

13/99

République Algérienne Démocratique et populaire
Ministère de l'enseignement supérieur et de la recherche scientifique

ECOLE NATIONALE POLYTECHNIQUE
D.E.R DE GENIE ELECTRIQUE ET INFORMATIQUE
DEPARTEMENT : ELECTRONIQUE



PROJET DE FIN D'ETUDE

Pour l'obtention du diplôme d'Ingénieur d'Etat en Electronique

Thème

**Application des algorithmes génétiques
dans la commande de bras manipulateurs :
Optimisation d'un contrôleur flou**

Proposé par :
Mr. C. LARBES
Mr. M.S. AIT CHEIKH

Etudié par :
SEMMAR Bedri
MOKHTARI Lakhdar

Promotion : JUIN 1999

E.N.P. 10, Avenue HASSEN BADI -EL-HARRACH- ALGER

REMERCIEMENT :

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

Nous remercions Mr. *C.LARBES* et Mr. *M.S.AIT CHEIKH*, enseignants à l'E.N.P pour leur encadrement et pour l'aide qu'ils nous ont prodigués.

Nous remercions également toute personne ayant contribué, de près ou de loin, à la réalisation de ce travail.

DEDICACES :



A ma très chère mère
A mon très cher père
A mes frères et sœurs
A tous mes amis.

Lakhdar.

A mes très chers parents
A mes frères et sœurs
A toute ma famille
A tous mes amis.

Bedri.

موضوع هذه الدراسة يتعلق بتطبيق الخوارزميات الوراثة
كطريقة فعالة للبحث عن قيم ضمن مجال معين يسمح بالوصول على أحسن إجابة
لتبراع آليه. تقنية التحكم المستعملة هي طريقة المنطق الغامض التي تسمح
بالتحكم في جهاز لاجلبي متعدد المتغيرات بقواعد منطوية سهلة
التطبيق.

هذا العمل يمثل خطوة نحو استعمال الخوارزميات الوراثة للوصول على
تكم بواسطة المنطق الغامض لأي جهاز دون معرفة المعادلات التي تحكمه.

RESUME :

L'objet de cette étude est d'appliquer les algorithmes génétiques pour optimiser la réponse d'un bras manipulateur pour un type de commande donné.

Dans ce travail, nous avons utilisé la logique floue pour la commande d'un système multivariable couplé et non linéaire (bras de robot à 2 degrés de liberté). L'application de ce type de commande n'exige pas la connaissance du modèle dynamique du bras à commander et permet son contrôle avec des règles simples et faciles à mettre en œuvre.

Ce travail constitue donc une étape primaire dans l'utilisation des algorithmes génétiques pour l'optimisation des paramètres du contrôleur par logique floue utilisé.

Mots clés: Algorithme génétique(AG), bras manipulateur, chromosome, croisement, déffuzzification, facteur d'appartenance, fonction d'appartenance, fuzzification, gain de sortie du régulateur, gène, matrice de transformation homogène, modèle dynamique, mutation, opérateur génétique, population, règles d'inférence, régulateur par logique floue, variable articulaire, variable linguistique.

ABSTRACT :

The object of this thesis is to apply the Genetic Algorithms to optimise the response of a manipulator controlled by a specified technic.

In this work, we have used the fuzzy logic to control a multivariable, coupled and nonlinear system with simple rules of control.

This work constitue a primary step in GAs using for fuzzy control optimization.

Key words: Appartenance factor, appartenance function, articular variable, chromosom, crossover, deffuzzification, dynamic model, fuzzification, fuzzy logic controller, gene, genetic algorithm, genetic operator, homogene transform matrix, inference rules, linguistic variable, manipulator, mutation, output controller gain, population.

SOMMAIRE

INTRODUCTION GENERALE

CHAPITRE I : Généralités sur les algorithmes génétiques	1
Introduction	2
I.1. Algorithmes génétiques : Généralités	3
I.2. AsG : Comment fonctionnent-ils ?	6
I.3. Dans quel but travaillent-ils ?	8
I.4. Différents sujets	16
I.5. Binaire ou flottant ?	26
I.6. Choix local	28
CHAPITRE II : Exemple d'application	30
Exemple d'application	31
Algorithme de recherche	32
CHAPITRE III : Généralités sur la robotique	33
III.1. Définitions et classification des systèmes robotiques	34
III.2. Modélisation cinématique	38
III.3. Génération des trajectoires	50
III.4. Modélisation dynamique	54

CHAPITRE IV : Généralités sur la logique floue	60
IV.1. Introduction	35
IV.2. Variables linguistiques	62
IV.3. Définition des variables linguistiques	63
IV.4. Fonction d'appartenance	63
IV.5. Dédutions floues (inférences)	63
IV.6. Opérateurs de la logique floue	64
IV.7. Réglage par logique floue	66
CHAPITRE V : Commande par logique floue des bras manipulateurs avec optimisation par AsG	71
V.1. Introduction	72
V.2. Choix d'un contrôleur	72
V.3. Schéma de contrôle	72
V.4. Algorithme de contrôle flou	73
V.5. Application à la commande du bras à un seul degré de liberté	75
V.6. Simulation et interprétation des résultats	78
V.7. Application des algorithmes génétiques	98
CONCLUSION GENERALE	102
ANNEXE	
BIBLIOGRAPHIE	

المدرسة الوطنية المتعددة التقنيات
BIBLIOTHEQUE — المكتبة
Ecole Nationale Polytechnique

Introduction générale

INTRODUCTION GENERALE

Les algorithmes génétiques commencent à devenir un outil très puissant d'optimisation et de recherche, cela est dû principalement à l'aspect organisé de la recherche par laquelle ils procèdent. Il suffit, pour un algorithme génétique, de connaître le domaine de recherche dans laquelle l'opérateur compte trouver l'optimum de la fonction à laquelle il s'intéresse (fonction d'adéquation).

L'algorithme est ensuite lancé pour une population initiale de solutions remplissant l'unique condition d'appartenir à l'intervalle de recherche imposé par l'opérateur.

Dans notre étude, nous appliquons, en premier lieu, la commande par logique floue à deux types de bras de robot : bras à un degré de liberté, et bras PUMA à deux degrés de liberté (articulations rotatives).

Le contrôleur flou reste capable d'assurer de bonnes réponses au système commandé dans une plage assez large de variation de ses paramètres. Nous nous intéressons à l'influence de la valeur de son gain de sortie et à l'optimisation de cette valeur.

Notre travail s'articule autour des chapitres suivants:

Premier chapitre

C'est une introduction théorique des algorithmes génétiques incluant les principales notions de leur fonctionnement et leur application.

Deuxième chapitre

Il englobe un exemple assez simple consistant à la minimisation du module d'une impédance en fonction de la pulsation du courant la traversant: il permet d'avoir une idée sur la capacité de recherche et la vitesse des AsG dans un domaine donné.

Troisième chapitre

C'est une introduction générale à la robotique incluant la cinématique directe et inverse faisant apparaître la notion de transformation homogène. Nous présentons ensuite la théorie liée à la modélisation dynamique des bras manipulateurs permettant d'écrire les équations dynamiques les représentant; nous utilisons pour cela le formalisme d'EULER-LAGRANGE.

Enfin, nous introduisons les principales notions de génération des trajectoires pour les bras de robots; les principales règles découlent de la continuité des courbes de vitesse et de position.

Quatrième chapitre

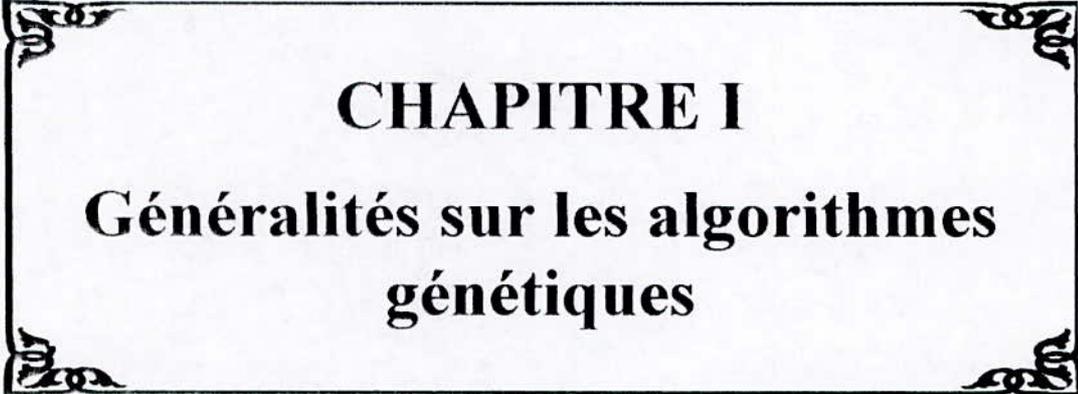
Dans ce chapitre, nous parlons des parties principales de la théorie de la logique floue permettant l'implémentation d'un régulateur utilisant cette logique. Nous présentons également les différentes étapes de traitement des variables d'entrée du régulateur jusqu'à sa sortie ou l'on obtient la commande.

Cinquième chapitre

Ce chapitre concerne la partie commande des deux bras de robot déjà cités, nous appliquons le contrôle par logique floue au deux bras de robot, ensuite nous faisons varier la valeur du gain de sortie du régulateur pour voir son influence sur la réponse de chaque bras.

A la fin de ce chapitre nous faisons l'implémentation d'un algorithme génétique permettant la recherche d'une valeur optimale du gain de sortie du régulateur utilisé, le domaine de recherche est estimé à partir des paramètres du bras à commander (la masse et la longueur des segments).

Enfin nous achevons ce mémoire par une conclusion générale.



CHAPITRE I

Généralités sur les algorithmes génétiques

Introduction [MAN 96], [BAC 97] :

Les origines des Algorithmes Génétiques (AsG) remontent aux années 50 avec les premiers travaux de Bremermann, Friedberg, Box et autres. Ce domaine restait relativement inconnu par la communauté scientifique durant presque trois décennies.

Les AsG ont été introduit par J. H. Holland dans les années 70, et par la suite étudiés par De Jong, Goldberg, Eshelman, Schaffer et autres.

Les travaux de J. H. Holland ont confirmé la contribution considérable des (AsG) dans les applications scientifiques. Depuis, les travaux de recherche dans ce champ ont augmenté d'une manière exponentielle. Cependant, le problème major dans l'utilisation des AsG est le temps élevé que prend le processus de calcul et la nature aléatoire intrinsèque de la recherche de la solution optimale. Malgré ça, les AsG ont atteint un stade de maturité avancé, et ceux grâce aux efforts fournis dans ces quelques dernières années par les académies et les chercheurs dans le monde entier. Les problèmes qui ont été considérés à l'époque difficile ou même impossibles à résoudre, ont trouvé aujourd'hui leur issu en utilisant les AsG.

De plus, les AsG ne sont pas gérés par des formules mathématiques standards dans l'élaboration d'une solution d'une génération à une autre, comme c'est le cas des procédures d'optimisation traditionnelles. EN effet, les AsG sont très différents dans ce contexte. Ils ne sont autre que des processus aléatoires, discrets et non linéaires.

L'optimum est le produit final d'un ensemble de bons individus obtenus à partir des générations précédentes où les attribues du meilleur individu tendent à paraître précocement dans la génération suivante.

I.1 ALGORITHMES GENETIQUES :GENERALITES

I.1.1 Introduction

Les algorithmes génétiques appartiennent à une classe dite « classe d'intelligence artificielle » ; ils sont inspirés de l'évolution des caractéristiques des individus d'une génération à une autre.

Ils sont utilisés pour des problèmes assez compliqués d'optimisation en faisant combiner les membres d'une population données (solutions potentielles) ; ces solutions sont plus ou moins « mauvaises ».

Les AsG consistent en la manipulation de ces solutions pour obtenir d'autres dont les meilleures sont celles minimisant ou maximisant une fonction dite « fonction d'adéquation (ou d'évaluation) » représentant l'environnement.

I.1.2 Structure et opérateurs

La structure d'un algorithme génétique est donnée par ce qui suit :

```

DEBUT
  T=0
  Initialiser P(t)
  Evaluer P(t)
  Tant que (condition non vérifiée) faire
  Début
    T :=t+1
    Sélection de P(t) de P(t-1)
    Altérer P(t)
    Evaluer P(t)
  Fin
Fin

```

Figure I.1 : structure d'un algorithme génétique.

$P(t)$ est la population de solutions à l'instant t . Les solutions sont aussi dites « vecteurs » ou « chromosomes ».

$P(t)=\{x1t, \dots, xnt\}$, n :taille de la population de solutions. Une nouvelle population est produite à $t+1$ à partir de la précédente et dont les individus sont « meilleurs ».

Deux opérateurs sont utilisés à chaque itération :opérateur de croisement et ce_lui de mutation .On va les présenter des avec un exemple :

Exemple :

Soient les deux vecteurs :

$x1^t=(a1,b1,c1,d1,e1)$;

$$x2^t=(a2,b2,c2,d2,e2).$$

Si on leurs applique l'opérateur de croisement après la seconde composante(gène) on obtient les vecteurs correspondant à l'instant t+1 :

$$x1^{t+1}=(a1,b1,c2,d2,e2) ;$$

$$x2^{t+1}=(a2,b2,c1,d1,e1) ;$$

les premiers correspondant à l'instant t+1 .

L'application de l'opérateur de mutation consiste généralement a altérer un gène choisi d'un vecteur donné. Il peut y avoir altération de plusieurs gènes d'une manière aléatoire et de probabilité égale au « taux de mutation »,un des paramètres de l'AG.

Un algorithme génétique doit avoir :

- Une représentation «génétique» des solutions potentielles du problème.
- Un moyen pour initialiser la population de solutions.
- Une fonction d'adéquation (jouant le rôle de l'environnement).
- Des opérateurs génétiques altérant la composition des « chromosomes » descendants.
- Valeurs des différents paramètres utilisés (taille de la population, probabilités d'utilisation des opérateurs,...) .

Exemple :

On se propose de maximiser la fonction d'adéquation $f(x)=x.\sin(10.\pi.x)$ dans l'intervalle [-1,2]. Les solutions recherchées seront écrites sous forme de vecteurs

binaires. Pour une précision de 6 chiffres après la virgule on a affaire à des vecteurs d'une longueur de 22 bits. x^b : valeur de x en binaire.

a- Population initiale

On se donne des vecteurs de 22 bits chacun ,tous les bits seront choisis Aléatoirement.

b- Fonction d'évaluation

Eval(v)=f(x).v : sera la représentation de la valeur de x.

$$v1=(1000101110110101000111)=0.637197.$$

$$v2=(0000001110000000010000)=-0.958973.$$

$$v3=(1110000000111111000101)=1.627888.$$

$$f(x1)=eval(v1)=1.586345.$$

$$f(x2)=eval(v2)=0.078878.$$

$$f(x3)=eval(v3)=2.250650.$$

Il est clair que le meilleur chromosome est v3.

c- Opérateurs génétiques

La mutation cause l'altération d'un ou plusieurs bits d'un chromosome avec une probabilité égale au taux de mutation. Si on choisit le 5^{ème} gène pour subir la mutation on aura :

$$v3'=(1110100000111111000101).$$

$$x3'=1.721638 ;$$

$$f(x3')=-0.082257.$$

Si le 10^{ème} gène est soumis à la mutation on aura :

$v3''=(111000000111111000101)$ correspondant à $x3''=1.630818$ ayant pour Valeur de fonction d'adéquation $f(x3'')=2.343555$. Ainsi $v3''$ est meilleur que $v3$.

Le croisement des deux vecteurs $v2$ et $v3$ après 5 gènes :

$$v2=(00000/01110000000010000).$$

$$v3=(11100/00000111111000101).$$

nous donne :

$$v2'=(00000/00000111111000101).$$

$$v3'=(11100/01110000000010000).$$

$$\text{eval}(v3')=2.459245.$$

d- Paramètres

La taille de la population est de 50.

$p_c=0.25$ (probabilité de croisement).

$p_m=0.01$ (probabilité de mutation).

Après 150 générations on obtient le vecteur $v_{\max}=(1111001101000100000101)$ correspondant à $x_{\max}=1.850773$.

Sans perdre de généralité on vient de voir comment on peut utiliser les AsG pour optimiser une fonction donnée sans rentrer dans le détail de leur fonctionnement et c'est ce que nous allons voir par la suite.

I.2 AsG : COMMENT FONCTIONNENT-ILS ?

Pour maximiser (ou minimiser) une fonction :

$f(x_1, \dots, x_n)$ telle que : $x_i \in [a_i, b_i]$, $i : 1..n$. L'intervalle $[a_i, b_i] \subseteq R$.

On peut montrer aisément qu'il faut diviser l'intervalle $D_i = [a_i, b_i]$ par $(b_i - a_i) * 10^{pr}$ segments égaux pour obtenir une représentation binaire d'un nombre (pr est la précision).

On cherche le plus petit entier m vérifiant :

$$(b_i - a_i) * 10^{pr} \leq 2^m - 1. \quad (I.2.1)$$

On aura ainsi :

$$x_i = a_i + \text{decimale}(1..01) * (b_i - a_i) : (2^m - 1). \quad (I.2.2)$$

Decimale(1..01) donne la valeur du vecteur entre parenthèses en décimale.

Chaque chromosome sera un vecteur binaire de longueur :

$m = \sum m_i$, et donc la juxtaposition de tous ces vecteurs binaires donne un vecteur

représentant la solution (x_1, \dots, x_n) de notre problème.

Ainsi, on construit la génération de solutions potentielles aléatoirement ou suivant un critère optimal si on a un, l'algorithme génétique consiste en l'altération des chromosomes par les deux opérateurs déjà cités l'évaluation sera faite en utilisant la fonction :

$$\text{eval}(v_i) = f(x_1, \dots, x_n). \quad (I.2.3)$$

Souvent, on arrête l'algorithme après un certain nombre d'itérations dépendant de la vitesse et des critères imposés.

La sélection de la nouvelle population se fait en respectant la distribution de probabilité suivant la valeur de la capacité (fitness).

- Calcul de la valeur de la fonction d'adéquation pour chaque chromosome, ($i=1, \dots, \text{taille de la population}$).

- Calcul de la fonction d'adéquation totale : $F = \sum_{i=1}^{\text{taille-pop}} F_i$.

- Calcul de la probabilité de choix pour chaque chromosome :

$$p_i = \text{eval}(v_i) / F. \quad (I.2.4)$$

- Calcul d'une probabilité cumulée du chromosome v_i .

$$q_i = \sum_{j=1}^i p_j. \tag{1.2.5}$$

Pour sélectionner un chromosome pour une nouvelle génération, on procède comme suit :

- Générer un nombre aléatoire de l'intervalle [0..1] ;
- Si $r < q_i$ alors on prend (v_i) , sinon on prend (v_i) tel que $q_{i-1} < r \leq q_i$.

Ce choix se fait à chaque instant $t = 0, \dots, \text{taille} - \text{pop}$.

Ainsi, des chromosomes seront plus souvent choisis que d'autres, les meilleurs et les moyens survivent, les mauvais meurent.

La probabilité de croisement p_c nous donne le nombre de chromosomes soumis Au croisement :

$$r_c = p_c \cdot \text{taille} - \text{pop} . \tag{1.2.6}$$

Pour chaque chromosome de la nouvelle population on procède comme suit :

- Générer un nombre aléatoire de l'intervalle [0..1],
- Choisir v_i si $r < p_c$.

Ensuite, on accouple les chromosomes aléatoirement, et pour chaque couple on génère aléatoirement un nombre entier pos de l'intervalle [0..n-1] (n longueur totale d'un vecteur). Le nombre pos indiquera la position du croisement. Les deux chromosomes :

$$\begin{aligned} &(b_1 b_2 \dots b_{pos} b_{pos+1} \dots b_m) \\ &(c_1 c_2 \dots c_{pos} c_{pos+1} \dots c_m) \end{aligned}$$

sont remplacés par :

$$\begin{aligned} &(b_1 b_2 \dots b_{pos} c_{pos+1} \dots c_m) \\ &(c_1 c_2 \dots c_{pos} b_{pos+1} \dots b_m) \end{aligned}$$

Pour la mutation, la procédure est analogue :

Pour une probabilité de mutation p_m le nombre de bits mutés est :

$$p_m * m * \text{taille} - \text{pop} .$$

Pour chaque gène, après croisement, on procède comme suit :

- On génère un nombre aléatoire r de l'intervalle [0..1].
- Si r est inférieur à p_m on inverse le bit.

Ainsi, on obtient la nouvelle génération sur laquelle on applique le même procédé d'une manière répétitive jusqu'à l'obtention d'une certaine stabilisation de la fonction d'évaluation (maximum global).

I.3 DANS QUEL BUT TRAVAILLENT ILS ?

Les AsG sont fondés sur la représentation binaire des solutions et la notion de schéma.

Un schéma est construit par l'introduction du symbole d'indifférence (*) dans l'alphabet des gènes. Un schéma représente tous les individus lui étant égaux dans les positions où il n'y a pas le symbole (*).

Exemple :1

(*11110) est égal aux deux chromosomes (111110) et (011110).

Chaque chromosome représente 2^r chromosome (r est le nombre de symbole (*)).

Chaque chromosome est représenté par 2^m schémas (m est la longueur du chromosome).

Exemple :2

(01) est représenté par (01) (0*) (*1) et (**).

Pour une population de taille n le nombre de différents schémas varie entre 2^m et $n * 2^m$ qui ont différentes caractéristiques.

I.3.1 Propriétés d'un schéma :

On définit deux propriétés principales d'un schéma :

Soit S_1 un schéma : $S_1 = (** * 001 ** 10)$.

- L'ordre : l'ordre du schéma S_1 , noté $o(S_1)$, est la longueur du schéma moins le nombre de conditions indifférentes, $o(S_1) = 10 - 5 = 5$. La notion de l'ordre est utile pour calculer la probabilité de survie d'un schéma pour la mutation, il définit la spécialité d'un schéma.

- La longueur définissante : la longueur définissante d'un schéma S, noté $\delta(S)$, est la différence entre la première et la dernière positions fixes du chromosome. Pour le schéma S_1 : $\delta(S_1) = 10 - 4 = 6$. La longueur définissante d'un schéma est utile pour calculer la probabilité de survie d'un schéma après croisement, elle définit aussi le degré avec lequel l'information contenue dans le schéma est compacte.

I.3.2 Processus d'évolution :

On a déjà donné le processus d'évolution des AsG qui se résume en quatre étapes consécutivement répétées.

- 1^{ère} étape : faire avancer le temps d'une unité $t \leftarrow t + 1$.
- 2^{ème} étape : sélectionner $P(t)$ de $P(t-1)$.
- 3^{ème} étape : recombinaison $P(t)$.
- 4^{ème} étape : évaluer $P(t)$.

Le phénomène principal a lieu dans les 2^{ème} et 3^{ème} étapes (sélection et recombinaison). Discutons l'effet de ces deux étapes sur le nombre espéré des schémas représentés par la population.

On se donne une population de taille 20 (taille-pop=20), et une longueur m du chromosome égale à 33 ($m=33$). La population est constituée des chromosomes suivants :

$v_1 = (100110100000001111111010011011111)$
 $v_2 = (111000100100110111001010100011010)$
 $v_3 = (000010000011001000001010111011101)$
 $v_4 = (100011000101101001111000001110010)$
 $v_5 = (000111011001010011010111111000101)$
 $v_6 = (00010100001001010100101011111011)$
 $v_7 = (00100010000011010111101101111011)$
 $v_8 = (100001100001110100010110101100111)$
 $v_9 = (010000000101100010110000001111100)$
 $v_{10} = (000001111000110000011010000111011)$
 $v_{11} = (011001111110110101100001101111000)$
 $v_{12} = (110100010111101101000101010000000)$
 $v_{13} = (111011111010001000110000001000110)$
 $v_{14} = (010010011000001010100111100101001)$
 $v_{15} = (111011101101110000100011111011110)$
 $v_{16} = (110011110000011111100001101001011)$
 $v_{17} = (011010111111001111010001101111101)$
 $v_{18} = (011101000000001110100111110101101)$
 $v_{19} = (000101010011111111110000110001100)$
 $v_{20} = (101110010110011110011000101111110)$

Notons par $\xi(S, t)$ le nombre de chromosomes correspondant au schéma S à l'instant t .

Le schéma $S_0 = (****111*****)$, représente trois chromosomes : v_{13} , v_{15} , v_{16} .

Notons que l'ordre du schéma S_0 est $o(S_0) = 33 - 30 = 3$ et sa longueur définissante est : $\xi(S_0) = 7 - 5 = 2$.

Une autre propriété de S_0 est la valeur de la fonction d'adéquation à l'instant t , $eval(S, t)$. Elle est définie comme étant la moyenne des fonctions d'adéquation de tous les chromosomes qu'elle représente.

Supposons qu'on a p chromosomes $\{v_{i1}, v_{i2}, \dots, v_{ip}\}$ dans la population représentée par le schéma S à l'instant t . Donc :

$$eval(S, t) = \sum_{j=1}^p eval(v_j) / p. \quad (I.3.1)$$

Durant l'étape de sélection, une population intermédiaire de même taille prend naissance. Chaque individu est copié zéro, une, ou plusieurs fois suivant sa fonction d'adéquation. Un chromosome v_i a une probabilité de sélection donnée par :

$$p_i = eval(v_i) / F(t). \quad (I.3.2)$$

avec

$$F(t) = \sum_{i=1}^{20} eval(v_i). \quad (I.3.2)$$

qui représente la fonction d'adéquation totale .

Après l'étape de sélection nous nous attendons à avoir $\xi(S, t+1)$ individus représentés par le schéma S , puisque :

1. Pour un individu moyen la probabilité du choix est :

$$eval(S, t) / F(t). \quad (1.3.3)$$

2. Le nombre d'individus représentés par S est $\xi(S, t)$.
3. Le nombre d'individus parmi lequel on effectue notre choix est: taille-pop .
Il obtient alors:

$$\xi(S, t+1) = \xi(S, t) \cdot taille - pop \cdot eval(S, t) / F(t). \quad (1.3.4)$$

La moyenne de la fonction d'adéquation de la population est :

$$\overline{F(t)} = F(t) / taille - pop. \quad (1.3.5)$$

On peut alors écrire :

$$\xi(S, t+1) = \xi(s, t) \cdot eval(S, t) / \overline{F(t)}. \quad (1.3.6)$$

la croissance du nombre d'individus dans une population est proportionnelle au rapport $eval(S, t) / \overline{F(t)}$. Cela signifie qu'un schéma au dessus de la moyenne reçoit un nombre d'individus croissant pour la prochaine génération, le contraire a lieu pour un schéma au dessous de la moyenne. Un schéma moyen reçoit le même nombre d'individus à chaque génération. Si on suppose que schéma S est au dessus de la moyenne avec un pourcentage, i.e. :

$$eval(S, t) = \overline{F(t)} + \varepsilon \cdot \overline{F(t)}. \quad (1.3.7)$$

on écrit alors :

$$\xi(S, t) = \xi(S, 0) \cdot (1 + \varepsilon)^t. \quad (1.3.8)$$

avec :

$$\varepsilon = (eval(S, t) - \overline{F(t)}) / \overline{F(t)}. \quad (1.3.9)$$

Pour un schéma au dessus de la moyenne $\varepsilon > 0$.
Pour un schéma au dessous de la moyenne $\varepsilon < 0$.

Le nombre d'individus reçus par un schéma est une fonction exponentielle du temps (génération d'après).

L'équation (1.3. 7) est dite 'équation de croissance reproductive d'un schéma'.

Retournons à notre exemple du schéma S_0 qui représente v_{13} , v_{15} et v_{16} à l'instant t , la fonction d'adéquation sera :

$$eval(S_0, t) = (27.316702 + 30.060205 + 23.867227) / 3 = 27.081378. \quad (1.3.10)$$

La moyenne de la fonction d'adéquation pour toute la population est :

$$\overline{F(t)} = \sum_{i=1}^{20} eval(v_i) / taille - pop = 387.776822 / 20 = 19.388841. \quad (1.3.11)$$

on a aussi :

$$1 + \varepsilon = eval(S_0, t) / \overline{F(t)} = 1.396751. \quad (1.3.12)$$

Alors, le schéma S_0 reçoit un nombre de chromosomes croissant exponentiellement avec le temps. Si S_0 reste constant, à l'instant $t + 1$ S_0 reçoit $3 \times 1.396751 = 4.19$ individus (i.e. 4 ou 5 individus), à $t + 2$, il reçoit $3 \times (1.396751)^2 = 5.85$ individus (i.e. 5 ou 6 individus) etc.

Cependant, la sélection seule n'introduit aucun nouveau individu (solution potentielle); elle copie seulement quelques chromosomes de la population intermédiaire.

L'introduction de nouveaux individus est faite par la recombinaison en utilisant les deux opérateurs génétiques, à savoir : le croisement et la mutation.

a- Le croisement

Après une génération (temps $t + 1$) on obtient une nouvelle population. Soit l'élément v_{18}' de cette population :

$$v_{18}' = (11101111101000100011/0000001000110).$$

qui peut être représenté par 2^{33} schémas, en particulier par les deux suivants:

$$S_0 = (****111*****) \text{ et } S_1 = (111*****10).$$

S'il est croisé après la 20^{ème} position (pos = 1) avec le chromosome :

$$v_{13}' = (00010100001001010100/1010111111011).$$

v_{13}' et v_{18}' produisent :

$$v_{18}'' = (11101111101000100011/1010111111011) \\ v_{13}'' = (00010100001001010100/0000001000110).$$

On aura ‘destruction’ du schéma S_1 , il n’est représenté ni par v_{13} ni par v_{18} .

La longueur définissante joue un rôle important, $\delta(S_1) = 32$, le croisement se fait par le choix d’un gène parmi $m-1 = 33 - 1 = 32$. La probabilité de destruction d’un schéma est ainsi donnée par :

$$p_d(S) = \frac{\delta(S)}{m-1} \tag{1.3.13}$$

qui est d’autant plus grande que la longueur définissante de S est élevée. la probabilité de survie sera donc :

$$p_s(S) = 1 - \frac{\delta(S)}{m-1} \tag{1.3.14}$$

Dans notre exemple :

$$p_d(S_0) = 2/32, p_s(S_0) = 30/32, p_d(S_1) = 32/32 = 1, p_s(S_1) = 0.$$

En pratique, la probabilité de croisement est p_c , ainsi :

$$p_s(S) = 1 - p_c \cdot \frac{\delta(S)}{m-1} \tag{1.3.15}$$

Il faut noter que même si le croisement se fait entre les positions (gènes) fixes, le schéma a une chance de survivre (v_{13} et v_{18}) commençant par 111 et se terminant par 10. La formule de la probabilité de croisement devient alors :

$$p_c(S) \geq 1 - p_c \cdot \frac{\delta(S)}{m-1} \tag{1.3.16}$$

Ainsi , l’effet de la sélection et de croisement nous donne une autre forme :

$$\xi(S, t+1) \geq \xi(S, t) \cdot \overline{eval(S, t) / F(t)} \cdot \left[1 - p_c \cdot \frac{\delta(S)}{m-1} \right] \tag{1.3.17}$$

Pour un schéma de petite longueur définissante la croissance est exponentielle avec le temps. Pour notre schéma S_0 :

$$eval(S_0, t) / \overline{F(t)} \left[1 - p_c \cdot \frac{\delta(S)}{m-1} \right] = 1.396751 \cdot 0.984375 = 1.374927 \tag{1.3.18}$$

Cela veut dire que le schéma S_0 reçoit un nombre exponentiellement croissant de chromosomes dans la génération d’après: à l’instant $t+1$, $\xi(S, 1) = 3 \times 1.374297$ donnant 4.12

chromosomes (au lieu de 4.19 pour la sélection seule), à l'instant t+1, $\xi(S, 2) = 3 \times (1.374297)^2$ donnant une valeur de 5.67 chromosomes (au lieu de 5.85 pour la sélection seule).

Le deuxième opérateur à considérer est la mutation.

b- La mutation

Pour qu'un schéma survive à une mutation, il faut que cette dernière ne touche pas aux positions fixes du schéma en question. Soient le chromosome v_{19} de la deuxième génération de notre exemple :

$$V_{19} = (111011101101110000100011111011110).$$

et le schéma S_0 :

$$S_0 = (**** 111*****).$$

S_0 serait détruit par la mutation si la position du gène muté est entre 5 et 7 pour v_{19} . Cela dépend des positions fixes du schéma. Ainsi la probabilité de survie d'un schéma à une mutation est donnée par:

$$p_s(S) = (1 - p_m)^{o(S)}. \tag{1.3.19}$$

avec p_m la probabilité de mutation et $o(S)$ l'ordre du schéma. Pour $p_m \ll 1$, cette probabilité peut être approximée par :

$$p_s(S) = 1 - o(S) \times p_m. \tag{1.3.20}$$

Pour le schéma S_0 et pour une probabilité $p_m = 0.01$, $p_s(S_0)$ sera égale à :

$$p_s(S_0) = 1 - 3 \times 0.01 = 0.97.$$

La sélection, le croisement et la mutation donnent l'équation du schéma reproductrice :

$$\xi(S, t+1) \geq \xi(S, t) \times \overline{eval(S, t) / F(t)} \left[1 - p_c \times \frac{\delta(S)}{m-1} - o(S) \times p_m \right]. \tag{1.3.21}$$

Vue cette équation et ces précédentes (1.3.7) et (1.3.18) on note que la fonction d'adéquation doit être positive, ce qui impose, parfois, la nécessité de faire une translation des valeurs de la fonction à optimiser (si cette dernière prend des valeurs négatives).

Revenons à notre exemple du schéma S_0 :

$$\overline{eval(S, t) / F(t)} \times \left[1 - p_c \times \frac{\delta(S_0)}{m-1} - o(S) \times p_m \right] = 1.396751 \times .954375 = 1.333024.$$

Cela signifie que S_0 reçoit un nombre de chromosomes croissant exponentiellement dans la génération d'après : à l'instant t+1, on a $3 \times 1.333024 = 4.00$ chromosomes (4.19 pour la sélection

seule et 4.12 pour la sélection et le croisement), à l'instant $t+2$, on a: $3 \times (1.333024)^2 = 5.33$ (aussi, moins que 5.85 et 5.67).

En résumé :

- La sélection n'apporte aucune nouveauté sur les schémas existants .
- Le croisement permet un échange structuré (pourtant aléatoire) d'information .
- La mutation permet une grande variabilité dans la population .

L'effet cumulé de ces opérations n'est pas significatif (comme on a vu pour le schéma S_0) pour un schéma court et d'ordre réduit. Ce qui mène au théorème du schéma suivant.

Théorème du schéma :

Un schéma au dessus de la moyenne, court et d'ordre réduit reçoit un nombre exponentiellement croissant avec le temps .

Ainsi, ces schémas sont utilisés pour explorer la population en échangeant les informations grâce au croisement.

Hypothèse du bloc de construction

Un AG cherche une performance presque optimale à travers la juxtaposition de schémas courts, d'ordre réduit et de haute performance, dite ' bloc de construction '.

Dans les dernières années, quelques AsG ont été en faveur de cette hypothèse qui renvoie au fait que le problème de codage est critique dans le sens d'obtenir des blocs de construction.

Dans une population de taille $taille-pop$, d'individus de longueur m , on peut avoir au minimum 2^m schémas et au maximum $2^{taille-pop}$.

Pour quelques problèmes, cette hypothèse est non vérifiée.

Considérons des schémas de longueur totale de 11 positions, soient :

$$S_1 = (111*****)$$

$$S_2 = (*****11)$$

leur combinaison donne :

$$S_3 = (111*****11).$$

qui est moins adéquat que :

$$S_4 = (000*****00).$$

Supposons que le chromosome optimal est :

$$S_0 = (1111111111).$$

L'algorithme génétique aura des difficultés pour converger vers S_0 , par contre, il peut converger vers un autre point comme :

$$S_5 = (0001111100) .$$

Ce phénomène est appelé 'phénomène de déception' : quelques blocs de construction (schémas courts et d'ordre réduit) peuvent conduire à la convergence de l'algorithme génétique vers un point sous-optimal (optimum local).

Le phénomène de déception est fortement lié à l'interaction entre les gènes de positions différentes d'un chromosome. Ainsi, la fonction d'adéquation d'un gène dépend des autres gènes.

I.4 Différents sujets

La théorie des algorithmes génétiques fournit quelques informations pour la recherche d'un point optimal. Malheureusement, des applications pratiques ne suivent pas la théorie des AsG pour des raisons telles que :

- Le codage du problème fait que l'AG travaille dans un espace autre que celui du problème lui-même.
- La limitations des itérations et la taille de la population.

Une des implications de ces observations est l'incapacité des AsG, sous certaines conditions, de trouver une solution optimale. Un tel échec est causé par la convergence prématurée vers un optimum local. Lorsque l'AG converge assez rapidement, l'information utile dans une partie de la population est perdue.

Eshelman et Schaffer discutent des stratégies contre cette convergence :

- Stratégie de couplage appelée "incest prevention".
- Utilisation d'un croisement uniforme.
- Détection des solutions doubles.

L'étude dans ce domaine revient à :

- La grandeur et le type des erreurs introduites par le mécanisme d'échantillonnage (ou du choix).
- Les caractéristiques de la fonction elle-même.

Il y a deux paramètres intervenant dans la recherche génétique, à savoir : la diversité et la pression sélective qui sont fortement liés l'un à l'autre : si l'un est important l'autre devient forcément faible. La pression sélective peut causer la convergence indésirable. Il est donc commode d'établir un équilibre entre ces deux paramètres.

Des modifications ont été introduites par Dejong dans la manière avec laquelle se fait la sélection, ces modifications sont :

1. Le modèle 'elitist' : consiste à renforcer la préservation des meilleurs chromosomes.
2. Le modèle de la valeur recherchée: consiste à attribuer à chaque valeur choisie une valeur $f(v)/\bar{f}$ qui diminue de 0.5 si le chromosome est choisi pour le croisement et de 1 s'il est choisi pour la mutation ; si la valeur devient négative le chromosome n'est plus choisi.
3. La troisième modification consiste à combiner les premières méthodes.

D'autres méthodes sont basées sur le fait que le choix est proportionnel à un poids attribuer au chromosome, dit 'ordre'.

L'existence de super-individus peut causer la convergence rapide faisant perdre l'information utile existante sans les autres individus.

Concernant cette méthode, une valeur Max définie par l'opérateur, comme borne supérieure du nombre espéré d'individus, et une courbe linéaire était prise à travers Max de sorte que la surface de la courbe soit égale à la taille de la population.

Ainsi, on connaît la différence des nombres de descendants espérés des individus adjacents.

Exemple :

Max =20 et taille-pop=50, la différence entre les nombres espérés de descendants des individus adjacents est 0.04.

Une autre possibilité est de prendre un paramètre q et définir une fonction linéaire :

$$prob(rank) = q - (rank - 1)r. \quad (1.4.1)$$

ou une fonction non linéaire :

$$prob(rank) = q(1 - q)^{rank-1}. \quad (1.4. 2)$$

Les deux fonctions donnent la probabilité du choix à la position $rank$ d'un individu ($rank = 1$: best (meilleur) individu ; $rank = taille-pop$: mauvais individu).

Les deux formes permettent à l'opérateur d'influencer la pression sélective de l'algorithme. Dans le cas de fonction linéaire :

$$\sum_{i=1}^{taille-pop} prob(i) = 1. \quad (1.4. 3)$$

ce qui donne :

$$q = r(taille - pop - 1)/2 + 1/taille - pop. \quad (1.4. 4)$$

Si $r = 0$ on a $q = 1/taille-pop$, alors, la pression sélective n'existe pas : tous les individus ont la même probabilité de sélection.

Si $q - (taille-pop)r = 0$, on aura :

$$r/(taille - pop(taille - pop - 1)), \text{ et } q = 2/taille - pop. \quad (1.4. 5)$$

qui donne une pression sélective maximale.

Ainsi, cette pression est contrôlée par le paramètre q appartenant à l'intervalle

$$[1/taille-pop, 2/taille-pop].$$

Pour la fonction non linéaire, le paramètre q ($0 \leq q \leq 1$) ne dépend pas de $Taille-pop$; pour q croissant, on obtient une pression sélective de plus en plus importante.

Pour certains cas, ces approches destinées à améliorer les performances des AsG ont quelques inconvénients, parmi lesquels :

- La responsabilité de choisir ces mécanismes est attribuée à l'opérateur.
- L'ignorance de l'information sur l'évaluation relative des chromosomes.
- Le théorème de schéma est non respecté (classement des chromosomes).

D'autres changements sont également apportés pour que l'AG ait le squelette suivant :

- r chromosomes (non nécessairement distincts) choisis pour reproduire (parents).
- r chromosomes (distincts) choisis pour mourir.

Les chromosomes restants sont "neutres".

Le nombre des individus neutres (égal au minimum à $Taille-pop - 2r$ et au maximum à $Taille-pop - r$) dépend de r et de la diversité entre "parents" et "morts", ainsi, $P(t+1)$ est formée des individus neutres et de r descendants de r parents.

La structure d'un AG modifié est donnée par ce qui suit :

```

DEBUT
T = 1
Initialiser P(t)
Evaluer P(t)
Tant que (condition de terminaison fausse) Faire
    Début

```

T = t+1
 Sélection – parents de P(t-1)
 Sélection – morts de P(t-1)
 Former P(t) : reproduire les parents
 Evaluer P(t)
 Fin

FIN

Figure I.2 AG modifié.

On choisit les individus "morts" de telle manière à ce que les individus puissants auront moins de chance pour mourir suivant les étapes :

- Etape 1 : sélection de r parents (chacun est applicable à une seule opération)
- Etape 2 : sélection *Taille-pop* – r chromosomes de P(t) et les mettre dans P(t+1).
- Etape 3 : permettre r chromosomes de produire r nouveaux individus.
- Etape 4 : insérer les nouveaux individus dans P(t+1).

Ainsi, on obtient les caractéristiques suivantes :

- Chance d'un chromosome (parent et descendant) d'être choisi comme parent (parmi r) et comme membre de P(t+1).
- Application des opérateurs génétiques sur tous les membres de la population.

L'équation de grandissement devient :

$$\xi(S, t+1) \geq \xi(S, t) \cdot p_s(S) \cdot p_g(S). \quad (1.4. 6)$$

ou : $p_s(S)$: probabilité de survie du schéma S , elle est donnée par :

$$p_g = \overline{eval(S)} / \overline{F(t)}. \quad (1.4. 7)$$

Un schéma est meilleur qu'un schéma moyen si $p_g(S) > 1$. La formule (1.4. 6) implique que, pour un schéma court et d'ordre réduit :

$$p_s(S) * p_g(S) > 1 \quad (1.4. 8)$$

Le schéma reçoit un nombre d'individus croissant exponentiellement avec le temps.

1.4.1 Caractéristiques de la fonction

Des approches vis à vis de la fonction elle même pour la résolution du problème ont pris plusieurs directions. Une de ces approches est "la loi d'échelle en puissance", la fonction d'adéquation est prise comme suit :

$$f_i' = f_i^k, \quad (1.4. 9)$$

avec k proche de 1.

Ces approches ont également pour objectif d'éviter la convergence vers les optimums locaux. Dans toute approche faite, on doit assurer à l'algorithme pour les premières générations une évolution lente, tandis qu'il doit avoir une vitesse de convergence de plus en plus élevée au fur et à mesure qu'il s'approche de la dernière génération.

L'un des meilleurs systèmes est GENESIS 1.2 ucsd, utilisant deux paramètres pour le contrôle de la recherche en respectant les caractéristiques de la fonction à optimiser, à savoir, la fenêtre de réduction 'W' et le facteur de troncature. La fonction d'évaluation à utiliser est :

$$eal(x) = F - f(x). \quad (1.4.1)$$

où, F est une constante tel que :

Si $W > 0$, F prend la valeur maximale de $f(x)$ pour les W dernière génération,

Si $W = 0$, $F = \max. \{ f(x) \}$,

Si $W < 0$, l'opérateur peut utiliser une méthode.

Pour la condition d'arrêt de l'AG, on a deux catégories :

- l'une se base sur la structure du chromosome (génotype) : la convergence de la population est liée à la convergence des allèles (valeur d'un gène). Cette convergence est établie lorsque le nombre d'individus excède un certain pourcentage de la population.

- L'autre condition d'arrêt se base sur l'obtention d'un progrès inférieur à un certain ε petit donné.

I.4.2 Application contractante : AG

L'une des approches expliquant la convergence d'un AG est celle du point fixe. L'AG peut être défini comme étant une transformation dans un espace métrique et dont la propriété est de rapprocher les points d'une itération à une autre. Une application f est contractante si la distance entre $f(x)$ et $f(y)$ soit inférieure à la distance entre x et y .

Soit l'application $\delta : S \times S \rightarrow R$, et $x, y \in S$, si on a :

$$\delta(x, y) \geq 0 \text{ et } \delta(x, y) = 0 \text{ ssi } x = y.$$

$$\delta(x, y) = \delta(y, x).$$

$$\delta(x, y) + \delta(y, z) \geq \delta(x, z).$$

L'application δ est dite : distance. On note l'espace métrique par : $\langle S, \delta \rangle$.

Soit l'application $f : S \rightarrow S$, f est contractante ssi :

$$\exists \varepsilon \in [0,1] : \forall x, y \in S : \delta(f(x), f(y)) \leq \varepsilon * \delta(x, y).$$

Pour que p_0, p_1, \dots soit une suite de Cauchy :

$$\forall \varepsilon > 0 : \exists k : \forall m, n > k, \delta(p_m, p_n) < \varepsilon.$$

Un espace métrique est complet si toute suite de Cauchy a pour limite $p = \lim_{n \rightarrow \infty} p_n$

Théorème :

Soit $\langle S, \delta \rangle$ un espace métrique complet, et soit l'application $f : S \rightarrow S$ une transformation contractante. f a un point fixe tel que pour tout x_0 de S :

$$x = \lim_{i \rightarrow \infty} f^i(x_0).$$

où, $f^0(x_0) = x_0$ et $f^{i+1}(x_0) = f(f^i(x_0))$.

Si notre espace métrique a pour éléments les populations de solutions, une construction adéquate de cet espace fera que la convergence vers le point fixe puisse avoir lieu.

On suppose qu'on a un problème de maximisation. Soit la population $P = \{\bar{x}_1, \dots, \bar{x}_n\}$ de taille n fixe, et S l'ensemble de populations possibles.

On considère une fonction pour la population P , soit:

$$Eval(P) = \frac{1}{n} \sum_{\bar{x}_i \in P} eval(\bar{x}_i). \quad (1.4.11)$$

Soit $M = \max(eval(\bar{x}_i))$, pour tout \bar{x}_i du domaine de travail (en conséquence, $Eval(P) \leq M, \forall P$)

L'application $\delta: S \times S \rightarrow R$ est donnée par :

$$\delta(P_1, P_2) = \begin{cases} 0 & \text{si } P_1 = P_2 \\ |1 + M - Eval(P_1)| + |1 + M - Eval(P_2)| & \text{ailleurs} \end{cases} \quad (1.4.12)$$

et f une application contractante, $f: S \rightarrow S$ (i.e., $P(t+1) = f(P(t))$), on peut vérifier que $\langle S, \delta \rangle$ est un espace métrique, il est complet également, $(\exists k, \forall n > k: P_n > P_k)$.

L'AG devient :

Procédure CM-GA (Contractive Mapping GA)

DEBUT

$t = 0$

Initialiser $P(t)$

Evaluer $P(t)$

Tant que (condition d'arrêt non-vérifiée) Faire

 Début application contractante $f(P(t+1)) \rightarrow P(t)$

$t = t + 1$

 Sélectionner $P(t)$ de $P(t-1)$

 Recombinaison $P(t)$

 Evaluer $P(t)$

 Si $Eval(P(t-1)) \geq Eval(P(t))$

 alors $t = t - 1$

 Fin

FIN

Figure 1.3 Algorithme génétique CM.

Tant qu'il n'y a pas augmentation de $Eval(P(t))$, l'itération n'est pas comptée. L'espace δ étant déjà définie est en effet minimisée par cet algorithme, quelle que soit la population initiale.

Le point fixe sera la population P^* ne contenant que le maximum global (tous les individus sont les mêmes).

Pour terminer, le choix de δ et de f peut dépendre du problème lui-même.

I.4.3 AG à taille de population variable (GAVaPS)

Il est clair que la taille de la population joue sur la rapidité de convergence de l'AG.

L'AG qu'on va voir introduit la notion d'âge d'un chromosome (nombre de populations qu'il a survécu) dépendant de sa fonction d'adéquation et jouant ainsi sur la taille de la population, et le paramètre 'temps de vie' (nombre de populations qu'il a à survivre). L'GAVaPS utilise une population auxiliaire *Popaux* construite à partir de $P(t)$ de taille $TaillePopaux = TaillePop \times \rho$ où: ρ est la raison de reproduction. Chaque individu a une chance de reproduire indépendamment de fonction d'adéquation.

Procédure GAVaPS

DEBUT

$t = 0$

Initialiser $P(t)$

Evaluer $P(t)$

Tant que (condition de d'arrêt non-vérifiée) Faire

 Début

$t = t + 1$

 âge = âge + 1 (à chaque individu)

 Recombinaison $P(t)$

 Evaluer $P(t)$

 Enlever de $P(t)$ tout individu dont âge est supérieur au temps de vie

 Fin

FIN

Figure I.4 Algorithme d'un GAVaPS.

Un temps de vie est assigné au chromosome dès qu'il existe, et reste constant durant toute l'évolution du processus (de la naissance d'un chromosome jusqu'à sa mort). Une fois âge est plus grand que temps de vie le chromosome meurt. Ainsi, pour une itération la taille de la population est :

$$TaillePop(t+1) = TaillePop(t) + TaillePopaux(t) - D(t). \quad (1.4.13)$$

où: $D(t)$ est le nombre d'individus mourants à la génération t .

Une stratégie d'attribuer un temps de vie doit :

- Renforcer les individus de fonction d'adéquation au-dessus de la moyenne (leur attribuer un temps de vie plus grand).
- Choix de la taille de la population courante.

Le calcul doit prendre en considération l'état de recherche mesuré par *AvgFit*, *MaxFit*, *MinFit*, représentant respectivement, la valeur moyenne, maximale et minimale de la fonction d'adéquation de la population courante, et par *AbsMaxFit* et *AbsMinFit* qui sont les valeurs trouvées à travers toutes les générations.

Le temps de vie pour le $i^{\text{ème}}$ individu peut être calculé par :

- Attribution proportionnelle :

$$\min(\text{MinLT} + \eta \frac{\text{fonction d'adéquation}[i]}{\text{AvgFit}}, \text{MaxLT}). \quad (1.4. 14)$$

- Attribution linéaire :

$$\text{MinLT} + 2\eta \frac{\text{fonction d'adéquation}[i] - \text{AbsFitMin}}{\text{AbsFitMax} - \text{AbsFitMin}}. \quad (1.4. 15)$$

- Attribution bilinéaire :

$$\begin{cases} \text{MinLT} + \eta \frac{\text{fonction d'adéquation}[i] - \text{MinFit}}{\text{AvgFit} - \text{MinFit}} & \text{si } \text{AvgFit} \geq \text{fonction d'adéquation}[i] \\ \frac{1}{2}(\text{MinLT} + \text{MaxLT}) + \eta \frac{\text{fonction d'adéquation}[i] - \text{AvgFit}}{\text{MaxFit} - \text{AvgFit}} & \text{si } \text{AvgFit} < \text{fonction d'adéquation}[i] \end{cases}$$

MaxLT (MinLT) référence au temps de vie maximum (minimum) attribué, ce sont des paramètres du GAVaPS, et $\eta = \frac{1}{2}(\text{MaxLT} - \text{MinLT})$.

La première stratégie ne tient pas compte de la "bonté" d'un individu, en comparant sa fonction d'adéquation avec celle 'max' trouvée, ce qui conduit à la deuxième stratégie qui risque de donner un temps de vie trop élevé pour la valeur fonction d'adéquation $\cong \text{AbsFitMax}$. La troisième stratégie vient mettre un compromis entre les deux premières stratégies.

Des travaux expérimentaux ont montré l'influence de TaillePop sur la performance et le coût d'un SAG (simple AG) et d'un GAVaPS.

1.4.4 AG, contraintes et le problème de knapsack (sac à dos)

Pour un problème avec contraintes, on utilise la population et on lui associe une fonction de pénalité qui agit sur la fonction d'adéquation d'un individu si ce dernier viole les contraintes. Cette fonction peut être constante ou fonction du degré de violation. La pénalité peut atteindre l'élimination des solutions impossibles.

Une seconde approche consiste à corriger les solutions impossibles, mais cela peut parfois devenir plus difficile que le problème lui-même.

La troisième approche consiste à utiliser des "décodeurs" de manière à garantir l'obtention des solutions possibles ou bien utiliser des opérateurs spécifiques au problème.

a- Le problème de "knapsack" 0/1

Pour un ensemble donné de poids $W[i]$, de profits $P[i]$ et une capacité C , on cherche un vecteur binaire $x = \langle x[1], \dots, x[n] \rangle$ tel que :

$$\sum_{i=1}^n x[i] \cdot W[i] \leq C. \quad (1.4. 16)$$

et pour que :

$$P(x) = \sum_{i=1}^n x[i] \cdot P[i]. \quad (1.4. 17)$$

soit maximum.

On discute le comportement de quelques AsG basés sur des ensembles générés aléatoirement. La difficulté du problème dépend de la corrélation entre poids et profits. Trois ensembles sont générés aléatoirement :

- *Decorrélés:*

$$\begin{aligned} W[i] &:= (\text{uniforme})\text{alé atoire}([1..v]), \text{ et} \\ P[i] &:= (\text{uniforme})\text{alé atoire}([1..v]) \end{aligned}$$

- *Faiblement corrélés :*

$$\begin{aligned} W[i] &:= (\text{uniforme})\text{alé atoire}([1..v]), \text{ et} \\ P[i] &:= W[i] + (\text{uniforme})\text{alé atoire}([-r..r]) \end{aligned}$$

- *Fortement decorrélés :*

$$\begin{aligned} W[i] &:= (\text{uniforme})\text{alé atoire}([1..v]), \text{ et} \\ P[i] &:= W[i] + r \end{aligned}$$

Si la corrélation augmente la différence $\max_{i=1..n} \{ P[i]/W[i] \} - \min_{i=1..n} \{ P[i]/W[i] \}$ diminue.

b- Description des algorithmes

La solution x est une chaîne binaire : la $i^{\text{ème}}$ composante est choisie pour le "knapsack" (sac à dos) ssi $x[i]=1$. La fonction d'adéquation est donnée par :

$$\text{eval}(x) = \sum_{i=1}^n x[i] \cdot P[i] - \text{Pen}(x). \quad (1.4. 18)$$

avec :

$$\text{Pen}(x) = \begin{cases} 0 & \text{si } \sum_{i=1}^n x[i] \cdot W[i] \leq C \\ > 0 & \text{ailleurs.} \end{cases} \quad (1.4. 19)$$

La fonction de pénalité est donnée par différents types (\log_2 , linéaire, quadratique). La fonction de pénalité linéaire, par exemple, est donnée par :

$$A_p[1]: \text{Pen}(x) = \rho \left(\sum_{i=1}^n x[i] \cdot W[i] - C \right). \quad (1.4. 20)$$

où: $\rho = \max_{i=1..n} \{P[i] \cdot W[i]\}$.

- Algorithme $A_r[i]$

La même condition que précédemment pour le choix d'un bit pour le "knapsack". La fonction d'adéquation est donnée par :

$$\text{eval}(x) = \sum_{i=1}^n x'[i] \cdot P[i]. \quad (1.4. 21)$$

où: le vecteur x' est la version corrigée de x .

Procédure 'réparer (x)':

DEBUT

knapsack-trop-plein = faux

$x' = x$

Si $\sum_{i=1}^n x'[i] \cdot W[i] > C$

Alors : knapsack-trop-plein = vraie

Tant que (Knapsack-trop-plein) Faire

Début

i = sélectionner un bit de knapsack

Enlever ce bit, i.e., $x'[i] = 0$

Si $\sum_{i=1}^n x'[i] \cdot W[i] \leq C$

Alors "knapsack-trop-plein = faux

Fin

FIN

Figure I.5 Procédure de réparation.

- Algorithme $A_d[i]$ (décodage)

Chaque chromosome est un vecteur de n composantes entières (représentation ordinaire), la $i^{\text{ème}}$ composante est comprise entre 1 et $n-i+1$. Cette représentation à pour référence une liste L d'éléments. Soit la liste $L = (1,2,3,4,5,6)$, par exemple, le vecteur $x = \langle 4,3,4,1,1,1 \rangle$ est décodé selon la séquence 4,3,6 (l'élément 6 est la quatrième position déplacement de 4 et 3) 1, 2, et 5. Il est clair que le croisement de deux parents "possibles" donnera un descendant possible. Pour la mutation, le $i^{\text{ème}}$ gène prend une valeur aléatoire appartenant à l'intervalle $[1..n-i+1]$.

Il y a deux types d'algorithme qui se différent dans la construction de la liste L :

- $A_d[1]$: (décodage aléatoire) : l'ordre des éléments de L est celui de la rentrée des données.

- $A_d[2]$: (décodage gradué) : l'ordre des éléments de la liste L correspond au rapport profit sur poids.

Procédure 'décoder (x)' :

```
DEBUT
  Construire une liste  $L$ 
   $i = 0$ 
   $WS = 0$  (somme des poids )
   $PS = 0$  (somme des profits )
  Tant que  $i \leq n$  Faire
    Début
       $j = x[i]$ 
      enlever le  $j^{\text{ème}}$  élément de  $L$ 
      Si  $WS + W[j] \leq C$  alors
        Début
           $WS = WS + W[j]$ 
           $PS = PS + P[j]$ 
        Fin
      Fin
    Fin
  FIN
```

Figure 1.6 Procédure de décodage pour la représentation ordinaire.

1.5 Binaire ou flottant ?

L'utilisation d'une représentation binaire a très souvent du mal à s'adapter avec les problèmes multidimensionnels (plusieurs variables). Il faudrait donc opérer avec un alphabet plus large, et éventuellement de nouveaux opérateurs (codage réel par exemple).

La représentation en virgule flottante permet de rapprocher l'espace problème et l'espace représentation (deux points proches dans l'espace problème doivent être proches dans l'espace représentation).

Exemple :

Le code Gray (le passage d'une valeur à sa suivante ne provoque que la variation d'un seul bit).

Nous utilisons également le flottant pour nous rapprocher de l'espace problème.

Nous avons choisi un problème à résoudre pour comparer les deux représentations: binaire et en virgule flottante. A noter que la seconde représentation permet un domaine plus large de recherche ou une meilleure précision pour le même nombre de bits. Pour un vecteur (plusieurs composantes), on juxtapose les composantes pour donner un chromosome.

1.5.1 Mutation est croisement aléatoires

Pour garder similaire le croisement des deux représentations, on les effectue entre les éléments (composantes) des deux chromosomes.

Le croisement pour la représentation en virgule flottante est similaire à celui de la représentation binaire.

La mutation, qui est aléatoire, est appliquée sur des nombres en flottant plutôt que sur des bits. Une telle mutation donne un nombre aléatoire de l'intervalle $\langle LB, UB \rangle$ (LB et UB sont les bornes inférieure et supérieure respectivement de la variable mutée), cela assure au moins que l'élément résultant reste dans le domaine.

Les résultats ont montré la supériorité du flottant en précision et en vitesse.

1.5.2 Mutation non uniforme

Cet opérateur est défini comme suit :

-Flottant

Si $s_v^t = \langle v_1, \dots, v_n \rangle$ est un chromosome, et l'élément v_k est choisi pour la mutation, le vecteur résultant sera $s_k^{t+1} = \langle v_1, \dots, v'_k, \dots, v_n \rangle$ où :

$$v'_k = \begin{cases} v_k + \Delta(t, UB - v_k) & \text{si le digit aléatoire} = 0, \\ v_k + \Delta(t, v_k - LB) & \text{si le digit aléatoire} = 1, \end{cases}$$

La fonction $\Delta(t, y)$ retourne une valeur dans le domaine $[0, y]$ et la probabilité de s'approcher de zéro augmente avec t . Le nombre généré sera de plus en plus proche de son précédent au fur et à mesure qu'on s'approche des dernières générations. On utilise la fonction suivante :

$$\Delta(t, y) = y \cdot (1 - r^{(1-\frac{t}{T})^p})$$

où : r : nombre aléatoire de l'intervalle $[0,1]$,
 T : nombre de générations maximum,
 b : paramètre dépendant du nombre d'itérations.

-Binaire

Le bit muté est donné par :

$$v'_k = mutate(v_k, \nabla(t, n)),$$

où, n nombre de bits par élément, $mutate(v_k, pos)$ signifie : muter le bit de position pos du $k^{\text{ème}}$ élément.

$$\nabla(t, n) = \begin{cases} \lfloor \Delta(t, n) \rfloor & \text{si le bit aléatoire} = 0, \\ \lceil \Delta(t, n) \rceil & \text{si le bit aléatoire} = 1, \end{cases}$$

1.5.3 Autres opérateurs

-Binaire

On utilise le croisement multi-points.

-Flottant

On utilise aussi un croisement multi-points similaire, et un croisement multi-points faisant la moyenne des deux éléments choisis pour le croisement.

I.6 Choix local

On peut se servir des AsG pour arriver à une recherche locale des solutions, mais dans un stade supérieur, où l'on exige une assez grande précision, les "schémas" doivent être d'ordre et de longueur élevés.

Pour cela, on utilise une nouvelle technique de mutation : la probabilité de mutation est d'autant plus importante qu'on soit proche du côté droit de la chaîne binaire représentant une solution.

Pour les problèmes de contrôle optimal, où l'on préfère l'utilisation du flottant, on a recours à utiliser des opérateurs spécialisés.

I.6.1 Groupe de mutation

Si $x_i^t = \langle v_1, \dots, v_n \rangle$ est un chromosome. Le résultat de la mutation du vecteur v_k sera: $x_i^{t+1} = \langle v_1, \dots, v_k', \dots, v_n \rangle$.

Mutation uniforme

Tous les éléments ont la même probabilité d'être mutés.

Mutation non uniforme

Le vecteur v_k choisi pour la mutation a pour domaine : $[l_k, u_k]$, le résultat de la mutation est v_k' qui sera :

$$v_k' = \begin{cases} v_k + \Delta(t, u_k - v_k) & \text{si le digit aléatoire} = 0, \\ v_k + \Delta(t, v_k - l_k) & \text{si le digit aléatoire} = 1, \end{cases}$$

où, la fonction $\Delta(t, y)$ est donnée par :

$$\Delta(t, y) = y \cdot (1 - r^{(1 - \frac{t}{T})^b})$$

r : nombre aléatoire de l'intervalle $[0,1]$,

T : nombre maximum de générations,

b : paramètre du système déterminant le degré de non uniformité.

I.6.2 Groupe de croisement

a-Croisement simple

Il s'agit du croisement qu'on a déjà défini, mais il est réalisé entre les composantes (v_i) du chromosome.

b-Croisement arithmétique

Cet opérateur est défini comme étant une combinaison linéaire de deux chromosomes, si s_v^t et s_w^t sont choisis pour le croisement. Le résultat sera :

$$S_v^{t+1} = a \cdot S_w^t + (1-a) \cdot S_v^t, \text{ et}$$

$$S_w^{t+1} = a \cdot S_v^t + (1-a) \cdot S_w^t.$$

Le paramètre a peut être constant (croisement arithmétique uniforme) ou en fonction de l'âge de la population (croisement arithmétique non uniforme).

Cet opérateur peut être appliqué à deux chromosomes choisis ou à tous les chromosomes.



CHAPITRE II

Exemple d'application

Exemple d'application :

On se propose de minimiser le module de l'impédance vue entre les bornes A et B de la figure ci-dessous telle que :

$$Z^2 = R^2 + (L\omega - 1/C\omega)^2.$$

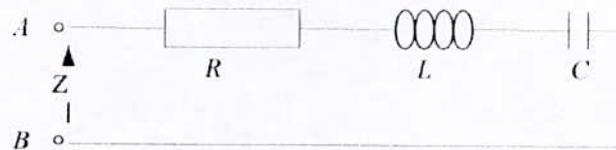


Figure *.*: Circuit RLC

Z : est le module de cette impédance.

R : est la valeur de la résistance.

L : est la valeur de l'inductance de la bobine.

C : est la valeur de la capacité du condensateur.

ω : est la pulsation de la tension entre les bornes A et B .

Cet exemple manque d'intérêt pratique car il suffit de dériver par rapport à ω pour trouver la valeur ω_0 minimisant le module; on peut aisément vérifier que:

$$\omega_0 = 1/(LC)^{1/2}.$$

Mais cela n'empêche qu'on puisse vérifier la capacité de l'algorithme génétique de trouver ω_0 on lui donnant :

- l'intervalle où elle se situe.
- La précision sur ω correspondant au nombre de chiffres après la virgule désiré.
- Une population initiale de solutions aléatoirement choisies (appartenant à l'intervalle donné).
- Les paramètres propres à l'algorithme tels que:
 - La taille de la population de solutions.
 - Probabilité de croisement (P_c).
 - Probabilité de mutation (P_m).

Pour notre exemple on choisit la taille de la population : $Taille_{pop} = 20$, $P_c = 0.25$, $P_m = 0.01$, et la précision $Pr = 0.001$ (trois chiffres après la virgule).

La fonction d'adéquation est évidemment le module de l'impédance à minimiser.

On a choisi la représentation binaire des solutions potentielles pour multiples raisons:

- La facilité de manipuler les chaînes binaires en utilisant les vecteurs.
- La présence de toutes les solutions possibles dans l'intervalle de résolution donné quelque soit la transformation subite par n'importe quel individu.

Il faut noter par contre que cette notation (binaire) reste sous-optimale devant la notation en virgule flottante du point de vue compromis précision/nombre de bits utilisés pour représenter une solution.

On a pris l'intervalle de résolution $[a, b] = [700, 1500]$.

Les valeurs numériques des différents composants sont:

- $R = 100 \Omega$.
- $L = 10 \text{ mH}$.
- $C = 100 \mu\text{F}$.

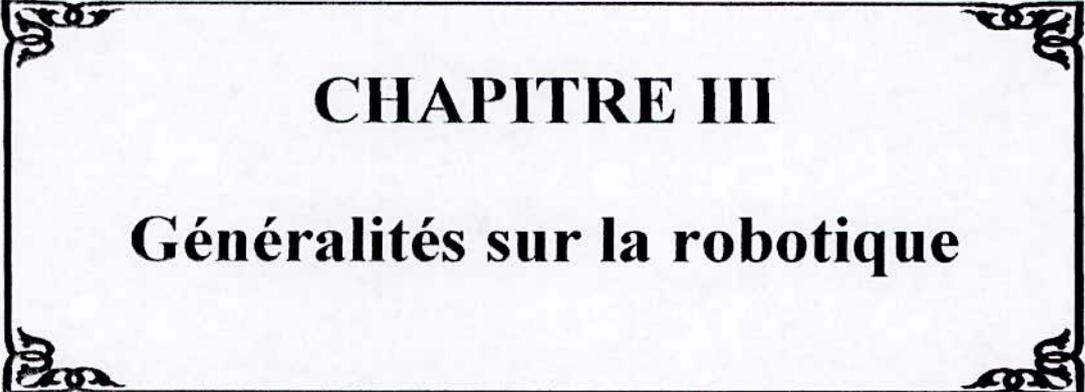
Vue la simplicité du problème (forme mathématique simple de la fonction d'adéquation), le temps de calcul de la solution avec la précision imposée reste assez court (en générale ne dépasse pas 20 secondes); il devient de plus en plus court lorsque l'erreur permise grandit (comme toute méthode itérative de recherche).

Algorithme de recherche:

Algorithme utilisé contient les étapes suivantes:

1. Population initiale choisie aléatoirement sans dépasser les bornes de l'intervalle de recherche.
2. Evaluation de cette population après conversion de ses individus en des réels.
3. Opération de sélection des individus subissant les changements (effet des opérateurs génétiques utilisés).
4. Opération de croisement.
5. Opération de mutation.
6. Conversion de la nouvelle population en des réels et leur évaluation.
7. Fusion des deux populations (initiale et celle obtenue après application des opérateurs génétiques).
8. Choix des 20 meilleurs individus.

Ce processus se répète jusqu'à l'arrivée à un optimum satisfaisant.



CHAPITRE III

Généralités sur la robotique

III.1. Définitions et classification des systèmes robotiques

Les problèmes de définition et de classification des robots ne sont pas simples, jusqu'aujourd'hui ils ne sont pas résolus de manière satisfaisante, en dépit de l'existence de certaines définitions et de certaines classifications proposées par des spécialistes.

III.1.1 Définitions des robots:

On trouve la définition du terme "robot" évidemment, dans les dictionnaires ou encyclopédies, mais surtout dans les textes des associations de robotique de divers pays.

- Ainsi, le dictionnaire Robert propose comme définition :

"Mécanisme automatique pouvant se substituer à l'homme pour effectuer certaines opérations, et capable d'en modifier de lui même le cycle et d'exercer un certain choix (par la détection photoélectrique, les "cerveaux" électroniques, des servomoteurs, etc.)"

- La **JIRA** (**J**apan **I**ndustrial **R**obot **A**ssociation), quant à elle, met plus l'accent sur la notion d'anthropomorphisme en donnant comme définition :

"Dispositif versatile et flexible, offrant des fonctions de déplacement similaires à l'être humain ou dont les fonctions de déplacement sont commandées par des capteurs et des moyens de reconnaissance."

- La **RIA** (**R**obotic **I**ndustries **A**ssociation), aux Etats-Unis, insiste sur l'utilisation en proposant comme définition (1979) :

"Manipulateur multifonctions reprogrammable conçu pour déplacer des matériaux, des pièces, des outils ou des dispositifs spécialisés au travers des déplacements variables et programmables pour accomplir diverses tâches."

Le robot industriel est présenté comme un manipulateur asservi en position, reprogrammable, polyvalent et capable de positionner et d'orienter des matériaux, pièces, outils ou dispositifs spécialisés. Il est présent dans des environnements hostiles, où il est destiné à accomplir des tâches dont l'homme est incapable de les effectuer.

Deux caractéristiques essentielles du robot le distinguent d'une machine automatisée (machines à commande numériques par exemple).

- La **versatilité** ou **flexibilité** qui concerne la possibilité, pour un robot, d'exécuter des tâches différentes et/ou d'exécuter une même tâche de plusieurs manières.

- L'automatisation complète de l'exécution de la tâche indépendamment d'une connaissance "complète et à l'avance" de l'environnement dont le quel la tâche est réalisée; et que la machine puisse à elle seule appréhender la situation de cet environnement et établir les conditions à respecter en performance dans la mise en oeuvre de son travail.

III.1.2 Structure générale d'un robot :

Généralement, le robot manipulateur est constitué de cinq parties. Le schéma synoptique d'un système robotique est le suivant :

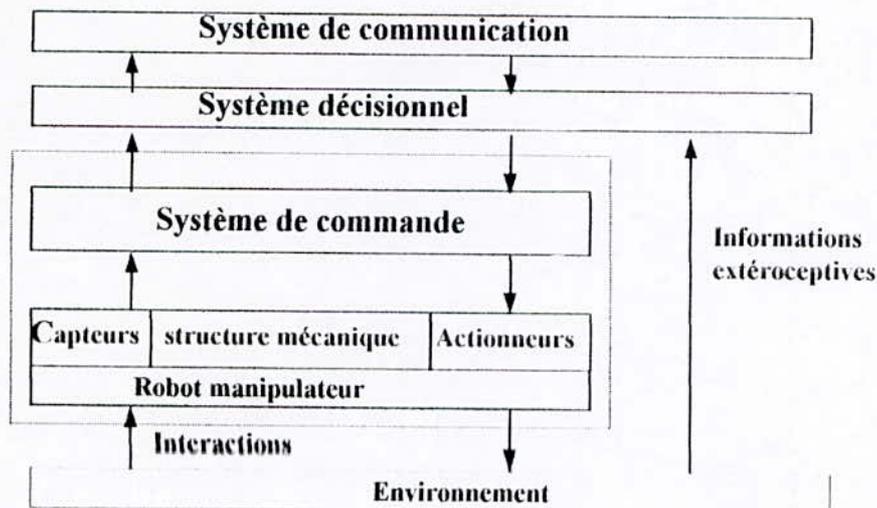


Figure III.1.1 : Structure générale d'un robot manipulateur

- La **structure mécanique** est la partie la plus visible du robot. Elle supporte l'outil terminal.
- Les **actionneurs** qui représentent la source d'énergie du bras manipulateur. Les actionneurs produisent les forces nécessaires pour en modifier sa configuration.
- les **capteurs** qui sont nécessaires pour la perception et le contrôle des actions, parmi lesquels on distingue les capteurs proprioceptifs qui permettent de saisir l'état mécanique du robot, et les capteurs extéroceptifs qui indique l'état de l'environnement du robot manipulateur.
- Le **système de commande** chargé de piloter les actionneurs à partir des informations reçues par le système décisionnel sur le mouvement à exécuter et les informations délivrées par les capteurs.
- Le **système décisionnel** qui élabore le mouvement du bras manipulateur à partir de la définition de la tâche à exécuter transmise par l'opérateur à l'aide du système de communication: il travaille en coordination avec la commande.
- Le **système de communication** qui gère les messages transmis entre le système décisionnel et l'opérateur.

III.1.3 Classification des robots manipulateurs :

La classification des robots peut se faire selon plusieurs critères :

- Classification selon le niveau d'automatisation (développement technologique de leur commande).
- Classification selon leur type d'asservissement (boucle ouverte, boucle fermée).
- Classification selon leur type d'application.
- Classification selon la structure du bras (morphologique).

C'est surtout cette dernière qui est la plus souvent utilisée, elle donne à la fois la description de la chaîne mécanique que constitue les différents segments et les degrés de liberté concernant leur mouvement. Il existe quatre types de bras à trois degrés de liberté, à savoir :

a) Bras cartésien (TTT) :

Robot dont le bras est constitué de trois articulations prismatiques dont les axes sont disposés selon un système de coordonnées cartésiennes.

b) Bras cylindrique (TRT) :

Il comprend deux articulations prismatiques et une articulation rotative dont les axes sont disposés selon un système de coordonnées cylindriques.

c) Bras sphérique (RRT) :

Robot dont le bras est constitué de deux articulations pivots et une articulation prismatique dont les axes sont disposés suivant un système de coordonnées polaires. Le volume de travail de ce bras est une portion de sphéroïde.

d) Bras articulé (RRR) :

Ce robot présente trois articulations rotatives et permet donc un mouvement dans différents plans (très grande flexibilité).

III.1.4 Sources d'énergie dans un système robotique :

La source d'énergie pour un système robotique est présentée par les actionneurs, leur rôle est la mise en mouvement en déplaçant le bras à partir d'ordres fournis par l'ordinateur.

L'industrie des robots utilise trois sortes d'actionneurs :

a) Les actionneurs hydrauliques :

Ce système utilise un fluide incompressible (l'huile mise dans un cylindre sous pression avec un piston). Ils sont destinés pour la motorisation de robots lourds. Leur puissance varie entre 80 et 300 bars. Les inconvénients de ce type d'actionneurs sont :

- Difficulté de la commande.
- Technologie onéreuse.
- Sensibilité des électrovalves quand la pression est faible.

b) Les actionneurs pneumatiques :

Ces actionneurs utilisent de l'air à la place de l'huile. Ils ont une puissance qui va de 4 à 8 bars. Les inconvénients de ce type d'actionneurs sont :

- Réclamation des compresseurs.
- Ils ne peuvent pas être asservis.
- Energie limitée.

c) Les actionneurs électrique :

Ces actionneurs sont très utilisés en robotique. Ils présentent un rendement énergétique assez élevé et permettent de déplacer les parties mécaniques avec une grande précision. En général, ils utilisent des moteurs pas à pas ou des moteurs à courants continus. L'énergie électrique est

intéressante de point de vue de la propreté, du silence, des facilités de maintenance et de la sécurité, mais atteint des limites pour les charges importantes, les bras de grande longueur et les vitesses très importantes ou très faibles.

III.2 Modélisation cinématique

III.2.1 Introduction :

Tout manipulateur peut être considéré comme une chaîne de corps solides appelés segments, connectés par des articulations rotatives ou prismatiques. Chaque segment, appelé aussi liaison, est caractérisé par son propre repère. Utilisant les transformations homogènes, on peut décrire la situation d'un repère par rapport à un autre.

Chaque configuration géométrique d'un manipulateur est associée à chaque position et orientation de l'organe terminal du robot, et peut être définie par un ensemble de variables articulaires. Une variable articulaire traduit les déplacements relatifs intermédiaires entre deux corps rigides adjacents sous l'effet d'une force extérieure produite par un ou plusieurs actionneurs. Ainsi les variables articulaires (appelées également coordonnées généralisées) permettent de faire le lien entre la position et l'orientation de l'organe terminal dans un repère (R_0).

La cinématique des bras manipulateurs est la partie qui traite les caractéristiques géométriques du mouvement sans s'intéresser aux forces et aux moments qui causent ces mouvements.

Dans ce chapitre, on s'intéresse à la description de la position et de l'orientation des différents segments du bras manipulateur dans la situation statique.

Le modèle cinématique direct d'un bras manipulateur permet d'exprimer la situation de l'organe terminal (orientation et position) en fonction des variables du bras lui-même.

La modélisation inverse permet de résoudre le problème inverse : connaissant la situation du bras, on peut connaître sa configuration.

Pour décrire la relation de translation et de rotation entre les segments, DENAVIT et HARTENBERG, en 1955, proposaient une méthode matricielle qui détermine systématiquement les coordonnées du système pour chaque segment de la chaîne articulée. La représentation D-H détermine une matrice de transformation homogène (4×4), qui représente les coordonnées de chaque segment du système, ainsi à travers cette transformation les coordonnées de l'outil terminal peuvent être représentées et transformées au repère de la base.

III.2.2 La matrice de transformation homogène :

La transformation qui nous permet de passer d'un référentiel $R_{i-1}(O_{i-1}, X_{i-1}, Y_{i-1}, Z_{i-1})$ à un référentiel $R_i(O_i, X_i, Y_i, Z_i)$ fait appel à une matrice de transformation homogène ${}^{i-1}T_i$. La matrice ${}^{i-1}T_i$ est composée de :

- un vecteur de translation ${}^{i-1}P_i$ décrivant la position de O_i dans R_{i-1} .
- Une matrice de rotation ${}^{i-1}R_i$ (3×3) appelée matrice de passage de R_{i-1} à R_i .

La matrice de transformation homogène (4×4) contient à la fois la translation et la rotation et est définie par :

$${}^{i-1}T_i = \begin{bmatrix} {}^{i-1}R_i & {}^{i-1}P_i \\ 0 & 1 \end{bmatrix} \quad (\text{III.2.1})$$

Avec :

$${}^{i-1}R_i = [n \quad o \quad a]$$

$n_i = [n_{ix} \quad n_{iy} \quad n_{iz}]^T$: vecteur normal,

$o_i = [o_{ix} \quad o_{iy} \quad o_{iz}]^T$: vecteur d'orientation,

$a_i = [a_{ix} \quad a_{iy} \quad a_{iz}]^T$: vecteur d'approche,

${}^{i-1}P_i = [P_{ix} \quad P_{iy} \quad P_{iz}]$: vecteur position.

Propriétés :

1. Le déterminant de ${}^{i-1}T_i$ est égal à 1.
2. L'inverse de ${}^{i-1}T_i$ est ${}^i T_{i-1}$ donnée par :

$$({}^{i-1}T_i)^{-1} = {}^i T_{i-1} = \begin{bmatrix} {}^i R_{i-1} & {}^i P_{i-1} \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} ({}^{i-1}R_i)^T & -{}^{i-1}R_i \cdot {}^{i-1}P_i \\ 0 & 1 \end{bmatrix} \quad (III.2.2)$$

Si on a une rotation d'un angle θ_i autour de l'axe X_i , d'après la figure (III.2.1) on aura :

$${}^{i-1}T_i = Rot(X_i, \theta_i) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\theta_i & -S\theta_i & 0 \\ 0 & S\theta_i & C\theta_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (III.2.3)$$

Avec: $C\theta_i = \cos(\theta_i)$, $S\theta_i = \sin(\theta_i)$.

Pour une rotation autour de l'axe Y_i on aura :

$$Rot(Y_i, \theta_i) = \begin{bmatrix} C\theta_i & 0 & S\theta_i & 0 \\ 0 & 1 & 0 & 0 \\ -S\theta_i & 0 & C\theta_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (III.2.4)$$

De la même manière pour une rotation autour de l'axe Z_i :

$$Rot(Z_i, \theta_i) = \begin{bmatrix} C\theta_i & -S\theta_i & 0 & 0 \\ S\theta_i & C\theta_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (III.2.5)$$

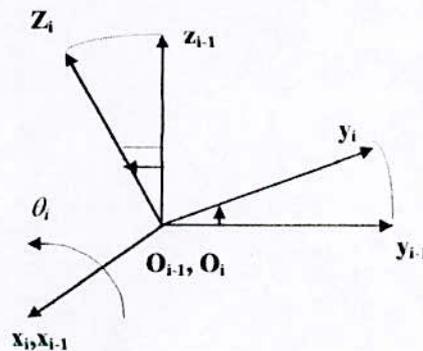


Figure III.2.1 : rotation du R_i autour de l'axe x_{i-1}

III.2.3 Les équations de transformation :

Considérons la situation décrite par la figure suivante

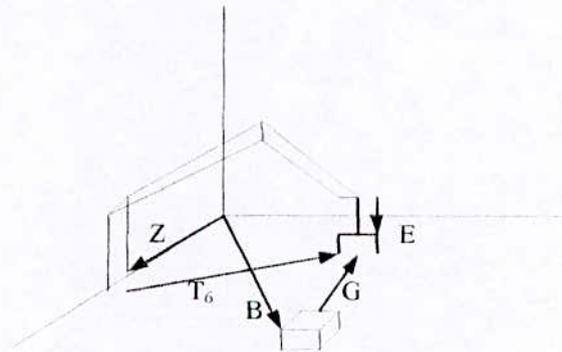


Figure III.2.2 : outil terminal - objet

On peut décrire l'outil terminal par rapport à un repère de référence par deux chemins :

1^{er} chemin : $Z \times T_6 \times E$

2^{ème} chemin : $B \times G$

Les deux chemins sont équivalents :

$$Z \times T_6 \times E = B \times G \tag{III.2.6}$$

Afin de simplifier la solution de l'équation de transformation, on utilise le graphe de transformation :

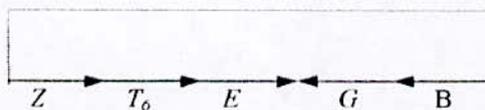


Figure III.2.3 : Graphe de transformation

Si on veut déterminer une transformation donnée, on part de sa base et on fait le tour. La transformation T_6 , par exemple, est donnée par :

$$T_6 = Z^{-1} \times B \times G \times E^{-1}$$

III.2.4 Présentation de la méthode D-H

DENAVIT et HARTENBERG ont établi la transformation qui permet le passage d'un repère $R_i(O_i, X_i, Y_i, Z_i)$ à un autre $R_j(O_j, X_j, Y_j, Z_j)$ en utilisant quatre paramètres θ_i, d_i, a_i et α_i .

Celle-ci peut être décomposée en quatre transformations élémentaires comme le montre la figure (III.2.4) :

$${}^{i-1}T_i = \text{Rot}(Z_{i-1}, \theta) \text{Trans}(0,0,d_i) \cdot \text{Trans}(a_i,0,0) \cdot \text{Rot}(X_i, \alpha) \quad (\text{III.2.7})$$

où :

$\text{Rot}(Z_{i-1}, \theta)$: rotation de θ , autour de Z_{i-1} .

$\text{Rot}(X_i, \alpha)$: Rotation de α autour de X_i .

$\text{Trans}(0,0,d_i)$: Translation d'une distance d le long de l'axe Z_{i-1} .

$\text{Trans}(a_i,0,0)$: translation d'une distance a le long de l'axe X_{i-1} .

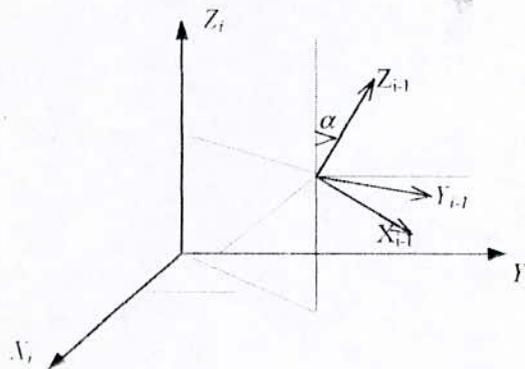


Figure III.2.4

L'implémentation des repères pour chaque segment, en utilisant la transformation D-H, est présentée ci-dessous. Pour appliquer cette transformation il faut d'abord fixer les différents repères dans chaque articulation d'une façon successive, en procédant comme suit :

Etape 1

Numéroter chaque liaison (articulation) et segment en commençant de la base, notée segment "0" et l'élément terminal noté segment "n", le segment "i" se déplace soit autour (rotation) de l'axe "i-1", soit le long (glissière) de cet axe.

Etape 2

Etablir les repères de coordonnées de chaque segment en suivant les règles suivantes :

- L'axe Z_{i-1} correspond au déplacement du segment "i" dans le référentiel "i-1".
- L'axe X_i est choisi perpendiculaire à Z_i et Z_{i-1} .
 - si Z_i et Z_{i-1} ne se coupent pas X_i est choisi colinéaire à la perpendiculaire commune à Z_i et Z_{i-1} .
 - si Z_i et Z_{i-1} se coupent on choisi X_i arbitrairement dans le plan perpendiculaire à Z_{i-1} .
- L'axe Y_i est choisi de telle sorte qu'il forme un trièdre direct avec les deux autres axes.

Étape 3

Définir les paramètres θ_i , d_i , a_i et α_i (mettre sous forme de tableau de préférence).

- θ_i : L'angle entre X_i et X_{i-1} obtenu par rotation de X_i vers X_{i-1} autour de Z_{i-1} , θ_i peut être une variable ou une constante selon le type de l'articulation pivot ou glissière.

- d_i : C'est la valeur de la coordonnée de l'origine O_i sur l'axe Z_{i-1} , autrement dit c'est la distance entre O_i et l'intersection de Z_{i-1} avec X_i , d_i peut être une variable ou une constante selon le type de l'articulation (pivot ou glissière).

- α_i : C'est un angle constant entre Z_{i-1} et Z_i obtenu en vissant Z_{i-1} vers Z_i autour de X_i .

- a_i : C'est la distance entre Z_{i-1} et Z_i mesurée sur l'axe X_i négatif, a_i constante.

Étape 4

Former la matrice de déplacement homogène relative à l'articulation i .

D'après les trois étapes précédentes on a :

1. Une rotation du référentiel O_{i-1} de θ_i autour de Z_{i-1} .
2. Une translation du référentiel O_{i-1} de d_i le long de l'axe Z_{i-1} ($Trans(0,0,d_i)$).
3. Une translation du référentiel O_{i-1} de a_i le long de l'axe X_{i-1} .
3. Une rotation du référentiel O_{i-1} de α_i autour de l'axe X_{i-1} .

Donc la matrice de transformation homogène ${}^{i-1}T_i$ est :

$${}^{i-1}T_i = Rot(Z_{i-1}, \theta_i) \times Trans(0,0,d_i) \times Trans(a_i,0,0) \times Rot(X_{i-1}, \alpha_i).$$

$$\cos(\theta_i) = C\theta_i, \quad \cos(\alpha_i) = C\alpha_i$$

$$\sin(\theta_i) = S\theta_i, \quad \sin(\alpha_i) = S\alpha_i.$$

$${}^{i-1}T_i = \begin{bmatrix} C\theta_i & -S\theta_i & 0 & 0 & 1 & 0 & 0 & a_i \\ S\theta_i & C\theta_i & 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & C\alpha_i & -S\alpha_i & 0 \\ 0 & S\alpha_i & C\alpha_i & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^{i-1}T_i = \begin{bmatrix} C\theta_i & -S\theta_i C\alpha_i & S\theta_i S\alpha_i & a_i C\theta_i \\ S\theta_i & C\theta_i C\alpha_i & -C\theta_i S\alpha_i & a_i S\theta_i \\ 0 & S\alpha_i & C\alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{III.2.8})$$

III.2.5 Modélisation géométrique directe de quelques bras manipulateurs :

Les matrices de transformations ${}^{i-1}T_i$ seront notées par la suite A_i .

a) Le bras sphérique R.R.T:

On applique la matrice de transformation pour le bras manipulateur sphérique qui est représenté par la figure (III.2.5).

Ce bras à trois degrés de liberté, deux degrés de type rotatif et un degré de type glissière.

L'implémentation des repères en utilisant la méthode D-H est représentée dans la figure (III.2.6).

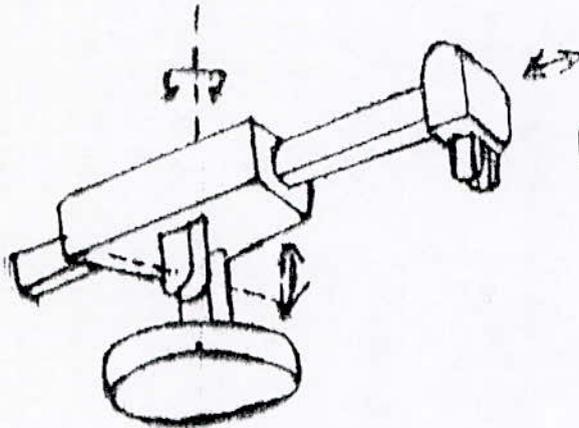


Figure III.2.5 : Représentation du bras sphérique à 3 d.d.l.

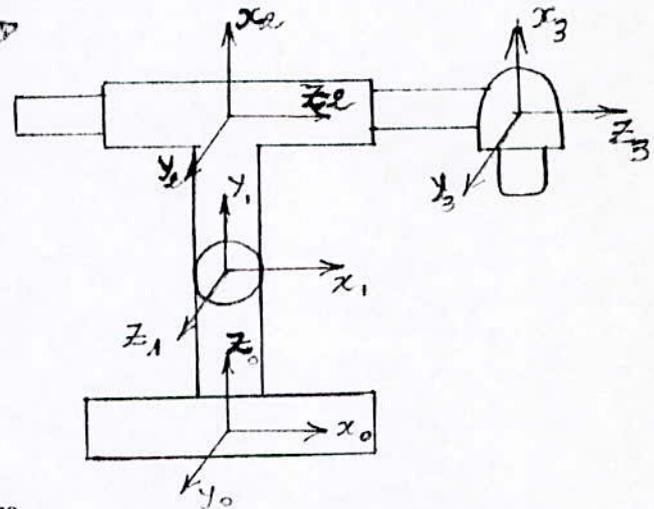


Figure III.2.6 : Implémentation des repères.

On définit les paramètres suivant :

θ_1, θ_2 : variables articulaires rotatives.

d_3 : variable articulaire prismatique.

Les paramètres D-H sont donnés par le tableau suivant :

segments	variables	θ_i	α_i	a_i	d_i
segment 1	θ_1	90	90	0	d_1
segment 2	θ_2	90	90		0
segment 3	d_3	0	0	0	d_3

Les matrices de transformations entre chaque deux repères consécutifs sont données par :

$$A_1 = \begin{bmatrix} -S\theta_1 & 0 & C\theta_1 & 0 \\ C\theta_1 & 0 & S\theta_1 & 0 \\ 0 & 1 & 0 & d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (III.2.9)$$

$$A_2 = \begin{bmatrix} -S\theta_2 & 0 & C\theta_2 & -a_2 \cdot C\theta_2 \\ C\theta_2 & 0 & S\theta_2 & a_2 \cdot S\theta_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (III.2.10)$$

$$A_3 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (III.2.11)$$

La matrice de transformation globale est donnée par :

$${}^0T_3 = A_1 A_2 A_3 = \begin{bmatrix} S_1 \cdot C_2 & C_1 & -S_1 \cdot C_2 & -S_1 \cdot C_2 \cdot d_3 + a_2 \cdot S_1 \cdot S_2 \\ -C_1 \cdot S_2 & S_1 & C_1 \cdot S_2 & d_3 \cdot C_1 \cdot C_2 - a_2 \cdot C_1 \cdot S_2 \\ C_2 & 0 & S_2 & d_3 \cdot S_2 + a_2 \cdot C_2 + d_1 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (III.2.12)$$

b) Le robot Puma à 2 d.d.l:

Les figures (III.2.7) et (III.2.8) donnent respectivement la représentation du robot PUMA à 2 d.d.l et l'implémentation des repères dans chaque liaison.

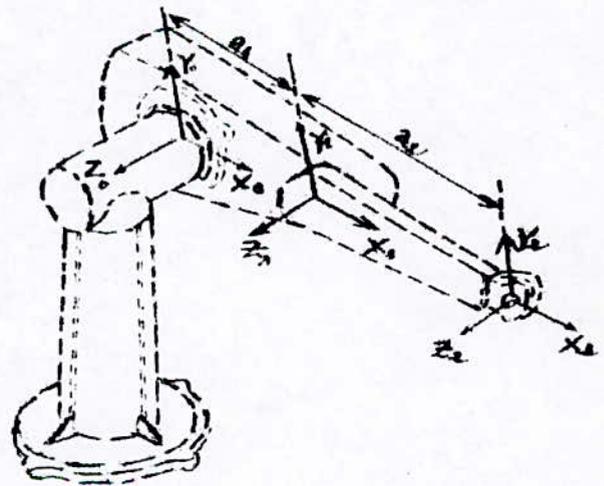
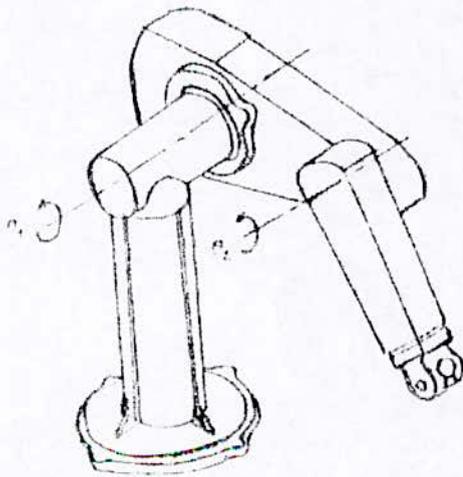


Figure III.2.7 : Représentation du robot PUMA à 2 d.d.l. Figure III.2.8 : Implémentation des repères.

On définit deux variables articulaires : θ_1, θ_2 .

Les paramètres D-H sont donnés par le tableau suivant :

segments	variables	$\theta_{initial}$	α_i	a_i	d_i
segment 1	θ_1	0	0	a_1	0
segment 2	θ_2	0	0	a_2	0

Les matrices A_1 et A_2 sont données par :

$$A_1 = \begin{bmatrix} C_1 & -S_1 & 0 & a_1 C_1 \\ S_1 & C_1 & 0 & a_1 S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{III.2.13})$$

$$A_2 = \begin{bmatrix} C_2 & -S_2 & 0 & a_2 C_2 \\ S_2 & C_2 & 0 & a_2 S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{III.2.14})$$

La matrice de transformation globale est donnée par :

$${}^0T_2 = \begin{bmatrix} C_{12} & -S_{12} & 0 & a_1 C_{12} + a_2 C_{12} \\ S_{12} & C_{12} & 0 & a_1 S_{12} + a_2 S_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{III.2.15})$$

III.2.6 Modélisation cinématique inverse :

La connaissance du modèle cinématique inverse est nécessaire pour le contrôle de la position et de l'orientation et c'est ce modèle qui se pose en pratique.

Le problème cinématique direct est toujours soluble. Par contre, le problème cinématique inverse n'est pas toujours soluble, ce qui signifie que l'on ne peut pas atteindre n'importe quelle position, sous une attitude quelconque, avec un robot donné. De plus on se trouve face à des équations ou à un système d'équations dont la résolution est généralement complexe.

Le problème géométrique inverse peut se résoudre par plusieurs méthodes. Dans ce qui suit, nous allons présenter l'une de ces méthodes qui est la méthode analytique, cette méthode fait appel aux transformations homogènes.

III.2.7 Présentation de la méthode de la transformation inverse:

Cette méthode se base sur l'utilisation des matrices de transformation homogène directe et inverse. La solution est obtenue d'une manière séquentielle en isolant chaque variable à chaque étape de calcul.

Pour un bras manipulateur à "n" degrés de liberté, le modèle géométrique inverse s'obtient en suivant les étapes suivantes :

Etape1

Lecture de la situation de l'organe terminal (matrice 0T_n).

Etape2

Evaluer la matrice de passage homogène correspondante à la situation désirée:

$${}^0T_n = A_1(q_1) \cdot A_2(q_2) \cdot \dots \cdot A_n(q_n) \quad (\text{III.2.16})$$

q_i : variables articulaires ($i=1,2,\dots,n$ avec $n \leq 6$)

Etape3

Voir s'il est possible de trouver une solution en identifiant la matrice de transformation homogène avec celle de la relation (III.2.16), sinon on multiplie les deux membres par $(A_i)^{-1}$ et chercher une solution possible.

$$(A_i)^{-1} (A_{i+1})^{-1} \dots (A_1)^{-1} {}^0T_n = {}^i T_n \quad (\text{III.2.17})$$

qui prend l'une des quatre formes suivantes :

- Première forme: $\alpha \cos(q_i) - \beta \sin(q_i) = 0$
deux solutions se présentent :

$$q_i = \text{atan2}(\alpha, \beta) \text{ et } q_i = q_i + 180^\circ$$

- Deuxième forme: $-\alpha \sin(q_i) + \beta \cos(q_i) = \gamma$, une solution n'existe pas que si : $\alpha^2 + \beta^2 \geq \gamma^2$, deux solutions sont possibles :

$$q_i = \text{atan2}(\beta, \alpha) - \text{atan}(\gamma, \pm \sqrt{\alpha^2 + \beta^2 - \gamma^2})$$

- Troisième forme: $\begin{cases} d_j \cdot \sin(q_i) = \alpha \\ d_j \cdot \cos(q_i) = \beta \end{cases}$

la solution unique est :

III.2.7 Présentation de la méthode de la transformation inverse:

Cette méthode se base sur l'utilisation des matrices de transformation homogène directe et inverse. La solution est obtenue d'une manière séquentielle en isolant chaque variable à chaque étape de calcul.

Pour un bras manipulateur à "n" degrés de liberté, le modèle géométrique inverse s'obtient en suivant les étapes suivantes :

Etape1

Lecture de la situation de l'organe terminal (matrice 0T_n).

Etape2

Evaluer la matrice de passage homogène correspondante à la situation désirée:

$${}^0T_n = A_1(q_1) \cdot A_2(q_2) \cdot \dots \cdot A_n(q_n) \quad (\text{III.2.16})$$

q_i : variables articulaires ($i=1,2,\dots,n$ avec $n \leq 6$)

Etape3

Voir s'il est possible de trouver une solution en identifiant la matrice de transformation homogène avec celle de la relation (III.2.16), sinon on multiplie les deux membres par $(A_i)^{-1}$ et chercher une solution possible.

$$(A_i)^{-1} (A_{i+1})^{-1} \dots (A_n)^{-1} {}^0T_n = {}^i T_n \quad (\text{III.2.17})$$

qui prend l'une des quatre formes suivantes :

- Première forme: $\alpha \cos(q_i) - \beta \sin(q_i) = 0$

deux solutions se présentent :

$$q_i = \text{atan2}(\alpha, \beta) \text{ et } q_i = q_i + 180^\circ$$

- Deuxième forme: $-\alpha \sin(q_i) + \beta \cos(q_i) = \gamma$, une solution n'existe pas que si : $\alpha^2 + \beta^2 \geq \gamma^2$, deux solutions sont possibles :

$$q_i = \text{atan2}(\beta, \alpha) - \text{atan}(\gamma, \pm \sqrt{\alpha^2 + \beta^2 - \gamma^2})$$

- Troisième forme:
$$\begin{cases} d_j \cdot \sin(q_i) = \alpha \\ d_j \cdot \cos(q_i) = \beta \end{cases}$$

la solution unique est :

Cette matrice est en fonction du vecteur d'orientation θ et non de la position. On passe alors à la seconde étape :

$$A_1^{-1}T_3 = A_2A_3 \quad (\text{III.2.19})$$

On trouve :

$$\begin{bmatrix} f_{11}(n) & f_{11}(o) & f_{11}(a) & f_{11}(P) \\ f_{12}(n) & f_{12}(o) & f_{12}(a) & f_{12}(P) \\ f_{13}(n) & f_{13}(o) & f_{13}(a) & f_{13}(P) \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -S_2 & 0 & C_2 & a_2C_2 \\ C_2 & 0 & S_2 & a_2S_2 \\ 0 & 1 & 0 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{III.2.20})$$

Avec :

$$\begin{cases} f_{11} = -xS_1 + yC_1 \\ f_{12} = z \\ f_{13} = xC_1 + yS_1 \end{cases}$$

On voit que même dans cette étape on ne peut pas isoler une variable en fonction du vecteur position, on continue la multiplication par A_2^{-1} , on obtient :

$$A_3 = A_2^{-1}A_1^{-1}T_3$$

$$\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_3 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} g_{11}(n) & g_{11}(o) & g_{11}(a) & g_{11}(P) - d_1C_2 - a_2S_{22} \\ g_{12}(n) & g_{12}(o) & g_{12}(a) & g_{12}(P) \\ g_{13}(n) & g_{13}(o) & g_{13}(a) & g_{13}(P) - d_1S_2 + a_2C_{22} \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{III.2.21})$$

Avec :

$$\begin{cases} g_{11} = -f_{11}S_2 + f_{12}C_2 \\ g_{12} = f_{13} \\ g_{13} = f_{11}C_2 + f_{12}S_2 \end{cases}$$

Egalant les éléments (2, 4), on aura : $g_{12}(P) = f_{13}(P) = 0$ cela donne :

$P_xC_1 + P_yS_1 = 0$ cette équation est de la première forme d'où : $\theta_1 = \text{atan2}(-P_x, P_y)$ ou $\theta_1 = \theta_1 + 180^\circ$.

Revenant à l'équation (III.2.20) et égalant les éléments (1, 4) et (2, 4), on aura :

$$\begin{cases} a_2C_2 = f_{11}(P) \\ a_2S_2 = f_{12}(P) \end{cases} \text{ la solution unique est : } \theta_2 = \text{atan2}(P_z, -S_1P_x + C_1P_y), \text{ avec } a_2 \text{ différent de}$$

zéro.

On trouve aisément la troisième variable d_3 du système (III.2.21) :

$$\begin{aligned} d_3 &= g_{13}(P) - d_1 S_2 + a_2 C_{22} \\ &= (-P_x S_1 + P_y C_1) C_2 + (P_z - d_1) S_2 + a_2 C_{22}. \end{aligned}$$

b) Le bras PUMA à 2 d.d.l :

La matrice de transformation globale de ce bras est :

$${}^0T_2 = \begin{bmatrix} C_{12} & -S_{12} & 0 & a_1 C_{12} + a_2 C_{12} \\ S_{12} & C_{12} & 0 & a_1 S_{12} + a_2 S_{12} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{III.2.22})$$

La position de l'outil terminal est donnée par le système suivant :

$$\begin{cases} P_x = a_1 C_{12} + a_2 C_1 \\ P_y = a_1 S_{12} + a_2 S_1 \end{cases}$$

On trouve alors les deux variables θ_1 et θ_2 données par :

$$\begin{aligned} \theta_1 &= a \tan 2[P_x(a_1 C_2 + a_2) - a_1 P_x S_2, P_x(a_1 C_2 + a_2) + a_1 P_y S_2] \\ \theta_2 &= a \tan 2(S_2, C_2) \end{aligned}$$

Avec :

$$\begin{cases} C_2 = \frac{P_x^2 + P_y^2 - a_1^2 - a_2^2}{2a_1 a_2} \\ S_2 = \pm \sqrt{1 - C_2^2} \end{cases}$$

III.2.9 Conclusion :

On peut modéliser l'orientation du bras manipulateur par d'autres méthodes, celles qui sont les plus intéressantes et les plus utilisées sont **les angles d'EULER** et la transformation **RPY [GOR 84]**.

Le modèle géométrique inverse s'impose dans la pratique, pour le résoudre on utilise la méthode analytique qui est une méthode trigonométrique, cette dernière fait appel aux transformations homogènes et nous permet d'obtenir tous les modèles géométriques inverses sous forme littérale. Cependant dès que le degré de couplage entre les différentes variables articulaires augmente la résolution qui est d'une certaine mesure intuitive devient difficile.

On peut résoudre le problème géométrique inverse par d'autres techniques, celles qui sont les plus utilisées sont les méthodes itératives [GOR 84], la méthode algébrique [CRA 95] et la méthode géométrique [CRA 95], [FU 87].

III.3 Génération des trajectoires

III.3.1 Introduction :

On s'intéresse maintenant aux méthodes de calcul des trajectoires dans l'espace tridimensionnel décrivant un mouvement désiré du bras manipulateur.

On considère le mouvement de l'outil terminal par rapport à la base.

Le problème essentiel est de déplacer l'outil terminal d'une position initiale à une position finale, en spécifiant des points intermédiaires que l'outil terminal doit passer par eux.

Pour avoir des mouvements lisses, on doit imposer certaines contraintes sur les qualités temporelles et spatiales.

III.3.2 Génération des trajectoires dans l'espace des variables articulaires :

Il est plus facile de générer des trajectoires dans l'espace articulaire que dans l'espace cartésien.

Pour chaque point intermédiaire correspond un ensemble de variables articulaires qui doivent être calculés en utilisant la cinématique inverse.

a) Interpolation des polynôme cubiques :

Pour le cas où il n'y a pas des points intermédiaires : quatre contraintes au moins sur la position et la vitesse relatives à la fonction au points initial et final.

$$\begin{cases} q(t_0) = q_0 \\ q(t_f) = q_f \\ \dot{q}(t_0) = 0 \\ \dot{q}(t_f) = 0 \end{cases} \quad (\text{III.3.1})$$

La condition de continuité implique que les vitesses initiale et finale soient nulles. Ces quatre conditions peuvent être satisfaites par un polynôme d'ordre au moins égal à trois :

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \quad (\text{III.3.2})$$

En résolvant le système (III.3.1), on trouve :

$$\begin{cases} a_0 = q_0 \\ a_1 = 0 \\ a_2 = 3(q_f - q_0) / t_f^2 \\ a_3 = -2(q_f - q_0) / t_f^3 \end{cases} \quad (\text{III.3.3})$$

Cette solution s'applique pour des vitesses initiale et finale nulles.

Si on a des points intermédiaires : pour chaque point correspond des valeur de variables articulaires qu'on doit les trouver en appliquant la cinématique inverse. On procède ensuite au calcul du polynôme.

En considère les conditions imposées (vitesses aux points intermédiaires ne sont pas nulles), on aura alors :

$$\begin{cases} \dot{q}(0) = \dot{q}_0 \\ \dot{q}(t_f) = \dot{q}_f \end{cases} \quad (\text{III.3.4})$$

ce qui donne :

$$\begin{cases} a_0 = q_0 \\ a_1 = \dot{q}_0 \\ a_2 = \frac{3(q_f - q_0)}{t_f^2} - \frac{2\dot{q}_0 + \dot{q}_f}{t_f} \\ a_3 = -\frac{2(q_f - q_0)}{t_f^3} + \frac{\dot{q}_f + \dot{q}_0}{t_f^2} \end{cases} \quad (\text{III.3.5})$$

Il existe diverses méthodes pour déterminer les vitesse aux points intermédiaires, parmi lesquelles on peut citer :

1. Le système choisit les points intermédiaires appliquant un heuristique convenable (selon le problème).
2. On choisit les points intermédiaires de sorte que l'accélération soit continue (vitesse aussi continue).

• **Remarque :**

Si on veut aquerir la position, la vitesse et l'accélération un polynôme de degré au moins égal à cinq est requis.

b) Interpolation par une fonction linéaire avec des limites paraboliques :

La discontinuité d'une ligne droite à ces extrémités entraîne une accélération infinie, et pour que la courbe totale soit continue en position et en vitesse on limite notre droite des deux extrémités par des régions paraboliques.

Au point b les vitesse sont égales :

$$\ddot{q}t_b = \frac{q_f - q_b}{t - t_b} \quad (\text{III.3.6})$$

t : durée totale.

La valeur de la variable articulaire est :

$$q_b = q_0 + \frac{\ddot{q} t_b^2}{2} \quad (\text{III.3.7})$$

ce qui donne :

$$\ddot{q} t_b^2 - \ddot{q} t_b + (q_f - q_0) = 0 \quad (\text{III.3.8})$$

si q_f , q_0 , t sont donnés, on peut générer la courbe par le choix de \ddot{q} et t_b .

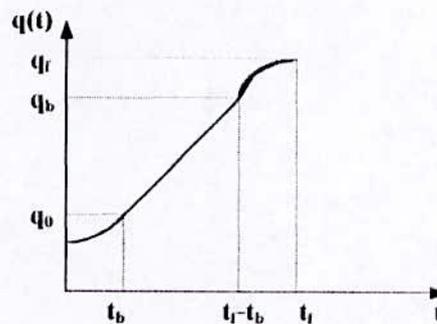


Figure III.3.1 : Trajectoire linéaire avec des limites paraboliques

En pratique, on choisit l'accélération \ddot{q} , et à partir de (III.3.8) on trouve :

$$t_b = \frac{\ddot{q} t - \sqrt{(\ddot{q} t)^2 - 4\ddot{q}(q_f - q_0)}}{2\ddot{q}} \quad (\text{III.3.9})$$

pour que t_b existe, il faut que :

$$\ddot{q} \geq \frac{4(q_f - q_0)}{t^2} \quad (\text{III.3.10})$$

L'égalité a lieu quand la partie linéaire a une longueur nulle.

Plus l'accélération est grande, plus la longueur des parties paraboliques est faible.

c) Génération des trajectoires cycloïdes :

La fonction cycloïde assure une continuité d'ordre deux pour commander l'accélération.

$$\begin{cases} q(t) = A(\omega t - \sin \omega t) + B \\ \dot{q}(t) = A\omega(1 - \cos \omega t) \\ \ddot{q}(t) = A\omega^2 \sin \omega t \end{cases} \quad (\text{III.3.11})$$

La condition de continuité impose que la vitesse et l'accélération soient nulles aux positions initiale et finale. Ceci donne :

$$\begin{cases} A = \frac{q_f - q_0}{2\pi} \\ B = q_0 \\ \omega t_f = 2\pi \end{cases} \quad (\text{III.3.12})$$

d'où :

$$q(t) = \frac{q_f - q_0}{2\pi} \left(2\pi \frac{t}{t_f} - \sin \left(2\pi \frac{t}{t_f} \right) \right) + q_0 \quad (\text{III.3.13})$$

III.3.3 Génération des trajectoires dans l'espace cartésien :

La génération des trajectoires est effectuée, en générale, dans l'espace des variables articulaires. La génération dans l'espace cartésien n'est utilisée que si l'application l'exige, et ce à cause des problèmes suivants:

- La génération dans l'espace cartésien nécessite un temps de calcul élevé.
- Difficulté d'interpoler à chaque échantillon la matrice de rotation pour donner une matrice valide.
- Les trajectoires cartésiennes font face à des problèmes variés relatifs à l'espace cartésien et aux singularités.
- Certaines trajectoires ne sont pas réalisables à cause des limitations en temps d'échantillonnage.

III.4 Modélisation dynamique

III.4.1 Introduction :

Après avoir défini la modélisation cinématique du bras manipulateur, on s'intéresse maintenant à l'élaboration du modèle dynamique des robots manipulateur à structure de chaîne simple.

Le modèle dynamique d'un bras manipulateur est l'ensemble des équations mathématiques décrivant le comportement dynamique du système, il a donc pour objet la commande des actionneurs de chaque articulation du système mécanique articulé.

On peut trouver différentes approches pour la détermination du modèle dynamique; on peut citer: le formalisme d'EULER-LAGRANGE (L-E), approche de NEWTON-EULER (N-E) et l'approche de D'ALEMBERT.

Dans ce qui suit nous allons présenter l'approche d'EULER-LAGRANGE.

III.4.2 Approche d'EULER-LAGRANGE (L-E) :

Le formalisme de (L-E) est utilisé pour modéliser le comportement dynamique d'un robot, ce formalisme associé à la représentation D-H, peut constituer un algorithme efficace et compact afin d'aboutir aux équations décrivant le mouvement du robot.

Le formalisme de mise en équations est assez pratique, car les énergies cinétiques et potentielles sont des grandeurs additives, et leur détermination est aisée.

L'équation de L-E est donnée par :

$$\frac{d}{dt} \left(\frac{\partial L}{\partial \dot{q}_i} \right) - \left(\frac{\partial L}{\partial q_i} \right) + \left(\frac{\partial E_d}{\partial \dot{q}_i} \right) = \Gamma_i \quad (\text{III 4.1})$$

$i=1, \dots, n$ pour un bras à "n" degrés de liberté.

Où:

L : fonction de Lagrange, $L = E_c - E_p$.

E_c : énergie cinétique totale du bras manipulateur.

E_p : énergie potentielle totale du bras.

E_d : fonction de dissipation de Rayleigh.

Γ_i : forces généralisées (effort actionneur).

Cette formulation conduit à écrire les mécanismes en mouvement par des équations dynamiques générales. Pour ceci on doit suivre les étapes suivantes :

- 1) Calcul de la vitesse de chaque point pour chaque segment.
- 2) Calcul de l'énergie cinétique pour chaque segment et l'énergie cinétique totale.
- 3) Calcul de l'énergie potentielle pour chaque segment et l'énergie potentielle totale.
- 4) Evaluation du Lagrangien "L".
- 5) Obtention des équations dynamiques par dérivation de "L".

III.4.2.1 Calcul de la vitesse d'un point du manipulateur :

On définit les coordonnées d'un point du segment (i) par :

$$r = T_i \cdot {}^i r_i \quad (\text{III.4.2})$$

r : position du point par rapport à la base.

${}^i r_i$: position du point par rapport au repère "i".

T_i : matrice de transformation reliant le repère "i" à celui de la base.

La vitesse de ce point est :

$$v = \frac{dr}{dt} = \left(\sum_{j=1}^i \frac{\partial T_i}{\partial q_j} \dot{q}_j \right) \cdot {}^i r \quad (\text{III.4.3})$$

et son carré vaut :

$$v^2 = \left(\frac{dr}{dt} \right)^2 = \text{Trace} \left[\left(\frac{dr}{dt} \right) \otimes \left(\frac{dr}{dt} \right)^T \right]$$

La trace d'une matrice est la somme de ses éléments diagonaux, d'où :

$$v^2 = \text{Trace} \left[\sum_{j=1}^i \sum_{k=1}^i \frac{\partial T_i}{\partial q_j} \cdot ({}^i r) \otimes ({}^i r)^T \left(\frac{\partial T_i}{\partial q_k} \right)^T \cdot \dot{q}_j \cdot \dot{q}_k \right]$$

III.4.2.2 L'énergie cinétique du bras manipulateur :

L'énergie cinétique d'une particule de masse dm d'un segment "i" est :

$$\begin{aligned} dK_i &= \frac{1}{2} \left(\frac{dr}{dt} \right)^2 \cdot dm \\ &= \frac{1}{2} \text{Trace} \left[\sum_{j=1}^i \sum_{k=1}^i \frac{\partial T_i}{\partial q_j} ({}^i r \otimes {}^i r)^T \cdot dm \left(\frac{\partial T_i}{\partial q_k} \right)^T \cdot \dot{q}_j \cdot \dot{q}_k \right] \end{aligned}$$

L'énergie cinétique du segment (i) sera :

$$K_i = \int dK_i = \frac{1}{2} \text{Trace} \left[\sum_{j=1}^i \sum_{k=1}^i \frac{\partial T_i}{\partial q_j} \left(\int {}^i r \otimes {}^i r^T \cdot dm \right) \left(\frac{\partial T_i}{\partial q_k} \right)^T \cdot \dot{q}_j \cdot \dot{q}_k \right] \quad (\text{III.4.4})$$

L'intégrale qui apparaît dans l'équation ci dessus est appelée matrice pseudo-inertie du segment (i). Elle sera notée J_i .

$$J_i = \begin{bmatrix} \int x_i^2 dm & \int x_i y_i dm & \int x_i z_i dm & \int x_i dm \\ \int x_i y_i dm & \int y_i^2 dm & \int y_i z_i dm & \int y_i dm \\ \int x_i z_i dm & \int z_i y_i dm & \int z_i^2 dm & \int z_i dm \\ \int x_i dm & \int y_i dm & \int z_i dm & \int dm \end{bmatrix} \quad (III.4.5)$$

On peut exprimer cette matrice à l'aide des moments d'inertie de la liaison (i) par rapport aux principaux axes du référentiel $R_i(x_i, y_i, z_i)$.

$$J_i = \begin{bmatrix} \frac{-I_{xx} + I_{yy} + I_{zz}}{2} & I_{xy} & I_{xz} & m_i \bar{x}_i \\ I_{xy} & \frac{I_{xx} - I_{yy} + I_{zz}}{2} & I_{yz} & m_i \bar{y}_i \\ I_{xz} & I_{yz} & \frac{I_{xx} + I_{yy} - I_{zz}}{2} & m_i \bar{z}_i \\ m_i \bar{x}_i & m_i \bar{y}_i & m_i \bar{z}_i & m_i \end{bmatrix} \quad (III.4.6)$$

L'énergie cinétique totale du bras manipulateur à "n" segments est donnée par :

$$K = \frac{1}{2} \sum_{i=1}^n \text{Trace} \left[\sum_{j=1}^i \sum_{k=1}^i \frac{\partial T_i}{\partial q_j} \cdot J_i \cdot \left(\frac{\partial T_i}{\partial q_k} \right)^T \cdot \dot{q}_j \cdot \dot{q}_k \right] \quad (III.4.7)$$

On inclue aussi l'énergie cinétique liée à l'actionneur du segment (i) donnée par :

$$K_{\text{actionneur}(i)} = 1/2 \cdot I_{ai} \cdot \dot{q}_i^2 \quad (III.4.8)$$

I_{ai} : moment d'inertie de l'actionneur (i).

L'énergie cinétique totale du bras est :

$$E_c = K = \frac{1}{2} \sum_{i=1}^n \text{Trace} \left[\sum_{j=1}^i \sum_{k=1}^i \frac{\partial T_i}{\partial q_j} \cdot J_i \cdot \left(\frac{\partial T_i}{\partial q_k} \right)^T \cdot \dot{q}_j \cdot \dot{q}_k \right] + \frac{1}{2} \sum_{i=1}^n I_{ai} \cdot \dot{q}_i^2 \quad (III.4.9)$$

III.4.2.3 L'énergie potentielle du bras manipulateur :

L'énergie potentielle P_i du segment (i) est donnée par :

$$P_i = -m_i \cdot g^T \cdot r_i = -m_i \cdot g^T \cdot T_i^{-1} \cdot r_i \quad (III.4.10)$$

où: r_i représente dans ce cas le centre de masse du segment (i) par rapport au repère de base.

$g = [g_x, g_y, g_z]^T$: champ de la pesanteur

L'énergie potentielle totale du manipulateur sera :

$$E_p = P = \sum_{i=1}^n -m_i \cdot g^T \cdot T_i^{-1} r_i \quad (III.4.11)$$

III.4.2.4 Le LAGRANGIEN :

La fonction de Lagrange est définie par $L = E_c - E_p$, elle sera donc :

$$L = \frac{1}{2} \sum_{i=1}^n \text{Trace} \left[\sum_{j=1}^i \sum_{k=1}^i \frac{\partial T_i}{\partial q_j} \cdot J_i \left(\frac{\partial T_i}{\partial q_k} \right)^T \cdot \dot{q}_j \cdot \dot{q}_k \right] + \frac{1}{2} \sum_{i=1}^n I_{ai} \dot{q}_i^2 + \sum_{i=1}^n m_i \cdot g^T \cdot T_i^{-1} r_i \quad (III.4.12)$$

III.4.2.5 L'équation dynamique du bras manipulateur :

La fonction de dissipation de Rayleigh est donnée par :

$$E_d = \frac{1}{2} \sum_{i=1}^n f_i (\dot{q})^2 \quad (III.4.13)$$

f_i : coefficient de frottement visqueux ramené à l'articulation (i).

Pour le premier terme de l'équation de L-E, $\frac{\partial T_p}{\partial q_j} = 0$ on peut écrire

$$\begin{aligned} \frac{d}{dt} \frac{\partial L}{\partial \dot{q}_p} &= \sum_{i=p}^n \sum_{k=1}^i \text{Trace} \left(\frac{\partial T_i}{\partial q_p} J_i \left(\frac{\partial T_i}{\partial q_k} \right)^T \right) \ddot{q}_k + I_{ap} \ddot{q}_p + \sum_{i=p}^n \sum_{k=1}^i \sum_{m=1}^i \text{Trace} \left(\frac{\partial^2 T_i}{\partial q_m \partial q_p} J_i \left(\frac{\partial T_i}{\partial q_k} \right)^T \right) \dot{q}_j \dot{q}_k \\ &+ \sum_{i=p}^n \sum_{k=1}^i \sum_{m=1}^i \text{Trace} \left(\frac{\partial T_i}{\partial q_m \partial q_k} J_i \left(\frac{\partial T_i}{\partial q_p} \right)^T \right) \dot{q}_k \dot{q}_m \end{aligned} \quad (III.4.14)$$

Le second terme de l'équation de L-E :

$$\frac{\partial L}{\partial q_p} = \sum_{i=p}^n \sum_{k=1}^i \sum_{j=1}^i \text{Trace} \left(\frac{\partial^2 T_i}{\partial q_j \partial q_p} J_i \left(\frac{\partial T_i}{\partial q_k} \right)^T \right) \dot{q}_j \dot{q}_k + \sum_{i=p}^n m_i g^T \frac{\partial T_i}{\partial q_p} r_i \quad (III.4.15)$$

A partir des équations (III.4.1), (III.4.14), (III.4.15), on peut calculer l'équation dynamique générale :

$$\Gamma_i = \sum_{j=1}^n D_{ij} \ddot{q}_j + I_{ai} \ddot{q}_i + f_i \dot{q}_i + \sum_{j=1}^n D_{ijk} \dot{q}_j \dot{q}_k + D_i \quad (III.4.16)$$

On fait le changement d'indice suivant : "p", "i", "k" par "i", "p", "j" respectivement, on aura:

$$D_{ij} = \sum_{p=\max(i,j)}^n \text{Trace} \left(\frac{\partial T_p}{\partial q_j} J_p \left(\frac{\partial T_p}{\partial q_i} \right)^T \right) \quad (\text{III.4.17})$$

$$D_{ijk} = \sum_{p=\max(i,j)}^n \text{trcace} \left(\frac{\partial^2 T_p}{\partial q_j \partial q_k} J_p \left(\frac{\partial T_p}{\partial q_i} \right)^T \right) \quad (\text{III.4.18})$$

$$D_i = - \sum_i^n m_p g^T \frac{\partial T_p}{\partial q_i} r_p \quad (\text{III.4.19})$$

• **Remarques :**

1) $D_{ij} = D_{ji}$

2) $D_{ijk} = D_{ikj}$

3) Il est clair que D_{ij} , D_{ijk} , D_i dépendent de l'inertie des segments, des positions et des vitesses articulaires.

On peut interpréter physiquement les termes D_i , D_{ij} , D_{ijk} comme suit :

- D_{ii} : Inertie propre que présente l'articulation (i).

- D_{ij} : Inertie entre les articulations (i) et (j) est interprétée comme un effet d'accélération angulaire sur l'articulation (i) causé par l'effort Γ_i , et inversement, il est fonction des positions relatives des deux segments.

- D_{ijk} : Couple de réaction exercer sur l'articulation (i) par les vitesses angulaires des articulations (j) et (k). Pour $j=k$, il s'agira de la force centrifuge appliquée sur l'articulation (i) (force de coriolis générée par les vitesses angulaires des articulations (j) et (k)).

- D_i : Effet de la gravitation sur l'articulation (i).

4) Ecriture matricielle :

On peut écrire le modèle dynamique sous la forme matricielle suivante :

$$T = H(q)\ddot{q} + h(q, \dot{q}) + g(q) \quad (\text{III.4.20})$$

Avec :

$T = [\Gamma_1, \Gamma_2, \dots, \Gamma_n]^T$ représente le vecteur des efforts actionneurs.

$H(q) = [D_{ij}(q)]$ matrice d'inertie (n x n).

$h(q, \dot{q}) = \dot{q}^T C(q) \dot{q}$ où : $C(q) = [D_{ijk}]$ représente les termes centrifuges et de coriolis.

$g(q)$ vecteur (n x 1) des efforts gravitationnels.

III.4.3 Modélisation dynamique de quelques bras manipulateurs :

Dans cette partie on va donner les modèles dynamiques du bras à un degré de liberté et du bras PUMA à 2 d.d.l.

Le modèle dynamique du bras à un degré de liberté (articulation rotative) est donné par:

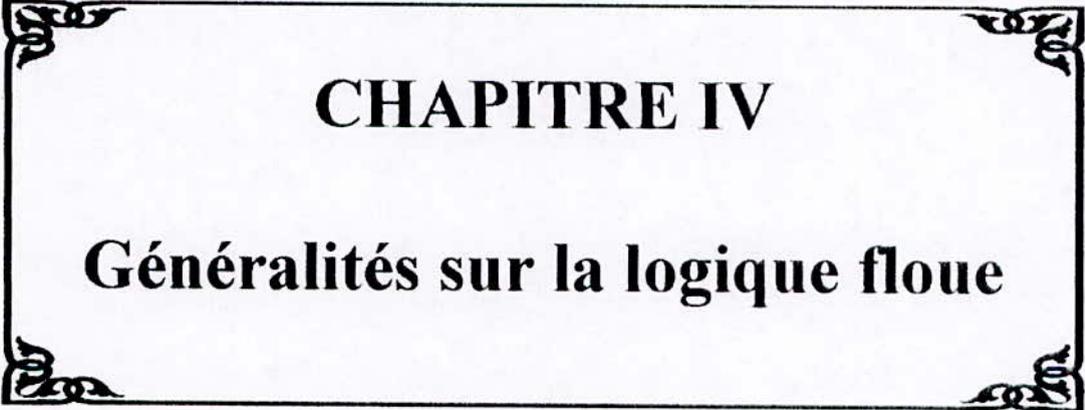
$$\Gamma = \frac{1}{3} m l^2 \ddot{q} + b \dot{q} + \frac{1}{2} m g l C \quad (\text{III.4.21})$$

Le modèle dynamique du bras PUMA à 2 d.d.l est donné par les formules suivantes :

$$\begin{aligned} \Gamma_1 = & \left(\frac{1}{3} m_1 l^2 + \frac{4}{3} m_2 l^2 + m_2 l^2 C_2 \right) \ddot{q}_1 + \left(\frac{1}{3} m_2 l^2 + \frac{1}{2} m_2 l^2 C_2 \right) \ddot{q}_2 - m_2 l^2 S_2 \dot{q}_1 \dot{q}_2 \\ & - \frac{1}{2} m_2 l^2 S_2 \dot{q}_2^2 + \left(m_1 g \frac{1}{2} + m_2 g l \right) + \frac{1}{2} m_2 g l C_{12} + b_1 \dot{q}_1 \end{aligned} \quad (\text{III.4.22})$$

$$\begin{aligned} \Gamma_2 = & \left(\frac{1}{3} m_2 l^2 + \frac{1}{2} m_2 l^2 C_2 \right) \ddot{q}_1 + \frac{1}{3} m_2 l^2 \ddot{q}_2 + \frac{1}{2} m_2 l^2 S_2 \dot{q}_1 \dot{q}_2 \\ & + \frac{1}{2} m_2 g l C_{121} + b_2 \dot{q}_1 \end{aligned} \quad (\text{III.4.23})$$

Les différentes étapes de calcul intermédiaires pour les bras à un ddl et 2 ddl sont données dans les annexes A1 et A2 respectivement.



CHAPITRE IV

Généralités sur la logique floue

Généralités sur la logique floue

IV-1 Introduction

Les premières bases de la logique floue ont été introduites par le professeur LA.ZADEH. Cette méthode était de nature différente de ce qui existait comme méthode de traitement sur les ordinateurs existant à l'époque (fonctionnement en 0 ou en 1).

Les premières applications des concepts de la logique floue, dans le domaine de réglage, ont commencé dès 1975, et ce n'est qu'à partir d'environ 1985 que les Japonais commençaient à l'introduire dans le domaine industriel pour des problèmes de réglages et de commande.

Plus récemment, cette méthode a commencé à prendre sa place en Europe dans le domaine de la recherche en technologie.

Afin de mettre en évidence le principe fondamental de cette technique, on présente un exemple simple, celui de la classification des personnes en trois ensembles, « jeune », « entre deux âges » et « âgé ».

Selon la logique classique (logique de Boole), qui n'admet que les deux variables 0 et 1, une telle classification pourrait se faire comme le montre la figure suivante :

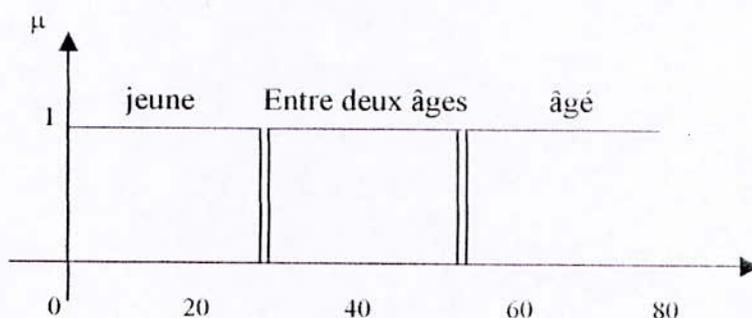


Figure IV-1 : Classification des personnes selon la logique classique.

Toutes les personnes âgées de moins de 30 ans sont alors considérées comme appartenant à l'ensemble « jeune » et toutes les personnes âgées de plus de 50 ans comme appartenant à l'ensemble « âgé ».

Cependant, une telle classification n'est même pas logique. Pourquoi une personne de 50 ans, doit-elle être considérée comme appartenant à l'ensemble « âgé » ? En réalité, un tel passage se fait progressivement et individuellement.

La logique floue, dont les variables peuvent prendre n'importe quelle valeur entre 0 et 1, permet de tenir compte de cette réalité. Les limites ne varient pas soudainement, mais progressivement.

La figure (IV-2) montre une classification possible pour l'ensemble précédent, à l'aide de la logique floue.

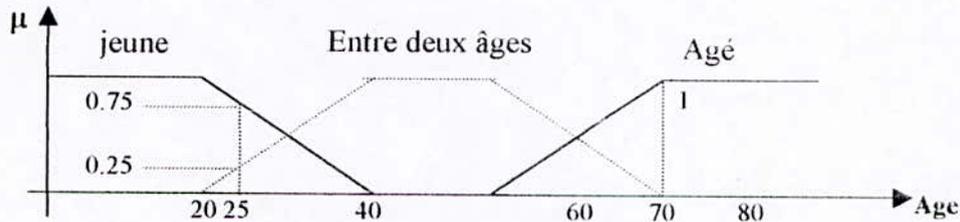


Figure IV-2 : classification des personnes en trois ensembles selon la logique

Le paramètre 'âge' est une variable floue, la figure (IV-2) nous donne la définition de cette variable par les ensembles « jeune », « entre deux âges » et « âgé », qui sont représentés par les fonctions d'appartenance, définissant la variable floue 'âge'. La valeur de ces ensembles est comprise entre 0 et 1.

Ainsi, une personne de 25 ans appartient à l'ensemble « jeune » avec une valeur $\mu = 0.75$ de la fonction d'appartenance et à l'ensemble « entre deux âges » avec $\mu = 0.25$. Par contre une personne de 70 ans appartient avec une valeur de $\mu = 1$ de la fonction d'appartenance à l'ensemble « âgé ». Evidemment, la forme des fonctions d'appartenance n'est pas rigide et peut dépendre du contexte envisagé.

La structure de l'algorithme de réglage par logique floue se présente sous forme d'une série de règles linguistiques de la forme :

Si alors ,ou,
Si alors ,ou,

⋮
⋮
⋮

Les éléments de base de la logique floue sont :

- les variables linguistiques
- la définition des variables linguistiques
- les déductions floues (inférences)
- les opérateurs.

IV-2 Variables linguistiques

Les variables linguistiques sont des expressions floues comme :

Quelque, beaucoup, souvent, ...

Chaud, froid, ...

Grand, petit, ...etc.

IV-3 Définition des variables linguistiques

Afin de permettre un traitement numérique, il est indispensable de soumettre les variables linguistiques à une définition à l'aide des fonctions d'appartenance.

On attribue à chaque valeur de la variable linguistique des fonctions d'appartenance μ , dont la valeur varie entre 0 et 1 – comme on a déjà vu pour l'exemple précédent – en tenant compte de la classification en un certain nombre d'ensembles flous par $\mu_E(x)$ où :

E : indique l'ensemble concerné (NB, ZE, PB, ...)

x : variable linguistique

Une valeur précise de $\mu_E(x)$ liée à une valeur déterminée de la variable x sera désignée par : facteur d'appartenance.

IV-4 Fonction d'appartenance :

Le plus souvent on utilise pour les fonctions d'appartenance des formes trapézoïdales ou triangulaires composées par des morceaux de droites.

L'allure de la fonction d'appartenance est complètement définie par trois points P_1 , P_2 et P_3 , pour la forme triangulaire (figure IV-3) ; et par quatre points P_1 , P_2 , P_3 et P_4 pour la forme trapézoïdale (figure IV-4).

Le triangle peut être considéré comme un cas particulier du trapèze lorsque $P_2 = P_3$.

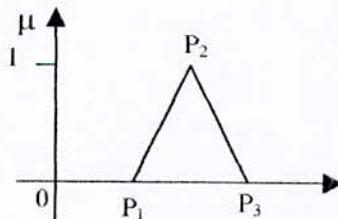


Figure IV-3 : forme triangulaire

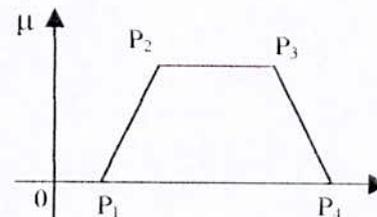


Figure IV-4 : forme trapézoïdale

Cependant, il existe d'autres formes pour les fonction d'appartenance [BUH 94].

IV-5 Dédutions floues (inférences)

En général, plusieurs valeurs de variables linguistiques convenablement définies par des fonctions d'appartenance, sont liées entre elles par de règles, afin de tirer des conclusions. On parle alors de déductions floues ou inférences.

Les règles peuvent être exprimées sous la forme générale :

Opération := Si condition1 alors opération1 ou

Si condition2 alors opération2 ou

.

Si condition m alors opération m .

Les conditions peuvent dépendre d'une ou de plusieurs variables. Dans ce deuxième cas, les variables sont liées entre elles par des opérations de la logique floue de forme OU et ET.

IV-6 Opérateurs de la logique floue

Les opérateurs ET ou OU sont, le plus souvent, réalisés respectivement par les règles 'min' et 'max'.

De plus, il existe un opérateur NON (complément, négation, inverse).

a) Opérateur NON

Cet opérateur désigne l'ensemble complémentaire, et est exprimé par les fonctions d'appartenance de la manière suivante :

$$\mu_c(x) = 1 - \mu_a(x). \quad (IV.1)$$

La figure (IV-5) met en évidence cette relation :

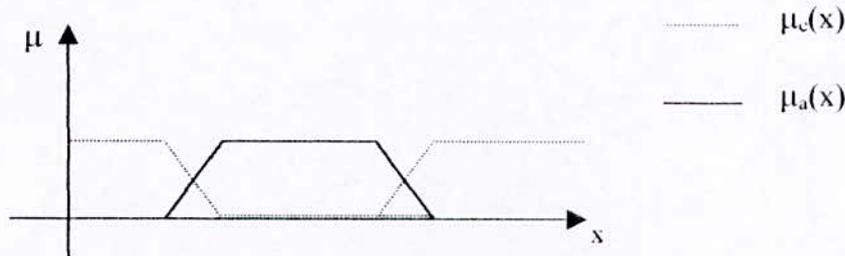


Figure IV-5 : opérateur NON

b) Opérateur ET

Cet opérateur est réalisé dans la plupart des cas par la formation du minimum, appliquée aux fonctions d'appartenance $\mu_a(x)$ et $\mu_b(x)$ des deux ensembles a et b, à savoir :

$$\mu_c(x) = \min [\mu_a(x), \mu_b(x)]. \quad (IV.2)$$

On parle alors d'opérateur minimum.

La figure (IV-6) met en évidence cette relation :

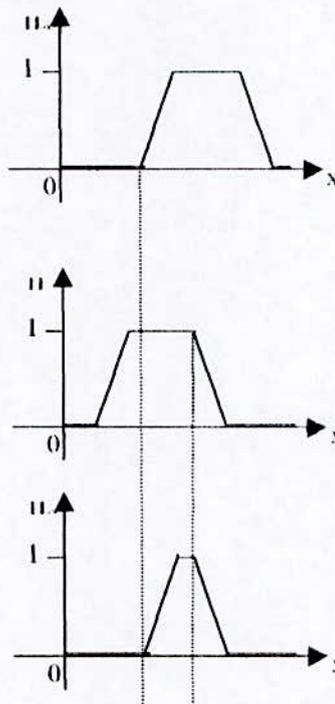


Figure IV-6 : opérateur ET (réalisé par le

c) Opérateur OU

Cet opérateur est réalisé en général par la formation du maximum, appliqué aux fonctions d'appartenance $\mu_a(x)$ et $\mu_b(x)$ des deux ensembles a et b. On a donc l'opérateur maximum :

$$\mu_c(x) = \max [\mu_a(x) , \mu_b(x)] . \tag{IV.3}$$

La figure (IV-7) montre cette relation :

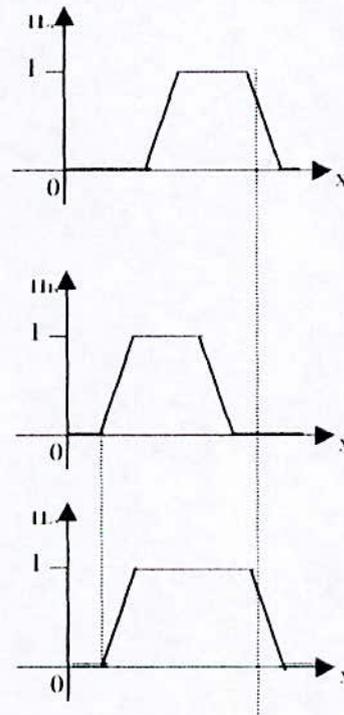


Figure IV-7 : opérateur OU (réalisé par le maximum)

Il existe d'autres réalisations pour les opérateurs ET et OU, pour en savoir plus voir [BUH94].

IV-7 Réglage par logique floue

IV-7-1 Procédé de la conception d'un réglage par logique floue

Le procédé à suivre lors de la conception d'un réglage par logique floue est présenté par la figure suivante (figure IV-8) :

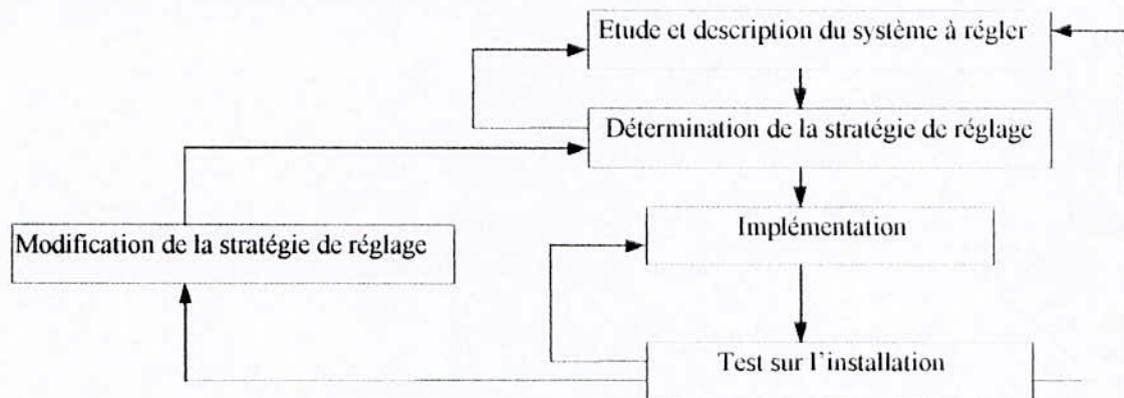


Figure IV-8: Procédé lors de la conception de réglage par logique floue

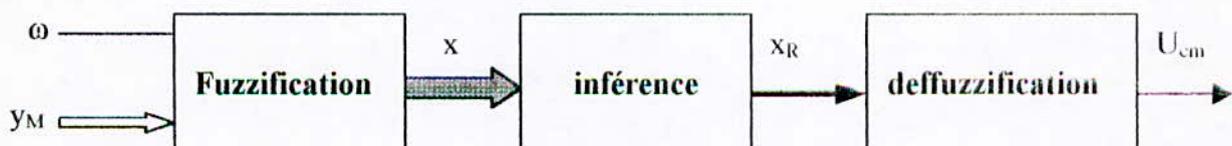
IV-7-2 Propriétés d'un réglage par logique floue

Les principales propriétés d'un réglages par logique floue sont :

- Mesure de plusieurs variables (multivariables).
- Comportement non linéaire.
- Aptitude à régler convenablement surtout des systèmes à régler avec un comportement dynamique compliqué, dont la modélisation est difficile, voir impossible.
- Modélisation du système superflue (pouvant tout de même être utile).
- Connaissance approximative des paramètres du système à régler.

IV-7-3 Régulateur par logique

Le régulateur par logique floue est formé de trois parties :



ω : grandeur de consigne.

Y_M : grandeur à régler et d'autres grandeurs pour saisir la dynamique du système à régler.

x : vecteur de variables floues.

x_R : variable floue de sortie.

U_{cm} : signal de commande.

1. Fuzzification : consiste à traiter les grandeurs d'entrée par des variables linguistiques. Ce qui nécessite leur définition par des fonctions d'appartenance.

La fuzzification fournit une série de variables floues réunies par le vecteur x .

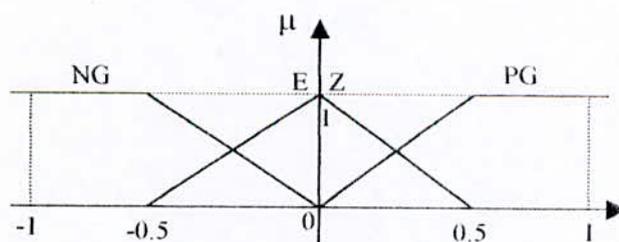
2. Inférence : les valeurs des variables linguistiques sont liées par plusieurs règles à l'aide des opérateurs de logique floue.

3. Defuzzification : consiste à transformer la valeur x_R , fournie par le bloc inférence, en une valeur déterminée.

IV-7-4 Fuzzification :

La fuzzification consiste à définir les fonctions d'appartenance pour les différentes variables. En particulier pour les variables d'entrée qui peuvent alors être traitées par les inférences. On utilise en général des formes trapézoïdales et des formes triangulaires, qui sont, dans la plupart des cas, suffisantes pour délimiter des ensembles flous.

Dans ce but, des grandeurs sont réduites à des grandeurs normalisées x ($-1 \leq x \leq +1$). En général, on introduit pour une variable x trois, cinq ou sept ensembles représentées par des fonctions d'appartenance comme le montre la figure IV-9 pour un cas de trois ensembles.



FigIV-9 : Fuzzification avec trois fonctions d'appartenance

La forme des fonctions d'appartenance peut ne pas être symétrique par rapport à $x=0$, ou non équidistante lorsque les maxima des fonctions d'appartenance des différents ensembles ne sont pas écartés de manière équidistante.

IV-7-5 Inférence

On peut exprimer les inférences par différentes possibilités, à savoir :

- description linguistique

- description symbolique
- description par matrice d'inférence
- description par tableau d'inférence

Pour le réglage par logique floue, et afin de traiter numériquement les inférences, on utilise en général une des méthodes suivantes :

i) Méthode d'inférence max-min

Cette méthode peut être décrite en toute généralité de la manière suivante :
A la condition de chaque règle R_i ($i=1, \dots, m$) est attribué un facteur d'appartenance μ_{Ci} ; si la condition n'est pas du tout vérifiée, on a évidemment $\mu_{Ci} = 0$.

Noter que les opérateurs ET et OU sont réalisés respectivement par la formation du minimum et du maximum.

Pour chaque règle on obtient la fonction d'appartenance partielle par la relation :

$$\mu_{Ri}(x_R) = \min(\mu_{Ci}, \mu_{oi}(x_R)) \quad i = 1, \dots, m. \quad (IV.4)$$

où : μ_{oi} : est la fonction d'appartenance liée à l'opération imposée par la règle.

La fonction d'appartenance résultante est donnée par :

$$\mu_{res}(x_R) = \text{Max} [\mu_{Ri}(x_R)]. \quad (IV.5)$$

ii) Méthode d'inférence max-prod

Le principe de la méthode est le même que pour la méthode précédente (max-min) ; mais cette fois-ci, la conclusion dans chaque règle, introduite par 'alors', qui lie le facteur d'appartenance de la condition avec la fonction d'appartenance de la variable de sortie x_R par l'opérateur ET est réalisé cette fois par la formation du produit.

Pour chaque règle, la fonction d'appartenance partielle $\mu_{Ri}(x_R)$ est donnée par la relation :

$$\mu_{Ri}(x_R) = \mu_{Ci} \cdot \mu_{oi}(x_R) \quad i = 1, \dots, m. \quad (IV.6)$$

Pour la fonction d'appartenance résultante, on obtient :

$$\mu_{res}(x_R) = \text{Max} [\mu_{Ri}(x_R)]. \quad (IV.7)$$

iii) Méthode d'inférence somme-prod

Dans ce cas, le 'alors' de chaque règle est réalisé par la formation du facteur d'appartenance μ_{Ci} de la condition avec la fonction d'appartenance de la variable de sortie par l'opérateur ET et le OU liant les règles est réalisé par la formation de la somme, plus précisément par la valeur moyenne.

On obtient la fonction d'appartenance partielle $\mu_{Ri}(x_R)$ de chaque règle par la relation :

$$\mu_{Ri}(x_R) = \mu_{Ci} \cdot \mu_{oi}(x_R) \quad i = 1, \dots, m. \quad (IV.8)$$

En fin la fonction d'appartenance résultante est donnée par :

$$\mu_{res}(x_R) = [\mu_{R1}(x_R) + \dots + \mu_{Rm}(x_R)] / m. \quad (IV.9)$$

m : nombre de règles intervenant dans l'inférence.

IV-7-6 Defuzzification :

Elle consiste à transformer l'information fournie par le bloc d'inférence [$\mu_{res}(x_R)$] en une information déterminée x_R^* .

La méthode de defuzzification la plus utilisée est celle de la détermination du centre de gravité x_R^* de la fonction d'appartenance résultante $\mu_{res}(x_R)$. Il est déterminé par la relation :

$$x_R^* = \frac{\int_{-1}^1 x_R \cdot \mu_{res}(x_R) dx_R}{\int_{-1}^1 \mu_{res}(x_R) dx_R} \quad (IV.10)$$

Ce calcul est notablement simplifié lorsque la fonction d'appartenance résultante est déterminée par la méthode d'inférence somme-prod :

Le centre de gravité est donné par la relation :

$$x_R^* = \frac{\sum_{i=1}^m \mu_{Ci} \cdot x_{Ri}^* \cdot S_i}{\sum \mu_{Ci} \cdot S_i} \quad (IV.11)$$

où : $S_i = \int_{-1}^1 \mu_{0i}(x_R) dx_R$ est la surface de la fonction d'appartenance $\mu_{0i}(x_R)$.

$x_{Ri}^* = \frac{1}{S_i} \int_{-1}^1 x_R \cdot \mu_{0i}(x_R) dx_R$ est le centre de gravité de S_i .

si les fonctions d'appartenance ne sont pas chevauchées, le centre de gravité donné par la relation précédente devient :

$$x_R^* = \frac{\sum \mu_{CE} \cdot x_{Ei}^* \cdot S_{Ei}}{\sum \mu_{CE} \cdot S_{Ei}} \quad (IV.12)$$

où : S_{Ei} désigne l'ensemble de surface de chaque ensemble flou.

x_{Ei}^* : centre de S_{Ei} .

Pour la méthode d'inférence max-min et max-prod :

$$\mu_{CE} = \text{Max}[\mu_{CE}]. \quad (IV.13)$$

Pour la méthode d'inférence somme-prod :

$$\mu_{CE} = \frac{1}{m} \sum_{i=1}^m \mu_{CEi} \quad (IV.14)$$

- Centre de gravité par la méthode des hauteurs pondérées

Cette méthode est un cas particulier des fonctions d'appartenance sans chevauchement. Dans ce cas on associe à chaque ensemble une barre (rectangle de largeur étroite) située à l'abscisse x^*_E . La hauteur de chaque barre est pondérée par le facteur d'appartenance μ_{CEi} à déterminer soit par la relation (IV-13) ou (IV.14).

On obtient la formule suivante dans le cas de m règles :

$$x^*_R = \frac{\sum_{i=1}^m \mu_{CEi} \cdot x^*_E}{\sum_{i=1}^m \mu_{CEi}} \quad (IV-15)$$

Dans ce cas aussi les sommations doivent être étendues à tous les ensembles de la variable de sortie x_R .

- Méthode des hauteurs pondérées modifiées

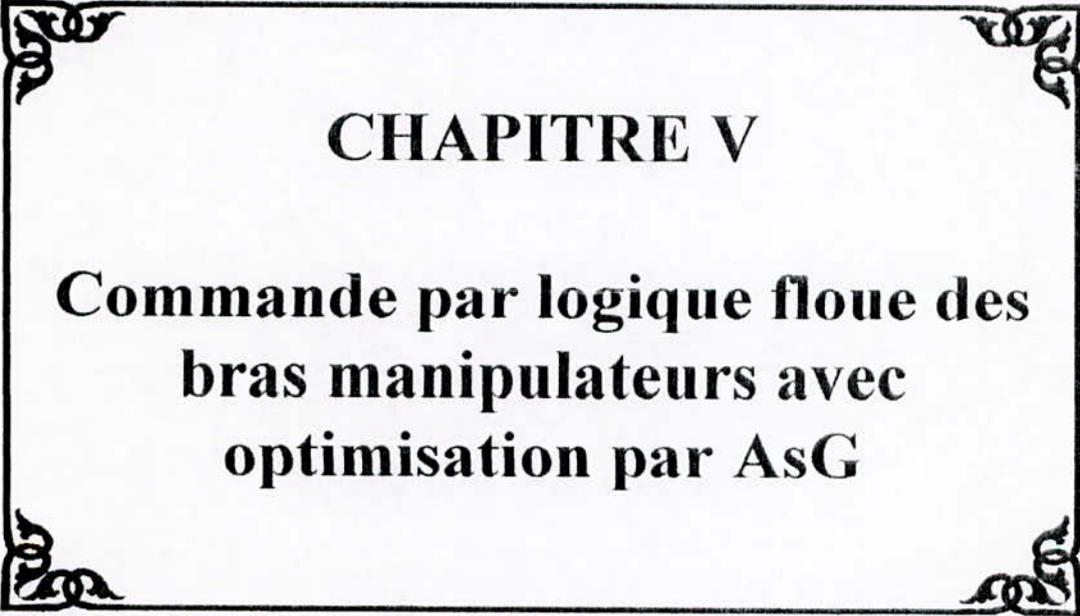
Il s'agit de la version améliorée de la méthode précédente. Ici, le calcul de la fonction d'appartenance résultante est effectué ; et un coefficient de correction s est introduit. La formule (IV.15) devient :

$$x^*_R = \frac{\sum_{i=1}^m (\mu_{CEi} \cdot x^*_E / \delta^2_E)}{\sum_{i=1}^m (\mu_{CEi} / \delta^2_E)} \quad (IV-16)$$

Le coefficient δ mesure l'étendue de la fonction d'appartenance de la variable de sortie associée à la règle R_i .

- Méthode de la moyenne des maxima

Avec cette méthode, on détermine d'abord dans l'ensemble flou E issu de l'inférence R_i les valeurs de x pour lesquelles $\mu_E(x)$ est un maximum, ensuite on détermine la commande en faisant la moyenne de ces valeurs.



CHAPITRE V

**Commande par logique floue des
bras manipulateurs avec
optimisation par AsG**

Commande par logique floue des bras manipulateurs avec optimisation par AsG

V.1 Introduction:

L'utilisation d'un contrôleur flou (ou par logique floue) est, en réalité, très pratique si on a à commander un système couplé et non linéaire.

En général, c'est le contrôleur du type 'Mamdani' qui est le plus utilisé, spécialement, lorsqu'il s'agit d'un processus industriel lent [FAL 98].

V.2 Choix d'un contrôleur:

Après avoir fixé le type de contrôleur à utiliser, on doit choisir les caractéristiques suivantes:

- La forme des fonctions d'appartenance (triangulaire, trapézoïdale ou autre), ainsi que leur distribution sur l'axe des variables linguistiques.
- Les règles d'inférence utilisées.
- La méthode de déffuzification.

V.3 Schéma de contrôle :

C'est le schéma proposé par Mamdani présenté dans la figure ci-dessous [FAL 98] :

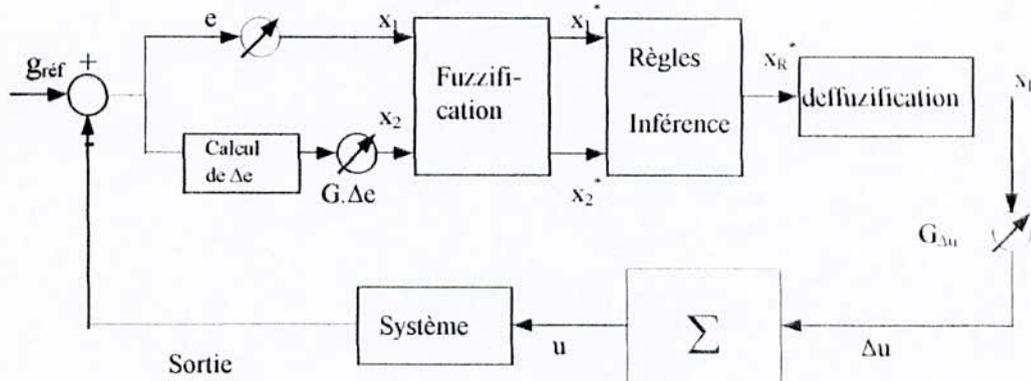


Figure V-1: Schéma bloc d'une boucle de régulation par contrôleur flou, type Mamdani.

Le système est composé des éléments suivants:

- Un contrôleur flou ayant une entrée recevant l'erreur qui est l'écart entre la grandeur de consigne et la sortie du système.
- Un bloc de calcul de la variation de l'erreur.
- Des gains variables appliqués à l'erreur et à sa variation afin de les maintenir dans la gamme des variables linguistiques (généralement entre -1 et 1).
- Un bloc de fuzzification x_1 et x_2 (erreur et sa dérivée normalisées).
- Une base des règles d'inférence.

- Les inférences traitant les différents facteurs d'appartenance utilisant les règles déjà mentionnées.
- Un bloc de défuzzification convertissant la variable linguistique de sortie en une valeur précise.
- Un gain ajustable pour donner la variation de la commande Δu à appliquer.
- Le système à contrôler (machine, bras de robot, processus industriel, ...) précédé d'un sommateur ajoutant la variation de la commande à la commande précédente u appliquée au système à contrôler.

V.4 Algorithme de contrôle flou :

Les règles utilisées pour traiter les valeurs des facteurs d'appartenance (et donc les variables linguistiques), découlent d'une logique simple et très facile à mettre en œuvre. L'ensemble des règles permet de dresser une matrice ou un tableau d'inférence représentant la logique de fonctionnement du contrôleur flou.

Une fois les grandeurs d'entrée fuzzifiées, elles possèdent une valeur numérique pour chaque sous-ensemble flou (ou fonction d'appartenance); il s'agit du facteur d'appartenance. A la variable x_i est associé la valeur du facteur d'appartenance μ_i représentant le degré de son appartenance au sous-ensemble flou d'indice i , μ_i dépend de la forme de la fonction d'appartenance choisie.

Dans notre application, on a utilisé les opérateurs suivants pour la construction des règles d'inférence:

- L'union : $A \cup B$. (Le "ou" représenté dans notre application par le maximum).
- L'intersection : $A \cap B$. (Le "et" est représenté par le minimum).
- L'implication : si A alors B. (alors est représenté par le minimum également).

V.4.1 Loi de commande:

Sur une période d'échantillonnage Te , la loi de commande est de la forme suivante:

$$U(KTe) = f(e(KTe), \Delta e(KTe)).$$

Le fonctionnement du contrôleur donne x_R , à laquelle on applique un gain pour obtenir la variation de la commande Δu à l'instant KTe . D'où:

$$U(KTe) = \Delta U(KTe) + U(KTe - Te) \quad (V.2)$$

$$\Delta U(KTe) = G_{\Delta u} x_R(KTe). \quad (V.3)$$

$G_{\Delta u}$ est généralement choisi faible pour des raisons de stabilité.

V.4.2 Implémentation

Le processus de commande à l'instant KTe est donné par les étapes suivantes :

1. Calcul de l'erreur présente et de sa variation :

$$e(KTe) = \text{consigne} - \text{valeur de sortie} \quad (\text{V.4})$$

$$\Delta e(KTe) = \frac{e(KTe) - e(KTe - Te)}{Te} \quad (\text{V.5})$$

2. Normalisation de l'erreur et de sa variation à des gains correspondants :

$$x_1(KTe) = G_e \cdot e(KTe) \quad (\text{V.6})$$

$$x_2(KTe) = G_{\Delta e} \cdot \Delta e(KTe) \quad (\text{V.7})$$

3. Fuzzification des variables obtenues en variables floues.

4. Obtention de la variable floue par application des règles d'inférence à partir de la table de décision.

5. Defuzzification de la valeur floue résultante en une valeur déterminée, et normalisation à l'aide du gain de sortie approprié.

6. Application de Runge-Kutta pour la détermination de la nouvelle valeur de sortie.

Toutes ces étapes sont présentées par l'organigramme suivant :

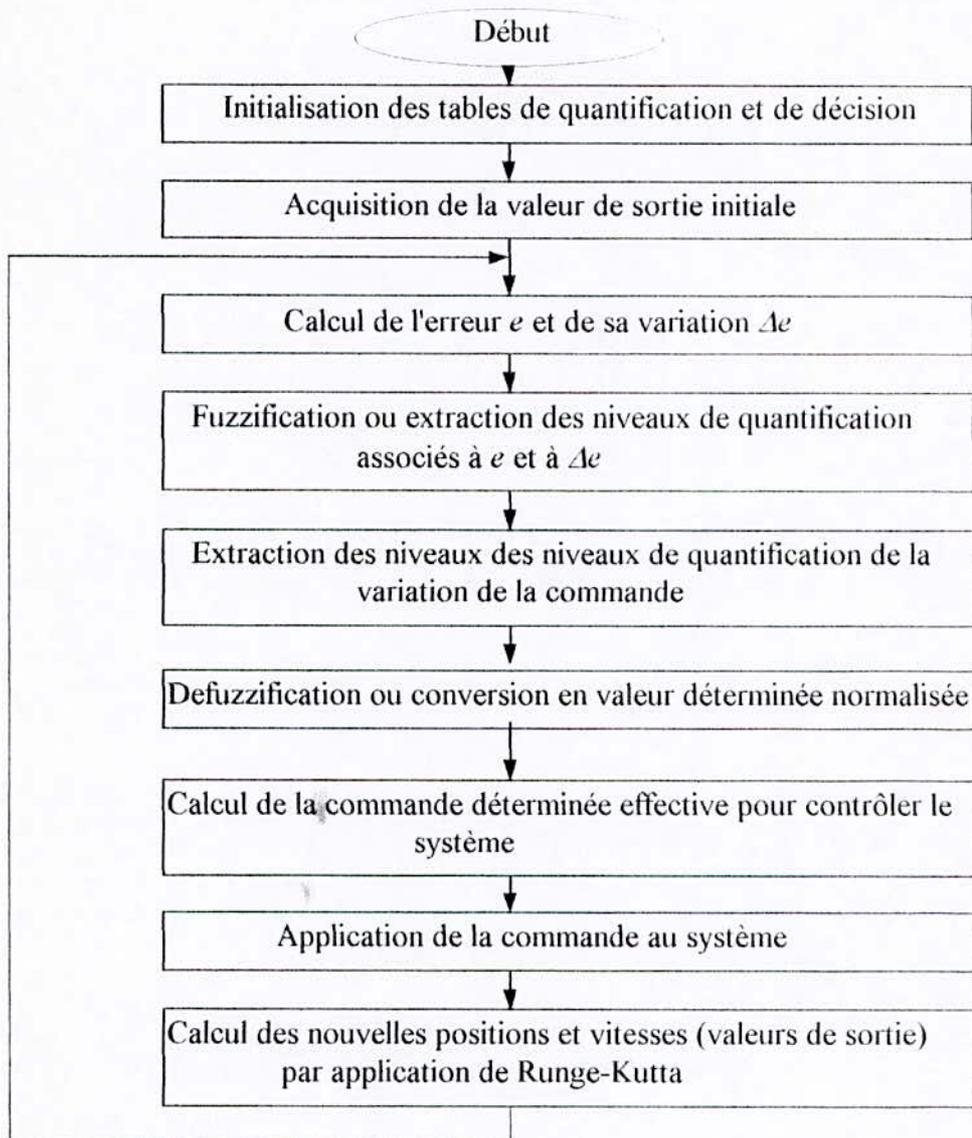


Figure V-2 : Organigramme de l'algorithme de contrôle flou

V.5 Application à la commande du bras à un degré de liberté

V.5.1 Choix des fonctions d'appartenance :

Vu la simplicité de mise en œuvre et la facilité de programmation avec une distribution triangulaire uniforme (équidistante) et symétrique (ce qui donne un chevauchement de 50% entre les sous ensembles), nous avons opté pour cette dernière pour le choix de la forme des fonctions d'appartenance.

En général, on utilise 3, 5 ou 7 ensembles flous pour les fonctions d'appartenance. Pour des raisons de performance [FAL 98], nous avons choisi de travailler avec 7 ensembles flous.

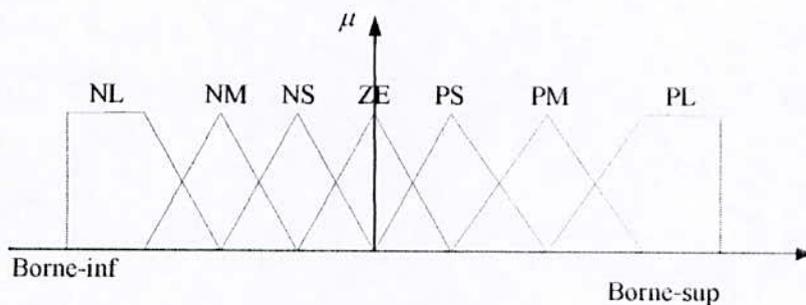


Figure V-3 : Distribution des ensembles flous

V.5.2 Univers de discours :

L'erreur est sa variation sont des angles (articulation rotative) comprises entre +3.15 et -3.15 ($\pm\pi$) radian.

Les valeurs d'entrée et de sortie sont normalisées à 1.

V.5.3 Règles décisionnelles :

Les règles décisionnelles sont de la forme :

si (x_1 est A) et (x_2 est B) alors (x_R est C)

L'ensemble des réglés doit regrouper toutes les situations possibles que peut rencontrer le contrôleur flou lors de son fonctionnement. Les différentes règles possibles sont présentées dans la matrice de Mac.Vicar-Whelan qui constitue une base de données très pratique pour le calcul des règles. Ce tableau est donné par la figure (V-4) :

$x_1 \backslash x_2$	NL	NM	NS	ZE	PS	PM	PL
NL	NL	LN	NL	NL	NM	NS	ZE
NM	NL	NL	NL	NM	NS	ZE	PS
NS	NL	NL	NM	NS	ZE	PS	PM
ZE	NL	NM	NS	ZE	PS	PM	PL
PS	NM	NS	ZE	PS	PM	PL	PL
PM	NS	ZE	PS	PM	PL	PL	PL
PL	ZE	PS	PM	PL	PL	PL	PL

Figure V-4 : Matrice de Mac-Vicar

V.5.4 Méthode d'inférence :

Pour l'implémentation des règles décisionnelles, nous avons choisi les opérateurs arithmétiques suivants :

et : représenté par le minimum.

ou : représenté par le maximum.

alors : représenté par le minimum.

V.5.5 Période d'échantillonnage :

les résultats de simulation montrent qu'il est, généralement, suffisant de simuler sur dix seconde. Le système (bras) approche la consigne assez parfaitement durant cette période.

On remarque que la durée de simulation est également acceptable, pour un choix d'un pas d'échantillonnage égal à 0.01 sec (10 ms). Cela est dû à l'inertie du système (vitesse et accélération articulaires faibles).

V.5.6 Entrées/sorties du régulateur :

Vu l'aspect pratique du bras à un degré de liberté, l'erreur et sa variation ont été normalisées à $\pm\pi$ ce qui permet de choisir les gains d'entrée égaux à $1/\pi$ sans avoir de dépassement de la gamme du contrôleur flou (cela est valable dans le cas de la stabilité du système).

Tandis qu'en faisant varier le gain de sortie, on voit que le système peut perdre sa stabilité: l'accroissement de la commande Δu fait que le système oscille autour de la consigne, et diverge parfois.

Nous avons fait des simulations du bras à un degré de liberté dont les paramètres sont donnés en annexe.

V.6 Simulations et interprétations des résultats :

V.6.1 Influence du gain de sortie :

Nous avons fait plusieurs simulations du bras à un degré de liberté (liaison rotative) pour différentes valeurs du gain de sortie du régulateur flou. Les valeurs sont comprises entre 0.01 et 0.1

Cela permet de voir l'effet de ce paramètre sur la réponse du bras à une entrée échelon (consigne constante).

Les figures 5 à 10 montrent les résultats de simulations pour différentes valeurs du gain de sortie.

Les figures 11 à 16 montrent les résultats de simulation pour une commande appliquée à un nombre de pas d'intégration différent pour les mêmes valeurs du gain de sortie.

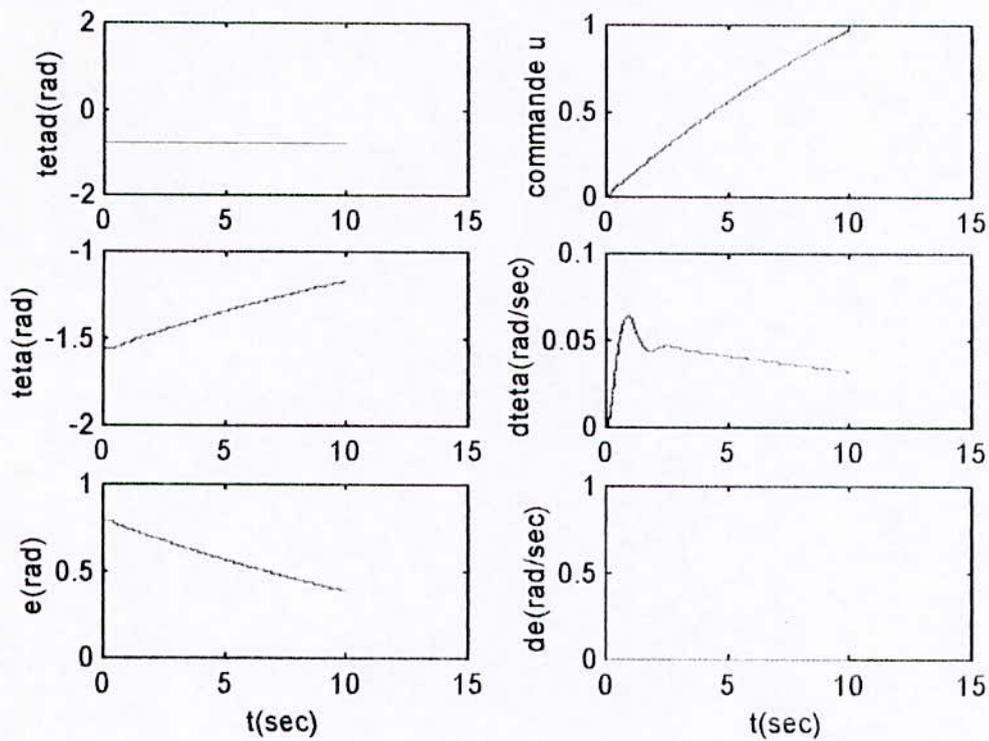


Figure V-5 : résultats de simulation du bras à un degré de liberté. $G_s=0.01$

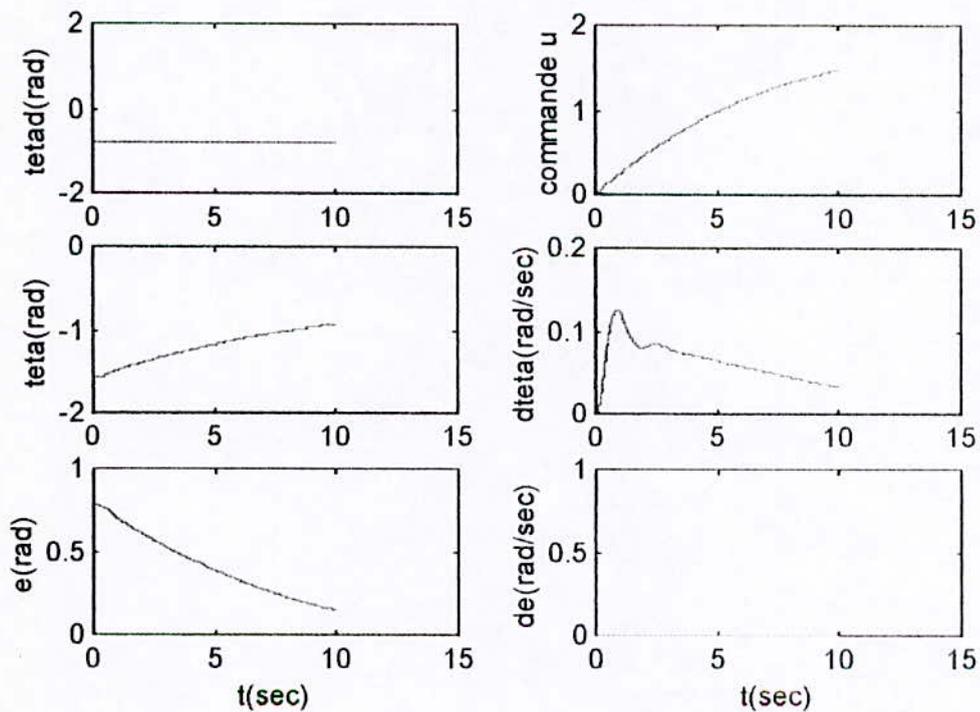


Figure V-6 : résultats de simulation du bras à un degré de liberté. $G_s=0.02$

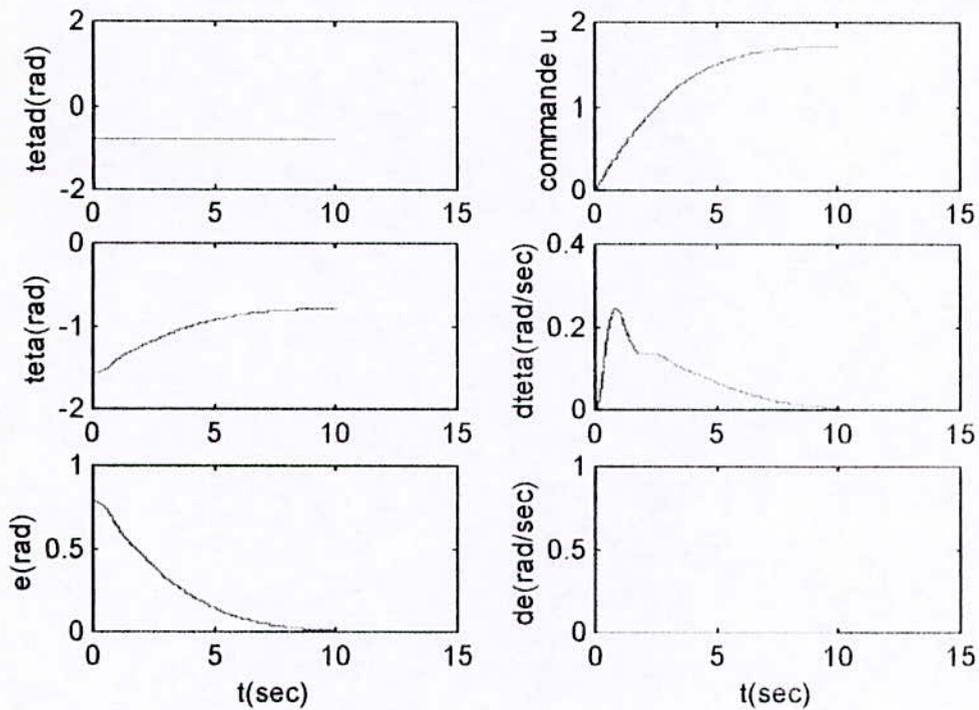


Figure V-7 :résultats de simulation du bras à un degré de liberté. $G_s=0.04$

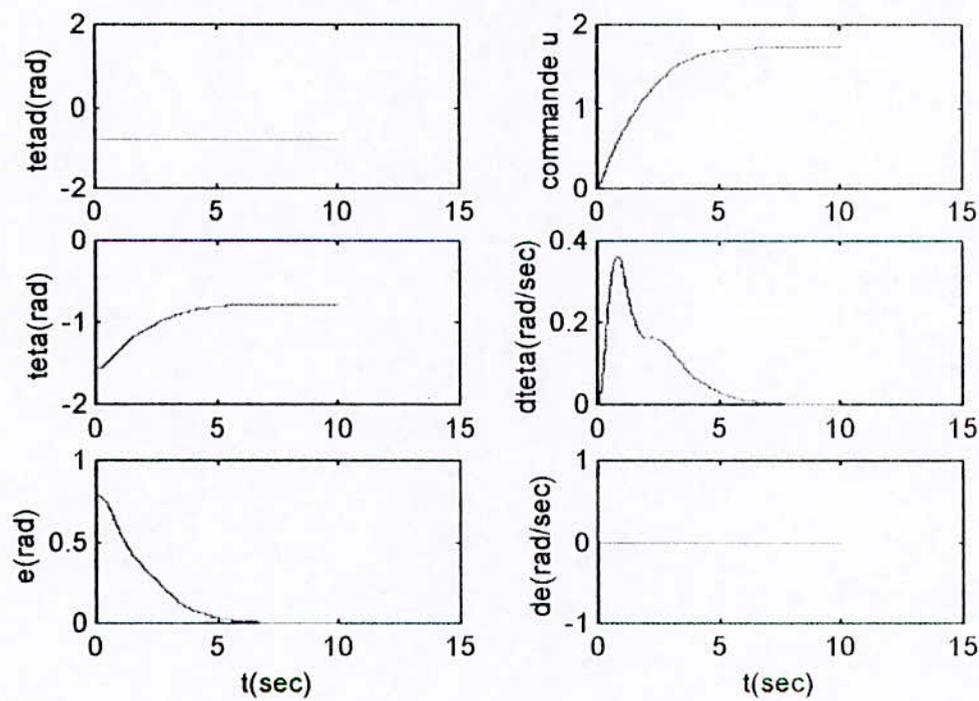


Figure V-8 :résultats de simulation du bras à un degré de liberté. $G_s=0.06$

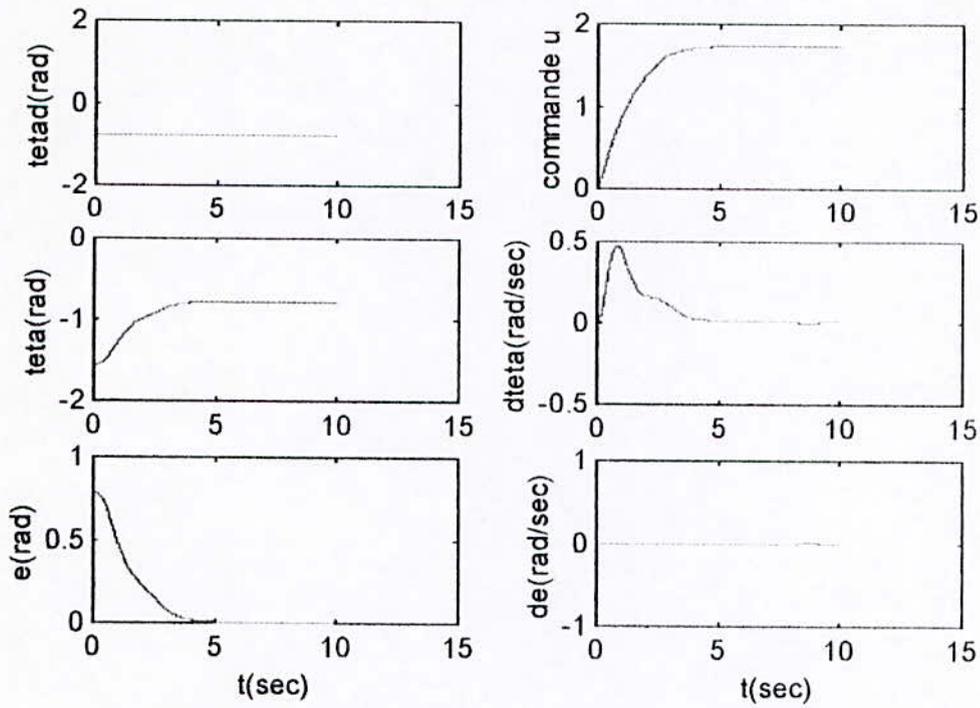


Figure V-9 :résultats de simulation du bras à un degré de liberté. $G_s=0.08$

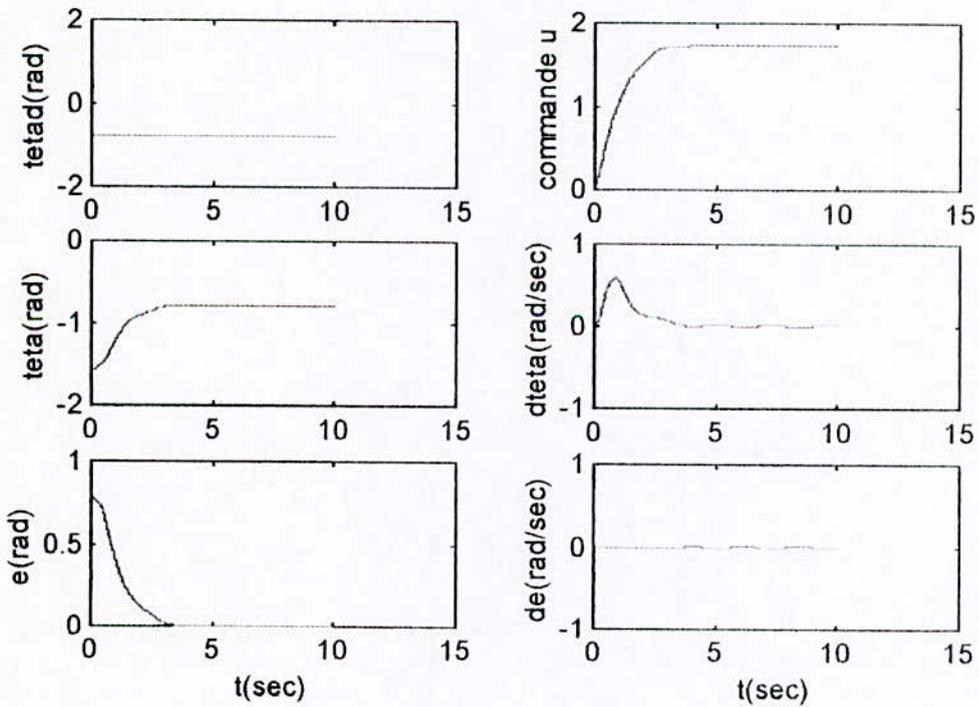


Figure V-10 :résultats de simulation du bras à un degré de liberté. $G_s=0.1$

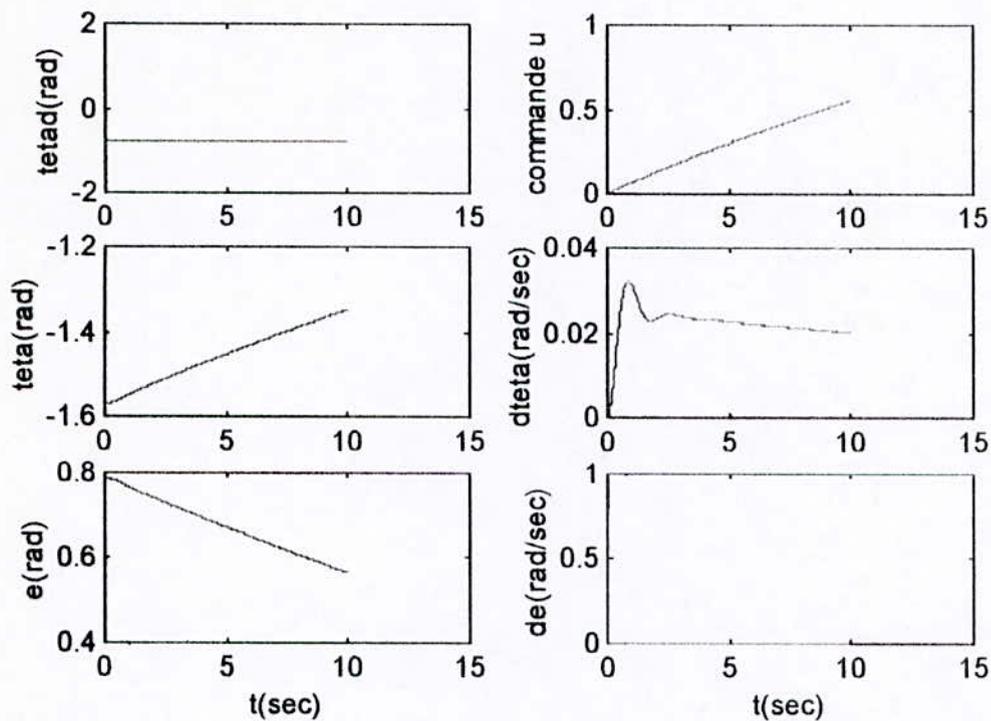


Figure V-11 :résultats de simulation du bras à un degré de liberté. $G_s=0.01$

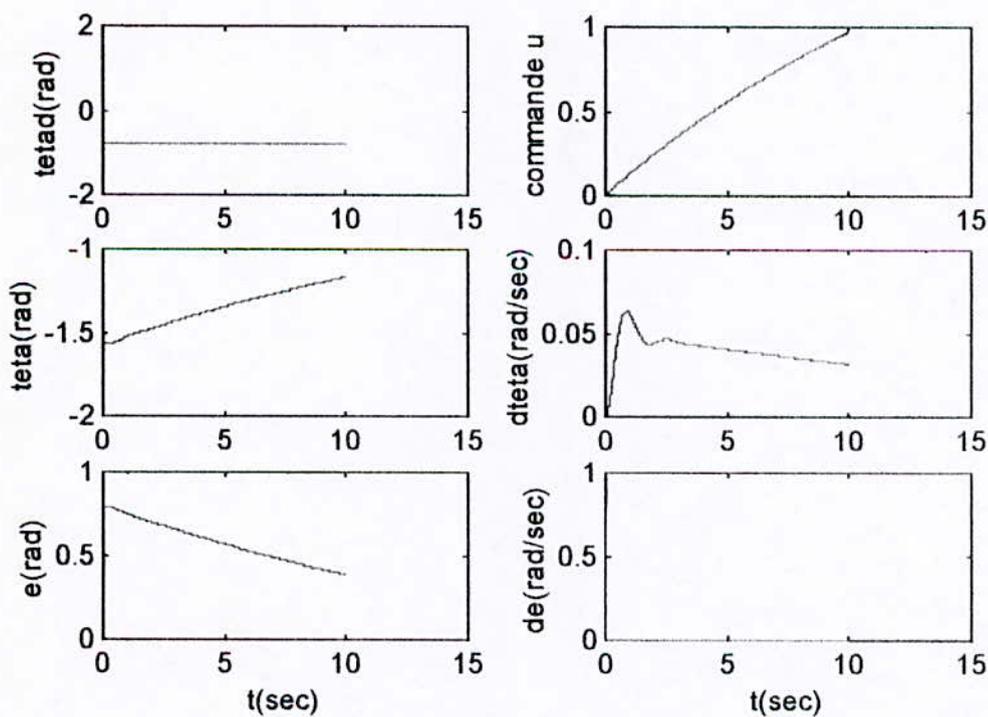


Figure V-12 :résultats de simulation du bras à un degré de liberté. $G_s=0.02$

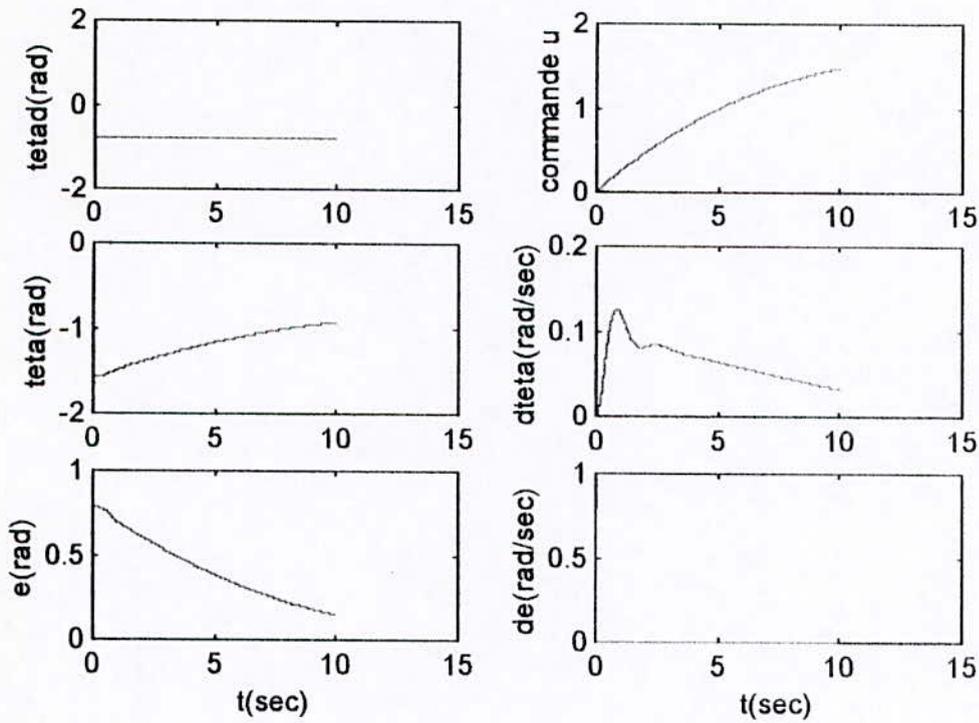


Figure V-13 :résultats de simulation du bras à un degré de liberté. $G_s=0.04$

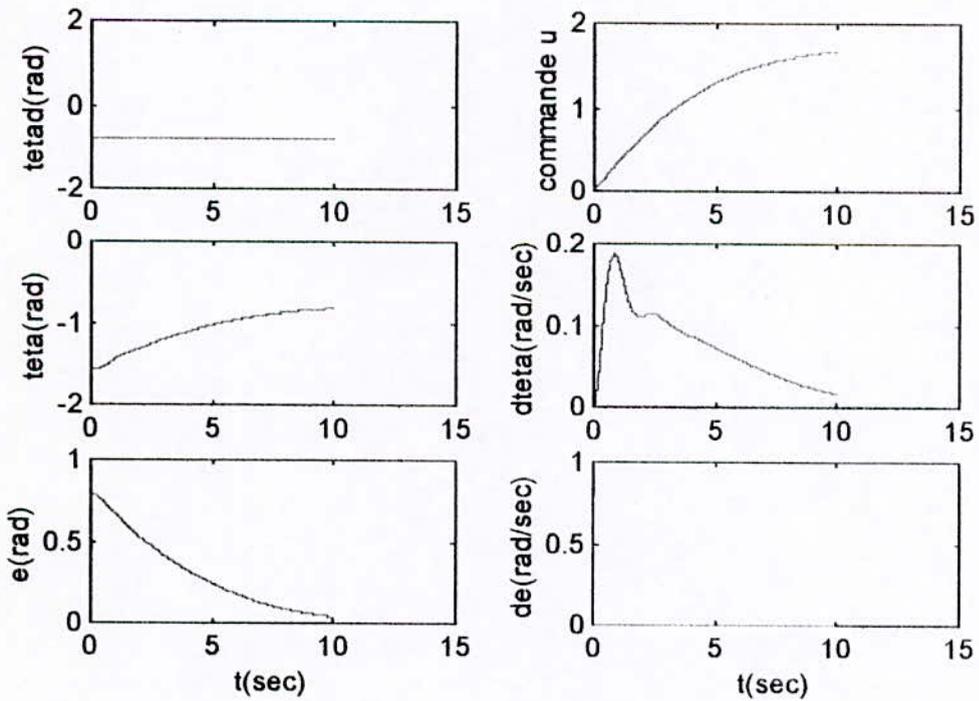


Figure V-14 :résultats de simulation du bras à un degré de liberté. $G_s=0.06$

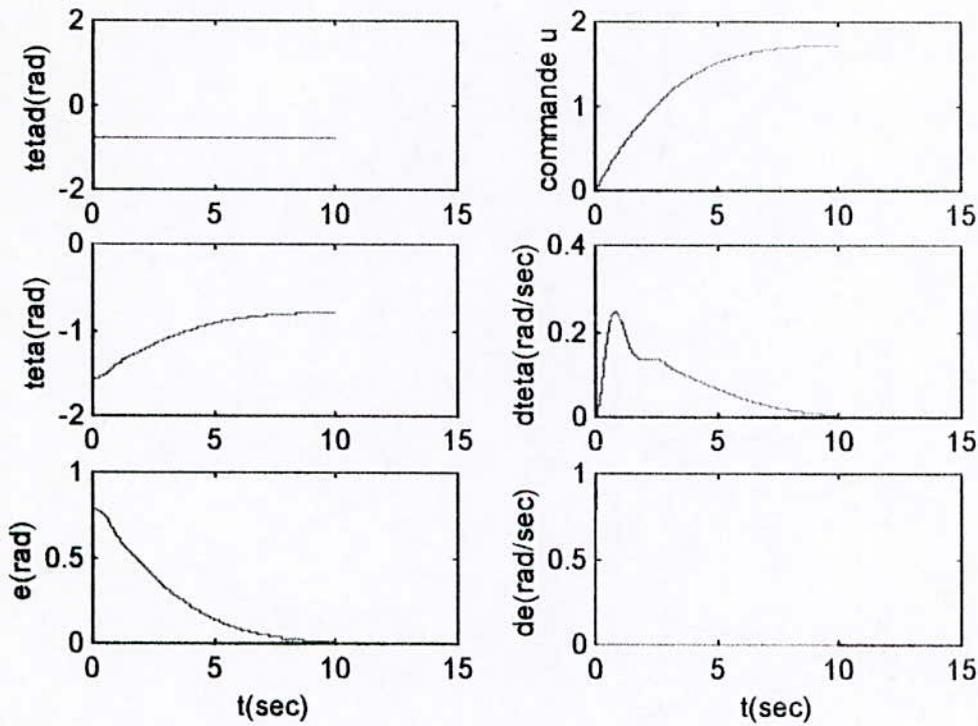


Figure V-15 :résultats de simulation du bras à un degré de liberté. $G_s=0.08$

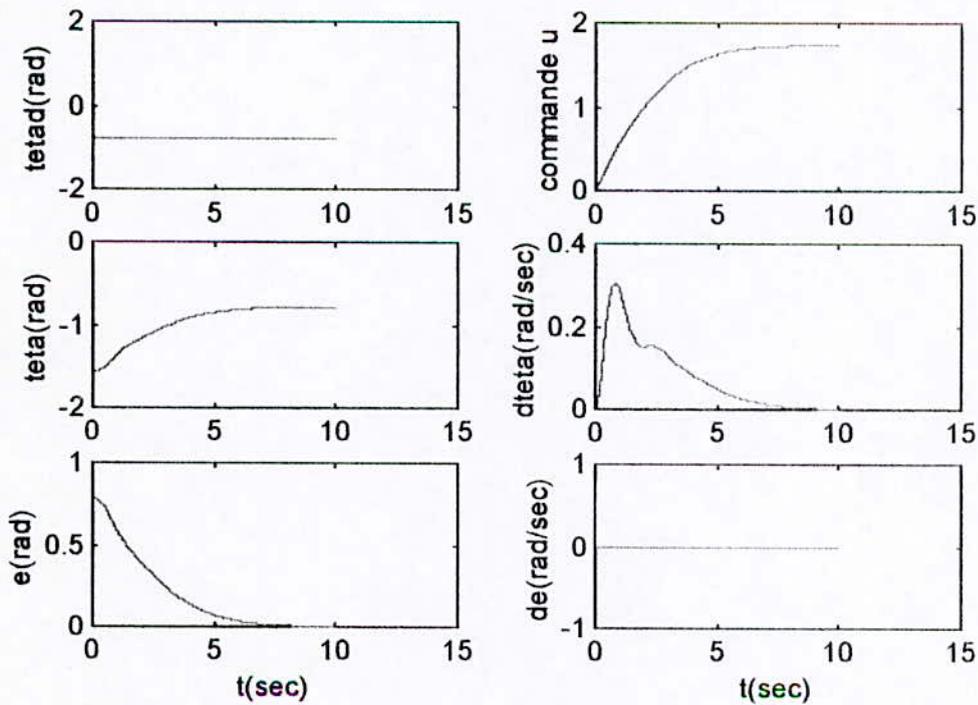


Figure V-16 :résultats de simulation du bras à un degré de liberté. $G_s=0.1$

V.6.2 Observations et interprétation des résultats :

La durée de simulation, comme le montre les figures, ne dépasse pas les 10 sec.

Les résultats de simulation montrent que le bras répond à la consigne de plus en plus rapidement au fur et à mesure que le gain augmente; à partir d'une certaine valeur de ce gain, le bras oscille autour de la consigne avant d'atteindre l'état permanent (il s'approche de l'instabilité). Si on continue à faire augmenter le gain de sortie le système risque de ne jamais répondre à la consigne.

Dans le dernier cas, la dynamique du régulateur flou (accroissement de la commande) est trop forte pour le système.

Des simulations pour un bras ayant des paramètres réduits ont donné une meilleure réponse pour des gains plus élevés. Cela est dû à l'inertie du système: plus cette inertie est importante, moins grand doit être le gain du régulateur et plus la réponse du bras sera lente.

Nous concluons que pour des valeurs réduites du gain de sortie que l'effet du régulateur est faible devant l'inertie du bras de robot, cela cause la lenteur de la réponse de celui-ci. En faisant augmenter le gain la réponse devient de plus en plus rapide jusqu'à l'obtention de dépassement : le régulateur applique des accroissements de commande de plus en plus importants.

V.6.3 Influence de la période d'application de la commande par logique floue :

Nous avons considéré deux cas dans les simulations faites pour le bras à un degré de liberté:

- A chaque pas d'intégration nous calculons la nouvelle commande (figures 5 à 10)
- A chaque deux pas d'intégration nous calculons la nouvelle commande (figures 11 à 16).

V.6.4 Observations et interprétations des résultats:

Les résultats des simulations montrent l'existence d'un compromis entre le nombre de pas durant lequel on applique la même commande et la valeur du gain de sortie du régulateur : si nous calculons la nouvelle commande pour un petit nombre de pas d'intégration l'accroissement de la commande doit être réduit, cela est obtenu en réduisant la valeur du gain de sortie du régulateur. Pour réduire la durée des simulations nous avons préféré l'utilisation des valeurs élevées du gain avec un calcul moins fréquent de la nouvelle commande.

V.6.5 Tests de robustesse de la commande par logique floue :

Pour voir à quel point notre commande est robuste vis à vis des perturbations éventuelles lors de l'utilisation du bras de robot, nous avons supposé que le bras porte une charge qu'il lâche brusquement. Nous avons également simulé pour plusieurs valeurs du gain de sortie pour comparer les résultats.(figures V-17 à V-20).

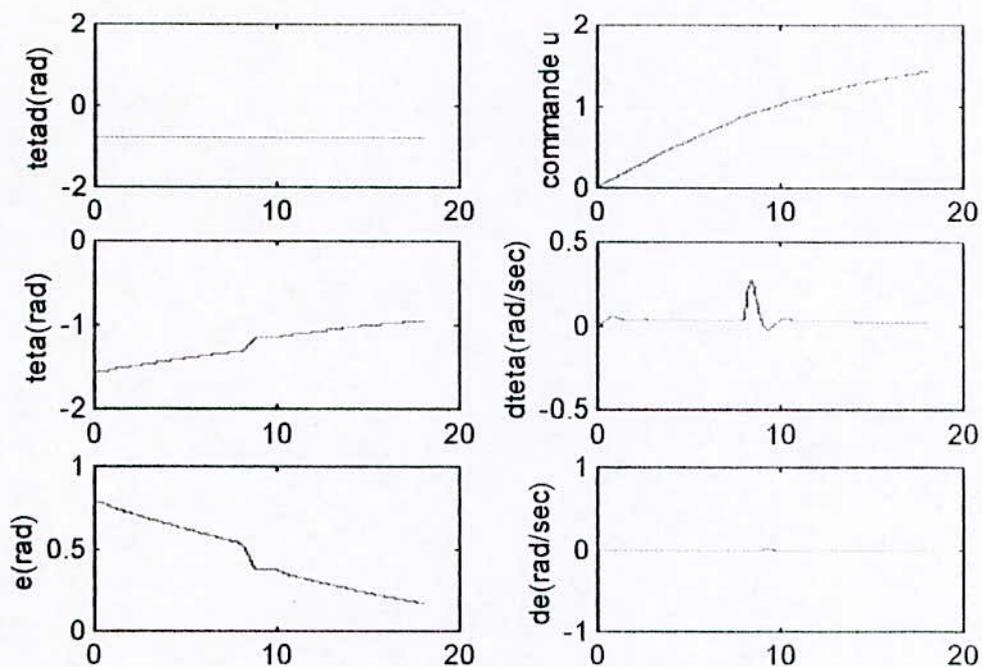


Figure V-17 :résultats de simulation du bras à un degré de liberté. $G_s=0.02$

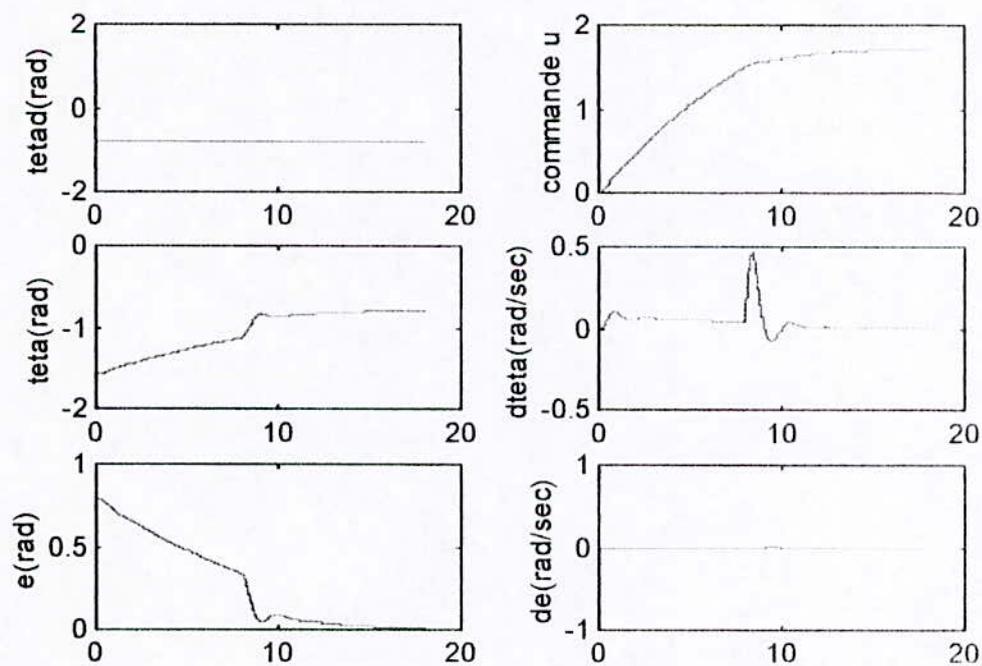


Figure V-18 :résultats de simulation du bras à un degré de liberté. $G_s=0.04$

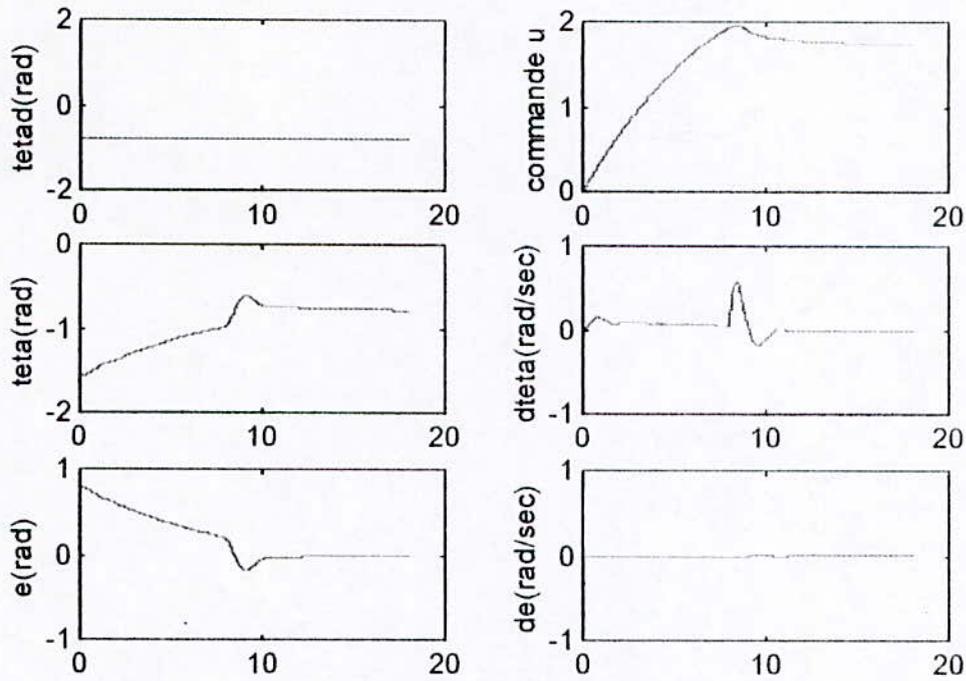


Figure V-19 :résultats de simulation du bras à un degré de liberté. $G_s=0.06$

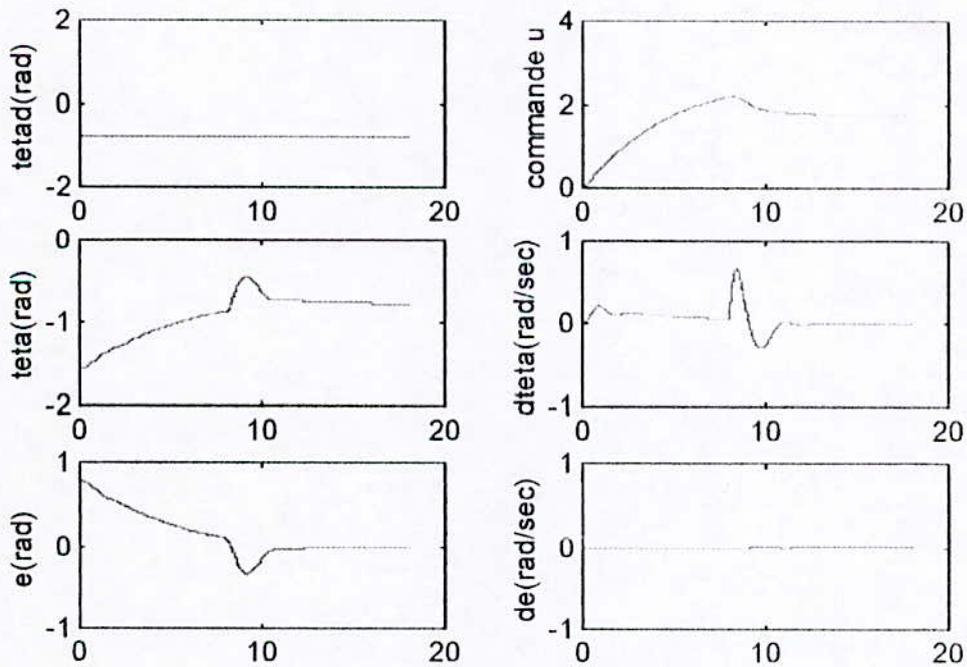


Figure V-20 :résultats de simulation du bras à un degré de liberté. $g_s=0.08$

V.6.6 Simulation dans le cas d'un suivi d'une trajectoire :

Nous avons simulé à un degré de liberté pour une consigne d'allure sinusoïdale.

Les figures V-21 à V-28 montrent les réponses du bras pour différentes valeurs du gain de sortie du régulateur flou.

Nous avons utilisé deux variantes du régulateur flou :

la première variante :

Les deux variables linguistiques sont :

- L'erreur sur la variable articulaire
- L'accroissement de cette erreur.

Les figures V-21 à V-24 montrent les réponses du bras commandé par le régulateur déjà cité.

La deuxième variante :

Les deux variables linguistiques sont :

- L'erreur sur la variable articulaire.
- L'erreur sur la vitesse articulaire.

Les figures V-25 à V-28 montrent la réponse du même bras commandé par le régulateur légèrement modifié.

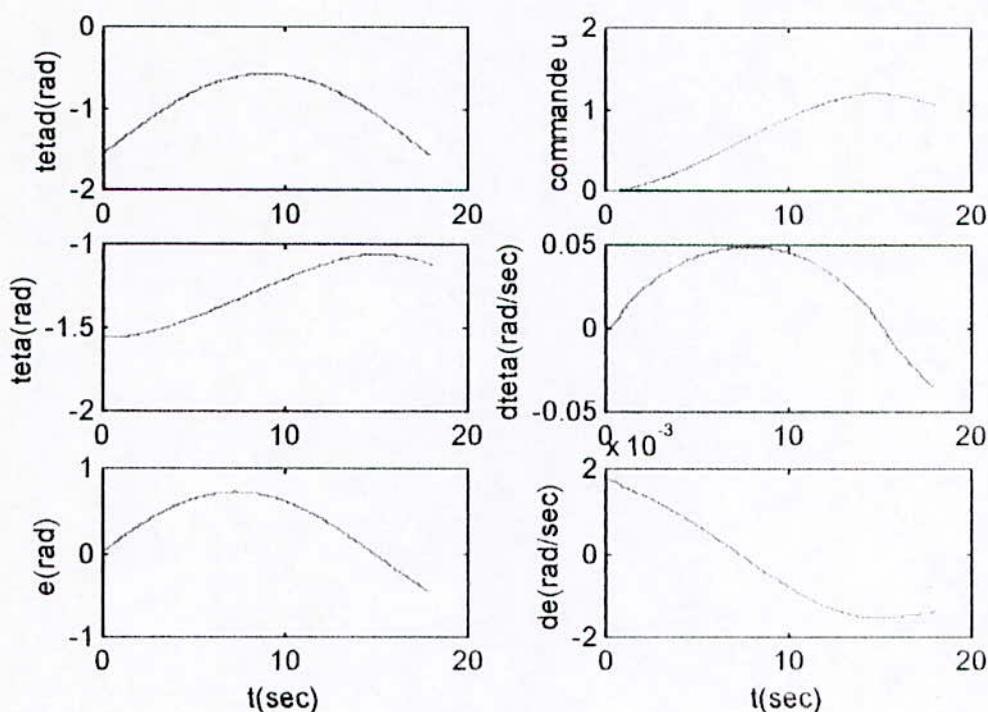


Figure V-21 :résultats de simulation du bras à un degré de liberté. (allure sinusoïdale de la variable articulaire). $G_s=0.02$

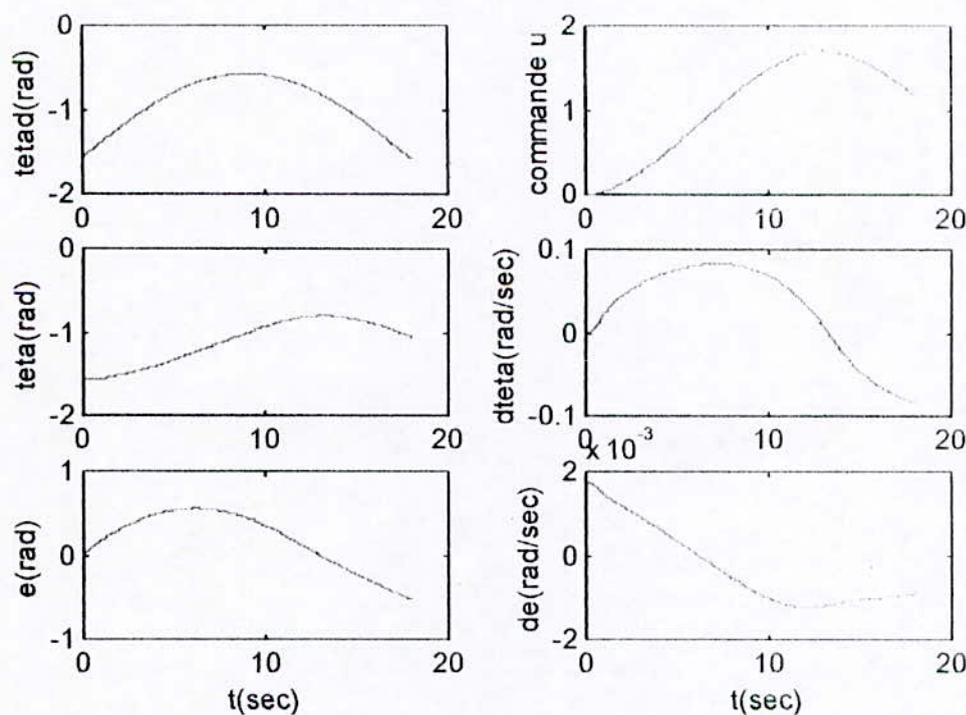


Figure V-22 :résultats de simulation du bras à un degré de liberté. (allure sinusoïdale de la variable articulaire). $G_s=0.04$

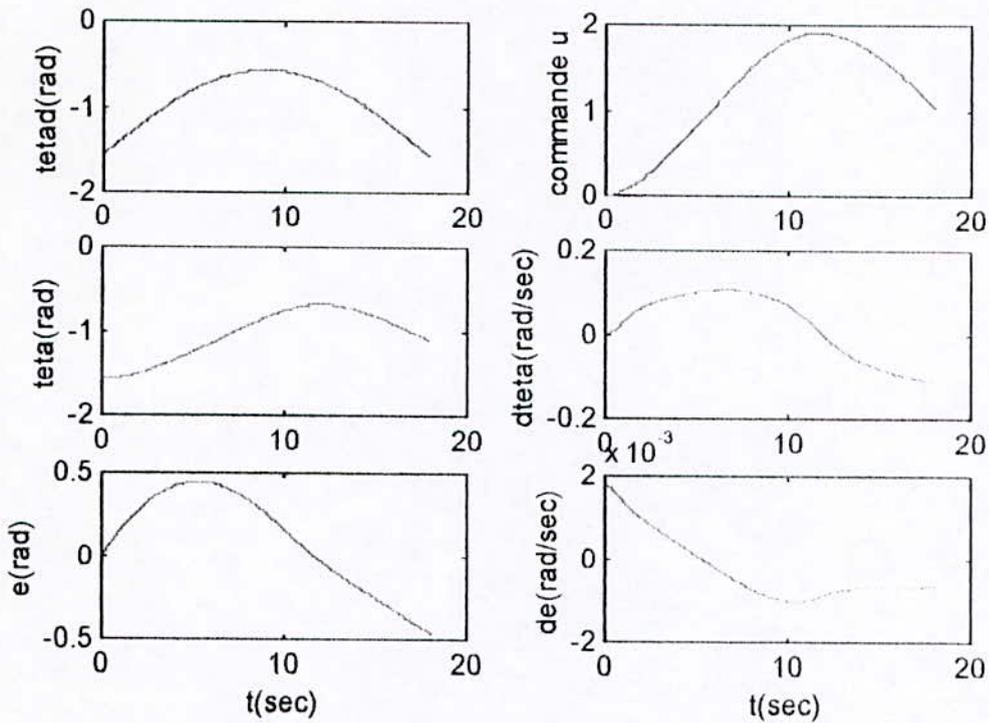


Figure V-23 :résultats de simulation du bras à un degré de liberté.(allure sinusoïdale de la variable articulaire). $G_s=0.06$.

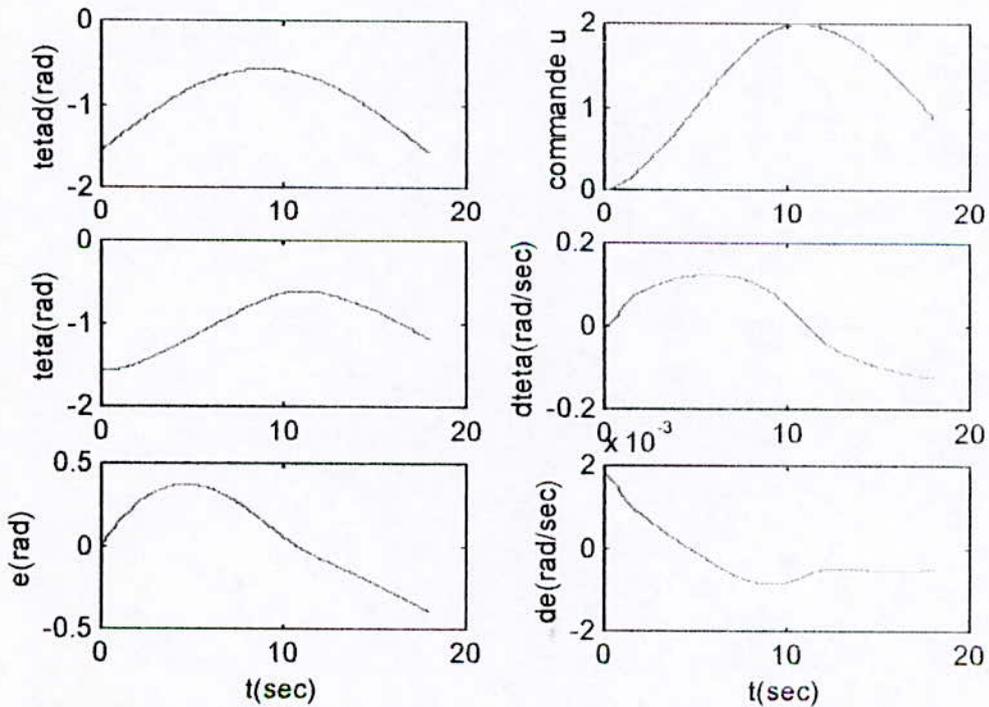


Figure V-24 :résultats de simulation du bras à un degré de liberté.(allure sinusoïdale de la variable articulaire). $G_s=0.08$.

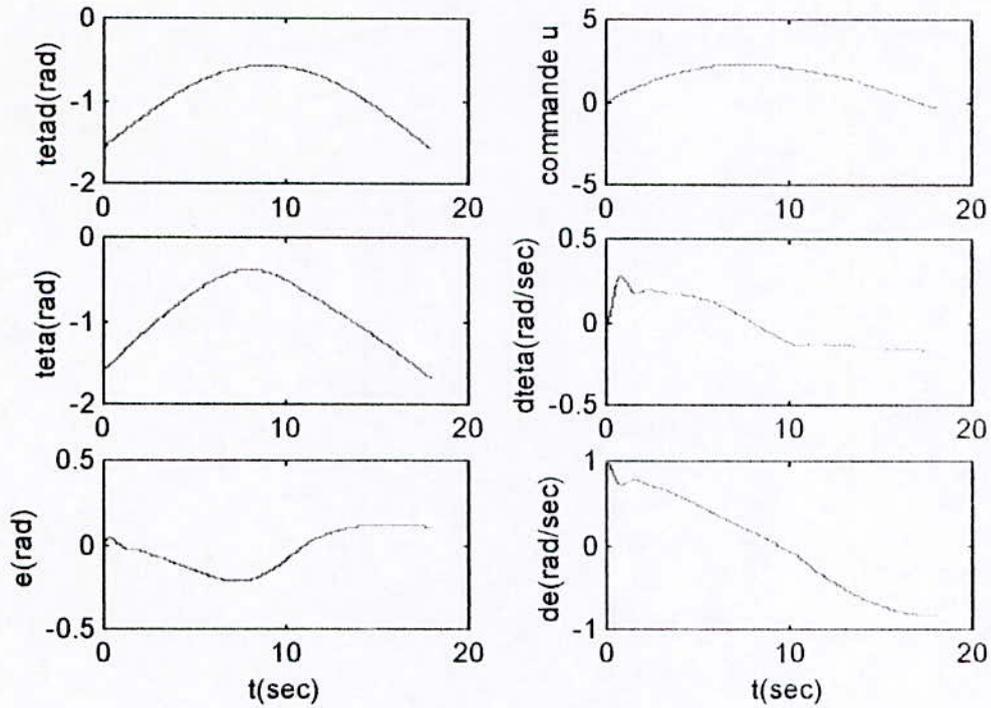


Figure V-25 :résultats de simulation du bras à un degré de liberté. $G_s=0.02$

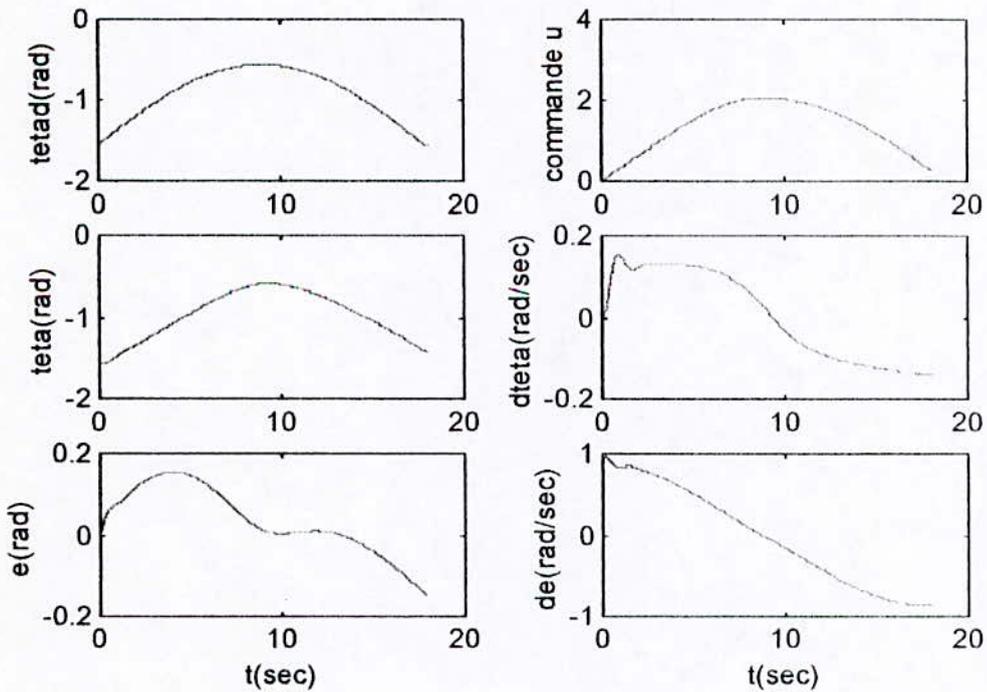


Figure V-26 :résultats de simulation du bras à un degré de liberté. $G_s=0.04$

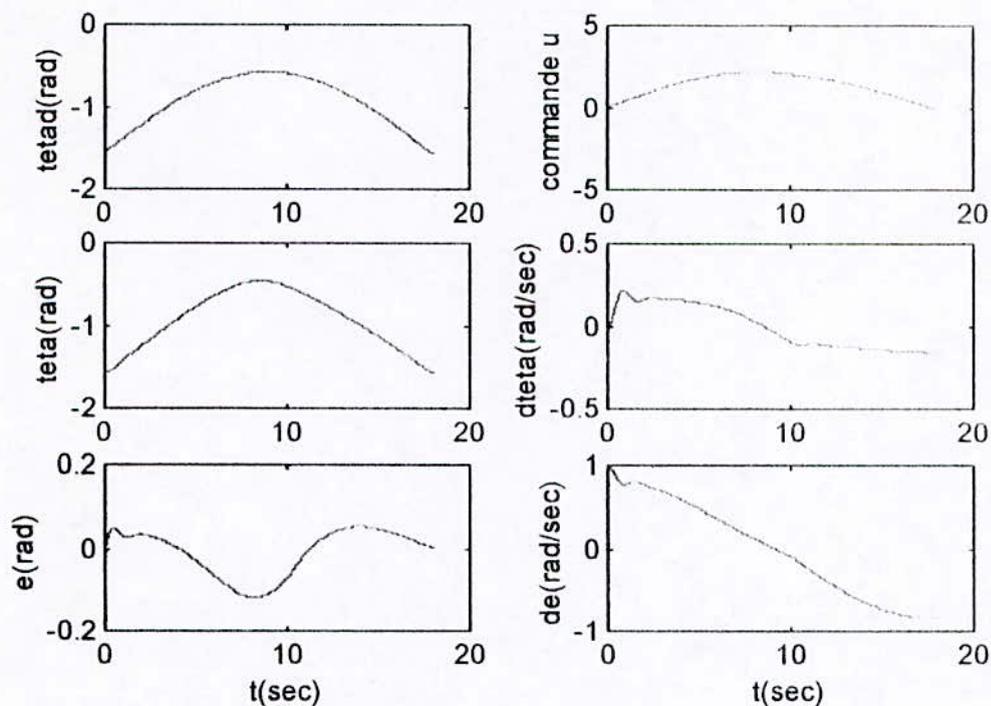


Figure V-27 :résultats de simulation du bras à un degré de liberté. $G_s=0.06$

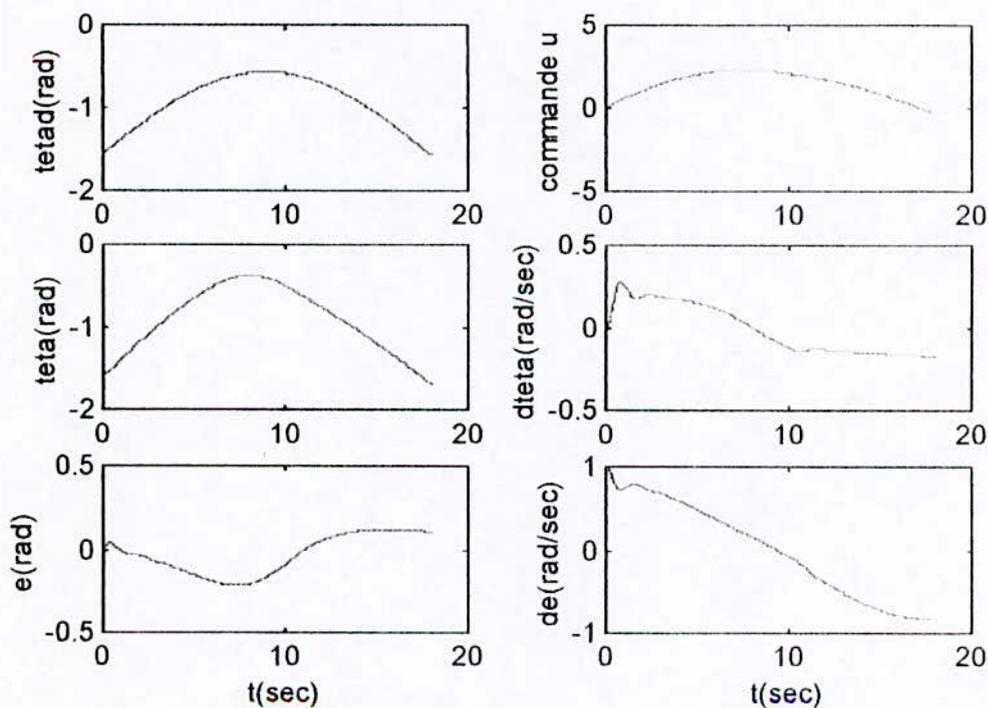


Figure V-28 :résultats de simulation du bras à un degré de liberté. $G_s=0.08$

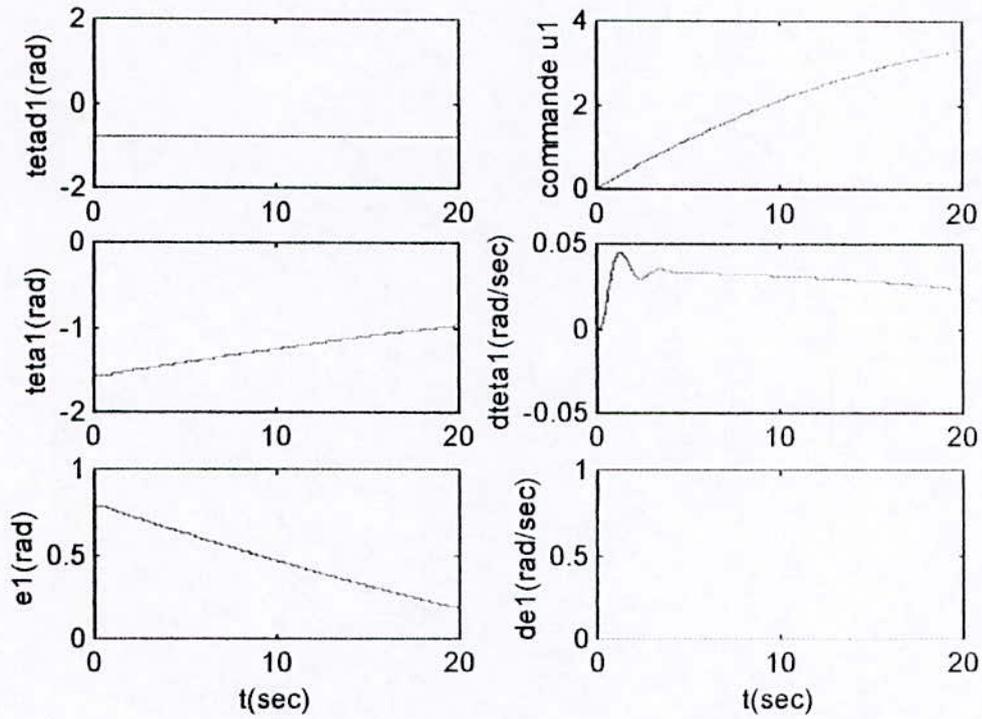
Conclusion :

Pour une consigne variable, il est préférable d'utiliser l'erreur sur la vitesse à la place de l'accroissement de l'erreur sur la position : cela permet au régulateur d'avoir un comportement dynamique meilleur (consigne variable au cours du temps).

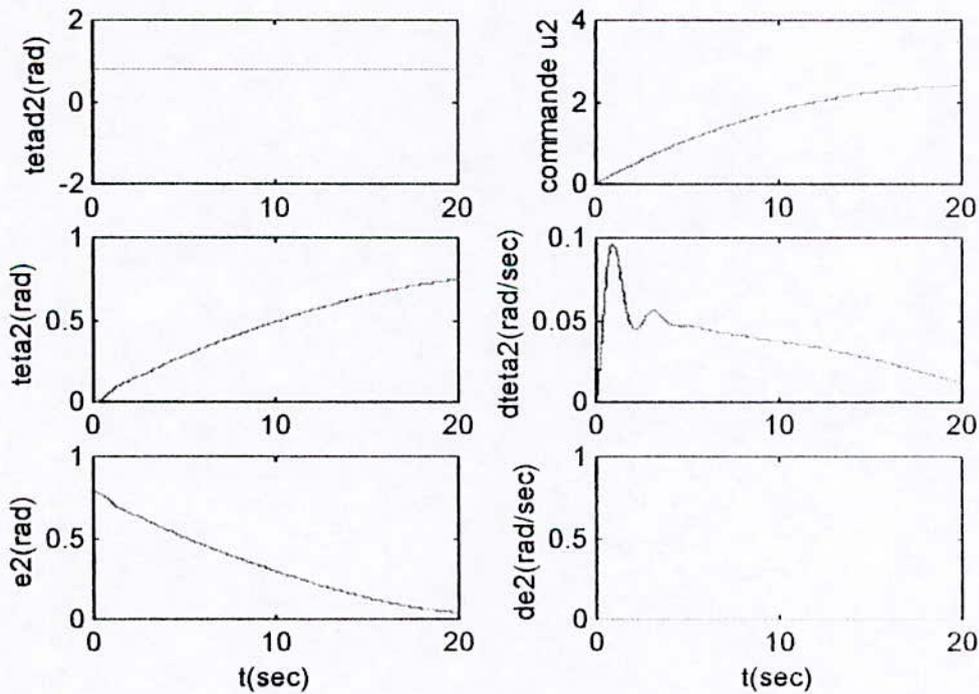
V.6.7 Simulation du bras PUMA à 2 ddl :

Nous avons également simulé le bras PUMA à deux degrés de liberté, dont le modèle dynamique est et les paramètres sont donnés en annexe A2.

La réponse à un échelon (consigne constante) dépend du gain de sortie du régulateur comme le montre les figures V-29 à V-32.

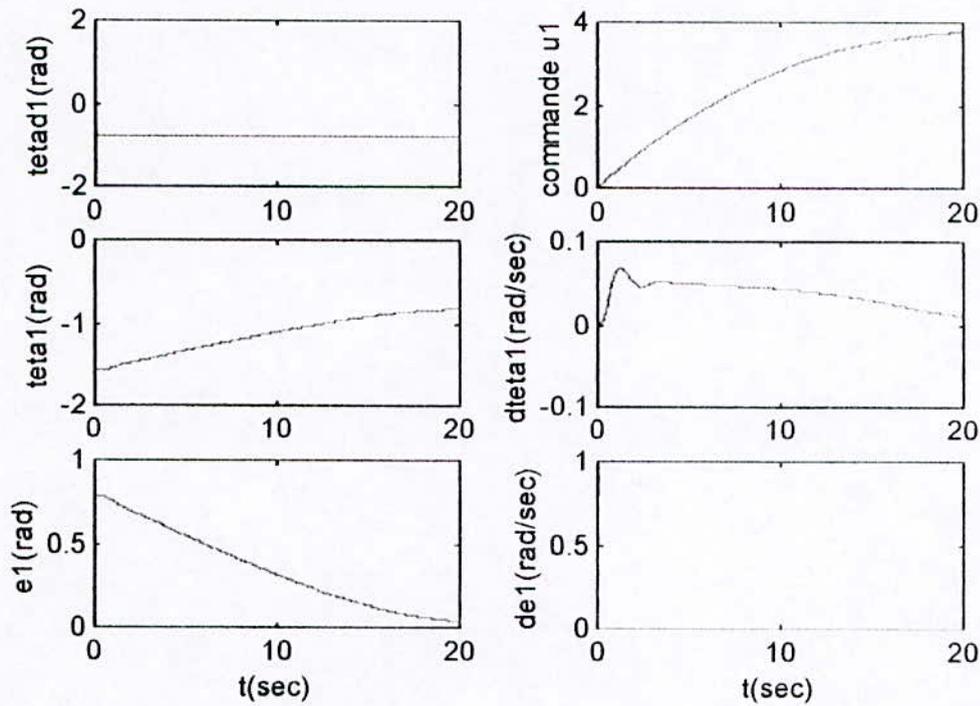


Premier segment du bras PUMA 2ddl.

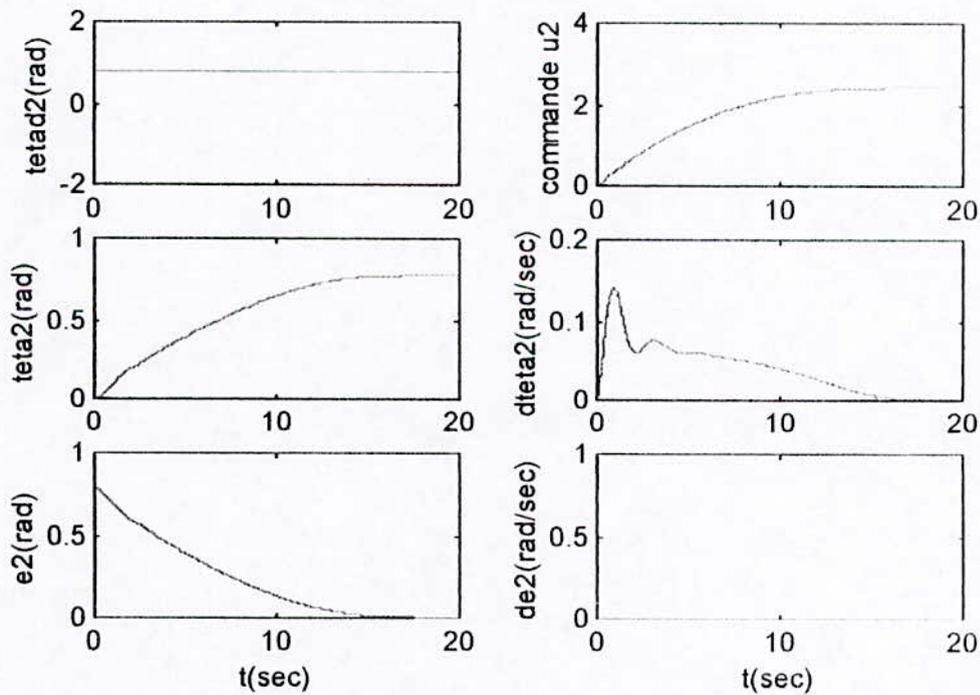


Deuxième segment du bras PUMA 2 ddl.

Figure V-29 : résultats de simulation du bras PUMA 2ddl. (consigne échelon, $Gs1=Gs2=0.04$).

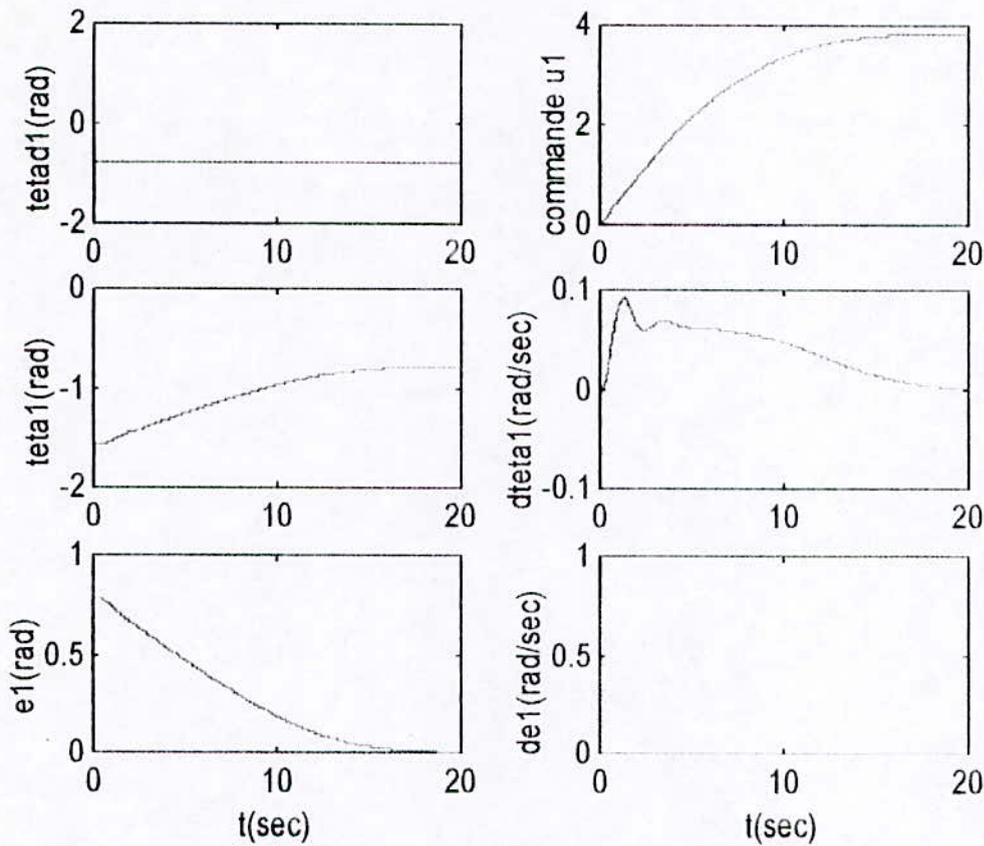


Premier segment du bras PUMA.

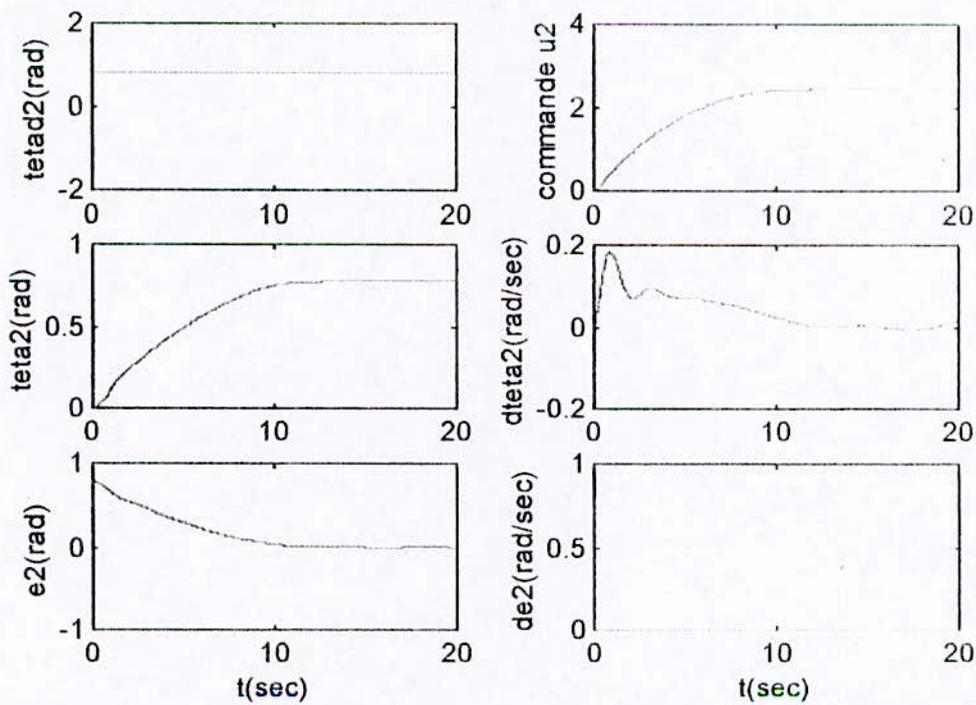


Deuxième segment du bras PUMA.

Figure V-30 : résultats de simulation du bras PUMA à 2 ddl, (consigne échelon, $Gs1=Gs2=0.06$).

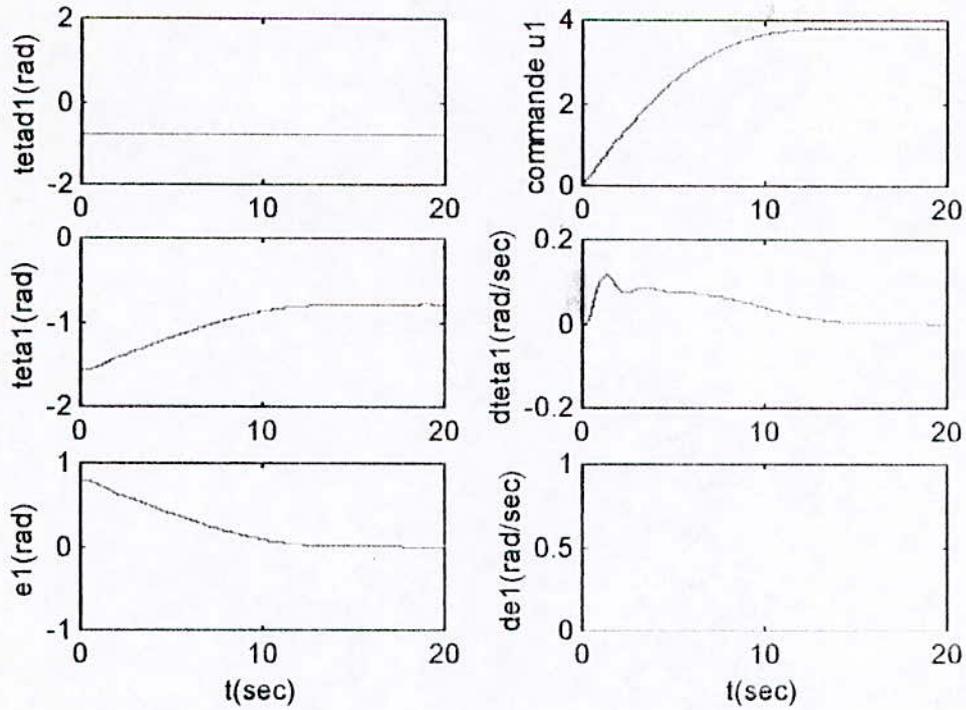


Premier segment du bras PUMA 2ddl

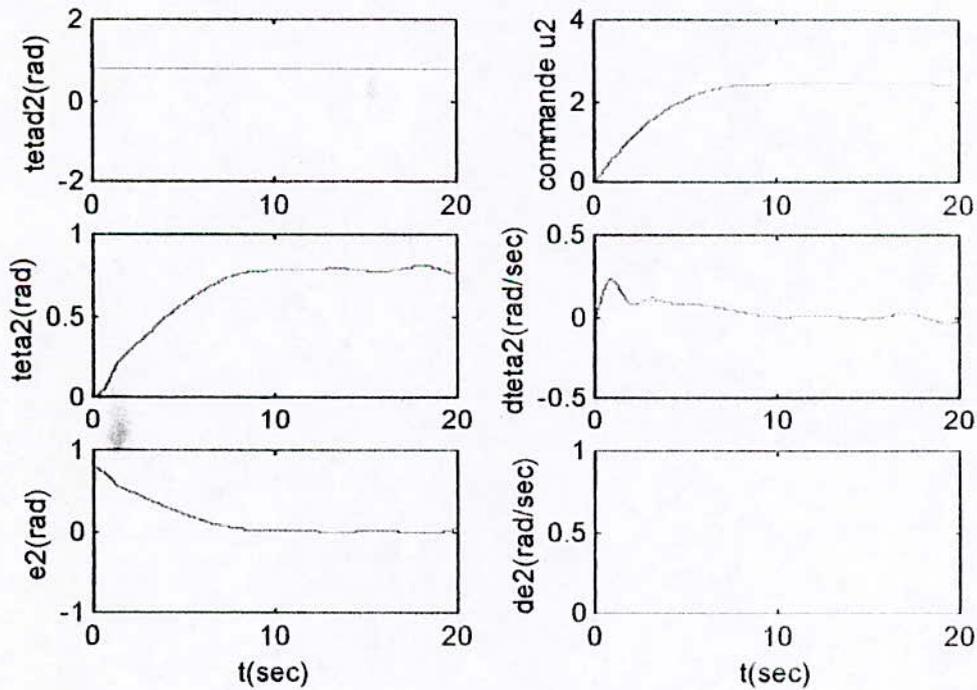


Deuxième segment du bras PUMA 2ddl

Figure V-31 : résultats de simulation du bras PUMA 2ddl, (consigne échelon, $Gs1=Gs2=0.08$).



Premier segment du bras PUMA 2ddl.



Deuxième segment du bras PUMA 2ddl.

Figure V-32 : résultats de simulation du bras PUMA 2ddl. (consigne échelon, $Gs1=Gs2=0.1$).

V.7 Application des Algorithmes Génétiques

V.7.1 Introduction :

Nous avons vu dans le chapitre précédent que la réponse du système commandé par le contrôleur flou présentait une grande dépendance de la valeur du gain de sortie de ce dernier ; la réponse du système dépend de l'accroissement Δu de la commande u appliquée.

L'introduction des Algorithmes génétiques permet d'assurer une recherche plus ou moins « organisée » des valeurs des gains donnant des réponses optimales du système. L'intervalle de recherche doit être spécifié en estimant l'effort maximal pour suivre une consigne donnée.

V.7.2 Estimation de l'intervalle de recherche

La modélisation dynamique d'un bras manipulateur montre la similarité avec un pendule. Pour une consigne donnée θ_c , l'effort statique (effort d'équilibre) aura la valeur :

$$\Gamma = \frac{1}{2} mgl \cos(\theta)$$

ce qui permet de limiter l'effort à $\Gamma_0 = 1/2 mgl$ ($\theta = 0$) en valeur absolue. Ainsi, u est limitée à Γ_0/k , tel que k : est le gain de l'actionneur cela permet un choix plus ou moins rigoureux du gain de sortie $G_{\Delta u}$ du régulateur flou.

V.7.3 Représentation des solutions potentielles et population initiale

- Comme pour l'exemple d'application, nous avons choisi la représentation binaire (10100...) des valeurs des gains de sortie pour des raisons de simplicité d'implémentation déjà citées (chapitre Algorithmes génétiques)

- La population initiale est choisie d'une façon aléatoire dans l'intervalle de recherche avec taille-pop = 5 (5 individus par génération).

V.7.4 Fonction d'adéquation :

Dans cette application, la fonction d'adéquation n'a pas une forme mathématique; elle est représentée par la qualité de la réponse du système vis à vis d'une réponse désirée : il s'agit de minimiser la surface d'intégration de l'erreur (surface comprise entre la réponse désirée et la réponse du système).

V.7.5 Taux des opérations génétiques

Nous avons également gardé les mêmes valeurs de mutations et de croisement que celles utilisées dans l'exemple de minimisation de l'impédance.

$$p_c = 0.25$$

$$p_m = 0.01$$

V.7.6 Résultats de la recherche

Le bras à un degré de liberté, Gs optimal est trouvé égal à 0.085 (Figure V-33).

Le bras à un degré de liberté avec une charge Gs optimal est trouvé égal à 0.094 (Figure V-34).

Le cas d'un suivi d'une trajectoire, Gs optimal est trouvé égal à 0.061 (Figure V-35).

Conclusion

L'application des AsG en la recherche du gain optimal exige des populations de simulation du bras en question; elle assure l'obtention d'un optimum, mais peut durer plusieurs heures.

La durée de recherche dépend de la population initiale contenant plusieurs valeurs du gain de sortie du régulateur flou, elle dépend aussi des opérateurs génétiques utilisés dans la recherche.

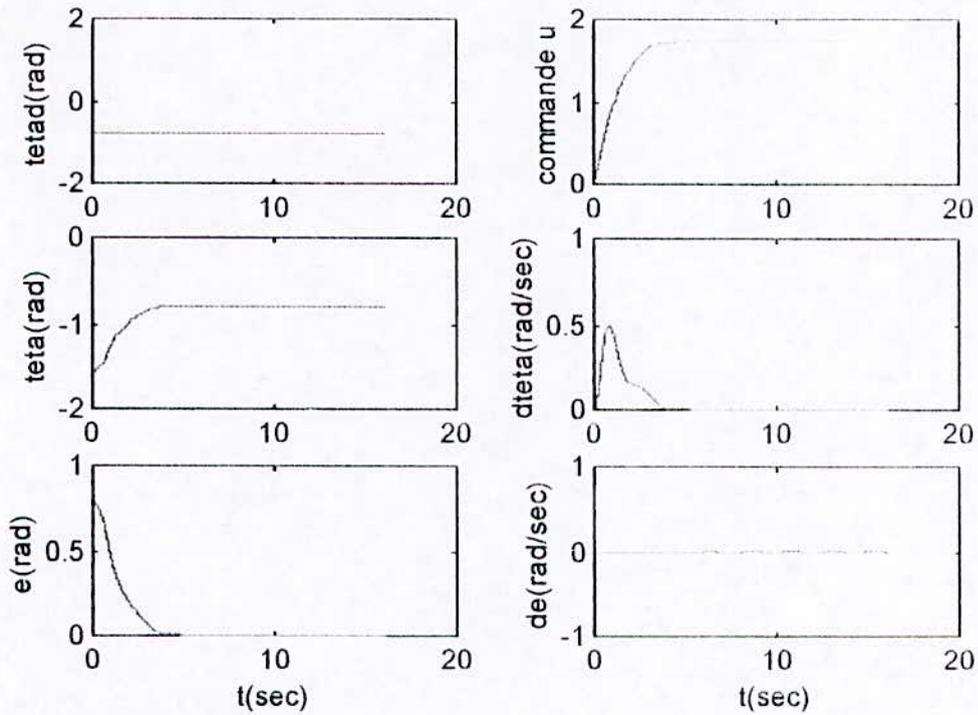


Figure V-33 : résultats de simulation du bras à un degré de liberté par application des AsG. $G_s = 0.085$

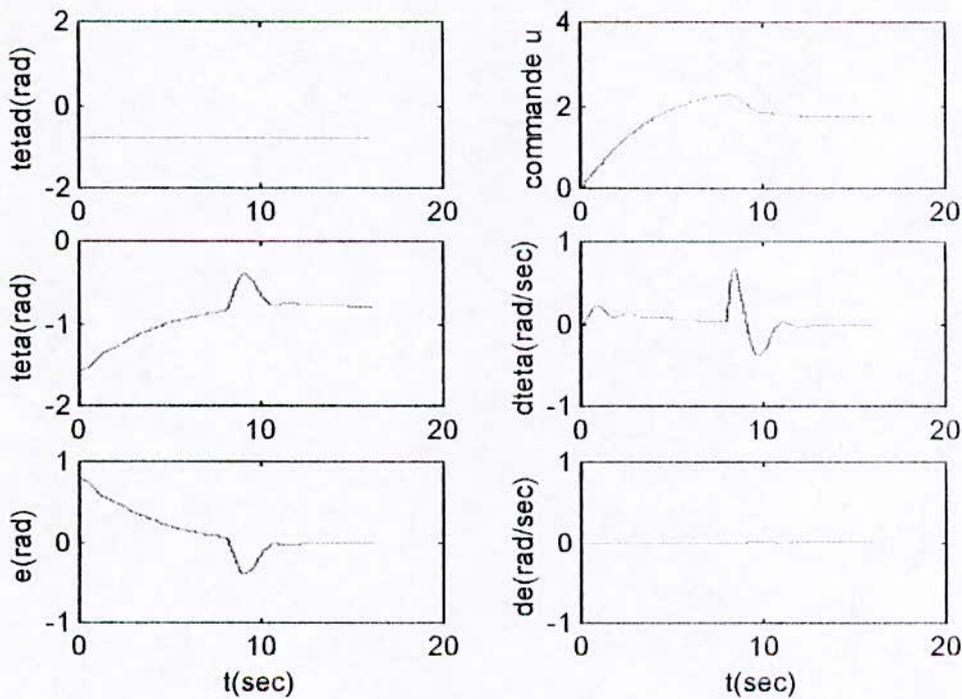


Figure V-34 : résultats de simulation du bras à un degré de liberté avec une charge par application des AsG . $G_s = 0.094$

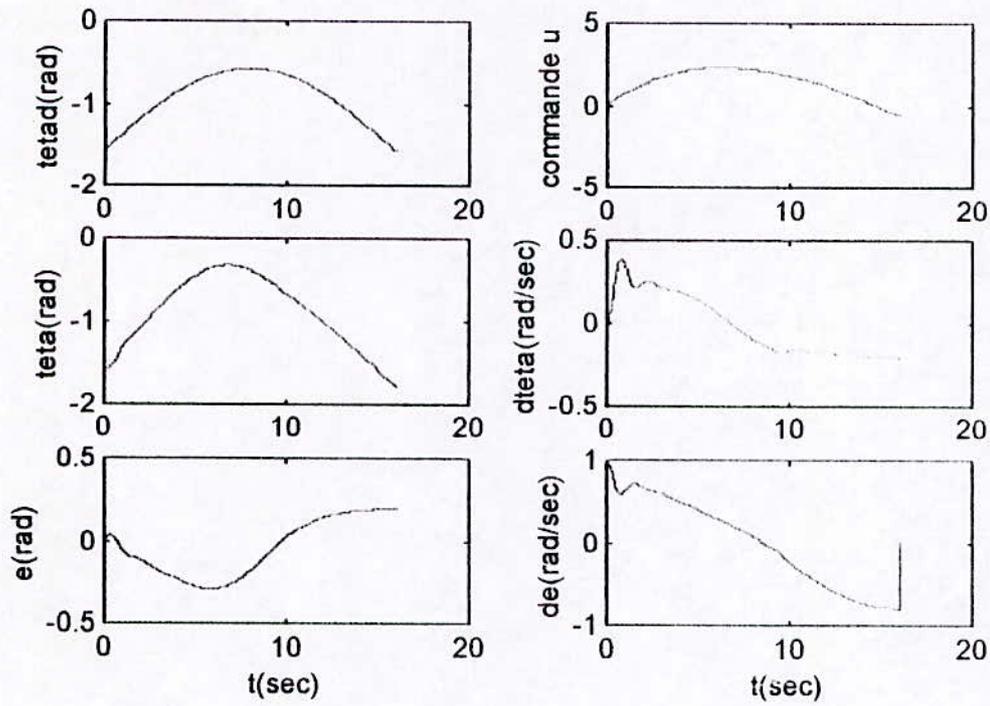
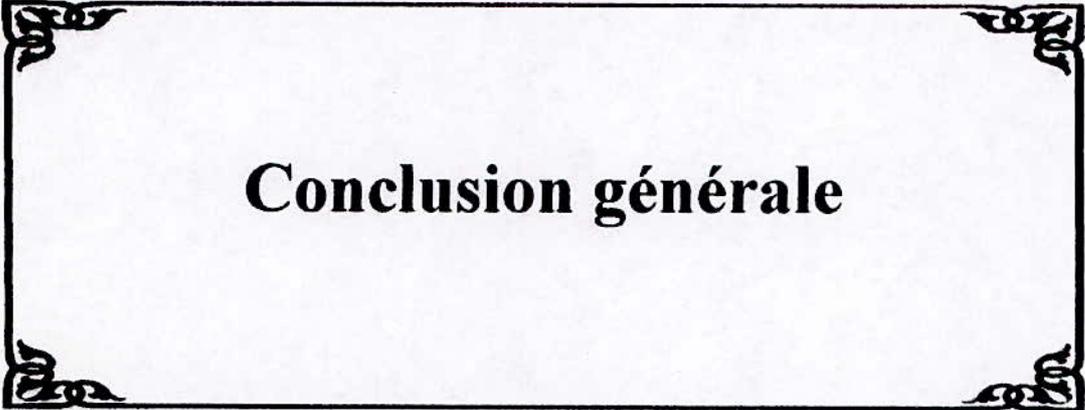


Figure V-35 : résultats de simulation du bras à un degré de liberté avec optimisation par AsG (cas d'un suivi d'une trajectoire). $G_s = 0.061$



Conclusion générale

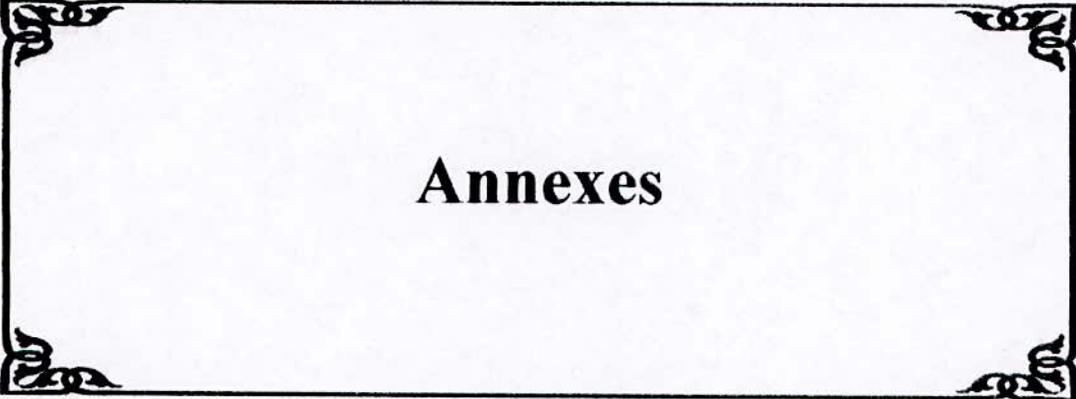
Conclusion générale

Les résultats de ce travail donnent une estimation sur les performances du réglage par logique floue dans le domaine de robotique, et la capacité des AsG dans la recherche des valeurs du gain de sortie du régulateur permettant d'obtenir des réponses optimales. Cela fera éviter les simulations en prenant des valeurs du gain au hasard.

Les simulations ont été faites pour un bras à un degré de liberté pour une consigne échelon, portant une charge et une poursuite d'une trajectoire d'allure sinusoïdale, ainsi que le bras PUMA à 2 ddl pour une entrée échelon.

Nous avons laissé en perspective les points suivants :

- Application des AsG pour une commande PID (recherche de k_u , k_p , k_i).
- Utilisation des formes autres que la triangulaire pour les fonctions d'appartenance.
- Optimisation des formes des fonctions d'appartenance par AsG.



Annexes

Modélisation dynamique d'un bras à un seul degré de liberté (articulation rotative) :

Soit la figure (A.1) d'un bras à un degré de liberté :

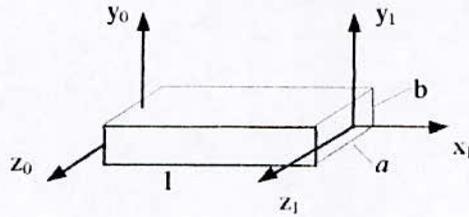


Figure A-1 : Bras à un degré de liberté

La matrice de transformation est donnée par :

$$T = A = \begin{bmatrix} C & -S & 0 & lC \\ S & C & 0 & lS \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\frac{\partial T}{\partial q} = \begin{bmatrix} -S & -C & 0 & -lC \\ C & -S & 0 & lC \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Calcul de la matrice d'inertie :

La matrice d'inertie est :

$$J = \begin{bmatrix} \frac{1}{3}ml^2 & 0 & 0 & -\frac{1}{2}ml \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{1}{2}ml & 0 & 0 & m \end{bmatrix}$$

Cette matrice a six moments d'inertie d'ordre 2 :

$$\int_{-l}^l x^2 dm = \frac{1}{3}ml^2$$

$$\int_{\frac{b}{2}}^{\frac{b}{2}} y^2 dm = \int_{\frac{a}{2}}^{\frac{a}{2}} z^2 dm = 0$$

$$\int xy dm = \int xz dm = \int yz dm = 0$$

Trois moment d'inertie d'ordre 1 :

$$\int_{-l}^0 x dm = -\frac{1}{2} ml$$

$$\int_{\frac{b}{2}}^{\frac{b}{2}} y dm = \int_{\frac{a}{2}}^{\frac{a}{2}} z dm = 0$$

Un moment d'inertie d'ordre 0 :

$$\int_m dm = m$$

Modélisation dynamique :

$$D_{11} = \text{Trace} \left(\frac{\partial T}{\partial q} J \left(\frac{\partial T}{\partial q} \right)^T \right)$$

$$= \frac{1}{3} ml^2$$

$$D_{111} = \text{Trace} \left(\frac{\partial^2 T}{\partial^2 q} J \left(\frac{\partial T}{\partial q} \right)^T \right)$$

$$\frac{\partial^2 T}{\partial^2 q} = \begin{bmatrix} -C & S & 0 & -lC \\ -S & -C & 0 & -lS \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

D'où :

$$D_{111} = 0$$

$$D_1 = -m \cdot g^T \cdot \frac{\partial T}{\partial q} \cdot r_1$$

$$= \frac{1}{2} m \cdot g \cdot l \cdot C$$

L'effort de l'actionneur est alors donné par :

$$\Gamma = ku = \frac{1}{3} ml^2 \ddot{q} + b \dot{q} + \frac{1}{2} mglC$$

De cette dernière équation on a :

$$\ddot{q} = \frac{3}{ml^2} (ku - b\dot{q} + 1/2 mglC)$$

En faisant le changement de variable suivant :

$$X_1 = q$$

$$X_2 = \dot{q}$$

On obtient le système d'état suivant :

$$\dot{X}_1 = X_2$$

$$\dot{X}_2 = \frac{3}{ml^2} (ku - bX_2 + 1/2mg l C)$$

Paramètres du bras à un degré de liberté :

$$l = 1 \text{ m}$$

$$m = 10 \text{ kg}$$

$$b = 10 \text{ N/rad/s}$$

$$k = 20 \text{ Nm/V}$$

$$g = 9.81 \text{ m/s}^2$$

Modélisation du bras PUMA à 2 D.D.L par le formalisme d'Euler-Lagrange :

Par application de la formule générale de l'équation dynamique on trouve les forces généralisées suivantes :

$$\begin{aligned}\Gamma_1 &= \left(\frac{1}{3} m_1 l^2 + \frac{4}{3} m_2 l^2 + m_2 l^2 C_2 \right) \ddot{q}_1 + \left(\frac{1}{3} m_2 l^2 + \frac{1}{2} m_2 l^2 C_2 \right) \ddot{q}_2 - m_2 l^2 S_2 \dot{q}_1 \dot{q}_2 \\ &\quad - \frac{1}{2} m_2 l^2 S_2 \dot{q}_2^2 + \left(m_1 g \frac{1}{2} + m_2 g l \right) + \frac{1}{2} m_2 g l C_{12} + b_1 \dot{q}_1 \\ \Gamma_2 &= \left(\frac{1}{3} m_2 l^2 + \frac{1}{2} m_2 l^2 C_2 \right) \ddot{q}_1 + \frac{1}{3} m_2 l^2 \ddot{q}_2 + \frac{1}{2} m_2 l^2 S_2 \dot{q}_1 \\ &\quad \frac{1}{2} m_2 g l C_{121} + b_2 \dot{q}_1\end{aligned}$$

Avec $l_1 = l_2 = l$

b_i : coefficients de frottement visqueux,

Γ_i : effort de l'actionneur du segment i .

Pour des actionneurs idéaux :

$$\Gamma_1 = k_1 \cdot u_1$$

$$\Gamma_2 = k_2 \cdot u_2$$

$k_i (i=1,2)$: coefficients de proportionnalité des actionneurs.

Calcul de la matrice d'inertie

La matrice d'inertie est donnée par :

$$J = \begin{bmatrix} \int x^2 dm & \int xy dm & \int xz dm & \int x dm \\ \int xy dm & \int y^2 dm & \int yz dm & \int y dm \\ \int xz dm & \int yz dm & \int z^2 dm & \int z dm \\ \int x dm & \int y dm & \int z dm & \int dm \end{bmatrix}$$

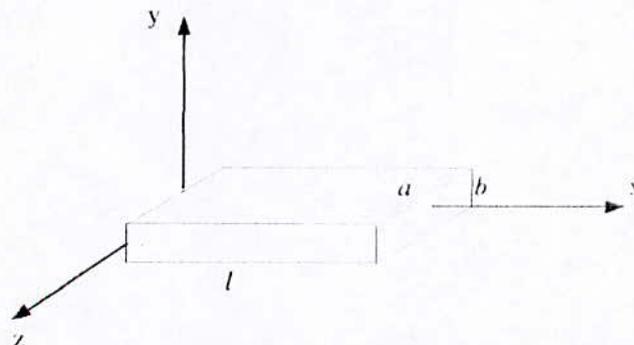


Figure A-2 : segment sous forme parallépipède

La matrice d'inertie a :

six moment d'inertie d'ordre 2 :

$$\int x^2 dm = \frac{1}{3} ml^2$$

$$\int y^2 dm = \frac{1}{12} mb^2$$

$$\int z^2 dm = \frac{1}{12} mc^2$$

$$\int xy dm = 0$$

$$\int xz dm = 0$$

$$\int yz dm = 0$$

- Trois moment d'inertie d'ordre 1 :

$$\int x dm = \frac{1}{2} ml$$

$$\int y dm = 0$$

$$\int z dm = 0$$

- Un moment d'inertie d'ordre 0 :

$$\int dm = m$$

Ces formules sont valables pour les deux segments, d'où les matrices :

- Segment 1 :

$$J_1 = \begin{bmatrix} \frac{1}{3} m_1 l^2 & 0 & 0 & -\frac{1}{2} m_1 l \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{1}{2} m_1 l & 0 & 0 & m_1 \end{bmatrix}$$

- Segment 2 :

$$J_2 = \begin{bmatrix} \frac{1}{3} m_2 l^2 & 0 & 0 & -\frac{1}{2} m_2 l \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ -\frac{1}{2} m_2 l & 0 & 0 & m_2 \end{bmatrix}$$

Les paramètres a et b sont négligeables.

Les deux matrices de transformations sont :

$$A_1 = \begin{bmatrix} C_1 & -S_1 & 0 & a_1 C_1 \\ S_1 & C_1 & 0 & a_1 S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2 = \begin{bmatrix} C_2 & -S_2 & 0 & a_2 C_2 \\ S_2 & C_2 & 0 & a_2 S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T_2 = \begin{bmatrix} C_{12} & -S_{12} & 0 & a_2 C_{12} + a_1 C_1 \\ S_{12} & C_{12} & 0 & a_2 S_{12} + a_1 S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Modélisation dynamique :

Calcul des termes D_{ij}

$$D_{ij} = \sum_{p=\max(i,j)}^N \text{Trace} \left(\frac{\partial T_p}{\partial q_j} J_p \left(\frac{\partial T_p}{\partial q_i} \right)^T \right)$$

$$D_{11} = \text{Trace} \left\{ \frac{\partial T_1}{\partial q_1} J_1 \left(\frac{\partial T_1}{\partial q_1} \right)^T + \frac{\partial T_2}{\partial q_1} J_2 \left(\frac{\partial T_2}{\partial q_1} \right)^T \right\}$$

$$D_{12} = D_{21} = \text{Trace} \left\{ \frac{\partial T_2}{\partial q_1} J_2 \left(\frac{\partial T_2}{\partial q_2} \right)^T \right\}$$

$$D_{22} = \text{Trace} \left\{ \frac{\partial T_2}{\partial q_2} J_2 \left(\frac{\partial T_2}{\partial q_2} \right)^T \right\}$$

$$\frac{\partial T_1}{\partial q_1} = \begin{bmatrix} -S_1 & -C_1 & 0 & -a_1 S_1 \\ C_1 & -S_1 & 0 & a_1 C_1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\frac{\partial T_2}{\partial q_1} = \begin{bmatrix} -S_{12} & -C_{12} & 0 & -a_2 S_{12} - a_1 S_1 \\ C_{12} & -S_{12} & 0 & a_2 C_{12} + a_1 C_1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\frac{\partial T_2}{\partial q_2} = \begin{bmatrix} -S_{12} & -C_{12} & 0 & -a_2 S_{12} \\ C_{12} & -S_{12} & 0 & a_2 C_{12} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

On obtient les éléments D_{ij} après calcul :

$$D_{11} = \frac{1}{3} m_1 l^2 + \frac{4}{3} m_2 l^2 + C_2 m_2 l^2$$

$$D_{12} = D_{21} = \frac{1}{3} m_2 l^2 + \frac{1}{2} C_2 m_2 l^2$$

$$D_{22} = \frac{1}{3} m_2 l^2$$

Calcul des termes D_{ijk} :

les termes D_{ijk} sont données par :

$$D_{ijk} = \sum_{p=\max(i,j,k)}^N \text{Trace} \left(\frac{\partial^2 T_p}{\partial q_j \partial q_k} J_p \left(\frac{\partial T_p}{\partial q_i} \right)^T \right)$$

$$D_{111} = \text{Trace} \left\{ \frac{\partial^2 T_1}{\partial q_1^2} J_1 \left(\frac{\partial T_1}{\partial q_1} \right)^T + \frac{\partial^2 T_2}{\partial q_1^2} J_2 \left(\frac{\partial T_2}{\partial q_1} \right)^T \right\}$$

$$D_{112} = D_{121} = \text{Trace} \left\{ \frac{\partial^2 T_2}{\partial q_2 \partial q_1} J_2 \left(\frac{\partial T_2}{\partial q_1} \right)^T \right\}$$

$$D_{122} = \text{Trace} \left\{ \frac{\partial^2 T_2}{\partial q_2^2} J_2 \left(\frac{\partial T_2}{\partial q_1} \right)^T \right\}$$

$$D_{211} = \text{Trace} \left\{ \frac{\partial^2 T_2}{\partial q_1^2} J_2 \left(\frac{\partial T_2}{\partial q_2} \right)^T \right\}$$

$$D_{212} = D_{221} = \text{Trace} \left\{ \frac{\partial^2 T_2}{\partial q_1 \partial q_2} J_2 \left(\frac{\partial T_2}{\partial q_2} \right)^T \right\}$$

$$D_{222} = \text{Trace} \left\{ \frac{\partial^2 T_2}{\partial q_2^2} J_2 \left(\frac{\partial T_2}{\partial q_2} \right)^T \right\}$$

$$\frac{\partial^2 T_1}{\partial q_1^2} = \begin{bmatrix} -C_1 & S_1 & 0 & -a_1 C_1 \\ -S_1 & -C_1 & 0 & -a_1 S_1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\frac{\partial^2 T_2}{\partial q_1 \partial q_2} = \begin{bmatrix} -C_{12} & S_{12} & 0 & -a_2 C_{12} \\ -S_{12} & -C_{12} & 0 & -a_2 S_{12} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\frac{\partial^2 T_2}{\partial q_1^2} = \begin{bmatrix} -C_{12} & S_{12} & 0 & -a_2 C_{12} - a_1 C_1 \\ -S_{12} & -C_{12} & 0 & -a_2 S_{12} - a_1 S_1 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

$$\frac{\partial^2 T_2}{\partial q_2^2} = \begin{bmatrix} -C_{12} & S_{12} & 0 & -a_1 C_{12} \\ -S_{12} & -C_{12} & 0 & -a_1 S_{12} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

Après calcul on obtient :

$$D_{111} = 0$$

$$D_{112} = D_{121} = -\frac{1}{2} S_2 m_2 l^2$$

$$D_{122} = -\frac{1}{2} S_2 m_2 l^2$$

$$D_{211} = \frac{1}{2} S_2 m_2 l^2$$

$$D_{212} = D_{221} = 0$$

$$D_{222} = 0$$

Calcul des paramètres D_i :

Les termes D_i sont donnés par :

$$D_i = -\sum_{p=1}^N m_p \cdot g^T \cdot \frac{\partial T_p}{\partial q_i} \cdot r_p$$

avec $g^T = [0, -g, 0, 1]$

$$D_1 = -m_1 g^T \frac{\partial T_1}{\partial q_1} r_1 - m_2 g^T \frac{\partial T_2}{\partial q_1} r_2$$

$$D_2 = -m_2 g^T \frac{\partial T_2}{\partial q_2} r_2$$

Après calcul on obtient :

$$D_1 = \frac{1}{2} m_1 g C_1 + m_2 g \left(\frac{1}{2} C_{12} + C_1 \right)$$

$$D_2 = \frac{1}{2} m_2 g C_{12}$$

On pose :

$$\alpha = \frac{1}{3} (m_1 l^2 + 4m_2 l^2 + 3m_2 l^2 C_2)$$

$$\beta = m_2 l^2 \left(\frac{1}{3} m_2 + \frac{1}{2} m_2 C_2 \right)$$

$$\gamma = \frac{1}{3} m_2 l^2$$

$$g_1 = \left(\frac{1}{2} m_1 g l + m_2 g l \right) C_1 + \frac{1}{2} m_2 g l C_{12}$$

$$g_2 = \frac{1}{2} m_2 g l C_{12}$$

$$f_1 = -\frac{1}{2} m_2 l^2 S_2 \ddot{q}_2 - m_2 l^2 S_2 \dot{q}_1 \dot{q}_2$$

$$f_2 = \frac{1}{2} m_2 l^2 S_2 \dot{q}_1^2$$

d'où :

$$\Gamma_1 = k_1 u_1 = \alpha \ddot{q}_1 + \beta \ddot{q}_2 + f_1 + g_1 + b_1 \dot{q}_1$$

$$\Gamma_2 = k_2 u_2 = \beta \ddot{q}_1 + \gamma \ddot{q}_2 + f_2 + g_2 + b_2 \dot{q}_2$$

De la relation Γ_1 on a :

$$\ddot{q}_1 = \frac{k_1 u_1 - (\beta \ddot{q}_2 + f_1 + g_1 + b_1 \dot{q}_1)}{\alpha}$$

on met cette dernière équation dans la formule de Γ_2 on trouve :

$$\ddot{q}_2 = \frac{k_2 u_2 - f_2 - g_2 - b_2 \dot{q}_2 + (k_1 u_1 + f_1 + g_1 + b_1 \dot{q}_1) \beta}{\gamma \left(1 - \frac{\beta^2}{\alpha \gamma}\right)}$$

$$\ddot{q}_1 = \frac{k_1 u_1 - f_1 - g_1 - b_1 \dot{q}_1 + (-k_2 u_2 + f_2 + g_2 + b_2 \dot{q}_2) \beta}{\gamma \left(1 - \frac{\beta^2}{\alpha \gamma}\right)}$$

En faisant le changement de variables suivant :

$$X_1 = q_1$$

$$X_2 = \dot{q}_1$$

$$X_3 = q_2$$

$$X_4 = \dot{q}_2$$

On obtient le système d'état suivant :

$$\begin{aligned} \dot{X}_1 &= X_2 \\ \dot{X}_2 &= \frac{k_1 u_1 - f_1 - g_1 - b_1 X_2 + (-k_2 u_2 + f_2 + g_2 + b_2 X_4) \beta}{\gamma \left(1 - \frac{\beta^2}{\alpha \gamma}\right)} \end{aligned}$$

$$\begin{aligned} \dot{X}_3 &= X_4 \\ \dot{X}_4 &= \frac{k_2 u_2 - f_2 - g_2 - b_2 X_4 + (-k_1 u_1 + f_1 + g_1 + b_1 X_2) \beta}{\gamma \left(1 - \frac{\beta^2}{\alpha \gamma}\right)} \end{aligned}$$

Paramètres du bras PUMA à 2 ddl :

$$l_1 = l_2 = l = 1 \text{ m}$$

$$m_1 = m_2 = 10 \text{ kg}$$

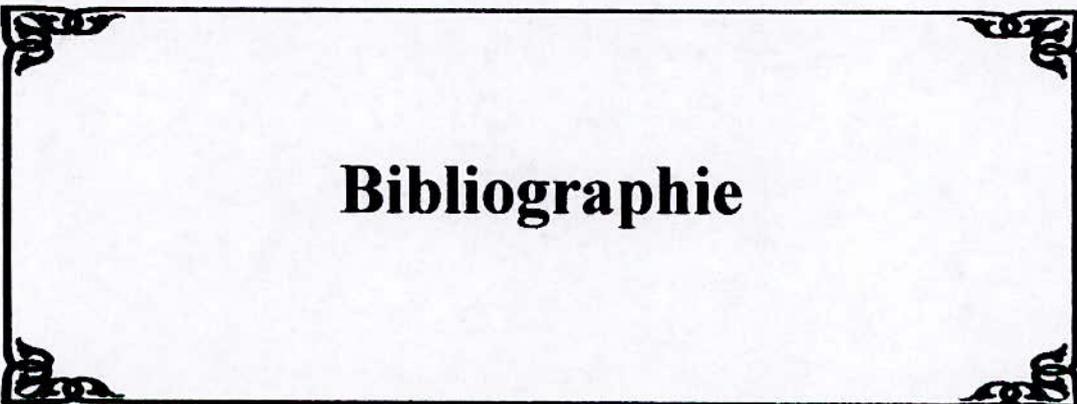
$$b_1 = 75 \text{ N/rad/s}$$

$$b_2 = 10 \text{ N/rad/s}$$

$$k_1 = 40 \text{ Nm/V}$$

$$k_2 = 20 \text{ Nm/V}$$

$$g = 9.81 \text{ m/s}^2$$



Bibliographie

Bibliographie

- [ARA 94] ARAGO 14, "*logique floue*", ed. MASSON, 1994.
- [BAC 97] BACK.T, HAMMEL.U et SCWEFEL.H-P, "*Evolutionary Computation: Comments on the History and Current State*", IEEE transaction, Vol 1, No 1, Avril 1997.
- [BEN 97] BENKRID.M et MAMMA.M, "*Modélisation et commandes de bras manipulateurs*", Thèse d'ingénieur, ENP, 1997.
- [BUH 94] BUHLER.H, "*Réglage par logique floue*", PPUR, 1994.
- [COI 95] COIFFET.C, "*Robots: définitions et classification*", techniques de l'ingénieur (traité mesure et contrôle), Doc R7700, Paris, 1995.
- [CRA 86] CRAIG.J, "*Introduction to robotics: Mechanics and control*", Ed. Addison Wsley, 1986.
- [DJE 96] DJEDDI.N et DRIBINE.D, "*Modélisation cinématique et dynamique des bras manipulateurs*", Thèse d'ingénieur, ENP, 1996.
- [FAL 98] FALL.P.EL-H et LASSOUAOUI.N, "*Application de la logique floue à la commande d'un bras manipulateur: conception d'un logiciel de simulation et de commande ROBSICOM (Robots Simulation and Commands)*", Thèse d'ingénieur, ENP, 1998.
- [FU 87] FU.S.K, GONZALES R.C et LEE.C.S.G, "*Robotics: Control, Sensing, Vision and Intelligence*", Mac Graw Hill Company, 1987.
- [GAC 93] GACOGNE.L, "*Compte-rendu sur la recherche de tables de contrôleur flou par algorithme génétique*", rapport interne de LAFORIA CNRS-Université Paris VI, 1993.
- [GOR 84] GORLA.P et RENAUD.M, "*Modèles des robots manipulateurs, application à la commande*", Cepadues-Edition, Paris, 1984.
- [MAN 96] MAN.K.F, TANG.K.S et KWONG.S, "*Genetic Algorithms: Concepts and Applications*", IEEE transaction, Vol 43, No. 5, Octobre 1996.
- [MIC 96] MICHALEWICZ.Z, "*Genetic Algorithms+ Data Structures= Evolution Programs*", ed. Springer, 1996.