

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
Ministère de l'enseignement supérieur et de la recherche scientifique  
**République Algérienne Démocratique et Populaire**  
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique



**Université Batna 2 – Mostefa Ben Boulaïd**  
**Faculté de Technologie**  
**Département de Génie Industriel**



**Thèse**

Préparée au sein du Laboratoire d'Automatique et Productique (LAP) de l'Université  
Batna 2 – Mostefa Ben Boulaïd

Présentée pour l'obtention du diplôme de : **Doctorat LMD 3<sup>ème</sup> Cycle**  
**Option : Génie Industriel**

Sous le Thème :

**Supervision d'un robot manipulateur virtuel par les réseaux  
de neurones**

Présentée par :

**GHRIEB Abdel Ouahab**

**Devant le jury composé de :**

MOUSS Kinza Nadia	Prof.	Université de Batna 2	Président
KOURD Yahia	MCA	Université de Souk-Ahras	Rapporteur
MOUSS Mohamed Djamel	Prof.	Université de Batna 2	Co-Rapporteur
MELAKHESSOU Hayat	MCA	Université de Batna 2	Examineur 1
RAMDANI Messaoud	Prof.	Université de Annaba	Examineur 2

**Soutenue publiquement le 02 Juillet 2020**

---

## *Table des matières*

---

Résumé.....	I
Abstract.....	III
ملخص.....	IV
Remerciements.....	V
Dédicace .....	VI
Sommaire .....	VII
Liste des figures.....	X
Liste des tableaux.....	XII
Liste des abréviations.....	XIII
Introduction générale.....	01

---

## *Résumé*

---

Dans ce travail de thèse, nous avons proposé un système de supervision appliqué sur un robot manipulateur à deux degrés de liberté. La supervision est utilisée pour assurer la reconfiguration en temps réel du robot. Dans ce système nous avons utilisé une nouvelle méthode de détection de défaut (FD) de frottement visqueux du robot supervisé combinée avec un module de commande tolérante aux défauts (FTC). Le premier module, basé sur une méthode de traitement appliquée sur des résidus, va permettre la détection de défaut pour bien estimer les corrections nécessaires du deuxième module. Une évaluation de l'effet de défaut durant la supervision a été faite. Par ailleurs, le protocole TCP pour le transfert des données entre le robot superviseur et le robot supervisé a été utilisé. Les résultats de simulation montrent que la méthode proposée corrige l'effet de défaut en utilisant les données qui arrivent d'un robot superviseur à distance. Ensuite, nous avons proposé une implémentation matérielle sur cible FPGA de l'algorithme de supervision dont le but est de valider notre contribution et d'assurer un traitement en temps réel dans le cas où il y a des robots réels. Par ailleurs, une étude comparative entre les performances des deux implémentations a été effectuée.

**Mots-clés** -*Détection de défaut, Commande tolérante aux défauts, Défaut de frottement visqueux, robot superviseur, système de supervision, Implémentation matérielle.*

---

## *Abstract*

---

In this work of thesis, a supervision system applied on a robot manipulator with two degrees of freedom are proposed. The supervision system is used to ensure real-time reconfiguration of the robot. In this system we used a new fault detection of viscous friction fault in the supervised robot combined with a fault tolerance control method. The first module, based on analysis of residuals, is used to fault detection approach based on the Transmission Control Protocol for data transmission between the supervisor and the supervised robot. The simulation results show that the proposed method adjusts fault effect using information transferred from the remote supervisor robot. Moreover, we proposed a new FPGA hardware implementation of the supervision algorithm using system generator of Xilinx. This hardware implementation is used in order to validate the proposed contribution and to ensure real time processing in the case of industrial robots. The simulation results and the response times of both proposed systems are compared and discussed.

**Keywords**—*Fault Detection, Fault Tolerance Control, Viscous Friction Fault, Supervisor Robot, FPGA, Hardware Implementation.*

---

## ملخص

---

العمل المقدم في هذه الأطروحة يتضمن دراسة جديدة للإشراف على ذراع صناعي ذات 2 درجة حرية لضمان الأداء المطلوب في الوقت الحقيقي، وتم استخدام طريقة جديدة في اكتشاف الخطأ مع التحكم المناسب للخطأ لوجود خطأ الاحتكاك اللزج المتوقع حدوثه في الروبوت المشرف عليه، اكتشاف الخطأ يعتمد على تحليل البقايا من أجل تحسين أداء التحكم المناسب للخطأ بالإضافة الى ذلك تم استخدام تقنية بروتوكول تبادل البيانات على الانترنت (TCP) وذلك في نقل المعطيات بين الروبوتات.

المحاكات أظهرت نتيجة فعالة في تصحيح أثر الخطأ بالاعتماد على بيانات الروبوت المشرف عن بعد وبالإضافة إلى هذه الدراسة نقتراح التنفيذ على لوحة البوابات المنطقية القابلة للبرمجة FPGA لنظام الإشراف باستخدام نظام المحاكات Xilinx. يستخدم هذه اللوحة من أجل التحقق من صحة المساهمة المقترحة وضمان المعالجة في الوقت الحقيقي في حالة الروبوتات الصناعية. تمت مقارنة نتائج المحاكاة وأوقات الاستجابة لكل من الأنظمة المقترحة.

### الكلمات المفتاحية:

اكتشاف الخطأ، التحكم المناسب للخطأ، خطأ الاحتكاك اللزج، الروبوت المشرف، البوابات المنطقية القابلة للبرمجة (FPGA)

---

## *Remerciements*

---

Je tiens à remercier tout d'abord notre Dieu le tout-puissant pour la force et la volonté qui m'a procuré pour finaliser cette thèse de doctorat.

Ce travail a été effectué en collaboration avec le Laboratoire d'Automatique et Productique (LAP) de l'Université Batna2 Mostefa Ben Boulaïd et le Laboratoire de l'électrotechnique et énergies renouvelable (LEER) de l'université Mohamed-Chérif Messaadia de Souk-Ahras, sous la direction du Monsieur KOURD Yahia, maître de conférences classe A de l'université de Souk-Ahras et Monsieur MOUSS Mohamed Djamel, Professeur de l'université de Batna02 avec l'assistance du Monsieur MASSOUDI Kamel, maître de conférences classe A de l'université de Souk-Ahras et Monsieur GUERSI Noureddine, Professeur de l'université de ANNABA et Monsieur BAKIR Toufik, maître de conférences classe A de l'université de Bourgogne France.

Enfin, un grand merci à ma famille, spécialement mes parents, qui m'ont toujours soutenue tout au long de ma vie et merci à mes amis qui m'a soutenue pendant ces années de thèse.

---

## *Dédicace*

---

À mes chers parents.

À mes frères.

À mes SŒURS.

À toute ma famille.

À tous mes amis.

À tous ceux qui m'ont aidé à faire ce travail.

---

## *Sommaire*

---

### **Chapitre I : Introduction à la supervision**

1.1 Introduction .....	5
1.2 Quelques Notions .....	5
1.3 La supervision .....	6
1.3.1 La surveillance .....	8
1.3.1.1 Méthode de surveillance à base de modèle .....	8
1.3.1.2 Méthode de surveillance sans modèle .....	9
1.3.2 Diagnostic de défauts .....	10
1.3.2.1 Détection de défaut .....	12
1.3.2.2 Localisation de défaut .....	14
1.3.2.3 Identification de défaut .....	14
1.3.3 Commande tolérante aux défauts .....	14
1.4 Supervision des robots manipulateurs .....	15
1.5 Conclusion .....	18

### **Chapitre II : Modélisation et commande de robots manipulateurs**

2.1 Introduction .....	20
2.2 Modélisation des robots manipulateurs .....	20
2.3 Modèle géométrique .....	21
2.3.1 Modèle géométrique direct .....	22
2.3.2. Modèle géométrique inverse .....	26
2.4 Modèle cinématique .....	28
2.4.1 Modèle cinématique direct .....	29
2.4.2 Modèle cinématique inverse .....	30
2.5 Modèle dynamique .....	30
2.5.1 Modèle dynamique directe .....	31
2.5.2 Modèle dynamique inverse .....	32
2.6 Commande d'un robot manipulateur .....	38
2.6.1 Commande adaptative .....	38



2.6.2 Commande dynamique .....	40
2.6.3 Commande PD .....	40
2.6.4 Commande PID.....	42
2.7 Conclusion .....	43

### **Chapitre III : Les techniques de l'intelligence artificielle**

3.1 Introduction .....	45
3.2 Logique flou .....	45
3.2.1 Ensemble Flou .....	46
3.2.2 Fonctions d'appartenance .....	47
3.2.3 Variables linguistiques.....	47
3.2.4 Règles et opérateurs flous .....	48
3.2.5 Structure de base d'un système flou .....	49
3.2.5.1 La base de connaissance .....	49
3.2.5.2 Fuzzification .....	49
3.2.5.3 Inférence .....	50
3.2.5.4 Défuzzification .....	51
3.3 Réseaux de neurones .....	52
3.3.1 Architecture des réseaux de neurones.....	53
3.3.1.1 Les réseaux de neurones non bouclés (feed forward).....	53
3.3.1.2 Les réseaux de neurones bouclés (récurrents) .....	55
3.3.2 Apprentissage dans les réseaux de neurones.....	55
3.3.2.1 L'apprentissage supervisé .....	56
3.3.2.2 L'apprentissage par renforcement .....	56
3.3.2.3 L'apprentissage non supervisé .....	56
3.4 Système hybride Neuro-Flou .....	56
3.4.1 Définition .....	57
3.4.2 Système ANFIS .....	57
3.4.2.1 Couche 1 .....	58
3.4.2.2 Couche 2 .....	58
3.4.2.2 Couche 3 .....	58
3.4.2.2 Couche 4 .....	58

3.4.2.2 Couche 5 .....	59
3.4.3 Supervision industriel à basée de l'intelligence artificielle .....	59
3.5 Conclusion .....	60
<b>Chapitre IV Simulation et Résultat</b>	
4.1 Introduction .....	62
4.2 Communication TCP IP.....	62
4.2.1 protocoles TCP .....	63
4.2.1.1 Couche hôte réseau .....	63
4.2 1.2. Couche Internet .....	63
4.2.1.3 Couche Transport .....	64
4.2.1.4 Couche Application .....	64
4.3 Implémentation Matérielle sur cible FPGA .....	65
4.3.1 System-Generator de chez Xilinx .....	65
4.3.2 Implémentation Matérielle .....	66
4.4 Schémas de système global .....	66
4.4.1 La Supervision d'un robot manipulateur .....	67
4.4.1.1 La détection de défaut.....	67
4.4.1.2 Commande tolérante aux défauts .....	69
4.5 Résultat de simulation et discussions .....	70
4.5.1 Résultat de défaut de la première articulation .....	72
4.5.2 Résultat de défaut de la deuxième articulation .....	75
4.5.3 Résultat de synthèse de défaut en première articulation .....	79
4.6 Conclusion .....	84
<b>Conclusion générale et Perspectives.....</b>	<b>86</b>
<b>Références bibliographiques .....</b>	<b>88</b>

---

## *Liste des figures*

---

### **Chapitre I :**

Figure I.1 : Structure d'un système avec une procédure de supervision.....	7
Figure I.2 : Méthode à base de modèle et sans modèle de surveillance.....	10
Figure I.3 : principe de fonctionnement et les étapes de diagnostic .....	12
Figure I.4 : Commande tolérante aux fautes .....	15

### **Chapitre II :**

Figure II.1 : Robot manipulateur à deux degrés de liberté .....	21
Figure II.2 : Paramètres de Denavit et Hartenberg .....	23
Figure II.3 : Les articulation ( $q_2$ et $q_1$ ) d'un bras à deux degrés de liberté .....	27
Figure II.4 : Structure de robot à deux degrés de liberté (2ddl) .....	33
Figure II.5 : Commande adaptative avec modèle de référence .....	39
Figure II.6 : Commande auto-adaptative .....	39
Figure II.7 : La commande dynamique avec PD .....	41
Figure II.8 : La commande dynamique avec PID .....	42

### **Chapitre III :**

Figure III.1 : Fonctions d'appartenances.....	46
Figure III.2 : Exemple de fonctions d'appartenance .....	47
Figure III.3 : les variables linguistiques .....	48
Figure III.4 : Structure de base de système flou .....	49
Figure III.5 : Exemple de fuzzification .....	50
Figure III.6 : Compositions des ensembles flous issus de l'inférence .....	51
Figure III.7 : Architecture réseaux de neurones .....	53
Figure III.8 : Architecture générale d'un réseau multicouche .....	55
Figure III.9 : Schéma de système neuro-flou .....	57
Figure III.10 : Les différentes couches d'ANFIS .....	59

## Chapitre IV :

Figure IV.1 : Architecture de protocole TCP .....	64
Figure IV.2 : L'interconnexion de système globale par protocole TCP .....	64
Figure IV.3 : Schémas de système globale .....	67
Figure IV.4 : Diagramme de base du système global proposé sous Matlab -Simulink .....	71
Figure IV.5 : La fonction de Résidu F1 .....	72
Figure IV.6 : Résidu de première articulation .....	73
Figure IV.7 Résidu de deuxième articulation .....	73
Figure IV.8 : Résidu de fonction F2 .....	74
Figure IV.9 : Résidu de première articulation .....	74
Figure IV.10 : Résidu de deuxième articulation .....	75
Figure IV.11 : La fonction de Résidu F1 .....	75
Figure IV.12 : Résidu de première articulation .....	76
Figure IV.13 : Résidu de deuxième articulation .....	76
Figure IV.14 : La fonction de Résidu F2 .....	77
Figure IV.15 : Résidu de première articulation .....	77
Figure IV.16 : Résidu de deuxième articulation .....	78
Figure IV.17 : Implantation du superviseur propose en utilisant les blocs Xilinx .....	79
Figure IV.18 : La fonction de Résidu F1 .....	80
Figure IV.19 : Résidu de première articulation .....	81
Figure IV.20 : Résidu de deuxième articulation .....	81
Figure IV.21 : La fonction de Résidu F2 .....	82
Figure IV.22 : Résidu de première articulation .....	82
Figure IV.23 : Résidu de deuxième articulation .....	83
Figure IV.24 : Schémas RTL des deux fonctions de résidus F1 et F2 .....	84
Figure IV.25 : Schémas RTL des deux fonctions de résidus F1 et F2 .....	84

---

## *Liste des tableaux*

---

### **Chapitre I:**

Tableau II.1 : Les paramètres de Denavit Hartenberg du robot à 2ddl ..... 24

### **Chapitre IV :**

Tableau IV.1 : Les paramètres de modèle ANFIS 1 ..... 69

Tableau IV.2 : Les paramètres de modèle ANFIS 2 ..... 70

---

## *Abréviations*

---

ANFIS : Adaptive Neuro Fuzzy Inference System : système d'inférence adaptive neuro-floue.

DDL : Degré De Liberté.

FD: Fault Detection : Détection de défaut.

FTC : Fault Tolerance Control : Contrôle tolérance de défaut.

FPGA : Field-Programmable Gate Array.

MCD : Modèle Cinématique Direct.

MCI : Modèle Cinématique Inverse.

MDD : Modèle Dynamique Direct.

MDI : Modèle Dynamique Inverse.

MGI : Modèle Géométrique Inverse.

MGD : Modèle Géométrique Direct.

LF : Logique Floue.

RNA : Les réseaux de neurones artificiels.

PD : Proportionnel Dérivateur.

PID : Proportionnel Intégrateur Dérivateur.

TCP : Transmission Protocol Control.

---

## *Introduction générale*

---

Aujourd'hui, l'évolution des processus industriels est marquée par le nombre de robots utilisés. Ces robots sont utilisés principalement pour l'augmentation du volume de production et pour l'amélioration de la qualité de ce dernier. Les robots manipulateurs sont utilisés dans des applications vastes et variées, plus particulièrement dans les domaines de médecine et de l'industrie. Dans ces domaines et avec l'augmentation des contraintes diverses, les sujets concernant la précision et la rapidité des robots sont devenus très courants et d'actualité. En-effet, les effets de défaut affectent directement ces deux caractéristiques. Cette situation rend les tâches de diagnostic, de surveillance et de supervision des robots très importantes et nécessaires.

Ainsi, les exigences de l'opérateur des robots industriels contribuent à l'utilisation du système évolue de diagnostic, de surveillance ou de supervision pour donner une évaluation précise du fonctionnement de ces robots. D'autre part le développement de mécanisme qui permet d'indiquer les défauts dans le système sera nécessaire avec en cas de présence plusieurs des défauts non observables. La méthodologie de supervision de principe consiste généralement à générer les informations en cas de fonctionnement anormal de système industriel.

Les robots manipulateurs sont caractérisés par le nombre de degrés de liberté, et connus comme des systèmes complexes et non linéaires, ces contraintes rendent leur supervision difficile. Certaines approches ont été proposées par des chercheurs qui se sont penchés sur cette difficulté. Nous citons les techniques d'intelligence artificielle dans plusieurs travaux existents, par exemple la logique floue, les réseaux neurones et les algorithmes génétiques. Bien que par ces outils la tâche de supervision a été effectuée avec une certaine réussite. Suite d'un autre point de vue, la communication informatique sera importante pour l'utilisation par l'opérateur humain avec l'extension d'utilisation des robots industriels. Il y'a plusieurs types de communication, chaque type est caractérisé par un protocole de communication. L'un des moyens de communication le plus utilisable est la communication de protocole TCP. Il faut intégrer des techniques pour la perfection de supervision industrielle. La disposition de travaux actuelle est de trouver une connaissance précise sur les natures et les causes de mauvais fonctionnement du procédé, et l'implémentation des outils principaux pour donner un système plus d'efficacité. Par la suite, l'objectif de notre étude est le développement et l'application d'un système de supervision basé sur des outils modernes,

notamment les techniques d'intelligence artificielle, une méthode de communication informatique de protocole TCP. Le système de supervision capable d'éviter le mauvais fonctionnement à cause de défauts qui apparaissent pendant la réalisation de certaines opérations de robot manipulateur à deux degrés de liberté.

Cette thèse se décompose en quatre chapitres et est organisée de la façon suivante :

Nous commençons par une introduction sur la supervision, et plus précisément, une définition et les composants de ce système, les différentes approches proposées dans la littérature dédient généralement à deux composants pour la conception d'un module de supervision, L'une est le diagnostic, et l'autre est la commande tolérante aux défauts.

Le deuxième chapitre explique les différents modèles utilisés pour présenter le mouvement des articulations d'un robot manipulateur, Il existe trois approches de modélisation d'un robot manipulateur qui sont le modèle géométrique, le modèles cinématique et le modèle dynamique. Nous donnerons quelques définitions de ces modèles. Aussi nous allons présenter la commande des bras manipulateurs, et deux types de correcteurs qui sont le proportionnel dérivateur, et le proportionnel intégrateur dérivateur qui sont les plus employés dans la commande.

Le troisième chapitre est consacré à l'étude des méthodes et des outils de l'intelligence artificielle. Nous présenterons le système flou et les notions de bases sur lesquelles reposent se système puis nous expliquons leur principe de fonctionnement et leur différente composantes. Nous aborderons ensuite, les outils réseaux de neurones ainsi qu'une classification des différentes structures. Enfin une description de système hybride neuro-flou (ANFIS).

Le chapitre quatre montre notre contribution pour la supervision d'un robot manipulateur, cependant nous présenterons un type de communication informatique, ce type est le protocole de communication TCP-IP. Par la suite, nous proposons une stratégie qui rassemble les outils précédents pour le suivi du comportement d'un robot manipulateur esclave et la détection des défaillances. Enfin une simulation par System-Generator en MATLAB pour l'implémentation matérielle sur cible FPGA.



L'approche de supervision proposée est caractérisée par :

- Un schéma d'un maître et esclave.
- Une communication de protocole TCP.
- Une méthode mathématique pour enrichir l'étape de détection et l'évaluation de de système de supervision.
- Une étape de commande tolérante aux défauts d'ANFIS.
- Nous montrerons l'efficacité de l'approche de supervision proposée. Par la simulation sur un bras manipulateur de deux degrés de liberté.

Enfin, on termine ce travail par les conclusions et perspectives des travaux de recherches.

# **CHAPITRE 1**

## Introduction à la supervision

## 1.1 Introduction

Les techniques de diagnostic, de surveillance et de supervision sont très importantes en l'industrie, car le volume de production et la sécurité de systèmes industriels sont vérifiés à base de ces tâches, ils sont encore en pleine évolution. Plusieurs travaux ont été faits par des chercheurs pour satisfaire les objectifs exigés comme la sécurité et l'amélioration de fonctionnement du système industriel. La diversité des comportements des systèmes industriels qui peuvent être observés rend les étapes de système de supervision différentes pour chaque système industriel. L'adoption des méthodes de supervision de système non linéaire, souvent jugées difficiles à comprendre, compliquées à mettre en œuvre et dont l'analyse systématique des performances se révèle complexe. Alors, il faut comprendre le fonctionnement de système et les techniques qui s'appliquent à tous les systèmes non linéaires. Les techniques de supervision de systèmes non linéaires sont bien maîtriser depuis plusieurs années surtout avec l'utilisation des techniques neuro-flous pour l'analyse de ces systèmes, en particulier dans le but de préciser les étapes de supervision.

Ainsi, on s'intéresse à la supervision des systèmes industriels et l'étude porte sur les étapes et la structure de la supervision. Dans ce contexte, nous décrivons quelques notions d'état de système industriel ainsi que les outils qui peuvent être utilisés pour la supervision.

## 1.2 Quelques Notions

Dans cette section, nous décrivons les définitions de base concernant la supervision des systèmes industriels [Villemeur, 1988], [Dubuisson, 1990], [Basseville et al. 1996].

- **Signatures** : les signatures sont des informations qui sont souvent utilisées pour la détection d'un fonctionnement anormal de système. La complexité des signatures des systèmes et des processus à diagnostiquer est liée à l'anomalie recherchée.
- **Anomalie** : c'est un état extraordinaire que la loi naturelle ou à la logique.
- **Dégradation** : c'est une décroissance de performances des fonctions assurées par un équipement.
- **Erreur** : Il s'agit d'un évènement qui se produit au niveau du système ne correspondant pas au cahier des charges.
- **Dysfonctionnement** : Fonctionnement de manière incomplète du système.

- **Perturbation** : c'est une mesure différente que les mesures originales du système physique. Les bruits de mesure sont des exemples de perturbation.
- **Défaut** : un défaut est un mauvais fonctionnement du système menant à une dégradation du procédé et de ses performances. Ce défaut peut trouver en les composants de processus (les capteurs et les actionneurs).
- **Panne** : c'est un état du système après un fonctionnement anormal, cette panne est souvent à cause de défaillance. La **panne** est l'inaptitude d'un dispositif à accomplir une fonction requise.
- **Résidus** : Sont des valeurs qui permettent d'indiquer la présence d'un défaut de système (régulateur, capteur ou actionneur). La génération de résidus s'effectue par l'analyse d'informations disponibles et l'information théorique d'un modèle.

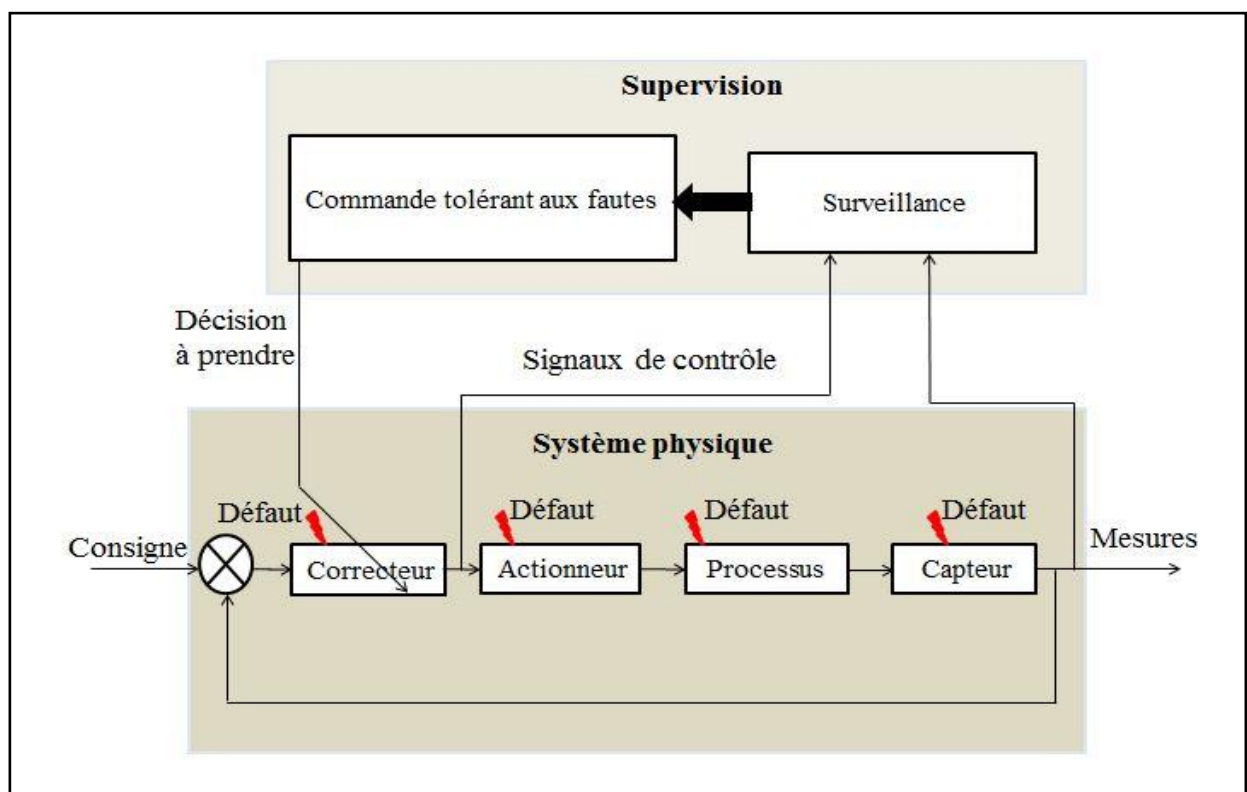
### 1.3 La supervision

Les installations de système industriel peut contenir des erreurs ou des défauts, ce qui implique de mauvais fonctionnements de ce système, ces défauts ou ces erreurs pourraient se trouvées dans différents niveaux, par exemple, au niveau capteur et actionneur, tous ces contraintes indésirables rendent la supervision impérative afin de protéger le système industriel et optimiser sa production. Ainsi, la supervision a été appliquée dans différents domaines et par conséquent, il y a eu plusieurs stratégies et structures qui ont été largement étudiées par des chercheurs [Khedhri, 2002], [Zemouri, 2003] et [Guillaume, 2004]. Dans ce contexte, nous avons essayé de dresser un état de l'art en décrivant les techniques les plus répandues pour la supervision des systèmes industriels.

Plusieurs études existent pour l'analyse de dysfonctionnement d'un robot manipulateur. Dans le travail de [Renato et al, 2007] les auteurs ont fait une commande de reconfiguration de système suivant le défaut localisé dans les manipulateurs coopératifs liés de manière rigide à une charge indéformable. Les performances du système de détection et localisation des défauts peuvent être améliorées avec des tests supplémentaires, à base de l'analyse de la position et de la vitesse et les réseaux neurones. Pour le travail de [Tesheng et Mao-ChiaoWeng, 2011], Ils **ont** proposé un schéma hiérarchique à multi-modèles pour détecter et isoler les défauts d'actionneurs de robot manipulateur. Le schéma de multi-modèles utilise un modèle pour un défaut particulier. Ensuite, tous les modèles de défauts ainsi que le modèle normal, c'est-à-dire le modèle décrivant le fonctionnement normal du robot, sont contenus dans un ensemble des modèles. Pour chaque défaut

survenant, le modèle qui correspond le mieux au comportement actuel du robot passe au modèle de défaut associé. Dans ce travail l'auteur a utilisé la théorie bayésienne et l'algorithme du filtre de Kalman. Aussi, les auteurs de [Saleh Ahmad et al, 2013] ont développé un système de détection de pannes réparties, pour la reconfiguration de robot manipulateur, alors la commande de couple est traitée et comparée avec le couple estimé à partir du modèle dynamique non linéaire.

Les fonctions de supervision servent alors à éviter ou lorsque ce n'est pas possible de gérer les états indésirables du procédé, à prendre les actions appropriées pour maintenir une bonne conduite et éviter tout endommagement ou accident [Nan Zhang, 2010], la figure 1 présente un schéma de système de supervision.



**Figure I.1 :** Structure d'un système avec une procédure de supervision

En général, un système de supervision s'articule autour des étapes suivantes :

- L'extraction des informations à partir de mesures appropriées ou d'observations du système industriel.
- L'élaboration des caractéristiques et signatures associées à des symptômes.
- L'exploration de dysfonctionnement.

- La mise en œuvre d'une méthode de rapport de la défaillance à partir de l'utilisation des connaissances sur les relations de cause à effet.
- La prise de décision.

L'objectif principal de la supervision était la mise en place d'un cadre à base de mécanisme généralement permettant de :

- ✓ Donnée une visualisation de l'état et le fonctionnement de système industriel.
- ✓ Pour l'atteinte des objectifs de production (quantité, qualité, traçabilité, sécurité...).

Suivant les recherches, nous pouvons dire que le système de supervision se compose par deux modules, la première étant le diagnostic ou la surveillance et la seconde est la commande tolérante aux défauts.

### **1.3.1 La surveillance**

La surveillance d'un système industriel permet de détecter, d'isoler, et même d'identifier toute défaillance susceptible de survenir, en se basant sur l'association d'informations de fonctionnement réel fournies par des capteurs et de comportement nominal théorique fourni par un modèle [Combacau, 1991], [Toguyeni, 1992] et [Zwingelstein, 1995]. Il s'agit notamment de détecter et classer les défaillances en observant l'évolution du système puis à le diagnostiquer en localisant les éléments défaillants et en identifiant les causes premières. Dans ce contexte, il y a deux approches qui peuvent être utilisées : *méthode à base de modèle* et *méthode sans modèle* [Dash et al. 2000]. Dans le travail de [Zemouri, 2003], la surveillance est une méthode d'observation et évaluation de système industriel en temps réel, informationnel qui traite l'état du système et fournit des indicateurs.

#### **1.3.1.1 Méthode de surveillance à base de modèle**

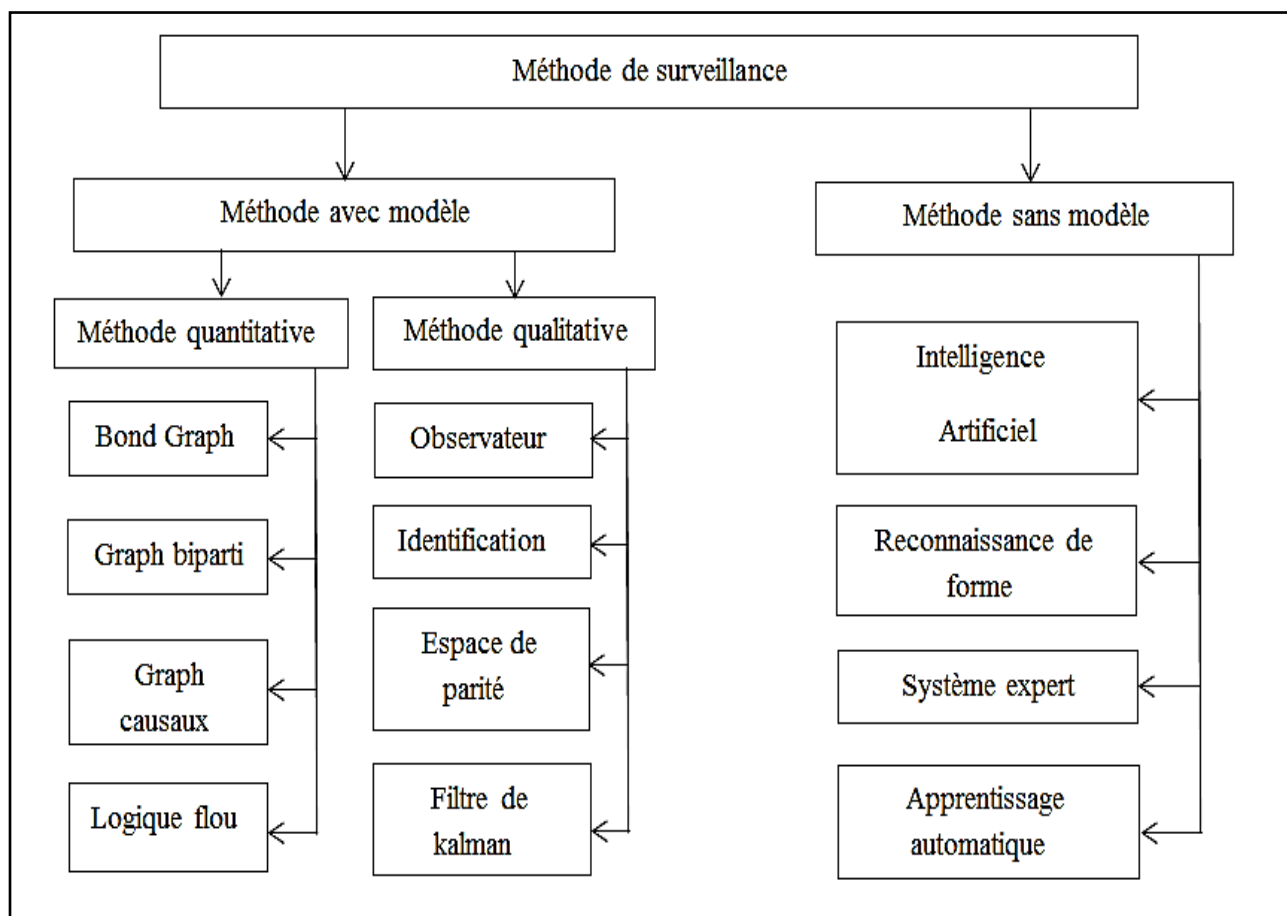
Les méthodes de surveillance à base de modèles emploient une place importante dans la littérature. Particulièrement leur utilisation dans les applications critiques (systèmes énergétiques, systèmes de transport, industrie lourde), s'est considérablement développée. Une vaste littérature existe dans le cas des systèmes dynamiques linéaires, en ce qui concerne les systèmes dynamiques non linéaires, par contre il y a moins des travaux sur les systèmes dynamiques non-linéaire, ont été réalisés à ce jour. La méthode à base de modèle dispose le comportement normal et les comportements défaillance du système, cette méthode est largement interprétée dans des études de

[Chow et Willsky, 1984] qui propose un système de détection et d'identification pouvant être réalisé en concevant un processus de génération résiduelle robuste. La redondance analytique, à base de la génération résiduelle, est caractérisée en termes d'espace de parité. En utilisant le concept des relations de parité, les résidus peuvent être générés de plusieurs façons et la conception d'un processus de génération résiduelle robuste peut être formulée comme un problème d'optimisation minimax. Dans [Fussel et Isermann, 1998] la détection et le diagnostic des défauts. Il commence par la prise en compte d'une procédure basée sur la connaissance et les informations analytiques. Ensuite, différentes méthodes de détection de défauts sont considérées, qui extraient les caractéristiques des signaux mesurés et utilisent des modèles de processus et de signaux. Ces méthodes sont basées sur l'estimation de paramètres, l'estimation d'état et les équations de parité. Par contre l'objectif de travail de [Baroni et al, 1999] est la reconstruction du comportement du système actif à partir d'un ensemble d'événements observables. Le processus de surveillance comporte trois étapes: l'interprétation, la fusion et la génération de diagnostic. L'interprétation génère une représentation du comportement d'une partie du système actif sur la base d'événements observables. La fusion combine le résultat de plusieurs interprétations dans une nouvelle interprétation plus large. Les informations sont générées sur la base d'événements de défaut éventuellement incorporés dans le comportement reconstruit.

### **1.3.1.2 Méthode de surveillance sans modèle**

La méthode sans modèle est fondée sur les observations historiques ou présentes antérieures, ce type de surveillance est utilisée dans le cas de systèmes complexes traitant des données très variées (analogique, numérique) ou imprécises (bruit de mesure). Suivant [Basseville, 1988] la fonction surveillance est vérifiée à base des données quantifiables et qualifiables, en conséquence cette méthode indique le dysfonctionnement, avant qu'il n'arrive et sans fausses alarmes qui peuvent ralentir le fonctionnement de système industriel. Dans [Dubuisson et al. 2001] la surveillance peut se atteindre à travers la méthode de reconnaissance de formes, l'ensemble des états est même à un ensemble de classes et le vecteur forme est le vecteur composantes des paramètres observés sur le système. Dans ce modèle, nous citons les techniques de l'intelligence artificielle qui servent comme outils pour l'aide à la décision. Leur réponse est donc plus élaborée que celle des techniques de bas niveau. Cette réponse peut être obtenue à partir des données brutes venant directement des variables de surveillance ou des données traitées venant des sorties des traitements de bas niveau.

La figure I.2 montre les deux méthodes de surveillance.



**Figure I.2 :** Les différentes méthodes à base de modèle et sans modèle de surveillance

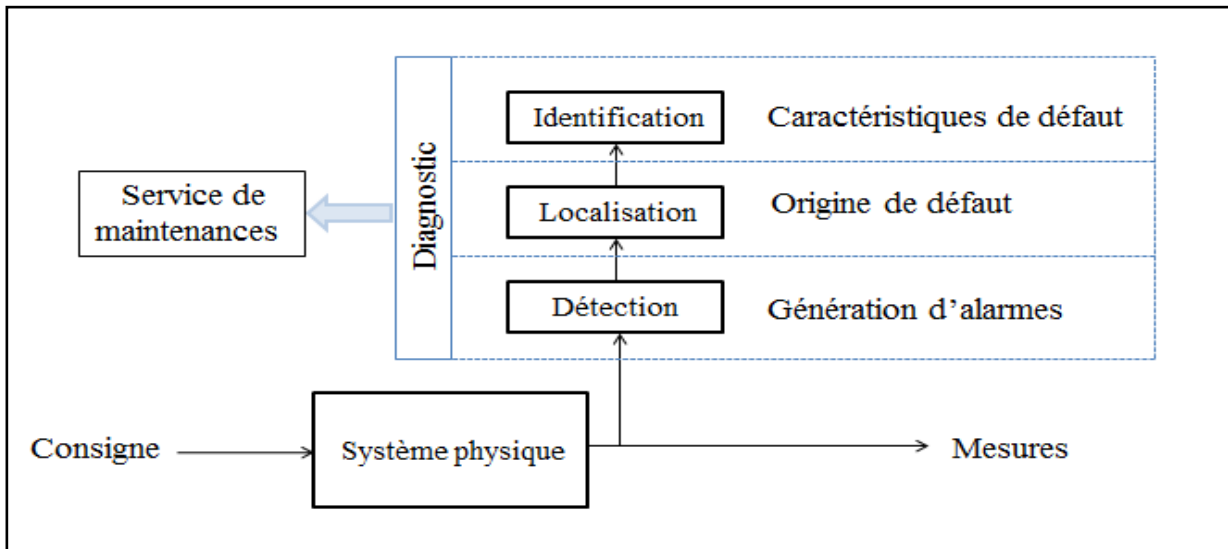
### 1.3.2 Diagnostic de défauts

L'une des caractéristiques désagréables de tous les systèmes physiques est le fait qu'ils sont vulnérables aux défauts, aux dysfonctionnements et, plus généralement, aux modes de comportement inattendus. Ceci explique pourquoi il existe un besoin continu de systèmes de surveillance fiables et universels basés sur des stratégies de diagnostic de pannes adaptées et efficaces. Cela est particulièrement vrai pour les systèmes d'ingénierie, dont la complexité augmente de manière permanente en raison du développement inévitable de l'industrie moderne ainsi que de la révolution des technologies de l'information et de la communication. En effet, la conception et le fonctionnement des systèmes d'ingénierie nécessitent une attention accrue en termes de disponibilité, de fiabilité, de sécurité et de tolérance aux pannes. Ainsi, il est naturel que le diagnostic de défaut joue un rôle fondamental dans la théorie et la pratique du contrôle moderne.



Cela se reflète dans de nombreux articles sur le diagnostic des erreurs dans de nombreuses conférences et revues axées sur le contrôle. En effet, une grande quantité de connaissances sur diagnostique de défaut a été accumulé à travers la littérature scientifique depuis le début des années 1970. En conséquence, un large éventail de techniques de diagnostic de pannes ont été développées.

L'une des principales catégories des techniques de diagnostic des défauts est celle fondée sur un modèle, dans laquelle se trouve le modèle analytique de système industrie à surveiller est supposé disponible. Malheureusement, une difficulté fondamentale liée à l'approche basée sur un modèle est le fait qu'il existe toujours des incertitudes de modélisation de perturbations non modélisées, simplifications, idéalizations, linéarisations, inexactitudes des paramètres du modèle, etc. Une autre difficulté importante concerne la caractéristique intrinsèque non linéaire de la plupart des systèmes physiques. En effet, à quelques exceptions près, la plupart des approches bien établies présentées dans la littérature peuvent être appliquées aux systèmes linéaires seulement. Ce fait, bien sûr, limite considérablement leur application dans les systèmes de contrôle industriels modernes. Par conséquent, il est clair que les techniques de modélisation et de diagnostic de pannes sont nécessaires pour les systèmes dynamiques non linéaires, ce qui doit garantir la robustesse aux incertitudes de modélisation. Actuellement, nombreux chercheurs dans ce domaine, considère que les Systèmes neuro-flou comme une alternative solide aux méthodes classiques, utilisées dans le cadre de diagnostic des défauts basé sur un modèle. En effet, en raison de leurs propriétés intéressantes comme approximation fonctionnelle, les Systèmes neuro-flous s'avèrent être un outil très prometteur pour traiter les processus non linéaires. Dans les processus industriels, généralement, le diagnostic est comme des ensembles d'actions visant à évaluer l'état d'un système physique à partir de la connaissance des divers états de fonctionnement et l'observation du procédé. Donc le diagnostic détermine les divers états de fonctionnement, le degré de défaillance du système, l'origine de ces défaillances et leurs causes hors ligne. En industrie, les principales activités concernées par le diagnostic sont trois étapes : la détection, localisation et l'identification respectivement, il y'a aussi possibilité de faire le diagnostic sans les étapes de localisation et d'identification par exemple en cas où les défauts sont connus.



**Figure I.3 :** Principe de fonctionnement et les étapes de diagnostic

L'objectif principal du système de diagnostic de panne est de déterminer l'emplacement et le temps d'apparition des défauts possibles sur la base de données accessibles et de connaissances sur le comportement du processus diagnostiqué.

### 1.3.2.1 Détection de défaut

La détection joue un rôle important et il est nécessaire dans la supervision, l'importance est prononcée dans le temps de détection de défaut, avec un temps de détection minimal le système de supervision sera précis et plus acceptable par l'opérateur, généralement la méthode de détection mise en forme par la génération de résidus avec la technique de seuil est la plus utilisée. Par définition, la détection a pour objectif d'indiquer la présence d'un défaut, à base de comparaison entre le comportement courant du système et des données de référence (historique fonctionnement du système), Les techniques utilisées pour atteindre la détection sont : le seuil et les résidus. L'un des problèmes décisifs dans la surveillance et la supervision est la génération de fausses alarmes et non détection. Ces problèmes peuvent être trouvés dans le cas des incertitudes de mesure, paramétriques et l'évaluation des dérivées des résidus. Une méthode de détection de défauts permettant de détecter les défauts structurelles et/ou de capteurs dans les robots manipulateurs, tels que les défauts d'actionneurs ou les mesures de capteur défectueuses, peuvent être identifiés comme des variations rencontrées dans le vecteur de paramètres. Cette méthode basée sur un algorithme d'identification est nommé par la relation interactive [Vasso R. et Anthony, 2008]. Une autre

stratégie de détection et d'isolement des défauts, est-elle basée sur l'analyse du comportement du robot en fonction des défauts mécaniques et électriques réels [Cesare Fantuzzi, 2003].

#### **a) Les techniques de seuillage**

Le seuil généralement est une fonction mathématique développée à partir d'analyse du signal de résidus en cas de système sain. L'utilisation de cet outil afin de définir la limite du fonctionnement normal de système industriel, selon les recherches précédentes, il y' a deux types de seuils :le seuil fixe [Pickholtz R.L et al, 1982] et le seuil adaptatif, ces techniques doivent assurer de une bonne détection et un taux d'alarme réduit. Donc les seuils peuvent être déterminés à partir d'une analyse des résidus. La détection à base de seuil fixe consiste à comparer les signaux (signaux capteurs ou résidus) avec des vecteurs d'amplitude fixe, le passage de ce seuil par l'un des signaux de résidus génère une alarme. Ce type de méthode est très simple à mettre en œuvre mais elle est très sensible aux fausses alarmes ou la non détection [Dreyfus et al, 2002]. Par contre, le seuil adaptatif est dynamiquement changé selon la nature de résidus ou les mesures du système industriel.

A ce titre, le calcul du seuil peut être effectué par plusieurs méthodes. Par exemple, dans [Zhang et al. 2002] une sélection de la valeur du seuil a été posée par l'observation de résidus en cas de présence d'un défaut. Le seuil est défini donc comme étant la limite supérieure de résidus d'un système sain [Caccavale et al. 2008]. En effet, il faut choisir un seuil bas pour assurer également que les défauts de petits effets peuvent être détectés et sans générer de fausse alarme. Dans le but d'améliorer la détection, nous proposons une méthode de détermination des seuils en se basant sur une analyse de résidus en état sain de robot manipulateur.

#### **b) Génération de résidu**

Le diagnostic est basé sur l'existence d'un banc d'observateurs capable de générer suffisamment de résidus sensibles aux défauts. La génération de résidu exploite la différence entre des informations disponibles sur le système et des informations historiques fournies par un modèle, aussi les résidus sont des valeurs d'un signal numérique ou analogique. Par conséquent les résidus généralement doivent être proches de zéro ou s'y ramener dans les cas de système sain et ils doivent être très différents de zéro après l'apparition d'un défaut. Dans [Frank, 1990],on trouve une technique basée sur un observateur pour détecter un défaut dans un système dynamique non linéaire. Une méthode de contrôle adaptatif de paramètre pour déterminer les limites de l'erreur a été

faite dans [Isermann, 1990]. Un algorithme d'évaluation de résidus basé sur l'approche d'observateur adaptatif non linéaire pour la détection de défauts a été développé dans [Ding et Frank, 1991]. Dans [Isermann et Freyermuth, 1991], les auteurs ont utilisé une technique d'estimation de paramètres pour détecter une défaillance dans un système dynamique non linéaire. Le générateur de seuil est utilisé pour produire des seuils dynamiques basés sur l'état actuel du système à chaque position du contrôleur pour chaque articulation de robot a été traité dans [Paredis, 1994].

La génération et l'analyse des résidus par les réseaux de neurones artificiels (RNA) ont également utilisées par [Kourad, 2012]. Pour détection et isolation des défauts dans les roues du robot mobile à base de modèle de réseau neuronal est proposée dans [Elias, 2004]. Un algorithme de détection pour une génération de résidus fiables a été déjà proposé dans le travail de [Basseville et Nikiforov 1993].

### **1.3.2.2 Localisation de défaut**

La localisation s'exécute après qu'il ait une détection de défaut. Généralement, cette étape est une spécification d'un défaut qui a été détecté au système industriel pour donner leur classement (les types et la place de défaut) : elle attribue le défaut à un sous-système particulier comme les capteurs d'instrumentation, la commande ou le système commandé en indiquant quel organe ou composant est affecté par celui-ci.

### **1.3.2.3 Identification de défaut**

L'identification peut comprendre une procédure visant à déterminer des paramètres de défaillance (l'instant de détection, ..), c'est-à-dire son origine. Ainsi, elle peut servir d'assurer le suivi d'évolution détaillée du système.

Les méthodes de détection et localisation de défaut sont basées sur des techniques analytiques ou de calcul pour l'obtention des signaux. Ces différents signaux appelés les résidus, le point de séparation ne peut être seulement les méthodes utilisées mais aussi le type de linéarité (linéaire-non linéaire-bilinéaire) du système qui sera traité. Comme les robots manipulateurs sont des systèmes non linéaires, les études sur les systèmes non linéaires devraient être prises en compte.

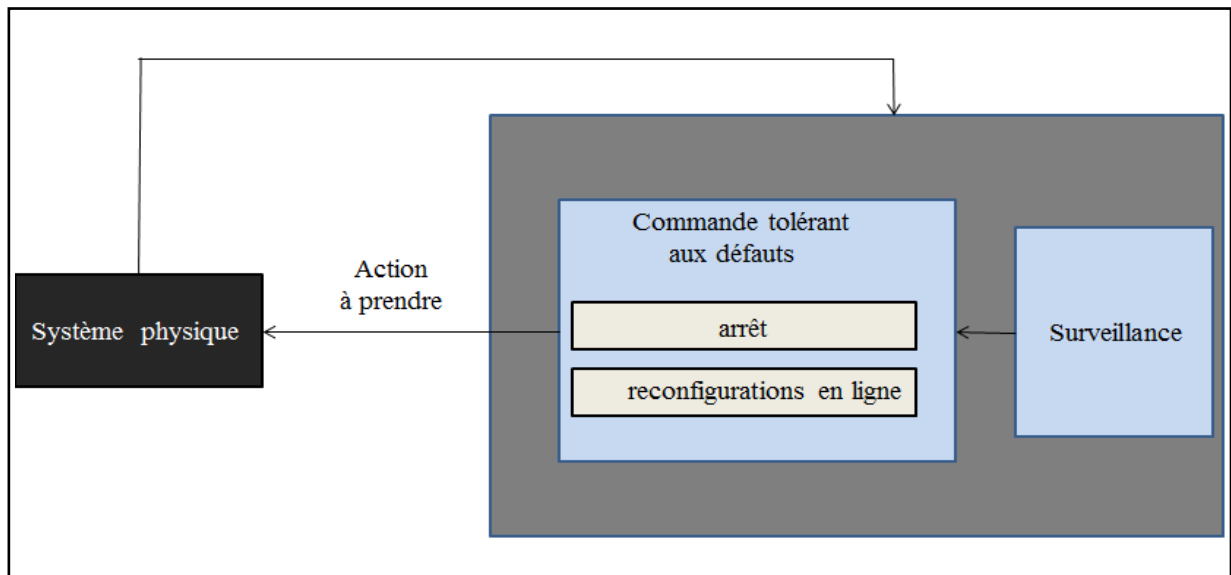
### **1.3.3 Commande tolérante aux défauts**

La commande tolérante aux défauts comme des instructions générés, qu'il faut appliquer par l'opérateur pour compléter la tâche de supervision suivant les informations de diagnostic ou la

surveillance, pour atteindre un fonctionnement normal de système [Blanke M. et al 2003]. Par conséquent, ce module permet de retenir la stabilité et un certain degré de performance en présence de défauts [K. Astrom et al 2000]. Les critères de choix de l'action à mener et les conditions d'application pour remettre le système à l'un des fonctionnements normaux qu'il faut exécuter par l'opérateur (homme ou machine) relie à l'affect de défaut. Généralement les phases principales de commande tolérante aux défauts d'un système de supervision sont:

- ✓ Commande d'arrêt : Ce type de commande par exemple dans les cas de détection d'un ou des défauts comme le délabrement de système industriel.
- ✓ Commande de reconfigurations en ligne : Ce type est la plus utilisé comme par exemple en cas de présence d'un défaut de moins risque dans les systèmes industriels.

La commande tolérante aux défauts agit immédiatement sur les actionneurs du système industriel qui permet d'assurer le fonctionnement en absence de défaillance. La figure I.4 donne cas général de la commande tolérante aux défauts.



**Figure I.4 :** Schéma générale commande tolérante aux défauts

#### 1.4 Supervision des robots manipulateurs

Il existe plusieurs sources de défauts dans les robots, tels que l'électricité, la mécanique et l'hydraulique, [Renato, 2007] Quatre catégories de défauts sont considérés: vibration d'articulation, verrouillées d'articulations, mesures incorrectes de positions et mesures incorrectes de vitesses, une

méthode de détection et localisation a été faite sur un robot manipulateur pour détecter et localiser les défauts d'actionneur [Tesheng Hsiao, 2012]. Le cadre de cette étude, nous avons particulièrement étudié les défauts de frottement. Il faut noter que le frottement existe dans tous les mécanismes. Il peut être défini comme une force de réaction tangentielle entre deux surfaces en contact. Il existe différents types de frottement **sec**, **visqueux**, **lubrifié**, de **paroi** ou interne. Le frottement n'est pas une force fondamentale, mais le résultat d'interactions complexes entre des surfaces en contact dans une perspective nanométrique. La tribologie ; la science des surfaces en interaction dans un mouvement relatif, est donc principalement basée sur des études **empiriques**. Une raison de l'intérêt pour le frottement dans les articulations de manipulateur est la nécessité de modéliser pour faire la commande. Un modèle de frottement précis peut considérablement améliorer la performance globale d'un manipulateur en ce qui concerne la précision et la stabilité de la commande [Olsson et al, 1998], [Bona et Indri, 2005], [Guo et al. 2008], [Susanto et al, 2008] et [Kim et al. 2009]. Le frottement peut concerner le processus d'usure des systèmes mécaniques [Blau 2009], y compris les articulations de robots. Il existe également un intérêt pour la modélisation du frottement pour la surveillance de l'état du robot et la détection des défauts [Freyermuth, 1991], [Vemuri et Polycarpou, 2004], [McIntyre et al. 2005], [Brambilla et al, 2008] et [Mattone et Luca 2009].

Dans une articulation de robot, avec plusieurs composants interagissant tels que des engrenages, des paliers et des arbres qui tournent / glissent à différentes vitesses et sous différents niveaux de lubrification, il est difficile de séparer et de modéliser le frottement au niveau des composants. Une approche typique consiste à considérer ces effets collectivement, comme un frottement articulaire.

Le frottement dépend de la géométrie du contact, de la topologie, des propriétés des matériaux, de la vitesse relative, du lubrifiant, ... etc. [Al-Bender et Swevers 2008]. Selon la configuration, chacun de ces facteurs sera plus ou moins important pour le frottement total. Dans une articulation de robot, il est connu que le frottement peut se rapporter à :

- Température dans l'articulation.
- La force ou le couple exercé.
- L'accélération de l'articulation.
- Propriétés lubrifiantes.

Par conséquent, l'accent dans cette thèse sera mis sur l'étude du frottement dans les axes principaux. Les effets sont ainsi résumés [Al-Bender et Swevers 2008].

- **Charge :** Les effets de la charge résultent de l'augmentation consécutive de la pression de contact entre les surfaces en contact, conduisant à une augmentation généralisée de frottement, avec une augmentation plus importante aux très basses vitesses.
- **Lubrifiant :** Dans les mécanismes lubrifiés, l'épaisseur de la couche de lubrifiant et sa viscosité jouent un rôle important dans les propriétés de frottement résultantes. La viscosité plus élevée conduit à des forces de cisaillement plus élevées et donc le frottement sera plus élevés.
- **Température :** La viscosité dépend également de la température du lubrifiant, plus la température est élevée, plus la viscosité est basse.
- **L'usure :** L'augmentation du frottement avec l'usure est liée à l'accumulation de débris d'usure dans le lubrifiant circulant.
- **Contrecoup :** La décroissance du frottement peut s'expliquer par un desserrage conséquent des composants de la boîte de vitesses, ce qui réduit les pressions de contact. Notez que le contrecoup peut résulter d'un processus d'usure dégénéré, où la quantité de matériau éliminée par l'usure commence à être suffisamment importante pour créer des autorisations non souhaités entre les surfaces.

Pour la représentation mathématique de frottement, nous pouvons dire qu'en raison de la nature complexe du frottement dans l'articulation de robot, il est courant d'accepter des modèles basés sur des observations empiriques des phénomènes. L'histoire du développement des modèles empiriques de frottement est vaste, par exemple dans [Dowson, 1998] différentes modèles ont été proposés pour modéliser ce comportement dynamique de frottement, dans [Dupont et al, 2002] une approche typique consiste à considérer toutes les dynamiques dans un seul état et quelques exemples sont présentés dans [Al-Bender et Swevers 2008], [Harnoy et al. 2008] et [Strström et Canudasde Wit 2008].

Plusieurs travaux sur diagnostic et la supervision des robots manipulateurs, nous citons, celui de [Renato, 2007] qui propose une stratégie de supervision. Il comporte deux types de

reconfiguration, l'une pour les défauts de la mesure incorrecte de positions et vitesses est basée sur l'estimation de positions et vitesses angulaire. L'autre pour les défauts de vibration d'articulation la commande de reconfiguration va permettre de générer un signal de couple de correction. Dans [Tesheng Hsiao, 2012] l'auteur propose une méthode basée sur la théorie de pseudo bayésienne qui permet l'estimation des paramètres pour localiser l'articulation qui contient le défaut. Dans [Mien Van, 2013] l'auteur propose un observateur de réseau de neurones pour estimer les incertitudes de modélisation. Basé sur l'estimation d'incertitude obtenue, un observateur de mode glissant du second ordre est alors conçu, pour fournir l'estimation de vitesses et fournir l'information de défaut pour la détection, l'isolement et l'identification de défaut.

## **1.5 Conclusion**

Nous nous sommes intéressés dans ce chapitre au système de supervision, **et ceci** pour comprendre son principe de fonctionnement, connaître des notions et citer les différents états de système industriel. Pour identifier les fonctionnements normaux et anormaux par l'application de supervision, Il y'a des différents structures de supervision possible pour réaliser la tâche exigée, Donc le choix exacte d'une structure de la supervision selon le système industriel étudié, ce choix est important pour garantir l'efficacité.

Le chapitre suivant sera consacré sur la présentation de principales théories relatives à la modélisation d'un robot manipulateur à deux degrés de liberté.



## **CHAPITRE 2**

Modélisation et commande de robots manipulateurs

## 2.1 Introduction

Pour réaliser des commandes performantes, comme commande de position, commande de vitesse, commande hybride position force...etc. d'un robot manipulateur suivant leur objectif de manipulation précise d'une pièce ou d'un outillage, il est impératif de connaître leur cinématique et la dynamique de ce robot. Généralement les robots manipulateurs sont des systèmes mécaniques poly articulés, système à évènement discret, fortement non linéaires et d'une dynamique qui peut être mal définie. Une modélisation multi variable est impérative pour exécuter des différentes actions. Aussi pour savoir le positionnement de différents axes ou la connaissance l'état de robot comme le position angulaire, vitesse angulaire et la force exercé sur un temps quelconque. Donc la modélisation doit être assurée une définition mathématique de système robotisé pour réaliser des commandes ou connaître les fonctionnements de robot facilement par la simulation.

Objectif de ce chapitre est l'étude d'un bras manipulateur à deux degrés de liberté (2ddl) et résoudre des équations mathématiques régissant le mouvement de ce bras. Nous commençons par les modèles géométriques directs et inverses des manipulateurs, les éléments concernant la modélisation cinématique et modèle dynamique, direct et inverse des manipulateurs ainsi que la matrice inertie. Enfin un aperçu sur le signal de commande et les types de commande.

## 2.2 Modélisation des robots manipulateurs

Les modélisations de manipulateurs industries sont très importantes pour simuler leur comportement selon les objectifs. La modélisation est une méthodologie permettant de définir des différents modèles mathématiques, modèles géométriques, cinématiques et dynamiques. La difficulté de cette méthodologie réside dans la complexité de la cinématique de la chaîne articulée. Aussi le nombre de degrés de liberté, le type des articulations le fait que la chaîne soit ouverte simple, arborescente ou fermée. Les outils mathématiques que nous présentons dans ce travail sont basés sur des descriptions des mécanismes fondés sur des notations originales qui permettent d'avoir la même approche quelle que soit la structure.

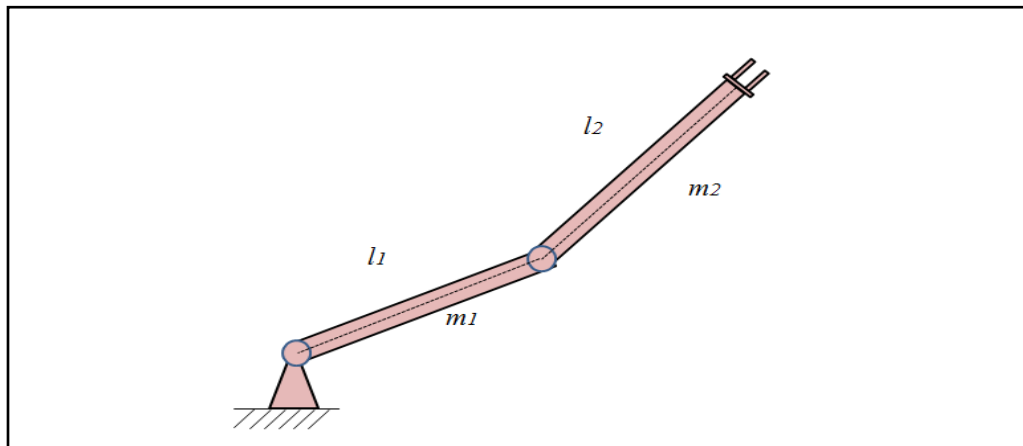
L'utilisation de modèle géométrique, cinématique et dynamique en la simulation ou en la commande de robot manipulateur réel, doivent démontrer le comportement géométrique, cinématique ou dynamique du robot de façon précise. Cette contrainte nous oblige de mettre en œuvre des procédures efficaces d'identification de paramètres de robot manipulateur. Donc Les techniques proposées permettent de résoudre des modèles plus performants.

Les modèles de bras manipulateur étudiés sont : le modèle géométrique direct et inverse qui relatent la situation de l'organe terminal en fonction des variables articulaires du mécanisme et inversement, le modèle cinématique direct et inverse qui informent la vitesse de l'organe terminal en fonction des vitesses articulaires et inversement et le modèle dynamiques définissant les équations du mouvement du robot, qui permettent d'établir les relations entre les couples ou forces exercées par les actionneurs et les positions, vitesses et accélérations des articulations. Nous avons cité des travaux qui ont permis de construire le modèle de robot manipulateur [Dillon, 1973], [Khalil, 1976], [Zabala, 1978], [Kreuzer, 1979], [Aldon, 1982], [Cesareo, 1984], [Megahed, 1984], [Murray, 1984], [Kircanski, 1984], [Burdick, 1988], [Izaguirre, 1986] et [Khalil, 1989a].

La figure I.1 illustre le robot industriel considéré dans ce travail. Il s'agit d'un robot manipulateur à 2ddl. Le mouvement des articulations est rotoïde avec les paramètres :

$l_i$  : Longueur du segment  $i$ .

$m_i$  : La masse du segment  $i$ .



**Figure II.1** : Robot manipulateur à deux degrés de liberté

### 2.3 Modèle géométrique

Les modèles géométriques permettent de connaître l'attitude (position et orientation) par rapport aux positions angulaires du robot. La modélisation des robots de façon systématique et automatique exige une méthode adéquate pour la description de leur morphologie. Nombreux méthodes procédés ont été proposées [Khalil, 1976] et [Khalil et Dombre, 1979].

### 2.3.1 Modèle géométrique direct

Le modèle géométrique est une relation mathématique qui basée sur les variables articulaires  $q_i$  afin de déterminer les variables opérationnelles  $X_i$  [Cyril, 2009]. Le modèle géométrique direct est présenté par l'équation explicite suivante :

$$X_i = f(q_i) \quad (\text{II.1})$$

Ou:

$q = [q_1, q_2, \dots, q_n]^T \in R^n$  : Vecteur des variables articulaires.

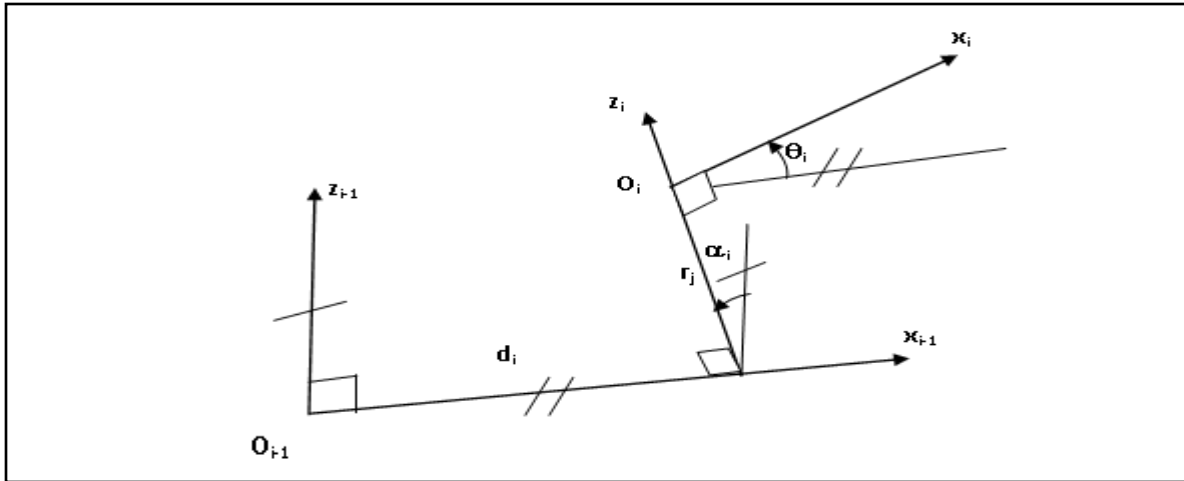
$X = [x_1, x_2, \dots, x_m]^T \in R^m$  : Vecteur des variables opérationnelles.

Nous supposons que les segments du robot sont parfaitement rigides et les articulations mécaniquement parfaites, il existe plusieurs méthodes pour calculer le modèle géométrique direct. La plus utilisé est celle de Denavit-Hartenberg [Denavit, 1955]. Méthode de Denavit Hartenberg a été développée pour des structures ouvertes simples, présente des ambiguïtés lorsqu'elle est appliquée sur des robots ayant des structures fermées ou arborescentes. La modification de cette méthode permet d'éviter les inconvénients cités précédemment, mais elle présente des redondances pour les mécanismes à structures de chaînes simples. Ce travail connu sou le nom de la méthode de Khalil-Kleinfinger.

- La convention de Denavit-Hartenberg :

La méthode de Denavit Hartenberg (DHM) Modifié est bien appropriée pour trouver les matrices de transformation homogène et choisir le système de référence d'un corps en robotique.

La figure II.2 présente les Paramètres de Denavit et Hartenberg.



**Figure II.2 :** Paramètres de Denavit et Hartenberg

- l'axe  $z_i$  est confondu avec l'axe de l'articulation  $j$
- l'axe  $x_i$  est la perpendiculaire commune à  $z_i$  et  $z_{i+1}$
- l'axe  $y_i$  est choisi de manière à créer un repère orthonormé direct avec  $(O_i, x_i, y_i, z_i)$
- $\alpha_i$  est l'angle correspondant à une rotation d'axe  $x_{i-1}$  entre  $z_{i-1}$  et  $z_i$
- $\theta_i$  est l'angle correspondant à une rotation d'axe  $z_i$  entre  $x_{i-1}$  et  $x_i$
- $d_i$  est la distance portée par  $x_{i-1}$  entre  $z_{i-1}$  et  $z_i$
- $r_i$  est la distance portée par  $z_i$  entre  $x_{i-1}$  et  $x_i$

Avec les paramètres :

$$q_i = \bar{\sigma}_i \theta_i + \sigma_i r \quad \text{et} \quad \bar{\sigma}_i = 1 - \sigma_i$$

où  $\sigma_i = 0$  et  $q_i = \theta_i$  en cas de liaison rotoïde.

et  $\sigma_i = 1$  et  $q_i = r$  en cas de liaison prismatique.

La rotation  $\theta$  autour de l'axe  $z$  représenté par la matrice de transformation homogène **Rot** telle que :

$$\mathbf{Rot}(z, \theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{II.2})$$

La translation par une distance  $d$  le long de l'axe  $x$  montré par la matrice de transformation homogène **Trans** telle que :

$$\mathbf{Trans}(x, d) = \begin{bmatrix} 1 & 0 & 0 & d \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{II.3})$$

d'après les définitions du paramétrage de Denavit Hartenberg modifié, la matrice de transformation homogène du repère  $i$  au repère  $i-1$  est donnée par la relation :

$${}^{i-1}T_i = \mathbf{Rot}(x, \alpha_i)\mathbf{Trans}(x, d_i)\mathbf{Rot}(z, \theta_i)\mathbf{Trans}(z, r_i)$$

$${}^{i-1}T_i = \begin{bmatrix} \cos\theta_i & -\sin\theta_i & 0 & d_i \\ \cos\alpha_i \sin\theta_i & \cos\alpha_i \cos\theta_i & -\sin\alpha_i & r_i \sin\alpha_i \\ \sin\alpha_i \sin\theta_i & \sin\alpha_i \cos\theta_i & \cos\alpha_i & r_i \cos\alpha_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{II.4})$$

A partir de paramétrage de Denavit Hartenberg on obtient le tableau II.1 d'état initiale du robot à 2ddl.

**Tableau II.1** : Les paramètres de Denavit Hartenberg du robot à 2ddl

$i$	$\sigma_i$	$l_i$	$\alpha_i$	$\theta_i$	$d_i$
1	0	$l_1$	0	$\theta_1$	0
2	0	$l_2$	0	$\theta_2$	0

On obtient les matrices de transformations homogènes de chaque articulation suivant le paramétrage de DH par :

$${}^0T_1 = \begin{bmatrix} C_1 & -S_1 & 0 & l_1 C_1 \\ S_1 & C_1 & 0 & l_1 S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{II.5})$$

$${}^1T_2 = \begin{bmatrix} C_2 & -S_2 & 0 & l_2 C_2 \\ S_2 & C_2 & 0 & l_2 S_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{II.6})$$

La matrice homogène de passage est donnée par la matrice finale  ${}^0T_2 = {}^0T_1 {}^1T_2$  :

$${}^0T_2 = \begin{bmatrix} C_{12} & -S_{12} & 0 & l_2 C_{12} + l_1 C_1 \\ S_{12} & C_{12} & 0 & l_2 S_{12} + l_1 S_1 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{II.7})$$

Avec :  $\cos \theta_i = C_i$ ,  $\sin \theta_i = S_i$

Et :  $\cos(\theta_i + \theta_j) = C_{ij}$ ,  $\sin(\theta_i + \theta_j) = S_{ij}$

D'après les deux expressions de  ${}^0T_2$  précédentes peut s'écrire sous la forme générale comme suivant :

$${}^0T_i = \begin{bmatrix} A(3 \times 3) & P(3 \times 1) \\ 0 & 1 \end{bmatrix} \quad (\text{II.8})$$

$$P(3 \times 1) = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} \quad (\text{II.9})$$

$$A(3 \times 3) = \begin{bmatrix} x_x & y_x & z_x \\ x_y & y_y & z_y \\ x_z & y_z & z_z \end{bmatrix} \quad (\text{II.10})$$

Avec :

$A(3 \times 3)$  : est la matrice la matrice de rotation du repère outil des vecteurs unitaires du repère  $\mathbf{R}_{i-1}$  dans le repère  $\mathbf{R}_i$ .

$P(3 \times 1)$  : est le vecteur de position qui définit l'origine du repère  $\mathbf{R}_i$  dans le repère  $\mathbf{R}_{i-1}$ .

Par la comparaison entre la forme générale et la forme calculé de matrice de transformation homogène nous obtenons les équations suivantes :

$$\begin{cases} x_x = C_{12} \dots\dots\dots (1) \\ x_y = S_{12} \dots\dots\dots (2) \\ x_z = 0 \dots\dots\dots (3) \end{cases} \quad (\text{II.11})$$

$$\begin{cases} y_x = -S_{12} \dots\dots\dots (4) \\ y_y = C_{12} \dots\dots\dots (5) \\ y_z = 0 \dots\dots\dots (6) \end{cases} \quad (\text{II.12})$$

$$\begin{cases} Z_x = 0 \dots\dots\dots (7) \\ Z_y = 0 \dots\dots\dots (8) \\ Z_z = 1 \dots\dots\dots (9) \end{cases} \quad (\text{II.13})$$

Les coordonnées de position

$$\begin{cases} p_x = l_2 C_{12} + l_1 C_1 \dots\dots\dots (10) \\ p_y = l_2 S_{12} + l_1 S_1 \dots\dots\dots (11) \\ p_z = 0 \dots\dots\dots (12) \end{cases} \quad (\text{II.14})$$

### 2.3.2 Modèle géométrique inverse

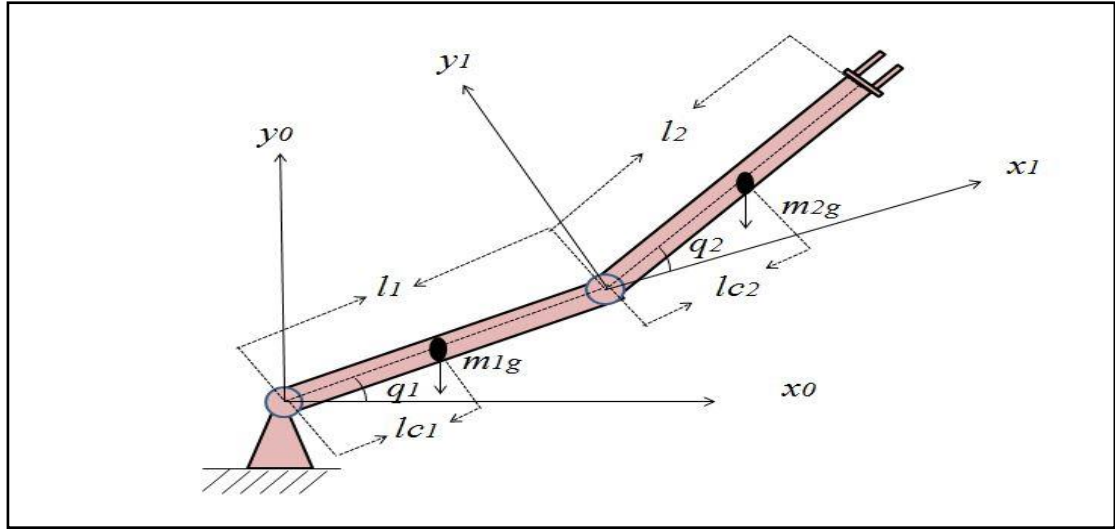
Le modèle géométrique inverse d'un robot manipulateur est nécessaire pour trouver les coordonnées articulaires en fonction des coordonnées opérationnelles de l'organe terminal. Le modèle s'écrit comme suit :

$$q = f^{-1}(x) \quad (\text{II.15})$$

Le modèle géométrique inverse (MGI) en général a des solutions multiples. La résolution du MGI est un problème complexe, car il est difficile de résoudre directement ces équations. Pourtant et selon la structure du manipulateur étudié, plusieurs méthodes ont été proposées qui permettant de résoudre le MGI. Les plus utilisables sont : Méthode de Paul basée sur le traitement de chaque cas particulier et convient pour la plupart des robots industriels [Paul, 1981], la méthode de Pieper agréé de trouver le MGI d'un robots à six degrés de liberté possédant trois articulations rotoïde d'axes concourants ou trois articulations prismatiques [Pieper, 1968] et méthode générale de Raghavan et Roth qui donne la solution générale des robots à six articulations à partir d'un polynôme de degrés au plus égal à 16 [Raghavan, 1990].

Nous présentons dans ce travail une approche de solution géométrique pour trouver le modèle géométrique inverse, La figure II.3 présente l'extraction d'équation (II.13) et (II.14) pour calculer les inconnus  $q_2$  et  $q_1$  .





**Figure II.3** : Les articulations ( $q_2$  et  $q_1$ ) d'un robot à deux degrés de liberté

Avec :

$l_{ci}$  : Longueur du centre de gravité du segment  $i$ .

$$\begin{cases} p_x = l_1 C_1 + l_2 C_{12} \\ p_y = l_1 S_1 + l_2 S_{12} \end{cases} \quad (\text{II.16})$$

$$\begin{cases} p_x^2 = l_1^2 C_1^2 + l_2^2 C_{12}^2 + 2l_1 C_1 l_2 C_{12} \\ p_y^2 = l_1^2 S_1^2 + l_2^2 S_{12}^2 + 2l_1 S_1 l_2 S_{12} \end{cases} \quad (\text{II.17})$$

$$p_x^2 + p_y^2 = l_2^2 C_{12}^2 + l_1^2 C_1^2 + 2l_2 C_{12} l_1 C_1 + l_2^2 S_{12}^2 + l_1^2 S_1^2 + 2l_2 l_1 S_{12} S_1 \quad (\text{II.18})$$

Par l'introduction des relations trigonométriques  $\sin^2 \alpha + \cos^2 \alpha = 1$ ,

$$\sin(\alpha + \beta) = \sin \alpha \cdot \cos \beta + \cos \alpha \cdot \sin \beta \quad \text{et} \quad \cos(\alpha + \beta) = \cos \alpha \cdot \cos \beta - \sin \alpha \cdot \sin \beta$$

Nous obtenons :

$$p_x^2 + p_y^2 = l_1^2 + l_2^2 + 2l_1 l_2 C_2 \quad (\text{II.19})$$

$$\begin{cases} C_2 = \frac{p_x^2 + p_y^2 - l_1^2 - l_2^2}{2l_1l_2} \\ S_2 = \pm \sqrt{1 - \frac{p_x^2 + p_y^2 - l_1^2 - l_2^2}{2l_1l_2}} \end{cases} \quad (\text{II.20})$$

Alors il y'a deux solutions possible de  $q_2$  peut écrire comme suivant :

$$q_2 = \text{ATAN2}(S_2, C_2) \quad (\text{II.21})$$

En simplifiant les équations pouvons encore écrire en développant  $C_{12}, S_{12}$  :

$$\begin{cases} p_x = l_1C_1 + l_2(C_1C_2 - S_1S_2) \\ p_y = l_1S_1 + l_2(S_1C_2 + S_2C_1) \end{cases} \quad (\text{II.22})$$

En regroupant suivant les inconnues  $C_1$  et  $S_1$ , nous trouvons :

$$\begin{cases} p_x = (l_1 + l_2C_2)C_1 - l_2S_2S_1 \\ p_y = l_2S_2C_1 + (l_1 + l_2C_2)S_1 \end{cases} \quad (\text{II.23})$$

Après l'utilisation de relations (II.15), (II.16) et (II.19), les variables  $C_1$  et  $S_1$  sont donnés par :

$$\begin{cases} C_1 = \frac{p_x(l_1 + l_2C_2) + p_y l_2 C_2}{p_x^2 + p_y^2} \\ S_1 = \pm \sqrt{1 - \left( \frac{p_x(l_1 + l_2C_2) + p_y l_2 C_2}{p_x^2 + p_y^2} \right)^2} \end{cases} \quad (\text{II.24})$$

D'où

$$q_1 = \text{ATAN2}(S_1, C_1) \quad (\text{II.25})$$

## 2.4 Modèle cinématique

Le modèle cinématique est un point de passage nécessaire pour la connaissance de fonctionnement de robot manipulateur. Il permet d'introduire la notion de vitesse de déplacement du mécanisme, pour déterminer la relation mathématique entre les vitesses de translation et de rotation

du corps terminal et des vitesses articulaires. Le modèle cinématique est une extension de modélisation basée sur le modèle géométrique, ce modèle utilisé en l'application d'une commande de vitesse.

#### 2.4.1 Modèle cinématique direct

Le modèle cinématique direct (MCD) permet de déterminer la vitesse de l'organe terminal dans l'espace opérationnel en fonction de la vitesse des variables articulaires. Le modèle cinématique est caractérisée par sa linéarité par rapport aux vitesses. La matrice jacobienne est un outil nécessaire pour résoudre le problème de la cinématique des robots manipulateur, et le modèle est donné par l'équation :

$$\dot{X} = J(q) \dot{q} \quad (\text{II.26})$$

Avec :

$J(q) \in R^{m \times n}$  est la matrice jacobienne de dimension ( $m \times n$ ) du mécanisme, la matrice jacobienne est une matrice des dérivées qui basée sur la configuration articulaire  $q$  et les coordonnées opérationnelles  $X$ .

L'utilisation de la matrice jacobienne pour établir la relation liant les efforts exercés par l'organe terminal sur l'environnement aux forces et couples des actionneurs et la facilité de calcul des singularités et de la dimension de l'espace opérationnel accessible du robot [Whitney, 1969], [Paul, 1981], [Borrel, 1986] et [Wenger, 1989].

L'équation mathématique de la matrice jacobienne est écrit par :

$$J(q) = \frac{\partial X}{\partial q} \quad (\text{II.27})$$

$J(q)$  peut-être décomposée en deux matrices, la matrice  $J_v$  (3x6) qui permet de calculer le vecteur de vitesse cartésienne et la matrice  $J_\omega$  (3x6) qui permet de calculer le vecteur de vitesse de rotation, donc on peut l'écrire comme suit :

$$J(q) = \begin{bmatrix} J_v(q) \\ J_\omega(q) \end{bmatrix} \quad (\text{II.28})$$

Pour le calcul de la matrice jacobienne nous utilisons la méthode de dérivation du modèle géométrique direct sous la forme suivante :

$$J(q) = \begin{bmatrix} \frac{\partial f_1}{\partial q_1} & \cdots & \frac{\partial f_1}{\partial q_n} \\ \cdot & & \cdot \\ \cdot & & \cdot \\ \frac{\partial f_m}{\partial q_1} & \cdots & \frac{\partial f_m}{\partial q_n} \end{bmatrix} \quad (\text{II.30})$$

### 2.4.2 Modèle cinématique inverse

Le modèle cinématique inverse (MCI) permet de préciser les vitesses articulaires  $\dot{q}$  d'articulation en fonction de vitesse opérationnelle  $\dot{X}$  imposée, c'est à dire déterminer la différentielle articulaire  $dq$  correspondant à une différentielle des coordonnées opérationnelles  $dX$  spécifiée :

$$\dot{q} = J(q)^{-1} \dot{X} \quad (\text{II.31})$$

On peut choisir une méthode qui réside dans la dérivée d'équation du MCI pour obtenir le modèle cinématique inverse, nous avons:

$$a_t = \dot{J}(q)\dot{q}^2 + J(q)\ddot{q} \quad (\text{II.32})$$

$$\ddot{q} = J(q)^{-1}[a_t - \dot{J}(q)\dot{q}^2] \quad (\text{II.33})$$

### 2.5 Modèle dynamique

L'intégration du modèle dynamique permet de préciser la relation entre les couples et/ou forces exercées sur les actionnaires et les positions, vitesses et accélérations. Le modèle dynamique peut s'écrire sous l'équation :

$$\Gamma = f(q, \dot{q}, \ddot{q}, f_e) \quad (\text{II.34})$$

Avec :

$\Gamma$  : Vecteur des couple/forces des actionneurs pour chaque articulation.

$q$  : Vecteur des positions articulaires.

$\dot{q}$  : Vecteur des vitesses articulaires.

$\ddot{q}$  : Vecteur des accélérations articulaire.

$f_e$  : Vecteur des forces et moments qu'exerce le manipulateur sur l'environnement.

Le modèle dynamique direct et inverse sont des modèles permettent de connaître les couples durant le mouvement du robot et les forces exercées sur le robot, nous pouvons les voir dans les travaux [Khalil et Dombre,2002] et [Moberg, 2007].

### 2.5.1 Modèle dynamique direct

Le modèle dynamique direct (MDD) est une relation mathématique qui exprime les accélérations articulaires en fonction des positions, vitesses et couples des articulations, il est décrit par la relation II.35 :

$$\ddot{q} = M^{-1}(q)(\tau - N(q, \dot{q}) - G(q) - \tau_f) \quad (\text{II.35})$$

Nous pouvons citer deux méthodes pour obtenir le MDD, la première relative à l'inversion de la matrice d'inertie et l'autre est obtenu sans inversion.

❖ Avec l'inversion de la matrice d'inertie du robot :

Nous avons l'équation :

$$\Gamma = A(q)\ddot{q} + C(q, \dot{q})\dot{q} + Q(q) \quad (\text{II.36})$$

Avec :

$A(q)$  : est la matrice d'inertie cinétique (ou matrice d'inertie) qui est symétrique est définie positive. Ses éléments sont fonction des paramètres articulaires  $q$ .

Le modèle dynamique direct se donne par :

$$\ddot{q} = A^{-1}[\Gamma - H(q, \dot{q})] \quad (\text{II.37})$$

Avec :

$$H(q, \dot{q}) = C(q, \dot{q})\dot{q} + Q + J^T f_{en} \quad (\text{II.38})$$

❖ Sans inversion de la matrice d'inertie :

Le calcul du modèle dynamique direct sans inversion par l'utilisation d'équations de Newton-Euler [Armstrong, 1979], [Featherstone, 1983b] et [brandl, 1986].

La résolution de la matrice  $\mathbf{A}$  par une méthode lagrangienne peut être moins couteuse en temps [Megahed, 1982] et [Renaud, 1985]. Nous pouvons cependant utiliser l'algorithme de Newton-Euler pour déterminer ses éléments [Walker, 1982], en remarquant à partir de la formule (II.36), que la  $i^{\text{ème}}$  colonne est égal à  $\Gamma$  si :

$$\dot{q} = 0, \quad g = 0, \quad \ddot{q} = u_i, \quad f_{en} = 0, \quad m_{en} = 0$$

Ou  $u_i$  le vecteur de dimension  $(nx1)$ , leur éléments sont nuls sauf le  $i^{\text{ème}}$  composant qui est égal à un.

## 2.5.2 Modèle dynamique inverse

Le modèle dynamique inverse (MDI) représente les couples appliqués par les actionneurs en fonction des positions, vitesses et accélérations des articulations il est donné par la formule :

$$\Gamma = f(q, \dot{q}, \ddot{q}) \tag{II.39}$$

Il existe plusieurs formalismes pour résoudre le MDI, le plus abordable, Le formalisme de Lagrange-Euler et formalisme de Newton-Euler.

Nous pouvons faire la modélisation d'un bras manipulateur de dynamique rigide à chaîne ouverte simple à deux degrés de liberté et en tenant compte des effets de la vitesse, nous avons choisi le formalisme de Lagrange pour calculer les équations du mouvement pour un bras manipulateur constitue de deux degrés de liberté à chaîne ouverte simple.

### 2.5.2.1 Formalisme de Lagrange

La formule de Lagrange permet d'obtenir directement les relations entre le mouvement en termes de procédure et d'énergie du robot. La forme générale clarifie le mouvement d'un robot manipulateur à  $n$  degrés de liberté et de  $n$  articulations, l'équation de méthode de Lagrange est donnée par :

$$\Gamma_i = \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}_i} \right) - \frac{\partial L}{\partial \theta_i} \tag{II.40}$$

Avec :  $i = 1, 2, \dots, n.$

$L$  : Fonction lagrangienne de robot et leur fonction mathématique.

$$L = E_c - E_p \quad (\text{II.41})$$

$E_c$  : Energie cinétique totale du robot manipulateur.

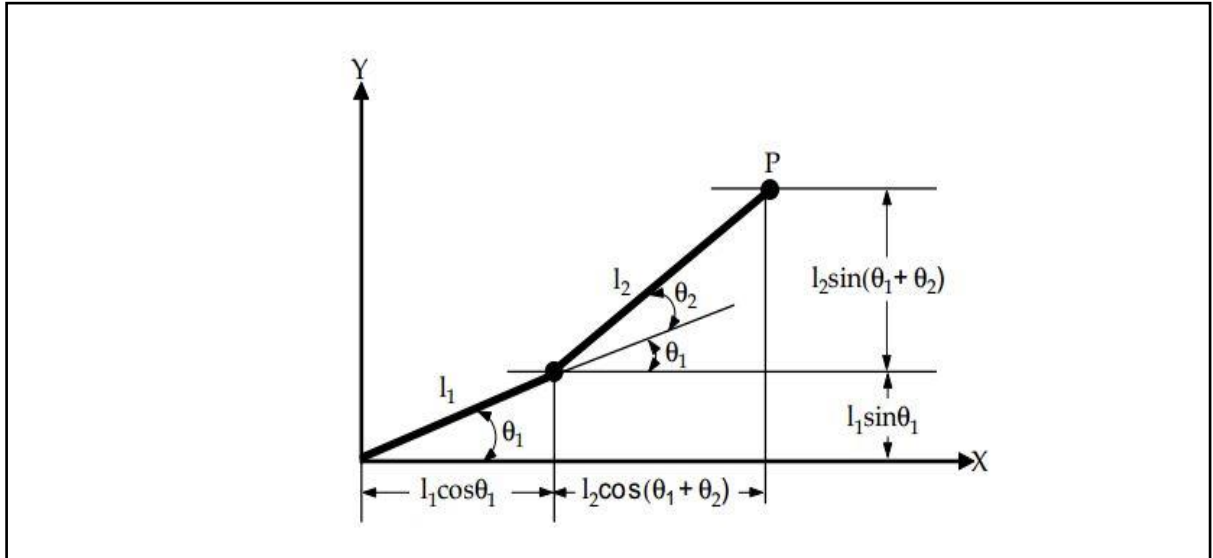
$E_p$  : Energie potentielle totale du robot manipulateur.

Dans le cas d'une articulation rotoïde  $q_i \equiv \theta_i$ , tandis que pour une articulation prismatique  $q_i \equiv d_i$ .

Dans ce travail, l'étude sur un robot manipulateur à 2ddl, caractérisé par un mouvement des articulations rotoïde c'est-à-dire  $q_i \equiv \theta_i$ . Le robot est représenté par la figure (II-3).

Donc :

$$q = [\theta_1 \quad \theta_2]^T \text{ et } \Gamma_i = [\tau_1 \quad \tau_2]^T \quad (\text{II.42})$$



**Figure II.4 :** Structure de robot à deux degrés de liberté

Nous commençons par le calcul d'énergie cinétique  $E_c$  :

$$E_c = \frac{1}{2} I_1 \dot{\theta}_1^2 + \frac{1}{2} I_2 \dot{\theta}_2^2 + \frac{1}{2} m_1 v_1^2 + \frac{1}{2} m_2 v_2^2 \quad (\text{II.42})$$

Ce qui donne :

$$E_c = \frac{1}{2} I_1 \dot{\theta}_1^2 + \frac{1}{2} I_2 \dot{\theta}_1^2 + \frac{1}{2} I_2 \dot{\theta}_2^2 + I_2 \dot{\theta}_1 \dot{\theta}_2 + \frac{1}{2} m_1 v_1^2 + \frac{1}{2} m_2 v_2^2 \quad (\text{II.43})$$

Avec :

$$\begin{cases} v_1^2 = \dot{x}_1^2 + \dot{y}_1^2 = l_{c1}^2 \dot{\theta}_1^2 \\ v_2^2 = \dot{x}_2^2 + \dot{y}_2^2 = l_1^2 \dot{\theta}_1^2 + l_{c2}^2 \dot{\theta}_1^2 + l_{c2}^2 \dot{\theta}_2^2 + 2l_{c2}^2 \dot{\theta}_1 \dot{\theta}_2 + 2l_1 l_{c2} \dot{\theta}_1^2 \cos \theta_2 + 2l_1 l_{c2} \dot{\theta}_1 \dot{\theta}_2 \cos \theta_2 \end{cases} \quad (\text{II.44})$$

A partir de la figure (II.3), nous déterminons les valeurs  $\dot{x}_1^2$  et  $\dot{y}_2^2$  :

$$\begin{cases} x_1 = l_{c1} \cos \theta_1 \\ y_1 = l_{c1} \sin \theta_1 \end{cases} \quad (\text{II.45})$$

$$\begin{cases} x_2 = l_1 \cos \theta_1 + l_{c2} \cos \phi \\ y_2 = l_1 \sin \theta_1 + l_{c2} \sin \phi \end{cases} \quad (\text{II.46})$$

Ce qui donne :

$$\begin{cases} \dot{x}_1 = -l_{c1} \dot{\theta}_1 \sin \theta_1 \\ \dot{y}_1 = l_{c1} \dot{\theta}_1 \cos \theta_1 \end{cases} \quad (\text{II.47})$$

$$\begin{cases} \dot{x}_1^2 = l_{c1}^2 \dot{\theta}_1^2 \sin^2 \theta_1 \\ \dot{y}_1^2 = l_{c1}^2 \dot{\theta}_1^2 \cos^2 \theta_1 \end{cases} \quad (\text{II.48})$$

$$\begin{cases} \dot{x}_2 = -(l_1 \dot{\theta}_1 \sin \theta_1 + l_{c2} \dot{\phi} \sin \phi) \\ \dot{y}_2 = l_1 \dot{\theta}_1 \cos \theta_1 + l_{c2} \dot{\phi} \cos \phi \end{cases} \quad (\text{II.49})$$

$$\begin{cases} \dot{x}_2^2 = l_1^2 \dot{\theta}_1^2 \sin^2 \theta_1 + l_{c2}^2 \dot{\phi}^2 \sin^2 \phi + 2l_1 l_{c2} \dot{\theta}_1 \dot{\phi} \sin \phi \sin \theta_1 \\ \dot{y}_2^2 = l_1^2 \dot{\theta}_1^2 \cos^2 \theta_1 + l_{c2}^2 \dot{\phi}^2 \cos^2 \phi + 2l_1 l_{c2} \dot{\theta}_1 \dot{\phi} \cos \phi \cos \theta_1 \end{cases} \quad (\text{II.50})$$

L'équation (II.26) devient :

$$\begin{aligned} E_c = & \frac{1}{2} I_1 \dot{\theta}_1^2 + \frac{1}{2} I_2 \dot{\theta}_1^2 + \frac{1}{2} I_2 \dot{\theta}_2^2 + I_2 \dot{\theta}_1 \dot{\theta}_2 + \frac{1}{2} m_1 l_{c1}^2 \dot{\theta}_1^2 \\ & + \frac{1}{2} m_2 \left( l_1^2 \dot{\theta}_1^2 + l_{c2}^2 \dot{\theta}_2^2 + l_{c2}^2 \dot{\theta}_1^2 + 2l_{c2}^2 \dot{\theta}_1 \dot{\theta}_2 + 2l_1 l_{c2} \dot{\theta}_1^2 \cos \theta_2 + 2l_1 l_{c2} \dot{\theta}_1 \dot{\theta}_2 \cos \theta_2 \right) \end{aligned} \quad (\text{II.51})$$



La simplification d'équation d'énergie potentielle  $E_p$  donne :

$$E_p = m_1 g h_1 + m_2 g h_2 \quad (\text{II.52})$$

Avec :

$$\begin{cases} h_1 = y_1 = l_{c1} \sin \theta_1 \\ h_2 = y_2 = l_1 \sin \theta_1 + l_{c2} \sin(\theta_1 + \theta_2) \end{cases} \quad (\text{II.53})$$

L'équation (II.28) devient :

$$E_p = m_1 g l_{c1} \sin \theta_1 + m_2 g (l_1 \sin \theta_1 + l_{c2} \sin(\theta_1 + \theta_2)) \quad (\text{II.54})$$

Donc la formule  $L = E_c - E_p$  sera :

$$\begin{aligned} L = & \frac{1}{2} I_1 \dot{\theta}_1^2 + \frac{1}{2} I_2 \dot{\theta}_1^2 + \frac{1}{2} I_2 \dot{\theta}_2^2 + I_2 \dot{\theta}_1 \dot{\theta}_2 + \frac{1}{2} m_1 l_{c1}^2 \dot{\theta}_1^2 - m_1 g l_{c1} \sin \theta_1 \\ & - m_2 g (l_1 \sin \theta_1 + l_{c2} \sin(\theta_1 + \theta_2)) \\ & + \frac{1}{2} m_2 (l_1^2 \dot{\theta}_1^2 + l_{c2}^2 \dot{\theta}_2^2 + l_{c2}^2 \dot{\theta}_1^2 + 2l_{c2}^2 \dot{\theta}_1 \dot{\theta}_2 + 2l_1 l_{c2} \dot{\theta}_1^2 \cos \theta_2 + 2l_1 l_{c2} \dot{\theta}_1 \dot{\theta}_2 \cos \theta_2) \end{aligned} \quad (\text{II.55})$$

Nous appliquons maintenant l'équation d'Euler-Lagrange donnée par équation (II.21) pour obtenir le modèle dynamique inverse de robot manipulateur à 2ddl, soit :

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}_i} \right) - \frac{\partial L}{\partial \theta_i} = \Gamma_i \quad i=1,2 \quad (\text{II.56})$$

On a :

$$\frac{\partial L}{\partial \theta_1} = -m_1 g (l_{c1} \cos \theta_1) - m_2 g (l_1 \cos \theta_1 + l_{c2} \cos(\theta_1 + \theta_2)) \quad (\text{II.57})$$

$$\begin{aligned} \frac{\partial L}{\partial \dot{\theta}_1} = & I_1 \dot{\theta}_1 + I_2 \dot{\theta}_1 + I_2 \dot{\theta}_2 + m_1 l_{c1}^2 \dot{\theta}_1 + m_2 (l_1^2 \dot{\theta}_1 + l_{c2}^2 \dot{\theta}_1 + l_{c2}^2 \dot{\theta}_2 \\ & + 2l_1 l_{c2} \dot{\theta}_1 \cos \theta_2 + l_1 l_{c2} \dot{\theta}_2 \cos \theta_2) \end{aligned} \quad (\text{II.58})$$

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}_1} \right) = I_1 \ddot{\theta}_1 + I_2 \ddot{\theta}_1 + I_2 \ddot{\theta}_2 + m_1 l_{c1}^2 \ddot{\theta}_1 + m_2 (l_1^2 \ddot{\theta}_1 + l_{c2}^2 \ddot{\theta}_1 + l_{c2}^2 \ddot{\theta}_2 + 2l_1 l_{c2} \ddot{\theta}_1 \cos \theta_2 - 2l_1 l_{c2} \dot{\theta}_1 \dot{\theta}_2 \sin \theta_2 + l_1 l_{c2} \ddot{\theta}_2 \cos \theta_2 - l_1 l_{c2} \dot{\theta}_2^2 \sin \theta_2) \quad (\text{II.59})$$

Ce qui donne :

$$\begin{aligned} \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}_1} \right) - \frac{\partial L}{\partial \theta_1} &= (I_1 + I_2 + m_1 l_{c1}^2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos \theta_2)) \ddot{\theta}_1 \\ &\quad + (I_2 + m_2 (l_{c2}^2 + l_1 l_{c2} \cos \theta_2)) \ddot{\theta}_2 + (-m_2 l_1 l_{c2} \sin \theta_2) \dot{\theta}_2^2 \\ &\quad + 2(-m_2 l_1 l_{c2} \sin \theta_2) \dot{\theta}_1 \dot{\theta}_2 + m_1 g l_{c1} \cos \theta_1 + m_2 g (l_1 \cos \theta_1 + l_{c2} \cos(\theta_1 + \theta_2)) \end{aligned} \quad (\text{II.60})$$

On a aussi :

$$\frac{\partial L}{\partial \theta_2} = -m_2 (l_1 l_{c2} \dot{\theta}_1^2 \sin \theta_2 + l_1 l_{c2} \dot{\theta}_1 \dot{\theta}_2 \sin \theta_2) - m_2 g l_{c2} \cos(\theta_1 + \theta_2) \quad (\text{II.61})$$

$$\frac{\partial L}{\partial \dot{\theta}_2} = I_2 \dot{\theta}_2 + I_2 \dot{\theta}_1 + m_2 (l_{c2}^2 \dot{\theta}_2 + l_{c2}^2 \dot{\theta}_1 + l_1 l_{c2} \dot{\theta}_1 \cos \theta_2) \quad (\text{II.62})$$

$$\frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}_2} \right) = I_2 \ddot{\theta}_2 + I_2 \ddot{\theta}_1 + m_2 (l_{c2}^2 \ddot{\theta}_2 + l_{c2}^2 \ddot{\theta}_1 + l_1 l_{c2} \ddot{\theta}_1 \cos \theta_2 - l_1 l_{c2} \dot{\theta}_1 \dot{\theta}_2 \sin \theta_2) \quad (\text{II.63})$$

Ce qui donne :

$$\begin{aligned} \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}_2} \right) - \frac{\partial L}{\partial \theta_2} &= (I_2 + m_2 (l_{c2}^2 + l_1 l_{c2} \cos \theta_2)) \ddot{\theta}_1 + (I_2 + m_2 l_{c2}^2) \ddot{\theta}_2 + (m_2 l_1 l_{c2} \sin \theta_2) \dot{\theta}_1^2 \\ &\quad + m_2 g l_{c2} \cos(\theta_1 + \theta_2) \end{aligned} \quad (\text{II.64})$$

A partir de l'équation (II.21), nous obtenons l'équation II.65 :

$$\begin{cases} \tau_1 = \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}_1} \right) - \frac{\partial L}{\partial \theta_1} \\ \tau_2 = \frac{d}{dt} \left( \frac{\partial L}{\partial \dot{\theta}_2} \right) - \frac{\partial L}{\partial \theta_2} \end{cases} \quad (\text{II.65})$$

Le mouvement des articulations constituant le robot manipulateur est rotoïde, Donc l'expression du couple peut être donnée sous la forme matricielle :

$$\Gamma_i = M(\theta)\ddot{\theta} + C(\theta, \dot{\theta})\dot{\theta} + G(\theta), \quad (i = 1, 2) \quad (\text{II.66})$$

Avec :

$\Gamma$  : Vecteur des couples ou des forces généralisés.

$\theta$  : Vecteur des variables articulaires du bras manipulateur.

$\dot{\theta}$  : Vecteur des vitesses articulaires.

$\ddot{\theta}$  : Vecteur des accélérations articulaires.

$M(\theta)$  : Matrice d'inertie de dimension  $(n \times n)$ .

$C(\theta, \dot{\theta})\dot{\theta}$  : Vecteur des forces centrifuges et de Coriolis.

$G(\theta)$  : Vecteur de force de gravité de la charge.

Suivant l'équation (II.40) nous obtenons la formule matricielle :

$$\begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} = \begin{bmatrix} M_{11} & M_{12} \\ M_{21} & M_{22} \end{bmatrix} \begin{bmatrix} \ddot{\theta}_1 \\ \ddot{\theta}_2 \end{bmatrix} + \begin{bmatrix} 2h_{122}\dot{\theta}_2 & h_{211}\dot{\theta}_2 \\ h_{211}\dot{\theta}_1 & 0 \end{bmatrix} \begin{bmatrix} \dot{\theta}_1 \\ \dot{\theta}_2 \end{bmatrix} + \begin{bmatrix} g_1 \\ g_2 \end{bmatrix} \quad (\text{II.67})$$

Ce qui donne :

$$\begin{cases} \tau_1 = M_{11}\ddot{\theta}_1 + M_{12}\ddot{\theta}_2 + h_{122}\dot{\theta}_2^2 + 2h_{112}\dot{\theta}_1\dot{\theta}_2 + g_1 \\ \tau_2 = M_{21}\ddot{\theta}_1 + M_{22}\ddot{\theta}_2 + h_{211}\dot{\theta}_1^2 + g_2 \end{cases} \quad (\text{II.68})$$

L'observation d'équation (II.43) et équation (II.40) peut définir les éléments suivants:

$$\begin{cases} M_{11} = I_1 + I_2 + m_1 l_{c1}^2 + m_2(l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos \theta_2) \\ M_{12} = M_{21} = I_2 + m_2(l_{c2}^2 + l_1 l_{c2} \cos \theta_2) \\ M_{22} = I_2 + m_2 l_{c2}^2 \end{cases} \quad (\text{II.69})$$

$$h_{122} = h_{112} = -h_{211} = -m_2 l_1 l_{c2} \sin \theta_2 \quad (\text{II.70})$$

$$\begin{cases} g_1 = m_1 g (l_{c1} \cos \theta_1) + m_2 g (l_1 \cos \theta_1 + l_{c2} \cos(\theta_1 + \theta_2)) \\ g_2 = m_2 g l_{c2} \cos(\theta_1 + \theta_2) \end{cases} \quad (\text{II.71})$$

La matrice d'inertie M est égale à :

$$M(\theta) = \begin{bmatrix} I_1 + I_2 + m_1 l_{c1}^2 + m_2 (l_1^2 + l_{c2}^2 + 2l_1 l_{c2} \cos \theta_2) & I_2 + m_2 (l_{c2}^2 + l_1 l_{c2} \cos \theta_2) \\ I_2 + m_2 (l_{c2}^2 + l_1 l_{c2} \cos \theta_2) & I_2 + m_2 l_{c2}^2 \end{bmatrix} \quad (\text{II.72})$$

Le vecteur de termes Coriolis et centrifuge est donnée par :

$$c(\theta, \dot{\theta})\dot{\theta} = \begin{bmatrix} -2m_2 l_1 l_{c2} \sin \theta_2 \dot{\theta}_1 \dot{\theta}_2 - m_2 l_1 l_{c2} \sin \theta_2 \dot{\theta}_2^2 \\ m_2 l_1 l_{c2} \sin \theta_2 \dot{\theta}_1^2 \end{bmatrix} \quad (\text{II.73})$$

Le vecteur des termes de forces de gravité G est donné par l'équation suivante:

$$G(\theta) = \begin{bmatrix} m_1 g l_{c1} \cos \theta_1 + m_2 g l_1 \cos \theta_1 + m_2 g l_{c2} \cos(\theta_1 + \theta_2) \\ m_2 g l_{c2} \cos(\theta_1 + \theta_2) \end{bmatrix} \quad (\text{II.74})$$

## 2.6 Commande d'un robot manipulateur

La commande généralement est une théorie en robotique nécessaire pour assurer la réalisation de la tâche demandée, Pour cela, différents schémas, issus de diverses méthodes de modélisation du système peuvent être utilisées, Les méthodes de commande les plus utilisables sont : linéaires ou non-linéaires, statiques ou dynamiques, par rétroaction (feedback) ou anticipation (feedforward), robustes ou adaptatives.

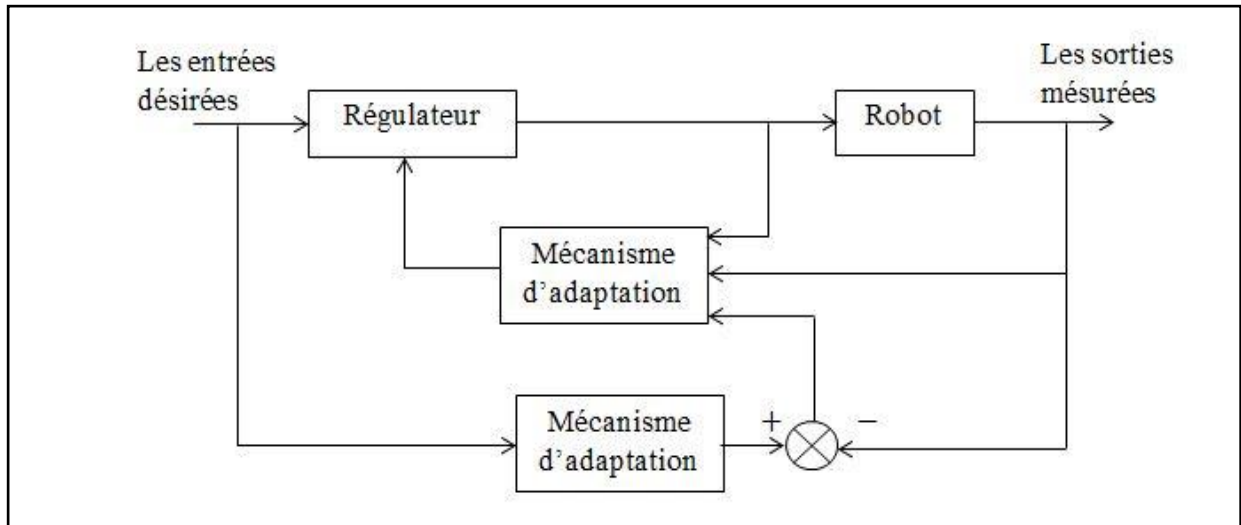
Nous expliquons ici, deux stratégies de commande, la première est commande adaptative et la deuxième est la commande dynamique.

### 2.6.1 Commande adaptative

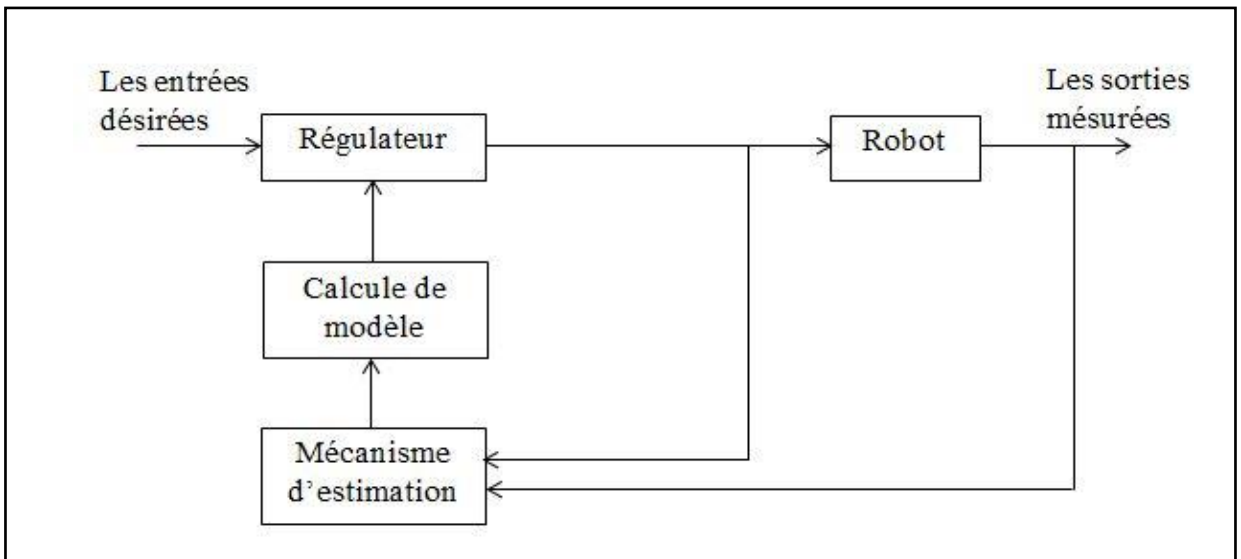
L'objectif de la commande adaptative est l'estimation des paramètres du robot quand ils sont inconnus et/ou ont des variations imprévisibles durant le mouvement de robot manipulateur. Généralement la théorie de la commande adaptative est employée pour estimer ou ajuster en ligne les valeurs de paramètres dynamiques. Deux types sont largement appliquées, celle de la commande adaptative avec modèle de référence qui repose sur l'adaptation des gains de retour pour que le

robot suive le mouvement désiré, le modèle de référence étant calculé à priori, et l'autre la commande auto-adaptative repose sur la commande adaptative indirecte qui consiste à identifier les paramètre du modèle qui sera donc ajusté à tout instant [Ghorbel et al., 1989], [Lozano et Brogliato, 1992] et [Chien et Huang, 2007].

La figure II.5 et figure II.6 représente respectivement, la commande adaptative avec modèle de référence et la commande auto-adaptative :



**Figure II.5 :** Commande adaptative avec modèle de référence



**Figure II.6 :** Commande auto-adaptative

## 2.6.2 Commande dynamique

Le but de commande dynamique du robot manipulateur, est de calculer les torseurs de forces à appliquer à chaque articulation de robot manipulateur, afin d'assurer le découplage et la linéarisation des équations du modèle, ce qui se traduit par une réponse uniforme quelle que soit la configuration du robot, c'est-à-dire, la précision dynamique [Dombre et Khalil, 1988].

## 3.6.3 Commande PD

La commande la plus simple d'asservir un robot manipulateur est bien entendu d'utiliser un correcteur linéaire de type PD (proportionnel dérivateur) qui génère à partir de l'erreur entre la consigne et la mesure, une consigne pour l'actionneur, Le contrôleur PD est largement utilisée pour la commande des robots manipulateurs industriels en raison de sa simplicité, notamment le travail de Tomei [Tomei, 1991]. La commande PD assure la stabilité asymptotique globale de la dynamique du robot en boucle fermée mais en utilisant des gains plus grands [Canudas et al. 1992] et [Lewis et al. 1993].

La dynamique de robot manipulateur représenté par l'équation :

$$\tau = M(q)\ddot{q} + N(q, \dot{q}) \quad (\text{II.75})$$

Avec :

$N(\theta, \dot{\theta})$  : Présente Les limites non linéaires, il est donné par :

$$N(q, \dot{q}) = V(q, \dot{q})\dot{q} + F_v(\dot{q}) + F_d(\dot{q}) + G(q) \quad (\text{II.76})$$

$$F(\dot{q}) = F_v(\dot{q}) + F_d(\dot{q}) \quad (\text{II.77})$$

$F_v(\dot{q})$  : est le vecteur  $n \times 1$  des coefficients de frottement visqueux et  $F_d(\dot{q})$  : est le vecteur  $n \times 1$  des coefficients de frottement dynamique, ou  $n$  correspond au nombre de degrés de liberté du robot.

$q(t) \in R^n$  : Variable articulaire,  $q$  et  $\dot{q}$  sont respectivement les vecteurs  $n \times 1$  de la position et de la vitesse communes.

$\tau(t)$  : est le vecteur  $n \times 1$  des couples appliqués par le bras. Si cette équation comprend la dynamique de l'actionneur à moteur, alors  $\tau(t)$  est une tension d'entrée.

$M(q)$  est la matrice d'inertie  $n \times n$  bien déterminée positive et symétrique,  $V(\theta, \dot{\theta})\dot{\theta}$  est le vecteur des forces centrifuges et de Coriolis,  $G(q)$  est le vecteur  $n \times 1$  de forces de la gravité, Des

perturbations inconnues bornées (incluant les dynamiques non structurées, et non modélisées) représentées par  $\tau_d$ .

L'observation de trajectoire suivant la variable articulation est calculée par la relation :

$$e(t) = q_d(t) - q(t) \quad (\text{II.78})$$

$$\dot{e}(t) = \dot{q}_d(t) - \dot{q}(t) \quad (\text{II.79})$$

Avec :

$e(t)$  : Vecteur de l'erreur de position.

$\dot{e}(t)$  : Vecteur de l'erreur de vitesse.

Nous choisissons une commande qui stabilise le system, de sorte que les erreurs tendent vers zéro pour assurer la précision.

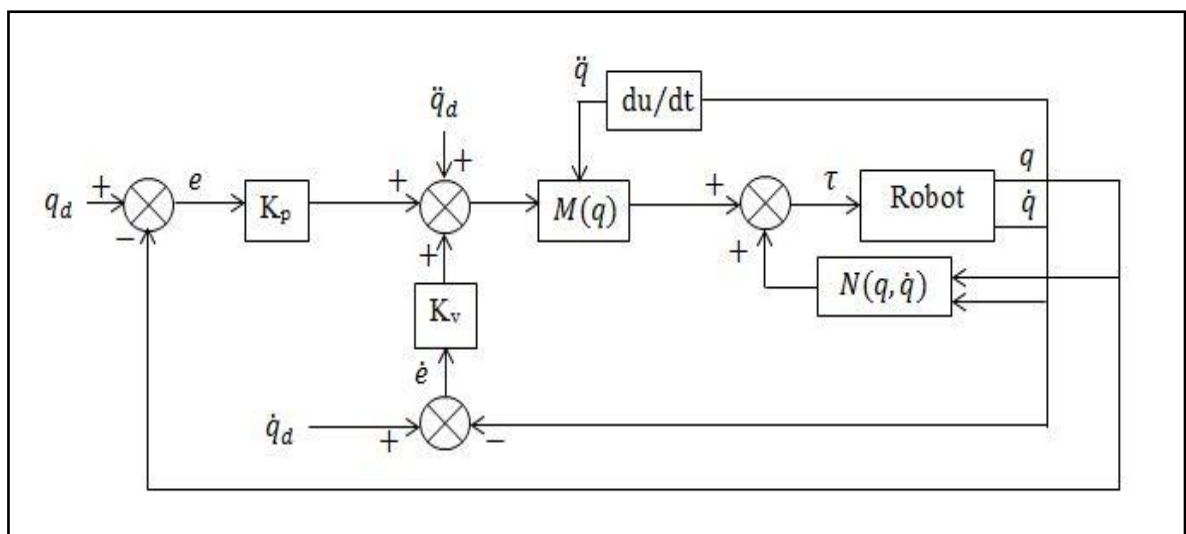
La fonction d'entrée de commande avec un contrôleur proportionnel dérivateur PD est donnée par:

$$\mu(t) = K_p e(t) + K_v \dot{e}(t) \quad (\text{II.80})$$

Donc, La loi de commande de couple (PD) est donnée par :

$$\tau = M(q)(\ddot{q}_d + K_v \dot{e} + K_p e) + N(q, \dot{q}) \quad (\text{II.81})$$

La commande par calcul de couple avec PD est illustrée sur la figure II.7 :



**Figure II.7 :** La commande dynamique avec PD

### 2.6.4 Commande PID

La stratégie de commande PID consiste de trouver à partir des équations des erreurs articulaire la couple, Dans ce cas, les positions et vitesses articulaires désirées sont comparées avec les positions et vitesses obtenues de robot manipulateur [Choi et al., 2001]. La commande PID peut être utilisée, pour résoudre le problème des gains plus hauts de la commande PD. C'est-à-dire, l'intégration peut ajouter à la commande PD pour agir sur les forces de gravité qui peuvent être considérées, dans certaines mesures, comme des perturbations constantes [Canudas et al. 1992] et [Lewis et al. 1993].

Nous avons :

$$\varepsilon(t) = \int e(t) dt \quad (\text{II.82})$$

Avec:

$\varepsilon(t)$  : Vecteur de l'intégrale erreur de position.

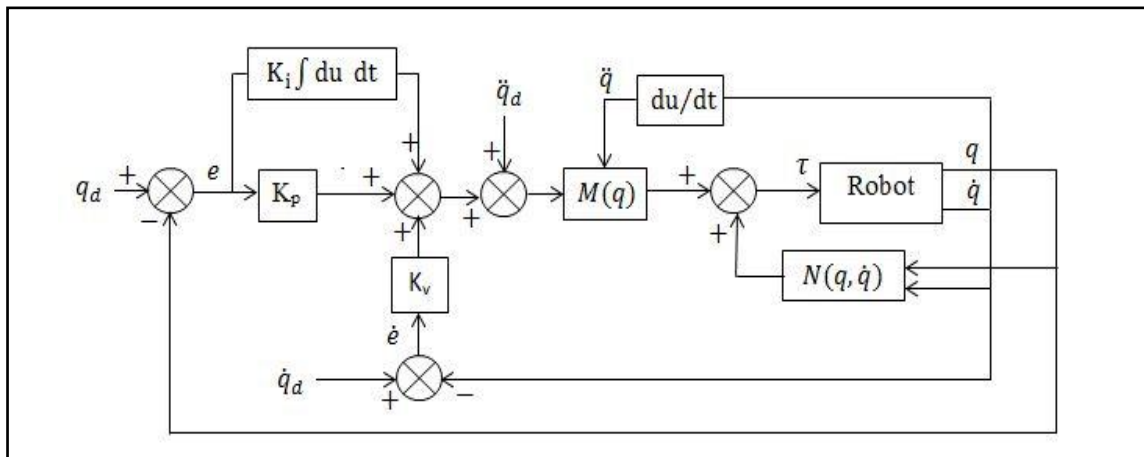
La fonction d'entrée de commande avec un contrôleur proportionnel intégrateur dérivateur PID est de la forme :

$$\mu(t) = K_p e(t) + K_v \dot{e}(t) + K_i \int e(t) dt \quad (\text{II.83})$$

La loi de commande par calcul de couple (PID) sera :

$$\tau = M(q)(\ddot{q}_d + k_v \dot{e} + k_p e + k_i \varepsilon) + N(q, \dot{q}) \quad (\text{II.84})$$

La figure II.8 représente la commande avec proportionnel intégrateur dérivateur PID :



**Figure II.8** : La commande dynamique avec PID



## **2.7 Conclusion**

Ce chapitre présente les différents modèles de robot manipulateur à deux degrés de liberté dans son environnement pour extraire les modèles géométriques direct et inverse, les modèles cinématiques direct et inverse et les modèles dynamiques, et comprendre les mouvements des articulations d'un bras manipulateur, la modélisation porte sur la position cartésienne, les orientations cartésiennes et position articulaire afin de donner modèle géométrique, la connaissance de vitesses cartésiennes, positions et des vitesses articulaires pour donner le modèle cinématique, et le calcul d'accélération articulaires positions, vitesses et couples des articulations pour représenter le modèle dynamique. Généralement le but est de donner les équations mathématiques qui permettent de simuler, commander et exécuter les tâches désirées. Les modèles proposés dans ce chapitre vont être utilisés et validés dans notre simulateur concernant la commande et supervision appliquées à ces types de robots manipulateurs.

Dans le chapitre suivant, nous présentons les techniques d'intelligence artificielle et les communications via internet.

## **CHAPITRE 3**

# **Les techniques de l'intelligence artificielle**

### 3.1 Introduction

Les outils d'intelligence artificielle sont des techniques permettant l'optimisation d'analyse de processus industriels. Leurs champs d'application sont très vastes, afin d'atteindre les buts d'automatisation (coût, émission, pertes de puissance..), la modélisation de système industriel, et bien d'autres domaines. La raison de ce grand nombre d'application est claire, simple et l'efficace. Les réseaux neurones et la logique floue ne donnent pas toujours les meilleures performances sachant que la mise en œuvre d'application exige une connaissance précise du système industriel. Ces limitations motivent les chercheurs à développer une nouvelle procédure intelligente et adaptative pour améliorer le rendement des systèmes non linéaires. Généralement les techniques d'hybridation donnent des résultats très intéressants dans le cadre de l'étude des systèmes non linéaire où il est difficile d'établir un modèle mathématique.

### 3.2 Logique Floue

La logique floue est une méthodologie mathématique permettant de prendre en compte toutes les sortes de connaissances qualitatives de concepteurs et d'opérateurs dans l'automatisation des systèmes. Les systèmes flous ont été utilisées pour les systèmes complexes non linéaires [Takagi 1985], le traitement d'images pour améliorer les caractéristiques d'images numériques couleur (luminosité, teinte, brillance) [Asgharzadeh 1998], la robotique mobile [Pagès 1998]. Généralement la logique floue s'y développe car il s'agit d'une approche efficace et générique.

Aussi La théorie des ensembles flous fournit une méthode appropriée, facilement et réalisable dans des applications temps réel. Ces caractéristiques rendent la logique floue abordable par les chercheurs avec une continuité d'applications.

Nous présentons les principaux travaux et la diffusion de cette méthode.

- En 1965 le concept introduit par Lotfi Zadeh "Fuzzy set theory". Définition des ensembles flous et opérateurs associés.
- En 1970 l'application des théories de la logique flou en médecine, commerce,...etc.
- En 1974 Mamdani expérimental la théorie énoncée par Zadeh sur la régulation floue d'une chaudière à vapeur.
- En 1985 les Japonais introduisent des produits grand public "Fuzzy Logic Inside".
- En 1990 la généralisation d'utilisation de cette technique.

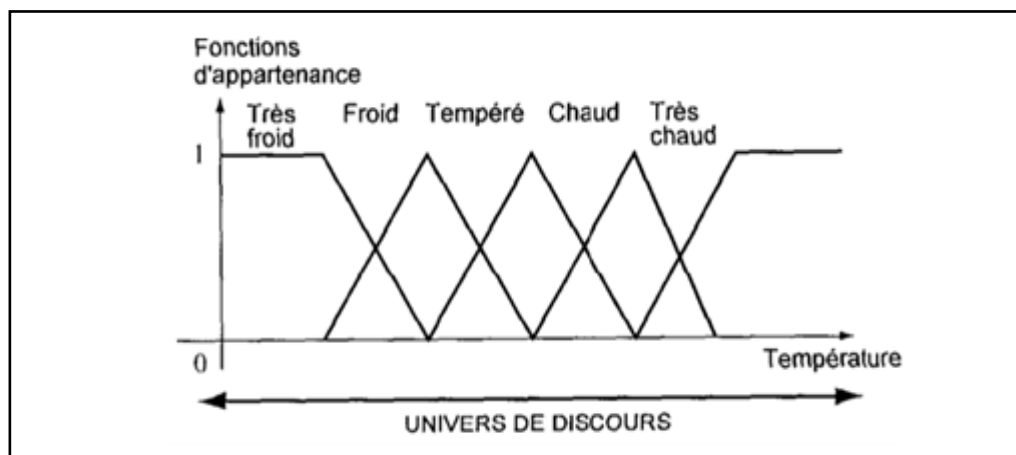
Les systèmes flous peuvent être considérés comme des systèmes logiques qui basées sur règles linguistiques et l'utilisation des variables booléennes qui ne peuvent prendre que deux valeurs (0 ou 1) pour mettre des relations mathématique. Dans ce contexte les concepts de base de la logique floue sont les ensembles flous, variables flous, la décision à partir d'une base de règles *SI...ALORS...* et l'inférence floue. Nous avons cité trois classe, les systèmes flous linguistiques ou systèmes de Mamdani et systèmes flous de type Takagi-Sugeno-Kang (TSK) [Kim et al., 1997] et [Wang et Langari, 1996].

### 3.2.1 Ensemble flou

Les ensembles flous sont des extensions de théorie des ensembles classiques d'un système floue. Chaque ensemble flou est caractérisé par sa fonction d'appartenance, qui projette les variables linguistiques de l'univers de discours dans l'intervalle [0,1] [Zadeh, 1965]. Pour un univers de discours X donné le degré d'appartenance est  $\mu : X \rightarrow [0,1]$ , et l'ensemble flou A est défini comme ceci :

$$A = \{(x, \mu(x)) \mid x \in X\}$$

La figure III.1 illustre les concepts cités précédemment.



**Figure III.1 : Fonctions d'appartenances**

Avec :

L'univers du discours est une gamme de température.

Le variable linguistique est la température.

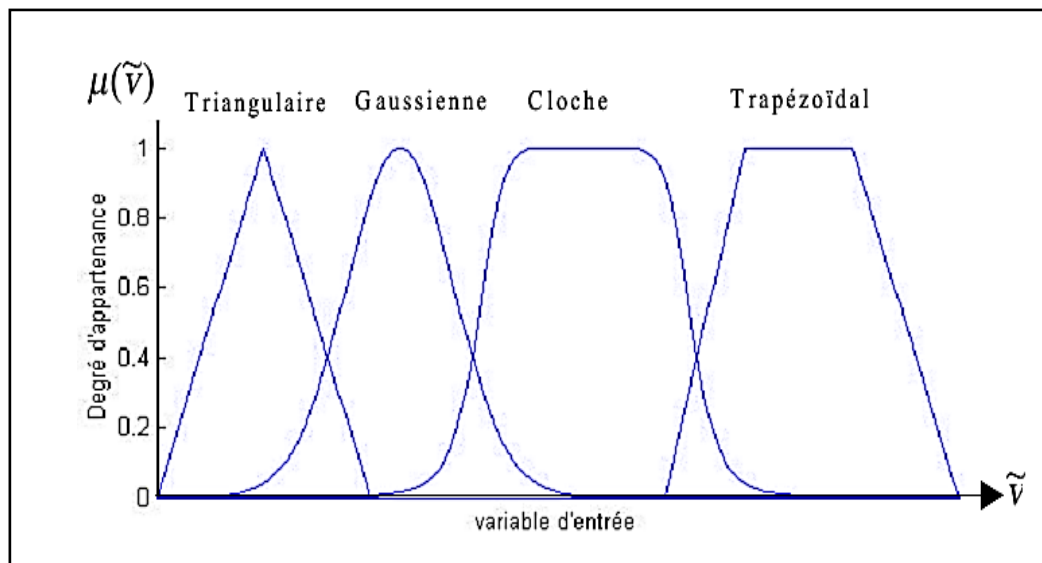
Les valeurs linguistiques sont Très froid, Froid, Tempéré, Chaud et Très Chaud.

### 3.2.2 Fonctions d'appartenance

Le système flou est caractérisé par une fonction mathématique, connu par la fonction d'appartenance  $\mu_E(\tilde{v})$  et des valeurs dans l'intervalle  $[0,1]$  qui permettent de déterminer le degré d'appartenance. Soit le variable  $\tilde{v}$  varie dans un univers de discours, ce dernier est distribué en sous-ensembles flous de façon que dans chaque espace. Ces espaces sont décrites par des fonctions convexes, généralement sous forme triangulaire ou trapézoïdale, elles admettent comme argument la position de la variable  $\tilde{v}$  dans l'univers de discours, et comme sortie le degré d'appartenance de  $\tilde{v}$  au endroit décrite par la fonction notée :

$$\mu_E(\tilde{v}) : \text{Degré d'appartenance de } \tilde{v} \text{ au sous ensemble E.}$$

La figure III.2 représente la répartition des fonctions d'appartenance, leurs chevauchements ainsi que leurs formes [Büh, 94].

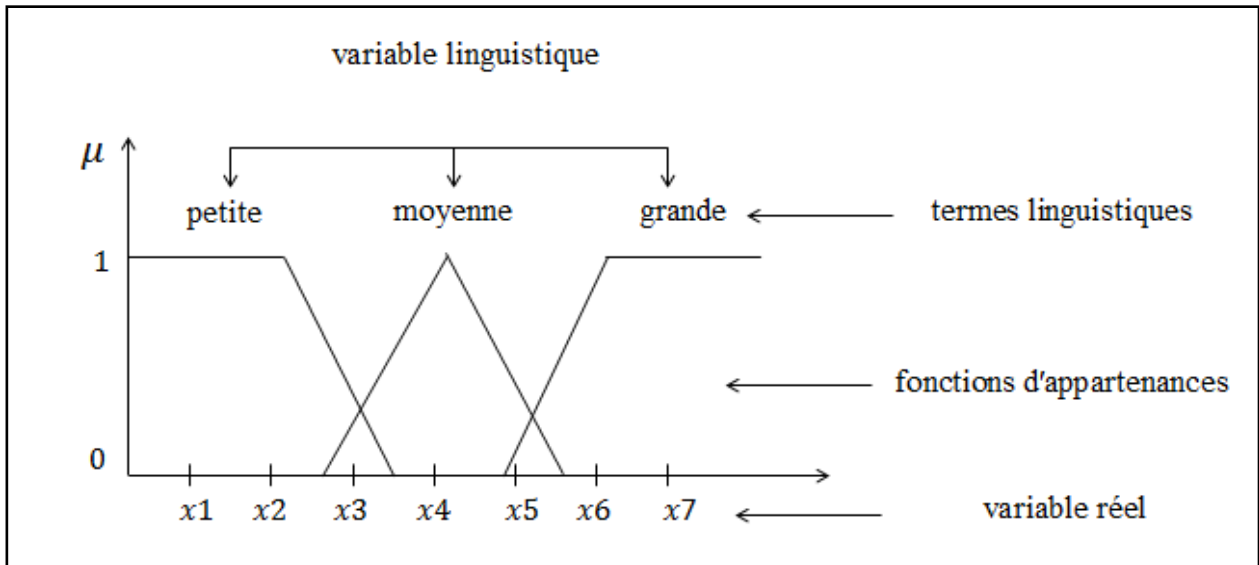


**Figure III.2 :** Exemple de fonctions d'appartenance

### 3.2.3 Variables linguistiques

les variables linguistiques sont des attributs linguistiques qui peuvent être définie à partir de valeurs  $(x, U, T_x)$ , où  $x$  est une variable définie sur l'univers de discours  $U$  et  $T_x = A_1, A_2$ , est un ensemble composé de sous-ensembles flous de  $U$  qui caractérise  $x$  [Pedrycz et Zadeh, 1995].

La figure III.3 présente un exemple de la variable linguistique avec trois termes linguistiques (petite, moyenne et grande) et trois fonctions d'appartenances.



**Figure III.3 :** Les variables linguistiques

### 3.2.4 Règles et opérateurs flous

La sortie d'un système flou dédié par l'utilisation des opérateurs logique tels que : "ET", "OU", et "NON" avec des théories qui sont interprétés par les opérations : "Minimum", "Maximum" et "complément à un" [Klir et Yuan, 1994]. Suit à la théorie de système de la logique flou on définit la réunion, l'intersection, le complément....d'ensembles flous.

- **La réunion**

L'ensemble flou de l'opération OU est un ensemble flou de fonction d'appartenance réalisé par :

$$\mu_{A \cup B}(x) = \max(\mu_A(x), \mu_B(x)) \quad \forall x \in U \quad (\text{III.1})$$

- **L'intersection**

L'ensemble flou de l'opération ET est un ensemble flou de fonction d'appartenance donné par :

$$\mu_{A \cap B}(x) = \min(\mu_A(x), \mu_B(x)) \quad \forall x \in U \quad (\text{III.2})$$

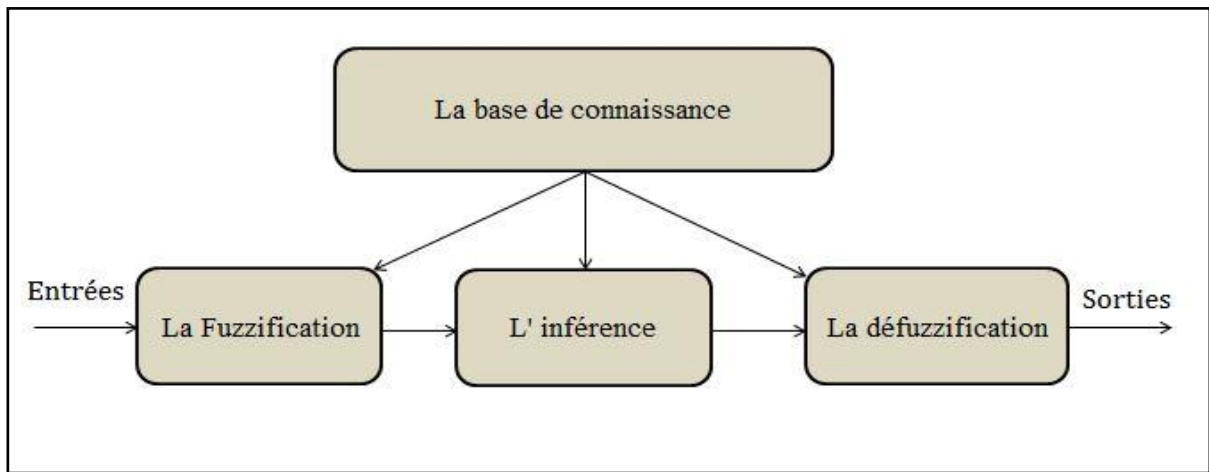
- **Le complément**

L'ensemble flou de l'opération NON est un ensemble flou de fonction d'appartenance représenté par l'équation :

$$\mu_{\bar{A}}(x) = 1 - \mu_A(x) \quad \forall x \in U \quad (\text{III.3})$$

### 3.2.5 Structure de base d'un système flou

Le système flou repose sur quatre partition sont : la base de connaissances, la fuzzification des variables d'entrée, avec éventuellement un prétraitement de l'information, l'inférence à partir d'une base de connaissance et la défuzzification, avec éventuellement un post-traitement de l'information [fig III.4].



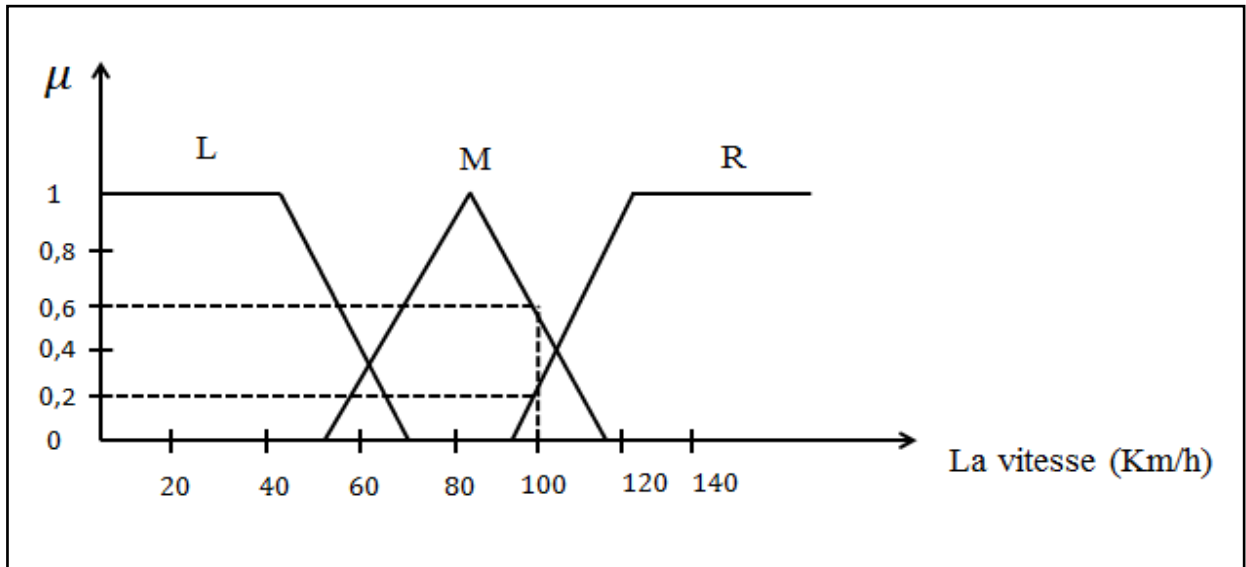
**Figure III.4 :** Structure de base de système flou

#### 3.2.5.1 La base de connaissances

Il s'agit d'un module contenant des indications des -fonctions d'appartenance (formes et paramètres) associées aux variables d'entrée/sortie ainsi que l'ensemble des règles floues (les connaissances de l'expert humain).

#### 3.2.5.2 La fuzzification

La fuzzification est un procédure de transformation des grandeurs physiques (entrées) en valeurs linguistiques qui peuvent être traitées, ensuite, par le mécanisme d'inférence et ce grâce aux fonctions d'appartenances des valeurs linguistiques concernées par la valeur physique. La figure 3.7 présent un exemple de fuzzification.



**Figure III.5 :** Exemple de fuzzification

Avec :

Les variables linguistiques sont : Lente (L), Moyenne (M) et Rapide (R). Et le variable d'entrée est x.

Par exemple si le variable d'entrée x est égale à 100 (Km/h) Après fuzzification l'entrée est égale à M avec un degré 0.6 et est égale aussi à R avec un degré 0.2.

### 3.2.5.3 L'inférence

Cette étape permet de calculer le degré de vérité des différentes règles du système et ajuster chacune de ces règles à une valeur de sortie. La valeur de sortie dépend de la partie conclusion des règles qui peut prendre plusieurs formes. Le mécanisme d'inférence est basé sur plusieurs règles établies par l'expertise et le savoir-faire humain concernant le système à régler. L'inférence floue est une reproduction linguistique et symbolique à base de règles prédéfinies dans la matrice d'inférence. Nous prendrons dans cette partie la forme de règle de Mamdani et Sugeno.

Cette dernière est donnée par la forme :

*Si* (.....) *Alors* Y est B avec :B est ensemble flou.

Et le règle de type Sugeno sous la forme suivant :

*Si* (.....) *Alors*  $y = f(x_1, x_2, \dots, x_n)$  avec :  $x_1, \dots, x_n$  sont des entrées réel.

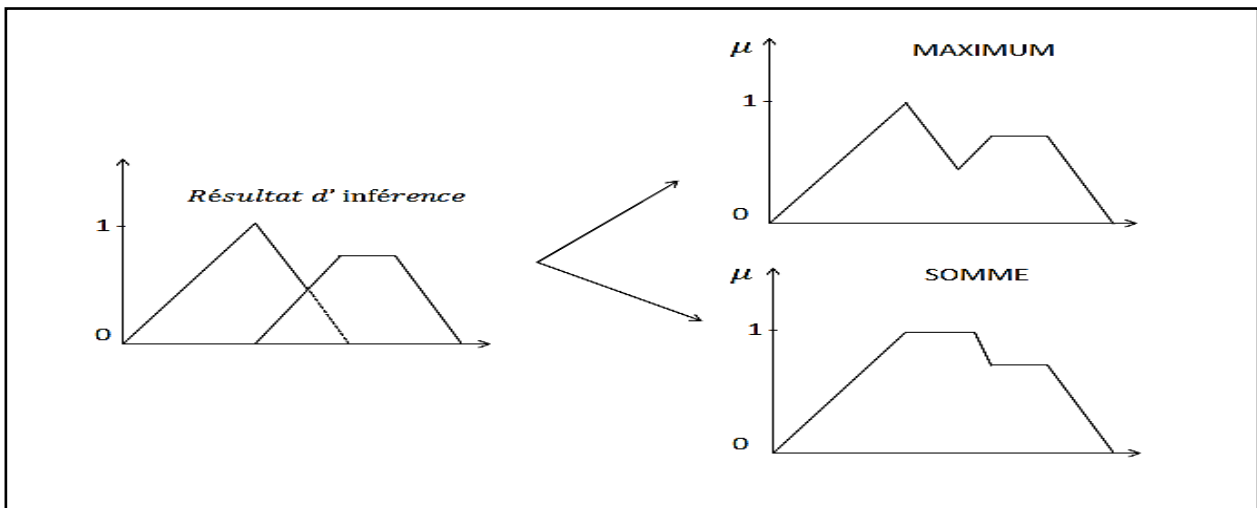


### 3.2.5.4 La défuzzification

La défuzzification consiste à remplacer l'ensemble des valeurs linguistiques de sorties des différentes règles résultant de l'inférence par une valeur réelle de cet ensemble. Cette unité se compose de deux étapes qui sont la composition de règles et la transformation de la valeur symbolique respectivement.

#### - Composition des règles

La méthode de composition est nécessaire pour obtenir un seul ensemble représentatif des différentes conclusions des règles floues. Cette étape connue par agrégation, fait correspondre à chaque règle un degré, qui dépend de sa prémisse. Particulièrement nous pouvons discuter de deux méthodes. L'une relative au MAXIMUM qui donne l'ensemble de sorties par une fonction d'appartenance égale au maximum des fonctions d'appartenance des sous-ensembles flous. Et l'autre dédiée à la SOMME qui consiste à déterminer l'ensemble de sorties par une fonction d'appartenance égale à la somme des fonctions d'appartenance des sous-ensembles flous. La figure III.6 montre la compositions des ensembles flous issus de l'inférence.



**Figure III.6 :** Compositions des ensembles flous issus de l'inférence

#### a) Transformation la valeur symbolique

L'étape de défuzzification dont est l'objectif de générer une valeur numérique à partir de l'ensemble obtenu par composition des règles. Il existe plusieurs méthodes de défuzzification,

notamment les méthodes du centre de gravité et celle de moyenne des maxima [Braae et Rutherford, 1978].

- **Méthode du centre de gravité (COG)**

La méthode de centre de gravité consiste à calculer la valeur de sortie par le calcul de l'abscisse du centre de gravité de la fonction d'appartenance. Cette méthode est réalisée par la relation :

$$y^* = \int y U_r(y) dy / \int U_r(y) dy \quad (\text{III.4})$$

- **La méthode de la moyenne des maxima (MOM)**

La défuzzification par la méthode des moyennes maximums permet de calculer la valeur de sortie par le calcul de l'abscisse du point correspondant au centre de l'intervalle M pour lequel la fonction d'appartenance est maximale. Cette valeur est fournie par l'expression (III.5).

$$y^* = ( \inf(M) + \sup(M) ) / 2 \quad (\text{III.5})$$

où  $\inf(M)$  et  $\sup(M)$  sont respectivement les bornes inférieure et supérieure de l'intervalle M.

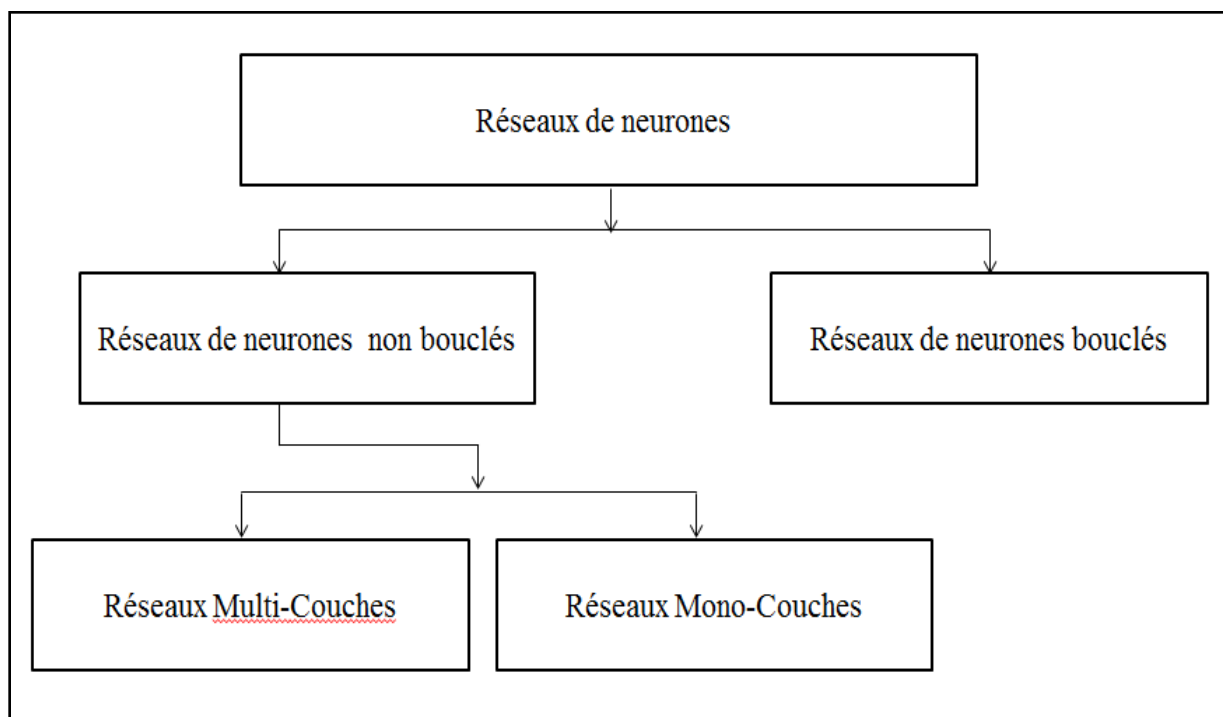
### 3.3 Réseaux de neurones

Les réseaux de neurones artificiels se révèlent être un outil très efficace. Différents exemples dans la littérature mettent en exergue l'utilité et l'efficacité des réseaux de neurones artificiels et surtout l'avenir prometteur quant à leur utilisation. Dans la dernière décennie, ils étaient presque exclusivement un domaine d'entreprises révolutionnaires avec des ressources pratiquement illimitées pour financer leurs recherches, comme Google, Facebook, Netflix, etc. Aujourd'hui, les méthodes d'intelligence artificielle semblent former une partie indivisible de toute entreprise réussie. Certains experts prédisent même que l'intelligence artificielle mènera à la prochaine révolution industrielle.

Les réseaux de neurones ont été proposés en 1943 par W. McCulloch et W. Pitts. Ces techniques sont capables de reproduire des calculs complexes. En effet, plusieurs des travaux de recherche ont montrés que les réseaux de neurones sont des approximateurs universels, tant du côté architecture où plusieurs modèles sont proposés, ce qui permet de modéliser n'importe quel système non linéaire, Ces travaux de recherche ont donné lieu à des applications très intéressantes dans plusieurs domaines.

### 3.3.1 Architecture des réseaux de neurones

Un réseau de neurone est un composant des éléments de traitement d'information de système physique, consacré à une méthode d'interconnexions entre ces éléments et une loi d'apprentissage pour adapter les poids de connexions. La forme de réseaux de neurones est caractérisée par leurs partitions. Ces partitions sont les couches [Burns, 2001].



**Figure III.7 :** Différents types des réseaux de neurones

#### 3.3.1.1 Réseaux de neurones non bouclés (feed forward )

Un réseau de neurone non bouclé se compose par un ensemble de neurones connectés entre eux, et l'opération des entrées vers les sorties sans retour. Ce type de réseau devisé en deux architectures: les réseaux Mono-Couches et les réseaux Multi-Couches.

##### a) Réseaux Mono-Couche

Ce type de réseau se forme par deux couches, une couche recevant les valeurs d'entrée qui applique une traitement de ces valeurs par l'intermédiaire des nœuds et une couche de sortie transmettant les résultats du traitement au milieu extérieur.

## b) Réseaux Multi-Couches

Ces réseaux proactifs comprennent d'une ou de plusieurs couches cachées, dont les neurones de réseau sont des nœuds cachés ou unités cachées qui effectuent le calcul correspondants. Nous pouvons citer les plus utilisables, les réseaux multicouches perceptrons (Multi Layer Perceptron) MLP. Les réseaux neurones MLP basés sur l'algorithme de rétropropagation. Ce type est très répandu dans le réseau neuronal et peut être utilisé comme un système pratique pour effectuer une cartographie d'entrée / sortie non linéaire de nature générale, les étapes d'algorithme de rétro-propagation sera décrit ci-dessous:

### Algorithme : Algorithme de rétro-propagation

#### Les entrées :

- Modèle d'entraînement  $(\vec{x}^p, \vec{d}^p)$ ,  $1 < p < = P$
- Modèle de validation  $(\vec{v}^q, \vec{d}^q)$ ,  $1 < q < = Q$
- Fonction d'activation  $f$  et sa dérivation  $f'(\xi)$ 
  - par exemple. La fonction sigmoïdale:  $f(\xi) = 1 / (1 + e^{-\xi})$ ;  $f'(\xi) = f(\xi) (1 - f(\xi))$
- Paramètre d'apprentissage  $\eta \in (0, 1)$
- Réseau neuronal M avec des poids initialisés de manière aléatoire.

#### Les sorties:

- Modèle réseau neuronal M'

#### L'algorithme:

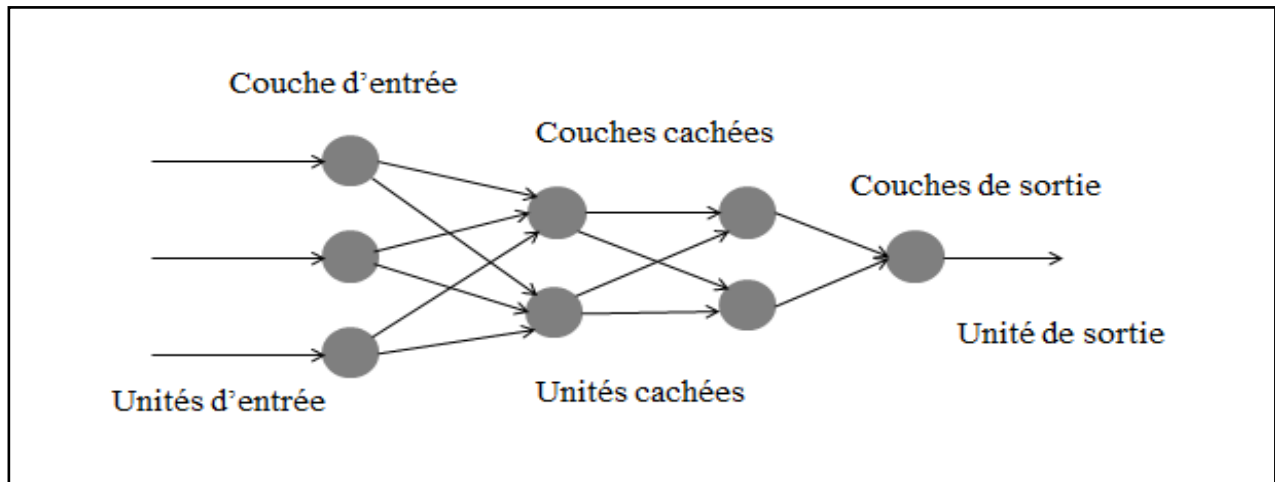
1. Définissez  $E_{avg} = \infty$ .
2. Commencer une nouvelle époque.
3. Pour chaque  $p \in \{1, \dots, P\}$  présente le motif  $(\vec{x}^p, \vec{d}^p)$ 
  - Poser  $\vec{y}^0 = \vec{x}^p$
  - Phase avant
    - Pour  $l = 1, 2, \dots, \kappa$  calcule la sortie de M pour  $\vec{x}^p$  de la façon suivante:  
 $\forall j \in L_l$  calculer  $\xi_j = \sum_{i=1}^n w_{ij} x_i + \theta$  et  $y_j = f(\xi)$
  - Phase rétrograde
    - $\forall j \in L_{\kappa-1, j} \in L_{\kappa}$  calculer  $\delta_j = (y_j - d_j^p) f'(\xi)$  et  $\Delta w_{i,j} = -\eta \delta_j y_j$
    - Pour  $l = \kappa - 1, \dots, 1$  faire :
      - $\forall i \in L_{k-1, j} \in L_l$  calculer  $\delta_j = \sum_{k=1}^{m_{l+1}} w_{j,k} f'(\xi_j)$ ,  $\Delta w_{i,j} = -\eta \delta_j y_j$
      - Pour  $l = 1, 2, \dots, \kappa \forall (i, j) \in L_{l-1} \times L_l$  ajuster  $w_{i,j}$  par  $\Delta w_{i,j}$
    - Calculer l'erreur pour le motif  $p$ :  $E_p = 1 / m_k \sum_{k=1}^{m_k} (y_j^p - d_j^p)^2$
4. Poser  $E_{prev} = E_{avg}$  et calculer nouveau  $E_{avg} = 1/P \sum_{p=1}^P E_p$  pour l'ensemble de données de validation.

5. Si  $E_{avg} < E_{prev}$  Alors passez à l'étape 2.
6. Fin.

Les réseaux MLP contiennent des couches. Chaque couche de connexion peut être assimilée à une fonction mathématique linéaire appliquée aux entrées. Celle-ci étant répartie sur l'ensemble des connexions. Ces réseaux se composent par trois couches :

- Les couches d'entrée avec  $R_i$  des unités d'entrée.
- Les couches cachées avec  $R_c$  des unités cachées.
- Les couches de sorties avec  $R_s$  des unités de sortie.

La structure d'un tel réseau multicouche perceptrons de trois entrées et une seule sortie est illustrée sur la figure II.2.



**Figure III.8 :** Architecture générale d'un réseau multicouche

### 3.3.1.2 Réseaux de neurones bouclés (récurrents)

Un réseau de neurone bouclé réalise une ou plusieurs équations aux différences non linéaires, telle que la direction des fonctions réalisées ver la sortie avec retour [Sastry et al., 1994].

### 3.3.2 Apprentissage dans les réseaux de neurones

L'apprentissage de réseau de neurone permet la modification des poids entre les neurones ainsi que la valeur des biais de façon à améliorer la performance des réseaux de neurones afin d'adapter le

traitement effectué [Man et Halang, 1997]. Nous pouvons citer deux principaux types d'apprentissage :

### **3.3.2.1 Apprentissage supervisé**

Ce type consiste à un superviseur qui fournit une valeur de sortie (sortie désirée), que le réseau de neurones doit associer à une valeur d'entrée. Cet apprentissage permet d'ajuster les paramètres du réseau afin de minimiser l'erreur entre la sortie désirée et la sortie réelle du réseau.

### **3.3.2.2 Apprentissage par renforcement**

Dans ce cas d'apprentissage, nous supposons qu'un comportement de référence n'est pas possible, mais en compensation, il est possible d'obtenir des indications qualitatives (vrai, faux, ...) sur les performances du réseau.

### **3.3.2.3 Apprentissage non supervisé**

Cet apprentissage est caractérisé par l'absence de d'information sur la sortie désirée des données et l'absence de superviseur. La tâche du réseau consiste par exemple, à générer des regroupements de données selon des propriétés communes.

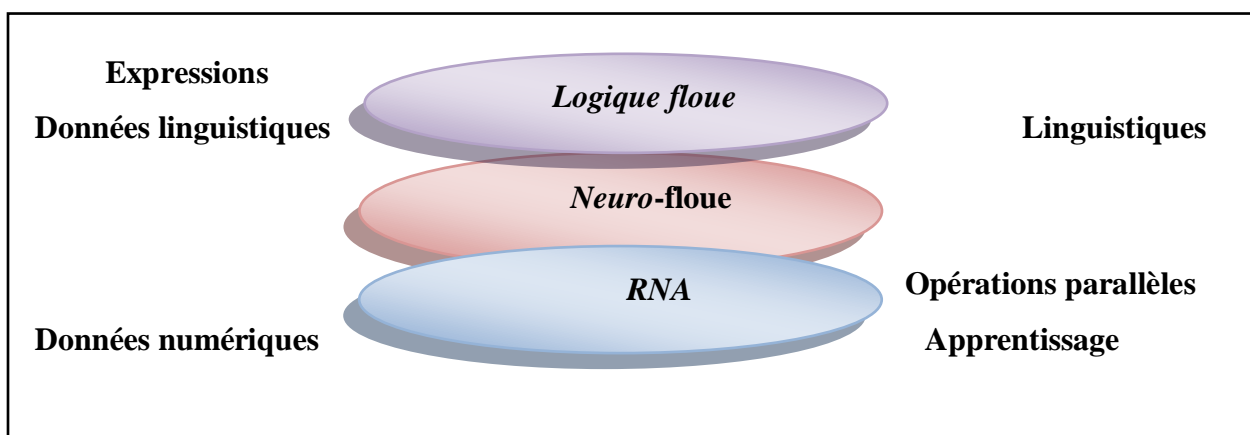
## **3.4 Système hybride Neuro-Flou**

Chaque technique intelligente possède des propriétés de calcul particulières comme par exemple la capacité d'apprendre, l'explication des décisions. Les réseaux de neurones reconnaissent les modèles pendant l'apprentissage, mais ils prennent leurs décisions sans explication. Aussi les systèmes de logique floue peuvent raisonner en compagnie des informations imprécises avec explication pour prendre les décisions, mais ils ne peuvent pas acquérir automatiquement leurs règles pour prendre les décisions. Ces limitations ont été motivées de création des systèmes hybrides intelligents. Ces systèmes combinés de manière à surmonter les limites des techniques individuelles. Les systèmes hybrides sont également importants lorsqu'on considère la nature variée des domaines d'application. Le système hybride neuro flou a été utilisé pour le traitement des informations complexes. Il offre des résultats plus précis que le système de la logique flou et les réseaux neurones. Ce système généralement fourni une solution intéressante pour le traitement des informations des systèmes industriels.

### 3.4.1 Définition

Le système neuro-floue est un système floue formé par un algorithme d'apprentissage inspiré de la théorie des réseaux de neurones et règles de la logique floue. La figure III.9 expose L'intégration des réseaux de neurones et les systèmes d'inférence floue[Nauck, 1997].

La figure III.9 présente le système neuro-floue.



**Figure III.9 :** Système neuro-floue

Le système neuro-floue hybride modifie sa structure interne pour refléter le rapport entre les entrées et les sorties dans l'ensemble de la formation. Aussi la précision d'un système neuro-floue est vérifiée après que le cycle de d'apprentissage soit complet en utilisant un ensemble séparé d'entrées et sorties appeler l'ensemble de la validation.

Plusieurs modèles d'hybridation ont été proposés par des chercheurs comme FALCON, ANFIS, NEFCON, NEFCLASS, NEFPROX et celui NEFDIAG [Fuller, 1995], [Khodja, 1997], [Abraham et B, 2001], [Rutkowski, 2004] et [Racoceanu, 2006].

### 3.4.2 Le système ANFIS

ANFIS (Adaptive-Network-based Fuzzy Inference System) est un système qui combine les concepts de la logique floue et les réseaux de neurones pour développer un système intelligent hybride qui augmente la capacité d'apprentissage et d'adaptation. Des systèmes hybrides ont été employés par des chercheurs pour la modélisation et la prévision. Ce système a été proposé par Jang[Jang, 1993]. La structure de ce système est composée de cinq couches [Constantin, V.A., 1995].

### 3.4.2.1 Couche 1

Dans cette couche, chaque nœud exécute une fonction d'appartenance à un sous-ensemble floue de l'univers de discours des entrées. Cette fonction d'appartenance de nœud est donnée par:

$$\begin{cases} O_i^1 = \mu_{A_i}(x) & i = 1, 2 \\ O_i^1 = \mu_{B_i}(x) & i = 1, 2 \end{cases} \quad (\text{III.6})$$

Ou :  $\mu_A$  et  $\mu_B$  peuvent être n'importe quelle fonction d'appartenance flou (MF).

### 3.4.2.2 Couche 2

Chaque nœud de cette couche applique une règle floue de type Sugeno. Ce nœud calcule la force d'activation de sorties de fuzzification. Le calcul est réalisé avec l'opérateur produit. Donc la force d'activation d'une règle se fait par multiplication :

$$O_i^2 = w_i = \mu_{A_i}(x) * \mu_{B_i}(y) \quad i = 1, 2 \quad (\text{III.7})$$

### 3.4.2.3 Couche 3

Dans cette couche chaque nœud calcule le degré d'activation normalisé d'une règle floue donnée. La valeur obtenue représente la contribution de la règle floue au résultat final. Donc l'i éme règle par rapport à la somme des forces d'activation de toutes les règles:

$$O_i^3 = \bar{w}_i = \frac{w_i}{w_1 + w_2} \quad i = 1, 2 \quad (\text{III.8})$$

Ou :  $\bar{w}_i$  représente la forces d'activation normalisée.

### 3.4.2.4 Couche 4

Dans ce couche la sortie de chaque nœud se donnée suivant la fonction de chaque nœud  $i$  par l'équation :

$$O_i^4 = \bar{w}_i z_i = \bar{w}_i (p_i x + q_i y + r_i) \quad i = 1, 2 \quad (\text{III.9})$$

Ou :  $\bar{w}_i$  est la sortie de la couche 3, et  $(p_i, q_i, r_i)$  est l'ensemble des paramètres ajustés de sortie de la règle  $i$ .



### 3.4.2.5 Couche 5

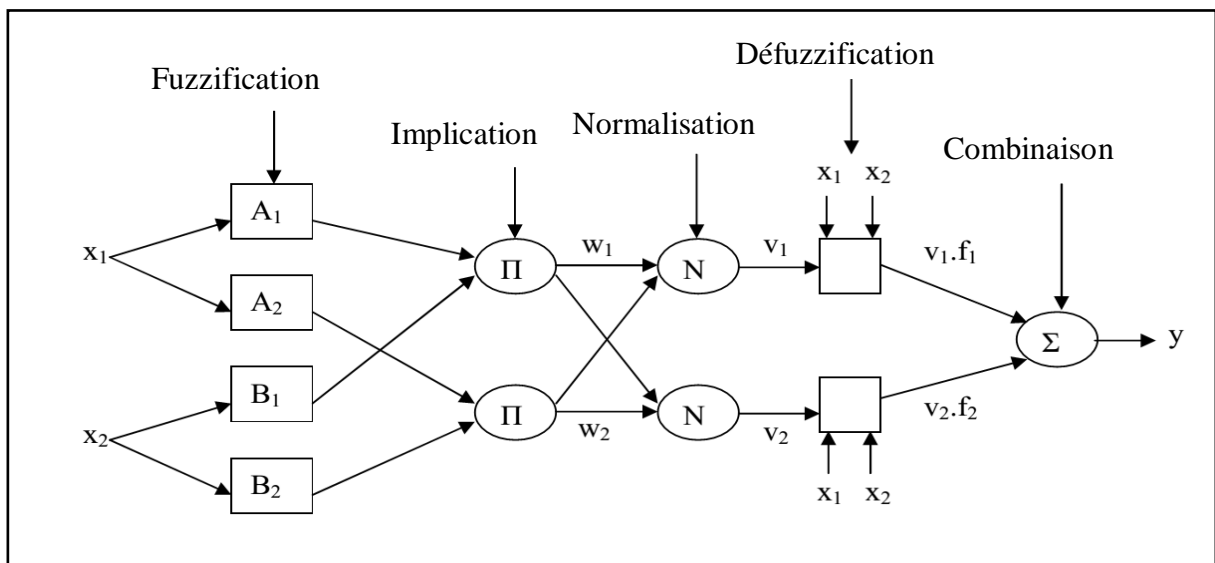
Cette couche contient un seul nœud qui génère la sortie de l'ANFIS par le calcul de la somme des sorties de tous les nœuds de défuzzification. L'opération de ce couche est donnée par :

$$O_i^5 = \sum_i \bar{w}_i z_i = \frac{\sum_{i=1}^2 w_i z_i}{\sum_{i=1}^2 w_i} \quad i = 1, 2 \quad (\text{III.10})$$

La sortie  $y$  sur le schéma 2 peut être réécrite comme :

$$z = (\bar{w}_1 x) p_1 + (\bar{w}_1 y) q_1 + (\bar{w}_1) r_1 + (\bar{w}_2 x) p_2 + (\bar{w}_2 y) q_2 + (\bar{w}_2) r_2 \quad (\text{III.11})$$

La figure suivant représente un système ANFIS à deux entrées  $x_1$  et  $x_2$  et une sortie  $y$ .



**Figure III.10** : Les différentes couches d'ANFIS

### 3.4.3 Supervision industriel à basée de l'intelligence artificielle

Dans l'industrie nous avons les systèmes sont complexes et de comportement non linéaire rend l'étape de modélisation difficile voire impossible. Pour cette raison, des travaux se sont orientés vers l'exploitation des réseaux adaptatifs, plus spécifiquement, du système neuro-floue ANFIS : différents travaux montrent que le système ANFIS est capable de "capturer" le comportement d'un système rapidement et précisément à court terme, et cela, mieux encore que

d'autres méthodes, réseaux de neurones artificiel (RNA) inclus. Plusieurs travaux relatifs à ces techniques d'intelligence artificielle ont été vérifiées sur des problèmes de prédiction et ont donné de meilleures performances [Wang W. Q. et al., 2002], [Yam et al., 2001] et [Zhang, 1998].

### **3.5 Conclusion**

Nous avons présenté dans ce chapitre les trois principales techniques de l'intelligence artificielle, Le système flou avec une discussion sur la procédure de traitement de l'information, c'est à dire leur notions et modules. Et les réseaux neurones avec un aperçu sur les différentes architectures de ces systèmes, et les différents types d'apprentissage associés. Enfin parmi tous les systèmes neuro-floue nous choisirons le système ANFIS.

Le prochain chapitre sera consacré aux principales définitions de réseaux de Pétri avec présentations de différentes extensions, ensuite nous proposons des études par l'utilisation de réseaux Pétri.

## **CHAPITRE 4**

### **Simulation et résultats**

## **4.1 Introduction**

Ce chapitre présente la simulation d'un système de supervision, ce système est appliqué sur un robot manipulateur à base d'une stratégie de télécommunication entre robot superviseur et robot supervisé avec l'utilisation de système neuro flou ANFIS, dans le but de la détection de défaut et la correction de commande de robot qui déjà présenté en cas de présence de défaut de frottement.

La communication informatique offrent la transmission d'ensemble de données entre les systèmes, dans ce travail nous utilisons communication informatique de protocole TCP (Instrument Control Toolbox for MATLAB), pour le transfert de donnée entre le robot superviseur et le robot supervisé. Aussi, nous utilisons le système ANFIS (Neuro-Fuzzy logic toolbox and Robotics Toolbox for MATLAB), pour générer un modèle ANFIS, et nous utilisons ce modèle dans la correction de l'effet de défaut de frottement. Une simulation de blocs de System-Generator en MATLAB a été ajoutée, pour la réalisation des fonctions proposées sur une carte électronique de Zync kentex.

Les simulations nous permettront, de tirer des conclusions pertinentes en la supervision d'un robot manipulateur.

## **4.2 Communication TCP IP**

La communication par internet offre à nos jours plusieurs services très diversifiés dans le domaine de l'industrie. Cette technique a été utilisée pour la commande à distance et la surveillance à distance. Un ensemble de protocoles ont été proposées pour relier les machines dans un réseau. Il est possible d'utiliser la forme actuelle d'Internet comme moyen de communication pour la supervision d'un système industriel en temps réel. Nous pouvons grandement améliorer les avantages de cette architecture de supervision. L'intégration d'Internet réduira le coût et le temps nécessaires à la mise en place du système de supervision distribué en temps réel dans les grandes industries. Les applications logicielles peuvent être largement utilisées pour surveiller et contrôler à distance les processus à travers l'Internet. L'intégration d'Internet dans la supervision donnera également aux ingénieurs la liberté de répartir les tâches de contrôle, les capteurs et les actionneurs dans des emplacements optimaux. Ainsi, les ingénieurs seront en mesure de surveiller les applications fonctionnant dans des environnements difficiles offrant une accessibilité limitée. Nous citons comme exemple de protocoles l'UDP, le HTTP, le FTP et le TCP.

Le protocole de transport TCP / IP est le plus utilisé sur l'Internet. C'est le protocole utilisé pour les applications qui nécessitent une communication fiable. TCP / IP soutient la livraison garantie. En d'autres termes, il transmet les données et attend l'accusé de réception des données du client avant l'envoi des données suivantes. Ceci introduit plus de retard dans la transmission de données en utilisant TCP / IP, par contre l'UDP élimine le temps système de retransmission, UDP ne soutient pas la livraison garantie, donc les données risquent d'être perdues pendant la transmission par la communication de UDP, Ce fait affecte la stabilité des applications critiques de données.

Nous avons choisi la communication de protocole TCP (fig IV.1) à cause de sa compatibilité avec la plupart des applications et aussi à cause de l'interfonctionnement de n'importe quel serveur avec n'importe quel protocole de bas niveau [Atakan, 2009], [Melha, 2005] et [Ghrieb A. et al, 2017].

#### **4.2.1 Protocoles TCP**

Dans le présent travail, le robot superviseur permet de guider le robot supervisé à travers un système de supervision qui reçoit les paramètres nécessaires à partir des deux robots [Xinxing Tang]. En plus, le robot supervisé reçoit les informations nécessaires à partir du robot superviseur par l'intermédiaire d'un système de communication avancé (le protocole TCP) [Ghrieb A. et al, 2017] et [Ghrieb A. et Kourd Y., 2018].

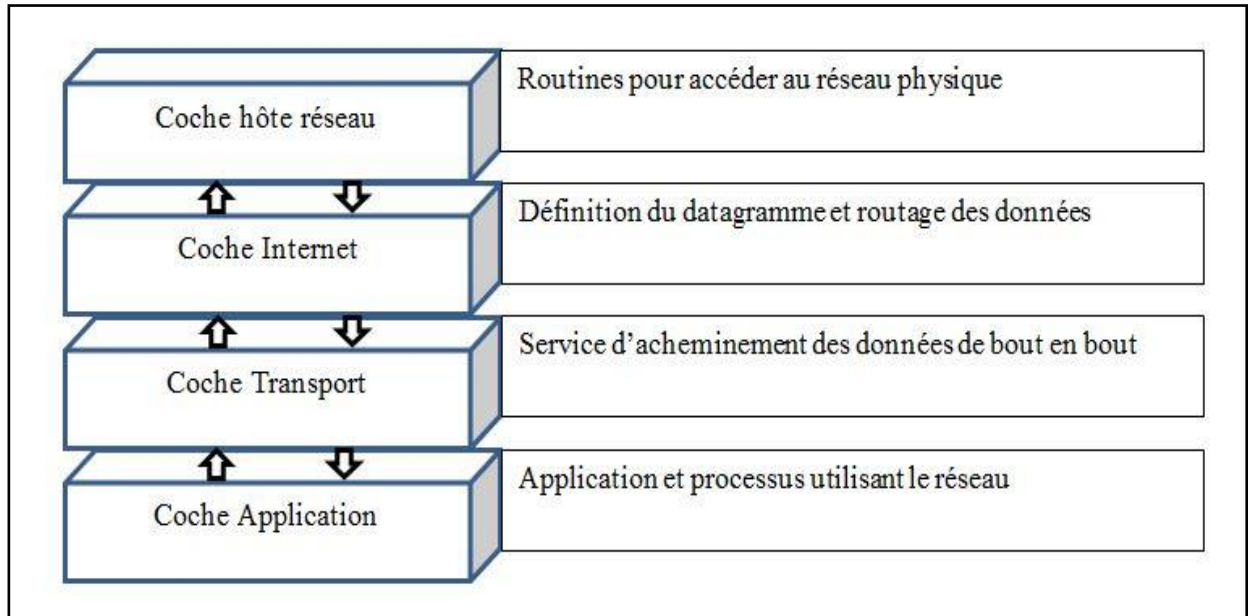
Le protocole TCP se compose par plusieurs modules, ces modules sont connus par les couches, chaque couche de ce protocole effectuant une tâche précise (figureV.1).

**4.2.1.1 Couche hôte réseau :** Cette couche contient des protocoles qui gèrent l'acheminement des informations entre émetteur et récepteur. Cette couche comporte les adresses MAC(Media Access Control) ainsi que le protocole d'Ethernet et le protocole Wifi (802.11).

**4.2.1.2 Couche Internet :** La couche internet principalement comprend deux protocoles. L'un est le protocole IP qui permet le routage des informations entre réseaux et l'autre est le protocole ICMP qui permet le contrôle d'erreur et de signalisation.

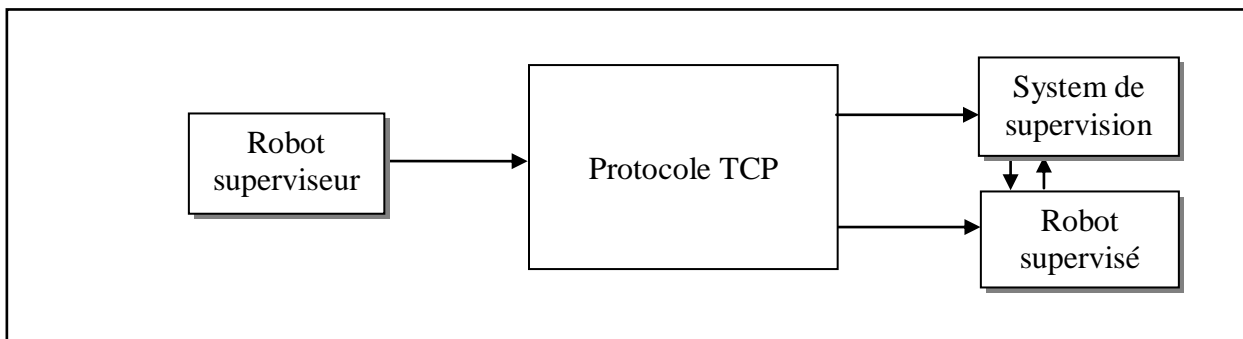
**4.2.1.3 Couche Transport :** Cette couche contient les protocoles de transport des données.

**4.2.1.4 Couche Application :** La couche application absorbe les protocoles qui gèrent communication entre les clients et les serveurs.



**Figure IV.1 :** Architecture de protocole TCP

Dans notre travail nous utilisons ce protocole de communication entre trois systèmes, robot superviseur, robot supervisé et système de supervision. La figure IV.2 présente la communication par le protocole TCP.



**Figure IV.2 :** L'interconnexion de système global par protocole TCP

### **4.3 Implémentation Matérielle sur cible FPGA**

Actuellement, les circuits FPGA (Field-Programmable Gate Array) sont devenus très utilisés aussi bien pour la réalisation des prototypes que pour l'implémentation en temps réel des algorithmes de traitement. Les FPGAs sont utilisés dans le domaine industriel pour la réalisation des lois de commande en temps réel. Dans cette partie de ce travail, nous proposons une implantation matérielle de l'algorithme de supervision détaillé précédemment. Nous convertissons les blocs Simulink vers des blocs Xilinx sous Simulink, et nous utilisons le System-Generator pour la conversion automatique des modèles Simulink en un code HDL (Hardware Description Language). A partir de ce dernier point, nous utilisons les outils de chez Xilinx pour la synthèse et l'implémentation des codes HDL sur les circuits FPGA. Le but de l'utilisation des FPGAs est d'assurer en temps réel, d'utilisation des robots manipulateurs industriels (robots réels).

La comparaison des résultats de simulation après le passage de la précision en virgule flottante utilisée par Matlab vers la précision en virgule fixe utilisé dans les implémentations matérielles. L'implémentation matérielle sur FPGA peut être aussi utilisée avec les modèle Simulink en boucle (in the loop). En effet, dans telle architecture, les robots sont simulés sous Matlab-Simulink par contre le superviseur est implémenté physiquement sur un circuit FPGA et c'est au System-Generator d'assurer la liaison temps réel entre le circuit FPGA et le Modèle Simulink.

#### **4.3.1 System-Generator de chez Xilinx**

Matlab-Simulink est un environnement de développement graphique destiné à la conception, basé sur des modèles. Simulink possède une bibliothèque prédéfinie enrichie par la modélisation sous forme de blocs et d'une grande partie des fonctions Matlab. Dernièrement, les ingénieurs de Mathworks et les ingénieurs de Xilinx ont proposé un nouveau système sous Matlab-Simulink nommé System-Generator. Il maintient un niveau très harmonieux avec les blocks set Simulink, mais en même temps permet la traduction automatique des modèles vers des implémentations matérielles synthétisable et efficace. La mise en œuvre est rendu efficace par l'instanciation de la propriété intellectuelle (IP) blocs qui fournissent une gamme de fonctionnalités à partir des opérations arithmétiques à des fonctions complexe. Les blocs IP définis par l'utilisateur peuvent être intégrés dans un modèle de système générateur comme des boîtes noires qui seront incorporées par l'outil dans l'HDL de la conception. L'utilisation de Matlab-Simulink permet la création facile de testbench pour la simulation et la co-simulation des modèles Matlab et Xilinx en même temps.

Dans une implémentation matérielle sous Simulink et System Generator, les étapes suivantes sont nécessaires:

1. Décrire l'algorithme en termes mathématiques ;
2. Réaliser l'algorithme dans l'environnement de conception (Matlab/Simulink), d'abord à l'aide de blocs Simulink travaillant en double précision ;
3. Réalisation de l'algorithme avec des blocs Xilinx travaillant sur des données en point fixe.
4. Simulation et co-simulation des modèles en utilisant les outils de Simulink et comparaison des résultats.

La conversion automatique des modèles Simulink (à base de blocs Xilinx uniquement) en un code HDL et synthèse des résultats [Ghrieb A. et al, 2019].

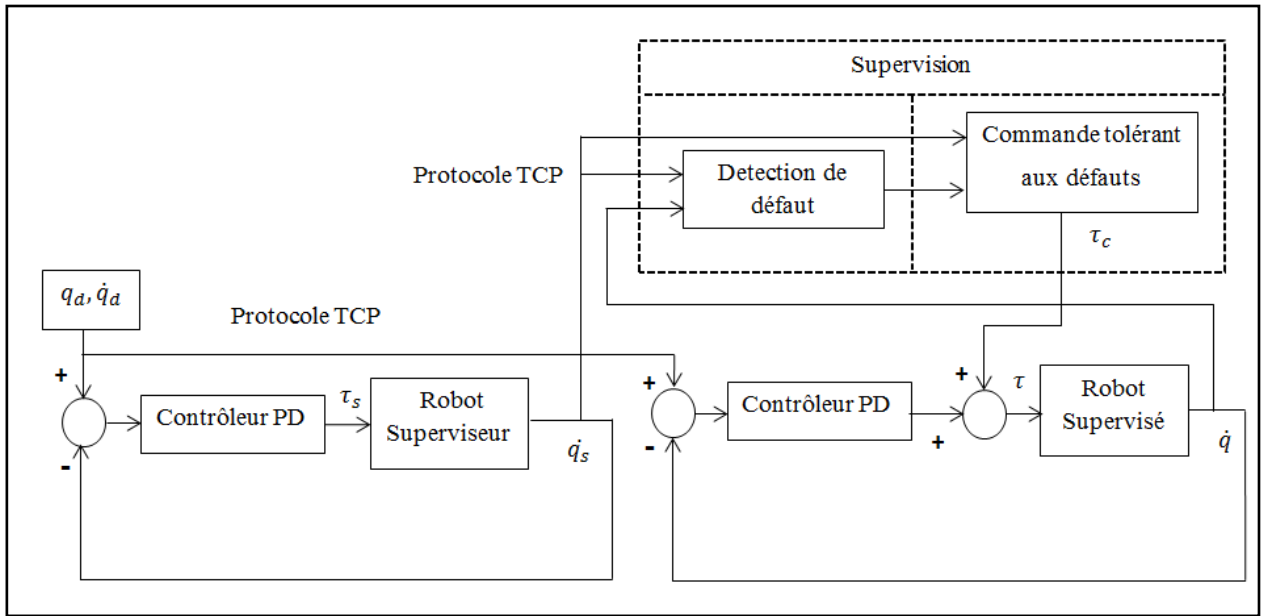
### **4.3.2 Implémentation Matérielle**

Dans un premier temps, nous avons recopié le même modèle Simulink du superviseur mais en utilisant uniquement les blockset Xilinx. Plusieurs essais sont réalisés pour le bon redimensionnement des variables et la synchronisation des fréquences de fonctionnement des blocs Simulink et des blocs Xilinx respectivement. Le but de cette partie consiste à assurer le passage de la précision en virgule flottante (utilisée par les blocs Simulink) vers la précision en virgule fixe (utilisée par les blocs Xilinx) pour que les résultats soient comparables.

### **4.4 Schémas de système global**

Le système global se compose par trois parties, un robot manipulateur superviseur dit maitre, un robot esclave et un module de supervision. Pour cette raison, nous avons utilisé une technique de communication informatique qui relie le robot superviseur au robot supervisé et au module de supervision. Cette technique est la communication de protocole TCP qui offre l'envoi du signal de trajectoire désiré vers le robot esclave et l'envoi du signal de vitesse angulaire du robot maitre vers le module de supervision. En effet le schéma assure la privation de retard d'envoi en la supervision. L'envoi des signaux (trajectoire désirée et vitesse angulaire de robot maitre) est fait simultanément. La figure IV.3 montre le schéma de système global proposé.





**Figure IV.3 :** Schémas de système global

Le système de supervision comprend, un module de surveillance et un module dit de commande tolérante aux défauts.

#### 4.4.1 La supervision de robot manipulateur

Nous avons proposé un système de supervision basé sur des informations de robot manipulateur superviseur. Ces informations indiquent la présence d'un défaut de robot supervisé par la détection de défaut, et de commande tolérante aux défauts pour prendre la décision en ce mauvais fonctionnement.

##### 4.4.1.1 La détection de Défaut

La fonction de résidus est généralement réalisée pour deux lesquels principaux : la détection de défaut et le développement d'un signal d'évaluation de supervision, c'est-à-dire que les fonctions de résidus représentent l'effet de défaut par une amplitude après la correction de système de supervision. Par ces fonctions nous pouvons déduire le volume d'effet de défaut sur le système et l'existence de mauvais fonctionnement. Les résidus sont obtenus par la différence entre les vitesses articulaires de robot superviseur et les vitesses du robot supervisé. Les résidus  $R_1$  et  $R_2$  sont représentés par les équations IV.1 et IV.2 [Ghrieb A. et al, 2017] et [Ghrieb A. et al, 2019].

$$R_1 = \dot{q}_1 - \dot{q}_{1s} \quad (IV.1)$$

$$R_2 = \dot{q}_2 - \dot{q}_{2s} \quad (IV.2)$$

Ou :

$\dot{q}_1$  et  $\dot{q}_2$  : Les vitesses articulaire du robot supervisé.

$\dot{q}_{2s}$  et  $\dot{q}_{1s}$  : Les vitesses articulaire du robot superviseur.

Les fonctions de résidus proposées sont présentées par les équations suivantes :

$$F1 = \int_{ta}^{tc} (R_1^2 + R_2^2) dt \quad (IV.3)$$

$$F2 = \int_{ta}^{tc} (R_1 + R_2)^2 dt \quad (IV.4)$$

Si nous prenons en considération les bruits des vitesses, les fonctions de résidus deviennent :

$$F1 = \int_{ta}^{tc} ((R_1 + n_1)^2 + (R_2 + n_2)^2) dt + \int_0^{tf} (n_1^2 + n_2^2) dt - \int_{ta}^{tc} (n_1^2 + n_2^2) dt \quad (IV.5)$$

$$F2 = \int_{ta}^{tc} (R_1 + R_2 + n_1 + n_2)^2 dt + \int_0^{tf} (n_1 + n_2)^2 dt - \int_{ta}^{tc} (n_1 + n_2)^2 dt \quad (IV.6)$$

Ou :

$$n_1 = w_1 - w_{1s} \quad (IV.7)$$

$$n_2 = w_2 - w_{2s} \quad (IV.8)$$

$w_1$  et  $w_2$  sont les bruits des vitesses du robot supervisé et  $w_{1s}$  et  $w_{2s}$  sont les bruits des vitesses de robot superviseur.

ta : Temps affect de défaut.

tc : Temps début de correction.

tf : Temps fin de simulation.

La détection de défaut est vérifiée par le traitement de résidus en état sain du robot manipulateur. Premièrement, la génération de résidus, ensuite le calcul d'amplitude de l'effet de bruit pendant le fonctionnement de robot manipulateur, pour voir le maximum de l'effet de bruit par

les fonctions de résidus (équations 9 et equ.10). Donc les seuils de fonction de résidus F1 et F2 sont s1 et s2 respectivement.

$$s1=2\int_0^{tf}(n_1^2 + n_2^2) dt \quad (IV.9)$$

$$s2= 2 \int_0^{tf}(n_1 + n_2)^2 dt \quad (IV.10)$$

#### 4.4.1.2 Commande tolérante aux défauts

La commande tolérante aux défauts est créée principalement pour la génération d'un signal de correction. Ce signal est utilisé uniquement en cas de l'existence de défaut. C'est-à-dire l'utilisation de ce module est en fonction de la décision fournie par le module de détection de défaut. Généralement, la sortie de ce module est basée sur les vitesses fournies par le robot maître (superviseur). La sortie de ce module est un vecteur de couple  $\tau_c$ , ce vecteur permet d'assurer la reconfiguration de couple de robot supervisé pendant la présence de défaut. Dans cette étude la commande tolérante aux défauts (FTC) est basée sur le protocole TCP, pour récupérer les entrées représentées par les vitesses articulaires du robot à distance (fig IV. 3).

Nous avons utilisé un modèle ANFIS pour satisfaire la tâche de commande tolérante aux défauts. Le système d'ANFIS basé sur les vitesses articulaires de deux articulations comme entrées et fourni une sortie  $\tau_c$  (couple de correction). Après l'apprentissage d'ANFIS à base de données de simulation, nous avons récupéré deux systèmes pour la première et la deuxième articulation. Chaque système est caractérisé par les paramètres présentés dans le tableau IV.1 et IV.2.

**Tableau IV.1 :** Les paramètres de modèle ANFIS 1

Système ANFIS	Information	
ANFIS pour la correction de défaut de première articulation	Nombre de nœuds	35
	Nombre de paramètres linéaires	27
	Nombre de paramètres non linéaires	18
	Nombre total de paramètres	45
	Nombre de données d'entraînement	1000
	Nombre de données de contrôle	1000
	Nombre de règles floues	9

**Tableau IV.2 :**Les paramètres de modèle ANFIS 2

Système ANFIS	Information	
ANFIS pour la correction de défaut de deuxième articulation	Nombre de nœuds	35
	Nombre de paramètres linéaires	9
	Nombre de paramètres non linéaires	18
	Nombre total de paramètres	27
	Nombre de données d'entraînement	1000
	Nombre de données de contrôle	1000
	Nombre de règles floues	9

#### 4.5 Résultats de simulation et discussions

Dans cette partie de notre travail nous présentons les résultats de l'implémentation du système de supervision appliqué à un robot supervisé. Ce robot est un robot à deux degrés de liberté modélisé en utilisant Matlab-Simulink (§2). Le robot superviseur et supervisé sont caractérisés par les paramètres suivants :  $m_1 = 1$  kg,  $m_2 = 2.3$  kg et  $l_1 = l_2 = 1$  m (voir la figure IV.3). Supposant aussi que la gravité égal à 9.8 m/s. Pour la commande, nous avons utilisé un contrôleur PD qui se caractérise par les paramètres  $K_p = 700$  et  $K_d = 25$ .

La trajectoire de référence est représentée par les équations suivantes :

$$q_1 = 3 + 6 (\sin(t) + \sin(2t)) \tag{IV.11}$$

$$q_2 = 2 + 4 (\cos(t) + \sin(2t)) \tag{IV.12}$$

A partir des équations (II.73) et (II.74) on obtient:

$$\tau = M(q)\ddot{q} + V(q, \dot{q})\dot{q} + F_v(\dot{q}) + F_a(\dot{q}) + G(q) \tag{IV.13}$$

Nous supposons que le défaut de robot supervisé est de type frottement visqueux, donc le défaut est représenté par les équations suivantes :

$$F_a(\dot{q}) = \begin{bmatrix} 0 \\ 0 \end{bmatrix} \tag{IV.14}$$

$$F_v(\dot{q}) = \begin{bmatrix} F_{v1}\dot{q}_1 \\ F_{v2}\dot{q}_2 \end{bmatrix} \quad (IV.15)$$

$$F_{vi}(\dot{q}) = 0.0009 t \quad (IV.16)$$

Avec :

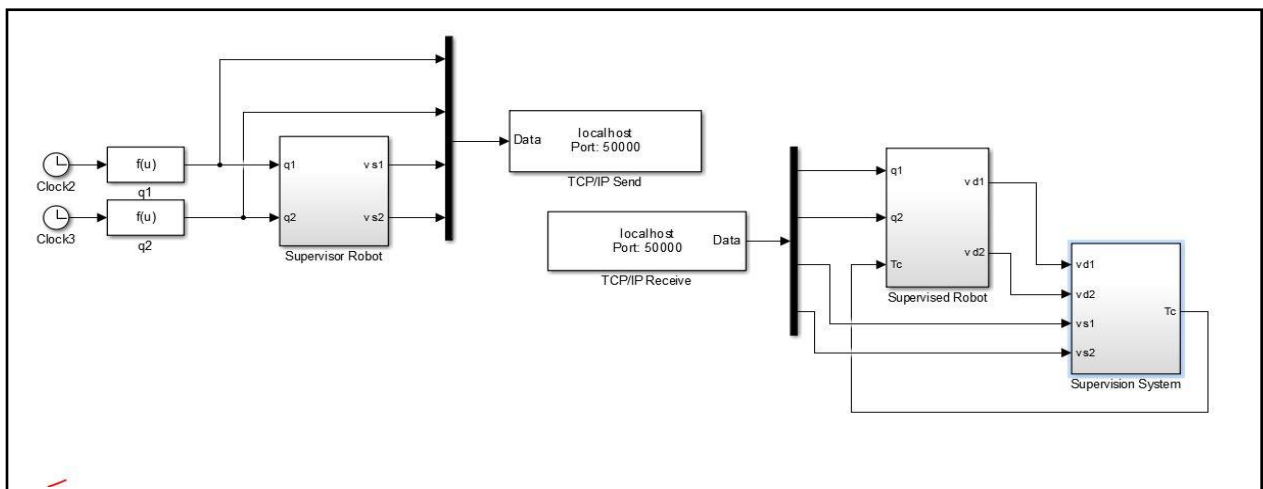
$i = 1$  le défaut dans l'articulation 1

$i = 2$  le défaut dans l'articulation 2

Le modèle de robot supervisé est défini par la matrice suivante:

$$\begin{bmatrix} (m_1+m_2)l_1^2 + m_2l_2^2 + 2m_2l_1l_2\cos(q_2) & m_2l_2^2 + m_2l_1l_2\cos(q_2) \\ m_2l_2^2 + m_2l_1l_2\cos(q_2) & m_2l_2^2 \end{bmatrix} \begin{bmatrix} \ddot{q}_1 \\ \ddot{q}_2 \end{bmatrix} + \begin{bmatrix} -m_2l_1l_2(2\dot{q}_1\dot{q}_2 + \dot{q}_2^2)\sin(q_2) \\ m_2l_1l_2\dot{q}_1^2\cos(q_2) \end{bmatrix} + \begin{bmatrix} (m_1+m_2)gl_1\cos(q_1) + m_2gl_2\cos(q_1 + q_2) \\ m_2gl_2\cos(q_1 + q_2) \end{bmatrix} + \begin{bmatrix} F_{v1}\dot{q}_1 \\ F_{v2}\dot{q}_2 \end{bmatrix} = \begin{bmatrix} \tau_1 \\ \tau_2 \end{bmatrix} \quad (IV.17)$$

La figure IV.4 présente le système global en utilisant Matlab-Simulink. Le défaut de frottement visqueuse influe sur les sorties (positions et vitesses) du robot supervisé. Dans notre étude, l'évaluation de fonctionnement est basée sur l'observation des vitesses articulaires de robot supervisé et de robot superviseur par l'analyse des résidus.

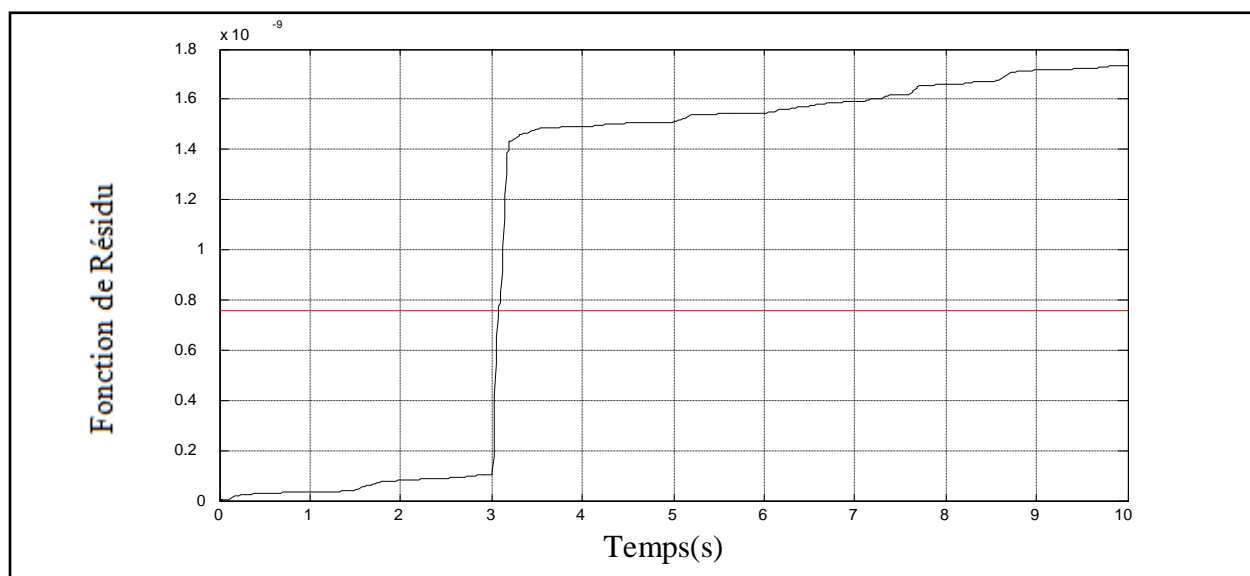


**Figure IV.4 :** Diagramme de base du système global proposé sous Matlab-Simulink

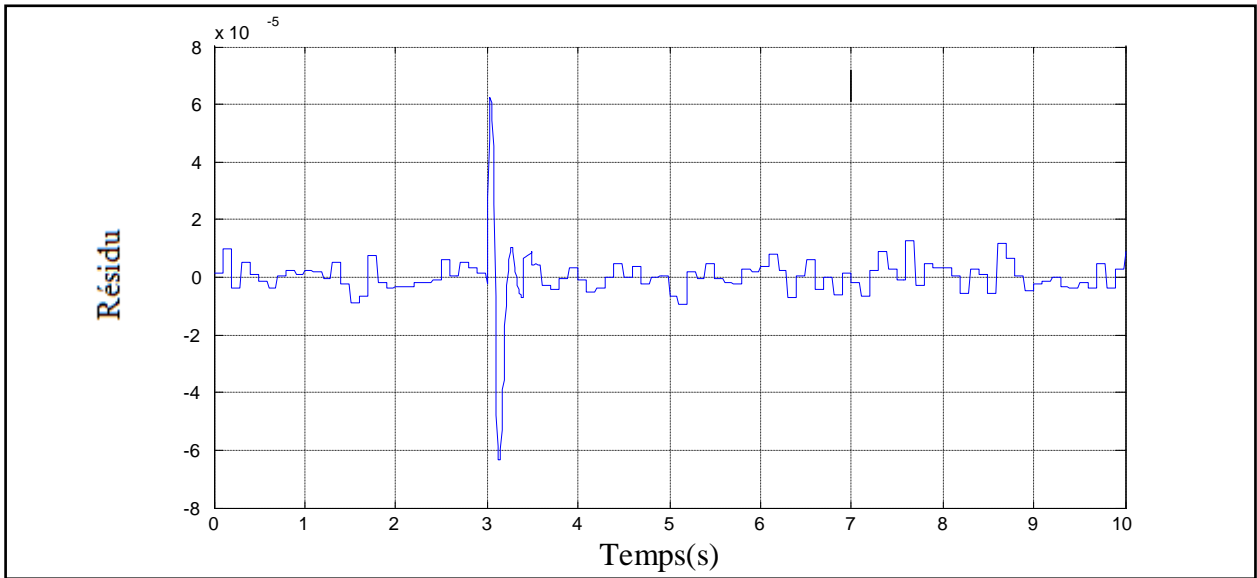
#### 4.5.1 Résultat de défaut de la première articulation

Le dépassement de seuil par le vecteur de fonction de résidu indique le mauvais fonctionnement du système ainsi que la présence de défaut dans le robot supervisé. Les résidus montrent l'existence de défaut par les changements dans les vitesses (fig.IV.5, IV.6, IV.7, IV.8, IV.9 et IV.10). Le dépassement de la fonction F1 est remarqué à partir du temps 3.0755 s et le dépassement de seuil de la fonction F2 est observé à partir du temps 3.1765 s. Donc, la détection de défaut par l'utilisation de fonction de résidu F1 est plus rapide que l'utilisation de fonction de résidu F2.

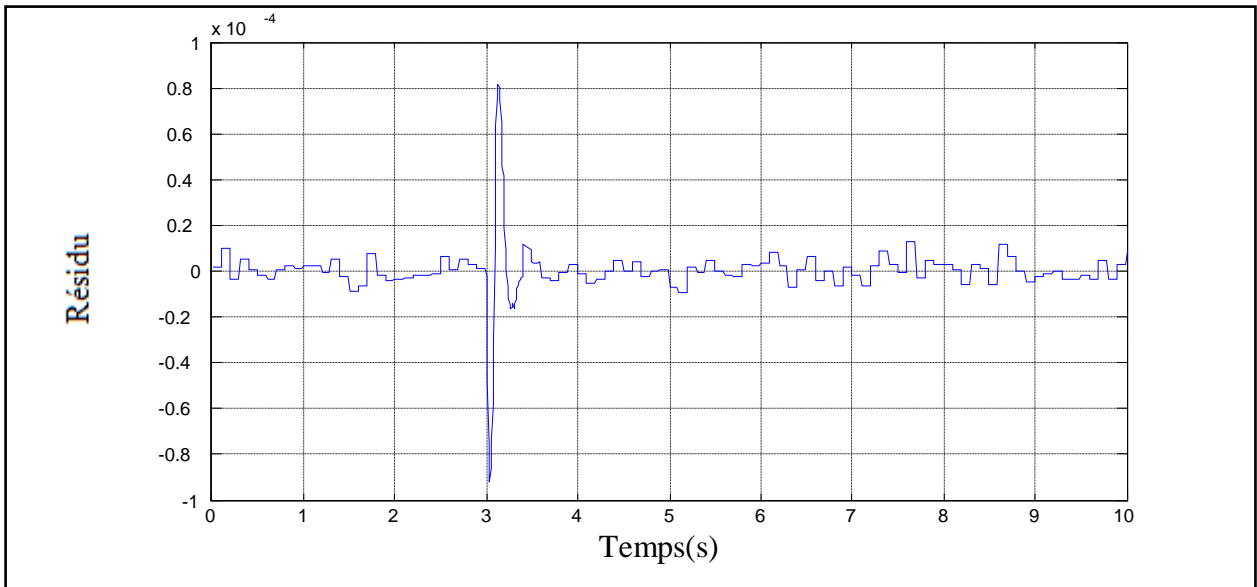
L'utilisation de la fonction de résidu F1 montre que l'effet de défaut est présent à partir du temps 3.0755s jusqu'au 3.5s. De la même méthode, l'effet de défaut est présent à partir du temps 3.1765 s jusqu'au 3.6 s en utilisant la fonction de résidu F2. Ce qui implique que la correction fournie par la fonction de résidu F1 est plus rapide.



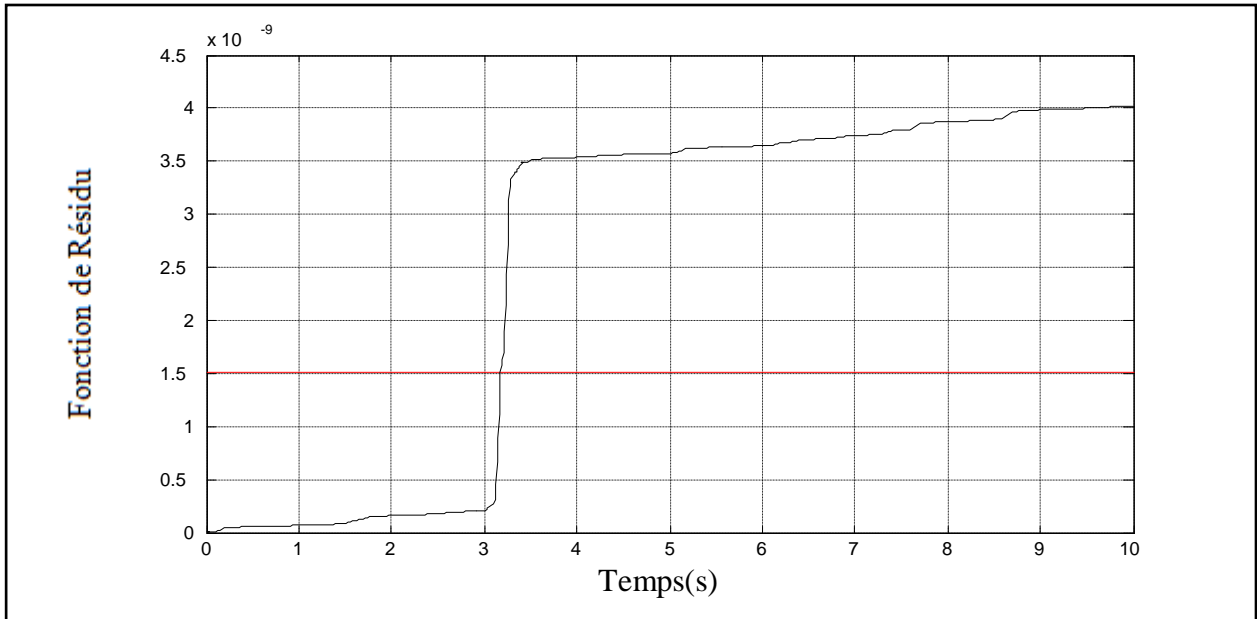
**Figure IV.5 :** La fonction de Résidu F1



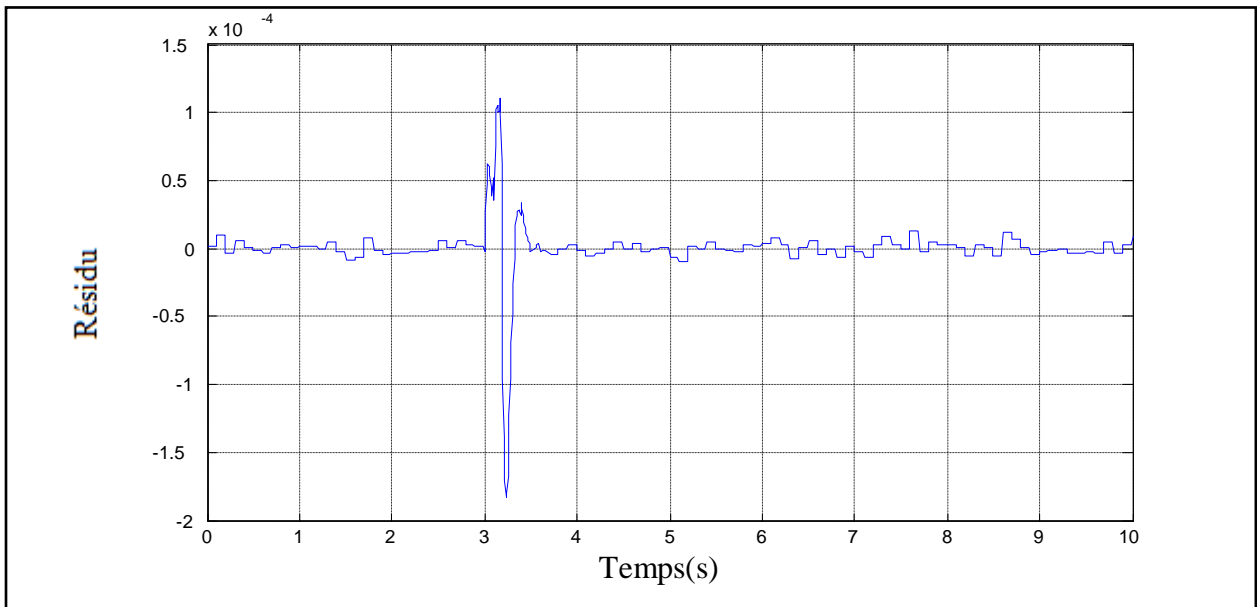
**Figure IV.6 :** Résidu de première articulation



**Figure IV.7 :** Résidu de deuxième articulation

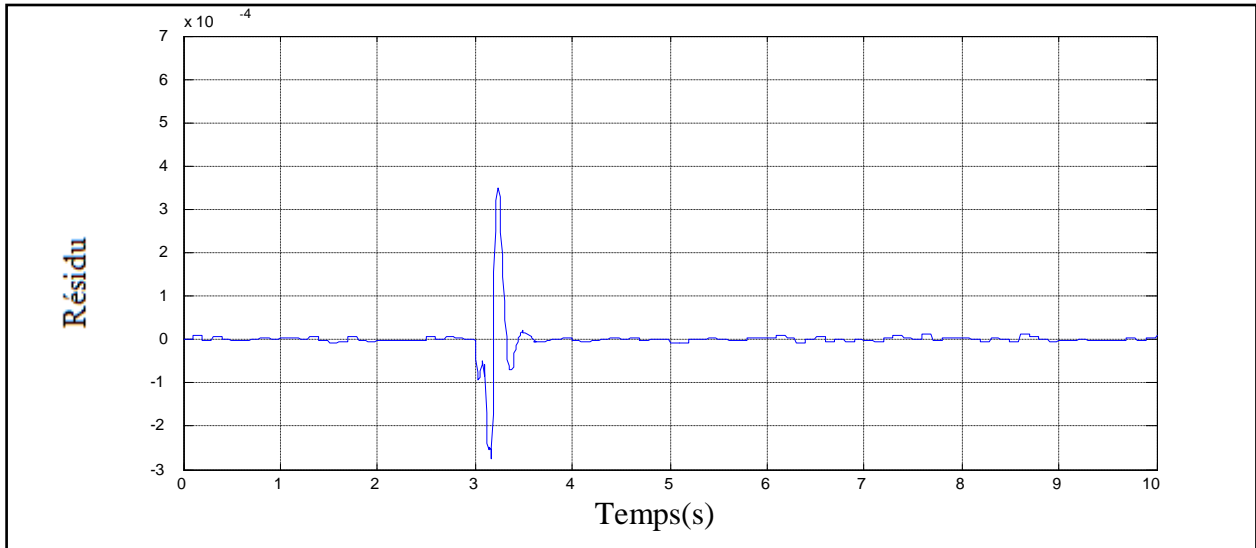


**Figure IV.8 :** La fonction de Résidu F2



**Figure IV.9 :** Résidu de première articulation

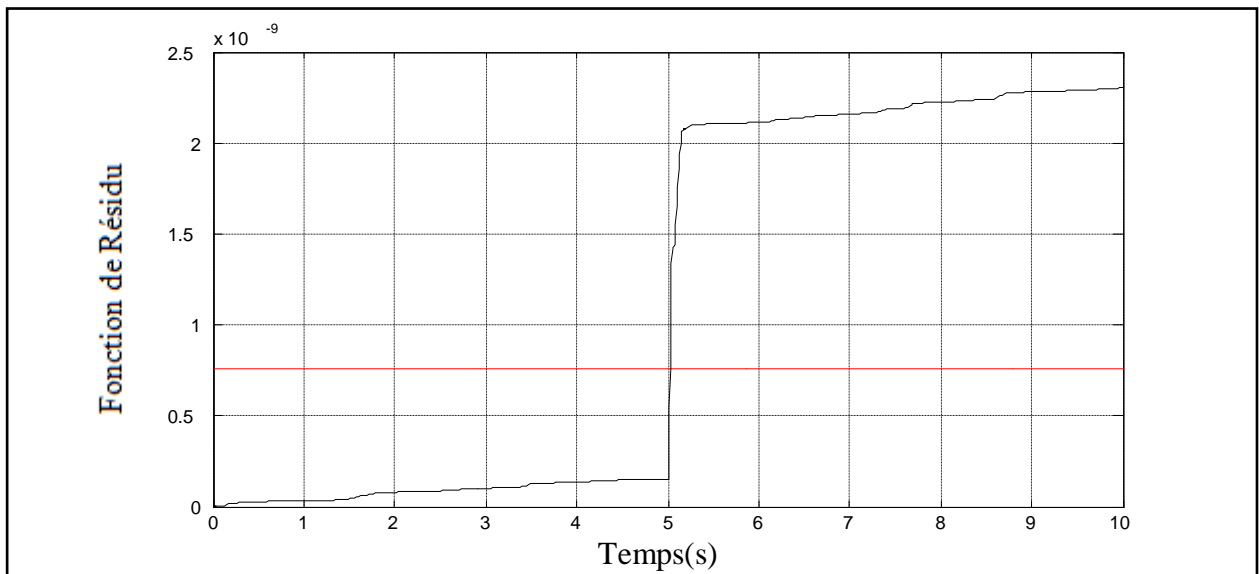




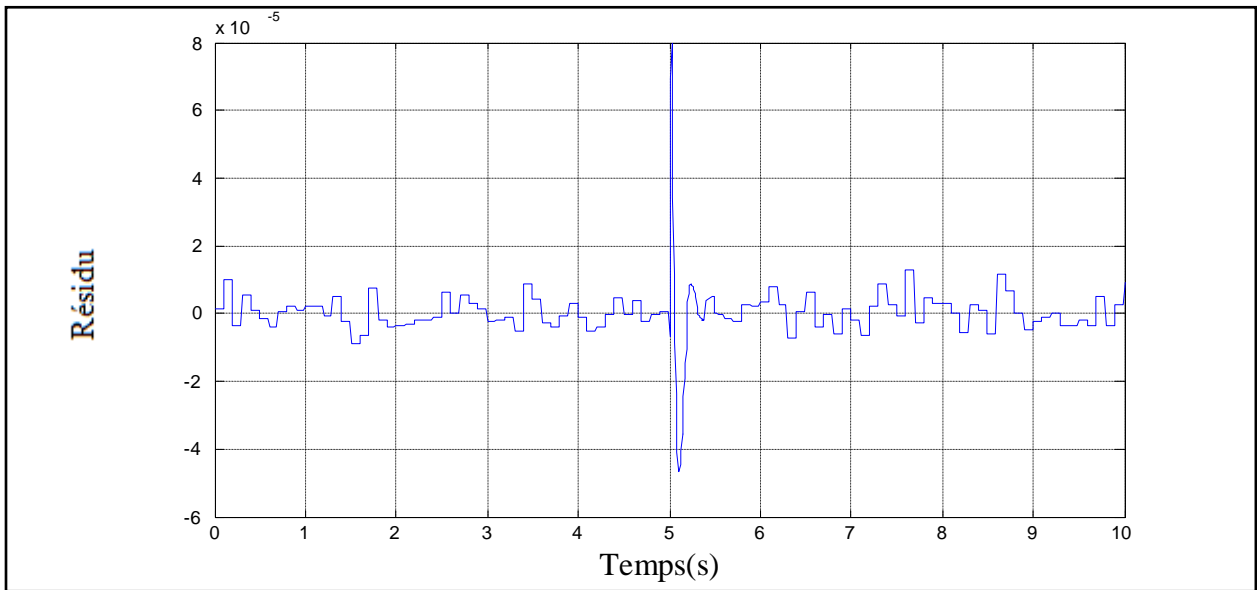
**Figure IV.10 : Résidu de deuxième articulation**

#### 4.5.2 Résultat de défaut de la deuxième articulation

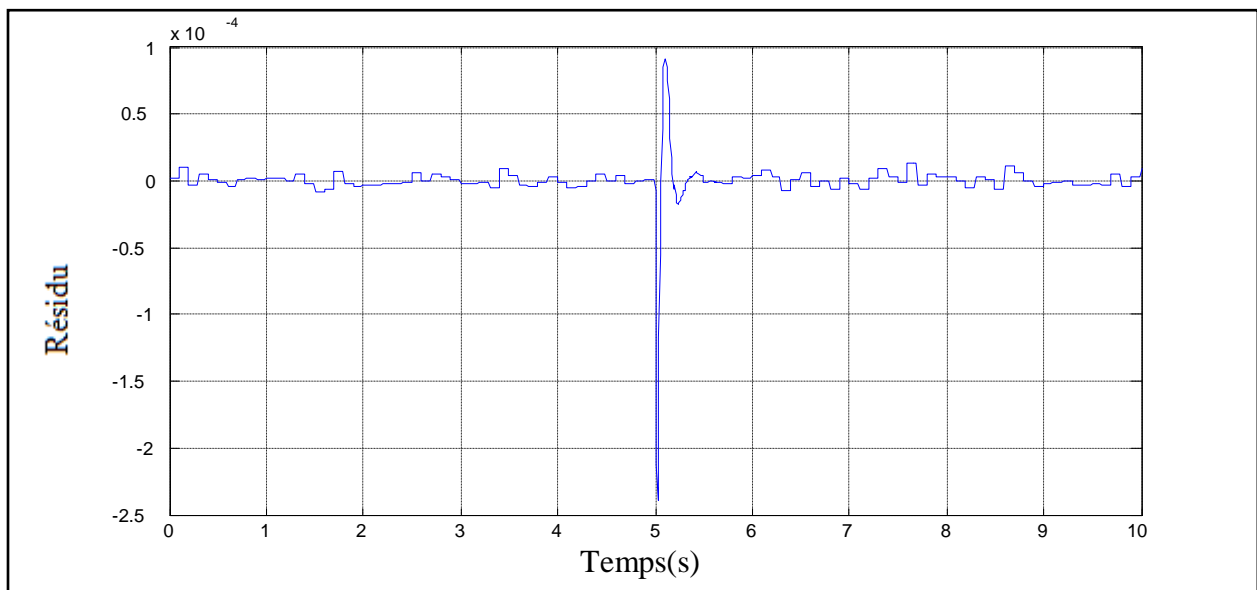
Pour un défaut dans la deuxième articulation et illustre par les figures IV.11, IV.12, IV.13, IV.14, IV.15 et IV.16. La détection par l'utilisation de la fonction de résidu F1 est distinguée à partir du temps 5.0237 s et la détection par l'utilisation de la fonction de résidu F2 est remarquée à partir du temps 5.0497 s. Tant-dis-que, le système de supervision basé sur la fonction de résidu F1 prend 0.3663 s et le système de supervision basé sur la fonction de résidu F2 prend 0.4503 s pour éliminer complètement l'effet de défaut.



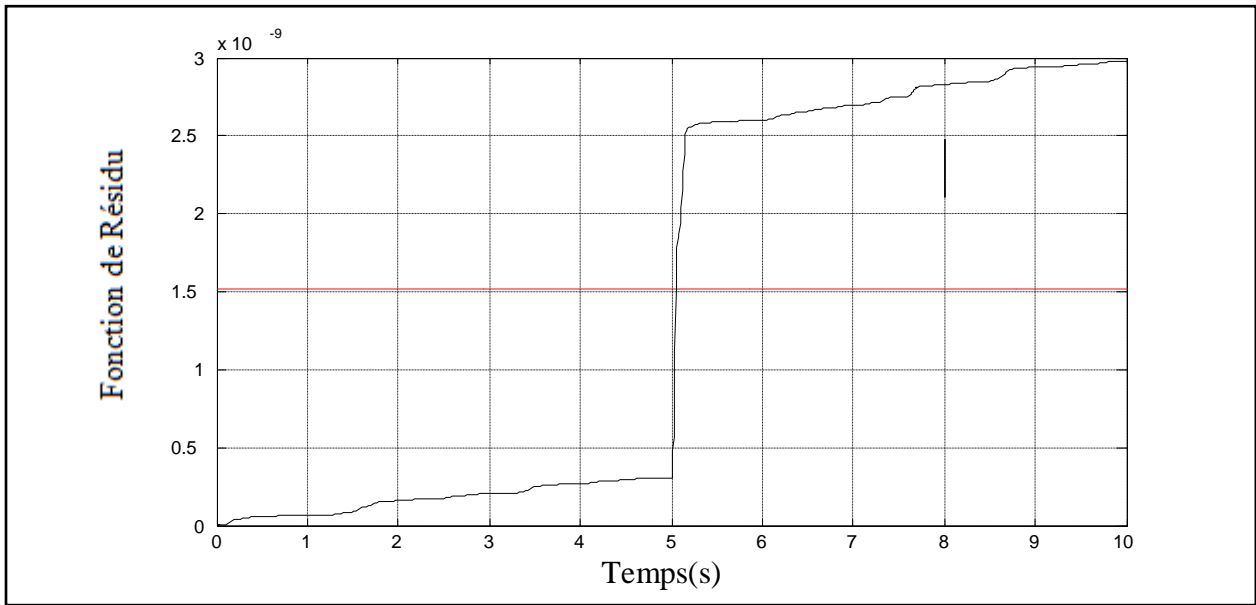
**Figure IV.11 : La fonction de Résidu F1**



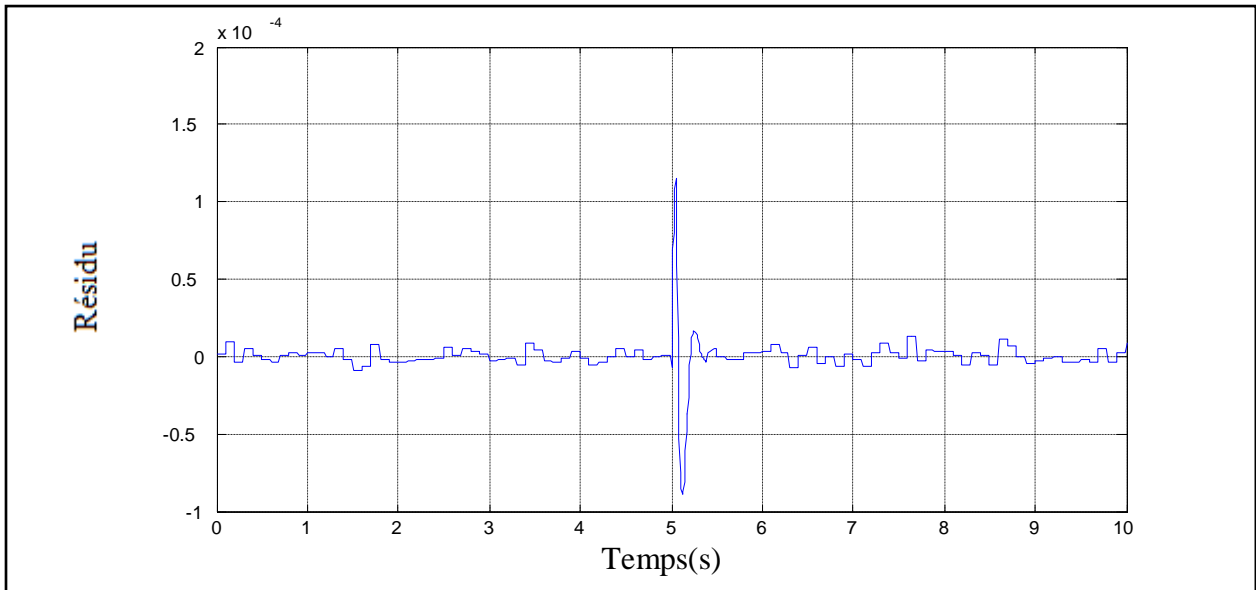
**Figure IV.12 : Résidu de première articulation**



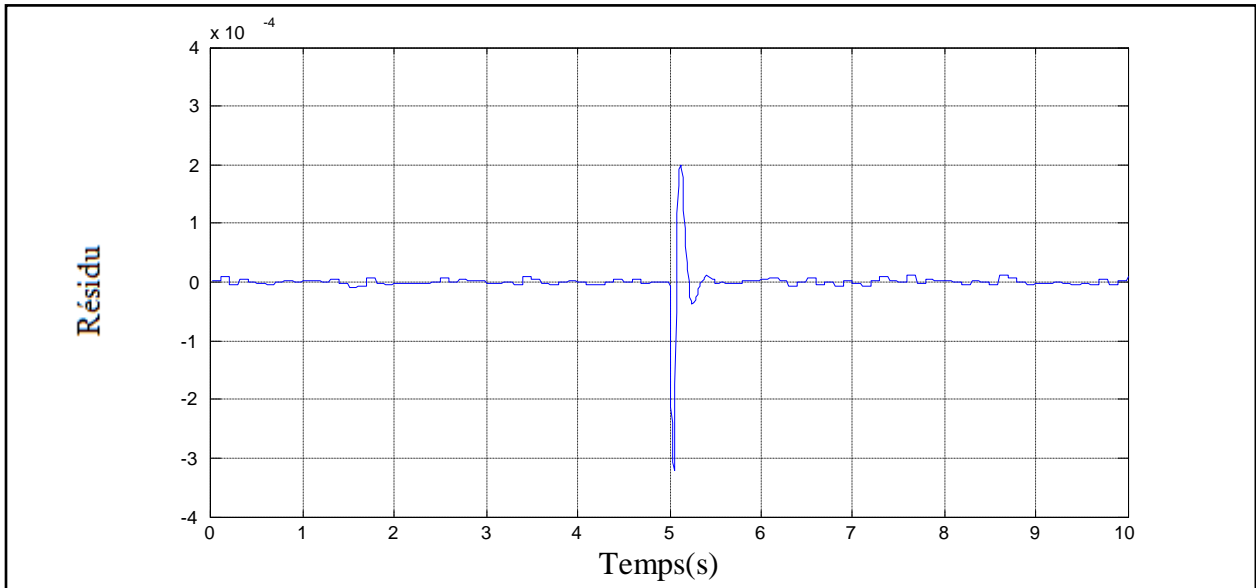
**Figure IV.13 : Résidu de deuxième articulation**



**Figure IV.14 :** La fonction de Résidu F2



**Figure IV.15 :** Résidu de première articulation



**Figure IV.16 : Résidu de deuxième articulation**

La fonction de résidu F1 est très sensible aux changements de résidu. Cette méthode est à la base de la somme de deux vecteurs résiduels à la puissance de deux de signes positifs ( $R_1^2 + R_2^2$ ). La fonction de résidu F2 est basée sur la somme de deux vecteurs ( $R_1 + R_2$ ) avec des éléments qui peuvent être de signes différents, ce qui implique la possibilité de compensation des erreurs, ce qui implique aussi que la fonction de résidu F1 offre une détection plus rapide que la fonction de résidu F2.

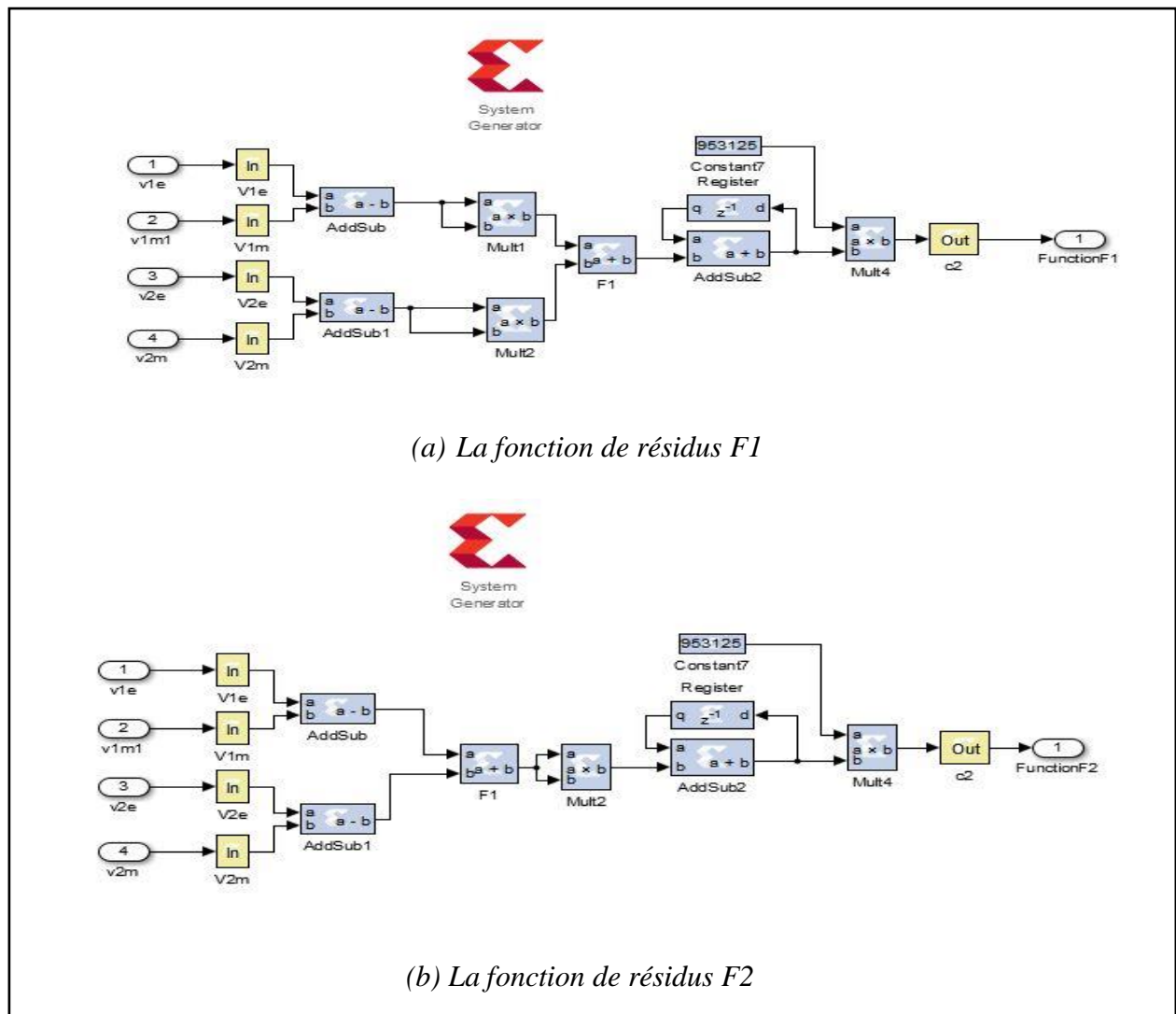
Sur les figures IV.5, IV.6, IV.7, IV.11, IV.12 et IV.13, nous remarquons aussi que les valeurs maximales de la fonction de résidu F1 sont de l'ordre de  $1.7358 \cdot 10^{-09}$  et  $2.3027 \cdot 10^{-09}$  pour la correction de défaut de première articulation et deuxième articulation respectivement, tandis que les valeurs maximales de la fonction de résidu F2 sont de l'ordre de  $4.0244 \cdot 10^{-09}$  et  $2.9816 \cdot 10^{-09}$  pour la correction de défaut de première articulation et deuxième articulation respectivement (fig IV.8, IV.9, IV.10, IV.14, IV.15 et IV.16). L'amplitude des fonctions de résidus est une fonction directe du temps de présence de défaut et des amplitudes maximales et minimales des résidus. En-effet, cette valeur augmente avec l'augmentation des deux derniers paramètres.

Les résultats de simulation montrent que les résidus indiquent la durée des défauts, et la valeur finale de vecteur des fonctions des résidus représentant l'effet de défauts sur les sorties de robot manipulateur. Il y'a possibilité de mettre une procédure de correction ou d'arrêter le robot

manipulateur suivant l'effet de défaut par la fonction de résidus qui ont été représentées. Les schémas global proposé avec l'interconnexion grâce à protocole TCP, l'envoi des valeurs des positions désirées et les vitesses de robot superviseur ont été réalisés de manière simultanée, c'est-à-dire la supervision se fait sans les inconvénients de retard de transmission.

#### 4.5.3 Résultat de synthèse de défaut en première articulation

La figure IV.17 montre l'implantation matérielle en utilisant les blocs Xilinx pour les deux cas des fonctions proposées F1 et F2.

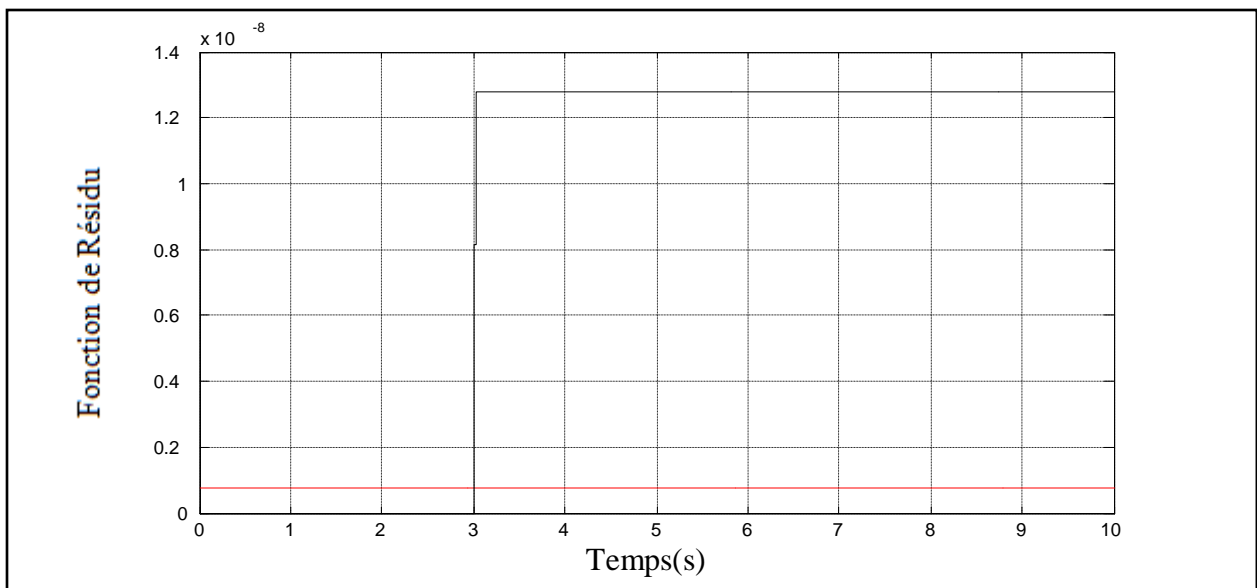


**Figure IV.17 :** Implantation du superviseur propose en utilisant les blocs Xilinx

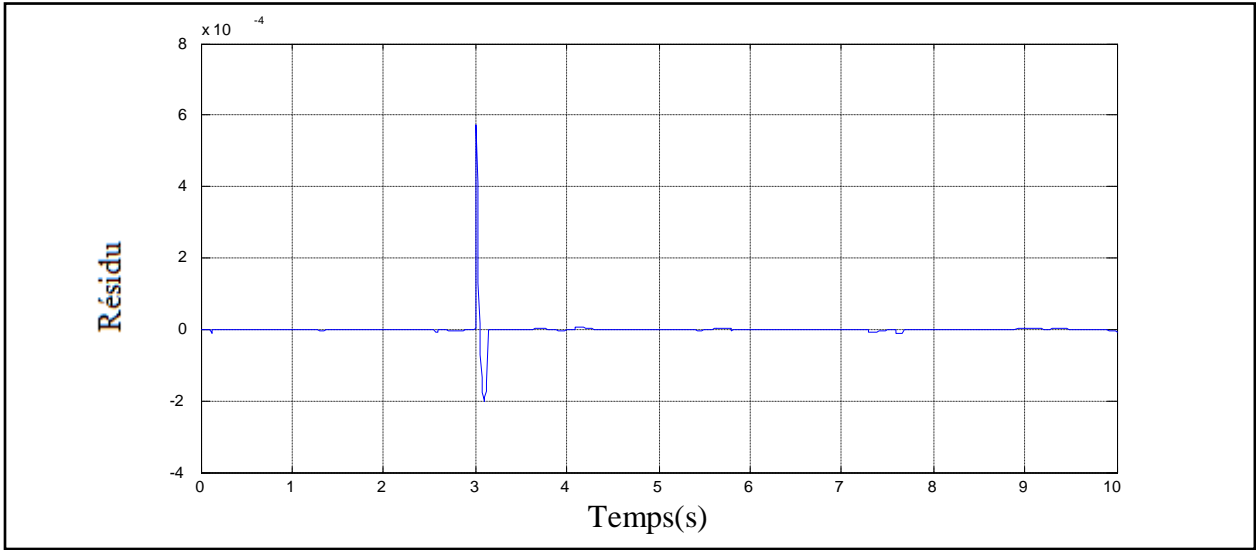
Matlab-Simulink permet la co-simulation des modèles (à base des blocs Simulink et des blocs Xilinx). En-effet, nous avons réalisé deux systèmes de robots manipulateurs : dans le premier

système nous avons utilisé le superviseur détaillé précédemment (réalisé uniquement à base des blocs Simulink) ; dans le deuxième système nous avons remplacé le superviseur uniquement par un autre réalisé uniquement à base des blocs Xilinx. Les figures IV.18, 19, 20, 21, 22 et 23 montrent les résultats de simulation de la première articulation.

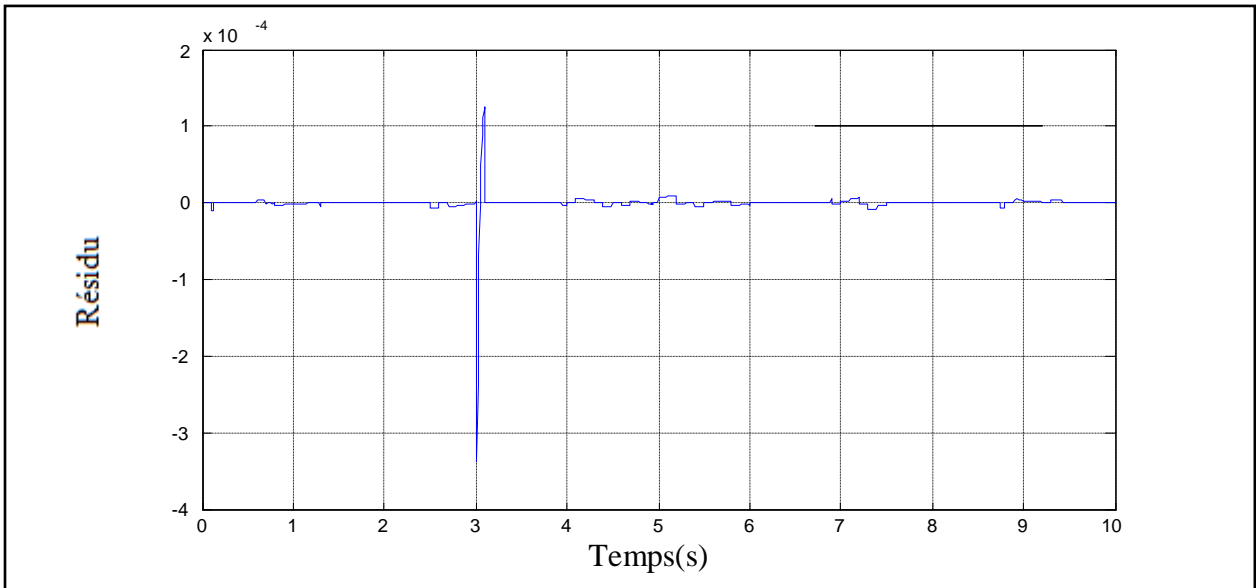
Les figures IV.18, IV.19, IV.20, IV.21, IV.22 et IV.23 illustrent la détection basée sur la fonction de résidu F1 est distinguée à partir du temps 3.01s tans-dis-que la détection basée sur fonction de résidu F2 est remarquée à partir du temps 3.02 s. La correction assurée par la fonction F1 prend 0.014 s et celle assurée la fonction F2 prend aussi 0.014 s. donc la détection par l'utilisation de la fonction F1 est la plus rapide durant le fonctionnement de robot que l'utilisation de la fonction F2.



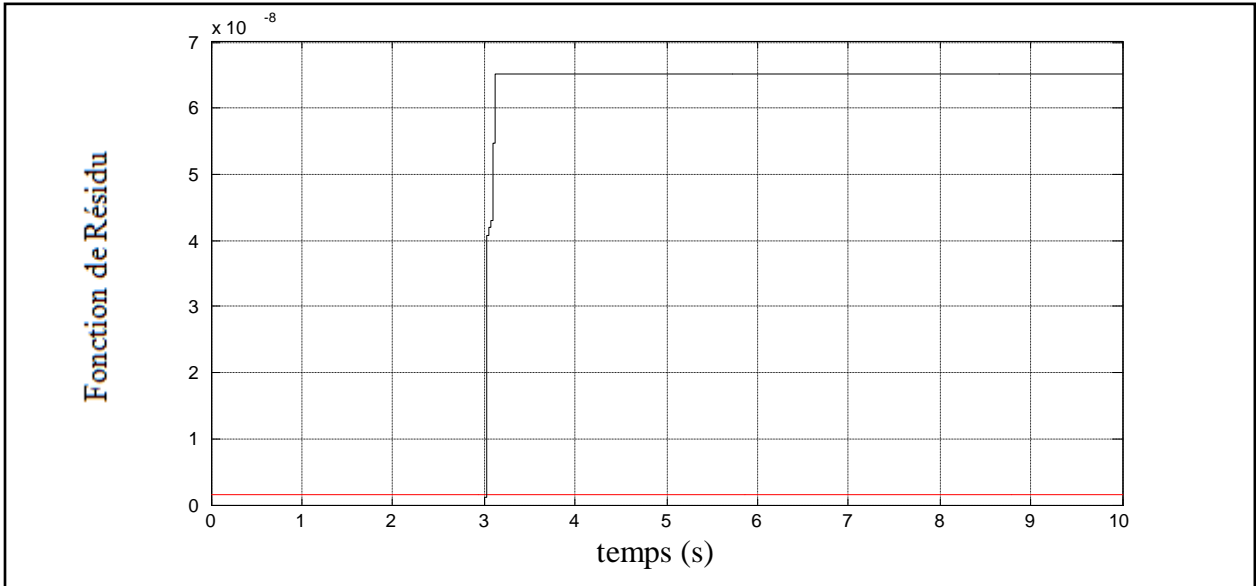
**Figure IV.18 :** La fonction de Résidu F1



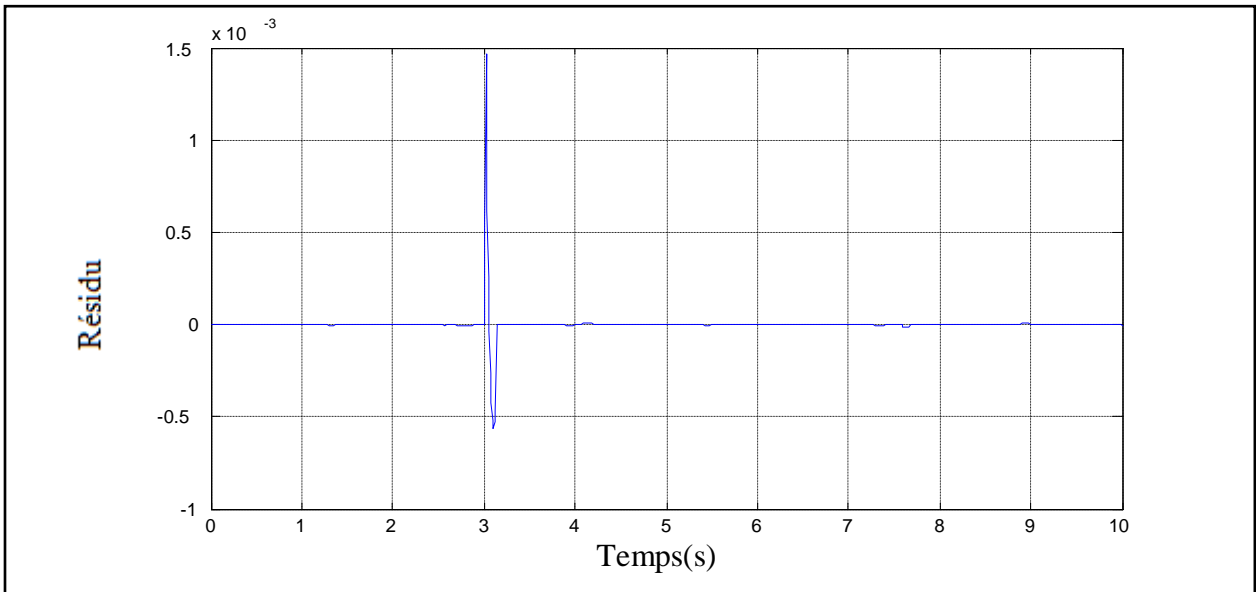
**Figure IV.19 : Résidu de première articulation**



**Figure IV.20 : Résidu de deuxième articulation**

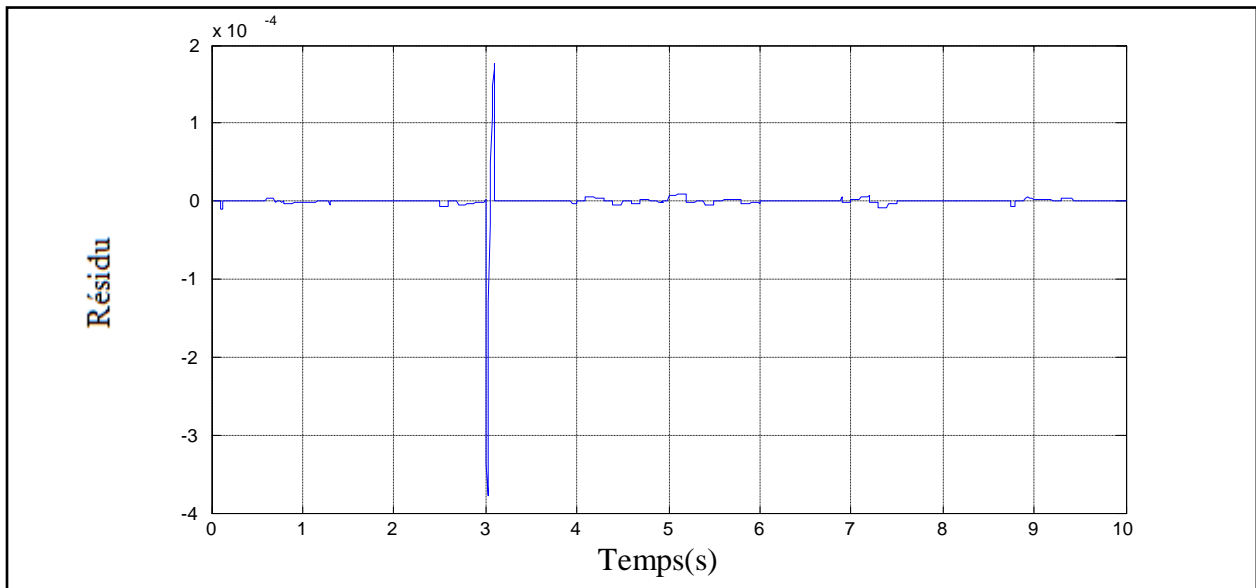


**Figure IV.21 :** La fonction de Résidu F2



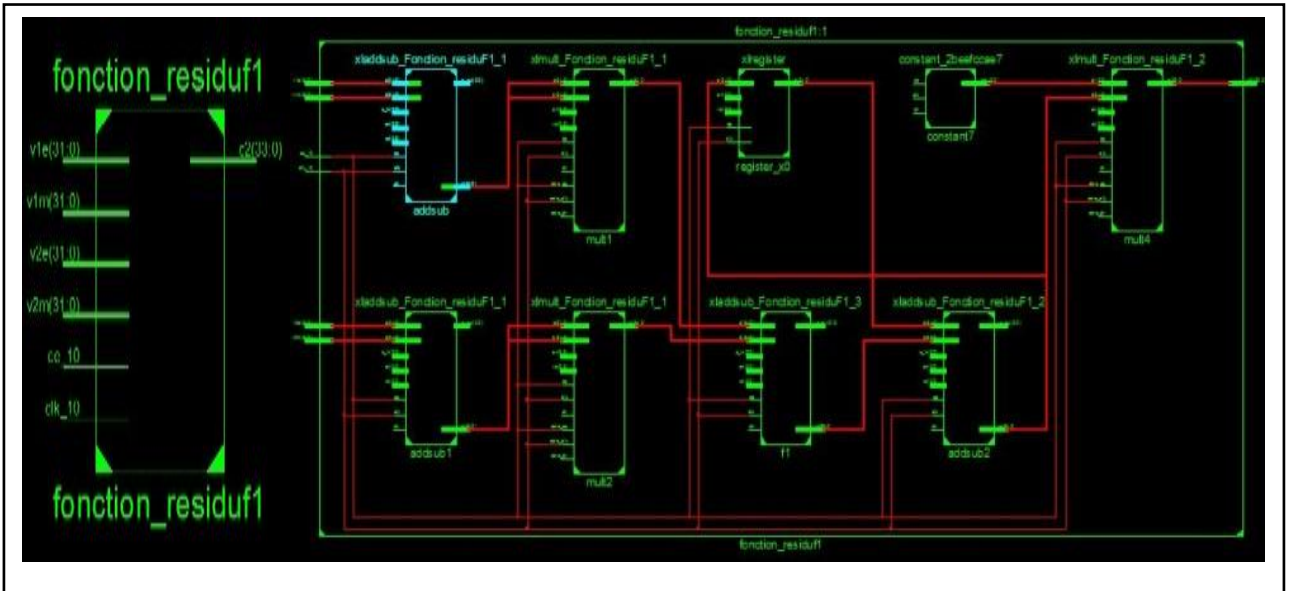
**Figure IV.22 :** Résidu de première articulation



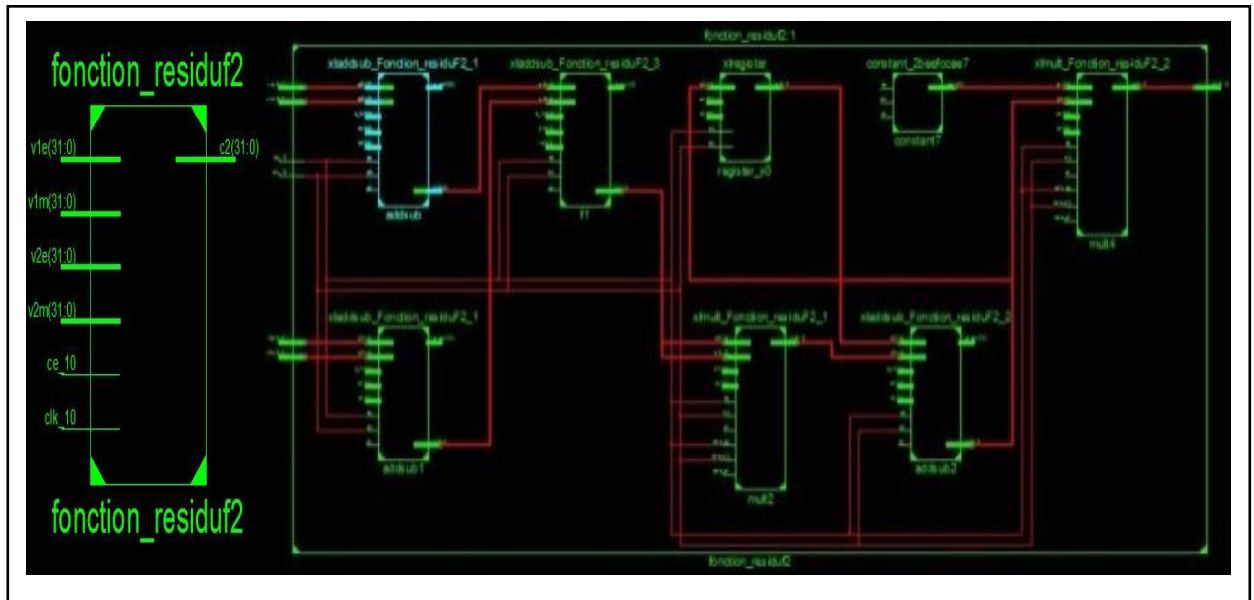


**Figure IV.23 :** Résidu de deuxième articulation

Après l'étape de simulation et de vérification fonctionnelle des schémas proposés, les System-Generator assurera la conversion automatique des modèles Xilinx uniquement en un code VHDL. En utilisant le System-Generator, nous pouvons passer aussi directement vers le code NGC ou bien vers l'implémentation directe sur FPGA. Cette dernière étape serait utile dans le cas de l'utilisation du circuit FPGA « in the loop » avec le modèle Simulink. Nous avons utilisé par la suite les outils de chez Xilinx pour la synthèse des codes VHDL en utilisant le circuit Zync kentex. Les figures IV.24 et IV.25 montrent les schémas RTL des deux codes générés pour les deux fonctions F1 et F2. Pour la fonction de résidu F1, les résultats de synthèse montrent l'utilisation de 32 slices registres et 70 slices LUTs avec une fréquence maximale de fonctionnement autour de 900MHz. Pour la fonction de résidu F1, les résultats de synthèse montrent l'utilisation de 32 slices registres et 36 slices LUTs avec la même fréquence maximale de fonctionnement. Les taux d'utilisation de circuit FPGA ne dépasse pas le 1%, ce qui donne la possibilité d'ajouter d'autres parties sur le même circuit ou bien l'ajout d'un autre bloc du schéma Simulink réalisé (comme exemple : l'implantation du modèle du robot superviseur dans le circuit FPGA).



**Figure IV.24 :** Schémas RTL des deux fonctions de résidus F1 et F2



**Figure IV.25 :** Schémas RTL des deux fonctions de résidus F1 et F2

## 4.6 Conclusion

D'après les résultats obtenus, le système de supervision proposé a bien fonctionné et a permis d'une façon générale, d'éviter la dégradation de fonctionnement du robot manipulateur a deux degrés de liberté en cas d'existence de frottement visqueux en les articulations.

L'outil d'ANFIS montre une fiabilité dans le système de supervision d'un robot manipulateur. Cet outil a été utilisé dans la deuxième partie de supervision (commande tolérante aux défauts). Le système d'ANFIS est une technique valable pour traiter les comportements des robots manipulateurs.

L'utilisation de technique de système maître et esclave rendre la supervision très claire. Ce système offre plusieurs paramètres (les données de ce système peut être utilisés dans la supervision) possibles de permettant de prendre la procédure de supervision.

Les résultats de simulation par les blocks Xilinx montrent qu'il y'a une possibilité de l'implémentation sur la carte électronique FPGA.

La communication informatique est une technique de transport d'information entre les systèmes industriels et l'opérateur, cette technique remplace des instruments de communication (les files de câblages).

# Conclusion générale et Perspectives

La supervision des systèmes robotiques est une procédure très délicate due au fait que ces derniers peuvent posséder des dynamiques complexes, des structures extrêmement variées et des frottements non linéaires. Donc, la modélisation est primordiale pour caractériser et analyser les différentes structures de commande et de supervision en robotique. L'apport principal de ce travail est le développement d'un système de supervision à base de deux techniques, la première repose sur l'intelligence artificielle et la communication d'internet, particulièrement l'approche neuro-floue qui combine les concepts de la logique floue et les réseaux de neurones artificiels pour créer un correcteur intelligent hybride appelé ANFIS, la commande tolérante aux défauts. L'autre technique est basée sur la communication d'internet et du protocole TCP pour relier le robot superviseur et le robot supervisé (robot maître et robot esclave) avec le système de supervision. Ces techniques sont utilisées pour la supervision d'un bras manipulateur à deux degrés de liberté. Nous avons proposé un système de supervision capable d'indiquer le mauvais fonctionnement lié à un défaut de frottement d'articulation et de générer une commande qui assure la reconfiguration de robot.

Dans un premier temps, nous avons étudié la structure générale d'un système de supervision typique, et accordé un intérêt particulier au diagnostic et à la commande tolérante aux défauts. Ensuite, nous avons défini différents modèles mathématiques d'un robot manipulateur : géométrique, cinématique et dynamique, et ceci pour simuler le mouvement des articulations d'un bras manipulateur, et générer les résidus et/ou les indicateurs de défauts qui sont nécessaires au diagnostic.

Le développement du modèle dynamique du robot examiné résulte de l'association des modèles dynamiques et mécaniques. A ce titre, la problématique de commande des robots ainsi que les différentes stratégies de commande, notamment les contrôleurs PD et PID, la commande adaptative ont été présentés. De même, les outils de l'intelligence artificielle (réseaux de neurones, logique flou et neuro-flou) ont été motivés et décrits. Cette étude a permis de développer un correcteur intelligent très performant.

Enfin, nous avons présenté le système de supervision et de communication par internet de protocole TCP. Le système de supervision est implémenté sur le modèle de robot manipulateur à l'aide du logiciel Matlab/Simulink, avec une simulation pour une éventuelle mise en œuvre sur la carte électronique FPGA. Comme perspectives, on peut mentionner :

- l'extension du système de supervision pour appréhender les situations de défauts multiples,
- l'implémentation sur des robots manipulateurs plus complexes,
- Assurer un post-traitement des résidus, des symptômes et/ou des indicateurs de défauts par des systèmes flous à base de règles pour obtenir des seuils adaptatifs permettant de rendre le module de diagnostic plus robuste,
- Elaboration d'une étude statistique très élaborée pour déterminer des limites de confidences pour les seuils de détection de défauts.

# Références Bibliographiques

[**Abraham & Nath, 2001**] Abraham A. et Nath B., "Hybrid intelligent systems design- a review of a decade of research", Rapport technique (5/2000), School of computing and information technology, Université de Monach, Churchill, Australia, 2001.

[**Al-Bender & J. Swevers, 2008**] Al-Bender & J. Swevers, "Characterization of friction force dynamics", IEEE Control Systems Magazine, 28(6):64-81, 2008.

[**Aldon, 1982**] Aldon M.J., "Elaboration automatique de modèles dynamiques de robots en vue de leur conception et de leur commande", Thèse d'Etat, USTL, Montpellier, France, oct. 1982.

[**Armstrong, 1979**] Armstrong W.W., "Recursive solution to the equation of motion of an N-links manipulator", Proc. 5<sup>th</sup> World Congress on Theory of Machines and Mechanisms, Montréal, Canada, p. 1343-1346, 1979.

[**Atakan, 2009**] Atakan Doğan, "A study on performance of dynamic file replication algorithms for real-time file access in Data Grids ", Future Generation Computer Systems, v. 25, p. 829-839, 2009.

[**Baroni et al, 1999**] Baroni P., Lamperti G., Pogliano P. et Zennella M., "Diagnosis of large active systems", Artificial Intelligence Journal, v. 110 (1), p. 135-183, May 1999.

[**Basseville et al., 1996**] Basseville, M. et M-O. Cordie, "Surveillance et diagnostic de systèmes dynamiques: approche complémentaire du traitement de signal et de l'intelligence artificielle", Rapport INRIA, no. 2861, 1996.

[**Blanke M. et al., 2003**] Blanke M., Kinnaert M., Lunze M. & Staroswiecki M., "Diagnosis and fault tolerant control", Springer-Verlag, 2003.

[**Borrel, 1986**] Borrel P., "Contribution à la modélisation géométrique des robots manipulateurs ; application à la conception assistée par ordinateur", Thèse d'Etat USTL Montpellier, France, juillet 1986.

[**Brandl, 1986**] Brandl H., Johanni R., Otter M., "A very efficient algorithm for the simulation of robots and multibody systems without inversion of the mass matrix", Proc. IFAC Symp, On Theory of Robots, Vienne. Autriche. p. 365-370, déc. 1986.

[**Burdick, 1988**] Burdick J.W., "Kinematic analysis and design of redundant manipulators", Thèse de PhD, Université de Stanford, Stanford, USA, 1988.

[**Burns, 2001**] Burns, R.S. "Advanced Control Engineering", Butterworth-Heinemann ed, 2001.

[**Canudas et al. 1992**] Canudas de Wit C., B. Brogliato, A. De Luca, B. Siciliano, P. Tomei, C. Samson, R. Ortega, R. Lozano, G. Bastin, G. Campion, B. D'Andrea-Novell, W. Khalil, "Theory of

robot control", Ecole d'Été d'Automatique de Grenoble, Laboratoire d'Automatique de Grenoble, France, Sep. 7-11. 1992.

[**Cesareo, 1984**] Cesareo G., Nicolo F., Nicosia S., "DYMIR: a code for generating dynamic model of robots", Proc. IEEE Int. Conf. on Robotics, Atlanta, p. 115-120, mars 1984.

[**Chien & Huang, 2007**] Chien, M.-C. & A.-C. Huang, "Adaptive control for flexible-joint electrically driven robot with time-varying uncertainties", IEEE Transactions on Industrial Electronics, v. 54(2), p. 1032–1038, 2007.

[**Choi & al., 2001**] Choi, Y., W. K. Chung & I. H. Suh, "Performance and optimality of pid trajectory tracking controller for lagrangian systems", IEEE Transactions on Robotics and Automation, v. 17(6), p. 857–869, 2001.

[**Chow & Willsky, 1984**] Chow E.Y., Willsky A.S., "Analytical redundancy and the design of robots failure detection system", IEEE Transactions, Automatic Control Journal, v. 29, no.7, p.603-614, 1984.

[**Combacau, 1991**] Combacau, M., "Commande et Surveillance des Systèmes à Événements Discrets Complexes : Application aux Ateliers Flexibles", Thèse de doctorat, Université Paul Sabatier, Toulouse, 1991.

[**Constantin, V.A., 1995**] Constantin, V.A., "Fuzzy logic and neuro-fuzzy applications explained", Englewood Cliffs, Prentice- Hall, 1995.

[**Cyril, 2009**] Cyril Quennouelle, "Modélisation Géométrico-Statique Des Mécanismes Parallèles Complains", Thèse de doctorat, Université Laval, Québec, 2009.

[**Dash et al., 2000**] Dash, S. et V. Venkatasubramanian., "Challenges in the industrial applications of fault diagnostic systems", Proceedings of the Conference on Process Systems Engineering Computing & Chemical Engineering, Keystone, Colorado, p. 785-791, 2000.

[**Denavit, 1955**] Denavit J., Hartenberg R.S., "A kinematic notation for lower pair mechanism based on matrices", Trans, of ASM E, J. of Applied Mechanics, v. 22. p. 215-221, juin 1955.

[**Dillon, 1973**] Dillon S.R., "Computer assisted equation generation in linkage dynamics", Thèse de PhD, Université d'État de l'Ohio, USA, Aout 1973.

[**Dubuisson, 1990**] Dubuisson, B., "Diagnostic et reconnaissance des formes", Paris, Edition Hermès, 1990.

[**Dubuisson et al., 2001**] Dubuisson, B., E. Boutleux, P. Dague, T. Denoeux, E. Didelet, Y. Gandvalet et M. Masson, "Diagnostic, Intelligence Artificielle et reconnaissance de formes", Edition Hermes, 2001.

[**Dombre et Khalil, 1988**] Dombre, E. et Khalil, W., "Modélisation et commande des robots", Hermès, 1988.

[**Dombre & Khalil, 2002**] Dombre, E. & Khalil, W., "Modeling, Identification and Control of Robots", 3<sup>ème</sup> édition, Taylor and Francis Group, New York, 2002.

[**Dreyfus et al, 2002**] Dreyfus G., Martinez J., Samuelides M., Gordon M. , Badran F., Thiria S. et Hérault L., "Réseaux de neurones, méthodologies et application", Edition Eyrolles, 2002.

[**Featherstone, 1983b**] Featherstone R., "The calculation of robot dynamics using articulated-body inertias", The Int. J. of Robotics Research, v. 2(3), p. 87-101, 1983.

[**Fussel & Isermann, 1998**] Fussel D. & Isermann R., "Model-based fault detection and diagnosis methods Part A: methods", Proceeding of the 4<sup>th</sup> International Conference on Probabilistic Safety Assessment and Management, PSAM4, New York, USA, 1998.

[**Fuller, 1995**] Fuller R., "Neural fuzzy systems", Université de Abo Akademi, ESF Séries A: 443, 1995.

[**Ghrieb A. et al, 2017**] Ghrieb Abdel Ouahab, Kourid Yahia & Guersi Noureddine, "Supervision of industrial manipulators using ANFIS system", International Conference on Control, Automation and Systems, Jeju, Korea, Oct. 18-21, 2017.

[**Ghrieb A. & Kourid Y., 2018**] Ghrieb Abdel Ouahab, KOURID Yahia "Supervision of industrial manipulators station by analytical corrector", IEEE-2018 International Conference on Control, Automation and Diagnosis (ICCAD'18), Marrakech-Morocco, March 19-21/2018.

[**Ghrieb A. et al, 2019**] Abdel Ouahab Ghrieb, Yahia Kourid, Kamel Messaoudi, Djamel M. Mouss, Toufik Bakir "Hardware implementation using XSG of new fault detection method applied to robot manipulator", international journal of Mechatronic Systems and Control. DOI: 10.2316/J.2020.201-0011, 2020.

[**Ghorbel et al., 1989**] Ghorbel, Fathi, John Y. Hung & Mark W. Spong, "Adaptive control of flexible-joint manipulators", IEEE Control Systems Magazine, v. 9(7), p. 9–13, 1989.

[**Guillaume, 2004**] Guillaume BCQE, "Contribution au développement d'un système portable automatique d'aide à la détermination de l'état d'urgence d'un blessé polytraumatisé, apprentissage supervisé de classes de choc hémorragique", Thèse de doctorat, Université de Grenoble 1, 2004.

[**Izaguirre, 1986**] Izaguirre A., Paul R.C.P., "Automatic generation of the dynamic equations of the robot manipulators using a LISP program", Proc. IEEE Int. Conf. on Robotics and Automation, San Francisco, p. 220-226, avril 1986.

[**Jang, J.-S.R., 1993**] Jang, J.-S.R., "Anfis: adaptive-network-based fuzzy inference system", IEEE Transaction on System Man and Cybernetics, v. 23(3), p. 665–685, 1993.



**[Khalil, 1976]** Khalil W., "Modélisation et commande par ordinateur du manipulateur MA- 23 ; extension à la conception par ordinateur des manipulateurs", Thèse de Docteur- Ingénieur, USTL, Montpellier, sept. 1976.

**[Khalil, 1989]** Khalil W., Bennis F., Chevallereau C., Kleinfinger J.-F., "SYMORO: à software package for the symbolic modelling of robots", Proc. 20th Int. Symp, on Industrial Robots. Tokyo, p. 1023-1030, oct. 1989.

**[Khedhri, 2002]** Khedhri K., "Modélisation et simulation d'une application industrielle par le générateur de systèmes experts G2", Thèse de Magister, Université de Batna, Algerie, 2002.

**[Khodja, 1997]** Khodja L., "Contribution à la classification floue non supervisée", Thèse de doctorat, Université de savoie, novembre 1997.

**[Kim et al., 1997]** Kim, E., M. Park, S. Ji & M. Park, "A new approach to fuzzy modeling", IEEE Transactions on Fuzzy Systems, v. 5(3), p. 328–337, 1997.

**[Klir & Yuan, 1994]** Klir, G.J. & B. Yuan, "Fuzzy sets and fuzzy logic: theory and applications", PrenticeHall. Inc. Upper Saddle River, NJ, USA, 1994.

**[Kourd, 2012]** Kourd Yahia, "Génération des résidus par les outils de l'intelligence artificielle pour le diagnostic des systèmes complexes", Thèse de doctorat, Université de Badji Mokhtar, Juin 2012.

**[Kreuzer, 1979]** Kreuzer E.J., "Dynamical analysis of mechanisms using symbolical equation manipulation", Proc. 51<sup>th</sup> World Congress on Theory of Machines and Mechanisms, Montréal, p. 599-602, 1979.

**[Lewis et al. 1993]** Lewis F.L., C.T. Abdellah & D. M. Dawson, "Control of robot manipulators", New York, Macmillan, 1993.

**[Lozano & Brogliato, 1992]** Lozano, R. & B. Brogliato, "Adaptive control of robot manipulators with flexible joints", IEEE Transactions on Automatic Control, v. 37(2), p. 174–181, 1992.

**[Man & Halang, 1997]** Xiang, T., K.F. Man, K.M. Luk & C.H. Chan, "Design of multiband miniature handset antenna by mom and hga", IEEE Antennas and wireless propagation letters 5, 179–182, 1997.

**[Megahed, 1982]** Megahed S., Renaud M., "Minimization of the computation time necessary for the dynamic control", Proc. 12<sup>th</sup> Int. Symp, on Industrial Robots, Paris, p. 469-478, juin 1982.

**[Megahed, 1984]** Megahed S. "Contribution à la modélisation géométrique et dynamique des robots manipulateurs ayant une structure de chaîne cinématique simple ou complexe ; application à leur commande", Thèse d'Etat. UPS. Toulouse, juillet 1984.

[**Melha, 2005**] Melha Bitam, "Modélisation et étude de comportement d'une ligne de communication TCP/IP", Automatique / Robotique. Université Joseph-Fourier - Grenoble I, 2005.

[**Moberg, 2007**] Moberg, S., "Modeling and Control of Flexible Manipulators", Thèse de doctorat. Linkoping University. Linkoping, Sweden, 2007.

[**Murray, 1984**] Murray J.J., Newman C.P., "ARM: an algebraic robot dynamic modeling program", Proc. IEEE Int. Conf, on Robotics and Automation, Atlanta, p. 103-104, mars 1984.

[**Nan Zhang, 2010**] Nan Zhang, "Détection et Isolation de Pannes basées sur la Platitude Différentielle Application aux Engins Atmosphériques", Thèse de doctorat, Université de Toulouse France, 2010.

[**Nauck, 1997**] D. Nauck, "Neuro-fuzzy systems: review and prospects", European congress on intelligent technique and soft computing (EUFIT'97), Aachen, p.1044-1053, 1997.

[**Paul, 1981**] Paul R.C.P., "Robot manipulators: mathematics, programming and control", MIT Press. Cambridge, 1981.

[**Pedrycz & Zadeh, 1995**] Pedrycz, W. & L.A. Zadeh, "Fuzzy Set Engineering", CRC Press. Boca Raton, USA, 1995.

[**Pickholtz R.L. et al, 1982**] Pickholtz R.L. et al., "Theory of spread spectrum communications - A tutorial", IEEE Trans. Commun., v. COM-30, p. 855-884, May 1982.

[**Pieper, 1968**] Pieper D.L., "The kinematics of manipulators under computer control", Ph.D. Thesis, Stanford University, 1968.

[**Racoceanu, 2006**] Racoceanu D., "Contribution à la surveillance des systèmes de production en utilisant les techniques de l'intelligence artificielle", Université de Franche-Comté, France, 2006.

[**Raghavan, 1990**] Raghavan M. & Roth B., "Inverse kinematics of the general 6R manipulator and related linkages", Trans, of the ASME, J. of Mechanical Design, v. 115, p. 502-508, 1990.

[**Renato et al, 2007**] Renato Tinos, Marco Henrique Terra & Marcel Bergerman, "A fault tolerance framework for cooperative robotic manipulators", Control Engineering Practice v. 15 no.5 p. 615-625, 2007.

[**Renaud, 1985**] Renaud M., "A near minimum iterative analytical procedure for obtaining a robot-manipulator dynamic model", LUTAM/IFTOMM Symp, on Dynamics of Multi body Systems, Udine, 1985.

[**Rutkowski, 2004**] Rutkowski L., "Flexible neuro-fuzzy systems, Structures, Learning and Performance Evaluation", Kluwer Academic Publishers, New York, 2004.

[**Saleh Ahmad et al, 2013**] Saleh Ahmad, Hongwei Zhang & Guangjun Lu, "Distributed fault detection for modular and reconfigurable robots with joint torque sensing: A prediction error based approach", *Mechatronics*, v. 23, p. 607–616, 2013.

[**Sastry et al., 1994**] Sastry, P.S., G. Santharam & K.P. Unnikrishnan, "Memory neuron networks for identification and control of dynamical systems", *IEEE Transactions on Neural Networks* v. 5(2), p. 306–319, 1994.

[**Tao et al, 2013**] Tao Ni, Hongyan Zhang, Peng Xu & Hironao Yamada, "Vision-based virtual force guidance for tele-robotic system", *Computers and Electrical Engineering*, v. 39, p. 2135–2144, 2013.

[**Tesheng & Mao-Chiao Weng, 2011**] Tesheng et Mao-Chiao Weng, "A hierarchical multiple-model approach for detection and isolation of robotic actuator faults", *Robotics and Autonomous Systems*, 60, 154–166, 2011.

[**Tomei, 1991**] Tomei, P., "A simple pd controller for robots with elastic joints", *IEEE Transactions on Automatic Control* 36(10), 1208–1213, 1991.

[**Toguyeni, 1992**] Toguyeni, A.K.A., "Surveillance et diagnostic en ligne dans les ateliers flexibles de l'industrie manufacturière", Thèse de doctorat, Université de Lille, 1992.

[**Tromp, 2000**] Tromp, L., "Surveillance et diagnostic de systèmes industriels complexes: une approche hybride numérique/symbolique", Thèse de doctorat, Université de Rennes 1, 2000.

[**Villemeur, 1988**] Villemeur, A., "Sûreté de fonctionnement des systèmes industriels", Edition Eyrolles, Collection DER-EDF, v. 67, 1988.

[**Vukobratovic & Kircanski, 1984**] Vukobratovic, M. & Kircanski, M., "A dynamic approach to nominal trajectory synthesis for redundant manipulators", *IEEE Trans. Systems Man Cyber*, 14:580–586, 1984.

[**Walker, 1982**] Walker M.W. & Orin D.E., "Efficient dynamic computer simulation of robotics mechanism", *Trans, of ASME, J. of Dynamic Systems. Measurement, and Control*, v. 104, p. 205–211, 1982.

[**Wang & Langari, 1996**] Wang, L. & R. Langari, "Complex systems modeling via fuzzy logic", *IEEE Transactions On Systems, Man, And Cybernetics-Part B: Cybernetics*, v. 26(1), p. 100–106, 1996.

[**Wang W. Q. et al., 2002**] Wang, X., G. Yu, M. Koç & J. Lee, "Wavelet Neural Network for Machining Performance Assessment and Its Implications to Machinery Prognostics. In: *Managing*

Innovative Manufacturing", e-Manufacturing and e-Business Intégration (MIM'2002), Milwaukee, USA, 2002.

[**Wenger, 1989**] Wenger P., "Aptitude d'un robot manipulateur à parcourir son espace de travail en présence d'obstacles", Thèse de Doctorat, ENSM. Nantes, sept. 1989.

[**Whitney, 1969**] Whitney D.E., "Resolved motion rate control of manipulators and human prostheses", IEEE Trans, on Man Machine Systems, v. MMS-10(2), p. 47-53, juin 1969.

[**Yam et al., 2001**] Yam, JR.C.M., P.W. Tse, L. Li et P. Tu, "Intelligent prédictive décision support System for condition based maintenance", International Journal of Advanced Manufacturing Technology, v.17, p. 383-391, 2001.

[**Xinxing & Hironao, 2011**] Xinxing Tang & Hironao Yamada, "Tele-operation Construction Robot Control System with Virtual Reality Technology", Procedia Engineering, v.15, p.1071 – 1076, 2011.

[**Zabala, 1978**] Zabala Iturralde J., "Commandes des robots manipulateurs à partir de la modélisation de leurs dynamique", Thèse de Troisième Cycle, UPS, Toulouse, juillet 1978.

[**Zadeh, 1965**] Zadeh, L.A., "fuzzy sets", Informatic and Control, v. 8, p. 338–353, 1965.

[**Zemouri, 2003**] Zemouri M., "Contribution à la surveillance des systèmes de production à l'aide des réseaux de neurones dynamiques : application à la maintenance", Thèse de doctorat de l'université de Franche-Comté, France, 2003.

[**Zhang, 1998**] Zhang, G., B.E. Patuwo and M.Y. Hu., "Forecasting with artificial neural networks: thé state of thé art", International Journal of Forecasting, v.14, p.35-62, 1998.

[**Zwingelstein, 1995**] Zwingelstein, G,"Diagnostic des défaillances : Théorie et pratique pour les systèmes industriels", Editions Hermès, 1995.