexte, si on suppose que l'intervalle de tolérance de la sortie est définie par $[y_{min}, y_{max}]$, alors la loi de commande synthétisée doit assurer l'évolution de la sortie du système à l'intérieur de cette zone de tolérance. Cette contrainte sur la sortie peut s'exprimer par l'équation suivante:

$$y_{min} \leq y(k) \leq y_{max} \quad (I.14)$$

Cette contrainte sur la sortie est généralisée sur toutes les sorties futures sur l'horizon N , c'est-à-dire:

$$I \cdot y_{min} \leq \bar{Y} \leq I \cdot y_{max} \quad (I.15)$$

avec:

- $\bar{Y} = [y(k) \dots \dots \dots y(k + N - 1)]^T$,
- I : un vecteur de dimension N (avec des éléments égaux à 1).

I.3.3. Contrainte sur l'état terminal

Dans ce type de contrainte, la sortie prédite du système est forcée à suivre un signal référence $w(k)$ durant un certain nombre de m périodes après un certain horizon N_y .

$$\begin{aligned} y_m &= [y(k + N_y + 1) \dots \dots \dots y(k + N_y + m)]^T \\ &= [w(k + N_y + 1) \dots \dots \dots w(k + N_y + m)] \end{aligned} \quad (I.16)$$

Dans ce contexte, on peut démontrer que les contraintes sur l'état terminal du système peuvent s'exprimer comme des contraintes d'égalités sur les incréments de la commande futures [11]. Ce type de contrainte est souvent rencontré dans les commandes de type CRHPC (Constrained Receding Horizon Predictive Control) [11].

I.3.4. Autres types de contraintes

D'autres contraintes peuvent être imposées sur le fonctionnement d'un système. Par exemple, dans le cas d'un système à phase non minimale plusieurs situations sont à éviter à cause de la grande divergence de la sortie $y(k)$ pour une entrée $w(k)$. Pour remédier à ce problème, une contrainte peut être ajoutée dans le problème d'optimisation de la commande, c'est-à-dire:

$$\begin{aligned} y(k + j) &\geq y(k) \quad \text{si : } y(k) < w(k) \\ y(k + j) &\leq y(k) \quad \text{si : } y(k) > w(k) \end{aligned} \quad (I.17)$$

I.4. La commande prédictive linéaire

I.4.1. Introduction

Dans ce paragraphe nous supposons que la dynamique du système à commander peut être représentée par un modèle linéaire. Comme discuté précédemment, la méthodologie varie selon le type de modèle de prédiction utilisé. La figure I. 6 donne quelques algorithmes de commande

prédictive linéaire, CPL.

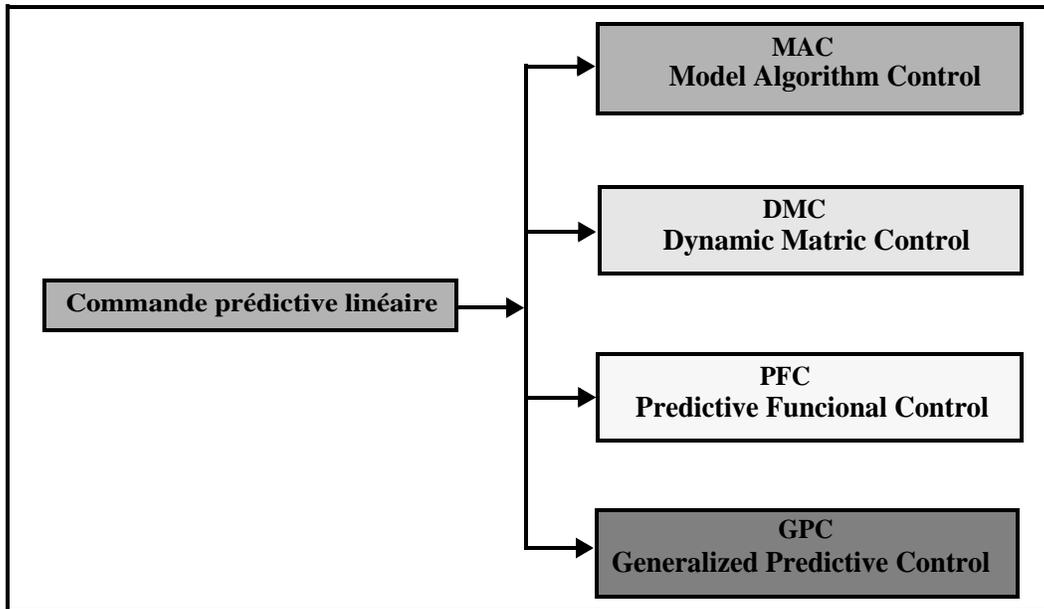


Figure I.5: Différentes méthodes de commande prédictive linéaire.

Figure I.6

A. La méthode MAC (Model Algorithmic Control)

Cette méthode développée par Culter et Ramaker [12] a un grand succès dans l'industrie chimique. Elle repose sur:

- L'utilisation de la réponse impulsionnelle comme modèle de prédiction (cf. équation (I.1)), où seulement les N_2 premiers termes sont pris en compte.
- La restriction à des systèmes stables et sans intégrations.

En l'absence des contraintes, il a été démontré dans [4], que la solution optimale minimisant le critère (I.8) est facilement déterminée.

De part sa simplicité, cette méthode est largement utilisée dans le monde industriel. Cependant, son inconvénient majeur réside dans sa représentation non minimale. De plus, le nombre des poids h_i ($i = 1 \dots N_2$) à identifier pour des systèmes complexes est souvent très important ce qui limite son déploiement à une large échelle.

B. La Commande DMC (Dynamic Matrix Control)

Cette méthode proposée par Culter et Ramaker [12] repose sur:

- L'utilisation de la réponse indicielle comme modèle de prédiction du système à commander (cf. équation Tableau 2, page 9), où seulement les N_2 premiers termes sont pris en compte.
- La restriction à des systèmes stables et sans intégrations.

De la même manière que la méthode MAC, la séquence de commande est obtenue par minimisation de la fonction du coût (I.8) où une solution optimale peut être trouvée facilement [4]. Les inconvénients cités précédemment pour la méthode MAC reste valable pour cette méthode.

C. La commande PFC (Predictive Functional Control)

Le principe de cette méthode proposée par Richalet [38] consiste à:

- chercher un modèle de prédiction imitant un modèle de référence sous forme d'équation

d'état imposé par l'utilisateur.

- attacher une composante additive à ce modèle de prédiction pour prendre en compte les incertitudes.

La loi de commande est obtenue par minimisation d'un critère quadratique sous la forme (I.8) où la contrainte sur l'énergie minimale de la commande est relaxée. Dans la même manière que précédemment, il à été démontré [37] qu'une solution optimale à ce problème peut être trouvé.

D. La commande GPC (Genarelized Predictive Control)

Cette commande suppose que la dynamique du système peut être représenté par un système CARMA (Controlled AutoRegressive Moving Average) donnée par l'équation Tableau 4, page 9. Le modèle de prédiction est obtenue en résolvant une équation de diophontine [9]. La séquence de commande est déterminée par minimisation d'un critère sous la forme (I.6). Nous allons revenir sur les détails de cette technique dans les paragraphes suivants.

I.4.2. Synthèse et étude de la stabilité de la commande prédictive linéaire

Dans cette section, la Commande Prédictive Linéaire (C.P.L) d'un point de vue synthèse et stabilité est présentée et formalisée dans le contexte de la représentation d'état.

I.4.2.1. C.P.L avec un horizon de commande et de prédiction finis

Le problème d' optimisation est de type quadratique programming «QP» donnée par l'équation suivante:

$$\text{Min}_{\Pi} \quad V_{N_2}(x(k), \Pi), \text{ avec: } \begin{cases} u(k+j/k) \in U \\ x(k+j/k) \in X \end{cases} \quad (\text{I.18})$$

où X et U sont respectivement les domaines définis par les contraintes sur les variables d'état et sur la commande. Ces derniers sont définies par les inéquations linéaires suivantes:

$$\begin{aligned} X &= \{x/G^x x < g^x \in R^n\} \\ U &= \{u/G^u u < g^u \in R^m\} \end{aligned} \quad (\text{I.19})$$

où g^x et g^u sont des vecteurs constants strictement positives.
 G^u, G^x matrices de contraintes.

Le modèle de prédiction est donné par I.3. Dans ce cas, la critère à optimiser peut s'écrire sous la forme suivante:

$$\begin{aligned} V_{N_2}(x(k), \Pi) &= \sum_{j=0}^{N_2-1} x^T(k+j/k) Q x(k+j/k) + \sum_{j=0}^{N_u-1} (u^T(k+j/k) R u(k+j/k)) \\ \text{avec: } &\begin{cases} x(k+j+1/k) = Ax(k+j/k) + Bu(k+j/k) \\ Q \geq 0, R > 0 \end{cases} \end{aligned} \quad (\text{I.20})$$

La variable de la commande est supposée nulle après l'horizon de commande N_u , c'est-à-dire:

$$u(k+j/k) = 0, j \geq (k+j+N_u). \quad (\text{I.21})$$

Cet algorithme ne garantit pas une stabilité pour un quelconque choix de Q, R, N_u, N_2 . Dans ce contexte, la plupart des praticiens vérifient à travers des simulations si leur choix de paramètres

est pertinent et conduit à une solution stable. Bien évidemment, ceci n'est pas suffisant et il faut s'orienter vers la recherche des algorithmes plus performants. Parmi ces algorithmes, nous allons citer deux approches dans les paragraphes suivants.

I.4.2.2. C.P.L avec un horizon de commande et de prédiction infinis

La synthèse de ce contrôleur repose sur une transformation, après quelques manipulations, d'un problème à horizon infini en un problème à horizon fini. Sous l'hypothèse que pour $j \geq N_u$, la loi de commande [39] appliquée est un retour d'état:

$$u(k+j-1/k) = -Kx(k+i-j/k) \quad (I.22)$$

Dans ce cas, la fonction coût peut être réécrite de la manière suivante:

$$V_\infty(x(k), \Pi) = \sum_{j=0}^{N_u-1} x^T(k+j/k)Qx(k+j/k) + u^T(k+j/k)Ru(k+j/k) \\ + \sum_{j=N_u}^{\infty} (x^T(k+j/k)Qx(k+j/k) + u^T(k+j/k)Ru(k+j/k)) \quad (I.23)$$

En développant le second terme de l'expression précédente, il résulte:

$$\sum_{j=N_u}^{\infty} x^T(k+j/k)Qx(k+j/k) + u^T(k+j/k)Ru(k+j/k) = \\ x^T(k+N/k)(Q-K^TRK)x(k+N/k) + \\ x^T(k+N+1/k)(Q-K^TRK)x(k+N+1/k) + \\ \dots \dots \dots \\ x^T(k+N+j/k)(Q-K^TRK)x(k+N+j/k) + \quad (I.24)$$

ou encore d'une manière équivalente:

$$= x^T(k+N/k)(Q-K^TRK)x(k+N/k) + \\ x^T(k+N/k)(A-BK)^T(Q-K^TRK)(A-BK)x(k+N/k) + \\ \dots \dots \dots \\ x^T(k+N/k)(A-BK)^{Tj}(Q-K^TRK)(A-BK)^j x(k+N/k) + \\ \dots \dots \dots \quad (I.25)$$

soit:

$$\sum_{j=N_u}^{\infty} x^T(k+j/k)Qx(k+j/k) + u^T(k+j/k)Ru(k+j/k) = \\ x^T(k+N/k) \sum_{i=0}^{\infty} [(A^T-K^TB^T)^i (Q-K^TRK)(A-BK)^i] x(k+N/k) \quad (I.26)$$

avec:

$$P = \sum_{i=0}^{\infty} (A^T-K^TB^T)^i (Q-K^TRK)(A-BK)^i \quad (I.27)$$

Cette série converge si le système est stable. Dans ce cas, il est possible d'écrire:

$$P = (Q + K^TRK) + \sum_{i=1}^{\infty} [(A^T-K^TB^T)^i (Q + K^TRK)(A-BK)^i] \\ = Q + [(A-BK)^T P (A-BK)] + K^TRK \quad (I.28)$$

A travers ce résultat, on peut dire que le problème de la commande prédictive à base de modèle à horizon infini se transforme en un problème de minimisation à horizon fini. Ce dernier est donné par la formulation suivante:

$$\left\{ \begin{array}{l} \text{Min}_{\Pi} \left(\begin{array}{l} \sum_{j=0}^{N_u-1} x^T(k+j/k) Q x(k+j/k) + u^T(k+j/k) R u(k+j/k) \\ + (x^T(k+N/k) P x(k+N/k)) \end{array} \right) \\ \text{avec: } u(k+j/k) \in U, \text{ et } : x(k+j/k) \in X \end{array} \right. \quad (\text{I.29})$$

Si au lieu de considérer $u(k+j-1/k) = -Kx(k+j-1/k)$ pour $j \geq N_u$ on utilise $u(k+j-1/k) = 0$ [35], ceci implique qu'on laisse le système fonctionner en boucle ouverte. Bien évidemment, ceci n'est pas possible que si le système est stable en boucle ouverte. Dans ce cadre, Scokaert et Rawlings [40] ont proposé l'algorithme suivant:

Etape 0

- Choix d'un horizon fini N_2

Etape 1

- Résolution du problème Tableau 18, page 16-Tableau 20, page 16

Si $x(k+N_2/k) \in X_k$

où X_K est un ensemble invariant positive dans le quel le système linéaire avec la loi de commande $u(k)=-Kx(k)$ satisfait tous les états et les contraintes sur un horizon infini.

$$X_K = \left\{ \begin{array}{l} x(k) \in R^n : x(k+1) = (A - BK)x(k), u(k) = -Kx(k) \\ \text{avec: } G^u u(k) \leq g^u, G^x x(k) \leq g^x, \forall k \end{array} \right.$$

Allez à l'étape 2

Si non

- augmenter N_2

Allez à l'étape 1

Fin de Si

Etape 2:

- $\Pi^* = \Pi$.

I.4.2.3. C.P.L avec un horizon de prédiction infini

L'approche développée dans les travaux de Rawlings et Muske [35] est un régulateur à horizon infini en utilisant un ensemble de variables de décision fini:

$$\begin{aligned} V_{N_u}(x(k), \Pi) = & \sum_{j=0}^{N_u-1} x^T(k+j/k) Q x(k+j/k) + u^T(k+j/k) R u(k+j/k) \\ & + \sum_{j=N_u}^{\infty} x^T(k+j/k) Q x(k+j/k) \end{aligned} \quad (\text{I.30})$$

En supposant $u(k+j/k) = 0, \forall j > k + N_u$, ce qui garantit la convergence vers zero de la partie infinie de la fonction de coût.

Rawlings et Muske [35] imposent que les modes instables du système doivent être mis à zéro à la fin de l'horizon de commande. Le reste de la somme serait limité à l'évolution libre des

La stabilité dans ce cas est équivalente à la controlabilité des modes instables qui sont identifiés par la matrice de stabilité:

$$S = \begin{bmatrix} M_u^T B & T_{22} M_u^T B & \dots & T_{22}^{n_u} M_u^T B \end{bmatrix} \quad (\text{I.39})$$

où n_u représente la dimension de T_{22} : le nombre des modes instables $n_u \leq n$.

Pour les systèmes linéaires sans contraintes, la stabilisation est équivalente à S ayant le rang plein. Dans le cas où la variable de commande $u(k)$ a des contraintes, le rang de S est une condition nécessaire pour la contrôlabilité des modes instables à l'origine. Cette dernière condition n'est pas suffisante. En effet, il faut également intégrer une condition sur le choix de l'horizon de commande pour vérifier la contrôlabilité ou la stabilité dans le cas de la présence des contraintes sur la variable de commande. D'une manière générale, il n'y a aucune forme ou méthode pour vérifier la contrôlabilité en présence de contraintes sur les entrées.

I.5. Commande Prédictive Non-linéaire (C.P.N)

I.5.1. Introduction

Le modèle du procédé peut être écrit sous la forme discrète donnée par l'équation I.6 où $x(k) \in X \subset R^n$, $u(k) \in U \subset R^m$, $y \in R^p$ où X et U sont respectivement les domaines définis par les contraintes sur les variables d'état et sur la commande. Dans toute la suite, des hypothèses communes, à savoir que le couple d'équilibre $(0,0)$ est tel que: $f(0,0) = 0$ et qu'il existe toujours un changement de coordonnées pour s'y ramener.

La synthèse de loi de commande est déterminée en minimisant le critère donné par l'équation Tableau 8, page 11 où l'évolution prédite du système à l'instant $k+j+1$ est donnée par le modèle Tableau , page 20 de prédiction non linéaire (I.6) et peut s'écrire comme suit:

$$\left\{ \begin{array}{l} x(k+j+1/k) = f(x(k+j/k), u(k+j/k)) \\ \text{avec: } x(k/k) \equiv x(k) \\ u(k+j/k) \in U \quad j = 0, \dots, N-1 \\ x(k+j/k) \in X \quad j = 0, \dots, N-1 \end{array} \right. , \quad (\text{I.40})$$

avec une contrainte terminale :

$$x(k+N/k) = 0 \quad (\text{I.41})$$

ou

$$x(k+N/k) \in X_f \quad (\text{I.42})$$

- X_f : c'est la région terminale (région d'attraction)

Etant donné que $x(k/k) = x(k)$ est l'état actuel du système, le problème d'optimisation est un problème de programmation mathématique dans lequel les variables de décisions sont les éléments de la séquence de commandes futures $\Pi(x)$ d'une taille $N \times m$. Dans ce contexte:

- Si les fonctions qui interviennent dans le problème d'optimisation: F , V et L sont des fonctions continues
- si l'ensemble U est compact, et si les ensembles X et X_f sont fermés

alors on peut garantir au moins une solution admissible au problème d'optimisation ci dessus[41]. En effet, ce problème d'optimisation dépend de l'état du système $x(k)$ à chaque instant k et sa solution est obtenue en appliquant le principe de l'horizon glissant à chaque période d'échantillonnage k pour obtenir la séquence de commande optimale $\Pi^*(x)$. Par la suite, seulement le premier élément de cette séquence sera appliqué sur le système, c'est-à-dire:

$$u_{opt} = u^*(k/k) \quad (I.43)$$

Il est important d'insister sur le fait qu'en absence de la divergence entre le système à commander et son modèle de prédiction, l'équation (22) du modèle représentera le système, c'est à dire $x(k+1) = x(k+1/k)$. Cependant, dans certaines situations la commande optimale à l'instant $k+j$ à partir de $x(k)$ peut être différente de la commande optimale au même instant à partir de $x(k+1)$, c'est à dire:

$$u^*(k+1/k) \neq u^*(k+1/k+1). \quad (I.44)$$

Pour cela les trajectoires optimales prédites à chaque instant sont distinctes. Par conséquent l'évolution du système en boucle fermée est différente de l'évolution de la solution prédite du système.

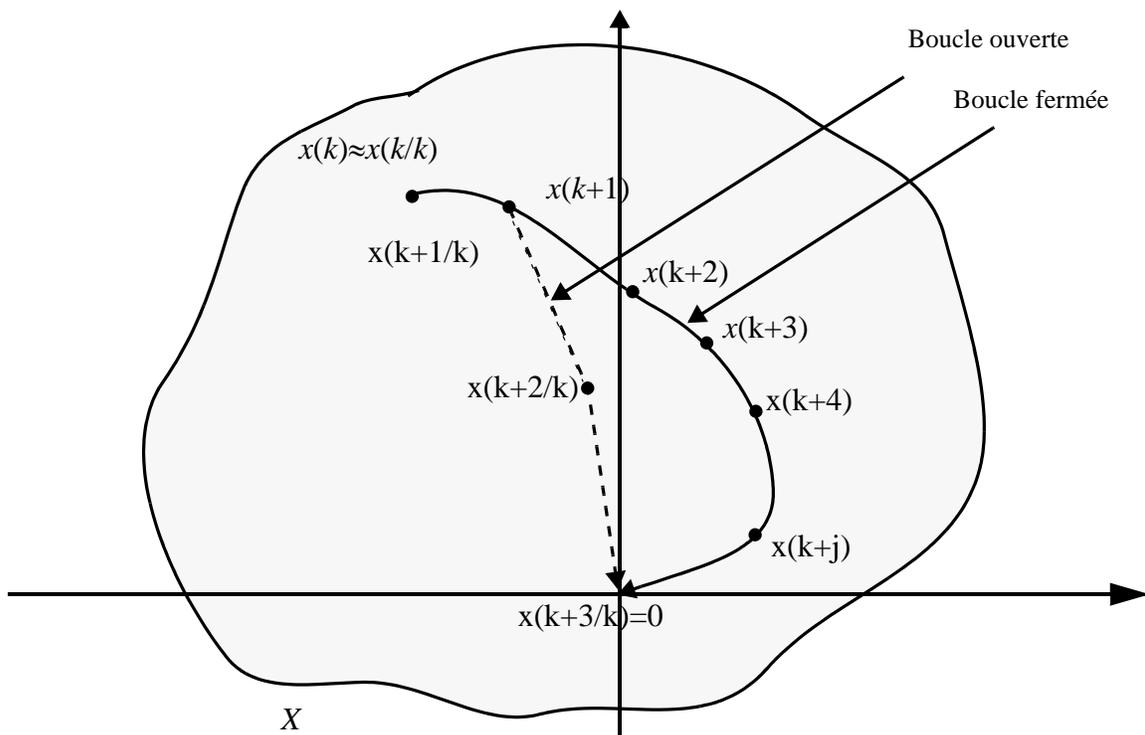


Figure I.7: Performance de la commande prédictive à horizon fini ($N=3$) en BO et BF

I.5.2. Synthèse et étude de la stabilité de la commande prédictive Non Linéaire

Dans un premier temps, on va présenter l'étude de la stabilité d'un contrôleur prédictif non linéaire, en commençant par un contrôleur avec un horizon infini. Comme le problème d'optimisation dans ce cas est insolvable, on s'oriente vers un contrôleur prédictif non linéaire à horizon fini. Dans ce contexte, une contrainte sur l'état final qui permet d'imposer un certain degré de stabilité pour l'optimisation est assumée. En revanche, l'exigence de trouver un minimum global qui peut satisfaire la contrainte finale à zéro, n'est pas possible dans un temps fini, ce qui peut rendre le problème d'optimisation infaisable en pratique. La solution de ce problème consiste en l'application d'une commande hybride, ou bien la recherche d'une solution sous opti-

mal qui assure la convergence et la stabilité, en résolvant un problème d'optimisation qui n'exige pas une recherche d'un optimum global.

I.5.2.1. Conditions nécessaire sur la stabilité

La plupart des travaux sur la stabilité font l'hypothèse que le problème d'optimisation à une solution optimale ou admissible et réalisable. L'algorithme d'optimisation choisi jouera un rôle primordial dans l'étude de la stabilité de la commande prédictive.

C'est pour cette raison que la majeure partie des hypothèses rencontrées dans la littérature portent sur le critère. On présente ici les hypothèses de l'existence d'une solution optimale

A. Hypothèses sur le critère

- Décroissance du critère: cette hypothèse revient à supposer que:

$$\Delta V(x) = V(x(k+1)) - V(x) < 0 \quad (\text{ou } V(x(k+1)) - V(x) \leq 0)$$

Cette hypothèse est primordiale pour la stabilité car elle assure la décroissance du critère tout au long des trajectoires.

- La propriété de fonction propre du critère par rapport à la commande u : c'est-à-dire:

$$V(., u) \rightarrow +\infty \quad \text{quand} \quad \|u\| \rightarrow +\infty$$

Cette hypothèse permet de construire un ensemble compact dans lequel la solution au problème d'optimisation est garantie [26].

B. Hypothèses sur le système et ses contraintes

L'ensemble défini par les contraintes n'est pas vide et contient le couple de la solution (0,0), ce qui confère une commandabilité ou une faisabilité de la solution. Quelque soit l'état initial, on suppose l'état (0,0) est atteignable. Plusieurs auteurs [26], ont utilisé une version du théorème de Lyapunov pour démontrer la stabilité d'un contrôleur prédictif non linéaire à horizon fini et infini.

I.5.2.2. Théorème [29]

Soit $V: R^n \rightarrow R^+$ qui satisfait les conditions suivantes:

1. S'il existe une fonction α non négative et non décroissante telle que:

$$\alpha(0) = 0, \text{ et } 0 < \alpha(\|x(k)\|) \leq V(x(k)), \forall x(k) \neq 0$$

2. S'il existe une fonction γ strictement croissante et continue à l'origine telle que:

$$\gamma(0) = 0, \text{ et } V(x(k)) - V(x(k+1)) \geq \gamma(\|x(k)\|)$$

3. Si $V(0) = 0$ et V est continue.

Alors le système défini par l'équation Tableau , page 20(22) est asymptotiquement stable.

L'application de ce théorème exige la satisfaction des hypothèses suivantes sur la fonction L , et la fonction à optimiser V_N .

- H₁: $L(0, 0) = 0$ et L est continue à l'origine.
- H₂: Il existe une fonction K^+ et $\gamma: [0, \infty) \rightarrow [0, \infty)$ telle que:

$$0 < \gamma(\|x, u\|) \leq L(x, u) \quad \forall (x, u) \neq (0, 0)$$
- H₃: V_N^* (l'optimum de V_N) est continue à l'origine.

Ces hypothèses conduisent à des propriétés sur L, V_N^* :

- P₁: $L(x, u) > 0 \quad \forall (x, u) \neq (0, 0)$
- P₂: $L(x, u) \rightarrow (0, 0) \Rightarrow (x, u) \rightarrow (0, 0)$
- P₃: $L(x, u) = (0, 0) \Rightarrow (x, u) = (0, 0)$
- P₄: $V_N^*(x) \geq 0$
- P₅: $V_N^*(x) = 0 \Leftrightarrow x = 0$
- P₆: $\gamma(\|x\|) \leq V_N^*(x)$
- P₇: $V_{N+k}^*(x) \leq V_N^*(x) \quad \forall k \geq 0$

Donc si les trois hypothèses H₁, H₂ et H₃ sont satisfaites, alors l'origine est un point d'équilibre asymptotiquement stable pour le système dynamique:

$$x(k+1) = f(x(k), u^*(x(k))). \quad (\text{I.45})$$

I.5.2.3. Démonstration

L'étude de la stabilité est réalisée pour le cas d'un horizon fini et infini. L'idée principale est basée sur l'utilisation d'une solution admissible pour vérifier la stabilité asymptotique. Cela est réalisé en deux parties. La première étape est l'utilisation des deux premières hypothèses H₁ et H₂ et les propriétés de $V_N^*(x)$ pour montrer la convergence de la solution vers zéro. La deuxième étape de la démonstration est basée sur la troisième hypothèse H₃ qui est l'argument clé de la stabilité. La stabilité combinée avec la convergence implique la stabilité asymptotique.

On va considérer que l'horizon de la prédiction et de la commande sont égaux, c'est-à-dire:

$$N_u = N_2 = N \quad (\text{I.46})$$

Premièrement, on considère que le critère n'a pas de contrainte terminale de la façon suivante:

$$V_N(x) = \sum_{j=k}^{k+N-1} L(x(j/k), u(j/k)) \quad (\text{I.47})$$

et

$$\begin{aligned} x(k+j/k) &= f(x(k+j-1/k), u(k+j-1/k)) \\ x(k/k) &\equiv x(k) \\ u(k+j/k) &\in U \\ x(k+j/k) &\in X \end{aligned} \quad (\text{I.48})$$

A. Convergence

- Pour un horizon infini, on a:

$$\begin{aligned}
V_{\infty}^*(x(k)) &= L(x(k), u^*(k/k)) + L(x^*(k+1/k), u^*(k+1/k)) + \dots \\
&= L(x(k), u^*(k)) + V_{\infty}(f(x(k), u^*(k)), \tilde{\Pi}) \\
&= L(x(k), u^*(k)) + V_{\infty}(x(k+1), \tilde{\Pi}) \\
V_{\infty}^*(x(k)) &\geq L(x(k), u^*(k)) + V_{\infty}^*(x(k+1))
\end{aligned} \tag{I.49}$$

avec $\tilde{\Pi} = \{u^*(k+1/k), u^*(k+2/k), \dots\}$

La dernière inéquation est obtenue à partir du principe d'optimalité. Dans ce cas, on remarque facilement que:

$$V_{\infty}^*(x(k)) - V_{\infty}^*(x(k+1)) \geq L(x(k), u^*(k)) \tag{I.50}$$

Comme la fonction L est positive, alors $V_{\infty}^*(x(k))$ est une fonction décroissante. Elle est borné par zéro et par conséquent à une limite quand $k \rightarrow \infty$. Alors la partie gauche de l'équation (I.50) converge vers zéro et par conséquent:

$$L(x(k), u^*(k)) \rightarrow 0 \tag{I.51}$$

Enfin la propriété 2 de L implique que la séquence $(x(k), u(k)) = (x(k), u^*(k)) \rightarrow 0$ converge vers zéro.

- Pour un horizon fini, on a:

$$\begin{aligned}
V_N^*(x(k)) &= L(x(k), u^*(k/k)) + \dots + L(x^*(k+N-1/k), u^*(k+N-1/k)) \\
&= L(x(k), u^*(k)) + V_{N-1}(f(x(k), u^*(k)), \tilde{\Pi}) \\
V_N^*(x(k)) &\geq L(x(k), u^*(k)) + V_{N-1}^*(f(x(k), u^*(k)))
\end{aligned} \tag{I.52}$$

où $\tilde{\Pi}$ est donnée par: $\tilde{\Pi} = (u^*(k+1/k), u^*(k+2/k), \dots, u^*(k+N-1/k))$.

Selon la propriété 7 sur la fonction d'objectif d'un contrôleur à horizon glissant $V_N^*(x)$, on a:

$$V_{N+k}^*(x) \leq V_N^*(x) \quad \forall k \geq 0 \tag{I.53}$$

Cette inégalité implique que:

$$V_N^*(x(k)) \geq L(x(k), u^*(k)) + V_N^*(f(x(k), u^*(k)))$$

Enfin, on remarque que:

$$V_N^*(x(k)) - V_N^*(x^+(k)) \geq L(x(k), u^*(k)) \tag{I.54}$$

Avec cette inéquation on montre la convergence de la fonction d'objectif du contrôleur glissant à horizon fini de la même manière que dans le cas d'un horizon infini.

B. Stabilité

- On choisit $R > 0$. A partir de la continuité de V_N^* à l'origine, il existe un $r > 0$ tel que:

$$V_N^*(x(k)) < \gamma(R) \text{ pour tout } \|x(k)\| < r$$

Selon l'équation (I.54), on a $V^*(x(j)) \leq V^*(x(k))$ pour tout $j > k$.

Maintenant on suppose que le contrôleur à horizon glissant fini ou infini n'est pas stable.

- Dans ce cas, pour un R suffisamment petit $\|x(k)\| < r$ et $j > k$, alors $\|x(j)\| > R$. Comme γ est une fonction non décroissante, la borne inférieurs de V^* est donnée par:

$$V^*(x(j)) \geq \gamma(\|x(j)\|) \geq \gamma(R)$$

Par combinaison des deux inégalités, il résulte que:

$$V^*(x(j)) \geq \gamma(\|x(j)\|) \geq \gamma(R) > V^*(x(k)) \geq V^*(x(j))$$

ce qui conduit à une contradiction. Donc la stabilité est vérifiée en plus de la convergence du contrôleur. Ceci impliquera la stabilité asymptotique du contrôleur.

Dans le cas d'un horizon infini sous l'existence du couple de solution (x, u) , le critère est décroissant, donc l'algorithme converge vers cette solution. Cependant, pour des raisons de faisabilité numérique qui conduit à un problème d'optimisation insolvable, cet horizon infini est souvent approximé par un horizon fini glissant. En revanche, en pratique la recherche en ligne d'une solution optimale globale d'un problème d'optimisation non linéaire (non convexe) sous contraintes, dans un temps fini N , n'est en général pas faisable. Pour cela plusieurs auteurs [18][36] ont fournit les conditions nécessaires de la stabilité de la commande prédictive en imposant une contrainte terminale sur l'état final de type (cf.Figure I.8):

$$x(k + N/k) = 0, \tag{I.55}$$

Autrement dit, à la fin de l'horizon de prédiction l'état du système doit être nul.

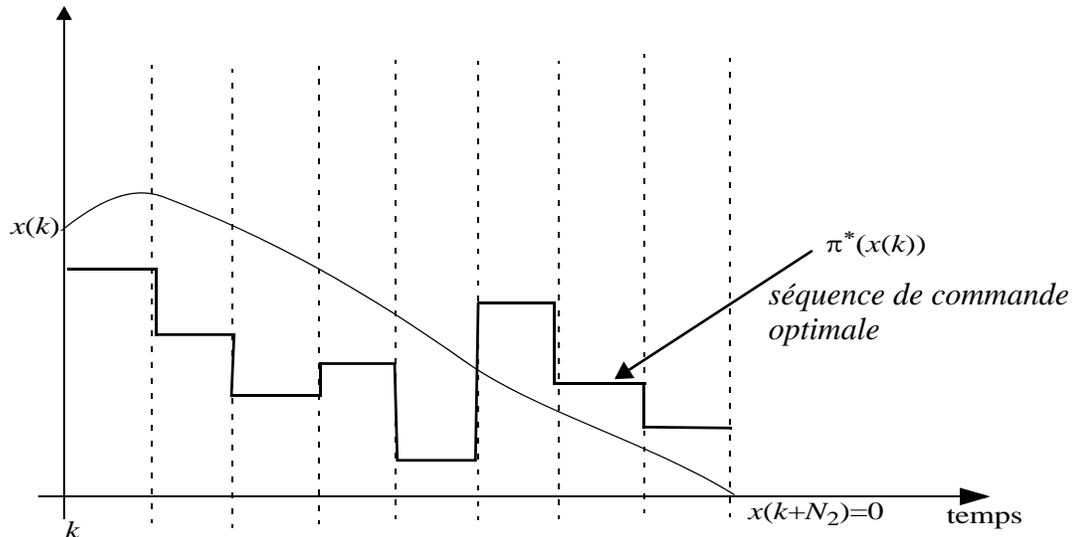


Figure I.8 La séquence optimale de la C.P.N.L

Dans ce cas, il faut tout d'abord trouver une nouvelle séquence faisable Π . Le fait d'exiger que l'état terminal $x(k + N/k) = 0$, implique que cette séquence sera choisie comme suit:

$$\Pi = (u^*(k + 1/k), u^*(k + 2/k), \dots, u^*(k + N - 1/k), 0) \quad (I.56)$$

Ce permet d'écrire:

$$\begin{aligned} V_N^*(x(k)) &= \\ L(x(k), u^*(k/k)) + \dots + L(x^*(k + N - 1/k), u^*(k + N - 1/k)) + L(0, 0) \\ &= L(x(k), u^*(k)) + V_N^*(f(x(k), u^*(k)), \tilde{\Pi}) \\ V_N^*(x_k) &\geq L(x(k), u^*(k)) + V_N^*(f(x(k), u^*(k))) \end{aligned} \quad (I.57)$$

En fait, cette contrainte Tableau 55, page 25 permet, en théorie, de forcer la convergence de l'algorithme d'optimisation à la fin de l'horizon de prédiction. Par ce fait, elle assure la décroissance de la fonction de coût qui devient une condition nécessaire pour la stabilité de la trajectoire d'équilibre du système en boucle fermée (cf. Figure I.9).

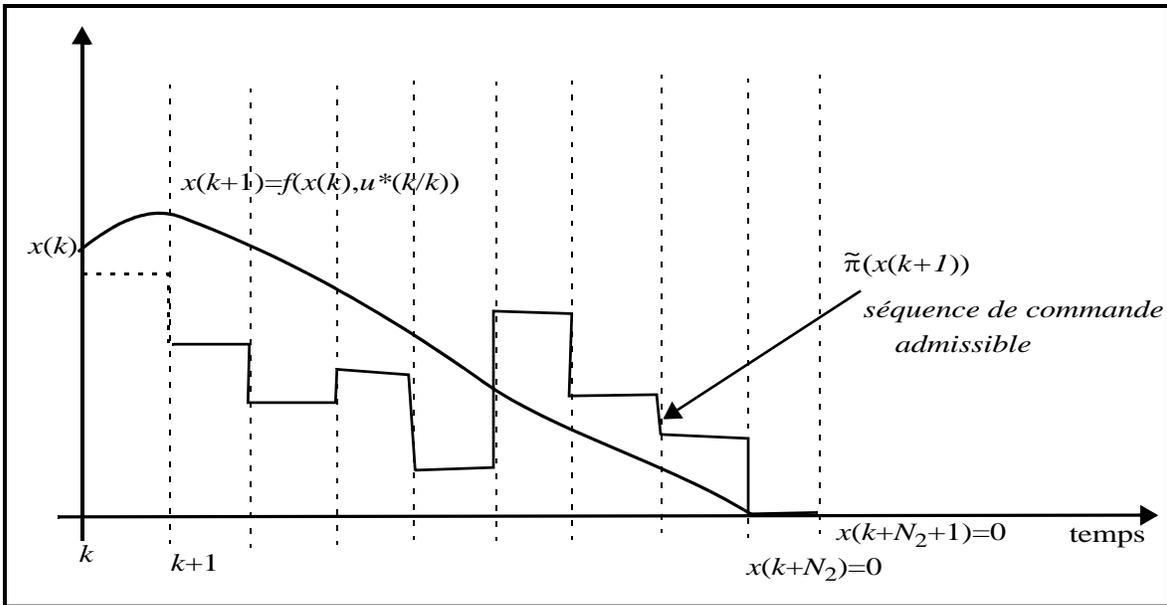


Figure I.9 : La séquence admissible de la C.P.N.L

Dans la même continuité de ce raisonnement Mazen Alamir [28] montre que si :

- $\Delta V_N^*(x(k)) \leq -L(x(k), u^*(k))$ et
- $L(x(k), u^*(k))$ est définie positive

Alors, le principe de LaSalle permet de conclure à la stabilité globale asymptotique [20]. C'est-à-dire :

$$\lim_{k \rightarrow \infty} (x(k), u^*(k)) = (0, 0) \quad (I.58)$$

Cependant, cette contrainte est assez restrictive. En effet, elle traduit la commandabilité exacte du systèmes en temps fini. Dans ce cadre, la détermination à la fin de l'horizon de la solution optimale globale par l'algorithme est difficilement réalisable en pratique. C'est pour ces raisons que cette contrainte terminale est relaxée dans [1],[6]et [34] en le transformant en une contrainte d'inégalité terminale:

$$x(k + N/k) \in X_f \quad (\text{I.59})$$

X_f représente la région terminale constituant un domaine d'attraction pour le système non linéaire. Dans ce contexte, Mayne et Michalska [27] proposent un contrôleur dual, où en dehors de la région terminale, un régulateur à horizon glissant est appliqué. En revanche, à l'intérieur (région d'attraction) une loi de retour d'état linéaire sera appliquée avec la condition que l'état à la fin de l'horizon fini se trouve sur la frontière de la région terminale. Dans ce cas, on impose une contrainte sur la fonction de coût dans la région terminale comme suit:

$$L(x(k), Kx(k)) = 0 \quad \forall x(k) \in X_f \quad (\text{I.60})$$

Le succès de cette méthode est bien basée sur le principe suivant:

Etant donnée une linéarisation du système non linéaire f autour du point d'origine,

$$A = \partial f(x, u) / \partial x|_{(0,0)}, B = \partial f(x, u) / \partial u|_{(0,0)} \quad (\text{I.61})$$

si la matrice (A, B) est contrôlable, alors il existe une matrice de gain K tel que le système en boucle fermée $x(k + 1) = f(x(k), Kx(k))$ est localement asymptotiquement stable.

Pour vérifier ce principe il faut vérifier les conditions triviales suivantes:

- la région terminale X_f appartient a l'ensemble X .
- $Kx(k) \in U$ pour tout $x(k) \in X_f$
- l'origine doit être à l'intérieur de l'ensemble défini par les contraintes X et U
- La région terminale doit être petite.

I.5.3. Commande prédictive sous optimale

La question qui se pose est de savoir s'il est possible de remplacer l'optimisation globale par la recherche de la séquence de commande qui assure la décroissance de la fonction de coût afin d'éviter les inconvénients de l'algorithme de la commande prédictive non linéaire avec une égalité terminale. Surtout pour ce qui concerne la recherche d'un optimal global à chaque période d'échantillonnage, ce qui rend le problème d'optimisation très prohibitif coté calcul et inapplicable (infaisable) en pratique. Dans ce contexte, Scokaert et all [41] ont proposé la stratégie de la commande prédictive sous optimale, qui se base sur la recherche d'une solution admissible mais non optimale, en vérifiant les conditions de la convergence. Cette méthode est résumée par l'algorithme suivant: [41].

Algorithme de la commande prédictive sous optimale avec une contrainte terminale

Etape 0 : $k=0$

- trouver la séquence de commande en boucle ouverte
 $\Pi_N = \{u(0/0), u(1/0), \dots, u(N-1/0)\}$ qui satisfait Tableau 40, page 20 et Tableau 42, page 20
- $u(0)=u(0/0)$.

Etape 1: $k > 0$, état $x(k)$

- trouver la séquence de commande Π_k qui satisfait Tableau 40, page 20 et Tableau 42, page 20 et

$$V_N(x(k), \Pi_k) \leq V_N(x(k-1), \Pi_{k-1}) - \eta L(x(k-1), u(k-1))$$

en utilisant $\tilde{\Pi} = \{u(k/(k-1)), \dots, u(k+N-2/k-1), 0\}$ comme estimation initiale.

- $u(k) = u(k/k)$
- aller à l'étape 1.

Le facteur $\eta \in [0, 1]$ est introduit pour le cas où il y a des perturbations. Dans le cas contraire on prend $\eta = 1$. On remarque que la condition $\eta < 1$ conduit à la décroissance de la fonction du coût ce qui implique la convergence de $(x(k), u(k))$.

I.5.4. Conclusions sur Stabilité de la commande prédictive non Linéaire

Dans la littérature, plusieurs méthodes pour étudier la stabilité ont été proposées. Ces méthodes utilisent souvent le critère de la commande prédictive comme une fonction de Lyapunov. La méthode la plus utilisée est la méthode directe où il s'agit de chercher des conditions sur F, X_f, k_f pour assurer que:

$$\Delta V_N(x, k_f(x)) + l(x, k_N(x)) \leq 0 \quad (\text{I.62})$$

Cette méthodologie est employée, entre autres, dans les travaux de Keerthi et Gilbert [18] et Mayne et Michalska [27], Michalska et Mayne [32], Rawlings et Muske [35], Sokaert et Rawlings [40], Chen et Allgower [6] où le calcul de la fonction:

$$V_N(x^+, x^+ = f(x, k_f(x)))$$

est remplacé par le calcul de la borne supérieure $V_N(x^+, \tilde{u}(x))$ de $V_N(x^+)$ en utilisant la solution admissible non optimale $\tilde{u}(x)$ de $P_N(x^+)$. Donc pour vérifier la stabilité asymptotique de la boucle fermée du système, il faut que les hypothèses suivantes soient vérifiées:

$$\begin{aligned} A1: & X_f \subset X \text{ fermé}, 0 \in X_f \\ A2: & k_f(x) \in U, \forall x \in X_f \\ A3: & f(x, k_f(x)) \in X_f, \forall x \in X_f \\ A4: & [\Delta F + L](x, k_f(x)) \leq 0, \forall x \in X_f \end{aligned} \quad (\text{I.63})$$

Sans oublier le rôle fondamental joué par:

- La contrainte finale.
- La condition que le critère soit définie positif par rapport à x .
- Le caractère optimal de la solution (au moins sous-optimal).

I.6. Conclusion

Dans ce premier chapitre, le principe de base de la commande prédictive à base de modèles (linéaires et non linéaires) a été présenté. Nous avons également décrit les contraintes essentielles que peut subir un système dans le contexte de la commande prédictive. Sans faire référence à toutes les techniques de commande prédictive existantes, nous avons discuté les problèmes d'analyse de la stabilité et la synthèse de correcteurs pour des systèmes linéaires et non linéaires. Quelques conditions et hypothèses ont été supposées afin que les problèmes de commande et d'optimisation puissent être résolus.

Enfin, nous avons pu montrer que dans tous les cas étudiés que la convergence de l'algorithme d'optimisation est une condition nécessaire pour assurer la stabilité de l'algorithme de commande prédictive. Cette stabilité qui reste un domaine difficile et vaste pour les recherches futures.

CHAPITRE II:
COMMANDE PREDICTIVE
NON LINEAIRE FLOUE

II.7. Introduction

Nous avons vu dans le chapitre précédent que la commande prédictive non linéaire a pour objectif la recherche d'une séquence de commande optimale en minimisant un critère non linéaire (souvent non convexe). La résolution de ce problème d'optimisation par des méthodes séquentielles, comme la programmation quadratique séquentielle SQP [3] n'est pas satisfaisante. En effet, ces algorithmes convergent souvent vers des optimaux locaux, ce qui influe largement sur les performances de la commande prédictive non linéaire. En revanche, l'utilisation des techniques d'optimisation discrètes, telles que la programmation dynamique, Branch & Bound et les algorithmes génétiques [16][23][44], peuvent résoudre ce problème de convergence. Toutefois, malgré les avantages de ces techniques discrètes, quelques inconvénients persistent, notamment leur lourdeur ainsi que les complexités qu'elles engendrent. Ces constatations ont été relatées dans plusieurs travaux de recherche [43], surtout pour des systèmes rapides, à savoir les systèmes non linéaires possédant une période d'échantillonnage relativement petite. Ces remarques et inconvénients ont motivé un bon nombre de chercheurs, dans ce domaine, en proposant des améliorations sur la commande prédictive. Ces modifications ont pour objectif le remplacement du critère d'optimisation par un autre critère simple à mettre en oeuvre et qui serait capable d'interpréter les objectifs que l'on désire et les contraintes avec un degré de flexibilité spécifié. Dans ce contexte, une fonction coût floue construite peut être utilisée comme critère d'optimisation. Dans la littérature, plusieurs auteurs ont proposé l'utilisation de la commande prédictive floue [30][31][44]. Cette dernière repose sur l'exploitation de la théorie de la logique floue avec le principe de la commande prédictive, autant pour la modélisation des modèles de prédiction que pour les procédures et les méthodologies d'optimisation. Dans ce contexte, deux grandes familles de méthodes se dégagent.

La première famille de méthodes consiste à utiliser un modèle de prédiction analytique conventionnelle (sous forme d'une représentation d'état non linéaire ou une équation aux différences non linéaires) avec une fonction de coût floue. Dans cette thèse, cette méthode sera nommée la C.P.N.O.F (Commande Prédictive Non-linéaire avec Optimisation Floue). Dans ce cas, le problème d'optimisation peut être vu comme un problème de prise de décision floue multi-critères [30][44]. La simplicité d'utilisation de cette méthode, son implantation efficace et son intérêt dans l'exploitation des objectifs et des contraintes floues, en bénéficiant des connaissances humaines, font d'elle une méthode très intéressante. La deuxième famille de méthodes intègre l'utilisation d'un modèle de prédiction flou pour représenter la dynamique du système. En effet, en l'absence d'un modèle de prédiction analytique, cette dernière repose sur l'utilisation et l'identification, à partir des données entrées-sorties du système, d'un modèle flou de type Takagi-Sugeno ou Mamdani [25][45][46]. L'intérêt majeur de cette technique réside dans son utilisation pour des systèmes non linéaires incertains. Dans un premier temps, l'exploitation de la première méthode est détaillée. Par la suite, la méthodologie d'implantation de la deuxième technique sera donnée. Ce chapitre est organisé comme suit. Afin d'assurer une implantation efficace de la commande prédictive non linéaire, une stratégie d'optimisation floue est introduite dans le paragraphe 2. Cette dernière conduit à la Commande Prédictive Non-linéaire avec Optimisation Floue (C.P.N.L.O.F) où deux méthodologies d'optimisation sont détaillées et implantées (la programmation dynamique et la méthode Branch & Bound). Afin de contourner le problème d'une possible absence de modèles analytiques (le cas des systèmes complexes) une approche basée sur l'utilisation d'un modèle de prédiction flou identifié à partir des données en-

trées-sorties du procédé est utilisé. Celle-ci conduit à la Commande Prédictive Non linéaire Floue (C.P.N.L.F).

II.8. Commande Prédictive Non-linéaire avec Optimisation Floue (C.P.N.L.O.F)

II.8.1. Principe de la méthode

Le principe de la commande prédictive non linéaire avec une optimisation floue (C.P.N.O.F) est illustrée sur la Tableau 10, page 32. Le modèle de prédiction du système et supposé connu ou a été identifié sous la forme du modèle d'état analytique donné par l'équation (I.6).

Par rapport à la commande prédictive non linéaire conventionnelle, cette méthodologie se caractérise par l'utilisation d'un critère flou (fonction de coût floue) issue de l'expertise humaine sur le système. Dans ce contexte, des techniques d'optimisation de ce critère flou en se basant sur la théorie de décision floue sont nécessaires pour déterminer la séquence de commande optimale.

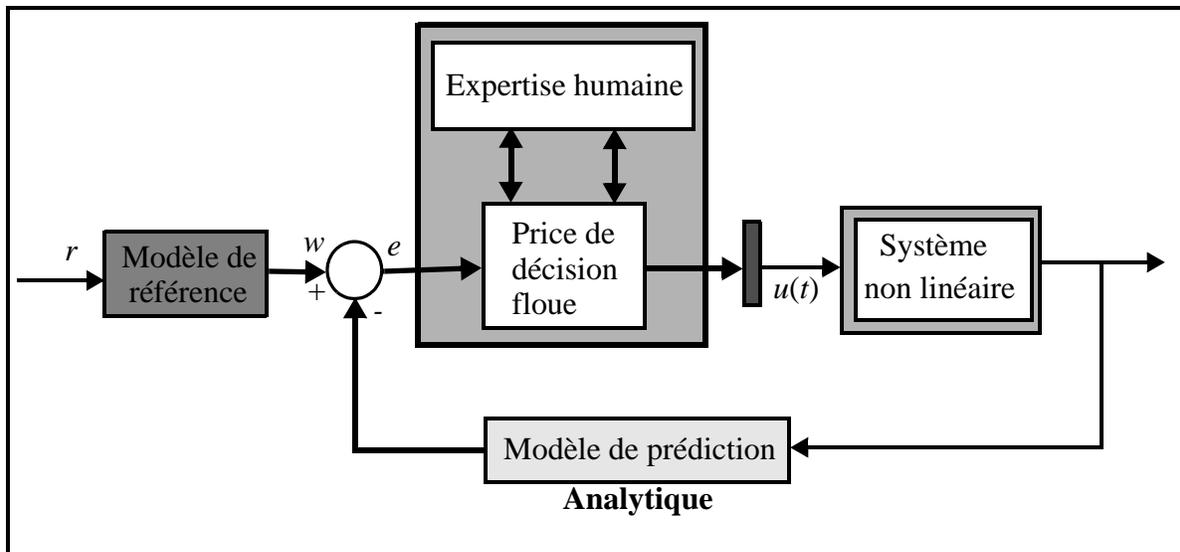


Figure II.10: Principe de la CPNA avec décision floue

II.8.2. Eléments de base de la prise de décision floue

Les deux éléments les plus importants dans la prise de décision floue sont:

- La définition de l'objectif de la décision ainsi que les contraintes imposées sur l'espace de recherche de la solution.
- La définition du mécanisme de prise de décision à utiliser.

D'une manière générale, dans la prise de décision floue, les valeurs numériques des objectifs et des contraintes sont remplacées par leurs versions floues représentées par des fonctions d'appartenance. Dans ce contexte, afin de pouvoir appréhender le mécanisme de décision floue, nous allons donner dans un premier temps quelques définitions nécessaires à la compréhension de ce formalisme. Ces définitions sont celles définies initialement par Bellman et Zadeh dans [2].

II.8.2.1. Définitions

- **Définition 1**

Soit A l'ensemble des décisions possibles contenant la solution du problème de la prise de décision floue considérée. Un objectif flou G est un ensemble flou de A caractérisé par sa fonction d'appartenance suivante:

$$\mu_G : \begin{cases} A \rightarrow [1,0] \\ a \rightarrow \mu_G(a) \end{cases} \quad (\text{II.64})$$

Pour des raisons de simplicité, dans notre travail, cette fonction d'appartenance est considérée comme triangulaire (cf. Tableau 11, page 33).

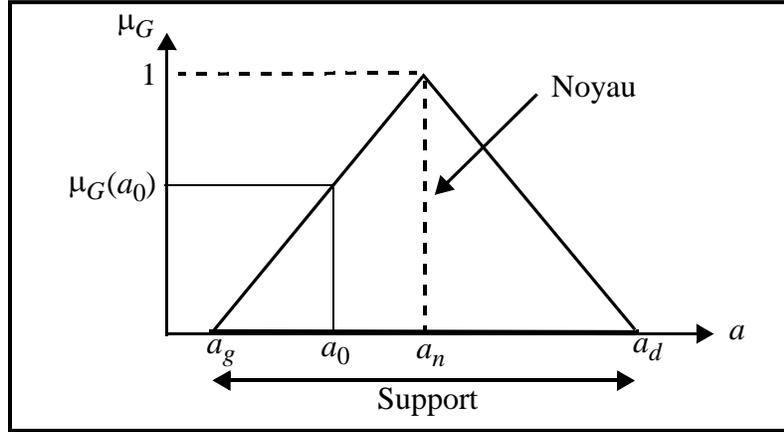


Figure II.11: Objectif flou triangulaire

Pour une valeur de a donnée, la fuzzification à travers cette fonction d'appartenance représente le degré pour lequel cette valeur doit satisfaire l'objectif de la décision spécifiée.

Dans la représentation de la Tableau 11, page 33, le support représente l'univers de discours ou l'intervalle de définition de l'objectif. A l'extérieur de ce dernier, la satisfaction du critère d'objectif est nul, c'est-à-dire:

$$\mu_G(a) = 0, \text{ si } a \notin [a_g, a_d] \quad (\text{II.65})$$

Le noyau représente la satisfaction totale de l'objectif, c'est-à-dire:

$$\mu_G(a) = 1, \text{ si } a = a_n \quad (\text{II.66})$$

A l'intérieur de l'intervalle $[a_g, a_d]$, le degré de satisfaction de l'objectif ne se situe pas d'une manière également probable en tout point. En effet, il est donné par le degré d'appartenance μ_G . Dans ce cas, pour une valeur $a_0 \in [a_g, a_d]$, son degré de satisfaction est donné par:

$$\begin{cases} \mu_G(a_0) = (a_0 - a_g)/(a_n - a_g), \text{ si } : a_0 \in [a_g, a_n] \\ \mu_G(a_0) = (a_d - a_0)/(a_d - a_n), \text{ si } : a_0 \in [a_n, a_d] \end{cases} \quad (\text{II.67})$$

- **Définition 2**

De la même manière que précédemment, une contrainte floue C est un ensemble flou de A caractérisé par sa fonction d'appartenance suivante:

$$\mu_C : \begin{cases} A \rightarrow [1,0] \\ a \rightarrow \mu_C(a) \end{cases} \quad (\text{II.68})$$

Pour une valeur a , la fuzzification à travers cette fonction d'appartenance représente le degré de

décision qui satisfait la contrainte. Comme illustré sur la Tableau 12, page 34, cette fonction d'appartenance est souvent considérée comme trapézoïdale.

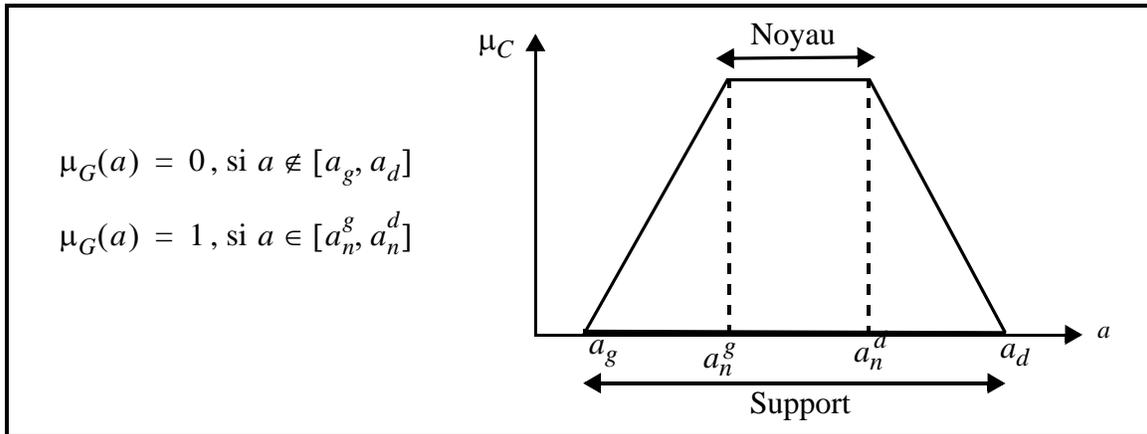


Figure II.12 Contrainte floue sous forme trapézoïdale

Les valeurs du noyau représentent une contrainte totale. Les valeurs du support donne le degré de respect de la contrainte (cf. Tableau 12, page 34). A l'extérieur de ce domaine, la contrainte est totalement violée.

• **Définition 3**

Considérons un objectif flou sous la forme Tableau 64, page 33 et une contrainte floue de la forme Tableau 68, page 33, selon le mécanisme de décision de Bellman et Zadeh [2], la décision floue D résultante est donnée par l'intersection de l'objectif G et de la contrainte C , c'est-à-dire:

$$D = G \cap C \tag{II.69}$$

En s'appuyant sur les fonctions d'appartenance définies précédemment, la décision floue sera caractérisée par sa fonction d'appartenance donnée par l'expression suivante:

$$\mu_D(a) = \mu_G(a) \wedge \mu_C(a), a \in A \tag{II.70}$$

ou \wedge indique l'opérateur minimum (cf. Figure II.4).

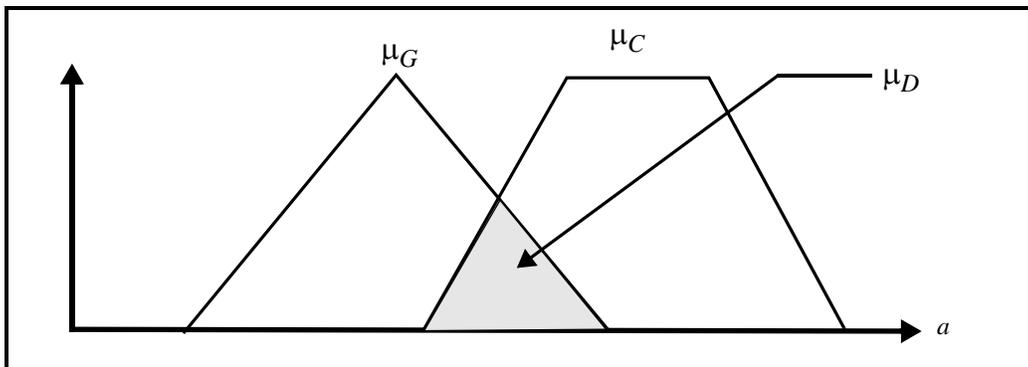


Figure II.13: Principe d'une intersection floue

• **Définition 4**

Dans le mécanisme de prise de décision floue, une décision optimale a^* fait référence à la décision avec le plus grand degré d'appartenance (meilleure décision). Cette décision optimale est aussi appelée maximisation de la décision. Sous forme mathématique cette dernière s'exprime

comme suit:

$$a^* = \arg \text{Max}_{a \in A} (\mu_G(a) \wedge \mu_C(a)) \quad (\text{II.71})$$

La figure Tableau 14, page 35 illustre ce principe de décision floue optimale.

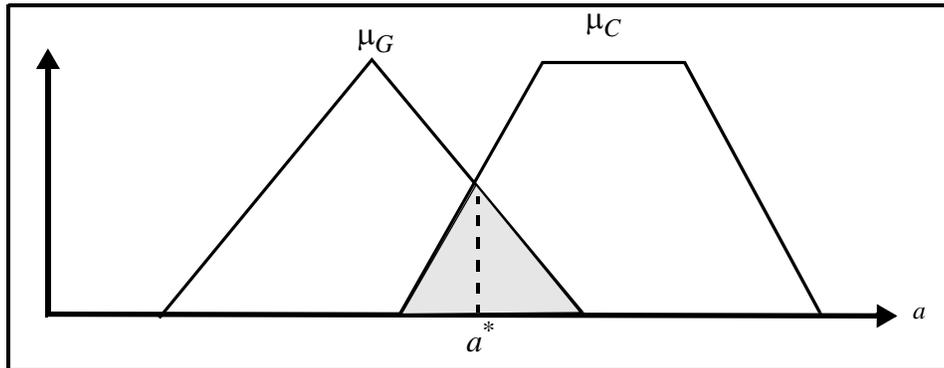


Figure II.14: Principe de la décision floue optimale

II.8.2.2. Principe de la fonction de coût floue

D'une manière conventionnelle, en commande prédictive la fonction du coût est représentée par l'association:

- d'un critère quadratique de l'erreur de suivi (dans un problème de suivi de trajectoire) ou de l'état (dans un problème de régulation ou de stabilisation).
- d'un critère d'énergie sur la commande.

Pour des raisons de simplicité et sans perte de généralité, nous allons nous intéresser uniquement à un problème de stabilisation, c'est-à-dire $x(k)$ est utilisée pour représenter l'état et $u(k)$ pour représenter la commande. Dans le cas d'un problème de suivie, le principe reste le même, la seule différence réside dans le remplacement de l'état $x(k)$ par l'erreur $e(k) = x(k) - x_d(k)$ entre l'état $x(t)$ et la trajectoire désirée $x_d(k)$.

Le principe de la fonction de coût floue repose sur:

- L'utilisation de deux fonctions d'appartenance floues pour décrire un objectif pour l'état $x(k)$ et une contrainte sur la commande $u(k)$. Soit μ_G et μ_C respectivement deux fonctions d'appartenance pour représenter $x(k)$ et $u(k)$. Les deux fonctions d'appartenance sont définies sur leurs univers de discours respectifs X et U .
- L'utilisation du principe de décision floue de Bellman et Zadeh [2] décrit précédemment pour déterminer la solution optimale réalisant l'objectif sous les contraintes spécifiées.

$$u^* = \arg \text{Max}_{u \in U} (\mu_G(x(k)) \wedge \mu_C(u(k))) \quad (\text{II.72})$$

Il est clair que l'utilisation de ce critère permet une flexibilité dans le problème l'optimisation. En effet, l'utilisateur peut modéliser, en fonction de son expertise sur le système, ses objectifs et ses contraintes à travers la définition des univers de discours X et U et les formes des fonctions d'appartenance les représentant.

II.8.2.3. Opérateurs de combinaison dans la décision floue

Bien que le principe de décision proposé initialement par Bellman et Zadeh [2], exploite l'opérateur «minimum» pour combiner l'objectif et les contraintes, d'autres types d'opérateurs peu-

vent être utilisés. Dans ce contexte, nous citons les opérateurs T -normes les plus connus ainsi que quelques opérateurs d'agrégation.

- **T-normes**

Une T -norme T est une application de $[0,1]^2 \rightarrow [0,1]$ possédant les propriétés suivantes :

1. $T(0, 0) = 0, T(x_1, 1) = T(1, x_1) = x_1$ (conditions aux limites)
2. $T(x_1, x_2) = T(x_2, x_1)$ (commutativité)
3. $x_1 \leq x_2, x_3 \leq x_4 \Rightarrow T(x_1, x_3) \leq T(x_2, x_4)$ (monotonie croissante)
4. $T(x_1, T(x_2, x_3)) = T(T(x_1, x_2), x_3)$ (associativité)

A titre d'exemple, nous rappelons ici quelques T -normes qui peuvent être utilisées à la place du minimum dans la formalisation du problème de décision floue.

T-norme : $T(x_1, x_2)$	Nom de la T-norme
$\min(x_1, x_2)$	Zadeh
$x_1 \cdot x_2$	Pobabiliste
$\max(0, x_1 + x_2 - 1)$	Lukasiewicz
$\max(0, 1 - [(1-x_1)^\gamma + (1-x_2)^\gamma]^{1/\gamma}), \gamma > 0$	Yager
$(x_1 \cdot x_2) / [\gamma + (1-\gamma)(x_1 + x_2 - a \cdot x_2)], \gamma > 0$	Hamacher
$\max(x_1 + x_2 - 1 + \gamma_1 \cdot x_1 x_2 / 1 + \gamma_1, 0), \gamma_1 >$	weber-Sugeno

Tableau 1: Définition de quelques T -normes populaires.

- **Opérateurs d'agrégation**

L'opérateur «minimum» dans le principe de Bellman-Zadeh peut aussi être remplacé par un opérateur d'agrégation. En effet, en présence des objectifs et/ou de contraintes antagonistes ou avec un effet de compensation, il est parfois insuffisant de se restreindre à des opérateurs sous forme de T -normes. En effet, d'autres opérateurs d'agrégation peuvent être utilisées. Dans ce cas, on peut citer:

1. Les opérateurs de compromis: dans cette classe on trouve les opérateurs de type moyenne et somme pondérées. Parmi ces opérateurs, on trouve le minimum et le maximum qui apparaissent comme des cas limites de cette famille.
2. Les opérateurs de compensation: ces opérateurs permettent de se trouver entre le mode conjonctif et le mode disjonctif, cela par l'intermédiaire du choix d'un paramètre positif qui a des valeurs entre 0 et 1.
3. Les opérateurs OWA (Ordered Weighted Averaging): l'opérateur OWA et ses versions permet de considérer en priorité les points fort ou faibles d'un critère (introduction de la notion de pondération et d'ordre entre les critères).
4. Intégrales Floues: les définitions d'une intégrale floue sont liées à la définition d'une mesure floue. La difficulté majeure pour une utilisation pratique est la définition de la mesure floue as-

sociée, pour laquelle on est souvent obligé de faire appel à des méthodes numériques assez lourdes. Dans cette classe, on peut citer les intégrales de Choquet et de Sugeno.

Pour plus de détail sur les opérateurs d'agrégation et leurs utilisations, le lecteur est invité à consulter les références du domaine [19].

II.8.3. Méthodologie d'implantation

Etant donné un système non linéaire sous la forme (I.6).

L'objectif de ce paragraphe est de montrer comment la méthodologie de commande prédictive non linéaire avec optimisation floue (C.P.N.O.P) sera appliquée afin d'atteindre l'objectif principale désiré, c'est-à-dire la stabilisation de l'état $x(k)$.

Il est important d'insister ici sur le fait que la méthodologie de la prise de décision floue sera seulement appliquée aux systèmes déterministes avec un temps final fini et spécifié. Ce dernier correspond à l'horizon de prédiction N_2 de la commande prédictive.

Pour des raisons de simplicité et de clarté dans les notations, nous allons supposer que $N_2 = N_u$ et les états prédites $x(k+1/k), \dots, x(k+N_2/k)$ aient le même objectif flou donné par une fonction d'appartenance notée μ_G et définie sur son univers de discours X . De la même manière, toute la séquence de commande prédite $u(k/k), \dots, u(k+N_2-1/k)$ est contrainte par la même fonction d'appartenance notée μ_C et définie sur son univers de discours U . La généralisation au cas multi-contraintes sera donnée ultérieurement.

A chaque instant k , toute variable de commande $u(k+i/k), i = 1 \dots N_2-1$ est soumise à la contrainte floue donnée par :

$$\mu_C(u(k+i/k)), \text{ pour } i = 0 \dots N_2-1 \quad (\text{II.73})$$

De la même manière, chaque état $x(k+i/k), i = 1, \dots, N_2$ a pour objectif la fonction d'appartenance donnée par:

$$\mu_G(x(k+i/k)), i = 1, \dots, N_2 \quad (\text{II.74})$$

Dans ce cas, la décision D portant sur la séquence de commande $u(k/k), \dots, u(k+N_2-1/k)$ est donnée par la fonction d'appartenance résultante de la combinaison entre les objectifs sur les états et les contraintes sur les commandes, c'est-à-dire:

$$\mu_D(u(k/k), \dots, u(k+N_2-1/k)) = \Lambda_{i=0}^{N_2-1} [\mu_C(u(k+i/k)) \Lambda \mu_G(x(k+i+1/k))] \quad (\text{II.75})$$

L'optimisation floue vise à déterminer la séquence de commande optimale $u^*(k/k), \dots, u^*(k+N_2-1/k)$ optimisant le critère de décision flou suivant:

$$\begin{aligned} \mu_D(u^*(k/k), \dots, u^*(k+N_2-1/k)) = \\ \text{Max}_{u(k/k), \dots, u(k+N_2-1/k)} \Lambda_{i=0}^{N_2-1} [\mu_C(u(k+i/k)) \Lambda \mu_G(x(k+i+1/k))] \end{aligned} \quad (\text{II.76})$$

Partant de l'état final $x(k+N_2/k)$, la séquence optimale $u^*(k/k), \dots, u^*(k+N_2-1/k)$ sera obtenue en optimisant le critère suivant:

$$\begin{aligned} & \mu_D(u^*(k/k), \dots, u^*(k + N_2 - 1/k)) \\ = & \text{Max}_{u(k/k), \dots, u(k + N_2 - 1/k)} \left(\begin{array}{c} \mu_C(u(k/k)) \wedge \mu_G(x(k + 1/k)) \wedge \dots \wedge \mu_C(u(k + N_2 - 1/k)) \\ \wedge \mu_G(x(k + N_2/k)) \end{array} \right) \quad (\text{II.77}) \end{aligned}$$

On remarque alors que les deux derniers termes :

$$\mu_C(u(k + N_2 - 1/k)) \wedge \mu_G(x(k + N_2/k)) \quad (\text{II.78})$$

dépendent uniquement de $u(k + N_2 - 1/k)$ et non pas des commandes précédentes $u(k + N_2 - 2/k), \dots, u(k/k)$. Dans ce cas le problème d'optimisation peut être décomposé en deux parties, la première dépendante des commandes $u(k + N_2 - 2/k), \dots, u(k/k)$ et la deuxième dépendante uniquement de $u(N_2 - 1)$, c'est-à-dire:

$$\begin{aligned} & \mu_D(u^*(k/k), \dots, u^*(k + N_2 - 1/k)) = \\ & \text{Max}_{u(k/k), \dots, u(k + N_2 - 2/k)} \mu_C(u(k/k)) \wedge \dots \wedge \mu_C(u(k + N_2 - 2/k)) \wedge \\ & [\text{Max}_{u(k + N_2 - 1/k)} \mu_C(u(k + N_2 - 1/k)) \wedge \mu_G(f(x(k + N_2 - 1/k), u(k + N_2 - 1/k)))] \quad (\text{II.79}) \end{aligned}$$

En répétant cette opération sur le terme $\mu_C(u(k + N_2 - 1/k))$, qui n'est dépendant que de la commande $u(k + N_2 - 2/k)$ on obtient:

$$\begin{aligned} & \mu_D(u^*(k/k), \dots, u^*(k + N_2 - 1/k)) = \\ & \text{Max}_{u(k/k), \dots, u(k + N_2 - 3/k)} \mu_C(u(k/k)) \wedge \dots \wedge \mu_C(u(k + N_2 - 3/k)) \wedge \\ & \text{Max}_{u(N_2 - 2)} \{ [\mu_C(u(N_2 - 2/k)) \wedge \\ & [\text{Max}_{u(k + N_2 - 1/k)} \mu_C(u(k + N_2 - 1/k)) \wedge \mu_G(f(x(k + N_2 - 1/k), u(k + N_2 - 1/k)))]] \} \quad (\text{II.80}) \end{aligned}$$

L'application de cette méthodologie pour toutes les autres commandes de $u(N_2 - 3)$ jusqu'à $u(0)$ conduit aux équations récurrentes de la programmation dynamique suivantes:

$$\left\{ \begin{array}{l} \mu_G(x(k + N_2 - i/k)) = \text{Max}_{u(k + N_2 - i/k)} \mu_C(u(k + N_2 - i/k)) \wedge \mu_G(x(k + N_2 - i + 1/k)) \\ x(k + N_2 - i + 1/k) = f(x(k + N_2 - i/k), u(k + N_2 - i/k)) \\ i = 1, \dots, N_2 \end{array} \right. \quad (\text{II.81})$$

où le terme $\mu_G(x(k + N_2 - i/k))$ peut être vue comme un objectif due à la commande à l'instant $k + N_2 - i$ induit par l'objectif flou à l'instant $k + N_2 - i + 1$.

II.8.4. Généralisation au cas multi-contraintes

Le principe de la Programmation Dynamique Floue (PDF) peut être facilement généralisé aux problèmes d'optimisation multi-objectifs et multi-contraintes.

si on suppose que:

- A chaque état du système, $x(k/k), \dots, x(k + N_2/k)$ on associe un objectif défini par une fonction d'appartenance $\mu_{G^i}, i = 1, \dots, N_2$.
- A chaque commande $u(k/k), \dots, u(k + N_2 - 1/k)$ on associe une contrainte définie par sa fonction d'appartenance $\mu_{C^i}, i = 0, \dots, N_2 - 1$.

La formulation du problème d'optimisation reste la même avec le remplacement des fonctions d'appartenances μ_G et μ_C par des fonctions d'appartenance μ_{G^i} et μ_{C^i} indexées sur un état et une commande donnée.

Sous forme mathématique, les équations récurrentes de la P.D.F deviennent les suivantes:

$$\left\{ \begin{array}{l} \mu_{G_i}(x(k + N_2 - i/k)) = \\ \text{Max}_{u(k + N_2 - j/k)} \mu_{C_j}(u(k + N_2 - j/k)) \wedge \mu_{G_i}(x(k + N_2 - i + 1/k)) \\ \text{avec } x(k + N_2 - i + 1/k) = f(x(k + N_2 - i/k), u(k + N_2 - i/k)) \end{array} \right. \quad (\text{II.82})$$

Dans la littérature plusieurs méthodes ont été proposées pour résoudre le problème d'optimisation. Sans avoir la prétention d'être exhaustif, nous nous intéressons à celles les plus rencontrées dans la littérature, à savoir la Programmation Dynamique (DP) et la technique de Branch & Bound.

II.8.5. Résolution par la Programmation dynamique

A. Principe de la méthode

La programmation dynamique «P.D» est une méthode d'optimisation développée par Bellman dans les années 50. Cette dernière est un outil de calcul intéressant pour la résolution des problèmes d'optimisation dynamiques. Son avantage majeur réside dans sa capacité de manipuler des systèmes et des critères de performances généraux (linéaire, non linéaire, non quadratique). Contrairement aux méthodes itératives, la P.D assure la convergence de la solution optimale vers un optimum global et non local. De plus, elle offre une simplicité dans l'implantation et dans le traitement des contraintes.

Une version de la P.D floue où la fonction de coût standard a été remplacée par une fonction floue a été proposée dans [2][17]. Cette méthodologie a conduit à la naissance de la programmation dynamique floue (P.D.F) .

L'implantation de la P.D.F repose sur deux étapes:

- La première étape consiste en la génération d'une liste (tableau) de décision partant de l'état initial jusqu'à l'état final sur l'horizon de sortie choisie.
- La deuxième étape concerne le parcours de cette liste, en partant de l'état final jusqu'à l'état initial, afin de trouver la solution optimale (chemin optimal à travers cette liste) au sens du critère flou utilisé en appliquant le principe d'optimalité de Bellman.

• Phase 1: Construction du tableau de décision

La liste (tableau) de décision est construite de l'état initial $x(k/k)$ jusqu'à l'état final $x(k+N_2/k)$.

Supposons que le domaine de définition de l'état x est donné par l'univers de discours $X = [x_{min}, x_{max}]$.

L'état x est discrétisée en $m = (x_{max} - x_{min})/\Delta_x$ valeurs d'états possibles selon l'ensemble suivant:

$$\underline{x} = \{x_j = x_{min} + j \cdot \Delta_x, j = 1, \dots, m\} \quad (\text{II.83})$$

où Δ_x représente le pas de discrétisation.

Supposons aussi que le domaine de définition de la variable de commande u est donné par l'univers de discours $U = [u_{min}, u_{max}]$.

La variable de commande u est discrétisée en $n = (u_{max} - u_{min}) / \Delta_u$ valeurs d'entrées possibles selon l'ensemble suivant:

$$\underline{u} = \{u_l = u_{min} + l \cdot \Delta_u, l = 1, \dots, n\} \quad (\text{II.84})$$

où Δ_u représente le pas de discrétisation.

Au premier niveau $i = 0$, chaque commande u_l de \underline{u} sera appliquée au système en partant de l'état initial $x(0) = x(0/k)$ pour l'amener à son état $x_j(1)$ qui appartient à l'ensemble \underline{x} . De la même manière, en partant de ce dernier état du niveau 1 chaque commande u_l lui sera appliquée pour le conduire à son état suivant. Ce processus va se poursuivre jusqu'au niveau final $x_j(N_2)$. Dans ce cas, la liste de décision est totalement construite (cf. Tableau 15, page 40).

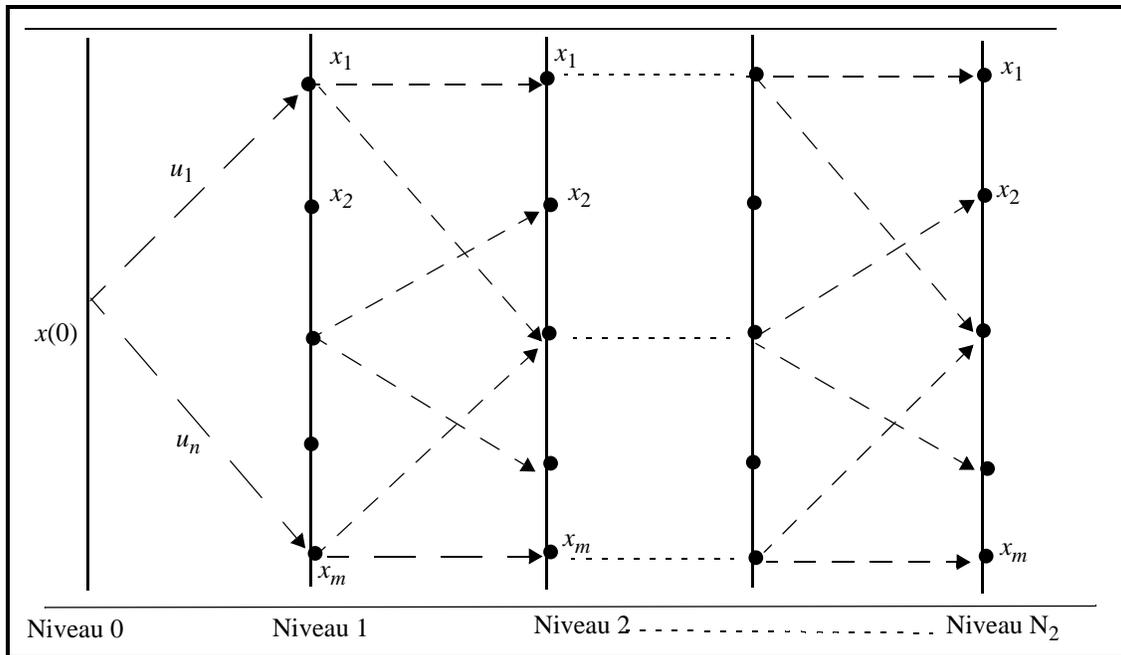


Figure II.15: Principe de la génération d'un tableau de décision pour la P.D.F

• **Phase 2 : Exploration du tableau à la recherche d'un chemin optimal**

Partant du niveau N_2-1 , nous cherchons la branche optimale pour chaque état, en optimisant le critère suivant:

$$\mu_D(x_j(k + N_2 - 1/k)) = \text{Max}_{\underline{u}} \{ \mu_G(f(x_j(k + N_2 - 1/k), \underline{u})) \wedge \mu_C(\underline{u}) \} \wedge \{ \mu_D(x_j(k + N_2/k)) \} \quad (\text{II.85})$$

avec $\mu_D(x_j(k + N_2/k)) = \mu_D(f(x_j(k + N_2 - 1/k), \underline{u})) = 1, j = 1, \dots, m$.

En répétant cette opération (procédure) pour tous les états de tous les niveaux et ce à partir du niveau N_2-2 jusqu'au niveau 0. Finalement, la solution optimale correspond à la branche optimale au niveau zéro (racine de la liste).

Afin de réaliser notre objectif, nous allons exploiter l'approche P.D.F pour implanter la commande prédictive avec un critère de décision flou. Dans ce contexte, l'implantation est faite avec le choix de ces paramètres:

- un horizon de prédiction N_2 .
- un horizon de commande $N_u=N_2$.
- les pas de discrétisation de l'état et de la commande sont fixés respectivement à Δ_x et Δ_u .

les fonctions d'appartenance sont choisies en fonction de notre expertise sur le système, associées à l'état et à la commande. Ces dernières sont en général triangulaire et trapézoïdale définies sur leur univers de discours X et U (cf. Figure II.16).

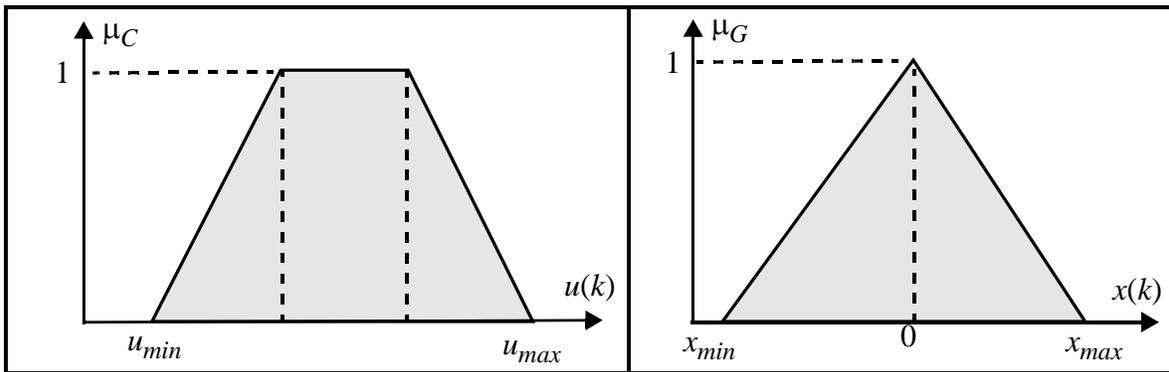


Figure II.16 Objectif flou et contrainte floue

L'algorithme à implanter dans ce cas est donné par:

Initialisation

Entrées

Horizon de prédiction N_2

Horizon de commande N_u

Remarque:

on prend l'horizon de commande égale à l'horizon de prédiction $N_u=N_2$.

Pas de discrétisation Δ_x, Δ_u .

Taille de séquence de commande $n = (u_{max}-u_{min})/\Delta_u$

Taille de l'ensemble d'état $m = (x_{max}-x_{min})/\Delta_x$

Condition initiale $x(0)=x(0/k)$

Début

• Phase 1 : Génération de la liste de décision

```

Pour  $i = 1 \dots N_2$            /* pour chaque étape */
  Pour  $j = 1 \dots m$          /* pour chaque état */
    Pour  $l = 1 \dots n$        /* pour chaque commande */
       $u_l = u_{min} + (l-1)*\Delta_u$ 
  
```

$$x_j^{(l)}(i) = f(x_j(i-1), u_l) \in \underline{x}$$

Fin pour
Fin pour
Fin pour

• **Phase 2 : Exploration du tableau à la recherche d'un chemin optimal**

Pour $i = N-1 \dots 0$ /* pour chaque étape */
Pour $j = 1 \dots m$ /* pour chaque état */

$$\mu_D(x_j(i)) = \text{Max}_{\underline{u}} \{ \mu_G(f(x_j(i), \underline{u})) \wedge \mu_C(\underline{u}) \} \wedge \{ \mu_D(f(x_j(i), \underline{u})) \}$$

$$u^*(j, i) = \arg [\text{Max}_{\underline{u}} \{ \mu_G(f(x_j(i), \underline{u})) \wedge \mu_C(\underline{u}) \} \wedge \{ \mu_D(f(x_j(i), \underline{u})) \}]$$

Fin pour
Fin pour

on prend $\mu_D(f(x_j(N+1))) = 1$.

II.8.6. Solution par la méthode de Branch & Bound

A. Principe de la méthode

La méthode Branch & Bound est associée à une arborescence dont les sommets correspondent à des solutions réalisables. Elle repose sur deux principes:

• **Détermination du chemin optimal initial**

De la même manière que la PDF, on suppose que l'ensemble \underline{u} des commandes discrétisées sous la forme (II.84) est disponible, par contre l'état ne sera pas discrétisé.

Dans ce cas, toutes les commandes u_l de \underline{u} sont appliquées au système en partant de l'état initial $x(k)=x(k/k)$. A ce niveau, afin de déterminer la meilleure branche initiale partant de l'état $x(k)$ du niveau 0 à l'état suivant $x_l(k+1)$, le critère suivant est minimisé:

$$\mu_D(x_l(k+1/k)) = T(\mu_G(f(x(k/k), u_l)), \mu_C(u_l)) \tag{II.86}$$

où T représente une T -norme (le minimum dans notre cas)
Autrement dit, l'objectif est de déterminer la meilleure branche (valeur de \underline{u}) telle que:

$$\text{Max}_{\underline{u}} \mu_D(x_l(k+1/k)) \tag{II.87}$$

Pour des raisons d'implantation efficace, le problème de maximisation du critère précédent est transformé en un problème de minimisation en prenant le complément flou du degré d'appartenance du critère considéré, c'est-à-dire:

$$\mu_{D_c}(x_l(k+1/k)) = \overline{\mu_D(x_l(k+1/k))} \tag{II.88}$$

c'est-à-dire:

$$\text{Max}_{\underline{u}} \mu_D(x_i(k+1/k)) = \overline{\text{Min}_{\underline{u}} \mu_D(x_i(k+1/k))} = \text{Min}_{\underline{u}} \mu_{D_c}(x_i(k+1/k)) \quad (\text{II.89})$$

Dans ce cas, la meilleur commande u_i^* de \underline{u} associée à la branche donnant la meilleure valeur du critère sera sélectionnée.

Par la suite, cette valeur de commande déterminée sera également appliquée au système en partant de l'état $x(0)$ pour l'amener à son état $x^*(1)$. De la même manière, un critère est minimisé pour sélectionner la deuxième meilleure branche à partir de $x^*(1)$ (la meilleure commande de \underline{u}):

$$\text{Min}_{\underline{u}} \mu_{D_c}(x(k+2/k)) = \overline{\text{Min}_{\underline{u}} T(\mu_D(x^*(k+1/k)), (\mu_G(f(x^*(k+1/k), \underline{u})), \mu_C(\underline{u})))} \quad (\text{II.90})$$

Dans ce cas, la deuxième branche est sélectionnée.

Ce principe de sélection va se poursuivre (cf. Figure II.17), jusqu'à l'horizon de prédiction N_2 . A chaque niveau le critère suivant est utilisé pour déterminer la meilleure branche:

$$\begin{aligned} \text{Min}_{\underline{u}} \mu_{D_c}(x(k+i/k)) = \\ \overline{\text{Min}_{\underline{u}} T(\mu_D(x^*(k+i-1/k)), \mu_G(f(x^*(k+i-1/k), \underline{u})), \mu_C(\underline{u}))} \end{aligned} \quad (\text{II.91})$$

Une fois arrivé au niveau de l'horizon de prédiction, c'est-à-dire $i = N_2$, toutes les autres branches associées à la même famille de la branche optimale seront bloquées (cf. Tableau 17, page 44). Dans ce cas, le chemin optimal initial a été calculé.

Maintenant, nous allons déterminer la borne supérieure initiale de ce chemin qui peut servir à comparer le coût des autres branches. En effet, cette borne est donnée par:

$$\mu_{D_c}^U = \mu_{D_c}(x^*(k+N_2/k)) \quad (\text{II.92})$$

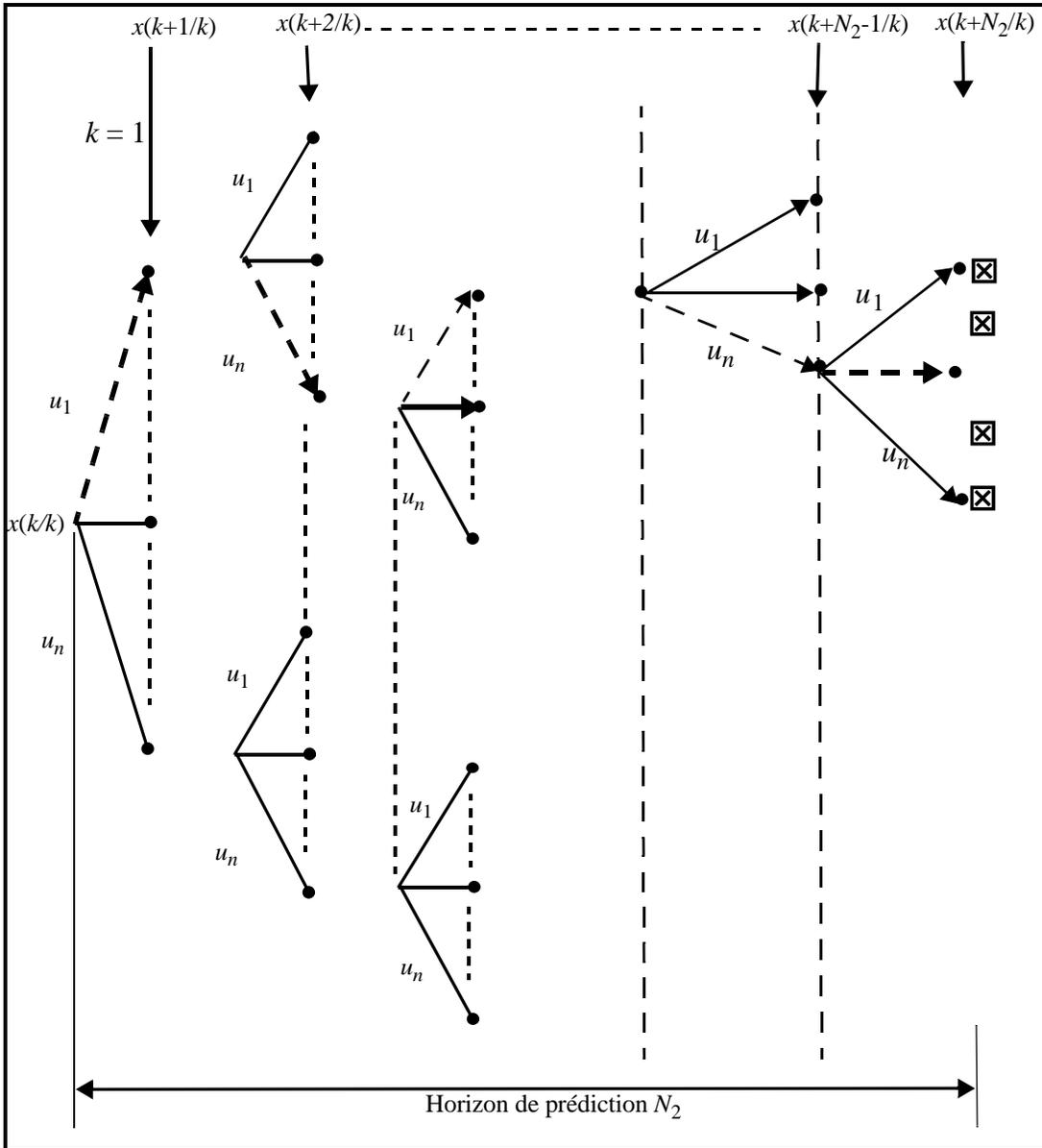


Figure II.17: Principe de Branch & Bound (B&B) pour $N_u=N_2$

- **Exploration de l'arbre à la recherche du chemin optimal**

L'objectif de cette exploration est de rechercher à travers l'arbre s'il reste encore des chemins non explorés pouvant conduire à l'obtention d'un chemin plus optimal que le chemin optimal initial.

Pour ce faire, sachant que toutes les branches non optimales au niveau de l'horizon de commande ($i = N_2$) ont été bloquées, il nous reste à explorer les branches en remontant l'arbre des niveaux $i = N_2 - 1$ jusqu'à $i = 1$. Dans ce cas, à chaque niveau il est nécessaire de déterminer une borne inférieure estimée servant à donner une indication, à priori, sur le coût du critère ainsi qu'une condition de branchement primordiale pour pouvoir éliminer les branches non optimales.

A. Estimation de la borne inférieure

A un niveau i , l'application de la commande amène le système à son niveau $i+1$. Dans ce cas, il serait intéressant de pouvoir estimer la valeur du critère d'optimisation de $i+1$ jusqu'à l'horizon

de prédiction N_2 . Cette estimation a pour objectif d'éliminer les branches qui sont supérieures à cette estimation.

Si cette estimation μ_{Dest} est disponible, alors la valeur inférieure sera donnée par:

$$\mu_L(x(k+i/k)) = \overline{T(\mu_D(x(k+i/k)), \mu_{Dest}(x(k+i+1/k)))} = \perp (\overline{\mu_D(x(k+i/k))}, \overline{\mu_{Dest}(x(k+i+1/k))}) \quad (\text{II.93})$$

où \perp est une T -conorme.

B. Condition de branchement

Après avoir déterminé le chemin optimal initial et sa borne supérieure initiale, nous allons déterminer les autres chemins (branches) qui peuvent donner un critère inférieur à la borne supérieure.

Pour la sélection de ces branches, la condition de branchement de l'algorithme de Branch & Bound est la suivante:

$$\overline{T(\mu_D(x(k+i/k)), (\mu_G(f(x(k+i/k), u_l), \mu_C(u_l))))} < \mu_{D_c}^U \quad (\text{II.94})$$

Autrement dit, on va explorer toutes les branches et vérifier s'il existe un autre chemin plus optimal que la borne supérieure initiale décrivant le chemin optimal initial. Dans le cas où ce chemin existe, lui même devient optimal et la borne supérieure sera adaptée avec le critère associé en accord avec le nouveau chemin optimal trouvé. Dans le cas contraire, la borne supérieure reste inchangé.

Si cette condition n'est pas vérifiée, le chemin de i à $i+1$ sera bloqué (cf. Tableau 18, page 45). En pratique, si aucune information n'est disponible sur l'estimation de la borne inférieure entre $i+1$ et N_2 , cette dernière sera prise égale à 1.

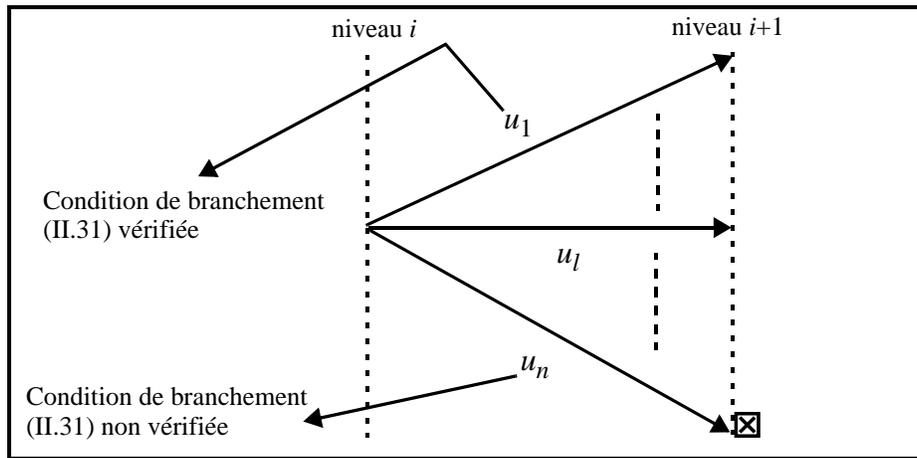


Figure II.18: Condition de branchement.

L'algorithme s'arrêtera quand il ne restera aucune branche à explorer et qu'à chaque niveau il ne restera qu'un seul noeud.

Il est important d'attirer l'attention sur le fait que l'horizon de prédiction est souvent plus large que l'horizon de commande ($N_2 > N_u$). Dans ce cas, dans la formalisation du principe d'optimisation les commandes restent constantes après l'horizon de commande, N_u c'est-à-dire:

$$u(k+N_u-1/k) = u(k+N_u/k) = \dots = u(k+N_2-1/k) \quad (\text{II.95})$$

et le principe de la méthode Branch & Bound dans ce cas sera illustré par (cf. Figure II.10)

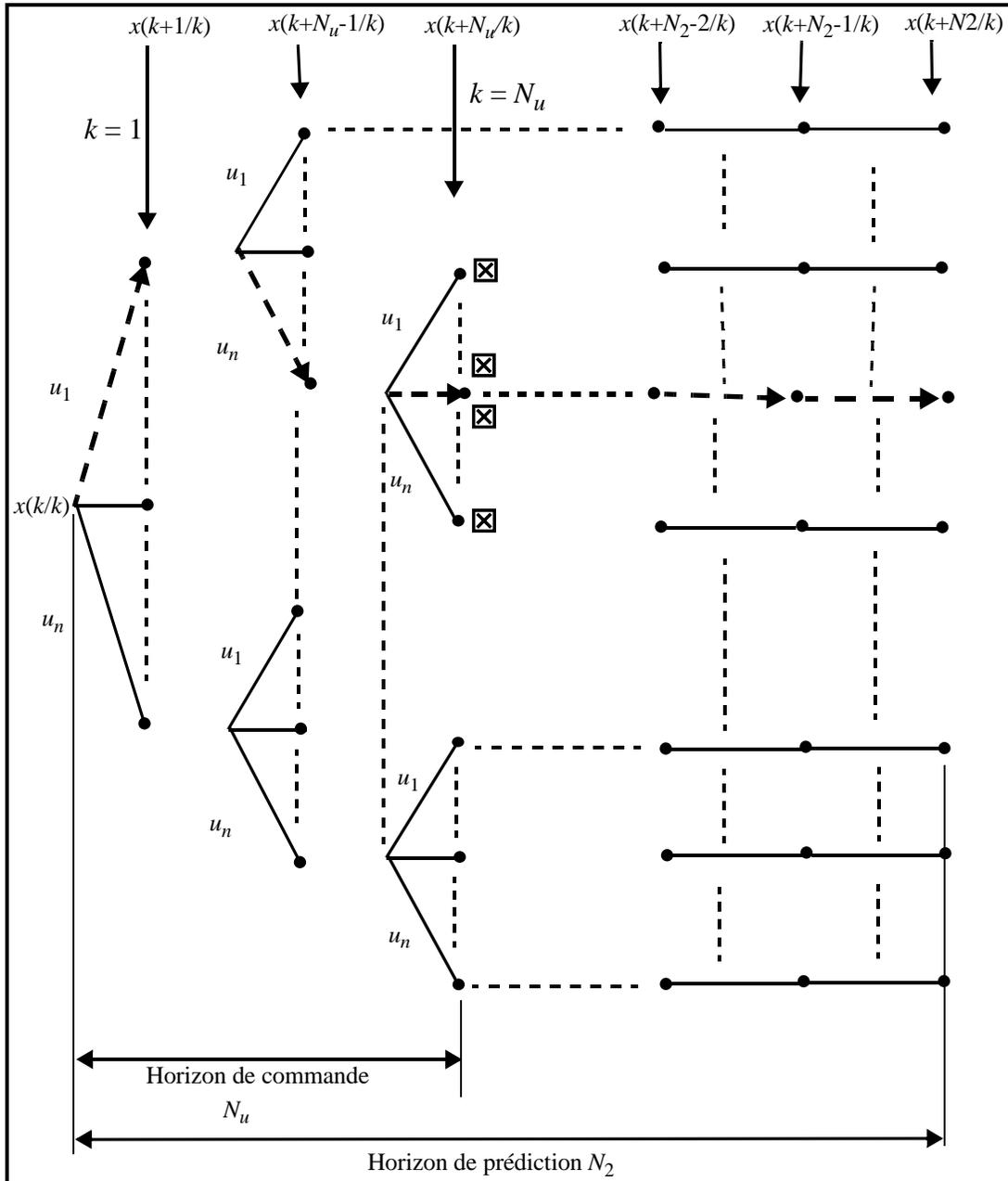


Figure II.19: Principe de l'algorithme B&B pour $N_u < N_2$

II.8.7. Performances de la C.P.N.O.F

II.8.7.1. Calculs associés

Les deux méthodes d'optimisation présentées précédemment se basent sur l'utilisation d'un espace de commande discrétisé. Dans ce cas, ces dernières aboutissent à des solutions approximatives du problème d'optimisation total (choix du pas de discrétisation). Pour remédier à ce problème, il est possible d'utiliser une discrétisation fine (un pas très faible). Si cette façon de procéder va nous permettre une meilleure approximation de la solution optimale, elle va conduire à un calcul rigoureux et très volumineux. Pour une implantation efficace, un compromis doit être trouvé.

Contrairement à la PDF, la méthode branch & bound a introduit les notions des bornes estimées afin de pouvoir réduire le nombre de noeuds produits dans l'arbre, et ainsi la complexité des calculs de l'algorithme. Si cette idée reste exploitable sur un horizon de commande réduit, son déploiement sur un grand horizon de commande reste difficile (complexité exponentielle). Dans le but d'éliminer ces inconvénients, Sousa et al. ont proposé [42] l'utilisation d'un ensemble de décisions adaptées, avec un mécanisme de filtrage. Pour plus de détails sur l'implantation de cette technique, le lecteur est invité à consulter les références [42][43].

II.8.7.2. Stratégie de décision

Le mécanisme de décision et les opérateurs utilisés peuvent avoir une influence sur la dynamique des systèmes commandés. En effet les T -normes et les opérateurs d'agrégation cités précédemment peuvent être exploités. Dans ce cas, si la méthode d'optimisation a été choisie, il est possible d'utiliser l'opérateur de combinaison et/ou d'agrégation comme paramètre de réglage pour les performances du système en boucle fermée.

Pour illustrer ce concept, nous Prenons à titre d'exemple la T -norme de Yager.

Supposons qu'on dispose de deux informations qu'on désire prendre en considération dans notre critère de décision floue:

- L'erreur de suivi entre la sortie du système et la trajectoire de référence $w(t)$, c'est-à-dire: $\hat{e}(k+i) = w(k+i) - \hat{y}(k+i)$. Une fonction d'appartenance $\mu_e(\hat{e}(k+i))$ est utilisée pour représenter l'objectif sur cette erreur.
- Le changement de la variable de commande défini par $\Delta u(k+i) = u(k+i) - u(k+i-1)$. Sa fonction d'appartenance représentant la contrainte sur la commande est notée $\mu_u(\Delta u(k+i))$.

Pour deux variables x_1 et x_2 , la T -norme de Yager s'exprime comme suit:

$$T_{yager} = \max(0, 1 - [(1 - x_1)^\gamma + (1 - x_2)^\gamma]^{1/\gamma}) \quad (\text{II.96})$$

ou γ est un paramètre de réglage.

Par substitution de x_1 et x_2 par les fonctions d'appartenance des objectifs et de la contrainte, c'est-à-dire:

$$\begin{cases} x_1 = \mu_e(e(k+i)) \\ x_2 = \mu_u(\Delta u(k+i)) \end{cases} \quad (\text{II.97})$$

l'expression de la T -norme de Yager sera donnée par:

$$T_{yager} = \max(0, 1 - [(1 - \mu_e(e(k+i)))^\gamma + (1 - \mu_u(\Delta u(k+i)))^\gamma]^{1/\gamma}) \quad (\text{II.98})$$

Sachant que:

$$\begin{aligned} 1 - \mu_e(e(k+i)) &= \overline{\mu_e(e(k+i))} \\ 1 - \mu_u(\Delta u(k+i)) &= \overline{\mu_u(\Delta u(k+i))} \end{aligned} \quad (\text{II.99})$$

Dans ce cas, l'équation (II.99) sera donnée par:

$$T_{Yager} = \max(0, 1 - [(\overline{\mu_e(e(k+i))})^\gamma + (\overline{\mu_u(\Delta u(k+i))})^\gamma]^{1/\gamma}) \quad (\text{II.100})$$

Il est clair que:

$$\begin{aligned} \max(0, 1 - [(\overline{\mu_e(e(k+i))})^\gamma + (\overline{\mu_u(\Delta u(k+i))})^\gamma]^{1/\gamma}) &= \\ \min(1, [(\overline{\mu_e(e(k+i))})^\gamma + (\overline{\mu_u(\Delta u(k+i))})^\gamma]^{1/\gamma}) &= \\ [(\overline{\mu_e(e(k+i))})^\gamma + (\overline{\mu_u(\Delta u(k+i))})^\gamma]^{1/\gamma} & \end{aligned} \quad (\text{II.101})$$

Sur un horizon N_2 spécifié, le critère à optimiser s'écrit comme suit:

$$J(u) = (\sum_{i=1}^{N_2} (\overline{\mu_e(e(k+i))})^\gamma + \sum_{i=1}^{N_u} (\overline{\mu_u(\Delta u(k+i))})^\gamma)^{1/\gamma}, \gamma > 0 \quad (\text{II.102})$$

Il s'avère donc qu'à travers cette T norme de yager l'objectif flou peut être interprété par une minimisation d'un terme qui contient l'erreur de prédiction et un autre terme qui contient le changement de la variable de commande (semblable à un critère quadratique).

Le paramètre γ peut être utilisé comme paramètre de réglage dans le problème d'optimisation.

II.9. Commande Prédictive Non linéaire Floue (C.P.N.F)

II.9.1. Principe et implantation de la méthode

Le principe de la commande prédictive non linéaire floue repose sur l'utilisation d'un modèle flou. En effet, afin de contourner les problèmes d'optimisation dus à l'utilisation de méthodes itératives, on propose d'exploiter une représentation floue du modèle non linéaire (le remplacement du modèle non linéaire analytique par un modèle flou) puis l'application du principe de la commande prédictive. Le principe de la C.P.N.F est illustré sur la figure suivante.

Supposons un système non linéaire mono entrée/mono sortie (SISO) représenté par un l'équation dynamique (I.6).

L'utilisation d'un modèle d'état de la forme (I.6), dans un cadre flou, nécessite l'approximation d'un nombre important de fonctions non linéaires, les n_y fonctions non linéaires f_1, \dots, f_{n_y} du champ de vecteurs f et la fonction non linéaire h , d'où une grande complexité. Ainsi, il est très difficile de pouvoir maîtriser la dynamique de (n_y+1) familles de paramètres relatives à (n_y+1) fonctions non linéaires. De plus, le modèle d'état (I.6) ne présentant pas une relation directe entre la commande et la sortie, la synthèse d'une loi de commande n'en est que plus compliquée, tout au moins dans un cadre flou.

L'ensemble de ces remarques nous conduit à transformer la représentation d'état en une représentation entrée-sortie de façon à exhiber une relation explicite entre la commande et la sortie.

C'est cette représentation qui sera approximée par un système flou. En effet, pour exprimer explicitement la relation liant entrée et sortie du système, la sortie y est évaluée à des instants d'échantillonnage successifs jusqu'à apparition de l'entrée u .

On obtient alors:

$$\left\{ \begin{array}{l} y(k) = h(x(k)) \\ y(k+1) = hof(x(k)) \\ \dots\dots\dots \\ y(k+r-1) = hof^{(r-1)}(x(k)) \end{array} \right. \quad (\text{II.103})$$

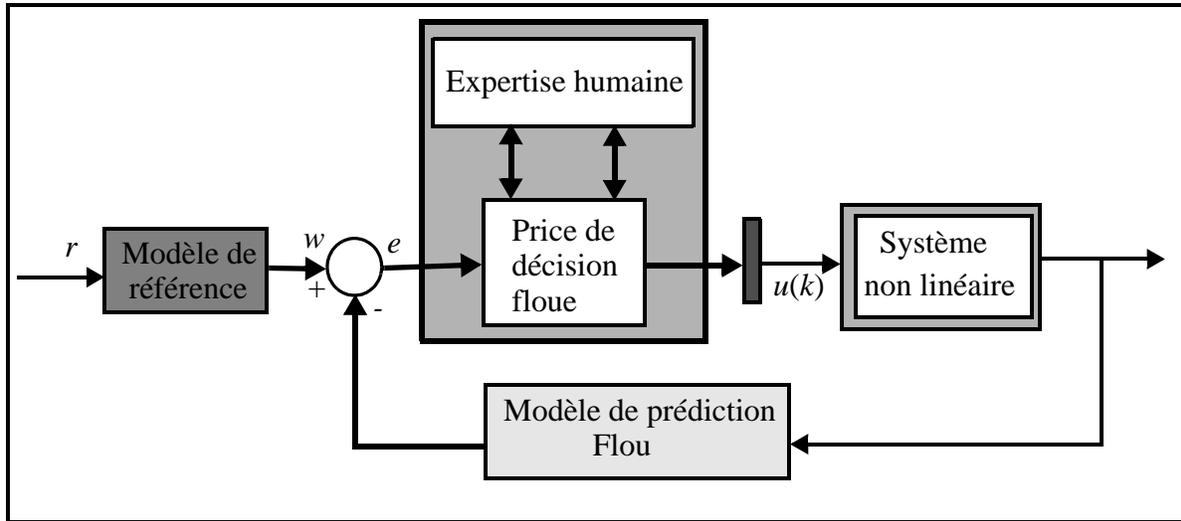


Figure II.20

La relation entre la sortie $y(k+r)$ et l'entrée $u(k)$ est donc la suivante:

$$y(k+r) = hof^{(r)}(x(k), u(k)) \quad (\text{II.104})$$

où r représente le degré relatif du système. Il représente le retard entre l'entrée $u(k)$ et la sortie $y(k)$. En d'autres termes, l'entrée du système à l'instant k n'affecte la sortie du système qu'après r unités de temps.

Si $\partial y(k+r)/\partial x(k)$ (le Jacobien de $hof^{(r)}$ par rapport à x) est non singulier à l'origine (point d'équilibre), l'état $x(k)$ peut être exprimé en fonction des entrées et sorties passées. Dans ce cas, l'équation Tableau 104, page 49 peut être réécrite comme suit:

$$y(k+r) = \Psi[y(k), y(k-1), \dots, y(k-n_y+1), u(k), u(k-1), \dots, u(k-m_u+1)] \quad (\text{II.105})$$

avec: $m_u \leq n_y$

De manière générale, dans cette thèse, nous nous intéressons à tout système non linéaire discret qui peut s'exprimer sous la forme suivante:

$$y(k+r) = \Psi[Z(k), u(k)] \quad (\text{II.106})$$

avec : $Z(k) = [y(k), \dots, y(k-n_y+1), u(k-1), \dots, u(k-m_u+1)]$

Pour des raisons de simplicité et sans perte de généralité, dans toute la suite le degré relatif r est supposé égal à 1. La méthode de commande reste valable pour un r quelconque. Dans ce cas, la représentation du système pour $r = 1$ est donnée par:

$$y(k+1) = \Psi[Z(k), u(k)] \quad (\text{II.107})$$

La représentation adoptée pour modéliser les systèmes non linéaires discrets est la représentation entrée-sortie donnée par l'équation (II.107). Ce choix permet de limiter le nombre de fonctions non linéaires à approximer par le modèle flou. Dans ce contexte, la fonction non linéaire Ψ est représentée par un système flou.

On distingue classiquement deux grandes familles de systèmes flous: systèmes flous à conclusions symboliques (systèmes de Mamdani) [25] et systèmes flous à conclusion fonctionnelle (systèmes flous de Takagi-Sugeno-Kang) [46]. Ces deux types de systèmes flous sont basés sur une collection de règles "Si ... Alors". Dans les deux cas, les prémisses de règles sont exprimées symboliquement. Seule l'expression des conclusions de règles permet alors de dissocier les deux familles de systèmes. Les contrôleurs de Mamdani utilisent en effet des conclusions symboliques de même nature que les prémisses alors que les systèmes de Sugeno exploitent des conclusions numériques. Cette différence dans l'écriture des conclusions de règles permet de distinguer les deux types de systèmes à leur aspect externe. D'un point de vue interne, un mécanisme de calcul est associé à chaque famille de systèmes. Pour les systèmes de Sugeno, celui-ci est purement numérique et s'exprime aisément de manière analytique selon une approche unique, commune à l'ensemble des systèmes de la famille. En revanche l'implantation des systèmes de Mamdani peut être envisagée de différentes façons. On aboutit alors à une typologie des systèmes de Mamdani telle qu'elle est développée dans [25]. Il est à noter que les systèmes flous relationnels peuvent être assimilés à des systèmes de Mamdani à implantation symbolique.

Dans ce travail on limite la présentation des systèmes flous à ceux qui seront directement exploités dans les approches de commande développées, à savoir les systèmes flous de Takagi-Sugeno à conclusion polynomiale.

Dans ce cas, le système non linéaire est représenté par un système flou de Takagi-Sugeno dont la base de règles est une collection de règles de la forme suivante:

$$\begin{aligned}
 & R^{(i)} : \\
 & \text{Si } y(k) \text{ est } A_1^i \text{ et } \dots \text{ et } y(k-n_y+1) \text{ est } A_{n_y}^i \text{ et } u(k) \text{ est } B_1^i \text{ et } \dots \text{ et} \\
 & \quad u(k-m_u+1) \text{ est } B_{m_u}^i \text{ Alors } y_i(k+1) = \phi(i_1, \dots, i_{n_y}, j_1, \dots, j_{m_u}) = \\
 & \quad a_1^i \cdot y(k) + \dots + a_{n_y}^i \cdot y(k-n_y+1) + b_1^i \cdot u(k) + \dots + b_{m_u}^i \cdot u(k-m_u+1) + c^i
 \end{aligned} \quad (\text{II.108})$$

où: A_p^i et B_q^i , $p = 1, \dots, n_y$ et $q = 1, \dots, m_u$ sont respectivement les symboles flous associés aux variables $y(k-p+1)$ et $u(k-q+1)$.

et: a_p^i et b_q^i sont respectivement des paramètres constants associés aux variables $y(k-p+1)$ et $u(k-q+1)$ de la règle indexée i .

Supposons N_p et M_q symboles flous définis sur les univers de discours pour décrire respectivement les entrées $y(k-p+1)$ et $u(k-q+1)$, dans ce cas la base de règles complète est composée de:

$$S = \prod_{p=1}^{n_y} N_p \prod_{q=1}^{m_u} M_q \quad (\text{II.109})$$

La règle Tableau 108, page 50, peut s'écrire sous la forme simplifiée suivante:

$$\begin{aligned}
 & R^i : \\
 & \text{Si } Z(k) \text{ est } S^i \text{ et } u(k) \text{ est } B^i \text{ Alors } y_i(k+1) = a^i \cdot Z(k) + b^i \cdot u(k) + c^i
 \end{aligned} \quad (\text{II.110})$$

où:

$$Z(k) = \begin{bmatrix} y(k) \\ \dots \\ y(k-n_y+1) \\ u(k-1) \\ \dots \\ u(k-m_u+1) \end{bmatrix}, S^i = \begin{bmatrix} A_1^i \\ \dots \\ A_{n_y}^i \\ B_2^i \\ \dots \\ B_{m_u}^i \end{bmatrix}, a^i = \begin{bmatrix} a_1^i \\ \dots \\ a_{n_y}^i \\ b_2^i \\ \dots \\ b_{m_u}^i \end{bmatrix}, B^i = B_1^i, \text{ et } : b^i = b_1^i \quad (\text{II.111})$$

Pour une entrée $Z_1(k) = [y(k), \dots, y(k-n_y+1), u(k), \dots, u(k-m_u+1)] \in R^{n_y+m_u}$, la sortie générée par le système flou est donnée par:

$$y(k+1) = \frac{\sum_{i=1}^S \lambda^i(Z_1) \cdot [a^i \cdot Z(k) + b^i \cdot u(k) + c^i]}{\sum_{i=1}^S \lambda^i(Z_1)} \quad (\text{II.112})$$

où λ^i est le vecteur des degrés de validité des prémisses de règles donnée par:

$$\lambda^i(Z_1) = \prod_{p=1}^{n_y} \mu_{A_p^i}(y(k-p+1)) \cdot \prod_{q=1}^{m_u} \mu_{B_q^i}(u(k-q+1)) \quad (\text{II.113})$$

Une fois le modèle de prédiction flou identifié, la méthodologie d'optimisation exploitée repose sur les mêmes techniques décrites précédemment, c'est-à-dire l'utilisation de la programmation dynamique floue (PDF) ou la méthode Branch & Bound.

II.10. Conclusion

Dans ce chapitre, la commande prédictive non linéaire floue a été introduite. Après avoir défini les notions de objectif, de contrainte flous et de décision flous, le critère d'optimisation flou a été introduit dans la cadre de commande prédictive. Les méthodologies de solutions appropriées à savoir la programmation dynamique floue et la méthode Branch&Bound ont été décrites. Nous avons également mis en évidence l'intérêt de ces méthodes par la simplification du formalisme du problème d'optimisation. Dans ce contexte, un niveau de flexibilité sur les objectifs et les contraintes peut être spécifié à travers la définition des fonctions d'appartenance et leurs univers de discours.

CHAPITRE III:

SUR LA STABILITE DE LA COMMANDE PREDICTIVE FLOUE

III.1. Introduction

Dans la synthèse et la conception des systèmes de commande, l'une des exigences les plus importantes à vérifier est celle de la stabilité du système commandé. Dans un contexte, il est nécessaire d'analyser le problème de stabilité avant la mise en oeuvre de l'algorithme de commande sur le système réel. Sachant que la commande prédictive non-linéaire (C.P.N) permet de définir un retour d'état, déterminé par un algorithme d'optimisation, il est naturel de vérifier que ce retour d'état assure la stabilité de la trajectoire d'équilibre et cela sous des hypothèses d'existence de la solution et de la faisabilité du problème.

Dans la conception et l'étude de la stabilité des systèmes non linéaires, la méthodologie à base du théorème de stabilité de Lyapunov est souvent utilisée. Cette dernière sera utilisée dans notre approche. En effet, la méthode utilisée se base sur le fait qu'il est possible de démontrer l'existence d'une solution admissible, si la fonction de coût est décroissante et qu'elle peut être interprétée comme une fonction de Lyapunov.

Dans la majorité des travaux [26] le problème de la stabilité de la commande prédictive non linéaire est basé sur la définition des conditions suffisantes pour assurer la stabilité du système en boucle fermée autour de la trajectoire d'équilibre. En général, les hypothèses couramment utilisées sont réparties entre le système et le critère.

Pour l'étude de la stabilité de la commande prédictive floue on utilise l'approche de Lyapunov où l'on a un critère d'optimisation floue qui est considéré comme fonction candidate de Lyapunov. Ce choix peut se justifier intuitivement par le fait que le critère est une mesure de la qualité de poursuite. Nous nous limitons au cas de l'horizon fini pour des raisons évidentes concernant l'applicabilité et la réalisabilité.

III.2. Commande Prédictive Floue Duale

III.2.1. Formulation générale

Nous rappelons ici le modèle d'état non linéaire pour un cas discret donnée dans le chapitre 1:

$$x(k+1) = f(x(k), u(k))$$

avec

$$x \in X \subset R^n, u \in U \subset R^m$$

où X et U sont respectivement les domaines définis par les contraintes sur les variables d'états et sur la commande.

L'algorithme de la commande prédictive floue utilisé consiste à résoudre, à chaque période d'échantillonnage, le problème d'optimisation suivant:

$$P_N(x): \quad \text{Max}_u [\Lambda_{j=1}^N (\mu_C(u(k+j-1/k)) \wedge \mu_G(x(k+j/k)))]$$

Il s'agit alors de vérifier que la commande générée par l'algorithme d'optimisation permet assure la stabilité. Les étapes de ce travail sont illustrées sur la figure suivante:

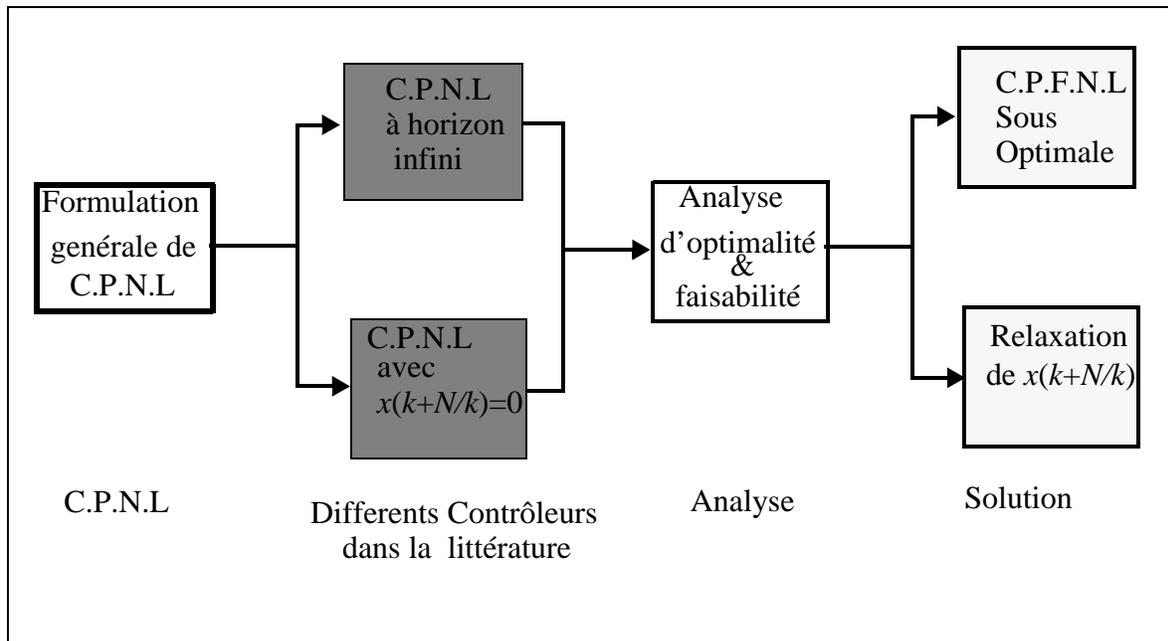


Figure III.2: Stratégie de l'approche

Certains travaux sur la commande prédictive à base de modèles ont utilisé un horizon de prédiction infini. En effet, il se trouve qu'un horizon infini présente un traitement plus facile et notamment une stabilité asymptotique. Il permet également de prendre en compte des contraintes plus dures. Cependant, en implantation pratique seul un horizon fini est applicable. Les principaux travaux existants sur l'étude de la stabilité de la C.P.N sont ceux donnés dans [26] pour le cas discret où il s'agit de forcer la sortie prédite du modèle:

- à rejoindre la référence dans le cas de poursuite ou
- égale à zéro dans le cas de régulation.

et cela après un nombre fini d'instant. Cette convergence est assurée en ajoutant une contrainte d'égalité sur la sortie finale, atteinte par le système, c'est-à-dire:

$$x(k + N/k) = 0, \forall k \quad (\text{III.1})$$

Cette condition finie, relativement assez sévère, est la base de la majorité des hypothèses assumées dans les différents travaux de recherche [26] pour le cas à horizon fini.

Toutefois, cette condition permet de forcer la convergence de l'algorithme d'optimisation à la fin de l'horizon de prédiction. En effet, elle assure la décroissance de la fonction de coût qui devient une condition nécessaire pour la stabilité de la trajectoire d'équilibre du système en boucle fermée. Dans ce contexte, il est important d'insister sur le fait que la difficulté majeure pour l'algorithme d'optimisation est de réaliser une solution en un temps fini et réduit avec une telle contrainte. L'analyse de la stabilité dans les problèmes de C.P.N avec un horizon fini, est une tâche compliquée, et plus spécialement dans le cas avec des contraintes [5]. Dans ce cadre, afin de remédier à ce problème, un grand effort a été fait pour le résoudre à travers des nouveaux outils et sous certaines hypothèses de base. Dans l'approche développée dans cette thèse, la contrainte terminale Tableau III.1, page 54 est relaxée. En effet, une nouvelle contrainte, comme définie dans quelques autres travaux [6][34], sera posée. Cette dernière est détaillée dans les paragraphes suivants.

III.2.2. Hypothèse de l'approche

Les hypothèses utilisées dans notre méthode pour l'étude de la stabilité de la C.P.N.L sont les hypothèses qui concernent la commandabilité. Ces dernières peuvent être vue sous deux facettes différentes:

- L'ensemble défini par les contraintes n'est pas vide et contient le couple de solution (0,0). Cela confère une commandabilité ou une faisabilité en un certain sens puisque, quelque soit l'état initial, on suppose que l'état (0,0) est atteignable.
- L'horizon de prédiction permettant d'atteindre cet état est supposé fini. Ceci est équivalent à l'existence d'un temps fini N tel que:

$$x(k + N/k) \in X_f, \forall k \quad (\text{III.2})$$

On remarque ici que la contrainte d'égalité terminale Tableau III.1, page 54 a été remplacée par une contrainte d'appartenance. En d'autres termes, on impose que l'état final de prédiction appartienne à une région bien spécifiée (cf. équation Tableau III.2, page 55).

Dans ce cas, en supposant cette contrainte d'appartenance, il faut garantir la stabilité du système à l'intérieur de cette région terminale X_f où une loi de commande linéaire stabilisante ($k_f=Kx$) est synthétisée pour le système linéarisé à l'origine.

Le succès de cette approche est basée, en premier lieu, sur le résultat bien-connu suivant:

- **Théorème** [26]

Soit $A = f_x(0,0)$ et $B = f_u(0,0)$ la linéarisation de la fonction du système non linéaire f autour de l'origine. Si la paire (A, B) est stabilisable alors il existe une matrice de gain K tel que le système en boucle fermée:

$$x(k + 1) = f(x(k), u(k)) = f(x(k), K \cdot x(k)) \quad (\text{III.3})$$

est localement asymptotiquement stable.

Dans ce cas, une commande prédictive à mode dual, dual mode predictive control, est proposée. Elle est basée sur les équations suivantes:

$$x(k + 1) = \begin{cases} f(x(k), u(k)), & x \in X \\ A \cdot x(k) + B \cdot u(k), & x \in X_f \end{cases} \quad (\text{III.4})$$

C'est à dire à l'extérieur de la région terminale, la loi de commande est calculée en se basant sur le système non linéaire. En revanche à l'intérieur de la région X_f , cette loi de commande est calculée en fonction du système linéarisé.

La stratégie de la commande prédictive floue permet, sous des hypothèses d'existence de la solution au problème d'optimisation, de ramener le système vers la région terminale ou d'attraction. Une fois cette région terminale atteinte, une commutation vers un régulateur linéaire est opéré. Ainsi la méthodologie de commande prédictive flou à mode dual, proposée repose sur les deux étapes suivantes:

- En dehors de la région terminale X_f , la commande prédictive floue est appliquée.
- Dans la région terminale X_f , la loi de retour d'état linéaire est utilisée.

III.3. Analyse de stabilité

Comme on l'a déjà constaté dans le premier chapitre de cette thèse, la stabilité et la faisabilité de la C.P.N sont indexées sur la vérification de plusieurs hypothèses sur le critère qui sont souvent très restrictives. Ainsi la C.P.N à horizon infini n'est pas applicable. Donc, l'algorithme de la commande prédictive floue à horizon fini avec des contraintes moins restrictives sera utilisé pour résoudre ce problème et cela tout en bénéficiant des avantages de l'utilisation d'un critère flou.

L'idée principale sous-jacente à la démonstration de la stabilité réside dans le constat suivant: *S'il est possible de trouver une solution admissible et que l'horizon de prédiction suffit pour ramener le système vers la région terminale alors la fonction coût sera décroissante et il sera possible de l'interpréter comme une fonction de Lyapunov garantissant la stabilité.*

III.3.1. Problème

Rappelons que le problème d'optimisation est donné comme suit:

$$\begin{aligned}
 & \text{Max}_u (\bigwedge_{j=1}^N [\mu_C(u(k+j-1/k)) \wedge \mu_G(x(k+j/k))]) \\
 & x(k+j/k) = f(x(k+j-1/k), u(k+j-1/k)) \\
 & u(k+j/k) \in U \\
 & x(k+j/k) \in X \\
 & x(k+N/k) \in X_f
 \end{aligned} \tag{III.5}$$

Le problème de la stabilité de la commande prédictive floue porte sur la définition des conditions suffisantes pour assurer la stabilité du système Tableau , page 56 en boucle fermée autour de la trajectoire d'équilibre. Les hypothèses utilisées sont réparties entre le système Tableau , page 56 et le critère Tableau III.5, page 56. Sachant que l'expression définissant le critère est un élément clé dans la synthèse de la commande prédictive non linéaire, l'hypothèse la plus prépondérante porte sur ce dernier.

Pour prouver la stabilité de la commande prédictive floue double, on exige la vérification des hypothèses suivantes:

- HP_1 : $k_f(x) \in U, \forall x \in X_f$
- HP_2 : $0 \in X_f, X_f \subset X$ et X_f est un ensemble fermé.
- HP_3 : $f(x, k_f(x)) \in X_f, \forall x \in X_f$
- HP_4 : $\mu_C(k_f(x)) = 1$ et $\mu_G(x) = 1$.

On remarque que les trois premières hypothèses (HP_1, HP_2 et HP_3) sont triviales et elles sont déjà utilisées dans la commande prédictive non linéaire [26]. En revanche, la 4^{ème} hypothèse impose que:

- dans la région d'attraction, le critère flou sera égal à 1:

$$\mu_C(k_f) \wedge \mu_G(x) = 1, \forall x \in X_f \tag{III.6}$$

- à l'extérieur de cette région d'attraction:

$$\mu_C(u) \wedge \mu_G(x) < 1 \tag{III.7}$$

Dans le cas de la commande prédictive floue, l'objectif est de ramener le système vers son point

d'équilibre (l'origine). Pour ce faire, on a choisi une fonction d'appartenance de forme triangulaire pour représenter l'objectif flou dans le critère flou. Sa valeur modale est le point d'équilibre. Autrement dit, on le favorise par rapport aux autres points. En revanche, dans le cas de la commande prédictive floue double où l'objectif de la loi de commande est de ramener le système vers la région terminale, on choisira une fonction trapézoïdale (cf. Tableau 3, page 57) pour présenter les contraintes et les objectifs flous. Cette représentation a pour objectif de s'assurer que le noyau de l'objectif flou représentera les frontières de chaque état dans la région terminale et le noyau de la contrainte floue représentera les limites de la commande linéaire locale.

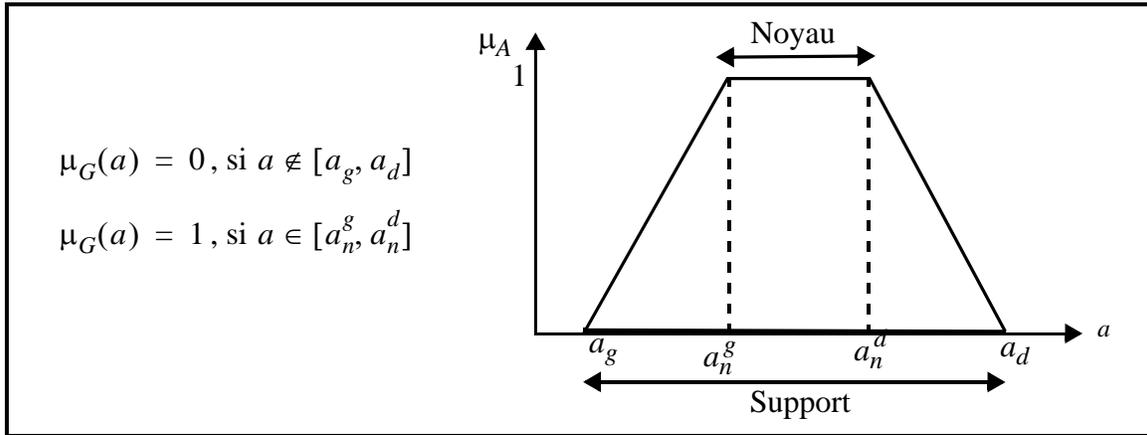


Figure III.3: Forme Trapézoïdale

Avant de prouver la stabilité de la commande prédictive floue double, il est utile de préciser les points suivants:

- La condition $0 \leq J_N(x, u) \leq 1$ est toujours vérifiée pour le critère.
- Le complément du critère est donné par $1 - J_N(x, u)$
- Le problème d'optimisation $Max_u(J_N(x, u))$ est équivalent à un problème d'optimisation $Min(1 - J_N(x, u))$, c'est-à-dire:

$$Max_u(J_N(x, u)) = Min((1 - J_N(x, u))) = J_N'(x, u) \quad (III.8)$$

et

$$V_N(x^*, u^*) = Min(J_N'(x, u)) \quad (III.9)$$

On remarque également que:

- $V_N(x, u)$ est une fonction positive.
- $V_N(x, u) \neq 0$ si $x \notin X_f$ et $V_N(x, u) = 0$ si $x \in X_f$

La stabilité énoncée par le théorème suivant est assurée à travers la décroissance du critère d'optimisation:

$$V_N(x^{+*}, u^{+*}) - V_N(x^*, u^*) < 0 \quad (III.10)$$

III.3.2. Théorème [1]

Le système non linéaire Tableau III.3, page 55 contrôlé par une commande prédictive floue duale et vérifiant les hypothèses HP_1 - HP_4 est asymptotiquement stable avec X_f comme région

d'attraction.

III.3.3. Démonstration

La démonstration de la stabilité est basée sur le fait que s'il est possible de trouver une solution admissible et qu'il existe un horizon N_j suffisant pour ramener le système vers la région terminale, alors la décroissante du critère sera une condition primordiale pour assurer la stabilité (la décroissante du critère le long des trajectoires).

La valeur optimale de la fonction coût à l'instant k est donnée par:

$$V_N(x^*, u^*) = \text{Min}_u (J_N'(x, u)) \quad (\text{III.11})$$

avec:

$$0 \leq V_N(x^*, u^*) \leq 1 \quad (\text{III.12})$$

De plus à l'intérieur de la région d'attraction X_f , les contraintes et objectifs flous vérifient l'équation:

$$V_N(x, u) = 1 - J_N(x, u) = 0, x \in X_f \quad (\text{III.13})$$

Dans ce cas, pour prouver la stabilité, il faut démontrer la décroissante de la fonction Tableau III.11, page 58, quand le système à commander passe de l'état $x(k)$ à l'état $x(k+1)=x^+$. C'est à dire:

$$\Delta V = V_N(x^{+*}, u^{+*}) - V_N(x^*, u^*) \leq 0 \quad (\text{III.14})$$

Supposons que la séquence de commandes optimales et la trajectoire associée à l'instant k sont déterminées en vérifiant le problème d'optimisation proposé Tableau III.5, page 56:

$$\begin{aligned} u^* &= (u^*(k/k), u^*(k+1/k), \dots, u^*(k+N-1/k)) \\ x^* &= (\hat{x}^*(k+1/k), \hat{x}^*(k+2/k), \dots, \hat{x}^*(k+N/k)) \end{aligned} \quad (\text{III.15})$$

avec

$$V_N(x^*, u^*) = 1 - \Lambda_{j=1}^N (\mu_C(u^*(k+j-1/k)) \Lambda(\mu_G(x^*(k+j/k)))) \quad (\text{III.16})$$

Le premier élément de la séquence de commande optimale $u_{opt}(x) = u^*(k/k)$ est appliqué au système en le conduisant vers l'état suivant:

$$x(k+1) = f(x(k/k), u_{opt}(x)) \text{ avec un coût optimal} \quad (\text{III.17})$$

Nous allons maintenant déterminer la séquence admissible u^+ qui conduit le système de l'état $x(k+1)$ vers l'état $\hat{x}^+(k+N/k) \in X_f$. Cette séquence est admissible mais non optimale.

Son coût associé $J_N(x^+, u^+)$ est la borne supérieure du coût optimal $V_N(x^{+*}, u^{+*})$.

Cette séquence est obtenue en prenant la séquence optimale u^* calculée au moment précédent, sans son premier élément et en ajoutant un élément de commande admissible u_1 qui conduit: l'état $\hat{x}^*(k+N/k) \in X_f$ à l'état $\hat{x}^+(k+N/k) = f(\hat{x}^*(k+N/k), u_1) \in X_f$. C'est-à-dire:

$$u^+ = (u^*(k+1/k), u^*(k+2/k), \dots, u_1) \quad (\text{III.18})$$

avec:

- $u_1 \in U$:il faut que la commande locale stabilisante vérifie les contraintes de la commande
- $f(x, u_1) \in X_f$: la sortie du système doit toujours rester dans la région terminale et converger vers le point d'équilibre quand on lui applique la commande locale u_1 et ce quelle que soit la valeur de x appartenant à la région terminale.

Sachant que $\hat{x}^*(k + N/k) \in X_f$ et à partir des hypothèses HP_1 - HP_3 , il est possible d'écrire:

$$\begin{aligned} u_1 &= k_f(\hat{x}^*(k + N/k)), u_1 \in U, \forall x \in X_f \\ f(x, u_1) &\in X_f, \forall x \in X_f \end{aligned} \quad (III.19)$$

La séquence de trajectoire correspondante est donnée par:

$$x^+ = (\hat{x}^*(k + 2/k), \hat{x}^*(k + 3/k), \dots, \hat{x}^*(k + N/k), f(\hat{x}^*(k + N/k), u_1)) \quad (III.20)$$

et son coût associé J_N est de la forme suivante:

$$\begin{aligned} J_N(x^+, u^+) &= \\ 1 - \Lambda_{j=1}^{N-1} &(\mu_C(u^*(k + j/k)) \Lambda \mu_G(\hat{x}^*(k + j + 1/k))) \Lambda (\mu_C(u_1) \Lambda \mu_G(f(\hat{x}^*(k + N/k), u_1))) \end{aligned} \quad (III.21)$$

La différence entre $J_N(x^+, u^+)$ et $V_N(x^*, u^*)$ est donnée par:

$$\begin{aligned} J_N(x^+, u^+) - V_N(x^*, u^*) &= \\ \{ 1 - \Lambda_{j=1}^{N-1} &(\mu_C(u^*(k + j/k)) \Lambda \mu_G(\hat{x}^*(k + j + 1/k))) \\ \Lambda (\mu_C(u_1) \Lambda \mu_G &(f(\hat{x}^*(k + N/k), u_1))) \} - \\ \{ 1 - \Lambda_{j=1}^N &(\mu_C(u^*(k + j - 1/k)) \Lambda (\mu_G(x^*(k + j/k)))) \} \end{aligned} \quad (III.22)$$

où:

$$\begin{aligned} J_N(x^+, u^+) - V_N(x^*, u^*) &= \\ \{ -\Lambda_{j=1}^{N-1} &(\mu_C(u^*(k + j/k)) \Lambda \mu_G(\hat{x}^*(k + j + 1/k))) \\ \Lambda (\mu_C(u_1) \Lambda \mu_G &(f(\hat{x}^*(k + N/k), u_1))) \} + \\ \{ \mu_C(u^*(k/k)) \Lambda \mu_G &(x^*(k + 1/k)) \\ \Lambda_{j=2}^N &(\mu_C(u^*(k + j - 1/k)) \Lambda (\mu_G(x^*(k + j/k)))) \} \end{aligned} \quad (III.23)$$

Le fait que la séquence de commande u^+ soit simplement une version décalée dans le temps de la séquence optimale u^* additionnant l'élément u_1 et qui vérifie les hypothèses HP_1 - HP_4 , implique que le problème, en plus d'être admissible car il l'était à l'instant précédent, va entraîner une diminution de la valeur de la fonction coût.

Etant donné que le vecteur u^+ est tel que $u^*(k+1/k) = u^*(k+1/k+1)$, l'implantation à l'instant $k+1$ du vecteur de commande u^+ produit les prévisions $\hat{y}(k+i/k+1) = \hat{y}(k+i/k)$.

On peut écrire la différence de la façon suivante:

$$\begin{aligned}
& J_N(x^+, u^+) - V_N(x^*, u^*) = \\
& -T(x^*, u^*) \Lambda(\mu_C(u_1) \Lambda \mu_G(f(\hat{x}^*(k + N/k), u_1))) + \\
& T(x^*, u^*) \Lambda \mu_C(u^*(k)) \Lambda(\mu_G(x^*(k + 1/k)))
\end{aligned} \tag{III.24}$$

avec le terme commun:

$$T(x^*, u^*) = \Lambda_{j=2}^N(\mu_C(u^*(k + j - 1/k)) \Lambda(\mu_G(x^*(k + j/k))) \tag{III.25}$$

en utilisant la condition Tableau III.6, page 56 nous obtenons:

$$\begin{aligned}
& J_N(x^+, u^+) - V_N(x^*, u^*) = \\
& -T(x^*, u^*) + T(x^*, u^*) \Lambda \mu_C(u^*(k/k)) \Lambda(\mu_G(x^*(k + 1/k)))
\end{aligned} \tag{III.26}$$

et avec la condition Tableau III.7, page 56:

$$-T(x^*, u^*) + T(x^*, u^*) \Lambda \mu_C(u^*(k/k)) \Lambda(\mu_G(x^*(k + 1/k))) < 0 \tag{III.27}$$

ce qui conduit à;

$$J_N(x^+, u^+) - V_N(x^*, u^*) < 0 \tag{III.28}$$

d'où finalement le résultat suivant:

$$V_N(x^{+*}, u^{+*}) - V_N(x^*, u^*) < 0 \tag{III.29}$$

Comme $V_N(x, u)$ est la fonction candidate de Lyapunov associé au système commandé par la commande prédictive à mode dual sa décroissance permet donc de conclure la stabilité asymptotique du système commandé.

III.4. Méthodologie d'optimisation

Dans le chapitre précédent, nous avons évoqué l'importance de la méthode de résolution du problème d'optimisation qui permet de déterminer le vecteur de commande optimal ou admissible en un temps inférieur à la période d'échantillonnage. Dans la démarche proposée, notre choix est basé sur deux méthodes discrètes qui garantissent la convergence vers la solution optimale et avec un moindre calcul. En effet, cette solution est basée sur la programmation dynamique (DP) associée à la technique de Branch et Bound (B&B). Contrairement à l'utilisation la plus courante de la DP (Backward), nous proposons d'utiliser ici sa version Forward. Autrement dit, pour appliquer cette dernière avec la programmation dynamique, nous devons développer la programmation dynamique sous sa forme récurrente *forward* dans le problème d'optimisation de commande prédictive floue.

Le problème de la commande prédictive floue à base de modèles est un problème de prise de décision multi-niveaux où à chaque niveau $i=1, \dots, N$, l'état prédit et la variable de commande sont discrétisés en deux ensembles finis:

$$\begin{aligned}
& \hat{x}(k + i/k) \in X = \{\zeta_1, \zeta_2, \dots, \zeta_n\}, i = 1 \dots N \\
& u(k + i - 1/k) \in U = \{v_1, v_2, \dots, v_m\}, i = 1 \dots N.
\end{aligned} \tag{III.30}$$

Cette discrétisation nous mène vers un tableau de N niveaux.

En commençant par l'état initial $x(k)$ où la décision initiale sera calculée au niveau $i=1$ comme suit:

$$\mu_D(u(k/k)) = \text{Max}_{u(k/k)} \mu_C(u(k/k)) \wedge \mu_G(\hat{x}(k+1/k)) \quad (\text{III.31})$$

pour tout

$$u(k/k) \in U, \text{ et } \hat{x}(k+1/k) \in X, \text{ avec } \hat{x}(k+1/k) = f(x(k/k), u(k/k))$$

En utilisant le principe d'optimalité au niveau $i=2$, la décision intermédiaire est calculée de la façon suivante:

$$\begin{aligned} \mu_D(u(k/k), u(k+1/k)) = \\ \text{Max}_{u(k+1/k)} \{ \mu_C(u(k+1/k)) \wedge \mu_G(\hat{x}(k+2/k)) \wedge \mu_D(u(k/k)) \} \end{aligned} \quad (\text{III.32})$$

pour tout

$$u(k+1/k) \in U, \text{ et } \hat{x}(k+2/k) \in X, \text{ avec } \hat{x}(k+2/k) = f(\hat{x}(k+1/k), u(k+1/k))$$

Le même principe est appliqué au niveau $i=3$ conduisant au résultat suivant:

$$\begin{aligned} \mu_D(u(k/k), u(k+1/k), u(k+2/k)) = \\ \text{Max}_{u(k+3/k)} \{ \mu_C(u(k+2/k)) \wedge \mu_G(\hat{x}(k+3/k)) \\ \wedge \mu_D(u(k/k), u(k+1/k)) \} \end{aligned} \quad (\text{III.33})$$

pour tout

$$u(k+2/k) \in U, \text{ et } \hat{x}(k+3/k) \in X, \text{ avec } \hat{x}(k+3/k) = f(\hat{x}(k+2/k), u(k+2/k))$$

En répétant cette procédure, la récurrence *forward* de la programmation dynamique au niveau $i+1$ est obtenue et formalisée par l'équation suivante:

$$\begin{aligned} \mu_D(u(k/k), \dots, u(k+j/k)) = \\ \text{Max}_{u(k+j/k)} \{ \mu_C(u(k+j/k)) \wedge \mu_G(\hat{x}(k+j+1/k)) \\ \wedge \mu_D(u(k/k), \dots, u(k+j-1/k)) \} \end{aligned} \quad (\text{III.34})$$

pour tout

$$u(k+j/k) \in U, \text{ et } \hat{x}(k+j/k) \in X, \text{ avec } \hat{x}(k+j+1/k) = f(\hat{x}(k+j/k), u(k+j/k))$$

Dans ce cas, pour obtenir la solution de la commande prédictive floue, la forme de récurrence *forward* de la programmation dynamique DP est exécutée jusqu'à l'horizon de prédiction N . Toutefois, comme discuté précédemment, aucune garantie sur le fait que l'état converge vers la région terminale n'est apportée. La même constatation peut être faite concernant toutes les trajectoires partant de $x(k)$ (la convergence durant l'horizon N de ces trajectoires vers la région terminale n'est pas garantie).

En revanche s'il y a des trajectoires qui convergent vers la région terminale durant un horizon N , leur dynamique de convergence n'est pas forcément la même (quelques-unes convergent plus tôt que d'autres). Par conséquent, nous définissons une séquence faisable de contrôle, notée $\Pi_i(\hat{x}(k+N_i)/k)$ comme une séquence qui force l'état prédit à converger vers la région termi-

nale X_f dans un temps fini $N_i < N$. Dans ce cas, la décision correspondante est donnée par:

$$\begin{aligned} & \mu_D(\Pi_i(\hat{x}(k + N_i)/k)) = \\ & \text{Max}_{u(k+N_i-1)} \{ \mu_C(u(k + N_i - 1/k)) \wedge \mu_G(\hat{x}(k + N_i/k)) \wedge \\ & \mu_D(u(k/k), \dots, u(k + N_i - 2/k)) \} \end{aligned} \quad (\text{III.35})$$

avec la contrainte:

$$\hat{x}(k + N_i/k) \in X_f \quad (\text{III.36})$$

La séquence optimale (solution de Tableau III.35, page 62 et Tableau III.36, page 62) est donnée par l'expression suivante:

$$(u^*(k/k), u^*(k + 1/k), \dots, u^*(k + N_i - 1/k))$$

telle que:

$$\begin{aligned} & \mu_D(u^*(k/k), u^*(k + 1/k), \dots, u^*(k + N_i - 1/k)) = \\ & \text{Max}_{u(k+N_i-1)} (\Pi_i(\hat{x}(k + N_i)/k)) \end{aligned} \quad (\text{III.37})$$

Il est important d'insister sur le fait que le choix de l'horizon de prédiction est crucial. Quant à la qualité de la solution et les calculs qui lui sont associée, parmi les stratégies existantes il faut adopter celles qui:

- commencent avec une valeur initiale $x(k)$ pour un horizon de la prédiction N ,
- si aucune solution n'est obtenue avec cette valeur, la récurrence Tableau III.34, page 61 est utilisée, en augmentant l'horizon de prédiction et ce jusqu'à ce que la première solution faisable soit obtenue.

A ce stade, pour améliorer les performances de notre approche, nous allons intégrer le concept de la technique de Branch de Bound. En effet, cette dernière arrive à éliminer des solutions partielles qui ne mènent pas à la solution recherchée. De ce fait, on arrive souvent à obtenir la solution en des temps raisonnables.

Dans la recherche de la solution optimale, chaque sous ensemble de solutions abandonnées s'appelle «*fathoming*» et la meilleure solution faisable s'appelle «*incumbent*» [33].

Dans la suite, le paramètre « V_{inc} » correspond au coût de la meilleure solution appelée «*incumbent*». Ce paramètre est vu comme une borne discriminatoire. Autrement dit, à travers cette borne certaines solutions seront exclues ou maintenues comme des solutions potentielles. Bien entendu, la performance de la méthode de Branch & Bound dépend, entre autres, de la qualité de cette borne supérieure «*incumbent*» et de sa capacité d'exclure des solutions partielles tôt.

Les différents étapes de l'algorithme vont être résumées par les trois tests suivants:

- **Test d'optimalité**

La trajectoire ou la branche est éliminée si:

$$V(\hat{x}(k + j/k)) \geq V_{inc}$$

- **Test de solution**

Si la trajectoire finale converge vers la région terminale avec un coût:

$$V(\hat{x}(k + N/k)) < V_{inc}$$

alors la trajectoire est bloquée et son coût correspondant sera la nouvelle solution admissible,

c'est-à-dire:

$$V_{inc} = V(\hat{x}(k + j/k))$$

- **Test de faisabilité**

Toute trajectoire est bloquée si elle conduit à une solution instable ou oscillante et ne respecte pas les contraintes des problèmes d'optimisation.

III.4.1. Algorithme de la méthode

Dans l'implantation de l'algorithme de notre méthode, une première étape de calcul «Hors ligne» est nécessaire pour déterminer la région terminale X_f et la loi de commande locale stabilisante k_f . Ce point est détaillé dans le paragraphe suivant. Une deuxième étape «en ligne» est donnée par la procédure suivante:

Etape 0

Initialisation

$$k = 0, x(0), V_{inc} = 0$$

Si $x(0) \in X_f$ **Alors**

allez à l'étape 2

Sinon

- Choix de $N = N_i$.
- Calcul de la solution faisable:

$$\Pi_0 = \{u(0/0), u(1/0), \dots, u(N+1/0)\}$$

qui respecte la contrainte Tableau III.2, page 55 et ce par exécution des trois tests précédents (test d'optimalité, de solution et de faisabilité).

Si la première solution faisable est trouvée **Alors**

- la recherche est arrêtée.
- implémentation de $u(0) = u(0/0)$

Sinon

- on augmente l'horizon de prédiction jusqu'à ce que la première solution soit trouvée.
- $N \leftarrow N + 1$

Fin Si

Fin Si

Etape 1

Si $x(k) \in X_f$ **Alors**

allez à l'étape 2

Sinon

- $V_{inc} = V_N(x^+, u^+)$
- exécuter le critère Tableau III.5, page 56 pour $k=1, \dots, N$
- calcul de la solution faisable:

$$\Pi_k = \{u(k/k), u(k+1/k), \dots, u(k+N-1/k)\}$$

en exécutant les trois tests précédents (test d'optimalité, de solution et de faisabilité) et en re-

spectant la contrainte Tableau III.2, page 55.

- implémentation de $u(k) = u(k/k-1)$

Fin Si

Etape 2

Si le temps $k > 0$ et $x(k) \in X_f$ **Alors**

- Implémentation de la commande locale $u(x(k))=k_f(x(k))$

Fin Si

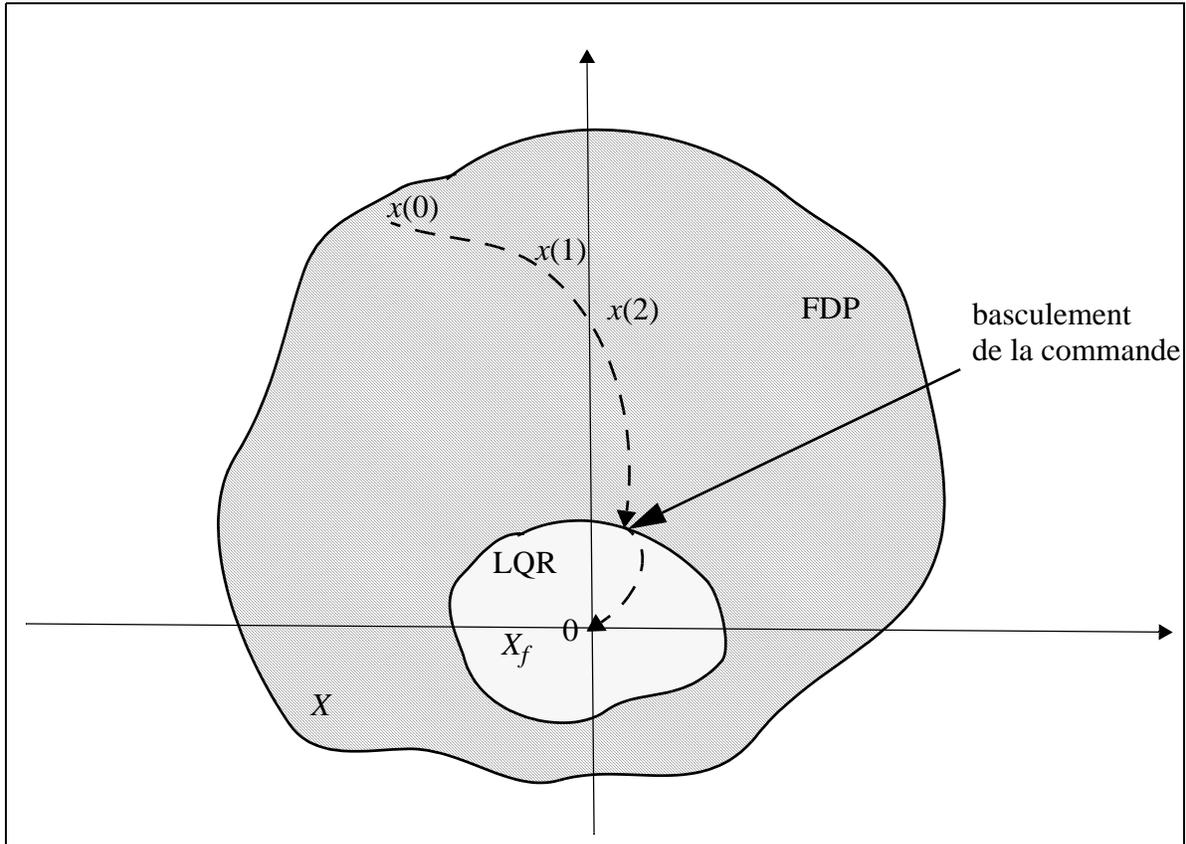


Figure III.4: Principe de la commande prédictive floue dual

III.4.2. Calcul de la région terminale

Il existe plusieurs méthodes, basées sur les mêmes principes (étapes de construction), pour la construction de la région terminale X_f et le calcul de la loi de commande locale k_f .

La première étape consiste à trouver la loi de commande linéaire $k_f(x) = K \cdot x$ pour le système linéarisé autour du point d'origine (d'équilibre). Dans la deuxième étape, la région terminale du système en boucle fermée est déterminée par la résolution de l'équation de Lyapunov. Dans la troisième étape, on cherche la plus grande région X_f telle que la commande locale respecte la contrainte de commande $k_f(x) \in U$ à l'intérieur de ce domaine d'attraction.

Toutes ces étapes seront considérées comme un problème d'optimisation hors ligne. A titre d'exemple, nous donnons l'algorithme de calcul de la région d'attraction dans le cas discret proposé par [34]

Algorithme

Etape 1

- Calculer la linéarisation Jacobienne (A, B) pour aboutir à:

$$x(k+1) = A \cdot x(k) + B \cdot u(k)$$

avec:

$$A = \left. \frac{\partial f(x, u)}{\partial x} \right|_{(0,0)}, B = \left. \frac{\partial f(x, u)}{\partial u} \right|_{(0,0)} \quad (\text{III.38})$$

- Déterminer le retour d'état linéaire stabilisant $k_f = K \cdot x(k)$. Dans ce cas, à titre d'exemple, le régulateur linéaire quadratique (LQR), Placement des pôles peuvent être utilisés.

Etape 2

- Choisir γ et η tel que:

$$\begin{cases} \lambda_{\min}(K^T R K) < \gamma \leq \lambda_{\min}(\bar{Q}) \\ \eta > \frac{(\gamma - \lambda_{\min}(K^T R K))}{\lambda_{\min}(Q)} \end{cases} \quad (\text{III.39})$$

avec $\bar{Q} = Q + K^T R K$

- Résoudre l'équation de Lyapunov suivante:

$$(A + BK)^T P (A + BK) - P + \eta Q + K^T R K = 0 \quad (\text{III.40})$$

afin d'obtenir la matrice définie positive P avec Q, R des matrices définies positives.

Etape 3

- Trouver la valeur c définissant la plus grande région possible $X_f \subset X$ telle que:

$$X_f = \{x \in \mathfrak{R}^n / x^T P x \leq c\} \quad (\text{III.41})$$

en résolvant le problème d'optimisation suivant:

$$c = \max_x (x^T P x) \quad (\text{III.42})$$

$$\text{telque } f(x, Kx)^T P f(x, Kx) - x^T P x \leq -\gamma x^T x$$

$$\text{avec } Kx \in U, \forall x \in X_f = \{x \in \mathfrak{R}^n / x^T P x \leq c\}$$

Dans l'implantation de la méthode, la considération pratique suivante peut être prise en compte. en effet, les paramètres γ et η peuvent être optimisés afin d'agrandir la région X_f et cela autant que possible mais en respectant les caractéristiques de la loi de commande locale stabilisante.

III.5.Conclusion

Dans ce chapitre, une nouvelle approche pour l'étude de la stabilité de la commande prédictive floue est proposée. La fonction de coût est interprétée comme une fonction de Lyapunov et la stabilité du système commandé est établi en démontrant sa décroissance le long de la trajectoire. Par ailleurs un algorithme d'optimisation est construit sur la base d'une combinaison de deux approches discrètes, DP et BB. Notons enfin que cette approche a l'inconvénient, celui de la discontinuité qui s'introduira sur la commande (commutation de $x(k) \in X$ à $x(k) \in X_f$).

CHAPITRE IV:

EXEMPLES D'APPLICATIONS

IV.4. Introduction

Dans ce dernier chapitre, nous allons présenter des exemples d'applications pour mettre en évidence l'efficacité de la commande prédictive floue dans la commande des systèmes non linéaires. En effet, nous allons montrer la capacité de la méthode proposée à ramener l'état du système dans la région terminale ou d'attraction, où une loi de commande linéaire sera appliquée pour faire converger l'état du système à son point d'équilibre.

Une fois la région terminale notée X_f et la commande locale linéaire k_f déterminées, nous allons choisir les univers de discours des fonctions d'appartenance des contraintes et des objectifs flous, de telle sorte que les hypothèses de stabilité de la commande prédictive floue double seront vérifiées.

Pour évaluer la performance de notre stratégie de commande, des systèmes largement exploités dans les travaux traitant de la commande prédictive non linéaire sont choisis. Le premier système sera utilisé pour montrer l'efficacité de la commande prédictive floue à faire converger un système non linéaire vers son point d'équilibre ainsi que l'influence du choix de l'opérateur d'agrégation sur les performances de commande.

Les deux autres systèmes nous serviront à montrer la convergence et la stabilité asymptotique des systèmes non linéaires multi-variables commandés par une commande prédictive floue double.

IV.5. Exemples d'applications

VI.5.1. Exemple 1: Système Non Linéaire 1^{er} ordre

Supposons un système non linéaire [41] perturbé par une entrée additive $p(k)$ et donné par l'équation d'état suivante:

$$x(k+1) = x^2(k) + u^3(k) + p(k), \text{ avec : } p(k) = 1/k \quad (\text{IV.4})$$

où:

- $x(k)$, $u(k)$ sont respectivement l'état et l'entrée (commande) du système.
- $p(k)$ est une perturbation additive.

Dans ce cas, le modèle de prédiction utilisé est choisi comme suit:

$$x(k+1) = x^2(k) + u^3(k) \quad (\text{IV.5})$$

L'objectif principal ici est de stabiliser l'état du système autour de son point d'équilibre tout en restant dans le domaine de tolérance de l'état, c'est-à-dire:

$$X = [x_{min}, x_{max}] = [-2.5, 2.5], \text{ i.e., } \{x \in R^n / |x| \leq 2.5\} \quad (\text{IV.6})$$

La contrainte imposée à la commande est donnée par:

$$U = [u_{min}, u_{max}] = [-2, 2], \text{ c'est-à-dire } \{u \in R / |u| \leq 2\} \quad (IV.7)$$

A titre illustratif, l'évolution de $x(k+1)$ en fonction de $x(k)$ et $u(k)$ est illustrée sur la Tableau 21, page 69.

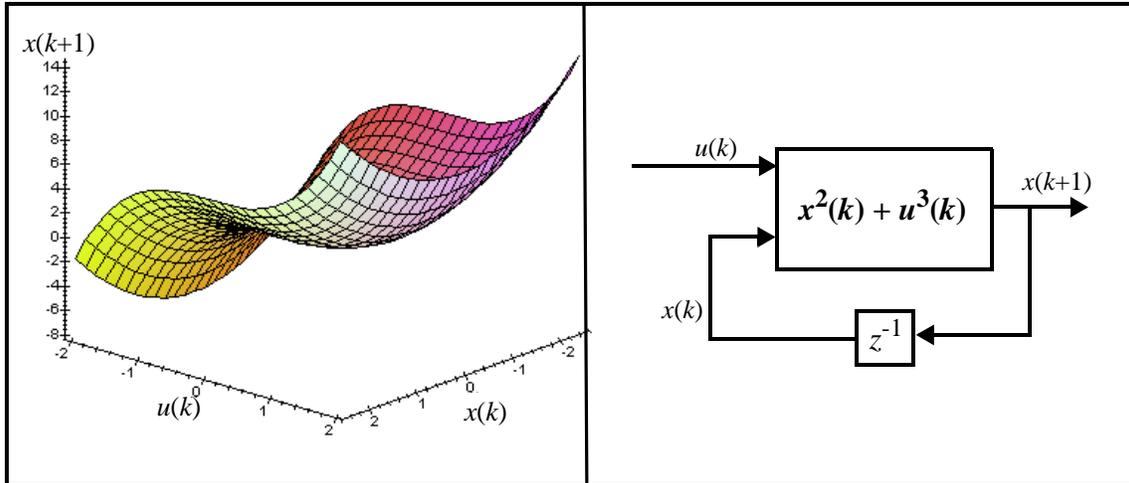


Figure IV.21 : Représentation non linéaire du système considéré

Afin de réaliser notre objectif, nous allons exploité l'approche P.D.F pour implanter la commande prédictive avec un critère de décision floue. Dans ce contexte, l'implantation est faite avec les paramètres suivants:

- l'horizon de prédiction N_2 est fixé à 5.
- l'horizon de commande N_u est fixé à 2.
- le pas de discrétisation est fixé à $\Delta = 0.2$.
- En fonction de notre expertise sur le système, les fonctions d'appartenance associées à l'état et à la commande sont choisies respectivement triangulaire et trapézoïdale définies sur leur univers de discours X et U (cf. Tableau 22, page 70).

Dans ce cas, le critère de décision à optimiser est donné par l'équation suivante:

$$\begin{aligned} \mu_D(u(0), \dots, u(4)) &= \Lambda_{i=0}^4 [\mu_C(u(i)) \Lambda \mu_G(x(i+1))] \\ &= [\mu_C(u(0)) \Lambda \mu_G(x(1))] \Lambda \dots \Lambda [\mu_C(u(4)) \Lambda \mu_G(x(5))] \end{aligned} \quad (IV.8)$$

avec: $u(1) = u(2) = u(3) = u(4)$ (l'horizon de commande $N_u = 2$).

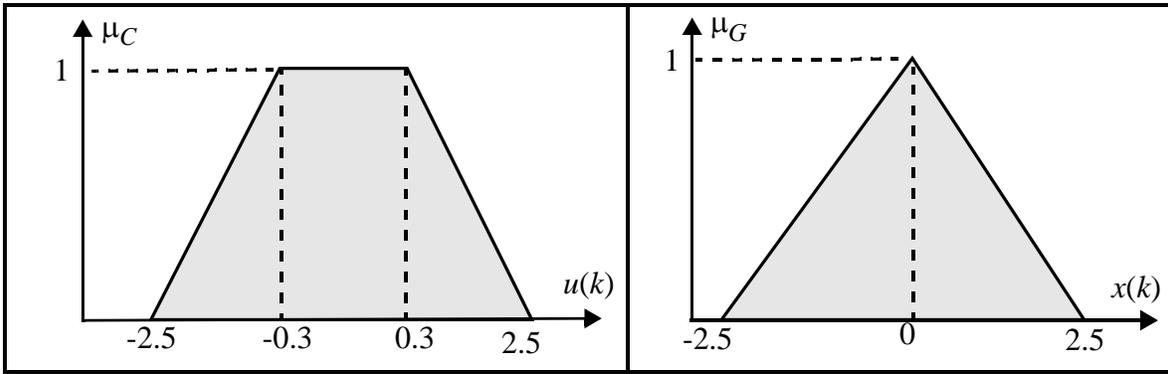


Figure IV.22 Contrainte floue et objectif flou (Exemple1)

Par application de l’algorithme de la P.D.F avec les opérateurs minimum et produit, les résultats de simulation avec une condition initiale $x(0) = 3$ sont illustrés sur les Tableau 23, page 70 et Tableau 24, page 70 suivantes:

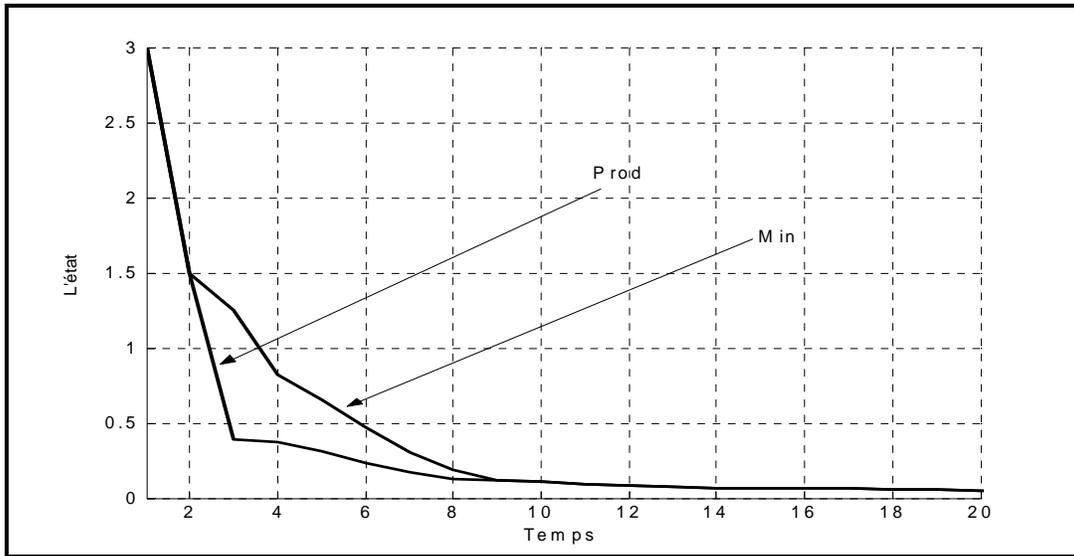


Figure IV.23 : Evolution de l'état $x(k)$

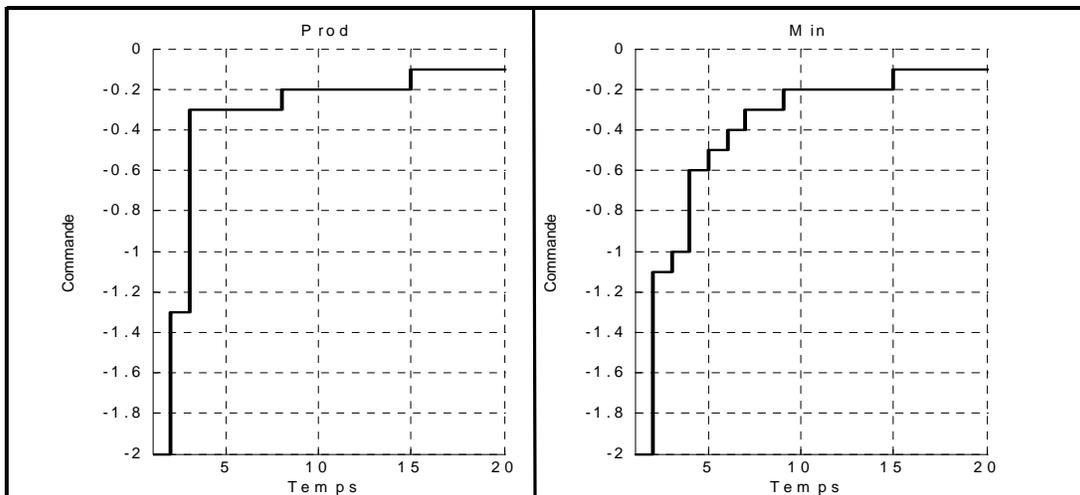


Figure IV.24 : Evolution de la commande $u(k)$

On remarque que le système converge vers son état d'équilibre, et ce malgré l'existence d'une perturbation. La T -norme produit donne un régime transitoire moins important que la T -norme min. En effet, sachant que le min est la plus grande T -norme, elle est la plus contraignante et donne souvent des résultats moins bon que la majorité des T -normes et des opérateurs d'agrégation.

Afin de pouvoir vérifier la robustesse de la méthode de commande utilisée, nous allons conserver le même modèle de prédiction Tableau 5, page 68 utilisé précédemment pour commander le système suivant, qui n'a pas la même structure (incertitudes structurelles):

$$x_{k+1} = x_k^3 + 0,5u_k^3 + p_{k+1},$$

$$p_k = \begin{cases} 0, & k \neq 10 \\ 1, & k = 10 \end{cases} \quad (\text{IV.9})$$

Dans les simulations, le même algorithme d'optimisation avec les mêmes paramètres que précédemment seront utilisées. Les résultats sont illustrés sur la Tableau 25, page 71 et la Tableau 26, page 72 où les performances de la commande sont établies. De plus, ces résultats montrent une certaine robustesse de la méthode utilisée vis-à-vis des incertitudes dans la structure du système.

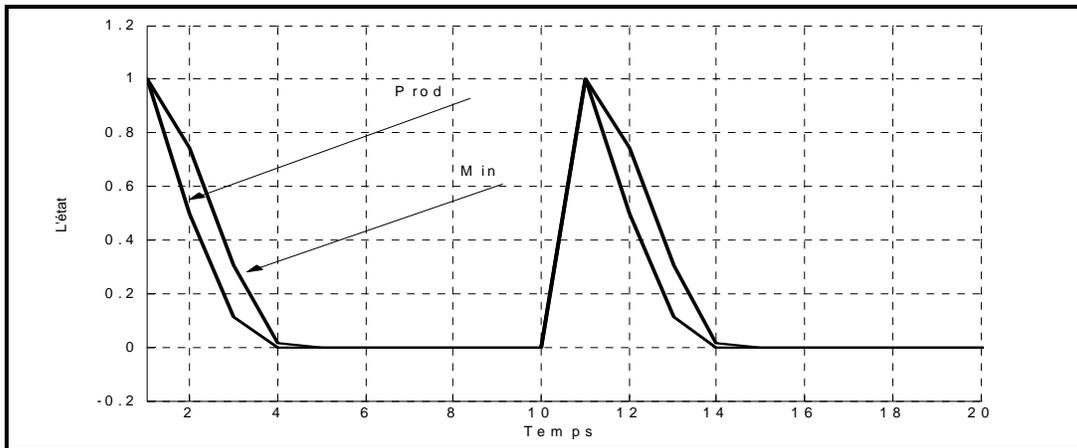


Figure IV.25 Evolution de l'état $x(k)$

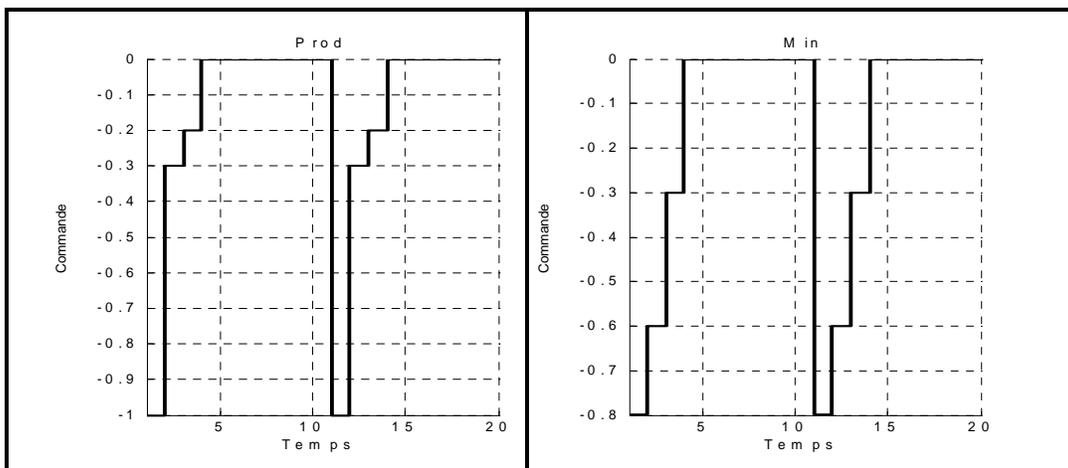


Figure IV.26 Evolution de l'état $u(k)$.

VI.5.2. Exemple 2: Système Non Linéaire 2^{ème} ordre

Cet exemple est utilisé dans [21] et [41] où le système est décrit par:

$$\begin{cases} \dot{x}_1 = x_2 + u \cdot (\mu + (1 - \mu) \cdot x_1) \\ \dot{x}_2 = x_1 + u \cdot (\mu - 4(1 - \mu) \cdot x_2) \end{cases} \quad (\text{IV.10})$$

Le paramètre μ est donné par $\mu = 0.5$ et la variable de commande u sera soumise à une contrainte donnée par: $u \in \{u / -2 \leq u \leq 2\}$

Le système est localement asymptotiquement stable et commandé par une loi de retour d'état linéaire $k_f = K \cdot x$, tel que $K = [2.118, 2.118]$ dans sa région terminale [41]:

$$X_f = \{x^T \cdot P \cdot x \leq 0, 7\} \text{ avec } P = \begin{bmatrix} 16,5926 & 11,5926 \\ 11,5926 & 16,5926 \end{bmatrix} \text{ et } \alpha=0.7 \quad (\text{IV.11})$$

Les univers de discours des contraintes et des objectifs flous sont choisis de façon à ce que l'hypothèse HP_4 (données dans le troisième chapitre) est toujours vérifiée pour tout x appartenant à la région terminale X_f (cf. Tableau 27, page 72).

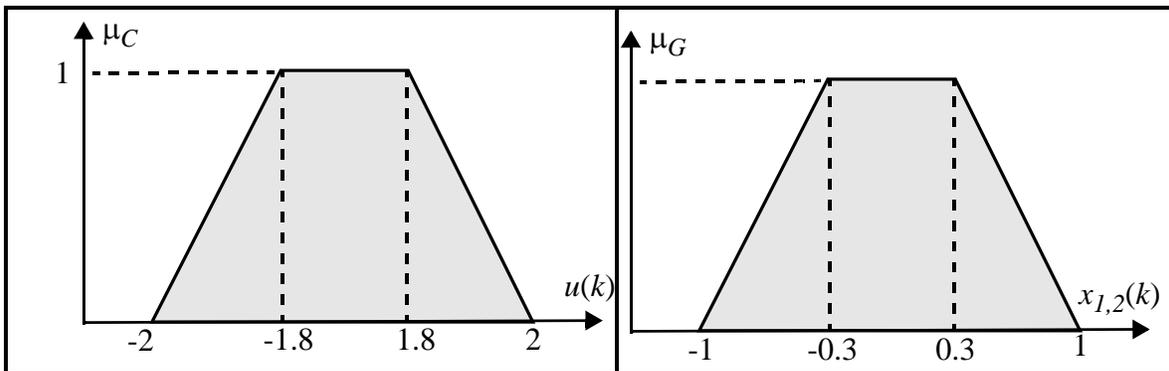


Figure IV.27 Contrainte floue et objectifs flous (Exemple2)

Enfin, la programmation dynamique floue avec la technique de Branch & Bound est utilisée pour résoudre le problème d'optimisation de la commande prédictive floue pour un horizon de prédiction égale $N_2=5$, et l'algorithme de commande sera testé sur les deux conditions initiales suivantes:

$$[x_1(0), x_2(0)]^t = [-0,683, -0,847]^t \text{ et } [x_1(0), x_2(0)]^t = [0,808, 0,122]^t. \quad (\text{IV.12})$$

l'implantation est faite avec les paramètres suivants:

- l'horizon de prédiction N_2 est fixé à 5.
- le pas de discrétisation est fixé à $\Delta = 0.1$ pour la variable de commande et les états x_1 et x_2 .

Les résultats de simulation sont présentés par les figures Tableau 28, page 73 , Tableau 29, page 74 et Tableau 30, page 74.

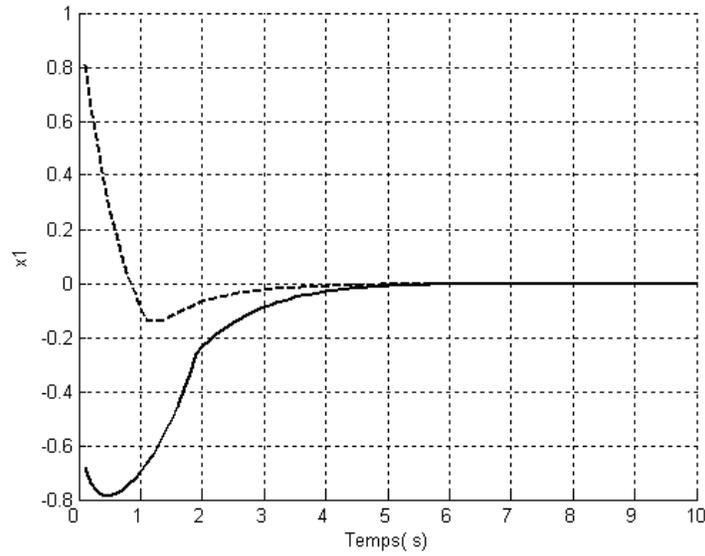


Figure IV.28 : Evolution de l'état x_1

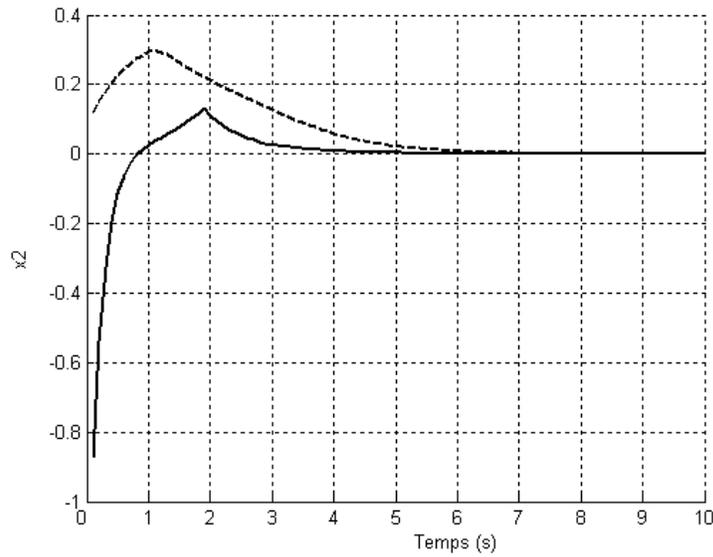


Figure IV.29 : Evolution de l'état x_2

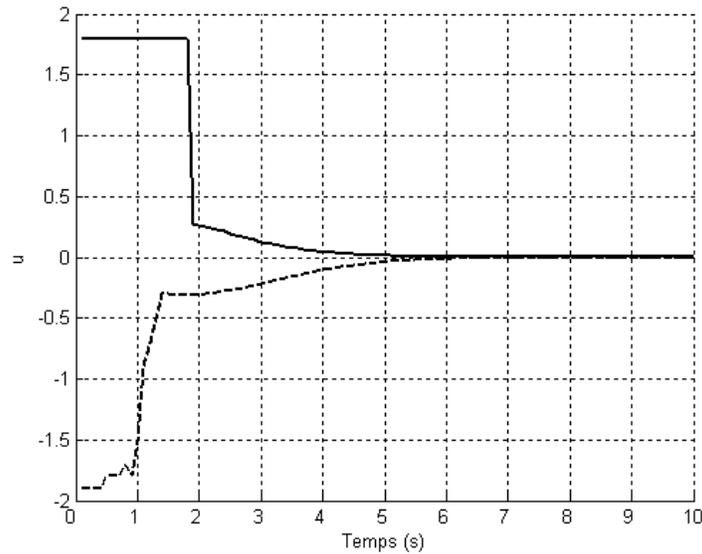


Figure IV.30 : Evolution de la commande u .

Durant les douze premiers instants (huit premiers instants pour la deuxième condition initiale), la commande est calculée par la commande prédictive floue, à l'instant 13 (09), l'état du système entre dans la région terminale X_f . A ce moment là, la commande locale stabilisante conduit l'état du système asymptotiquement vers l'origine (point d'équilibre).

VI.5.3. Exemple 3: Réacteur Chimique CSTR

Le procédé utilisé est un réacteur exothermique continu complètement agité (Continued Stirred Tank Reactor CSTR) [22][24]. Il est représenté par une réaction exothermique irréversible $A \rightarrow B$, pour un volume constant (cf. Tableau 31, page 75).

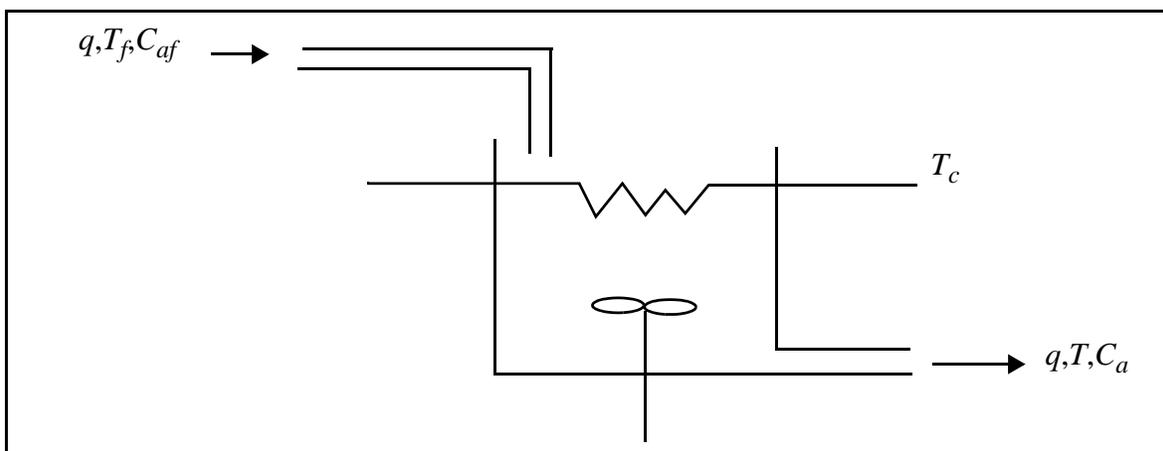


Figure IV.31 : Représentation du réacteur chimique CSTR

Où C_a est la concentration du produit A dans le reacteur, T est la température du recateur et T_c est la température du débit de refroidissant.

Le modèle de ce procédé peut être décrit par les deux équations différentielles suivantes:

$$\begin{aligned} \dot{C}_a &= \left(\frac{q}{V}\right)(C_{af} - C_a) - k_0 \cdot e^{(-E/RT)} \cdot C_a \\ \dot{T} &= \left(\frac{q}{V}\right)(T_f - T) - \frac{(\Delta H)}{\rho C_p} \cdot k_0 \cdot e^{(-E/RT)} \cdot C_a(t) + \frac{UA}{V\rho C_p}(T_c - T) \end{aligned} \quad (IV.13)$$

où C_{af} et T_f représentent respectivement la concentration et la température du processus à l'entrée.

Les paramètres du modèle sont:

$$\begin{aligned} \rho &= 1000 \text{ g/l}, C_p = 0.239 \text{ J/gK}, \Delta H = -5 \times 10^4 \text{ J/mol}, E/R = 8750 \text{ K}, \\ k_0 &= 7.2 \times 10^{10} \text{ min}^{-1}, UA = 5 \times 10^4 \text{ J/minK} \end{aligned}$$

Et ces conditions nominales sont:

$$q = 100 \text{ l/min}, T_f = 350 \text{ K}, V = 100 \text{ l}, C_{af} = 1.0 \text{ mol/l}$$

IV.5.3.1. Commande prédictive floue double

Le but de la commande prédictive floue double est de stabiliser la concentration C_a et la température du réacteur T vers leurs point d'équilibre, en manipulant T_c .

Pour ce faire, pour pouvoir appliquer l'algorithme de la commande prédictive floue double, il faut calculer dans un premier temps la commande linéaire locale ainsi la région terminale.

Le modèle continu non linéaire est discrétisé avec un pas de discrétisation $dt = 0.03 \text{ min}$ en utilisant l'approximation d'Euler et en choisissant le vecteur d'état suivant:

$$\begin{aligned} x &= [(C_a - C_a^{eq}); (T - T^{eq})] \\ u &= (T_c - T_c^{eq}) \end{aligned} \quad (IV.14)$$

$$\text{Où } C_a^{eq} = 0.5 \text{ mol/l}, T^{eq} = 350 \text{ K}, T_c^{eq} = 300 \text{ K}$$

On obtient alors le modèle non linéaire discrétisé suivant:

$$\left\{ \begin{aligned} x_1(k+1) &= \\ x_1(k) + \left(\frac{q}{V}\right) \cdot dt \cdot (C_{af} - (x_1(k) + 0, 5)) - k_0 \cdot dt \cdot (x_1(k) + 0, 5) e^{-\left(\frac{E}{R}\right) \cdot 1/(x_1(k) + T^{eq})} \\ x_2(k+1) &= \\ x_2(k) + \left(\frac{q}{V}\right) \cdot dt \cdot (T_f - (x_1(k) + 350)) - k_0 \cdot dt \cdot (x_1(k) + 0, 5) e^{-\left(\frac{E}{R}\right) \cdot 1/(x_1(k) + T^{eq})} + \\ &\quad k_2 \cdot dt \cdot ((u + T_c^{eq}) - (x_1(k) + T^{eq})) \end{aligned} \right. \quad (IV.15)$$

Dans la synthèse de la commande, les contraintes du cahier de charges à respecter sont les suivantes:

- L'objectif de commande est de produire la substance A avec une concentration comprise entre:

$$0 \leq C_a \leq 1 \Rightarrow -0.5 \leq x_1 \leq 0.5 \quad (\text{IV.16})$$

- La température du réacteur T varie entre:

$$280 \leq T \leq 370 \Rightarrow -70 \leq x_2 \leq 20 \quad (\text{IV.17})$$

- La température d'alimentation varie entre:

$$280 \leq T_c \leq 370 \Rightarrow -20 \leq u \leq 70 \quad (\text{IV.18})$$

Il est important d'attirer l'attention sur le fait que le point de fonctionnement choisi est instable et que le système ait un comportement non linéaire comme on peut le constater à travers les deux réponses en boucle ouverte à un échelon négatif -5 et un échelon positif +5 illustrées sur les figures Tableau 32, page 76 et Tableau 33, page 77.

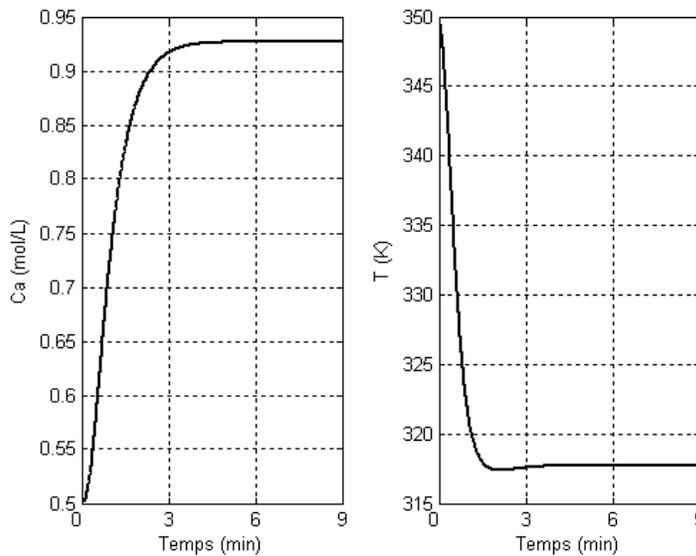


Figure IV.32 : Les réponses à un échelon -5

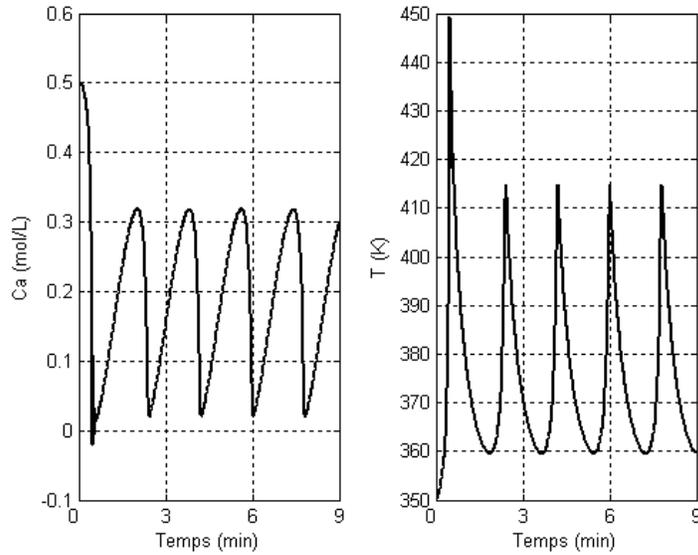


Figure IV.33 : Les réponses à un échelon +5

Le modèle discret linéarisé autour de l'origine est obtenu en calculant le Jacobien du modèle non linéaire discret Tableau 15, page 75 comme suit:

$$x(k+1) = Ax(k) + Bu(k) \quad (IV.19)$$

avec:

$$A = \begin{bmatrix} 0,96 & -0,007 \\ 4,1838 & 1,0876 \end{bmatrix}, B = \begin{bmatrix} 0 \\ 0,0418 \end{bmatrix}$$

On remarque que le système en boucle ouverte est instable.

Dans ce contexte, pour pouvoir appliquer la stratégie de la commande prédictive floue double, il est nécessaire de calculer la loi de commande locale $k_f = Kx$ qui satisfait les hypothèses HP_2 et HP_3 et ce dans la région terminale définie par:

$$X_f = \{x/x^T \cdot P \cdot x \leq \alpha\}. \quad (IV.20)$$

La matrice du gain K du contrôleur local linéaire est calculée en utilisant la technique LQR avec les matrices de pondérations suivantes:

$$Q = \begin{bmatrix} 2 & 0 \\ 0 & 1/300 \end{bmatrix}, R = 1/350$$

Finalement, on obtient le gain suivant:

$$K = [-91.33 \quad -4.70]$$

Nous allons maintenant calculer de la région terminale du réacteur CSTR. Dans ce contexte, la

matrice P de la fonction de Lyapunov qui définit la région terminale X_f est calculée en résolvant, pour $\lambda=1$, l'équation de Lyapunov suivante:

$$A_k^T \cdot P_\lambda \cdot A_k - P_\lambda = -\lambda \cdot Q^* \quad (\text{IV.21})$$

avec:

- $A_K = A + K \cdot B$ et
- la matrice du système en boucle fermée est donnée par:

$$Q^* = Q + K^T \cdot R \cdot K$$

Finalement, la matrice P est donnée par:

$$P = \begin{bmatrix} 54.394 & 0.592 \\ 0.592 & 0.035 \end{bmatrix}$$

De plus, il est nécessaire de déterminer le plus grand α définissant les frontières de la région d'attraction telle que les contraintes sur l'état et sur l'entrée soient satisfaites lorsque $x \in X_f$ et la condition (III.42) est vérifiée. Ce problème est résolu à travers une stratégie d'optimisation telle que détaillée dans le chapitre 3 [22]. Par exploitation du toolbox d'optimisation de Matlab, il résulte que $\alpha=0.62$.

Pour stabiliser la température du réacteur T et la concentration C_a vers leurs points d'équilibre, la commande prédictive floue double exploitant la programmation dynamique floue (P.D.F) et la technique de Branch & Bound est appliquée. L'implantation pratique est faite avec les paramètres suivants:

- l'horizon de prédiction N_2 est fixé à 2.
- les pas de discrétisation sont fixés à : $\Delta_{Tc} = 3$ pour la variable de commande, $\Delta_{Ca} = 0.1$ pour la concentration et $\Delta_T = 4$ pour la température.

Les univers de discours des contraintes et des objectifs flous sont choisis de façon à ce que l'hypothèse HP_4 (données dans le troisième chapitre) est toujours vérifiée pour tout x appartenant à la région terminale X_f (cf. Tableau 34, page 79).

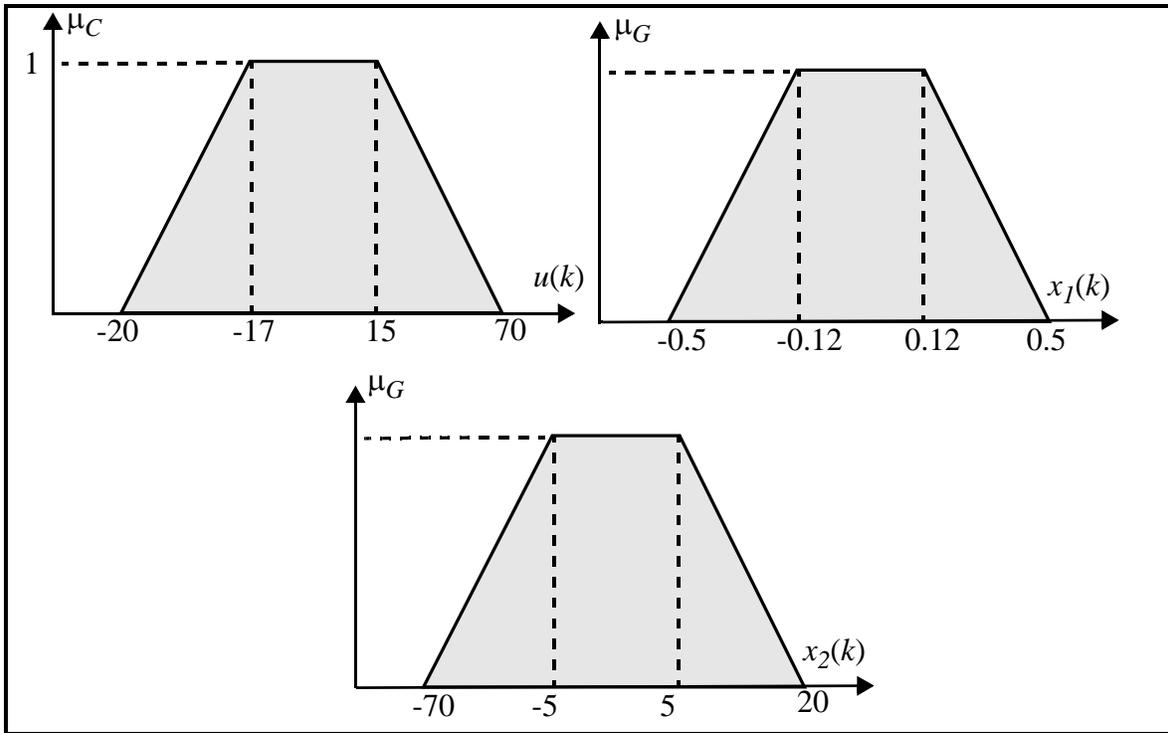


Figure IV.34 Contrainte floue et objectifs flous (Exemple3)

L'algorithme de commande sera testé sur les deux conditions initiales suivantes:

$$[C_a(0), T(0)]^t = [0.7 \ 350.5]^t \text{ et } [C_a(0), T(0)]^t = [0.3 \ 363]^t$$

Les résultats de simulation sont illustrés sur les figures Tableau 35, page 80, Tableau 36, page 80 et Tableau 37, page 81. A travers ces résultats on peut constater l'efficacité de l'algorithme de la commande prédictive floue pour un horizon de prédiction $N_2=2$. De plus, sa convergence vers la région terminale est rapide.

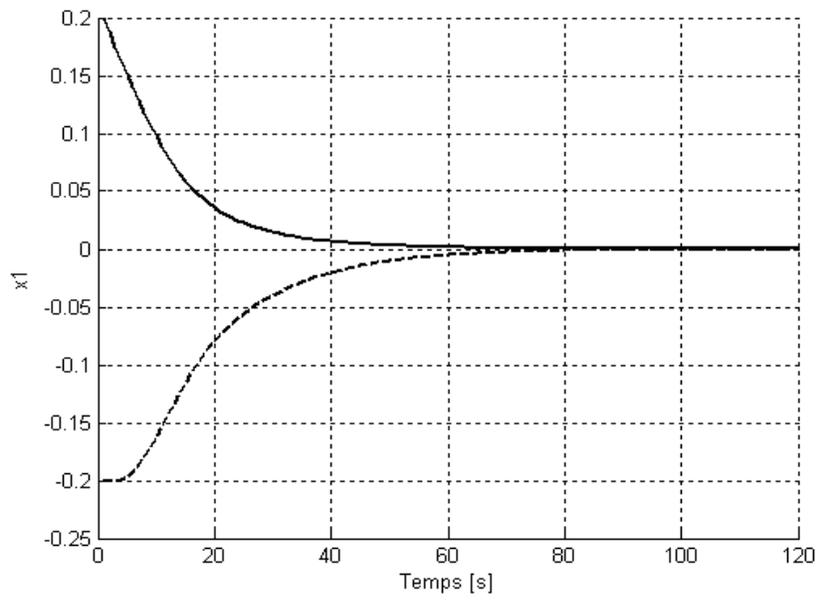


Figure IV.35 : L'évolution de l'état x_1

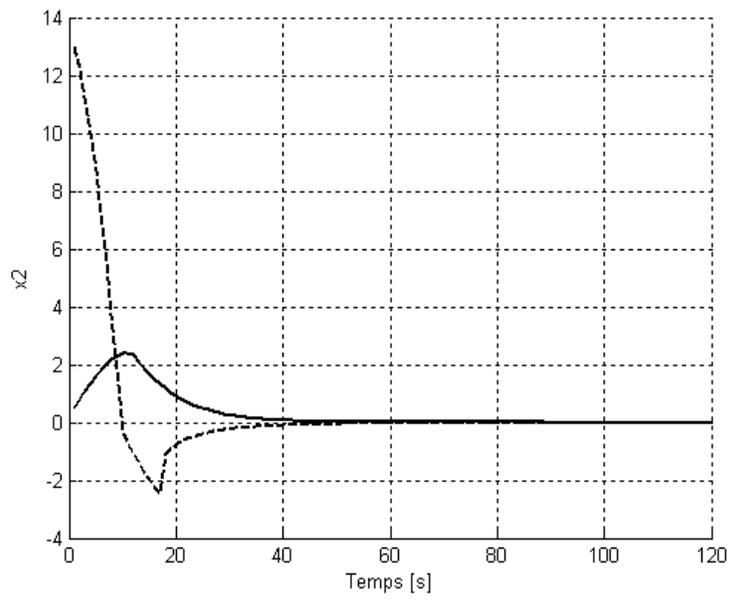


Figure IV.36 : L'évolution de l'état x_2

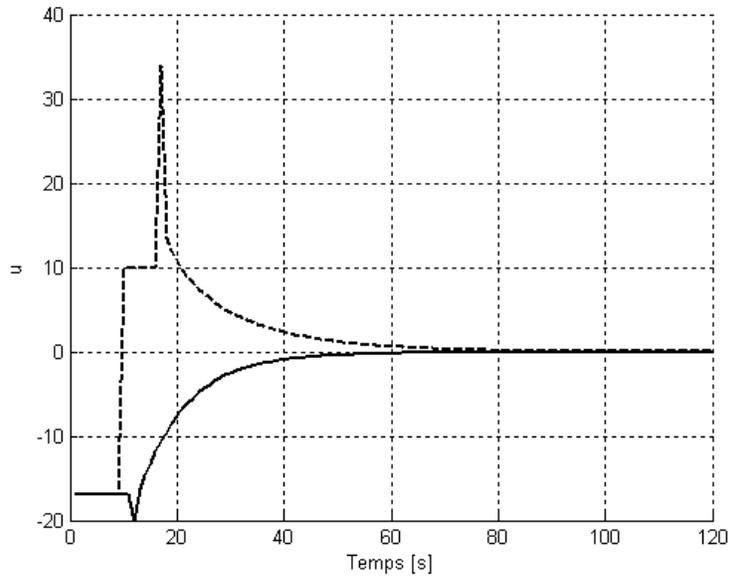


Figure IV.37 : l'évolution de la commande u

IV.6. Conclusion

A travers ce chapitre, nous avons illustré l'applicabilité de l'approche de commande prédictive floue développée dans les précédents chapitres afin de valider sa faisabilité.

Nous avons également mené une série de simulations sur des systèmes non linéaires largement exploités dans la littérature relative au domaine et ce conformément à la procédure détaillée au chapitre 2 et 3. A travers ces exemples de simulation, nous avons mis l'accent sur les performances de la méthode développée en termes de stabilité et convergence.

La croissance sans cesse importante de la puissance des logiciels de calcul, permet actuellement la mise en oeuvre de la commande prédictive floue double d'une manière relativement simple. Cependant, une attention particulière devrait être donnée au choix de l'horizon de prédiction afin de s'affranchir de calculs aussi lourds que inintéressants.

CONCLUSION GENERALE

Conclusion générale

Dans ce travail, nous avons proposé une méthodologie de synthèse d'une stratégie de commande dans le cadre de la commande prédictive des systèmes non linéaires. En effet, un contrôleur prédictif flou est avancé comme solution, dans le but de ramener l'état du système dans une région terminale où une commande locale asymptotiquement stable sera appliquée. La démarche de la méthode proposée peut être résumée par les deux étapes suivantes:

- Dans la première étape, nous avons choisi les conditions nécessaires concernant la synthèse de la loi de commande locale. En effet, les fonctions d'appartenance des contraintes et des objectifs flous, ainsi que la région terminale ont été fixés afin d'assurer la stabilité de la méthode à travers un critère flou vu comme une fonction de Lyapunov.
- Dans la seconde étape, nous avons opté pour des méthodes fiables pour résoudre le problème d'optimisation dans un contexte de temps réel (un calcul moins coûteux) et plus proche de la réalité (prise en compte de la présence de contraintes).

Sur le plan applicatif, l'accent a été mis sur la faisabilité et les performances de la démarche de commande proposée sur des systèmes non linéaires utilisés dans des références reconnues.

Ce travail ouvre plusieurs voies de recherche, aussi bien sur la conception de la commande que sur les systèmes non linéaires à considérer. En effet, parmi les extensions possibles on peut citer:

- L'élimination de l'application de la loi de commande locale dans la commande prédictive floue avec une démonstration de stabilité.
- L'élargissement des techniques de commande développées pour des systèmes dynamiques incertains. En effet, la majorité des techniques de commande développées au sein de la communauté automatique utilisent des entrées précises. Cependant, les informations délivrées par les organes de mesure sont par nature imprécises (qualité du capteur, sa tolérance et son environnement). Dans ce cadre, si on se place dans un schéma de commande en boucle fermée, il serait intéressant d'analyser et d'intégrer les effets induits par l'utilisation des entrées imprécises sur la performance des algorithmes de commande employés. Une attention particulière sera donnée à la représentation des grandeurs numériques incertaines par des approches possibilistes. En effet, si on considère des incertitudes paramétriques, une représentation de ces derniers selon l'approche possibiliste conduit à la représentation et à la résolution des systèmes à base d'intervalles et/ou d'intervalles flous.

BIBLIOGRAPHIE

- [1] K.Belarbi and F.Megri, « A stable-based fuzzy predictive control based on fuzzy dynamic programming», IEEE Transactions on fuzzy systems, Vol.15,pp.764-754, N°4, 2007
- [2] R.E.Bellman,and L.A.Zadeh. Decision Making In A Fuzzy Environment.Management Science, Vol 17,No,December,1970.
- [3] P.T.Boggs,J.W.Tolle, «Sequantial Quadratic Programming», Acta numerica,1996
- [4] E.F. Camacho and C. Bordons, « Model Predective Control», Springer-Verlag Berlin hdelberg New York , 1999
- [5] C.C.Chen and L.Shaw, «On receding horizon control», Automatica, Vol 16,pp.349-352,1982.
- [6] H.Chen and F.Allgower, «A quasi-infinite horizon nonlinear model predictive control scheme with guaranted stability», Automatica, Vol. 14, N° 10, pp. 1205-1217, 1998.
- [7] D.W. Clarke, and C. Mohtadi, «Properties of Generalized Predictive Control.Automatica. Part II. Extensions and Interpretations», Automatica, Vol. 25, N°. 6, pp. 859-875, 1989.
- [8] D.W. Clarke «Application of Generalized Predictive Control To Industrial Processes» IEEE Control Systems Magazine, 122:49655, 1988.
- [9] D.W. Clarke, C. Mohtadi, and P.S. Tuffs., «Generalized Predictive Control.Automatica. Part I.The Basic Algorithm», Automatica, Vol. 23, N° 2, pp. 137-148, 1987.
- [10] D.W. Clarke, C. Mohtadi, and P.S. Tuffs. «Generalized Predictive Control. Part II. Extensions and Interpretations», Automatica, Vol. 23, N° 2, pp. 149-160, 1987.
- [11] D.W. Clarke and R. Scattolini, «Constrained Receding-Horizon Predictive Control», Proceedings IEE, Vol. 122, pp. 929-934, 1975.
- [12] C.R. Culter and B.C. Ramaker., «Dynamic Matrix Control - A Computer Control Algorithm», In Automatic Control Conférence, San Francisco,1980.
- [13] C.E. Garcia, D.M. Prett, and M. Morari., «Model Predictive Control Theory and Practice. A Survey», Automatica, Vol. 25, N° 3, pp. 335-348, 1989.
- [14] P.E. Gill, W. Murray and M. Wright, «Pratical Optimization», Acedemic Press, New York, NY, 1981.
- [15] R.E. Kalman, «Contributions to the theory of optimal control», Bull. Soc. Math. Mex.Vol. 15, pp. 102-119, 1960
- [16] J.Kacprzyk, «Multistage Control of a Fuzzy system using a genetic algorithm ,Evolutionary Computation», IEEE International Conference, Vol.2, pp.842-845, 1995.
- [17] J.Kacprzyk and A.O. Esogbue., «Fuzzy Dynamic Programming. Main developments and applications», Fuzzy Sets and Systems, Vol. 81, pp. 31-45, 1996.
- [18] S.S. Keerthi and E.G.Gilbert, «Optimal infinite-horizon feedback laws for a general class of constrained discrete-time systems:Stability and moving-horizon approximations», Journal of optimization theory and applications ,Vol. 57, N° 2, pp. 265-293, May 1988.
- [19] G. J. Klir., «Fuzzy Sets and Fuzzy Logic: Theory and Applications», Prentice Hall, 1995.
- [20] J.P.LASALLE, «The stability of dynamical systems», in regional Conference series in Ap-

plied Mathematics, Vol 25, SIAM, 1976.

- [21] D.Limon, T.Alamo and E.F.Camacho, «Robust MPC control based on a contractive sequence of sets», Conference on decision and control, Maui, Hawaii USA, Decembre 2003.
- [22] D.Limon, J.M.Bravo, T.Alamo and E.F.Camacho, «Robust MPC of constrained discrete-time nonlinear systems based on uncertain evolution sets: application to a CSTR model», International symposium on computer, Glasgow, Scotland, U.K, Semptembre 2002.
- [23] A. P. de Madrid, M. Santos, S. Dormido and F. Morilla «Constrained generalized predictive control with dynamic programming», Advances in Model-Based Predictive Control, pp. 276-290. Oxford Science Publications, Oxford University Press, 1994 .
- [24] L.Magni, G.De.Nicolao, L.Lagnani and R.Scattolini, «A Stabilizing model-based predictive control algorithm for nonlinear systems», Automatica, Vol 37, pp.1351-1362, 2001.
- [25] E.H.Mamdani , “Application of Fuzzy Algorithms for Control of a simple Dynamic Plant”, In proceeding of the IEE Control and Science, Vol. 121, N° 12, pp. 1585–1588, 1974.
- [26] D.Q. Mayne, J.B. Rawlings, C.V. Rao and P.O.M. Scokaert, «Constrained model predictive control:Stability and optimality», Automatica, Vol. , N° 36, pp. 789-814, 2000.
- [27] D.Q. Mayne,and H. Michilska, «Receding horizon control of non-linear systems», IEEE Tansactions on Automatic Control, Vol. 5, N° 35, pp. 814-824, 1990
- [28] A. Mazen , «Le principe de la commande à horizon glissant », Groupe de travail sur La Commande Prédictive Non Linéaire, Paris, 2003.
- [29] E.S. Meadows, K.R. Muske and J.B.Rawlings, «Implementable model predictive control in the state space», Proceeding of the American Control Conference, Seattle, Washington Vol. 5, pp. 3699-3709, 1995.
- [30] F.Megri, M.Chemachema, M.Bahita, K.Belarbi, «Application de la Programmation dynamique floue à la commande predictive non linéaire», Conférence maghrébine sur la génie électrique, CMGE 2001, Algerie, 5-6 Novembre 2001.
- [31] L.F.Mendonça,J.M.Sousa,J.M.G.costa.Optimization problems in multivariable Fuzzy Predictive Control,International Journal of Approximate Reasoning,36,199-221,2004.
- [32] H. Michalska, and D.Q. Mayne, «Robust receding horizon control of constrained nonlinear systems», IEEE Transactions on Automatic Control, Vol 38, pp.1623-1632.
- [33] M. Minoux, «Mathematical Programming theory and Algorithms,» A Wiley Hall New York, London.
- [34] G.De Nicolao, L. Magnani, L. Magni, and R.Scattolini, «A Stabilizing receding horizon controller for nonlinear discrete time systems», proceedings of the Amecican Control Conference, Chicaginillinois, Vol 1, N° 6, pp.270-271, 2000
- [35] J.B. Rawlings and K.R. Muske, «Stability of constrained receding horizon control», IEEE Transactions on Automatic Control, Vol. 10, N° 38, pp. 1512-1516, 1993.
- [36] J.B. Rawlings, E.S. Meadows and K.R.Muske, «Nonlinear model predictive control:A tutorial», In ADCHEM’94 Proceeding, Kyoto, Japan, 1994.
- [37] J.Richalet., «Practique de la commande predictive», Edts. Hermes, 1992.

- [38] J. Richalet, A. Rault, J.L Testud, and J. Papon., «Model Predictive heuristic Control: Application to Industrial Processes», *Automatica*, Vol. 14, N° 2, pp. 413-428, 1978.
- [39] J.A.Roubos, S.Mollov, R.Babuska and H.B.Verbruggen, «Fuzzy model-based predictive control using Takagi-Sugeno models», *International Journal of Approximate Reasoning*, Vol 11, pp. 3-30, 1999
- [40] P.O.M.Scokaert, J.B.Rawlings, «Constrained linear quadratic regulation», *IEEE Transactions on automatic control*, Vol.43, N°8, pp.1163-1169, 1998.
- [41] P.O.M. Scokaert, D.Q.Mayne and J.N.Rawlings, «Suboptimal Model Predictive Control(feasibility implies Stability)», *IEEE Transactions on Automatic Control*, Vol 44, N°3, 1999.
- [42] J.M.C.Sousa, M.Setnes and U.Kaymak, «Adaptive Decision Alternatives in Fuzzy Predictive Control», *IEEE International Conference*, Vol.1, pp.698-703, 1998.
- [43] J.M.C.Sousa and U.Kaymak, «Model Predictive Control Using Fuzzy Decision Functions», *IEEE Transactions On Systems Man And Cybernetics-Part B*, Vol.31, N°1,2001.
- [44] J.M.C.Sousa, «Optimization Issues in Predictive Control with Fuzzy Objective Functions», *International Journal of Intelligent Systems*, Vol.15, pp.879-899, 2000.
- [45] M.Sugeno , “On stability of fuzzy systems expressed by fuzzy rules with singleton consequents”, *IEEE Transactions on Fuzzy set and Systems*, Vol. 7, N° 2, pp. 201–224, 1999.
- [46] T.Takagi and M.Sugeno, «Fuzzy Identification of Systems and Its Applications to Modeling and Control», *IEEE Transactions on Systems Man and Cybernetics*, Vol SMC-15,N°1,116-131,1985.

Annexe

Annexe

1. Rappels sur la stabilité des systèmes

D'une manière générale, la question de stabilité revête une importance évidente en matière de commande. Bien qu'elle n'aille pas jusqu'à constituer l'essentiel du problème de la commande, qui est un problème de performances et de précision, c'est une condition nécessaire, presque toujours impérative, pour le fonctionnement des systèmes de commande. L'importance de la stabilité s'explique par l'abondance des travaux qui lui ont été consacrés dans la littérature. Pour diminuer le risque d'implantation d'une commande et minimiser le danger des conséquences d'une synthèse insuffisante, il est donc nécessaire d'analyser le problème de stabilité. D'une manière générale, la stabilité des boucles de commande est généralement assurée dans la mesure où tous les signaux d'entrée et de sortie restent bornés.

Dans le contexte de la commande prédictive, il est impératif de vérifier que la commande générée par l'algorithme permet d'assurer la stabilité. Cette annexe vise alors à donner quelques définitions primordiales pour l'étude de la stabilité. Ces définitions sont essentiellement données autour de l'approche la plus utilisée, à savoir la méthode directe ou la deuxième méthode de Lyapunov.

2. Stabilité des systèmes dynamiques

Dans ce paragraphe, les concepts de la stabilité au sens de Lyapunov des systèmes à temps discrets seront abordés.

Prenons un système libre décrit par une équation de la forme:

$$x(k+1) = f[x(k)] \quad (22)$$

où f peut représenter une fonction linéaire ou non linéaire.

• Définition 1

Un point d'équilibre x^* est stable au sens de Lyapunov si pour chaque $R > 0$, il existe un $r > 0$ tel que:

$$\text{si } \|x(k) - x^*\| < r \Rightarrow \|x(k+l) - x^*\| < R \quad \forall l \geq 0 \quad (23)$$

• Définition 2

Un point d'équilibre x^* est asymptotiquement stable au sens de Lyapunov, s'il est stable et s'il existe un $r > 0$ quelconque tel que :

$$\|x(k) - x^*\| < r \Rightarrow \|x(k+l) - x^*\| \rightarrow 0, \text{ si } : l \rightarrow \infty \quad (24)$$

• **Définition 3**

Un point d'équilibre x^* est instable (au sens de Lyapunov) s'il n'est pas stable.

L'étude de la stabilité d'une solution arbitraire peut toujours être réduite à l'étude de la stabilité de l'origine en faisant un changement variable adéquat. Il est clair que la définition de la stabilité donnée précédemment nécessite le calcul de manière explicite de chaque solution correspondante à chacune des conditions initiales. Cependant, les théorèmes suivants permettent de remédier à ce problème.

• **Définition 4 (Fonction de Lyapunov)**

Une fonction de Lyapunov est dite candidate de Lyapunov, si et seulement si:

1. V est une fonction continue.
2. $V(x) > 0 \forall x \neq 0$, et: $V(x) = 0$ pour: $x = 0$.
3. $\Delta V(x) = 0$, pour: $x = 0$
 $\Delta V(x) = V(x(k+1)) - V(x(k))$

Le théorème fondamental de stabilité de Lyapunov peut maintenant être énoncé de la manière suivante:

2.1. **Théorème 1** (Méthode directe, dite aussi Seconde méthode de Lyapunov)

Pour un système donné, s'il existe une fonction de Lyapunov qui respecte les trois conditions énoncées précédemment, le système est alors stable. De plus, si la fonction de Lyapunov est strictement décroissante, alors la stabilité est dite asymptotiquement stable.

2.2. **Théorème 2** (Théorème d'invariance de LaSalle)

S'il existe $l > 0$ et un ensemble (région) $\Omega_l = \{x / V(x) \leq l\}$ fermé et borné possédant les propriétés suivantes:

1. $\forall x \in \Omega_l$ on a: $\Delta V \leq 0$
2. S'il existe un autre ensemble (région) tels que: $\Xi \subset \Omega_l$ et $\Xi = \{x / \Delta V(x) = 0\}$
3. Si M est le plus grand ensemble invariant, $M \subset \Xi$, alors la solution d'équilibre $\chi(x_0, t)$ converge vers M , c'est-à-dire:

$$\forall x_0 \in \Omega_l, \chi(x_0, t) \rightarrow M \text{ lorsque } t \rightarrow \infty \tag{25}$$

A travers cette méthode, il est important de mentionner deux aspects importants:

- Il n'est plus question de stabilité, mais uniquement de convergence.
- La fonction $V(x)$ n'est pas nécessairement définie positive.

Dans ce contexte, la théorie de LaSalle donne la stabilité asymptotique d'un point d'équilibre lorsque les conditions $V > 0, \forall x \neq 0$ et $V(0) = 0$ sont explicitement ajoutées. Ce théorème possède l'avantage de pouvoir s'appliquer à l'analyse de la convergence asymptotique vers un cycle comme l'illustre la figure I.5. De plus, le théorème d'invariance s'applique aussi bien aux points d'équilibre qu'aux cycles limites.

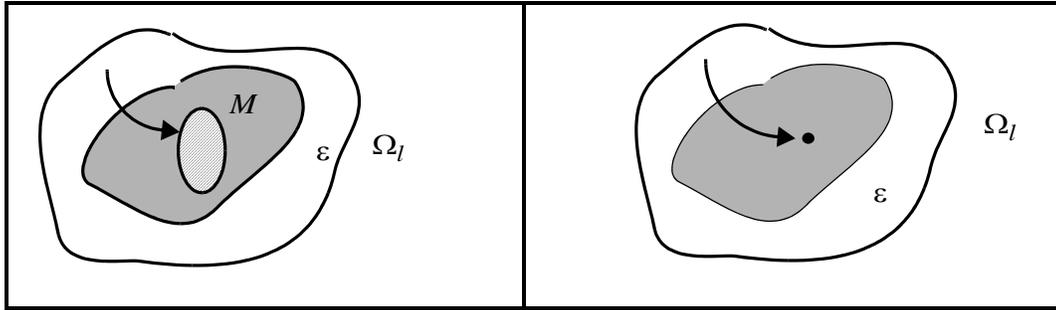


Figure I.38: Interprétation du théorème de LaSalle

- **Définition 5**

Une fonction $\alpha : R^+ \rightarrow R^+$ appartient à la classe K^0 si elle est continue et $\alpha(0) = 0 \Leftrightarrow x = 0$.

- **Définition 6**

Une fonction α appartient à la classe K^+ si $\alpha \in K^0$ et elle est non décroissante.

- **Définition 7**

Une fonction α appartient à la classe K^∞ si $\alpha \in K^+$ et $\alpha(x) \rightarrow \infty$ quand $x \rightarrow \infty$.

- **Définition 8**

Une fonction α appartient à la classe K_- si $\alpha \in K^+$ et elle est strictement croissante.

Résumé

Cette thèse présente une étude de la stabilité d'un contrôleur prédictif flou basé sur la programmation dynamique floue. L'objectif de ce contrôleur est de ramener l'état du système vers la région terminale où un contrôleur local stable sera utilisé. Ce qui nous amène vers une stratégie de commande duale. La fonction de coût est interprétée comme une fonction de Lyapunov avec un horizon de prédiction déterminé et spécifié. La stabilité de la commande prédictive floue sera garantie en imposant certaines conditions sur la région terminale, la loi de commande locale et les fonctions d'objectifs et contraintes flous. La solution au problème est donnée par la programmation dynamique floue et la technique Branch & Bound.

Mots-clefs :

Programmation dynamique, contrainte floue, objectif flou, système non linéaire, commande prédictive

Abstract

This thesis introduces a study on a stable model based fuzzy predictive controller based on fuzzy dynamic programming. The objective of the fuzzy predictive controller is to drive the state of the system to a terminal region where a local stabilizing controller is invoked, leading to a dual mode strategy. The prediction horizon is fixed and specified. The stability of the controlled system is studied by using the value function as a Lyapunov function. Guaranteed stability is obtained under conditions on the terminal region, the local control law and the membership functions of fuzzy goal and constraints therein. The solution procedure is based on dynamic programming with branch and bound.

Keywords: Dynamic programming, fuzzy constraints, fuzzy goals, nonlinear systems, predictive control.