



UNIVERSITE DE BATNA
FACULTE DE TECHNOLOGIE
DÉPARTEMENT D'ELECTRONIQUE



Mémoire

Présenté en Vue de l'Obtention de Diplôme de

Magistère en Electronique

Option : **ROBOTIQUE**

Par : **GHENNAM Abdellatif**
Ingénieur d'Etat en Electronique

Thème

**Commande Compliant Intelligente
d'un Bras Manipulateur Rigide pour
des Applications de Chirurgie Médicale**

Soutenu le : 26/01/2012

Jury d'Examen :

- Dr **SLIMANE Nouredine**, MC 'A', **Président de Jury**, Départ. d'Electronique, Univ. de Batna.
- Dr **ABDESSEMED Yassine**, MC 'A', **Rapporteur**, Départ. d'Electronique, Univ. de Batna.
- Dr **MESSAI Abderraouf**, MC 'A', **Examineur**, Départ. d'Electronique, Univ. de Constantine.
- Dr **AOUGHLENT Saïd**, MC 'A', **Examineur**, Département d'Electronique, Univ. de Batna.



REMERCIEMENTS

REMERCIEMENTS

Je tiens à remercier tout d'abord mon directeur de recherche Dr. Abdessamed Yacine, pour la qualité de leur encadrement et de leur soutien à tous points de vue.

Notre profonde gratitude et notre entière reconnaissance vont à tous ceux qui ont voulu accepter de juger notre travail.

Mes remerciements vont également à l'ensemble des enseignants et des responsables du département d'Electronique qui ont contribué à notre formation par leurs conseils et leur aide.

Je remercie mon ami Amine pour m'avoir soutenu et aidé lors des longues heures de réalisation du prototype de ce projet.

Bien évidemment, j'adresse une pensée à ma mère qui m'a soutenu et encouragé durant ces années.

Cette section ne serait pas complète si je n'avais pas un mot pour toutes les autres personnes que j'ai rencontrées ici. Ils sont trop nombreux pour être cités. Sans eux, une bonne partie du projet n'aurait pu avoir lieu.

TABLE DES MATIERES

TABLE DES MATIERES

INTRODUCTION GENERALE	01
CHAPITRE I	
La commande compliant et son application dans la chirurgie Médicale	
I.1 Introduction.....	03
I.2 fonctionnement du système robotique.....	04
I.3 Mode de fonctionnement en défaillance.....	04
I.4 Mécanisme d'équilibrage.....	04
I.4.1 Contrepoids.....	05
I.4.2 Ressort linéaire.....	06
I.4.3 Ressort angulaire.....	06
I.5 La compliance.....	07
I.5.1 Introduction.....	07
I.5.2 Le besoin de compliance en Robotique.....	08
I.5.3 Définition de la compliance.....	09
I.5.4 Comment obtenir un comportement compliant de la part d'un robot ?.....	11
I.5.5 La compliance logicielle : les commandes compliantes.....	12
I.5.5.1 Commande par retour d'effort explicite.....	12
I.5.5.2 Commande en impédance.....	13
I.5.5.3 Commande par raideur active.....	14
I.5.5.4 Commande hybride.....	15
I.5.6 Les organes compliantes.....	17
I.5.7 Compliance intrinsèque à la structure mécanique.....	19
I.5.8 Les actionneurs compliantes.....	20
I.5.9 Conclusions : comparaison des différentes méthodes.....	20
CHAPITRE II	
Le logiciel Matlab/simmechanics et la réalité virtuelle	
II.1 Introduction.....	23
II.2 FONCTIONS PRINCIPALES DE SIMMECHANICS.....	24
II.3 LIBRAIRIE DE SIMMECHANICS.....	25
II.4 LA BOITE A OUTILS DE LA REALITE VIRTUELLE.....	27
II.5 CONCLUSION.....	29
CHAPITRE III	
Commande des Robots manipulateur	
III.1 Introduction.....	30
III.2 Commande des bras manipulateurs	30
III.3 Commande dans l'espace libre.....	31
III.3.1 Commande par articulation.....	31
III.3.2 Commande jacobéenne.....	32
III.3.3 Commande par découplage non linéaire.....	32
III.3.4 Commande fondée sur une fonction de Lyapunov.....	32
III.3.5 Commande passive.....	33
III.3.6 Commande adaptative.....	33
III.3.7 Commande prédictive.....	33
III.3.8 Commande robuste.....	33
III.3.9 Commande optimale.....	34
III.4 Commande dans l'espace contraint.....	34

III.5	Généralités sur la sélection de la loi de commande.....	35
III.6	régulation classique de type P.D.....	35
	III.6.1 Commande articulaire par un régulateur du type P.D.....	36
III.7	conclusion.....	38

CHAPITRE IV

Commande des Robots manipulateur avec Les réseaux de neurones artificiels (compliance naturel)

IV.1	Introduction.....	39
IV.2	Neurone biologique.....	40
	IV.2.1 Le corps cellulaire.....	40
	IV.2.2 Les dendrites.....	40
	IV.2.3 L'axone.....	40
	IV.2.4 Synapse.....	41
	IV.2.5 Fonctionnement.....	41
IV.3	Représentation d'un réseau de neurone artificiel.....	42
	IV.3.1 Unités du traitement.....	42
	IV.3.2 Connexions entre les unités.....	43
	IV.3.3 Mise en activité et règles de la sortie.....	44
	IV.3.4 Réseaux avec les fonctions de la mise en activité du seuil.....	44
	IV.3.5 Règle d'apprentissage.....	46
IV.4	La commande neuronale d'un bar manipulateur à deux degré de liberté.....	47
	IV.4.1 Le robot manipulateur.....	47
	IV.4.2 Paramètres du robot.....	47
	IV.4.3 Equations générales du robot.....	48
	IV.4.4 Les positions désirées.....	48
	IV.4.5 Les Matrices $M(\theta)$, $N(\theta, \dot{\theta})$, $G(\theta)$ et $H(\dot{\theta})$	49
IV.5	La commande neuronale pour T_p négligeables.....	50
	IV.5.1 L'algorithme de rétro propagation du gradient.....	53
	4.5.1.1 Introduction de l'algorithme.....	53
	4.5.1.2 Algorithme de rétro propagation du gradient.....	57
IV.6	La commande neuronale pour T_p non négligeables.....	57
IV.7	Conclusion.....	58

CHAPITRE V

Résultats de simulation

V.1	Introduction.....	59
V.2	Les courbes obtenues pour les deux angles (θ_1, θ_2)	60
V.3	Les courbes obtenues pour les deux vitesses $(\dot{\theta}_1, \dot{\theta}_2)$	62
V.4	L'évolution de l'erreur en fonction de la charge.....	64
V.5	Conclusion.....	70
	Conclusion générale	71
	Annexe A	73
	Annexe B	77
	Bibliographie	81

INTRODUCTION GENERALE

INTRODUCTION GENERALE

Quand on parle de robotique, plusieurs idées viennent à l'esprit de chacun de nous. Historiquement, nous pourrions nous référer aux premiers concepts et automates de l'antiquité ou aux premiers robots comme à des personnages de la mythologie.

Même le mot robot a sa propre histoire. Séparer la science de la science-fiction n'est pas une chose aisée, surtout lorsque en robotique nous cherchons parfois à faire réalité la fiction.

Un exemple de l'influence des fictions nous est donné par les lois de la robotique. C'est au siècle dernier que le « boom » de la robotique industrielle a amorcé l'explosion des thèmes de recherche. A cette époque les robots étaient conçus en respectant les contraintes imposées par le milieu industriel, comme la répétabilité, la précision dans la réalisation des tâches, le respect des cadences de production, etc. Prenons l'exemple des bras manipulateurs industriels des années 80 très semblables aux bras industriels actuels, ils se déplacent à très grandes vitesses sur des trajectoires calculées hors ligne dans des environnements protégés et structurés, sans interaction avec les opérateurs humains.

C'est avec les développements scientifiques, spécifiquement de l'électronique et de l'informatique mais aussi automatique, mathématique, mécanique, matériaux, que la technologie robotique a progressé.

Les robots actuels sont dotés d'une « intelligence » qui leur donne une certaine autonomie qui va leur permettre de se diffuser dans de nouveaux domaines. Nous parlons maintenant de robots d'exploration, de robots d'intervention, de robots ludiques, de robots de service, des robots chirurgicaux, etc.

Dans ce travail de thèse, nous nous sommes intéressés aux robots manipulateurs pour des applications en robotique chirurgicale.

Les systèmes de santé occidentaux font aujourd'hui face à une pression grandissante causée en partie par le vieillissement de la population et la sédentarité. Les ressources financières, humaines et techniques sont tour à tour sollicitées par cette croissance de la demande.

Différentes solutions à ce problème peuvent être explorées, telle l'éducation pour vaincre la sédentarité ou une gestion différente des ressources. Les innovations technologiques représentent l'une des avenues de solutions étudiées. L'amélioration des instruments chirurgicaux, la conception d'un nouveau lit motorisé ou la création d'un nouvel appareil de diagnostic sont des exemples d'innovations technologiques destinées à améliorer les soins prodigués aux patients.

La robotique chirurgicale est l'un de ces domaines d'innovations technologiques. Les robots chirurgicaux peuvent accomplir diverses tâches dans une salle d'opération, allant du robot autonome qui effectue une coupe osseuse à l'aide de l'imagerie 3D au système qui permet au chirurgien d'opérer à distance tout en compensant pour les tremblements de ses mains.

Précision, efficacité et accessibilité aux soins sont des exemples d'avantages potentiels de ces robots. Quelques systèmes ont déjà obtenu les accréditations nécessaires et sont présentement utilisés pour aider les chirurgiens en Europe et en Amérique.

Les forces et faiblesses des robots différant grandement de celles des humains, les robots chirurgicaux ont plusieurs avantages qui expliquent leur utilisation croissante en salle d'opération. Cependant, la robotique chirurgicale demeure un domaine qui comporte de nombreux défis.

Organisation de la thèse

Ce document est organisé de la façon suivante :

- Le chapitre 1 : est dédié à la commande compliant et son application dans la chirurgie médicale.
- Le chapitre 2 : présente le logiciel Matlab/simmechanics et la réalité virtuelle.
- Le chapitre 3 : traite la Commande des robots manipulateurs avec un régulateur PID.
- Le chapitre 4 : traite la Commande des robots manipulateurs avec les réseaux de neurones (compliance naturel).
- Le chapitre 5 : présente les résultats de simulation obtenus.

Chapitre I

La commande compliante et son application dans la chirurgie Médicale

CHAPITRE I

LA COMMANDE COMPLIANTE ET SON APPLICATION DANS LA CHIRURGIE MÉDICALE

I.1 Introduction

La robotique médicale est un des domaines privilégiés des travaux de recherche qui s'appuie en particulier sur ses compétences reconnues dans des domaines des asservissements visuels. La robotique médicale a pour objectif principal l'assistance au geste médical pour une amélioration de la qualité des soins. Les spécificités de ce domaine de recherche proviennent, entre autres, de la nécessaire interaction du système robotique avec le patient et le praticien qui doit se faire en toute sécurité, ainsi que du respect des contraintes particulières de la salle d'opération.

La sécurité des robots, consiste à éliminer les risques de blessures en rendant impossible la présence d'une personne dans l'espace de travail du robot pendant son fonctionnement. Cette approche industrielle de « cellule robotique » a prouvé son efficacité depuis des décennies. Dans une situation d'examen médical, le robot doit fonctionner en contact avec le patient et être opéré par un médecin : la cellule robotique apparaît ainsi hors de question. L'exclusion de toute personne de l'espace d'opération du robot est impossible. Ce qui amène une redéfinition des exigences spécifiques à ce type d'utilisation.



I.2 fonctionnement du système robotique

La figure I.1 illustre un schéma simple du fonctionnement du système robotique. Le médecin qui effectue l'examen agit sur l'interface utilisateur en haut du schéma. Un ordinateur central commande les moteurs du robot (via des amplificateurs de puissance) et reçoit les informations de posture des moteurs et du robot via des capteurs. Enfin, le patient est au bout de la chaîne du système, c'est-à-dire l'environnement de travail du robot.

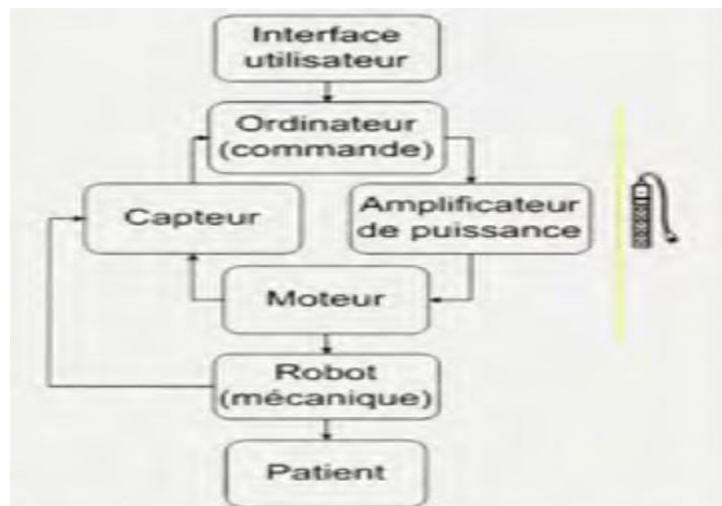


Figure I.1 Schéma du fonctionnement normal du robot

I.3 Mode de fonctionnement en défaillance

La mise hors tension soudaine du robot est une situation comportant des forces et des mouvements imprévus susceptibles d'être dangereux. Le système robotique lorsqu'il y a coupure de l'alimentation électrique. La chaîne est brisée et l'utilisateur n'a plus le contrôle du système. Le patient se retrouve seul avec le robot libre de bouger. La structure mécanique du robot est alors soumise aux effets de la gravité.

I.4 Mécanisme d'équilibrage

Les mécanismes d'équilibrage mécanique disponibles sur le marché sont peu nombreux : les contrepoids, les ressorts linéaires, les ressorts angulaires et les ressorts angulaires à couple constant. L'installation de ceux-ci requiert souvent des câbles, des poulies, des tiges ou des structures parallélogrammes pour orienter les vecteurs de forces déployées par les ressorts vers les axes annulant la force gravitationnelle. Puisque les membres du robot sont en mouvement les uns par rapport aux autres, il faut isoler chaque mouvement indépendamment des autres membres pour les balancer.

I.4.1 Contrepoids

Le contrepoids est le mécanisme d'équilibrage considéré comme le plus sécuritaire (Jean et al. 1996). Il ne se brise pas par fatigue contrairement aux ressorts. Faisant partie intégrante de la structure des membres d'un robot, le contrepoids balance les membres reliés par des articulations rotoïde contre la gravité en appliquant un couple (la masse du contrepoids multipliée par le bras de levier) égal et opposé de l'autre côté de l'axe de rotation. Le contrepoids est idéal pour les robots sériels et peut aussi être utilisé dans certaines configurations de robots parallèles. Son désavantage majeur est l'ajout de masses réduisant les capacités cinématiques en plus de nécessiter des membres et articulations plus robustes.

La masse plus importante augmente la force cinétique des mouvements les rendant potentiellement plus dangereux.

La figure I.2 illustre la structure robotique complètement équilibrée par des contrepoids. En plus de la complexité accrue de la structure, l'ajout de contrepoids introduit de nouvelles pièces en mouvement, encombrant l'espace de travail

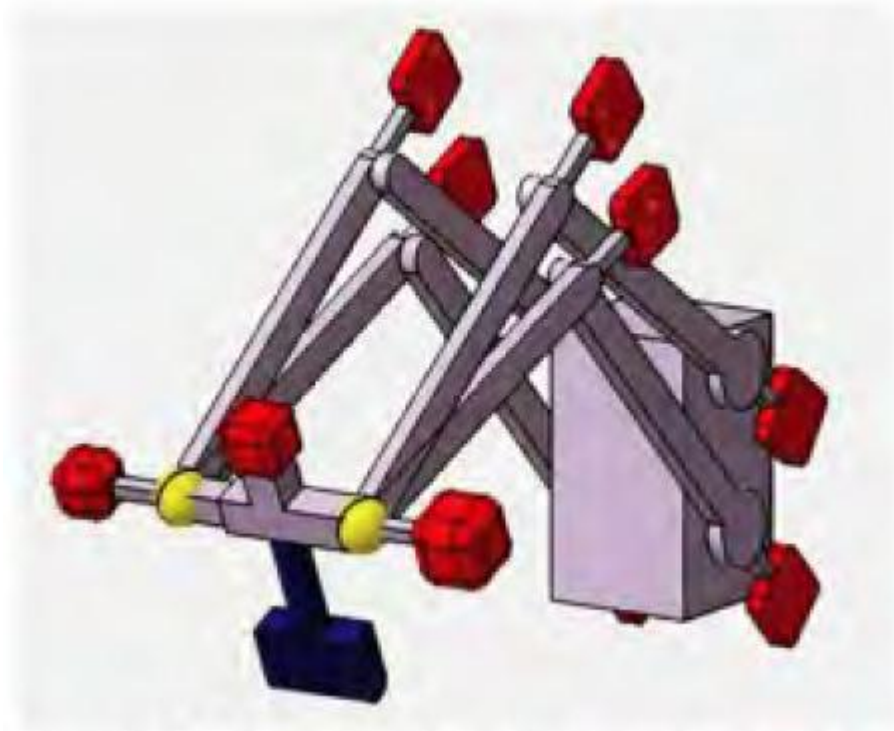


Figure I.2

I.4.2 Ressort linéaire

Un ressort de type linéaire fournit une force proportionnelle à sa déformation. La forme initiale du ressort détermine la polarité de la force. La polarité différencie deux types: les ressorts en extension (figure I.3 (b)) et les ressorts en compression (figure I.3 (a)).

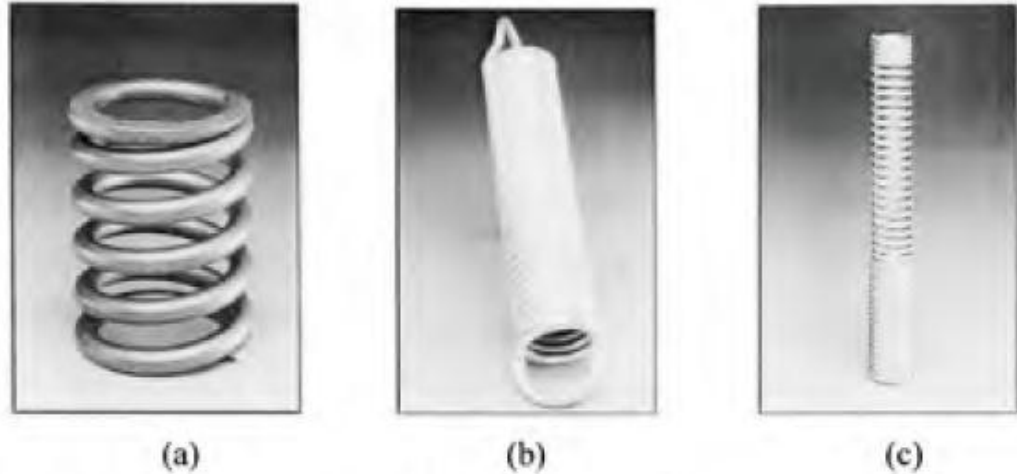


Figure I.3 ressort linéaire

Le ressort de compression possède généralement une déformation totale assez courte Pour éviter qu'il ne fléchisse en se déformant. Il peut aussi offrir une force en extension, plus courte que la compression (figure I.3 (c)). Le ressort exclusivement d'extension, quant à lui, peut fournir une grande déformation. Il est le plus utilisé pour cette raison. Une force de compression peut toujours être remplacée par une force d'extension placée et attachée de l'autre côté du membre. L'utilisation de câbles, de poulies et de parallélogrammes permettent de transposer et d'isoler le vecteur des forces de déformation.

L'équilibrage devient alors exact, mais son assemblage génère un encombrement et rend la structure beaucoup plus complexe.

I.4.3 Ressort angulaire

Les ressorts angulaires offrent un couple proportionnel à la déformation angulaire. Ils sont compacts et s'installent directement sur les articulations rotoïde. Les manufacturiers ont récemment amélioré les techniques de fabrication, ce qui leur permet d'offrir des produits très diversifiés, fabriqués sur mesure. Les plus simples sont composés d'un fil de métal rigide torsadé dont les extrémités prolongées servent de bras de levier pour générer un couple (figure I.4 (a) et I.4 (b)).

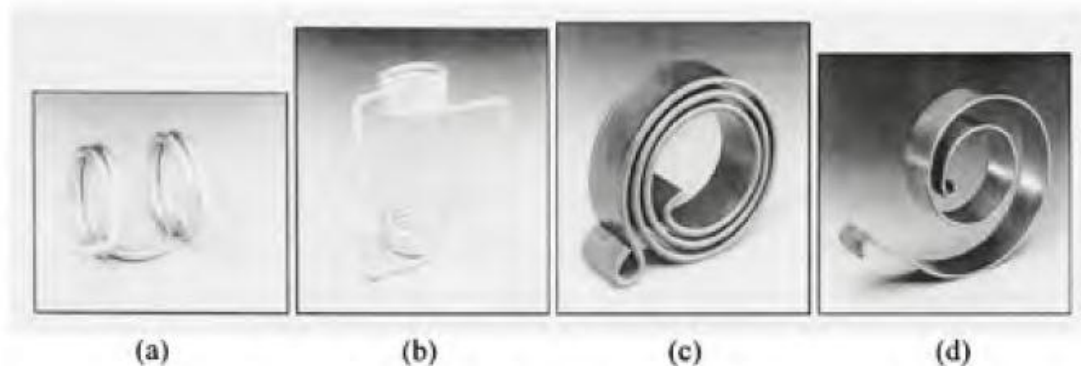


Figure I.4 ressorts angulaires

Comme les ressorts linéaires, les ressorts angulaires peuvent être déformés dans deux directions, mais ils offrent un plus grand déplacement s'ils sont unidirectionnels. La direction du couple requise n'est pas contraignante, puisque les ressorts angulaires s'installent d'un côté comme de l'autre d'une articulation rotoïde.

Il existe également des ressorts angulaires développant un couple constant à n'importe quel angle de déformation, la force n'étant aucunement fonction de la position angulaire (figure I.4 (c) et I.4 (d)). Ils sont, par contre, beaucoup plus gros que les ressorts angulaires à couples progressifs et leurs diamètres varient selon l'angulation.

Le ressort angulaire est le dispositif mécanique privilégié dans l'équilibrage statique du robot dans ce chapitre. Il est compact, facile à installer, n'ajoute pas de masse à la structure et ne requiert pas l'ajout de dispositifs d'attache encombrants, tel que requis par les ressorts linéaires.

Le mécanisme de stabilisation statique privilégié est le ressort angulaire simple. Les paramètres ajustables sont la constante de ressort K , le bras de levier l_k et la position angulaire initiale q_k . La force F engendrée par le ressort peut alors s'exprimer sous une forme linéaire par rapport à l'ensemble des paramètres. La linéarité de la force est un avantage important puisqu'elle permet une formulation linéaire du problème d'optimisation.

I.5 La compliance

I.5.1 Introduction

Ce premier chapitre a pour objectif de définir la compliance en général. Il en montre l'intérêt pour certaines tâches robotisées, puis la formalise.

Ensuite, sont passées en revue et comparées les différentes techniques compliantes. Nous parcourons ainsi la chaîne que forme le robot en contact avec son environnement et envisageons différentes possibilités pour rendre compliant cette chaîne.

I.5.2 le besoin de compliance en Robotique

Les tâches que l'on souhaite robotiser ont souvent été auparavant réalisées par l'homme. Dans la plupart des cas, celui-ci était en contact manuel avec la tâche qu'il effectuait. Il gérait donc (inconsciemment ou non) les efforts de contact engendrés au cours de l'exécution de la tâche. Certaines de ces opérations tendent à être robotisées [3], soit pour décharger l'homme de tâches fastidieuses, soit parce que les tâches demandent une précision ou une dextérité, impossibles à atteindre humainement.

Pour réaliser correctement ces tâches, le robot doit donc, comme l'homme, tenter de s'adapter en permanence aux efforts qu'il exerce sur son environnement (constitué par la ou les pièces en cours de traitement). Nous pouvons citer par exemple :

➤ **Les tâches de préhension d'objets fragiles ou fortement déformables** : en effet, l'éventail des pièces à manipuler par robot est de plus en plus vaste. On ne se limite plus à des objets rigides et solides. Il est intéressant de prévoir que le préhenseur puisse s'adapter à différents objets ; sans changer d'outil, on gagnerait ainsi en productivité. Pour manipuler correctement de tels objets, la force de serrage doit être suffisamment importante pour éviter le glissement, mais elle doit cependant être maîtrisée de façon à éviter de briser ou de déformer l'objet saisi [3].

➤ **L'assemblage de pièces ajustées** : aujourd'hui encore une grande partie des opérations d'assemblage se fait manuellement. En effet, réaliser correctement cette tâche en mode automatique, lorsque les pièces à assembler sont très ajustées (quelques microns par exemple), nécessite une très bonne précision du robot et une répétabilité meilleure que l'ajustement entre les pièces. Ces excellentes performances requises pour le robot sont actuellement techniquement possibles, mais souvent à des coûts prohibitifs. Dans le cas où l'on ne dispose pas d'une précision et d'une répétabilité suffisante, un mauvais positionnement relatif des deux pièces à assembler peut mener à des situations de blocage susceptibles d'empêcher l'insertion ou même de détériorer les pièces en présence. Pour résoudre ce problème, l'assemblage de précision robotisé doit réaliser l'absorption des écarts de positionnement dus au manque de précision du robot. Une solution consiste à rendre flexible la chaîne mécanique fermée, constituée à l'instant du contact par le robot, les pièces et la table de travail. En effet, la flexibilité permet de réduire les efforts de contact entre les pièces pendant la phase d'insertion. Ceci compense les défauts d'alignement et les défauts angulaires dus aux incertitudes de positionnement et d'orientation de l'organe terminal du robot par rapport à l'alésage [3].

➤ **L'ébarbage ou l'ébavurage** : dans de nombreux secteurs industriels, les opérations d'ébarbage et d'ébavurage, et plus généralement de parachèvement de pièces, restent souvent manuelles, la plupart des robots industriels disponibles sur le marché étant dépourvus de la compliance nécessaire pour l'exécution de ces tâches.

L'automatisation de ce type de tâches se heurte à un certain nombre de difficultés, dont les principales sont les suivantes (Figure 1a) [3]:

- Dispersion dimensionnelle des pièces qui sont en général obtenues par moulage. Cette dispersion interdit la programmation rigoureuse de la trajectoire réelle à suivre, parce que inconnue;
- Formes et dimensions aléatoires des bavures qui résultent des plans de joint du moule (ou des usinages effectués);
- Matière souple dans le cas de pièces en matière plastique.

La robotisation d'une telle tâche (Figure I.5) nécessite un contact permanent entre la fraise et la pièce, avec un suivi du contour réel inconnu, et avec une réaction R provenant de la coupe due à l'épaisseur (aléatoire) de la bavure.

De nombreux auteurs ont apporté leur contribution dans ce domaine [3]; on peut, en particulier, citer les travaux de Rodic [3] concernant la modélisation de la tâche d'ébavurage et sa commande permettant de limiter l'effort de contact tout en maintenant une force de coupe constante.

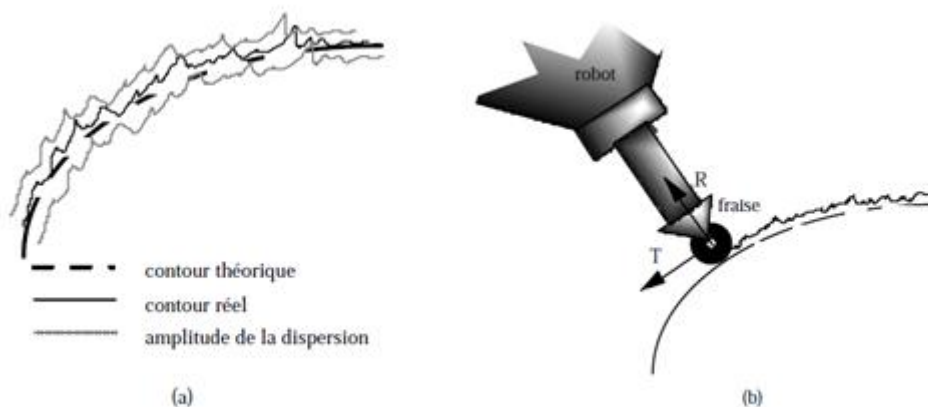


Figure I.5 : Exemple de l'ébavurage

Ces exemples montrent que les tâches mettant en contact le robot avec son environnement nécessitent une certaine souplesse pour ne pas détériorer les pièces en contact. La souplesse est même parfois indispensable pour permettre la réalisation proprement dite de la tâche.

I.5.3 Définition de la compliance

Ce concept de comportement flexible du robot face à son environnement est désigné sous le terme de compliance (La traduction française exacte de ce terme, complaisance, est rarement utilisée).

La compliance peut être définie comme étant la capacité d'un manipulateur à avoir un comportement souple ; le robot compliant peut subir un déplacement sous l'effet d'un effort dû au contact du manipulateur avec son environnement, cet effort restant cependant dans des limites raisonnables pour la tâche considérée.

La compliance définit donc la capacité du manipulateur à s'adapter à son environnement pour la tâche qu'il a à accomplir. En effet, si le contact se faisait avec une raideur infinie, la moindre erreur de position se traduirait par l'apparition de forces excessives au point de contact.

Un contact en un point du dispositif engendre un couple (effort/déplacement) en un autre point du dispositif.

Soit F , le torseur des forces et moments appliqués à ce point et D le torseur des déplacements linéaires et angulaires associés.

On a :

$$F = K(D)D + B(D)\frac{dD}{dt} + I(D)\frac{d^2D}{dt^2} \quad (I.1)$$

Cette expression décrit le mouvement d'un solide soumis à une force de frottement visqueux et à une force élastique de rappel.

Les matrices d'inertie $I(D)$, de frottement $B(D)$ et de raideur $K(D)$ sont des fonctions du déplacement D . Elles dépendent aussi du point considéré.

Ceci amène donc à établir un classement selon les différentes natures de la compliance :

Quand les forces appliquées au solide sont de nature élastique, nous aurons une compliance *élastique*. Dans le cas où les efforts appliqués sont du type frottement visqueux, la compliance sera **anélastique amortie**. Quand les forces agissant sur le solide sont dues aux masses en mouvement, nous avons une compliance **anélastique inertielle**.

Ces trois types de compliance sont le plus souvent couplés mais, très fréquemment, la compliance élastique est prédominante. Dans ce cas, nous pouvons donc écrire :

$$F = K(D) D \quad (I.2)$$

Dans le cas où cette expression est bijective et où K peut être ramené à une expression linéaire autour du point d'application des efforts, nous avons :

$$D = C F \quad (I.3)$$

Dans cette expression, C est la matrice de compliance du dispositif autour du point considéré. Elle représente l'inverse de la matrice de raideur K . D'une manière générale, la matrice C n'est pas diagonale, mais il existe une configuration particulière pour laquelle la matrice de compliance peut être rendue diagonale.

On appellera centre de compliance, le point pour lequel la matrice C est diagonale. Une force latérale exercée en ce point le déplace uniquement dans la direction de la force exercée. De même, un couple appliqué en ce point ne provoque qu'un déplacement angulaire. Le centre de compliance présente donc un découplage des mouvements résultants des efforts appliqués.

Les diverses natures de la compliance étant ainsi définies, nous pouvons encore distinguer deux méthodologies de mise en œuvre de la compliance :

➤ **La compliance passive**, provenant exclusivement de la souplesse des structures et de leur capacité à absorber les erreurs de positionnement entre le robot et la tâche en se déformant naturellement sous l'effet des efforts de contact.

➤ **La compliance active**, nécessitant une détection des efforts de contact et étant pilotée par des actionneurs asservis.

I.5.4 Comment obtenir un comportement compliant de la part d'un robot?

Une fois la notion de compliance définie, il convient de se poser la question de l'obtention d'un tel comportement pour une tâche de contact.

Considérons un robot effectuant une tâche, en contact avec son environnement (Figure I. 6).

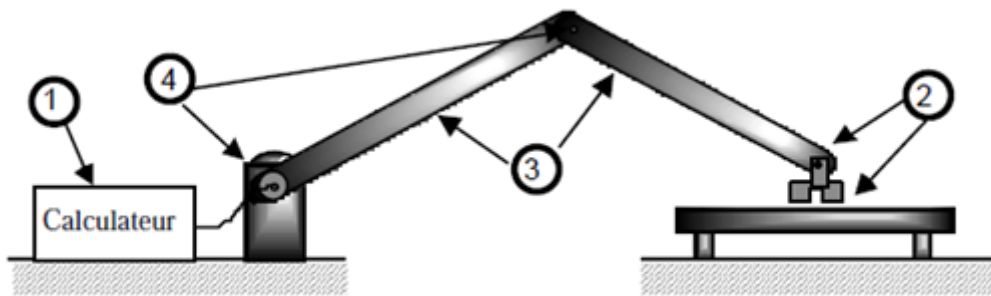


Figure I. 6 : Localisation de la Compliance dans la chaîne cinématique

Nous pouvons schématiser cette chaîne de la façon suivante :

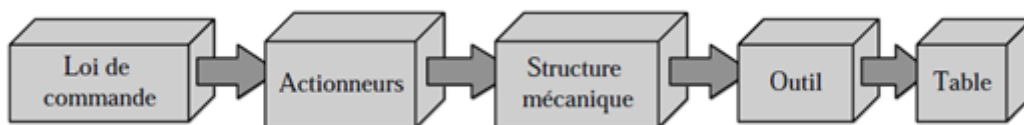


Figure I.7 : Schéma de la chaîne robot- environnement

Un comportement globalement compliant sera obtenu en introduisant de la flexibilité dans cette chaîne.

Dans un premier temps, on peut tenter d'obtenir cette flexibilité de façon logicielle uniquement (1). En effet, cela semble être une solution relativement simple et "propre" à mettre en œuvre, car elle n'implique pas de modification mécanique de l'ensemble robot-environnement.

Une deuxième solution (2), ne modifiant pas non plus la structure même du robot, consiste à ajouter un organe flexible au point de contact. Cet organe sera soit fixé en bout de robot, en tant que poignet compliant, soit sur l'environnement (table compliant).

La troisième solution consiste à concevoir un robot possédant une structure mécanique volontairement souple (solution 3), par l'utilisation de bras flexibles.

La dernière solution envisagée déporte la compliance dans les actionneurs (solution 4). La souplesse de l'actionneur se répercutera ainsi au point de contact, par l'intermédiaire de la matrice jacobéenne.

Nous analysons maintenant ces différents modes d'obtention de la compliance, afin de comparer leurs performances respectives.

I.5.5 La compliance logicielle : les commandes compliantes

Le concept de compliance active nécessite de doter le robot de capteurs. Il utilise ces informations pour contrôler la trajectoire. Ceci confèrera au robot un comportement compliant dans le sens où il minimise, ou tout au moins, limite les efforts de contact. Les techniques de commande compliant peuvent être classées en deux catégories. La solution à adopter dépendra de la tâche que l'on doit accomplir.

Dans une première catégorie, on classera les tâches qui demandent une commande en position, mais qui en cas de contact ne doivent pas engendrer de force de contact infinie (ex : assemblage). Dans ce cas, on donnera uniquement une consigne en position (ou en vitesse selon la nature de la tâche et les performances recherchées).

Dans la seconde catégorie seront classées les tâches dont la force de contact doit être commandée selon une trajectoire de consigne (suivi de contour, ébavurage). Dans ce cas, on aura en général plusieurs consignes, dont au moins une en effort.

I.5.5.1 Commande par retour d'effort explicite

Cette loi de commande est née des travaux de Nervins et Whitney [31].

Avec cette méthode, le robot est commandé en position (ou en vitesse) et la mesure de l'effort est utilisée pour corriger le signal d'entrée. En fait, d'une manière générale, le robot est piloté en position ; il suit sa trajectoire de consigne. Dès qu'un effort extérieur apparaît, c'est-à-dire dès que l'outil du robot rencontre son environnement, l'effort de contact apparaissant est alors détecté. Cette information en effort est ensuite utilisée pour modifier la commande en position. La commande s'adapte donc à l'environnement. La force mesurée intervient sur la commande par l'intermédiaire d'une matrice CF ayant pour dimension l'inverse d'une raideur.

Le déplacement ΔX , duquel la consigne sera diminuée, sera alors proportionnel à la force mesurée :

$$\Delta X = C_F F \quad (I.4)$$

Où C_F est la matrice diagonale de retour d'effort.

La valeur de l'élément C_{fi} sera faible si le degré de liberté correspondant est commandé en position et importante s'il est commandé en effort, de manière à pouvoir augmenter la compliance.

On représente le principe de fonctionnement de cette méthode par le diagramme de la (Figure I.8).

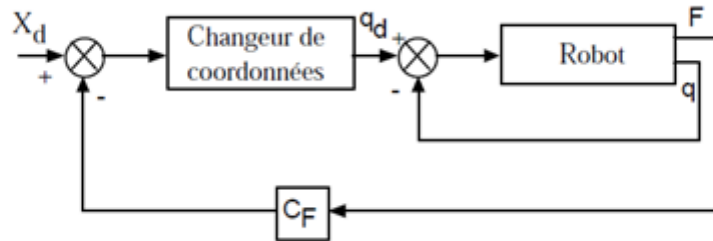


Figure I.8 : Principe de la commande par retour d'effort explicite

On peut utiliser le même principe quand le robot est commandé en vitesse. Dans ce cas, la matrice de retour est une matrice d'amortissement.

I.5.5.2 Commande en impédance

Ce type de commande a été introduit par Hogan en 1985 [31] et a également été étudiée au LAI [31].

Quand le robot est en contact avec l'environnement, il y a interaction entre eux. Il doit donc y avoir compatibilité physique entre eux. L'idée de base de la commande en impédance repose sur le fait que si le robot se comporte comme une impédance mécanique, l'environnement se comportera comme une admittance mécanique ; de sorte que le robot puisse faire face aux contraintes en position imposées par l'environnement.

Le vecteur de commande choisi pour piloter le robot à la fois en position et en vitesse, tout en maintenant un couple de consigne Γ_d est le suivant :

$$\Gamma - \Gamma_d = J^T [K_p (x_d - x) + K_v (\dot{x}_d - \dot{x})] \tag{I.5}$$

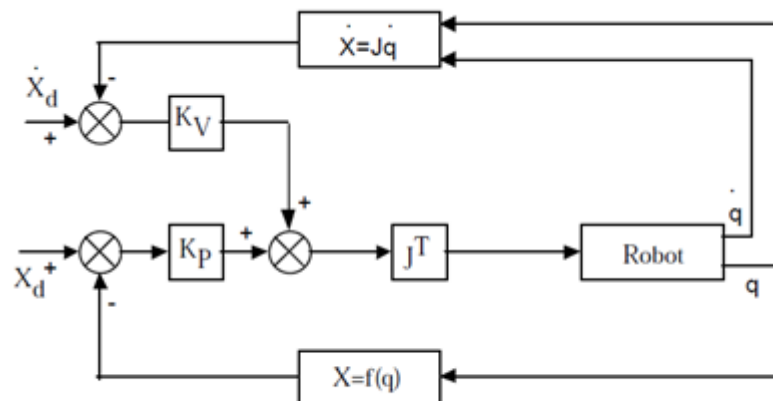


Figure (I.9) : Principe de la commande en Impédance

La matrice K_p est choisie en fonction de la tâche compliant à réaliser. C'est une matrice diagonale dont les termes seront faibles lorsqu'ils correspondent à un degré de liberté commandé en effort et importants quand le degré de liberté est commandé en position. Les éléments de K_v déterminent l'amortissement dans chaque direction. La commande en impédance consiste donc à réguler l'interaction dynamique entre un robot et son environnement directement dans l'espace de la tâche en engendrant la commande précédente dans l'espace articulaire.

L'avantage de cette méthode est qu'elle ne nécessite pas l'inversion de la matrice jacobéenne.

En 1995, Volpe et Khosla [31] ont montré qu'un contrôle d'impédance du second ordre donnait des résultats équivalents à une commande par retour d'effort explicite.

I.5.5.3 Commande par raideur active

En 1980, Salisbury [31] propose de contrôler activement la raideur apparente d'un manipulateur. La raideur est changée de façon logicielle pour pouvoir s'adapter aux évolutions de la tâche. L'idée de cette commande consiste à contrôler simultanément le robot en force et en position.

Cette méthode définit, en statique, une fonction linéaire qui lie les forces d'interaction à la position finale du manipulateur, via une matrice de raideur en coordonnées cartésiennes.

$$F_C = K_C \Delta D_C \quad (I.6)$$

Avec :

F_C : effort de contact

K_C : matrice de raideur souhaitée

ΔD_C : déplacement différentiel

Le vecteur-couple différentiel des actionneurs est donné par la relation suivante :

$$\Delta \Gamma = J^T F_C \quad (I.7)$$

Où J est la matrice jacobéenne du manipulateur.

Donc : $\Delta D_C = J \Delta q$

Où dq représente le vecteur des variables articulaires.

On obtient alors la relation suivante dans l'espace des variables articulaires :

$$\Delta \Gamma = \underbrace{J^T K_C J}_{K_q} \Delta q \quad (I.8)$$

Où K_q représente la matrice de raideur dans l'espace articulaire.

La commande a pour principe de superposer des forces de biais sur le comportement raide précédemment décrit. Cela permet d'appliquer des forces indépendantes de la position. Le réglage de la raideur (donc de la compliance) se fera par l'intermédiaire du gain KF.

$$\Gamma = K_q(q^d - q) + K_v(\dot{q}^d - \dot{q}) + J^T(F^d + K_f(F^d - F)) \tag{I.9}$$

Cette loi de commande est représentée par le diagramme fonctionnel de la (FigureI.10).

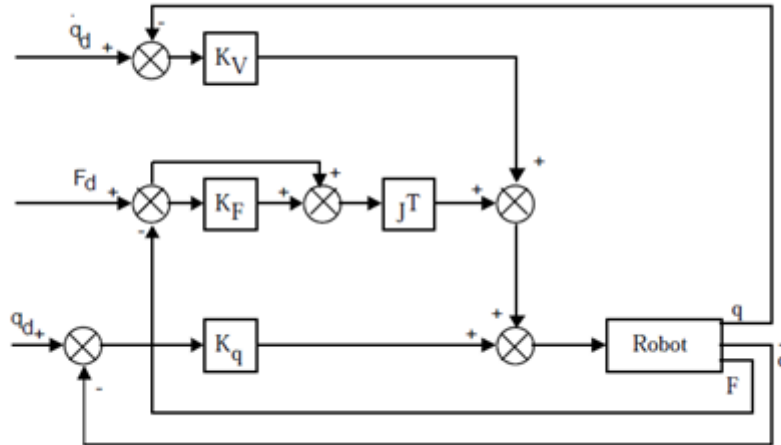


Figure (I.10) : Principe de la commande par raideur active

I.5.5.4 Commande hybride

Ce type de commande a été proposé par Raibert et Craig [31]. Elle a ensuite été reprise par de nombreux auteurs, dont [31].

L'idée originale de cette méthode est déduite de la constatation suivante : un même degré de liberté ne peut être commandé à la fois en force et en position. En effet, ces deux types de contrôle sont antagonistes. D'où l'idée de partitionner l'espace en deux sous-espaces : l'un contenant les degrés de liberté commandés en position et l'autre les degrés de liberté pilotés en force. Puisqu'un même actionneur peut intervenir sur plusieurs degrés de liberté, le principe de la commande hybride consiste à envoyer simultanément aux actionneurs des contributions en position et en effort.

On note S, la matrice de sélection qui désigne le type de commande nécessaire pour chaque degré de liberté.

En général S est diagonale. L'élément si sera égal à 1 si le degré de liberté est commandé en position et 0 s'il est commandé en effort.

Le système subissant une série de contraintes physiques, la commande hybride impose à l'organe terminal une série duale de contraintes artificielles choisies de façon à obtenir la trajectoire désirée et à appliquer à l'environnement les forces souhaitées.

La structure de commande peut être représentée par le schéma fonctionnel de la (Figure I.11).

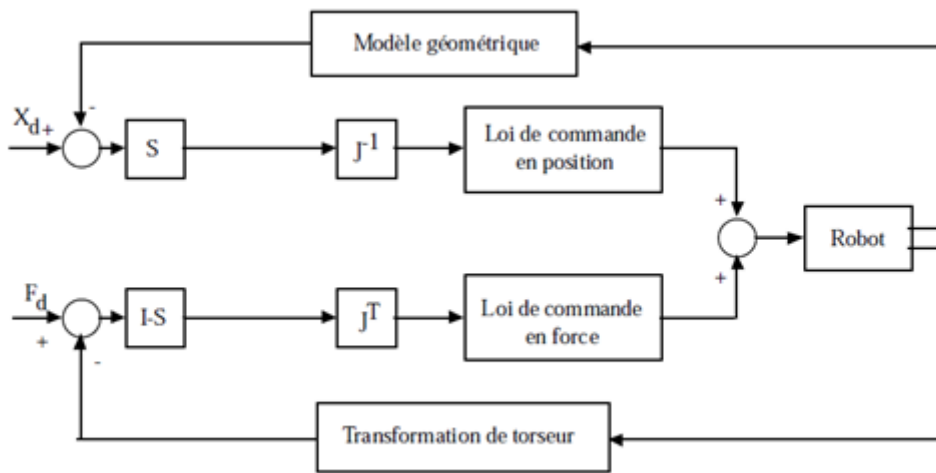


Figure (I.11) : Principe de la commande hybride

Cette technique de commande est bien adaptée aux applications nécessitant des commandes différentes dans différentes directions. Par exemple, l'ébavurage nécessite une commande en position dans l'axe radial et une commande en effort dans l'axe transversal à la surface à ébavurer. Un inconvénient de cette stratégie de commande est qu'elle requiert l'inversion de la matrice jacobéenne. De plus, Berthe [31], dans sa thèse, met en avant un problème d'instabilité cinématique de la commande hybride. En effet, la condition suffisante de stabilité dépend de la matrice jacobéenne, donc de la configuration du robot.

L'approche hybride permet de commander en force et en position, mais elle nécessite un modèle géométrique détaillé de l'environnement. D'autre part, les données provenant des capteurs passent par une sélection liée à la tâche à exécuter. Par contre, une commande en impédance est capable de s'adapter à une classe d'environnement, mais sans contrôler à la fois la force et la position.

Le contrôle simultané en force et en position peut également être obtenu par commande parallèle [31].

La commande parallèle combine les avantages des deux méthodes décrites précédemment. Ici, la commande en force est calculée de façon à prévaloir sur la commande en position en cas de conflit ; les déviations de la tâche planifiée donnent priorité à la commande en effort, en augmentant les gains de cette dernière. Par exemple, on peut utiliser une commande Proportionnelle-Intégrale en effort et une commande Proportionnelle-Dérivée en position.

Cette technique de commande a été rendue adaptative par Siciliano et Villani [31] en agissant sur les dynamiques du modèle ; elle améliore la poursuite de trajectoire, quand les paramètres sont mal estimés.

Nous pouvons aussi citer la commande compliant partagée décrite dans [31]. Elle est utilisée dans le domaine de la télé opération ; les commandes classiques précédemment décrites

fonctionnent bien quand le transfert prend moins d'une seconde entre l'opérateur et le manipulateur. Par contre, des retards de transmission importants (applications spatiales) causent des problèmes d'instabilité. La commande compliant partagée traite localement les forces de contact sans attendre la réaction de l'opérateur à ces efforts. La commande est partagée en ce sens que l'opérateur garde son contrôle manuel classique d'un côté et de l'autre, le robot à une fonction compliant autonome.

Les lois de commande conférant un comportement compliant au robot ne sont pas toujours adaptées ; parfois complexes à mettre en œuvre expérimentalement et industriellement, elles ne donnent pas toujours un résultat satisfaisant. En effet, afin de garantir de bonnes précisions statique et dynamique, les concepteurs de robots industriels confèrent à sa structure une grande raideur mécanique, que l'on peut la plupart du temps considérer comme infinie. Dans ce cas, l'obtention d'un comportement compliant par logiciel ne confère pas toujours assez de souplesse au système, ou bien est parfois trop peu rapide. Les techniques de compliance décrites par la suite ont pour but d'introduire de la flexibilité dans la chaîne mécanique formée par les actionneurs, les bras du robot, l'outil, les pièces et la table de travail.

I.5.6 Les organes compliants

Cette technique consiste à interposer dans la chaîne mécanique un organe compliant, soit au niveau du poignet du robot, soit dans la table de travail.

L'avantage de cette solution est qu'elle peut s'adapter à n'importe quel robot existant. En effet, aucune aptitude particulière du robot n'est requise en ce qui concerne sa compliance naturelle, puisqu'on insère dans le premier cas entre le robot et son organe terminal un dispositif qui rendra le système complet compliant ; la seconde solution n'intervient pas directement sur le robot. L'organe se déformera donc sous l'effet des efforts de contact pour permettre la correction des erreurs de positionnement et/ou d'orientation.

Depuis plus de trente ans, de nombreux *organes compliants passifs* ont été étudiés et développés, tant dans les laboratoires de recherche universitaires que dans les bureaux d'étude industriels. L'intérêt de cette technique est qu'elle est simple à mettre en œuvre, car elle ne nécessite ni capteur ni loi de commande sophistiquée : elle utilise seulement l'élasticité appropriée des composants de l'organe compliant. Un intérêt supplémentaire de cette technique est que l'on obtient en général des réponses rapides. De plus, les systèmes compliants passifs sont peu chers et relativement fiables en exploitation en milieu industriel.

Le plus connu et le plus utilisé industriellement est sans doute le RCC (Remote Center Compliance) . Ce dispositif élastique constitué de liens flexibles est à l'origine de nombreux autres organes. On peut trouver un grand nombre de références traitant de ces dispositifs [31].

Ces organes passifs sont en grande partie destinés au domaine de l'assemblage. Ceci peut s'expliquer par le fait que, dans ce domaine, les critères à satisfaire sont les suivants : réussite de l'insertion le plus rapidement possible et sans blocage. Il y a d'autre part, une forte demande industrielle, car on admet qu'environ un tiers des tâches sont des tâches d'assemblage, les organes passifs étant simples et sans commande sophistiquée, ils sont souvent à même de remplir ces objectifs.

Par contre, l'inconvénient majeur de telles méthodes réside dans le fait que la compliance est fixée a priori pour une tâche donnée et cela nécessite de changer l'élément compliant pour toute nouvelle application.

Pour certaines applications, de suivi de contour par exemple, il est nécessaire de limiter l'effort de contact, ou même de l'asservir. Dans ce cas, on fait appel à la *compliance active*.

Contrairement aux dispositifs précédemment décrits, les *organes compliants actifs* sont pilotés par des actionneurs asservis. Leurs déplacements en réponse à une mesure de l'effort de contact peuvent être contrôlés.

On aura donc un bouclage entre les efforts ressentis par les différents capteurs et l'action correctrice. Dans ce cas, les déplacements sont engendrés par des actionneurs qui répondent à une loi de commande cherchant à annuler ou à minimiser les efforts de contact.

La compliance active utilise un (ou des) capteur(s) comme moyen de détection de l'effort et/ou de la position entre le robot et la tâche. Une loi de commande minimisera alors un critère qui, suivant les cas, peut être une erreur de position, ou un effort (effort dans le sens de l'insertion en cas d'assemblage par exemple).

Un système de compliance active nécessite trois types d'éléments :

- Des capteurs pour détecter la position relative du robot par rapport à son environnement,
- Une loi de commande pilotant la compliance,
- Un ou plusieurs actionneurs exécutant la correction.

Pour détecter les erreurs de positionnement on peut utiliser des capteurs de déplacement, mais compte-tenu de la très faible amplitude de ces déplacements, il est souvent préférable de mettre en œuvre des capteurs d'effort montés en association avec des éléments déformables pré-étalonnés. La sensibilité s'en trouve ainsi fortement améliorée. D'autres technologies (capteurs de proximité, capteurs tactiles ou vision) peuvent être envisagées, mais la sensibilité et la qualité de la mesure sont moins bonnes.

Là encore, comme en compliance passive, on trouve de nombreuses réalisations d'organes compliants dans la littérature.

Les organes compliant fixés à l'extrémité du robot ont donc largement été étudiés. Ils offrent l'avantage d'être adaptables à n'importe quel robot industriel. Par contre, leur inconvénient majeur vient de la masse non négligeable qu'ils ajoutent en bout de bras. Les inerties, ainsi présentes quand le robot est en mouvement, engendrent des problèmes d'imprécision et d'oscillation qui rendent complexe la commande de position du robot. De plus, ils limitent la charge utile du robot.

Pour pallier ces inconvénients, tout en insérant toujours un organe compliant au point de contact, il suffit de doter l'environnement de souplesse sous la forme d'une *table compliant* passive ou active. L'utilisation d'une telle table permet ainsi de conserver une charge transportable maximale pour le robot.

Notons que plusieurs réalisations de poignets compliant peuvent se transformer sans problème majeur en table compliant.

I.5.7 Compliance intrinsèque à la structure mécanique

Une autre solution compliant prend en compte la souplesse de la structure mécanique. Pour cela, des équipes de recherche se sont penchées sur l'étude des segments flexibles.

La souplesse de la structure mécanique d'un robot provient généralement de la souplesse des bras.

Cette souplesse résulte en général d'un mauvais choix des éléments technologiques: sections trop faibles, bras trop longs... Cependant cette souplesse peut être volontairement recherchée dans le but de doter le robot d'un comportement compliant, ce qui semble constituer un avantage puisqu'il n'est pas alors nécessaire de rajouter un organe compliant à son extrémité. Par ailleurs, cette souplesse se traduisant technologiquement par des bras de sections et de masses moindres, il est toujours intéressant d'alléger le plus possible le robot afin de diminuer les masses en mouvement et les inerties : il peut en résulter alors une augmentation de la rapidité et une diminution de la taille (donc du coût) des actionneurs.

Malheureusement, cette structure allégée rend le robot assez fortement oscillatoire en mode transitoire et sa commande en devient alors très délicate.

De nombreuses équipes de recherche ont travaillé sur la modélisation et la commande de bras flexibles. Nous pouvons par exemple citer les travaux de Matsuno [31] concernant la commande hybride force/position d'un manipulateur à deux degrés de liberté, formé de deux bras flexibles en séries. Ce manipulateur est capable de se déplacer le long d'une surface à une vitesse de 2.1 cm/s tout en maintenant un effort de 19.6 N. Des expérimentations menées à une vitesse supérieure ont conduit à des problèmes de vibration causant l'instabilité. Les résultats obtenus sont donc souvent décevants : en effet, si les solutions théoriques, ou de laboratoire, peuvent exister pour

éliminer ces phénomènes oscillatoires néfastes, leur mise en œuvre au plan matériel et logiciel est lourde, onéreuse et en général non-compatible avec les impératifs industriels de coût, de fiabilité, d'encombrement... La modélisation et la commande de tels segments étant par ailleurs largement étudiées, nous nous sommes naturellement intéressées à la solution proposant l'emploi d'actionneurs compliant.

1.5.8 Les actionneurs compliant

Une autre stratégie pour rendre un robot industriel compliant consiste non plus à ajouter un organe compliant, poignet ou table, pour permettre une adaptation entre le robot et son environnement mais à déporter la compliance dans les actionneurs. A priori, cette technique est intéressante, car elle n'ajoute pas de masse supplémentaire en bout de robot. Cela minimise les problèmes dynamiques dus à une inertie trop importante et permet d'augmenter la charge portée par le robot. De plus, le même actionneur ayant pour mission de déplacer le robot en l'absence de contact, puis de contrôler sa compliance lors du contact, nous conférons au manipulateur une compliance active, car cette dernière pourra être pilotée selon la tâche à accomplir. La loi de commande appliquée à l'actionneur devra, selon la configuration du robot, asservir le robot en position ou contrôler sa compliance.

Nous allons considérer les trois énergies généralement utilisées pour piloter un robot : électrique, hydraulique et pneumatique.

1.5.9 Conclusions : comparaison des différentes méthodes

L'étude précédente nous permet de dresser le tableau comparatif suivant, énonçant les avantages et inconvénients des techniques passives et actives.

	AVANTAGES	INCONVENIENTS
COMPLIANCE PASSIVE	<ul style="list-style-type: none"> • peu chère • fiable • simple à mettre en oeuvre • réponses rapides 	<ul style="list-style-type: none"> • dédiée à une seule application • ne contrôle pas l'effort de contact
COMPLIANCE ACTIVE	<ul style="list-style-type: none"> • adaptable suivant la tâche • commande fine des efforts de contact 	<ul style="list-style-type: none"> • nécessite une loi de commande • nécessite des actionneurs et capteurs supplémentaires • complexe • onéreuse • fiabilité moyenne en milieu industriel

De plus, l'éventail des solutions envisagées dans ce chapitre fait ressortir des points forts et des points faibles pour chacune des méthodes. Nous les résumons dans le tableau 1.

Nous pouvons ainsi définir une méthodologie pour concevoir un système compliant.

Dans un premier temps, nous nous intéressons à la tâche compliant que nous souhaitons effectuer. Il nous faut savoir quel type de tâche nous aurons à effectuer. Si la tâche est susceptible de changer dans le temps ou si l'environnement est mal connu ou de raideur variable, la stratégie sera plutôt de choisir une compliance active.

Par contre, si c'est une tâche unique et répétitive, nous opterons pour une solution passive dont la compliance est définie au départ. En effet, il n'est pas nécessaire de développer une compliance active qui peut s'avérer coûteuse et exagérément complexe si la tâche est très simple et répétitive.

Une fois ce choix fait, il faut définir la tâche proprement dite et la situer dans son contexte (par exemple dans une chaîne de production) et savoir si l'on dispose déjà d'éléments imposés (par exemple un robot). Ceci éliminera certaines solutions non adaptées à la situation. Par exemple l'utilisation de tables compliantes dans certaines chaînes d'assemblage est impensable car trop complexe à mettre en œuvre. De même que l'on se dirigera plutôt vers la fixation d'un organe compliant à l'extrémité du robot déjà présent sur le site pour pouvoir s'adapter facilement à l'environnement. Le cadre de la manipulation nous indiquera aussi l'énergie à adopter (pneumatique, électrique....).

TECHNOLOGIE UTILISEE	AVANTAGES	INCONVENIENTS
Organe compliant passif	<ul style="list-style-type: none"> • rapide, simple, fiable • adaptable à un robot industriel 	<ul style="list-style-type: none"> • peu adaptatif • masse additionnelle
Organe compliant actif	<ul style="list-style-type: none"> • adaptable à plusieurs tâches • adaptable à un robot industriel 	<ul style="list-style-type: none"> • masse additionnelle • coût élevé • fiabilité moyenne
Table compliant	<ul style="list-style-type: none"> • pas de masse additionnelle • adaptable à un robot industriel 	<ul style="list-style-type: none"> • difficilement applicable sur une chaîne de production
Actionneur électrique	<ul style="list-style-type: none"> • pas de modification du robot 	<ul style="list-style-type: none"> • irréversibilité du réducteur
Actionneur fluide	<ul style="list-style-type: none"> • pas de modification du robot 	<ul style="list-style-type: none"> • mouvement de translation • mise en oeuvre onéreuse • frottements importants
Bras flexible	<ul style="list-style-type: none"> • robot souple, léger 	<ul style="list-style-type: none"> • conception d'un robot particulier • oscillatoire • contrôle complexe • peu adaptatif
Transmission flexible	<ul style="list-style-type: none"> • simple 	<ul style="list-style-type: none"> • oscillations • contrôle très complexe • peu adaptatif
Commande compliant	<ul style="list-style-type: none"> • pas de modification du robot • adaptable à plusieurs tâches 	<ul style="list-style-type: none"> • parfois lent • trop complexe

Tableau 1 : Comparaison des solutions compliantes

Chapitre II

Le logiciel Matlab/simmechanics et la réalité virtuelle

CHAPITRE II

LE LOGICIEL MATLAB/SIMMECHANICS ET LA RÉALITÉ VIRTUELLE

II.1 Introduction

SimMechanics est un des outils du logiciel MATLAB pour la modélisation de systèmes mécaniques **3D** (tridimensionnels) au sein de l'environnement Simulink. Au lieu de dériver et de programmer des équations, on peut utiliser cet outil de simulation pour construire un modèle composé de corps, de liaisons et de contraintes, et d'éléments de force reflétant la structure du système. Une animation **3D**, générée automatiquement, permet de visualiser la dynamique du système. On peut importer des modèles complets, avec la masse, l'inertie, les contraintes et la géométrie **3D** à partir de plusieurs systèmes de **CAO** (Conception Assistée par Ordinateur).

SimMechanics peut servir à développer des systèmes robotiques, des appareils chirurgicaux, des systèmes de train d'atterrissage et divers autres systèmes physiques, afin d'arriver à la production voir (figure II.1).

Le développement d'un nouveau produit dans l'industrie passe par une phase de simulation et une phase de réalisation comme il est montré dans la (figure II.1).

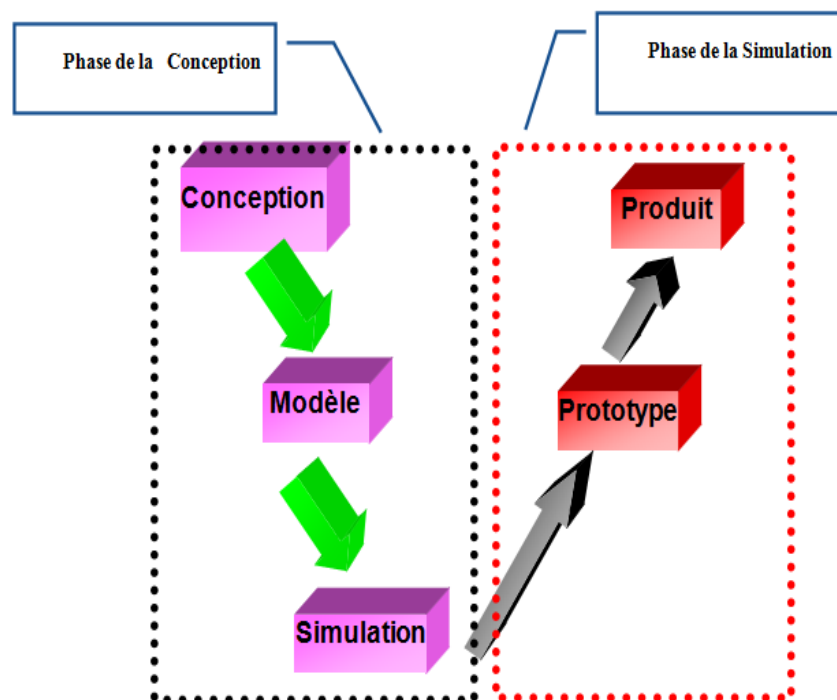


Figure (II.1) : Conception basée sur un Modèle

II.2 FONCTIONS PRINCIPALES DE SIMMECHANICS

L'utilisation de SimMechanics, sous l'environnement de Matlab, permet de construire un modèle de systèmes mécaniques 3D en particulier un robot manipulateur dans notre étude voir

(Figure II.2), comme si on assemble un système physique réel. En utilisant les corps, les systèmes de coordonnées, les liaisons et les éléments de force, on définit un système mécanique capable de se connecter directement à d'autres modèles Simulink. Cette approche vous permet d'étendre les fonctionnalités de SimMechanics et de réutiliser les modèles Simulink existants. Il est possible d'enregistrer des modèles combinant des blocs Simulink et SimMechanics en tant que sous-systèmes pour les réutiliser dans de nombreuses autres applications.



Figure (II.2) : Bras Manipulateur (Réal) SRX10 (4 axes + pince de préhension)

Ces sous-systèmes peuvent inclure :

- Des ressorts linéaires et non-linéaires modélisés sous Simulink.
- Des modèles de traînée aérodynamique qui relient des distributions de pression à des composants dans le domaine de l'aéronautique, comme des ailerons et des gouvernes de direction .
- Des sous-systèmes de suspension active pour véhicules, notamment des mécanismes de protection anti-retournement et des modules de commande
- Des modèles de pneus pour avions et véhicules terrestres.

En utilisant des blocs de capteurs dans **SimMechanics**, on peut mesurer les valeurs des différentes grandeurs physiques (force, couple et vitesse, position), puis transmettre ces signaux à des blocs **Simulink** standard. Les blocs d'actionneur permettent aux signaux Simulink d'attribuer des valeurs à n'importe laquelle de ces variables pour actionner un système mécanique modélisé. Les blocs de capteurs et d'actionneurs permettent de connecter un algorithme de contrôle développé dans Simulink avec un modèle SimMechanics, afin d'élaborer la partie commande.

Le contrôleur dans notre projet est conçu en utilisant un modèle non-linéaire développé dans Simulink.

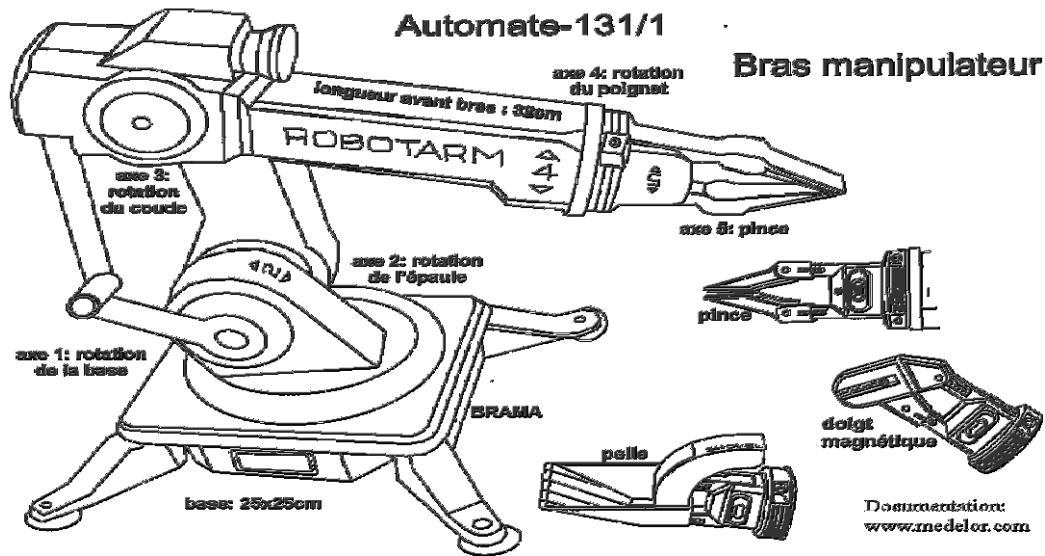


Figure (II.3) : Schéma d'un bras manipulateur avec organe préhenseur.

La boîte à outils (toolbox) pour la modélisation physique simmechanics de Math Works contient un ensemble complet de composants avec des paramètres critiques identifiés pour le programmeur, qui rendent possible de construire rapidement un système physique. Afin de modéliser de nouveaux composants, on trouve dans la boîte à outils un ensemble de blocs de base pour leur utilisation dans la modélisation de machines. On peut automatiquement régler leurs paramètres pour être concordant avec les données de mesure. Le résultat est que le travail de construire des modèles physiques demande moins de temps ; et laisse à l'ingénieur plus de temps pour optimiser la conception.

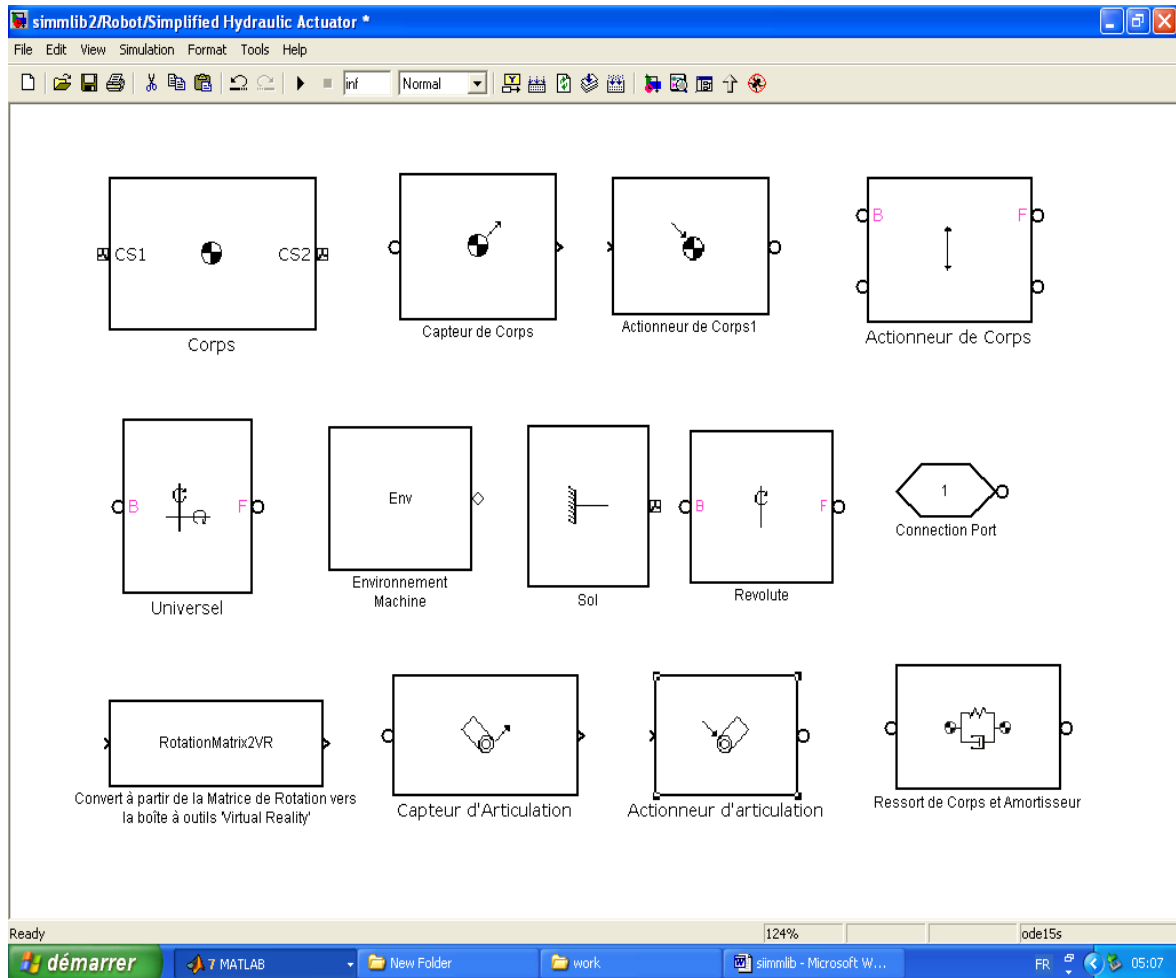


Figure (II.4) : Boîte à outils principale de Simmechanics

Les différents composants de base de la boîte à outils de **simmechanics** sont comme suit :

- Des corps (organes principaux d'une machine).
- Des capteurs et des actionneurs.
- modèles de traînée aérodynamique qui relie des distributions de pression à des composants dans le domaine de l'aéronautique, comme des ailerons et des gouvernes de direction.
- Des ressorts linéaires et non-linéaires modélisés sous **Simulink**.
- Des modèles de traînée aérodynamique qui relie des distributions de pression à des composants dans le domaine de l'aéronautique, comme des ailerons et des gouvernes de direction .
- Des sous-systèmes de suspension active pour véhicules, notamment des mécanismes de protection anti-retournement et des modules de commande
- Des modèles de pneus pour avions et véhicules terrestres

Un exemple de programme de simulation avec Simmechanics d'un bras manipulateur simplifié est montré dans la (figure II.5).

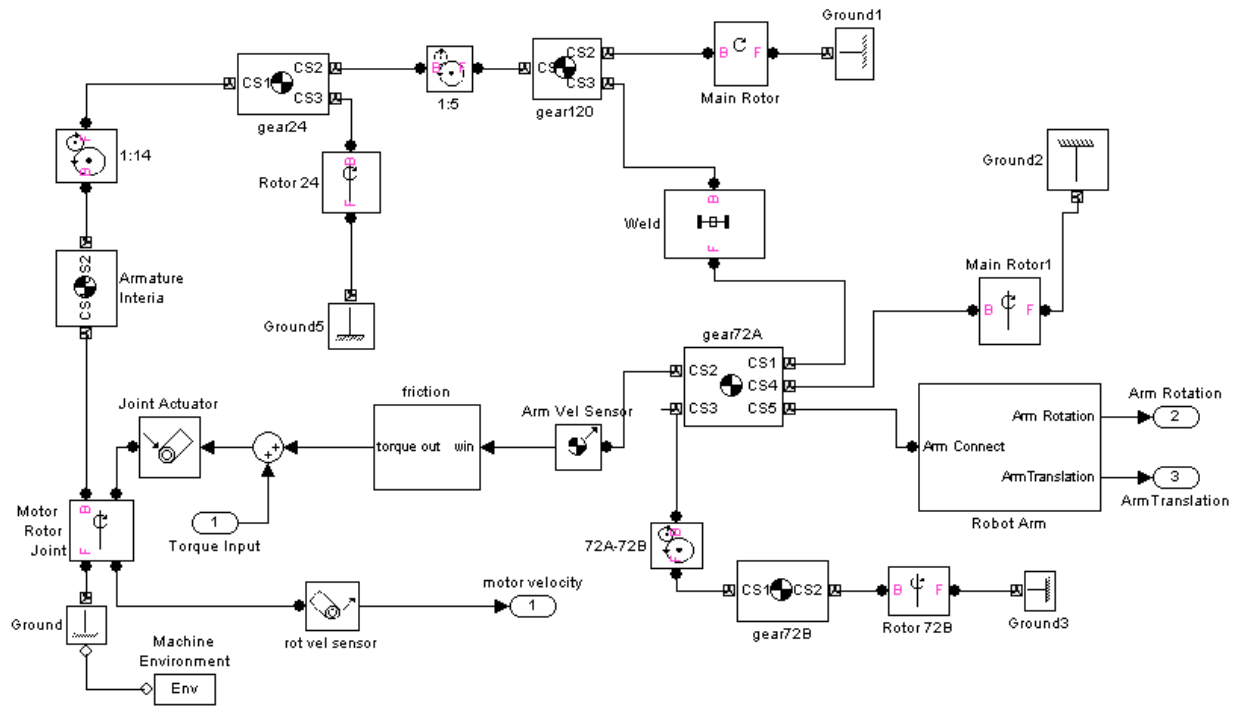


Figure (II.5)

II.4 LA BOITE A OUTILS DE LA REALITE VIRTUELLE

La boîte à outils de la réalité virtuelle est une solution pour voir et interagir avec les dynamiques des systèmes dans un environnement virtuellement réel à trois dimensions. Il étend les capacités de MATLAB et de Simulink dans un monde de réalité virtuelle graphique.

- **Virtual worlds** : Crée des mondes virtuels ou des scènes tridimensionnelles en utilisant le langage standard de la technologie ‘Virtual Reality Modeling Language’ (VRML).
- **Dynamics system** : — Crée et définit les systèmes dynamiques avec MATLAB et Simulink.
- **Animation** : Montre des scènes tridimensionnelles entraînées par des signaux à partir de l’environnement e Simulink.
- **Manipulation** : Change la position et les propriétés des f objets dans un monde virtuel, ou change les paramètres dans le modèle Simulink. Pour fournir un environnement de travail complet, le ‘Virtual Reality Toolbox’ comprend des composants additionnels telles que le joystick (Figure II.6) de commande:



Figure II.6

- Exemple de système de composant de manipulation dans un monde virtuel avec VR toolbox :

On ouvre la librairie de la boîte à outils de la Réalité Virtuelle qui est montrée sur la (figure II.7) suivante :

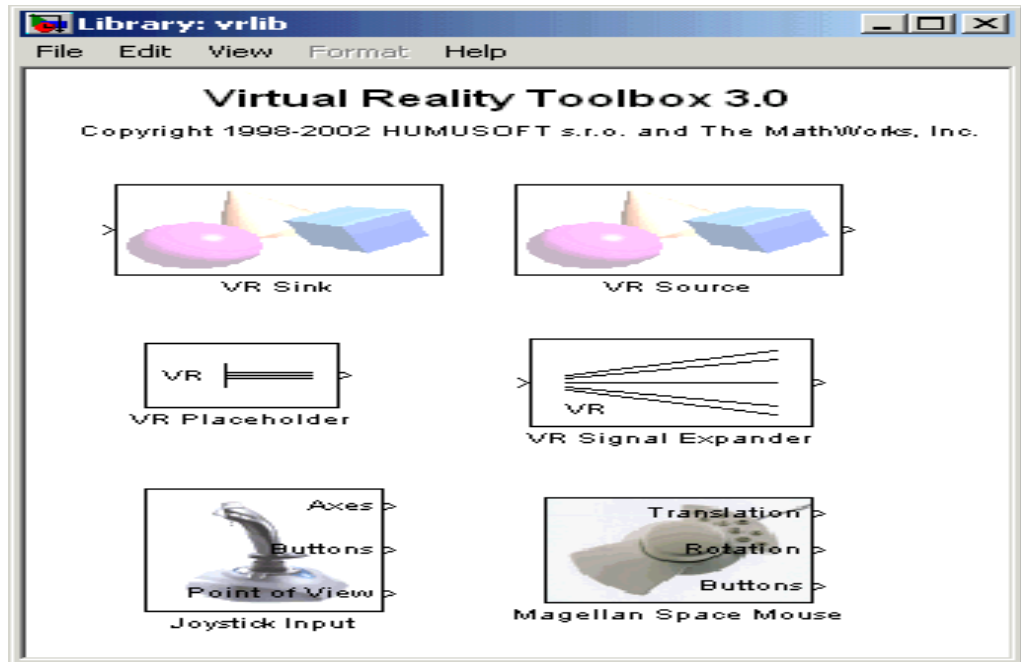


Figure II.7

A partir de la fenêtre de la librairie, on peut tirer et jeter, par exemple le « VR Sink » bloc dans le modèle de Simulink.

Exemple d'application :

Creation d'une liaison rotoide entre deux tiges comme montre la (Figure II.8).

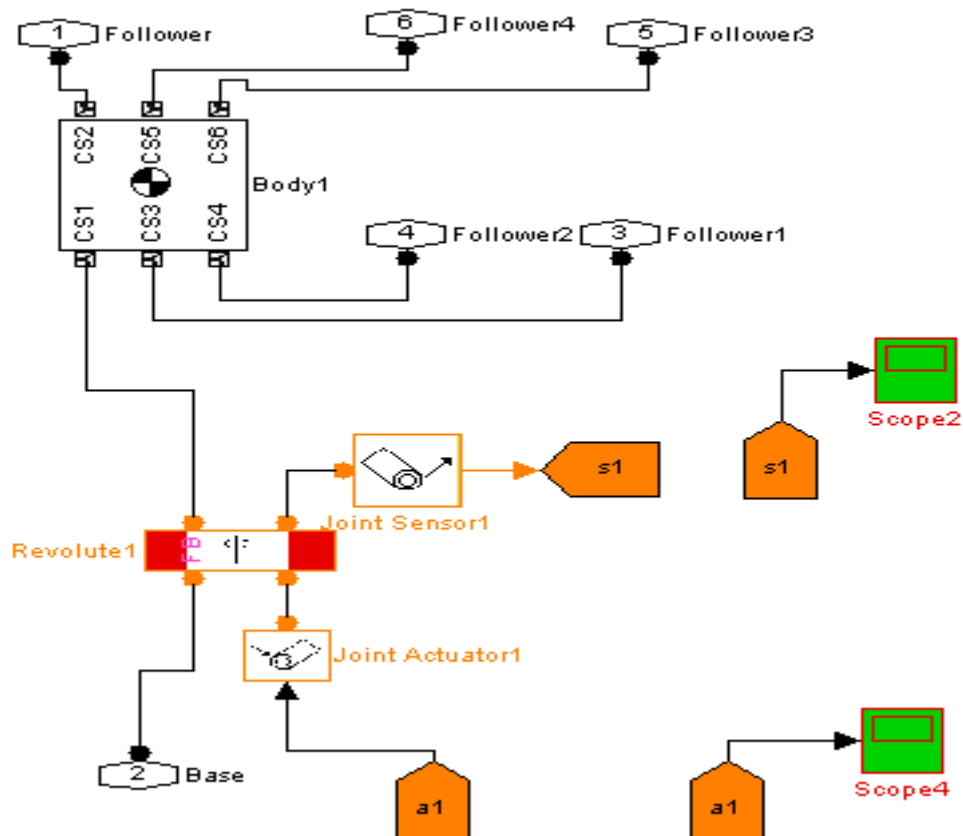


Figure (II.8): Creation d'une liaison rotoide dans MATLAB VR toolbox.

II.5 CONCLUSION

Dans ce chapitre j'ai décrit les deux boîtes à outils matlab/Simulink qui vont nous permettre de mener à terme la modélisation d'un robot bras manipulateur, la conception de commande et la visualité dans un monde virtuel l'animation du bras tout en ayant commande sur les paramètres du contrôleur lors de l'évolution de l'animation. Dans le prochain chapitre, les différentes techniques de commande qui peuvent être envisagées pour la commande du bras manipulateur dans monde virtuel sous l'environnement Matlab/Simulink sont discutées.

Chapitre III

Commande des Robots manipulateur

CHAPITRE III

COMMANDE DES ROBOTS MANIPULATEUR

III.1 Introduction

La plupart des tâches demandées à un manipulateur impliquent un contact avec l'environnement. Ceci implique de réguler l'effort exercé par le robot. Les commandes en effort constituent donc l'essentiel des publications relatives à la commande des manipulateurs ; Ceci est à prendre spécifiquement en compte lors du choix de la structure de commande pour piloter le manipulateur.

III.2 Commande des bras manipulateurs

Le problème de la commande d'un robot manipulateur peut être formulé comme la détermination de l'évolution des forces généralisées (forces ou couples) que les actionneurs doivent exercer pour garantir l'exécution de la tâche tout en satisfaisant certains critères de performance.

Différentes techniques sont utilisées pour la commande des bras manipulateurs. La conception mécanique du bras manipulateur a une influence sur le choix du schéma de commande. Un robot manipulateur est une structure mécanique complexe dont les inerties par rapport aux axes des articulations varient non seulement en fonction de la charge mais aussi en fonction de la configuration, des vitesses et des accélérations.

La plupart des robots utilisent des servomoteurs électriques comme actionneurs. Dans le cas de servomoteurs ayant de faibles rapports de réduction, ce sont les servomoteurs qui doivent compenser les effets des variations des forces d'inertie et de gravité. Dans le cas de servomoteurs avec de forts rapports de réduction, l'inertie vue par les moteurs varie beaucoup moins et il est alors possible de modéliser le robot par un système linéaire qui permet de découpler les articulations.

Dans le contexte de ce document nous considérons uniquement l'utilisation de servomoteurs avec de forts rapports de réduction comme actionneurs, ce qui produit des robots à articulations rigides. Le problème de la rigidité des articulations est évident lorsqu'on parle d'interaction avec l'environnement ou des collisions. Des imprécisions dans la modélisation de l'environnement peuvent se traduire par des efforts de contact importants qui peuvent endommager les mécanismes internes du robot ou son environnement.

En robotique lorsque l'articulation est rigide, les blessures qu'un robot peut produire sont fonction de l'inertie du mouvement. Pour réduire les dommages produits par une collision, il est recommandé d'utiliser des robots ayant « une élasticité », on parle alors de robots avec des

articulations flexibles. On peut considérer aussi des actionneurs élastiques, ou une action parallèle distribué.

Un chapitre du livre W. Khalil [31] est dédié à la commande de robots à articulations flexibles que nous recommandons au lecteur intéressé. Les travaux d'Albu-Schäffer et al utilisent la commande en couple pour chacune des joints pendant que les travaux de Luca et al, considèrent la détection des collisions et la façon de réagir après la collision.

Deux types de mouvements apparaissent quand on parle de commande du bras manipulateur. Un premier type considère que les mouvements nécessaires pour la réalisation de la tâche sont exécutés dans l'espace libre. Le deuxième type considère des mouvements spécifiques avec des forces de contact pour l'organe terminal qui se déplace dans un espace contraint. Toute tâche de robotique est réalisée par une combinaison de ces deux types de mouvement. Pour simplifier la commande, les deux types de mouvement sont abordés séparément.

III.3 Commande dans l'espace libre

Nous ne pouvons pas dans le cadre de ce mémoire traiter en détail l'ensemble des techniques de commande des bras manipulateurs. Pour une étude plus détaillée, le lecteur pourra se référer à [32], [33]. Nous présentons juste une vue générale de différentes techniques rapportées dans la littérature.

Dans les paragraphes suivants nous nous limiterons à une description simple des différentes techniques.

Nous présentons tout d'abord le schéma de commande par articulation pour progresser jusqu'à des techniques plus performantes comme la commande optimale. À la fin de cette section, nous décrivons notre approche pour la commande.

III.3.1 Commande par articulation

Cette technique est utilisée par des robots manipulateurs qui utilisent des servomoteurs avec de forts rapports de réduction. Lorsque le système présente un comportement linéaire, l'asservissement du mouvement peut être réalisé par des techniques classiques de commande. Nous parlons alors d'une commande décentralisée de type PID. Le schéma classique est amélioré avec des signaux d'anticipation pour corriger les effets de la force de gravité et de couplage.

III.3.2 Commande jacobéenne

Cette technique est utilisée depuis les travaux de Whitney et elle est appelée de cette façon lorsqu'elle utilise la matrice jacobéenne inverse du bras manipulateur pour calculer les vitesses de consigne aux articulations. Elle est aussi connue sous le nom de commande à mouvement résolu.

Les approches les plus courantes sont de type : mouvement à vitesse résolu, mouvement à accélération résolu et mouvement à force résolu. La technique de mouvement résolu commande la position de l'organe terminal du manipulateur dans l'espace cartésien, par combinaison des mouvements de plusieurs articulations.

Ce type de schémas peut se présenter sous la forme d'un schéma cinématique quand les vitesses calculées aux articulations sont utilisées directement comme des consignes pour les boucles de commande de chaque articulation, ou sous la forme d'un schéma dynamique quand le schéma utilise le modèle dynamique pour découpler les articulations.

Son principal inconvénient est l'utilisation de l'inverse de la matrice jacobéenne, qui peut devenir singulière dans certaines conditions.

III.3.3 Commande par découplage non linéaire

Cette technique est aussi connue sous les noms de commande dynamique ou commande par couple calculé. Lorsque l'application exige des évolutions rapides avec des contraintes dynamiques, la commande doit prendre en compte les forces d'interaction. Ce type de technique considère l'ensemble des articulations et, pour les découpler, utilise la théorie du découplage non linéaire. Cette théorie utilise le modèle dynamique du robot pour le calcul de la loi de commande, ce qui conduit à des lois de commande centralisées non linéaires. Des signaux d'anticipation peuvent être utilisés pour minimiser des effets non linéaires.

Ce type de technique permet la commande dans l'espace des articulations ou dans l'espace cartésien, avec l'avantage que les articulations sont découplées et peuvent évoluer à grandes vitesses avec de fortes inerties. Cette méthode dépend fortement du modèle du système, elle est très sensible aux imprécisions du modèle qui entraînent un découplage imparfait. Ceci constitue son principal inconvénient.

III.3.4 Commande fondée sur une fonction de Lyapunov

Des méthodes basées sur une fonction de Lyapunov ont été utilisées pour la commande des bras manipulateurs de façon satisfaisante pour des tâches de suivi. Particulièrement lorsqu'on cherche à garantir la convergence asymptotique et non à linéariser le système ou à obtenir le découplage [32].

III.3.5 Commande passive

Cette technique considère le robot comme un système passif, c'est à dire un système qui dissipe de l'énergie. De telles lois de commande permettent de modifier l'énergie naturelle du robot pour qu'il réalise la tâche. En utilisant le formalisme de Hamilton, la commande cherche à minimiser l'énergie du système en utilisant un bloc non linéaire passif dans la boucle de retour. La

commande passive tend à être plus robuste que le découplage non linéaire, lorsque la technique ne recherche pas à annuler les non linéarités.

III.3.6 Commande adaptative

Ce type de techniques vise à corriger les déficiences de la commande par découplage non linéaire,

Comme la connaissance approximative des paramètres du modèle du robot ou pour s'adapter aux différentes conditions opératoires. Ce type de schémas cherche à estimer ou ajuster en-ligne les valeurs des paramètres utilisées dans le calcul de la loi de commande. Un des travaux les plus intéressants sur ce sujet est la commande proposé par Slotine et al appelée commande de Slotine-Li ou commande adaptative passive. Plusieurs travaux sur la commande adaptative sont présentés dans [30].

Les avantages de ce type de techniques sont évidents, malheureusement la puissance de calcul demandée au système constitue un inconvénient important.

III.3.7 Commande prédictive

Ce type de schémas, en utilisant le modèle du système et les consignes, est capable de prédire son évolution, de telle manière qu'il est capable d'agir en fonction de l'erreur de prédiction. Trois schémas différents sont proposés : point final fixe, horizon fini et une combinaison des deux précédents. Un grand avantage de ce type d'approche est lié au fait que l'erreur de prédiction n'est pas contaminé par les bruits de mesure mais la dépendance au modèle reste forte.

III.3.8 Commande robuste

Dans le cas de paramètres fixes, il est connu que la technique de découplage non linéaire peut devenir instable en présence d'incertitudes. Si les paramètres du modèle ne sont pas connus de façon précise et si l'incertitude sur les paramètres admet des bornes connus, alors les techniques de commande robuste peuvent être utilisées. Par exemple, les travaux de Slotine [1], considèrent la technique de modes glissants appelés aussi commande de structure variable. Cette technique utilise une surface de glissement où la stabilité du système est garantie.

III.3.9 Commande optimale

Pour réaliser une tâche, il peut exister un grand nombre de solutions. Dans ce cas, il peut être souhaitable de choisir une solution qui satisfasse un certain critère.

La littérature présente différents types de critères pour la commande optimale : la commande en temps minimal, du domaine des neurosciences la minimisation du jerk pour maximiser la souplesse du mouvement, entre autres.

III.4 Commande dans l'espace contraint

Lorsque l'organe terminal entre en contact avec une surface, la chaîne cinématique du robot est fermée sur l'environnement, et du fait de la raideur de l'ensemble, de faibles variations de position du point de contact peuvent induire des efforts importants. Lorsque le robot est contraint par l'environnement suivant toutes les directions, aucun déplacement de l'organe terminal n'est possible et il ne peut qu'exercer des efforts sur l'environnement; en revanche, lorsque le robot est dans l'espace libre, l'absence de tout contact empêche la génération d'efforts. Ainsi, les notions de commande en position et de commande en effort s'excluent mutuellement : on ne peut contrôler simultanément une force et un déplacement suivant une même direction.

Dans la pratique, on associe un repère de compliance à l'organe terminal, de façon à ce que pour chacun des 6 degrés de liberté on associe soit une « position » soit un « effort ».

La littérature propose cinq schémas de commande, les trois premiers n'utilisent pas de consigne d'effort : compliance passive, commande par raideur active et commande en impédance. Les deux suivantes utilisent la consigne d'effort : la commande hybride et la commande hybride externe.

Plus généralement, la fonction d'un robot manipulateur est de situer dans l'espace son organe terminal OT. La Situation de l'organe terminal est définie par la position et l'orientation d'un repère lié à son but. Une tâche simple peut être définie comme le mouvement libre réalisé par le robot manipulateur pour aller d'une situation initiale à une situation finale. Une tâche complexe peut être définie de la même façon, en précisant que le robot doit suivre un chemin prédéfini tout en respectant des contraintes cinématiques liées au mouvement. Dans les deux cas, le mouvement que le robot réalise devra être défini par des consignes qui correspondent à des trajectoires temporelles.

III.5 Généralités sur la sélection de la loi de commande

Tout d'abord, un robot manipulateur est une machine constituée par :

- une structure mécanique de support.
- des actionneurs.
- des capteurs.
- un système de commande.

Donc, nous pouvons définir un robot manipulateur comme une structure mécanique complexe dont les caractéristiques dynamiques varient non seulement en fonction de la charge mais aussi en fonction de la configuration, des vitesses et des accélérations.

Dans ce travail, nous n'abordons pas la construction d'un robot manipulateur. Nos travaux portent sur l'utilisation d'un bras manipulateur à deux degrés de liberté du type 2R.

La plupart des robots utilisent des servomoteurs électriques comme actionneurs. Les caractéristiques des servomoteurs ont un rôle important pour la sélection du système de commande. Dans le cas où le robot utilise des servomoteurs avec de faibles rapports de réduction, les boucles de commande doivent compenser les effets des variations des forces d'inertie et de gravité. Les lois de commande basées sur les modèles dynamiques des robots donnent de très bons résultats dans ce cas. Ils permettent de maintenir la réponse dynamique du système dans certains critères de performance. En utilisant ce type de techniques les robots peuvent évoluer à grandes vitesses. L'inconvénient est lié à la forte dépendance au modèle dynamique. En opposition, quand les articulations sont actionnées par l'intermédiaire de réducteurs à forts rapports de réduction, l'inertie vue par les moteurs varie peu. Dans ce cas, les asservissements peuvent être assurés axe par axe par des boucles de commande classiques.

Pour le robot manipulateurs de type 2R, les capteurs les plus couramment utilisés sont les capteurs de position et de vitesse angulaire. Nous n'utilisons que des capteurs de position absolue qui ne demandent pas de processus d'initialisation.

III.6 régulation classique de type P.D

Une technique de régulation classique, à savoir un contrôleur de type P.D comme montre la (Figure III.1). Dont l'implémentation est relativement simple, mais cette commande n'est pas robuste puisqu'elle est sensible aux variations de la charge.

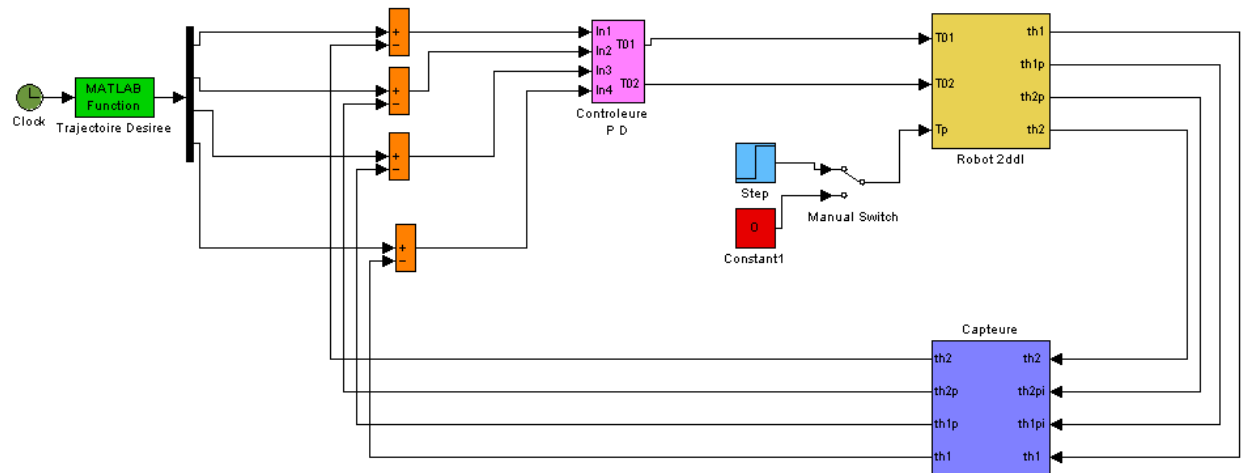


Figure (III.1) : Commande d'un robot manipulateur avec un régulateur PD
III.6.1 Commande articulaire par un régulateur du type P.D

La première commande articulaire est issue des commandes appliquées sur les bras manipulateurs rigides elle nécessite un régulateur du type P.D ayant des gains assez élevé et elle dépend de la trajectoire articulaire choisie, elle n'implémente pas une des caractéristique des robots qui est le faible rapport poids-charge.

Cette commande est sensible aux variations de la charge parce que les oscillations dans les positions articulaires et les positions de l'organe effecteur augmentent avec l'augmentation de la charge, alors elle est moins robuste.

Voici les graphes de th_1 , th_2 qui illustre la sensibilité aux variations de la charge après $t = 1.5s$ dans les (figures III.2, 3) respectivement (avec perturbation).

On voit bien qu'après $t = 1.5 s$, l'écart entre les thétas désiré (th_d) et les thétas obtenue augmentent proportionnellement avec la charge (ou perturbation).

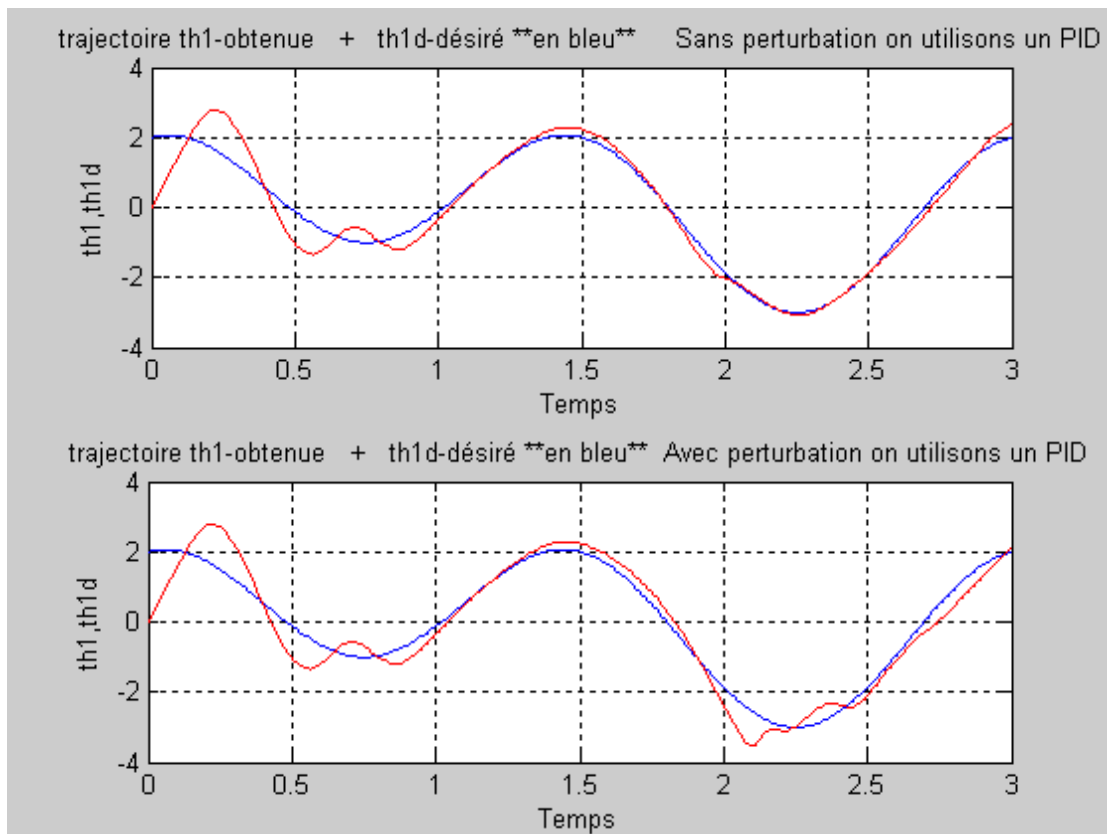


Figure (III.2) : allures de th_1 , th_{1d} sans et avec perturbation en utilisons un Régulateur PD

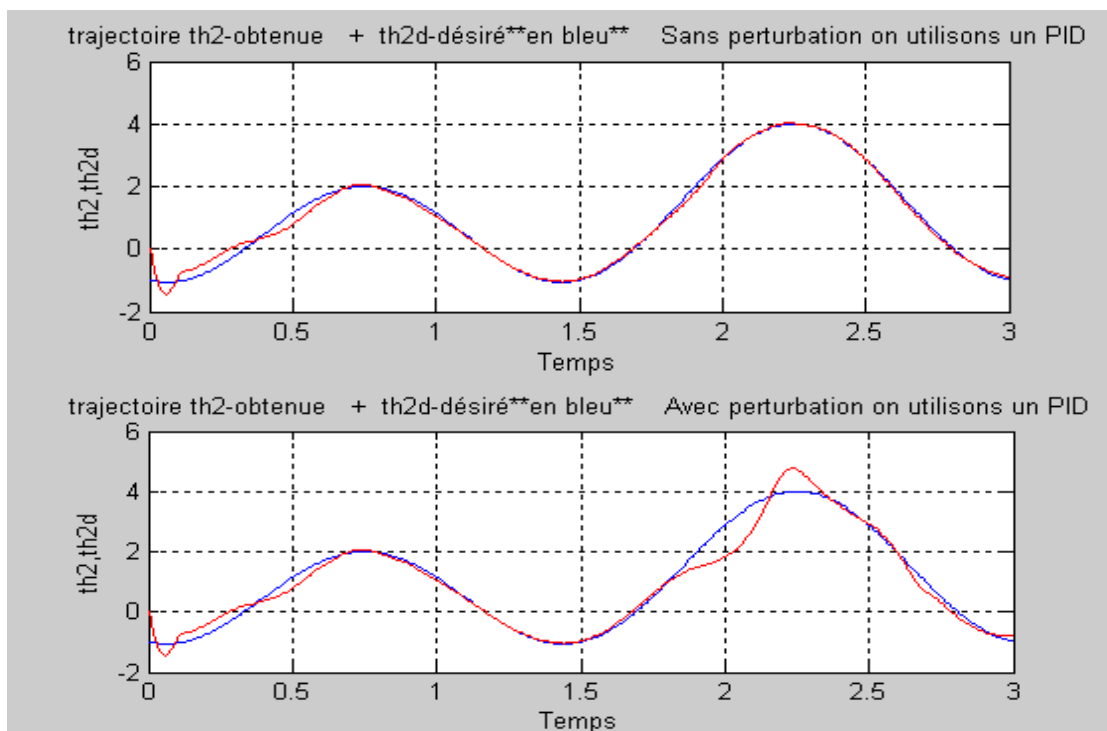


Figure (III.3) : allures de th_2 , th_{2d} sans et avec perturbation en utilisons un régulateur PD

III.7 Conclusion

Cette commande est sensible aux variations de la charge alors elle est moins robuste, pour cela une deuxième commande intelligente basée sur les réseaux de neurones est appliquée dans le prochain chapitre afin de modifier les trajectoires articulaires de références du bras manipulateur.

Chapitre IV

Commande des Robots manipulateur avec Les réseaux de neurones artificiel (compliance naturel)

CHAPITRE IV

COMMANDE DES ROBOTS MANIPULATEURS AVEC LES RÉSEAUX DE NEURONES ARTIFICIELS (COMPLIANCE NATUREL)

IV.1 Introduction

Malgré la constante augmentation de puissance des calculateurs, malgré les approches théoriques de plus en plus sophistiquées, un certain nombre de tâches résistent encore aux algorithmes et aux méthodes classiques de traitement des signaux et des données. Ces tâches relèvent typiquement du traitement, en temps réel, de très grands flots de données souvent multidimensionnelles et arrivant à des cadences élevées. Le grand nombre des données, leur variabilité, le fait qu'elles ne répondent pas à des modèles physiques connus nous laissent souvent démunis devant des tâches de caractérisation, de reconnaissance et de prise de décision.

Il y a des centaines d'exemples nous montrons à la fois combien on peut espérer de la modélisation du système nerveux mais aussi combien il sera difficile d'imaginer et de comprendre les divers aspects des problèmes de perception. Il paraît donc naturel d'essayer de comprendre comment les systèmes biologiques sont capables de telles performances, et si possible, de s'inspirer de leurs principes pour imaginer de nouveaux algorithmes ou de nouvelles machines plus efficaces que ceux dont nous disposons actuellement. Les techniques de réseaux de neurones relèvent d'une telle approche : comprendre les principes selon lesquels les systèmes biologiques traitent l'information et s'en inspirer pour élaborer de nouvelles techniques en sciences de l'ingénieur. C'est donc une double démarche, à la fois cognitive et synthétique où le monde biologique doit être considéré comme une source de référence et de connaissance.

D'un point de vue technique, il est clair que seuls les principes seront importants. Il ne sera généralement pas nécessaire, pour modéliser telle ou telle fonction, de simuler toutes les molécules chimiques et les enzymes qu'elle implique, l'adéquation fine aux modèles biologiques ne sera retenue que dans la mesure où elle conduit à des réalisations techniquement économiques.

Les caractéristiques essentielles des réseaux de neurones réels que nous conserverons dans les modèles mathématiques étudiés, concernent le grand nombre de connexions, la non-linéarité des relations entrée-sortie et la faculté de "plasticité" ou d'adaptabilité. Ces caractéristiques, même simplifiées, leur confèrent déjà de multiples possibilités en traitement des signaux et des

informations ainsi que la faculté d'apprendre à classer, à reconnaître des formes ou à réaliser des tâches complexes.

IV .2 Neurone biologique

On pense que le système nerveux compte plus de 1000 milliards de neurones interconnectés. Bien que les neurones ne soient pas tous identiques, leur forme et certaines caractéristiques permettent de les répartir en quelques grandes classes. En effet, il est aussi important de savoir, que les neurones n'ont pas tous un comportement similaire en fonction de leur position dans le cerveau.

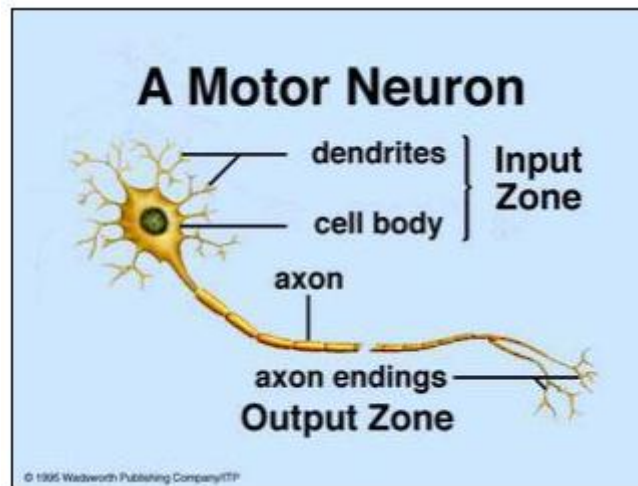


Figure (IV.1) : Schéma d'un neurone biologique

On peut le décomposer en trois régions principales:

IV.2.1 Le corps cellulaire

Il contient le noyau du neurone ainsi que la machine biochimique nécessaire à la synthèse des enzymes. Ce corps cellulaire de forme sphérique ou pyramidale contient aussi les autres molécules essentielles à la vie de la cellule. Sa taille est de quelques microns de diamètre.

IV.2.2 Les dendrites

Ce sont de fines extensions tubulaires qui se ramifient autour du neurone et forment une sorte de vaste arborescence. Les signaux envoyés au neurone sont captés par les dendrites. Leur taille est de quelques dizaines de microns de longueur.

IV.2.3 L'axone

C'est le long de l'axone que les signaux partent du neurone. Contrairement aux dendrites qui se ramifient autour du neurone, l'axone est plus long et se ramifie à son extrémité ou il se connecte aux dendrites des autres neurones. Sa taille peut varier entre quelques millimètres à plusieurs mètres.

IV.2.4 Synapse

Une synapse est une jonction entre deux neurones, et généralement entre l'axone d'un neurone et une dendrite d'un autre neurone (mais il existe aussi des synapses axo-axonales par exemple).

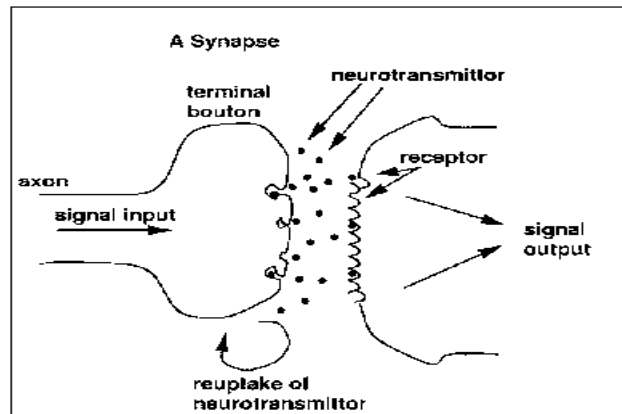


Figure (IV.2) : Schéma d'une synapse

IV.2.5 Fonctionnement

Au point de vue fonctionnel, il faut considérer le neurone comme une entité polarisée, c'est-à-dire que l'information ne se transmet que dans un seul sens : des dendrites vers l'axone.

Le neurone va donc recevoir des informations, venant d'autres neurones, grâce à ses dendrites. Il va ensuite y avoir sommation, au niveau du corps cellulaire, de toutes ces informations et via un potentiel d'action (un signal électrique) le résultat de l'analyse va transiter le long de l'axone jusqu'aux terminaisons synaptiques. A cet endroit, lors de l'arrivée du signal, des vésicules synaptiques vont venir fusionner avec la membrane cellulaire, ce qui va permettre la libération des neurotransmetteurs (médiateurs chimiques) dans la fente synaptique. Le signal électrique ne pouvant pas passer la synapse (dans le cas d'une synapse chimique), les neurotransmetteurs permettent donc le passage des informations, d'un neurone à un autre.

Les neurotransmetteurs excitent (neurotransmetteurs excitateurs) ou inhibent (neurotransmetteurs inhibiteurs) le neurone suivant et peuvent ainsi générer ou interdire la propagation d'un nouvel influx nerveux.

En effet, au niveau post-synaptique, sur la membrane dendritique, se trouvent des récepteurs pour les neurotransmetteurs. Suivant le type de neurotransmetteur et le type des récepteurs, l'excitabilité du neurone suivant va augmenter ou diminuer, ce qui fera se propager ou non l'information.

Les synapses possèdent une sorte de «mémoire» qui leur permet d'ajuster leur fonctionnement. En fonction de leur «histoire», c'est-à-dire de leur activation répétée ou non entre deux neurones, les connexions synaptiques vont donc se modifier.

Ainsi, la synapse va faciliter ou non le passage des influx nerveux. Cette plasticité est à l'origine des mécanismes d'apprentissage.

IV.3 Représentation d'un réseau de neurone artificiel

Un réseau artificiel consiste en une mise en commun d'unités du traitement simples qui communiquent en envoyant des signaux entre eux sur un grand nombre de connexions chargées d'un poids.

Un réseau artificiel contient les éléments suivant :

- un ensemble d'unités de traitement (neurones, cellules);
- un état y_k de la mise en activité pour chaque unité, équivalent à la sortie de l'unité;
- connexions entre les unités. Généralement chaque connexion est définie par un w_{jk} du poids qui détermine l'effet de signal d'unité j sur unité k ;
- une règle de la propagation qui détermine le s_k de l'entrée effective d'une unité de ses entrées externes;
- une fonction de la mise en activité F_k qui détermine le nouveau niveau de mise en activité basé sur l'entrée effective $s_k(t)$ et la mise en activité courante $y_k(t)$
- une entrée externe (alias biais, décalage) θ_k pour chaque unité;
- une méthode pour rassemblement de l'information (la règle d'apprentissage);
- un environnement dans lequel le système doit opérer, en fournissant des signaux d'entrée et (si nécessaire) signaux d'erreur.

IV.3.1 Unités du traitement

Chaque unité exécute un travail relativement simple: recevez l'entrée de voisins ou source externes et utilisez ceci pour calculer un signal de la sortie qui est propagé à d'autres unités. À part ce traitement, une deuxième tâche est l'ajustement des poids. Le système est fondamentalement placé parallèlement dans le sens que beaucoup d'unités peuvent emporter leurs calculs en même temps.

Dans les systèmes neuronaux c'est utile à distinguer trois types d'unités: unité d'entrée (indiqué par un index i) qui reçoit des données de l'extérieur du réseau neuronal, unités de sortie (indiqué par un index o) qui envoie des données en dehors de réseau neuronal, et unités cachées (indiqué par un index h) dont sont entrées et les signaux de la sortie restent à l'intérieur de réseau neuronal.

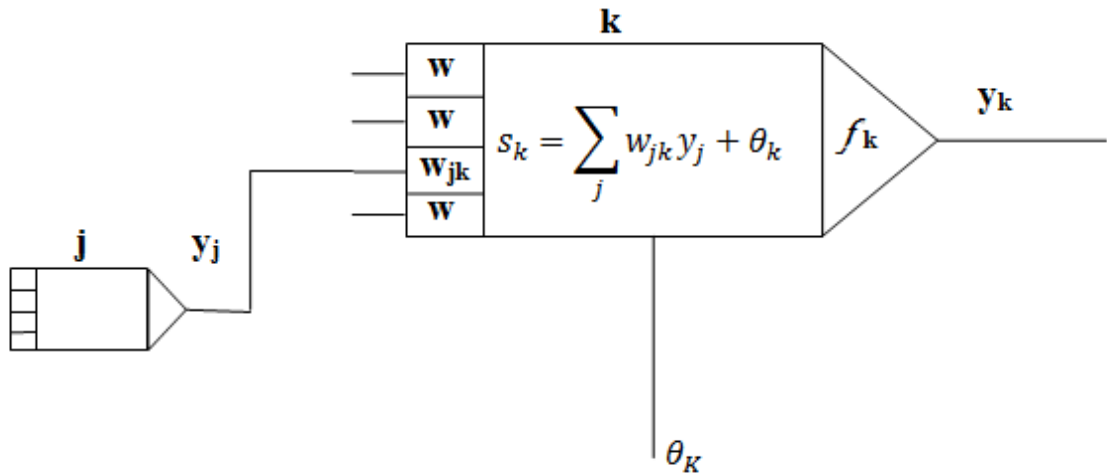


Figure (IV. 3) : Les objets de base d'un réseau neuronal. La règle de la propagation utilisée ici est le 'norme' addition chargée d'un poids.

Pendant le fonctionnement, les unités peuvent être actualisées soit de façon synchrone ou de façon asynchrone. Avec mise à jour synchrone, toutes les unités actualisent leur mise en activité en simultanéité; avec mise à jour asynchrone, chaque unité a une probabilité d'actualiser à la fois sa mise en activité, et habituellement seulement une unité sera capable de faire à la fois ceci. Dans quelques cas le modèle dernier a des avantages.

IV .3.2 Connexions entre les unités

Dans la plupart des cas nous supposons que chaque unité fournit une contribution additive à l'entrée de l'unité avec laquelle il est connecté. L'entrée totale à unité k est simplement la somme pondérée des sorties séparées de chacun des unités suivies d'un biais ou terme du décalage θ_k :

$$s_k(t) = \sum_j w_{jk}(t) y_j(t) + \theta_k \quad (\text{IV .1})$$

La contribution pour le w_{jk} positif est considérée comme une excitation et pour w_{jk} négatif comme inhibition. Dans quelques cas les règles plus complexes pour mixer des entrées sont utilisées dans lequel une distinction est faite entre entrées excitatrices et inhibitrices. Nous appelons des unités avec une règle de la propagation unités du sigma.

Une règle de la propagation différente, introduite par Feldman et Ballard (Feldman & Ballard, 1982), est connue comme la règle de la propagation pour l'unité du sigma-pi:

$$s_k(t) = \sum_j w_{jk}(t) \prod_m y_{jm}(t) + \theta_k \quad (\text{IV .2})$$

IV.3.3 Mise en activité et règles de la sortie :

Nous avons besoin aussi d'une règle qui donne l'effet de l'entrée totale sur la mise en activité de l'unité. Nous avons besoin d'une fonction F_k qui prend l'entrée totale $s_k(t)$ et la mise en activité courante $y_k(t)$ et produit une nouvelle valeur de la mise en activité de l'unité k :

$$y(k+1) = f_k(y_k(t), s_k(t)) \quad (\text{IV .3})$$

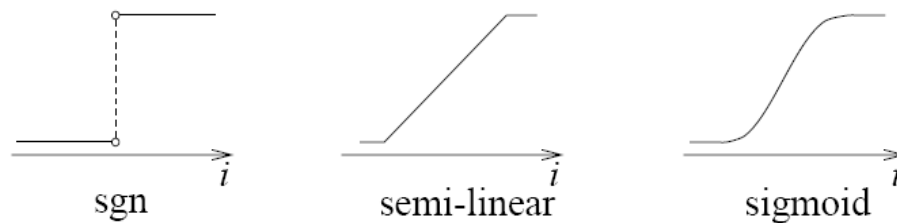
Souvent, la fonction de la mise en activité est une fonction du nondecreasing de l'entrée totale de l'unité:

$$y(k+1) = f_k(s_k(t)) = f_k(\sum_j w_{jk}(t)y_j(t) + \theta_k) \quad (\text{IV .4})$$

Bien que les fonctions de la mise en activité ne soient pas restreintes aux fonctions du nondecreasing. Généralement, quelque sorte de fonction de seuil est utilisée: (un sgn), ou une fonction linéaire ou semi-linéaire.

$$y_k = f(s_k) = \frac{1}{1+e^{-s_k}} \quad (\text{IV .5})$$

Dans quelques applications une tangente hyperbolique est utilisée. Les valeurs de la sortie donnée dans la plage $[-1, +1]$.



Dans quelques cas, la sortie d'une unité peut être une fonction stochastique de l'entrée totale de l'unité. Dans ce cas la mise en activité n'est pas déterminée de façon déterministique par l'entrée du neurone, mais l'entrée du neurone détermine la probabilité p qu'un neurone obtient à une haute valeur de la mise en activité:

$$P(y_k \leftarrow 1) = \frac{1}{1+e^{-s_k/T}} \quad (\text{IV .6})$$

IV.3.3 Réseaux avec les fonctions de la mise en activité du seuil

Un réseau feed-forward à une seule couche se compose d'un ou plusieurs neurones o de sortie, qui est connecté avec un w_{i0} de facteur de pondération à toutes les entrées i . Dans le cas le plus simple le réseau a seulement deux entrées et une sortie unique. L'entrée du neurone est la somme pondérée des entrées plus le terme du biais.

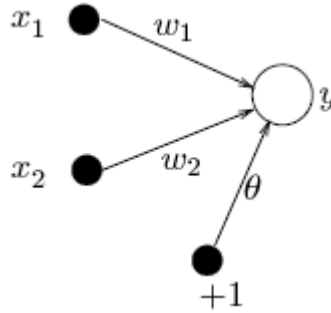


Figure (IV. 4) : Réseau à une seule couche avec une sortie et deux entrées.

Le réseau est formé par la mise en activité du neurone de la sortie.

$$y = F(\sum_{i=1}^2 w_i x_i + \theta) \tag{IV .7}$$

Avec :

$$F(s) = \begin{cases} 1 & \text{If } s > 0 \\ -1 & \text{Otherwise} \end{cases} \tag{IV .8}$$

La sortie du réseau donc est +1 ou -1, selon l'entrée Le réseau utilisé pour une tâche de la classification: il peut décider si un modèle de l'entrée appartient à une de deux classes. Si l'entrée totale est positive, le modèle sera affecté pour en classer +1, si l'entrée totale est négative, l'échantillon sera affecté pour classer -1. La séparation entre les deux classes dans ce cas est une ligne droite, donnée par l'équation:

$$w_1 x_1 + w_2 x_2 + \theta = 0 \tag{IV .9}$$

$$x_2 = -\frac{w_1}{w_2} x_1 - \frac{\theta}{w_2} \tag{IV .10}$$

Le réseau à une seule couche représente une fonction discriminante linéaire.

Une représentation géométrique du réseau neuronal de seuil linéaire est donnée dans la (Figure IV. 5).

Nous voyons que les poids déterminent la pente de la ligne et le biais détermine le 'décalage.' à quelle distance la ligne est de l'origine. Les poids peuvent être tracés dans l'espace de l'entrée: le vecteur du poids est toujours perpendiculaire à la fonction du discriminant.

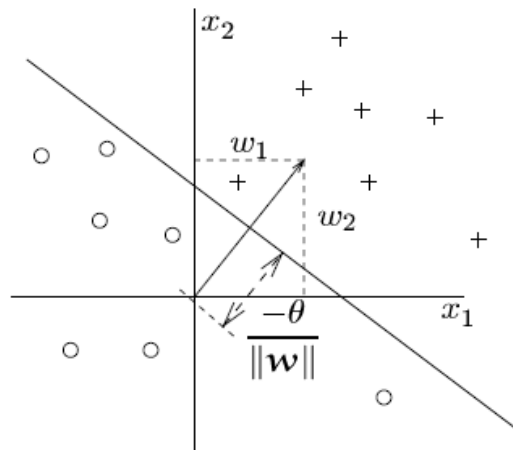


Figure (IV. 5) : Représentation géométrique de la fonction du discriminant et les poids

Maintenant que nous avons montré la puissance du réseau à une seule couche avec les unités du linearthreshold, nous venons à la deuxième question: comment apprenons-nous les poids et les biais dans le réseau? Nous décrirons deux méthodes apprentissage pour ces types de réseaux: le 'perceptron' la règle apprentissage et la règle de 'delta' ou 'LMS'. Les deux méthodes sont des procédures itératives qui ajustent les poids. Un échantillon d'apprentissage est présenté au réseau. Pour chaque poids la nouvelle valeur est calculée en ajoutant une correction à l'ancienne valeur. Le seuil est actualisé de la même façon:

$$w_i(t+1) = w_i(t) + \Delta w_i(t) \quad (\text{IV .11})$$

$$\theta(t+1) = \theta(t) + \Delta\theta(t)$$

Le problème apprentissage peut être formulé maintenant comme: comment calculons-nous le $\Delta w_i(t)$ et le $\Delta\theta(t)$ afin de classifier les modèles d'apprentissage correctement?

IV.3.5 Règle d'apprentissage

Suppose nous avons un ensemble d'échantillons d'apprentissage qui consistent en un vecteur d'entrée x et d'un résultat désiré $d(x)$. Pour une tâche de la classification le $d(x)$ est habituellement $+1$ ou -1 . La règle apprentissage est très simple et peut être déclarée comme suit:

1. Commencer avec les poids aléatoires pour les connexions;
2. Sélectionne un vecteur pour l'entrée x à partir de l'ensemble d'échantillons d'apprentissage;
3. Si $y \neq d(x)$ (le perceptron donne une réponse inexacte), modifie tous les w_i des connexions selon: $\Delta w_i(t) = d(x) x_i$;
4. Reviens à 2.

Note que la procédure est très semblable à la règle Hebb; la seule différence est que, quand le réseau répond correctement, aucun poids de la connexion n'est modifié. Nous devons modifier aussi le seuil θ . Ce θ est considéré comme un w_0 de la connexion entre le neurone de la sortie et l'unité du prédicat sur laquelle est toujours: $x_0 = 1$. Donnée le perceptron qui apprend la règle comme déclaré au-dessus, ce seuil est modifié d'après:

$$\Delta\theta = \begin{cases} 0 & \text{If the perceptron responds correctly} \\ d(x) & \text{Otherwise} \end{cases} \quad (\text{IV .12})$$

IV.4 La commande neuronale d'un bras manipulateur à deux degré de liberté

L'objectif de ce travail est la commande d'un bras manipulateur on utilise les réseaux de neurone. L'étude est divisée en deux parties, partie où les perturbations sont négligeables et l'autre où les perturbations sont prises en compte. Le robot avec son contrôleur est simulé à l'aide du logiciel **MATLAB/SIMULINK**.

IV.4.1 Le robot manipulateur

Le schéma du robot est le suivant :

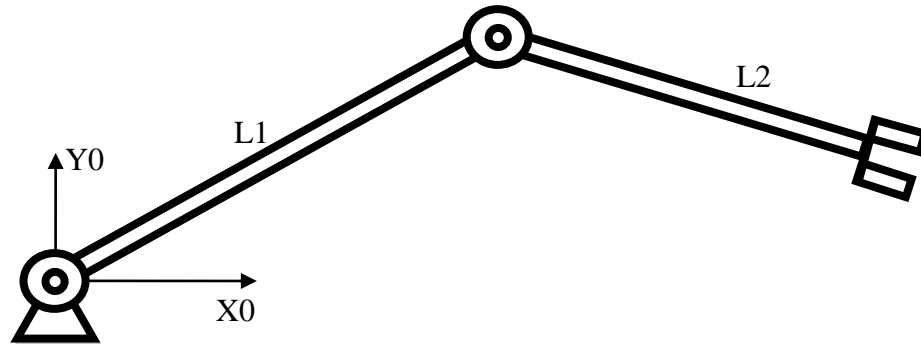


Figure (IV. 7) : schéma du robot à deux degré de liberté

IV.4.2 Paramètres du robot

Le manipulateur est celui présenté à la (figure IV. 7). Le tableau suivant récapitule ses différents paramètres de longueur et de masse indispensables aux algorithmes de contrôle.

$a_1 \dots a_5$: sont des paramètres constants obtenus à partir des masses (m_1 , m_2) et de longueurs (L_1 , L_2) des deux bras.

V_1 , V_3 : sont les coefficients des frottements visqueux.

V_2 , V_4 : sont les coefficients des frottements secs.

Paramètre	Valeur numérique
m_1 [Kg]	15.91
m_2 [Kg]	11.36
L_1 (m)	0.432
L_2 (m)	0.432
a_1	3.82
a_2	2.12
a_3	0.71
a_4	81.82
a_5	24.06
V_1 , V_3 (Nt.m.s/rad)	1.0
V_2 . V_4 (Nt.m)	0.5

IV.4.3 Equations générales du robot

Le modèle dynamique d'un tel mécanisme est habituellement décrit par l'équation matricielle suivante :

$$\mathbf{T} = \mathbf{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}} + \mathbf{G}(\boldsymbol{\theta}) + \mathbf{T}_p \quad (\text{IV .13})$$

Où :

- \mathbf{T} (nx1) : vecteur des couples généralisés. $\boldsymbol{\theta}$ (nx1) : vecteur des variables articulaires généralisées.
- $\dot{\boldsymbol{\theta}}$ (nx1) : vecteur des vitesses angulaires.
- $\ddot{\boldsymbol{\theta}}$ (nx1) : vecteur des accélérations.
- $\mathbf{M}(\boldsymbol{\theta})$ (nxn) : matrice d'inertie symétrique définie positive.
- $\mathbf{C}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$: matrice de dimension (nxn) spécifiant l'effet de centrifuge et de Coriolis.
- $\mathbf{G}(\boldsymbol{\theta})$: vecteur de dimension (nx1) exprimant l'effet gravitationnel.
- \mathbf{T}_p (nx1) : vecteur regroupant les perturbations.
- n : degré de liberté (nombre d'articulations).

IV.4.4 Les positions désirées

Les erreurs de position des deux articulations seront calculées s au cours d'un mouvement du bras en poursuite d'une trajectoire assurant une continuité en position, vitesse et accélération.

Les positions désirées sont donnée par :

$$\theta_{d1}(t) = 2 \cdot \cos(4\pi t/3) + \sin(2\pi t/3) ; 0 \leq t \leq 3. \quad (\text{IV .14})$$

$$\theta_{d2}(t) = 1 - 2 \cdot \cos(4\pi t/3) - \sin(2\pi t/3) ; 0 \leq t \leq 3. \quad (\text{IV .15})$$

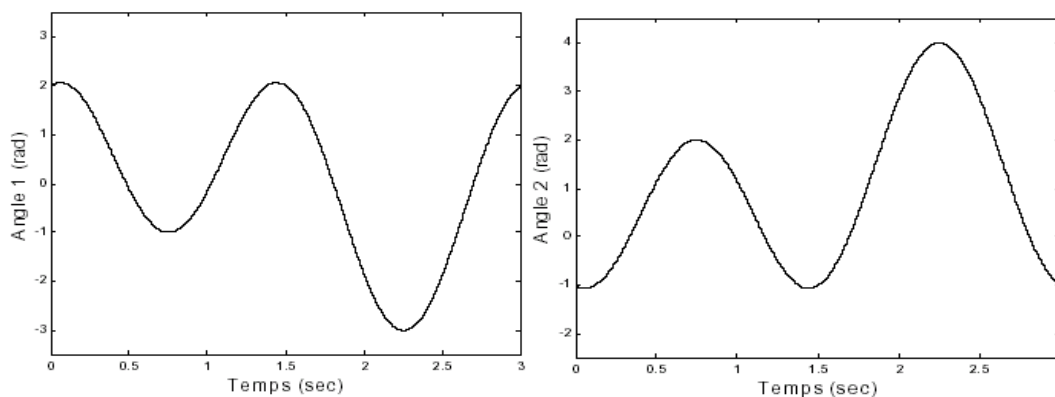


Figure (IV. 8) : Trajectoires désirées des deux articulations

Le modèle dynamique de ce robot avec la charge m (sur l'OT) est donné par l'équation matricielle suivante :

$$\mathbf{T} = \mathbf{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{N}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{G}(\boldsymbol{\theta}) + \mathbf{H}(\dot{\boldsymbol{\theta}}) + \mathbf{m}\mathbf{J}^T(\mathbf{J}\ddot{\boldsymbol{\theta}} + \dot{\mathbf{J}}\dot{\boldsymbol{\theta}}) \quad (\text{IV.16})$$

"Jacobien "

$$J := \begin{bmatrix} -L2 \sin(q1(t) + q2(t)) - L1 \sin(q1(t)) & -L2 \sin(q1(t) + q2(t)) \\ L2 \cos(q1(t) + q2(t)) + L1 \cos(q1(t)) & L2 \cos(q1(t) + q2(t)) \\ 0 & 0 \end{bmatrix}$$

"***** Calcul la vitesse angulaire *****"

"La Matrice antisymétrique "

$$AsOmeg_{0,2} := \begin{bmatrix} 0 & -\left(\frac{d}{dt} q2(t)\right) - \left(\frac{d}{dt} q1(t)\right) & 0 \\ \left(\frac{d}{dt} q2(t)\right) + \left(\frac{d}{dt} q1(t)\right) & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Donc le Jacobien est donnée comme suit :

$$J := \begin{bmatrix} -L2 \sin(q1(t) + q2(t)) - L1 \sin(q1(t)) & -L2 \sin(q1(t) + q2(t)) \\ L2 \cos(q1(t) + q2(t)) + L1 \cos(q1(t)) & L2 \cos(q1(t) + q2(t)) \\ 0 & 0 \end{bmatrix}$$

IV.4.5 Les Matrices $\mathbf{M}(\boldsymbol{\theta})$, $\mathbf{N}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})$, $\mathbf{G}(\boldsymbol{\theta})$ et $\mathbf{H}(\dot{\boldsymbol{\theta}})$

Les matrices M et N sont données comme suit :

$$M(\boldsymbol{\theta}) = \begin{bmatrix} a1 + a2 \cos \theta_2 & a3 + \frac{a2}{2} \cos \theta_2 \\ a3 + \frac{a2}{2} \cos \theta_2 & a3 \end{bmatrix}$$

$$N(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) = C(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}})\dot{\boldsymbol{\theta}} = \begin{bmatrix} -(a2 \sin \theta_2) \left(\dot{\theta}_1 \dot{\theta}_2 + \frac{\dot{\theta}_2^2}{2} \right) \\ (a2 \sin \theta_2) \frac{\dot{\theta}_1}{2} \end{bmatrix}$$

$$G(\boldsymbol{\theta}) = \begin{pmatrix} a4 \cos \theta_1 + a5 \cos(\theta_1 + \theta_2) \\ a5 \cos(\theta_1 + \theta_2) \end{pmatrix} \quad H(\dot{\boldsymbol{\theta}}) = \begin{pmatrix} V_1 \dot{\theta}_1 + V_2 \sin(\dot{\theta}_1) \\ V_3 \dot{\theta}_2 + V_4 \sin(\dot{\theta}_2) \end{pmatrix}$$

IV.5 La commande neuronale pour T_p négligeables

On simulera t le comportement du robot en poursuite de la trajectoire désirée dans le cas où les perturbations sont négligeables. Bien que les erreurs en position pour la commande sont très faibles, elles sont bien moins importantes en introduisant les termes de correction, et l'amélioration devient évidente.

L'équation dynamique de robot est :

$$\mathbf{T} = \mathbf{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{N}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{G}(\boldsymbol{\theta}) \tag{IV .17}$$

Selon cette équation on constate qu'on a comme entrer le couple T et comme sortie l'angle θ , donc pour pouvoir simuler le robot il faut trouver l'angle θ en fonction de couple T

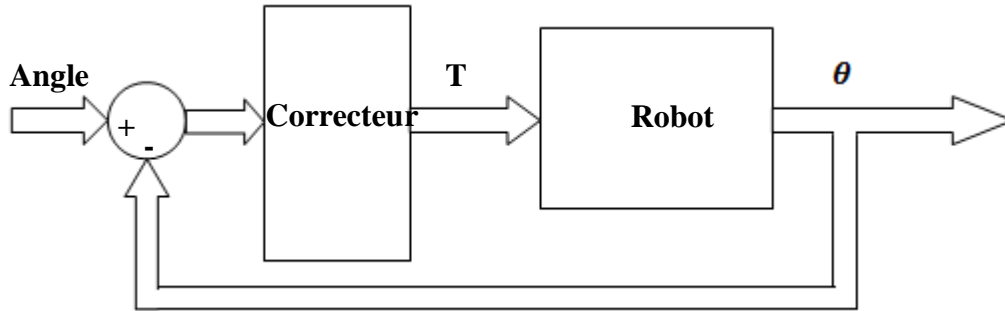
$$\ddot{\boldsymbol{\theta}} = \mathbf{f}^{-1}(\mathbf{T}) \tag{IV .18}$$


Figure (IV. 9). Schéma synoptique de la commande

a. Calcul l'inverse de l'équation dynamique ($\ddot{\boldsymbol{\theta}} = \mathbf{f}^{-1}(\mathbf{T})$)

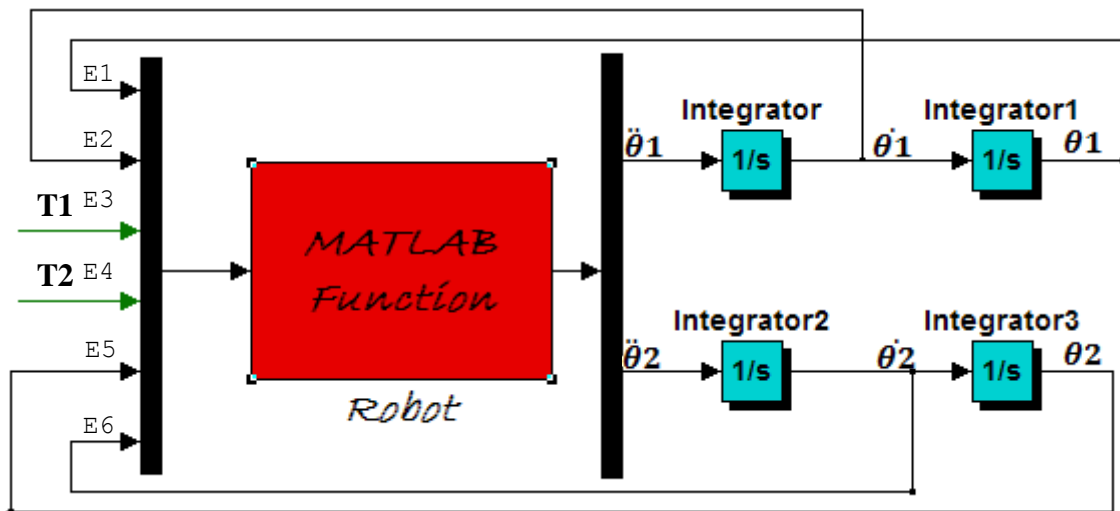


Figure (IV. 10)

On constate selon cette figure qu'on peut obtenir $\dot{\boldsymbol{\theta}}$ et $\boldsymbol{\theta}$ à partir de $\ddot{\boldsymbol{\theta}}$

Entrée(1)= θ_1

Entrée(2)= $\dot{\theta}_1$

Entrée(3)= T1

Entrée(4)= T2

Entrée(5)= θ_2

$$\text{Entrée}(6) = \dot{\theta}_2$$

➤ **Matrice d'inertie**

$$m_{11} = a_1 + a_2 \cos(\theta_2)$$

$$m_{12} = a_3 + (a_2/2) \cos(\theta_2)$$

$$m_{21} = a_3 + (a_2/2) \cos(\theta_2)$$

$$m_{22} = a_3$$

➤ **Matrice Coriolis**

$$c_{q1} = -(a_2 \sin(\theta_2)) * (\dot{\theta}_1 * \dot{\theta}_2 + \dot{\theta}_2^2 / 2)$$

$$c_{q2} = (a_2 \sin(\theta_2)) * (\dot{\theta}_1^2 / 2)$$

➤ **Matrice de gravitation**

$$g_1 = a_4 \cos(\theta_1) + a_5 \cos(\theta_1 + \theta_2)$$

$$g_2 = a_5 \cos(\theta_1 + \theta_2)$$

➤ **Inverse de la Matrice d'inertie**

$$\det m = m_{11} * m_{22} - m_{21} * m_{12};$$

$$h_1 = c_{q1} + g_1;$$

$$h_2 = c_{q2} + g_2;$$

➤ **Calcul des accélérations angulaires**

$$\ddot{\theta}_1 = (m_{22} / \det m) * (T_1 - h_1) - (m_{12} / \det m) * (T_2 - h_2)$$

$$\ddot{\theta}_2 = -(m_{21} / \det m) * (T_1 - h_1) + (m_{11} / \det m) * (T_2 - h_2)$$

b. Correction (réseaux de neurone)

Le réseau de neurone est constitué de deux couches, une couche contient trois neurones et l'autre deux neurones. Les entres de ce réseaux sont des erreurs de position et de vitesse, et les sorties se sont des couples T1 et T2 pour la commande du robot.

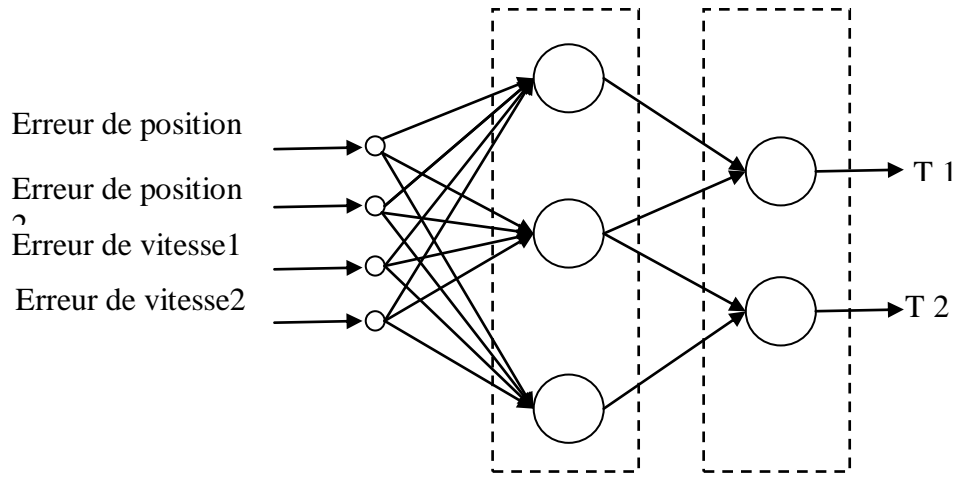


Figure (IV. 11). Réseau de neurone utilisé pour la commande de robot

c. Apprentissage

Les sorties et les entrées de réseau sont des inconnues et pour cette raisons on va utiliser un correcteur PI pour entrainer le réseau.

Cette figure démontre la méthode d'utilisation de réseau de neurone dans la boucle de régulation PI :

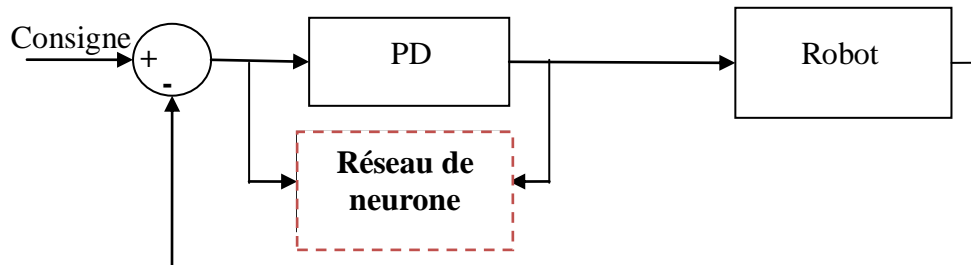


Figure (IV. 12)

• Détermination des Kv et Kp pour le correcteur PD:

L'équation dynamique de robot :

$$\tau = M(\theta)\ddot{\theta} + Q(\theta, \dot{\theta}) \tag{IV .19}$$

Loi de la commande :

$$\tau = M(\theta) \left[(\dot{\theta}_d(t) - \dot{\theta}(t)) K_v + K_p (\theta_d(t) - \theta(t)) \right] + Q(\theta, \dot{\theta}) \tag{IV .20}$$

K_v et K_p sont les matrices diagonales constantes de gain de nXn

θ_d Trajectoire désirée

L'équation d'erreur : (IV .19) dans (IV .20)

$$\ddot{E} + K_v \dot{E} + K_p E = 0 \tag{IV .21}$$

Il faut choisir \mathbf{K}_v et \mathbf{K}_p tel que les pôles de la boucle fermée du système sont placés à la situation stable désirée.

IV.5.1 L'algorithme de rétro propagation du gradient

Une cellule élémentaire à n entrées réelles $\vec{x} = (x_1, \dots, x_n)$ est définie par les poids synaptiques réels $\vec{w} = (w_0, \dots, w_n)$ et la sortie o est calculée par la formule suivante :

$$o(\vec{x}) = F[y] = \sum_{i=0}^n w_i x_i \quad \text{Avec } x_0=1$$

La fonction d'activation est une fonction identité.

Dans notre cas on a choisi à utiliser un perceptron multicouche (MPL) pour la commande de robot, l'architecture est définie précédemment.

IV.5.1.1 Introduction de l'algorithme

Le principe de l'algorithme est, comme dans le cas du perceptron linéaire, de minimiser une fonction d'erreur. Il s'agit ensuite de calculer la contribution à cette erreur de chacun des poids synaptiques. C'est cette étape qui est difficile. En effet, chacun des poids influe sur le neurone correspondant, mais, la modification pour ce neurone va influencer sur tous les neurones des couches suivantes. Ce problème est parfois désigné sous le nom de << Credit Assignment Problem >>.

Soit un MPL défini par une architecture à n entrées et à p sorties, soit \vec{w} le vecteur des poids synaptiques associés à tous les liens du réseau. L'erreur du MLP sur un échantillon d'apprentissage S d'exemples $(\vec{x}, \vec{c}$: la sortie désirer) est définie par :

$$E(\vec{w}) = \frac{1}{2} \sum_{(\vec{x}, \vec{c}) \in S} \sum_{k=1}^p (c_k - o_k)^2 \quad (\text{IV} .22)$$

Où o_k est la k -ième composante du vecteur de sortie \vec{o} calculé par le MLP sur l'entrée \vec{x} . L'erreur mesure donc l'écart entre les sorties attendue et calculée sur l'échantillon complet. On suppose (S : l'ensemble d'enchations ou exemples) fixé, le problème est donc de déterminer un vecteur \vec{w} qui minimise $E(\vec{w})$. Cependant, de la même façon que pour le perceptron avec la règle de Widrow-Hoff, plutôt que de chercher à minimiser l'erreur globale sur l'échantillon complet, on cherche à minimiser l'erreur sur chaque présentation individuelle d'exemple. L'erreur pour un exemple est :

$$E_{(\vec{x}, \vec{c})}(\vec{w}) = \frac{1}{2} \sum_{k=1}^p (c_k - o_k)^2 \quad (\text{IV} .23)$$

Nous notons E la fonction $E_{(\vec{x}, \vec{c})}$, E est une fonction des poids synaptiques, pour appliquer la méthode du gradient, il nous faut évaluer les dérivées partielles de cette fonction E par rapport aux poids synaptiques. Les calculs qui suivent sont faciles. La seule complication provient de la complexité des notations et des indices utilisés, complication due à la structure du MLP. Nous utilisons les notations suivantes :

- chaque cellule est définie par un indice,

- le réseau comporte p cellules de sortie,
- si i est l'indice d'une cellule de sortie, c_i est la sortie attendue pour cette cellule sur l'entrée \vec{x} ,
- w_{ij} est le poids synaptique associé au lien entre cellule j vers la cellule i, ce qui implique qu'elles se trouvent sur deux couches successives par définition de l'architecture,
- x_{ij} est l'entrée associée au lien entre cellule j vers cellule i,
- y_i l'entrée totale de la cellule i.
- o_i est la sortie de la cellule i, soit $o_i = F[y_i] = y_i$. fonction d'activation identité
- Succ(i) est l'ensemble des cellules qui prennent comme entrée la sortie de la cellule i, ceci implique la cellule n'est pas une cellule de sortie et que tous les éléments de Succ(i) appartiennent à la couche suivante de celle à laquelle appartient la cellule i.
- Pred(i) est l'ensemble des cellules dont la sortie est une entrée de la cellule i ; ceci implique que la cellule n'est pas une cellule d'entrée et que tous les éléments de Pred(i) appartiennent à la couche précédente de celle à laquelle appartient la cellule i,

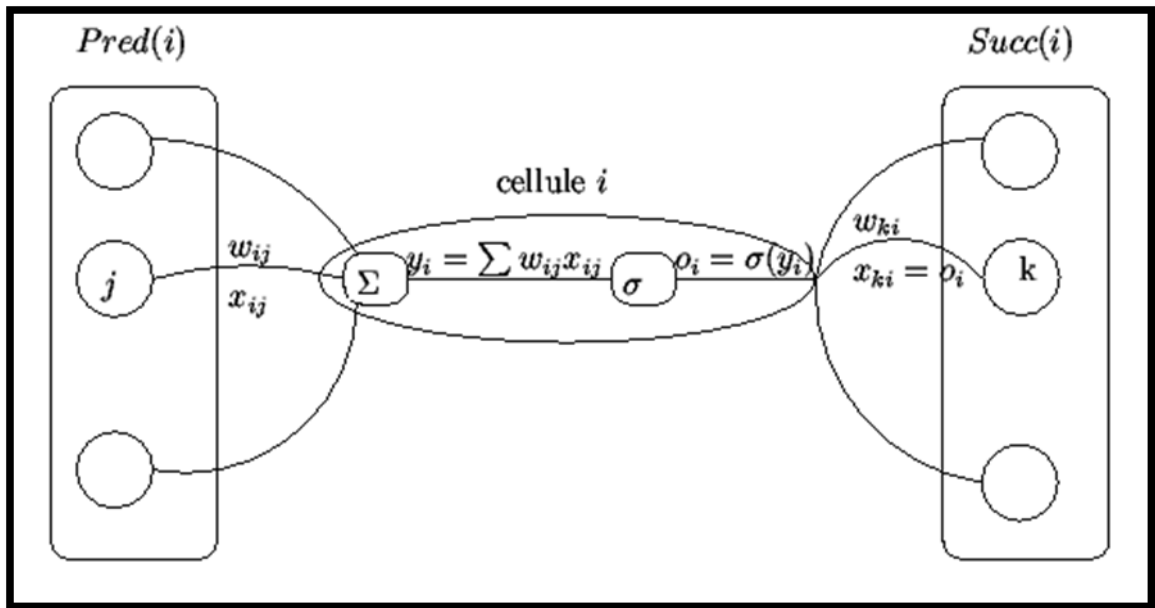


Figure (IV.13)

Il nous reste maintenant à évaluer $\frac{\partial E(\vec{w})}{\partial w_{ij}}$ que nous noterons $\frac{\partial E}{\partial w_{ij}}$. Tout d'abord remarquons que w_{ij} ne peut influencer la sortie du réseau qu'à travers le calcul de la quantité y_i , ce qui nous autorise à écrire que :

$$\frac{\partial E}{\partial w_{ij}} = \frac{\partial E}{\partial y_i} \frac{\partial y_i}{\partial w_{ij}} = \frac{\partial E}{\partial y_i} x_{ij} \tag{IV .24}$$

Il nous suffit donc de calculer $\frac{\partial E}{\partial y_i}$ pour cela, nous allons distinguer deux cas : le cas où la cellule i est une cellule de sortie et le cas où c'est une cellule interne.

• **La cellule i est une cellule de sortie**

Dans ce cas, la quantité y_i ne peut influencer la sortie du réseau que par le calcul de o_i . Nous avons donc :

$$\frac{\partial E}{\partial y_i} = \frac{\partial E}{\partial o_i} \frac{\partial o_i}{\partial y_i} \tag{IV .25}$$

Nous allons maintenant calculer chacune des deux dérivées partielles. Pour la première de ces deux dérivées nous avons :

$$\frac{\partial E}{\partial o_i} = \frac{\partial}{\partial o_i} \frac{1}{2} \sum_{k=1}^p (c_k - o_k)^2 \tag{IV .26}$$

Seul le terme correspondant à $k=i$ a une dérivée non nulle, ce qui nous donne finalement :

$$\frac{\partial E}{\partial o_i} = \frac{\partial}{\partial o_i} \frac{1}{2} (c_i - o_i)^2 = -(c_i - o_i) \tag{IV .27}$$

Pour la seconde des deux dérivées équation encadré, en utilisant la définition du calcul de la sortie d'une cellule élémentaire nous avons : $\frac{\partial o_i}{\partial y_i} = \frac{\partial f(y_i)}{\partial y_i} = 1$ fonction identité

$$\frac{\partial E}{\partial y_i} = -(c_i - o_i) \tag{IV .28}$$

• **La cellule i est une cellule interne**

Dans ce cas, la quantité y_i va influencer le réseau par tous les calculs des cellules de l'ensemble $Succ(i)$. Nous avons alors :

$$\frac{\partial E}{\partial y_i} = \sum_{k \in Succ(i)} \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial y_i} = \sum_{k \in Succ(i)} \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial o_i} \frac{\partial o_i}{\partial y_i} = \sum_{k \in Succ(i)} \frac{\partial E}{\partial y_k} \times w_{ki}$$

Donc :

$$\frac{\partial E}{\partial y_i} = \sum_{k \in Succ(i)} \frac{\partial E}{\partial y_k} \times w_{ki} \tag{IV .29}$$

Par l'étude de ces deux cas, nous avons obtenu deux équations qui nous permettent de calculer les dérivées partielles $\frac{\partial E}{\partial y_i}$ pour toute cellule i . Le calcul devra être fait pour les cellules de sortie puis des cellules de l'avant-dernière couche jusqu'aux cellules de la première couche. C'est pour cette raison que l'on parle de « rétropropagation ». Grace à l'équation (IV .25) , nous pouvons calculer toutes les dérivées partielles $\frac{\partial E(\vec{w})}{\partial w_{ij}}$ Enfin, pour en déduire la modification à effectuer sur les poids synaptiques, il nous reste simplement à rappeler que la méthode du gradient nous indique que :

$$\Delta w_{ij} = - \varepsilon \frac{\partial E(\vec{w})}{\partial w_{ij}} \quad (\text{IV .30})$$

Tous les éléments sont donc en place pour nous permettre de définir l'algorithme de rétropropagation du gradient.

IV.5.1.2 Algorithme de rétropropagation du gradient

- pour une cellule i de sortie, nous avons :

$$\delta_i = - \frac{\partial E}{\partial y_i} \quad (\text{IV .31})$$

donc:

$$\delta_i = (c_i - o_i) \quad (\text{IV .32})$$

- pour une cellule i interne, nous avons :

$$\delta_i = \sum_{k \in \text{Succ}(i)} \delta_k w_{ki} \quad (\text{IV .33})$$

La modification du poids w_{ij} est alors définie par :

$$\Delta w_{ij} = \varepsilon x_{ij} \delta_i \quad (\text{IV .34})$$

La règle de modification des poids pour le réseau MLP est : $w_{ij} \rightarrow w_{ij} + \varepsilon \delta_i x_{ij}$.

- Le choix de ε est empirique,
- Si ε est trop petit, le nombre d'itérations peut être très élevé,
- Si ε est trop grand, les valeurs de la suite risquent d'osciller autour du minimum sans converger,
- Rien ne garantit que le minimum trouvé est un minimum global.

IV.6 Perturbations non-négligeables $T_p \neq 0$

On étudiera le comportement du robot en poursuite de la même trajectoire mais cette fois-ci sous l'influence de perturbations. Nous supposons que le bras porte une charge de 10 kg et subit des frottements au niveau des actionneurs (secs et visqueux) au début du parcours, les incertitudes paramétriques sont de l'ordre de 100%. A l'instant $t=1,5$ seconde, on introduit une charge supplémentaire de 5 kg. On essayera de montrer l'erreur de poursuite par rapport au cas précédent, en tenant compte des non-linéarités introduites dans les équations du modèle par l'effet des perturbations.

L'équation dynamique de robot dans ce cas est donnée comme suit :

$$\mathbf{T} = \mathbf{M}(\boldsymbol{\theta})\ddot{\boldsymbol{\theta}} + \mathbf{N}(\boldsymbol{\theta}, \dot{\boldsymbol{\theta}}) + \mathbf{G}(\boldsymbol{\theta}) + \mathbf{H}(\dot{\boldsymbol{\theta}}) + \mathbf{m}\mathbf{J}^T(\mathbf{J}\ddot{\boldsymbol{\theta}} + \dot{\mathbf{J}}\dot{\boldsymbol{\theta}})$$

Avec :

$$\mathbf{T}_p = \mathbf{H}(\dot{\boldsymbol{\theta}}) + \mathbf{m}\mathbf{J}^T(\mathbf{J}\ddot{\boldsymbol{\theta}} + \dot{\mathbf{J}}\dot{\boldsymbol{\theta}})$$

Calcul de T_p

Pour éviter les erreurs de calcul on a utilisé logiciel Maple pour calculer T_p

Calcul l'inverse de l'équation dynamique ($\ddot{\boldsymbol{\theta}} = \mathbf{f}^{-1}(\mathbf{T})$) pour $T_p \neq 0$

➤ Matrice d'inertie plus le terme $\mathbf{m}\mathbf{J}^T\mathbf{J}$

$$m11 = a1 + a2 * \cos(\theta2) + m * L1^2 + 2 * m * L2 * L1 * \cos(\theta2) + m * L2^2;$$

$$m12 = a3 + (a2/2) * \cos(\theta2) + m * L2 * L1 * \cos(\theta2) + m * L2^2;$$

$$m21 = a3 + (a2/2) * \cos(\theta2) + m * L2 * L1 * \cos(\theta2) + m * L2^2;$$

$$m22 = a3 + m * L2^2;$$

➤ Matrice Coriolis plus le terme $\mathbf{m}\mathbf{J}^T\mathbf{j}\dot{\boldsymbol{\theta}}$

$$cq1 = -a2 * \sin(\theta2) * (\dot{\theta1} * \dot{\theta2} + (\dot{\theta2}^2/2)) - \dot{\theta2} * m * L2 * L1 * \sin(\theta2) * (2 * \dot{\theta1} + \dot{\theta2});$$

$$cq2 = a2 * \sin(\theta2) * (\dot{\theta1}^2/2) + m * L2 * L1 * (\dot{\theta1}^2) * \sin(\theta2);$$

➤ Matrice de gravitation

$$g1 = a4 * \cos(\theta1) + a5 * \cos(\theta1 + \theta2)$$

$$g2 = a5 * \cos(\theta1 + \theta2)$$

➤ Matrice de frottement

$$H1 = V1 * \dot{\theta1} + V2 * \text{sign}(\dot{\theta1});$$

$$H2 = V3 * \dot{\theta2} + V4 * \text{sign}(\dot{\theta2});$$

$$\det m = m11 * m22 - m21 * m12;$$

$$h1 = cq1 + g1 + H1;$$

$$h2 = cq2 + g2 + H2;$$

➤ Calcul des accélérations angulaires

$$\ddot{\theta}_1 = (m22/\det m) * (T1 - h1) - (m12/\det m) * (T - h2);$$

$$\ddot{\theta}_2 = -(m21/\det m) * (T1 - h1) + (m11/\det m) * (T - h2);$$

IV .7 Conclusion

Ce chapitre a présenté les réseaux de neurone artificiel, ainsi que des algorithmes d'apprentissage tel que l'algorithme de rétro propagation du gradient.

L'objectif de ce chapitre est la commande d'un bras manipulateur on utilise les réseaux de neurone, des résultats de poursuite de trajectoire sont détaillé dans le prochaine chapitre.

Chapitre V

Résultats de simulation

CHAPITRE V

RÉSULTATS DE SIMULATION

V.1 Introduction

Dans ce chapitre nous présentons les résultats de simulations qui sont présentés dans ce qui suit ont été obtenus pour une articulation définie .Nous remarquons la très bonne poursuite entre les angles désirés et obtenus .On a utilisé des oscilloscopes de Simulink et des Works pace pour tracer les différentes courbes.

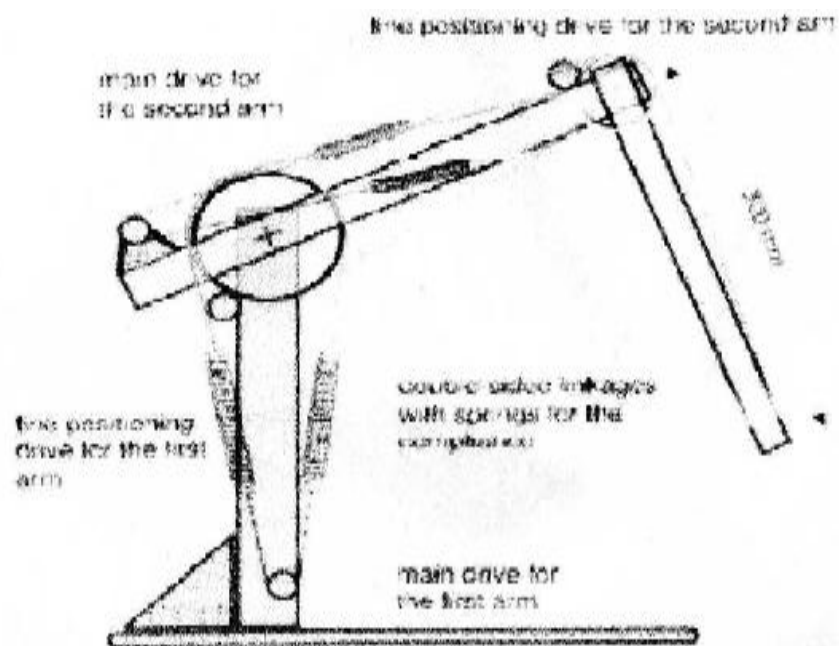


Schéma du robot a deux degré de liberté avec ressorts

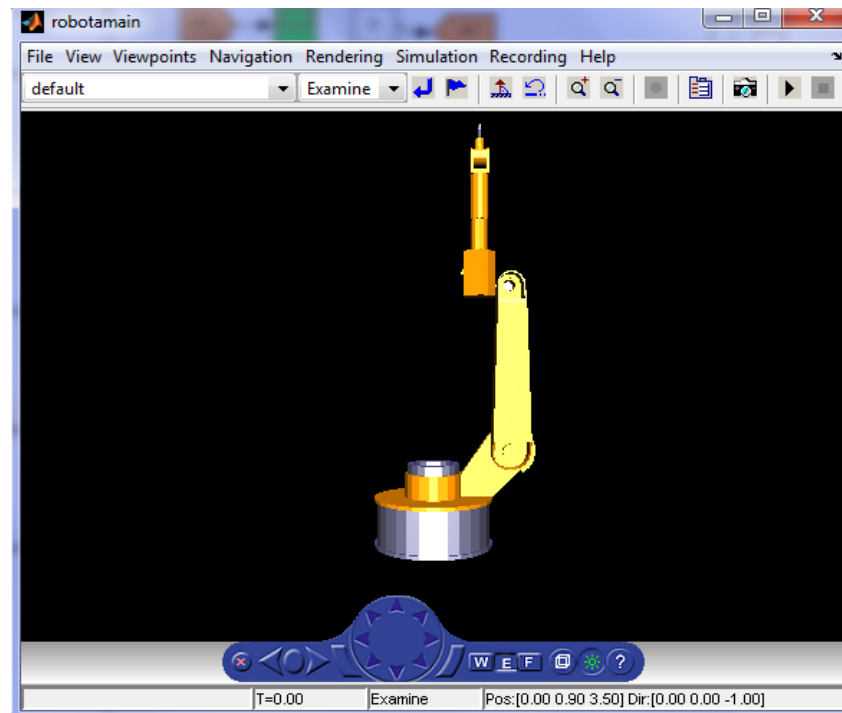
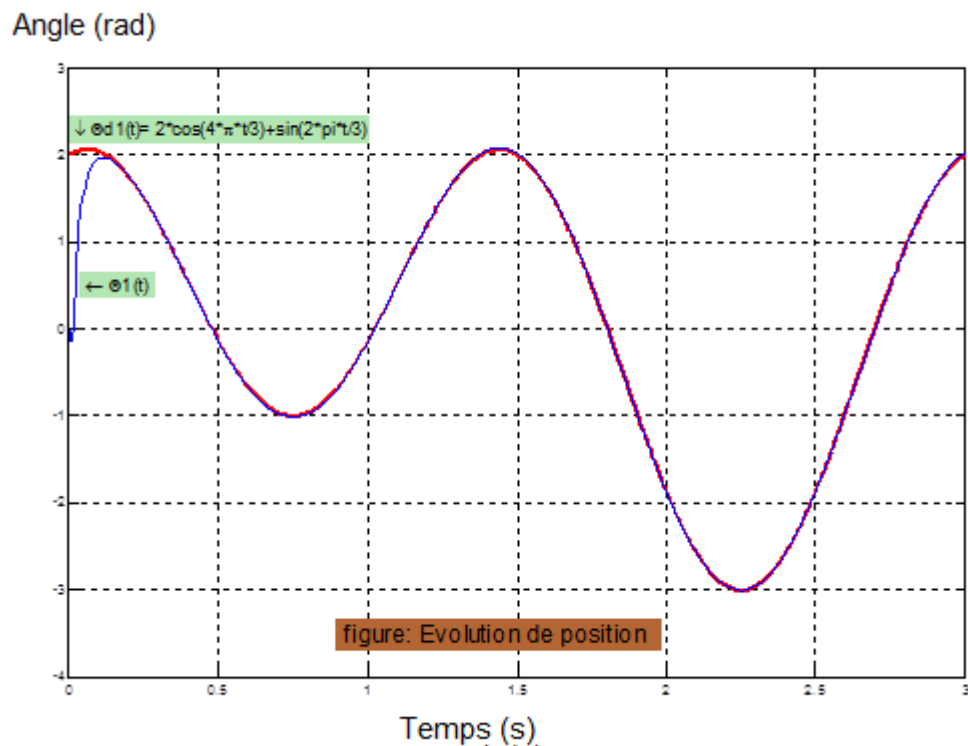
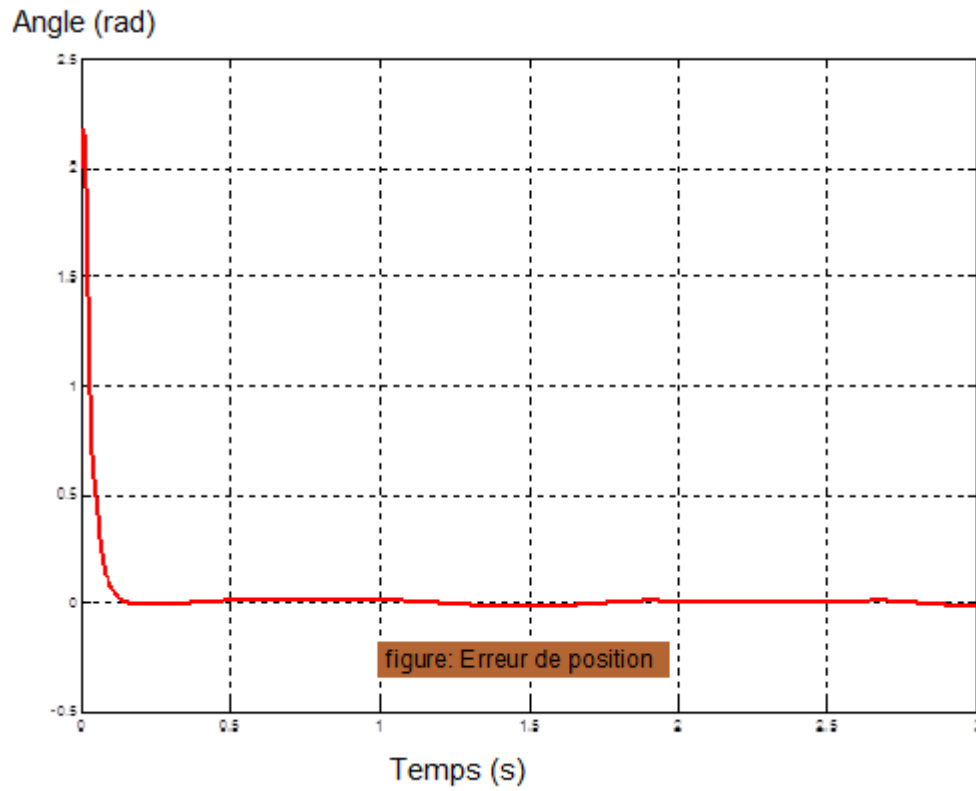
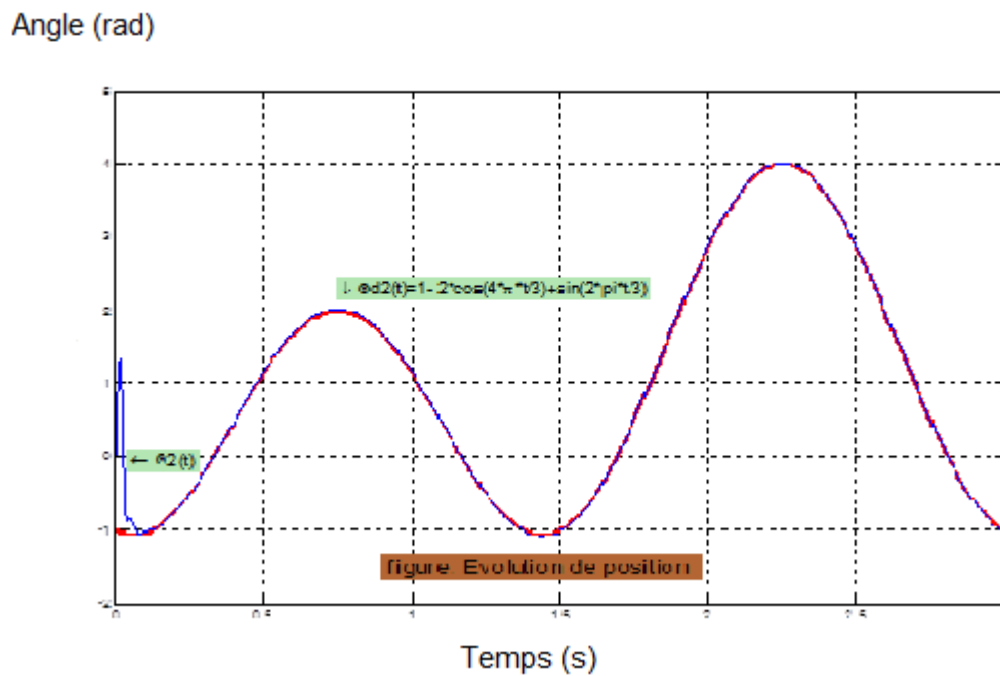


Figure (V. a) : Bras Manipulateur virtuel

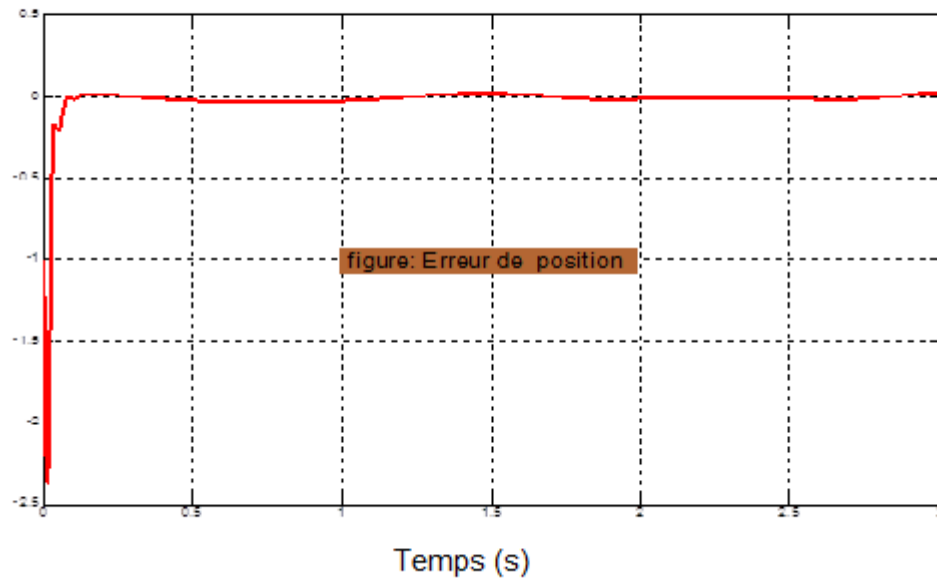
V.2 Les courbes obtenues pour les deux angles (θ_1, θ_2)

Les courbes obtenues pour les deux angles (θ_1, θ_2)

Figure (V.1) : La poursuite de θ_1 de la trajectoire désirée

Figure (V.2) : Erreur de position θ_1 Figure (V.3) : La poursuite de θ_2 de la trajectoire désirée

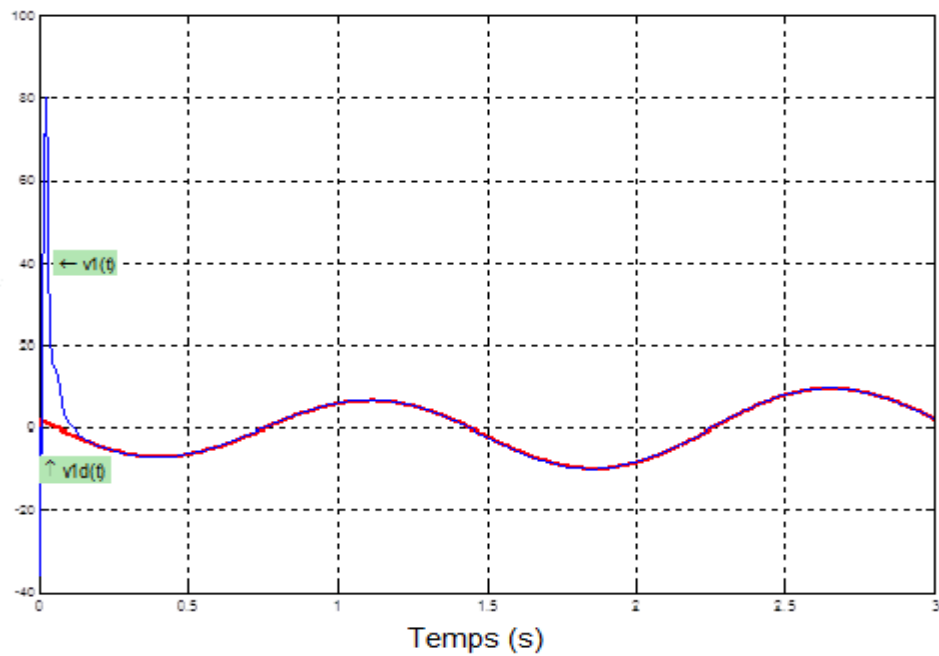
Angle (rad)

Figure (V.4) : Erreur de position θ_2

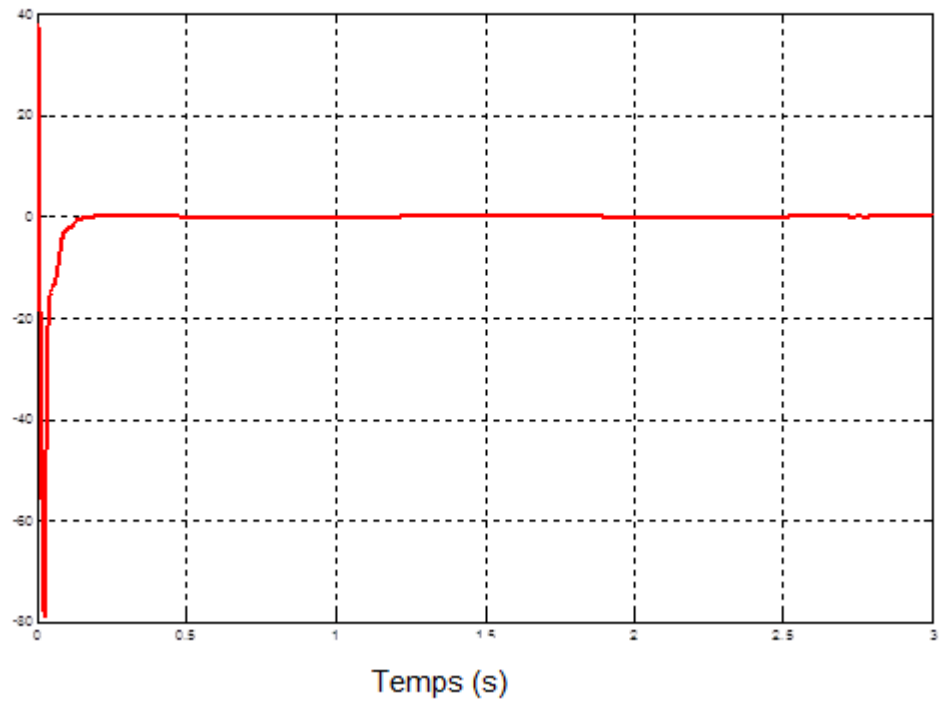
V.3 Les courbes obtenues pour les deux vitesses $(\dot{\theta}_1, \dot{\theta}_2)$

Les courbes obtenues pour les deux vitesses $(\dot{\theta}_1, \dot{\theta}_2)$

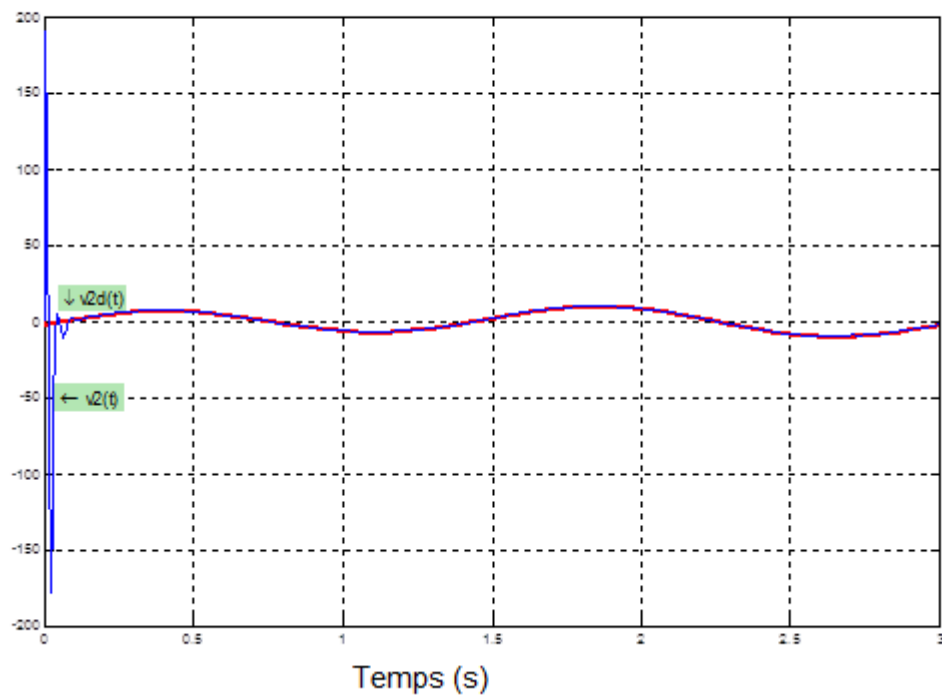
Vitesse (rad/s)

Figure (V.5) : La poursuite de $\dot{\theta}_1$ de la trajectoire désirée

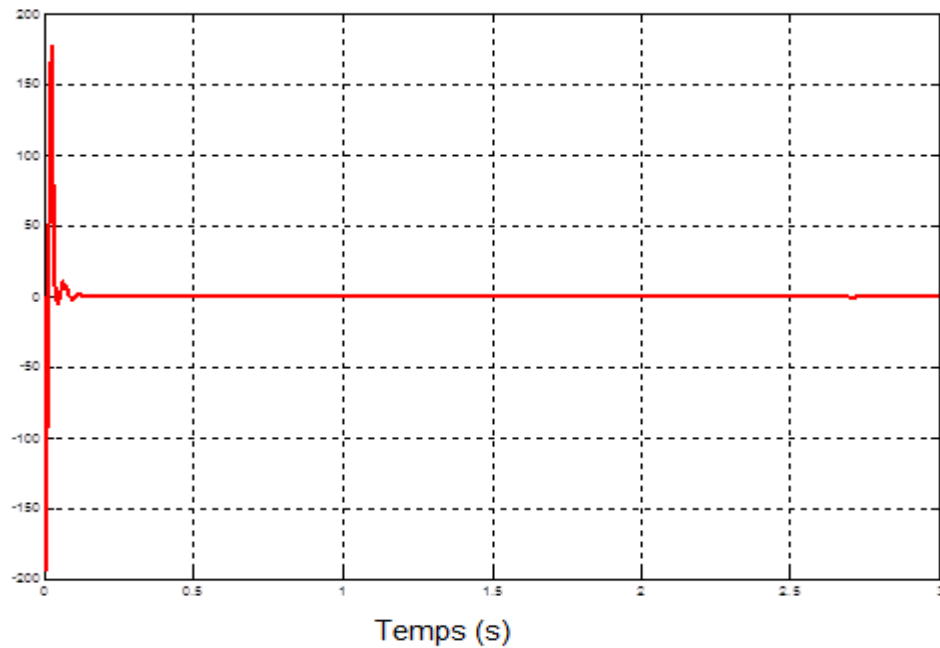
Vitesse (rad/s)

Figure (V.6) : Erreur de vitesse $\dot{\theta}_1$

Vitesse (rad/s)

Figure (V.7) : La poursuite de $\dot{\theta}_2$ de la trajectoire désirée

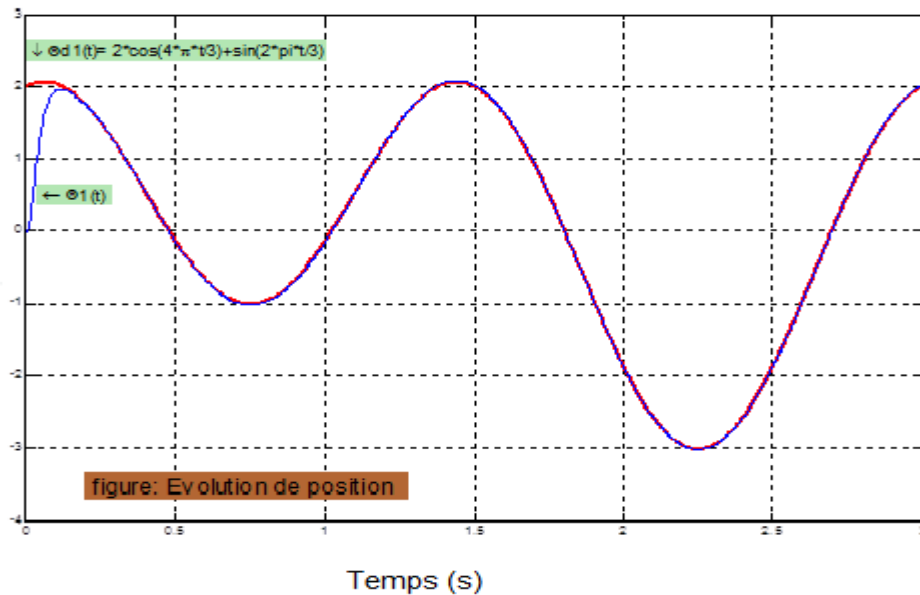
Vitesse (rad/s)

Figure (V.8) : Erreur de vitesse $\dot{\theta}_2$

V.4 L'évolution de l'erreur en fonction de la charge

➤ Pour une charge $m=10\text{kg}$ ➔
 Erreur de position négligeable

Angle (rad)

Figure (V.9) : La poursuite de θ_1 de la trajectoire désirée

Angle (rad)

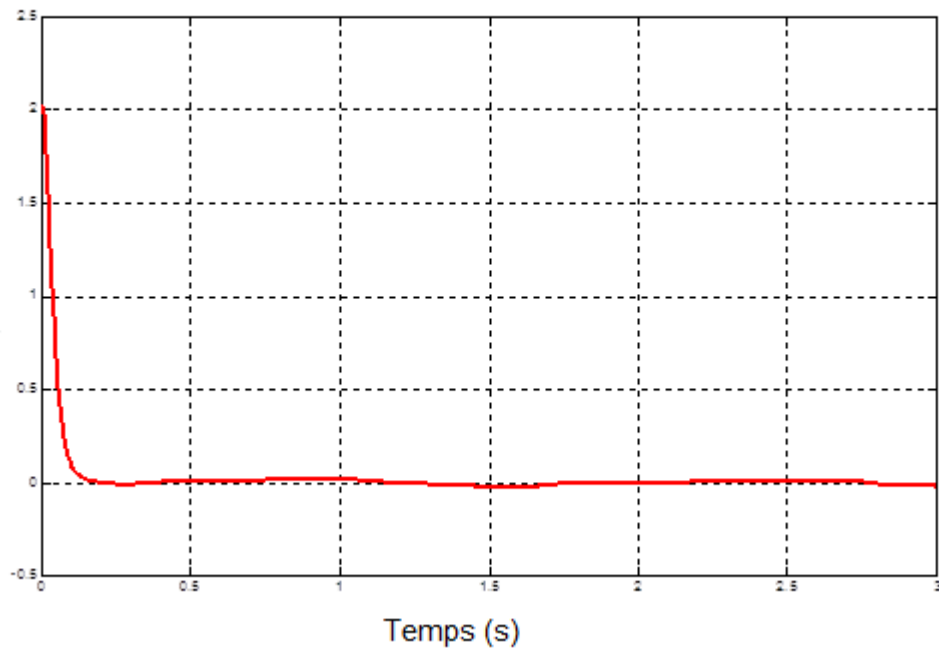
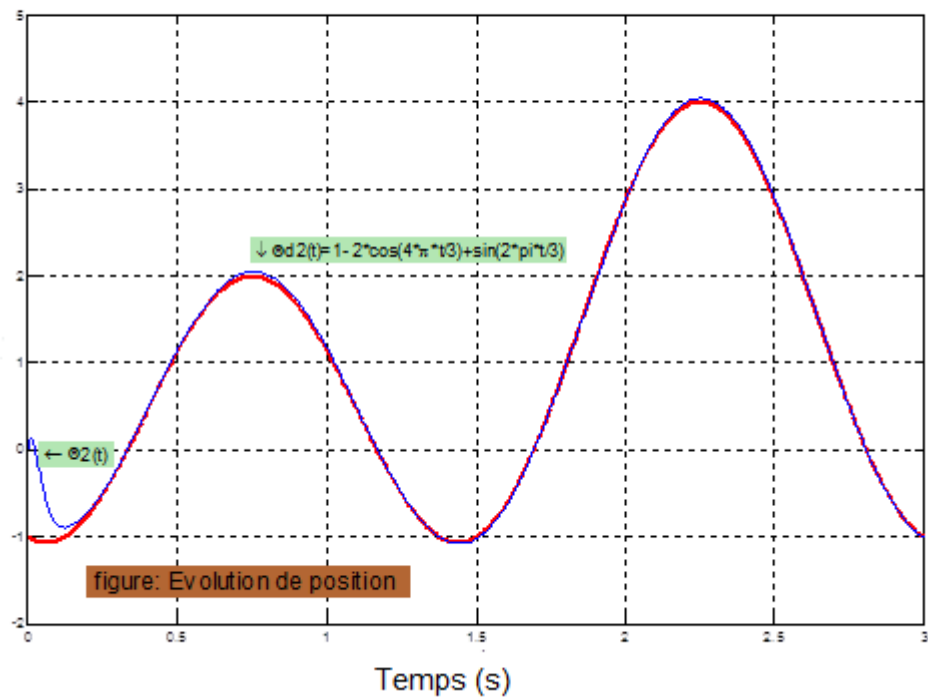


Figure (V.10) : Erreur de position

Angle (rad)

Figure (V.11) : La poursuite de θ_2 de la trajectoire désirée

Angle (rad)

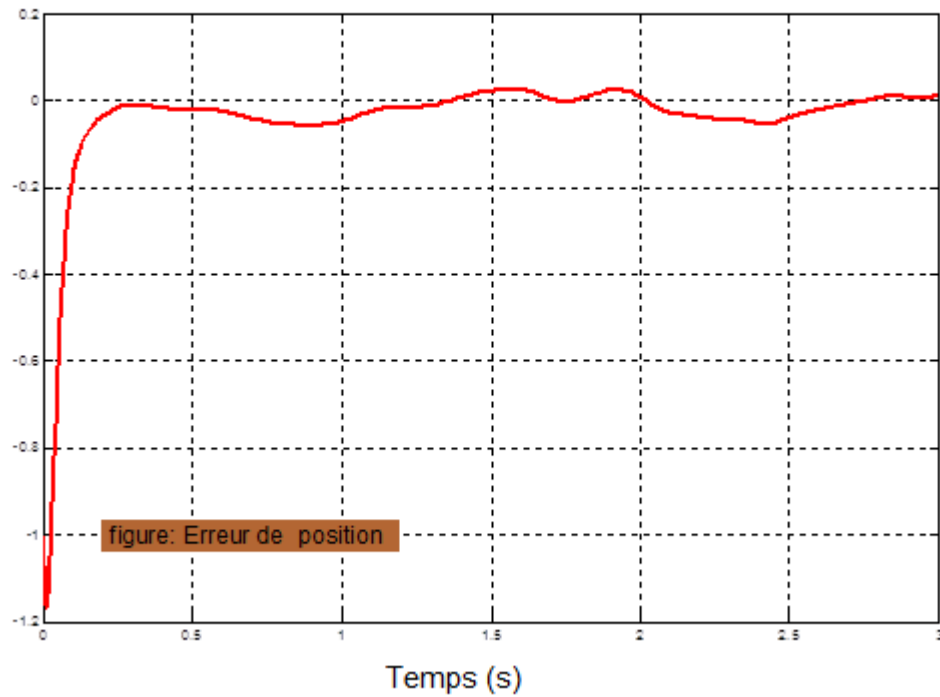
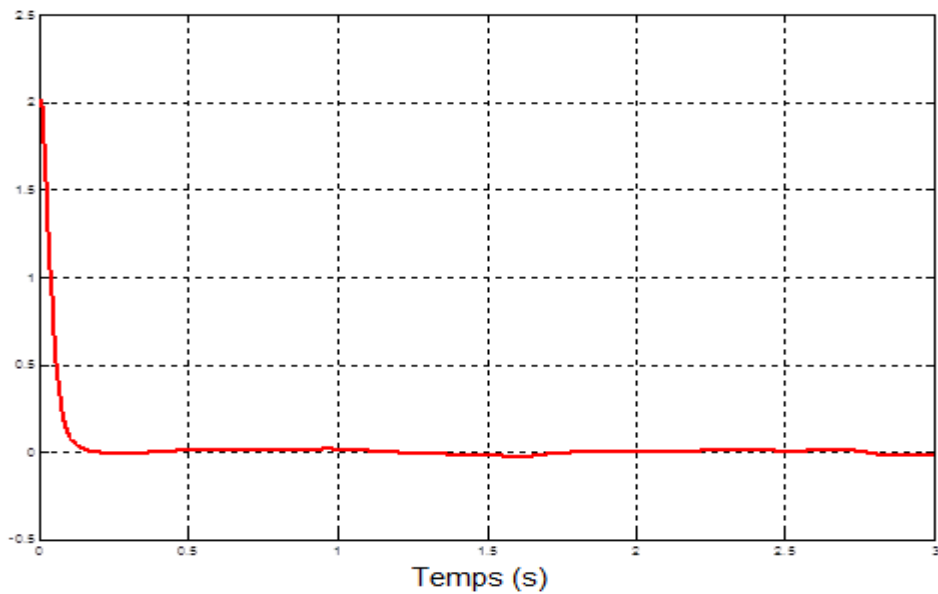


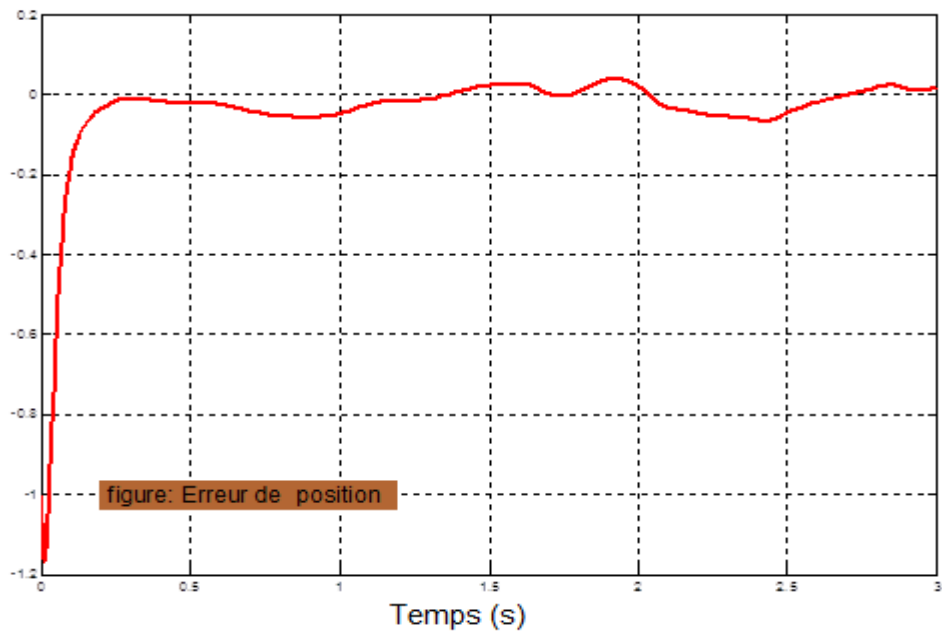
Figure (V.12) : Erreur de position

➤ Pour une charge $m=15\text{kg}$ →
Erreur de position presque négligeable

Angle (rad)

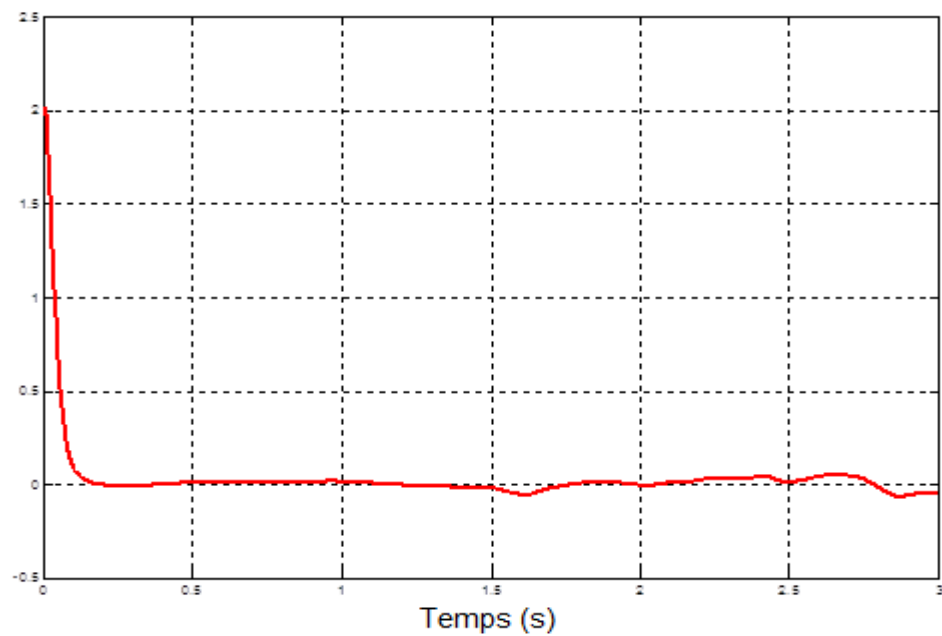
Figure (V.13) : Erreur de poursuite de θ_1

Angle (rad)

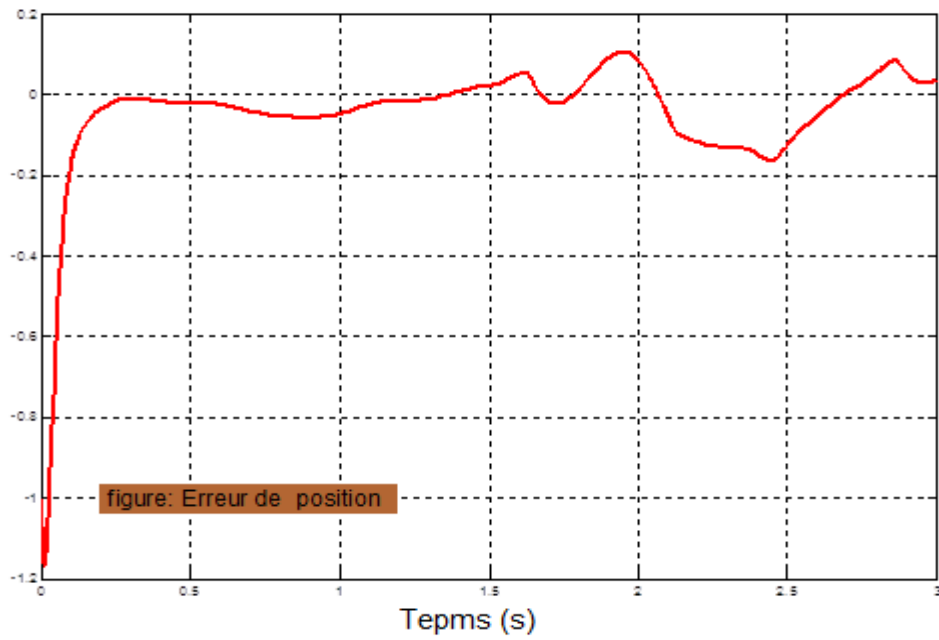
Figure (V.14) : Erreur de poursuite de θ_2

➤ Pour une charge $m=50\text{kg}$ →
Il y a une erreur de position

Angle (rad)

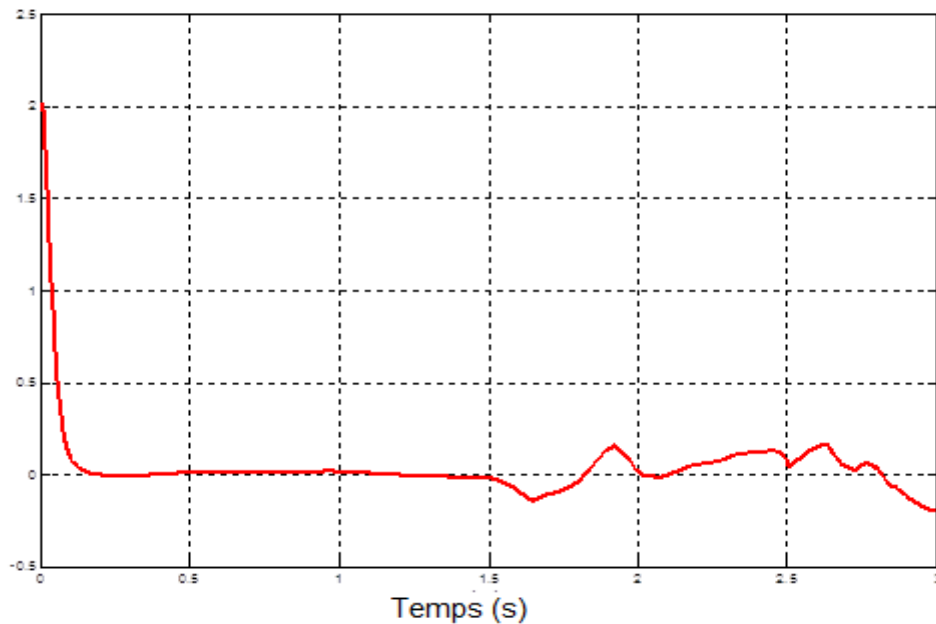
Figure (V.15) : Erreur de poursuite de θ_1

Angle (rad)

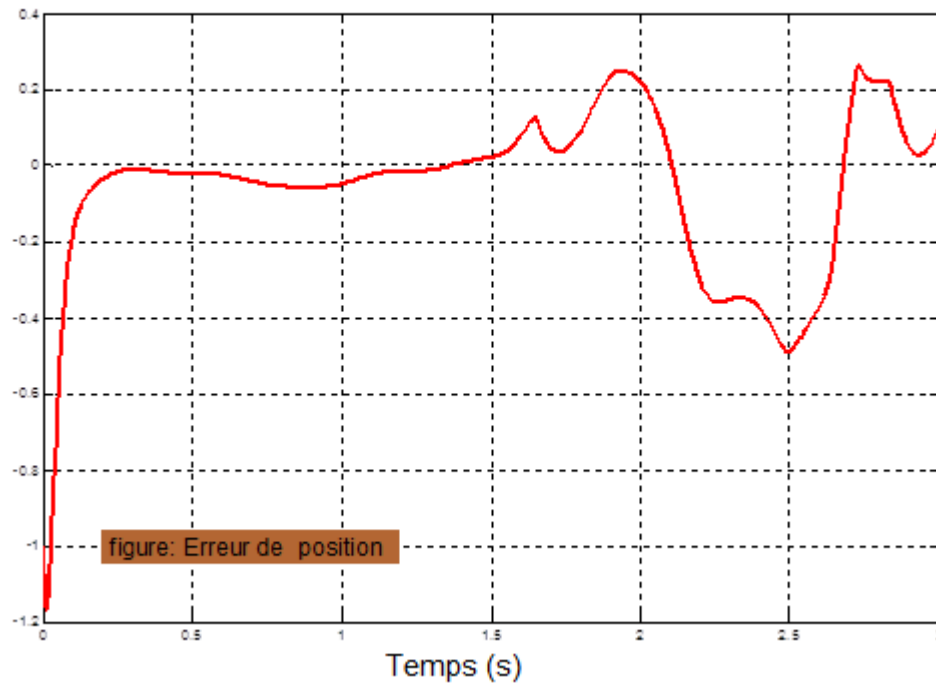
**Figure (V.16) : Erreur de poursuite de θ_2**

➤ La charge $m=150\text{kg}$ ➔ une croissance des deux erreurs.

Angle (rad)

**Figure (V.17) : Erreur de poursuite de θ_1**

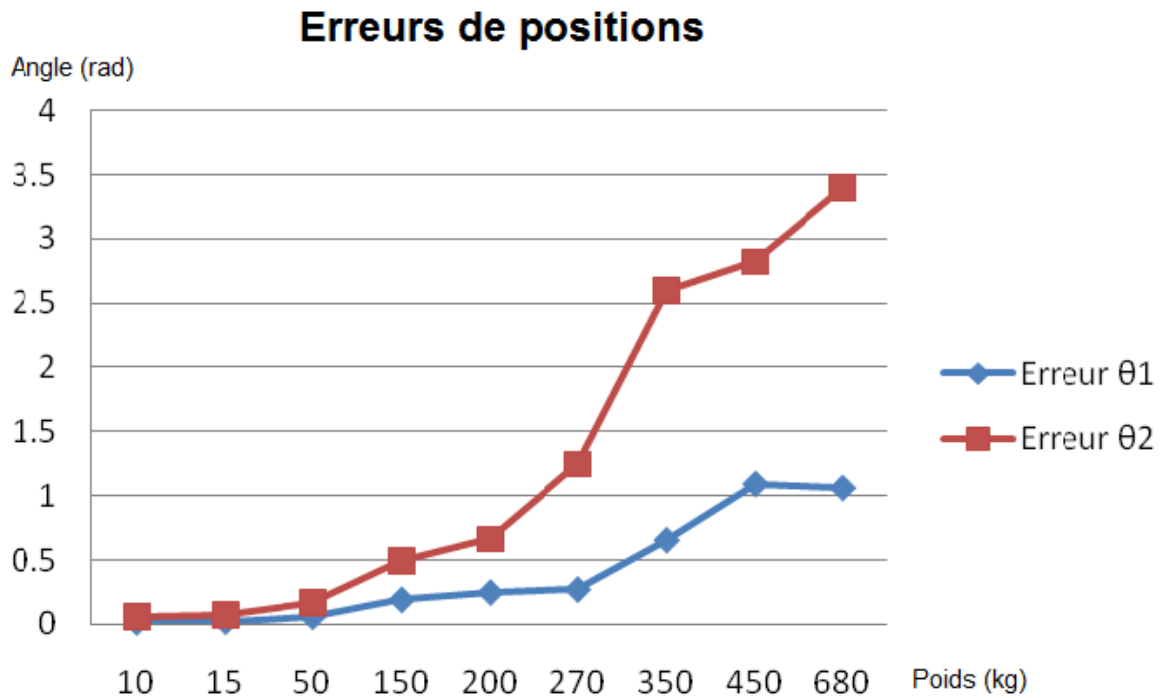
Angle (rad)

Figure (V.18) : Erreur de poursuite de θ_2

4 L'évolution de l'erreur en fonction de la charge

Ce tableau démontre l'évolution de l'erreur en fonction de la charge

Charge(Kg)	10	15	50	150	200	270	350	450	680
(θ_1) Erreur maximal (rad)	0.0202	0.02535	0.06371	0.1954	0.2516	0.2743	0.6569	1.092	1.061
(θ_2) Erreur maximal (rad)	0.05248	0.06588	0.1641	0.4904	0.6617	1.246	2.596	2.822	3.4



Selon le graphe on constate qu'il y a une croissance des deux erreurs en fonction de la charge.

V.5 Conclusion

Dans ce dernier chapitre nous avons réalisé l'aspect pratique de ce projet. des méthodes ont été mises en œuvre tels que la régulation classique, à savoir un contrôleur de type P.D. dont l'implémentation est relativement simple, mais cette commande n'est pas robuste puisqu'elle est sensible aux variations de la charge.

La commande intelligente basée sur les réseaux de neurones pour ce système dynamique nécessite un entraînement en ligne permet une adaptation des poids des neurones des différentes couches pour une position donnée de l'organe effecteur du bras.

Conclusion Générale

CONCLUSION GÉNÉRALE

L'objectif principal assigné à ce travail est de concevoir un robot capable de fournir une position précise et d'effectuer l'examen, et peut subir un déplacement sous l'effet d'un effort dû au contact avec l'environnement sans altérer sa précision de positionnement.

Le bras manipulateur flexible étudié dans le présent mémoire est un système hautement non linéaire, du fait que sa modélisation cinématique et dynamique abouti à des équations différentielles obtenues sous forme d'équations d'état en utilisant la théorie d'Euler-Lagrange, dont les coefficients sont variables suivant les paramètres mécaniques du bras et les configurations des mouvements désirés.

Malgré ces complexités, on a pu élaborer un modèle et on a pu l'implémenter pour une simulation de commande classique et intelligente sous l'environnement Matlab-Simulink.

L'utilisation des robots à liaisons flexibles complique le problème de la commande en position du fait que ces liaisons sont assujetties à des déflexions et /ou des vibrations. On a pu mettre en oeuvre quelques techniques utilisées dans le domaine de la commande des robots flexibles.

Deux techniques de commande ont été appliquées pour la commande de ce bras manipulateur: une technique de régulation classique, à savoir un contrôleur de type P.D. dont l'implémentation est relativement simple, mais cette commande n'est pas robuste puisqu'elle est sensible aux variations de la charge.

Une deuxième commande intelligente basée sur les réseaux de neurones pour la commande de ce système dynamique nécessite un entraînement en ligne permet une adaptation des poids des neurones des différentes couches pour une position donnée de l'organe effecteur du bras.

La réalité virtuelle est devenue un outil très puissant pour la conception et la fabrication des produits industriels de haute gamme dans l'industrie occidentale : des logiciels comme le très populaire java ou le VRML ont couramment utilisés. Grace au logiciel matlab/Simulink et les boites à outils simmechanics et Virtual, on a pu concevoir un bras manipulateur virtuel qui montre les fonctionnalités principales.

Une Bibliographie assez riche concernant notre recherche est incluse afin d'aider et orienter les futurs chercheurs dans ce domaine.

Propositions pour travail future

La modélisation et les méthodes de commande utilisées dans ce mémoire ont été mises en oeuvre par des simulations utilisant le logiciel MATLAB-SIMULINK, pour faire une demonstration réelle

et actuelle dans le but de prouver la validation de ces méthodes un travail expérimental peut être envisagé.

Le modèle de simmechanics peut être modifié ou amélioré pour d'autres applications ainsi que la commande .Ceci est laissé comme suggestions pour d'autres étudiants dans ce domaine très attirant et intéressant.

ANNEXE

ANNEXES

Annexe A

Perturbation négligeable

- Les gains

$$K_{p1} = 19000$$

$$K_{v1} = 800$$

$$K_{p2} = 19000$$

$$K_{v2} = 1000$$

$$b_4 = 2.9142$$

$$b_5 = 76.7808$$

- Les pois

$$w_{11} = 108.9503$$

$$w_{21} = -6.3160$$

$$w_{31} = 0.5081$$

$$w_{41} = -18.4609$$

$$w_{12} = 80.3560$$

$$w_{22} = 56.9637$$

$$w_{32} = 21.5510$$

$$w_{42} = 87.8017$$

$$w_{13} = 146.4295$$

$$w_{23} = 71.1466$$

$$w_{33} = 4.1115$$

$$w_{43} = -0.0815$$

$$= 126.0039$$

$$w_{24} = 28.2039$$

$$w_{25} = 2.1472$$

$$w_{35} = 125.1828$$

$$b_1 = -0.8966$$

$$b_2 = 4.4091$$

$$b_3 = -0.1437$$

$$\varepsilon_1 = 0.00000000610$$

$$\varepsilon_2 = 0.00000000773$$

$$\varepsilon_3 = 0.00000000598$$

$$\varepsilon_4 = 0.0000000101$$

$$\varepsilon_5 = 0.0000000116$$

Perturbation non négligeable

- Les gains

$$K_{p1} = 19000$$

$$K_{v1} = 800$$

$$K_{p2} = 19000$$

$$K_{v2} = 1000$$

- Les pois

$$w_{11} = 70.7672$$

$$w_{21} = 9.9362$$

$$w_{31} = 8.0257$$

$$w_{41} = -13.7450$$

$$w_{12} = 142.5287$$

$$w_{22} = 27.1620$$

$$w_{32} = -0.4969$$

$$w_{42} = 13.7103$$

$$w_{13} = 117.0524$$

$$w_{23} = 85.5028$$

$$w_{33} = 12.6330$$

$$w_{43} = -9.6499$$

$$w_{14} = 102.0774$$

$$w_{24} = 97.9333$$

$$w_{25} = 65.9974$$

$$w_{35} = 55.0021$$

$$b_1 = 1.3257$$

$$b_2 = -1.1725$$

$$b_3 = 1.2403$$

$$b_4 = 2.9511$$

$$b_5 = 77.0000$$

$$\varepsilon_1 = 0.00000000640$$

$$\varepsilon_2 = 0.00000000833$$

$$\varepsilon_3 = 0.00000000518$$

$$\varepsilon_4 = 0.0000000291$$

$$\varepsilon_5 = 0.000000000001$$

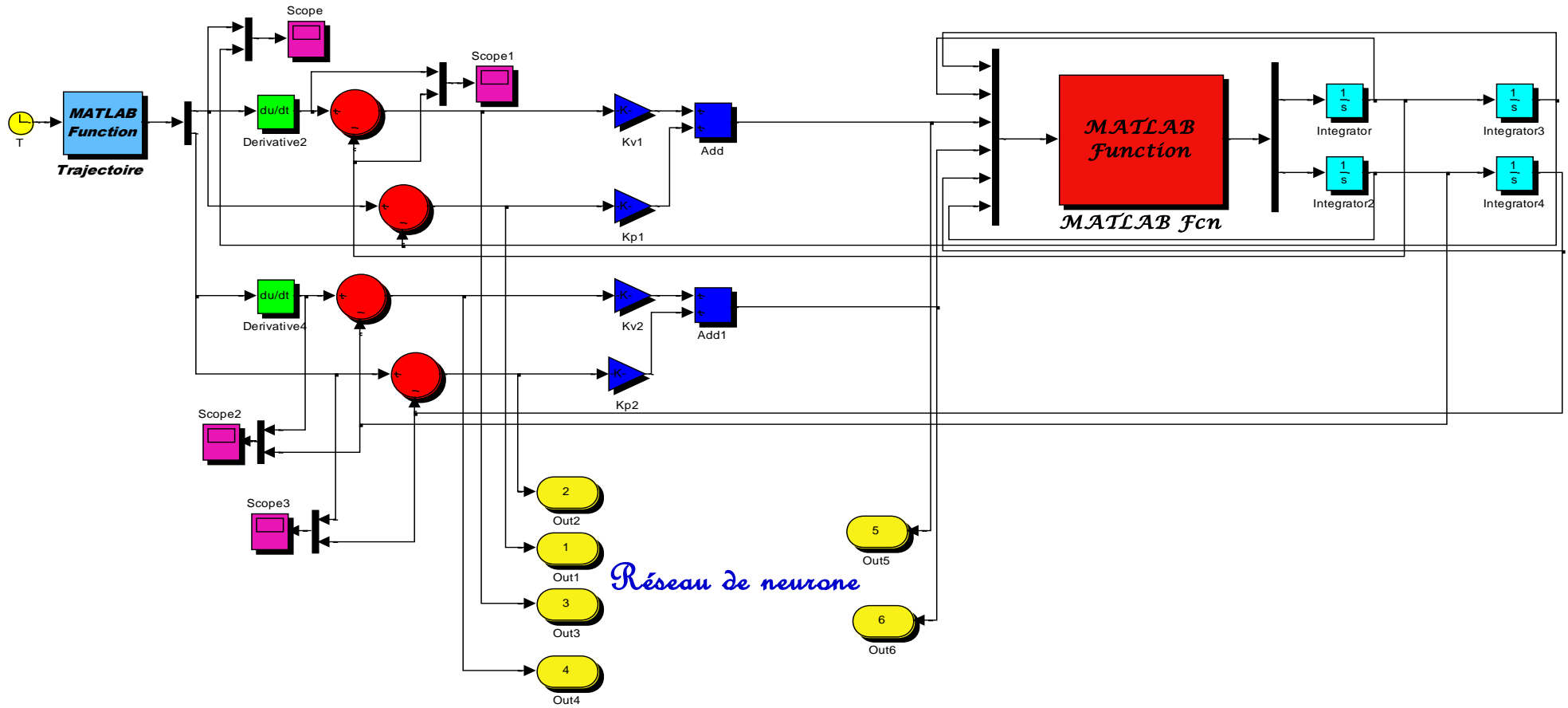


Figure A1 : Utilisation de la boucle de régulation PD pour entrainer le réseau de neurone

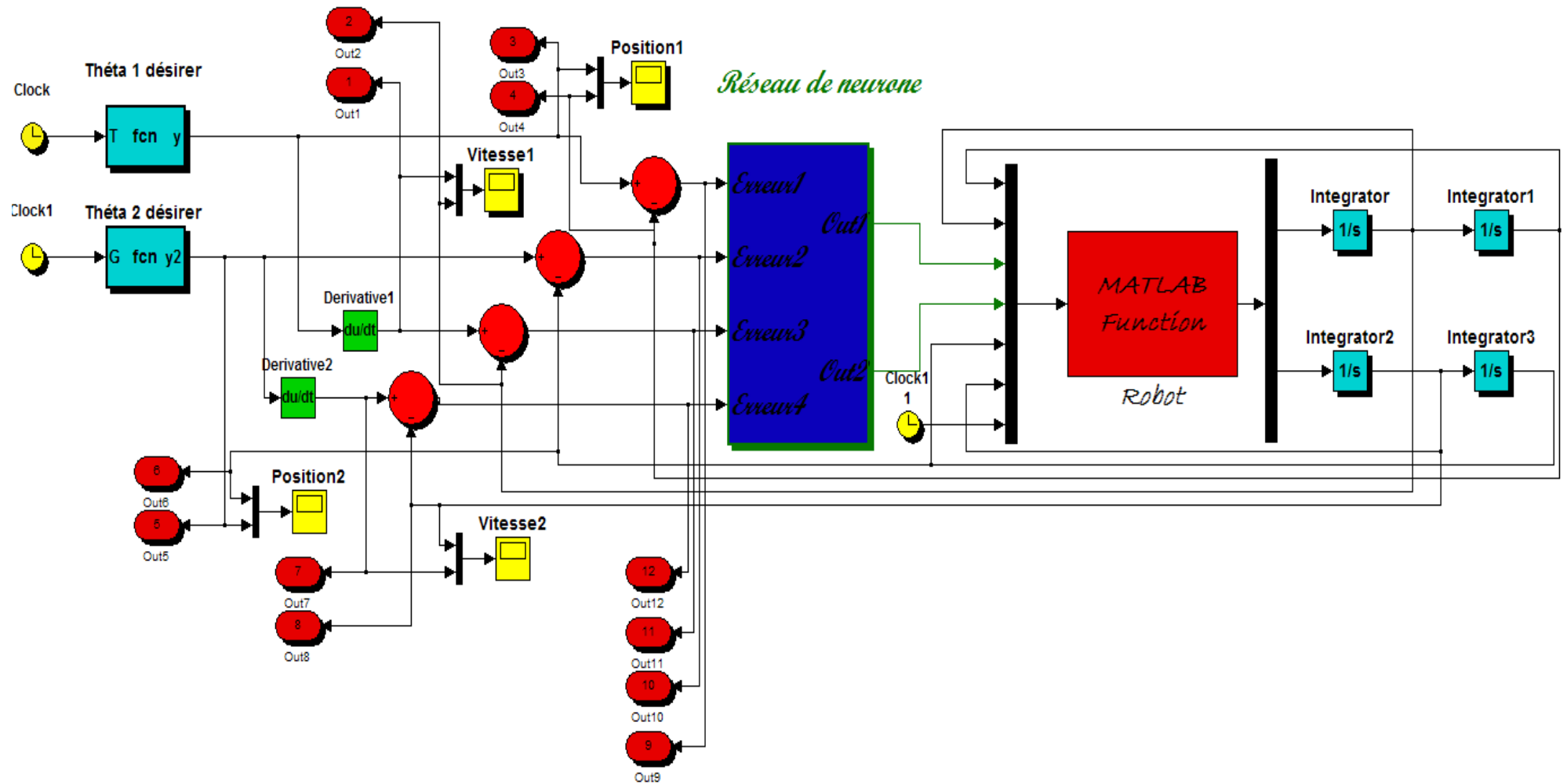


Figure A2 : la boucle de régulation de réseau de neurone

Annexe B

- **Fichier script d'initialisation des paramètres du modèle de robot**

Perturbation négligeable

Apprentissage

```
[t,x,a,z,e,r,y,u]=sim('PI',[0 3]);
[m,n]=size(a)
W1=[2 50 20 20 -2 ];
W2=[5 60 66 77 88 ];
W3=[4 88 99 44 22 ];
W4=[3 33 44 ];
W5=[77 66 55 ];
for(ii=1:344)
for(j=1:1:m)
entree=[1;a(j,1);z(j,1);e(j,1);r(j,1)];
O1=W1(1,1)+W1(1,2)*a(j,1)+W1(1,3)*z(j,1)+
W1(1,4)*e(j,1)+W1(1,5)*r(j,1);
O2=W2(1,1)+W2(1,2)*a(j,1)+W2(1,3)*z(j,1)+
W2(1,4)*e(j,1)+W2(1,5)*r(j,1);
O3=W3(1,1)+W3(1,2)*a(j,1)+W3(1,3)*z(j,1)+
W3(1,4)*e(j,1)+W3(1,5)*r(j,1);
O12=[1;O1;O2];
O23=[1;O2;O3];
O4=W4(1,1)+W4(1,2)*O1+W4(1,3)*O2;
O5=W5(1,1)+W5(1,2)*O2+W5(1,3)*O3;
CeSor1=(y(j,1)-O4);
CeSor2=(y(j,1)-O5);
CeIn1=(W4(1,2)*CeSor1);
CeIn2=(W4(1,3)*CeSor1+W5(1,2)*CeSor2);
CeIn3=(W5(1,3)*CeSor2);
for(i=1:1:5)
W1(1,i)=W1(1,i)+0.00000000610*CeIn1*entree
e(i,1);
W2(1,i)=W2(1,i)+0.00000000773*CeIn2*entree
e(i,1);
W3(1,i)=W3(1,i)+0.00000000598*CeIn3*entree
e(i,1);
End
```

```
for (i=1:1:3)
W4(1,i)=W4(1,i)+0.0000000101*CeSor1*O12
(i,1);
W5(1,i)=W5(1,i)+0.0000000116*CeSor2*O23
(i,1);
end
end
end
W1
W2
W3
W4
W5
```

Robot

```
function [ths]=robot(entree)
%simulation du Robot deux degre de libertes
th1=entree(1);
thp1=entree(2);
to1=entree(3);
to2=entree(4);
th2=entree(5);
thp2=entree(6);
% initialisation des termes dynamiques
m1=15.91;
m2=11.36;
L1=0.432;
L2=0.432;
a1=3.82;
a2=2.12;
a3=0.71;
a4=81.82;
a5=24.06;
V1=1;
V3=1;
V2=0.5;
V4=0.5;
% Matrice d'inertie
m11=a1+a2*cos(th2);
m12=a3+(a2/2)*cos(th2);
m21=a3+(a2/2)*cos(th2);
m22=a3;
% Matrice coriolis
cq1=-(a2*sin(th2))*(thp1*thp2+(((thp2)^2)/2));
```

```

cq2=(a2*sin(th2)*(((thp1)^2)/2));
%Matrice de gravitation
g1=a4*cos(th1)+a5*cos(th1+th2);
g2=a5*cos(th1+th2);
%invers de la Matrice d'inertie
detm=m11*m22-m21*m12;
h1=cq1+g1;
h2=cq2+g2;
% calcul des accelerations angulaires
ths1=(m22/detm)*(to1-h1)-(m12/detm)*(to2-
h2);
ths2=-(m21/detm)*(to1-h1)+(m11/detm)*(to2-
h2);
ths=[ths1 ths2];

```

Robot commande PD

```

function [ths]=Robot2(entree)
%simulation du Robot deux degre de libertes
th1=entree(1);
thp1=entree(2);
to1=entree(3);
to2=entree(4);
th2=entree(5);
thp2=entree(6);
% initialisation des termes dynamiques
m1=15.91;
m2=11.36;
L1=0.432;
L2=0.432;
a1=3.82;
a2=2.12;
a3=0.71;
a4=81.82;
a5=24.06;
V1=1;
V3=1;
V2=0.5;
V4=0.5;
%Matrice d'inertie
m11=a1+a2*cos(th2);
m12=a3+(a2/2)*cos(th2);
m21=a3+(a2/2)*cos(th2);
m22=a3;

```

```

%Matrice coriolis
cq1=-(a2*sin(th2))*(thp1*thp2+(((thp2)^2)/2));
cq2=(a2*sin(th2)*(((thp1)^2)/2));
%Matrice de gravitation
g1=a4*cos(th1)+a5*cos(th1+th2);
g2=a5*cos(th1+th2);
%invers de la Matrice d'inertie
detm=m11*m22-m21*m12;
h1=cq1+g1;
h2=cq2+g2;
% calcul des accelerations angulaires
ths1=(m22/detm)*(to1-h1)-(m12/detm)*(to2-
h2);
ths2=-(m21/detm)*(to1-h1)+(m11/detm)*(to2-
h2);
ths=[ths1 ths2];

```

Perturbation non négligeable

Apprentissage

```

[t,x,a,z,e,r,y,u]=sim('PI2',[0 3]);
[m,n]=size(a)
W1=[2 50 20 20 -2 ];
W2=[5 60 66 77 88 ];
W3=[4 88 99 44 22 ];
W4=[3 33 44 ];
W5=[77 66 55 ];
for(ii=1:260)
for(j=1:1:m)
entree=[1;a(j,1);z(j,1);e(j,1);r(j,1)];
O1=W1(1,1)+W1(1,2)*a(j,1)+W1(1,3)*z(j,1)+
W1(1,4)*e(j,1)+W1(1,5)*r(j,1);
O2=W2(1,1)+W2(1,2)*a(j,1)+W2(1,3)*z(j,1)+
W2(1,4)*e(j,1)+W2(1,5)*r(j,1);
O3=W3(1,1)+W3(1,2)*a(j,1)+W3(1,3)*z(j,1)+
W3(1,4)*e(j,1)+W3(1,5)*r(j,1);
O12=[1;O1;O2];
O23=[1;O2;O3];
O4=W4(1,1)+W4(1,2)*O1+W4(1,3)*O2;
O5=W5(1,1)+W5(1,2)*O2+W5(1,3)*O3;
CeSor1=(y(j,1)-O4);
CeSor2=(y(j,1)-O5);
CeIn1=(W4(1,2)*CeSor1);
CeIn2=(W4(1,3)*CeSor1+W5(1,2)*CeSor2);
CeIn3=(W5(1,3)*CeSor2);

```

```

for(i=1:1:5)
    W1(1,i)=W1(1,i)+0.00000000640*CeIn1*entree(i,1);
    W2(1,i)=W2(1,i)+0.00000000833*CeIn2*entree(i,1);
    W3(1,i)=W3(1,i)+0.00000000518*CeIn3*entree(i,1);
end
for(i=1:1:3)
    W4(1,i)=W4(1,i)+0.0000000291*CeSor1*O12(i,1);
    W5(1,i)=W5(1,i)+0.000000000001*CeSor2*O23(i,1);
end
end
end
W1
W2
W3
W4
W5

Robot
function [ths]=robottp(entree)
%simulation du Robot deux degre de libertes
th1=entree(1);
thp1=entree(2);
to1=entree(3);
to2=entree(4);
th2=entree(5);
thp2=entree(6);
T=entree(7);
% initialisation des termes dynamiques
m1=15.91;
m2=11.36;
L1=0.432;
L2=0.432;
a1=3.82;
a2=2.12;
a3=0.71;
a4=81.82;
a5=24.06;

V1=1;
V3=1;
V2=0.5;
V4=0.5;
if (T <= 1.5 )
    m=10;
else
    m=10;
end
%Matrice d'inertie plus mjTJ
m11=a1+a2*cos(th2)+m*((L1)^2)+2*L2*L1*cos(th2)+m*((L2)^2);
m12=a3+(a2/2)*cos(th2)-m*L2*L1*cos(th2)+m*((L2)^2);
m21=a3+(a2/2)*cos(th2)+m*L2*L1*cos(th2)+m*((L2)^2);
m22=a3+m*((L2)^2);
%Matrice coriolis plus mjTjpont
cq1=-
(a2*sin(th2))*(thp1*thp2+(((thp2)^2)/2))-
thp2*m*L2*L1*sin(thp2)*(2*thp1+thp2);
cq2=(a2*sin(th2)*(((thp1)^2)/2))+m*L2*L1*(((thp1)^2)*sin(thp2));
%Matrice de gravitation
g1=a4*cos(th1)+a5*cos(th1+th2);
g2=a5*cos(th1+th2);
%Matrice de frottement
H1=V1*thp1+V2*sign(thp1);
H2=V3*thp2+V4*sign(thp2);
%invers de la Matrice d'inertie
detm=m11*m22-m21*m12;
h1=cq1+g1+H1;
h2=cq2+g2+H2;
% calcul des accelerations angulaires
ths1=(m22/detm)*(to1-h1)-(m12/detm)*(to2-h2);
ths2=-(m21/detm)*(to1-h1)+(m11/detm)*(to2-h2);
ths=[ths1 ths2];

Robot commande PD
function [ths]=robot23(entree)

```

```

% simulation du Robot deux degre de libertes
th1=entree(1);
thp1=entree(2);
to1=entree(3);
to2=entree(4);
th2=entree(5);
thp2=entree(6);
% initialisation des termes dynamiques
m1=15.91;
m2=11.36;
L1=0.432;
L2=0.432;
a1=3.82;
a2=2.12;
a3=0.71;
a4=81.82;
a5=24.06;
V1=1;
V3=1;
V2=0.5;
V4=0.5;
m=10;
% Matrice d'inertie
m11=a1+a2*cos(th2)+m*((L1)^2)+2*m*L2*L1*cos(th2)+m*((L2)^2);
m12=a3+(a2/2)*cos(th2)+m*L2*L1*cos(th2)+m*((L2)^2);
m21=a3+(a2/2)*cos(th2)+m*L2*L1*cos(th2)+m*((L2)^2);
m22=a3+m*((L2)^2);
% Matrice coriolis
cq1=-
(a2*sin(th2))*(thp1*thp2+(((thp2)^2)/2))-
thp2*m*L2*L1*sin(thp2)*(2*thp1+thp2);
cq2=(a2*sin(th2))*(((thp1)^2)/2)+m*L2*L1*
((thp1)^2)*sin(th2);
% Matrice de gravitation
g1=a4*cos(th1)+a5*cos(th1+th2);
g2=a5*cos(th1+th2);
% Matrice de frottement
H1=V1*thp1+V2*sign(thp1);
H2=V3*thp2+V4*sign(thp2);
% invers de la Matrice d'inertie
detm=m11*m22-m21*m12;

```

```

h1=cq1+g1+H1;
h2=cq2+g2+H2;
% calcul des accelerations angulaires
ths1=(m22/detm)*(to1-h1)-(m12/detm)*(to2-h2);
ths2=-(m21/detm)*(to1-h1)+(m11/detm)*(to2-h2);
ths=[ths1 ths2];

```

Générateur de trajectoire

- Pour θ_1

```

function y = fcn(T)
y = 2*cos(4*pi*T/3)+sin(2*pi*T/3);

```

- Pour θ_2

```

function
y2 = fcn(G)
y2 = 1-2*cos(4*pi*G/3)-sin(2*pi*G/3);

```

BIBLIOGRAPHIE

RÉFÉRENCES BIBLIOGRAPHIQUES

- [1] Jean – Jacques E .Slotine & Weiping LI « **Applied Nonlinear Control** ». Prentice Hall.1991.
- [2] Maurice Rivoire,Jean-Louis Ferrier, « **Cours d'Automatique, T1 0, T2 & T3**» , Chihab – Eyrolles 1994.
- [3] Wisama Khalil,Etienne Dombre , « **Modélisation identification et commande des robots** »,Hermes Science 1999.
- [4] SimMechanics user's,Copyright 2001-2002 ,by the Mathwork.
- [5] Tomas Wysser. «**Model – Based Desing of Control of a Robot Arm**»,Rapport de Simminaire,IEEE,Mai 2003.
- [6] Vania Andronova , «**Utilisation de données météo et des réseaux de neurones pour la prédiction de vitesses de vent** »,Projet de fin d'étude Master 2nd année,18 juillet 2006.
- [7] Marc Parizeau, «**Réseaux de neurones** », édition 2004.
- [8] Ben Krose , Patrick van der Smagt, «**An introduction to Neural Networks**»,édition 1996.
- [9] Mark W . Spong , Seth Hutchinson , and M. Vidyassagar , «**Robot Dynamics and Control** ». Second édition 2004.
- [10] A.Katbab , « **Nonlinear Adaptive Control of Robotic Manipulators-Hyperstability Approach** ». Robotics and Autonomous systems Vol(4). pp.265-273(1988).
- [11] Breton S, « **Simulation d'une scène robotique**». Guide utilisateur Matlab EEATROP-TR-94-06, *Groupe TROP, Faculté des Sciences et Techniques, Mulhouse, France*, 49 p., 1994
- [12] Coiffet P, « **Les robots. Interaction avec l'environnement, Tome 2** ». *Hermès*,1981
- [13] Coiffet P, « **La robotique. Principes et applications, Traité des Nouvelles Technologies** ». Série Robotique, *Hermès*, 1986
- [14] Corke P.I, « **A robotics TOOLBOX for Matlab** ». *IEEE Robotics & Automation Magazine*, Vol. 3 (1), pp. 24-32, 1996
- [15] Craig J.J, « **Introduction to robotics: mechanics and control** ». 2nd Edition,*Addison-Wesley Publishing Company*, 1989
- [16] Dixon L.C.W. « **Nonlinear Optimisation** ». *English Universities Press*, 1972
- [17] Ferretti G., Magnani G., Rocco P, « **Toward the implementation of hybrid position/force control in industrial robots** ». *IEEE Transactions on Robotics and*

Automation, Vol. 13 (6), pp. 838-845, 1997

[18] Hérault J., Jutten C, « **Réseaux neuronaux et traitement du signal** ». Traité des Nouvelles Technologies, Série Traitement du Signal, *Hermès*, 1994

[19] Paul R.P, « **Robot manipulators. Mathematics, programming, and control** ». *The MIT Press Series in Artificial Intelligence*, 1981

[20] Raibert M., Craig J, « **Hybrid position/force control of manipulators** ». *Trans. Of the ASME, Journal of Dynamic Systems, Measurement and Control*, Vol. 103(2), 1981

[21] Samson C, « **Une approche pour la synthèse et l'analyse de la commande des robots manipulateurs rigides** ». Rapport de Recherche N° 0669, *INRIA Rennes, France*, 1997.

[22] Alessandro De Luca et Bruno Siciliano. « **trajectory control of a non linear one-link flexible arm** ». 1989.

[23] S.cetinkunt , B.siciliano , et W.J.Book, « **symbolic modeling and dynamic analysis of flexible manipulators**». in Proc.1986 IEEE Int.Conf.syst.Man,cyber.Atlanta 1986.

[24] Philippe Coiffet « **La Robotique : Principes et Applications** ». Editions Hermes Paris 1992.

[25] P.I.Corke « **Robotic Toolbox For Matlab** ». 2002.

[26] Silété Alain Amouzou. « **Étude du contrôle de force-position avec modélisation des forces d'impact** ». Master's thesis, École Polytechnique de Montréal, April 1998.

[27] Stéphane Blouin. « **Étude de l'interaction entre un manipulateur et son environnement : Identification des paramètres et réglage du contrôleur** ». Master's thesis, École Polytechnique de Montréal, 1998.

[28] Romano M. De Santis. « **Asservissement En Position D'un Manipulateur Robotique, Notions et Techniques Fondamentales** ». École Polytechnique de Montréal, 1995.

[29] Stéphane Germain. « **Mise au point d'un banc d'essai pour l'étude théorique et expérimentale de robots flexibles** ». Master's thesis, École Polytechnique

de Montréal, 1996.

[30] [Bruyninckx et al., 2001] H. Bruyninckx et J. Shutter (2001). « **Introduction to intelligent robotics. Rapport technique** ». Katholieke Universiteit de Leuven.

[31] [Khalil, 2002] W. Khalil (2002). « **Commande des robots manipulateurs** ». Hermes Science - Lavoisier.

[32] [Canudas et al., 1997] C. Canudas, B. Siciliano, et G. Bastin (1997). « **Theory of Robot Control** ». Springer.

[33] [Gorla et al., 1984] B. Gorla et M. Renaud (1984). « **Modèles des Robots Manipulateurs application à leur commande** ». Cepadues Editions.

[34] [Nakamura, 1991] Y. Nakamura (1991). « **Advanced Robotics. Robotics and Optimization** ». Addison- Wesley Publishing Company.