الجمهورية الجزائرية الديمقراطية الشعبية

République Algérienne démocratique et populaire
Ministère de L'Enseignement Supérieur & de la Recherche Scientifique Et de la recherche scientifique
Universite Docteur Yahia Fares de MEDEA

Faculté Des Sciences Et De La Technologie
Département du Génie de la matière

Projet de Fin d'Études

Présenté pour l'obtention du Diplôme
De Master II en génie de la mécanique
Spécialité : mécatronique

THEME

ETUDE, CONCEPTION ET REALISATION D'UN BRAS MANIPULATEUR à COMMANDE VISUELLE

Proposet et dirigé par :

Dr. TCHOUKETCH KEBIR Sofiene

Dr. BACHENE Mourad

Réalisé par :

Mr. DILMI Ahmed

Mr. GOTIT Mahmoud

Année universitaire 2015-2016

Remerciement

Nous tenons à remercier notre Dieu, le tout puissant, de nous avoir donné la santé et la volonté pour compléter ce modeste travail.

Nous tenons à exprimer nos profondes gratitudes remerciements à nos promoteurs, le Docteur **BACHEN MOURAD** et **TCHOKETCH KEBIR SOFIANE** pour sa disponibilité et de nous avoir fait profiter de ses qualités aussi bien sur le plan scientifique que sur le plan professionnel. Un grand merci pour les enseignants d'INSFP de Médéa M. MAHFOUD ZOUAUINE et M^{me} KARIMA OUDJOUADJ à leurs aides, leurs remarques attentives, et leurs conseils bienveillants.

Nous tenons aussi à adresser nos vifs remerciements à M.DILMI Mustapha chef atelier mécanique à Blida, et Salah à leur majeur aide pour finir notre réalisation et ne pas oublier M. Ismail chef atelier construction métallique à Mouzaia.

Nos remercîments s'adressent également aux membres de jury d'avoir accepté de lire et d'évaluer notre mémoire. Nous exprimons nos remerciements à tous les professeurs qui nous ont encadrés pendant notre cursus universitaire. Enfin nous remercions tous ceux qui ont participé de près ou de loin à la réalisation de ce mémoire.

Dédicaces

Je remercier ALLAH le tout puissant de m’avoir donnée la force et le courage pour accomplir ce modeste travail.

J'ai l'honneur de dédier ce modeste travail à:

A mes très chers parents qui m'ont soutenu et encouragé durant toute la période de mes étude et à qui je leur souhaite une longue et heureuse vie, que dieu les protège et me donne la force pour que je puisse leur rendre un petit peu de leur bienfait et leur bonté.

A mes très chers frères et mes très chères sœurs.

A toute les personnes de ma famille.

La famille DILMI.

Je remercie mon ami et binôme Mahmoud

Et mes enseignants ZOUAUINE Mahfoud et OUEDJUADJE Karima

Et tous mes amis

A toute ma promotion de Mécatronique

Et enfin, à tous ceux qui m'aiment

AHMED

Dédicaces

Que ce travail témoigne de mes respects :
A mes parents :
Grâce à leurs tendres encouragements et leurs grands sacrifices, ils ont pu créer le climat affectueux et propice à la poursuite de mes études.
Aucune dédicace ne pourrait exprimer mon respect, ma considération et mes profonds sentiments envers eux.
Je prie le bon Dieu de les bénir, de veiller sur eux, en espérant qu'ils seront toujours fiers de moi.
A ma sœur et à mon frère.
A la famille HAMAMI et AKHIYAT.
Ils vont trouver ici l'expression de mes sentiments de respect et de reconnaissance pour le soutien qu'ils n'ont cessé de me porter.
A tous mes professeurs :
Leur générosité et leur soutien m'oblige de leurs témoigner mon profond respect et ma loyale considération.
A tous mes amis et mes collègues :
Ils vont trouver ici le témoignage d'une fidélité et d'une amitié infinie.

ملخص:

العمل المطور في هذه المذكرة يتعلق بانجاز روبوت(ذراع مناور) للاوامر المرئية .

ونحن مهتمون بالربوتات ذات السلسلة الواحدة المفتوحة.لذا نقدم اولا دراسة عن نماذج لرصد ومراقبة الربوتات والتي تتمثل في النماذج الهندسية والحركية والديناميكية, المباشرة والعكسية. و نعرض ايضا تحقيق الهيكل الميكانيكي للربوت ولنا كذلك جانب مع معالجة عنصر التحكم بالكاميرا"كيناكت"عن طريق المتحكم اردوينو اينو. .

ننتهي عملنا من خلال تطبيق عملي الذي يظهر أداء أعمالنا للذراع مناور.

الكلمات المفتاحية: ذراع مناور, كيناكت, اردوينو اينو.

Résumé :

Le travail développé dans ce mémoire concerne la réalisation d'un robot (bras manipulateur) à commande visuelle. Nous nous sommes intéressés à des robots manipulateurs rigides à chaîne ouverte simple. Nous présentons premièrement une étude sur les modèles utilisés pour le contrôle et la commande du robot, à savoir le modèle géométrique, cinématique et dynamique : directs et inverses. Puis la réalisation de la structure mécanique du robot Nous avons également traité la commande visuelle par la caméra « Kinect »à base d'Arduino Uno. On termine notre travail par une application pratique qui montre les performances du notre « bras manipulateur ».

Mots clé: bras manipulateur, Kinect, Arduino Uno.

Abstract:

The work developed in this thesis concerns the realization of a robot (manipulator arm) to Visual command. We are interested in a rigid robots open single-chain. We present first a study on models for monitoring and control of the robot, namely model geometric, kinematic and dynamic: direct and inverses. Then the realization of the mechanical structure of the robot we also processed the Visual control by the camera "Kinect." based on Arduino Uno. We finish our work through a practical application which shows the performance of our "manipulator arm.

Keywords: manipulator arm, Kinect, Arduino Uno

INTRODUTION

GENERALE

Quand on parle de robotique, plusieurs idées viennent à l'esprit de chacun de nous, historiquement les robots manipulateurs industriels sont utilisés pour réaliser des tâches de **déplacements d'outils**, de **manutention** ou d'**assemblage**.

Ils se substituent à l'homme ou prolongent son action en apportant précision, rapidité ou capacité à appliquer d'importants efforts.

Ils sont généralement constitués d'un bras manipulateur plus ou moins anthropomorphe mis en mouvement par des actionneurs électriques, hydrauliques ou pneumatiques et dont la position est repérée par une série de capteurs.

Une interface permet de spécifier la tâche à réaliser et un dispositif de commande en contrôle l'exécution par l'unité mécanique.

Le développement dans le domaine de la robotique a été poussé en premier lieu par une motivation d'imitation des capacités basiques de l'être humain, afin de lui faciliter les tâches difficiles et répétitives dans le domaine de l'industrie, telles que le soulèvement de pièces lourdes, serrage, découpage, assemblage, etc.

En fait, la robotique s'agit d'un domaine multidisciplinaire : on y trouve des aspects concernant la mécanique, l'électronique numérique et analogique, électrotechnique, Programmation, Intelligence artificielle, Automatique…..)

Les premiers robots furent alors des robots à structure série simple : La structure mobile articulée est une chaîne ouverte formée d'une succession de segments reliés entre eux par des liaisons à un seul degré de liberté ,chaque articulation est commandée par un actionneur situé à l'articulation ou sur l'un des segments précédents .Ces robots ont pour avantage de disposer d'un grand volume de travail et d'être relativement simples sur le plan des calculs liés à leur commande.

Par contre ces robots présentent des inconvénients quand ils s'agissent d'effectuer des tâches demandant de la précision ou des tâches devant être exécutées avec rapidité.

Notre travail consiste à modéliser et réaliser un robot (bras manipulateur) à base mobile, ce bras est formé de deux segments en suivant des procédures et des techniques de calcul habituellement utilisées pour les robots sériels. Ainsi, les méthodes fondées sur l'utilisation de formulations de Newton-Euler récursives et sur le développement des modèles dynamiques inverses linéaires vis-à-vis des paramètres inertiels, qui seront adaptées aux caractéristiques structurales de ces robots complexes.

Ce mémoire est organisé de la manière suivante :

Dans le premier chapitre, en commençant par une généralité sur les robots,(les différents types des robots existants dans l'industrie). Les degrés de liberté, ainsi que quelques types des robots.

En chapitre 2 on présentera avec plus d'approfondissement les méthodes les plus importantes pour la modélisation des robots, et nous enchainons ici à l'étude de notre réalisation par la détermination des équations qui gèrent les mouvements de notre robot.

Le troisième chapitre traite la présentation de la carte arduino avec les servomoteurs.

Enfin le contenu du quatrième chapitre présente les performances et tests obtenus avec les applications qu'on a réalisées.

Nous terminons ce mémoire par une conclusion et quelques perspectives.

Chapitre I : Généralité sur les robots

I.1 Introduction :

La **robotique** peut être définie comme l'ensemble des techniques et études tendant à concevoir des systèmes mécaniques, électroniques, informatiques ou mixtes, capables de se substituer à l'homme dans ses fonctions motrices, sensorielles et intellectuelles. À partir de cette raison elle est considérée de plus comme le facteur primordial d'une nouvelle mutation des moyens de production.

I.2 Historique de la robotique :

La transition entre les automates et les robots va devenir possible grâce à George Boole. Ce professeur du Queens Collège de Cork (Irlande) établit en 1854 le système d'algèbre booléenne, fait de zéro et de un, qui permettra un siècle plus tard l'apparition du premier ordinateur.

Le premier à employer le terme robot est Karel Capek, écrivain et dramaturge tchèque du XXème siècle. Il utilisa le terme robot dans sa pièce de science-fiction *Rossumovi univerzální roboti* (RUR) écrite en 1920. Il inventa le mot robot à partir du mot «*rabota*» qui signifie «corvée» en slave ancien. De nos jours encore, le mot «*robotnik*» est utilisé en slovaque et polonais pour désigner l'ouvrier. Il faut noter que dans la pièce de Capek, le terme robot désigne un cyborg, c'est-à-dire un être bionique fait à la fois de parties organiques et artificielles, et non pas une créature mécanique.

Dès 1936, Alan M. Turing énonce le principe de «machine à penser» dans un article intitulé 'Computable Numbers', où il décrit un système capable de traiter un très grand volume d'informations à une vitesse vertigineuse.

C'est précisément dans les années 1940 que l'arrivée de l'électronique puis de l'ordinateur vont bouleverser l'évolution de la robotique. De nombreux mathématiciens et informaticiens vont approfondir la question et en 1956, un professeur de l'université de Carnegie Mellon (Pittsburgh), Herbert Simon, conçoit un programme informatique capable de résoudre des théorèmes mathématiques simples. C'est la première fois qu'un système non vivant parvient à simuler un raisonnement logique, marquant ainsi les débuts de l'intelligence artificielle. Allen Newell l'épaulera dans ses travaux d'élaboration de programmes capables de modéliser les activités intellectuelles humaines, notamment grâce à des mécanismes issus de la psychologie.

La cybernétique permet de comprendre les mécanismes sous-jacents à l'exécution d'une tâche. William Grey Walter applique cette théorie à une machine autonome pour étudier les comportements des animaux. Il fabrique Elsie et Elmer, deux tortues mobiles, capables de se déplacer de manière autonome et de se repérer grâce au son et à la lumière par le biais de capteurs photoélectriques. [1]

I.3 Quelques définitions de base :

La robotique est une science qui fait intervenir un nombre de connaissances, la conception des systèmes, les mathématiques, la mécanique, l'électronique, les asservissements, la fluidique l'informatique, l'analyse des gestes et de la scène dans un environnement .l'intelligence artificiel permet les prises de décision et les modifications du comportement.

I.3.1 La structure mécanique articulée (SMA) :

Un robot manipulateur est constitué généralement par deux sous-ensembles distincts : un organe terminal qui est le dispositif destiné à manipuler des objets,

et une structure mécanique articulée (SMA), constituée d'un ensemble de solides reliés entre eux, généralement les uns à la suite des autres où chaque solide est mobile par rapport au précédent. Cette mobilité s'exprime en terme de degrés de liberté (d.d.l) qui est par définition le nombre de mouvements indépendants possibles d'un solide S1 par rapport au solide qui lui est directement relié S2.

I.3.1.1 La structure mécanique articulée à chaîne cinématiques ouverte simple :

C'est une chaîne cinématique dont chaque membre possède un degré de connexion (nombre de liaisons mécaniques) inférieur ou égal à deux. Un robot sériel est formé d'une chaîne cinématique simple dont la base et l'organe effecteur possèdent un degré de connexion de un (c'est-à-dire qu'il n'est relié qu'à un seul corps) et les autres éléments un degré de connexion de deux. Ainsi, chaque corps Cn est relié par ses deux extrémités à un seul corps, et la base et l'effecteur ne sont liés qu'à un seul corps Cn.

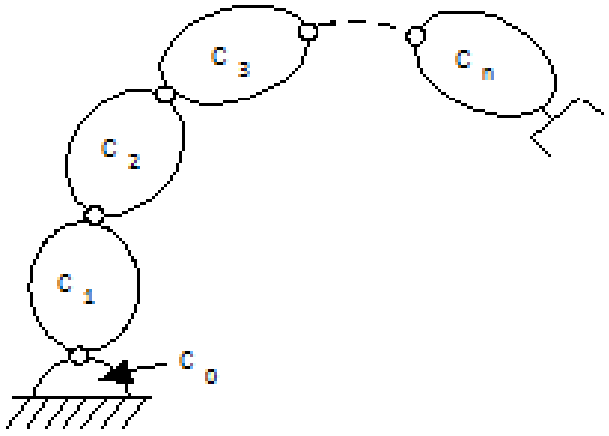


Figure I.1: Robot à structure ouverte simple.

I.3.1.2 La structure mécanique articulée à chaîne cinématiques fermée :

C'est une chaîne cinématique dont l'un des membres, différent de la base, possède un degré de connexion supérieur ou égal à trois.

Dans notre exemple (Figure I.2), les corps C1, C2, C3 et C4 forment une chaîne cinématique fermée. Ces manipulateurs sont intrinsèquement plus rigides que les manipulateurs sériels. [2]

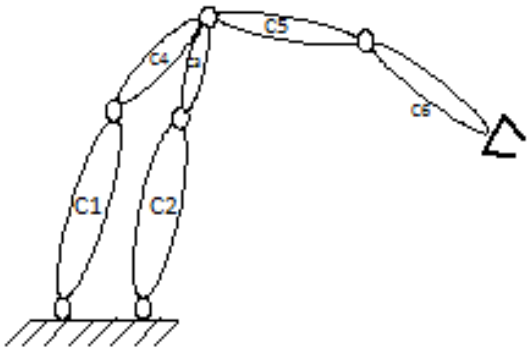


Figure I.2: Manipulateur possède une chaîne cinématique fermée.

I.4 Les liaisons mécaniques usuelles :

Une articulation lie deux corps successifs en limitant le nombre de degrés de liberté de l'un par rapport à l'autre. Soit m le nombre de d.d.l. résultant ou encore appelé mobilité de l'articulation, la mobilité est telle que 0 <m <6.

Lorsque m = 1 ce qui est le cas le plus fréquent en robotique, l'articulation est soit rotoïde, soit prismatique. On peut créer des liaisons de mobilité supérieure à un, en combinant des articulations simples. Par exemple, une liaison sphérique (rotule) est obtenue avec trois articulations rotoïdes d'axes concourants.

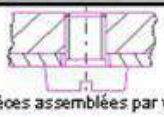
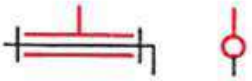
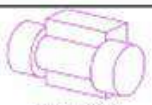
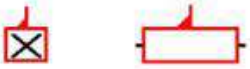
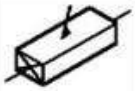
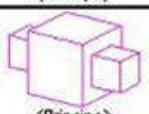
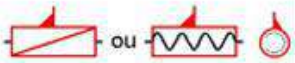
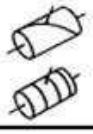
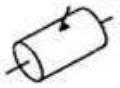
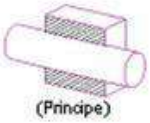
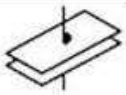
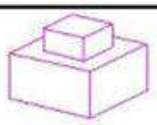
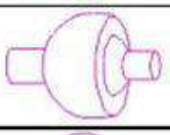
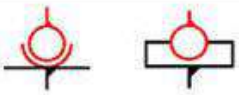
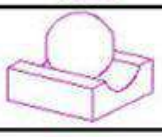
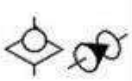
Nom de la liaison	Degrés de liberté (d.d.l)	Mouvements relatifs	Symbole		Exemples
			Représentation plane	Perspective	shaysite.net.tc
Encastrement ou Fixe	0	Translation Rotation			 Pièces assemblées par vis
Pivot	1	Translation Rotation			 (Principe)
Glissière	1	Translation Rotation			 (Principe)
Hélicoïdale	1	Translation Rotation Translation et rotation conjuguées			 (vis + Ecrou)
Pivot glissant	2	Translation Rotation Rotation			 (Principe)
Appui plan	3	Translation Rotation			
Rotule ou sphérique	3	Translation Rotation			
Linéaire annulaire ou sphère-cylindre	4	Translation Rotation			
Linéaire rectiligne	4	Translation Rotation			
Ponctuelle ou Sphère-plan	5	Translation Rotation			

Tableau I.1: Les différentes représentations des liaisons mécaniques. [3]

I.5 Particularités d'un robot :

I.5.1 Définition d'un robot :

Un robot est un manipulateur à plusieurs degrés de libertés, commandé en position reprogrammable, polyvalent. Il est capable de manipuler des matériaux, pièces, outils, ou dispositifs spécialisés au cours de mouvements variables en vue de l'exécution d'une variété de taches. Son unités de commande utilise un dispositif de mémoire et éventuellement de perception et d'adaptation à l'environnement et aux circonstances. Ces machines polyvalentes sont généralement étudier pour effectuer la même fonction de façon cyclique, et peuvent être adaptées à d'autre fonction sans modification permanente du matériel.

I.5.2 Le dictionnaire donne plusieurs définitions:

1) Traduit du dictionnaire Webster:

"Un appareil automatique qui peut effectuer des fonctions normalement effectué par des êtres humains."

2) Petit Larousse :

"Appareil automatique capable de manipuler des objets ou d'exécuter des Opérations selon un programme fixe ou modifiable."

3) Institut de robotique d'Amérique :

"Un manipulateur reprogrammable multifonctionnel conçu pour déplacer des matériaux, des outils, des pièces ou des composantes spécialisés à travers une série de mouvements programmés pour effectuer une tache précise."

4) whatis.com:

"Un robot est une machine conçue pour exécuter une ou plusieurs tâches à plusieurs reprises, avec rapidité et précision [4]

I.6 Les constituants d'un bras manipulateur:

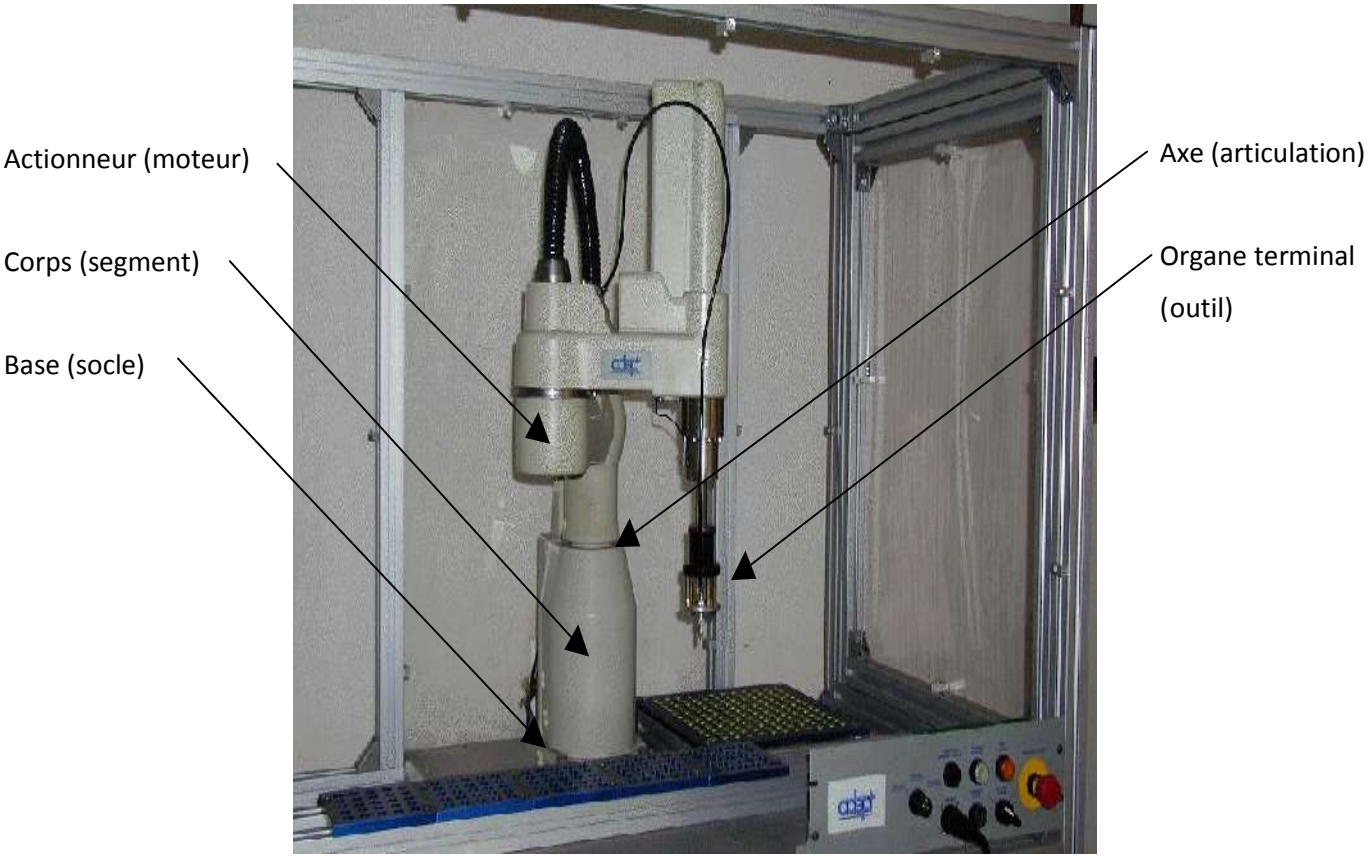


Figure I.3 : vocabulaire d'un bras manipulateur

I.7 Classification des robots :

On retiendra pour notre part 3 types de robot :

I.7.1 Les manipulateurs :

I.7.1.1 leurs Avantages:

- Les trajectoires sont non quelconques dans l'espace.
- Les positions sont discrètes avec 2 ou 3 valeurs par axe.
- La commande est séquentielle.



Figure I.4 : robot manipulateur « Kawasaki ».

I.7.2 Les télémanipulateurs :

Les appareils de manipulation à distance (pelle mécanique, pont roulant) apparus vers 1945 aux USA :

I.7.2.1 Leurs Avantages:

- Les trajectoires peuvent être quelconques dans l'espace et sont définies d'une manière instantanée par l'opérateur,

Généralement à partir d'un pupitre de commande.



Figure I.5 : pelle mécanique

I.7.3 Les robots :

I.7.3.1 Leurs Avantages:

- Les trajectoires peuvent être quelconques dans l'espace,
- L'exécution est automatique,
- Les informations extéroceptives peuvent modifier le comportement du robot.

Pour cette dernière classe, on peut distinguer :

I.7.3.1.1 Les robots manipulateurs industriels :

Chargés de manipuler, soient :

Des pièces : Stockage – déstockage, Palettisation, Chargement – déchargement de machine-outil, Manipulation d'éprouvettes, Assemblage de pièces, …

Des outils : Soudure en continu ou par points, Peinture,Collage, Ebavurage, ...



Figure I.6 : robots soudeurs

I.7.3.1.2 Les robots mobiles autonomes :

Les possibilités sont plus vastes, du fait de leur mobilité. Notamment, ils peuvent être utilisés en zone dangereuse (nucléaire, incendie, sécurité civile, déminage), inaccessible (océanographie, spatial). De telles sortes que ces robots font appel à des capteurs et à des logiciels sophistiqués.

On peut distinguer 2 types de locomotion :

Les robots *marcheurs* qui imitent la démarche humaine, et les robots *mobiles* qui ressemblent plus à des véhicules.

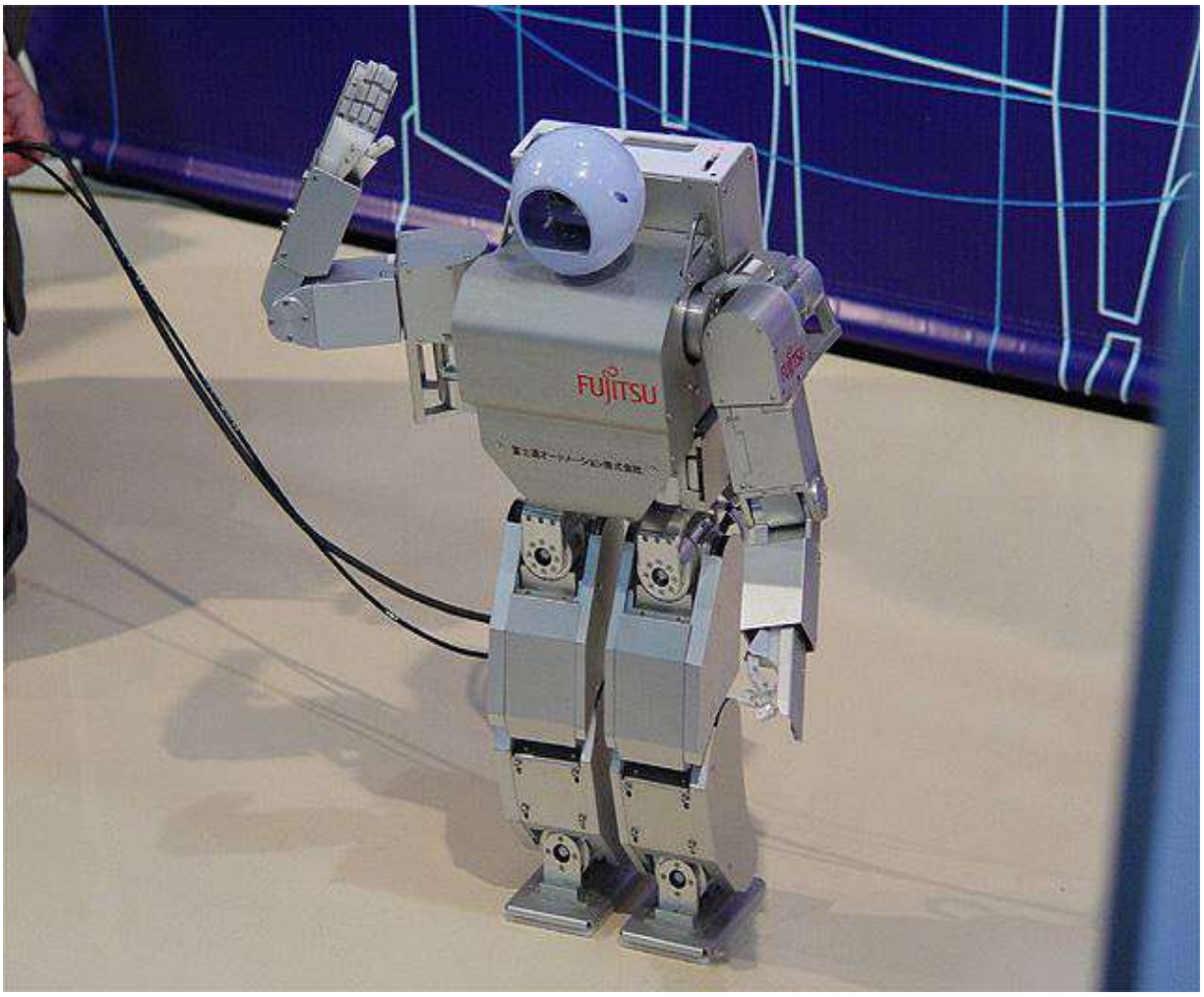


Figure I. 7 : robot autonome Fujitsu

I.8 Caractéristiques d'un robot :

Un robot doit être choisi en fonction de l'application qu'on lui réserve. Voici quelques paramètres à prendre, éventuellement, en compte :

- La charge maximale transportable (de quelques kilos à quelques tonnes), à déterminer dans les conditions les plus défavorables (en élongation maximum).
- L'architecture de la structure mécanique articulée. le choix est guidé par la tâche à réaliser.
- Le volume de travail, défini comme l'ensemble des points atteignables par l'organe terminal. Tous les mouvements ne sont pas possibles en tout point du volume de travail.

  L'espace de travail (reachable workspace), également appelé espace de travail maximal, est le volume de l'espace que le robot peut atteindre via au moins une orientation.

  L'espace de travail dextre est le volume de l'espace que le robot peut atteindre avec toutes les orientations possibles de l'effecteur (organe terminal). Cet espace de travail est un sous-ensemble de l'espace de travail maximal.
- Le positionnement absolu, correspondant à l'erreur entre un point souhaité (réel), défini par une position et une orientation dans l'espace cartésien et le point atteint est calculé via le modèle géométrique inverse du robot. Cette erreur est due au modèle utilisé, à la quantification de la mesure de position, à la flexibilité du système mécanique. En général, l'erreur de positionnement absolu, également appelée précision, est de l'ordre de 1*mm*.
- La répétabilité, ce paramètre caractérise la capacité que le robot à retourner vers un point (position, orientation) donné. la répétabilité correspond à l'erreur maximum de positionnement sur un point prédéfini dans le cas de trajectoires répétitives. En général, la répétabilité est de l'ordre de 0,1 mm

- La vitesse de déplacement (vitesse maximum en élongation maximum).
- La masse du robot.
- Le coût du robot.
- La maintenance, …

I.9 Les différents types des robots :

I.9.1 Robot Cartésien:

Robots cartésiens typiques ont 3 axes linéaires de liberté qui sont orientés perpendiculairement de l'un à l'autre. Comme vous pouvez le comprendre, cela garantit une enveloppe de travail sous la forme d'une boîte rectangulaire.



Figure I.8: 3-axes Cartésien du robot d'EPSON'

I.9.2 Robot Cylindrique :

Un système robotique cylindrique à trois axes de mouvement, l'axe de mouvement circulaire et les deux axes linéaires dans le mouvement horizontal et

vertical du bras. Alors que beaucoup de gens compare un robot cylindrique à un robot SCARA à cause de leur travail similaire enveloppe, selon un article de tous les Robots, les applications qui peuvent fournir un système robotique cylindrique sont très différentes d'un SCARA. Car il peut être combiné avec un outillage. Un robot cylindrique est en mesure d'effectuer la manipulation et l'assemblage, mais aussi des tâches telles que soudage par points. Cette polyvalence est importante pour un fabricant.

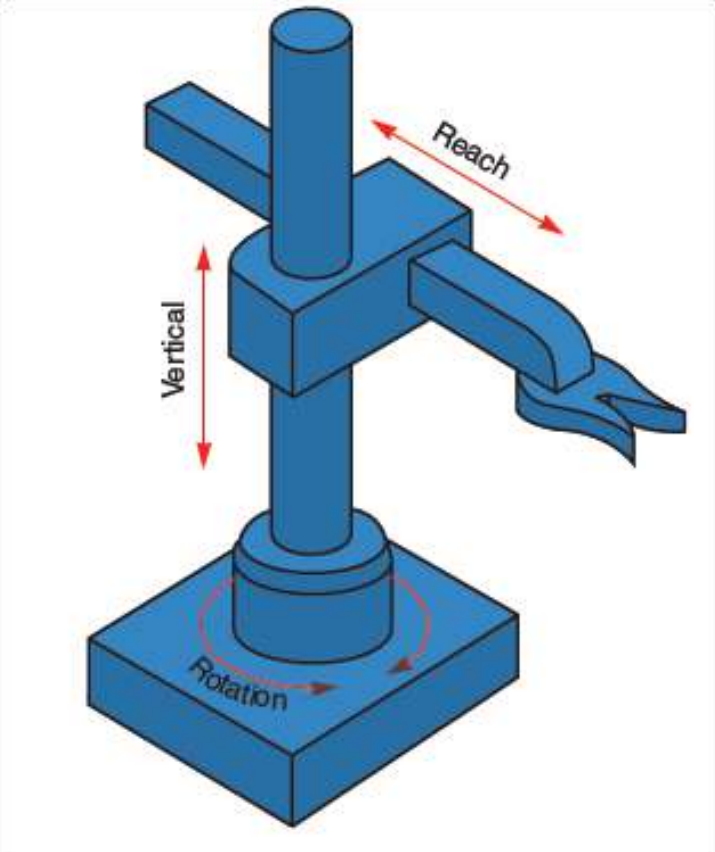


Figure I.9: Robot Cylindrique

I.9.3 Robot SCARA :

Le terme SCARA est un acronyme qui signifie sélectif conforme bras Robot de l'Assemblée. Le robot SCARA est basé sur une conception de 4 axes. Il est idéal pour l'assemblage à grande vitesse, emballages et autres applications

de manutention. Innovations de robot (robot 4 axes) d'adepte haute vitesse SCARA ont conduit l'industrie depuis plus de vingt-cinq ans. En 1984, Adept introduit le robot SCARA AdeptOne, premier robot d'entraînement direct du monde. Le robot SCARA AdeptOne changé le visage de la robotique grâce à sa conception novatrice.



Figure I.10: Robot SCARA

I.9.4 Robot Articulateur*:*

Un robot articulé est un robot qui est équipé de joints rotatifs. Articulations rotatives permettent une gamme complète de mouvement, car elles tournent dans des plans multiples, et accroissent considérablement les capacités du robot. Un robot articulé peut comporter un ou plusieurs joints rotatifs et autres types de joints peuvent être utilisés aussi bien, selon la conception du robot et sa

fonction. Ces robots sont visibles également au travail dans les laboratoires et dans nombreux autres paramètres.

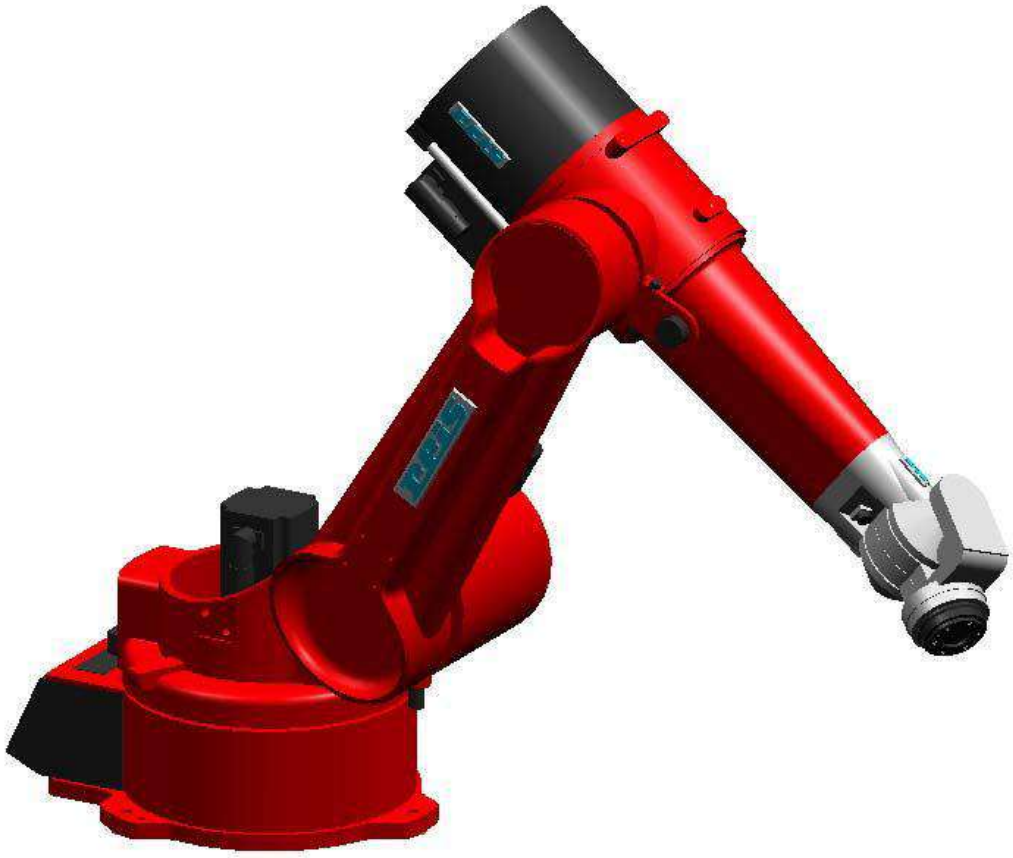


Figure I.11: Robot Articulateur

I.9.5. Robot sphérique/Polar :

Utilisé pour la manutention au soudage par points, machines-outils, machines d'ébavurage, moulage sous pression, gaz de soudage et soudage à l'arc. C'est un robot dont les axes forment un système de coordonnées polaire.



Figure I.12: robot Hexapode

I.9.6 Robot Parallel:

Un manipulateur parallèle est un système mécanique qui utilise plusieurs chaînes séries commandé par ordinateur pour soutenir une plateforme unique, ou un effecteur. Le manipulateur parallèle connu peut-être formé de six actionneurs linéaires qui prennent en charge une base mobile pour périphériques tels que les simulateurs de vol. Les robots parallèles également connu sous le nom, ou la plates-formes de Stewart généralisées (dans la plate-forme de Stewart, les actionneurs sont associés ensemble sur la base et la plate-forme),. Leur distinction « parallèle », par opposition à un manipulateur de série, est que l'effecteur (ou "main") de ce lien (ou "bras") est connectée à sa base par un

certain nombre de liens séparés et indépendants (habituellement trois ou six) travaillant en parallèle. [5]



Figure I.13: Le robot FlexPicker

I.10 Les performances des robots et leur mesure :

I.10.1 Les performances taches sont :

1) le volume atteignable par l'extrémité du robot.
2) les possibilités de l'orientation de l'organe terminal : de par la présence de butté sur chaque articulation et aussi parfois de la présence de couplage mécanique entre articulation.
3) La charge utile : c'est la charge maximale transportable dans un domaine réduit du volume atteignable.
4) La précision en position
5) La résolution : c'est un paramètre très important dans les applications d'assemblage ou de montage automatique.
6) La vitesse : c'est une caractéristique fondamentale pour les robots industriels.

7) La fiabilité : c'est une donnée qui s'intéresse au plus haut point l'utilisateur même s'il ignore tout du système.

I.10.2 Les performances-homme :

Elles correspondent à la facilité d'utilisation du robot, ce dernier étant installé sur le site et entre les mains d'un non-spécialiste. Les compétences exigées pour faire fonctionner un robot ont trait :

La connaissance du matériel **:** réglage d'asservissement électrique ou hydraulique etc.…

La connaissance du logiciel **:** programmation du robot.

Il est important à savoir si un technicien, après quelque jour de l'entrainement, peut avoir la responsabilité du robot y compris la maintenance courante et les réparations mineures ou si ces taches exigent à chaque incident la présence d'un ingénieur voir d'un spécialiste de chez le constructeur.

I.10.3 Les performances économiques :

Il s'agit de la rentabilité d'utilisation de robot dans l'entreprise, dans l'avenir elle ne peut que s'améliorer elle dépend d'un grand nombre de facteur pas toujours bien cernes, dont :

- Les coûts d'investissement : prix d'achat cout d'installation…
- Les coûts d'exploitation : énergie, maintenance, espérance de vie.
- La nature des taches a exécuté : taches qui revenaient cher en main-d'œuvre
- La qualité de production sera-t-elle améliorée.

- Le degré d'automatisation déjà présent dans l'atelier.

I.11 Les intérêts de la robotique :

I.11.1 L'intérêt humain :

La robotique permet de réaliser des tâches répétitives, fastidieuses, pénibles, dangereuses, dans des milieux hostiles ou difficiles d'accès pour l'homme, celui-ci peut se consacrer à des taches plus gratifiantes.

I.11.2 L'intérêt technique :

C'est une amélioration des qualités techniques, de la précision des cadences élevées et continues, de la fiabilité dans les taches, de la flexibilité et du travail dans tout environnement.

I.11.3 L'intérêt économique :

La robotique contribue à pallier au manque de main-d'œuvre. Elle favorise la haute productivité, le travail continu sans variation de cadences et la flexibilité, l'homme et son intérêt restent au-dessus de la machine [6]

I.12 CONCLUSION :

Ce chapitre fait une apparition sur le domaine de la robotique avec une présentation des différents type des robots avec leurs caractéristiques, puis dans le chapitre suivant on s'intéressera à l'étude et la modélisation mécanique, et on abordera aussi notre model mécanique.

Chapitre II: Modélisation mécanique

II.1 Introduction :

La modélisation des robots, comme tout procédé industriel que l'on peut commander est une étape que l'on peut qualifier d'intermédiaire, puisqu'il s'agit de définir les équations qui décrivent le comportement dynamique d'un robot avec assez de précision pour que leur utilisation soit suffisamment simple au moment de la conception.

La modélisation est la base de la commande, à partir de la bonne connaissance du comportement du système étudié, il est plus facile de choisir les lois et stratégies des commandes adaptées pour aboutir aux performances souhaitées.

II.2 Structure mécanique :

La structure mécanique du robot manipulateur peut être divisée en deux parties :

-Organe terminal : les tâches qui sont dévolues aux robots sont très variées. Pour chaque opération ou travail spécifique, l'organe terminal prend un aspect particulier.

-Elément porteur : il est composé d'un ensemble de corps souples ou rigides liés par des articulations, servant à déplacer l'organe terminal d'une configuration à une autre.

Avant de décrire les relations géométriques entre les différents corps du robot, on définit les notions suivantes :

- **Degré de liberté** : nombre de paramètres utilisés pour spécifier la configuration d'un élément de la chaîne cinématique par rapport à un autre.
- **Degré de mobilité** : toute articulation est caractérisée par son degré de mobilité (DDM), c'est-à-dire le nombre de degrés de liberté entre deux corps successifs de la chaîne cinématique ($0 \leq m \leq 6$)

- **Espace articulaire** : représente l'état des corps composant le robot en fonction des variables articulaires, sa dimension n est égale au nombre de degrés de liberté du robot.
- **Espace opérationnel** : décrit la position et l'orientation de l'organe terminal du robot par rapport à un repère de référence, sa dimension m≤6.

II.3 Les différentes méthodes de modélisation :

La conception et la commande des robots nécessitent le calcul de certains modèles mathématiques tels que les modèles de transformation entre l'espace opérationnel (dans lequel est définie la situation de l'organe terminal), et l'espace articulaire (dans lequel est définie la configuration du robot).

On distingue :

- La modélisation géométrique :(directe et inverse).
- La modélisation cinématique (différentielle) : (directe et inverse).
- La modélisation dynamique :(directe et inverse).

II.3.1 Modèle géométrique direct :

Le modèle géométrique direct (MGD) est l'ensemble des relations qui permettent d'exprimer la situation de l'organe terminal, c'est-à-dire les coordonnées opérationnelles du robot, en fonction de ses coordonnées articulaires.

Le modèle géométrique direct du robot peut représenter par la relation :

$$\mathbf{X} = \mathbf{f}(\mathbf{q}) \ldots\ldots (II.1)$$

Où:

q étant le vecteur des variables articulaires tel que :

$\mathbf{q} = [q1\ q2\ \ldots\ qn]^T \in R^n$

Les coordonnées opérationnelles sont définies par :

$X = [X_1, X_2, \cdots X_n]^T \in R^m$ $\quad m \leq n.$

Pour les rotations, d'autres représentations sont couramment utilisées comme les angles d'Euler, les angles de Roulis-Tangage-Lacet (RTL) ou les paramètres d'Euler (quaternions). [7]

On note la convention de Denaviet-Hartenberg et les matrices 4*4 :

Considérons la (figure II.1) deux articulations successives pour passer de O_{i-1} à O_i et de Z_{i-1} à Z_i

On peut toujours suivre le chemin suivant :

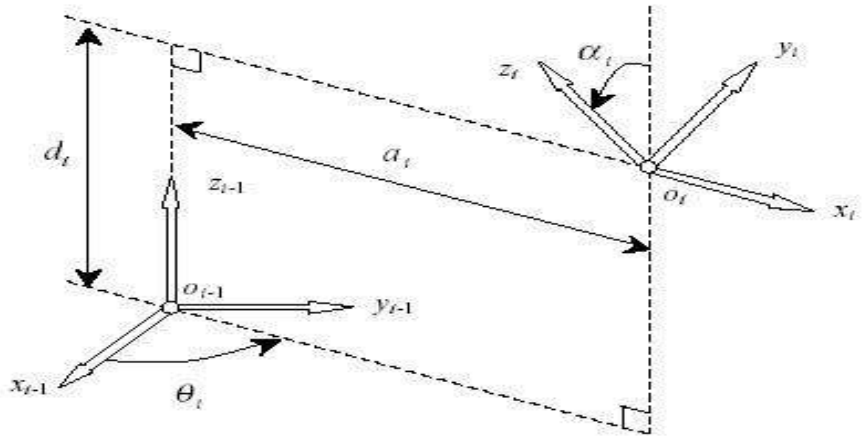


Figure II.1 : Systèmes de coordonnées et les paramètres de Denavit - Hardenberg

1) une rotation autour de Z_{i-1} d'un angle θ_i: Rot (z,θ)
2) une translation le long de Z_{i-1} d'une longueur d_i *:* Tr (z, d)
3) une translation le long de X_i d'une longueur a_i : Tr(x, a)

4) une rotation autour de X_i d'un angle α_i : Rot (x, α)

Les grandeurs sont définies de la façon suivante :

d_i : distance de O_{i-1} à la projection de O_i sur l'axe Z_{i-1}

a_i : distance de O_{i-1}à la projection de O_i sur le plan Y_i O_i Z_i

θ_i: angle compris entre X_{i-1} et le plan défini par l'axe Z_{i-1} et le point O_i

α_i : angle forme par la projection de Z_{i-1} sur le plan normal au plan (Z_{i-1}, O_i) et l'axe Z_i .

La forme générale de la matrice faisant passer d'une articulation à la suivante devra donc toujours :

La matrice de transformation homogène est donnée comme suit : [8]

$$^{i-1}T_{i} = Trans(x_{i-1}, a_{i-1})\,Rot(x_{i-1}, \alpha_{i-1})\,Trans(z_i, d_i)\,rot(z_i, \overline{\omega})$$

$$^{i-1}T_{I} = \begin{pmatrix} 1 & 0 & 0 & \alpha_{i-1} \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & c\alpha_{i-1} & -s\alpha_{i-1} & 0 \\ 0 & s\alpha_{i-1} & c\alpha_{i-1} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} c\overline{\omega}_i & s\overline{\omega}_i & 0 & 0 \\ c\overline{\omega}_i & c\overline{\omega}_i & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$^{i-1}T_{I} = \begin{pmatrix} c\omega_i & -s\overline{\omega}_i & 0 & d_{i-1} \\ C\alpha_{i-1}s\overline{\omega}_i & C\alpha_{i-1}s\overline{\omega}_i & -s\alpha_{i-1} & -s\alpha_{i-1}d_i \\ s\alpha_{i-1}s\overline{\omega}_i & s\alpha_{i-1}s\overline{\omega}_i & c\alpha_{i-1} & -c\alpha_{i-1}d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$^{i-1}T_{I} = \begin{pmatrix} ^{i-1}A_I & ^{i-1}P_i \\ 0_3 & 1 \end{pmatrix} \ldots\ldots\ldots (II.2)$$

Avec :

$$ {}^{i-1}A_i = \begin{pmatrix} c\overline{\omega_i} & -s\overline{\omega_i} & 0 \\ c\alpha_{i-1}s\overline{\omega_i} & c\alpha_{i-1}c\overline{\omega_i} & -s\alpha_{i-1} \\ s\alpha_{i-1}s\overline{\omega_i} & s\alpha_{i-1}c\overline{\omega_i} & c\alpha_{i-1} \end{pmatrix}, \; {}^{i-1}P_i = \begin{pmatrix} d_{i-1} \\ -s\alpha_{i-1}d_i \\ c\alpha_{i-1}d_i \end{pmatrix} $$

II.3.2 Modèle géométrique inverse :

On a vu que le modèle géométrique direct d'un robot permettait de calculer les coordonnées opérationnelles donnant la situation de l'organe terminal en fonction des coordonnées articulaires.

Le model inverse consiste à calculer les coordonnées articulaires correspondant à une situation donnée de l'organe terminal. Lorsqu'elle existe, la forme explicite qui donne toutes les solutions possibles (il y a rarement unicité de solution) constitue ce que l'on appelle le modèle géométrique inverse (MGI). On peut distinguer trois méthodes de calcul du MGI :

- la méthode de Paul qui traite séparément chaque cas particulier et convient pour la plupart des robots industriels ;
- la méthode de Peiper qui permet de résoudre le problème pour les robots à six degrés de liberté possédant trois articulations rotoïdes d'axes concourants ou trois articulations prismatiques.
- la méthode générale de Raghavan et Roth donnant la solution générale des robots à six articulations à partir d'un polynôme de degré au plus égal à 16.

Le modèle géométrique inverse peut s'écrit :

$$ q = f^{-1}(X) \ldots\ldots\ldots (II.3) $$

Dans le calcul du MGI trois cas se présentent :

a) absence de solution lorsque la situation désirée est en dehors de la zone accessible du robot. Celle-ci est limitée par le nombre de degrés de liberté, les débattements articulaires et la dimension des segments.

b) infinité de solutions lorsque :

- le robot est redondant vis-à-vis de la tâche.
- le robot se trouve dans certaines configurations singulières.

c) solutions en nombre fini, exprimées par un ensemble de vecteurs {**q**1, …, **q**i}.

On dit qu'un robot manipulateur est résoluble lorsqu'il est possible de calculer toutes les configurations permettant d'atteindre une situation donnée.
Aujourd'hui, tous les manipulateurs série ayant jusqu'à six degrés de liberté et qui ne sont pas redondants peuvent être considérés comme résolubles.

d) Le nombre de solutions dépend de l'architecture du robot manipulateur.

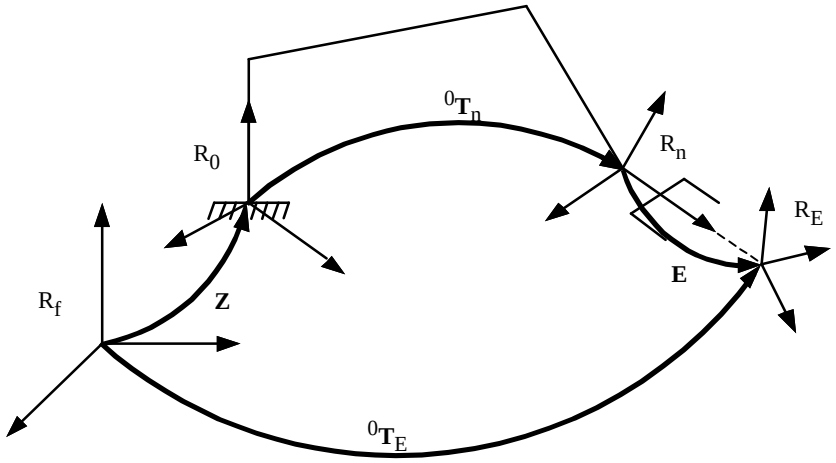


Figure II.2: *Transformations entre l'organe terminal et le repère atelier*

II.3.3 Le model cinématique direct :

Nous venons de constater que le model géométrique posait deux catégories principales de problème dans son utilisation pour la commande des robots :

1) la non-linéarité de l'équation X=f (θ) entraine des difficultés d'obtention du transformateur de coordonnées inverse .le robot est soumis à la restriction solvable.
2) avec la commandes géométrique on ne contrôle pas (ou très mal) la vitesse de déplacement du robot. Dans les applications courantes il est très important de connaitre la dure d'exécution d'une tache ou plus exactement d'en être maitre, d'où la nécessite de contrôler la vitesse.

L'utilisation du model cinématique permet dans une certaine mesure de contourner ces deux difficultés d'où son intérêt.

Le model cinématique direct d'un robot manipulateur décrit les variations élémentaire des coordonnées opérationnels en fonction des variations élémentaire des coordonnes articulaires il est noté :

$$\dot{\mathbf{X}} = \mathbf{J(q)}\dot{\mathbf{q}} \quad \ldots\ldots\ldots \quad (II.4)$$

Où **J**(**q**) désigne la matrice jacobéenne de dimension (mxn) du mécanisme, égale à $\partial\mathbf{X}/\partial\mathbf{q}$ et fonction de la configuration articulaire **q**. La même matrice jacobéenne intervient dans le calcul du modèle différentiel direct qui donne les variations élémentaires **dx** des coordonnées opérationnelles en fonction des variations élémentaires des coordonnées articulaires **dq** soit :

$$\mathbf{dx} = \mathbf{J(q)}\ \mathbf{dq} \ldots\ldots\ldots \quad (II.5)$$

II.3.3.1 Matrices jacobéenne cinématique:

On peut obtenir la matrice jacobéenne par une méthode de calcul direct, fondée sur la relation entre les vecteurs des vitesses de translation et de rotation **V**n et wn du repère Rn, et les vitesses articulaires $\dot{q}$ **:**

$$\begin{pmatrix} \mathbf{Vn} \\ \boldsymbol{\omega}\mathbf{n} \end{pmatrix} = \mathbf{Jn}\dot{\boldsymbol{q}} \ldots\ldots\ldots (II.6)$$

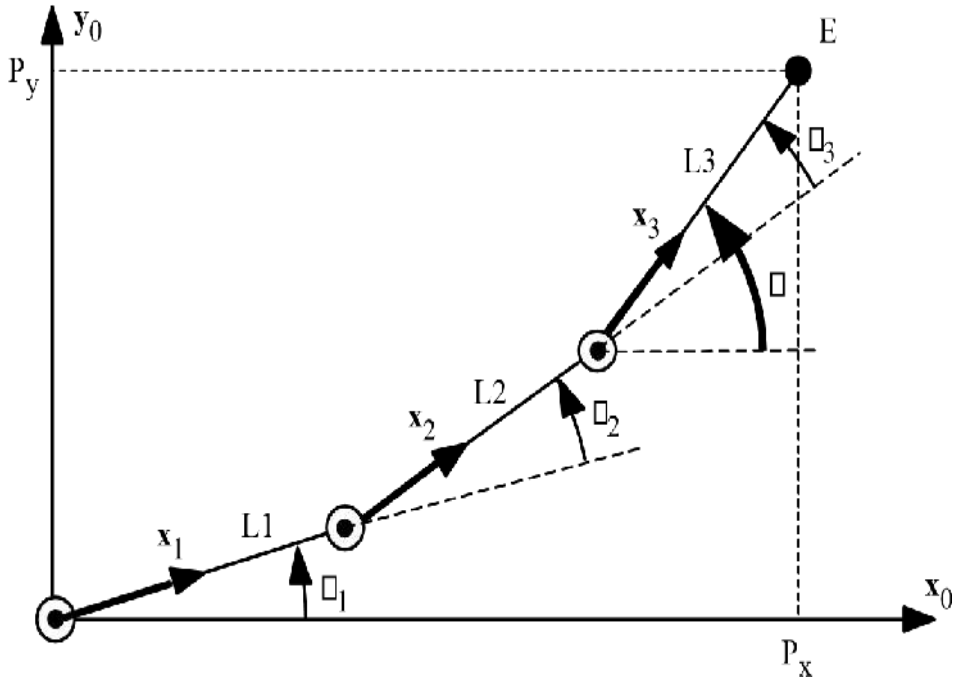


Figure II.3: Exemple d'un robot plan à trois degrés de liberté.

On note que **V**n est la dérivée par rapport au temps du vecteur **P**n.

En revanche, $\boldsymbol{\omega n}$ n'est pas la dérivée d'une représentation quelconque de l'orientation.

II.3.4 Modèle cinématique inverse :

L'objectif du modèle cinématique inverse est de calculer, à partir d'une configuration **q** donnée, les vitesses articulaires qui assurent au repère terminal une vitesse opérationnelle.

Cette définition est analogue à celle du modèle différentiel inverse : ce dernier permet de déterminer la différentielle articulaire **dq** correspondant à une différentielle des coordonnées opérationnelles **dx** spécifiée. Pour obtenir le modèle cinématique inverse, on inverse le modèle cinématique direct en résolvant un système d'équations linéaires.

Les algorithmes traitent de façon unifiée les cas réguliers, singuliers et redondants. Elles nécessitent un temps de calcul relativement important.

Nous présentons dans ce paragraphe les techniques à mettre en œuvre pour établir un modèle cinématique inverse dans les cas réguliers, singuliers et redondants.

II.3.5 Modélisation dynamique

Le modèle dynamique est la relation entre les couples (et/ou forces) appliqués aux actionneurs et les positions, vitesses et accélérations articulaires.

On représente le modèle dynamique par une relation de la forme :

$\Gamma = \mathbf{f}(\boldsymbol{q}, \dot{\boldsymbol{q}}, \ddot{\boldsymbol{q}}, \mathbf{fe})$ ……. (*II.7*)

Avec :

Γ: vecteur des couples/forces des actionneurs, selon que l'articulation est rotoïde ou prismatique.

$\boldsymbol{q}$**:** Vecteur des positions articulaires.

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

$\ddot{\boldsymbol{q}}$: Vecteur des accélérations articulaires.

fe: vecteur représentant l'effort extérieur (forces et moments) qu'exerce le robot sur l'environnement.

Le *modèle dynamique direct* est celui qui exprime les accélérations articulaires en fonction des positions, vitesses et couples des articulations. Il est alors représenté par la relation **:**

$$\ddot{q} = g(q, \dot{q}, \Gamma, fe) \quad \ldots\ldots\ldots \quad (II.8)$$

Parmi les applications du modèle dynamique, on peut citer :

- la simulation, qui utilise le modèle dynamique direct.
- le dimensionnement des actionneurs.
- l'identification des paramètres inertiels et des paramètres de frottement du robot.
- la commande, qui utilise le modèle dynamique inverse.

Plusieurs formalismes ont été utilisés pour obtenir le modèle dynamique des robots, les formalismes les plus souvent utilisés sont :

a) le formalisme de Lagrange pour le calcul du model dynamique direct.

b) le formalisme de Newton-Euler, leur méthodes sont les mieux adaptés pour le calcul du model dynamique inverse.

II.4 Modélisation de notre robot :

II.4.1 Structure du robot :

Notre robot est un bras manipulateur a des articulations rotoïde, possède 3DDL et muni d'un poigne à son extrémité.

Les trois DDL permettent de positionner l'organe terminal en un point quelconque de l'espace opérationnel.

Le bras manipulateur est muni à quatre servomoteurs, l'asservissement de ces moteurs est réalisé par un programme Matlab.

II.4.2 Détermination de la position finale du bras en fonction de : θ_1, θ_2, et θ_3

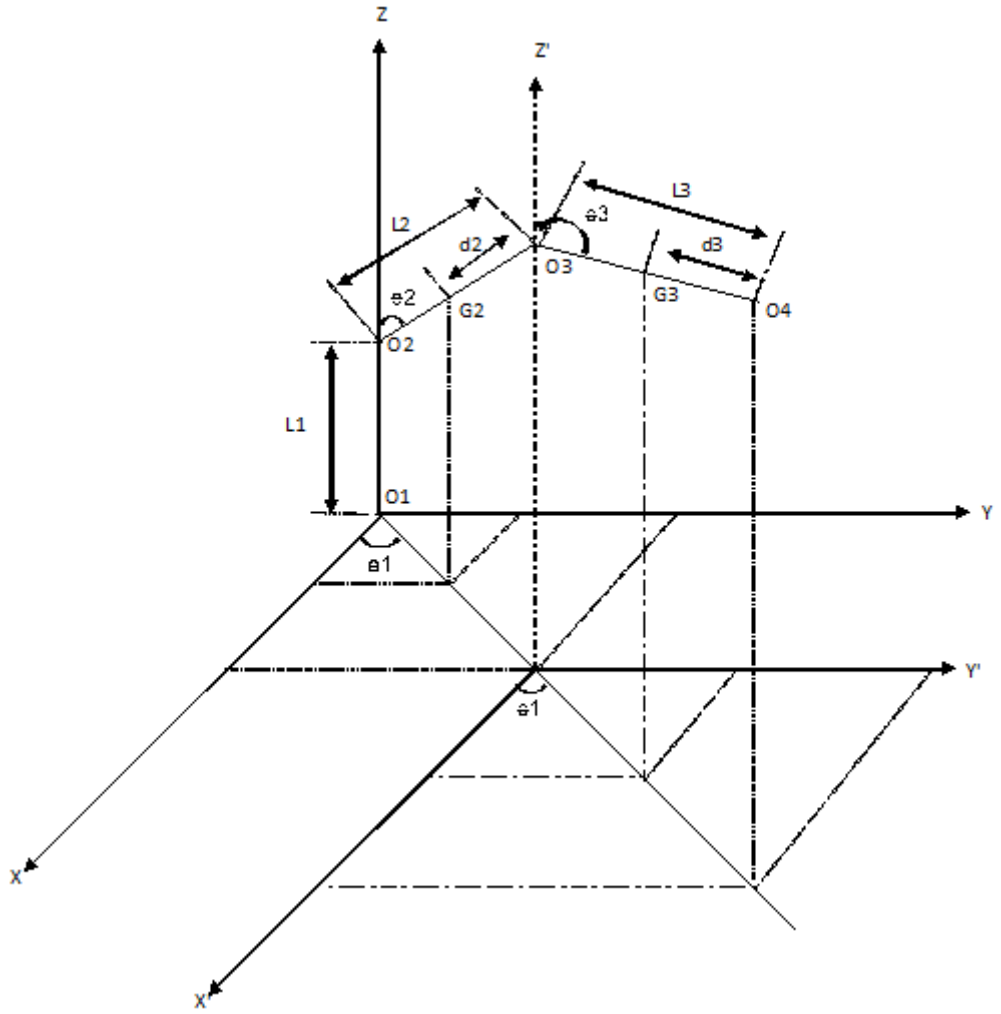


Figure II.4: schéma représente les cordonnée des positions du bras

II.4.3 Mise en équation de notre modelé :

II.4.3.1 Equations de mouvements :

Les liaisons rotoïdes sont parfaites (pas de jeux ni frottements).

Les coordonnées de G_2 sont :

$X_{G2}= d_2.\sin(\theta_2).\cos(\theta_1)$

$Y_{G2}= d_2.\sin(\theta_2).\sin(\theta_1)$

$Z_{G2}= d_2.\cos(\theta_2) + l_1$

Les coordonnées de G_3 sont :

$X_{G3}= d_3.\sin(\theta_3) + l_2.\cos(\theta_1).\sin(\theta_2)$

$Y_{G3}= d_3.\sin(\theta_3).\sin(\theta_1) + l_2.\ \mathrm{Sin}(\theta_1).\sin(\theta_2)$

$Z_{G3}= d_3.\cos(\theta_3) + l_1 + l_2 \cos(\theta_1)$

Les coordonnées de point O_4 sont :

$Xo_4= l_3.\sin(\theta_3).\ \mathrm{Cos}(\theta_1) + l_2 \cos(\theta_1).\sin(\theta_2)$

$Yo_4= l_3.\sin(\theta_3).\sin(\theta_1) + l_2.\ \mathrm{Sin}(\theta_1).\sin(\theta_2)$

$Zo_4= l_3.\cos(\theta_3) + l_1 + l_2 \cos(\theta_2)$

$\theta_1\theta_2\theta_3$ **:** sont des variables articulaires

G_2 : centre de gravité de l'épaule

G_3 : centre de gravité du code

L_1 : longueur de base

L_2:longuere de l'épaule

L_3:longuere du code.

On veut donner les masses et les moments d'inertie de chaque élément :

M1(le socle) :.1610 g

M2 (1erbras) :1050 g

M3 (2èmebras) :550 g

Socle : est assimilé comme un cylindre donc son moment d'inertie est :

$$J_1=M_1 (R^2+r^2)/2$$

1erbras : est assimilé à un segment, donc son moment d'inertie :

$$J_2 =1/3\ M_2L_2^2$$

2èmebras : est assimilé à un segment, donc son moment d'inertie :

$$J_3 =1/3\ M_3L_3^2 +M_4l_3^2$$

II.5 Conclusion:

Le chapitre présent fait une description sur les différents modèles de la modélisation mécanique et on s'est intéressé surtout par notre modèle du bras manipulateur.

À propos de la présentation de la partie commande qui réunit la carte de commande et le type d'actionneur utilisé feront l'objet du chapitre suivant.

Chapitre III: Arduino et Servomoteur

III.1 INTRODUCTION:

Le "système Arduino" est un projet créé par une équipe de développeurs, composée de six individus. Ce système est un outil qui va permettre aux débutants, amateurs ou professionnels de créer des systèmes électroniques plus ou moins complexes. Le système Arduino nous permet de réaliser un grand nombre de choses, qui ont une application dans tous les domaines.

Nous voulons aussi présenter dans ce chapitre les servomoteurs récents. Aujourd'hui dans le monde robotique on trouve beaucoup d'application qui sont réalisées à l'aide des actionneurs électriques de type servomoteur car ils sont facile pour la commande avec une précision élevée et délivre un couple considérable.

III.2 ARDUINO:

III.2.1 DEFINITION:

L'Arduino est une plateforme open source d'électronique programmée qui est basée sur une carte à microcontrôleur et un logiciel. Plus simplement, on peut dire que l'Arduino est un module électronique, doté d'un microcontrôleur programmable, il peut être utilisé pour développer des objets interactifs, munis d'interrupteurs ou de capteurs, et peut contrôler une grande variété de lumières, moteurs ou toutes autres sorties matérielle.

III.2.2 Microcontrôleur:

Un microcontrôleur est un circuit intégré (ou IC, integrated Circuit), qui rassemble sur une puce plusieurs élément complexes dans un espace réduit, au temps des pionniers de l'électronique, on soudait un grand nombre de composants encombrants, tels que les transistors, les résistances ou les condensateurs sur des cartes plus ou moins grandes. Aujourd'hui, tout peut loger dans un petit boîtier en plastique noir muni d'un certain nombre de broches. Ces dernières sont les connexions du circuit intégré un moyen desquelles s'effectue la communication.

La figure montre un microcontrôleur **ATmega328**, qu'on trouve sur la carte Arduino. **[10]**



Figure III.1: Microcontrôleur ATmega328 (source: Atmel).

III.2.3 Les différentes cartes Arduino:

Les cartes Arduino doivent satisfaire des exigences diverses et variées. Certains utilisateurs souhaiteront effectuer du prototypage et tester de nouveaux montages ou idées, la carte Arduino Uno disposant d'assez d'entrées-sorties pour

les projets d'envergure raisonnable. De part sa taille, la carte devrait aussi pouvoir être utilisée à l'avenir dans des projets plus ambitieux. D'autres auront besoin d'un grand nombre de ports afin de pouvoir raccorder de nombreux capteurs ou actionneurs. Une troisième catégorie d'utilisateurs ne cherchera qu'à attirer l'attention et à transmettre des signaux à leurs semblables à l'aide de diodes clignotantes.

La forme, la taille ou les possibilités de connexion jouent un rôle décisif dans le choix de la carte adaptée. C'est pourquoi les développeurs d'Arduino ont mis au point un vaste choix de cartes à microcontrôleur afin que chacun trouve le modèle qui répondre à ses besoin.

- Arduino uno
- Arduino Leonardo
- Arduino Mega 2560
- Arduino Esplora
- Boarduino V2.0
- Arduino Nano
- Arduino LilyPad
- Arduino Due
- Arduino Yûn

III.2.4 La carte Arduino Uno:

La pièce maîtresse de tous les montages est : la carte à microcontrôleur Arduino

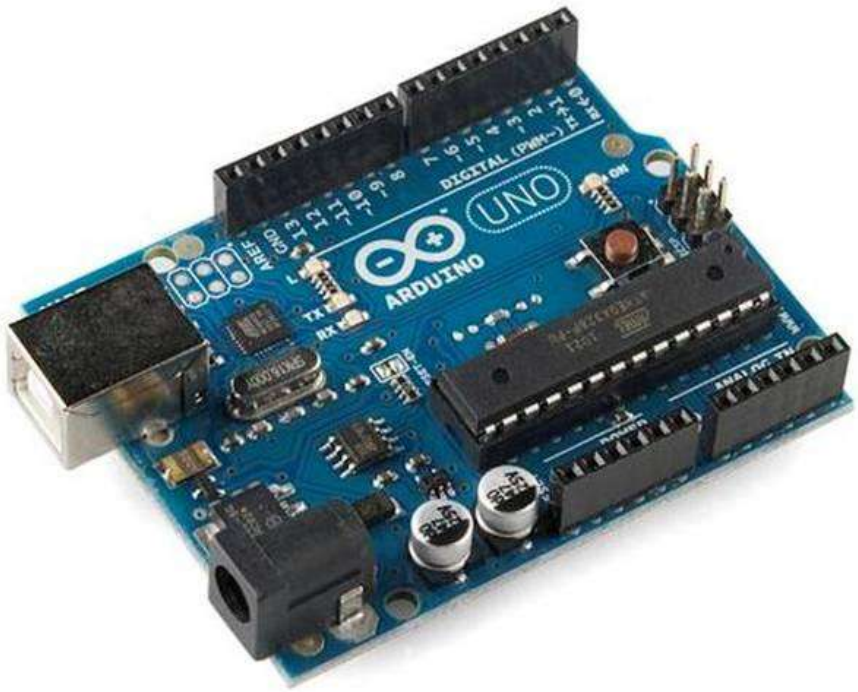


Figure III.2 : La carte à microcontrôleur Arduino.

Sur cette image, vous ne pouvez évidemment pas vous rendre compte à quel point les dimensions de la carte Arduino sont réduites (environ 7 cm de large et 5 cm de long); elle est vraiment très maniable, tient sans problème dans une main et s'avère vraiment compacte.

Au fil des années, plusieurs cartes Arduino ont été développées. Nous avons choisi de prendre l'Uno, car c'est la plus populaire.

Les composants les plus divers y sont reconnaissable (voir **FIGURE III.3**)

Certains penseront certainement qu'on ne peut pas concevoir quelque chose de

sérieux sur une surface aussi réduite. Mais grâce à la miniaturisation des composants ces dernières années, ce qui nécessitait auparavant cinq puces électroniques n'en requiert aujourd'hui plus qu'une.

Le plus gros élément qui saute directement aux yeux est le microcontrôleur proprement dit. Il est de type **ATmega328**.

Elle convient tout particulièrement à ceux qui veulent faire leurs premiers pas dans le monde des microcontrôleurs.

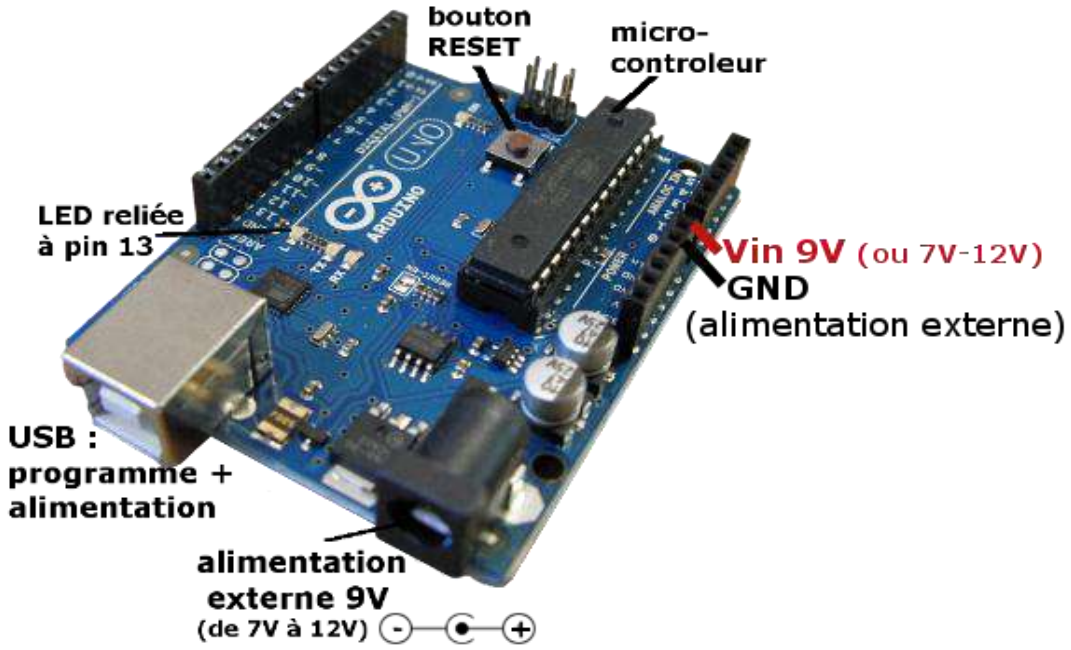


Figure III.3 : les composants de la carte Arduino.

Ces éléments sont les plus importants de la carte Arduino mais, bien entendu, cela ne veut pas dire que les autres sont à négliger.

Voici les principales caractéristiques de la carte Arduino :

- Microcontrôleur Atmega328.

- Tension de service 5V.
- 14 entrées et sortis numériques (6 sorties commutables en MLI).
- 6 entrées analogiques (résolution 10 bit).
- 32 Ko de mémoire flash (0.5 Ko occupé par le chargeur d'amorçage ou bootloarder).
- 2 Ko de SRAM.
- 1 Ko d'EEPROM.
- fréquences d'horloge 16 MHz.
- interface USB.

III.2.5 Les modes de communication:

Un microcontrôleur de type Arduino a déjà beaucoup de connexions qu'il s'agit de bien distinguer.

III.2.5.1. Le port USB:

Sans le port USB, vous ne seriez pas en mesure d'initialiser une communication.

Le travail avec la carte Arduino peut se diviser en deux étapes: le temps consacré à la mise en œuvre du montage et celui dédié à la programmation, appelé phase de développement (design time).

La programmation s'effectue dans un environnement de développement que vous allez apprendre à connaître très rapidement. C'est dans cet environnement que vous allez saisir le programme créé par vos soins pour le transmettre au microcontrôleur. Si tout s'est bien passé, le temps de l'exécution (run time) commence. Vous n'avez pas besoin de dire explicitement au microcontrôleur, il démarre en effet immédiatement après avoir reçu toutes les instructions de votre

part. Vous pouvez en outre échanger des données avec votre ordinateur via le port USB.

III.2.5.2. Les ports d'entrée ou de sortie (E/S):

Les ports E/S représentent l'interface du microcontrôleur. Il existe plusieurs chemins ou canaux pour échanger des données, comme chez l'Homme avec les yeux, les oreilles et la bouche. Il se produit, Grâce et par le biais de ces canaux de communication, une interaction avec l'environnement.

III.2.5.2.1. les ports d'entrée et de sortie sur la carte Arduino:

Quand vous la tenez de manière à pouvoir lire l'inscription <<UNO>> vous verrez les ports d'entrée et de sortie sur le bord supérieur (un bloc de dix broches et un autre de huit broches).

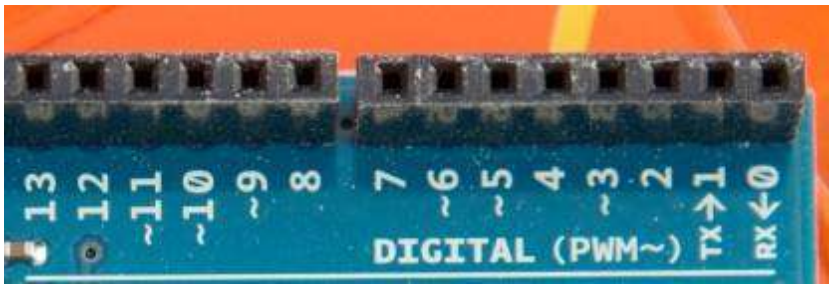


Figure III.4 : Entrée et Sortie numérique.

Il est bien entendu important de savoir pour chaque port quelle est son adresse afin de pouvoir communiquer avec lui au cours de la programmation. C'est pour cela que chaque broche possède un numéro.

Sous certains chiffres se trouve un tilde (~) indiquant que la broche peut être commutée en sortie analogique. Il s'agit là d'une broche MLI (rappelez-vous la modulation de largeur d'impulsion, on y reviendra un peu plus loin).

À l'extrémité inférieure de la carte se situent les ports d'alimentation (à gauche) et les ports d'entrée analogique (un bloc de huit broches et un autre de six broches).

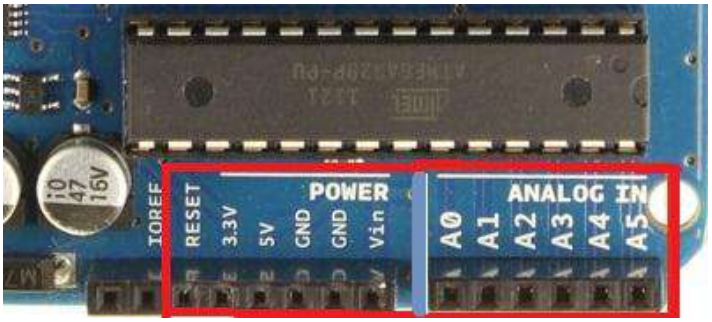


Figure III.5 : Alimentation et entrée Analogiques.

III.2.6. L'alimentation électrique:

La carte Arduino Uno doit être alimentée en énergie pour pouvoir travailler.

Cette alimentation peut s'effectuer tout d'abord via l'interface USB qui relie la carte à l'ordinateur-ce chemin sert aussi à l'échange de données entre la carte et l'ordinateur. En phase de développement avec votre Arduino, la connexion USB va servir d'alimentation primaire de la carte.

La seconde possibilité consiste à brancher une batterie ou bloc secteur au connecteur, appelé prise jack. Vous pouvez, par exemple, employer cette variante si vous avez construit un engin manœuvrable, commandé par la carte Arduino. Le Véhicule doit pouvoir évoluer librement dans l'espace, sans câble. En effet, l'utilisation d'un câble USB, généralement trop court, limiterait alors la mobilité de l'engin. L'emploi d'une batterie rend le dispositif autonome.

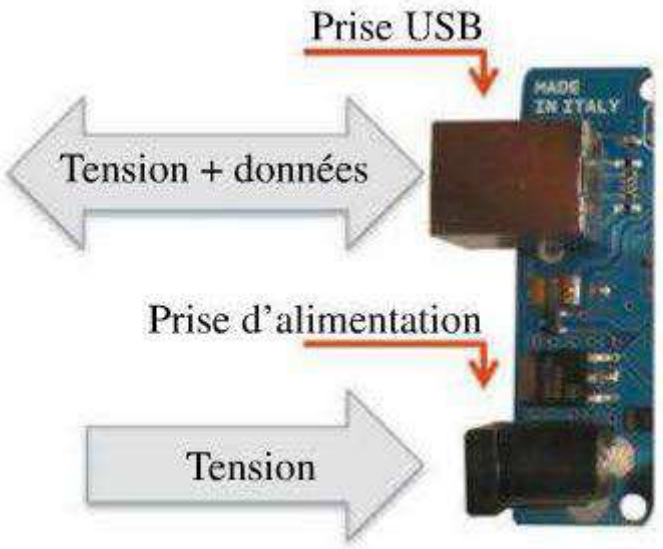


Figure III.6: Alimentation de la carte Arduino.

Nous montrons ici les différentes prises. Attention, elles ne peuvent pas être interverties, car elles ont des formes et des fonctions différentes.



Figure III.7: les différentes prises.

Dès qu'il s'agit de courant ou de tension, il convient de consulter le tableau.

III.2.7.Valeurs de courant ou de tension:

Catégorie	Valeur
Tension de service	**5 V (DC)**
Alimentation depuis l'extérieur	**7-12 V(DC)**
(recommandée)	**6-20 V(DC)**
Alimentation depuis l'extérieur (valeur limité)	**40 mA**
Courant continu par broche (maximal)	

Tableau III.1 : Valeurs de courant ou de tension**.**

L'interface USB peut fournir un courant maximal de 500mA; c'est en principe suffisant pour réaliser la plupart des circuits d'essai. Elle est protégée contre le court-circuit et les courants forts grâce à un poly fusible. Mais attention, cela ne doit pas vous empêcher de construire votre circuit avec le plus grand soin.

III.2.8 PROGRAMMER ARDUINO

III.2.8.1 Le Langage De Programmation:

Un langage de programmation est un langage permettant à un être humain d'écrire un ensemble d'instructions (code source) qui seront directement converties en langage machine grâce à un compilateur (c'est la compilation). L'exécution d'un programme Arduino s'effectue de manière séquentielle, c'est-à-dire que les instructions sont exécutées les unes à la suite des autres.

III.2.8.2 La Structure D'un Programme:

Un programme Arduino comporte trois parties :

1) la partie déclaration des variables (optionnelle)

2) la partie initialisation et configuration des entrées/sorties : la fonction **setup ()**

3) la partie principale qui s'exécute en boucle : la fonction **loop ()**

Dans chaque partie d'un programme sont utilisées différentes instructions issues de la syntaxe du langage Arduino.

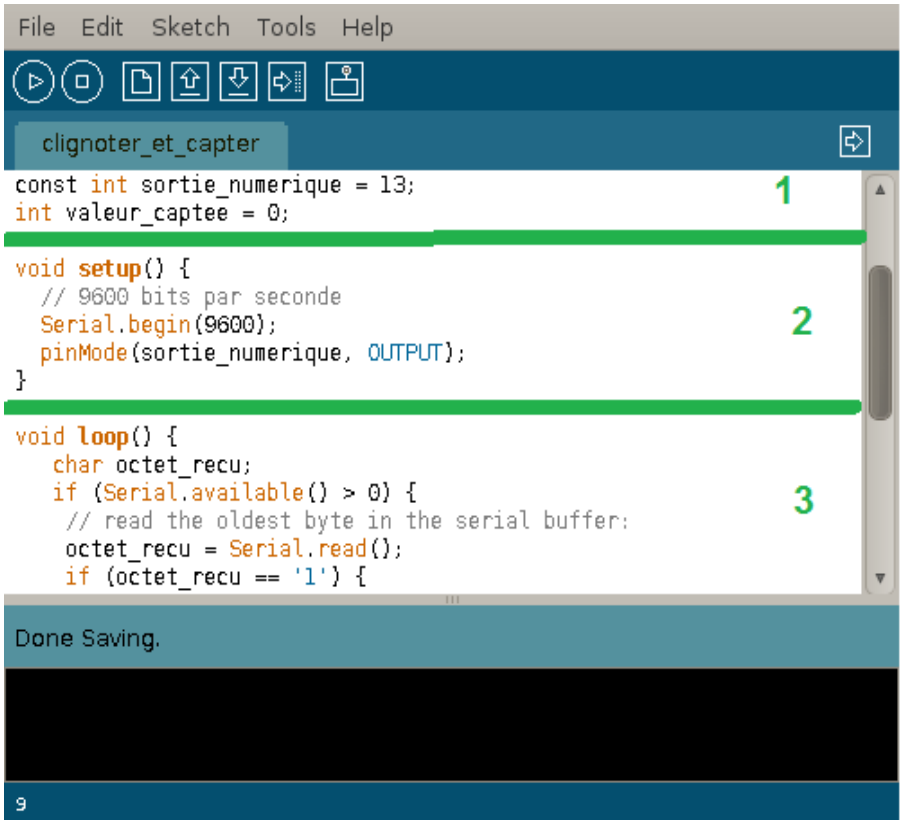


Figure III.8: Coloration Syntaxique**.**

Lorsque du code est écrit dans l'interface de programmation, certains mots apparaissent en différentes couleurs qui clarifient le statut des différents éléments :

- En **orange**, apparaissent les mots-clés reconnus par le langage Arduino comme des **fonctions** existantes. Lorsqu'on sélectionne un mot coloré en orange et qu'on effectue un clic avec le bouton droit de la souris, l'on a la possibilité de choisir « Find in reference » : cette commande ouvre directement la documentation de la fonction sélectionnée.
- En **bleu**, apparaissent les mots-clés reconnus par le langage Arduino comme des **constantes**.
- En **gris**, apparaissent les commentaires qui ne seront pas exécutés dans le programme. Il est utile de bien commenter son code pour s'y retrouver facilement ou pour le transmettre à d'autres personnes. L'on peut déclarer un commentaire de deux manières différentes : dans une ligne de code, tout ce qui se trouve après « // » sera un commentaire. l'on peut encadrer des commentaires sur plusieurs lignes entre « **/*** » et « ***/** ».

III.3 SERVOMOTEUR

III.3.1 Définition :

Les servomoteurs sont des actionneurs très utilisés en modélisme et dans l'industrie, ils ont comme caractéristique principale leur « couple », c'est-à-dire la force de rotation qu'ils peuvent exercer. Plus un servomoteur aura de couple et plus il pourra actionner des « membres » lourds comme déplacer un bras qui porte une charge.

Pour la robotique de loisirs, les servomoteurs ont en général peu de couple et sont de taille réduite, bien adaptée à un encombrement minimal et à une énergie disponible limitée.



Figure III.9 : les différents types de servomoteur**.**

Les servomoteurs sont pilotés par un fil de commande et alimentés par deux autres fils. Habituellement, ces 3 fils sont rassemblés dans une prise au format.



Figure III.10: Les Trois Files.

Un fil rouge est relié à l'alimentation positive (+5 ou +6 V selon le servo), le fil noir est relié à la masse (GND) et le fil jaune est utilisé pour la commande. Il y aurait beaucoup à dire sur le fonctionnement d'un servomoteur, ses composants, son moteur et le petit potentiomètre qui permet de connaître sa position mais cette fiche va droit au but et se limitera à son utilisation avec l'Arduino. **[11]**

III.3.2 Principe de l'asservissement:

Ce schéma explique le principe de l'asservissement d'un servomoteur. Le but de ce montage est de réglé la position angulaire du moteur avec un potentiomètre de réglage P1.

Ce montage électronique réalisé à partir d'AOP essaye en permanence que P2 soit égale à P1. Pour cela il faut effectuer une comparaison avec l'aide d'un AOP monté en comparateur.

Il faut deux potentiomètres de même valeur, R1 = P1/5 et R2 est à testé.

Si R2 est trop grand il y aura pas de précision et si trop faible instable. **[12]**

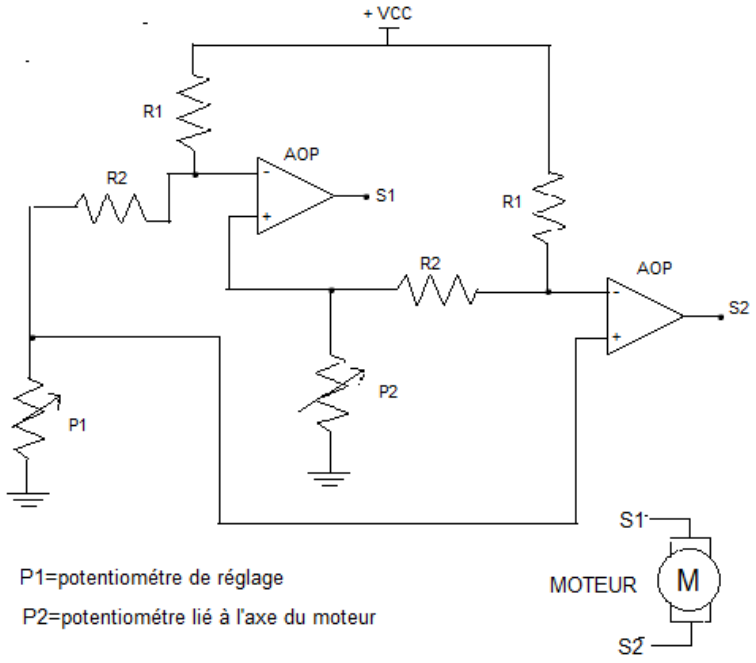


Figure III.11: Principe de l'asservissement.

- Retour d'informations possibles (nécessite modification du servomoteur)
- Retour de la position du palonnier du servomoteur.
- Retour du courant avec une résistance.

III.3.3 Le servomoteur est un actionneur :

Il permet de réaliser des déplacements en translation ou en rotation sur des modèles réduits.

Il existe une grande variété et qualité de servomoteur se différenciant par leur vitesse de rotation/seconde, par leur course en degré et par leur couple exprimé en kg/cm.

Ils se distinguent également par la matière utilisée et le montage éventuel sur roulements. **[13]**

III.3.4 Fonctionnement d'un servomoteur standard:

Mécaniquement, il s'agit d'un moteur avec réducteur. Le mouvement de sortie est une rotation. Lorsque le moteur tourne, l'axe du servomoteur change de position, ce qui modifie la résistance du potentiomètre. Le rôle de l'électronique est alors de commander le moteur pour que la position de l'axe de sortie soit conforme à la consigne reçue.

La commande se fait via une petite impulsion dont la durée est comprise entre 0,5ms et 2,5ms. La valeur de 1,5ms donne au servomoteur la position centrale. Les ordres de positions sont transmis sous forme d'un signal codé en largeur d'impulsion, répété périodiquement, en général toutes les 50 ms, ce qui permet à l'électronique de contrôle et de corriger continuellement la position angulaire de l'axe de sortie (en général, course limitée à 90° ou 120°).

L'augmentation de l'amplitude des impulsions agit sur le contrôle de la vitesse jusqu'au moment où le bras du servomoteur se trouve dans la position souhaitée. Durant la rotation, le potentiomètre indique au circuit électronique le moment où la position désirée est obtenue. Les impulsions diminuent l'amplitude jusqu'à ce que plus aucune tension soit appliquée au moteur du Servomoteur tout en conservant le bras dans sa nouvelle position.

III.3.4.1 Inconvénient :

L'impulsion est forte et instantanée, entrainant aucune continuité de la rotation initiée comme pourrait le faire une mise sous tension continue. Les petites impulsions données au moyen de la manette de commande ne produisent que de faibles effets. Ils peuvent se révéler inefficaces et sont les causes de l'importante zone neutre qui caractérise les servomoteurs traditionnels.

III.3.5 Les composants de servomoteur:

1) Un moteur électrique CC

2) Une réduction en sortie de ce moteur pour avoir du couple

3) Un potentiomètre qui induit une résistance variable en fonction de la position angulaire de sortie, donc le potentiomètre est lié à l'axe du moteur.

4) Un asservissement électronique pour contrôler la position angulaire

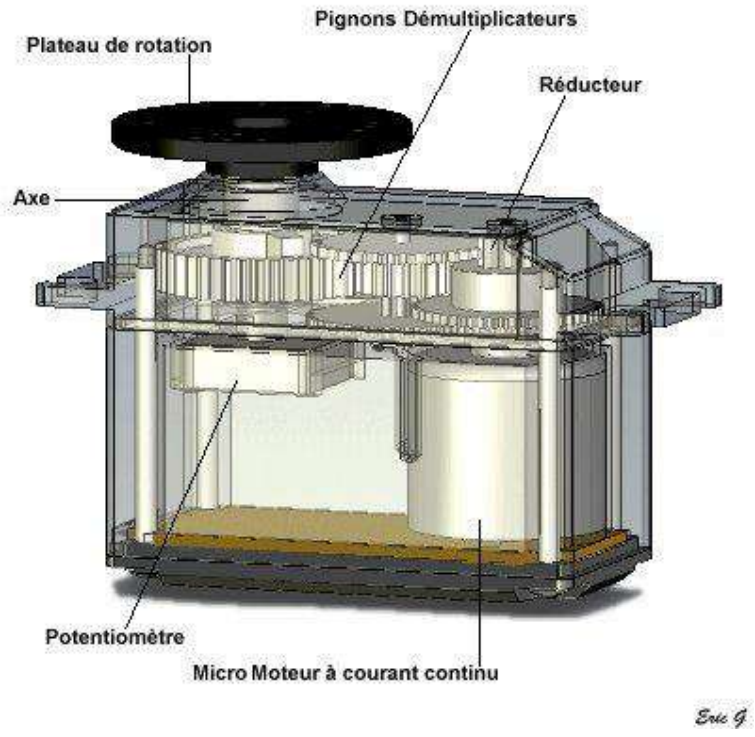


Figure III.12: Les composants de servomoteur.

III.3.6 Commande théorie:

Le signal de commande d'un servomoteur est un signal PWM de fréquence 50Hz et dont le niveau haut doit être entre 1 et 2ms (en théorie).

- 1ms : -90° en angle de sortie
- 1.5ms : 0° en angle de sortie
- 2ms : +90° en angle de sortie

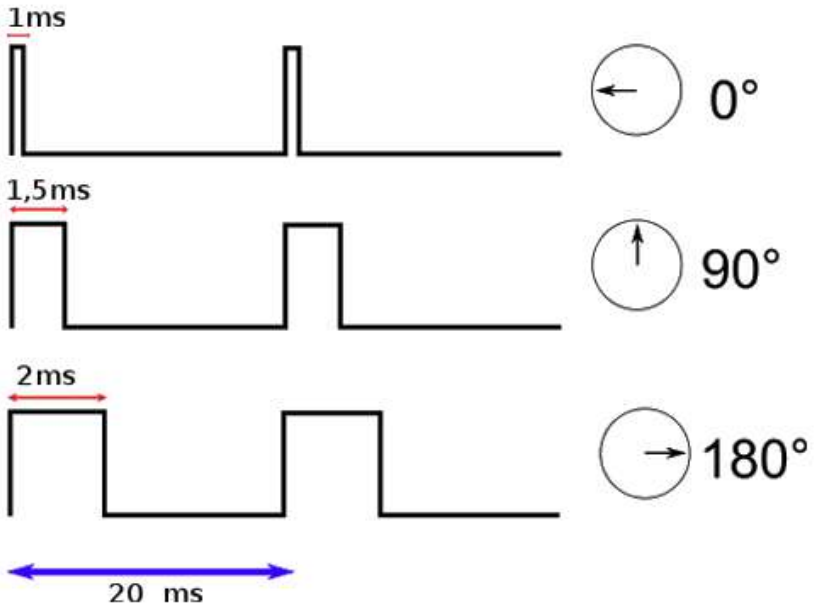


Figure III.13 Chronogramme de commande de Servomoteur.

Le déplacement angulaire total est de 180°.

III.3.7 Commande en pratique:

La commande d'un servomoteur en pratique est effectuée par un microcontrôleur avec l'utilisation:

- Périphérie PWM

- Périphérie Timer
- Ou utilisation des fonctions de type delay ms.

III.3.8 Différences entre servomoteurs digital et analogique:

Outre le fait qu'il contient le même moteur, les mêmes engrenages que les analogiques et disposent eux aussi d'un potentiomètre, le servomoteur digital se différencie du servomoteur analogique par le fait qu'il incorpore un microprocesseur qui analyse le signal reçu du récepteur et contrôle le moteur du servomoteur.

La différence principale est donc la façon de traiter l'information reçue du récepteur et le contrôle exercé sur la tension électrique appliquée au moteur , ce qui permet de réduire la zone neutre , augmentant ainsi l'amplitude du mouvement et générant une grande stabilité du positionnement statique .

III.3.8.1 Avantages du servomoteur digital :

- Des impulsions de plus grande fréquence : 300 impulsions par seconde contre 50 impulsions par seconde dans le cas d'un servomoteur analogique.
- Le moteur réagit mieux à l'ordre donné.
- Une diminution de la zone neutre du servomoteur entrainant une meilleure rapidité de réponse, une accélération rapide, un freinage progressif.
- Une meilleure stabilité dans le positionnement statique.

III.3.8.2 Inconvénient du servomoteur digital :

- Une plus grande consommation d'énergie.

III.3.9 Commander un servomoteur :

III.3.9.1 Piloter un servomoteur avec l'Arduino:

Le mode de commande d'un servomoteur est standardisé : on envoie sur son fil de commande une impulsion dont la durée correspond à l'angle désiré. Historiquement, cette impulsion était délivrée par un circuit oscillateur. Le circuit intégré NE555 est un exemple vedette du circuit utilisé.

Avec la programmation de l'Arduino, ce circuit n'est plus nécessaire. Une bibliothèque (library) dédiée, la bibliothèque « servo », permet de piloter un servomoteur en lui transmettant simplement l'angle sur lequel il souhaite se positionner.

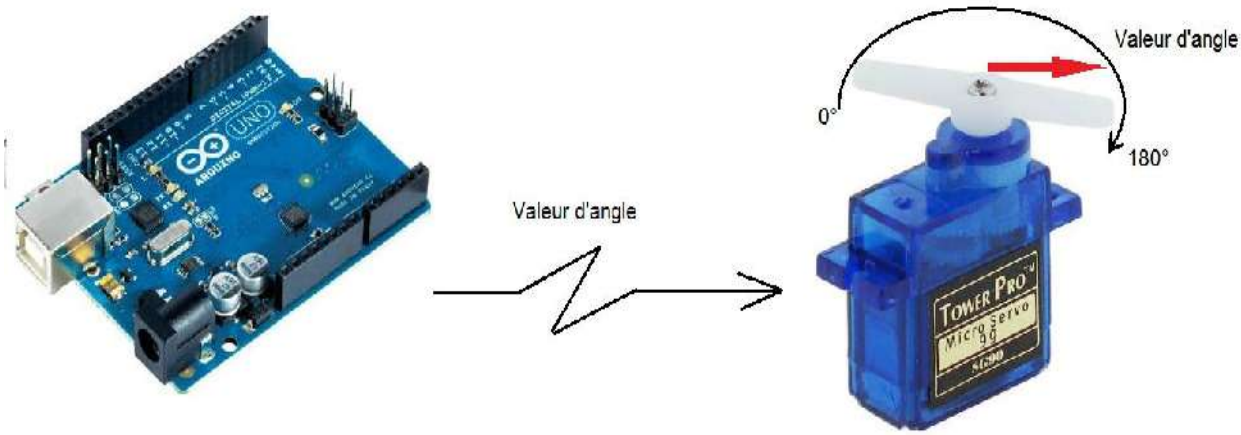


Figure III.14 Piloter un servomoteur avec l'Arduino.

Ce qui permet de réaliser simplement des mouvements qui peuvent être complexes et de les automatiser. **[14]**



Figure III.15: Piloter des servomoteurs avec l'Arduino**.**

III.3.9.2 Les connaissances nécessaires au pilotage d'un servomoteur :

#include <Servo.h>	L'utilisation de la librairie servo.h qui prend en charge la communication de bas niveau entre l'Arduino et le servomoteur. Les commandes précédées par # sont des ordres particuliers donnés au compilateur
Servo myservo;	La création d'un objet de type **Servo**, appelé ici myservo

myservo.**attach**(9);	Le fil de commande de ce servo sera connecté au PIN 9 et l'objet myservo pilotera ce PIN
myservo.**write**(90); myservo.**write**(Angle);	Demander au servomoteur de se déplacer à l'angle désiré, soit de façon absolue en lui indiquant une valeur entière (90°) dans le premier cas, soit en lui passant le contenu d'une variable (Angle) compris entre 0 et 180, ce qui peut être utile par exemple pour donner une progressivité au déplacement en faisant varier l'angle d'un pas fixe (quelques degrés) par une boucle
myservo.**read**(Angle);	Pour lire la valeur de l'angle du servomoteur

Tableau III.2: Les connaissances nécessaires au pilotage d'un servomoteur.

III.4 CONCLUSION:

En effet ce chapitre est divisé en deux parties, on a exposé dans la 1er partie les différents types de la carte Arduino précisément la carte Arduino Uno avec sa microcontrôleur et ses principales caractéristiques.

La 2ème partie rassemble le fonctionnement du servomoteur et ses deux modes de commandes (analogique, digital) en tenant la méthode de pilotage d'un servomoteur avec la carte Arduino.

Chapitre VI: Etude, réalisation, résultats et interprétations.

VI.1 Introduction :

Dans le cadre de notre projet de fin d'étude nous visons à concevoir et réaliser un prototype d'un robot très répandu (bras manipulateur) qui pourra imiter les gestes d'un être humain à travers d'une caméra de type **Kinect**, notons qu'il est beaucoup plus utilisé dans différents domaines à savoir le domaine médical dans le bloc opératoire ; le domaine de produit chimiques dangereux etc.

La concrétisation de ce projet va suivre l'organigramme de la figure suivante :

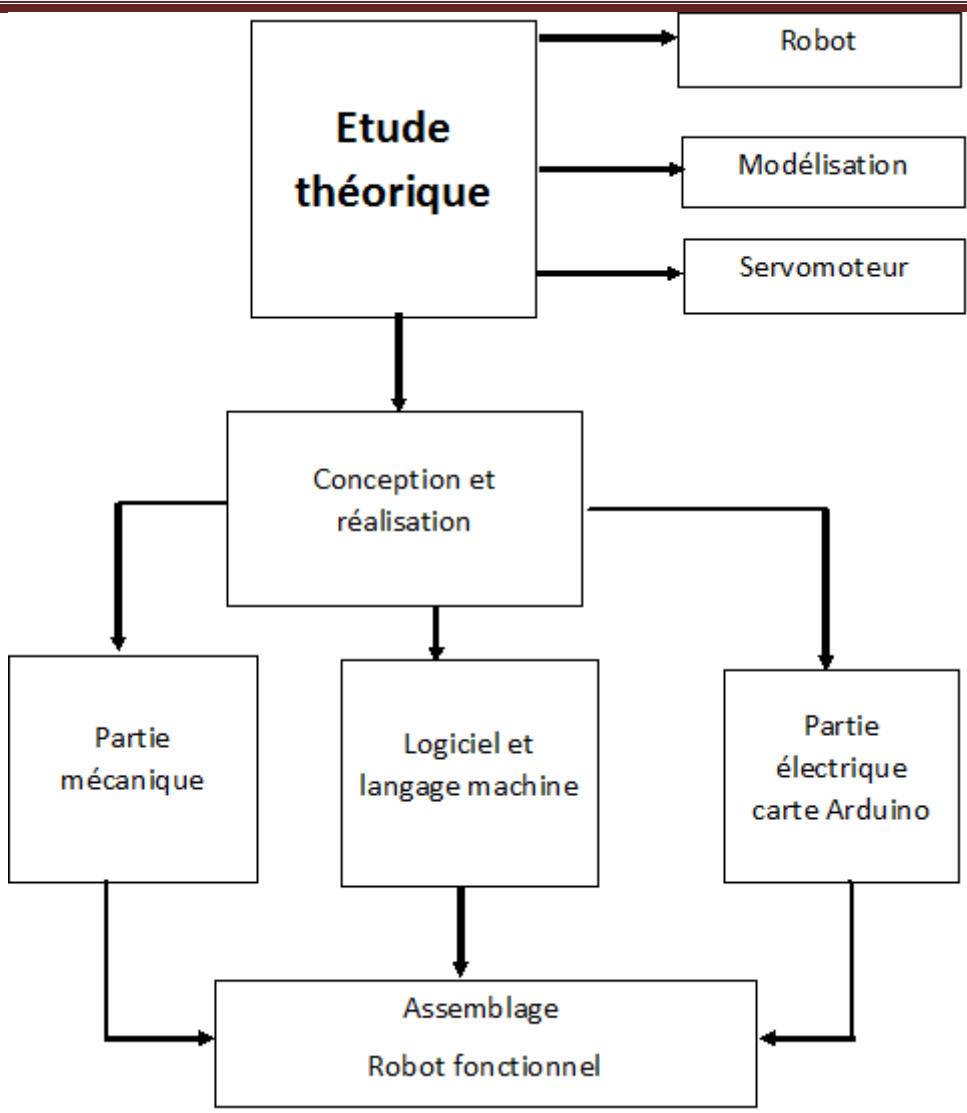


Figure VI.1: Méthodologie de la conception

VI.2. La compatibilité des différents paramètres du robot :

Il est indispensable de connaitre les paramètres essentiels qui gèrent le fonctionnement du robot, et de déterminer leurs actions sur les différentes parties du système avant de commencer la conception.

Pour concevoir et commander optimalement notre robot il est nécessaire de suivre les étapes suivantes :

VI.3. Etape 1 :

La partie mécanique, principalement elle est compose d'une base rotoïde qui supporte un porteur par lequel se termine par un organe terminale.

VI.3.1. Etude mécanique :

La conception mécanique permet de concevoir des produits mécaniques. La connaissance des matériaux, du dessin industriel, du calcul mécanique sont indispensables pour une bonne conception mécanique, mais dans notre cas une est complexité très grand.

Tout d'abord ; on a fait l'étude cinématique, dimensionnement des grandeurs des pièces et surtout le choix de la matière.

Nous avons basé notre choix sur des critères et norme bien précises parmi lesquelles :

- Poids des pièces légères

- Résistance des matériaux utilisé, c'est-à-dire qu'il ne subit pas de déformation ou flexion.

On réalité nous avons rencontré beaucoup de difficulté quand a commencé la réalisation, on n'a pas trouvé facilement la matière voulue, les moyens existants nous imposent de suivre une fabrication de pièce simple et d'autres obstacles en cours de l'assemblage.

Nous avons débuté notre travail par l'utilisation du logiciel SolidWorks qui nous a donné un grand pas à la conception fiable de notre « bras manipulateur ».

VI.3.2. SOLIDWORKS :

Conception innovante est au cœur de n'importe quel produit à succès. **SOLIDWORKS** mécanique conceptuel, propulsé par Dassault Systèmes 3D EXPERIENCE plate-forme, supprime les contraintes des programmes traditionnels de conception structurée avec un environnement de modélisation flexible, intuitif et puissant qui transforme votre avantage commercial de tous les défis de conception. **SOLIDWORKS** permet aux concepteurs de rapidement créer, innover et collaborer au cours de la phase de conception du concept. La solution offre un environnement de conception véritablement collaborative qui permet aux équipes de conception, clients et fournisseurs à communiquer, collaborer et développer des produits quand ils le veulent et où qu'ils se trouvent. Mélangeant les meilleurs des technologies de classe de Dassault Systèmes avec convivialité familier. **SOLIDWORKS** mécanique donne conception équipes flexibilité inégalée et des flux de travail intuitif pour créer des modèles avec vrais comportements réalistes pour comprendre comment les produits seront comportent dans le monde réel.

Notre model du « bras manipulateur » est divise en trois grandes parties :

- Socle.
- Porteur
- Organe terminal

VI.3.3. Les constituants du socle :

Le socle est la base de notre robot, il est constitué d'un châssis sous forme d'un cylindre divisé en deux comme une boite en raison d'éviter les frottements et pour bien fixer le porteur.

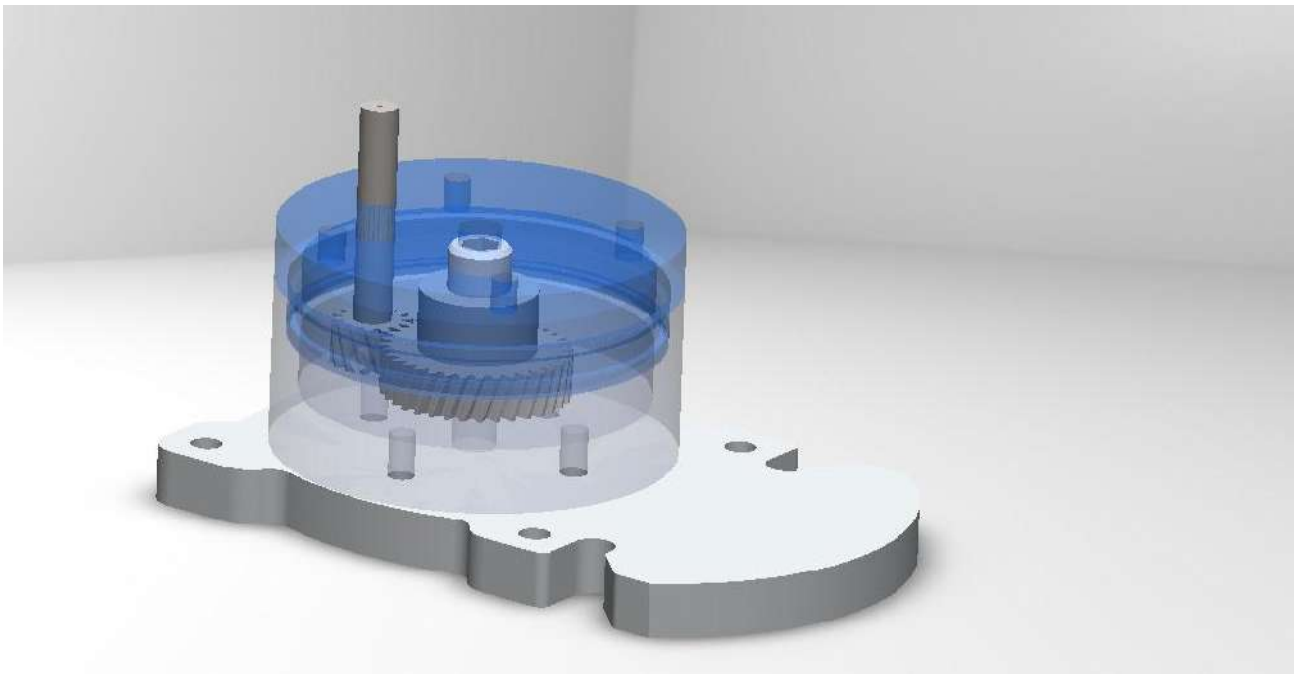


Figure VI.2 : socle du « bras manipulateur ».

Pour transmettre le mouvement du premier moteur, un système de transmission par engrenage est utilisé en vue de créer un mouvement de rotation. Le système est composé de deux pignons ; les deux sont accrochées, un grand est fixe, le $2^{ème}$ est mobile a été fait transaxle pour lier avec l'axe du premier moteur.

Le tableau suivant résume les principales dimensions des deux pignons :

	Nombre des dents	L'épaisseur de dent (mm)	Diamètre de base (mm) d=M.Z	Module	Diamètre extérieur (mm) d_a=d+2m
P_1	28	7	13	1	15
P_2	14	7	33	1	35

Tableau VI.1: les principales dimensions des deux pignons

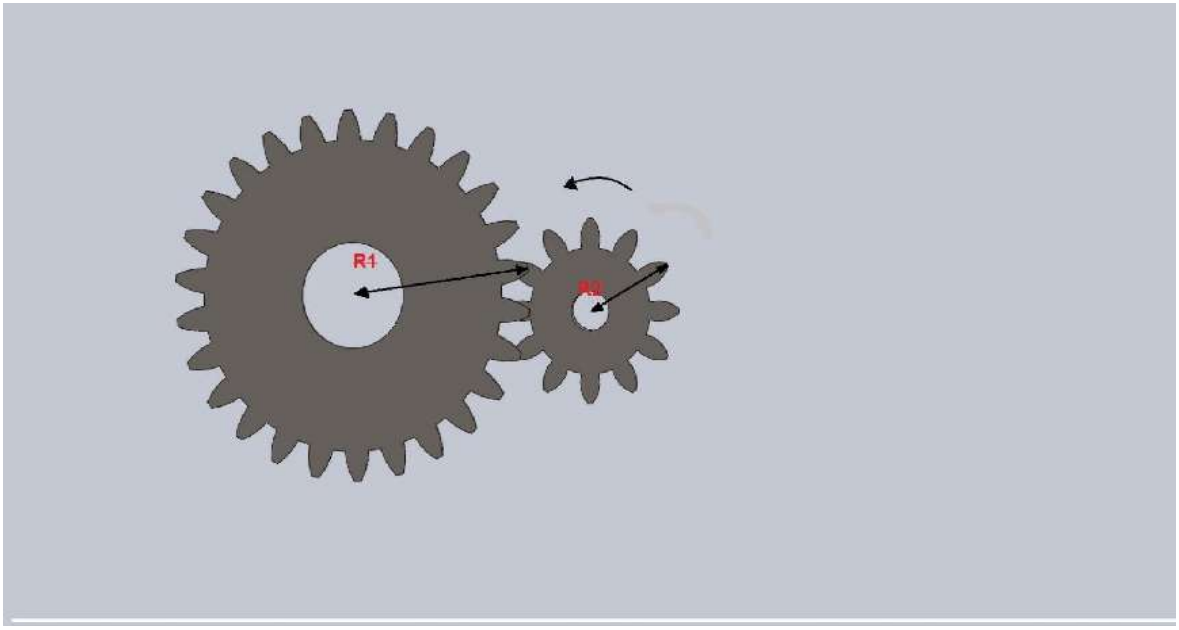


Figure VI.3 : Engrènement des deux roues.

VI.3.3.1. Détermination des caractéristiques :

Toutes les roues à denture hélicoïdale de même module (réel ou apparent) et de même angle d'hélice engrènent entre elles, quels que soient leur diamètre et leur nombre de dents, mais les hélices doivent être de sens contraire (l'une à droite et l'autre à gauche).

Module réel	m_n	Déterminé par la résistance des matériaux et choisi dans les modules normalisés (§ 73.12)**.
Nombre de dents	z	Déterminé à partir des rapports des vitesses angulaires : $\frac{\omega_A}{\omega_B} = \frac{n_A}{n_B} = \frac{z_B}{z_A}$
Angle d'hélice	β	Choisi habituellement entre 20° et 30°.
Sens de l'hélice « à droite » ou « à gauche »		Pour un même engrenage, les hélices des roues sont de sens contraire.
Module apparent	m_t	$m_t = m_n / \cos\beta$
Pas apparent	P_t	$P_t = m_t . \pi$
Pas réel	P_n	$P_n = m_n . \pi$ $\quad P_n = P_t . \cos\beta$
Pas de l'hélice primitive	P_z	$P_z = \pi_d / \tan\beta$
Saillie	h_a	$h_a = m_n$
Creux	h_f	$h_f = 1{,}25\ m$
Hauteur de dent	h	$h = h_a + h_t = 2{,}25\ m_n$ Fin du tableau page suivante.

TableauVI.2 : les caractéristiques d'une roue hélicoïdale

VI.3.4. Constituants du porteur :

Le porteur est divisé en deux parties le premier et le deuxième bras.



FigureVI.4 : porteur du « bras manipulateur ».

L'ensemble de ces deux bras constitue une masse repose sur la base, pour éviter tous frottement on a utilisé des axes lors de l'assemblage des pièces.

VI.3.4.1. Principe de transmission de mouvement du premier et deuxième bras :

Le premier « bras » repose sur l'axe commun entre les deux moteurs, le mouvement sera effectué à l'aide d'une came fixé sur l'axe du deuxième moteur lorsque le moteur est en état de marche.

La même chose pour le deuxième « bras », qu'il est reposé sur un axe qui joindre énormément de pièce ; son mouvement est à partir d'une tige, qui est en rotation avec l'ensemble came et bout du bras pour éviter le décallement.

Parmi l'avantage majeur qu'on a pu récoler, c'est que le système présente une bonne stabilité du mécanisme.

VI.3.5. L'organe terminal :

Il est formé comme un support de pincette mais malheureusement on n'a pas pu la réaliser à cause de manque du temps. Le guidage de ce poignet est en fonction des mouvements de rotation grâce à deux tiges, l'une fixe sur la plaque de la base et l'autre en bout du poignet.



FigureVI.5 : photo réelle de notre bras manipulateur réalisé

VI.4. Etape2 :

On parle dans cette étape sur la partie intermédiaire de visualisation, entre le circuit de commande et l'ordinateur. C'est la camera Kinect.

VI.4.1. La « Kinect » :

La Kinect de Xbox ou l'Asus Xtion pro, est un périphérique destiné à la console de jeux vidéo Xbox 360, permettant de contrôler des jeux vidéo sans utiliser de manette. Le nom « Kinect » est issu des mots anglais « kinetic ».

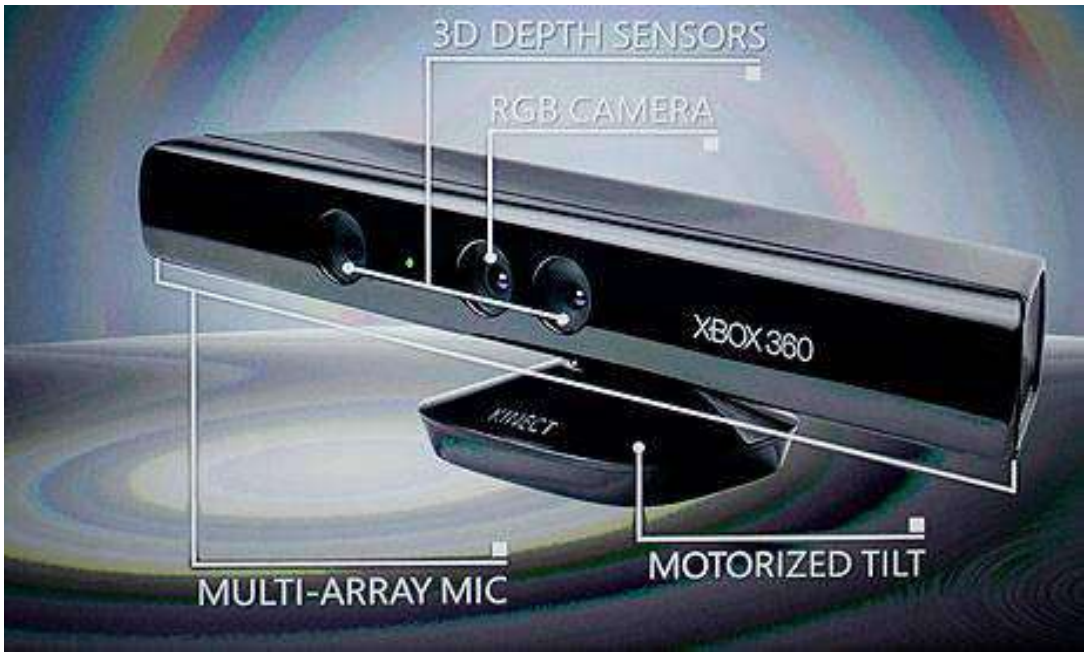
Il s'agit d'une caméra utilisant des techniques d'interaction développées par la société israélienne Prime Sens, longtemps nommée par le nom de code du projet, « Project Natal ». Elle est basée sur un périphérique d'entrée branché sur la console Xbox 360 qui permet d'interagir par commande vocale, reconnaissance de mouvement et d'image.

VI.4.2. Principe de Fonctionnement :

IV.4.2.1. Concept :

Ce périphérique de Microsoft est constitué d'une barre horizontale connectée à sa base via un petit moteur. Celui-ci permet à la caméra d'effectuer des petits mouvements vers le haut ou le bas (27° semble-t-il) afin d'adapter la perception de la caméra en fonction de votre position dans la pièce. La barre horizontale constitue l'élément principal de la technologie Kinect. Elle contient une série de multi-microphones, une caméra RGB, et enfin un capteur « 3D depth sensor » permettant d'améliorer l'analyse de mouvement par rapport à une caméra comme l'EyeToy. L'association de ces trois technologies devrait ainsi

permettre la reconnaissance vocale et faciale de l'utilisateur, ainsi que la « capture » de l'ensemble de la pièce (et donc du corps) en 3D.



FigureVI.6 : Un petit récapitulatif des technologies embarquées dans Kinect.

IV.4.3. Description de la méthode utilisée pour notre application :

La solution de développement Live Pro Xtion permet aux développeurs d'appliquer la dernière technologie de détection de mouvement dans diverses applications, industries, le marketing, l'orientation ou le tourisme et ainsi de suite. La réadaptation physique est aussi un bon terrain pour les applications de détection de corps. Il ya un certain nombre de domaines qui pourraient bénéficier de la technologie basée sur le mouvement, tels que l'éducation, médical, des conférences, des jeux et beaucoup plus.

Le kit de développement LIVE PRO Xtion est largement ouvert. Vous pouvez créer vos propres applications pour les affaires ou pour rendre la vie des gens plus facile et intuitive. N'importe qui peut développer leurs propres applications. Une application de grande envergure peut être créé et appliqué sur la base de l'imagination des développeurs



Figure IV.7: Kinect ASUS Xtion PRO LIVE RGB and Depth Sensor

IV.4.3.1. Principe :

Basé sur le même principe de fonctionnement que la Kinect XBOX 360 déjà expliqué. Le Xtion Pro Live utilise des capteurs infrarouges, la technologie de détection de profondeur adaptatif, image couleur détection et flux audio pour

capturer en temps réel d'image, le mouvement et la voix des usagers, ce qui rend le suivi des utilisateurs plus précis. Le PRO solution de développement LIVE Xtion est livré avec un ensemble d'outils de développement pour le rendre plus facile pour les développeurs de créer leurs propres applications gestuelles sans la nécessité d'écrire des algorithmes de programmation complexes.

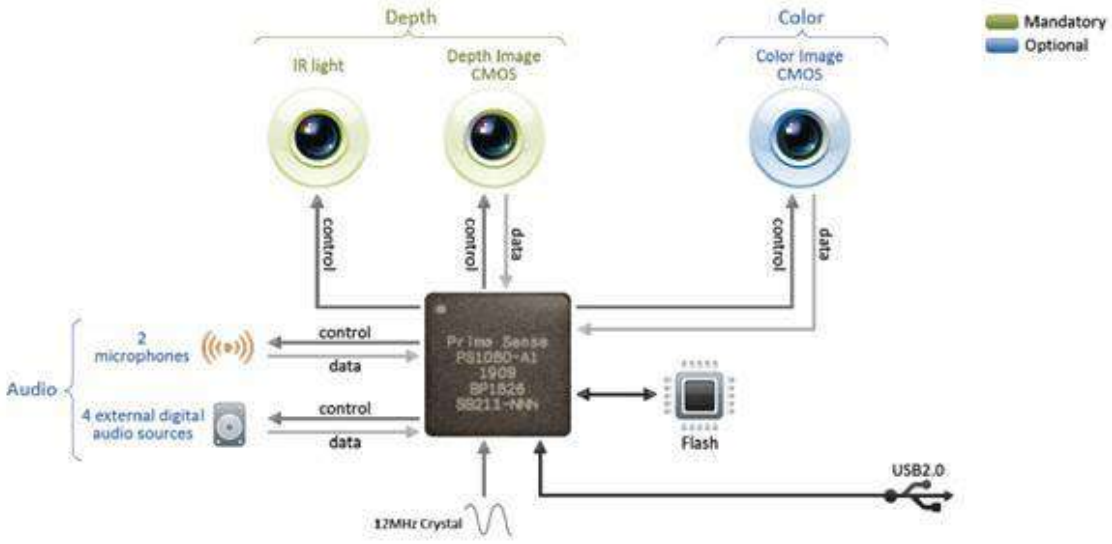


Figure IV.8: Kinect ASUS Xtion PRO LIVE RGB

- **Détection de Geste:** Le PRO solution de développement LIVE Xtion enregistre les mouvements de la main des gens sans aucun retard, qui transforme votre main dans un contrôleur. Il a plus de 8 poses prédéfinies pour vous permettre de pousser, clic, cercle, vague et beaucoup plus parfait pour une utilisation, commandé une interface utilisateur. Basé sur cette fonction, il peut être largement développé dans divers types d'applications, un gain de temps et d'efforts pour le développement de logiciels de développeur.
- **Toute détection de corps*:*** Le PRO solution de développement LIVE Xtion permet aux développeurs de suivre tout mouvement du

corps des usagers, ce qui le rend idéal pour tout jeu de corps, tout en soutenant la reconnaissance de lecteur multiples.

- **Camera RGB**: Xtion Pro Live permet la détection des couleurs (RVB) d'image. Xtion PRO LIVE peut capturer l'image des utilisateurs, ce qui est utile pour la détection humaine, la signalisation numérique, système de sécurité et d'autres applications à être créés.
- **Audio:** le flux audio permet de soutenir la commande vocale et d'autres applications de reconnaissance vocale. La vidéoconférence au bureau et à domicile sont également possibles avec la fonction RVB.
- **Facile à utiliser :** Le Xtion Pro Live dispose d'une prise facile et jouer conception USB, avec le Xtion PRO développement, LIVE solution OPNI NITE middleware compatible.



FigureVI.9: Schéma résumant l'ensemble des technologies Kinect

IV.4.4 L'organigramme de fonctionnement de Xtion Pro:

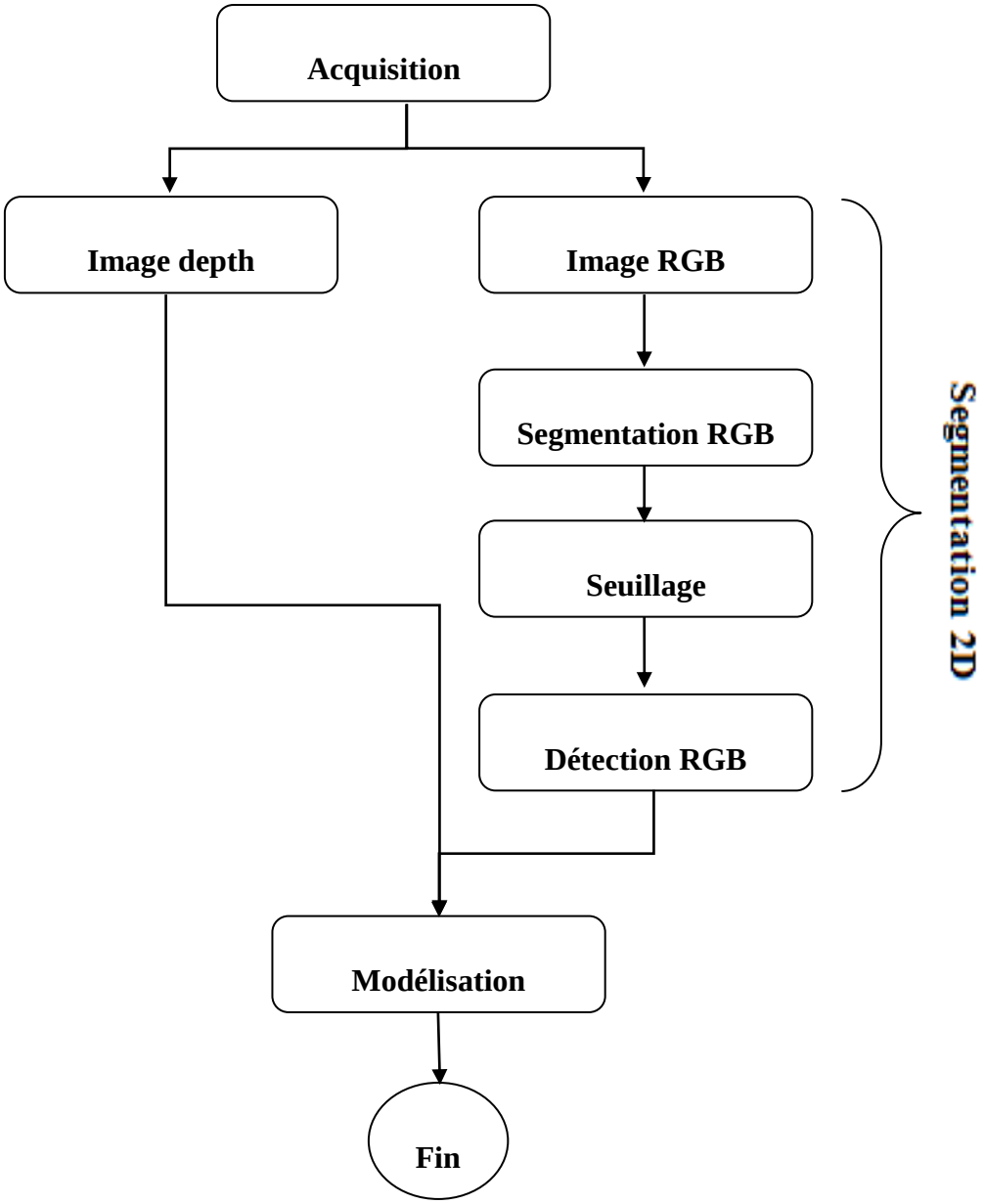


Figure VI.10: L'organigramme de fonctionnement de la Kinect.

IV.5. Etape3 :

Dans l'étape 3 on décrit le circuit de commande avec tous les servomoteurs nous désirons dire l'adaptation de la carte Arduino avec différentes parties opératives.

IV.5.1 Arduino Uno:

Nous avons choisi Arduino pour des bonnes raisons, Il existe pourtant une multitude des plateformes qui permettent de faire la même chose. Notamment les microcontrôleurs « PIC » du fabricant Micro chip. Ces raisons sont les suivants:

- **Le prix :**

Les cartes Arduino sont relativement peu couteuses, ce qui est un critère majeur de les utiliser.

- **Le logiciel :**

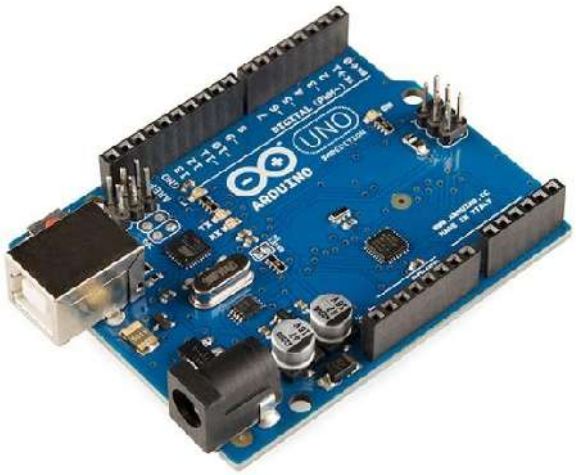
Gratuit et open source, développé en « Java », dont la simplicité d'utilisation relève du savoir cliquer sur la souris.

- **La compatibilité :**

Le logiciel, tout comme la carte, est compatible sous les plateformes les plus courantes (Windows, Linux et Mac)

- **La communauté** :

La communauté Arduino est impressionnante et le nombre de ressources à son sujet est en constante évolution sur internet.



FigureVI.11 : présentation de la Arduino Uno.

IV.5.2. Servomoteur de référence HS-5765 MH:

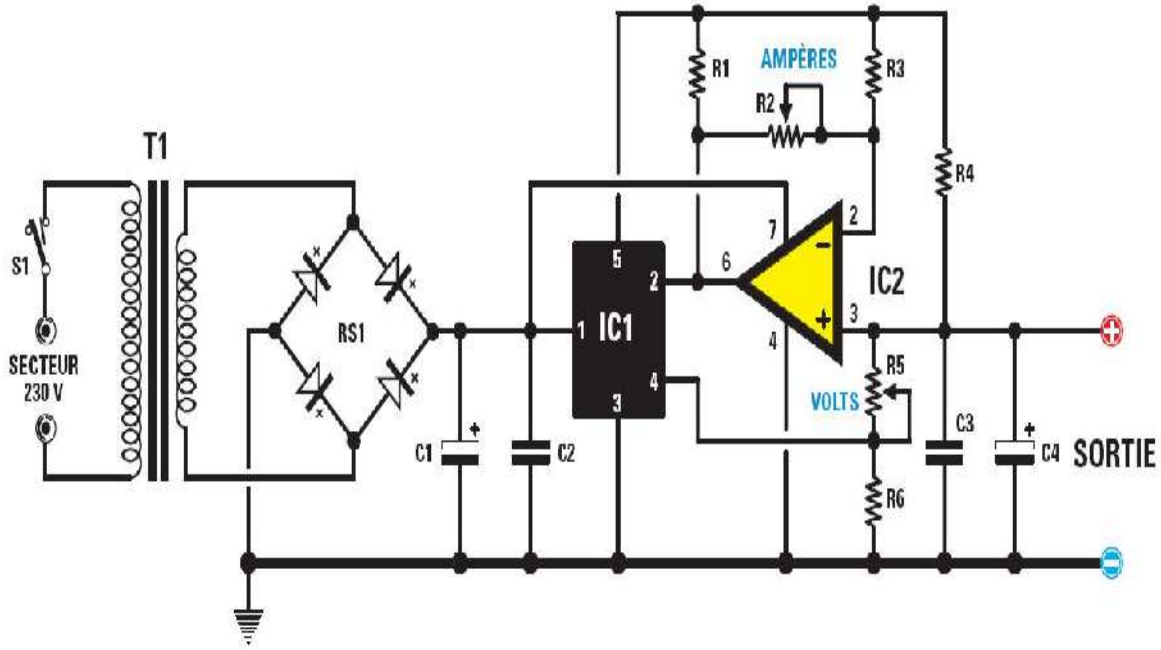
On a choisi ce type d'actionneur car il est facile pour la commande avec une précision élevée et peut supporter des poids, il a des engrenages en métal avec un arbre de sortie de 10mm, avec un couple considérable de 25Kg-cm, le HS-5765MH est suffisamment robuste pour relever des poids considérable.



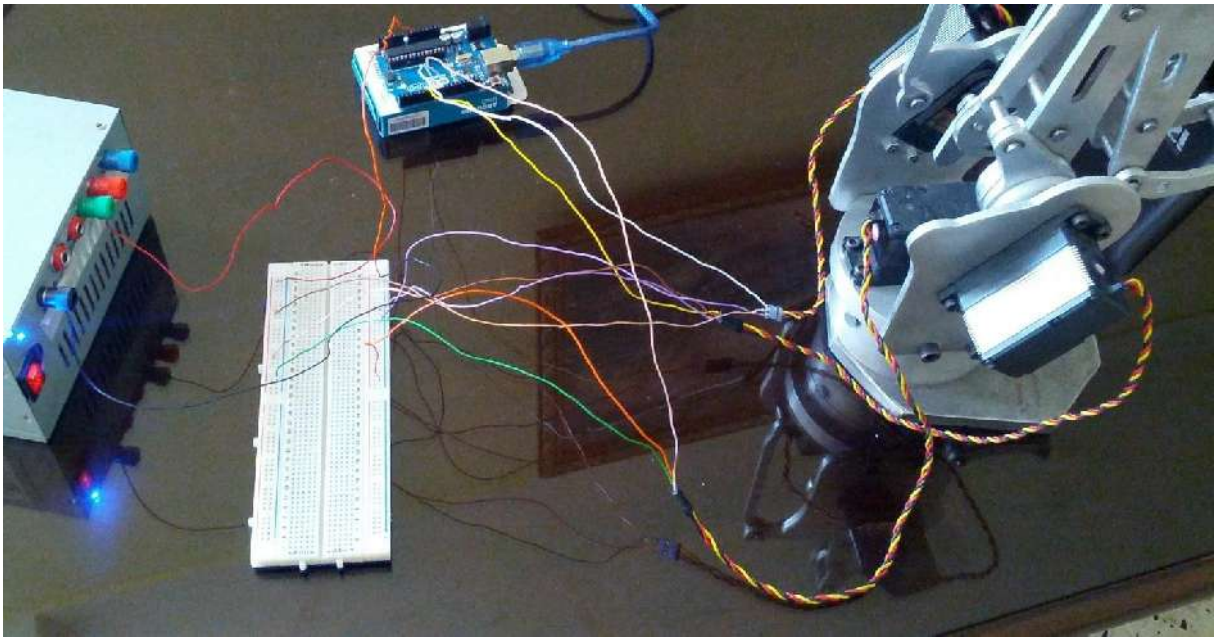
FigureVI.12 : servomoteur utilisé

IV.5.3. L'alimentation Stabilisé :

Pour le bon fonctionnement du notre robot manipulateur, nous avons besoin d'utilisé une tension stabilisée de 0 à 5V avec une puissance autour de 2Watts, pour ces raisons, on utilise une alimentation stabilisé d'un PC ordinaire, car elle produit un courant très fort et elle est sécurisée (des limitations), elle est convenable pour les servomoteurs, vue que chacun consomme autour de 0.6 Ampère.



FigureVI.13 : synoptique de l'alimentation stabilisé.



FigureVI.14 : Raccordement entre les différentes parties du robot manipulateur.

IV.6. Etape 4 :

Nous abordons maintenant la partie la plus importante de notre travail, c'est l'adaptation entre les différentes parties hardwares et softwares.

Suivant une étude bien établie vue précédemment, nous avons aventureux de :

- Commander notre robot avec le langage Arduino, ce choix pour le but de programmer séquentiellement tous les gestes et les mouvements de notre robot. Une étape d'exécution sera l'étape finale pour réaliser tous les commandes séquentielles du départ jusqu'au la fin de l'action.

- Commander notre robot avec le langage Matlab, ce choix de langage est nécessaire a cause d'une partie importante de traitement de signal et de l'image, afin de manipuler notre robot d'une manière aléatoire (suivant le

geste de l'utilisateur), dans ce choix, notre robot manipulateur fonctionne au temps réel.

Pour chaque choix il est nécessaire d'optimiser la trajectoire pour passer d'un point source (initial) vers un point but (final).

IV.6.1 La commande du bras manipulateur :

IV.6.1.1 La commande par le langage Arduino :

La commande du bras manipulateur est reçue à partir du PC vers les servomoteurs à travers le microcontrôleur de la carte Arduino, l'angle de chaque actionneur est calculé par langage Arduino et le transfert de données est réalisé à l'aide de la liaison série par un adaptateur USB. Finalement les actionneurs transforment l'information électrique vers une rotation mécanique de l'arbre du moteur en conséquence l'ensemble de la structure mécanique du bras manipulateur sera mettre en mouvement.

IV.6.1.1.1 L'espace de travail :

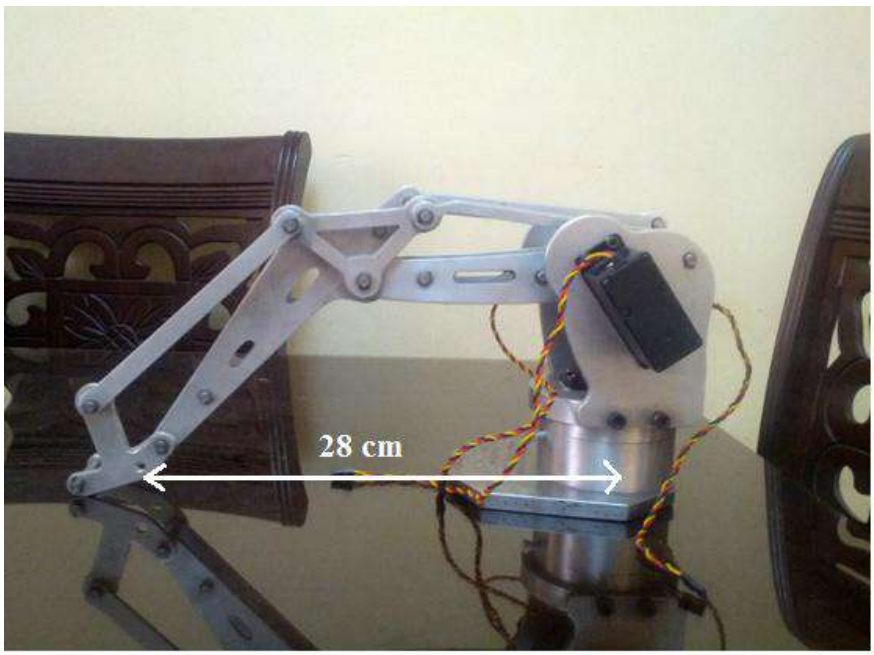
Parmi les caractéristiques importantes du robot son espace de travail, qui est l'espace physique engendré par un point de l'organe terminal lorsque le robot est en mouvement. Dans cet espace, on a rencontré plusieurs contraintes qui limitent le mouvement du robot parmi lesquelles :

-Les limitations sur les coordonnées articulaires : les articulations actives ont un angle minimum et un angle maximum (limitation de la rotation des deux bras), les deux bras sont limités par 90^0 ,à cause du mécanisme réaliser

-Une limitation à la base qui ne peut pas dépasser 70^0 car le pignon mobile du système d'engrenage conçu, fait une moitié de diamètre du pignon fixe.



FigureVI.15: photo réelle de la fermeture maximale.



FigureVI.16 : Photo réelle de l'ouverture maximale.

IV.6.1.2 implémentation de La commande visuelle :

En premier lieu on va présenter le logiciel Matlab.

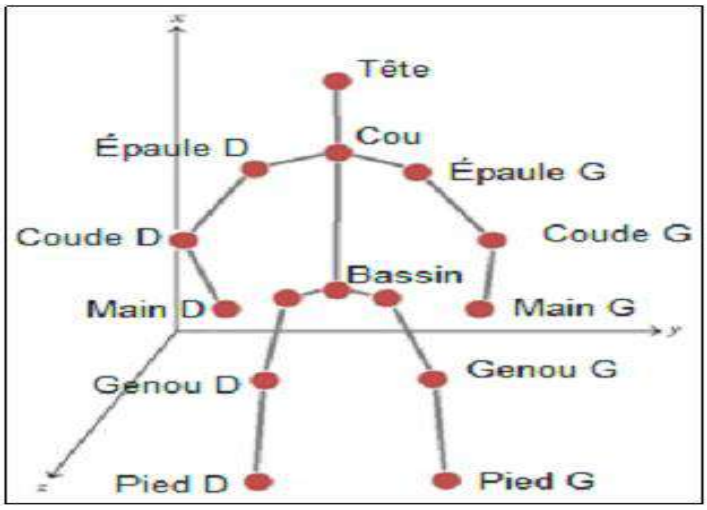
Présentation du logiciel Matlab :

MATLAB est- un programme interactif de The MathWorks pour « numeric computation » de haute performance et la visualisation. MATLAB intègre l'analyse numérique, calcul matrice, traitement du signal et des graphismes dans un environnement facile à utiliser. MATLAB est construit sur le logiciel de « sophisticated matrix » pour analyser des équations linéaires.

Les outils fournis peuvent être utilisés pour « applied mathematics », physique, chimie, ingénierie, finance et autres questions relatives à des calculs de « complex numerical ».

Pour réaliser le suivi du mouvement humain en 3D par vision artificielle, nous somme obliger d'utiliser un modèle 3D visuel articulé qui représente le corps humain. Nous avons commencé à développer l'application de suivi 3D exploitant les données de la caméra ASSUS Xtion Pro live. Ainsi les différentes librairies d'OPEN NI qui intègrent un système de reconnaissance de suivi pour des objets bien définis. Nous avons intégré ces techniques dans un environnement de développement Matlab version 2014b.

Le modèle humain comprends 15 joints représentant les points clés du corps dont nous voulons suivre le mouvement du bras gauche, avec :



FigureVI.17 : Le modèle de Vitruve à 15 points d'articulation.

Lors de la connexion de la Kinect à l'ordinateur, un flux de données est créé afin de récupérer toutes les informations détectées. Ce flux de données correspond à une image de la scène : 30 images sont envoyées par seconde entre la Kinect (émettrice) et l'ordinateur (récepteur).

Une image est composée de deux flux :

- Un flux « couleur » permettant de différencier les couleurs de la scène et d'afficher le flux vidéo en couleur.

- Un flux « profondeur » principalement lié au capteur infrarouge permettant de gérer l'axe Z de la scène 3D et capturer en temps réel l'image ainsi connaitre la distance à laquelle se trouve un squelette ce qui rend le suivi des utilisateurs plus précis.

Pour gérer la scène 3D on a besoin d'une interface graphique, de notre part l'interface graphique que nous utilisons elle était fournie par notre promoteur et elle consiste d'ajouter la troisième dimension qui manque aux interfaces graphiques classiques des logiciels de modélisation 3D, interfaces pour lesquelles on n'utilise que des périphériques d'entrée 2D (souris/clavier).

Concernant le programme pour atteindre une certaine position, chaque moteur doit obtenir certain angle, nous avons réalisé un programme Matlab qui calcule l'angle pour chaque moteur selon les axes « x »et »y » donc en mode planaire car on n'a pas la possibilité de faire la modélisation mathématique qui corresponde au modèle 3D.

Les coordonnées doivent être dans les limites du volume de travail, le programme va prend le maximum de la valeur qui est valable dans le volume de travail. C'est des mesures de sécurité qui sont prise par le bon fonctionnement du robot.

Afin d'obtenir les angles pour le mouvement désiré capter à l'aide de la caméra Kinect qui détecte les points sélectionnés sur le bras gauche du squelette, le programme transfère les données obtenue vers le système de commande puis vers les moteurs qui par leur rôle donnent un mouvement de rotation du bras manipulateur similaire au mouvement réel.

Dans un premier temps, nous avons procédé à la modélisation d'un être humain en temps réel qui est nécessaire pour la suite du travail. Donc le modèle (Squelette virtuel) est superposé sur un sujet volontaire. Pendant cette phase, on s'assure que l'utilisateur est bien positionné devant la caméra et de cette façon le premier alignement Modèle/volontaire est réalisé.



FigureVI.18: Modélisation d'un être humain avec un modèle de Vitruve à 15 points d'articulations.

On peut représenter cette partie par le synoptique suivant :

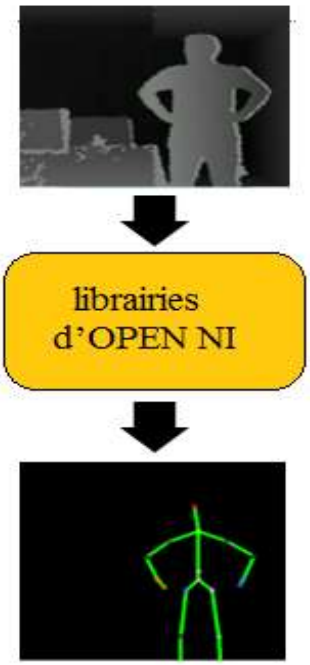


Figure VI.19: schéma synoptique de détection.

Une fois que le modèle est bien superposée sur le volontaire, on peut gérée la commande visuelle à partir des positions : épaule, coude et main gauche

Dans cette phase de notre réalisation, on programme notre commande a 3 niveaux d'excusions (position de repos, position 1 et position 2).

- En position de repos, le geste qui correspond cette action est représenté par la figure (VI.20), tell que l'épaule gauche sera plus haut que le coude et la main gauche par rapport à l'axe Y.

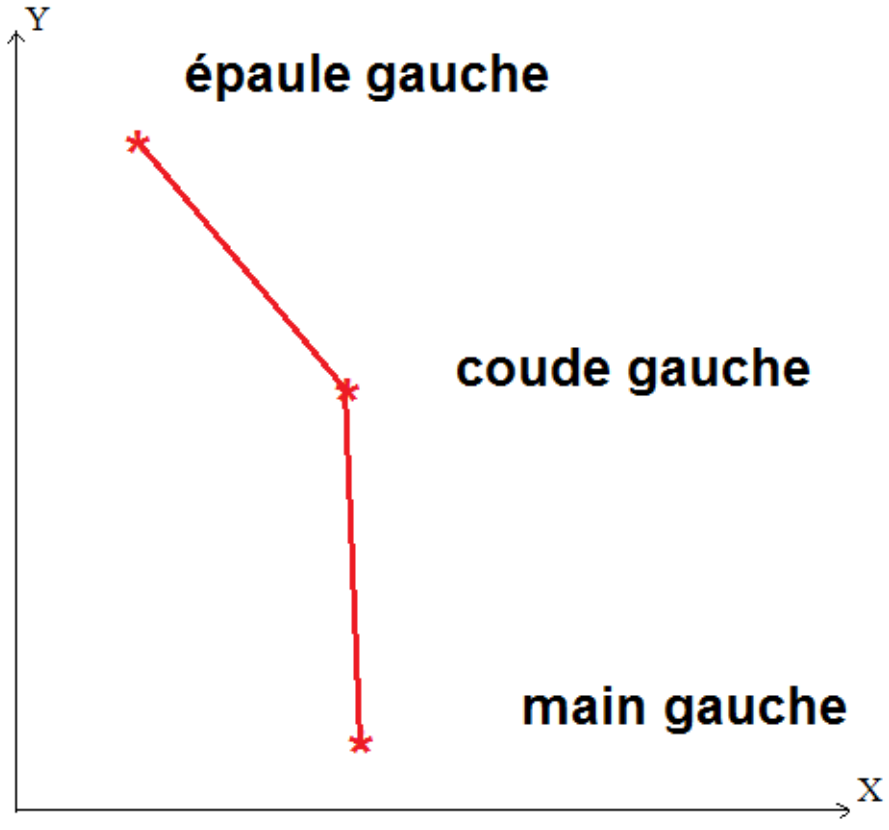
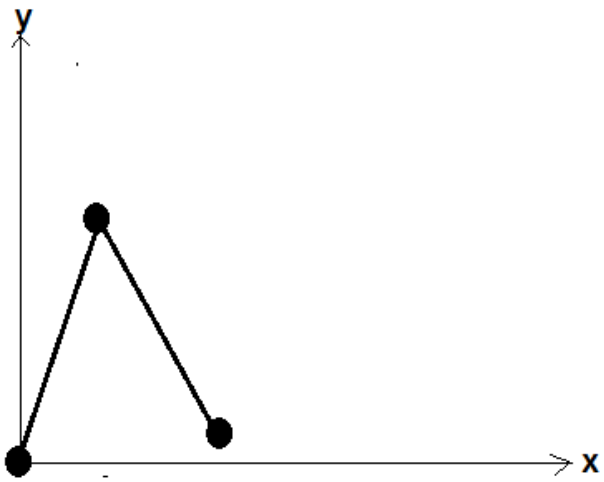
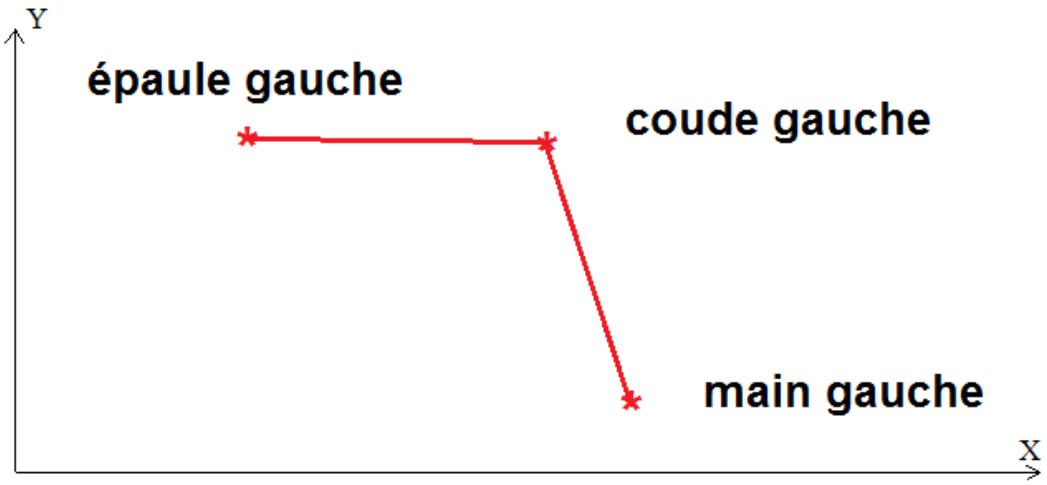


Figure VI.20: schéma représente le geste du bras gauche en position de repos.

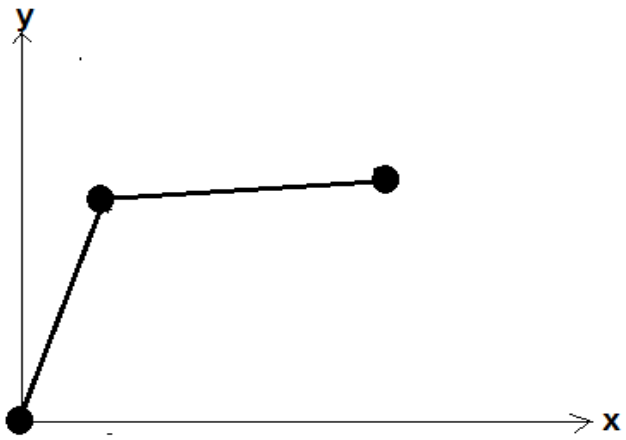


FigureVI.21 : schéma représente le suivi du bras en position de repos.

- En position 1, le geste qui correspond cette action nécessite que le coude est de même niveau ou d'un niveau supérieure à l'épaule gauche, mais le niveau de la main reste toujours inferieure au coude gauche par rapport à l'axe Y, figure (VI.22).

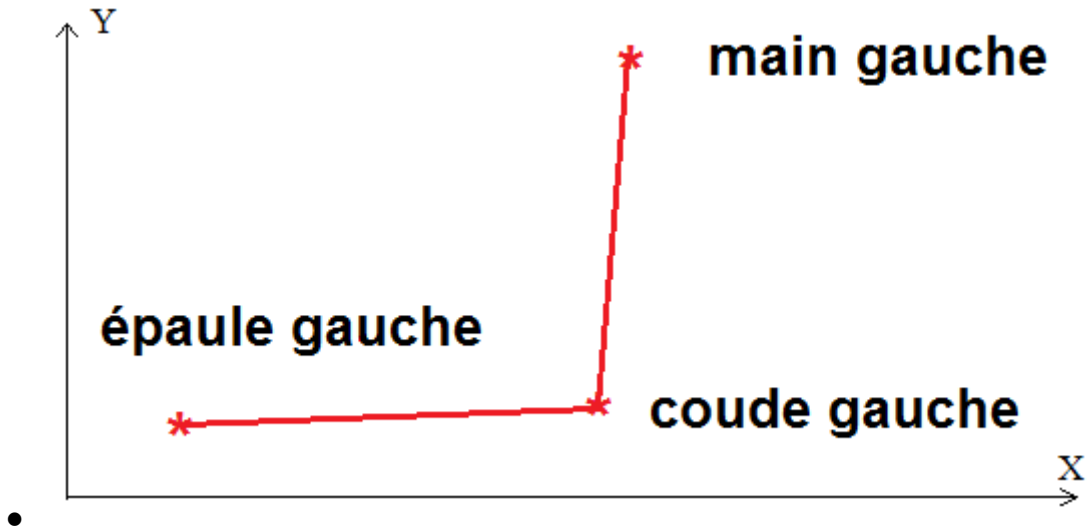


FigureVI.22 : schéma représente le bras gauche en 1 ème position. .



FigureVI.23 : schéma représente suivi du bras en 1ème position.

- En position 2, le geste qui correspond cette action nécessite que l'épaule, le coude aient les mêmes niveaux mais la main prend un niveau supérieure à ceux-ci par rapport à l'axe Y, figure (VI 24).



FigureVI.24 : schéma représente le bras gauche en 2 ème position**.**

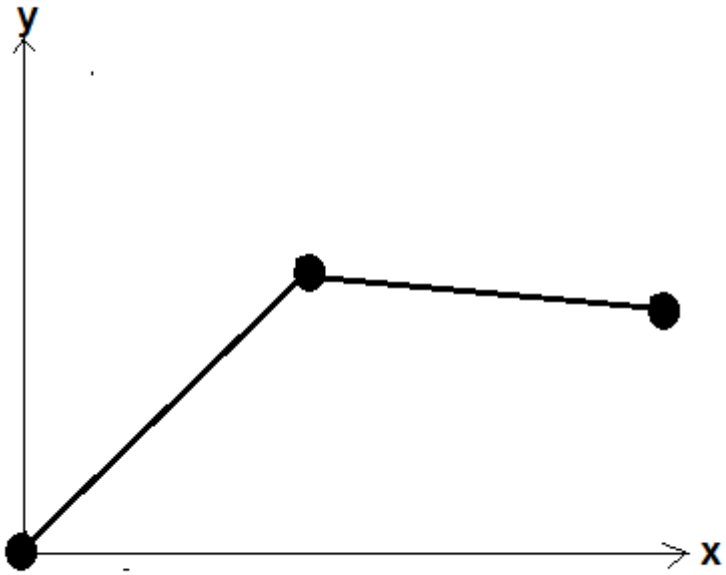


Figure VI.25 : schéma représente le suivi du bras en 2ème position.

IV.6.2 Résultats et interprétations de suivi de mouvement :

Le volontaire est en postions de repos.



Figure VI.26: photo prend les points d'articulations sur le volontaire en position de repos



FigureVI.27 : le suivi du bras à la position de repos.

On remarque que le bras fait l'action de retour en position de repos similaire au geste de bras du volontaire.

- Le volontaire en 1ème position.



FigureVI.28 : photo prend les points d'articulations sur le volontaire en 1ème position.



FigureVI.29 : le suivi du bras à la 1ème position.

Lorsque le volontaire change le geste vers la 1ème position le bras suit cette action en temps réel.

- Le volontaire en 2ème position.



Figure VI.30: photo prend les points d'articulations sur le volontaire en 2ème position**.**

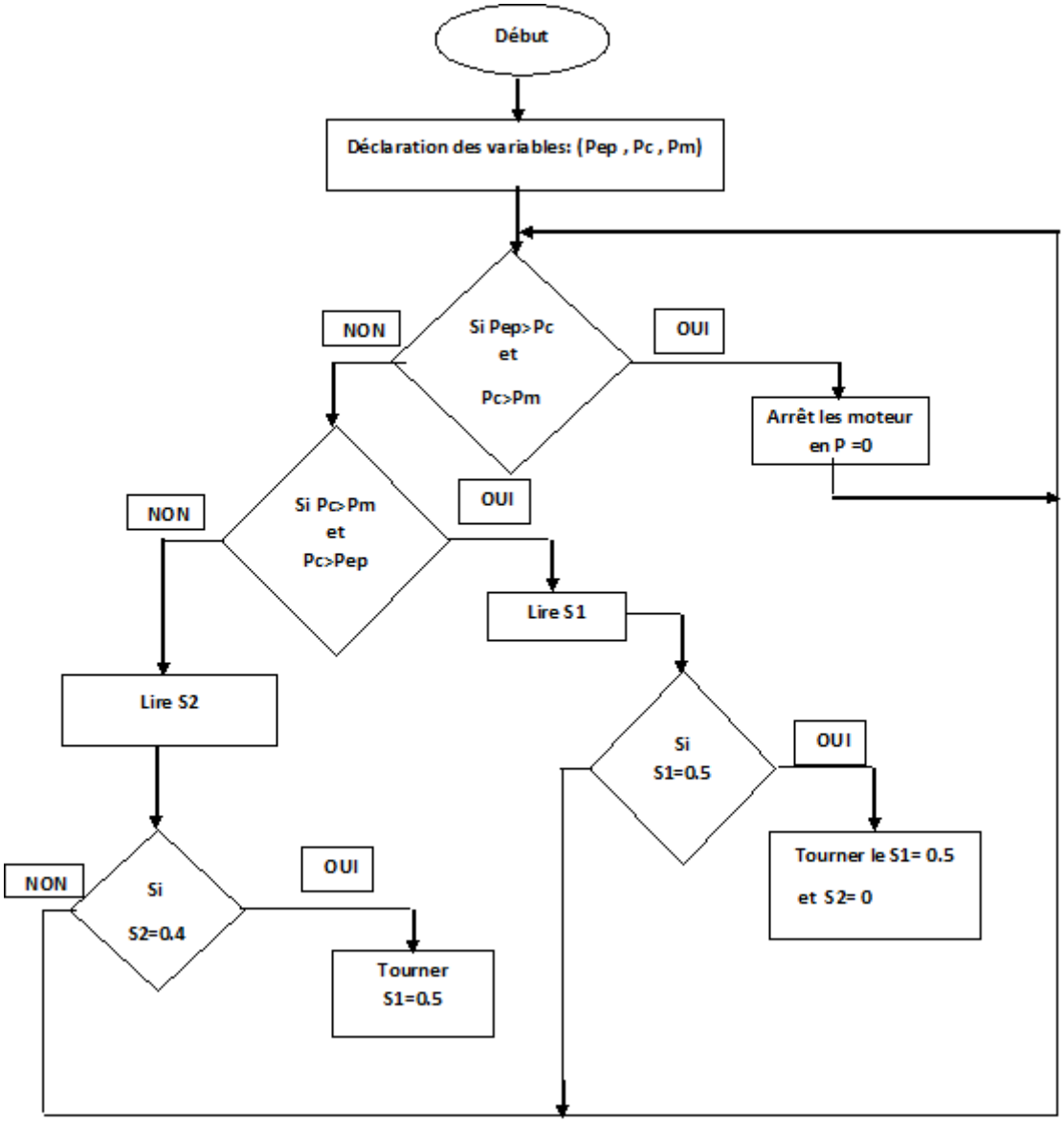


FigureVI.31 : le suivi du bras pour la 2ème position**.**

Quand le volontaire met son bras à la 2ème position on voit que le bras fait cette action avec sollicitude.

Nous constatons que notre bras suit bien les mouvements établis par le volontaire, et ça signifier que la commande par langage Matlab est réalisé comme nous avons prévu en mode planaire.

IV.6.2 L'organigramme de la commande visuelle:



FigureVI.32 : L'organigramme de la commande visuelle

IV.7 Conclusion :

On a exposé dans ce chapitre toutes les étapes de notre réalisation. On a commencé par la partie mécanique puis la partie électronique vers la commande pour montre que notre système est un système mécatronique répond à des besoins industriels.

D’après les tests réalisés, le robot montre de bonnes performances et peut être utilisé en plusieurs tâches.

Conclusion générale

Nous avons abordé dans cette thèse plusieurs aspects de la commande des robots manipulateurs, notamment l'utilisation du retour visuel pour faire évoluer les robots dans les environnements non structurés propres à l'homme.

Une première partie de ce travail est dédiée à la réalisation du bras manipulateur cette partie est faite avec minutie, en effet nous somme arrivé à faire l'étude sur les différents modèles utilisés pour décrire le mouvement des articulations puis la conception et la réalisation en utilisant des servomoteurs en vue d'avoir une bonne précision pour localiser les points cibles ,avec toutes les difficultés rencontrées nous avons pu faire fonctionner notre bras manipulateur .

Une deuxième partie a été consacrée à la partie software dans le but de commander le bras manipulateur en utilisant deux logiciels différents : le logiciel Arduino et le logiciel Matlab. Ce dernier est utilisé de sorte à ce qu'on introduit des gestes humains à travers une caméra Kinect, en l'occurrence Matlab est servi alors comme interface entre notre bras manipulateur et la camera ; chose qui nous a beaucoup aidé à la programmation en exploitant ces deux logiciels.

La programmation via le langage Arduino est une tâche très importante, elle nous a conduit à mieux maitriser le coté software de manière à ce qu'on puisse traduire les mouvements du bras manipulateur en instructions exécutables par le logiciel.

La programmation Sous Matlab est également importante, par conséquent nous avons beaucoup appris à programmer sous le logiciel Matlab sauf qu'on a eu certaine difficultés lors d'utilisation la camera Kinect pour commander visuellement notre Bras manipulateur en 3D, nous nous sommes contentés seulement par la commande visuelle en 2D (mode planaire).

En effet ce projet de réalisation nous a permis de traiter des problèmes d'ordre pratique (problèmes mécaniques, programmation) et de vérifier des connaissances théoriques acquises tout au long de notre formation. En d'autre partie ce projet nous a permet d'acquérir une bonne expérience pratique et théorique dans le monde de la robotique.

L'analyse des travaux présentés dans ce mémoire nous permet d'envisager quatre axes concernant les perspectives de recherche en futur :

- Espérer de rendre la commande visuelle fonctionnelle en 3D (introduire la 3ème DDL' la rotation de la base')
- Amélioration de la précision et la rapidité du robot.
- Ajout d'une pincette permettant de soulever des pièces cibles et rendre notre bras manipulateur très complet.
- Faire commander le bras manipulateur par des techniques de l'intelligence artificielle.

Bibliographie

[1] Site Paris TECH lien http://www.paristechreview.com publié/2013/07/18/histoire-robots/ Date de dernière consultation : 19/06/2016.

[2] Cherfia Abdelhakim, « analyse cinématique des robots parallèles, thèse de Doctorat, filière : génie mécanique », université de Mentouri Constantine 2007

[3]Site tableau des liaisons mécaniques lien http://www.amboilati.org/chantier/wp-content/uploads/ Date de dernière consultation :19/06/2016.

[4]H.DEMOUVEAU ' Poly-robotique' centre de recherche MINE DOUAI'Lille'octobre 2007

[5] Site Cours_robot_boimond lien /www.ensta bretagne.fr/jaulin/mastersds cours robot boimon.pdf Date de dernière consultation : 19/06/2016.

[6]Livre : traite de robotique 1 les architectures publié en 2010

[7]KHALIL- DOMBRE 'Bases de la modélisation et des robots de la-manipulateurs de type série commande 'université numérique08 mai 2012

[8]Robotique_Ind_SMART_10-11 'Cours et TP de Robotique de Manipulation, Master SMART, 09-10, R. Merzouki' 2010-2011

[9] Khelil cherfi « Conception et realisation d'un robot 3D de liberté thèse d'ingéniera filière :électrotechnique » Université de Médéa 1995

[10] Le grand livre d'Arduino De Erik Bartmann 2eme édition 2015

[11] SE2I ROBOTIQUE-Principe d'un servomoteur pdf 2009

[12]Site le-servomoteur lien http://ericaeromodelisme974.unblog.fr/2011/06/04/ Date de dernière consultation : 19/06/2016.

[13] Commander_des_servomoteurs_classiques lien http://www.planete-sciences.org/iledefrance/images/psidf/pdf/Fiche_F4_Commander_des_servomoteurs_classiques_et_a_rotation_continue.pdf Date de dernière consultation : 19/06/2016.