

N° d'ORDRE : 02/2008-E/EL

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université des Sciences et de la Technologie
Houari Boumediene
Faculté d'Electronique et d'Informatique



Thèse

Présentée pour l'obtention du diplôme de **DOCTORAT D'ETAT**

En : **ELECTRONIQUE**

Spécialité : Contrôle de Processus et Robotique

Par : Mr. Farid FERGUENE

Thème :

Contribution à la Commande en Effort d'un Bras Manipulateur

Soutenue publiquement le **02 Décembre 2008**, devant le jury composé de:

Mr M. ATTARI	Professeur à l'USTHB	Président
Mr R. TOUMI	Professeur à l'USTHB	Directeur de thèse
Mr F. BOUDJEMA	Professeur à L'ENP	Examineur
Mr M. HAMERLAIN	Directeur de Recherche au CDTA	Examineur
Mme N. Achour	Maître de conférence à l'USTHB	Examinatrice

REMERCIEMENTS

Avant tout, je veux exprimer ma gratitude envers Dieu qui m'a aidé à finaliser ce travail.

Les travaux présentés dans ce mémoire de Thèse ont été réalisés au Laboratoire de Robotique, de Parallélisme et d'Electroénergétique (LRPE) de la Faculté d'Electronique et d'Informatique de l'USTHB sous la direction de Monsieur le Professeur R. TOUMI.

Que Monsieur TOUMI Redouane, mon Directeur de Thèse, trouve ici l'expression de ma profonde gratitude pour tous les conseils avisés qu'il a su me communiquer, ainsi que pour le soutien et le dévouement dont il a fait preuve à mon égard et qui ont joué un rôle important dans l'aboutissement de mon travail de Thèse. Disponibilité, patience mais aussi sympathie et humour, autant de qualités qui le caractérisent et qui m'ont fait apprécier les moments passés en sa compagnie, au travail ou en dehors.

Je tiens à exprimer ma gratitude à Monsieur M. ATTARI Professeur à l'USTHB pour l'honneur qu'il me fait en présidant ce Jury.

J'adresse mes remerciements à Monsieur F. BOUDJEMA Professeur à l'ENP qui m'a fait le plaisir d'accepter de venir juger ce travail.

Mes remerciements s'adressent aussi à Monsieur M. HAMERLAIN Directeur de Recherche au CDTA pour la considération qu'il accorde à ce travail, en participant à ce jury.

J'adresse également mes remerciements à Madame N. ACHOUR Maître de Conférences à l'USTHB, qui, en participant au jury, a bien voulu marquer son intérêt pour ce travail.

Je remercie également tous les membres du Laboratoire LRPE pour leur encouragement, leur soutien et leur disponibilité.

Ma gratitude va enfin, vers tous ceux qui ont participé de près ou de loin à l'élaboration de ce travail. Je leur exprime ici toute ma reconnaissance.

Résumé

Notre travail illustre des apports notables de l'Intelligence Artificielle, discipline relativement récente, dans la synthèse de lois de commande d'un robot manipulateur exécutant une tâche dans un milieu contraint. L'intelligence artificielle, a permis en particulier de lever une contrainte en contrôle relativement à la nécessité de disposer des modèles les plus exacts possibles, sachant que les erreurs de modélisation tant du robot lui même que de l'environnement ont un impact sur la réponse dynamique d'un manipulateur en contact.

L'exploitation des réseaux de neurones et de la logique floue dans les structures de commande des processus a augmenté ces dernières années compte tenu de la capacité d'apprentissage des fonctions non linéaires des réseaux de neurones de leur adaptativité, et de la possibilité d'exploiter l'expérience et la connaissance humaine dans la commande grâce à la logique floue (élaboration de règles floues).

Notre travail débute par une étude bibliographique des différentes structures de commande conventionnelles en effort des robots manipulateurs. Nous avons analysé les avantages et les inconvénients de chacune de ces structures. Cette étude nous a permis de mettre en évidence deux structures de commande intéressantes, basées tous les deux sur une stratégie de dominance de la boucle de commande en force sur celle en position, permettant de contrôler à la fois la position et l'effort désirés, le long d'une même direction de l'espace de la tâche, sans avoir besoin d'utiliser un mécanisme de sélection.

Cependant les performances de ces deux structures de commande classiques, dont l'une est basée sur un concept parallèle et l'autre sur un concept hiérarchique, des deux boucles de commande en effort et en position, ont été illustrées uniquement dans le cas où la quasi-totalité de la dynamique du manipulateur est négligée (vitesses de déplacement considérées très faibles) ou bien lorsque le modèle dynamique du robot est supposé connu parfaitement. Ces cas sont particulièrement critiques lorsqu'on cherche à obtenir de bonnes performances en termes d'optimisation des réponses et d'adaptativité vis-à-vis de la variation de type de tâche et d'environnement des milieux contraints. Il apparaît donc difficile dans ces conditions de concevoir des commandes performantes des systèmes inconnus ou connus partiellement. Notre intérêt s'est porté sur ces deux structures de commande classiques pour améliorer leurs performances en leurs associant des contrôleurs basés sur des concepts issus de l'intelligence artificielle (les réseaux de neurones et la logique floue).

Deux stratégies de commande neuronale force/position ont été synthétisées, pour compenser l'effet des incertitudes sur la dynamique du manipulateur en contact sur les performances des réponses en force et en position. La première stratégie, basée sur une technique de modification du couple de commande, exploite le modèle dynamique nominal du manipulateur dans l'élaboration de la commande. Quant à la deuxième stratégie, basée sur une technique de modification de la trajectoire de référence, exploite des simples contrôleurs PID conventionnels. Une étude de la stabilité, en considérant le cas idéal (compensation parfaite de l'effet des incertitudes), de ces deux stratégies de commande neuronales, en présence d'un environnement supposé élastique, nous a permis de montrer une performance très importante relative à la deuxième stratégie de commande, donnant la condition de stabilité du système en boucle fermée indépendamment d'une connaissance préalable de la raideur de l'environnement. Pour améliorer d'avantage les réponses en effort obtenues (sensibles aux variations de la raideur de l'environnement en régimes transitoires), deux solutions différentes ont été proposées pour la boucle de commande en effort de ces deux structures de commande force/position. La première solution introduit un contrôleur flou adaptatif à la place d'un contrôleur conventionnel. Quant à la deuxième solution, elle introduit un bloc de commande adaptatif basé sur une méthode d'identification en ligne de la raideur de l'environnement.

Les performances des différents schémas de commande proposés sont illustrées à l'aide de nombreux résultats de simulation en utilisant les modèles du robot PUMA 560 pour les trois premières articulations face à diverses géométries de l'environnement et en présence d'incertitudes sur sa raideur.

Mots-clés: *commande force/position, structure parallèle, structure externe, commande neuronale, réseaux de neurones, logique floue, contrôleur PID, robot.*

TABLE DES MATIERES

Introduction générale.....1

Chapitre 1 : Les structures de commande en effort des robots manipulateurs

1.1 Introduction.....4

1.2 Commande par raideur active.....5

1.3 Commande par retour d'effort explicite.....7

 1.3.1 Commande avec retour par matrice de raideur.....7

 1.3.2 Commande avec retour par matrice d'amortissement8

1.4 Commande en impédance.....8

1.5 Commande hybride force/position.....13

 1.5.1 Structure hybride de CRAIG et RAIBERT.....14

 1.5.2 Structure hybride de KHATIB.....15

1.6 Structure de commande parallèle force/position.....16

1.7 Structure de commande en effort externe.....18

1.8 Conclusion.....19

Chapitre 2 : Structures de commande neuronales et floues

2.1 Introduction.....21

2.2 Les réseaux de neurones.....21

 2.2.1 Introduction.....21

 2.2.2 Généralités sur les réseaux de neurones.....21

 2.2.2.1 Structure d'un réseau de neurones.....21

 2.2.2.2 Les réseaux de neurones statiques.....23

 2.2.2.3 Apprentissage des réseaux de neurones artificiels.....23

 2.2.2.4 Algorithme de rétropropagation.....25

 2.2.2.5 Différentes étapes de calcul d'un réseau de neurones.....25

 2.2.3 Les réseaux de neurones et la commande de processus.....27

 2.2.3.1 Réseaux de neurones appliqués à l'identification.....27

 2.2.3.2 Le contrôleur neuronal.....28

2.2.3.3 Le contrôleur neuronal inverse direct.....	28
2.2.3.4 Commande avec modèle de référence	30
2.2.3.5 Commande à modèle interne (IMC).....	30
2.2.3.6 Commande adaptative à modèle de référence.....	31
2.2.3.7 Contrôleur neuronal inverse avec une boucle de commande conventionnelle.....	33
2.2.3.8 Technique basée sur la modification de la trajectoire de référence.....	34
2.2.3.9 Commande neuronale d'un robot basée sur son modèle dynamique.....	34
2.2.3.9.1 Equation dynamique du robot.....	35
2.2.3.9.2 La loi de commande dynamique.....	35
2.2.3.9.3 Technique de compensation adaptative d'OZAKI.....	36
2.2.3.9.4 Technique de compensation d'ISHIGURO.....	37
2.2.3.9.5 Commande dynamique avec réseau de neurones auxiliaire.....	38
2.3 La logique floue.....	39
2.3.1 Introduction.....	39
2.3.2 Ensemble flou.....	39
2.3.3 Opération sur les sous ensembles flous.....	40
2.3.4 Raisonnement linguistique par des règles floues.....	41
2.3.5 Variable linguistique.....	42
2.3.6 Structure générale d'un régulateur flou.....	42
2.3.7 Description des éléments d'un régulateur flou.....	42
2.3.8 Modèles flous (système d'inférence flou).....	43
2.3.9 La logique floue et les contrôleurs PID.....	46
2.3.9.1 Relation entre un contrôleur PI et un contrôleur PD.....	49
2.3.9.2 Contrôleur PI flou avec autoréglage de son paramètre de sortie.....	49
2.3.9.3 Contrôleur PI conventionnel adaptatif par logique floue.....	50
2.4 Conclusion	51

Chapitre 3 : Structures de commande intelligentes parallèle force/position

3.1 Introduction.....	52
3.2 Equation dynamique du robot en contact avec son environnement.....	52
3.3 Structure de commande dynamique parallèle force/position.....	53
3.3.1 Analyse des performances en présence d'un milieu élastique.....	55
3.3.1.1 Cas d'un environnement de position mal connue.....	58

3.3.1.2 cas d'un environnement curviligne.....	58
3.3.2 Problématique.....	59
3.4 Structure neuronale de la commande dynamique parallèle force/position.....	61
3.5 Implémentation.....	65
3.6 Structure parallèle avec un contrôleur flou pour sa boucle de commande en effort.....	79
3.6.1 Description du contrôleur flou	79
3.6.1.1 Gains de normalisation	82
3.6.1.2 Autoréglage des gains de dénormalisation.....	82
3.6.1.3 Base des règles de α	83
3.6.2 Résultats de simulation.....	84
3.7 Structures neuronales parallèles force/position non basées sur le modèle.....	86
3.7.1 Structure de commande PID conventionnelle parallèle force/position.....	87
3.7.2 Structure neuronale parallèle basée sur une technique de modification du couple de commande.....	88
3.7.3 Structure neuronale parallèle basée sur une technique de modification de la trajectoire de référence.....	89
3.7.4 Architecture du réseau de neurones.....	91
3.7.5 Analyse des performances en présence d'un milieu élastique.....	92
3.7.5.1 Cas d'un environnement de position mal connue.....	94
3.7.6.2 Cas d'un environnement curviligne.....	94
3.7.6 Résultats de simulation.....	94
3.8 Structure neuronale parallèle avec boucle de commande en effort adaptative.....	99
3.9 Conclusion.....	103
 Chapitre 4 : Structures intelligentes de la commande en effort externe	
4.1 Introduction.....	105
4.2 Structure dynamique de la commande en effort externe.....	105
4.2.1 Analyse des performances en présence d'un milieu élastique.....	108
4.2.1.1 Cas d'un environnement de position mal connue.....	109
4.2.1.2 Cas d'un environnement curviligne.....	110
4.3 Structure neuronale de la commande dynamique en effort externe.....	110
4.4 Résultats de simulation.....	112
4.5 Structure neuronale avec une boucle de commande floue en effort	121
4.6 Structure neuronale de la commande en effort externe non basée sur le modèle.....	125
4.6.1 Structure PID de la commande en effort externe.....	125

4.6.2 Approche neuronale basée sur une technique de modification de la trajectoire de référence.....	126
4.6.3. Analyse des performances en présence d'un milieu élastique.....	129
4.6.3.1 Cas d'un environnement de position mal connue.....	129
4.6.3.2 cas d'un environnement curviligne.....	130
4.6.4 Résultats de simulation.....	130
4.7 Structure neuronale de la commande en effort externe avec boucle de commande en effort adaptative.....	133
4.8 Structure neuronale de la commande en effort externe avec boucle de commande en effort flou.....	135
4.9 Conclusion.....	136
Conclusion générale.....	137
Références bibliographiques.....	140
Annexe A.....	144
Annexe B.....	149

Introduction générale

L'industrie moderne exige certaines tâches demandant une grande précision et répétées inlassablement qui ne peuvent être réalisées par des êtres humains . De ce besoin sont nés des robots manipulateurs relativement simples et souvent destinés à une tâche bien précise.

Les premiers robots introduits dans l'industrie aux environs des années 80 réalisaient des tâches de soudure, de peinture, et de manutention. Ils étaient commandés en position ou en vitesse le plus souvent par apprentissage. L'effecteur en décrivant sa trajectoire n'entrait pratiquement jamais en contact direct avec son environnement.

Actuellement, les tâches en robotique conduisent de plus en plus à des interactions du robot avec son environnement. Des exemples typiques de ces tâches sont l'assemblage et l'usinage de pièces , les tâches exécutées par les robots dans le domaine du bâtiment, la robotique médicale avec le geste médical assisté (GMAO) entre autres . En effet lorsque le robot est en contact avec son environnement l'organe terminal du robot manipulateur ne peut pas se déplacer librement dans toutes les directions. Le mouvement résultant est un mouvement contraint, et la chaîne cinématique du manipulateur est fermée sur l'environnement. Du fait de la raideur de l'ensemble, la faible variation du point terminal peut induire des efforts élevés qui, s'ils ne sont pas pris en compte par le système de contrôle, peuvent provoquer des rebonds et la détérioration de l'effecteur et/ou de l'environnement.

L'exécution d'une tâche en milieu contraint peut être effectuée en utilisant un système de contrôle qui régule les efforts engendrés par le contact de l'effecteur avec son environnement grâce aux capteurs d'effort placés au niveau de ce dernier.

Il existe dans la littérature deux principales méthodes pour commander en effort un robot manipulateur: la commande en impédance et la commande hybride force/position. La commande en impédance [HOG 85] permet d'ajuster la position de l'effecteur en réponse aux forces de contact ressenties grâce aux réglages des paramètres reliant les forces d'interaction à la variation de la position de l'effecteur. Dans la commande hybride, stratégie qui est largement adoptée dans la commande force/position des robots manipulateurs [RAI 81, KHA 87], l'idée de base réside dans la possibilité de choisir à commander l'effort ou la position le long de chaque direction de l'espace en utilisant des matrices de sélection. Deux autres approches, conceptuellement différentes de la commande hybride force/position des robots manipulateurs, sont aussi proposées dans la littérature. La première approche, est basée sur une stratégie de commande parallèle [CHI 92, 93,94]. Quant à la deuxième, elle est basée sur une stratégie de commande hiérarchique [SHU 88] [PER 91]. Par opposition à la stratégie de commande hybride, les deux variables force et position sont utilisées le long d'une même direction de l'espace de la tâche sans aucun mécanisme de sélection. L'efficacité de chacune de ces deux structures de commande est assurée grâce à la dominance de la boucle de commande en force sur la boucle de commande en position le long de la direction contrainte de la tâche, lorsque des interactions se produisent.

Cependant les performances de ces deux structures de commande classiques ont été illustrées uniquement dans le cas où la quasi-totalité de la dynamique du manipulateur est considérée négligeable [PER 91][CHI 94] ou bien lorsque le modèle dynamique du robot, exploité dans la commande, est supposé parfaitement connu [CHI 93]. Dans la pratique, le modèle dynamique exact du manipulateur n'est jamais connu, et la raideur de l'environnement n'est qu'approximativement connue. Plusieurs approches ont été proposées dans la littérature pour résoudre le problème du suivi de trajectoire en milieux contraints lorsqu'on est en présence des incertitudes sur les modèles du robot manipulateur et

l'environnement. On peut citer, entre autre, la commande à structure variable (VSC)[YAO 92], commande adaptative [WIH 97] [SIN 95] [CHI 04], commande robuste parallèle [HUA 07]. Bien que ces techniques proposées, améliorent les performances, elles tendent par contre à augmenter la complexité du système dynamique, ce qui nécessite une attention particulière quant à la stabilité du système global.

Par ailleurs l'utilisation des réseaux de neurones et de la logique floue dans les structures de commande des robots manipulateurs s'est développée ces dernières années, vu la capacité d'apprentissage des fonctions non linéaires des réseaux de neurones due à leur adaptativité [HAY 99] et à la possibilité d'exploiter l'expérience et la connaissance humaine dans la commande grâce à la logique floue (élaboration des règles floues) [YIN 00]. L'avantage de ces concepts, issus de l'intelligence artificielle, est qu'ils ne requièrent pas de connaissances théoriques détaillées en Mathématiques et en commande, ce qui permet de réduire ainsi le coût des contrôleurs des systèmes complexes, en particulier les systèmes non linéaires. Ainsi, des structures de commande neuronale force/impédance [JUN 95a, 98, 00][BAP 99][MAR 97] et des structures de commande neuro-floues force/position [KIG 95, 97] ont été proposées. Néanmoins, les résultats obtenus dans ce sens présentent des limites soit en termes de performances soit vis-à-vis de certains types de tâches effectuées par le robot (utilisation des matrices de sélection).

Le travail présenté dans cette thèse rentre dans le contexte des commandes en effort des robots manipulateurs. En effet, nous nous sommes intéressés aux cas des tâches exécutées par des robots manipulateurs en milieux contraints faisant intervenir l'effort et la position désirée lorsqu'on est en présence d'incertitudes sur la dynamique du robot manipulateur et d'incertitudes sur le modèle de l'environnement.

L'objectif principal de nos travaux est l'introduction des techniques de l'automatique moderne pour la synthèse des lois de commande force/position des robots manipulateurs. Ainsi nous proposons de nouvelles approches, associées à des structures de commande force/position conventionnelles de la littérature [CHI 93, 94] [SHU 88], en utilisant la théorie des réseaux de neurones artificiels et la logique floue comme outils de commande.

L'ensemble de nos travaux est divisé en quatre chapitres:

Dans le premier chapitre, nous présentons les différentes structures de commande en effort des robots manipulateurs qu'on retrouve dans la littérature. Nous montrons les avantages et les inconvénients de chacune de ces structures ainsi que les améliorations apportées à certaines de ces structures de commande conventionnelles. Nous accordons plus d'attention dans ce chapitre aux structures de commande jugées intéressantes par leurs caractéristiques particulières, et que nous avons exploitées par la suite dans notre travail.

Le deuxième chapitre est consacré à la présentation succincte des notions théoriques nécessaires pour la compréhension des conceptions qui suivent, ainsi que la présentation de certaines structures de commande, basées sur ces notions. Nous commençons d'abord par une description des réseaux de neurones artificiels ainsi que leur utilisation en commande de processus à travers une présentation des structures de commande les plus citées dans la littérature. Nous donnons, en suite, des notions de base sur la logique floue et son application dans le domaine de la commande. La relation existante entre la logique floue et les contrôleurs PID classiques sera démontrée, et nous terminerons ce chapitre par une

présentation de quelques schémas de commande floue proposés dans la littérature, que nous avons jugés intéressants pour notre cas.

L'objet du troisième chapitre est la commande intelligente parallèle force/position des robots manipulateurs. Nous commençons par présenter la structure dynamique conventionnelle proposée par Chiaverini [CHI 93] tout en montrant ses caractéristiques et ses limites à travers une étude de la problématique.

En se basant sur cette structure de commande conventionnelle, nous proposons une nouvelle structure de commande neuronale parallèle force/position. Notre méthode constitue une extension au cas de la commande dynamique parallèle force/position d'une méthode déjà proposée pour le cas d'une commande en position [JUN 95b]. Dans le développement de la loi de commande neuronale auxiliaire, nous utilisons la propriété d'approximation universelle des fonctions non linéaires des réseaux de neurones artificiels à couches. L'adaptation des poids du réseau proposé est réalisée en ligne grâce à l'algorithme de rétro propagation entraîné par un vecteur signal issu de l'équation du système en boucle fermée. Pour améliorer les performances de la réponse en force, un contrôleur flou est implémenté dans la boucle de commande en effort de cette structure parallèle. Afin de tester la validité des schémas de commande proposés, des simulations sont effectuées sur le modèle du robot PUMA 560 en présence d'un environnement supposé élastique.

Une autre stratégie de commande neuronale qui ne fait pas appel, cette fois ci, au modèle dynamique du robot dans l'élaboration de la commande, est aussi proposée dans le reste de ce chapitre. Le réseau de neurones à couche que nous proposons d'implémenter au niveau de la trajectoire de référence joue le rôle d'un compensateur de la dynamique du robot manipulateur en contact. La stabilité de cette nouvelle approche de commande, en considérant le cas idéal, sera démontrée.

Pour faire face aux effets des incertitudes sur la raideur de l'environnement, la boucle de commande en effort de cette structure de commande est rendue adaptative grâce à une technique d'estimation en ligne de la valeur de cette raideur. Des résultats de simulation montrant l'efficacité de ces méthodes seront présentés et comparés.

Le dernier chapitre est une extension au cas de la structure de commande en effort externe conventionnelle, des approches intelligentes utilisées dans le cas des structures parallèles présentées dans le chapitre 3. Ainsi de nouvelles structures de commande intelligentes, basée et non basée sur le modèle dynamique du robot, seront élaborées. Des résultats de simulations validant et justifiant notre démarche seront présentés, interprétés et comparés à ceux obtenus avec les structures parallèles.

Enfin, nous terminons notre travail par une conclusion générale et quelques perspectives.

Chapitre 1

**Les structures de commande en effort
des robots manipulateurs**

1.1 Introduction:

La commande en effort consiste à prendre en compte les interactions de contact entre un robot et son environnement lors de l'exécution d'une tâche donnée. Ce contrôle des interactions permet de pallier aux difficultés inhérentes à la commande en position qui donne en réalité un comportement aveugle au robot manipulateur vis-à-vis de son environnement extérieur. Ainsi, pour des tâches nécessitant un contact robot environnement une commande en position peut s'avérer totalement inappropriée et source d'échec et d'instabilité.

Pour une réalisation correcte de ses tâches, le robot doit donc s'adapter en permanence aux efforts de contact qu'il exerce sur son environnement. La réalisation correcte d'une tâche d'assemblage par exemple en mode automatique nécessite une très bonne précision et une bonne répétabilité, ce qui est difficile à réaliser. Toutefois, en pratique il est nécessaire de prendre en compte les actions de contact, c'est-à-dire de donner au manipulateur la capacité de réagir aux forces de contact par des mouvements. Le système mécanique doit être compliant. Cette complianc e ou flexibilité permet de réduire les forces de contact entre les pièces pendant la phase d'insertion. Cela permet de compenser les défauts d'alignement et les défauts angulaires dus aux incertitudes de positionnement et d'orientation de l'organe terminal du robot par rapport à l'alésage de la pièce cible.

La robotisation de l'opération d'usinage de pièces nécessite aussi un contact permanent entre l'outil attaché à l'organe terminal du robot manipulateur et la pièce à usiner, avec un suivi de contact réel qui pourra être connu ou inconnu .

Les exemples cités ci-dessus montrent que les différentes tâches mettant en contact le robot avec son environnement nécessitent une certaine souplesse pour ne pas endommager les pièces à assembler ou à usiner. Cette souplesse ou comportement flexible du robot face à son environnement est désignée sous le terme de complianc e.

La complianc e peut être définie comme étant la capacité d'un manipulateur à avoir un comportement souple. Le robot compliant peut subir un déplacement sous l'effet d'un effort dû au contact du manipulateur avec son environnement. Dans la littérature on trouve deux types de définitions de la complianc e qui sont la complianc e passive, et la complianc e active.

La complianc e passive repose sur l'utilisation d'une structure déformable sur laquelle on vient appliquer les efforts de contact. Dans le cas d'une tâche d'assemblage la variation de sa géométrie va corriger les erreurs de positionnement de l'organe terminal du robot grâce à cette structure déformable attachée à ce dernier. Le principal inconvénient de ce type de complianc e réside dans leur dépendance vis-à-vis de la tâche à accomplir et de la pièce à insérer ce qui limite leur utilisation systématique et constitue un frein à leur généralisation.

La complianc e active permet d'adapter le comportement du robot pour différentes classes de tâches. Le principe consiste à introduire la notion de complianc e au niveau du système de commande. Cette solution offre la possibilité de définir la position du centre de complianc e ainsi que la forme de complianc e en fonction de la tâche souhaitée.

Un contact en un point de dispositif engendre un couple (effort/déplacement) en un autre point du dispositif. Soit F , le torseur des forces et moment appliqués à ce point et D le torseur des déplacements linéaire et angulaire associés. L'équation du mouvement d'un solide soumis à une force de frottement visqueux et à une force élastique de rappel est donnée par :

$$F = K(D).D + B(D).\frac{dD}{dt} + I(D).\frac{d^2D}{dt^2} \quad (1.1)$$

Les matrices d'inertie $I(D)$, de frottement $B(D)$ et de raideur $K(D)$ sont des fonctions du déplacement D . Elles dépendent aussi du point considéré. Ceci amène donc à établir un classement selon les différentes natures de la compliance.

Lorsque les forces appliquées à l'objet sont de nature élastique, nous avons une compliance élastique. Dans le cas où les efforts appliqués sont de type frottement visqueux, la compliance sera anélastique amortie. Quand les forces agissant sur le solide sont dues aux masses en mouvement nous avons une compliance anélastique inertielle. Ces trois types de compliance sont le plus souvent couplés mais très fréquemment, la compliance élastique est prédominante [PRE 97].

Dans ce cas nous pouvons donc écrire:

$$F = K(D)D \quad (1.2)$$

C'est cette expression que nous allons utiliser dans notre thèse pour modéliser les forces de contact robot/environnement.

Au cours de ce chapitre, nous décrirons les différentes approches de commande en effort d'un robot manipulateur qu'on trouve dans la littérature, qui utilisent la notion de compliance active définie précédemment et qui est basée sur la mise en œuvre d'une loi de commande logicielle. Pour chacune, nous dégagerons les avantages et les inconvénients respectifs et les améliorations proposées dans la littérature pour certaines de ces structures. Nous justifierons notre choix pour les structures de commandes classiques que nous avons retenues afin de concevoir des structures intelligentes, grâce aux apports des réseaux de neurones et de la logique floue.

1.2 Commande par raideur active

SALISBURY [SAL 80] propose de contrôler activement la raideur apparente d'un manipulateur. Cette raideur peut être modifiée pour répondre aux besoins de diverses tâches à réaliser. Il est aussi possible de contrôler le comportement compliant du manipulateur. Cette raideur est définie dans l'espace cartésien par rapport à un repère de compliance situé au niveau de l'organe terminal du manipulateur et dont le centre est fixé en fonction de la tâche à accomplir. Cette méthode définit en statique une fonction linéaire qui lie les forces d'interaction à la position finale du manipulateur, via une matrice de raideur en coordonnées cartésiennes.

$$F = K_c \Delta P \quad (1.3)$$

Avec:

F : Effort de contact

ΔP : Déplacement différentiel

K_c : Matrice de raideur souhaitée

Dans la commande envisagée, les consignes sont les position et vitesse dans l'espace articulaires, soient q_d et \dot{q}_d . L'espace de commande est celui des coordonnées articulaires. Il

est donc nécessaire de définir la matrice de raideur dans l'espace articulaire correspondant à la matrice de raideur K_c définie dans l'espace cartésien. Son expression s'obtient en considérant les deux équations suivantes:

$$\Gamma = J^T . F \tag{1.4}$$

$$\Delta P = J . \Delta q \tag{1.5}$$

L'équation (1.5) correspondant au modèle cinématique direct d'un robot avec J matrice Jacobienne du manipulateur et Δq les variations élémentaires des coordonnées articulaires q . La combinaison des équations (1.3) (1.4) et (1.5) nous donne la relation suivante dans l'espace articulaire.

$$\Gamma = J^T . \underbrace{K_c . J}_{K_q} . \Delta q \tag{1.6}$$

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

La matrice de raideur K_q est symétrique mais non diagonale cela signifie qu'il existe un fort couplage entre les articulations. Une erreur de position sur une articulation affecte le couple appliqué aux actionneurs des autres articulations. Dans le cas où l'on désire introduire une consigne d'effort, il est possible d'ajouter un terme provenant d'une boucle en effort (terme proportionnel et prédictif).

Dans ce cas la loi de commande globale proposée est la suivante:

$$\Gamma = K_q(q_d - q) + K_v(\dot{q}_d - \dot{q}) + J^T (F_d + K_f(F_d - F_e)) \tag{1.7}$$

Cette loi de commande est représentée par le schéma suivant:

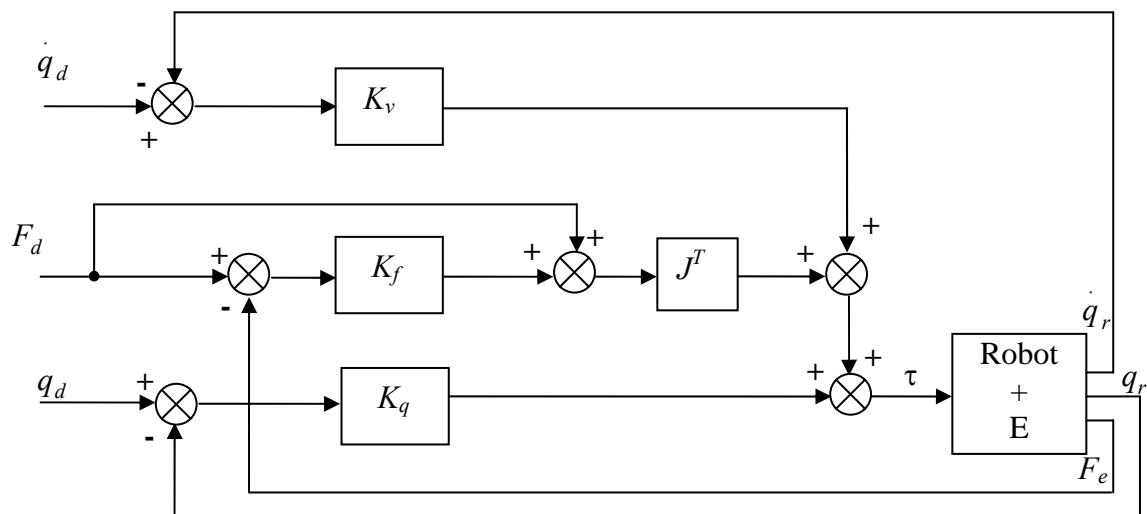


Figure 1.1 Commande par raideur active

Cependant cette commande présente certains inconvénients dus à l'utilisation de la matrice Jacobienne J . En effet, cette méthode ne fonctionne pas lorsque le robot se trouve proche d'une singularité. Le degré de liberté perdu à cause de la singularité ne pourra être commandé en effort, certains termes de K_q devenant nuls. Ainsi les couples correspondants appliqués aux moteurs sont nuls tant que le robot ne s'écarte pas suffisamment de sa position singulière.

1.3 Commande par retour d'effort explicite [WHI 85]

Avec cette commande, le robot est commandé en position ou en vitesse. Et grâce à la mesure des efforts, la consigne initiale se trouve modifiée pour corriger le signal d'entrée. On distingue deux cas possibles selon le type de la boucle de retour:

1.3.1 Commande avec retour par matrice de raideur

Pour ce type de commande le robot est considéré comme un ressort. La mesure des efforts de contact F_e fournit un déplacement ΔP dans l'espace cartésien, en considérant une relation de proportionnalité de la forme:

$$\Delta P = C_f F_e \quad (1.8)$$

Où C_f est la matrice diagonale de retour d'effort qui correspond à l'inverse de la matrice de raideur du robot désiré. Le déplacement ΔP issu de la prise en compte des efforts de contact, vient se soustraire de la consigne de position désirée P_d .

En fait, le robot est piloté en position, il suit sa trajectoire de consigne et dès qu'un effort extérieur apparaît, du au contact, avec son environnement, l'effort ressenti est alors détecté. Cette information en effort est ensuite utilisée pour modifier la commande en position qui vise à réduire les efforts de contact. Par conséquent, l'utilisation d'un gain faible dans une direction donnée contraint le robot à être commandé en position dans cette direction, par contre l'utilisation d'un gain élevé permet de le commander en effort, de manière à pouvoir augmenter la compliance. On représente le principe de fonctionnement de cette méthode par la figure suivante:

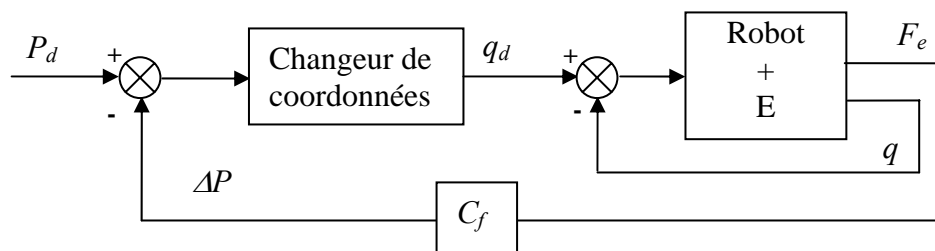


Figure 1.2 Commande par matrice de raideur

Le choix des gains de la matrice C_f joue un rôle primordial dans la stabilité de cette structure de commande, une valeur désirée arbitraire pouvant conduire à un comportement instable du robot.

1.3.2 Commande avec retour par matrice d'amortissement

Dans ce schéma de commande (voir figure 1.3), le robot est considéré comme un amortisseur. La mesure des efforts modifie la consigne désirée de vitesse \dot{X}_d via une matrice diagonale K_f qui correspond à l'inverse de la matrice d'amortissement désirée. Un intégrateur placé dans la chaîne directe de l'asservissement transforme la valeur modifiée en une consigne de position X_c à envoyer au système.

La relation linéaire liant les efforts à la vitesse est donnée par:

$$\Delta \dot{X} = K_f \cdot F_e \quad (1.9)$$

Puisque l'effort mesuré agit directement sur la vitesse de déplacement du robot. Ce dernier se comporte bien comme un amortisseur.

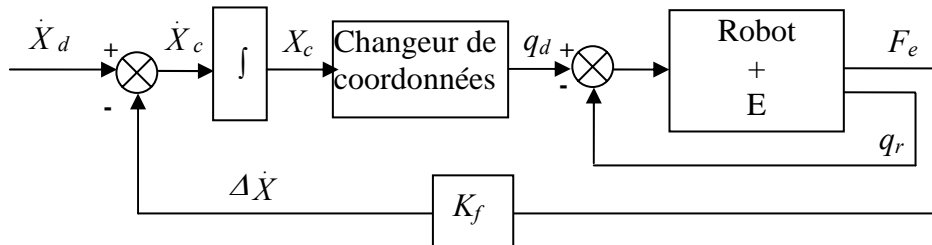


Figure 1.3 : Commande par matrice d'amortissement

La commande par retour d'effort par ses deux versions qu'elle soit par matrice de raideur ou matrice d'amortissement a deux avantages : le contrôle s'effectue dans l'espace cartésien (espace dans lequel est décrite la tâche), et sa simplicité de mise en œuvre avec peu de calculs. Cependant son principal inconvénient c'est qu'elle ne permet pas de façon explicite d'imposer un effort désiré. Son action se limite à minimiser l'effort de contact. L'une des applications possibles avec ce type de commande est l'opération de nettoyage des vitres.

1.4 Commande en impédance

Ce type de commande a été introduit par HOGAN [HOG 85]. Elle repose sur le principe que toute boucle d'asservissement appliqué à un système physique modifie son comportement dynamique propre. Cette notion reste valable dans le cas d'un robot en interaction avec son environnement. Quand le robot est en contact avec son environnement, il y a interaction entre eux. Il doit donc y avoir compatibilité physique entre eux.

L'idée de base de la commande en impédance repose sur le fait que si le robot se comporte comme une impédance mécanique, l'environnement se comporte comme une admittance mécanique de sorte que le robot puisse faire face aux contraintes en position imposée par l'environnement. La tâche doit être décrite non pas en terme de déplacement ou de force mais en terme de relation entre les deux, le terme reliant les deux grandeurs correspondant à une impédance sur laquelle on vient agir par le biais de la commande.

HOGAN a formalisé la notion d'impédance en définissant comme fonction de transfert $Z(p)$:

$$F(p) = Z(p) \cdot \delta X(p) \quad (1.10)$$

où,

$\delta X(p)$, est un déplacement opérationnel à partir d'une position nominale X_0 et $F(p)$ la force produite en réponse au mouvement $\delta X(p)$. Le manipulateur doit donc se comporter comme une impédance $Z(p)$ variable en fonction des différentes phases de la tâche. Généralement c'est une fonction de second ordre définie comme suit:

$$Z(p) = Mp^2 + Bp + K \quad (1.11)$$

Avec :

K : matrice de raideur

B : matrice d'amortissement

M : matrice inertielle.

Les valeurs de ces paramètres sont choisies en fonction du comportement souhaité [PER 91]:

M permet d'assouplir la réponse du système vis-à-vis du contact externe, des valeurs importantes sont données à B quand il est nécessaire de dissiper de l'énergie cinétique et donc d'amortir la réponse et K la raideur représente la précision avec laquelle on peut espérer satisfaire la commande en position de l'extrémité.

Différentes techniques ont été élaborées pour mettre en œuvre la commande en impédance dont on peut citer:

- La technique basée sur le contrôle du couple: elle utilise une mesure de la position et en déduit l'effort qu'il faut exercer sur l'organe terminal compte tenu de l'impédance Z souhaitée (figure 1.4).
- la technique basée sur la position: elle consiste à mesurer l'effort d'interaction, cette mesure est utilisée pour modifier la position de consigne compte tenu de l'impédance Z souhaitée (figure 1.5).

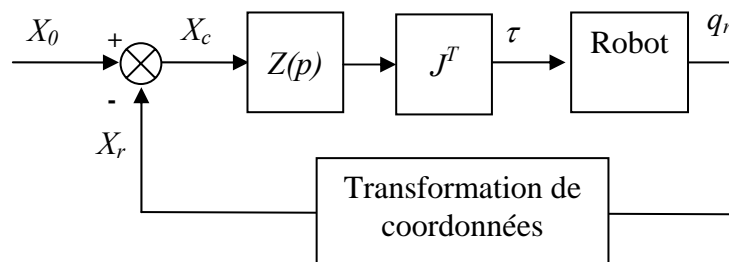


Figure 1.4. Commande en impédance par contrôle de force

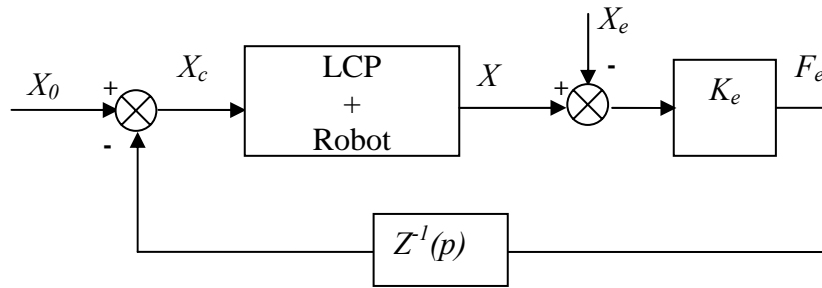


Figure 1.5 Commande en impédance par contrôle de position

Le schéma de la figure 1.4 peut être vu comme un contrôleur implicite de force qui ne nécessite pas une mesure externe de l'effort, de plus il ne tient pas compte du modèle dynamique du robot (à l'exception de la compensation des forces de gravité). Dans cette structure, le vecteur de commande est choisi donc comme suit :

$$\tau = J^T (B_d \dot{E} + K_d E) + G(q) \quad (1.12)$$

Où :

B_d et K_d sont respectivement les matrices d'amortissement et de raideur désirées.

Ce type de commande, simple à implémenter, est similaire à une commande proportionnelle dérivée dans l'espace opérationnel. En réalité, les performances de cette structure ne deviennent intéressantes que lorsque on désire avoir un comportement compliant du robot évoluant à des vitesses faibles c'est-à-dire lorsqu'on peut négliger les paramètres dynamiques du robot. Dans ce cas nous obtenons la compliancé désirée dans la direction définie par l'opérateur sans avoir besoin d'utiliser un capteur d'effort. On note ici que les structures de commande par matrice de raideur et celle par matrice d'amortissement définies précédemment peuvent être considérées comme des cas particuliers de la structure de commande en impédance donnée par la figure 1.5. Dans le premier cas l'impédance $Z(p)$ est réduite à une matrice de raideur K et dans la seconde à une matrice d'amortissement B .

Les principaux inconvénients de ces deux structures de commande classiques c'est qu'elles ne donnent aucune possibilité d'imposer un effort désiré à l'effecteur du robot, de plus, leurs performances ne sont valables que pour des tâches pour lesquelles on peut négliger la dynamique du robot ce qui limite le champ des tâches exécutées par le robot en milieux contraints. Pour remédier à ces problèmes, des structures de commande dynamique force/impédance à base des schémas précédents ont été proposées dans la littérature [JUN 95,98], elles sont représentées sur les figures suivantes :

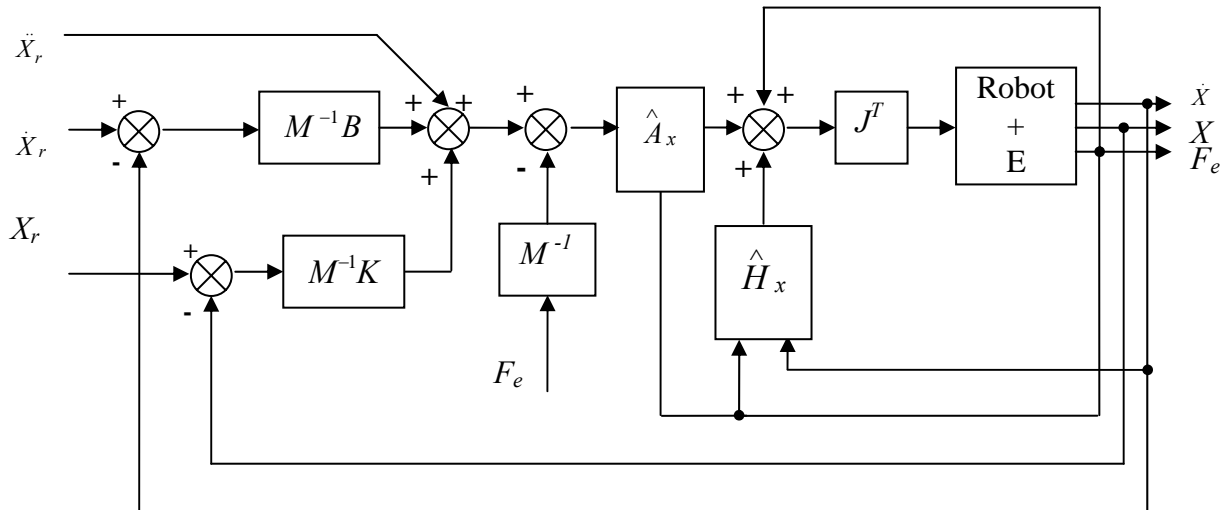


Figure 1.6 : Structure dynamique de la commande force/ impédance basée sur l'effort

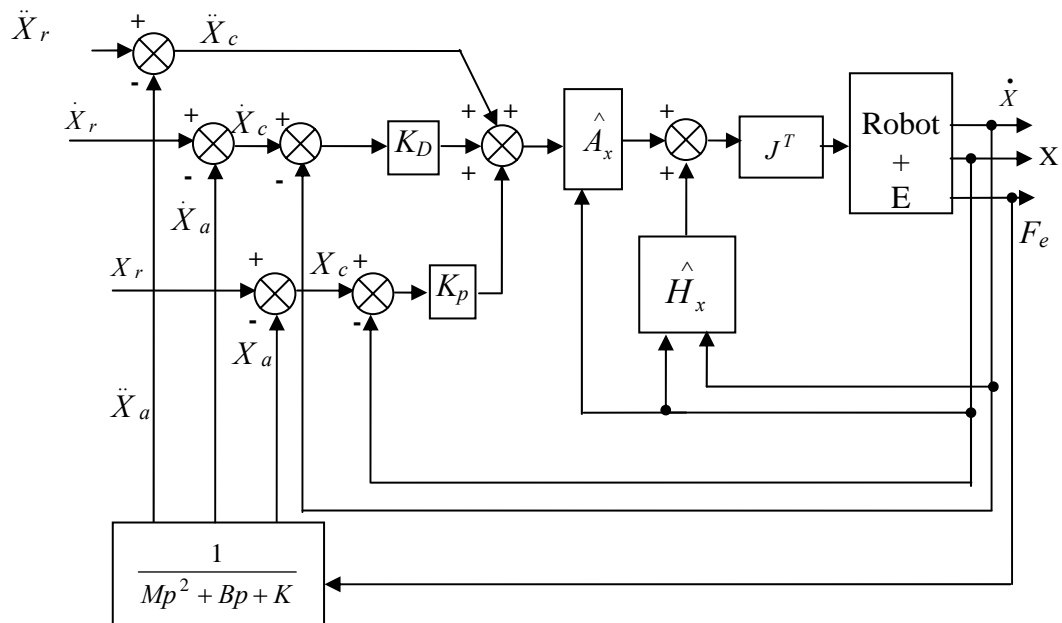


Figure 1.7 : Structure dynamique de la commande force/ impédance basée sur la position

Où

- \hat{A}_x et \hat{H}_x sont respectivement les matrices d'inertie et des forces centrifuges et de Coriolis estimées du manipulateur définies dans l'espace cartésien,
- M , B et K sont les matrices diagonales symétriques définies positives, qui représentent respectivement les paramètres, d'inertie, d'amortissement et de raideur de l'impédance désirée du robot manipulateur.

Dans le cas idéal (connaissance parfaite du modèle dynamique du robot) les deux structures de commande en boucle fermée précédentes sont régies par une même équation reliant les forces de contact à la variation de la position, qui représente l'impédance du système, donnée par :

$$F_e = M.\ddot{E} + B.\dot{E} + K.E \quad (1.12)$$

Avec,

$B = M.K_D$ et $K = M.K_p$ dans le cas de la structure de la figure 1.7 où K_D et K_p sont des matrices diagonales de réglage.

Dans ces deux structures de commande, l'effort désiré est généré de manière implicite grâce à une technique de modification de la trajectoire de référence pour les directions commandées en effort donnée par l'algorithme suivant [JUN 98] :

$$x_r = \begin{cases} x_e + \frac{f_d}{k} & \text{si } f_e = 0 \\ x_e + f_d \left[\frac{k(x - x_e) + f_e}{kf_e} \right] & \text{si } f_e \neq 0 \end{cases} \quad (1.13)$$

Où,

k , x_r , f_d et f_e sont respectivement les éléments de K , X_r , F_d et F_e ,

x_e élément du vecteur X_e qui définit la position de l'environnement.

Bien que cet algorithme permet de générer une consigne d'effort F_d , son inconvénient réside dans le fait qu'il est basé sur la détection de l'effort ressenti pour sélectionner la trajectoire de référence pour les directions commandées en effort, qui peut être sensible au bruit généré par les capteurs d'effort.

En pratique les performances de ces deux structures de commande dépendent de la bonne connaissance du modèle dynamique du robot exploité dans l'élaboration des commandes. Généralement la connaissance de ces modèles est toujours partielle, ce qui rend l'obtention de l'équation (1.12) difficile à atteindre.

Pour remédier à ce problème, des techniques de compensation de la dynamique inconnue du manipulateur, basées sur des approches de compensation neuronales [JUN 95a, 98, 00] et adaptatives [JUN 04], à base des deux structures classiques précédentes, ont été proposées dans la littérature.

D'autre part une méconnaissance de la raideur de l'environnement rend difficile le réglage des paramètres de l'équation d'impédance (1.12). Pour remédier à ce problème une technique adaptative en ligne de ces paramètres utilisant la logique floue [SUR 99] et une autre basée sur un apprentissage de l'équation d'impédance en utilisant les réseaux de neurones [TSU 96] [TAN 04] ont été proposées dans la littérature.

1.5 Commande Hybride force/position

La commande hybride force/position consiste à avoir simultanément des consignes d'effort et des consignes de position. Ainsi lorsqu'un effecteur entre en contact avec l'environnement il forme avec ce dernier une chaîne cinématique fermée, et de faibles variations de position de l'effecteur peuvent faire apparaître des efforts élevés. C'est pourquoi, il est important de distinguer une commande en effort d'une commande en mouvement.

Lorsque le robot est contraint par l'environnement suivant toutes les directions, aucun déplacement de l'organe terminal n'est possible et il ne peut qu'exercer des efforts sur ce dernier grâce à des déplacements infinitésimaux. Cependant si le robot évolue dans un espace libre, l'absence de contact avec l'environnement empêche la création de tout effort, il existe donc une dualité entre la commande en effort et la commande en position.

Il apparaît donc que l'espace de commande doit être subdivisé en deux sous espaces complémentaires, l'un constitué des directions commandées en effort et l'autre des directions commandées en position. La synthèse de ce type de commande dite commande hybride a été formalisée par MASON [MAS 81]. Les schémas de commande hybride force/position font apparaître deux boucles d'asservissement l'une munie d'un contrôleur en position et l'autre d'un contrôleur en effort. Les commandes issues de chacune des boucles sont additionnées sous la forme d'un vecteur de commande globale avant d'être envoyées au robot. On distingue deux grandes classes :

- L'approche associée à la première classe permet de calculer les couples à appliquer aux articulations en fonction des consignes de position et de force (figure 1.8).
- L'approche utilisée à la seconde classe permet de considérer que toute application d'effort par l'effecteur est réalisée par un déplacement du robot. L'écart entre l'effort mesuré et la consigne devient un déplacement relatif qui s'ajoute à la vitesse de commande de l'asservissement de position (figure 1.9).

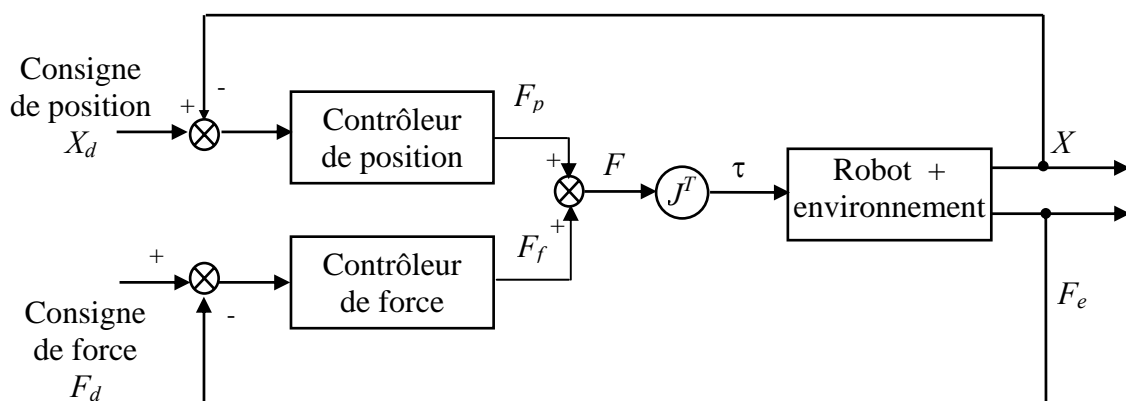


Figure 1.8 : Commande basée sur l'addition des efforts

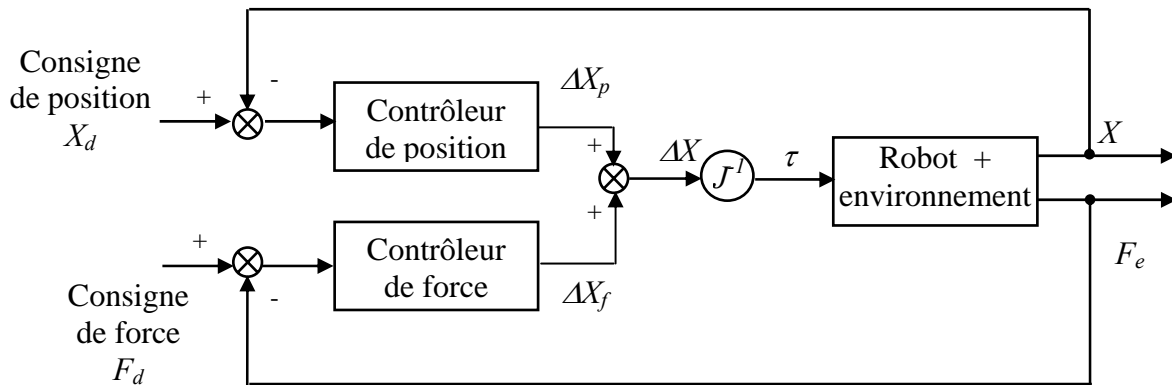


Figure 1.9 : Commande basée sur l'addition de déplacements

Pour assurer l'orthogonalité des commandes issues de chaque boucle et éviter tout conflit au niveau des actionneurs, il est nécessaire d'inclure dans ces structures une matrice de sélection apportée par la théorie de MASON [MAS 81] qui détermine quels sont les degrés de liberté (ddl) commandés respectivement en position ou en force selon la tâche qu'on veut réaliser ($S_i=1$ si le degré de liberté est commandé en force et $S_i=0$ dans le cas contraire).

En réalité la matrice de sélection dépend de la surface de contrainte. La position de cette matrice dans le schéma de commande dépend du repère dans lequel le contrôle des forces et des positions est effectué. Elle dépend aussi de l'espace de commande qui est utilisé (espace des coordonnées généralisées ou espace de la tâche). Ainsi les variantes de structure diffèrent par l'enchaînement des transformations géométriques.

1.5.1 Structure Hybride de CRAIG ET RAIBERT

La structure de CRAIG et RAIBERT [RAI 81], représentée par la figure 1.10, est le premier schéma apparu dans la commande hybride force/position. L'idée générale de cette méthode est déduite de la constatation suivante: un même degré de liberté ne peut être commandé à la fois en position et en effort. En effet, ces deux types de contrôle sont antagonistes. D'où l'idée de partitionner l'espace en deux sous-espaces: l'un contenant les degrés de liberté commandés en position et l'autre les degrés de liberté commandés en force. Puisque un même actionneur peut intervenir sur plusieurs degrés de liberté, le principe de la commande hybride consiste à envoyer simultanément aux actionneurs des contributions en position et en effort.

Notons que les consignes de force et de position ainsi que les erreurs de suivi et les matrices de sélection sont spécifiées dans le repère de la tâche tandis que les corrections se font dans l'espace des coordonnées articulaires. Ceci constitue le point faible de cette architecture de commande car elle requiert l'inversion de la matrice Jacobienne J qui peut être une source d'instabilité [PER 91]. En effet, la condition suffisante de stabilité dépend de la matrice Jacobienne, donc de la configuration du robot.

L'étude de la stabilité de ce schéma de commande a été reprise par [ZHA 89]. Mais l'inconvénient principal de cette analyse réside dans la simplification considérée en supposant un modèle linéarisé du robot, alors que le système dynamique du manipulateur sous une structure de commande hybride est fortement non linéaire. YABUTA [YAB 92b] a proposé des

améliorations à cette structure en proposant une commande hybride non linéaire dont la stabilité a été analysée en utilisant une fonction de Lyapunov. Mais le problème qui reste inchangé est que cette structure utilise toujours des matrices de sélection qui peuvent en réalité modifier les normes des vecteurs de commande pour certaines tâches exécutées par le robot [PER 91].

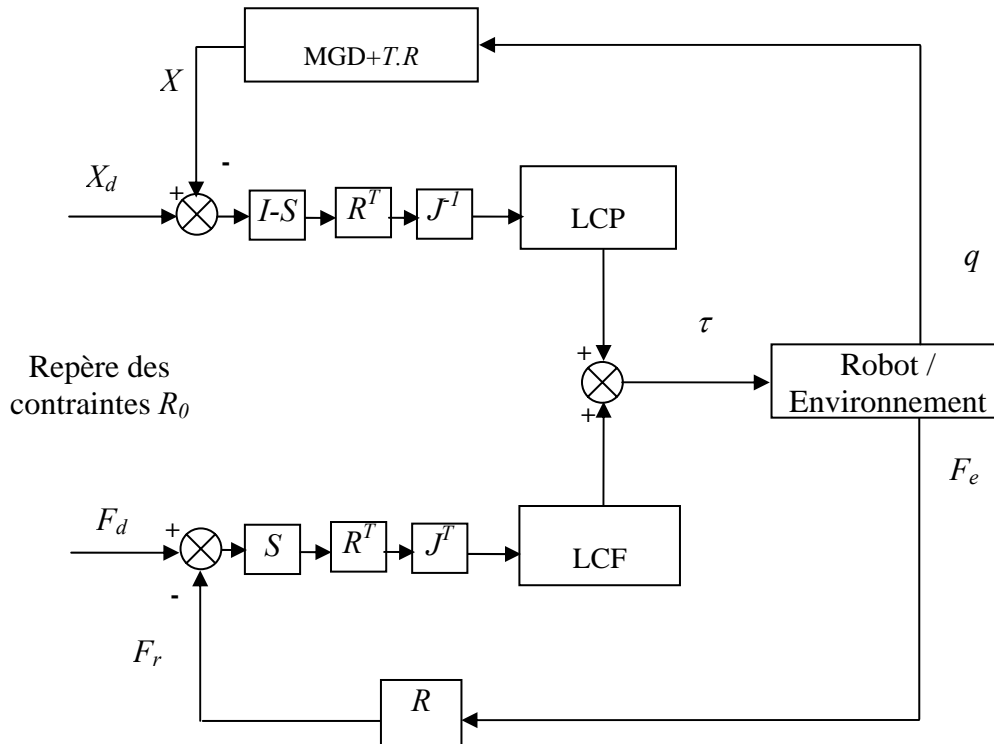


Figure 1.10 : Structure de commande de CRAIG et RAIBERT

1.5.2 Structure de KHATIB [KHA 86, 87]

KHATIB propose une autre formulation de la commande hybride force position dans l'espace cartésien dans laquelle tous les niveaux de cette structure de commande sont placés dans le repère de référence. Il introduit une matrice de spécification de la tâche Ω et sa matrice complémentaire $\bar{\Omega}$. Ces deux matrices assurent l'orthogonalité des commandes des contrôleurs. Pour compenser les perturbations dues à la dynamique du robot manipulateur, KHATIB introduit la matrice inertielle nominale estimée Λ_x dans l'asservissement de position pour découpler les déplacements du point terminal dus aux commandes issues du contrôleur de position, dans l'espace opérationnel; auquel il ajoute les vecteurs de compensation des forces de gravité $\hat{g}(q)$, de Coriolis et centrifuge $\hat{b}_0(q, \dot{q})$ dans l'espace articulaire (figure 1.11). Cependant aucune solution n'est proposée pour le découplage de l'effort alors que l'effet est le même. Bien que cette structure semble mieux adaptée, puisque les asservissements sont réalisés dans le repère de référence, ses performances dépendent de la connaissance préalable de la géométrie de l'environnement (utilisation des matrices de sélection) et du degré de connaissance du modèle dynamique du robot.

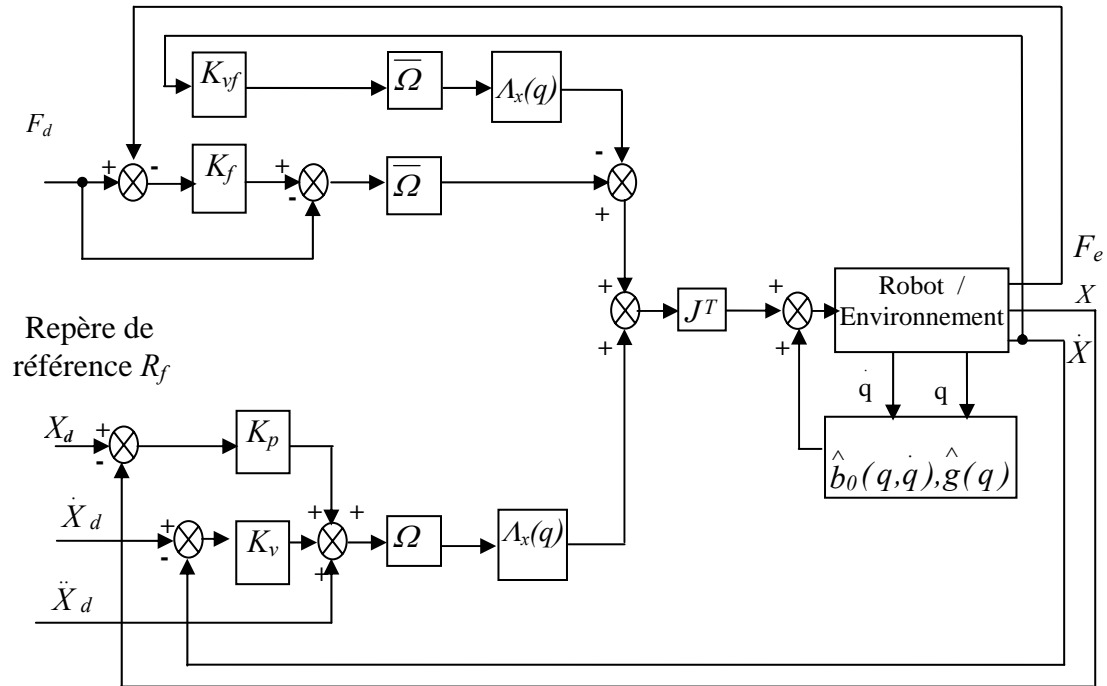


Figure 1.11. Structure hybride dynamique de KHATIB.

Les approches hybrides citées ci-dessus permettent de commander le robot en force et en position mais elles nécessitent un modèle géométrique détaillé de l'environnement, d'autre part les données provenant des capteurs passent par une sélection liée à la tâche à exécuter. Par contre une commande en impédance proposée par HOGAN [HOG 85] est capable de s'adapter à une classe d'environnements mais sans contrôler à la fois la force et la position.

Deux autres structures sont proposées dans la littérature qui combinent les avantages des deux méthodes décrites précédemment. Contrairement à la stratégie du contrôle hybride, l'ensemble des variables force et position sont utilisées le long d'une même direction de l'espace de la tâche sans utiliser le mécanisme de sélection, il s'agit de la structure de commande parallèle proposée par CHIAVERINI [CHI 92, 93, 94] et la structure de commande en effort externe proposée par SHUTER [SHU 88]

1.6. Structure de commande parallèle force/position :

CHIAVERINI [CHI 92, 94] a proposé une structure de commande parallèle force/position qui combine les avantages des commandes hybrides et la commande en impédance décrite précédemment. Dans cette structure, la commande en force est calculée de façon à prévaloir sur la commande en position. En cas de conflit, les écarts de la tâche planifiée donnent priorité à la commande en effort. Ce schéma de commande donnée par la figure 1.12, néglige la quasi-totalité de la dynamique du manipulateur, il est composé de la somme d'un effort désiré, d'une action PD dans la boucle de commande en position, d'une action PI dans la boucle de commande en effort et le vecteur des forces de gravité estimé.

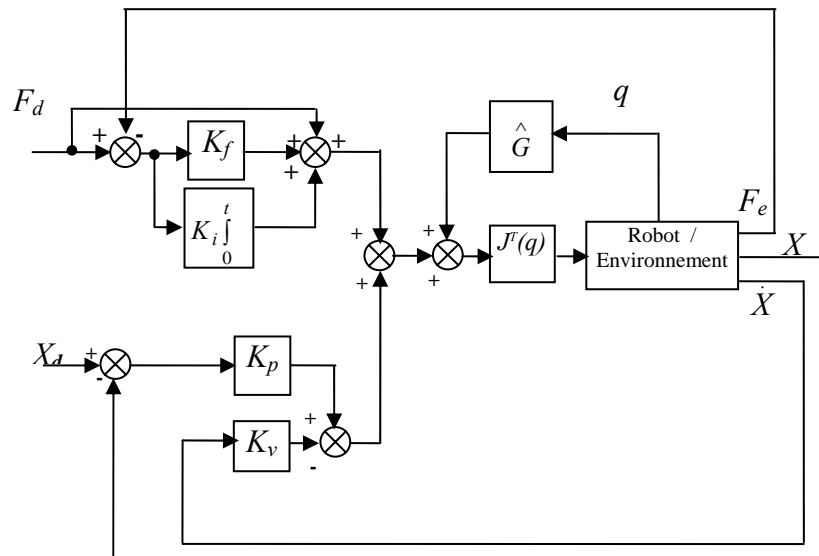


Figure 1.12 : Structure parallèle force/position simplifiée de CHIAVERINI

Pour prouver la stabilité de cette structure de commande pour des tâches appliquées dans des milieux compliant CHIAVERINI a proposé une étude basée sur l'exploitation d'une fonction de Lyapunov. Cependant les performances de cette commande ont été illustrées uniquement pour des tests de compliance en utilisant des grands gains dans la boucle de commande en position ($K_p = 10^5$ et $K_v = 10^4$). Cependant le choix de grands gains dans une commande risque de poser problème de stabilité du système au moment de sa discrétisation. [DUM 87]. Un autre inconvénient de ce choix est le coût matériel au moment de sa réalisation.

Une autre structure de commande parallèle force/position, basé sur la même stratégie que la précédente, qui tient compte cette fois ci de la dynamique du robot est aussi proposée dans la littérature [CHI 93]. Comparativement à la structure dynamique proposée par KHATIB, cette structure permet de faire un découplage simultané en position et en effort, de plus la compensation de la dynamique du manipulateur est réalisée en totalité dans l'espace cartésien. Aucune matrice de sélection n'est utilisée dans cette structure de commande, une stratégie de dominance de la loi de commande en effort sur celle en position le long des directions contraintes où il y a interaction qui est exploitée ici. De plus la stabilité de cette structure ne dépend que de réglage des paramètres scalaires des contrôleurs [CHI 93].

Cependant les performances de cette structure de commande dépendent de la connaissance préalable de la dynamique du manipulateur qui est généralement difficile à atteindre. Cette technique de commande a été rendue adaptative par SICILIANO et VILANI [SIC 94] en agissant sur les dynamiques du modèle pour améliorer la poursuite de trajectoire lorsque les paramètres sont mal estimés. Mais les performances ont été démontrées uniquement pour des tâches exécutées dans le plan, lorsque les repères de référence et celui lié à l'environnement sont choisis parallèles.

1.7 Structure de commande en effort externe :

Dans la structure de commande en effort externe formalisée par SHUTER et VAN BRUSSEL [SHU 88] et développé et étudié par PERDEREAU [PER 91], les deux boucles d'asservissement ne sont pas en parallèle, mais on a plutôt une hiérarchisation de la boucle d'effort sur la boucle de position. La figure 1.13 représente le schéma de principe de la structure de commande en effort externe.

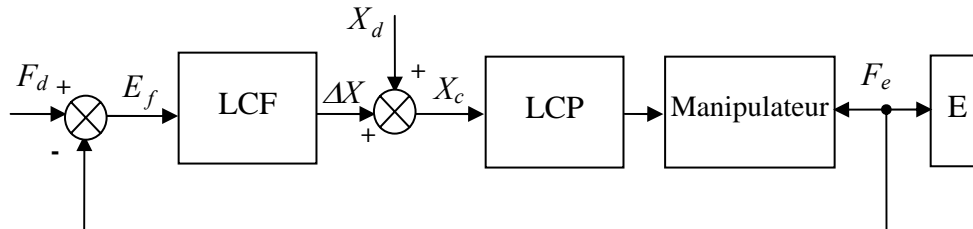


Figure 1.13. Structure de commande en effort externe

L'idée originale de cette structure est donc de convertir l'erreur en effort en déplacement ΔX , grâce à un contrôleur d'effort (LCF). Ce déplacement est ajouté à la consigne de position X_d qui se trouve ainsi modifier et devenir la nouvelle consigne de la boucle de commande en position. Donc au lieu d'avoir deux boucles parallèles dont on va additionner les commandes, on a deux boucles hiérarchiques dont on somme les déplacements. Si l'analyse de la tâche désirée est parfaite (connaissance exacte de l'environnement), l'orthogonalité des commandes est directement assurée par les consignes. Un premier avantage de cette structure est donc de pouvoir utiliser des consignes déduites de l'analyse de la tâche. Un autre avantage de cette structure que l'on peut citer ici est que l'asservissement en position reste le même quelque soit la configuration de la tâche, donc aucun changement au niveau de la structure.

Lorsque l'environnement est mal connu, le problème de l'orthogonalité peut se poser par le fait que la trajectoire désirée ne sera plus tangente à l'environnement, dans ce cas la consigne de position impose un déplacement selon une direction contrainte, qui agit comme une perturbation sur l'asservissement en effort, mais qui peut être compensée par le contrôleur d'effort mené d'action de type intégrale en imposant des forces désirées nulles selon ces directions.

Un autre avantage de cette structure de commande (très utilisé dans les laboratoires français) est sa simplicité de mise en œuvre, elle donne la possibilité de concevoir des approches de commande en position et en effort de manière indépendante. Ainsi SAADIA [SAA 97] a proposé une approche neuronale adaptative pour la boucle de commande en effort, appliquée à une tâche d'assemblage en exploitant l'asservissement en position d'un robot existant. DEGOULANGE [DEG 93] a exploité cette structure de commande en utilisant des contrôleurs classiques de type PID pour la boucle de commande en effort appliqués pour des tâches de manipulation d'objet par des robots coopérants. FRAISSE [FRA 94] a proposé deux approches jugées robustes, l'une pour la boucle de commande en position (utilisant les modes glissants) et l'autre pour la boucle de commande en effort (utilisant une approche basée sur un vecteur de commande utilisant un environnement virtuel dans le but de minimiser l'influence de la variation de la raideur de l'environnement sur la réponse en force).

Cependant toutes les approches citées ci-dessus ont été validées dans des cas de tâches relativement simples c'est-à-dire les cas où la dynamique du robot peut être négligée. Les applications sont limitées à des cas quasi-statiques alors que cette hypothèse n'est plus valable lorsqu'il s'agit par exemple du cas d'un suivi du contour dans un milieu contraint faisant varier considérablement les configurations du manipulateur à des vitesses relativement élevées.

Avec cette structure, la boucle de commande en effort est toujours définie dans l'espace cartésien alors que la boucle de commande en position peut être définie dans les deux espaces cartésien (figure 1.14) ou articulaire (figure 1.15). Cependant ce dernier choix peut poser problème quant à l'analyse de la stabilité du système global faisant intervenir une transformation non linéaire (modèle géométrique inverse MGI) pour passer de l'espace cartésien à l'espace articulaire.

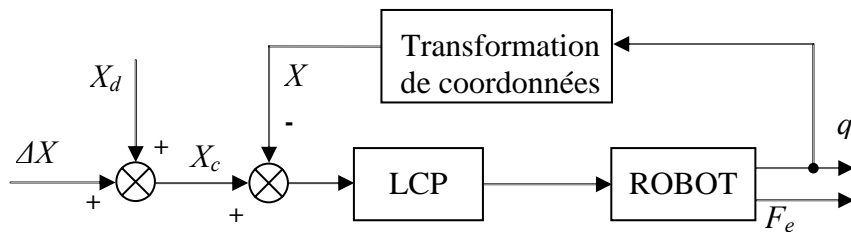


Figure 1.14 : Commande en position dans l'espace cartésien

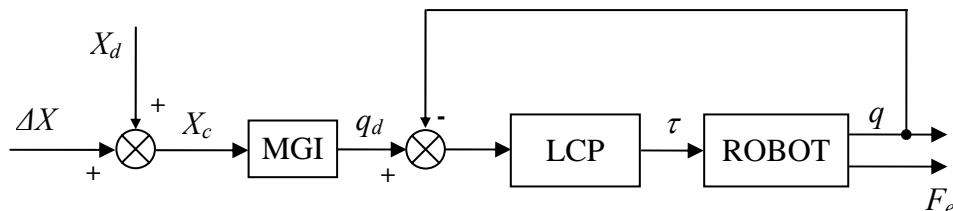


Figure 1.15 : Commande en position dans l'espace articulaire

1.8 Conclusion

Dans ce chapitre nous avons présenté les différentes structures de commande permettant de contrôler les efforts de contact robot/environnement. Nous avons vu que certaines de ses structures telles que la commande par retour d'effort explicite et la commande en impédance ne donnent pas la possibilité d'asservir les efforts. La structure proposée dans [RAI 81], bien qu'elle comporte deux boucles parallèles, l'une asservie en position, l'autre en force, elle utilise plusieurs transformations géométriques et des matrices de sélection destinées à commuter entre la boucle d'asservissement de force et de position et à assurer l'orthogonalité des commandes. Ces transformations et ces commutations compliquent la mise en œuvre de cette structure. De plus cette structure néglige complètement la dynamique du robot. Malgré les améliorations apportées à cette commande par KHATIB [KHA 86,87] en proposant une

structure permettant de réaliser les asservissements dans le repère de référence utilisant le modèle dynamique du robot dans la boucle de position pour le découplage, le problème qui reste inchangé est l'utilisation des matrices de sélections dans les boucles de commande en position et en effort qui rend relativement difficile la mise en œuvre de cette structures. La structure de commande force/position proposée par CHIAVERINI [CHI 93, 94](structure parallèle) et celle proposée par SHUTTER [SHU 88] (structure en effort externe) sont basées, contrairement aux deux structures précédentes, sur la stratégie de dominance de la boucle de commande en effort sur la boucle de commande en position permettant de commander le robot en force et en position le long d'une même direction sans avoir besoin d'utiliser un mécanisme de sélection. Cependant les performances de ces deux structures dépendent de la connaissance exacte de la dynamique du robot utilisé et celle de son interaction avec l'environnement. En effet les variations des différents paramètres caractéristiques tant du robot lui-même que de l'environnement ont un impact sur la réponse dynamique d'un manipulateur en contact.

Devant la complexité des modèles mis en jeu et les incertitudes des valeurs numériques de ces modèles, il s'avère difficile de concevoir des commandes classiques performantes d'un système non linéaire inconnu ou partiellement connu.

Notre contribution dans ce travail consiste à associer à ces deux structures de commande classiques des éléments compensateurs, basés sur des techniques issues de l'intelligence artificielle (les réseaux de neurones et la logique floue) pour améliorer leurs performances.

Dans le chapitre suivant, nous présenterons les notions fondamentales concernant les réseaux de neurones artificiels et la logique floue et quelques structures de commandes proposées dans la littérature basées sur ces concepts et celles dont nous nous sommes inspirés pour développer et proposer dans la suite de notre travail, de nouveaux schémas de commande force/position permettant de répondre aux questions soulevées par les différentes méthodes de la littérature.

Chapitre 2

Structures de commande neuronales et floues

2.1 Introduction

Dans ce chapitre, nous allons présenter une brève introduction aux domaines des réseaux de neurones et de la logique floue afin d'introduire les familles d'algorithmes que nous avons utilisé dans ce travail puis de présenter quelques structures utilisant les réseaux de neurones et la logique floue pour commander un processus. Cette démarche est menée afin de lever une contrainte en contrôle relative à la nécessité de disposer des modèles parfaits du robot et de son environnement. En effet les erreurs de modélisation et les imperfections des modèles tant du robot lui-même que de l'environnement, contribuent à dégrader notablement les performances des lois de commandes force/position classiques pour des tâches faisant intervenir simultanément le mouvement et l'effort appliqué par l'organe terminal d'un bras manipulateur sur son environnement.

Dans la première partie, nous discuterons des algorithmes dits "réseaux de neurones", dont les propriétés et les caractéristiques peuvent être exploitées pour traiter le problème posé, puis nous passerons en revue certaines structures rencontrées dans la littérature.

La deuxième partie de ce chapitre sera consacrée à la théorie de la logique floue, où nous donnons quelques définitions mathématiques sur les ensembles flous et leurs caractéristiques ainsi que les systèmes d'inférence flous les plus usuels, puis nous passerons en revue certaines structures rencontrées dans la littérature.

2.2 Les Réseaux de neurones

2.2.1 Introduction

Les réseaux de neurones artificiels sont des éléments de calcul parallèle qui permettent "d'imiter" certaines des fonctions du cerveau humain en reproduisant certaines de ses structures de base, suivant une topologie de connexions.

La recherche en réseaux de neurones artificiels a été menée afin de donner plus « d'intelligence » aux calculateurs. Les réseaux de neurones sont utilisés dans plusieurs domaines pour résoudre des problèmes complexes en contrôle, en traitement du signal, en reconnaissance de forme, en économie...etc. La fonction majeure d'un réseau de neurones est sa capacité de reproduire toute fonction non linéaire grâce à sa caractéristique d'apprentissage. Lorsque le réseau est entraîné sur des exemples d'une base de données, il pourra encore prédire le comportement futur d'un système donné.

Dans le domaine du contrôle d'un processus physique donné, les réseaux de neurones sont utilisés comme des identificateurs de la dynamique d'un système ou bien comme contrôleurs pour commander un système. En effet la majorité des systèmes physiques sont non linéaires, les réseaux de neurones peuvent être un bon choix pour la réalisation des contrôleurs de ces systèmes. Leurs capacités du calcul parallèle, d'apprentissage, d'adaptation, de généralisation sont des caractéristiques très importantes qui peuvent être intéressantes et exploitées dans plusieurs domaines.

2.2.2 Généralités sur les réseaux de neurones

2.2.2.1 Structure d'un réseau de neurones

Un neurone artificiel formel est un modèle mathématique d'un neurone biologique. Il est composé d'un processus d'entrée, d'une jonction de sommation, de poids, et des sorties qui

correspondent approximativement aux dendrites, le noyau (ou corps cellulaire) et l'axone d'un neurone biologique donné par la figure 2.1 suivante [HAY 99] :

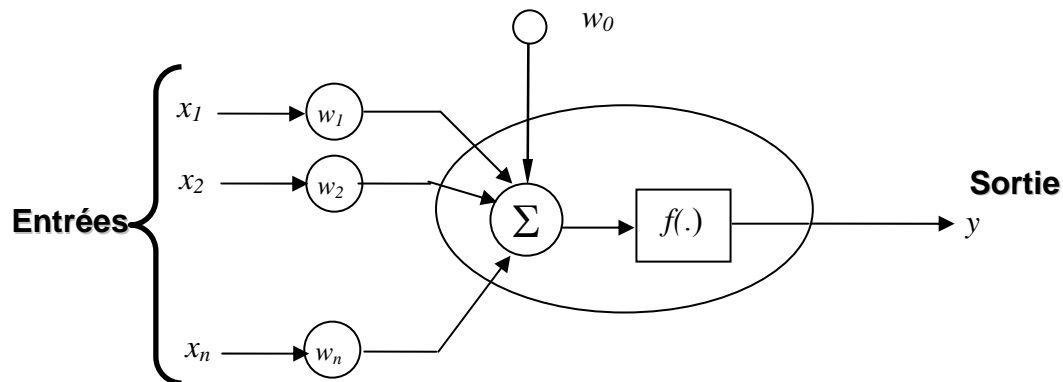


Figure 2.1 Modèle d'un neurone artificiel

Un neurone élémentaire ou formel est défini comme une unité fonctionnelle caractérisée par une fonction d'activation $f(.)$ décrivant l'état de sa sortie y en fonction de ses entrées x_i . Ainsi la sortie du neurone sera :

$$y = f\left(\sum_{i=1}^n x_i w_i + w_0\right) \quad (2.1)$$

Où w_i sont les poids de pondération des connexions et w_0 est le paramètre de biais. Le biais peut être considéré comme un poids qui peut être adapté de la même manière que les autres poids. La fonction d'activation $f(.)$ peut être une fonction linéaire, une fonction seuil, ou une fonction sigmoïde dont l'utilisation dépend du problème posé.

$$\text{Fonction linéaire} \quad f(x) = ax \quad -B \leq f(x) \leq B \quad (2.2)$$

$$\text{Fonction sigmoïdale} \quad f(x) = \frac{1}{1 + e^{-\lambda x}} \quad 0 \leq f(x) \leq 1 \quad (2.3)$$

λ détermine la pente de l'activité de la fonction $f(x)$. La fonction sigmoïdale est saturée en 0 et 1, donc elle génère des valeurs positives. Une autre fonction qu'on peut définir appelée tangente hyperbolique qui permet de générer des valeurs entre -1 et 1 est donnée par l'équation suivante :

$$f(x) = \frac{1 - e^{-\lambda x}}{1 + e^{-\lambda x}} \quad -1 \leq f(x) \leq 1 \quad (2.4)$$

Un réseau de neurones est l'interconnexion pondérée de plusieurs neurones considérés comme éléments de base. Ce réseau effectue une application de l'espace des entrées dans l'espace des sorties de façon parallèle et distribuée. On peut décomposer un réseau de

neurones en plusieurs champs de neurones, appelé couches. On distingue alors : la couche d'entrée, les couches cachées et la couche de sortie.

En pratique on peut distinguer plusieurs architectures de réseaux selon la topologie des connexions des neurones ainsi que la forme des fonctions d'activation des neurones. Les réseaux de neurones qui utilisent l'algorithme de rétropropagation de l'erreur peuvent être classés en deux groupes qui sont les réseaux statiques et les réseaux récurrents [HAY 99]. Nous nous limiterons aux réseaux statiques que nous utiliserons dans les différentes structures de commande force/position à base des réseaux de neurones proposées dans la suite de cette thèse.

2.2.2.2 Les réseaux de neurones statiques

Un réseau de neurones statique est caractérisé par la topologie de ses connexions qui relie les différentes couche du réseau et qui ne contient pas de boucle synaptiques fermée. La figure suivante (figure 2.2) montre un réseau de neurone typique à trois couches, une couche d'entrée, une couche cachée et une couche de sortie.

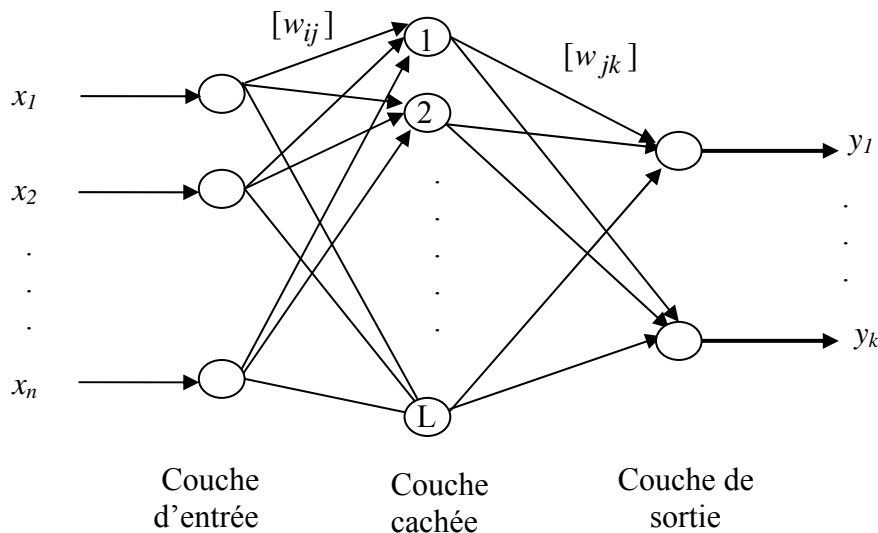


Figure 2.2: Réseau de neurones statique

2.2.2.3 Apprentissage des réseaux de neurones artificiels

L'apprentissage des réseaux de neurones est l'ensemble des algorithmes d'ajustement de ses poids pour lui permettre une application entre l'espace d'entrée qu'on lui présente et l'espace de sortie correspondant. Il existe plusieurs règles d'apprentissage qui peuvent être regroupées en trois catégories: les règles d'apprentissage supervisé, non supervisé et renforcé [HAY 99]. Mais l'objectif de l'apprentissage reste le même: soit la classification, l'approximation de fonction ou encore la prévision.

Dans le cas d'apprentissage supervisé (le plus utilisé en contrôle), nous devons disposer de deux espaces entrée/sortie et le réseau peut être entraîné en ajustant ses paramètres systématiquement en cherchant à approximer au mieux les sorties désirées correspondant à chaque vecteur d'entrées en se basant sur le principe essai-erreur. La figure suivante donne un

exemple dans le cas d'un réseau à trois couches (une couche d'entrée, une couche cachée et une couche de sortie).

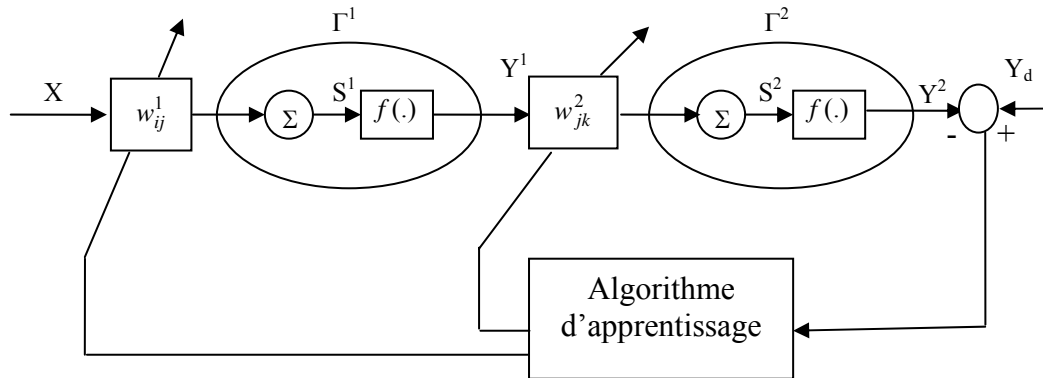


Figure 2.3. Réseau de neurones à trois couches

Le nombre de neurones pour la couche d'entrée, la couche cachée et la couche de sortie sont respectivement notés l , m , n . la sortie de la couche cachée, noté Y^1 est donnée par l'équation suivante :

$$Y^1 = \Gamma^1 \left[W_{ij}^{1T} X \right] \quad (2.5)$$

Où X est le vecteur d'entrée de dimension $l \times 1$,

W_{ij}^1 est la matrice des poids entre la couche d'entrée et la couche cachée de dimension $l \times m$, Y^1 est le vecteur de sortie de la couche cachée, et Γ^1 est un opérateur matriciel diagonal de dimension $m \times m$ entre la couche d'entrée et la couche cachée.

L'opérateur Γ est donné par :

$$\Gamma = f(S) \quad (2.6)$$

Le passage de la couche d'entrée vers la couche de sortie est donné par l'équation suivante :

$$\begin{aligned} Y^2 &= \Gamma^2 \left[W_{jk}^{2T} Y^1 \right] \\ &= \Gamma^2 \left[W_{jk}^{2T} \Gamma^1 \left[W_{ij}^1 X \right] \right] \end{aligned} \quad (2.7)$$

Où W_{jk}^{2T} est la matrice de dimension $m \times n$ entre la couche cachée et la couche de sortie,

Y^2 est le vecteur de sortie, et Γ^2 est un opérateur matriciel diagonal de dimension $n \times n$ entre la couche cachée et la couche de sortie.

Le vecteur d'erreur E est l'écart entre le vecteur de sortie actuel et le vecteur d'entrée désiré donné par :

$$E = Y_d - Y^2 \quad (2.8)$$

L'algorithme d'apprentissage est basé sur la minimisation de la fonction coût suivante :

$$J = \frac{1}{2} E^T E \quad (2.9)$$

2.2.2.4 Algorithme de rétropropagation

L'algorithme d'apprentissage de rétropropagation pour les réseaux de neurones est basé sur la méthode du gradient descendant [HAY 99] qui recherche le minimum d'une fonction coût spécifiée. Si on considère la fonction coût donnée par l'équation (2.9), on suppose que la règle d'apprentissage se base sur un certain nombre d'exemples P. Si on définit y_k comme k ième sortie du réseau et y_{dk} comme sortie désirée, alors la fonction coût sera :

$$J_p = \frac{1}{2} \sum (y_{dk} - y_k)^2 \quad (2.10)$$

Le gradient de cette fonction sera :

$$\frac{\delta J_p}{\delta w} = -\sum e_k \frac{\delta y_k}{\delta w} \quad (2.11)$$

Avec $e_k = y_{dk} - y_k$

La règle d'adaptation des poids à partir de la fonction coût précédente est donné par :

$$\Delta w = -\eta \frac{\delta J_p}{\delta w} = \eta \sum e_k \frac{\delta y_k}{\delta w} \quad (2.12)$$

Où η s'appelle le pas d'apprentissage (ou pas d'adaptation)

2.2.2.5 Différentes étapes de calcul d'un réseau de neurones

Quelque soit le nombre de couches d'un réseau de neurones qui sont structurées de l'entrée vers la sortie, les poids de chacune de ces couches seront adaptés. Pour simplifier, on considère un réseau à trois couches une couche d'entrée i une couche cachée j , et une couche de sortie k . le nombre des entrées est N_e , le nombre de neurones de la couche cachée est N_c et le nombre de neurones de la couche de sortie est N_s .

1. les signaux d'entrée x_i sont multipliés par les poids w_{ij} et additionnés.

$$s_j = \sum_{i=1}^{N_e} w_{ij} x_i \quad (2.13)$$

Le signal passe à travers une fonction d'activation f et devient l'entrée de la couche suivante :

$$y_j = f(s_j) \quad (2.14)$$

Les entrées de la couche cachée sont multipliées par les poids w_{jk} et additionnées avant de passer à travers une fonction d'activation f pour donner la sortie suivante :

$$y_k = f(s_k) \quad (2.15)$$

2. l'adaptation des poids entre la couche cachée et la couche de sortie

$$\begin{aligned} \Delta w_{jk} &= -\eta \frac{\delta J_p}{\delta w_{jk}} \\ &= \eta e_k \frac{\delta y_k}{\delta w_{jk}} \\ &= \eta e_k f'(s_k) \frac{\delta s_k}{\delta w_{jk}} \end{aligned} \quad (2.16)$$

3. adaptation des poids w_{ij} entre l'entrée et la couche cachée

$$\begin{aligned} \Delta w_{ij} &= -\eta \frac{\delta J_p}{\delta w_{ij}} \\ &= \eta e_k \frac{\delta y_k}{\delta w_{ij}} \end{aligned} \quad (2.17)$$

Avec,

$$\begin{aligned} \frac{\delta y_k}{\delta w_{ij}} &= \frac{\delta y_k}{\delta y_j} \frac{\delta y_j}{\delta w_{ij}} \\ &= \frac{\delta y_k}{\delta y_j} \frac{\delta y_j}{\delta s_j} \frac{\delta s_j}{\delta w_{ij}} \\ &= \frac{\delta y_k}{\delta y_j} f'(s_j) x_i \end{aligned} \quad (2.18)$$

2.2.3 Les réseaux de neurones et la commande des processus

La récente émergence de nouveaux systèmes caractérisés par une forte complexité et le développement des moyens de calcul ont conduit à adopter de nouvelles approches qui ont contribué à résoudre certains problèmes liés à la commande des processus. Vu le caractère non linéaire des réseaux de neurones et vu leur capacité d'apprentissage, de généralisation et leur parallélisme, les automaticiens ont de plus en plus eu recours aux réseaux de neurones en identification et en commande des systèmes dynamiques non linéaires.

Nous présentons dans ce qui suit certaines stratégies usuelles de commande par réseaux de neurones trouvées dans la littérature en particulier celles dont nous nous sommes inspirés pour développer notre stratégie de commande pour résoudre le problème de la robustesse des structures de commande force/position classiques proposées dans la littérature (structure parallèle force/ position et structure de commande en effort externe).

2.2.3.1 Réseaux de neurones appliqués à l'identification

L'identification est une procédure qui permet d'obtenir un modèle d'un processus à partir d'un ensemble fini de données d'entrée/sortie du processus. Les réseaux de neurones sont utilisés pour l'identification des modèles directs et inverses des processus physiques grâce à leur capacité d'auto-organisation et de stockage de l'information. L'idée de base est d'apprendre leurs caractéristiques au lieu de formuler leurs modèles mathématiques.

L'objectif de l'identification directe est d'avoir une copie du processus P dans le réseau $N(\tau, w)$ tel que après convergence on aura $N(\tau, w) = \hat{P}$. Les entrées du réseau identificateur sont les mêmes que les entrées du processus comme il est montré à la figure 2.4.a. De même l'objectif de l'identification du modèle inverse donné par la figure 2.4.b est d'avoir l'inverse du processus P^{-1} dans le réseau $N(q, w)$ tel que après convergence on aura $N(q, w) = \hat{P}^{-1}$.

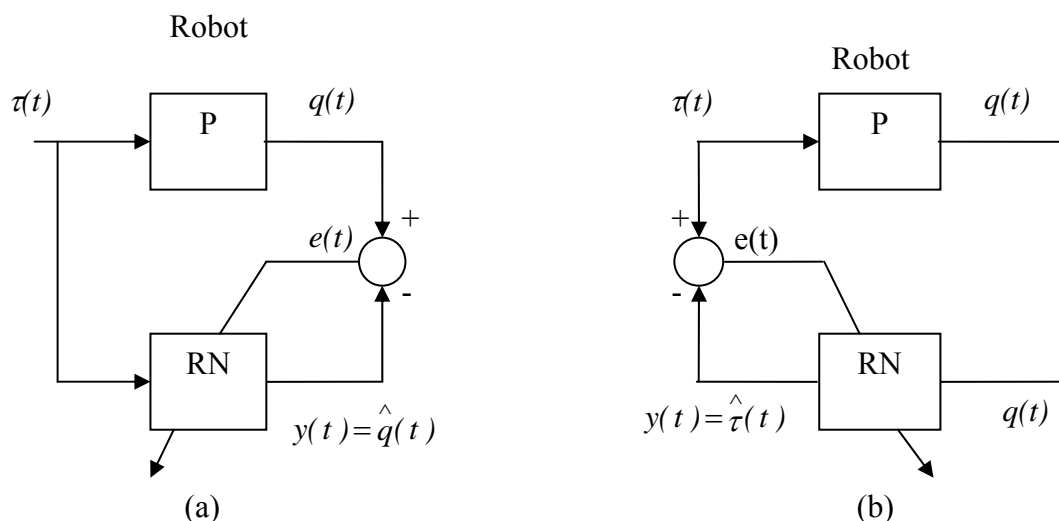


Figure 2.4 Identification d'un processus non linéaire (Robot) (a) directe
(b) inverse

Dans une structure de commande, le modèle neuronal inverse peut être utilisé comme un contrôleur du processus en le plaçant en cascade avec le système. Dans ce cas la sortie du système reproduit l'entrée de référence. Cette technique de commande est plus efficace pour commander des systèmes linéaires, elle devient complexe pour les systèmes non linéaires.

Pour les deux méthodes d'identification, la fonction coût à minimiser est donnée par:

$$J = \frac{1}{2} e^T(t) e(t) \quad (2.19)$$

La dérivée de la fonction coût donne:

$$\frac{\delta J}{\delta w} = \frac{\delta e^T}{\delta w} e = -\frac{\delta y^T}{\delta w} e \quad (2.20)$$

Donc les poids du réseau peuvent être adaptés en minimisant la fonction coût, puisque $\frac{\delta y^T}{\delta w}$ peut se calculer facilement.

2.2.3.2 Le contrôleur neuronal

Une structure de commande basée sur un contrôleur neuronal inverse direct, obtenu à partir de l'apprentissage du réseau du modèle inverse du système, peut être représenté par la figure 2.5. Cette structure de commande est très simple mais elle nécessite une connaissance préalable de la Jacobienne du système (équations dynamiques du système à contrôler) lorsque l'algorithme de rétropropagation de l'erreur est utilisé. Plusieurs possibilités ont été proposées dans la littérature pour surmonter le problème lié au calcul de cette Jacobienne soit:

- Par un calcul direct en exploitant les équations dynamiques du système supposées connus, ou bien par estimation de la Jacobienne en utilisant une méthode d'approximation différentielle entre l'entrée et la sortie.
- Par estimation de la Jacobienne dynamique par un autre réseau de neurones identifiant le modèle direct du système.
- En utilisant une boucle de commande classique dont la sortie sera utilisée comme signal d'entraînement du réseau à la place de la Jacobienne estimée.

Dans ce qui suit, nous détaillerons chacune de ces possibilités.

2.2.3.3 Le contrôleur neuronal inverse direct

La méthode la plus classique est de placer le réseau de neurones en cascade avec le système à commander comme il est montré sur la figure 2.5. Cette structure, appelée commande neuronale inverse directe ou bien schéma d'apprentissage spécialisé [PSA 88][YAB 92a], permet la commande en ligne. Les poids du contrôleur neuronal sont adaptés en minimisant l'erreur entre la position désirée et la position réelle (actuelle). Cela nécessite la connaissance de la Jacobienne du système à commander. Cependant dans la majorité des cas cette dernière n'est connue qu'approximativement.

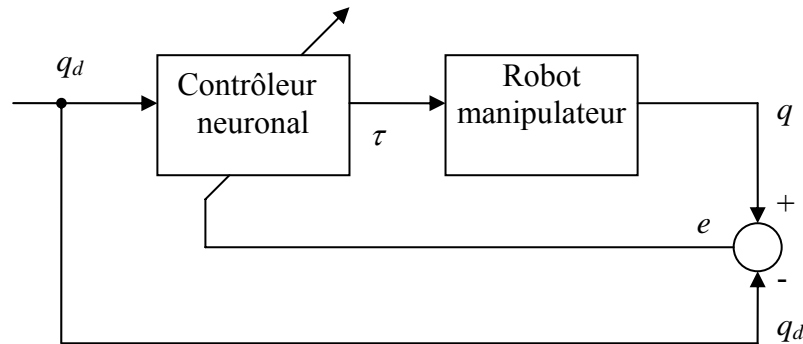


Figure 2.5 : Commande neuronale inverse directe

La fonction coût peut être donnée par:

$$J = \frac{1}{2} e^T e \quad (2.21)$$

Le gradient de cette fonction est donné par :

$$\frac{\delta J}{\delta w} = \frac{\delta e^T}{\delta w} \cdot e = -\frac{\delta \tau^T}{\delta w} \cdot \frac{\delta q^T}{\delta \tau} \cdot e \quad (2.22)$$

Où,

$\frac{\delta \tau^T}{\delta w}$ est généralement connu ($\tau = f(q_d, w)$), par contre $\frac{\delta q}{\delta \tau}$ dépend de la connaissance des modèles du système, chose qui n'est pas évidente. Une solution consiste à approximer cette fonction en ligne comme suit [PSA 88]:

$$\frac{\delta q}{\delta \tau} = \frac{q(t) - q(t-1)}{\tau(t) - \tau(t-1)} \quad (2.23)$$

Avec cette approximation le schéma de commande de la figure 2.5 peut être utilisé sans avoir besoin d'une connaissance préalable du modèle de processus.

Cependant cette structure de commande peut poser un problème de stabilité lors de l'utilisation de l'algorithme de rétropropagation dont la convergence dépend du choix des poids initiaux et du taux de propagation utilisé [YAB 92a]. Pour remédier à ce problème des modifications ont été apportées à cette structure, donnée par la figure 2.6 suivante.

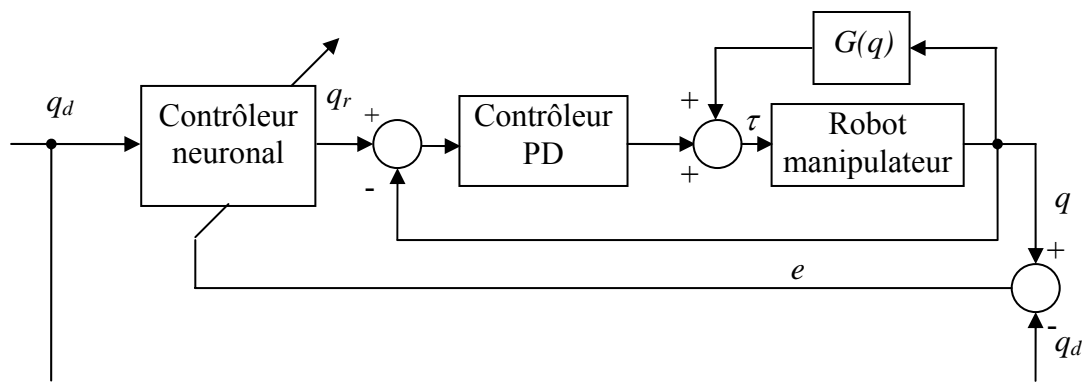


Figure 2.6 Commande neuronale inverse directe avec stabilisation du système

Cette structure introduit donc une boucle de commande proportionnelle dérivée (PD) et une compensation des forces de la gravité $G(q)$ pour assurer sa stabilité [JUN 96].

2.2.3.4 Commande avec modèle de référence [LIN 00]

Dans ce schéma de commande les performances désirées du système en boucle fermée sont obtenues à travers un modèle de référence stable défini par le couple d'entrée/sortie $\{r(t), y(t)\}$. Comme il est montré sur la figure 2.7, la commande essaie de mettre la sortie du système $y(t)$ asymptotiquement égale à l'entrée $r(t)$. Dans ce schéma l'erreur entre la sortie du système et la sortie du modèle de référence est utilisée pour ajuster les poids du contrôleur neuronal.

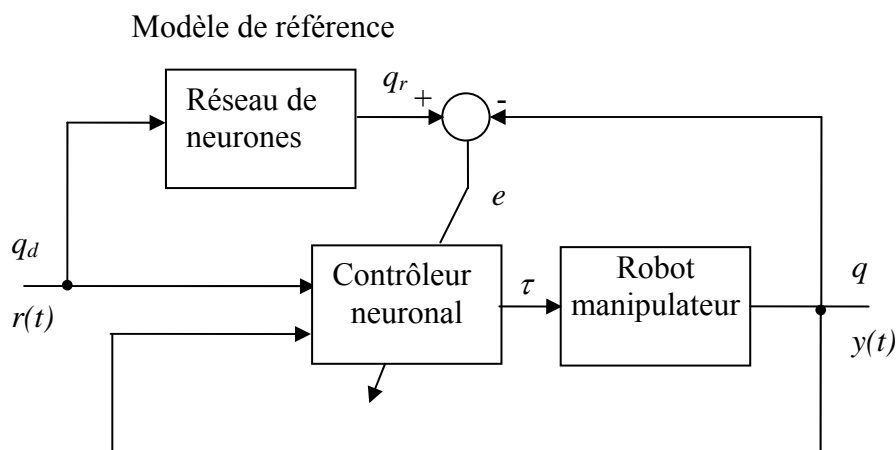


Figure 2.7 Commande avec modèle de référence

2.2.3.5 Commande à modèle interne (IMC) [HUN 91]

Dans ce schéma de commande (appelé aussi IMC Internal Model Control) deux réseaux de neurones sont employés dans une boucle de commande. Le premier réseau RN_1 , entraîné hors

ligne, fait une copie du modèle dynamique du système. Pendant l'opération en ligne, l'erreur entre le modèle et la sortie mesurée du système est utilisée comme boucle de retour qui attaque le réseau de neurones RN_2 représentant la dynamique inverse du même système comme il est montré sur la figure 2.8. Le rôle de RN_1 est de maintenir la sortie du processus égale au modèle RN_1 quelque soit les perturbations P . En effet le signal de retour n'est autre que les effets des perturbations P .

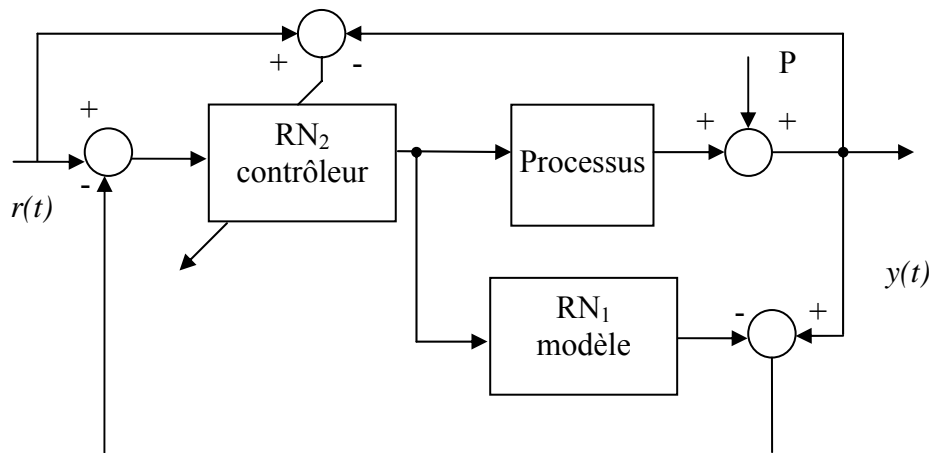


Figure 2.8 Commande à modèle interne

2.2.3.6 Commande adaptative à modèle de référence

La stratégie utilisée par cette approche exploite un modèle de référence défini par des couples entré/sortie $\{r(t), y(t)\}$. Ce modèle est utilisé pour spécifier la réponse idéale du système à commander. Le contrôleur du système quant à lui est implémenté pour forcer le système réel à égaler la réponse du modèle de référence. En se basant sur cette stratégie, Narendra [NAR 90] a proposé deux structures pour l'identification et le contrôle du système non linéaire, l'une directe et l'autre indirecte.

- La commande directe :

Dans cette structure (figure 2.9), les paramètres du contrôleur sont directement adaptés pour réduire l'erreur en sortie du système. Une fois la phase d'apprentissage achevée, l'action de contrôle est activée de sorte que la sortie du système suive la référence.

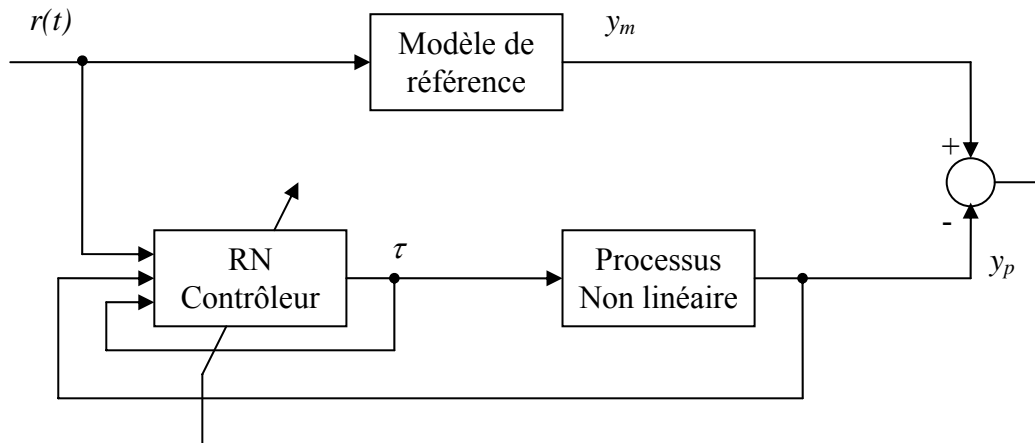


Figure 2.9. Commande adaptative directe

- La commande indirecte :

Dans cette structure, une identification et un contrôle sont effectués simultanément. Deux réseaux de neurones sont utilisés. Le premier, monté en parallèle avec le système réel assure la fonction d'identification, le deuxième réseau de neurones force la sortie du système réel à suivre celle du système de référence en considérant le modèle neuronal issu de l'identification au lieu du système lui-même. Cependant les paramètres temps, T_i pour l'identification et T_c pour le contrôle, doivent être choisis judicieusement.

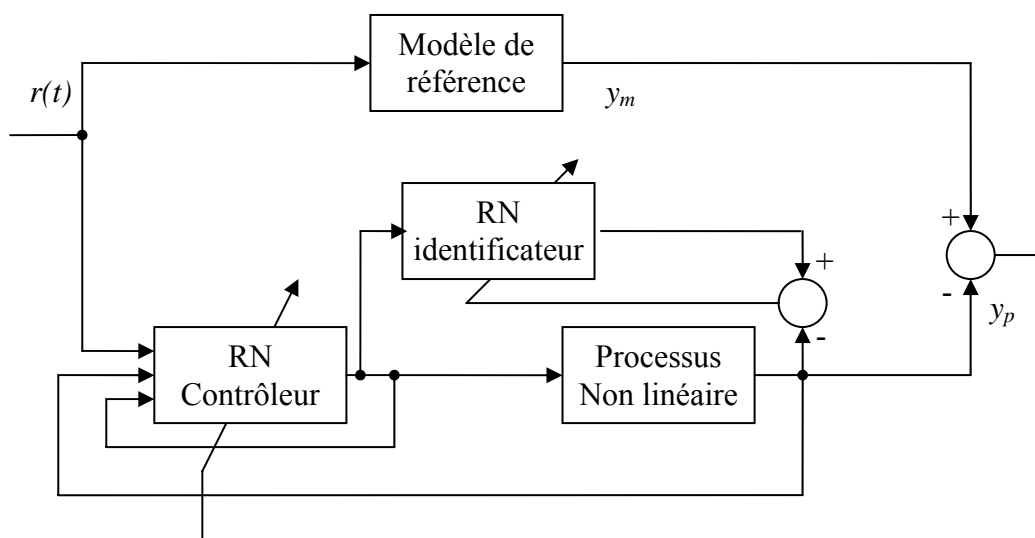


Figure 2.10. Commande adaptative indirecte

2.2.3.7 Contrôleur neuronal inverse avec une boucle de commande conventionnelle

Contrairement aux différents schémas précédents, cette structure de commande utilise la boucle d'une commande conventionnelle comme signal d'erreur à minimiser pour entraîner le modèle neuronal inverse [MIY 88] (figure 2.11). L'application ici de l'algorithme de rétro propagation peut se faire sans avoir besoin de connaître la matrice Jacobienne du système à commander. Un autre avantage de cette structure réside dans la possibilité de contrôler le système en ligne (adaptation en temps réel du réseau de neurones).

Le signal de commande du processus est donné par (voir figure 2.11) :

$$U = U_e + \hat{U} \quad (2.24)$$

Où \hat{U} est la sortie du réseau de neurones et U_e est le signal de la boucle d'erreur. Dans le cas idéal on désire avoir $U_e=0$. Donc en minimisant $U_e = U - \hat{U}$ on obtient :

$$\hat{U} = U \quad (2.25)$$

Cette équation représente en réalité le modèle dynamique inverse du système. La sortie de la boucle de commande conventionnelle U_e est utilisée pour ajuster les poids du réseau de neurones. Le choix le plus adéquat du contrôleur conventionnel qui assure la stabilité du système en régime transitoire est le contrôleur PD [JUN 96].

L'inconvénient de cette structure réside dans la difficulté du choix des gains de la boucle de commande qui dépend en réalité de la connaissance de la matrice d'inertie du robot. Donc le choix des gains influe directement sur la stabilité asymptotique (rapidité de la convergence de l'erreur). Généralement des gains relativement élevés sont choisis pour assurer les performances désirées (minimiser les effets de la non linéarité et les bruits).

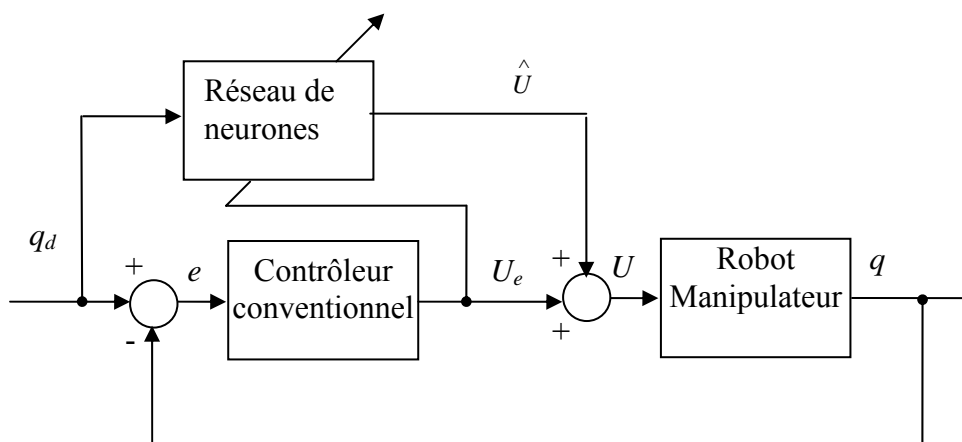


Figure 2.11 Commande neuronale avec une boucle de commande conventionnelle

2.2.3.8 Technique basée sur la modification de la trajectoire de référence

Cette technique de compensation est basée sur la modification de la trajectoire de référence pour compenser la dynamique du manipulateur [JUN 95b, 95c] (voir figure 2.12). Son avantage par rapport à la structure précédente est que le réseau de neurones peut être introduit sans toucher à la boucle de commande existante du système.

Il a été démontré [JUN 96] que cette structure de commande est plus performante que la précédente en termes de compensation et sensibilité aux perturbations des paramètres des contrôleurs.

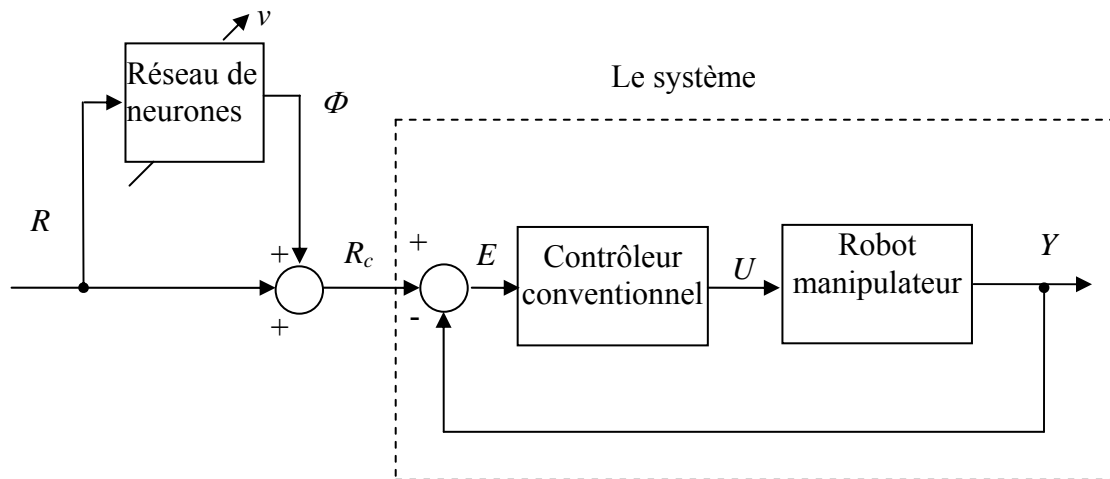


Figure 2.12. Commande avec modification de la trajectoire de référence.

2.2.3.9 Commande neuronale d'un robot basée sur son modèle dynamique

L'application des réseaux de neurones pour commander un processus décrit précédemment se base essentiellement sur l'identification du modèle dynamique inverse du procédé à contrôler. Cependant, il s'avère difficile pour certaines structures de commande d'obtenir le signal d'entraînement du réseau d'apprentissage d'un procédé inconnu à contrôler.

Dans le cas d'un robot manipulateur, l'obtention toujours possible, d'un modèle dynamique approximatif a permis de proposer dans la littérature des commandes dynamiques non linéaires [BOI 88] [SAM 87]. L'idée était d'introduire les réseaux de neurones artificiels autour de ce type de commande non pas pour l'apprentissage de son modèle dynamique inverse mais pour la compensation des éventuelles incertitudes (les perturbations et les non linéarités non modélisables) sur ce modèle.

Nous présentons dans ce qui suit les structures de commande neuronale proposées dans la littérature et qui utilisent le modèle dynamique approximatif du robot à commander comme élément qui fait partie de la commande globale. L'illustration de la différence entre ces schémas sera présentée à travers le rôle joué par le réseau de neurones dans chaque cas.

2.2.3.9.1 Equation dynamique du robot

Considérons un manipulateur à n degrés de liberté, en chaîne cinématique ouverte. Son modèle dynamique qui exprime les couples articulaires appliqués aux actionneurs en fonction des positions, vitesses et accélérations peut être décrit par l'équation suivante :

$$\tau(t) = A(q)\ddot{q} + B(q, \dot{q}) + \tau_f \quad (2.26)$$

où, $A(q) \in \mathfrak{R}^{m \times n}$, est la matrice d'inertie des axes (matrice définie positive), $B(q, \dot{q}) \in \mathfrak{R}^{m \times 1}$ qui englobe les termes de couple dus aux forces de Coriolis et centrifuge et des couples de gravité, τ_f représente les couples de frottements au niveau des articulations.

2.2.3.9.2 La loi de commande dynamique

La loi de commande par découplage non linéaire appliquée ici (voir figure 2.13) est donnée par l'équation suivante [BOI 88] :

$$\tau = \hat{A}(q)U + \hat{B}(q, \dot{q}) \quad (2.27)$$

Avec,

$$U = \ddot{q}_d + K_v(\dot{q}_d - \dot{q}) + K_p(q_d - q) \quad (2.28)$$

K_p et K_v sont des matrices diagonales définies positives de dimension $(m \times n)$.

Où, \hat{A}, \hat{B} sont respectivement les valeurs estimées de A et B .

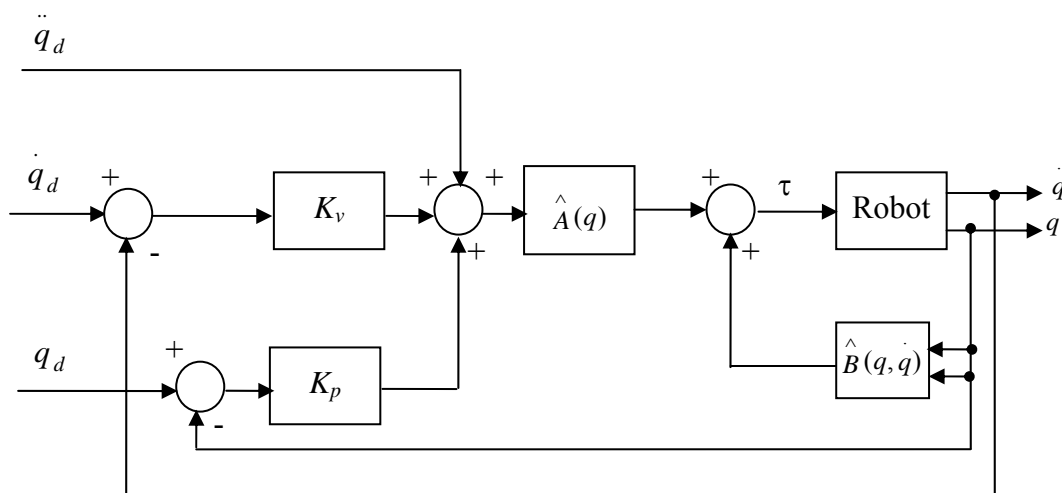


Figure 2.13. Structure classique de la commande dynamique

Ainsi et à partir des équations (2.26) - (2.28) le système en boucle fermée sera représenté par l'équation découplée et linéarisée suivante :

$$\ddot{e} + K_v \dot{e} + K_p e = \hat{A}^{-1} (\Delta A(q) \ddot{q} + \Delta B(q, \dot{q}) + \tau_f) \quad (2.29)$$

où, $e = q_d - q$

avec, $\Delta A(q) = A(q) - \hat{A}(q)$; et $\Delta B(q, \dot{q}) = B(q, \dot{q}) - \hat{B}(q, \dot{q})$;

Dans le cas idéal où le modèle dynamique est parfaitement connu c'est-à-dire, lorsque $\Delta A = \Delta B = 0$ et que les frottements secs et visqueux sont supposés nuls ($\tau_f = 0$), l'équation (2.29) sera de la forme:

$$v = \ddot{e} + K_v \dot{e} + K_p e = 0 \quad (2.30)$$

Donc, le but de la commande dynamique est de découpler et linéariser l'ensemble du système en boucle fermée. Cependant il existe toujours des incertitudes sur le modèle dynamique du robot. Généralement la réponse idéale donnée par l'équation (2.30) est très difficile à vérifier. En effet l'équation réelle qui régit le système est bien l'équation (2.29). Les termes de droite de cette équation, représentant les incertitudes sur le modèle dynamique du robot, peuvent dégrader complètement les performances de cette commande

Donc, en pratique la commande dynamique classique est une commande non robuste. Une solution à ce problème consiste à choisir des grands gains dans les boucles de commande, mais une conséquence évidente de l'utilisation de grands gains dans la commande est l'augmentation de la bande passante du système en boucle fermée [DUM 87][SAM 87]. Ce qui a pour effet néfaste de rendre le système plus sensible aux bruits générés par les capteurs utilisés. De plus un problème de stabilité du système peut se poser au moment de sa discrétisation, car les gains limites assurant sa stabilité en boucle fermée sont liés directement à la valeur de la période d'échantillonnage utilisée dans la commande [DUM 87].

Pour remédier à tous ces problèmes, des structures de commande neuronales permettant la compensation des incertitudes sur les modèles dynamiques sans avoir recours au choix de gains excessivement élevés sont proposées dans la littérature, thème que nous allons aborder dans la partie suivante de ce chapitre.

2.2.3.9.3 Technique de compensation adaptative d'OZAKI [OZA 91]

Cette structure de commande propose un compensateur neuronal non linéaire appliqué au suivi de trajectoire. Le schéma de commande introduit deux réseaux RN1 et RN2, le premier pour identifier la matrice d'inertie A et le deuxième est destiné à l'identification en ligne de la matrice regroupant les forces centrifuge et de Coriolis, les dynamiques non modélisées ainsi que les perturbations externes (figure 2.14). Ces réseaux sont adaptatifs en ligne en utilisant un signal d'entraînement des réseaux de neurones issu d'un filtre PD implémenté.

La démarche proposée consiste à identifier chaque élément de la matrice d'inertie A sur une sortie du réseau RN₁ et chaque élément du vecteur B sur une sortie du réseau RN₂. Cette solution devient de plus en plus complexe avec le nombre de degrés de liberté du robot, de plus dans cette structure les algorithmes de calcul sont développés dans un contexte centralisé de l'information est traitée dans une seule unité de commande qui rend en général son calcul long. Une solution à ce problème a été proposée par [BOU 98] pour décentraliser cette

structure en la décomposant en plusieurs sous systèmes interconnectés, mais cela demande un investissement important et une méthodologie particulière.

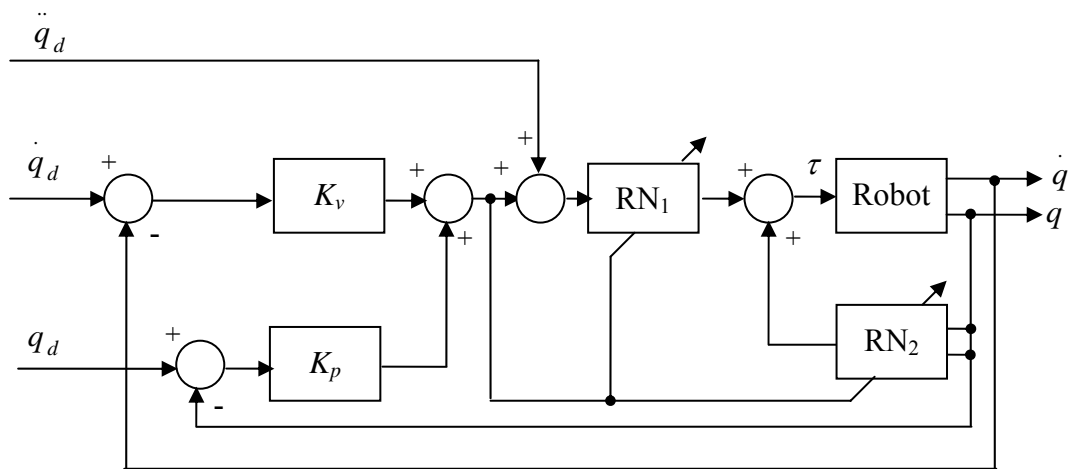


Figure 2.14. Technique de compensation adaptative d'OZAKI

2.2.3.9.4 Technique de compensation d'ISHIGURO[ISH 92]

Ce schéma de commande exploite une fonction coût qui minimise l'erreur entre le couple actuel τ et le couple τ_m obtenu à partir du modèle dynamique inverse du robot. Pour compenser les incertitudes sur la dynamique du robot manipulateur, un signal issu du réseau implémenté est ajouté à la loi de commande dynamique (voir figure 2.15).

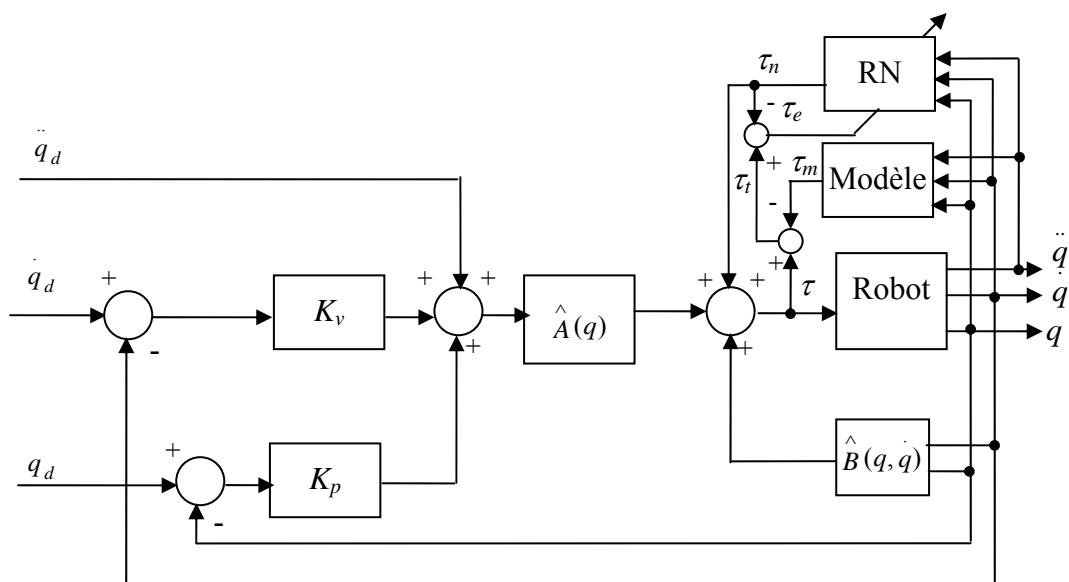


Figure 2.15. Structure neuronale proposée par ISHIGURO

Cette structure de commande exploite le modèle dynamique de référence du robot pour générer le signal d'entraînement du réseau, qui rend cette structure assez compliquée et

gourmande en temps de calcul. Une autre structure plus simple permettant la réalisation du même objectif mais avec un signal d'entraînement du réseau beaucoup plus simple a été aussi proposée dans la littérature. Nous la détaillerons dans le paragraphe suivant.

2.2.3.9.5 Commande dynamique avec réseau de neurones auxiliaire [JUN 95b]

Contrairement au schéma de commande précédent, cette structure de commande exploite une fonction coût qui minimise l'équation de la dynamique de l'erreur. Pour compenser les incertitudes sur la dynamique du robot manipulateur, un signal issu du réseau implémenté est introduit autour de la commande dynamique comme il est montré sur la figure 2.16 suivante :

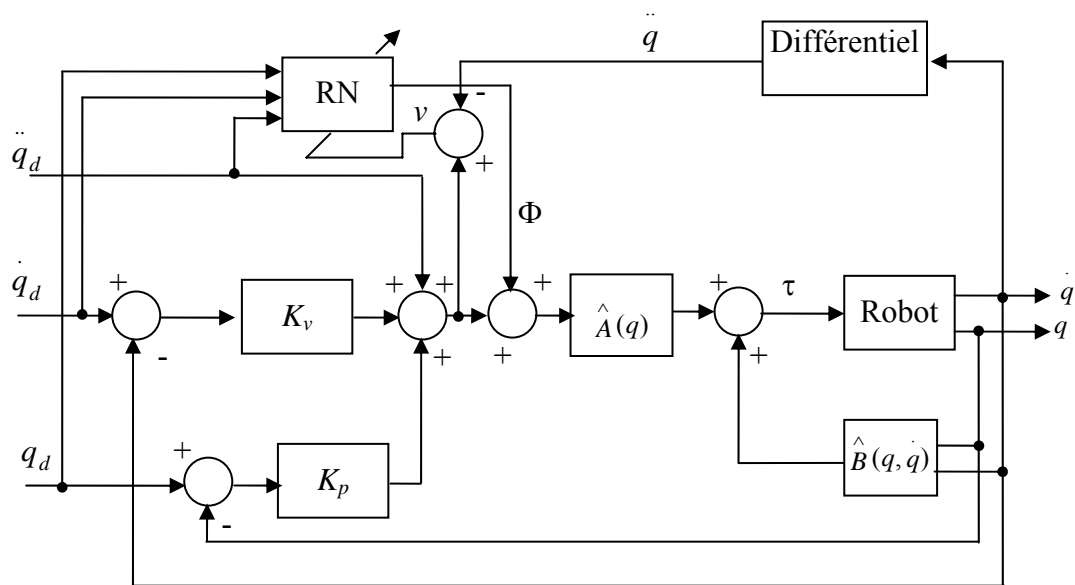


Figure 2.16 Commande dynamique avec réseau de neurones auxiliaire

Nous remarquons que le signal d'entraînement de ce réseau de neurones est plus simple à implémenter et à mettre en œuvre en temps réel par rapport à celui du schéma de commande proposé par ISHIGURO.

2.3 La Logique Floue

2.3.1 Introduction

La logique floue a été introduite en 1965 par Zadeh. Ses principes ont été appliqués en 1974 par Mamdani à la construction d'un premier régulateur flou. Mais depuis quelques années uniquement que la commande par logique floue a connu un essor remarquable dans des applications telles que les systèmes de purification de l'eau, les chaînes de fabrication du ciment, etc...

Il s'est avéré pour les automaticiens que la satisfaction des performances d'une commande nécessite d'identifier le mieux possible le système à commander. Chose qui n'est pas facile pour les systèmes qui présentent des non linéarités difficiles à identifier et à modéliser. La commande par logique floue proposait de remplacer des commandes basées sur l'identification par des techniques s'appuyant sur le savoir et l'expérience humaine plutôt que sur des équations. Aujourd'hui la commande par logique floue constitue un outil précieux pour la commande des systèmes difficilement identifiables ou dont les paramètres subissent des variations brutales.

Le principe du réglage par la logique floue se rapproche de la démarche humaine dans le sens où les variables à traiter ne sont pas des variables logiques, mais des variables linguistiques. En outre, ces variables linguistiques sont traitées à l'aide de règles qui font référence à une certaine connaissance du comportement du système à régler. En effet, sur la base de ce principe, les systèmes flous et l'ensemble des règles qui leurs sont rattachées forment une classe de fonctions non linéaires, qui permet de construire par apprentissage une très large classe de modèles et de correcteurs.

Partant de ces idées fondamentales, on peut dire que la logique floue est une logique fondée sur des variables pouvant prendre, outre les valeurs "vrai" ou "faux" des valeurs intermédiaires avec une certaine probabilité, en faisant intervenir les principes de la logique floue.

2.3.2 Ensemble flou

Très souvent les classes des objets rencontrés dans le monde physique ne possèdent pas des critères d'appartenance bien définis. Contrairement aux ensembles booléens dont la fonction caractéristique ne prend que des valeurs possibles 0 et 1, le concept d'un ensemble flou permet de considérer des classes d'objets par l'introduction d'une fonction caractéristique (fonction d'appartenance des objets à la classe) prenant des valeurs courantes entre 0 et 1. En logique floue, un ensemble flou est la généralisation de la notion de l'ensemble en logique classique. La propriété d'appartenance est définie par la fonction caractéristique suivante :

$$\begin{aligned} \mu_A : U_x &\rightarrow [0 \ 1] \\ x &\rightarrow \mu_A(x) \end{aligned}$$

Où, U_x est l'univers de discours de x . $\mu_A(x)$ est appelée fonction d'appartenance.

L'ensemble flou noté A est défini par l'ensemble des couples $(x, \mu_A(x))$ qui peut être noté comme suit :

$$A = \{(x, \mu_A(x)) / x \in U_x\} \quad \text{forme générale}$$

$$A = \int_{U_x} \mu_A(x) / x \quad \text{forme continue}$$

$$A = \sum_{x \in U_x} \mu_A(x) / x \quad \text{forme discrète}$$

Avec,

$\mu_A(x)$: Fonction d'appartenance.

U_x : Ensemble de référence

x : valeur prise par une variable floue donnée.

Remarque :

La sommation et l'intégration dans les équations précédentes sont pour l'union des couples $(x, \mu_A(x))$; elles n'indiquent pas la sommation ou l'intégration classiques. De même le symbole '/' n'implique pas une division.

L'univers de discours est l'ensemble qui contient tous les sous-ensembles mis en jeu. Parmi les fonctions d'appartenance les plus utilisées, on peut citer les fonctions triangulaires, trapézoïdales, Gaussiennes, sigmoïde...etc. [JAN 95]. La figure 2.17 montre quelques unes.

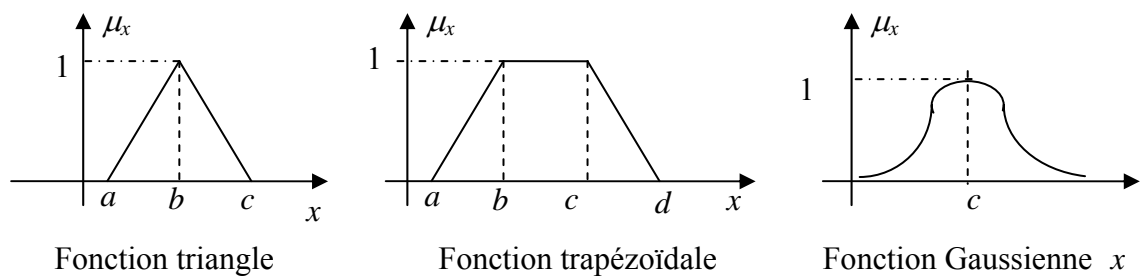


Figure 2.17 Exemple de fonctions d'appartenance

2.3.3 Opération sur les sous ensembles flous [JAN 95][YIN 00]

La manipulation des sous ensembles flous nécessite de définir certaines opérations élémentaires.

Soit A et B deux sous ensembles flous dans U_x , caractérisé respectivement par $\mu_A(x)$ et $\mu_B(x)$:

- Inclusion :

On dit que A est inclus dans B ($A \subseteq B$), si et seulement si :

$$\mu_A(x) \leq \mu_B(x), \forall x \in U_x$$

- Union (conjonction) :

L'union de deux ensembles flous A et B est un ensemble flou C . Noté $C=A \cup B$ ou bien $C=A \text{ OR } B$, dont la fonction d'appartenance est relié à A et B par :

$\mu_C(x) = \max(\mu_A(x), \mu_B(x)) = \mu_A(x) \vee \mu_B(x)$: Opérateur *OR* de Zadeh

Ou bien

$\mu_C(x) = \min(\mu_A(x) + \mu_B(x), 1)$: Opérateur *OR* de Lukasiewicz.

- Intersection (disjonction)

L'intersection de deux ensembles flous *A* et *B* est l'ensemble Flou *C* notée $C=A \wedge B$ ou bien *A AND B*, dont la fonction d'appartenance est reliée à celles de *A* et *B* par :

$\mu_C(x) = \min(\mu_A(x), \mu_B(x)) = \mu_A(x) \wedge \mu_B(x)$: Opérateur *AND* de Zadeh

Ou bien

$\mu_C(x) = \mu_A(x) \times \mu_B(x)$: Opérateur *AND* produit de Larsen

- Complément

Le complément de l'ensemble flou *A*, noté \bar{A} (not *A*) est défini par :

$$\mu_{\bar{A}} = 1 - \mu_A(x), \quad \forall x \in U_x$$

2.3.4 Raisonnement linguistique par des règles floues :

On peut représenter et raisonner sur des expressions linguistiques ou des mots du langage naturel tout en utilisant les ensembles et les relations floues. Si on considère par exemple l'expression suivante [BOU 02]:

« Si c'est l'erreur actuelle est négative grande et la variation de l'erreur est positive grande, alors il faut changer la commande en l'incrémentant pour qu'elle ait la propriété d'être négative petite ».

Des notions clefs suivantes interviennent dans cette expression :

- Des variables linguistiques : Erreur, variation d'erreur
- Propriété des variations linguistiques : positive grande, négative grande ...
- Des connecteurs linguistiques : « et », « ou »
- Propositions :
 - Erreur négative grande
 - Variation de l'erreur positive grande
- Condition : Si ...alors...

L'objectif recherché est de représenter l'expression précédente à l'aide de la logique floue en introduisant une représentation symbolique des différentes notions citées ci-dessus. Cette expression peut être quantifiée si on associe des ensembles et des relations floues à la représentation symbolique.

2.3.5 Variable linguistique [NAK 97]

Une variable linguistique est une variable dont les valeurs sont des mots ou des phrases exprimées dans une langue naturelle ou un langage artificiel. Une variable linguistique est définie par le quadruplet $\langle X, L(X), U_X, M_x \rangle$

Où,

X est le nom de la variables linguistiques, comme, âge, erreur, température... etc.

$L(X) = \{L_1, \dots, L_n\}$ est l'ensemble des valeurs linguistiques que peut prendre la variable X . Par exemple $L(X) = \{\text{froid, frais, tiède, chaud}\}$ peut être défini pour la variable température. $L(X)$ est aussi appelé ensemble de symboles associés à la variable X .

U_X est le domaine physique associé à la variable X , il est aussi appelé univers de discours.

M_x : fonction sémantique, qui attribue à chaque valeur de X son sens (signification floue). Ce dernier est caractérisé par un ensemble flou défini sur l'univers de discours U_X .

2.3.6 Structure générale d'un régulateur flou

La structure d'un régulateur flou est représentée par la figure 2.18. Le régulateur comprend le système à réguler et le régulateur flou. Ce dernier fournit à sa sortie le signal de commande U . Généralement il reçoit à son entrée l'erreur et la variation de l'erreur entre la consigne et la sortie mesurée.

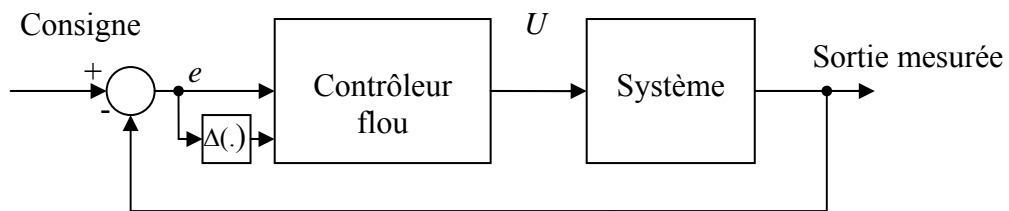


Figure 2.18 Schéma d'un système en boucle fermée avec régulateur flou

2.3.7 Description des éléments d'un régulateur flou

Les différents blocs constituant un régulateur flou sont représentés par la figure 2.19 suivante :

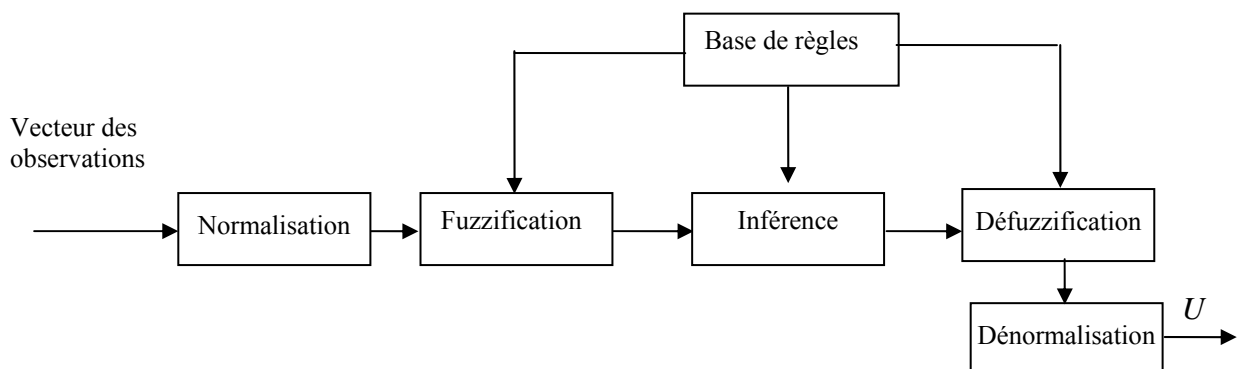


Figure 2.19 Schéma d'un régulateur flou

- **Module de normalisation**

Le vecteur d'entrée du module de normalisation qu'on appellera vecteur d'observations ou de mesures, a ses composantes dépendantes des états du processus. Ce module transforme les valeurs physiques actuelles du vecteur d'observation dans un intervalle normalisé, souvent entre -1 et 1.

- **Bloc de Fuzzification :**

C'est une interface « numérique linguistique » qui permet de transformer les grandeurs observées et normalisées sur le processus (valeur numériques) en un ensemble de valeurs floues (valeur linguistique). La réalisation de ce bloc se fait en utilisant des fonctions d'appartenance de type triangulaire, trapézoïdale...etc.

- **Bloc d'inférence**

C'est un mécanisme de décision qui, en manipulant les implications floues et le bloc des règles, établit une décision. En pratique on peut faire appel à plusieurs stratégies d'inférence, telles que : la méthode de Mamdani (max, min), la méthode de Larsen (max, Prod) et la méthode de Sugeno (Somme, Prod) [PAS 98]

- **Base de règles :**

C'est une base de connaissances qui contient les règles floues décrivant le comportement du système.

- **Bloc de défuzzification**

C'est une interface "Linguistique - Numérique" qui transforme la partie floue issue du raisonnement flou en valeurs numériques directement exploitables par le processus. Il existe plusieurs méthodes pour réaliser cette opération, souvent c'est la méthode du centre de gravité qui est utilisée [PAS 98].

- **Bloc de dénormalisation :**

Le module de défuzzification délivre généralement un signal de commande normalisé qui appartient à l'intervalle [-1 1]. Le bloc de dénormalisation permet de transformer le signal du domaine normalisé au domaine physique.

2.3.8 Modèles flous (système d'inférence flou)

Grâce à la propriété d'approximation des fonctions, les systèmes à logique floue permettent la modélisation de ces connaissances sous forme de modèles flous. Considérons un système de n valeurs mesurées $x_1...x_n \in X_1...X_n$ respectivement, et une valeur de contrôle $y \in Y$. Il existe plusieurs types de modèle parmi eux on cite :

- Modèle de Sugeno
- Modèle de Mamdani.

a- Modèle de Sugéno

Les règles de type Sugéno sont des règles dont les prémisses sont symboliques tandis que les conclusions sont numériques. Les règles sont de la forme :

$$Si(X_1 \text{ est } A_1^{(r)} \dots \text{et } X_n \text{ est } A_n^{(r)}) \text{ ALORS } (Y = f_r(x_1 \dots x_n)) \quad r=1 \dots k$$

Dans ce type de régulateur les conséquences des règles contiennent des fonctions numériques $(f_m)_{m=1 \dots k}$ qui dépendent des valeurs actuelles des variables mesurées $(x_i)_{i=1 \dots n}$

b- Modèle de Mamdani [JAN 95]

Dans ce modèle les conséquences des règles sont de types linguistiques basées sur l'expérience de l'opérateur humain, exprimées sous la forme :

$$Si(X_1 \text{ est } A_1^{(r)} \dots \text{et } X_n \text{ est } A_n^{(r)}) \text{ ALORS } (Y = B^{(r)}) \quad r=1 \dots k$$

Généralement les ensembles $X_i \in \mathfrak{R}$, et les ensembles flous A_i sont déterminés par des fonctions triangulaires sigmoïdales...etc.

La figure 2.20 suivante est une illustration montrant comment deux règles floues d'un système d'inférence flou de Mamdani (max, min) génère la sortie totale z soumise à deux entrées x et y.

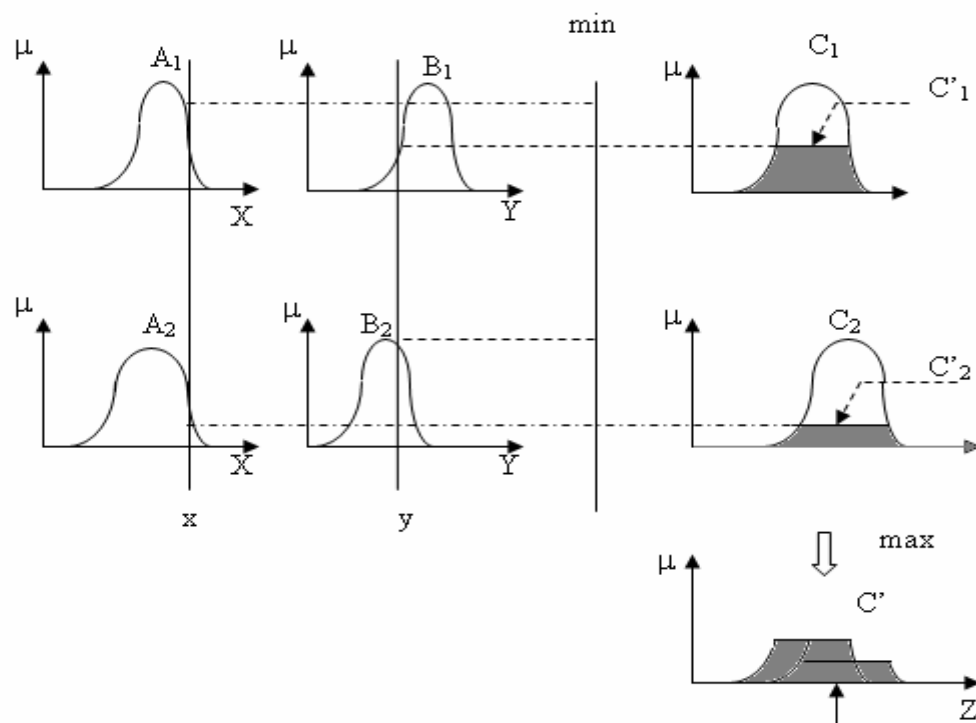


Figure 2.20. Système d'inférence flou de Mamdani utilisant min et max pour les opérateurs flou AND et OR respectivement

La valeur exacte de la sortie peut être déduite en utilisant la méthode du centre de gravité définie par :

$$z_{CG} = \frac{\int_Z \mu_c(z) z dz}{\int_Z \mu_c(z) dz} \quad (2.31)$$

Où, $\mu_c(z)$ est la fonction d'appartenance résultante de la sortie.

Dans un cas plus général un modèle flou de Mamdani se réalise comme suit :

• Base des règles

La base des règles est constituée de règles de la forme suivante :

R_r : if X_1 est A_1^r et...et X_n est A_n^r alors η est B^r ($r=1,\dots,k$)

Où, A_1^r, \dots, A_n^r, B^r sont les termes linguistiques attribués aux fonctions

d'appartenance $\mu_{1,r}, \dots, \mu_{n,r}, \mu_r$, correspondants à la subdivision floue des ensembles

$X_1 \dots X_n$ et Y_n respectivement.

Les K règles correspondent à la fonction définie par morceaux $\eta = \lambda(X_1, \dots, X_n)$ donné par :

$$\eta = \begin{cases} B^1 & \text{si } X_1 = A_1^1 \text{ et...et } X_n = A_n^1 \\ \vdots & \vdots \\ B^k & \text{si } X_1 = A_1^k \text{ et...et } X_n = A_n^k \end{cases}$$

• Moteur d'inférence

Il combine les règles floues et les entrées après fuzzification pour tirer des conclusions. Cette opérations regroupe : la conjonction des prémisses, l'implication floue et l'agrégation des règles.

Au départ, cette opération fait évoluer chaque règle de manière indépendante. Pour chaque règle R_r , le degré d'appartenance des variables mesurées est déterminé en calculant les valeurs $\mu_{v,r}(x_v)$ pour $v=1,\dots,n$, qui représente le degré auquel x_v correspond à la fonction floue

$\mu_{v,r}$.

La conjonction des prémisses est réalisée par l'opération Min, comme suit :

$$\alpha_r = \text{Min}\{\mu_{1,r}(x_1), \dots, \mu_{n,r}(x_n)\}$$

Avec,

α_r est le degré d'appartenance de l'antécédent de la règle R_r .

La sortie de la règle R_r est l'ensemble flou des variables de commande obtenu en décomposant l'ensemble flou de la sortie μ_r de la règle R_r au degré d'appartenance déterminé par α_r . Ainsi la règle R_r induit l'ensemble flou suivant :

$$\mu_{x_1, \dots, x_n}^{sortie(R_r)} : Y \rightarrow [0,1], y \rightarrow \text{Min} \{ \mu_{1,r}(x_1), \dots, \mu_{n,r}(x_n), \mu_r(y) \}$$

La combinaison de toutes les règles en un seul ensemble flou en utilisant l'opérateur max(union), est évaluée comme suit :

$$\mu_{x_1, \dots, x_n}^{sortie(R_r)} : Y \rightarrow [0,1], \eta \rightarrow \max_{r \in \{1, \dots, k\}} \{ \text{Min} \{ \mu_{1,r}(x_1), \dots, \mu_{n,r}(x_n), \mu_r(\eta) \} \}$$

• Défuzzification

Cet ensemble flou résultant doit être transformé en une valeur de commande par l'interface de défuzzification en utilisant par exemple la méthode de centre de gravité :

Dans la littérature, plusieurs schémas de commande utilisant les modèles flous décrits précédemment, et qui tendent à améliorer les performances désirées sont proposés. Certains de ces schémas ont été décrits dans [LIN 00]. Pour notre cas, vu les applications envisagées en contrôle force/position nous nous sommes intéressés particulièrement aux contrôleurs PID flous qu'on propose de présenter dans les paragraphes suivants.

2.3.9 La logique floue et les contrôleurs PID

Ces dernières années les contrôleurs flous, spécialement les contrôleurs proportionnel-intégral-dérivé (PID) ont été largement utilisés dans la commande des processus industriels grâce à leurs nature heuristique, leur simplicité et leur efficacité pour les deux types de systèmes linéaire et non linéaire. Grâce à leur propriété de non linéarité de leurs gains de commande, les contrôleurs PID flous peuvent s'adapter aux différentes situations par rapport aux contrôleurs PID conventionnels si leur non linéarité est convenablement exploitée. Plusieurs schémas de commande PID flous sont proposés dans la littérature, mais avant de présenter certains de ces schémas que nous avons jugés intéressants pour notre cas, nous allons montrer une propriété fondamentale de la commande à gains variables des contrôleurs PID flous, en analysant la structure parallèle proposée dans [XU 00] et qui est donnée par la figure 2.21 suivante:

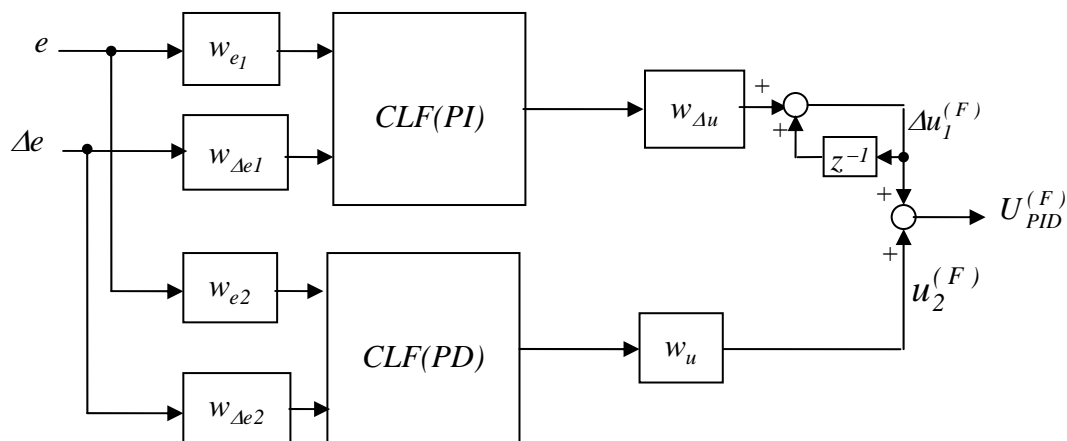


Figure 2.21: Structure générale d'un contrôleur PID flou

Dans la figure 2.21:

- e et Δe sont respectivement l'erreur et la variation de l'erreur qui représentent les entrées des deux contrôleurs,
- w_{e1} , w_{e2} , $w_{\Delta e1}$ et $w_{\Delta e2}$ sont des gains de normalisation,
- Exposant (F) des variables des sorties signifie flou.

Comme il est montré sur la figure 2.21 ci-dessus, cette structure combine donc un contrôleur PD flou, qui génère à sa sortie une commande de type position ($u_2^{(F)}$), et un contrôleur PI flou, qui génère à sa sortie une commande de type vitesse ($\Delta u_1^{(F)}$), permettant ainsi de réaliser un contrôleur global de type PID flou. Les fonctions d'appartenance $\mu(\cdot)$ des entrées et sorties sont respectivement de type trapézoïdal et singleton données par la figure 2.22 suivante :

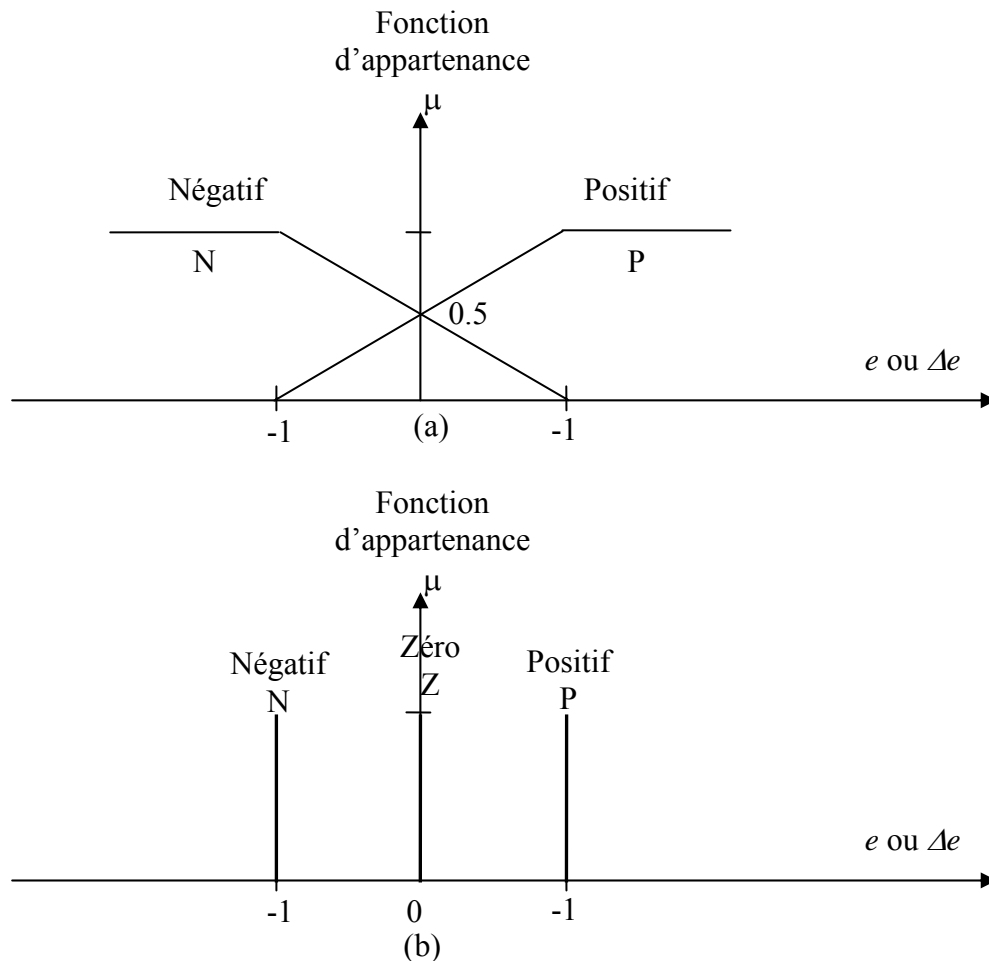


Figure 2.22 : Définitions graphiques des entrées et sorties des ensembles flous du contrôleur PID flou : (a) : entrée et (b) sortie

Avec ces ensembles flous la base des règles de chaque sortie est limitée donc à quatre règles résumées par les tableaux suivants :

		Δe	
	Δu	N	P
e	N	N	Z
	P	Z	P

Tableau1 : Base des règles
contrôleur PI flou

		Δe	
	u	N	P
e	N	N	Z
	P	Z	P

Tableau2 : Base des règles
contrôleur PD flou

Pour simplifier les calculs on pose $w_{e1} = w_{e2} = w_e$ et $w_{\Delta e1} = w_{\Delta e2} = w_{\Delta e}$.

Dans ce cas si on utilise la méthode d'inférence produit de Larsen avec l'opérateur AND de Zadeh, et l'opérateur OR de Lukasiewicz, et si on utilise la méthode de centre de gravité pour la défuzzification, alors la sortie de chaque contrôleur flou peut être exprimée dans l'univers de discours par les expressions suivantes:

$$\Delta u_1^{(F)} = \frac{w_{\Delta u1}}{4 - 2 \max(w_{e1}|e|, w_{\Delta e1}|\Delta e|)} (w_{e1}e + w_{\Delta e1}\Delta e) = \frac{w_{\Delta u}}{4 - 2\alpha} (w_e e + w_{\Delta e} \Delta e) \quad (2.32)$$

$$u_2^{(F)} = \frac{w_{u2}}{4 - 2 \max(w_{e2}|e|, w_{\Delta e2}|\Delta e|)} (w_{e2}e + w_{\Delta e2}\Delta e) = \frac{w_u}{4 - 2\alpha} (w_e e + w_{\Delta e} \Delta e) \quad (2.33)$$

Où

$$\alpha = \max(w_{e1}|e|, w_{\Delta e1}|\Delta e|) = \max(w_{e2}|e|, w_{\Delta e2}|\Delta e|) = \max(w_e|e|, w_{\Delta e}|\Delta e|)$$

La sortie totale du contrôleur flou sera donc :

$$U_{PID}^{(F)} = \sum_0^k \Delta u_1^{(F)} + u_2^{(F)} = \sum_0^k \frac{w_{\Delta u} w_{\Delta e}}{4 - 2\alpha} \left(\Delta e + \frac{\Delta t}{w_{\Delta e} / w_e \Delta t} e \right) + \frac{w_u w_e}{4 - 2\alpha} \left(e + \frac{w_{\Delta e} \Delta t}{w_e \Delta t} \Delta e \right) \quad (2.34)$$

Si on choisit

$$\begin{aligned} K_c^{(F)} &= \frac{w_{\Delta u} w_{\Delta e}}{4 - 2\alpha}, \\ T_i^{(F)} &= \frac{w_{\Delta e}}{w_e} \Delta t, \\ K_c^{(F)} \frac{T_d^{(F)}}{T_i^{(F)}} &= \frac{w_u w_e}{4 - 2\alpha} \Delta t, \end{aligned} \quad (2.35)$$

Alors la sortie du contrôleur flou dans (2.34) peut être réécrite comme suit :

$$U_{PID}^{(F)} = \sum_0^k K_c^{(F)} \left(\Delta e + \frac{\Delta t}{T_i^{(F)}} e \right) + K_c^{(F)} \frac{T_d^{(F)}}{T_i^{(F)}} \left(e + T_i^{(F)} \frac{\Delta e}{\Delta t} \right) \quad (2.36)$$

Si on pose

$$\frac{\Delta e}{\Delta t} \approx \dot{e} \text{ alors,}$$

L'équation (2.36) peut être approximée comme suit

$$\begin{aligned} U_{PID}^{(F)} &\approx \int_0^{k.\Delta t} K_c^{(F)} \left(de + \frac{e}{T_i^{(F)}} dt \right) + K_c^{(F)} \frac{T_d^{(F)}}{T_i^{(F)}} \left(e + T_i^{(F)} \frac{de}{dt} \right) \\ &= \int_0^{k.\Delta t} K_c^{(F)} \dot{e} dt + \int_0^{k.\Delta t} \frac{K_c^{(F)}}{T_i^{(F)}} e dt + \frac{K_c^{(F)} T_d^{(F)}}{T_i^{(F)}} \left(e + T_i^{(F)} \dot{e} \right) \end{aligned} \quad (2.37)$$

Sachant qu'un contrôleur PID linéaire à temps continu est défini par l'expression suivante :

$$U_{PID}^{(C)} = \int_0^t K_c \dot{e} dt + \int_0^t \frac{K_c}{T_i} e dt + \frac{K_c T_d}{T_i} (e + T_i \dot{e}) \quad (2.38)$$

En comparant les équations (2.37) et (2.38) on peut donc conclure que le contrôleur PID flou (2.37) est un contrôleur PID nonlinéaire avec un gain proportionnel variable.

2.3.9.1 Relation entre un contrôleur PI et un contrôleur PD

Dans la pratique, la commande d'un système avec un contrôleur PID plein peut devenir dans certains cas indésirable. En effet le terme dérivé peut amplifier les bruits donc il devrait être évité si la sortie du système est plutôt bruité, d'un autre côté, le terme intégral peut causer une réponse lente au système et des dépassements, donc il ne devrait pas être inclus dans certaines applications des contrôleurs PID. En pratique on a souvent recours à des contrôleurs de type PI ou PD que l'on considère non pas comme des contrôleurs PID incomplets, mais plutôt comme des contrôleurs de classes séparées. En effet plusieurs études dans la littérature traitent les contrôleurs PI, PD, et PID séparément et différemment. Dans les applications où on a besoin d'éviter les bruits générés par certains capteurs et lorsque on a besoin d'assurer un bon suivi en régime permanent c'est les contrôleurs de type PI qui sont recommandés. Cependant dans les processus utilisant les contrôleurs PID flou, une fois les fonctions d'appartenance sont choisies et la base des règles est construite, le problème suivant qui peut se poser c'est le réglage des gains de normalisation. Le choix de ces gains peut dramatiquement influencer sur la dynamique du système global en boucle fermée. Pour remédier à ce problème des schémas de réglage de ces paramètres sont proposés dans la littérature. Nous présentons dans ce qui suit, deux structures que nous avons jugé intéressantes pour notre cas proposées dans [ZHA 03].

2.3.9.2 Contrôleur PI flou avec un autoréglage de son paramètre de sortie :

Développer une méthode de réglage généralisé est une tâche très difficile car le calcul des valeurs optimales des paramètres de réglage nécessite de connaître les objectifs exigés de la commande aussi bien qu'un modèle fixe pour le contrôleur. Pour remédier à ce problème, une technique d'autoréglage du gain de sortie d'un contrôleur PI flou a été proposée. Le schéma bloc est représenté par la figure 2.23 suivante :

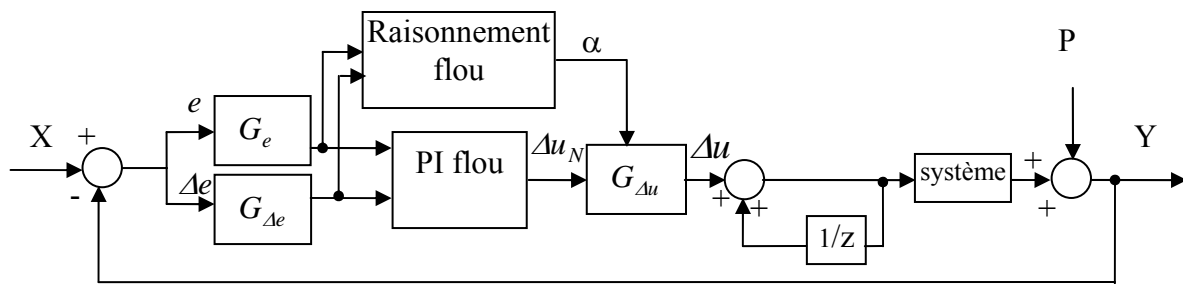


Figure 2.23 : Système en boucle fermée avec un contrôleur PI flou auto réglé

Basé sur ce mécanisme de réglage, la variation de l'incrément de sortie du contrôleur Δu est obtenue par l'équation suivante :

$$\Delta u = (\alpha \cdot G_{\Delta u}) \cdot \Delta u_N \quad (2.39)$$

Ainsi le gain $G_{\Delta u}$ ne reste pas fixe lorsque le contrôleur est opérationnel ; plutôt il se modifie à chaque période d'échantillonnage grâce à un gain d'adaptation α obtenu en ligne par un raisonnement flou utilisant l'erreur et la variation de l'erreur à chaque période d'échantillonnage qui, en réalité change la surface à la sortie du contrôleur flou [ZHA 03].

2.3.9.3 Contrôleur PI conventionnel adaptatif par logique floue

Dans ce schéma le signal de commande est généré selon un réglage en ligne des gains proportionnel et intégral basé sur la fonction de transfert discrète suivante :

$$H(z) = K_p + K_i T_s \frac{z}{z-1} = K_p \left(1 + \frac{T_s}{T_i} \frac{z}{z-1} \right) \quad (2.40)$$

Où K_p est le gain proportionnel, K_i est le gain intégral, $T_i = K_p / K_i$ est la constante intégral du temps, et T_s est la période d'échantillonnage. Le schéma proposé est donné par la figure suivante :

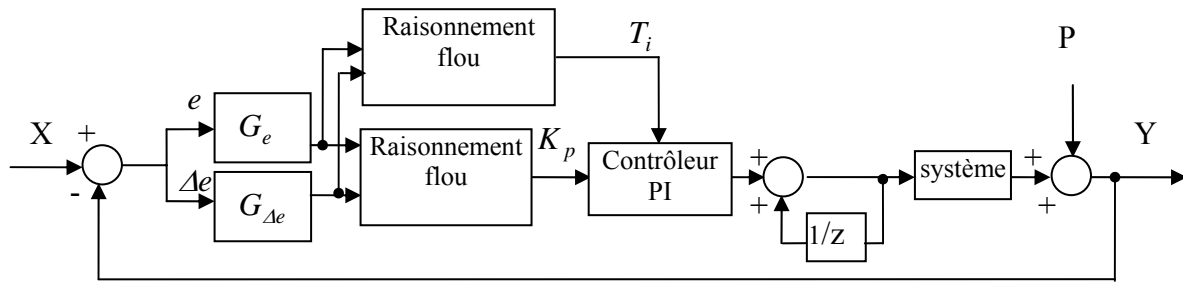


Figure 2.24 Système en boucle fermée avec un contrôleur conventionnel adaptatif par la logique floue

Dans cette structure de commande nous utilisons deux systèmes de raisonnement flou. L'un possède deux entrées $e(k)$ et $\Delta e(k)$ et une sortie K_p et l'autre a les mêmes entrées mais avec une sortie T_i . Ainsi K_i s'obtient en calculant K_p/T_i .

2.4 Conclusion

Dans ce chapitre, nous avons présenté des notions sur les réseaux de neurones et la logique floue qui seront utilisés dans les chapitres suivants.

Dans une première partie, nous avons décrit les réseaux de neurones artificiels. Nous avons mis en évidence leurs structures, leurs caractéristiques ainsi que leurs applications dans le domaine de la commande à travers une présentation de différentes structures les plus citées dans la littérature. Leur capacité d'approximation de fonctions non linéaires leur a ouvert un grand champ d'application, en particulier, dans les problèmes de commande des systèmes non linéaires tels que les robots manipulateurs.

Dans la deuxième partie de ce chapitre nous avons introduit la notion de la logique floue et son application dans le domaine du contrôle de processus. Grâce à la nature linguistique des variables manipulées, il est possible d'exploiter l'expérience humaine dans l'élaboration des stratégies de commande. Nous avons présenté deux types de régulateurs flous, le régulateur de Sugeno et celui de Mamdani. Nous avons donné plus d'attention au modèle de Mamdani car il s'adapte bien aux applications envisagées par le fait que les prémisses et les conclusions sont des variables floues. Par la suite nous avons présenté des structures de commande non linéaires de type PID flou que nous avons jugé intéressant pour nos applications en commande force/position.

Compte tenu des caractéristiques intéressantes des réseaux de neurones pour l'identification et la commande de processus, et la logique floue pour l'exploitation de l'expérience de l'opérateur humain dans la commande, nous proposons dans les chapitres suivants de nouvelles approches des commandes force/position basées sur ces concepts (les réseaux de neurones et la logique floue), permettant de traiter le problème de la commande en effort de la classe des tâches du suivi de contour dans des milieux contraints.

Chapitre 3

**Structures de commande intelligentes
parallèle force/position**

3.1 Introduction

La commande force/position est un concept bien adapté pour traiter le problème de la commande des tâches de suivi de trajectoire dans des milieux contraints. Dans le chapitre 1, nous avons vu que les structures parallèles force/position conventionnelles [CHI 93, 94] s'adaptent bien à ce type de tâches, car elles permettent de contrôler à la fois la position et l'effort dans une même direction de l'espace de la tâche sans aucun mécanisme de sélection. La stratégie de ces schémas de commande est basée sur la dominance de la boucle de commande en effort sur la boucle de commande en position le long de la direction contrainte de la tâche où les interactions avec l'environnement se produisent.

Les performances de ces schémas de commande parallèle sont facilement atteintes si le modèle du robot manipulateur exploité dans la commande (commande parallèle basée sur le modèle dynamique [CHI 93]) est parfaitement connu ou bien lorsque la tâche désirée permet de négliger la quasi-totalité de la dynamique du robot (commande non basé sur le modèle dynamique [CHI 94]). L'inconvénient s'est que dans la pratique le modèle dynamique complet d'un robot manipulateur n'est jamais connu avec exactitude alors que la commande parallèle qui néglige la dynamique du robot est limitée à des tâches quasi-statiques du robot. Ces inconvénients sont particulièrement critiques lorsqu'on cherche à obtenir de bonnes performances en termes d'optimisation des réponses et d'adaptativité vis-à-vis de la diversité des types de tâches et d'environnements des milieux contraints. Il apparaît donc difficile dans ces conditions de concevoir des commandes classiques performantes pour des systèmes inconnus ou connus partiellement.

Pour remédier à ces contraintes et tirer profit des caractéristiques intéressantes des structures parallèles proposés dans la littérature [CHI 93, 94], nous proposons dans ce chapitre d'associer à ces schémas de commande des compensateurs issus des concepts de l'intelligence artificielle qui sont les réseaux de neurones et la logique floue pour améliorer leurs performances en suivi de trajectoire et en réponse aux efforts appliqués. Dans ce chapitre, deux nouvelles stratégies de commande intelligentes parallèles force/position seront proposées. La première est basée sur le modèle dynamique du manipulateur. Quant à la deuxième, elle est basée sur des contrôleurs de type PID.

3.2 Equation dynamique du robot en contact avec son environnement

L'équation dynamique d'un bras manipulateur à n degrés de liberté dans l'espace articulaire est donnée par :

$$\tau = A(q)\ddot{q} + b(q, \dot{q})\dot{q} + g(q) + \tau_f + J^T F_e \quad (3.1)$$

où,

q, \dot{q}, \ddot{q} sont respectivement les positions, vitesses et accélérations articulaires,

$A(q) \in \mathfrak{R}^{n \times n}$, est la matrice d'inertie des axes (matrice définie positive),

$b(q, \dot{q}) \in \mathfrak{R}^{n \times 1}$ englobe les termes de couple dus aux forces de Coriolis et centrifuges,

$g(q) \in \mathfrak{R}^{n \times 1}$ est le vecteur des couples des termes de gravité,

$F_e \in \mathfrak{R}^{n \times 1}$ est le vecteur des forces généralisées que le manipulateur exerce sur son environnement,

$J^T \in \mathcal{R}^{n \times n}$ est la matrice Jacobienne transposée du manipulateur, et τ_f représente les frottements au niveau des articulations.

Pour simplifier l'écriture, posons :

$$C(q, \dot{q}) = b(q, \dot{q}) \dot{q} + g(q)$$

Donc, l'équation 3.1 peut être réécrite ainsi :

$$\tau = A(q) \ddot{q} + C(q, \dot{q}) + \tau_f + J^T \cdot F_e \quad (3.2)$$

La relation entre la vitesse articulaire et la vitesse cartésienne est donnée par l'équation suivante :

$$\dot{X} = J(q) \dot{q} \quad (3.3)$$

Où $J(q)$ représente la matrice Jacobienne du manipulateur. La dérivation de l'équation (3.3) permet d'obtenir l'accélération dans l'espace cartésien :

$$\ddot{X} = J(q) \ddot{q} + \dot{J} \dot{q} \quad (3.4)$$

L'équation du mouvement du robot dans l'espace articulaire peut être encore représentée dans l'espace cartésien par la relation suivante :

$$\ddot{q} = J(q)^{-1} (\ddot{X} - \dot{J} \dot{q}) \quad (3.5)$$

La substitution de la relation (3.5) dans (3.2) donne :

$$\tau = A(q) \cdot J^{-1} \cdot (\ddot{X} - \dot{J} \cdot \dot{q}) + C(q, \dot{q}) + \tau_f + J^T \cdot F_e \quad (3.6)$$

Les forces sont liées aux couples articulaires par la relation suivante :

$$\tau = J^T \cdot F \quad (3.7)$$

Le modèle du robot dans l'espace cartésien est donné donc par :

$$F = D \ddot{X} + H + F_f + F_e \quad (3.8)$$

Où, $D = J^{T-1} \cdot A \cdot J^{-1}$, $H = J^{T-1} C - D \cdot \dot{J} \cdot J^{-1} \cdot \dot{X}$ et $F_f = J^{T-1} \cdot \tau_f$

3.3. Structure de commande dynamique parallèle force/position [CHI 93]

La commande force/position proposée par CHIAVERINI est basée sur le concept d'une parallélisation de la boucle de commande en effort avec la boucle de commande en position décrite dans le chapitre 1. Pour compenser la dynamique du robot manipulateur régi par un

Avec,

$$U_p = M_d \ddot{X}_d + K_v (\dot{X}_d - \dot{X}) + K_p (X_d - X) \quad (3.11)$$

$$U_f = K_f (F_d - F_e) + K_i \int_0^t (F_d - F_e) d\tau$$

Où K_p , K_v , K_f et K_i sont des matrices diagonales scalaires de réglage de type λI . La combinaison des équations (3.8)-(3.11) donne l'équation dynamique suivante :

$$M_d \ddot{E}_p + K_v \dot{E}_p + K_p E_p + K_f E_f + K_i \int_0^t E_f d\tau = \hat{D}^{-1} (\Delta D \ddot{X} + \Delta H + F_f + \Delta F_e) \quad (3.12)$$

Où,

$$\Delta D = D - \hat{D}, \Delta H = H - \hat{H}, \Delta F_e = F_e - \hat{F}_e, E_p = X_d - X \text{ et } E_f = F_d - F_e.$$

Dans le cas idéal (modèle dynamique connu parfaitement et frottements considérés nuls) on a $\Delta D = \Delta H = F_f = \Delta F_e = 0$, et l'équation du système global en boucle fermée est donnée par la relation suivante :

$$M_d \ddot{E}_p + K_v \dot{E}_p + K_p E_p + K_f E_f + K_i \int_0^t E_f d\tau \quad (3.13)$$

A l'état d'équilibre, l'erreur en position reste constante pendant que l'erreur en effort est nulle. Cela implique que la boucle de commande en effort est prédominante sur la boucle de commande en position [CHI 93].

3.3.1 Analyse des performances en présence d'un milieu élastique

L'objectif recherché est d'analyser les performances de la commande proposée lorsque le robot interagit avec l'environnement. Un modèle exact du contact entre le manipulateur et l'environnement est généralement difficile à obtenir sous forme analytique, à cause de la complexité des phénomènes physiques impliquée pendant l'interaction.

Il est tout à fait possible d'avoir recours à un modèle simple et signifiant, en comptant sur la robustesse du système de commande pour atténuer les effets de la modélisation incorrecte. L'élasticité totale, due au capteur d'effort lié à l'effecteur et l'environnement, est estimée à travers la compliance du plan. Les effets des frottements sont supposés négligeables par rapport à la compliance élastique qui est prédominante.

Lorsque le robot est en contact avec son environnement, le modèle des forces de contact considéré, est donné par (seul le cas de mouvements en translation et les forces linéaires appliquées est étudié ici, c'est-à-dire que $n=3$) :

$$F_e = K_e(X - X_e) \quad (3.14)$$

Où

- X est la position du l'organe terminal,
- X_e est la position de l'environnement au repos,
- et K_e est une matrice de raideur constante et symétrique de dimension (3x3) (cas de notre étude, robot à trois axes).

On note ici que l'équation (3.14) est valable uniquement lorsque le manipulateur est en contact avec son environnement et que toutes les grandeurs de cette équation sont exprimées dans le même repère de référence.

Grâce à l'équation (3.14) on peut écrire :

$$E_f = K_e \cdot E_p + F_d - K_e(X_d - X_e) \quad (3.15)$$

La substitution de l'équation (3.15) dans (3.13) permet d'obtenir l'équation suivante:

$$\begin{aligned} M_d \ddot{E}_p + K_v \dot{E}_p + (K_p + K_e \cdot K_f) E_p + K_i \cdot K_e \int_0^t E_p d\tau \\ = -K_f F_d + K_f \cdot K_e \cdot (X_d - X_e) - K_i \int_0^t F_d d\tau + K_i \cdot K_e \int_0^t (X_d - X_e) d\tau \end{aligned} \quad (3.16)$$

On a donc un système du troisième ordre et sa stabilité peut être analysée en terme de stabilité du système non forcé donné par l'équation suivante [CHI 93] :

$$M_d \ddot{X} + K_v \dot{X} + (K_p + K_e \cdot K_f) X + K_i \cdot K_e \int_0^t X d\tau = 0 \quad (3.17)$$

Le robot étant en contact avec son environnement, pour analyser l'équation (3.17) couplée par la matrice K_e , on pose les hypothèses suivantes :

- l'effort de contact est orthogonal au plan selon l'un des vecteurs $(X-X_e)$; donc, la base de l'image de la matrice K_e ($\text{Im}(K_e)$) est un vecteur unitaire \vec{n} orthogonal au plan et le $\text{rang}(K_e)=1$.
- Tous les autres vecteurs $(X - X_e)$ liés au plan et qui ne contribuent pas aux forces de contact sont tangents au plan et la base du noyau de la matrice K_e ($N(K_e)$) est constituée de deux vecteurs unitaires $\begin{pmatrix} \rightarrow \\ t_1, t_2 \end{pmatrix}$.

Sous ces conditions les colonnes de la matrice

$$R_{rot} = (n \ t_1 \ t_2) \quad (3.18)$$

forment une base orthonormée de \mathfrak{R}^3 . Et comme la matrice K_e est symétrique elle peut être décomposée dans le repère de référence comme suit:

$$K_e = R_{rot} \text{diag}(k_e, 0, 0) R_{rot}^T = k_e n n^T \quad (3.19)$$

Avec R_{rot} la matrice de rotation du repère attaché au plan par rapport au repère de référence.

Posons x_n, x_1, x_2 les composantes de X dans le repère attaché au plan défini par:

$$\begin{pmatrix} x_n \\ x_1 \\ x_2 \end{pmatrix} = R_{rot}^T X \quad (3.20)$$

Sachant que

$$R_{rot} R_{rot}^T = n n^T + t_1 t_1^T + t_2 t_2^T = I \quad (3.21)$$

L'équation (3.17) peut s'écrire dans le repère attaché au plan comme suit:

$$\begin{aligned} n \left[m_d \ddot{x}_n + k_v \dot{x}_n + (k_p + k_f k_e) x_n + k_i k_e \int_0^t x_n d\tau \right] \\ + t_1 \left[m_d \ddot{x}_1 + k_v \dot{x}_1 + k_p x_1 \right] + t_2 \left[m_d \ddot{x}_2 + k_v \dot{x}_2 + k_p x_2 \right] = 0 \end{aligned} \quad (3.22)$$

Grâce au fait que n, t_1, t_2 sont orthonormés l'équation (3.22) est équivalente au système d'équations scalaires suivant:

$$\begin{aligned} m_d \ddot{x}_n + k_v \dot{x}_n + (k_p + k_f k_e) x_n + k_i k_e \int_0^t x_n d\tau &= 0 \\ m_d \ddot{x}_1 + k_v \dot{x}_1 + k_p x_1 &= 0 \\ m_d \ddot{x}_2 + k_v \dot{x}_2 + k_p x_2 &= 0 \end{aligned} \quad (3.23)$$

Donc le système est découplé dans l'espace des variables x_n, x_1, x_2 .

En appliquant le critère de Routh au système d'équations (3.23), la condition nécessaire et suffisante pour que le système soit stable est que:

$$k_p, k_v, m_d, k_i > 0, k_f \geq 0 \text{ et } k_v(k_p + k_f k_e) > m_d k_i k_e \quad (3.24)$$

Avec, k_v, k_p, k_f, k_i, k_e et m_d respectivement les éléments des matrices K_v, K_p, K_f, K_i, K_e et M_d .

Remarques :

- la vérification de l'ensemble de ces relations dépend du choix des paramètres de réglage ainsi que de la connaissance de la raideur de l'environnement.
- Dans le cas d'un mouvement en espace libre ($F_d = F_e = 0$), l'équation (3.13) est découplée et elle devient du second ordre. La condition nécessaire et suffisante de stabilité est vérifiée dans ce cas pour $m_d, k_p, k_v > 0$.

3.3.1.1. Cas d'un environnement de position mal connue

Dans le cas où la position de l'environnement est parfaitement connue, l'équation (3.16) peut se réécrire en remplaçant X_d par X_e ($X_d = X_e$), avec X_e position de l'environnement.

Si l'environnement est de position mal connue, on aura :

$$X_d = X_e + \delta X_e = X_e' \quad \text{avec } \delta X_e \text{ l'incertitude sur } X_e$$

Dans ce cas l'équation (3.16) s'écrira:

$$\begin{aligned} M_d \ddot{E}_p + K_v \dot{E}_p + (K_p + K_e K_f) E_p + K_i K_e \int_0^t E_p d\tau \\ = -K_f F_d - K_i \int_0^t F_d d\tau - M_d \delta \ddot{X}_e - K_v \delta \dot{X}_e - K_p \delta X_e \end{aligned} \quad (3.25)$$

L'application de la transformée de Laplace à l'équation (3.25) nous permet d'écrire:

$$\begin{aligned} E_p(p) = - \left[M_d p^3 + K_v p^2 + (K_p + K_e K_f) p + K_e K_i \right]^{-1} \\ \times \left[(K_f p + K_i) F_d + (M_d p^3 + K_v p^2 + K_p p) \delta X_e(p) \right] \end{aligned} \quad (3.26)$$

Avec $\delta X_e(p)$ l'entrée et $E_p(p)$ la sortie

Si on suppose que la perturbation est un échelon ($\delta X_e(p) = \frac{1}{p}$), et que la consigne $F_d = 0$,

le point d'équilibre du système (3.26) sera:

$$E_p(\infty) = \lim_{p \rightarrow 0} p E_p(p) = 0 \Rightarrow X_\infty \rightarrow X_e \Rightarrow F_e = K_e (X_\infty - X_e) \rightarrow 0$$

Ce résultat prouve que le système est compliant et que la loi de commande en effort domine la loi de commande en position.

3.3.1.2. Cas d'un environnement curviligne

L'étude de la stabilité développée dans le paragraphe 3.3.1 n'est valable que dans le cas du suivi de contour sur une surface plane. Dans le cas où la surface est curviligne, la matrice de rotation R_{rot} (équation (3.18)) ne sera plus constante, elle deviendra variable dans le temps et dépendante de la variation du point du contact.

Vu la complexité du problème, nous supposons que la surface est lisse et connue analytiquement. Dans ce cas le principe de la stabilité locale peut être appliqué en chaque voisinage d'un point d'équilibre et le plan tangent sera une bonne approximation en ces points. La matrice R_{rot} pourra être considérée constante en chaque point d'équilibre.

Une analyse quasi-statique peut être développée par rapport à ces points d'équilibres successifs, si la trajectoire désirée est choisie telle que R_{rot} varie lentement par rapport à l'ensemble du système dynamique.

Sous ces hypothèses les conditions de stabilité précédentes restent valables. Dans ce cas les modèles des efforts de contact donnés par les équations (3.14) et (3.19) peuvent se réécrire sous la forme [CHI 93]:

$$F_e = K_e(X_e(X))(X - X_e) \quad (3.27)$$

Avec,

$$K_e(X_e) = k_e n(X_e) n^T(X_e) \quad (3.28)$$

3.3.2 Problématique

Les performances en terme de réponses en position et en effort de la structure de commande parallèle force/position proposée par CHIAVERINI reposent essentiellement sur une connaissance parfaite du modèle dynamique du robot. Cependant, les performances de cette structure de commande peuvent être dégradées si la connaissance exacte des modèles du robot et de son environnement n'est pas vérifiée c'est-à-dire si ΔD , ΔH , ΔF_e et F_f sont différentes de zéro (terme de droite de l'équation (3.12)).

En effet il y a toujours des incertitudes sur le modèle dynamique du robot exploité dans la commande qui peuvent en réalité dégrader les performances des réponses en position et en effort particulièrement lorsqu'on est en présence des incertitudes sur les caractéristiques de l'environnement (sa position et sa raideur).

Pour montrer comment les réponses en effort et en position peuvent se dégrader en présence de ces incertitudes, nous proposons d'analyser le cas critique d'un environnement mal connu en position et en raideur et en présence des incertitudes dues à une méconnaissance de la dynamique du robot. Pour se faire nous proposons d'analyser l'équation (3.26) en considérant les perturbations données par le terme de droite de l'équation (3.12) qu'on note:

$$\Delta M = \hat{D}^{-1} (\Delta D \ddot{X} + \Delta H + F_f + \Delta F_e)$$

Dans ce cas l'équation (3.25) devient donc :

$$\begin{aligned} M_d \ddot{E}_p + K_v \dot{E}_p + (K_p + K_e K_f) E_p + K_i K_e \int_0^t E_p d\tau \\ = -K_f F_d - K_i \int_0^t F_d d\tau - M_d \delta \ddot{X}_e - K_v \delta \dot{X}_e - K_p \delta X_e + \Delta M \end{aligned} \quad (3.29)$$

Pour simplifier cette analyse, on suppose que le repère de référence et celui attaché au plan soient parallèles ($R_{rot}=I$). Dans ce cas l'équation (3.29) s'écrit selon une direction contrainte comme suit :

$$\begin{aligned} m_d \ddot{e}_p + k_v \dot{e}_p + (k_p + k_e k_f) e_p + k_i k_e \int_0^t e_p d\tau \\ = -k_f f_d - k_i \int_0^t f_d d\tau - m_d \delta \ddot{x}_e - k_v \delta \dot{x}_e - k_p \delta x_e + \Delta m \end{aligned} \quad (3.30)$$

L'application de la transformée de Laplace à l'équation (3.30) nous permet d'écrire:

$$\begin{aligned} e_p(p) = & -\frac{(k_f p + k_i)}{m_d p^3 + k_v p^2 + (k_p + k_f k_e) p + k_e k_i} f_d \\ & - \frac{(m_d p^3 + k_v p^2 + k_p p)}{m_d p^3 + k_v p^2 + (k_p + k_f k_e) p + k_e k_i} \delta x_e(p) \\ & + \frac{p}{m_d p^3 + k_v p^2 + (k_p + k_f k_e) p + k_e k_i} \Delta m(p) \end{aligned} \quad (3.31)$$

Sachant que $e_p = x_d - x = x_e - x$, l'effort de contact f_e peut donc être déduit en multipliant les deux membres de l'équation (3.31) par k_e . L'analyse de l'équation (3.31) montre que le terme dû à la dynamique mal identifiée ($\Delta m(p)$) influe comme entrée perturbatrice sur l'erreur e_p donc de même sur la réponse en effort. On remarque aussi que si la raideur k_e de l'environnement varie le placement des pôles de cette équation varie aussi.

Cette variation peut engendrer des réponses oscillatoires qui peuvent être pénalisantes et même nuisibles sur un robot manipulateur si le réglage des paramètres du contrôleur en effort reste fixe. On note ici qu'en absence de perturbations ($\Delta m(p)=0$), le système reste toujours compliant vis-à-vis d'une perturbation δx_e choisi ici de type échelon (l'erreur statique $e_p(\infty) = f_e(\infty) = 0$ pour $f_d = 0$). Cette caractéristique est due à la présence d'un intégrateur dans la boucle de commande en effort.

Cependant en présence de perturbation non linéaire ($\Delta m(p)$), l'erreur statique peut prendre des valeurs importantes. Prenons l'exemple de perturbations régies par une fonction parabolique ($\Delta m(p) = \frac{I}{p^3}$). Dans ce cas l'erreur statique de l'équation (3.31) prend la valeur $e_p(\infty) = f_e(\infty) \rightarrow \infty$. Donc on peut conclure que la compliance du système dépend directement de la caractéristique de la fonction régissant les perturbations sur le système. Dans la partie implémentation nous montrons en simulation les effets de toutes ces perturbations sur les réponses en position et en effort.

L'analyse menée ci-dessus nous a montré donc deux principaux problèmes à résoudre pour cette structure de commande dynamique parallèle force/position :

- le problème des incertitudes affectant le modèle dynamique du robot.
- le problème lié aux incertitudes sur la raideur de l'environnement en présence d'une méconnaissance de sa position.

Nous proposons dans ce qui suit deux techniques de compensation issues de l'intelligence artificielle (réseaux de neurones et la logique floue), permettant de compenser les incertitudes affectant à la fois la dynamique du robot et son environnement.

3.4 Structure neuronale de la commande dynamique parallèle force/position [FER 05]

Dans la pratique les performances d'une commande dynamique sont affectées par les incertitudes des modèles exploités dans la commande. Pour remédier à ce problème, plusieurs structures de commande dynamique utilisant des compensateurs neuronaux ont été proposées dans la littérature et qui traitent le cas de tâches du robot manipulateur en espace libre (voir chapitre deux précédent). La différence entre ces structures de commande proposées réside dans le rôle joué par le réseau de neurones dans la commande du système et la manière dont il est entraîné pour atteindre les performances du suivi de la trajectoire désirée.

Les structures de commande les plus populaires sont celles qui utilisent les réseaux de neurones entraînés en ligne et qui génère un couple auxiliaire permettant de compenser les incertitudes sur le modèle nominale exploité dans la commande. La différence entre les solutions proposées réside dans la position du contrôleur neuronale auxiliaire autour de la commande dynamique classique. La complexité de la mise en œuvre du réseau de neurones autour de cette commande varie d'une structure à une autre. La plus simple et la plus efficace est celle proposée dans [JUN 95b](voir figure 2.16). Cette structure exploite l'équation de la dynamique de l'erreur en position du système en boucle fermée pour entraîner en ligne le réseau de neurones.

Nous nous sommes inspirés de cette structure de commande pour développer une nouvelle approche neuronale appliqué à la structure de commande dynamique parallèle force/position de CHIAVERINI. Le réseau de neurones que nous proposons dans cette partie est conçu pour générer une commande auxiliaire qui vient modifier la commande parallèle force/position précédente (équation (3.10)) dans le but de compenser les perturbations sur l'ensemble du système dues à une mauvaise identification du modèle dynamique du robot en contact avec son environnement. L'idée donc (voir figure 3.2) est de générer ϕ à la sortie du réseau de neurones qui vient s'ajouter parallèlement aux deux lois de commande en force et en position.

La dimension du vecteur de sortie de ce réseau dépend du nombre de degrés de liberté de la tâche (dans notre cas il est de dimension 3, voir annexe A).

La loi de commande donnée par l'équation (3.9) devient donc:

$$F = \hat{D} M_d^{-1} (U + \phi) + \hat{H} \quad (3.32)$$

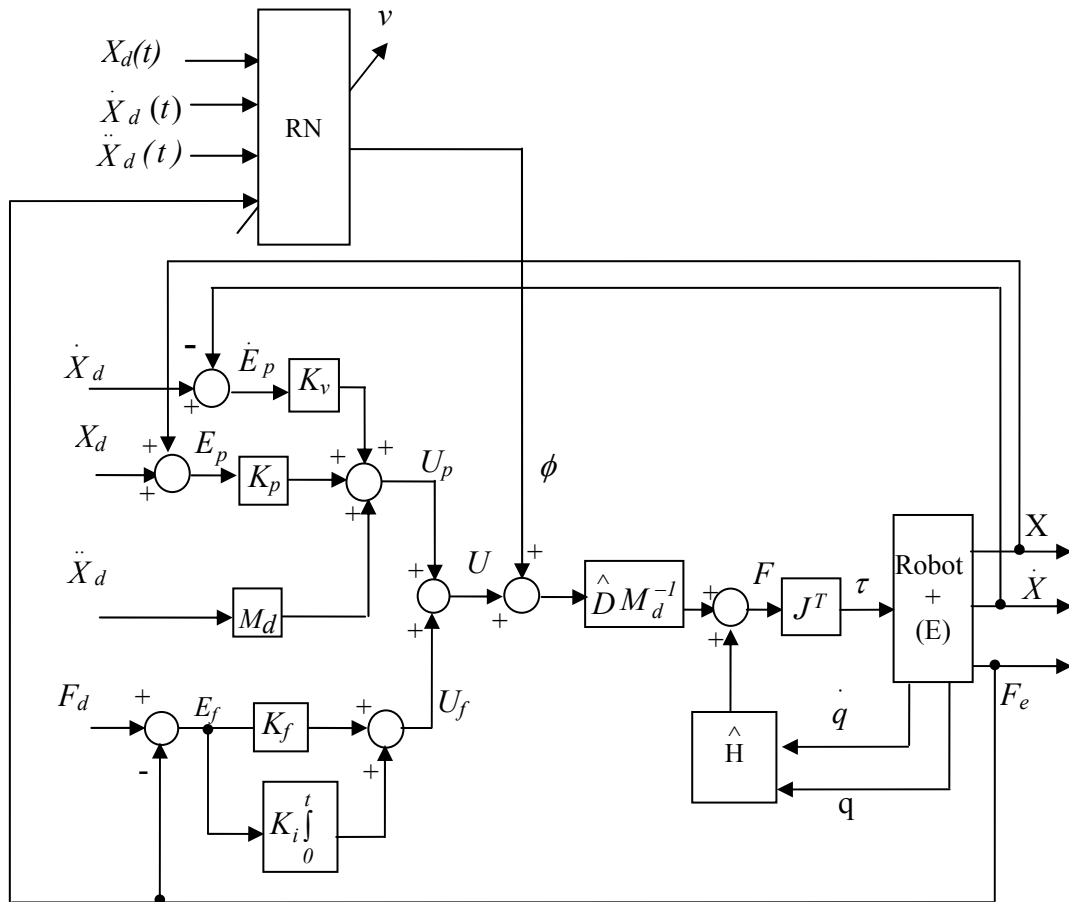


Figure 3.2 : Structure neuronale de la commande parallèle force/position

La combinaison des équations (3.32) et (3.8) donne l'équation en boucle fermée suivante :

$$M_d \ddot{E}_p + K_v \dot{E}_p + K_p E_p + K_f E_f + K_i \int_0^t E_f d\tau = \hat{D}^{-1} (\Delta D \ddot{X} + \Delta H + F_f + F_e) - \phi \quad (3.33)$$

L'objectif recherché est de générer ϕ pour faire tendre l'équation (3.33) vers zéro.

Si on pose

$$v = M_d \ddot{E}_p + K_v \dot{E}_p + K_p E_p + K_f E_f + K_i \int_0^t E_f d\tau \quad (3.34)$$

Alors, dans le cas idéal lorsque $v=0$ on aura :

$$\phi = \hat{D}^{-1} (\Delta D \ddot{X} + \Delta H + F_f + F_e) \quad (3.35)$$

On remarque que l'équation (3.35) est non linéaire et dépend de la position, de la vitesse et de l'accélération de l'organe terminal ainsi que des efforts de contact entre le robot et son environnement. Les efforts de contact, la position, la vitesse et l'accélération seront les

entrées du réseau. La loi de commande ϕ générée par le réseau de neurones est basée sur la minimisation d'un critère quadratique dépendant du signal v (équation (3.34)).

Le réseau proposé (figure 3.3) est à trois couches : une couche d'entrée , une couche de sortie, linéaires et une couche cachée intermédiaire non linéaire de type sigmoïde bilatérale qui renvoie des valeurs sur l'intervalle $[-1 \ 1]$ donné par l'équation suivante :

$$f(\cdot) = \frac{1 - \exp(-(\cdot))}{1 + \exp(-(\cdot))} \quad (3.36)$$

L'entrée du réseau est donc le vecteur :

$$X_{RN} = \begin{bmatrix} X_d^T(t) & \dot{X}_d^T(t) & \ddot{X}_d^T(t) & F_e^T(t) \end{bmatrix} \quad (3.37)$$

D'après la figure 3.3 la sortie ϕ_k peut s'écrire en fonction des entrées sous la forme :

$$\phi_k = \left[\sum_{j=1}^{n_H} w_{jk}^2 \frac{\left(1 - \exp\left(-\left(\sum_{i=1}^{n_I} x_{di} w_{ij}^1 + b_j^1 \right) \right) \right)}{\left(1 + \exp\left(-\left(\sum_{i=1}^{n_I} x_{di} w_{ij}^1 + b_j^1 \right) \right) \right)} \right] + b_k^2 \quad (3.38)$$

Où :

n_I est le nombre d'entrées,

n_H est le nombre de neurones cachés,

x_{di} est le $i^{\text{ième}}$ élément du vecteur d'entrées X ,

w_{ij}^1 sont les poids entre le $i^{\text{ième}}$ neurone de la couche d'entrée et le $j^{\text{ième}}$ neurone de la couche cachée

w_{jk}^2 sont les poids du $j^{\text{ième}}$ neurone de la couche cachée et le $k^{\text{ième}}$ neurone de la couche de sortie.

b_j^1 est le biais du $j^{\text{ième}}$ neurone caché et b_k^2 est le biais du $k^{\text{ième}}$ neurone du sortie.

Remarques

- En absence d'une règle générale sur le choix du nombre de neurones de la couche cachée [HAY 99], le nombre de neurones minimal de cette couche est choisi expérimentalement et il est de six.
- Le réseau est adaptatif en ligne et ses poids initiaux sont choisis nuls, aucune étape d'apprentissage n'est imposée.

La loi d'adaptation des poids de la relation (3.38) est obtenue en minimisant la fonction quadratique dépendante du signal v de l'équation (3.34) donnée par l'équation suivante :

$$J = \frac{1}{2} v^T v \quad (3.39)$$

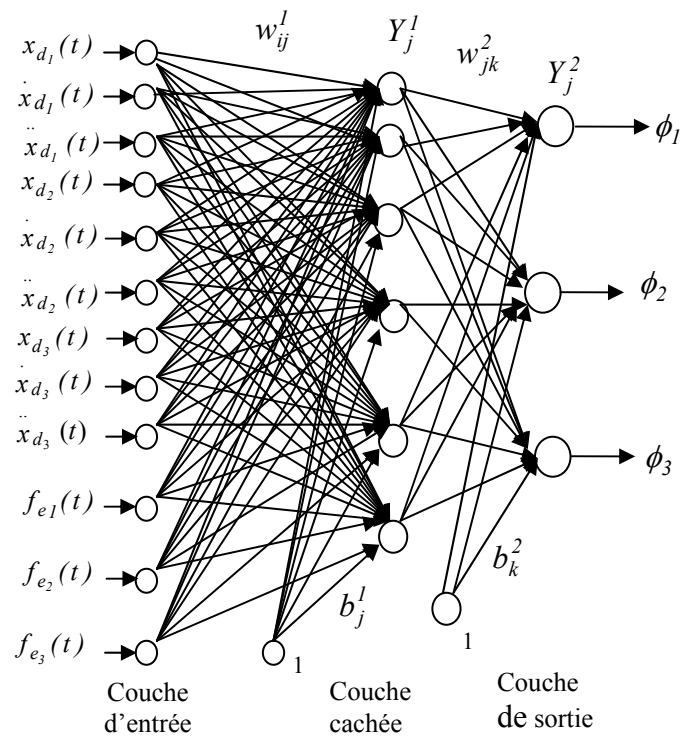


Figure 3.3 Structure du réseau de neurones

La dérivation de l'équation (3.39) par rapport au poids w donne

$$\frac{\delta J}{\delta w} = \frac{\delta v^T}{\delta w} v = -\frac{\delta \phi^T}{\delta w} v \quad (3.40)$$

L'adaptation des poids est obtenue en utilisant l'algorithme de rétro-propagation du gradient donnée par l'équation suivante:

$$\Delta w(t) = \eta \frac{\delta \phi^T}{\delta w} v + \mu \Delta w(t-1) \quad (3.41)$$

Où η est le taux d'adaptation et μ est le facteur d'oubli ou momentum [HAY 99].

Les étapes de calcul de l'algorithme de rétropropagation du gradient dans la commande sont données comme suit:

Initialiser les vecteurs des poids ($w=0, \Delta w=0$)

Pour le temps de simulation :

1. Calculer le vecteur d'entrée du réseau de neurones (équation (3.37))
2. Calculer la sortie du réseau de neurones par propagation du vecteur d'entrée:

$$Y_j^l = \frac{1 - \exp\left(-\left(\sum_{i=1}^{n_l} x_{d_i} w_{ij}^l + b_j^l\right)\right)}{1 + \exp\left(-\left(\sum_{i=1}^{n_l} x_{d_i} w_{ij}^l + b_j^l\right)\right)}$$

$$\phi_k = Y_k^2 = \sum_{j=1}^{n_H} Y_j^l w_{jk}^2 + b_k^2$$

3. Calculer le vecteur du signal v d'entraînement du réseau (équation (3.34))

4. Calculer Δw :

- $\Delta w_{ij}^l(t) = \eta \cdot 0.5 \cdot (1 - Y_j^l)^2 \cdot x_{d_i} \left[\sum_{k=1}^n v_k w_{jk}^2 \right] + \mu \Delta w_{ij}^l(t-1)$
- $\Delta w_{jk}^2(t) = \eta v_k Y_j^l + \mu \Delta w_{jk}^2(t-1)$
- $\Delta b_k^2(t) = \eta v_k + \mu \Delta b_k^2(t-1)$
- $\Delta b_j^l(t) = \eta \cdot 0.5 \cdot (1 - Y_j^l)^2 \left[\sum_{k=1}^n v_k w_{jk}^2 \right] + \mu \Delta b_j^l(t-1)$

5. Adapter les poids du réseau et aller vers l'étape 1

3.5. Implémentation

Afin de vérifier l'efficacité de la structure neuronale de la commande parallèle force-position que nous avons proposée dans cette partie du travail, plusieurs simulations ont été effectuées grâce au modèle du robot PUMA 560 pour ses trois premiers degrés de liberté [ARM 86] en supposant que ses trois dernières articulations sont bloquées (la structure de ce bras ainsi que ses modèles sont donnés en annexe A).

Les simulations effectuées visent à évaluer les performances de la commande proposée pour des tâches en milieu contraint. Trois types de simulations sont proposés dans ce cas là, la première consiste à tester la compliance du robot vis-à-vis de la méconnaissance de la position de son environnement, pour cela nous avons introduit une perturbation de position δX_e de type échelon. Dans ce cas l'environnement est supposé rectiligne et parallèle au repère de référence. La deuxième simulation proposée consiste à faire suivre à l'effecteur un contour circulaire sur un plan incliné rectiligne tout en exerçant un effort désiré selon la normale au plan alors que la troisième consiste à faire suivre à l'effecteur un contour curviligne.

Le choix adéquat des paramètres de réglage de la structure de commande donnée par la figure 3.1 peut être obtenu par essais successifs, opération délicate à réaliser en particulier pour les directions commandées en effort où les paramètres du réglage sont nombreux. Nous

proposons dans ce qui suit une méthode simple qui permet de choisir convenablement ces paramètres.

Considérons l'équation du troisième ordre du système d'équations (3.23). Cette équation peut s'écrire dans le domaine de Laplace comme suit:

$$(m_d p^3 + k_v p^2 + (k_p + k_f k_e) p + k_i k_e) x_n(p) \quad (3.42)$$

Le choix adéquat des paramètres de réglage de cette équation (3.42) peut être obtenu en lui imposant de se comporter comme un système du troisième ordre de pulsation w_0 et de facteur d'amortissement ξ convenablement choisi :

$$G(p) = (p + w_0)(p^2 + 2\xi w_0 p + w_0^2) \quad (3.43)$$

Pour assurer la stabilité discrète du système en boucle fermée et limiter les bruits générés par les capteurs, les valeurs des gains de réglage ne doivent pas être choisies trop élevées [DUM 87]. Pour cela, nous avons posé $w_0=20$ rd/s et $\xi=1$. Si on considère que la raideur de l'environnement k_e est égale à 10^4 N/m, dans ce cas la comparaison des équations (3.42) et (3.43) nous permet de déduire les réglages suivants:

$$k_p=1200\text{N/m}, k_v=60\text{NS/m}, m_d=1\text{kg}, k_i=0.8\text{s}^{-1} \text{ et } k_f = 0.$$

Les paramètres de réglage des deux équations de second ordre du système (3.23) sont choisis les mêmes vu que la condition de stabilité de l'ensemble reste vérifiée ($m_d, k_p, k_v > 0$). Donc le réglage global des paramètres du système pour l'ensemble des simulations est obtenu comme suit:

$$M_d = I, K_p = 1200 \times I, K_v = 60 \times I, K_i = 0.8 \times I \text{ et } K_f = 0$$

La période d'échantillonnage est fixée à $T_e=5$ ms, Les poids initiaux du réseau sont fixés à zéro ($w=0$).

Simulation 1:

Une surface plane est choisie dans ce cas caractérisée par: $n=(1,0,0)^T$ et la position de l'environnement est à $X_e=(0.45,0.04,0.45)^T$ m (voir figure 3.4).

Une perturbation de type échelon $\delta X_e=(0.005,0,0)m$, due à une méconnaissance de la position de l'environnement, est introduite ici.

Les résultats de simulation sont représentés par la figure 3.5. On remarque que dans le cas adapté le système est compliant avec un net rétablissement de l'impact en un temps de réponse inférieur à 0.5s et une adaptation de la position de l'effecteur par rapport à une mauvaise connaissance de la localisation de l'environnement.

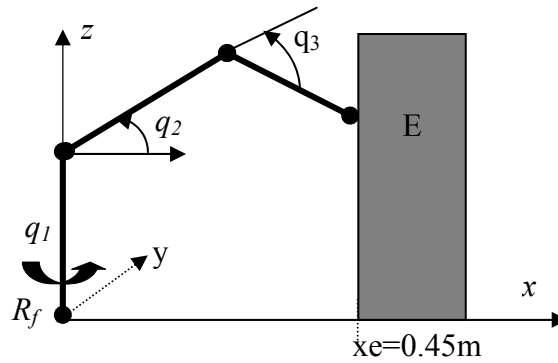


Figure 3.4. Robot à trois axes avec son environnement

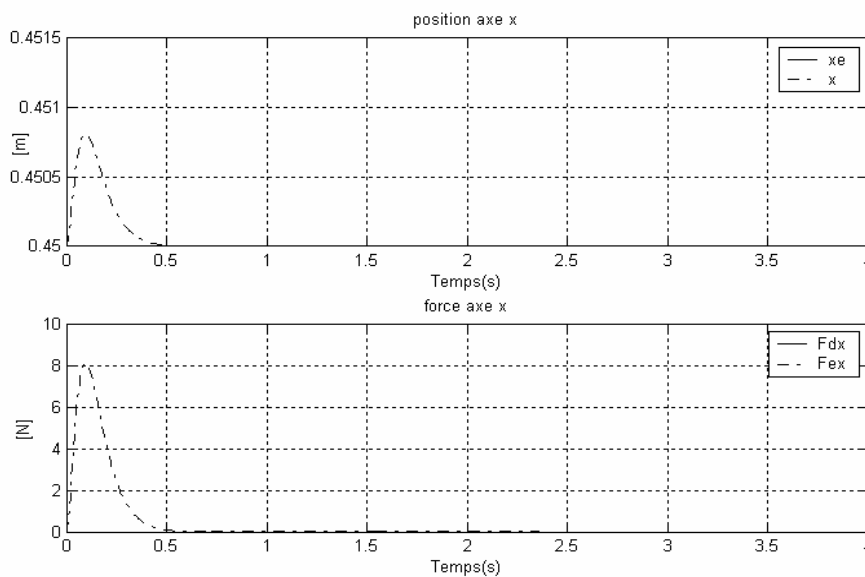


Figure 3.5. Evolution de la position et l'effort (cas adapté)

Pour évaluer la robustesse de notre approche, nous avons introduit les incertitudes suivantes sur les modèles du robot et de son environnement :

- Charge portée par le robot $m=10\text{kg}$
- Raideur de l'environnement plus grande $k_e=5 \times 10^4 \text{ N/m}$.

Nous avons gardé les mêmes valeurs des paramètres du réglage précédent auxquels nous avons ajouté le réglage des paramètres du réseau de neurones (figure 3.3), choisi expérimentalement comme suit: $\eta=0.0008$; $\mu=0.9$

Les résultats de simulation obtenus avec la structure de la figure 3.1 sont donnés par les figures 3.6 et 3.8 et ceux obtenus avec la structure de la figure 3.2 correspondent aux figures 3.7 et 3.9.

On remarque une nette dégradation des performances avec la structure classique (réponses oscillatoires amplifiées) alors que les réponses sont nettement meilleures avec le compensateur

neuronal (réponses oscillatoires amorties avec temps de réponse inférieur à 0.5s). Les figures 3.8 et 3.9 représentant les signaux de perturbations v (équation (3.34)) montrent comment ces derniers sont réduits par notre approche neuronale.

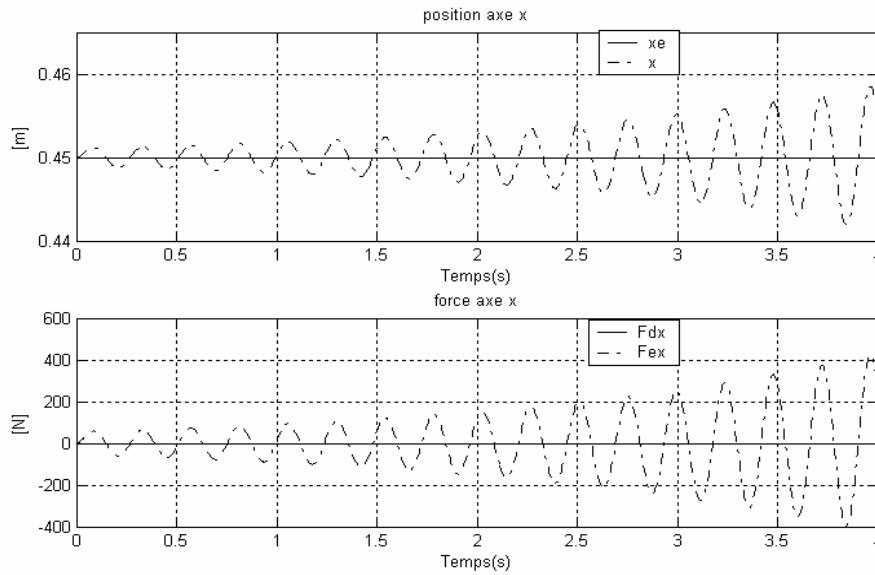


Figure 3.6. Evolution de la position et l'effort (Sans compensateur neuronal)

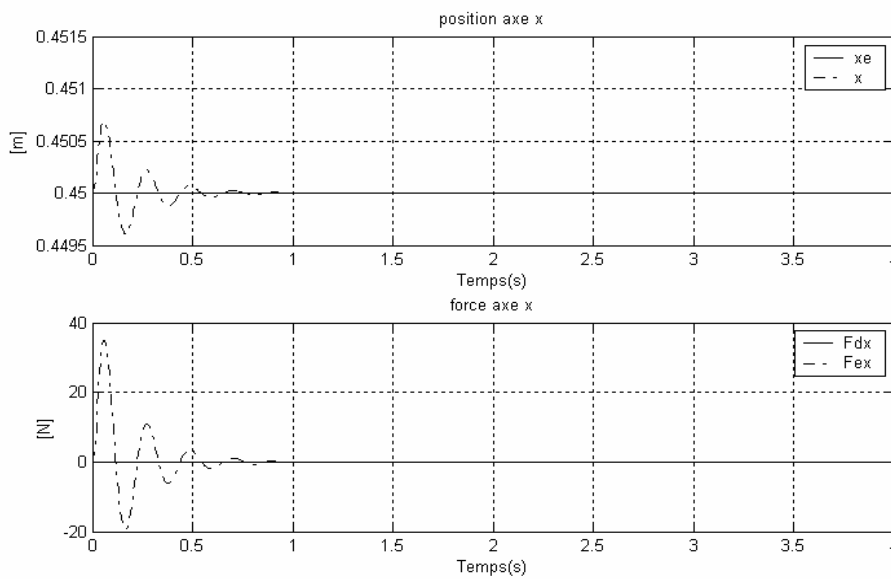


Figure 3.7. Evolution de la position et l'effort (Avec compensateur neuronal)

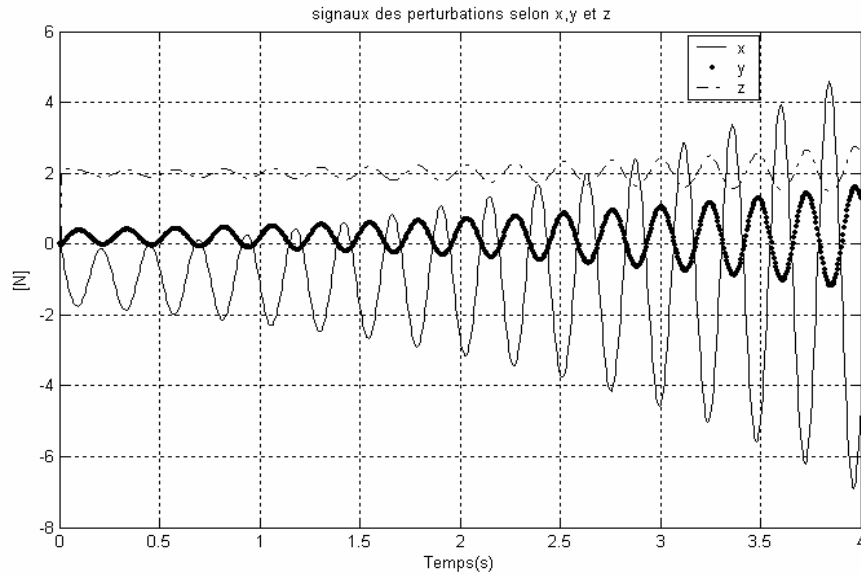


Figure 3.8. Evolution des signaux des perturbations (sans compensateur neuronal)

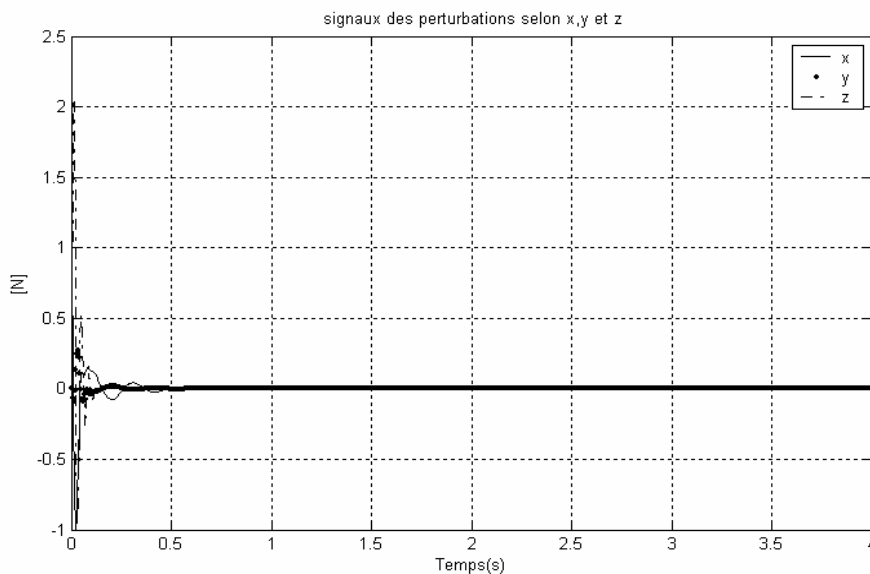


Figure 3.9. Evolution des signaux des perturbations (avec compensateur neuronal)

Simulation 2 :

Cette deuxième simulation vise à évaluer les performances de la commande que nous avons proposée dans le cas du suivi d'un cercle sur un plan incliné (figure 3.10). Le robot étant sur son environnement à la position $X = (0.53, 0.3, 0.1)$ m, il exerce un effort désiré de 5 N selon la normale $n = (\cos(\pi/4), 0, -\sin(\pi/4))^T$ et le maintient constant pendant que l'effecteur dresse un cercle sur cette surface.

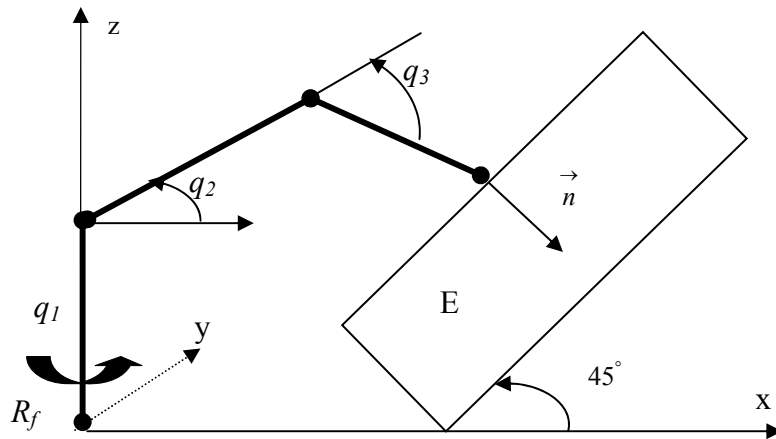


Figure 3.10. Robot à trois axes sur son environnement incliné de 45°

Les équations définissant la trajectoire désirée dans le repère de référence R_0 sont:

$$\begin{aligned}
 x_d(t) &= 0.106 \cos\left(\frac{2\pi}{T}t\right) + 0.424 \\
 y_d(t) &= 0.15 \sin\left(\frac{2\pi}{T}t\right) + 0.3 \\
 z_d(t) &= 0.1 \cos\left(\frac{2\pi}{T}t\right)
 \end{aligned} \tag{3.44}$$

Avec, $T=8s$ (la période).

Pour évaluer la robustesse de notre approche, nous avons introduit les incertitudes suivantes sur les modèles du robot ainsi que sur son environnement :

- Charge portée par le robot $m=10kg$
- Frottements au niveau des articulations, modélisés par:

$$\tau_f = 0.8 \dot{q} + 0.5 \text{sign}(\dot{q})$$

- Raideur de l'environnement variable et discontinue donnée par :

$$k_e = \begin{cases} 10000 \text{ N/m} & 1 < t \leq 4 \\ 40000 \text{ N/m} & 4 < t \leq 8 \\ 20000 \text{ N/m} & 8 < t \leq 12 \\ 30000 \text{ N/m} & 12 < t \leq 16 \end{cases}$$

Nous avons maintenu les mêmes valeurs des paramètres de réglage que dans la simulation précédente, les résultats sont représentés par les figures 3.11-3.17. On remarque une nette amélioration du suivi de la trajectoire selon la direction y (mouvement libre) (figure 3.11 et 3.12) en introduisant le compensateur neuronal (les erreurs selon la direction x et z représentent la pénétration de l'effecteur dans la surface).

Dans le cas du suivi de l'effort appliqué (figure 3.13 et 3.14), on remarque que l'introduction des variations brusques de la raideur de l'environnement induit des pics suivis d'oscillations d'amplitude nettement amortie et de durée très faible (inférieur à 0.5s) en introduisant le compensateur neuronal. On note ici que les petites fluctuations apparentes aux instants $t=1, 5, 9$ et 13 s (figure 3.14) sont dues aux effets des frottements secs et visqueux)

La comparaison des figures 3.15 et 3.16 montre de façon claire la compensation des perturbations par notre approche neuronale. La figure 3.17 représente l'évolution de la trajectoire dans l'espace avec le réseau de neurones.

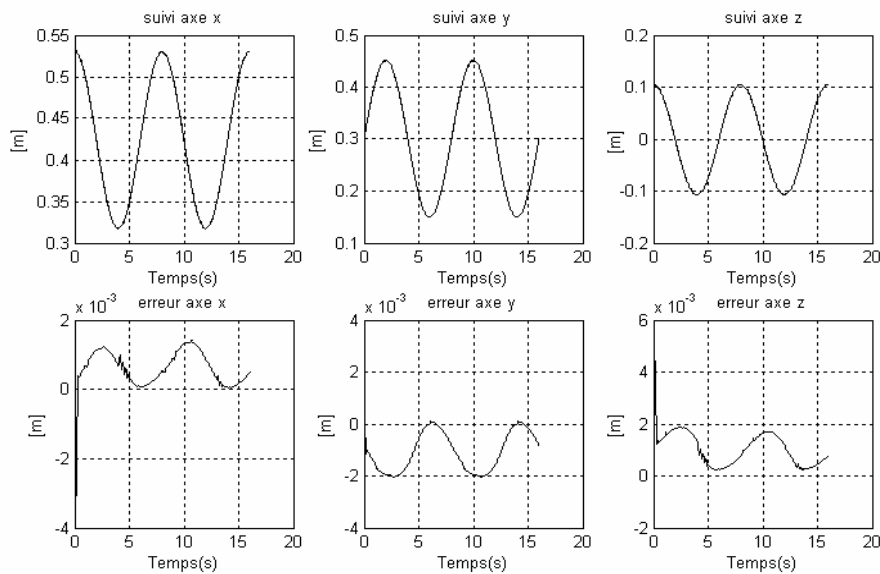


Figure 3.11. Evolution de la position de l'effecteur (sans compensateur neuronal)

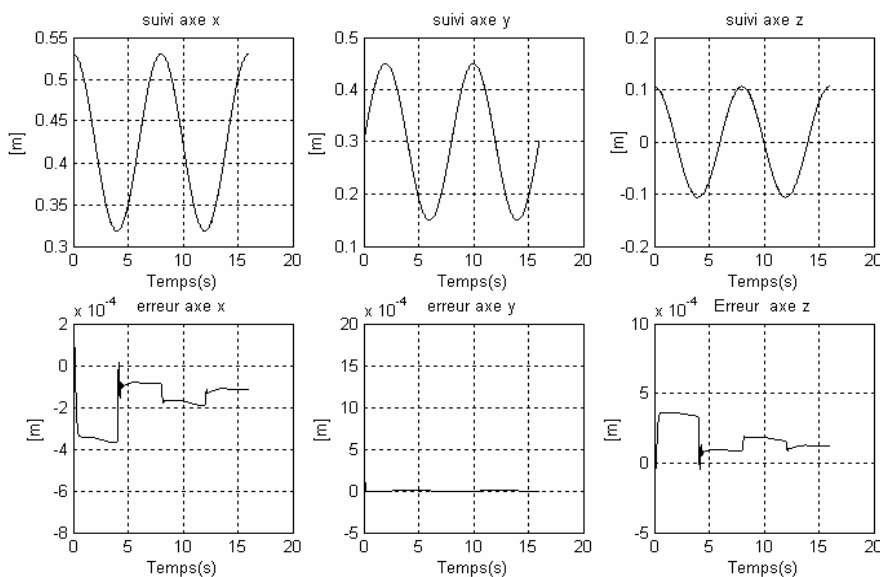


Figure 3.12. Evolution de la position de l'effecteur (avec compensateur neuronal)

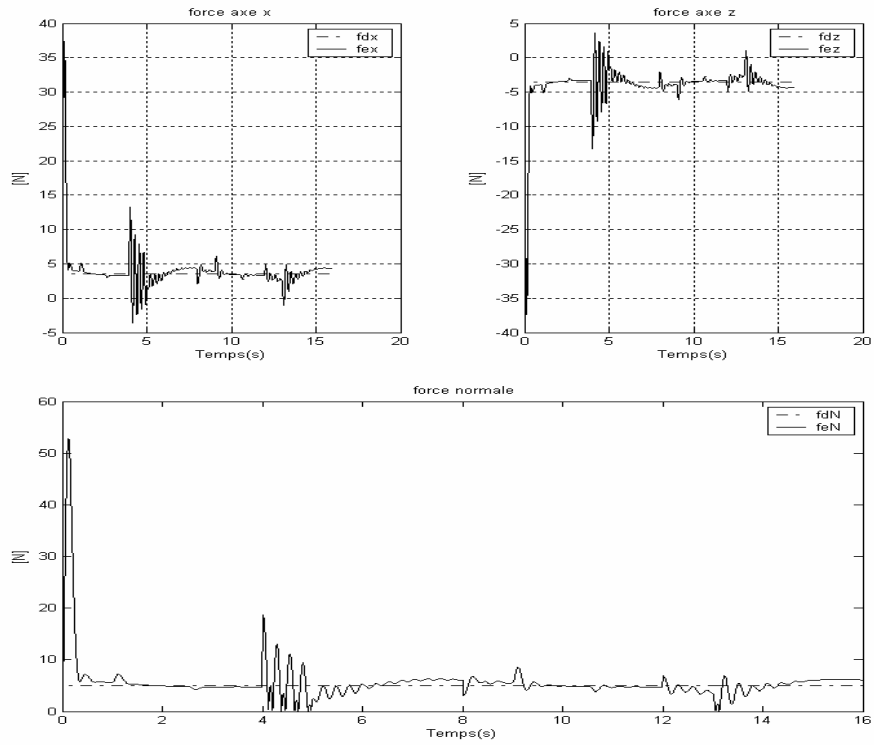


Figure 3.13. Evolution des efforts appliqués sur la surface (sans compensateur neuronal)

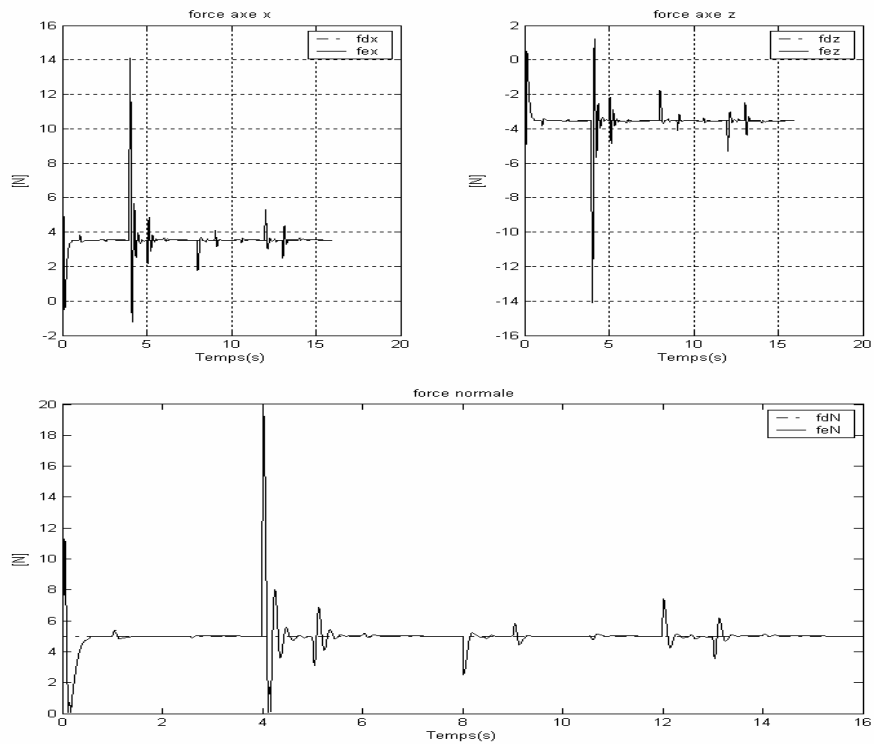


Figure 3.14. Evolution des efforts appliqués sur la surface (avec compensateur neuronal)

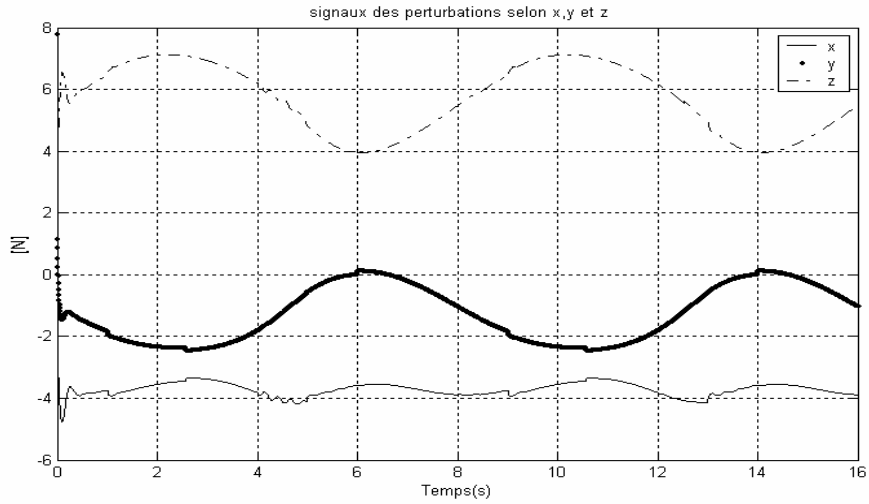


Figure 3.15. Evolution des signaux des perturbations (sans compensateur neuronal)

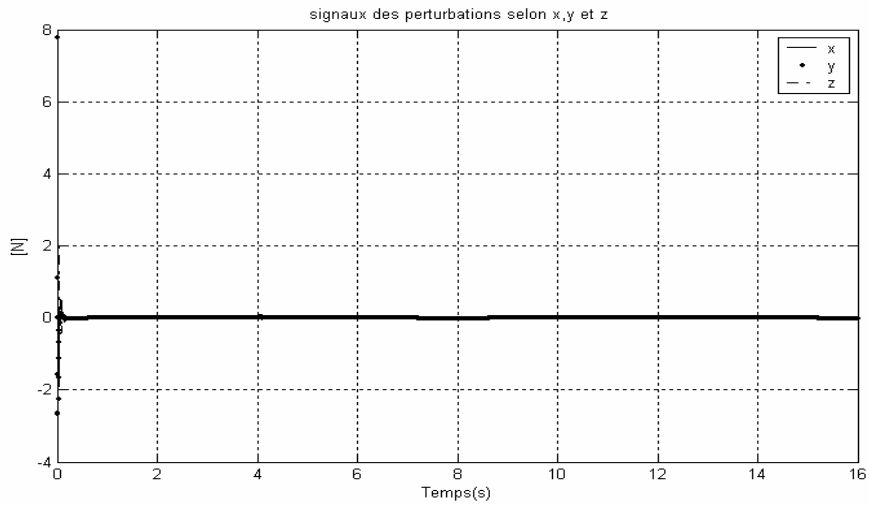


Figure 3.16. Evolution des signaux des perturbations (avec compensateur neuronal)

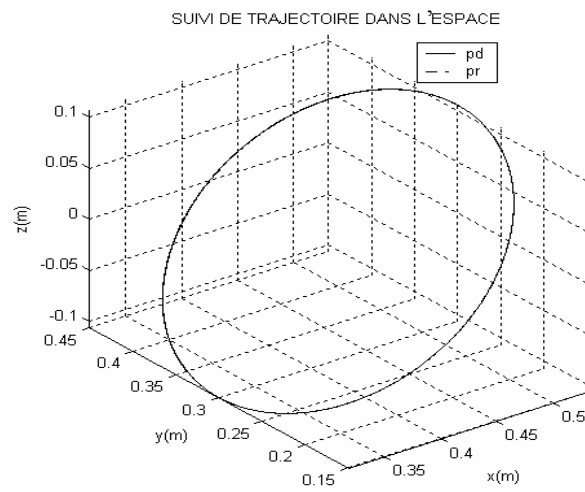


Figure 3.17. Evolution de la trajectoire dans l'espace (avec compensateur neuronal)

Simulation 3:

La troisième simulation proposée dans cette partie du travail consiste à faire suivre par l'extrémité du robot une trajectoire sur une surface curviligne (figure 3.18).

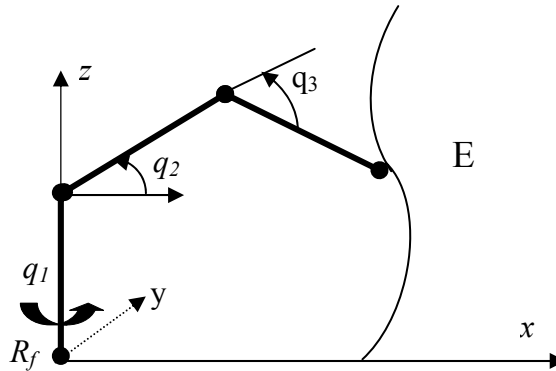


Figure 3.18. Robot à trois axes en contact avec une surface curviligne

Rappelons qu'une tâche de commande force/position consiste à suivre une trajectoire sur la surface considérée en imposant un effort F_d normal. Nous supposons connue à tout instant, la trajectoire $X_d(t)$ à suivre par l'extrémité du robot. Lorsque cette trajectoire est sur l'environnement il faut de plus assurer un effort normal dont le module $|F_d|$ est imposé. Les composantes du vecteur de consigne \vec{F}_d peuvent alors être déduites dans l'analyse de la tâche par le calcul de la normale \vec{n} en chaque point de la trajectoire.

Soit une surface de type sinusoïdal donnée par :

$$x_e = x_{em} + \Delta z \sin\left(\frac{2\pi}{l}(z_e - z_{em})\right) \quad (3.45)$$

Où,

x_{em} est l'amplitude moyenne de la surface et Δz , l et z_{em} étant respectivement l'amplitude, la période et le déphasage de l'onde sinusoïdale selon la direction z_e .

Le vecteur normal à cette surface est donné par [CHI 93] :

$$n(X_e) = \frac{1}{\sqrt{1 + \left(\Delta z \frac{2\pi}{l} \cos\left(\frac{2\pi}{l}(z_e - z_{em})\right)\right)^2}} \begin{pmatrix} 1 \\ 0 \\ -\Delta z \frac{2\pi}{l} \cos\left(\frac{2\pi}{l}(z_e - z_{em})\right) \end{pmatrix} \quad (3.46)$$

Pour valider notre approche, nous avons considéré l'équation (3.45) caractérisée par les valeurs des paramètres suivants: $z_{em} = 0 \text{ m}$, $\Delta z = 0.03 \text{ m}$, $l = 0.3 \text{ m}$ et $x_{em} = 0.2 \text{ m}$. La vitesse de déplacement sur la surface est choisie égale à 0.015 m/s et la force normale appliquée est de

module $|F_d| = 5N$. La tâche désirée consiste à déplacer l'effecteur de la position $A = (0.175, 0.3, 0.2)$ m vers la position $B = (0.175, 0.3, 0.5)$ m en suivant la surface définie par (3.45).

Nous avons considéré les mêmes perturbations que précédemment sauf que la variation de la raideur est supposée dans ce cas comme suit :

$$k_e = \begin{cases} 10000 \text{ N/m} & 1 < t \leq 8 \\ 40000 \text{ N/m} & 8 < t \leq 12 \end{cases}$$

Le réglage des paramètres des contrôleurs est maintenu et les résultats de simulation obtenus sont représentés par les figures 3.19-3.25.

On remarque qu'une nette amélioration de la précision en position selon la direction fixe y (figures 3.19 et 3.20) en introduisant le compensateur neuronal (les erreurs selon la direction x et z représentent la pénétration de l'effecteur dans la surface).

Dans le cas du suivi de l'effort appliqué (figures 3.21 et 3.22), on remarque que les incertitudes sur les modèles du robot et son environnement ont induit des réponses en effort oscillatoires mais nettement amortie (durée inférieure à 0.5s) en utilisant le compensateur neuronal.

La comparaison des figures 3.23 et 3.24 met en évidence la compensation des perturbations par notre approche neuronale. La figure 3.25 représente l'évolution de la trajectoire dans le plan (x,y) .

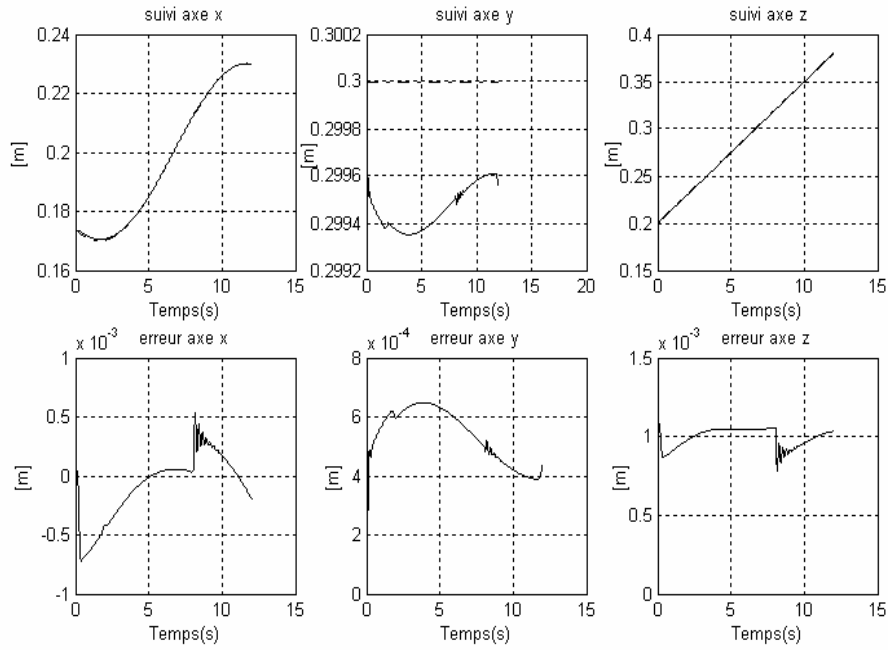


Figure 3.19. Evolution de la position de l'effecteur (sans compensateur neuronal)

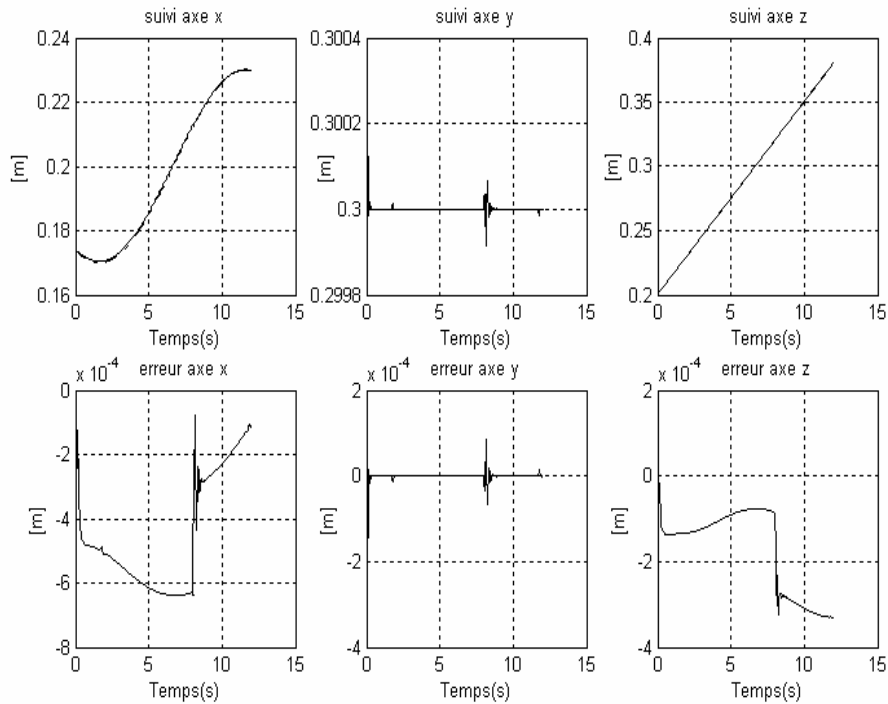


Figure 3.20. Evolution de la position de l'effecteur (avec compensateur neuronal)

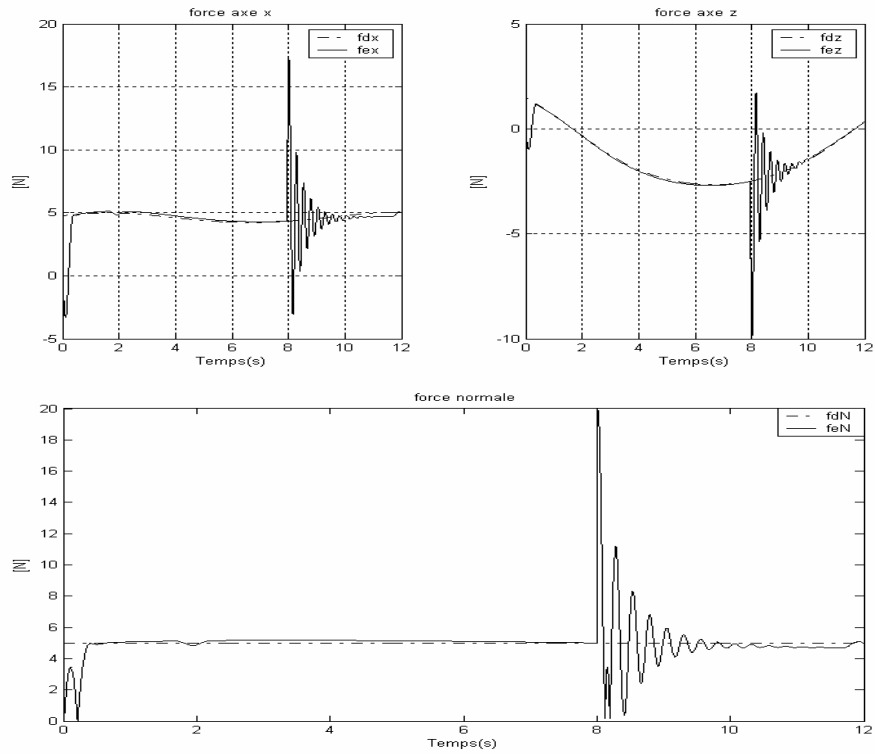


Figure 3.21. Evolution des efforts appliqués sur la surface (sans compensateur neuronal)

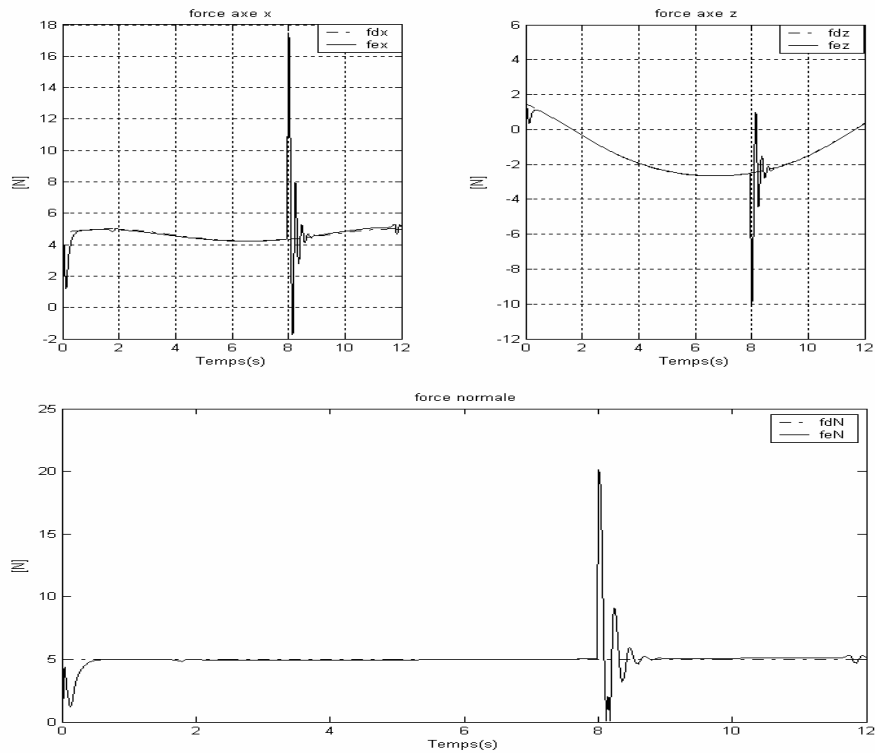


Figure 3.22. Evolution des efforts appliqués sur la surface (avec compensateur neuronal)

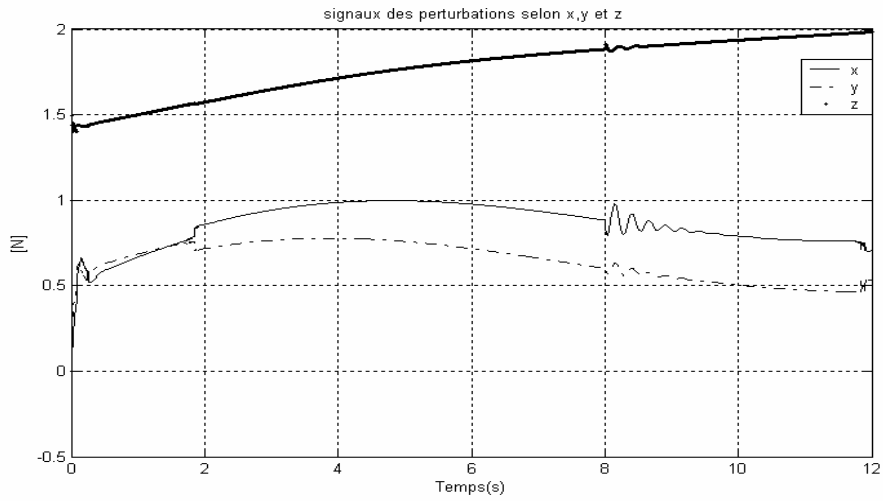


Figure 3.23. Evolution des signaux des perturbations (sans compensateur neuronal)

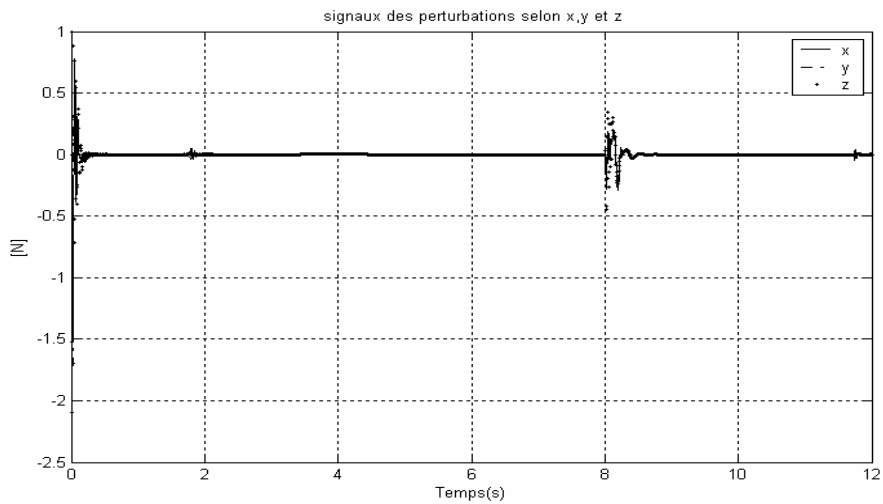


Figure 3.24. Evolution des signaux des perturbations (avec compensateur neuronal)

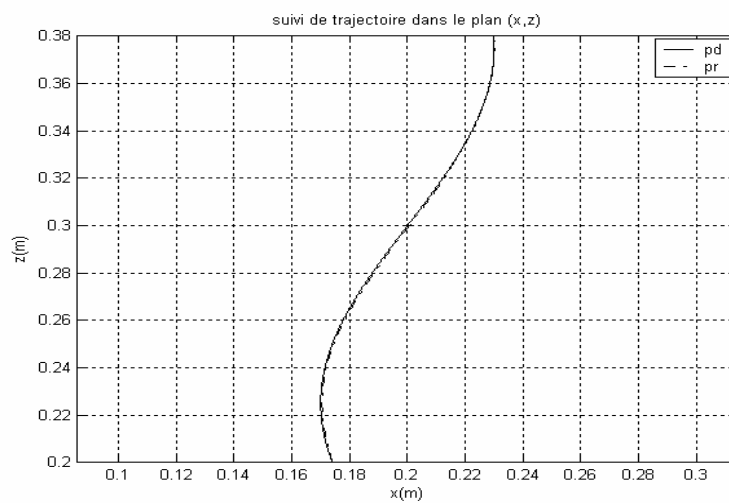


Figure 3.25. Evolution de la trajectoire dans l'espace (avec compensateur neuronal)

3.6 Structure neuronale parallèle avec un contrôleur flou de sa boucle de commande en effort

Les oscillations résiduelles observées pour les réponses en effort des simulations précédentes sont dues au fait que notre système décrit par l'équation (3.42) n'obéit plus à l'équation (3.43) dont les paramètres ont été imposés (fixés) alors que le système dépend en réalité de la raideur k_e de l'environnement qui varie dans le temps. En effet cette variation de la raideur de l'environnement change le placement des pôles choisis de l'équation (3.42).

Dans le chapitre 2 nous avons montré que l'on peut concevoir des contrôleurs flous qui jouent le même rôle qu'un contrôleur PI classique mais avec des gains variables qui peuvent s'adapter en ligne pour faire face dans notre cas aux effets de la variation de la raideur de l'environnement.

Nous proposons dans ce qui suit une nouvelle structure de commande parallèle utilisant un contrôleur flou dans la boucle de commande en effort, permettant :

- d'améliorer les performances du régime transitoire, c'est-à-dire réduire les dépassements et les oscillations.
- d'atteindre la force désirée indépendamment de la connaissance préalable de la raideur de l'environnement (raideur incertaine).

Dans la nouvelle structure de commande neuronale parallèle force/position que nous avons proposée, le contrôleur conventionnel de la boucle de commande en effort est remplacé par un contrôleur flou. Ce qui permet de générer à sa sortie une commande en fonction de l'écart entre la force actuelle (mesurée) et la force désirée ainsi que la variation de cet écart. Cette nouvelle structure est représentée par la figure 3.27.

3.6.1 Description du contrôleur flou

Le contrôleur flou utilisé dans la structure de commande donné à la figure 3.27 est un contrôleur de type PI auto ajustable. Son schéma bloc détaillé est donné par la figure suivante:

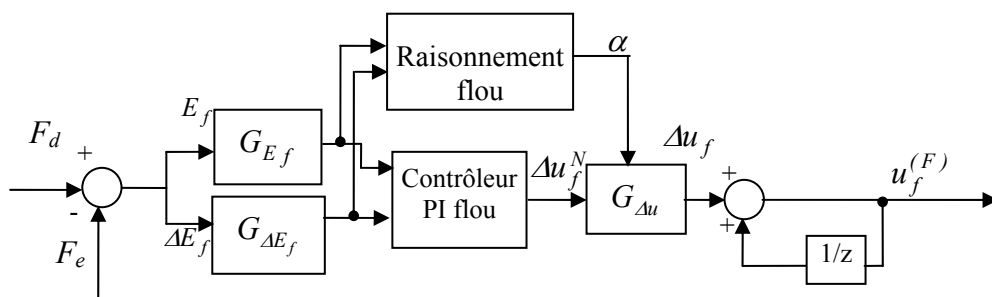


Figure 3.26 : Structure du contrôleur PI flou auto ajustable

Ce contrôleur flou a donc deux entrées l'erreur $E_f(n)$ et la variation de l'erreur $\Delta E_f(n)$ qui sont définies par $E_f(n) = F_d(n) - F_e(n)$, $\Delta E_f(n) = E_f(n) - E_f(n-1)$. Les indices n et $n-1$ indiquent respectivement l'état présent et l'état précédent du système.

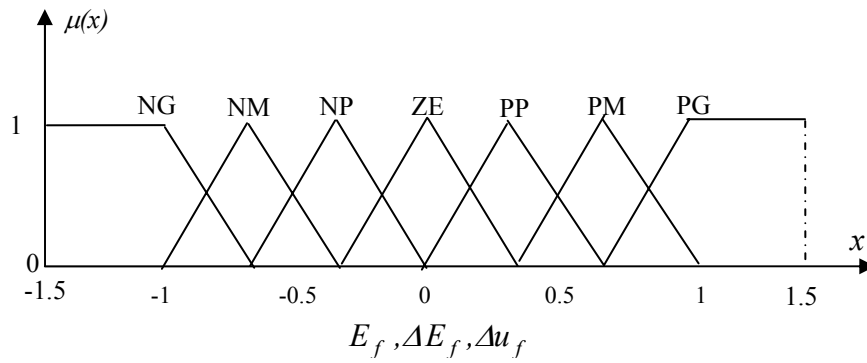


Figure 3.28: Fonctions d'appartenance de E_f , ΔE_f and Δu_f

$E_f(n)/\Delta E_f(n)$	NG	NM	NP	ZE	PP	PM	PG
NG	NG	NG	NG	NM	NP	NP	ZE
NM	NG	NM	NM	NM	NP	ZE	PP
NP	NG	NM	NP	NP	ZE	PP	PM
ZE	NG	NM	NP	ZE	PP	PM	PG
PP	NM	NP	ZE	PP	PP	PM	PG
PM	NP	ZE	PP	PM	PM	PM	PG
PG	ZE	PP	PP	PM	PG	PG	PG

Tableau 1 : base des règles pour le calcul de Δu_f .

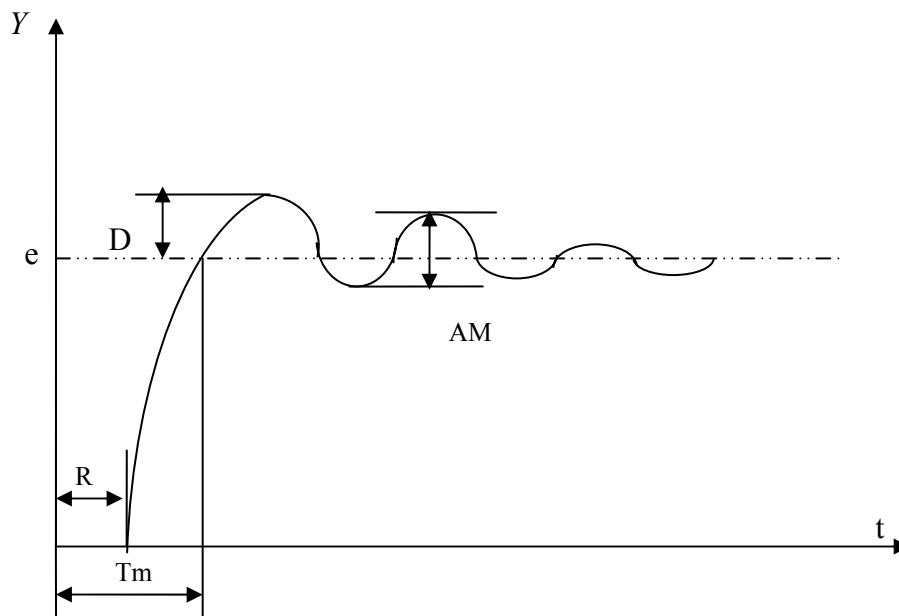


Figure 3.29 : Paramètres des performances d'une réponse indicielle

Les paramètres d'évaluations sont les suivants :

- D : dépassement
- Tm : temps de montée
- R : retard
- AM : amplitude
- e : régime permanent.

3.6.1.1 gains de normalisation

Les gains de normalisation des entrées permettent de transformer les valeurs physiques de l'état du système vers un domaine normalisé prédéfini. Par ailleurs les gains de dénormalisation à la sortie permettent de transformer les sorties normalisées dans le domaine physique. Ces gains de normalisation jouent un rôle similaire aux gains des contrôleurs conventionnels. D'où, leur réglage joue un rôle primordial dans la détermination de la stabilité du système en boucle fermée. La relation entre les gains de normalisation ($G_{E_f}, G_{\Delta E_f}, G_{\Delta u_f}$) et les variables d'entrées/sorties du contrôleur flou est donnée comme suit :

$$E_f^N = G_{E_f} \times E_f, \quad \Delta E_f^N = G_{\Delta E_f} \times \Delta E_f, \quad \Delta u_f = G_{\Delta u_f} \times \Delta u_f^N$$

3.6.1.2 Auto réglage des gains de dénormalisation

La méthode utilisant un réglage fixe des gains de normalisation et de dénormalisation peut devenir inadéquate dans certains cas où les écarts en effort deviennent importants (écarts sortants du domaine de normalisation) et qui sont dus à la variation de la raideur de l'environnement. Pour remédier à ce problème (uniquement pour une certaine plage de variation de la raideur de l'environnement) un régulateur des gains de dénormalisation des sorties du contrôleur basé sur un raisonnement flou est implémenté comme il est montré sur la figure 3.26. En effet cet auto réglage en ligne des gains de dénormalisation modifie les surfaces des sorties du contrôleur flou [ZHA 03].

Les incréments des sorties du contrôleur flou Δu_f sont obtenus par l'équation suivante :

$$\Delta u_f = (\alpha \cdot G_{\Delta u_f}) \times \Delta u_f^N$$

Ainsi les gains de réglage des sorties ne restent pas fixes lorsque le contrôleur est opérationnel. Ils sont variables à chaque période d'échantillonnage grâce à un gain adaptatif α obtenu en ligne en se basant sur un raisonnement flou, utilisant l'erreur et la variation de l'erreur de l'effort ressenti à chaque période d'échantillonnage.

Les fonctions d'appartenance des entrées E_f et ΔE_f sont définies dans un même domaine de normalisation $[-1, 1]$, où chacune des entrées a trois ensembles flous, comme le montre la figure 3.30.

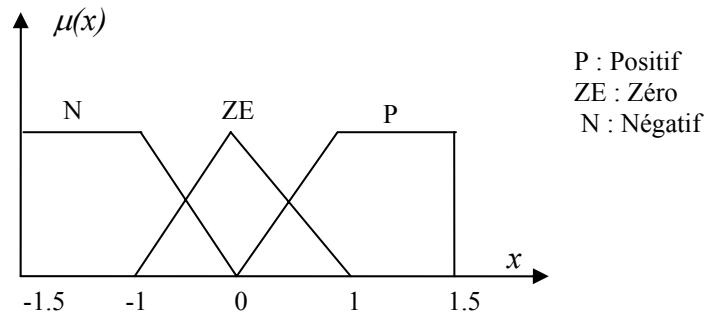


Figure 3.30 : Fonction d'appartenance de E_f et ΔE_f

Les fonctions d'appartenance de α sont définies par trois ensembles flous triangulaires. Le domaine de variation est défini selon le comportement de la réponse en effort observé vis-à-vis de la variation de la raideur de l'environnement. Pour notre cas nous avons choisi ce domaine de variation comme suit (figure 3.31) :

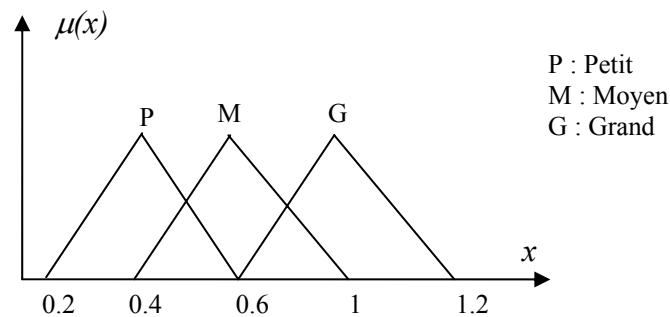


Figure 3.31 : Fonction d'appartenance de α

3.6.1.3. Base des règles de α

La base des règles de calcul de α est donnée par le tableau 2 ci-dessous. L'établissement de ces règles dépend directement de la base des règles donnée dans le tableau 1.

Par exemple, lorsque l'effort ressenti est loin par rapport à l'effort désiré, le gain de réglage devrait être large ; cela peut être atteint comme suit :

Si l'erreur et la variation de l'erreur sont négatives (ou bien positives) alors α est grand.

Lorsque la réponse en effort tend vers le régime permanent, le gain devrait être moyen ; cela peut être atteint comme suit :

Si peu importe l'erreur ou bien la variation de l'erreur qui est à zéro alors le gain α est moyen.

A l'équilibre le gain devrait être très petit ; cela peut être achevé par la règle suivante :

Si l'erreur et la variation de l'erreur sont zéros alors le gain α est petit.

$E_f(n) \setminus \Delta E_f(n)$	N	ZE	P
N	G	M	P
ZE	M	P	M
P	P	M	G

Tableau 2 : Base des règles pour le calcul de α

3.6.2. Résultats de simulation

Nous avons repris les mêmes conditions de simulation, avec les mêmes perturbations sur les modèles que précédemment. Nous avons gardé le même réglage des paramètres des contrôleurs auxquels nous avons rajouté le réglage des paramètres du contrôleur flou comme suit:

$$G_{e_f} = 0.09, \quad G_{\Delta e_f} = 0.9 \quad \text{et} \quad G_{\Delta u_f} = 20$$

Les résultats de simulation obtenus pour les réponses en effort pour les simulations 2 et 3 sont représentés par les figures suivantes:

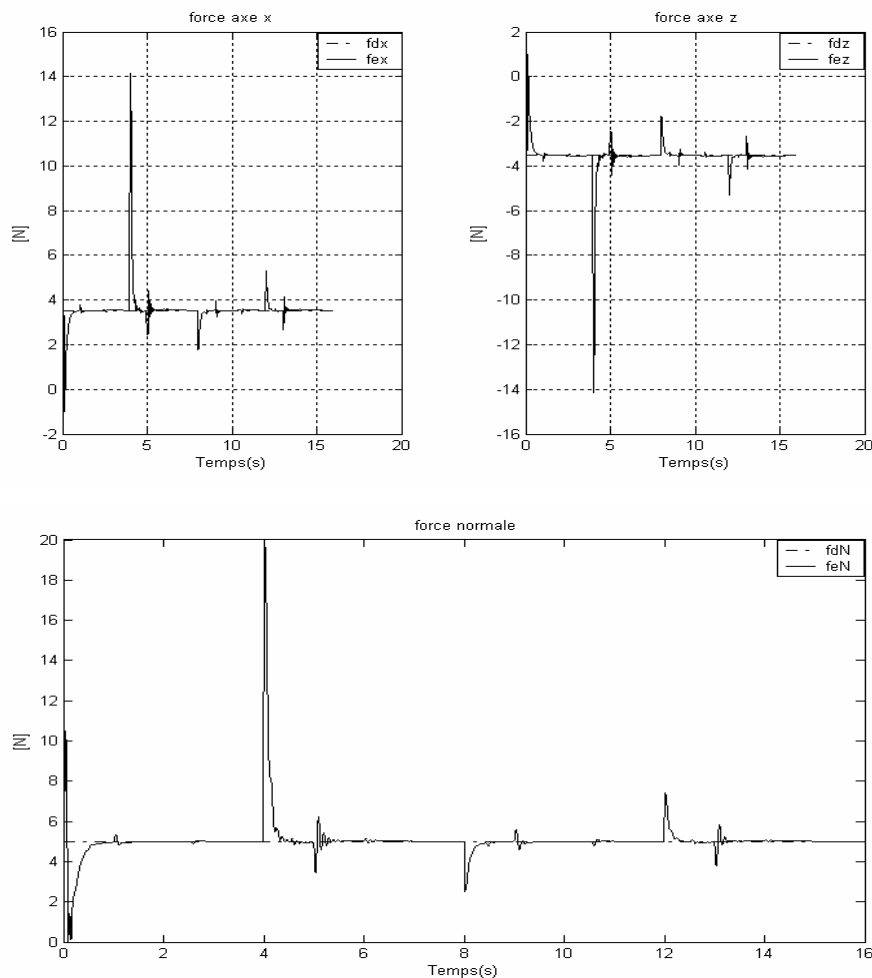


Figure 3.32. Evolution des efforts appliqués sur la surface en utilisant un contrôleur d'effort flou (simulation2)

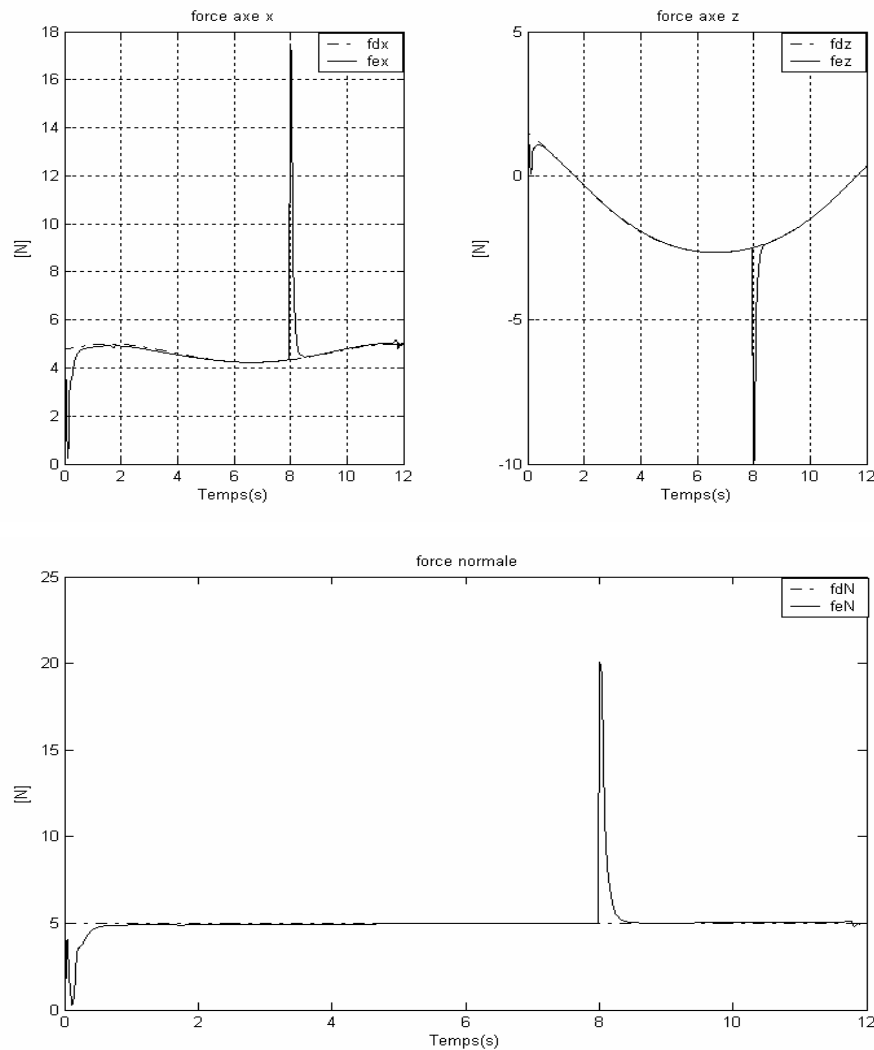


Figure 3.33. Evolution de l'effort appliqué sur la surface en utilisant un contrôleur d'effort flou (simulation 3)

Comparativement avec les réponses utilisant un contrôleur en effort conventionnel (3.14 et 3.22), on remarque que les performances des réponses en effort avec un contrôleur PI flou sont nettement meilleures (retour au régime permanent sans oscillations et avec des temps de réponse des régimes transitoires nettement diminué au moment de la variation de la raideur). On note ici que pour la simulation 2, les fluctuations résiduelles observées aux instants $t=1, 5, 9$ et 13 s sont dues aux frottements secs et visqueux.

Pour le test de la compliance vis-à-vis d'une méconnaissance de la position et la raideur de l'environnement nous avons constaté que la variation de la raideur de l'environnement induit de grands écarts pour l'effort appliqué à l'entrée du contrôleur situation conduisant le domaine de normalisation dont les gains sont fixes à être complètement débordé donnant ainsi des résultats erronés. Cependant pour une raideur supposée connue et avec le même réglage des gains donné précédemment, la compliance du système est facilement vérifiée, comme il est montré sur la figure 3.34 suivante :

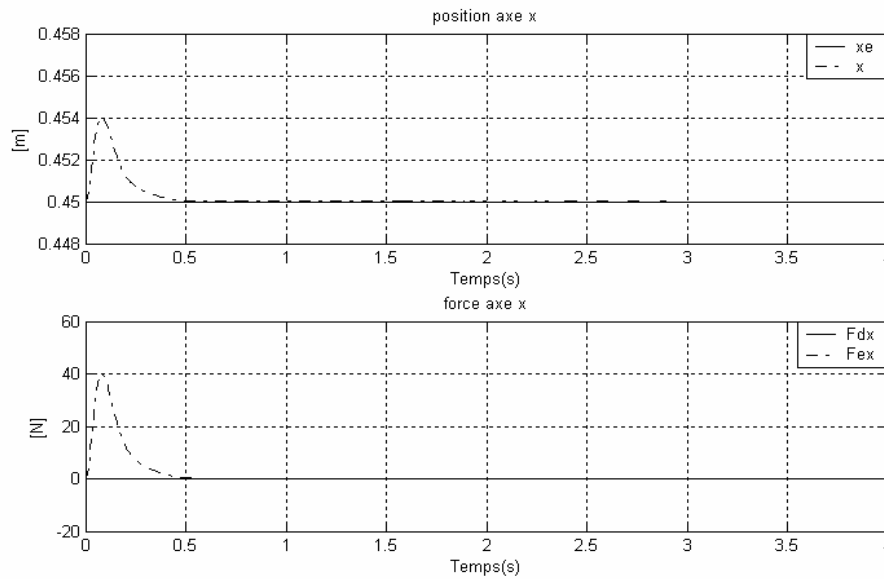


Figure 3.34. Evolution de la position et l'effort en utilisant un contrôleur d'effort PI flou (simulation 1)

3.7 Structures neuronales parallèles force/position non basées sur le modèle

Les structures de commande intelligentes parallèles force/position que nous avons proposées dans les paragraphes précédents sont basées sur l'exploitation du modèle dynamique nominale du manipulateur dans la commande. Cependant, dans la pratique, on peut être amenés à commander des robots dont les modèles et paramètres dynamiques sont inconnus.

Dans ce cas, il est intéressant de concevoir des lois de commande robuste qui ne font pas appel à ces modèles. La puissance des réseaux de neurones artificiels réside essentiellement dans leurs caractéristiques fondamentales d'approximer toute fonction non linéaire si leurs poids synaptiques sont bien adaptés. Grâce à cette caractéristique particulière des réseaux de neurones, ils peuvent être utilisés comme des identificateurs et en même temps des compensateurs de la dynamique non linéaire des bras manipulateurs.

Les approches de commandes neuronales proposées dans la littérature, et que nous avons décrit dans le chapitre 2 diffèrent par leur architecture, leur complexité de mise en oeuvre. Les structures de commande basées sur l'exploitation d'un contrôleur conventionnel pour minimiser la fonction coût (figure 2.11 et figure 2.12) nous paraissent très intéressantes et s'adaptent bien à nos applications en contrôle force/position.

Dans les parties suivantes nous proposons une nouvelle approche de commande neuronale parallèle force/position d'un bras manipulateur utilisant un simple contrôleur PID conventionnel. Dans un premier temps nous analyserons les performances d'une structure PID conventionnelle.

3.7.1 Structure de commande PID conventionnelle parallèle force/position

Les compensateurs PID ont été largement utilisés pour contrôler les bras manipulateur en position ou en effort à cause de leur simplicité [CHI 94][DEG 93][PER 91]. La structure parallèle classique force/position de type PID que nous proposons d'analyser ici est représentée par la figure 3.35 suivante :

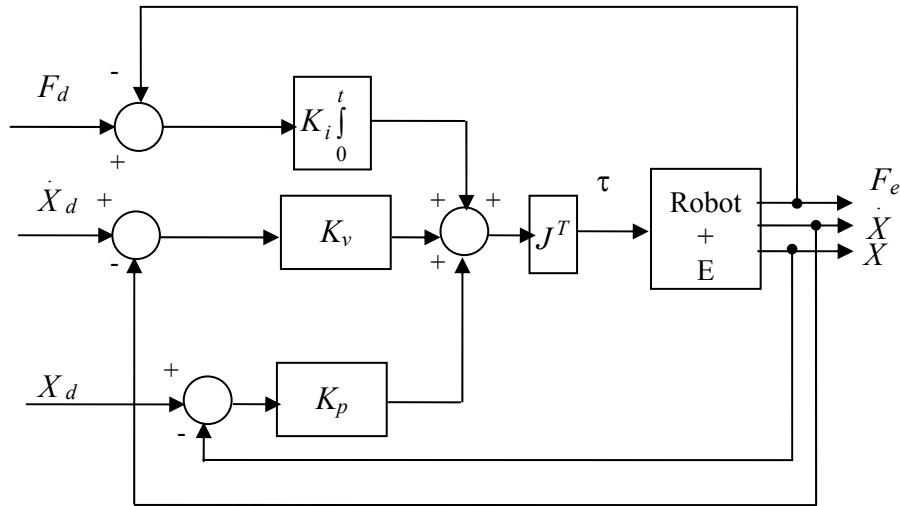


Figure 3.35 : Structure PID classique parallèle force/position

Cette structure de commande consiste en une parallélisation d'un contrôleur de position du type PD avec un contrôleur d'effort du type intégral. Elle est régie par l'équation suivante :

$$U = K_v \dot{E}_p + K_p E_p + K_i \int_0^t E_f d\tau \quad (3.47)$$

Avec,

$$E_p = X_d - X \text{ et } E_f = F_d - F_e,$$

K_v , K_p et K_i sont des matrices diagonales de réglage.

Si on combine les équations (3.8) et (3.47), on obtient pour le système en boucle fermée l'équation suivante :

$$D \ddot{E}_p + K_v \dot{E}_p + K_p E_p + K_i \int_0^t E_f d\tau = D \ddot{X}_d + H + F_f + F_e \quad (3.48)$$

Si on multiplie cette équation par D^{-1} on aura :

$$\ddot{E}_p + D^{-1} K_v \dot{E}_p + D^{-1} K_p E_p + D^{-1} K_i \int_0^t E_f d\tau = D^{-1} (D \ddot{X}_d + H + F_f + F_e) \quad (3.49)$$

Il est clair que les gains de cette équation varient avec la configuration du robot. De plus cette équation est fortement couplée par la matrice d'inertie D , ce qui rend la minimisation de

son effet sur les performances des réponses en position et en effort très difficiles. Une solution consiste à choisir des grands gains pour les matrices diagonales K_p et K_v . Dans ce cas on peut approximer l'équation (3.49) comme suit :

$$\dot{E}_p + K_v^{-1} K_p E_p + K_v^{-1} K_i \int_0^t E_f d\tau \cong -K_v^{-1} D \ddot{E}_p + K_v^{-1} (D \ddot{X}_d + H + F_f + F_e) \cong 0 \quad (3.50)$$

Dans la pratique, il y a toujours une limite pour la sélection des grands gains, comme cela a été démontré dans le chapitre 2, ce qui rend cette solution inappropriée.

Dans ce qui suit, nous proposons deux nouvelles structures neuronales parallèles force/position à base du contrôleur PID proposé ci-dessus (figure 3.35) qui permettent de découpler et de compenser la dynamique du manipulateur en contact avec son environnement.

3.7.2 Structure neuronale parallèle basée sur une technique de modification du couple de commande

La structure de commande neuronale parallèle que nous proposons ici est inspirée du schéma de commande donné à la figure 2.11 (voir chapitre 2). Dans cette structure de commande (figure 3.36), la sortie du réseau de neurones génère un signal équivalent au modèle dynamique inverse du robot qui sera donc identifié par le réseau de neurones proposé

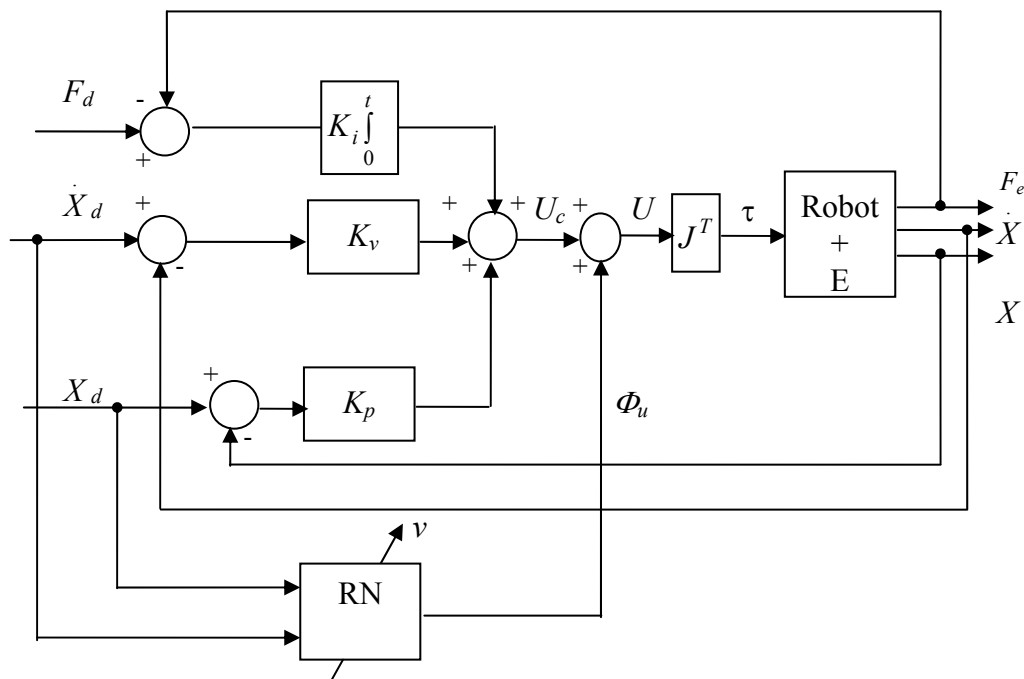


Figure 3.36 : Structure neuronale parallèle force/position basée sur une technique de modification du couple de commande (schéma 1)

D'après la figure ci-dessus, la loi de commande devient donc:

$$U = K_v \dot{E}_p + K_p E_p + K_i \int_0^t E_f d\tau + \Phi_u \quad (3.51)$$

La combinaison des équations (3.51) et (3.8) nous donne l'équation de la dynamique de l'erreur du système en boucle fermée suivante:

$$K_v \dot{E}_p + K_p E_p + K_i \int_0^t E_f d\tau + \Phi_u = D \ddot{X} + H + F_f + F_e \quad (3.52)$$

Si on définit le signal d'entraînement du réseau à minimiser tel que :

$$v = U_c = K_v \dot{E}_p + K_p E_p + K_i \int_0^t E_f d\tau \quad (3.53)$$

Dans ce cas, (3.52) peut s'écrire :

$$U_c = v = D \ddot{X} + H + F_f + F_e - \Phi_u \quad (3.54)$$

Dans le cas idéal lorsque $v=0$ la sortie du réseau de neurones est équivalente au modèle dynamique inverse du robot en contact avec son environnement tel que :

$$\Phi_u = D \ddot{X} + H + F_f + F_e \quad (3.55)$$

Ce réseau de neurones a pour tâche donc, d'assurer en même temps l'identification et la commande en temps réel du système qui sera régi à tout instant en boucle fermée par l'équation de la dynamique de l'erreur donnée par:

$$K_v \dot{E}_p + K_p E_p + K_i \int_0^t E_f d\tau = 0 \quad (3.56)$$

3.7.3 Structure neuronale parallèle basée sur la modification de la trajectoire de référence

Le deuxième schéma de commande neuronale parallèle force/position que nous proposons dans ce paragraphe est inspiré du schéma de commande donné à la figure 2.12 (voir chapitre 2). Le réseau de neurones est introduit au niveau de la trajectoire de référence, il génère à sa sortie un signal qui vient s'ajouter à la trajectoire désirée, qui se trouve ainsi modifiée, comme il est montré sur la figure 3.37 suivante :

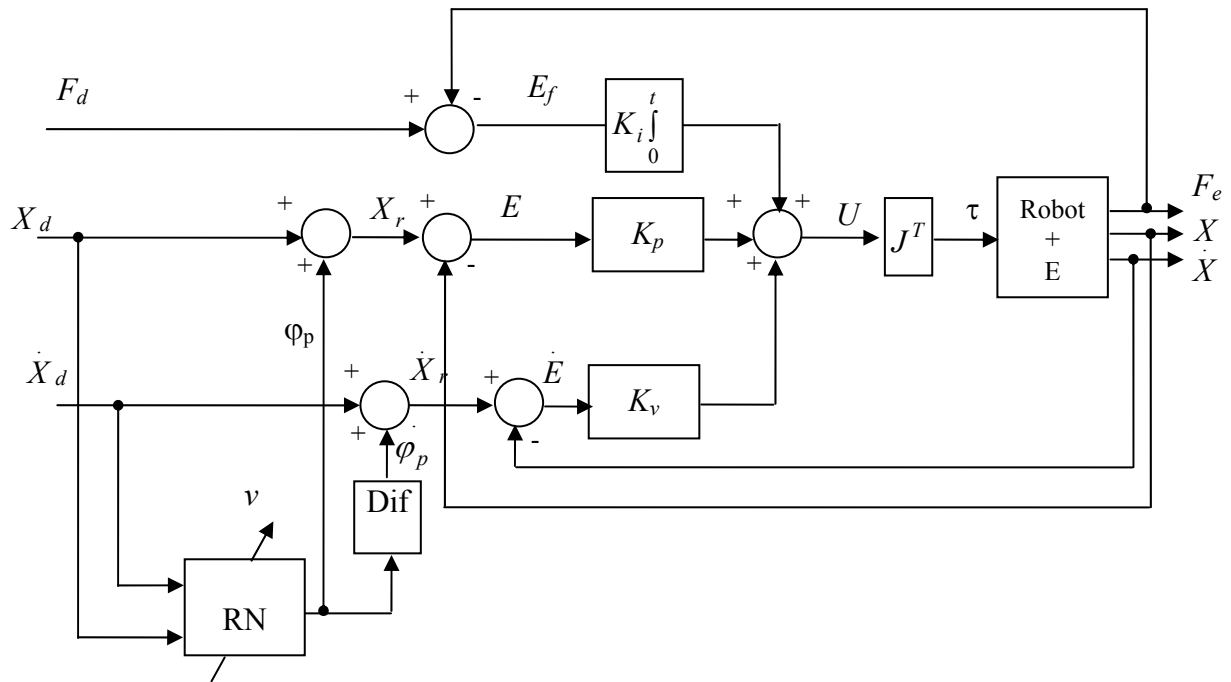


Figure 3.37 : Structure neuronale parallèle force/position basée sur une technique de modification de la trajectoire de référence (Schéma 2)

La loi de commande parallèle force/position appliquée ici (voir figure 3.37) est donnée par l'équation suivante:

$$U = K_v(\dot{X}_d + \dot{\varphi}_p - \dot{X}) + K_p(X_d + \varphi_p) + K_i \int_0^t E_f d\tau \quad (3.57)$$

Avec K_p, K_v, K_i sont des matrices diagonales de réglage.

En combinant les équations (3.57) et (3.8), l'équation en boucle fermée devient donc:

$$K_v \dot{E}_p + K_p E_p + K_i \int_0^t E_f d\tau = D \ddot{X} + H + F_f + F_e - (K_v \dot{\varphi}_p + K_p \varphi_p) \quad (3.58)$$

Avec, $E_p = X_d - X$ et $\dot{E}_p = \dot{X}_d - \dot{X}$

Où le terme $\dot{\varphi}_p$ est estimé par la méthode des différences finies comme suit :

$$\dot{\varphi}_p(t) \cong \frac{\varphi_p(t) - \varphi_p(t-1)}{T_e} \quad (3.59)$$

Avec T_e période d'échantillonnage.

On pose:

$$\Phi_p = K_p \varphi_p + K_v \dot{\varphi}_p \quad (3.60)$$

L'objectif recherché est de générer Φ_p pour faire tendre l'équation (3.58) vers zéro.

Si on pose

$$v = K_v \dot{E}_p + K_p E_p + K_i \int_0^t E_f d\tau \quad (3.61)$$

Alors, dans le cas idéal lorsque $v=0$ on aura :

$$\Phi_p = D \ddot{X} + H + F_f + F_e \quad (3.62)$$

Donc dans le cas idéal, la sortie Φ_p génère la dynamique inverse du manipulateur en contact avec son environnement.

Remarque

A partir des équations (3.60) et (3.62) on peut écrire :

$$\varphi_p = K_p^{-1} [D \ddot{X} + H + F_f + F_e - K_v \dot{\varphi}_p] \quad (3.63)$$

Comparativement avec l'équation (3.55) du cas précédent (figure 3.36), on peut dire qu'avec ce réseau, les perturbations dues à la dynamique du manipulateur se trouvent réduites de K_p^{-1} , ce qui constitue un avantage par rapport au cas précédent.

3.7.4 Architecture du réseau de neurones :

Les réseaux de neurones des deux structures de commande précédentes sont choisis les mêmes. Leur architecture est donnée par la figure 3.38.

L'entrée du réseau est donc le vecteur :

$$X_{RN} = \begin{bmatrix} X_d^T(t) & \dot{X}_d^T(t) \end{bmatrix} \quad (3.64)$$

Les β_i ($i=1,2,3$) à la sortie du réseau représentent des gains de réglage en amplitude.

L'adaptation des poids de ce réseau est obtenue en minimisant la fonction coût donnée par :

$$J = \frac{1}{2} v^T v \quad (3.65)$$

Où v est le vecteur signal à minimiser donné par l'équation (3.53) ou bien (3.61).

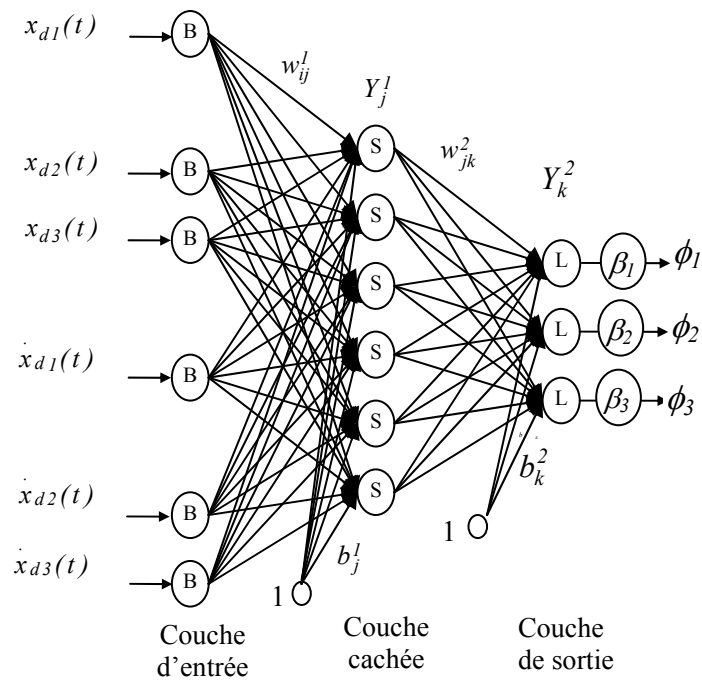


Figure 3.38 : structure du réseaux de neurones

Si on dérive l'équation (3.65) et on utilise les équations (3.54) et (3.58), le gradient J des deux schémas de commande sera :

$$\frac{\delta J}{\delta w} = \frac{\delta v^T}{\delta w} v = -\frac{\delta \Phi_u^T}{\delta w} v \quad (\text{schéma 1}) \quad (3.66)$$

$$\frac{\delta J}{\delta w} = \frac{\delta v^T}{\delta w} v = -\frac{\delta \Phi_p^T}{\delta w} v \quad (\text{schéma 2}) \quad (3.67)$$

L'adaptation des poids est obtenue en utilisant l'algorithme de rétro-propagation du gradient donnée par l'équation suivante:

$$\Delta w(t) = \eta \frac{\delta \phi^T}{\delta w} v + \mu \Delta w(t-1) \quad (3.68)$$

Où η est le taux d'adaptation et μ est le facteur d'oubli ou momentum [HAY 99].

Avec, $\phi = \Phi_u$ (schéma 1) ou bien $\phi = \Phi_p$ (schéma 2).

3.7.5 Analyse des performances en présence d'un milieu élastique

Pour simplifier l'analyse de la stabilité des deux structures de commande proposées ci-dessus, on suppose que la compensation de la dynamique du manipulateur par le réseau de neurones est parfaitement réalisée, dans ce cas, le système en boucle fermée sera régi à tout instant par l'équation hybride de la dynamique de l'erreur :

$$K_v \dot{E}_p + K_p E_p + K_i \int_0^t E_f d\tau = 0 \quad (3.69)$$

Si on considère que l'environnement est de type élastique régi par l'équation linéaire (3.14) et en substituant l'équation (3.15) alors, (3.69) peut se réécrire ainsi :

$$K_v \dot{E}_p + K_p E_p + K_i \cdot K_e \int_0^t E_p d\tau = -K_i \int_0^t F_d d\tau + K_i \cdot K_e \int_0^t (X_d - X_e) d\tau \quad (3.70)$$

Le système (3.70) est du second ordre. Sa stabilité peut être analysée en terme de stabilité du système non forcé donnée par l'équation suivante :

$$K_v \dot{X} + K_p X + K_i \cdot K_e \int_0^t X d\tau = 0 \quad (3.71)$$

Le robot étant en contact avec son environnement, pour analyser l'équation (3.71) couplée par la matrice K_e , nous avons maintenu les hypothèses sur les repères de référence et ceux attachés au plan décrites dans le paragraphe 3.3.1. Grâce aux équations 3.18-3.21 on peut déduire que l'équation (3.71), projeté dans le repère attaché au plan, est régi par le système d'équations suivant:

$$\begin{aligned} k_v \dot{x}_n + k_p x_n + k_i k_e \int_0^t x_n d\tau &= 0 \\ k_v \dot{x}_1 + k_p x_1 &= 0 \\ k_v \dot{x}_2 + k_p x_2 &= 0 \end{aligned} \quad (3.72)$$

k_v, k_p, k_i et k_e éléments scalaires des matrices K_v, K_p, K_i et K_e .

Donc le système (3.71) est découplé dans l'espace des variables x_n, x_1, x_2 .

En appliquant le critère de Routh aux équations (3.72), on en déduit la condition nécessaire et suffisante pour que le système soit stable :

$$k_p, k_v, k_i > 0. \quad (3.73)$$

Remarque importante

On remarque que la condition de stabilité dans ce cas est indépendante de la valeur de la raideur de l'environnement. Ce résultat constitue un grand avantage par rapport à la structure dynamique parallèle précédente dont la stabilité dépend directement de cette valeur de la raideur.

3.7.5.1 Cas d'un environnement de position mal connue

Dans le cas où la position de l'environnement est parfaitement connue, l'équation (3.70) peut se réécrire en remplaçant X_d par X_e ($X_d = X_e$), avec X_e position de l'environnement. Si l'environnement est de position mal connue, on aura :

$$X_d = X_e + \delta X_e = X_e' \quad (\delta X_e \text{ représente l'incertitude sur } X_e).$$

Dans ce cas l'équation (3.70) s'écrira:

$$K_v \dot{E}_p + K_p E_p + K_i K_e \int_0^t E_p d\tau = -K_i \int_0^t F_d d\tau - K_v \delta \dot{X}_e - K_p \delta X_e \quad (3.74)$$

L'application de la transformée de Laplace à l'équation (3.74) nous permet d'écrire:

$$E_p(p) = -[K_v p^2 + K_p p + K_e K_i]^{-1} \times [K_i F_d + (K_v p^2 + K_p p) \delta X_e(p)] \quad (3.75)$$

Avec $\delta X_e(p)$ l'entrée et $E_p(p)$ la sortie

Si on suppose que la perturbation de position est de type échelon ($\delta X_e(p) = \frac{1}{p}$), et que $F_d = 0$, le point d'équilibre du système (3.75) sera:

$$E_p(\infty) = \lim_{p \rightarrow 0} p E_p(p) = 0 \Rightarrow X_\infty \rightarrow X_e \Rightarrow F_e = K_e (X_\infty - X_e) \rightarrow 0$$

Ce résultat montre que dans le cas idéal (compensation parfaite de la dynamique du robot par le réseau de neurones), le système reste compliant et la loi de commande en effort domine la loi de commande en position.

3.7.5.2. Cas d'un environnement curviligne

Dans le cas d'un environnement curviligne on peut dire qu'avec les mêmes hypothèses sur l'environnement que précédemment (paragraphe 3.3.1.2), les conditions de stabilité restent valables.

3.7.6 Résultats de simulation

Les tests en simulation des deux structures de commande neuronales proposées ci-dessus visent à évaluer leurs performances et à comparer les résultats obtenus. Pour se faire, nous avons testées ces deux structures dans le cas du suivi d'un cercle sur un plan incliné (figure 3.10) en considérant les conditions de simulation suivantes.

La raideur de l'environnement et la charge portée par l'organe terminal sont supposées au départ connues ($k_e = 10^4$ N/m, $m = 2$ Kg). Les gains de réglage des contrôleurs ainsi que les paramètres du réseau de neurones sont choisis comme suit:

$K_p = 5000 \times I, K_v = 200 \times I, K_i = 0.8 \times I, \eta = 0.00003 ; \mu = 0$ et $w(0) = 0, \beta_i = 1$ (schéma 1)

Et

$K_p = 1200 \times I, K_v = 60 \times I, K_i = 0.8 \times I, \eta = 0.0008 ; \mu = 0$ et $w(0) = 0, \beta_i = 1/3 \times 10^{-3}$ (schéma 2).

La période d'échantillonnage T_e est choisie égale à 1 ms.

Les résultats de simulations sont représentés par les figures 3.39-3.42.

On remarque un net suivi de la trajectoire désirée et de l'effort appliqué (avec un temps de réponse en effort inférieur à la seconde) (figure 3.41) en introduisant le compensateur neuronal selon la structure donnée par la figure 3.37 (schéma 2). Par contre les erreurs restent assez importantes (figure 3.39), en particulier pour le cas du suivi de l'effort appliqué, en utilisant la structure donnée par la figure 3.36 (schéma 1). Les erreurs selon les directions x et z représentent la pénétration de l'effecteur dans le mur.

Donc on peut conclure que les performances du schéma de commande donné par la figure 3.37 sont meilleures que celles du schéma donné par la figure 3.36.

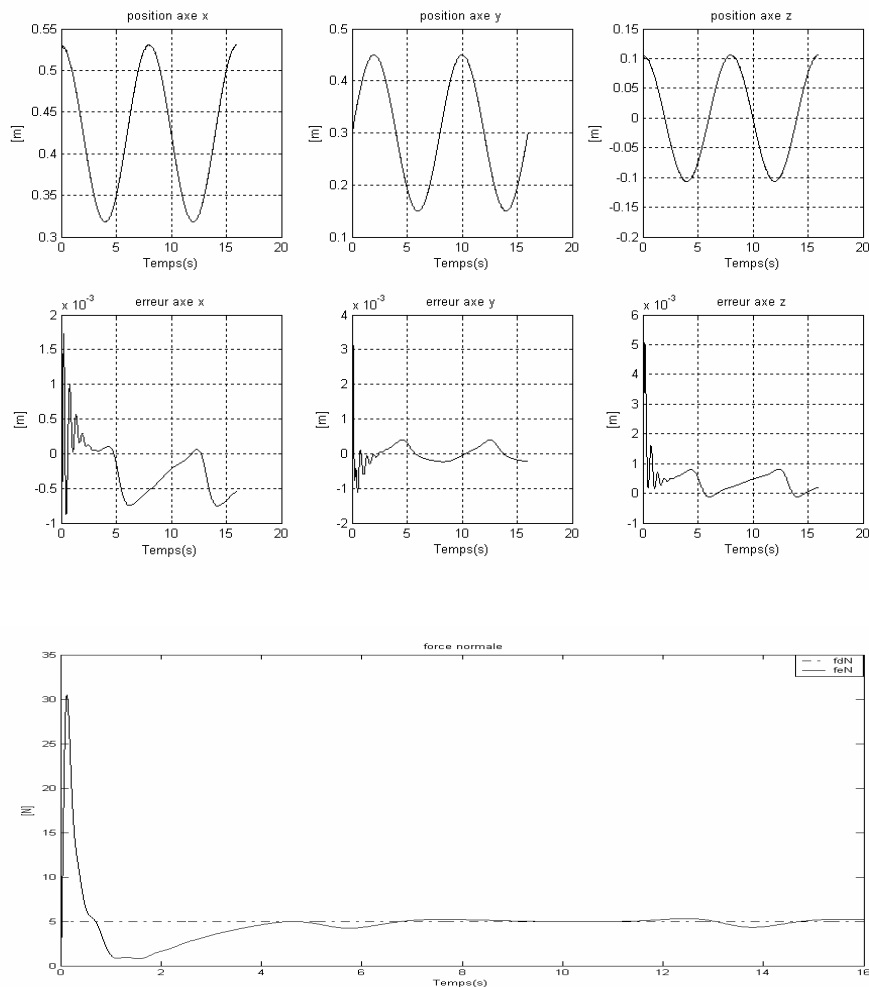


Figure 3.39 : Evolution de la position et l'effort appliqué (schéma 1)

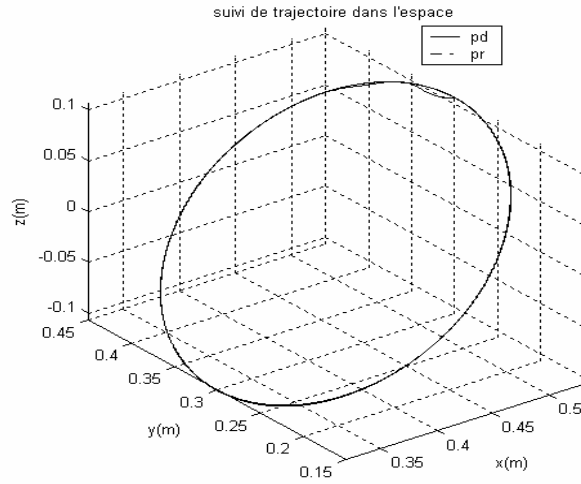


Figure 3.40 : Evolution de la trajectoire dans l'espace (schéma 1)

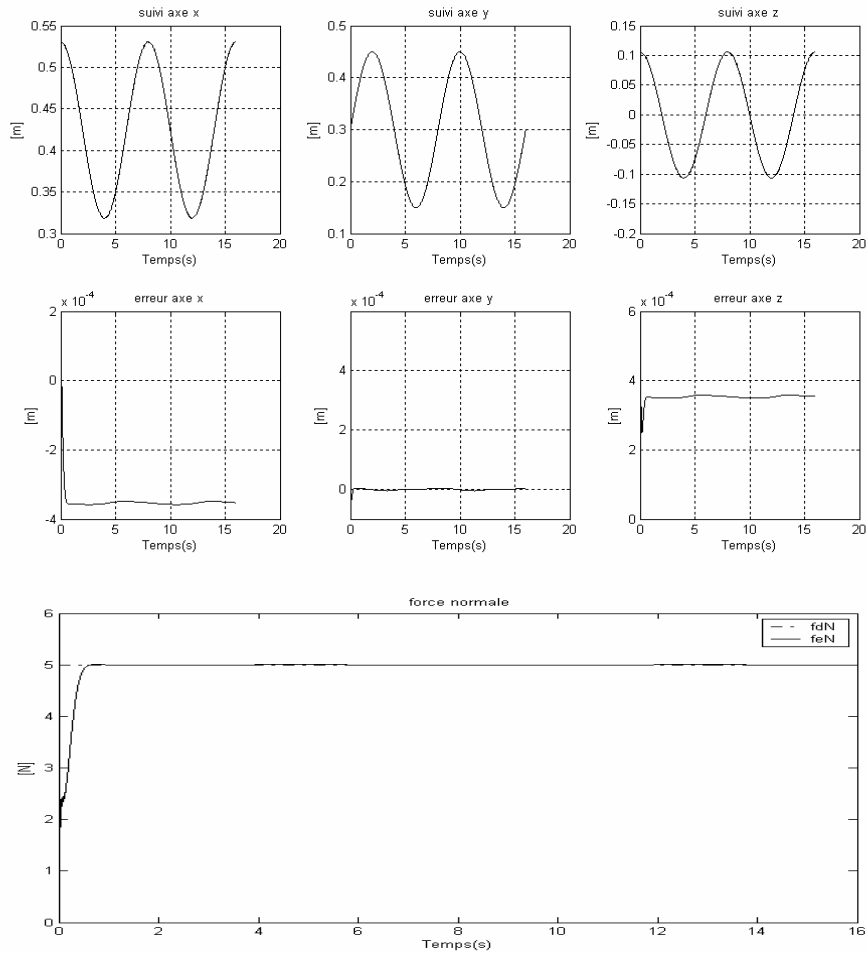


Figure 3.41 : Evolution de la position et l'effort appliqué (schéma 2)

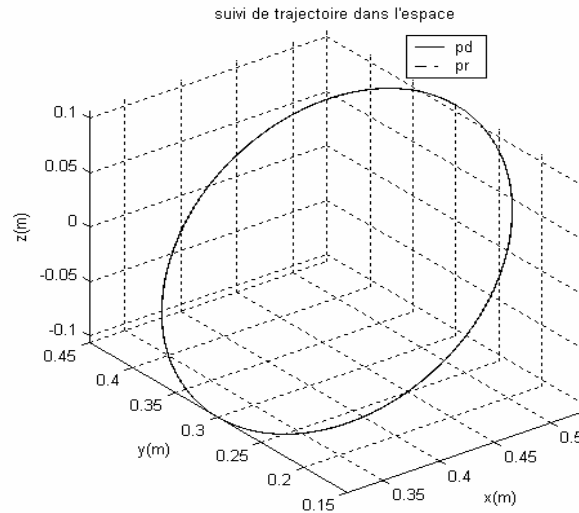


Figure 3.42 : Evolution de la trajectoire dans l'espace (schéma 2)

Pour évaluer sa robustesse (schéma 2), nous avons introduit les incertitudes suivantes sur les modèles du robot ainsi que son environnement comme suit:

- Charge portée par le robot $m=10\text{kg}$
- Frottements au niveau des articulations, modélisés par:

$$\tau_f = 0.8 \dot{q} + 0.5 \text{sign}(\dot{q})$$

- Raideur de l'environnement variable et discontinue donnée par :

$$k_e = \begin{cases} 10000 \text{ N/m} & 1 < t \leq 4 \\ 40000 \text{ N/m} & 4 < t \leq 8 \\ 20000 \text{ N/m} & 8 < t \leq 12 \\ 30000 \text{ N/m} & 12 < t \leq 16 \end{cases}$$

Les résultats de simulation obtenus sont donnés par les figures 3.43 et 3.44.

On remarque que les incertitudes introduites sur le système n'ont pas influé sur le suivi de la trajectoire du robot. Dans le cas du suivi de l'effort appliqué, on remarque que l'introduction des variations brusques de la raideur de l'environnement induit des pics suivis d'oscillations sur l'amplitude, mais rapidement amorties au bout d'un temps inférieur à 0.5s. La figure 3.44 représente l'évolution de la trajectoire dans l'espace.

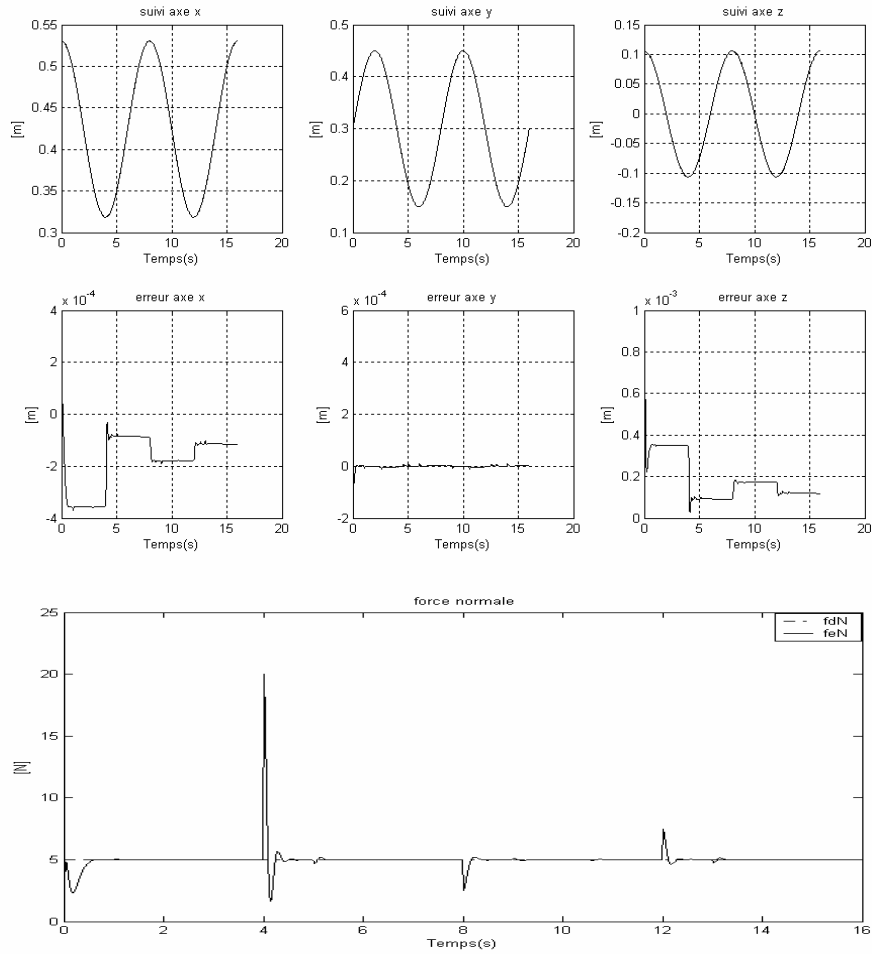


Figure 3.43 : Evolution de la position et de l'effort appliqué (schéma 2)

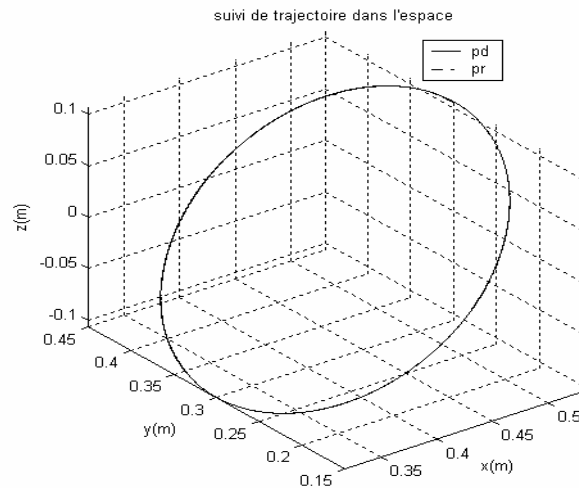


Figure 3.44 : Evolution de la trajectoire dans l'espace (schéma 2)

En comparant ces résultats avec ceux obtenus avec la structure neuronale parallèle basée sur le modèle dynamique proposé ci-dessus (figure 3.2), on remarque que cette structure donne de meilleures performances pour la réponse en effort.

3.8 Structure neuronale parallèle avec boucle de commande en effort adaptative

Les performances des tests de robustesse obtenus avec le schéma de la figure 3.37 pour le suivi de l'effort appliqué ont été très satisfaisantes pour une raideur de l'environnement qui varie sur une faible proportion comme il est montré sur la figure 3.43 ci-dessus.

Cependant de mauvaises performances peuvent apparaître si cette raideur varie sur une grande proportion. En réalité cette dégradation des performances est due à la caractéristique de l'équation du second ordre (3.71) régissant le système en boucle fermée qui dépend directement de cette raideur, qui change en réalité le placement des pôles de cette équation. Donc il est pratiquement impossible dans ce cas d'assurer de bonnes performances si seulement des gains fixes K_i du contrôleur intégral sont utilisés dans la boucle de commande en effort.

Dans ce qui suit nous proposons une approche de commande adaptative appliquée à la boucle de commande en effort de la structure parallèle de la figure 3.37 permettant de se débrouiller face à n'importe quelle variation de cette raideur.

L'approche proposée est basée sur l'estimation en ligne de la raideur de l'environnement permettant ainsi d'adapter le gain de réglage du contrôleur intégral de la boucle de commande en effort de cette structure (figure 3.37).

L'idée vient du fait que les valeurs scalaires de la matrice K_i de l'équation (3.69) peuvent être réécrites comme suit :

$$k_i = \hat{k}_i \times \frac{I}{k_e^{est}} \quad (3.76)$$

Avec,

\hat{k}_i : une valeur fixe

k_e^{est} : la valeur estimée de la raideur de l'environnement.

Le concept de base de cette architecture est qu'un module d'identification de la raideur de l'environnement est introduit dans la boucle de commande en effort de la figure 3.37 permettant d'estimer à tout instant la raideur du contact k_e . Ainsi les paramètres du contrôleur de l'effort deviennent adaptatifs vis-à-vis de la variation de la raideur de l'environnement.

Les forces de contact F_e sont modélisées par l'équation (3.14). Si on utilise (3.19), on peut écrire (3.14) comme suit:

$$F_e = k_e n n^T (X - X_e) \quad (3.77)$$

Dans la pratique les efforts de contact sont mesurés par des capteurs d'effort, donc la raideur de l'environnement peut être estimée de la manière suivante :

Supposons que $F_e(k-1)$ et $F_e(k)$ sont deux forces de contact mesurées successives délivrées par des capteurs de forces. Supposons que $X(k-1)$ et $X(k)$ sont respectivement

les points du contact aux étapes $(k-1)$ et k , la raideur de l'environnement peut être estimée comme suit :

$$k_e^{est} = [nn^T (X(k) - X(k-1))]^{-1} \times [(F_e(k) - F_e(k-1))] \quad (3.78)$$

Où, $X(k) \neq X(k-1)$

Cet estimateur ne s'applique que lorsque l'effecteur rentre en contact avec l'environnement. par conséquent il doit satisfaire la condition suivante pour rentrer en action [SON 95]:

$$\text{sgn}[(X(k) - X(k-1)) \times (F_e(k) - F_e(k-1))] > 0$$

En d'autres termes, c'est seulement sous cette condition que k_e^{est} est exploitée pour calculer la nouvelle valeur de la matrice des gains K_i donnée par:

$$K_i = \hat{k}_i \times \text{diag} \left\{ \frac{1}{k_e^{est}} \right\} \quad (3.79)$$

Cependant un problème numérique peut apparaître lorsque le système approche de l'état d'équilibre lorsque $X(k) \approx X(k-1)$ ou $F_e(k) \approx F_e(k-1)$. Nous vérifions cette condition en examinant k_e^{est} s'il est devenu extrêmement petit ou bien grand alors nous l'omettons et nous maintenons la valeur initiale de k_e^{est} . Nous assurons ainsi la stabilité du système (équation (3.72)). La figure 3.45 présente la nouvelle structure neuronale avec l'adaptation des paramètres de réglage de la boucle de commande en effort.

Résultats de simulation

Pour montrer l'efficacité de notre approche, nous avons gardé les mêmes conditions de simulation que la partie précédente sauf que l'on considère une variation plus large de la raideur de l'environnement :

$$k_e = \begin{cases} 10000 \text{ N/m} & 1 < t \leq 4 \\ 100000 \text{ N/m} & 4 < t \leq 8 \\ 20000 \text{ N/m} & 8 < t \leq 12 \\ 50000 \text{ N/m} & 12 < t \leq 16 \end{cases}$$

Les paramètres du contrôleur en force sont sélectionnés comme suit :

$$\hat{k}_i = 8000 \text{ et la valeur initiale de } k_e^{est} = 10^4 \text{ N/m}$$

Les résultats sont reportés dans les figures 3.46 et 3.47. On remarque qu'aux instants des variations brusques de la raideur de l'environnement, avec un profil discontinu, des pics en effort sont observés dont la valeur est assez importante lorsque la raideur subit une grande variation mais rapidement amortie et sans oscillation en un temps inférieur à 0.5s en utilisant notre contrôleur adaptatif (figure 3.47). Nous remarquerons qu'avec un contrôleur intégral

dont les paramètres sont choisis fixes, des oscillations critiques apparaissent au moment de cette grande variation de la raideur (figure 3.46).

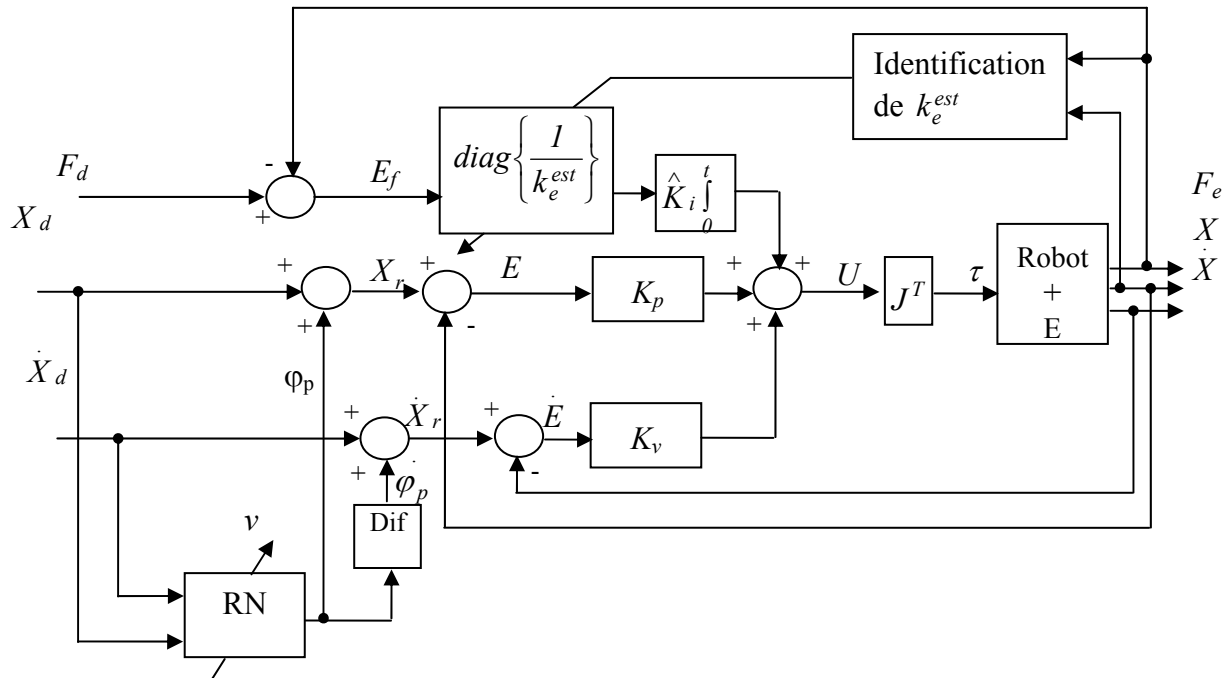


Fig. 3.45. Structure neuronale parallèle avec une boucle de commande en effort adaptative.

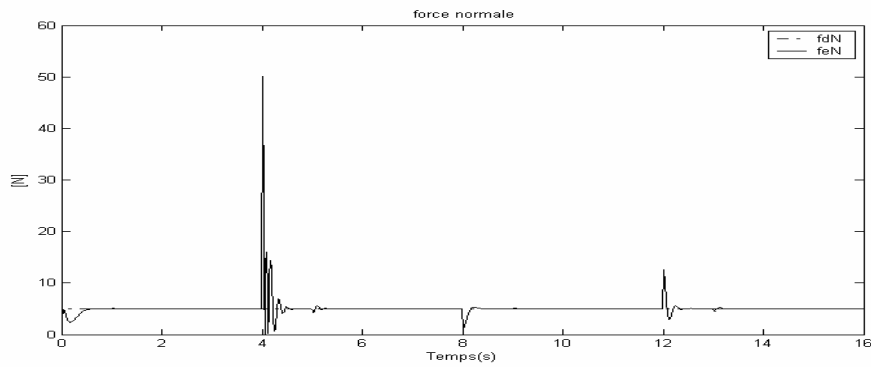


Fig. 3.46. Suivi de l'effort sans contrôleur adaptatif

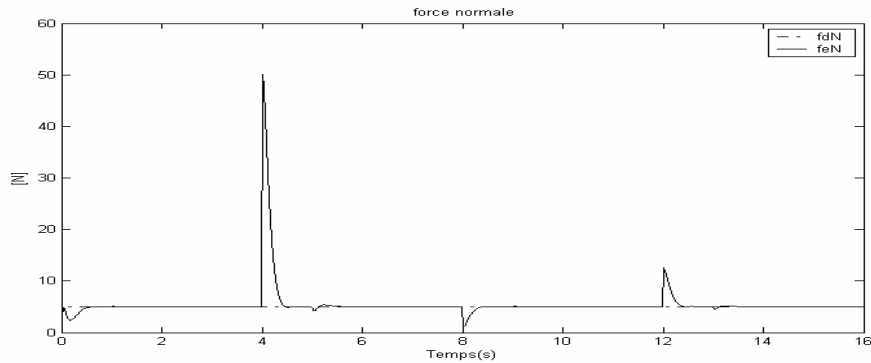


Fig. 3.47. Suivi de l'effort avec contrôleur adaptatif

Cas d'une surface curviligne

Nous avons fait une extension de la méthode proposée au cas d'une surface curviligne (figure 3.18) en considérant le cas critique d'une forte variation de sa raideur k_e comme suit :

$$k_e = \begin{cases} 10000 \text{ N/m} & 1 < t \leq 6 \\ 100000 \text{ N/m} & 6 < t \leq 12 \end{cases}$$

Les paramètres de réglage sont maintenus et les résultats de simulation sont représentés par les figures 3.48 - 3.50.

On remarque que la force normale appliquée à la surface génère un pic au moment de la variation brusque de la raideur mais rapidement amortie et sans oscillations avec un temps de convergence inférieur à 0.5s en utilisant le compensateur adaptatif (figure 3.50), alors qu'avec un réglage constant des paramètres de la boucle en effort des oscillations critiques apparaissent au moment de la variation de la raideur (figure 3.48). La figure 3.50 représente l'évolution de la trajectoire dans le plan.

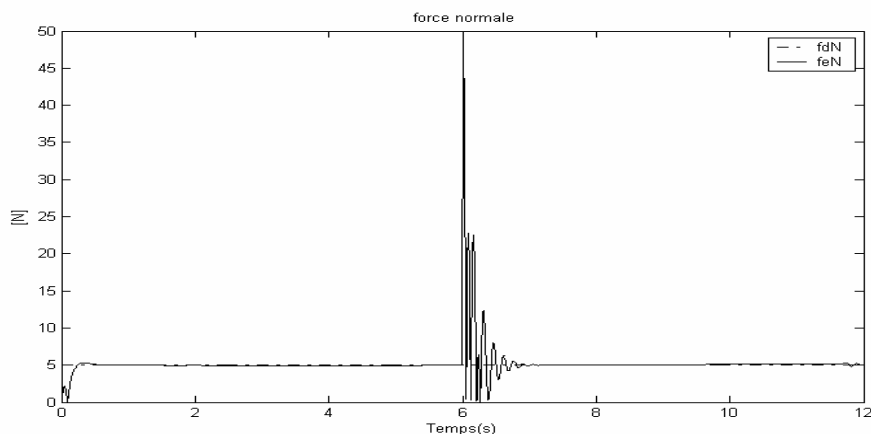


Fig.3.48. Suivi de l'effort sans contrôleur adaptatif (cas d'une surface curviligne)

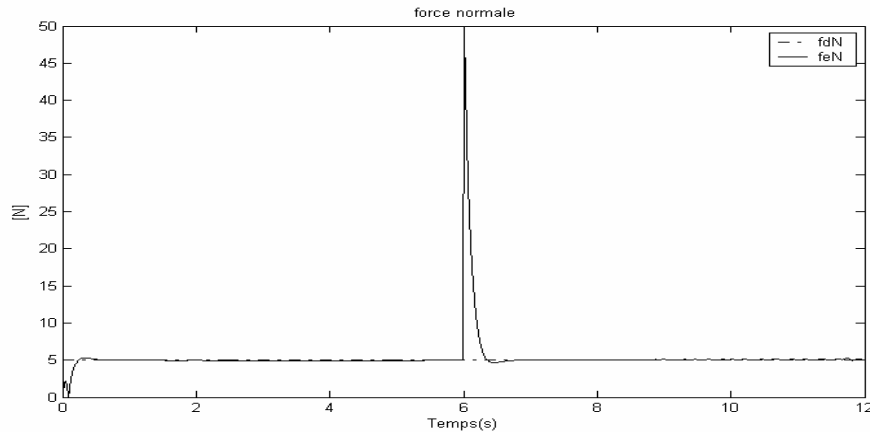


Fig. 3.59. Suivi de l'effort avec contrôleur adaptatif (cas d'une surface curviligne)

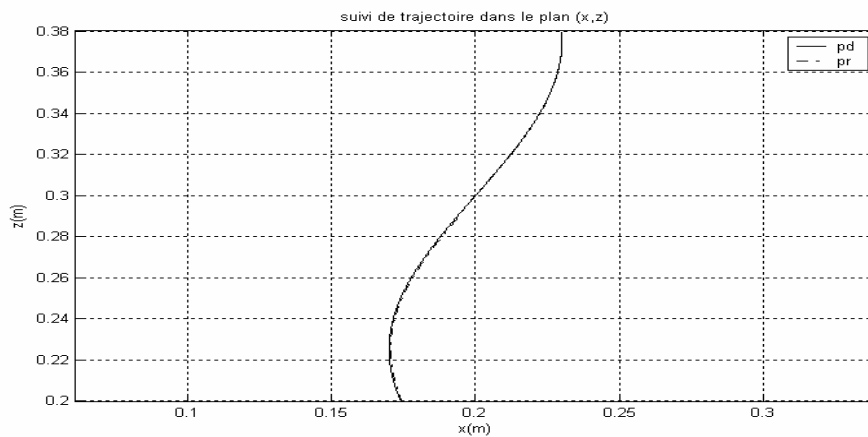


Fig.3.50: Evolution de la trajectoire dans le plan (avec contrôleur adaptatif)

Il est à noter que malgré ces grandes variations de la raideur de l'environnement, la stabilité du système est toujours restée vérifiée avec et sans le module adaptateur. Ce résultat confirme la condition théorique établie sur la stabilité du système en boucle fermée (équation 3.73) qui reste indépendante de la raideur de l'environnement.

3.9 Conclusion

Dans ce chapitre nous avons proposé de nouvelles structures de commande intelligentes parallèles force/position. Les approches proposées introduisent des techniques de compensation neuronales et floues pour améliorer les performances de deux structures parallèles conventionnelles force/position proposées dans la littérature [CHI 93, 94], appliquées dans le cas des classes de tâches du suivi de contour dans des milieux contraints et en présence des incertitudes sur la dynamique du robot et son environnement.

Dans la première partie de ce chapitre, nous avons présenté et analysé la structure dynamique parallèle force/position de CHIAVERINI [CHI 93]. Nous avons montré à travers une étude de la problématique, les faiblesses de cette structure de commande lorsque on est en présence des incertitudes sur les modèles du robot et son environnement.

Pour remédier à ce problème, nous avons associé à cette structure de commande deux compensateurs issues de l'intelligence artificielle le premiers est un compensateur neuronale permettant de supprimer les incertitudes sur le modèle dynamique du robot et le deuxième est un contrôleur adaptatif flou permettant de compenser les incertitudes sur la valeur de la raideur de l'environnement. Les résultats de simulations présentées pour plusieurs types d'environnement et de trajectoires ont montré l'efficacité de notre approche par rapport à une structure classique proposée dans la littérature [CHI 93] (précision et temps de réponse réduit).

Dans la deuxième partie de ce chapitre nous avons proposé une autre approche de commande neuronale parallèle force/position mais cette fois ci non basée sur le modèle dynamique pour l'élaboration de la commande.

Nous avons commencé par présenter la structure parallèle force/position à base d'un PID classique où nous avons montré ses limites et ses défauts. Pour résoudre ces difficultés, deux approches neuronales ont été proposées, à base de cette structure PID classique.

L'analyse de la stabilité en présence d'un milieu élastique de ces deux structures de commande, dans le cas idéal (compensation parfaite de la dynamique du robot par le compensateur neuronal) a montré une caractéristique très importante de ces deux structures par rapport à la structure dynamique précédente : la stabilité du système en boucle fermée est vérifiée indépendamment d'une connaissance préalable de la valeur de la raideur de l'environnement.

Une étude comparative en simulation de ces deux approches a montré la qualité des résultats de la deuxième approche (structure basée sur la modification de la trajectoire de référence) par rapport à la première (structure basée sur la modification du couple de commande).

Pour améliorer les performances de la réponse en effort appliqué, un module adaptatif basé sur l'estimation en ligne de la raideur de l'environnement est introduit dans la boucle de commande en effort.

Une extension des approches intelligentes développées dans ce chapitre, au cas d'une autre structure de commande conventionnelle force/position, qui possède les mêmes caractéristiques que celles de la commande parallèle fait l'objet du chapitre suivant.

Chapitre 4

Structures intelligentes de la commande en effort externe

4.1 Introduction

Au chapitre 1 nous avons présenté une structure de commande force/position, non basée sur un concept parallèle, mais qui possède une caractéristique équivalente du contrôle à la fois de la position et l'effort désiré le long d'une même direction sans mécanisme de sélection, appelée dans la littérature commande en effort externe [SHU 88].

En effet avec cette structure de commande, la compliance de l'effort appliqué est obtenue grâce à un vecteur de déplacement issu de la boucle de commande en effort qui vient modifier la trajectoire désirée du robot manipulateur au lieu de générer, comme c'était le cas avec la structure parallèle, un vecteur force qui modifie l'effort exercé sur l'organe terminal. Par ailleurs, avec cette structure, la boucle de commande en effort est définie dans l'espace cartésien alors que la boucle de commande en position peut être définie soit dans l'espace cartésien (figure 1.14) ou bien dans l'espace articulaire grâce à une transformation géométrique (figure 1.15) [PER 91].

Cependant cette structure de commande telle qu'elle est proposée dans la littérature, néglige généralement l'effet de la dynamique du manipulateur (forces d'inertie, termes de couple dus aux forces de Coriolis et centrifuges) par le fait que les vitesses de déplacement sont supposées faibles de telle sorte que l'effet des forces de contact domine les perturbations engendrées par la dynamique du robot sur le système [PER 91], ce qui n'est pas toujours vrai vis-à-vis de la diversité du type de tâches à réaliser par le robot en milieu contraint.

A notre connaissance, aucune structure dynamique permettant de découpler simultanément la position et l'effort dans l'espace cartésien n'a été proposée dans la littérature pour cette structure, et aucune approche de commande robuste n'a été proposée pour traiter à la fois les problèmes liés aux incertitudes sur la dynamique du robot et son environnement qui en réalité, affectent tous les deux les performances des suivis de trajectoires et des efforts exercés par l'effecteur sur l'environnement.

Dans le chapitre précédent nous avons proposé des approches de commande issues de l'intelligence artificielle associées à des structures de commande force/position conventionnelles basées sur un concept parallèle. Nous proposons dans ce chapitre une extension de ces approches de commande associées cette fois-ci à la structure de commande en effort externe afin d'améliorer ses performances en poursuite de trajectoire et en réponse aux efforts exercés par l'effecteur sur son environnement, lorsqu'on est en présence d'incertitudes sur la dynamique du robot et son environnement.

4.2 Structure dynamique de la commande en effort externe

Rappelons que l'idée de base mise en œuvre dans ce type de commande (figure 1.13 chapitre 1) consiste à ajouter à une consigne de position cartésienne, un déplacement relatif issu de la boucle d'effort. Cette structure fait apparaître un asservissement de position, en l'absence de contraintes dues à l'environnement, le système est alors contrôlé selon un asservissement d'effort dès que des contraintes apparaissent. En effet deux boucles imbriquées sont ainsi mises en évidence avec une hiérarchisation de la boucle d'effort par rapport à la boucle de position.

La combinaison des équations (3.8) et (4.1)-(4.2) donne l'équation dynamique suivante :

$$M_d \ddot{E}_p + K_v \dot{E}_p + K_p E = \hat{D}^{-1} (\Delta D \ddot{X} + \Delta H + F_f + \Delta F_e) \quad (4.3)$$

Où, $\Delta D = D - \hat{D}$, $\Delta H = H - \hat{H}$, $\Delta F_e = F_e - \hat{F}_e$, $E = X_c - X$ et $E_p = X_d - X$

Dans le cas idéal où le modèle dynamique du robot est connu parfaitement et que les frottements secs et visqueux sont considérés nuls c'est-à-dire : $\Delta D = \Delta H = F_f = \Delta F_e = 0$, le système global en boucle fermée est donné par la relation suivante :

$$M_d \ddot{E}_p + K_v \dot{E}_p + K_p E = 0 \quad (4.4)$$

D'après la figure 4.1, on peut écrire :

$$E = X_c - X = X_d + \Delta X - X = E_p + \Delta X \quad (4.5)$$

Si on remplace (4.5) dans l'équation (4.4) on aura:

$$M_d \ddot{E}_p + K_v \dot{E}_p + K_p E_p + K_p \Delta X = 0 \quad (4.6)$$

La sortie du contrôleur en effort, choisi de type intégral, est donnée par:

$$\Delta X = K_i \int_0^t E_f d\tau \quad (4.7)$$

Avec,

K_i est une matrice diagonale scalaire de réglage de type λI .

Donc l'équation (4.6) devient donc:

$$M_d \ddot{E}_p + K_v \dot{E}_p + K_p E_p + K_p K_i \int_0^t E_f d\tau = 0 \quad (4.8)$$

Comparativement à la structure de commande dynamique parallèle proposée par CHIAVERINI (voir chapitre 3), on remarque qu'on retrouve le même type d'équation de la dynamique de l'erreur du système en boucle fermée (voir équation (3.13)).

Donc on peut déduire que cette structure de commande est aussi basée sur la stratégie de la dominance de la boucle de commande en effort sur celle en position. Il est à noter ici que, si la tâche exécutée par l'effecteur assure la condition $F_d = F_e (E_f = 0)$ alors le système devient contrôlable par la commande dynamique par découplage non linéaire classique (très cité dans la littérature) définie dans l'espace cartésien [BOI 88].

4.2.1 Analyse des performances en présence d'un milieu élastique

Les performances de cette structure de commande sont analysées en considérant que les forces de contact robot/environnement, comme c'était le cas dans le chapitre 3, sont de type élastique, régies par le modèle donné par l'équation (3.14). La substitution de l'équation (3.15) dans l'équation (4.8) nous permet d'écrire:

$$\begin{aligned} M_d \ddot{E}_p + K_v \dot{E}_p + K_p E_p + K_p K_i \cdot K_e \int_0^t E_p d\tau \\ = - K_p K_i \int_0^t F_d d\tau + K_p K_i \cdot K_e \int_0^t (X_d - X_e) d\tau \end{aligned} \quad (4.9)$$

On a donc un système d'équations du troisième ordre.

La stabilité de ce système peut être analysée en terme de stabilité du système non forcé donnée par l'équation suivante :

$$M_d \ddot{X} + K_v \dot{X} + K_p X + K_p K_i \cdot K_e \int_0^t X d\tau = 0 \quad (4.10)$$

Où X est la position de l'organe terminal.

On remarque que l'équation (4.10) est couplée grâce à la matrice K_e (matrice de raideur constante et symétrique).

Le robot étant en contact avec son environnement, si on considère que n dénote le vecteur unitaire le long de la normale à la surface plane du contact, la matrice K_e peut être décomposée comme suit (voir chapitre 3 paragraphe 3.3.1) :

$$K_e = R_{rot} \text{diag}(k_e, 0, 0) R_{rot}^T = k_e n n^T. \quad (4.11)$$

Avec R_{rot} la matrice de rotation du repère attaché au plan par rapport au repère de référence et k_e la valeur de la raideur selon la normale au plan.

Cette décomposition de la matrice K_e conduit à un découplage du système dans le repère lié à la surface de l'environnement. Ainsi l'équation du système non forcé (4.10) devient:

$$\begin{aligned} m_d \ddot{x}_n + k_v \dot{x}_n + k_p x_n + k_p k_i k_e \int_0^t x_n dx &= 0 \\ m_d \ddot{x}_1 + k_v \dot{x}_1 + k_p x_1 &= 0 \\ m_d \ddot{x}_2 + k_v \dot{x}_2 + k_p x_2 &= 0 \end{aligned} \quad (4.12)$$

Avec x_n, x_1, x_2 sont les composantes de X dans le repère attaché au plan.

L'application du critère de Routh au système 4.12 nous permet de déduire la condition nécessaire et suffisante de la stabilité du système global donnée comme suite:

$$k_p, k_v, m_d, k_i > 0, \text{ et } k_v > m_d k_i k_e \quad (4.13)$$

Avec, k_v, k_p, k_i, k_e et m_d respectivement éléments des matrices K_v, K_p, K_i, K_e et M_d .

Remarque

On remarque que la condition de stabilité dépend directement de la connaissance de la valeur de la raideur de l'environnement.

4.2.1.1 Cas d'un environnement de position mal connue

Dans le cas où la position de l'environnement est parfaitement connue, l'équation (4.9) peut se réécrire en remplaçant X_d par X_e ($X_d = X_e$), avec X_e position de l'environnement.

Si l'environnement est de position mal connue, on aura, $X_d = X_e + \delta X_e = X_e'$, où δX_e représente l'incertitude sur X_e .

Dans ce cas l'équation (4.9) s'écrira:

$$\begin{aligned} M_d \ddot{E}_p + K_v \dot{E}_p + K_p E_p + K_p K_i \cdot K_e \int_0^t E_p d\tau \\ = -K_p K_i \int_0^t f_d d\tau - M_d \delta \ddot{X}_e - K_v \delta \dot{X}_e - K_p \delta X_e \end{aligned} \quad (4.13)$$

L'application de la transformée de Laplace à l'équation (4.13) nous permet d'écrire:

$$\begin{aligned} E_p(p) = - \left[M_d p^3 + K_v p^2 + K_p p + K_p K_e K_i \right]^{-1} \\ \times \left[K_p K_i F_d + (M_d p^3 + K_v p^2 + K_p p) \delta X_e(p) \right] \end{aligned} \quad (4.14)$$

Avec $\delta X_e(p)$ est l'entrée et $E_p(p)$ est la sortie.

Si on suppose que l'entrée (la perturbation) est un échelon ($\delta X_e(p) = \frac{1}{p}$), et que $F_d = 0$, le point d'équilibre du système (4.14) sera:

$$E_p(\infty) = \lim_{p \rightarrow 0} p E_p(p) = 0 \Rightarrow X_\infty \rightarrow X_e \Rightarrow F_e = K_e (X_\infty - X_e) \rightarrow 0$$

Ce résultat prouve que la structure de commande en effort externe est une structure compliant, et que la boucle d'effort domine celle en position grâce à la présence d'un intégrateur dans la boucle de commande en effort.

4.2.1.2 Cas d'un environnement curviligne

Dans le chapitre précédent nous avons montré que le choix d'une surface plane est motivé par le fait que localement on peut avoir une bonne approximation pour une courbure régulière. Dans ce cas le modèle des efforts de contact donnés par l'équation 3.14 peut se réécrire sous la forme [CHI 93]:

$$F_e = K_e(X_e(X))(X - X_e) \quad (4.15)$$

Avec,

$$K_e(X_e) = k_e n(X_e) n^T(X_e)$$

Sous ces conditions, la stabilité locale en chaque point de la surface considérée reste valable.

Cependant les performances de cette structure de commande dynamique proposée dépendent principalement de la connaissance exacte des modèles du robot et de son environnement. Chose qui n'est pas toujours vérifiée, c'est-à-dire que ΔD , ΔH , ΔF_e et F_f sont souvent différentes de zéro.

Nous proposons dans ce qui suit une nouvelle structure dynamique de la commande en effort externe basée sur l'approche neuronale proposée dans le chapitre 3 permettant de compenser les incertitudes sur ce modèle dynamique.

4.3 Structure neuronale de la commande dynamique en effort externe [FER 09]

Dans le chapitre précédent, nous avons proposé une technique de compensation neuronale, permettant de supprimer l'effet des incertitudes sur les performances désirées dans le cas d'une architecture de commande basée sur un concept parallèle. Une extension de cette technique au cas de la structure dynamique de la commande en effort externe fait l'objet de cette partie de ce chapitre.

Le réseau de neurones qu'on propose ici est conçu pour générer à sa sortie un vecteur du signal de commande ϕ qui vient s'ajouter à la loi de commande en position donnée par l'équation (4.2). En effet cette conception est basée sur la modification du vecteur de commande U de la structure précédente (figure 4.1) par un vecteur de commande neuronal auxiliaire pour compenser les effets des perturbations dues aux incertitudes sur le modèle dynamique du robot en contact avec son environnement.

Cette nouvelle structure neuronale de la commande en effort externe que nous proposons ici est donnée par la figure 4.2 suivante :

L'architecture du réseau proposé reste la même que celle proposée dans le cas de la structure dynamique parallèle (voir figure 3.3) sauf ses entrées qui changent et deviennent dans ce cas comme suit:

$$X_{RN} = \begin{bmatrix} X_c^T(t) & \dot{X}_d^T(t) & \ddot{X}_d^T(t) & F_e^T(t) \end{bmatrix} \quad (4.20)$$

La loi d'adaptation des poids du réseau de neurones est obtenue en minimisant la fonction quadratique dépendante du signal v de l'équation (4.18) donné par :

$$J = \frac{1}{2} v^T v \quad (4.21)$$

La dérivée de l'équation (4.21) par rapport au poids w donne

$$\frac{\delta J}{\delta w} = \frac{\delta v^T}{\delta w} v = -\frac{\delta \phi^T}{\delta w} v \quad (4.22)$$

L'adaptation des poids est obtenue en utilisant l'algorithme de rétro-propagation du gradient donnée par l'équation suivante :

$$\Delta w(t) = \eta \frac{\delta \phi^T}{\delta w} v + \mu \Delta w(t-1) \quad (4.23)$$

Où η est le taux d'adaptation et μ est le facteur d'oubli ou momentum [HAY 99].

4.4 Résultats de simulations

Afin de vérifier l'efficacité de la structure neuronale de la commande dynamique en effort externe proposée dans cette partie, et de comparer les résultats avec ceux obtenus au chapitre 3 précédent (structure neuronale dynamique parallèle), nous avons choisi de maintenir les mêmes conditions de simulations que précédemment.

Les paramètres de réglage du système pour les trois simulations présentées sont fixés comme suit:

$$M_d = I, K_p = 1200 \times I, K_v = 60 \times I, K_i = 6.66 \cdot 10^{-4} \times I, \eta = 0.0008, \mu = 0.9, w(0) = 0,$$

et $T_e = 5\text{ms}$.

Simulation 1

Rappelons que cette simulation (voir figure 3.4 du chapitre 3) consiste à tester la compliance de la structure de commande proposée vis-à-vis d'une erreur due à la méconnaissance de la position de l'environnement et en présence des perturbations suivantes :

- Charge portée par le robot $m = 10\text{kg}$
- Raideur de l'environnement plus grande $k_e = 5 \times 10^4 \text{ N/m}$:

Les résultats de simulation obtenus avec la structure de la figure 4.1 sont donnés par les figures 4.3 et 4.5 et ceux obtenus avec la structure de la figure 4.2 sont donnés par les figures 4.4 et 4.6. On remarque une nette dégradation des performances avec la structure classique (réponses oscillatoires amplifiées) alors que les réponses sont nettement meilleures en utilisant le compensateur neuronal (réponses oscillatoires amorties avec temps de réponse inférieur à 0.5s). Les figures 4.4 et 4.6 représentant l'évolution des signaux des perturbations du vecteur v (équation (4.18)) montrent comment ces derniers sont réduits par l'approche neuronale que nous avons proposée.

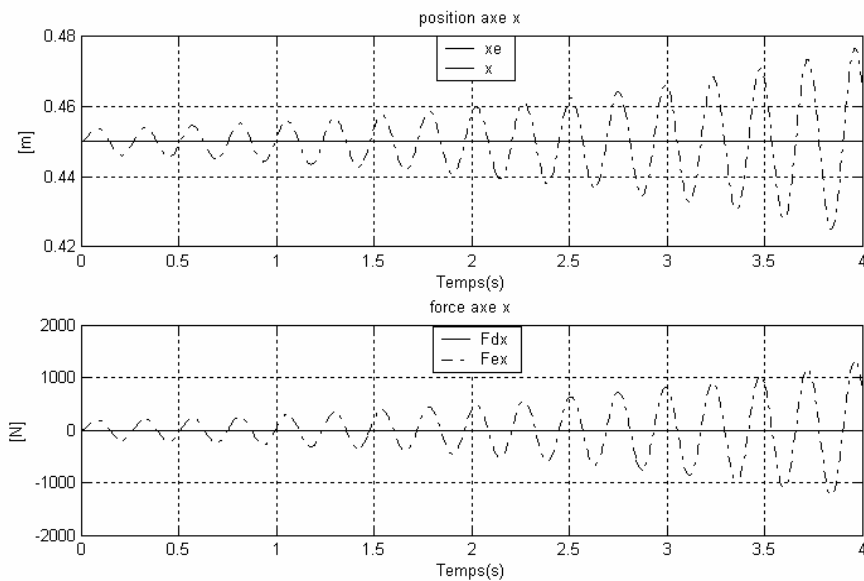


Figure 4.3 : Evolution de la position et de l'effort (sans compensateur neuronal)

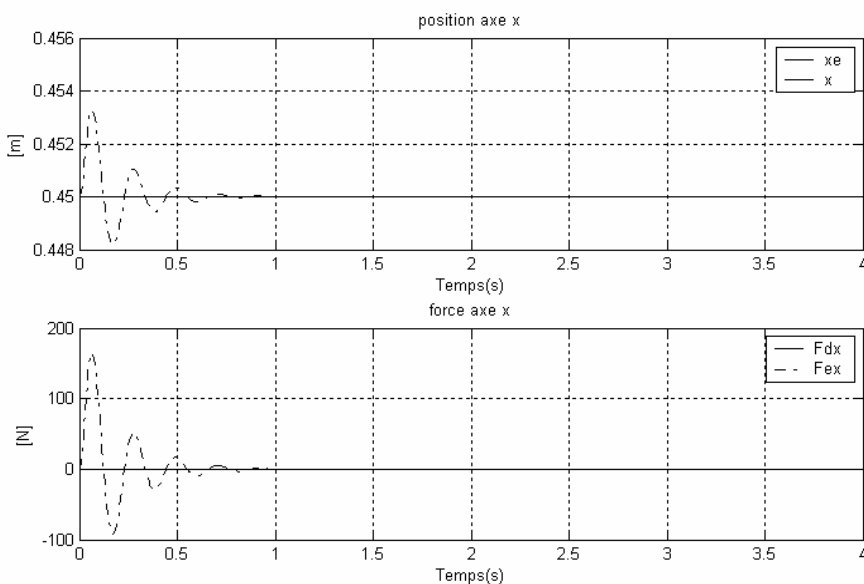


Figure 4.4 : Evolution de la position et de l'effort (avec compensateur neuronal)

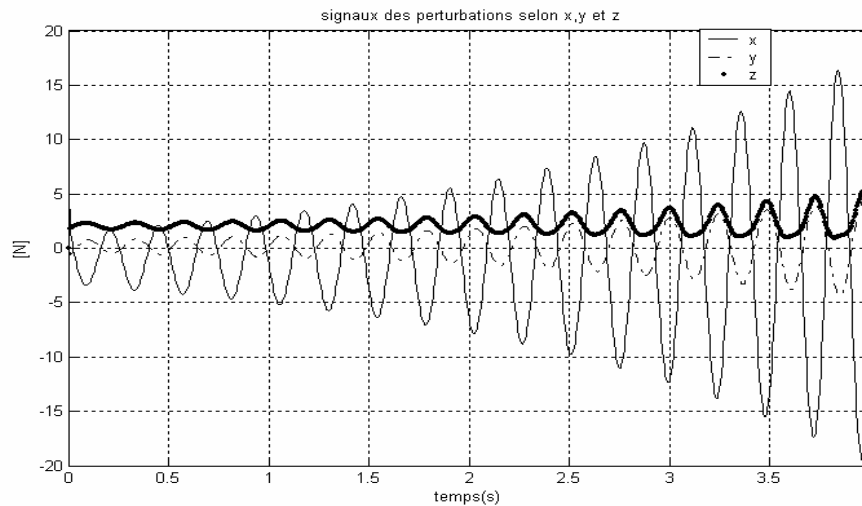


Figure 4.5 : Evolution des signaux des perturbations (sans compensateur neuronal)

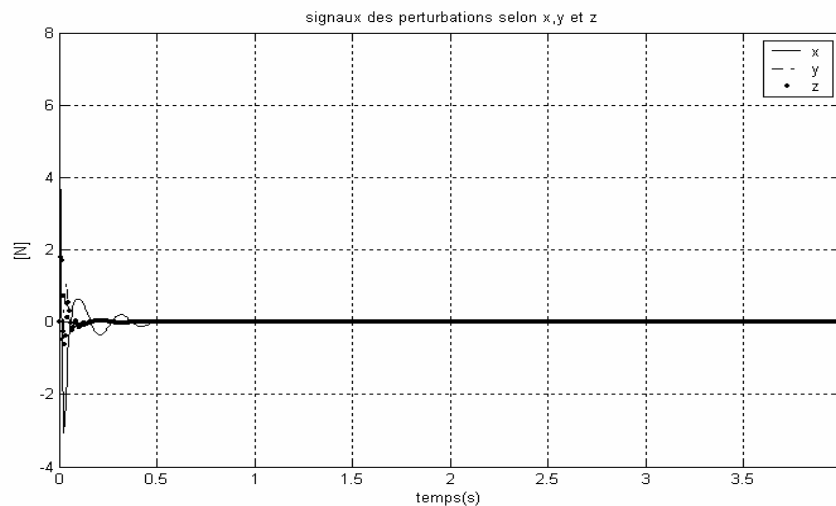


Figure 4.6 : Evolution des signaux des perturbations (avec compensateur neuronal)

Simulation 2

Cette deuxième simulation effectuée vise à évaluer les performances de la commande proposée dans le cas du suivi d'un cercle sur un plan incliné (voir figure 3.10 du chapitre 3).

Les résultats de simulation sont représentés par les figures 4.7- 4.13. On remarque une nette amélioration du suivi de la trajectoire selon la direction y (mouvement libre) (figure 4.7 et 4.8) en introduisant le compensateur neuronal (les erreurs selon la direction x et z représentent la pénétration de l'effecteur dans le mur).

Dans le cas du suivi de l'effort appliqué (figure 4.9 et 4.10), on remarque que l'introduction des variations brusques de la raideur de l'environnement induit des pics suivis d'oscillations d'amplitude nettement amortie et de durée très faibles (inférieure à 0.5s) en introduisant le compensateur neuronal.

La comparaison des figures 4.11 et 4.12 montre de façon claire la compensation des perturbations par le réseau de neurones. La figure 4.13 représente l'évolution de la trajectoire dans l'espace avec le compensateur neuronal.

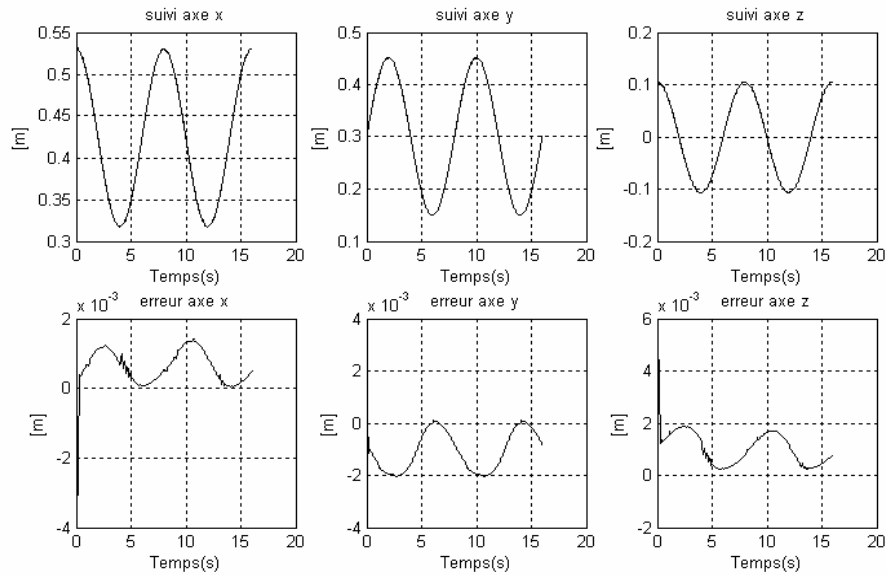


Figure 4.7 : Evolution de la position de l'effecteur (sans compensateur neuronal)

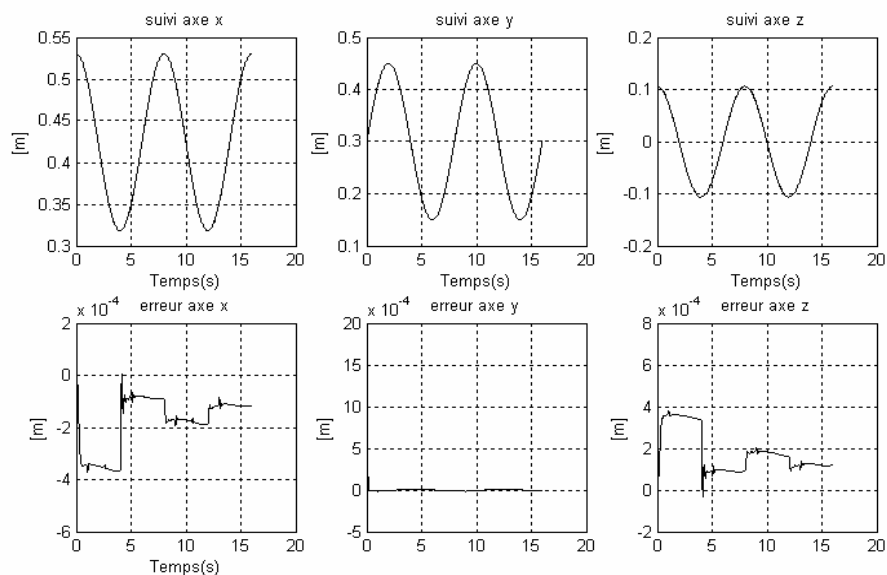


Figure 4.8 : Evolution de la position de l'effecteur (avec compensateur neuronal)

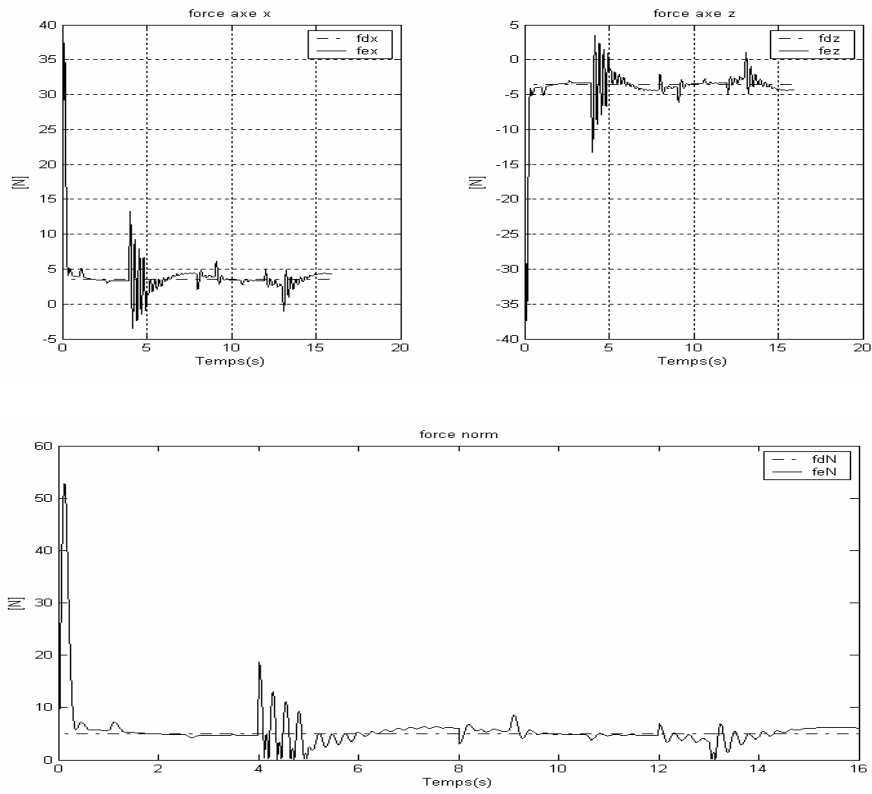


Figure 4.9 : Evolution des efforts appliqués sur la surface (sans compensateur neuronal)

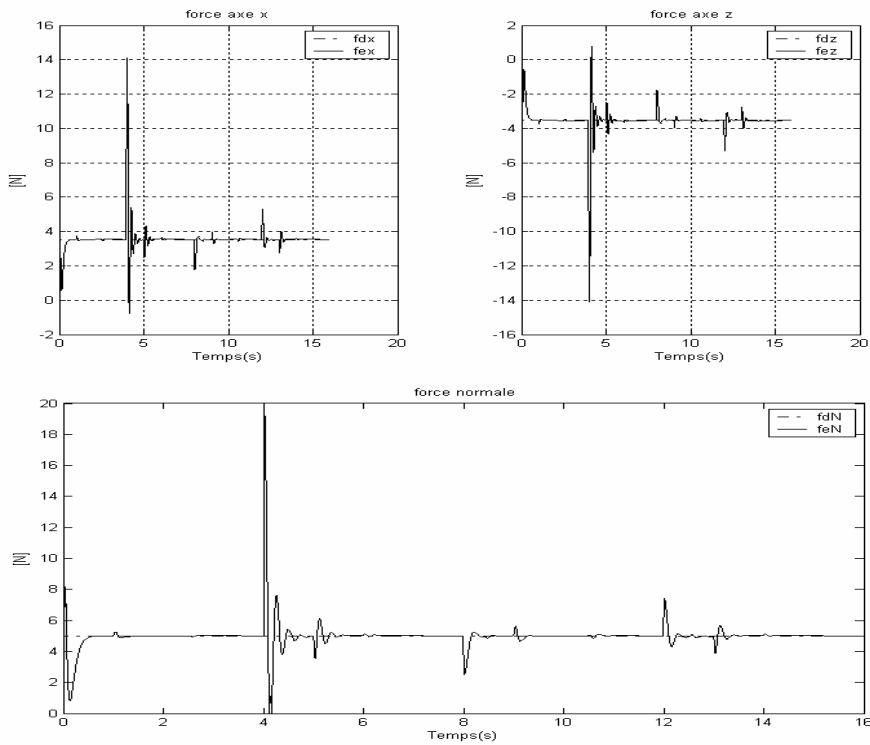


Figure 4.10 Evolution des efforts appliqués sur la surface (avec compensateur neuronal)

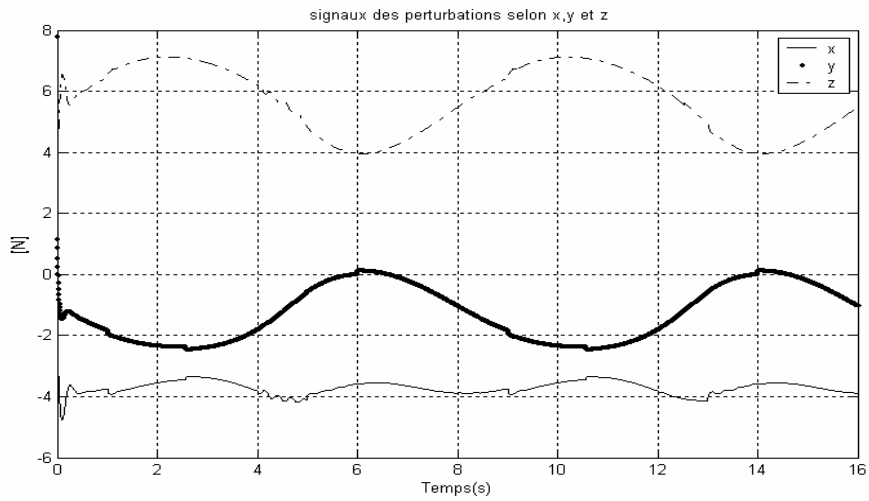


Figure 4.11 : Evolution des signaux des perturbations (sans compensateur neuronal)

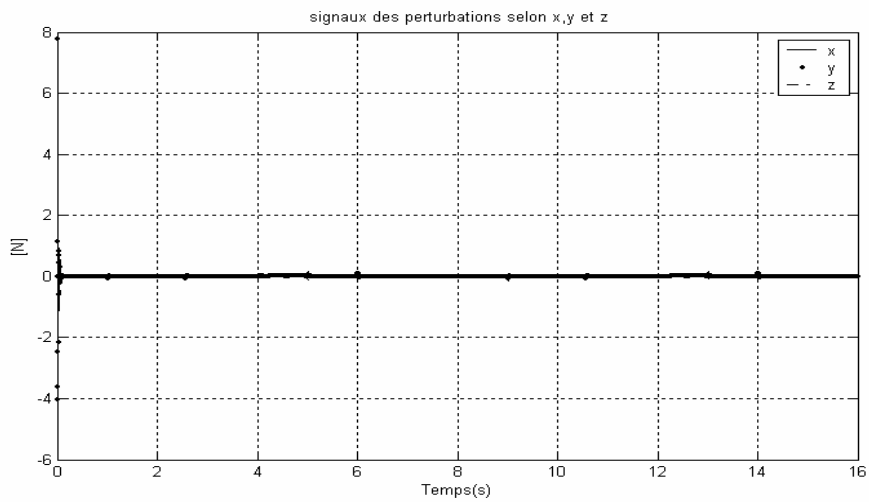


Figure 4.12 : Evolution des signaux des perturbations (avec compensateur neuronal)

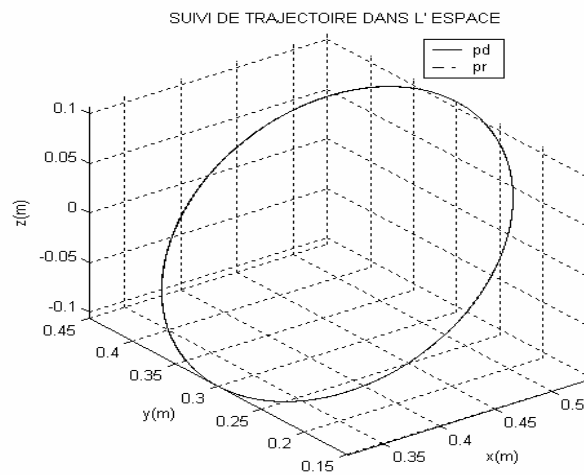


Figure 4.13 : Evolution de la trajectoire dans l'espace (avec compensateur neuronal)

Simulation 3

La troisième simulation proposée dans cette partie du travail consiste à faire suivre par l'extrémité du robot une trajectoire sur une surface curviligne (voir figure 3.18 du chapitre 3). Les résultats de simulation obtenus sont donnés par les figures 4.14-4.20.

On remarque une nette amélioration de la précision en position selon la direction fixe y (figure 4.14 et 4.15) en introduisant le compensateur neuronal (les erreurs selon la direction x et z représentent la pénétration de l'effecteur dans la surface).

Dans le cas du suivi de l'effort appliqué (figure 4.16 et 4.17), on remarque que la variation de la raideur de l'environnement a induit un pic suivi d'oscillations mais plus rapidement amorties en introduisant le compensateur neuronal.

La comparaison des figures 4.11 et 4.12 montre de façon claire la compensation des perturbations par notre approche neuronale. La figure 4.20 représente l'évolution de la trajectoire dans le plan (x, z).

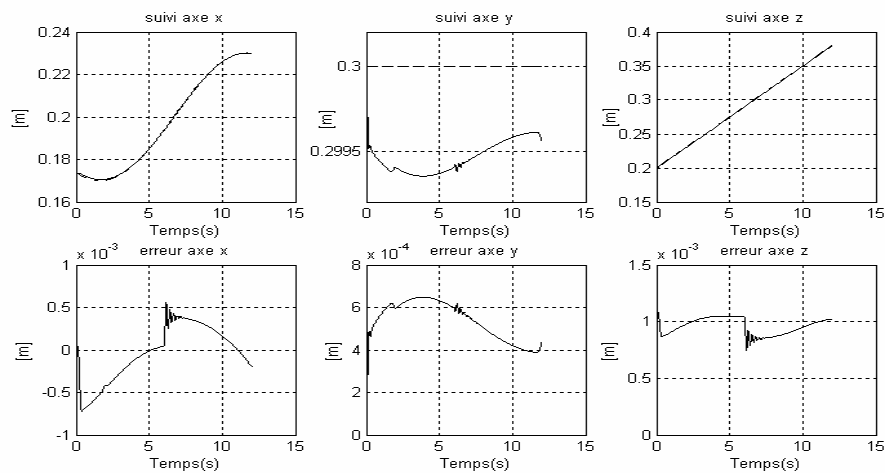


Figure 4.14 : Evolution de la position de l'effecteur (sans compensateur neuronal)

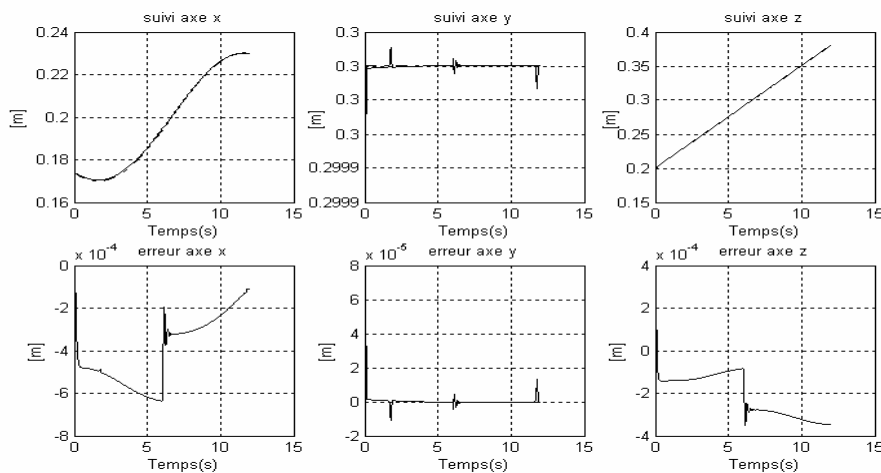


Figure 4.15 : Evolution de la position de l'effecteur (avec compensateur neuronal)

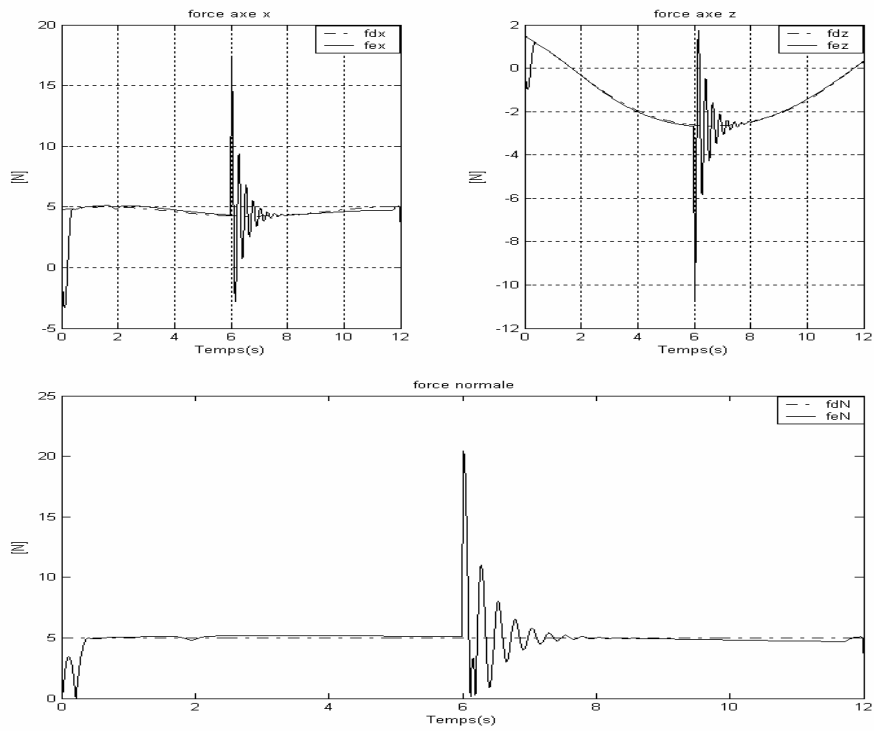


Figure 4.16 : Evolution des efforts appliqués sur la surface (sans compensateur neuronal)

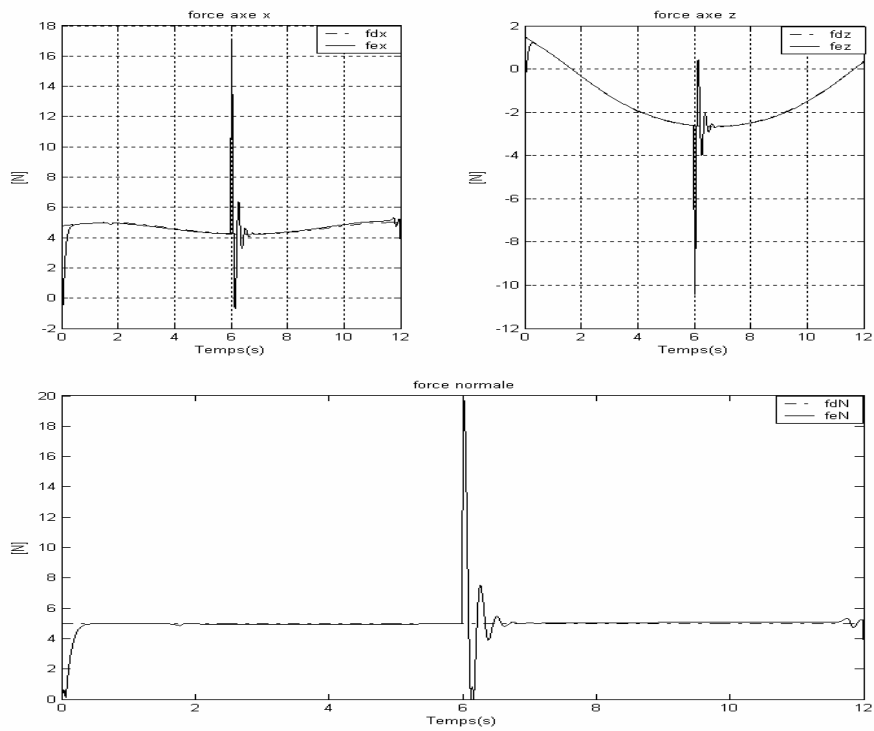


Figure 4.17 : Evolution des efforts appliqués sur la surface (avec compensateur neuronal)

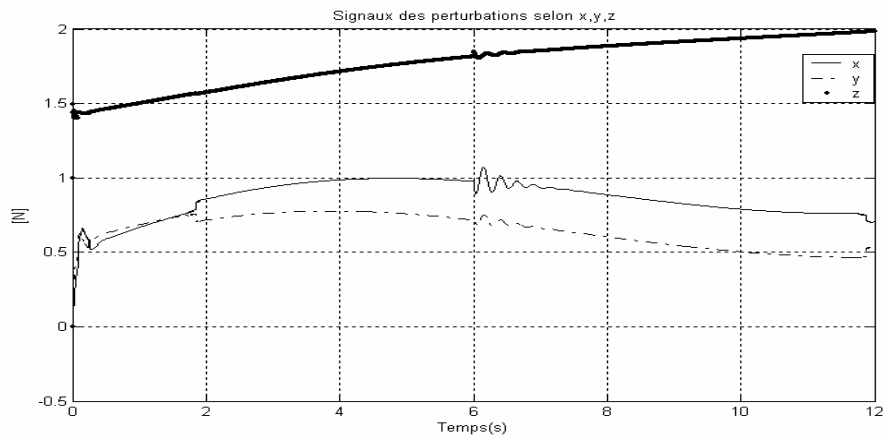


Figure 4.18 : Evolution des signaux des perturbations (sans compensateur neuronal)

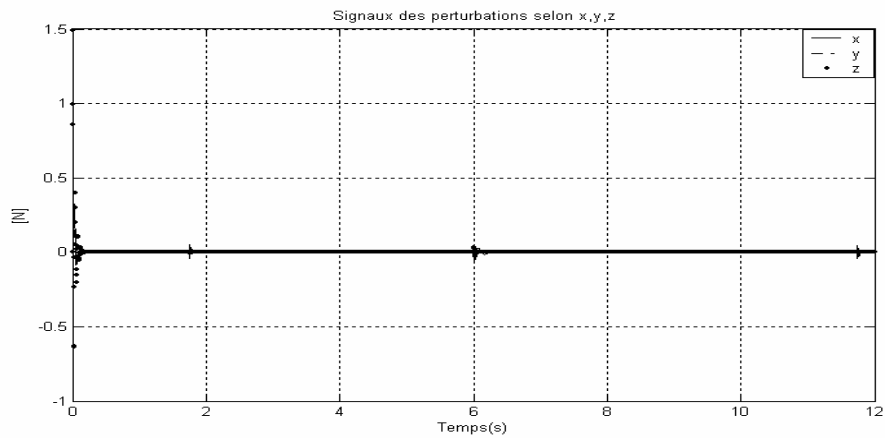


Figure 4.19 : Evolution des signaux des perturbations (avec compensateur neuronal)

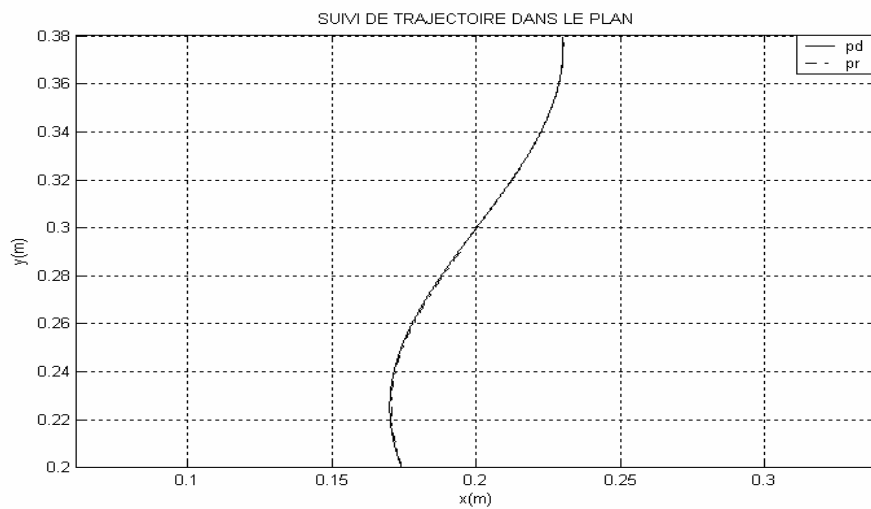


Figure 4.20 : Evolution de la trajectoire dans le plan (avec compensateur neuronal)

En comparant ces résultats avec ceux obtenus avec la structure neuronale parallèle, on en déduit que les deux structures donnent relativement les mêmes performances. Cependant nous avons constaté que pour la simulation1 (test de compliance), l'amplitude de l'effort ressenti avec cette structure de commande, en régime transitoire, est plus important que celui observé dans le cas de la structure parallèle (voir les figures 3.7 et 4.4).

Une extension de cette approche neuronale dans le cas d'une boucle de commande en position de cette structure, définie dans l'espace articulaire (figure 1.15), est proposée dans l'annexe B.

4.5 Structure neuronale externe avec une boucle de commande floue en effort

L'approche neuronale proposée ci-dessus ne traite en réalité que les effets des incertitudes du modèle dynamique du manipulateur sur les performances des réponses en position et en effort. Cependant la variation de la raideur de l'environnement qui agit comme une perturbation sur les réponses en force reste toujours sans compensation si les paramètres de réglage de la boucle de commande en effort restent choisis fixes.

Pour y remédier, nous proposons d'implémenter un contrôleur flou dans la boucle de commande en effort, le même que celui présenté dans le chapitre 3, sauf que sa sortie devient dans ce cas équivalente à un déplacement $\Delta X^{(F)}$ (figure 4.21).

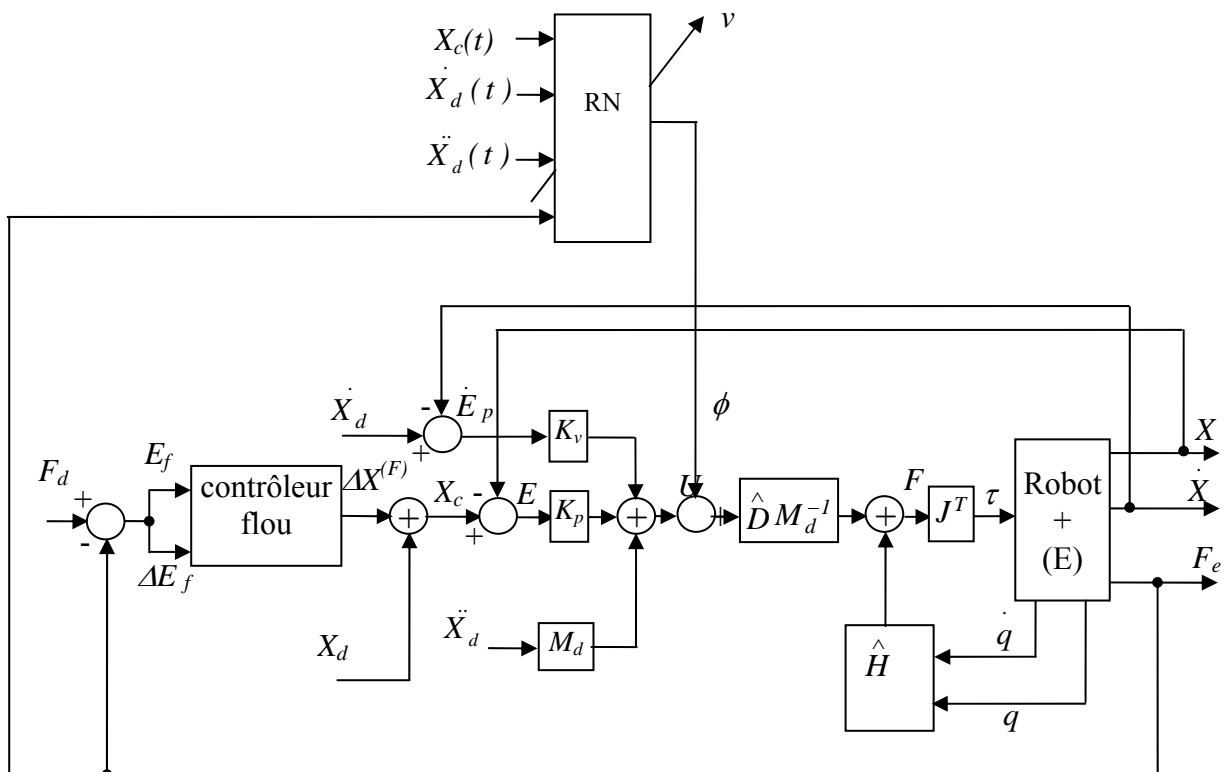


Figure 4.21 Structure neuronale de la commande externe avec boucle en effort de type PI flou

Résultats de simulation

Les paramètres du réglage restent les mêmes que le cas des simulations précédentes seulement on leur ajoute le réglage des gains du contrôleur flou, choisis comme suit :

$$G_{e_f} = 0.09, \quad G_{\Delta e_f} = 0.9 \quad \text{et} \quad G_{\Delta X^{(F)}} = 0.2$$

Nous avons repris les mêmes conditions de simulations que le cas précédent. Les réponses en effort obtenues pour les simulations 2 et 3 sont données par les figures suivantes :

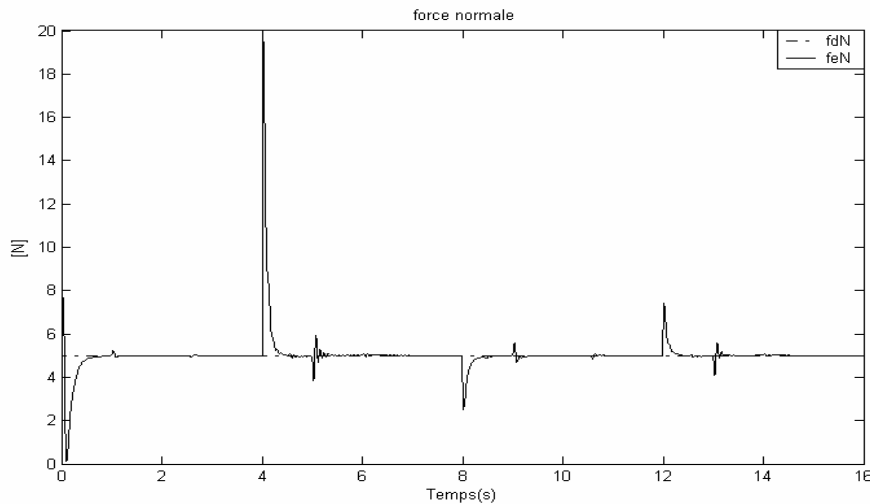


Figure 4.22 : Evolution de l'effort appliqué selon la normale (simulation 2)

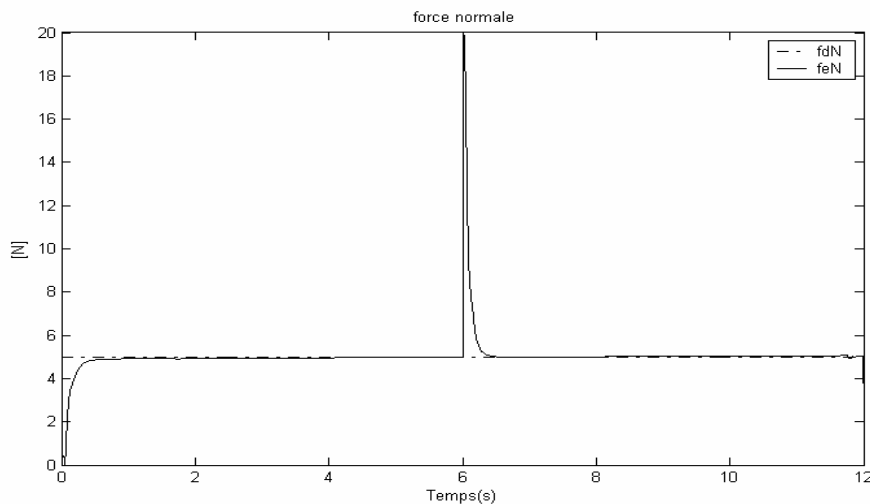


Figure 4.23 : Evolution de l'effort appliqué selon la normale (simulation 3).

On remarque qu'en utilisant ce contrôleur flou l'effort ressenti atteint son régime permanent sans oscillation et au bout d'un temps inférieur à 0.5s aux moments des variations de la raideur. Donc on peut déduire que cette structure donne des résultats nettement meilleurs que

$$G_{E_f}^* = \frac{I}{k_e^{est}} \times G_{E_f} \text{ et } G_{\Delta E_f}^* = \frac{I}{k_e^{est}} \times G_{\Delta E_f}$$

Donc si la raideur estimée de l'environnement change, les gains de normalisation changent aussi.

Les résultats de simulation donnés par la figure 4.25 montrent l'efficacité de l'approche proposée. On remarque que le retour à l'état d'équilibre se fait sans oscillation et en un temps inférieur à 0.5s, donc nettement meilleur que celui obtenu avec le contrôleur conventionnel (figure 4.4).

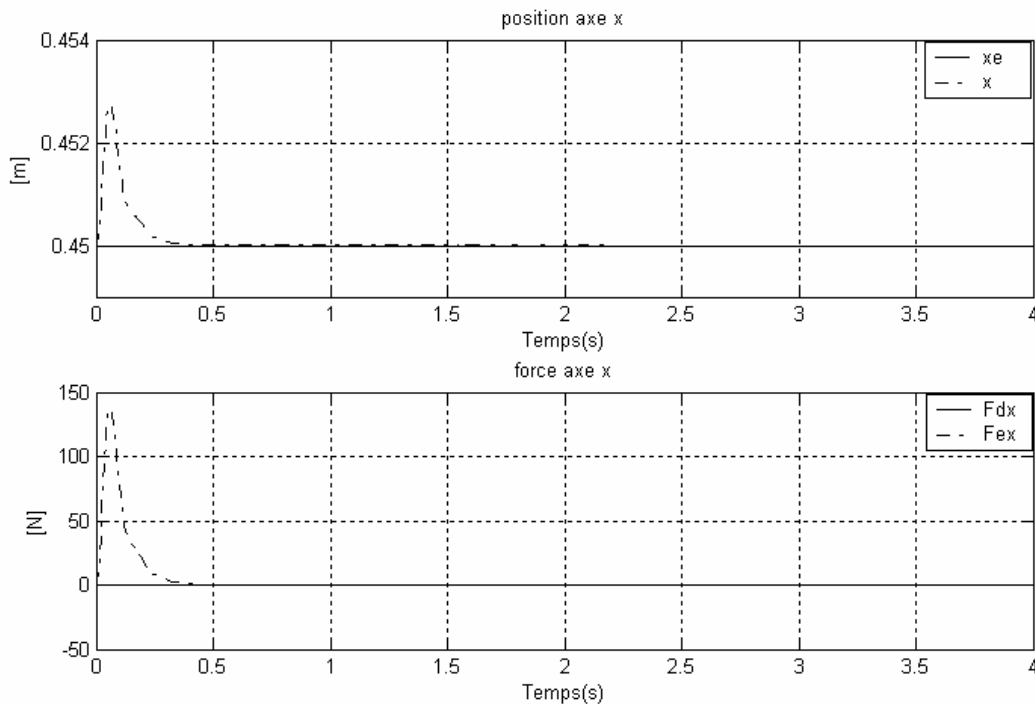


Figure 4.25 Evolution de la position et l'effort (simulation 1)

Remarques :

- Le problème qu'on peut avoir en appliquant cette nouvelle approche réside dans le calcul, par le programme de commande, de la valeur estimée de la raideur au moment où la réponse en effort atteint son régime permanent, comme cela a été expliqué dans le chapitre 3 paragraphe 3.7 (k_e^{est} qui devient extrêmement petit ou grand). Dans ce cas il faut négliger cette valeur et reprendre la valeur initiale de k_e^{est} . Nous assurons ainsi la stabilité du système global.
- On note ici que sous ces conditions, la méthode proposée ci-dessus reste valable aussi dans le cas de la structure dynamique parallèle proposée dans le chapitre 3.

4.6 Structure neuronale de la commande en effort externe non basée sur le modèle

L'objectif recherché ici est de concevoir une structure neuronale force/position permettant d'atteindre les performances souhaitées en poursuite de trajectoire et de l'effort appliqué sans avoir recours, comme le cas précédent, à l'utilisation du modèle dynamique nominal du manipulateur dans l'élaboration de la commande désirée. En effet la structure neuronale que nous proposons ici exploite une structure de commande en effort externe classique à base de contrôleurs de type PID conventionnels.

4.6.1 Structure PID de la commande en effort externe

Comme il est montré sur la figure 4.26 ci-dessous, deux boucles imbriquées sont mises en évidence avec une hiérarchisation de la boucle d'effort de type intégral (I) par rapport à celle de position de type proportionnelle dérivé (PD).

Selon cette figure la loi de commande en position est donnée par :

$$U = K_p (X_c - X) + K_v (\dot{X}_d - \dot{X}) \quad (4.24)$$

Avec K_p, K_v sont des matrices diagonales de réglage de type λI .

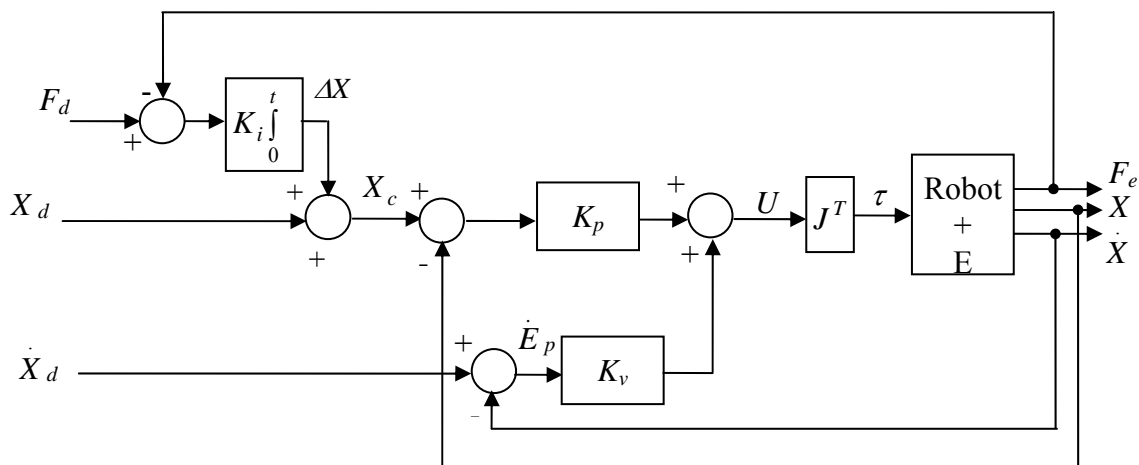


Figure 4.26 : Structure PID de la commande en effort externe dans l'espace cartésien

En combinant les équations (4.24) et (3.8), l'équation en boucle fermée est donnée par:

$$K_p E + K_v \dot{E}_p = D \ddot{X} + H + F_f + F_e \quad (4.25)$$

Avec, $E_p = X_d - X$ et $E = X_c - X$

Or d'après la figure 4.26,

$$E = X_c - X = X_d + \Delta X - X = E_p + \Delta X \quad (4.26)$$

Si on remplace (4.26) dans l'équation (4.25) on aura:

$$K_p E_p + K_v \dot{E}_p + K_p \Delta X = D \ddot{X} + H + F_f + F_e \quad (4.27)$$

Sachant que la loi de commande en effort (LCF) choisi s'écrit :

$$\Delta X = K_i \int_0^t E_f d\tau \quad (4.28)$$

Avec, K_i est une matrice diagonale scalaire de réglage de type λI .

Dans ce cas l'équation (4.27) devient donc:

$$K_p E_p + K_v \dot{E}_p + K_p K_i \int_0^t E_f d\tau = D \ddot{X} + H + F_f + F_e \quad (4.29)$$

On remarque que l'on retrouve le même type d'équation de la dynamique de l'erreur que le cas de la commande parallèle non basée sur le modèle dynamique (chapitre 3 paragraphe 3.6).

Cependant les performances de cette structure de commande sont affectées par les effets de la dynamique du bras manipulateur en contact avec son environnement (terme de droite de l'équation 4.29), régi par un modèle non linéaire et fortement couplé. Pour compenser cette dynamique indésirable du manipulateur, nous proposons dans ce qui suit une approche neuronale implémentée autour de la trajectoire de référence de la structure PID précédente.

4.6.2 Approche neuronale basée sur la technique de modification de la trajectoire de référence

Dans le chapitre 3 nous avons montré la qualité des performances en suivi en force et en position obtenues en utilisant l'approche de commande neuronale basée sur la technique de compensation par modification de la trajectoire de référence dans le cas d'une structure parallèle.

Dans cette partie du travail nous proposons une extension de cette approche dans le cas de la structure en effort externe à base du PID classique donnée ci-dessus (figure 4.26). La nouvelle structure neuronale de la commande en effort externe que nous proposons ici est donnée par la figure 4.27. Le réseau de neurones implémenté ici est conçu pour identifier et en même temps compenser en ligne la dynamique du robot en contact avec son environnement.

L'idée (voir figure 4.27) est de générer un vecteur signal φ_p à la sortie du réseau de neurones, de calculer son dérivé, qui viennent s'ajouter respectivement aux vecteurs des trajectoires de consigne en position et en vitesse, qui se trouvent ainsi tous les deux modifiées. Les poids de ce réseau sont adaptatifs grâce à la minimisation d'une fonction coût

prédéfinie. La dimension de ce réseau dépend du nombre du degré de libertés de la tâche désirée.

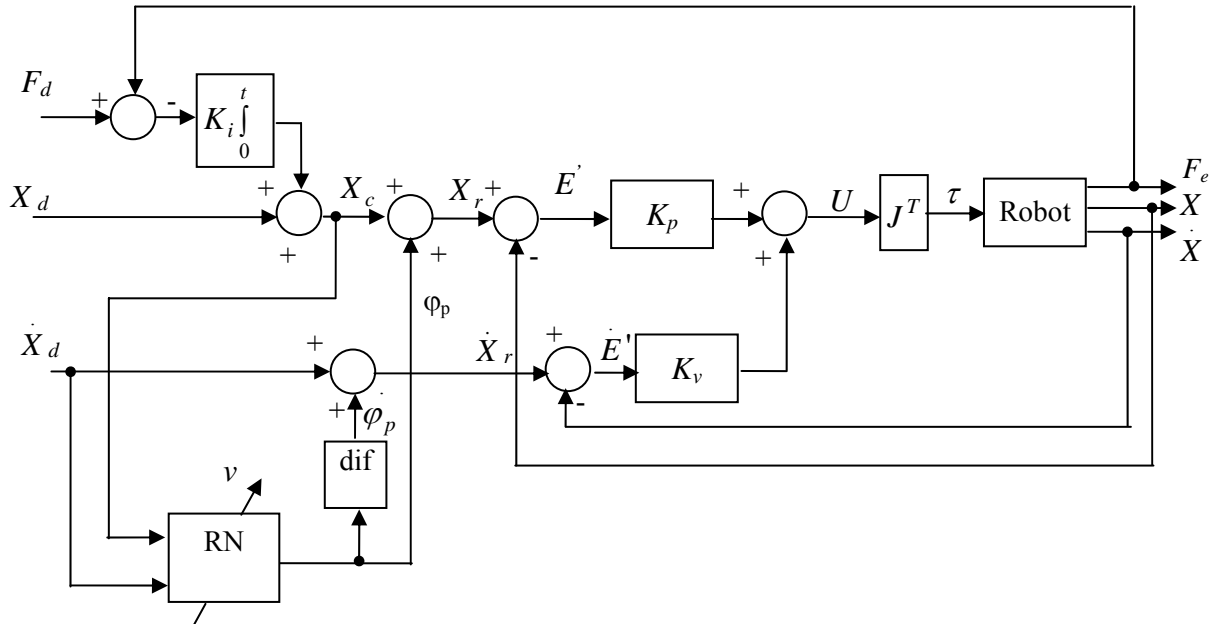


Figure 4.27 : Structure neuronale de la commande en effort externe dans l'espace cartésien

Avec cette nouvelle approche, la loi de commande en position devient donc:

$$U = K_p (X_c + \varphi_p - X) + K_v (\dot{X}_d + \dot{\varphi}_p - \dot{X}) \quad (4.30)$$

La combinaison de l'équation (4.30) et (3.8) nous donne l'équation en boucle fermée suivante :

$$K_v \dot{E}_p + K_p E = D \ddot{X} + H + F_f + F_e - (K_p \varphi_p + K_v \dot{\varphi}_p) \quad (4.31)$$

On pose:

$$\Phi_p = K_p \varphi_p + K_v \dot{\varphi}_p$$

L'objectif recherché est de générer Φ_p pour faire tendre l'équation (4.31) vers zéro.

Si on pose

$$v = K_v \dot{E}_p + K_p E \quad (4.32)$$

Alors, dans le cas idéal lorsque $v=0$ on aura :

$$\Phi_p = D \ddot{X} + H + F_f + F_e \quad (4.33)$$

Le réseau proposé est de même architecture que celui de la structure parallèle non basée sur le modèle dynamique (chapitre3), sauf ses entrées qui deviennent dans ce cas comme suit:

$$X_{RN} = \begin{bmatrix} X_c^T(t) & \dot{X}_d^T(t) \end{bmatrix} \quad (4.34)$$

Donc le réseau de neurones proposé est le suivant :

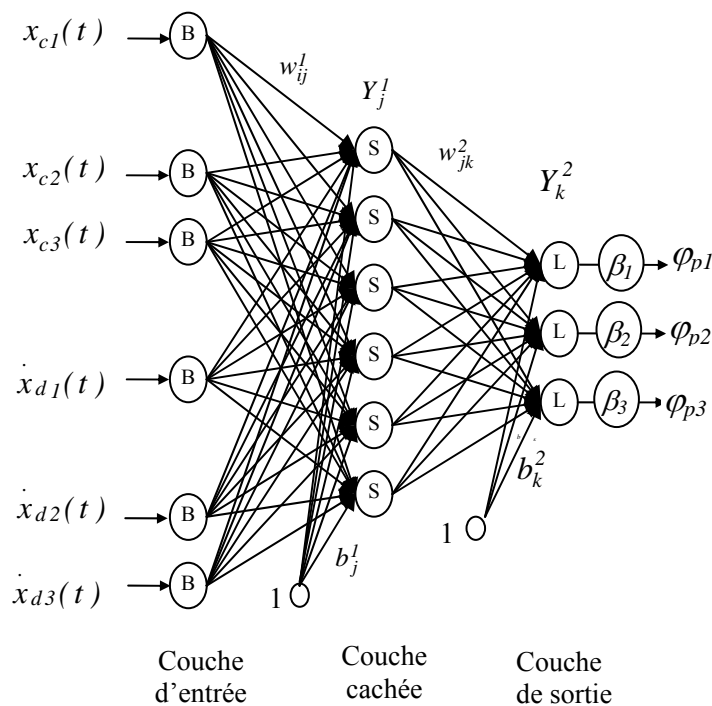


Figure 4.28 : Structure du réseau de neurones

La loi d'adaptation des poids du réseau est obtenue en minimisant la fonction quadratique dépendante du signal v de l'équation (4.32) donnée par :

$$J = \frac{1}{2} v^T v \quad (4.35)$$

La dérivée de l'équation (4.35) par rapport au poids w donne :

$$\frac{\delta J}{\delta w} = \frac{\delta v^T}{\delta w} v = -\frac{\delta \Phi_p^T}{\delta w} v \quad (4.36)$$

L'adaptation des poids est obtenue en utilisant l'algorithme de rétro propagation du gradient donnée par l'équation suivante:

$$\Delta w(t) = \eta \frac{\delta \Phi_p^T}{\delta w} v + \mu \Delta w(t-1) \quad (4.37)$$

Où η est le taux d'adaptation et μ est le facteur d'oubli ou momentum .

Remarque :

On remarque qu'avec cette structure de commande, la position désirée X_d subit deux modifications, la première issue de la boucle de commande en effort permettant de contrôler l'effort appliqué et la deuxième issue du réseau de neurones permettant de compenser la dynamique du robot.

4.6.3 Analyse des performances en présence d'un milieu élastique

La condition nécessaire et suffisante de la stabilité de cette structure de commande se déduit dans le cas idéal (compensation parfaite de la dynamique du robot par le réseau de neurones) directement par analyse de l'équation du système non forcé donnée par l'équation suivante:

$$K_p X + K_v \dot{X} + K_p K_i K_e \int_0^t X d\tau = 0 \quad (4.38)$$

On remarque qu'il s'agit d'analyser un même type d'équation que celle dans le cas de la structure parallèle non basée sur le modèle dynamique donnée précédemment (voir chapitre 3 équation 3.70). Donc, sous les mêmes hypothèses sur l'environnement, on déduit que la condition de la stabilité du système en boucle fermée est vérifiée pour :

$$k_p, k_v, k_i > 0 \quad (4.39)$$

Remarque :

On remarque qu'avec cette structure la stabilité est vérifiée indépendamment de la valeur de la raideur de l'environnement.

4.6.3.1 Cas d'un environnement de position mal connue

Dans le cas où la position de l'environnement est mal connue, l'équation (4.38) s'écrira:

$$K_v \dot{E}_p + K_p E_p + K_p K_i K_e \int_0^t E_p d\tau = -K_p K_i \int_0^t F_d d\tau - K_v \delta \dot{X}_e - K_p \delta X_e \quad (4.40)$$

(δX_e représente l'incertitude sur X_e).

L'application de la transformée de Laplace à l'équation (4.40) nous permet d'écrire:

$$E_p(p) = -[K_v p^2 + K_p p + K_p K_e K_i]^{-1} \times [K_p K_i F_d + (K_v p^2 + K_p p) \delta X_e(p)] \quad (4.41)$$

Avec $\delta X_e(p)$ l'entrée et $E_p(p)$ la sortie

Si on suppose que la perturbation est un échelon ($\delta X_e(p) = \frac{I}{p}$), et que $F_d = 0$, le point d'équilibre du système (4.41) sera:

$$E_p(\infty) = \lim_{p \rightarrow 0} p E_p(p) = 0 \Rightarrow X_\infty \rightarrow X_e \Rightarrow F_e = K_e (X_\infty - X_e) \rightarrow 0$$

Ce résultat montre que dans le cas idéal, le système est compliant et que la loi de commande en effort domine la loi de commande en position.

4.6.3.2 Cas d'un environnement curviligne

Dans le cas d'un environnement curviligne on peut dire que sous les mêmes hypothèses sur l'environnement supposées précédemment (chapitre 3 paragraphe 3.3.1.2), les conditions de la stabilité restent valables pour ce cas.

4.6.4 Résultats de Simulation :

Pour illustrer les performances de cette nouvelle structure de commande en effort externe basée sur cette nouvelle approche neuronale, nous avons repris les mêmes conditions de simulation dans le cas de poursuite par l'organe terminal d'un cercle sur un plan incliné.

La raideur du mur supposée au départ connue, est de 10^4 N/m, ainsi que l'outil porté par l'organe terminal du robot supposé de masse $m = 2$ Kg.

Le réglage des paramètres des contrôleurs et ceux du réseau de neurones sont choisis comme suit:

$$K_p = 1200 \times I, K_v = 60 \times I, K_i = 6.66 \cdot 10^{-4} \times I, \eta = 0.0008; \mu = 0 \text{ et } w(0) = 0; \beta_i = 1/3 \cdot 10^{-4} \text{ (} i=1,2,3 \text{) et } T_e = 1 \text{ms}$$

Les résultats de simulations sont représentés par les figures 4.29 et 4.30 ci-dessous :

On remarque qu'avec ce réglage des paramètres des contrôleurs, l'effort appliqué et la trajectoire désirée sont nettement suivis (les erreurs selon la direction x et z représentent la pénétration de l'effecteur dans le mur).

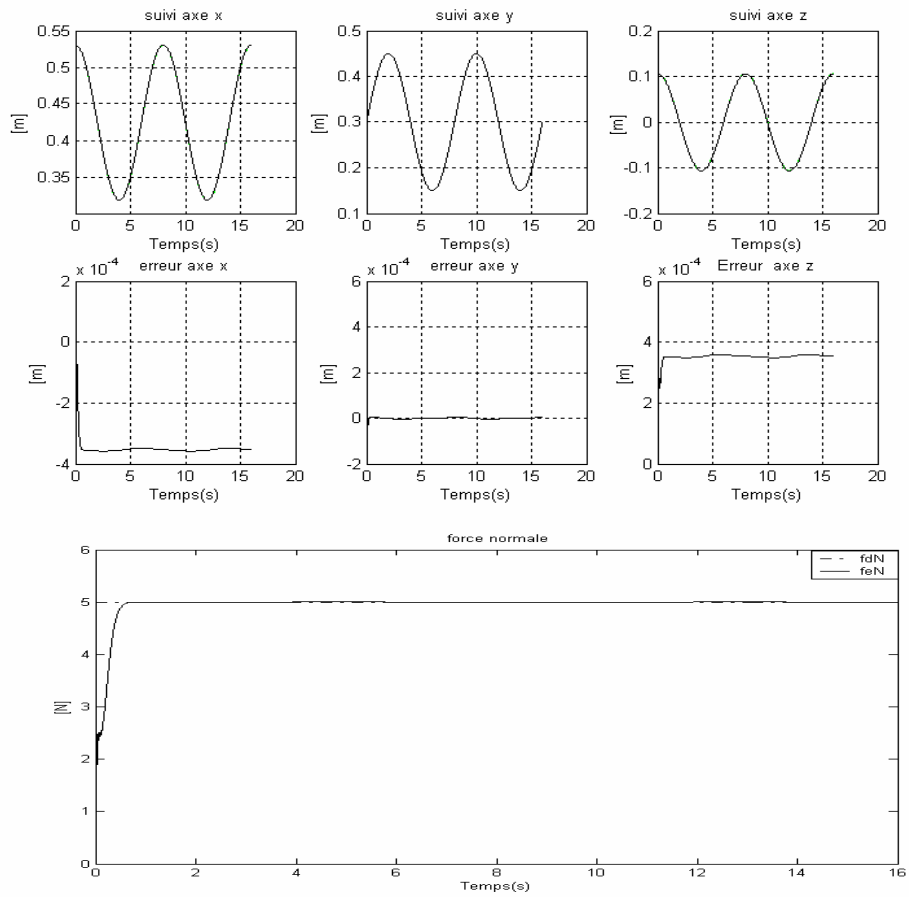


Figure 4.29 : Evolution de la position et de l'effort normal appliqué

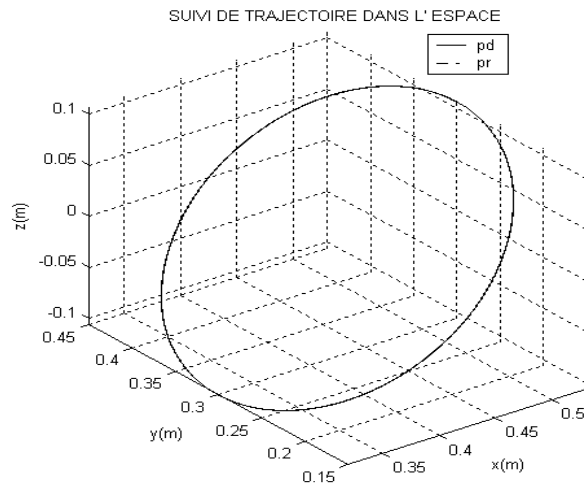


Figure 4.30 : Evolution de la trajectoire dans l'espace.

Pour évaluer la robustesse de notre approche nous avons introduit les incertitudes suivantes sur les modèles du robot ainsi que son environnement :

- Charge portée par le robot $m=10\text{kg}$
- Frottements au niveau des articulations, modélisés par:

$$\tau_f = 0.8 \dot{q} + 0.5 \text{sign}(\dot{q})$$

- Raideur de l'environnement variable et discontinue donnée par :

$$k_e = \begin{cases} 10000 \text{ N/m} & 1 < t \leq 4 \\ 40000 \text{ N/m} & 4 < t \leq 8 \\ 20000 \text{ N/m} & 8 < t \leq 12 \\ 30000 \text{ N/m} & 12 < t \leq 16 \end{cases}$$

Les résultats de simulation sont donnés par les figures 4.31 et 4.32.

On remarque que les incertitudes introduites sur le système n'ont pas influé sur le suivi de la trajectoire du robot. Dans le cas du suivi de l'effort appliqué on remarque que l'introduction des variations brusques de la raideur de l'environnement induit des pics suivis d'oscillations mais d'amplitude rapidement amorties au bout d'un temps inférieur à 0.5s (figure 4.31). La figure 4.32 représente l'évolution de la trajectoire dans l'espace.

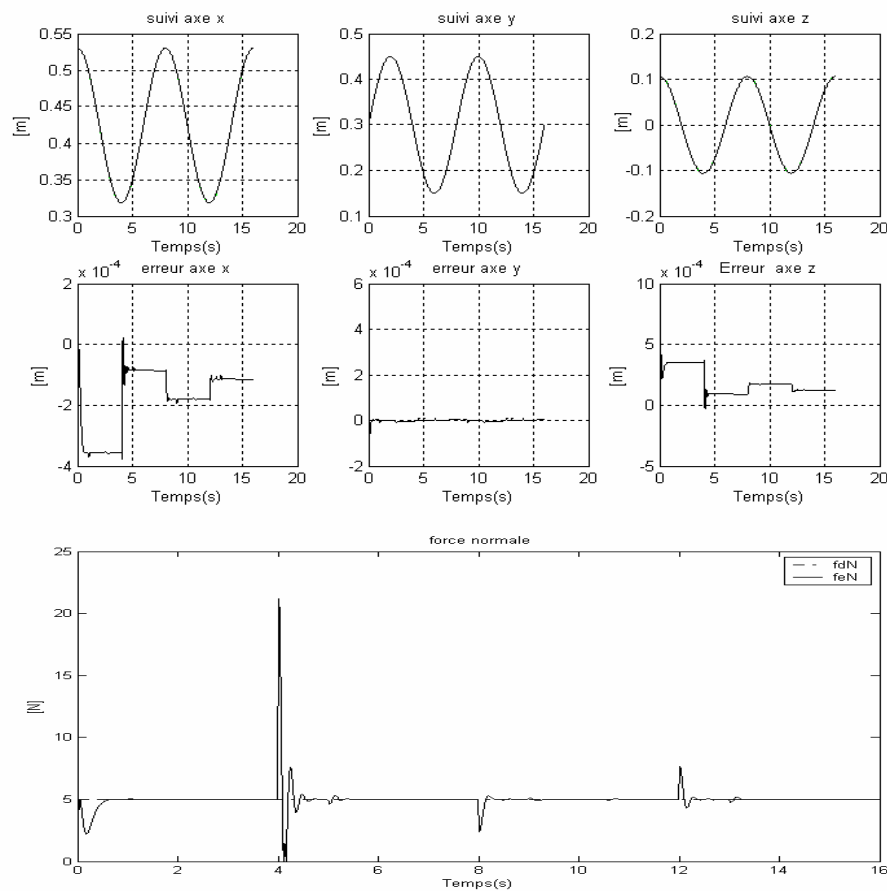


Figure 4.31 Evolution de la position et l'effort normal appliqué

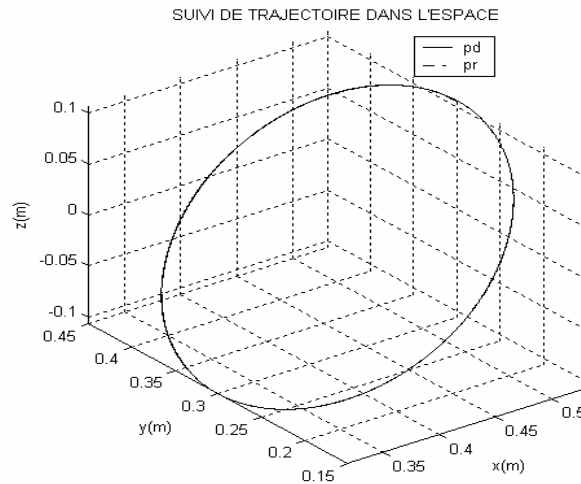


Figure 4.32 : Evolution de la trajectoire dans l'espace

Remarques :

- En comparant ces résultats avec ceux obtenus avec la structure neuronale en effort externe basée sur le modèle dynamique proposé ci-dessus (figure 4.2), on remarque que cette structure donne de meilleures performances pour la réponse en effort.
- Une extension de cette approche de commande au cas d'une boucle de commande en position, de cette structure externe, définie dans l'espace articulaire est proposée en annexe B.

4.7 Structure neuronale de la commande externe avec boucle de commande en effort adaptative

Les performances des réponses en effort de la structure de commande, proposée ci-dessus, peuvent s'améliorer si on trouve le moyen de rendre la boucle de commande en effort adaptative vis-à-vis de la variation de la raideur de l'environnement. Nous proposons dans ce qui suit une extension de l'approche basée sur l'identification en ligne de la raideur de l'environnement, développée dans le chapitre 3 précédent (paragraphe 3.7), appliquée à cette structure de commande. Le nouveau schéma de commande est donné par la figure 4.33 suivante:

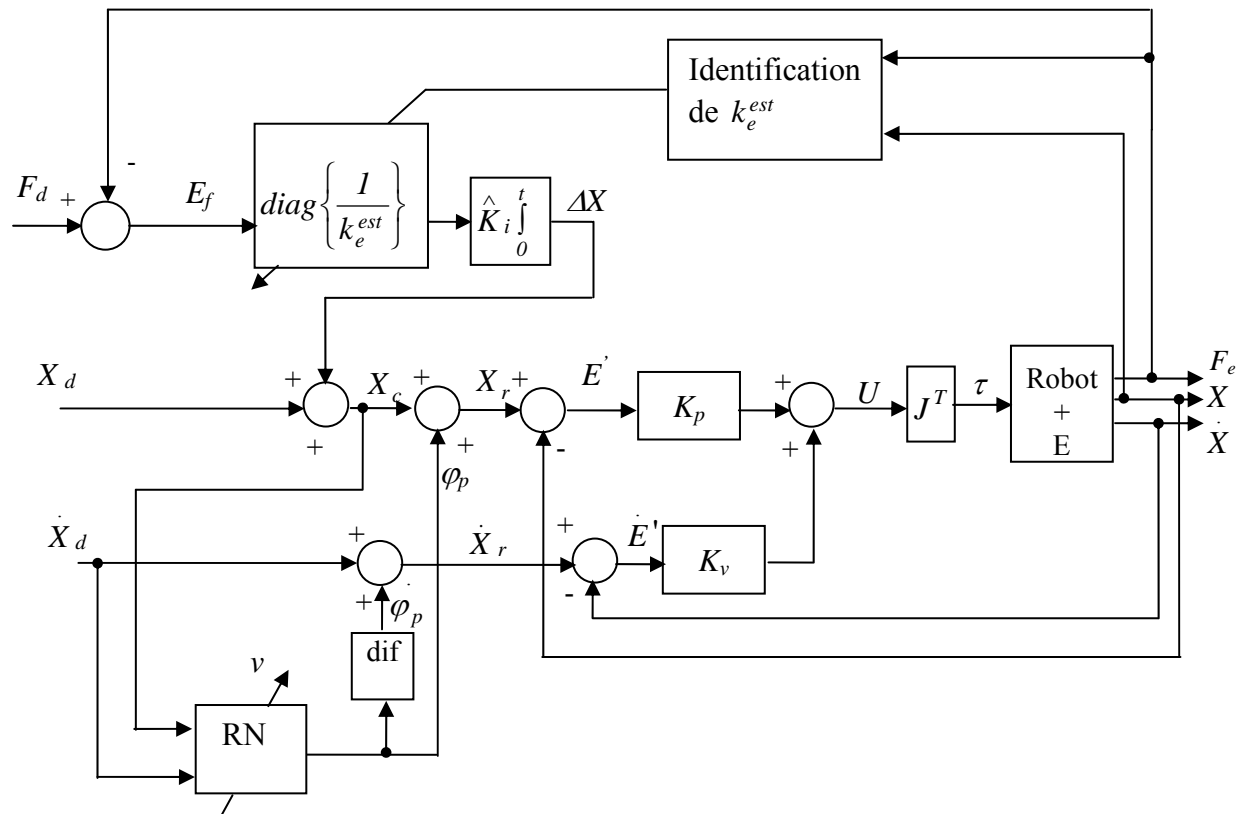


Figure 4.33 : Structure neuronale de la commande externe avec une boucle de commande en effort adaptative.

Avec la structure proposée ci-dessus les gains de réglage de la boucle de commande en effort changent lorsque la raideur estimée de l'environnement change.

Résultats de simulation

Pour montrer l'efficacité de notre approche, nous avons gardé les mêmes conditions de simulation que le cas précédent (sans le bloc adaptatif)

Les paramètres du contrôleur en force sont fixés comme suit :

$$\hat{K}_i = 6.66 \times I \quad \text{et la valeur initiale de } k_e^{est} = 10^4 \text{ N/m}$$

La réponse en effort selon la normale à la surface de contact est représentée par la figure 4.34. On remarque qu'avec ce contrôleur adaptatif les pics observés aux moments où la raideur de l'environnement subit des variations sont rapidement amortis et sans oscillations au bout d'un temps inférieur à 0.5s. Donc on peut déduire que la réponse en effort est nettement améliorée avec cette approche adaptative de la boucle de commande en effort.

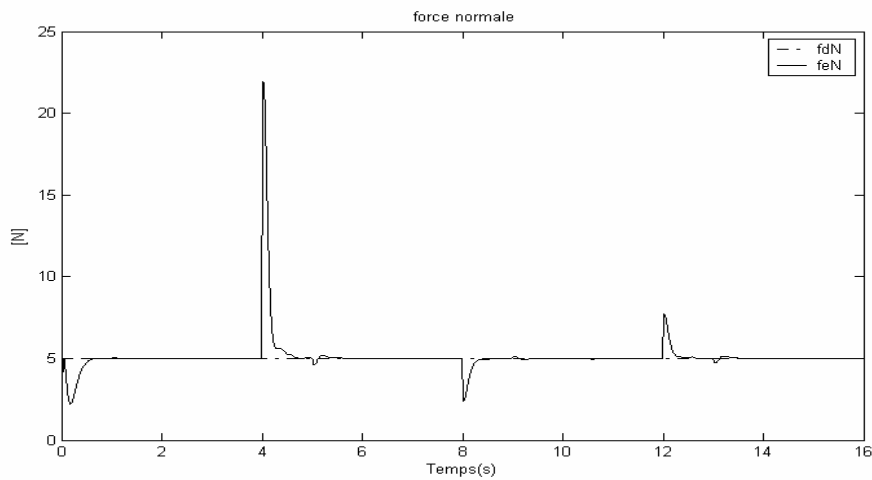


Fig. 4.34 : Suivi de l'effort normal avec le contrôleur adaptatif de la boucle en effort

4.8 Structure neuronale de la commande en effort externe avec boucle de commande en effort flu

L'approche de commande utilisant un contrôleur flou dans la boucle de commande en effort (figure 4.21) peut être facilement étendue à la structure neuronale de la commande en effort externe non basée sur le modèle dynamique. Le schéma de commande proposé est donné par la figure 4.35 suivante:

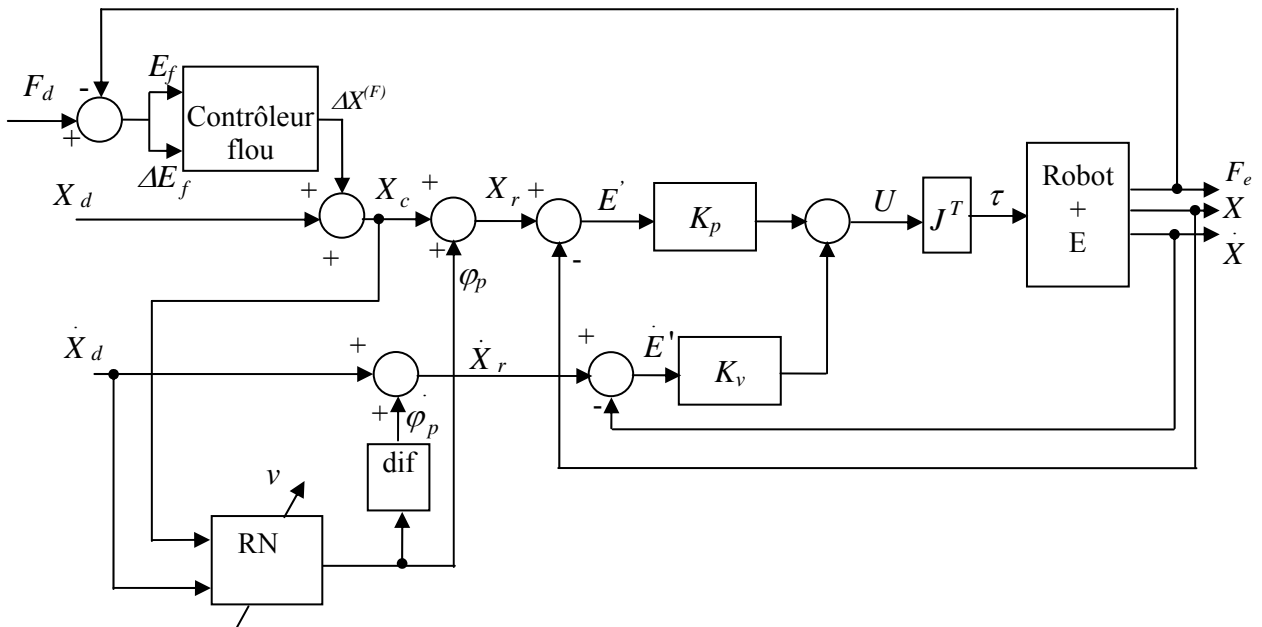


Figure 4.35 : Structure neuronale de la commande en effort externe avec boucle de commande en effort flu

Le contrôleur flou utilisé ici est le même que celui décrit dans le chapitre 3.

Avec les mêmes incertitudes sur la dynamique du robot et son environnement et pour un même réglage des paramètres des contrôleurs, en ajoutant le réglage des gains du contrôleur flou comme suit : $G_{e_f} = 0.09$, $G_{\Delta e_f} = 0.9$ et $G_{\Delta X(F)} = 0.3$, la réponse en effort selon la normale à la surface est donnée par la figure 4.36.

On remarque qu'on obtient une nette amélioration des temps de réponses (inférieur à 0.1s) en utilisant le contrôleur flou. On remarque aussi que les effets des perturbations engendrées par les variations brusques de la raideur sont rapidement maîtrisées en utilisant ce contrôleur flou.

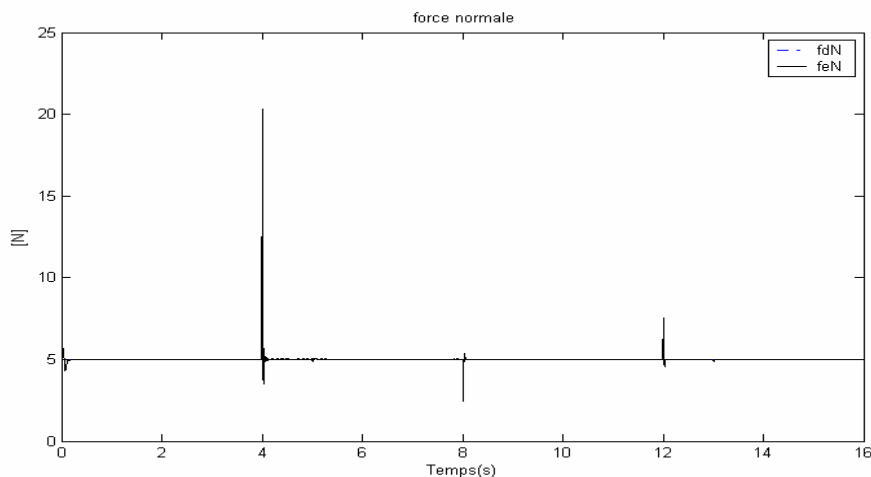


Figure 4.36 : Evolution de l'effort selon la normale en utilisant un contrôleur flou dans la boucle en effort

Remarque

On tient à signaler ici que cette approche de commande floue peut être étendue à la boucle de commande en effort de la structure parallèle non basée sur le modèle dynamique proposée dans le chapitre 3 précédent.

4.9 Conclusion

Nous avons proposé dans ce chapitre de nouvelles approches de commande force/position basées sur des schémas de commande en effort externe classiques et sur des concepts neuronaux et flous. Une analyse théorique de deux schémas proposés pour cette structure en effort externe dont l'un est basé sur le modèle dynamique du manipulateur et l'autre sur de simples contrôleurs PID, nous ont permis de montrer une caractéristique importante que partagent ces deux structures avec celles présentées dans le chapitre 3 précédent : la stratégie de la dominance de la boucle d'effort sur celle en position. Pour améliorer les performances de ces deux structures de commande conventionnelles en suivi de trajectoire et en effort appliqué, nous avons développé de nouveaux schémas issus de ces deux structures en utilisant des approches neuronales et floues déjà proposées dans le cas des structures parallèles présentées dans le chapitre 3. Les résultats obtenus ont été très satisfaisants et similaires à ceux obtenus avec la structure parallèle.

Conclusion générale

Conclusion générale

Nous avons présenté dans ce mémoire de nouvelles approches de commande pour traiter le problème du contrôle en effort des robots manipulateurs qui entrent en interaction avec leur environnement lors de l'exécution de tâches de suivi de trajectoire en milieu contraint.

Ce type de tâches implique des contraintes très sévères et constitue en général un processus dont les paramètres varient dans le temps. Certaines structures de commande en effort proposées dans la littérature reposent sur des techniques de commande classiques linéaires et non linéaires. Notre objectif a été d'essayer de donner de nouvelles orientations à ce type de contrôle, en introduisant des outils modernes de l'Automatique, tels que les réseaux de neurones et la logique floue.

Dans une première étape, pour mettre en évidence les limites des commandes conventionnelles, nous avons présenté les différentes structures de commande en effort proposées dans la littérature. Nous avons montré l'intérêt que possèdent les structures de commande permettant de contrôler à la fois la position et l'effort désirés le long d'une même direction.

Les caractéristiques de dominance de la boucle de commande en force sur celle en position leur permettent de gérer les situations de conflit entre les deux boucles de commande lors de l'exécution des tâches confiées au robot. Il a été montré que l'inconvénient majeur de ces structures réside dans la nécessité de disposer d'un modèle fiable robot/environnement.

Dans une deuxième étape, nous avons décrit les outils utilisés issus de l'Intelligence Artificielle, en particulier les réseaux de neurones ainsi que leur utilisation dans les structures de commande des processus. Cette classe des systèmes intelligents est réputée par sa capacité d'approximation des fonctions non linéaires.

Nous nous sommes intéressés à la théorie de la logique floue qui est utilisée pour la modélisation des systèmes dont les modèles sont difficiles à établir. Un intérêt particulier a été donné dans cette partie à la relation entre la logique floue, les contrôleurs PID classiques et le caractère non linéaire de ce type de contrôleur. Certaines structures de commande floue proposées dans la littérature intéressantes pour notre cas, ont été retenues.

Dans une troisième étape, nous avons abordé le problème de la commande parallèle force/position des robots manipulateurs. Deux structures neuronales de ce type de commande ont été présentées. La première structure, qui exploite le modèle dynamique nominal dans l'élaboration de la commande, utilise une approche de commande neuronale basée sur une technique de modification de l'effort appliqué à l'organe terminal pour compenser les effets des incertitudes sur le modèle dynamique utilisé dans la commande. La deuxième structure, n'utilisant pas le modèle dynamique, utilise quant à elle une technique de modification de la trajectoire de référence pour compenser la dynamique globale du robot manipulateur en contact.

En effet, dans les deux structures proposées, le réseau de neurones est adaptatif en ligne grâce à l'algorithme de rétro propagation entraîné par le vecteur signal d'erreur hybride du système en boucle fermée propre à chaque structure de commande parallèle force/position.

Nous avons montré que le deuxième schéma proposé, présente non seulement l'avantage d'être stable, dans le cas idéal, en présence d'un milieu élastique, pour toute valeur de la raideur de l'environnement mais en plus ses performances sont nettement meilleures pour les réponses en efforts appliqués.

L'analyse théorique des équations en boucle fermée des deux structures de commande proposées, nous a montré que le placement des pôles de ces équations caractéristiques dépend directement d'une connaissance préalable de la raideur de l'environnement qui peut être dans certains cas de valeur incertaine.

Cette analyse nous a permis de montrer que le choix des paramètres de réglage fixes de la boucle de commande en force peut devenir inadéquat pour certaines variations de cette raideur de l'environnement.

Pour remédier à ce problème, deux solutions ont été proposées. La première solution dédiée à la première structure neuronale consiste à remplacer le contrôleur conventionnel de la boucle de commande en effort par un contrôleur flou muni d'un bloc d'autoréglage de ses paramètres de sortie. La seconde solution, dédiée à la deuxième structure de commande neuronale parallèle force/position, quant à elle, permet de transformer la boucle de commande en effort classique en une boucle adaptative grâce à une technique d'estimation en ligne de la raideur de l'environnement.

Les différents schémas de commande proposés dans ce chapitre ont été validés en simulation en considérant plusieurs types d'environnement et de trajectoires exécutées par un robot de type PUMA560.

Dans la dernière partie de notre travail, nous avons abordé le problème de la commande en effort externe. Nous avons proposé une nouvelle structure dynamique de ce type de commande. Une analyse détaillée de l'équation du système en boucle fermée de cette structure de commande, nous a permis de montrer une équivalence de cette équation à celle que nous avons établi avec la structure dynamique parallèle proposée par CHIAVERINI [CHI 93].

Pour améliorer ses performances, une nouvelle structure intelligente de la commande dynamique en effort externe a été proposée. La méthode constitue une extension au cas de la structure de commande dynamique en effort externe d'une méthode déjà proposée pour le cas de la structure dynamique parallèle conventionnelle, présenté dans le chapitre 3.

Ces techniques ont été validées en simulation sur les modèles du robot PUMA 560 en considérant les mêmes tâches et dans les mêmes conditions de simulation que pour la structure parallèle. Les résultats obtenus ont montré, de manière significative, l'efficacité de la commande et sa robustesse par rapport aux variations paramétriques du robot et de l'environnement. Comparativement aux résultats obtenus avec la structure intelligente parallèle, les résultats ont été jugés similaires.

Une nouvelle version intelligente (au sens de l'I.A) de la structure de commande en effort externe, basée sur de simples contrôleurs PID conventionnels pour ses deux boucles de commande a été proposée et des résultats de simulations ont été validés et comparés.

L'ensemble des structures de commande proposées dans ce travail ont été testées en simulation en supposant que le robot est équipé de capteurs de position, de vitesse et de capteurs d'effort. Ces derniers sont supposés munis de filtres permettant d'éliminer les bruits des signaux. Le contrôle est supposé être assuré par des calculateurs performants permettant une exécution en temps réel des commandes.

En conclusion, nous avons essayé de proposer à travers ce travail des solutions robustes issues des concepts de l'intelligence artificielle permettant d'améliorer de manière très significative les performances de certaines structures de commande force/position classiques proposées dans la littérature.

Ces méthodes illustrent tout l'intérêt que l'on peut tirer, de l'hybridation de méthodes classiques dont les performances sont bien connues, mais qui s'avèrent limitées lorsqu'on possède une connaissance partielle ou très imparfaite des modèles du robot et/ou de l'environnement, avec des méthodes issues de l'Intelligence Artificielle, réseaux de neurones et logique floue.

Les perspectives à ce travail sont de valider les méthodes proposées sur un robot PUMA 560 réel et de comparer les résultats obtenus avec ceux de nos simulations et les résultats d'autres approches de commande.

References Bibliographiques

- [ARM 86] Armstrong B., Khatib O., “*The explicit dynamic model and inertial parameters of the PUMA 560 arm*”, IEEE Int. Conf. on Robotics and Automation, San Francisco –U.S.A, pp. 510-518, 1986.
- [BAP 99] Baptista L. F., Sada Costa M. G., “*A force control approach of robotic manipulators with neural network compensation*”, 14th world congress of IFAC, Beijing, P.R. China 1999, pp.173-178.
- [BOI 88] Boissonat JD., Faverjon B., “*Technique de la robotique*”, Tome I Editions Hermès, Paris, 1988.
- [BOU 98] Bouhali O., Boukhetala D., “*Commande neuronale adaptative décentralisée du robot PUMA560*”, 1^{er} colloque National sur la productique pp. 200-204, Tizi-Ouzou, Algérie, Mai 1998.
- [BOU 02] Boukhetala D., “*Contribution à la commande décentralisée des systèmes complexes interconnectés : Applications en Robotique*”, thèse de doctorat d'état - Ecole polytechnique d'Alger, 2002.
- [CHI 04] Chian S. C., Kuang Y. L. “*Robust adaptive Motion/Force Tracking Design for Uncertain Constrained Robot Manipulators*”. Automatica 40, pp. 2111-2119, 2004.
- [CHI 92] Chiaverini S., Siciliano B., “*A stable force/position controller for robot manipulators*”. in proc. IEEE Int. conf. on Decision and control, (1992), pp. 1869-1874.
- [CHI 93] Chiaverini S. and Sciavisco L., “*The parallel approach to force/position control of robotic manipulators*”, IEEE Trans. on Robotics and Automation, Vol. 9, n°. 4, pp. 361-373, August 1993.
- [CHI 94] Chiaverini S., Siciliano B., “*Force/position regulation of compliant Robot manipulators*”, IEEE Trans. on Automatic Control, Vol. 39, n° 3, pp. 647-652, March 1994.
- [CAR 95] Carelli R., Eduardo F., “*A neural Network Based Feedforward Adaptive Controller for Robots*”, IEEE Trans. on systems, man, and cybernetics-part B: Cybernetics, vol. 25, n° 9. September 1995.
- [DEG 93] Degoulange E., “*Commande en effort d'un robot manipulateur à deux bras : Application au contrôle d'une chaîne cinématique fermée*”, Thèse de Doctorat. Université de Montpellier II, France, 93.
- [DER 03] Dérragui N., “*Commande coordonnée de Robots Coopérant*”, thèse de Magister, USTHB, Algérie, 2003.
- [DUM 87] Dumas R., Samson C., “*Robust nonlinear control of robotic manipulators: implementation aspect and simulation*”, Internal report, INRIA Rennes, France, 1987.
- [FER 97] Ferguene F., Toumi R., “*Sensibilité d'un Modèle Cinématique de Robot Manipulateur à cinq axes*” 17^{èmes} Journées Tunisiennes d'Electrotechnique et d'Automatique 5 et 6 Novembre 1997 Nabeul-Tunisie.
- [FER 03] Ferguene F., Toumi R. “*Commande hybride force/position d'un bras manipulateur par une approche prédictive*”, CIP 2003-Conférence internationale sur la productique, Alger, Algérie 2003.
- [FER 05] Ferguene F., Toumi R., “*A Neural Approach to Force/Position Parallel Control of Robotic Manipulators Application to the Follow-up of Trajectory in Unknown Stiffness Environment*”, in Proc. International computer systems and information technology conference Vol. N°1, pp. 247-251, July 19-21 2005, Algiers, Algeria.
- [FER 09] Ferguene F., Toumi R., “*Dynamic External Force Feedback Loop Control with Neural Compensator of Robot Manipulator. Application to the Trajectory Following in an Unknown Environment*”. Int. J. Appl. Math. Comput. Sci., vol. 19, N° 1, 2009. A paraître. Acceptation définitive.
- [FER 08] Ferguene F., Achour N. and Toumi R., “*Neural Network Parallel Force/Position Control of Robot Manipulators under environment and robot dynamics uncertainties*”, Article soumis au Journal Archives of control sciences.
- [FRA 94] Fraisse P., “*Contribution à la commande robuste position/force des robots manipulateurs à architecture complexe : Application à un robot à deux bras*”, Doctorat, Université Montpellier II, France, 1994.
- [FUL 95] Fuller R., “*Neural Fuzzy systems*”, Donner visiting Professor, Abo Akademi University, 1995.
- [GOR 84] Gorla M., Renaud M., “*Modèles des robots manipulateurs: Application à leur commande*”, Editions Cepadues, 1984.
- [HAY 99] Haykin S., “*Neural Networks: A comprehensive Fondation*”, Prentice Hall 1999, 2^{ème} édition.
- [HOG 85] Hogan N., “*Impedance control: An approach to manipulator, parts I, II and III*”, ASME J. Dynam. Syst., Meas., Contr., vol. 107, pp. 1-24, Mar. 1985.

- [HUA 07] Huang C., Liu L., “*Robust scheme of global parallel force/position regulators for robot manipulators under environment uncertainty*” Jour. of cont. theory and application, vol. 5, N°3, pp. 271-277, 2 007.
- [HUN 91] Hunt, K. J., Sbarbaro-Hofer D., “*Neural networks for nonlinear internal model control*”. IEE proceedings. Part D. Control theory and applications vol. 138, n°5, pp. 431-438 (17 ref.) 1991.
- [ISH 92] Ishiguro A., Furuhashi T., “*A neural Network Compensator for Uncertainties of Robotics Manipulators*”. IEEE, Trans. on Industrial Electronics. Vol. 39. n° 6, pp. 565-569, December 1992.
- [JAN 95] Jang J.S.R., Sun C.T., “*Neuro-Fuzzy Modeling and Control*”, Proc. IEEE, Vol. 83, n°3. pp. 378-404, Mars 95.
- [JUN 95a] Jung S., Hsia T.C., “*On neural network application to robust impedance control of robot manipulators*”, in Proceedings of IEEE Int. Conf. on Robotics and Automation. Nagoya, Japan, May 1995, Vol. n°1, pp. 869-874.
- [JUN 95b] Jung S., Hsia T.C., “*On Reference Trajectory Modification Approach for Cartesian Neural Network Control of Robot Manipulator*”, Proc. IEEE Int. Conf. on Robotics and Automation, pp. 575-580, 1995.
- [JUN 95c] Jung S., Hsia T.C., “*New Neural Network Control Technique for Non Model Based Robot Manipulator control*”. In Proc. of the IEEE Int. Conf. on Robotics and Automation, pp. 2928-2933, January 1995.
- [JUN 95d] Jung S., Hsia T.C., “*A New Neural Network Control Technique for Robot Manipulator*”, IEEE, in Proc. of the American control Conference, Seattle, Washington, pp. 878-882, June 1995.
- [JUN 96] Jung S., “*Neural Network Controllers for Robot Manipulators*”, doctorate thesis, university of California Davis, USA 1996.
- [JUN 98] Jung S., Hsia T.C., “*Neural network impedance force control of robot manipulator*”, IEEE Trans. on Industrial Electronics. Vol. 45. N° 3, pp. 451-461, June 1998.
- [JUN 00] Jung S., Hsia T.C., “*Robust neural force control scheme under uncertainties in robot dynamics and unknown environment*”, IEEE Trans. on Industrial Electronics. Vol. 47. n° 2, pp. 403-412, April 2000.
- [JUN 01] Jung S., HSIA T.C “*Experimental studies of neural network impedance force control for robot manipulators*”, in Proceedings of the IEEE Int. Conf. on Robotics and Automation . Seoul, Korea. May 21-26, 2001, pp. 3453-3458.
- [JUN 04] Jung S., Hsia T.C., “*Force Tracking Impedance Control of Robot Manipulators Under Unknown environment*”, IEEE Trans. on control systems technology, vol. 12, n° 3, pp. 474-482, May 2004.
- [KHA 86] Khatib O, Burdick J., “*Motion and force control of robot manipulators*”, IEEE international conference on robotics and automation, San Francisco, USA, 7-10 April 1986, pp. 1381-1386.
- [KHA 87] Khatib O., “*A Unified approach for motion and force control of robot manipulators : the operational space formulation*”, IEEE Journal of Robotics and Automation, Vol. RA-3, n° 1, pp. 43-53, February 1987.
- [KIG 95] Kigushi K., Fukuda T., “*Robot manipulator contact force control application of fuzzy neural network*”, IEEE Int. Conf. on Robotics and Automation, Vol. n°1, pp 875-880.
- [KIG 97] Kigushi K., Fukuda T., “*Intelligent position/force controller for industrial robot manipulators-application of fuzzy neural networks*”, IEEE Trans. on Indus. Electronics. Vol. 44. n° 6, pp. 753-761, December 97.
- [LIA 04] Liamini R., “*Commande hybride force-position d’un bras manipulateur par une approche prédictive : Application à la coopération de deux bras*”, Mémoire de PFE, USTHB, Algérie 2004.
- [LIN 00] Lin C., Su H., “*Intelligent Control theory in guidance and Control System Design: an overview*”, proc. Natl. Sci, Counc. Roc(A) Vol. 24, n°. 1, pp. 15-30, 2000.
- [MAS 81] Mason M. T., “*Compliance and force control for computer controller manipulator*”, IEEE Tran. on systems, Man, and cybernetics, vol. SMC-11(6), 1981, pp. 418-432.
- [MAR 97] Marques S. J., Baptista L., “*Force control of robot manipulators with neural networks compensation: A comparative study*”, in Proc. IEEE ISIE, pp. 872-876, 1997.
- [MIL 89] Mills J., Goldenberg A. “*Force and position control of manipulators during constrained motion tasks*”, IEEE Trans. on Robotics and Automation, Vol. 5, n° 1, pp. 30-46, February 1989.

- [MIY 88] Miyamoto H., Kawato T., Setayama T. and Susuki R., "Feedback error learning neural network for trajectory control of a robotic manipulator", IEEE Trans. On Neural Networks, vol. 1, pp. 251-265, 1988.
- [NAK 03] Nakawono K., Katagiri M., "Force and position control of robot manipulator using neurocontroller with GA based training", Proc. of IEEE Inter. Sym. on Computational Intelligence in Robotics and Automation, July 16-20, 2003, Kobe, Japan, pp. 1354-1356.
- [NAK 97] Nakoula Y. « *Apprentissage des modèles linguistiques flous, par jeu de règles pondérées* », Thèse de Doctorat, Université de Savoie, France, Juillet 1997.
- [NAR 90] Narandra K. S., Parthasarathy K., "Identification and control of dynamic systems using neural networks", IEEE Transaction on neural networks, Vol. 1, n°1, pp. 4-27, 1990.
- [OZA 91] Ozaki T., Suzuki T., "Trajectory Control of Robotic Manipulators Using Neural Networks". IEEE Trans. on Industrial Electronics. Vol. 38. N° 3, pp. 195-201, June 1991.
- [OKU 90] Okuma S., Ishiguro A., "A neural Network Compensator for Uncertainties of Robotic Manipulators", IEEE Proc. of the 29th conf. on decision and control, Honolulu, Hawaii, December 1990, pp. 3303-3307, 1990.
- [PAS 98] Passino K., Yurkovich M., "Fuzzy Control", Addison Wesley 1998.
- [PER 91] Perdereau V., "Contribution à la commande hybride force/position : Application à la coopération de deux robots", Thèse de Doctorat Université Pierre et Marie Curie ; France, 1991.
- [PRE 97] Prelle C., "Contribution au contrôle de la compliance d'un bras de robot à actionnement électropneumatique", Thèse INSA ; Lyon, France, 1997.
- [PSA 88] Psaltis, D., Sideris A., Yamura A., "A multilayered neural network controllers", IEEE Control System Magazine, pp. 17-21, 1988.
- [RAI 81] Raibert M. H., Craig J. J., "Hybrid position/force control of manipulators", Trans. of ASME, J. Dynamics Systems Measurement and Control, vol. 103, pp. 126-133, June 1981.
- [SAA 97] Saadia N., "Contribution à la commande hybride force/position des robots compliants selon une approche neuronale", Thèse de Doctorat -Université Paris XII. France. 1997
- [SAL 80] Salisbury J.K., "Active Stiffness control of a manipulator in Cartesian coordinates" Proc. 19th IEEE conf. On Decision and Control, Albuquerque, Déc. 1980, pp. 95-100.
- [SAM 87] Samson C., "Commande non linéaire robuste des robots manipulateurs", Rapport interne, Ins. National de recherche en informatique et en automatique (Laboratoire de Rennes).
- [SCH 88] Schutter J., Van Berussel H. "Compliant robot motion II. A control approach based on external control loop", Int. Journal of Robotics Research, Vol. n°4, August 88, pp. 18-33.
- [SIC 94] Siciliano B., Villani L., "A Passivity-Based Force/Position Control Scheme for Robot Manipulators", Proc. IEEE International Conference on Robotics and Automation, vol.4, pp. 3265-3270, May 1994.
- [SHI 96] Shibata M., Murakami T., "A unified Approach to Position and Force Control by Fuzzy Logic", IEEE Trans. on Industrial Electronics. Vol. 43. n° 1, pp. 81-87, February 1996.
- [SIN 95] Singh S. K. and Popa D. O., "An Analysis of some Fundamental Problems in Adaptive Control of Force And Impedance Behavior: Theory and Experiments", IEEE Trans. Robotics and Automation. Vol.11. N° 6, pp. 912-921, 1995.
- [SON 04] Song D-H., Eom Y., "Comparison Studies of Two Neural Network Compensation Techniques for Standard PD- Like Fuzzy Controlled Robotic Manipulators", the 30th annual conference of the IEEE industrial Electronics Society, November 2-6, 2004, Busan, Korea, pp. 3178-3182.
- [SON 95] Song K. T and Li H. P., "Design and Experiment of Fuzzy force Controller for an Industrial Robot", in Proc. Natl. Sci. Counc. Roc(A), Vol. 19, N° 1, pp. 26-36.
- [SUR 99] Surdilovic D., "Robust Robot Compliant Motion Control Using Intelligent Adaptive Impedance Approach", IEEE Int. Conf. on Robotics and Automation, Detroit, U.S.A pp. 2128-2133, Mai 1999.

- [TSU 96] Tsuji T., Ito K. “*Neural Network Learning of Robot Arm Impedance in Operational Space*”. IEEE Trans. on systems, man, and cybernetics-part B: Cybernetics, vol. 26, April. 1996.
- [TAN 04] Tanaka Y., Tsuji T., “ On-line learning of robot arm impedance using neural networks”, Proc. IEEE int. conf. on robotics and biomimetics, August 22-26, 2004, Shenyang, China.
- [WHI 97] Whitcomb L. L. “*Adaptive Model-Based Hybrid Control of Geometrically Constrained Robot Arms*”. IEEE Trans. on Robotics and Automation. Vol. 13. N° 1, pp. 105–116.
- [WHI 85] Whitney D. E., “*Historical perspective and state of the art in robot force control*”, Proc. IEEE conf. on Robotics and automation, St Louis, Mars 1985, pp. 262-268.
- [XU 00] Xu J. X., Hang C.C. and Chen L., “*Parallel structure and tuning of a fuzzy PID controller*”, Automatica 36 (2000) pp. 673- 684.
- [YAB 89] Yabuta T., Tsujimura T., “*On the characteristics of the Robot Manipulator Controller Using Neural Networks*”, Int. workshop on industrial application of machine intelligence and vision, Tokyo, April 10-12, 1989, pp. 76-80.
- [YAB 92a] Yabuta T., Tsujimura T., “*Neural Network Controller characteristics with Regard to Adaptive Control*”, IEEE Trans. on systems, man, and cybernetics, vol. 22, n° 1, pp. 170-176, January/February 1992.
- [YAB 92b] Yabuta T., “ Nonlinear Basic Stability Concept of the Hybrid Position/Force Control Scheme for Robot Manipulators” IEEE Trans. on Robotics and Automation, Vol. 8, n°. 5, pp. 663-670, October 1992.
- [YAO 92] Yao B. and Chan S. P., “*Robust Motion and Force Control of Robot Manipulators in the Presence of Environmental Constraint Uncertainties*”. in Proc. IEEE of the 31st conf. on decision and Control , pp. 1875-1880, Tucson, Arisona 1992.
- [YED 98] Yeddou Y.M., “*Etude de synthèse sur les réseaux de neurones et leurs application*”, thèse de Magister école nationale polytechnique, Algérie, 1998.
- [YIN 00] Ying H., “*Fuzzy Control and Modeling: Analytical Foundation and Application*”, IEEE Press Series on Biomedical Engineering, 2000.
- [YOS 00a] Yoshikawa T., Sudou A., “*Dynamic Hybrid Position/Force Control of Robot Manipulators- On-Line Estimation of Unknown Constraint*”, IEEE Trans. on Robotics and Automation, Vol. 9, n° 2, pp 220-226, April 1993.
- [YOS 00b] Yoshikawa T., “*Force control of robot manipulators*”, Proceedings of the IEEE Int. Conf on Robotics and Automation, San Francisco, USA. April 2000, pp.220-226.
- [ZHA 89] Zhang H. “*Kinematic stability of robot manipulators under force control,*” Proc. IEEE int. Conf. robotics and Automation, pp. 80-85, 1989.
- [ZHA 03] Zhao Y. Collins E. G., “*Fuzzy PI Control Design for an Industrial Weigh Belt Feeder*”, IEEE Trans. On fuzzy systems, Vol. 11, N°. 3, pp. 311-319, June 2003.

Annexe A

Modélisation du robot PUMA560

A/ Les modèles du robot PUMA560

La figure A.1 suivante représente la structure du robot PUMA560.

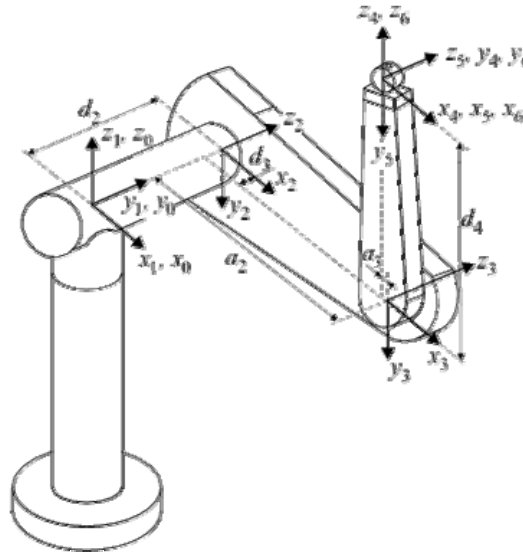


Figure A.1 : Structure du robot PUMA 560

Les modèles du robot sont présentés en supposant que les trois dernières articulations sont à l'arrêt.

Pour faciliter les notations, les abréviations suivantes sont utilisées pour les fonctions géométriques :

C_i, S_i, C_{ij}, S_{ij} et C_{ijk} désignent respectivement $\cos(q_i)$, $\sin(q_i)$, $\cos(q_i + q_j)$, $\sin(q_i + q_j)$ et $\cos(q_i + q_j + q_k)$

A.1 Modèle géométrique direct et inverse

Le modèle géométrique direct (MGD) est obtenu en utilisant le formalisme de Denavith et Hartenberg [GOR 84]. La position de l'objet attaché à l'organe terminal du bras manipulateur en fonction de ses coordonnées articulaires, par rapport à un repère de référence, est donnée par les expressions suivantes [LIA 04] :

$$\begin{aligned} p_x &= a_2 C_1 C_2 - (d_2 + d_3) S_1 + a_3 C_1 C_{23} + d_4 C_1 S_{23} ; \\ p_y &= a_2 S_1 C_2 + (d_2 + d_3) C_1 + a_3 S_1 C_{23} + d_4 S_1 S_{23} ; \\ p_z &= -a_2 S_2 - a_3 S_{23} + d_4 C_{23} ; \end{aligned} \tag{A.1}$$

Avec,

$$a_2 = 0,4318 \text{ m} ; d_2 = 0,2435 \text{ m} ; a_3 = -0,0203 \text{ m} ; d_3 = -0,0934 \text{ m} \text{ et } d_4 = 0,4331 \text{ m}.$$

Le modèle géométrique inverse que l'on note MGI calculé analytiquement est donné par les expressions suivantes [LIA 04]:

$$\begin{aligned} q_1 &= \text{Atan2}(p_y, p_x) - \text{Atan2}\left(d_2+d_3, \pm \sqrt{p_x^2 + p_y^2 - (d_2 + d_3)^2}\right); \\ q_3 &= -\text{Atan2}(a_3, d_4) + \text{Atan2}\left(k, \pm \sqrt{a_3^2 + d_4^2 - k^2}\right); \\ q_2 &= \text{Atan2}(k_1, k_2) - q_3; \end{aligned} \quad (\text{A.2})$$

Avec,

$$k = \frac{p_x^2 + p_y^2 + p_z^2 - a_2^2 - a_3^2 - d_4^2 - (d_2 + d_3)^2}{2a_2},$$

$$k_1 = \frac{-(a_3 + a_2) \times p_z + (C_1 p_x + S_1 p_y) \times (a_2 S_3 + d_4)}{p_z^2 + (C_1 \times p_x + S_1 \times p_y)^2},$$

et

$$k_2 = \frac{(d_4 + a_2 S_3) p_z + (C_1 p_x + S_1 p_y) \times (a_2 C_3 + a_3)}{p_z^2 + (C_1 p_x + S_1 p_y)^2}.$$

A.2 Modèle cinématique du premier et du second ordre direct et inverse:

a) Modèle cinématique direct du premier ordre

Le modèle cinématique direct du premier ordre (MCD) a pour but d'exprimer les vitesses des coordonnées opérationnelles en fonction des vitesses articulaires, donné par la relation suivante :

$$\dot{X} = J(q) \dot{q} \quad (\text{A.3})$$

$J(q)$: représente la matrice Jacobienne du robot et s'obtient par dérivation de son modèle géométrique direct à partir de la relation suivante :

$$J_{ij} = \frac{\partial f_i(q)}{\partial q_j} \quad i=1,2 \text{ et } 3 ; \text{ et } j=1,2 \text{ et } 3.$$

- $f_i(q)$ sont les expressions des coordonnées opérationnelles en fonction des coordonnées articulaires représentant le modèle géométrique direct.
- J_{ij} représente l'élément (i,j) de la matrice Jacobienne J donnée par:

$$J = \begin{bmatrix} J_{11} & J_{12} & J_{13} \\ J_{21} & J_{22} & J_{23} \\ J_{31} & J_{32} & J_{33} \end{bmatrix}$$

Avec,

$$\begin{aligned} J_{11} &= -S_1 (a_2 C_2 + a_3 C_{23}) - (d_2 + d_3) C_1 - d_4 S_1 S_{23}; \\ J_{12} &= -C_1 (a_2 S_2 + a_3 S_{23}) + d_4 C_1 C_{23}; \\ J_{13} &= -a_3 C_1 S_{23} + d_4 C_1 C_{23}; \\ J_{21} &= C_1 (a_2 C_2 + a_3 C_{23}) - d_2 + d_3 S_1 + d_4 C_1 S_{23}; \\ J_{22} &= -S_1 (a_2 S_2 + a_3 S_{23}) + d_4 S_1 C_{23}; \\ J_{23} &= -a_3 S_1 S_{23} + d_4 S_1 C_{23}; \\ J_{31} &= 0; \\ J_{32} &= -(a_2 C_2 + a_3 C_{23}) - d_4 S_{23}; \\ J_{33} &= -a_3 C_{23} - d_4 S_{23}; \end{aligned}$$

b) Modèle cinématique inverse du premier ordre

Le modèle cinématique inverse du premier ordre (MCI) a pour but de calculer les vitesses articulaires \dot{q} qui assure une vitesse opérationnelle \dot{X} imposée grâce à la relation inverse suivante:

$$\dot{q} = J^{-1} \dot{X} \quad (\text{A.4})$$

c) Modèle cinématique du second ordre direct

Le modèle cinématique direct du second ordre (MCD2) permet de calculer les accélérations des coordonnées opérationnelles en fonction des positions, vitesses et accélérations articulaires.

Il s'obtient par dérivation du modèle cinématique par rapport au temps.

$$\ddot{X} = J \ddot{q} + \dot{J} \dot{q} \quad (\text{A.5})$$

Avec,

$$J(q, \dot{q}) = \frac{\delta J(q)}{dt}$$

d) Modèle cinématique inverse du second ordre

Le modèle cinématique inverse du second ordre (MCI2) est obtenu directement à partir de la relation (A.5) comme suit:

$$\ddot{q} = J^{-1} (\ddot{X} - \dot{J} \dot{q}); \quad (\text{A.6})$$

A.3 Le modèle dynamique

Le modèle dynamique des trois premières articulations du robot PUMA 560 est donné par:

$$A(q)\ddot{q} + B(q, \dot{q})\dot{q} + G(q) = \tau - m J^T(q) \left[J(q)\ddot{q} + \dot{J}(q, \dot{q})\dot{q} + g \right] \quad (\text{A.7})$$

Avec,

$q = [q_1, q_2, q_3]$ est le vecteur des coordonnées généralisées, m la charge (outil), $J(q)$ la matrice Jacobienne et $g^T = [0 \ 0 \ +9.81]$

Les expressions des matrices A , B et G sont : [ARM 86]

$$A = \begin{bmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{bmatrix}$$

Avec,

$$a_{11} = 2,57 + 1,38 C_2^2 + 0,30 S_2 S_3 + 7,44 \times 10^{-1} C_2 S_{23};$$

$$a_{12} = a_{21} = 6,90 \times 10^{-1} S_2 - 1,34 \times 10^{-1} C_{23} + 2,38 \times 10^{-2} C_2;$$

$$a_{13} = a_{31} = -1,34 \times 10^{-1} C_{23} - 3,97 \times 10^{-3} S_{23};$$

$$a_{22} = 6,79 + 7,44 \times 10^{-1} S_3;$$

$$a_{23} = a_{32} = 0,333 + 3,72 \times 10^{-1} S_3 - 1,10 \times 10^{-2} C_3;$$

$$a_{33} = 1,16;$$

$$B = \begin{bmatrix} b_{11} & b_{12} & b_{13} \\ b_{21} & b_{22} & b_{23} \\ b_{31} & b_{32} & b_{33} \end{bmatrix}$$

Avec,

$$b_{11} = (-2,76 S_2 C_2 + 7,44 \times 10^{-1} C_{223} + 0,60 S_2 C_3 - 2,13 \times 10^{-2} (1 - 2 S_2 S_3)) \dot{q}_2;$$

$$b_{12} = (6,90 \times 10^{-1} C_2 + 1,34 \times 10^{-1} S_{23} - 2,38 \times 10^{-2} S_2) \dot{q}_2 + (2,67 \times 10^{-1} S_{23} - 7,58 \times 10^{-3} C_{23}) \dot{q}_3;$$

$$b_{13} = (7,44 \times 10^{-1} C_2 C_{23} + 0,60 S_2 C_3 + 2,20 \times 10^{-2} C_2 S_{23} - 2,13 \times 10^{-2} (1 - 2 S_2 S_3)) \dot{q}_1 + 0,5 \times (2,67 \times 10^{-1} S_{23} - 7,58 \times 10^{-3} C_{23}) \dot{q}_3;$$

$$b_{21} = -0,5(-2,76 S_2 C_2 + 7,44 \times 10^{-1} C_{223} + 0,60 S_2 C_3 - 2,13 \times 10^{-2} (1 - 2 S_2 S_3)) \dot{q}_1;$$

$$b_{22}=(2,20 \times 10^{-2} \times S_3+7,44 \times 10^{-1} \times C_3) \dot{q}_3 ;$$

$$b_{23}=0,5 (2,20 \times 10^{-2} S_3+7,44 \times 10^{-1} C_3) \dot{q}_3 ;$$

$$b_{31}= -0,5 (7,44 \times 10^{-1} C_2 C_{23}+0,60 S_2 \times C_3+2,20 \times 10^{-2} \times C_2 S_{23} -2,13 \times 10^{-2} (1-2 S_2 S_3)) \dot{q}_1 ;$$

$$b_{32}= -0,5 (2,20 \times 10^{-2} S_3 + 7,44 \times 10^{-1} C_3) \dot{q}_2 ;$$

$$b_{33}=0 ;$$

$$G= \begin{bmatrix} g_1 \\ g_2 \\ g_3 \end{bmatrix}$$

Avec,

$$g_1=0 ;$$

$$g_2= -37,2 C_2-8,4 S_{23}+1,02 S_2 ;$$

$$g_3= -8,4 S_{23}+0,25 C_{23} ;$$

Annexe B

**Structures de commande externes
avec boucle neuronale en position
définie dans l'espace articulaire**

Avec,

$$U = \ddot{q}_d + K_v(\dot{q}_d - \dot{q}) + K_p(q_d - q) \quad (\text{B.2})$$

Où K_p et K_v sont des matrices diagonales définies positives de dimension $(n \times n)$.

\hat{A} , \hat{B} sont respectivement les valeurs estimées de A et B , et $\hat{\tau}_e = J^T \hat{F}_e$ le vecteur des couples de forces de contact estimées exercées par le robot sur son environnement.

Ainsi et à partir des équations (3.2), (B.1) et B .2) le système en boucle fermée sera représenté par l'équation découplée et linéarisée suivante:

$$\ddot{e} + K_v \dot{e} + K_p e = \hat{A}^{-1}(\Delta A(q)\ddot{q} + \Delta C(q, \dot{q}) + \Delta \tau_e + \tau_f) \quad (\text{B.3})$$

où :

$e = q_d - q$, $\Delta A(q) = A(q) - \hat{A}(q)$; $\Delta C(q, \dot{q}) = C(q, \dot{q}) - \hat{C}(q, \dot{q})$ et $\Delta \tau_e = \hat{\tau}_e - \tau_e$, avec $\tau_e = J^T F_e$.

Dans le cas idéal où le modèle dynamique est connu parfaitement et que les frottements sec et visqueux sont considérés nuls c'est-à-dire :

$\Delta A = \Delta C = \tau_f = \Delta \tau_e = 0$, l'équation (B.3) sera de la forme:

$$\ddot{e} + K_v \dot{e} + K_p e = 0 \quad (\text{B.4})$$

Dans le cas idéal les performances de cette commande sont facilement atteintes si on impose à cette équation de l'erreur (équation (B.4)) une dynamique désirée comme suit:

$$\ddot{e} + 2\xi_p \omega_p \dot{e} + \omega_p^2 e \quad (\text{B.5})$$

Avec, ξ_p le coefficient d'amortissement et ω_p la pulsation propre. Par identification de (B.5) avec (B.4) on obtient:

$$\begin{aligned} K_p &= \omega_p^2 \\ K_v &= 2\xi_p \omega_p \end{aligned} \quad (\text{B.6})$$

En général, on choisit un amortissement égal à 1 pour avoir une réponse sans dépassement.

Cependant, les performances de cette structure de commande peuvent être dégradées si la connaissance exacte des modèles du robot et de son environnement n'est pas vérifiée c'est-à-dire si les termes ΔA , ΔC , $\Delta \tau_e$ et τ_f sont différents de zéro.

La loi de commande neuronale ϕ générée est basée sur la minimisation d'un critère quadratique dépendant du signal v (équation B.9).

La structure du réseau de neurones proposée ici est la même que celles utilisées dans le chapitre 3 et 4 sauf ses entrées qui deviennent dans ce cas :

$$X_{RN} = \begin{bmatrix} q_d^T(t) & \dot{q}_d^T(t) & \ddot{q}_d^T(t) \end{bmatrix} \quad (\text{B.11})$$

Le réseau de neurones est donc le suivant :

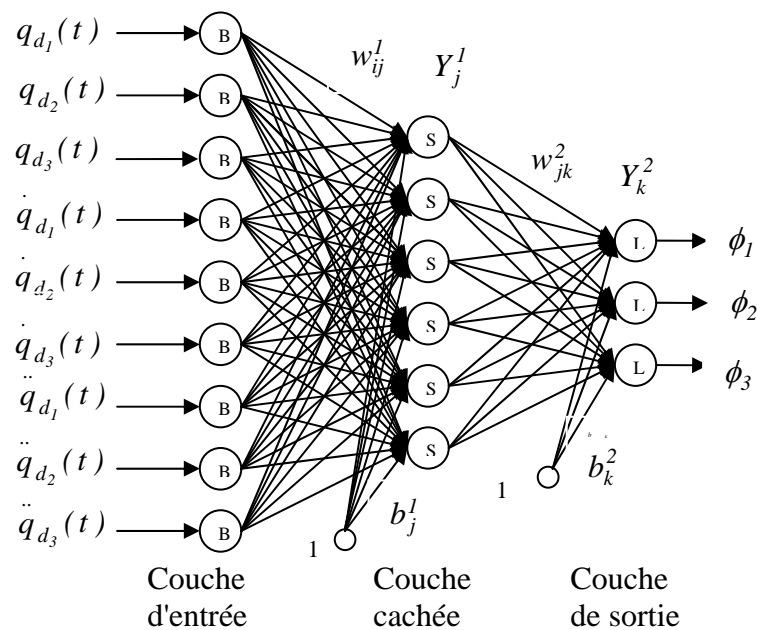


Figure B.3 : Structure du réseau de neurones

La loi d'adaptation des poids du réseau de neurones donné par la figure ci-dessus est obtenue en minimisant la fonction quadratique dépendant du signal v de l'équation (B.8) donnée par l'équation suivante :

$$J = \frac{1}{2} v^T v \quad (\text{B.12})$$

La dérivée de l'équation (B.12) par rapport au poids w donne :

$$\frac{\delta J}{\delta w} = \frac{\delta v^T}{\delta w} v = -\frac{\delta \phi^T}{\delta w} v \quad (\text{B.13})$$

L'adaptation des poids est obtenue en utilisant l'algorithme de rétro propagation du gradient donnée par l'équation suivante :

$$\Delta w(t) = \eta \frac{\delta \phi^T}{\delta w} v + \mu \Delta w(t-1) \quad (\text{B.14})$$

Où η est le taux d'adaptation et μ est le facteur d'oubli ou momentum.

Résultats de simulation

Les tests de simulation effectués visent à évaluer les performances de la structure de commande proposée ci-dessus, dans le cas du suivi d'un cercle sur un plan incliné (figure 3.10) en considérant les mêmes conditions de simulation énoncées dans le chapitre 3 pour cette même tâche.

Les gains de la loi de commande par découplage ainsi que ceux de la boucle de commande en effort sont choisis dans le cas idéal comme suit :

$$K_p = \text{diag}[900,900,900] ; K_v = \text{diag}[60,60,60] \text{ et } K_i = \text{diag}[3,3,3] \times 10^{-4}$$

Les valeurs des paramètres du réseau de neurones sont fixées comme suit:

$$\eta=0.0018 ; \mu=0.9 \text{ et } (w=0).$$

Les résultats de simulation obtenus sont donnés par les figures suivantes : On remarque une nette amélioration du suivi de la trajectoire selon la direction y (mouvement libre) (figure B.4 et B.5) en introduisant le compensateur neuronal (les erreurs selon la direction x et z représentent la pénétration de l'effecteur dans la surface). Dans le cas du suivi de l'effort appliqué (figure B.6 et B.7), on remarque que l'introduction des variations brusques de la raideur de l'environnement induit des pics suivi d'oscillations d'amplitude nettement amortie et de durée très faibles (inférieur à 0.5s) en introduisant le compensateur neuronal. La comparaison des figures B.8 et B.9 montre de façon claire la compensation des perturbations par cette approche neuronale. Les figures B.10 et B.11 représentent respectivement l'évolution de la trajectoire dans l'espace sans et avec réseau de neurones.

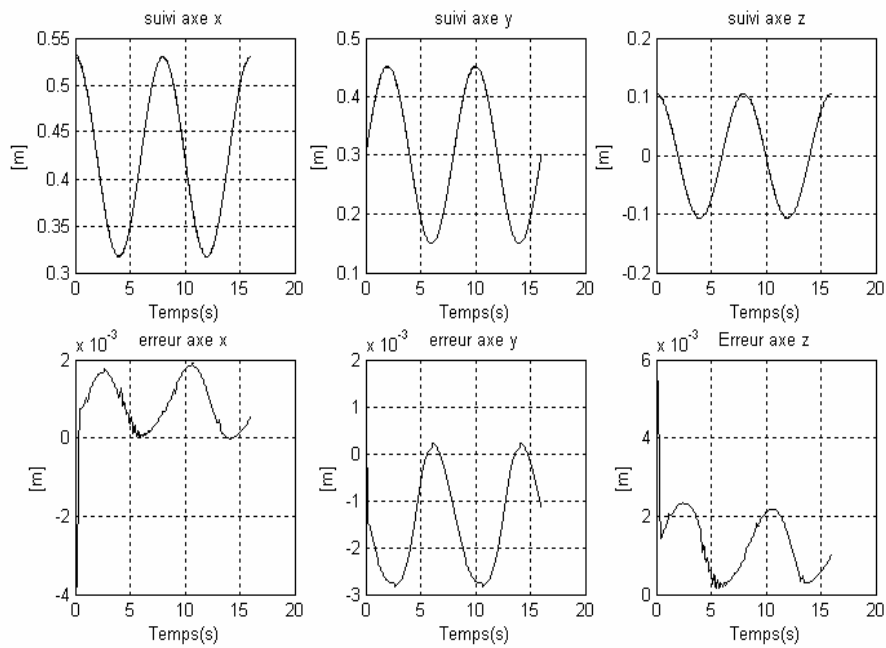


Figure B.4 Evolution de la position (sans le compensateur neuronal)

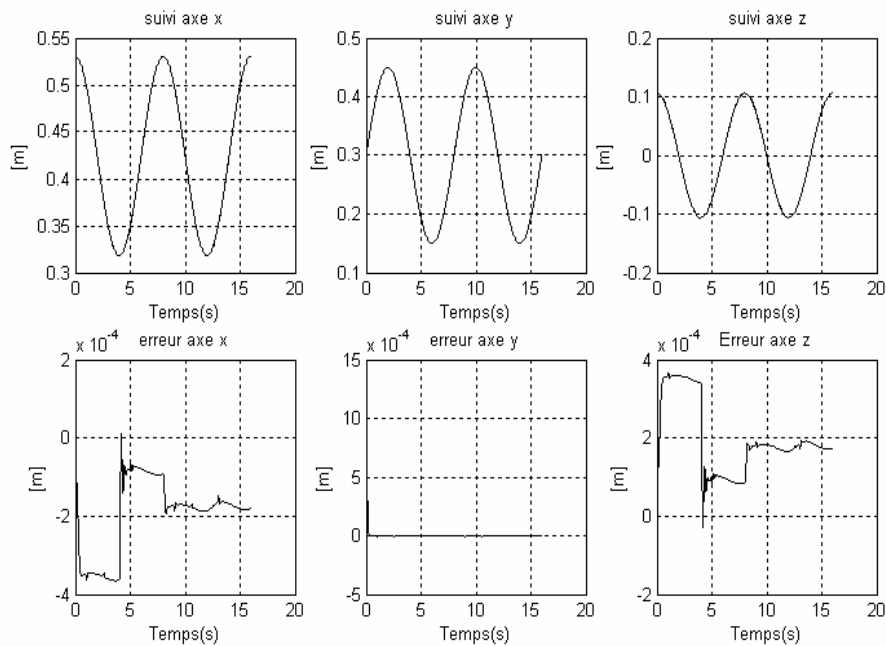


Figure B.5 Evolution de la position (avec compensateur neuronal)

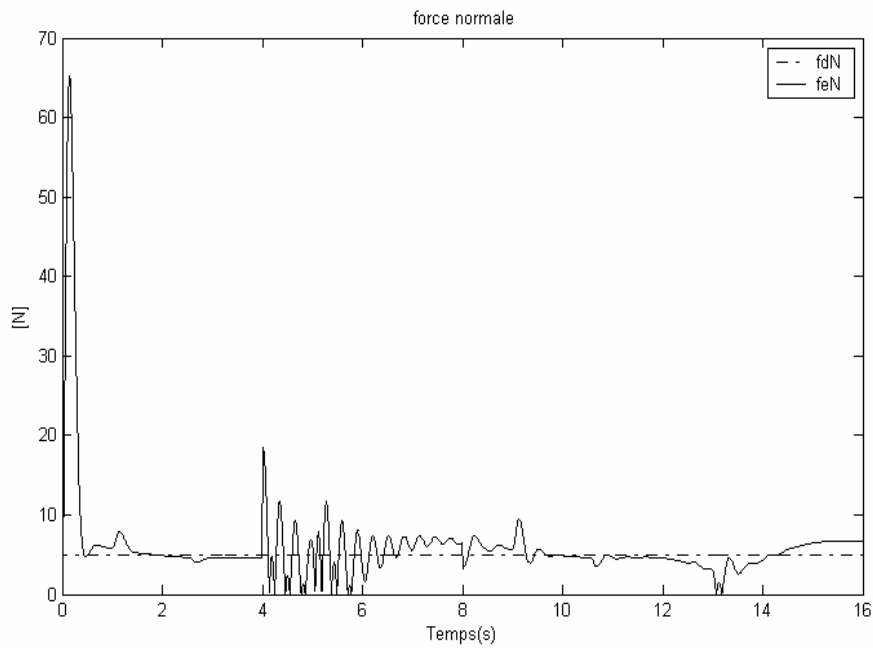


Figure B.6 : Evolution de l'effort (sans compensateur neuronal)

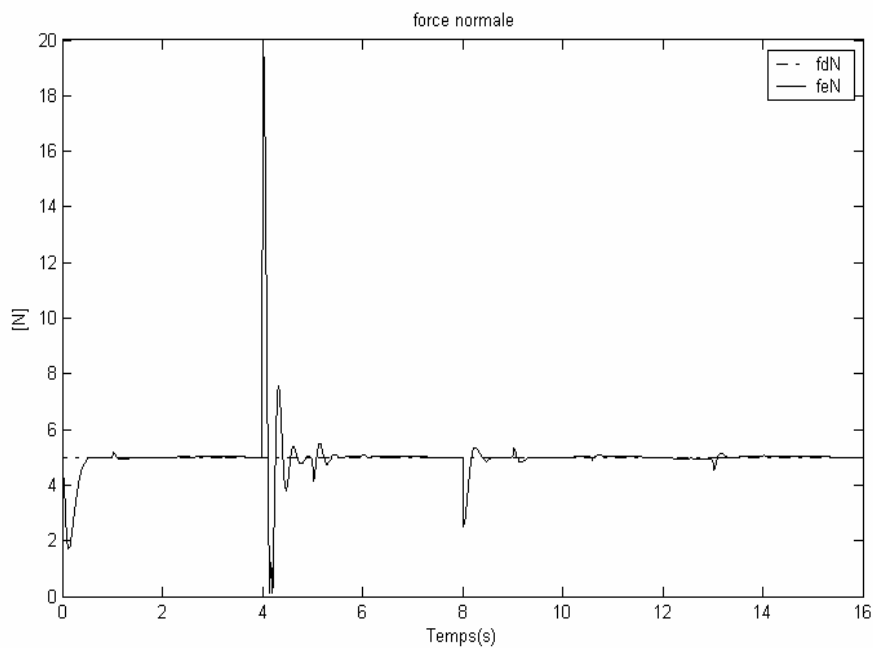


Figure B.7 : Evolution de l'effort (avec compensateur neuronal)

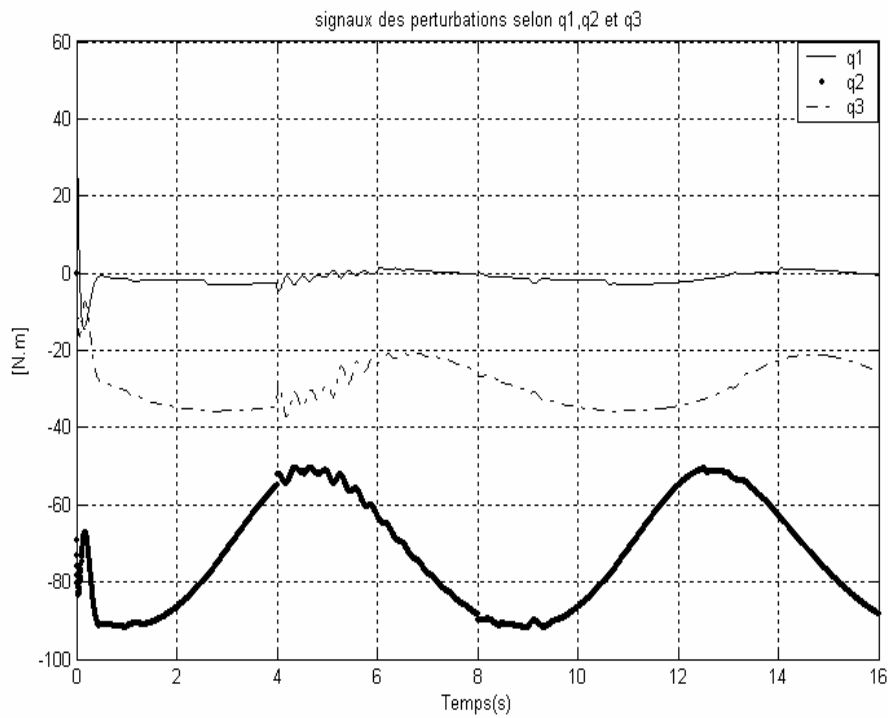


Figure B.8 : Evolution des signaux des perturbations (sans compensateur neuronal)

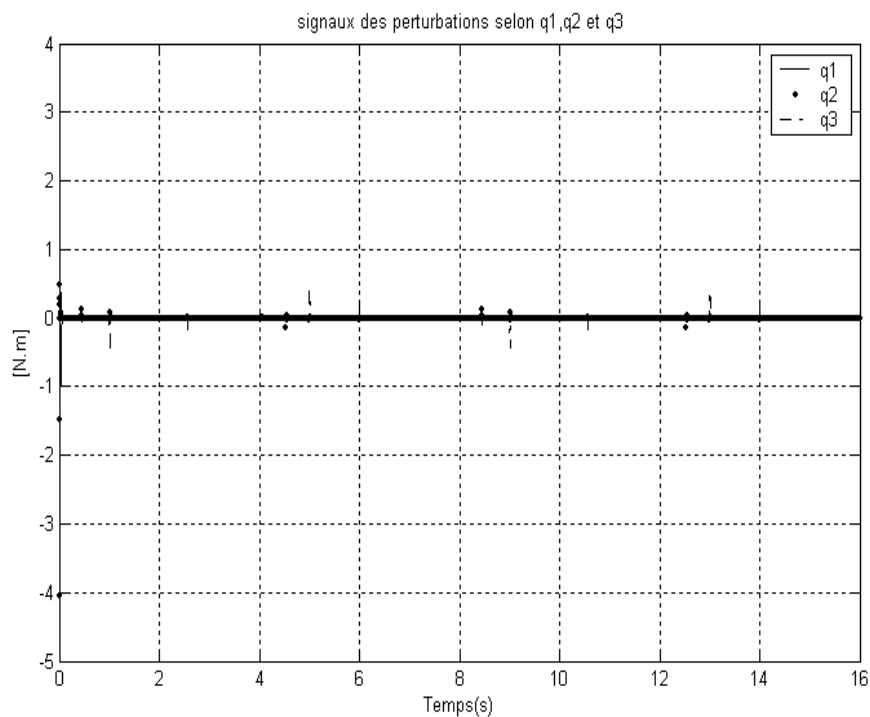


Figure B.9 : Evolution des signaux des perturbations (avec compensateur neuronal)

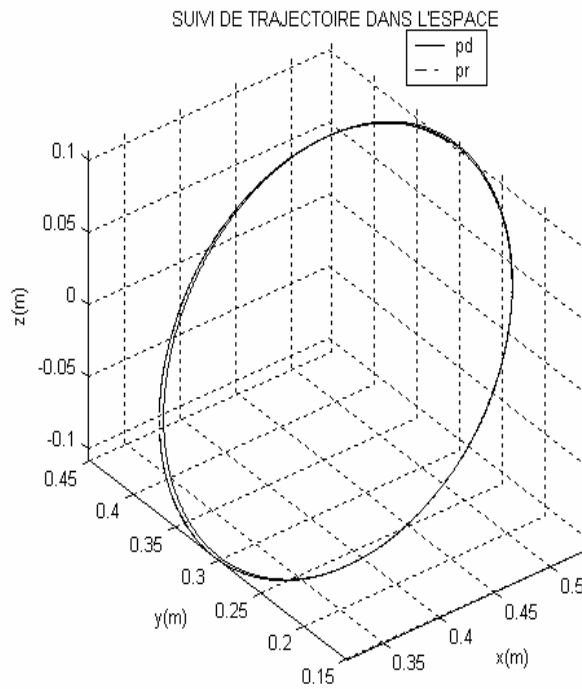


Figure B.10 Evolution de la trajectoire dans l'espace (sans compensateur neuronal)

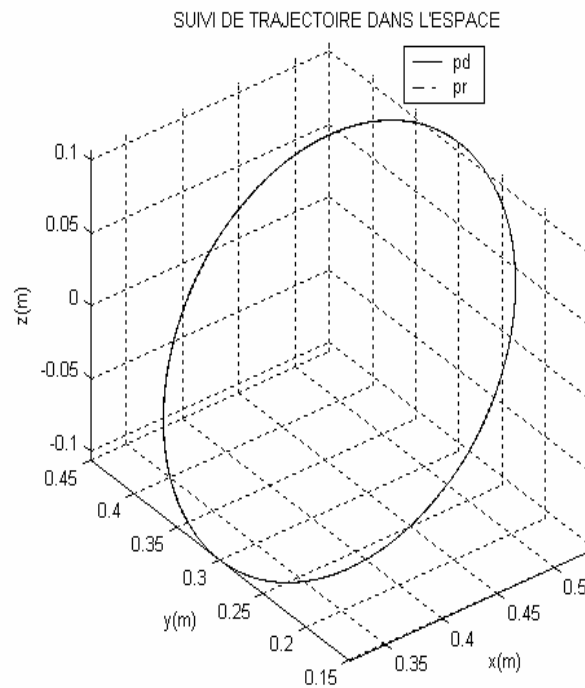


Figure B.11 Evolution de la trajectoire dans l'espace (avec compensateur neuronal)

B.2 Structure externe avec boucle de commande neuronale en position à base d'un contrôleur PD classique

La technique de compensation neuronale basée sur la modification de la trajectoire de référence proposée dans les chapitres 3 et 4 peut être étendue facilement à la structure de commande en effort externe, utilisant une boucle en position dans l'espace articulaire. La structure proposée ici est donnée par la figure B.12 suivante :

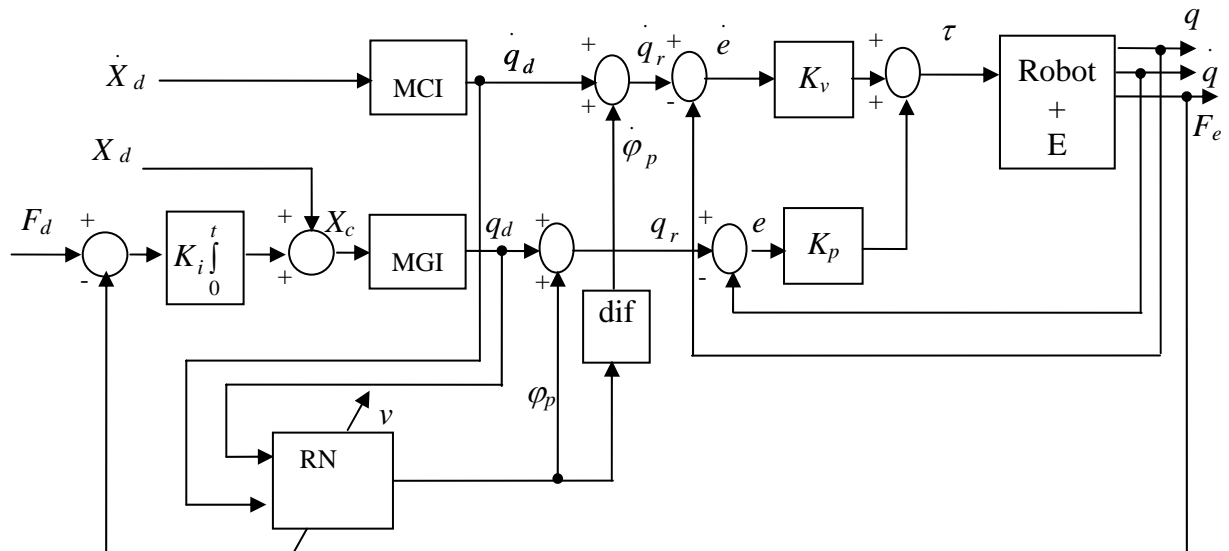


Figure B.12 : Structure externe avec boucle de commande neuronale en position à base d'un contrôleur PD classique

En regardant la figure B.12, la loi de commande en position est donnée par:

$$\tau = K_p (q_d + \varphi_p - q) + K_v (\dot{q}_d + \dot{\varphi}_p - \dot{q}) \quad (\text{B.15})$$

La combinaison de l'équation (B.15) et (3.2) nous donne l'équation en boucle fermée suivante :

$$K_v \dot{e}_p + K_p e_p = A(q)\ddot{q} + C(q, \dot{q}) + \tau_f + \tau_e - (K_p \varphi_p + K_v \dot{\varphi}_p) \quad (\text{B.16})$$

Avec, $e_p = q_d - q$

On pose:

$$\Phi_p = K_p \varphi_p + K_v \dot{\varphi}_p \quad (\text{B.17})$$

L'objectif recherché est de générer Φ_p pour faire tendre l'équation (B.16) vers zéro.

Si on pose

$$v = K_v \dot{e}_p + K_p e_p \quad (\text{B.18})$$

Alors, dans le cas idéal lorsque $v=0$ on aura :

$$\Phi_p = A(q)\ddot{q} + C(q, \dot{q}) + \tau_f + \tau_e \quad (\text{B.19})$$

Le réseau proposé (figure B.13) reste de même architecture que le cas précédent, sauf ses entrées qui deviennent dans ce cas comme suit:

$$X_{RN} = \begin{bmatrix} q_d^T(t) & \dot{q}_d^T(t) \end{bmatrix} \quad (\text{B.20})$$

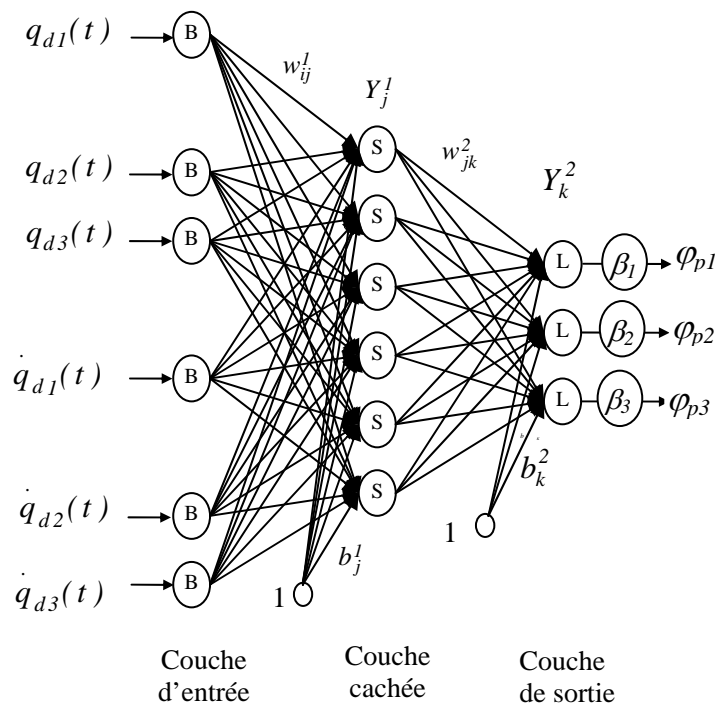


Figure B.13 : structure du réseaux de neurones

Les β_i ($i=1,2,3$) sont des gains permettant de régler l'amplitude du vecteur signal φ_p . La loi d'adaptation des poids du réseau est obtenue en minimisant la fonction quadratique dépendante du signal v de l'équation (B.18) donnée comme suit:

$$J = \frac{1}{2} v^T v \quad (\text{B.21})$$

La dérivée de l'équation (B.21) par rapport au poids w donne

$$\frac{\delta J}{\delta w} = \frac{\delta v^T}{\delta w} v = - \frac{\delta \Phi_p^T}{\delta w} v \quad (\text{B.22})$$

L'adaptation des poids est obtenue en utilisant l'algorithme de rétro propagation du gradient donnée par l'équation suivante :

$$\Delta w(t) = \eta \frac{\delta \Phi_p^T}{\delta w} v + \mu \Delta w(t-1) \quad (\text{B.24})$$

Où η est le taux d'adaptation et μ est le facteur d'oubli ou momentum.

Résultats de simulation

Pour valider et comparer les performances de cette méthode avec celles obtenues ci-dessus (structure basée sur le modèle dynamique), nous avons repris la même simulation précédente (suivi d'un cercle sur un plan incliné de 45°).

Les paramètres du contrôleur PD et ceux du réseau de neurones sont choisis comme suit :

$$K_p = \text{diag}[900,900,900] ; K_v = \text{diag}[60,60,60] \text{ et } K_i = \text{diag}[6,6,6] \times 10^{-4}.$$

Les valeurs des paramètres du réseau de neurones sont fixées comme suit:

$$\eta=0.0003 ; \mu=0 ; \beta_1=1/2 \times 10^{-4} ; \beta_2=10^{-4} ; \beta_3=1/6 \times 10^{-4}.$$

Les résultats de simulations sont représentés par les figures B.14 - B.16.

On remarque que la trajectoire est nettement suivie selon la direction libre y (les autres directions représentent la pénétration de l'effecteur dans la surface). Pour la réponse en effort on remarque que l'introduction des variations brusques de la raideur de l'environnement induit des pics suivis d'oscillations d'amplitude rapidement amortie en un temps inférieur à 0.5s.

Comparativement avec les résultats obtenus avec l'approche neuronale basée sur le modèle dynamique (figure B.5 et B.7), on peut conclure que les performances du suivi de trajectoire et de l'effort appliqué sont similaires.

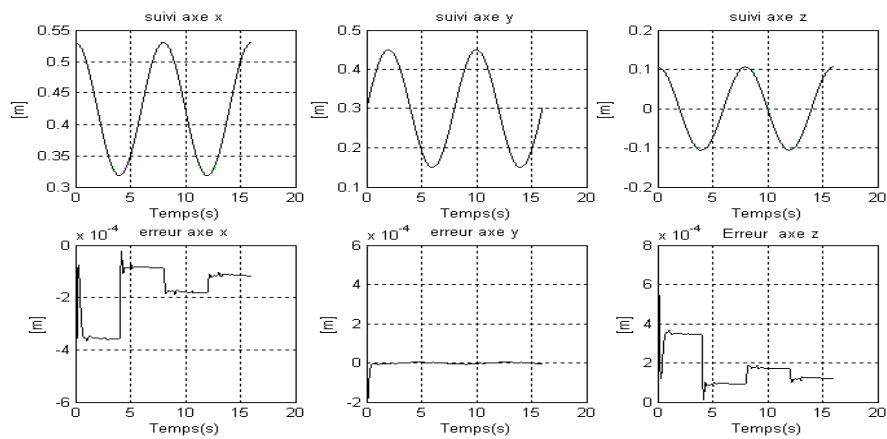


Figure B.14 : Evolution de la position (structure neuronale non basée sur le modèle)

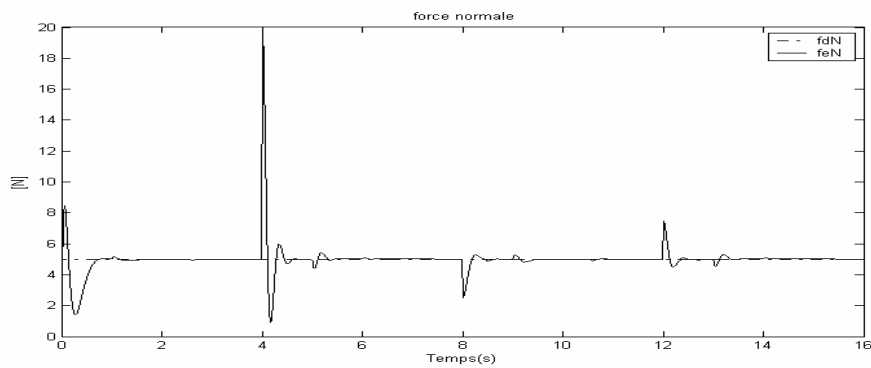


Figure B.15 : Evolution de l'effort normal appliqué (structure neuronale non basée sur le modèle)

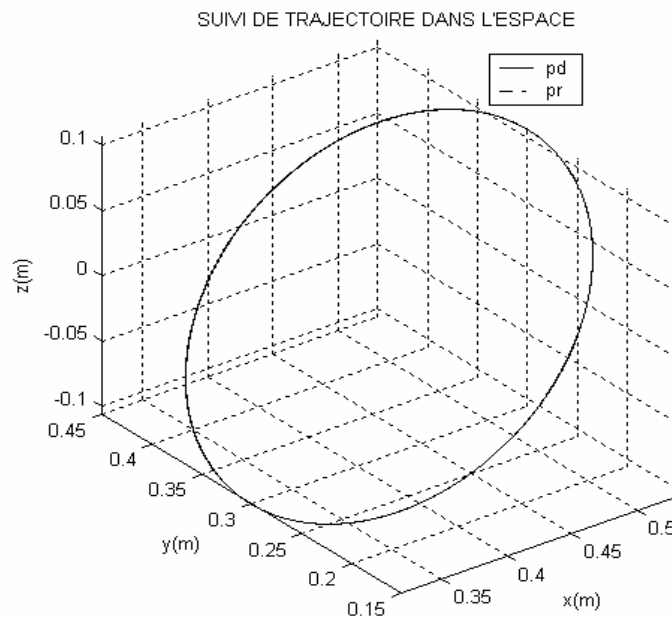


Figure B.16 : Evolution de la trajectoire dans l'espace (Structure neuronale non basée sur le modèle)