



FACULTE DE GENIE ELECTRIQUE ET D'INFORMATIQUE
DEPARTEMENT D'automatique

**Mémoire de Fin d'Etude
de MASTER ACADEMIQUE**
Spécialité : **automatique option commande des
Systèmes**

Présenté par
Ouahmed MALKI

Mémoire dirigé par Rabah MELLAH

Thème
**Commande en force et position d'une
architecture de télé opération**

Mémoire soutenu publiquement le j u i n 2015 devant le jury composé de :

M Ahmed Maidi

MCA, UMMTO, Président

M Rabah MELLAH

MCA, UMMTO, Promoteur

M Youcef Messar

MAA, UMMTO, Examineur

M Fadhila Boudjema

MAA, UMMTO, Examineur

Remerciement

Je tiens à remercier, tout d'abord mon promoteur Monsieur Mellah Rabah de m'avoir encadré et dirigé durant mon travail, ainsi tous les membres de juré d'avoir accepté à examiner ce modeste travail.

J'exprime toute ma reconnaissance à Monsieur Talem Hand, je te remercie pour ton aide, pour tes encouragements malgré la distance qui nous sépare.

J'exprime aussi ma reconnaissance à Monsieur Mouloudj Kamel pour son aide, ces conseils. Sans oublié mademoiselle Djouher snaoui pour son orientation.

Enfin je tiens à exprimer mes remerciements les plus sincères à ma grand-mère et mes parents, je serai toujours redevable pour tous les efforts que vous m'avez fournis à mon égard. Vous m'avez transmis de belles valeurs et vous m'avez toujours encouragé dans cette voie, merci au fond du cœur ! Mes profonds remerciements s'adressent également à ma tante, mes frères et sœurs qui mon soutenu tout au long de mon cursus universitaires. Je remercie également mes amis et mes proches qui m'ont encouragé chacun à sa façon.

Introduction Générale.....3

Chapitre I : Généralités sur une architecture de télé opération

Première partie : Généralités sur la télé opération

I.1 Introduction.....	5
I.2 bref Historique.....	6
I.3 Modélisation du système.....	7
I.3.1 Modélisation du comportement de l'opérateur humain.....	8
I.3.2 Modélisation du comportement de l'environnement.....	8
I.3.3 Modélisation du comportement du manipulateur maître et esclave.....	8
I.3.3 Modélisation du comportement du réseau de communication.....	9
I.4 Les systèmes de télé opération sans retour d'effort.....	9
I.5 Les systèmes de télé opération avec retour d'effort.....	10
I.6 Stratégies de commande bilatéral.....	10
I.6.1 Asservissement symétrique bilatéral.....	10
I.6.2 Asservissement Asymétrique bilatéral.....	10
I.6.2.1 Principe.....	10
I.7 Performances et stabilité en télé opération	11
I.7.1 Stabilité.....	11
I.7.2 Passivité	11
I.7.3 Transparence	12
I.8 couplage par les positions, couplage par les vitesses.....	12
I.9 Influence des retards sur les systèmes avec retour d'efforts	12
I.9.1 Qualité de la télé présence.....	13
I.9.2 Asservissement Asymétrique.....	13
I.10 Structure à deux canaux Force-position.....	14
Deuxième partie: Etude d'un système de télé opération.....	14
II.11 Exemple du système de télé opération.....	14
II.11.1 Représentation du système de télé opération.....	14
II.11.2 Critères de performances.....	17
II.11.3 Contrôle des systèmes de télé opération bilatérale... ..	18
II.11.3.1 contrôle force-position (FP).....	18
I.12 Conclusion.....	19

Chapitre II : Présentation du matériel

II.1 Introduction.....	21
II.2 Matériel.....	21
II.2.1 Les cartes électroniques.....	21
II.2.2 Les composants électroniques.....	22
II.2.2.1 Les transistors MOSFET / principe.....	23
II.2.2.2 Caractéristique du transistor utilisé/IRF640.....	23
II.2.2.3 Diode et résistance	23
II.3 Présentation du pont en H.....	23
II.3.1 Technique du pont H de base.....	24
II.3.2 Technique du pont H électronique.....	25
II.4 Le choix d'utilisation.....	25

SOMMAIRE

II.5 Capteurs.....	25
II.5.1 Capteur de position.....	25
II.5.2 Capteur d'effort.....	25
II.5.2.1 Capteur utilisé.....	26
II.5.2.2 Fonctionnalités	27
II.5.2.3 Caractéristiques techniques.....	27
II.5.2.4 branchement du capteur à Arduino.....	27
II.6 Caractéristiques des moteurs utilisés.....	27
II.7 Mise en œuvre.....	28
II.7.1 Partie mécanique	28
II.7.1 Le principe de fonctionnement de l'encodeur	28

Deuxième partie : Description du moteur à courant continu(MCC)	30
II.8 Principe de fonctionnement du moteur à courant continu.....	30
II.9 Modélisation et Asservissement du moteur	30
II.9.1 Modélisation du moteur.....	30
II.9.2 Asservissement du moteur.....	31
II.9.2.1 Asservissement du courant/couple... ..	32
II.10 La technique de la PWM pour La variation de vitesse.....	32
II.10.1 Pilotage du sens et vitesse du moteur.....	33
II.11 Conclusion.....	34

Chapitre III : Programmation avec Arduino

III.1 Introduction.....	36
III.2 Présentation Arduino	36
III.3 Avantage Arduino.....	36
III.4 Les outils de l'Arduino.....	37
III.4.1 Architecture hardware de la carte.....	37
III.4.2 Architecture de la carte.....	38
III.4.3 Le microcontrôleur.....	38
III.4.3.1 Caractéristiques du microcontrôleur ATMEGA 2560.....	38
III.4.4 Les entrées-sorties de la carte.....	39
III.4.5 La mémoire.....	39
III.4.6 L'alimentation.....	40
III.4.7 Visualisation.....	40
III.4.7 La connectique.....	40
III.5 Architecture software de la carte.....	41
III.5.1 Présentation de l'Espace de développement Intégré (EDI) Arduino.....	41
III.5.2 Différentes composition de l'espace de développement intégré (EDI)	42
III.5.2.1 Une barre de menus.....	42
III.5.2.2 Une barre des boutons.....	42
III.5.2.3 Structure d'un programme Arduino.....	43
III.5.2.4 Le programme Arduino.....	43
III.5.2.4.1 Description de la structure d'un programme Arduino.....	43
III.5.2.4.2 Le jeu d'instructions du langage Arduino.....	44
III.5.2.4.3 Les étapes à suivre pour programmer la carte.....	46
III.6 Conclusion.....	49

SOMMAIRE

Chapitre IV : La réalisation

IV.1 Introduction	51
IV.2 Simulation du pont H sur ISIS protues.....	51
IV.3 Schéma électrique du pont H.....	51
IV.4 Schéma de réalisation	51
IV.5 Carte de puissance avec le pont utilisé	52
IV.6 moteur coté esclave (capteur de couple).....	53
IV.7 Commande de deux moteurs (télé opération : Force/position) avec arduino	53
IV.8 Les résultats obtenus.....	54
IV.8 La maquette réalisée	55
IV.9 Conclusion	55
Conclusion général.....	57

Introduction

Générale

Introduction générale

Introduction générale [1]

Le mot (télé opération) a son origine dans le mot grec (têle), qui signifie loin, à distance, et le mot latin (operor), qui signifie travailler, œuvrer, accomplir. La notion de télé opération désigne donc une interaction entre l'opérateur et le robot. L'opérateur envoie une consigne et le robot l'exécute. Cette consigne peut prendre plusieurs formes : une destination à atteindre, un plan (suite d'actions) à exécuter, une vitesse à atteindre, un comportement à adopter etc.

Un télémanipulateur maître-esclave (TME) est un dispositif électromécanique qui permet à l'opérateur humain de réaliser une tâche à distance tout en manipulant l'interface maître. Les mouvements qu'il réalise avec cette interface sont transmis vers le manipulateur esclave via un réseau de communication, de l'autre côté, le manipulateur esclave reproduit les mouvements de manière à accomplir la tâche.

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. La transmission entre les deux sites était purement mécanique opérant à courte distance et il transmettait à l'opérateur les forces exercées sur la partie terminale. L'apparition des ordinateurs et le progrès en matière de technologie ont permis le développement de systèmes plus en plus performants dans diverses fonctions et les munir d'actionneurs électriques et de différents capteurs : capteur de position, de vitesse, de couple.

En présence de retards, il y a 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, 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 télé opération.

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 télé opération, 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 kinesthésiques de l'opérateur.

La kinesthésie étant, je le rappelle, la perception consciente des positions dans l'espace et des forces appliquées sur chacune des parties du corps. On parle donc de bonnes sensations kinesthésiques quand les sensations de l'opérateur sont cohérentes avec les interactions entre le robot esclave et son environnement.

Mon projet est basé sur la réalisation d'une architecture de télé opération bilatérale à un degré de liberté.

Problématique :

Introduction générale

Un système de télé opération consiste en trois composants principaux :

- l'interface de l'opérateur (ensemble d'outils qui lui permettent de Contrôler le système).
- le système distant qui réalise la tâche souhaitée.
- le schéma de communication qui relie les deux.

L'interface de l'opérateur est située à une distance de l'environnement de travail, pour lui permettre de travailler d'une façon efficace, il est donc indispensable de lui présenter toute l'information nécessaire à l'accomplissement de la tâche. De plus ces informations doivent lui parvenir sous une forme facile à interpréter et à travers un système de communication fluide.

Cette problématique particulière est présentée par trois aspects : la gestion de la Latence (qui est définie comme le temps que prend le système pour répondre aux commandes de l'utilisateur, en d'autres termes, le temps entre la communication des instructions et la visualisation des résultats), le contrôle du dispositif et l'efficacité de l'interface homme-machine.

Chapitre

I

Première partie : Généralités sur la télé opération

I.1 Introduction : [2]

L'objectif d'un système de télé opération haptique est de permettre à son utilisateur de réaliser une tâche à distance tout en ressentant les forces mises en jeu à l'endroit de la manipulation. Son fonctionnement est basé sur l'échange d'informations de positions et de forces 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.

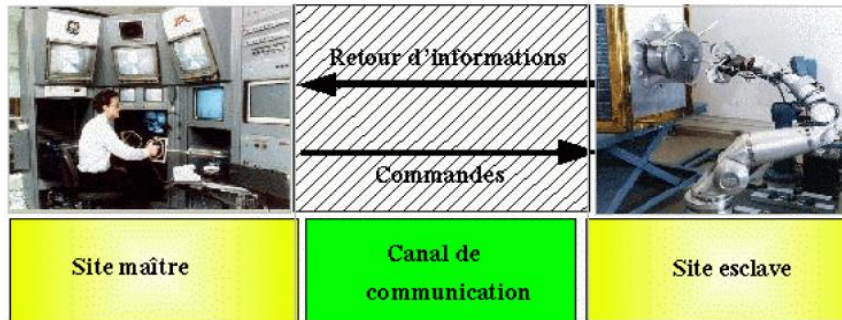


Figure1 : Illustration de l'architecture générale d'un système de Télé opération

Idéalement, le système de télé opération 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 télé opération, 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 [2]

L'histoire de la télé opération commence avec le manipulateur maître-esclave développé par Ray Goertz à l'Argonne National Laboratory (ANL) en 1948. Ce système conçu pour la manipulation de substance toxique (manipuler le carburant radioactif). Dans ce premier système, le manipulateur maître possédait la même structure mécanique et les mêmes propriétés cinématiques que le manipulateur esclave donc les limitations physiques de l'esclave étaient perçus naturellement par l'utilisateur. En 1954 l'ANL développe la deuxième génération de

télémanipulateurs, électriques à retour d'effort. Dans un système à retour d'effort, toute force externe expérimentée par le manipulateur esclave est produite sur le manipulateur maître. Ceci est utilisé pour implanter le contrôle bilatéral : une force appliquée sur l'esclave (respectivement le maître) produira un mouvement du maître (respectivement l'esclave). Les systèmes hydrauliques ont aussi été présents depuis le début de la télé opération. Le premier système de ce type est le (Handyman) de Ralph Mosher développé aux laboratoires de general electric en 1958. Le Handyman, manipulateur à retour d'effort consistait en deux bras hydraulique à 10 degrés de liberté, 2 degrés de liberté pour chaque doigt.



Figure2 : Environnement de télémanipulation au Argonne National Laboratory

Au début des années 60 le champ d'application de la télé opération s'étend à l'exploration spéciale donc à la télémanipulation et au contrôle de véhicule avec des retards de transmission. En 1970, l'exploration sous-marine devient l'un des principaux champs d'applications des systèmes télé opérés.

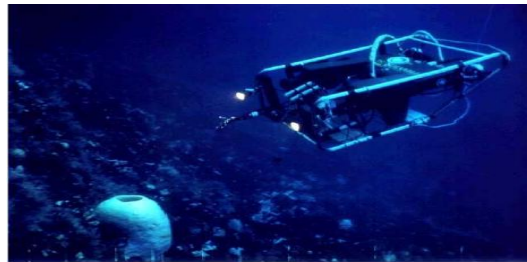


Figure3 : ROV de la NASA pendant la réalisation d'une mission en Antarctique

Dans les années 80, les véhicules opérés à distance ROV (Remotely operated vehicles) commencent à être amplement utilisés dans les industries des gaz et de pétrole. Les ROV sont utilisés pour la surveillance oléoducs et pour l'inspection des structures artificielles sous-marines.

L'apparition des ordinateurs et le progrès en matière de technologie ont permis le développement de systèmes de plus en plus performants dans diverses fonctions : le dépannage de satellite, l'exploration de volcans, le déminage, l'aide aux personnes handicapées (avec, par exemple les fauteuils roulant intelligents...etc.)

I.3 Modélisation du système : [3]

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, particulièrement bien adaptée à l'analyse et la modélisation des systèmes de télé opération avec retour d'efforts, elle permet de représenter de manière assez intuitive les échanges d'énergie et les interactions entre les éléments d'un TME, Ce type d'approche est basé sur une analogie entre modèle électrique et mécanique, où les courants sont remplacés par les vitesses et les tensions par les efforts.

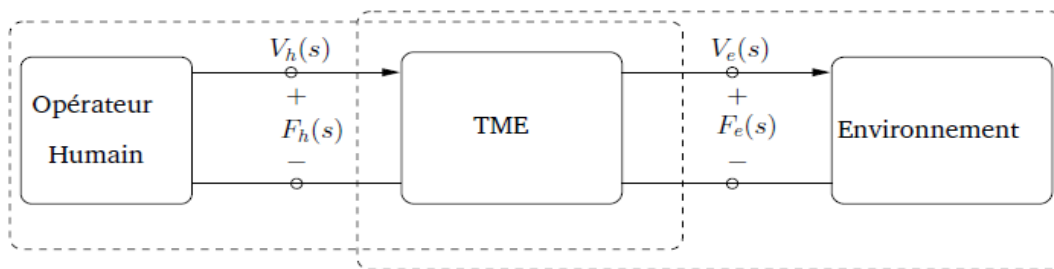


Figure4 : Représentation sous forme quadripôle d'un système de télé opération avec retour d'efforts.

Il est alors possible d'exprimer une relation entrée/sortie du TME sous forme matricielle, les représentations les plus courantes utilisent l'une ou l'autre des matrices d'impédance, d'admittance ou hybride.

La représentation sous forme hybride est certainement la plus utilisée, pour cette représentation les relations entre les efforts et les positions s'écrivent de la manière suivante :

$$\begin{matrix} T_h \\ \theta_s \end{matrix} = H(s) \begin{matrix} \theta_m \\ -T_e \end{matrix}$$

Avec la matrice hybride H $s = \begin{matrix} h_{11} & h_{12} \\ h_{21} & h_{22} \end{matrix}$

Les autres représentations matricielles (impédance et admittance) sont données par les expressions suivantes:

- $T_s = Z_s \theta(s)$

- ❖ La relation entre variables indépendantes (θ_m, θ_s) et les variables dépendantes (T_h, T_e) est décrite par une matrice Z qu'on appelle la matrice d'impédance qui généralise l'impédance opérationnelle d'un dipôle.

$$\begin{matrix} T_h \\ T_e \end{matrix} = \begin{matrix} Z_{11} & Z_{12} \\ Z_{21} & Z_{22} \end{matrix} \begin{matrix} \theta_m \\ -\theta_s \end{matrix} \dots\dots\dots \text{Impédance}$$

- ❖ Si on choisi comme variables indépendantes les couples (T_h, T_e) au port du quadripôle on obtient la représentation :

- $\theta(s) = Y(s)T(s)$

$$\begin{matrix} \theta_m \\ -\theta_s \end{matrix} = \begin{matrix} Y_{11} & Y_{12} & T_h \\ Y_{21} & Y_{22} & T_e \end{matrix} \dots\dots\dots \text{Admittance}$$

I.3.1 Modélisation du comportement de l'opérateur humain [3]

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. Bien que les problématiques en télé robotique soient similaires, les chercheurs du domaine ne se sont intéressés à ces travaux que bien plus tard. 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.3.2 Modélisation de l'environnement [3]

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 $f_e t^*$. 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.3.3 Modélisation des manipulateurs maître et esclave : [3]

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.

Le schéma bloc d'un tel manipulateur en interaction avec l'environnement est représenté par cette figure :

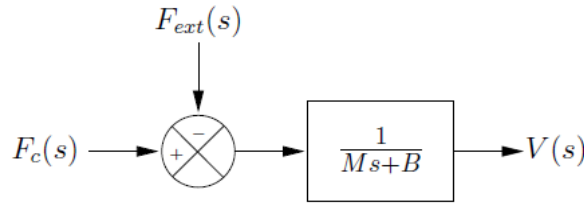


Figure5 : Manipulateur de type impédance

L'impédance est alors notée $Z(s)=Ms+B$, avec M et B représentent l'inertie et l'amortissement qui caractérisent le comportement dynamique du manipulateur. La vitesse $V(s)$ du manipulateur résulte alors de l'effort de commande $F_c(s)$ et de l'effort d'interaction avec l'environnement $F_{ext}(s)$, suivant l'expression suivante :

$$V(s) = Z(s)^{-1}[F_c(s) - F_{ext}(s)] \dots\dots\dots I.2$$

La vitesse du manipulateur maître est désignée par $V_h(s)$ du fait de l'interaction avec l'utilisateur :

$$V_h(s) = Z_m(s)^{-1}[F_m(s) + F_h(s)] \dots\dots\dots I.3$$

Si le manipulateur esclave est en contact avec un environnement, sa vitesse dépend $V_e(s)$ dépend de l'effort d'interaction $F_e(s)$ et l'effort de commande $F_s(s)$, de la manière suivante :

$$V_e(s) = Z_s(s)^{-1}[F_s(s) - F_e(s)] \dots\dots\dots I.4$$

Les signes dépendent naturellement du type d'interaction considérée : l'interaction entre l'utilisateur et interface maître, ou celle du manipulateur esclave avec l'environnement.

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, à cet environnement en réponse à un contact avec celui-ci. Ces manipulateurs sont généralement non réversibles et présentent une grande raideur.

I.3.4 Modélisation du réseau de communication [3]

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.4 Les systèmes de télé opération sans retour d'effort [4]

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 6).

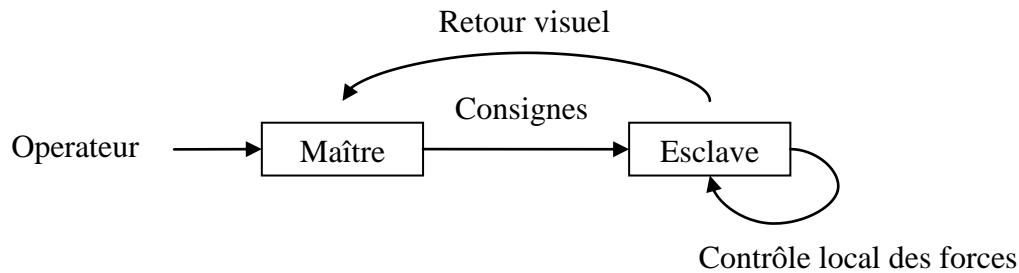


Figure 6 : Télé opération sans retour d'efforts

I.5 Les systèmes de télé opération avec retour d'effort

Le système de télé opération 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. 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 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. [4]

I.6.1 Asservissement symétrique bilatéral :

Chaque robot est asservi sur la position de l'autre robot, l'asservissement bilatéral est réalisé en envoyant comme commande aux actionneurs la différence de position entre les deux robots, chaque robot reçoit une consigne proportionnelle à l'écart de position entre les deux robots.

I.6.2 Asservissement Asymétrique bilatéral : [4]

Equipé les robots de capteurs d'efforts, le système de couplage résultant est asymétrique : asservissement de l'esclave sur la position de maître, et asservissement du robot maître sur les forces de l'esclave.

I.6.2.1 Principe :

Le schéma d'asservissement bilatéral asymétrique avec retour d'effort améliore généralement la télé présence (l'opérateur sent les forces exercées sur le robot maître comme si elles étaient exercées directement sur sa propre main).

Le robot esclave est asservi sur la position du maître et le robot maître reçoit comme consigne les forces appliquées par le robot esclave sur son environnement, et mesurées au moyen d'un capteur d'effort (figure7)

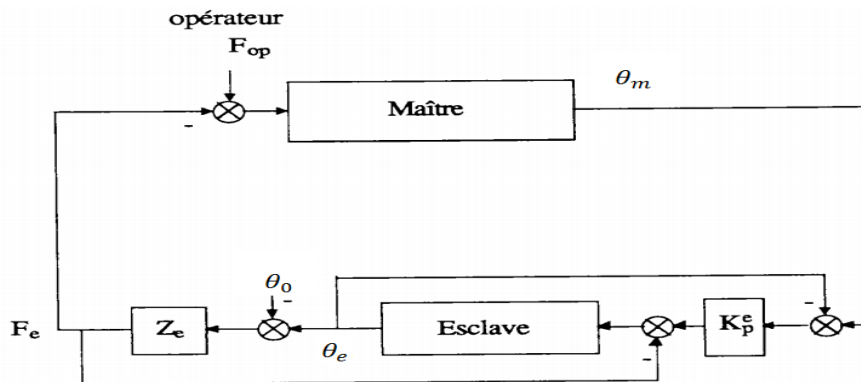


Figure7 : Asservissement force-Position

La force mesurée F_e résulte des interactions de l'esclave avec un environnement de raideur Z_e . Si le contact a lieu en un point de coordonnées θ_0 la relation liant la force de contact au déplacement est :

$$F_e = Z_e(\theta_e - \theta_0) \dots \dots \dots \text{I.5}$$

La figure 7 représente le cas idéal : la force mesurée est la force réellement appliquée sur l'environnement par le robot esclave. En pratique, On cherchera à mesurer une force qui soit le plus proche possible de cette force réelle.

I.7 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.7.1 Stabilité [5]

Pour les systèmes à retour d'effort, il n'y a pas de problème tant que le robot esclave est libre (la force retournée à l'opérateur est nulle), par contre les contacts poseront de sérieux problèmes de stabilité. Lorsque l'on considère un robot en contact avec un environnement, la stabilité de la commande en boucle fermée dépend non seulement de la dynamique du robot, mais aussi de celle de l'environnement.

Pour notre application, la stabilité de la commande peut être obtenue en se basant sur un modèle connu de l'environnement. L'outil privilégié pour garantir la stabilité de tel système est l'analyse de passivité. Plus précisément, on cherche à ce que le port d'interaction (correspond au point d'échange de puissance entre le système et la source d'énergie extérieur) entre le robot et l'environnement soit passif (ne crée pas d'énergie), en d'autres termes, le transfert entre les efforts externes appliqués au système et la vitesse v du système ne crée pas d'énergie. Une fois la passivité de ce port vérifiée, le système interagira via ce port, de façon stable, avec n'importe quel environnement passif. Donc la stabilité est déduite de la passivité.

I.7.2 passivité [5]

Une définition générale de la passivité, inspirée de la théorie des réseaux, permet d'énoncer qu'un système est passif en un port d'interaction lorsque, pour un état initial quelconque du système et pour une entrée quelconque appliquée au système, la puissance extérieure, fournie au système via ce port, est positive.

Chaque port d'interaction d'entrée T (un effort), et de sortie V ou θ (vitesse ou position), est caractérisé par ses opérateurs immittance Z (impédance) ou Y (admittance) avec $\theta = Z(s)T$ et $T = Y(s)\theta$. Ces opérateurs sont des matrices de transfert de fonctions réelles rationnelles en la variable de Laplace.

Ce port d'interaction est passif si et seulement si la matrice $Y(s)$ est Positive Réelle (PR).

De plus, d'après la Propriété 1 ci-après, la passivité d'un port d'interaction peut aussi être montrée par l'étude de la Positivité Réelle de $Z(s)$.

Propriété 1: Si $Y(s)$ est positive réelle et que $Z(s) = Y(s)^{-1}$ et $[I_n + Y(s)]^{-1}$ existent, alors $Z(s)$ est PR.

Cette propriété est très pratique car elle permet d'avoir le choix de montrer la passivité d'un port d'interaction soit par l'étude de la positivité réelle de son admittance $Y(s)$, soit par l'étude de la positivité réelle de son impédance $Z(s)$.

I.7.3 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 citées, 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.8 couplage par les positions, couplage par les vitesses :

Pour piloter des déplacements, on pouvait envoyer des consignes de position ou de vitesse, ces deux couplages sont strictement équivalents. Mais si le robot esclave rencontre un obstacle, il y a une différence importante qu'il est bon d'avoir à l'esprit.

Prenons notre cas qui est un robot à un degré de liberté repéré par l'axe θ , si le robot esclave rencontre un obstacle à l'abscisse θ_e , l'asservissement commute automatiquement en asservissement de force et impose une force constante positive sur l'obstacle. Pendant ce temps l'opérateur a pu commander un déplacement au delà de l'obstacle, qui n'est donc pas réalisé par le robot esclave.

Dans le cas où le robot esclave est asservi sur les positions du robot maître, il faudra que l'opérateur envoie une consigne hors de l'obstacle pour que le robot esclave quitte l'obstacle, c'est-à-dire inférieure à θ_e , par contre Dans le cas où le robot esclave est asservi sur les vitesses du robot maître, le robot esclave quittera l'obstacle dès que le robot maître aura envoyé une consigne de vitesse négative.

Donc dans le 1^{er} cas, le robot esclave est asservi sur la position absolue du robot maître, dans le second cas, il y a perte de cohérence entre la position du maître et celle de l'esclave.

I.9 influence des retards sur les systèmes avec retour d'efforts :

La présence de retard de transmission entre les deux sites d'un système de télé opération à asservissement bilatéral a pour effet principal de générer un frottement visqueux rapidement très important, cet effet ne peut être éliminé qu'en diminuant les gains des asservissements de position.

Un contact avec l'environnement est alors moins sensible à l'opérateur et la qualité de la télé présence est forcément diminuée.

I.9.1 Qualité de la télé présence :

Supposons que le robot esclave soit en contact avec un environnement d'une certaine raideur. Lorsque l'opérateur exerce un effort sur le robot maître, le mouvement résultant est transmis comme consigne au robot esclave.

Comme celui-ci est en contact avec l'environnement l'asservissement de position génère une force (d'autant plus importante que l'environnement est raide), cette force est ramenée au robot maître comme une contre réaction.

Le système bouclé maître-esclave de la figure 8 réalise donc un asservissement sur la force par l'opérateur et sur la position de l'environnement. La sensation de l'opérateur d'être réellement en contact avec l'environnement dépendra donc de la bande passante et du gain statique du système complet, on cherchera à augmenter au maximum le gain de retour de force.

I.9.2 Asservissement Asymétrique : [4]

Pour les systèmes à retour d'effort, dont le principe de couplage est représentée par la (figure 8), il n'y a pas de problème tant que le robot esclave est libre (la force retournée à l'opérateur est nulle), par contre les contacts poseront de sérieux problèmes de stabilité.

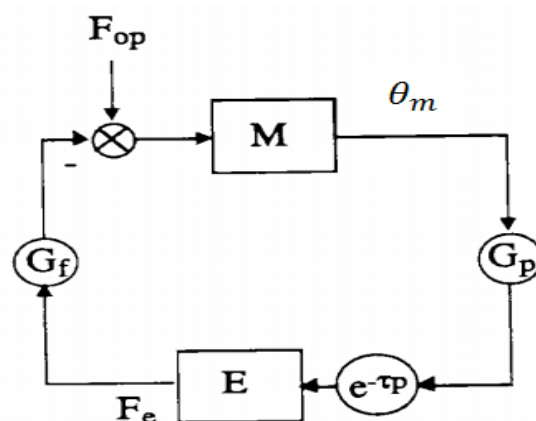


Figure8 : Couple Maître-esclave à retour d'effort.

Le robot maître M est asservi en force, il reçoit comme consigne la force F_e exercée par l'esclave sur l'environnement à travers un gain G_f . Le robot esclave E est asservi en position, il reçoit comme consigne la position θ_m mesurée du maître à travers un gain G_p .

On note M la fonction de transfert de l'ensemble composé par le robot maître et l'opérateur, cette fonction de transfert est caractérisée par un gain statique (raideur $\frac{1}{Z_m}$) et par une dynamique D1(p). la relation liant la position θ_m du robot maître à la force F_m exercée sur celui-ci est :

$$\theta_m = \frac{1}{Z_m} D1(p) F_m \quad \text{avec} \quad F_m = F_{op} - G_f F_e \dots\dots\dots(I.6)$$

On note E la fonction de transfert de l'ensemble composé par le robot esclave asservi en position et l'environnement. On note Z_e la raideur de l'asservissement de position du robot esclave en contact avec son environnement, D2(p) est la dynamique de E.

La relation liant la force F_e mesurée par le capteur d'effort du robot esclave et la position commandée θ_s est :

$$F_e = Z_e D2(p) \theta_s \quad \text{avec} \quad \theta_s = G_p \theta_m \dots\dots\dots(I.7)$$

Pour simplifier on regroupe dans un terme D(p), la dynamique du couple maître-esclave en boucle ouverte : $D(p)=D1(p)D2(p)$. la fonction de transfert en boucle ouverte F(p) du couple maître-esclave est donc :

$$F(p)=e^{-Tp} G_f G_p \frac{Z_e}{Z_m} D(p) \dots\dots\dots(I.8)$$

Où T est le retard de transmission total dans la boucle.

On a représenté le diagramme de Bode du système sans prendre en compte le retard (en trait fin), et le diagramme de Bode du système en prenant compte le retard (en trait épais) sur la figure 7.

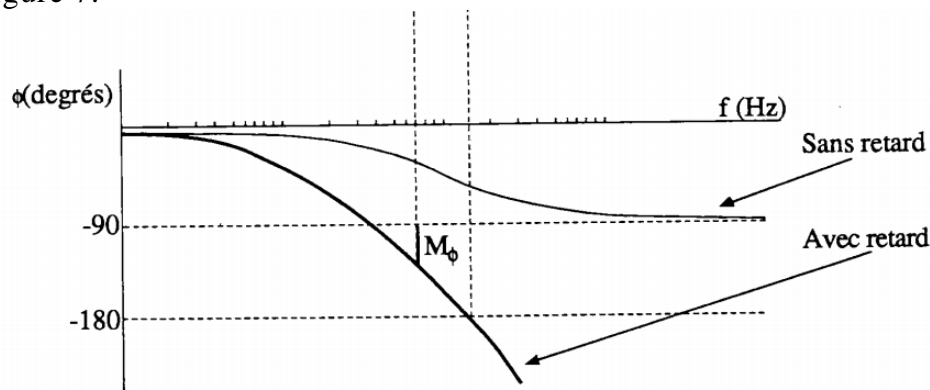


Figure9 : Diagramme de Bode d'un modèle simplifié d'un couple maître-esclave, influence de retard.

Pour que le système soit le plus transparent possible à l'opérateur, il faut que le gain total dans la boucle soit élevé et que la dynamique D(p) soit rapide.

Le retard pur amène un déphasage, il y a donc une diminution de la marge de phase M_ϕ . Le retard pur pose donc forcément une instabilité, on peut jouer sur plusieurs facteurs pour tenter de stabiliser le système malgré le retard :

- Diminuer les gains G_f et G_p .
- Augmenter Z_m .
- Diminuer Z_e .
- Diminuer la bande passante de D(p).

I.10 Structure à deux canaux Force-position :

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 coté esclave seulement ou des deux coté. La version originale présentée n'utilise que les efforts mesurés coté esclave. L'utilisation de la mesure d'effort coté maître permet d'accroître les performances du système, et d'assurer, si nécessaire, une meilleure réversibilité du manipulateur maître.

Deuxième partie: Etude d'un système de télé opération

I.11 Exemple du système de télé opération :[2]

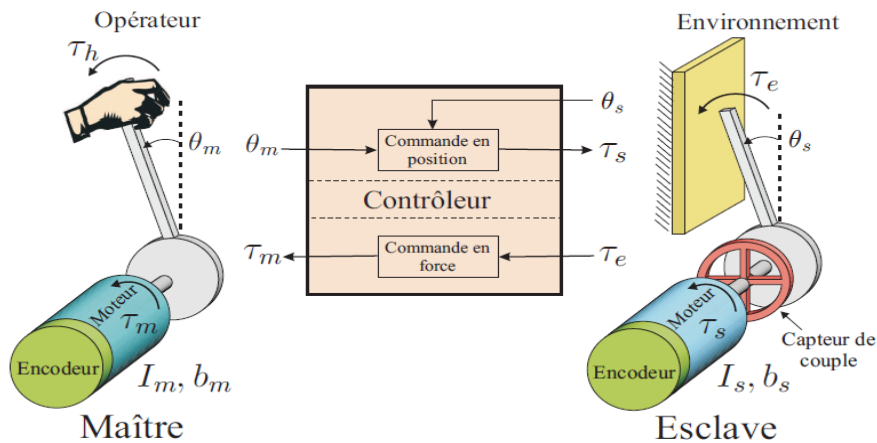


Figure10 : Exemple de système de télé opération haptique à 1ddl (mode de contrôle Force-Position)

I.11.1 Représentation d'un système de télé opération : [2]

La Figure 10 représente un système classique de télé opération haptique à 1 degré de liberté. 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 la relation suivante :

$$\Sigma \text{couple} = \text{la matrice d'inertie} * \text{l'accélération angulaire} = I\ddot{\theta} \quad \text{.....(I.9)}$$

On applique la relation (I.9) pour le maître on obtient :

$$T_h + T_m - b_m \dot{\theta}_m = I_m \ddot{\theta}_m \quad \text{.....(I.10)}$$

On applique la relation (I.9) pour l'esclave on obtient :

$$T_s + T_e - b_s \dot{\theta}_s = I_s \ddot{\theta}_s \quad \text{.....(I.11)}$$

De là on aura ces deux équations ci-dessous :

$$\begin{aligned} I_m \ddot{\theta}_m + b_m \dot{\theta}_m &= T_h + T_m \\ I_s \ddot{\theta}_s + b_s \dot{\theta}_s &= T_s + T_e \end{aligned} \quad \text{.....(I.12)}$$

Avec θ_m et θ_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.12) en coordonnées de Laplace :

$$\begin{aligned} s^2 I_m \theta_m + s b_m \theta_m &= T_h + T_m \\ s^2 I_s \theta_s + s b_s \theta_s &= T_s + T_e \\ (I_m s^2 + s b_m) \theta_m &= T_h + T_m \\ (I_s s^2 + s b_s) \theta_s &= T_s + T_e \end{aligned} \quad \dots\dots\dots (I.13)$$

En introduisant la notion d'impédance du maître et de l'esclave $Z_m(s)$ et $Z_s(s)$:

$$\begin{aligned} Z_m \theta_m &= T_h + T_m \\ Z_s \theta_s &= T_s + T_e \end{aligned} \quad \dots\dots\dots (I.14)$$

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

On pourra développer l'équation (I.14) pour avoir une forme matricielle :

$$\begin{aligned} T_h &= Z_m \theta_m - T_m \\ \theta_s &= \frac{1}{Z_s} \cdot T_s + \frac{1}{Z_s} \cdot T_e \end{aligned} \quad \dots\dots\dots (I.15)$$

On aura la forme matricielle de notre système comme suit :

$$\begin{aligned} T_h &= Z_m \theta_m - T_m \\ \theta_s &= 0 \cdot \frac{1}{Z_s} \cdot T_h + \frac{1}{Z_s} \cdot T_e \end{aligned} \quad \dots\dots\dots (I.16)$$

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 (θ_m, θ_s) et les forces d'interactions (T_h, T_e) et cette relation est exprimée par :

$$\begin{aligned} T_h &= h_{11} \theta_m + h_{12} \theta_s \\ \theta_s &= h_{21} T_h + h_{22} T_e \end{aligned} \quad \dots\dots\dots (I.17)$$

Cette matrice est la matrice d'immittance hybride H de dimension 2×2 . 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).

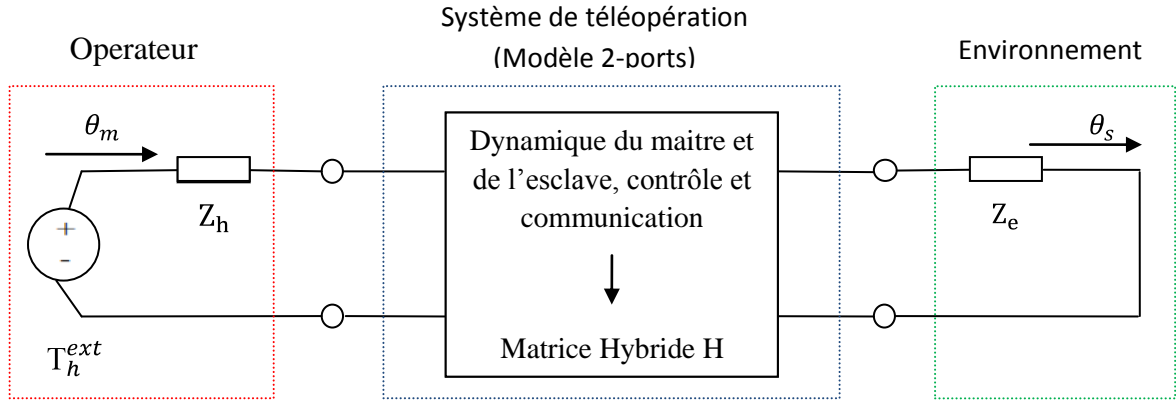


Figure 11: représentation d'un système de téléopération par modèle 2-ports

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,

$$T_e = -k_e \theta_s = -Z_e \theta_s \quad \text{.....(I.18)}$$

L'opérateur est modélisé par l'impédance ;

$$Z_h = I_h s^2 + b_h s + k_h \quad \text{..... (I.19)}$$

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,

$$T_h = T_h^{ext} - Z_h \theta_m \quad \text{.....(I.20)}$$

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

I.11.2 Critères de performances [2]

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,

$$\forall t \quad \begin{matrix} \theta_s t = \theta_m t \\ T_h t = -T_e t \end{matrix} \quad \text{..... (I.21)}$$

Le signe négatif dans l'expression du tracking en force est dû à l'opposition entre le couple de l'utilisateur et celui de l'environnement à l'équilibre, suivant les équations (I.16) et (I.17) la transparence parfaite est atteinte lorsque la matrice hybride H présente la forme suivante :

$$\begin{matrix} T_h \\ \theta_s \end{matrix} = H_{ideale} * \begin{matrix} \theta_m \\ T_e \end{matrix} = \begin{bmatrix} 0 & -1 \\ 1 & 0 \end{bmatrix} * \begin{matrix} \theta_m \\ T_e \end{matrix} \quad \text{.....(I.22)}$$

La matrice hybride correspond à une méthode de contrôle donnée peut alors être comparée à H_{ideale} pour évaluer la transparence.

$$\begin{aligned} T_h &= h_{11}\theta_m + h_{12}T_e \\ \theta_s &= h_{21}\theta_m + h_{22}T_e \end{aligned}$$

Chaque terme de la matrice H possède une signification physique :

- $h_{11} = \frac{T_h}{\theta_m}$; (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.

- $h_{21} = \frac{\theta_s}{\theta_m}$; (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.

I.11.3 Contrôle des systèmes de télé opération bilatérale

Les principales méthodes de contrôle mises en œuvre dans les systèmes de téléopération haptique 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.

I.11.3.1 contrôle force-position (FP) :

La méthode de contrôle Force-Position, également appelée contrôle par impédance ou en anglais Direct Force Feedback, est la méthode la plus naturelle à imaginer pour un retour de force. En effet, elle est basée sur la mesure de l'interaction entre l'esclave et l'environnement par un capteur de force, dans lesquelles la force d'interaction est calculée par la simulation et renvoyée comme consigne vers le maître.

Nous avons le schéma de contrôle ci-dessous :

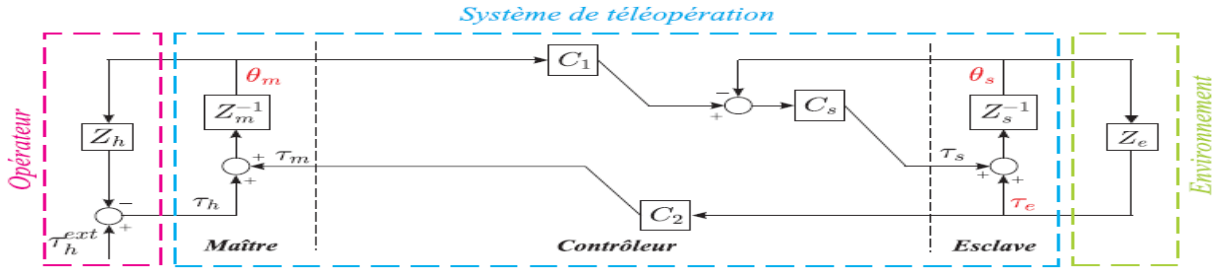


Figure12 : Contrôle Force-Position (FP)

La position de l'utilisateur θ_m est mesurée par l'encodeur du maître et envoyée comme consigne vers l'esclave, à travers le canal de communication C_1 . La différence avec la mesure de l'encodeur du robot θ_s est alors injectée dans le contrôleur de position C_s pour calculer la commande motrice T_s . En même temps, l'interaction entre l'esclave et son environnement T_e , mesurée par le capteur de force, est renvoyée comme consigne vers le maître, à travers le canal de communication C_2 , pour générer le retour d'effort T_m .

Les commandes des moteurs sont donc définies par :

$$\begin{aligned} T_m &= C_2 T_e \\ T_s &= C_s (C_1 \theta_m - \theta_s) \end{aligned} \quad \text{.....(I.23)}$$

Avec $C_s = (k_d^s s + k_p^s)$ la fonction de transfert du contrôleur de position de type proportionnel/dérivé .les blocs C_1 et C_2 représentent respectivement le transfert de position du maître vers l'esclave et le transfert de force de l'esclave vers le maître.ils modélisent les éventuels gains d'amplification α (position) et β (force), et les délais dans les communications Δ_1 et Δ_2 . La transformé de Laplace d'un délai d'une durée Δ est donnée par $e^{-s\Delta}$, ce qui permet d'écrire :

$$C_1 = \alpha e^{-s\Delta_1} \quad C_2 = \beta e^{-s\Delta_2}$$

En exploitant le schéma de contrôle force-position et l'équation (I.23) on pourra écrire T_h et θ_s en fonction de θ_m et T_e pour avoir une représentation hybride de notre système :

$$\begin{aligned} \theta_m Z_m &= T_m + T_h \\ T_m &= C_2 T_e \end{aligned} \quad \text{.....(I.24)}$$

On remplace T_m par sa valeur pour obtenir T_h :

$$T_h = \theta_m Z_m - C_2 T_e \quad \text{.....(I.25)}$$

De la même manière pour θ_s :

$$\begin{aligned} \theta_s &= Z_s^{-1} T_s + T_e = Z_s^{-1} C_s C_1 \theta_m - \theta_s + T_e \\ \theta_s Z_s &= C_s C_1 \theta_m - C_s \theta_s + T_e \\ \theta_s (Z_s + C_s) &= C_s C_1 \theta_m + T_e \quad \text{.....(I.26)} \\ \theta_s &= \frac{C_1 C_s}{Z_s + C_s} \theta_m + \frac{1}{Z_s + C_s} T_e \end{aligned}$$

La représentation hybride de notre system télé opération est la suivante :

$$\begin{matrix} T_h \\ \theta_s \end{matrix} = \begin{matrix} Z_m & C_2 \\ \frac{C_1 C_s}{Z_s + C_s} & \frac{1}{Z_s + C_s} \end{matrix} \begin{matrix} \theta_m \\ T_e \end{matrix} \dots\dots\dots (I.27)$$

I.12 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 du ce chapitre, consacrée à l'historique, a permis de situer la télé opération dans un contexte plus général, Puis 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, j'ai 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 télé opération bilatérale, à savoir la structure force-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

Première partie : le matériel utilisé

II.1 Introduction

Dans ce chapitre, nous présenterons le matériel utilisé, les différents composants électroniques et leurs caractéristiques techniques, ainsi les différents techniques à appliquer afin d'avoir une idée précise sur la maquette à réaliser.

Ce chapitre n'est pas un des plus simples car il va faire apparaître des notions de physique moment cinétique et des équations d'électromagnétique, électrique, qui sont indispensables pour comprendre le mouvement. Avant de passer à la télé opération nous devons avoir une idée précise sur les caractéristiques des moteurs utilisés, leurs performances tout on les testons indépendamment afin de comprendre le bon fonctionnement du moteur à courant continu, l'asservissement....etc.

II.2 Le matériel : [7]

Afin de réaliser notre projet qui est un système classique de télé opération haptique à 1 degré de liberté, un pont H est demandé pour le déplacement des robots (Maître/Esclave) dans les deux sens (sens positif, négatif), ainsi l'utilisation d'une carte de puissance pour l'alimentation et la commande des deux moteurs, et une carte Arduino Méga 2560 (carte d'acquisition).

La Figure 13 représente les composants du système de télé opération maître/esclave utilisés. Ce système 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'informations (capteurs et commandes) entre les deux moteurs en fonction du type de contrôle haptique implémenté, et un PC qui me permet d'introduire mon programme dans la carte de contrôle et d'interagir avec l'expérience.



Figure 13: Diagramme de connexion du dispositif de contrôle et de commande

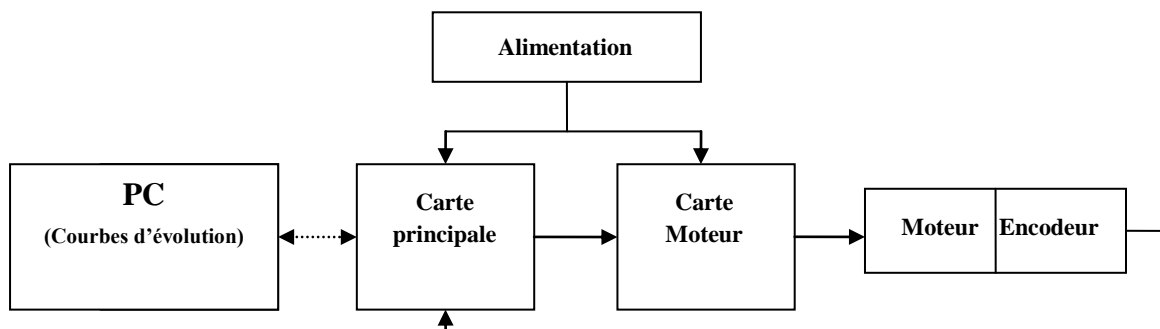
II.2.1 Les cartes électroniques:

Deux cartes électroniques sont utilisées pour cet asservissement :

- La première carte (carte d'acquisition), 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.
- La deuxième carte (carte puissance), permet de contrôler la tension à fournir aux moteurs (par une modulation PWM) ainsi que le sens de déplacement selon les ordres de la carte principale. 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 de commande.



II.2.2 les composants électroniques:

II.2.2.1 Les transistors MOSFET :

• Principe du transistor MOSFET :

Contrairement au transistor Bipolaire dont le principe est d'avoir un courant au niveau de la Base et à partir du moment où la tension (entre: Base-Emetteur) atteint une tension de seuil environ 0.7v, ce courant de la Base va pouvoir passer à l'émetteur ce qui permet la libération d'un autre courant (entre : Collecteur-Emetteur) qui est proportionnel au courant de la base, le transistor MOSFET est un transistor à effet de champ (MOSFET : Métal-Oxyde-Semiconductor-Field-Effect-Transistor), qui est piloté par une tension appliquée à la grille, et non par un courant. On parle donc plutôt de Grille : Résistance au courant à une très forte impédance, Drain : zone de sortie des charges, Source : zone d'entrée des électrons dans le dispositif.

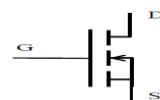


Figure 14: Transistor MOSFET

On mesure V_{gs} (tension entre Grille-Source) :

$$\left\{ \begin{array}{l} \text{si } V_{gs} = 0 \Rightarrow I_D = 0 \text{ (le courant ne passe pas)} \\ \text{si } V_{gs} > 0 \Rightarrow I_D > 0 \text{ (le courant passe)} \end{array} \right. \Rightarrow \text{type: canal N à enrichissement}$$

$$\left\{ \begin{array}{l} \text{si } V_{gs} = 0 \Rightarrow I_D > 0 \text{ (le courant passe)} \\ \text{si } V_{gs} > 0 \Rightarrow I_D = 0 \text{ (le courant ne passe pas)} \end{array} \right. \Rightarrow \text{type: canal N à appauvrissement}$$

II.2.2.2 Caractéristique du transistor utilisé (IRF 640):

Pour ma réalisation nous avons utilisé un transistor de type MOSFET (IRF640), Ces paramètres et caractéristiques sont donnés dans le tableau ci-dessous :

Type désignation	IRF 640
Type IRF transistor	MOSFET
Type de canal transistor	N-Channel
Dissipation de puissance maximale (Pd)	Pd =125 w
Tension maximale drain-source (Vds)	Vds =200 v
Tension maximale Grille-source (Vgs)	Vgs =20 v
Courant de drain maximum (Id)	Id =18 A
Température de jonction maximale (Tj)	C°=150

Figure 15: Caractéristique du transistor IRF 640

II.2.2.3 Diodes et résistances:

Pour une bonne protection on dispose des résistances (par exemple pour limiter le courant de base....), des diodes qui sont capables (de passer de l'état bloquant à l'état passant) de basculer de manière très rapide dès qu'il ya une surtension engendrée par le moteur, lorsqu'on le coupe de l'alimentation, la diode va l'absorber aussitôt avant que le transistor ait le temps d'avoir des dommages.

II.3 Présentation du pont en H:

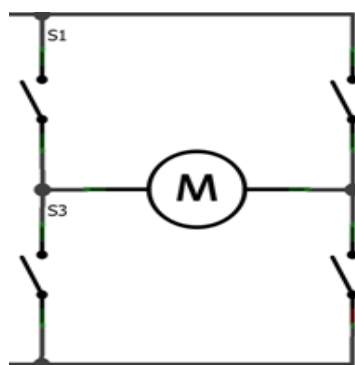


Figure 16: Structure du Pont H

Le pont en H est une structure électronique servant à contrôler la 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.

II.3.1 Technique du pont H de base:[6]

Le pont H est un montage électrique simple qui permet de changer le sens du courant dans un moteur ou d'inverser la tension, c'est ce qu'on envoie dans les deux schémas ci-dessous :

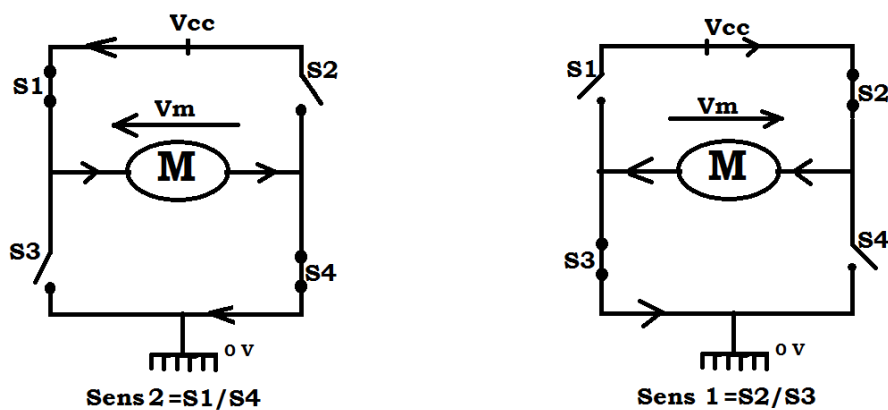


Figure 17: structure du pont H électrique (pont H de base)

M : le moteur.

S_1, S_2, S_3, S_4 : Interrupteurs peuvent être modalisés dans la réalité par des relais électromécaniques ($R_{L1}, R_{L2}, R_{L3}, R_{L4}$).

On remarque bien que ces relais fonctionnent par pair :

R_{L1}	R_{L2}	R_{L3}	R_{L4}	Résultats
X	Ok	Ok	X	Le moteur tourne dans le sens inverse des aiguilles d'une montre
Ok	X	X	Ok	Le moteur tourne dans le sens des aiguilles d'une montre
Ok	X	Ok	X	Court circuit
X	Ok	X	Ok	Court circuit

Si on ouvre les relais R_{L1}, R_{L4} et je ferme les relais R_{L2}, R_{L3} on revanche le courant se dirige dans le sens1, et pour les dipôles récepteurs la tension va dans le sens inverse du courant.

Si on ferme les relais R_{L1}, R_{L4} et j'ouvre les relais R_{L2}, R_{L3} le moteur tourne dans le sens2.

Il ne faut jamais effectuer une autre combinaison du type R_{L1} et R_{L3} car on relie directement l'alimentation à la masse ce qui fait gréer l'alimentation et c'est ce qui fait carrément un court circuit.

II.3.2 Technique du pont H électronique:

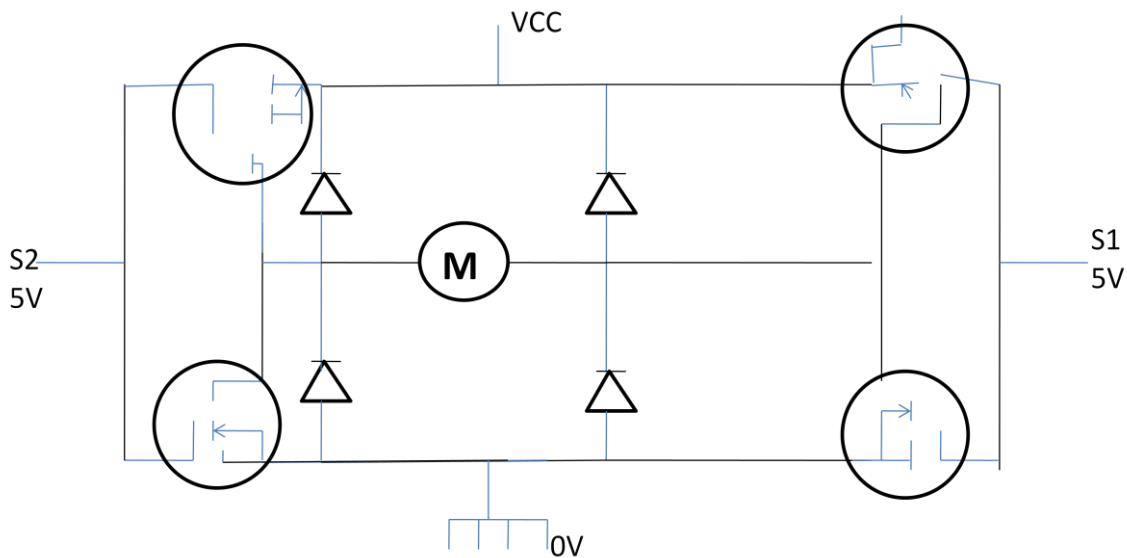


Figure 18: Structure du pont H électronique

II.4 Le choix d'utilisation:

A la place des relais on utilise des transistors MOSFET, cette technique est plus adaptée car si on varie la vitesse de rotation des moteurs on préfère d'utiliser ces transistors, puisque il fallait commuter les relais (c'est-à-dire passer de 0 à 5V, de 0 à 5V ...etc.) à une fréquence assez élevée (20 KHz) c'est beaucoup trop pour un relais (destruction du composant et générer des étincelles), par contre le transistor MOSFET supporte une fréquence de commutation plus élevée que les relais.

II.5 Capteurs :

Les capteurs permettent de transformer une grandeur physique en une grandeur généralement électrique, comme dans notre cas : la position et la force.

II.5.1 Capteur de position :

Les capteurs de position ou de vitesse sont généralement mis en série sur l'axe de l'actionneur. Ils permettent de fournir des mesures nécessaires aux asservissements de vitesse et de position des axes du robot.

Dans notre cas La position est mesurée par un encodeur incrémental, cette technologie a été sélectionnée pour sa facilité d'intégration avec le moteur DC (montage en usine) ainsi qu'avec la carte d'acquisition.

II.5.2 Capteur d'effort:

Pour mesurer les forces d'interactions entre le robot esclave et l'environnement un capteur de couple de type FSR a été utilisé.

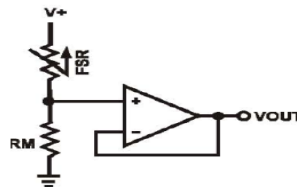
II.5.2.1 Capteur utilisé:

Nous avons utilisé un capteur de couple de type FSR 402 (Force Sensing Resistor), est un appareil à deux fil avec une résistance qui dépend de la force appliquée d'où son appellation : résistances à détection de force ou FSR.



Figure 19: capteur de couple (FSR 402)

Pour une conversion simple (force- tension), le dispositif de FSR est lié à une résistance de mesure dans une configuration de division de tension :



La sortie est décrite par l'équation:

$$V_{OUT} = \frac{R_M V_+}{(R_M + R_{FSR})}$$

La résistance de mesure R_M est choisie pour maximiser la plage de sensibilité de la force souhaitée et pour limiter le courant. Cette sensibilité de la force est optimisée pour une utilisation dans la commande tactile humaine de dispositifs électroniques tels que l'électronique automobile, les systèmes médicaux, et dans les applications industrielles et robotiques.

Dans la configuration représentée, la tension de sortie augmente avec une force croissante, comme montre cette figure ci-dessous :

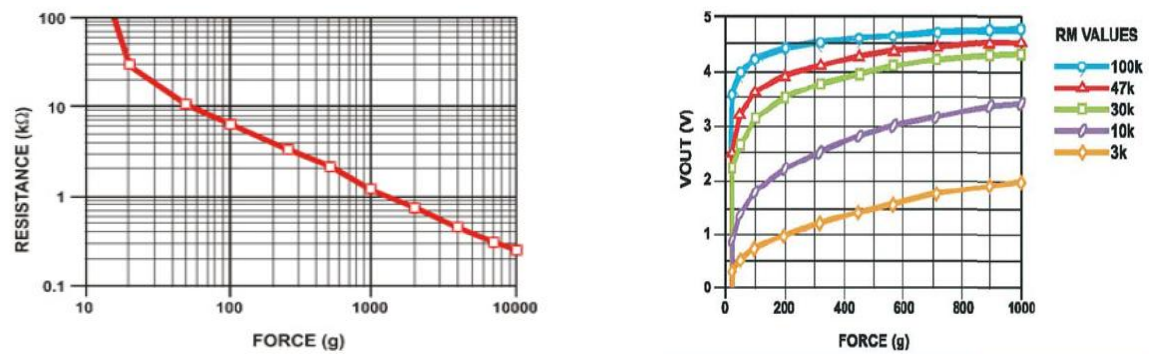


Figure 20: La tension de sortie en fonction de force appliquée.

II.5.2.2 Fonctionnalités :

- Force d'actionnement de 0,1N et gamme de sensibilité jusqu'à 10N
- Facilement personnalisable à une vaste gamme de tailles
- Économique
- Solide, jusqu'à 10 millions d'actionnements
- Simple et facile à intégrer

II.5.2.3 Caractéristiques techniques :

Type	FSR402
Résistance nominale	$1 \text{ m}\Omega < R < 3 \text{ k}\Omega$
Gamme de mesure	10 g - 10 kg
Modèle	moyen rond, surface active 12,7 mm
Hauteur	0.46 mm
(Ø)	14.68 mm

Figure 21: Caractéristique du capteur FSR 402

II.5.2.4 branchement du capteur à Arduino :

Le capteur est lié à notre carte Arduino comme suit :

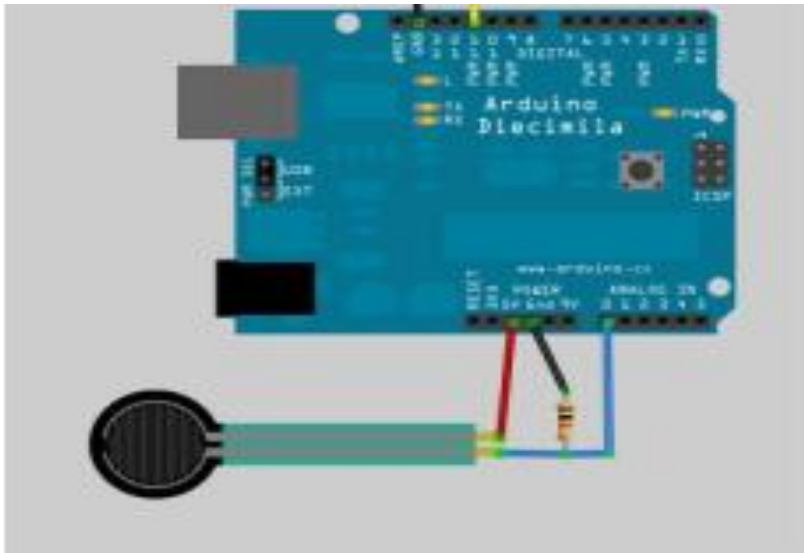


Figure 22: Schéma de branchement du capteur à Arduino.

II.6 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. (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. Et le nombre d'impulsion de son encodeur est 360 impulsions par tour. Il comporte 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 23 : Moteur réducteur avec encodeur de quadrature

II.7 Mise en œuvre

II.7.1 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'incrément angulaire 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.

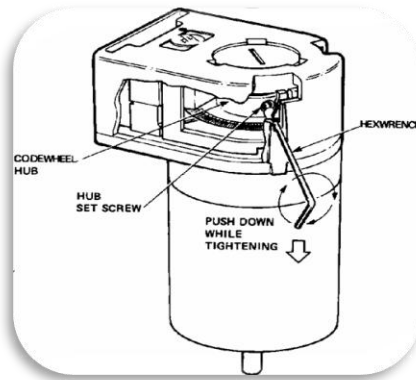


Figure 24 : Utilisation d'un encodeur optique sur l'axe du moteur

II.7.1.1 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.

Deuxième partie : Description du moteur à courant continu(MCC)

II.8 Principe de fonctionnement du moteur à courant continu : [6]

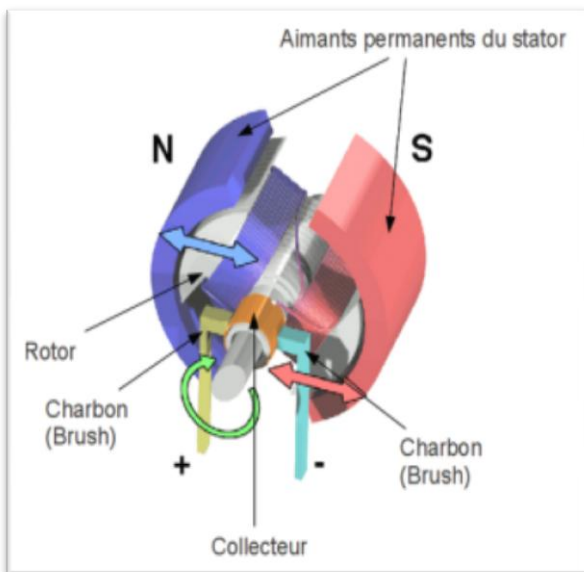


Figure 25 : Fonctionnement du MCC



Figure 26 : moteur à courant continu

Comme montre cette figure ci-dessus le stator du moteur crée un champ magnétique fixe, en robotique ce champ est crée par un aimant permanent .un système de balais permet d'amener le courant au rotor, ce courant (continu) crée un champ en quadrature par rapport à celui du stator générant ainsi un couple proportionnel à ce courant.

Historiquement, le MCC fut le premier moteur à être utilisé en robotique.

Son principal avantage est qu'il est facile à commander (couple proportionnel au courant d'induit).

Son principal inconvénient est qu'il utilise des contacts glissants (maintenance régulière)

II.9 Modélisation et Asservissement du moteur :

II.9.1 Modélisation du moteur :

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 :

1-Equation électrique : $U(t) = E(t) + Ri(t) + \frac{Ldi(t)}{dt}$

2-Equation mécanique : $J \frac{d\Omega(t)}{dt} = r(t) - f\Omega(t)$

3-Equations électromécaniques : $E(t) = K\Omega(t)$

$$r(t) = Ki(t)$$

U : Tension appliquée au moteur

E : Force contre électromotrice

i : Le courant de l'induit

Ω : Vitesse de rotation de l'axe

r : Couple moteur

f : Couple de frottement

J : Inertie du rotor

II.9.2 Asservissement du moteur :

A partir des équations électriques et mécaniques ci-dessus (relation de proportionnalité entre force contre électromotrice et vitesse de rotation ainsi la même relation avec le même coefficient entre le courant et le couple), on pourra établir le schéma bloc d'un MCC ayant en entrée $u(s)$ et en sortie $\Omega(s)$ on pourra voir clairement l'asservissement (un asservissement qui est réalisé où qui est due à la force contre électromotrice E , qui s'oppose à la tension qu'on met aux bornes de l'induit et qui est proportionnelle à la vitesse de rotation) qu'il ya dans la machine :

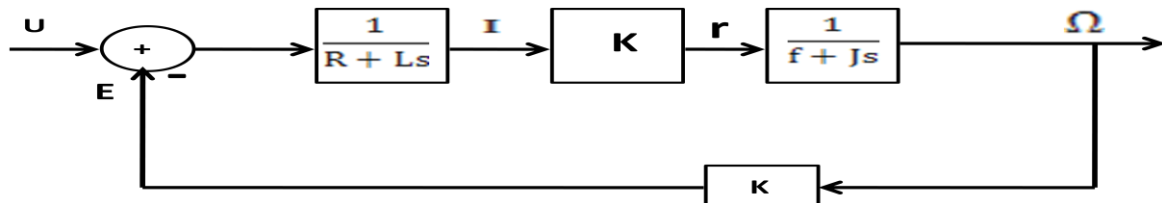


Figure 27 : Schéma bloc du moteur à courant continu (MCC)

On obtient :
$$\frac{\Omega(s)}{U(s)} = \frac{K}{LJs^2 + (RJ + Lf)s + Rf + K^2} \dots\dots\dots (II.1)$$

La fonction de transfert d'un moteur de robotique et, en général, de tout moteur utilisé dans un servomécanisme est une fonction de transfert du second ordre.

II.9.2.1 Asservissement couple/courant :

Ce qui est sympa pour un moteur à courant continu c'est que on' a pas besoin de faire un asservissement de couple, le couple est contrôlable très facilement : on a une relation linéaire entre le courant et le couple, naturellement ya déjà linéarisation du comportement mécanique de la machine, donc on veut se rendre maître du courant, le maîtriser parfaitement en temps réelle avec une rapidité qui correspond à la dynamique la plus rapide du système. Pour une raison technologique pourquoi se rendre maître du courant, le maîtriser à tout moment, tout simplement pour qu'il ne dépasse pas la valeur maximale et cette valeur est imposée par les composant de puissance tels que les MOSFET.....etc. c'est des composants qui sont relativement sensible à tout sur courant même très sensible, il suffit de dépasser une valeur nominale que ça entraine leur destruction.

Le correcteur $C(s)$ de l'asservissement de courant et en général du type PI :

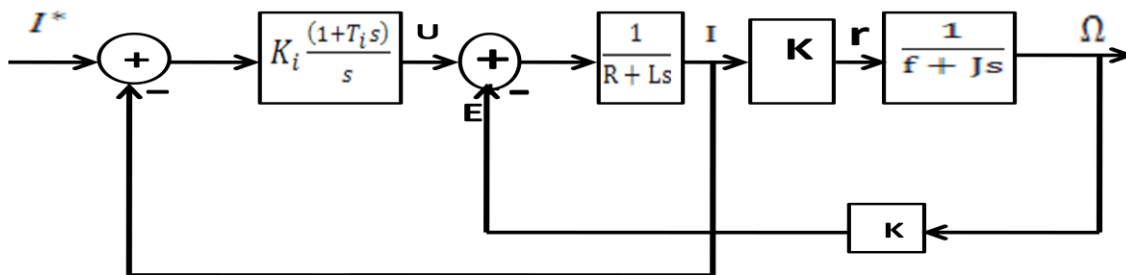


Figure 28 : Schéma bloc de l'asservissement du courant.

Tous les actionneurs de robotique ont une commande qui réalise asservissement de couple. Comme le couple est proportionnel au courant de l'induit, la fonction de transfert $F(s)$ entre $U(s)$ et $I(s)$ est donnée comme suit :

$$\frac{I(s)}{U(s)} = \frac{f + Js}{LJs^2 + (RJ + Lf)s + Rf + K^2} = F(s) \dots\dots\dots (II.2)$$

La fonction de transfert $H(s)$ entre I^* (la consigne de courant) et I (la valeur mesuré du courant), on a:

$$H(s) = K \frac{C(s)F(s)}{1 + C(s)F(s)} \dots\dots\dots (II.3)$$

$$\text{avec} \quad C(s)F(s) = \frac{K_i(1 + T_i s)(f + Js)}{s(LJs^2 + (RJ + Lf)s + Rf + K^2)} \dots\dots\dots (II.4)$$

$$\text{D'où on obtient} \quad \Rightarrow \quad H(s) = \frac{KK_i(1 + T_i s)(f + Js)}{LJs^3 + (RJ + Lf + K_i T_i J)s^2 + (Rf + K^2 + J + T_i f)s + K_i f} \dots\dots\dots (II.5)$$

La fonction de transfert entre la consigne de courant et le couple moteur est de degré 3. La fonction de transfert entre la consigne de courant et la vitesse du rotor est donc de degré 3.

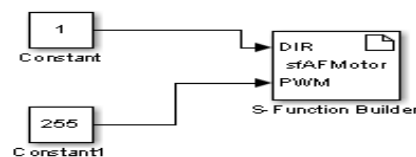
II.10 La technique de la PWM pour La variation de vitesse :

Un moteur à courant continu possède une relation directe entre sa tension d'alimentation et sa vitesse de rotation. En effet, plus la tension à ses bornes est élevée et plus son axe tournera rapidement. Nous sommes en mesure de produire à l'aide de microcontrôleur un signal carré dont le rapport cyclique est variable. Et grâce à cela, nous obtenons une tension moyenne (comprise entre 0 et 5V) en sortie de la carte Arduino. En utilisant la PWM, on va générer une tension par impulsions plus ou moins grandes.

La technique de la MLI (modulation de largeur d'impulsion) appelée en anglais PWM, me permettra donc de générer une tension par impulsions plus ou moins grandes, Ce signal va commander le transistor ou le relais qui va à son tour commander le moteur. Le moteur va donc être alimenté par intermittences à cause des impulsions de la PWM. Ce qui aura pour effet de modifier la vitesse de rotation du moteur en modifiant le rapport cyclique de la PWM. Plus il est faible (un état BAS plus long qu'un état HAUT), plus le moteur ira doucement. Inversement, plus le rapport cyclique sera élevé (état HAUT plus long que l'état BAS), plus le moteur ira vite.

II.10.1 Pilotage du sens et vitesse du moteur :

Avant de s'intégrer dans la télé opération c'est-à-dire dans la commande de deux moteurs (le 1^{er} moteur est commandé en force et le 2^{ème} en position) il faut d'abord commander (sens et vitesse) chaque moteur seul pour voir s'ils fonctionnent correctement ou non, ainsi pour visualiser les signaux moteurs. Pour ce faire nous avons intégré la librairie AFMotor de Arduino à MATLAB, toute en créant une nouvelle fonction sur SIMULINK qui nous permettra d'exécuter le programme de pilotage du sens et de vitesse de rotation du moteur comme suit :



Pour la direction (le sens) :

Entrées	Déplacements
1	Gauche
2	Droite
3	Relâcher

Pour la vitesse :

Entrées	Rotation
0	Vitesse nulle
255	Grande vitesse
[0..... 255]	Variation de vitesse

- Résultats expérimental :

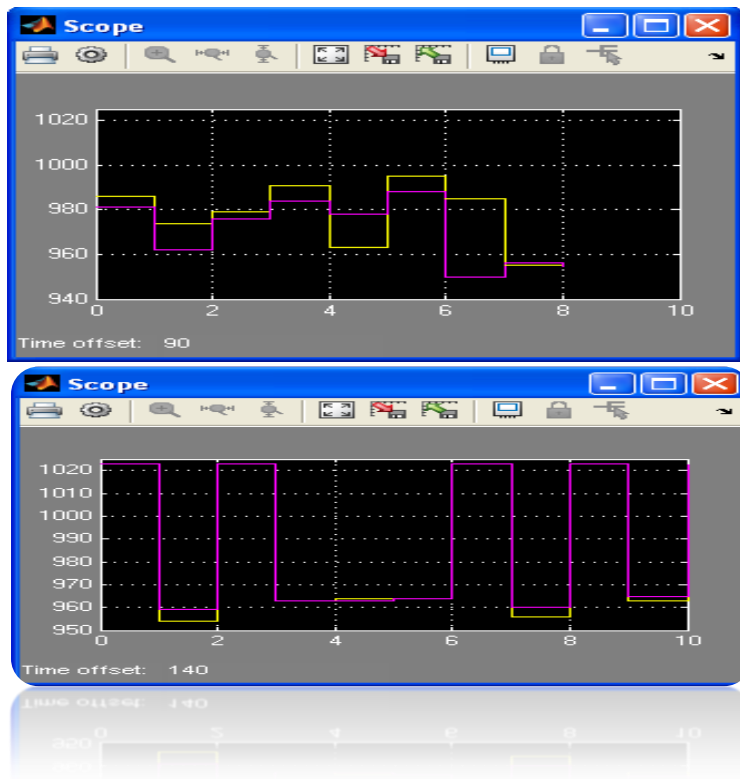


Figure 29 : Signaux moteurs (coté maître et esclave)

- On remarque bien que les signaux des deux moteurs sont des signaux carrés, pour pouvoir commander les deux moteurs (les robots : Maître/Esclave), on utilise la technique de l'encodeur incrémental, pour ce la nous avons intégré la librairie de l'encodeur à Arduino (Donc lorsque j'applique une force au robot maitre ou plus précisément lui donner une position j'aurais une valeur numérique de cette position souhaitée), et c'est ce qu'on va voir dans le chapitre IV.

II.11 Conclusion

Dans ce chapitre nous avons présenté les différents composants utilisés pour la réalisation de la maquette, ainsi présenter le principe de fonctionnement de tout actionneurs, capteurs utilisés dans la réalisation, à la fin nous avons donné un exemple sur MATLAB/SIMULIK et Arduino dont le but de comprendre la technique de la PWM pour la commande des moteurs, tout en les testons indépendamment chaque moteur seul afin d'avoir une idée précise sur le fonctionnement du moteur et quels sont les moyens de le piloter.

Chapitre

III

:

III.1 Introduction

Avant de commencer à intégrer dans ce chapitre qui est la programmation avec Arduino, il ya une idée importante qu'il est bon d'avoir à l'esprit, est bien le type de moteur à utiliser (moteur réducteur avec encodeur). C'est bien cet encodeur qui nous permettra d'aboutir sur la programmation de la commande des deux moteurs, dont le 1^{er} moteur (robot maître) est asservi en force, et le 2^{ème} moteur (robot esclave) est asservi en position. Pour cela nous avons intégré une nouvelle librairie qui est la bibliothèque de « Rotary Encoder » à Arduino pour commander l'encodeur et donc commander le moteur. En outre le système Arduino permet de faire un pont entre le monde virtuel de la programmation informatique (concepts formels et logiques) et le monde physique (interaction électromécanique des objets).

Le système Arduino se compose essentiellement de deux composants principaux qui sont le matériel et le logiciel, soit :

- la carte électronique Arduino qui s'occupe de l'acquisition et du traitement des signaux venant de l'encodeur optique, de la communication entre les deux parties à savoir, la partie commande (ordinateur) et les deux sites (maître et esclave).
- programme Arduino.

III.2 Présentation Arduino : [7]

Arduino est un projet électronique qui à vue le jour en Italie en 2005, Il a été créé et perfectionné par une équipe de développeurs, composée de six individus: (Massim Banzi, David Cuartielles, Tom Igoe, Gianluca Martino, David Mellis et Nicholas Zambetti). Cette équipe a créé ce qu'on appelle le "système Arduino". C'est un outil qui va permettre aux débutants, amateurs ou professionnels de créer des systèmes électroniques plus ou moins complexes. En outre ce système permet de faire le pont entre le monde virtuel de la programmation informatique (concepts formels et logiques) et le monde physique (interaction électromécanique des objets).

Arduino est donc une plate-forme open-source d'électronique programmées qui est basé sur une simple carte à microcontrôleur, et un logiciel, véritable environnement du développement intégré, pour écrire, compiler et transférer le programme vers la carte à microcontrôleur. Ces cartes sont basées sur une interface entrée /sortie simple pouvant recevoir des entrées d'une grande variété d'interrupteurs ou de capteurs, et pouvant contrôler une grande variété de lumières, moteurs ou tout autre types d'actionneurs et tous cela se fait sur un environnement de développement proche du langage C.

III.3 Avantage de l'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 ATMEGA8, ATMEGA168, ATMEGA 328, etc. Les schémas des modules sont publiés sous une licence Créative 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.

III.4 Les outils de l'Arduino :

Le système Arduino se compose essentiellement de deux composants principaux qui sont le matériel et le logiciel, soit la **carte électronique Arduino** et son **programme Arduino**.

III.4.1 Architecture HARDWARE de la carte :

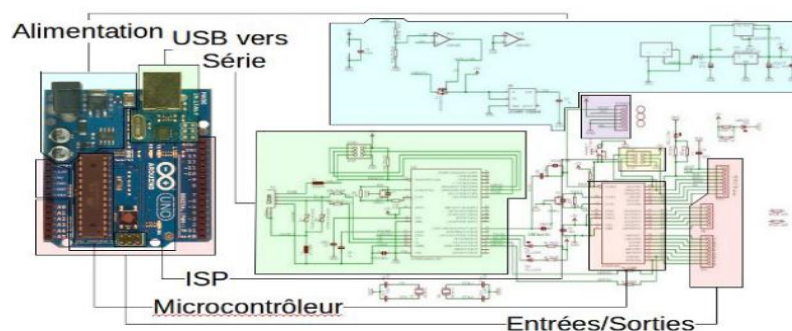


Figure 30 : Architecture interne de la carte Arduino méga 2560.

III.4.2 Architecture de la carte

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

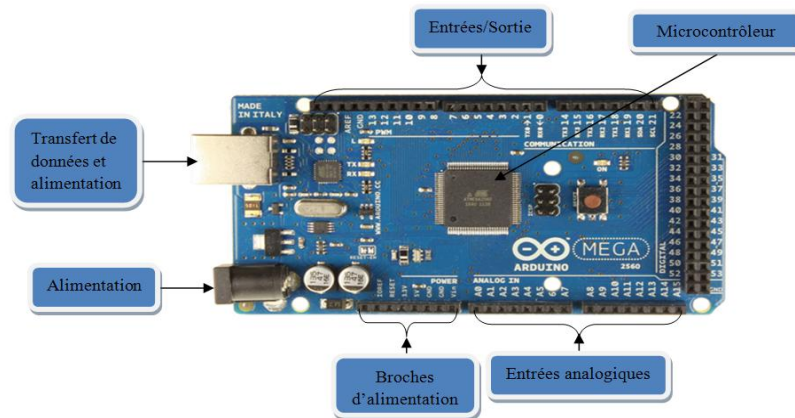


Figure 31: Architecture de la carte Arduino méga 2560

Les éléments de base qui composent la carte Arduino sont :

III.4.3 Le microcontrôleur

Un microcontrôleur est un petit processeur informatique relié à des entrées et des sorties numériques (0 ou 1) ou analogiques (tension variable). Il est capable de mémoriser et d'exécuter un programme visant à interpréter les entrées pour agir sur les sorties. Il se programme en général à l'aide d'un ordinateur (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) mais il peut fonctionner aussi de manière autonome.

III.4. 3.1 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.
- 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.
- Puissance sur la remise et la détection programmable d'arrêt partiel.

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

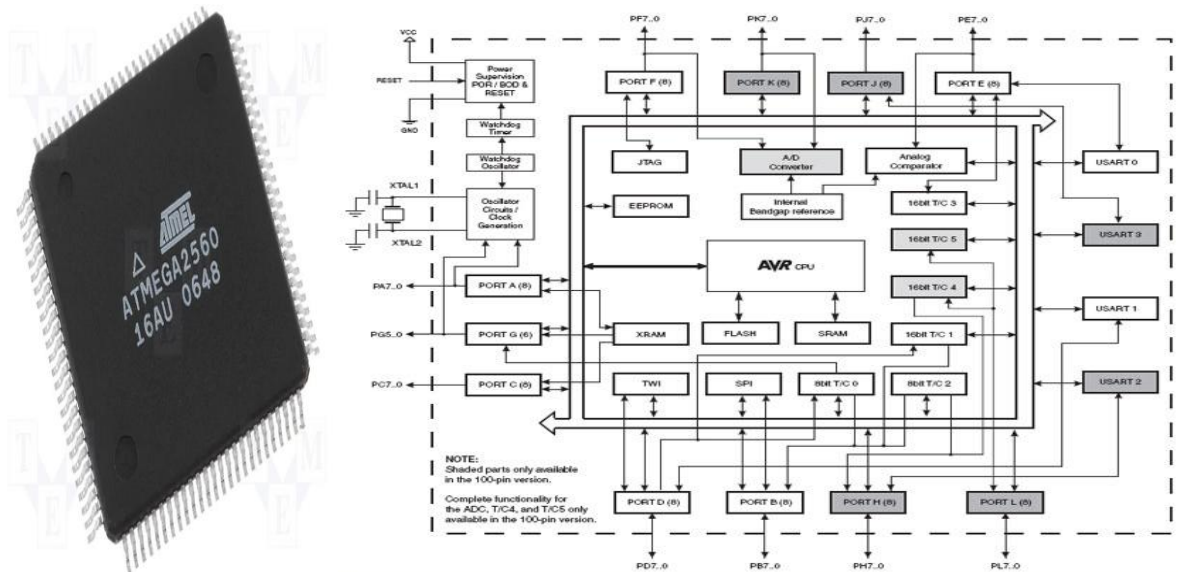


Figure31: Architecture du microcontrôleur ATMEGA 2560

III.4.4 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 entrées/sorties de l'Arduino sont les éléments avec lesquels nous serons le plus souvent en contact. En effet, c'est par leur intermédiaire que le microcontrôleur communique avec le monde extérieur.

- Les pins utilisés en numérique 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 en mode tout ou rien.
- Les pins utilisés en analogique ont une sortie de valeur (0 à 255)

III.4.5 Mémoire :

L'ATmega2560 qui équipe l'Arduino Mega 2560 dispose de trois types de mémoires :

1. La mémoire flash d'une capacité 256 Ko pour le programme, analogue à celle que l'on trouve par exemple dans les clés USB. une fois programmée, cette mémoire Flash conserve son contenu indéfiniment, même lorsque la carte n'est pas alimentée.
2. La mémoire vive ou RAM, analogue dans son principe à la mémoire vive du PC, mais en plus petite quantité puisque l'on ne dispose ici de 8 Ko (contre plusieurs Go dans un PC par exemple). Cette mémoire est généralement utilisée pour les variables employées dans les programmes, pour stocker des résultats temporaires lors de calculs, etc. elle présente la particularité de pouvoir être lue et écrite à tout instant par le microcontrôleur mais elle oublie son contenu dès qu'il n'est plus alimenté. Il est donc hors de question de l'utiliser pour stocker des informations qui devront être conservées

lorsque la carte ne sera pas alimentée. pour cela l'ATmega 2560 dispose également de mémoire EEPROM.

3. La mémoire EEPROM qui se voit doter de 4 Ko, dans lequel le microcontrôleur peut lire à tout instant. Il peut aussi y écrire et effacer son contenu avec plus de facilité.

III.4.6 L'alimentation

L'alimentation qui assure la distribution d'énergie aux différents composants de la carte Arduino, peut être effectuée via :

-Une connexion USB de l'ordinateur qui fournit toujours directement du 5v (la tension de fonctionnement).

-par la prise d'adaptateur qui fournit une tension comprise entre 7 et 12 v c'est ce qu'on appelle la tension recommandée (la tension est ensuite convertie toute seul en 5v par arduino).

Ainsi La carte peut fonctionner sur une alimentation externe de 6 à 20 volts c'est ce qu'on appelle la tension limite. 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.

III.4.7 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.

III.4.8 La connectique

La carte Arduino ne possédant pas de composants (résistances, diodes, moteurs...) qui peuvent être utilisés pour un programme, mis à par 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.

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

III.5 L'architecture SOFTWARE de la carte :

La carte Arduino présente le noyau de notre système, et pour qu'on puisse la programmer on doit disposer du logiciel compatible avec cette carte, ce logiciel est **Arduino EDI (Espace de développement intégré)**.

III.5.1 Présentation de l'Espace de développement Intégré (EDI) Arduino :

Comme déjà dit La carte Arduino présente le noyau de notre système, elle est programmée par un logiciel compatible appelé **Arduino EDI** (Espace de développement intégré) qui est une application écrite en Java inspiré du langage Processing.

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

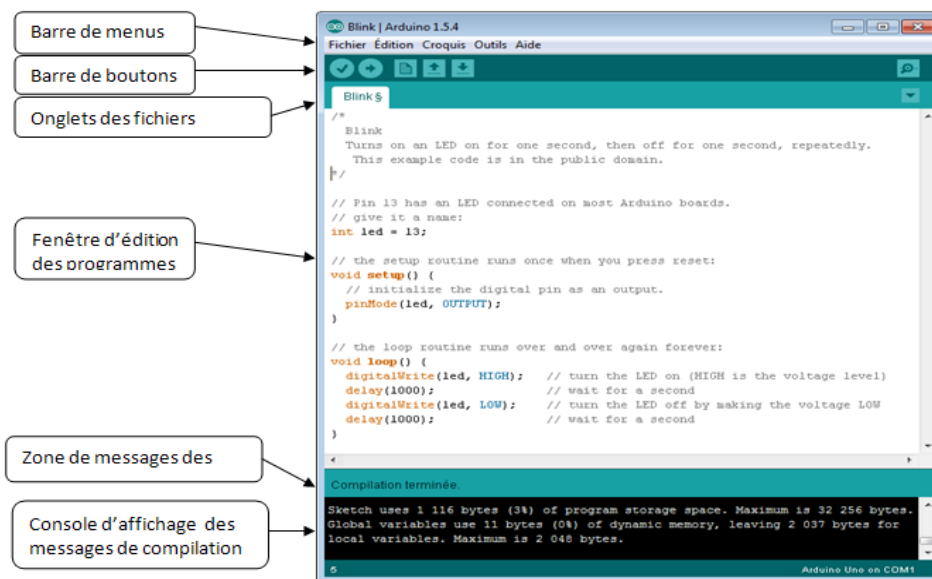
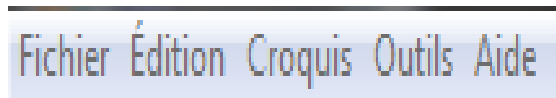


Figure 33 : les différentes parties de la fenêtre principale du logiciel Arduino.

III.5.2 Différentes composition de l'espace de développement intégré (EDI) :

III.5.2.1 Une barre de menus :



Cette barre contient les icones suivantes :

- ✓ **File (Fichier)** : ce menu contient les différentes options de créations, d'ouverture' de sauvegarde, d'impression du programme, ou l'ouverture d'un exemple parmi les exemples qui accompagnent le logiciel Arduino.
- ✓ **Edite (Editer)** : ce menu contient les options de copier/coller, sélection, et les options de recherche.
- ✓ **Sketch (Programme ou séquence)** : ce menu contient les différentes fonctions de la barre des boutons, ainsi que les options d'ajouts de librairies ou de fichiers.

- ✓ **Tools (Outils) :** c'est dans ce menu qu'on sélectionne le type de carte à programmer, et le port série utilisé.
- ✓ **Help (Aide) :** ce menu est fait pour donner de l'aide concernant les différents problèmes rencontrés au niveau du logiciel Arduino.

III.5.2.2 Une barre des boutons :



Vérifier/compiler : Vérifie le code à la recherche d'erreur.



Transférer vers la carte : Compile votre code et le transfère vers la carte Arduino.



Nouveau : Crée un nouveau code (ouvre une nouvelle fenêtre)



Ouvrir : Ouvre la liste de tous les programmes dans le "livre de programmes".

Enregistrer : Enregistre un programme.



Moniteur Série : Ouvre la fenêtre du moniteur (ou terminal) série.

III.5.2.3 Structure d'un programme Arduino :

- ✓ **Un éditeur:**
C'est l'espace utilisé pour écrire le programme à réaliser, comme il dispose aussi des onglets de navigation.
- ✓ **Une zone de messages :**
Comme son nom l'indique cette espace est utilisée par l'EDI pour communiquer à l'utilisateur des informations sur l'état du programme et des actions en cours.
- ✓ **Une console texte :**
Qui affiche les messages concernant le résultat de la compilation du programme (il indique si le programme comporte des erreurs).

III.5.2.4 Le programme Arduino :

III.5.2.4.1 Description de la structure d'un programme Arduino :

Le langage Arduino est basé sur les langages C/C++.

Un programme utilisateur Arduino est une suite d'instructions élémentaires sous forme textuelle, ligne par ligne. La carte lit puis effectue les instructions les unes après les autres, dans l'ordre défini par les lignes de code, comme lors d'une programmation classique. Cette structure se décompose en trois parties :

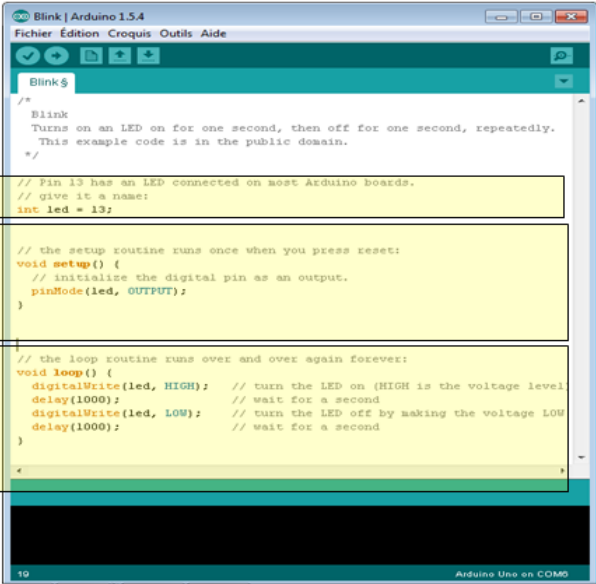
	
Définition des constantes et variables globales	<pre> // Pin 13 has an LED connected on most Arduino boards. // give it a name: int led = 13; </pre>
Fonction principale : <i>VOID SETUP()</i> Initialisation des ressources de la carte Configuration des entrées/sorties Définition de la vitesse de fonctionnement	<pre> // the setup routine runs once when you press reset: void setup() { // initialize the digital pin as an output. pinMode(led, OUTPUT); } </pre>
Fonction boucle : <i>VOID LOOP()</i> Description du fonctionnement général du programme Gestion des interactions entre les entrées/sorties Cette partie sera exécutée en boucle	<pre> // the loop routine runs over and over again forever: void loop() { digitalWrite(led, HIGH); // turn the LED on (HIGH is the voltage level) delay(1000); // wait for a second digitalWrite(led, LOW); // turn the LED off by making the voltage LOW delay(1000); // wait for a second } </pre>

Figure 34 : les différentes composantes du programme Arduino.

Le code minimal :

Avec Arduino, nous devons utiliser un *code minimal* lorsque l'on crée un programme. Ce code permet de diviser le programme que nous allons créer en deux grosses parties, soit void setup() et void loop() décrites en haut.

III.5.2.4.2 Le jeu d'instructions du langage Arduino :

1. LES SYNTAXES DU PROGRAMME :

- Chaque instruction se termine par un « ; » ;
- Les **accolades** « { » et « } » sont les "conteneurs" du code du programme. Elles sont propres aux fonctions, aux conditions et aux boucles. Les instructions du programme sont écrites à l'intérieur de ces accolades ;
- Les **commentaires** sont des lignes de texte incluses dans le programme et qui ont pour but de vous informer vous-même ou les autres de la façon dont le programme fonctionne. Ces lignes ajoutées sont ignorées par le compilateur. Les commentaires sont précédés des caractères « // » ou bien encadrés par « /* » et « */ » ;

- Il est formellement interdit de mettre des accents en programmation, sauf dans les commentaires.
- Un **nombre en binaire** doit être précédé de la lettre « **B** » ;
- Un **nombre** écrit en **hexadécimal** doit être précédé par les caractères « **0x** ».

2. Commandes du programme Arduino :

❖ Structure générale

- **void setup()** (configuration-préparation)
- **void loop()** (exécution)

❖ Contrôle et conditions

- **if** (si...)
- **if...else** (si...alors...)
- **for** (pour...)
- **switch case** (dans le cas où...)
- **while** (pendant que ...)

❖ Opérations de comparaison

- **==** (équivalent à)
- **!=** (différent de)
- **<** (Inférieur à)
- **>** (Supérieur à)
- **<=** (inférieur ou égal à)
- **>=** (supérieur ou égal à)

❖ Opérations booléennes

- **&&** (Et)
- **||** (ou)
- **!** (Et pas)

❖ Autres commandes

- **//** (Commentaire simple ligne)
- **/* */** (Commentaire multi-lignes)
- **#define** (donner une valeur à un nom)

❖ Variables

- **char** (variable 'caractère')
- **int** (variable 'nombre entier')
- **long** (variable 'nombre entier de très grande taille')

- **string** (variable 'chaîne de caractères')
- **array** (tableau de variables)

Variables

❖ Niveaux logiques des connecteurs numériques

- **HIGH** (état 1)
- **LOW** (état 0)
- **INPUT** (configuré en entrée)
- **OUTPUT** (configuré en sortie)

Fonctions

❖ Entrées-sorties numériques

- **pinMode** (**broche**, état) (configuration des broches en entrée ou en sortie)
- **digitalWrite** (**broche**, état) (écrire un état sur une broche numérique.)
- **digitalRead** (**broche**) (lire un état sur une broche numérique.)
- **unsigned long pulseIn** (**broche**, état) (lire une impulsion sur une broche numérique.)

❖ Entrées analogiques

- **int analogRead** (**broche**) (lire la valeur d'une broche analogique.)
- **analogWrite** (**broche**, valeur) (PWM : écrire une valeur analogique sur les broches 5, 6, 9, 10 et 11)

❖ Gestion du temps

- **unsigned long millis()** (temps de fonctionnement du programme)
- **delay**(ms) (attente, en millisecondes)
- **delayMicroseconds**(us) (attente, en microsecondes).

III.5.2.4.3 Les étapes à suivre pour programmer la carte :

1. L'écriture du programme :

Cette première étape consiste à saisir le programme.

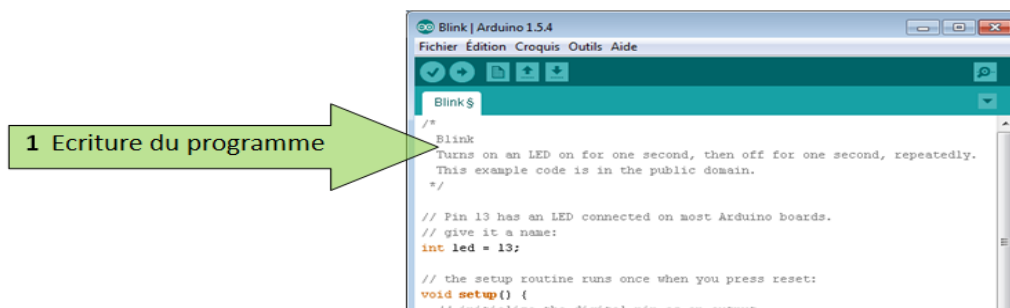


Figure 35 : l'écriture du programme

2. Compilation du programme :

Dans cette partie, on vérifie si le code contient des erreurs de syntaxes. En cas d'anomalie dans le programme, le compilateur nous renseigne sur le type d'erreur et la ligne où elle se trouve. (Pour la correction des erreurs il suffit de consulter l'aide disponible dans la barre de menus).

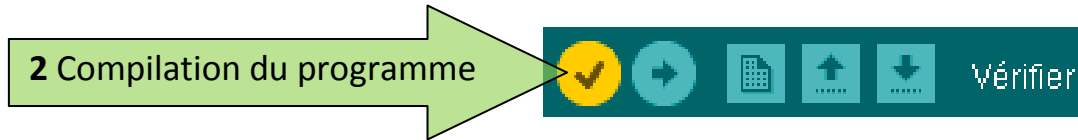


Figure 36 : compilation du programme

Si aucune erreur n'est détectée, le programme est compilé et le message «Compilation terminée» apparaît, suivi de la taille du programme.

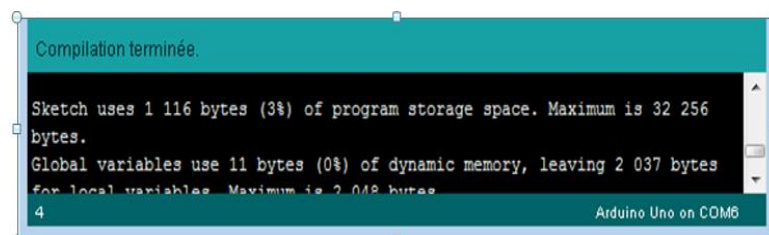


Figure 37 : console d'affichage

3. Sélection de la cible (type de carte) :

Avant de transférer le programme vers la carte Arduino, il faut sélectionner le type carte Arduino (cible) depuis le menu **Outils>Board**.

La carte doit évidemment être connectée à l'ordinateur via un câble USB.

Vérifions que c'est bien le nom " Arduino Méga 2560" qui est coché. Si ce n'est pas le cas, on le coche.

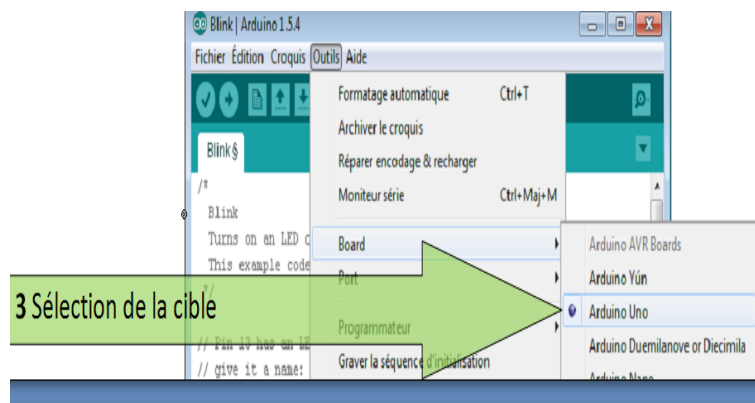


Figure 38 : Sélection de la cible

4. Sélection du port :

Sélectionner le port série sur lequel est branché cette carte (depuis le menu **Outils > Port**).

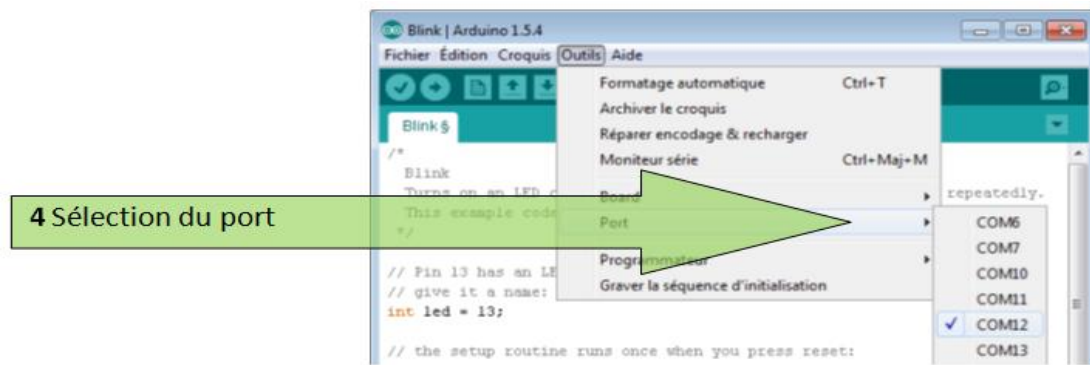


Figure 39 : Sélection du port

5. Transfert du programme :

Cela se fait en cliquant sur le bouton.

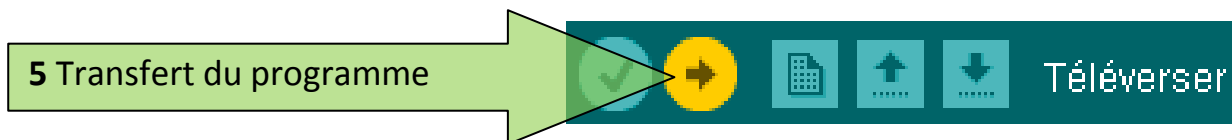


Figure 40 : Transfert du programme

Le programme est téléchargé vers la carte, le message « télé versement terminé » s'affiche une fois l'opération terminée

Sur la plupart des cartes, nous devons voir les LEDs des lignes RX et TX clignoter rapidement, témoignant que le programme est bien transféré. Durant le transfert, le bouton devient jaune et le logiciel Arduino affiche un message indiquant que le transfert est en cours

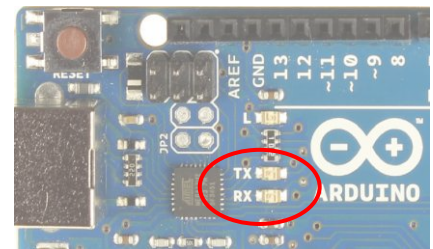


Figure 41 : clignotement des deux LED du port série.

III.6 Conclusion

Dans ce chapitre, notre étude s'est focalisée sur la présentation de la carte (Arduino), tout en commençant par un bref historique et quelques détails concernant la création et l'utilité de cette carte. Puis nous avons parlé de son côté matérielle, enfin présenter le logiciel et le langage de programmation

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

Chapitre

IV

IV.1 Introduction :

Réaliser un système quel que soit sa complexité est bien c'est la tâche la plus délicate à faire, vue comme disaient les chercheurs « la preuve de la connaissance est l'expérience ». Donc il faut combiner entre la théorie et la pratique pour pouvoir faire un système performant qui aboutira au cahier de charge souhaité. Dans ce chapitre qui consiste à la réalisation d'une architecture de télé opération avec retour d'efforts, nous présenterons la maquette réalisée, avec ses différents composants ainsi illustrer les différentes techniques utilisées pour la commande des deux moteurs (robots Maître/ Esclave).

IV.2 Simulation du pont H sur ISIS protues :

Nous avons utilisé un pont H double : deux ponts H en parallèle (8 transistors : 4 transistors pour chaque pont) :

Justification :

- Le 1^{er} pont H pour le moteur 2 (robot esclave), pour qu'il tourne dans les deux sens suivant la position donnée par le robot maître qui est le moteur 1.
- Lorsque le 2^{ème} moteur (robot esclave) rencontre un obstacle, l'asservissement de position de ce dernier commute automatiquement en asservissement de force, cette force est ramenée au robot maître comme une contre-réaction. (autrement dit : en cas d'obstacle le robot esclave devient lui-même le maître donc il faut prévoir un autre pont H afin d'exécuter cette commande pour actionner un robot qui doit s'appeler l'esclave).

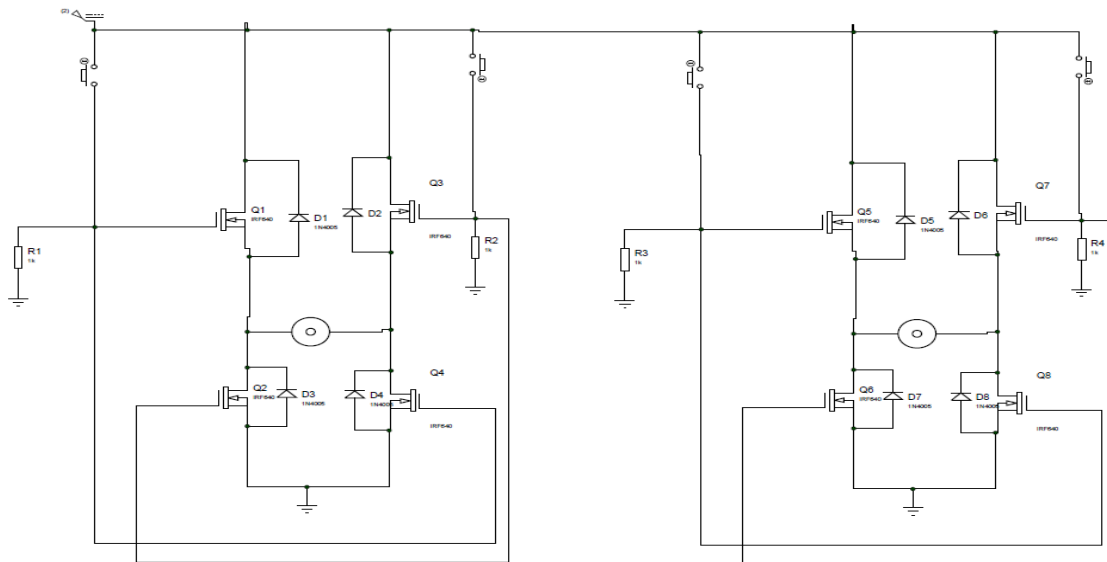


Figure 42: Schéma de simulation du pont H sur ISIS PROTEUS

IV.3 Schéma électrique du pont H :

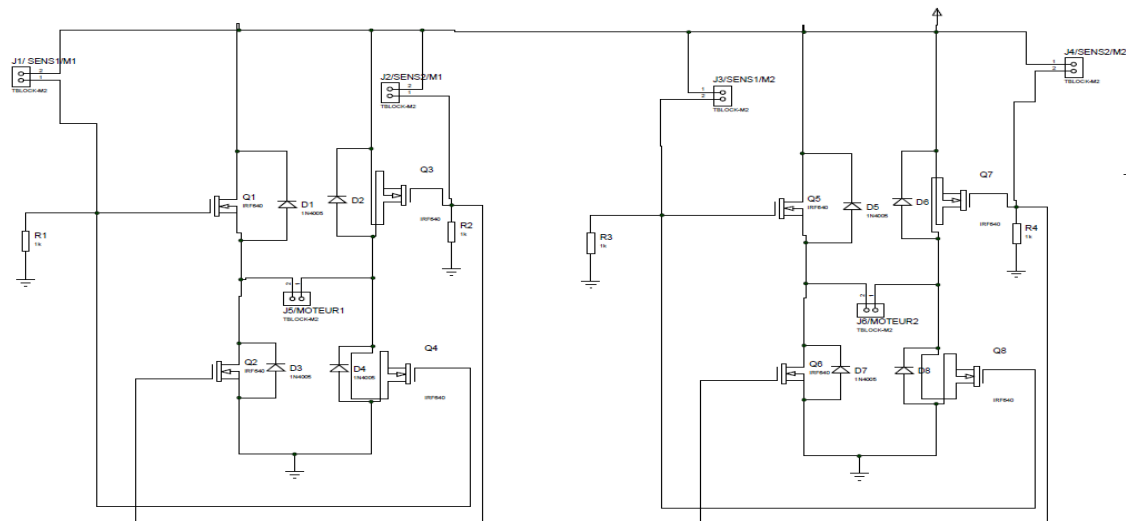


Figure 43: Schéma électrique du pont H.

IV.4 Schéma de réalisation :

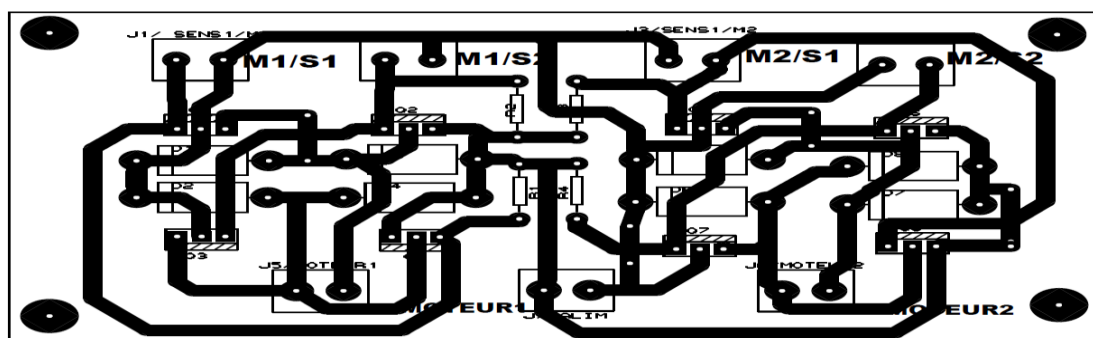


Figure 44: Schéma de la réalisation de la carte de puissance.

IV.5 Carte de puissance avec le pont H utilisé :

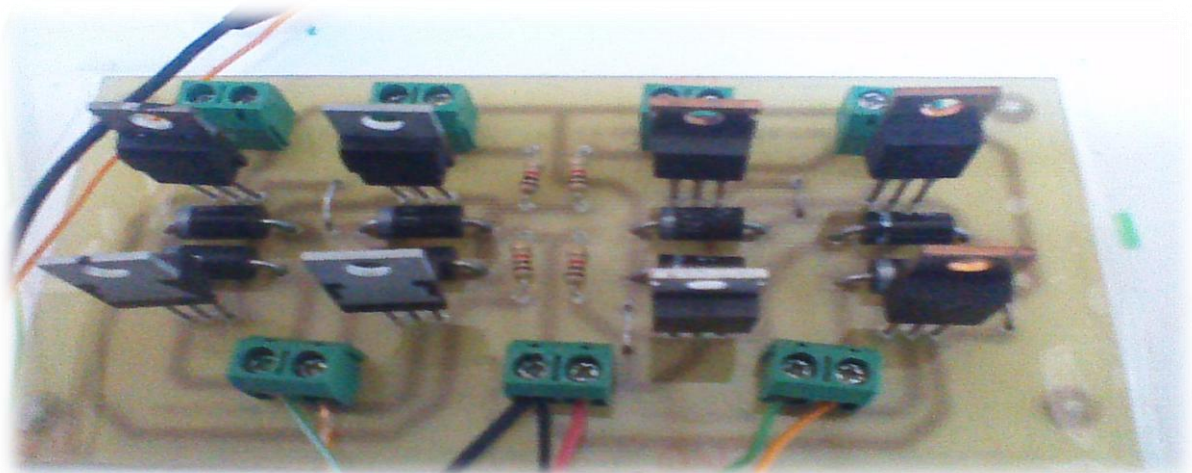


Figure 45: Carte de puissance réalisée

IV.6 moteur coté esclave (capteur de couple):

On a le robot esclave doté d'un capteur d'effort placé sur le levier comme suit :

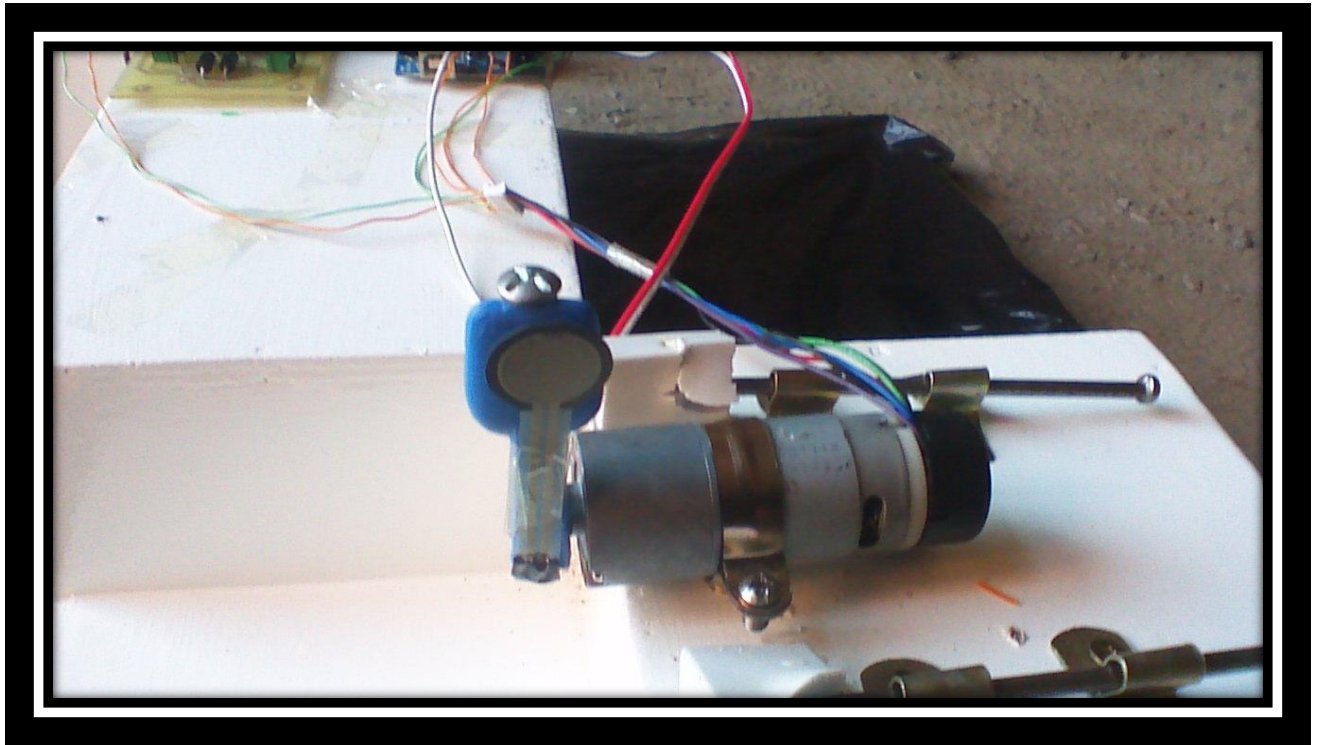


Figure 46: le capteur de couple (FSR 402) coté esclave

IV.7 Commande de deux moteurs (télé opération : Force/position) avec Arduino:

La commande des deux moteurs :

- Le 1^{er} moteur (robot maître) est commandé en force (la force appliquée par l'environnement sur le robot esclave) au moyen d'un capteur de couple, et le 2^{ème} moteur (robot esclave) est commandé en position (la position du robot maître envoyé comme consigne au robot esclave) au moyen d'un capteur de position (encodeur).
- J'ai utilisé un programme sur Arduino de façon que le 1^{er} moteur (le robot maître) lorsque on lui applique une force c'est-à-dire on lui donne une position souhaitée, le robot esclave reproduit les mouvements de manière accomplir la tâche demandée. En cas ce dernier rencontre un obstacle, le robot maître doit savoir que ce dernier est bloqué et si on essaye de le bouger il maintien toujours sa position initiale jusqu'à où le robot esclave sera libéré.
- Lorsque sous l'action d'une force constante de l'opérateur le robot esclave rencontre un obstacle, le centre de compliance continue à se déplacer et la force exercée sur l'environnement croit régulièrement il faut donc prévoir un système qui sature la consigne de déplacement lorsqu'une force maximal est détecter.
- les fonctions utilisées sont :

Encoder. Readencoder() : pour lire la valeur de la position souhaitée.

ValeurFSR=analogRead(capteurFSR) : pour lire la valeur de la force exercée.

Tourner MD : pour pouvoir déplacer dans le sens positive (M et D : indices relatifs aux mots Maître, Droit).

Tourner MG : pour pouvoir déplacer dans le sens négatif (M et G : indices relatifs aux mots Maître, Gauche).

Tourner EG : pour pouvoir déplacer dans le sens positive (E et G : indices relatifs aux mots Esclave, Gauche).

Tourner ED : pour pouvoir déplacer dans le sens négatif (E et D : indices relatifs aux mots Esclave, Droit).

Stop moteur : c'est une procédure pour éviter un court circuit car par défaut tout les pins sont alimentés à la fois.

IV.8 Les résultats obtenus :

```
COM7
|
encodeur 1 =1
encodeur 2 =1
encodeur 1 =2
encodeur 2 =2
encodeur 1 =3
encodeur 2 =3
encodeur 1 =4
encodeur 2 =4
encodeur 1 =5
encodeur 2 =5
encodeur 2 =6
encodeur 1 =6
encodeur 1 =7
encodeur 2 =7
```

Figure 47: Les valeurs données par l'encodeur (coté maître et esclave).

```
COM7
|
14
16
24
28
37
45
58
65
71
76
83
95
105
109
118
119
126
127
```

Figure 48: valeurs données par le capteur FSR coté esclave

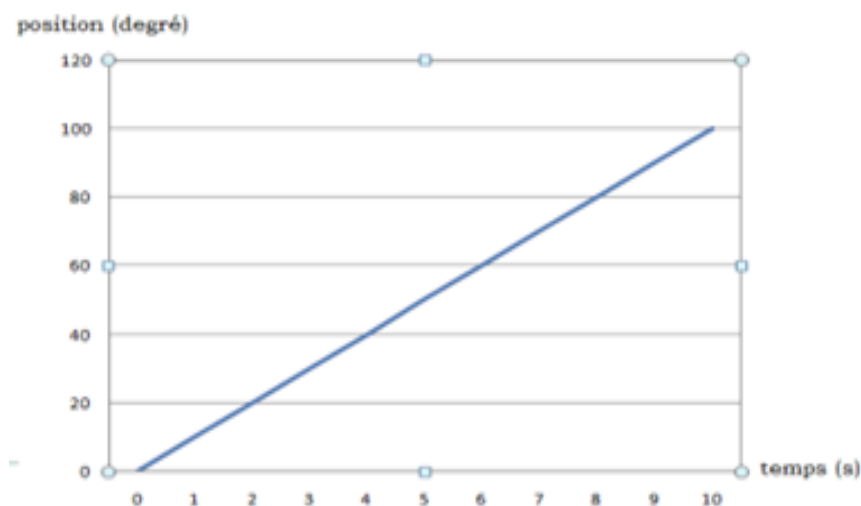


Figure 49: position du robot maître en fonction du temps

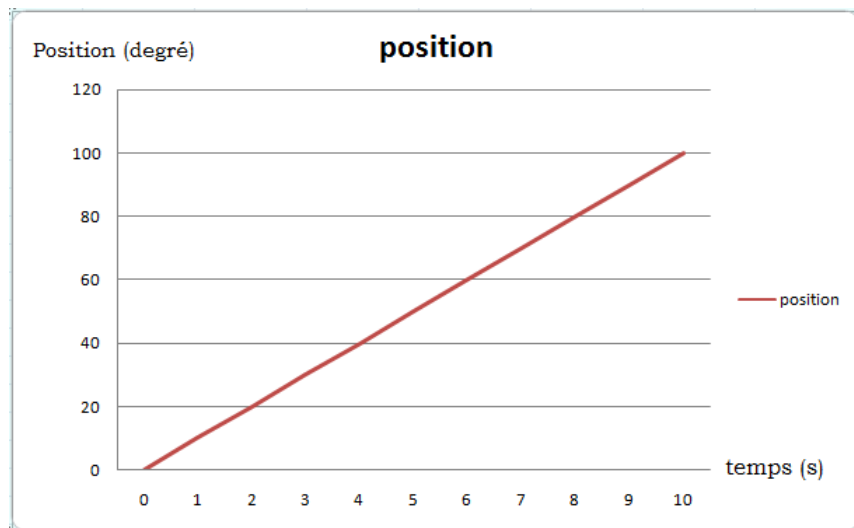


Figure 50: position du robot esclave en fonction du temps.

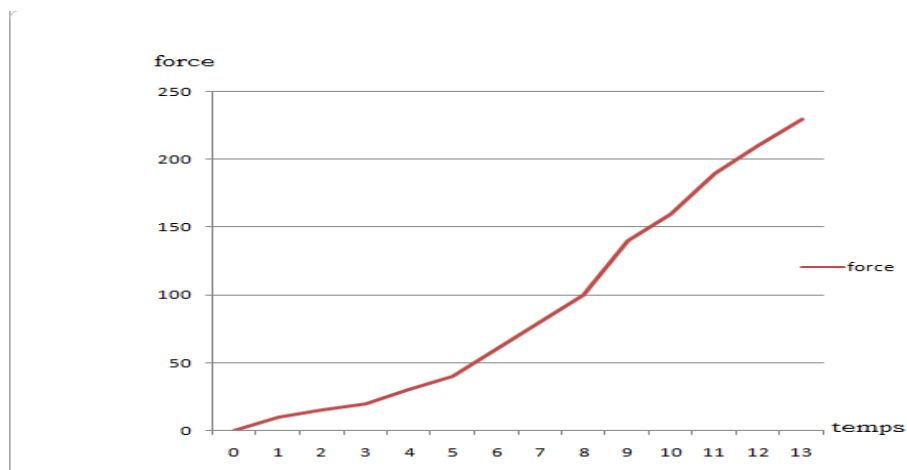


Figure 50: Force appliqué sur le robot esclave en fonction du temps

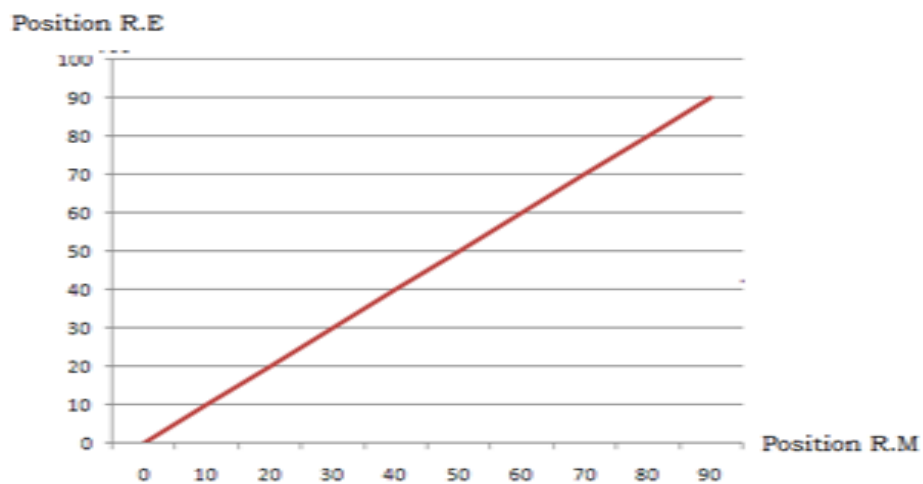


Figure 52: Position du robot esclave an fonction de la position du robot maître

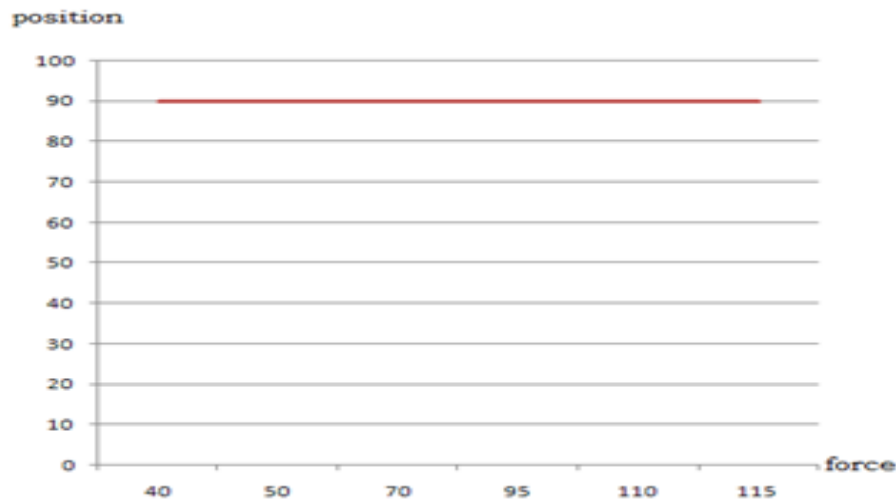


Figure 53: Position du robot maître en fonction de la force de l'environnement.

Interprétation des résultats obtenus :

On constate bien que le robot esclave suit bien le robot maître en position, selon la position donnée au robot maître l'encodeur 1 capte cette dite valeur et la transmet pour l'encodeur 2 et donc au robot esclave (asservissement en position). En cas d'obstacle (valeur indiquée par le capteur de couple côté esclave), l'asservissement de position de ce dernier commute automatiquement en asservissement de force, celle-ci est ramenée au robot maître comme une contre-réaction, ce qui oblige le robot maître à rester dans la position initiale comme montre la figure 53.

IV.9 La maquette réalisée :

Notre maquette réalisée est donné par cette figure ci-dessous :

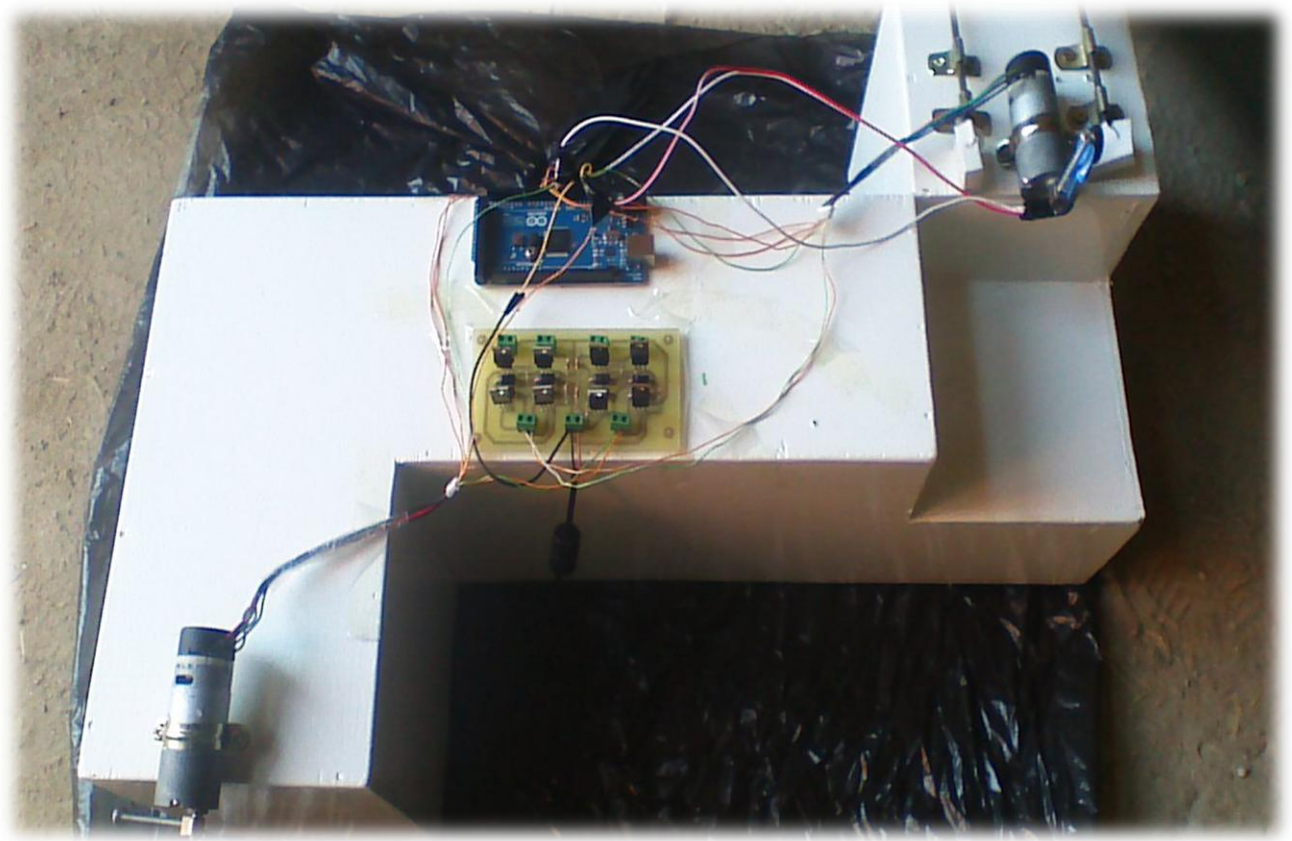


Figure 53: La maquette réalisée

IV.10 Conclusion :

Dans ce chapitre final nous avons présenté notre réalisation, ainsi les différents schémas utilisés et quelques fonctions pour la programmation de la commande, et à la fin nous avons détaillé les résultats obtenus lors de la simulation pour bien comprendre l'asservissement force et position.

Conclusion

Générale

Conclusion générale

Conclusion Général :

Ce travail de recherche présenté concerne un des domaines les plus intéressants et d'actualité de la robotique, à savoir la télé opération avec retour d'efforts.

Nous avons étudié une architecture de télé opération à deux canaux de communication qui utilise les signaux de position et de force respectivement au moyen d'un encodeur et un capteur de couple. Cette dernière est très souhaitable afin de permettre une commande intuitive du robot télé opéré tout en reproduisant le plus fidèlement possible les efforts. Afin d'améliorer les performances de la commande à distance, des solutions ont été envisagées dans le but de la stabilisation du système, à savoir les conditions sur la passivité et l'élimination du retard de transmission.

Dans le 2^{ème} chapitre nous avons présenté tout le matériel (les différents composants, capteurs, actionneurs) pour la réalisation de notre système ce qui va nous permettre d'avoir une idée précise sur le type de maquette à réaliser.

Pour le chapitre III nous sommes intégré dans un notre monde qui est le monde de la programmation en utilisant Arduino qui est un langage de programmation évolué, en outre il permet de faire un pont entre le monde virtuel de la programmation informatique (concepts formels et logiques) et le monde physique (interaction électromécanique des objets). La compréhension de données exposées dans ce chapitre nous offre des bases de connaissances pour l'usage de la carte d'acquisition. Le chapitre IV qui est le plus difficile car il faut faire une combinaison entre la théorie et la pratique afin de réaliser un système performant qui aboutira au cahier de charge souhaité. Donc nous avons donné toute les techniques nécessaire pour l'accomplissement de la tâche ainsi détaillé les résultats de la simulation pour bien comprendre l'asservissement force et position.

Je souhaite vivement que mon mémoire ainsi notre réalisation servira comme plateforme pour des travaux de recherche futurs en robotique ainsi dans la sécurité de fonctionnement des robots télé opérés en particulier.

Table des figures

Figure 1 : Illustration de l'architecture générale d'un système de Télé opération.....	5
Figure 2 : Environnement de télémanipulation au Argonne National Laboratory.....	6
Figure 3 : ROV de la NASA pendant la réalisation d'une mission en Antarctique	6
Figure 4 : Représentation sous forme quadripôle d'un système de télé opération avec retour d'efforts.....	6
Figure 5 : Manipulateur de type impédance.....	9
Figure 6 : Télé opération sans retour d'efforts.....	10
Figure 7 : Asservissement force-Position.....	11
Figure 8 : Couple Maître-esclave à retour d'effort.....	13
Figure 9 : Diagramme de Bode d'un modèle simplifié d'un couple maître-esclave.....	14
Figure 10 : Exemple de système de télé opération haptique	15
Figure 11: représentation d'un système de télé opération par modèle 2-ports.....	17
Figure12 : Contrôle Force-Position (FP).....	18
Figure 13: Diagramme de connexion du dispositif de contrôle et de commande.....	22
Figure 14: Transistor MOSFET.....	23
Figure 15: Caractéristique du transistor IRF 640.....	24
Figure 16: Structure du Pont H.....	24
Figure 17: structure du pont H électrique (pont H de base).....	25
Figure 18: Structure du pont H électronique.....	26
Figure 19: capteur de couple (FSR 402).....	27
Figure 20: La tension de sortie en fonction de force appliquée.....	28
Figure 21: Caractéristique du capteur FSR 402.....	28
Figure 22: Schéma de branchement du capteur à Arduino.....	29
Figure 23 : Moteur réducteur avec encodeur de quadrature.....	29
Figure 24 : Utilisation d'un encodeur optique sur l'axe du moteur.....	30
Figure 25 : Moteur réducteur avec encodeur de quadrature.....	30
Figure 26 : moteur à courant continu.....	31
Figure 27 : Schéma bloc du moteur à courant continu (MCC).....	31
Figure 28 : Schéma bloc de l'asservissement du courant.....	32
Figure 29 : Signaux moteurs.....	34
Figure 30 : Architecture interne de la carte Arduino méga 2560.....	37
Figure 31: Architecture de la carte Arduino méga 2560.....	38
Figure31: Architecture du microcontrôleur ATMEGA 2560.....	39
Figure 33 : les différentes parties de la fenêtre principale du logiciel Arduino.....	41
Figure 34 : les différentes composantes du programme Arduino.....	43
Figure 35 : l'écriture du programme.....	46
Figure 36 : compilation du programme.....	46
Figure 37 : console d'affichage.....	46
Figure 38 : Sélection de la cible.....	47
Figure 39 : Sélection du port.....	47
Figure 40 : Transfert du programme.....	47
Figure 41 : clignotement des deux LED du port série.....	48
Figure 42: Schéma de simulation du pont H sur ISIS PROTEUS.....	50
Figure 43: Schéma électrique du pont H.....	51
Figure 44: Schéma de la réalisation de la carte de puissance.....	53
Figure 46: Les valeurs données par l'encodeur (coté maître et esclave).....	52
Figure 47: valeurs données par le capteur FSR coté esclave.....	52

Table des figures

Figure 48: position du robot maître en fonction du temps	52
Figure 49: position du robot esclave en fonction du temp.....	53
Figure 50: Force appliqué sur le robot esclave en fonction du temps	53
Figure 51: Position du robot esclave an fonction de la position du robot maître.....	53
Figure 52 : Position du robot maitre en fonction de la force de l'environnement.....	53
Figure 53: La maquette réalisée.....	55

Bibliographie :

- [1] : Aura Nancy Rodriguez Architecture générale pour la télé robotique : Thèse de doctorat de l'Université Paul Sabatier de Toulouse (Sciences). Janvier 2003
- [2] : Pierre LETIER Bras Exosquelette Haptique : Conception et Contrôle (Laboratoire des Structure Actives Service des Constructions Mécaniques et Robotique) juillet 2010.
- [3] : Laurent Barbé, Thèse de l'école doctorale mathématique, sciences de l'information et de l'ingénieur télé opération avec retour d'efforts pour les interventions percutanées
- [4] : Yves BRIERE, Télé opération en présence de retard : Thèse de doctorat de l'école national supérieure de l'aéronautique et de l'espace.
- [5] : Nabil Zemiti, Commande en Effort des Systèmes Robotiques pour la Chirurgie Mini-Invasive : Université Paris 6 Spécialité Mécanique-Robotique. 2005
- [6] : Cours sur la Robotique : Audiovidéocours Univ-R AV - Visualisation de l'enregistrement ENSPS 3A - Robotique -.
- [7] : Cours sur la programmation sur Arduino : www.Arduino.cc.