

**REPUBLIQUE ALGERIENNE DÉMOCRATIQUE ET POPULAIRE**

MINISTÈRE DE L'ENSEIGNEMENT SUPÉRIEUR ET DE LA RECHERCHE

SCIENTIFIQUE

UNIVERSITÉ DES SCIENCES ET DE TECHNOLOGIE D'ORAN

**MOHAMED BOUDIAF**



FACULTÉ DE GÉNIE ÉLECTRIQUE

DÉPARTEMENT D'ÉLECTRONIQUE

Spécialité: Electronique

Option : Automatique

MÉMOIRE EN VUE DE L'OBTENTION DU DIPLÔME DE  
DOCTORAT EN SCIENCE

Présenté par

**Mr OUAMRI Bachir**

Sujet de mémoire

**Développement d'un Système Intelligent Distant Basé sur les Concepts  
d'Intelligence artificielle : Application aux Bras Manipulateurs**

**Soutenu le 06 / 04 / 2014 Devant le Jury Composé de:**

<i>Président</i>	Mr OMARI Abdelhafid	Professeur	USTO-MB
<i>Rapporteur</i>	Mr AHMED-FOITIH Zoubir	Professeur	USTO-MB
<i>Examineur</i>	Mr KHALFI Med Fayçal	Professeur	Univ. Oran
<i>Examineur</i>	Mr MANSOURI Abdallah	Professeur	ENP-Oran
<i>Examineur</i>	Mr TAHOUR Ahmed	Professeur	Univ. Mascara
<i>Examineur</i>	Mr NOUBAT Wahid	MCA	USTO-MB

## **Remerciements**

*J'adresse tout d'abord mes remerciements à mon directeur de thèse, Ahmed-Foitih Zoubir Professeur à l'université des sciences et de la technologie d'oran USTO-MB, à la fois présent et disponible, il a encouragé mes initiatives au travers de la grande liberté d'actions qu'il m'a autorisé et pour la confiance qu'il m'a témoignée tout au long de la préparation de ce travail.*

*Je tiens à exprimer mes remerciements et ma reconnaissance au Dr. Omari Abdelhafid Professeur à l'université des sciences et de la technologie d'oran USTO-MB, pour l'honneur qu'il me fait en présidant le jury d'examen de ce travail, aux professeurs : Khalfi M. F. Professeur à l'université d'oran, Manssouri A. Professeur à l'ENP d'oran, Tahour A. Professeur à l'université de Mascara, Nouibat W. Maitre de conférence à l'université des sciences et de la technologie d'oran USTO-MB membres de jury de cette thèse, pour l'attention qu'ils ont portée à ce travail par la lecture de ce mémoire et témoignant ainsi de leurs intérêts pour cet axe de recherche et pour la pertinence de remarques, qui ont permis d'améliorer ce manuscrit et d'ouvrir de nouvelles perspectives.*

*Je remercie sincèrement mes collègues du travail pour leur bonne humeur et l'ambiance amicale qu'ils ont su créer.*

*J'adresse mes sincères remerciements à tous mes enseignants de graduation et post-graduation, intervenants et toutes les personnes qui par leurs paroles, leurs écrits, leurs conseils et leurs critiques ont guidé mes réflexions et ont accepté à me rencontrer et répondre à mes questions durant mes recherches.*

*Mes reconnaissances vont également aux personnels et aux responsables du service de la post-graduation pour leurs qualités de services, pour leurs bons accueils et pour leurs gentilleses et souplesses pendant mes différentes années d'inscription.*

*Enfin, je veux remercier toute ma famille pour leurs encouragements et leur soutien moral.*

## Résumé

Dans le contexte de faire fonctionner un robot, pour la réalisation de tâches complexes et/ou dangereuses (cas d'un milieu hostile), le degré d'autonomie des systèmes robotiques existants est insuffisant et les techniques de téléopération classique sont difficiles à mettre en œuvre à cause de problèmes de récupération des informations et les délais de transmission des données entre l'opérateur et le robot dans son environnement. Afin de contribuer au développement d'un système de télérobotique, pour contrôler un robot Puma 600 à travers Internet, nous avons procédé en premier lieu à l'introduction des techniques d'intelligence artificielle pour la réalisation et la mise en œuvre d'un système intelligent distant de commande en position et en poursuite de trajectoire basé sur les concepts de la logique floue et les réseaux de neurones et neuro-flou (ANFIS). Ces techniques montrent l'amélioration des performances du robot distant du point de vue robustesse, précision de suivi et de stabilité et d'assurer une certaine autonomie au robot. Dans le deuxième lieu, nous avons essayé de combiner des planificateurs avec un système de téléopération dans un environnement virtuel (utilisation de la réalité virtuelle (RV) et de la réalité augmentée (RA)) afin de limiter les actions de l'opérateur à la prise de décision et à la récupération des données (erreurs de précision sur la réalisation des tâches planifiés). L'utilisation de la réalité virtuelle permet à l'opérateur de manipuler, par simulation, le robot, de programmer ses tâches à exécuter et de servir comme interface pour le piloter au cours d'une mission et la réalité augmentée fournit à l'utilisateur une sensation d'immersion et pour mettre à jour l'environnement virtuel au cours du processus de contrôle. Nous avons déduit que dans le cas où nous utilisons intelligemment les capacités de la RV et de la RA, la téléopération est possible même dans le cas d'existence de retards imprévisibles sur le net. L'un des principaux objectifs du système que nous avons développé est de fournir la possibilité de planifier et d'exécuter les trajectoires du robot distant à travers Internet.

**Mots clés :** *Robotique, Robot Puma 600, Commande par couple calculé, Commande position/force, Commande par logique floue, Commande par réseaux de neurones, Commande neuro-floue (ANFIS), Control distant, Internet, Réalité Virtuelle/Augmentée.*

## Abstract

In the context of operating a robot, for the realization of complex tasks and/or dangerous (case of hostile environment), the degree of autonomy of existing robotic systems is insufficient and conventional techniques teleoperation are difficult to implement because of problems of informations recovery and deadlines for data transmission between the operator and the robot in its environment. To contribute to the development of a telerobotic system to control a Puma 600 robot through the Internet, we proceed in first place to the introduction of artificial intelligence techniques for the realization and implementation of an intelligent remote system of position control and trajectory tracking based on the concepts of fuzzy logic and neural networks and neuro-fuzzy. These techniques show improved performance of the remote robot from the point of view robustness, tracking precision and stability and to ensure a degree of autonomy to the robot. in the second place, we have tried to combine planners with a teleoperation system in a virtual environment (use of virtual reality (VR) and augmented reality (AR)) to limit the actions of the operator to decision making and to retrieving data (precision errors on the achievement of planned tasks). The use of virtual reality allows the operator

to manipulate, by simulation, the robot, programmed to perform its tasks and serve as an interface to control it during a mission and augmented reality provides the user a feeling of immersion and to update the virtual environment during the control process. We deduced that if we intelligently use the capabilities of the VR and AR, the teleoperation is possible even in the case of existence of unpredictable delays on the net. One of the main goals of the system that we have developed is to provide the opportunity to plan and execute trajectories of the remote robot over the internet.

**Keywords:** *Robotic, Puma 600 Robot, Computed torque control, position/force control, Fuzzy logic control, Neural Network Control, Adaptive neuro-fuzzy inference system (ANFIS), Remote control, Internet, Virtual/Augmented reality.*

### ملخص

في سياق تشغيل الروبوت، لأداء مهام معقدة و/أو خطيرة (حالة البيئة المعادية)، درجة استقلالية أنظمة الروبوتات الحالية غير كافية وتقنيات التحكم عن بعد الكلاسيكية يصعب استخدامها بسبب مشاكل استرجاع المعلومات ووقت نقل المعطيات بين المشغل والروبوت في بيئته. للمساهمة في تطوير نظام التحكم الآلي عن بعد للسيطرة على الروبوت Puma 600 من خلال شبكة الإنترنت، شرعنا في المرحلة الأولى في إدخال تقنيات الذكاء الاصطناعي لانجاز وتنفيذ نظام ذكي عن بعد للتحكم في الوضعية وتتبع المسار استنادا إلى مفاهيم المنطق الضبابي والشبكات العصبية والعصبية-غامض. تظهر هذه التقنيات تحسين أداء الروبوت البعيد من وجهة نظر المتانة، الدقة في التتبع والاستقرار وضمان درجة من التحكم الذاتي للروبوت. في المقام الثاني، حاولنا دمج مخططات مع نظام التشغيل عن بعد في بيئة افتراضية (استخدام الحقيقة الخيالية (VR) والحقيقة الملقمة (RA)) للحد من إجراءات المشغل لأخذ القرار واستعادة المعطيات (أخطاء الدقة على تحقيق المهام المخططة). استخدام الحقيقة الخيالية يسمح للمشغل تشغيل، عن طريق المحاكاة، الروبوت، برمجة أداء مهامه، وتكون بمثابة واجهة للسيطرة عليه خلال مهمة والحقيقة الملقمة توفر للمستخدم شعور من الغمر، وتحديث البيئة الافتراضية أثناء عملية التحكم. استنتجنا أنه إذا استخدمنا بذكاء قدرات RA وRV، التحكم عن بعد، يكون ممكنا حتى في حالة وجود تأخيرات لا يمكن التنبؤ بها على الشبكة. واحدة من الأهداف الرئيسية للنظام الذي وضعناه هو توفير الفرصة لتخطيط وتنفيذ مسارات الروبوت البعيد عبر شبكة الإنترنت.

**مفتاح:** أنضمه الروبوتات، الروبوت Puma 600 ، التحكم في الوضعية/القوة، التحكم بالمنطق الضبابي، التحكم بالشبكات العصبية، التحكم بالعصبية-غامض (ANFIS)، التحكم عن بعد، الإنترنت، الحقيقة الخيالية/الحقيقة الملقمة.

## Principales Notations et Abréviations

### Liste des figures

<b>Introduction générale</b> .....	1
------------------------------------	---

## *Chapitre I*

### *Etat de l'art sur les modélisations appliquées aux systèmes robotisés / Applications*

Introduction.....	5
I-1. Généralités sur les robots manipulateurs.....	5
I-1-1. Constituants mécaniques des robots manipulateurs.....	5
I-1-2. Définition d'une articulation/nombre de degré de liberté.....	6
I-1-2-1. Articulation rotoïde.....	6
I-1-2-2. Articulation prismatique .....	6
I-1-2-3. Espace articulaire.....	6
I-1-2-4. Espace opérationnel .....	6
I-1-2-5. Redondance.....	6
I-1-2-6. Configuration singulière .....	7
I-2. Description géométrique des robots manipulateurs .....	7
I-2-1. Modèle géométrique direct des robots manipulateurs à chaîne ouverte simple .....	7
I-2-2. Modèle géométrique inverse des robots manipulateurs à chaîne ouverte simple .....	10
I-2-2-1. Position du problème .....	10
I-2-2-2. Calcul du modèle inverse par la méthode de Paul .....	11
I-3. Modèle cinématique directe des robots à structures ouvertes simples.....	12
I-4. Modèle cinématique inverse des robots à structures ouvertes simples .....	14
I-5. Modélisation dynamique d'un robot manipulateur .....	15
I-5-1. Calcul du modèle dynamique par le formalisme de Lagrange.....	15
I-5-1-1. Détermination de l'énergie cinétique.....	16
I-5-1-2. L'énergie potentielle.....	19
I-5-1-3. Formulation du modèle de mouvement du robot.....	19
I-5-2. Propriétés structurelles du modèle dynamique .....	21
I-5-2-1. Propriétés de la matrice d'inertie.....	21
I-5-2-2. Propriétés de la matrice des forces centrifuges et de Coriolis.....	21
I-5-2-3. Propriété du vecteur des forces de gravité.....	23
I-5-2-4. Propriété du vecteur de frottement .....	23
I-5-2-5. Propriété du vecteur des perturbations .....	23
I-5-2-6. Paramétrisation linéaire .....	23
I-6. Exemple de modélisation d'un bras manipulateur Puma 600 .....	24
I-6-1. Modèle géométrique direct du robot Puma 600.....	24
I-6-2. Modèle géométrique inverse du robot Puma 600 .....	26

I-6-3. Modèle dynamique du robot Puma 600 .....	28
Conclusion .....	34
Références bibliographiques .....	35

## **Chapitre II**

### ***Etat de l'art sur les commandes appliquées aux systèmes robotisés / Applications***

Introduction.....	37
II-1. Commande dynamique des robots manipulateurs rigides.....	37
II-1-1. Génération de mouvement entre deux points .....	37
II-1-1-1. Interpolation polynomiale .....	38
II-1-1-2. Interpolation linéaire .....	38
II-1-1-3. Interpolation polynomiale de degré trois.....	38
II-1-1-4. Interpolation polynomiale de degré cinq.....	39
II-1-1-5. Interpolation Bang-Bang .....	40
II-1-2. Lois de commande dynamique du robot manipulateur.....	41
II-1-2-1. Commande classique de type <i>PD</i> .....	41
II-1-2-2. Commande classique de type <i>PID</i> .....	43
II-1-2-3. Commande de type <i>PD</i> avec compensation de l'effet de gravité (Commande point à point) .....	45
II-1-2-4. Commande par découplage non linéaire .....	47
II-1-2-5. Commande de Paden et Panja (poursuite de trajectoire).....	49
II-1-2-6. Loi de commande couple calculé (Computed torque).....	50
II-1-2-7. Commande de Slotine et Li (poursuite de trajectoire) .....	52
II-2. Commande en effort.....	53
II-2-1. Compliance passive .....	54
II-2-2. Commande par raideur active .....	55
II-2-3. Commande en impédance.....	57
II-2-4. Commande hybride force/position .....	58
II-2-4-1. Différentes classes de commandes hybrides .....	58
II-2-5. Commande hybride externe .....	60
II-3. Applications à la commande d'un bras manipulateur Puma 600.....	62
II-3-1. Commande point à point.....	62
II-3-2. Commande en poursuite de trajectoire ( <i>Couple Calculé</i> ).....	64
Conclusion .....	66
Références bibliographiques .....	68

**Chapitre III****Etat de l'art sur les techniques intelligentes de commande / Applications**

Introduction.....	71
III-1. La logique floue.....	71
III-1-1. Historique de la logique floue .....	71
III-1-2. Principe de base et définition .....	71
III-1-2-1. Définition d'un sous-ensemble flou.....	72
III-1-2-2. Opérateurs logiques.....	73
III-1-2-2-1. t-norme et s-norme .....	73
III-1-2-3. Inférence floue .....	74
III-1-3. Structure d'un contrôleur flou .....	75
III-1-3-1. Fuzzification.....	76
III-1-3-2. Inférence.....	76
III-1-3-2-1. Méthode d'inférence Max/Min .....	77
III-1-3-2-2. Méthode d'inférence Max/Produit.....	77
III-1-3-2-3. Méthode d'inférence Somme/Produit .....	77
III-1-3-2-4. Méthode d'inférence TVFI (True Value Fuzzy Interférence ou méthode des valeurs concrètes) .....	77
III-1-3-3. Défuzzification.....	78
III-1-3-3-1. Méthode du centre de gravité.....	78
III-1-3-3-2. Méthode par valeur maximum .....	78
III-2. Les réseaux de neurones .....	79
III-2-1. Principe.....	79
III-2-2. Réseaux de neurones .....	80
III-2-3. Architectures des réseaux de neurones.....	81
III-2-3-1. Réseaux non bouclés.....	81
III-2-3-2. Réseaux bouclés.....	81
III-2-3-3. Réseaux à connexions complexes .....	81
III-2-4. Quelques modèles de réseaux de neurones .....	82
III-2-4-1. Le perceptron multicouche.....	82
III-2-4-2. Le modèle Adaline .....	82
III-2-4-3. Réseaux de neurones à fonctions de base radiale .....	83
III-2-5. Apprentissage .....	83
III-2-5-1. Apprentissage supervisé.....	84
III-2-5-2. Apprentissage non supervisé.....	84
III-2-6. Algorithme d'apprentissage .....	84
III-2-6-1. L'algorithme de rétro-propagation du gradient.....	84
III-2-7. Applications des réseaux de neurones.....	88
III-2-7-1. Identification par réseaux de neurones.....	88

III-2-7-1-1. Principe de l'identification.....	88
III-2-7-1-2. Identification directe .....	88
III-2-7-1-3. Identification inverse.....	89
III-2-7-2. Commande neuronale .....	89
III-2-7-2-1. Commande inverse.....	89
III-2-7-2-2. Commande neuronale hybride .....	90
III-2-7-2-3. Commande par modèle de référence.....	90
III-2-7-2-4. Commande par modèle interne (IMC).....	91
III-2-8. Systèmes neuro-flous .....	91
III-2-8-1. ANFIS (Adaptive Neuro Fuzzy Inference System) .....	92
III-3. Application à la commande du bras manipulateur Puma 600 .....	94
III-3-1. Commande par Couple calculé basé sur la logique floue .....	94
III-3-1-1. Stratégie de la commande floue .....	94
III-3-2. Commande par réseaux de neurones .....	99
III-3-3. Commande neuro-floue.....	101
Conclusion .....	104
Références bibliographiques .....	105

## *Chapitre IV*

### *Etat de l'art sur la téléopération distante à travers le web, la réalité virtuelle et la réalité augmentée appliquées aux contrôles / Applications*

Introduction.....	108
IV-1. Historique .....	108
IV-2. Téléopération.....	110
IV-2-1. Problèmes rencontrés dans la téléopération .....	111
1V-2-1-1. Les retards de transmission .....	111
1V-2-1-2. La prise en compte du facteur humain et sécurité.....	112
1V-2-1-3. Degré d'autonomie du système.....	112
IV-3. Télérobotique.....	112
IV-3-1. Quelques domaines d'application de la téléopération.....	113
IV-4. Les outils de réalité virtuelle et de réalité augmentée pour la téléopération .....	118
IV-4-1. Définition de la RV .....	119
IV-4-2. L'Environnement virtuel.....	119
IV-4-3. Définition de la réalité augmentée (RA) .....	120
IV-4-4. Domaines d'application de la RV et de la RA .....	120
IV-5. Téléopération/Télérobotique via Internet.....	122
IV-5-1. Bref historique.....	122
IV-5-2. Architecture matérielle et logicielle .....	124
IV-6. Application à la téléopération du bras manipulateur Puma 600.....	125

IV-6-1. Téléopération par l'utilisation d'un joystick .....	126
IV-6-1-1. Réalisation expérimentale.....	126
IV-6-2. Téléopération distante par Internet du bras manipulateur Puma 600.....	132
IV-6-2-1. Description du système de téléopération développé.....	132
IV-6-2-2. Le système distant.....	133
IV-6-2-3. L'interface Homme-Machine .....	134
IV-6-2-4. Communication Client-Serveur .....	135
IV-6-2-5. Présentation des résultats .....	136
Conclusion .....	138
Références bibliographiques .....	139
<b>Conclusion générale et perspectives .....</b>	<b>144</b>
<b>Annexes.....</b>	<b>148</b>

${}^{j-1}A_j$	Matrice d'orientation de la base du repère $R_j$ par rapport à celle du repère $R_{j-1}$ .
$C(q, \dot{q})$	Matrice des forces centrifuges et de Coriolis.
$dq$	Les différentielles des coordonnées articulaires.
$d_n$	Vecteurs de translation différentielle du repère $R_n$ .
$dX$	La déformation ou déplacement opérationnel due au contact.
${}^c dX_c$	Déplacement différentiel calculé et exprimé dans le repère $R_c$ .
$(e, \Delta e)$	Valeurs physiques de l'erreur de position et de sa dérivée.
$(e_n, \Delta e_n)$	Valeurs normalisées de l'erreur de position et de sa dérivée.
$F$	Force produite en réponse à un déplacement $dX$ .
$F_e$	Couples/forces extérieurs qu'exerce le robot sur l'environnement par l'organe terminal.
$F_s$	Terme de frottements secs.
$F_v$	Matrice de coefficients des frottements visqueux.
$G(q)$	Vecteur de force de gravité.
$I_n$	Matrice identité d'ordre $n$ .
$J(q)$	Matrice jacobienne du mécanisme (robot).
$J_{ij}$	L'élément $(i, j)$ de la matrice jacobienne $J$ .
$J_n$	Jacobien de base.
$K(q, \ddot{q})$	Energie cinétique du robot.
$K_c$	Matrice de raideur désirée, diagonale par rapport au repère $R_c$ .
$K_E$	La raideur effective de l'environnement et du capteur.
$K_f$	Le gain en effort.
$K_p, K_v$	Matrices diagonales définies positives respectivement des gains de position et de vitesse. Aussi $K_p$ : Matrice de raideur dans l'espace articulaire correspondant à $K_c$ .
$L(q, \dot{q})$	Le Lagrangien.
$M(q)$	Matrice d'inertie du bras.
$m_{ij}(q)$	Eléments génériques de la matrice d'inertie $M(q)$ .
$n$	Nombre de degrés de liberté de la structure mécanique
$O_c$	Centre de compliance.
$P$	Matrice symétrique définie positive.
$P_0$	Borne supérieure pour la norme de $P$ .
${}^{j-1}P_j$	Matrice des coordonnées de l'origine $O_j$ du repère $R_j$ dans le repère $R_{j-1}$ .
$q$	Vecteur des positions articulaires (coordonnées généralisées).
$\dot{q}$	Vecteur des vitesses articulaires (vitesses généralisées).

$\ddot{q}$	Vecteur des accélérations articulaires.
$q_d$	Vecteur des positions articulaires désirées.
$\dot{q}_d$	Vecteur des vitesses articulaires désirées.
$\ddot{q}_d$	Vecteur des accélérations articulaires désirées.
$\tilde{q}$	Erreur de position.
$\dot{\tilde{q}}$	Erreur de vitesse.
$\ddot{\tilde{q}}$	Erreur d'accélération
$Q$	Vecteur des forces de compensation de la gravité
$\mathfrak{R}^n$	La dimension de l'espace articulaire
$\mathfrak{R}^m$	La dimension de l'espace opérationnel
${}^i r$	Vecteur des coordonnées d'un point du corps $i$ , par rapport au repère $R_i$
${}^i \bar{r}$	Vecteur des coordonnées de centre de gravité du corps $i$ dans le repère $R_i$
$S = \{(\tilde{q}, \dot{\tilde{q}})\}$	Ensemble des points d'équilibres du système en boucle fermée
${}^{j-1}T_j$	Matrice de transformation homogène
${}^0T_n$	Matrice de transformation homogène du repère terminal $R_n$ dans le repère de base $R_0$ .
${}^fT_{Ed}$	Matrice de transformation homogène définissant la situation désirée du repère outil $R_E$ par rapport au repère atelier.
${}^fT_0$	Matrice de transformation homogène définissant la situation du robot (repère $R_0$ ) dans le repère atelier.
${}^nT_E$	Matrice de transformation homogène définissant le repère outil $R_E$ dans le repère terminal $R_n$ .
$U(q)$	Energie potentielle du robot.
$U_0$	Une constante positive choisie pour que la fonction de Lyapunov $V$ soit définie positive.
$V(\tilde{q}, \dot{\tilde{q}})$	Fonction candidate de Lyapunov.
$v_n$	Vecteur des vitesses de translation du repère $R_n$ .
$\omega_n$	Vecteur des vitesses de rotation du repère $R_n$ .
$w_{ji}$	Le poids de la connexion entre le neurone $i$ de la couche précédente et le neurone $j$ de la couche courante.
$x$	Représentation du vecteur de position dans l'espace d'état.
$X$	Vecteur des positions et orientations (coordonnées opérationnelles) de l'effecteur.
$Y(q, \dot{q}, \ddot{q})$	Matrice des fonctions dépendantes des variables articulaires, appelée aussi regresser.
$Z(p)$	Impédance mécanique équivalente du robot.
$\Gamma$	Vecteur des couples/forces appliqués par les actionneurs (couples généralisés).

$\Gamma_d$	Terme de perturbations ou bien des couples résistants.
$\Gamma_0$	Un terme auxiliaire de commande avec unité d'accélération.
$\mu_1, \mu_2$	Des scalaires positifs minorant et majorant la matrice d'inertie.
$\theta$	Vecteur contenant tous les paramètres du robot.
$\delta_n$	Vecteurs de rotation différentielle du repère $R_n$ .
$\sigma$	Une constante égale à 1 si l'articulation est prismatique et 0 si elle est rotoïde.
$\eta$	Une constante positive appelée pas du gradient.
$\mu_A(x)$	Le degré d'appartenance de l'élément $x$ à l'ensemble $A$ .
ADALINE	Réseau de neurones adaptatif linéaire ( <i>ADaptive LInear NEuren</i> ).
FIS	Fuzzy Inference System
ANFIS	Adaptive Neuro-Fuzzy Inference System.
ANN	Artificiel Neural Network
GUI	<i>Graphical User Interface</i>
IMC	<i>Internal Model Control</i> ou Commande par modèle interne.
$Kp\_Flou$	Gain proportionnel flou
$Kv\_Flou$	Gain de vitesse flou
$Kp\_Rés\_Neur$	Gain neuronal proportionnel
$Kv\_Rés\_Neur$	Gain neuronal de vitesse
LCP ou LCPA	Loi de commande en position articulaire
LCF	Loi de commande en effort (force).
MGD	Modèle géométrique direct
MGI	Modèle géométrique inverse
$PD$	Commande Proportionnelle dérivée.
$PID$	Commande Proportionnelle Intégrale et Dérivée
RCC	<i>Remote Center Compliance</i> : une structure de commande par compliance passive.
RN	réseaux de neurone
RV	Réalité Virtuelle
RA	Réalité Augmentée
TCP / IP	Transmission Control Protocol/Interconnection Protocol
VRML	<i>Virtual Reality Modeling Language</i> , un standard de modélisation 3D sous matlab/simulink.

## Liste des Figures

Figure I-1 : Schéma symbolique d'une articulation rotoïde .....	6
Figure I-2 : Schéma symbolique d'une articulation prismatique.....	6
Figure I-3 : Structure géométrique d'un robot manipulateur à chaîne ouverte simple.....	8
Figure I-4 : Paramètres de Denavit-Hartenberg modifiés.....	8
Figure I-5 : les différentes transformations entre repères du robot.....	10
Figure I-6 : vitesse de la $k^{ième}$ articulation dans le repère $R_n$ .....	13
Figure I-7 : Schéma de principe du robot Puma 600 avec ses paramètres de Denavit-Hartenberg (D-H).....	24
Figure I-8 : Représentation détaillées des paramètres de D-H du robot Puma 600.....	26
Figure II-1: Interpolation linéaire sur une articulation donnée $j$ .....	38
Figure II-2 : Interpolation polynomiale de degré trois sur une articulation donnée $j$ .....	39
Figure II-3 : Interpolation polynomiale de degré cinq sur une articulation donnée $j$ .....	40
Figure II-4 : Interpolation Bang-Bang sur une articulation donnée $j$ .....	41
Figure II-5 : Schéma bloc de la commande classique $PD$ .....	42
Figure II-6 : Schéma bloc d'une commande classique $PID$ .....	44
Figure II-7 : Schéma bloc d'une commande $PD$ avec compensation de l'effet de gravité.....	46
Figure II-8 : Schéma bloc de la commande par découplage non linéaire .....	48
Figure II-9 : Schéma bloc de la commande de Paden et Panja.....	50
Figure II-10 : Schéma bloc de la commande du couple calculé .....	50
Figure II-11 : Schéma bloc de la commande de Slotine et Li.....	52
Figure II-12: Exemple de principe du RCC (ATI Industrial Automation) .....	55
Figure II-13 : Schéma bloc de la commande par raideur active .....	56
Figure II-14 : Schéma bloc d'une commande par raideur active avec mesure d'effort.....	56
Figure II-15 : Schéma de principe de la commande en impédance .....	57
Figure II-16 : Schéma de principe de la commande hybride force/position.....	58
Figure II-17 : Schéma de la commande hybride force/position avec somme de couples articulaires : $a$ ) Structure initiale de Raibert, $b$ ) Structure de Zhang .....	59
Figure II-18 : Schéma de la commande hybride force/position avec somme des : $a$ ) Erreurs de positions cartésiennes, $b$ ) Efforts .....	60
Figure II-19 : Schéma de principe de la commande hybride externe .....	60
Figure II-20 : Schéma de principe de l'asservissement cartésien avec commande en position cartésienne.....	61
Figure II-21 : Schéma de principe de l'asservissement articulaire avec commande en vitesse cartésienne.....	61
Figure II-22 : Schéma de principe de la commande externe avec asservissement articulaire : $a$ . Transformation indépendante des consignes, $b$ . Transformation globale des consignes.....	62
Figure II-23 : Schéma d'implémentation de la commande point à point de type $PD$ sous Matlab/simulink .....	63

Figure II-24: Les positions mesurées et les erreurs de positions pour la commande Point à Point de type <i>PD</i> .....	64
Figure II-25 : Schéma d'implémentation de la commande Couple Calculé de type <i>PD</i> sous Matlab/Simulink.....	65
Figure II-26 : Les poursuites de trajectoire de position et les erreurs de poursuite de position pour la loi de commande Couple Calculé de type <i>PD</i> .....	65
Figure II-27 : Les poursuites de trajectoire de vitesse et les erreurs de poursuite de vitesse pour la loi de commande Couple Calculé de type <i>PD</i> .....	66
Figure III-1: Exemple de représentation d'une variable en logique booléenne.....	72
Figure III-2: Exemple de représentation d'une variable en logique floue .....	72
Figure III-3: Exemple d'une représentation d'un ensemble flou par sa fonction d'appartenance.....	73
Figure III-4: Schéma de principe d'un régulateur flou .....	75
Figure III-5 : Fuzzification de l'erreur $e$ .....	76
Figure III-6 : Exemple de defuzzification par valeur maximum .....	78
Figure III-7 : Structure générale d'un neurone formel.....	79
Figure III-8 : Fonction binaire à seuil .....	80
Figure III-9 : Fonction linéaire à seuil .....	80
Figure III- 10 : Fonction sigmoïde .....	80
Figure III-11 : Structure de base d'un réseau à couches .....	81
Figure III-12 : Structure de base d'un réseau récurrent .....	81
Figure III-13 : Structure de base d'un réseau totalement interconnecté .....	82
Figure III-14 : Structure d'un Adaline .....	82
Figure III-15 : Principe d'un réseau de neurone de type RBF.....	83
Figure III-16 : Modèle du neurone $j$ .....	85
Figure III-17 : Principe de l'identification neuronale directe d'un processus .....	89
Figure III-18: Principe de l'identification neuronale inverse d'un processus.....	89
Figure III-19: Principe de la commande inverse avec un réseau de neurones.....	90
Figure III-20: Structure de la commande neuronale hybride.....	90
Figure III-21: Principe de la commande neuronale avec modèle de référence.....	91
Figure III-22: Principe de la commande par modèle interne .....	91
Figure III-23: Principe du système neuro-flou.....	92
Figure III-24: Architecture de l'ANFIS .....	92
Figure III-25 : Schéma de principe du régulateur flou proposé.....	95
Figure III-26 : fonction d'appartenance des entrées et des sorties du régulateur flou .....	96
Figure III-27 : Schéma d'implémentation de la commande Couple Calculé de type <i>PD</i> basée sur la logique floue sous Matlab/Simulink .....	96
Figure III-28 : Les poursuites de trajectoire de position et les erreurs de poursuite de position pour la loi de commande Couple Calculé de type <i>PD</i> basée sur la logique floue .....	97
Figure III-29 : Les poursuites de trajectoire de vitesse et les erreurs de poursuite de vitesse pour la loi de commande Couple Calculé de type <i>PD</i> basée sur la logique floue.....	98

Figure III-30 : Principe d'identification des gains $K_p\_Flou$ et $K_v\_Flou$ du correcteur par réseau de neurones .....	99
Figure III-31 : Schéma d'implémentation de la commande Couple Calculé de type $PD$ basée sur les réseaux de neurones sous Matlab/Simulink .....	100
Figure III-32 : Les poursuites de trajectoire de position et les erreurs de poursuite de position pour la loi de commande Couple Calculé de type $PD$ basée sur les réseaux de neurones.....	100
Figure III-33 : Structure du régulateur ANFIS développé.....	102
Figure III-34 : Schéma d'implémentation de la commande Couple Calculé de type $PD$ basée sur le régulateur neuro-flou (ANFIS) sous Matlab/Simulink .....	102
Figure III-35 : Les poursuites de trajectoire de position et les erreurs de poursuite de position pour la loi de commande Couple Calculé de type $PD$ basée sur l'ANFIS.....	103
Figure III-36 : Les poursuites de trajectoire de vitesse et les erreurs de poursuite de vitesse pour la loi de commande Couple Calculé de type $PD$ basée sur l'ANFIS.....	104
Figure IV-1: Télémanipulateur mécanique .....	109
Figure IV-2 : Schéma d'une structure générale d'un système de téléopération .....	111
Figure IV-3: Exemple de robots téléopérés en milieu nucléaire.....	113
Figure IV-4: Exemples de robots téléopérés sous-marin .....	114
Figure IV-5: Exemples de Robot téléopéré dans l'espace .....	115
Figure IV-6: Le système Da Vinci : la console maître et les manipulateurs esclaves .....	116
Figure IV-7. Le système Zeus.....	116
Figure IV-8: Exemples d'engins militaires téléopérés.....	117
Figure IV-9 : Exemples de Robots téléopérés dans le domaine de mines .....	118
Figure IV-10 : Architecture, logicielle et matérielle, générale d'un système de téléopération .....	124
Figure IV-11 : Structure de l'interface virtuelle réalisée du robot Puma 600.....	125
Figure VI- 12 : Le Joystick utilisé dans la télémanipulation avec son principe de fonctionnement ..	126
Figure IV-13 : Vue de couplage par USB entre le joystick et l'ordinateur .....	127
Figure IV-14 : Implémentation de la télémanipulation avec un joystick dans l'espace articulaire du bras virtuel Puma 600 sous Matlab/Simulink .....	128
Figure IV-15: Comportement du robot virtuel Puma 600 pour les différentes manipulations du joystick.....	128
Figure IV-16 : Images du robot virtuel Puma 600 correspondantes aux différentes manipulations en positions articulaires du joystick.....	129
Figure IV-17 : Implémentation de la télémanipulation avec un joystick dans l'espace opérationnel du bras virtuel Puma 600 sous Matlab/Simulink .....	130
Figure IV-18: Comportement du robot virtuel Puma 600 pour les différentes manipulations du joystick dans l'espace opérationnel.....	131
Figure IV-19: Images du robot virtuel Puma 600 correspondantes aux différentes manipulations du joystick dans l'espace cartésien .....	131
Figure IV-20: Architecture du system de téléopération.....	132
Figure IV-21 : Description du système distant .....	133

Figure IV-22 : L'interface graphique utilisateur développé .....	134
Figure IV-23 : Architecture de communication client-serveur avec le protocole TCP/IP.....	135
Figure IV-24: Les positions du robot Puma 600, distant et virtuel, lors de l'exécution d'une trajectoire planifiée.....	136
Figure IV-25 : Algorithme de superposition développé .....	137
Figure IV-26 : Superposition de l'image virtuelle (en fil de fer) sur l'image réelle .....	138

# *Introduction générale*

La manipulation des objets dangereux et la réalisation des travaux dans des environnements difficilement supportables par l'homme (conditions extrêmes de température ou de pression, radioactivité élevée, apesanteur, trop petit, trop grand, très loin, etc.) et où les tâches sont suffisamment complexes ou imprévisibles pour rendre difficile une automatisation complète, conduit les chercheurs à développer des systèmes de commande à distance. C'est la téléopération : qui signifie le contrôle d'un système à un site distant appelé esclave. L'opérateur ou maître et le système distant sont physiquement séparés et doivent interagir pour accomplir une tâche donnée.

Le parcours de la téléopération a connu plusieurs problèmes sur les plans techniques et économiques (Interfaces homme-machine plus complexe, canaux de communication dédiés, limitation dans la distance, etc.). Avec l'évolution rapide de la robotique, des calculateurs et la technologie de transmission et surtout avec l'utilisation d'Internet comme moyen de communication, ont contribué à ouvrir un champ de recherche énorme dans le domaine de la téléopération et de la télérobotique.

La téléopération/télérobotique, avec cette percée technologique, permet de fournir plus de rentabilité, de souplesse et de facilité d'accéder à des systèmes de contrôle qui ne sont pas limités à une région géographique donnée. Ce qui conduit au développement des outils et des interfaces plus sûres et plus fiables de fonctionnement des systèmes de téléopération/télérobotique assurant un soulagement physique et décisionnel de l'opérateur humain.

Malgré ces grandes avancées technologiques, la téléopération/télérobotique est toujours limitée aux systèmes qui incluent constamment l'opérateur dans la boucle de contrôle. Afin de réaliser des manipulations intelligentes avec le minimum possible d'interventions humaines, des recherches plus approfondies, en robotique autonome, motivent les spécialistes actuellement.

L'objectif visé dans ce travail est la contribution au développement d'un système de télérobotique, qui consiste à combiner des planificateurs avec un système de téléopération dans un environnement virtuel (utilisation de la réalité virtuelle et de la réalité augmentée) afin de limiter les actions de l'opérateur à la prise de décision et à la récupération des données (erreurs de précision sur la réalisation de la tâche planifiée).

L'utilisation d'un environnement virtuel permet d'intégrer des outils de planification dans le processus de décision, aussi bien pour la configuration de l'environnement de travail distant (placement de caméras pour l'asservissement visuel par exemple) qu'en ligne (génération de trajectoires sans collision, choix de la saisie d'une pièce,...).

L'utilisation de nouveaux moyens d'interaction tels que : les outils graphiques, les outils d'animation 3D et de réalité virtuelle, outils de réalité augmentée, ... donnent à l'opérateur plus d'immersion dans le processus de contrôle et de commande du système distant et plus de facilité

pour programmer ses tâches à exécuter et de servir comme interface plus sûre et plus fiable pour le piloter au cours d'une mission donnée.

L'introduction des techniques intelligentes de commande, basée sur les concepts de la logique floue et les réseaux neuromimétiques, permettent de donner une grande précision et une certaine autonomie aux systèmes distants réduisant ainsi l'intervention humaine uniquement à la supervision.

Dans ce sens, ce sujet a pour but ; l'étude de certains problèmes engendrés par la téléopération via Internet (Retards, stabilité, sûreté de fonctionnement, ...etc.) et la contribution au développement d'un système intelligent de télérobotique combinant les techniques d'intelligence artificielle au niveau du système distant pour lui donner une certaine autonomie et les outils de réalité virtuelle et augmentée au niveau du site local pour donner plus d'émergence à l'opérateur dans le système de contrôle. Pour cela cette thèse est divisée en quatre chapitres :

L'objectif du premier chapitre est de présenter les différentes définitions et terminologies utilisées dans le domaine de la robotique; les techniques de modélisation géométrique, cinématique et dynamique utilisées pour faire une bonne synthèse de lois de commande des bras manipulateurs seront présentées. Comme application, nous présentons un exemple détaillé sur les démarches à suivre pour le calcul des différents modèles du robot Puma 600.

Le second chapitre sera consacré à la présentation de quelques procédures sur les techniques de génération des trajectoires utilisées dans la commande des manipulateurs. Nous détaillons les différentes lois de commande classique qui traitent les problèmes de régulation et de poursuite de trajectoire. Les différentes techniques de commande en effort et hybrides (position/force) quant le robot rentre en contact avec son environnement seront exposées. Des applications de commande sur un robot Puma 600 seront montrées.

Dans le troisième chapitre, nous abordons les différentes techniques intelligentes de raisonnement et de contrôle appliquées aux systèmes robotisés. Nous donnons les définitions et les terminologies utilisées en logique floue, la théorie des ensembles flous et ainsi que les modes de raisonnement propres aux variables floues. Nous détaillons les étapes nécessaires pour le développement et la synthèse d'un régulateur flou. Quelques notions de base, les différentes architectures et les méthodes d'apprentissage des réseaux de neurones artificiels (RNA) seront exposés. Nous présentons les principales applications des RNA dans le domaine de l'identification ainsi les différentes approches de commande rencontrées dans la littérature scientifique. La technique d'optimisation des systèmes d'inférence flous de type Takagi Sugeno (ANFIS) sera exposée. Afin de montrer l'amélioration des performances de la commande Couple Calculé développée dans le deuxième chapitre, nous proposons deux techniques de commande intelligente, floue et neuro-floue, appliquées sur le modèle dynamique du robot Puma 600.

Le quatrième et le dernier chapitre donne un état de l'art sur l'évolution de la téléopération et la télérobotique. Les définitions, les différents problèmes rencontrés et les domaines d'application de la téléopération/télérobotique seront présentées. Nous détaillons les nouveaux outils de réalité virtuelle et de réalité augmentée appliqués au développement de la téléopération/télérobotique. Les différentes techniques et architectures de la téléopération via internet seront développées. Enfin, nous proposons deux applications pour la téléopération d'un bras manipulateur Puma 600 ; une pour la télémanipulation par un Joystick et l'autre est le développement d'une interface de télérobotique d'un bras Puma 600 via Internet contenant un générateur de trajectoire, les outils de réalité virtuelle et augmentée, retour vidéo et sensoriels. Nous introduisons un régulateur intelligent au niveau du système distant (robot) afin de lui donner une certaine autonomie de fonctionnement.

## *Chapitre I*

# *Etat de l'art sur les modélisations appliquées aux systèmes robotisés / Applications*

## Introduction

La commande des robots manipulateurs constitue à l'heure actuelle l'une des préoccupations majeures des recherches en robotique. Afin de concevoir un système de simulation ou de commande performant, certaines exigences et connaissances sur le robot doivent être connues. La modélisation, constitue une étape très importante dans ce sens, et qui consiste à représenter le système, qui est le robot, sous la forme d'équations mathématiques afin de procéder à des analyses et des études détaillées par simulation sur son comportement. Selon le type de la commande envisagée, plusieurs niveaux de modélisation sont possibles : modèles géométriques, différentiels et dynamiques à partir desquels peuvent engendrés les mouvements du robot. L'obtention de ces différents modèles présente plusieurs difficultés selon la complexité de la cinématique de la chaîne articulée : Type des articulations, le nombre de degrés de liberté, chaîne ouverte simple, arborescente ou fermée ... .

Dans ce chapitre, nous allons essayer de donner quelques terminologies et principes de base sur les robots manipulateurs ; les principales techniques de modélisation des robots à chaîne ouverte simple (mêmes principes pour les autres types d'architectures : arborescente ou fermée) telles que le modèle géométrique direct (MGD) et inverse (MGI), cinématique direct et inverse et dynamique utilisées pour faire une bonne synthèse de lois de commande seront présentées. Ensuite, nous présenterons une application détaillée pour l'obtention des différents modèles du robot Puma 600. Enfin, nous terminerons ce chapitre par une conclusion.

### I-1. Généralités sur les robots manipulateurs

La robotique est une technique fondamentalement pluridisciplinaire car elle utilise de nombreuses technologies: mécanique, pneumatique, hydraulique, électronique, informatique etc., elle est donc un champ vaste. La définition qui peut être la plus précise est celle de l'AFNOR, qui définit ce qu'est un manipulateur comme : « mécanisme multi- fonctionnel à plusieurs degrés de libertés commandé directement par un opérateur humain ou par un calculateur logique » [7].

Le robot industriel est présenté comme un manipulateur automatique asservi en position, reprogrammables, polyvalent et capable de positionner des matériaux, pièces, outils ou dispositifs spécialisés. Il est aussi présent dans les environnements hostiles, où il est destiné à accomplir des tâches pour lesquelles l'homme est incapable de les effectuer (manipulation des matériaux radioactifs dans les cellules chaudes des réacteurs nucléaires, etc.).

#### I-1-1. Constituants mécaniques des robots manipulateurs

Un robot manipulateur est constitué de deux sous-ensembles : la structure mécanique articulée et l'organe terminal. Ce dernier, présente une interface permettant au robot d'interagir avec son environnement par la manipulation de différents objets. Le rôle de la structure mécanique articulée est d'amener l'organe terminal à une position et une orientation donnée (lieu géométrique imposé par la tâche) selon des caractéristiques de vitesse et d'accélération spécifique. Son architecture est une chaîne cinématique de corps généralement rigides, reliés par des liaisons appelées articulations. Les chaînes peuvent être soit simples, soit arborescentes, soit fermées ou parallèles [5, 8, 9, 11].

### I-1-2. Définition d'une articulation/nombre de degré de liberté

Une articulation lie deux corps successifs en limitant le nombre de degré de liberté de l'un par rapport à l'autre. En général, le nombre des paramètres indépendants qui fixent la situation de l'organe terminal est égal au nombre de degré de liberté d'un robot manipulateur.

En robotique nous avons deux types d'articulations : rotoïde et prismatique.

#### I-1-2-1. Articulation rotoïde

Cette articulation notée R, est de type pivot réduisant le mouvement entre deux corps à une rotation autour d'un axe qui leur est commun. Dans ce cas la situation est donnée par un angle de rotation. La figure I-1 représente le symbole d'une articulation rotoïde.

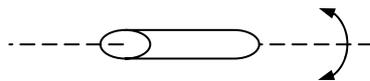


Figure I-1 : Schéma symbolique d'une articulation rotoïde.

#### I-1-2-2. Articulation prismatique

Cette articulation notée P, est de type glissière réduisant le mouvement entre deux corps à une translation le long d'un axe commun. La situation est donnée par un déplacement. La représentation symbolique de celle-ci est donnée par la figure I-2.

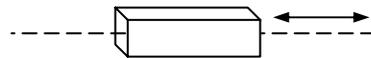


Figure I-2 : Schéma symbolique d'une articulation prismatique.

#### I-1-2-3. Espace articulaire

Est celui qui exprime l'état du robot représentant la situation de ces différents corps. La solution la plus classique pour décrire cet état consiste à utiliser les variables ou coordonnées articulaires. Soit  $\mathcal{R}^n$  cet espace. La valeur  $n$  est égale au nombre de variables articulaires indépendantes et correspond au nombre de degrés de liberté de la structure mécanique. Quant on considère un robot à  $n$  degrés de liberté, nous disposons de  $n$  articulations motorisées.

#### I-1-2-4. Espace opérationnel

Est celui dans lequel est représenté la situation de l'organe terminal. La solution la plus simple consiste à utiliser les coordonnées cartésiennes. Soit  $\mathcal{R}^m$  cet espace. La valeur  $m$  constitue le nombre de degrés de liberté maximum que peut avoir l'organe terminal.

#### I-1-2-5. Redondance

Un robot est redondant lorsque le nombre de degrés de liberté de l'organe terminal est inférieur au nombre d'articulations motorisées. Cette propriété permet d'augmenter le volume du domaine accessible et de préserver les capacités de déplacement de l'organe terminal en présence d'obstacles, le ou les degrés de liberté supplémentaires autorisant leur contournement.

### I-1-2-6. Configuration singulière

Pour tous les robots, qu'ils soient redondants ou non, il se peut que dans certaines configurations dites singulières le nombre de degrés de liberté de l'organe terminal soit inférieur à la dimension de l'espace opérationnel. Ce cas se présente par exemple lorsque :

- Deux axes d'articulation prismatiques se retrouvent parallèles.
- Deux axes d'articulation rotoïdes se retrouvent confondus.

## I-2. Description géométrique des robots manipulateurs

La description géométrique des robots manipulateurs est l'étape de base pour la mise en équation de tous les modèles de robots. Parmi les quels on trouve le modèle géométrique direct et le modèle géométrique inverse.

### I-2-1. Modèle géométrique direct des robots manipulateurs à chaîne ouverte simple

Ce modèle permet de calculer les coordonnées opérationnelles donnant la situation de l'organe terminal en fonction des coordonnées articulaires.

Pour décrire la structure géométrique de tels robots manipulateurs, la méthode la plus utilisée et la plus répandue est la représentation de Denavit-Hartenberg modifiée, présentée par Khalil et Kleinfinger [2, 5, 6, 9, 13]. Cette représentation permet une description homogène, avec un minimum de paramètres, des systèmes mécaniques articulaires à chaînes ouvertes simples et complexes. Cette description est basée sur les règles et les conventions suivantes :

- Les corps sont parfaitement rigides. Ils sont connectés par des articulations considérées comme idéales (pas de jeu mécanique, pas d'élasticité), sont soit rotoïdes soit prismatiques.
- La variable de l'articulation  $j$  est notée  $q_j$  ;
- Le corps  $j$  est noté  $C_j$  ;
- Le repère  $R_j$  est lié au corps  $C_j$  ;
- Les paramètres définissant le repère  $R_j$  par rapport au repère  $R_{j-1}$  sont indicés ( $j$ ).

Le système, qui est le robot, est composé de  $n$  articulations et de  $n+1$  corps notés  $C_0, \dots, C_n$ . Le corps  $C_0$  est la base du robot et le corps  $C_n$  porte l'organe terminal, l'articulation  $j$  connecte le corps  $C_j$  au corps  $C_{j+1}$  comme l'indique la figure I-3.

Le repère  $R_j$ , fixé au corps  $C_j$ , est défini tels que :

- L'axe  $Z_j$  est porté par l'axe de l'articulation  $j$  ;
- L'axe  $X_j$  est porté par la perpendiculaire commune aux axes  $Z_j$  et  $Z_{j+1}$

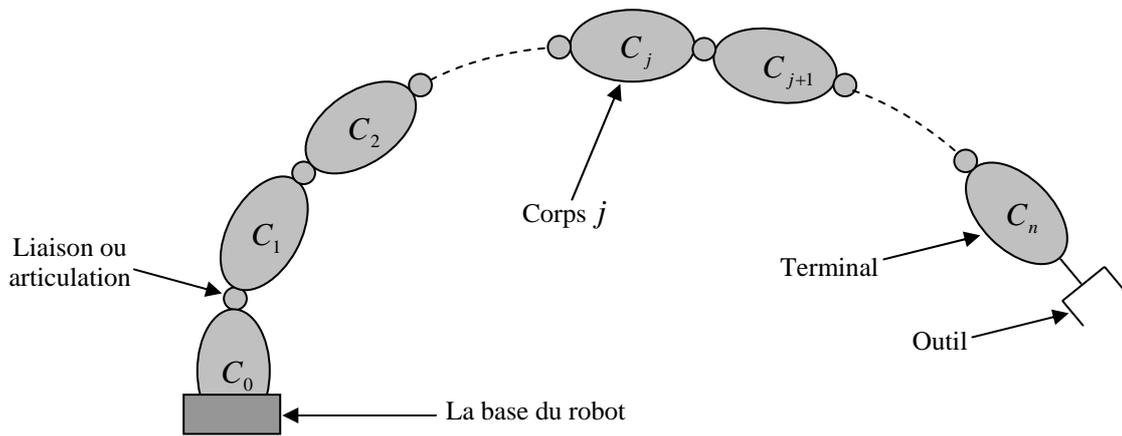


Figure I-3 : Structure géométrique d'un robot manipulateur à chaîne ouverte simple.

Le passage d'un repère  $R_{j-1}$  à un repère  $R_j$  peut être déterminé complètement par les quatre paramètres de Denavit-Hartenberg suivants (voir figure I-4) :

1.  $\alpha_j$  : l'angle entre l'axe  $Z_{j-1}$  et  $Z_j$  mesuré autour de l'axe  $X_{j-1}$  ;
2.  $d_j$  : la distance entre le centre  $O_{j-1}$  du repère  $R_{j-1}$  et l'axe  $Z_j$  ;
3.  $\theta_j$  : l'angle entre les axes  $X_{j-1}$  et  $X_j$  autour de l'axe  $Z_j$  ;
4.  $r_j$  : la distance entre le centre  $O_j$  du repère  $R_j$  et l'axe  $X_{j-1}$ .

Les paramètres  $\alpha_j$  et  $d_j$  sont constants, un des paramètres  $\theta_j$  ou  $r_j$  varie lorsque l'articulation se déplace. Si l'articulation est rotoïde, le paramètre  $\theta_j$  est la variable représentant le déplacement de l'articulation. Si elle est prismatique, le paramètre  $r_j$  est la variable représentant le déplacement de l'articulation, tandis que  $\theta_j$  est constant.

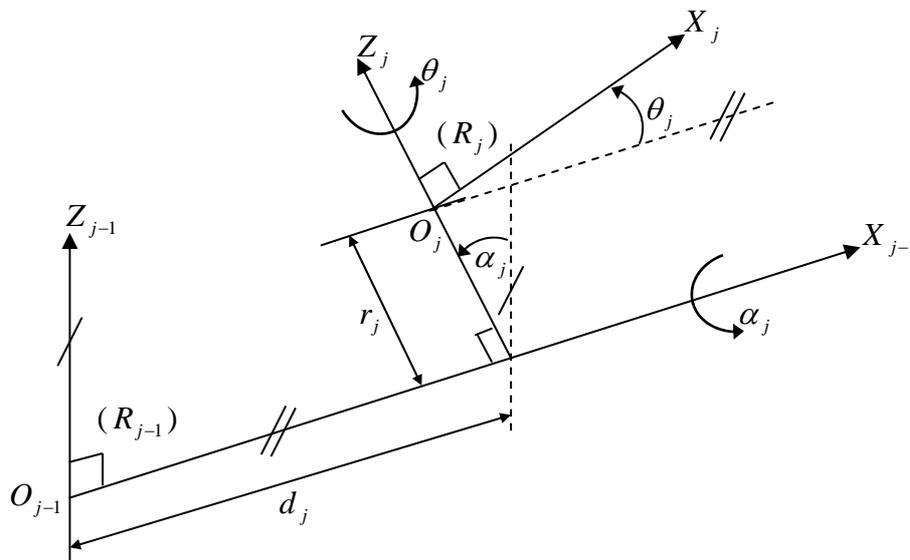


Figure I-4 : Paramètres de Denavit-Hartenberg modifiés.

La variable articulaire  $q_j$  associée à la  $j^{ième}$  articulation est définie par :

$$q_j = (1 - \sigma_j) \theta_j + \sigma_j r_j \tag{I-1}$$

où

$\sigma_j = 0$  si l'articulation est rotoïde.

$\sigma_j = 1$  si l'articulation est prismatique.

La matrice de transformation homogène définissant le repère  $R_j$  par rapport au repère antécédent  $R_{j-1}$  peut être obtenue ainsi :

$${}^{j-1}T_j = Rot(x, \alpha_j) Trans(x, d_j) Rot(z, \theta_j) Trans(z, r_j) \quad (I-2)$$

avec :

$$Rot(x, \alpha_j) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_j) & -\sin(\alpha_j) & 0 \\ 0 & \sin(\alpha_j) & \cos(\alpha_j) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (I-3)$$

$$Rot(z, \theta_j) = \begin{pmatrix} \cos(\theta_j) & -\sin(\theta_j) & 0 & 0 \\ \sin(\theta_j) & \cos(\theta_j) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (I-4)$$

$$Trans(x, d_j) = \begin{pmatrix} 1 & 0 & 0 & d_j \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (I-5)$$

$$Trans(z, r_j) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & r_j \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (I-6)$$

alors:

$${}^{j-1}T_j = \begin{pmatrix} \cos(\theta_j) & -\sin(\theta_j) & 0 & d_j \\ \cos(\alpha_j)\sin(\theta_j) & \cos(\alpha_j)\cos(\theta_j) & -\sin(\alpha_j) & -r_j\sin(\alpha_j) \\ \sin(\alpha_j)\sin(\theta_j) & \sin(\alpha_j)\cos(\theta_j) & \cos(\alpha_j) & r_j\cos(\alpha_j) \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (I-7)$$

La matrice homogène (I-7) peut être écrite sous la forme générale suivante :

$${}^{j-1}T_j = \begin{pmatrix} {}^{j-1}A_j & {}^{j-1}P_j \\ 0 & 1 \end{pmatrix} \quad (I-8)$$

où  ${}^{j-1}A_j$  est une matrice  $\in \mathfrak{R}^{3 \times 3}$  qui définit l'orientation de la base du repère  $R_j$  par rapport à celle du repère  $R_{j-1}$ , et  ${}^{j-1}P_j$  est une matrice  $\in \mathfrak{R}^3$  qui définit les coordonnées de l'origine  $O_j$  du repère  $R_j$  dans le repère  $R_{j-1}$ .

Le modèle géométrique direct du robot peut être exprimé par la relation :

$$X = f(q) \quad (I-9)$$

avec  $q = [q_1 \quad q_2 \quad \dots \quad q_n]^T$  est le vecteur des variables articulaires, et  $X = [x_1 \quad x_2 \quad \dots \quad x_m]^T$  est le vecteur des coordonnées opérationnelles (cartésiennes).

### I-2-2. Modèle géométrique inverse des robots manipulateurs à chaîne ouverte simple [9, 12]

Ce modèle consiste à calculer les coordonnées articulaires correspondant à une situation donnée de l'organe terminal. Il n'y a pas de solution générale pour obtenir directement ce modèle. Pour calculer ce modèle, plusieurs méthodes peuvent être utilisées. Parmi ces méthodes on peut citer :

- la méthode de Paul.
- La méthode de Pieper.
- La méthode générale de Raghavan et Roth.

Dans ce travail, nous développons la méthode de Paul. Cette méthode traite séparément chaque cas particulier et convient pour la plupart des robots industriels.

#### I-2-2-1. Position du problème

Soit  ${}^fT_{Ed}$  la matrice de transformation homogène définissant la situation désirée du repère outil  $R_E$  par rapport au repère atelier. En général, nous pouvons écrire  ${}^fT_{Ed}$  comme:

$${}^fT_{Ed} = Z {}^0T_n E \quad (\text{I-10})$$

où :

$Z = {}^fT_0$ , est la matrice de transformation homogène définissant la situation du robot (repère  $R_0$ ) dans le repère atelier.

$E = {}^nT_E$ , est la matrice de transformation homogène définissant le repère outil  $R_E$  dans le repère terminal  $R_n$ .

${}^0T_n$ , est la matrice de transformation homogène du repère terminal  $R_n$  dans le repère de base  $R_0$ , fonction du vecteur des variables articulaires  $q$ .

Ces différentes transformations entre repères sont représentées sur la figure I-5.

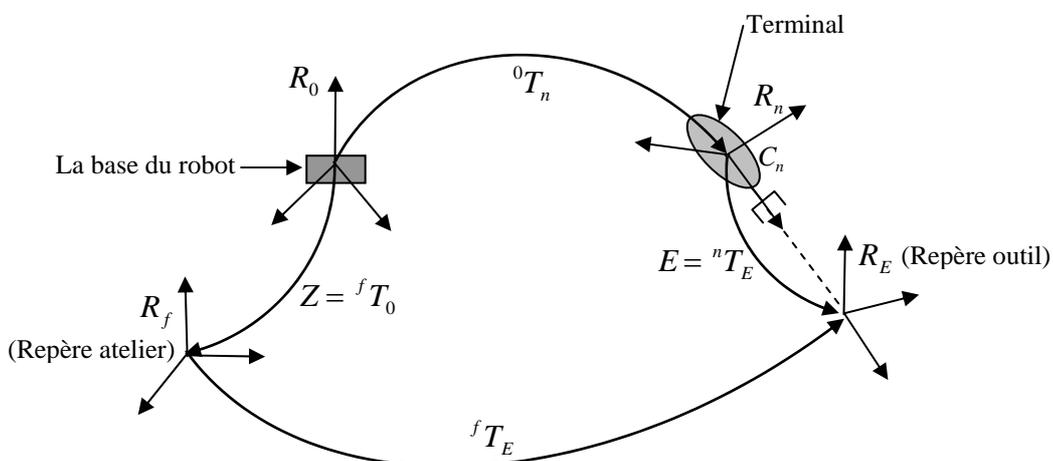


Figure I-5 : les différentes transformations entre repères du robot.

Notons  $U_0 = {}^0T_n$ , de l'équation (I-10) nous obtenons :

$$U_0 = Z^{-1} {}^fT_{Ed} E^{-1} \quad (\text{I-11})$$

Le problème est ainsi d'obtenir les variables articulaires tel que  ${}^0T_n$  soit égal à  $U_0$  (c à d : la résolution d'un système d'équation  $U_0 = {}^0T_1 {}^1T_2 \dots {}^{n-1}T_n$ ).

### I-2-2-2. Calcule du modèle inverse par la méthode de Paul

Considérons un robot manipulateur dont la matrice de transformation homogène a pour expression :

$${}^0T_n = {}^0T_1 {}^1T_2 \dots {}^{n-1}T_n \quad (\text{I-12})$$

On note:

$$U_0 = {}^0T_1 {}^1T_2 \dots {}^{n-1}T_n \quad (\text{I-13})$$

et:

$$U_0 = \begin{bmatrix} n_x & s_x & a_x & p_x \\ n_y & s_y & a_y & p_y \\ n_z & s_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = [n \quad s \quad a \quad p] \quad (\text{I-14})$$

$U_0$ , est une matrice connue : définit la situation désirée du repère lié à l'organe terminal. Le vecteur  $[n \quad s \quad a]$  définit l'orientation du repère  $R_n$  et  $[p]$  sa position.

La détermination des variables articulaires à partir de l'équation (I-13) en fonction des éléments  $n, s, a$  et  $p$  n'est pas aisée. Paul a proposé une méthode qui consiste à prémultiplier successivement les deux membres de l'équation (I-13) par les matrices  ${}^jT_{j-1}$  pour  $j=1, \dots, n-1$ , opérations qui permettent d'isoler et d'identifier l'une après l'autre les variables articulaires que l'on recherche.

Par exemple pour un robot à six degrés de liberté, nous procédons comme suit :

La multiplication des deux côtés de l'équation (I-13) par  ${}^1T_0$  donne :

$${}^1T_0 U_0 = {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 \quad (\text{I-15})$$

Le terme de droite est fonction des variables  $q_2, \dots, q_6$ , alors que le terme de gauche est fonction de la variable  $q_1$  et de  $U_0$ .

La variable  $q_1$  est obtenue par identification d'un ou de deux éléments des deux côtés de l'équation précédente.

Partant de l'équation (I-15), on prémultiplie par  ${}^2T_1$  et on réitère le même processus. La succession des équations permettant le calcul de tous les  $q_j$  est la suivante :

$$\begin{aligned}
U_0 &= {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 \\
U_1 &= {}^1T_2 {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 \\
U_2 &= {}^2T_3 {}^3T_4 {}^4T_5 {}^5T_6 \\
U_3 &= {}^3T_4 {}^4T_5 {}^5T_6 \\
U_4 &= {}^4T_5 {}^5T_6 \\
U_5 &= {}^5T_6
\end{aligned} \tag{I-16}$$

avec  $U_j = {}^jT_{j-1} U_{j-1}$

La solution des équations (I-16) demande de l'intuition, mais l'utilisation de la méthode sur un grand nombre de robots industriels a permis de constater que les principaux types rencontrés sont peu nombreux.

### I-3. Modèle cinématique directe des robots à structures ouvertes simples

Ce modèle exprime les vitesses cartésiennes (vitesses des coordonnées opérationnelles) en fonction des vitesses articulaires [7, 9, 10, 14]. Il est donné par la relation suivante :

$$\dot{X} = J(q) \dot{q} \tag{I-17}$$

où  $J(q) = \frac{\partial X}{\partial q} \in \mathfrak{R}^{m \times n}$  désigne la matrice jacobienne du mécanisme et  $X \in \mathfrak{R}^6$  est le vecteur des position et orientation (coordonnées opérationnelles) de l'effecteur.

Le calcul de cette matrice peut se faire en dérivant le Modèle Géométrique Direct ( $X = f(q)$ ), à partir de la relation suivante :

$$J_{ij} = \frac{\partial f_i(q)}{\partial q_j}, \quad i=1, \dots, m; \quad j=1, \dots, n \tag{I-18}$$

où  $J_{ij}$  est l'élément  $(i, j)$  de la matrice jacobienne  $J$ .

Cette méthode est facile à mettre en œuvre pour des robots à deux ou trois degrés de liberté. Pour les robots ayant plus de trois degrés de liberté, le calcul de la matrice jacobienne de base est plus pratique.

#### A. Matrice jacobienne de base

On peut obtenir la matrice jacobienne par une méthode de calcul direct, fondée sur la relation entre les vecteurs des vitesses de translation et de rotation  $v_n$  et  $\omega_n$  du repère  $R_n$ , représentant les éléments de réduction du torseur cinématique du repère  $R_n$ , et les vitesses articulaire  $\dot{q}$  :

$$V_n = \begin{bmatrix} v_n \\ \omega_n \end{bmatrix} = J_n \dot{q} \tag{I-19}$$

L'expression du jacobien est identique si l'on considère la relation entre les vecteurs de translation et de rotation différentielle  $(d_n, \delta_n)$  du repère  $R_n$  et les différentielles des coordonnées articulaires  $dq$  :

$$\begin{bmatrix} d_n \\ \delta_n \end{bmatrix} = J_n dq \tag{I-20}$$

Le jacobien correspondant à une représentation quelconque des vitesses opérationnelles  $\dot{X}$  peut être déduit de ce jacobien  $J_n$ , appelé jacobien de base.

**B. Calcul du jacobien de base**

Considérons la  $k^{ième}$  articulation d'une chaîne articulée. La vitesse  $\dot{q}_k$  induit sur le repère terminal  $R_n$  la vitesse de translation  $v_{k,n}$  et la vitesse de rotation  $\omega_{k,n}$ .

Dans ce cas, nous avons :

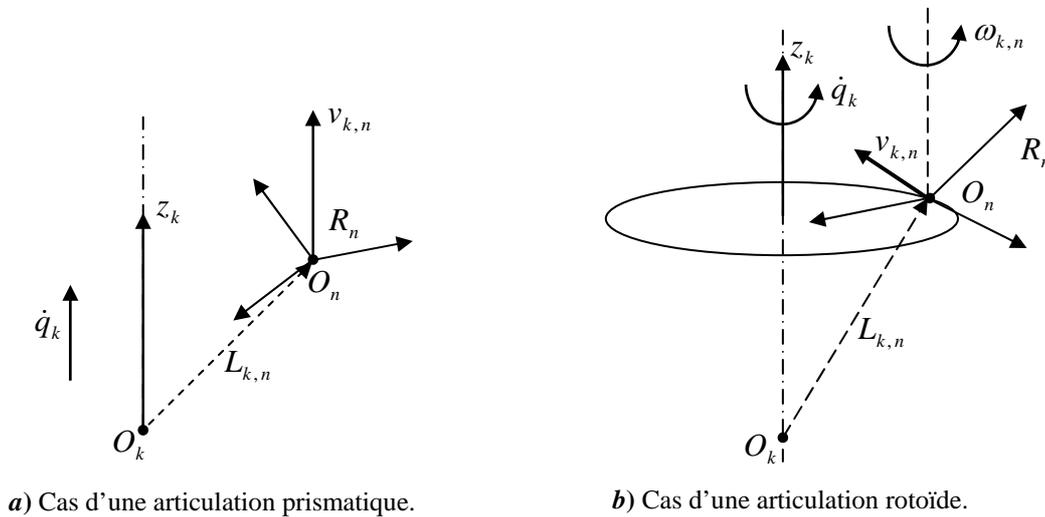
- si l'articulation est prismatique (figure I-6-a) :

$$\begin{cases} v_{k,n} = a_k \dot{q}_k \\ \omega_{k,n} = 0 \end{cases} \tag{I-21}$$

- si l'articulation est rotoïde (figure I-6-b) :

$$\begin{cases} v_{k,n} = a_k \dot{q}_k \times L_{k,n} = (a_k \times L_{k,n}) \dot{q}_k \\ \omega_{k,n} = a_k \dot{q}_k \end{cases} \tag{I-22}$$

où  $a_k$  est le vecteur unitaire porté par l'axe  $z_k$  de l'articulation  $k$  et le terme  $L_{k,n}$  désignant le vecteur d'origine  $O_k$  et d'extrémité  $O_n$ .



**Figure I-6 :** vitesse de la  $k^{ième}$  articulation dans le repère  $R_n$ .

Les vecteurs  $v_{k,n}$  et  $\omega_{k,n}$  s'écrivent donc sous la forme générale suivante :

$$\begin{cases} v_{k,n} = [\sigma_k a_k + \bar{\sigma}_k (a_k \times L_{k,n})] \dot{q}_k \\ \omega_{k,n} = \bar{\sigma}_k a_k \dot{q}_k \end{cases} \tag{I-23}$$

avec  $\sigma$  est une constante égale à 1 si l'articulation est prismatique et 0 si elle est rotoïde.

En appliquant le théorème de composition des vitesses, les vitesses de translation et de rotation du repère terminal s'écrivent :

$$\begin{cases} v_n = \sum_{k=1}^n v_{k,n} = \sum_{k=1}^n [\sigma_k a_k + \bar{\sigma}_k (a_k \times L_{k,n})] \dot{q}_k \\ \omega_n = \sum_{k=1}^n \omega_{k,n} = \sum_{k=1}^n \bar{\sigma}_k a_k \dot{q}_k \end{cases} \quad (\text{I-24})$$

En mettant ce système sous la forme matricielle et en l'identifiant à la relation (I-19), on déduit que :

$$J_n = \begin{bmatrix} \sigma_1 a_1 + \bar{\sigma}_1 (a_1 \times L_{1,n}) & \dots & \sigma_n a_n + \bar{\sigma}_n (a_n \times L_{n,n}) \\ \bar{\sigma}_1 a_1 & \dots & \bar{\sigma}_n a_n \end{bmatrix} \quad (\text{I-25})$$

Si l'on projette les éléments de la relation (I-25) dans un repère  $R_i$ , on obtient le jacobien  ${}^i J_n$  de dimension  $(6 \times n)$  tel que :

$${}^i V_n = {}^i J_n \dot{q} \quad (\text{I-26})$$

$v_n$  et  $\omega_n$  sont généralement exprimés soit dans le repère  $R_n$ , soit dans le repère  $R_0$ . La matrice jacobienne correspondante est notée  ${}^n J_n$  ou  ${}^0 J_n$  respectivement. Ces matrices peuvent aussi être calculées en utilisant une matrice  ${}^i J_n$ ,  $i=0, \dots, n$ , grâce à la relation de transformation de la matrice jacobienne entre repères suivante :

$${}^s J_n = \begin{bmatrix} {}^s A_i & 0_3 \\ 0_3 & {}^s A_i \end{bmatrix} {}^i J_n \quad (\text{I-27})$$

où  ${}^s A_i$  est la matrice d'orientation, de dimension  $(3 \times 3)$ , du repère  $R_i$  exprimée dans le repère  $R_s$ .

#### I-4. Modèle cinématique inverse des robots à structures ouvertes simples

Dans le cas régulier, la matrice jacobienne  $J$  est carrée d'ordre  $n$  et son déterminant est non nul, parmi les méthodes utilisées pour le calcul du modèle cinématique inverse nous trouvons celle basée sur l'inversion de la matrice  $J$  [4, 8, 9]. Son principe est donné comme suit :

Dans cette méthode, on calcule l'inverse de la matrice jacobienne  $J$  ( $J^{-1}$ ) qui permet d'écrire :

$$\dot{q} = J^{-1} \dot{X} \quad (\text{I-28})$$

Lorsque la matrice  $J$  a la forme suivante :

$$J = \begin{bmatrix} A & 0 \\ B & C \end{bmatrix} \quad (\text{I-29})$$

$A$  et  $C$  étant des matrices carrées inversibles, l'inverse de cette matrice s'écrit :

$$J^{-1} = \begin{bmatrix} A^{-1} & 0 \\ -C^{-1} B A^{-1} & C^{-1} \end{bmatrix} \quad (\text{I-30})$$

La résolution du problème se ramène donc à l'inversion de deux matrices de dimension moindre. Lorsque le robot manipulateur possède six degrés de liberté et un poignet rotule, la forme générale de  $J$  est celle de la relation (I-29),  $A$  et  $C$  étant de dimension  $(3 \times 3)$ .

### I-5. Modélisation dynamique d'un robot manipulateur

C'est un ensemble de formules mathématiques qui représentent les équations de mouvement d'un robot. Ces formules, sont un ensemble d'équations différentielles décrivant le comportement dynamique du robot manipulateur [1, 10, 13, 14].

Il existe deux types de modèles dynamiques :

#### \* Le modèle dynamique inverse

Ce modèle fournit les couples articulaires, exercés par les actionneurs, en fonction des positions, vitesses et accélérations articulaires. Il est donné par la relation suivante :

$$\Gamma = f(q, \dot{q}, \ddot{q}, F_e) \quad (\text{I-31})$$

où :

$\Gamma \in \mathcal{R}^n$  : vecteur des couples/forces appliqués par les actionneurs.

$q \in \mathcal{R}^n$  : vecteur des positions articulaires.

$\dot{q} \in \mathcal{R}^n$  : vecteur des vitesses articulaires.

$\ddot{q} \in \mathcal{R}^n$  : vecteur des accélérations articulaires.

$F_e$  : couples/forces extérieurs qu'exerce le robot sur l'environnement par l'organe terminal.

Le modèle dynamique inverse ou tout simplement le modèle dynamique est utilisé dans les applications de commande.

#### \* Le modèle dynamique direct

Ce modèle fournit les accélérations articulaires en fonction des positions, vitesses et couples articulaires. Il est donné par la relation suivante :

$$\ddot{q} = h(q, \dot{q}, \Gamma, F_e) \quad (\text{I-32})$$

Le modèle dynamique direct est utilisé en simulation.

Les formalismes les plus utilisés pour obtenir le modèle dynamique d'un robot sont :

- Le formalisme de Lagrange qui sera utilisé dans la suite de ce travail.
- Le formalisme de Newton-Euler.

#### I-5-1. Calcul du modèle dynamique par le formalisme de Lagrange

L'obtention de ce modèle par le formalisme de Lagrange-Euler se fait par la détermination de l'énergie cinétique  $K(q, \dot{q})$  et l'énergie potentielle  $U(q)$  du robot. Le Lagrangien exprime la différence entre ces deux énergies, il est donné comme suit :

$$L(q, \dot{q}) = K(q, \dot{q}) - U(q) \quad (\text{I-33})$$

En tenant compte de l'équation (I-33), les équations de mouvement du robot manipulateur sont données comme suit :

$$\frac{d}{dt} \left[ \frac{\partial L}{\partial \dot{q}} \right] - \frac{\partial L}{\partial q} = \Gamma \quad (\text{I-34})$$

où  $q$  et  $\dot{q} \in \mathfrak{R}^n$  sont respectivement les coordonnées et vitesses généralisées,  $\Gamma \in \mathfrak{R}^n$  vecteur des couples généralisés.

### I-5-1-1. Détermination de l'énergie cinétique

Soit le vecteur  ${}^i r$ , qui donne les coordonnées d'un point du corps  $i$ , par rapport au repère  $R_i$ , les coordonnées de ce point dans le repère de base  $R_0$ , s'écrivent :

$$r = {}^0 T_i \quad {}^i r \quad (\text{I-35})$$

Avec :  ${}^0 T_i = {}^0 T_1 \quad {}^1 T_2 \quad \dots \quad {}^{i-1} T_i \in \mathfrak{R}^{4 \times 4}$  est la matrice de transformation homogène qui définit le repère  $R_i$  dans le repère  $R_0$ , elle est fonction des variables articulaires  $q_1, q_2, \dots, q_i$ . Par conséquent, la vitesse du point considéré dans le repère  $R_0$  est donnée par l'expression suivante :

$$V = \frac{dr}{dt} = \sum_j^i \left[ \frac{\partial {}^0 T_i}{\partial q_j} \dot{q}_j \right] {}^i r \quad (\text{I-36})$$

Puisque  $\frac{\partial {}^0 T_i}{\partial q_j} = 0$  pour  $j > i$ , nous pouvons remplacer la limite supérieure de la sommation par "n" qui est le nombre des articulations.

L'énergie cinétique d'une masse infinitésimale  $dm$  à  ${}^i r$  ayant une vitesse  $V = [V_x \quad V_y \quad V_z]^T$  est :

$$\begin{aligned} dK_i &= \frac{1}{2} (V_x^2 + V_y^2 + V_z^2) dm \\ &= \frac{1}{2} \text{trace}(V^T V) dm \end{aligned} \quad (\text{I-37})$$

En utilisant l'expression de la vitesse "V" donnée par l'équation (I-36), on obtient :

$$dK_i = \frac{1}{2} \text{trace} \left[ \sum_{j=1}^n \sum_{k=1}^n \frac{\partial {}^0 T_i}{\partial q_j} ({}^i r \quad {}^i r^T dm) \frac{\partial {}^0 T_i^T}{\partial q_k} \dot{q}_j \quad \dot{q}_k \right] \quad (\text{I-38})$$

L'énergie cinétique totale du corps  $i$  est :

$$K_i = \int dK_i \quad (\text{I-39})$$

En substituant  $dK_i$  par l'expression (I-38), nous pouvons déplacer le symbole d'intégration à l'intérieur des sommations. Alors, la matrice de pseudo-inertie  $\in \mathfrak{R}^{n \times n}$  pour le corps  $i$  est donnée comme suit :

$$I_i = \int {}^i r {}^i r^T dm \quad (\text{I-40})$$

L'énergie cinétique du corps  $i$  peut s'écrire :

$$K_i = \frac{1}{2} \text{trace} \left[ \sum_{j=1}^n \sum_{k=1}^n \frac{\partial {}^0 T_i}{\partial q_j} I_i \frac{\partial {}^0 T_i}{\partial q_k} \dot{q}_j \dot{q}_k \right] \quad (\text{I-41})$$

Trouvons la matrice de pseudo-inertie avant de déterminer l'énergie cinétique totale. Soit  ${}^i r = [x \ y \ z \ 1]$ , les coordonnées de la masse infinitésimale  $dm$  dans le repère  $R_i$ . Alors, le développement de la relation (I-40) donne :

$$I_i = \begin{bmatrix} \int x^2 dm & \int yx dm & \int zx dm & \int x dm \\ \int xy dm & \int y^2 dm & \int zy dm & \int y dm \\ \int xz dm & \int yz dm & \int z^2 dm & \int z dm \\ \int x dm & \int y dm & \int z dm & \int dm \end{bmatrix} \quad (\text{I-42})$$

où les intégrales sont prises sur le volume du corps  $i$ . C'est une matrice constante qui est évaluée une fois pour chaque corps. Elle dépend de la géométrie et la distribution de la masse du corps  $i$ . En effet, elle est exprimée en termes des moments d'inertie du corps  $i$ .

$$\begin{aligned} I_{xx} &= \int (y^2 + z^2) dm \\ I_{yy} &= \int (x^2 + z^2) dm \\ I_{zz} &= \int (x^2 + y^2) dm \end{aligned} \quad (\text{I-43})$$

produits croisés de l'inertie :

$$\begin{aligned} I_{xy} &= \int xy dm \\ I_{xz} &= \int xz dm \\ I_{yz} &= \int yz dm \end{aligned} \quad (\text{I-44})$$

et des moments premiers :

$$\begin{aligned} m \bar{x} &= \int x dm \\ m \bar{y} &= \int y dm \\ m \bar{z} &= \int z dm \end{aligned} \quad (\text{I-45})$$

avec la masse totale  $m$  du corps  $i$ , et  ${}^i \bar{r} = [\bar{x} \ \bar{y} \ \bar{z} \ 1]^T$  représente le vecteur des coordonnées de centre de gravité du corps  $i$  dans le repère  $R_i$ , nous pouvons écrire :

$$I_i = \begin{bmatrix} \frac{-I_{xx} + I_{yy} + I_{zz}}{2} & I_{xy} & I_{xz} & m\bar{x} \\ I_{xy} & \frac{I_{xx} - I_{yy} + I_{zz}}{2} & I_{yz} & m\bar{y} \\ I_{xz} & I_{yz} & \frac{I_{xx} + I_{yy} - I_{zz}}{2} & m\bar{z} \\ m\bar{x} & m\bar{y} & m\bar{z} & m \end{bmatrix} \quad (\text{I-46})$$

Ces grandeurs sont soit données par les spécifications du constructeur, soit elles peuvent être calculées à partir des grandeurs trouvées dans ces spécifications.

L'énergie cinétique totale du bras manipulateur est :

$$K = \sum_{i=1}^n K_i = \frac{1}{2} \sum_{i=1}^n \text{trace} \left[ \sum_{j=1}^n \sum_{k=1}^n \frac{\partial^0 T_i}{\partial q_j} I_i \frac{\partial^0 T_i^T}{\partial q_k} \dot{q}_j \dot{q}_k \right] \quad (\text{I-47})$$

Puisque la trace d'une somme des matrices est la somme des traces individuelles, nous pouvons interchanger les sommations et l'opérateur de la *trace* pour obtenir :

$$K = \frac{1}{2} \sum_{j=1}^n \sum_{k=1}^n m_{jk}(q) \dot{q}_j \dot{q}_k \quad (\text{I-48})$$

ou :

$$K = \frac{1}{2} \dot{q}^T M(q) \dot{q} \quad (\text{I-49})$$

avec  $M(q) \in \mathfrak{R}^{n \times n}$  est la matrice d'inertie du bras, elle contient les éléments génériques définis comme suit :

$$m_{jk}(q) = \sum_{i=1}^n \text{trace} \left[ \frac{\partial^0 T_i}{\partial q_j} I_i \frac{\partial^0 T_i^T}{\partial q_k} \dot{q}_j \dot{q}_k \right] \quad (\text{I-50})$$

et  $\dot{q} \in \mathfrak{R}^n$  est le vecteur de vitesse articulaire.

Puisque  $\frac{\partial^0 T_i}{\partial q_j} = 0$  pour  $j > i$ , nous pouvons écrire :

$$m_{jk}(q) = \sum_{i=\max(j,k)}^n \text{trace} \left[ \frac{\partial^0 T_i}{\partial q_j} I_i \frac{\partial^0 T_i^T}{\partial q_k} \dot{q}_j \dot{q}_k \right] \quad (\text{I-51})$$

L'équation (I-49) fournit une expression de l'énergie cinétique du bras en fonction des grandeurs connues et des variables articulaires  $q_i$ . Puisque  $m_{jk} = m_{kj}$ , la matrice d'inertie  $M(q)$  est symétrique. L'énergie cinétique est toujours positive, s'annulant seulement lorsque les vitesses généralisées  $\dot{q}$  deviennent nulles, donc la matrice d'inertie  $M(q)$  est aussi définie positive.

**I-5-1-2. L'énergie potentielle**

Si le corps  $i$  a une masse  $m_i$  et un centre de gravité  ${}^i\bar{r}$  exprimé dans les coordonnées du repère  $R_i$ , l'énergie potentielle du corps est donnée par :

$$U_i = -m_i g^T {}^0T_i {}^i\bar{r} \quad (\text{I-52})$$

Où  $g^T$  est le vecteur de gravité exprimé avec les coordonnées de repère de base  $R_0$  ainsi :

$$g^T = [g_x \quad g_y \quad g_z \quad 1] \quad (\text{I-53})$$

L'énergie potentielle totale du bras est alors :

$$U = \sum_{i=1}^n U_i = - \sum_{i=1}^n m_i g^T {}^0T_i {}^i\bar{r} \quad (\text{I-54})$$

Puisque seule  ${}^0T_i$  est fonction de  $q$  alors  $U$  est fonction seulement de  $q$ .

**I-5-1-3. Formulation du modèle de mouvement du robot**

En utilisant les expressions (I-49) et (I-54), le formalisme de Lagrange s'écrit :

$$\frac{d}{dt} \frac{\partial K(q, \dot{q})}{\partial \dot{q}} - \frac{\partial K(q, \dot{q})}{\partial q_k} + \frac{\partial U(q)}{\partial q_k} = \Gamma_k \quad (\text{I-55})$$

avec

$$\begin{aligned} \frac{d}{dt} \frac{\partial K}{\partial \dot{q}} &= \frac{d}{dt} \left[ \frac{1}{2} \sum_i^n m_{ik}(q) \dot{q}_i + \frac{1}{2} \sum_j^n m_{kj}(q) \dot{q}_j \right] \\ &= \frac{d}{dt} \left[ \sum_j^n m_{kj}(q) \dot{q}_j \right] = \sum_j^n m_{kj}(q) \ddot{q}_j + \sum_{i,j}^n \frac{dm_{kj}(q)}{dt} \dot{q}_j \\ &= \sum_j^n m_{kj}(q) \ddot{q}_j + \sum_{i,j}^n \frac{\partial m_{kj}(q)}{\partial q_i} \dot{q}_i \dot{q}_j \end{aligned} \quad (\text{I-56})$$

$$\frac{\partial K}{\partial q_k} = \frac{1}{2} \sum_{i,j}^n \frac{\partial m_{ij}(q)}{\partial q_k} \dot{q}_i \dot{q}_j \quad (\text{I-57})$$

donc le formalisme de Lagrange peut s'écrire :

$$\sum_j^n m_{kj}(q) \ddot{q}_j + \sum_{i,j}^n \left[ \frac{\partial m_{kj}(q)}{\partial q_i} - \frac{1}{2} \sum_{i,j}^n \frac{\partial m_{ij}(q)}{\partial q_k} \right] \dot{q}_i \dot{q}_j + \frac{\partial U}{\partial q_k} = \Gamma_k, \quad \text{avec } k=1, \dots, n \quad (\text{I-58})$$

En interchangeant les ordres de sommation et profitant de l'avantage de symétrie de  $M(q)$ , nous pouvons démontrer que :

$$\sum_{i,j}^n \frac{\partial m_{kj}(q)}{\partial q_i} \dot{q}_i \dot{q}_j = \frac{1}{2} \sum_{i,j}^n \left[ \frac{\partial m_{kj}(q)}{\partial q_i} + \frac{\partial m_{ki}(q)}{\partial q_j} \right] \dot{q}_i \dot{q}_j \quad (\text{I-59})$$

Donc :

$$\sum_{i,j}^n \left[ \frac{\partial m_{kj}(q)}{\partial q_i} - \frac{1}{2} \frac{\partial m_{ij}(q)}{\partial q_k} \right] \dot{q}_i \dot{q}_j = \sum_{i,j}^n \frac{1}{2} \left[ \frac{\partial m_{kj}(q)}{\partial q_j} + \frac{\partial m_{ki}(q)}{\partial q_j} - \frac{\partial m_{ij}(q)}{\partial q_k} \right] \dot{q}_i \dot{q}_j \quad (\text{I-60})$$

Les termes :

$$C_{ij,k} = \frac{1}{2} \left[ \frac{\partial m_{kj}(q)}{\partial q_i} + \frac{\partial m_{ki}(q)}{\partial q_j} - \frac{\partial m_{ij}(q)}{\partial q_k} \right] \quad (\text{I-61})$$

Sont appelés symboles de Christoffel. Nous remarquons que, pour un  $k$  donné, nous avons  $C_{ij,k} = C_{ji,k}$ . Finalement, nous avons les termes :

$$g_k = \frac{\partial U}{\partial q_k} \quad (\text{I-62})$$

Alors nous pouvons écrire le formalisme de Lagrange ainsi :

$$\sum_j^n m_{kj}(q) \ddot{q}_j + \sum_{i,j}^n C_{ij,k}(q) \dot{q}_i \dot{q}_j + g_k(q) = \Gamma_k \quad (\text{I-63})$$

Dans l'équation (I-63), il y a trois types de termes. Les premiers comportent la dérivée seconde des coordonnées généralisées. Les seconds peuvent être classifiés en deux types, des termes concernant un produit du type  $q_i^2$  sont appelés centrifuges, par contre, les termes concernant un produit de type  $q_i q_j$ , où  $i \neq j$ , sont appelés des termes de Coriolis. Le troisième type de termes est fonction seulement de  $q$  qui est issu de la dérivation de l'énergie potentielle et qui est appelé vecteur des forces de gravité. Il est habituel d'écrire (I-63) sous une forme matricielle suivante :

$$\Gamma = M(q) \ddot{q} + C(q, \dot{q}) \dot{q} + G(q) \quad (\text{I-64})$$

Avec ;

$\Gamma \in \mathfrak{R}^n$  : vecteur des couples appliqués aux actionneurs.

$q, \dot{q}, \ddot{q} \in \mathfrak{R}^n$  : vecteurs des positions, des vitesses et d'accélération articulaires.

$M(q) \in \mathfrak{R}^{n \times n}$  : matrice d'inertie.

$G(q) \in \mathfrak{R}^n$  : vecteur de force de gravité.

$C(q, \dot{q}) \in \mathfrak{R}^{n \times n}$  : matrice des forces centrifuges et de Coriolis dont les éléments sont :

$$\begin{aligned} C_{kj} &= \sum_i^n C_{ijk}(q) \dot{q}_i \\ &= \sum_i^n \frac{1}{2} \left[ \frac{\partial m_{kj}(q)}{\partial q_i} + \frac{\partial m_{ki}(q)}{\partial q_j} - \frac{\partial m_{ij}(q)}{\partial q_k} \right] \dot{q}_i \end{aligned} \quad (\text{I-65})$$

Si en tenant compte des frottements qui sont difficiles à modéliser. Pourtant, ils sont loin d'être négligeables sur les robots réels et aussi les efforts extérieurs exercés par et sur l'organe terminal dans et par l'environnement ; donc le modèle (I-64) peut être généralisé sous la forme suivante :

$$\Gamma = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) + \Gamma_d \quad (\text{I-66})$$

où :

$$F(\dot{q}) = F_v \dot{q} + F_s(\dot{q}) \quad (\text{I-67})$$

Avec  $F_v \in \mathfrak{R}^{n \times n}$  est la matrice de coefficients des frottements visqueux et  $F_s \in \mathfrak{R}^n$  un terme de frottements secs, ainsi qu'un terme  $\Gamma_d \in \mathfrak{R}^n$  de perturbations ou bien des couples résistants.

### I-5-2. Propriétés structurelles du modèle dynamique

La connaissance de certaines propriétés des différents termes du modèle (I-66) conduit à une analyse efficace du comportement du système et une conception plus facile des lois de commande.

#### I-5-2-1. Propriétés de la matrice d'inertie

La matrice  $M(q)$  est symétrique et définie positive ( $M(q) = M(q)^T > 0$ ). En effet, l'énergie cinétique du bras est :

$$K = \frac{1}{2} \dot{q}^T M(q) \dot{q} \quad (\text{I-68})$$

Une autre propriété importante de  $M(q)$  est qu'elle est bornée comme suit :

$$\mu_1 I_n \leq M(q) \leq \mu_2 I_n \quad (\text{I-69})$$

où  $\mu_1$  et  $\mu_2$  sont des scalaires positifs minorant et majorant la matrice d'inertie qui peuvent être calculés pour chaque bras donné et  $I_n$  matrice identité d'ordre  $n$ . En plus, la matrice d'inertie inverse est bornée :

$$\frac{1}{\mu_2} I_n \leq M^{-1}(q) \leq \frac{1}{\mu_1} I_n \quad (\text{I-70})$$

Si les articulations sont rotoïdes, les bornes  $\mu_1$  et  $\mu_2$  sont constants, puisque  $q$  apparaît dans  $M(q)$  seulement à travers des termes des sinus ou cosinus, qui sont bornés par 1. Par contre, si le bras a des articulations prismatiques, alors  $\mu_1$  et  $\mu_2$  peuvent être des fonctions scalaires de  $q$ .

#### I-5-2-2. Propriétés de la matrice des forces centrifuges et de Coriolis

La matrice des forces centrifuges et de Coriolis est caractérisée par trois propriétés et qui sont:

1. la matrice définie par :

$$N(q, \dot{q}) = \dot{M}(q) - 2C(q, \dot{q}) \quad (\text{I-71})$$

est antisymétrique ( $\forall q \in \mathfrak{R}^n : q^T N(q, \dot{q}) q = 0$ ) ce qui veut dire que les termes de  $N(q, \dot{q})$  vérifient l'égalité  $n_{jk} = -n_{kj}$ .

En effet :

$$\begin{aligned} n_{ij} &= \dot{m}_{kj}(q) - 2C_{kji}(q, \dot{q}) \dot{q}_k \\ &= \frac{\partial m_{ij}(q)}{\partial q_k} \dot{q}_k - 2C_{kji}(q) \dot{q}_k \end{aligned} \quad (\text{I-72})$$

En remplaçant l'expression de  $C_{k,j,i}(q)$  donnée par l'équation (I-61), nous obtenons :

$$\begin{aligned} n_{ij} &= \left[ \frac{\partial m_{ij}(q)}{\partial q_k} - \frac{\partial m_{ij}(q)}{\partial q_k} + \frac{\partial m_{ki}(q)}{\partial q_j} + \frac{\partial m_{jk}(q)}{\partial q_i} \right] \dot{q}_k \\ &= \left[ \frac{\partial m_{ki}(q)}{\partial q_j} + \frac{\partial m_{jk}(q)}{\partial q_i} \right] \dot{q}_k \end{aligned} \quad (\text{I-73})$$

En profitant du fait que  $M(q)$  est symétrique, nous pouvons écrire aussi :

$$\begin{aligned} n_{ij} &= \left[ \frac{\partial m_{ki}(q)}{\partial q_j} + \frac{\partial m_{jk}(q)}{\partial q_i} \right] \dot{q}_k \\ &= - \left[ \frac{\partial m_{ik}(q)}{\partial q_j} + \frac{\partial m_{kj}(q)}{\partial q_i} \right] \dot{q}_k \\ &= -n_{ji} \end{aligned} \quad (\text{I-74})$$

2. la matrice  $C(q, \dot{q}) \dot{q}$  vérifie la relation suivante :

$$\forall x, y \in \mathfrak{R}^n; \quad C(q, x) y = C(q, y) x \quad (\text{I-75})$$

avec  $x, y$  deux vecteurs de vitesse.

En effet, la  $i^{\text{ième}}$  coordonnée du vecteur  $C(q, x) y$  s'écrit :

$$C_{ij}(q, x) y_j = C_{kji}(q) x_k y_i = C_{jk,i}(q) x_k y_i \quad (\text{I-76})$$

alors

$$C_{kji}(q) y_k x_i = C_{ji}(q, y) x_i \quad (\text{I-77})$$

Puisque les symboles de Christoffel sont symétriques par rapport aux deux premiers indices.

3. la norme de  $C(q, \dot{q})$  vérifie la relation :  $\|C(q, x)\| \leq \mu \|x\| \quad \forall x \in \mathfrak{R}^n$

En effet, à partir de l'équation (I-61), nous pouvons écrire :

$$\begin{aligned} C_{ij}(q, x) &= \frac{1}{2} \left[ \frac{\partial m_{ij}(q)}{\partial q_k} + \frac{\partial m_{ki}(q)}{\partial q_j} - \frac{\partial m_{jk}(q)}{\partial q_i} \right] x_k \\ &= C_{kji}(q) x_k \end{aligned} \quad (\text{I-78})$$

Dans le cas d'un robot manipulateur à  $n$  degré de liberté constitué seulement par des articulations rotoïdes, nous avons :

$$\max_{i,j,k} \sup_{q \in \mathfrak{R}^n} \left[ \frac{\partial m_{ij}(q)}{\partial q_k} + \frac{\partial m_{ki}(q)}{\partial q_j} - \frac{\partial m_{jk}(q)}{\partial q_i} \right] \leq \alpha \quad (\text{I-79})$$

ce qui donne :

$$\|C_{ij}(q, x)\| \leq \frac{n\alpha}{2} \quad (\text{I-80})$$

dans le cas des articulations prismatiques,  $\alpha$  est fonction scalaire de  $q$ .

**I-5-2-3. Propriété du vecteur des forces de gravité**

La norme du vecteur de gravité est bornée supérieurement par :

$$\|G(q)\| \leq g_m(q) \quad (\text{I-81})$$

où  $g_m$  est une fonction scalaire, constante dans le cas d'une articulation rotoïde, mais dans le cas d'une articulation prismatique  $g_m$  dépend de la variable  $q$ .  $\|G(q)\|$  est bornée parce que la variable articulaire apparaît dans  $G(q)$  seulement à travers des fonctions sinus et cosinus.

**I-5-2-4. Propriété du vecteur de frottement**

La relation (I-67) qui exprime le vecteur de frottements est composée de frottements visqueux et de frottements secs. Supposant que le frottement sur chaque articulation dépend uniquement de la vitesse de l'articulation considérée, nous avons alors pour  $i=1, \dots, n$  :

$$\begin{aligned} F_v &= \text{diag}(\dot{q}_i) \\ F_s &= [k_1 \text{sgn}(\dot{q}_1) \dots k_i \text{sgn}(\dot{q}_i) \dots k_n \text{sgn}(\dot{q}_n)] \end{aligned} \quad (\text{I-82})$$

une majoration des vecteurs de frottements peut être considérée comme suit :

$$\|F_v \dot{q} + F_s(\dot{q})\| \leq F_1 + F_2 \|\dot{q}\| \quad (\text{I-83})$$

avec  $F_1$  et  $F_2$  sont des termes connus pour chaque bras et  $\|\cdot\|$  est une norme vectorielle appropriée.

**I-5-2-5. Propriété du vecteur des perturbations**

Le vecteur  $\Gamma_d$  dus aux perturbations et aux incertitudes de modélisation dynamique est borné par :

$$\|\Gamma_d\| \leq d \quad (\text{I-84})$$

où  $d$  est une constante scalaire qui peut être calculée pour chaque articulation du robot manipulateur et  $\|\cdot\|$  est une norme vectorielle appropriée.

**I-5-2-6. Paramétrisation linéaire**

Le modèle dynamique (I-66) est linéaire par rapport aux paramètres, une propriété exploitée la première fois par Craig J. J. dans la commande adaptative [3]. Cela est important, car la plupart ou tous les paramètres de l'équation dynamique du robot peuvent être inconnus, donc la dynamique du robot est linéaire par rapport à ces paramètres. Cette propriété peut être exprimée comme suit :

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) + F(\dot{q}) + \Gamma_d = Y(q, \dot{q}, \ddot{q})\theta \quad (\text{I-85})$$

avec  $\theta$  est un vecteur qui contient tous les paramètres du robot et  $Y(q, \dot{q}, \ddot{q})$  est une matrice des fonctions dépendantes des variables articulaires, elle est appelée regresser. Cette matrice peut être calculée pour chaque robot manipulateur donné, donc elle est connue.

### I-6. Exemple de modélisation d'un bras manipulateur Puma 600

Le robot que nous avons utilisé dans notre travail est un robot UNIMATION PUMA 600. C'est un robot à six axes. Les robots de la gamme PUMA sont probablement les robots les plus répandus dans les universités et les robots d'assemblage les plus utilisés dans l'industrie [21].

Le PUMA (Programmable Universal Machine for Assembly) a été originalement conçu par Vic Schienman et financé par General Motors et The Massachusetts Institute of Technology au milieu des années 70, et fut produit pendant de nombreuses années par Unimation (société qui fut rachetée plus tard par Westinghouse, avant d'être revendue à Staubli, une grande société Suisse de robotique).

Afin de procéder au calcul des différents modèles du robot Puma 600 et pour des raisons de simplification de travail, nous avons choisi ses trois premiers degrés de liberté. Puisque ceux-ci suffisent pour permettre de positionner l'organe terminal en un point quelconque de l'espace atteignable du robot.

#### I-6-1. Modèle géométrique direct du robot Puma 600

Le robot Puma 600 utilisé, est illustré à la figure I-7. Les référentiels représentés dans cette figure respectent les règles énoncées plus haut. En appliquant les notations de Denavit-Hartenberg (D-H), on obtient le tableau I-1 [15].

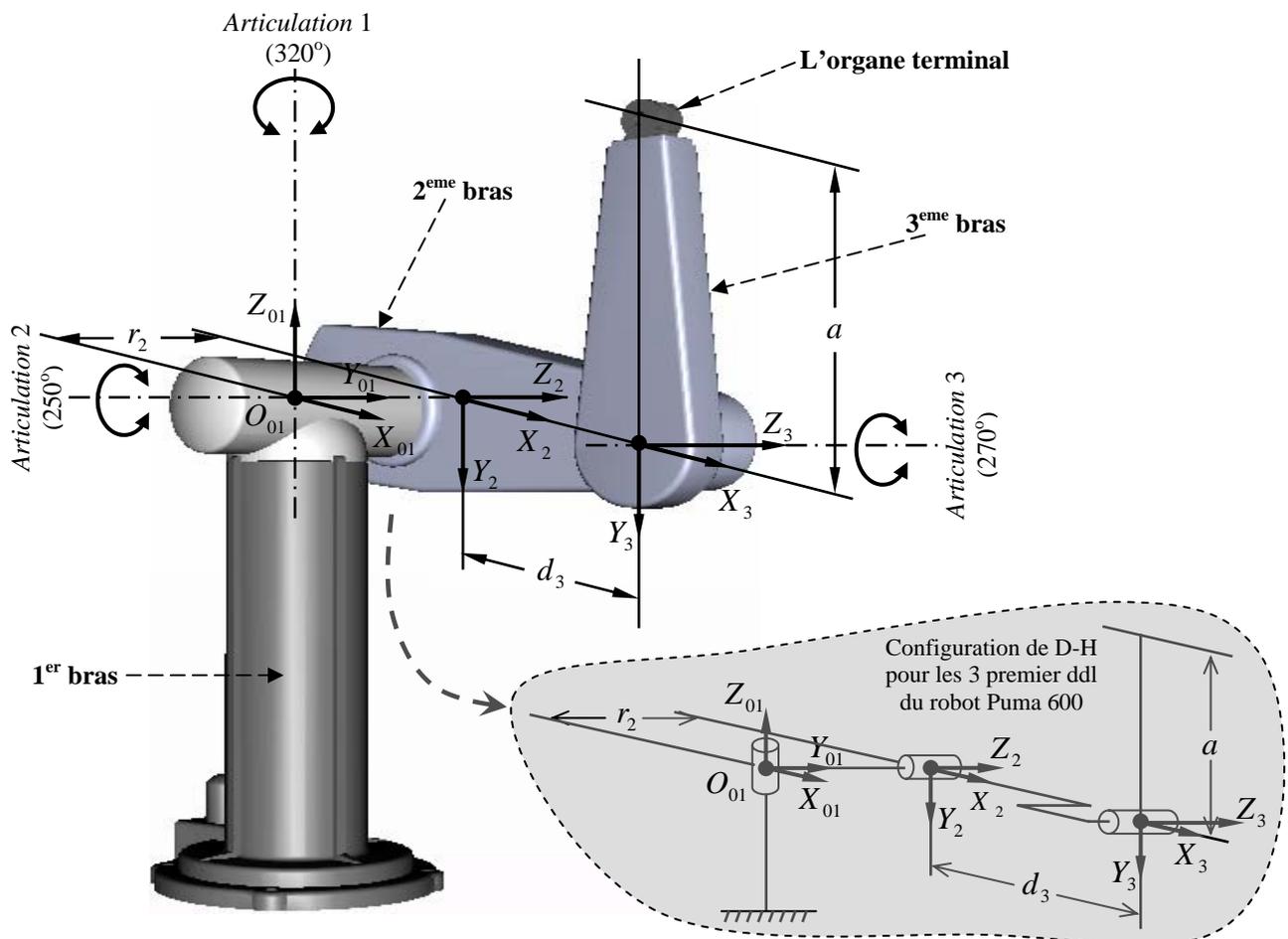


Figure I-7 : Schéma de principe du robot Puma 600 avec ses paramètres de Denavit-Hartenberg (D-H)

Articulation $i$	$\alpha$ (Degré)	$d$ (Mètre)	$r$ (Mètre)	$\theta_{initiale}$ (Degré)	$\theta$ (Degré)
1	0	0	0	0	$\theta_1$
2	-90	0	0.149	0	$\theta_2$
3	0	0.432	0	0	$\theta_3$

**Tableau I-1:** Paramètres de Denavit-Hartenberg modifiée du robot Puma 600

On a donc les matrices de transformations (ou de passages du repère  $R_0$  au repère  $R_3$ ) suivantes :

$${}^0T_1 = \begin{bmatrix} \cos(q_1) & -\sin(q_1) & 0 & 0 \\ \sin(q_1) & \cos(q_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{I-86})$$

$${}^1T_2 = \begin{bmatrix} \cos(q_2) & -\sin(q_2) & 0 & 0 \\ 0 & 0 & 1 & -r_2 \\ -\sin(q_2) & -\cos(q_2) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{I-87})$$

$${}^2T_3 = \begin{bmatrix} \cos(q_3) & -\sin(q_3) & 0 & d_3 \\ \sin(q_3) & \cos(q_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{I-88})$$

En utilisant les relations  ${}^0T_2 = {}^0T_1 {}^1T_2$  et  ${}^0T_3 = {}^0T_1 {}^1T_2 {}^2T_3$ , nous pouvons calculer, après multiplications, les matrices de passages suivantes:

$${}^0T_2 = \begin{bmatrix} \cos(q_1)\cos(q_2) & -\cos(q_1)\sin(q_2) & -\sin(q_1) & \sin(q_1)r_2 \\ \sin(q_1)\cos(q_2) & -\sin(q_1)\sin(q_2) & \cos(q_1) & -\cos(q_1)r_2 \\ -\sin(q_2) & -\cos(q_2) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{I-89})$$

$${}^0T_3 = [T_1 \quad T_2 \quad T_3 \quad T_4] \quad (\text{I-90})$$

Où :

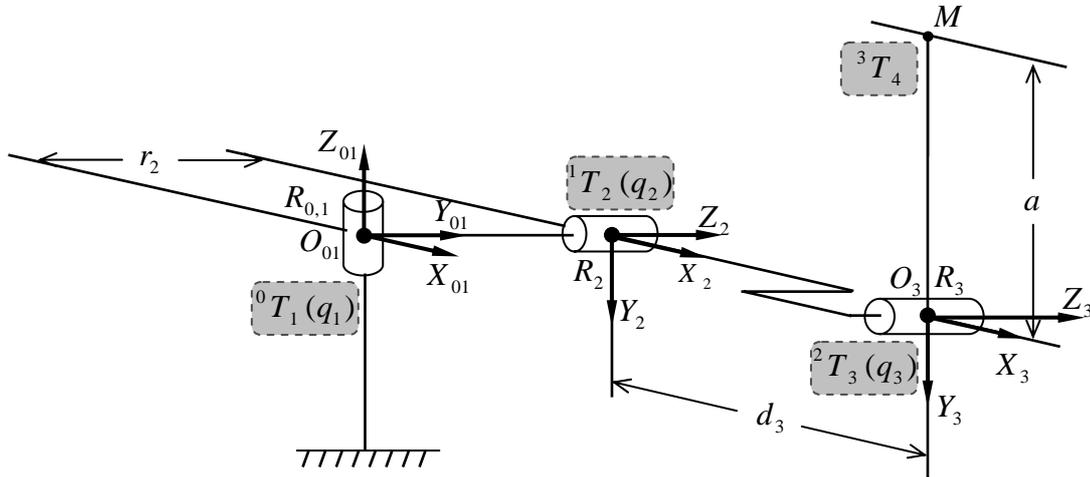
$$T_1 = \begin{bmatrix} \frac{1}{2}\cos(-q_3 + q_1 - q_2) + \frac{1}{2}\cos(q_3 + q_1 + q_2) \\ \frac{1}{2}\sin(q_3 + q_1 + q_2) + \frac{1}{2}\sin(-q_3 + q_1 - q_2) \\ -\sin(q_2 + q_3) \\ 0 \end{bmatrix}$$

$$T_2 = \begin{bmatrix} \frac{1}{2} \sin(-q_3 + q_1 - q_2) - \frac{1}{2} \sin(q_3 + q_1 + q_2) \\ \frac{1}{2} \cos(q_3 + q_1 + q_2) - \frac{1}{2} \cos(-q_3 + q_1 - q_2) \\ -\cos(q_2 + q_3) \\ 0 \end{bmatrix}$$

$$T_3 = \begin{bmatrix} -\sin(q_1) \\ \cos(q_1) \\ 0 \\ 0 \end{bmatrix} \quad \text{et} \quad T_4 = \begin{bmatrix} \frac{1}{2} d_3 \cos(q_1 - q_2) + \frac{1}{2} d_3 \cos(q_1 + q_2) + \sin(q_1) r_2 \\ \frac{1}{2} d_3 \sin(q_1 + q_2) + \frac{1}{2} d_3 \sin(q_1 - q_2) - \cos(q_1) r_2 \\ -\sin(q_2) d_3 \\ 1 \end{bmatrix}$$

**I-6-2. Modèle géométrique inverse du robot Puma 600**

Afin de calculer le modèle géométrique inverse (MGI) du robot Puma 600, nous reprenons la configuration de Denavit-Hartenberg (D-H) suivante (figure I-8) :



**Figure I-8 :** Représentation détaillée des paramètres de D-H du robot Puma 600

La matrice  ${}^0T_3$ , calculée précédemment (équation (I-90)), définit les coordonnées du point  $O_3$  du centre de repère  $R_3$  par rapport au repère  $R_0$ . Les coordonnées homogènes, du point  $M$  du centre de l'organe terminal, exprimées dans le repère  $R_3$  sont données par le vecteur :

$${}^3D_4 = [0 \quad -a \quad 0 \quad 1]^T \tag{I-91}$$

La matrice  ${}^3T_4$  sur la figure I-8, définit la matrice de translation suivant l'axe ( $y$ ) tel que :

$${}^3T_4 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & -a \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{I-92}$$

La situation de l'organe terminal (le point  $M$ ) dans le repère  $R_0$  est définie par :

$${}^0T_4 = {}^0T_3 {}^3T_4 = U_0 \quad (\text{I-93})$$

A partir de l'équation (I-93) nous pouvons déduire, après calcul, que la quatrième colonne du produit des transformations  ${}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4$  est égale à la quatrième colonne de  $U_0$  (voir l'équation (I-14)), soit :

$$\begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = {}^0T_1 {}^1T_2 {}^2T_3 {}^3T_4 \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = {}^0T_1 {}^1T_2 {}^2T_3 {}^3D_4 = {}^0T_3 {}^3D_4 \quad (\text{I-94})$$

Après développement de l'équation (I-94), nous arrivons à un système de 4 équations de position suivant :

$$\begin{bmatrix} p_x \\ p_y \\ p_z \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(q_1) [a \sin(q_2 + q_3) + d_3 \cos(q_2)] + r_2 \sin(q_1) \\ \sin(q_1) [a \sin(q_2 + q_3) + d_3 \cos(q_2)] - r_2 \cos(q_1) \\ a \cos(q_2 + q_3) - d_3 \sin(q_2) \\ 1 \end{bmatrix} \quad (\text{I-95})$$

A partir de l'équation (I-95) et par l'application de la méthode de Paul (§ I-2-2-2), le calcul des positions articulaires ( $q_1 \quad q_2 \quad q_3$ ) en fonction des positions cartésiennes se fait comme suit :

En multipliant, par la matrice  ${}^1T_0$ , les deux côtés du système (I-94), on obtient :

$${}^1T_0 \times [p_x \quad p_y \quad p_z \quad 1]^T = {}^1T_2 {}^2T_3 {}^3D_4 \quad (\text{I-96})$$

Après développement, il en résulte les équations suivantes :

$$\begin{bmatrix} p_x \cos(q_1) + p_y \sin(q_1) \\ -p_x \sin(q_1) + p_y \cos(q_1) \\ p_z \\ 1 \end{bmatrix} = \begin{bmatrix} a \sin(q_2 + q_3) + d_3 \cos(q_2) \\ -r_2 \\ a \cos(q_2 + q_3) - d_3 \sin(q_2) \\ 1 \end{bmatrix} \quad (\text{I-97})$$

La deuxième équation, du système (I-97), permet de déduire la valeur  $q_1$ . En effet, on a :

$$Y \cos(q_1) + X \sin(q_1) = Z, \text{ avec : } Y = p_y ; X = -p_x ; Z = -r_2$$

La résolution de l'équation précédente, conduit à l'obtention de la valeur  $q_1$  comme suit :

$$\begin{cases} \sin(q_1) = \frac{XZ + \varepsilon Y \sqrt{X^2 + Y^2 - Z^2}}{X^2 + Y^2} \\ \cos(q_1) = \frac{YZ - \varepsilon X \sqrt{X^2 + Y^2 - Z^2}}{X^2 + Y^2} \end{cases} ; \text{ avec : } \varepsilon = \pm 1 \quad (\text{I-98})$$

L'équation (I-98), montre que cette solution n'existe que si  $X^2 + Y^2 \geq Z^2$ . On en déduit que :

$$q_1 = \arctg \left( \frac{XZ + \varepsilon Y \sqrt{X^2 + Y^2 - Z^2}}{YZ - \varepsilon X \sqrt{X^2 + Y^2 - Z^2}} \right) = \arctg \left( \frac{r_2 p_x + \varepsilon p_y \sqrt{p_x^2 + p_y^2 - r_2^2}}{-r_2 p_y + \varepsilon p_x \sqrt{p_x^2 + p_y^2 - r_2^2}} \right) \quad (\text{I-99})$$

Prémultipliant par la matrice  ${}^2T_1$  l'équation (I-96), on obtient :

$${}^2T_1 \times {}^1T_0 \times [p_x \ p_y \ p_z \ 1]^T = {}^2T_3 \ {}^3D_4 \quad (\text{I-100})$$

$$\begin{bmatrix} \cos(q_2)(p_x \cos(q_1) + p_y \sin(q_1)) - p_z \sin(q_2) \\ -\sin(q_2)(p_x \cos(q_1) + p_y \sin(q_1)) - p_z \cos(q_2) \\ -p_x \sin(q_1) + p_y \cos(q_1) + r_2 \\ 1 \end{bmatrix} = \begin{bmatrix} a \sin(q_3) + d_3 \\ -a \cos(q_3) \\ 0 \\ 1 \end{bmatrix} \quad (\text{I-101})$$

Soit :  $B = p_x \cos(q_1) + p_y \sin(q_1)$ , il en résulte le système d'équations suivant :

$$\begin{cases} a \sin(q_3) = B \cos(q_2) - p_z \sin(q_2) - d_3 \\ a \cos(q_3) = B \sin(q_2) + p_z \cos(q_2) \end{cases} \quad (\text{I-102})$$

En élevant les deux équations du système (I-102) au carré et en les additionnant, on obtient :

$$a^2 = B^2 + p_z^2 + d_3^2 + 2Bd_3 \cos(q_2) - 2d_3 p_z \sin(q_2) \quad (\text{I-103})$$

$$a^2 - B^2 - p_z^2 - d_3^2 = 2Bd_3 \cos(q_2) - 2d_3 p_z \sin(q_2) \quad (\text{I-104})$$

Soit :

$$Y \cos(q_2) + X \sin(q_2) = Z, \text{ avec : } Y = 2Bd_3 ; X = -2d_3 p_z ; Z = a^2 - B^2 - p_z^2 - d_3^2$$

Comme le cas de la variable  $q_1$ , on en déduit par la même méthode la valeur de la variable  $q_2$  :

$$q_2 = \arctg \left( \frac{XZ + \varepsilon Y \sqrt{X^2 + Y^2 - Z^2}}{YZ - \varepsilon X \sqrt{X^2 + Y^2 - Z^2}} \right); \text{ avec : } \varepsilon = \pm 1 \quad (\text{I-105})$$

Ce qui donne :

$$q_2 = \arctg \left( \frac{-p_z (a^2 - d_3^2 - B^2 - p_z^2) + \varepsilon B \sqrt{(2d_3 p_z)^2 + (2Bd_3)^2 - (a^2 - d_3^2 - B^2 - p_z^2)^2}}{B (a^2 - d_3^2 - B^2 - p_z^2) + \varepsilon p_z \sqrt{(2d_3 p_z)^2 + (2Bd_3)^2 - (a^2 - d_3^2 - B^2 - p_z^2)^2}} \right);$$

$$\text{avec : } \varepsilon = \pm 1 \quad (\text{I-106})$$

A partir du système (I-102) et connaissant la valeur de la variable  $q_2$ , on peut déduire la valeur de la variable  $q_3$ , soit :

$$q_3 = \arctg \left( \frac{B \cos(q_2) - p_z \sin(q_2) - d_3}{B \sin(q_2) + p_z \cos(q_2)} \right) \quad (\text{I-107})$$

### I-6-3. Modèle dynamique du robot Puma 600

Dans ce paragraphe, nous allons calculer le modèle dynamique du robot Puma 600 en se basant sur les formules présentées dans ce chapitre. Pour des raisons de simplification de calcul, nous avons utilisé le logiciel Matlab (Pour le calcul symbolique).

Les valeurs des paramètres mesurés du robot Puma 600 utilisés, le long de notre travail, en simulation sont résumées dans le tableau I-2 [15]. Dans cette application, nous considérons le modèle dynamique donné par l'équation (I-66) sans perturbations extérieures ( $\Gamma_d = 0$ ).

<i>Paramètres</i>	<i>Valeurs</i>
masses du premier corps $m_1$	10.521 Kg
masses du deuxième corps $m_2$	10.236 Kg
masses du troisième corps $m_3$	8.767Kg
coefficient de frottement visqueux $f_1$ du 1 <sup>er</sup> corps	2.52 N.m.s/rd
coefficient de frottement visqueux $f_2$ du 2 <sup>ème</sup> corps	7 N.m.s/rd
coefficient de frottement visqueux $f_3$ du 3 <sup>ème</sup> corps	1.75 N.m.s/rd
coefficient de frottement sec $f^4$ du 1 <sup>er</sup> corps	3.6 N.m.s/rd
coefficient de frottement sec $f^5$ du 2 <sup>ème</sup> corps	10 N.m.s/rd
coefficient de frottement sec $f^6$ du 3 <sup>ème</sup> corps	2.5 N.m.s/rd
longueur du premier corps $l_1 = r_2$	0.149 m
longueur du deuxième corps $l_2 = d_3$	0.432 m
longueur du troisième corps $l_3 = a$	0.431 m

**Tableau I-2:** Les paramètres mesurés du robot manipulateur Puma 600

Pour calculer les différents éléments du modèle dynamique de ce robot et pour des raisons de simplification et d'absence des valeurs des paramètres, nous considérons que les corps du bras sont des tiges. Donc tous les produits d'inerties sont nuls, ainsi que les moments d'inertie par rapport aux axes  $x$  et  $y$ .

#### a) Calcul de la matrice d'inertie

La matrice de pseudo-inertie du premier corps est:

$$I_1 = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{1}{3}m_1 r_2^2 & 0 & \frac{1}{2}m_1 r_2 \\ 0 & 0 & 0 & 0 \\ 0 & \frac{1}{2}m_1 r_2 & 0 & m_1 \end{bmatrix} \quad (\text{I-108})$$

La matrice de pseudo-inertie du deuxième corps est:

$$I_2 = \begin{bmatrix} \frac{1}{3}m_2 d_3^2 & 0 & 0 & \frac{1}{2}m_2 d_3 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{2}m_2 d_3 & 0 & 0 & m_2 \end{bmatrix} \quad (\text{I-109})$$

La matrice de pseudo-inertie du troisième corps est:

$$I_3 = \begin{bmatrix} \frac{1}{3}m_3 a^2 & 0 & 0 & \frac{1}{2}m_3 a \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{2}m_3 a & 0 & 0 & m_3 \end{bmatrix} \quad (\text{I-110})$$

Les éléments de la matrice d'inertie sont:

$$M(q) = \begin{bmatrix} m_{11}(q) & m_{12}(q) & m_{13}(q) \\ m_{21}(q) & m_{22}(q) & m_{23}(q) \\ m_{31}(q) & m_{32}(q) & m_{33}(q) \end{bmatrix} \quad (\text{I-111})$$

où les  $m_{ij}$  avec  $i=1, 2, 3$  et  $j=1, 2, 3$  sont exprimés, à l'aide de l'équation (I-51), ainsi:

$$\begin{aligned} m_{11}(q) &= \text{trace} \left\{ \frac{\partial^0 T_1}{\partial q_1} I_1 \frac{\partial^0 T_1^T}{\partial q_1} \dot{q}_1 \dot{q}_1 + \frac{\partial^0 T_2}{\partial q_1} I_2 \frac{\partial^0 T_2^T}{\partial q_1} \dot{q}_1 \dot{q}_1 + \frac{\partial^0 T_3}{\partial q_1} I_3 \frac{\partial^0 T_3^T}{\partial q_1} \dot{q}_1 \dot{q}_1 \right\} \\ &= \frac{1}{6}m_3 r_2^2 + \frac{1}{6}d_3^2 (3m_3 + m_2) \cos(2q_2) + \frac{1}{6}m_3 a^2 \cos(2q_3 + 2q_2) + \frac{1}{2}m_3 d_3^2 + \frac{1}{6}m_2 d_3^2 \\ &\quad + \frac{1}{3}m_1 r_2^2 + \frac{1}{6}m_3 a^2 + \frac{1}{2}m_3 a d_3 \cos(2q_2 + q_3) + \frac{1}{2}m_3 a d_3 \cos(q_3) + r_2^2 m_2 \end{aligned}$$

$$\begin{aligned} m_{21}(q) &= m_{12}(q) = \text{trace} \left\{ \frac{\partial^0 T_2}{\partial q_1} I_2 \frac{\partial^0 T_2^T}{\partial q_2} \dot{q}_1 \dot{q}_2 + \frac{\partial^0 T_3}{\partial q_1} I_3 \frac{\partial^0 T_3^T}{\partial q_2} \dot{q}_1 \dot{q}_2 \right\} \\ &= -\frac{1}{2}m_3 a r_2 \sin(q_3 + q_2) - \frac{1}{2}d_3 r_2 (m_2 + 2m_3) \sin(q_2) \end{aligned}$$

$$m_{31}(q) = m_{13}(q) = \text{trace} \left\{ \frac{\partial^0 T_3}{\partial q_1} I_3 \frac{\partial^0 T_3^T}{\partial q_3} \dot{q}_1 \dot{q}_3 \right\} = -\frac{1}{2}m_3 a r_2 \sin(q_3 + q_2)$$

$$\begin{aligned} m_{22}(q) &= \text{trace} \left\{ \frac{\partial^0 T_2}{\partial q_2} I_2 \frac{\partial^0 T_2^T}{\partial q_2} \dot{q}_2 \dot{q}_2 + \frac{\partial^0 T_3}{\partial q_2} I_3 \frac{\partial^0 T_3^T}{\partial q_2} \dot{q}_2 \dot{q}_2 \right\} \\ &= \frac{1}{3}m_2 d_3^2 + m_3 d_3^2 + \frac{1}{3}m_3 a^2 + m_3 a d_3 \cos(q_3) \end{aligned}$$

$$m_{32}(q) = m_{23}(q) = \text{trace} \left\{ \frac{\partial^0 T_3}{\partial q_2} I_3 \frac{\partial^0 T_3^T}{\partial q_3} \dot{q}_2 \dot{q}_3 \right\} = \frac{1}{3}m_3 a^2 + \frac{1}{2}m_3 a d_3 \cos(q_3)$$

$$m_{33}(q) = \text{trace} \left\{ \frac{\partial^0 T_3}{\partial q_3} I_3 \frac{\partial^0 T_3^T}{\partial q_3} \dot{q}_3 \dot{q}_3 \right\} = \frac{1}{3}m_3 a^2$$

Après le calcul numérique, les éléments génériques  $m_{ij}$  de la matrice d'inertie sont :

$$\begin{aligned}
m_{11}(q) &= 3,24 + 1,77 \cos(2q_2) + 0,814 \cos(2q_3 + 2q_2) + 1,63(2q_2 + q_3) + 1,63 \cos(q_3) \\
m_{12}(q) &= -1,22 \sin(q_2) - 0,563 \sin(q_3 + q_2) \\
m_{13}(q) &= -0,563 \sin(q_3 + q_2) \\
m_{21}(q) &= -1,22 \sin(q_2) - 0,563 \sin(q_3 + q_2) \\
m_{22}(q) &= 5,17 + 3,26 \cos(q_3) \\
m_{23}(q) &= 1,63 + 1,63 \cos(q_3) \\
m_{31}(q) &= -0,563 \sin(q_3 + q_2) \\
m_{32}(q) &= 1,63 + 1,63 \cos(q_3) \\
m_{33}(q) &= 1,63
\end{aligned}$$

### b) Calcul de la matrice des forces Centrifuges et de Coriolis

La matrice des forces Centrifuges et de Coriolis est donnée comme suit :

$$C(q, \dot{q}) = \begin{bmatrix} c_{11}(q, \dot{q}) & c_{12}(q, \dot{q}) & c_{13}(q, \dot{q}) \\ c_{21}(q, \dot{q}) & c_{22}(q, \dot{q}) & c_{23}(q, \dot{q}) \\ c_{31}(q, \dot{q}) & c_{32}(q, \dot{q}) & c_{33}(q, \dot{q}) \end{bmatrix} \quad (\text{I-112})$$

En utilisant les formules du paragraphe I-5-1-3, d'où :

$$\begin{aligned}
c_{11}(q, \dot{q}) &= \frac{1}{2} \dot{q}^T \frac{\partial m_{11}(q)}{\partial q} = -\frac{1}{4} m_3 a d_3 (2\dot{q}_2 + \dot{q}_3) \sin(2q_2 + q_3) - \frac{1}{6} m_3 a^2 (\dot{q}_2 + \dot{q}_3) \sin(2q_3 + 2q_2) \\
&\quad - \frac{1}{2} d_3^2 \left( m_3 + \frac{1}{3} m_2 \right) \dot{q}_2 \sin(2q_2) - \frac{1}{4} m_3 a d_3 \sin(q_3) \dot{q}_3
\end{aligned}$$

$$\begin{aligned}
c_{12}(q, \dot{q}) &= \frac{1}{2} \dot{q}^T \frac{\partial m_{12}(q)}{\partial q} + \frac{1}{2} \left\{ \left( \frac{\partial m_{11}(q)}{\partial q_2} - \frac{\partial m_{12}(q)}{\partial q_1} \right) \dot{q}_1 + \left( \frac{\partial m_{12}(q)}{\partial q_2} - \frac{\partial m_{22}(q)}{\partial q_1} \right) \dot{q}_2 \right\} \\
&\quad + \frac{1}{2} \left( \frac{\partial m_{12}(q)}{\partial q_2} - \frac{\partial m_{32}(q)}{\partial q_1} \right) \dot{q}_3 \\
&= -\frac{1}{2} d_3^2 \left( m_3 + \frac{1}{3} m_2 \right) \dot{q}_1 \sin(2q_2) - m_3 a \frac{1}{2} \left\{ \frac{1}{3} a \sin(2q_3 + 2q_2) + d_3 \sin(2q_2 + q_3) \right\} \dot{q}_1 \\
&\quad - m_3 a \frac{1}{4} r_2 (2\dot{q}_2 + \dot{q}_3) \cos(q_3 + q_2) - d_3 r_2 \left( m_3 + \frac{1}{2} m_2 \right) \dot{q}_2 \cos(q_2)
\end{aligned}$$

$$\begin{aligned}
c_{13}(q, \dot{q}) &= \frac{1}{2} \dot{q}^T \frac{\partial m_{13}(q)}{\partial q} + \frac{1}{2} \left\{ \left( \frac{\partial m_{11}(q)}{\partial q_3} - \frac{\partial m_{13}(q)}{\partial q_1} \right) \dot{q}_1 + \left( \frac{\partial m_{12}(q)}{\partial q_3} - \frac{\partial m_{23}(q)}{\partial q_1} \right) \dot{q}_2 \right\} \\
&\quad + \frac{1}{2} \left( \frac{\partial m_{13}(q)}{\partial q_3} - \frac{\partial m_{33}(q)}{\partial q_1} \right) \dot{q}_3 \\
&= -\frac{1}{2} \left( \frac{1}{3} m_3 a^2 \sin(2q_3 + q_2) + \frac{1}{2} m_3 a d_3 \sin(2q_2 + q_3) + \frac{1}{2} m_3 a d_3 \sin(q_3) \right) \dot{q}_1 \\
&\quad - \frac{1}{4} m_3 a r_2 \cos(q_3 + q_2) \dot{q}_2
\end{aligned}$$

$$\begin{aligned}
c_{21}(q, \dot{q}) &= \frac{1}{2} \dot{q}^T \frac{\partial m_{21}(q)}{\partial q} + \frac{1}{2} \left\{ \left( \frac{\partial m_{21}(q)}{\partial q_1} - \frac{\partial m_{11}(q)}{\partial q_2} \right) \dot{q}_1 + \left( \frac{\partial m_{22}(q)}{\partial q_1} - \frac{\partial m_{21}(q)}{\partial q_2} \right) \dot{q}_2 \right\} \\
&\quad + \frac{1}{2} \left( \frac{\partial m_{23}(q)}{\partial q_1} - \frac{\partial m_{31}(q)}{\partial q_2} \right) \dot{q}_3 \\
&= \frac{1}{2} \left( d_3^2 \left( m_3 + \frac{1}{3} m_2 \right) \sin(2q_2) + \frac{1}{3} m_3 a^2 \sin(2q_3 + 2q_2) + m_3 a d_3 \sin(2q_2 + q_3) \right) \dot{q}_1 \\
&\quad - \frac{1}{4} m_3 a r_2 \cos(q_3 + q_2) \dot{q}_3
\end{aligned}$$

$$c_{22}(q, \dot{q}) = \frac{1}{2} \dot{q}^T \frac{\partial m_{22}(q)}{\partial q} = -\frac{1}{2} m_3 a d_3 \sin(q_3) \dot{q}_3$$

$$\begin{aligned}
c_{23}(q, \dot{q}) &= \frac{1}{2} \dot{q}^T \frac{\partial m_{23}(q)}{\partial q} + \frac{1}{2} \left\{ \left( \frac{\partial m_{21}(q)}{\partial q_3} - \frac{\partial m_{13}(q)}{\partial q_2} \right) \dot{q}_1 + \left( \frac{\partial m_{22}(q)}{\partial q_3} - \frac{\partial m_{23}(q)}{\partial q_2} \right) \dot{q}_2 \right\} \\
&\quad + \frac{1}{2} \left( \frac{\partial m_{23}(q)}{\partial q_3} - \frac{\partial m_{33}(q)}{\partial q_2} \right) \dot{q}_3 = -\frac{1}{2} m_3 a d_3 (\dot{q}_2 + \dot{q}_3) \sin(q_3) - \frac{1}{4} m_3 a r_2 \cos(q_3 + q_2) \dot{q}_1
\end{aligned}$$

$$\begin{aligned}
c_{31}(q, \dot{q}) &= \frac{1}{2} \dot{q}^T \frac{\partial m_{13}(q)}{\partial q} + \frac{1}{2} \left\{ \left( \frac{\partial m_{11}(q)}{\partial q_3} - \frac{\partial m_{13}(q)}{\partial q_1} \right) \dot{q}_1 + \left( \frac{\partial m_{12}(q)}{\partial q_3} - \frac{\partial m_{23}(q)}{\partial q_1} \right) \dot{q}_2 \right\} \\
&\quad + \frac{1}{2} \left( \frac{\partial m_{13}(q)}{\partial q_3} - \frac{\partial m_{33}(q)}{\partial q_1} \right) \dot{q}_3 \\
&= \frac{1}{2} \left( \frac{1}{3} m_3 a^2 \sin(2q_3 + 2q_2) + \frac{1}{2} m_3 a d_3 \sin(2q_2 + q_3) + \frac{1}{2} m_3 a d_3 \sin(q_3) \right) \dot{q}_1 \\
&\quad + \frac{1}{4} m_3 a r_2 \cos(q_3 + q_2) \dot{q}_2
\end{aligned}$$

$$\begin{aligned}
c_{32}(q, \dot{q}) &= \frac{1}{2} \dot{q}^T \frac{\partial m_{23}(q)}{\partial q} + \frac{1}{2} \left\{ \left( \frac{\partial m_{21}(q)}{\partial q_3} - \frac{\partial m_{13}(q)}{\partial q_2} \right) \dot{q}_1 + \left( \frac{\partial m_{22}(q)}{\partial q_3} - \frac{\partial m_{23}(q)}{\partial q_1} \right) \dot{q}_2 \right\} \\
&\quad + \frac{1}{2} \left( \frac{\partial m_{23}(q)}{\partial q_3} - \frac{\partial m_{33}(q)}{\partial q_2} \right) \dot{q}_3 = \frac{1}{2} m_3 a d_3 \sin(q_3) \dot{q}_2 + \frac{1}{4} m_3 a r_2 \cos(q_3 + q_2) \dot{q}_1
\end{aligned}$$

$$c_{33}(q, \dot{q}) = \frac{1}{2} \dot{q}^T \frac{\partial m_{11}(q)}{\partial q} = 0$$

Après application numérique, les éléments génériques  $C_{ij}$  de la matrice des forces centrifuges et de Coriolis sont :

$$\begin{aligned}
c_{11}(q, \dot{q}) &= -1,63(\dot{q}_2 + 0,5\dot{q}_3) \sin(2q_2 + q_3) - 0,814(\dot{q}_2 + \dot{q}_3) \sin(2q_3 + 2q_2) \\
&\quad - 1,77 \dot{q}_2 \sin(2q_2) - 0,816 \sin(q_3) \dot{q}_3
\end{aligned}$$

$$\begin{aligned}
c_{12}(q, \dot{q}) &= -1,22 \dot{q}_2 \cos(q_2) - 0,563(\dot{q}_2 + 0,5\dot{q}_3) \cos(q_2 + q_3) - 1,77 \dot{q}_1 \sin(2q_2) \\
&\quad - 0,5(1,63 \sin(2q_2 + 2q_3) + 3,26 \sin(2q_2 + q_3)) \dot{q}_1
\end{aligned}$$

$$c_{13}(q, \dot{q}) = 0,5(-1,63 \sin(2q_2 + 2q_3) - 1,63 \sin(2q_2 + q_3) - 1,63 \sin(q_3)) \dot{q}_1 - 0,282 \cos(q_2 + q_3) \dot{q}_2$$

$$c_{21}(q, \dot{q}) = 0,5(3,55 \sin(2q_2) + 1,63 \sin(2q_2 + 2q_3) + 3,26 \sin(q_3 + 2q_2)) \dot{q}_1 - 0,282 \cos(q_3 + q_2) \dot{q}_3$$

$$c_{22}(q, \dot{q}) = -1,63 \sin(q_3) \dot{q}_3$$

$$c_{23}(q, \dot{q}) = -1,63(\dot{q}_3 + \dot{q}_2) \sin(q_3) - 0,282 \cos(q_3 + q_2) \dot{q}_1$$

$$c_{31}(q, \dot{q}) = 0,5(1,63 \sin(2q_2 + 2q_3) + 1,63 \sin(2q_2 + q_3) + 1,63 \sin(q_3)) \dot{q}_1 + 0,282 \cos(q_3 + q_2) \dot{q}_2$$

$$c_{32}(q, \dot{q}) = 1,63 \sin(q_3) \dot{q}_2 + 0,282 \cos(q_3 + q_2) \dot{q}_1$$

$$c_{33}(q, \dot{q}) = 0$$

### c) Calcul du vecteur de gravité du robot Puma 600

Afin de calculer le vecteur de gravité, nous commençons par la détermination de l'énergie potentielle du robot, et cela par l'application des équations (I-52) et (I-54):

$$U_1 = -m_1 {}^0g^T {}^0T_1 {}^1\bar{r} = 0$$

et 
$$U_2 = -m_2 {}^0g^T {}^0T_2 {}^2\bar{r} = -\frac{1}{2} m_2 g d_3 \sin(q_2)$$

et 
$$U_3 = -m_3 {}^0g^T {}^0T_3 {}^3\bar{r} = -m_3 g \left\{ \frac{1}{2} a \sin(q_2 + q_3) + d_3 \sin(q_2) \right\}$$

où :

$${}^0g^T = [0 \quad 0 \quad -g \quad 0]$$

$${}^1\bar{r} = \left[ 0 \quad \frac{1}{2} r_2 \quad 0 \quad 1 \right]^T$$

$${}^2\bar{r} = \left[ \frac{1}{2} d_3 \quad 0 \quad 0 \quad 1 \right]^T$$

$${}^3\bar{r} = \left[ \frac{1}{2} a \quad 0 \quad 0 \quad 1 \right]^T$$

alors : 
$$U(q) = \frac{1}{2} m_2 g d_3 \sin(q_2) - m_3 g \left\{ \frac{1}{2} a \sin(q_2 + q_3) + d_3 \sin(q_2) \right\}$$

L'utilisation de cette dernière expression avec l'équation (I-62), conduit au calcul des éléments du vecteur de gravité :

$$g_1(q) = \frac{\partial U(q)}{\partial q_1} = 0$$

$$g_2(q) = \frac{\partial U(q)}{\partial q_2} = -\frac{1}{2} g d_3 (m_2 + 2m_3) \cos(q_2) - \frac{1}{2} m_3 g a \cos(q_2 + q_3)$$

$$g_3(q) = \frac{\partial U(q)}{\partial q_3} = -\frac{1}{2} m_3 g a \cos(q_2 + q_3)$$

Finalement, après application numérique, le vecteur de gravité devient :

$$g(q) = \begin{bmatrix} 0 \\ -80,5 \cos(q_2) - 37,1 \cos(q_3 + q_2) \\ -37,1 \cos(q_3 + q_2) \end{bmatrix}$$

**d) Calcul du vecteur de frottements du robot Puma 600**

Par l'utilisation des équations (I-67) et (I-82), le vecteur de frottement peut s'exprimer ainsi:

$$F(\dot{q}) = \begin{bmatrix} f_1 \dot{q}_1 + f_4 \operatorname{sgn}(\dot{q}_1) \\ f_2 \dot{q}_2 + f_5 \operatorname{sgn}(\dot{q}_2) \\ f_3 \dot{q}_3 + f_6 \operatorname{sgn}(\dot{q}_3) \end{bmatrix} \quad (\text{I-113})$$

Après application numérique, le vecteur de frottements est :

$$F(\dot{q}) = \begin{bmatrix} 2,52 \dot{q}_1 + 3,6 \operatorname{sgn}(\dot{q}_1) \\ 7 \dot{q}_2 + 10 \operatorname{sgn}(\dot{q}_2) \\ 1,75 \dot{q}_3 + 2,5 \operatorname{sgn}(\dot{q}_3) \end{bmatrix}$$

**Conclusion**

L'objet de ce chapitre concerne un état de l'art sur les différentes notions de base et les techniques de modélisations appliquées aux systèmes robotisés. Les modélisations géométriques, cinématiques et dynamique ont été détaillées. Quelques propriétés structurelles sur les différents paramètres du modèle dynamique d'un robot manipulateur ont été détaillées vu leur importance dans l'étude et la conception de leur différentes lois de commande. Enfin, nous avons présenté un exemple d'application détaillé montrant les différentes étapes utilisées pour la modélisation d'un robot Puma 600 et qui sera utilisé dans la suite de notre travail.

Notons qu'un bras manipulateur en boucle ouverte reste toujours instable. Afin de le stabiliser et de le rendre utile, nous devons l'insérer, en boucle fermée, avec des lois de commande bien calculées.

L'objectif du prochain chapitre sera la présentation des différentes techniques de commande appliquées aux bras manipulateurs pour améliorer leurs performances et leurs stabilités.

**Références bibliographiques**

- [1] Canudas de Wit C., B. Brogliato, A. De Luca, B. Siciliano, P. Tomei, C. Samson, R. Ortega, G. Campion, B. D'Andrea-Novel, W. Khalil, *Théorie de la commande des robots*, Laboratoire d'automatique de grenoble, grenoble, France, 1992.
- [2] Coiffet P., *La robotique: principes et applications*, 3<sup>e</sup> édition revue et complétée, Hermès, paris, 1992.
- [3] Craig J. J., *Adaptive control of mechanical manipulators*, Adisson-Wesley Publishing Company, Inc, New York, 1988.
- [4] Chevallereau C., Khalil W., *Efficient method for the calculation of the pseudo inverse kinematic problem*, Proc. IEEE Int. Conf. On robotics and Automation, Raleigh, p. 1842-1848, 1987.
- [5] Dombre E., Wisama Khalil, *Modélisation et commande des robots*, Hermès, paris, 1988.
- [6] Denavit J., R. S. Hartenberg, *A kinematic notation for lower-pair mechanisms based on matrices*, Journal of Applied Mechanics, p. 215-221, 1955.
- [7] Giordano M. et J. Lottin, *COURS DE ROBOTIQUE : Description et fonctionnement des robots industriels*, Armand Colin, paris, France, 1990.
- [8] Gorla B., M. Renaud, *Modèles des robots manipulateurs : application à leur commande*, Cepadues-éditions, Toulouse, France, 1984.
- [9] Khalil W., Etienne Dombre, *Modélisation identification et commande des robots*, Hermès, paris, 1999.
- [10] Lewis F. L., C. T. abdallah, D. M. Dawson, *Control of robot Manipulators*, Macmillan, New York, 1993.
- [11] Lozano R. et D. Taoutaou, *Commande adaptative et applications*, Hermès, paris, France, 2001.
- [12] Paul R. C. P., *Robot manipulators: mathematics, programming and control*, MIT press, Cambridge, MA, 1981.
- [13] Spong M. W., M. Vidyasagar, *Robot dynamics and control*, John Wiley & Sons, New York, 1989.
- [14] Vibet C., *ROBOTS : principes et contrôle*, ellipses, paris, France, 1987.
- [15] KADDOUR E. L., *Simulation de la commande dynamique d'un bras manipulateur à 6 ddl appliqué au robot puma 600*, USTO, 1995.

## *Chapitre II*

*Etat de l'art sur les commandes appliquées  
aux systèmes robotisés / Applications*

## Introduction

L'objectif de la commande des manipulateurs est la recherche des meilleures lois de commandes qui améliorent certains critères de précision, de rapidité et de stabilité des robots afin d'assurer un bon fonctionnement. La conception d'une loi de commande en robotique se fait généralement soit par l'application de la théorie linéaire par la linéarisation du système au point de fonctionnement, soit par l'application de l'approche non linéaire. Les techniques fondamentales de commande peuvent être répertoriées en : commandes classiques ; commandes adaptatives et commandes robustes.

Dans ce chapitre, nous allons exposer quelques techniques sur le calcul des consignes de commande (génération de mouvement) des manipulateurs. Nous allons donner une description détaillée sur les différentes lois de commande classique qui concernent les problèmes de position ou déplacement (régulation) et de mouvement (poursuite de trajectoire) avec une étude précise sur leur stabilité. Ensuite, nous allons détailler le problème de commande en effort quand le robot rentre en contact avec son environnement telle que la compliance passive et les commandes hybrides qui constituent le cœur des recherches actuelles. Nous montrons quelques applications sur un robot Puma 600. Enfin, nous terminons ce chapitre par une conclusion.

### II-1. Commande dynamique des robots manipulateurs rigides

Quelle que soit la tâche à exécuter par le robot manipulateur, on peut avoir deux cas de mouvement de celui-ci selon l'environnement de son travail; le mouvement point à point et le mouvement selon une trajectoire bien déterminée. Dans le premier cas le robot est commandé pour aller d'une configuration initiale à une configuration finale, dont la trajectoire est libre entre ces deux points. Cette approche est convenable lorsque l'espace de travail ne contient pas d'obstacles. Pour le deuxième cas, le robot doit suivre entre ces deux points une trajectoire bien définie (désirée) qui lui permet d'éviter les obstacles lors de son déplacement.

#### II-1-1. Génération de mouvement entre deux points

On considère un robot à  $n$  degrés de liberté. Soit  $q^i$  et  $q^f$  les vecteurs des coordonnées articulaires correspondant aux configurations initiale et finale. Le mouvement interpolé entre  $q^i$  et  $q^f$  en fonction du temps  $t$  est décrit par l'équation suivante [3, 9] :

$$q(t) = q^i + r(t)D \quad 0 \leq t \leq t_f \quad (\text{II-1})$$

avec :

$$D = q^f - q^i \quad (\text{II-2})$$

Les valeurs aux limites de la fonction d'interpolation  $r(t)$  sont données par :

$$\begin{cases} r(0) = 0 \\ r(t_f) = 1 \end{cases} \quad (\text{II-3})$$

Plusieurs fonctions permettent de satisfaire le passage par  $q^i$  à  $t=0$  et par  $q^f$  à  $t=t_f$ . Nous étudierons successivement l'interpolation polynomiale et la loi Bang-Bang.

### II-1-1-1. Interpolation polynomiale

Les modes d'interpolation polynomiale les plus fréquemment rencontrés sont l'interpolation linéaire et l'interpolation par des polynômes de degrés trois et cinq [2, 3, 9].

### II-1-1-2. Interpolation linéaire

Dans ce cas, le mouvement de chaque articulation est décrit par une équation linéaire en temps. L'équation du mouvement est donnée comme suit :

$$q(t) = q^i + r(t)D; \quad r(t) = \frac{t}{t_f} \quad (\text{II-4})$$

La représentation graphique de ce mouvement est donnée par la figure II-1.

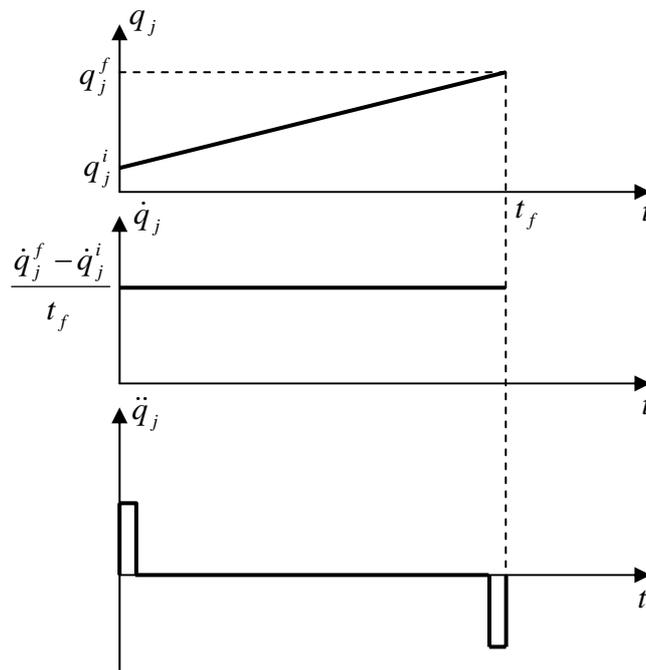


Figure II-1: Interpolation linéaire sur une articulation donnée  $j$ .

Cette loi de mouvement impose une vitesse constante le long de la trajectoire, elle est continue en position et en vitesse mais discontinue en accélération. Cette loi n'est pas utilisée dans le cas des robots réels à cause des à-coups qu'elle provoque.

### II-1-1-3. Interpolation polynomiale de degré trois

Si nous imposons une vitesse nulle aux points initial et final, nous ajouterons deux contraintes aux deux contraintes de position citées dans la méthode d'interpolation linéaire. Le polynôme qui satisfait ces quatre contraintes est de degré trois. Il a la forme générale suivante :

$$q(t) = a_0 + a_1 t + a_2 t^2 + a_3 t^3 \quad (\text{II-5})$$

avec les conditions initiales et finales :

$$\begin{cases} q(0) = q^i, & q(t_f) = q^f \\ \dot{q}(0) = 0, & \dot{q}(t_f) = 0 \end{cases} \quad (\text{II-6})$$

sous ces conditions nous définissons les coefficients de la relation (II-5) :

$$a_0 = q^i, \quad a_1 = 0, \quad a_2 = \frac{3}{t_f^2} D, \quad a_3 = \frac{2}{t_f^3} D \quad (\text{II-7})$$

Nous remplaçons (II-7) dans (II-5) nous obtenons l'équation du mouvement :

$$q(t) = q^i + r(t) D; \quad r(t) = 3 \left( \frac{t}{t_f} \right)^2 - 2 \left( \frac{t}{t_f} \right)^3 \quad (\text{II-8})$$

La figure II-2 donne l'évolution de la position, la vitesse et l'accélération en fonction du temps pour l'axe  $j$  donné. Cette loi de mouvement assure la continuité de vitesse mais pas celle d'accélération.

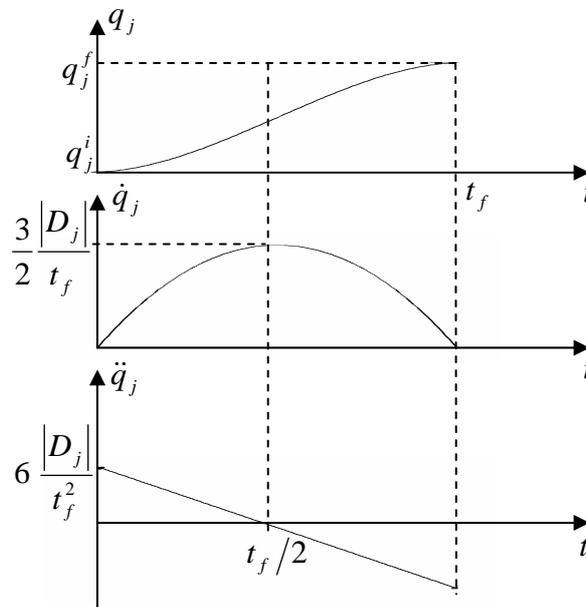


Figure II-2 : Interpolation polynomiale de degré trois sur une articulation donnée  $j$ .

**II-1-1-4. Interpolation polynomiale de degré cinq**

Pour avoir la continuité des accélérations, nous sommes obligés de satisfaire six contraintes et le polynôme d'interpolation sera de degré cinq. Avec les conditions supplémentaires :

$$\ddot{q}(0) = 0, \quad \ddot{q}(t_f) = 0 \quad (\text{II-9})$$

Nous pouvons démontrer que l'équation du mouvement, en utilisant les mêmes paramètres avant, s'écrit comme suit :

$$q(t) = q^i + r(t)D, \quad r(t) = 10\left(\frac{t}{t_f}\right)^3 - 15\left(\frac{t}{t_f}\right)^4 + 6\left(\frac{t}{t_f}\right)^5 \quad (\text{II-10})$$

L'évolution de la position, la vitesse et l'accélération de l'articulation  $j$  est représentée sur la figure II-3.

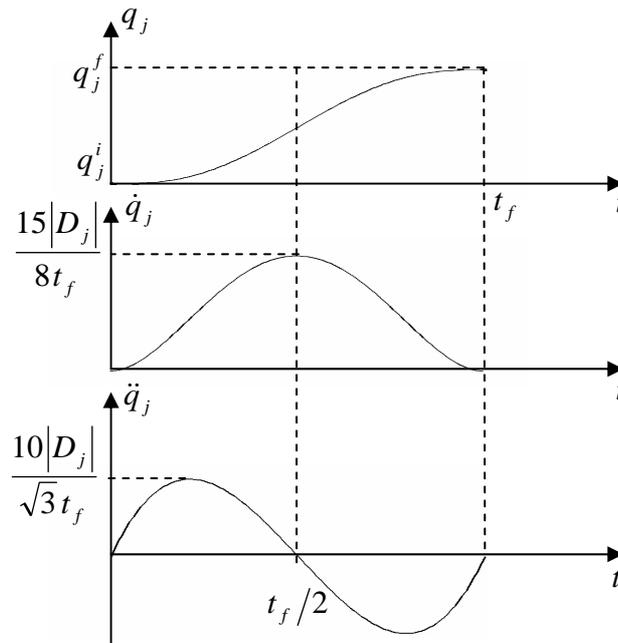


Figure II-3 : Interpolation polynomiale de degré cinq sur une articulation donnée  $j$ .

### II-1-1-5. Interpolation Bang-Bang

Le mouvement est représenté par une phase d'accélération constante jusqu'à  $\frac{t_f}{2}$  puis par une phase de décélération constante jusqu'à  $t_f$ . Les vitesses de départ et d'arrivée sont nulles. La trajectoire est continue pour la position et la vitesse mais discontinue pour l'accélération (voir figure II-4). L'équation du mouvement dans ce cas est donnée comme suit :

$$\begin{cases} q(t) = q^i + 2\left(\frac{t}{t_f}\right)^2 D, & 0 \leq t \leq \frac{t_f}{2} \\ q(t) = q^i + \left[-1 + 4\left(\frac{t}{t_f}\right) - 2\left(\frac{t}{t_f}\right)^2\right] D, & \frac{t_f}{2} \leq t \leq t_f \end{cases} \quad (\text{II-11})$$

La génération du mouvement a pour objectif de calculer les consignes de référence en position, vitesse et accélération qui sont fonction du temps.

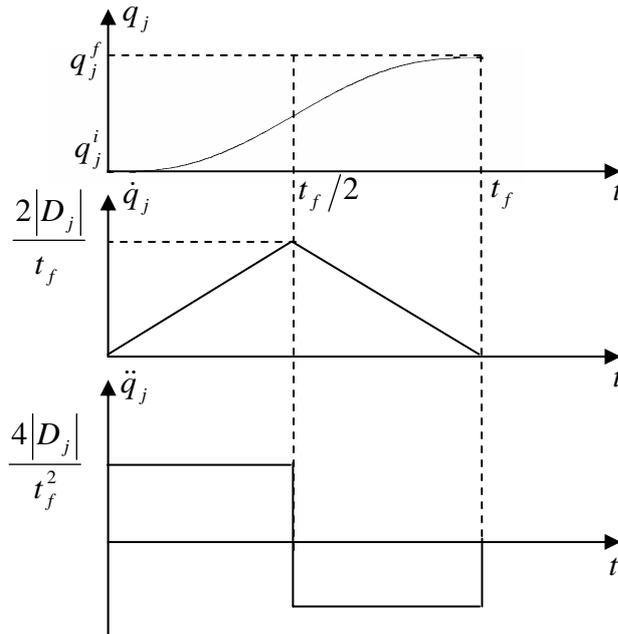


Figure II-4 : Interpolation Bang-Bang sur une articulation donnée  $j$ .

**II-1-2. Lois de commande dynamique du robot manipulateur**

Les objectifs de commande pour des manipulateurs robotiques recouvrent un vaste spectre allant de ; la commande de position pure (régulation) telles que : la commande  $PD$ , la commande  $PID$ , la commande  $PD$  avec compensation de l'effet de gravité et la commande par découplage non-linéaire ; du mouvement (poursuite) telles que : la commande basée sur la linéarisation avec retour d'état (la commande Couple Calculé) et les commande basées sur la théorie de Lyapov (la commande de Paden et Panja, la commande de Slotine et Li, ...) ; la commande adaptative, de la force ; de l'impédance mécanique jusqu'à la commande hybride position/force.

**II-1-2-1. Commande classique de type  $PD$  [1, 15, 23]**

Par la négligence des termes de frottements et les perturbations extérieures, le modèle dynamique du robot devient :

$$\Gamma = M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) \tag{II-12}$$

Pour cette approche la commande  $PD$  (Proportionnelle dérivée) a la forme de base suivante :

$$\Gamma = K_p (q_d - q) - K_v \dot{q} \tag{II-13}$$

Où  $K_p, K_v \in \mathfrak{R}^{n \times n}$  sont respectivement des matrices diagonales définies positives des gains de position et de vitesse. L'objectif de cette régulation est de maintenir la position de la variable articulaire  $q$  du robot manipulateur constante autour d'une position désirée noté  $q_d$ .

Soit  $\tilde{q} = q - q_d$  l'erreur de position, donc  $\dot{\tilde{q}} = \dot{q}$  et  $\ddot{\tilde{q}} = \ddot{q}$  puisque  $\dot{q}_d = 0$ . En appliquant cette loi sur la dynamique du robot, nous obtenons la dynamique du système en boucle fermée suivante :

$$M(q)\ddot{\tilde{q}} + C(q, \dot{q})\dot{\tilde{q}} + G(q) + K_v \dot{\tilde{q}} + K_p \tilde{q} = 0 \quad (\text{II-14})$$

La figure II-5, présente le principe de cette commande.

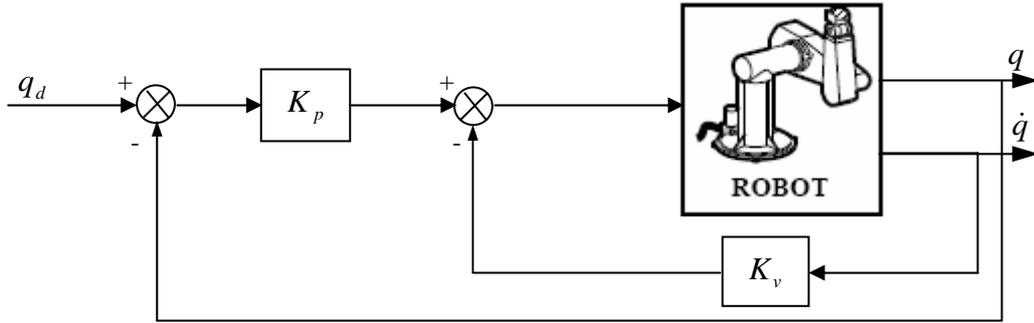


Figure II-5 : Schéma bloc de la commande classique PD.

Les points d'équilibre du système en boucle fermée sont définis par l'ensemble :

$$S = \{(\tilde{q}, \dot{\tilde{q}}) : K_p \tilde{q} + G(\tilde{q} + q_d) = 0, \dot{\tilde{q}} = 0\} \quad (\text{II-15})$$

La stabilité du système bouclé est démontrée au sens de Lyapunov. En prenant pour cela la fonction candidate suivante :

$$V(\tilde{q}, \dot{\tilde{q}}) = \frac{1}{2} \dot{\tilde{q}}^T M(q) \dot{\tilde{q}} + \frac{1}{2} \tilde{q}^T K_p \tilde{q} + U(q) + U_0 \quad (\text{II-16})$$

où  $U(q)$  et  $U_0$  représentent respectivement l'énergie potentielle et une constante positive choisie pour que la fonction  $V$  soit définie positive.

La dérivée par rapport au temps de la relation (II-16) est :

$$\dot{V}(\tilde{q}, \dot{\tilde{q}}) = \dot{\tilde{q}}^T M(q) \ddot{\tilde{q}} + \frac{1}{2} \dot{\tilde{q}}^T \dot{M}(q) \dot{\tilde{q}} + \dot{\tilde{q}}^T K_p \tilde{q} + \dot{U}(q) \quad (\text{II-17})$$

Le calcul de la dérivée d'énergie potentielle se fait comme suit :

$$\dot{U}(q) = \left( \frac{\partial U}{\partial q} \right)^T \frac{\partial U}{\partial q} = \dot{q}^T G(q) \quad (\text{II-18})$$

Car ; 
$$\frac{\partial U}{\partial q} = G(q) \quad (\text{II-19})$$

Par le remplacement de  $M(q)\ddot{\tilde{q}}$  obtenue de la relation (II-14) dans (II-17) nous trouvons :

$$\dot{V}(\tilde{q}, \dot{\tilde{q}}) = -\dot{\tilde{q}}^T C(q, \dot{q}) \dot{\tilde{q}} - \dot{\tilde{q}}^T K_v \dot{\tilde{q}} - \dot{\tilde{q}}^T K_p \tilde{q} - \dot{\tilde{q}}^T G(q) + \frac{1}{2} \dot{\tilde{q}}^T \dot{M}(q) \dot{\tilde{q}} + \dot{\tilde{q}}^T K_p \tilde{q} + \dot{\tilde{q}}^T G(q)$$

En exploitant la propriété  $\tilde{q}^T \left[ \frac{1}{2} \dot{M}(q) - C(q, \dot{q}) \right] \dot{\tilde{q}} = 0$ , nous obtenons le résultat suivant :

$$\dot{V}(\tilde{q}, \dot{\tilde{q}}) = -\tilde{q}^T K_v \dot{\tilde{q}} \quad (\text{II-20})$$

qui est une fonction semi-définie négative, alors le point d'équilibre  $(\tilde{q}, \dot{\tilde{q}}) = (0, 0)$  est stable au sens de Lyapunov. Mais la négativité de la relation (II-20) assure uniquement la convergence de l'état  $\tilde{q}$  à 0 (zéro), donc pour compléter l'analyse de la stabilité on peut appliquer le théorème de LaSalle et montrer que les états du système convergeront vers l'ensemble  $S$  (équation (II-15)) dans un temps fini.

Nous avons  $\dot{V}(\tilde{q}, \dot{\tilde{q}}) = 0$  seulement pour  $\dot{\tilde{q}} = 0$  ( $\ddot{\tilde{q}} = 0$ ), alors par le remplacement de cette dernière dans la dynamique de la boucle fermée (II-14), nous avons :

$$K_p \tilde{q} + G(q) = 0 \quad (\text{II-21})$$

qui est sûrement vrai si,

$$\tilde{q} = -K_p^{-1} G(q) \quad (\text{II-22})$$

alors,

$$\|\tilde{q}\| \leq \|K_p^{-1}\| \|G(q)\| \leq g_b \|K_p^{-1}\| \quad (\text{II-23})$$

où  $g_b$  est une borne supérieure du vecteur de gravité donnée dans l'équation (I-81).

Donc nous concluons que le système converge localement vers l'ensemble  $S$  et que la précision du système dépend de la matrice du gain  $K_p$ .

En outre, si  $K_p$  et  $K_v$  sont des matrices définies positives alors toutes les solutions  $(\tilde{q}, \dot{\tilde{q}})$  convergent asymptotiquement globalement (si  $K_p$  est grand) ou localement (si  $K_p$  est petit) à l'ensemble des points d'équilibre  $S$ .

L'erreur de l'état stable peut être réduite en augmentant arbitrairement le gain  $K_p$ , néanmoins le bruit de mesure et autres dynamiques non modélisables limiteront l'utilisation des hauts gains.

### II-1-2-2. Commande classique de type PID [1, 15, 23]

La commande PID peut être utilisée pour résoudre le problème des hauts gains de la commande PD. Pour cela, l'action intégrale peut être soit ajoutée à la commande PD pour agir sur les forces de gravité qui peuvent être considérées dans certaines mesures comme des perturbations constantes. Cela mène à une commande d'une structure PID (Proportionnelle Intégrale et Dérivée).

La figure II-6 présente le principe de cette commande et qui est donnée par :

$$\Gamma = -K_v \dot{q} + K_p (q_d - q) + K_i \int_0^t (q_d - q) dt \quad (\text{II-24})$$

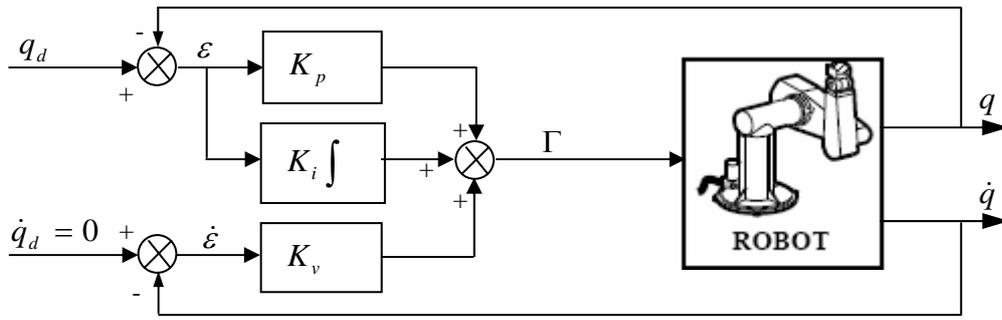


Figure II-6 : Schéma bloc d'une commande classique PID.

Nous représentons l'équation (II-12) sous la forme d'espace d'état, avec :  $X_1 = q$ ,  $X_2 = \dot{q}$  et  $X^T = (X_1^T, X_2^T)$  comme suit :

$$\begin{cases} \dot{X}_1 = X_2 \\ \dot{X}_2 = M(X_1)^{-1} (-C(X_1, X_2) X_2 - G(X_1) + \Gamma) \end{cases} \quad (II-25)$$

Les erreurs du vecteur d'état sont définies comme suit :

$$\begin{cases} \tilde{X}_1 = X_1 - X_{d1} \\ \tilde{X}_2 = X_2 \end{cases} \quad (II-26)$$

avec  $X_{d1}$  est le vecteur des positions désirées  $q_d$ . Nous introduisons l'intégrale de l'erreur de position plus la composante  $-G(X_{d1})$ , comme un troisième état.

$$\tilde{X}_3 = \int_0^t \tilde{X}_1 d\tau - G(X_{d1}) \quad (II-27)$$

Le remplacement de l'équation (II-24) dans (II-25) donne :

$$\begin{cases} \dot{\tilde{X}}_1 = \tilde{X}_2 \\ \dot{\tilde{X}}_2 = M(X_1)^{-1} [-C(X_1, \tilde{X}_2) \tilde{X}_2 - G(X_1) - K_p \tilde{X}_1 - K_v \tilde{X}_2 - K_i (\tilde{X}_3 + G(X_{d1}))] \\ \dot{\tilde{X}}_3 = \tilde{X}_1 \end{cases} \quad (II-28)$$

Pour étudier la stabilité locale de l'équation d'erreur du système précitée, nous considérons une approximation de sa tangente linéaire autour de l'origine,

$$\dot{\tilde{X}} = A \tilde{X} + o(\tilde{X}) \quad (II-29)$$

où :

$$A = \begin{bmatrix} 0 & I & 0 \\ (-B_0 K_p + A_0) & -B_0 K_v & K_i \\ I & 0 & 0 \end{bmatrix}, \quad \tilde{X} = \begin{bmatrix} \tilde{X}_1 \\ \tilde{X}_2 \\ \tilde{X}_3 \end{bmatrix} \quad (II-30)$$

et  $o(\tilde{X})$  décrit les termes du plus haut ordre de l'extension de Taylor.

Notons que pour la dynamique du robot,  $o(\tilde{X})$  satisfait :  $\|o(\tilde{X})\| \leq c_1 \|X\|^2$ .

$$\text{Les matrices } A_0 \text{ et } B_0 \text{ sont données par : } A_0 = \left. \frac{\partial M^{-1}G}{\partial X_1} \right|_{X_1=X_2}, \quad B_0 = M(X_{d1})^{-1} \quad (\text{II-31})$$

Si les gains  $K_p, K_v$  et  $K_i$  sont choisis pour rendre le système autonome linéaire asymptotiquement stable, les états du système en boucle fermée (II-29) tendent vers zéro. Donc  $\tilde{X}_1 \rightarrow 0$ , l'erreur d'état stable peut être nulle sans la présence des perturbations.

Si  $A$  est une matrice stable, nous pouvons alors définir  $V$  comme :

$$V = \tilde{X}^T P \tilde{X} \quad (\text{II-32})$$

où  $P$  est une matrice symétrique définie positive satisfaisant :  $PA + A^T P = -Q$  pour  $Q > 0$ . Alors nous obtenons :

$$\dot{V} = -\tilde{X}^T Q \tilde{X} + 2\tilde{X}^T P_0(\tilde{X}) \quad (\text{II-33})$$

$$\leq -\lambda_{\min} Q \|\tilde{X}\|^2 + 2c_0 P_0 \|\tilde{X}\|^3 \quad (\text{II-34})$$

$$= -\|\tilde{X}\|^2 (\lambda_{\min} Q - 2c_0 P_0 \|\tilde{X}\|) \quad (\text{II-35})$$

avec  $c_0$  est une constante positive et  $P_0$  est une borne supérieure pour la norme de  $P$ . Au voisinage de l'origine, le terme carré négatif domine le terme cubique positif. Le point d'équilibre est donc localement stable au sens de Lyapunov. Il est maintenant simple à prouver en se basant sur le théorème des ensembles invariants que le vecteur de l'erreur converge vers zéro.

Il est important de savoir que l'application de la commande *PID* sur des robots manipulateurs réels peut causer de sérieux problèmes si elle n'est pas bien calculée. Ces problèmes sont dus au phénomène appelé «Integrator windup».

### II-1-2-3. Commande de type *PD* avec compensation de l'effet de gravité (Commande point à point) [3, 9, 10, 16]

Les non linéarités du système du robot ne sont pas compensées par une simple loi de commande *PD*. Alors, en ajoutant quelques termes dynamiques non linéaires dans cette commande (*PD*) on peut avoir de bons résultats. En effet, la compensation de la gravité agit comme un correcteur qui compense seulement l'ensemble des forces qui créent le dépassement et le comportement transitoire asymétrique du système.

Donc la commande *PD* sera remplacée par une commande *PD* plus le terme des forces de gravité. Ce type de loi de commande s'appelle commande point à point. L'objectif de cette loi de commande consiste à maintenir les positions articulaires du robot manipulateur autour d'une valeur désirée constante  $q_d = C^{ste}$  et  $\dot{q}_d = 0$ . Pour cette approche, le couple  $\Gamma$  appliqué aux actionneurs est donné par :

$$\Gamma = G(q) - K_p \tilde{q} - K_v \dot{q} \quad (\text{II-36})$$

où  $\tilde{q} = q - q_d$  désigne l'écart de position,  $K_p \in \mathcal{R}^{n \times n}$ ,  $K_v \in \mathcal{R}^{n \times n}$  sont des matrices diagonales définies positives des gains respectivement de position et de vitesse. La loi (II-36) est composée d'un terme de compensation de la gravité et d'un autre terme de positionnement.

En insérant la relation (II-36) dans le modèle (II-12) nous obtenons la dynamique du système en boucle fermée (voir figure II-7) :

$$\begin{aligned} M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) &= G(q) - K_p \tilde{q} - K_v \dot{q} \\ M(q)\ddot{q} + C(q, \dot{q})\dot{q} + K_p \tilde{q} + K_v \dot{q} &= 0 \end{aligned} \quad (\text{II-37})$$

Pour montrer la stabilité du système (II-37) en boucle fermée, on utilise une fonction de Lyapunov candidate définie positive, composée de l'énergie cinétique et d'une énergie potentielle introduite par la loi de commande donnée par :

$$V(\dot{q}, \tilde{q}) = \frac{1}{2} \dot{q}^T M(q) \dot{q} + \frac{1}{2} \tilde{q}^T K_p \tilde{q} \quad (\text{II-38})$$

La dérivée de la fonction (II-38) donne :

$$\dot{V}(\dot{q}, \tilde{q}) = \dot{q}^T M(q) \ddot{q} + \frac{1}{2} \dot{q}^T \dot{M}(q) \dot{q} + \dot{q}^T K_p \tilde{q} \quad (\text{II-39})$$

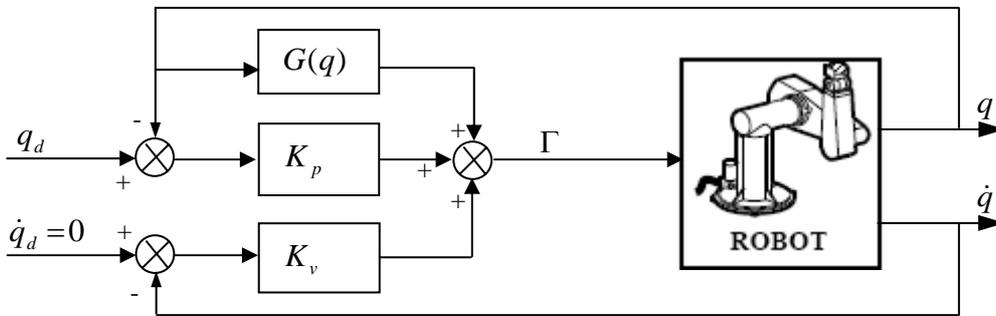


Figure II-7 : Schéma bloc d'une commande PD avec compensation de l'effet de gravité.

En substituant l'expression de  $M(q)\ddot{q}$  dans la relation (II-39) nous obtenons :

$$\dot{V}(\dot{q}, \tilde{q}) = \dot{q}^T [-C(q, \dot{q})\dot{q} - K_v \dot{q} - K_p \tilde{q}] + \frac{1}{2} \dot{q}^T \dot{M}(q) \dot{q} + \dot{q}^T K_p \tilde{q} \quad (\text{II-40})$$

$$= -\dot{q}^T K_v \dot{q} + \frac{1}{2} \dot{q}^T [\dot{M}(q) - 2C(q, \dot{q})] \dot{q} \quad (\text{II-41})$$

En utilisant le fait que la matrice  $\dot{M}(q) - 2C(q, \dot{q})$  est antisymétrique, l'équation (II-41) devient :

$$\dot{V}(\dot{q}, \tilde{q}) = -\dot{q}^T K_v \dot{q} \leq 0 \quad (\text{II-42})$$

L'état  $x^T = [\tilde{q}^T \quad \dot{q}^T]^T$  du système bouclé est borné. On considère alors une trajectoire  $(q(t), \dot{q}(t))$  continue dans l'ensemble :

$$Z = \{(\dot{q}, \tilde{q}) \mid \dot{V}(\dot{q}, \tilde{q}) = 0 \text{ (} \dot{q} = 0 \text{)}\} \quad (\text{II-43})$$

D'après l'équation (II-37), nous avons forcément  $\tilde{q} = 0$  d'où  $q = q_d$ . La seule trajectoire contenue dans  $Z$  est le point d'équilibre du robot manipulateur. D'après le théorème de LaSalle, le point d'équilibre  $(q, \dot{q}) = (q_d, 0)$  est asymptotiquement stable.

Lorsque l'application nécessite des évolutions rapides du robot et une grande précision dynamique, il est nécessaire de concevoir un système de commande plus avancé qui prenne en compte tout ou partie des forces d'interaction dynamiques [8, 11, 13, 26]. L'emploi de la commande par découplage non linéaire constitue une bonne approche en ce sens. Ce type de commande est aussi connu sous le nom de commande dynamique parce qu'il est basé sur l'utilisation du modèle dynamique.

#### II-1-2-4. Commande par découplage non linéaire [3, 9]

Cette commande consiste à transformer par retour d'état le problème de commande d'un système non linéaire en un problème de commande d'un système linéaire. Dans le cas général le problème de linéarisation par retour d'état d'un système non linéaire est difficile à résoudre et il n'existe pas toujours de solution. Cependant, dans le cas des robots manipulateurs, l'élaboration d'une loi de commande qui linéarise et découple les équations est simplifiée du fait que le modèle dont nous disposons est un modèle inverse.

La loi de commande dans ce cas est donnée par :

$$\Gamma = M(q)\Gamma_0 + C(q, \dot{q})\dot{q} + g(q) \quad (\text{II-44})$$

En substituant cette commande dans le modèle dynamique (II-12) nous obtenons l'équation du système en boucle fermée :

$$M(q)(\ddot{q} - \Gamma_0) = 0 \quad (\text{II-45})$$

En tenant compte du fait que la matrice  $M(q)$  est régulière (inversible), nous aurons  $n$  systèmes linéaires découplés ;

$$\ddot{q} = \Gamma_0 \quad (\text{II-46})$$

où  $\Gamma_0$  est une entrée auxiliaire de la commande à designer. Pour résoudre le problème de régulation, un choix possible de  $\Gamma_0$  est de prendre :

$$\Gamma_0 = K_p (q_d - q) - K_v \dot{q} \quad (\text{II-47})$$

où  $K_p$  et  $K_v \in \mathfrak{R}^{n \times n}$  sont des matrices diagonales définies positives, déterminées en fonction de la dynamique désirée.

Dans le cas d'une modélisation parfaite et pour des erreurs initiales nulles, et à partir des équations (II-46) et (II-47), l'équation de la boucle fermée du système est :

$$\ddot{q} + K_v \dot{q} + K_p \tilde{q} = 0 \quad (\text{II-48})$$

qui est une équation linéaire découplée du deuxième ordre.

La figure II-8 présente le principe de cette approche.

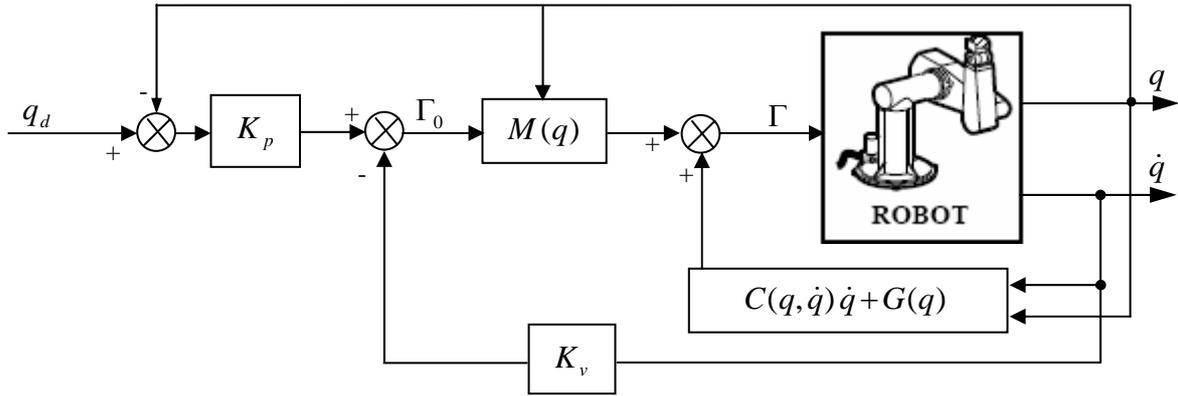


Figure II-8 : Schéma bloc de la commande par découplage non linéaire.

La commande définie par les équations (II-44) et (II-47) comporte deux parties ; la première compense les forces centrifuges, les forces de Coriolis, de gravité et de frottement, alors que la deuxième est une correction de position et un retour de vitesse à gains variables.

L'équation d'erreur (II-48) est asymptotiquement stable par un choix convenable des matrices des gains  $K_p$  et  $K_v$ . Pour démontrer ça, nous pouvons écrire l'équation (II-48) sous la forme d'espace d'état, avec  $x = (\tilde{q}^T, \dot{\tilde{q}}^T)^T$  :

$$\dot{x} = \begin{pmatrix} 0 & I \\ -K_p & -K_v \end{pmatrix} x = Ax \tag{II-49}$$

$A$  est une matrice stable puisque les matrices  $K_p$  et  $K_v$  sont des matrices définies positives. Donc nous pouvons trouver, pour une matrice  $Q > 0$ , une matrice  $P$  symétrique définie positive satisfaisant :

$$A^T P + P A = -Q \tag{II-50}$$

Pour démontrer que le point d'équilibre  $x=0$  est stable, nous choisissons la fonction candidate de Lyapunov suivante :

$$V(x) = x^T P x \tag{II-51}$$

qui est définie positive.

La dérivée de la fonction (II-51) par rapport au temps est :

$$\dot{V}(x) = \dot{x}^T P x + x^T P \dot{x} \tag{II-52}$$

En tenant compte de l'équation (II-49), l'équation (II-52) devient :

$$\begin{aligned} \dot{V}(x) &= x^T A^T P x + x^T P A x \\ &= x^T (A^T P + P A) x \end{aligned} \tag{II-53}$$

La substitution de (II-50) dans (II-53) donne :

$$\dot{V}(x) = -x^T Q x \tag{II-54}$$

qui est une matrice définie négative, ce qui implique que le point d'équilibre  $(\tilde{q}, \dot{\tilde{q}}) = (0, 0)$  est asymptotiquement stable.

**II-1-2-5. Commande de Paden et Panja (poursuite de trajectoire) [10, 17]**

Cette commande est basée sur la théorie de Lyapunov, elle est donnée par :

$$\Gamma = M(q)\ddot{q}_d(t) + C(q, \dot{q})\dot{q}_d(t) + G(q) - K_p(q - q_d(t)) - K_v(\dot{q} - \dot{q}_d(t)) \quad (\text{II-55})$$

En remplaçant (II-55) dans le modèle dynamique (II-12), nous obtenons la dynamique suivante du robot manipulateur :

$$M(q)\ddot{\tilde{q}} + C(q, \dot{q})\dot{\tilde{q}} + K_v\dot{\tilde{q}} + K_p\tilde{q} = 0 \quad (\text{II-56})$$

où  $\tilde{q} = q - q_d(t)$ ,  $\dot{\tilde{q}} = \dot{q} - \dot{q}_d(t)$ ,  $\ddot{\tilde{q}} = \ddot{q} - \ddot{q}_d(t)$  désignent respectivement l'erreur de poursuite en position, en vitesse et en accélération. L'expression de l'erreur de poursuite en accélération est donnée par :

$$\ddot{\tilde{q}} = -M^{-1}(q)\left(C(q, \dot{q})\dot{\tilde{q}} + K_v\dot{\tilde{q}} + K_p\tilde{q}\right) \quad (\text{II-57})$$

La stabilité du système bouclé est prouvée en prenant comme fonction de Lyapunov définie positive :

$$V(\tilde{q}, \dot{\tilde{q}}, t) = \frac{1}{2}\dot{\tilde{q}}^T M(\tilde{q} + q_d(t))\dot{\tilde{q}} + \frac{1}{2}\tilde{q}^T K_p\tilde{q} \quad (\text{II-58})$$

dont la dérivée temporelle le long des trajectoires du système bouclé (II-56) est donnée par :

$$\dot{V}(\tilde{q}, \dot{\tilde{q}}, t) = \dot{\tilde{q}}^T M(\tilde{q} + q_d(t))\ddot{\tilde{q}} + \frac{1}{2}\dot{\tilde{q}}^T \dot{M}(\tilde{q} + q_d(t))\dot{\tilde{q}} + \dot{\tilde{q}}^T K_p\tilde{q} \quad (\text{II-59})$$

En substituant l'expression de l'accélération articulaire donnée par la relation (II-57) dans la relation (II-59), nous obtenons :

$$\dot{V}(\tilde{q}, \dot{\tilde{q}}, t) = -\dot{\tilde{q}}^T K_v\dot{\tilde{q}} + \frac{1}{2}\dot{\tilde{q}}^T \left( \dot{M}(\tilde{q} + q_d(t))\dot{\tilde{q}} - 2C(\tilde{q} + q_d(t), \dot{\tilde{q}}(t) + \dot{q}_d(t)) \right) \dot{\tilde{q}} \quad (\text{II-60})$$

En exploitant la propriété de la matrice  $\dot{M}(q)\dot{q} - 2C(q, \dot{q})$ , la relation (II-60) devient :

$$\dot{V}(\tilde{q}, \dot{\tilde{q}}, t) = -\dot{\tilde{q}}^T K_v\dot{\tilde{q}} \leq 0 \quad (\text{II-61})$$

Les trajectoires du système bouclé sont bornées, en utilisant le lemme de Barbalat nous pouvons montrer que le point d'équilibre  $(\tilde{q}, \dot{\tilde{q}}) = (0, 0)$  est globalement asymptotiquement stable. En effet, il est clair que si  $\lim_{t \rightarrow \infty} \dot{V}(\tilde{q}, \dot{\tilde{q}}, t) = 0$ , alors,  $\lim_{t \rightarrow \infty} \dot{\tilde{q}}(t) = 0$  (cf. (II-61)). Il nous reste à montrer que  $\lim_{t \rightarrow \infty} \tilde{q}(t) = 0$ . Sachant que  $\lim_{t \rightarrow \infty} \dot{\tilde{q}}(t) = 0$ , et si nous montrons que  $\dot{\tilde{q}}(t)$  est uniformément continue ( $\ddot{\tilde{q}}(t)$  est bornée), alors d'après le lemme de Barbalat,  $\lim_{t \rightarrow \infty} \ddot{\tilde{q}}(t) = 0$ . Ainsi, en développant la dérivée temporelle de la relation (II-61), nous remarquons que  $\ddot{\tilde{q}}(t)$  est bornée (donc  $\lim_{t \rightarrow \infty} \ddot{\tilde{q}}(t) = 0$ ). En remplaçant  $\lim_{t \rightarrow \infty} \dot{\tilde{q}}(t) = 0$  et  $\lim_{t \rightarrow \infty} \ddot{\tilde{q}}(t) = 0$  dans la relation (II-61), nous pouvons conclure que  $\lim_{t \rightarrow \infty} \tilde{q}(t) = 0$  (car  $K_p \in \mathfrak{R}^{n \times n} > 0$ ). Le seul point d'équilibre globalement asymptotiquement stable est donc  $(\tilde{q}, \dot{\tilde{q}}) = (0, 0)$ .

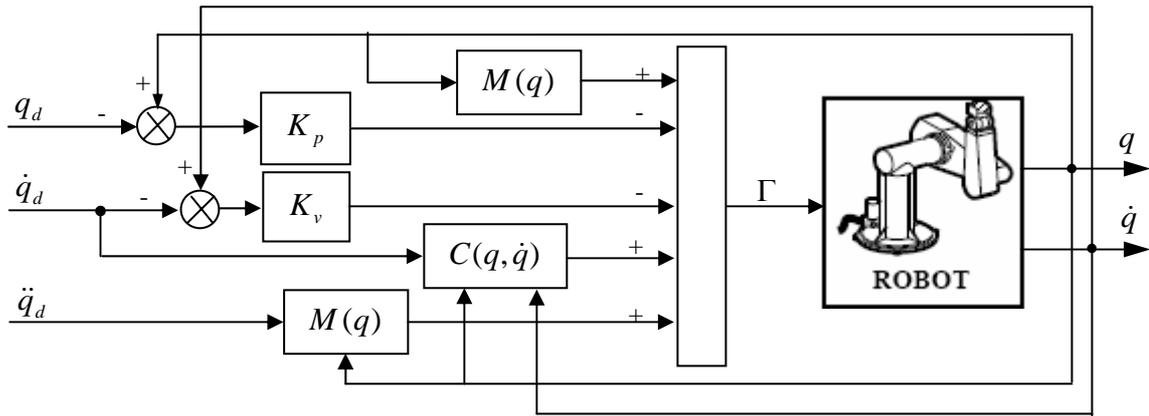


Figure II-9 : Schéma bloc de la commande de Paden et Panja.

**II-1-2-6. Loi de commande Couple Calculé (Computed torque) [1, 15]**

Cette commande a été développée au début des années soixante dix, dite aussi linéarisante puisqu'elle est considérée comme une application spéciale de la linéarisation en boucle fermée des systèmes non linéaires. Donc elle repose sur l'annulation des termes non linéaires et le découplage de chaque segment. Son utilisation pour les robots manipulateurs dépend de l'inversion de quelques termes dynamiques du robot, et pour cette raison qu'elle s'appelle aussi la dynamique inverse. Pour avoir cet objectif, la loi de commande à appliquer dans ce cas est donnée par :

$$\Gamma = M(q)\Gamma_0 + C(q, \dot{q})\dot{q} + g(q) \tag{II-62}$$

La figure II-10 donne le principe de cette commande.

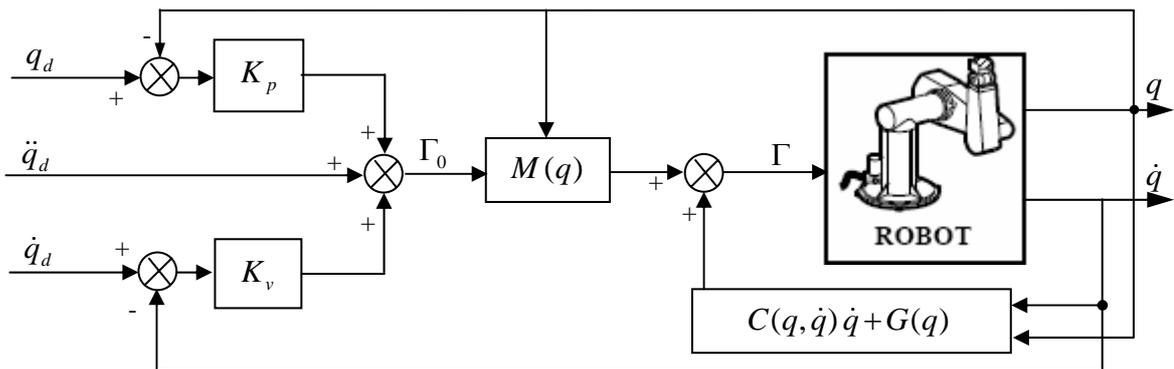


Figure II-10 : Schéma bloc de la commande du couple calculé.

En appliquant cette loi de commande sur le modèle (II-12) du robot, nous aurons l'équation en boucle fermée suivante :

$$M(q)(\ddot{q} - \Gamma_0) = 0 \tag{II-63}$$

En prenant en considération que la matrice  $M(q)$  est régulière (inversible), il s'ensuit  $n$  systèmes linéaires découplés :

$$\ddot{q} = \Gamma_0 \tag{II-64}$$

où  $\Gamma_0$  est un terme auxiliaire de commande avec unité d'accélération, qui peut être défini par un terme de compensation dynamique linéaire  $K(p)$  en fonction de  $p$  (opérateur de Laplace), de type  $PD$  ou  $PID$  et est exprimé ainsi :

$$\Gamma_0 = \ddot{q}_d - K(p)\tilde{q} \quad (\text{II-65})$$

En substituant l'expression de  $\Gamma_0$  donnée par (II-65), l'équation (II-64) devient :

$$\begin{aligned} \ddot{q} &= \ddot{q}_d - K(p)\tilde{q} \\ \ddot{\tilde{q}} + K(p)\tilde{q} &= 0 \end{aligned} \quad (\text{II-66})$$

Le choix typique de  $K(p)$  est :

$$K(p) = K_v p + K_p \quad (\text{II-67})$$

c'est-à-dire un compensateur proportionnel-dérivé ( $PD$ ). Les équations (II-66) et (II-67) donnent l'équation d'erreur du second ordre suivante :

$$\ddot{\tilde{q}} + K_v \dot{\tilde{q}} + K_p \tilde{q} = 0 \quad (\text{II-68})$$

Avec  $K_p, K_v \in \mathfrak{R}^{n \times n}$  des matrices diagonales définies positives donc le système bouclé devient linéaire découplé.

L'équation d'erreur (II-68) est asymptotiquement stable par un choix convenable des matrices  $K_p$  et  $K_v$ . Pour montrer cela, nous pouvons réécrire le système (II-68) sous forme d'espace d'état suivant :

$$\dot{x} = \begin{pmatrix} 0 & I \\ -K_p & -K_v \end{pmatrix} x = A x \quad (\text{II-69})$$

où  $x \in \mathfrak{R}^{2n}$  est un vecteur défini comme suit :  $x^T = (\tilde{q}^T, \dot{\tilde{q}}^T)^T$  et  $\dot{x} \in \mathfrak{R}^{2n}$  est sa dynamique défini ainsi  $\dot{x}^T = (\dot{\tilde{q}}^T, \ddot{\tilde{q}}^T)^T$ .

Nous pouvons remarquer que  $A$  est stable puisque les matrices  $K_v$  et  $K_p$  sont définies positives. Donc nous pouvons trouver, pour une matrice  $Q > 0$ , une matrice  $P = P^T > 0$  symétrique définie positive satisfaisant :

$$A^T P + P A = -Q \quad (\text{II-70})$$

Choisissons une fonction de Lyapunov de la forme suivante :

$$V(x) = x^T P x \quad (\text{II-71})$$

qui est définie positive.

La dérivée de (II-71) est :

$$\begin{aligned} \dot{V}(x) &= \dot{x}^T P x + x^T P \dot{x} \\ &= x^T A^T P x + x^T P A x \\ &= x^T (A^T P + P A) x \\ &= -x^T Q x \end{aligned} \quad (\text{II-72})$$

qui est une fonction définie négative. Donc, l'équation (II-68) est globalement asymptotiquement stable. C'est-à-dire pour n'importe quelle position initiale, l'erreur de poursuite va converger asymptotiquement.

La commande du couple calculé est une bonne approche pour commander les robots manipulateurs lorsque les paramètres sont parfaitement connus et elle nécessite un calculateur suffisamment puissant. Cependant, les paramètres du robot sont toujours connus partiellement donc les performances du système n'atteignent jamais l'idéal.

**II-1-2-7. Commande de Slotine et Li (poursuite de trajectoire) [1, 16]**

Dans cette commande le robot doit suivre une trajectoire prédéfinie par les variables articulaires désirées  $(q_d(t), \dot{q}_d(t), \ddot{q}_d(t))$ . Pour cela la loi de commande est donnée par :

$$\Gamma = M(q)\ddot{q}_r + C(q, \dot{q})\dot{q}_r + G(q) - K_d(\dot{q} - \dot{q}_r) \tag{II-73}$$

où :

$$K_d > 0, \quad \tilde{q} = q - q_d, \quad \dot{q}_r = \dot{q}_d - \Lambda \tilde{q} \tag{II-74}$$

La figure II-11 présente le principe de cette approche.

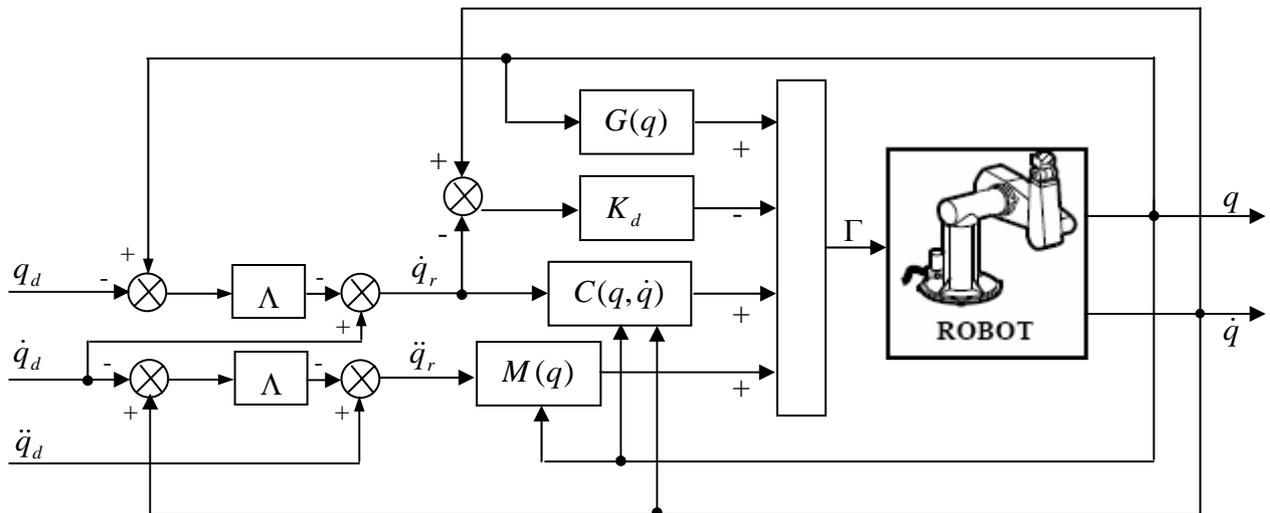


Figure II-11 : Schéma bloc de la commande de Slotine et Li.

En remplaçant (II-73) dans le modèle dynamique (II-12), nous obtenons la dynamique suivante :

$$M(q)\ddot{q} + C(q, \dot{q})\dot{q} + G(q) = M(q)\ddot{q}_r + C(q, \dot{q})\dot{q}_r + G(q) - K_d(\dot{q} - \dot{q}_r) \tag{II-75}$$

Soit ;  $s = \dot{q} - \dot{q}_r = \dot{\tilde{q}} + \Lambda \tilde{q}$   $\Lambda > 0$  et pour  $s \rightarrow 0 \Rightarrow \tilde{q} \rightarrow 0$  et  $\dot{\tilde{q}} \rightarrow 0$ . Notons aussi que  $\dot{q}_r$  et  $\ddot{q}_r$  ne sont pas en fonction de l'accélération  $\ddot{q}$ . L'équation (II-75) s'écrit aussi :

$$M(q)\dot{s} + C(q, \dot{q})s + K_d s = 0 \tag{II-76}$$

Soit la fonction :

$$V(q, \dot{q}, q_d(t), \dot{q}_d(t)) = \frac{1}{2} s^T M(q) s \tag{II-77}$$

alors,

$$\begin{aligned}
\dot{V} &= s^T M(q) \dot{s} + \frac{1}{2} s^T \dot{M}(q) s \\
&= s^T [-C(q, \dot{q}) s - K_d s] + \frac{1}{2} s^T \dot{M}(q) s \\
&= -s^T K_d s + \frac{1}{2} s^T [\dot{M}(q) - 2C(q, \dot{q})] s \\
&= -s^T K_d s < 0
\end{aligned} \tag{II-78}$$

La convergence de  $\tilde{q}$  vers zéro est démontrée par l'utilisation du lemme de Barbalat, puisque le théorème de LaSalle ne peut pas être utilisé dans ce cas, car la fonction  $V$  n'est pas définie uniquement en fonction de l'état  $q, \dot{q}$ .

Si  $V(t)$  est une fonction continue, convergente lorsque  $t \rightarrow \infty$  et si  $\frac{dV}{dt}$  est uniformément continue, alors  $\dot{V} \rightarrow 0$ .

Pour montrer que  $\frac{dV}{dt}$  est continue, il suffit de montrer que  $\frac{d^2V}{dt^2}$  est bornée. De l'équation (II-77) et (II-78), il est clair que  $V$  est bornée et  $s$  est bornée aussi. De l'équation (II-76), nous concluons que  $\dot{s}$  est bornée. Par ailleurs,  $\ddot{V}$  est égale à :

$$\ddot{V} = -2s^T K_d \dot{s} \tag{II-79}$$

qui est bornée, ce qui implique que  $\dot{V}$  est uniformément continue. Donc le lemme de Barbalat nous permet de conclure que lorsque  $\dot{V} \rightarrow 0 \Rightarrow \tilde{q} \rightarrow 0$ . Alors, le système est asymptotiquement globalement stable.

## II-2. Commande en effort [3, 6, 9, 15]

Lorsque l'effecteur en mouvement entre en contact avec une surface rigide ou déformable, la chaîne cinématique du robot est fermée sur l'environnement et, du fait de la raideur de l'ensemble, de faibles variations de positions du point terminal peuvent induire des efforts élevés.

Lorsque le robot s'écarte de la trajectoire de consigne, la loi de commande en mouvement tend à minimiser l'erreur, indépendamment de l'origine de la perturbation. Si cette erreur est due à l'interaction entre le robot et l'environnement, l'action de la loi de commande en mouvement pour s'y opposer tend à accroître les efforts d'interaction, ceci d'autant plus que la raideur de l'environnement est élevée.

Pour pouvoir contrôler l'interaction robot-environnement on doit distinguer alors la commande en effort de la commande en mouvement. Si l'effecteur n'étant pas en interaction avec un quelconque objet de son environnement, il ne peut exercer d'effort ; la commande appropriée dans ce cas est une commande en position et si l'effecteur est en contact direct avec une pièce fixe, aucun mouvement n'est alors autorisé pour ne pas endommager le mécanisme et son environnement, dans ce

cas, l'effecteur peut exercer un effort dans n'importe quelle direction. La seule commande autorisée dans ces conditions est alors une commande en effort.

On ne peut pas contrôler simultanément une force et un déplacement suivant une même direction. Une description intermédiaire consiste à définir des tâches compliantes en relation avec les degrés de liberté de l'espace opérationnel du robot : l'effort est spécifié selon certaines directions orthogonales, constituant le sous-espace des efforts, et les déplacements selon des directions complémentaires, constituant le sous-espace des positions. On définit l'espace de compliance comme la somme de ces deux espaces conditionnés par la nature de la tâche à réaliser et par les modèles du robot et de l'environnement.

Les structures de commande en effort sont nombreuses en robotique : depuis la fin des années 70, plusieurs travaux s'intéressent à celles-ci, mais peu de recherches abordent l'étude de leur stabilité. Chacune de ces méthodes étant basée sur une approche classique puis dynamique.

On peut classer la commande en effort en deux catégories selon les consignes de commande :

- Aucune consigne d'effort désiré n'est présente dans la structure de commande.
- Une consigne d'effort désiré permet l'asservissement de l'effort exercé à cette valeur, grâce là encore à la présence ou non d'un capteur d'effort.

### II-2-1. Compliance passive

La compliance passive repose sur l'utilisation d'une structure mécanique déformable sous l'action des efforts de contact. La variation de la géométrie de cette structure corrige en fait les erreurs de positionnement du robot. L'utilisation d'un tel mécanisme est généralement propre à la tâche à effectuer, ce qui constitue son principal inconvénient : après détermination de la matrice de raideur propre à la tâche (prenant en compte la raideur équivalente du robot et de son environnement), le système mécanique présentant cette compliance est réalisé ; la position du centre de compliance est fixée une fois pour toutes et doit être adaptée à la pièce à insérer. La structure la plus connue est le RCC (*Remote Center Compliance*) [9, 25] utilisée pour l'insertion de pièces dans des alésages.

La relation entre les efforts et les déformations est donnée par :

$$dX = C F \quad (\text{II-80})$$

où  $F = \begin{bmatrix} f^T & m^T \end{bmatrix}^T$ , représente le torseur des efforts dont la résultante est une force  $f$  et un moment  $m$ .  $dX = \begin{bmatrix} d^T & \delta^T \end{bmatrix}^T$ , est le vecteur des déplacements différentiels en position  $d$  et en orientation  $\delta$ , et  $C \in \mathfrak{R}^{6 \times 6}$  est la matrice de compliance. Elle est diagonale pour un point  $O_C$  appelé « centre de compliance » : l'application d'une force passant par  $O_C$  suivant une direction donnée se traduit par une translation pure dans cette direction ; l'application d'un couple se traduit par une rotation autour d'un axe passant par  $O_C$ .

La figure II-12 présente le schéma de principe d'un tel dispositif.

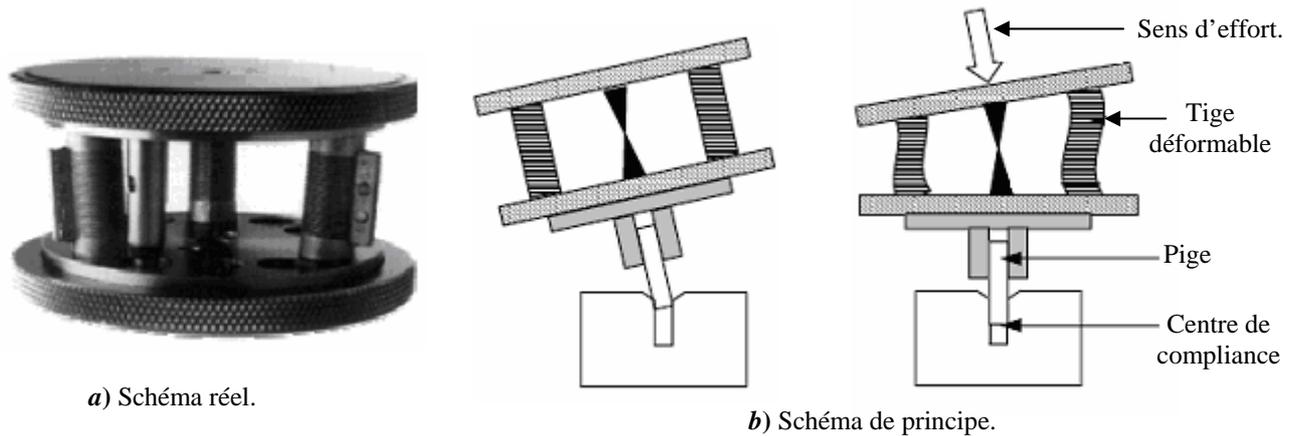


Figure II-12: Exemple de principe du RCC (ATI Industrial Automation).

L'avantage d'un tel système est d'augmenter la précision dans les insertions en des temps très courts sans nécessiter de stratégies complexes.

### II-2-2. Commande par raideur active

Dans cette approche, le robot est considéré comme un ressort dont la raideur est programmable, pour répondre aux besoins de la tâche à réaliser. Il est ainsi possible de contrôler la compliance du manipulateur. La commande est réalisée dans l'espace articulaire et permet de programmer un ressort opérationnel par rapport à un repère donné.

L'équation de la raideur est donnée par :

$${}^c F_c = K_c {}^c dX_c \quad (\text{II-81})$$

où  $K_c$  est la matrice de raideur désirée, diagonale par rapport au repère  $R_c$ .  $K_c \in \mathcal{R}^{6 \times 6}$  dans le cas général. L'effort  ${}^c F_c$  et le déplacement différentiel  ${}^c dX_c$  étant toujours calculés et exprimés dans le repère  $R_c$ . Les relations entre les couples, forces et déplacements sont données par :

$$\Gamma = J^T F \quad (\text{II-82})$$

$$dX = J dq \quad (\text{II-83})$$

avec  $F \equiv {}^c F_c$ ,  $dX \equiv {}^c dX_c$  et  $\Gamma$  représente le vecteur des couples des actionneurs.  $J$  est la matrice jacobienne du robot.

A partir des équations (II-81), (II-82) et (II-83), nous obtenons :

$$\Gamma = J^T K_c J dq = K_p dq \quad (\text{II-84})$$

où  $K_p$  définit la matrice de raideur dans l'espace articulaire correspondant à  $K_c$ . Elle n'est pas diagonale mais symétrique : elle détermine les gains proportionnels des asservissements dans l'espace articulaire.

Le vecteur des couples des actionneurs est donné par :

$$\Gamma = K_p (q_d - q) + K_v (\dot{q}_d - \dot{q}) + Q \quad (\text{II-85})$$

avec :  $Q$  est un vecteur représentant les forces de compensation de la gravité et  $K_v$  est une matrice qui peut être assimilée à une matrice d'amortissement.

La figure II-13 représente le schéma de principe de cette commande.

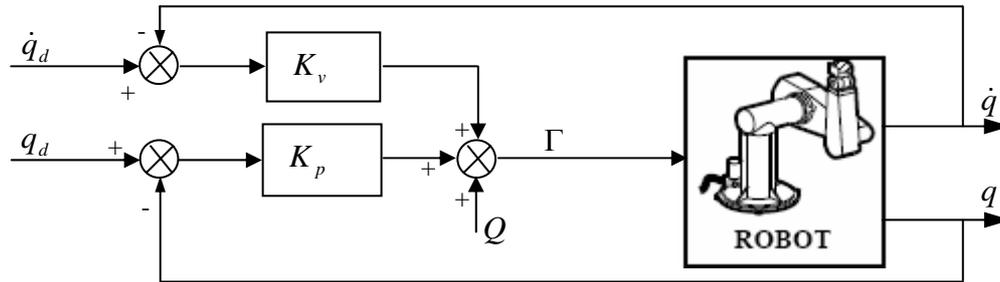


Figure II-13 : Schéma bloc de la commande par raideur active.

Dans le cas où l'on désirerait une consigne d'effort désiré  $F_d$ , il est possible d'ajouter à la commande classique par raideur active présentée précédemment, une boucle en effort en utilisant un capteur d'effort. Le couple de commande des actionneurs contient alors un terme supplémentaire (terme proportionnel et prédictif d'effort) par rapport à la relation (II-85), et le nouveau vecteur des couples des actionneurs sera :

$$\Gamma' = \Gamma + J^T [F_d + K_f (F_d - F)] \tag{II-86}$$

où  $K_f$  désigne le gain en effort et où  $F$  représente l'effort mesuré égal à :

$$F = K_E dX \tag{II-87}$$

avec  $K_E$  étant la raideur effective de l'environnement et du capteur, et  $dX$  est la déformation due au contact.

Dans ces conditions, le schéma de principe de la commande par raideur active est donné par la figure II-14.

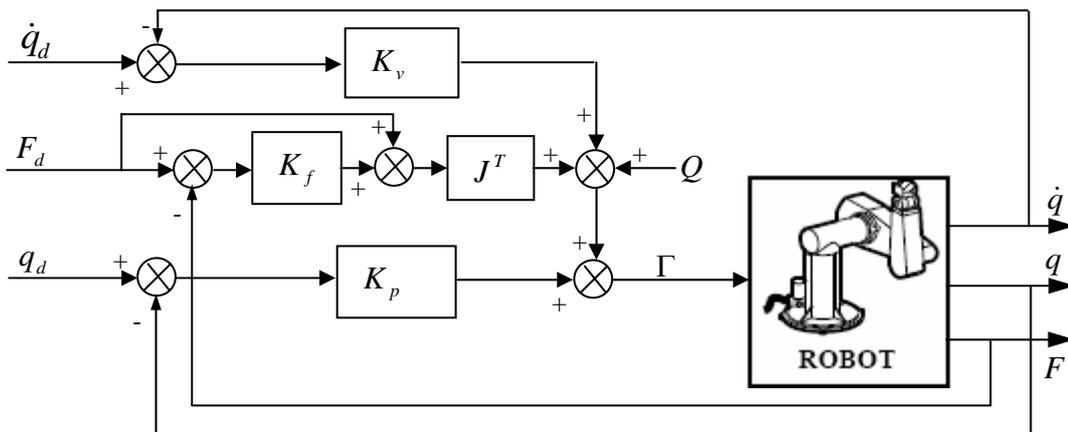


Figure II-14 : Schéma bloc d'une commande par raideur active avec mesure d'effort.

Cette méthode présente un avantage est qu'elle est relativement simple à implanter. La matrice  $K_p$  est modifiable en ligne pour adapter le comportement du robot aux contraintes de la tâche.

**II-2-3. Commande en impédance**

Les structures de commande en impédance ont pour principe non pas la commande directe de l'effort exercé ou celle de la position, mais celle de la relation entre cet effort et la position de l'effecteur, donc de l'impédance de sortie associée au robot [7]. Cette relation peut être formulée simplement des deux manières suivantes dans l'espace de Laplace :

$$F(p) = Z(p) dX(p) \tag{II-88}$$

ou  $F(p) Z(p)^{-1} = dX(p) \tag{II-89}$

avec  $dX$  est un déplacement opérationnel à partir d'une position nominale  $X_0$  et  $F$  est la force produite en réponse à ce déplacement.

Dans ce cas le robot est considéré comme un système mécanique (généralement équivalent à un système masse-amortissement-ressort du deuxième ordre) d'impédance  $Z$  variable en fonction des différentes phases de la tâche. Le choix de cette impédance est :

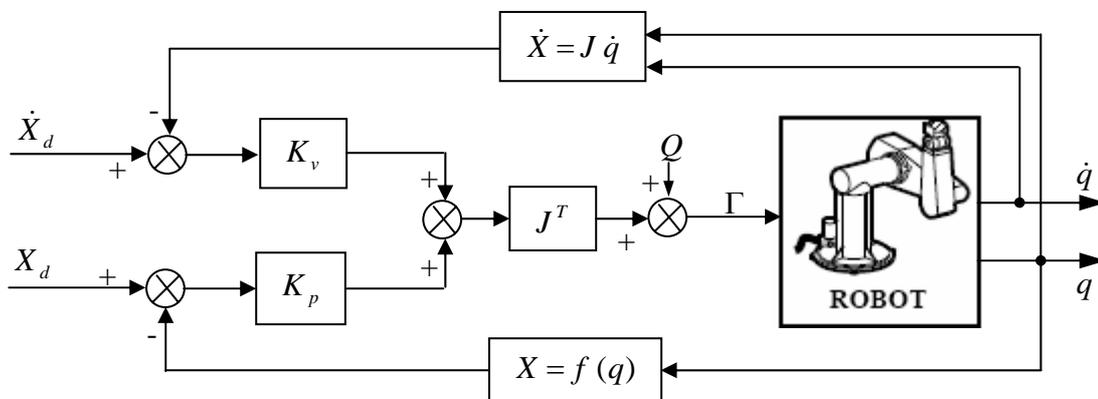
$$Z(p) = \Lambda p^2 + K_v p + K_p \tag{II-90}$$

Où  $\Lambda, K_v$  et  $K_p$  sont respectivement : la matrice d'inertie, matrice d'amortissement et la matrice de raideur. La figure II-15 présente le principe de la commande en impédance sans mesure d'effort et qui ne fait pas intervenir le modèle dynamique du robot.

Dans ce cas la loi de commande est donnée par la relation suivante :

$$\Gamma = J^T [K_v (\dot{X}_d - \dot{X}) + K_p (X_d - X)] + Q \tag{II-91}$$

Réellement  $K_p$  et  $K_v$  désignent les gains proportionnels et dérivés dans l'espace opérationnel que l'on peut interpréter comme les matrices de raideur et d'amortissement du robot. Le vecteur  $Q$  représente les forces de compensation de la gravité. Cette commande place le centre de compliancance au point  $X_d$ .



**Figure II-15 :** Schéma de principe de la commande en impédance.

Il existe bon nombre de variantes de la commande en impédance [14, 25], notamment celles où l'on dispose d'un capteur d'effort.

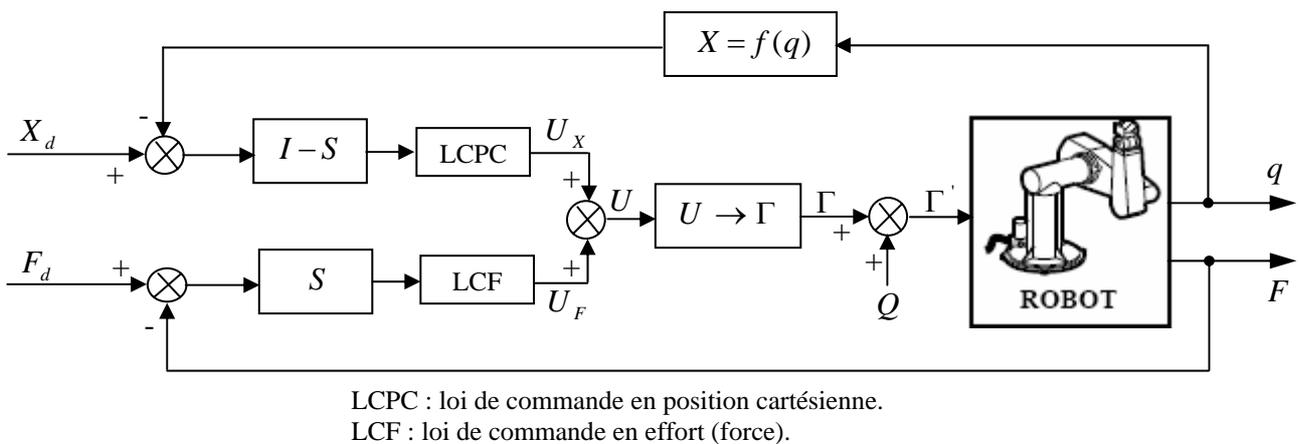
**II-2-4. Commande hybride force/position**

La commande hybride force/position permet un contrôle en position et un contrôle en effort dits parallèles [21]. Les informations d'effort et de position sont utilisées simultanément et les directions contraintes en effort sont commandées en position et celles contraintes en position sont commandées en effort : une matrice de sélection diagonale notée  $S$  permet de désigner le type de commande (position ou effort) à affecter à chaque direction du repère de compliance, assurant de ce fait l'exclusivité de l'une ou l'autre des commandes. Généralement cette matrice est donnée par :

$$S = \text{Diag}(s_1 \quad s_2 \quad \dots \quad s_6) \tag{II-92}$$

où  $s_i = 1$  si le  $i^{\text{ème}}$  degré de liberté de l'espace cartésien (opérationnel) est commandé en effort, à 0 s'il est commandé en position.

Le schéma de principe de cette commande est donné par la figure II-16 : il met en évidence l'existence de deux boucles d'asservissement, l'une en position pour les directions contraintes en effort, l'autre en effort pour celles contraintes en position.



**Figure II-16 :** Schéma de principe de la commande hybride force/position.

**II-2-4-1. Différentes classes de commandes hybrides**

Il existe diverses variantes et implémentations du schéma de principe proposé par Raibert [21]. On les différencie en fonction du type de la grandeur  $U$  qui est liée aux grandeurs  $U_x$  et  $U_F$  utilisées pour réaliser la sommation avant transformation en couple articulaire :

**A/ Somme de couples :**

C'est la structure initiale proposée par Raibert [21]. Les grandeurs  $U_x$  et  $U_F$  sont directement les couples articulaires : pour la position, une transformation par la jacobienne inverse et un

correcteur adéquat articulaire suffisent ; pour l'asservissement en effort, la transformation est obtenue par utilisation de la jacobienne transposée et d'un correcteur (voir figure II-17-a).

Une autre variante de cette structure a été proposée par Zhang [27] avec boucle de retour articulaire (voir figure II-17-b).

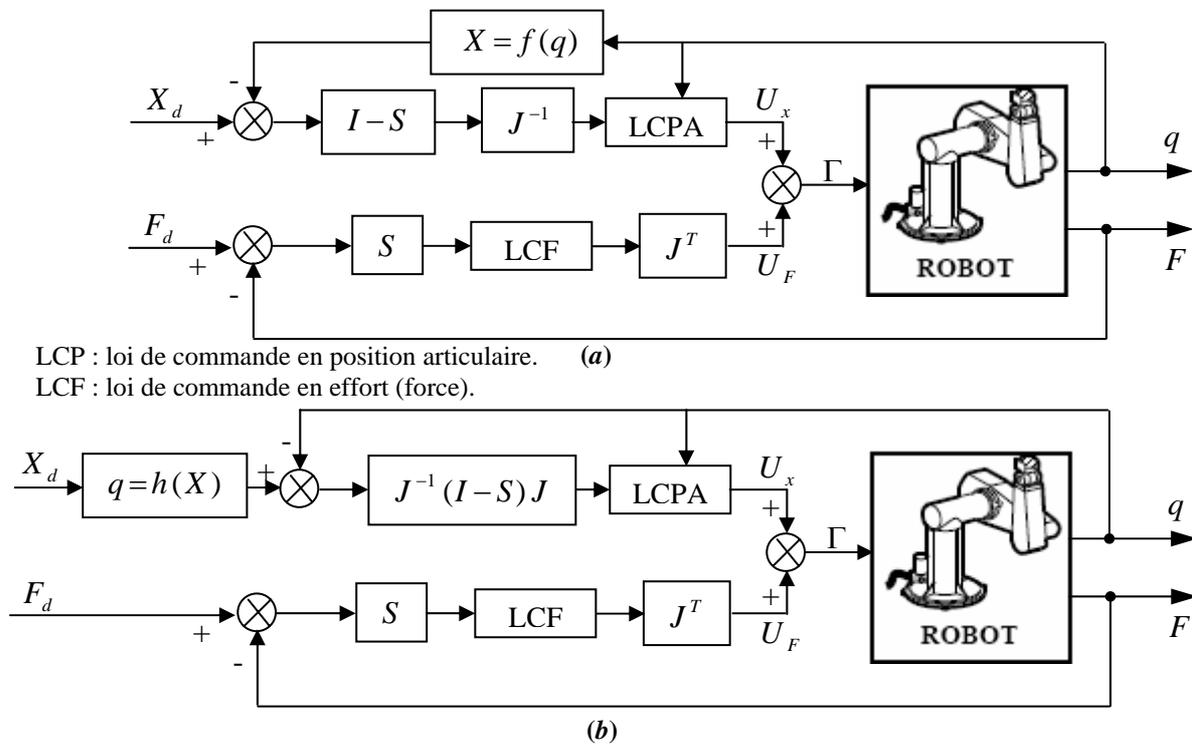


Figure II-17 : Schéma de la commande hybride force/position avec somme de couples articulaires : *a*) Structure initiale de Raibert [21], *b*) Structure de Zhang [27].

**B/ Somme des erreurs de positions cartésiennes :**

$U$  est la somme des erreurs de positions cartésiennes issues de la boucle d'effort et de celle de position [20, 22] ; la transformation en couple articulaire est réalisée par l'intermédiaire de la matrice jacobienne inverse puis d'un correcteur articulaire de type PID par exemple (voir figure II-18-a). Le contrôleur en effort est généralement une matrice homogène à l'inverse de la raideur estimée.

**C/ Somme des efforts :**

$U$  est la somme des erreurs en effort [12]. Un correcteur cartésien permet de transformer l'erreur de position en erreur d'effort ; en général, la matrice d'inertie du robot est introduite dans la commande en position pour permettre un découplage non linéaire. La somme est ensuite convertie en un couple articulaire par utilisation de la jacobienne transposée.

Le schéma de principe de cette structure est représenté sur la figure II-18-b.

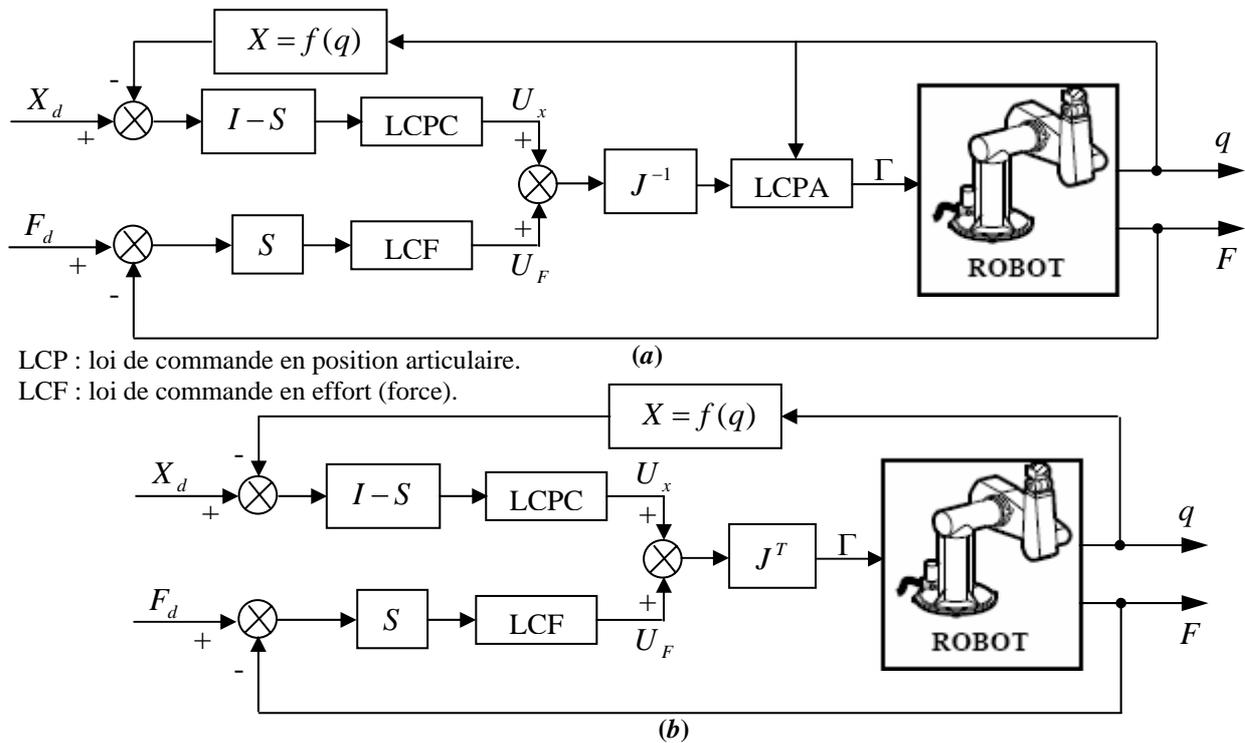


Figure II-18 : Schéma de la commande hybride force/position avec somme des :  
 a) Erreurs de positions cartésiennes [22], b) Efforts [12].

**II-2-5. Commande hybride externe**

Cette commande, appelée encore commande en série ou en cascade, permet la contribution selon une même direction d'un terme en effort et d'un autre en position. L'idée maîtresse de cette commande réside dans le fait qu'une variation de l'effort se traduit par un déplacement (en modélisant classiquement le contact du robot avec l'environnement par un ressort dans le cas d'un correcteur d'effort le plus simple possible). Il y a donc asservissement de l'effort mesuré à la valeur de consigne par conversion de l'erreur d'effort en un déplacement relatif de l'organe terminal qui vient corriger une consigne initiale de position. Le déplacement résultant permet donc d'exercer l'effort désiré sur l'environnement.

La commande hybride externe ne nécessite pas que les commandes (en effort ou/et en position) appliquées selon les directions soient exclusives. Cependant, une matrice de sélection  $S$  est toujours ajoutée dans la boucle d'effort permettant la sélection des directions commandées en effort. Par défaut donc, toute direction non commandée en effort (donc contrainte en effort) sera asservie en position à la commande  $X_d$ .

La figure II-19 présente le principe de cette commande.

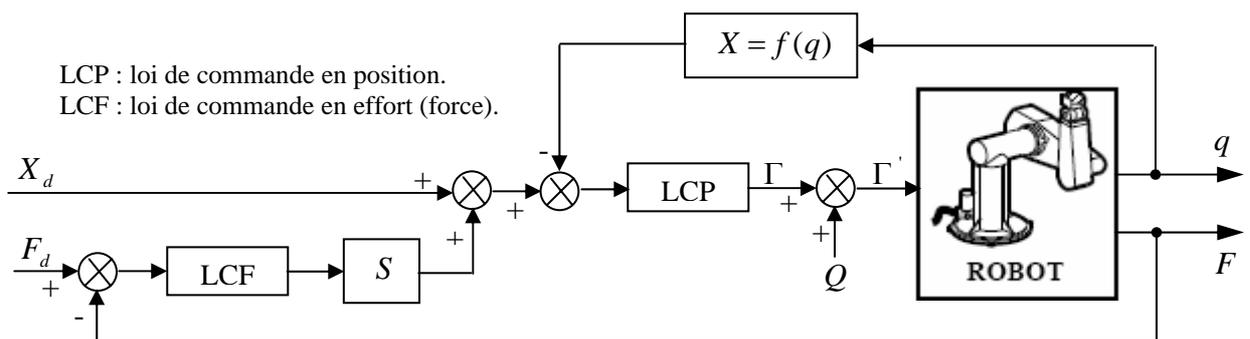
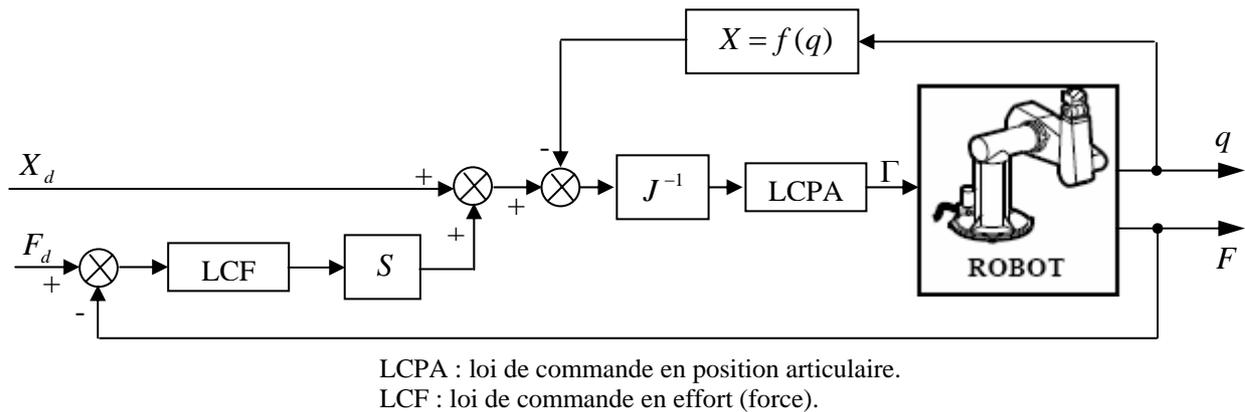


Figure II-19 : Schéma de principe de la commande hybride externe.

Il existe diverses variantes de la commande externe qui peuvent être regroupées en deux familles. On les distingue généralement en fonction de l'espace dans lequel est réalisé l'asservissement en position (espace opérationnel ou articulaire) [18] :

**a) Asservissement de position cartésien (opérationnel)**

Ce schéma de structure est celui initial proposé par De Schutter [4]. La boucle de retour en position est réalisée par l'intermédiaire du modèle géométrique du robot et la transformation de l'incrément de position en couple articulaire est réalisée via la matrice jacobienne inverse et un correcteur articulaire. Le principe de cet asservissement est donné par le schéma de la figure II-20.

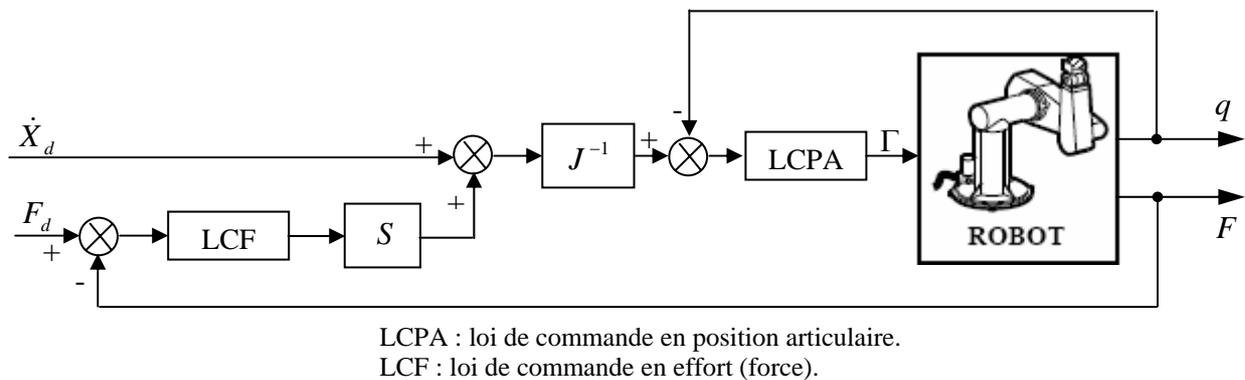


**Figure II-20** : Schéma de principe de l'asservissement cartésien avec commande en position cartésienne.

**b) Asservissement de position articulaire**

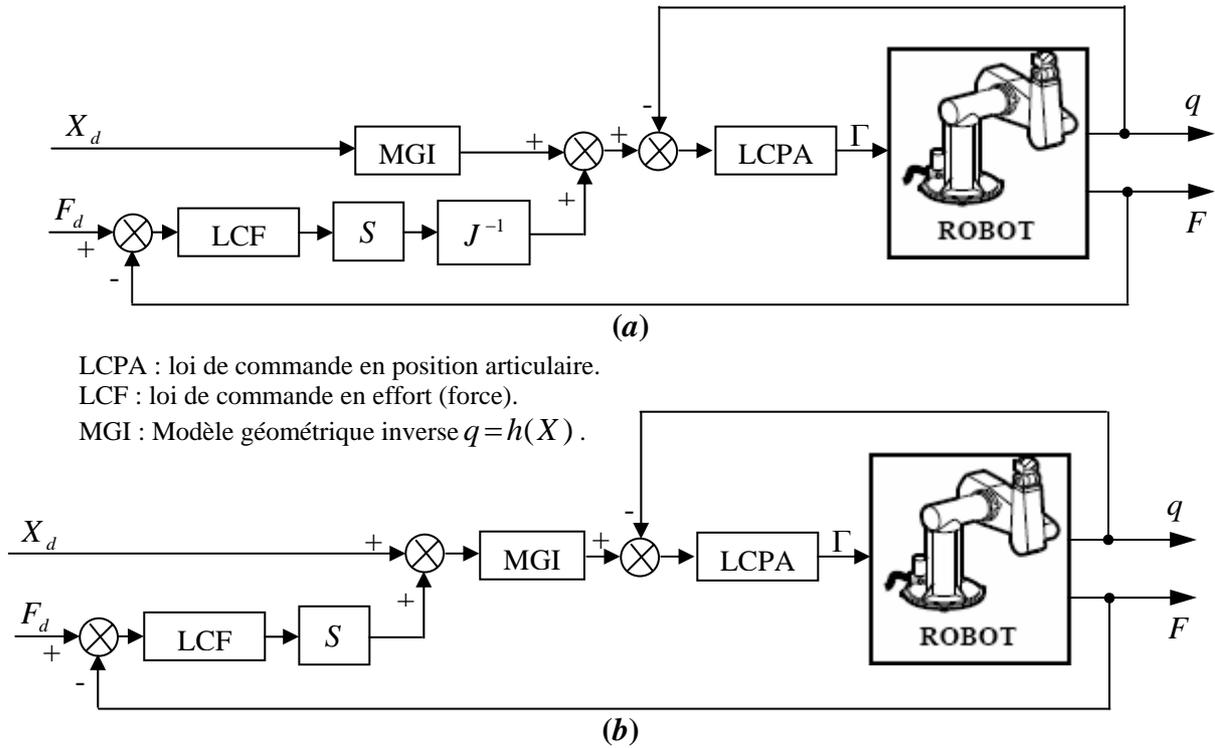
Dans ce cas, plusieurs structures peuvent être envisagées. La consigne peut ne plus être la position mais la vitesse opérationnelle comme décrit sur la figure II-21 [4].

On peut d'autre part, comme pour la commande hybride, sommer les consignes soit dans l'espace articulaire, soit dans l'espace cartésien. C'est cette dernière structure qui est la plus rencontrée dans la littérature [5, 19, 20, 24]: la consigne opérationnelle issue de la sommation de la consigne d'effort pure et de position pure est transformée en consigne articulaire par utilisation du modèle géométrique inverse.



**Figure II-21** : Schéma de principe de l'asservissement articulaire avec commande en vitesse cartésienne.

La figure II-22 donne le principe de cette commande dans le cas où on travaille dans l'espace articulaire et dans le cas de l'espace opérationnel.



LCPC : loi de commande en position articulaire.  
 LCF : loi de commande en effort (force).  
 MGI : Modèle géométrique inverse  $q = h(X)$ .

**Figure II-22 :** Schéma de principe de la commande externe avec asservissement articulaire :  
 a. Transformation indépendante des consignes.  
 b. Transformation globale des consignes.

**II-3. Applications à la commande d'un bras manipulateur Puma 600**

Dans ce paragraphe nous allons procéder à l'application, par simulation en utilisant le modèle dynamique du robot Puma 600 développé dans le chapitre I, de quelques lois de commandes classiques présentées dans ce chapitre à savoir la commande en position *Point à Point* et la commande de poursuite de trajectoire *Couple calculé* de type *PD (Computed torque)*.

Toutes les simulations sont faites par l'utilisation de l'interface simulink sous Matlab.

**II-3-1. Commande point à point**

Pour mettre en œuvre cette commande, nous avons choisi comme application la commande point à point de type *PD* (§ II-1-2-1).

Les positions désirées utilisées dans notre simulation sont :

$$q_d = \begin{bmatrix} q_{d1} \\ q_{d2} \\ q_{d3} \end{bmatrix} = \begin{bmatrix} 1.5 \\ 1.6 \\ 1.7 \end{bmatrix}$$

Les paramètres convenables obtenus par simulation, pour les gains, sont :

$$K_p = \begin{bmatrix} 3500 & 0 & 0 \\ 0 & 3500 & 0 \\ 0 & 0 & 3500 \end{bmatrix} ; \quad K_v = \begin{bmatrix} 250 & 0 & 0 \\ 0 & 350 & 0 \\ 0 & 0 & 190 \end{bmatrix}$$

La figure II-23, donne la réalisation du schéma bloc de la commande PD utilisée en simulation.

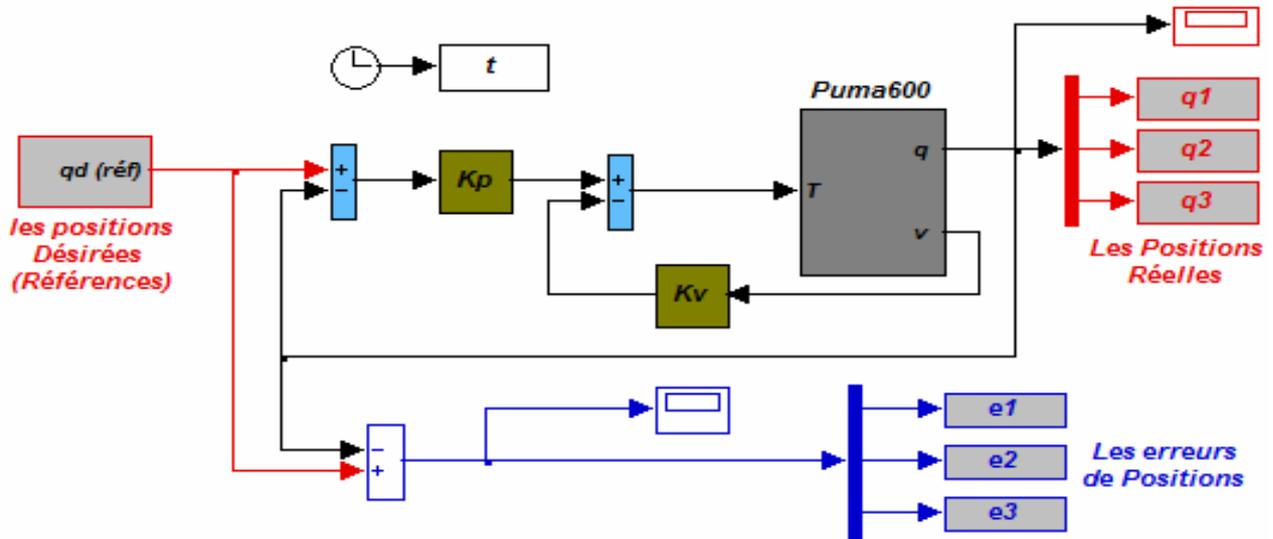


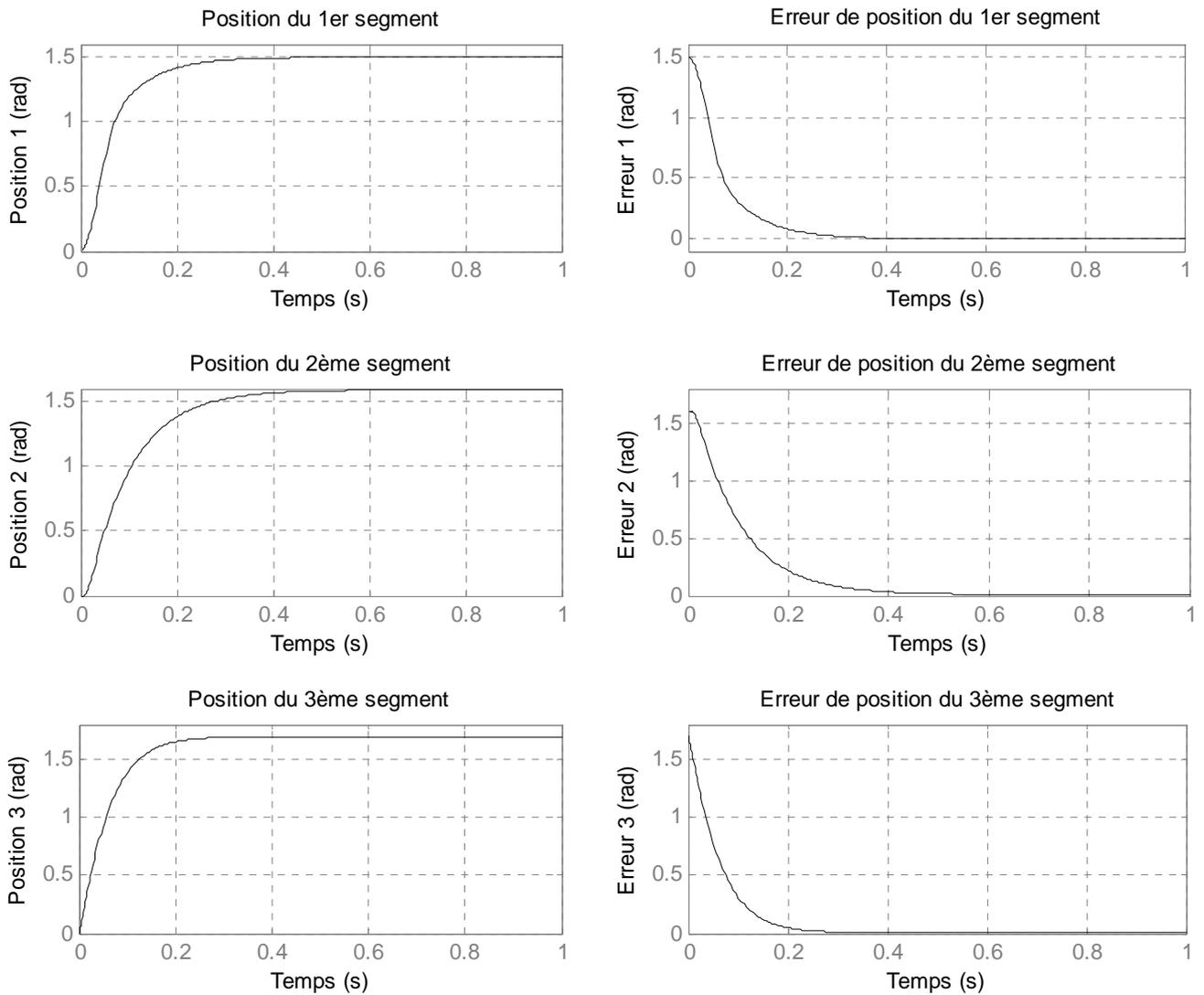
Figure II-23 : Schéma d'implémentation de la commande point à point de type PD sous Matlab/simulink

Les résultats de simulation obtenus par cette commande sont résumés sur la figure II-24.

**Analyse des résultats**

D'après les résultats présentés dans la figure II-24, on constate que les différents segments du robot atteignent les positions désirées dans des intervalles très courts cela signifie que les valeurs des gains ont été bien choisies. Les erreurs de position convergent aussi vers zéros indiquent que cette loi de commande est stable.

L'inconvénient présenté dans cette méthode est les grandes valeurs des gains utilisés qui peuvent provoquer des problèmes sur les actionneurs du robot lorsqu'une perturbation se présente.



**Figure II-24:** Les positions mesurées et les erreurs de positions pour la commande Point à Point de type PD

**II-3-2. Commande en poursuite de trajectoire (Couple Calculé)**

Dans ce cas, nous avons choisi comme application une commande Couple Calculé de type PD.

Pour les trajectoires désirées utilisées dans notre simulation, nous avons considéré une trajectoire de référence, assurant une continuité en position, vitesse et accélération, donnée par :

$$\begin{cases} q_{d1} = 3 + 6 (\sin t + \sin 2t) \\ q_{d2} = 2 + 4 (\cos t + \sin 2t) \\ q_{d3} = 4 + 5 (\sin t + \cos 2t) \end{cases}$$

Les paramètres convenables obtenus par simulation, pour les gains, sont :

$$K_p = \text{diag}(300 \ 300 \ 300); \ K_v = \text{diag}(35 \ 35 \ 35)$$

La figure II-25, représente la réalisation du schéma bloc de la commande Couple Calculé type PD utilisée en simulation.

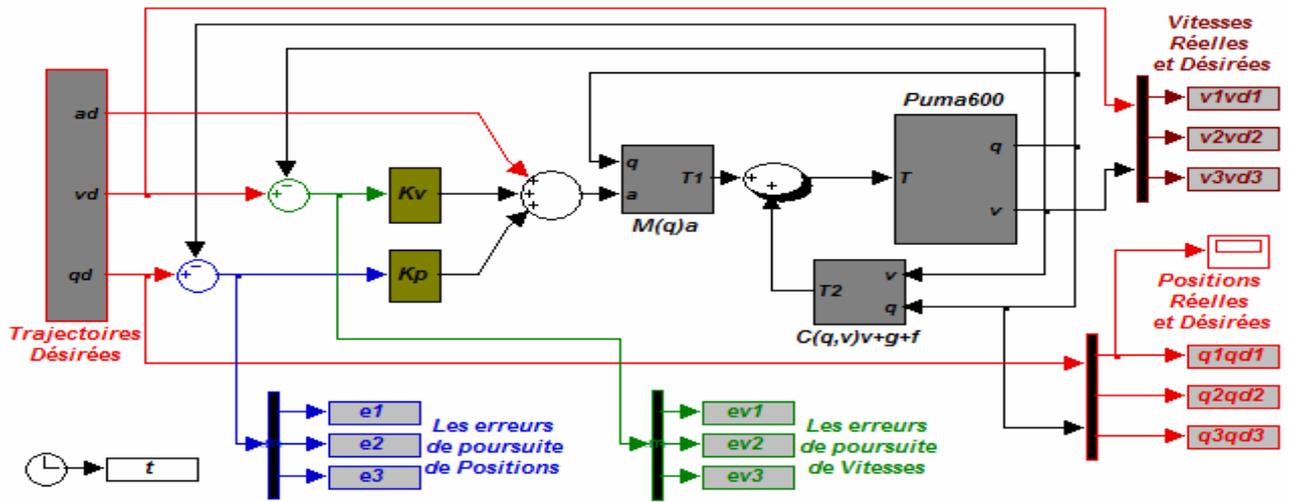


Figure II-25 : Schéma d'implémentation de la commande Couple Calculé de type PD sous Matlab/Simulink

Le comportement du robot en poursuite de la trajectoire désirée, pour cette commande, est résumé dans la figure II-26.

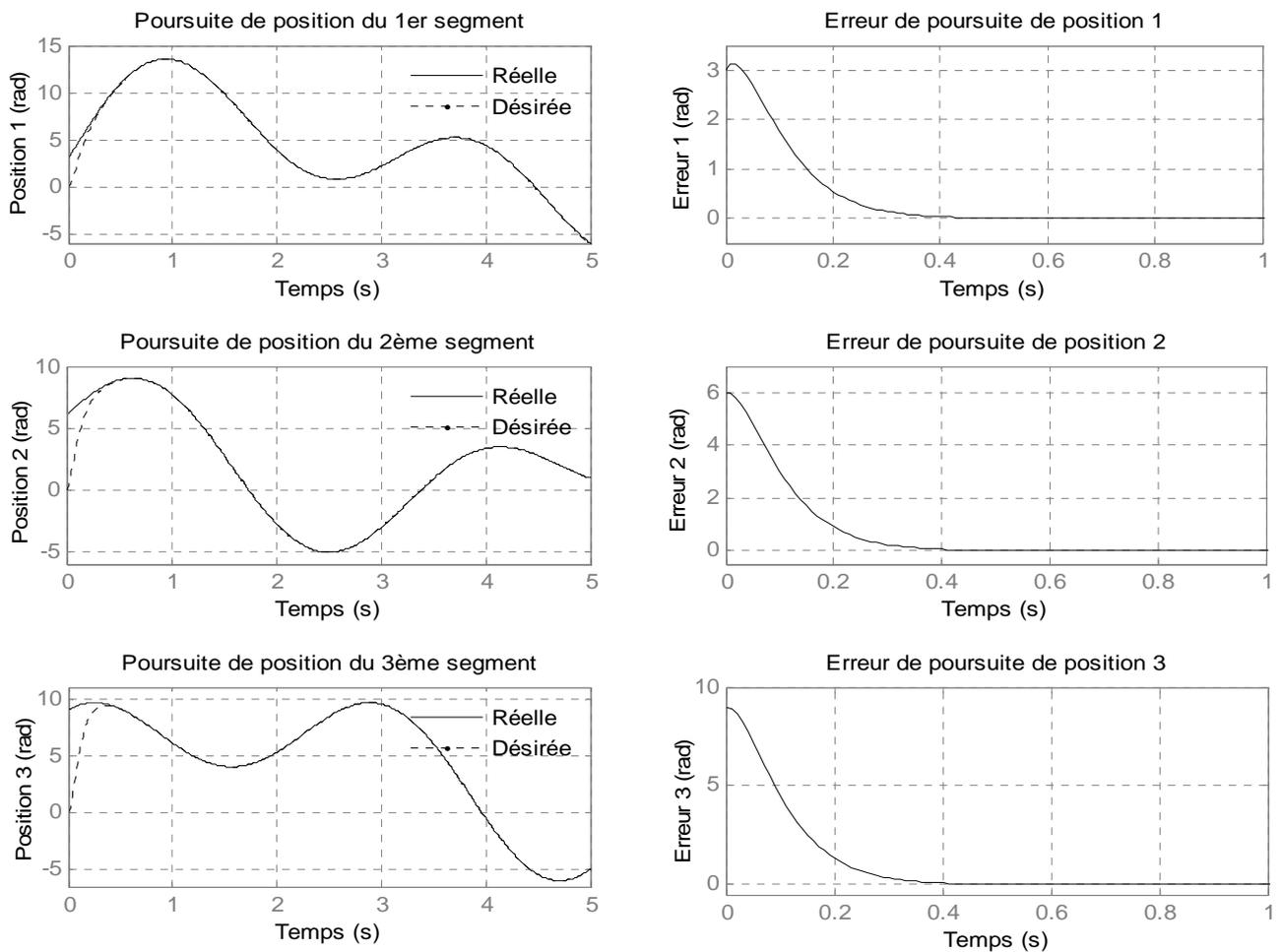
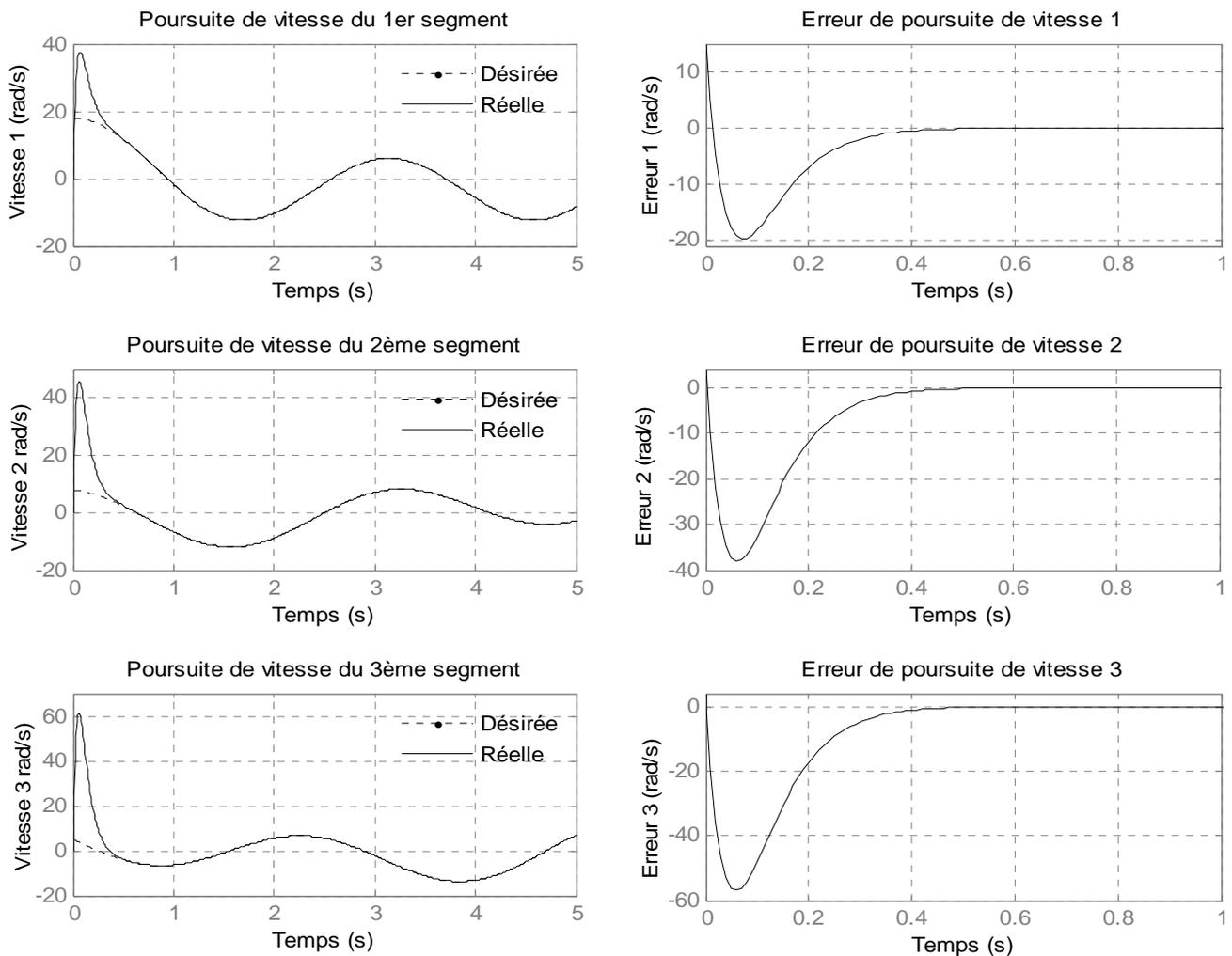


Figure II-26 : Les poursuites de trajectoire de position et les erreurs de poursuite de position pour la loi de commande Couple Calculé de type PD

La figure II-27, montre le comportement du robot en poursuite de la vitesse désirée.



**Figure II-27 :** Les poursuites de trajectoire de vitesse et les erreurs de poursuite de vitesse pour la loi de commande Couple Calculé de type *PD*

**Analyse des résultats**

D’après les résultats présentés sur les figures II-26 et II-27, nous constatons que les différents segments du robot suivent bien les trajectoires et les vitesses désirées. Cela signifie que les valeurs des gains du régulateur ont été bien choisies. D’où la convergence des erreurs de poursuite de position ou de vitesse à zéro est garantie et que le système est parfaitement linéarisé et découplé avec les gains cités auparavant.

**Conclusion**

Ce chapitre concerne un état de l’art sur les différentes structures de commande appliquées aux systèmes robotisés. Nous avons développé les différentes méthodes de génération de trajectoires utilisées pour le calcul des consignes de commande et les techniques de commande classique à savoir les méthodes qui traitent le problème de régulation (*PD*, *PID*, *PD* avec compensation de l’effet de gravité et la commande par découplage non-linéaire) et celles qui traitent le problème de poursuite de

trajectoire (commande basée sur la linéarisation avec retour d'état (la commande Couple Calculé) et les commandes basées sur la théorie de Lyapunov (la commande de Paden et Panja, la commande de Slotine et Li, ...)).

Les différentes méthodes de commande en effort (compliance passive, commande d'impédance mécanique, commandes hybrides position/force et la commande hybride externe) lorsque le robot rentre en contact avec son environnement ont été présentées.

Il faut noter qu'il y'a aussi d'autres techniques de commande telles que les commandes adaptatives (qui tiennent compte des variations paramétriques du modèle dynamique du robot) et les commandes robustes (développement des techniques permettant de résoudre ou de contourner le problème de l'existence dans la phase de modélisation, des incertitudes structurelles et paramétriques et dans la phase opératoire des perturbations causées par des bruits de mesure ou incident particulier (présence d'obstacles)) que nous n'avons pas développé dans ce chapitre.

Nous avons mis en œuvre, par simulation, deux lois de commande classiques à savoir la régulation de position *Point à Point* et la poursuite de trajectoire basée sur le *Couple calculé*. Les résultats de simulation montrent la convergence des erreurs de positions et de poursuite de trajectoire vers zéros dans les deux cas. L'inconvénient présenté dans la première méthode est les grandes valeurs des gains utilisés qui peuvent provoquer des problèmes sur les actionneurs du robot lorsqu'une perturbation se présente.

L'implémentation de la deuxième méthode nécessite la connaissance précise et complète du modèle dynamique du robot et qui y a donné, dans ce cas, des résultats parfaits. Dans le cas général, les vecteurs des couples de Coriolis, centrifuges et de gravité sont très compliqué, par conséquent, très difficile à les modéliser. Ce qui rend cette méthode très sensible aux incertitudes de modélisation et aux perturbations extérieures.

Nous montrons dans le chapitre III, comment nous procédons au développement de nouvelles versions de commandes intelligentes, tout en gardant l'architecture de cette commande, afin de limiter les effets des éléments non modélisables agissant sur la dynamique du robot et de regagner l'efficacité de cette méthode.

## Références bibliographiques

- [1] Canudas de Wit C., B. Brogliato, A. De Luca, B. Siciliano, P. Tomei, C. Samson, R. Ortega, G. Campion, B. D'Andrea-Novell, W. Khalil, *Théorie de la commande des robots*, Laboratoire d'automatique de grenoble, grenoble, France, 1992.
- [2] Craig J. J., *Adaptive control of mechanical manipulators*, Adisson-Wesley Publishing Company, Inc, New York, 1988.
- [3] Dombre E., Wisama Khalil, *Modélisation et commande des robots*, Hermès, paris, 1988.
- [4] De Schutter J. et Van Brussel H., *Compliant robot motion II. A control approach based on external control loops*, Int. Jour. of Robotics Research, vol. 7(4), p. 18–33, Août 1988.
- [5] Dégoulange É., *Commande en effort d'un robot manipulateur à deux bras : application au contrôle de la déformation d'une chaîne cinématique fermée*, Thèse de Doctorat, Université Montpellier II, Montpellier, France, Décembre 1993.
- [6] Duchemin G., *Commande et programmation d'un robot d'assistance au geste médical pour des tâches de suivi au contact de tissus mous*, thèse de doctorat, Université Montpellier II, France, Décembre 2002.
- [7] Hogan N., *Stable execution of contact tasks using impedance control*, Proc. IEEE Int. Conf. on Robotics and Automation (ICRA), Raleigh, NC, p. 1047–1054, 1987.
- [8] Khalil W., *Contribution à la commande automatique des manipulateurs avec l'aide d'un modèle mathématique des mécanismes*, Thèse d'état, USTL, Montpellier, 1978.
- [9] Khalil W., Etienne Dombre, *Modélisation identification et commande des robots*, Hermès, paris, 1999.
- [10] Khelfi M. F., *Observateurs non linéaires : Application à la commande des robots manipulateurs*, Thèse de Doctorat, Univ. Henri Poincaré, Nancy I, France, 1995.
- [11] Khatib O., *Commande dynamique dans l'espace opérationnel des robots manipulateurs en présence d'obstacles*, Thèse de Docteur-Ingénieur, Ecole Nationale Supérieure de l'Aéronautique et de l'Espace, Toulouse, 1980.
- [12] Khatib O., *The operational space formulation in robot manipulator control*, Proc. 15<sup>th</sup> Int. Symp. on Industrial Robots, Tokyo, Japan, p. 165–172, Septembre 1985.
- [13] Khosla P. K., *Real-time control and identification of direct drive manipulators*, Thèse Ph. D., Carnegie Mellon University, Pittsburgh, 1986.
- [14] Lawrence D.A., *Impedance control stability properties in common implementations*, Proc. IEEE Int. Conf. on Robotics and Automation (ICRA), Philadelphie, PA, p. 1185–1190, Avril 1988.
- [15] Lewis F. L., C. T. abdallah, D. M. Dawson, *Control of robot Manipulators*, Macmillan, New York, 1993.
- [16] Lozano R. et D. Taoutaou, *Commande adaptative et applications*, Hermès, paris, France, 2001.
- [17] Paden B. et R. Panja, *Globally asymptotically stable "PD+" controller for robot manipulators*, Int. Journal on control, Vol. 47, p. 1697-1712. 1988.

- [18] Perdereau V., *Contribution à la commande hybride force-position – Application à la coopération de deux robots*, Thèse de Doctorat, Université Pierre et Marie Curie, Paris, France, Février 1991.
- [19] Pierrot F., Dombre E., Dégoulange E., Urbain L., Caron P., Boudet S., Gariépy J. et Mégnien J.L., *Hippocrate: a safe robot arm for medical applications with force feedback*, Medical Image Analysis, vol. 3(3), p. 285–300, 1999.
- [20] Pujas A., *Étude de la robustesse de schéma de commande position/force pour robots à deux bras*, Thèse de Doctorat, Université Montpellier II, Montpellier, France, Juin 1995.
- [21] Raibert M.H. et Craig J.J., *Hybrid position/force control of manipulators*, Trans. of the ASME, Jour. of Dynamic Systems, Measurement and Control, vol. 103, p. 126–133, Juin 1981.
- [22] Reboulet C., Robert A., Poilve H. et Gaillet A., *Hybrid position force control – Application to assembly*, Proc. 15<sup>th</sup> Int. Symp. on Industrial Robots, Tokyo, Japan, p. 157–164. Septembre 1985.
- [23] Spong M. W., M. Vidyasagar, *Robot dynamics and control*, John Wiley & Sons, New York, 1989.
- [24] Théron X., *Étude et réalisation d'un prototype de robot de positionnement de sondes pour des applications médicales*, Thèse de Doctorat, Université Montpellier II, Montpellier, France, Décembre 1996.
- [25] Whitney D.E., *Historical perspective and state of the art in robot force control*, Int. Jour. of Robotics Research, vol. 6(1), p. 3–14. 1987.
- [26] Zabala Iturralde J., *Commande des robots manipulateurs à partir de la modélisation de leur dynamique*, Thèse de troisième cycle, Univ. P. Sabatier, Toulouse, 1978.
- [27] Zhang H. et Paul R.P., *Hybrid control of robot manipulators*, Proc. IEEE Int. Conf. on Robotics and Automation (ICRA), Scottsdale, AZ, Saint Louis, MO, p. 602–607, Mars 1985.

## *Chapitre III*

# *Etat de l'art sur les techniques intelligentes de commande / Applications*

## Introduction

L'un des défis de l'homme aujourd'hui est de copier la nature et de reproduire des modes de raisonnement et de comportement qui lui sont propres. Les réseaux de neurones, la logique floue, les systèmes neuro-flous etc. sont nés de cette envie. Bien que ces approches se soient imposées rapidement dans des domaines tels que le traitement de l'image, la reconnaissance de forme, la classification, la gestion financières, etc., leur application dans le domaine de la robotique est très répandues pour la réalisation de l'identification et de la commande.

Dans ce chapitre, nous donnons quelques aperçus de ces nouveaux outils, nous commençons par définir et expliquer la terminologie utilisée en logique floue. La théorie des ensembles flous et ainsi que les modes de raisonnement propres aux variables floues. Nous abordons les différentes étapes nécessaires pour la réalisation de l'inférence floue et développons une méthode de synthèse de régulateur flou.

Nous présentons par la suite quelques notions de base sur les réseaux de neurones artificiels, les différentes architectures et modèles seront exposés, les méthodes de l'apprentissage de ces réseaux seront également abordées. Nous citons les principales applications des RNA dans le domaine de l'identification et les approches de commande rencontrées dans la littérature scientifique. Nous exposons une technique d'optimisation de systèmes d'inférence floue de type Takagi Sugeno (ANFIS). Enfin et comme applications, nous proposons deux techniques de commandes intelligentes, floue et neuro-floue, afin d'améliorer les performances de la commande Couple Calculé développée dans le chapitre précédent.

### III-1. La logique floue

#### III-1-1. Historique de la logique floue

La logique floue a été développée par le professeur L. A. Zadeh de l'Université de Berkeley (Californie) en 1965. En 1973, Zadeh propose d'appliquer la logique floue aux problèmes de réglage des systèmes. La première application expérimentale d'un contrôleur flou a été effectuée par Mamdani en 1974, suivie d'une application industrielle en 1980 dans une cimenterie. En 1987, les premiers régulateurs flous industriels sont proposés sur le marché [1], [2], [3].

#### III-1-2. Principe de base et définition

Contrairement aux notions de la logique classique, la logique floue traite les notions imprécises, dont l'objectif est de modéliser le langage naturel de l'être humain. Elle repose sur la théorie des ensembles flous.

Pour mieux distinguer la différence entre la logique classique (booléenne) et la logique floue, considérons l'exemple d'une personne pour lequel on définit trois classes d'âges, « jeune » dans l'intervalle  $[0 \ 35[$  ans, « âge moyen » dans  $]35 \ 55[$  ans et « âgé » de 55 ans et plus.

Selon la logique booléenne, une personne de 34 ans est classée comme jeune et celle de 35 ans est classée comme moyen avec les mêmes degrés de vérité «1». La remarque ici, ce que cette représentation ne permet pas de rendre compte de l'aspect vague du qualificatif « âge moyen » (figure III-1).

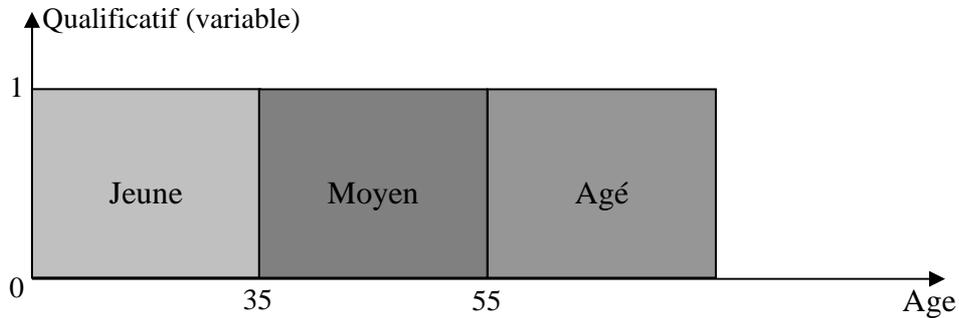


Figure III-1: Exemple de représentation d'une variable en logique booléenne.

En logique floue, ce degré de vérité peut prendre plusieurs valeurs réelles entre 0 et 1 de façon progressive. Ceci, nous permet de mieux représenter l'aspect vague du raisonnement humain. Cette représentation est donnée sur la figure III-2. Dans ce cas, une personne peut être classée, à la fois, comme d'âge moyen avec un degré de vérité de 0.8, et comme jeune avec un degré de 0.2.

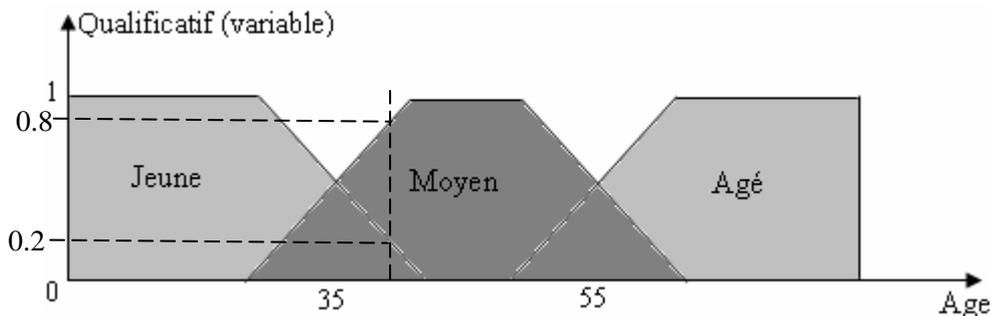


Figure III-2: Exemple de représentation d'une variable en logique floue.

**III-1-2-1. Définition d'un sous-ensemble flou**

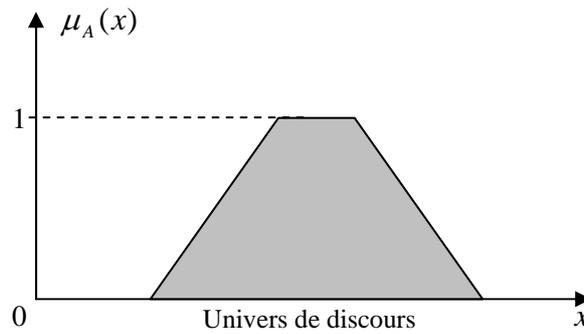
Pour la variable floue  $x$ , on définit un sous-ensemble flou  $A$  sur un univers de discours  $X$  par une fonction d'appartenance :

$$\mu_A : X \rightarrow [0 \ 1]$$

$$x \rightarrow \mu_A(x)$$

L'univers de discours est l'ensemble des valeurs réelles que peut prendre la variable floue  $x$ , et  $\mu_A(x)$  est le degré d'appartenance de l'élément  $x$  à l'ensemble  $A$ .

**Remarque.** Par abus de langage, un sous-ensemble flou est fort souvent confondu avec l'ensemble flou.



**Figure III-3:** Exemple d'une représentation d'un ensemble flou par sa fonction d'appartenance

Dans l'exemple de la figure III-3, la variable floue est l'âge d'une personne, l'univers de discours est l'ensemble des valeurs réelles de 0 à plus de 55 ans. Les qualificatifs « Jeune », « âge moyen » et « Agé » sont appelés sous ensembles flous, et chacun est caractérisé par sa fonction degré d'appartenance par exemple  $\mu_{Jeune}(\text{âge})$ .

### III-1-2-2. Opérateurs logiques

De façon analogue à la théorie des ensembles classiques, on définit l'intersection, l'union des ensembles flous ainsi que le complémentaire d'un ensemble flou. Ces relations sont traduites par les opérateurs logiques « et », « ou » et « non » :

$$x \text{ appartient à } A \text{ et } B \Leftrightarrow x \in A \cap B$$

$$x \text{ appartient à } A \text{ ou } B \Leftrightarrow x \in A \cup B$$

$$x \text{ appartient au complément de } A \Leftrightarrow x \in \bar{A}$$

Ces opérateurs en logique floue se font à partir des fonctions d'appartenance aux ensembles flous correspondants :

$$x \text{ appartient à } A \text{ et } B \Leftrightarrow \mu_{A \cap B}(x)$$

$$x \text{ appartient à } A \text{ ou } B \Leftrightarrow \mu_{A \cup B}(x)$$

$$x \text{ appartient au complément de } A \Leftrightarrow \mu_{\bar{A}}(x)$$

#### III-1-2-2-1. t-norme et s-norme

De façon générale, l'opérateur « et » se définit par une t-norme (norme triangulaire) comme suit :

$$T : [0 \ 1] \times [0 \ 1] \rightarrow [0 \ 1]$$

$$(x, \ y) \rightarrow z = xT y$$

Tel que  $T$  possède les propriétés suivantes :

- Commutativité :  $xT y = yT x$
- Associativité :  $(xT y)T z = xT (yT z)$
- Monotonie :  $xT w \leq yT z$  si  $x \leq y, w \leq z$
- Admet 0 comme élément absorbant et 1 comme élément neutre :

$$0T x = xT 0 = 0, 1T x = xT 1 = x$$

De même, l'opérateur « ou » se définit par une s-norme (co-norme triangulaire) comme suit :

$$S : [0 \ 1] \times [0 \ 1] \rightarrow [0 \ 1]$$

$$(x, y) \rightarrow z = xS y$$

Tel que  $S$  possède les propriétés suivantes :

- Commutativité :  $xS y = yS x$
- Associativité :  $(xS y)S z = xS (yS z)$
- Monotonie :  $xS w \leq yS z$  si  $x \leq y, w \leq z$
- Admet 1 comme élément absorbant et 0 comme élément neutre :  $1S x = xS 1 = 1, 0S x = xS 0 = x$

A l'aide de la loi de Morgan, on peut associer à chaque t-norme, une s-norme définie par :

$$xS y = 1 - (1 - x)T(1 - y)$$

La réalisation des opérateurs les plus utilisés en logique floue, est donnée comme suit :

- Pour l'opérateur «et», on peut utiliser soit la fonction «min» soit la fonction arithmétique «produit» :

$$\begin{cases} \mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)) \\ \mu_{A \cap B}(x) = \mu_A(x) \cdot \mu_B(x) \end{cases}$$

- Pour l'opérateur «ou», on peut le réaliser soit par la fonction «max» soit par la fonction arithmétique «somme» :

$$\begin{cases} \mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)) \\ \mu_{A \cup B}(x) = \mu_A(x) + \mu_B(x) \end{cases}$$

- Pour l'opérateur «non», on le réalise par :  $\mu_{\bar{A}}(x) = 1 - \mu_A(x)$

### III-1-2-3. Inférence floue

En logique classique, la règle de raisonnement du *modus ponens* permet, à partir des deux assertions :

- $x$  est  $A$
- si  $x$  est  $A$  alors  $y$  est  $B$ ,

de conclure que  $y$  est  $B$

En logique floue, la règle de raisonnement s'appelle *modus ponens généralisé* et qui permet à partir des assertions :

- $x$  est  $A'$
- si  $x$  est  $A$  alors  $y$  est  $B$ ,

de conclure que  $y$  est  $B'$

Sans entrer dans les détails de formalisation mathématique, qui se basent sur les notions de sous-ensembles flous, de graphes ( $\Gamma$  est le graphe définissant la relation ( $R$ ) de  $A$  vers  $B$ ) et de projection ( $B'$  est la projection sur  $B$  de  $A'$  par le graphe de  $\Gamma$ ), il est possible de définir l'ensemble  $B'$  par :

$$\forall y \in B, \mu_{B'}(y) = \sup_{x \in A} \mu_{A' \times B \cap \Gamma}(x, y) = \sup_{x \in A} \min(\mu_{A'}(x), \mu_R(x, y))$$

C'est-à-dire que le degré d'appartenance de chaque élément  $y$  de  $B$  à l'ensemble flou  $B'$  est égal au plus grand degré d'appartenance des couples  $(x, y)$  à l'intersection de l'ensemble  $A'$  avec le graphe  $\Gamma$  de la relation  $R$  [2], [4].

**III-1-3. Structure d'un contrôleur flou**

La commande floue est une application de la logique floue au contrôle des processus dynamiques pour lesquels on ne possède pas de modèles satisfaisant. Son principe est simple : il s'agit dans la plupart des cas d'imiter le comportement d'un opérateur humain dans la régulation d'un processus complexe, à l'aide de règles floues.

La structure de base d'un régulateur flou est inspirée de celle du régulateur *PI* discret dont l'équation, sous forme incrémentale, est [2], [4], [12] :

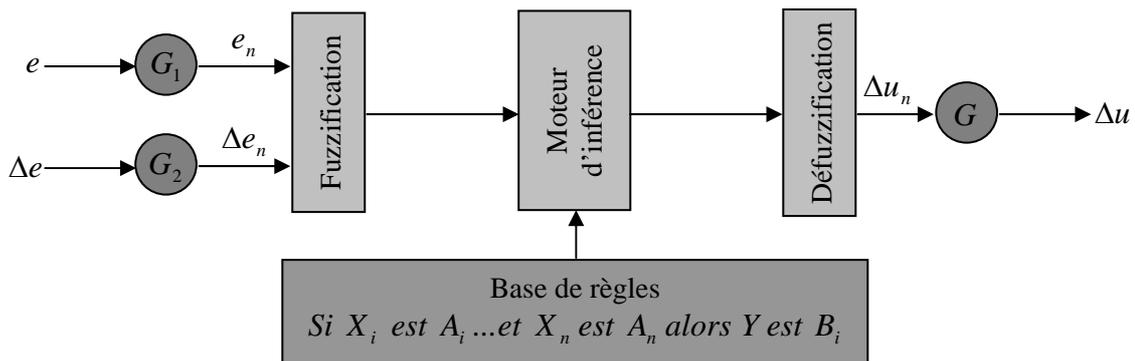
$$u(k) = u(k-1) + K \Delta e(k) + K \frac{T_e}{T_i} e(k-1) \tag{III-1}$$

où  $K$ ,  $T_i$ ,  $T_e$ ,  $e(k)$ ,  $\Delta e(k) = e(k) - e(k-1)$  et  $u(k)$  sont respectivement : la constante proportionnelle, la constante intégrale, la période d'échantillonnage, l'écart entre la consigne et la mesure, sa variation entre deux instants d'échantillonnage et l'action à appliquer.

L'équation (III-1) peut être réécrite sous la forme :

$$\Delta u = f(e, \Delta e) \tag{III-2}$$

Dans la régulation floue ;  $\Delta u$ ,  $e$  et  $\Delta e$  sont exprimées par des règles manipulant des variables linguistiques.



**Figure III-4:** Schéma de principe d'un régulateur flou

Avec :  $G_1$  et  $G_2$  représentent les gains de normalisation (facteurs d'échelles) transformant les valeurs physiques  $(e, \Delta e)$  des entrées en des valeurs normalisées  $(e_n, \Delta e_n)$  appartenant à l'intervalle  $[-1 \ 1]$  (univers de discours normalisé). Le gain  $G$  permet de dénormaliser (transformer) les valeurs normalisées de la variable de commande  $(\Delta u_n)$  en des valeurs en conformité avec son domaine physique d'application  $(\Delta u)$ .

### III-1-3-1. Fuzzification

La fuzzification consiste à transformer les valeurs numériques relatives aux entrées du système ( $e$  et  $\Delta e$ ) en quantités floues. Cela se fait après la définition des ensembles flous de ces variables d'entrées (valeurs linguistiques par exemple :  $N$ ,  $Z$ ,  $P$ ) et leurs fonctions d'appartenances. Cette opération permet de calculer le degré d'appartenance de la variable floue en fonction de la grandeur réelle d'entrée.

Le choix de la forme des fonctions d'appartenances est réalisé soit par des courbes triangulaires, trapézoïdales ou des courbes en cloche (gaussiennes) pour les variables d'entrées.

La figure III-5, représente un exemple de fuzzification de l'erreur  $e$  (de la même manière pour  $\Delta e$ ).

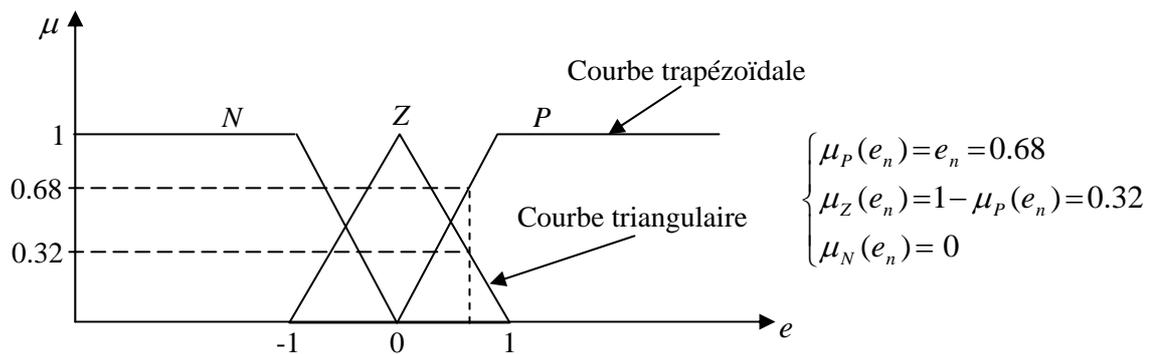


Figure III-5 : Fuzzification de l'erreur  $e$

Le recouvrement de deux fonctions d'appartenances voisines est de 1 ; c'est-à-dire que leur croisement s'effectue à  $\mu = 0.5$ . Un recouvrement insuffisant voir inexistant conduit à une zone où aucune règle d'inférence n'est sollicitée. De même, un recouvrement trop important, surtout avec un degré d'appartenance près de l'unité, conduit à un aplatissement de la caractéristique du régulateur [1], [5]. Le passage d'une fonction d'appartenance à sa voisine doit s'effectuer en douceur de manière à ce qu'il y ait au moins deux règles d'inférence qui soient sollicitées en même temps.

### III-1-3-2. Inférence

Cette étape consiste à appliquer les règles de la logique floue sur les grandeurs trouvées après la fuzzification. Ces règles peuvent être représentées de plusieurs formes [5-8]:

- Linguistiquement, par exemple : **SI** (le niveau est haut) **ET** (le débit est faible) **ALORS** (la vanne est complètement fermée).

**OU**

**SI** (le niveau est moyen) **ET** (le débit est faible) **ALORS** (la vanne est fermée).

- Symboliquement : une description linguistique où l'on remplace la désignation des ensembles flous par abréviations (par exemple : grand positif par GP).
- Par une matrice d'inférence. Celle-ci regroupe toutes les règles d'inférences sous forme de tableau. Dans le cas d'un tableau à deux dimensions, les entrées du tableau représentent les

ensembles flous des variables d'entrées ( $e$  et  $\Delta e$ ). L'intersection d'une colonne et d'une ligne donne l'ensemble flou de la variable de sortie définie par la règle.

Afin de définir les degrés d'appartenance de la variable de sortie à ses sous-ensembles flous. Nous présentons les méthodes d'inférence les plus utilisées, et qui se différencient essentiellement de la manière de représenter les opérateurs «**ET**» et «**OU**» utilisées dans les règles d'inférence, qui sont : la méthode d'inférence Min/Max, Max/Produit, Somme/Produit et la méthode d'inférence TVFI.

#### III-1-3-2-1. Méthode d'inférence Max/Min

Cette méthode, dénommée aussi «implication de Mamdani», réalise l'opérateur «**ET**» et la conclusion «**ALORS**» de chaque règle par la fonction **Min** et l'opérateur «**OU**» de la liaison entre toutes les règles par la fonction **Max**.

#### III-1-3-2-2. Méthode d'inférence Max/Produit

Cette méthode, dénommée aussi «implication de Larsen», réalise l'opérateur «**ET**» par la fonction **Min** et l'opérateur «**OU**» de la liaison entre toutes les règles par la fonction **Max**. Par contre, la conclusion dans chaque règle, introduite par «**ALORS**», lie le facteur d'appartenance de la condition avec la fonction d'appartenance de la variable de sortie par l'opérateur «**ET**», est réalisé cette fois-ci par la formation du **Produit**.

#### III-1-3-2-3. Méthode d'inférence Somme/Produit

Cette méthode réalise l'opérateur «**ET**» et la conclusion «**ALORS**» par le produit. Par contre, l'opérateur «**OU**» est réalisé par la formation de la somme et plus précisément par la valeur moyenne des degrés d'appartenance intervenant dans l'inférence.

Cette méthode est la plus utilisée car son temps de calcul est court. Par opposition aux méthodes d'inférences précédentes.

#### III-1-3-2-4. Méthode d'inférence TVFI (True Value Fuzzy Interférence ou méthode des valeurs concrètes)

Cette méthode, dénommée aussi «méthode de Sugeno», est une extension de la méthode Min/Max et de la méthode Somme/Produit, mais elle utilise des sous-ensembles «discrets». Par sous-ensemble «discret», on entend sous-ensembles limités à des singletons.

#### Remarque

L'établissement des règles d'inférence est généralement basé sur un des points suivants :

- L'expérience de l'opérateur et/ou du savoir faire de l'ingénieur en régulation et en contrôle.

- Un modèle flou du processus pour lequel on souhaite synthétiser le régulateur.
- Les actions de l'opérateur ; s'il n'arrive pas à exprimer linguistiquement les règles qu'il utilise implicitement.
- L'apprentissage ; c'est-à-dire la synthèse de règle se fait par un procédé automatique également appelé superviseur. Souvent, des réseaux de neurones y sont associés.

### III-1-3-3. Défuzzification

Cette étape consiste à produire une sortie réelle (non floue) de telle façon à traduire au mieux le résultat de l'inférence floue, c'est à dire la fonction d'appartenance de la variable de sortie. Pour cela, il existe plusieurs méthodes de calcul de la sortie, dont les principales sont:

#### III-1-3-3-1. Méthode du centre de gravité

C'est la méthode la plus utilisée qui consiste à calculer le centre de gravité de la fonction d'appartenance résultant de l'inférence. L'abscisse de celui-ci correspond à la valeur de sortie du régulateur.

$$\Delta u_n = \frac{\int x \mu_R(x) dx}{\int \mu_R(x) dx} \quad (\text{III-3})$$

Lorsque la fonction d'appartenance résultante est compliquée, le calcul de la sortie devient relativement complexe et long.

#### III-1-3-3-2. Méthode par valeur maximum

Dans cette méthode, la valeur de sortie est choisie comme l'abscisse de la valeur maximale de la fonction d'appartenance (figure III-6-a).

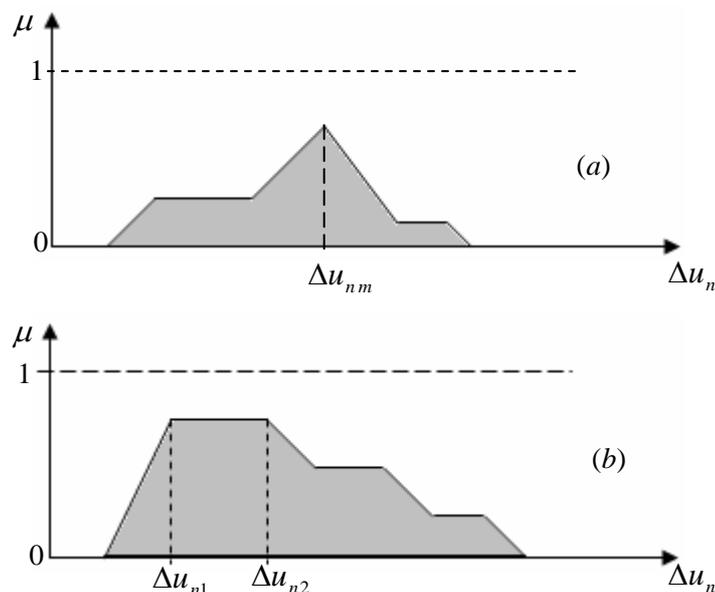


Figure III-6 : Exemple de defuzzification par valeur maximum

Dans le cas où la fonction d'appartenance à plus d'un maximum, on prend la moyenne ( $\Delta u_n = \frac{\Delta u_{n1} + \Delta u_{n2}}{2}$ ) des maxima (figure III-6-b).

Cette méthode a l'avantage de ne nécessiter que peu de calculs mais présente l'inconvénient de pouvoir engendrer des discontinuités dans la loi de commande. Ceci explique pourquoi elle n'est pas fréquemment utilisée en commande floue.

## III-2. Les réseaux de neurones

### III-2-1. Principe

Les réseaux de neurones artificiels (*Artificial Neural Network* ANN) sont apparus en 1943 lors d'essais de modélisation du neurone biologique par Warren McCulloch et Walter Pitts. Ils supposent que l'impulsion nerveuse est le résultat d'un calcul simple effectué par chaque neurone et que la pensée est née grâce à l'effet collectif d'un réseau de neurones interconnectés [9], [11].

Le modèle général d'un neurone formel est donné par le schéma suivant :

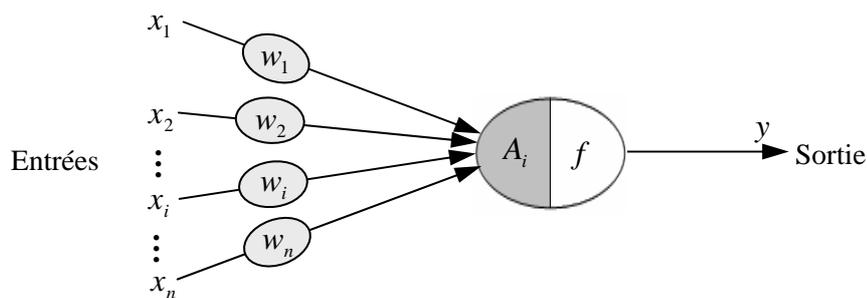


Figure III-7 : Structure générale d'un neurone formel

Les entrées du neurone sont désignées par  $x_i$  ( $i = 1, \dots, n$ ). Les paramètres  $w_i$  reliant les entrées aux neurones sont appelés poids synaptiques ou tout simplement poids. La sortie du neurone est donnée par :

$$y = f(A_i) \text{ où } A_i = \sum_{i=1}^n w_i x_i \quad (\text{III-4})$$

$A_i$  est l'état d'activation du neurone (également appelé état ou activité).

$f$  est la fonction d'activation (appelée aussi fonction de transfert ou de transition) du neurone.

Conformément au modèle biologique, les fonctions d'activation sont généralement croissantes et bornées. Les fonctions de transfert les plus connues sont présentées comme suit :

#### a. La fonction binaire à seuil

Dans ce cas, on peut utiliser la fonction Heaviside ou la fonction signe comme indiqué sur la figure III-8.

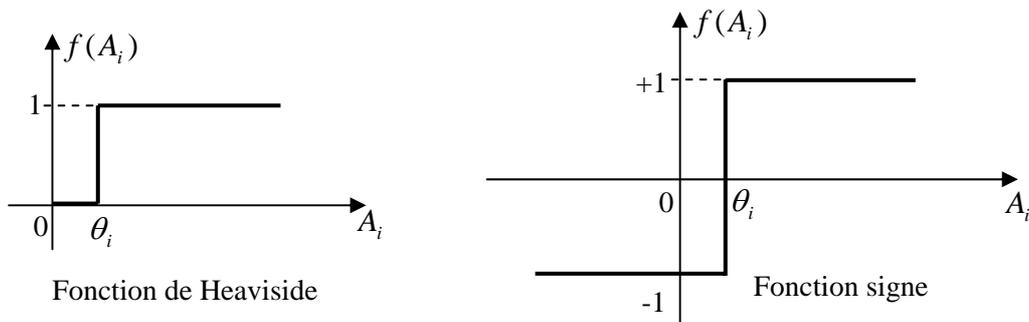


Figure III-8 : Fonction binaire à seuil

**b. La fonction linéaire à seuil**

La figure III-9, présente un exemple d'une fonction linéaire à seuil défini comme suit :

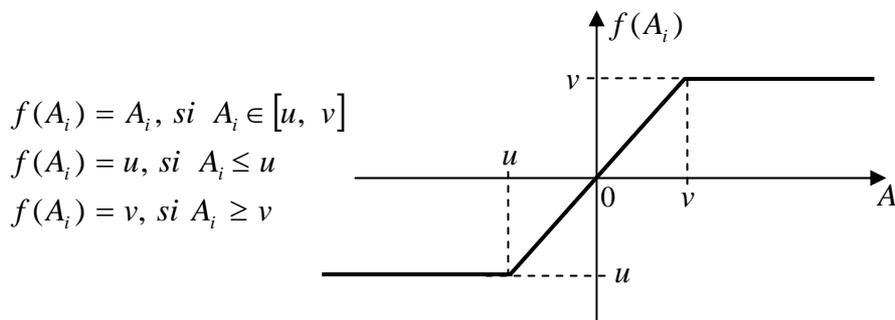


Figure III-9 : Fonction linéaire à seuil

**c. La fonction sigmoïde**

Cette fonction est inspirée directement de l'examen du comportement des cellules nerveuses face aux signaux qui leur arrivent. La forme d'une telle fonction est donnée comme suit :

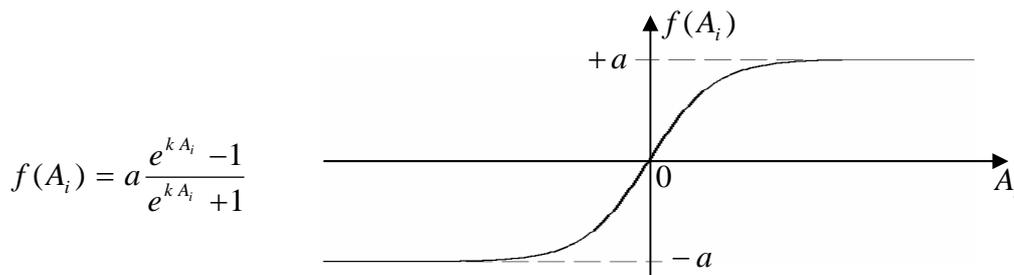


Figure III- 10 : Fonction sigmoïde

**III-2-2. Réseaux de neurones**

Les réseaux de neurones sont constitués des neurones élémentaires connectés entre eux par l'intermédiaire des poids qui jouent le rôle des synapses. L'information est portée par la valeur de ses poids tandis que la structure du réseau de neurones ne sert qu'à traiter cette information et l'acheminer vers la sortie.

Les connexions entre neurones peuvent se faire de plusieurs manières. Cette interconnexion est appelée l'architecture d'un réseau.

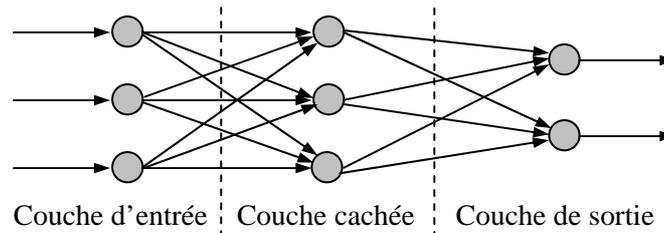
### III-2-3. Architectures des réseaux de neurones

On peut distinguer trois types d'architectures de réseaux de neurones [9], [13], [14] :

#### III-2-3-1. Réseaux non bouclés

Appelés aussi réseaux à couches ou réseaux unidirectionnels (*feedforward networks*) ont leurs neurones organisés sous la forme d'une ou de plusieurs couches successives.

La figure III-11, présente le principe d'un réseau bouclé à trois couches.



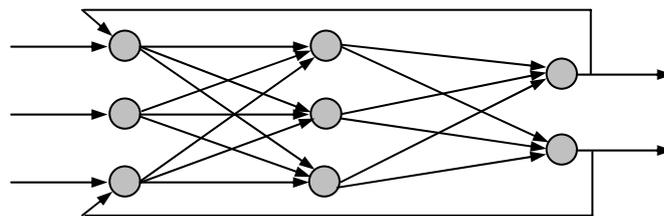
**Figure III-11** : Structure de base d'un réseau à couches

Dans ce type de réseaux, la propagation de l'information se déroule de la couche d'entrée vers la couche de sortie au travers d'éventuelles couches intermédiaires mais sans retour en arrière.

#### III-2-3-2. Réseaux bouclés

Appelés aussi réseaux récurrents (*recurrent networks*) possèdent une structure similaire à celle des réseaux unidirectionnels mais complétée par des connexions entre éléments de la même couche ou vers des couches en amont.

La figure III-12, présente le principe d'un réseau bouclé.



**Figure III-12** : Structure de base d'un réseau récurrent

Ces réseaux sont assez puissants car leur fonctionnement est séquentiel et adopte un comportement dynamique.

#### III-2-3-3. Réseaux à connexions complexes

Sont des réseaux totalement interconnectés et qui ne trouvent pas leur place dans les deux catégories précédemment citées. Le principe de ce type de réseau est donné par la figure III-13.

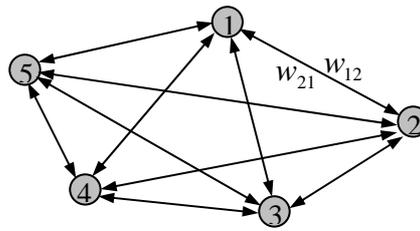


Figure III-13 : Structure de base d'un réseau totalement interconnecté

**III-2-4. Quelques modèles de réseaux de neurones**

Selon le type d'architecture utilisée, on peut trouver plusieurs modèles de réseaux de neurones parmi lesquels en obtient :

**III-2-4-1. Le perceptron multicouche**

Le mécanisme perceptron fut inventé par le psychologue Frank Rosenblatt à la fin des années 50, lors de sa tentative d'illustrer certaines propriétés fondamentales des systèmes intelligents en général.

Les perceptrons multicouche sont des réseaux de type *feedforward* (voir figure III-11), ses fonctions d'activations, pour les couches intermédiaires, sont du type sigmoïde. La sortie possède une fonction d'activation de type linéaire bornée.

**III-2-4-2. Le modèle Adaline**

Le réseau de neurones adaptatif dit Adaline (*ADAPtive LInear NEuren*) appartient à la famille des perceptrons. Il possède un seul neurone à fonction d'activation linéaire ( $f(A_i) = A_i$ ) et une entrée sous forme d'un vecteur de plusieurs éléments. Il a été proposé et développé par Widrow et Hoff.

Le schéma de principe d'un Adaline est décrit dans la figure III-14.

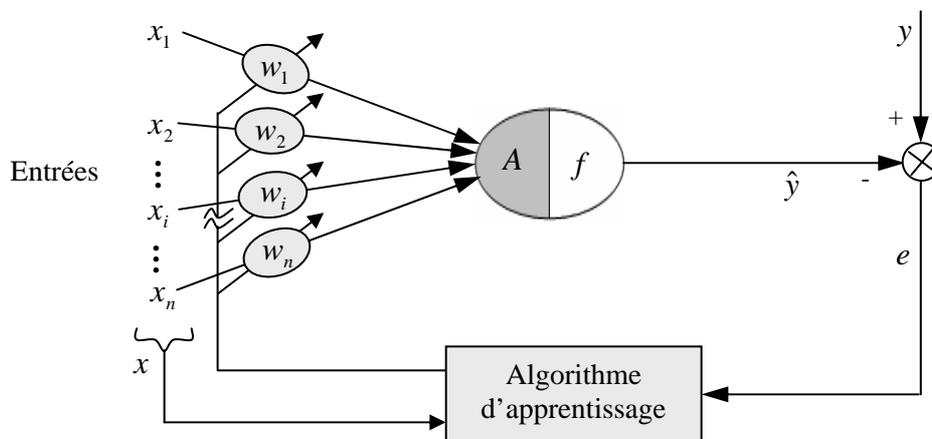


Figure III-14 : Structure d'un Adaline

La sortie estimée  $\hat{y}$  de la référence  $y$  est donnée par la relation linéaire :

$$\hat{y} = W^T x \tag{III-5}$$

où  $W^T$  représente le vecteur des poids estimés par l'Adaline et  $x$  le vecteur des entrées constitué des composantes issues des connaissances a priori de la référence  $y$ .

**III-2-4-3. Réseaux de neurones à fonctions de base radiale**

Les réseaux de neurones à fonction de base radiale (*Radial Basis Functions*), proposés par J. Moody et C. Darken, sont des réseaux de neurones à une seule couche cachée dont les fonctions d'activation sont des fonctions à base radiale, le plus souvent des gaussiennes. La fonction d'activation du neurone de la couche de sortie est l'identité. Les entrées sont connectées directement aux neurones de la couche cachée. Ces réseaux appartiennent à la famille des réseaux non bouclés.

La figure III-15, présente le principe d'un réseau de type RBF.

Pour un réseau de  $n$  entrées et  $m$  neurones cachés, la sortie d'un neurone de la couche cachée est donnée par :

$$A_j = \exp\left(-\frac{1}{2} \sum_{i=1}^n \frac{(x_i - c_{i,j})^2}{\sigma_{i,j}^2}\right) \tag{III-6}$$

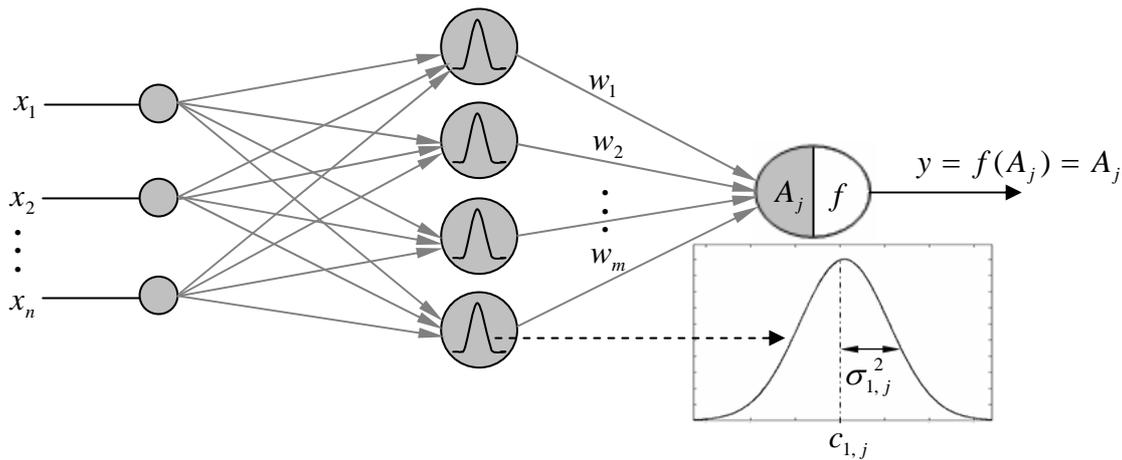


Figure III-15 : Principe d'un réseau de neurone de type RBF

où  $j$  désigne l'indice du neurone,  $i$  parcourt l'ensemble des entrées notées  $x_i$ ,  $c_{i,j}$  et  $\sigma_{i,j}^2$  sont des paramètres appelés respectivement centres et variances des gaussiennes (où  $\sigma_{i,j}$  est l'écart type).

La sortie du réseau est :

$$y = \sum_{j=1}^m w_j y_j \tag{III-7}$$

**III-2-5. Apprentissage**

Une fois la structure est fixée, il faut passer par le processus d'apprentissage, par lequel les poids vont être ajustés de manière à satisfaire un critère d'optimisation. On distingue deux types d'apprentissage en fonction de la nature des informations disponibles et du but recherché :

### III-2-5-1. Apprentissage supervisé

L'apprentissage est dit supervisé lorsque les informations (ou les exemples de comportement) sont constitués de couples de valeurs du type : (valeurs d'entrées, valeurs de sorties désirées). Dans ce type d'apprentissage, un superviseur fait apprendre au réseau l'ensemble de ces couples, par une méthode d'apprentissage, comme la rétro-propagation du gradient de l'erreur, en comparant pour chacun d'entre eux la sortie effective du réseau et la sortie désirée. L'apprentissage est terminé lorsque tous les couples entrée/sortie sont reconnus par le réseau. Cet apprentissage se retrouve, par exemple, dans le cas des réseaux perceptrons.

### III-2-5-2. Apprentissage non supervisé

L'apprentissage est qualifié de non supervisé lorsque seules les valeurs d'entrée sont disponibles. Dans ce cas, les informations présentées à l'entrée provoquent une auto-adaptation du réseau afin de produire des valeurs de sortie qui soient proches en réponse à des valeurs d'entrée similaires.

Le choix d'utiliser telle ou telle architecture de réseau de neurones, tel ou tel type d'apprentissage dépend de l'application et aussi des capacités de traitement du système sur lequel ces architectures vont être implantées.

### III-2-6. Algorithme d'apprentissage

Afin d'adapter les différents poids d'un réseau donné, plusieurs méthodes d'apprentissage peuvent être utilisées. Parmi ces méthodes on trouve l'algorithme de rétro-propagation du gradient.

#### III-2-6-1. L'algorithme de rétro-propagation du gradient

C'est l'algorithme le plus connu pour réaliser l'adaptation des réseaux à couches (*feedforward*). Il a été développé par Rumelhart et Parkenet le Cun en 1985. Il s'agit d'une méthode d'apprentissage supervisé.

Le principe de la rétro-propagation consiste à présenter au réseau un vecteur d'entrées, de procéder au calcul de la sortie par propagation à travers les couches, de l'entrée vers la sortie passant par les couches cachées. Cette sortie obtenue est comparée à la sortie désirée, une erreur est alors obtenue. A partir de cette erreur, est calculé le gradient de l'erreur qui est à son tour propagé de la couche de sortie vers la couche d'entrée, d'où le terme de rétro-propagation.

Pour décrire cet algorithme, considérons le couple de données entrées/sorties  $(x(k), y(k))$  de la  $k^{\text{ème}}$  donnée d'entraînement du réseau avec :

$$x(k) = \{x_1(k), \dots, x_n(k)\} \text{ et } y(k) = \{y_1(k), \dots, y_q(k)\} \quad (\text{III-8})$$

L'objectif de cet algorithme est d'adapter les poids des connexions du réseau de manière à minimiser la somme des erreurs sur tous les neurones de la couche considérée (cas de la couche de sortie ou celle cachée).

Cas de la couche de sortie

Soit le schéma de la figure III-16, représentant le  $j^{ème}$  neurone de la couche de sortie d'un réseau donné.

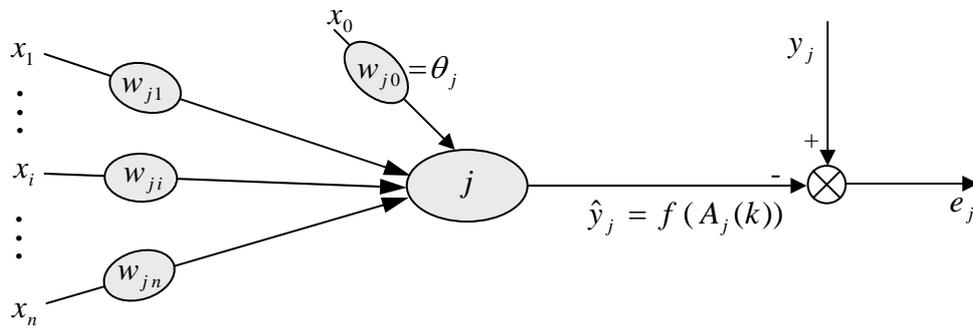


Figure III-16 : Modèle du neurone  $j$

avec :  $y_j$  est la sortie désirée du neurone  $j$ .

$\hat{y}_j$  est la sortie estimée du neurone  $j$ .

$A_j(k) = \sum_{i=0}^n w_{ji}(k) x_i(k)$  est l'activation ou la somme pondérée des entrées du neurone  $j$ .

$f$  est la fonction d'activation du neurone  $j$ .

$w_{ji}$  est le poids de la connexion entre le neurone  $i$  de la couche précédente et le neurone  $j$  de la couche courante.

$x_i$  est la sortie du neurone  $i$ .

On suppose que :

$w_{j0}(k)$  correspond au biais du neurone  $j$  et  $x_0 = -1$ .

L'erreur mesurée sur le  $j^{ème}$  neurone de sortie est donnée comme suit :

$$e_j(k) = y_j(k) - \hat{y}_j(k) \tag{III-9}$$

L'objectif de l'algorithme est d'adapter les poids des connexions du réseau de manière à minimiser la somme des erreurs sur tous les neurones de la couche de sortie.

Soit  $E(k)$  l'erreur quadratique globale sur l'ensemble  $C$  des neurones de sortie :

$$E(k) = \frac{1}{2} \sum_{j \in C} e_j^2(k) \tag{III-10}$$

Pour corriger l'erreur mesurée, il s'agit de modifier le poids  $w_{ji}(k)$  dans le sens opposé au gradient de l'erreur, en suivant la règle :

$$\Delta w_{ji}(k) = -\eta \frac{\partial E(k)}{\partial w_{ji}(k)} \tag{III-11}$$

Où  $\eta$  est une constante positive appelée pas du gradient.

Le calcul de (III-11) est décomposé, utilisant la règle de chaînage des dérivées partielles qui nous dit que  $\frac{\partial f(y)}{\partial x} = \frac{\partial f(y)}{\partial y} \frac{\partial y}{\partial x}$ , de la manière suivante :

$$\frac{\partial E(k)}{\partial w_{ji}(k)} = \frac{\partial E(k)}{\partial e_j(k)} \cdot \frac{\partial e_j(k)}{\partial \hat{y}_j(k)} \cdot \frac{\partial \hat{y}_j(k)}{\partial w_{ji}(k)} \quad (\text{III-12})$$

L'évaluation des différents termes de l'équation (III-12) est :

- $\frac{\partial E(k)}{\partial e_j(k)} = \frac{\partial \left[ \frac{1}{2} \sum_{l \in C} e_l^2(k) \right]}{\partial e_j(k)} = \frac{1}{2} \cdot \frac{\partial e_j^2(k)}{\partial e_j(k)} = e_j(k)$
- $\frac{\partial e_j(k)}{\partial \hat{y}_j(k)} = \frac{\partial (y_j(k) - \hat{y}_j(k))}{\partial \hat{y}_j(k)} = -1$
- $\frac{\partial \hat{y}_j(k)}{\partial w_{ji}(k)} = \frac{\partial [f(A_j(k))]}{\partial w_{ji}(k)} = f'(A_j(k)) \cdot \frac{\partial A_j(k)}{\partial w_{ji}(k)} = f'(A_j(k)) \cdot \frac{\partial \left[ \sum_{l=0}^n w_{jl}(k) \cdot x_l(k) \right]}{\partial w_{ji}(k)}$   
 $= f'(A_j(k)) \cdot x_i(k)$

Par le remplacement de ces données dans l'équation (III-12), nous obtenons :

$$\frac{\partial E(k)}{\partial w_{ji}(k)} = -e_j(k) \cdot f'(A_j(k)) \cdot x_i(k) \quad (\text{III-13})$$

Finalement, l'expression de la variation du poids  $w_{ji}(k)$  est exprimé par :

$$\Delta w_{ji}(k) = \eta \cdot \delta_j(k) \cdot x_i(k) \quad (\text{III-14})$$

où :  $\delta_j(k) = e_j(k) \cdot f'(A_j(k))$

### Cas d'une couche cachée

Considérons le cas des neurones sur la dernière couche cachée (même raisonnement pour les autres couches cachées). Le problème posé dans ce cas est qu'on ne dispose plus de l'erreur mesurée. Pour procéder au calcul du gradient dans ces conditions, on peut passer à la décomposition de l'équation (III-11) de la manière suivante :

$$\frac{\partial E(k)}{\partial w_{ji}(k)} = \frac{\partial E(k)}{\partial \hat{x}_j(k)} \cdot \frac{\partial \hat{x}_j(k)}{\partial A_j(k)} \cdot \frac{\partial A_j(k)}{\partial w_{ji}(k)} \quad (\text{III-15})$$

Avec  $i$  et  $j$  des indices désigneront respectivement un neurone sur la couche précédente et un neurone sur la couche courante.

Le calcul des différents termes de l'équation (III-15) se fait, comme le cas précédent, de la façon suivante :

- Pour le premier terme, nous avons :

$$\frac{\partial E(k)}{\partial \hat{x}_j(k)} = \frac{\partial \left[ \frac{1}{2} \sum_{l \in C} e_l^2(k) \right]}{\partial \hat{x}_j(k)} \quad (\text{III-16})$$

Puisque les erreurs  $e_l(k)$  dépendent, dans ce cas, de  $x_j(k)$ , l'expression (III-16) est calculée de la manière suivante :

$$\begin{aligned} \frac{\partial E(k)}{\partial \hat{x}_j(k)} &= \frac{\partial \left[ \frac{1}{2} \sum_{l \in C} e_l^2(k) \right]}{\partial \hat{x}_j(k)} = \sum_{l \in C} \left[ e_l(k) \cdot \frac{\partial e_l(k)}{\partial \hat{x}_j(k)} \right] = \sum_{l \in C} \left[ e_l(k) \cdot \frac{\partial e_l(k)}{\partial A_l(k)} \cdot \frac{\partial A_l(k)}{\partial \hat{x}_j(k)} \right] \\ &= \sum_{l \in C} \left[ e_l(k) \cdot \frac{\partial [x_l(k) - f(A_l(k))]}{\partial A_l(k)} \cdot \frac{\partial \sum_i w_{li}(k) \cdot x_i(k)}{\partial \hat{x}_j(k)} \right] = \sum_{l \in C} \left[ e_l(k) \cdot f'(A_l(k)) \cdot w_{lj}(k) \right] \end{aligned}$$

Donc :

$$\frac{\partial E(k)}{\partial \hat{x}_j(k)} = \sum_{l \in C} \delta_l(k) \cdot w_{lj}(k) \quad (\text{III-17})$$

Où :  $\delta_l(k) = e_l(k) \cdot f'(A_l(k))$

- Pour le deuxième terme :

$$\frac{\partial \hat{x}_j(k)}{\partial A_j(k)} = \frac{\partial f(A_j(k))}{\partial A_j(k)} = f'(A_j(k))$$

- Pour le troisième terme :

$$\frac{\partial A_j(k)}{\partial w_{ji}(k)} = \frac{\partial (\sum_l w_{jl(k)} \cdot x_l(k))}{\partial w_{ji}(k)} = x_j(k)$$

En substituant les différents termes calculés précédemment dans l'équation (III-15), nous obtenons :

$$\frac{\partial E(k)}{\partial w_{ji}(k)} = -f'(A_j(k)) \cdot \left[ \sum_{l \in C} \delta_l(k) \cdot w_{lj}(k) \right] \cdot x_j(k)$$

Et :

$$\Delta w_{ji}(k) = -\eta \cdot \frac{\partial E(k)}{\partial w_{ji}(k)} = \eta \cdot \delta_j(k) \cdot x_j(k) \quad (\text{III-18})$$

Où :  $\delta_j(k) = f'(A_j(k)) \cdot \left[ \sum_{l \in C} \delta_l(k) \cdot w_{lj}(k) \right]$

La mise à jour des poids dans les deux cas, couche de sortie et couches cachées, est donnée par :

$$w_{ji}(k) = w_{ji}(k-1) + \Delta w_{ji}(k) = w_{ji}(k-1) + \eta \cdot \delta_j(k) \cdot x_j(k) \quad (\text{III-19})$$

Avec :  $\delta_j(k) = \begin{cases} e_j(k) \cdot f'(A_j(k)) & \text{Si } j \in \text{Couche de sortie.} \\ f'(A_j(k)) \cdot \sum_{l \in C} \delta_l(k) \cdot w_{lj}(k) & \text{Si } j \in \text{Couche cachée.} \end{cases}$

D'après cet algorithme, on peut remarquer que la vitesse avec laquelle se fait la mise à jour des poids lors de l'apprentissage est choisie selon les valeurs de  $\eta$ , qui représente le facteur d'apprentissage.

Le choix de  $\eta$  ne doit pas être trop grand sinon il entraînerait des oscillations de l'erreur autour d'un minimum qu'on ne pourra pas atteindre et si  $\eta$  est trop petit le temps d'apprentissage serait trop grand.

Une autre variante de cet algorithme permet l'introduction d'un terme inertiel qui aide à la convergence [13] :

$$w_{ji}(k) = w_{ji}(k-1) + \Delta w_{ji}(k) + \alpha \cdot w_{ji}(k-1) \quad (\text{III-20})$$

### Remarque

Cette algorithme possède aussi d'autres variantes correspondent généralement à l'utilisation de valeurs variables pour la constante  $\eta$ , ou à l'utilisation de méthodes du deuxième ordre pour le calcul du gradient [13].

## III-2-7. Applications des réseaux de neurones

Les réseaux de neurones sont beaucoup plus utilisés dans le domaine de l'automatique, soit pour l'identification, soit pour le contrôle et la commande de processus.

### III-2-7-1. Identification par réseaux de neurones

L'identification d'un système consiste à proposer un modèle établissant une relation entre son entrée et sa sortie et à déterminer, à partir du couple des signaux d'entrée-sortie, le comportement du modèle.

#### III-2-7-1-1. Principe de l'identification

Comme pour le cas des méthodes classiques, la méthodologie de l'identification par les réseaux de neurones passe par quatre étapes :

- Choix de la structure du réseau.
- Choix de l'entrée.
- Algorithme d'apprentissage.
- Validation.

Parmi les techniques d'identification à base de réseaux de neurones multicouches, nous présenterons deux méthodes :

#### III-2-7-1-2. Identification directe

Le principe de la méthode d'identification directe (*forward modeling*) est montré sur la figure III-17. Sur cette figure, le réseau de neurones identificateur est utilisé en parallèle avec un processus de type boîte noire. La sortie  $y$  du processus est comparée avec la sortie  $\hat{y}$  du réseau de neurones, puis l'erreur  $e = y - \hat{y}$  est utilisée afin d'affiner les paramètres du système.

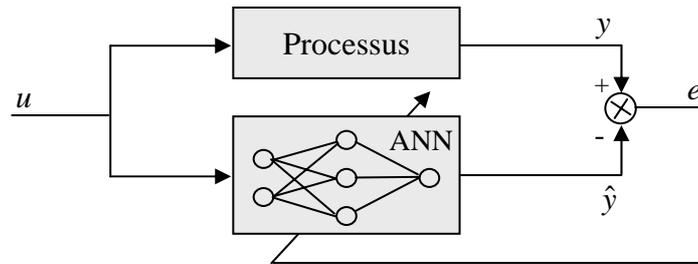


Figure III-17 : Principe de l'identification neuronale directe d'un processus

### III-2-7-1-3. Identification inverse

Dans ce cas, l'entrée du processus est comparée avec la sortie de l'identificateur neuronal et la sortie du processus est injectée comme entrée du réseau de neurones. Le principe de cette méthode est donné par la figure III-18.

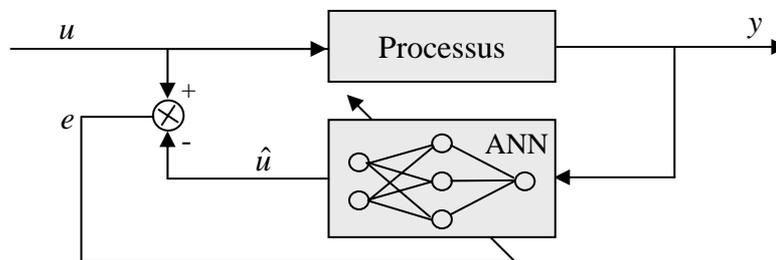


Figure III-18: Principe de l'identification neuronale inverse d'un processus

Après un apprentissage hors-ligne du modèle inverse, le réseau de neurones identificateur peut être configuré afin d'assurer un contrôle direct du processus.

### III-2-7-2. Commande neuronale

Plusieurs architectures de commande sont mentionnées dans la littérature scientifique. Les plus simples se basent sur l'apprentissage d'un contrôleur conventionnel déjà existant, d'autres opèrent un apprentissage hors-ligne du modèle inverse du processus ou d'un modèle de référence et enfin, d'autres travaillent complètement en ligne.

#### III-2-7-2-1. Commande inverse

La commande inverse avec un réseau de neurones contrôleur est représentée sur la figure III-19. Cette architecture reprend le même principe que celui de l'identification inverse montrée dans la figure III-18. En effet, l'entrée du réseau de neurone est la référence  $r$  qui sera comparée à la sortie  $y$  du processus. Si le réseau de neurones apprend le modèle inverse du processus, l'erreur  $e = r - y$  sera nulle et donc  $y$  sera égale à  $r$ . La sortie  $u$  du réseau de neurones est la commande injectée en entrée du processus. Ce principe est identique au réseau de neurones utilisé en identification de la figure III-18 où lorsque l'apprentissage du modèle inverse est accompli, la sortie du réseau de neurone est égale à l'entrée du processus.

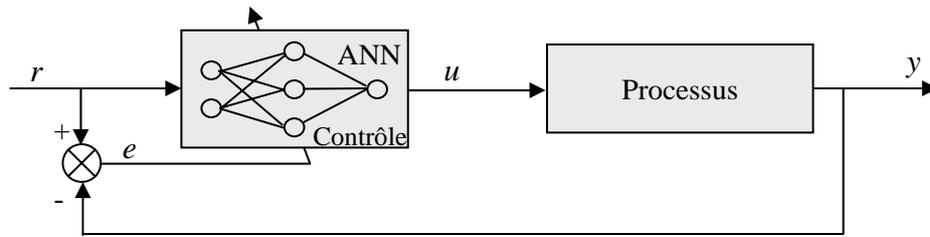


Figure III-19: Principe de la commande inverse avec un réseau de neurones

L'avantage de cette commande est le suivi en temps réel de l'évolution du processus, car l'apprentissage est réalisé en ligne.

**III-2-7-2-2. Commande neuronale hybride**

Cette architecture est utilisée pour ajuster la commande en sortie d'un contrôleur conventionnel. La figure III-20 montre un réseau de neurones monté en parallèle avec un correcteur classique (PD, PID, etc.). Le signal de commande  $\hat{u}$  qui est la sortie du réseau de neurones est utilisé pour corriger le signal de commande  $u$  à la sortie du correcteur conventionnel, et cela afin de minimiser l'erreur  $e$ , différence entre le signal de référence  $r$  et la sortie du processus  $y$ . Le but principal du correcteur neuronal dans ce cas, est de pallier à l'insuffisance d'adaptation de la commande classique qui elle-même permet une meilleur convergence des poids du réseau de neurones.

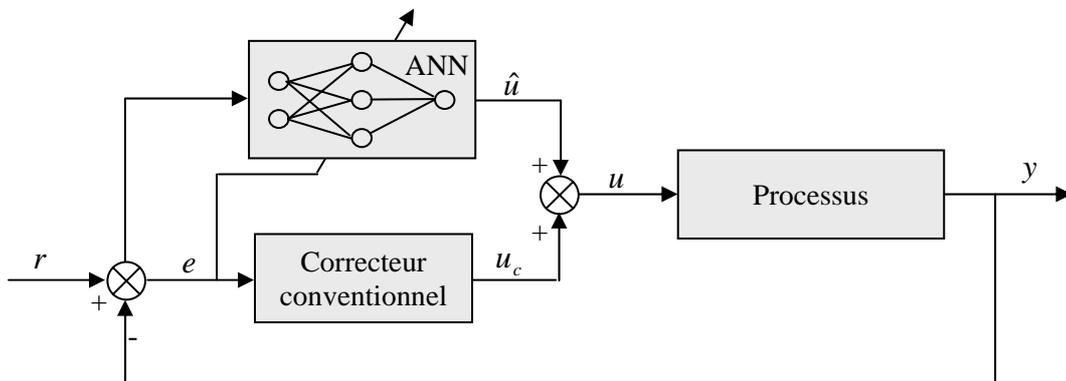


Figure III-20: Structure de la commande neuronale hybride

Pour réaliser cette architecture, on définit d'abord la régulation avec le correcteur conventionnel (sans correcteur neuronal) puis on ajoute le réseau de neurones afin d'améliorer la commande  $u = u_c + \hat{u}$  à l'entrée du processus et pour travailler en ligne [17].

**III-2-7-2-3. Commande par modèle de référence**

Dans ce cas les paramètres du réseau de neurones sont ajustés pour minimiser l'erreur  $e$  entre la sortie du modèle de référence  $y_m$  et celle du processus  $y_p$  comme le montre la figure III-21. En d'autres termes, le réseau de neurones est synthétisé afin que la sortie actuelle  $y_p$  atteigne la sortie désirée  $y_m$ .

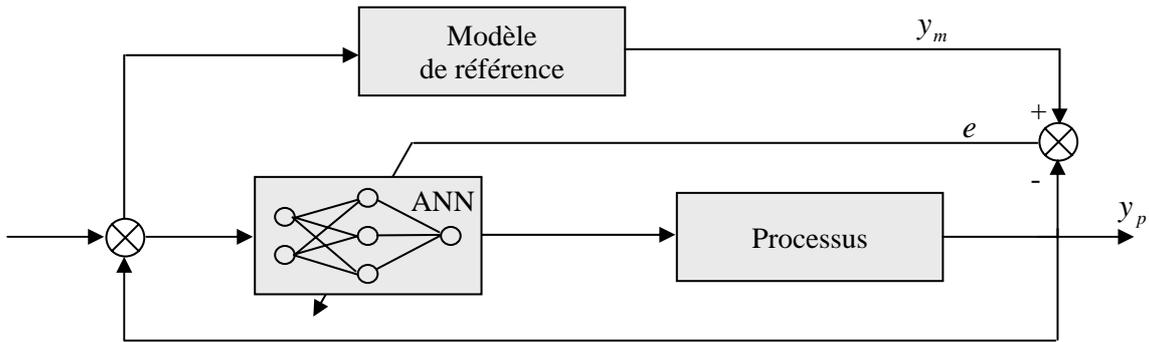


Figure III-21: Principe de la commande neuronale avec modèle de référence

**III-2-7-2-4. Commande par modèle interne (IMC)**

La figure III-22 présente le principe de la commande à modèle interne (*Internal Model Control*). Comme son nom l'indique, la commande IMC met en jeu, outre un contrôleur, un modèle de processus, dit modèle interne ; l'erreur de modélisation est utilisée pour modifier la consigne, si bien que le système est robuste aux erreurs de modélisation, ce qui n'est pas le cas pour la commande inverse.

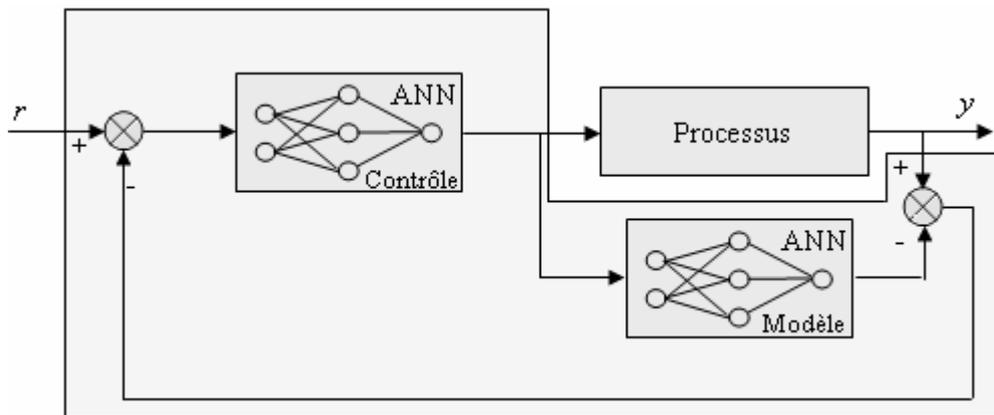


Figure III-22: Principe de la commande par modèle interne

**III-2-8. Systèmes neuro-flous**

L'utilisation conjointe des réseaux de neurones et de la logique floue permet de tirer les avantages des deux méthodes ; les capacités d'apprentissage de la première et la lisibilité et la souplesse de la seconde. Diverses combinaisons de ces deux méthodes ont été développées depuis 1988. Elles ont donné naissance aux systèmes neuro-flous, qui sont le plus souvent orientés vers la commande de système complexes et les problèmes de classification.

La figure III-23, résume le principe du système neuro-flou qui représente l'intersection entre la logique floue et les réseaux de neurones.

Le système neuro-flou peut être interprété comme étant un réseau neuronal multicouche avec des paramètres flous, ou comme un système flou mis en application sous une forme distribuée parallèle. Nous limiterons, dans la suite, à la présentation du deuxième cas et en particulier le modèle ANFIS.

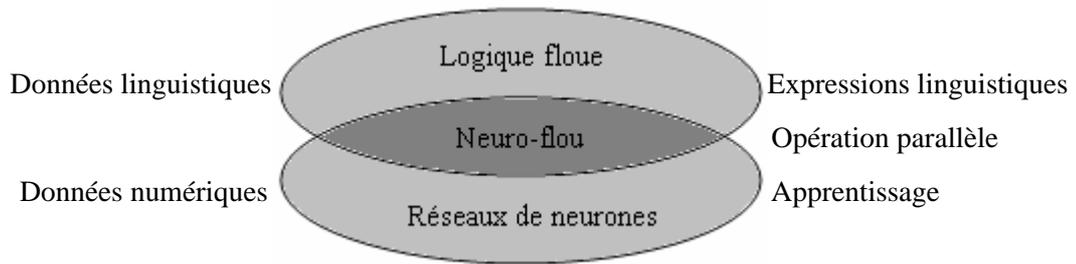


Figure III-23: Principe du système neuro-flou

**III-2-8-1. ANFIS (Adaptive Neuro Fuzzy Inference System)**

Le modèle ANFIS présente une technique d'optimisation des systèmes d'inférence floue du type Takagi Sugeno. C'est le modèle le plus utilisé en pratique. Il utilise, dans son architecture, des réseaux multicouches. Ces réseaux ont pour particularité le fait de posséder des cellules réalisant une fonction qui leurs est propre en respectant les paramètres qui leurs ont été fournis [10], [15], [16].

Pour décrire le principe de l'ANFIS, considérons le système d'inférence floue qui contient deux entrées  $x$  et  $y$  et une seule sortie  $z$  donné par le schéma de la figure III-24.

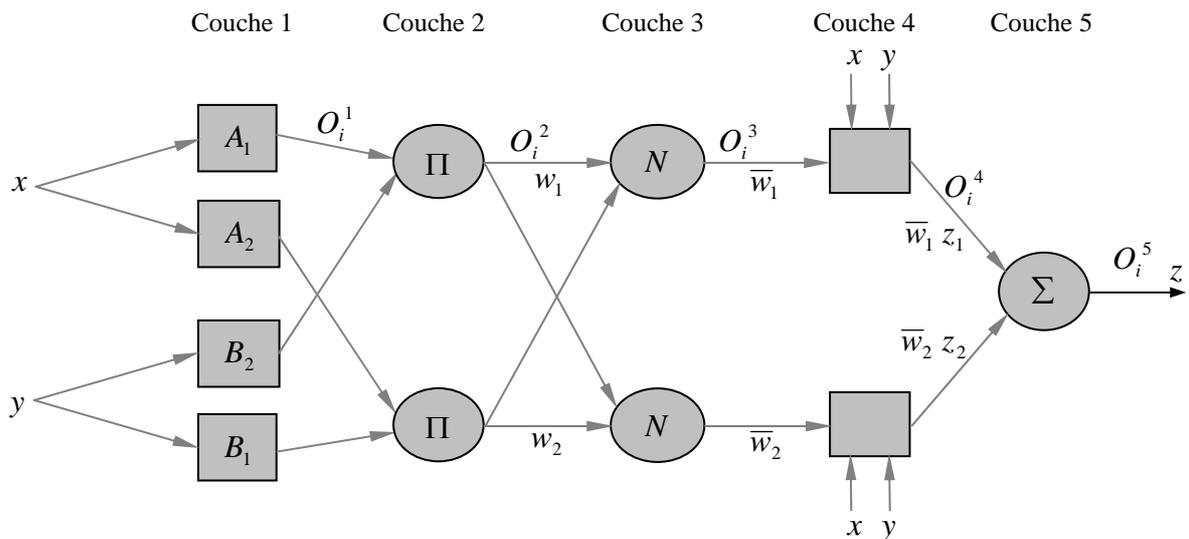


Figure III-24: Architecture de l'ANFIS

L'architecture de l'ANFIS dépend de deux ensembles de paramètres d'entraînement ; les paramètres des fonctions d'appartenance prémisses et les paramètres conséquents.

L'apprentissage de l'ANFIS utilise un algorithme de gradient descendant pour optimiser les paramètres prémisses et l'algorithme des moindres carrés pour résoudre les paramètres conséquents. Le système ANFIS peut être résumé comme suit :

$$\text{Règle } i : \text{ si } x \text{ est } A_i \text{ et } y \text{ est } B_i \text{ alors } z_i = p_i x + q_i y + r_i$$

Où  $A_i$  et  $B_i$  représentent les ensembles flous associés au nœud de fonction.

Comme dans la figure III-24, l'ANFIS est composé de cinq couches.

**Couche 1 :** Chaque nœud  $i$  dans cette couche utilise une fonction d'appartenance donnée par :

$$O_i^1 = \mu_{A_i}(x)$$

Avec  $x$  est l'entrée du nœud  $i$  (même chose pour l'entrée  $y$ ,  $O_i^1 = \mu_{B_i}(y)$ )

$\mu_{A_i}$  représente la fonction d'appartenance de  $A_i$  et il spécifie le degré d'appartenance avec lequel  $x$  satisfait.

Généralement  $\mu_{A_i}(x)$  est choisi sous forme d'une fonction gaussienne.

$$\mu_{A_i}(x) = \exp\left[-\frac{(x - c_i)^2}{\sigma_i}\right]$$

Avec  $c_i$ ,  $\sigma_i$  sont des paramètres qui font référence aux paramètres prémisses. Les valeurs de ces derniers changent selon diverse exhibition de la fonction d'appartenance.

**Couche 2 :** Les sorties de cette couche sont les poids des règles, ils sont obtenus par une simple multiplication des entrées dans chaque cellule.

$$O_i^2 = w_i = \mu_{A_i}(x) \times \mu_{B_i}(y)$$

**Couche 3 :** Cette couche correspond à la normalisation des poids des règles par la relation :

$$O_i^3 = \bar{w}_i = \frac{w_i}{w_1 + w_2}, \quad i = 1, 2.$$

**Couche 4 :** chaque nœud  $i$  dans cette couche est un nœud qui se calcule de la façon suivante :

$$O_i^4 = \bar{w}_i z_i = \bar{w}_i (p_i x + q_i y + r_i)$$

Où :  $\bar{w}_i$  sont les sorties de la couche 3 et  $(p_i, q_i, r_i)$  sont les paramètres conséquents de la fonction de sortie.

**Couche 5 :** Le nœud simple dans cette couche fait la somme de tous les signaux d'entrées et retourne, en sortie, la valeur approximative de la fonction désirée exprimée par :

$$O_i^5 = \sum_i \bar{w}_i z_i = \frac{\sum_{i=1}^2 w_i z_i}{\sum_{i=1}^2 w_i}$$

Dans l'architecture de l'ANFIS, nous constatons que nous avons des paramètres prémisses en entrée et des paramètres conséquents exprimés en combinaison linéaire en sortie.

$$\begin{aligned} z &= \frac{w_1}{w_1 + w_2} z_1 + \frac{w_2}{w_1 + w_2} z_2 \\ &= \bar{w}_1 z_1 + \bar{w}_2 z_2 \\ &= (\bar{w}_1 x) p_1 + (\bar{w}_1 y) q_1 + (\bar{w}_1) r_1 + (\bar{w}_2 x) p_2 + (\bar{w}_2 y) q_2 + (\bar{w}_2) r_2 \end{aligned}$$

Avec  $p_i$ ,  $q_i$  et  $r_i$  sont des paramètres conséquents linéaires.

Les paramètres prémisses sont identifiés par la méthode de descente de gradient et les paramètres conséquents sont reconnus par la méthode des moindres carrés. Les paramètres conséquents ainsi identifiés sont optimaux à la condition que les paramètres prémisses sont fixés. Le choix de ces méthodes est basé sur l'efficacité entre la complexité du calcul et la performance du résultat. En conséquent, l'approche hybride est plus facile que la descente du gradient et c'est facile de voir la possibilité de décomposer l'ensemble des paramètres [16].

### III-3. Application à la commande du bras manipulateur Puma 600

Nous avons vu, dans le chapitre précédent, que le système en boucle fermée avec la loi de commande *Couple calculé* classique, dans le cas où le modèle du robot est connu avec exactitude, est asymptotiquement stable. Dans le cas d'une connaissance imprécise des paramètres du robot et/ou d'une présence de certaines dynamiques non modélisées, la commande par *Couple calculé* montre ses limites.

Afin de montrer l'apport des commandes intelligentes : par la logique floue, par réseau de neurones et par neuro-flou et leurs améliorations par rapport à la méthode *Couple calculé* classique, nous avons effectué des simulations sur le modèle dynamique du robot manipulateur Puma 600, en se basant sur les résultats obtenus dans le deuxième chapitre.

#### III-3-1. Commande par *Couple calculé* basé sur la logique floue

Dans ce cas nous proposons une solution qui consiste à utiliser un régulateur flou dont le rôle est d'adapter les valeurs des coefficients de  $K_p$  et de  $K_v$ , pour compenser les parties négligées du modèle dynamique du robot tout en gardant l'architecture classique de la commande *Couple calculé* [18].

##### III-3-1-1. Stratégie de la commande floue

Toujours dans le but de faire suivre le robot, dans l'espace articulaire, une trajectoire désirée  $q_d(t)$  et ses dérivées successives  $\dot{q}_d(t)$  et  $\ddot{q}_d(t)$  qui décrivent respectivement la vitesse désirée et l'accélération désirée, la stratégie de la commande floue consiste à donner des valeurs aux gains du correcteur d'une façon progressive contrairement à celui du correcteur classique qui sont fixes.

La structure de base de notre contrôleur flou est donnée par la figure III-25.

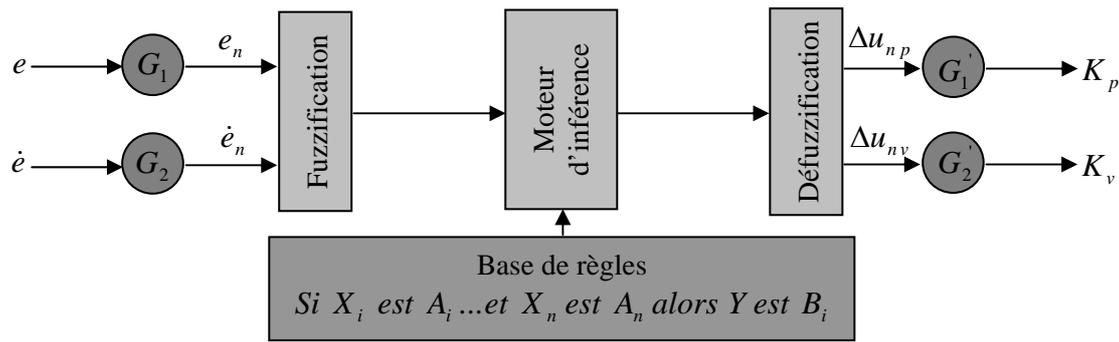


Figure III-25 : Schéma de principe du régulateur flou proposé

Avec :  $G_1$  et  $G_2$  représentent les gains de normalisation (facteurs d'échelles) transformant les valeurs physiques ( $e, \dot{e}$ ) des entrées en des valeurs normalisées ( $e_n, \dot{e}_n$ ) appartenant à l'intervalle  $[-1 \ 1]$ . Les gains  $G_1'$  et  $G_2'$  permettent de dénormaliser les valeurs normalisées des variables de commande ( $\Delta u_{np}, \Delta u_{nv}$ ) en des valeurs réelles ( $K_p, K_v$ ).

Ce contrôleur flou comporte les trois blocs : Fuzzification des variables d'entrées par l'utilisation des *fonctions triangulaires*, ensuite l'inférence où ces variables fuzzifiées sont comparées avec des ensembles prédéfinis pour déterminer la réponse appropriée. Et enfin la défuzzification pour convertir les sous-ensembles fuzzifiés en valeurs réelles en utilisant la *défuzzification centroïde*.

**A. Méthode d'inférence / base de règles**

Les règles d'inférences que nous avons adoptées pour notre régulateur sont les suivantes :

Règle 1 : Si  $e$  est  $P$  Alors  $K_i$  est  $P$ .

Ce qui veut dire que : Si la trajectoire du robot s'éloigne d'une façon importante, dans le sens positif, de la trajectoire désirée Alors augmenter au maximum le gain de commande (Le même raisonnement pour les autres règles).

Règle 2 : Si  $e$  est  $PP$  Alors  $K_i$  est  $PP$

Règle 3 : Si  $e$  est  $Z$  Alors  $K_i$  est  $Z$

Règle 4 : Si  $e$  est  $PN$  Alors  $K_i$  est  $PN$

Règle 5 : Si  $e$  est  $N$  Alors  $K_i$  est  $N$

où les symboles  $P, PP, Z, PN$  et  $N$  sont des variables linguistique qui signifient respectivement : positif, petit positif, zéro, petit négatif et négatif.

Nous avons choisi dans notre cas la méthode d'inférence Max/Min de Mamdani, qui consiste à réaliser l'opérateur «ET» et la conclusion «ALORS» de chaque règle par la fonction Min et l'opérateur «OU» de la liaison entre toutes les règles par la fonction Max et la défuzzification par centre de gravité (centroïde).

La figure III-26 représente, la fonction d'appartenance servant à fuzzifier les entrées et défuzzifier les variables linguistiques de sorties.

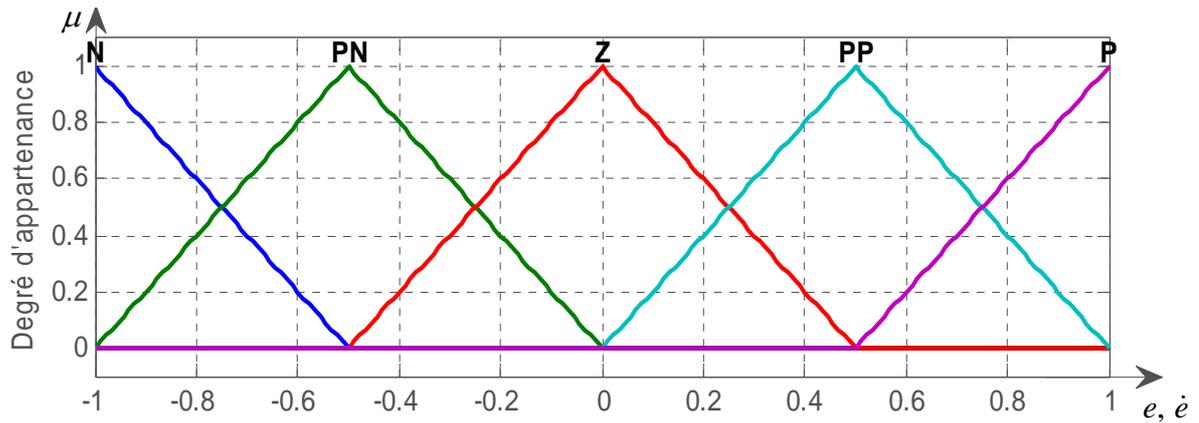


Figure III-26 : fonction d'appartenance des entrées et des sorties du régulateur flou

Dans cette application, nous avons considéré la même trajectoire de référence ( $q_d$ ) et les mêmes paramètres ( $K_p$ ,  $K_v$ ) utilisés dans le cas de la commande Couple calculé classique présentée dans le chapitre précédent.

La figure III-27, donne la réalisation du schéma bloc de la commande Couple Calculé type PD basée sur la logique floue utilisée en simulation.

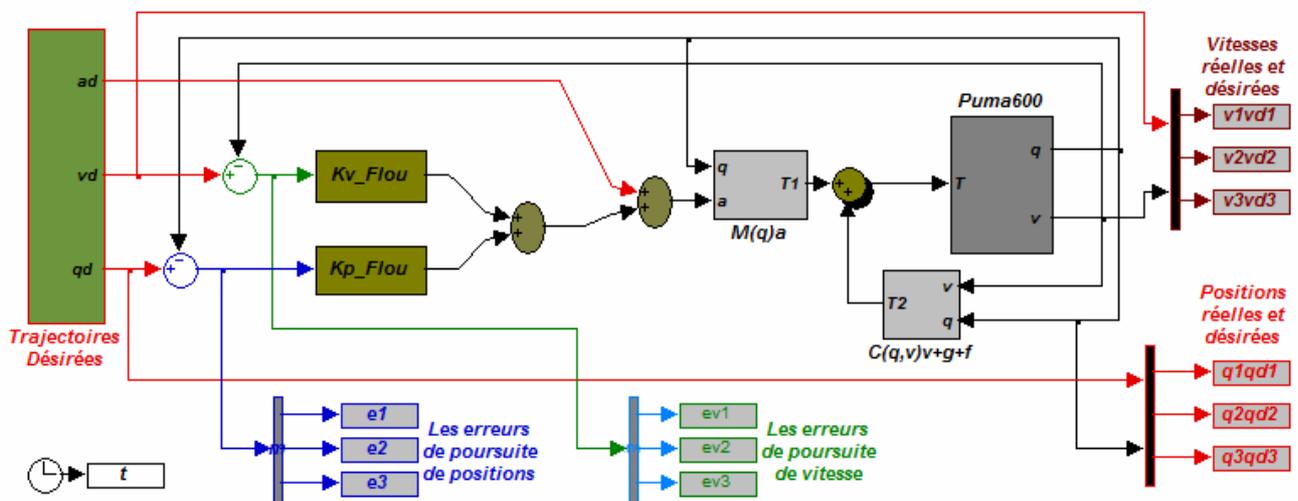
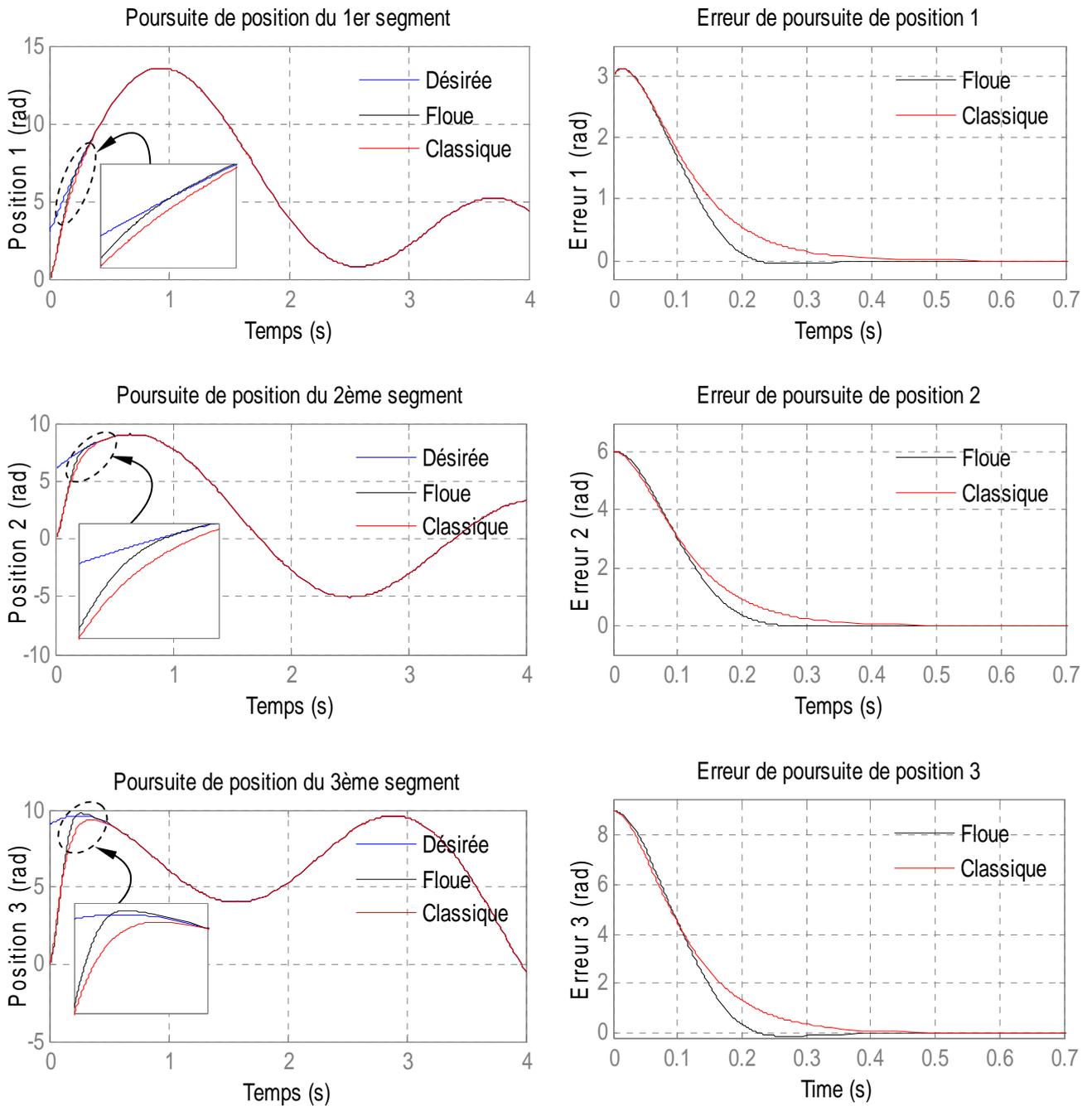


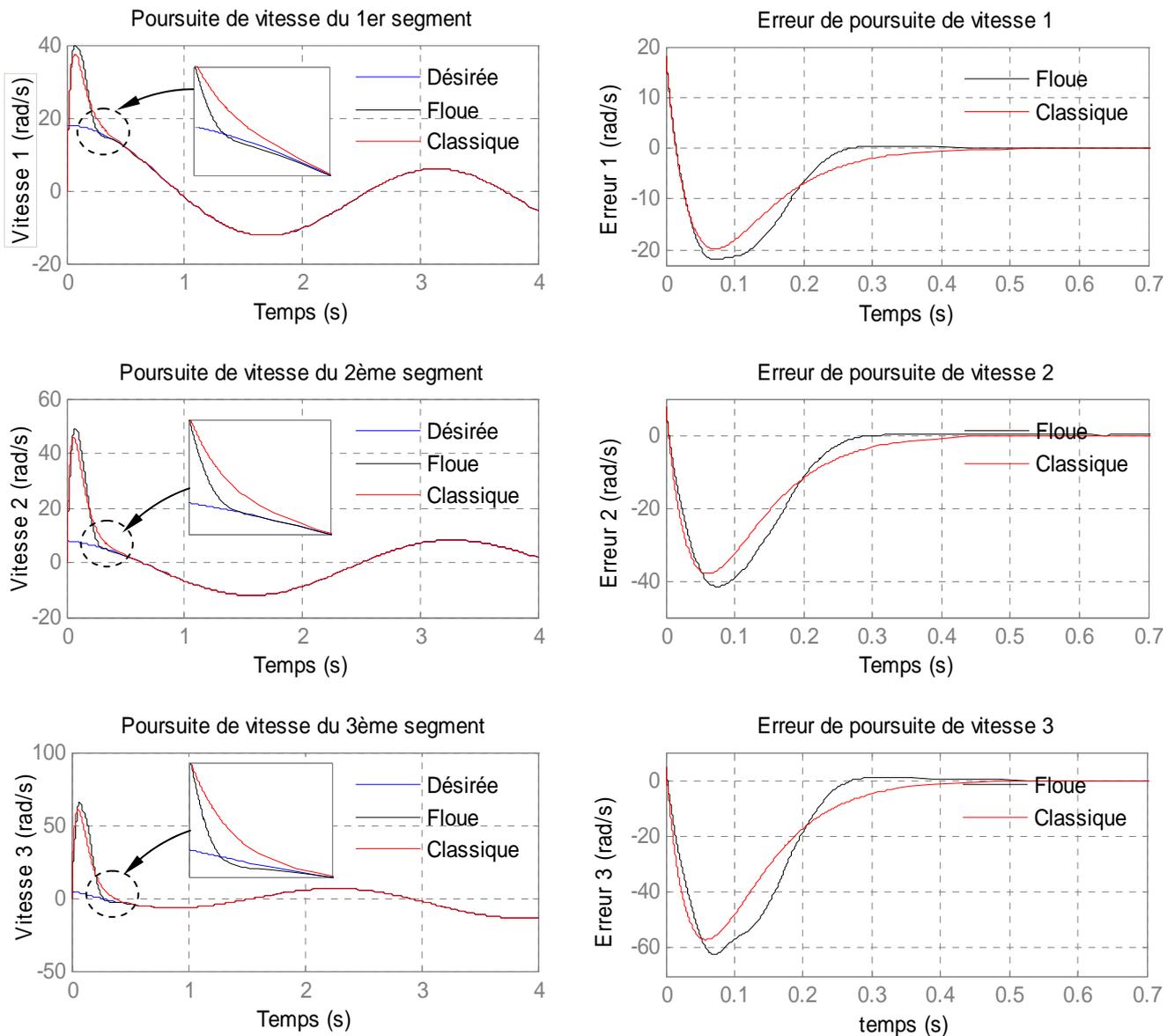
Figure III-27 : Schéma d'implémentation de la commande Couple Calculé de type PD basée sur la logique floue sous Matlab/Simulink

La figure III-28, montre le comportement du robot en poursuite de la trajectoire désirée dans les deux cas de la commande du Couple calculé, classique et par la logique floue.



**Figure III-28 :** Les poursuites de trajectoire de position et les erreurs de poursuite de position pour la loi de commande Couple Calculé de type *PD* basée sur la logique floue

Le comportement du robot en poursuite de la vitesse désirée dans les deux cas de la commande du Couple calculé, classique et par la logique floue, est résumé dans la figure III-29.



**Figure III-29 :** Les poursuites de trajectoire de vitesse et les erreurs de poursuite de vitesse pour la loi de commande Couple Calculé de type *PD* basée sur la logique floue

### Analyse et interprétation des résultats

D'après les résultats présentés sur les figures III-28 et III-29, on voit clairement que les performances de la commande Couple calculé basée sur la logique floue sont meilleures que celles de la commande du Couple calculé classique. Ceci est interprété par la convergence plus rapide des erreurs de poursuites de position et de vitesse vers zéro, dans le cas du régulateur flou. Le robot atteint la trajectoire désirée dans un temps moins important, dans le cas de la commande floue, que celui de la commande classique (On trouve les mêmes résultats pour le cas de la poursuite en accélération).

La stratégie de détermination des gains de commande présente, par rapport aux gains fixes, les avantages suivants :

- un grand gain est appliqué seulement lorsque l'écart est très important ;

- si l'écart est faible, le gain sera également faible.

La robustesse de cette commande réside dans son caractère adaptatif pour la détermination des gains du régulateur ce qui veut dire une commande adaptative directe. Ces gains adaptatifs permettent le lissage des signaux de commande évitant ainsi, toute sollicitation des actionneurs.

### III-3-2. Commande par réseaux de neurones

Dans cette application nous avons pensé au remplacement du régulateur flou développé précédemment, qui a donné des résultats convaincants, par un correcteur neuronal tout en gardant l'architecture de la commande Couple calculé de type *PD*.

L'objectif visé dans ce cas est :

- D'étudier la faisabilité tant que du point de vue de la méthode d'apprentissage que de celui de la structure de réseaux de neurones à adopter.
- De tenir compte de la robustesse d'un tel correcteur au sein de la commande envisagé [20].
- De comparer le temps d'exécution requis par le correcteur neuronal et celui équivalent flou.

Pour la réalisation de notre correcteur neuronal, nous avons choisi la structure de type perceptron multicouche, en suivant les étapes suivantes :

- Selon notre analyse, on a fixé un réseau à une seule couche cachée de dix (10) neurones,
- Pour la fonction d'activation, on a choisi une fonction de type sigmoïde,
- L'apprentissage a été réalisé à l'aide de l'algorithme de rétro-propagation du gradient.

Les modèles neuronaux des gains *Kp\_Rés\_Neur* et *Kv\_Rés\_Neur* sont identifiés pour remplacer les gains *Kp\_Flou* et *Kv\_Flou* (voir figure III-30). Chaque modèle utilise une seule entrée *e* (erreur de position) pour le gain proportionnel et  $\dot{e}$  (erreur de vitesse) pour le gain de dérivation ou de vitesse, et une seule sortie qui est l'accélération.

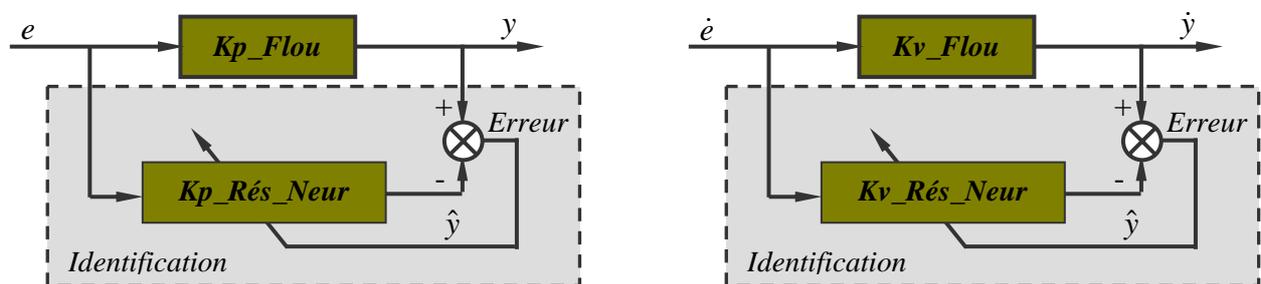


Figure III-30 : Principe d'identification des gains *Kp\_Flou* et *Kv\_Flou* du correcteur par réseau de neurones

La figure III-31, donne la réalisation du schéma bloc de la commande Couple Calculé type *PD* basé sur les réseaux de neurones utilisé en simulation.

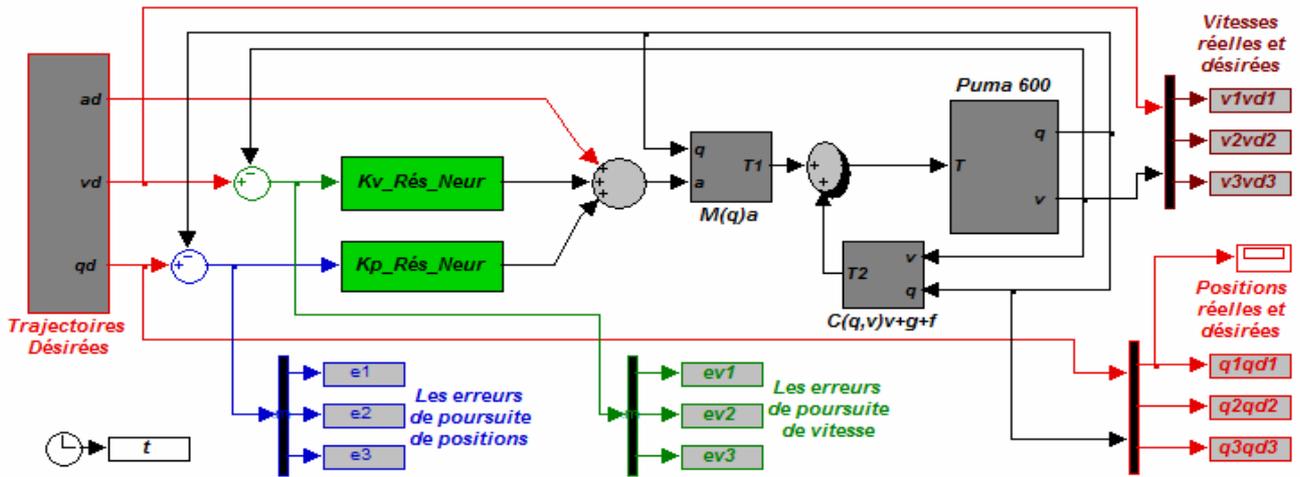


Figure III-31 : Schéma d'implémentation de la commande Couple Calculé de type PD basée sur les réseaux de neurones sous Matlab/Simulink

La figure III-32, montre le comportement du robot en poursuite de la trajectoire désirée dans les deux cas de la commande du Couple calculé, par la logique floue et par réseaux de neurones.

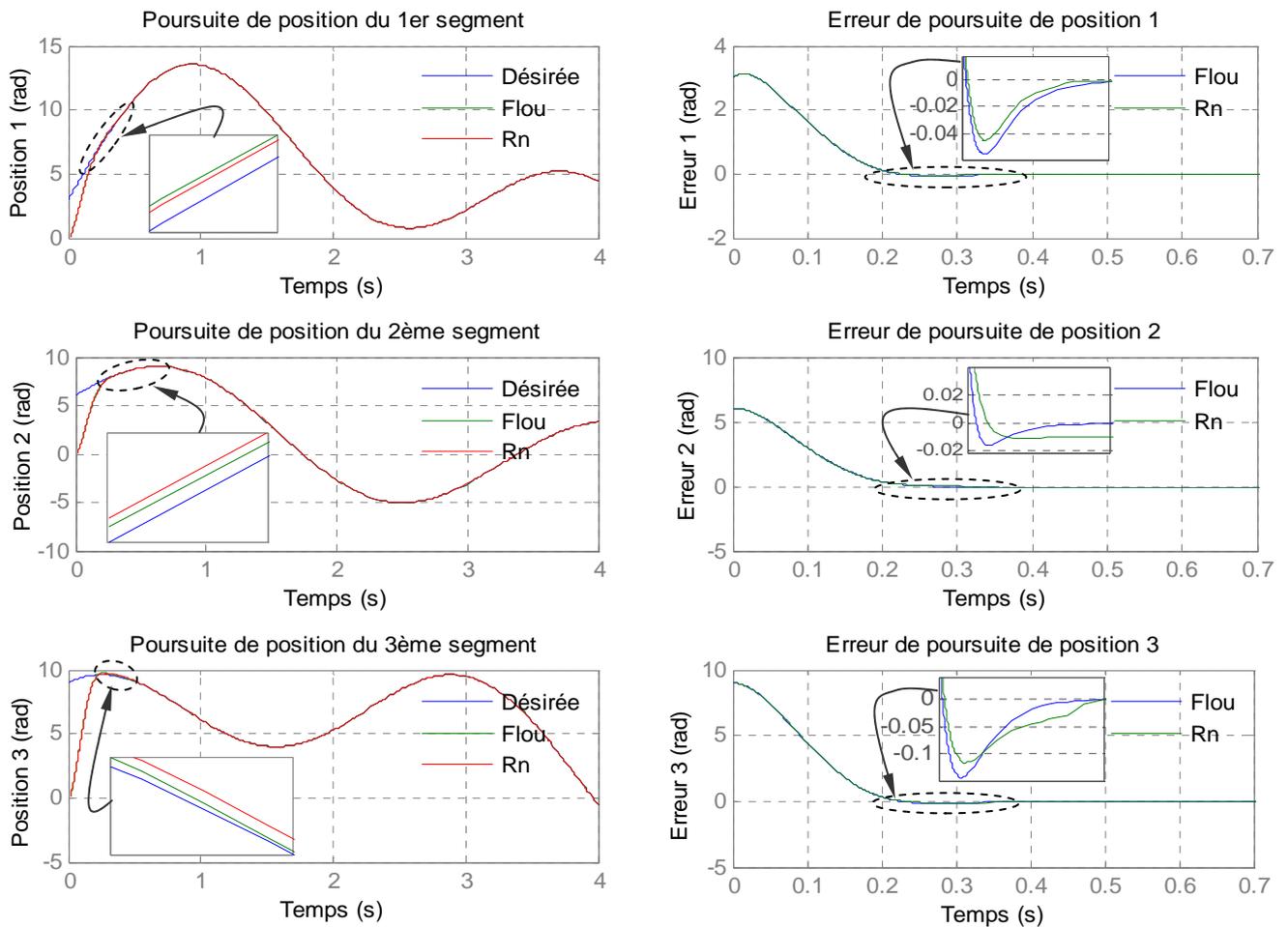


Figure III-32 : Les poursuites de trajectoire de position et les erreurs de poursuite de position pour la loi de commande Couple Calculé de type PD basée sur les réseaux de neurones

### Analyse et interprétation des résultats

Les résultats présentés sur la figure III-32, ne montrent plus d'amélioration sur les performances de notre système à l'exception de la première articulation, ceci est interprété par la nature mécanique de celle-ci qui est plus stable et qui ne présente pas de vibrations. Par contre pour les deux autres articulations, on voit clairement que cette commande est moins performante par rapport à la commande par logique floue. Ce résultat est interprété par la sensibilité mécanique des deux articulations qui est dû au fort couplage de modèle du robot manipulateur.

Ces résultats, nous permettent de conclure que la stratégie de commande par réseaux de neurones est moins robuste que celle basée sur la logique floue.

L'avantage de ce régulateur par rapport à celui du flou réside dans le temps d'exécution minimale dans la boucle de régulation.

### III-3-3. Commande neuro-floue

Toujours dans l'objectif d'améliorer les performances de la méthode de la commande Couple calculé de type *PD*, nous avons procédé au remplacement du régulateur flou développé précédemment par un régulateur neuro-flou (*Adaptive Neuro-Fuzzy Inference system ANFIS*) [19].

La procédure suivie pour le développement de notre régulateur ANFIS est comme suit :

- Choix des variables d'entrées et de sorties : les variables d'entrées choisies sont : l'erreur entre la position désirée et la position réelle du robot ( $e = q_d - q$ ) et la dérivée de l'erreur ou bien l'erreur entre la vitesse désirée et la vitesse réelle du robot ( $\dot{e} = de = \dot{q}_d - \dot{q}$ ). Pour la variable de sortie, on a choisi de calculer l'accélération.
- Choix de la base de données pour l'apprentissage : ce choix est basé sur les résultats des contrôleurs qui ont donné de meilleures performances.
- Choix des fonctions d'appartenance : pour les deux variables d'entrée, la fonction d'appartenance associé à chacune de ces variables est du type triangulaire.
- La technique d'apprentissage choisie étant la méthode hybride.

La figure III-33, donne la structure de l'ANFIS développé.

Avec :

- input : représente les deux variables d'entrées ( $e$ ) et ( $de$ ) ;
- inputmf : représente les fonctions d'appartenances de huit ensembles flous pour chaque entrée ;
- rule : évoque et normalise les poids des règles ;
- outputmf : représente l'étape de défuzzification et la dernière couche donne la somme de toutes les données d'entrées.

Pour cette structure, nous avons choisi le modèle du premier ordre de Sugeno.

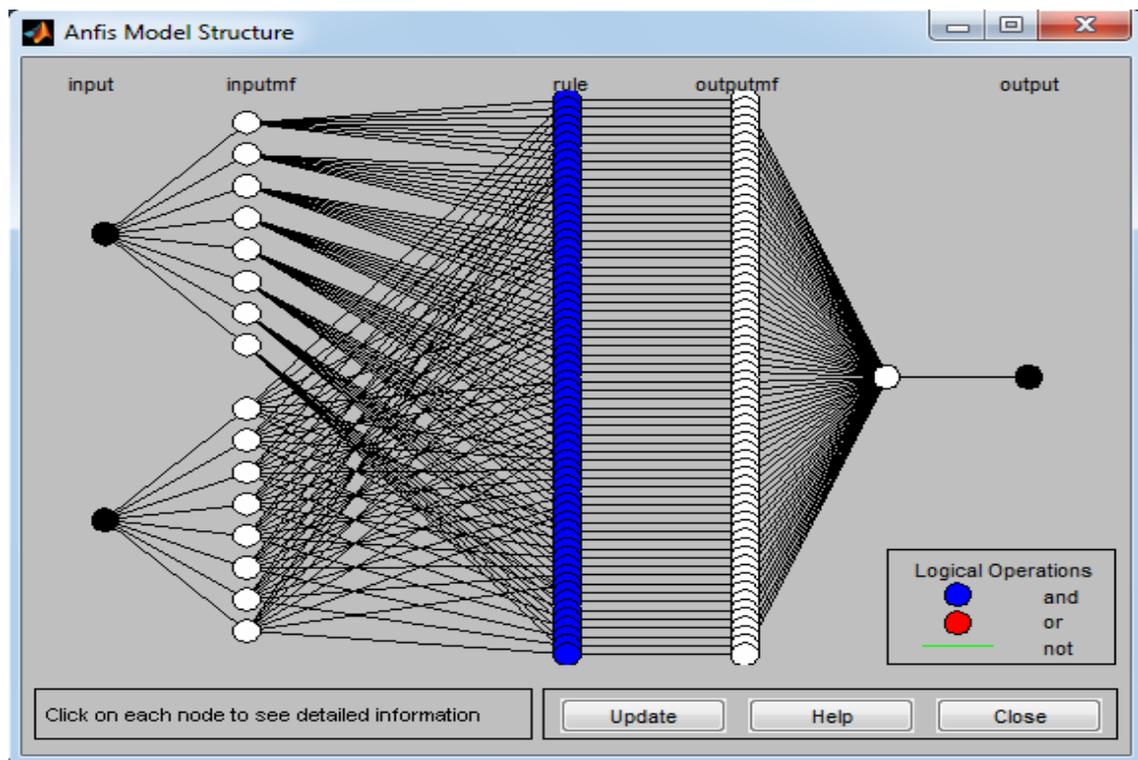


Figure III-33 : Structure du régulateur ANFIS développé

La figure III-34, présente la réalisation du schéma bloc de la commande Couple Calculé type *PD* basé sur la commande neuro-floue (ANFIS) utilisée en simulation.

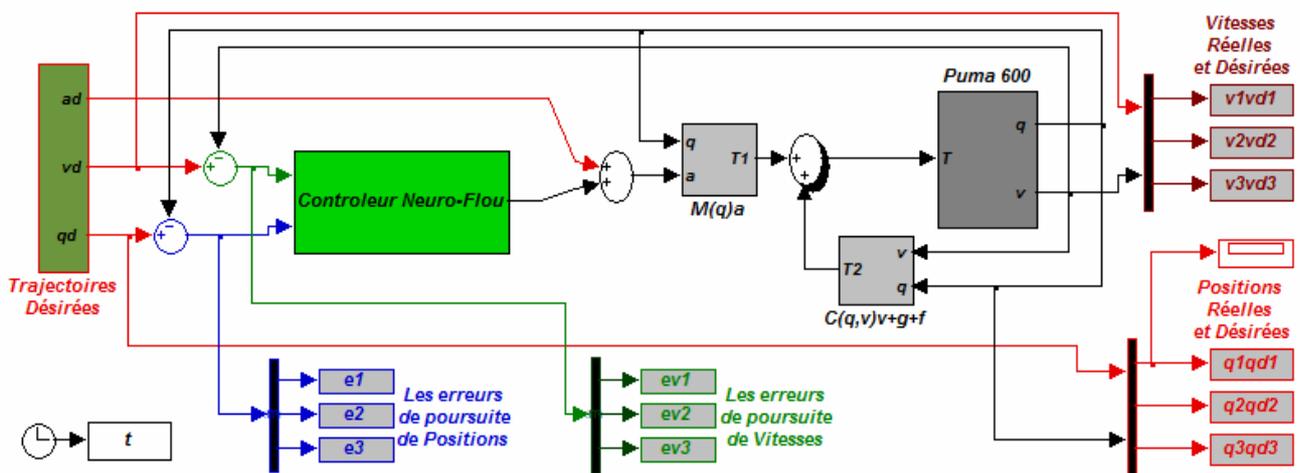
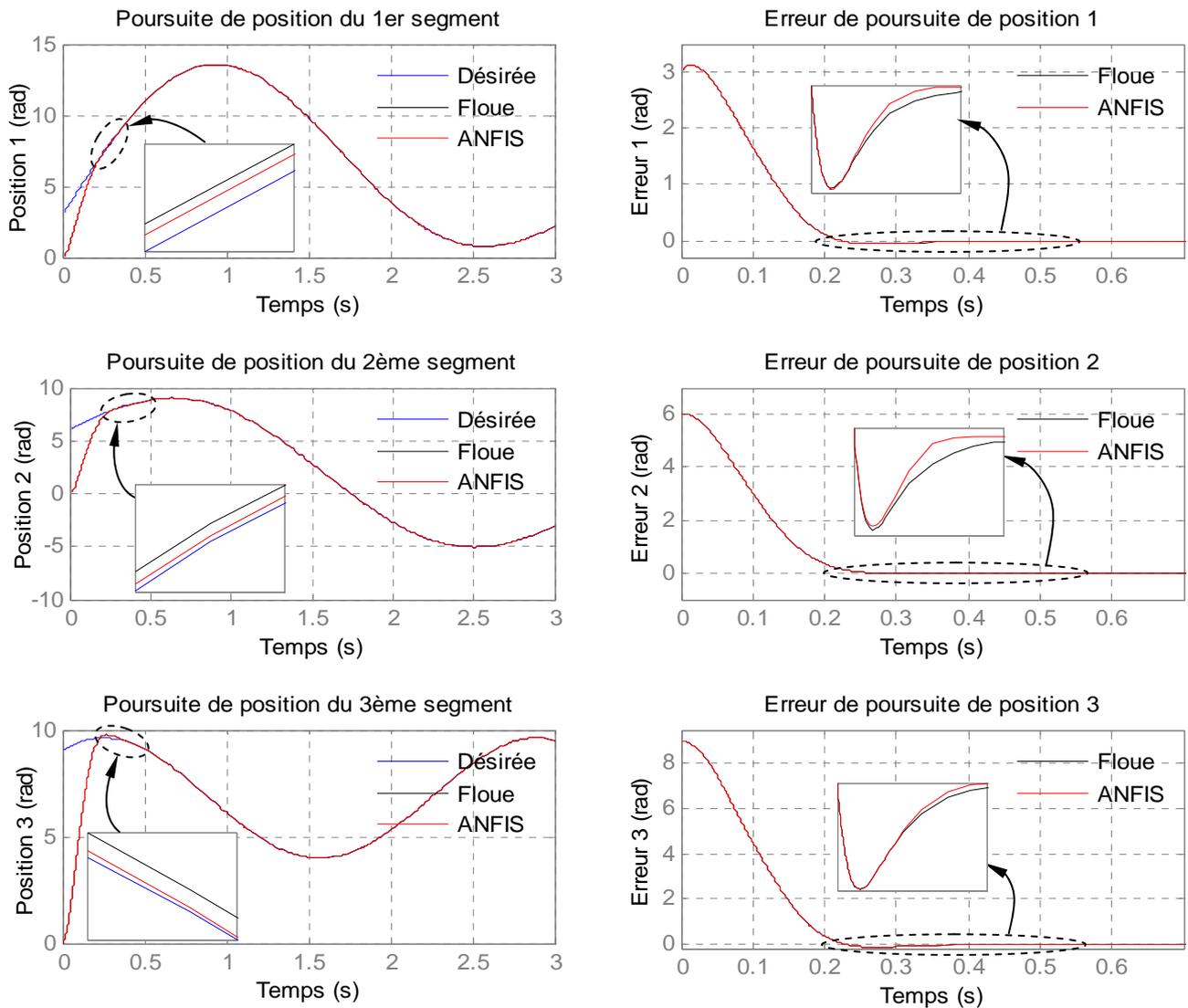


Figure III-34 : Schéma d'implémentation de la commande Couple Calculé de type *PD* basée sur le régulateur neuro-floue (ANFIS) sous Matlab/Simulink

Le comportement du robot en poursuite de la trajectoire désirée dans les deux cas de la commande du Couple calculé de type *PD*, par la logique floue et par le régulateur neuro-floue (ANFIS) est donné par la figure III-35.



**Figure III-35 :** Les poursuites de trajectoire de position et les erreurs de poursuite de position pour la loi de commande Couple Calculé de type *PD* basée sur l'ANFIS

Le comportement du robot en poursuite de la vitesse désirée dans les deux cas de la commande du Couple calculé, par la logique floue et par l'ANFIS, est résumé dans la figure III-36.

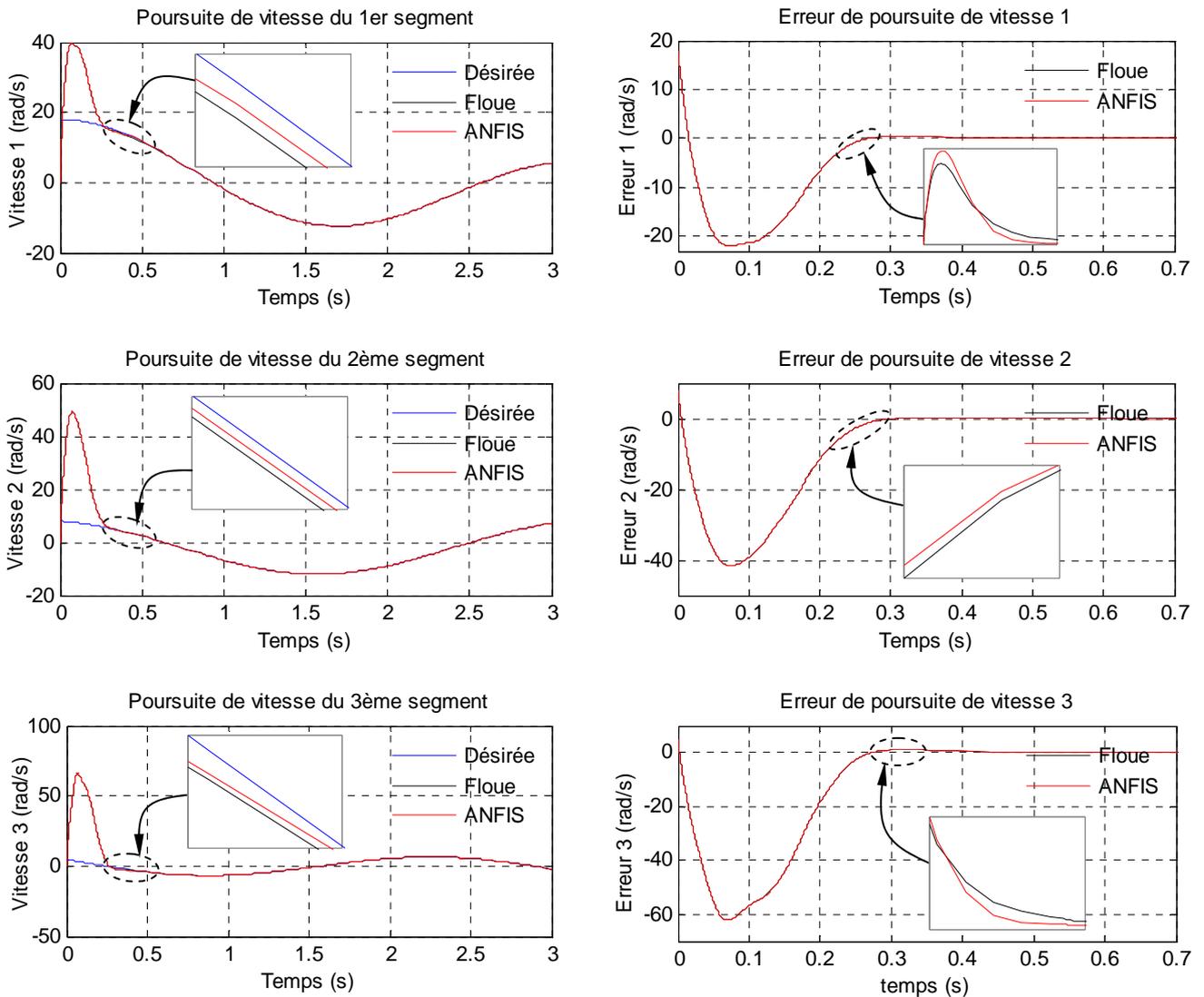
**Analyse et interprétation des résultats**

Suivant les résultats présentés sur les figures III-35 et III-36, on voit clairement que les performances de la commande neuro-floue sont meilleures que celles par logique floue. Ceci est interprété par la convergence plus rapide des erreurs de poursuites de position et de vitesse vers zéros (précision et stabilité) dans le cas du contrôleur ANFIS. Le robot atteint la trajectoire désirée dans un temps moins inférieur, dans le cas de la commande neuro-floue, que celui de la commande floue.

Ces résultats montrent que le contrôleur ANFIS est préférable que le contrôleur flou en robustesse (ajustement du taux de variation des gains *PD*) et en précision de suivi et de stabilité.

Cette étude par simulation indique clairement les bonnes performances du contrôleur neuro-flou adaptatif, car il est fondamentalement de nature adaptatif.

Il apparaît à partir des propriétés de réponse qu'il a une haute performance en présence d'incertitudes sur les paramètres du système et des perturbations inconnues. Il est utilisé pour contrôler un système avec un modèle inconnu.



**Figure III-36 :** Les poursuites de trajectoire de vitesse et les erreurs de poursuite de vitesse pour la loi de commande Couple Calculé de type *PD* basée sur l'ANFIS

### Conclusion

Ce chapitre présente un état de l'art sur les nouveaux outils de raisonnement intelligent utilisés dans les systèmes automatiques. Nous avons présenté les différentes notions et terminologies utilisées en logique floue. La théorie des ensembles flous, les différentes étapes de l'inférence floue et la méthode de réalisation d'un régulateur flou ont été développés. Les notions de base sur les réseaux de neurones, les différentes architectures et modèles les plus utilisés ont été exposés. Nous avons cité quelques applications des RNA dans le domaine de l'identification et de la commande neuronale. Le système d'inférence flou ANFIS a été exposé.

Des applications par simulation ont été effectuées avec les trois techniques de commandes intelligentes : par logique floue ; par réseaux de neurones et par neuro-flou (ANFIS). Toutes ces commandes montrent leurs stabilités vues la convergence des erreurs de poursuite de position et de vitesse dans les trois cas.

A travers les résultats de simulations obtenus, nous avons remarqué que la méthode de commande *Couple calculé* basée sur l'utilisation de la logique floue présente un caractère adaptatif dans la détermination des gains de commande contrairement à ceux classique qui sont fixes. Ces gains adaptatifs permettent le lissage des signaux de commande évitant ainsi, toute sollicitation des actionneurs. La stratégie de commande par réseaux de neurones est moins robuste que celle basée sur la logique floue. Son avantage, par rapport à celle du flou, réside dans sa rapidité d'exécution dans la boucle de régulation. Finalement la commande neuro-floue présente de meilleures performances, que celle par logique floue, en robustesse, en précision de suivi et de stabilité.

L'objectif du prochain chapitre sera le développement d'un système de téléopération pour le contrôle d'un robot distant Puma 600 sous un régulateur neuro-flou adaptatif (ANFIS) que nous avons développé (système intelligent distant).

### Références bibliographiques

- [1] Bühler H., *Réglage par logique floue*, Presse Polytechnique et Universitaires Romandes, Lausanne 1994.
- [2] Jean-Marie Flaus, *La régulation industrielle ; régulateur PID, prédictifs et flou*, Hermès, Paris, 1994.
- [3] Denis Meyer, *Logique floue & régulation PID ; Fuzzy, mode d'emploi*, PUBLITRONIC, Elektor, 1995.
- [4] Rambault, L., *Conception d'une commande floue pour une boucle de régulation*, Thèse de Doctorat de l'université de Poitiers, 1993.
- [5] Hala Bezzine, Nabil Derbel and Adel M. Alimi: *Fuzzy control of robot manipulators: some issues on design and rule base size reduction*, Engineering Applications of Artificial Intelligence, vol. 15, Issue 5, pp. 401-416, Sept 2002
- [6] Rakesh K. Arya: *Analytical Structures and Analysis of Simplest Fuzzy PD Controller with Asymmetrical/Symmetrical, Trapezoidal/Triangular/Singleton Output Membership Function*, international journal of computational cognition, vol. 5, no. 2, pp. 10-24, June 2007
- [7] Moon B.S.: *Equivalence between fuzzy logic controllers and PI controllers for single input systems* – Fuzzy sets and Systems 69 (1995) 105-113, Elsevier Science
- [8] Petr Pivonka: *Comparative analysis of fuzzy PI/PD/PID Controller Based on Classical PID Controller approach*, Proceedings of IEEE international conference on fuzzy systems, vol. 1, pp.541-546, May 2002.
- [9] Hevê Abdi, *les réseaux de neurones*, Presses Universitaires De Grenoble, 1994

- [10] C. S. G. Lee, J. S. Wang et J. Yuh, *Self-adaptive neuro-fuzzy systems for autonomous underwater vehicle control*, Advanced Robotics p. 589-608, 2001.
- [11] M. Norgaard, O. Ravn, N. Poulsen et L. Hansen, *Neural Networks for Modelling and Control of Dynamic Systems*, Springer, London, 1<sup>st</sup> edition, 2003.
- [12] M. Marie, M. Mokhtari, *Applications de MATLAB<sup>®</sup> 5 et SIMULINK<sup>®</sup> 2; Contrôle de procédés, Logique floue, Réseaux de neurones, Traitement du signal*, Springer-Verlag France, 1998.
- [13] S. Omatu, M. Khalid et R. Yusof, *Neuro-Control and its Applications Advances in Industrial Control*, Springer, London, 1996.
- [14] Patrick Naim, *Des Réseaux de Neurones*, Eyrolles, 1989.
- [15] Ajith Abraham, *Neuro Fuzzy Systems: State-of-the-art Modelling Techniques*, School of Computing & Information Technology, Monash University, Australia, 2001.
- [16] Brown M, Haris C., *Neuro-fuzzy adaptive modelling and control*, Englewood Cliffs: Prentice-Hall, 1994.
- [17] H. Bleuler, D. Diez, G. Lauber, U. Meyer and D. Zlatnik, *Nonlinear neural network control with application example*, In International Neural Networks Conference, vol.1, Paris, 1990, pp.201-204
- [18] B. Ouamri, Z. Ahmed-Foitih, *Computed Torque Control of a Puma 600 Robot by Using Fuzzy Logic*. International Review of Automation Control (IREACO), Vol. 4, N. 2, 2011, pp 248-252.
- [19] B. Ouamri, Z. Ahmed-Foitih, *Adaptive Neuro-fuzzy Inference System Based Control of Puma 600 Robot Manipulator*, International Journal of Electrical and Computer Engineering (IJECE), Vol. 2, N. 1, 2012, pp 90, 97.
- [20] A. Tahour, *Contribution à l'optimisation de la commande de l'ensemble convertisseur machine à réluctance variable par réseaux de neurones*, Thèse de Doctorat, Université Djilali Liabes de Sidi Belabbes, Juin 2007.

## *Chapitre IV*

*Etat de l'art sur la téléopération distante à  
travers le web, la réalité virtuelle et la réalité  
augmentée appliquées aux contrôles /  
Applications*

## Introduction

La téléopération, depuis sa naissance, a connue plusieurs problèmes sur les plans techniques et économiques : Interfaces homme-machine plus complexes, canaux de communication dédiés, limitation dans la distance, etc. Avec l'avancée technologique dans le domaine de la robotique, l'évolution rapide des calculateurs et de la technologie de transmission ont contribué à ouvrir un champ de recherche énorme dans le domaine de la téléopération et de la télérobotique. Cette percée technologique a conduit au développement des outils et des interfaces plus sûres et plus fiables de fonctionnement des systèmes de téléopération/télérobotique assurant un soulagement physique et décisionnel de l'opérateur humain et permet de fournir plus de rentabilité, de souplesse et de facilité d'accéder à des systèmes de contrôle qui ne sont pas limités à une région géographique donnée.

L'introduction des techniques de réalité virtuelle et de réalité augmentée permettent à l'opérateur de manipuler, par simulation, le robot, de programmer ses tâches à exécuter et de servir comme interface (entre l'opérateur et le robot) pour le piloter au cours d'une mission. Les recherches actuelles dans le domaine des systèmes autonomes motivent les spécialistes pour la réalisation des manipulations intelligentes dont l'objectif est la limitation de l'intervention humaine dans la boucle de téléopération.

Dans ce chapitre, nous commençons par un aperçu historique sur l'évolution de la téléopération depuis sa naissance jusqu'à nos jours. Nous abordons les différents problèmes rencontrés dans la téléopération. Les définitions de la téléopération, la télérobotique et leurs domaines d'application seront présentés. Nous détaillons les techniques de réalité virtuelle et de réalité augmentée comme outils indispensables pour la téléopération ainsi leurs domaines d'application. Nous exposons les différentes techniques et architectures de téléopération via internet.

Comme applications, nous proposons deux techniques de téléopération d'un bras manipulateur Puma 600 : La première consiste à l'utilisation d'un Joystick comme initiation à la téléopération. La seconde, consiste à la contribution au développement d'une interface de télérobotique d'un bras manipulateur Puma 600 via Internet, combinant un planificateur de trajectoires, les outils de réalités virtuelle et augmentée et des retours vidéo et sensoriels. Le système distant (robot) est mis en boucle fermée avec un régulateur intelligent.

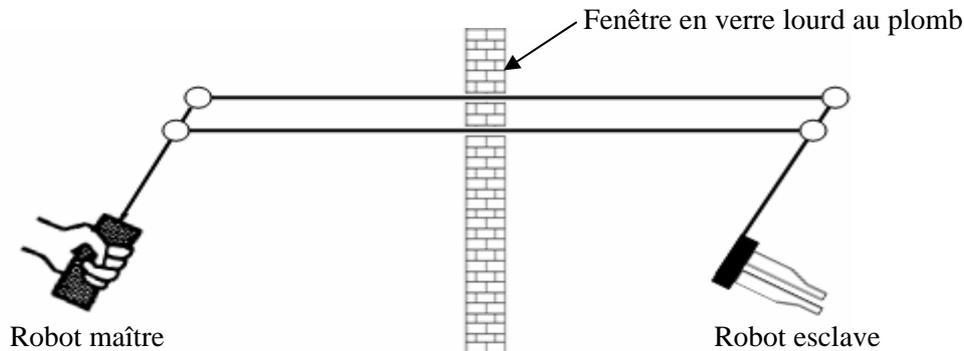
### IV-1. Historique

La téléopération, prend ses origines depuis 1947 pour laquelle Raymond Goertz et son équipe de l'Argonne National Laboratory, dont l'objectif de manipuler des produit radioactifs, réalisent le premier télémanipulateur mécanique, ancêtre des manipulateurs maîtres-esclaves (figure IV-1). L'opérateur, dans ce cas, manipule un outil localisé dans l'environnement esclave à l'aide d'une poignée située du côté maître grâce à un système entièrement mécanique.

Ce type de télémanipulateurs, présente l'inconvénient de provoquer une fatigue physique importante sur l'opérateur et impose une distance très limitée entre lui et la zone de travail.

Dès 1954 et afin de pallier ces inconvénients, le même Raymond Goertz crée le premier système de télémanipulation électrique doté d'un asservissement bilatéral position/position [1]. La

transmission mécanique entre le maître et l'esclave est remplacée par une liaison électrique (utilisation de moteurs électriques) radio ou par câble, permettant l'amélioration de l'indépendance entre les deux univers (maître et esclave).



**Figure IV-1:** Télémanipulateur mécanique

En effet, les efforts sont transmis artificiellement à l'opérateur, grâce à un système de démultiplication électrique, dans un rapport multiple comparativement aux rotations mesurées au niveau des articulations réelles. De plus, le rayon d'action de l'opérateur sur l'esclave est plus important, et peut s'étendre tant que le temps de transmission aller-retour des informations ne nuit pas à la stabilité du système.

L'inconvénient de ces systèmes, lorsque le site esclave est très loin, est de ne plus offrir de retour visuel direct sur la scène de la tâche. Cet inconvénient a été remplacé par des caméras vidéo situées dans la zone esclave qui recopient les différentes étapes de la tâche pour être transmises et exploitées au niveau du site maître à l'aide des écrans vidéo.

Dans ces systèmes de téléopération classiques, l'opérateur fait tout. D'une part, il connaît la tâche à réaliser et il élabore en permanence une stratégie d'action sur les organes maîtres en fonction de l'état d'avancement de la tâche. D'autre part, il apporte toute l'énergie mécanique nécessaire à la réalisation du travail.

Depuis les années soixante-dix, l'évolution rapide des calculateurs conduit les chercheurs, dans le domaine de la téléopération, à mettre en place de nombreuses fonctionnalités qui facilitent le travail de l'opérateur et qui permettent la mise en oeuvre de systèmes mécaniques plus complexes et mieux adaptés à l'opérateur et à l'environnement (développement des outils et des interfaces plus sûres de fonctionnement des systèmes de téléopération assurant un soulagement physique et décisionnel de l'opérateur humain). Cette évolution a notamment rendu possible le développement de la téléopération assistée par ordinateur où l'homme n'est désormais plus seul à intervenir [2].

En 1961, un sous-marin expérimental destiné à la submersion profonde, le bathyscaphe Trieste, a été équipé d'un télémanipulateur basé sur un contrôle unilatéral par un clavier [3]. En 1966, un véhicule sous-marin de recherche contrôlé par un câble, a récupéré une arme nucléaire qu'était effondrée dans la mer au large de la côte espagnole à Palomares. En 1967, l'arpenteur III atterri sur la surface de la lune. Il était équipé de bras manipulateurs, pour expérimenter des échantillons du sol

lunaire [3]. Ce fut le premier exemple de téléopération dans l'espace. Ce véhicule est télécommandé directement à partir de la terre avec quelques secondes de retard dans le transfert d'informations. Ce retard de transmission bidirectionnelle des informations (envoi des commandes et retour d'informations) présente le principal inconvénient de la téléopération dans l'espace.

La sonde Viking, qui a atterri sur Mars en 1976, a été programmée pour effectuer des opérations strictement automatisées. Ce bras manipulateur, qui est plus efficace que l'arpenteur, est utilisé pour prélever des échantillons et faire des analyses sur le sol martien [3].

Une autre amélioration de la commande distante d'un véhicule sous-marin (Jason RUV (Remotely Underwater Vehicle)) a été développée par Yoerger en 1985 [4]. Le sous-marin Jason a été téléopéré à partir de la plate-forme d'imagerie ARGO tracté de la surface. Ce système intègre un système de vision pour aider les chercheurs de la surface au cours de la tâche d'exploration sous-marine. Aujourd'hui, le système d'exploration sous-marine est communément connu comme le système ARGO / JASON [5].

Le projet Mercury, est la première mise en œuvre réussie d'une téléopération via Internet qui a été développé par Goldberg en 1994 à l'Université de Californie du Sud. Ce projet, comprend le fonctionnement d'un manipulateur robotique avec une interface simple, programme CGI (Common Gateway Interface) et un retour vidéo. Les utilisateurs ont utilisé, pour la première fois, le World Wide Web pour commander un robot pour effectuer des tâches dans le but de découvrir des artefacts enfouis dans un terrarium rempli de sable [3]. Cette percée technologique a contribué à ouvrir un champ de recherche énorme dans le domaine de la téléopération via internet.

La téléopération, malgré ses grandes avancées technologiques, est toujours limitée aux systèmes qui incluent constamment l'opérateur dans la boucle de contrôle.

Afin de réaliser des manipulations intelligentes avec le minimum possible d'interventions humaines, des recherches plus approfondies, en robotique autonome, motivent les spécialistes actuellement.

## **IV-2. Téléopération**

Elle a pris ses origines dans le besoin de prolonger le geste de l'homme au delà de la main, et se poursuit par l'ambition de se trouver là où on ne se trouve pas (Téléprésence).

La téléopération signifie le contrôle d'un système à un site distant (esclave), généralement inaccessible par l'opérateur. L'opérateur (maître) et le système distant sont physiquement séparés et doivent interagir pour accomplir une tâche donnée. Le concept des systèmes de téléopération a été développé autour des dernières décennies. Un système de téléopération est généralement constitué des éléments de base suivants: système distant avec des capteurs, actionneurs, contrôleurs et systèmes d'alimentation, une interface locale qui permet à l'opérateur de générer des commandes et de recevoir les retours sensoriels de l'environnement à distance et un support de communication à travers lequel l'information est échangée entre les environnements locaux et distants.

La figure IV-2 représente le principe d'un système de téléopération.



Figure IV-2 : Schéma d'une structure générale d'un système de téléopération

Traditionnellement, les systèmes de téléopération utilisent un canal de communication dédié qui entraîne des coûts élevés, l'inflexibilité et des temps de transmission important. Toutefois, la croissance de l'Internet ouvre une nouvelle étape pour les systèmes téléopérés.

#### IV-2-1. Problèmes rencontrés dans la téléopération

Un système de téléopération dispose de divers composants et nécessite la connaissance de nombreux domaines différents. Les principaux problèmes qui affectent les performances de la téléopération sont les suivants:

##### IV-2-1-1. Les retards de transmission

Le retard de transmission (dans le cas du réseau Internet, nous parlons de la latence) est défini comme le temps que prend le système distant (esclave) pour répondre aux commandes de l'utilisateur (maître), en d'autres termes, le temps entre la communication des instructions et la visualisation des résultats.

Le retard dans les systèmes de téléopération est essentiellement dû à la transmission d'information entre l'interface de l'opérateur et le système distant. Comme nous l'avons vu, dans la téléopération classique et dans le contrôle bilatéral, l'opérateur doit attendre le retour d'information (images vidéo, forces de contacts, etc.) du système distant. Les retards dans la communication augmentent le temps d'achèvement des tâches car l'utilisateur essaie de s'adapter à eux en utilisant des stratégies comme «déplacer et attendre» : réaliser une action très simple et attendre le retour pour continuer [24]. En outre, lorsque les retards de communications sont importants, la perception du site de travail est dégradée et le système devient difficile à contrôler.

Plusieurs solutions ont été proposées pour améliorer la performance des systèmes de téléopération en présence de retards de transmission. Deux approches ont été utilisées afin d'analyser ce problème [22]. La première est basée sur l'automatique classique et la seconde est basée sur la théorie des lignes et de la passivité. La première approche, est de nature compensatoire, l'idée était d'introduire des correcteurs dans la boucle d'asservissement afin de compenser les systèmes à retards [25]. La seconde approche, considère le retard comme subi, c'est à dire comme partie intégrante du système [26, 27]. Malheureusement, ces deux approches exigent que le délai de transmission soit connu, alors que la majorité des protocoles publics actuels de transmission de données informatique ne garantissent pas la constance des délais de transmission. En effet indépendamment de la taille des données transmises, le délai peut varier d'une manière très aléatoire pour certains protocoles de transmission (par exemple Internet). Les travaux de Kosuge et son équipe [28] ont

montré que ce problème entraîne une instabilité même si les variations de délai sont relativement petites.

#### 1V-2-1-2. La prise en compte du facteur humain et sécurité

Dans les systèmes de téléopération originels, l'opérateur doit impérativement subir un apprentissage qui peut être long avant d'obtenir une bonne adaptation et une bonne maîtrise du système de téléopération. En plus l'adaptation obtenue peut ne pas être adéquate pour une autre architecture. La sophistication et le nombre important d'applications téléopérées engendre une augmentation de la charge de travail de l'opérateur, sa fatigue et ce qui entraîne une diminution de ses performances et augmente ainsi les risques d'erreurs de téléopération. Pour améliorer l'interface et la qualité de l'information présentée à l'opérateur, les systèmes de téléopération ont cherché à interagir les différentes sources d'information en incorporant des techniques de réalité virtuelle et de réalité augmentée (ergonomie du poste de travail) [22] [23].

#### 1V-2-1-3. Degré d'autonomie du système

Dans le cadre de la téléopération, plusieurs architectures ont été proposées pour la distribution des responsabilités entre l'opérateur et le système distant (coopération homme-machine). Cette distribution est étroitement liée au degré d'autonomie du système téléopéré et du niveau d'intervention de l'opérateur.

Nous parlons de *contrôle direct* si l'opérateur prend en charge entièrement la réalisation de la tâche (contrôle purement manuel). Dans ce cas, le système distant est classé comme non-autonome. Nous parlons de *contrôle automatique* quand le système distant exécute la tâche indiquée par l'opérateur, il est aussi possible d'avoir un *contrôle partagé* : certains mouvements ou degrés de liberté peuvent être gérés par l'ordinateur de façon à permettre à l'opérateur de se concentrer sur l'essentiel de la tâche.

Le *contrôle supervisé*, (une stratégie intermédiaire entre le contrôle manuel et automatique), donne de l'autonomie au système distant pour permettre la réalisation de certaines tâches sans l'intervention humaine [24]. Dans ce cas, le système distant est classé comme semi-autonome (possède un certain degré d'intelligence artificielle). Le contrôle supervisé est souvent employé dans la télémanipulation [33] et dans une moindre mesure dans la téléopération de véhicules [29, 30, 31].

Le *contrôle collaboratif*, le système distant possède des algorithmes à base d'intelligence artificielle pour la réalisation des différentes tâches telles que la planification de trajectoires, évitement de collisions, etc. L'opérateur dans ce cas n'est plus le centre du système téléopéré [32]. Il est utilisé comme une source limitée d'information à laquelle le robot peut faire des requêtes. Le système distant dans ce cas est classé comme autonome.

### IV-3. Télérobotique

Il y a beaucoup d'endroits qui ne conviennent pas pour la présence humaine, tels que l'océan profond, l'espace, les sites nucléaires, etc. Toutefois, nous devons encore effectuer de nombreuses tâches dans ces lieux. La réalisation de ces tâches s'effectuent, généralement, à distance par l'envoi

d'un robot. Ce dernier travaille sous la direction d'un opérateur humain, on parle donc de la télérobotique. L'opérateur humain supervise le robot à travers un outil de commande (ordinateur) par l'intermédiaire d'un canal ou un réseau de communication. Le robot exécute des tâches basées sur les informations reçues de l'opérateur humain ainsi que ses propres détections autonomes et de renseignement.

### IV-3-1. Quelques domaines d'application de la téléopération

Les domaines d'application de la téléopération sont très variés, en dimensions et en lieux géographiques, et touchent la plupart des domaines de recherche (recherche médicale, spatiale, militaire, nucléaire, etc.). Parmi lesquels nous citons :

#### a. Le domaine nucléaire

Le domaine nucléaire a motivé le développement d'un grand nombre de systèmes de téléopération suite à la nécessité de disposer de moyens de manipulation et de transfert à distance de produits radioactifs, la maintenance des installations et le démantèlement d'installations suite à un incident. Un des organismes de recherche qui s'est mis à développer un savoir-faire dans le domaine de la téléopération est le Commissariat à l'Energie Atomique (C.E.A.) qui a développé le robot Sherpa (figure IV-3 (a)), un hexapode pour ambiance hostile, capable de se faufiler à l'intérieur d'une installation nucléaire par des chemins initialement prévus pour l'homme, de soulever et d'emporter des charges lors d'interventions, d'amener des bras manipulateurs en position de travail pour réparer ou remplacer du matériel en panne.

Les opérations de décontamination sont coûteuses, prennent du temps et peuvent nécessiter l'exposition des personnes aux risques de radiations. Dans ce sens, les chercheurs travaillent sur la mise au point de systèmes plus approfondis et plus pratiques sur les interventions et la réalisation de différentes missions à distance.

La figure IV-3 (b), illustre l'utilisation d'un système de télérobotique du groupe Intra (Intervention robotique sur accidents) pour le démantèlement d'installations et l'intervention dans le milieu nucléaire en cas d'accident.



a) Robot Sherpa



b) Robot de démantèlement et d'intervention

Figure IV-3: Exemple de robots téléopérés en milieu nucléaire

### b. Le domaine sous-marin

Le développement de l'exploitation pétrolière offshore et des recherches océanographiques au cours des dernières décennies a progressivement nécessité la mise au point d'engins ou de systèmes d'intervention sous-marins à distance. Dans de telles applications, les systèmes de téléopération gèrent les inconvénients qui résultent du milieu, à savoir la communication, les pressions élevées et la faible visibilité [6] [7].

La figure IV-4 (a) présente un véhicule sous-marin, Victor 6000, développé par L'IFREMER (Institut Français de Recherche pour l'Exploration de la Mer) pour l'exploration des fonds sous-marins. Il s'agit d'un engin à câble piloté à partir d'un navire support. Il a été conçu pour faire de l'investigation optique et effectuer des missions locales incluant l'imagerie, la mise en oeuvre d'instrumentation ainsi que des prélèvements d'eau, de sédiments ou de roches.

La figure IV-4 (b) illustre un robot sous-marin, appelés ROV (Remote Operated Vehicle) participe à l'inspection, la maintenance et l'ensouillage des câbles. Les ROV *Hector*, fabriqués par SIMEC, filiale de FT Marine, peuvent opérer jusqu'à des profondeurs de 2000 mètres et disposent de bras articulés pour saisir, couper, tenir.

Le domaine sous-marin continue à être le principal domaine d'activité en nombre de systèmes de télérobotique.



a) Robot Victor 6000



b) Robot ROV

**Figure IV-4:** Exemples de robots téléopérés sous-marin

### c. Le domaine spatial

Les programmes spatiaux constituent le domaine qui a fortement influencé le développement des systèmes de téléopération. Dans ce secteur, les applications visées sont essentiellement la maintenance des satellites et des plates-formes spatiales [8], ou encore l'exploration des planètes [9] avec notamment le robot Rocky (Figure IV-5 (a)).

Les problèmes majeurs rencontrés dans ces applications sont sur :

- le plan économique et humain (nécessité de quantités énormes d'efforts et d'argent pour couronner une mission de succès),
- le plan scientifique (délais de transmission non négligeables dans le cas où le contrôle se fait depuis la terre).

La résolution du deuxième problème conduit à l'utilisation de robots ayant une grande autonomie (basée sur une architecture complexe et intégrant de nombreux capteurs) sans pour autant exclure la possibilité de les téléopérer.

La NASA, en janvier 2004, a envoyé deux grands robots d'exploration (figure IV-5 (b)), vers la planète rouge, équipés d'éléments scientifiques perfectionnés se sont posés sur le sol martien et explore la surface de la planète à la recherche de preuves supplémentaires sur la présence d'eau liquide sur Mars dans un lointain passé.



a) Robot Rocky



b) Les deux robots jumeaux Spirit et Opportunity pour la mission «Mars ExplorationRovers»

**Figure IV-5:** Exemples de Robot téléopéré dans l'espace

#### d. Le domaine médical

La qualité et la réussite des opérations chirurgicales dépendent de plusieurs facteurs, dont la dextérité du chirurgien : accessibilité, précision du geste, pertinence, rapidité, etc. Les nouvelles technologies de robotique, de réalité virtuelle et de réalité augmentée peuvent aider les chirurgiens dans leurs gestes et améliorer ainsi certaines opérations. Les dispositifs de robotique chirurgicale ont pour objectif d'apporter une aide au chirurgien, lors de l'opération, forts de la capacité de la robotique à commander, contrôler et automatiser des mouvements mécaniques précis en environnement complexe [34].

Tout ce développement, dans le domaine de la robotique chirurgicale, offre plus d'applications pour la téléopération dans le domaine médicale. Ce qui permet de faciliter le travail en chirurgie laparoscopique grâce à une interface redonnant de la visibilité et des degrés de liberté au chirurgien et qui permet aussi d'échanger des informations à distance (images, vidéo, son) au cours d'une opération (la téléassistance chirurgicale). Ainsi, le télédiagnostic permet l'examen d'un patient à distance sans aucun contact physique, ni déplacement du malade ou du médecin. Des techniques chirurgicales plus sophistiquées peuvent être appliquées en utilisant un toucher, une vision ou une manipulation améliorée.

Les chirurgiens et les patients du Service d'urologie peuvent désormais compter sur les services d'un outil perfectionné : le robot chirurgical Da Vinci (figure IV-6), utilisé pour pratiquer des interventions délicates à effraction minimale [10].

Le robot chirurgical Da Vinci, utilisé pour pratiquer des interventions délicates à effraction minimale, libère le chirurgien de la nécessité de manipuler directement des instruments chirurgicaux.

En effet, les quatre bras robotiques de l'appareil Da Vinci, chacun portant son propre instrument, pénètrent dans le corps du patient par de petites incisions.

Le chirurgien, assis à plusieurs mètres de distance à une console spéciale équipée d'un écran 3D, utilise des leviers et des pédales pour manipuler les bras du robot avec un degré de précision difficile à atteindre au cours d'une chirurgie laparoscopique traditionnelle.

La chirurgie assistée par robot est moins invasive et mieux tolérée, ce qui fait que les patients se rétablissent à une vitesse remarquable et ressentent moins d'effets indésirables.



**Figure IV-6:** Le système Da Vinci : la console maître et les manipulateurs esclaves

Le concept de téléopération mise en oeuvre sur Da Vinci a été développé à l'extrême avec Zeus (Figure IV-7). En effet, en septembre 2001, le professeur Marescaux (IRCAD : Institut de Recherche contre les Cancers de l'Appareil Digestif, Strasbourg, France) a téléopéré ce système depuis New York en réalisant avec succès une cholécystectomie (ablation de la vésicule biliaire) sur une patiente hospitalisée à Strasbourg. Cette première opération transatlantique baptisée «Opération Lindbergh», fut réalisable grâce à l'utilisation d'un réseau de communication à haut débit de France Télécom (10 mégabits/seconde). Le délai entre le geste du chirurgien sur la console maître et le retour visuel des mouvements des manipulateurs a pu être réduit à 150 ms [11].

Nous noterons que l'évolution des systèmes de téléopération, offre aussi plus de possibilités pour le développement des dispositifs d'aide aux handicapés [12] [13].



**a)** La console de commande avec les bras maîtres



**b)** Les manipulateurs esclaves

**Figure IV-7.** Le système Zeus

### e. Le domaine militaire

Les applications concernant la téléopération de plate-formes robotisées (pilotage des véhicules à distance) est très utilisée dans le domaine militaire, et surtout dans les secteurs comme le déminage et la reconnaissance de zone, d'assister l'homme dans certaines tâches pénibles ou en milieu hostile. En effet, on peut ainsi commander des robots (véhicules), dans des situations critiques permettant au combattant de ne plus être en contact avec le danger tout en effectuant son travail de désamorçage des charges explosives. Actuellement, les chercheurs font des études avancées sur la mise en oeuvre de Drones (avions sans pilote) et de robots téléopérés à partir de plate-formes de combat pour des missions d'acquisition.

La figure IV-8, illustre un exemple d'un robot commandé à distance pour des missions de désamorçage de mines, bombes, engins piégés, utilisé aussi pour surveiller certaines zones et partir en reconnaissance et un drone pour l'acquisition des images vidéo sur les zones à opérer des ennemis.



a) Plateforme mobile tout terrain muni d'un bras articulé et de caméras.



b) Drône militaire de haute technologie

**Figure IV-8:** Exemples d'engins militaires téléopérés

### f. Le domaine des mines

Dans les mines souterraines, les opérateurs d'équipements travaillent souvent dans des conditions qui peuvent se révéler dangereuses pour leur santé et leur sécurité. En plus d'effectuer une tâche répétitive, qui comporte des risques d'accidents graves, ils font face à un environnement souvent hostile: bruit, poussières, vibrations, fumée d'échappement, chocs répétés, etc., lorsqu'ils sont dans leurs machines. Les chercheurs se sont concentrés sur l'automatisation des chargeurs transporteurs en faisant appel à deux technologies adaptées au cycle d'opération de ces équipements: la téléopération du véhicule dans la zone de chargement et son guidage automatique lors du transport et au point de déchargement.

La figure IV-9 (a), illustre un exemple d'un robot mobile semi-autonome (camion lourd) construit par les ingénieurs de Caterpillar. Possède un poids de 700 tonnes, avec la possibilité de charger 240 tonnes de mines. GPS conserve le suivi de ces véhicules, tandis que les scanners laser surveillent la route et détectent les obstacles.



a) Robot (camion lourd) construit par Caterpillar



b) Mine australienne téléopérée

**Figure IV-9:** Exemples de Robots téléopérés dans le domaine de mines

La figure IV-9(b) illustre une mine de fer australienne, contenant des robots chargeurs et des camions transporteurs, contrôlée à distance réalisée par le leader de Rio Tinto dans le domaine de l'innovation minière. Cette mine utilise actuellement cinq camions équipés de radars autonomes, des lasers, des antennes de communication et de haute précision GPS.

En dehors de ces domaines, particulièrement importants car correspondant à un environnement peu accessible à l'homme ou complexe, d'autres applications apparaissent dans d'autres domaines :

- La télémaintenance : Hydro-Québec (l'E.D.F. du Québec) étudie cet outil afin de surveiller et de réparer ses installations électriques réparties sur son immense territoire.
- La gestion d'un parc automobile : Dans le cas de parking, l'opérateur, peut gérer et déplacer la voiture depuis son poste de surveillance.
- La robotique de service (bulldozer), ou encore la protection civile (lutte contre l'incendie).

### Remarque

L'évolution des technologies de communication et des systèmes informatiques dans le domaine d'interaction et d'interface homme/machine conduit fortement à l'évolution des systèmes de téléopération. Nous nous intéressons tout particulièrement aux technologies de la Réalité Virtuelle (RV) et de la Réalité Augmentée (RA) qui ont été rapidement adoptées au profit de la téléopération.

### IV-4. Les outils de réalité virtuelle et de réalité augmentée pour la téléopération

La Réalité Virtuelle (RV) et la Réalité Augmentée (RA) constituent des éléments importants dans le développement des systèmes de téléopération du point de vue amélioration des boucles de communication et la réalisation des interfaces de contrôle homme/machine (l'ergonomie du poste de travail) plus fiable.

**IV-4-1. Définition de la RV**

La réalité virtuelle est une expression qui désigne d'elle-même une contradiction : la réalité représente ce qui existe, alors que le virtuel désigne ce qui n'existe pas.

La réalité virtuelle est une discipline qui se situe à la croisée de plusieurs domaines tels que l'infographie, la conception assistée par ordinateur, la simulation, la téléopération, l'audiovisuel, le travail collaboratif, etc. L'interaction 3D est la composante motrice de la réalité virtuelle. Elle permet à l'utilisateur d'être un acteur capable d'interagir avec les entités de l'environnement virtuel. Le terme "réalité virtuelle" a été employé pour la première fois en 1986 par Jaron Lanier [19] [21].

Selon les domaines d'applications de la réalité virtuelle, plusieurs définitions existantes dans la littérature. Cependant, il n'existe pas de consensus général sur une définition unique.

Fuchs définit la RV à travers les techniques sur lesquelles elle repose comme suit :

Les techniques de la réalité virtuelle sont fondées sur l'interaction en temps réel avec un monde artificiel, à l'aide d'interfaces comportementales permettant l'immersion « pseudo-naturelle » de(s) l'utilisateur(s) dans cet environnement. Ce monde artificiel est imaginaire ou une simulation de certains aspects du monde réel [18].

D'autres définitions de la RV d'un point de vue fonctionnel et technique sont détaillées dans [14] :

**a. Définition fonctionnelle**

La réalité virtuelle va permettre de s'extraire de la réalité physique pour changer virtuellement de temps, de lieu et (ou) de type d'interaction : interaction avec un environnement simulant la réalité ou interaction avec un monde imaginaire ou symbolique [14];

**b. Définition technique**

La réalité virtuelle est un domaine scientifique et technique exploitant l'informatique et des interfaces comportementales en vue de simuler dans un monde virtuel le comportement d'entités 3D, qui sont en interaction en temps réel entre elles et avec un ou des utilisateurs en immersion pseudo-naturelle par l'intermédiaire de canaux sensori-moteurs [14].

Quelle que soit la définition de cette réalité dite virtuelle, le principe d'un système de RV reste toujours le même [15]:

Un système de réalité virtuelle est une interface qui implique la simulation en temps réel et des interactions via de multiples canaux sensoriels. Ces canaux sensoriels sont ceux de l'homme (vision, audition, toucher, odorat et le goût).

**IV-4-2. L'Environnement virtuel**

Le terme «environnement virtuel» a été introduit par les chercheurs du MIT (Massachusetts Institute of Technology) au début des années 90 comme synonyme de RV [16]. Il est considéré comme le lieu suggéré par la RV pour accueillir un ou plusieurs utilisateurs et leur permettre d'accomplir certaines tâches avec l'impression d'être dans un cadre spécifique. L'environnement

virtuel est représenté par un modèle 3D de données réelles ou imaginaires qu'on peut visualiser et avec lesquelles on peut interagir en temps réel [17].

L'environnement virtuel est constitué d'un ensemble de techniques permettant de reproduire le plus fidèlement, par calcul, le comportement d'entités 3D.

#### **IV-4-3. Définition de la réalité augmentée (RA)**

La réalité augmentée regroupe l'ensemble des techniques permettant d'intégrer des éléments virtuels (images de synthèse, objets virtuels, graphiques, etc.) produit par l'ordinateur dans un monde réel. Elle a été introduite dans les travaux de Sutherland en 1965 et 1968 [35], qui a réalisé le premier système de RA, basé sur un casque de RV transparent «See Through System». En effet, Sutherland introduira le concept de contact entre l'homme et la machine, en plaçant un utilisateur à l'intérieur d'un environnement en trois dimensions généré par ordinateur.

D'une manière générale, la RA consiste à augmenter la scène réelle avec des informations virtuelles supplémentaires. Cette augmentation peut prendre différentes formes selon les domaines d'application. Deux façons possibles pour l'utilisation de la RA :

- Augmenter directement ce qui est vu par l'utilisateur en superposant des informations virtuelles sur la vue réelle de l'utilisateur (cas de la RA en vision directe). Ce type d'augmentation peut se faire soit via des dispositifs d'affichage de type See-Through (Semi-Transparents), soit par projection sur des surfaces (cas des surfaces augmentées).
- Augmenter ce qui est vu par un ou plusieurs capteurs en superposant des informations virtuelles sur des retours d'informations obtenus par ces capteurs (cas de la RA en vision indirecte). Par exemple dans le domaine de la télérobotique, cette opération consiste à superposer un modèle virtuel sur une image réelle d'un robot distant grâce à un mécanisme appelé la calibration [22].

#### **IV-4-4. Domaines d'application de la RV et de la RA**

L'évolution de la technologie d'affichage et des systèmes informatiques, ainsi que le traitement temps réel d'images vidéo a rendu possible l'affichage d'images virtuelles correctement appariées aux images vidéo. Les applications potentielles de la RV et de la RA sont nombreuses et touchent différents domaines d'applications, parmi les quels nous citons :

##### **a. L'Architecture**

Contrairement aux plans construits sur papier, la RV et la RA donnent la possibilité de construction des maisons et des bâtiments d'une façon plus claire, donner la possibilité de visiter l'intérieur et la vérification de différents problèmes et obstacles existants lors des déplacements dans les locaux surtout pour les personnes handicapés.

##### **b. L'Education**

La possibilité de réalisation des animations sur des mouvements des objets (mouvement circulaire, rectiligne, ...) offrent plus d'avantages pour la compréhension et la simplification des

différentes notions de la physique, qu'il soit pour l'enseignant ou l'élève. La possibilité de créer des scènes parfaites sur la vie des êtres préhistoriques donne plus de compréhension sur leur histoire, ce qui aide les professeurs dans ce domaine de présenter leurs travaux et cours d'une façon plus claire...etc.

### **c. Le Divertissement**

C'est sans doute dans ce domaine que la RV et la RA sont plus exploités à leurs états purs. Des jeux très réels dans un monde imaginaire où tout est permis. La majorité des jeux virtuels qui existent sont basés sur le principe de la simulation.

### **d. Militaire**

Grâce à ces techniques de RV et de RA, l'armée peut entraîner ses régiments pour des missions spécifiques ou bien pour les entraîner de façon générale. Ceci ce fait par l'utilisation des simulateurs virtuels avec des données qui provenaient directement des stations de contrôle et de surveillance, des satellites ou d'espions.

### **e. Robotique/Télérobotique**

La télérobotique trouve des applications partout où l'homme a des difficultés à travailler directement (milieu hostile, lointain, trop grand ou trop petit, spatial, ...) et où les tâches sont suffisamment complexes ou imprévisibles pour rendre difficile une automatisation complète.

Grâce à la RV les opérateurs peuvent s'entraîner à manipuler le robot, améliorer leurs réflexes en causant de fausses situations d'urgence ou de faux problèmes techniques. Ils peuvent aussi enregistrer une série de mouvements ou de tâches, que le robot devra faire, avant de les faire par le robot dans une situation réelle. Comme elle peut servir d'interface entre l'opérateur et le robot au cours d'une mission, par laquelle il peut le manipuler et l'assister.

La RA est utilisée pour améliorer la perception visuelle ou encore anticiper une situation réelle et prévenir des cas dangereux. En effet, la superposition d'un modèle virtuel sur une image réelle est très utilisée en Téléopération. Si l'opérateur désire effectuer un déplacement sur le robot réel, il peut le réaliser sur le robot virtuel qui n'est rien d'autre qu'un rehaussement graphique du vrai robot. L'opérateur peut alors décider de l'exécution de la tâche après avoir vu le résultat de la simulation, ainsi les problèmes d'une opération de télérobotique liés à la distance séparant les deux sites sont éliminés (problème de délai, image bruitée, etc. ...).

### **f. Médical**

Dans les laboratoires d'opérations chirurgicales, on a tendance à réduire la taille de l'accès à la zone à opérer dont l'objectif est de réduire le risque d'infection, la morbidité postopératoire et bien sûr le temps d'hospitalisation. Ces techniques se nomment vidéoscopie et endoscopie. Ces deux techniques fonctionnent très bien, mais la visibilité du chirurgien et l'accessibilité à la zone à opérer sont fortement réduites. Grâce à la RV et la RA, nous allons pouvoir enfin résoudre ce problème.

À l'aide d'échographie, de scanographie ou d'imagerie par résonance magnétique nucléaire, il est possible de reconstituer la zone à opérer en une image tridimensionnelle visible à l'aide d'un système de RV. Ce procédé se nomme téléchirurgie. Le chirurgien a donc une meilleure visibilité et puisqu'il peut grossir l'image qu'il reçoit, il est même plus précis.

Les techniques de RV et de RA offrent plus d'avantages aux jeunes chirurgiens de s'entraîner et de pratiquer des opérations chirurgicales sur des modèles virtuels avant de les pratiquer sur de vrais sujets. Ces techniques, augmentent la précision de l'acte chirurgical et la visibilité.

Dans le domaine de la téléchirurgie, le chirurgien opérerait sur un modèle virtuel, et un robot assistant opérerait réellement le patient. Les capteurs d'efforts devraient, dans ce cas, fournir et faire ressentir, les plus légers efforts appliqués par les outils chirurgicaux du robot.

Le 28 Février 2003, le premier service de chirurgie télérobotique a été établie entre Saint-Joseph de Hamilton (un hôpital d'enseignement affilié à l'Université McMaster) et l'Hôpital général North Bay (un hôpital communautaire) situé à 400 km. Le service a été conçu pour offrir la chirurgie télérobotique et l'assistance par des chirurgiens experts aux chirurgiens locaux à North Bay, et d'améliorer la gamme et la qualité des chirurgies laparoscopies avancées offertes localement [20].

## **IV-5. Téléopération/Télérobotique via Internet**

### **IV-5-1. Bref historique**

Dans la plupart des systèmes de téléopération classiques l'interface homme-machine nécessite un matériel complexe. Chaque panne survenue au niveau de la machine cliente provoque plus de difficultés, techniques et économiques, pour poursuivre une mission donnée (machine dédiée). En plus, la communication entre les deux sites (maître et esclave) utilise des canaux de communication dédiés. Ce qui fait, chaque déplacement du site esclave et/ou du site maître exige le déplacement du matériel ainsi que la reconfiguration réseau du canal de communication reliant ces deux sites. Dans ces conditions, les opérateurs sont obligés de se déplacer sur les nouveaux sites client pour travailler et pour préparer les différentes missions.

Cependant, avec l'utilisation d'Internet, comme moyen de communication, le système de téléopération permet de fournir plus de rentabilité, de souplesse et de facilité à accéder à des systèmes de contrôle qui ne sont pas limités à une région géographique précise. Ce qui conduit à une délocalisation facile des opérateurs à faible coût et l'amélioration de l'ergonomie des postes maîtres par le développement des interfaces plus sûres et plus fiables.

Cette idée a été exploitée par la NASA lors de la préparation de la mission *sojourner* sur Mars. Initialement les scientifiques étaient obligés de se déplacer jusqu'au centre de contrôle en *Californie* afin de travailler et de contrôler le robot, sachant que la communication Terre-Mars se fait avec une bande passante très limitée. Le développement d'une interface Internet, a donné la possibilité aux scientifiques de travailler et de collaborer ensemble pour contrôler le robot de n'importe où dans le monde [22] [36].

Les interfaces Internet ont été développées depuis les années 1980, bien avant l'introduction du World Wide Web en 1992. Le premier dispositif distribué sur Internet a été le «Cambridge Coffee pot». Ce système utilisait une Webcam pour transmettre des images d'une cafetière placée dans le

laboratoire d'informatique de l'Université de Cambridge [37]. Le projet Mercury [39] a été mis en ligne sur Internet en août 1994. Il permettait aux utilisateurs d'attraper et de manipuler différents objets en utilisant un bras robotique. En septembre 1994, Ken Taylor, de l'Université Western Australie, a mis en fonction une télémanipulation d'un robot à six axes avec une caméra fixe d'observation [45]. Bien que le système initial de Taylor exige aux utilisateurs de taper des coordonnées spatiales pour spécifier les mouvements du manipulateur, lui et ses collègues ont ensuite exploré une variété d'interfaces utilisateur pour simplifier le contrôle des robots [46]. Toujours en Octobre 1994, Wallace a exposé une caméra robotisée [47] et Cox a mis en œuvre un système qui permet aux utilisateurs distant d'utiliser une interface Web pour partager des photos à partir d'un télescope robotique [48]. Un robot de Peinture PumaPaint (bras manipulateur PUMA 760), a été développé au USA-Pennsylvanie [43], capable de reproduire un dessin réalisé par un utilisateur depuis une interface de dessin dédiée. KhepOnTheWeb, un robot de navigation avec évitement d'obstacles (Un robot mobile miniature, type Khepera) a été développé à EPFL – Suisse [44]. Le système DIGIMUSE, conçu pour la visualisation interactive des objets d'art [41]. Le robot Xavier de l'Université Carnegie Mellon [42] est un robot de navigation avec évitement d'obstacles (robot mobile), le projet Museum Tour-guide [41] et le projet WebPioneer [32].

Au cours des dix dernières années, beaucoup de ces systèmes ont été mis en place par les différents chercheurs du monde entier. Dans tous les cas, ces recherches ont pour but :

- Etudier les différents problèmes engendrés par la téléopération via l'Internet (Latence, stabilité, le débit de transfert de donnée, la qualité de service, etc.),
- Développement et conception des interfaces ergonomiques plus simples et plus fiables assurant l'adaptation rapide des opérateurs,
- Etudier les problèmes de sûreté de fonctionnement notamment pour les systèmes qui travaillent en long durée (24 heures sur 24, 365 jours par an),
- L'amélioration, via l'interface, des temps d'activité de l'opérateur,
- Le partage entre laboratoires de la commande de sites d'expérimentation distants,
- Etudier les architectures permettant d'interfacer un robot avec un serveur web.

Dans ce qui suit, nous citons quelques sites Internet développés par les différents chercheurs dans le domaine de la télérobotique [38] :

Ken Goldberg's Telegarden, <http://www.usc.edu/dept/garden/>

Railroad at the University of ULN, Denmark, <http://rr-vs.informatik.uni-ulm.de/rr/>

A web interface for NASA's Sojourner Rover, <http://mars.graham.com/wits/>

Lunar Rover Initiative, <http://www.frc.ri.cmu.edu/projects/lri/>

CyberCut, <http://CyberCut.berkeley.edu/>

Mercury Project, <http://www.usc.edu/dept/raiders/>

Drinking Maiden, <http://digimuse.usc.edu/robot/>

PumaPaint, <http://yugo.mme.wilkes.edu/?villanov/>

Robotic Telescope Observatory site, <http://www.eia.brad.ac.uk/rti/>

Ken Taylor's Telerobot, <http://telerobot.mech.uwa.edu.au/>

IV-5-2. Architecture matérielle et logicielle

La figure IV-10 montre l'architecture, logicielle et matérielle, générale d'un système de téléopération robotisée. L'opérateur a besoin d'interagir avec le robot à distance via l'interface utilisateur. Parce que les systèmes de téléopération robotisée sont ceux basés sur Internet, l'interface utilisateur est une applet dans un navigateur Web ou une application Internet autonome [49] [50]. L'interface peut être créée par l'utilisation de HTML et CGI, ou JAVA. Il existe plusieurs modèles pour l'interface utilisateur: modèle du robot et de l'environnement virtuel, le modèle des images vidéo, un modèle de retour des capteurs, modèle de retour pour le dispositif de réalité augmentée et le modèle de téléopération.

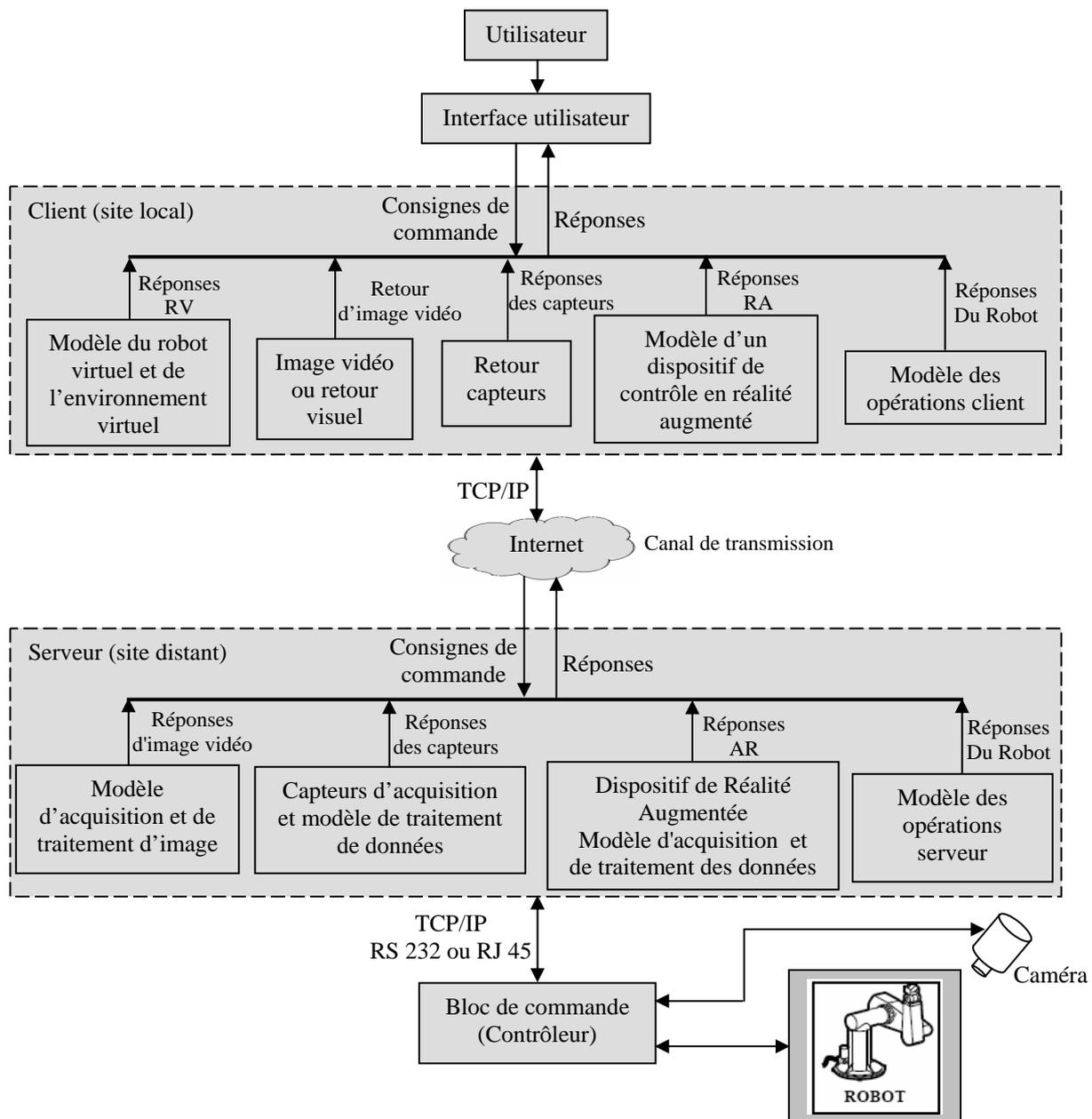


Figure IV-10 : Architecture, logicielle et matérielle, générale d'un système de téléopération

Les modèles du robot et de l'environnement virtuels sont utilisés pour simuler le robot réel et de l'environnement réel. L'opérateur peut commander le robot réel en faisant fonctionner le robot

virtuel dans l'interface utilisateur. Les autres modèles peuvent également être combinés ensemble pour fournir des informations de différents formats, y compris l'image ou la vidéo. Les données d'images vidéo peuvent aussi être utilisées pour mettre à jour le robot virtuel et l'environnement dans le processus de contrôle. Le modèle de retour capteur peut également fournir d'autres informations, y compris la pression, la tactilité, la température, goût, etc. Les retours des dispositifs de réalité augmentée peuvent aider l'utilisateur à obtenir plus d'immersion de contrôle de l'environnement virtuel.

Toutes les commandes et les exigences du côté client peuvent être envoyées sur le côté serveur par l'intermédiaire de l'Internet. Du côté du serveur, l'ordinateur serveur peut envoyer des commandes au contrôleur du robot réel, de faire l'acquisition et le traitement des données des différents capteurs, des caméras ou des dispositifs de réalité augmentée. Après le processus, le serveur envoie les données traitées vers le client via Internet.

Le contrôleur de robot commande le robot réel pour faire ce que le client a besoin.

**Remarque :** Pour plus de définitions et principes sur les protocoles de transfert des données et de communication entre machines via internet voir « Les Notions du réseau informatique » en Annexes.

#### IV-6. Application à la téléopération du bras manipulateur Puma 600

Afin de mettre en application nos expériences de téléopération, nous avons utilisé un robot virtuel Puma 600 dont la structure est donnée par la figure IV-11.



**Figure IV-11 :** Structure de l'interface virtuelle réalisée du robot Puma 600

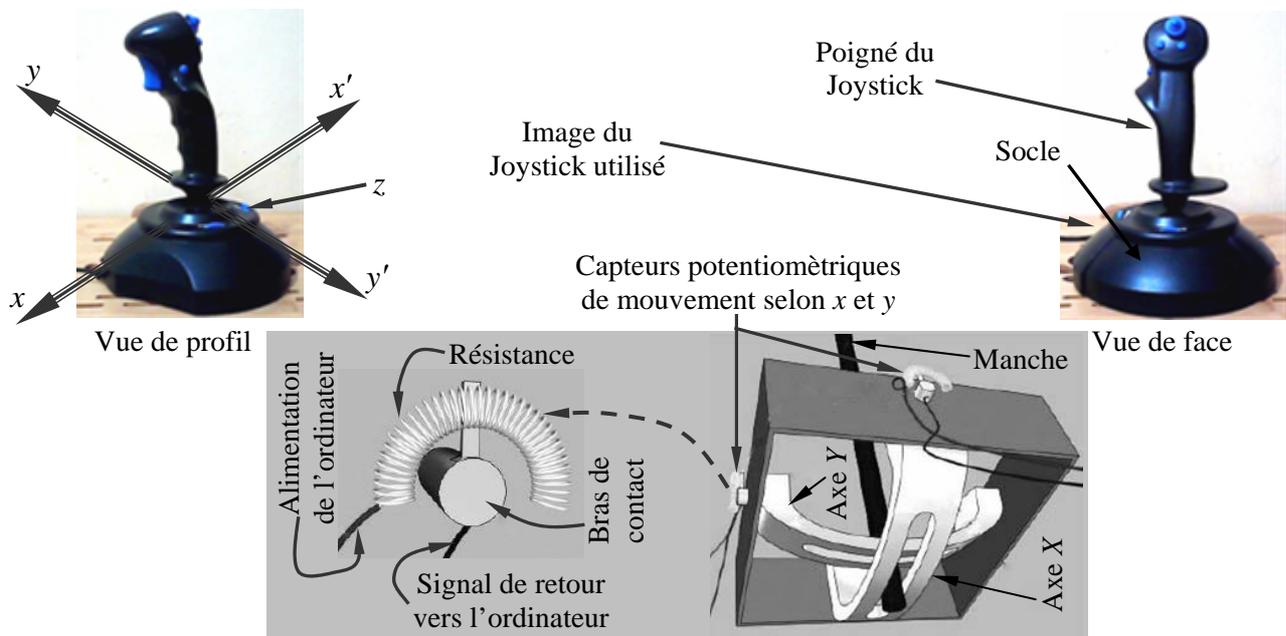
Pour la réalisation de l'interface virtuelle de ce robot, nous avons recouru à l'utilisation du logiciel de simulation mécanique *Solide Works* qui nous permet de réaliser les différents segments de notre bras manipulateur. Ensuite converti et enregistrés dans matlab (.vrl). L'assemblage des différents segments et l'animation 3D de notre robot sont réalisés par l'utilisation de la boîte à outils Virtual reality Toolbox vrml (*Virtual Reality Modeling Language*, un standard de modélisation 3D) sous matlab/simulink.

Simulink 3D Animation offre aussi une interface reliant les modèles Simulink et MATLAB à des algorithmes 3D des objets graphiques. Il nous permet de visualiser et de vérifier le comportement du système dynamique dans un environnement de réalité virtuelle.

#### IV-6-1. Téléopération par l'utilisation d'un joystick.

Un joystick est un dispositif qui permet à l'utilisateur de contrôler manuellement (télémanipulation), en temps réel, la position/vitesse d'un robot. Les joysticks trouvent leurs applications dans de nombreux domaines tels que : l'industrie (grues, machines agricoles, ...), l'aviation (simulateurs de vol civils), la chirurgie, l'aide aux handicapés, militaires et principalement pour des applications professionnelles de téléopération.

La figure IV- 12, présente le joystick utilisé dans notre réalisation pour la manipulation d'un bras virtuel Puma 600.



**Figure VI- 12 :** Le Joystick utilisé dans la télémanipulation avec son principe de fonctionnement

La génération des positions  $x$  et  $y$  est assurée par le déplacement, avec une simple inclinaison du poigné, du manche qui se trouve entre deux arbres à fente suivant les deux axes  $x'$  et  $y'$ . Le poignet pivote sur le point d'intersection entre ces deux axes. Le retour du poignet à sa position d'équilibre est assuré par un ressort de rappel sur chaque axe. La position  $z$  est générée par une simple action de rotation sur un potentiomètre rotatif.

##### IV-6-1-1. Réalisation expérimentale

Afin de télémanipuler le bras virtuel Puma 600 avec un joystick, nous avons choisi le schéma de la commande en position point à point que nous avons développé au chapitre II (§ II-3-1).

La version de Matlab/Simulink que nous avons utilisé, permet de reconnaître directement notre joystick. La communication entre ce dernier et l'ordinateur est assurée par le port USB. La figure IV-13 présente le couplage entre le joystick et l'ordinateur.



Figure IV-13 : Vue de couplage par USB entre le joystick et l'ordinateur

Puisque les valeurs générées par le joystick sont des grandeurs définies dans l'espace cartésien  $(x, y, z)$ , donc deux possibilités de télémanipulation se présentent :

- soit dans l'espace articulaire, en considérant les valeurs  $(x, y, z)$  directement comme étant des positions articulaires.
- soit dans l'espace opérationnel et dans ce cas, en doit convertir ces valeurs en positions articulaires (changement de coordonnées). Cette conversion est réalisée soit par l'utilisation de la matrice jacobienne ( $J$ ), en réalisant la commande en vitesse cartésienne ( $\dot{q} = J^{-1}(q) \dot{X}$ , §I-3) ou par l'utilisation du modèle géométrique inverse (MGI) du robot en réalisant une commande en position opérationnelle ( $q = g(X)$ , §I-6-2).

#### A) Télémanipulation dans l'espace articulaire

Dans cette expérience, nous considérons les entrées désirées générées par le joystick, comme étant des positions articulaires qui sont définies par :

$$\begin{cases} q_1 = z, & \text{position du 1er segment} \\ q_2 = x, & \text{position du 2ème segment} \\ q_3 = y, & \text{position du 3ème segment} \end{cases}$$

Le principe et la mise en œuvre de cette application est donné par le schéma bloc simulink de la figure IV-14.

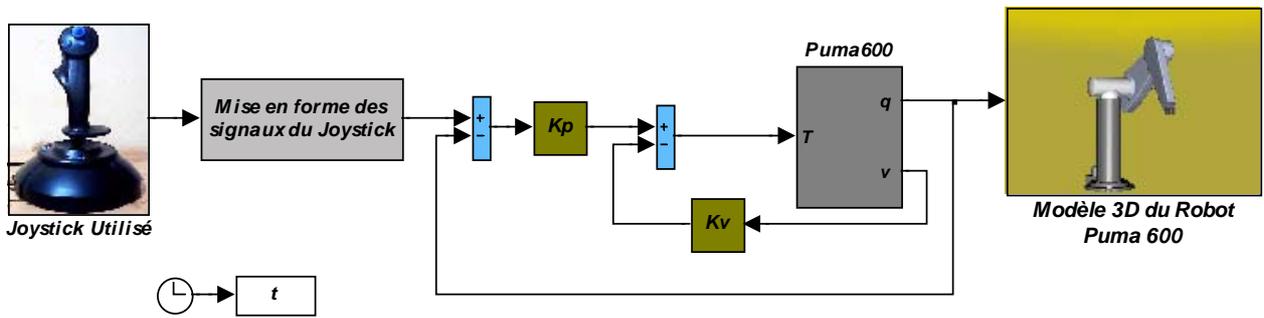


Figure IV-14 : Implémentation de la télémanipulation avec un joystick dans l'espace articulaire du bras virtuel Puma 600 sous Matlab/Simulink

La figure IV-15, représente un résumé sur le comportement en position du robot virtuel Puma 600 correspondant aux différentes manipulations du joystick (une succession de positions articulaires de type échelon).

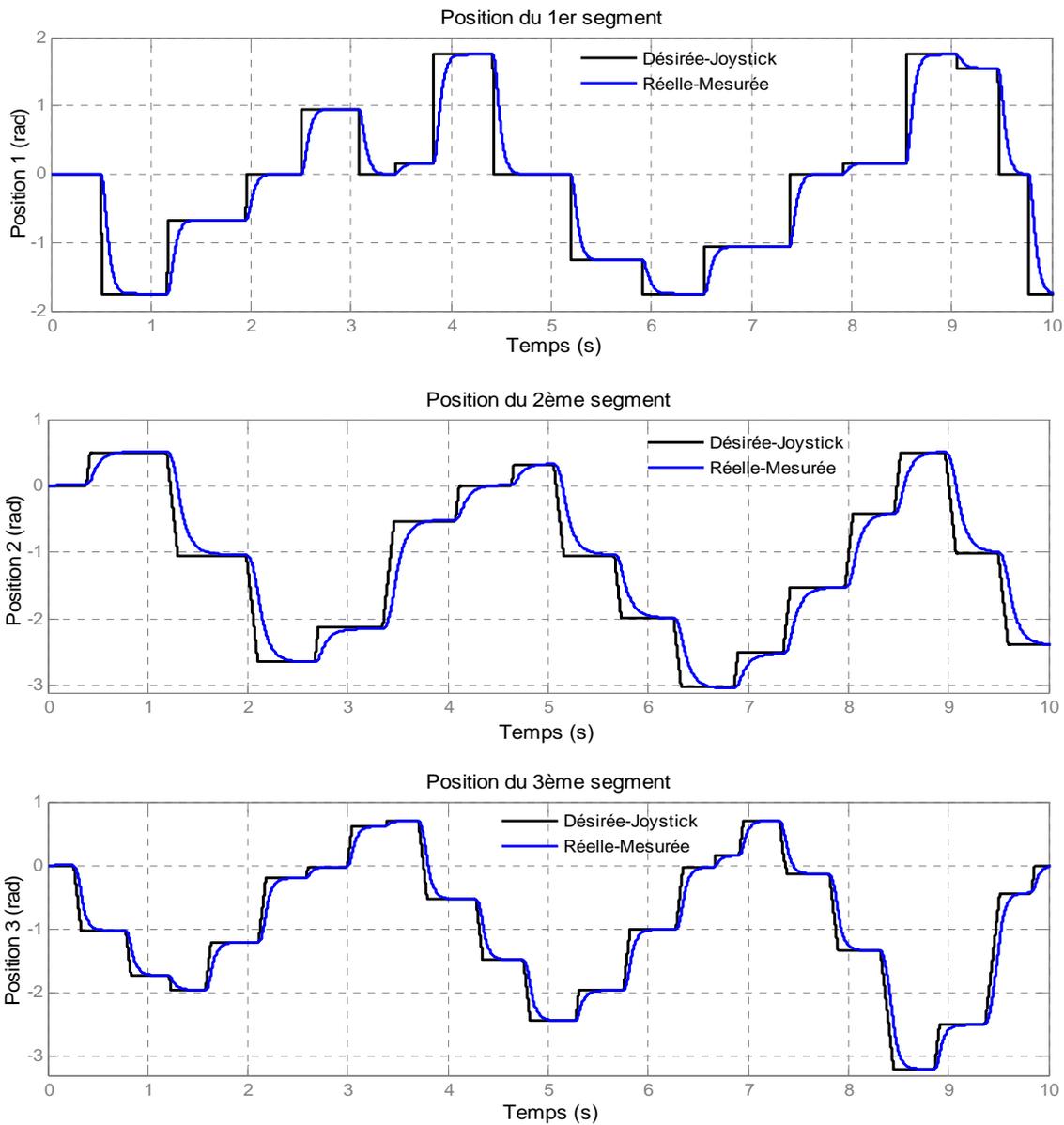
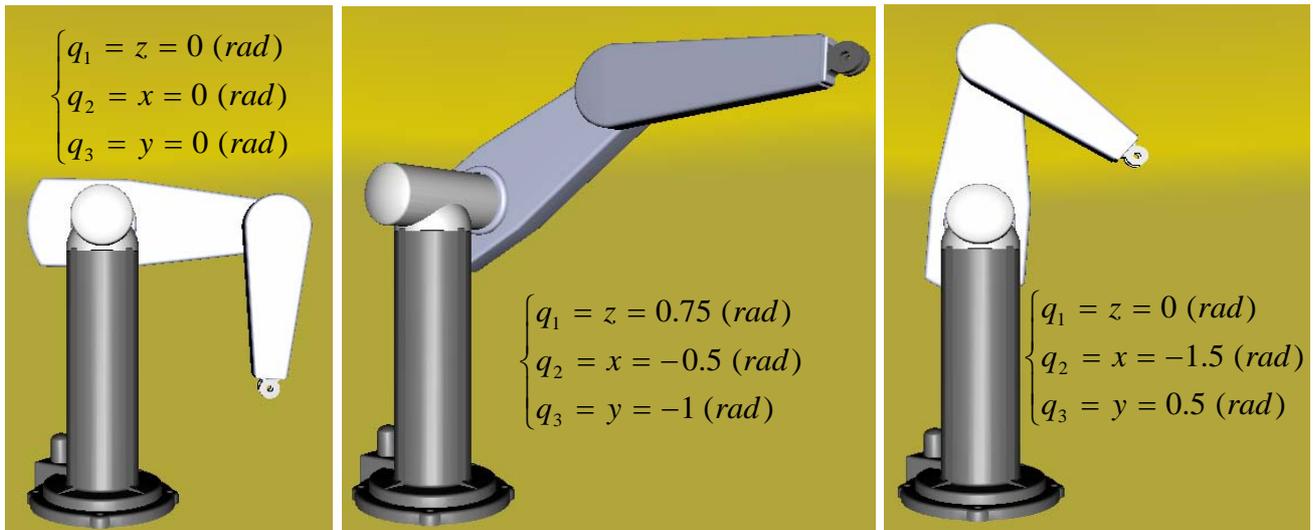


Figure IV-15: Comportement du robot virtuel Puma 600 pour les différentes manipulations du joystick

Les images du robot virtuel correspondantes à quelques positions générées par le joystick sont présentées sur la figure IV-16.



**Figure IV-16 :** Images du robot virtuel Puma 600 correspondantes aux différentes manipulations en positions articulaires du joystick

### Analyse des résultats

Les résultats présentés dans les figures IV-15 et IV-16, montrent que le robot suit d'une façon acceptable les différentes positions générées par le joystick. La commande des différents segments, dans ce cas de télémanipulation, se fait séparément c-à-d : articulation par articulation.

La télémanipulation par un joystick dans l'espace articulaire présente quelques avantages :

- un mouvement minimal sur chaque articulation,
- nécessite moins de calcul en ligne puisqu'elle n'utilise pas de convertisseur de coordonnées,
- le mouvement n'est pas affecté par le passage sur les configurations singulières.

### B) Télémanipulation dans l'espace opérationnel

Dans cette application, nous considérons les entrées désirées générées par le joystick, comme étant des positions cartésiennes  $(x, y, z)$ .

Puisque l'architecture de notre commande nécessite l'utilisation des consignes articulaires, donc une transformation, par l'utilisation d'un convertisseur de coordonnées, des données  $(x, y, z)$  en grandeurs articulaires  $(q_1, q_2, q_3)$  est nécessaire.

Notre choix pour cette conversion est basé sur l'utilisation du modèle géométrique inverse (MGI) du robot. Cette technique conduit à la réalisation d'une commande en position opérationnelle (voir §I-6-2).

La mise en œuvre de cette application est donnée, par le schéma block simulink, sur la figure IV-17.

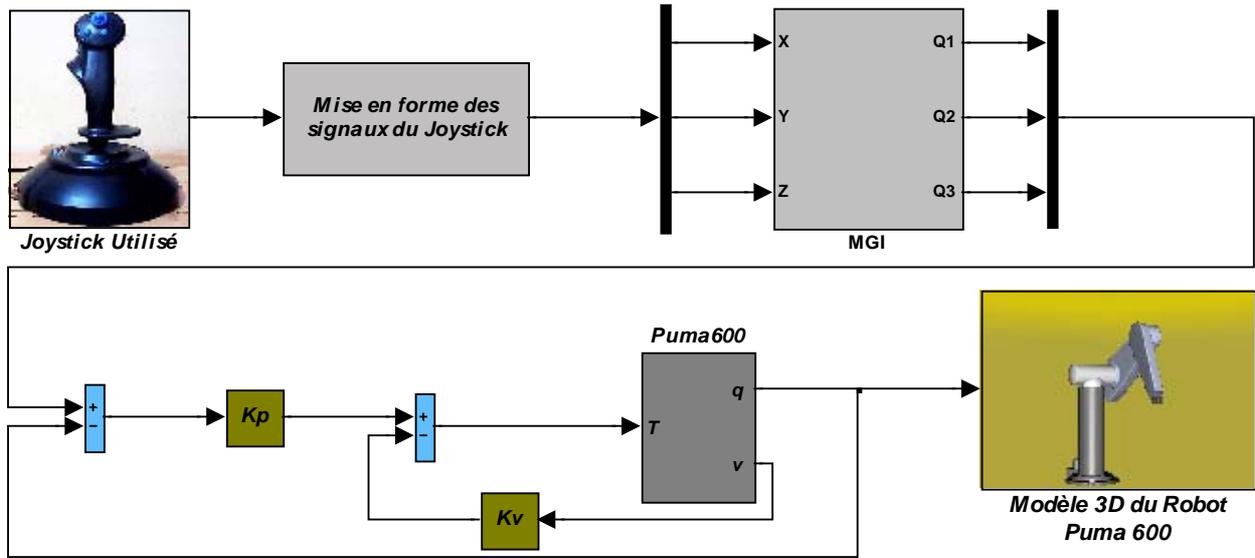
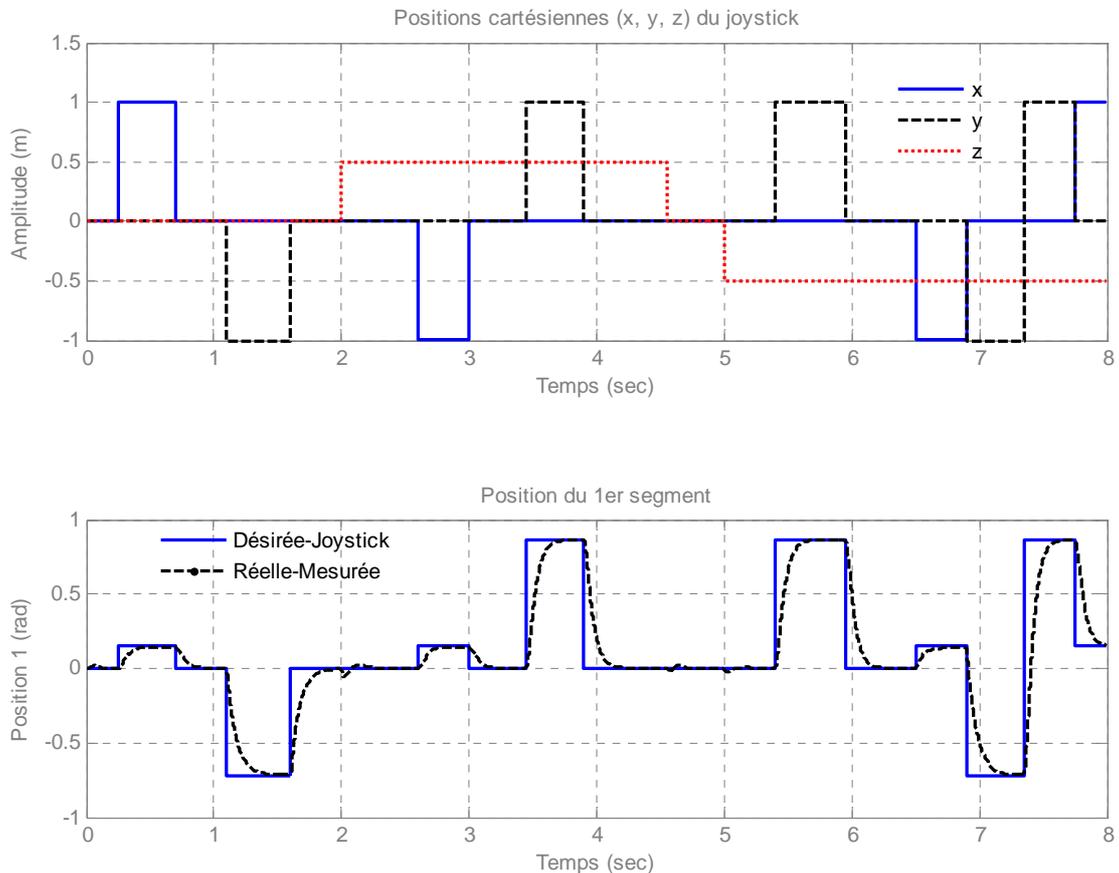


Figure IV-17 : Implémentation de la télémanipulation avec un joystick dans l'espace opérationnel du bras virtuel Puma 600 sous Matlab/Simulink

Le comportement en position du robot virtuel Puma 600 correspondant aux différentes manipulations du joystick dans l'espace opérationnel est donné par la figure IV-18.



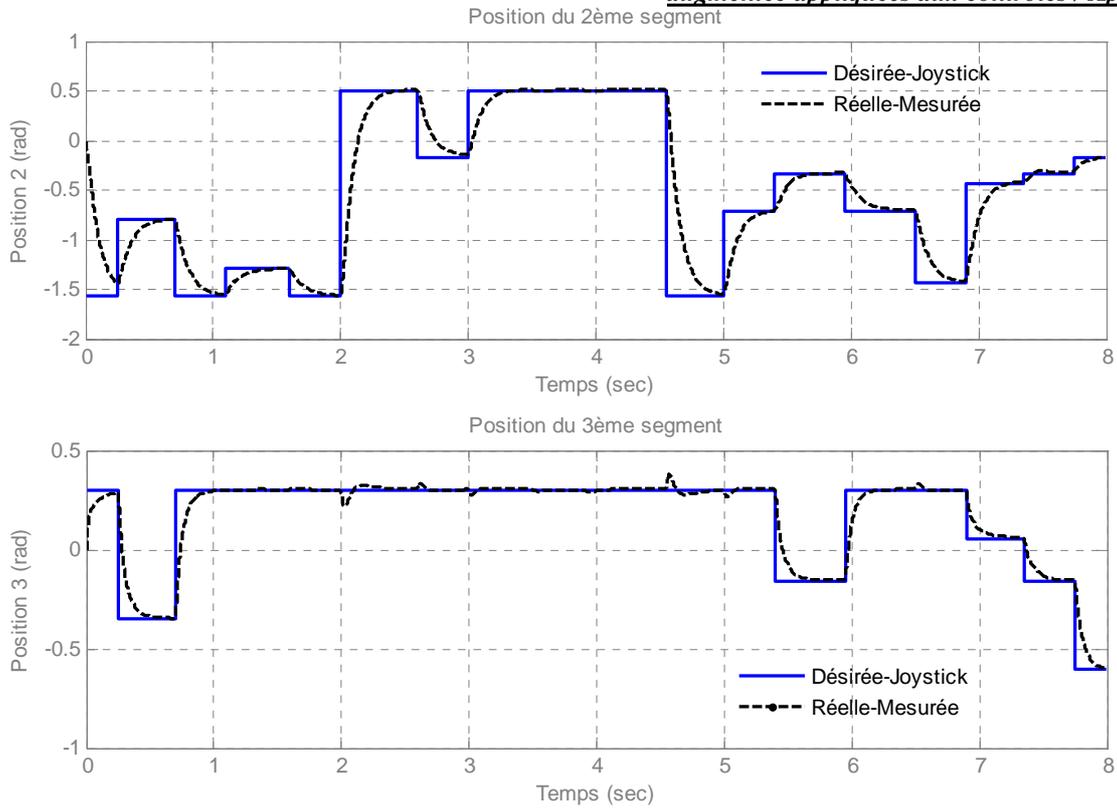


Figure IV-18: Comportement du robot virtuel Puma 600 pour les différentes manipulations du joystick dans l'espace opérationnel

Les images du robot virtuel correspondantes à quelques positions générées par le joystick sont présentées sur la figure IV-19.

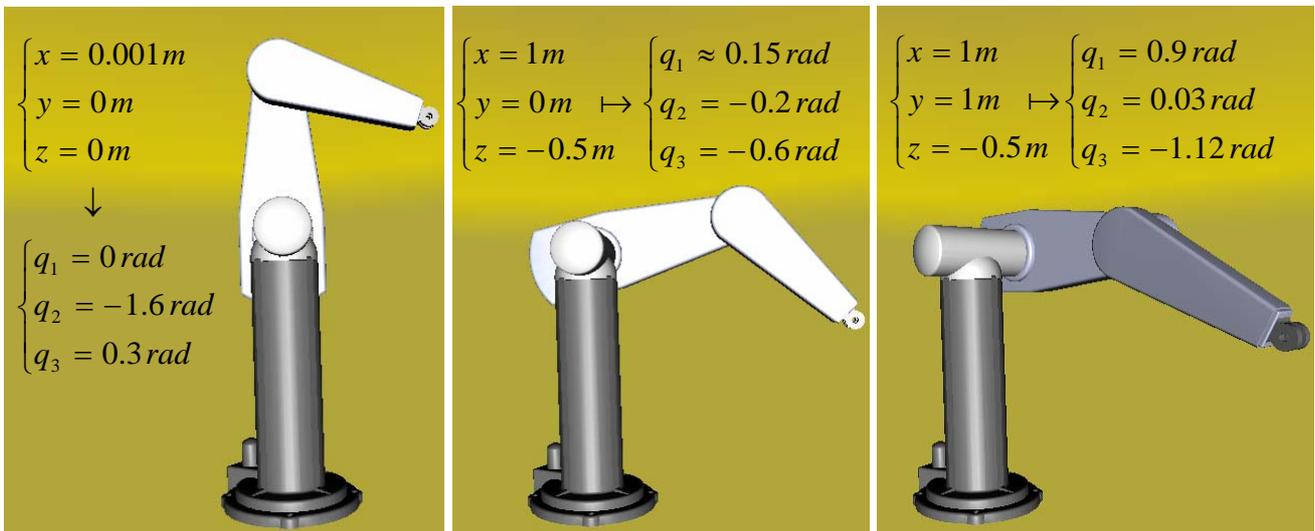


Figure IV-19 : Images du robot virtuel Puma 600 correspondantes aux différentes manipulations du joystick dans l'espace cartésien

### Analyse des résultats

Les résultats présentés dans les figures IV-18 et IV-19, montrent que le robot suit d'une façon tolérable les différentes positions générées par le joystick.

La commande des différentes articulations dans ce cas, après conversion des coordonnées cartésiennes par le MGI, ce fait simultanément. La télémanipulation par un joystick dans l'espace opérationnel, permet de contrôler la géométrie de la trajectoire, par exemple : réalisation d'un mouvement circulaire, une droite, un rectangle .... En contrepartie, elle présente quelques inconvénients :

- nécessite un convertisseur de coordonnées (transformation en coordonnées articulaires),
- elle peut être mise en échec lorsque le point généré par le joystick présente une position singulière,
- elle peut être mise en échec chaque fois que les points générés ne sont pas dans le volume accessible du robot.
- Généralement cette technique est limitée pour la réalisation des mouvements rapides dans un environnement qui ne présente pas d'obstacle (espace dégagé).

**IV-6-2. Téléopération distante par Internet du bras manipulateur Puma 600**

Dans cette application, nous présentons la structure de notre système de téléopération par Internet qui concerne la réalisation d'une interface ergonomique simple et fiable assurant une adaptation rapide de l'opérateur pour le contrôle et la planification de trajectoires du robot virtuel Puma 600.

Notre contribution dans cette partie est de résoudre quelques problèmes liés à la téléopération (Latence, fiabilité, etc.) par l'utilisation des outils de réalité virtuelle et de réalité augmentée, aussi de donner une certaine autonomie au robot distant, par l'introduction d'un contrôleur intelligent, dont l'objectif est de minimiser l'intervention de l'opérateur uniquement à sa supervision [51].

**IV-6-2-1. Description du système de téléopération développé**

L'architecture de notre système de téléopération est donnée par la figure IV-20. L'opérateur peut interagir avec le robot distant à travers l'interface utilisateur.

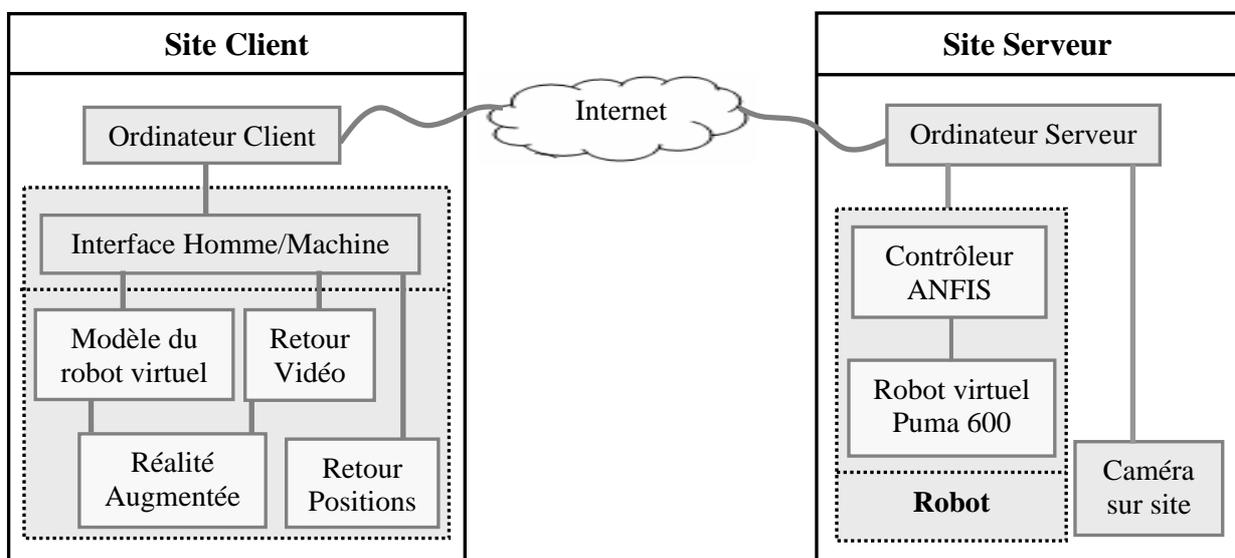


Figure IV-20: Architecture du system de téléopération

Dans cette architecture, nous avons :

- **Site Client :** Dans ce côté, un modèle virtuel 3D simule le robot réel. L'opérateur peut contrôler le robot réel par la manipulation du robot virtuel dans l'interface utilisateur. Le retour vidéo est utilisé pour mettre à jours le robot virtuel dans le processus de contrôle. Les retours positions sont utilisés pour fournir les positions réelles du robot distant (robot virtuel Puma 600). La réalité augmentée, la superposition de l'image virtuelle et de l'image réelle du robot, fourni à l'utilisateur une sensation d'immersion et pour mettre à jour l'environnement virtuel. Toutes les commandes et les exigences du côté client peuvent être envoyées au coté serveur via Internet.
- **Site Serveur :** Sur ce côté, l'ordinateur serveur peut envoyer des commandes au contrôleur du robot virtuel Puma 600, de faire les différentes acquisitions et le traitement des données des capteurs de positions et de la caméra vidéo. Après ces procédures, le serveur envoie les données traitées vers le client via Internet. Le contrôleur du robot contrôle le robot virtuel Puma 600 pour faire tout ce que le client lui exige.

**IV-6-2-2. Le système distant**

Pour réaliser notre système de téléopération par internet, le système distant (bras manipulateur Puma 600) est inséré dans une boucle de régulation avec un régulateur intelligent de type adaptatif neuro-flou (ANFIS) que nous avons développé dans le chapitre III (§III-3-3).

La figure IV-21 montre la description de ce système en boucle fermée. La commande *Couple calculé* de type *PD* basée sur le régulateur adaptatif neuro-flou (ANFIS) est appliquée au modèle dynamique du robot Puma 600. L'objectif est de forcer le robot à suivre une trajectoire désirée.

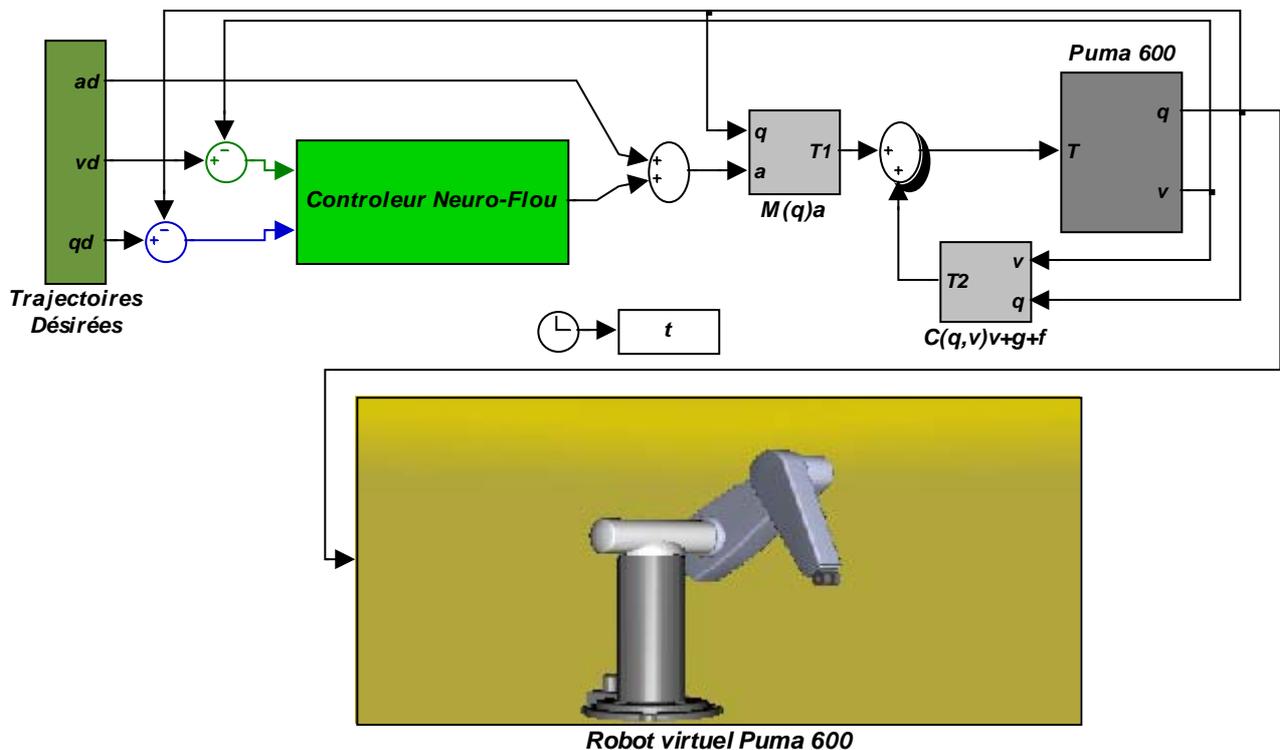


Figure IV-21 : Description du système distant

Ce système est implémenté sur l'ordinateur serveur (simulation sous Matlab/Simulink). Le serveur peut transmettre les commandes reçues à partir du client directement aux entrées de références (trajectoires désirées) et lire les positions mesurées du robot pour les transmettre, à son tour, au client.

#### IV-6-2-3. L'interface Homme-Machine

Afin de pouvoir réaliser cette interface, nous avons utilisé l'interface utilisateur GUI (*Graphical User Interface*) dans MATLAB qui nous offre plus d'outils de communication et d'animation.

La description de notre interface développée est donnée par la figure IV-22. Cette interface fournit tous les outils nécessaires pour assurer un bon contrôle du robot distant.

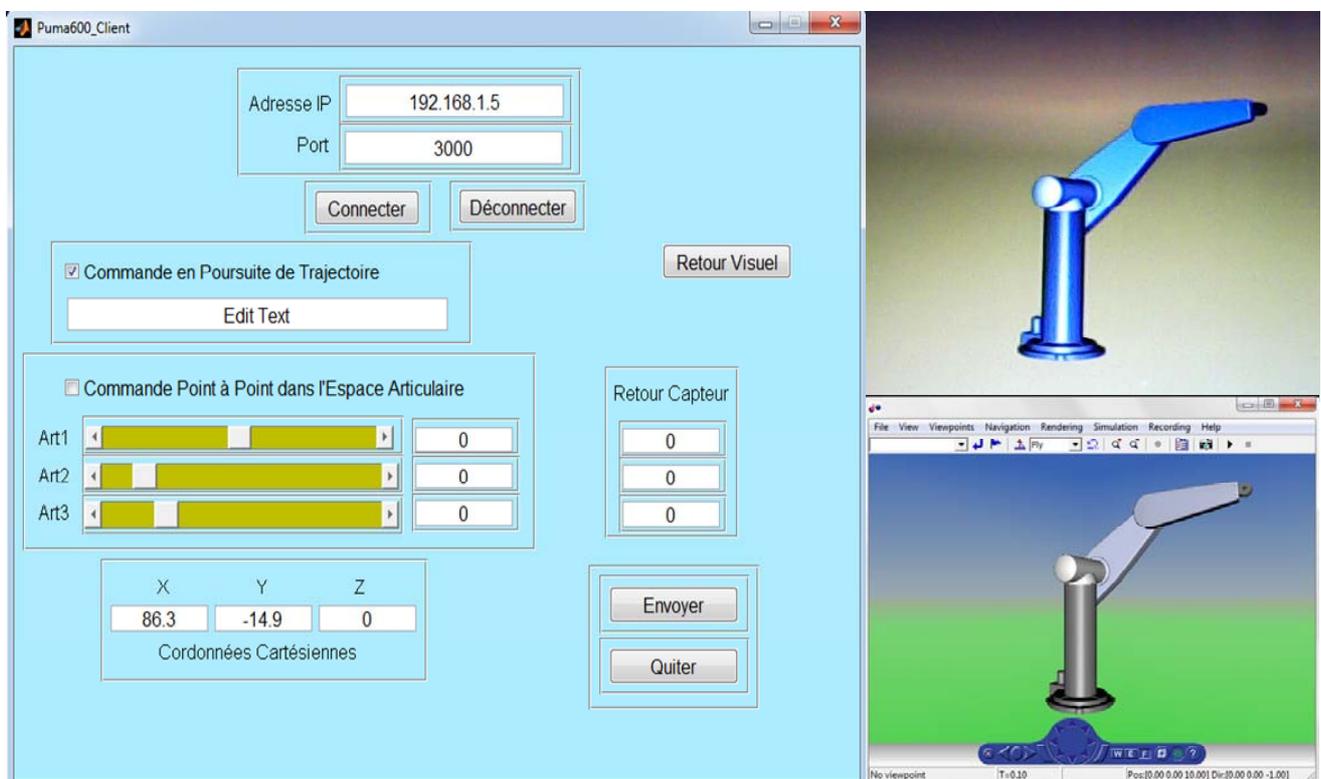


Figure IV-22 : L'interface graphique utilisateur développée

Sur cette interface, l'opérateur peut contrôler les axes du robot distant de deux façons:

- Soit par l'ajustement de la position du robot dans l'espace articulaire par l'utilisation des curseurs (sliders) ou par l'introduction des valeurs numériques de référence pour chaque articulation du robot dans la zone d'édition (coché la case de la commande points à point).
- Soit au moyen d'un planificateur de trajectoires avec l'utilisation de la commande de poursuite de trajectoire (coché la case de la commande en poursuite de trajectoire).

Nous trouvons aussi sur cette interface :

- Un modèle virtuel 3D simulant le robot Puma 600 permettant la visualisation en temps réel de la réalisation de sa tâche ;

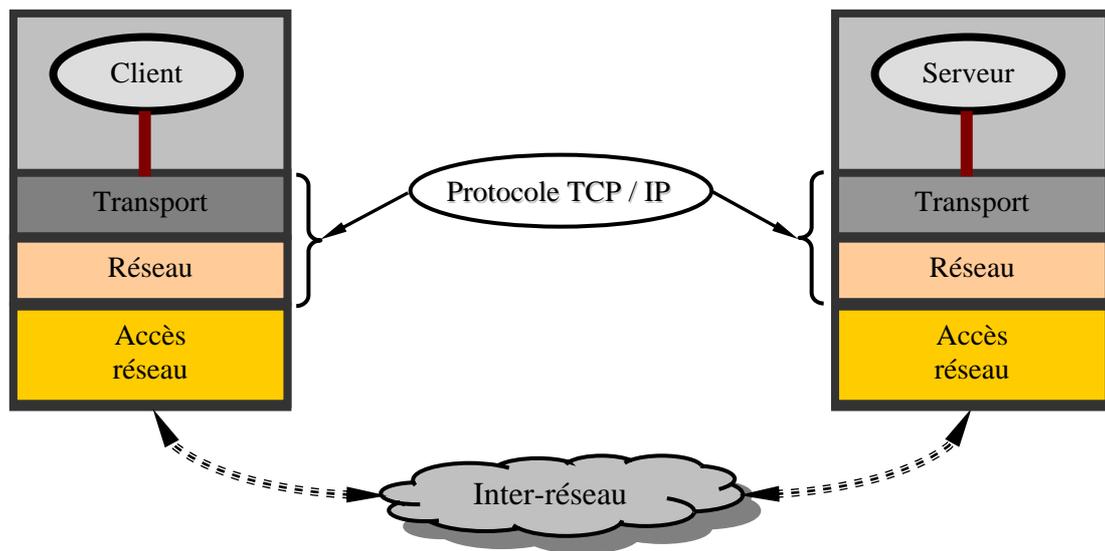
- Un retour vidéo pour la visualisation de la scène réelle et qui permet la mise à jour de l'image virtuelle ;
- Un retour des positions permettant de visualiser les positions réelles du robot ;
- Deux zones d'édition pour l'introduction de l'adresse IP et le numéro de port d'écoute du serveur pour assurer la connexion entre le client et le serveur.

#### IV-6-2-4. Communication Client-Serveur

Comme la plupart des programmes d'application, un client et un serveur utilisent un protocole de transport pour communiquer.

Comme le montre la figure IV-23, une application client ou serveur interagit directement avec le protocole de la couche transport pour établir une communication et pour envoyer ou recevoir des informations.

Le protocole transport utilise à son tour les protocoles de couches inférieures pour envoyer et recevoir tous les messages.



**Figure IV-23 :** Architecture de communication client-serveur avec le protocole TCP/IP

Pour assurer la communication et le transfert des données entre l'opérateur et le système distant, nous avons utilisé l'architecture client-serveur avec le protocole TCP / IP. La mise en œuvre de cette architecture est basée sur les sockets TCP / IP (voir Annexes (D)).

La procédure de communication commence par la mise en écoute du serveur et le client demande la connexion. Lorsque la connexion du client est acceptée, le serveur effectue le traitement des requêtes envoyées par le client. Le client envoie des requêtes sous forme de chaîne de commandes.

Afin d'effectuer un mouvement point à point, le client envoie les positions de références désirées ( $q_{d1}$ ,  $q_{d2}$ ,  $q_{d3}$ ), ensuite le serveur les transmet au contrôleur du robot. Les positions actuelles du robot sont envoyées au serveur. Le serveur à son tour, envoie ces informations au client.

Ces positions sont utilisées dans le site client pour actualiser les positions du modèle virtuel, et de les afficher sur l'interface client.

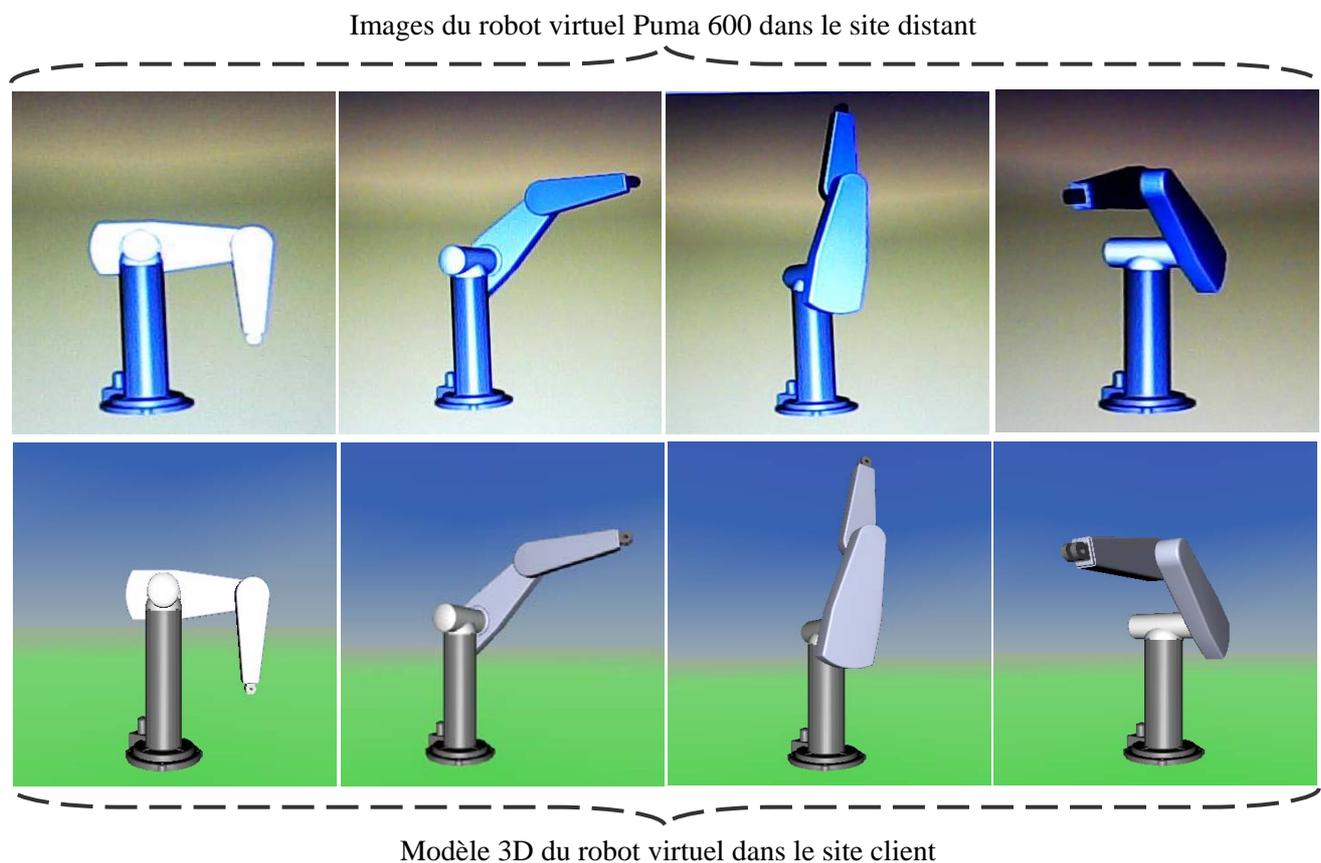
Dans le cas de la commande de poursuite de trajectoire, l'utilisateur peut entrer dans le planificateur de trajectoires les trajectoires désirées dans l'espace articulaire, et dans ce cas le message à envoyer contient le nombre de points de la trajectoire désirée. L'exécution de la trajectoire peut être visualisée en parallèle sur le modèle virtuel du robot et sur le retour vidéo à partir de la Webcam.

Les applications client et serveur ont été réalisées par l'introduction des commandes Java dans Matlab.

#### IV-6-2-5. Présentation des résultats

Afin de montrer l'efficacité de notre interface développée, nous nous limitons à présenter les résultats de télérobotique dans le cas de la commande en poursuite de trajectoire.

La figure IV-24, présente une démonstration de l'exécution d'une trajectoire planifiée par le client. A noter que les différentes séquences d'images sur cette figure correspondent à la même trajectoire mais pour des temps d'exécution différents.



**Figure IV-24:** Les positions du robot Puma 600, distant et virtuel, lors de l'exécution d'une trajectoire planifiée

#### A) Analyse des résultats

D'après les résultats de la figure IV-24, il est clair que la séquence d'images du robot distant est comparable à celle du robot virtuel dans le site client lors de l'exécution d'une trajectoire planifiée.

La commande en poursuite de trajectoire, avec l'introduction du contrôleur ANFIS améliore les performances du système et offre une certaine autonomie au robot distant et par conséquent, moins d'intervention de l'opérateur pour effectuer les tâches distantes. Dans le cas où l'environnement est connu à l'avance, le travail de l'opérateur peut être limité uniquement à la planification de la trajectoire et à la supervision sans contrôle permanent du robot téléopéré.

### B) Utilisation des techniques de réalité augmentée

Afin de valider les résultats présentés précédemment, nous faisons appel aux techniques de réalité augmentée (superposition d'images virtuelles sur des images vidéo). Pour ce faire, nous avons utilisé la représentation en fil de fer de l'image virtuelle. L'interface 3D de matlab offre cette option d'affichage et de représentation en fil de fer.

L'algorithme de superposition que nous avons développé est donné par la figure IV-25.

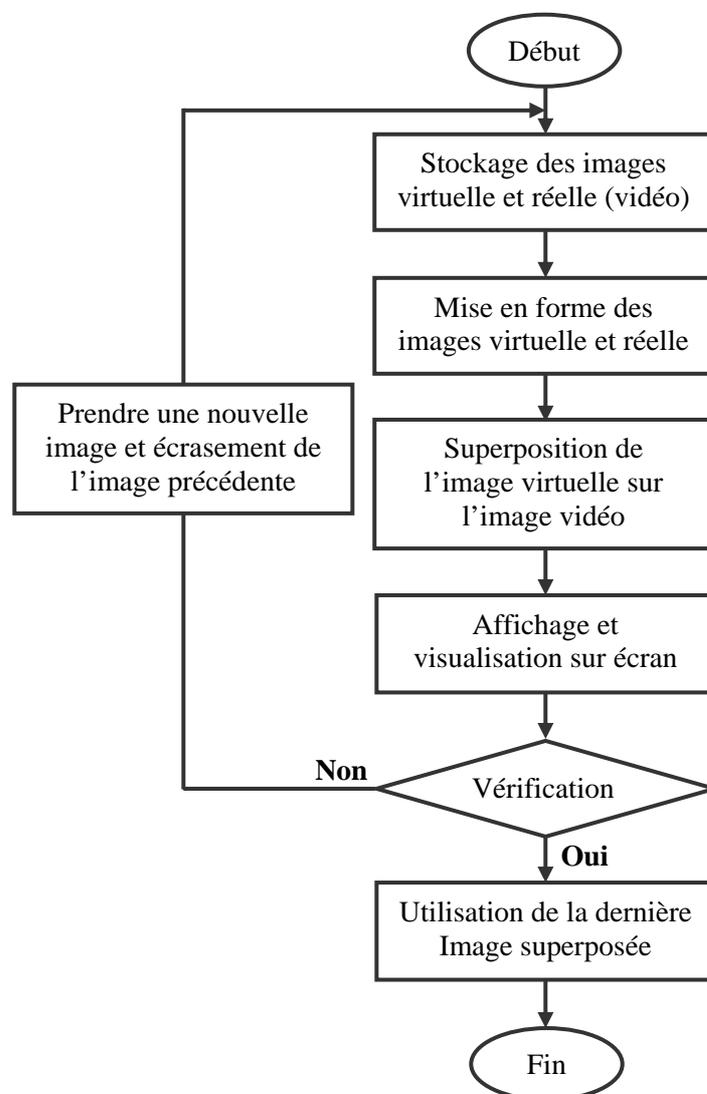


Figure IV-25 : Algorithme de superposition développé

La figure IV-26, représente la superposition d'une image réelle du retour vidéo et d'une image virtuelle du robot Puma 600. Cette superposition montre bien la réalisation de la tâche demandée par l'opérateur.



**Figure IV-26 :** Superposition de l'image virtuelle (en fil de fer) sur l'image réelle

Le résultat de la figure IV-26 conduit à conclure que le contrôle d'un robot distant peut se faire par le contrôle du robot virtuel sans attendre le retour des informations sur les positions et l'image réelle du robot distant.

### **Conclusion**

Ce chapitre présente un état de l'art sur les différentes techniques et systèmes de téléopération. Nous avons donné un aperçu historique sur le parcours du développement des techniques de la téléopération depuis sa naissance. Les principes, les différents problèmes rencontrés et les différents domaines d'application de la téléopération/télérobotique ont été détaillés. Nous avons présenté les différentes notions sur la réalité virtuelle et sur la réalité augmentée ainsi leurs domaines d'application et leurs utilisations pour le développement de la téléopération. Un bref historique sur l'architecture matérielle et logicielle de la téléopération/télérobotique via Internet ont été développés.

Deux applications de la téléopération ont été mise en œuvre :

- Télémanipulation d'un bras virtuel par un joystick (commande manuel), dans l'espace articulaire et dans l'espace opérationnel. Les résultats trouvés montrent la simplicité et la rapidité d'utiliser le contrôle dans l'espace articulaire que dans l'espace opérationnel en ce qui concerne le temps d'exécution (pas de convertisseur de coordonnées) et pas de problèmes de singularités pour les positions générées par le joystick.
- Téléopération distante par internet dont laquelle nous avons développé une interface homme-machine (IHM), plus fiable et plus simple à utiliser, pour le contrôle d'un robot virtuel Puma 600 (le modèle dynamique du robot est commandé par un contrôleur intelligent de type ANFIS). A travers les essais effectués sur cette interface, nous avons déduit que dans le cas où nous utilisons intelligemment les capacités de la réalité virtuelle et de la réalité augmentée, la téléopération est

possible même dans le cas d'existence de retards imprévisibles sur le net. L'un des principaux objectifs du système que nous avons développé est de fournir la possibilité de planifier et d'exécuter les trajectoires du robot distant à travers Internet. L'introduction d'un contrôleur intelligent (ANFIS) fournit une certaine autonomie au robot distant et évite son contrôle permanent, ce qui réduit l'effort de l'opérateur à sa supervision.

### Références bibliographiques

- [1] Goertz, R.C., Thompson, W.C., *Electronically Controlled Manipulator*, Nucleonics, pp. 46-47, Nov. 1954.
- [2] J. Vertut, P. Coiffet, *Téléopération : Evolution des Technologies*, vol. 3A et 3B, Editions Hermès, Paris, France, 1984.
- [3] 2601050 *Robotics and Teleoperation Lecture Notes*, Tampere University of Technology [Online]: [http://www.iha.tut.fi/education/IHA-3506/book/Teleoperation Notes 2004.pdf](http://www.iha.tut.fi/education/IHA-3506/book/Teleoperation%20Notes%202004.pdf)
- [4] D.R. Yoerger and J. E. Slotine, 1991, *Adaptive sliding control of an experimental underwater vehicle*, in Proceedings of 1991 IEEE International Conference on Robotics and Automation, Vol. 3, 9-11 April, pp. 2746-2751.
- [5] D. R. Yoerger, J. Newman, and J. E Slotine, 1986, *Supervisory control system for the JASON ROV*, IEEE Journal of Oceanic Engineering, Vol. 11, Issue 3, July 1986, pp. 392 – 400.
- [6] C. Sayers, M. Stein, A. Lai, R. Paul, *Teleprogramming to Perform Sophisticated Underwater Manipulative Tasks Using Acoustic Communications*, Proceedings of the IEEE Oceans'94, 1994.
- [7] D.Maddalena, W.Prendin et A. Terribile, *Supervisory Control Telerobotics reaches the Underwater Work Site*, Proc. of the 6th IARP Workshop on Underwater Robotics, Toulon, France, March 27-29 1996.
- [8] M.C. Nechyba, Y. Xu, *SM<sup>2</sup> for New Space Station Structure: Autonomous Locomotion and Teleoperation Control*, Proceedings of IEEE International Symposium on Robotics and Automation, pp. 911-918, 1993.
- [9] S. Hayati, R. Volpe, P. Backes, J. Balaram, R. Welch, R. Ivlev, G. Tharp, S. Peters, T. Ohm, R. Petras, *The Rocky 7 Rover: A Mars Sciencecraft Prototype*, Proceedings of the IEEE International Conference on Robotics and Automation 1997 (ICRA'97), Albuquerque, Etats-Unis, 20-25 avril 1997.
- [10] GUTHART G., SALISBURY J.K., *The intuitive telesurgery system: overview and applications*, IEEE International Conference on Robotics and Automation ICRA'00, San Francisco, p. 618-621, 2000.
- [11] GHODOUSSI M., BURTNER S. E., WANG Y., *Robotic surgery- the transatlantic case*. IEEE International Conference on Robotics and Automation ICRA'02, p. 1882-1888, 2002.

- [12] GALERNE S., *Architecture ouverte de la commande adaptée à la robotique de coopération h/m. Application au domaine médicale*, Thèse de doctorat, Université Paris XII, 1989.
- [13] CAMMOUN, DTRICHE, LAUTURE, et LESIGNE, *Telerobotics in the service of disabled persons*, ORIA'94 de la téléprésence vers la réalité virtuelle, 5ème colloque international et convention d'affaires, p. 249-254, décembre, Marseille, 1994.
- [14] FUCHS P., MOREAU G., *Le traité de la réalité virtuelle*, v.1, *Les Presses de l'Ecole des Mines de Paris*, 2003 (2ème édition).
- [15] G.C. Burdea, P. Coiffet, *Réalité Virtuelle*, Ed. Hermes, 1993.
- [16] Heim, M. Featherstone, M. and Burrows, R. (1995). *The design of virtual reality*. In *Cyberspace/Cyberbodies/Cyberpunk. Cultures of Technological Embodiment*, London : Sage, pages 65–78.
- [17] Bowman, D. A. (1999). *Interaction Techniques for Common Tasks in Immersive Virtual Environments: Design, Evaluation, and Application*. PhD thesis, Georgia Institute of Technology.
- [18] Fuchs, P., Arnaldi, B., and Tisseau, J. (2003). *La réalité virtuelle et ses applications*, chapter 1, pages 3-52. Les Presses de l'Ecole des Mines de Paris.
- [19] Lanier, J. (1988). *A vintage virtual reality interview*. <http://www.jaronlanier.com/vrint.html>.
- [20] M. Anvari, *Remote telepresence surgery: the Canadian experience*, *Surg Endosc* (2007) 21: 537–541, Springer Science + Business Media, Inc. 2007
- [21] Burdea Grigore: C.. *Force and Touch Feedback for Virtual Reality*. John Wiley & Sons. 1996.
- [22] Otmane, S., *Télétravail Robotisé et Réalité Augmentée : Application à la Téléopération via Internet*, thèse de Doctorat, Université d'Evry-Val d'Essonne, Déc. 2000.
- [23] KHEDDAR A., et COIFFET P., *Téléopération et réalité virtuelle*, Traité IC2, série Systèmes automatisés, Hermès, 2002.
- [24] Sheridan, T. B. (1992). *Telerobotics, Automation and Human Supervisory Control*. The MIT Press, Cambridge, USA.
- [25] DELARMINA P., *Automatique : commande des systèmes linéaires*, Hermes, 1993.
- [26] SPONG M., *Communication delay and control in telerobotics*, *Journal of the Robotics Society of Japan*, v. 11, N°. 6, p. 803-810, 1993.
- [27] ANDRIOT C., *Automatique des systèmes de téléopération avec retour d'effort : limitation des performances*, Thèse de doctorat, Université Pierre et Marie Curie, 1997.
- [28] KOSUGE K., MURAYAMA H., et TAKEO T., *Bilateral feedback control of telemanipulators via computer network*, *IEEE/RSJ International Conference on Intelligent Robots and System, IROS'96*, v. 3, p. 1380-1385, Osaka, Japan, 1996.
- [29] D. Wettergreen, H. Pangels, J. Bares, *Behavior-based Gait Execution for the Dante II Walking Robot*, *IEEE International Conference on Intelligent Robots and Systems (IROS)*, USA, Août 1995.
- [30] I. Lin, F. Wallner, R. Dillmann, *An Advanced Telerobotic Control System for a Mobile Robot with Multisensor Feedback*, 4<sup>th</sup> *International Conference on Intelligent Autonomous Systems*, Allemagne, Mars 1995.

- [31] H. W. Stone, *Mars Pathfinder Microrover A Low-Cost, Low-Power Spacecraft*, AIAA Forum on Advanced Developments in Space Robotics, USA, Août 1996.
- [32] T. Fong, C. Thorpe, C. Baur, *Advanced Interfaces for Vehicle Teleoperation: Collaborative Control, Sensor Fusion Displays, and Web-based Tools*, Vehicle Teleoperation Interfaces Workshop, IEEE International Conference on Robotics and Automation, USA, April 2000.
- [33] T. Blackmon, L. Stark. *Model-Based Supervisory Control in Telerobotics*, Presence Vol. 5, No. 2, pp205-223, 1996.
- [34] HUNTER, JONES, SAGAR «A teleoperated microsurgical robot and associated virtual environment for eye surgery », *PRESENCE*, v. 2, N° 4, p. 265-280, 1994.
- [35] Sutherland, I. E. (1968). *A head-mounted three-dimensional display*. In IFIPS Conference, volume 33, pages 757–764.
- [36] P. G. Backes, K. S. Tao, et G. K. Tharp, *Mars pathfinder mission internetbased operations using wits*. IEEE International Conference on Robotics and Automation, pages 284–291, 1998.
- [37] Q. Stafford-Fraser, *The Trojan Room Coffee Pot: a (non-technical) Biography*, Document accessible via l'URL: <http://www.cl.cam.ac.uk/coffee/qsf/coffe.html>. Mai 1995.
- [38] Yan Liu, *A Study of Teleoproration of Robotic System via the Internet*, Master of Science, University of Alberta, Edmonton, Alberta Spring 2001.
- [39] K. Goldberg, M. Mascha, S. Gentner, et N. Rothenberg, *Desktop teleoperation via the world wide web*. IEEE International Conference on Robotics and Automation, Nagoya, Japan, Mai 1995.
- [40] S. Goldberg, G. A. Bekey, Y. Akatsuka, *DIGIMUSE: An Interactive Telerobotic System for Remote Viewing of 3D Art Objects*, IROS'98: Workshop on Web Robots, Canada, Octobre 1998.
- [41] W. Burgard, A. B. Cremers, D. Fox, G. Lakemeyer, D. Hähnel, D. Schulz, W. Steiner, S. Thrun, *The interactive museum tour-guide robot*, 15<sup>th</sup> National Conference on Artificial Intelligence, USA, Juillet 1998.
- [42] Simmons, R. *Xavier: An autonomous mobile robot on the web*. International Conference on Intelligent Robots and Systems IROS'98, pages 43–48, Victoria, Canada, Octobre 1998.
- [43] M. Stein, *Painting on the World Wide Web: The pumapaint project*. IEEE International Conference on Intelligent Robots and Systems IROS'98, pages 37–42, Victoria, Canada, Octobre 1998.
- [44] P. Saucy, et F. Mondala, *Khepontheweb: One year of access to a mobile robot on the internet*. IEEE International Conference on Intelligent Robots and Systems IROS'98, pages 23–30, Victoria, Canada, Octobre 1998.
- [45] Barney Dalton and Ken Taylor, *A framework for internet robotics*, IEEE International Conference on Intelligent Robots and Systems IROS'98, Workshop on Web Robots, Victoria, Canada, 1998.
- [46] <http://telerobot.mech.uwa.edu.au/>
- [47] <http://found.cs.nyu.edu/cgi-bin/rsw/labcam1>
- [48] <http://www.eia.brad.ac.uk/rti/>

- [49] Ho, T. and Zhang, H., *Internet-Based Tele-Manipulation*, IEEE Canadian Conference on Electrical and Computer Engineering, Edmonton, Canada, pp. 1425–1430, May 1999.
- [50] Xiaoli Yang, Qing Chen, *Virtual Reality Tools for Internet-Based Robotic Teleoperation*, IEEE International Symposium on Distributed Simulation and Real-Time Applications (DS-RT'04) 1550-6525/04, IEEE Computer Society, 2004.
- [51] B. Ouamri, Z. Ahmed-Foitih, *Internet-Based Teleoperation of a Puma 600 Virtual Robot with ANFIS Based Controller*, AWERProcedia Information Technology & Computer Science, Vol.04 (2013) 122-129, 3rd World Conference on Innovation and Computer Sciences (INSODE-2013).

## *Conclusion générale*

L'objectif de ce travail est le développement d'un système de télérobotique, pour le contrôle d'un robot Puma 600, fondé sur l'introduction des techniques d'intelligence artificielle pour la réalisation et la mise en œuvre d'un système intelligent distant de commande en position et en poursuite de trajectoire basé sur les concepts de la logique floue et les réseaux neuromimétiques assurant une certaine autonomie au robot distant et de combiner des planificateurs avec un système de téléopération dans un environnement virtuel par l'utilisation de la réalité virtuelle et de la réalité augmentée afin de limiter les actions de l'opérateur à la prise de décision et à la récupération des données (erreurs de précision sur la réalisation des tâches planifiés) et aussi pour minimiser le problème de délais de transmission sur le réseaux Internet.

Au cours du premier chapitre, nous avons présenté un état de l'art sur les notions et les techniques de modélisations appliquées aux bras manipulateurs. Nous avons détaillé les différentes techniques de modélisation géométrique, cinématique et dynamique utilisées pour l'étude et la conception des différentes lois de commande appliquées aux systèmes robotisés. Comme exemple d'application, nous avons montré toutes les démarches à suivre pour le calcul des différents modèles du robot Puma 600.

Dans le deuxième chapitre, nous avons présenté un état de l'art sur les différentes techniques et structures de commande classiques appliquées aux bras manipulateurs à savoir : les techniques de commande qui traitent le problème de régulation et celles qui traitent le problème de poursuite de trajectoire. Le problème de stabilité dans toutes les structures de commande que nous avons présenté a été traité. Nous avons détaillé les différentes architectures et structures de commande en effort et hybride position/force utilisées lorsque le robot rentre en contact avec son environnement.

Comme application des lois de commande classiques, nous avons mis en œuvre, par simulation, la commande de position *Point à Point* (régulation) et la commande linéarisante connu sous le nom *Couple Calculé* ou *Computed Torque* (poursuite de trajectoire). Les résultats de simulation montrent la stabilité des deux méthodes vue la convergence des erreurs de position et de poursuite de trajectoire. Les grandes valeurs des gains utilisés dans la première méthode présentent un inconvénient majeur sur les actionneurs du robot et l'implémentation de la deuxième méthode nécessite la connaissance précise et complète du modèle dynamique du robot. En général, le vecteur des couples de Coriolis, centrifuges et de gravité est très compliqué, par conséquent, très difficile à modéliser. Dans le but d'améliorer les performances de la deuxième commande, nous avons proposé des techniques simple et plus robuste basé sur les commandes : par la logique floue, par réseaux de neurones et par neuro-flou tout en gardant l'architecture de cette méthode.

Nous avons choisi, dans notre applications, la méthode de commande *Couple calculé* vue sa simplicité d'implémentation et qui présente une application spéciale de la linéarisation en boucle fermée des systèmes non linéaires et qui a gagné sa popularité dans la théorie des systèmes modernes. Elle nous permet de dériver facilement des contrôleurs très efficaces pour la commande des robots,

tout en fournissant un cadre générale pour rassembler des contrôleurs classiques et des techniques modernes de conception. Elle peut apparaître dans la commande robuste, commande adaptative, etc.

A travers le troisième chapitre, nous avons exposé un état de l'art sur les nouveaux outils de raisonnement intelligent utilisés dans les systèmes de contrôle et de commande automatique tels que : La logique floue, les réseaux de neurones et neuro-flou (ANFIS). Nous avons développé les différentes étapes à suivre pour la réalisation d'un régulateur floue et nous avons exposé les différentes architectures et modèles les plus utilisés pour le calcul et la réalisation des différents régulateurs neuronaux et aussi le système d'inférence floue (ANFIS).

Nous avons mis en application, par simulation, les trois techniques de commandes intelligentes : par logique floue ; par réseaux de neurones et par neuro-flou (ANFIS). Toutes ces commandes montrent leurs stabilités vues la convergence des erreurs de poursuite de position et de vitesse dans les trois cas. A travers les résultats de simulations obtenus, nous avons remarqué que la méthode de commande *Couple calculé* basée sur l'utilisation de la logique floue, par rapport à celle classique, montrent plusieurs avantages : un grand gain est appliqué seulement lorsque l'écart est très important et si l'écart est faible, le gain sera également faible. La robustesse de cette technique de commande réside dans son caractère adaptatif pour la détermination des gains du régulateur ce qui veut dire une commande adaptative directe. Ces gains adaptatifs permettent le lissage des signaux de commande évitant ainsi, toute sollicitation des actionneurs. La stratégie de commande par réseaux de neurones est moins robuste que celle basée sur la logique floue. L'avantage de ce régulateur par rapport à celui floue réside dans le temps d'exécution minimale dans la boucle de régulation. Les performances de la commande neuro-floue (contrôleur ANFIS) sont meilleures que celles par logique floue du point de vue robustesse et précision de suivi et de stabilité. Cette étude par simulation indique clairement les bonnes performances du contrôleur neuro-floue adaptatif, car il est fondamentalement de nature adaptatif. Il est surtout utilisé pour contrôler un système avec un modèle inconnu.

Dans le quatrième et le dernier chapitre, nous avons donné un état de l'art et un aperçu historique sur le développement et l'évolution des différents systèmes de téléopération. Nous avons détaillé les différents problèmes rencontrés et les différents domaines d'application de la téléopération/télérobotique. L'utilisation des nouveaux outils graphiques, de réalité virtuelle et de réalité augmentée ainsi leurs domaines d'application et leurs utilisations pour le développement et l'amélioration des interfaces homme-machine pour la téléopération ont été bien exposés. Nous avons présenté un aperçu historique détaillé avec une architecture matérielle et logicielle générale de la téléopération/télérobotique via Internet.

Comme application, nous avons développé une commande manuelle du bras virtuel Puma 600 par l'utilisation d'un joystick, dans l'espace articulaire et dans l'espace opérationnel. Les résultats trouvés montrent la simplicité et la rapidité d'utiliser le contrôle dans l'espace articulaire que dans

l'espace opérationnel en ce qui concerne le temps d'exécution (pas de changeur de coordonnées) et pas de problèmes de singularités pour les positions générées par le joystick.

Nous avons développé une interface homme-machine (IHM) plus fiable et plus simple à utiliser, contenant plusieurs outils graphiques et d'affichages 3D par l'utilisation des outils de réalité virtuelle et de réalité augmentée, pour la téléopération via internet du robot virtuel Puma 600. L'utilisation de la réalité virtuelle permet à l'opérateur de manipuler, par simulation, le robot, de programmer ses tâches à exécuter et de servir comme interface pour le piloter au cours d'une mission et la réalité augmentée fournit à l'utilisateur la possibilité de vérifier la réalisation des tâches demandées et une sensation d'immersion et pour mettre à jour l'environnement virtuel au cours du processus de contrôle.

A travers les essais effectués sur cette interface, nous avons déduit que dans le cas où nous utilisons intelligemment les capacités de la réalité virtuelle et de la réalité augmentée, la téléopération est possible même dans le cas d'existence de retards imprévisible sur le net. L'un des principaux objectifs du système que nous avons développé est de fournir la possibilité de planifier et d'exécuter les trajectoires du robot distant à travers Internet. L'introduction d'un contrôleur intelligent fournit une certaine autonomie au robot distant et évite son contrôle permanent, ce qui réduit l'effort de l'opérateur à la prise de décision, à la récupération des données et à sa supervision.

Comme perspectives :

- Nous souhaitons faire une implémentation du système développé pour le contrôle d'un robot réel avec le développement d'une interface détaillé de réalité augmentée en temps réel.
- Développement d'un système de télérobotique basé sur une évaluation correcte des forces d'interaction lorsque le robot distant rentre en contact avec son environnement pour la réalisation d'une commande hybride position/force.
- Développement d'un système de téléméasures et d'acquisition de données pour le contrôle distant des installations techniques et industrielles assurant la gestion d'un grand nombre d'instruments.
- Contribution au développement d'un système de téléenseignement pour la réalisation des travaux pratiques distants, la possibilité d'accéder aux différents documents et l'assistance à des cours et à des conférences en ligne.

# *Annexes*

## A. Notions sur les systèmes non linéaires

Un système non linéaire est totalement différent de celui d'un système linéaire. En effet, aucune propriété des systèmes linéaires, par exemple le principe de superposition, n'est vrai dans le cas non linéaire. Pour le cas non linéaire on parle du bassin d'attraction du point d'équilibre et les cycles limites qui ne se présentent pas dans le cas linéaire.

Plusieurs formes se présentent dans le cas des systèmes non linéaires :

- **Système non linéaire général.** Ce cas est donné par le système d'équations d'état suivant :

$$\begin{cases} \dot{x}(t) = f(x(t), u(t)) \\ y(t) = h(x(t)) \end{cases} \quad (1)$$

- **Système affine en la commande.** Est donné par :

$$\begin{cases} \dot{x}(t) = f(x(t)) + \sum_{i=1}^n u_i(t) g_i(x(t)) \\ y(t) = h(x(t)) \end{cases} \quad (2)$$

- **Système affine en l'état.** Donné par :

$$\begin{cases} \dot{x}(t) = A(u(t))x(t) + B(u(t)) \\ y(t) = C(u(t))x(t) \end{cases} \quad (3)$$

- **Système bilinéaire.** Donné par :

$$\begin{cases} \dot{x}(t) = Ax(t) + \sum_i u_i(t) D_i x(t) \\ y(t) = C(x(t)) \end{cases} \quad (4)$$

avec  $x(t)$  est le vecteur d'état,  $u(t)$  le vecteur de commande,  $y(t)$  est le vecteur de sortie ou de mesures,  $f, g$  et  $h$  sont des fonctions de classe  $C^\infty$  (cf. rappel suivant).

### Rappel sur les fonctions

- Une fonction  $f$  définie de  $\mathbb{R}^+$  dans  $\mathbb{R}^+$  est dite de classe  $K$ , si elle est continue, strictement croissante et  $f(0)=0$ .
- Une fonction  $f : [0, \infty] \rightarrow [0, \infty]$  est dite de classe infinie  $C^\infty$  si elle est continue, strictement croissante,  $f(0)=0$  et non bornée :  $f(t) \rightarrow \infty$  quand  $t \rightarrow \infty$ .

### A-1. Systèmes autonomes et non autonomes

Un système non linéaire est dit autonome ou bien indépendant du temps si le deuxième terme ne dépend pas explicitement de  $t$  :

$$\dot{x}(t) = f(x, u) \quad (5)$$

avec  $f : \Omega \rightarrow \mathfrak{R}^n$  une fonction continue sur un domaine  $\Omega$  de  $\mathfrak{R}^n$ .

Si le système dépend du temps, on dit qu'il est non autonome :

$$\dot{x}(t) = f(x(t), u(t)) \quad (6)$$

### A-2. Domaine d'attraction

Considérons le système non linéaire :

$$\dot{x}(t) = f(x(t), u(t)) \quad (7)$$

tel que  $f(0) = 0$ , admettant  $x = 0$  comme équilibre (on peut toujours ramener, par un changement de variable, le point d'équilibre à l'origine).

Le point d'équilibre  $x = 0$  du système (7) est dit attractif si pour toute solution  $x(t)$  de (7) on ait :

$$\|x(0)\| < r \Rightarrow \lim_{t \rightarrow \infty} x(t) = 0 \quad (8)$$

L'équilibre  $x = 0$  du système (7) est dit globalement attractif si pour toute solution  $x(t)$  de (7) on a :

$$\lim_{t \rightarrow \infty} x(t) = 0 \quad (9)$$

L'ensemble  $B$  défini par la propriété :

$$x(0) \in B \Rightarrow \lim_{t \rightarrow \infty} x(t) = 0 \quad (10)$$

s'appelle le bassin d'attraction de l'origine. Ainsi,  $x = 0$  est globalement attractif si  $B = \mathfrak{R}^n$ .

### B. Normes

Une norme est une généralisation des idées de la distance et de la longueur. Puisque l'analyse de la stabilité est basée sur la taille de quelques vecteurs et matrices, la description des normes est utile.

### B-1. Normes d'un vecteur

Une norme  $\|\cdot\|$  d'un vecteur  $x = [x_1, \dots, x_n]^T$  est une fonction à valeur réelles définie sur l'espace vectoriel  $X$  tels que :

- a-  $\|x\| > 0$ , pour tous  $x \in X$  avec  $\|x\| = 0$  si et seulement si  $x = 0$  ;
- b-  $\|\alpha x\| = |\alpha| \|x\|$ , pour tous  $x \in X$  et n'importe quel scalaire  $\alpha$  ;
- c-  $\|x + y\| \leq \|x\| + \|y\|$ , pour tous  $x, y \in X$ .

où  $|\alpha|$  est la valeur absolue de  $\alpha$  s'il est réel, ou le module de  $\alpha$  s'il est complexe.

Dans le cas où  $X = \mathfrak{R}^n$ , les définitions suivantes sont des normes importantes sur  $\mathfrak{R}^n$  :

- norme d'ordre 1 :

$$\|x\|_1 = \sum_{i=1}^n |x_i| \quad (11)$$

- norme d'ordre 2 (ou norme euclidienne) :

$$\|x\|_2 = \left( \sum_{i=1}^n |x_i|^2 \right)^{\frac{1}{2}} \quad (12)$$

- norme d'ordre  $p$  :

$$\|x\|_p = \left( \sum |x_i|^p \right)^{\frac{1}{p}} \quad (13)$$

- norme d'ordre infini :

$$\|x\|_\infty = \max_{1 < i < n} |x_i| \quad (14)$$

### B-2. Normes induite d'une matrice

Dans les équations mathématiques du modèle du robot, un vecteur particulier  $x$  peut être multiplié par une matrice  $A$  pour obtenir un autre vecteur  $y = Ax$ . Pour relier les tailles de  $x$  et  $y$ , nous définissons une norme induite de la matrice comme suit :

Soit  $\|x\|$  une norme donnée du vecteur  $x \in \mathfrak{R}^n$ , chaque matrice  $A \in \mathfrak{R}^{n \times n}$  a une norme induite définie par :

$$\|A\|_i = \max_{\|x\|=1} \|Ax\| \quad (15)$$

A partir de cette définition, nous pourrions montrer que ces normes induites des matrices satisfont la condition de la définition présentée au §B-1. Il est possible de montrer qu'elles satisfont la condition :

$$\|AB\|_i \leq \|A\|_i \|B\|_i \quad (16)$$

pour toutes matrices  $A$  et  $B$  de dimension appropriée. Des normes importantes dans  $\mathfrak{R}^n$  sont alors données comme suit : avec  $A = (a_{ij})$

$$\begin{aligned} \|A\|_{i,\infty} &= \max_i \sum_j |a_{ij}| \\ \|A\|_{i,1} &= \max_j \sum_i |a_{ij}| \\ \|A\|_{i,2} &= \sqrt{\lambda_{\max}(A^T A)} \end{aligned} \quad (17)$$

où  $\lambda_{\max}(\cdot)$  est la valeur propre maximale ( $\cdot$ ).

### B-3. Propriétés d'une matrice

Quelques propriétés de la matrice jouent un rôle important dans l'étude de la stabilité des systèmes dynamiques. Nous exposons quelques unes les plus utilisées :

#### Définition 1

- Une matrice réelle  $A \in \mathfrak{R}^{n \times n}$  est dite définie positive si  $x^T A x > 0$  pour tous  $x \in \mathfrak{R}^n$ , et  $x \neq 0$ .
- Une matrice réelle  $A \in \mathfrak{R}^{n \times n}$  est dite semidéfinie positive si  $x^T A x \geq 0$  pour tous  $x \in \mathfrak{R}^n$ .
- Une matrice réelle  $A \in \mathfrak{R}^{n \times n}$  est dite définie négative si  $x^T A x < 0$  pour tous  $x \in \mathfrak{R}^n$ , et  $x \neq 0$ .
- Une matrice réelle  $A \in \mathfrak{R}^{n \times n}$  est dite semidéfinie négative si  $x^T A x \leq 0$  pour tous  $x \in \mathfrak{R}^n$ .
- Une matrice  $A$  est dite indéfinie si  $x^T A x > 0$  pour certains  $x \in \mathfrak{R}^n$  et  $x^T A x < 0$  pour les autres  $x \in \mathfrak{R}^n$ .

#### Théorème 1

Soit  $A = (a_{ij}) \in \mathfrak{R}^{n \times n}$  une matrice réelle symétrique. Par conséquent, toutes les valeurs propres de  $A$  sont réelles. Nous avons donc :

- Une matrice réelle  $A \in \mathfrak{R}^{n \times n}$  est définie positive si toutes ses valeurs propres sont positives.
- Une matrice réelle  $A \in \mathfrak{R}^{n \times n}$  est semidéfinie positive si toutes ses valeurs propres sont non négatives.
- Une matrice réelle  $A \in \mathfrak{R}^{n \times n}$  est définie négative si toutes ses valeurs propres sont négatives.
- Une matrice réelle  $A \in \mathfrak{R}^{n \times n}$  est semidéfinie négative si toutes ses valeurs propres sont non positives.
- Une matrice réelle  $A \in \mathfrak{R}^{n \times n}$  est indéfinie si quelques unes de ses valeurs propres sont positives et les autres sont négatives.

## C. Définitions de la stabilité

Soit  $x_e$  un point d'équilibre au temps  $t_0$ , et  $\|\cdot\|$  dénote n'importe quelle norme de fonction précitée.

### C-1. Stabilité

Le vecteur  $x_e$  est stable à  $t_0$  dans le sens de Lyapunov, si le vecteur  $x_e(t)$  commence suffisamment rapproché de  $x_0$  à  $t_0$ , il reste toujours rapproché de  $x_e(t)$  à  $t \geq t_0$ . Plus précisément,  $x_e(t)$  est stable à  $t_0$  si  $\forall \varepsilon > 0, \exists \delta(\varepsilon, t_0) > 0$  tel que si :

$$\|x_0 - x_e\| \leq \delta(\varepsilon, t_0), \text{ alors } \|x(t) - x_e(t)\| \leq \varepsilon \text{ pour tous } t \geq t_0 \quad (18)$$

$x_e$  est stable au sens de Lyapunov, s'il est stable pour n'importe quel  $t_0$ . En plus,  $x_e$  est uniformément stable si  $\delta(\varepsilon, t_0)$  ne dépend pas de  $t_0$ .

### C-2. Instabilité

Le vecteur  $x_e$  est instable au sens de Lyapunov, si tout état commençant au voisinage de  $x_e$  à  $t_0$ , ne restera pas confiné à proximité de  $x_e$  à un temps postérieur. En d'autres termes,  $x_e$  est instable s'il n'est pas stable à  $t_0$ .

### C-3. Convergence

$x_e$  est convergent si pour chaque  $t_0 \in \mathfrak{R}$ , il existe un  $\eta(t_0) > 0$  tel que si :

$$\|x_0 - x_e\| \leq \eta(t_0), \text{ alors } \|x(t) - x_e(t)\| \rightarrow 0 \text{ lorsque } t \rightarrow \infty.$$

$x_e$  est uniformément convergent s'il y a un nombre  $\eta > 0$  tel que si :

$$\|x_0 - x_e\| \leq \eta, t_0 \geq 0 \text{ alors } \|x(t) - x_e(t)\| \rightarrow 0 \text{ pour tous } t \rightarrow \infty.$$

### C-4. Stabilité asymptotique

Le point d'équilibre  $x_e$  est asymptotiquement stable si les états  $x$  commençant suffisamment rapprochés de  $x_e$  vont rester ou éventuellement converger vers  $x_e$ . D'autre part, s'il est stable et convergent.

$x_e$  est uniformément asymptotiquement stable s'il est uniformément stable et uniformément convergent.

### C-5. Stabilité asymptotique uniforme globale

$x_e$  est globalement uniformément asymptotiquement stable si :

- il est asymptotiquement stable ;

- pour chaque paire de nombre positif  $M, \varepsilon$  avec  $M$  arbitrairement grand et  $\varepsilon$  arbitrairement petit, il existe un nombre fini  $T=T(M, \varepsilon)$  tel que si  $\|x_0 - x_e\| < M, t_0 \geq 0$  alors  $\|x(t) - x_e(t)\| < \varepsilon, \forall t \geq T=T(M, \varepsilon)$ .

### C-6. Stabilité exponentielle

Le point d'équilibre  $x_e$  est exponentiellement stable s'ils existent des constantes  $r, \alpha, \beta > 0$  tel que :

$$\|x(t) - x_e(t)\| \leq \alpha \|x_0\| \exp^{(-\beta t)}, \forall t \geq t_0, \forall x_0 \in B_r \quad (19)$$

### C-7. Stabilité exponentielle globale

$x_e$  globalement exponentiellement stable s'il existe des constantes  $\alpha, \beta > 0$  tel que :

$$\|x(t) - x_e(t)\| \leq \alpha \|x_0\| \exp^{(-\beta t)}, \forall t \geq t_0, \forall x_0 \in \mathfrak{R}^n \quad (20)$$

### C-8. Théorie de stabilité de Lyaponov

Cette théorie traite le comportement des systèmes non linéaires libres décrits par le système d'équations différentielles suivant :

$$\dot{x}(t) = f(t, x(t)) \quad (21)$$

$t \geq 0, x \in \mathfrak{R}^n$ , et  $f(., .) : \mathfrak{R}_+ \times \mathfrak{R}^n \rightarrow \mathfrak{R}$  est continue.

La méthode de Lyaponov est basée sur l'observation de l'énergie du système, en remarquant que si l'énergie du système décroît de manière continue en fonction du temps celui-ci évoluera jusqu'à atteindre un point d'équilibre stable. Le principe de la méthode s'appuie donc sur la construction d'une fonction « énergétique » généralisée, dite fonction de Lyaponov, possédant des propriétés spécifiques. Si l'énergie croît au-delà de sa valeur initiale, alors il est possible de conclure que le système est instable.

#### Stabilité

L'origine du système (21) est stable, s'il existe une fonction de classe  $C^1$  et localement définie positive  $V : \mathfrak{R}_+ \times \mathfrak{R}^n \rightarrow \mathfrak{R}$  et une constante  $r > 0$  telle que :

$$\dot{V}(t, x) \leq 0, \forall t \geq t_0 \text{ et } \forall x \in B_r$$

où  $\dot{V}$  est évaluée le long des trajectoires du système (21).

#### Stabilité uniforme

L'origine du système (21) est uniformément stable, s'il existe une fonction de  $C^1$  définie positive localement  $V : \mathfrak{R}_+ \times \mathfrak{R}^n \rightarrow \mathfrak{R}$  et une constante  $r > 0$  telle que :

$$\dot{V}(t, x) \leq 0, \forall t \geq 0 \text{ et } \forall x \in B_r, B_r = \mathfrak{R}^n$$

### Stabilité asymptotique

L'origine du système (21) est asymptotiquement stable, s'il existe une fonction scalaire  $V : \mathfrak{R}_+ \times \mathfrak{R}^n \rightarrow \mathfrak{R}$  telle que :

1.  $V(t, x)$  est définie positive ;
2.  $\dot{V}(t, x)$  est définie négative.

En plus, si  $V(t, x)$  est décroissante, alors l'origine est uniformément asymptotiquement stable.

### Stabilité asymptotique uniforme globale

L'origine du système (21) est asymptotiquement uniformément globalement stable, s'il existe une fonction de classe  $C^1$   $V : \mathfrak{R}_+ \times \mathfrak{R}^n \rightarrow \mathfrak{R}$  telle que :

1.  $V(t, x)$  est définie positive ;
2.  $V(t, x)$  est décroissante et radialement non bornée ;
3.  $\dot{V}(t, x)$  est définie négative.

### Stabilité exponentielle

L'origine du système (21) est exponentiellement stable, s'il existe des constantes  $\alpha, \beta, \eta$  et  $r > 0, p \geq 1$ , et une fonction de classe  $C^1$   $V : \mathfrak{R}_+ \times \mathfrak{R}^n \rightarrow \mathfrak{R}$  telle que :

$$\alpha \|x\|^p \leq V(t, x) \leq \beta \|x\|^p \text{ et } \dot{V}(t, x) \leq -\eta \|x\|^p, \forall t \geq 0, \forall x \in B_r$$

### Stabilité exponentielle uniforme globale

L'origine du système (21) est globalement uniformément exponentiellement stable, s'il existe des constantes  $\alpha, \beta, \eta$  et  $r > 0, p \geq 1$ , et une fonction de classe  $C^1$   $V : \mathfrak{R}_+ \times \mathfrak{R}^n \rightarrow \mathfrak{R}$  telle que :

$$\alpha \|x\|^p \leq V(t, x) \leq \beta \|x\|^p \text{ et } \dot{V}(t, x) \leq -\eta \|x\|^p, \forall t \geq 0, \forall x \in \mathfrak{R}^n$$

### Remarque

Si la fonction  $V$  est semi définie négative, le théorème de Lyapunov ne permet pas de démontrer la stabilité asymptotique de celle-ci. Il est alors possible d'utiliser le théorème de La Salle sur les trajectoires invariantes des systèmes dynamiques. Pour un complément de démonstration on peut utiliser aussi le Lemme de Barbalat.

La méthode de Lyapunov présente l'avantage de s'appuyer sur des propriétés physiques du système. Elle est cependant délicate à mettre en oeuvre en raison du choix « intuitif » de la fonction d'énergie (fonction de Lyapunov candidate).

### Théorème de La Salle [21]

Supposons  $V$  une fonction de Lyapunov définie positive vérifiant  $\lim_{\|x\| \rightarrow \infty} \|V(x)\| \rightarrow \infty$  et  $\dot{V}(x) \leq 0$  pour tout  $x \in \mathfrak{R}^n$ . Soit  $D = \{x \in \mathfrak{R}^n / \dot{V}(x) = 0\}$  et supposons que la seule trajectoire

continue dans  $D$  soit la trajectoire triviale, alors le point d'équilibre  $x=0$  est globalement asymptotiquement stable.

### Lemme de Barbalat [21]

Soit  $f(t)$  une fonction dérivable, admet une limite finie à l'infini (c-à-d :  $\lim_{t \rightarrow \infty} f(t) < \infty$ ) et si

$\dot{f}(t) = \frac{df}{dt}$  est uniformément continue (c-à-d :  $\ddot{f}(t) = \frac{d^2 f(t)}{dt^2}$  est bornée), alors  $\lim_{t \rightarrow \infty} \dot{f}(t) \rightarrow 0$ .

Autrement dit, si  $f(t) \geq 0$  et  $\dot{f}(t)$  est bornée, alors  $\lim_{t \rightarrow \infty} \dot{f}(t) = 0$ .

## D. Notions sur le réseau informatique

### D-1. Définition

Un réseau informatique (Network) est un ensemble des moyens matériels et logiciels mis en œuvre pour assurer les communications entre ordinateurs, stations de travail et terminaux informatiques interconnectés. Il permet de faire circuler des données informatiques et ainsi d'échanger du texte, des images, de la vidéo ou du son entre chaque équipement selon des règles et protocoles bien définies.

### D-2. Le modèle de référence OSI

La norme OSI (*Open System Interconnection*), En français «Interconnexion de systèmes ouverts», est un modèle de référence (théorique) proposé par l'ISO (*International Standardisation Organisation*) qui décrit les normes et les protocoles (règles) utilisés pour assurer le transfert des données d'un ordinateur à l'autre d'une façon fiable.

La figure D-1, présente l'architecture de ce modèle. Nous notons que les couches qui ont besoin d'échanger les données, entre la Machine A et la Machine B, correspondent aux couches applications (couche 7). Les autres couches correspondent aux différentes procédures et protocoles pour assurer cet échange.

Ce modèle peut être décomposé en trois couches :

#### Couches basses

Regroupe l'ensemble des couches 1, 2 et 3. Elles assurent la transmission et l'acheminement des informations à travers le réseau sur le support de transmission. Le rôle de chaque couche est :

- **Couche 1 (physique) :** Responsable de la transmission du signal (les éléments binaires constitutifs des trames) sur le médium de communication (Fibre optique, Paire torsadée, Câble coaxial...).
- **Couche 2 (liaison) :** Appelée Media Access Control (MAC) : mise en forme de l'information binaire et assure également le contrôle de flux au niveau des trames (Ethernet, Token Ring).
- **Couche 3 (réseau) :** Pour trouver le meilleur chemin entre la source et la destination des données groupées en paquets au travers le réseau (voir le protocole IP, IPv4, IPv6, IPx, ...).

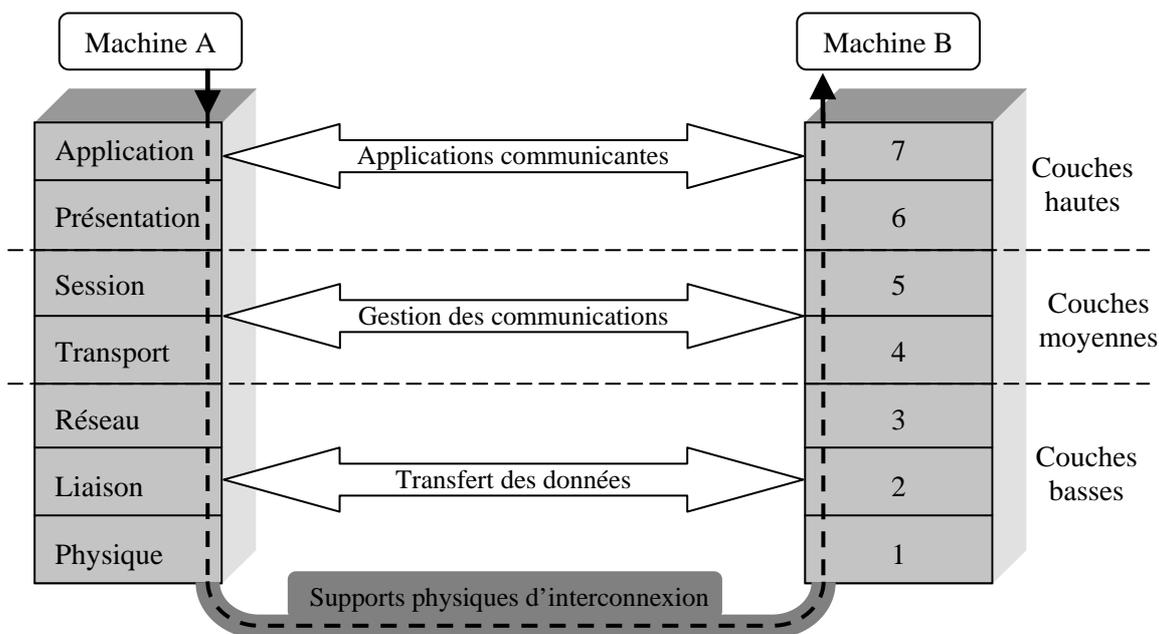


Figure D-1: Principe du modèle OSI

### Couches moyennes

Regroupe les couches 4 et 5. Elles gèrent les communications et les ressources nécessaires à l'échange des messages entre les équipements terminaux, indépendamment du ou des réseaux utilisés. L'objectif de chaque couche dans cet ensemble est :

- **Couche 4 (transport) :** Assure le transport des segments de l'information de la source vers la destination (bout-en-bout). Elle réalise aussi le contrôle de flux (TCP, UDP).
- **Couche 5 (session) :** Assure la communication entre le point de départ et le point d'arrivée. Elle sert d'interface entre les fonctions liées à l'application et celles liées au transport des données (Synchronisation du dialogue entre les différentes machines).

### Couches hautes

Regroupe les couches 6 et 7. Elles traitent les données échangées. L'objectif de cet ensemble de couches est :

- **Couche 6 (présentation) :** Est la couche responsable de la mise en forme des données avant leur fourniture aux applications (fonction de compression, d'encryptage, ...).
- **Couche 7 (application) :** Responsable de la communication entre les applications distantes impliquées dans l'échange de données, elle est aussi le point de contact entre l'utilisateur et le réseau (page web (http), messagerie (Simple mail Transfer Protocol (Smt)), ...).

### D-3. Constituants matériels d'un réseau

Les principaux équipements matériels mis en place dans les réseaux informatique sont illustrés sur la figure D-2.

La signification des différents éléments de la figure D-2 est :

### a) Serveur

Un serveur est généralement un ordinateur plus puissant qu'un ordinateur de bureau habituel. Il est spécialement conçu pour fournir des informations et des logiciels à d'autres ordinateurs qui lui sont reliés via un réseau. Les serveurs sont dotés de composants matériels qui gèrent la mise en réseau par câble Ethernet ou sans fil, généralement via un routeur. Capables de traiter des charges de travail plus importantes et d'exécuter davantage d'applications, les serveurs tirent parti de leurs composants matériels spécifiques pour augmenter la productivité et réduire les temps d'inactivité. Les serveurs offrent également des outils de gestion à distance (tâche de maintenance, de diagnostic des problèmes réseaux, ...).

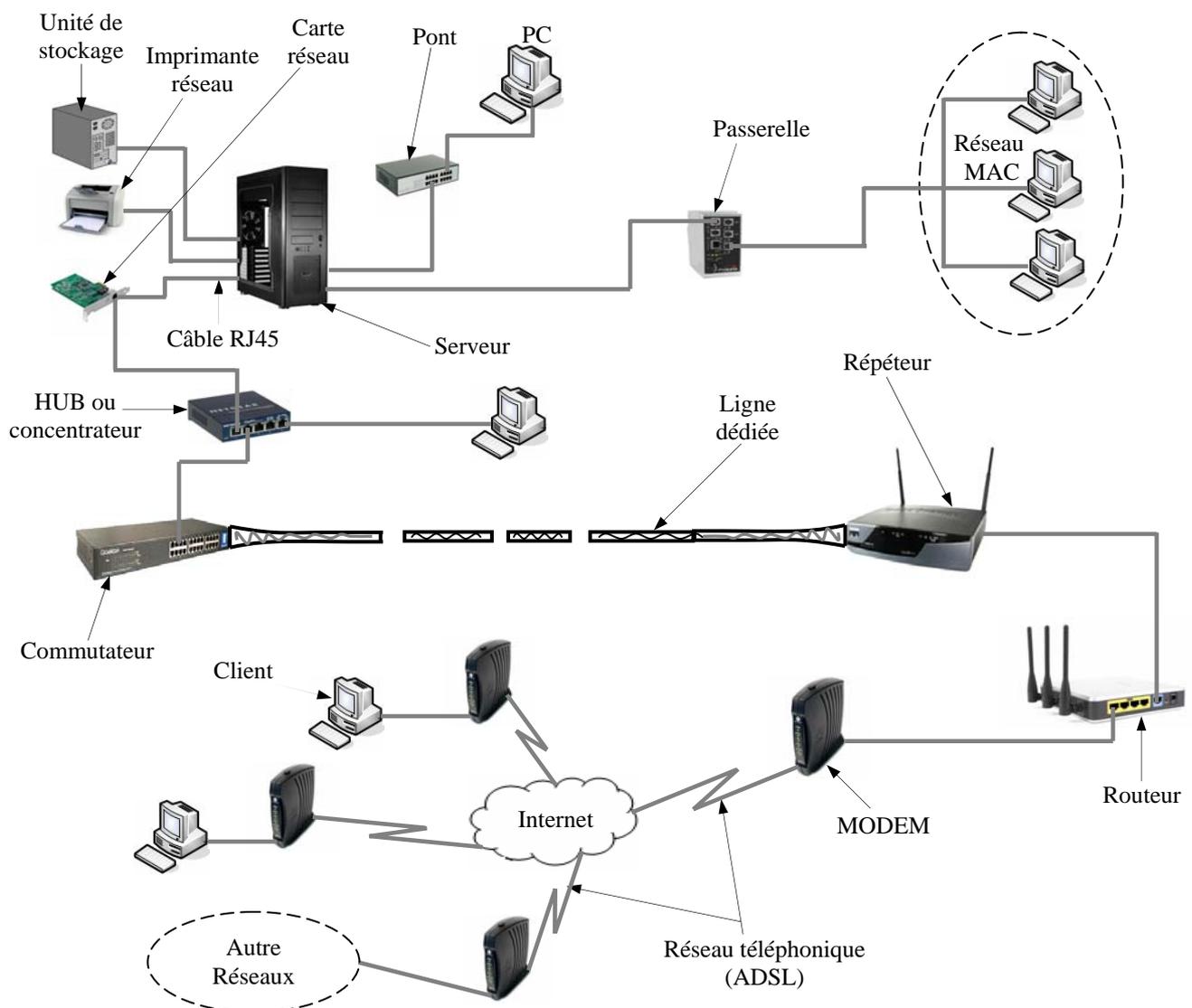


Figure D-2: Matériels et supports de transmission d'un réseau.

### b) Serveur de stockage (Unité de stockage)

Un serveur de stockage permet de stocker et surtout protéger tous ses fichiers: données, vidéos, films, musiques, photos ... . En réalité, ce type de serveur a souvent une grande capacité de stockage sur son espace disque. Ainsi, n'importe quel utilisateur connecté au réseau peut récupérer un

---

ou plusieurs fichiers grâce à un protocole (par exemples : FTP (*File Transfert Protocol*), CIFS (*Common Internet File System*), ...).

#### **c) Carte réseau**

La carte réseau constitue l'interface physique entre l'ordinateur et le support de communication. Pour qu'un ordinateur soit mis en réseau, il doit être muni d'une carte réseau.

#### **d) Serveur d'impression (Imprimante réseau)**

Un serveur d'impression est un serveur qui permet de partager une imprimante entre plusieurs utilisateurs situé sur un même réseau informatique. Il peut être constitué d'un ordinateur qui partage une imprimante qui lui est directement connectée, ou peut être un petit appareil spécialisé (Boîtier serveur d'impression avec ou sans fil). L'avantage de cette dernière solution est son faible prix. Bien entendu, un serveur d'impression doit toujours rester sous tension et il est préférable qu'il ait une adresse IP fixe.

#### **e) Pont (bridge en anglais)**

Est un équipement qui sert habituellement à faire transiter des paquets entre deux réseaux de même type (utilisant le même protocole). Quand il reçoit le paquet (la trame), il est en mesure d'identifier l'émetteur et le récepteur; comme ça il dirige la trame directement vers la machine destinataire (entre le réseau côté PC et le réseau côté serveur sur la figure D-2).

#### **f) Passerelle (Gateway)**

Est un système matériel et logiciel permettant de faire la liaison entre deux réseaux de types différents (utilisant deux protocoles et/ou architectures différentes); comme par exemple un réseau local et Internet. Lorsque un utilisateur distant contact un tel dispositif, celui-ci examine sa requête, et si celle-ci correspond aux règles que l'administrateur réseaux a défini, la passerelle crée un pont entre les deux réseaux. Les informations ne sont pas directement transmises, elles sont plutôt traduites pour assurer la transmission tout en respectant les deux protocoles (exemple entre réseau MAC et le réseau côté serveur sur la figure D-2).

#### **g) Concentrateur (Hub en anglais)**

Est un appareil informatique permettant de créer un réseau informatique local de type Ethernet. Il permet de concentrer le trafic réseau provenant de plusieurs hôtes (machines), et de régénérer le signal. Il est ainsi une entité possédant un certain nombre de ports (il possède autant de ports qu'il peut connecter de machines entre elles, généralement 4, 8, 16 ou 32). Son unique but est de récupérer les données binaires parvenant sur un port et de les diffuser sur l'ensemble des ports. Il opère au niveau 1 du modèle OSI, c'est la raison pour laquelle il est parfois appelé *répéteur multiports*.

Le concentrateur permet ainsi de connecter plusieurs machines entre elles, parfois disposées en étoile, ce qui lui vaut le nom de **Hub** (signifiant *moyeu de roue* en anglais; la traduction française

exacte est *répartiteur*), pour illustrer le fait qu'il s'agit du point de passage des communications des différentes machines.

#### **h) Commutateur (Switch en anglais)**

Est un équipement qui relie plusieurs segments (câbles ou fibres) dans un réseau informatique. Il s'agit le plus souvent d'un boîtier disposant de plusieurs ports Ethernet (entre 4 et plusieurs centaines). Il a donc la même apparence qu'un concentrateur (Hub). Sa seule différence avec celui-ci, c'est sa capacité de connaître l'adresse physique des machines qui lui sont connectés et d'analyser les paquets reçus pour les diriger vers la machine de destination. Il s'agit d'un élément actif agissant au niveau 2 du modèle OSI.

#### **i) Répéteur (*Repeater* en anglais)**

Est un équipement simple permettant de régénérer un signal entre deux nœuds du réseau, afin d'étendre la distance de câblage d'un réseau (amplification d'un signal). Il est à noter qu'on peut utiliser un répéteur pour relier deux supports de transmission de type différents. Le répéteur travaille uniquement au niveau physique (couche 1 du modèle OSI), c'est-à-dire qu'il ne travaille qu'au niveau des informations binaires circulant sur la ligne de transmission et qu'il n'est pas capable d'interpréter les paquets d'informations.

#### **j) Routeur**

Est un équipement d'interconnexion de réseaux informatiques permettant d'assurer le routage des paquets entre deux réseaux ou plus afin de déterminer le chemin qu'un paquet de données va emprunter. Ce routage est réalisé selon un ensemble de règles formant la table de routage pour trouver le chemin optimal que va emprunter le message depuis l'émetteur vers le récepteur. C'est un équipement de couche 3 par rapport au modèle OSI. Il ne doit pas être confondu avec un commutateur (couche 2) !

#### **k) Modem**

Signifie Modulateur/Démodulateur. Est un équipement qui sert à lier le réseau téléphonique au réseau informatique. Souvent pour transmettre des données informatiques à distance, on utilise la ligne téléphonique comme support de transmission. Et comme nous savons que la ligne téléphonique ne transporte que des signaux analogiques et que les réseaux informatiques n'utilisent que des signaux numériques, le modem a pour rôle de convertir le signal numérique en signal analogique (modulation) et vis versa (démodulation). Le modem permet par exemple de se connecter à Internet.

Il est à noter que la plupart des ordinateurs sont munis des modems intégrés.

#### **l) Supports de communication**

L'interconnexion entre les différents composants d'un réseau ce fait par l'intermédiaire de différents canaux de communication appelés supports de transmission. Les principaux supports

utilisés sont : supports filaires (le câble coaxial, la paire torsadée, la fibre optique), les supports sans fil (WiFi, Infrarouge, Ondes radio, Bluetooth) ou par satellites, ... .

#### D-4. Différents types de réseaux

On distingue différents types de réseaux selon leur taille (en terme de nombre de machines), leur vitesse de transfert des données ainsi que leur étendue.

On fait généralement trois catégories de réseaux :

##### a) LAN (Local Area Network en français Réseau Local)

Il s'agit d'un ensemble d'ordinateurs appartenant à une même organisation et reliés entre eux dans une petite aire géographique par un réseau, souvent à l'aide d'une même technologie (la plus répandue étant Ethernet).

##### b) MAN (Metropolitan Area Network)

Les MAN interconnectent plusieurs LAN géographiquement proches (au maximum quelques dizaines de km) à des débits importants. Ainsi un MAN permet à deux nœuds distants de communiquer comme si ils faisaient partie d'un même réseau local.

Un MAN est formé de commutateurs ou de routeurs interconnectés par des liens hauts débits (en général en fibre optique).

##### c) WAN (Wide Area Network ou réseau étendu)

Un WAN interconnecte plusieurs LANs à travers de grandes distances géographiques. Les débits disponibles sur un WAN résultent d'un arbitrage avec le coût des liaisons (qui augmente avec la distance) et peuvent être faibles.

Les WAN fonctionnent grâce à des routeurs qui permettent de "choisir" le trajet le plus approprié pour atteindre un nœud du réseau. Le plus connu des WAN est Internet.

Le tableau D-1 présente une simple comparaison entre ces trois réseaux :

	LAN	MAN	WAN
La portée	Quelques Km	75 Km	L'interconnexion à l'échelle de la planète
Débit	10 Mbits/s à 1000 Mbits/s	200 Mbits/s	va de quelques Kbits/s à quelques Mbits/s

**Tableau D-1:** Comparaison des trois réseaux (LAN, MAN, WAN)

#### D-5. Modèle de référence de l'Internet

Dans le monde d'aujourd'hui, l'Internet représente un grand outil de communication, cet interréseau, utilisant les mêmes protocoles de routage et de transport TCP/IP, se prête assez bien aux problèmes impliquant un contrôle à distance (par exemple : la télérobotique, la téléopération, la télémaintenance, le téléenseignement, la téléformation, le télédiagnostic, la télé médecine, ... etc.).

La naissance de l'Internet date du début des années 80 aux Etats-Unis, au moment où le DARPA (Defense Advanced Research Projects Agency) créa la série de protocoles TCP/IP pour ses réseaux de recherche Arpanet et militaire Milnet.

Les protocoles TCP/IP permet d'interconnecter tout type de réseau informatique et s'adaptent à chaque architecture réseau sur laquelle ils viennent se greffer et donnent ainsi l'illusion d'un réseau homogène. TCP/IP Impose un traitement supplémentaire pour assurer le bon transport des données.

### D-5-1. Architecture TCP/IP (Transmission Control Protocol/Interconnection Protocol)

Le modèle TCP/IP (ou *pile de protocoles TCP/IP*), parfois appelé modèle de référence pour l'interconnexion. Contient cinq couches, comme l'illustre la figure D-3.

Pour désigner les informations transmises et leur enveloppe, selon le niveau concerné, on parle de **message**(ou de flux) entre applications, de **datagramme** (ou segment) au niveau TCP, de **paquet** au niveau IP, et enfin, de **frames** au niveau de l'interface réseau (Ethernet ou Token Ring).

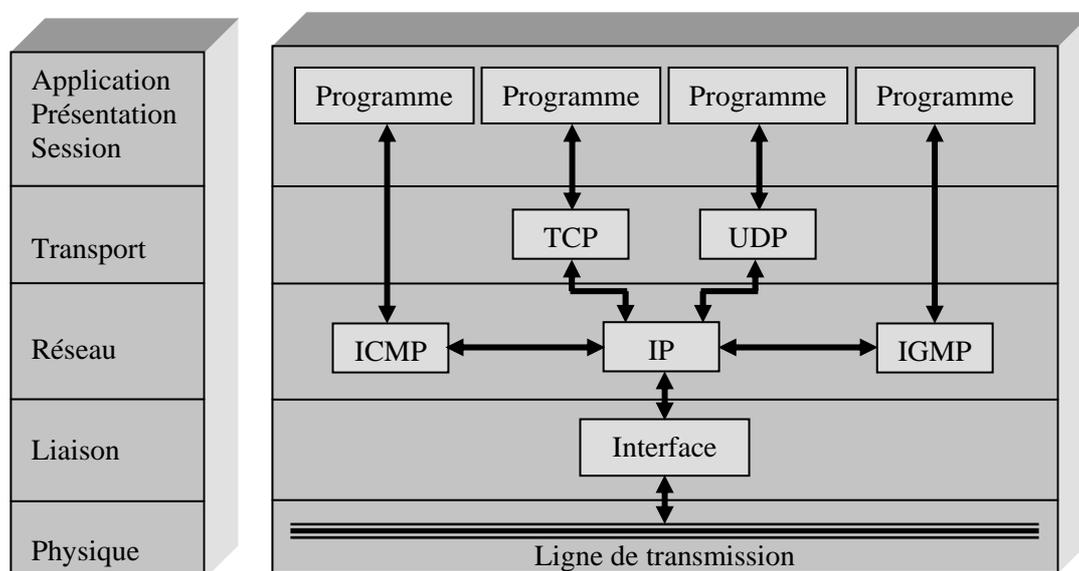


Figure D-3 : Comparaison entre l'architecture OSI et TCP/IP

TCP/IP combine plusieurs couches du modèle OSI en une couche unique et n'utilise pas certaines couches (OSI ne possède pas la couche d'interréseau).

Les couches 1 et 2 (ligne de transmission et l'Interface) correspondent aux couches 1 et 2 du modèle OSI. Au niveau de la couche 3 (interréseau) se trouve l'implantation du protocole IP (Internet Protocol). Ce protocole, en mode datagramme, offre des fonctions d'adressage et de routage (traitement du format des paquets envoyés à travers un interréseau et les mécanismes de propagation de ces paquets entre machines via les routeurs). Le protocole IP gère également la fragmentation des données. Cette couche contient aussi d'autres protocoles par exemple ICMP (*Internet Control Message Protocol*), qui permet de gérer les messages de contrôle et d'erreur, lors de l'émission des messages, que ne fournit pas IP (réseau coupé, échéance temporelles, ...) et IGMP (*Internet Group*

*Management Protocol*) qui est utilisé pour effectuer une communication multicast (plusieurs destinataires - groupe) dans un réseau IP. ICMP et IGMP sont basés sur IP.

Au niveau de la couche 4 se trouve le protocole TCP (transmission Control Protocol) qui offre aux utilisateurs un transfert fiable sur connexion et le protocole UDP (User Datagram Protocol) qui offre un transfert en mode datagramme.

La couche 5 regroupe les couches 5, 6 et 7 du modèle OSI. Les protocoles de cette couche indiquent comment une application particulière utilise l'interréseau (*HTTP, SNMP, SMTP, FTP, ...*).

Les principaux protocoles de la couche transport sont :

#### **D-5-1-1. Protocole TCP**

Le protocole de niveau transport TCP (Transmission Control Protocol), présente dans les équipements d'extrémités, est responsable du transfert des paquets de bout en bout indépendamment des couches inférieures ou équipements intermédiaires (routeurs, commutateurs, ...), le contrôle se faisant à partir des équipements d'extrémités. C'est TCP qui apporte la fiabilité sur Internet en veillant à ce que tous les paquets arrivent à destination et soient retransmis en cas de perte ou de saturation d'un routeur.

Ce protocole recouvre globalement les fonctionnalités des communications de niveau de transport avec connexion. Ses principales caractéristiques sont : établissement d'une connexion duplex et contrôle de cette connexion ; segmentation et réassemblage des données ; acquittement des datagrammes reçus et retransmission sur absence d'acquittement (un reséquencement est effectué si la couche IP ne le délivre pas dans l'ordre) ; contrôle de flux, gestion des priorités des données et de la sécurité de la communication.

#### **D-5-1-2. Protocole UDP**

UDP (*User Datagram Protocol*) est un complément du protocole TCP qui offre un service de datagrammes sans connexion et qui fonctionne au dessus du protocole de réseau IP. Il permet à une application d'envoyer des messages (paquets UDP) à une autre application avec un minimum de fonctionnalités et d'une manière autonome (pas de garanties d'arrivée, ni de contrôle de séquencement). Il n'apporte pas de fonctionnalités supplémentaires par rapport à IP et permet simplement de désigner les numéros de port correspondant aux applications envisagées.

Le protocole UDP prend également en charge l'envoi de données d'un unique expéditeur vers plusieurs destinataires.

#### **D-5-3. Définition d'un socket**

Un socket est un point de communication, homogène avec les identificateurs d'E/S, qui est utilisé pour établir un canal de communication permettant l'envoi et la réception d'informations entre des processus qui s'exécutent dans un environnement distribué.

L'interface *Berkeley Socket* ou *API* (*Application Program Interface*) a été développée et introduite au début des années 80 à l'université de Californie à Berkeley et permet une communication interprocessus dans l'environnement Unix. Elle est conçue pour être portable sur

n'importe quel type de système d'exploitation. Il s'agit d'une interface souple, d'assez bas niveau (couches 4 et 3). L'interface *Winsock* a été développée récemment et permet une communication interprocessus dans l'environnement Windows. Un socket peut être utilisé soit pour définir un client ou un serveur.

#### **D-5-4. Modèle de communication Client-Serveur**

Les programmes d'application qui communiquent, entre deux ou plusieurs machines, à travers un réseau ou un interréseau utilisent tous la même forme d'interaction qui est connue sous le nom de modèle client-serveur.

Le serveur est un programme exécuté sur une des machines, qui attend (de manière passive) les connexions de machines clientes. Dès qu'une ou plusieurs machines tentent de se connecter (d'une façon active) au serveur, ce dernier leur ouvre une session de communication qui sera fermée à l'initiative du client ou du serveur selon le cas. Un serveur qui n'accepte qu'une seule connexion simultanée est appelé : serveur itératif.

La programmation d'un client ou d'un serveur à l'aide des sockets prend en considération : le type du protocole utilisé (TCP, UDP, ...), l'adresse IP de la machine distante et le port d'écoute de connexions.

##### **D-5-4-1. Adresses IP**

Chaque ordinateur connecté à un réseau internet possède une adresse IP unique. Le moyen utilisé pour s'adresser à un serveur particulier est de connaître son *adresse IP*, un numéro que l'on peut comparer à un numéro de téléphone.

Il existe deux versions d'adresses IP : Une adresse codée sur 32 bits (version IPv4), se note généralement sous la forme de 4 nombres de 0 à 255 (notation décimale), séparés par des points (ex : 192.163.223.01) et une adresse codée sur 128 bits (version IPv6 pour augmenter le nombre d'adresse de manière à accompagner la croissance de l'Internet) utilisant une notations hexadécimale à deux point (ex : 805B:2D9D:0000:DC28:12F7:000A:765C:D4C8).

Pour chaque connexion à internet, le fournisseur d'accès internet (FAI), vous fournit un accès : il vous branche au réseau en vous attribuant une adresse IP dite *dynamique* parce qu'elle sera différente à la prochaine connexion. On attribue une adresse IP dite *fixe* aux serveurs parce qu'elle ne change pas; le serveur reste branché au réseau pour offrir ses services, on doit pouvoir le localiser de manière systématique.

##### **D-5-4-2. Notion de ports**

Pour qu'une machine puisse rendre divers services sous forme de serveurs (résolution de noms, accès distant, ...), chaque serveur écoute les connexions sur un port donné (point d'accès) qui représente un nombre codé sur 2 octets (16 bits).

Les ports sont des numéros allant de 1 à 65535 utilisables pour identifier l'application sur une machine. Les applications ou services habituels ont des ports fixés (Les ports 1 à 1023) : echo (7),

FTP (21), Telnet (23), Sntp (25), Dns (53), http (80), pop3 (110), etc. D'autres numéros de port sont disponibles pour les applications développées par les utilisateurs (1024 à 65535) qui représentent les ports dynamiques (ou privés). Ceux-ci n'ont Rien à avoir avec les ports physiques (série et parallèle) et n'existent que dans la mémoire de la machine.

Il est possible de faire une analogie avec une adresse postale : l'adresse IP correspond au nom de la rue et le numéro de port au numéro dans la rue.

Deux références importantes pour connaître plus sur les réseaux informatiques et les différents protocoles liés à l'Internet : [1] Stéphane Lohier, Dominique Présent, *Cours : Transmissions et Réseaux*, Dunod, Paris, 2010 ; [2] Douglas E. Comer, *Réseaux et Internet*, CompusPress, Paris, 2000.