

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

ECOLE NATIONALE POLYTECHNIQUE

DER Génie Electrique et Informatique  
DEPARTEMENT ELECTRONIQUE

**THESE DE MAGISTER**

**EN ELECTRONIQUE**

Option Acquisition et traitement de l'information

Thème

المكتبة — المكتبة  
BIBLIOTHEQUE — المكتبة  
Ecole Nationale Polytechnique

**ELABORATION DE STRATEGIES DE COMMANDE  
NEURO-DYNAMIQUE ADAPTATIVE  
POUR ROBOTS MANIPULATEURS.**

Etudié par : **KADRI MOHAMED**  
*Ingénieur d'Etat en Electronique, ENP.*

Soutenue le : 13/10/1998

Devant de jury composé de :

**Président**

Mr M.S. Boucherit      Maître de conférence      ENP

**Rapporteurs**

Mr N. Louam      Maître de conférence      ENP  
Mr B. Bouzouia      Docteur d'Etat, Chargé de recherche      CDTA

**Examineurs**

Mr C. Larbes      Ph.D.      ENP  
Mr M. Tadjine      Docteur d'Etat      ENP  
Mr R. Illoul      Chargé de Cours      ENP

من أجل تحقيق توافق جيد بين عدة تالقات مصنوية في العديد من الميادين خاصة منها الصناعية، تستمر الأبحاث حالياً في إعداد طرق تحكم منسقة للذراع الألي. هذه التالقات المفروضة خاصة من حيث دقة الموضع و سرعة التنفيذ جعلت من تقنيات التحكم البسيطة غير كافية وأدت إلى الحاجة إلى البحث عن تقنيات جديدة ملائمة. يهدف هذا العمل إلى إنشاء طريقة تحكم جديدة و فعالة تجمع بين التقنيات الثلاث ديناميكية عصبونية و تلاؤمية. التقنية المقترحة تلجأ إلى تفكيك معدلات لديناميكية لسرع و ثابت و متغير. يمكن استعمال ديناميكية لسرع في التحكم إما مباشرة أو غير مباشرة مما يؤدي إلى نوعين للتقنية المقترحة، كلتاهما تعتمد على استعمال الشبكات العصبونية لحساب المزدوجة المصنفة على مستوى كل صلة. شبكة عصبونية أخرى تقيم العناصر الديناميكية المتغيرة من أجل جعل التحكم صلب إزاء داءد تغيرات العناصر الديناميكية. نستعمل كذلك مصحح من نوع PD تلاؤمي لخفض أخطاء التتبع الناتجة عن أخطاء التقييم و قلة دقة العناصر الديناميكية و كذلك عن ديناميكية الذراع الغير مأخوذة بعين الإعتبار. تالقات فائقة لتقنية التحكم المقترحة ينتج عنها نتائج ممتازة العددية و التجريبية العمالية.

#### Abstract

To realize a good compromise between several performances required by diverse applications, the design of appropriate control laws for robots manipulators is still today a matter of intensive investigations. The required performances, related in particular to a high-speed and a good precision in task performing, have made conventional control methods obsolete. Hence, the main purpose in the present work is to derive an efficient control technique by combining the best characteristics of the following three methods, i.e dynamic, neural and adaptive controls, which are usually single employed. The approach is basically conceived on a decomposition of the robot dynamics model into two parts, fixed and variable, so that the model can be used to elaborate the required control, off or on-line. Thus, two possible solutions can be followed using the approach proposed. In both solutions the use of artificial neural networks (ANNs) is made to produce torque at each joint. Another ANN is still required to estimate the variable inertial parameters. To reduce the tracking error due to unmodeled dynamics, the use of a decentralized adaptive PD controller is made. High tracking performance from the proposed control approach are shown by the results of a numerical simulation and an experimental implementation.

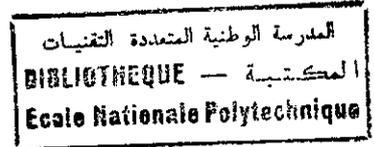
#### Résumé

Pour réaliser un bon compromis entre plusieurs performances exigées par diverses applications, notamment industrielles, l'élaboration de techniques de commande adéquates pour des bras manipulateurs continue de nos jours à faire l'objet d'études intensives de recherches. Les exigences de performances surtout en termes de précision en position et de rapidité d'exécution des tâches ont rendu en particulier les méthodes de commande conventionnelles insuffisantes et obsolètes. Le besoin de l'élaboration de nouvelles techniques bien appropriées à cet effet se fait de plus en plus sentir. L'objectif du présent travail consiste ainsi à concevoir une commande efficace par combinaison de la majorité des avantages offerts par trois techniques essayées souvent individuellement, à savoir les commandes dynamique, neurale et adaptative. L'approche adoptée à cet effet fait recours au préalable à une décomposition du modèle dynamique du bras manipulateur en deux parties, fixe et variable. Le modèle dynamique du bras dans ce cas peut être utilisé pour l'élaboration de la commande nécessaire hors ou en ligne. Ceci permet de dégager alors deux variantes possibles dans l'approche proposée. Les deux variantes utilisent des réseaux de neurones artificiels pour générer le couple dans chaque liaison. Dans l'estimation des paramètres dynamiques variables, un autre réseau de neurones est aussi employé afin de rendre la commande robuste vis à vis de la variation de ces paramètres.

Afin de réduire l'erreur de poursuite due aux erreurs d'estimation, aux imprécisions sur les paramètres inertiels et aux dynamiques non modélisées, une correction de retour de type PD adaptative décentralisée a été utilisée.

Les résultats de la simulation numérique et de l'implémentation expérimentale effectuées, ont montré que l'approche proposée dans le cadre de cette étude, permet de conférer au bras manipulateur considéré des performances dynamiques très élevées.

## REMERCIEMENTS



Ce travail a été effectué au sien de l'équipe de cellule flexible de production du Laboratoire Robotique et Intelligence Artificielle du Centre de Développement des Technologies Avancées (CDTA). Je remercie très vivement mes deux directeurs de thèse Monsieur BRAHIM BOUZOUIA, Chargé de recherche au CDTA, et Monsieur NADJIB LOUAM, Maître de Conférence à l'ENP de m'avoir donné la chance de travailler avec eux, de leurs aides et de leurs conseils qui ont permis de mener cette thèse à son aboutissement. Je tiens à exprimer toute ma gratitude à mes directeurs de thèse pour leurs qualités humaines et leurs connaissances scientifiques que j'ai appréciées.

Je tiens à remercier Monsieur M.S. BOUCHERIT Maître de Conférence à l'ENP pour l'intérêt qu'il porte à ce travail et pour l'honneur d'avoir présidé le Jury de ma soutenance.

Ma gratitude revient aussi à chacun des membres du Jury qui m'ont honoré de leur participation pour la soutenance de cette thèse.

Je tiens à exprimer ma profonde gratitude à chaque membre de l'équipe de la cellule flexible de production et en particulier Monsieur A. SOUILAH, pour leurs aides et leurs valeurs humaines.

Finalement je remercie tous ceux qui ont contribué de près ou de loin à l'aboutissement de ce travail.

المدرسة الوطنية المتعددة التقنيات  
BIBLIOTHEQUE — المكتبة  
Ecole Nationale Polytechnique

*A mes Parents.  
A toute ma Famille.  
A Djamel.  
A tous ceux que j'aime.*

# TABLE DES MATIERES

**INTRODUCTION** 1

**CHAPITRE I** 5  
**FORMULATION DU PROBLEME ET ETAT DE L'ART**

**CHAPITRE II**  
**DYNAMIQUE DES ROBOTS MANIPULATEURS**

II.1 Structure d'un Robot Manipulateur	15
II.2 Modélisation Géométrique	15
II.2.1 Modèle Géométrique Direct et Inverse	15
II.2.2 Modèle Différentiel Direct et Inverse	16
II.3 Modélisation Dynamique	17
II.3.1 Modèle Dynamique	17
II.3.2 Modèle des Actionneurs	18
II.3.3 Modèle Dynamique du Robot Manipulateur Avec Actionneurs	19
II.3.4 Modélisation des Frottements	20
II.4 Propriétés du Modèle Dynamique	22

**CHAPITRE III**  
**COMMANDE NEURO-DYNAMIQUE ADAPTATIVE**

III.1 Décomposition du Modèle Dynamique	25
III.2 Commande Neurale Dynamique Adaptative	26
III.3 Deux Méthodes de Mise en Œuvre de la Commande Neuro-Dynamique Adaptative	29
III.3.1 Commande Neurale Dynamique Anticipée Adaptative	29
III.3.2 Commande Neurale par Couple Calculé Adaptative	30
III.3.3 Estimation des Paramètres Variables	32
III.3.4 Apprentissage des Réseaux de Neurones	33
III.3.4 Correction PD Adaptative	38

**CHAPITRE IV**  
**SIMULATION**

IV.1 Modèle Dynamique du Robot	42
IV.2 Etude de l'influence des Termes Dynamiques	43

IV.2.1 Etude de l'influence de la Variation de la Masse de la Charge	43
IV.2.2 Etude de l'influence des Frottements	50
IV.3 Commande Neuro-Dynamique Adaptative	57
IV.3.1 Condition de Simulation	57
IV.3.2 Commande Neurale Dynamique Anticipée Adaptative	58
IV.3.2 Commande Neurale Adaptative par Calcul du Couple Généralisé	66
IV.4 Etude Comparative	72

## **CHAPITRE V**

### **IMPLEMENTATION SUR LE ROBOT RP41 DU CDTA**

V.1 Description du Site du RP41	79
V.2 Modèle Dynamique du RP41	81
V.3 Identification du Robot RP41	84
V.3.1 Identifiabilité des Paramètres Dynamiques	84
V.3.2 Modèles d'identification	85
V.3.3 Identifications en Ligne et Hors Ligne	86
V.3.4 Choix de l'excitation	86
V.3.5 Procédure d'Identification	87
V.3.6 Résultats Expérimentaux et Discussions	87
V.4 Mise en Œuvre Experimentale de la Commande Neuro-Dynamique Adaptative	94
V.4.1 Commande Neurale Dynamique Anticipée Adaptative	95
V.4.2 Commande Neurale par Couple Calculé Adaptative	95
V.4.3 Résultats	96

## **CONCLUSION ET PERSPECTIVES** 103

## **BIBLIOGRAPHIE** 106

<b>ANNEXES</b>	<b>112</b>
<i>Annexe A</i>	<b>113</b>
<i>Annexe B</i>	<b>116</b>
<i>Annexe C</i>	<b>120</b>

المدرسة الوطنية المتعددة التقنيات  
BIBLIOTHEQUE — المكتبة  
Ecole Nationale Polytechnique

## ***INTRODUCTION***

---

---

Durant les dernières années, l'utilisation des robots manipulateurs a connu une importance croissante dans divers domaines. Les aptitudes recherchées pour ces robots manipulateurs dans les différentes applications sont surtout la précision et la rapidité. Cependant réaliser une des aptitudes souvent diminue l'autre et la question de commande des robots manipulateurs est réaliser un bon compromis entre les différentes performances requises.

La dynamique d'un robot manipulateur à plusieurs liaisons est en fait caractérisée par un ensemble d'équations différentielles du second ordre, couplées et non linéaires. Ces équations sont en général entachées d'incertitudes sur les paramètres décrivant les propriétés dynamiques de l'ensemble charge + corps terminal. La complexité de cette dynamique va amplifier le problème du compromis entre les différentes performances. Pour traiter ce problème plusieurs approches de commande ont été considérées. Dans la littérature les toutes premières approches proposées sont basées sur l'utilisation de correcteurs de conception simple tels que les correcteurs PID, etc... convenant plutôt pour des systèmes linéaires. L'inefficacité de ces correcteurs peut être facilement vérifiée quand le robot essaie d'effectuer des tâches de précision et/ou de rapidité.

Les récentes approches de commande sont basées sur des lois utilisant le modèle dynamique dans l'élaboration de la loi de commande. Ces approches paraissent intéressantes, cependant pour l'obtention de bonnes performances elles exigent une connaissance plus précise du comportement dynamique du robot qui en plus de son caractère non linéaire, présent souvent des contraintes techniques comme :

- imprécision sur les paramètres inertiels et géométriques, en rapport avec les méthodes d'obtention de ces paramètres.
- négligence des variations de la charge et de quelques paramètres inertiels.
- imprécision de la dynamique non modélisée.

Dans ce cadre la commande adaptative pourrait être une solution intéressante vue qu'elle n'exige pas une modélisation exacte de la dynamique du robot. Cette commande est généralement basée sur des modèles linéaires à paramètres variables. Ces derniers sont estimés en ligne et leur nombre augmente d'une manière indésirable avec le nombre de liaisons du robot. A petites vitesses d'exécution la commande adaptative fournit de bonnes performances, Cependant dès qu'on passe à de grandes vitesses, la variation des paramètres devient trop rapide pour être compensée par la commande. Ceci provoque nécessairement une dégradation dans les performances du système.

L'approche proposée dans le présent travail repose sur l'utilisation du modèle dynamique du robot. Une connaissance même partielle du comportement dynamique du robot est à cet effet nécessaire. La loi de commande recherchée découlera d'une décomposition du modèle dynamique en deux parties, fixe et variable. La partie fixe fournit une commande basée sur les paramètres inertiels fixes. Cette partie rejoint l'approche dynamique citée précédemment. Face aux perturbations induites par la variation des paramètres dynamiques, la partie variable compense ces perturbations par le biais d'une poursuite des variations paramétriques. Cette approche de commande exige un temps de calcul important et présente une certaine complexité dans l'implémentation. Notant par ailleurs que l'approche de commande neurale est apparue ces dernières années comme une solution pour les problèmes de commande des robots où les méthodes du calcul conventionnel se sont avérées inefficaces. Ainsi, pour surmonter les problèmes de temps de calcul et de complexité, la partie dynamique de notre approche intègre des blocs de réseaux de neurones représentant d'un côté certaines parties de la dynamique du robot et d'un autre côté ils jouent le rôle de la fonction d'estimation paramétrique. Néanmoins certaines dynamiques ne peuvent pas être modélisées, et par conséquent elles ne peuvent pas être compensées par une loi de commande. Afin de surmonter ce problème et réduire aussi l'erreur de poursuite due aux autres perturbations, une correction P.D. adaptative est ajoutée.

Finalement L'approche de commande proposée dans ce travail peut être ainsi considérée comme une commande Neuro-Dynamique Adaptative.

Le présent document est organisé comme suit :

**Le premier chapitre** présente l'état de l'art de la commande des robots manipulateurs. Après la formulation du problème de commande nous exposons une synthèse des différents travaux entrepris dans ce cadre. Les avantages et inconvénients des approches étudiées sont mis en évidence.

**Le second chapitre** traite de l'aspect dynamique des robots manipulateurs. Nous nous intéresserons plus spécialement au modèle dynamique du robot comprenant le bras manipulateur et ses différents actionneurs.

**Le troisième chapitre** présente dans les détails l'approche de la commande neurale dynamique adaptative proposée. La commande sera étudiée selon ses deux variantes.

**Le quatrième chapitre** traite des résultats obtenus de la simulation par l'emploi du software Matlab sur PC. Après présentation de l'influence des termes dynamiques sur le

comportement du robot, nous exposerons les résultats de la simulation de la commande proposée avec ses deux variantes. Ces résultats seront comparés aux résultats obtenus déjà par la Méthode "Computed Torque" proposée par Ozaki (Ozaki et al., 1991).

**Le cinquième chapitre** est consacré à l'implémentation effective de la commande proposée au robot RP41 du CDTA. Cette implémentation est précédée par une modélisation dynamique ainsi que d'une identification des paramètres dynamiques du robot concerné. Le traitement des données expérimentales est étudié en détails. Les résultats de l'expérimentation pour l'identification et la commande y seront exposés et analysés.

Enfin, une conclusion générale et des perspectives d'investigation sont données à la fin de ce document.

***CHAPITRE I***

***FORMULATION DU PROBLEME ET  
ETAT DE L'ART***

---

---

Le problème de la commande des robots manipulateurs consiste en général à déterminer une séquence d'actions appropriées pour faire déplacer le bras du robot d'une position à une autre. Cette commande est à concevoir sous l'effet de plusieurs contraintes liées à l'environnement du robot manipulateur, à la nature non linéaire du système, aux variations paramétriques et aux dynamiques non modélisées.

Les exigences surtout en termes de précision en position et de rapidité d'exécution des tâches, rendent les commandes conventionnelles insuffisantes dans ce cadre, et font ressentir davantage le besoin de l'élaboration d'une nouvelle technique appropriée à cet effet.

Plusieurs schémas de commande ont été envisagés ces dernières années pour résoudre certains problèmes de la commande des robots manipulateurs (Paul, 1981 ; Hsia, 1986 ; Kawato et al., 1988 ; Leahy et al., 1991). Initialement la commande des robots manipulateurs reposait essentiellement sur les techniques linéaires. Les articulations sont asservies en général d'une manière indépendante au moyen de correcteurs linéaires. La simplicité de mises en œuvre de ces techniques linéaires avec des coûts raisonnables a permis une large utilisation de ces dernières procédures au niveau des robots industriels. Toutefois ces techniques ou méthodes linéaires de commande se sont avérées insuffisantes lorsque le robot manipulateur devrait effectuer des déplacements rapides et précis (Lu et al., 1993).

La recherche de méthodes adéquates pouvant résoudre les problèmes cités ci-dessus a donné naissance à plusieurs approches de commande, parmi lesquelles on cite :

- la Commande Dynamique.
- la Commande Adaptative.
- la Commande Neurale.

La commande dite dynamique (Figure I.1) est celle qui correspond à l'approche qui semble la plus directe dans le cas de la commande des robots, du fait qu'elle est élaborée, à la base, sur le modèle dynamique du robot manipulateur (Paul, 1981 ; Craig, 1986). Par conséquent la loi de commande associée à cette méthode n'est autre que le modèle dynamique inverse. Ce dernier modèle permet de calculer les couples généralisés qui permettent la poursuite de la trajectoire de référence (An et al., 1989 ; Lu et al., 1993). Cette technique est appelée méthode de calcul de l'effort généralisé, connue en anglais sous le nom de "computed torque method".

Cette commande est d'autant plus performante, en termes de précision et de vitesse d'exécution, que le modèle dynamique du robot manipulateur considéré est le plus exact et précis. Malheureusement, en pratique, il est difficile de modéliser le robot avec une grande précision puisque certains de ses paramètres dynamiques dépendent même de la configuration du robot au moment de manipulation d'objets. Par conséquent, la modélisation imparfaite et les incertitudes sur les paramètres dynamiques peuvent aboutir inévitablement à de mauvaises performances surtout en termes de poursuite des trajectoires (Egeland, 1986, Lu et al., 1993).

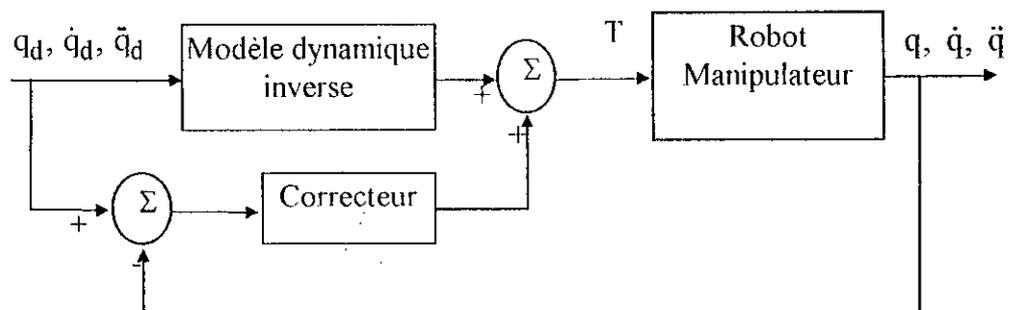


Figure I.1. Commande dynamique.

Afin d'éviter le besoin d'une modélisation complète et précise du robot manipulateur d'une part et de réduire la sensibilité de la commande aux variations paramétriques, d'autre part, l'approche de la commande adaptative (Figure I.2) semblait être la solution la mieux appropriée à cet effet (Landau, 1979 ; Hsia, 1986 ; Seraji, 1989). La plupart des approches adaptatives sont basées sur l'approximation du modèle dynamique par un modèle linéaire et déterminent la commande sur la base d'informations obtenues en ligne sur les sorties et/ou les variables d'état du robot (Tosunoglu and Tesar, 1987). L'application des techniques de commande adaptatives, telles que celles avec modèle de référence et la commande auto-ajustable, aux robots manipulateurs a été étudiée par plusieurs auteurs (Hsia, 1986 ; Tosunoglu and Tesar, 1987). Du fait que ces techniques adaptatives n'exigent pas une connaissance du modèle dynamique du robot, elles manifestent une robustesse aux variations paramétriques plus grande que celle due à la commande dynamique (Hsia, 1986).

Cependant, le problème majeur des techniques de la commande adaptative reste leur complexité numérique qui rend leur implémentation difficile et coûteuse. Aussi, de par leur nature, ces techniques prennent un temps de calcul considérable dans leurs tâches d'identification et d'adaptation paramétrique, ce qui les rend non efficaces, notamment quand le robot opère à grande vitesse (Miller et al., 1990 ; Ozaki et al., 1991). A noter que

l'identification des paramètres en temps réel et la sensibilité de la précision numérique peuvent engendrer une augmentation du temps de calcul d'une manière très importante surtout quand le nombre de liaisons du robot est élevé (Miller et al., 1990 ; Yegerlehner and Meckl, 1993).

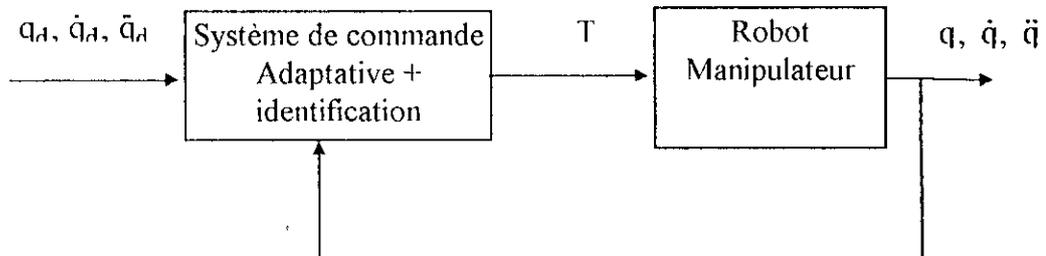


Figure I.2. Commande adaptative.

Durant la dernière décennie, l'approche neurale est proposée comme une solution aux tâches qui exigent une masse importante de calcul, c.-à-d. là où plusieurs approches conventionnelles se sont révélées inefficaces (Lippmann, 1987 ; Bavarian, 1988 ; Widrow and Lehr, 1990 ; Cichocki and Unberhauen, 1993). L'efficacité des réseaux de neurones est due essentiellement aux caractéristiques suivantes (Narendra and Parthasarathy, 1990 ; Krijgsman et al, 1992 ; Jervis, 1993) :

- leur capacité à approximer les fonctions non linéaires,
- leur faculté d'être entraînés pour reproduire des exemples appris,
- ils peuvent généraliser sur d'autres exemples non appris,
- ils peuvent opérer sur des données à valeurs réelles,
- l'information est transformée en une représentation interne (poids) qui tient compte des données quantitatives et qualitatives du signal d'apprentissage,
- leur structure est convenable à un traitement parallèle.

La possibilité d'utilisation des réseaux de neurones dans l'identification et la commande des systèmes dynamiques non linéaires a été largement prouvée dans la littérature (Psaltis et al. 1987 ; Psaltis et al. 1988 ; Narendra and Parthasarathy, 1990 ; Hunt and Sbarbaro, 1992 ; Billing and Chen, 1992 ; Burrows and Niranjana, 1993).

La conception d'une commande à l'aide de réseaux de neurones s'effectue par l'entraînement de ces derniers sur des données entrées permettant l'apprentissage de la dynamique inverse du robot manipulateur. Notez que dans le contexte de la commande à l'aide de réseaux de neurones la notion d'apprentissage de ces derniers est souvent mal distinguée de celle de l'adaptation. Selon Yabuta ( Yabuta and Yamada , 1992) la notion

l'adaptation poursuit la trajectoire en un seul processus de commande afin de s'adapter aux variations de la dynamique du robot. Par contre la commande par réseaux de neurones arrive par le biais de l'apprentissage à modéliser la dynamique du robot après plusieurs essais. Ainsi le réseau de neurone apprend le comportement du robot durant l'apprentissage, à travers des exemples.

Pour effectuer l'étape d'entraînement des réseaux de neurones, plusieurs architectures d'apprentissage pour la commande neurale ont été proposées par Psaltis (Psaltis et al., 1987 ; Psaltis et al., 1988) et Kawato (Kawato et al., 1988). Ces architectures d'apprentissage ont permis d'avoir une commande neurale qui donne la dynamique inverse du robot manipulateur. L'apprentissage s'effectue en utilisant l'algorithme de rétropropagation introduit par Rumelhart and McClelland (Lippmann, 1987 ; Widrow and Lehr, 1990 ; Cichocki and Unberhauen, 1993).

Afin de tirer profit des capacités des réseaux de neurones à contourner les imperfections des approches de commande dynamique et adaptative, la recherche s'est orientée vers la combinaison de l'approche neurale à ces dernières techniques. Cette manière de combiner ces approches de commande a abouti à deux types de techniques :

- Commande Neuro-Dynamique,
- Commande Neuro-Adaptative.

En ce basant sur des travaux de Kawato et dans le but de développer l'approche neuro-dynamique, Newton and Xu ont présenté une approche neurale pour l'apprentissage en ligne de la dynamique inverse du robot (Newton and Xu, 1993). Ils ont ainsi abouti à une implémentation en temps réel de la commande neuro-dynamique pour un robot manipulateur flexible, et cela par une mise à jour des poids des réseaux de neurones en utilisant l'erreur de poursuite, ceci a permis à la commande de compenser la variation de la dynamique du robot. Cependant au lieu d'utiliser un seul réseau pour tout le système, ils ont proposé d'associer dans cette approche un réseau à chaque articulation ce qui a conduit à une commande décentralisée. A noter qu'une telle approche de commande ne peut pas compenser les erreurs dues à la variation de la charge et aux dynamiques non modélisées, surtout si la variation de la charge est très rapide (Leahy et al., 1991). D'un autre côté il faut un entraînement hors ligne des réseaux de neurones pour éviter l'instabilité qui peut être induite au départ avec l'initialisation des poids par des données aléatoires.

Afin de surmonter les incertitudes dues aux dynamiques non modélisées, Ozaki et al ont proposé l'utilisation de deux réseaux de neurones pour représenter la dynamique inverse du robot manipulateur (Ozaki et al, 1991). Après avoir entraîné ces réseaux hors

ligne, ils utilisent une mise à jour en ligne des poids des deux réseaux de neurones pour réduire l'erreur de poursuite due aux perturbations causées par la dynamique non modélisée. Plus tard, et afin d'améliorer les performances de la commande proposée dans (Ozaki et al, 1991), Jung and Hsia ont introduit un autre bloc de réseau de neurones en amont du bloc de commande dans (Ozaki et al, 1991) comme un préfiltrage agissant sur l'erreur de poursuite. Ainsi au lieu que le bloc de commande reçoit la trajectoire désirée, on lui introduit une trajectoire combinée. Le but de cette combinaison est d'inverser la distorsion correspondant à l'écart entre la trajectoire désirée et la trajectoire réelle, ensuite l'introduire dans la commande afin de réduire l'erreur de poursuite (Jung and Hsia, 1995). La difficulté dans l'implémentation des deux techniques précédentes (Ozaki et al, 1991, Jung and Hsia, 1995) réside, d'un côté dans le temps de calcul important exigé pour la mise à jour des poids des réseaux, réalisée par l'algorithme d'adaptation, d'un autre côté, ce type de commande reste robuste uniquement pour de faibles erreurs de modélisation, et ne peut pas compenser une grande perturbation comme celle causée par la variation de la masse de la charge prise par le robot (Johnson and Leahy, 1990).

Afin de prendre en considération les variations paramétriques dans la commande dynamique, Johnson a proposé une commande neuro-dynamique utilisant le modèle dynamique du robot qui inclut les variations de la masse de la charge (Johnson and Leahy, 1990 ; Leahy et al., 1991). L'approche consiste à estimer la variation de la masse de la charge sur la base de l'erreur de poursuite en position et en vitesse en utilisant un réseau de neurones. La trajectoire de l'erreur est subdivisée en plusieurs intervalles; chaque intervalle correspond à une classe de la masse de la charge. Le réseau de neurones, qui est entraîné en utilisant l'algorithme de rétropropagation, fournit la masse estimée de la charge dans chaque intervalle. A partir de là, le calcul en ligne du couple d'anticipation est combiné à l'estimation de la masse de la charge permettant l'adaptation de la commande en vu de la compensation de la variation de la dynamique.

L'approche de commande proposée conduit à de bons résultats, mais néglige aussi autres sources de perturbation telles que la dynamique non modélisée et le problème des frottements qu'une simple compensation PD à gains fixes ne peut pas surmonter. A noter que l'estimation de la variation de la masse de la charge, basée sur l'erreur de poursuite, fournit à la commande la capacité de poursuivre la variation de la dynamique du robot manipulateur (Johnson and Leahy, 1990). Cependant cette solution peut aussi induire l'estimateur de la masse de la charge en erreur, car on peut obtenir par cette méthode, non pas la valeur de la masse de la charge mais la valeur d'un paramètre qui englobe toutes les perturbations. Aussi et dans le but de prendre en charge les effets de la variation de la masse de la charge, Yegerlehner and Meckl ont proposé une commande neuro-dynamique

capable de compenser les perturbations causées par cette variation (Yegerlehner and Meckl, 1993). Cette approche est basée sur l'utilisation du modèle dynamique du robot manipulateur en considérant la masse de la charge comme paramètre variable. Deux différents réseaux ont été utilisés ; l'un pour générer la commande qui correspond à la dynamique inverse du robot et l'autre pour estimer la masse de la charge du robot. Le premier réseau de neurones représente une commande dynamique conventionnelle. Les entrées de ce réseau sont les trajectoires de position, vitesse et accélération désirées ainsi que la masse estimée de la charge. Le deuxième réseau de neurones, qui se base sur les trajectoires réelles de position, vitesse et accélération ainsi que sur les couples exercés à chaque articulation, permet d'estimer la masse la charge. L'apprentissage des deux réseaux de neurones utilisant l'algorithme de la backpropagation est basé sur les architectures d'apprentissage suggérées dans les travaux de Psaltis et Kawato.

Dans le cadre de cette approche de commande, Yegerlehner and Meckl ont proposé une solution au problème de la variation de la masse de la charge. Mais avec l'inconvénient d'avoir utilisé un seul réseau de neurones pour le calcul du couple généralisé, c.-à-d. un réseau de taille nécessairement colossale qui nécessite un temps très long pour le calcul du couple. L'implémentation de cette approche est très problématique en temps réel et notamment à grande vitesse d'exécution. D'un autre côté cette approche de commande utilise uniquement une compensation proportionnelle et dérivé à gains fixes pour venir à bout des perturbations causées par les frottements et la dynamique non modélisée. La compensation employée n'est pas entièrement efficace face à ces perturbations (Ozaki et al, 1991).

L'idée d'une approche de commande neuro-adaptative est due d'une part à la similitude que présente la conception de l'approche neurale avec l'approche adaptative. D'autre part, c'est que les réseaux de neurones permettent de mieux réduire la taille des calculs, et de préserver des performances acceptables dans un intervalle d'incertitude où la commande adaptative ne peut pas agir convenablement (Bavarian, 1988).

Pour étudier l'approche de la commande neuro-adaptative Zalzal et Morris (1989) ont développé une technique adaptative basée sur des réseaux de neurones envisagée dans le cas du robot PUMA 560. Ils ont montré en particulier que le concept d'apprentissage des réseaux de neurones permet de donner à la commande une grande capacité à s'adapter au mieux aux changements dans l'environnement de travail du robot manipulateur, et aux incertitudes dues aux imprécisions sur les valeurs des paramètres inertiels dans le modèle du robot. Toutefois l'inconvénient au niveau de la mise en œuvre d'un temps de calcul

considérable et nécessaire pour l'adaptation des poids, demeure aussi posé dans cette approche.

Afin de surmonter l'importance du temps de calcul que confronte les techniques adaptatives, Saad a proposé une approche de commande neuro-adaptative basée sur l'habilité des réseaux de neurones à poursuivre la dynamique des robots manipulateurs, sans avoir recours à l'estimation des paramètres variables du modèle dynamique. Cette approche a été comparée aux commandes adaptatives directe et indirecte en utilisant des schémas de commande basés sur le modèle paramétrique. L'approche neuro-adaptative étudiée fournit de bons résultats par rapport à l'approche adaptative dans la poursuite de trajectoires. Cependant l'habilité des réseaux de neurones, entraînés hors ligne dans cette approche, demeure insuffisante pour fournir de bonnes performances en ligne et surtout en présence de perturbations importantes telle qu'une grande variation de la masse de la charge (Johnson and Leahy, 1990 ; Leahy et al., 1991).

En résumé, l'approche de la commande dite dynamique d'un bras manipulateur semble être la méthode la plus directe et recommandée du fait que son expression représente la dynamique inverse du robot manipulateur. Cependant elle résoud mal les problèmes de variation de la charge, d'imprécision du modèle dynamique et des dynamiques non modélisées.

L'approche adaptative répond mieux au problème de la modélisation mais elle ne peut pas fournir de bonnes performances quand le robot opère à grandes vitesses. Aussi elle n'est pas efficace vis à vis des incertitudes dues à certaines non linéarités du modèle du robot manipulateur.

Dans la représentation des systèmes non linéaires et en particulier ceux de la commande des robots manipulateurs, l'approche utilisant les réseaux de neurones semble prendre de l'avance devant les deux techniques citées précédemment.

Toutefois, de nos jours la recherche de solutions adéquates au problème de la commande des bras manipulateurs s'oriente de plus en plus vers la combinaison de méthodes différentes pour bénéficier du maximum d'avantages offerts par chacune d'elles. Ainsi et pour tirer le maximum de profit des avantages offerts par les trois approches de commande discutées ci-dessus, c.-à-d. dynamique, adaptative et neurale, une solution combinée de ces trois techniques est proposée. En effet l'objectif du présent travail consiste à élaborer une commande de type neurale dynamique adaptative qui serait en particulier capable de prendre en compte la compensation :

- des non-linéarités et les différents couplages dans le robot manipulateur,
- de la variation des paramètres dynamiques,
- le cas des incertitudes liées aux dynamiques non modélisées,
- et le cas des incertitudes liées aux erreurs d'estimation des paramètres du modèle.

A cet effet, le modèle dynamique du robot manipulateur est décomposé en deux parties, une partie fixe, basée sur la connaissance partielle de la dynamique du robot et une partie variable basée sur l'estimation des paramètres variables. Ce bloc de commande d'anticipation est basé sur l'utilisation des réseaux de neurones. Une rétroaction adaptative est prévue en vue de compenser les erreurs de poursuite dues aux perturbations causées par les incertitudes sur les paramètres dynamiques ainsi que les dynamiques non prises en compte par le modèle dynamique.

***CHAPITRE II***

***DYNAMIQUE DES ROBOTS***

***MANIPULATEURS***

---

---

## II.1 Structure D'un Robot Manipulateur

Un robot manipulateur est un système mécanique qui peut être formé de plusieurs corps  $C_i$  liés les uns aux autres par des liaisons ( $L_i$ ) autorisant un seul degré de liberté comme montrer la figure II.1. La liaison peut être de translation (prismatique) ou de rotation (rotoïde). L'ensemble repose sur un socle et forme une chaîne cinématique ouverte terminée par un organe dit de préhension (organe terminal).

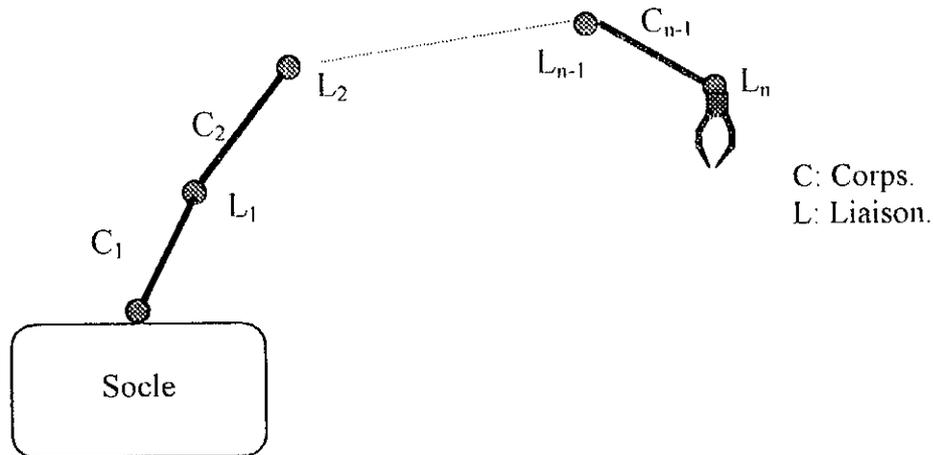


Figure II.1 Structure générale d'un robot manipulateur.

## II.2 Modélisations Géométriques

Le robot manipulateur est conçu pour travailler dans l'espace opérationnel à trois dimensions. Cet espace définit la notion de situation (position et orientation) de l'organe terminal du robot manipulateur. Dans cet espace est définie la configuration du robot manipulateur, c'est-à-dire la position et l'orientation des divers corps liés et qui constituent le robot manipulateur. Cependant l'action sur les liaisons se fait par les actionneurs localisés au niveau des liaisons, c'est-à-dire que la commande se fait dans l'espace généralisé. Pour permettre le passage d'un espace à un autre plusieurs transformations géométriques ont été développées (Coiffet, 1981 ; Craig, 1986).

### II.2.1 Modèle Géométrique Direct et Inverse

Le modèle géométrique direct permet de connaître la situation de l'organe terminal du robot en fonction de la configuration de ses articulations. La situation de l'organe terminal est décrite par les coordonnées opérationnelles, par contre la configuration des articulations est décrite par les variables articulaires (généralisées) :

$$x = f(q)$$

$x$  : désigne le vecteur colonne (dimension  $m \times 1$ ) des coordonnées opérationnelles,

$q$  : désigne le vecteur colonne (dimension  $n \times 1$ ) des coordonnées généralisées.

La fonction  $f$  est en générale non linéaire. Remarquons que le terme de coordonnée généralisée permet de tenir compte, à la fois, des liaisons rotoïdes (la coordonnée généralisée correspondante représente un angle) et prismatiques (la coordonnée généralisée correspondante représente une élongation).

Au moment de la conception ou de la synthèse d'une commande d'un robot manipulateur, le problème étant de déterminer les coordonnées généralisées qui correspondent à des coordonnées opérationnelles données. Résoudre ce problème consiste à calculer une fonction réciproque  $f^{-1}$ , si elle existe. La difficulté majeure de cette résolution réside dans le fait que la fonction  $f$  est non linéaire.

Le modèle géométrique inverse fournit les coordonnées généralisées en fonction des coordonnées opérationnelles.

$$q = g(x)$$

avec  $g = f^{-1}$

## II.2.2 Modèle Différentiel Direct et Inverse

Le modèle différentiel direct d'un robot manipulateur est le modèle qui permet de calculer la différentielle  $dx$ , des coordonnées opérationnelles, en fonction de la différentielle  $dq$ , des coordonnées généralisées. Il fait intervenir une matrice  $J(q)$ , fonction de la configuration du robot manipulateur, dénommée matrice jacobienne du robot manipulateur :

$$dx = J(q)dq$$

Il est à noter que lorsque l'on calcule les dérivées par rapport au temps (et non plus des différentielles) des coordonnées opérationnelles et généralisées  $\dot{x}$  et  $\dot{q}$  on obtient le modèle cinématique direct du robot manipulateur :

$$\dot{x} = J(q)\dot{q}$$

Le modèle différentiel inverse d'un robot manipulateur est le modèle qui permet de calculer la différentielle  $dq$ , si elle existe, des coordonnées généralisées, qui correspond à la différentielle  $dx$  imposée aux coordonnées opérationnelles à partir d'une situation

imposée à  $x$ . Le calcul est fonction de la configuration dans laquelle se trouve le robot manipulateur. Comme en général, plusieurs configurations permettent d'obtenir la même situation  $x$  de l'organe terminal, un modèle différentiel inverse est alors associé à chacune de ces configuration.

$$dq = J^\#(q)dx$$

Où  $J^\#$  représente la pseudo-inverse.

## II.3 Modélisation Dynamique

### II.3.1 Modèle Dynamique

Le modèle d'un bras manipulateur en action fait appel à plusieurs notions mathématiques précises qui peuvent être déduites par plusieurs méthodes (Graig, 1986). Ces dernières ont fait l'objet d'études détaillées dans (Wanpler, 1985 ; Graig, 1986 ; Schilling, 1990). De celles-ci nous retenons en particulier que l'équation de mouvement de la  $i^{\text{ème}}$  liaison d'un robot manipulateur, décrit par la figure II.1, par l'expression suivante :

$$\tau^i = M_{ii}(q) \ddot{q}^i + \sum_{\substack{j=1 \\ j \neq i}}^n M_{ij}(q) \ddot{q}^j + \sum_{j=1}^n \sum_{k=1}^n C_{jk}^i(q) \dot{q}^k \dot{q}^j + G^i(q) \quad (\text{II.1})$$

Pour  $i = 1, \dots, n$ , où  $n$  représente le nombre de liaison du robot.

Avec :

$\tau^i$  : représente la force (pour une liaison prismatique) ou le couple (pour une liaison rotoïde) exercé à la  $i^{\text{ème}}$  liaison.

$M_{ii}$  : Masse dans le cas d'une liaison prismatique ou moment d'inertie propre de la  $i^{\text{ème}}$  liaison pour une liaison rotoïde.

$M_{ij}$  : Moment d'inertie de couplage (Cross-Inertia) de la  $i^{\text{ème}}$  liaison avec les autres liaisons.

$C_{jk}^i$  : Coefficients centrifuges et de coriolis à la  $i^{\text{ème}}$  liaison.

$G^i$  : L'effet de la pesanteur.

$q^i$  : Angle ou élongation de la  $i^{\text{ème}}$  liaison selon qu'elle est rotoïde ou prismatique respectivement.

$\dot{q}^i, \ddot{q}^i$  : vitesse et accélération de la  $i^{\text{ème}}$  liaison.

Les termes :  $M_{ii}(q) \ddot{q}^i + \sum_{\substack{j=1 \\ j \neq i}}^n M_{ij}(q) \ddot{q}^j$  représentent l'effort d'inertie

d'accélération.

$M_{ii}(q) \ddot{q}^i$  l'effort d'inertie d'accélération du robot en mouvement autour de la  $i^{\text{ème}}$  liaison,

$\sum_{\substack{j=1 \\ j \neq i}}^n M_{ij}(q) \ddot{q}^j$  représentent l'influence des accélérations des autres liaisons en mouvement sur la  $i^{\text{ème}}$  liaison.

Les termes :  $\sum_{j=1}^n \sum_{k=1}^n C_{jk}^i(q) \dot{q}^k \dot{q}^j$  représentent les efforts d'inertie centrifuge et de coriolis.

### II.3.2 Modèle des Actionneurs

Dans ce travail, nous considérons le cas d'actionneurs électriques, et plus précisément les moteurs à courant continu commandés en tension. Ce choix est fait pour des considérations de mise en œuvre expérimentales.

Dans ce qui suit l'actionneur désigne l'ensemble (Amplificateur + Actionneur + Réducteur). Il est à noter que certains actionneurs ne possèdent pas de réduction, ils sont appelés actionneurs à commande directe connus en Anglais sous le nom de direct-drive actuators. Dans le cadre du présent travail l'actionneur est supposé doté d'un réducteur, d'où le schéma de principe est donné en figure II.2.

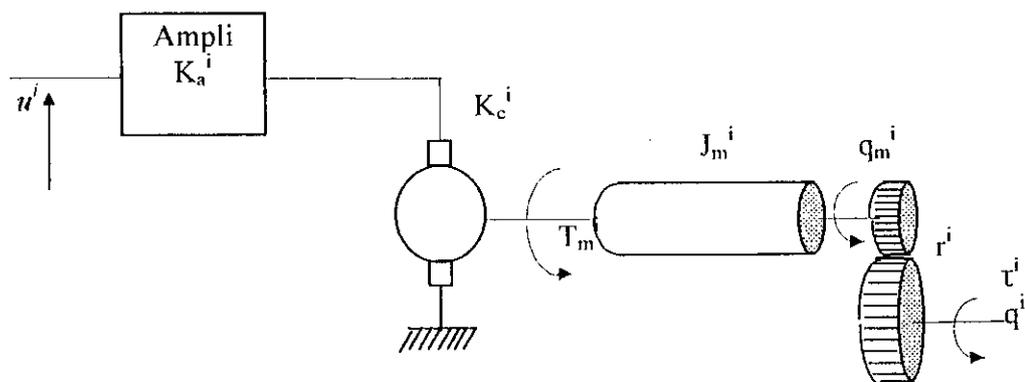


Figure II.2 Schéma de l'ensemble (Amplificateur + Actionneur + Réducteur).

Si l'on néglige les élasticités et les jeux, l'équation différentielle qui décrit le fonctionnement d'un tel moteur D.C. pour la liaison  $i$ , avec  $i = 1, \dots, n$  est (Vukobratovic, 1990) :

$$K_a^i K_c^i u^i - J_m^i \ddot{q}_m^i = \frac{1}{r^i} L_{ch}^i \quad (II.2)$$

Où :

$u^i$  : tension de commande [V].

$K_a^i$  : gain de l'amplificateur.

$K_c^i$  : constante de couple du moteur [NmV<sup>-1</sup>].

$r^i$  : rapport de réduction.

$J_m^i$  : inertie propre du moteur [Kgm<sup>2</sup>].

$L_{ch}^i$  : couple de charge à la sortie du réducteur [Nm].

$\ddot{q}_m^i$  : accélération angulaire du moteur [rad/s<sup>2</sup>].

Avec  $\ddot{q}_m^i = r^i \ddot{q}^i$

Ainsi l'équation (II.2) devient :

$$K_a^i K_c^i u^i - r^i J_m^i \ddot{q}^i = \frac{1}{r^i} L_{ch}^i \quad (II.3)$$

### II.3.3 Modèle Dynamique du Robot Manipulateur avec Actionneurs

Afin d'obtenir l'équation générale et complète du système robot manipulateur (c'est-à-dire en incluant la dynamique des actionneurs) l'équation dynamique du robot manipulateur (II.1) est combinée à l'équation dynamique (II.3) régissant le fonctionnement de l'ensemble (Amplificateur + Actionneur + Réducteur). Ainsi en remplaçant  $L_{ch}^i$  de l'équation (II.3) par le couple exercé sur chaque liaison  $\tau^i$  de l'équation (II.1) nous obtenons :

$$K_a^i K_c^i u^i - r^i J_m^i \ddot{q}^i = \frac{1}{r^i} \tau^i \quad (II.4)$$

d'où :

$$u^i = \frac{r^i}{K_a^i K_c^i} \left( J_m^i + \frac{1}{(r^i)^2} M_{ii}(q) \right) \ddot{q}^i + \frac{1}{r^i K_a^i K_c^i} \tau_1^i \quad (\text{II.5})$$

$$\tau_1^i = \sum_{\substack{j=1 \\ j \neq i}}^n M_{ij}(q) \ddot{q}^j + \sum_{j=1}^n \sum_{k=1}^n C_{jk}^i(q) \dot{q}^k \dot{q}^j + G^i(q) \quad (\text{II.6})$$

Cette expression est la forme générale de l'équation de l'ensemble (Amplificateur + Actionneur + Réducteur + Robot Manipulateur).

### II.3.4 Modélisation Des Frottements

Les frottements sont souvent négligés parce qu'ils sont difficiles à modéliser, bien qu'ils soient toujours présents dans les systèmes. Vu qu'ils s'opposent au mouvement d'un corps en contact avec un autre, les frottements représentent une partie indésirable d'un système dynamique et de là ils doivent être modélisés aussi précisément que possible.

En robotique, les frottements interviennent principalement au niveau des transmissions mécaniques et des actionneurs. La plupart des algorithmes d'identification proposés dans la littérature ne prennent pas en considération la dynamique des frottements dans le modèle dynamique du robot manipulateur (Lu et al., 1993). Par conséquent le modèle identifié ne sera pas suffisant pour la mise en œuvre d'une commande basée sur ce modèle.

Les frottements sont des phénomènes complexes dépendant de plusieurs paramètres dont principalement : la vitesse et le sens des mouvements, le type de contact, la température... En général, seuls des modèles découplés, fonctions des paramètres "vitesse et sens des mouvements" dans chaque liaison sont considérés (Pham, 1992).

Les modèles des frottements ont aussi fait l'objet d'études dans la littérature (Armstrong, 1988), et ne constituent pas l'objet essentiel de notre travail. A grandes vitesses de déplacement, le comportement des frottements  $\tau_{fi}$ , pour la liaison  $i$ , est généralement modélisé par l'équation suivante :

$$\tau_{fi} = f_{si} \text{sign}(\dot{q}^i) + f_{vi} \dot{q}^i \quad (II.7)$$

$f_{si}$  est un coefficient caractérisant le couple de frottement sec.

$f_{vi}$  est un coefficient caractérisant le couple de frottement visqueux,

$\text{sign}(\dot{q}^i)$  est défini par les relations :

$$(\text{sign}(\dot{q}^i) = 1 \quad \text{si } \dot{q}^i > 0)$$

$$(\text{sign}(\dot{q}^i) = -1 \quad \text{si } \dot{q}^i < 0)$$

En ajoutant le couple de frottements de l'équation (II.7) au couple total exercé sur chaque liaison de l'équation (II.6) le couple devient :

$$u^i = u^i + \tau_{fi} \quad (II.8)$$

$$u^i = \frac{r^i}{K_a^i K_c^i} \left( J_m^i + \frac{1}{(r^i)^2} M_{ii}(q) \right) \ddot{q}^i + \frac{1}{r^i K_a^i K_c^i} \tau_1^i + \frac{r^i}{K_a^i K_c^i} \tau_{fi} \quad (II.9)$$

Avec  $\tau_1^i$  est donné par l'équation (II.6) et  $\tau_{fi}$  est donné par l'équation (II.7).

L'équation (II.9) représente le modèle dynamique complet du robot manipulateur.

Pour plus de commodité nous réécrivons le modèle dynamique sous une forme matricielle. C'est cette forme qui sera prise en compte dans la suite de ce document.

$$U = M(q) \ddot{q} + H(q, \dot{q}) + F(\dot{q}) \quad (II.10)$$

Où

$U = [u^1, u^2, \dots, u^n]^T$  représente le vecteur (nx1) des couples (forces) de commandes.

$M(q) = [M_{ij}(q)]$  est la matrice d'inertie (nxn) qui est une fonction du vecteur de position  $q = [q^1, q^2, \dots, q^n]^T$ .

$H(q, \dot{q}) = \dot{q}^T C(q) \dot{q} + G(q)$  est le vecteur de dimension (nx1),

$\alpha(q) = [C_{jk}(q)]$  est une matrice de trois dimensions (nxn),

$G(q)$  est le vecteur  $n \times 1$  des moments (forces) de gravité.

$F(\dot{q})$  est le vecteur  $n \times 1$  des forces de frottement.

Comme il peut être constaté, le modèle dynamique du robot manipulateur représente un ensemble d'équations différentielles non linéaires du second ordre. Il peut être aussi remarqué que le mouvement de chaque liaison est fortement couplé avec le mouvement de toutes les autres liaisons. Nous insistons sur le fait que la matrice  $M$  et le vecteur  $H$  peuvent être des fonctions non linéaires entraînant dans le cas général les équations différentielles (II.10) à être très complexes.

## II.4 Propriétés du Modèle Dynamique

L'objectif de cette partie est de souligner quelques propriétés du modèle dynamique qui seront utilisées dans la suite de notre travail.

*Prop1 : La matrice d'inertie  $M$  est définie, positive et symétrique*

Cette propriété fondamentale a été la plus exploitée en robotique (Pham, 1992):

- La simulation numérique des robots utilise cette propriété pour l'intégration du système d'équations différentielles implicites et non linéaires qui constituent le modèle dynamique inverse.
- La commande non linéaire basée sur le modèle dynamique (Computed Torque Method) exploite les propriétés d'inversibilité de  $M$ .
- La stabilité des algorithmes de commande se base souvent sur le fait que  $M$  est définie et positive de  $M$  (Pham, 1992).

*Prop2 : Linéarité par rapport aux paramètres dynamiques*

Cette propriété est l'une des plus intéressantes du modèle dynamique. Il est démontré que les énergies cinétiques et potentielles peuvent être exprimées linéairement en fonction d'un vecteur des paramètres dynamiques (Pham, 1992 ; Kozlowski and Prüfer, 1992 ; Lu et al., 1993). Le modèle linéaire par rapport aux paramètres dynamiques sera utilisé dans la commande du robot RP41. Ceci est obtenu grâce à l'identification des paramètres dynamiques en utilisant la propriété de linéarité du modèle dynamique.

En premier lieu le modèle dynamique du robot manipulateur correspond à celui décrit dans la précédente section, mais sous cet aspect les paramètres dynamiques ne

peuvent pas être identifiés. La propriété de linéarité du modèle dynamique par rapport aux paramètres inertiels fournit une autre forme d'écriture du modèle. Sous cette forme et en rapport avec des considérations dynamiques de mouvement les paramètres dynamiques se regroupent en plusieurs combinaisons linéaires. Par conséquent il devient impossible d'obtenir chaque paramètre dynamique séparément. Ces paramètres peuvent être (Khosla, 1989) :

paramètres inertiels simples, ou une combinaison de plusieurs paramètres inertiels.

Puisque le modèle dynamique s'obtient par des opérations linéaires de dérivation des énergies, il est aussi fonction des mêmes paramètres (Pham, 1992). Ainsi, le modèle dynamique (II.2) peut être écrit comme suit :

$$U = W(q, \dot{q}, \ddot{q})P$$

où  $W$  représente une matrice contenant les informations cinématique qui décrivent le mouvement des liaisons. En générale, c'est une matrice non carrée. Dans la littérature cette matrice est appelée matrice d'information ou matrice d'observation.  $P$  représente le vecteur des paramètres dynamiques des combinaisons paramétriques.

## ***CHAPITRE III***

# ***COMMANDE NEURO-DYNAMIQUE ADAPTATIVE.***

---

---

Le but de ce troisième chapitre est de présenter une nouvelle stratégie de commande Neurale Dynamique Adaptative. Celle-ci est capable de surmonter les problèmes liés à la nature non linéaire du robot manipulateur, à la variation paramétrique et aux incertitudes dues aux dynamiques non modélisées.

### III.1 Décomposition Du Modèle Dynamique

Comme il a été mentionné dans le précédent chapitre, le fonctionnement du robot manipulateur, en absence de frottement et autres perturbations, peut être décrit par l'équation suivante :

$$T = M(q) \ddot{q} + H(q, \dot{q}) \quad (\text{III.1})$$

où  $T$  vecteur des couples exercés la matrice d'inertie  $M$  et les termes  $H$  (centrifuge et de coriolis) représentent un système d'équations différentielles du second ordre, non linéaires et couplées.

De plus le modèle dynamique de l'équation (III.1) est fonction des paramètres inertiels,  $P_i$ , des corps du robot (masse, inertie, ...). Ainsi le modèle dynamique peut être réécrit comme fonction explicite de ces paramètres :

$$T = M(q, P) \ddot{q} + H(q, \dot{q}, P) \quad (\text{III.2})$$

$P$  étant le vecteur composé des éléments inertiels  $P_i$ .

Dans les considérations précédentes nous avons supposé ces paramètres inertiels fixes et connus avec exactitude. En réalité, ces paramètres sont entachés d'erreurs d'incertitudes en rapport avec les procédés d'obtention et certains sont variables, en particulier les paramètres de l'ensemble du dernier corps (organe terminal + charge manipulé).

En effet les paramètres inertiels de la charge du robot manipulateur peuvent être décrits par un vecteur  $P_m$  comportant dix éléments : la masse de la charge, l'emplacement du centre de masse dans l'espace et les six moments d'inertie.

$$P_m = \left[ m \quad m_{cx} \quad m_{cy} \quad m_{cz} \quad I_{11} \quad I_{12} \quad I_{13} \quad I_{22} \quad I_{23} \quad I_{33} \right]^t$$

$m$  : la masse de la charge,

$(m_{cx}, m_{cy}, m_{cz})$  : coordonnées du centre de masse dans le repère XYZ associé.

$I_{11}, I_{12}, I_{13}, I_{22}, I_{23}, I_{33}$  : les moments d'inertie de la charge. En réalité ces moments forment une matrice d'inertie de neuf éléments, mais vu la propriété de symétrie de cette matrice, la connaissance de six éléments est suffisante.

En considérant le dernier corps du robot et sa charge comme étant un corps unique, les paramètres inertiels du robot varient selon les valeurs de la charge. Par conséquent le vecteur  $P$  comporte deux parties :

- Des paramètres inertiels fixes  $P_f$ .
- Des paramètres inertiels variables  $P_v$ .

Le modèle dynamique du robot (équation III.2) peut être donc décomposé en deux parties :

- Partie fixe, dont le modèle dynamique est fonction des paramètres dynamiques sont fixes, est décrite par :

$$T_f = M_f(q, P_f) \ddot{q} + H_f(q, \dot{q}, P_f) \quad (\text{III.3})$$

- Partie variable, dont le modèle dynamique est fonction des paramètres dynamiques sont variables, est décrite par :

$$T_v = M_v(q, P_v) \ddot{q} + H_v(q, \dot{q}, P_v) \quad (\text{III.4})$$

Il est évident que la propriété de linéarité du modèle dynamique par rapport aux paramètres inertiels demeure valable suite cette décomposition. Ainsi les modèles dynamiques fixe et variable sont eux aussi linéaire par rapport aux paramètres fixes et variables respectivement.

## III.2 Commande Neurale Dynamique Adaptative

L'approche de commande Neurale Dynamique Adaptative que nous proposons combine les trois techniques de commande dynamique, neurale et Adaptative (Kadri et al., 1996b).

Le concept de notre approche, dont le schéma de principe est montré en figure III.1, est fondé sur l'utilisation conjointe d'un modèle dynamique à deux composantes (fixe et variable) du robot manipulateur permettant le calcul d'une commande anticipée par un système de réseaux de neurones tirant profit de la connaissance du comportement dynamique du robot. Le bloc de retour permet la compensation des non linéarités non prises en compte par la modélisation dynamique.

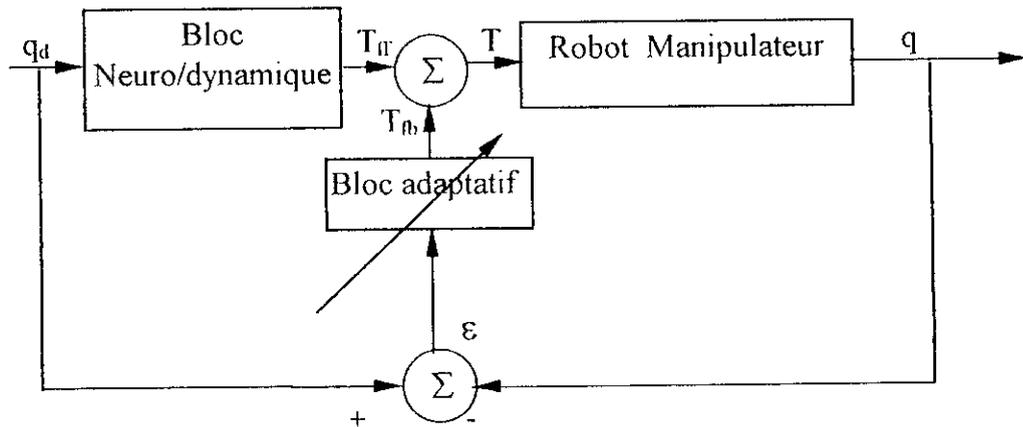


Figure III.1 . Commande Neurale Dynamique Adaptative

L'action neuro-dynamique de la stratégie proposée a pour objectif de compenser :

- les variations des efforts d'inertie d'accélération,
- les variations des efforts d'inertie de coriolis,
- les variations des efforts d'inertie centrifuge,
- les variations des efforts d'inertie induite par les paramètres dynamiques variables.

Le rôle de la partie de retour adaptative est de compenser :

- les variations dynamiques non prise en compte par le modèle dynamique,
- les erreurs dues à l'inexactitude de la modélisation dynamique du robot manipulateur,
- les erreurs dues aux incertitudes d'estimation des paramètres dynamiques fixes et variables du robot manipulateur,
- les phénomènes non linéaires tels que les frottements secs et visqueux, les jeux dans les réducteurs, etc.

Le couple de commande total est donné par :

$$T = T_{ff} + T_{fb} \quad (\text{III.5})$$

où :  $T_{ff}$  représente le couple d'anticipation,  
 $T_{fb}$  représente le couple de retour.

Le couple d'anticipation (figure III.2) est fourni par la partie neuro-dynamique en se basant sur le modèle dynamique inverse du robot manipulateur. Cette partie utilise la

décomposition de ce dernier en couple dynamique fixe et couple dynamique variable. Par conséquent le couple d'anticipation sera décrit comme suit :

$$T_{ff} = T_{ff1} + T_{ff2} \quad (\text{III.6})$$

où  $T_{ff1}$  est le vecteur  $n \times 1$  des couples fixes,

$T_{ff2}$  est le vecteur  $n \times 1$  des couples variables,  $n$  étant le nombre de liaisons.

$$T_{ff1} = M_f(q, P_f) \ddot{q} + H_f(q, \dot{q}, \ddot{q}, P_f) \quad (\text{III.7})$$

$$T_{ff2} = M_v(q, P_v) \ddot{q} + H_v(q, \dot{q}, \ddot{q}, P_v) \quad (\text{III.8})$$

$M_f$ ,  $M_v$  représentent les matrices d'inertie contenant les paramètres fixes et les paramètres variables respectivement.  $H_f$ ,  $H_v$  représentent les vecteurs de coriolis, centrifuge et de gravitation contenant les paramètres fixes et les paramètres variables respectivement.

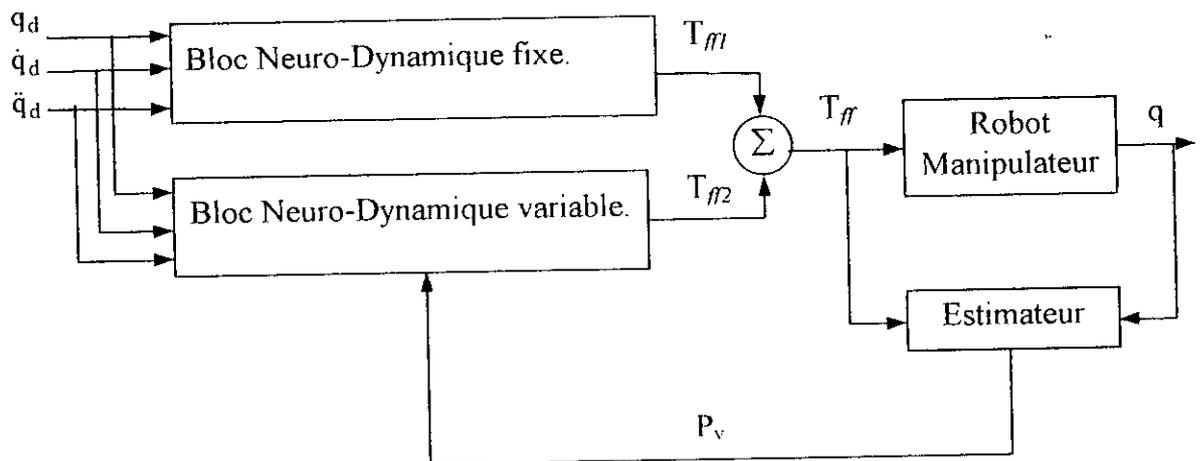


Figure III.2. Partie Neuro-Dynamique de la commande.

Cette décomposition permet de simplifier le problème en évitant la nécessité d'un modèle dynamique très précis. A cet effet, notre connaissance partielle de la dynamique du robot manipulateur est suffisante pour fournir le couple fixe. Le couple variable qui utilise une estimation des paramètres inertiels variables, compense le changement dans la dynamique du système durant le mouvement du robot. En plus cette décomposition nous permet l'utilisation de réseaux de neurones de taille acceptable. Ceci accélère l'apprentissage et améliore la généralisation des réseaux par réduction de la complexité des fonctions à représenter (Katić and vokobratović, 1992).

### III.3 Deux Méthodes De Mise En Œuvre De La Commande Neuro-Dynamique Adaptative

Pour mettre en œuvre notre solution de commande, deux variantes apparaissent selon la manière d'utiliser le modèle dynamique du robot dans la commande : hors ligne ou en ligne (An, 1986). Ainsi deux variantes découlent, de notre approche de commande :

- Commande Neurale Dynamique Anticipée Adaptative.
- Commande Neurale par Couple Calculé Adaptative.

#### III.3.1 Commande Neurale Dynamique Anticipée Adaptative

Dans la première variante, montrée en figure III.3, (Kadri et al., 1997a), le couple d'anticipation est donné par l'expression de l'équation (III.6). le couple fixe est calculé, préalablement pour chaque cycle de commande, hors ligne. Ceci est possible vu que le couple fixe est fonction uniquement des paramètres de la trajectoire désirée. Ces données sont stockées dans des tables ( Look-Up Tables ). De cette manière nous réduisons le temps nécessaire pour le calcul de la commande au temps d'accès aux tables. Le couple variable est obtenu par un réseau de neurone qui a comme entrées les trajectoires de position, vitesse et accélération désirées ainsi que les paramètres variables estimés à l'aide d'un estimateur par réseau de neurones. Il est à noter que si le nombre de liaisons et le nombre de paramètres estimés sont importants la dynamique variable devient très complexe ce qui rend la taille du réseau de neurones importante et par conséquent une mauvaise convergence de l'erreur dans l'apprentissage du réseau et peut être une très mauvaise généralisation sur les données non apprises (Cichocki and Unberhauen, 1993 ; Katić and vokobratović, 1992). De ce fait il est préférable pour réduire la taille du réseau, d'utiliser un réseau indépendant pour chaque liaison. Malgré que cette solution puisse apporter une amélioration dans la convergence de l'erreur durant l'apprentissage, il reste que la généralisation des réseaux demeure toujours mauvaise. Pour surmonter ce problème nous pouvant simplifier davantage la dynamique en utilisant la propriété de linéarité du modèle dynamique par rapport aux paramètres inertiels. Par conséquent, au lieu que le réseau de neurone représente tout le couple variable, il ne représentera que la matrice d'information (d'observation) ; et le couple variable sera donné par :

$$T_{ff2} = W(\theta_d, \dot{\theta}_d, \ddot{\theta}_d)P_v \quad (III.9)$$

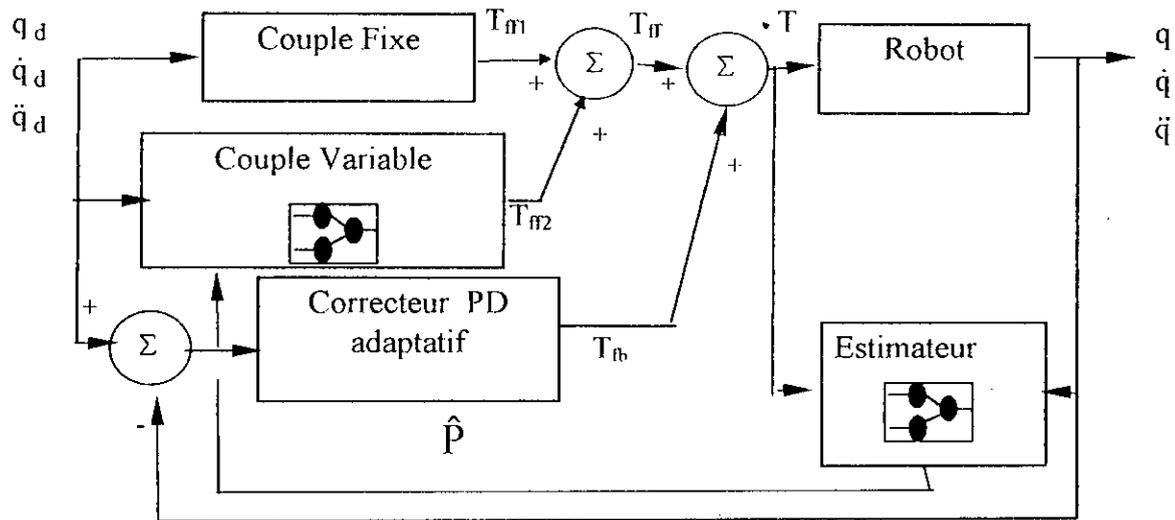


Figure III.3. Commande Neurale Anticipée Dynamique Adaptative.

### III.3.2 Commande Neurale Par Couple Calculé Adaptative

La seconde variante, montrée en figure III.4, utilise le modèle dynamique en se basant sur le concept de la méthode du calcul du couple généralisé. Par conséquent la loi de commande, qui fournit le couple exercé sur les liaisons, est donnée par les expressions suivantes (Kadri et al. 1997c ; Kadri et al., 1998) :

$$T = T_{ff1} + T_{ff2} \quad (\text{III.10})$$

$$T_{ff1} = M_f(q, P_f) \ddot{q}^* + H_f(q, \dot{q}, \ddot{q}, P_f) \quad (\text{III.11})$$

$$T_{ff2} = M_v(q, P_v) \ddot{q}^* + H_v(q, \dot{q}, \ddot{q}, P_v) \quad (\text{III.12})$$

$T$  est le couple total exercé sur les liaisons,  $T_{ff1}$  représente le couple fixe et  $T_{ff2}$  est le couple variable,  $\ddot{q}^*$  est donnée par :

$$\ddot{q}^* = \ddot{q}_d + K_p(q_d - q) + K_v(\dot{q}_d - \dot{q}) \quad (\text{III.13})$$

Comme pour la première variante de l'approche, nous utilisons des réseaux de neurones dans l'établissement des couples fixe et variable. Dans cette variante, comme le montre les expressions (III.10) – (III.13), en plus de la décomposition en parties fixe et variable, nous effectuons une décomposition en matrice d'inertie  $M$  et termes de coriolis et centrifuge  $H$ . L'équation (III.13) montre que l'action de correction adaptative de retour, qui est la même

que dans le cas de la première variante, est ajoutée au signal d'accélération. Le recours à cette deuxième variante est dicté par le besoin d'améliorer les performances de l'approche de commande et d'obtenir des réseaux de neurones de tailles plus réduite para rapport à la première variante. En effet cette décomposition en M et H, peut réduire substantiellement la taille des réseaux de neurones par rapport à la première variante car elle réduit, par décomposition, la complexité de la fonction à représenter par ces réseaux (Katić and vokobratović, 1992). Pour ce faire, nous considérons deux réseaux de neurones indépendants pour chaque matrice d'inertie M (fixe  $M_f$  et variable  $M_v$ ) et deux autres réseaux de neurones indépendants pour chaque vecteur H de coriolis et centrifuge (fixe  $H_f$  et variable  $H_v$ ).

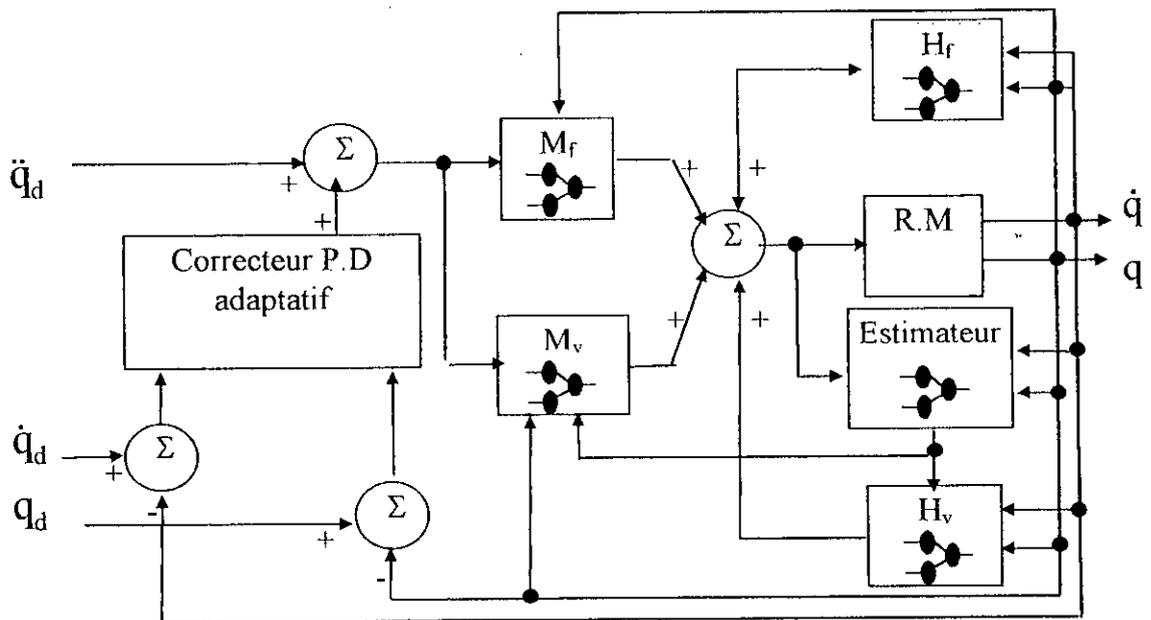


Figure III.4. Commande Neurale par couple calculé Adaptative.

le comportement de l'erreur est décrit par l'équation différentielle suivante :

$$\left(\hat{M}_f(q_d, P_f) + \hat{M}_v(q_d, P_v)\right) \left(\ddot{q}_d + K_p(q_d - q) + K_v(\dot{q}_d - \dot{q})\right) + \hat{H}_f(q, \dot{q}, P_f) + \hat{H}_v(q, \dot{q}, P_v) = \quad (III.14)$$

$$\begin{aligned} & M_f(q, P_f) \ddot{q} + H_f(q, \dot{q}, P_f) + M_v(q, P_v) \ddot{q} + H_v(q, \dot{q}, P_v) \\ & \left(\hat{M}_f(q_d, P_f) + \hat{M}_v(q_d, P_v)\right) \left(\ddot{e} + K_p \dot{e} + K_v e\right) = \left(\left(M_f(q_d, P_f) + M_v(q_d, P_v)\right) - \left(\hat{M}_f(q_d, P_f) + \hat{M}_v(q_d, P_v)\right)\right) \ddot{q} \\ & + \left(\left(H_f(q, \dot{q}, P_f) + H_v(q, \dot{q}, P_v)\right) - \left(\hat{H}_f(q, \dot{q}, P_f) + \hat{H}_v(q, \dot{q}, P_v)\right)\right) \end{aligned} \quad (III.15)$$

$$(\ddot{e} + K_v \dot{e} + K_p e) = (\hat{M}_f + \hat{M}_v)^{-1} \left( (M_f + M_v) - (\hat{M}_f + \hat{M}_v) \right) \ddot{q} + \left( (H_f + H_v) - (\hat{H}_f + \hat{H}_v) \right) \quad (\text{III.16})$$

l'erreur tendra vers zéro si :

$$(\hat{M}_f + \hat{M}_v) \rightarrow (M_f + M_v) \quad \text{et} \quad (\hat{H}_f + \hat{H}_v) \rightarrow (H_f + H_v).$$

Nous constatons donc que l'erreur est directement liée aux erreurs d'estimation de la matrice M et du vecteur H aussi bien pour la partie fixe que pour la partie variable. D'un autre côté, il a été montré que cette méthode (Computed Torque) est robuste pour de petites erreurs de modélisation (Gilbert and Ha, 1984 ; An et al., 1989). La décomposition en couple fixe et variable et la correction adaptative agissant sur les perturbations aux erreurs de modélisation permet d'avoir un modèle dynamique le plus précis possible, ce qui rend cette deuxième variante encore plus robuste vis à vis de grandes perturbations.

### III.3.3 Estimation Des Paramètres Variables

L'estimation des paramètres inertiels variables a fait l'objet de plusieurs études (Paul, 1981 ; Olsen and Bekey, 1985). En général, les techniques développées sont basées sur la méthode des moindres carrés et l'utilisation de la propriété de linéarité du modèle dynamique par rapport aux paramètres inertiels. Cette approche est assez simple à implémenter, bien qu'elle exige beaucoup de temps de calcul et confrontée au problème d'initialisation où la fluctuation des paramètres au début d'estimation est très importante.

Afin de contourner ce problème nous proposons d'utiliser un estimateur basé sur les réseaux de neurones. L'estimateur peut fournir une valeur estimée des paramètres inertiels variables sur la base des données d'entrées sorties du système. Pour se faire deux possibilités se présentent, la première en se basant sur l'erreur de poursuite en position et en vitesse (Leahy et al., 1991). Dans la deuxième solution, l'estimation se base sur les couples appliqués aux différentes liaisons du robot et leurs trajectoires réelles (Yegerlehner and Meckl, 1993). Vu que l'erreur de poursuite peut être causée par d'autres perturbations que la variation des paramètres, l'estimation de ces derniers avec la première solution peut être induite en erreur. En effet, l'estimateur ne va pas fournir une estimation des paramètres variables mais une valeur approchée par rapport à toutes les perturbations.

Dans la méthode que nous avons adopté le réseau de neurones utilisé fournit une estimation des paramètres variables avec comme entrées les couples exercés sur chaque liaison et les trajectoires de position, vitesse, et d'accélération réelles.

### III.3.4 Apprentissage Des Réseaux De Neurones

Tous les réseaux de neurones considérés dans le présent travail sont des réseaux multicouches d'anticipation "feedforward". Ce type de réseaux est en général constitué d'une ou plusieurs couches cachées entre les couches d'entrée et de sortie. Chaque couche est constituée d'unités de calcul appelées nœuds (neurones connectés les uns aux autres dans la structure de réseau multicouche comme illustré en figure III.5). La fonctionnalité du réseau est déterminée en spécifiant les degrés de connections des chemins appelés poids et le paramètre seuil de chaque neurone (Billings and Jamaluddin, 1991).

La couche d'entrée n'est pas une couche en réalité, elle ne se comporte que comme un registre pour les données d'entrées puis elle les distribue à la première couche cachée. Les données d'entrées se propagent alors en avant à travers le réseau et chaque neurone calcule sa sortie selon l'expression suivante :

$$x_i^k = g\left(\sum_{j=1}^{n_{k-1}} W_{ij}^k x_j^{k-1} + b_i^k\right) \quad (\text{III.17})$$

pour  $i=1, \dots, n_k$  et  $k=1, \dots, L$ , où  $x_i^k$  est la sortie du  $i^{\text{ème}}$  neurone de la  $k^{\text{ème}}$  couche,  $W_{ij}^k$  est le poids de connexion entre le  $j^{\text{ème}}$  neurone de la  $(k-1)^{\text{ème}}$  couche et le  $i^{\text{ème}}$  neurone de la  $k^{\text{ème}}$  couche.  $b_i^k$  représente le seuil du  $i^{\text{ème}}$  neurone de la  $k^{\text{ème}}$  couche.

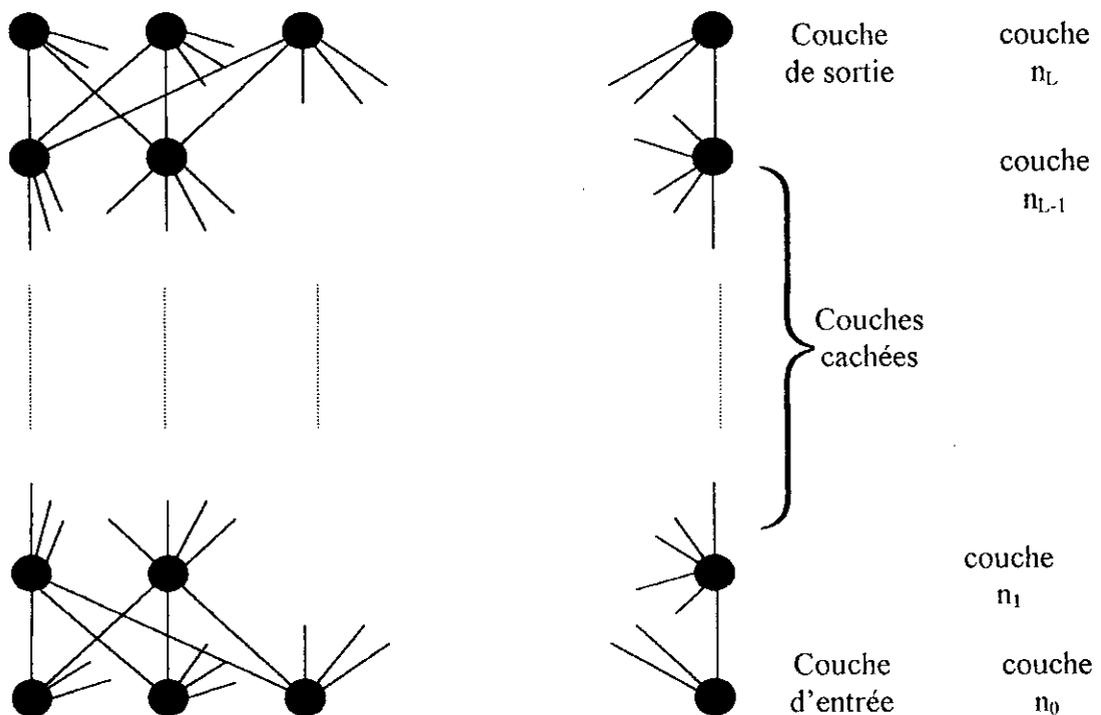


Figure III.5 Réseau de neurones multicouches.

Comme le montre la figure III.5 la couche zéro est la couche d'entrée, la  $L^{\text{ème}}$  couche est la couche de sortie. Ainsi,  $n_0$  et  $n_L$ , représentent respectivement les nombres d'entrées et de sorties du réseau. Pour une simplicité d'écriture  $x_i^0$  et  $x_i^L$  seront représentés par  $x_i$  et  $\hat{y}_i$ . La fonction  $g(\cdot)$  est appelée fonction d'activation du nœud. L'argument :

$$z_i^k = \sum_{j=1}^{n_{k-1}} W_{ij}^k x_j^{k-1} + b_i^k \quad (\text{III.18})$$

correspond à l'activation du  $i^{\text{ème}}$  neurone de la  $k^{\text{ème}}$  couche. La fonction  $g(\cdot)$  est supposée être différentielle. Pour le neurone situé dans les couches cachées, la fonction d'activation est souvent choisie comme :

$$g(z) = \frac{1 - e^{-z}}{1 + e^{+z}} \quad (\text{III.19})$$

Puisque dans notre cas la dynamique de la gamme des données de sortie n'est pas une fonction logique, c'est-à-dire que la sortie comporte des données réelles alors la fonction d'activation des nœuds de sortie est choisie linéaire. Ainsi, le  $i^{\text{ème}}$  nœud de sortie exécute la somme pondérée de ses entrées comme suit :

$$\hat{y} = \sum_{j=1}^{n_{L-1}} (W_{ij}^L x_j^{L-1} + b_i^{L-1}) \quad (\text{III.20})$$

Soit  $\theta = [\theta_1, \dots, \theta_{n_\theta}]$  un vecteur représentant tous les poids et les seuils du réseau à déterminer.  $n_\theta$  est la dimension du vecteur  $\theta$ .

Pour appliquer l'algorithme de rétropropagation basé sur la méthode de la descente du gradient de l'erreur quadratique moyenne entre la sortie actuelle du réseau et la sortie désirée, on définit une fonction objective comme suit :

$$J = \frac{1}{2} \sum_{i=1}^{n_L} (y_i(t) - \hat{y}_i(t))^2 \quad (\text{III.21})$$

où  $y_i(t)$  est la  $i^{\text{ème}}$  sortie désirée. Cette fonction "énergie" doit être minimisée par rapport à tous les paramètres inconnus. Dans la méthode de la descente du gradient, le vecteur  $\theta$  est ajusté en utilisant le vecteur d'incrementation  $[\Delta\theta_1, \dots, \Delta\theta_{n_\theta}]$  défini le long de la direction négative du gradient de  $J$  de la façon suivante :

$$\Delta \theta_i = -\eta \frac{\partial J}{\partial \theta_i} = -\eta \nabla_{\theta_i} J \quad (\text{III.22})$$

En commençant par la dernière couche du réseau, l'application de la règle de la descente du gradient donne :

$$\frac{\partial J}{\partial W_{ij}^L} = \frac{\partial J}{\partial \hat{y}_i} \frac{\partial \hat{y}_i}{\partial W_{ij}^L} \quad (\text{III.23})$$

De l'équation (III.21) on obtient le premier terme de (III.23) :

$$\frac{\partial J}{\partial \hat{y}_i} = -(y_i - \hat{y}_i) = -\delta_i^L \quad (\text{III.24})$$

où  $\delta_i^L$  est appelé signal d'erreur du  $i^{\text{ème}}$  neurone de la  $L^{\text{ème}}$  couche.

De l'équation (III.17) on obtient l'expression du deuxième terme de (III.23) :

$$\frac{\partial \hat{y}_i}{\partial W_{ij}^L} = x_j^{L-1} \quad (\text{III.25})$$

$$\text{D'où } \frac{\partial J}{\partial W_{ij}^L} = -\delta_i^L x_j^{L-1} \quad (\text{III.26})$$

Si on considère maintenant la  $(L-1)^{\text{ème}}$  couche, en utilisant la règle de la descente du gradient :

$$\frac{\partial J}{\partial W_{ij}^{L-1}} = \sum_k \frac{\partial J}{\partial \hat{y}_k} \frac{\partial \hat{y}_k}{\partial W_{ij}^{L-1}} \frac{\partial \hat{y}_k}{\partial z_i^{L-1}} \frac{\partial z_i^{L-1}}{\partial W_{ij}^{L-1}} \quad (\text{III.27})$$

on peut facilement remarquer que :

$$\frac{\partial \hat{y}_k}{\partial x_i^{L-1}} = W_{ki}^L$$

$$\frac{\partial x_i^{L-1}}{\partial z_i^{L-1}} = g'(z_i^{L-1})$$

$$\frac{\partial z_i^{L-1}}{\partial W_{ij}^{L-1}} = x_j^{L-2}$$

$$\text{où } g'(z) = \frac{\partial g(z)}{\partial z}$$

En définissant le signal d'erreur pour le  $i^{\text{ème}}$  neurone de la  $(L-1)^{\text{ème}}$  couche comme l'expression suivante :

$$\delta_i^{L-1} = \sum_i \left( -\frac{\partial J}{\partial y_k} \right) \frac{\partial y_k}{\partial x_i^{L-1}} \frac{\partial x_i^{L-1}}{\partial z_i^{L-1}} = g'(z_i^{L-1}) \sum_k \delta_k^L W_{ki}^L$$

L'équation (III.27) peut être réécrite alors comme suit

$$\frac{\partial J}{\partial W_{ij}^{L-1}} = -\delta_i^{L-1} x_j^{L-2}$$

D'une manière similaire il peut être montré que :

$$\frac{\partial J}{\partial b_{ij}^{L-1}} = -\delta_i^{L-1}$$

En poursuivant cette procédure les résultats généraux suivants sont obtenus pour  $m = L, L-1, \dots, 2$ .

$$\delta_i^{m-1} = g'(z_i^{m-1}) \sum_k \delta_k^m W_{ki}^m \quad (\text{III.28a})$$

$$\frac{\partial J}{\partial W_{ij}^{m-1}} = -\delta_i^{m-1} x_j^{m-2} \quad (\text{III.28b})$$

$$\frac{\partial J}{\partial b_{ij}^{m-1}} = -\delta_i^{m-1} \quad (\text{III.28c})$$

L'équation (III.28a) montre comment le signal d'erreur se propage en arrière, d'où le terme de rétropropagation.

La minimisation de la fonction d'énergie définie dans l'équation (III.21) produit les incréments de mise à jour de  $\theta$  comme suit :

$$\Delta W_{ij}^k(t) = \eta_w \delta_j^k(t) x_j^{k-1}(t) \quad (\text{III.29a})$$

$$\Delta b_{ij}^k(t) = \eta_b \delta_j^k(t) \quad (\text{III.29b})$$

où  $\delta_i^1(t) = y_i(t) - \hat{y}_i(t)$

$$\text{et } \delta_i^k = g'(z_i^k(t)) \sum_m \delta_m^{k+1} W_m^{k+1}(t-1), \quad k = (L-1), \dots, 2, 1$$

Les constantes  $\eta_w$  et  $\eta_b$  représentent les taux d'apprentissage pour les poids et les seuils respectivement. En pratique, une grande valeur du taux d'apprentissage est préférable car il entraîne un apprentissage rapide. Malheureusement, une grande valeur du taux d'apprentissage peut aussi aboutir à une oscillation ou même à une divergence. Pour surmonter ce problème, un terme momentum est, en général, inclus tel que l'équation (III.29a) et (III.29b) devient :

$$\begin{aligned} \Delta W_{ij}^k(t) &= \eta_w \delta_j^k(t) x_j^{k-1}(t) + \alpha_w \Delta W_{ij}^k(t-1) \\ \Delta b_{ij}^k(t) &= \eta_b \delta_j^k(t) + \alpha_b \Delta b_{ij}^k(t-1) \end{aligned}$$

Où  $\alpha_w$  et  $\alpha_b$  sont des constantes momentum qui déterminent l'effet des changements précédents de  $\Delta W_{ij}^k(t-1)$  et  $\Delta b_{ij}^k(t-1)$  sur la direction de la mise à jour courante dans les poids et les seuils respectivement.

En résumé, l'algorithme de rétropropagation met à jour les poids et les seuils du réseau selon :

$$\begin{aligned} W_{ij}^k(t) &= W_{ij}^k(t-1) + \Delta W_{ij}^k(t) \\ b_{ij}^k(t) &= b_{ij}^k(t-1) + \Delta b_{ij}^k(t) \end{aligned}$$

Avec  $\Delta W_{ij}^k(t)$  et  $\Delta b_{ij}^k(t)$  sont données par les équations (III.30a) et (III.30b).

L'implémentation de l'algorithme de rétropropagation nécessite deux phases. Les poids et les seuils sont généralement initialisés avec de petites valeurs aléatoires (Narandra and Parthasarathy, 1990 ; Lippmann, 1987). Dans la première phase, les données d'entrée sont présentées à la couche d'entrée puis ces signaux se propagent à travers le réseau jusqu'à la couche de sortie. Les sorties actuelles du réseau, obtenues des neurones de sortie, sont comparées avec les sorties désirées pour calculer les erreurs de prédiction (résiduelles). La seconde étape nécessite la propagation des signaux d'erreur en arrière de la couche de sortie vers la première couche. Les poids et les seuils sont mis à jour sur la base de ces signaux d'erreur. Ce procédé est répété jusqu'à la convergence des poids et des seuils.

### III.3.4 Correction PD Adaptative

Pour compenser les incertitudes des dynamiques non modélisées ainsi qu'autres incertitudes dues aux erreurs d'estimation et d'autres perturbations, un correcteur Proportionnel et Dérivé adaptatif est inclus dans la boucle de retour.

L'adaptation des gains du correcteur de ce type peut se faire selon plusieurs algorithmes. On distingue, dans la littérature, deux types d'algorithmes : explicite et implicite,

- Les algorithmes explicites, connus aussi sous la nomination d'algorithmes indirectes, procèdent par une estimation récursive du modèle du système suivi par un recalcul de la loi d'adaptation.
- Les algorithmes implicites, connus aussi sous la nomination d'algorithmes directes, procèdent à l'estimation directe des paramètres du correcteur sans passer par la détermination du modèle du système.

A noter que ces deux algorithmes rencontrent des difficultés de mise en œuvre en particulier l'initialisation et le coût important en temps de calcul. Pour éviter ces problèmes nous avons opté pour la loi d'adaptation des gains du PD basé sur les travaux de Seraji (Seraji, 1989).

Nous nous sommes limités aux actions proportionnelle et dérivée car l'action intégrale agit sur l'erreur de traînage, ceci est significatif dans le cas d'une régulation et non dans le cas de poursuite de trajectoire.

La loi de commande de l'action de retour adaptative (Figure III.6) , exprimé pour chaque liaison  $i$ , est donnée comme suit :

$$T_{fbi} = K_{pi}(t)e_i(t) + K_{vi}(t)\dot{e}_i(t) \quad (\text{III.31})$$

où le terme  $e_i(t) = q_{di}(t) - q_i(t)$  est l'erreur de poursuite en position.

Il est évident que le couple de retour  $T_{fbi}$  épouse l'expression d'une commande de type P.D. Les gains  $K_{pi}(t)$  et  $K_{vi}(t)$  du correcteur sont ajustés en se basant sur l'erreur de poursuite en position et en vitesse  $e_i(t)$  et  $\dot{e}_i(t)$  respectivement.

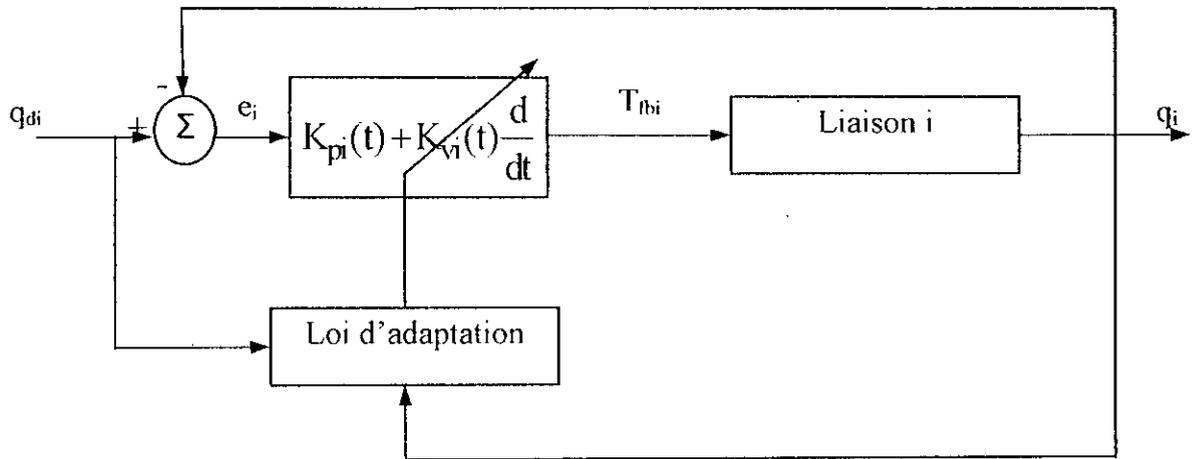


Figure III.6. Correcteur Adaptatif de la stratégie de commande.

La loi d'adaptation des gains  $K_{pi}$  et  $K_{vi}$  est basée sur le signal d'erreur pondéré  $r_i(t)$  donné comme suit :

$$r_i(t) = w_{pi}e_i(t) + w_{vi}\dot{e}_i(t) \quad (III.32)$$

$\{w_{pi}, w_{vi}\}$  sont les facteurs de pondération de l'erreur en position et de l'erreur en vitesse.

Les gains  $K_{pi}$  et  $K_{vi}$  sont donné par :

$$k_{pi}(t) = k_{pi}(0) + \alpha_0 \int_0^t r_i(\tau)e_i(\tau)d\tau \quad (III.33a)$$

$$k_{vi}(t) = k_{vi}(0) + \alpha_1 \int_0^t r_i(\tau)\dot{e}_i(\tau)d\tau \quad (III.33b)$$

$\alpha_0$  et  $\alpha_1$  sont les paramètres d'adaptation intégrale,  $K_{pi}(0)$  et  $K_{vi}(0)$  sont les gains proportionnel et dérivé initiaux.

De cette expression nous constatons que l'adaptation des gains du correcteur PD est fonction de l'erreur de poursuite. Les valeurs de ces gains convergent vers une lorsque l'erreur de poursuite tend vers zéro.

Dans une implémentation numérique de la commande, les équations de la loi d'adaptation peuvent être discrétisées et nous obtenons une loi d'adaptation récursive suivante (Seraji, 1989) :

$$r_i(N) = w_{pi}e_i(N) + w_{vi}\dot{e}_i(N) \quad (\text{III.34})$$

$$k_{pi}(N) = k_{pi}(N-1) + \alpha_0[r_i(N)e_i(N) + r_i(N-1)e_i(N-1)] \quad (\text{III.35a})$$

$$k_{vi}(N) = k_{vi}(N-1) + \alpha_1[r_i(N)\dot{e}_i(N) + r_i(N-1)\dot{e}_i(N-1)] \quad (\text{III.35b})$$

N : correspond à l'instant d'échantillonnage.

- Ces lois d'adaptation (III.32)-(III.33b) sont assez simple et exigent seulement peu d'opérations arithmétiques. Un regard près des équations (III.32)-(III.33b) révèle que chaque gain est dépendant uniquement du signal sur lequel il opère et donc sur l'erreur de poursuite pondérée (Seraji, 1989).
- Les lois d'adaptation proportionnelles plus intégrales (III.32)-(III.33b) donnent une large famille de schémas d'adaptation assurant la poursuite de trajectoire. L'utilisation de ces lois produit de meilleures réponses transitoires, améliore la convergence, et augmente la flexibilité en comparaison avec les lois d'adaptation intégrales conventionnelles (Seraji, 1989).
- Il est montré que les lois d'adaptation (III.32)-(III.33b) et l'action de correction (III.31) sont basées entièrement sur les performances du robot manipulateur observées à travers  $q(t)$  et  $q_d(t)$ . Ainsi le correcteur adaptatif peut compenser les incertitudes. Ceci est une caractéristique hautement désirée dans des applications pratiques où quelques effets dynamiques tel que les frottements ne peuvent pas être modélisés avec précision (Seraji, 1989).
- En plus de sa simplicité, l'avantage offert par cette loi d'adaptation en terme de temps de calcul rend cette loi efficace dans le cas d'une implémentation en ligne (Seraji, 1989).

## *CHAPITRE IV*

## *SIMULATION*

---

---

Dans le présent chapitre la commande proposée sera validée par une simulation numérique en utilisant le logiciel MATLAB. Nous envisageons cette simulation dans le cas du robot RP41 limité à ces deux premières liaisons.

## IV.1 Modèle Dynamique Du Robot

Le modèle dynamique du robot manipulateur adopté pour la simulation dans ce chapitre dont le schéma général est donnée en figure IV.2. Comme le montre la figure IV.1 le robot étudié correspond au deux premières liaisons du RP41. Afin d'augmenter la complexité du modèle nous avons considéré le robot dans une configuration où les termes gravitationnelles sont présents. Ainsi, l'équation dynamique similaire à celle énoncée dans le chapitre II est :

$$T = M(q)\ddot{q} + C(q, \dot{q}) + G(q) + F(\dot{q})$$

$$M(q) = \begin{pmatrix} a_1 + a_2 \cos(q_2) & a_3 + \frac{a_2}{2} \cos(q_2) \\ a_3 + \frac{a_2}{2} \cos(q_2) & a_3 \end{pmatrix} \quad C(q, \dot{q}) = \begin{pmatrix} -(a_2 \sin(q_2)) \left( \dot{q}_1 \dot{q}_2 + \frac{\dot{q}_2^2}{2} \right) \\ (a_2 \sin(q_2)) \frac{\dot{q}_1^2}{2} \end{pmatrix}$$

$$G(q) = \begin{pmatrix} a_4 \cos(q_1) + a_5 \cos(q_1 + q_2) \\ a_5 \cos(q_1 + q_2) \end{pmatrix}$$

où :

$$a_1 = i_1 + i_2 + (m_1 + m_2)L_1^2 + m_2L_2^2$$

$$a_2 = i_2 + 2m_2L_1L_2$$

$$a_3 = m_2L_2^2$$

$$a_4 = (m_1 + m_2)gL_1$$

$$a_5 = m_2gL_2$$

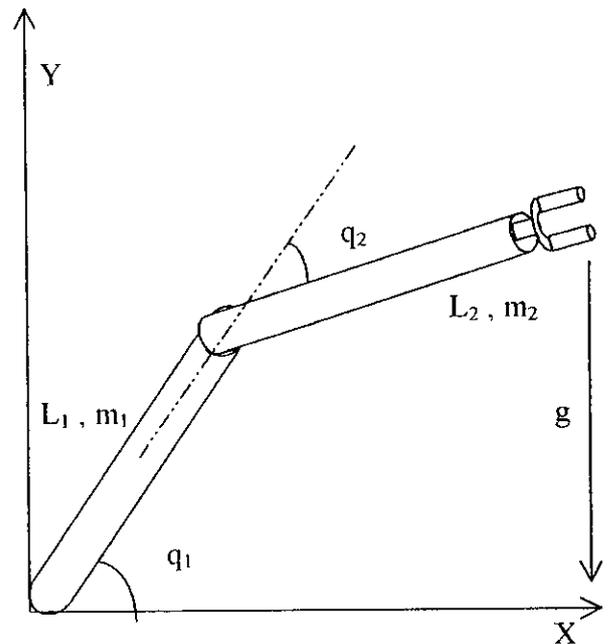


Figure IV.1 Robot Manipulateur à deux liaisons

Les valeurs des masses, des longueurs et des inerties des deux liaisons sont les suivantes :

$$m_1 = 15.91 \text{ Kg}, m_2 = 11.36 \text{ Kg}$$

$$L_1 = L_2 = 0.432 \text{ m}$$

$$g = 9.81 \text{ m/s}^2$$

$$i_1 = 0.25 \text{ Kg} \cdot \text{m}^2, i_2 = 0.18 \text{ Kg} \cdot \text{m}^2$$

## IV.2 Etude De L'influence Des Termes Dynamiques

Dans les considérations précédentes nous avons supposé que le modèle dynamique du robot et ses paramètres soient connus et déterminés au préalable. De telles suppositions se maintiennent pour certains paramètres tels que la masse et la longueur de la première liaison. Cependant, les valeurs de certains paramètres changent pendant le fonctionnement du robot.

Le but de cette section est de faire apparaître l'influence de ces phénomènes sur le comportement du robot manipulateur durant la prise et la manipulation des objets. Nous considérons l'influence de la variation paramétrique représentée dans notre cas par la variation de la masse de la charge, et des dynamiques non modélisées représentées ici par la présence des frottements.

### IV.2.1 Etude De L'influence De La Variation De La Masse De La Charge

Parmi les paramètres inertiels variables, la masse de la charge correspond au paramètre le plus prépondérant (Leahy et al., 1991). Par conséquent, nous considérons la variation de la masse de la charge comme paramètre variable suivant le profil de la figure IV.2. la simulation consiste à faire varier la masse du dernier corps selon le profil ci-dessous. Pour constater l'influence de la variation de la masse de la charge, la commande basée sur le modèle dynamique du robot ne prend pas en compte cette variation.

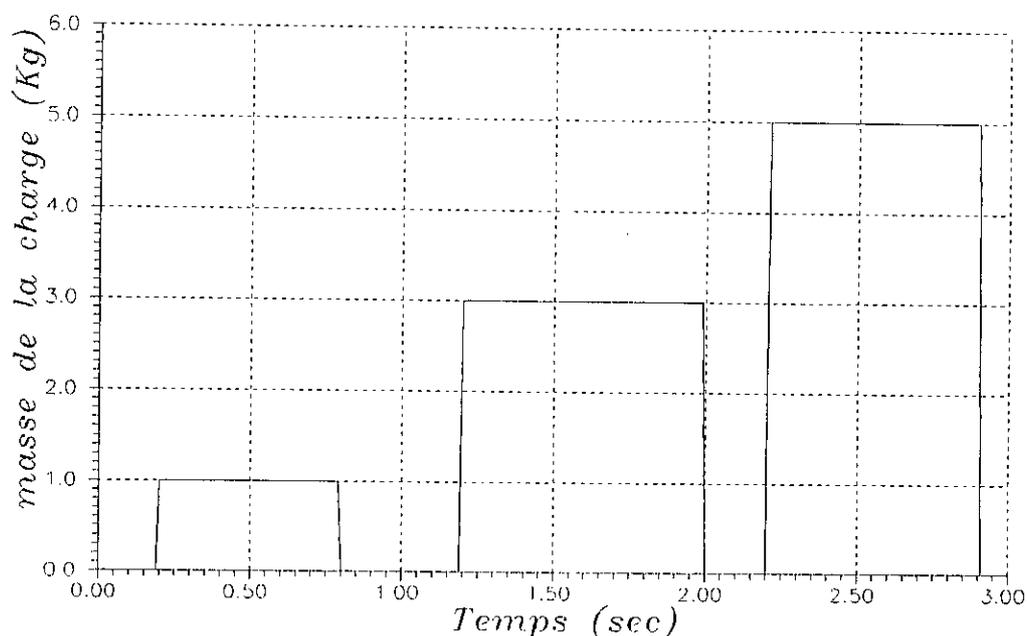
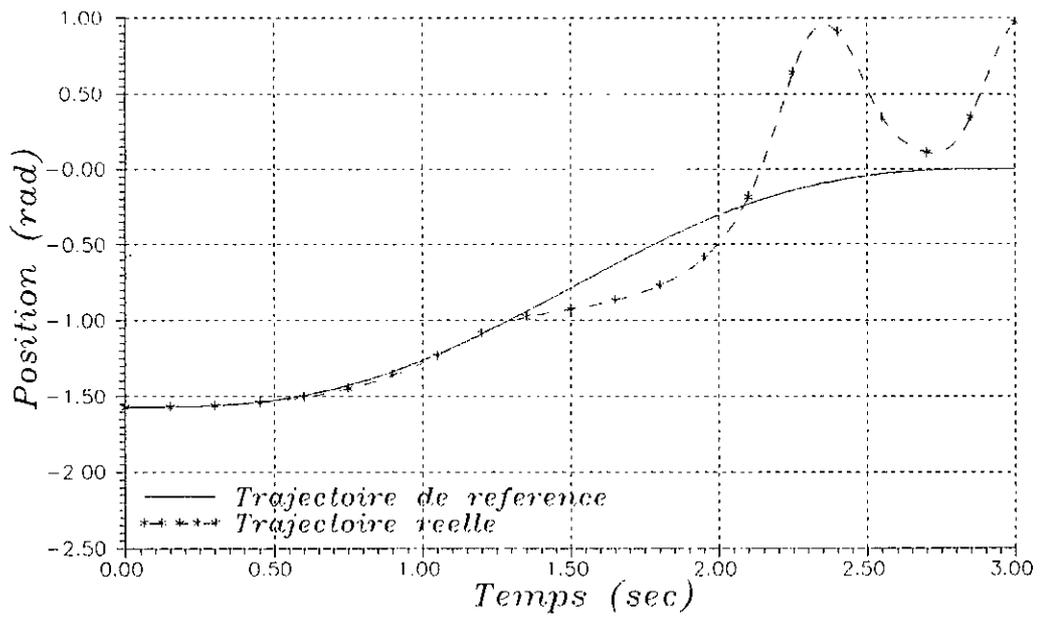
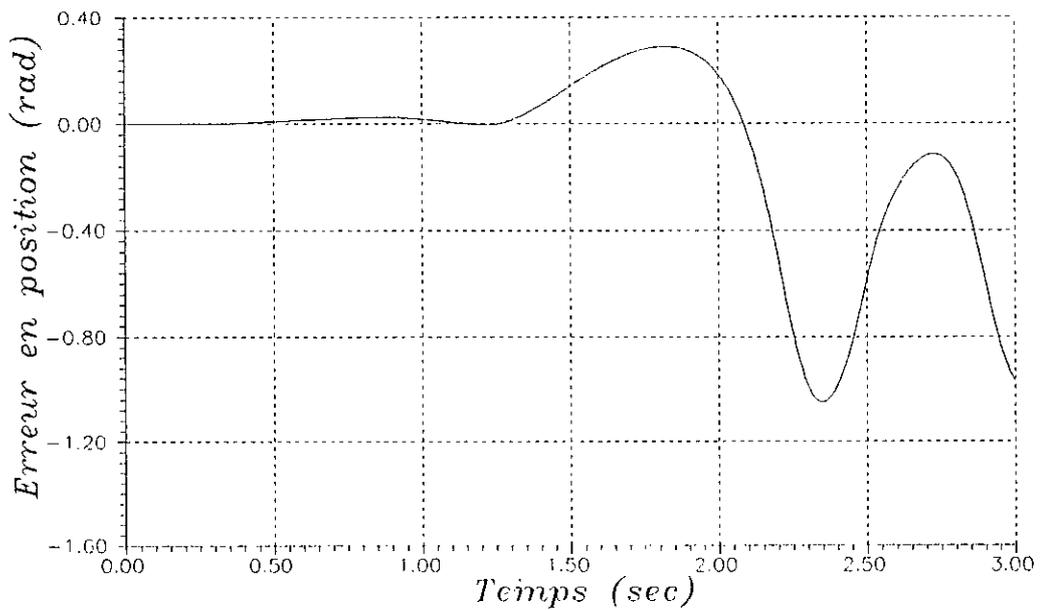


Figure IV.2. Profile de variation de la masse de la charge.

Les figures IV.3 à IV.6 reproduisent l'influence des variations des efforts d'inertie d'accélération, de coriolis et centrifuge introduites par la variation de la masse de la charge. Nous remarquons notamment que la dégradation du comportement des deux liaisons se fait progressivement pour des valeurs croissantes de la masse de la charge. Donc, à vide l'erreur de poursuite en position et en vitesse des deux liaisons est nulle. D'un autre côté si la masse de l'objet manipulé est assez faible, l'erreur de poursuite n'est pas très importante pour les deux liaisons. Cependant, dès que la masse de la charge devient importante la dégradation du comportement et par conséquent l'erreur de poursuite évolue d'une manière considérable (figures IV.3(b), IV.4(b), IV.5(b) et IV.6(b)). La comparaison des erreurs de poursuite en position et en vitesse des deux liaisons, nous permet de constater en particulier que l'influence de la variation de la masse de la charge sur la deuxième liaison est plus importante que sur la première liaison. Ceci est évident du fait que les dynamiques de deux liaisons sont différentes et le rapport entre le couple avec variation de la masse et le couple sans variation est plus important sur la deuxième liaison comme le montrent les figures VI.7a-VI.7c. A travers ces résultats de simulation nous constatons que l'influence de la variation paramétrique est très importante. Ceci nous conduit à la nécessité d'un couple d'anticipation qui permet de suivre la variation de la dynamique du robot manipulateur.



(a)

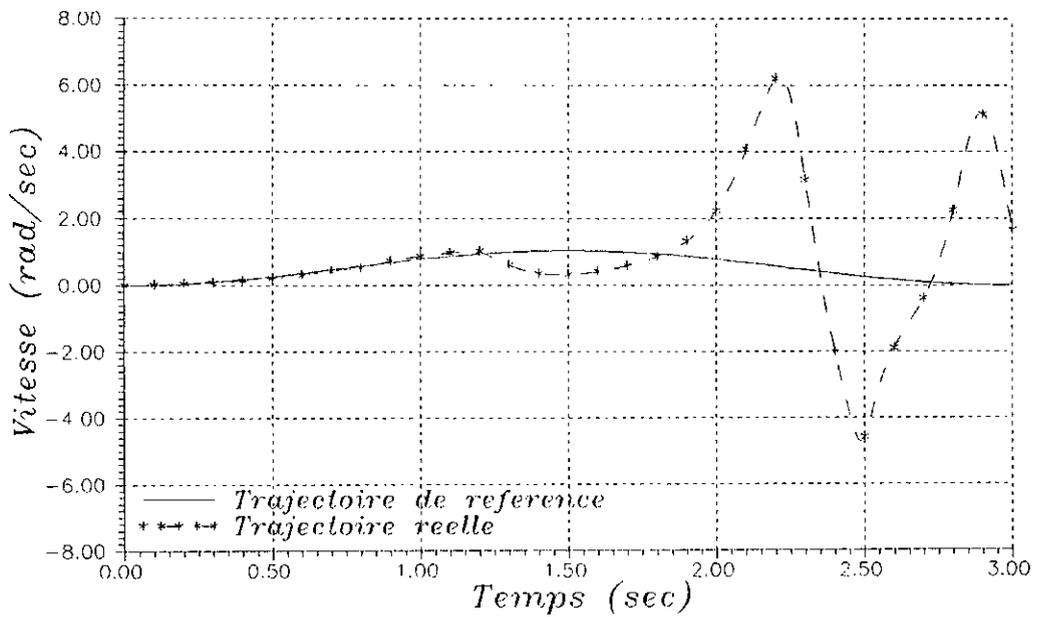


(b)

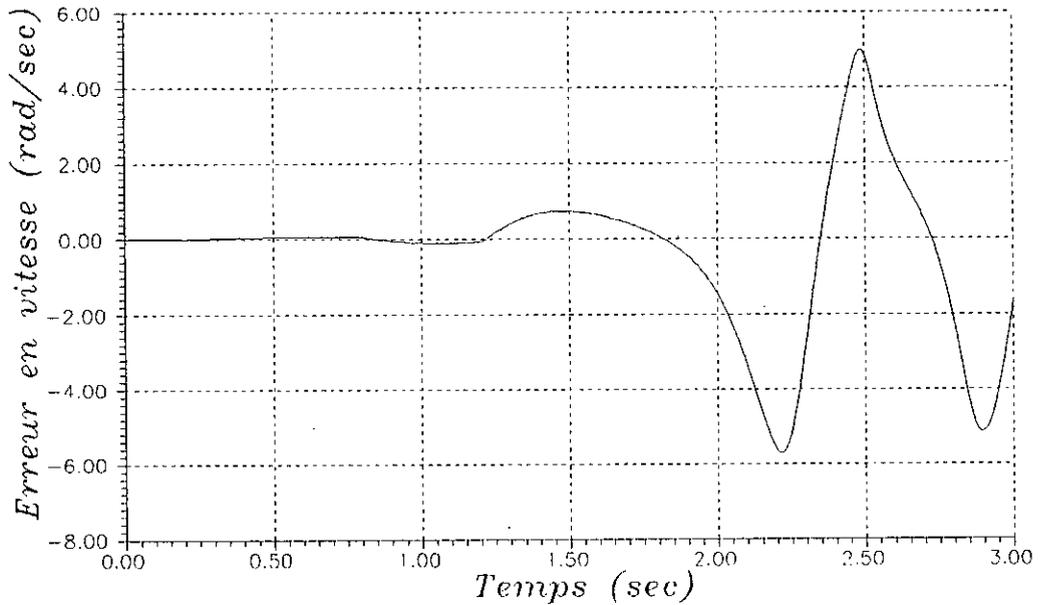
Figure IV.3. Influence de la variation de la masse de la charge sur la liaison 1.

(a) : trajectoire de position.

(b) : erreur de poursuite.



(a)

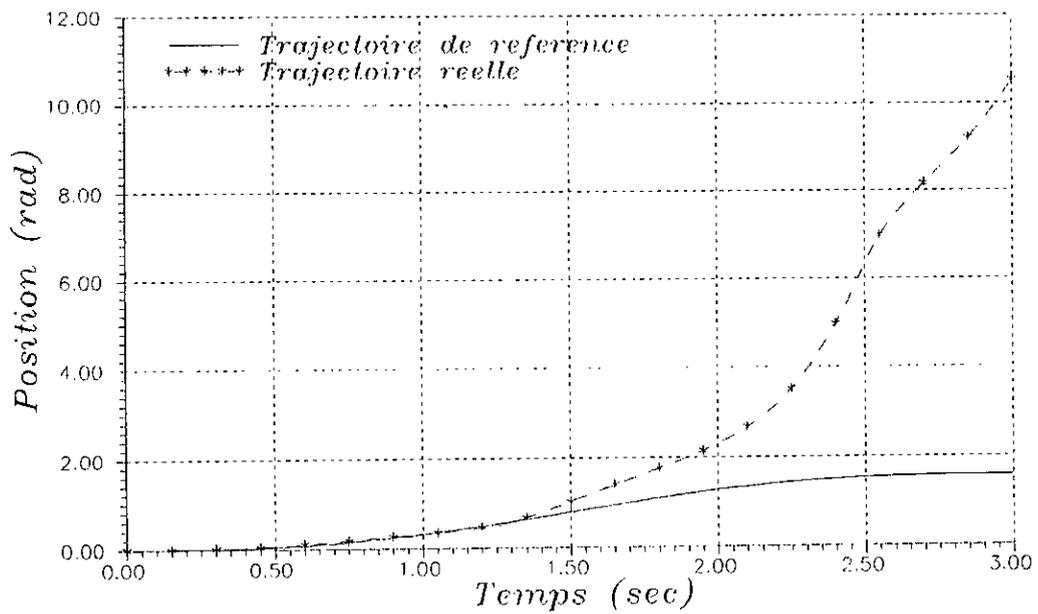


(b)

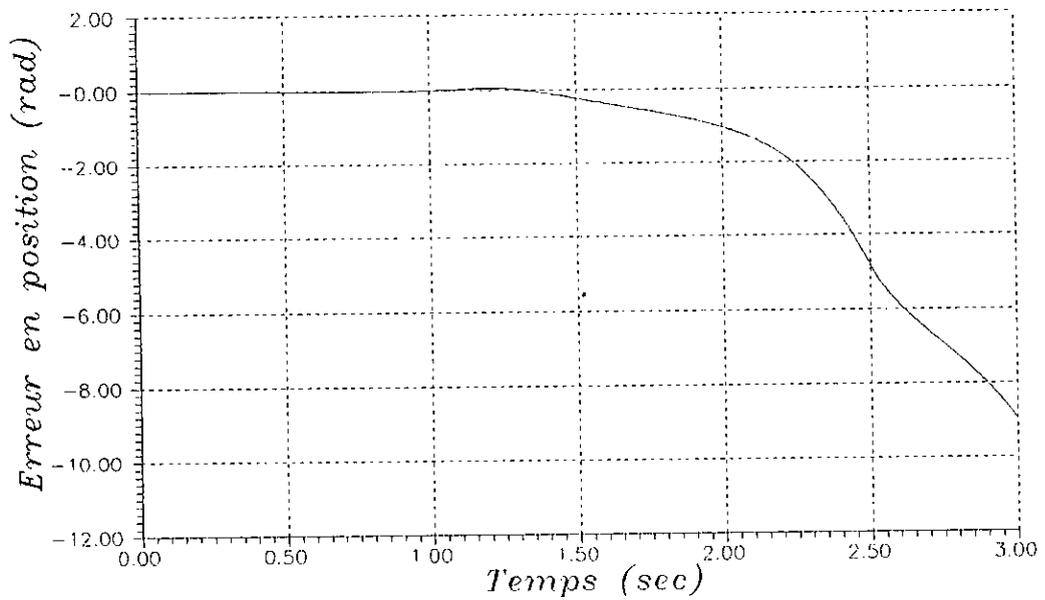
Figure IV.4. Influence de la variation de la masse de la charge sur la liaison I.

(a) : trajectoire de vitesse.

(b) : erreur de poursuite.



(a)

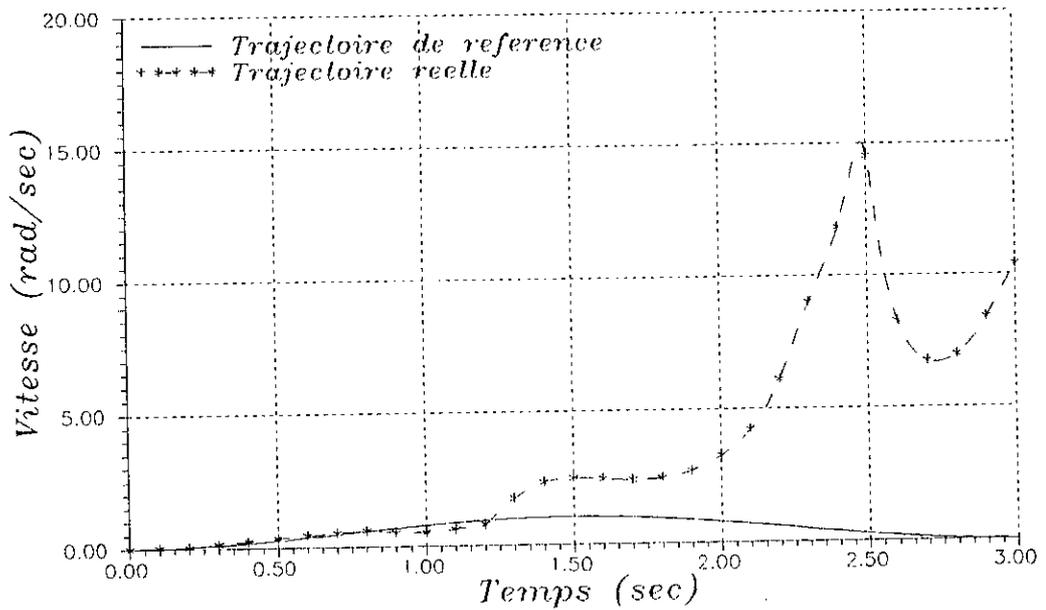


(b)

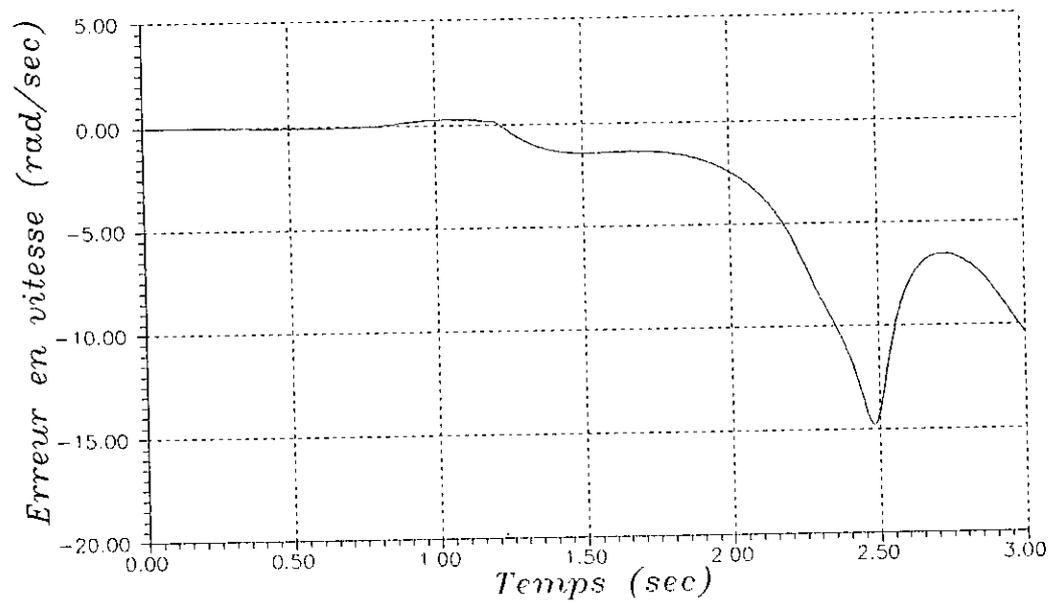
Figure IV.5. Influence de la variation de la masse de la charge sur la liaison 2.

(a) : trajectoire de position.

(b) : erreur de poursuite.



(a)

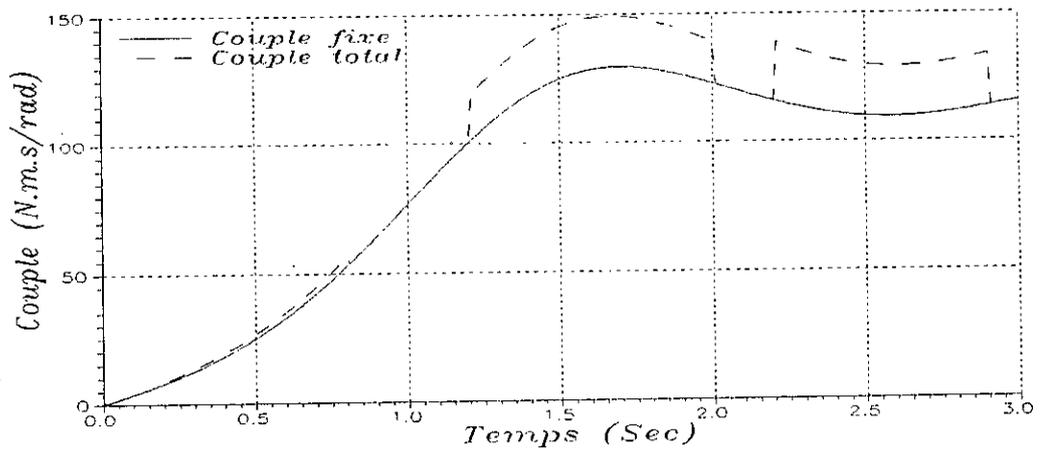


(b)

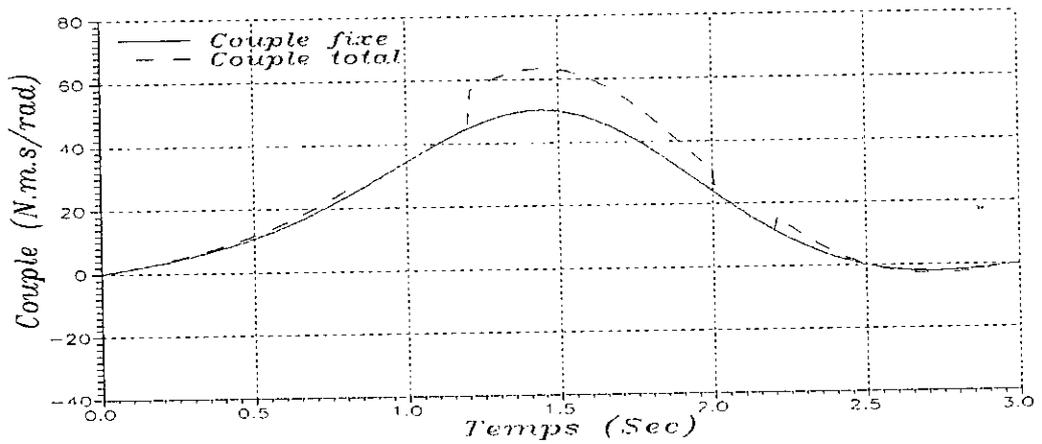
Figure IV.6. Influence de la variation de la masse de la charge sur la liaison 2.

(a) : trajectoire de vitesse.

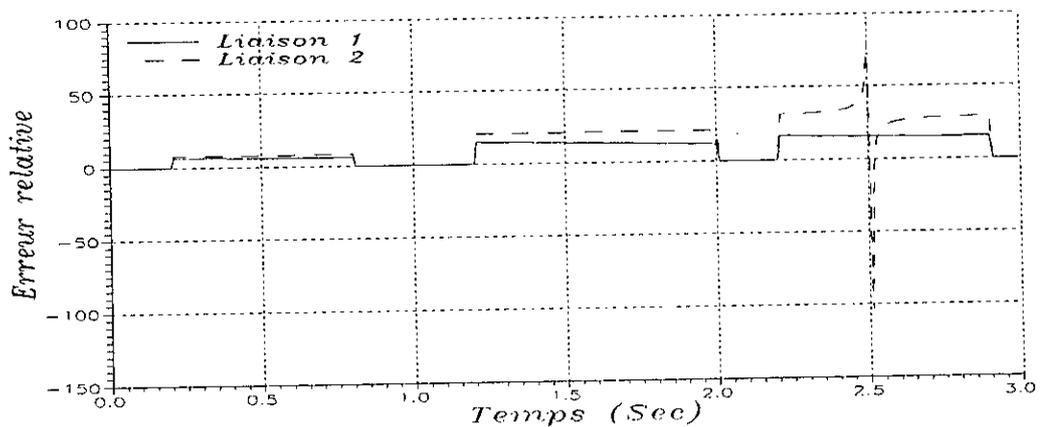
(b) : erreur de poursuite.



(a)



(b)



(c)

Figure VI.7 Influence de la variation de la masse de la charge sur les couple exercés.  
 (a) : Liaison 1 ; (b) : Liaison 2 ; (c) : Ecart relative entre les deux dynamiques

## IV.2.2 Etude De L'influence Des Frottements

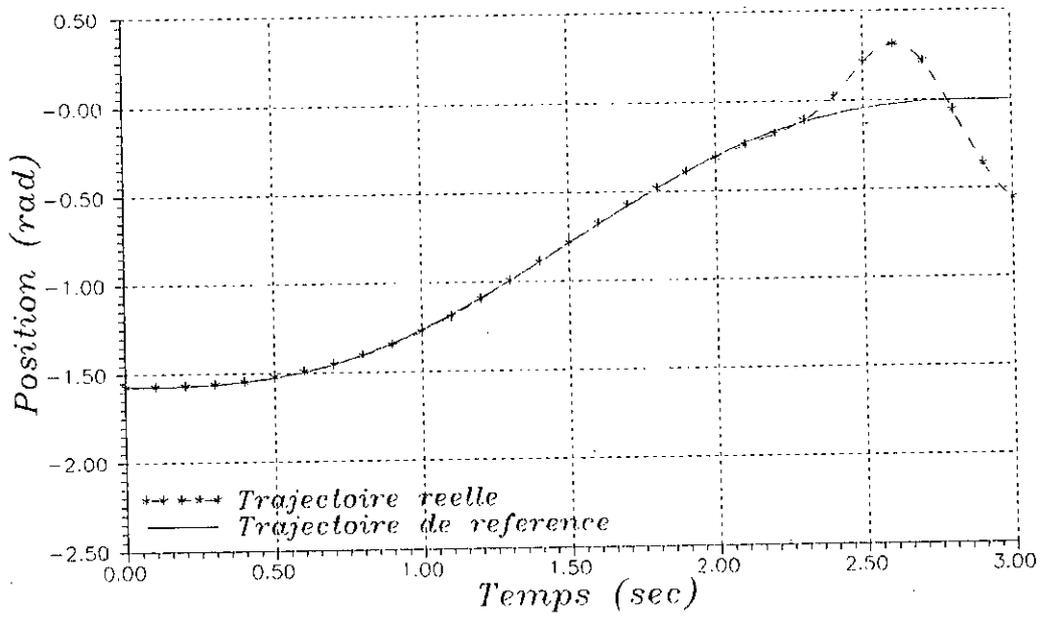
Afin de montrer la dégradation du comportement du robot manipulateur en fonction de la présence des frottements, nous effectuons une simulation dans laquelle les termes de frottements sont considérés comme une perturbation au système. La loi de commande reste, comme dans la cas de la variation paramétrique, basée sur le modèle dynamique sans inclure le modèle des frottements. nous considérons le modèle de frottements le plus utilisé dans la littérature (Pham, 1992) :

$$\tau_f = f_s \text{Sign}(\dot{q}) + f_v \dot{q}$$

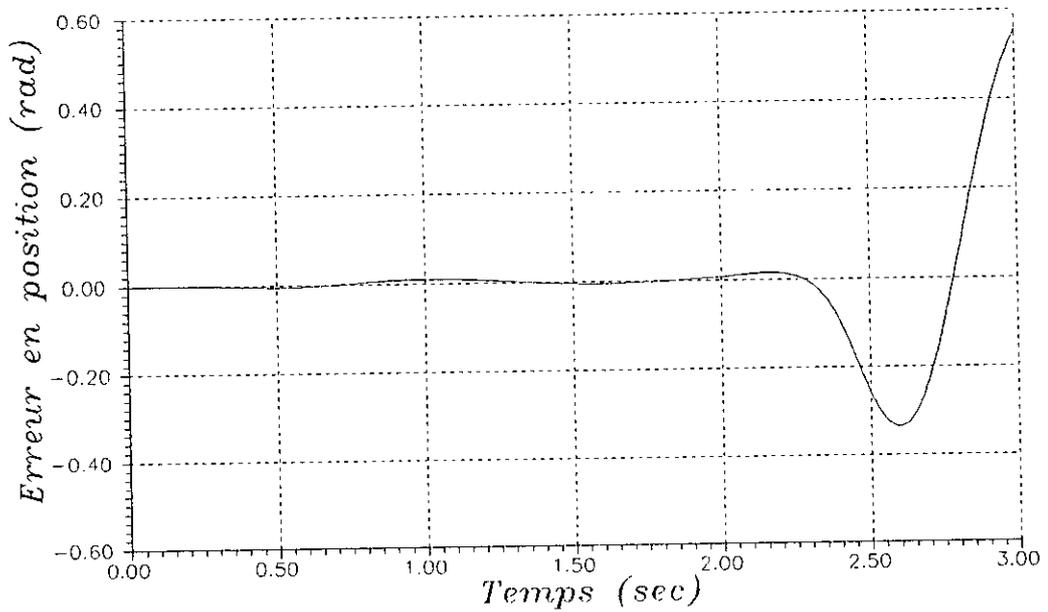
avec  $f_s = 0.5 \text{ Nt.m/rad.s}^{-1}$ : coefficient de frottements secs;  
 $f_v = 1.0 \text{ Nt.m/rad.s}^{-1}$ : coefficient de frottements visqueux.

Les figures IV.8 à IV.11 montrent l'allure des trajectoires de position et de vitesse des deux liaisons du robot considéré. Nous remarquons qu'à très faible vitesse les frottements ont une assez faible influence sur le comportement dynamique du robot. Cependant, dès que la vitesse du bras avoisine la valeur de 1 rad/sec la dégradation du comportement dynamique du robot devient importante.

De plus en comparant les figures IV.8(a) et IV.10(a) avec les figures IV.9(a) et IV.11(a); respectivement; des trajectoires de position et de vitesse des deux liaisons, nous constatons, comme dans le cas de variation de la masse, que l'influence des frottements sur la deuxième liaison est plus importante que sur la première liaison, et ceci est d'autant vrai que les dynamiques associés à chacune des deux liaisons sont différentes. Cette différence apparaît clairement sur les figures IV.12 à IV.13 où nous constatons que le rapport entre le couple avec frottements et le couple sans frottements est nettement plus important sur la deuxième liaison. Nous pouvons aussi conclure que les frottements ont une dynamique non négligeable dans ce cas. Ceci est bien connu pour les robots industriels (An, 1986). Pour le cas des robots à moteur sans réducteur, les frottements sont généralement négligés. Cependant leurs effets restent loin d'être négligeables (Kozlowski and Prüfer, 1992 ; Lu et al., 1993). D'un autre côté, la modélisation des frottements est très difficile (Pham, 1992), et afin d'éviter le besoin de les modéliser nous proposons de les compenser à travers une action de retour adaptative.



(a)

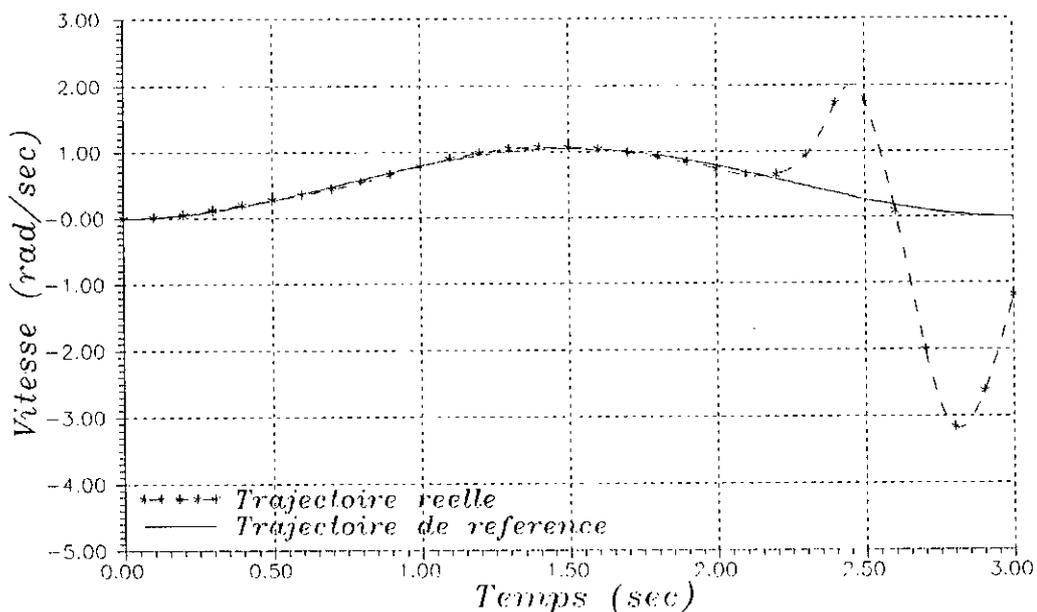


(b)

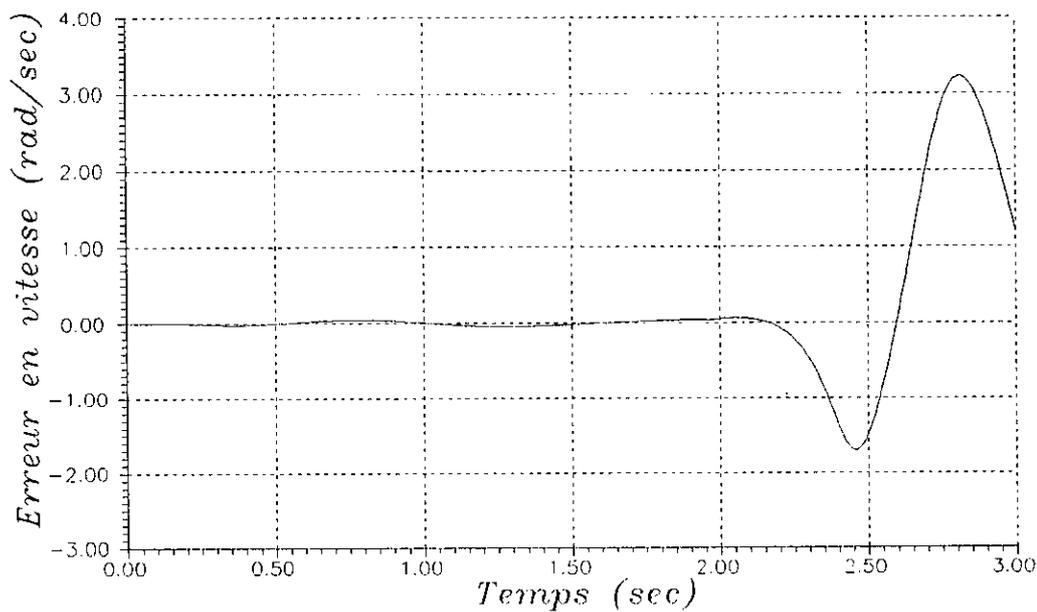
Figure IV.8. Influence des frottements sur la liaison 1.

(a) : trajectoire de position.

(b) : erreur de poursuite.



(a)

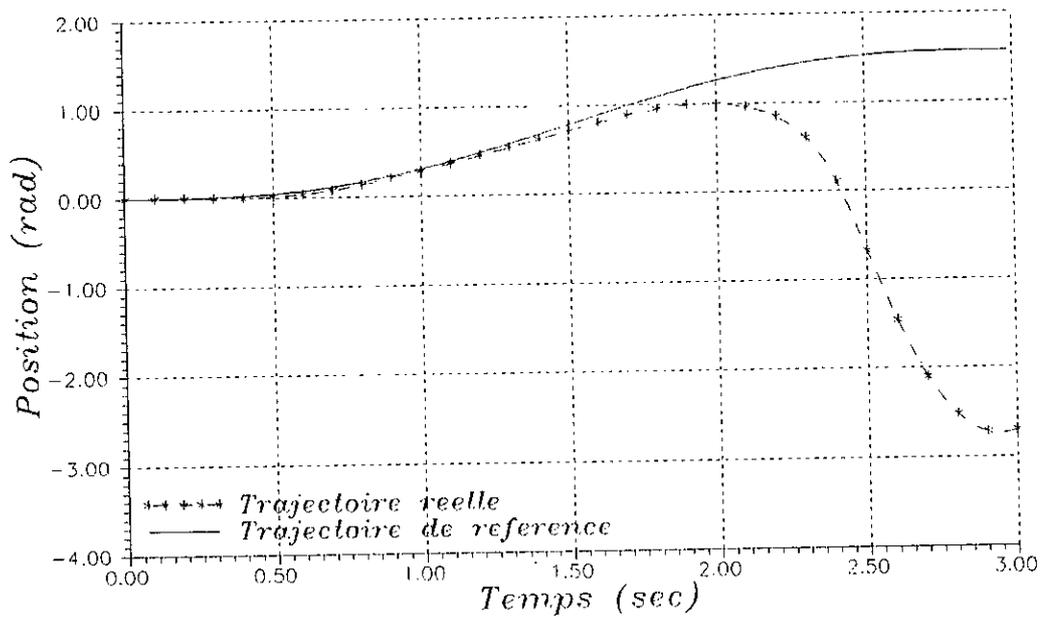


(b)

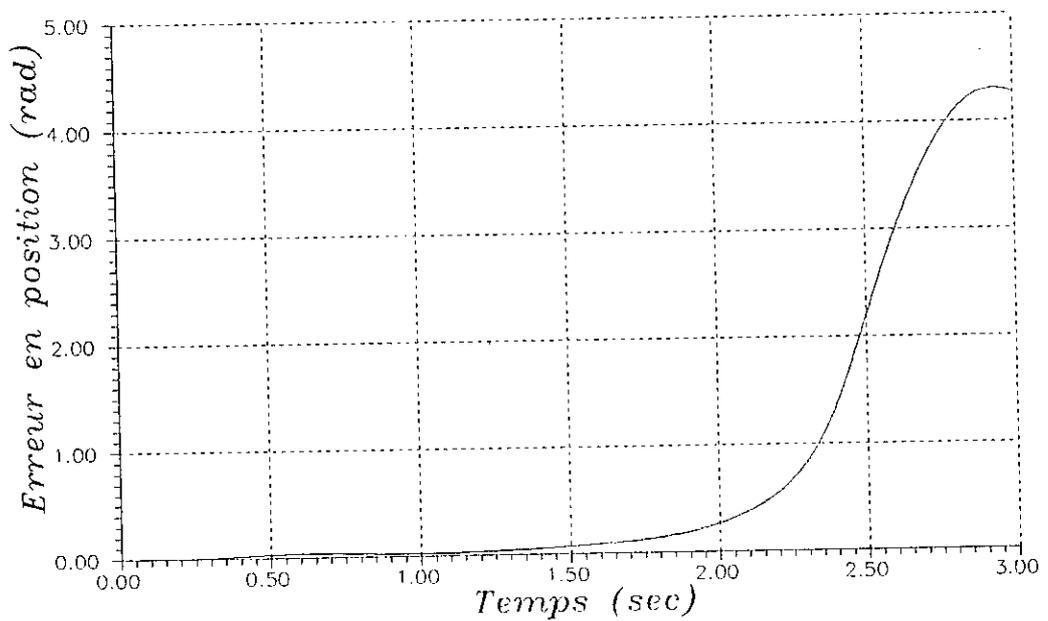
Figure IV.9. Influence des frottements sur la liaison 1.

(a) : trajectoire de vitesse.

(b) : erreur de poursuite.



(a)

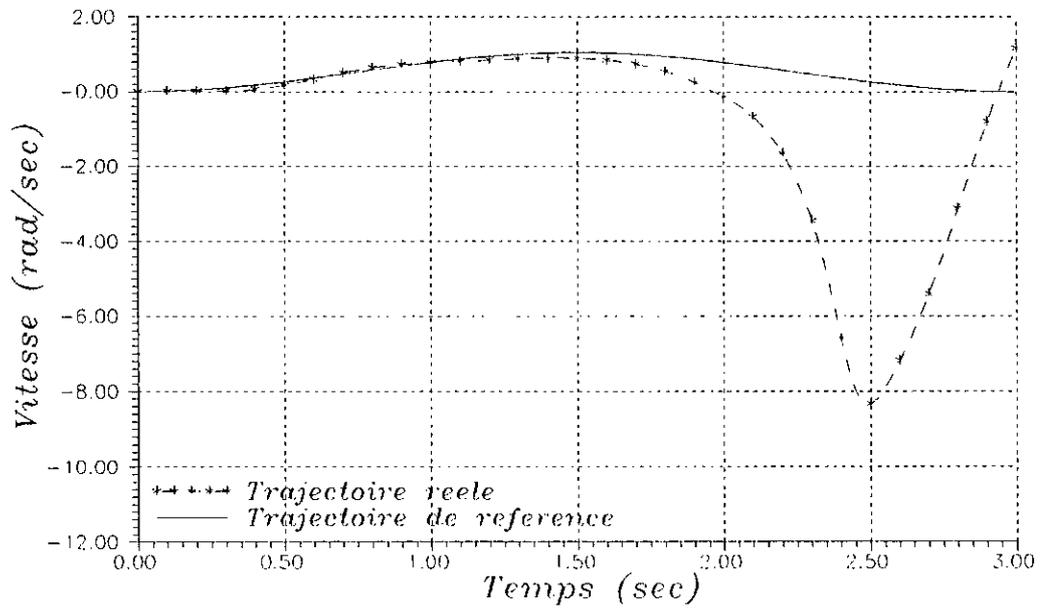


(b)

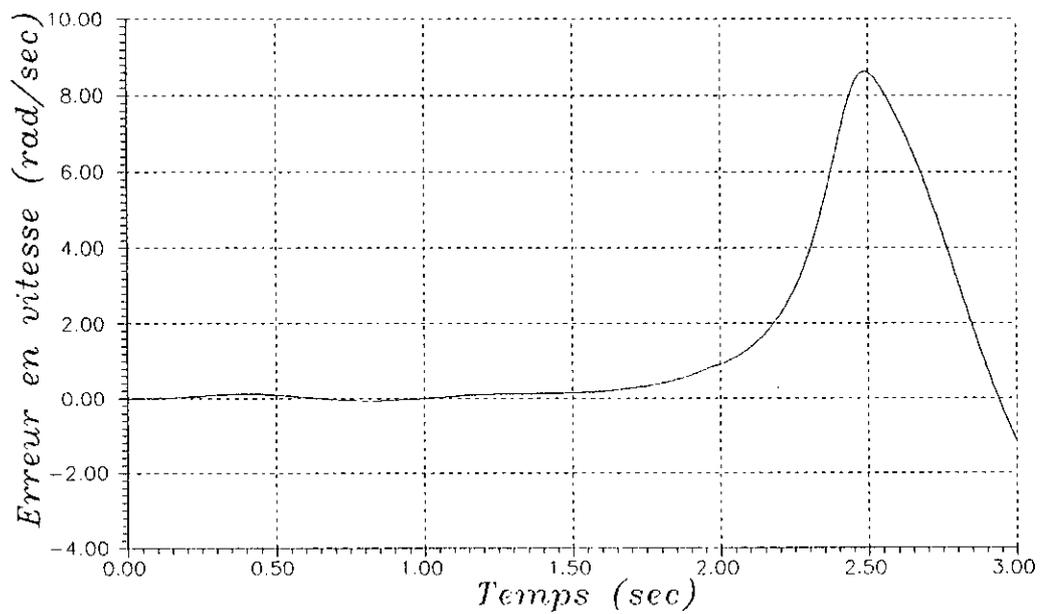
Figure IV.10. Influence des frottements sur la liaison 2.

(a) : trajectoire de position.

(b) : erreur de poursuite.



(a)

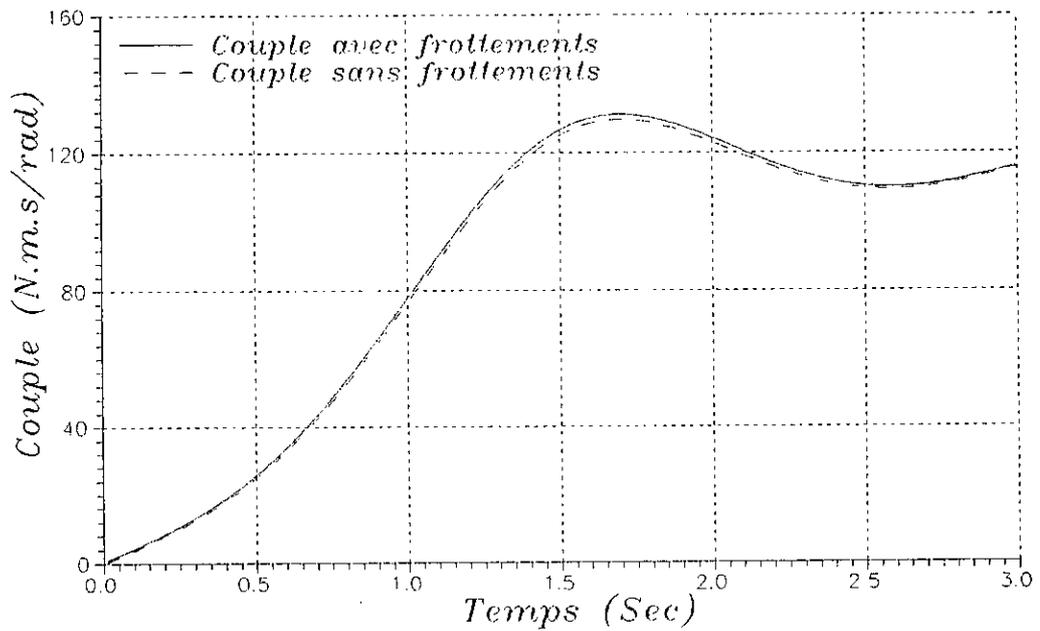


(b)

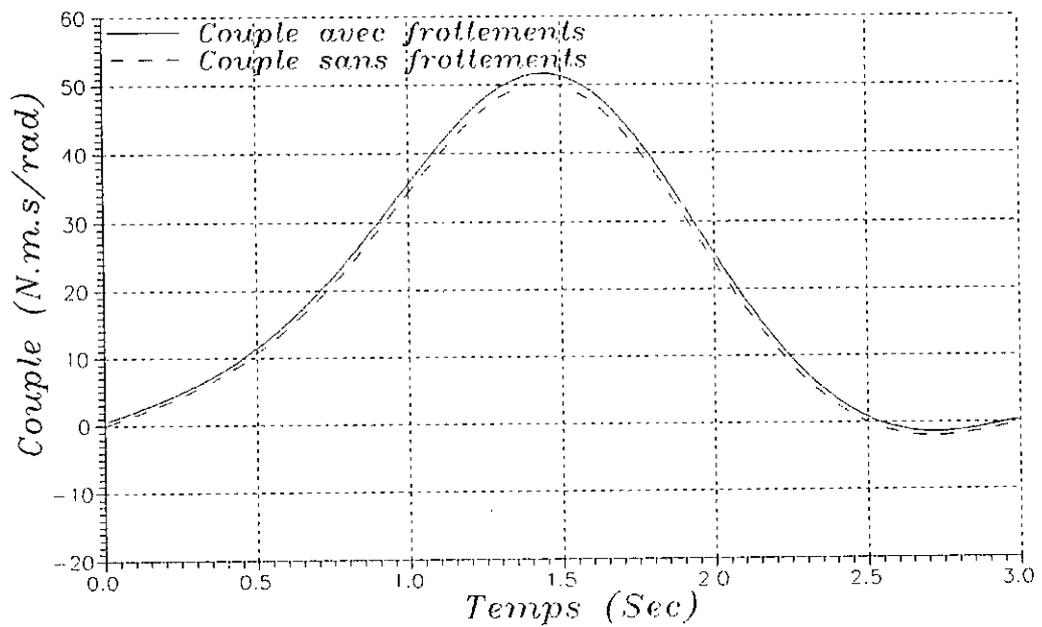
Figure IV.11. Influence des frottements sur la liaison 2.

(a) : trajectoire de vitesse.

(b) : erreur de poursuite.



(a)



(b)

Figure IV.12. Influence des frottements sur les couples exercés.

(a) : Liaison 1.

(b) : Liaison 2.

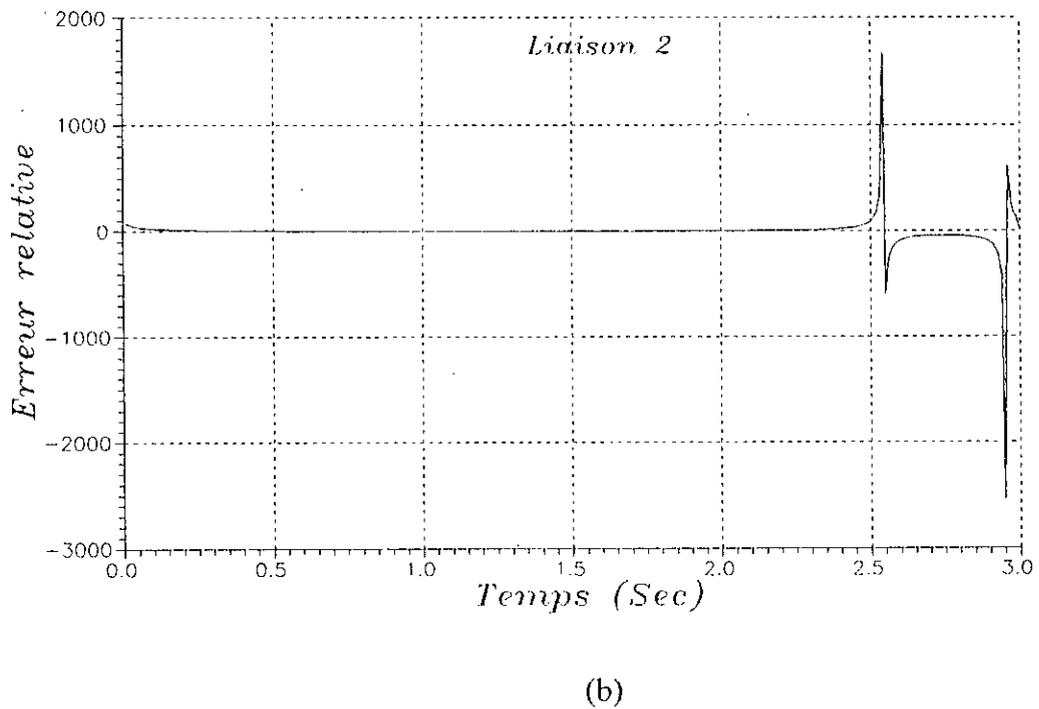
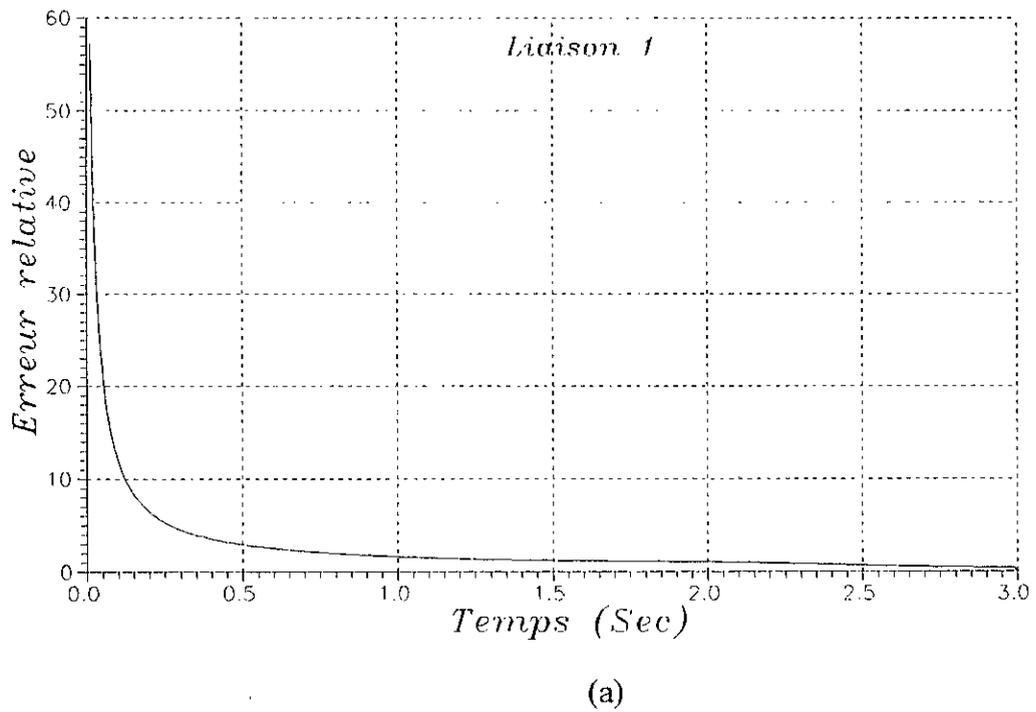


Figure IV.13. Erreur relative des couples exercés.

(a) : Liaison 1.

(b) : Liaison 2.

## IV.3 Commande Neuro-Dynamique Adaptative

Dans cette section nous allons valider notre solution de commande par simulation. Nous supposons la masse de la charge comme paramètre variable qui sera estimé par l'estimateur neural et les frottements comme dynamique non modélisées compensés par la correction PD adaptative.

### IV.3.1 Condition de Simulation

Pour tout le reste des simulations présentées dans cette section nous nous sommes mis sous les conditions suivantes :

- La masse de la charge est considérée comme paramètre variable selon le profil montré en figure IV.2
- Les frottements, dont les valeurs des coefficients données au § IV.2.1; sont considérés comme étant une dynamique non modélisée,
- Nous utilisons des réseaux de neurones multicouches pour la stratégie neuro-dynamique,
- Les signaux entrées de tout les réseaux de neurones sont normalisées dans l'intervalle  $[-0.8 \ 0.8]$  afin d'éviter la zone de saturation de la fonction d'activation des neurones. Cette normalisation nous permet aussi d'accélérer la convergence des réseaux,
- Nous avons utilisé des données aléatoires uniformément distribuées comme signaux d'apprentissage des réseaux,
- La simulation se fait sous Matlab en utilisant le toolbox de réseaux de neurones,
- Les équations différentielles sont intégrées à l'aide de l'algorithme de Rung-Kutta d'ordre 4 de Matlab avec une période d'échantillonnage de 1 ms.
- La trajectoire de référence est une trajectoire cycloïde donnée par l'expression suivante :

$$q_1(t) = -\frac{\pi}{2} + \frac{1}{4} \left[ \frac{2\pi t}{3} - \sin \frac{2\pi t}{3} \right] \text{ rad}, \quad 0 \leq t \leq 3$$

$$= 0, \quad 3 < t$$

$$q_2(t) = q_1(t) + \frac{\pi}{2}$$

- Le schéma général de la simulation est donné en figure IV.14.

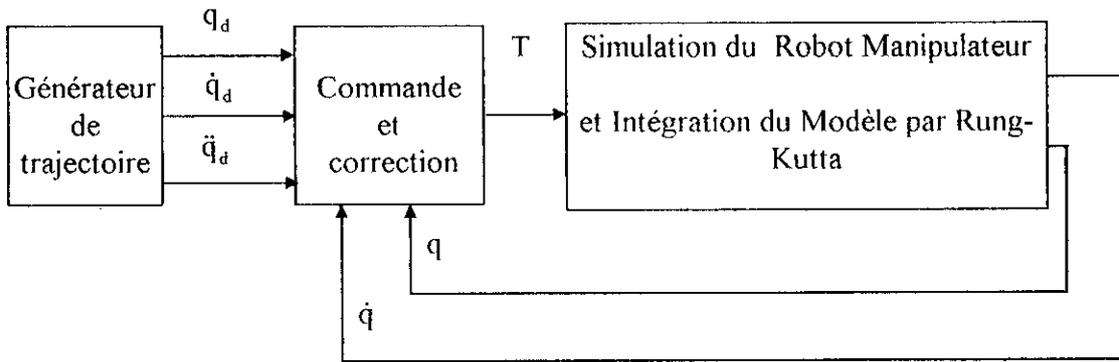


Figure IV.14. Schéma général de simulation.

La décomposition du modèle dynamique en parties dynamiques fixe et variable, en fonction la masse variable de la charge, aboutit aux expressions suivantes :

- Modèle dynamique fixe

$$T_f = M_f(q)\ddot{q} + C_f(q, \dot{q}) + G_f(q)$$

avec  $M_f$ ,  $C_f$  et  $G_f$  gardent les mêmes expressions que celles de  $M$ ,  $C$  et  $G$  données dans (§ IV.2) avec :

$$a_{1f} = i_1 + i_2 + (m_1 + m_{2f})L_1^2 + m_{2f}L_2^2, \quad a_{2f} = i_2 + 2m_{2f}L_1L_2, \quad a_{3f} = m_{2f}L_2^2$$

$$a_{4f} = (m_1 + m_{2f})gL_1, \quad a_{5f} = m_{2f}gL_2$$

- Modèle dynamique variable

$$T_v = M_v(q)\ddot{q} + C_v(q, \dot{q}) + G_v(q)$$

avec  $M_v$ ,  $C_v$  et  $G_v$  gardent les mêmes expressions que celles de  $M$ ,  $C$  et  $G$  données dans (§ IV.2) avec :

$$a_{1v} = m_{2v}(L_1^2 + L_2^2), \quad a_{2v} = 2m_{2v}L_1L_2, \quad a_{3v} = m_{2v}L_2^2, \quad a_{4v} = m_{2v}gL_1, \quad a_{5v} = m_{2v}gL_2$$

### IV.3.2 Commande Neurale Dynamique Anticipée Adaptative

L'objectif de ce paragraphe est d'évaluer les performances du schéma de commande de la figure III.3 du troisième chapitre. Nous rappelons que ce schéma constitue la première variante de l'approche de commande proposée.

Les couples fixes des deux liaisons sont préalablement calculés hors ligne pour chaque cycle de commande puis stockés dans une table. Le couple variable est fourni par un réseau de neurones multicouche ayant deux couches cachées.

Pendant l'étape d'apprentissage du réseau, nous avons choisi au début un seul réseau pour tout le couple variable. Le réseau avait pour entrées les trajectoires désirées de position, vitesse et accélération des deux liaisons et l'estimée de la masse. Le système ainsi considéré était assez complexe ce qui nous a conduit à une taille du réseau considérable dans les deux couches cachées (entre 30 à 40 neurones par couche) et d'un autre côté l'erreur convergait très lentement jusqu'à  $10^{-2}$ .

A l'étape de généralisation nous avons constaté que l'erreur entre le couple variable désiré et celui fourni par le réseau de neurones était assez importante. Afin de surmonter ce problème nous avons considéré des réseaux de neurones indépendants pour chaque liaison. Cette solution nous a permis d'avoir une taille des réseaux raisonnable (entre 20 et 30 neurones par couche), ce résultat était attendu vu que la taille du réseau augmente avec la complexité du système (Katić and Vokobratić, 1992). Cependant, malgré une bonne convergence des réseaux par rapport au premier cas, dans l'étape de généralisation l'erreur entre le couple désiré et le couple fourni par le réseau reste assez importante. Au vu de ces résultats nous avons tiré profit de la caractéristique de la linéarité du modèle dynamique par rapport aux paramètres. Ainsi au lieu d'utiliser un réseau pour fournir la totalité du couple variable, le réseau représente dans ce cas la matrice d'information. Le couple variable sera obtenu par un simple produit de la sortie du réseau avec la variation de la masse estimée. Par conséquent nous avons pu réduire la taille du réseau (entre 20 neurones pour la première couche cachée et 13 neurones pour la seconde). Le réseau convergait en moins de temps par rapport aux deux cas précédents, et l'erreur de généralisation entre le couple variable désiré et celui fourni par le réseau est de l'ordre de  $10^{-3}$ . Ceci étant satisfaisant et nous avons alors opté pour cette architecture des réseaux de neurones.

Pour le compensateur PD adaptatif, le problème majeur consiste en le choix des paramètres d'adaptation. Afin de résoudre ce problème nous avons effectué plusieurs essais qui nous ont conduit aux paramètres suivants :

liaison	$G_p$	$G_v$	$k_{p0}$	$k_{v0}$	$r_0$	$\alpha_0$	$\alpha_1$
1	40	30	40	30	0	100	800
2	40	30	40	30	0	100	800

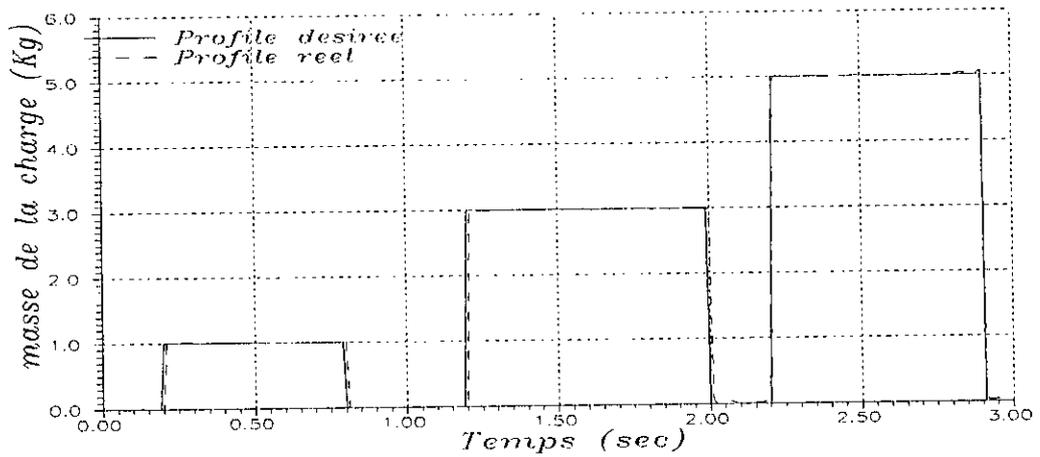
Pour fournir l'estimation de la variation de la masse de la charge nous avons utilisé un réseau de neurones qui a été entraîné, sur la base des données de position, vitesse,

accélération et couples exercés sur chaque liaison, en utilisant l'algorithme de la rétropropagation. Le réseau de neurones est constitué de deux couches cachées ayant respectivement 20 et 16 neurones.

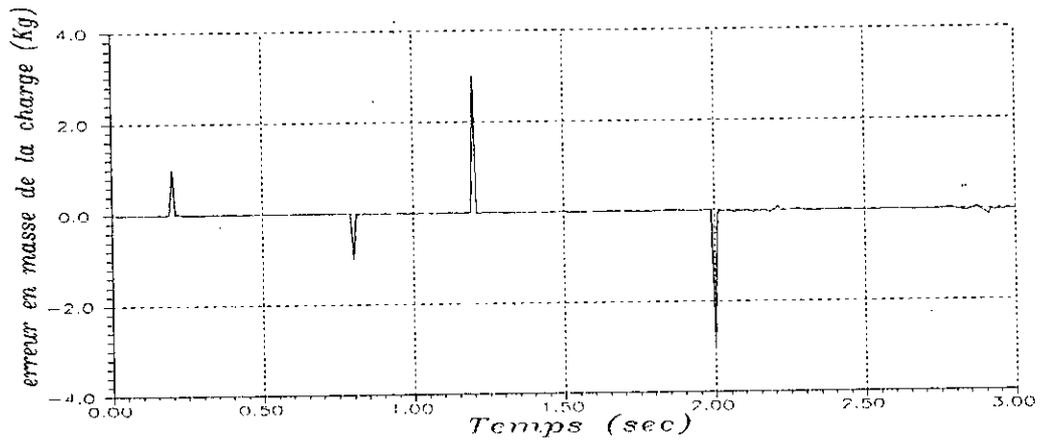
Pour normaliser les données d'entrées au réseau de neurone, on n'utilise pas comme d'habitude le même facteur de normalisation pour toutes ces entrées (Newton and Xu, 1993) car dans le cas présent le problème majeur pendant l'apprentissage de l'estimateur neurale reste la grande différence entre la plage des variations des données de position, vitesse et accélération avec la plage de variation des couples exercés. Ainsi, si on considère la valeur maximale des couples comme facteur de normalisation, les données de position, vitesse et accélération deviennent trop faibles pour être significatives. Afin de surmonter ce problème nous avons choisi d'utiliser deux différents facteurs de normalisation, l'un pour les données de position, vitesse et accélération. L'autre facteur est affecté aux couples exercés sur chaque liaison.

Après l'étape d'apprentissage et la convergence du réseau jusqu'à une erreur d'apprentissage de l'ordre de  $10^{-3}$ , l'estimateur neurale fournit, sur la base des données de position, vitesse, accélération et couples exercés à l'instant  $k$ , une valeur estimée de la masse de la charge pour le prochain cycle de commande ( c'est-à-dire l'instant  $(k+1)$  ).

La sortie de l'estimateur neurale de la masse de la charge, comme le montre la figure IV.15(a), suit parfaitement la variation réelle de cette masse.



(a)



(b)

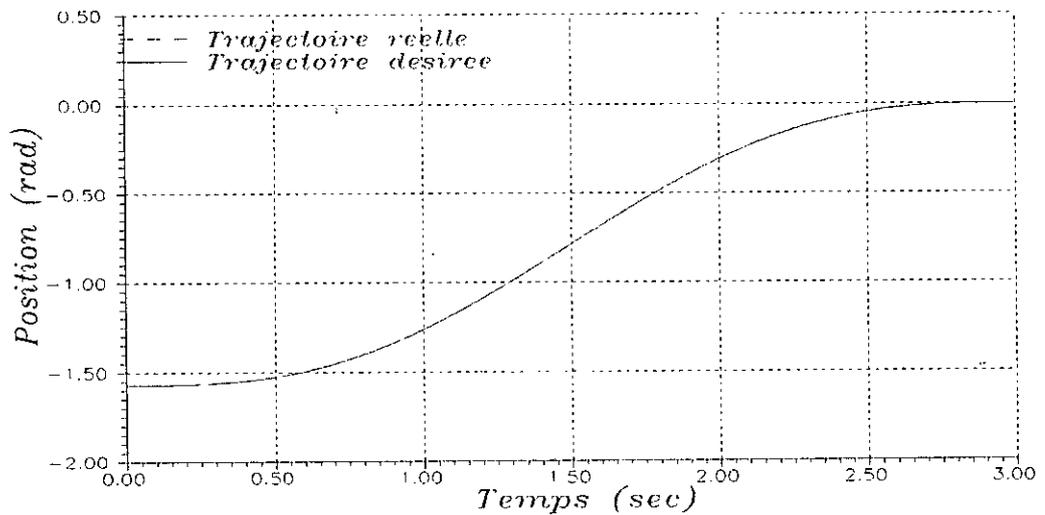
Figure IV.15. Estimation de la variation de la masse de la charge.

(a) : Profils réel et estimé.

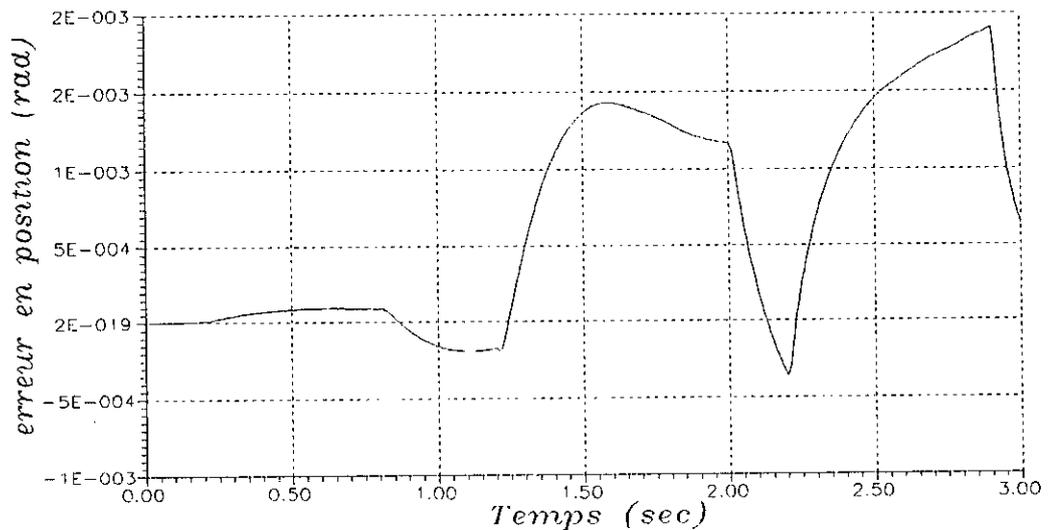
(b) : Erreur d'estimation.

La figure IV.15(b) indiquant l'erreur d'estimation, montre des pics qui correspondent à des transitions brusques de la variation de la masse. Cet effet peut être expliqué par le fait que l'estimateur neural n'est qu'un observateur qui tend à suivre la variation des paramètres dynamique du système. L'estimateur suit cette variation après un certain retard. Toutefois nous constatons que ce retard n'est pas très important ce qui nous permet de conclure que l'estimateur neural peut suivre très rapidement les changements de la valeur de la masse de la charge. Cette constatation est également confirmée par les travaux de Yegerlehner dans (Yegerlehner and Meckl, 1993).

En termes de poursuite de la trajectoire les résultats de simulation (figures IV.16 à IV.19) montrent que les performances de l'approche proposée sont très satisfaisantes et nous constatons en particulier une bonne poursuite de la trajectoire désirée. Les erreurs de poursuite en vitesse sont plus importantes que celles en position; mais ceci reste dans des proportions acceptables. Sur les figures IV.17(b) et IV.19(b) nous remarquons l'existence de pics qui correspondent aux transitions de la variation de la masse et par conséquent à l'erreur d'estimation. Ces pics d'erreur sont atténués par l'action PD adaptative puis réduits puisque l'erreur d'estimation de la masse redevient nulle.



(a)

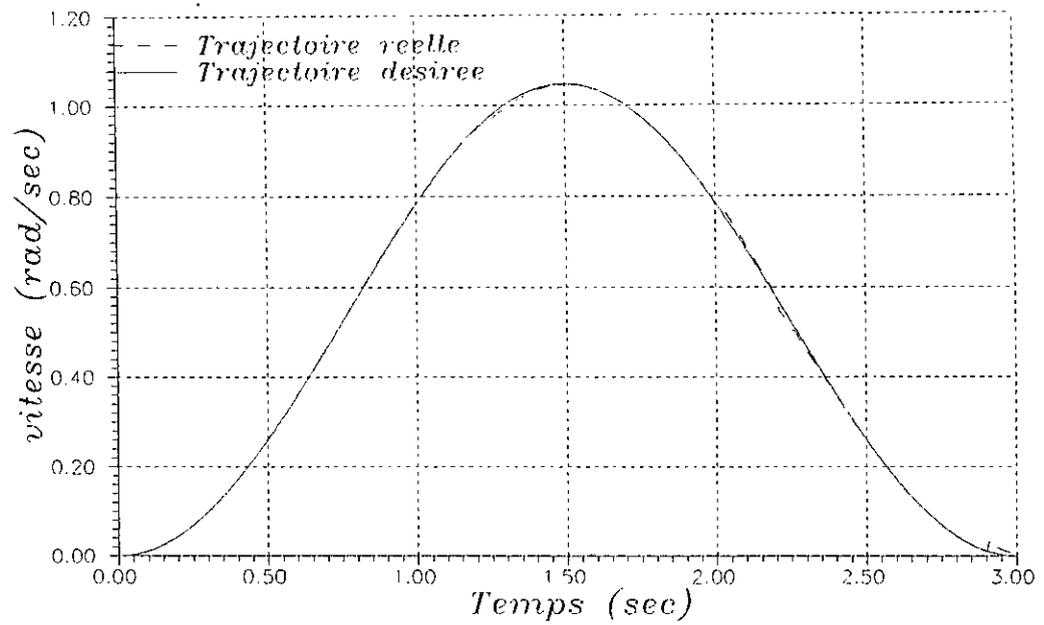


(b)

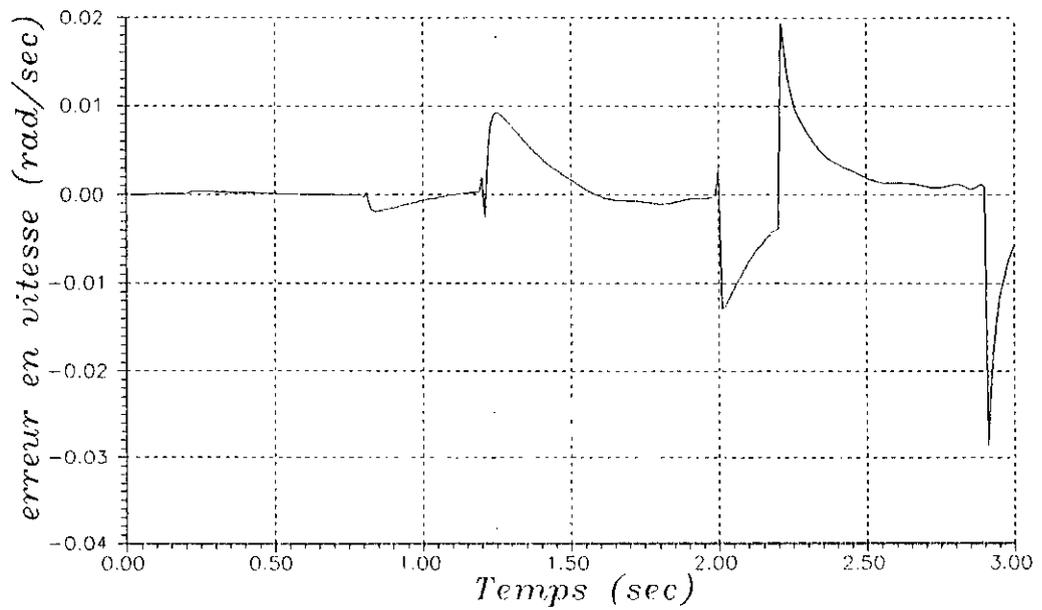
Figure IV.16. Commande Neurale Dynamique Anticipée Adaptative.

(a) : trajectoire de position (Liaison 1).

(b) : erreur de poursuite (Liaison 1).



(a)

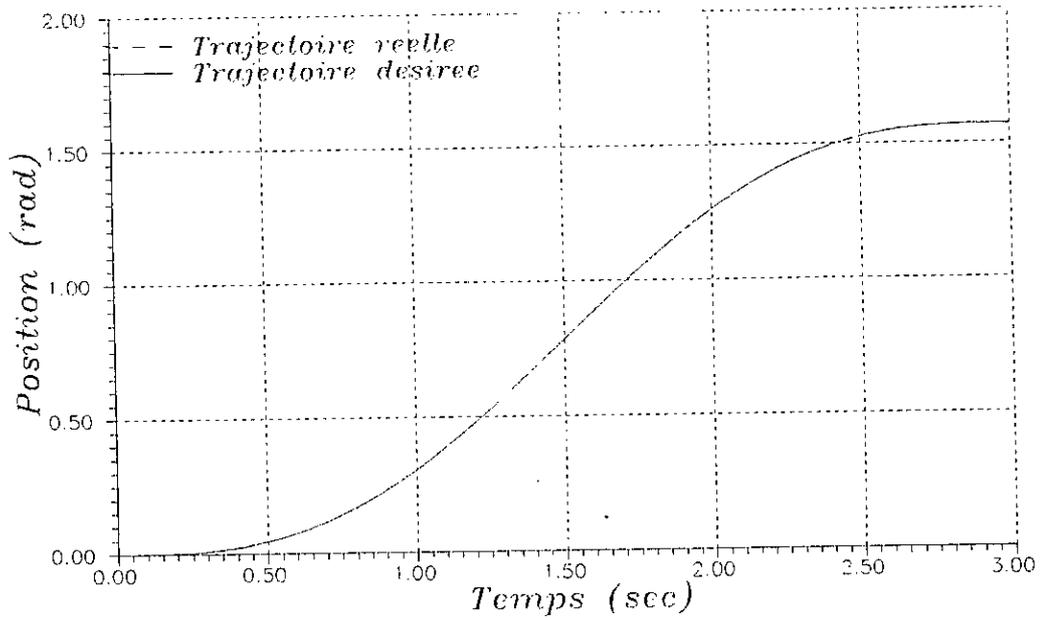


(b)

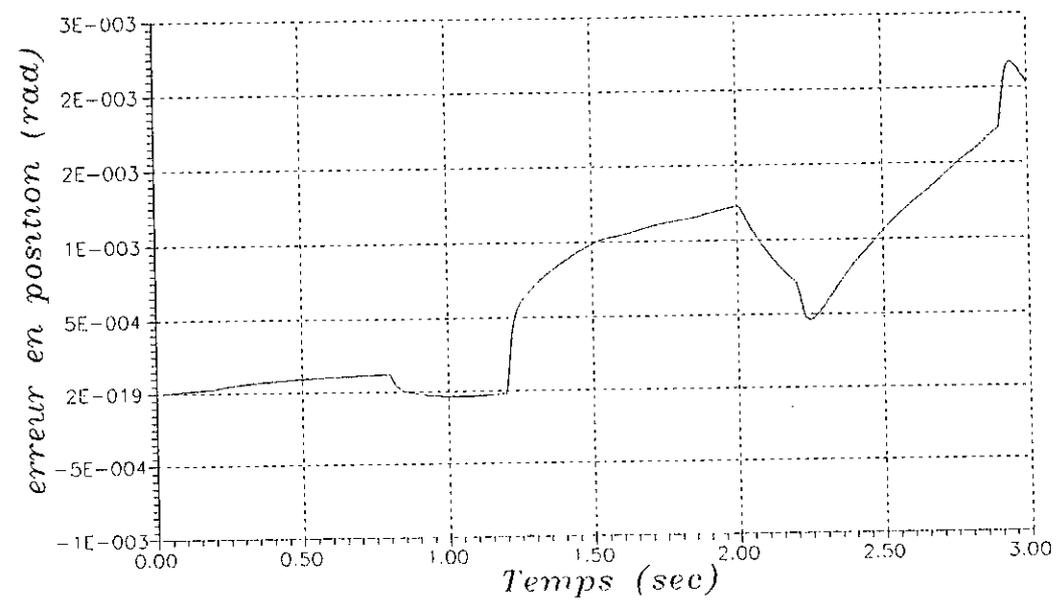
Figure IV.17. Commande Neurale Dynamique Anticipée Adaptative.

(a) : trajectoire de vitesse (Liaison 1).

(b) : erreur de poursuite (Liaison 1).

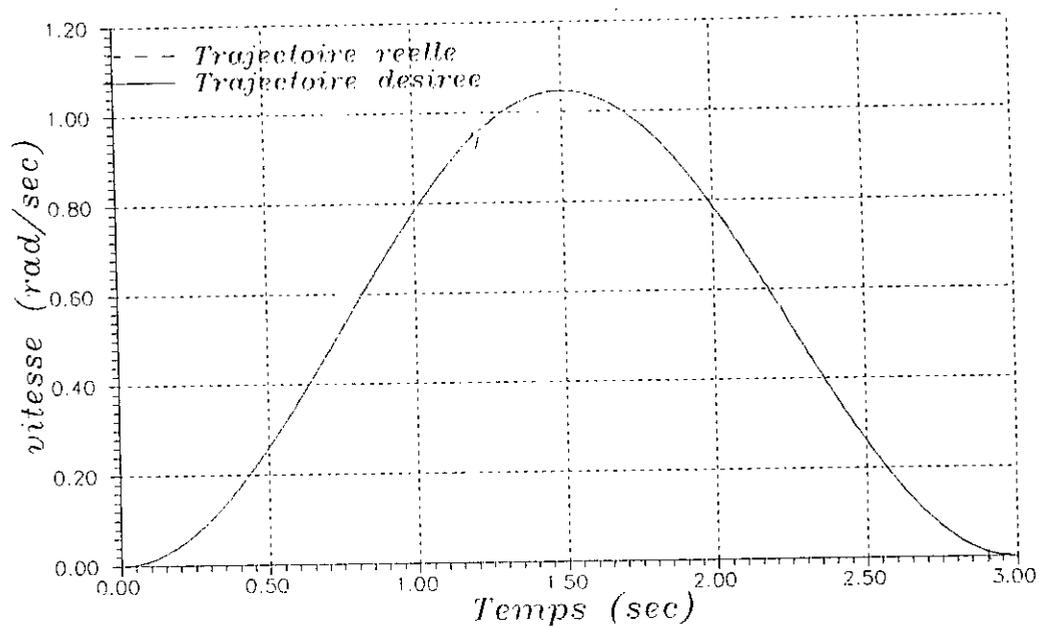


(a)

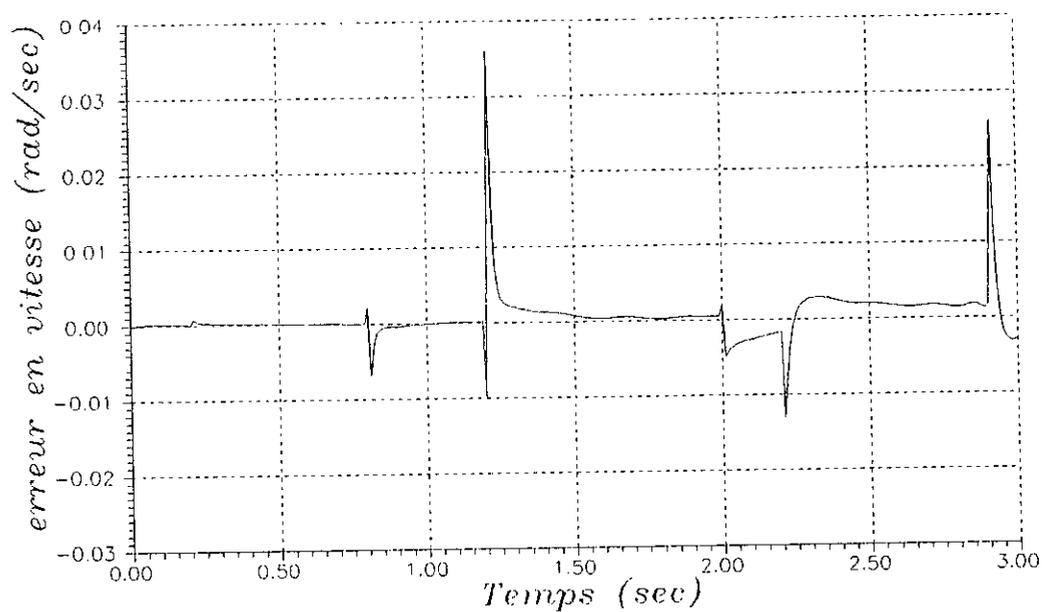


(b)

Figure IV.18. Commande Neurale Dynamique Anticipée Adaptative.  
 (a) : trajectoire de position (Liaison 2).  
 (b) : erreur de poursuite (Liaison 2).



(a)



(b)

Figure IV.19. Commande Neurale Dynamique Anticipée Adaptative.

(a) : trajectoire de vitesse (Liaison 2).

(b) : erreur de poursuite (Liaison 2).

### IV.3.2 Commande Neurale Adaptative par Calcul du Couple Généralisé

Dans ce paragraphe nous procédons à la validation, par simulation, de la deuxième variante de l'approche de commande montrée en figure III.4. Pour une telle approche de commande les couples fixe et variable sont obtenus par des réseaux de neurones. Le modèle dynamique décomposé en deux parties fixe et variable a été donné précédemment. Chaque partie à son tour est décomposée en matrice d'inertie  $M$  et vecteur  $H$  des termes de Coriolis et centrifuges comme suit :

Modèle dynamique fixe :

$$T_f = M_f(q)\ddot{q} + H_f(q, \dot{q})$$

$$H_f(q, \dot{q}) = C_f(q, \dot{q}) + G_f(q)$$

Modèle dynamique variable :

$$T_v = M_v(q)\ddot{q} + H_v(q, \dot{q})$$

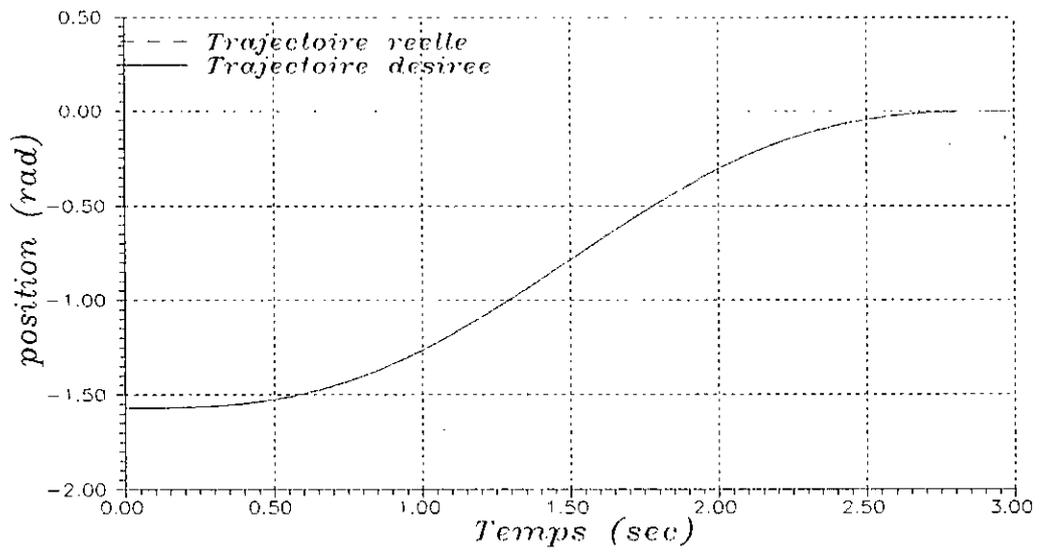
$$H_v(q, \dot{q}) = C_v(q, \dot{q}) + G_v(q)$$

Où  $M_f$  et  $M_v$  représentent les matrices d'inertie fixe et variable respectivement, et  $H_f$  et  $H_v$  représentent les vecteurs des termes de Coriolis et centrifuges fixe et variable respectivement. Les conditions de simulation demeurent les mêmes que celles de la première variante. Nous avons avec la même architecture des réseaux pour les matrices d'inerties fixe et variable abouti à un neurone en entrée laquelle correspond à la position de la deuxième articulation, 8 neurones dans la couche cachée et 4 neurones en sortie. Deux autres réseaux avec 4 neurones en couche d'entrée, 20 neurones dans la couche cachée et 2 neurones en sortie ont également été utilisés pour la détermination des vecteurs  $H_f$  et  $H_v$ . Concernant les réseaux de neurones de la partie variable nous utilisons la propriété de linéarité du modèle par rapport aux paramètres dynamiques, et ceci pour les mêmes raisons citées dans le paragraphe précédent.

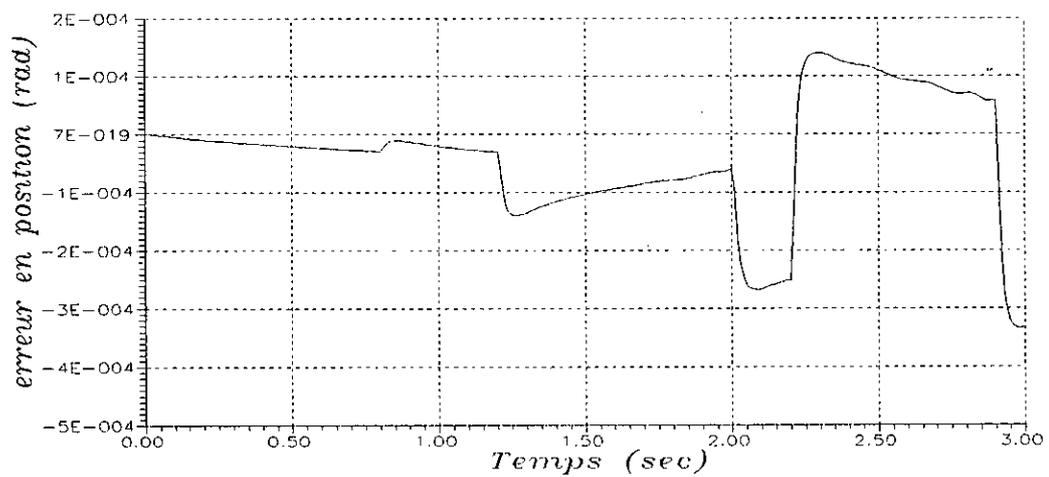
Il est à noter que contrairement à la première variante de l'approche de commande proposée, les entrées des réseaux de neurones pour les vecteurs  $H_f$  et  $H_v$  sont les trajectoires de position et de vitesse réelles et non désirées. D'un autre côté, cette variante n'a affecté en aucun cas les résultats de l'estimation de la masse de la charge. En effet nous

retrouvons pratiquement les mêmes résultats que ceux illustrés par la figure IV.15. Les paramètres d'adaptations du correcteur PD adaptatif ont été maintenus aux mêmes valeurs que celles de la précédente commande.

Les résultats de simulation donnés en figures IV.20 à IV.23 montrent que cette deuxième variante fournit également des performances satisfaisantes en poursuite de trajectoire de position et de vitesse. Généralement la technique du couple généralisé n'est robuste que vis à vis de faibles perturbations, alors que cette variante, qui repose sur le principe de cette technique, est robuste vis à vis de grandes perturbations. Ceci nous montre l'utilité de la partie variable du couple d'anticipation. Nous remarquons aussi que là où la masse de la charge est nulle (robot à vide) l'erreur est très faible ce qui montre l'efficacité du correcteur PD adaptatif vis à vis des frottements et des dynamiques non modélisées. D'un autre côté, aux instants de variation de la masse de la charge, l'amplitude de l'erreur en position et en vitesse est moins importante que celle avec la première variante et cela pour les deux liaisons. Cette erreur est rapidement réduite par le correcteur PD, ce qui nous amène à conclure que le correcteur PD est en mesure de compenser également les perturbations dues aux erreurs d'estimation.



(a)

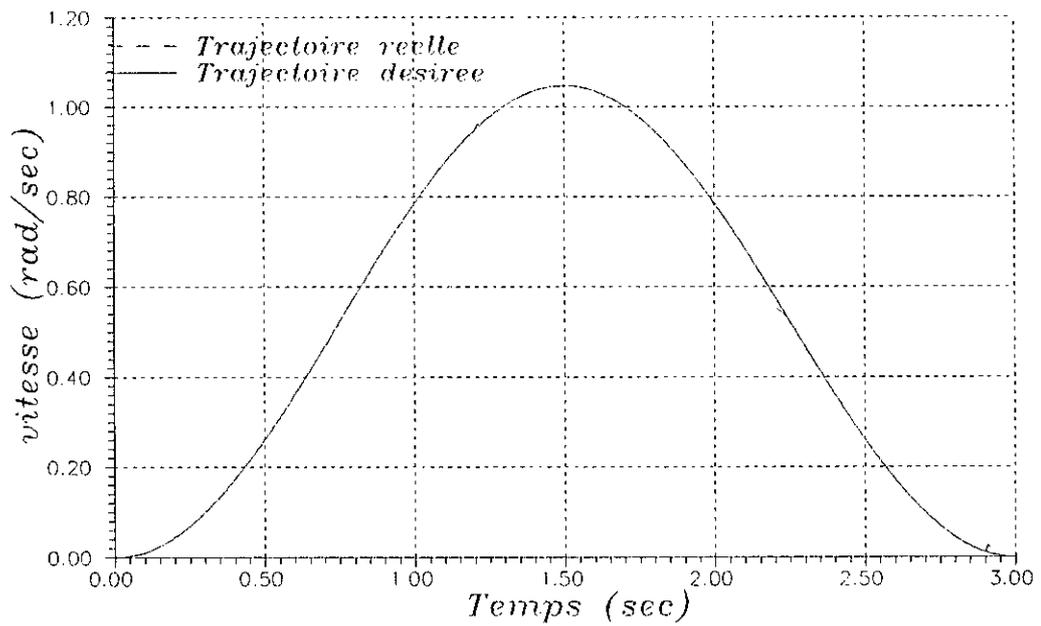


(b)

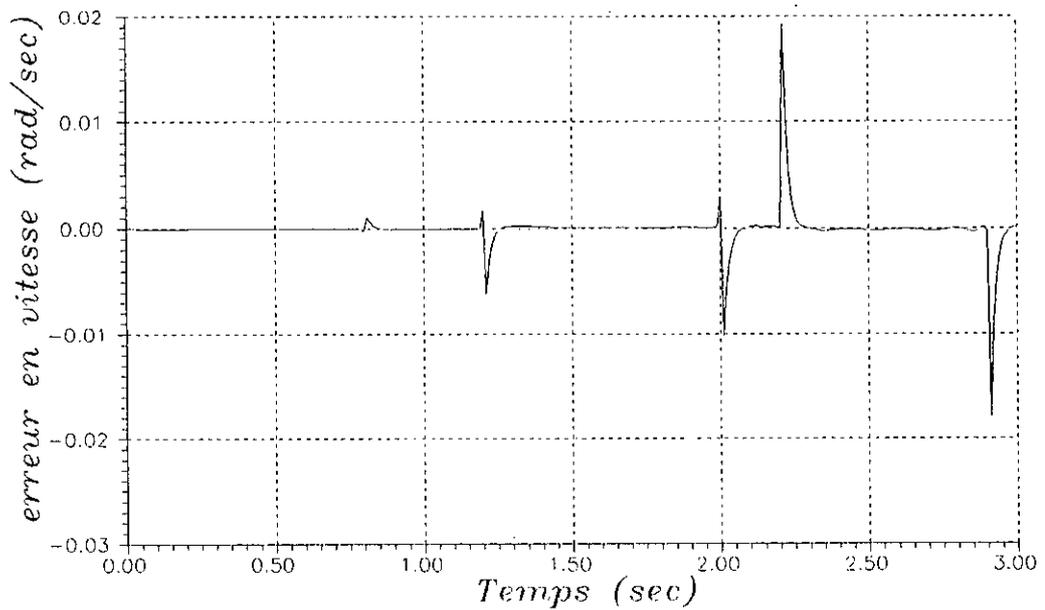
Figure IV.20. Commande Neurale par couple calculé Adaptative.

(a) : trajectoire de position (Liaison 1).

(b) : erreur de poursuite (Liaison 1).



(a)

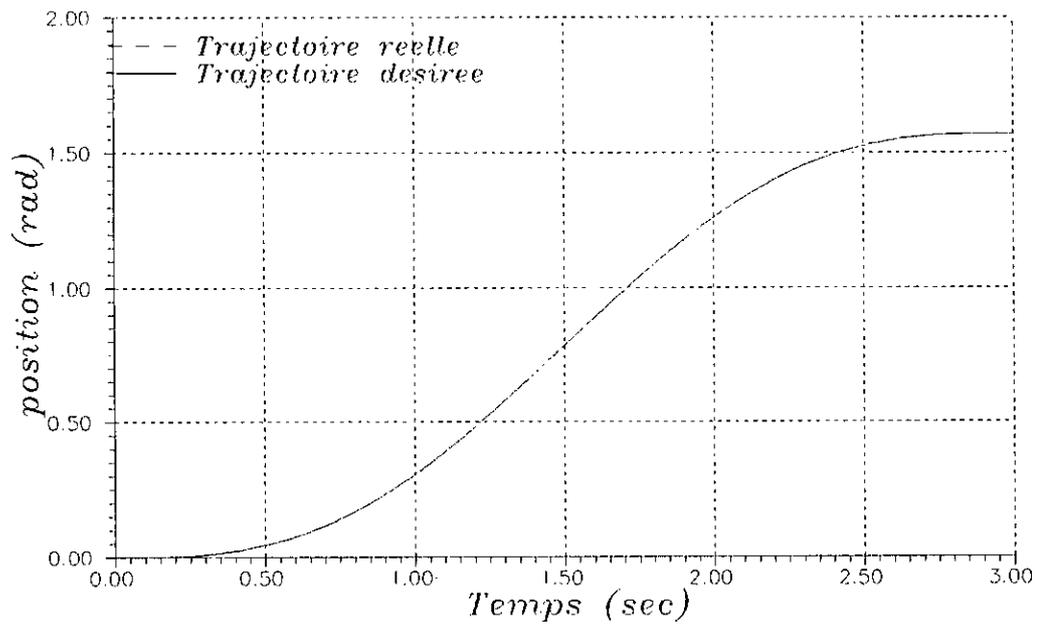


(b)

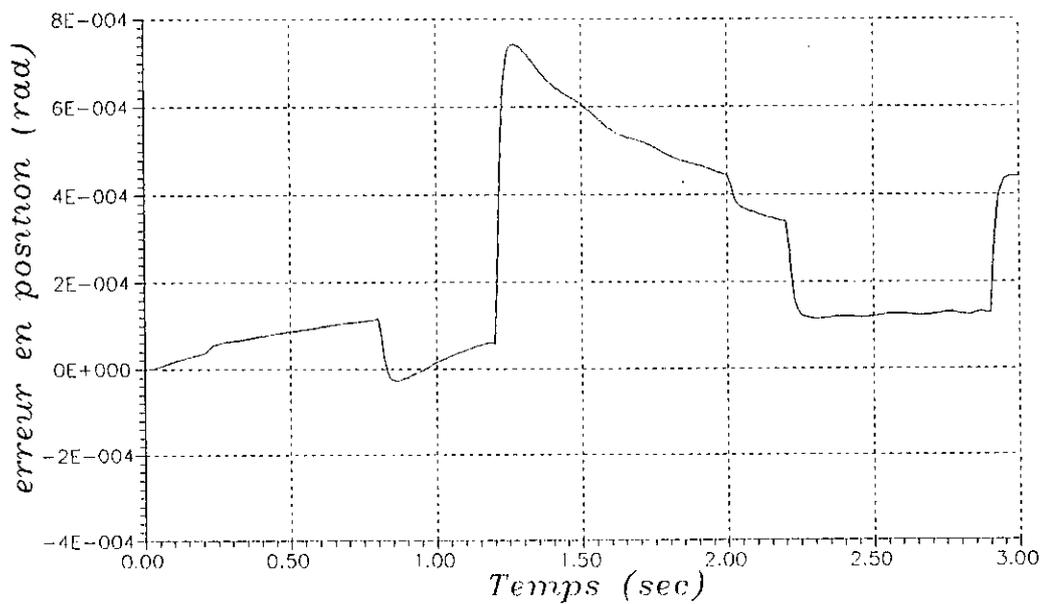
Figure IV.21. Commande Neurale par couple calculé Adaptative.

(a) : trajectoire de vitesse (Liaison 1).

(b) : erreur de poursuite (Liaison 1).



(a)

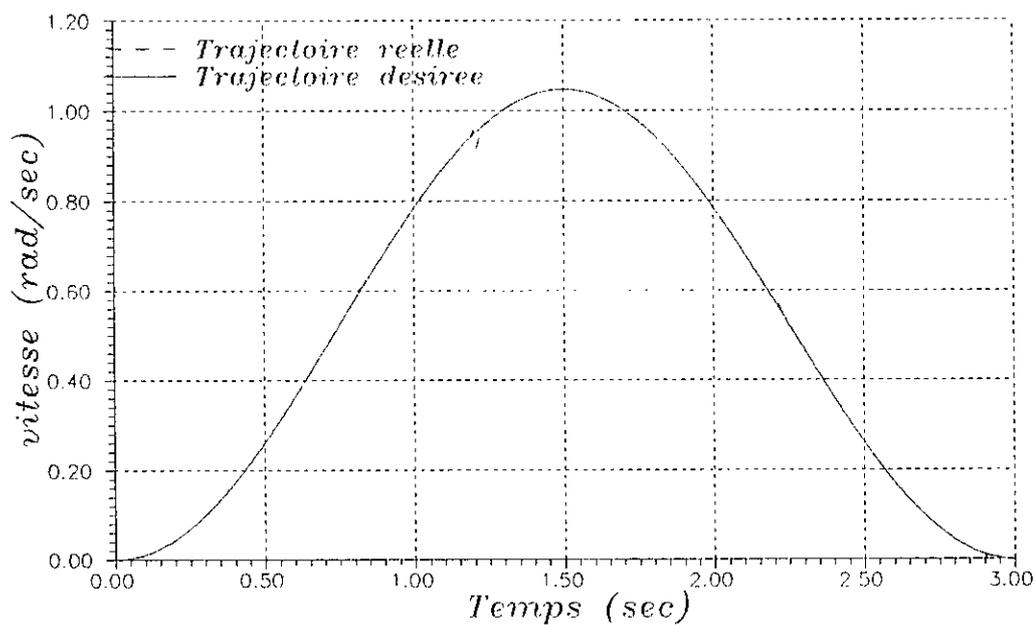


(b)

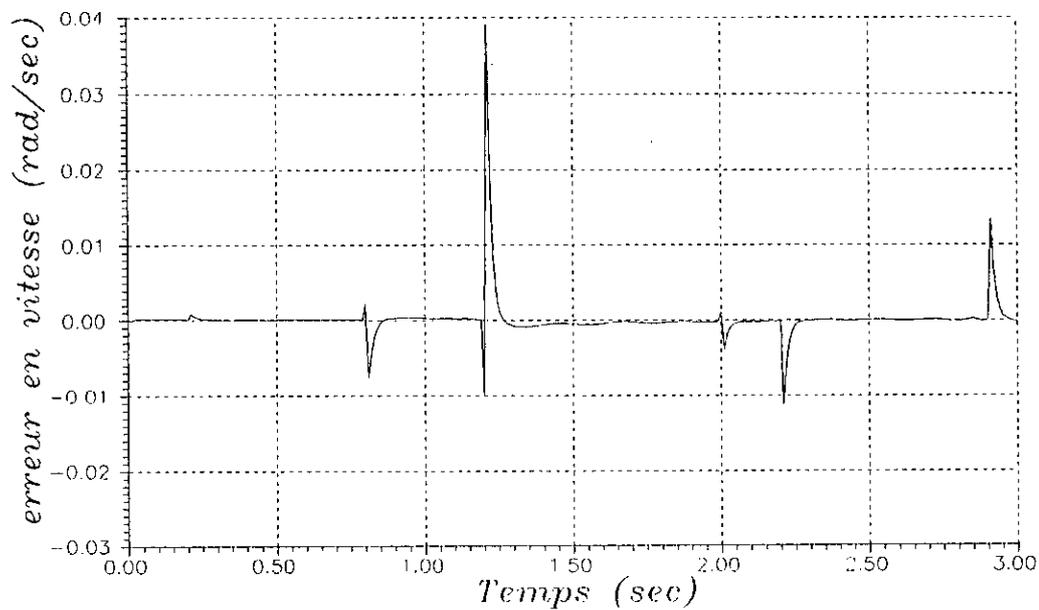
Figure IV.22. Commande Neurale par couple calculé Adaptive.

(a) : trajectoire de position (Liaison 2).

(b) : erreur de poursuite (Liaison 2).



(a)



(b)

Figure IV.23. Commande Neurale par couple calculé Adaptative.

(a) : trajectoire de vitesse (Liaison 2).

(b) : erreur de poursuite (Liaison 2).

## IV.4 Etude Comparative

pour évaluer les performances des deux méthodes de commande proposées, nous avons effectué une comparaison par simulation sur l'erreur de poursuite en position et en vitesse avec une commande neuro-dynamique proposée par Ozaki (Ozaki et al., 1991). L'objectif de cette comparaison est de montrer en premier lieu l'avantage que procure la décomposition du modèle dynamique et de l'introduction de l'estimateur neurale des paramètres inertiels variables.

La commande proposée par Ozaki est basée sur le principe de la méthode du couple généralisé, où le modèle dynamique est décomposé en matrice M et vecteur H selon la figure IV.24 (Ozaki et al., 1991).

La loi de commande est donnée par l'expression suivante :

$$T = \hat{M}(q)U + \hat{H}(q, \dot{q})$$

$$U = \ddot{q}_d + K_p(q_d - q) + K_v(\dot{q}_d - \dot{q})$$

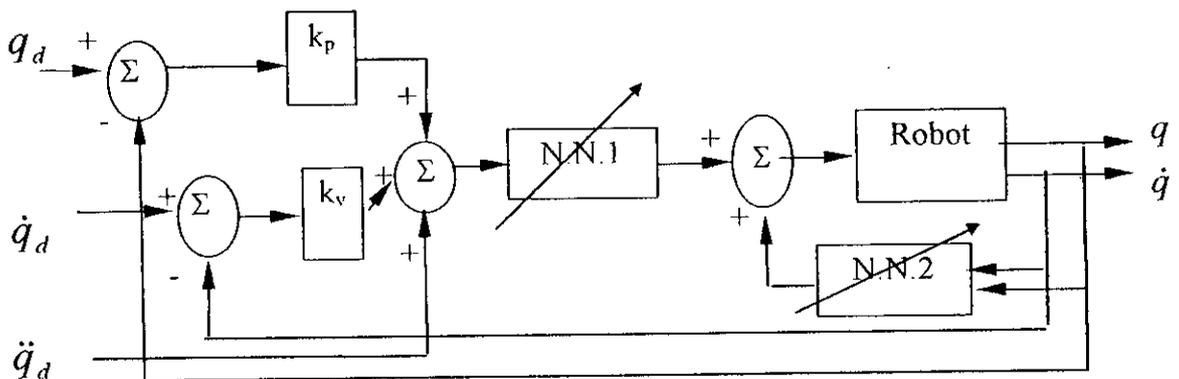


Figure IV.24. Commande Neuro-Dynamique.

Deux réseaux de neurones (NN1 et NN2) sont utilisés pour évaluer la matrice M et le vecteur H. Les gains  $K_p$  et  $K_v$  sont des matrices diagonales de dimension  $n \times n$  composées d'éléments positifs. L'architecture du réseau de neurones NN1 est choisie telle que la couche d'entrée possède un (1) neurone, 8 en couche cachée et 4 en couches de sortie. Le réseau NN2 est constitué de 4 neurones en entrée, 20 et 10 neurones dans la première et la seconde couche cachée respectivement et 2 neurones en sortie. A partir de cette architecture des réseaux nous pouvons constater que le fait de décomposer le modèle dynamique en parties fixe et variable (schéma II.4) a contribué à la réduction de la taille du réseau de neurones comparativement à la taille du réseau pour le schéma de commande de

la figure IV.24. Les gains du correcteur PD sont fixés à  $K_p=100$  et  $K_v=60$  pour les deux liaisons.

Les résultats de simulation donnés en figures IV.25 à IV.28 montrent parfaitement que les méthodes de commande proposées fournissent les meilleurs performances. Nous constatons que dans l'intervalle de temps [0 - 0.2 sec] où la masse de la charge est nulle (le robot à vide) les trois commandes présentent une robustesse proche vis à vis des perturbations causées par les frottements. Toutefois pour une faible variation de la masse de la charge le correcteur PD n'arrive plus à compenser correctement cette perturbation, alors que la technique du couple généralisé garde des performances satisfaisantes. Mais dès que la variation de la masse de la charge devient importante la méthode du couple généralisé perd de sa robustesse. Cet effet était attendu vu que cette technique n'est performante que pour de faibles perturbations. La table suivante est un moyen comparatif entre ces approches en considérant la somme des erreurs quadratiques, l'erreur moyenne et l'erreur maximale de poursuite en position et en vitesse. Cette table montre que ces erreurs sont minimales dans le cas des méthodes que nous avons élaborées.

		Liaison 1		Liaison 2	
		Position	vitesse	position	Vitesse
NDAA	$E_{qd}$	2.78E-4	0.006	2.45E-4	0.0035
	$E_{moy}$	7.07E-4	0.0025	6.80E-4	0.0015
	$E_{max}$	0.0019	0.0286	0.0022	0.0359
NCCA	$E_{qd}$	3.74E-6	0.0012	3.22E-5	0.0027
	$E_{moy}$	8.17E-5	4.68E-4	2.39E-4	7.22E-4
	$E_{max}$	3.33E-4	0.0192	7.43E-4	0.0390
MCC	$E_{qd}$	0.0675	0.2251	0.2348	1.0773
	$E_{moy}$	0.0079	0.0184	0.0193	0.0435
	$E_{max}$	0.0414	0.0698	0.0550	0.1420

Table comparative.

NDAA : commande Neurale Dynamique Anticipée Adaptative.

NCCA : commande Neurale par Couple Calculé Adaptative.

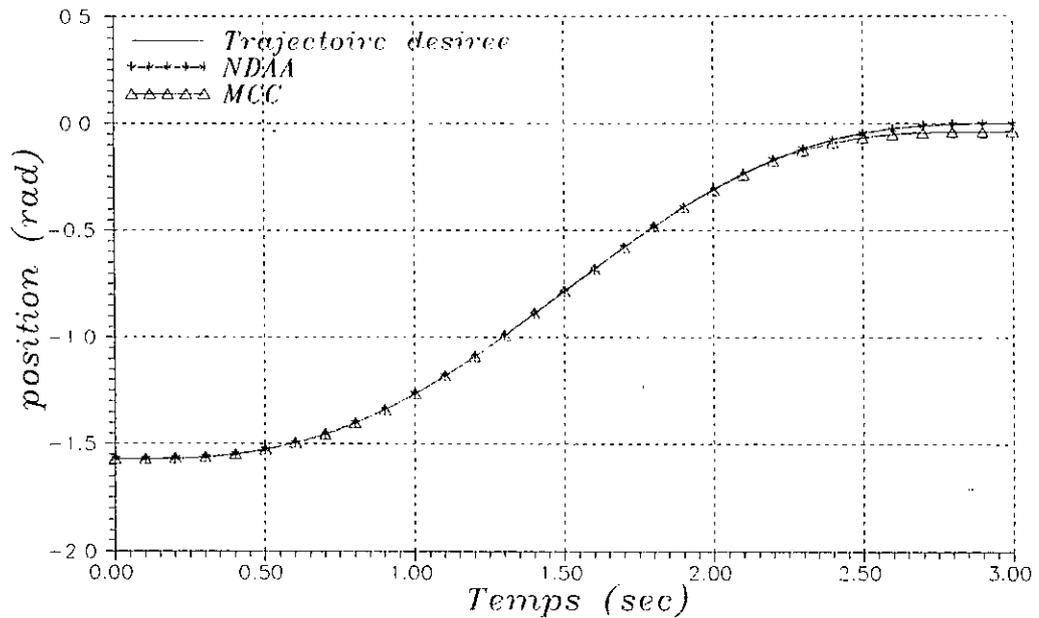
MCC : commande par Méthode du Couple Calculé.

$E_{qd}$  : somme des Erreurs Quadratiques.

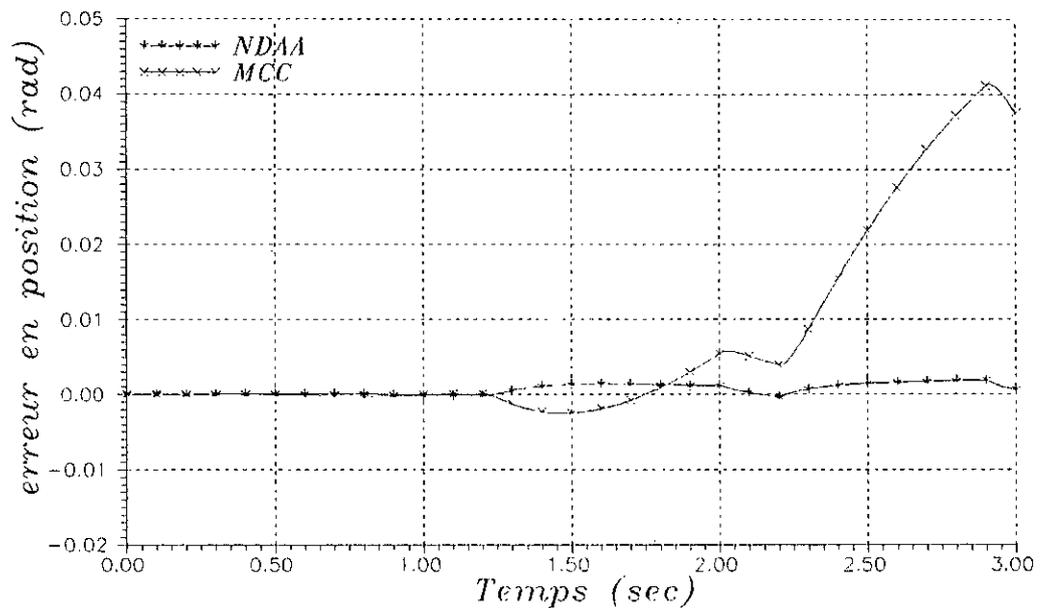
$E_{moy}$  : Erreur Moyenne.

$E_{max}$  : Erreur Maximale.

Cette comparaison nous a permis de prouver la supériorité de notre approche avec ses deux variantes, par rapport à l'approche d'Ozaki, en terme de robustesse vis à vis de la variation paramétrique et la présence de dynamiques non modélisées. Ce qui confirme notre démarche dans la décomposition du modèle dynamique et dans l'utilisation de l'estimateur neurale des paramètres inertiels variables.



(a)



(b)

Figure IV.25. Etude Comparative.  
 (a) : trajectoire de position (Liaison 2).  
 (b) : erreur de poursuite (Liaison 2).

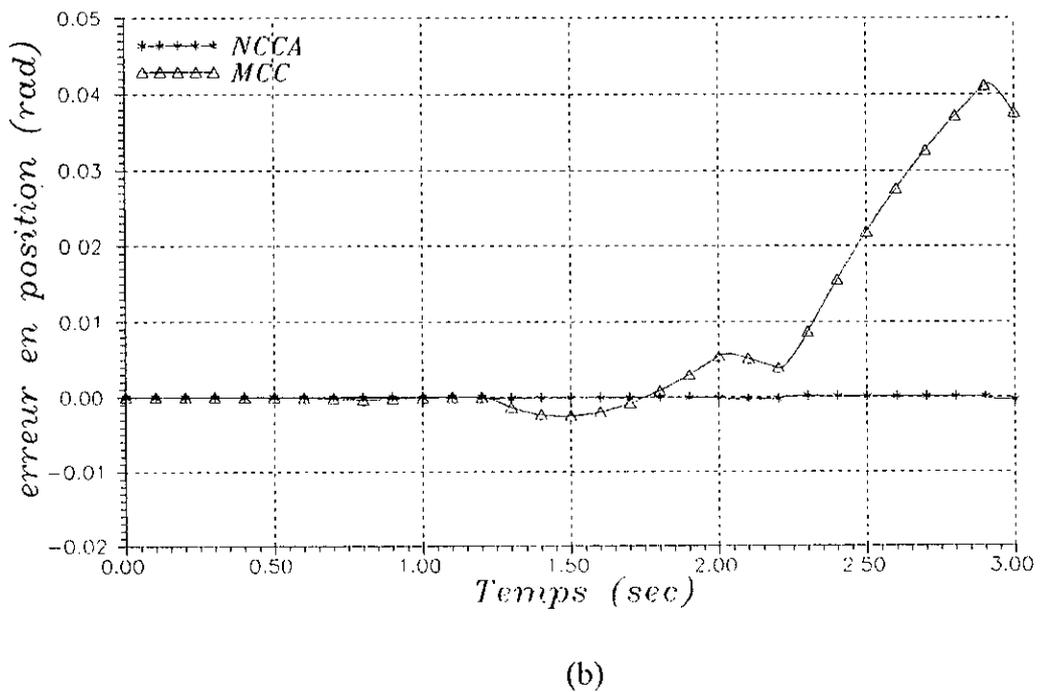
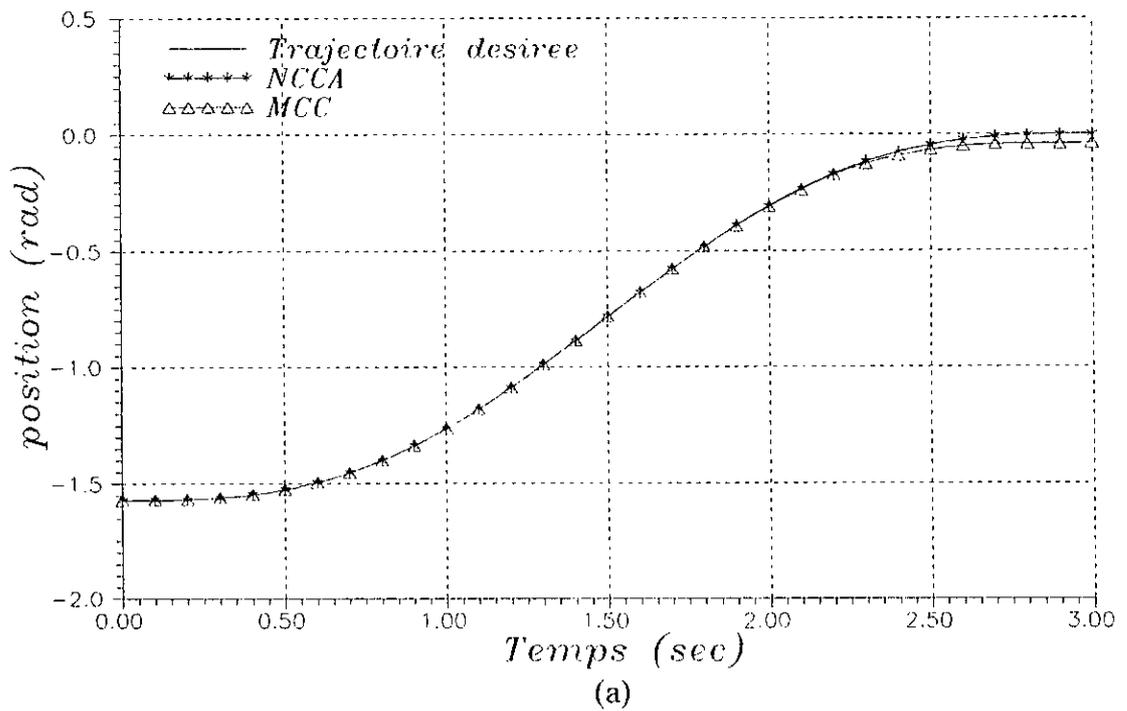
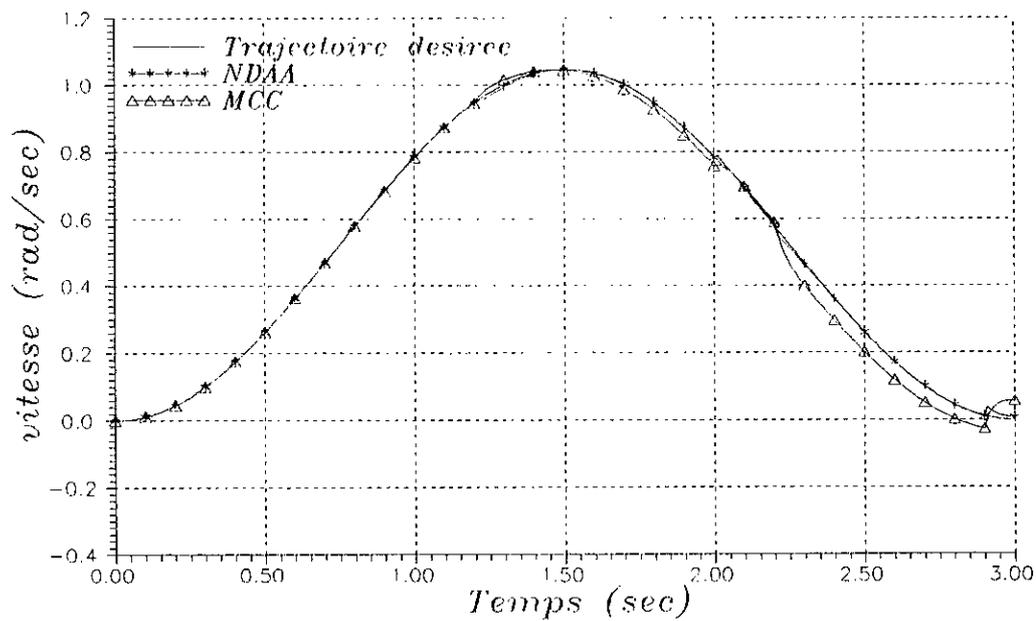
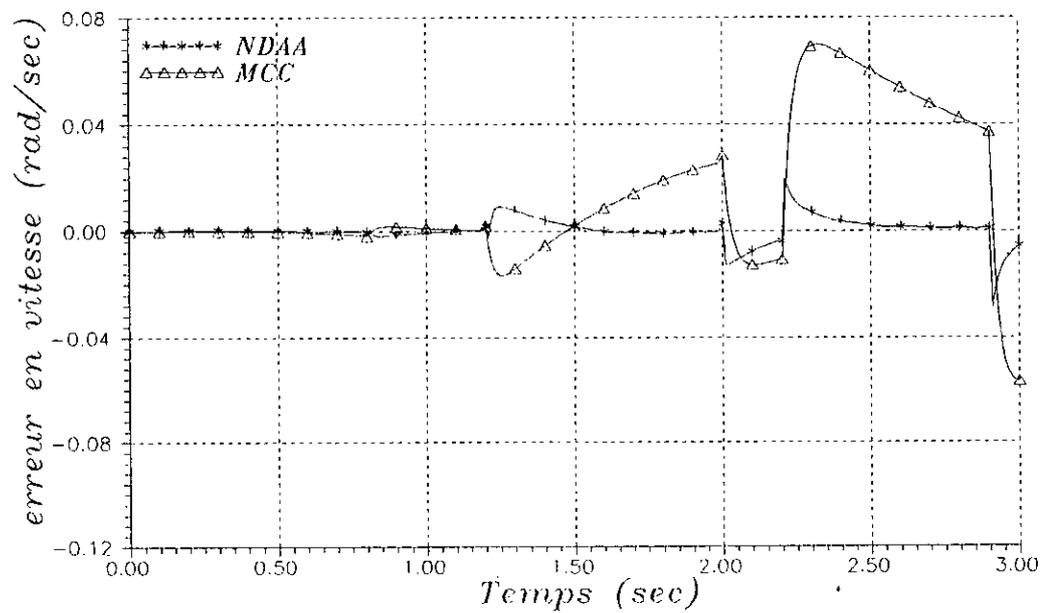


Figure IV.26. Etude Comparative.  
 (a) : trajectoire de position (Liaison 2).  
 (b) : erreur de poursuite (Liaison 2).

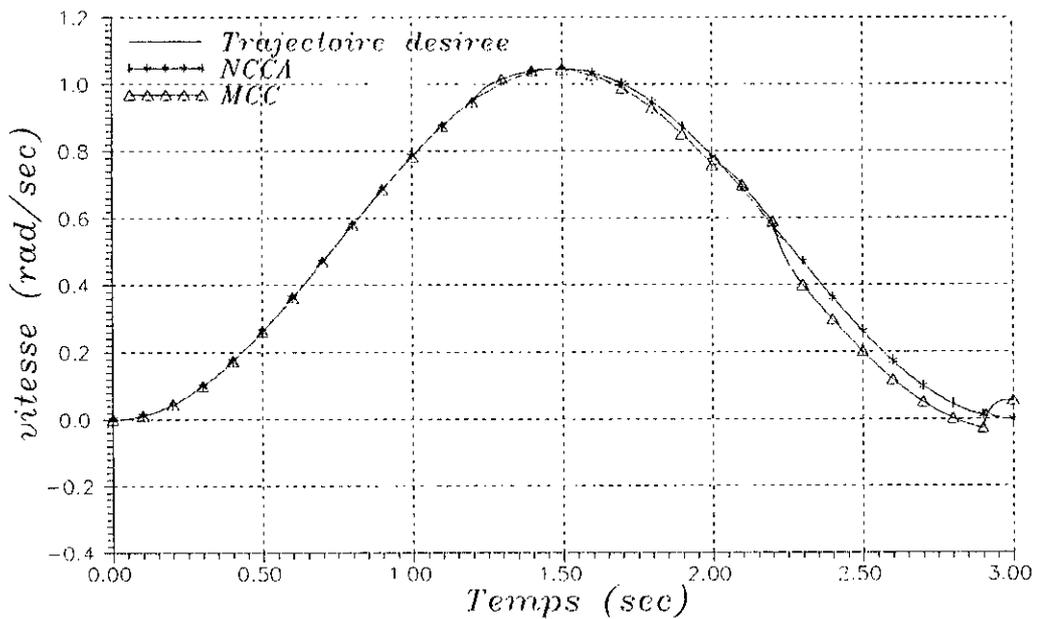


(a)

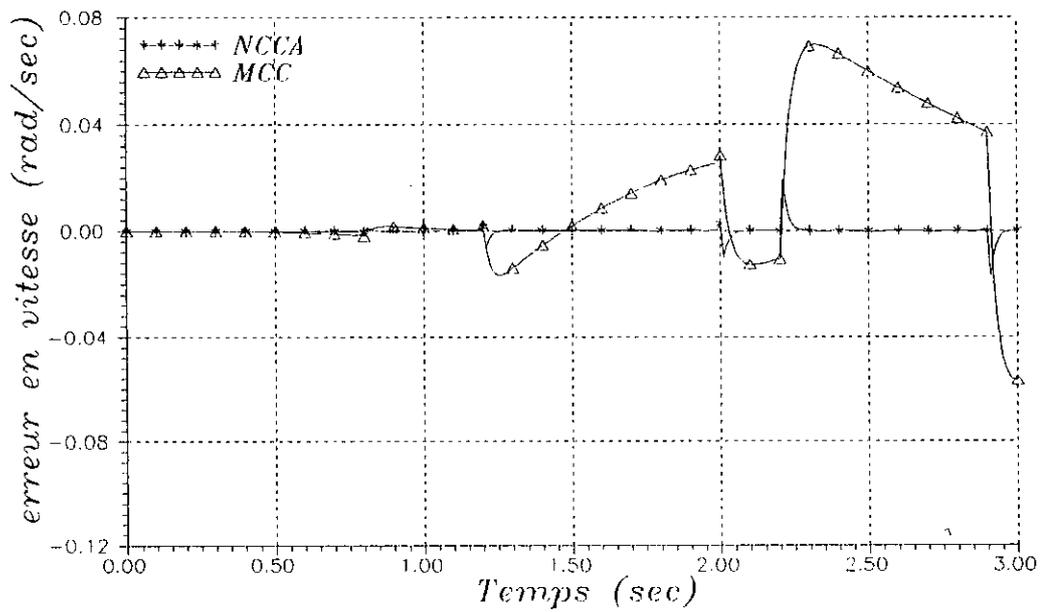


(b)

Figure IV.27. Etude Comparative.  
 (a) : trajectoire de position (Liaison 2).  
 (b) : erreur de poursuite (Liaison 2).



(a)



(b)

Figure IV.28. Etude Comparative.  
 (a) : trajectoire de position (Liaison 2).  
 (b) : erreur de poursuite (Liaison 2).

## ***CHAPITRE V***

# ***EXPERIMENTATION SUR LE ROBOT RP41 DU CDTA***

---

---

Dans ce dernier chapitre nous allons mettre en œuvre la méthode de commande sur un site réel. Il s'agit d'un robot à 3 liaisons rotoïdes (RP41 du CDTA).

## V.1 Description du Site du RP41

Le site expérimental, qui comprend la structure mécanique du robot RP41 ainsi que la structure de commande constituée du micro-ordinateur et de l'armoire de puissance, est montré en figure V.1

Le RP41 est un robot planaire. Sa structure mécanique est constituée de :

- Deux segments S1 et S2, pouvant tournant autour des axes A1 et A2 respectivement.
- Une pince à deux doigts tourne autour du troisième axe A3.

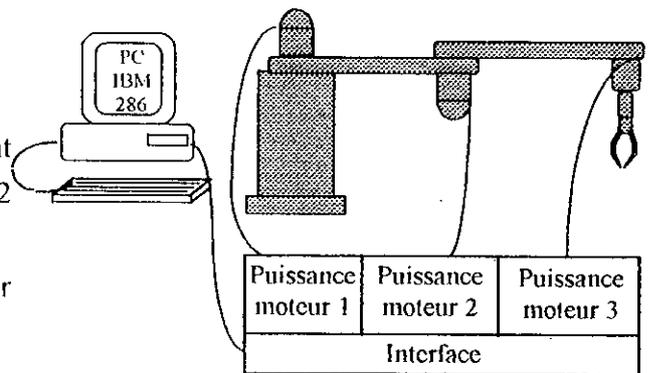


Figure V.1 Schématisation du site expérimental.

Les possibilités de mouvement du robot sont les trois rotations R1, R2 et R3, plus une translation T de la pince. Cette translation se résume à une position haute et à une position basse de la pince.

L'ensemble repose sur un socle rigide et réglable en hauteur à l'aide d'une crémaillère à vis de blocage.

Le RP41 possède un champ d'action assez important limité par des butées mécaniques qui indiquent les positions extrêmes de chaque liaison :

AXE 1  $52.65^\circ \leq q \leq 221.48^\circ$

AXE 2  $30.50^\circ \leq q \leq 258.30^\circ$

AXE 3  $0^\circ \leq q \leq 360.00^\circ$

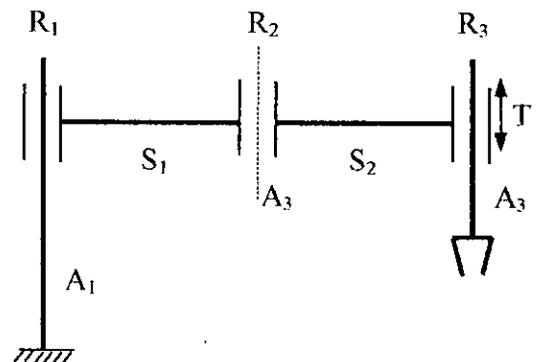


Figure V.2. Schéma de la structure mécanique du RP41.

Le rayon d'action de la pince est donné par :

Distance maximale de la pince à l'axe 1 : 50 cm.

Altitude maximale par rapport à la base du socle 6 cm.

Altitude minimale par rapport à la base du socle : 3 cm.

Le mouvement des trois liaisons rotatives est assuré par des actionneurs électriques. Il s'agit de motoréducteurs (moteur avec réducteur incorporé) à courant continu, et à deux sens de rotation. La vitesse maximale des deux premiers axes est de 4 *tr/mn* pour une tension de 24 volts, alors que celle de l'actionneur assurant la rotation de la pince est de 25 *tr/mn* pour la même tension d'excitation.

La pince du RP41 peut être actionnée par un actionneur pneumatique doté d'un système à vis écrou, et des vérins à air comprimé commandés par deux électrovannes qui fonctionnent en tout ou rien.

### Les capteurs

Le robot RP41 est doté de deux types de capteurs :

- Capteurs de position, chaque actionneur électrique est accouplé avec un capteur de position angulaire. Le capteur est constitué principalement d'un potentiomètre de précision et d'un convertisseur analogique/numérique d'une résolution de 10 bits.
- Capteurs de fin de course, associés à chaque vérin, sont des contacts qui se ferment sous l'action d'un champ magnétique créé par un aimant se trouvant à l'intérieur du vérin.

### Description de la structure de commande

La structure de commande est constituée par les éléments suivants :

- Un micro-ordinateur IBM PC doté d'une carte d'interface.
- Une armoire de puissance où sont installés les amplificateurs de puissance, les cartes des optocoupleurs pour l'isolation galvanique des signaux provenant des capteurs de position et enfin une carte à relais permettant de commuter une tension (24 V) vers l'une ou l'autre des deux entrées de chacun des électrovannes.

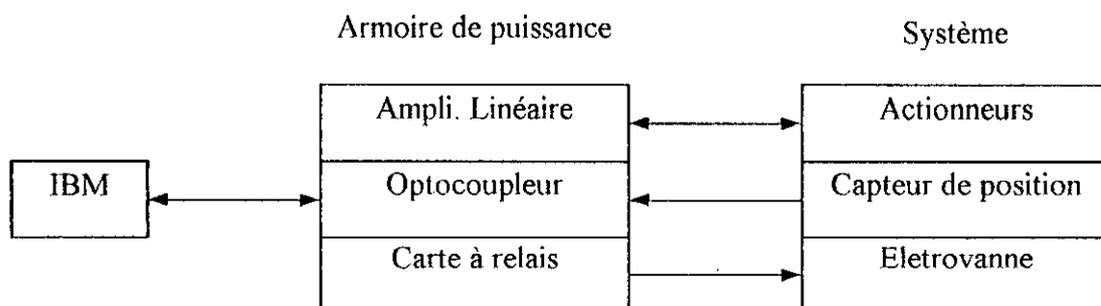


Figure V.3. Structure de commande du RP41.

### Carte d'interface

La carte d'interface permet la lecture par le micro-ordinateur à travers des ports d'entrées/sorties parallèles de trois capteurs de position et de quatre capteurs de fin de course. Elle permet aussi l'activation des trois moteurs à courant continu à travers des convertisseurs Numériques/Analogiques et des deux électrovannes à travers un port d'entrée/sortie parallèle.

Le modèle dynamique du robot manipulateur possède plusieurs caractéristiques. Parmi celles-ci la linéarité de ce modèle par rapport aux paramètres dynamiques. Elle sera utilisée

en vue de l'identification des paramètres dynamiques de ce robot. Pour cela nous avons besoin de connaître les couples exercés sur le système et les trajectoires de position, de vitesse et d'accélération. Dans le cas du RP41 nous n'avons pas accès aux couples exercés sur les liaisons. Cependant, nous pouvons connaître les consignes envoyées par l'ordinateur. Par conséquent, l'équation dynamique du robot manipulateur est combinée à l'équation dynamique régissant le fonctionnement de l'ensemble Amplificateur-Actionneur-Réducteur montré en figure II.2. Les actionneurs qui entraînent les différentes articulations sont des moteurs à courant continu commandés en tension.

## V.2 Modèle Dynamique Du RP41

Le modèle dynamique du robot RP41 (voir en annexe B les détails d'obtention du modèle) est donné par les équations suivantes

$$\begin{aligned} \tau_1 = & [I_1 + I_2 + I_3 + m_1 l_1^2 + m_2 (L_1^2 + l_2^2) + m_3 (L_1^2 + L_2^2) + 2L_1 (m_2 l_2 + m_3 L_2) \cos(q_2)] \ddot{q}_1 + \\ & [I_2 + I_3 + m_2 l_2^2 + m_3 L_2^2 + L_1 (m_2 l_2 + m_3 L_2) \cos(q_2)] \ddot{q}_2 + I_3 \ddot{q}_3 - \\ & L_1 (m_2 l_2 + m_3 L_2) \sin(q_2) \dot{q}_2 - 2L_1 (m_2 l_2 + m_3 L_2) \sin(q_2) \dot{q}_1 \dot{q}_2. \end{aligned} \quad (V.1a)$$

$$\begin{aligned} \tau_2 = & [I_2 + I_3 + m_2 l_2^2 + m_3 L_2^2 + L_1 (m_2 l_2 + m_3 L_2) \cos(q_2)] \ddot{q}_1 + [I_2 + I_3 + m_2 l_2^2 + m_3 L_2^2] \ddot{q}_2 + \\ & I_3 \ddot{q}_3 + [L_1 (m_2 l_2^2 + m_3 L_2^2) \sin(q_2) \dot{q}_1^2] \end{aligned} \quad (V.1b)$$

$$\tau_3 = I_3 (\ddot{q}_1 + \ddot{q}_2 + \ddot{q}_3) \quad (V.1c)$$

Le modèle dynamique complet du robot RP41, y compris la dynamique des actionneurs et celle des frottements (voir chapitre II), est donné par :

$$\begin{aligned} u_1 = & \frac{1}{K_1} \left( [I_1 + I_2 + I_3 + m_1 l_1^2 + m_2 (L_1^2 + l_2^2) + m_3 (L_1^2 + L_2^2) + 2L_1 (m_2 l_2 + m_3 L_2) \cos(q_2) + n_1^2 J_{m1}] \ddot{q}_1 \right) + \\ & \frac{1}{K_1} \left( [I_2 + I_3 + m_2 l_2^2 + m_3 L_2^2 + L_1 (m_2 l_2 + m_3 L_2) \cos(q_2)] \ddot{q}_2 + (I_3 \ddot{q}_3) \right) - \\ & \frac{1}{K_1} \left( L_1 (m_2 l_2 + m_3 L_2) \sin(q_2) \dot{q}_2 + 2L_1 (m_2 l_2 + m_3 L_2) \sin(q_2) \dot{q}_1 \dot{q}_2 \right) + \tau_{f1} \end{aligned} \quad (V.2a)$$

$$\begin{aligned}
u_2 = & \frac{1}{K_2} \left( [I_2 + I_3 + m_2 l^2_2 + m_3 L^2_2 + L_1 (m_2 l_2 + m_3 L^2) \cos(q_2)] \ddot{q}_1 \right) + \\
& \frac{1}{K_2} \left( [I_2 + I_3 + m_2 l^2_2 + m_3 L^2_2 + n_2^2 J_{m2}] \ddot{q}_2 + I_3 \ddot{q}_3 \right) + \\
& \frac{1}{K_2} \left( L_1 (m_2 l^2_2 + m_3 L^2_2) \sin(q_2) \dot{q}_1^2 \right) + \tau_{f2}
\end{aligned} \tag{V.2b}$$

$$u_3 = \frac{1}{K_3} \left( I_3 (\ddot{q}_1 + \ddot{q}_2) + (I_3 + n_3^2 J_{m3}) \ddot{q}_3 \right) + \tau_{f3} \tag{V.2c}$$

avec :

$$K_1 = n_1 K_{a1} K_{c1}$$

$$K_2 = n_2 K_{a2} K_{c2}$$

$$K_3 = n_3 K_{a3} K_{c3}$$

$$\tau_{fi} = f_{si} \operatorname{sgn}(\dot{q}_i) + f_{vi} \dot{q}_i$$

Nous constatons que le modèle dynamique de toute la chaîne est aussi linéaire par rapport aux paramètres. Le vecteur P contient les paramètres dynamiques du modèle.

Par conséquent le modèle linéaire par rapport aux paramètres dynamiques est donné par :

$$\begin{aligned}
u_1 = & [P_1 + P_2 \cos(q_2)] \ddot{q}_1 + [P_3 + 0.5 P_2 \cos(q_2)] \ddot{q}_2 + P_4 \ddot{q}_3 - (0.5 P_2 \sin(q_2) \dot{q}_2 + P_2 \sin(q_2) \dot{q}_1 \dot{q}_2) + \\
& P_{11} \dot{q}_1 + P_{12} \operatorname{sign}(\dot{q}_1)
\end{aligned} \tag{V.3a}$$

$$u_2 = [P_5 + P_6 \cos(q_2)] \ddot{q}_1 + P_7 \ddot{q}_2 + P_8 \ddot{q}_3 + P_6 \sin(q_2) \dot{q}_1^2 + P_{13} \dot{q}_2 + P_{14} \operatorname{sign}(\dot{q}_2) \tag{V.3b}$$

$$u_3 = P_9 (\ddot{q}_1 + \ddot{q}_2) + P_{10} \ddot{q}_3 + P_{15} \dot{q}_3 + P_{16} \operatorname{sign}(\dot{q}_3) \tag{V.3c}$$

où les paramètres P donnés comme suit :

$$P_1 = [I_1 + I_2 + I_3 + m_1 l_1^2 + m_2(L_1^2 + l_2^2) + m_3(L_1^2 + L_2^2) + n_1^2 J_{m1}] / K_1$$

$$P_2 = 2L_1(m_2 l_2 + m_3 L_2) / K_1$$

$$P_3 = [I_2 + I_3 + m_2 l_2^2 + m_3 l_2^2] / K_1$$

$$P_4 = I_3 / K_1$$

$$P_5 = [I_2 + I_3 + m_2 l_2^2 + m_3 l_2^2] / K_2$$

$$P_6 = L_1(m_2 l_2 + m_3 L_2) / K_2$$

$$P_7 = [I_2 + I_3 + m_2 l_2^2 + m_3 l_2^2 + n_2^2 J_{m2}] / K_2$$

$$P_8 = I_3 / K_2$$

$$P_9 = I_3 / K_3$$

$$P_{10} = [I_3 + n_2^2 J_{m2}] / K_3$$

$$P_{11} = f_{v1} N_1^2 / K_1, \quad P_{12} = f_{s1} N_1^2 / K_1$$

$$P_{13} = f_{v2} N_2^2 / K_2, \quad P_{14} = f_{s2} N_2^2 / K_2$$

$$P_{15} = f_{v3} N_3^2 / K_3, \quad P_{16} = f_{s1} N_3^2 / K_3$$

Par conséquent le modèle peut être écrit dans une forme linéaire  $U = WP$

où  $U$  est le vecteur contenant les trois entrées de commande des liaisons. La matrice d'information  $W$  est donnée comme suit :

$$W = [W_1 : W_2 : W_3]$$

avec :

$$W_1 = \begin{bmatrix} \ddot{q}_1 & \cos(q_2)(\ddot{q}_1 + 0.5\ddot{q}_2) - \sin(q_2)\dot{q}_2(0.5\dot{q}_2 + \dot{q}_1) & \ddot{q}_2 & \ddot{q}_3 & 0 \\ 0 & 0 & 0 & 0 & \ddot{q}_1 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}$$

$$W_2 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 \\ \cos(q_2)\ddot{q}_1 + \sin(q_2)\dot{q}_1^2 & \ddot{q}_2 & \ddot{q}_3 & 0 & 0 \\ 0 & 0 & 0 & \dot{q}_1 + \dot{q}_2 & \ddot{q}_3 \end{bmatrix}$$

$$W_3 = \begin{bmatrix} \dot{q}_1 & \text{sign}(\dot{q}_1) & 0 & 0 & 0 & 0 \\ 0 & 0 & \dot{q}_2 & \text{sign}(\dot{q}_2) & 0 & 0 \\ 0 & 0 & 0 & 0 & \dot{q}_{31} & \text{sign}(\dot{q}_3) \end{bmatrix}$$

### V.3 Identification Du Robot RP41

La majorité des schémas de commande des robots manipulateurs, formulées dans la littérature récente, requièrent un modèle dynamique des robots manipulateurs. Ainsi les performances en termes de précision, de stabilité, et de robustesse de tels schémas dépendent de l'exactitude des paramètres du modèle dynamique. D'où le recours à la modélisation de la dynamique du robot et à l'identification des paramètres de ce modèle.

En général, il existe trois méthodes par le biais desquelles il est possible de déterminer les paramètres du robot manipulateur (Lu et al., 1993) :

1. En utilisant la conception assistée par ordinateur appliquée sur la géométrie du manipulateur,
2. En mesurant physiquement les paramètres de chaque corps,
3. En utilisant les données expérimentales obtenues des mouvements du robot.

La première méthode est sujette à des erreurs de modélisation attribuées aux petites parties constituant la structure mécanique du robot telles que les boulons, les rondelles, les écrous, etc. La deuxième méthode consiste à désassembler les liaisons du robot (cas du robot PUMA 560 par Armstrong) puis de déterminer les paramètres dynamiques par des expériences physiques (Armstrong et al, 1986). L'inconvénient de cette dernière méthode est que les valeurs des inerties de couplage ne peuvent pas être déterminées. La troisième méthode a été trouvée la meilleure des trois méthodes en termes de facilité expérimentale et de précision des valeurs obtenues.

#### V.3.1 Identifiabilité des Paramètres Dynamiques

Les paramètres dynamiques qui décrivent le modèle du robot incluent les paramètres inertiels, de masse, et de frottement. Chaque liaison est décrite par six paramètres inertiels, quatre paramètres de masse et un à quatre paramètres des frottements selon la complexité de la modélisation adoptée (Armstrong, 1988). Les paramètres dynamiques sont dits identifiables lorsqu'une solution optimale des paramètres estimés est unique. Cependant, ce n'est pas tous les paramètres qui peuvent être identifiés. Selon la structure cinématique du robot, quelques paramètres peuvent jouer un rôle majeur dans le modèle dynamique, d'autres ont un moindre impact, alors que d'autres peuvent n'avoir aucun effet (Lu et al., 1993). Dans ce sens, Khosla a classé les paramètres dynamiques en trois catégories (Khosla, 1989) :

1. Uniquement identifiable;
2. Identifiable en une combinaison linéaire;
3. Non identifiable.

### V.3.2 Modèles d'identification

Dans la littérature (Lu et al., 1993 ; Prüfer et al., 1994) nous pouvons trouver deux modèles d'identification : le modèle dynamique (différentiel) ou le modèle énergétique (intégral).

Le modèle différentiel, utilisé dans la commande du robot, relie les couples imposés aux liaisons aux mouvements (position, vitesse, et accélération) des liaisons :

$$U = w(q, \dot{q}, \ddot{q})P$$

Le modèle intégral est basé sur le théorème de l'énergie, stipule que l'énergie mécanique totale appliquée au robot est égale à l'énergie potentielle et cinétique contenue dans le système. Cette énergie est entièrement déterminée par les positions et vitesses des liaisons (Prüfer, 1994).

$$y = \int U \dot{q} dt = w_i P$$

Il est à noter que dans les deux modèles les mêmes paramètres apparaissent avec la même interprétation physique. Comme pour le modèle différentiel, le modèle intégral est linéaire par rapport aux paramètres.

La différence des deux modèles est dans l'exigence des termes d'accélération pour le modèle différentiel. Ceci peut constituer un problème pratique vu que la valeur de l'accélération peut être obtenue de deux manières :

1. En mesurant l'accélération à l'aide d'un accéléromètre,
2. En dérivant la position une deuxième fois.

Il est à noter qu'une installation d'un accéléromètre cause un encombrement sur le bras et une difficulté de conception. De plus, il provoque des vibrations indésirables ce qui aboutit à des données bruitées. D'un autre côté, dériver la position deux fois amplifie le bruit, lequel provoque la présence de bruit de haute fréquence due à la dérivation.

La présence de ces bruits présente certaines répercussions sur l'identification:

1. Le paramètre estimé est perturbé et la vitesse de convergence dépend du rapport signal sur bruit.
2. Le paramètre estimé final peut avoir un biais.

Pour éviter ce problème les spécialistes du domaine font recours au modèle intégral, lequel n'exige pas la connaissance de l'accélération.

Cependant, d'autres chercheurs préfèrent le modèle différentiel car c'est le modèle le plus utilisé en commande. D'un autre côté le modèle différentiel possède quelques avantages par rapport au modèle intégral (Prüfer, 1994) :

1. Le modèle différentiel contient autant d'équations que le nombre d'liaison du robot. Par contre le modèle intégral ne constitue qu'une seule équation. La décomposition du modèle différentiel garantit l'indépendance linéaire des colonnes de la matrice d'information.
2. L'identification par le moyen d'un modèle différentiel minimise l'équation d'erreur des couples et non pas l'équation d'erreur d'énergie comme pour le modèle intégral. Un minimum de l'équation d'erreur avec le modèle intégral ne garantit pas que les couples seront correctement évalués en utilisant les paramètres identifiés.
3. Il paraît que l'intégration, par laquelle le modèle intégral est dérivé, amplifie les signaux de haute fréquence. Ce phénomène augmente la sensibilité au bruit. Ainsi il peut être noté que l'élimination du besoin de l'accélération (bruité) est compensée par l'augmentation de la sensibilité aux erreurs de bruit.

Vu toutes ces considérations, et le fait que dans notre cas nous ne disposons pas d'un tachymètre et le codeur de position n'étant pas de grande précision, la vitesse obtenue par dérivation sera bruitée par un bruit de haute fréquence. Nous avons opté pour le modèle différentiel en utilisant un filtrage passe bas afin d'éliminer le bruit dû à la dérivation.

### V.3.3 Identifications En Ligne et Hors Ligne

Dans la littérature nous pouvons rencontrer deux manières de procéder à l'identification des paramètres dynamiques du robot : identification en ligne ou identification hors ligne. Dans la première procédure nous procédons à chaque itération à l'identification et la commande du robot. Arrivé à une certaine précision et après convergence des paramètres la procédure d'identification est arrêtée. L'utilisation de cette procédure est justifiée par le besoin de poursuivre les paramètres dynamiques variables. Cependant cette procédure peut rencontrer des problèmes d'identifiabilité des paramètres. Par contre l'approche hors ligne, la plus utilisée, a le mérite de pouvoir surmonter ce problème (Prüfer, 1996). D'un autre côté, le but de notre identification n'est pas de traquer les paramètres variables mais d'identifier les paramètres dynamiques considérés comme constants. Vu cette constance des paramètres dans la dynamique du robot, et la contrainte infligée par ce site expérimental où l'ordinateur qui gère le robot ne nous permet pas une identification en ligne, nous avons choisi une identification hors ligne.

### V.3.4 Choix de l'Excitation

Un problème important pour l'estimation des paramètres dynamiques est la recherche des couples appliqués (des entrées), qui excitent suffisamment la dynamique du robot et

favorisent le processus d'estimation en termes de robustesse vis à vis des perturbations (Pham, 1992). Dans ce cadre Armstrong a montré comment le type des trajectoires de mouvement influence la précision de l'identification (Armstrong, 1987). Ainsi les trajectoires des vitesses et des accélérations doivent être régulières et non nulles (Kozłowski et al., 1992). D'un autre côté, la précision des paramètres identifiés dépend du conditionnement de la matrice d'information. Ce conditionnement peut être utilisé comme une mesure de l'excitation de la dynamique du robot le long des trajectoires (Lu et al., 1993).

### V.3.5 Procédure d'Identification

Pour l'étape d'identification nous utilisons la propriété de la linéarité du modèle dynamique par rapport aux paramètres. L'équation dynamique linéaire par rapport aux paramètres est donnée par :

$$U = w(q, \dot{q}, \ddot{q})P \quad (V.4)$$

Il faut rappeler que chaque élément  $P_i$  du vecteur des paramètres  $P$  peut être soit un paramètre indépendant ou une combinaison linéaire de plusieurs paramètres. Disposant du modèle d'identification où les données dépendent de façon linéaire des paramètres dynamiques recherchés, nous proposons d'utiliser la méthode globale de régression par moindres-carrés qui est d'une mise en oeuvre facile. Par la méthode d'approximation des moindres-carrés, le vecteur des paramètres inconnus  $P$  peut être déterminé comme suit :

$$P = W^{-1}U \quad (V.5)$$

où  $W^{-1}$  est la matrice inverse de  $W$ , qui en général, n'est pas une matrice carrée d'où l'utilisation de l'algorithme des moindres carrés.

### V.3.6 Résultats Expérimentaux et Discussion

Dans cette section nous présentons les résultats obtenus sur l'identification des paramètres dynamiques du robot RP41 du CDTA, en utilisant l'approche théorique mentionnée précédemment.

Dans l'identification expérimentale nous avons besoin des trajectoires de position, de vitesse, d'accélération et le couple exercé sur chaque liaison. Comme le robot ne possède que le codeur de position nous avons recours à la dérivation afin d'obtenir la vitesse et l'accélération. Dans une première expérience nous avons recueilli les données en boucle fermée en utilisant une commande proportionnelle et dérivée. Malheureusement nous avons constaté que le couple était trop bruité pour être utilisé. Afin d'éviter les bruits de la boucle de retour dus à la dérivation nous avons décidé d'éliminer la boucle de retour et de

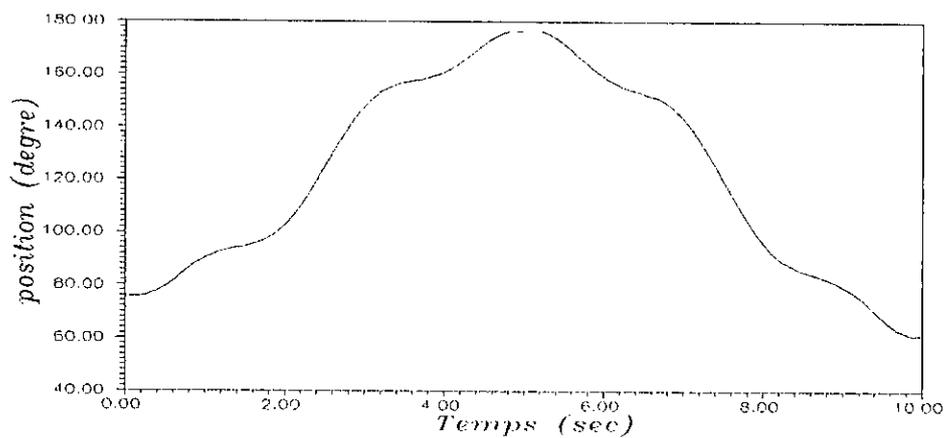
procéder à une identification en boucle ouverte. Pour cela il nous fallait trouver l'excitation convenable qui nous permet de bien conditionner la matrice d'information. Rappelons que les solutions standards de l'automatique consistent à utiliser comme entrées des séquences aléatoires de formes carrées ou en triangle. Ces séquences étant destinées à exciter toutes les fréquences de la structure. Cependant, en pratique les discontinuités inhérentes à ces excitations ne sont pas acceptables et risquent de déstabiliser le système (Pham, 1992). D'autre part, dans notre application, nous souhaitons identifier un modèle dont le comportement est par nature défini à de basses fréquences de fonctionnement. Par conséquent, nous avons considéré pour chaque liaison des couples de forme sinusoïdale :

$$\tau = a_1 \sin(2\pi f_1 t) + a_2 \sin(2\pi f_2 t)$$

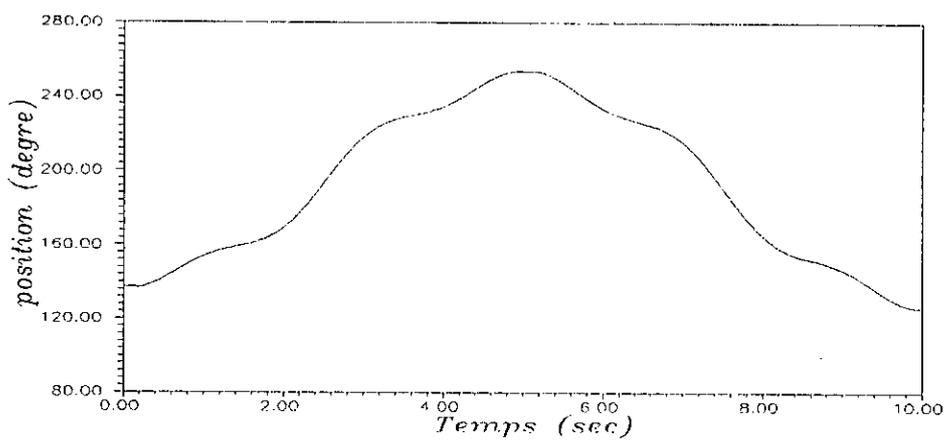
où  $a_1$ ,  $a_2$ ,  $f_1$ ,  $f_2$ , pour chaque liaison, sont donnés par le tableau suivant :

liaison	$a_1$	$a_2$	$f_1$	$f_2$
1	1200	500	0.1	0.5
2	1200	500	0.1	0.5
3	300	200	0.1	0.5

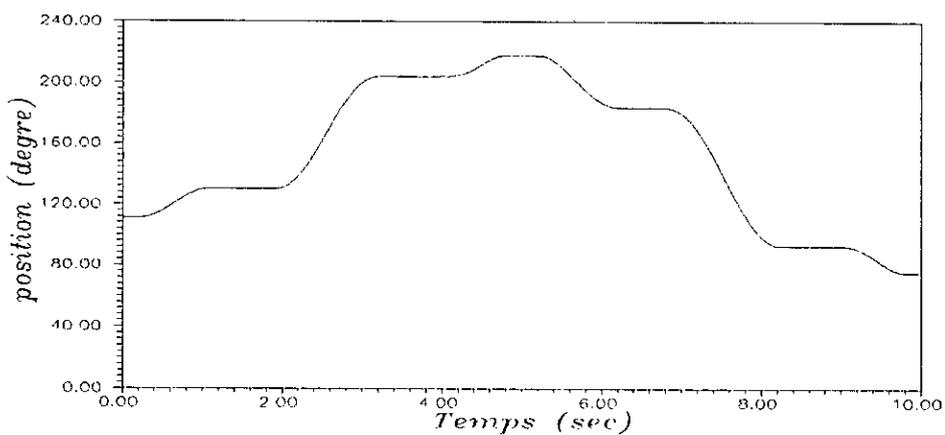
Dans notre expérience nous avons le couple exercé à chaque liaison et les mesures de position sont issues d'un codeur dont la résolution n'est pas très haute. L'acquisition des données se fait à une fréquence d'échantillonnage  $f_e=200\text{Hz}$ . Il est à noter que le robot fonctionne dans de basses fréquences, donc une fréquence d'une vingtaine de hertz suffit (par application du théorème de Shonnon). Toutefois, le choix de cette fréquence est lié aussi à la nature du problème la méthode d'estimation des différences finies qui voit ses performances dégradées si la période d'échantillonnage est très élevée. Cette dégradation se manifeste en particulier dans le bruit dans le signal de vitesse et d'accélération dû à la quantification du signal de position (Bélangier, 1991 ; Bobrovolny, 1994). Pour éliminer les bruits de mesure un filtrage numérique hors-ligne est nécessaire en utilisant Matlab. Il s'agit d'un filtre de butterworth d'ordre 4 et de fréquence de coupure de 3Hz. Ce filtrage concerne uniquement la position pour éliminer les bruits de mesure et la vitesse pour éliminer les bruits de dérivation. La vitesse et l'accélération sont obtenues par dérivations numériques en utilisant la méthode d'estimation des différences finies. En pratique, nous appliquons les opérations de dérivation sur les données filtrées de position. Les trajectoires de position, vitesse et accélération pour les trois liaisons sont montrées en figures V.4, V.5, et V.6 respectivement. Nous remarquons que les trajectoires de vitesse sont légèrement bruitées. Nous n'avons pas voulu trop les lisser par peur de perdre certaines données utiles dans le signal. Il est à noter aussi que le bruit est plus important sur l'accélération que sur la vitesse. Nous remarquons aussi que les trajectoires de vitesse et d'accélération pour les trois liaisons possèdent des zones où ils s'annulent. Par conséquent, la matrice d'information devient nulle et toute cette zone n'est pas utile pour l'identification .



(a)

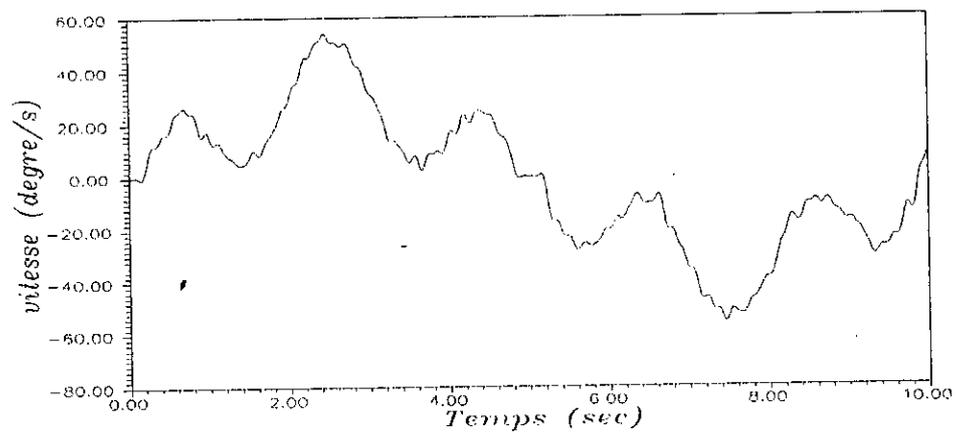


(b)

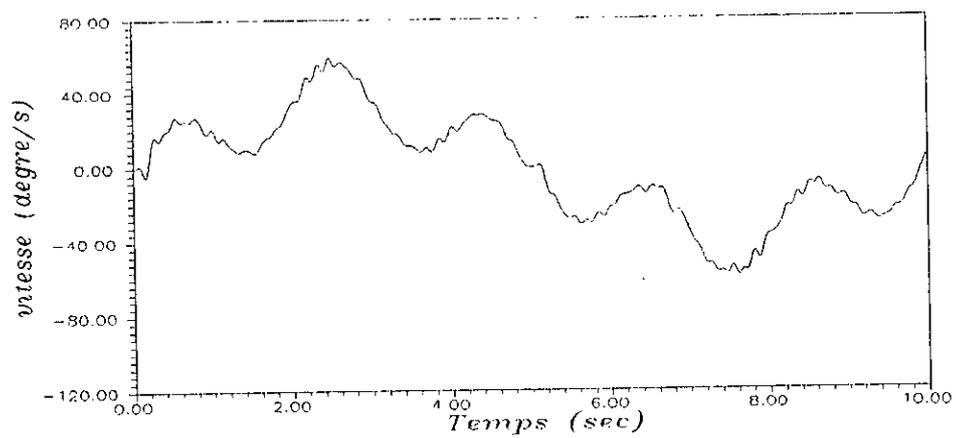


(c)

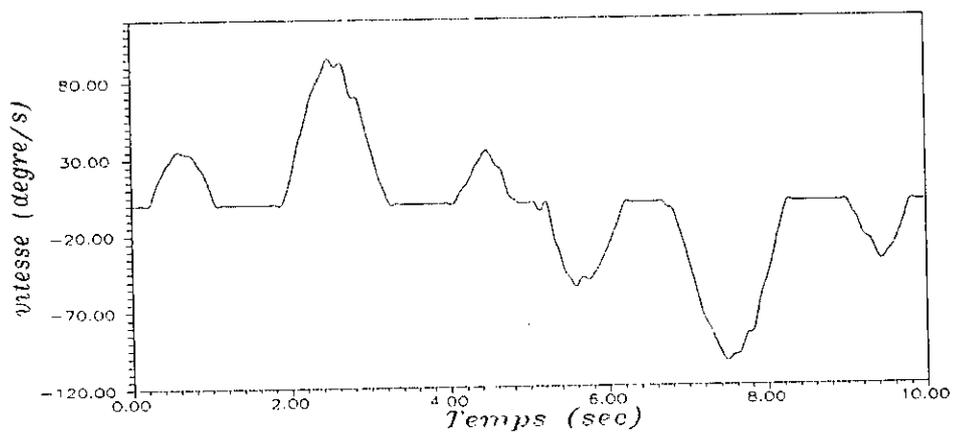
Figure V.4. Trajectoires de position.  
(a) Liaison 1 ; (b) Liaison 2 ; (c) Liaison 3



(a)

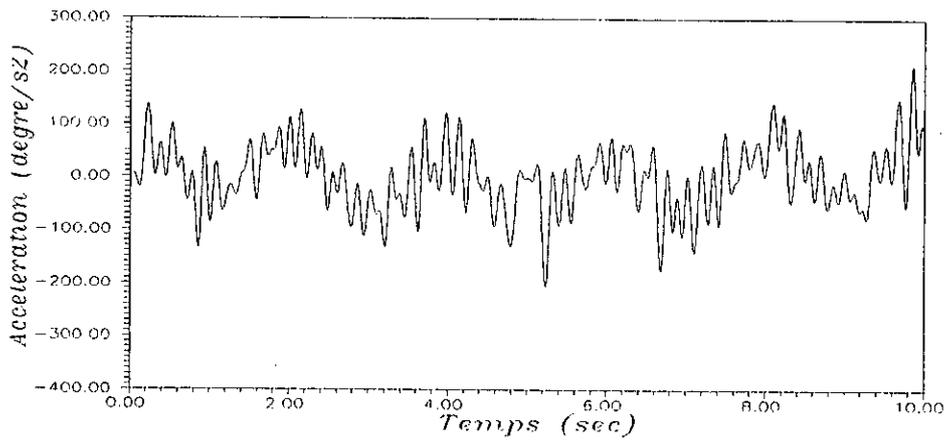


(b)

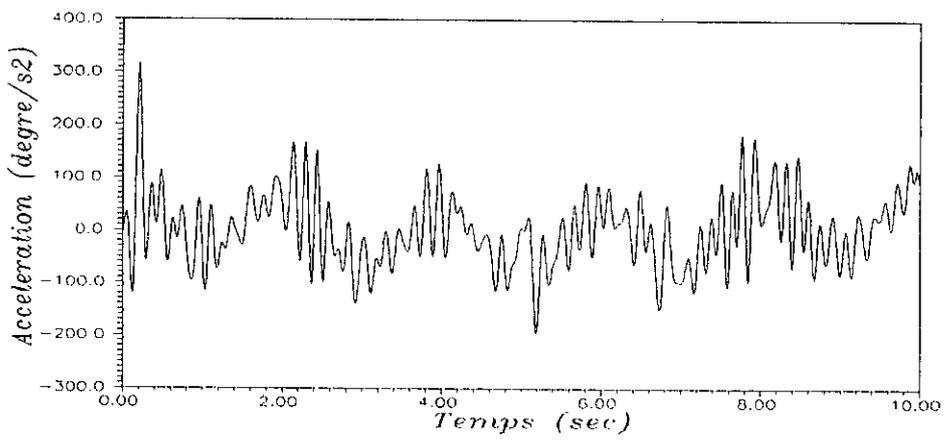


(c)

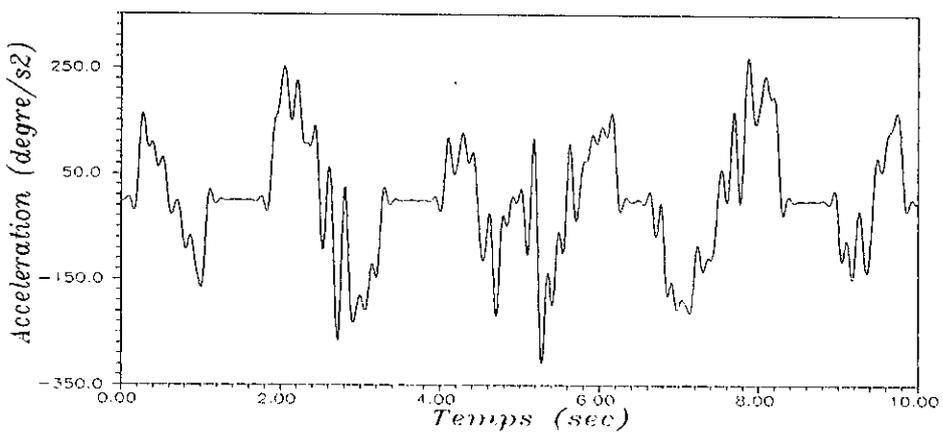
Figure V.5. Trajectoires de vitesse.  
(a) Liaison 1 ; (b) Liaison 2 ; (c) Liaison 3



(a)



(b)



(c)

Figure V.6. Trajectoires d'accélération.  
(a) Liaison 1 ; (b) Liaison 2 ; (c) Liaison 3

Deux milles points ont été recueillis pendant le fonctionnement du robot. Ces données sont utilisées pour trouver l'ensemble des paramètres dynamiques du robot. En utilisant l'équation dynamique linéaire par rapport aux paramètres, et la fonction des moindres-carrés de Matlab, nous avons obtenu les résultats suivants pour les paramètres dynamiques:

#### Paramètres inertiels

Axe 1 :  $p_1=0.1658$ ,  $p_2=0.0149$ ,  $p_3=0.0274$ ,  $p_4=0.1272$ ,

Axe 2 :  $p_5=0.1352$ ,  $p_6=0.0010$ ,  $p_7=0.1656$ ,  $p_8=0.0826$ ,

Axe 3 :  $p_9=0.0763$ ,  $p_{10}=0.0924$ ,

#### Paramètres de frottements :

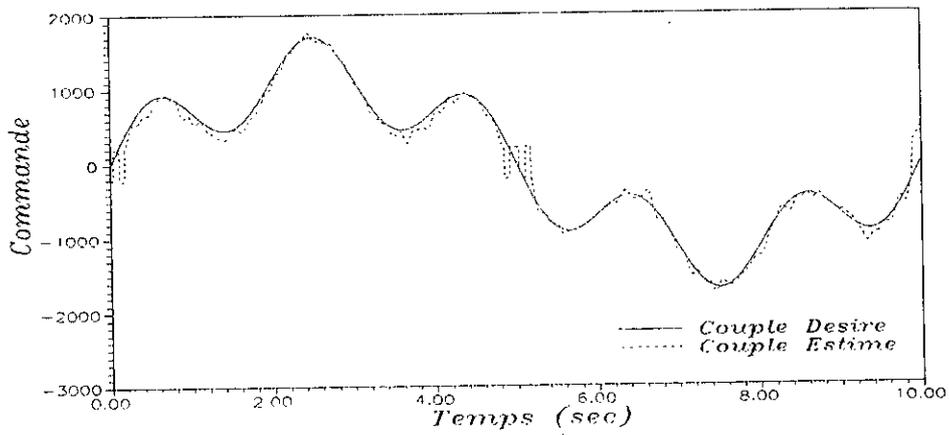
Axe 1 :  $p_{11}=28.1942$ ,  $p_{12}=200.6506$ ,

Axe 2 :  $p_{13}=27.3225$ ,  $p_{14}=147.3151$ ,

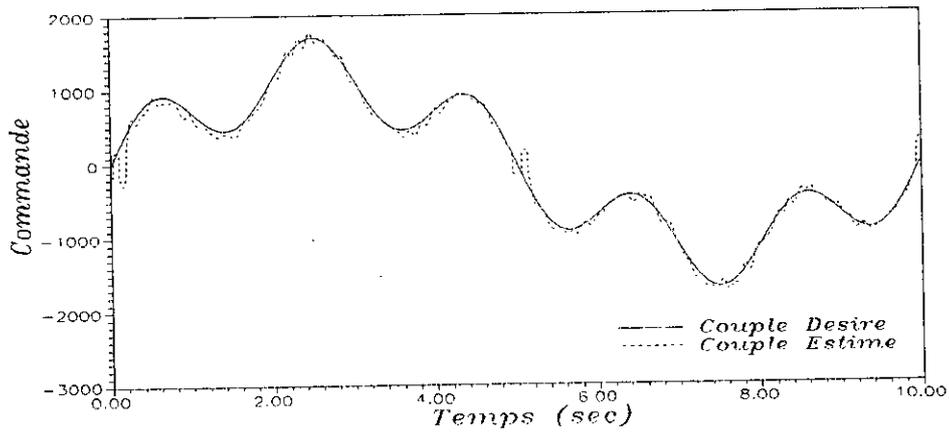
Axe 3 :  $p_{15}=5.1042$ ,  $p_{16}=55.0465$ .

Tous les paramètres, inertiels et de frottements, ont été identifiés. Nous remarquons que les paramètres de frottements sont beaucoup plus élevés en valeur que les paramètres inertiels. Ceci peut être expliqué par le fait que le robot manipulateur (RP41) possède de petits actionneurs et utilise des rapports de réduction très élevés. Cet arrangement des actionneurs introduit beaucoup de non linéarités indésirables telles que les frottements. Cet effet a été constaté dans le cas d'un robot PUMA 600 du laboratoire d'Intelligence Artificielle de MIT, où les termes de frottements approchés les 50% des couples moteurs (An, 1986 ; An et al., 1989). En effet, les effets de la dynamique du corps rigide des liaisons sont très petits pour ces manipulateurs à grand rapport de réduction puisque leurs effets sont réduits par le rapport de réduction, et d'un autre côté les facteurs de frottements sont multipliés par le rapport de réduction.

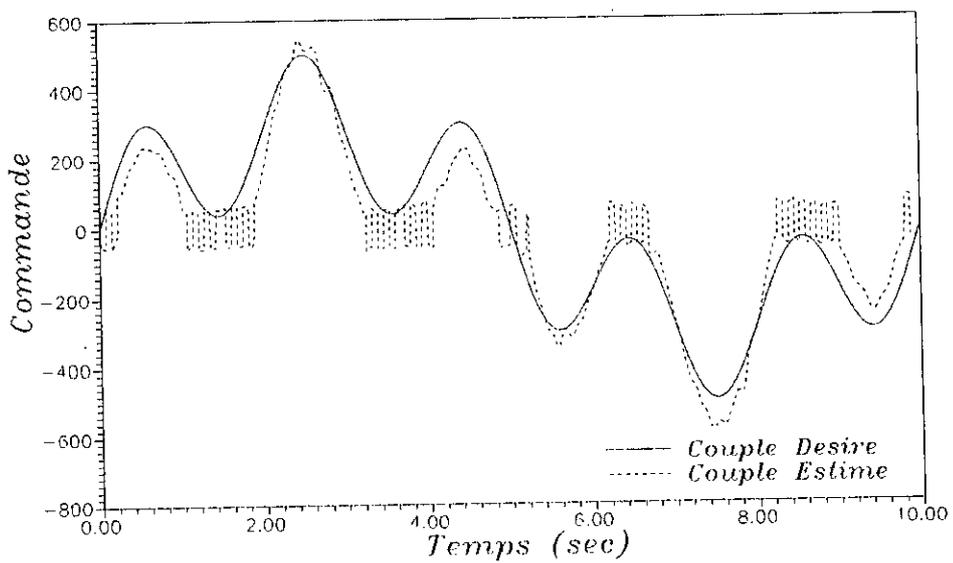
La précision de l'estimation est aussi vérifiée en comparant le couple désiré avec le couple estimé en utilisant les paramètres précédents. Comme montré en figure V.7, le couple estimé suit le couple désiré. Il est à noter toutefois que dans certaines zones de la trajectoire des couples, l'erreur est plus importante. Ces zones correspondent exactement aux zones où la matrice d'information s'annule ou bien elle devient mal conditionnée. Ce fait était bien sûr attendu du fait de la nature des trajectoires de vitesse et d'accélération.



(a)



(b)



(c)

Figure V.7. Couple Désiré et Couple Estimé.  
 (a) Liaison 1 ; (b) Liaison 2 ; (c) Liaison 3

## V.4 Mise En Œuvre Experimentale De La Commande

### Neuro-Dynamique Adaptative

Après l'obtention du modèle dynamique du robot RP41 et l'identification de ces paramètres dynamiques, nous avons validé ce modèle dynamique en l'utilisant pour fournir le couple d'anticipation avec une correction PD à gains fixes (Bouyoucef et al., 1998). Une comparaison expérimentale de cette commande avec un simple correcteur PD à gains fixes nous a permis de constater la validité du modèle dynamique obtenu (Kadri et al., 1997b) (voir Annexe C).

Le modèle ainsi obtenu et validé subit une décomposition en parties fixe et variable en considérant la masse du dernier corps comme paramètre variable. Nous obtenons les expressions suivantes :

Couple variable:

$$u_{1v} = \frac{1}{K_1} \left( [m_3(L_1^2 + L_2^2) + 2L_1m_3L_2 \cos(q_2)] \ddot{q}_1 \right) + \frac{1}{K_1} \left( [m_3L_2^2 + L_1m_3L_2 \cos(q_2)] \ddot{q}_2 \right) - \frac{1}{K_1} (L_1m_3L_2 \sin(q_2) \dot{q}_2 + 2L_1m_3L_2 \sin(q_2) \dot{q}_1 \dot{q}_2)$$

$$u_{2v} = \frac{1}{K_2} \left( [m_3L_2^2 + L_1m_3L_2 \cos(q_2)] \ddot{q}_1 + [m_3L_2^2] \ddot{q}_2 \right) + \frac{1}{K_2} (L_1m_3L_2^2 \sin(q_2) \dot{q}_1^2) \quad (V.6)$$

$$u_{3v} = 0$$

Couple fixe : Le couple fixe étant celui donné par les expressions (V.2a-V.2c).

Pour la mise en œuvre expérimentale du système nous avons considéré le robot initialement chargé avec une masse de 1.5 Kg qui tombe en cours de trajet du bras manipulateur. Etant donnée que les frottements sont intégrés dans la dynamique du robot, nous avons supposé comme perturbations les erreurs de dérivation (pour l'obtention de la vitesse), de modélisation et d'estimation. Les jeux et la flexibilité peuvent être considérés dans ce cas comme dynamiques non modélisés.

L'architecture des réseaux de neurones utilisés a été déterminée après plusieurs essais en se basant sur la convergence de l'erreur d'apprentissage et la généralisation de ces réseaux. Ces derniers ont été entraînés, avec l'algorithme rétropropagation, en hors ligne en utilisant des données aléatoires sur la base du modèle dynamique identifié et décomposé en partie fixe et variable. Par contre dans le cas de l'estimateur neural nous avons utilisé des données expérimentales pour l'étape d'entraînement en se basant uniquement sur les couples exercés sur les deux premières liaisons ainsi que leur positions.

### V.4.1 Commande Neurale Dynamique Anticipée Adaptative

Dans le cas de la première variante nous avons utilisé des tables pour fournir les valeurs du couple fixe pour chaque liaison. Le couple variable est fournit par des réseaux de neurones et pour des raisons pratiques nous avons utilisé un réseau pour chaque liaison. L'architecture de ces réseaux est donnée comme suit :

- Première liaison : 5 neurones en entrées (position de la deuxième liaison, vitesse et accélération des deux premières liaisons), 20 neurones dans la couche cachée et 1 neurone en sortie.
- Deuxième liaison : 4 neurones en entrées (position de la deuxième liaison, vitesse de la première liaison et accélération des deux premières liaisons), 20 neurones dans la couche cachée et 1 neurone en sortie.

### V.4.2 Commande Neurale par Couple Calculé Adaptative

Dans le cas de la deuxième variante le modèle dynamique est décomposé, en plus de la décomposition en parties fixe et variable, en matrice d'inertie  $M$  et vecteur de Coriolis et centrifuge  $H$ . Cette décomposition nous a permis de réduire considérablement la taille des réseaux utilisés. Contrairement à la première variante les couples fixe et variable sont fournis par des réseaux de neurones dans l'architecture est donnée comme suit :

- Matrices d'inertie fixe et variable : 1 neurone en entrée (position de la deuxième liaison), 13 neurones dans la couche cachée et 9 neurones en sortie (chacune).
- Termes de coriolis et centrifuges fixe et variable : 3 neurones en entrée (position de la deuxième liaison et vitesse des deux premières liaisons), 9 neurones dans la couche cachée et 3 neurones en sortie (chacun).

L'estimateur de la masse de la charge, pour les deux variantes, possède 4 neurones en entrées (position et couple des deux premières liaisons), 20 et 10 neurones dans les couches cachées respectivement et 1 neurone en sortie.

Le correcteur PD adaptatif possède les paramètres d'adaptation de ces gains comme indiquer dans le tableau suivant.

Liaison	$G_p$	$G_v$	$K_{p0}$	$K_{v0}$	$r_0$	$\alpha_0$	$\alpha_1$
1	1.5	1.5	50	2	0	0.01	0.01
2	1.5	1.5	50	2	0	0.01	0.01
3	0.5	0.5	25	0.06	0	0.01	0.01

La trajectoire de référence est une sinusoïde donnée par l'expression suivante :

Pour la première et la deuxième liaison :  $a_1 + \cos(2\pi f_1 t)$  avec  $a_1=2$  rad,  $f_1=0.1$  Hertz.

Pour la troisième liaison:  $a_2 + \cos(2\pi f_2 t)$  avec  $a_2=2$  rad,  $f_2=0.5$  Hertz.

### V.4.3 Résultats

Les résultats expérimentaux de la première et la seconde variante de notre solution de commande, illustrés par les figures de V.8 à V.14, ont permis de valider notre approche. Il est à noter que l'estimateur de la masse a fourni une très bonne estimation comme montré en figure V.8. ce résultats nous confirme la robustesse de cet estimateur vis à vis des perturbations de faibles amplitudes. Pour les résultats de poursuite, nous constatons en particulier que chaque liaison effectue une bonne poursuite de la trajectoire de référence. La variation brusque de la masse de la charge n'a pas perturbé le système de commande ce qui indique que notre approche est robuste vis à vis de la variation paramétrique. Les autres perturbations ont pu être compensées par l'action de retour adaptative. Il est à noter qu'au départ nous remarquons un grand écart entre les trajectoires de référence et réelle. Cette erreur est due au fait que la trajectoire de référence n'est pas initialisée à la position initiale réelle du bras manipulateur à cause des butées mécaniques. Nous avons considéré, par conséquent, ceci comme une perturbation que le correcteur PD adaptatif doit compenser et que les résultats obtenus confirmés. Les deux variantes de l'approche de commande fournissent pratiquement les mêmes performances en qualité de poursuite de trajectoire.

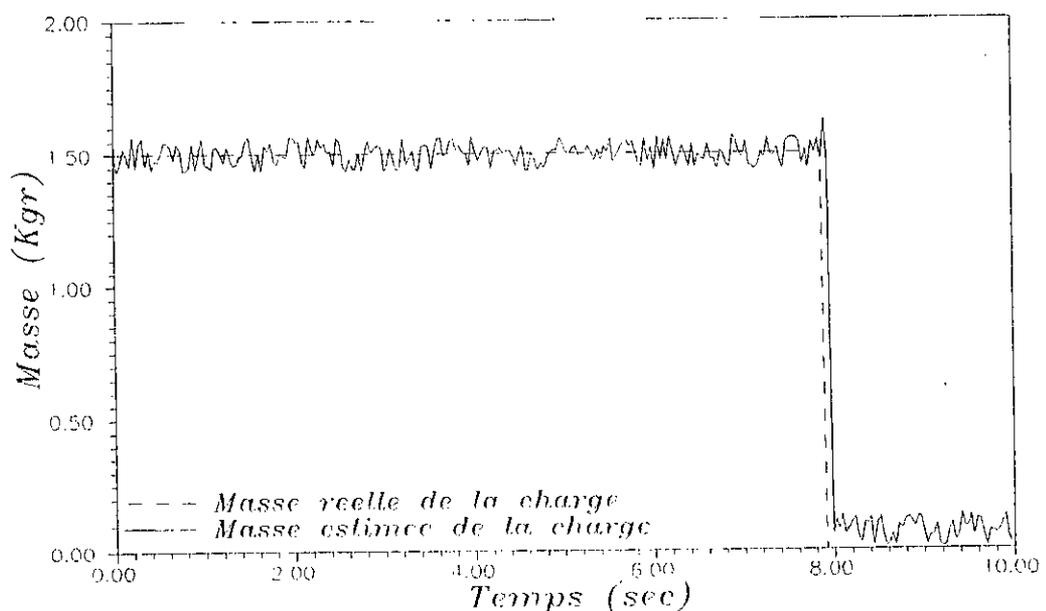
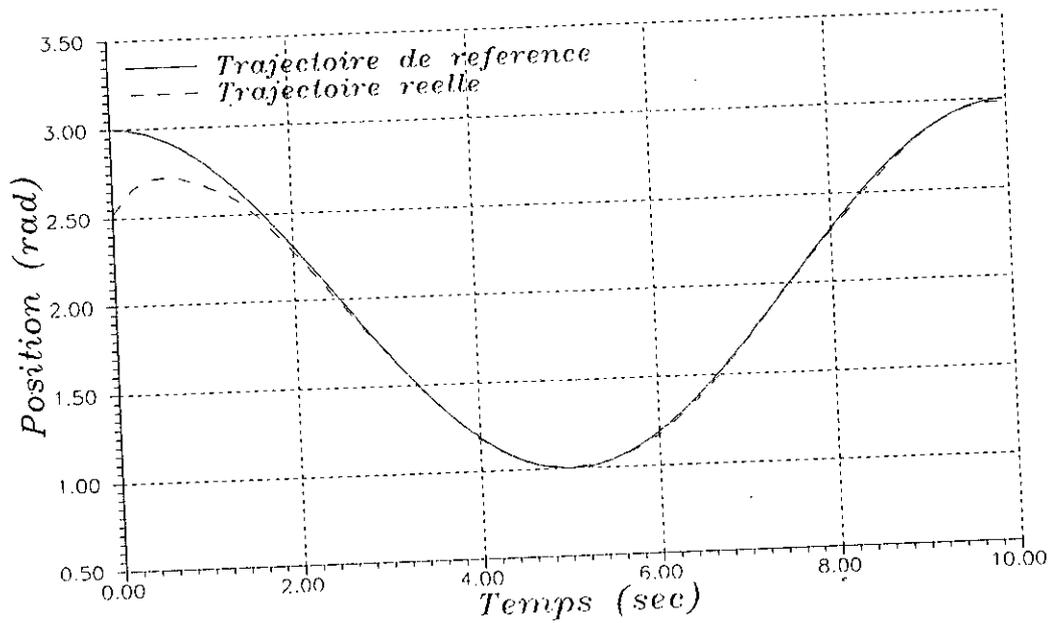
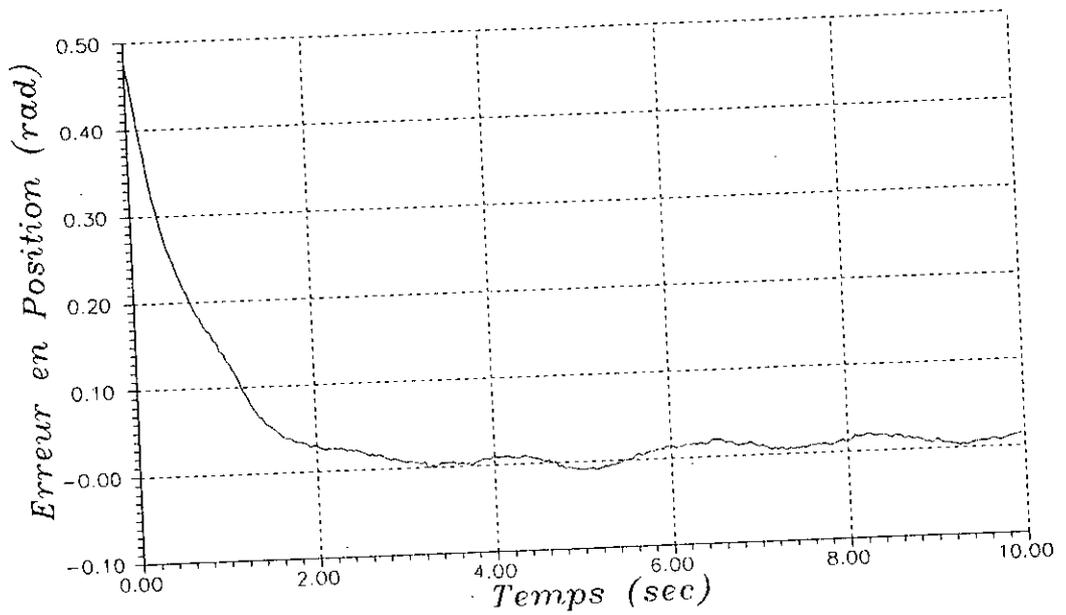


Figure V.8. Estimation de la masse de la charge.

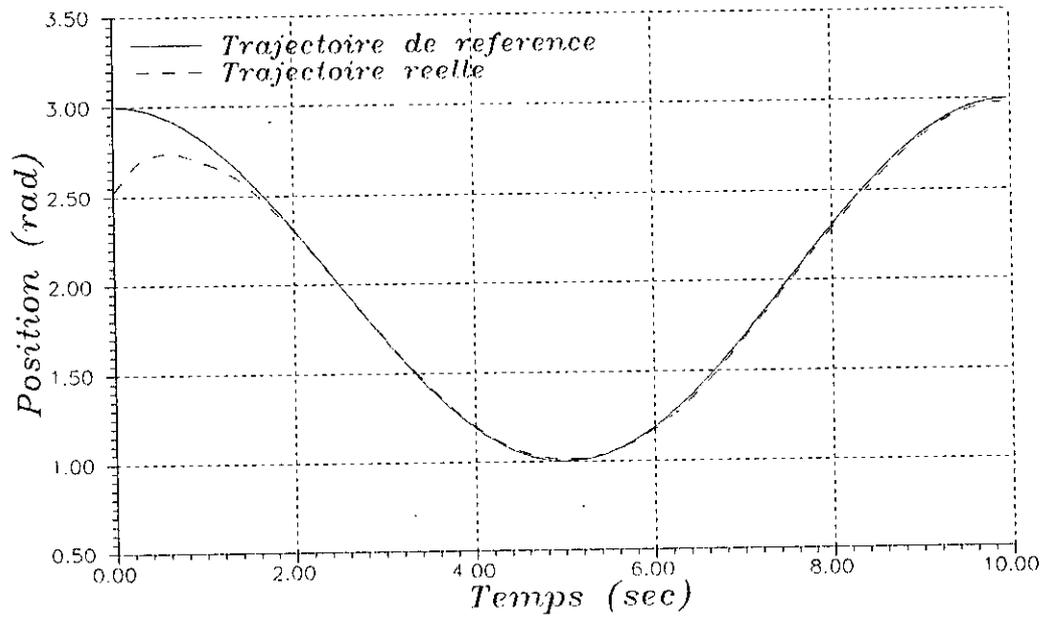


(a)

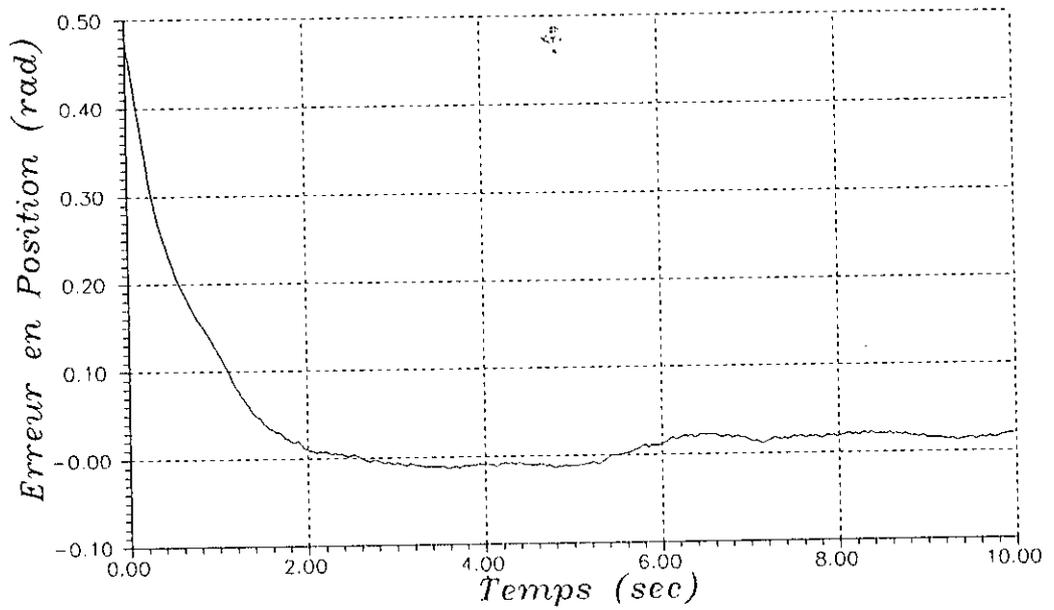


(b)

Figure V.9. Commande Neurale Dynamique Anticipée Adaptative Liaison 1, (a) : Trajectoire de position, (b) : Erreur de poursuite.

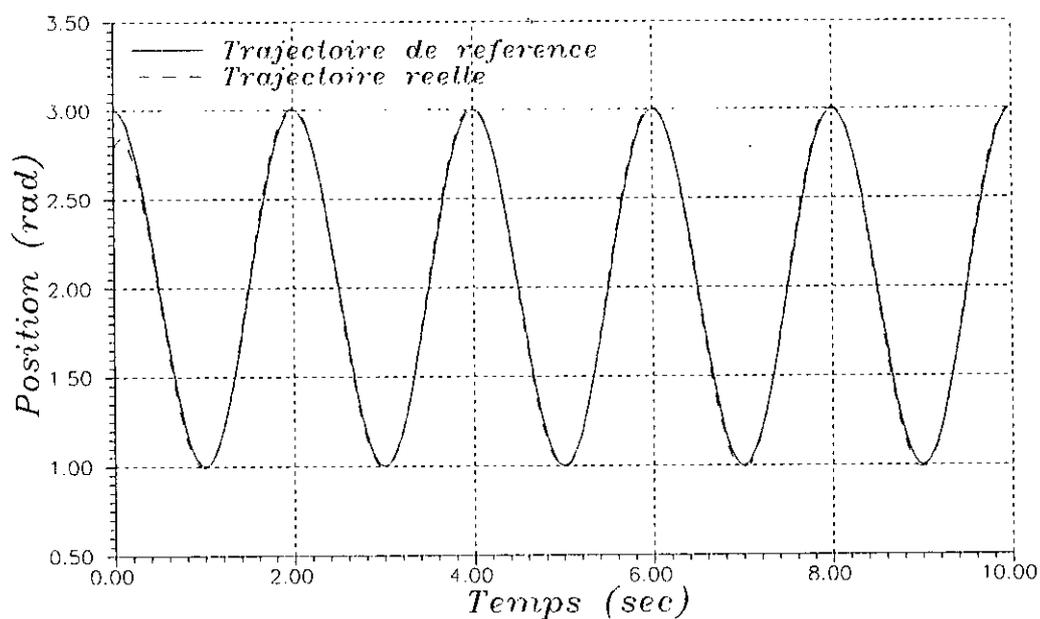


(a)

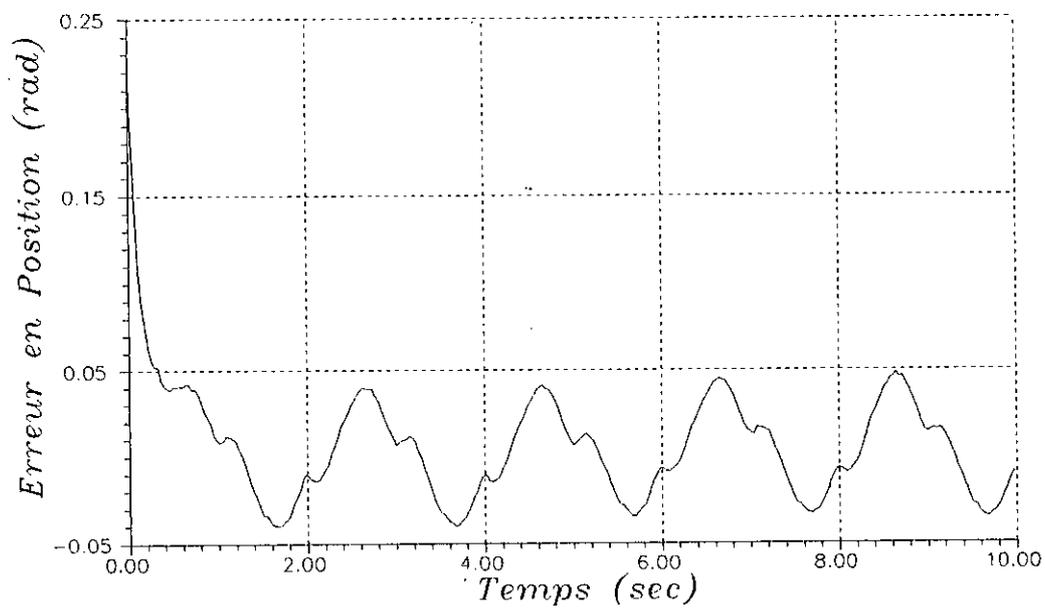


(b)

Figure V.10. Commande Neurale Dynamique Anticipée Adaptative Liaison 2, (a) : Trajectoire de position, (b) : Erreur de poursuite.

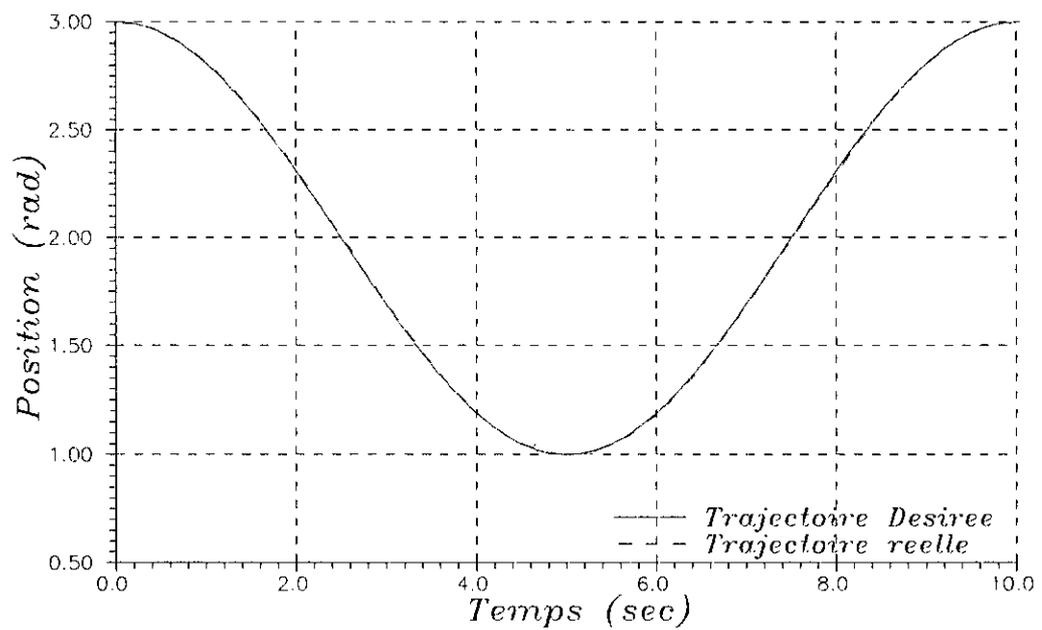


(a)

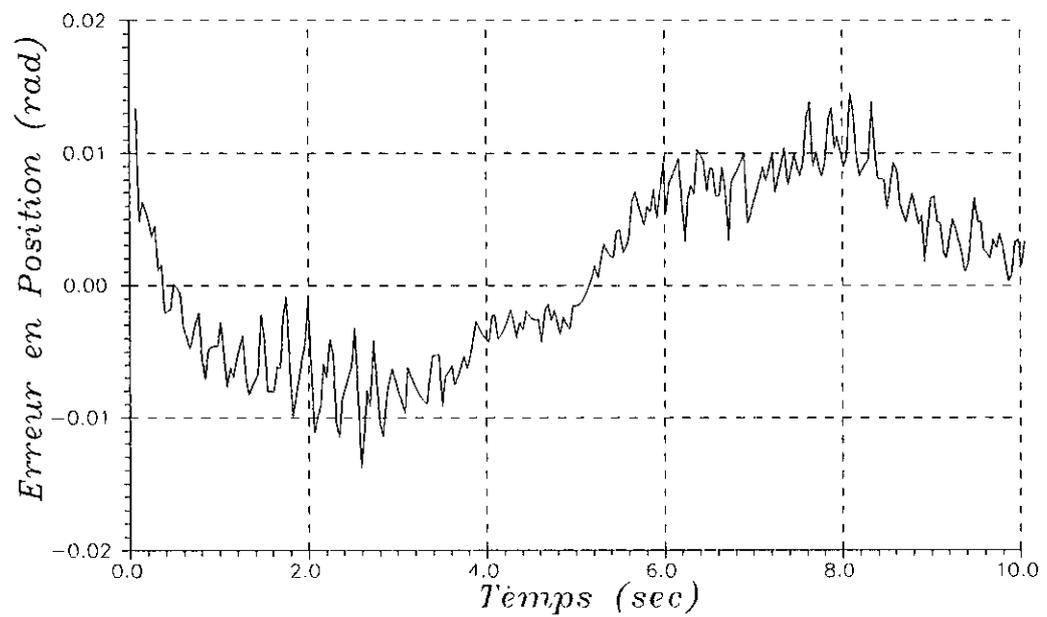


(b)

Figure V.11. Commande Neurale Dynamique Anticipée Adaptative Liaison 3, (a) : Trajectoire de position, (b) : Erreur de poursuite.

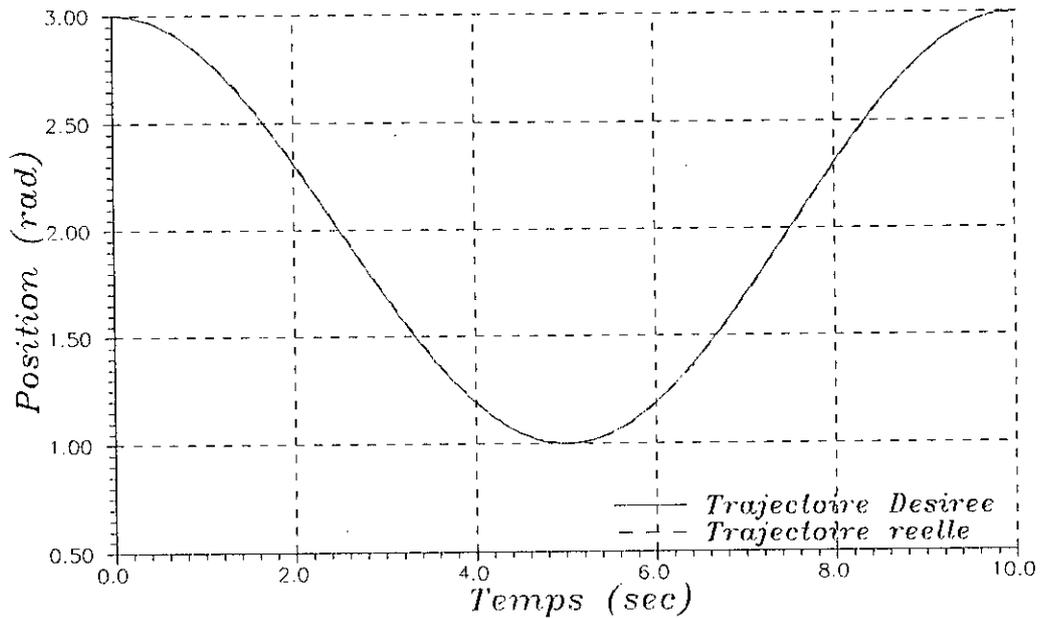


(a)

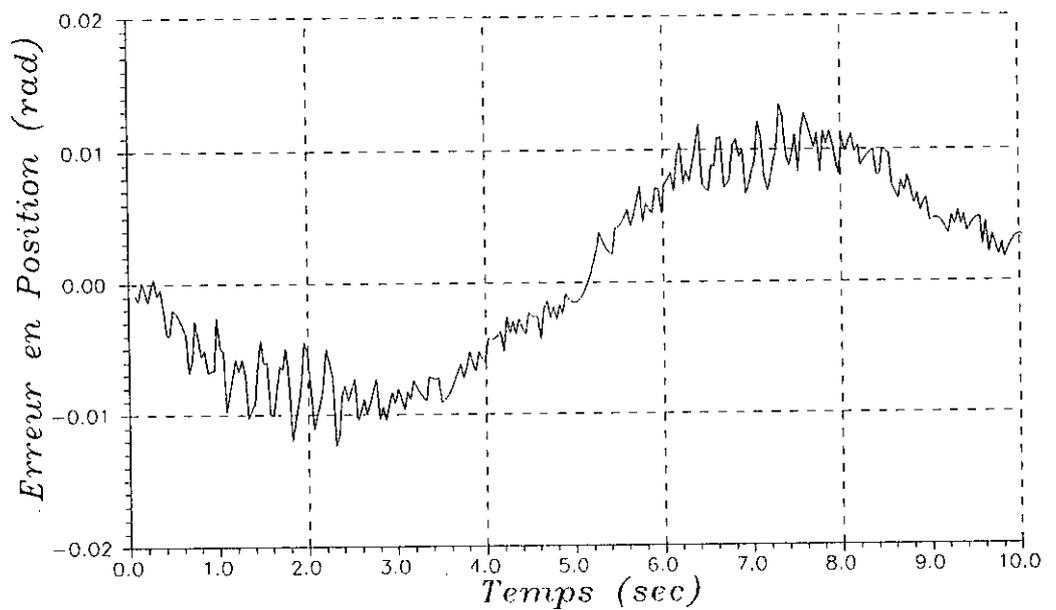


(b)

Figure V.12. Commande Neurale par Couple Calculé Adaptative  
Liaison 1, (a) : Trajectoire de position, (b) : Erreur de poursuite.

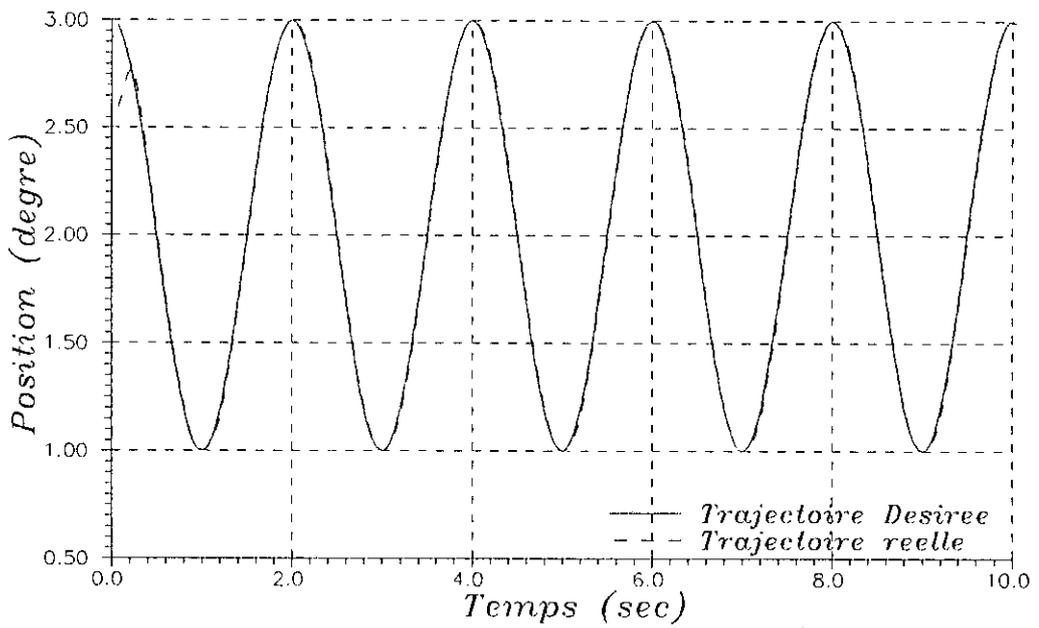


(a)

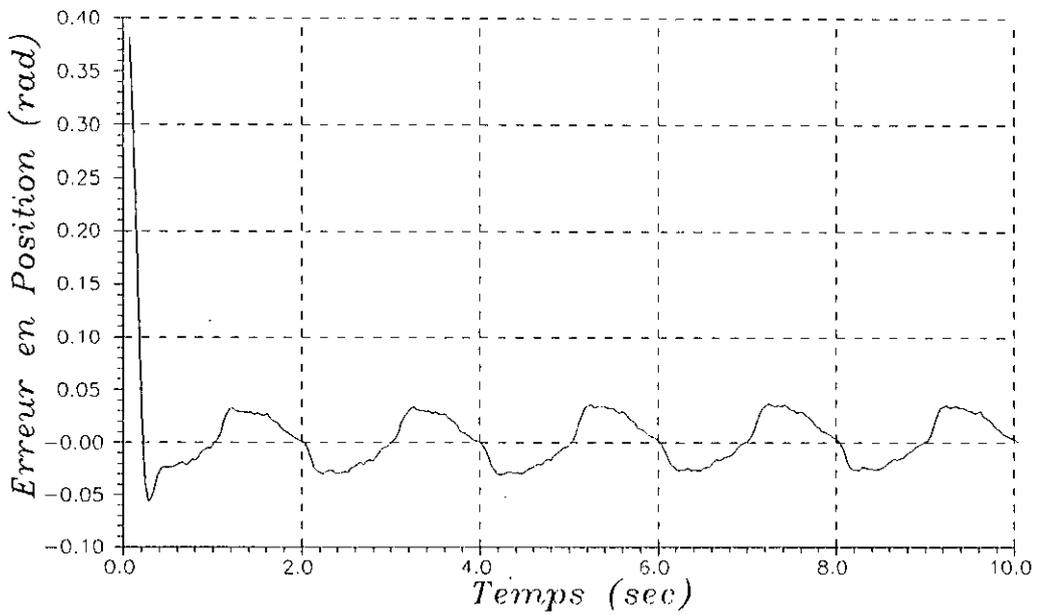


(b)

Figure V.13. Commande Neurale par Couple Calculé Adaptative Liaison 2, (a) : Trajectoire de position, (b) : Erreur de poursuite.



(a)



(b)

Figure V.14. Commande Neurale par Couple Calculé Adaptative Liaison 3, (a) : Trajectoire de position, (b) : Erreur de poursuite.

## ***CONCLUSION ET PERSPECTIVES***

---

---

## Conclusion

Dans ce travail une nouvelle et originale méthode de commande en position et en vitesse des robots manipulateurs a été proposée, validée et expérimentée sur le robot RP41. Le principe de cette méthode est de combiner trois techniques Neurale, Dynamique et adaptative. En se basant sur la décomposition du modèle dynamique du bras en parties fixe et variable, l'action Neuro-Dynamique prend en charge l'aspect non linéaire des robots manipulateurs. La partie variable de cette action peut même surmonter la variation des paramètres dynamiques.

La correction PD adaptative compense les erreurs de modélisation, d'estimation et autres perturbations, en particulier celles des dynamiques non modélisées.

Afin de valider cette méthode une simulation numérique a été entreprise sur un bras manipulateur à deux liaisons rotoïdes. Les résultats de la simulation ont montré l'efficacité de la méthode proposée, surtout dans sa partie Neuro-Dynamique, face aux problèmes liés à de la nature non linéaire des robots manipulateurs et à la variation paramétrique. Cette efficacité est due essentiellement, d'un côté au rôle joué par les réseaux de neurones dans la modélisation de la dynamique des robots manipulateurs et dans l'estimation des paramètres variables. D'un autre côté, la décomposition du modèle en parties fixe et variable a permis de poursuivre la variation dans la dynamique du robot. Les autres perturbations telles que les frottements ont été compensées efficacement par la correction PD adaptative. Ainsi la méthode, dans son principe de combinaison et de décomposition, a montré sa robustesse vis-à-vis de la variation paramétrique et la de présence des dynamiques non modélisées.

Les performances fournies par l'approche de la commande proposée ont fait l'objet d'une comparaison avec celles de l'approche de commande suggérée par Ozaki. Cette comparaison nous a permis de constater, d'un côté l'apport de la décomposition du modèle dynamique du robot manipulateur en parties fixe et variable. D'un autre côté, elle a mis en évidence l'importance de l'estimation des paramètres variables lors de la prise et manipulation des objets.

La validation expérimentale, de l'approche de la commande proposée, a été effectuée sur un robot (RP41 du CDTA) à trois liaisons rotoïdes. Du fait que dans le cas de notre approche de commande nous avons eu besoin du modèle dynamique du robot, une modélisation et une identification du robot manipulateur (RP41) furent nécessaires.

La méthode de commande développée a été implémentée avec succès et a donné des résultats très satisfaisants. Cette implémentation a en particulier confirmé sa robustesse

vis-à-vis de la variation paramétrique et de la présence des dynamiques non modélisées. La mise en œuvre n'a exigé ni un temps de calcul important ni un grand espace de mémoire. Ceci est acquis grâce à la décomposition du modèle dynamique en parties fixe et variable. La correction PD adaptative a pu quant à elle compenser les perturbations non prises en compte dans la partie d'anticipation Neuro-Dynamique.

### *Perspectives*

La commande des robots manipulateurs à l'aide des réseaux de neurones a fait l'objet de plusieurs études de recherche et une saturation semble attendue. Toutefois le présent travail peut être enrichi par les tâches suivantes :

- généraliser l'estimation à d'autres paramètres variables.
- l'optimisation des réseaux de neurones.
- une implémentation VLSI de la commande.

Aussi il est intéressant d'aborder le problème de la commande pour un multi-robot. Ce problème est d'actualité d'autant plus qu'en pratique une cellule flexible de production comprend plusieurs robots en général.

En fin un autre domaine qui reste aussi à explorer est l'approche de la commande de force/position qui suscite ces dernières années l'intérêt croissant chez la communauté scientifique.

## ***BIBLIOGRAPHIE***

---

---

- An C. H. (1986), Trajectory and Force Control of a Direct Drive Arm, *Ph.D. Thesis, MIT, Dept. Of Electrical Engineering and Computer Science.*
- An C. H., Atkeson C.G., Griffiths J.D., and Hollerbach J.M. (1989), "Experimental Avaluation of Feedforward and Computed Torque Control", *IEEE Trans. Robotics and Automation*, Vol. 5, No. 3, pp 368-373.
- Armstrong B. (1986), O. Khatib, and J. Burdick, "The explicit dynamic model and inertial parameters of the PUMA-560 arm," *Proceeding. of IEEE International Conference. On Robotics and Automation*, pp. 510-518.
- Armstrong B. (1987), "On finding exciting trajectories for identification experiments involving systems with nonlinear dynamics," *Proceeding. of IEEE International Conference On Robotics and Automation*, pp. 1131-1139.
- Armstrong B. (1988), "Friction : Experimental determination, modeling and compensation," *Proc. of IEEE International Conf. On Robotics and Automation*, pp. 1422-1427.
- Bavarian, B., (1988). " Introduction to Neural Networks for Intelligent Control " , *IEEE Control Systems Magazine*, Vol 8, No.2, pp 3-7.
- Bélangier Pierre R., Estimation Of Angular Velocity And Acceleration From Shaft Encoder Measurments, TR-CIM-91-1, Computer Vision and Robotics Laboratory, Mc Gill University, Montréal, Québec, Canada.
- Billing S., Jamaluddin, H. B., and chen S., (1991), "A comparaisn of the backpropagation and recursive prediction error algorithms for training neural networks ", *Mechanical System and Signal Processing*, Vol. 5, No.3, pp 233-255.
- Billing S. and chen S.(1992) "Neural networks and system identification" *In Neural Networks for control and systems* Edited by K. Warwick, G.W Irwin, K.J. Hunt. Peter Peregrinus.
- Burrows T.L. ans Niranjan M., (1993), "The use of feed-forward and recurrent neural networks for system identification", *TR 158, Engineering Department, Cambridge University*, Trumpington Street Cambridge CB2 1PZ England.
- Bouyoucef M.K, Kadri M., Bouzouia B., (1998), " Identification Expérimentale de la Dynamique du Robot Manipulateur RP41", *Proceeding du 1er colloque National de la Productique. Tizi-Ouzou*, 30-31 pp 346-352.**
- Cichocki, A., and Unbehauen, R., (1993) *Neural Networks for Optimization and Signal Processing*. John Wiley & Sons Ltd. & B.G..Tenbner.
- Coiffet P. (1981), *Les Robots, Tome 1: Modélisation et Commande*, Herms Publishing, France.

- Craig, John J., (1986), Introduction to Robotics Mechanics & Control. Addison-Wesley Publishing Company.
- Dobrovolny P., Kalman Filter Estimation Angular Velocity And Acceleration, TR-CIM-94-04, Computer Vision and Robotics Laboratory, Mc Gill University, Montréal, Québec, Canada.
- Egeland, O., (1986) "on the robustness of the computed torque technique in manipulator control", *Proceeding of IEEE International Conference on Robotics and Automation, San Francisco*, pp 1203-1208
- Gilbert, E.G., and Ha, I.J., "An approach to nonlinear feedback control with applications to robotics", *IEEE Trans. System, Man, and Cybernetic*, Vol. 14, pp 879-884.
- Hsia, T.C., (1986) , " Adaptive control of robot manipulators - A review ", *Proceeding IEEE Conf. on Robotics and Automation*, pp 183-189.
- Hunt K.J. and Sbarbaro D., (1992), " Studies in neural network based control ", in *Neural Networks for control and systems*, K.Warwick, G.W.Irwin, and K.J.Hunt (Editors).
- Jervis Timothy T., (1993), Connectionist Adaptive Control, *PhD thesis, Cambridge University, Engineering Department*, Trumpington Street, Cambridge, England.
- Johnson M.A., and Leahy Jr. M.B.,(1990), "Adaptive Model-Based Neural Network Control", *Proceeding of the IEEE International Conference on Robotics and Automation*, pp 1704-1709.
- Jung, S., and Hsia, T.C., (1995), " A new neural network control technique for robot manipulators ", *Robotica*, Vol. 13, pp 477-487,
- Kadri M., Bouzouia B., Louam N., (1996a)**, " Identification et commande des systèmes par réseaux de neurones : Application aux robots manipulateurs", *Rapport de Recherche*, C.D.T.A. Alger.
- Kadri M., Bouzouia B., Louam N., (1996b)**, " Commande neuro/ dynamique/ Adaptative des Robots Manipulateurs", *Rapport de Recherche*, C.D.T.A. Alger, Novembre 1996.
- Kadri M., Bouzouia B., Louam N., (1997a)**, "Neural Decomposed Dynamic Adaptive control for Robot Manipulator" in *Proceedings of the Third ECDP International Conference on Advanced Robotics, Intelligent Automation and Active Systems*, pp 203-208, September 15-17, 1997, Bremen, Germany.
- Kadri M., Bouyoucef M.K, Bouzouia B., (1997b)**, " Modélisation et Identification du Robot RP41", *Rapport de Recherche*, C.D.T.A. Alger, Novembre 1997.
- Kadri M., Bouzouia B., Louam N., (1997c)**, "Neural Adaptive Decomposed Computed Torque control for Robot Manipulator" in *Proceedings of the International Annual Conference IEEA of The University of Batna*, Vol.2 pp 83-87, December 7-9, 1997, Batna, Algeria.

- Kadri M., Bouzouia B., Louam N., (1998),** “ model-based neural network controllers for robot manipulator undergoing inertial parameters variation and unmodeled dynamics: comparison”, *Proceeding du 1er colloque National de la Productique*. Tizi-Ouzou, 30-31 Mai, pp 326-332.
- Katić.D, and Vokobratović M.,(1992) “ Decomposed Connectionist Architecture for Fast and Robust Learning of Robot Dynamics, ” *Proceeding of IEEE Conference On Robotics and Automation*, pp 2064-2069.
- Kawato M., Uno Y., Isobe M., and Suzuki R., (1988), “hierarchical Neural Network Model for Voluntary Movement with Application to Robotics”, *IEEE Control Systems Magazine*, **Vol 8, No.2**, pp 8-16.
- Khosla P. K (1989)., “Categorisation of parameters in the dynamic robot model,” *IEEE Trans. On Robotics and Automation*, **No 5**, pp. 261-268.
- Kozlowski K. and Prüfer M., (1992) “Parameter Identification of an Experimental 2-Link Direct Drive Arm,” *Proceeding of the IASTED International Conference on Control and Robotics*, pp. 313-316.
- Krijgsman A.J., Verbruggen H.B., Rodd M.G., (1992), “Intelligent Control : Theory and Applications”, *IFAC Intelligent Components and Instruments for Control Applications*, pp 57-67, Malaga, Spain.
- Landau, Y.D., (1979) Adaptive control. The model reference approach. Dekker Inc, New York.
- Leahy, M.B.Jr., Valavanis, K. P., and Saridis, G. N., (1989), “Evaluation of Dynamic Models for PUMA Robot Control”, *IEEE Trans. On Robotics and Automation*, **Vol. 5, No. 2**, pp 242-245.
- Leahy, M.B.Jr., Johnson, M.A. and Rogers, S.K., (1991) , “ Neural Network Payload Estimation for Adaptive Robot Control ”, *IEEE Trans. on Neural Networks*, **Vol.2, No.1**, pp 93-100.
- Lippmann, R.P., (1987) “ An introduction to computing with neural nets, ” *Acoust., speech, Signal process. Magazine* , **Vol 4**, pp 4-22, Apr 1987.
- Lu, Z. Shimoga, K. B. and Goldenberg, A. A. (1993). “Experimental Determination of Dynamic Parameters of Robotic Arms”, *Journal of Robotic Systems*. (By John Wiley & Sons inc, ed). **Vol. 10, No. 8**, pp. 1009-1029.
- Miller W.T., Hewes R.P., Glanz F.H., and Kraft L.G.,(1990) “ Real-Time Dynamic Control of an Industrial Manipulator Using a Neural Net-work Based Learning Controller”, *IEEE Trans. On Robotics and Automation*, **Vol.6, No.1**, pp 1-9.
- Narandra, K.S.and Parthasarathy, K., (1990), “ Identification and Control of Dynamical System Using Neural Networks ”, *IEEE Trans. on Neural Networks*, **Vol.1, No.1**, pp 4-27.

- Newton, R.T. and Xu, Y., (1993), "Real-time Implementation of Neural Network Learning Control of a Flexible Space Manipulator", *Proceeding of the IEEE International Conference on Robotics and Automation*, **Vol. 1**, pp135-141.
- Olsen, H.B., and Bekey, G.A., (1985), "Identification of parameters in models of robots without rotary joints", *Proceeding of the IEEE International Conference on Robotics and Automation*, pp. 1025-1030.
- Ozaki, T. Suzuki, T. Furuhashi, T. Okuma, S. and Uchikawa, Y., (1991), "Trajectory control of robotic manipulators using neural networks", *IEEE Trans. on Industrial Electronics*, **Vol.38, No.3**, pp 195-202.
- Paul, R.P., (1981), *Robotic manipulators, programming and control*, Mit Press.
- Pham C.M. (1992), *Identification et Commande des Robots Souples, Thèse de Doctorat, Université de Nantes, France.*
- Prüfer M., Schmidt C. and Wahl F. (1994), "Identification of Robot Dynamics with Differential and Integral Models: Comparison," *Proceeding of the IEEE International Conference on Robotics and Automation*, pp. 340-345.
- Prüfer M. and Schmidt C. (1996), "Practical Aspects Of Robot Dynamics Identification," *Proceeding of the Third International Symposium on Methods and Models in Automation and Robotics*, **Vol. 3** pp. 971-976.
- Psaltis, D. Sideris A., and Yamamura, A., (1987), "Neural Controllers", *Proceeding of IEEE International conference on Neural Networks, San-Diego, CA*, **Vol IV**, pp 551-558.
- Psaltis, D. Sideris A., and Yamamura, A., (1988), "A Multilayered Neural Network Controller", *IEEE Control Systems Magazine*, **Vol 8, No.2**, pp 17-21.
- Qin, S.Z., Su, H. and Mc Avoy, T.J., (1992), "Comparison of four Neural Net Learning Methods for dynamic system identification", *IEEE Trans. on Neural Networks*, **Vol.3, No.1**, pp 122-130.
- Saad M., Dessaint L.A., Bigras P., and Al-Haddad K., (1994), "Adaptive Versus Neural Adaptive Control : Application to Robotics", *International Journal of Adaptive Control and Signal Processing*, **Vol.8**, pp 223-236.
- Schilling R.J. (1990), *Fundamentals of Robotics : Analysis & control*. Printice-Hall, Inc.
- Seraji, H., (1989) "Decentralized Adaptive Control of Manipulators: Theory, Simulation, and Experimentation." *IEEE Trans. on Robotics and Automation*, **Vol.5, No.2**, pp 189-201.
- Tosunoglu, S. and Tesar, D., (1987), "A survey of adaptive control technology in robotics", *The*

*NASA WorkShop on Space Telerobotics, Jet Propulsion Laboratory Pasadena, California.*

Vukobratovic, M., *Applied Control of Manipulation Robots*

Wampler C.W. (1985), *Computer Methods in Manipulators Kinematics, Dynamics and Control : A comparative study. Phd Thesis, Stanford University.*

Widrow B., and Lehr, M.A., (1990) , “ 30 years of adaptive neural networks : Perceptron, madaline, and backpropagation, ” *Proceeding of IEEE*, **Vol 78**, pp 1415-1441.

Yabuta, T.and Yamada, T., (1992) , “ Neural Network Controller characteristics with regard to Adaptive Control ”, *IEEE Trans. on Systems, Man, and Cybernetics*, **Vol.22, No.1**, pp 170-177.

Yegerlehner, J.D. and Meckel, P.H. , (1993) “ Experimental Implementation of Neural Network Controller for robot undergoing Large Payload changes ”, *Proceeding of IEEE International Conference On Robotics and Automation*, **Vol.2**, pp 744-749.

Zalzala, A.M.S and Morris, A.S. , (1989) , “ Adaptive robot control using Artificial neural networks : Application in the theory of cognition ”, *Research Report # 374, Department Control Eng., University of Sheffield, UK.*

## *ANNEXES*

---

---

## Annexe A

# Concept de base

Un réseau de neurones est caractérisé par trois entités:

### Le Neurone

c'est l'unité de calcul qui reçoit des informations provenant des autres neurones, les traite puis envoie le résultat à d'autres unités identiques. Cette unité constitue le bloc principal du réseau. Comme montré en figure.1, le neurone est un dispositif multi-entrées/mono-sortie avec les paramètres suivants:

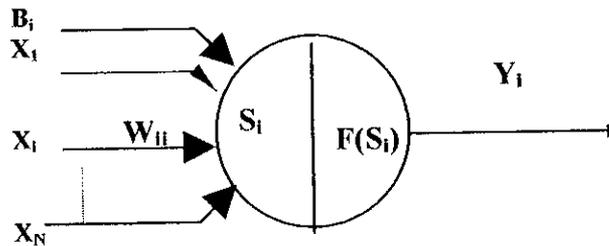


Figure A.1 Neurone Formel

- Les entrées du neurone: notées  $X_j$ ,  $j=1,2,\dots,N$ , lesquelles pourraient être des entrées externes ou des sorties d'autres neurones.
- poids de chaque entrée: notés  $W_{ij}$ , liant le neurone  $i$  avec d'autres neurones  $j=1,2,\dots,N$ . Ces poids sont adaptés durant l'apprentissage du réseau à fin d'améliorer les performances.
- Fonction combinaisante: relie tous les entrées du neurone, et peut être soit linéaire ou non-linéaire. Cette fonction n'est que la sommation des entrées pondérées

$$S_i = \sum_{j=1}^N W_{ij} X_j + B_i \quad (A.1)$$

• Fonction d'activation : cette fonction peut être aussi linéaire ou non linéaire et produit la sortie du neurone, laquelle peut être connectée à n'importe autre neurone du réseau, ou connectée comme entrée au même neurone (feedback). La fonction d'activation est typiquement choisie comme :

$$\left\{ \begin{array}{l} f(S_i) = \frac{1}{1 + \exp(-S_i)} \\ \text{ou} \\ f(S_i) = \frac{1 - \exp(-2S_i)}{1 + \exp(-2S_i)} \end{array} \right. \quad (\text{A.2})$$

Le traitement de l'information par le neurone s'effectue en deux étapes :

1<sup>ière</sup> étape : le neurone effectue une sommation pondérée de tous ces entrées.

2<sup>ième</sup> étape : pour calculer sa sortie, le neurone applique à cette somme la fonction d'activation.

### Topologie du réseau

Chaque neurone est connecté à d'autres (tous ou certains), d'où respectivement on obtient un réseau totalement ou partiellement connecté. Les neurones sont organisés en couches et les signaux se propagent en avant. Ainsi, on obtient un graphe orienté.

### Evolution du réseau

On entend par évolution du réseau la modification des liaisons inter-neurales par apprentissage cette évolution est définie par :

- l'architecture du réseau: nombre de couches, nombre de neurones dans chaque couche, et le graphe des connexions(topologie).
- L'algorithme d'apprentissage : au cours duquel le réseau apprend à accomplir une tâche à partir d'exemples. Cette algorithme sert à la mise à jour des poids synaptiques.

### Architecture du réseau

L'architecture du réseau est définie par :

- Le nombre de neurones,
- La nature des neurones (fonction d'activation),
- Le graphe d'interconnexion.

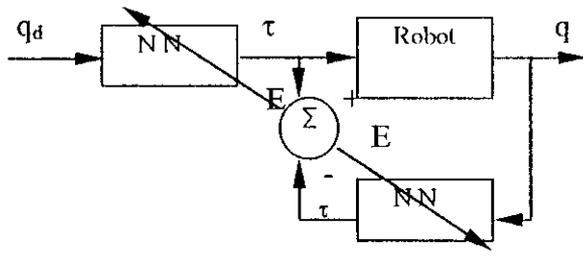
### Architecture d'apprentissage

Plusieurs architectures d'apprentissage furent proposées (Psaltis et al., 1987, Psaltis et al., 1988, Kawato et al., 1988). L'apprentissage indirect (Figure A.2(a)) vient du fait que faire tendre l'erreur d'entraînement  $E = \tau_d - \tau$  vers zéro, entraînera l'erreur total  $\epsilon = \theta_d - \theta$  vers zéro. Ce qui n'est pas toujours vrai.

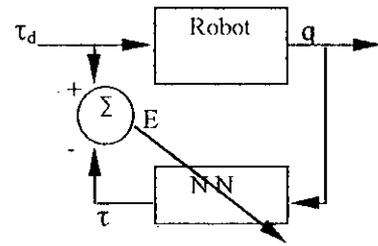
L'apprentissage généralisé montré en Figure A.2(b) peut être utilisé quand les couples d'entrées et les mouvements de réponse sont connus.

L'apprentissage spécialisé (Figure A.2(c)) utilise uniquement l'erreur de sortie du robot pour entraîner le système de commande. Mais ce type d'apprentissage exige une connaissance a priori sur la dynamique du robot.

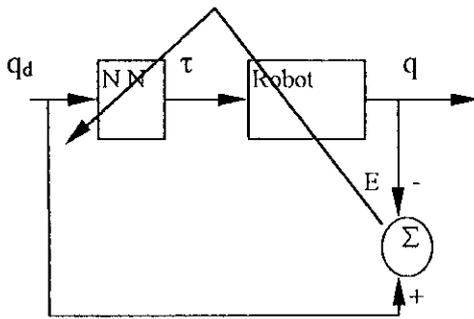
La dernière architecture (Figure A.2(d)) appelé apprentissage par erreur de retour; proposée per Kawato (1988); utilise l'erreur de retour directement sans passer à travers la dynamique du manipulateur pour entraîner le système de commande neural.



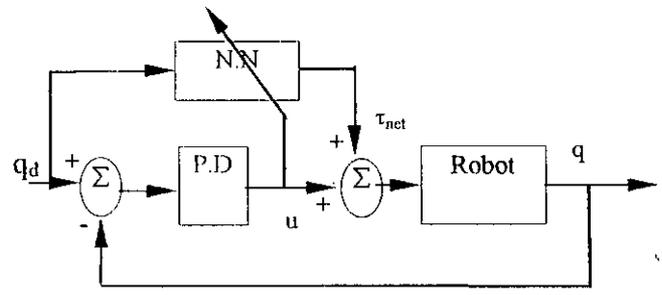
(a) Apprentissage indirecte



(b) Apprentissage généralisé



(c) Apprentissage Spécialisé.



(d) Apprentissage par erreur de retour

Figure A1.2 Architectures d'apprentissage

## Annexe B

### **Modélisation du robot RP41**

La structure générale du robot RP41 est illustrée en figure ci-dessous.

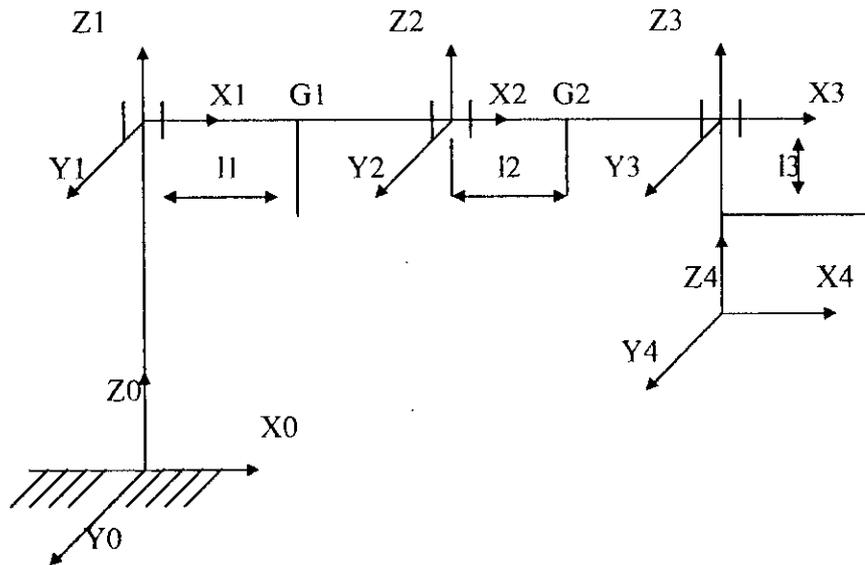


Figure B.1 Schéma Général de la structure du robot RP41.

Les paramètres de DENAVITT et HARTENBERG :

N	$\theta$	$\alpha$	d	a
1	$\theta_1$	0	$d_1$	0
2	$\theta_2$	0	0	$a_1$
3	$\theta_3$	0	0	$a_2$
4	0	0	$\pm T_4$	0

Etant donné que le RP41 est un robot de type SCARA ,alors l'énergie potentielle ( $U=0$ ), et ce car :

- il n'y a pas de déplacement par rapport à l'axe z .
- le couple exercé par la pesanteur est nul pour la même raison

L'énergie cinétique totale est la somme des énergies cinétiques de chaque bras du robot .

Calcul des vitesses de rotation (Coiffet, 1981) :

$$\Omega_0^0(R_0) = (0, 0, 0)^T, \tilde{\Omega}_1^0(R_0) = (0, 0, \dot{\theta}_1)^T, \tilde{\Omega}_2^1(R_1) = (0, 0, \dot{\theta}_2)^T, \Omega_3^2(R_2) = (0, 0, \dot{\theta}_2)^T$$

$$\Omega_1^0(R_1) = \tilde{\Omega}_1^0(R_0) + [M_0^1]^T \Omega_0^0(R_0) = (0, 0, \dot{\theta}_1)^T$$

$$\Omega_2^0(R_2) = \tilde{\Omega}_2^1(R_1) + [M_1^2]^T \Omega_1^0(R_1) = (0, 0, \dot{\theta}_1 + \dot{\theta}_2)^T$$

$$\Omega_3^0(R_3) = \tilde{\Omega}_3^2(R_2) + [M_2^3]^T \Omega_2^0(R_2) = (0, 0, \dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)^T$$

Calcul des vitesses de translation

$$V_{O_1}^{O_0}(R_1) = (0 \ 0 \ 0)^t, \quad V_{G_1}^{O_0}(R_1) = V_{O_1}^{O_0}(R_1) + \Omega_1^0(R_1) \wedge O_1 \vec{G}_1(R_1) = \begin{pmatrix} 0 \\ l_1 \dot{\theta}_1 \\ 0 \end{pmatrix}$$

$$V_{G_2}^{O_0}(R_2) = V_{O_2}^{O_0}(R_2) + \Omega_2^0(R_2) \wedge O_2 \vec{G}_2(R_2) = V_{O_2}^{O_0}(R_2)$$

$$V_{O_2}^{O_0}(R_2) = V_{O_2}^{O_0}(R_1) M_2^1 = \begin{pmatrix} L_1 \dot{\theta}_1 S \theta_2 \\ L_1 \dot{\theta}_1 C \theta_2 \\ 0 \end{pmatrix}$$

$$V_{G_2}^{O_0}(R_2) = \begin{pmatrix} L_1 \dot{\theta}_1 S \theta_2 \\ L_1 \dot{\theta}_1 C \theta_2 \\ 0 \end{pmatrix}$$

$$V_{G_3}^{O_0}(R_3) = V_{O_3}^{O_0}(R_3) + \Omega_3^0(R_3) \wedge O_3 \vec{G}_3(R_3) = V_{O_3}^{O_0}(R_3) \\ = V_{O_3}^{O_0}(R_2) M_3^2$$

avec :

$$V_{O_3}^{O_0}(R_2) = V_{O_2}^{O_0}(R_2) + \Omega_2^0(R_2) \wedge O_2 \vec{O}_3(R_2) = \begin{pmatrix} L_1 \dot{\theta}_1 S \theta_2 \\ L_1 \dot{\theta}_1 C \theta_2 + L_2(\dot{\theta}_1 + \dot{\theta}_2) \\ 0 \end{pmatrix}$$

$$V_{G_3}^{O_0}(R_3) = \begin{pmatrix} L_1 \dot{\theta}_1 S \theta_2 \\ L_1 \dot{\theta}_1 C \theta_2 + L_2(\dot{\theta}_1 + \dot{\theta}_2) \\ 0 \end{pmatrix} \begin{pmatrix} C \theta_3 & S \theta_3 & 0 \\ -S \theta_3 & C \theta_3 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$V_{G_3}^{O_0}(R_3) = \begin{pmatrix} L_1 \dot{\theta}_1 S \theta_2 C \theta_3 + [L_1 \dot{\theta}_1 C \theta_2 + L_2(\dot{\theta}_1 + \dot{\theta}_2)] S \theta_3 \\ -L_1 \dot{\theta}_1 S \theta_2 S \theta_3 + [L_1 \dot{\theta}_1 C \theta_2 + L_2(\dot{\theta}_1 + \dot{\theta}_2)] C \theta_3 \\ 0 \end{pmatrix}$$

$$T = \sum_{i=1}^3 T_i \quad \text{avec} \quad T_i = \frac{1}{2} m_i V_i^2 = \frac{1}{2} \left[ [m_i V_G^0(R_i)]^2 + [\Omega_i^0(R_i)] I^{(0)t} [\Omega_i^0(R_i)]^t \right]$$

$$T_1 = \frac{1}{2} m_1 V_{G_1}^0 (R_1)^2 + \frac{1}{2} \Omega_1^0 (R_1)' I^{(1)} \Omega_1^0 (R_1) = \frac{1}{2} (m_1 l_1^2 + I_1) \dot{\theta}_1^2$$

$$\begin{aligned} T_2 &= \frac{1}{2} m_2 V_{G_2}^0 (R_2)^2 + \frac{1}{2} \Omega_2^0 (R_2)' I^{(2)} \Omega_2^0 (R_2) \\ &= \frac{1}{2} \left[ m_2 \left[ (L_1 \dot{\theta}_1 S \theta_2)^2 + (L_1 \dot{\theta}_1 C \theta_2 + l_2 (\dot{\theta}_1 + \dot{\theta}_2))^2 \right] + (\dot{\theta}_1 + \dot{\theta}_2)^2 I_2 \right] \end{aligned}$$

$$\begin{aligned} T_3 &= \frac{1}{2} m_3 V_{G_3}^0 (R_3)^2 + \frac{1}{2} \Omega_3^0 (R_3)' I^{(3)} \Omega_3^0 (R_3) \\ &= \frac{1}{2} \left[ m_3 \left[ K_1^2 + K_2^2 \right] + (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)^2 I_3 \right] \end{aligned}$$

avec :

$$\begin{aligned} K_1 &= L_1 \dot{\theta}_1 S \theta_2 C \theta_3 + [L_1 \dot{\theta}_1 C \theta_2 + L_2 (\dot{\theta}_1 + \dot{\theta}_2)] S \theta_3 \\ K_2 &= -L_1 \dot{\theta}_1 S \theta_2 S \theta_3 + [L_1 \dot{\theta}_1 C \theta_2 + L_2 (\dot{\theta}_1 + \dot{\theta}_2)] C \theta_3 \end{aligned}$$

$$T = T_1 + T_2 + T_3$$

$$\begin{aligned} T &= \frac{1}{2} \left[ (I_1 + L_1^2 m_1) \dot{\theta}_1^2 + (L_1^2 m_2 (S \theta_2)^2 \dot{\theta}_1^2 + I_2 (\dot{\theta}_1 + \dot{\theta}_2)^2 + m_2 (l_2 \dot{\theta}_1 + L_1 C \theta_2 \dot{\theta}_1 + l_2 \dot{\theta}_2)^2 \right] + \\ &\quad \frac{1}{2} \left[ m_3 \left( (L_2 C \theta_3 \dot{\theta}_1 + L_1 C (\theta_2 + \theta_3) \dot{\theta}_1 + \theta_3 \dot{\theta}_1 + L_2 C \theta_3 \dot{\theta}_2)^2 + (L_2 S \theta_3 \dot{\theta}_1 + L_1 S (\theta_2 + \theta_3) \dot{\theta}_1 + L_2 S \theta_3 \dot{\theta}_2)^2 \right) \right] + \\ &\quad \frac{1}{2} \left[ I_3 (\dot{\theta}_1 + \dot{\theta}_2 + \dot{\theta}_3)^2 \right] \end{aligned}$$

A fin d'obtenir les expressions des couples exercés à chaque liaison nous avons fait recours au formalisme de Lagrange :

$$\tau_1 = \frac{d}{dt} \left( \frac{\partial T}{\partial \dot{\theta}_1} \right) - \frac{\partial T}{\partial \theta_1}, \quad \tau_2 = \frac{d}{dt} \left( \frac{\partial T}{\partial \dot{\theta}_2} \right) - \frac{\partial T}{\partial \theta_2}, \quad \tau_3 = \frac{d}{dt} \left( \frac{\partial T}{\partial \dot{\theta}_3} \right) - \frac{\partial T}{\partial \theta_3}$$

Pour effectuer les dérivées nous avons utilisé le Logiciel Mathematica sous la station Sun (SPARCstation LX). Cette dérivation nous a conduit aux expressions suivantes :

$$\begin{aligned} \tau_1 &= [I_1 + I_2 + I_3 + m_1 l_1^2 + m_2 (L_1^2 + l_2^2) + m_3 (L_1^2 + L_2^2) + 2L_1 (m_2 l_2 + m_3 L_2) \cos(\theta_2)] \ddot{\theta}_1 + \\ &\quad [I_2 + I_3 + m_2 l_2^2 + m_3 L_2^2 + L_1 (m_2 l_2 + m_3 L_2) \cos(\theta_2)] \ddot{\theta}_2 + I_3 \ddot{\theta}_3 - \\ &\quad [L_1 (m_2 l_2 + m_3 L_2) \sin(\theta_2) \dot{\theta}_2 - 2L_1 (m_2 l_2 + m_3 L_2) \sin(\theta_2) \dot{\theta}_1 \dot{\theta}_2] \end{aligned}$$

$$\tau_2 = [I_2 + I_3 + m_2 l_2^2 + m_3 L_2^2 + L_1(m_2 l_2 + m_3 L_2) \cos(\theta_2)] \ddot{\theta}_1 + [I_2 + I_3 + m_2 l_2^2 + m_3 L_2^2] \ddot{\theta}_2 + I_3 \ddot{\theta}_3 + [L_1(m_2 l_2^2 + m_3 L_2^2) \sin(\theta_2) \dot{\theta}_1^2]$$

$$\tau_3 = I_3 (\ddot{\theta}_1 + \ddot{\theta}_2 + \ddot{\theta}_3)$$

## Annexe C

# *Validation du modèle dynamique du robot RP41*

A fin de valider expérimentalement le modèle dynamique estimé, nous l'avons utilisé dans une commande anticipée + une correction proportionnelle et dérivée dans la boucle de retour. Le schéma de commande est montré en figure C.1

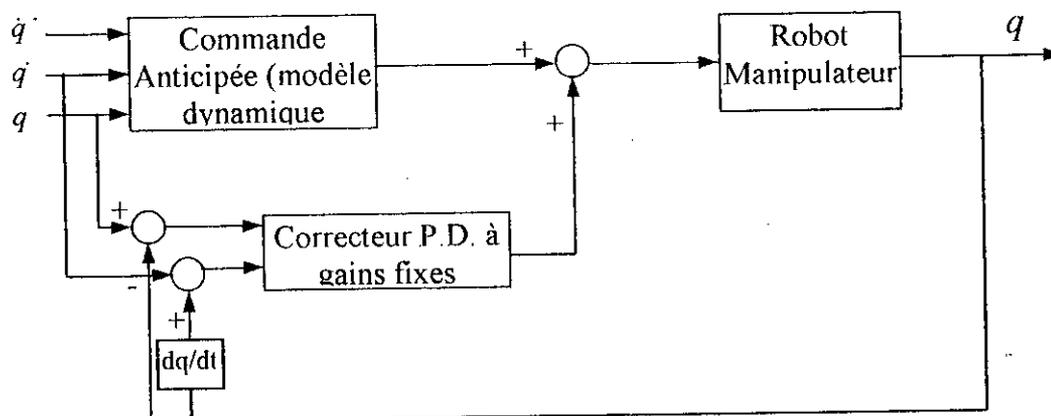


Figure C.1 Schéma de commande dynamique Anticipée du robot RP41.

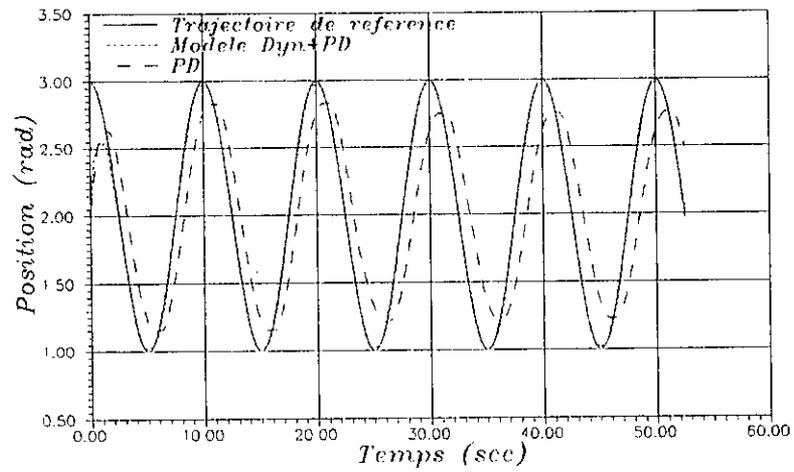
Pour cette étape de validation nous avons choisi un autre type de trajectoire donné par l'équation suivante :

pour la première et la deuxième articulation :  $\theta(t) = 2 + \cos(0.2\pi t)$

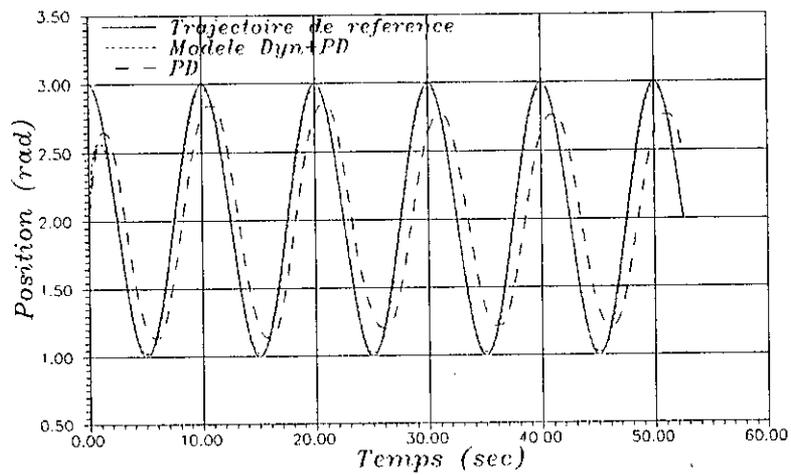
pour la troisième articulation :  $\theta(t) = 2 + \cos(0.5\pi t)$

Les gains de position  $K_p = \text{diag}[50, 50, 25]$  et de vitesse  $K_v = \text{diag}[2, 2, 0.06]$ .

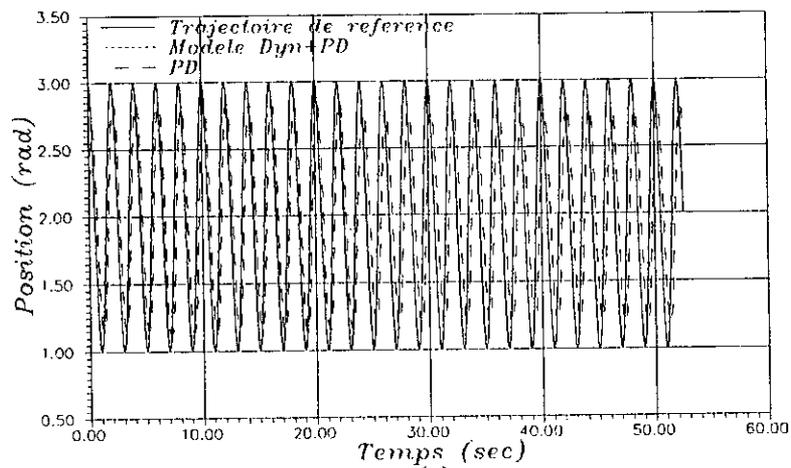
Pour mieux montrer l'apport du modèle dynamique, nous avons effectué une comparaison avec un correcteur PD à gains fixes dont les valeurs sont données précédemment. La figure C.2 montre les trajectoires de position, désirées et réelles, pour les trois liaisons. Nous constatons que les trajectoires réelles, dans le cas de la commande avec modèle, sont pratiquement confondues avec les trajectoires désirées. Par contre, pour le cas du correcteur PD seul, nous constatons une très mauvaise poursuite de la trajectoire de référence. Il est à noter qu'au départ nous remarquons que les trajectoires sont éloignées. Ceci est dû aux positions initiales des trajectoires désirées qui ne sont pas celles des positions initiales réelles des trois articulations du robot.



(a)



(b)



(c)

Figure C.2. Trajectoire de position des trois liaisons du RP41.

(a) : liaison 1. (b): liaison 2. (c): Liaison 3.