

**République Algérienne Démocratique et Populaire**  
**Ministère de l'Enseignement Supérieur et de la Recherche Scientifique**  
**Université Mouloud MAMMERY, Tizi-Ouzou**



Faculté de Génie Electrique et d'Informatique  
Département d'Automatique

**MEMOIRE DE FIN D'ETUDES**

En vue de l'obtention du diplôme

*De MASTER ACADEMIQUE EN AUTOMATIQUE*  
*OPTION : COMMANDE DES SYSTEMES*

*Thème*

**Commande neuro-floue d'une architecture  
de téléopération**

Proposé par : **M. Mellah Rabah**

Présenté par :

Dirigé par : **M. Mellah Rabah**

**M. LADJEL Lamara**  
**M. RAAB Rafik**

Soutenu le : / 07/2013

*Promotion 2013*

## **Remerciements**

*On tient à exprimer toute nos reconnaissances à notre promoteur Mr Mellah Rabah pour l'honneur qu'il nous a fait en assurant la direction et le suivi scientifique et technique du présent mémoire.*

*On remercie vivement le président et les membres de jury pour l'honneur qu'ils ont fait en acceptant de juger ce modeste travail.*

*Nos remerciements s'adressent également à Mr ChARIF MOUSSA et Mr SAIDI KHEREDINE pour leurs aides et leurs soutiens sans oublier l'ingénieur de laboratoire de l'automatique industrielle.*

# *Dédicaces*

*A qui puis-je dédier cet humble travail si ce n'est à mes très chers parents, dont le sacrifice, la tendresse, l'amour, la patience, le soutien, l'aide, l'encouragement. Sans eux, je ne serais pas ce je suis aujourd'hui.*

*Je dédie ce travail également,*

*A mes frères et sœurs.*

*A mon binôme.*

*A tous mes amis, que ce soit à Oued Aïssi ou à mon village Iguersafene.*

*A toute personne qui porte de l'estime pour moi.*

**Rafik**

# *Dédicaces*

*A qui puis-je dédier cet humble travail si ce n'est à mes très chers parents, dont le sacrifice, la tendresse, l'amour, la patience, le soutien, l'aide, l'encouragement. Sans eux, je ne serais pas ce je suis aujourd'hui.*

*Je dédie ce travail également,*

*A mes frères et sœur.*

*A mon binôme.*

*A tous mes amis, que ce soit à Oued Aïssi ou à mon village agouni fourou.*

*A toute personne qui porte de l'estime pour moi.*

*Lamara*

## **Résumé**

Le but de ce travail est l'étude, commande et réalisation d'une architecture de téléopération bilatérale composée d'un site maître qui est représenté par un moteur à courant continu avec encodeur qui entraîne une tige et un site esclave identique. Ces deux sites sont reliés à la partie commande par une carte d'acquisition afin d'assurer la communication entre eux.

Lorsque l'opérateur change la position angulaire pour le maître, ce déplacement sera répliqué sur le site esclave.

Cette application a été commandée à l'aide d'un régulateur classique et un régulateur neuro-flou.

Après avoir utilisé les deux commandes et tiré les conclusions et les bases théoriques nécessaires on a implémenté ces commandes sur la réalisation pratique citée ci-dessus pour avoir des bons résultats.

### **Mots clés**

Téléopération ; Télémanipulateur ; Robotique ; Arduino ; Moteur à courant continu ; asservissement bilatéral ; régulateur classique PID ; régulateur Neuro-flou.

Remerciements  
Dédicaces

Introduction générale.....1

## **Chapitre I : Généralités sur une architecture de téléopération**

### Première partie : Généralités sur la téléopération

I.1 Introduction .....	3
I.2 bref historique.....	3
I.3 La télé robotique.....	4
I.4 quelques domaines d’application de la télé robotique.....	5
I.4.1 Le domaine nucléaire .....	5
I.4.2 Le domaine spatial.....	5
I.4.3 Le domaine militaire.....	6
I.4.4 Le domaine sous-marin.....	6
I.5 Modélisation du système.....	6
I.5.1 Modélisation du comportement de l’opérateur humain.....	6
I.5.2 Modélisation de l’environnement.....	7
I.5.3 Modélisation des manipulateurs maître et esclave.....	7
I.5.4 Modélisation du réseau de communication.....	7
I.6 Les systèmes de télé opération avec retour d'effort.....	8
I.6.1 Asservissement bilatéral.....	8
I.7 Les systèmes de télé opération sans retour d'effort.....	8
I.8 Robot esclave hybride.....	9
I.9 Pilotage des déplacements.....	9
I.10 Performance et stabilité en télé opération .....	9
I.10.1 Stabilité.....	9
I.10.2 Transparence.....	10
I.10.3 Passivité .....	10
I.11 Stratégies de commande bilatérale.....	11
I.11.1 Structure à deux canaux position-position.....	11
Deuxième partie: dispositif de contrôle et de commande.....	12
I.12 Design du banc de test .....	12
I.13 Design électromécanique.....	12
I.14 Capteurs.....	13
I.15 Carte de Contrôle.....	13
I.16 Etude d’un système de télé opération.....	13
I.16.1 Représentation d’un système de télé opération.....	13
I.16.2 Critères de performances.....	15
I.16.3 Contrôle des systèmes de téléopération bilatérale .....	17
I.16.3.1 Méthodes de contrôle position-position.....	17
I.17 Conclusion.....	18

## Chapitre II : Présentation de la carte Arduino

II.1 Introduction.....	19
II.2 La carte d'acquisition .....	19
II.2.1 Définition de la carte Arduino.....	19
II.2.2 Historique.....	19
II.2.3 Caractéristique de la carte Arduino.....	20
II.3 Architecture de la carte .....	21
II.3.1 Le microcontrôleur.....	21
II.3.1.1 Caractéristiques du microcontrôleur ATMEGA 2560 .....	21
II.3.2 Le entrées-sorties de la carte.....	22
II.3.3 Mémoire .....	22
II.3.4 L'alimentation.....	22
II.3.5 Visualisation.....	23
II.3.6 La connectique.....	23
II.3.7 Carte d'extension « SHIELD ».....	23
II.3.7.1 Exemple d'usage de SHIELD pour connecter des moteurs à courant continu.....	24
II.4 Architecture SOFTWARE de la carte.....	24
II.4.1 Logiciel.....	24
II.4.1.1 L'interface logicielle .....	24
II.4.1.2 Structure de logiciel.....	24
II.4.1.3 Structure de programme.....	25
II.4.1.4 Programmation.....	25
II.4.2 Développement d'un projet.....	26
II.4.3 Utilisation de la carte Arduino comme périphérique d'E/S pour Matlab.....	26
II.4.3.1La bibliothèque Arduino IO.....	27
II.5 Logiciel à utiliser dans la suite de projet.....	27
IV.6 Conclusion.....	27

## Chapitre III : Commande à base de régulateurs classiques

### Ajustage des Paramètres de Régulateurs PID

III.1 Introduction .....	28
III.2 Principe général de la régulation.....	28
III.3 Les caractéristiques du régulateur PID.....	28
III.4 Asservissements et influence des coefficients.....	29
III.4.1 Action proportionnelle P .....	29
III.4.2 Action intégrale I .....	29
III.4.3 Action dérivé D .....	30
III.4.4 Asservissement PID.....	30
III.5 Réglage des coefficients.....	31
III.5.1 Méthode de Ziegler-Nichols.....	32
III.6 Description et modélisation du moteur à courant continu .....	32

III.6.1	Caractéristiques des moteurs utilisés.....	32
III.6.2	Mise en œuvre.....	33
III.6.2.1	Partie mécanique.....	33
III.6.2.2	Le principe de fonctionnement de l'encodeur .....	33
III.6.3	Partie électronique.....	33
III.6.3.1	Commande d'un moteur avec un pont en H.....	34
III.6.3.1.1	Principe de fonctionnement .....	34
III.6.4	Partie informatique embarquée.....	35
III.6.4.1	Acquisition des données.....	35
III.6.5	Partie informatique distante.....	35
III.7	La théorie du PID appliquée aux moteurs électriques.....	36
III.7.1	Modélisation mathématique du moteur à courant continu.....	36
III.8	Limitations et approximations.....	37
III.8.1	Limitation de la tension de commande.....	37
III.8.2	Approximations.....	37
III.9	Modèles simulés sous MATLAB/SIMULINK.....	37
III.10	Simulations et résultats expérimentaux.....	38
III.10.1	Cas théoriques.....	38
III.10.2	Cas pratiques.....	39
III.10.3	Interprétation des résultats .....	40
III.10.3.1	Cas théorique .....	40
III.10.3.2	Cas pratique .....	40
III.11	Conclusion.....	41

## Chapitre IV : Commande à base de régulateurs neuro flous

IV.1	Introduction.....	42
IV.2	Notions sur les réseaux de neurones artificiels .....	42
IV.2.1	Introduction.....	42
IV.2.2	Le neurone.....	42
IV.2.3	Fonctionnement.....	43
IV.3	Notions sur l'apprentissage.....	44
IV.3.1	Types d'apprentissage.....	44
IV.3.1.1	Apprentissage non supervisé.....	45
IV.3.1.2	Apprentissage supervisé.....	45
IV.3.1.2.1	La retro-propagation du gradient.....	45
IV.4	Logique floue.....	47
IV.4.1	Notion de sous-ensemble flou.....	47
IV.4.2	Univers de discours.....	47
IV.4.3	règles d'inférence.....	48
IV.4.4	représentations des fonctions d'appartenance.....	48
IV.4.5	Principe de la logique floue.....	48
IV.4.5.1	Opérateurs flous.....	49
IV.4.5.2	Structure interne d'un système d'inférence flou (SIF).....	49
IV.4.5.2.1	Fuzzification.....	49
IV.4.5.2.2	Défuzzification.....	50
IV.5	Systèmes neuro-flou.....	50
IV.5.1	Méthodes des combinaisons neuro-floues.....	51

## Sommaire

---

IV.5.2 Les types d'implémentation des réseaux neuro-flous.....	51
IV.5.2.1 Systèmes d'inférence neuro-flou adaptatifs (ANFIS).....	51
IV.6 Modèles simulés sous MATLAB/SIMULINK.....	53
IV.7 Simulations et résultats expérimentaux.....	54
IV.7.1 Cas théoriques.....	54
IV.7.2 Cas pratiques.....	55
IV.7.3 Interprétation des résultats.....	56
IV.7.3.1 Cas théorique.....	56
IV.7.3.2 Cas pratique.....	56
IV.8 Conclusion.....	56

# Introduction générale

## Introduction générale

Les premiers systèmes de téléopération sont apparus dans le nucléaire en vue de développer des outils permettant de manipuler des objets radioactifs dans des enceintes confinées. Les premiers télémanipulateurs étaient des systèmes mécaniques composés d'une partie maîtresse, tenue par l'opérateur et d'une partie esclave, exacte réplique de la partie maîtresse et munie d'un outil à son extrémité. Les étapes suivantes ont consisté à munir ces systèmes d'actionneurs électriques et de différents capteurs : Capteurs de position, de vitesse, de couple.

La téléopération a donc rejoint le champ de la robotique. On peut dire maintenant qu'un système de téléopération est composé de deux robots : un robot maître muni d'une poignée que peut tenir un opérateur et un robot esclave, muni d'un organe terminal où se situent un ou plusieurs outils selon la tâche à accomplir. L'originalité de la téléopération par rapport à la robotique classique tient à la place de l'homme dans la boucle, le fait que le système de téléopération sert à mesurer, transmettre et recopier les gestes de l'homme.

La téléopération s'impose par rapport à la robotique classique dans certaines situations typiques :

- Environnements dangereux (radioactifs, sous marins, espace) ;
- Environnements géographiquement distants ;
- Environnement inaccessibles (virtuel, fortement contraint, micro monde).

Depuis que la téléopération a quitté le monde du nucléaire, les distances ont augmentées entre le site maître et le site esclave. Dans le domaine spatial, des retards de l'ordre de quelques secondes peuvent être observés entre la terre et un satellite en orbite. Dans le domaine sous-marin, ce n'est pas la distance qui pose problème mais le milieu de transmission :

Par ailleurs, le fait que les ondes électromagnétiques ne peuvent pas traverser dans l'eau, ce qui nécessite l'utilisation d'un canal acoustique. On observe alors des temps de transmission de l'ordre de deux ou trois secondes entre la surface et le fond (à une profondeur de 2000m ; pour des profondeurs maximales, ce retard peut atteindre 10s). Dans les systèmes de téléopération sur terre, où les distances sont faibles, des retards de transmission sont présents si des traitements informatiques complexes sont réalisés avant l'envoi des consignes ; les retards sont alors de l'ordre de quelques dizaines, voir quelques centaines de millisecondes. Mais, si les retards de transmission dus à la distance sont incompressibles, les retards dus aux temps de calculs diminuent à mesure que les puissances de calcul augmentent.

En présence de retards, il ya un délai entre le moment où l'opérateur réalise une action (par exemple un déplacement du robot maître) et le moment où il observe l'effet de son action (par exemple, le moment où il voit le robot esclave bouger). La stratégie généralement adoptée est la stratégie dite « action attente » ('move and wait strategy') : l'opérateur réalise une succession de petits déplacements et attend d'observer le résultat de l'action précédente

pour passer à l'action suivante. Il en résulte une augmentation très importante du temps de réalisation d'une tâche. L'autre effet important, mis en évidence par [ferrell, 1965], est l'apparition d'instabilités dans les systèmes à retour d'effort. Or une bonne gestion de ces forces de contact est cruciale pour certaines tâches de téléopération.

Un aspect important de ces projets concerne le retour de force et la capacité, pour la personne qui commande les mouvements du robot, à ressentir les efforts qui lui sont appliqués. Le but est d'améliorer la qualité et l'immersion de la téléopération.

L'objectif de ce mémoire est précisément l'étude de nouveaux concepts permettant le contrôle des efforts par l'opérateur et la commande d'une interface haptique de la téléopération, ce système doit permettre une commande intuitive du robot télé opéré tout en reproduisant plus fidèlement possible les mouvements de robot maître, en particulier la qualité des sensations de l'opérateur.

Ce mémoire est organisé comme suit :

Dans le premier chapitre, nous présenterons les généralités et un bref historique sur les systèmes de téléopération et les différentes architectures favorisant la transparence et la stabilité du système. Ensuite, nous présenterons l'étude et la modalisation d'un système de téléopération haptique a un degré de liberté.

Le deuxième chapitre présentera notre carte d'acquisition utilisé qui assure la communication entre la partie commande et la partie opérative.

Le troisième chapitre sera consacré a l'étude de régulateur classique (PID) utilisé, qui nous a permet d'aboutir des bonnes résultats de simulations théoriques et pratiques.

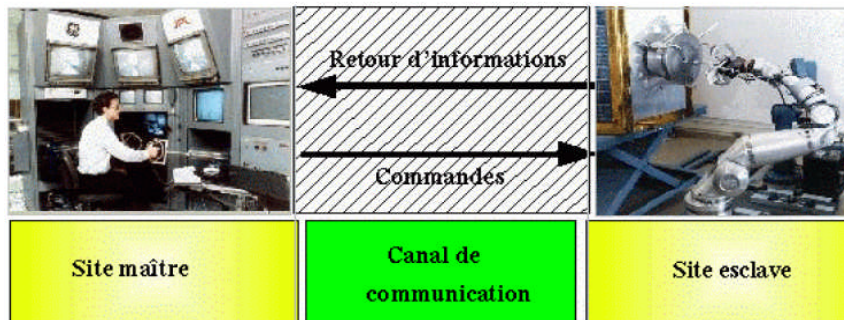
Le quatrième chapitre, sera consacré a l'étude d'un autre type de régulateur utilisé (neuro-flou), ce dernier nous a permet aussi d'aboutir des bonnes résultats de simulations théoriques et pratiques.

# Chapitre I

## Première partie : Généralités sur la téléopération

### I.1 Introduction

L'objectif d'un système de téléopération est de permettre à son utilisateur de réaliser une tâche à distance. Son fonctionnement est basé sur l'échange d'informations de position-position entre le maître, qui est la télécommande de l'utilisateur, et l'esclave qui est le robot réalisant effectivement la tâche. Ce principe est illustré à la (Figure 1) dont les différents éléments seront décrits dans la suite de ce chapitre.



**Figure1** : Illustration de l'architecture générale d'un système de Téléopération

Idéalement, le système de téléopération doit être complètement transparent, en reflétant fidèlement les interactions entre l'esclave et l'environnement sur l'opérateur.

Celui-ci doit pouvoir ressentir les efforts comme s'il réalisait directement la tâche à la place de l'esclave. En pratique, une transparence parfaite est impossible.

Elle sera limitée par la dynamique naturelle de l'interface du maître et celle de l'esclave (inerties, frottements, etc.), par le choix de la méthode de contrôle ainsi que par la présence de délais, de bruits et de la digitalisation. En fonction du nombre et du type de mesures disponibles différentes méthodes de contrôle peuvent être implémentées.

Cette étude se base sur un système à 1 degré de liberté (1 ddl) (linéaire). Malgré la présence en pratique de phénomènes non-linéaires (frottements secs, impacts,...), cette démarche permet de comprendre les principaux enjeux du contrôle.

Dans ce chapitre on donne une description mathématique d'un système de téléopération, avec un formalisme couramment rencontré dans la littérature.

Sur base de cette représentation, des critères de performances seront définis pour quantifier la transparence du système. Ensuite, les principales méthodes de contrôle seront décrites et comparées par rapport à ces critères. Pour finir, une étude sur la stabilité sera présentée pour mettre en évidence le compromis existant entre la recherche d'une meilleure transparence et la stabilité.

### I.2 bref historique

Le premier système de téléopération a été créé par Goertz en 1948, dans l'industrie nucléaire, avec la motivation de manipuler le carburant radioactif. La transmission entre le maître et l'esclave, été purement mécanique opérant a courte distance et il transmet à l'opérateur les forces

exercés sur la partie terminale via le système de couplage, qui sont insupportables par l'opérateur humain, lors de la manipulation des pièces lourdes.

L'apparition des contrôleurs bilatéraux à ouvrait le champ à de nombreuses améliorations des performances des systèmes de téléopération, de même a élargi leurs champ d'application : domaines comprenant des technologies de l'espace, les explorations marines, opérations militaires, utilisation matérielle et disposition de rebut de nucléaire/toxique, chirurgie, micro chirurgie, et réadaptations. Récemment, les applications des systèmes de téléopération ont été étendues à la formation, l'éducation, le divertissement, et les secteurs de la réalité virtuelle.

Néanmoins, les problèmes liés aux systèmes téléopération sont nombreux (retour sensoriel, compensation des mouvements physiologiques, le retard de transmission,..., etc.) à résoudre avant de donner à l'opérateur une impression qu'il manipule l'objet distant sans intermédiaire, l'inconvénient majeur de premiers TME (Télémanipulateur Maître Esclave) résidait dans la transmission des mouvements par un système mécanique composé de câbles et de poulies, ce qui limitait considérablement l'éloignement entre les deux sites. Afin de remédier à ce problème, R. Goertz présenta en 1954 une version modifiée du TME de 1945 avec des transmissions électriques et hydrauliques.

La structure d'un tel système ouvrait de nombreuses possibilités, en particulier dans les domaines militaires, de l'exploration sous-marine et spatiale.

Jusqu'à la fin des années 70, la structure mécanique des manipulateurs maître et esclave était exclusivement symétrique, rendant certaines manipulations peu confortables. L'avènement de l'informatique dans la commande des systèmes électromécaniques a permis de faire évoluer la structure du maître et esclave et ainsi d'abandonner cette symétrie. A partir des années 80, les TME sont ainsi devenus plus ergonomiques et mieux adaptés aux applications pour lesquelles ils étaient destinés. Les domaines d'applications se sont depuis diversifiés en vue de s'orienter aussi bien vers la micromanipulation pour la biologie ou la médecine, que vers la manipulation d'objets lourds et volumineux, pour les travaux publics. Cette évolution a suscité diverses problématiques de recherche, en particulier dans les domaines de l'automatique, de la mécanique et de l'électrotechnique.

Au début des années 90, les techniques de réalité virtuelle ont franchi une nouvelle étape en intégrant le retour d'efforts à la simulation.

### **I.3 La télé robotique**

La télé robotique est une forme de télé opération lorsque l'opérateur réalise des tâches à distance en utilisant un robot. Ce dernier peut fonctionner d'une façon autonome.

La télé robotique résulte en fait de la fusion des deux domaines originellement séparés qui sont la télé opération et la robotique. En effet, la robotique autonome n'étant pas encore tout à fait au point, le robot doit présentement être opéré à distance par un opérateur humain. On doit donc tenir compte des principes développés en télé opération. Cependant, comme le robot peut exécuter des tâches élémentaires de façon autonome, on parle de télé robotique plutôt que de télé

opération. La télé robotique trouve des applications partout où l'homme a des difficultés à travailler directement (milieu hostile, lointain, trop grand ou trop petit) et où les tâches sont suffisamment complexes ou imprévisibles pour rendre difficile une automatisation complète. La condition principale de développement de la télé robotique est sa capacité à concurrencer l'intervention directe d'un homme ou l'utilisation d'un système automatique très spécialisé. Dans le premier cas, l'atout de la télé robotique est tout d'abord le remplacement d'un travail humain pénible ou dangereux par un autre, plus sûr et confortable. Dans le second cas il faut montrer l'intérêt d'un matériel plus versatile que le système automatique dédié à l'application envisagée.

## I.4 quelques domaines d'application de la télé robotique

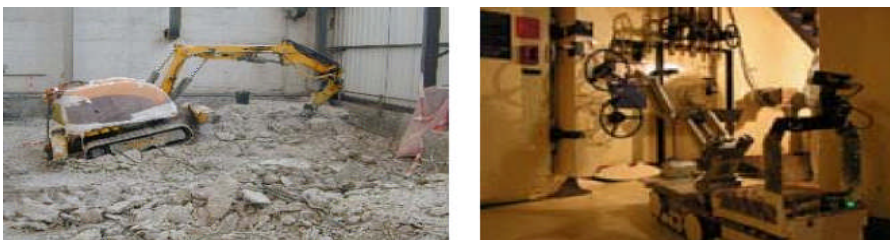
### I.4.1 Le domaine nucléaire

Le nucléaire a été le premier domaine à stimuler le développement de systèmes de télé robotique. L'industrie nucléaire s'intéresse généralement aux applications suivantes :

- la manipulation de produits radioactifs : elle se fait en cellule spécialisée, dans laquelle des télémanipulateurs mécaniques sont utilisés pour amener et retirer le produit. L'opérateur travaille en vision directe et il n'y a pas de problèmes d'éloignement. L'introduction de la télé robotique n'est pas encore envisagée dans ce type d'application.

- la maintenance des installations : ce type de tâche est essentiellement réalisé par des machines spécialisées. Ces dernières offrent l'avantage d'être parfaitement adaptées à la tâche. Par contre leur mise en œuvre est longue et elles ne sont pas adaptées aux changements de situation et ne peuvent être réutilisées dans d'autres tâches. La télé robotique est en train de trouver sa place suite aux recherches menées ces dernières années.

- le démantèlement d'installation et l'intervention suite à un incident : l'utilisation des systèmes télé robotique est une voie en cours de recherche concernant ces deux domaines. L'exploitation des nouvelles technologies comme la réalité virtuelle/augmentée est un atout favorable pour la poursuite de ces recherches. La figure 2 illustre l'utilisation d'un système de télé robotique pour le démantèlement d'installation et l'intervention dans le milieu nucléaire.



**Figure 2** , illustration de la télé robotique dans le domaine nucléaire : A gauche, le Démantèlement d'installation, à droite, l'intervention pour actionner des vannes.

### I.4.2 Le domaine spatial

Les principales applications de la télé robotique dans ce domaine sont l'exploration, la maintenance de satellites et de stations spatiales orbitales. L'éloignement du site de travail et l'important temps de transmission des informations, impliquent l'utilisation de robots ayant une

grande autonomie (basée sur une architecture complexe et intégrante de nombreux capteurs) sans pour autant exclure la possibilité de la télé opérer avec le robot.

### **I.4.3 Le domaine militaire**

La télé robotique intervient pour la désactivation de mines, l'observation de territoires ennemis ainsi que la télécommande d'engins tels que les chars, les avions et les hélicoptères. En raison de la complexité du milieu terrestre, l'apparition de robots autonomes au sein du système de combat n'est pas encore d'actualité. En revanche, la télé opération de plate-forme robotisée permet dès aujourd'hui d'assister l'homme dans certaines tâches pénibles ou en milieu hostile. Dans cette optique, l'industrie développe des technologies en matière de télé opération au combat afin de permettre aux combattants de poursuivre leur mission malgré les risques d'agression.

### **I.4.4 Le domaine sous-marin**

A l'exception des applications militaires, les applications civiles sont principalement liées à l'industrie de l'offshore, l'inspection, la construction et la maintenance de conduites, structures, câbles sous-marins voire aussi des investigations scientifiques (épaves, espèces marines, etc.). Des bras manipulateurs embarqués à bord de véhicules peuvent être autonomes ou même télé opérés. Ce domaine continue à être le principal domaine d'activité en nombre de systèmes de télé robotique.

## **I.5 Modélisation du système**

La modélisation d'un TME (Téléopérateur Maître Esclave) permet de mieux comprendre les interactions entre les différents éléments du système, réaliser des simulations pour mettre en place des stratégies de commande ou encore analyser la stabilité et les performances du TME. Pour cela, deux techniques sont classiquement utilisées dans la littérature :

- la modélisation sous forme de schémas blocs, où chaque élément est modélisé par une fonction de transfert ;
- la modélisation sous forme de quadripôles ou ports d'interaction, qui permet de mettre en évidence les échanges d'énergie entre ces éléments.

Sans être fondamentalement différentes, ces deux techniques sont complémentaires.

La modélisation sous forme de schémas blocs permet facilement de simuler et de synthétiser des lois de commande, alors que la représentation sous forme de quadripôles est particulièrement bien adaptée à l'analyse des structures de télé opération bilatérale.

### **I.5.1 Modélisation du comportement de l'opérateur humain**

La modélisation de l'être humain en interaction avec un système électromécanique est depuis très longtemps un domaine de recherche très actif. À notre connaissance, les premiers travaux sur le sujet datent de la fin des années 70, avec les modèles de McRuer ou de Kleinman qui sont utilisés dans le but de modéliser le comportement des pilotes de l'aérospatiale [McRuer67, Kleinman70]. Bien que les problématiques en télé robotique soient similaires, les chercheurs du domaine ne se sont intéressés à ces travaux que bien plus tard [Sheridan89, Brooks90]. Tous

s'accordent aujourd'hui pour dire que l'être humain est un système extrêmement difficile à modéliser, car il est à la fois un système multi-modèle, adaptatif, ayant la capacité d'apprendre et pouvant ajuster son comportement aux sollicitations extérieures.

Lors d'une manipulation, l'être humain a tendance à fonctionner selon trois modes :

- 1) le mode compensatoire qui fonctionne comme une boucle de rétroaction sur la trajectoire désirée et la trajectoire réelle ;
- 2) le mode poursuite qui permet d'anticiper la trajectoire dans le cas où l'utilisateur connaît l'environnement avec lequel il interagit ;
- 3) le mode de précognition qui s'appuie sur l'expérience et l'expertise de l'être humain.

Ces trois modes de fonctionnement sont généralement indissociables. L'être humain peut être amené à les combiner afin d'atteindre ses objectifs pour une tâche donnée.

### I.5.2 Modélisation de l'environnement

D'une manière générale en télé robotique la perception et l'interaction avec l'environnement posent des problèmes. En effet, les objets manipulés ont des propriétés mécaniques différentes les uns des autres, ce qui rend leur modélisation a priori difficile. Cependant, pour une application médicale, nous savons que l'environnement sera exclusivement constitué d'organes et de tissus vivants mous. Dans ce cas, l'environnement peut être décrit par une impédance mécanique équivalente mettant en relation les vitesses  $V_e(t)$ , les efforts d'interaction  $f_e(t)$  et les efforts internes  $f_i(t)$ . Les efforts internes représentent les efforts propres à l'environnement, qui ne résulte pas de l'interaction, comme par exemple les efforts dus aux mouvements physiologiques du patient.

### I.5.3 Modélisation des manipulateurs maître et esclave

En télémanipulation, les manipulateurs sont classés en deux familles, d'après les caractéristiques structurelles et le type d'actionneurs utilisés:

- 1) les manipulateurs de type impédance sont comparables à des sources d'efforts, car leurs actionneurs sont commandés en efforts ; lors d'une interaction avec un environnement, ils appliquent un effort en réponse à un déplacement de celui-ci. Généralement ils ont une faible inertie et sont réversibles. Dans la littérature, la plupart des interfaces haptiques sont de ce type.

- 2) les manipulateurs de type admittance sont comparables à des sources de position, ou de vitesse, car leurs actionneurs sont commandés en position, ou en vitesse ; lors d'une interaction avec un environnement, ils appliquent une position, ou une vitesse.

Ces manipulateurs sont généralement non réversibles et présentent une grande raideur.

### I.5.4 Modélisation du réseau de communication

Le réseau de communication permet de transmettre les signaux d'un site à l'autre, ce qui a pour effet d'introduire des temps de retard dans la transmission de données, voire dans certains cas la perte de données. Les problèmes associés aux retards sont un véritable défi pour la communauté automatique, comme en atteste le nombre de publications sur le sujet.

Dans ce mémoire, nous ne nous intéresserons pas à ce problème, puisque dans notre contexte applicatif la communication entre les deux sites se fait par une carte entièrement dédiée à cette utilisation et que la distance séparant le maître et l'esclave n'excède par quelques mètres.

## I.6 Les systèmes de télé opération avec retour d'effort

Le système de téléopération idéal est "transparent". Le robot esclave doit donc dupliquer exactement les déplacements du robot maître. Le robot maître doit restituer le plus fidèlement possible à l'opérateur les efforts exercés par l'esclave sur son environnement. Le système du pantographe réalise ce retour d'effort (aux imperfections mécaniques près) naturellement. Dans le cas où les robots sont équipés d'actionneurs électriques, plusieurs solutions techniques sont envisageables pour réaliser ce retour d'effort selon que l'on dispose ou non de capteurs d'effort. Pour illustrer chaque type de couplage de télé opération, nous avons pris un "modèle" simplifié de robot (maître et esclave). Les robots ont un seul degré de liberté, ce sont de simples impédances mécaniques (masse, amortissement, raideur) et les actionneurs sont des générateurs de force parfaits (ce qui revient à considérer que la structure mécanique est réversible).

### I.6.1 Asservissement bilatéral

#### Principe

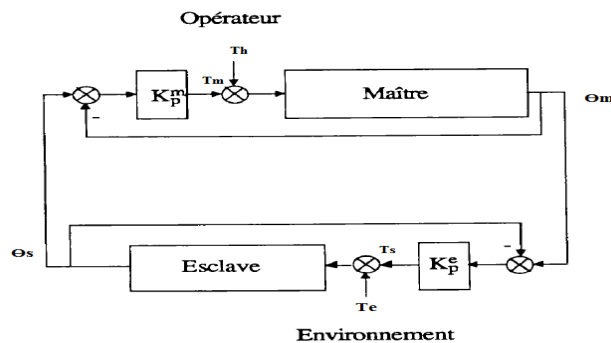


Figure 3 : asservissement bilatéral position- position

L'asservissement bilatéral présente une structure symétrique : le robot maître est asservi sur la position du robot esclave et vice-versa. Sur la figure 3 on a représenté schématiquement l'asservissement bilatéral pour un système à un seul degré de liberté. On note  $T_e$  et  $T_m$  les forces dues aux actionneurs et  $T_h$  et  $T_e$  les forces exercées par l'opérateur sur le robot maître et par l'environnement sur le robot esclave. Enfin, on note  $\theta_m$  et  $\theta_s$  les positions des robots.

## I.7 Les systèmes de télé opération sans retour d'effort

Les systèmes de télé opération avec retour d'effort présentent tous la même propriété : c'est l'opérateur, dans la boucle, qui gère les forces de contact. Si ces solutions sont idéales en théorie, en pratique le retour d'effort n'a plus aucun intérêt lorsque des retards de transmission sont présents dans la boucle.

Dans les systèmes de télé opération sans retour d'efforts la gestion des forces de contact est faite localement. Les problèmes d'instabilité ne se posent donc plus, même si des retards de transmission sont présents. L'opérateur n'intervient plus dans la boucle de contrôle d'effort, il envoie simplement des consignes qui sont appliquées par le robot esclave en fonctions de ses interactions avec l'environnement. (Figure 4).

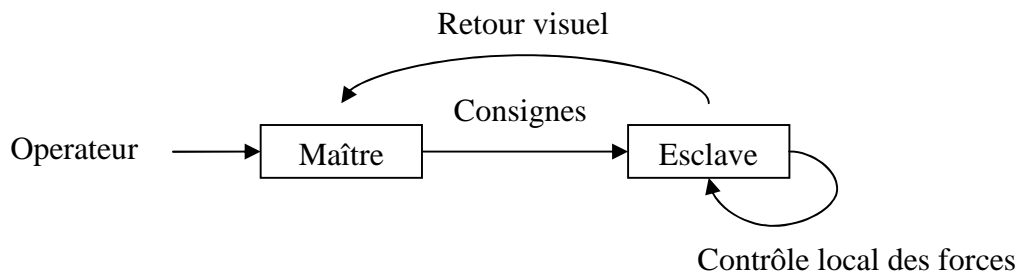


Figure 4 : Téléopération sans retour d'efforts

### I.8 Robot esclave hybride

Pour les axes commandés en position (car sans obstacle) la force exercée par l'opérateur est transformée en une consigne de vitesse. Lorsqu'un axe est commandé en force (car un contact a été détecté dans cette direction), la force commandée est proportionnelle à la force exercée par l'opérateur. On se ramène donc exactement au cas précédent.

### I.9 Pilotage des déplacements

Dans les deux cas (robot esclave hybride), la force exercée par l'opérateur est transformée en une vitesse. Il y a donc intégration entre une consigne constante (la force exercée par l'opérateur) et la position du robot esclave (puisque la vitesse du robot esclave est constante). Si l'opérateur veut piloter un déplacement (par exemple 3cm dans une certaine direction), il doit se fier à une représentation visuelle du robot esclave. Le pilotage des déplacements se fait donc grâce à une boucle avec retour d'image. Lorsqu'un retard de transmission est présent, l'opérateur ne voit pas instantanément l'effet de la force qu'il applique. Il est donc nécessaire de représenter, en plus d'une image du robot esclave réel, une image d'un robot esclave simulé piloté sans retard de transmission. Pour commander un déplacement, l'opérateur se fie à l'image simulée.

### I.10 Performance et stabilité en télé opération

Afin d'élaborer une loi de commande adéquate, il est nécessaire de définir les objectifs que l'on désire atteindre avec le système. L'objectif de la télémanipulation bilatérale est de fournir à l'utilisateur une transparence parfaite tout en garantissant la stabilité du système, quelles que soient les conditions d'utilisation. Ceci, traduit la qualité de restitution des efforts et des mouvements entre le maître et l'esclave.

#### I.10.1 Stabilité

Pour analyser la stabilité des systèmes de télé opération, deux approches peuvent être envisagées.

La première consiste à considérer l'utilisateur et l'environnement comme des systèmes passifs. Ensuite, la stabilité du TME est déterminée en cherchant les conditions de passivité du TME. Il s'agit d'une approche classique pour l'analyse de stabilité des systèmes non linéaires.

En télé robotique, l'analyse de passivité permet de résoudre les problèmes d'instabilité pour une large gamme d'applications.

La seconde approche, beaucoup plus classique en automatique, consiste à utiliser un modèle LTI (linéaire à temps invariant) du système complet, en y intégrant un modèle nominal de l'utilisateur et de l'environnement.

### I.10.2 Transparence

La notion de transparence est associée aux performances d'un système de télé opération. La transparence parfaite est atteinte lorsque l'utilisateur a l'impression de manipuler directement les objets de l'environnement, sans ressentir ni les effets dynamiques du manipulateur maître, ni ceux du manipulateur esclave. Dans la littérature, plusieurs définitions de la transparence ont été proposées. Parmi les plus cités, la correspondance d'impédance entre celle ressentie par l'utilisateur et celle de l'environnement permet d'exprimer le degré de transparence en termes d'impédance ressentie par l'utilisateur. Si le manipulateur esclave est en contact avec l'environnement et si l'interface maître est maintenue par l'utilisateur tout au long de la manipulation, il est alors possible d'établir une relation mathématique entre les mouvements et les efforts de chaque côté (cotés maître et esclave).

### I.10.3 Passivité

D'un point de vue pratique, la notion de passivité est bien souvent suffisante et offre un cadre théorique simple à mettre en œuvre, ce qui la rend très populaire auprès de la communauté de la télé robotique. La notion de passivité est basée sur l'observation de l'énergie dissipée par le système.

La définition mathématique de la passivité peut s'énoncer de la manière suivante :

**Définition (Passivité) :** Soit le système dynamique décrit par les équations d'état :

$$\begin{cases} \dot{x} = f(x; u; t) \\ y = g(x; u; t) \end{cases} \dots\dots\dots (I.1)$$

Où  $x$  est le vecteur d'état,  $u$  le vecteur d'entrée,  $y$  le vecteur de sortie,  $f$  et  $g$  sont deux fonctions quelconques. Alors, ce système est passif si et seulement si il existe une fonction  $V(x)$  semi-définie positive continûment différentiable telle que :

$$\int_0^t u^T(\tau)y(\tau)d\tau \geq V(x(t)) - V(x(0)), \forall t \geq 0 \dots\dots\dots (I.2)$$

$V$  est aussi appelée fonction de stockage. De manière équivalente, comme  $V$  est différentiable, l'expression (I.2) peut se réécrire sous forme différentielle :

$$u^T(t)y(t) \geq \dot{V}(x(t)) \dots\dots\dots (I.3)$$

La représentation sous forme de quadripôle est particulièrement bien adaptée à l'analyse de la passivité. En effet, avec la représentation sous forme de matrice hybride l'expression (I.2) peut se réécrire:

$$\int_0^t v_h(\tau)f_h(\tau) - v_e(\tau)f_e(\tau)d\tau \geq 0, \forall t \geq 0 \dots\dots\dots(I.4)$$

Une des propriétés remarquables de la passivité, très intéressante pour la télé opération, est que l'interconnexion de deux systèmes passifs donne un système asymptotiquement stable. En considérant que l'opérateur humain et l'environnement sont passifs, il est alors possible de montrer que le système complet est stable si le TME est passif. Par contre, les conditions de stabilité obtenues sont conservatives, ce qui détériore les performances du système.

### I.11 Stratégies de commande bilatérale

Aujourd'hui, l'utilisation des TME s'est diversifiée. L'élaboration d'un cahier des charges précis sur le degré de transparence que l'on souhaite atteindre devient un véritable enjeu. Dans le domaine médical, il est primordial que le système soit stable, quelle que soit son utilisation. Cependant, on constate que certaines manipulations, comme l'insertion d'aiguille, nécessitent une attention toute particulière en ce qui concerne la qualité de restitution des efforts.

Au fil des années, des problématiques classiques liées à l'utilisation des TME sont apparus. On mentionnera en particulier les problèmes de stabilité dus au contact avec des environnements très rigides, ou encore ceux liés aux retards dans le réseau de communication.

#### I.11.1 Structure à deux canaux position-position

À notre connaissance, la structure position-position est la première commande bilatérale implémentée sur un système de télé opération. Très largement reprise dans la littérature, cette structure a fait l'objet de nombreuses études. La version originale de la commande position-position, utilise uniquement les mesures de position des manipulateurs maître et esclave. Il n'est donc pas nécessaire d'intégrer un capteur d'efforts, ce qui est particulièrement bien adapté aux systèmes qui nécessitent un faible encombrement.

Suite aux structures à deux canaux position-position, il existe aussi la structure à deux canaux force-position, cette structure est contrairement à la structure position-position qui ne requiert pas de capteur d'efforts, la structure force-position nécessite au moins la mesure de l'effort d'interaction entre l'esclave et l'environnement, pour que celui-ci soit restitué à l'utilisateur, par l'intermédiaire du manipulateur maître. Il existe de nombreuses variantes à cette structure, selon que les efforts sont mesurés côté esclave seulement ou des deux côtés.

L'utilisation de la mesure d'effort côté maître permet cependant d'accroître les performances du système, et d'assurer, si nécessaire, une meilleure réversibilité du manipulateur maître.

## Deuxième partie: dispositif de contrôle et de commande

### I.12 Design du banc de test

La Figure 5 représente les composants du système de télé opération maître/esclave utilisé comme banc de test. Il est constitué de deux interfaces hardware (moteurs à courant continu) à 1 degré de liberté identiques représentant le maître et l'esclave, autour d'une carte de contrôle (Arduino-Méga 2560). Celle-ci permet d'assurer les échanges d'information (capteurs et commandes) entre les deux moteurs en fonction du type de contrôle haptique implémenté, et un PC qui nous permet d'introduire notre programme dans la carte de contrôle et d'interagir avec l'expérience.



Figure 5: Diagramme de connexion du dispositif de contrôle et de commande

### I.13 Design électromécanique

Le degré de liberté des interfaces maître et esclave a été choisi en rotation dans l'optique de pouvoir adapter ce design aux futurs joints de l'exosquelette, correspondant aux joints de rotation du bras. Une représentation de la réalisation est L'interface possède un actionneur au maître, pour générer le retour d'effort, et à l'esclave, pour commander sa position et interagir avec l'environnement. Dans le but de réaliser un premier système de "référence", la technologie d'actionnement correspond à celle la plus couramment rencontrée dans les interfaces haptiques.

Exemple: comme sur le Phantom de la société Sensable (Guy et al. 1996) et le Delta d'immersion (Grange et al. 2001). Elle est basée sur l'utilisation d'un moteur à courant continu (DC, type Maxon A-Max 32) associé à un cabestan.

Ce dernier est un réducteur à câble, sans jeu et présentant un très faible niveau de friction, assurant ainsi une bonne transparence naturelle du système. Le coefficient de réduction est défini par le rapport entre le diamètre du cabestan et celui de l'enrouleur attaché à l'axe moteur. Dans ce design il correspond à une valeur de 10. Cela permet de générer un couple continu de 0.5 Nm.

**I.14 Capteurs**

Pour pouvoir implémenter les lois de contrôle, l'information doit être mesurée : La position est mesurée par un encodeur. Cette technologie a été sélectionnée pour sa facilité d'intégration avec le moteur à courant continu (DC) ainsi qu'avec la carte d'acquisition. Il est à noter que l'encodeur étant placé avant le réducteur, la résolution est amplifiée par le rapport de réduction.

**I.15 Carte de Contrôle**

La carte de contrôle utilisée est la carte Arduino-Méga 2560 et programmée par l'intermédiaire MATLAB Simulink. Elle possède deux rôles principaux.

D'une part elle enregistre les valeurs des capteurs de position, à travers une interface encodeur et un convertisseur analogique/numérique. D'autre part, elle implémente un contrôleur haptique qui calcule la commande pour chaque côté du système. L'implémentation de ce contrôleur est similaire pour le maître et l'esclave.

**I.16 Etude d'un système de télé opération**

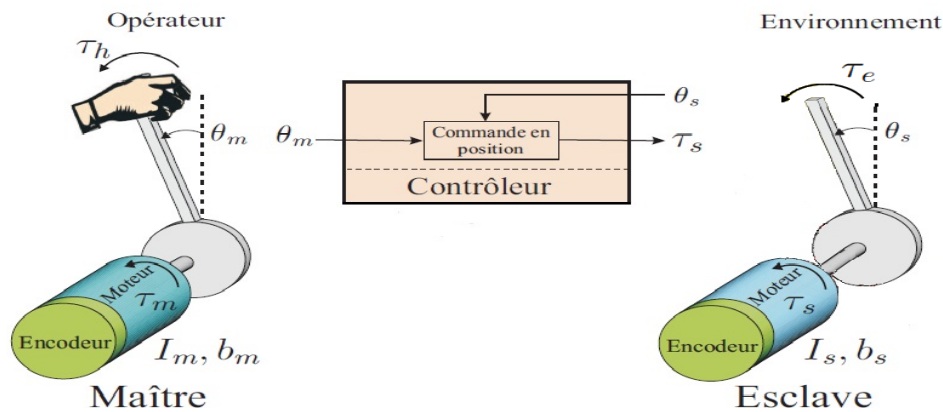


Figure 6 : système de téléopération à 1 degré de liberté (contrôle position- position)

**I.16.1 Représentation d'un système de télé opération**

La Figure 6 représente un système classique de télé opération haptique à 1 degré de liberté. Le comportement mécanique du maître et de l'esclave peut être approché par celui d'un système masse/dashpot, représenté par les inerties  $I_m$  et  $I_s$  des masses en mouvement, et les frottements visqueux  $b_m$  et  $b_s$  (les indices m et s sont relatif aux mots anglais master slave). En considérant les différents couples appliqués à l'arbre de chaque système, et en négligeant les effets de gravité de la barre, les comportements dynamiques du maître et de l'esclave sont définis par,

$$\begin{cases} I_m \ddot{\theta}_m + b_m \dot{\theta}_m = T_h + T_m \\ I_s \ddot{\theta}_s + b_s \dot{\theta}_s = T_s + T_e \end{cases} \dots\dots\dots(I.5)$$

Avec  $\theta_m$  et  $\theta_s$  la position du maître et de l’esclave,  $T_h$  le couple produit par l’opérateur sur le maître,  $T_e$  le couple de réaction de l’environnement sur l’esclave,  $T_m$  et  $T_s$  les couples délivrés par les actionneurs, respectivement pour créer le retour de force sur l’utilisateur pour commander le mouvement de l’esclave.

Nous pouvons exprimer l’équation (I.5) en coordonnées de Laplace en introduisant la notion d’impédance du maître et de l’esclave  $Z_m(s)$  et  $Z_s(s)$ ,

$$\begin{cases} Z_m \theta_m = T_h + T_m \\ Z_s \theta_s = T_s + T_e \end{cases} \dots\dots\dots (I.6)$$

Avec  $Z_m(s) = (I_m s^2 + b_m s)$  et  $Z_s(s) = (I_s s^2 + b_s s)$

Pour l’analyse des systèmes de télé opération et étudier leurs transparence, Hannaford a proposé de les représenter sur base d’un modèle 2-ports, dérivé des réseaux électriques. Un système de télé opération peut être considéré comme une boîte noire qui définit une relation entre les positions ( $\theta_m, \theta_s$ ) et les forces d’interactions ( $T_h, T_e$ ) et cette relation est exprimée par :

$$\begin{bmatrix} T_h \\ \theta_s \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{bmatrix} \begin{bmatrix} \theta_m \\ T_e \end{bmatrix} \dots\dots\dots (I.7)$$

Cette matrice est la matrice d’immittance hybride H de dimension 2x2. Elle caractérise complètement le système de télé opération (dynamique du maître et de l’esclave, contrôleurs, communications). Les composantes  $h_{ij}(s)$  définissent les relations entre les positions et force au maître et à l’esclave, elles seront utilisées pour étudier les performances et la stabilité du système. Il existe d’autres matrices de relation en fonction du choix des composantes des vecteurs mais c’est ce mode de représentation qui est le plus répandu dans la littérature.

En théorie, pour une analogie parfaite avec le formalisme des réseaux électriques, il faudrait exprimer la matrice H en fonction des vitesses du maître et de l’esclave. Cependant, nous avons choisi ici d’utiliser les positions étant donné que dans la grande majorité des cas, c’est la position et non la vitesse qui est mesurée (encodeurs).

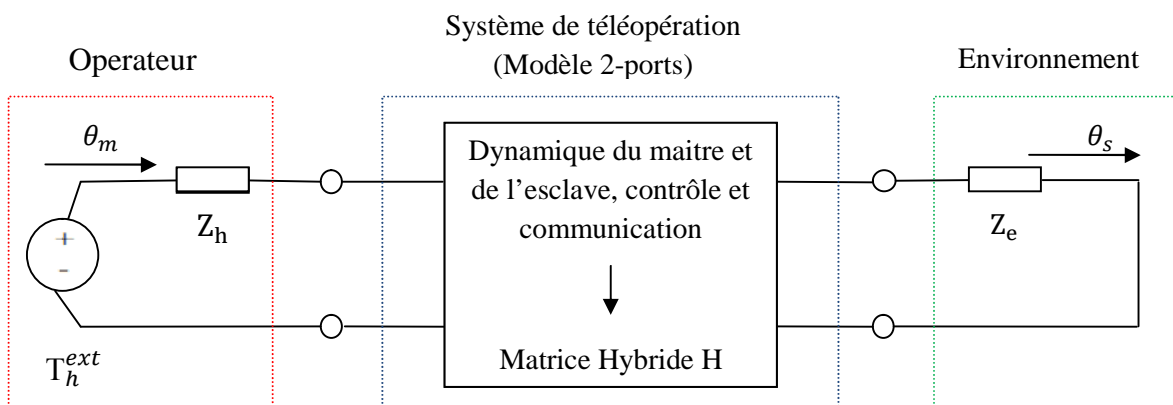


Figure 7: représentation d'un système de téléopération par modèle 2-ports

L’environnement est modélisé par l’impédance  $Z_e(s)$  qui représente dans ce cas une raideur  $k_e$ . Cela définit la relation entre la position et le couple de l’esclave,

$$T_e = -k_e \theta_s = -Z_e \theta_s \dots\dots\dots(I.8)$$

L'opérateur est modélisé par l'impédance ;

$$Z_h = I_h s^2 + b_h s + k_h \dots\dots\dots(I.9)$$

Avec  $I_h$  et  $b_h$  l'inertie et l'amortissement de l'utilisateur et  $k_h$  la raideur de l'interaction homme/machine. Ces termes seront en fonction de la manière dont l'utilisateur est fixé à l'interface haptique (préhension souple, ferme,...) la force d'interaction avec l'interface haptique  $T_h$  correspond à la force active  $T_h^{ext}$ , générée par l'activité musculaire, diminuée des effets dynamiques du bras ou de la main,

$$T_h = T_h^{ext} - Z_h \theta_m \dots\dots\dots(I.10)$$

Cette relation pourra être utilisée pour tenir compte de l'influence de l'opérateur dans l'étude de la stabilité.

**I.16.2 Critères de performances**

L'idéal pour un système de télé opération est d'atteindre la transparence, c'est à dire que l'opérateur ait l'impression de réaliser directement les opérations à la place de l'esclave. La transparence idéale peut être définie par la capacité du système à assurer en permanence le tracking en position entre le maître et l'esclave,

$$\forall t \begin{cases} \theta_s(t) = \theta_m(t) \\ T_h(t) = -T_e(t) \end{cases} \dots\dots\dots(I.11)$$

Donc la transparente parfaite est atteinte lorsque la matrice hybride H présente la forme suivante :

$$\begin{bmatrix} T_h \\ \theta_s \end{bmatrix} = H_{ideale} \begin{bmatrix} \theta_m \\ T_e \end{bmatrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} \begin{bmatrix} \theta_m \\ T_e \end{bmatrix} \dots\dots\dots(I.12)$$

- $h_{11} = \frac{T_h}{\theta_m} = Z_{min}$  ; (lorsque  $T_e=0$ ) (idéalement  $\rightarrow 0$ ) correspond à l'impédance ressentie par l'opérateur lorsque l'esclave est en mouvement libre (aucune interaction avec l'environnement). Ce terme doit donc être minimisé.
- $h_{21} = \frac{\theta_s}{\theta_m} = X_T$  ; (lorsque  $T_e=0$ ) (idéalement  $\rightarrow 1$ ) est la fonction de transfert du tracking en position, en mouvement libre, qui doit tendre vers 1 sur la plage de fréquence la plus large possible.
- $h_{12} = \frac{T_h}{T_e}$  ; (lorsque  $\theta_m=0$ ) représente le tracking en force, lors d'un contact, lorsque la position du maître est maintenue constante.
- $h_{22} = \frac{\theta_s}{T_e}$  ; (lorsque  $\theta_m=0$ ) est appelée l'admittance de contact à l'esclave. Pour une position du maître fixe, elle représente le mouvement de l'esclave en présence d'une force de

contact. Elle aura pour effet de diminuer la raideur présentée à l'utilisateur par rapport à celle de l'environnement.

Les deux derniers termes sont exprimés dans la situation où la position du maître est maintenue constante. Cela n'est cependant pas facilement réalisable en pratique.

Il a donc été proposé par (Aliaga, 2004) de définir un set de quatre critères de performance facilement identifiables par l'expérience et qui représentent également de manière complète le système de téléopération. Deux types d'expériences ont été identifiées comme facilement réalisables : le mouvement libre, lorsque  $T_e = 0$  et le contact d'un environnement infiniment rigide ( $k_e \rightarrow \infty$ ) qui impose  $\theta_s = 0$ . Les deux premiers termes de la matrice hybride présentés ci-dessus peuvent être utilisés pour définir le tracking en position, en mouvement libre (idéalement  $\rightarrow 1$ ),

$$X_T = \frac{\theta_s}{\theta_m} = h_{21} \quad (\text{lorsque } T_e = 0) \quad \dots\dots\dots(I.13)$$

et l'impédance minimum ressentie en mouvement libre (idéalement  $\rightarrow 0$ ),

$$Z_{min} = \frac{T_h}{\theta_m} = h_{11} \quad (\text{lorsque } T_e = 0) \quad \dots\dots\dots(I.14)$$

Pour le contact rigide, en considérant  $\theta_s = 0$  dans l'éq. (I.7), nous pouvons définir le tracking en force (idéalement  $\rightarrow 1$ ),

$$F_T = - \frac{T_e}{T_h} = \frac{h_{21}}{h_{12}h_{21} - h_{11}h_{22}} \quad (\text{lorsque } \theta_s = 0) \quad \dots\dots\dots(I.15)$$

et l'impédance maximum transmissible à travers le système de télé opération (idéalement  $\rightarrow \infty$ ),

$$Z_{max} = - \frac{T_h}{\theta_m} = \frac{h_{11}h_{22} - h_{12}h_{21}}{h_{22}} \quad (\text{lorsque } \theta_s = 0) \quad \dots\dots\dots(I.16)$$

Comme présenté dans (Aliaga et al., 2004), ces quatre critères permettent de retrouver les quatre termes de la matrice hybride H et constituent donc une description complète du système de télé opération. Ils seront utilisés dans la section suivante pour étudier les principaux schémas de contrôle.

Pour une raideur d'environnement  $Z_e(s)$  donnée, suivant les équations (I.7) et (I.8), l'impédance transmise à l'opérateur  $Z_{th}(s)$  est donnée par,

$$Z_{th} = - \frac{T_h}{\theta_m} = \frac{h_{11} + (h_{11}h_{22} - h_{12}h_{21}) Z_e}{1 + h_{22} Z_e} \quad \dots\dots\dots(I.17)$$

Le rapport  $Z_{th} / Z_e$  est utilisé par certains auteurs pour définir la transparence du système (idéalement  $\rightarrow 1$ ), (Lawrence, 1993). Les termes d'impédance minimum  $Z_{min}$  et d'impédance maximum  $Z_{max}$  sont les valeurs extrêmes de  $Z_{th}$  respectivement avec  $Z_e = 0$  et  $Z_e \rightarrow \infty$ .

Un autre critère régulièrement rencontré est la dynamic range, également appelé le Z-width, introduit par (Colgate and Brown, 1994), et défini par  $Z_{max} / Z_{min}$ .

Il représente la gamme d'impédances que le système de télé opération haptique peut présenter à l'utilisateur. Plus il est grand, plus le système sera capable de présenter des sensations différentes et ainsi marquer clairement la transition entre le mouvement libre et le contact.

**I.16.3 Contrôle des systèmes de téléopération bilatérale**

**I.16.3.1 Méthodes de contrôle position-position**

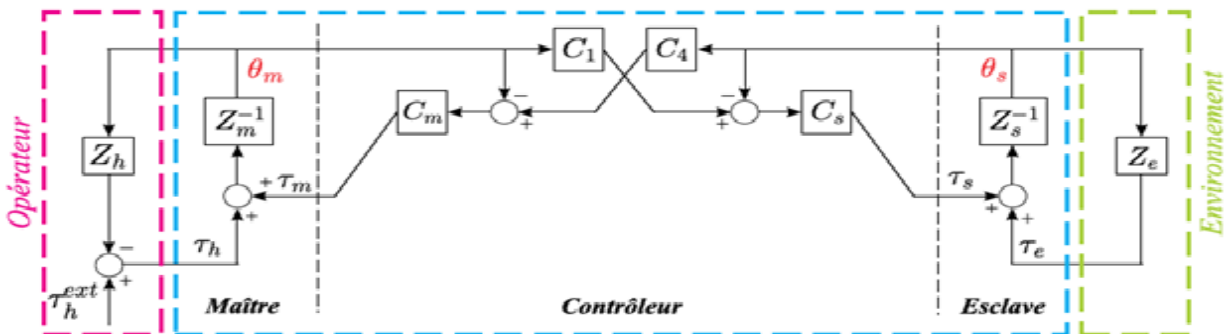


Figure 8 : schéma de contrôle position-position

Nous allons présenter dans cette section les principales méthodes de contrôle mises en œuvre dans les systèmes de téléopération haptique. Elles sont habituellement classées par rapport au nombre et au type d'informations échangées entre le maître et l'esclave. Les stratégies les plus simples implémentent un échange de deux signaux (un dans chaque sens), basés sur les mesures de position et de force. Les stratégies plus avancées utilisent un nombre supérieur de canaux de communication pour permettre d'atteindre de meilleures performances.

La méthode de contrôle position-position se base sur l'échange des mesures de positions entre le maître et l'esclave, donc le retour d'effort est représenté par l'erreur de position entre les deux entités, historiquement c'est la première méthode de contrôle implémentée sur les systèmes de téléopération.

La représentation du schéma de contrôle de la méthode illustré à la figure 8, le renvoi du signal de position de l'esclave vers le maître a travers le canal de communication  $C_4$ , signifie que chaque coté contrôlé en position sur base de consignes échangées, dont les commandes des deux moteurs sont :

$$\begin{cases} T_m = C_m(C_4\theta_s - \theta_m) \\ T_s = C_s(C_1\theta_m - \theta_s) \end{cases} \dots\dots\dots (I.18)$$

Avec  $C_m = (k_d^m s + k_p^m)$  et  $C_s = (k_d^s s + k_p^s)$  les contrôleurs de position proportionnel/dérivé du maître et de l'esclave. Sur base des équations dynamiques (I.6), la représentation hybride du système est alors fournie par,

$$\begin{bmatrix} T_h \\ \theta_s \end{bmatrix} = \begin{bmatrix} \frac{(Z_m + C_m)(Z_s + C_s) - C_1 C_4 C_m C_s}{Z_s + C_s} & \frac{C_4 C_m}{Z_s + C_s} \\ \frac{C_1 C_s}{Z_s + C_s} & \frac{1}{Z_s + C_s} \end{bmatrix} \begin{bmatrix} \theta_m \\ T_e \end{bmatrix} \dots\dots\dots (I.19)$$

Pour garder une cohérence entre l’envoi et la réception des signaux de position par rapport au maître, nous considérons que le produit des gains d’amplification est unitaire ( $C_1 C_4 = 1$ ). Dans ce cas, les critères de performances peuvent être calculés,

$$\begin{cases} X_T = \frac{\theta_s}{\theta_m} = \frac{C_1 C_s}{Z_s + C_s} \\ F_T = \frac{C_1 C_s}{Z_m + C_m} \\ Z_{min} = \frac{T_h}{\theta_m} = Z_m + \frac{C_m Z_s}{Z_s + C_s} \\ Z_{max} = Z_m + C_m \end{cases} \dots\dots\dots (I.20)$$

Le point faible de cette stratégie est la sensation ressentie par l’opérateur en mouvement libre, représentée par  $Z_{min}$ . A basse fréquence, avec un gain  $k_p^s$  tel que  $C_s \gg Z_s$ , nous pouvons écrire,

$$Z_{min} \approx Z_m + Z_s \frac{C_m}{C_s} \dots\dots\dots (I.21)$$

Si  $C_m = C_s$ , l’utilisateur ressentira donc en mouvement libre l’impédance du maître additionnée à celle de l’esclave. Augmenter le gain  $k_p^s$  dans  $C_s$  pour diminuer la contribution de  $Z_s$  se fera au prix d’une perte du tracking en force  $F_T$  dont le gain à basse fréquence est déterminé par le rapport des gains des contrôleurs (avec  $C_m \gg Z_m$ ). Cela aura pour effet concret que la force de l’esclave sur l’environnement sera supérieure à celle ressentie par l’utilisateur.

L’expression de l’impédance maximum  $Z_{max}$  ne dépend pas de  $C_s$  car en contact rigide, la consigne renvoyée vers le maître est constante et indépendante des gains de  $C_s$ . Le contrôleur de l’esclave aura cependant bien une influence sur la qualité de transmission de la raideur dans le cas  $k_e < 1$

En résumé, l’avantage principal de la méthode Position-Position est sa facilité d’implémentation au point de vue hardware.

**I.17 Conclusion**

Dans ce chapitre nous avons présenté les concepts de base pour l’étude et la commande bilatérale d’un système de télémanipulation avec retour d’efforts. La première partie de ce chapitre, consacrée à l’historique, a permis de situer la téléopération dans un contexte plus général, Puis nous avons présenté la modélisation d’un TME.

L’objectif d’une télémanipulation avec retour d’efforts est de fournir à l’utilisateur un système lui permettant de réaliser une tâche à distance tout en ayant l’impression de la réaliser directement. Pour cela nous avons introduit la notion de transparence liée aux performances générales du système. Bien que la transparence soit un élément essentiel dans ce type de système, il est nécessaire de garantir aussi la stabilité quelle que soit l’utilisation ou l’interaction.

Pour cela, nous avons introduit la notion de passivité, qui se réfère à l'énergie dissipée par le système.

Enfin, nous avons présenté dans la deuxième partie de ce chapitre la structure de base pour la téléopération bilatérale, à savoir la structure position-position.

Pour laquelle, nous avons détaillé les lois de commandes, la représentation sous forme de matrice hybride, ainsi que l'impédance perçue par l'utilisateur.

# Chapitre II

## II.1 Introduction

Pour la réalisation de notre système de télé opération, nous avons utilisé une carte d'acquisition de type Arduino afin d'assurer la communication entre les deux parties à savoir ; la partie commande (ordinateur) et les deux sites (maître et esclave) afin de pouvoir commander et piloter le système, et cette partie est.

## II.2 La carte d'acquisition :

### II.2.1 Définition de la carte Arduino

C'est une plate-forme open-source d'électronique programmée qui est basée sur une simple carte à microcontrôleur (de la famille AVR), et un logiciel, véritable environnement du développement intégré, pour écrire, compiler et transférer le programme vers la carte à microcontrôleur.

Arduino peut être utilisé pour développer des objets interactifs, pouvant recevoir des entrées d'une grande variété d'interrupteurs ou de capteurs, et pouvant contrôler une grande variété de lumières, moteurs ou toutes autres sorties matérielles. Les projets Arduino peuvent être autonomes, ou bien ils peuvent communiquer avec des logiciels tournant sur l'ordinateur (tels que Flash, Processing ou MaxMSP, Matlab).

Le langage de programmation Arduino est une implémentation de Writing, une plate forme de développement similaire, qui est basée sur l'environnement multimédia de programmation Processing.

Ces cartes sont basées sur une interface entrée/sortie simple et sur un environnement de développement proche du langage C.

Elle contient tout le nécessaire pour piloter le microcontrôleur, il suffit simplement de la connecter à un ordinateur avec un câble USB pour l'utiliser simplement (fournit avec la carte).

### II.2.2 Historique

D'après Wired Magazine : Le projet Arduino est né en hiver 2005. Massimo Banzi enseigne dans une école de Design à Ivrea en Italie, et souvent ses étudiants se plaignent de ne pas avoir accès à des solutions bas prix pour accomplir leurs projets de robotique. Banzi en discute avec David Cuartielles, un ingénieur Espagnol spécialisé sur les microcontrôleurs... Ils décident de créer leur propre carte en embarquant dans leur histoire un des étudiants de Banzi, David Mellis qui sera chargé de créer le langage de programmation allant avec la carte. En deux jours David écrira le code! Trois jours de plus et la carte était créée...Ils décidèrent de l'appeler Arduino (un bar fréquenté par les élèves à proximité de l'école)...

Il devient un hit tout de suite auprès des étudiants. Tout le monde arrive à en faire quelque chose très rapidement sans même avoir de connaissances particulière ni en électronique ni en informatique: réponse à des capteurs, faire clignoter des leds, contrôler des moteurs... Ils publient les schémas, investissent 3000 euros pour créer les premiers lots de cartes:

Les 50 premières partent directement à des élèves de l'école. En 2006 5 000 cartes vendues...En 2007 plus de 30 000, En 2011 plus de 120 000, sans compter les clones.

### II.2.3 Caractéristique de la carte Arduino

Parmi les caractéristiques les plus importantes de la carte Arduino, nous avons retenu les suivantes :

- **pas cher** : les cartes Arduino sont relativement peu coûteuses comparativement aux autres plateformes.
- **multiplateforme** : le logiciel Arduino, écrit en Java, tourne sous les systèmes d'exploitation Windows, Macintosh et Linux. La plupart des systèmes à microcontrôleurs sont limités à Windows.
- **un environnement de programmation clair et simple** : l'environnement de programmation Arduino est facile à utiliser pour les débutants, tout en étant assez flexible pour que les utilisateurs avancés puissent en tirer profit également.
- **logiciel Open Source et extensible** : le logiciel Arduino et le langage Arduino sont publiés sous licence open source, disponible pour être complété par des programmeurs expérimentés.
- **matériel Open source et extensible** : les cartes Arduino sont basées sur les microcontrôleurs Atmel ATMEGA2560, ATMEGA168, ATMEGA 328, etc. Les schémas des modules sont publiés sous une licence Creative Commons, et les concepteurs de circuits expérimentés peuvent réaliser leur propre version des cartes Arduino, en les complétant et en les améliorant.
- La carte Arduino est capable de stocker un programme et de le faire exécuter.
- La carte reçoit des informations analogiques ou numériques sur ses entrées et génère des informations analogiques ou numériques.
- Le microcontrôleur traitera ces informations et les transmettra vers les sorties numériques.
- Récupère les données des capteurs pour les transmettre à l'interface de commande (pc) et traduire les instructions pour faire fonctionner les actionneurs.

### II.3 Architecture de la carte

La figure suivante illustre le schéma synoptique de la carte Arduino.

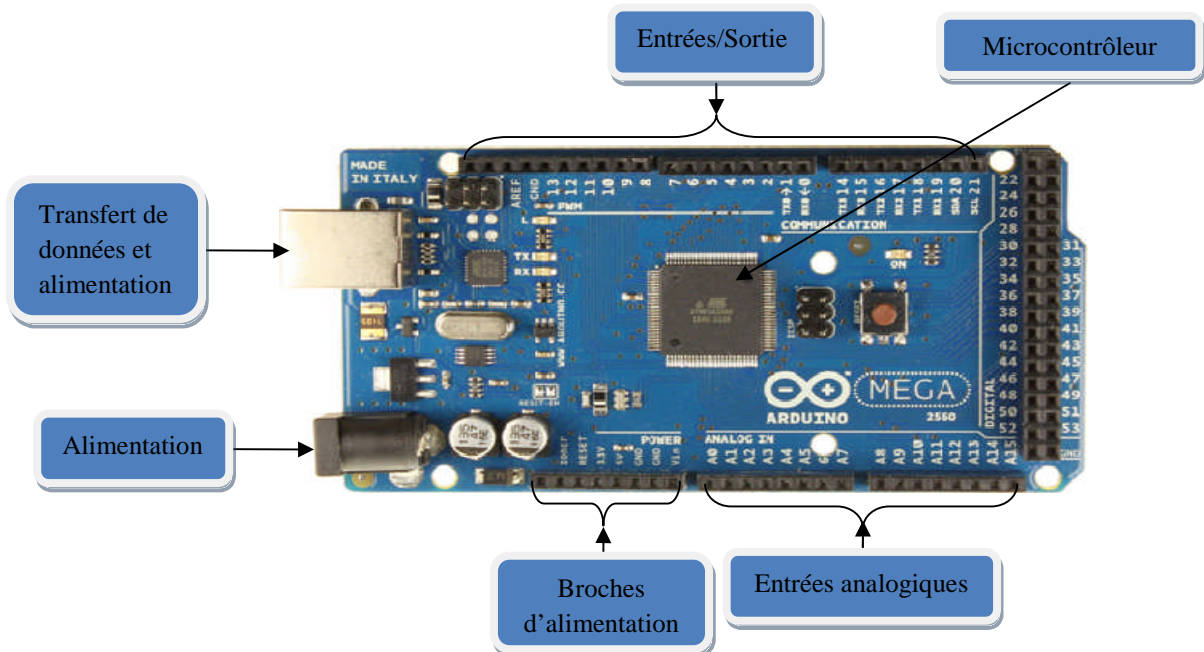


Figure1: Architecture de la carte Arduino méga 2560

Les éléments de base qui composent la carte Arduino sont :

#### II.3.1 Le microcontrôleur

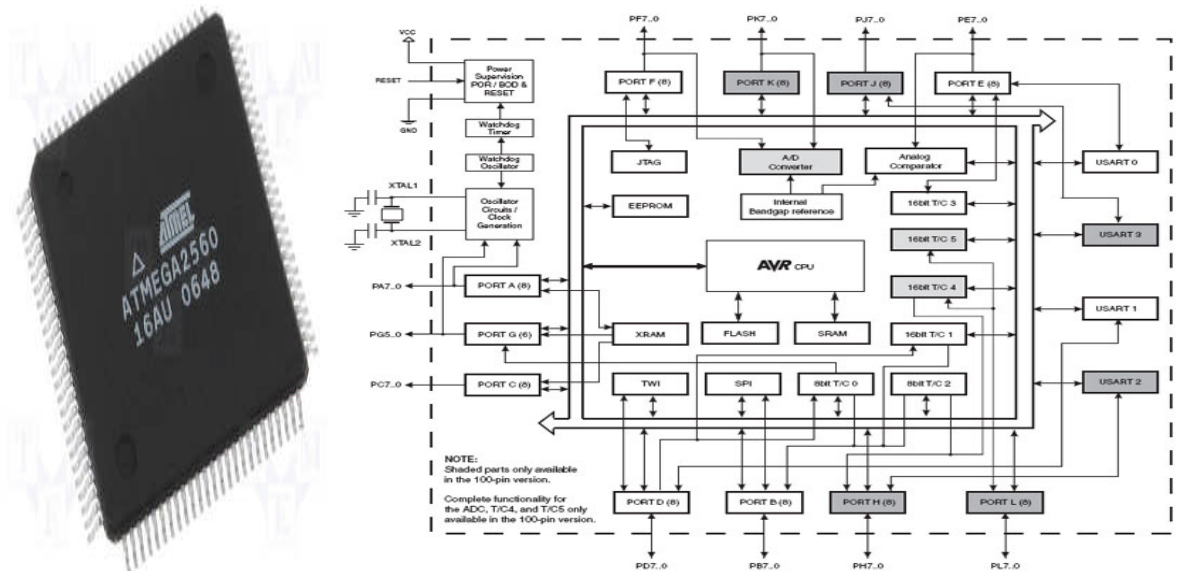
C'est le cerveau de la carte, permettant de recevoir le programme créé et stocké dans sa mémoire, puis l'exécuter. Grâce aux instructions de ce programme, il peut accomplir des tâches souhaitées, qui peuvent être : faire clignoter une LED, afficher des caractères sur un écran, commander des moteurs comme dans notre système, envoyer des données à un ordinateur...etc.

##### II.3.1.1 Caractéristiques du microcontrôleur ATMEGA 2560

- 135 instructions puissantes, la plupart d'exécution simple de rythme.
- 32x8 registres d'usage universel de fonctionnement.
- Opération entièrement statique.
- Jusqu'à 16 MIPS de sorties à 16 Mhz.
- Multiplicateur de cycle de Sur-Morceau2.
- Mémoires non-volatiles de programme et de données.
- Bytes 64K/128K/256K de flash Individu-Programmable de Dans-Système.
- Resistance : 10.000 écrire /cycles d'effacement.
- 4K Bytes EEPROM.
- Serrure de programmation pour la sécurité de logiciel.

- Quatre canaux à 8 bits de PWM.
- Interface série à 2 fils orienté vers le byte.
- Puissance sur la remise et la détection programmable d'arrêt partiel.

La figure ci-dessous montre la structure interne du microcontrôleur ATMEGA 2560



**Figure2:** Architecture du microcontrôleur ATMEGA 2560

### II.3.2 Le entrées-sorties de la carte

La carte Arduino possède 15 entrées analogiques, 54 entrées / sorties numériques dont les pins de (1 à 13) peuvent être utilisés comme des entrées/sorties analogiques.

- Les pins utilisés en numérique (PWM) ont une sortie binaire soit (0 logique qui correspond à l'état bas du pin ou 0V) ou (1 logique qui correspond à l'état haut du pin ou 5V) ce qui amène le fonctionnement des pins on mode tout ou rien.
- Les pins utilisés en analogique ont une sortie de valeur (0 à 255)

### II.3.3 Mémoire :

Le ATmega2560 a 128 Ko de mémoire flash pour stocker le code (dont 4 Ko est utilisée pour le boot loader), 8 Ko de SRAM et 4 Ko de mémoire EEPROM (qui peuvent être lues et écrites avec la bibliothèque EEPROM)

### II.3.4 L'alimentation

L'alimentation qui assure la distribution d'énergie aux différents composants de la carte Arduino, peut être effectuée via une connexion USB ou une alimentation externe. La source de courant est sélectionnée automatiquement.

L'alimentation externe (non USB) peut provenir d'un adaptateur CA/CC ou de piles, l'adaptateur peut être connecté par une fiche positive centrale de 2,1 mm dans l'alimentation de la carte. Les câbles de l'accu peuvent être insérés dans les prises de terre et de tension du connecteur d'alimentation.

La carte peut fonctionner sur une alimentation externe de 6 à 20 volts. Si moins de 7 volts sont fournis, la broche 5 V peut éventuellement fournir moins de 5 volts et la carte peut alors être instable. Si plus de 12 V sont utilisés, le régulateur de tension peut surchauffer et abîmer la carte. La tension recommandée varie entre 7 et 12 volts.

Les broches d'alimentation sont les suivantes :

- Tension d'entrée : la tension d'entrée de la carte Arduino quand une alimentation externe est utilisée (en opposition aux 5 volts de la connexion USB ou d'autres sources de courant régulées). On peut fournir une tension par cette prise ou, si l'alimentation est effectuée par un câble d'alimentation, y accéder via cette prise.
- 5 V : cette broche fournit une tension régulée de 5V. La carte peut être alimentée par le câble d'alimentation en courant continu, le connecteur USB (5V) ou la prise de terre sur la carte (7-12 V). Alimenter la tension par les broches 5 V ou 3,3 V contourne le régulateur et peut endommager la carte.

Une fois le programme chargé, on peut débrancher le cordon USB et connecter la carte soit à une pile, soit à un transformateur.

### II.3.5 Visualisation

Les trois "points blancs" entourés en rouge sont en fait des LED dont la taille est de l'ordre du millimètre. Ces LED servent à deux choses :

- Celle tout en haut du cadre : elle est connectée à une broche du microcontrôleur qui peut servir pour tester le matériel. Quand on branche la carte au PC, elle clignote quelques secondes.
- Les deux LED du bas du cadre : servent à visualiser l'activité sur la voie série (une pour l'émission et l'autre pour la réception). Le téléchargement du programme dans le microcontrôleur se faisant par cette voie, on peut les voir clignoter lors du chargement.

### II.3.6 La connectique

La carte Arduino ne possédant pas de composants (résistances, diodes, moteurs...) qui peuvent être utilisés pour un programme, mis à part la LED connectée à la broche 13 du microcontrôleur, il est nécessaire de les rajouter. Mais pour ce faire, il faut les connecter à la carte. C'est là qu'intervient la connectique de la carte.

Sur les Arduino et sur beaucoup de cartes compatibles Arduino, les connecteurs se trouvent au même endroit. Cela permet de fixer des cartes d'extension, appelée "shield".

Cette connectique est importante et a un brochage qu'il faudra respecter. Nous le verrons quand nous apprendrons à faire notre premier programme. C'est avec cette connectique que la carte est "extensible", car l'on peut y brancher tous types de montages et modules.

Par exemple, la carte Arduino peut être étendue avec des shields, comme le « **Shield Ethernet** » qui permet de connecter cette dernière à internet.

### II.3.7 Carte d'extension « SHIELD »

La carte EASYCON1 permet de connecter facilement 20 entrées-sorties à l'interface Arduino Méga. Les embases jacks stéréo 2,5mm servent à la connectique avec les cordons et les cartes I/O présentées. Extensible selon nos besoins, grâce à ces connecteurs stockables, et l'adjonction d'une carte EASYCON2 (18 entrées supplémentaires).

Deux kits optionnels de composants permettront d'étendre les capacités du shield en lui ajoutant :

- une commande de deux moteurs à courant continu (contrôle de la vitesse et de la direction).
- une ou deux "Banks" de mémoire EEPROM externe pour le stockage des projets autonomes.

### II.3.7.1 Exemple d'usage de SHIELD pour connecter des moteurs à courant continu

Comme on a évoquée précédemment l'existence des cartes shields pour Arduino. Voici un exemple qui permet de commander des moteurs en position et en vitesse en toute facilité, et offre aussi d'entrées/sorties supplémentaires

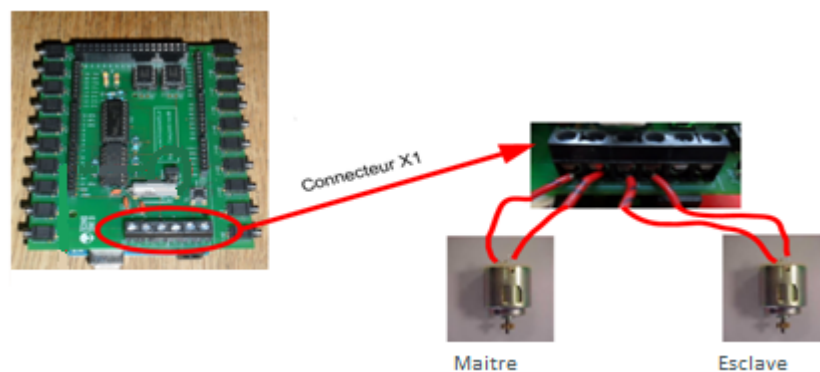


Figure3 : kit de commande de deux moteurs à (CC)

## II.4 Architecture SOFTWARE de la carte

Après la détermination des différentes entrées, sorties ainsi que la définition des conditions requises pour l'utilisation de la carte Arduino, nous allons décrire dans cette partie les éléments appropriés à l'architecture SOFTWARE de la carte, qui sont :

### II.4.1 Logiciel

L'environnement de programmation Arduino (IDE en anglais) est une application écrite en Java inspirée du langage Processing.

L'IDE permet d'écrire, de modifier un programme et de le convertir en une série d'instructions compréhensibles pour la carte.

#### II.4.1.1 L'interface logicielle

Sur un ordinateur, le logiciel de programmation de la carte Arduino sert d'éditeur de code (langage proche de C); une fois le programme tapé ou modifié au clavier, il sera transféré et mémorisé dans la carte à travers la liaison USB, le câble USB alimente à la fois en énergie la carte et transporte aussi l'information.

#### II.4.1.2 Structure de logiciel

Le logiciel open source (Windows, linux ou mac) représenté par la figure ci-dessous fournit avec l'Arduino est un éditeur de texte qui permet :

- a) de programmer la carte en utilisant un langage simple proche du C.
- b) de communiquer avec la carte grâce au "terminal série" (faire apparaître des informations de la carte sur l'écran de l'ordinateur)

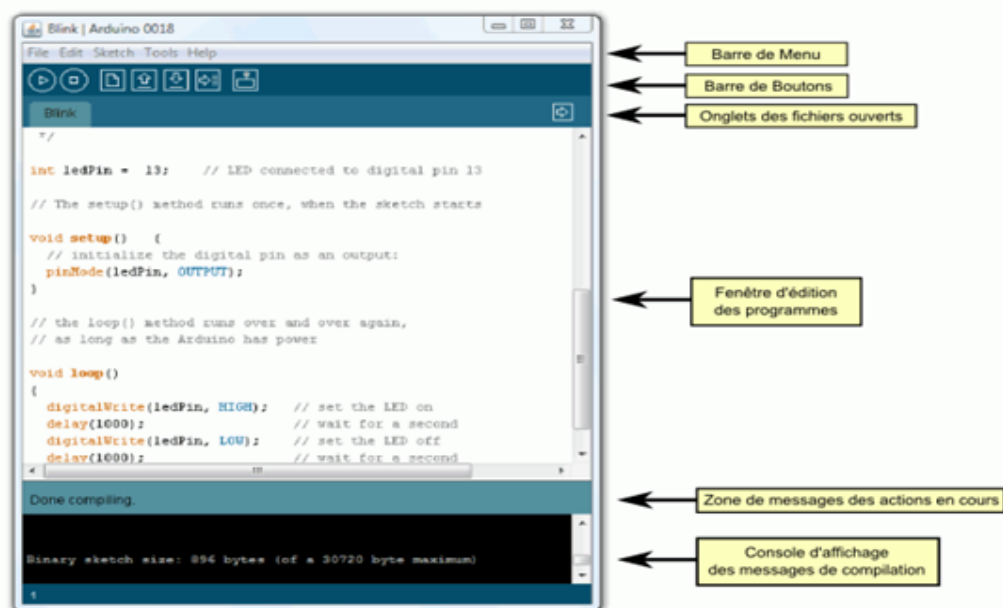


Figure3: Structure d'un programme Arduino

### II.4.1.3 Structure de programme

Comme il est présenté sur l'interface du logiciel, le programme est lu par le microcontrôleur de haut vers le bas.

Une variable doit être déclarée avant qu'elle soit utilisée par une fonction.

La structure minimale est constituée :

- en tête : déclaration des variables, des constantes, indication de l'utilisation de bibliothèques etc.
- un setup (= initialisation) cette partie n'est lue qu'une seule fois, elle comprend les fonctions devant être réalisées au démarrage (utilisation des broches en entrées ou en sortie, mise en marche du midi, du port série de l'I2C, etc.)
- une loop (boucle) : cette partie est lue en boucle ! C'est ici que les fonctions sont réalisées.

En plus de cette structure minimale, on peut ajouter :

- des « sous-programmes » ou « routines » qui peuvent être appelées à tout moment dans la boucle, très pratiqué pour réaliser des morceaux de codes répétitifs.
- Des « callbacks », ce sont des fonctions qui sont rappelées automatiquement depuis une bibliothèque.

### II.4.1.4 Programmation

Arduino Méga peut être programmée avec le logiciel Arduino, l'ATMEGA 2560 sur Arduino Méga vient avec un boot loader qui permet de télécharger un nouveau code sans l'utilisation d'un programme externe.

La programmation de cette carte est composée de deux fonctions principales :

**Setup () :**

Cette fonction est appelée dès l'exécution du programme. Elle a pour fonction d'initialiser les variables, indiquer les modes des broches, déclarer les bibliothèques, cette fonction s'exécute une seule fois, après chaque mise sous tension ou la réinitialisation de la carte Arduino.

**Loop() :**

Cette fonction permet au programme de se répéter en forme de boucle principale.

**II.4.2 Développement d'un projet**

Le développement sur Arduino est très simple :

- on code l'application : Le langage Arduino est basé sur les langages C/C++, avec des fonctions et des bibliothèques spécifiques à Arduino (gestions des e/s).
- on relie la carte Arduino au PC et on transfère le programme sur la carte.
- on peut utiliser le circuit.

Le logiciel de programmation des modules Arduino est une application Java multiplateformes (fonctionnant sur tout système d'exploitation), servant d'éditeur de code et de compilateur, et qui peut transférer le firmware (et le programme) au travers de la liaison série (RS232, Bluetooth ou USB selon le module).

**II.4.3 Utilisation de la carte Arduino comme périphérique d'entrée sortie pour Matlab**

Cette solution consiste à utiliser la carte Arduino comme une interface d'entrées/sorties pour le logiciel Matlab, assurant le transfert de données entre l'interface de commande (Matlab) et la partie opérative (actionneurs).

Pré-charger un programme dans la carte Arduino afin que celle-ci fonctionne en serveur : Ce programme consiste à "écouter" les requêtes envoyées via la liaison série (USB) par Matlab et de répondre à ces requêtes en renvoyant l'état d'une entrée ou en modifiant l'état d'une sortie.

Ces mêmes entrées/sorties sont vues dans Matlab comme des entrées logiques ou analogiques (Utilisation du CAN) ou des sorties numériques (mode PWM). Cette communication peut être réalisée lorsque on installe les mises à jour nécessaires pour Matlab. et on va avoir la bibliothèque montrée dans la figure (fig.4)

### II.4.3.1 La bibliothèque Arduino IO

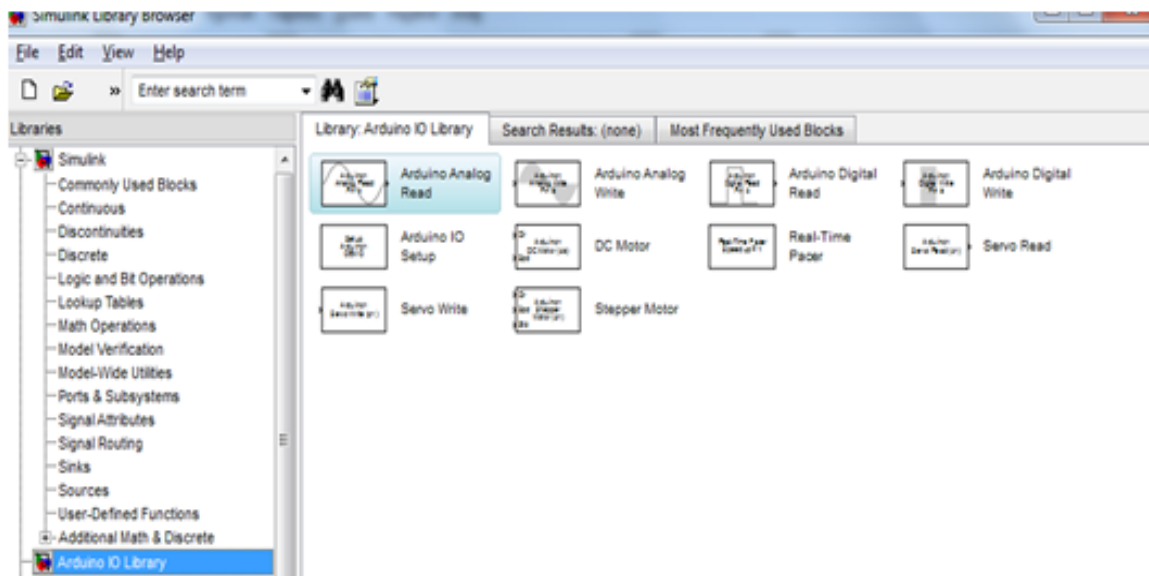


Figure4: bibliothèque Arduino sur Matlab (simulink)

Ces blocs représentés sur la figure ci-dessus permettent d'utiliser simulink comme interface de commande et de gestion d'informations en mode connexion direct avec la carte.

## II.5 Logiciel à utiliser dans la suite de projet

### MATLAB :

Il joue le rôle d'une plate forme, où tous les autres composants s'exécutent, c'est l'environnement le plus utilisés dans le monde par les ingénieurs, car il fournit plusieurs routines sophistiquées de calculs numériques, et encore certaines performances comme la vitesse d'exécution des programmes, ainsi que sa facilité et précision dans la gestion des données en communiquant avec la carte d'acquisition.

## IV.6 Conclusion

Dans ce chapitre, notre étude s'est focalisée sur la présentation de la carte (Arduino), et on a commencé par un bref historique et quelques détails concernant la création et l'utilité de cette carte. Puis on a parlé de son côté matérielle, enfin on a présenté le logiciel et le langage de programmation et Matlab qui est devenu un outil de commande et de programmation de cette dernière.

La compréhension de données exposées dans ce chapitre nous offre des bases de connaissances pour l'usage de la carte d'acquisition.

# Chapitre III

## Ajustage des Paramètres de Régulateurs PID

### III.1 Introduction

Dans ce chapitre, nous présentons un bref rappel théorique sur les différents outils de commandes classiques dont nous aurons besoin pour la synthèse des commandes.

Dans la plupart des processus industriels, il est indispensable de maîtriser certains paramètres physiques, en automatique lorsque l'on souhaite atteindre une certaine vitesse, température, position, angle...etc. Il est donc très souvent nécessaire d'avoir recours à un asservissement, permettant d'atteindre et de maintenir une consigne en utilisant une mesure. Il s'agit donc d'un système bouclé, capable de déterminer la fonction assurant la poursuite de la trajectoire de consigne par la sortie.

Les régulateur PID est une méthode qui répond à plus du 90% des besoins industriels et qui a fait ses preuves en donnant de bons résultats.

### III.2 Principe général de la régulation

Dans la plupart des appareils et installations industrielles, tertiaires et mêmes domestiques, il est nécessaire de maintenir des grandeurs physiques à des valeurs déterminées, en dépit des variations externes ou internes influant sur ces grandeurs. Le niveau d'un réservoir d'eau, le débit d'une conduite de gaz, vitesse ou position des moteurs, étant par nature variables, doivent donc être réglés par des actions convenables sur le processus considéré. Si les perturbations influant sur la grandeur à contrôler sont lentes ou négligeables, un simple réglage (dit en boucle ouverte) permet d'obtenir et de maintenir la valeur demandée.

Cependant, dans la majorité des cas, ce type de réglage n'est pas suffisant, parce que trop grossier ou instable. En revanche, il faut comparer en permanence, la valeur mesurée de la grandeur réglée à celle que l'on souhaite obtenir et agir en conséquence sur la grandeur d'action, dite grandeur réglante. On a, dans ce cas, constitué une boucle de régulation et plus généralement une boucle d'asservissement qui comprend trois maillons indispensables : l'organe de mesure, l'organe de régulation et l'organe de contrôle. L'organe de mesure récupère les principales grandeurs servant à contrôler le processus. L'organe de régulation compare ces mesures aux valeurs souhaitées (consignes). En cas de non concordance des valeurs de mesure et des valeurs de consigne, l'organe de régulation envoie un signal de commande à l'organe de contrôle (moteur, vanne ...etc.), afin que celui-ci agisse sur le processus.

### III.3 Les caractéristiques du régulateur PID

Le régulateur standard le plus utilisé dans l'industrie est le régulateur PID (proportionnel intégral dérivé), car il permet de régler à l'aide de ses trois paramètres les performances (amortissement, temps de réponse, ...etc.) d'un processus modélisé par un deuxième ordre. Si la dynamique dominante du système est supérieure à un deuxième ordre, ou si le système contient un retard important ou plusieurs modes oscillants, le régulateur PID n'est plus

adéquat et un régulateur plus complexe (avec plus de paramètres) doit être utilisé, tout dépend de la sensibilité aux variations des paramètres du procédé.

### III.4 Asservissements et influence des coefficients

#### III.4.1 Action proportionnelle P

L'asservissement de type P est le plus simple qui soit. Il s'agit d'appliquer une correction proportionnelle à l'erreur corrigeant de manière instantanée tout écart de la grandeur à régler.

##### ✓ Loi de commande du régulateur p

$$u(t) = K_P \cdot \varepsilon(t) \xrightarrow{L} u(p) = K_P \cdot \varepsilon(p) \dots\dots\dots(III.1)$$

Son rôle est d'amplifier virtuellement l'erreur pour que le système réagisse plus vivement, comme si l'erreur était plus grande qu'elle ne l'est en réalité.

Il permet de vaincre les grandes inerties du système et diminue le temps de montée en donnant de la puissance au moteur (plus l'erreur est grande, plus on donne de puissance au moteur). Lorsque l'on augmente  $K_P$ , le système réagit plus vite et l'erreur statique s'en trouve améliorée, mais en contrepartie le système perd en stabilité. Le dépassement se fait de plus en plus grand, et le système peut même diverger dans le cas d'un  $K_P$  démesuré.

Néanmoins, il faut toujours une certaine tension aux bornes du moteur pour que celui-ci puisse tourner. C'est pourquoi, lorsque l'on s'approche de la valeur demandée, l'erreur n'est plus assez grande pour faire avancer le moteur, ce qui fait que l'on n'atteint jamais vraiment la valeur demandée. Il subsiste alors une erreur statique, qui est d'autant plus faible que  $K_P$  est grand.

#### III.4.2 Action intégrale I

L'action intégrale agit proportionnellement à la surface de l'écart entre la consigne et la mesure, et elle poursuit son action tant que cet écart n'est pas nul. On dit que l'action intégrale donne la précision statique (Elle annule l'erreur statique). L'action intégrale est conditionnée par le temps d'intégrale  $T_i$ .

##### ✓ Loi de commande du régulateur I

$$u(t) = \frac{1}{T_i} \int_0^t \varepsilon(\tau) d\tau \dots\dots\dots(III.2)$$

Un dosage très important de l'action intégrale engendre une instabilité de la boucle de régulation. Ainsi, il faut trouver un compromis entre la stabilité et la rapidité.

L'ajout du terme intégral permet d'améliorer la précision mais en contrepartie, il introduit malheureusement un déphasage de  $-\frac{\pi}{2}$  ce qui risque de rendre le système instable du fait de la diminution de la marge de phase.

Enfin, le correcteur intégral présente le défaut de saturer facilement si l'écart ne s'annule pas rapidement ce qui est le cas des systèmes lents. En effet, tout actionneur est limité par exemples, un moteur est limité en vitesse, une vanne ne peut pas être plus que totalement ouverte ou totalement fermée. Il se peut que la variable de commande amène l'actionneur a sa limite ce qui suspend la boucle de retour et le système aura une configuration assimilable à une boucle ouverte puisque l'actionneur demeurera saturé indépendamment de la sortie du système.

**III.4.3 Action dérivé D**

Action dérivé amplifie les variations brusques de la consigne. Elle a une action opposée à l'action intégrale. Cette fonction est remplie par l'opérateur mathématique : « dériver par rapport au temps ». Ainsi, dans un régulateur, on définit l'action dérivé à partir du temps dérivé  $T_d$  avec :

$$u(t) = T_d \frac{d\varepsilon(t)}{dt} \dots\dots\dots (III.3)$$

Le temps dérivé  $T_d$  s'exprimer en unité de temps.

$T_d$ : Représente l'écart, en temps entre les réponses proportionnelles seules (P), et proportionnelles dérivées(PD).

$T_d$ : Est donc le temps d'avance d'une réponse PD par rapport à une réponse en P seule.

Afin de limiter la sortie d'un régulateur ayant une action dérivée, en pratique l'action dérivée est filtrée en ajoutant un élément de premier ordre. L'action dérivée pure  $T_d \cdot p$  devient alors :

$$u(p) = \frac{T_d p}{1+\tau p} \dots\dots\dots (III.4)$$

**III.4.4 Asservissement PID**

En pratique, à une catégorie donnée de systèmes à asservir correspond un type de correcteur adopté. Pour effectuer un choix judicieux, il faut connaître les effets des différentes actions, proportionnelle, intégrale et dérivée.

Un régulateur PID est obtenu par l'association de ces trois actions et il remplit essentiellement les trois fonctions suivantes :

1. Il fournit un signal de commande en tenant compte de l'évolution du signal de sortie par rapport à la consigne.
2. Il élimine l'erreur statique grâce au terme intégrateur.
3. Il anticipe les variations de la sortie grâce au terme dérivateur.

Les termes proportionnel et intégral peuvent amener un dépassement de la consigne et des oscillations. Cela implique pour le moteur des inversions de polarité, ce qui est loin d'être idéal. Pour limiter ce phénomène indésirable, on introduit un troisième élément : le terme dérivé. Son action va dépendre du signe et de la vitesse de variation de l'erreur, et sera opposée à l'action proportionnelle.

Elle devient prépondérante aux abords de la valeur demandée lorsque l'erreur devient faible, que l'action du terme proportionnel faiblit et que l'intégrale varie peu, elle freine alors le système, limitant le dépassement et diminuant le temps de stabilisation.

$$u(t) = K_p \varepsilon(t) + K_i \int_0^t \varepsilon(\tau) d\tau + K_d \frac{d\varepsilon(t)}{dt} \xrightarrow{L} u(p) = K_p \varepsilon(p) + K_i \frac{\varepsilon(p)}{p} + K_d p \varepsilon(p) \dots\dots (III.5)$$

On obtient alors un asservissement composée d'un terme proportionnel, un terme intégral et un terme dérivé : on parle d'asservissement PID, comme le montre la figure ci-dessous.

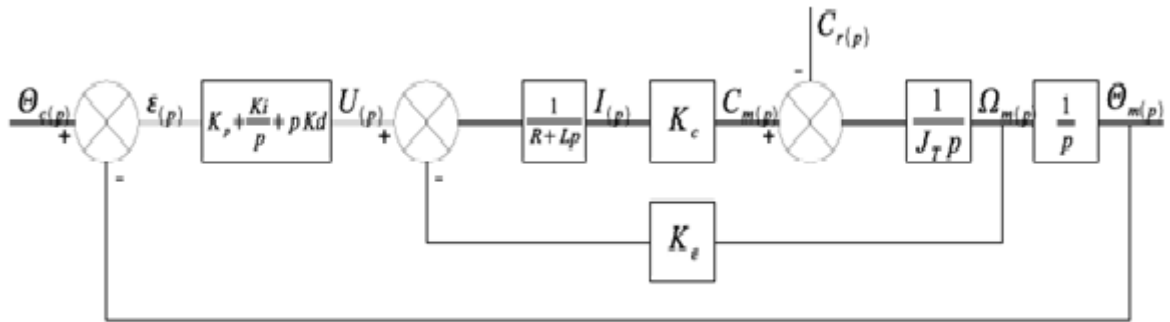


Figure 1: schéma bloc de l'implantation du PID en régulation de position

✓ **Fonction de transfert du régulateur PID série**

$$G_R(p) = K_p \left[ \left( 1 + \frac{1}{p \cdot T_i} \right) (1 + T_d \cdot p) \right] \dots\dots\dots (III.6)$$

✓ **Fonction de transfert du régulateur PID parallèle**

$$G_R(p) = K_p + \frac{1}{p \cdot T_i} + T_d \cdot p \dots\dots\dots (III.7)$$

✓ **Fonction de transfert du régulateur PID mixte**

$$G_R(p) = K_p \left( 1 + \frac{1}{p \cdot T_i} + T_d \cdot p \right) \dots\dots\dots (III.8)$$

❖ **Récapitulatif de l'action des coefficients**

Coefficients	Temps de montée	Temps de stabilisation	dépassement	Erreur statique
$K_p$	Diminue	Augmente	Augmente	Diminue
$K_i$	Diminue	Augmente	Augmente	Annulée
$K_d$	-	Diminue	Diminue	-

**III.5 Réglage des coefficients**

Le réglage d'un PID consiste à trouver les meilleurs coefficients  $K_p$ ,  $K_i$  et  $K_d$  dans le but d'obtenir une réponse adéquate du procédé et de la régulation. L'objectif est d'être robuste, rapide et précis tout en limitant les dépassements.

### III.5.1 Méthode de Ziegler-Nichols

La méthode empirique de Ziegler-Nichols est très répandue dans l'industrie ou chez les automaticiens pour régler les correcteurs de ce type de chaîne. Elle a l'avantage de ne pas nécessiter de modélisation précise du système asservi, mais se contente d'essais expérimentaux, ce qui rend cette méthode très simple.

On fixe  $K_i$  et  $K_d$  à 0 et on augmente le gain du correcteur proportionnel pur  $K_p$  jusqu'à obtenir des oscillations persistantes (pour des moteurs, il s'agira plutôt d'une vibration autour de la position demandée).

On mesure alors  $(K_p)_{lim}$ , le gain du correcteur proportionnel à la juste oscillation, et  $T_{osc}$ , la période des oscillations.

La méthode de Ziegler-Nichols indique alors des valeurs des coefficients permettant d'obtenir un bon compromis entre précision, rapidité et stabilité.

$$\begin{cases} K_p = 0,6 \cdot (K_p)_{lim} \\ K_i = \frac{2}{T_{osc}} \\ K_d = 0,125 \cdot T_{osc} \end{cases}$$

Ensuite, on peut légèrement faire varier ces valeurs moyennes selon le compromis précision/rapidité/stabilité recherché. Dans le cas d'un asservissement en position, on aura tendance à monter un peu plus le gain intégral pour avoir une erreur statique faible, quitte à « Accoster » plus lentement à la valeur demandée.

## III.6 Description et modélisation du moteur à courant continu

### III.6.1 Caractéristiques des moteurs utilisés

Le moteur utilisé dans notre réalisation est un moteur réducteur qu'on peut alimenter avec 12 V. Ce moteur possède un réducteur (rapport 30:1) et un encodeur intégré, dont la consommation de 530 mA environ. (Et 150 mA à vide). Son courant de blocage est de 2,5 A, et sa Vitesse de rotation est de 170 t/min (216 t/min à vide), son couple est de 1,5 kg.cm. Le nombre d'impulsion de son encodeur est 360 impulsions par tour de dimensions: Ø28,5 x 86,6 mm (axe 10 mm inclus). Il a six fils de branchement, à savoir le fil :

- Violet (1) : capteur à effet Hall B V out
- Bleu (2) : capteur à effet Hall A V out
- Vert (3) : capteur à effet Hall "terre"
- Brun (4) : capteur à effet Hall Vcc
- Rouge (5) : + Moteur
- Noir (6) : - Moteur



Figure 2 : Moteur réducteur avec encodeur de quadrature

### III.6.2 Mise en œuvre

#### III.6.2.1 Partie mécanique

Un encodeur optique incrémental à quadrature de phase est relié au rotor du moteur, ce qui nous permet de récupérer une information sur l'incrémentangulaire du rotor, à partir de laquelle on peut calculer la vitesse de rotation du rotor et sa position angulaire par rapport à sa position initiale. Il s'agit d'un disque en rotation doté de deux rangées de fentes décalées. Un système optique détecte ces fentes et émet 360 impulsions par tour de rotor, sur deux signaux en quadrature de phase.

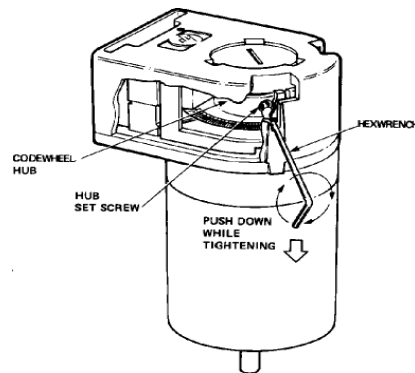


Figure 3 : Utilisation d'un encodeur optique sur l'axe du moteur

#### III.6.2.2 Le principe de fonctionnement de l'encodeur

L'encodeur incrémental est le capteur le plus utilisé pour mesurer les variations de position car il est très peu onéreux et facile à interfacer. Il est monté sur l'axe de la roue, pour délivrer des informations de rotation élémentaires qui, par intégration, donnent une mesure du mouvement global. Pour cela, on peut employer des encodeurs de résolution moyenne montés sur l'axe de la roue.

### III.6.3 Partie électronique

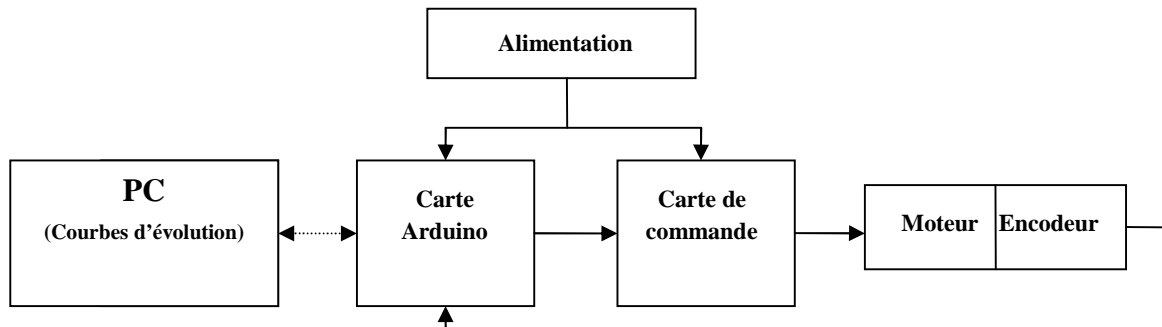
Deux cartes électroniques sont utilisées pour cet asservissement :

- La première, s'occupe de l'acquisition et du traitement des signaux. Elle s'articule autour d'un microcontrôleur (AtMega2560 à 16MHz) qui permet une grande souplesse dans la programmation de l'asservissement. Ce microcontrôleur acquiert les signaux venant de l'encodeur optique et calcule les corrections à appliquer en fonction de la commande initiale. Une liaison série permet de tester diverses valeurs pour les coefficients du PID et d'en récupérer les courbes de réponse (vitesse et/ou position) sur un ordinateur.
- La deuxième, permet de contrôler la tension à fournir aux moteurs (par une modulation PWM) ainsi que le sens de déplacement. Dans notre cas, on les a commandés seulement dans le sens horaire qui est imposé par notre carte de commande. Une modulation PWM (Pulse-Width Modulation, ou Modulation de Largeur d'Impulsion) permet d'obtenir un contrôle

proportionnel sur un moteur sans avoir à varier la tension de commande, selon le principe du hacheur électrique :

$$\begin{cases} V = V_{max} & \text{pour } 0 < t < \alpha T \\ V = 0 & \text{pour } \alpha T < t < T \end{cases} \Rightarrow V_{moy} = \alpha V_{max} \text{ avec } 0 < \alpha < 1$$

Cette carte joue le rôle d'une carte d'interface entre la carte de commande de la partie motrice comportant des composants électroniques, elle nous permet de commander les moteurs à courant continu suivant les ordres fournis par la carte d'acquisition.



Pour pouvoir commander les moteurs en deux sens de rotation (horaire/antihoraire), on aura besoin d'une carte de commande muni d'un pont en H.

### III.6.3.1 Commande d'un moteur avec un pont en H(Hacheur)

Lorsqu'on veut commander un moteur à courant continu on est souvent obligé d'inverser la polarité.

Le pont en H est une structure électronique servant à contrôler cette polarité aux bornes d'un dipôle, il est composé de quatre éléments de communication généralement disposés schématiquement en une forme de H d'où le nom. Cette structure se trouve dans plusieurs applications de l'électronique de puissance incluant le contrôle des moteurs, les convertisseurs et hacheurs. Il se présente sous différentes formes passant par les circuits intégrés pour les applications de faibles et moyennes puissances.

#### III.6.3.1.1 Principe de fonctionnement

Le pont est utilisé en activant les commutateurs de différentes combinaisons pour obtenir le branchement voulu. Le tableau suivant résume les combinaisons permises avec S1, S2, S3 et S4 qui sont des interrupteurs.

Toutes les combinaisons qui ne sont pas présentées dans le tableau sont interdites et créent un court-circuit de la source. Le courant de référence pour le moteur est assuré comme étant de gauche à droite.

S1	S2	S3	S4	Résultats
x	x	x	x	Aucune tension aux bornes de la charge
ok	x	x	ok	Courant positif à travers la charge
x	ok	ok	x	Courant négatif à travers la charge
x	x	ok	ok	Charge court-circuitée
ok	ok	x	x	Charge court-circuitée

### III.6.4 Partie informatique embarquée

#### III.6.4.1 Acquisition des données

A l'intervalle de temps régulier, le microcontrôleur lit la position angulaire donnée par le capteur (encodeur) et met à jour ses variables internes de position. La mesure des erreurs (erreur de position) se fait par une simple différence entre la valeur demandée et celle mesurée. C'est la fréquence de rafraîchissement des erreurs et de recalcul de la correction qui joue un rôle significatif dans la précision de l'asservissement :

- Si la fréquence de rafraîchissement est trop faible, le trop grand écart entre deux recalculs de correction opère une discrétisation du temps trop grossière, ce qui amoindrit la précision de l'asservissement.
- Si la fréquence de rafraîchissement est trop élevée, bien que l'on se rapproche du modèle de temps continu la précision de l'asservissement s'amoindrit puisque le nombre de pas comptés par le capteur entre deux recalculs de correction sera plus faible, donc moins précis.
- Une fréquence de 20Hz a été jugée empiriquement comme un assez bon compromis entre période d'échantillonnage et précision de chaque recalcul lors des expériences.

#### III.6.5 Partie informatique distante

Pour pouvoir régler l'asservissement, il a été nécessaire de récupérer l'historique d'évolution de diverses variables et d'en tracer la représentation graphique après adaptation des données brutes en un format plus exploitable.

1. Au début de chaque session de mesures, le microcontrôleur implémenté dans notre carte demande à l'ordinateur distant (via une connexion série) les paramètres à utiliser : coefficients du/des PID, consigne à atteindre, vitesse maximale, accélération maximale et durée de session.

Une fois ces paramètres renseignés, la session de mesures pouvait commencer.

2. Dès lors, en plus de gérer l'asservissement du moteur, le microcontrôleur renvoie à l'ordinateur (via la même connexion série) les variables intéressantes, et ce à chaque boucle de correction : position, vitesse, erreur, correction demandée et dépassement de tension (lorsque la tension exigée par le PID dépassait la tension admissible par le moteur).

3. Une fois ces données enregistrées, elles ont été confiées à un logiciel spécialisé dans l'exploitation de données et le tracé de courbes (Matlab/Simulink) afin d'en tracer la représentation graphique de manière simple après avoir traité les données brutes pour les convertir en unités intéressantes.

4. Une fois la session terminée, les variables internes (état de l'intégrale, position angulaire, ...) ont été réinitialisées avant de redemander les paramètres à utiliser pour une nouvelle session.

### III.7 La théorie du PID appliquée aux moteurs électriques

#### But à atteindre

On cherche à faire atteindre une certaine valeur à une variable du système. Dans notre cas (moteurs), il s'agit principalement d'une vitesse ou d'une position.

- Les effets des perturbations doivent être minimisés, voire effacés, et ce le plus vite possible (régulation)
- Les changements de consigne doivent être suivis rapidement et avec une bonne précision, si possible sans dépassement (poursuite)

Néanmoins, ces critères de performance sont souvent antagoniques. Généralement dans le réglage de l'asservissement en détermine un compromis.

La position du moteur esclave doit atteindre celle du moteur maître le plus parfaitement possible en minimisant l'écart même en présence des perturbations, qui est l'objectif de la réalisation.

#### III.7.1 Modélisation mathématique du moteur à courant continu

Un moteur électrique à courant continu est régi par les équations physiques découlant de ses caractéristiques électriques, mécaniques et magnétiques. En utilisant le Théorème du moment cinétique et des équations d'électromagnétique, on obtient un système d'équations différentielles linéaires :

$$\begin{cases} U(t) = e(t) + Ri(t) + L \frac{di(t)}{dt} \\ e(t) = K_e \cdot w(t) \\ C_m(t) = K_c \cdot i(t) \\ C_m(t) - C_r(t) = J_T \cdot \frac{dw(t)}{dt} \end{cases} \quad \begin{array}{l} U : \text{Tension appliquée au moteur} \\ e : \text{Force électromotrice} \\ i : \text{Intensité traversant le moteur} \\ w : \text{Vitesse de rotation du rotor} \\ C_m : \text{Couple moteur généré} \\ C_r : \text{Couple résistant} \end{array}$$

**Force électromotrice** : tension créée par le bobinage du moteur pour s'opposer à la variation du flux du champ magnétique le traversant.

On en déduit donc le modèle suivant dans le domaine de Laplace :

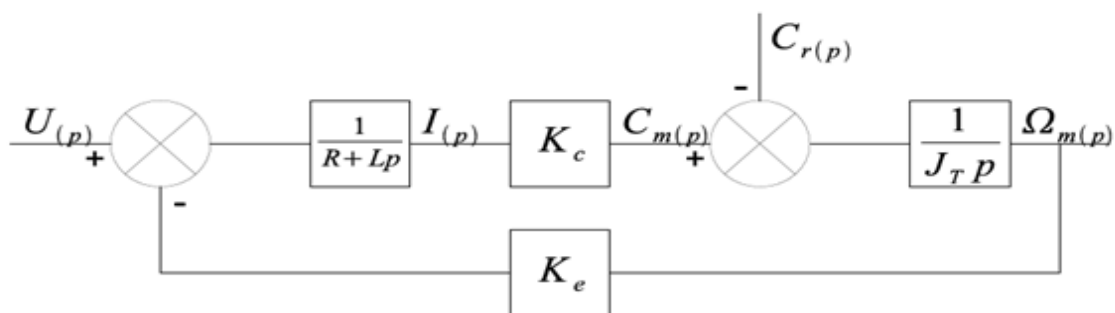


Figure 4 : Schéma bloc d'un modèle mathématique de moteur électrique

Le modèle utilisé pour la simulation est le modèle représentatif de notre système, tel que Le maître et l'esclave étant identiques, ils sont tous les deux représentés par les mêmes blocs. En injectant une consigne sur l'entrée du moteur maître, sa sortie vers l'entrée du moteur esclave, comme le montre la figure ci-dessous.

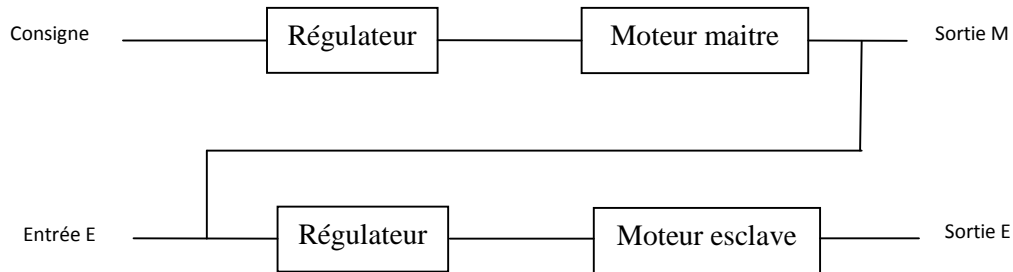


Figure 5 : modèle de simulation du système Maître/Esclave

### III.8 Limitations et approximations

#### III.8.1 Limitation de la tension de commande

Lors du réglage des coefficients du PID, il est primordial de se rappeler qu'un moteur n'est utilisable que dans une certaine plage de tensions ( $\pm 12V$  dans notre cas). Hors de cette zone, non seulement le comportement du moteur ne suivra plus obligatoirement le modèle établi, mais le moteur pourra éventuellement se détériorer irrémédiablement.

Nous ne sommes donc pas totalement libres de choisir les coefficients  $K_p$ ,  $K_i$  et  $K_d$ .

#### III.8.2 Approximations

Le choix du modèle de moteur peut induire certaines approximations :

- Le jeu mécanique des pièces notamment des réducteurs n'est pas pris en compte.
- L'usure des pièces n'est pas prise en compte.
- Le modèle n'est valable que pour une plage de tension et de courant au-delà de laquelle le comportement et la durée de vie du moteur sont incertains.

Le choix de la méthode d'implantation du PID peut elle aussi induire certaines approximations :

- Approximation des calculs du nombre dérivé et de l'intégrale (calculs approchés dans une implantation logicielle).
- Approximation en considérant le temps comme discret et non continu (fréquence de recalcul de la correction dans une implantation logicielle).
- Approximation au niveau des capteurs (nombre de pas des encodeurs, mauvais comptage de pas, conversion analogique-numérique, ...).

Finalement, l'erreur est faible, mais il est nécessaire d'en tenir compte. Pour minimiser son effet, diverses solutions sont envisageables :

- Utiliser un système hardware de dérivation et d'intégration (lequel reste soumis aux approximations des CAN).
- Diminuer  $\Delta t$  pour se rapprocher d'un modèle (idéal) où le temps est continu.
- Améliorer la précision des encodeurs.

- Recaler régulièrement le système (signal d'index sur les encodeurs, réaffectation des coordonnées en un point de coordonnées connues).

### III.9 Modèles simulés sous MATLAB/SIMULINK

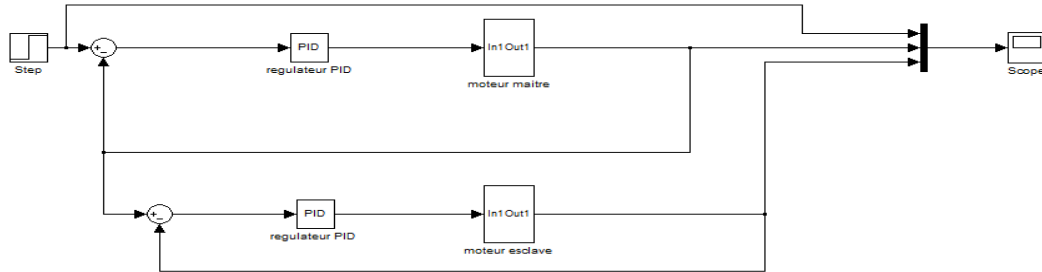


Figure 6 : Modèle de simulation du système Maître/Esclave

La figure 6 illustre les blocs simulés qui représente notre système maître/esclave, cette structure est inspiré du principe montré dans la figure 8 du chapitre I, où on a exprimé l'architecture de contrôle position-position généralisée et les méthodes de contrôle mises en œuvres dans les systèmes de téléopération.

Les blocs nommés respectivement moteur maître et esclave sont constitués du modèle représenté dans la figure 4 en exploitant les paramètres ( $R=471K\ \Omega$ ,  $J_T = 1.5Kg.Cm$ ,  $L$  est négligeable,  $C_r$  est faible) qui caractérise les deux moteurs.

Lors des premiers essais de simulation sans régulateur, on a obtenu un écart important, de ce dernier, on a remarqué que l'allure du l'esclave s'éloigne de celle du maître. A cet effet on a effectué plusieurs tests en introduisant ce régulateur afin d'aboutir à une erreur minimale en réglant ses coefficients  $k_p$ ,  $k_i$  et  $k_d$ .

L'exploitation des tests et résultats théoriques nous ont permet de synthétiser une loi de commande qui s'adapte aux dispositifs de communication utilisés dans la manipulation.

### III.10 Simulations et résultats expérimentaux

#### III.10.1 Cas théoriques

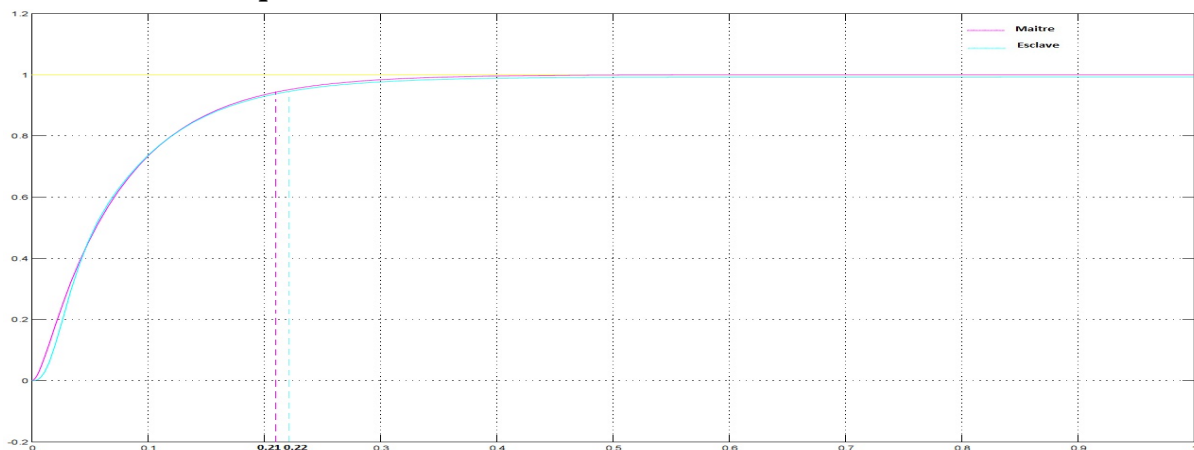


Figure 6: Résultats (théoriques) de simulation des deux positions (Maître/Esclave)

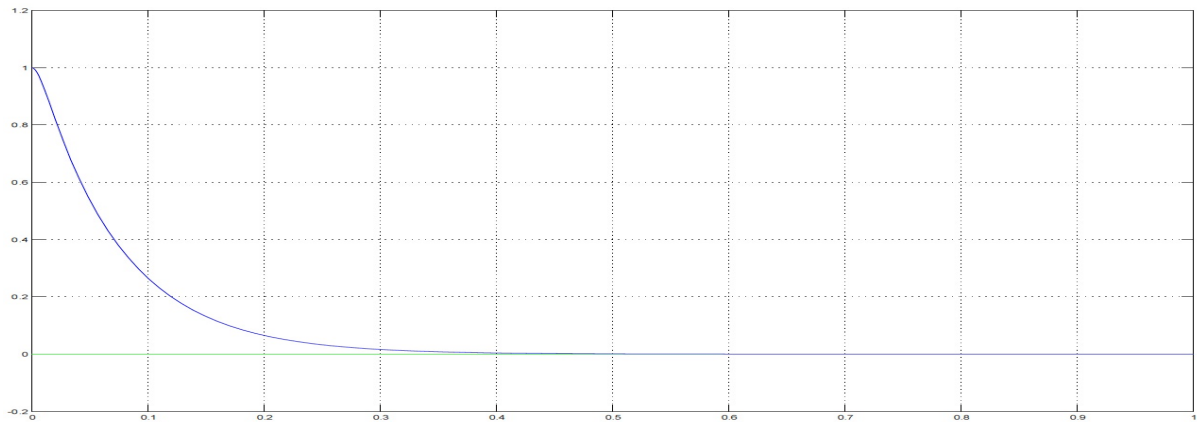


Figure 7 : erreur de position

### III.10.2 Cas pratiques

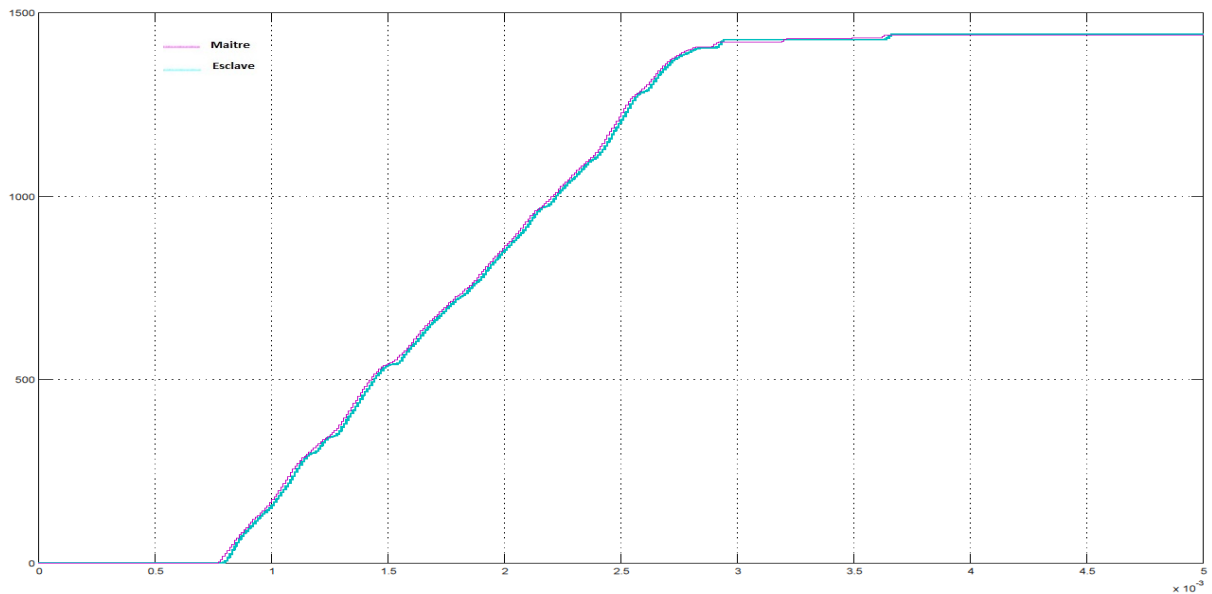


Figure 8 : Résultats (pratiques) de simulation des deux positions (Maître/Esclave) libre

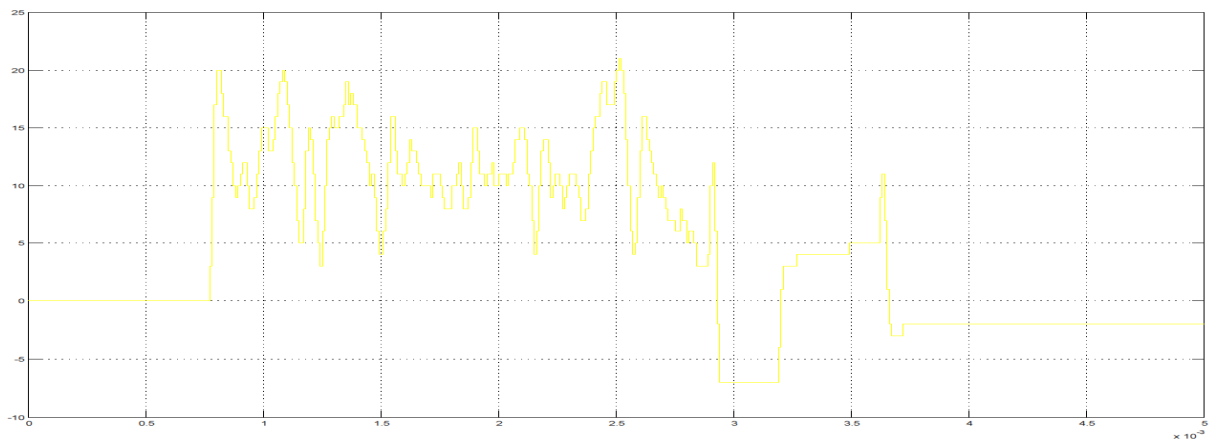
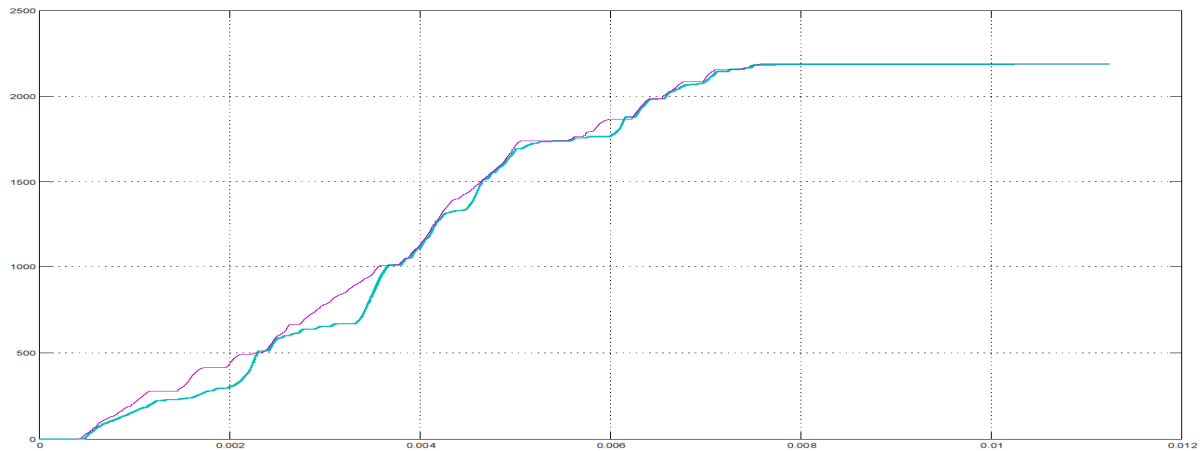
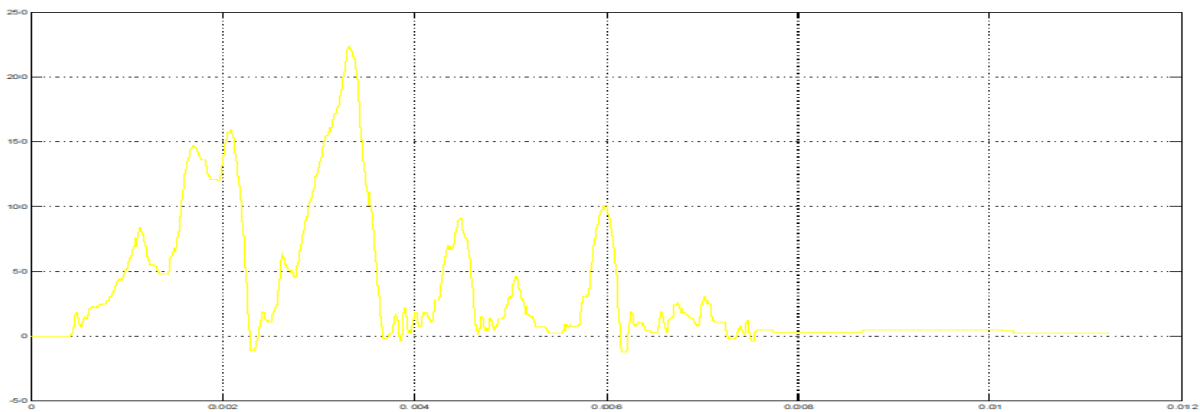


Figure 9 : Erreur de position (système libre)



**Figure 10 :** Résultats (pratiques) de simulation des deux positions (Maître/Esclave) avec perturbations



**Figure11 :** Erreur de position (système perturbé)

### III.10.3 Interprétation des résultats

#### III.10.3.1 Cas théorique

Les figures 6, 7 représentent respectivement les trajectoires des deux sites maîtres et esclave et l'écart entre les deux positions en mouvement libre.

L'observation de l'allure représentée dans la première figure citée, on remarque que les deux trajectoires sont presque identiques avec un temps de retard. Le contrôleur de l'esclave aura cependant bien une influence sur la qualité de transmission.

Des erreurs sont remarquables lors des simulations, d'une part au non connaissance du modèle précis et des incertitudes de modélisation et d'autre part au réglage des paramètres du contrôleur.

#### III.10.3.2 Cas pratique

Les figures 8, 9 illustrent respectivement les comportements du système, dont on a remarqué que l'erreur de position entre les deux moteurs varie entre des valeurs faibles dépendamment de l'influence de l'opérateur sur le site maître.

Le transfert des données entre les deux parties commande et opérative a engendré un temps de retard (retard de transmission) entre le maître et l'esclave. Le régulateur implémenté dans le système de commande, nous a permis d'obtenir deux trajectoires (maître/esclave) semblables en mouvement libre.

La présence d'une charge posée sur la tige du moteur esclave fait ralentir sa rotation par rapport à celle du maître, pour cela on peut dire que notre système est soumis aux perturbations, La figure 10 décrit les conséquences d'application.

Ces résultats font apparaître l'influence des perturbations sur le comportement de l'esclave, aussi la compensation introduite par le régulateur en agissant sur la vitesse du moteur esclave pour surmonter l'écart afin d'atteindre la position du moteur maître.

### III.11 Conclusion

Le régulateur PID appliqué aux moteurs électriques à courant continu était un régulateur qui pouvait offrir de bonnes performances, que ce soit au niveau de la précision, du temps de réaction, ou la stabilité. Diverses méthodes permettent d'optimiser les coefficients du PID.

Mais le régulateur idéal n'existe pas, chaque optimisation d'un paramètre (précision, ...etc.) se fait aux dépens d'un autre. Il convient alors de trouver le meilleur compromis en fonction des exigences initiales du cahier des charges.

Divers éléments, tels que le choix de la méthode d'implantation du PID, peuvent aussi avoir des conséquences sur les performances. L'essentiel est alors de s'éloigner le moins possible du modèle théorique pour qu'il reste valable.

Dans notre réalisation, ce régulateur nous a permis de satisfaire certains objectifs comme la précision, la robustesse de point vu perturbations.

# Chapitre IV

## IV.1 Introduction

Dans ce chapitre, nous allons présenter et analyser quelques structures intelligentes basées sur les concepts tels que la logique floue, les réseaux de neurones et les réseaux neuro-flous, dans le but de lever une contrainte en contrôle relativement à la nécessité de disposer d'un modèle le plus fidèle possible, sachant que les erreurs de modélisation et les imperfections des modèles, contribuent à dégrader notablement les performances des lois de commandes conventionnelles.

Dans la première partie de ce chapitre, nous discuterons des algorithmes dits "réseaux de neurones". A travers un historique rapide des origines de ces algorithmes, nous présenterons les propriétés principales de ces réseaux de neurones, puis plus en détail les algorithmes dont les caractéristiques peuvent permettre de traiter le problème posé.

Dans la deuxième partie, nous étudierons en détail les principes de la logique floue et ses applications dans le domaine de réglage puis on décrit différentes solutions pour un organe de commande basé sur les principes flou.

Ensuite, dans la troisième partie de ce chapitre nous présenterons l'interaction entre la logique floue et les réseaux de neurones pour former les réseaux neuro-flous. L'esprit de ces réseaux est d'intégrer a priori, une connaissance même imparfaite, mais permettant un apprentissage plus rapide qu'avec des poids initiaux aléatoires. On terminera par son utilisation en simulation.

## IV.2 Notions sur les réseaux de neurones artificiels

### IV.2.1 Introduction

Un réseau de neurones est un modèle de calcul dont la conception est très schématiquement inspirée du fonctionnement de neurones biologiques (humains ou animal). Les réseaux de neurones sont généralement optimisés par des méthodes d'apprentissage de type statistique, si bien qu'ils sont classés d'une part dans la famille des applications statistiques, et d'autre part dans la famille des méthodes de l'intelligence artificielle qu'ils enrichissent en permettant de prendre des décisions s'appuyant davantage sur la perception que sur le raisonnement logique ferme.

### IV.2.2 Le neurone

La cellule de base composant le cerveau est appelée NEURONE ; Le neurone est une cellule composée d'un corps cellulaire et d'un noyau. Le corps cellulaire se ramifie pour former ce que l'on nomme les dendrites. Celles-ci sont parfois si nombreuses que l'on parle alors de chevelure dendritique ou d'arborisation dendritique. C'est par les dendrites que l'information est acheminée de l'extérieur vers le soma, corps du neurone.

L'information traitée par le neurone chemine ensuite le long de l'axone (unique) pour être transmise aux autres neurones. La transmission entre deux neurones n'est pas directe. La jonction entre deux neurones est appelée la synapse.

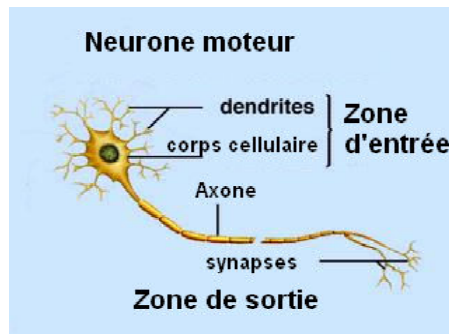


Figure 1: neurone biologique

La transmission de l'information se fait suivant deux modes :

- 1- à travers les axones ; par changement de potentiel ionique
- 2- à travers les synapses (transmission chimique)

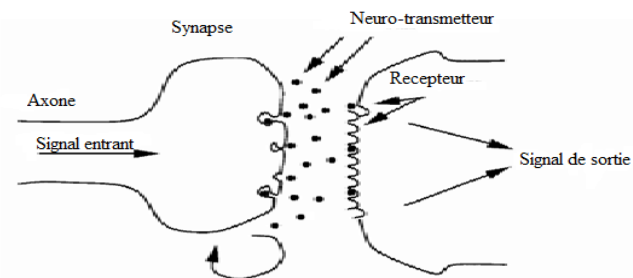


Figure 2: Synapse

### IV.2.3 Fonctionnement

On distingue deux phases. La première est habituellement le calcul de la somme pondérée des entrées selon l'expression suivante :

$$a = \sum W_i X_i - \theta \dots\dots\dots (IV.1)$$

A partir de cette valeur, une fonction de transfert calcule la valeur de l'état du neurone. C'est cette valeur qui sera transmise aux neurones aval. Il existe de nombreuses formes possibles pour la fonction de transfert. Les plus courantes sont présentées sur la figure suivante. On remarquera qu'à la différence des neurones biologiques dont l'état est binaire, la plupart des fonctions de transfert sont continues, offrant une infinité de valeurs possibles comprise dans l'intervalle [0, +1] ou [-1, +1].

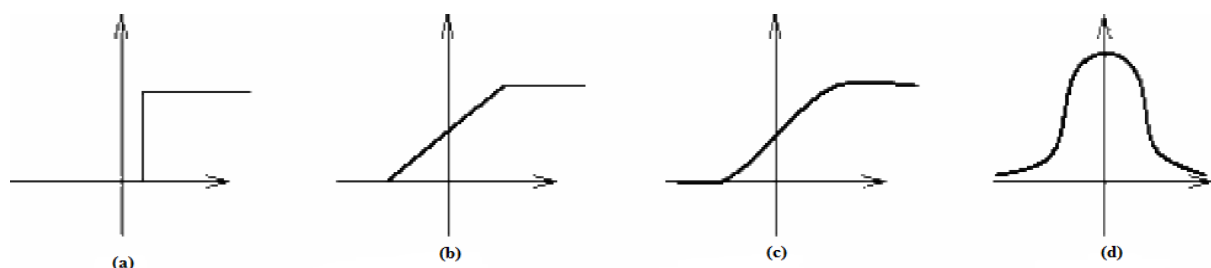


Figure 3: Différents types de fonctions de transfert pour le neurone artificiel

Cette figure représente les différents types de fonctions de transfert pour le neurone artificiel tel que :

- (a) : seuil (fonction de Heavyside)
- (b) : linéaire par morceaux
- (c) : sigmoïde  $g(x) = (1 - e^{-\beta x}) - 1$
- (d) : gaussienne

Nous constatons que les équations décrivant le comportement des neurones artificiels n'introduisent pas la notion de temps. En effet, et c'est le cas pour la plupart des modèles actuels de réseaux de neurones, nous avons affaire à des modèles à temps discret, synchrones, dont le comportement des composants ne varie pas dans le temps.

### IV.3 Notions sur l'apprentissage

L'apprentissage est vraisemblablement la propriété la plus intéressante des réseaux neuronaux; c'est la phase du développement d'un réseau de neurones durant laquelle le comportement du réseau est modifié jusqu'à l'obtention du comportement désiré. L'apprentissage neuronal fait appel à des exemples de comportement.

Dans le cas des réseaux de neurones artificiels, on ajoute souvent à la description du modèle l'algorithme d'apprentissage. Le modèle sans apprentissage présente en effet peu d'intérêt. Dans la majorité des algorithmes actuels, les variables modifiées pendant l'apprentissage sont les poids des connexions. L'apprentissage est la modification des poids du réseau dans l'optique d'accorder la réponse du réseau aux exemples et à l'expérience. Il est souvent impossible de décider a priori des valeurs des poids des connexions d'un réseau pour une application donnée. A l'issue de l'apprentissage, les poids sont fixés : c'est alors la phase d'utilisation.

Certains modèles de réseaux sont improprement dénommés à apprentissage permanent. Dans ce cas il est vrai que l'apprentissage ne s'arrête jamais, cependant on peut toujours distinguer une phase d'apprentissage (en fait de remise à jour du comportement) et une phase d'utilisation. Cette technique permet de conserver au réseau un comportement adapté malgré les fluctuations dans les données d'entrées. Au niveau des algorithmes d'apprentissage, il a été défini deux grandes classes selon que l'apprentissage est dit supervisé ou non supervisé.

Cette distinction repose sur la forme des exemples d'apprentissage. Dans le cas de l'apprentissage supervisé, les exemples sont des couples (Entrée, Sortie associée) alors que l'on ne dispose que des valeurs (Entrée) pour l'apprentissage non supervisé. Remarquons cependant que les modèles à apprentissage non supervisé nécessitent avant la phase d'utilisation une étape de labellisation effectuée par l'opérateur, qui n'est pas autre chose qu'une part de supervision.

#### IV.3.1 Types d'apprentissage

On appelle apprentissage des réseaux de neurones artificiels la procédure itérative qui consiste à estimer les paramètres des neurones du réseau afin que celui-ci remplisse au mieux la tâche qui lui est affectée.

L'apprentissage est la caractéristique principale des réseaux de neurones et il peut faire de différentes règles et manières.

#### IV.3.1.1 Apprentissage non supervisé

L'apprentissage est qualifié de non supervisé lorsque, seules les valeurs d'entrée sont disponibles. Il repose sur un critère interne de conformité du comportement du réseau par rapport à des spécifications générales et non sur des observations externes. Dans ce cas, l'apprentissage est basé sur des probabilités. Le réseau va se modifier en fonction des régularités statistiques de l'entrée. L'algorithme d'apprentissage exploite pour ce faire une mesure prédéterminée de la qualité de la représentation de la connaissance afin d'ajuster les paramètres du réseau. Ce type d'apprentissage est représenté schématiquement en figure 4.

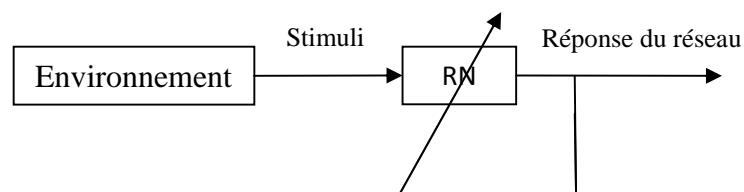


Figure 4: Apprentissage non supervisé

#### IV.3.1.2 Apprentissage supervisé

L'apprentissage supervisé consiste à présenter une série d'exemples sur ses entrées, ainsi le réseau calcule sur ses sorties un vecteur d'état qui est comparé au vecteur des états désirés. Une erreur est calculée entre le vecteur des sorties désirées et le vecteur des sorties calculées par le réseau. L'algorithme d'apprentissage adapte les poids du réseau de façon à corriger son comportement comme le montre la figure 5.

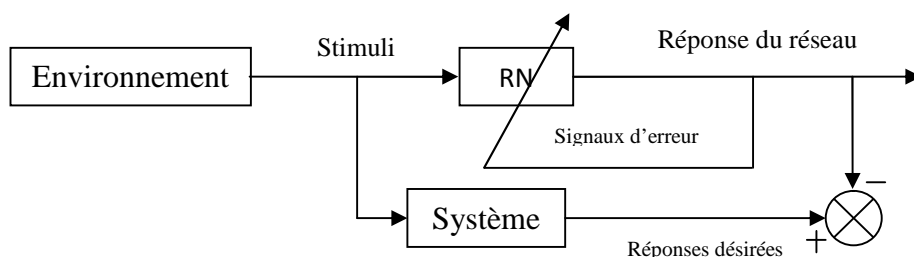


Figure 5 : Apprentissage supervisé

##### IV.3.1.2.1 La retro-propagation du gradient

La retro-propagation du gradient de l'erreur a été créée en généralisant les règles d'apprentissage de Widrow-Hoff, aux réseaux multicouches à fonction de transfert non linéaire.

La rétro-propagation du gradient de l'erreur est utilisé pour ajuster les poids et les biais du réseau afin de minimiser l'erreur quadratique entre la sortie du réseau et la sortie réelle. A chaque couple entrée/sortie, une erreur est calculée, le gradient, ou pente de l'erreur est

déterminé, ensuite les poids et les biais sont modifiés en ligne sur le réseau. On réitère ces calculs jusqu'à l'obtention du critère d'arrêt.

Soit une réponse spécifique désirée  $D_p$  à chaque vecteur d'entrée  $X_p$ . La mise à jours des coefficients synaptiques  $W$  s'effectuent progressivement jusqu'à ce que l'erreur entre les sorties du réseau et les résultats désirés soit minimisée. Pour ce faire, considérons le problème de minimisation de la fonction de coût défini n exemples  $(X_p; D_p)$  et  $Y_p$  la sortie du réseau, par l'erreur qui peut être quadratique de forme :

$$E(W) = \sum_p (Y_p - D_p)^2 \dots\dots\dots (IV.2)$$

L'apprentissage revient à déterminer  $W = ArgMin(E)$

En général, on utilise des méthodes de gradient partiel, total ou stochastique :

$$W(t + 1) = W(t) - \lambda grad_w(E) \dots\dots\dots (IV.3)$$

- Gradient partiel :  $W(t + 1) = W(t) + \lambda (D_p - f(WX_p)) f'(WX_p) X_p$
- Gradient total :  $W(t + 1) = W(t) - \lambda \cdot grad_w(E(t))$
- Gradient stochastique :  $W(t + 1) = W(t) - \lambda \cdot grad_w(E_p(t))$

Où  $E_p = (Y_p - D_p)^2$ , erreur calculée sur un seul exemple que l'on tire au hasard à chaque t.  
 f : fonction d'activation.

$\lambda(t)$  : Pas du gradient (fixe, décroissant, ou adaptatif)

Pour calculer  $\frac{dE_p}{dW_p}$ , on exploite la retro-propagation du gradient, qui est une méthode astucieuse de calcul du gradient de la fonction coût en utilisant la dérivation composée pour rétro-propager l'erreur, comme le montre la figure suivante :

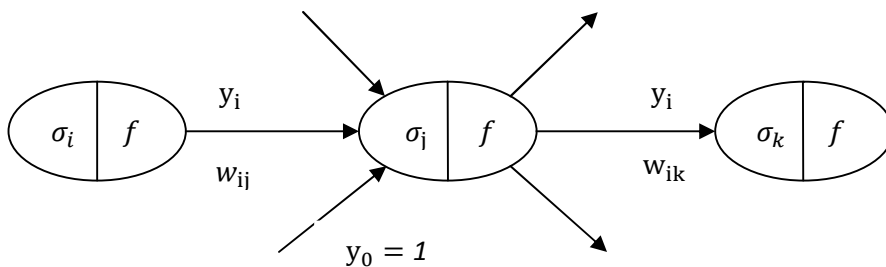


Figure 6 : retro-propagation du gradient

$$\frac{dE_p}{dW_{ij}} = \frac{dE_p}{d\sigma_j} \frac{d\sigma_j}{dW_{ij}} = \left( \frac{dE_p}{d\sigma_j} \right) y_i \dots\dots\dots (IV.4)$$

On pose  $\delta_j = \frac{dE_p}{d\sigma_j}$ , d'où  $W_{ij}(t + 1) = W_{ij}(t) - \lambda y_i \delta_j \dots\dots\dots (IV.5)$

Mais  $\delta_j = \frac{dE_p}{d\sigma_j} = \sum_k \left( \frac{dE_p}{d\sigma_k} \right) \left( \frac{d\sigma_k}{d\sigma_j} \right) = \sum_p \delta_k \left( \frac{dE_p}{d\sigma_k} \right) = \sum_p \delta_k W_{jk} \left( \frac{dE_p}{d\sigma_k} \right) \dots\dots\dots (IV.6)$

D'où  $\delta_j = (\sum_p \delta_k W_{jk}) f'(\sigma_j)$  si le neurone j est caché

Et  $\delta_j = \left(\frac{dE_p}{d\sigma_j}\right) = \left(\frac{dE_p}{dy_j}\right)\left(\frac{dy_j}{d\sigma_j}\right)$ ..... (IV.7)

D'où  $\delta_j = 2(y_j - D_j)f'(\sigma_j)$  si neurone j en sortie, cependant  $\delta_j$  se calcule de proche en proche par retro-propagation de l'erreur.

**IV.4 Logique floue**

La notion de logique floue permet d'étendre la notion de logique classique, associée aux variables booléennes ne prenant que deux valeurs 0 et 1. Il est alors possible d'associer à des variables des coefficients d'appartenance à des sous-ensembles flous prenant des valeurs dans l'intervalle [0,1] et quantifiant l'incertitude sur la variable. Un événement certain pour la variable se traduira par un coefficient d'appartenance au sous-ensemble flou, c'est-à-dire à la propriété, égal à 1 alors que la valeur sera inférieure à 1 en présence d'incertitudes. Il vient alors qu'une valeur nulle pour un coefficient d'appartenance indique que la possibilité d'appartenance au sous-ensemble sélectionné de la variable représentant la grandeur concernée est complètement rejetée.

**IV.4.1 Notion de sous-ensemble flou**

Dans la théorie classique des ensembles, un sous-ensemble A et B est défini par une fonction d'appartenance  $\mu_A(x)$  qui caractérise tout élément x appartenant à A. cette fonction prend la valeur 1 si x appartient à A et la valeur 0 dans le cas contraire :

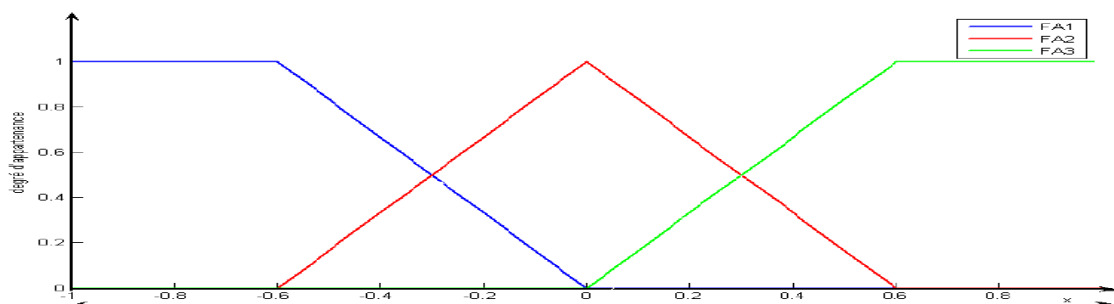
$$\mu_A(x) = \begin{cases} 1 & \text{si } x \in A \\ 0 & \text{si } x \notin A \end{cases}$$

En logique floue, un sous-ensemble flou A de B est défini par une fonction d'appartenance  $\mu_A(x)$  pouvant prendre différentes valeurs comprises entre 0 et 1, selon le degré d'appartenance de l'élément x au sous-ensemble A

$$\mu_A(x) \in [0, 1]$$

**IV.4.2 Univers de discours**

L'univers de discours représente l'ensemble de référence ou le domaine de variation de la variable linguistique, ou domaine de fonctionnement du processus dans le cas de réglage, comme illustré sur la figure 7 dont plusieurs sous-ensembles sont décrits par des fonctions d'appartenance (FA) triangulaires et trapézoïdales.



**Figure 7 :** Univers de discours normalisé uniforme de 3 ensembles

Les Valeur et variable linguistique représente un phénomène ou une grandeur, par des termes linguistiques : « quelque, beaucoup, souvent », « négative, environ zéro et positive » ce sont les valeurs linguistiques.

#### IV.4.3 règles d'inférence

On appelle règles d'inférence l'ensemble des différentes règles reliant les variables floues d'entrée d'un système aux variables floues de sortie de ce système à l'aide des différentes operateurs. Ces règles se présentent sous la forme :

Si condition 1 ET/OU condition 2(ET/OU) alors action sur les sorties OU

Si condition 3 ET/OU condition 4(ET/OU) alors action sur les sorties OU

⋮

⋮

Si condition n ET/OU condition n+1(ET/OU) alors action sur les sorties OU

Ou par une matrice d'inférence sous forme d'un tableau donné ci-dessous :

U	θ		
	N	EZ	P
N	N	N	EZ
EZ	N	EZ	P
P	EZ	P	P

#### IV.4.4 représentations des fonctions d'appartenance

On représente les variables linguistiques par leurs fonctions d'appartenances. Donc à chaque sous-ensemble flou  $A_i$  est associé une fonction d'appartenance  $\mu_{A_i}(x)$  où  $x$  est la variable linguistique. Tel que, à chaque  $x$  est associé une valeur précise de  $\mu_{A_i}(x)$ , qui désigne, le degré d'appartenance de  $x$  à  $A_i$ .

La fonction d'appartenance peut être représentée par plusieurs formes : triangulaire, trapézoïdale, sigmoïdale et gaussienne. On peut définir d'autres formes de fonctions d'appartenance, mais dans le réglage par logique floue, les formes déjà citées sont largement suffisantes pour délimiter les ensembles flous.

#### IV.4.5 Principe de la logique floue

La logique floue n'est autre que l'extension de la logique binaire, à l'aide d'une série de notions fondamentales développées qui ont permis de justifier et de montrer certaines notions de base. Les éléments indispensables à la compréhension du réglage par la logique floue sont:

- Les variables floues.
- Les règles d'inférence.
- Les opérateurs flous.

#### IV.4.5.1 Opérateurs flous

Les règles d'inférences font appel aux opérateurs ET, OU et NON qui s'appliquent aux variables floues. Dans le cas de la logique binaire, ces opérateurs sont définis de façon simple et univoque. En revanche, la définition de ces opérateurs, dans le cas de la logique floue n'est plus univoque et on utilise le plus souvent les relations présentées dans le tableau suivant :

Operateur	Operateur sur le degré de vérité des variables
ET	Minimum
	Produit
OU	Maximum
	Valeur moyenne
NON	Complément à 1

Les opérations minimum et maximum présentent l'avantage de la simplicité lors du calcul, par contre, elles privilégient l'une des deux variables. Quant aux opérations, produit et valeur moyenne, elles sont les plus complexes à calculer, mais produisent un résultat qui tient compte des valeurs des deux variables.

#### IV.4.5.2 Structure interne d'un système d'inférence flou (SIF)

Un système flou est un système à base de connaissances particulières, dont l'architecture de base, illustrée par la figure 8, se compose essentiellement de quatre modules.

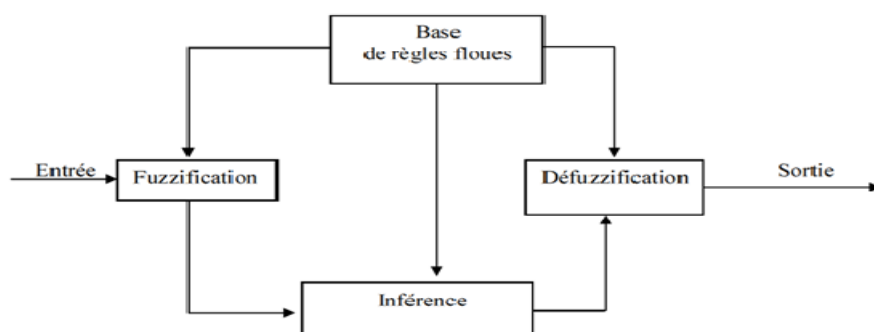


Figure 8 : Architecture de base d'un système flou

##### IV.4.5.2.1 Fuzzification

C'est le mécanisme réalisant l'interface " Numérique - Linguistique" qui permet de transformer une grandeur générée par un capteur à l'entrée en partie floue définie sur un espace de représentation lié à l'entrée. Du choix de cet espace de représentation (numérique ou linguistique) dépend la nature de la partie floue E résultante de la partie fuzzification.

Cette dernière peut être un sous ensemble numérique (élément de  $F(x)$ ) ou un sous ensemble flou symbolique (élément  $F(L(x))$ ). Dont on distingue alors deux types de fuzzification respectivement appelés fuzzification numérique et fuzzification symbolique.

#### IV.4.5.2 Défuzzification

C'est une interface "Linguistique - Numérique" qui transforme la partie floue de l'inférence en valeurs numériques directement exploitables par le processus. Le résultat de l'inférence peut être soit une partie floue numérique, soit une partie floue symbolique. Afin de les transformer en une valeur numérique, on aura besoin de deux types de défuzzification: une défuzzification numérique et une défuzzification symbolique.

### IV.5 Systèmes neuro-flou

Les réseaux de neurones (RN) multicouches constituent des approximateurs universels grâce à leurs capacités d'apprentissage, mais leurs structures et leurs paramètres n'ont pas toujours d'interprétation physique. De plus, la connaissance humaine ne peut pas être exploitée pour les construire.

En revanche les systèmes d'inférence floue possèdent deux points forts par rapport aux réseaux de neurones. D'une part, ils sont généralement construits à partir de la connaissance humaine, d'autre part, ils ont une capacité descriptive élevée due à l'utilisation de variables linguistiques. L'approche traditionnelle pour la conception des systèmes d'inférences floues est basée sur les connaissances acquises par des opérateurs experts. Cette approche a prouvé son efficacité dans plusieurs applications. Il se peut cependant que les opérateurs ne puissent pas traduire leurs connaissances et expériences sous forme de contrôleurs à logique floue. Il arrive parfois que le domaine d'expertise ne soit pas disponible. Il est donc évident de construire des systèmes hybrides qui combinent les concepts des systèmes d'inférence flous et des réseaux de neurones, ce sont des réseaux neuro-flous, dont l'apprentissage permet d'ajuster les paramètres et de s'adapter aux changements de conditions. Les inférences sont ensuite extraites après convergence. Le schéma général d'une structure neuro-flou est illustré par la figure suivante :

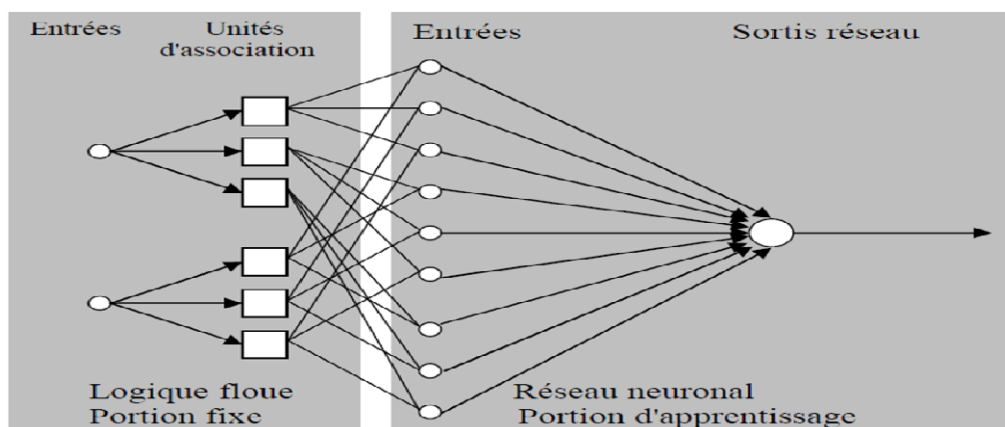


Figure 9 : structure générale d'un réseau neuro flou

### IV.5.1 Méthodes des combinaisons neuro-floues

Il existe quatre grandes catégories de combinaisons des réseaux de neurones avec la logique floue.

#### a) Réseau flou neuronal

Dans ces réseaux, les techniques floues employées pour augmenter les possibilités du processus d'apprentissage et d'exécution des réseaux de neurones.

#### b) Système neuronal/flou simultanément

Les réseaux de neurones et le système flou fonctionnent ensemble sur la même tâche, mais sans s'influencer, c'est-à-dire ni l'un ni l'autre n'est employé pour déterminer les paramètres de l'autre. Habituellement le réseau neuronal traite les entrées, ou post-traite les sorties des systèmes flou.

#### c) Méthodes neuro-floues coopératifs

Le réseau de neurone est employé pour déterminer les paramètres (les règles et les ensembles flous) d'un système flou. Après la phase d'apprentissage, le système flou fonctionne sans le réseau de neurone .c'est une forme simple des systèmes neuro-flous.

#### d) Modèles neuro-floues hybrides

Les approches neuro-floues modernes sont de cette forme. Un réseau neuronal et un système flou sont combinés dans une architecture homogène. Le système peut être interprété comme un réseau neuronal spécial avec des paramètres flous et comme un système flou mis en application sous une forme distribuée parallèle.

### IV.5.2 Les types d'implémentation des réseaux neuro-floues

Dans la littérature il ya une panoplie de systèmes neuro-flou comme : les systèmes neuro-floues coopératives, concurrents, les systèmes dit hybrides ou neuro flou adaptatif, le modèle NEFCLASS, ANFIS, FALCON et GARIC etc.

Selon l'interaction entre la logique floue et les réseaux de neurones, et compte tenu de nos objectifs, nous avons retenu un type de système d'inférence neuro-flou adaptatifs (ANFIS).

#### IV.5.2.1 Systèmes d'inférence neuro-flou adaptatifs (ANFIS)

Les systèmes d'inférence neuro-flou adaptatifs (ANFIS), sont des systèmes hybrides utilisant l'inférence floue de Takagi Sugeno. La structure ANFIS est constituée de cinq couches comme le montre la figure 10. La première couche cachée fuzzifie les variables d'entrées et les operateurs type T-norme calcule la partie prémisse des règles dans la deuxième couche cachée. La troisième couche cachée normalise les poids des règles suivit par la quatrième couche cachée où les paramètres des parties conclusions des règles sont déterminés. La cinquième couche calcule la somme de tous les signaux venant de la quatrième couche.

Sont entrainement est accomplit par l’algorithme du gradient descente (retro propagation) pour l’optimisation des paramètres des parties prémises, et l’algorithme des moindres carré pour la résolution des paramètres des parties conclusion. Afin de réduire l’erreur.

Pour simplifier la compréhension, nous considérons un système à deux entrées  $x_1$  et  $x_2$  et une sortie  $y$ , modélisé par un système flou de type TSK, composé des deux règles suivantes :

Si  $x_1$  est  $A_1$  et  $x_2$  est  $B_1$  alors  $y_1 = f_1(x_1, x_2) = a_1x_1 + b_1x_2 + c_1 \dots \dots \dots$  (IV.8)

Si  $x_1$  est  $A_2$  et  $x_2$  est  $B_2$  alors  $y_2 = f_2(x_1, x_2) = a_2x_1 + b_2x_2 + c_2 \dots \dots \dots$  (IV.9)

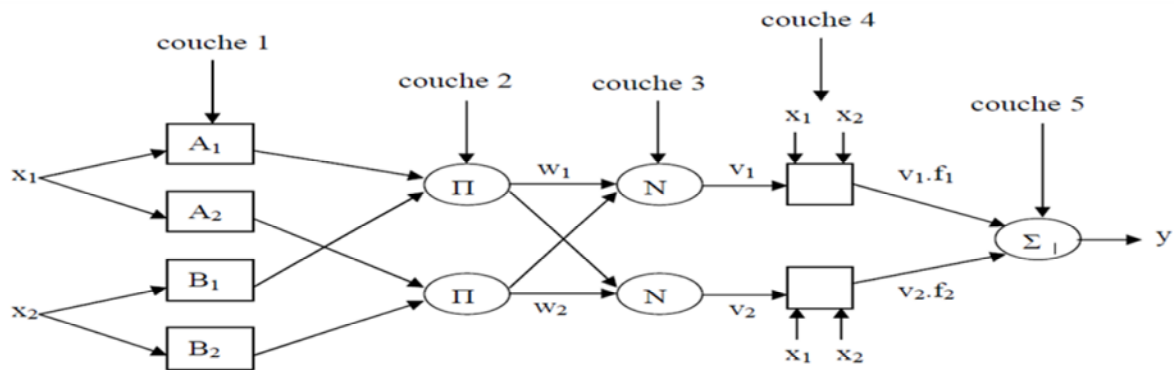


Figure 10 : Architecture ANFIS

Le réseau adaptatif ANFIS est un réseau multi couches dont les connexions ne sont pas pondérées. Les nœuds sont de deux types différents selon leurs fonctionnalités : les nœuds carrés (adaptatifs) contiennent des paramètres, et les nœuds circulaires (fixes) n’ont pas de paramètres. Toutes fois, chaque nœud (carré ou circulaire) applique une fonction sur ses signaux d’entrées. La sortie  $O_i^k$  du nœud  $i$  de la couche  $k$  (appelé nœud  $(i, k)$ ) dépend des signaux provenant de la couche  $(k-1)$ , c’est-à-dire :

$$O_i^k = f(O_1^{k-1}, \dots, O_{n_{k-1}}^{k-1}, a, b, c, \dots) \dots \dots \dots$$
 (IV.10)

Où  $n_{k-1}$  est le nombre de nœuds dans la couche  $k-1$ , et  $a, b, c \dots$  sont les paramètres du nœud  $(i, k)$ . Pour un nœud circulaire ces paramètres n’existent pas.

Dans le réseau de neurones de la figure précédente, les nœuds d’une même couche ont des fonctions issues d’une même famille que nous explicitons ci-dessous.

- **Couche 1** : Génération du degré d’appartenance :

Chaque nœud de cette couche possède des paramètres ajustables. La fonction du nœud est identique à la fonction d’appartenance à un sous-ensemble flou de l’univers de discours des entrées.

$$O_i^1 = f_i^1(x) = \mu_{A_i}(x) \dots \dots \dots$$
 (IV.11)

Où  $x$  est l’entrée du nœud  $i$ , et  $A_i$  le terme linguistique associé à sa fonction  $\mu_{A_i}$ . en d’autre termes,  $O_i^1$  est le degré d’appartenance de  $x$  à  $A_i$ , définie par une fonction d’appartenance  $\mu_{A_i}(x)$ .

- **Couche 2** : Génération du poids de la règle i :

Chaque nœud circulaire appelé  $\Pi$  dans cette couche corespnd à une regle floue Sugeno. Il recoit les sorties des nœuds de fuzzification et calcule son activation. La conjonction des antécédents est réalisée avec l'opérateur produit.

$$w_i = \mu_{A_i}(x_1) \cdot \mu_{B_{i-2}}(x_2), \quad (i=1 \dots 2) \dots \dots \dots (IV.12)$$

- **couche 3** : Normalisation des poids de la règle i:

Chaque nœud de cette couche est un nœud circulaire appelé N. il calcule le degré d'activation normalisé d'une règle floue donnée. La valeur obtenue représente la contribution de la règle floue au résultat final.

$$v_i = \frac{w_i}{w_1+w_2} \dots \dots \dots (IV.13)$$

- **Couche 4** : Calcul de la sortie des règles :

Chaque nœud de cette couche est un nœud carré avec une fonction réalisant le calcul suivant :

$$O_i^4 = v_i f_i = v_i (a_1 x_1 + b_i x_2 + c_i), \quad (i=1 \dots 2) \dots \dots \dots (IV.14)$$

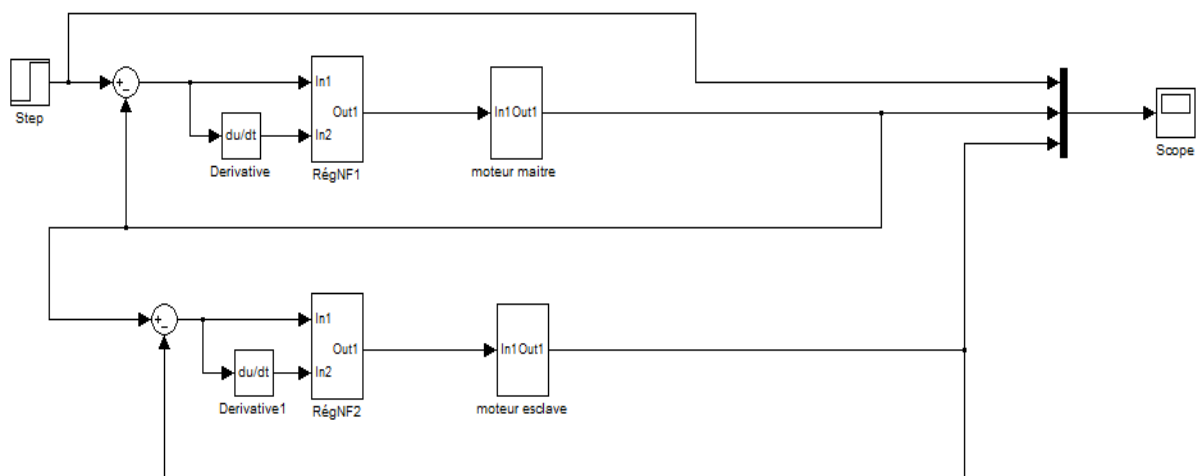
Où  $v_i$  est l sortie de la couche 3, et  $\{a_i, b_i, c_i\}$  est l'ensemble des parametres ajustables de sortie de la règle i.

- **Couche 5** : Faire la somme de toutes les entrées à partir de la couche 4

Comprend un seul nœud qui fournit la sortie de ANFIS en calculant la somme des sorties de tous les nœuds de défuzzification, c'est-à-dire,

$$O_1^5 = y = \sum_i v_i f_i \dots \dots \dots (IV.15)$$

**IV.6 Modèles simulés sous MATLAB/SIMULINK**



**Figure 11** : Modèle de simulation du système Maitre/Esclave

En parallèle de la réalisation physique du banc de test, un simulateur par blocs, basés sur les modèles physiques des composants, a été développé dans l'environnement MATLAB Simulink. Son rôle est de permettre une meilleure compréhension des comportements du système haptique grâce à un accès aisé à l'ensemble des variables et à la possibilité d'analyser l'influence de la modification des paramètres hardware du système.

Cette simulation permet également de prendre en compte les perturbations appliquée au système (moteur). Ce phénomène est difficile à analyser par l'étude linéaire présentée dans le premier chapitre.

Le modèle utilisé pour les différentes simulations est maintenu. Seule l'interaction externe est différente, avec l'opérateur pour le maître et l'environnement et autres perturbations pour l'esclave.

L'objectif du bloc opérateur est de fournir un couple d'entrée dans le système, pour générer le mouvement et la réaction de contact si nécessaire.

## IV.7 Simulations et résultats expérimentaux

### IV.7.1 Cas théoriques

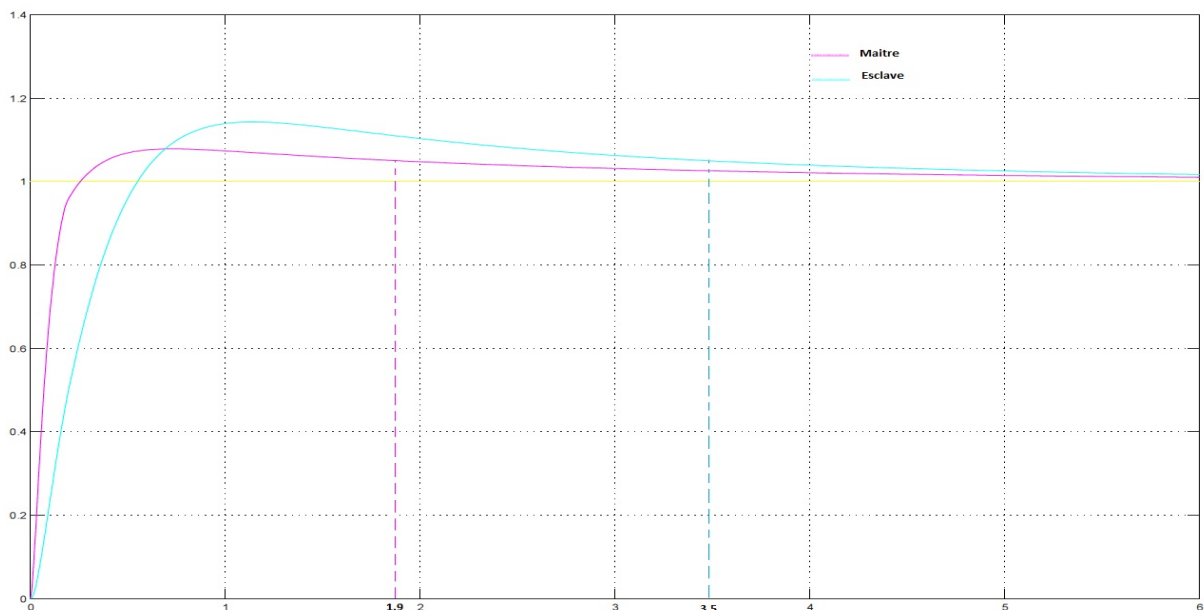


Figure 12 : résultats (théoriques) de simulation des deux positions Maître/Esclave

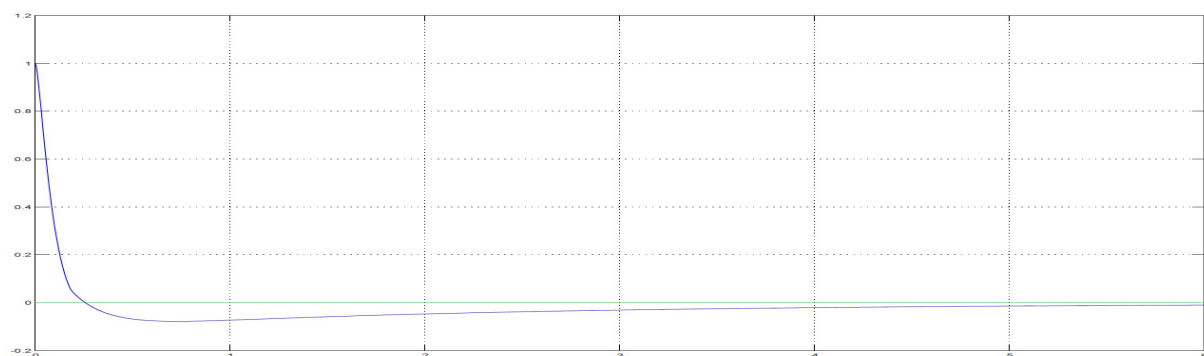


Figure 13 : erreur de position

IV.7.2 Cas pratiques

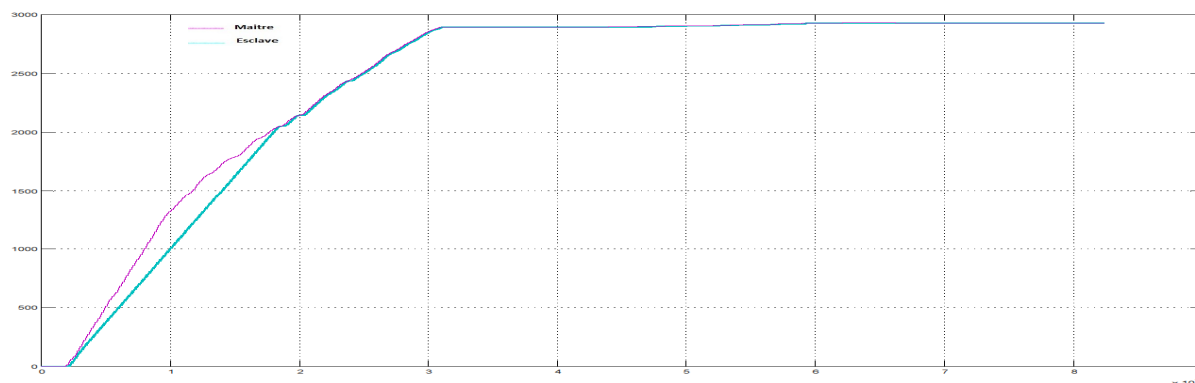


Figure 14: Résultats (pratiques) de simulation des deux positions Maître/Esclave libre

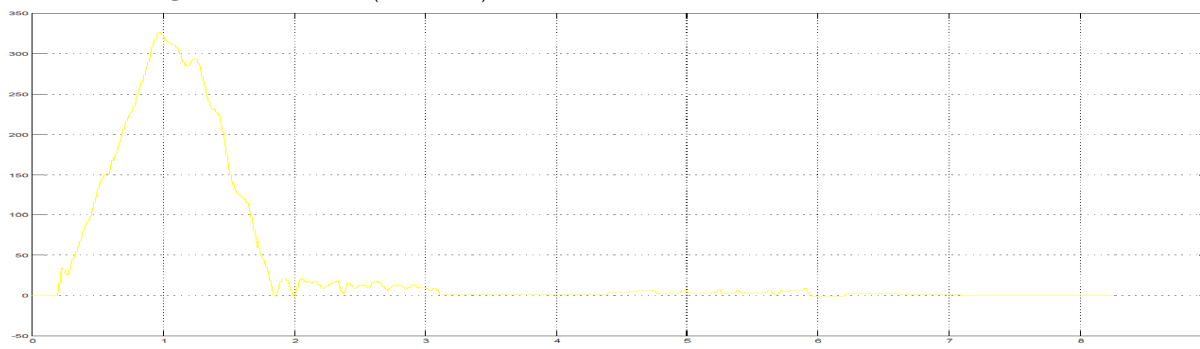


Figure 15 : erreur de position

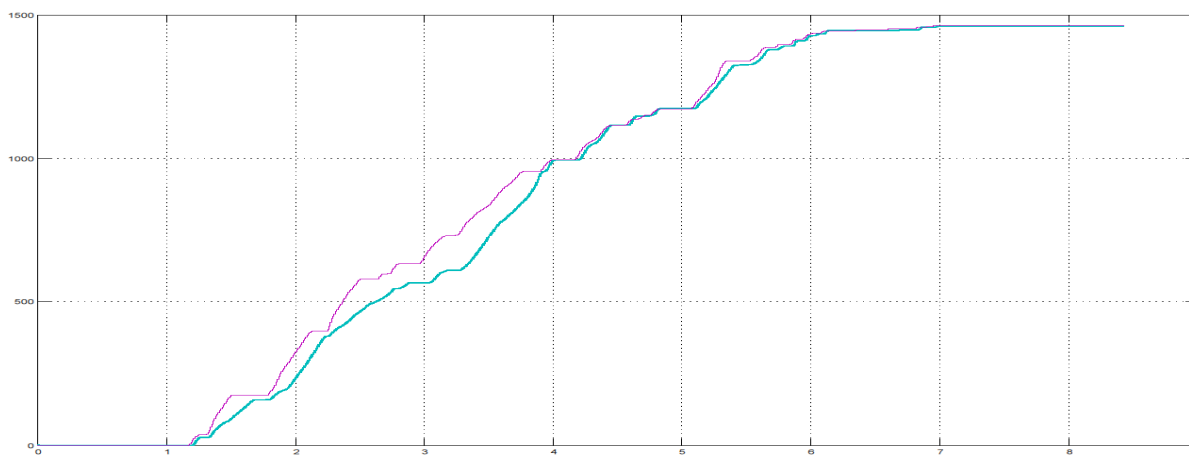


Figure 16 : Résultats (pratiques) de simulation des deux positions (Maître/Esclave) avec perturbations

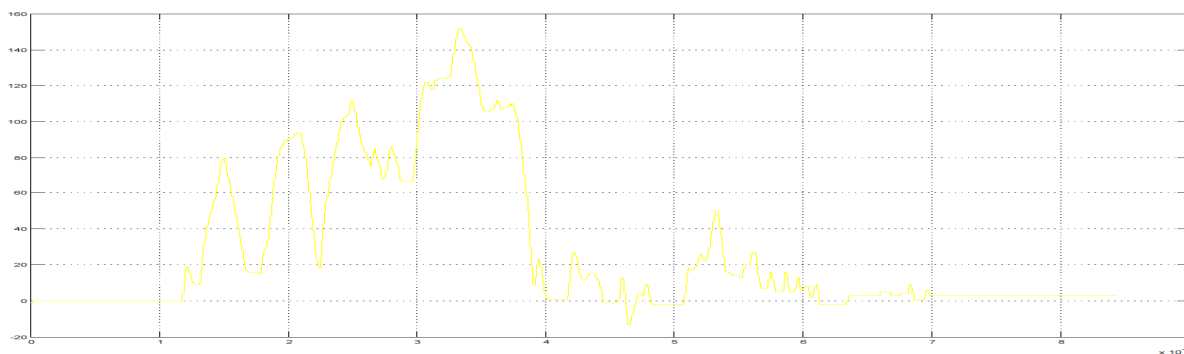


Figure 17: Erreur de position (système perturbé)

### IV.7.3 Interprétation des résultats

#### IV.7.3.1 Cas théorique

Les courbes 12, 13 décrivent les caractéristiques du système muni d'un régulateur neuro flou.

Le graphe d'évolution des positions (maitre/esclave), nous a permis de voir l'influence de ce régulateur, de constater ses caractéristiques et ses performances ainsi la méthode de réglage des paramètres. Pour une meilleure utilisation dans le cas pratique.

#### IV.7.3.2 Cas pratique

Les figures 14, 15 sont les résultats obtenus lors du test pratique. Ce test consiste à la mise en mouvement du système par l'opérateur en utilisant le régulateur neuro flou, avec des paramètres bien adaptés de manière à avoir un meilleur fonctionnement en minimisant l'écart entre les deux positions, en maintenant sa stabilité.

Cette expérience a été effectuée avec la présence d'un délai de  $3 \cdot 10^{-3}$  s dans l'échange de communication entre les deux sites.

Le réglage du régulateur est caractérisé par une grande robustesse vis-à-vis les changements paramétriques que peut subir le système (exemple : changement de charge), ceci est dû à deux raisons principales :

- Généralisation de réseau de neurone
- Raisonnement approximatif de régulateur flou

Nous avons procédé à l'implémentation de la commande testée théoriquement à notre réalisation pratique. Dont les résultats obtenus sont concluants et satisfaisants en termes de stabilité et de poursuite de trajectoire constatées sur la figure 14 et en écart minimal en phase finale de la manipulation, comme on peut le remarquer sur la figure 15. Cela est effectué avec un temps de réaction petit du moteur esclave. Ces derniers sont des conséquences introduites par la présence du régulateur neuro flou qui intervient pour éliminer les incertitudes et les perturbations internes et externes.

### IV.8 Conclusion

Dans ce chapitre, nous avons présenté les concepts fondamentaux des réseaux de neurones ainsi aux systèmes neuro-flous issus de la combinaison de la logique floue et les réseaux de neurones. Nous avons d'abord introduit la structure interne d'un système d'inférence flou, nécessaire à la compréhension d'un système flou. Nous avons ensuite présenté une méthodologie complète de la construction de la relation entrée-sortie à partir de règles floues. Par la suite les modèles dynamiques des réseaux de neurones sont décrits d'une manière plus détaillée avec l'algorithme d'apprentissage rétro propagation.

Comme la logique floue est une théorie très puissante qui permet d'obtenir des conclusions et de générer de réponses à partir de l'information vague, ambiguë, incomplète ou imprécise, là où les modèles mathématiques du système sont inconnus ou difficiles à extraire. Ainsi, l'introduction de la logique floue dans un processus a pour but de lui ajouter une

dimension. « Intelligente ». Néanmoins, la phase d'acquisition de connaissances représente une tâche très délicate, surtout en absence d'un expert qui peut fournir son savoir-faire sur le problème à traiter. Pour cette raison, des recherches très poussées ont conduit au développement des approches hybrides systématiques et optimales, basées sur l'interaction entre la logique floue et les réseaux de neurones, pour la conception des contrôleurs flous adaptatifs. Parmi ces techniques, on trouve les systèmes neuro-flous qui sont de puissants concurrents dans le domaine de l'optimisation et de l'apprentissage.

# Conclusion générale

## Conclusion générale

Le travail de la recherche présenté concerne un des domaines les plus intéressants et d'actualité de la robotique, à savoir la téléopération et la résolution des problèmes, qui peuvent surgir lors d'une ou plusieurs défaillances lors d'accomplissement d'une tâche.

On a étudié une architecture de téléopération à deux canaux de communication. Cette architecture utilise les signaux de la position. Cette dernière est très souhaitable pour obtenir une bonne transparence avec une certaine instabilité du système dans les conditions idéales. Afin d'améliorer les performances de la commande à distance des systèmes, des solutions ont été envisagées dans le but de stabilisation du système, à savoir les conditions sur la passivité du système et l'élimination du retard de transmission.

Dans le troisième et le quatrième chapitre nous avons exposé les résultats de simulation effectuée sous l'environnement Matlab/Simulink, afin de tester les performances d'une commande aux moteurs réducteur avec encodeur de quadrature télé opéré.

Les différentes simulations effectuées en réglant les paramètres du régulateur PID ainsi le régulateur neuro flou, montrent que ces types de régulateurs permettent de compenser les effets des défauts sur les performances du système.

On a aussi appliqué des approches qui permettent la reconfiguration de la loi de commande en temps réel garantissant au mieux les performances du système. On pourra ainsi doter le système d'une plus grande autonomie pour de tels systèmes robotisés télé opérés.

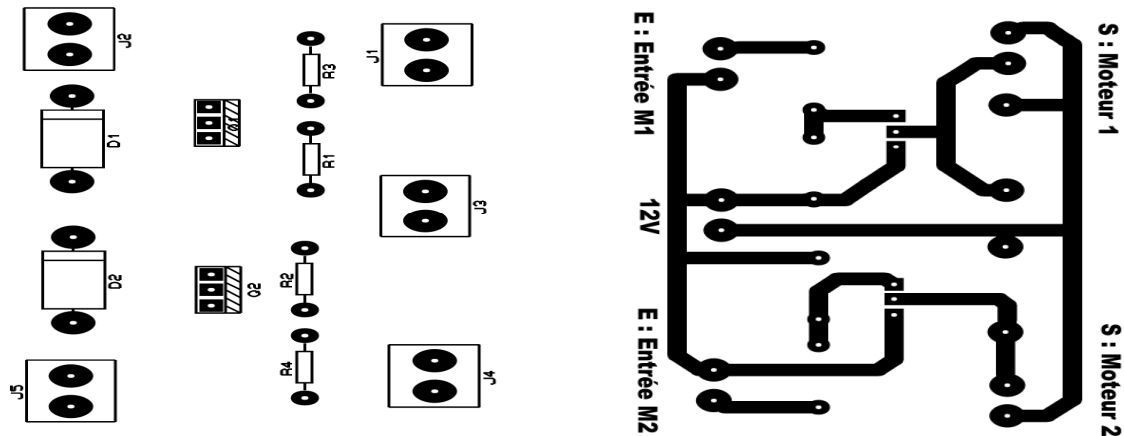
L'avantage principal de la méthode Position-Position est de ne pas utiliser de capteur d'effort, mais exclusivement les capteurs de position, largement plus répandus dans les systèmes mécatroniques.

On souhaite que ce mémoire servira comme plateforme pour des travaux de recherche futurs en robotique en général et dans la sécurité de fonctionnement des robots manipulateurs télé opérés en particulier.



- Thèse de l'Université Libre de Bruxelles, réalisé par [Pierre LETIER] “**Bras Exosquelette Haptique : Conception et Contrôle**”.
- Thèse de l'Ecole Nationale Polytechnique, Laboratoire de commande des processus (LCP), réalisé par [Nouar MESSAOUDI] et [Mounir KARA-ZAITRI] “**Commande Neuro floue d'une suspension active de véhicule**”.
- Thèse de L'ÉCOLE NATIONALE SUPÉRIEURE DE L'AÉRONAUTIQUE ET DE L'ESPACE [Yves BRIERE] “**TÉLÉOPÉRATION EN PRÉSENCE DE RETARD : LE CONCEPT DE TÉLÉOPÉRATION HYBRIDE DUALE**”.
- Thèse de l'École Doctorale Mathématiques, Sciences de l'Information et de l'Ingénieur [Laurent Barbé]“ **Téléopération avec retour d'efforts pour les interventions percutanées**”.
- Cours sur : **la Technologie et réglage des régulateurs PID.**
- Cours sur : **l'Étude Comparative des Régulateurs « PID et Flou »: « Autopilotage d'un moteur a courant continu».**
- Cours sur : **Fuzzy Neural Network-Based Model Reference Adaptive Inverse Control for Induction Machines.**
- Cours sur : **Fuzzy-PID Control Algorithm of the Helicopter Model Flight Attitude Control.**
- Livre [SIMON MONK] “**30 Arduino™ Projects for the Evil Genius™**”.
- Livre [BRIAN EVANS] “**Beginning Arduino Programming**” .

# Annexes



Architecture de la carte de commande (coté composants/coté cuivre)

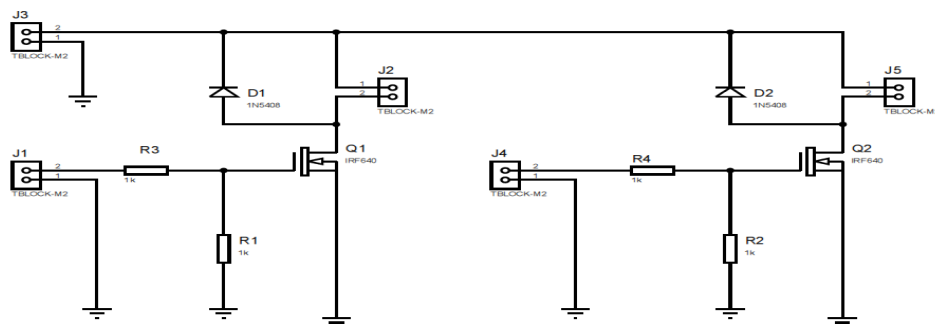


Schéma électrique de la carte de commande

## Module de commande de moteur "MD25"

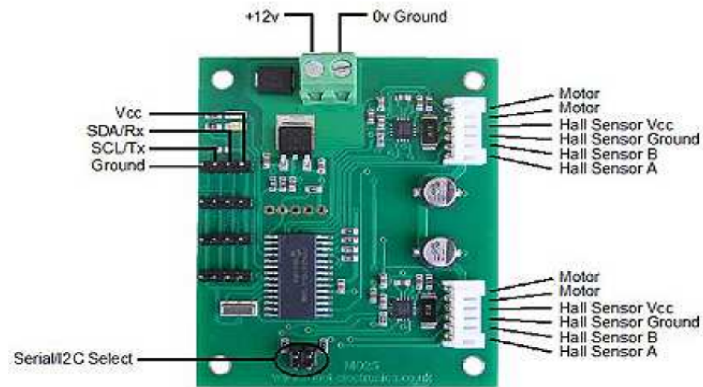
Ce module est destiné à être commandé par notre microcontrôleur au moyen d'une liaison série (niveau logique 0 - 5 V) ou via un bus I2C™, cette platine électronique intègre un double pont en "H" permettant le pilotage indépendant de 2 moteurs à courant continu avec encodeur (consommation: 2,5 A max. par moteur).

Des ordres simples nous permettent de définir le sens de rotation ainsi que la vitesse des moteurs, de lire les informations en provenance des encodeurs des moteurs, de lire les valeurs de la tension et du courant consommé par les moteurs, de connaître la vitesse de rotation des moteurs, de définir les accélérations, etc.

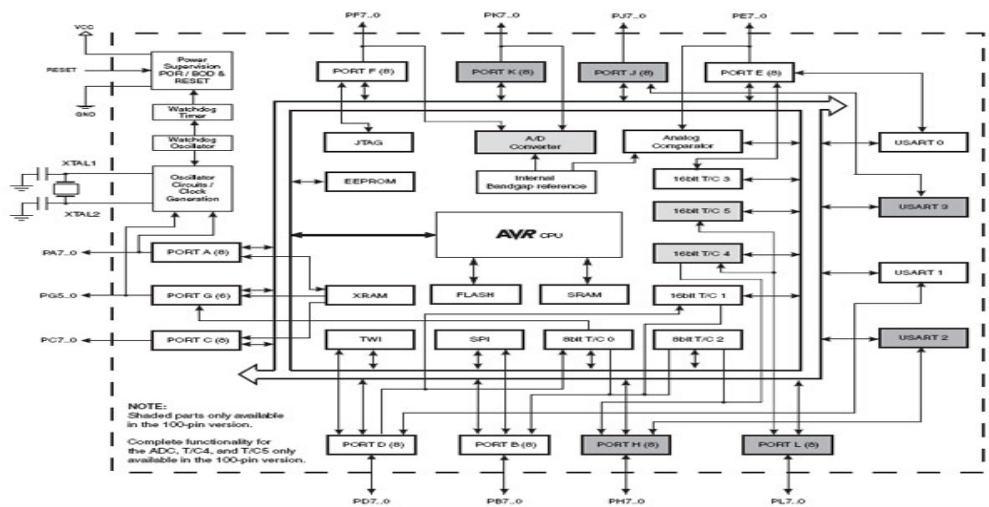
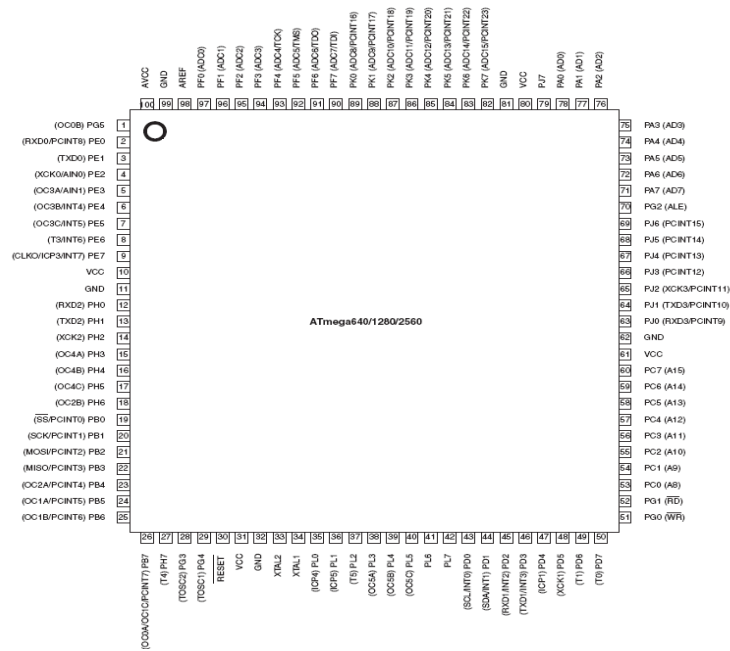
Alimentation nécessaire: 12 Vcc (un régulateur 5 Vcc intégré délivre 300 mA pour notre électronique de commande externe).

Plusieurs platines peuvent être pilotées en même temps en mode de pilotage I2C™. Dimensions: 70 x 60 x 25 mm. A noter que cette platine est idéalement conçue pour exploiter les moteurs "DCM2" (proposés ci-dessous).

**NB.** Il est préférable d'utiliser ce module pour pouvoir commander le système en deux sens.

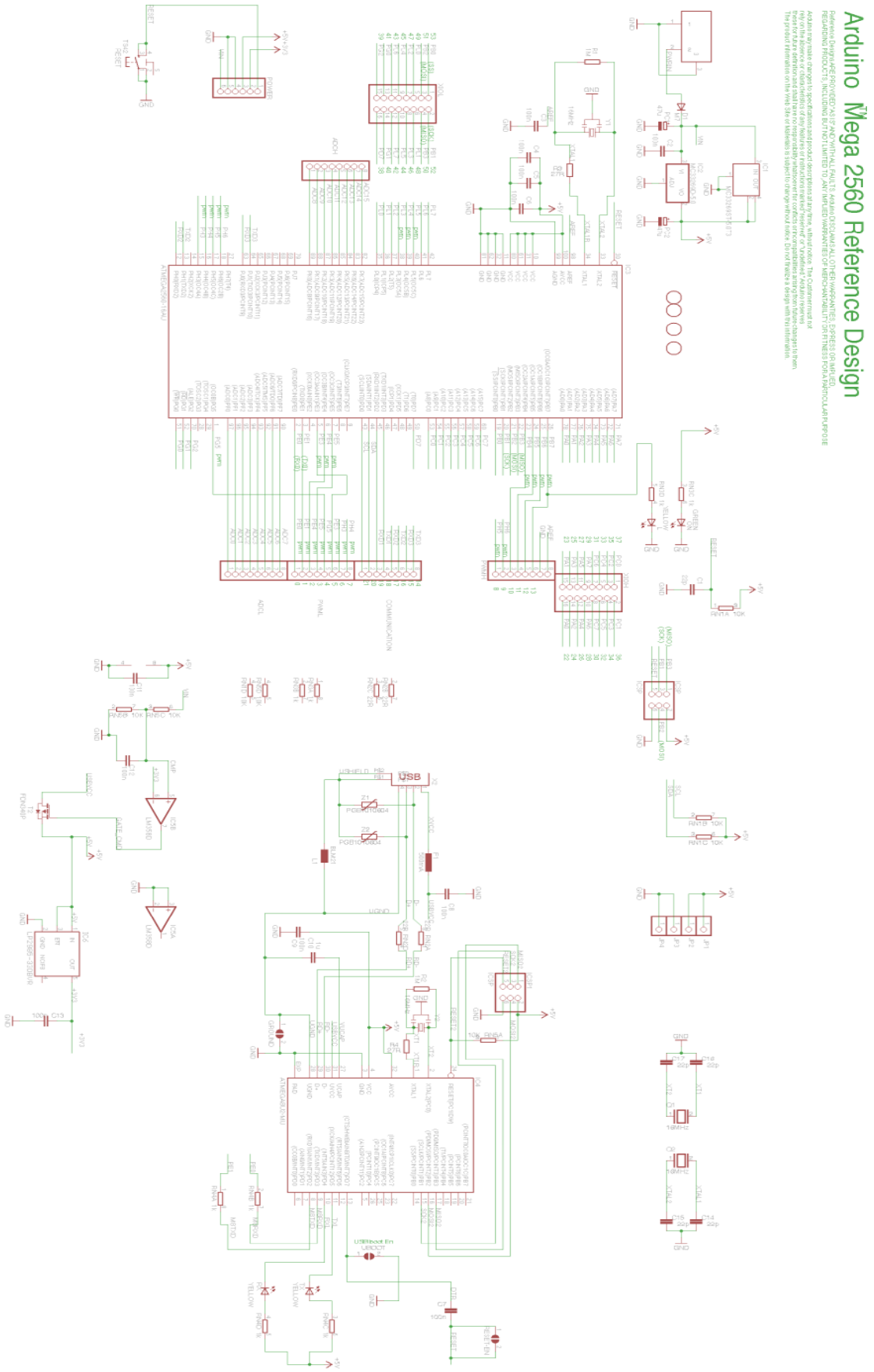


Module de commande de moteur (MD)

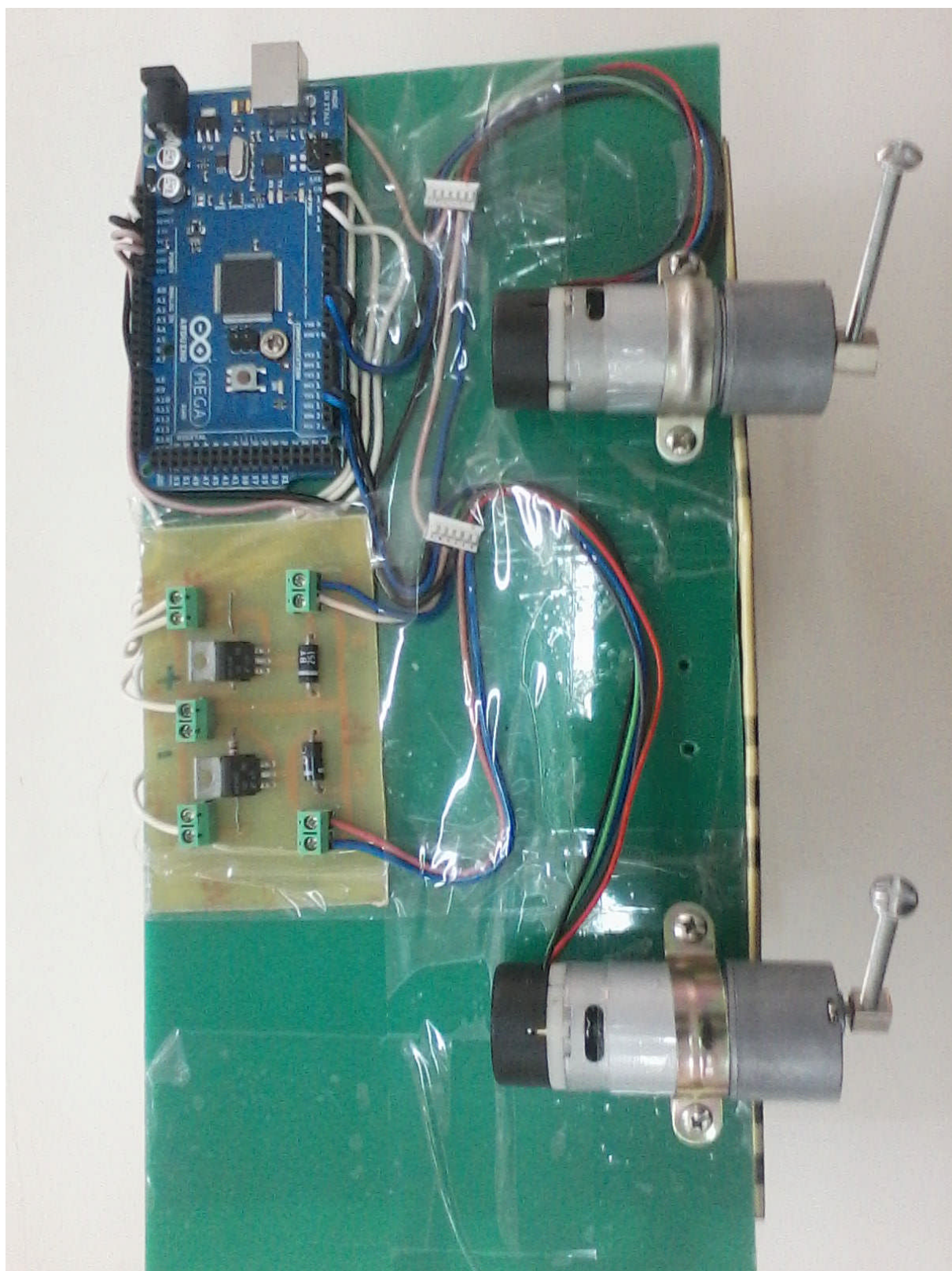


Architecture du microcontrôleur

Schéma électrique de la carte Arduino MEGA2560 :







Maquette de téléopération réalisée