

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

الجمهورية الجزائرية الديمقراطية الشعبية

MINISTRY OF HIGHER EDUCATION
AND SCIENTIFIC RESEARCH

HIGHER SCHOOL IN APPLIED SCIENCES
--T L E M C E N--



المدرسة العليا في العلوم التطبيقية
École Supérieure en
Sciences Appliquées

وزارة التعليم العالي والبحث العلمي

المدرسة العليا في العلوم التطبيقية
-تلمسان-

Mémoire de fin d'étude

Pour l'obtention du diplôme de Master

Filière : Automatique
Spécialité : Automatique

Présenté par : MOUAZIZ Ahmed Adlane
&
OULD BABAALI Racha

Thème

**Commande d'un pont roulant avec bras
manipulateur de 6 degrés de libertés par
radio-fréquence**

Soutenu publiquement, le 08 / 09 / 2020, devant le jury composé de :

Mr MEGNAFI Hicham	M.C.B	ESSA. Tlemcen	Président
Mr FEROUANI Abdel Karim	M.C.A	ESSA. Tlemcen	Directeur de mémoire
Mr SIDHOUM Riad	Ingénieur	METGAV INDUSTRY	Co- Directeur de mémoire
Mr BENADDA Belkacem	Professeur	Univ. Tlemcen	Examineur
Mr ABDELLAOUI Ghouti	M.C.B	ESSA. Tlemcen	Examineur

Année universitaire : 2019 / 2020

Remerciements

*Au terme de cette belle expérience qu'est le mémoire de master, nous tenons tout d'abord à adresser nos plus vifs remerciements au **Dr FEROUANI Abdel Karim** Encadreur, Maitre de conférence classe A à ESSA-Tlemcen, qui a dirigé ce travail avec beaucoup d'abnégation et de professionnalisme ; nous sommes particulièrement reconnaissants à la confiance qu'il nous a toujours témoignée. Nous avons également apprécié sa patience, sa disponibilité et ses conseils avisés. Nous tenons à lui exprimer notre profonde gratitude de nous avoir fait bénéficier de son expérience scientifique avec des qualités humaines indéniables.*

*Nous tenons à exprimer aussi nos respects et notre profonde gratitude à Monsieur **SIDHOUM Riad** Co-encadreur, Ingénieur à la Société METGAV INDUSTRY, pour son aide précieuse dans l'élaboration de notre mémoire et surtout pour son soutien technique qui est son domaine de créativité.*

*Nous adressons également nos plus vifs remerciements à Monsieur **MEGNAFI Hicham**, Maitre de conférences classe B à l'Ecole supérieure en sciences appliquées de Tlemcen, pour nous avoir fait l'honneur de présider le jury et pour l'intérêt qu'il a porté à ce travail. Il mérite de le remercier une seconde fois pour toutes les sciences qu'il nous a enseignées et son assistance lors de notre cursus.*

*A Monsieur **Pr BENADDA Belkacem**, Professeur à l'Université de Tlemcen qui a bien voulu faire partie du jury, et sa présence qui nous a orientée avec discernement, qu'il veuille bien accepter ici l'expression de notre sincère reconnaissance.*

*Nous remercions également Monsieur **ABDELLAOUI Ghouti**, Maitre de conférence classe B à l'Ecole supérieure en sciences appliquées de Tlemcen de nous avoir fait l'honneur et le plaisir d'examiner notre travail.*

Dédicaces

Tous les mots ne sauraient exprimer la gratitude, l'amour, le respect, la reconnaissance, je viens tout simplement dédier ce modeste mémoire :

A la mémoire de ma chère maman qui nous a quittée sitôt, celle qui m'a arrosé de tendresse et d'espoir, à la source d'amour qui a fait de moi ce que je suis aujourd'hui et qui m'a béni par ses prières. Qu'Allah la couvre de sa miséricorde.

A la personne qui m'a supporté et m'a dirigé vers la réussite : mon père. Ce paternel qui se souciait à chaque instant, et ne me suivait pas à pas pour réaliser mon rêve sans se lasser ; c'est l'ami, le conseiller et le confident.

A ma sœur Zeyneb, l'unique que je chéris tant pour son amour et sa tendresse qu'elle m'accorde et surtout pour son soutien dans le bon et le pire.

A mes frères, Aboubakr pour l'aide morale, ses conseils et sa présence. Abderrahman, pour son âme si vive et souriante à qui je souhaite une vie heureuse ; il voulait à tout prix que je réussisse.

A mon cousin MOUAZIZ Mohammed El Amine pour son aide précieuse lors de mes stages et à sa petite famille très accueillante.

Ahmed Adlane

Dédicaces

Je dédie ce modeste travail accompagné d'un profond amour :

A celle qui m'a arrosé de tendresse et d'espoir, à la source d'amour qui m'a fait ce que je suis aujourd'hui, à mon support dans ma vie qui ma bénie par ces prières ... Ma mère.

A la personne qui m'a supporté et ma dirigé vers la gloire Mon père.

A mes chères sœurs Hanane et Merieme et mon petit frère Marouane qui ont partagé avec moi tous les moments d'émotion lors de la réalisation de ce travail.

A mon ami Abdelbaki qui n'a pas cessé de me conseiller, encourager et soutenir tout au long cette période.

A ma meilleure amie Kenza pour son soutien moral, sa présence et sa compréhension tout au long mes années d'étude à l'ESSAT.

A tous ceux que j'aime et ceux qui m'aiment.

RACHA

Résumé

L'objectif de notre travail aboutit à la conception d'un système de radiocommande qui permet de contrôler le fonctionnement des ponts roulants avec un bras manipulateur à distance. Cette procédure fournit des connaissances et des outils qui permet d'autonomiser les opérations de levage, de déplacement des charges et du transport dans un intervalle de temps bien réduit. L'ajout de cette nouvelle technologie au pont roulant permet de manipuler d'une manière précise et sécurisée les charges. Plusieurs résultats de simulation sont présentés pour illustrer les performances des approches proposées.

Mots Clés : Ponts roulants, bras manipulateur, radiocommande, radiofréquence.

Abstract

The aim of our project is to design and implement a radio-command system which uses a robotic arm to manipulate and control the movements of an overhead crane. This procedure helps to automate the process of lifting, moving and transporting objects within the minimal possible amount of time. Adding this technology to such a system will improve movements precision and thus raises the security level while transporting charges. Simulation results are presented to illustrate the system performances.

Key words: overhead crane, robotic arm, radio-command, radio-frequency.

ملخص

يهدف هذا المشروع إلى تصميم وإنجاز نظام قيادة لاسلكي يستخدم ذراعًا آليًا للتحكم في حركات الرافعة المتنقلة. هذا الإجراء سيمكننا من حمل و نقل الأشياء و البضائع آليًا مما سيرفع نسبة الدقة والسلامة في الاداء إضافة الي اختصار الوقت. نتائج المحاكات معروضة لتوضيح اداء النظام الجديد.

الكلمات المفتاحية : الرافعات العلوية، رافعات قنطرية ، تحكم لاسلكي ، تردد لاسلكي. ذراع الي. ذراع مفصلي

Sommaire

<u>INTRODUCTION GÉNÉRALE</u>	1
<u>CHAPITRE I : GÉNÉRALITÉS SUR LES PONTS ROULANTS INDUSTRIELS</u>	
I.1 INTRODUCTION	4
I.2 MÉCANISME GÉNÉRALE	5
I.2.1 LES PRINCIPAUX MOUVEMENTS	6
I.3 LES MODÈLES INDUSTRIELS DES PONTS ROULANTS	7
I.3.1 PONTS ROULANTS POSÉS	7
I.3.2 PONTS ROULANTS SUSPENDUS	8
I.3.3 PORTIQUES	9
I.3.4 SEMI-PORTIQUES	9
I.4 COMMANDE DES PONTS ROULANTS	10
I.4.1 COMMANDE CABINE	10
I.4.2 COMMANDE AU SOL	11
I.4.2.a Boîte a boutons pendante	11
I.4.3 RADIOCOMMANDES	11
I.5 LE FONCTIONNEMENT DES PONTS ROULANTS	11
I.5.1 FONCTIONNEMENT MANUEL	12
I.5.2 FONCTIONNEMENT SEMI-AUTOMATIQUE	13
I.5.3 FONCTIONNEMENT AUTOMATISÉ	13
I.6 PONTS ROULANTS AVEC BRAS MANIPULATEUR	14
CONCLUSION	15
<u>CHAPITRE II : APERÇU GÉNÉRAL SUR LES BRAS ROBOTIQUES</u>	
II.1 HISTORIQUE	18
II.2 LES SEGMENTS ET LES MOUVEMENTS DU BRAS ROBOTIQUE	19
II.3 EFFECTEUR TERMINAL	19
II.4 LES TYPES DES BRAS ROBOTIQUES	20
II.4.1 ROBOT LINÉAIRE	20
II.4.2 ROBOT CYLINDRIQUE	22
II.4.3 ROBOT PARALLÈLE	23
II.4.4 ROBOT SPHÉRIQUE	25
II.4.5 SCARA	26
II.4.6 ROBOT ARTICULÉ	28
II.5 COMPARAISON ENTRE LES BRAS ROBOTIQUES FONDAMENTAUX	30
CONCLUSION	31

CHAPITRE III : MODÉLISATION DU BRAS MANIPULATEUR

III.1 MODÉLISATION DU SYSTÈME DE LEVAGE BRAS MANIPULATEUR	34
III.1.1 REPRÉSENTATION DU BRAS MANIPULATEUR.....	34
III.1.2 DÉTERMINATION DU REPÈRE R_j LIÉ AU COPS C_j	35
III.1.3 MODÈLE GÉOMÉTRIQUE DU BRAS MANIPULATEUR EN CHAÎNE SIMPLE.....	38
III.1.3.a Modele geometrique direct du bras manipulateur (M.G.D).....	38
III.1.3.b Modele geometrique inverse du bras manipulateur (M.G.I).....	40
III.2 SIMULATION DU BRAS MANIPULATEUR SOUS MATLAB.....	42
III.2.1 PRÉSENTATION DU TRAVAIL.....	44
III.2.2 TRAJECTOIRE DE L'ORGANE TERMINAL	46
III.2.2.a Trajectoire circulaire	46
III.2.2.b Trajectoire circulaire avec deplacement	47
CONCLUSION.....	49

CHAPITRE IV : LE BRAS MANIPULATEUR EN SIMULATION

PARTIE IV.I : HARDWARE

IV.I.1 L'ENVIREMENT DE SIMULATION HARDWARE	53
IV.I.1.1 PRÉSENTATION GÉNÉRALE	53
IV.I.2 DESCRIPTION DE LA CARTE ÉLECTRONIQUE DE COMMANDE	54
IV.I.3 SCHÉMA DU CIRCUIT ÉLECTRIQUE	54
IV.I.3.1 CIRCUIT ÉLECTRIQUE DE LA COMMANDE ET DE PUISSANCE	55
IV.I.3.2 CIRCUIT ÉLECTRIQUE DE LA BOITE DE COMMANDE PAR RF.....	56
IV.I.4 LES COMPOSANTS ÉLECTRONIQUES	56
IV.I.4.1 MICROCONTRÔLEUR.....	56
IV.I.4.2 LE CIRCUIT DE PUISSANCE A4988.....	59
IV.I.4.2.1 CARACTERISTIQUES ET DETAILS TECHNIQUES DU POLULU A4988	59
IV.I.4.2.2 BROCHAGE ET DESCRIPTION DES PINS DE DRIVER A4988	60
IV.I.4.2.3 AJUSTEMENT DE DRIVER A4988.....	62
IV.I.4.2.3.a Ajustement des entrees du controle	62
IV.I.4.2.3.b Configuration de mode de resolution	62
IV.I.4.3 LES MOTEURS PAS À PAS	63
IV.I.4.4 LES SERVOS MOTEURS	64
IV.I.4.4.1 MECANISME DES SERVOS MOTEURS	65
IV.I.4.4.2 PRINCIPE DE FONCTIONNEMENT DES SERVOMOTEURS	65
IV.I.4.4.3 CONTROLE DES SERVOMOTEURS.....	66
IV.I.4.4.4 MG996R SERVO MOTEUR.....	67
IV.I.4.4.5 MG90 MICRO SERVO MOTEUR.....	68
IV.I.4.5 LES CAPTEURS	68

IV.I.4.5.1 CAPTEUR DE DISTANCE.....	68
IV.I.4.5.1.a Principe de fonctionnement du capteur.....	69
IV.I.4.5.2 CAPTEUR DE FIN DE COURSE	70
IV.I.4.6 ÉMETTEUR / RÉCEPTEUR RADIO FRÉQUENCE	71
IV.I.4.6.1 LE TRAITEMENT DU SIGNAL	71
IV.I.4.6.1.a La modulation	71
IV.I.4.6.2 LE TRAITEMENT DES INFORMATIONS NUMERIQUES	73
IV.I.4.6.2.a La numération	73
IV.I.4.6.2.b Le codage canal	73
IV.I.4.6.3 LA CHAÎNE DE TRANSMISSION	74
IV.I.4.6.4 ÉMETTEUR RF 433MHZ.....	74
IV.I.4.6.5 RECEPTEUR RF 433MHZ	75
IV.I.4.6.6 ASK – AMPLITUDE SHIFT KEYING	75
IV.I.4.6.7 SPECIFICATIONS DU MODULE 433 MHZ.....	76
IV.I.4.6.8 CONFIGURATION DES BROCHES.....	77
IV.I.4.7 AFFICHEUR LCD :.....	77
IV.I.4.7.1 PRESENTATION DE L’AFFICHEUR.....	78
IV.I.4.7.2 DESCRIPTION DES PINS ET BROCHAGE	79
IV.I.4.8 KEYPAD- SMALLCLAC (4×4).....	79
IV.I.4.8.1 CONTROLER LE PONT ROULANT	80
IV.I.4.8.2 CONTROLER LE BRAS MANIPULATEUR.....	81
IV.I.4.9 LES BOUTONS POUSSOIRS.....	83
CONCLUSION.....	84

PARTIE IV.II : SOFTWARE

IV.II.1 LE LOGICIEL SOLIDWORKS	86
IV.II.1.1 MODÉLISATION DU PONT ROULANT AVEC SOLIDWORKS	87
IV.II.1.2 MODÉLISATION DE LA BOÎTE DE COMMANDE RF	88
IV.II.2 L’ENVIRONNEMENT DE DÉVELOPPEMENT ARDUINO IDE	89
IV.II.2.1 STRUCTURE D’UN PROJET ARDUINO	90
IV.II.3 LES BIBLIOTHÈQUES UTILISÉES.....	91
IV.II.3.1 BIBLIOTHÈQUE RADIOHEAD	91
IV.II.3.2 BIBLIOTHÈQUE SPI	92
IV.II.3.3 EXEMPLE D’UTILISATION	93
IV.II.3.4 BIBLIOTHÈQUE SOFTWARE SERVO.....	96
IV.II.3.5 BIBLIOTHÈQUE ACCELSTEPPER	96
IV.II.3.6 BIBLIOTHÈQUE LIQUIDCRYSTAL	97
IV.II.4 LES OUTILS DE LA MODÉLISATION	97

IV.II.4.1 LANGAGE DE MODÉLISATION	97
IV.II.4.2 UNIFIED MODELLING LANGUAGE (UML).....	97
IV.II.4.2.1 DIAGRAMMES STRUCTURELS	98
IV.II.4.2.2 DIAGRAMMES DE COMPORTEMENT	98
IV.II.4.2.3 DIAGRAMMES D'ACTIVITE.....	99
IV.II.4.2.3.a Notations du diagramme d'activité	100
IV.II.5 MODÉLISATION	102
IV.II.6 RÉSULTATS ET DISCUSSION.....	104
CONCLUSION.....	107
<u>CONCLUSION GÉNÉRALE</u>	<u>110</u>

Liste des figures

Fig.I.1 Quelques applications des ponts roulants, avec : (a) Pont roulant pour containers, (b) Pont roulant pour manutention de matériaux, (c) Pont roulant avec grappin et palan auxiliaire pour centrale d'incinération et (d) Pont roulant pour les scories proven.....	4
Fig.I.2 Pont roulant : structure générale d'un pont roulant.....	6
Fig.I.3 Principaux mouvements d'un pont roulant, avec : (A) La translation, (B) La direction et (C) Le levage.....	6
Fig.I.4 Pont roulant, avec : (a) Mono poutre et (b) Bipoutre.....	7
Fig.I.5 (a) Pont roulant bipoutre posé et (b) Pont roulant mono poutre posé.....	8
Fig.I.6 Pont roulant : (a) Mono poutre suspendue et (b) Bipoutre suspendu.....	8
Fig.I.7 Portique.....	9
Fig.I.8 Semi portique.....	10
Fig.I.9 Commande cabine.....	10
Fig.I.10 Classification des ponts roulants avec leurs degrés d'automatisation.....	12
Fig.I.11 Commande manuelle d'un pont roulant.....	12
Fig.I.12 Exemple d'un mode semi-automatique.....	13
Fig.I.13 Exemple d'un mode automatique.....	13
Fig.I.14 Pont roulant avec bras manipulateur et prototype.....	14
Fig.II.1 Robot portique Hercules x-y.....	21
Fig.II.2 Portée horizontale cartésienne et son symbole (à droite).....	21
Fig.II.3 Schéma du robot cylindrique et son symbole.....	23
Fig.II.4 Configuration d'un robot cylindrique.....	23
Fig.II.5 Le potentiel du premier robot parallèle breveté aux Etats-Unis.....	24
Fig.II.6 Robot parallèle (trépied avec trois axes).....	25
Fig.II.7 Robot parallèle (hexapode à six axes).....	25
Fig.II.8 Diagramme et symbole du robot sphérique.....	26
Fig.II.9 Configuration d'un robot sphérique.....	26
Fig.II.10 Schéma de SCARA.....	27
Fig.II.11 Schéma du robot Spine.....	27
Fig.II.12 Robot SCARA.....	28
Fig.II.13 Schéma et le symbole d'un robot articulé.....	28
Fig.II.14 Robot articulé typique.....	29
Fig.III.2 La cinématique du bras manipulateur (position et orientation).....	34
Fig.III.3 Structure du bras à poignet de type rotule.....	35
Fig.III.4 Repère des articulations $j-1$ et j	36
Fig.III.5 Paramètres de passage du repère des articulations $j-1$ et j	37
Fig.III.6 Génération de mouvement dans l'espace opérationnel.....	43
Fig.III.7 Bras manipulateur dans MATLAB.....	43
Fig.III.8 Les coordonnées des articulations.....	44
Fig.III.9 L'utilisation du capteur « Body Sensor ».....	44
Fig.III.10 Représentation du résultat du capteur « Body Sensor ».....	45
Fig.III.11 Représentation des corps du bras dans « Simulink ».....	45
Fig.III.12 Le programme de la trajectoire circulaire.....	46
Fig.III.13 La trajectoire dans l'espace d'étude (OXYZ).....	46
Fig.III.14 Simulation du mouvement du bras dans l'espace (OXYZ).....	47
Fig.III.15 La trajectoire circulaire avec déplacement.....	48

Fig.III.16 Simulation de la dynamique du bras dans l'espace (OXYZ).....	49
Fig.IV.I.1 L'enivrement de simulation ISIS Proteus.....	54
Fig.IV.I.2 Circuit électrique de commande et puissance.....	55
Fig.IV.I.3 Circuit de la commande par radiofréquence.....	56
Fig.IV.I.4 Carte Arduino méga 2560.....	58
Fig.IV.I.5 Pololu 4988.....	60
Fig.IV.I.6 Les drivers L297 ET L298.....	60
Fig.IV.I.7 Moteurs pas à pas avec leurs drivers.....	62
Fig.IV.I.8 Les moteurs des axes X et Y.....	64
Fig.IV.I.9 Configuration des moteurs pas à pas.....	64
Fig.IV.I.10 Commande d'un servo moteur par PWM.....	67
Fig.IV.I.11 Illustration du signal TRIGGER et ECHO.....	69
Fig.IV.I.12 Capteurs de distance HC-SR04.....	70
Fig.IV.I.13 Capteurs de fin de course.....	70
Fig.IV.I.14 Principe d'émission – réception.....	71
Fig.IV.I.15 Représentation électrique d'un signal audio.....	71
Fig.IV.I.16 Porteuse pure.....	72
Fig.IV.I.17 Modulation d'amplitude.....	72
Fig.IV.I.18 Modulation de fréquence.....	73
Fig.IV.I.19 Modulation de phase.....	73
Fig.IV.I.20 Emetteur Radio fréquence.....	75
Fig.IV.I.21 Récepteur radio fréquence 433MHz.....	75
Fig.IV.I.22 Modulation d'amplitude.....	76
Fig.IV.I.23 Emetteur (TX) et récepteur (RX) radio fréquence 433MHz.....	77
Fig.IV.I.24 Afficheur LCD LM016L.....	78
Fig.IV.I.25 Schéma interne du clavier.....	80
Fig.IV.I.26 Keypad- smallclac pour le pont (4×4).....	80
Fig.IV.I.27 Keypad- smallclac pour le bras (4×4).....	81
Fig.IV.I.28 Les boutons de commande.....	83
Fig.IV.I.29 Boutons de la boite de radiocommande.....	84
Fig.IV.II.1 Désigne du prototype du projet.....	87
Fig.IV.II.2 Boite de radiocommande fermée.....	88
Fig.IV.II.3 Boite de radiocommande ouverte.....	89
Fig.IV.II.4 Structure d'un code Arduino.....	90
Fig.IV.II.5 Paquet d'information Radio Head.....	91
Fig.IV.II.6 Exemple émetteur RF.....	93
Fig.IV.II.7 Exemple d'un récepteur rf.....	95
Fig.IV.II.8 La hiérarchie des diagrammes en UML2.0.....	99
Fig.IV.II.9 Diagramme d'activité du pont.....	103
Fig.IV.II.10 Affichage du message « Welcome ».....	104
Fig.IV.II.11 Message de demande d'entrer le code d'atelier.....	104
Fig.IV.II.12 Fonctionnement des moteurs X et Y.....	105
Fig.IV.II.13 Message d'arrivage.....	105
Fig.IV.II.14 Etat initial de chaque servo.....	106
Fig.IV.II.15 Exemple de manipulation.....	106

Liste des tableaux

Tableau II.1 Comparaison entre des bras fondamentaux.	30
Tableau III.1 Paramètres géométriques des trajectoires du bras manipulateur.	37
Tableau IV.I.1 Signification du circuit de commande et puissance.	55
Tableau IV.I.2 Signification du circuit de commande et puissance.	56
Tableau IV.I.3 La configuration de mode de résolution.	63
Tableau IV.I.4 Configurations des pins du module Rf.	77
Tableau IV.I.5 Description des broches du LCD.	79
Tableau IV.I.6 Signification des abréviations du keypad numéro 1.	81
Tableau IV.I.7 Signification des abréviations du keypad numéro 2.	82
Tableau IV.I.8 Descriptions des boutons de commande.	83
Tableau IV.II.1 Signification des éléments du prototype.	87
Tableau IV.II.2 Signification des abréviations de la boîte de radiocommande fermée.	88
Tableau IV.II.3 Signification des abréviations de la boîte de radiocommande ouverte.	89
Tableau IV.II.4 Combinaison polarité et phase.	93
Tableau IV.II.5 Notations du diagramme d'activité.	102

Introduction générale

Au sein de l'entreprise METGAV INDUSTRY, nous avons constaté des appareils de manutention qui ont une relation directe avec la production industrielle ce qui les rend indispensables et d'une grande importance.

La forte croissance du marché du transport des outils multiplie le volume des charges à manipuler, imposant des exigences de plus en plus strictes en termes de performance des grues et des ponts roulants. Ces performances sont exprimées sous la forme d'un compromis entre les critères de rapidité d'exécution, de précision et de sécurité des opérateurs. D'une manière générale, l'automatisation des systèmes de production a été une des réponses à ce compromis. Cette automatisation visait un double objectif : l'augmentation de l'efficacité du système technique (réduction des coûts, fiabilité, disponibilité, qualité) et l'amélioration de la sécurité directe des opérateurs.

L'objectif principal de notre travail se concentre sur la conception d'un bras manipulateur à six degrés de libertés, commandé à distance et adapté pour avoir un fonctionnement plus performant des ponts roulants d'une manière plus maîtrisée dans l'environnement de l'industrie. Cette étude fournit des connaissances et des outils qui permettront à long terme d'autonomiser les opérations de levage et de placement des charges et de les transporter dans un intervalle de temps bien optimisé.

L'idée de transformer le mécanisme des ponts manuels en système automatique nous a motivé à réaliser une nouvelle technologie adéquate qui est représentée par l'ajout d'un bras manipulateur. Ce projet reste un sujet d'actualité chez les industriels et chercheurs.

Le contrôle de ces systèmes nécessite d'abord l'étude de la géométrie du bras manipulateur à 6 D.D.L pour bien réaliser l'objectif fixé par l'utilisateur. Le but de l'étude consiste à commander le bras à distance par un opérateur en jouant sur les articulations qui composent ce dernier. Ce mécanisme a la possibilité d'enregistrer les mouvements des articulations lors de l'opération. La vitesse du pont ne provoque pas de risque sur le système ce qui nous a poussé à la maintenir constante.

Ce travail est consacré à :

- L'étude et la modélisation mathématique de la géométrie du bras manipulateur, étape indispensable pour le développement des techniques de commande.
- L'étude et la conception des trajectoires du bras par la méthode de Paul pour

obtenir la position souhaitée.

- Le prototype du pont roulant avec bras manipulateur et la commande radiofréquence qui ordonnera à distance des fonctions de déplacement du pont et du bras.
- L'utilisation de logiciel ISIS PROTEUS pour la simulation du projet.

Organisation du mémoire

Les travaux réalisés dans le cadre de ce mémoire s'articulent autour des quatre chapitres suivants :

Chapitre I

Dans ce chapitre nous avons énuméré les différents ponts existants dans le monde industriel en précisant les méthodes utilisées pour les commander, et un aperçu des ponts avec bras manipulateur.

Chapitre II

La discussion se base essentiellement sur la théorie des bras manipulateurs en mettant en évidence le fonctionnement de quelques bras de différents degrés de libertés.

Chapitre III

Dans ce chapitre nous avons évoqué le modèle mathématique de la géométrie du bras manipulateur. L'utilisation du logiciel MATLAB nous a permis la simulation du mouvement en calculant les articulations par la méthode de Paul pour différentes trajectoires.

Chapitre IV

Ce dernier chapitre est composé de deux parties. La première partie décrit et détail les caractéristiques du matériel utilisé (microcontrôleurs, capteurs, moteurs, servos moteurs ...) dans le prototype. La deuxième partie représente la programmation des microcontrôleurs ainsi qu'un diagramme d'activité du système de notre projet, et les résultats de la simulation.

Chapitre I
Généralités sur les ponts roulants
industriels

I.1 Introduction

Jusqu'à présent, les ponts roulants sont l'un des systèmes les plus importants pour la manutention indispensable dans plusieurs secteurs industriels (Fig.I.1). Bien que les ponts roulants automatiques soient relativement rares dans la pratique des usines. En raison du potentiel élevé de la rationalisation, plusieurs tentatives ont été faites dans le passé.

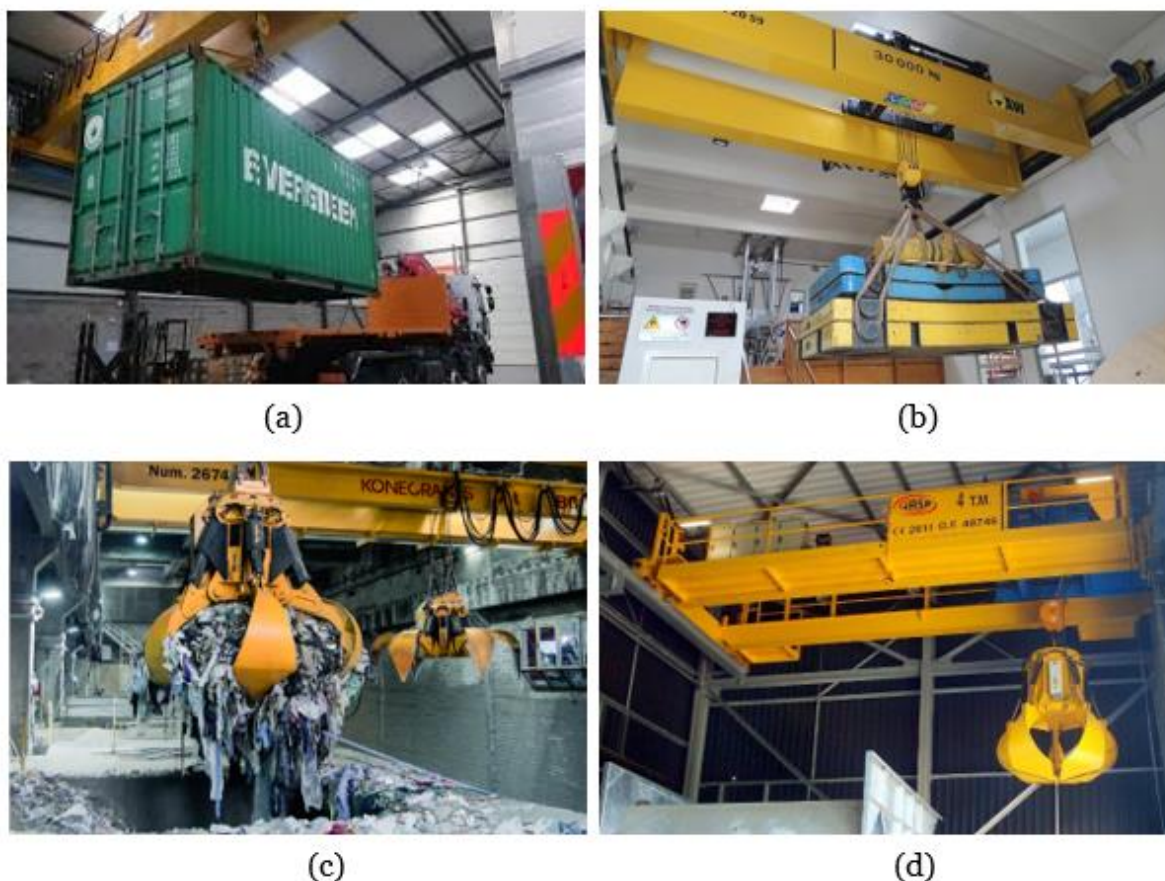


Fig.I.1 Quelques applications des ponts roulants, avec : (a) Pont roulant pour containers, (b) Pont roulant pour manutention de matériaux, (c) Pont roulant avec grappin et palan auxiliaire pour centrale d'incinération et (d) Pont roulant pour les scories proven.

Dans la plupart des domaines de fonctionnement des ponts roulants, il est très intéressant de réaliser avec précision le déplacement des charges d'un point à un autre, et plus particulièrement d'obtenir un balancement nul de la charge non seulement en fin du trajet, mais aussi au cours du déplacement.

L'automatisation des ponts roulants est l'un des éléments-clés pour relever la formation pratique des opérateurs qui est relativement complexe. Le phénomène du

balancement est l'un des grands défis face à l'augmentation des performances, tel que le temps de transfert, la précision et la sécurité des ponts roulants.

Les ponts roulants sont utilisés pour déplacer des charges extrêmement lourdes ou volumineuses dans l'espace aérien d'une installation, au lieu de passer par les allées ou sur le plancher afin d'éviter plusieurs accidents de travail. Diverses applications des ponts roulants dans l'industrie comme la fabrication des : automobiles, poteaux, boissons, produits chimiques, papiers, acier, entreposage et distribution etc. [1-2].

Un pont roulant (aussi appelé portique et semi-portique, poutres roulantes, grue industrielle, grue, ou pont roulant aérien) sont des appareils de levage destinés à soulever et à déplacer des charges. Ils se déplacent sur des chemins de roulement parallèles, leur organe de préhension (crochet ou autre accessoire de levage) est suspendu par l'intermédiaire d'un câble et de poulies à un mécanisme de levage (treuil ou palan) susceptible de se déplacer perpendiculairement aux chemins de roulement de l'appareil [3].

I.2 Mécanisme générale

Les ponts roulants couvrent une zone rectangulaire, déplaçant une charge d'un côté à l'autre, vers l'arrière et vers l'avant. Le dispositif de levage, appelé palan, est monté sur un chariot pour le mouvement horizontal à travers une poutre de pont reliée à une ou plusieurs poutres horizontales qui sont supportées à chaque extrémité par des bogies. Ces dernières sont fixées perpendiculairement aux poutres et se déplacent sur des pistes fixes. Ils diffèrent des grues, des portiques, des semi-portiques, des potences et des monorails principalement par ses conceptions. Ils sont constitués :

- D'une structure horizontale en acier se déplaçant sur deux voies de roulement
- De treuils ou palans suspendus ou posés ; ils permettent d'enrouler le ou les câbles ou la chaîne de levage.

Chaque axe de déplacement peut être manuel ou motorisé électriquement. À l'extrémité du câble, est monté le crochet de levage, soit directement soit par l'intermédiaire d'une moufle équipée de poulie (Fig.I.2) [4].

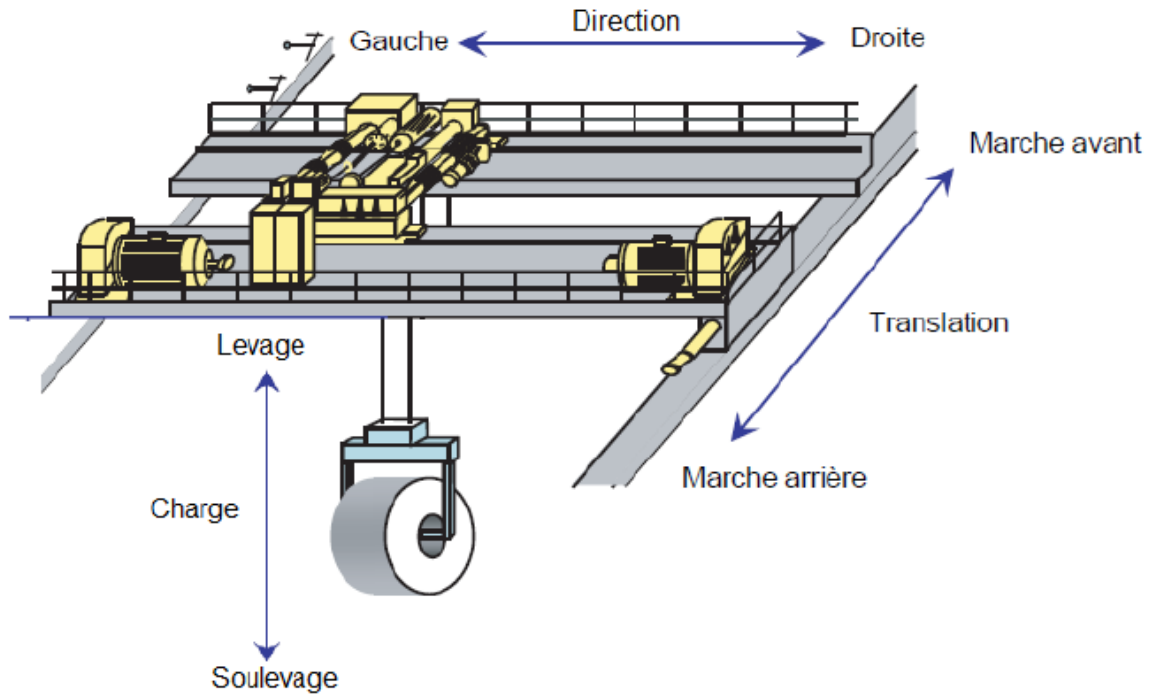


Fig.I.2 Pont roulant : structure générale d'un pont roulant.

I.2.1 Les principaux mouvements

Tel que le montre le croquis de la Fig.I.3, Les différents mouvements d'un pont roulant, translation, direction, levage, sont assurés par les mécanismes décrits [4].

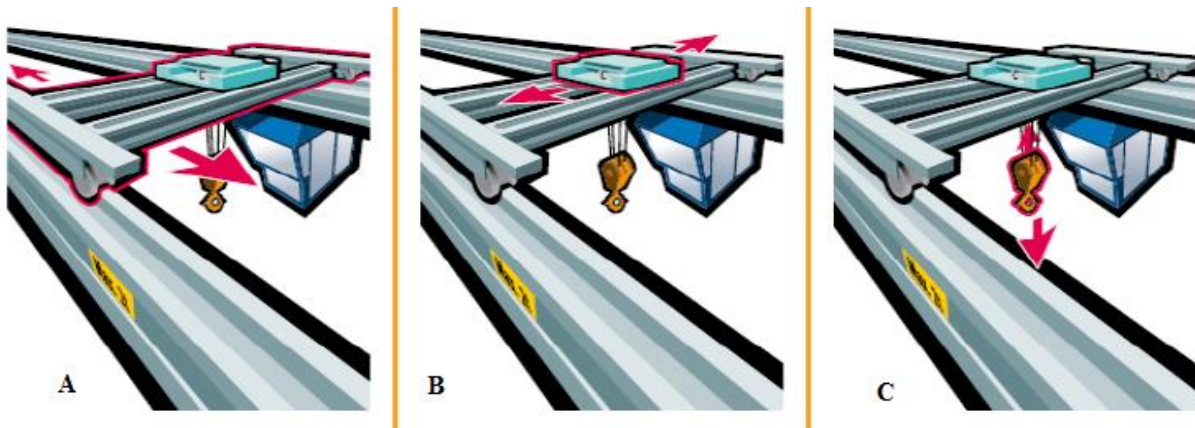


Fig.I.3 Principaux mouvements d'un pont roulant, avec : (A) La translation, (B) La direction et (C) Le levage.

- **Direction** : généré par un déplacement du chariot à gauche et à droite.
- **Translation** : correspondant à un mouvement d'ensemble du pont en avant et en arrière.

- **Levage** : axe vertical ; levage ou descente de la charge par un déplacement d'un mécanisme d'accrochage des charges en haut et en bas.

I.3 Les modèles industriels des ponts roulants

Les ponts roulants sont de différents styles et on peut distinguer 4 familles principales d'appareils dont la construction peut être des deux types suivants :

- Le pont (ou le portique) (Fig.I.4) est dit mono poutre, lorsque sa charpente est réalisée avec une seule poutre (le chariot de levage est alors monorail).
- Le pont est dit bipoutre lorsque sa charpente est constituée de deux poutres, le chariot de levage est alors du type bi rail [4] :

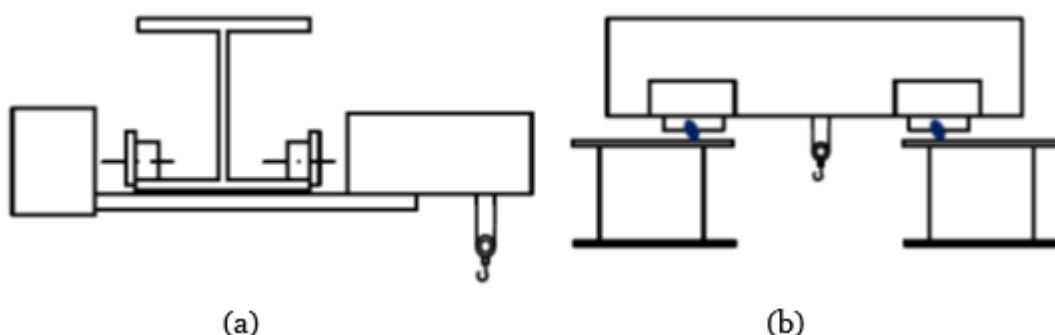


Fig.I.4 Pont roulant, avec : (a) Mono poutre et (b) Bipoutre.

I.3.1 Ponts roulants posés

L'appareil roule sur deux voies de roulement constituées par un rail qui repose sur une poutre de roulement reprise sur les poteaux du bâtiment ou de l'aire de travail ou de stockage par l'intermédiaire de corbeaux (pièce en forme de trapèze rectangle métallique ou en béton qui assure la liaison entre le chemin de roulement et le poteau) (Fig.I.5). Pour une portée donnée (inférieure à 30 m) et une hauteur sous ferme donnée, c'est le type de pont qui permet d'obtenir la meilleure hauteur de levage [1,2].

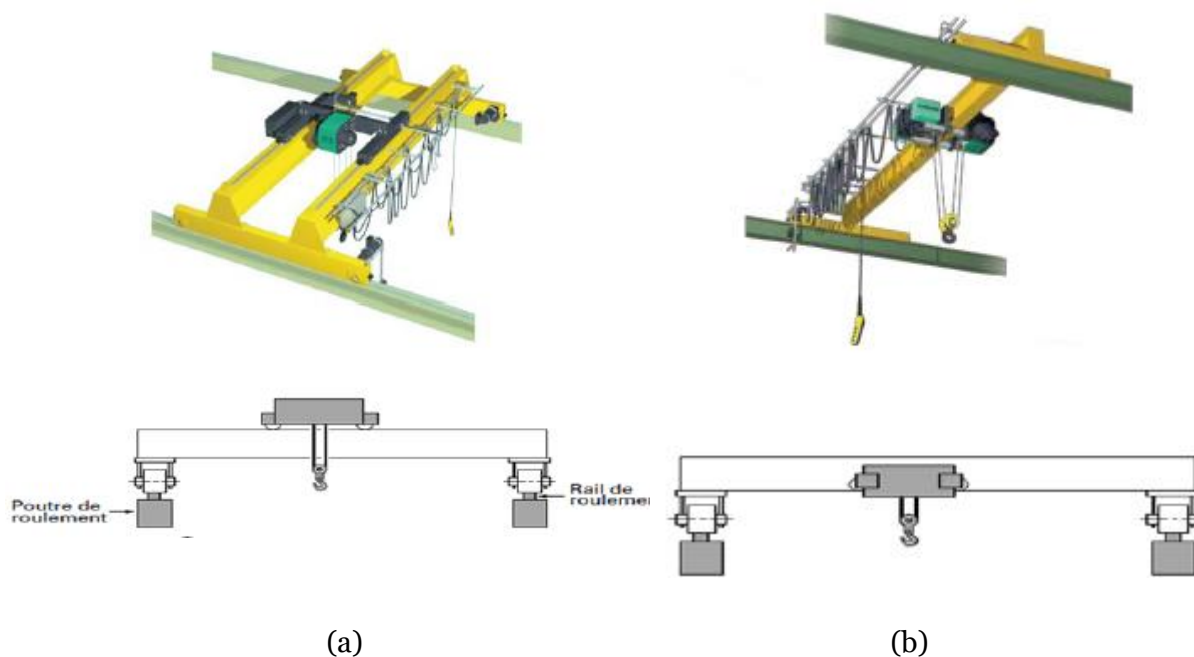


Fig.I.5 (a) Pont roulant bipoutre posé et **(b)** Pont roulant mono poutre posé.

I.3.2 Ponts roulants suspendus

L'appareil roule sur l'aile inférieure d'un profilé repris directement sur les fermes du bâtiment. Ces chemins de roulement peuvent comporter plus de deux voies de roulement. Ils peuvent en outre être équipés de plusieurs moyens de levage pour desservir toute la surface du bâtiment. Dans le cas de portée très importante on installera donc plutôt ce type de matériel (Fig.I.6).



Fig.I.6 Pont roulant : **(a)** Mono poutre suspendue et **(b)** Bipoutre suspendu.

I.3.3 Portiques

L'appareil roule sur deux voies de roulement constituées par un rail fixé directement au sol. Ils sont reliés par deux jambes ou palées (Fig.I.7). Ces portiques sont installés dans le cas où le bâtiment ne pourrait supporter les réactions qu'entraînerait l'installation d'un pont roulant ou si les transformations nécessaires s'avéraient très onéreuses, il en serait de même s'il n'y avait pas de bâtiment du tout (parc extérieur). Un déport des poutres de roulement permettant d'aller chercher des charges à l'extérieur des rails de roulement est possible, il se nomme, dans ce cas, portique à avant- bec.

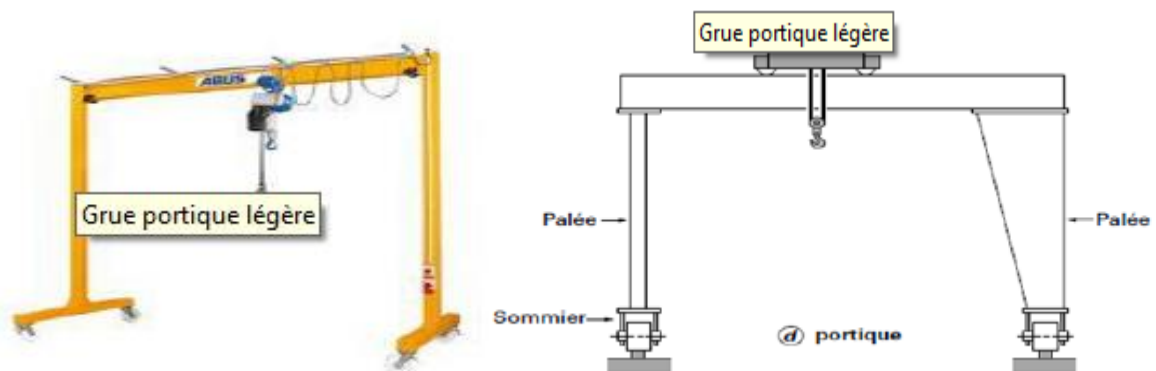


Fig.I.7 Portique.

I.3.4 Semi-portiques

L'appareil roule sur deux voies de roulement. L'une est constituée par un rail au sol, alors que l'autre s'appuie sur la charpente du bâtiment et roule sur un chemin de roulement aérien. Cette formule est utilisée le plus souvent en complément d'un pont roulant important qui assure la liaison entre les postes équipés chacun d'un semi-portique (Fig.I.8).

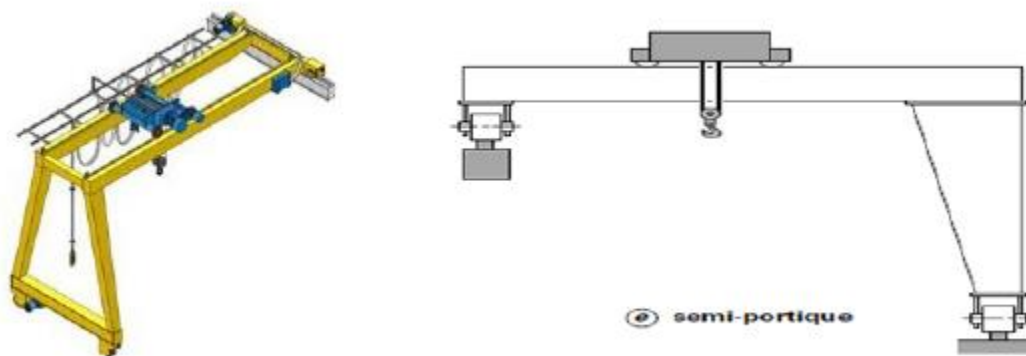


Fig.I.8 Semi portique

I.4 Commande des ponts roulants

Les ponts roulants peuvent être commandés de plusieurs façons : soit par une commande cabine, commande au sol ou par radio commande [5]. Si on a dans une installation plusieurs postes de commande pour un appareil de levage, et pour assurer la sécurité des opérateurs, un seul poste doit être fonctionnel à la fois.

I.4.1 Commande cabine

Les commandes en cabine permettent d'avoir une meilleure visibilité au pontier de la charge et de l'itinéraire à emprunter. Elles peuvent être fixes, mobiles ou orientables (Fig.I.9). Comme elles protègent le pontier contre les intempéries (chaleur, froid, courants d'air, averses de pluie) et les nuisances industrielles (rayonnements, poussières, vapeurs nocives) à condition d'être spécialement équipées [5].

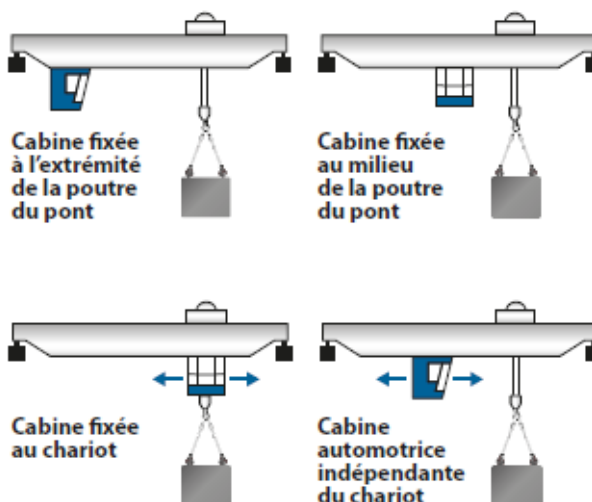


Fig.I.9 Commande cabine.

I.4.2 Commande au sol

Les commandes au sol s'effectuent à l'aide d'une boîte à boutons pendante ou d'une télécommande. Elles sont admissibles pour des vitesses de direction et translation compatibles avec le déplacement d'un homme au pas et avec l'encombrement au sol [5].

I.4.2.a Boite à boutons pendante

La boîte à boutons poussoirs est suspendue au pont roulant ; elle peut être selon les cas suivants [5] :

- ✓ Fixée à l'extrémité de la poutre principale.
- ✓ Mobile le long de la poutre principale (cette disposition est particulièrement conseillée).
- ✓ Fixée au palan ou au chariot porte treuil.

I.4.3 Radiocommandes

Pour transmettre les ordres du boîtier de commande aux pont roulant on peut utiliser :

- ✓ Voie hertzienne
- ✓ Rayonnement infrarouge

Chaque poste de commande devra désigner clairement le pont roulant commander et ne permettre la mise en mouvement que du ce dernier. Si on a plus plusieurs sources de commande sans fil, seul un poste peut être active à la fois.

I.5 Le fonctionnement des ponts roulants

Les ponts roulants peuvent être conçus pour un contrôle manuel, contrôle manuel assisté par des fonctionnalités semi-automatiques ou pour un contrôle totalement automatique, en fonction des besoins requis par chaque utilisation spécifique. La Fig.I.10 illustre un diagramme qui permet de classifier les ponts roulants selon leur degré d'automatisation [6,7].

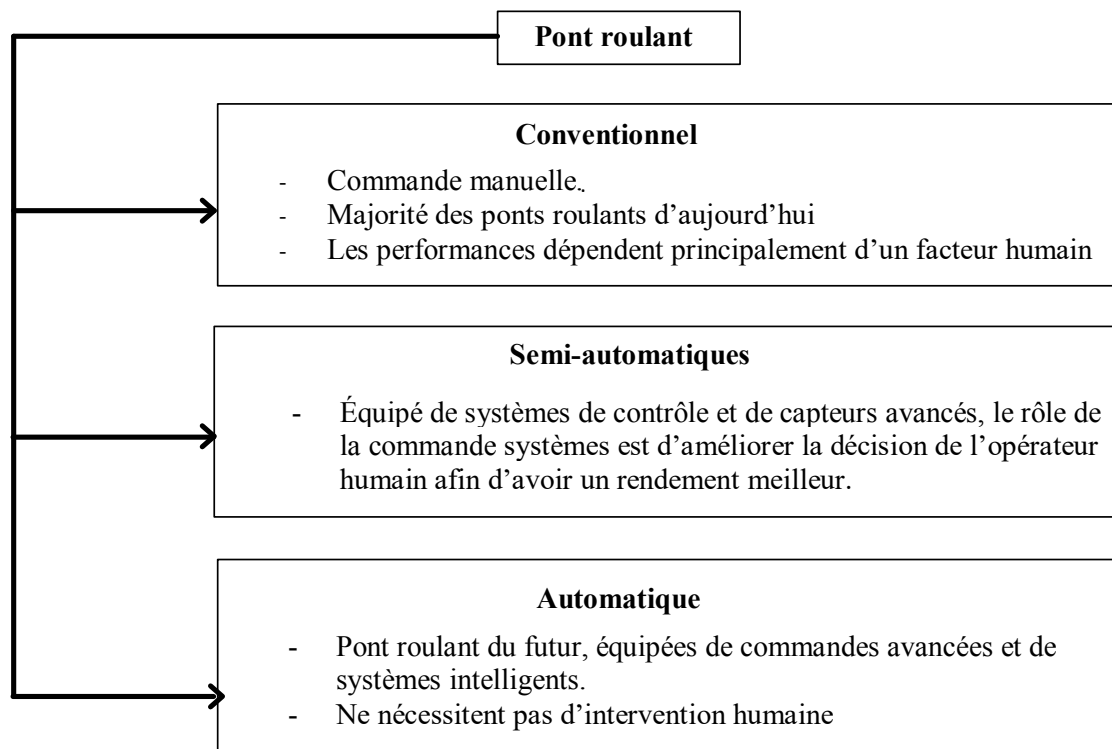


Fig.I.10 Classification des ponts roulants avec leurs degrés d'automatisation.

I.5.1 Fonctionnement manuel

On utilise ce mode si la tâche réclame la présence de l'opérateur en permanence. Le pontier doit actionner les commandes délicatement en évitant d'exercer des mouvements brutaux et entrecoupés pour n'avoir pas un balancement de la charge. La Fig.I.11 démontre un exemple d'une commande manuelle basé sur la manipulation des combineurs.

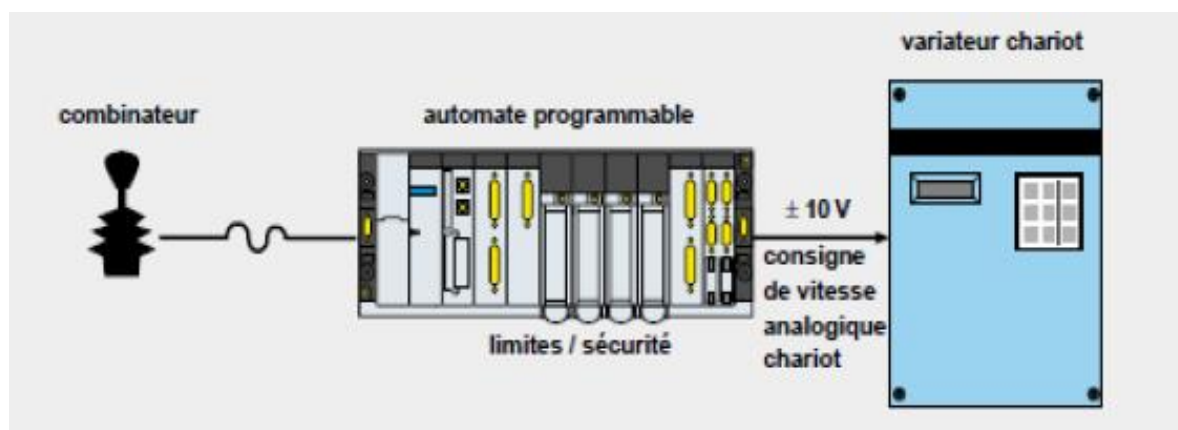


Fig.I.11 Commande manuelle d'un pont roulant.

I.5.2 Fonctionnement semi-automatique

Dans ce mode on a la position du chariot commandé par l'opérateur et le contrôle du ballant automatisé (Fig.I.12). Le système d'anti-ballant modifie la consigne de vitesse envoyée aux variateurs de chaque mouvement afin que le phénomène de balancement soit limité systématiquement.

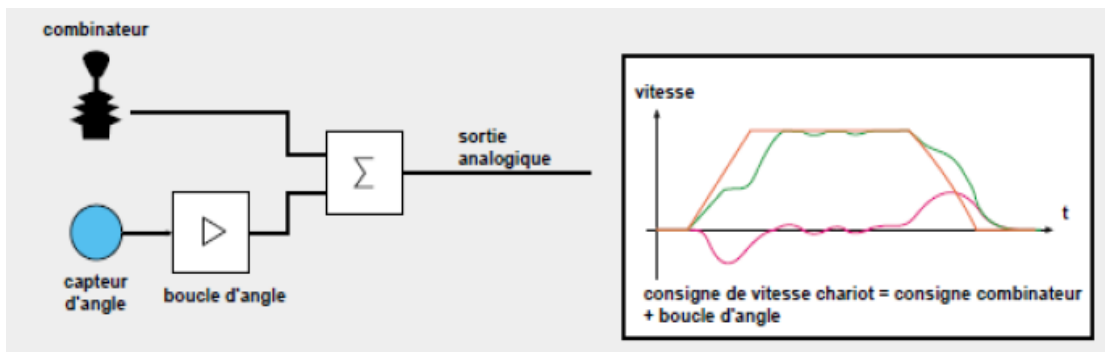


Fig.I.12 Exemple d'un mode semi-automatique

I.5.3 Fonctionnement automatisé

Les ponts roulants automatisés fonctionnent, durant leur phase automatique sans la participation de l'opérateur, Contrairement aux machines manuelles et semi-automatiques qui nécessitent une présence humaine dans leur voisinage (Fig.I.13).

Dans ce cas, la fonction d'automatisme :

- ✓ Contrôle le comportement dynamique (déplacement et balancement) de la charge au cours du transfert.
- ✓ Positionne la charge sur la cible sans ballant résiduel, quelles que soient les perturbations

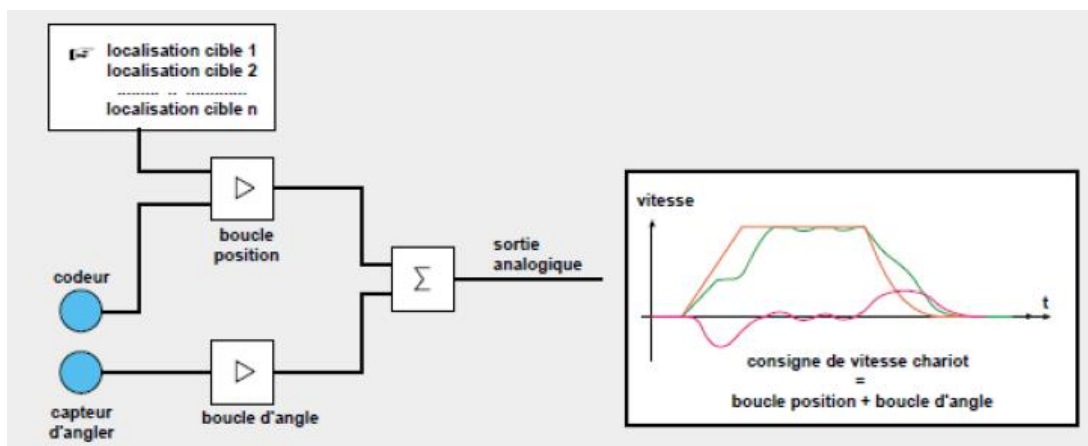


Fig.I.13 Exemple d'un mode automatique.

I.6 Ponts roulants avec bras manipulateur

La conception des ponts roulants avec bras manipulés a été rendue possible par des connaissances scientifiques et techniques en mécanique et en techniques électriques. Ces machines sont destinées à augmenter les possibilités de l'homme pour leur rapidité de leurs déplacements et de leurs actions, et pour l'amplification de sa force et de son champ d'action notamment dans l'accomplissement de tâches pénibles, dangereuses et répétitives. Les fonctions sont diverses à savoir : le déplacement (saisie, transport et dépôts) (Fig.I.14), la manipulation au sens large du terme, l'augmentation de la sécurité, amélioration de la qualité du produit grâce à une meilleure répétitivité et d'une flexibilité du produit. Les ponts roulants avec bras manipulateurs agissent sur l'environnement. Leur état est mesuré par des capteurs et modifié par des actionneurs. Leur installation est demandée dans les diverses industries.

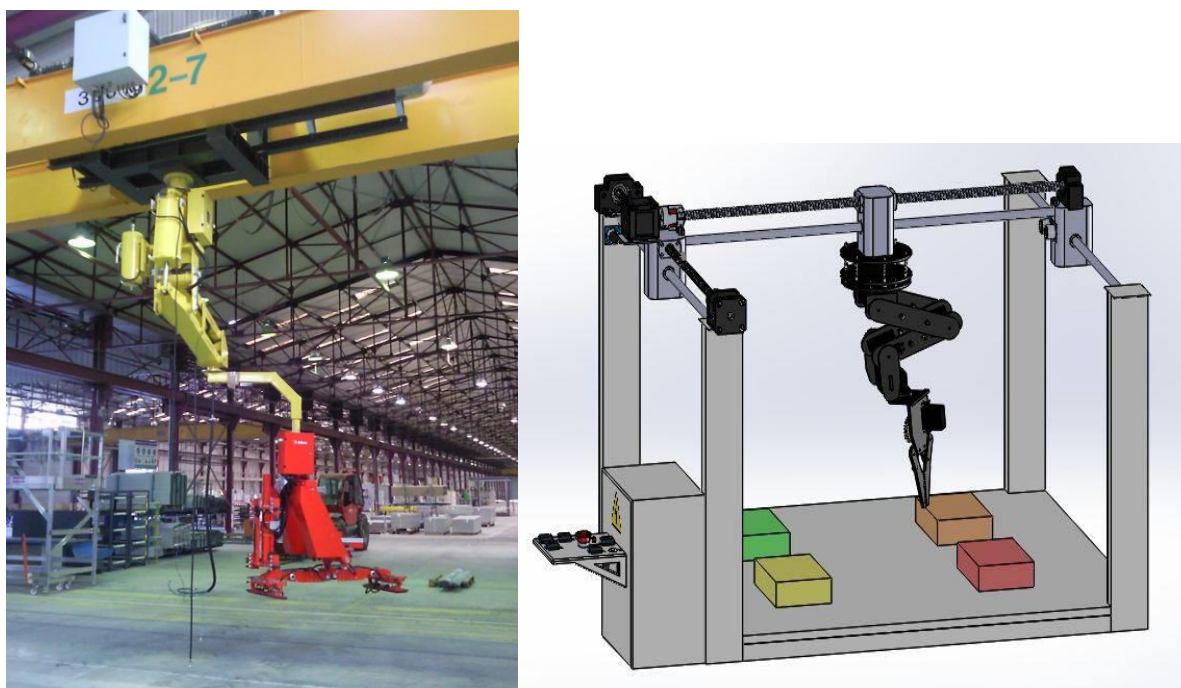


Fig.I.14 Pont roulant avec bras manipulateur et prototype.

Conclusion

Notre objectif dans ce chapitre était d'introduire la structure des ponts roulants, présenter leurs différents types et les classer selon les besoins ainsi que montrer les modèles de commandes existants dans l'industrie actuellement.

Le bras manipulateur est fixé sur le chariot, et utilisé pour effectuer une variété de tâches avec une grande précision (le contrôle humain).

Le but d'automatiser les ponts roulants est de perfectionner ses performances en termes de rapidité, de précision et de sécurité. Et pourtant, cette automatisation est confrontée à de nombreux défis importants ayant des origines multiples :

- ✓ Manipulation et interaction avec le monde physique.
- ✓ Flexibilités, et les environnements non structurés.
- ✓ Interaction avec l'homme : sécurité, coopération et coordina.

Références bibliographiques

[1] C. Pelletier, Appareils de levage : Généralités, *Techniques de l'ingénieur. L'Entreprise industrielle*, AG7010, 2000.

[2] C. Pelletier, Ponts roulants et portiques, *Techniques de l'ingénieur. L'Entreprise industrielle*, AG7020, 2000.

[3] E. Fadier, Automatisation et sécurité. *Techniques de l'ingénieur. L'Entreprise industrielle*, AG3530, 2009.

[4] O. Sawodny, H. Aschemann and S. Lahres, An automated gantry crane as a large workspace robot, *Control Engineering Practice*, **10**, 1323-1338, 2002.

[5] J. Pirotte, N. V. Gillieaux and E. Warnier, Contrôle du ballant lors du déplacement d'une charge suspendue à un pont roulant, *Revue scientifique des ISILF*, **21**, 79-99, 2007.

[6] E. M. Abdel-Rahman, A. H. Nayfeh and Z. N. Masoud, Dynamics and control of cranes : A review. *Journal of Vibration and control*, **9**, 863-908, 2003.

[7] O. Sawodny, H. Aschemann and S. Lahres, An automated gantry crane as a large workspace robot. *Control Engineering Practice*, **10**, 1323-1338, 2002.

Chapitre II

Aperçu général sur les bras robotiques

Un bras robotique, parfois appelé robot industriel, est souvent décrit comme un bras « mécanique ». C'est un dispositif qui fonctionne de la même manière qu'un bras humain, avec un certain nombre d'articulations qui se déplacent le long d'un axe ou peuvent tourner dans certaines directions. En fait, certains bras robotiques essaient d'imiter les mouvements exacts des bras humains. Ils sont, dans la plupart des cas, programmables et utilisés pour effectuer des tâches spécifiques, le plus souvent pour la fabrication et les applications industrielles. Il peut s'agir de petits appareils qui effectuent des tâches complexes et détaillées, assez petits pour être tenus dans une main ; ou si grand que leur portée est assez grande pour construire des bâtiments entiers.

Les bras robotiques ont été conçus à l'origine pour aider les usines de production de masse, notamment dans la fabrication de voitures. Ils ont également été mis en œuvre pour atténuer le risque de blessure pour les travailleurs et pour entreprendre des tâches monotones, afin de permettre aux travailleurs de se concentrer sur les éléments les plus complexes de la production. Ces premiers bras robotiques étaient principalement utilisés pour entreprendre des tâches de soudage simples et répétitives. À mesure que les technologies se développent, en particulier la vision robotique et la technologie des capteurs, le rôle des bras robotiques évolue. Ce chapitre fournit un bref aperçu des bras robotiques.

II.1 Historique

Le bras robotique a été inventé en 1954 par l'autodidacte George Devol qui a déposé une demande de brevet pour un bras mécanique qui pouvait être programmé pour répéter des tâches précises. M. Devol a ensuite expliqué son concept à un ingénieur du nom de Joseph Engelberger et les deux ont ensemble appelé le concept Unimation [1].

Joseph Engelberger et George Devol fonde aux États-Unis la première entreprise de robots, Unimation, en 1956. Puis, en 1962, General Motors a mis en œuvre le bras robotique Unimate dans sa chaîne de montage pour la production de voitures. Quelques années plus tard, un ingénieur en mécanique de l'université de Stanford, Victor Scheinman, développait un bras robotique qui fut l'un des premiers à être entièrement contrôlé par un ordinateur en 1969. Ce robot industriel, connu sous le nom de Stanford Arm, a été le premier bras robotique à six axes et a influencé un

certain nombre de robots commerciaux qui ont suivi. Une entreprise japonaise, Nachi, a développé son premier bras robotique industriel hydraulique en 1969 et, après cela, une entreprise allemande, Kuka, a lancé le premier bras robotique commercial à six axes, appelé Famulus, en 1973.

Principalement, ces robots ont été utilisés pour des tâches de soudage par points dans les usines de fabrication, mais au fur et à mesure que la technologie se développait, la gamme de tâches que les bras robotisés pouvaient effectuer s'est également élargie. Les progrès technologiques comprennent la variété croissante des outils de fin de bras qui sont devenus disponibles. Cela signifie que les bras robotisés peuvent effectuer beaucoup de tâches au-delà du soudage en fonction des outils qui sont fixés à l'extrémité de leurs bras. Les innovations actuelles dans les outils de fin de bras comprennent les têtes d'outils d'impression 3D, les dispositifs de chauffage pour mouler et plier les matériaux et les dispositifs d'aspiration pour plier la tôle [3].

II.2 Les segments et les mouvements du bras robotique

Le bras robotique est le type de robot le plus fabriqué. Il est généralement composé de sept segments avec six articulations entraînées à l'aide des servo moteurs. Un utilisateur peut contrôler le bras robotique via un ordinateur, un microcontrôleur en contrôlant les servo moteurs dans les articulations

En raison de la répétabilité et de la précision, les bras robotiques sont utilisés pour des fonctions difficiles, répétitives et souvent ennuyeuses pour les humains. Également pour les fonctions considérées comme dangereuses pour les êtres humains, elles peuvent être exécutées à l'aide de bras robotiques.

Le robot a six articulations, il peut pivoter de six manières différentes. On dit qu'il a six degrés de liberté. Le bras humain, par comparaison, a sept degrés de liberté. Comme l'épaule humaine est fixée à une structure fixe (le corps et le torse), l'équivalent « épaule » d'un bras robotique est également monté sur un élément de base fixe qui ne bouge pas. Les segments restants du bras robotique sont ceux qui se déplacent dans des directions différentes.

II.3 Effecteur terminal

Un effecteur terminal est un dispositif situé à l'extrémité d'un bras robotique. Il est conçu pour interagir avec l'environnement dans lequel le bras robotique doit être

utilisé. On peut considérer l'effecteur terminal comme l'application qui permet au bras robotique d'exécuter diverses fonctions. Le travail du bras robotique consiste à déplacer l'effecteur d'extrémité d'un endroit à un autre en fonction des commandes que l'utilisateur envoie via le contrôleur.

La conception de l'effecteur terminal dépendra des applications pour lesquelles le bras robotique est destiné à être utilisé. L'effecteur terminal est également connu sous le nom de main robotique. Il peut être conçu pour effectuer des opérations de filage, de préhension, de soudage et d'assemblage.

Un bras robotisé peut être équipé de divers effecteurs terminaux. L'un des effecteurs terminaux les plus couramment installés est celui qui ressemble à la main humaine. Il est largement utilisé pour ramasser, saisir et transporter différents types d'objets.

L'effecteur d'extrémité de la main humaine a des capteurs de pression intégrés qui communiquent à quel point l'effecteur d'extrémité saisit quelque chose. Cette rétroaction est très importante car elle empêche l'effecteur de casser un objet en le tenant trop fermement ou en faisant tomber un objet en le tenant trop lâchement.

II.4 Les types des bras robotiques

Il existe différents types de bras robotiques qui sont utilisés à des buts particuliers.

II.4.1 Robot linéaire

Un robot qui dispose d'actionneurs linéaires coopérant avec des moteurs linéaires liés à un axe linéaire est connu comme un robot linéaire (également connu comme portique ou cartésien). Ce lien peut être des liaisons fixes ou flexibles entre les actionneurs et le robot. Le moteur linéaire est fixé directement à l'axe linéaire [4].

Les robots qui utilisent deux moteurs pour contrôler un axe linéaire défini comme robots portique. Chaque moteur a une distance orthogonale limitée à l'axe linéaire [5]. Les vis à billes suivent les mêmes principes qui utilisent des moteurs linéaires ou des moteurs rotatifs. Ce genre de robots accomplir habituellement des tâches comme la palettisation, l'unification, l'empilage, la saisie des commandes, le chargement, et la mesure des coordonnées.

Toute action effectuée par les axes x et y doit avoir une précision de position inférieure à $\pm 5,0$ mm et répétition de $\pm 1,0$ mm. La figure II.1 montre le robot portique Hercules x-y fabriqué aux États-Unis.

La figure II.2 montre une portée horizontale cartésienne et son symbole où d_1 , d_2 et d_3 représentent respectivement les axes x, y et z [6].

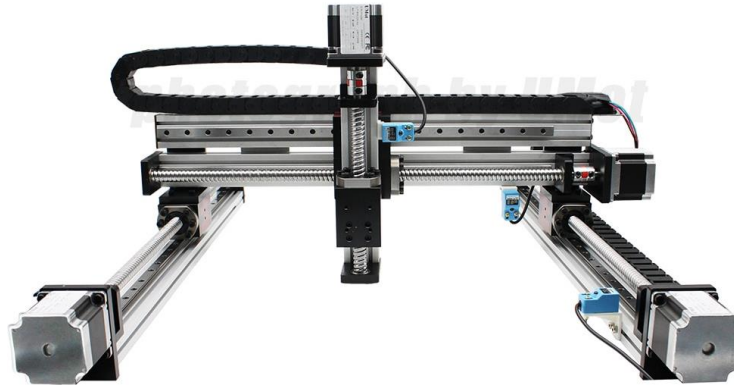


Fig.II.1 Robot portique Hercules x-y.

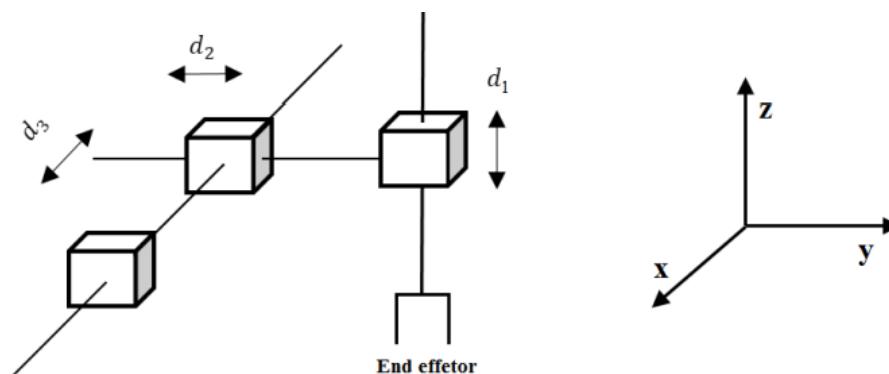


Fig.II.2 Portée horizontale cartésienne et son symbole (à droite).

L'expression mathématique concernant le mouvement du robot linéaire peut être écrit comme :

$$q = \begin{bmatrix} q_1 \\ q_2 \\ \vdots \\ q_n \end{bmatrix} \quad (\text{II.1})$$

Où $q \in \mathbb{R}$ avec une dimension $n \times 1$, représente la valeur de chaque articulation.

Dans l'équation (II.1), il existe n articulations. Le robot se déplace en fonction de la position donnée du manipulateur. Le mouvement souhaité du robot peut être représenté par un vecteur comme suit [7].

$$q_d = \begin{bmatrix} q_{d_1} \\ q_{d_2} \\ \vdots \\ q_{d_n} \end{bmatrix} \quad (\text{II.2})$$

Où q_d est la position souhaitée de l'effecteur terminal.

Si nous définissons $q_{\square i} = q_{d_i} - q_i$ comme la position souhaitée pour l'effecteur terminal, les positions (x, y) sont des fonctions des angles qui peuvent être exprimé comme suit [7] :

$$x_i = f_i(q) \quad \text{pour } i = 1, 2, \dots, k \quad (\text{II.3})$$

Dans notre cas la valeur de $i = 2$, ce qui implique pour le nombre d'effecteurs terminaux puisque nous ne traitons que des positions (x, y) . Si on prend q_1 et q_2 les articulations des déplacements, la cinématique directe sera la suivante [7] :

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} q_1 \\ 0 \end{bmatrix}; \quad \begin{bmatrix} x_2 \\ y_2 \end{bmatrix} = \begin{bmatrix} q_1 \\ q_2 \end{bmatrix} \quad (\text{II.4})$$

Où le premier vecteur est considéré par la valeur du premier déplacement q_1 et le deuxième vecteur considère le mouvement de q_1 et q_2 par rapport aux axes x et y .

II.4.2 Robot cylindrique

Les robots cylindriques ont deux articulations prismatiques : une articulation rotative pour la tâche de positionnement et l'effecteur terminal du robot forme un espace de travail cylindrique. L'idée principale du robot cylindrique consiste à monter un bras horizontal qui se déplace vers l'avant et vers l'arrière directions. Le bras horizontal est lié à un chariot qui monte et descend et connecté à la base rotative.

Le schéma du robot cylindrique et son symbole sont indiqués dans le Fig.II.3 [6]. Étant donné que les deux unités se déplacent sur la base, l'espace de travail est un espace annulaire du cylindre.

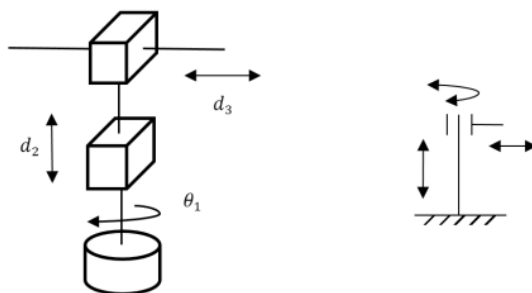


Fig.II.3 Schéma du robot cylindrique et son symbole.

Lorsque le bras du robot possède une révolution et deux articulations prismatiques, il peut fonctionner en axe z et chaque articulation peut être atteinte par ce robot et représenté par les coordonnées cylindriques. Comme le montre la figure II.4, le robot peut entrer et sortir (z direction), peut s'élever dans la direction y et peut tourner dans la direction θ . Le bras peut bouger dans les directions entre les limites supérieure et inférieure spécifiques [8].

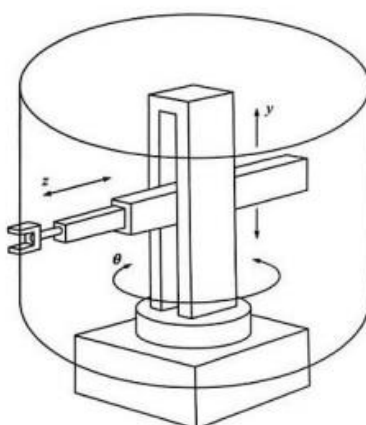


Fig.II.4 Configuration d'un robot cylindrique.

Il est possible d'avoir des articulations supplémentaires pour l'effecteur terminal de sorte que le bras horizontal peut s'étendre. Ce type de robots est largement utilisé dans la fabrication électronique [9].

II.4.3 Robot parallèle

Un robot parallèle a un effecteur de fin avec n D.D.L. qui est connecté à une base fixe. La connexion est effectuée par au moins deux chaînes cinématiques indépendantes qui fournissent les mouvements du robot. Un manipulateur parallèle généralisé a un mécanisme de chaîne cinématique en boucle fermée où le manipulateur

est relié à la base. La figure II.5 montre le potentiel du premier robot parallèle breveté aux États-Unis [10].

Il existe différentes définitions et différents types de robots parallèles, mais les propriétés les plus courantes sont les suivantes [11] :

- L'effecteur d'extrémité doit être soutenu par au moins deux chaînes. Chacune des chaînes doit avoir au moins un actionneur simple.
- Le nombre d'actionneurs doit être égal au nombre de DDL de l'effecteur terminal.
- Il ne devrait y avoir aucune mobilité du manipulateur lorsque les moteurs sont verrouillés.

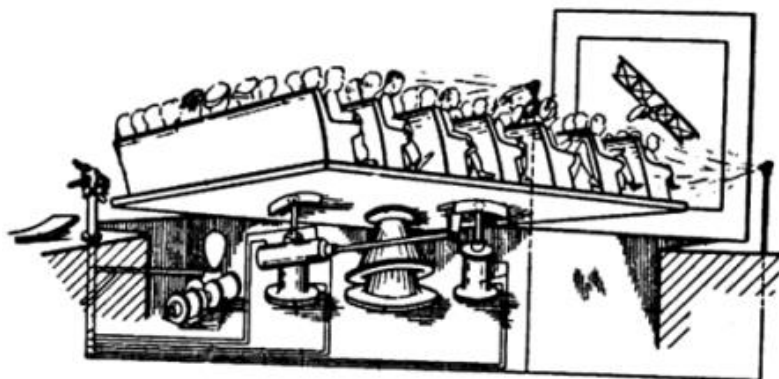


Fig.II.5 Le potentiel du premier robot parallèle breveté aux Etats-Unis.

Avec deux plates-formes, les robots parallèles fournissent généralement un mouvement relatif entre la plate-forme mobile et la plate-forme fixe. Les robots parallèles sont devenus essentiels dans l'industrie et dans les universités, les recherches récentes sur les théories des mécanismes, analyse de la mobilité, modélisation dynamique et cinématique, l'électronique et les optimisations ont fait des développements remarquables sur le parallèle robots [10].

Il existe différentes configurations de robots parallèles, mais deux conceptions cinématiques ont devenir populaire. La première conception est un robot parallèle illustré à la figure II.6, qui a un trépied avec trois axes reliant l'effecteur terminal à la plate-forme mobile et à la base, et a un poignet avec 2 ou 3 D.D.L. La deuxième conception est un hexapode à six axes qui entraîne un déplacement spatial complet comme le montre la figure II.7.

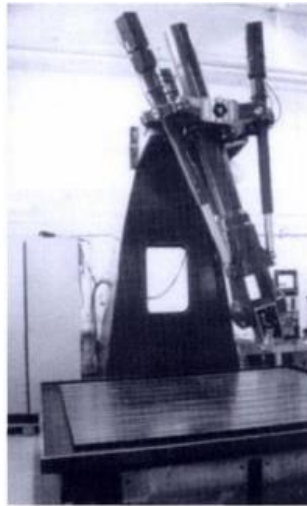


Fig.II.6 Robot parallèle (trépied avec trois axes).



Fig.II.7 Robot parallèle (hexapode à six axes).

II.4.4 Robot sphérique

Le robot sphérique (également connu sous le nom de robot polaire) est énorme en termes de taille et a un bras télescopique. Les mouvements de base du robot sphérique sont la rotation à la base et angulairement de haut en bas au bras [8].

Les robots sphériques ont au moins deux articulations mobiles et une articulation fixe. Le diagramme et le symbole du robot sphérique sont illustrés à la figure II.8. Le mouvement du robot sphérique se compose de trois étapes de mouvement ; le premier mouvement définit la rotation de la base le long de l'axe vertical. Le deuxième mouvement définit la rotation du bras et enfin le troisième mouvement définit le mouvement d'entrée et de sortie. L'espace de travail du robot sphérique dépend du volume du globe de la sphère [6].

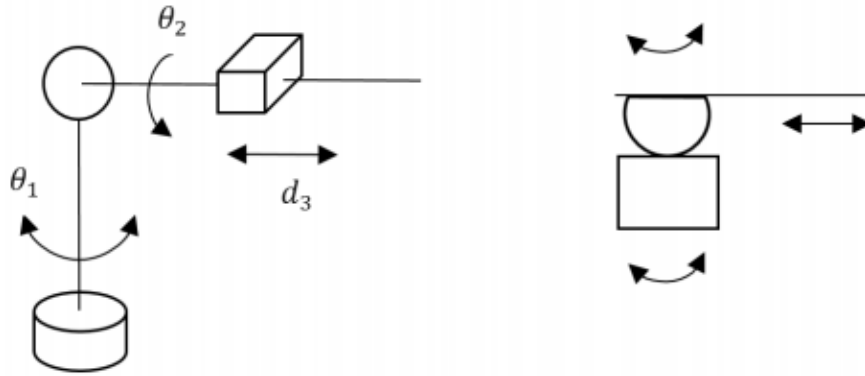


Fig.II.8 Diagramme et symbole du robot sphérique.

L'espace de travail du robot est l'espace entre deux hémisphères concentriques. Quand le bras est complètement rétracté, la portée du bras est l'hémisphère interne et lorsque le bras est complètement redressé, la portée est l'hémisphère externe [12]. Une configuration sphérique typique du robot est illustrée dans la figure II.9 [13].

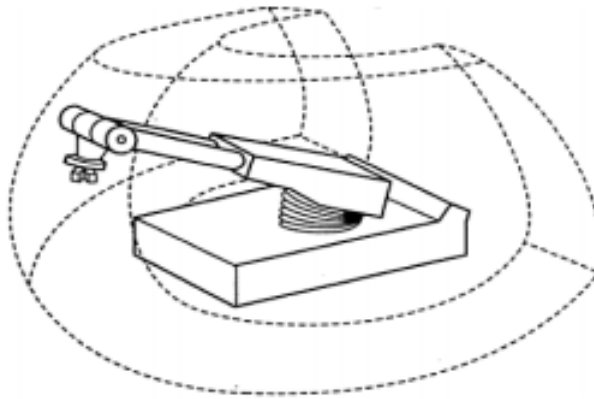


Fig.II.9 Configuration d'un robot sphérique.

II.4.5 SCARA

Le bras de robot d'assemblage de conformité sélective (SCARA) a été conçu et inventé au début des années 1960 au Japon. Les robots SCARA sont parfaits pour les applications qui nécessitent des mouvements point à point à grande vitesse et répétitifs. C'est pourquoi SCARA est largement utilisé dans les opérations d'assemblage [6]. Le mouvement spécial de l'effecteur terminal fait SCARA idéal pour les tâches qui nécessitent un mouvement uniforme et des accélérations sous forme circulaire [14].

SCARA se compose de deux articulations rotatives parallèles et d'une articulation prismatique. Les articulations rotatives peuvent se déplacer le long du plan horizontal et l'articulation prismatique se déplace le long du plan vertical.

Une des caractéristiques particulières de SCARA est que le robot est lisse tout en fonctionnant sur les axes x et y , mais très fort par rapport à l'axe z . La figure II.10 montre le schéma de SCARA.

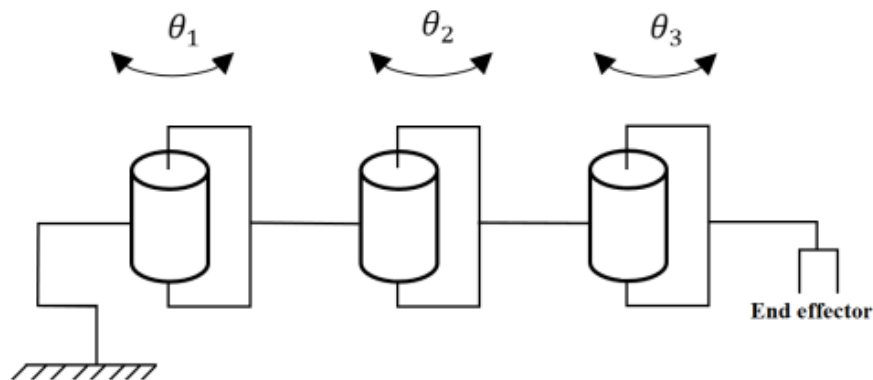


Fig.II.10 Schéma de SCARA.

Une conception spéciale de SCARA qui a été construite en Suède et qui est connue sous le nom de robot Spine (de la colonne vertébrale), est conçue de telle sorte qu'elle se compose de plusieurs disques qui sont reliés entre eux via deux paires d'actionneurs de câbles hydrauliques. Chaque robot a au moins quatre câbles dans chaque partie rigide, ce qui donne un total de huit câbles avec 4 D.D.L.

Le robot fonctionne lorsque ces câbles sont tirés par les cylindres placés dans la base du robot vertébrale [6]. Le schéma du robot Spine est illustré à la figure II.11.

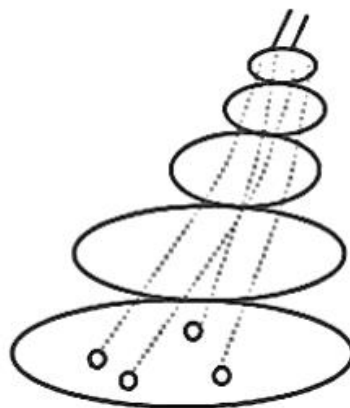


Fig.II.11 Schéma du robot Spine.

Le bras SCARA est capable de ramasser une pièce verticalement à partir d'une table placée horizontalement et se déplacer le long du plan horizontal jusqu'à un point souhaité et d'accomplir la tâche d'assemblage en abaissant le bras et en plaçant la pièce à robot SCARA typique [12].



Fig.II.12 Robot SCARA.

II.4.6 Robot articulé

Les robots articulés (également appelés robots révolutionnaires) ont trois axes fixes connectés à deux bases de révolution. Toutes les articulations d'un bras articulé sont révolutionnaires et très proches du bras humain. La figure II.13 montre le schéma et le symbole d'un robot articulé [6].

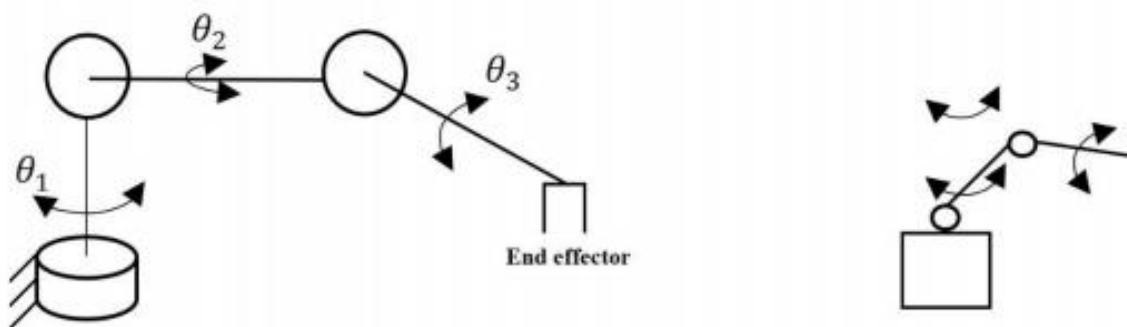


Fig.II.13 Schéma et le symbole d'un robot articulé.

Les objets rigides en mouvement sont appelés liens, les articulations de révolution sont appelées charnières et articulation prismatiques sont appelés articulations coulissantes. Chaque articulation définit le mouvement relatif de deux autres objets qu'il relie et qui détermine le sous-ensemble de l'espace de configuration entier.

Chaque sous-ensemble de configuration est une position différente pour chaque lien. Ces sous-ensembles sont simples à mesurer en considérant une distance ou un angle avec chaque articulation.

Un bras robotique peut être considéré comme un exemple typique de robot articulé. Il faut considérer que la dimension de l'espace de configuration augmente avec le nombre d'articulations, mais la vitesse de fonctionnement est limitée en raison différentes charges utiles au niveau du manipulateur et de l'environnement non linéaire [16].

80% des robots enregistrés sont articulés et jusqu'à 20% sont des robots linéaires. La figure II.14) montre un robot articulé typique [15].



Fig.II.14 Robot articulé typique.

II.5 Comparaison entre les bras robotiques fondamentaux

Configuration	Avantages	Inconvénients
<p>Cartésien (trois axes linéaires)</p> <p>x : parcours de base y : hauteur z : attente</p>	<ul style="list-style-type: none"> • Visualisation Facile • Structure rigide • Programmation facile hors ligne • Arrête mécanique facile. 	<ul style="list-style-type: none"> • Atteindre seulement avant et arrière • Nécessite une grande surface. • Les axes sont difficiles à sceller.
<p>Cylindrique (1 axe tournant et deux axes linéaires)</p> <p>Θ: rotation de base Φ: hauteur z: atteinte</p>	<ul style="list-style-type: none"> • Peut atteindre tout autour • Axes y,z rigides • Θ- axe facile à sceller 	<ul style="list-style-type: none"> • Impossible d'atteindre au-dessus de lui-même • y-z-axes difficiles à sceller • N'attendre pas les obstacles autour. • Mouvement horizontal circulaire
<p>Sphérique (2 axes rotatives et un axe linéaire)</p> <p>Θ : rotation de la base Φ : angle d'élévation z : atteinte</p>	<ul style="list-style-type: none"> • Peut atteindre tous autour. • Peut atteindre au-dessus ou sous les obstacles. • Grand volume de travail 	<ul style="list-style-type: none"> • Impossible d'atteindre au-dessus de lui-même. • Atteinte verticale courte
<p>Articulateur (3 axes rotatifs)</p> <p>Θ : rotation de la base Φ : angle d'élévation Ψ : angle de portée</p>	<ul style="list-style-type: none"> • Peut atteindre au-dessus ou sous les obstacles. • Plus grande zone de travail pour une plus petite surface du sol. 	<ul style="list-style-type: none"> • C'est difficile de le programmer hors ligne • Plusieurs façons d'atteindre un point • Le plus complexe robot.

Tableau II.1 Comparaison entre des bras fondamentaux.

Conclusion

Dans ce chapitre on a présenté les différents types des robots qui existe ainsi une comparaison entre leurs caractéristiques. Notre choix du bras manipulateur à six degrés de libertés est basé sur ces avantages, et aussi il répond à nos besoins pratiques.

Références bibliographiques

- [1] Site web : « <https://www.designrobotics.net/robotic-arms-in-manufacturing/> » Consulté le 04/06/2020
- [2] Site web : « <https://www.kuka.com/en-au/about-kuka/history> », Consulté le 09/06/2020
- [3] Site web : « <http://iptmajorprojectjacobheffernan.weebly.com/history-of-the-robotic-arm.html> », Consulté le 14/06/2020
- [4] Khatib, S., “Springer Handbook of Robotics”, USA: Springer-Verlag Berlin Heidelberg, 2008.
- [5] Gieras, F. J., Piech, J. Z., Tomczuk, I. “Linear Synchronous Motors : Transportation and Automation Systems, Second Edition” USA: Taylor and Francis Group, LLC, pp.407-413, 2012.
- [6] Kuttan, A., “Robotics”, India: I.K. International publishing house, 2007.
- [7] Sanchez, S. P., Cortes, R. F., “Cartesian Control for Robot Manipulators, Robot Manipulators Trends and Development”, 2010.
- [8] Saha, S. K., “Introduction to Robotics”, India: Tata McGraw-Hill publishing company Ltd., 2008.
- [9] Mathia, K., “Robotics for Electronics Manufacturing: Principles and Applications in Cleanroom Automation”, UK: Cambridge University Press, 2010.
- [10] Zhang, D., “Parallel Robotic Machine Tools”, Canada: University of Ontario Institute of Technology (UOIT), pp.1-5, 2010
- [11] Merlet, J. P., “Parallel Robots: Second Edition”, Netherlands: Springer, 2006.
- [12] Roa, P. N., “CAD/CAM Principles and Applications 3rd Edition”, India: Tata McGraw Hill Education private Ltd., 2010.
- [13] Bagad, V. S., “Mechatronics” India: Technical Publication Pune, 2008.
- [14] Nof, Y. S., “Handbook of industrial robotics, volume 1”, Canada: John Wiley & Sons, 1999.
- [15] International federation of robotics (IFR), “World Robotics”, Switzerland: United Nation, 2005
- [16] Kabuka, R. M., Glaskowsky, N. P., Miranda, J., "Microcontroller-Based Architecture for Control of a Six Joints Robot Arm", IEEE Transactions on Industrial Electronics, Issue 2, Vol. 35, pp.217-221, 1988.

Chapitre III
Modélisation du bras
manipulateur

Le déplacement du système (chariot, pont et bras manipulateur) se fait par l'action du moteur en provoquant une vitesse qui sera constante pour chaque poids de la charge. Le mouvement de ces derniers sera rectiligne suivant les deux axes (OXZ) et (OYZ).

III.1 Modélisation du système de levage bras manipulateur

III.1.1 Représentation du bras manipulateur

La structure de notre bras est la Structure Ouverte Simple (S.O.S.) dont la quelle chaque corps a aux plus deux articulations. La St.O.S permet d'amener l'organe terminal dans une situation (position et orientation) donnée. Elle est la plus utilisé et a pour avantage essentiel d'augmenter la rigidité ainsi que la précision.

Le bras manipulateur à étudier est un robot articulé à 6 D.D.L. Six degrés de liberté sont suffisants pour positionner l'organe terminal dans une position et orientation complet dans l'espace. La cinématique du porteur est de type RRR et le poignet comporte trois rotations comme la Fig.III.1 l'indique [1].

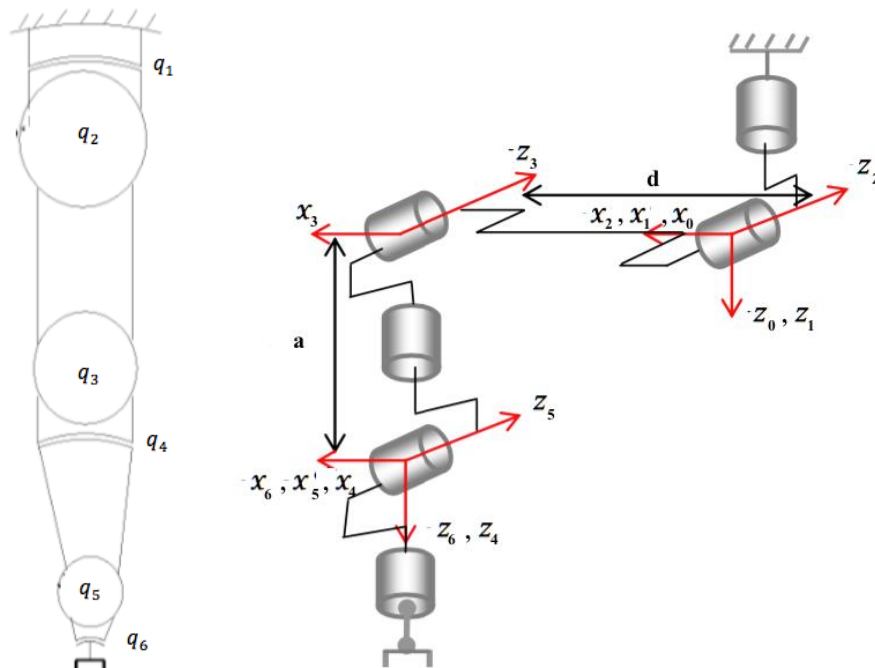


Fig.III.1 La cinématique du bras manipulateur (position et orientation).

Dans la pratique, le poignet de type rotule est très répandu. Le bras, obtenu en lui associant un porteur à 3 D.D.L., est la structure la plus classique à 6 D.D.L. Elle permet d'assurer un découplage entre la position et l'orientation de l'organe terminal :

- Le porteur a pour rôle de fixer la position du point d'intersection, noté P , des axes des 3 dernières articulations (centre du poignet) ; cette position (P) ne dépend que de la configuration des solides (corps) 1, 2 et 3 (i.e., du porteur),
- Le poignet est destiné à l'orientation de l'organe terminal (pince, outil) (Fig.III.2).

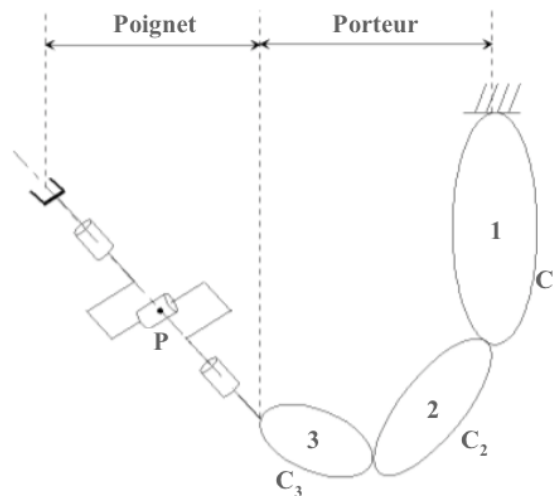


Fig.III.2 Structure du bras à poignet de type rotule.

Pour cela, on établit les deux matrices P et H qui représentent respectivement la position et l'orientation de la pince avec :

$$P = \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix} ; \quad H = [S \quad N \quad A] = \begin{bmatrix} S_x & N_x & A_x \\ S_y & N_y & A_y \\ S_z & N_z & A_z \end{bmatrix} \quad (\text{III. 1})$$

III.1.2 Détermination du repère R_j lié au cops C_j

On numérote les solides par ordre croissant en partant du socle. Ainsi le bras est composé de $n + 1$ corps, notés C_0, \dots, C_n et de n articulations ($n > 1$). Le corps C_0 désigne le socle (la base) du bras, le corps C_n portant l'organe terminal. Le repère R_j est lié au corps C_j du bras. La variable de l'articulation j , qui lie le corps C_j au corps C_{j-1} , est notée q_j [1].

Les repères entre les articulations $j - 1$ et j sont définis comme suit (Fig.III.3) :

- L'axe $\overrightarrow{o_j z_j}$ est porté par l'axe de rotation de l'articulation j .
- L'axe $\overrightarrow{o_j x_j}$ est porté par la perpendiculaire commune aux axes $\overrightarrow{o_j z_j}$ et $\overrightarrow{o_{j+1} z_{j+1}}$.

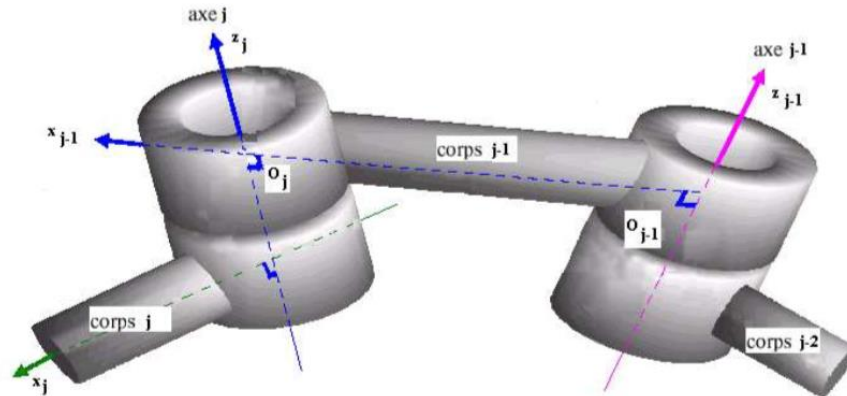


Fig.III.3 Repère des articulations $j-1$ et j .

La détermination du passage du repère R_j au R_{j-1} s'exprime en fonction des cinq paramètres suivants (Fig.III.4) [1] :

- σ_j : le paramètre qui détermine le type de la variable d'articulation q_j .
- α_j : l'angle de rotation entre les axes $\overrightarrow{z_{j-1}}$ et z_j autour de l'axe $\overrightarrow{x_{j-1}}$.
- d_j : la distance entre $\overrightarrow{z_{j-1}}$ et $\overrightarrow{z_j}$ mesurée le long de l'axe $\overrightarrow{x_{j-1}}$.
- θ_j : l'angle de rotation entre les axes $\overrightarrow{x_{j-1}}$ et $\overrightarrow{x_j}$ autour de l'axe $\overrightarrow{z_j}$.
- r_j : la distance entre $\overrightarrow{x_{j-1}}$ et $\overrightarrow{x_j}$ mesurée le long de l'axe $\overrightarrow{z_j}$.

Note 1 : La variable articulaire q_j , associée à l'articulation j , est soit θ_j , soit r_j , selon que cette articulation est respectivement de type rotoïde ou prismatique, ce qui se traduit par la relation :

$$q_j = (1 - \sigma_j)\theta_j + \sigma_j r_j \quad (\text{III.2})$$

Avec :
$$\begin{cases} \sigma_j = 0 & \text{si l'articulation } j \text{ est rotoïde} \\ \sigma_j = 1 & \text{si l'articulation } j \text{ est prismatique} \end{cases}$$

Autrement dit,

Si l'articulation est une rotation alors $\begin{cases} q_j = \theta_j \text{ est variable} \\ \alpha_j, d_j, r_j \text{ sont constants} \end{cases}$

Si l'articulation est une translation alors $\begin{cases} q_j = r_j \text{ est variable} \\ \alpha_j, d_j, \theta_j \text{ sont constants} \end{cases}$

Note 2 : Bien souvent, la rotation d'angle α_j est multiple de $\pi/2$, les deux axes de liaisons consécutifs d'un bras sont orthogonaux ou parallèles, il en résulte un angle α_j égale à 0° , $\pm 90^\circ$, 180° [1].

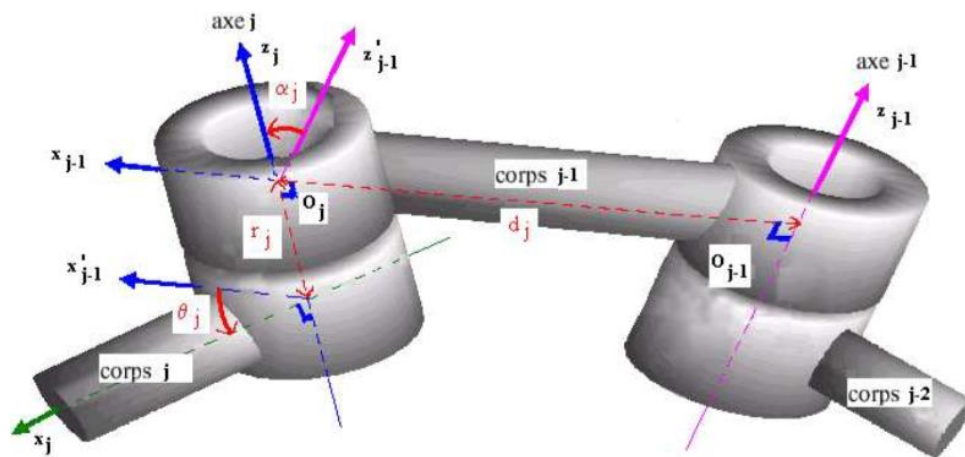


Fig.III. 4 Paramètres de passage du repère des articulations j-1 et j.

Les paramètres géométriques du bras manipulateur étudié sont représentés dans le tableau suivante :

J	σ_j	α_j	d_j	θ_j	r_j
1	0	0	0	θ_1	0
2	0	$\pi/2$	0	θ_2	0
3	0	$\pm \pi/2$	d	θ_3	0
4	0	$\pm \pi/2$	0	θ_4	a
5	0	$\pm \pi/2$	0	θ_5	0
6	0	$\pm \pi/2$	0	θ_6	0

Tableau III.1 Paramètres géométriques des trajectoires du bras manipulateur.

III.1.3 Modèle géométrique du bras manipulateur en chaîne simple

La conception et la commande des bras nécessitent le calcul de certains modèles mathématiques spécialement 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 bras). Parmi ces modèles :

- *Les modèles géométriques direct et inverse* qui expriment la situation de l'organe terminal en fonction de la configuration du mécanisme et inversement.

III.1.3.a Modèle géométrique direct du bras manipulateur (M.G.D)

On s'intéresse à la relation liant l'espace articulaire associé aux différentes liaisons du bras avec l'espace opérationnel dans lequel est définie la situation de l'organe terminal. Cette relation s'exprime à l'aide de l'équation suivante :

$$X = f(q) \quad (\text{III. 3})$$

Avec

- f : fonction vectorielle statique.
- X : le vecteur des coordonnées opérationnelles exprimées dans le repère de référence R_0 (la position (P)).
- q : les variables articulaires.

Soit La matrice $T_{0,n}$ représente la position et l'orientation, exprimées dans le repère de référence R_0 , de l'organe terminal du bras [1] :

$$T_{0,n} = T_{0,1}(q_1) \times T_{1,2}(q_2) \times \dots \times T_{n-1,n}(q_n) \quad (\text{III. 4})$$

Soit U_0 la situation du repère R_n (lié à l'organe terminal) décrit par [1] :

$$U_0 = \begin{bmatrix} s_x & n_x & a_x & p_x \\ s_y & n_y & a_y & p_y \\ s_z & n_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (\text{III. 5})$$

On a :

- Les trois (03) paramètres (indépendants) pour définir l'orientation du repère R_n par rapport au repère de base R_0 :

$$X_n = \begin{cases} \vec{x}_n = s_x \vec{x}_0 + s_y \vec{y}_0 + s_z \vec{z}_0 \\ \vec{y}_n = n_x \vec{x}_0 + n_y \vec{y}_0 + n_z \vec{z}_0 \\ \vec{z}_n = a_x \vec{x}_0 + a_y \vec{y}_0 + a_z \vec{z}_0 \end{cases} \quad (\text{III. 6})$$

- Les trois (03) paramètres (indépendants) pour définir la position du point P .

$$\vec{o_0 o_n} = p_x \vec{x}_0 + p_y \vec{y}_0 + p_z \vec{z}_0 \quad (\text{III. 7})$$

De cela, le modèle géométrique direct du bras est donné par les équations suivantes [2] :

$$S = \begin{cases} s_x = C_1(C_{23}(C_4 C_5 C_6 - S_4 S_6) - S_{23} S_5 C_6) - S_1(S_4 C_5 C_6 + C_4 S_6) \\ s_y = S_1(C_{23}(C_4 C_5 C_6 - S_4 S_6) - S_{23} S_5 C_6) + C_1(S_4 C_5 C_6 + C_4 S_6) \\ s_z = S_{23}(C_4 C_5 C_6 - S_4 S_6) + C_{23} S_5 C_6 \end{cases} \quad (\text{III. 8})$$

$$N = \begin{cases} n_x = C_1(-C_{23}(C_4 C_5 C_6 + S_4 S_6) + S_{23} S_5 C_6) + S_1(S_4 C_5 C_6 - C_4 S_6) \\ n_y = S_1(-C_{23}(C_4 C_5 C_6 - S_4 S_6) + S_{23} S_5 C_6) - C_1(S_4 C_5 C_6 - C_4 S_6) \\ n_z = -S_{23}(C_4 C_5 C_6 + S_4 S_6) - C_{23} S_5 S_6 \end{cases} \quad (\text{III. 9})$$

$$A = \begin{cases} a_x = -C_1(C_{23} C_4 S_5 + S_{23} C_5) + S_1 S_4 S_5 \\ a_y = -S_1(C_{23} C_4 S_5 + S_{23} C_5) - C_1 S_4 S_5 \\ a_z = -S_{23} C_4 S_5 + C_{23} C_5 \end{cases} \quad (\text{III. 10})$$

$$P = \begin{cases} p_x = -C_1(a S_{23} + d C_2) \\ p_y = -S_1(a S_{23} + d C_2) \\ p_z = a C_{23} + d S_2 \end{cases} \quad (\text{III. 11})$$

Avec

$$\begin{cases} C_1 = \cos(\theta_1) \\ S_1 = \sin(\theta_1) \end{cases} \quad (\text{III. 12})$$

$$\begin{cases} C_2 = \cos(\theta_2) = \frac{yz - \varepsilon x \sqrt{x^2 + y^2 - z^2}}{x^2 + y^2} \\ S_2 = \sin(\theta_2) = \frac{xz - \varepsilon x \sqrt{x^2 + y^2 - z^2}}{x^2 + y^2} \end{cases} \quad (\text{III. 13})$$

$$\text{Avec } x = -2dp_y; \quad y = -2dB_1; \quad B_1 = p_x C_1 + p_z S_1; \quad z = a^2 - d^2 - p_z^2 - B_1^2$$

$$\begin{cases} C_{23} = \cos(\theta_2 + \theta_3) \\ S_{23} = \sin(\theta_2 + \theta_3) \end{cases} \quad (\text{III. 14})$$

$$\begin{cases} C_4 = \cos(\theta_4) \\ S_4 = \sin(\theta_4) \end{cases} \quad (\text{III. 15})$$

$$\begin{cases} C_5 = -S_{23}(a_x C_1 + a_y C_2) + a_z C_{23} \\ S_5 = -C_4[C_{23}(a_x C_1 + a_y S_1) + a_z S_{23}] + S_4(a_x S_1 - a_y C_1) \end{cases} \quad (\text{III. 16})$$

$$- \begin{cases} C_6 = -S_4 [C_{23}(n_x C_1 + n_y S_1) + n_z S_{23}] - C_4 (n_x S_1 - n_y C_1) \\ S_6 = -S_4 [C_{23}(s_x C_1 + s_y S_1) + s_z S_{23}] - C_4 (s_x S_1 - s_y C_1) \end{cases} \quad (\text{III. 17})$$

III.1.3.b Modèle géométrique inverse du bras manipulateur (M.G.I)

Le modèle géométrique inverse est le problème inverse qui permet de connaître les variables articulaires en fonction de la situation de l'organe terminal, ce qui peut se représenter par la relation [1] :

$$q = g(X) \quad \text{Avec} \quad g = f^{-1} \quad (\text{III. 18})$$

Pour traiter le modèle géométrique inverse de notre bras nous avons utilisé la méthode de Paul car elle est permet de traiter séparément chaque cas particulier et convient pour la plupart des bras industriels.

Méthode de Paul : s'agit de déterminer les coordonnées articulaires q_j permettant d'obtenir une situation désirée pour l'organe terminal et spécifiée par les coordonnées opérationnelles X (détermination de q_1 , puis q_2 et ainsi de suite jusqu'à q_n) [1].

La position du point P (centre du poignet) est en fonction des variables articulaires q_1, q_2, q_3 . Aussi, ce type de structure du bras permet de décomposer le problème du calcul des 6 variables articulaires en deux :

- *Problème de position* : en fonction de q_1, q_2, q_3 il permet de les déterminer.
- *Problème d'orientation* : en fonction de q_4, q_5, q_6 , il permet de les déterminer.

Après les calculs de la methode de Paul et les equations du M.G.D, on obtient les solutions suivantes [1-2] :

➤ Equation du position (méthode de Paul [1-2])

Détermination de θ_1 :

$$\begin{cases} C_1 p_x + S_1 p_y = d S_{23} + a C_2 \\ -S_1 p_x + C_1 p_y = 0 \\ C_4 = d C_{23} - a \cdot S_2 \end{cases} \quad (\text{III. 19})$$

De la deuxième ligne d'équation :

$$\frac{S_1}{C_1} = \frac{p_y}{p_x} \Rightarrow \frac{\sin(\theta_1)}{\cos(\theta_1)} = \frac{p_y}{p_x} \Rightarrow \theta_1 = \arctg\left(\frac{p_y}{p_x}\right) \quad (\text{III. 20})$$

Détermination de θ_2 :

$$z = xS_2 + yC_2 \quad (\text{III. 21})$$

De l'équation (III. 11), on remplace dans (III. 19) :

$$\frac{S_2}{C_2} = \frac{\sin(\theta_2)}{\cos(\theta_2)} \Rightarrow \theta_2 = \arctg\left(\frac{S_2}{C_2}\right) \quad (\text{III. 22})$$

Détermination de θ_3 : Soit le système d'équations qui donne la relation entre θ_2 et θ_3 suivant :

$$\begin{cases} d \sin(\theta_3) = C_2 B_1 - S_2 p_z - a \\ d \cos(\theta_3) = S_2 B_1 + C_2 p_z \end{cases} \quad (\text{III. 23})$$

On obtient,

$$\frac{\sin(\theta_3)}{\cos(\theta_3)} = \frac{C_2 B_1 - S_2 p_z - a}{S_2 B_1 + C_2 p_z} \Rightarrow \theta_3 = \arctg\left(\frac{C_2 B_1 - S_2 p_z - a}{S_2 B_1 + C_2 p_z}\right) \quad (\text{III. 24})$$

➤ **Equation d'orientation (méthode de Paul [1-2])**

En connaissant les trois paramètres de la position du point P ($\theta_1, \theta_2, \theta_3$), la détermination des autres paramètres restant se fait comme suit :

Détermination de θ_4

$$-S_4[-C_{23}(a_x C_1 + a_y C_1) - a_z S_{23}] + C_4[a_x S_1 - a_y C_1] = 0 \quad (\text{III. 25})$$

De l'équation (III. 25) :

$$\frac{S_4}{C_4} = \frac{\sin(\theta_4)}{\cos(\theta_4)} = \frac{a_x S_1 - a_y C_1}{-C_{23}(a_x C_1 + a_y C_1) - a_z S_{23}} \Rightarrow \theta_4 = \arctg\left(\frac{a_x S_1 - a_y C_1}{-C_{23}(a_x C_1 + a_y C_1) - a_z S_{23}}\right) \quad (\text{III. 26})$$

Détermination de θ_5

$$\begin{cases} \sin(\theta_5) = S_5 = C_4[-C_{23}(a_x C_1 + a_y C_1) - a_z S_{23}] + S_4[a_x S_1 - a_y C_1] \\ \cos(\theta_5) = C_5 = -S_{23}(a_x C_1 + a_y C_2) + a_z C_{23} \end{cases} \quad (\text{III. 27})$$

De (III. 27), on trouve :

$$\frac{S_5}{C_5} = \frac{\sin(\theta_5)}{\cos(\theta_5)} \Rightarrow \theta_5 = \arctg\left(\frac{C_4[-C_{23}(a_x C_1 + a_y C_1) - a_z S_{23}] + S_4[a_x S_1 - a_y C_1]}{-S_{23}(a_x C_1 + a_y C_2) + a_z C_{23}}\right) \quad (\text{III. 28})$$

Détermination de θ_6

$$\begin{cases} \sin(\theta_6) = S_6 = -S_4[C_{23}(s_x C_1 + s_y S_1) + s_z S_{23}] - C_4(s_x S_1 - s_y C_1) \\ \cos(\theta_6) = C_6 = -S_4[C_{23}(n_x C_1 + n_y S_1) + n_z S_{23}] - C_4(n_x S_1 - n_y C_1) \end{cases} \quad (\text{III. 29})$$

On obtient de (III. 28): l'équation (III. 29)

$$\frac{S_6}{C_6} = \frac{\sin(\theta_6)}{\cos(\theta_6)} \Rightarrow \theta_6 = \arctg\left(\frac{-S_4[C_{23}(s_x C_1 + s_y S_1) + s_z S_{23}] - C_4(s_x S_1 - s_y C_1)}{-S_4[C_{23}(n_x C_1 + n_y S_1) + n_z S_{23}] - C_4(n_x S_1 - n_y C_1)}\right) \quad (\text{III. 30})$$

III.2 Simulation du bras manipulateur sous MATLAB

La tâche de déplacement du bras est spécifiée en définissant un chemin que le bras doit suivre.

Un chemin est une séquence de points définis soit dans l'espace des tâches opérationnel (position : afin de situer l'organe terminal), soit dans l'espace des configurations du bras (articulation : afin d'indiquer les valeurs des paramètres de liaison).

Dans notre travail, la trajectoire étant fixée à tout instant dans l'espace opérationnel, il est préférable de raisonner dans cet espace. La loi de commande engendrée doit ensuite être transformée en consignes articulaires par le changeur de coordonnées.

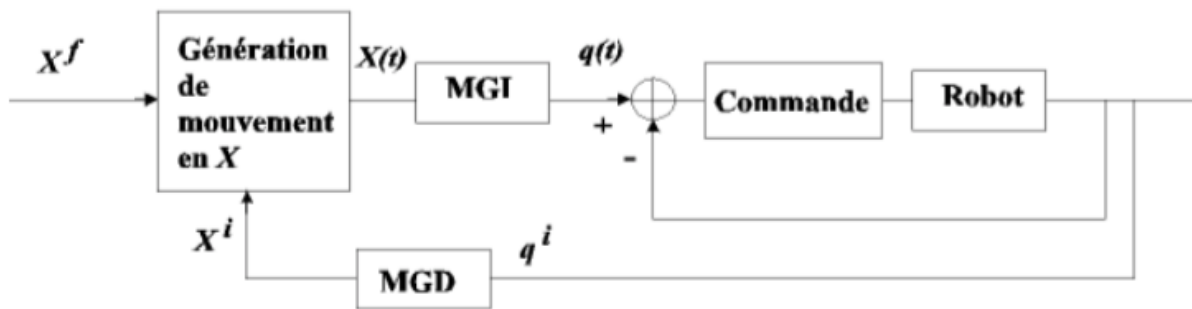


Fig.III.5 Génération de mouvement dans l'espace opérationnel.

Le bras manipulateur est constitué de segments reliés entre eux par des articulations. Chaque articulation constitue un axe autour duquel deux segments pivotent. Les mouvements des articulations du bras sont générés par des servomoteurs (moteurs asservis) (Fig.III.6).

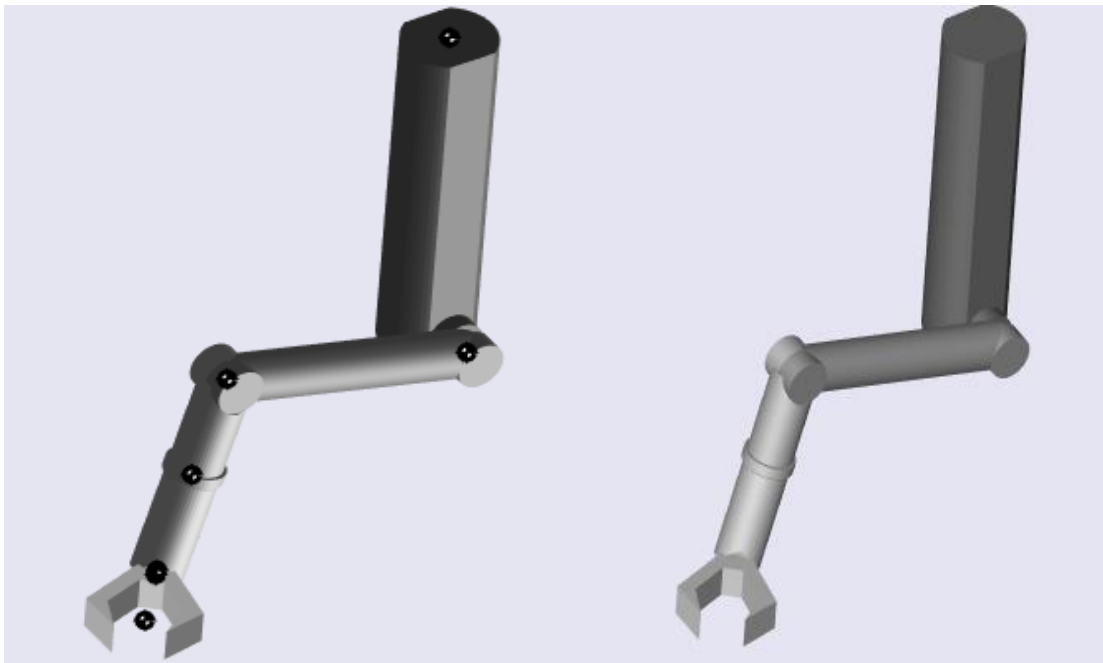


Fig.III.6 Bras manipulateur dans MATLAB.

Note : les dimensions du bras étudié dans MATLAB sont choisis pour montrer les articulations d'une manière claire, ils ne sont pas compatibles avec le prototype.

III.2.1 Présentation du travail

Tout d'abord, nous avons défini pour chaque articulation l'axe de rotation autorisé, la position par rapport au repère de base « CG » et le repère lié au corps « CS » que la Fig.III.7 le montre :

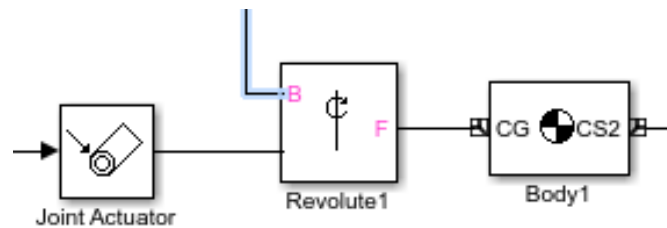


Fig.III.7 Les coordonnées des articulations.

Pour avoir la position exacte de l'organe terminal et l'orientation des articulations, l'espace « Simulink » nous a permis d'utiliser un capteur spécial pour les blocks « Body » qui signifie « Corps », la Figure III.8 le montre:

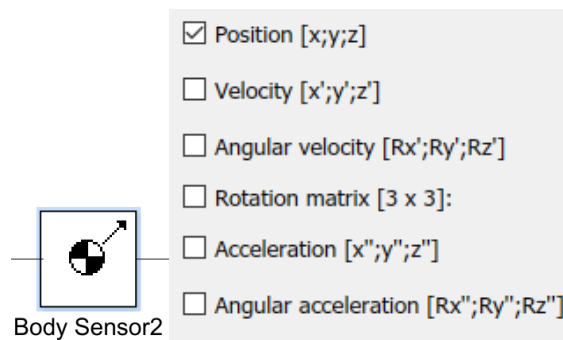


Fig.III.8 L'utilisation du capteur « Body Sensor ».

La représentation du résultat du capteur pour notre système est comme suite (Fig.III.9) :

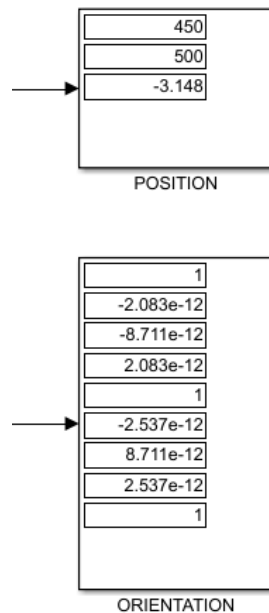


Fig.III.9 Représentation du résultat du capteur « Body Sensor ».

Tous les corps du bras C_j ont un repère relatif R_j , ils sont reliés entre eux par une liaison représentée dans la Fig.III.10 qui donne les coordonnées du repère R_j d'un corps à un autre.

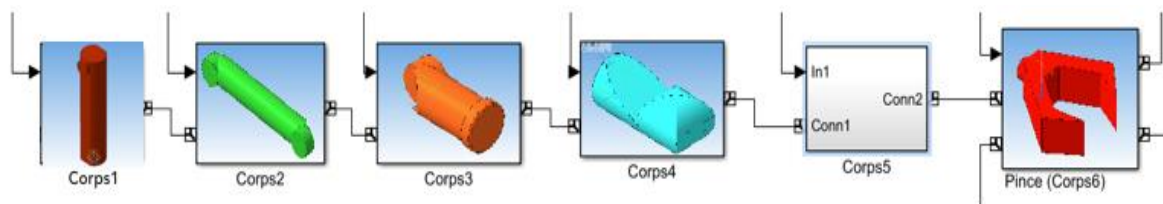


Fig.III.10 Représentation des corps du bras dans « Simulink ».

L'entrée de chaque articulation est calculée par la méthode de Paul dans le code MATLAB, le principe consiste à donner une trajectoire à l'organe terminal pour avoir les valeurs de chaque angle d'articulation qui ramène le système à l'endroit souhaité d'une manière à ce que la position de la pince sera précise (erreur de position tends vers « 0 »).

III.2.2 Trajectoire de l'organe terminal

III.2.2.a Trajectoire circulaire

Dans cette étape, nous avons donné au bras manipulateur une trajectoire circulaire autour d'un point de l'espace ($OXYZ$) égale à $p1 = (450, 750, 0)$ avec une position initiale $P = (450, 500, 0)$, un rayon de $c1 = 250$ (suivant la direction de l'axe OZ , par exemple) et un facteur de rotation $c2 = 2$ (si $c2 = 1$ le bras fait un demi cercle) comme le montre le code MATLAB suivant (Fig.III.11) :

```
% TRAJECTOIRE "cercle" :  
c1=250;  
c2=2;  
t=linspace(0,pi,not);  
tmpx=zeros(size(t))+p1(1,1);  
zkor=c1*sin(c2*t-pi)+p1(3,1);  
ykor=c1*cos(c2*t-pi)+p1(2,1);
```

Fig.III.11 Le programme de la trajectoire circulaire.

La représentation de la trajectoire dans l'espace ($OXYZ$) est comme suite Fig.III.12 :

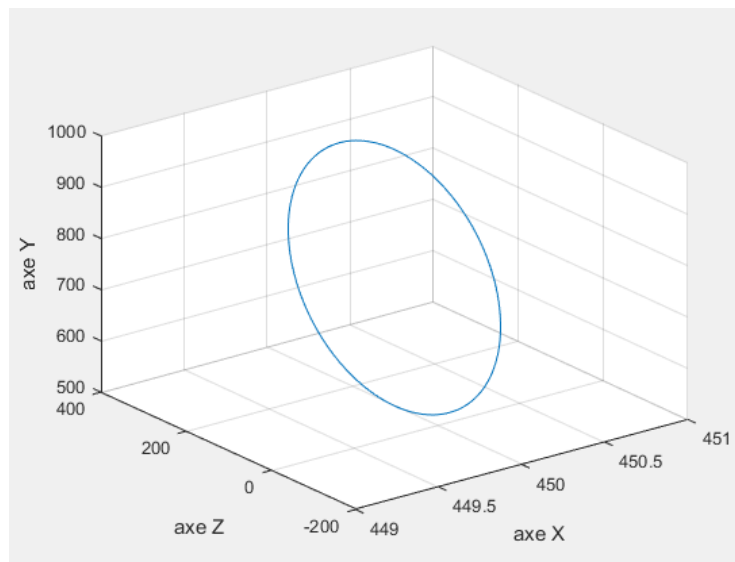


Fig.III.12 La trajectoire dans l'espace d'étude ($OXYZ$).

Le bras manipulateur va suivre cette trajectoire souhaitée à l'organe terminal avec une très grande précision par rapport à la position (une faible erreur). Après la simulation du mouvement du système la dynamique est représentée par la Fig.III.13 :

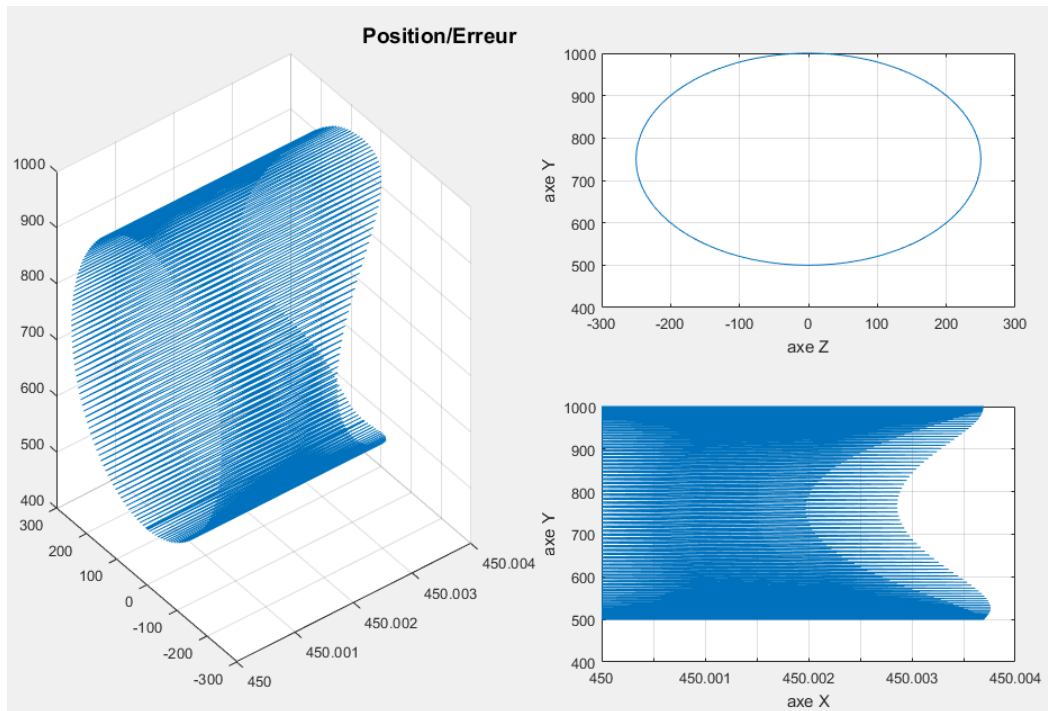


Fig.III.13 Simulation du mouvement du bras dans l'espace (OXYZ).

III.2.2.b Trajectoire circulaire avec déplacement

La position initiale de l'organe terminal dans l'espace (OXYZ) est égale à $P = (850, 500, 100)$, la trajectoire étudiée consiste à déplacer le système du repo au point $p1 = (500, 900, 150)$ pour faire un mouvement circulaire avec un rayon de 150 (suivant l'axe OZ), voir la Fig.III.14.

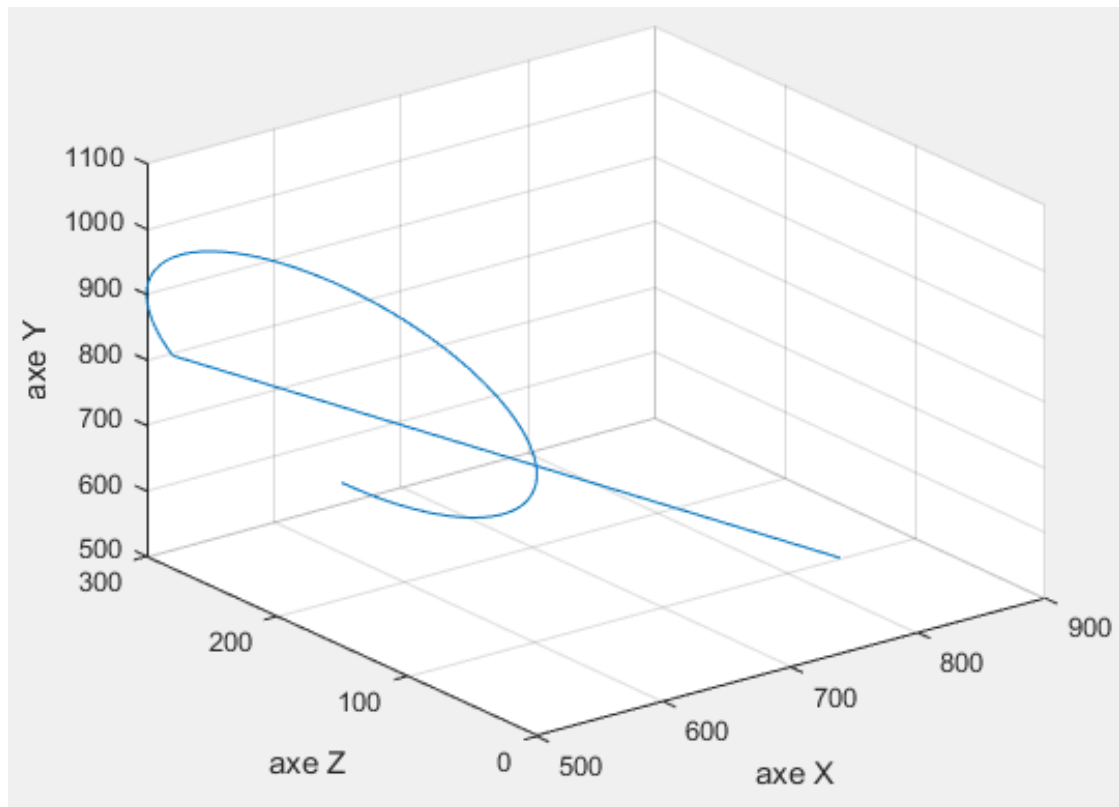


Fig.III.14 La trajectoire circulaire avec déplacement.

La simulation de la dynamique du mouvement montre la position de l'organe terminal et l'erreur de position pour chaque axe de l'espace (Fig.III.15).

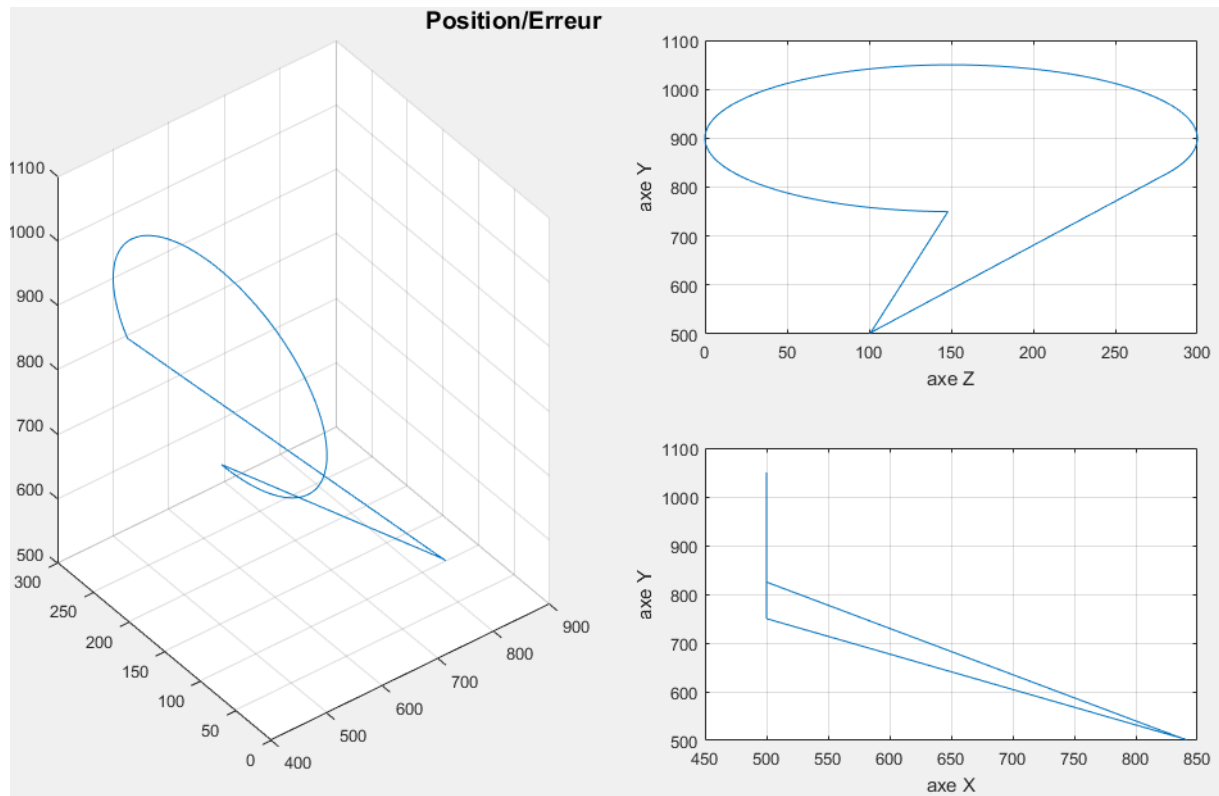


Fig.III.15 Simulation de la dynamique du bras dans l'espace (OXYZ).

Conclusion

Dans ce chapitre, nous avons utilisé pour déterminer les consignes des articulations la méthode de Paul en se basant essentiellement sur la géométrie du bras manipulateur directe et inverse. Cela nous a permis d'étudier le mouvement de ce mécanisme avec une grande précision.

Références bibliographiques

[1] B. Jean-Louis : Robotique, *Cours Robot, ISTIA Université Angers*, 85 pages, 2009.

[2] G. Hatem, M. Manel et K. Okba, Commande d'un bras manipulateur 6 D.D.L avec évitement d'obstacle par la logique floue. *Courrier du Savoir*, **10**, 15-22, 2010.

Chapitre IV
Le bras manipulateur en
simulation

Partie IV.I
Hardware

Dans cette première partie, nous allons entamer la partie hardware qui concerne la simulation du prototype de notre pont roulant en décrivant ses différents composants avec lesquels nous avons travaillé et en présentant la carte électronique de commande avec une description détaillée des différents éléments qui la constituent et leurs schémas de brochages.

IV.I.1 L'environnement de simulation hardware

IV.I.1.1 Présentation générale

Proteus Professional est une suite logicielle destinée à l'électronique. Développé par la société Labcenter Electronics, les logiciels incluent dans Proteus Professional permettent la CAO (Construction Assistée par Ordinateur) dans le domaine électronique. Deux logiciels principaux composent cette suite logicielle : (ISIS, ARES, PROSPICE) et VSM [1].

Cette suite logicielle est très connue dans le domaine de l'électronique. De nombreuses entreprises et organismes de formation (incluant lycée et université) utilisent cette suite logicielle. Outre la popularité de l'outil, Proteus Professional possède d'autres avantages

- Pack contenant des logiciels facile et rapide à comprendre et utiliser
- Le support technique est performant
- L'outil de création de prototype virtuel permet de réduire les coûts matériel et logiciel lors de la conception d'un projet

Le logiciel ISIS de **Proteus Professional** est principalement connu pour éditer des schémas électriques. Par ailleurs, le logiciel permet également de simuler ces schémas ce qui permet de déceler certaines erreurs dès l'étape de conception (Fig.IV.I.1). Indirectement, les circuits électriques conçus grâce à ce logiciel peuvent être utilisés dans des documentations car le logiciel permet de contrôler la majorité de l'aspect graphique des circuits [3].

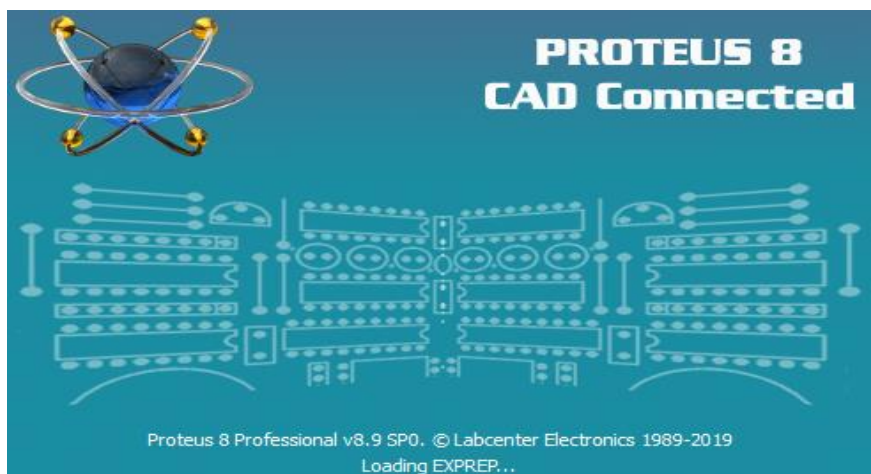


Fig.IV.I.1 L'environnement de simulation ISIS Proteus.

IV.I.2 Description de la carte électronique de commande

Le système de commande que nous avons travaillé avec est une carte électronique à microcontrôleur qui assure le control d'un pont roulant bipoutre en mode charge avec les bras robotique articulé. Ce système est composé de deux parties essentielles ; la première partie se compose d'un microcontrôleur qui est le responsable de la partie commande, communication et acquisition des données et le circuit électronique contenant la partie puissance de la carte qui sert à convertir les signaux de commande à des signaux de puissance pour piloter l'ensemble des moteurs pas a pas et les servos moteurs. La deuxième partie concerne la boîte de commande par radio fréquence.

IV.I.3 Schéma du circuit électrique

Notre schéma est composé de deux partie, la première partie représente le microcontrôleur qui contrôle le bras robotique et le pont et aussi représente le circuit de puissance, la deuxième partie représente la commande par Radio Fréquence qu'on va les détailler par la suite.

IV.I.3.1 Circuit électrique de la commande et de puissance

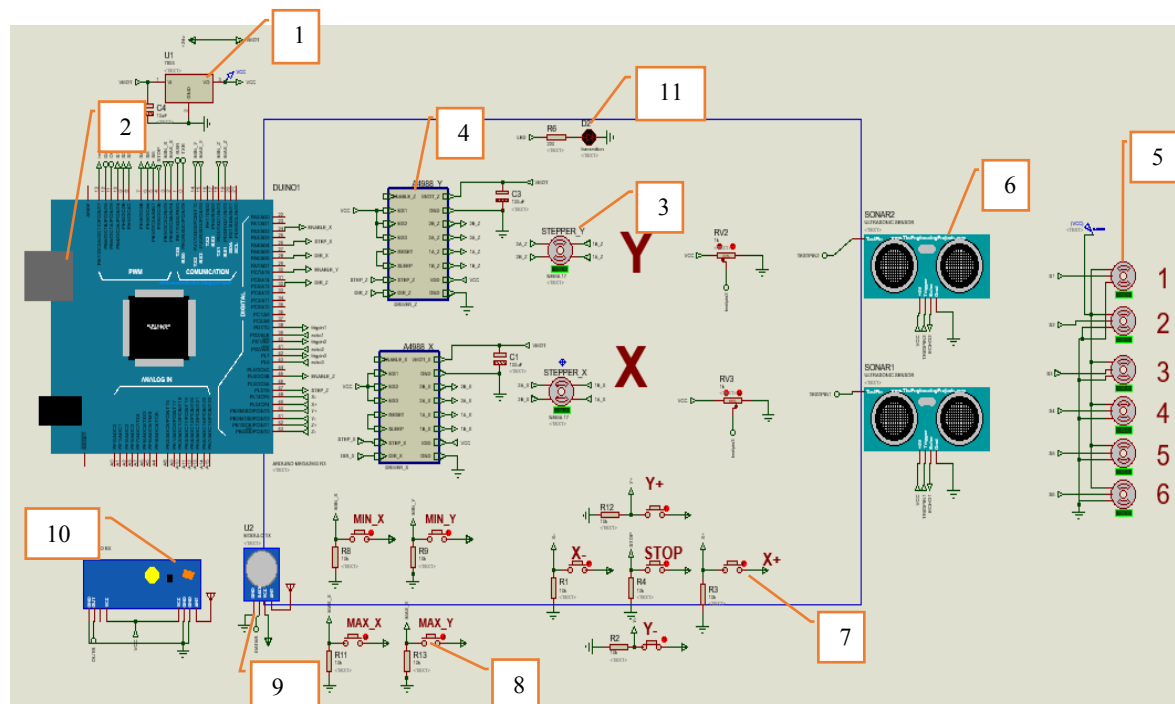


Fig.IV.I.2 Circuit électrique de commande et puissance.

Ce circuit est composé de :

Numéro	Signification
1	Boîte d'alimentation 24v
2	Carte Arduino méga 2560.
3	Moteur pas à pas de type NEMA17
4	Driver du moteur pas à pas polulu a4988
5	Servo moteur
6	Capteur de distance ultrason de type HC-SR-04
7	Bouton poussoir
8	Capteur de fin de course (représenté par des boutons poussoir)
9	Émetteur radio fréquence 433Mhz.
10	Récepteur radio fréquence 433Mhz.
11	LED

Tableau IV.I.1 Signification du circuit de commande et puissance.

IV.I.3.2 Circuit électrique de la boîte de commande par RF

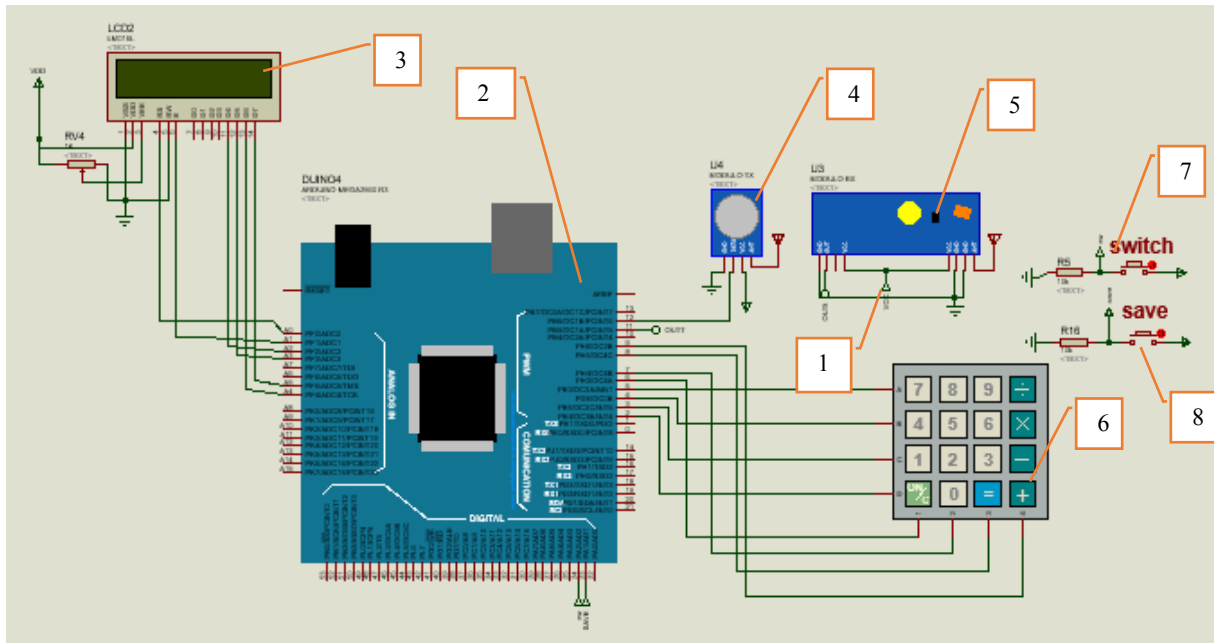


Fig.IV.I.3 Circuit de la commande par radiofréquence.

Ce circuit est composé de :

Numéro	Signification
1	Boite d'alimentation 5v
2	Carte Arduino méga 2560.
3	LCD de type LMO16L (2*16).
4	Émetteur radio fréquence 433Mhz
5	Récepteur radio fréquence 433Mhz.
6	Keypad - smallclac
7	Bouton de switch.
8	Bouton d'enregistrement.

Tableau IV.I.2 Signification du circuit de commande et puissance.

IV.I.4 Les composants électroniques

IV.I.4.1 Microcontrôleur

Un microcontrôleur (en notation abrégée μc , **MCU** en anglais) est un circuit intégré qui rassemble les éléments essentiels d'un ordinateur : processeur, mémoires (RAM et ROM), unités périphériques et interfaces d'entrées-sorties. Les

microcontrôleurs se caractérisent par un plus haut degré d'intégration, une plus faible consommation électrique, une vitesse de fonctionnement plus faible (de quelques mégahertz jusqu'à plus d'un gigahertz) et un coût réduit par rapport aux microprocesseurs polyvalents utilisés dans les ordinateurs personnels.

Par rapport à des systèmes électroniques à base de microprocesseurs et autres composants séparés, les microcontrôleurs permettent de diminuer la taille, la consommation électrique et le coût des produits. Ils ont ainsi permis de démocratiser l'utilisation de l'informatique dans un grand nombre de produits et de procédés.

Les microcontrôleurs sont fréquemment utilisés dans les systèmes embarqués, comme les contrôleurs des moteurs automobiles, les télécommandes, les appareils de bureau, l'électroménager, les jouets, la téléphonie mobile [2].

Notre choix s'est porté sur la carte de développement Arduino Mega qui est à base de microcontrôleur ATMEL Atmega2560 et cela pour des différentes raisons :

- La disponibilité.
- Connait une large utilisation actuellement.
- Peu coûteux.
- Performant.
- Outil de développement open source.

Arduino Mega 2560 (Fig.IV.I.4) dispose de 54 broches d'entrée/sortie numériques (dont 15 peuvent être utilisées comme sorties PWM), 16 entrées analogiques, 4 UART (ports série matériels), un oscillateur à cristaux de 16 Mhz, une connexion USB, une prise d'alimentation, un en-tête ICSP et un bouton de réinitialisation.

On trouve plusieurs IDE qui sont dédiés à la programmation de ce Microcontrôleur comme : ARDUINO IDE, AVR studio et ATMEL Studio.

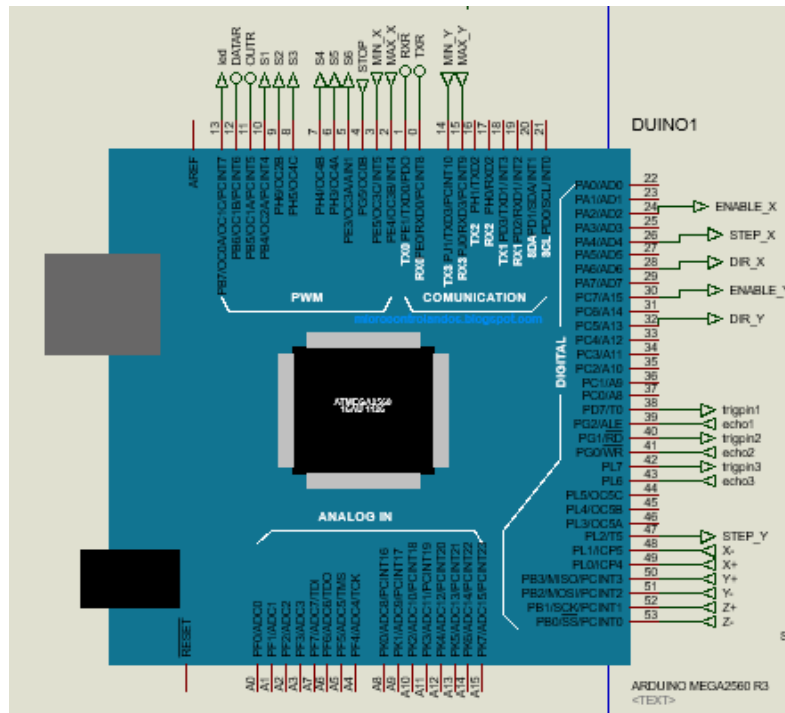


Fig.IV.I.4 Carte Arduino méga 2560.

Le microcontrôleur Atmega2560 est un microcontrôleur 8 bits de la famille AVR dont la programmation est faite avec le langage C [4].

Les principales caractéristiques de microcontrôleur ATMEL Atmega2560 sont les suivantes :

- **Mémoire Flash** : 256 Ko.
- **Mémoire SRAM** : 8 Ko.
- **EEPROM** : 4 Ko.
- **Fréquence maximale** : 16 MHz.
- **Nombre des E/S** : 54 pins entrées/sorties programmables.
- **Timers /Counters** : deux timers de 08 bits : timer0 et timer2 et quatre timers de 16 bits : timer1, timer3, timer4 et timer5.
- **Watchdog timer programmable.**
- **PWM** : Quatre canaux de PWM de 08 bits.
- Six/onze chaines de PWM avec une résolution programmable de 02 à 16 bits.
- **UART** : Deux/quatre sorties UART programmables.
- **Master/slave SPI interface.**
- **Interface de communication I2C.**
- **ADC** : 16 entrées analogiques avec une résolution de 10 bits.

- **Six sleep mode.**
- **Interruption** : il y'en a 57 sources d'interruption possibles qu'on les résume comme suit :
 - Interruptions liées au changement d'état des pins.
 - Interruptions liées aux timers, UsART, SPI, I2C.
 - Interruption liée au Watchdog.
 - Interruptions liées aux comparateurs analogiques.

Dans ce travail on a utilisé deux carte arduino méga 2560 une pour la partie commande des moteurs et puissance et une autre pour la partie commande par radio fréquence.

IV.I.4.2 Le circuit de puissance A4988

L'A4988 est un circuit intégré dédié à la commande et le pilotage des moteurs pas à pas bipolaires comme dans notre cas le Nema 17. Ce driver permet de faire fonctionner le moteur dans cinq modes de résolutions différents ainsi, il peut délivrer un courant jusque' à 2A par phase.

IV.I.4.2.1 Caractéristiques et détails techniques du polulu A4988

Voici quelques caractéristiques essentielles de ce driver de moteur pas à pas [5,6] :

- Tension de fonctionnement minimum : 8 V.
- Tension maximale de fonctionnement : 35 V.
- Courant par phase (en continu) : 1,2 A.
- Courant maximum par phase : 2.3 A.
- Logique de commande :
 - Tension minimale : 3 V.
 - Tension maximale : 5.5 V.
- Interface de contrôle de pas et de direction simple et facile.
- 05 résolutions différentes de pas :
 - Full-step : pas complet.
 - Halfstep : 1/2 pas.
 - Quarter-step : 1/4 pas.
 - Eighth-step : 1/8ième de pas.
 - Sixteenth-step : 1/16 ième de pas.

- Contrôle en courant ajustable : un potentiomètre permet de fixer le courant maximum. Cela permet d'utiliser une tension supérieure à celle recommandée par le moteur (puisque le courant est limité) et permet ainsi d'atteindre un débit de pas plus élevé (et donc une vitesse plus élevée).
- Une commande de hachage intelligente qui sélectionne automatiquement le bon mode de décroissance de courant (fastdecay = baisse/chute rapide du courant OU slow decay = chute lente).
- Protection (arrêt) en cas de surchauffe, sous tension, surcharge/sur-courant/court-circuit. Utile en cas de surcharge/blocage moteur (car le courant va monter en flèche, ce qui doit provoquer la mise en protection).

IV.I.4.2.2 Brochage et description des pins de driver A4988

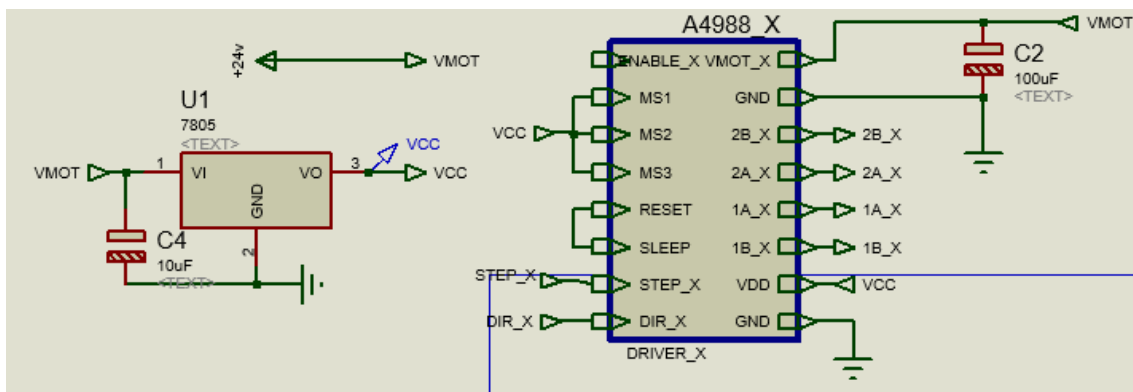


Fig.IV.I.5 Pololu 4988

La bibliothèque de l'environnement ISIS de Proteus ne contient pas un driver 4988 mais on peut le simuler en utilisant les deux driver L297 et L298 pour contrôler les moteurs pas à pas (Fig.IV.I.6).

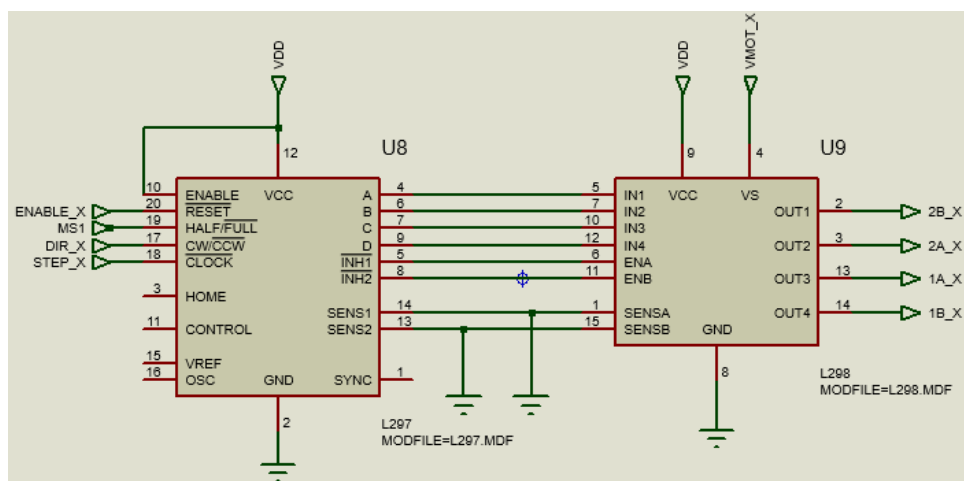


Fig.IV.I.6 Les drivers L297 ET L298.

- **ENABLE** : Logique Inversée, ce pin permet d'activer ou désactivé le moteur. Quand il est mis dans un état haut (5V) le moteur est désactivé et quand il est mis en état bas (0V) le moteur est activé.

- **MS1, MS2, MS3** : Ces pins permettent de sélectionner la configuration de mode de résolution de moteur pas à pas. Ces broches disposent de résistances Pull-Down ramenant le potentiel à 0V lorsque rien n'est connecté sur ces broches (ils sont connectés au GND par défaut).

- **RESET** : Logique inversée. Permet de faire une réinitialisation du module. Généralement connecté sur la broche "SLEEP".

- **SLEEP** : Logique inversée. Généralement connecté sur la broche "Reset" du module.

- **STEP** : En l'envoyant un signal d'horloge (haut puis bas) le moteur avancera d'un pas.

- **DIR** : Il permet de sélectionner la direction de rotation du moteur soit dans le sens horaire (CW) ou le sens antihoraire (CCW) et cela selon l'état de pin.

- **VMOT** : Tension d'alimentation du moteur. Habituellement 12V pour les moteurs pas à pas. Tension entre 8 et 12v.

- **GND** : C'est la masse (GROUND). Habituellement on mit toutes les masses en commun.

- **1B 1A** : Première bobine du moteur pas à pas bipolaire.

- **2A 2B** : Deuxième bobine du moteur pas à pas bipolaire.

- **VDD** : Alimentation de la logique de commande entre 3 et 5.5v. Habituellement 5V. [5,6]

Dans ce travail nous avons deux moteurs pas à pas donc nous avons besoin de deux drivers de type 4988 (Fig.IV.I.7).

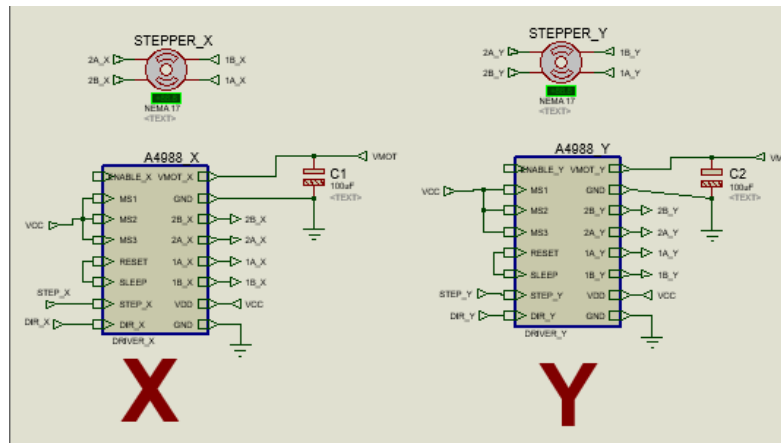


Fig.IV.I.7 Moteurs pas à pas avec leurs drivers.

IV.I.4.2.3 Ajustement de Driver A4988

IV.I.4.2.3.a Ajustement des entrées du contrôle

Chaque impulsion sur l'entrée STEP correspond à un micro pas du moteur pas-à-pas ou un pas complet dans la direction sélectionnée sur la broche DIR. Les broches STEP et DIR ne disposent pas de résistances pull-down ou pull-up interne. Donc le contrôle de deux états Haut et Bas de ces entrées est fait directement par le microcontrôleur. La tension sur ces broches ne doit jamais être flottante.

Le circuit dispose de trois entrées différentes pour contrôler son étage de puissance : RST (Reset), SLP (SLEEP) et EN (ENABLE), On peut connecter la broche RST avec la broche SLP pour la ramener à l'état haut et donc activer la carte [5].

IV.I.4.2.3.b Configuration de mode de résolution

Les moteurs pas-à-pas disposent de leur propre spécification physique de "pas" connu comme un "pas complet". Un pilot Micro-pas tel que l'A4988 permet d'obtenir une plus grande résolution en autorisant des positions intermédiaires dans un pas. Cela est rendu possible en modulant intelligemment la quantité de courant dans les bobines du moteur pas-à-pas. Par exemple, piloter un moteur en mode "1/4 de pas" permet d'obtenir 800 micro-pas sur un moteur prévu pour 200 pas par révolution et cela en utilisant 4 niveaux de courants différents pour chacun des Microsteps. La configuration de mode de résolution se fait à l'aide des broches MS1, MS2 et MS3.

- MS1 et MS3 disposent d'une résistance pull-down interne de 100kΩ. Une résistance pull-down signifie que si on ne place pas ces broches au niveau logique haut

VDD (la tension choisie pour la logique de commande) alors elles seront automatiquement ramenées au niveau logique bas.

- MS2 dispose d'une résistance pull-down de 50kΩ.

Si les broches MS1, MS2 et MS3 sont laissées flottantes, l'A4988 fonctionnera en mode "pas complet". Pour que le Microstepping fonctionne correctement, il faut que le courant soit limité d'une façon à ne pas activer la protection en sur-courant. Sinon, les niveaux des courants intermédiaires ne seront pas correctement maintenus et le moteur pourrait sauter des microsteps [6].

MS1	MS2	MS3	Mode de résolution
LOW	LOW	LOW	Pas complet.
HIGH	LOW	LOW	½ pas.
LOW	HIGH	LOW	¼ pas.
HIGH	HIGH	LOW	1/8 pas.
HIGH	HIGH	HIGH	1/16 pas.

Tableau IV.I.3 La configuration de mode de résolution.

IV.I.4.3 Les moteurs pas à pas

Le moteur pas à pas permet de transformer une impulsion électrique en un mouvement angulaire. Ce type de moteur est très courant dans tous les dispositifs où l'on souhaite faire du contrôle de vitesse ou de position, comme par exemple les imprimantes.

Le stator est constitué de bobines qui sont alimentées, à tour de rôle, en courant continu par une carte électronique. Le rotor est un croisillon, en métal ferreux ne conservant pas le magnétisme. Si on compte électroniquement les impulsions envoyées aux bobines on sait, connaissant le pas, le nombre de rotations que le rotor a effectuées.

Il existe trois types de moteurs pas à pas : les moteurs bipolaires, les moteurs unipolaires et les moteurs Hybrides. Ces moteurs existent entre différentes tailles, tout dépend des applications dans lesquelles ils seront utilisés.

Le moteur NEMA 17 est un moteur pas à pas bipolaire avec un angle de pas de 1,8° (200 pas / tour). Chaque phase tire 1,2 A à 4 V, permettant un couple de maintien

de 3,2 kg-cm. Le moteur pas à pas NEMA 17 est généralement utilisé dans les imprimantes, les machines CNC et les découpeuses laser.

Ce type de moteur est très courant dans tous les dispositifs où l'on souhaite faire du contrôle de vitesse ou de position en boucle ouverte, typiquement dans les systèmes de positionnement et d'indexation.

Pour assurer les mouvements selon les trois axes X, Y et Z on a besoin de trois moteurs avec une précision assez élevée (Fig.IV.I.8).

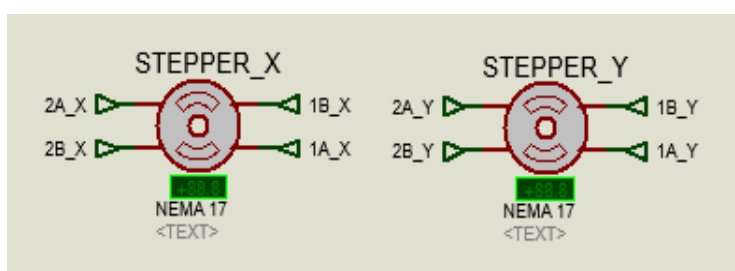


Fig.IV.I.8 Les moteurs des axes X et Y.

Dans l'environnement de simulation de ISIS Proteus on configure les moteurs comme suit (Fig.IV.I.9) [7] :

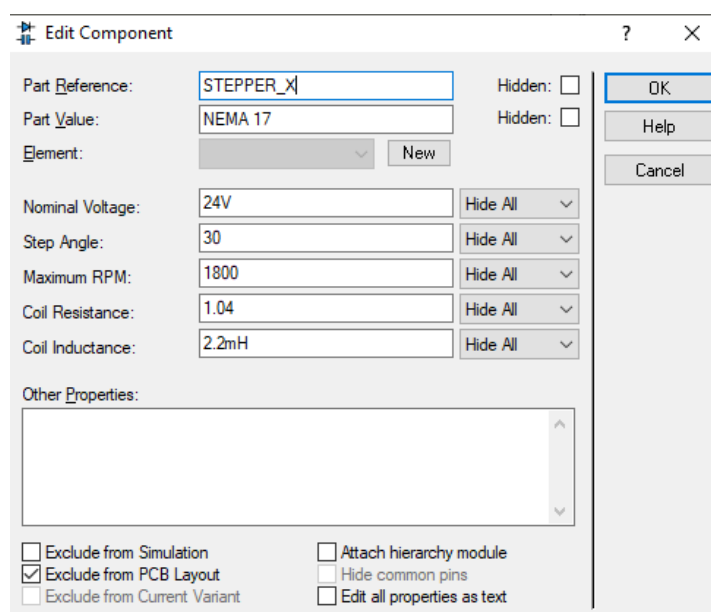


Fig.IV.I.9 Configuration des moteurs pas à pas.

IV.I.4.4 Les servos moteurs

Un servomoteur est un dispositif électrique qui peut pousser ou faire pivoter un objet avec une grande précision. Il est constitué d'un moteur simple qui fonctionne par servomécanisme. Si le moteur est alimenté en courant continu, il est appelé

servomoteur CC et s'il est alimenté en courant alternatif, il est appelé servomoteur AC. On peut obtenir un servomoteur à couple très élevé dans un petit boîtier léger. Grâce à ces caractéristiques, on peut l'utiliser dans de nombreuses applications comme voiture jouet, hélicoptères RC et avions, Robotique, Machine, etc.

IV.I.4.4.1 Mécanisme des servos moteurs

Un servomoteur est un servomécanisme en boucle fermée qui utilise le retour de position pour contrôler son mouvement et sa position finale. De plus, l'entrée de sa commande est un signal (analogique ou numérique) représentant la position commandée pour l'arbre de sortie.

Le moteur incorpore un certain type d'encodeur pour fournir un retour de position et de vitesse. Dans le cas le plus simple, on ne mesure que la position. Ensuite, la position mesurée de la sortie est comparée à la position de commande, l'entrée externe du contrôleur. Maintenant Si la position de sortie diffère de celle de la sortie attendue, un signal d'erreur est généré. Ce qui provoque alors la rotation du moteur dans les deux sens, selon la nécessité d'amener l'arbre de sortie dans la position appropriée. À l'approche de la position, le signal d'erreur se réduit à zéro. Enfin le moteur s'arrête [20].

Les servomoteurs très simples ne peuvent positionner que la détection via un potentiomètre et le contrôle bang-bang de leur moteur. De plus, le moteur tourne toujours à pleine vitesse. Bien que ce type de servomoteur n'ait pas beaucoup d'utilisations dans le contrôle de mouvement industriel, cependant il forme la base de servomoteur simple et bon marché utilisé pour les modèles de commande radio.

Les servomoteurs trouvent également des utilisations dans les encodeurs optiques rotatifs pour mesurer la vitesse de l'arbre de sortie et un variateur de vitesse variable pour contrôler la vitesse du moteur. Maintenant, ceci, combiné avec un algorithme de commande PID permet au servomoteur d'être dans sa position de commande plus rapidement et plus précisément avec moins de dépassement.

IV.I.4.4.2 Principe de fonctionnement des servomoteurs

Les servomoteurs contrôlent la position et la vitesse de manière très précise. Un potentiomètre peut détecter la position mécanique de l'arbre. Par conséquent, il se couple avec l'arbre du moteur par des engrenages. La position actuelle de l'arbre est

convertie en signal électrique par potentiomètre, et est comparée au signal d'entrée de commande. Dans les servomoteurs modernes, les codeurs ou capteurs électroniques détectent la position de l'arbre [21].

Nous donnons une entrée de commande selon la position de l'arbre. Si le signal de retour diffère de l'entrée donnée, un signal d'erreur avertit l'utilisateur. On amplifie ce signal d'erreur et l'applique comme entrée du moteur, d'où la rotation du moteur. Et lorsque l'arbre atteint la position requise, le signal d'erreur devient nul et, par conséquent, le moteur reste immobile en maintenant la position.

L'entrée de commande se présente sous forme d'impulsions électriques. Comme l'entrée réelle du moteur est la différence entre le signal de retour (position actuelle) et le signal requis, la vitesse du moteur est donc proportionnelle à la différence entre la position actuelle et la position requise. La quantité de puissance requise par le moteur est proportionnelle à la distance dont il a besoin pour parcourir [20].

IV.I.4.4.3 Contrôle des servomoteurs

Généralement, un servomoteur tourne de 90 degrés dans les deux sens, le mouvement maximum peut donc être de 180 degrés. Cependant, un servomoteur normal ne peut plus tourner après une butée mécanique intégrée.

Nous prenons trois fils qui sortent d'un servo : le fil positif, de masse et de contrôle. Un servomoteur est commandé en envoyant un signal modulé en largeur d'impulsion (PWM) à travers le fil de commande (Fig.IV.I.10). Une impulsion est envoyée toutes les 20 millisecondes. La largeur des impulsions détermine la position de l'arbre [21].

Par exemple, une impulsion de 1 ms déplacera l'arbre dans le sens inverse des aiguilles d'une montre à -90 degrés, une impulsion de 1,5 ms déplacera l'arbre à la position neutre qui est de 0 degré et une impulsion de 2 ms déplacera l'arbre dans le sens horaire à +90 degrés.

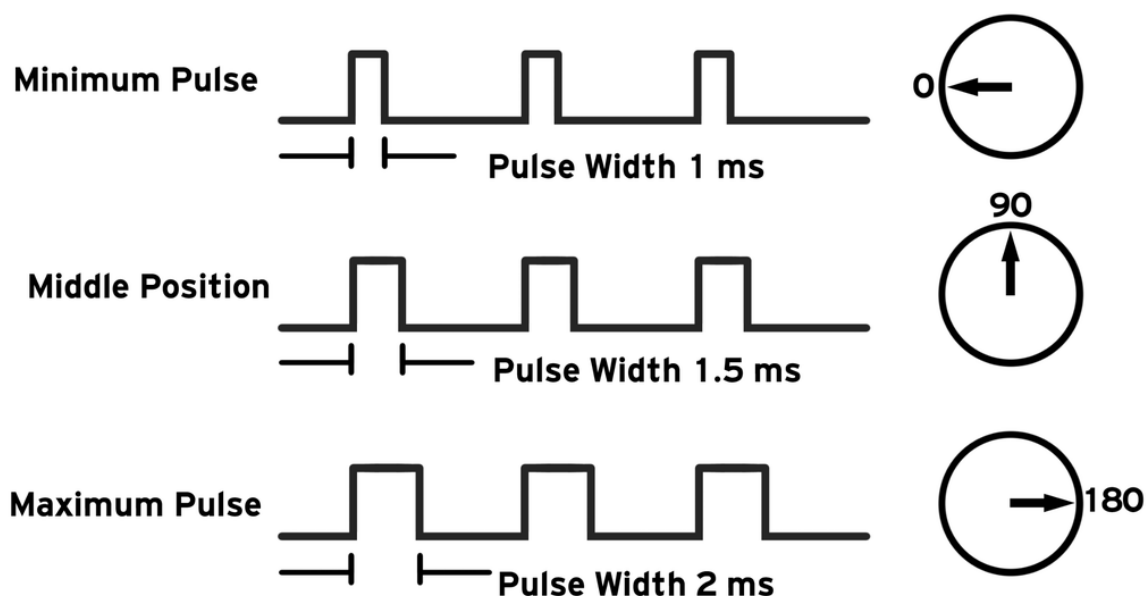


Fig.IV.I.10 Commande d'un servo moteur par PWM.

Lorsque on commande un servomoteur de se déplacer en appliquant une impulsion de largeur appropriée, l'arbre se déplace vers et maintient la position requise de l'arbre. Cependant, le moteur résiste au changement. Les impulsions doivent être répétées pour que le moteur maintienne la position.

Dans ce travail on a utilisé de types de servo moteur selon nos besoins et leurs caractéristiques.

IV.I.4.4.4 MG996R Servo moteur

Le MG996R est un servomoteur à engrenages métalliques avec un couple de décrochage maximal de 11 kg / cm. Comme les autres servos RC, le moteur tourne de 0 à 180 degrés en cycle d'utilisation de l'onde PWM fournie à sa broche de signal [22]. Ce servo est caractérisé par :

- La tension de fonctionnement est généralement de + 5V
- Courant : 2,5 A
- Couple de décrochage : 9,4 kg / cm (à 4,8 V)
- Couple de décrochage maximal : 11 kg / cm (à 6 V)
- La vitesse de fonctionnement est de 0,17 s / 60 °
- Type d'engrenage : métal
- Rotation : 0 ° -180 °
- Poids du moteur : 55gm

- Le forfait comprend des cornes et des vis

IV.I.4.4.5 MG90 Micro Servo moteur

MG90S est un micro servomoteur avec engrenage métallique. Ce servo petit et léger est livré avec une puissance de sortie élevée, donc idéal pour les bras robotiques [23]. Ce servo moteur est caractérisé par :

- Tension de fonctionnement : 4,8 V à 6 V (généralement 5 V)
- Couple de décrochage : 1,8 kg / cm (4,8 V)
- Couple de décrochage maximal : 2,2 kg / cm (6 V)
- La vitesse de fonctionnement est de 0,1 s / 60 ° (4,8 V)
- Type d'engrenage : métal
- Rotation : 0 ° -180 °
- Poids du moteur : 13,4 g
- Le forfait comprend des cornes et des vis

Pour notre bras robotique on a besoin de trois servomoteurs de type MG996R qui seront placer dans la base, épaule et le coude. On aussi besoin de trois autres servo moteur de type MG90 micro servo moteur qui seront placer dans la rotation du poignet (wrist roll), axe du poignet (wrist pitch) et la pince du robot.

IV.I.4.5 Les capteurs

Dans ce projet nous avons utilisé trois types de capteurs pour assurer le bon fonctionnement de la machine, ces capteurs sont les suivants

- Capteurs de distance
- Capteurs de fin de course
- Capteur de force

V.I.4.5.1 Capteur de distance

Le capteur HC-SR04 est un capteur à ultrason disponible dans la bibliothèque du logiciel de simulation ISIS de Proteus. Ce capteur fonctionne avec une tension d'alimentation de 5 volts, dispose d'un angle de mesure de 15° environ et permet de faire des mesures de distance entre 2 centimètres et 4 mètres avec une précision de 3m [8].

V.I.4.5.1.a Principe de fonctionnement du capteur

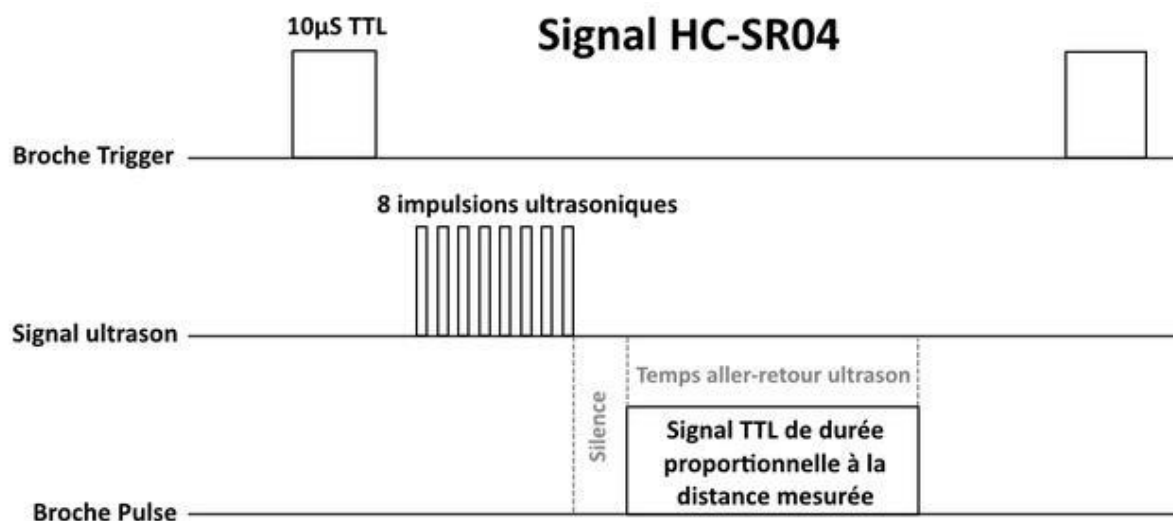


Fig.IV.I.11 Illustration du signal TRIGGER et ECHO.

Le principe de fonctionnement du capteur est entièrement basé sur la vitesse du son. Et la mesure se déroule comme suit :

1. On envoie une impulsion ECHO de 10µs sur la broche TRIGGER du capteur.
2. Le capteur envoie alors une série de 8 impulsions ultrasoniques à 40KHz.
3. Les ultrasons se propagent dans l'air jusqu'à toucher un obstacle et retournent dans l'autre sens vers le capteur.
4. Le capteur détecte l'écho et clôture la prise de mesure.

Le signal sur la broche ECHO du capteur reste à HIGH durant les étapes 3 et 4, ce qui permet de mesurer la durée de l'aller-retour des ultrasons et donc de déterminer la distance (Fig.IV.I.11) [8].

N.B. Il y a toujours un silence de durée fixe après l'émission des ultrasons pour éviter de recevoir prématurément un écho en provenance directement du capteur.

Pour pouvoir simuler ce capteur dans l'environnement ISIS de Proteus on rajoute une résistance variable, la variation de cette dernière représente la variation des ondes ultrasoniques (écho) détectées par le capteur et donc clôture la prise de mesure.

Dans ce projet nous avons besoin de deux capteurs de distance un 1^{er} pour la direction un 2^{ème} pour la translation.

Le capteur HC-SR04 est représenté dans l'environnement de simulation comme suit (Fig.IV.I.12) avec l'ajout du fichier .HEX dans ces paramètres :

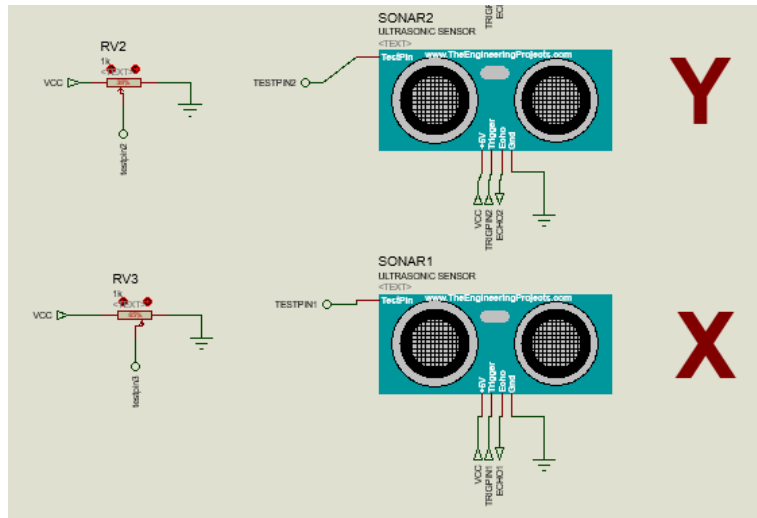


Fig.IV.I.12 Capteurs de distance HC-SR04.

IV.I.4.5.2 Capteur de fin de course

Les capteurs de fin de course sont utilisés pour détecter la présence du chariot sur les extrémités des axes X et Y, dans ce projet nous avons besoin de quatre capteurs de fin de course : deux dans les extrémités de la translation et deux dans les extrémités de la direction (Fig.IV.I.13).

Les capteurs de fin de course ne sont pas disponibles dans la bibliothèque du simulateur ISIS de Proteus pour cela nous avons les remplacer par des boutons poussoirs.

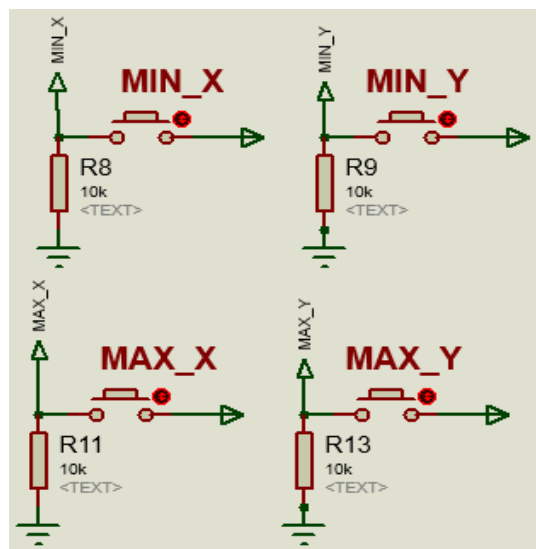


Fig.IV.I.13 Capteurs de fin de course

IV.I.4.6 Émetteur / Récepteur radio fréquence

Pour créer une liaison radio fréquence nous avons besoin d'un émetteur et un récepteur. L'émetteur reçoit un signal électrique qu'il convertit en signal électromagnétique (l'antenne).

Le rôle de l'antenne est d'émettre un signal qui va se propager dans la nature où il peut être capté par une ou plusieurs autres antennes (Fig.IV.14). L'antenne réceptrice conduit le signal reçu vers le récepteur d'où est extrait le même signal électrique que celui qui était en entrée de l'émetteur [24].



Fig.IV.I.14 Principe d'émission – réception.

IV.I.4.6.1 Le traitement du signal

IV.I.4.6.1.a La modulation

La figure IV.I.15 représente un signal audio que nous voulons le transmettre, il se présente électriquement sous la forme d'un signal alternatif dans lequel l'amplitude représente la puissance du son et la fréquence l'aspect grave ou aigu.



Fig.IV.I.15 Représentation électrique d'un signal audio.

Pour le transmettre, on émet le signal audio sur un signal haute fréquence sinusoïdal, que l'on appelle une porteuse.

La porteuse est un signal sinusoïdal pur, d'amplitude, de fréquence et de phase constantes (Fig.IV.I.16). Cette porteuse doit être modifiée afin de transporter le signal audio que l'on veut émettre. On dit alors qu'elle est modulée



Fig.IV.I.16 Porteuse pure.

La porteuse est décrite par l'équation [24] :

$$y = a \sin(2\pi ft + \varphi) \quad (\text{IV.1})$$

Où : a est l'amplitude de la porteuse, f est la fréquence de la porteuse et φ est la phase de la porteuse.

Pour moduler la porteuse, on va modifier un de ces trois paramètres en fonction du signal à transmettre [24].

a) Modulation d'amplitude

L'amplitude de la porteuse évolue en fonction de l'amplitude et de la fréquence du signal à transmettre. La forme extérieure du signal modulé, qui rappelle celle du signal d'entrée, est appelée l'enveloppe. La modulation d'amplitude est une modulation à enveloppe non constante puisque l'amplitude du signal modulé n'est pas constante (Fig.IV.I.17).

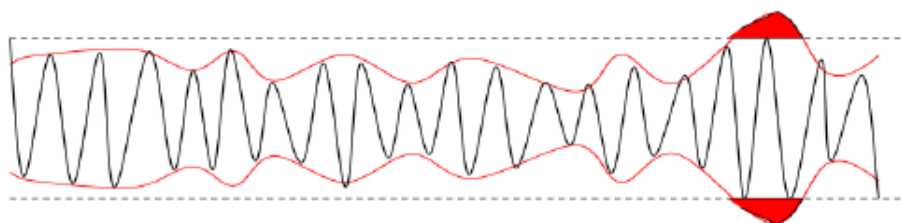


Fig.IV.I.17 Modulation d'amplitude.

C'est la forme de modulation qui fut utilisée, car simple à mettre en œuvre.

b) Modulation de fréquence

La modulation de fréquence est largement utilisée en radiodiffusion. Où la fréquence de la porteuse qui évolue en fonction du signal à transmettre (Fig.IV.I.18). L'amplitude de la porteuse est constante, c'est une modulation à enveloppe constante.

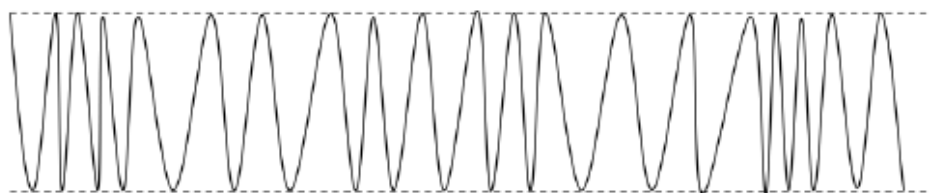


Fig.IV.I.18 Modulation de fréquence.

c) Modulation de phase

La modulation de phase consiste à agir sur la phase de la porteuse, en fonction du signal à émettre. C'est également une modulation à enveloppe constante, puisque l'amplitude de la porteuse n'est pas affectée (Fig.IV.I.19). Cette modulation est appliquée dans le domaine de la transmission de données.

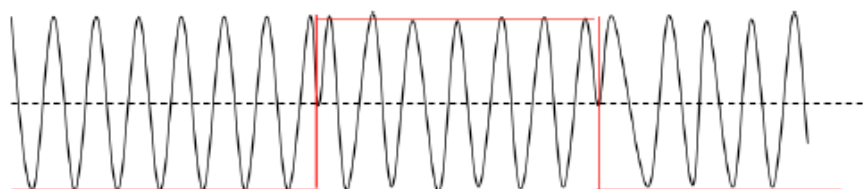


Fig.IV.I.19 Modulation de phase.

IV.I.4.6.2 Le traitement des informations numériques

IV.I.4.6.2.a La numération

La majorité des transmissions portent sur des signaux numériques binaires, composés de « 0 » et de « 1 ». Ce sont des signaux informatiques qui se présentent de manière native sous cette forme, mais ce sont aussi des signaux analogiques (voix, musique, vidéo, image...) qui vont être ainsi numérisés.

IV.I.4.6.2.b Le codage canal

Le codage est porté sur des trames, c'est-à-dire des blocs de longueur constante de l'information d'entrée. Une trame peut être un bloc issu d'un découpage en blocs de même longueur d'une donnée informatique ou d'une image.

Pour recevoir une information correcte, on exécute un codage avec redondance. C'est-à-dire un bit est combiné plusieurs fois avec des bits qui lui sont adjacents pour donner un nouveau bit. Par ce moyen, même s'il manque morceau dans la transmission, chaque bit peut être reconstitué à l'aide des combinaisons dans lesquelles il intervient.

Ceci est néanmoins un peu simpliste car chaque bit étant combiné avec ses voisins, des bits consécutifs vont se retrouver codés dans d'autres bits consécutifs de la trame codée et s'il arrive une grosse perturbation, tous vont disparaître, en même temps que la possibilité de restitution. Pour remédier à ceci, l'ordre des bits est modifié sur la trame codée, afin que des bits initialement consécutifs se retrouvent en différents emplacements de la trame. Cette opération s'appelle l'entrelacement.

IV.I.4.6.3 La chaîne de transmission

Un pontier veut déplacer un pont roulant à droite par radiocommande, Le signal est numérisé, par un codeur. Puis la trame issue du codeur est entrelacée. En réception, le signal reçu est amplifié, démodulé pour retrouver la trame numérique d'origine.

On fait à l'envers toutes les opérations décrites ci-dessus et en appliquant les corrections nécessaires à la récupération des erreurs de transmission. On retrouve alors une trame qui ressemble à celle sortie du codeur, qu'on repasse à l'envers dans un décodeur pour reconstituer le signal analogique qui représente.

Le module sans fil 433MHz est l'un des modules faciles à utiliser pour tous les projets sans fil. Ces modules ne peuvent être utilisés que par paires et seule la communication simplex est possible. Cela signifie que l'émetteur ne peut transmettre que de l'information et que le récepteur ne peut que la recevoir. Donc des données seront envoyées que du point A au point B et non du point B au point A. pour assurer la communication du point B au pont A on rajoute un autre émetteur et un autre récepteur [9]. Pour cela nous avons besoin de deux émetteurs et deux récepteurs pour assurer une communication bidirectionnelle.

Ce module pourrait couvrir un minimum de 3 mètres et avec une antenne appropriée une alimentation peut atteindre jusqu'à 100 mètres théoriquement. Mais pratiquement nous pouvons à peine obtenir environ 30-35 mètres dans des conditions d'essai normales.

IV.I.4.6.4 Emetteur RF 433MHz

Le cœur du module émetteur est le résonateur SAW qui est réglé pour l'opération 433.xx Mhz. Il y a un transistor de commutation et quelques composants passifs (Fig.IV.I.20).

Lorsqu'une logique HIGH est appliquée à l'entrée DATA, l'oscillateur exécute une onde porteuse de sortie RF constante à 433,xx Mhz et lorsque l'entrée DATA est prise à la logique LOW, l'oscillateur s'arrête. Cette technique est connue sous le nom d'Amplitude Shift Keying [10].

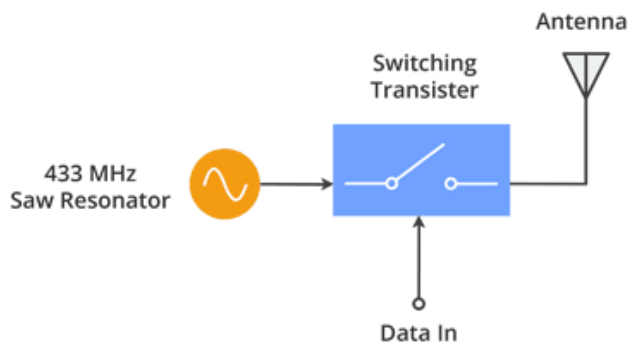


Fig.IV.I.20 Emetteur Radio fréquence.

IV.I.4.6.5 Récepteur RF 433MHZ

Le module récepteur bien qu'il semble complexe, il est aussi simple que le module émetteur. Il se compose d'un circuit accordé RF et de deux amplificateurs opérationnels pour amplifier l'onde porteuse reçue de l'émetteur (Fig.IV.I.21). Le signal amplifié est ensuite envoyé à une boucle de verrouillage de phase PLL (Phase Lock Loop) qui permet au décodeur de se "verrouiller" sur un flux de bits numériques qui offre une meilleure sortie décodée et une meilleure immunité au bruit [11].

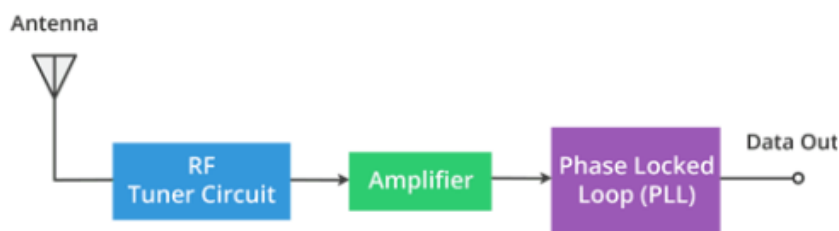


Fig.IV.I.21 Récepteur radio fréquence 433MHz.

IV.I.4.6.6 ASK – Amplitude Shift Keying

Pour l'envoi des données numériques par radio, ces modules utilisent une technique appelée Amplitude Shift Keying ou ASK. Dans Amplitude Shift Keying, l'amplitude de l'onde porteuse (dans notre cas, il s'agit d'un signal à 433 MHz) est modifiée en réponse au signal de données entrant.

Ceci est très similaire à la technique analogique de modulation d'amplitude. C'est parfois ce que l'on appelle le décalage d'amplitude binaire car il ne s'agit que de deux niveaux. On peut le considérer comme un interrupteur ON / OFF.

- Pour Digital 1 - Cela entraîne le support à pleine puissance.
- Pour Digital 0 - Cela coupe complètement la porteuse.

Voici à quoi ressemble la modulation d'amplitude (Fig.IV.I.22) :

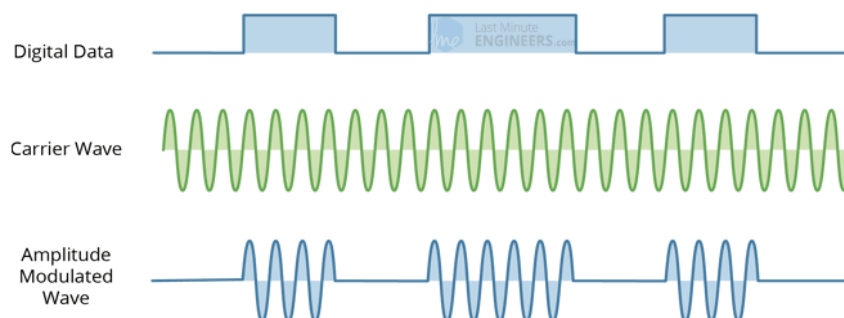


Fig.IV.I.22 Modulation d'amplitude.

Le codage Amplitude Shift a l'avantage d'être très simple à mettre en œuvre. Il est assez simple de concevoir les circuits du décodeur. ASK a également besoin de moins de bande passante que d'autres techniques de modulation comme FSK (Frequency Shift Keying). C'est une des raisons d'être peu coûteux.

L'inconvénient est cependant que ASK est sensible aux interférences d'autres appareils radio et au bruit de fond. Mais tant qu'on garde la transmission de données à une vitesse relativement lente, elle peut fonctionner de manière fiable dans la plupart des environnements.

IV.I.4.6.7 Spécifications du module 433 MHz

- Émetteur et récepteur simplex sans fil (RF)
- Tension de fonctionnement du récepteur : 3 V à 12 V
- Récepteur Courant de fonctionnement : 5,5 mA
- Fréquence de fonctionnement : 433 MHz
- Distance de transmission : 3 mètres (sans antenne) à 100 mètres (maximum)
- Technique de modulation : ASK
- Vitesse de transmission des données : 10 Kbps
- Type de circuit : Résonateur de scie

IV.I.4.6.8 Configuration des broches

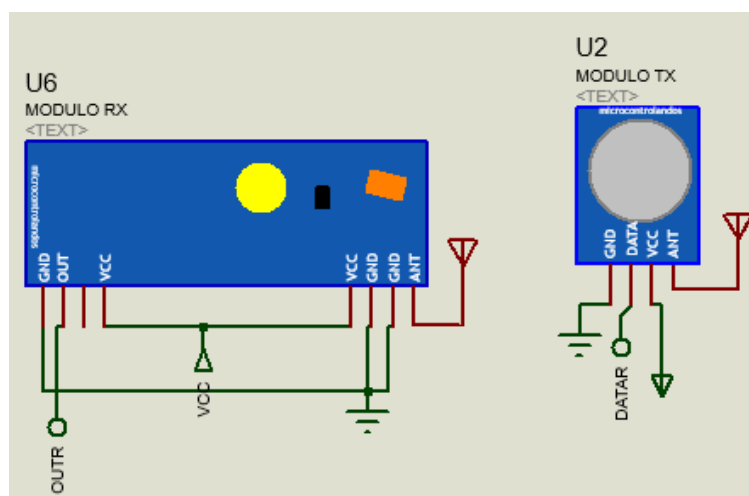


Fig.IV.I.23 Emetteur (TX) et récepteur (RX) radio fréquence 433MHz.

Le tableau IV.I.4 représente une description des broches du module rf433 :

Nom de la broche	Description
VCC	Alimentation (3V à 12V)
OUT	Les données reçues peuvent être obtenues à partir de cette broche
DATA	La broche accepte les données numériques à transmettre.
GND	Connecté à la masse du circuit
Antenne	Antenne pour améliorer la portée

Tableau IV.I.4 Configurations des pins du module Rf.

IV.I.4.7 Afficheur LCD :

Un afficheur LCD (Liquid Cristal Display) Alphanumérique (2 lignes x 16 colonnes) est interfacé avec la carte électronique pour nous permettre de visualiser en temps réel nos radiocommandes ainsi la position du chariot dans le prototype (Fig.IV.I.24).

IV.I.4.7.1 Présentation de l'afficheur

Le type d'afficheur étudié permet d'afficher [12] :

- Tous les caractères ASCII internationaux : chiffres et lettres, caractères de ponctuations, opérateurs arithmétiques, etc.
- Les caractères nationaux.
- L'afficheur permet aussi de placer un curseur ou de faire clignoter le caractère à l'endroit où le prochain caractère doit être affiché.

L'afficheur est constitué de :

- Un CI contrôleur Hitachi HD44780.
- Un ou plusieurs CI unité d'extension (le contrôleur seul ne permet que de piloter 2 lignes de 8 caractères) : un CI pour 2 lignes de 16 colonnes.
- Une matrice LCD de 2 lignes de 16 colonnes

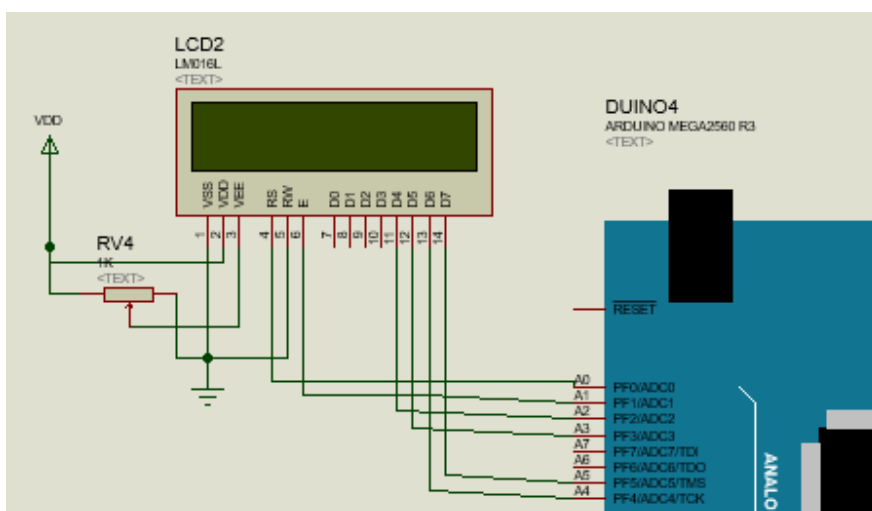


Fig.IV.I.24 Afficheur LCD LM016L.

IV.I.4.7.2 Description des pins et brochage

Le tableau IV.I.5 suivant représente une description des broches de l'afficheur LCD [12].

Numéro de pin	Symbole	Type	Description
01	GND	Alimentation	0V
02	VDD	Alimentation	5V
03	VEE	Alimentation	Réglage de contraste
04	RS	Entrée	Registre select
05	R/W	Entrée	Write/Read
06	E	Entrée	Enable
07	DB0	Entrée/sortie	Data bit 0
08	DB1	Entrée/sortie	Data bit 1
09	DB2	Entrée/sortie	Data bit 2
10	DB3	Entrée/sortie	Data bit 3
11	DB4	Entrée/sortie	Data bit 4
12	DB5	Entrée/sortie	Data bit 5
13	DB6	Entrée/sortie	Data bit 6
14	DB7	Entrée/sortie	Data bit 7

Tableau IV.I.5 Description des broches du LCD.

On applique une tension variable sur le pin VEE pour changer le contraste.

IV.I.4.8 Keypad- smallclac (4×4)

Un clavier est l'un des dispositifs les plus utilisés dans les applications de microcontrôleurs. Dans un clavier standard câblé comme une matrice de commutateur X-Y, les commutateurs normalement ouverts relient une ligne à une colonne lorsqu'ils sont enfoncés. Dans ce prototype nous avons besoin d'un keypad (4×4).

Le keypad de commande est utilisé comme dispositif d'entrée pour lire la touche enfoncée par l'utilisateur et pour la traiter.

Ce clavier 4x4 composé de 4 lignes et 4 colonnes. Des commutateurs sont placés entre les lignes et les colonnes (Fig.IV.I.25).

Une pression de touche établit une connexion entre la ligne correspondante et la colonne, entre laquelle le commutateur est placé [13].

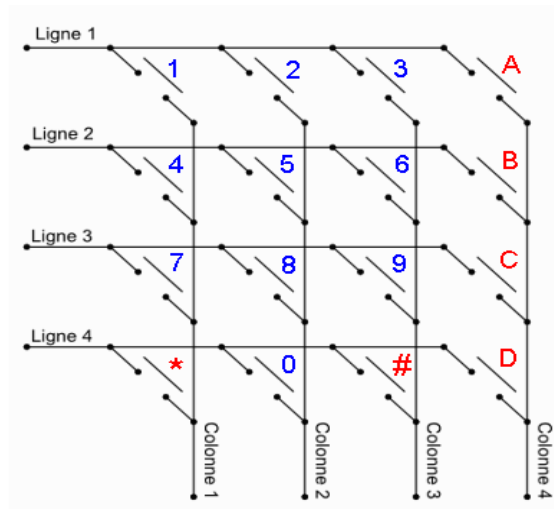


Fig.IV.I.25 Schéma interne du clavier.

Dans ce travail on utilise le keypad (4×4) smallclac trouvé dans la bibliothèque du logiciel de simulation ISIS de Proteus. Ce clavier est utilisé pour la manipulation du bras robotique et aussi pour le déplacement du pont roulant. Pour pouvoir faire ceci on a besoin d'un bouton switch qui sépare les deux fonctionnements.

Donc si le switch égale à **1** l'utilisateur est entrain de manipuler le pont roulant et si le switch égale à **0** ; il est entrain de manipuler le bras robotique

IV.I.4.8.1 Contrôler le pont roulant

Pour commander le pont roulant par radio fréquence on a besoin d'un clavier comme celui dans la figure IV.I.26.

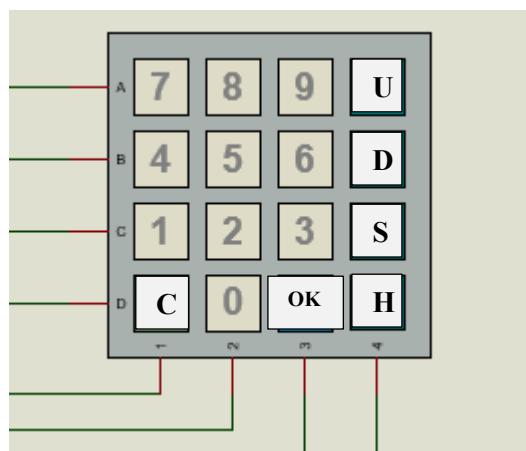


Fig.IV.I.26 Keypad- smallclac pour le pont (4×4).

Le tableau suivant représente la signification des abréviations du keypad :

Symbole	Signification
U	Up
D	Down
S	Stop
H	Home
C	Clear
OK	Pour la validation

Tableau IV.I.6 Signification des abréviations du keypad numéro 1.

Une pression sur la touche up permet de lever le crochet en haut et la touche Down c'est le contraire elle permet de descendre le crochet en bas.

Une pression sur la touche STOP permet d'arrête tous le chariot (joue le rôle d'un bouton d'urgence).

La touche Home permet au chariot de revenir à ces points initiaux prédéterminés.

IV.I.4.8.2 Contrôler le bras manipulateur

Pour commander le bras par radio fréquence on a besoin d'un clavier comme celui dans la figure (Fig.IV.I.27).

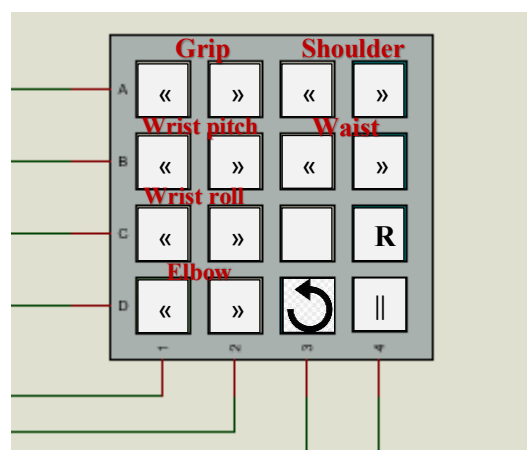


Fig.IV.I.27 Keypad- smallclac (4×4) pour le bras.

Le tableau IV.I.7 représente la signification des abréviations du clavier :



Symbole	Signification
Grip	Pince
Wrist pitch	Pas du poignet
Wrist roll	Rotation du poignet
Elbow	Coude
Shoulder	Epaule
Waist	Base
R	Run
	Reset
	Pause
»	Rotation à droite
«	Rotation à gauche

Tableau IV.I.7 Signification des abréviations du keypad numéro 2.

Une pression sur la touche « permet de tourner le servo moteur de l'articulation à gauche et une autre pression sur la touche » permet de le tourner à droite.

Pour faire en marche toutes les étapes qui ont été sauvegarder en avant il suffit d'appui sur le bouton **R** : Run.

Pour faire une pause il faut appui sur le bouton || qui permet de mettre notre système en pause.

Si on veut effacer tous ce qui été enregistrer en avant un appui sur le bouton  fera l'affaire.

IV.I.4.9 Les boutons poussoirs

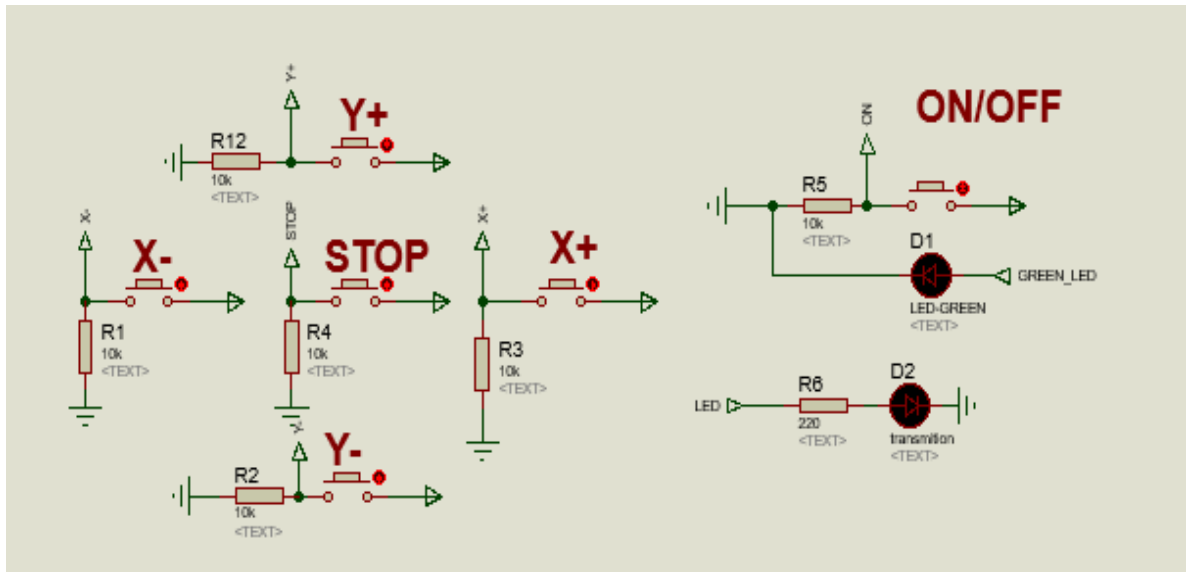


Fig.IV.I.28 Les boutons de commande.

Le rôle de ces boutons est d’assurer le bon fonctionnement et la commande du pont roulant en cas d’une défaillance dans la radio commande le tableau suivant monte le rôle de chaque bouton.

Bouton	Descriptions
ON / OFF	Pour mettre en marche / en arrêt le prototype
STOP	Joue le rôle d’un bouton d’urgence une fois poussé toutes les opérations vont s’arrêter immédiatement
X+	Pour assurer les déplacements à droite du chariot
X-	Pour assurer les déplacements à gauche du chariot
Y+	Pour assurer la marche en avant du pont roulant
Y-	Pour assurer la marche en arrière du pont roulant

Tableau IV.I.8 Descriptions des boutons de commande.

On a aussi deux autres boutons poussoir dans la boîte de radiocommande comme dans la figure (Fig.IV.I.29)

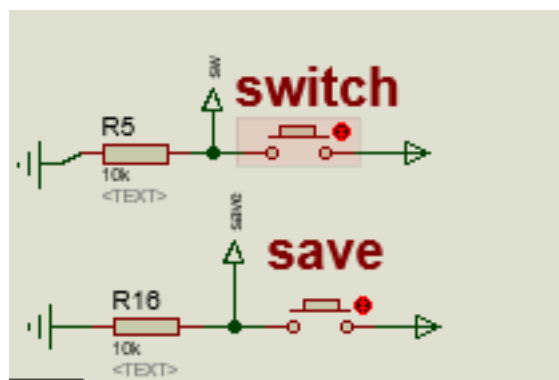


Fig.IV.I.29 Boutons de la boîte de radiocommande.

Le bouton switch permet de changer l'état de la radiocommande ; si le switch égale à 1 l'utilisateur est entrain de commander le pont roulant, si le switch égale à 0 ; il est entrain de manipuler le bras robotique.

Ce switch permet d'économiser un peu dans le matériel et l'espace de la boîte de la radiocommande, donc au lieu d'utiliser deux claviers ; un pour le pont roulant et un autre pour le bras robotique, on utilise un seul clavier avec un switch.

On a rajouté deux LED d'indication :

- LED-GREEN pour indiquer que le pont roulant est en état de marche.
- Transition pour indiquer qu'il y a une communication entre le pont et la boîte de commande.

Conclusion

Dans ce chapitre, nous avons commencé par une présentation du schéma général de notre montage, puis nous avons étudié les propriétés de tous les éléments électroniques que nous avons besoin et qui constituent notre prototype.

Par la suite nous allons entamer la partie software du fonctionnement du pont roulant à l'aide du logiciel ISIS de Proteus et Arduino IDE.

Partie IV.II

Software

*L*a partie software est indispensable pour la réalisation de notre projet vue que le fonctionnement de la partie mécanique et la partie hardware de projet est lié directement au software utilisé pour modéliser les différentes pièces mécaniques ainsi, le développement de firmware et le circuit de commande.

Dans cette partie nous allons parler du logiciel de modélisation 3D SOLIDWORKS, l'environnement de développement ARDUINO IDE et les bibliothèques utilisés dans la programmation du prototype et par la suite la modélisation de notre travail avec un diagramme d'activité.

IV.II.1 Le logiciel SolidWorks

SolidWorks est un logiciel de CAD (computer Aided Design) très puissant et facile à manipuler, dédié au design et la modélisation 3D. Il est très utilisé dans l'industrie pour la fabrication de différentes pièces mécaniques, hydrauliques, électriques ainsi l'assemblage des pièces et la conception des machines et des appareils de A à Z [14].

De plus, SolidWorks offre une variété de choix de matériaux à utiliser ainsi, il donne la possibilité de faire des simulations sur l'objet créé pour tester les effets de différents contraintes appliqués sur l'objet avant de le construire ou tout simplement pour le relier à un autre logiciel comme Matlab pour tester sa fonctionnalité avec le code écrit.

Dans ce projet nous avons utilisé le SolidWorks pour modéliser notre prototype du pont roulant avec le bras robotique (Fig.IV.II.1).

IV.II.1.1 Modélisation du pont roulant avec SolidWorks

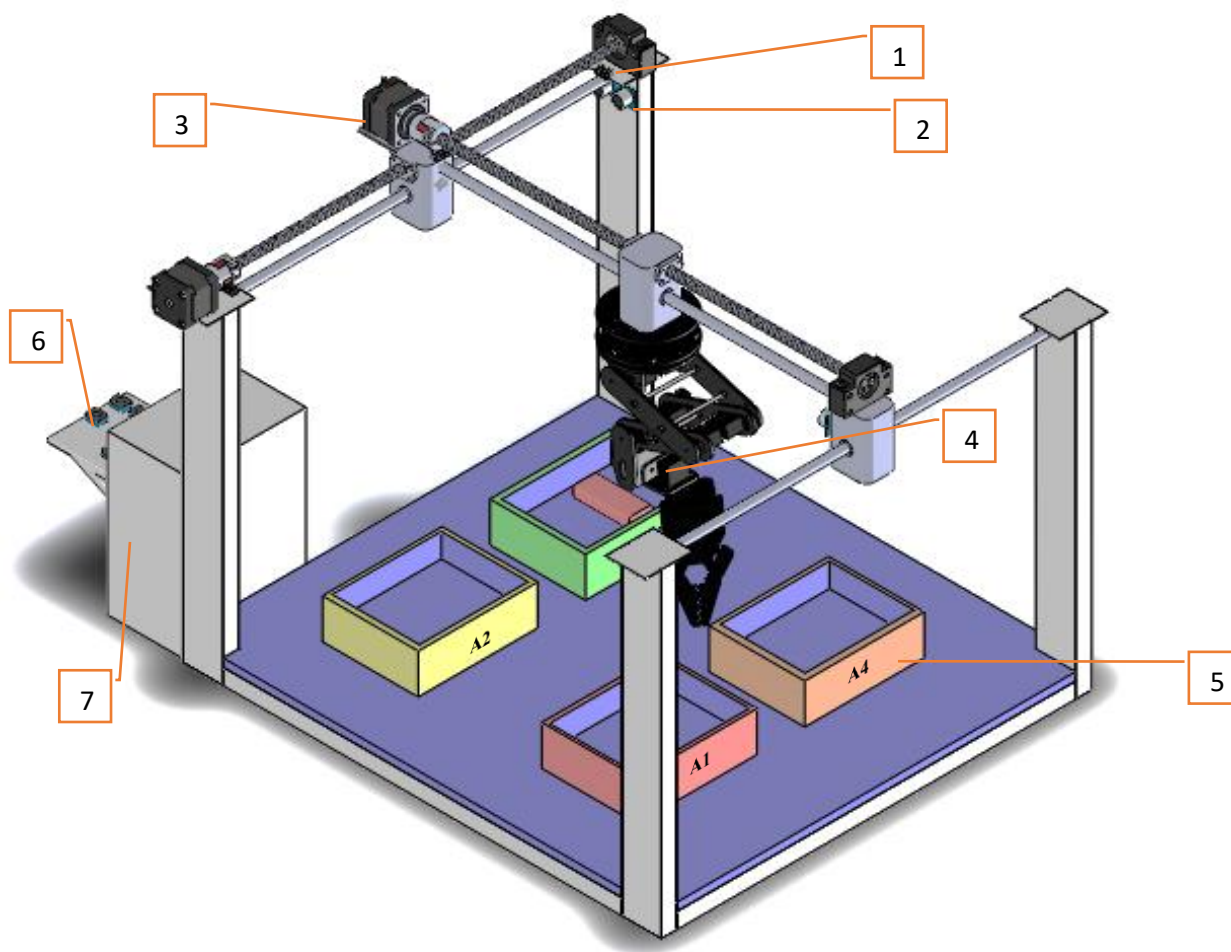


Fig.IV.II.1 Désigne du prototype du projet.

La figure IV.II.1 représente la modélisation de notre pont roulant avec sa boîte de commande et d'alimentation.

Numéro	Signification
1	Capteur de fin de course
2	Capteur de distance ultrason
3	Moteur pas à pas NEMA17
4	Bras robotique
5	Atelier
6	Boutons de commande
7	Boîte de contrôle et d'alimentation

Tableau IV.II.1 Signification des éléments du prototype.

IV.II.1.2 Modélisation de la boîte de commande RF

La figure IV.II.2 représente le désigne de la boîte de la radiocommande faite par le logiciel SolidWorks.

Cette boîte de commande en tant que fermée (Fig.IV.II.2) permet de déplacer le pont roulant, et de manipuler le bras robotique en tant qu'ouverte (Fig.IV.II.3)



Fig.IV.II.2 Boite de radiocommande fermée.

Numéro	Signification
1	Afficheur LCD 16*2
2	Bouton de commande
3	Bouton d'alimentation
4	Bouton d'enregistrement des mouvements
S	Stop
H	Home
C	Clear

Tableau IV.II.2 Signification des abréviations de la boîte de radiocommande fermée.



Fig.IV.II.3 Boite de radiocommande ouverte.

Numéro	Signification
1	Afficheur LCD 16*2
2	Bouton de commande
3	Bouton d'enregistrement des mouvements
4	Bouton d'alimentation
R	RUN
	Pause
↻	Reset

Tableau IV.II.3 Signification des abréviations de la boite de radiocommande ouverte.

IV.II.2 L'environnement de développement Arduino IDE

Arduino IDE est un environnement de développement se présente comme un ensemble d'outils qui comporte des bibliothèques de code, un éditeur de texte, un compilateur et un débogueur. Il est souvent dédié à un seul langage de programmation [15].

Le développement de Firmware de notre projet sur le microcontrôleur Atmega2560 nécessite l'utilisation d'un environnement de développement dédié à ce microcontrôleur de la famille Atmel. Il existe plusieurs environnements parmi eux le Arduino IDE, Atmel Studio et AVR Studio.

Le langage Arduino est inspiré de plusieurs langages. On retrouve notamment des similarités avec le C, le C++ et le Java. C'est pour cela qu'on a développé notre Firmware sous l'environnement Arduino IDE qui utilise un compilateur C/C++ avec un éditeur de texte et une interface Homme-machine très simple.

Le chargement de programme dans le microcontrôleur se fait directement par port USB d'une façon simple et facile

IV.II.2.1 Structure d'un projet Arduino

A l'ouverture d'un nouveau projet sur Arduino IDE, on trouvera deux fonctions déjà prédéfinies dans l'éditeur du texte qui sont les fonctions « void setup () » et « void loop() », l'appellation de ces deux fonctions est fixe et non modifiable.

- **La fonction « Setup »** : Tout ce que vous y mettez, la fonction sera exécutée par l'Arduino une seule fois lorsque le programme démarre.

- **La fonction « Loop »** : Une fois que l'Arduino a terminé avec le code de la fonction setup (), il se déplacera dans Loop (), et il continuera à l'exécuter en boucle, encore et encore, jusqu'à ce que vous le réinitialisiez ou coupiez l'alimentation. (Fig IV.II.4).

```
1 void setup() {  
2   // put your setup code here, to run once:  
3  
4 }  
5  
6 void loop() {  
7   // put your main code here, to run repeatedly:  
8  
9 }
```

Fig.IV.II.4 Structure d'un code Arduino.

Le cycle de développement est divisé en 4 phases :



- **Compiler** : Compiler signifie traduire l'esquisse en langage machine, également connu sous le nom de code objet

- Exécuter : Arduino croquis est exécuté dès que se termine l'étape de chargement sur la carte.

IV.II.3 Les bibliothèques utilisées

Dans la programmation de ce firmware on avait besoin de plusieurs bibliothèques pour assurer le bon fonctionnement de notre matériel.

IV.II.3.1 Bibliothèque RadioHead

RadioHead est une bibliothèque qui permet un transfert de données simple entre des cartes Arduino. Il est tellement polyvalent qu'il peut être utilisé pour piloter toutes sortes d'appareils de radiocommunication, y compris nos modules 433 MHz [11]. Pour utiliser cette bibliothèque : `#include <RH_ASK.h>`

Ce que fait la bibliothèque RadioHead, c'est de prendre nos données, de les encapsuler dans un paquet de données qui comprend un CRC (Cyclic Redundancy Check), puis de les envoyer avec le préambule et l'en-tête nécessaires à un autre Arduino. Si les données sont reçues correctement, l'Arduino récepteur est informé qu'il existe des données et procède à leur décodage.

Le paquet RadioHead est composé comme suit : Un flux de 36 bits de paires de bits «1 » et «0 », appelé « Training Preamble » (préambule d'apprentissage), est envoyé au début de chaque transmission. Ces bits sont nécessaires pour que le récepteur ajuste son gain avant d'obtenir les données réelles. Ensuite, un « start symbol » (symbole de début) de 12 bits, puis les données réelles (payload) sont ajoutées.

Une séquence de vérification de trame ou CRC est ajoutée à la fin du paquet qui est recalculée par RadioHead à l'extrémité du récepteur et si la vérification CRC est correcte, le périphérique récepteur est alerté. Si la vérification CRC est échouée, le paquet sera rejeté.

L'ensemble du paquet ressemble à ceci (Fig IV.II.5) :

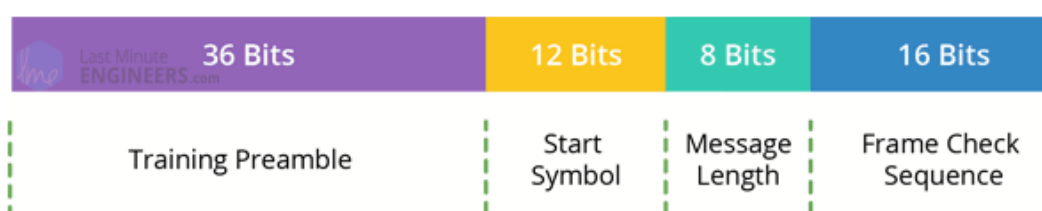


Fig.IV.II.5 Paquet d'information Radio Head.

IV.II.3.2 Bibliothèque SPI

Cette bibliothèque vous permet de communiquer avec des appareils SPI, avec l'Arduino comme appareil maître [16]. Pour utiliser cette bibliothèque : `#include <SPI.h>`

Serial Peripheral Interface (SPI) ou L'interface périphérique série (SPI) est un protocole de données série synchrone utilisé par les microcontrôleurs pour communiquer rapidement avec un ou plusieurs périphériques sur de courtes distances. Il peut également être utilisé pour la communication entre deux microcontrôleurs.

Avec une connexion SPI, il y a toujours un périphérique maître (généralement un microcontrôleur) qui contrôle les périphériques. En règle générale, trois lignes sont communes à tous les appareils :

- MISO (Master In Slave Out) - La ligne esclave pour envoyer des données au maître,
- MOSI (Master Out Slave In) - La ligne Master pour l'envoi de données aux périphériques,
- SCK (Serial Clock) - Les impulsions d'horloge qui synchronisent la transmission de données générée par le maître

Et une ligne spécifique pour chaque appareil :

- SS (Slave Select) - la broche sur chaque appareil que le maître peut utiliser pour activer et désactiver des appareils spécifiques.

Lorsque la broche Slave Select d'un appareil est LOW, il communique avec le maître. Quand il est HIGH, il ignore le maître. Cela vous permet d'avoir plusieurs appareils SPI partageant les mêmes lignes MISO, MOSI et CLK.

La norme SPI est lâche et chaque appareil l'implémente un peu différemment. D'une manière générale, il existe quatre modes de transmission. Ces modes contrôlent si les données sont déplacées vers l'intérieur et vers l'extérieur sur le front montant ou descendant du signal d'horloge de données (appelé la phase d'horloge), et si l'horloge est inactive lorsqu'elle est élevée ou basse (appelée la polarité d'horloge). Les quatre modes combinent polarité et phase selon le tableau IV.II.4 :

Mode	Polarité de l'horloge (CPOL)	Phase d'horloge / (CPHA)	Bord de sortie	Capture de données
SPI_MODE0	0	0	Falling	Rising
SPI_MODE1	0	1	Rising	Falling
SPI_MODE2	1	0	Rising	Falling
SPI_MODE3	1	1	Falling	Rising

Tableau IV.II.4 Combinaison polarité et phase.

IV.II.3.3 Exemple d'utilisation

La bibliothèque RadioHead nécessite l'utilisation de la bibliothèque SPI la figure (Fig IV.II.6) présente un exemple simple d'un sketch d'un émetteur

```

1 // Include RadioHead Amplitude Shift Keying Library
2 #include <RH_ASK.h>
3 // Include dependant SPI Library
4 #include <SPI.h>
5
6 // Create Amplitude Shift Keying Object
7 RH_ASK rf_driver;
8
9 void setup()
10 {
11     // Initialize ASK Object
12     rf_driver.init();
13 }
14
15 void loop()
16 {
17     const char *msg = "Hello World";
18     rf_driver.send((uint8_t *)msg, strlen(msg));
19     rf_driver.waitPacketSent();
20     delay(1000);
21 }

```

Fig.IV.II.6 Exemple émetteur RF.

Le sketch commence par inclure la bibliothèque RadioHead ASK. Nous devons également inclure la bibliothèque Arduino SPI car la bibliothèque RadioHead en dépend.

```

#include <RH_ASK.h>
#include <SPI.h>

```

Ensuite, nous devons créer un objet ASK afin d'accéder aux fonctions spéciales liées à la bibliothèque RadioHead ASK.

```
// Create Amplitude Shift Keying Object
RH_ASK rf_driver;
```

Dans la fonction de Setup(), nous devons initialiser l'objet ASK.

```
void setup()
{
    // Initialize ASK Object
    rf_driver.init();
}
```

En fonction Loop(), nous commençons par préparer un message. Il s'agit d'une simple chaîne de caractères stockée dans un pointeur de caractères nommé msg. Message transféré ne doit pas dépasser 27 caractères pour de meilleures performances. Dans notre cas, nous avons 11 caractères.

```
const char *msg = "Hello World";
```

Le message est ensuite transmis à l'aide d'une fonction send (). Il a deux paramètres : le premier est un tableau de données et le second est le nombre d'octets (longueur des données) à envoyer. La fonction send () est généralement suivie de la fonction waitPacketSent () qui attend jusqu'à ce que tout paquet de transmission précédent ait fini d'être transmis. Après cela, le croquis attend une seconde pour donner à notre récepteur le temps de tout comprendre.

```
rf_driver.send((uint8_t *)msg, strlen(msg));
rf_driver.waitPacketSent();
delay(1000);
```

Et c'est la même chose pour le récepteur figure IV.II.7 ;

```
// Include RadioHead Amplitude Shift Keying Library
#include <RH_ASK.h>
// Include dependant SPI Library
#include <SPI.h>

// Create Amplitude Shift Keying Object
RH_ASK rf_driver;

void setup()
{
    // Initialize ASK Object
    rf_driver.init();
    // Setup Serial Monitor
    Serial.begin(9600);
}

void loop()
{
    // Set buffer to size of expected message
    uint8_t buf[11];
    uint8_t buflen = sizeof(buf);
    // Check if received packet is correct size
    if (rf_driver.recv(buf, &buflen))
    {
        // Message received with valid checksum
        Serial.print("Message Received: ");
        Serial.println((char*)buf);
    }
}
```

Fig.IV.II.7 Exemple d'un récepteur rf

Tout comme l'émetteur, le code du récepteur commence par charger les bibliothèques RadioHead et SPI et créer un objet ASK.

Dans la fonction de setup() : nous initialisons l'objet ASK et configurons également le moniteur série car c'est ainsi que nous allons afficher notre message reçu..

```
rf_driver.init();
Serial.begin(9600);
```

Dans la fonction Loop() : nous créons un tampon de taille identique au message transmis. Dans notre cas, c'est 11.

```
uint8_t buf[11];  
uint8_t buflen = sizeof(buf);
```

Ensuite, nous appelons une fonction `recv()`. Cela met le récepteur sous tension s'il ne l'est pas déjà. Si un message valide est disponible, il copie le message dans son premier tampon de paramètres et retourne `true` sinon retourne `false`. Si la fonction renvoie `true`, le sketch entre dans l'instruction `if` et imprime le message reçu sur le moniteur série.

```
if (rf_driver.recv(buf, &buflen)  
{  
    Serial.print("Message Received: ");  
    Serial.println((char*)buf);  
}
```

Après avoir chargé le sketch, on ouvre le moniteur série. Si tout va bien, on aura le message « hello world ».

IV.II.3.4 Bibliothèque Software Servo

Cette bibliothèque permet à une carte Arduino de contrôler les servomoteurs RC. Ces servos moteurs ont des engrenages intégrés et un arbre qui peut être contrôlé avec précision. Les servos standard permettent de positionner l'arbre à différents angles, généralement entre 0 et 180 degrés. Les servos à rotation continue permettent de régler la rotation de l'arbre à différentes vitesses.

La bibliothèque de Software servos peut piloter des servos sur toutes nos broches simultanément. L'API est calquée sur la bibliothèque `wiring.org servo` mais le code est différent. On doit appeler `SoftwareServo::refresh()` au moins une fois toutes les 50 ms environ pour maintenir nos servos à jour. Pour utiliser cette bibliothèque :
`#include <SoftwareServo.h>`

IV.II.3.5 Bibliothèque AccelStepper

Cette bibliothèque fournit une interface orientée objet pour les moteurs pas à pas et les pilotes de moteur à 2, 3 ou 4 broches. Et convient parfaitement aux applications simples à moteur unique.

`AccelStepper` améliore considérablement la bibliothèque standard Arduino Stepper de plusieurs manières :

- Prend en charge l'accélération et la décélération.

- Les fonctions API ne retardent jamais () ou bloquent.
- Prend en charge les steppers à 2 et 4 fils.
- Prend en charge les pilotes pas à pas tels que le Sparkfun EasyDriver (basé sur la puce de pilote 3967).
- Les vitesses très lentes sont prises en charge

Pour utiliser cette bibliothèque : `#include <AccelStepper.h>`

IV.II.3.6 Bibliothèque LiquidCrystal

Cette bibliothèque permet à une carte Arduino de contrôler les écrans LiquidCrystal (LCD) basés sur le chipset Hitachi HD44780 (ou un compatible), que l'on trouve sur la plupart des LCD textuels. La bibliothèque fonctionne en mode 4 ou 8 bits (c'est-à-dire en utilisant 4 ou 8 lignes de données en plus des lignes rs, enable et, éventuellement, rw control) [17]. Pour utiliser cette bibliothèque `#include <LiquidCrystal.h>`

IV.II.4 Les outils de la modélisation

IV.II.4.1 Langage de modélisation

Un langage de modélisation est tout langage artificiel utilisé pour décrire et illustrer un système d'une manière structurelle définie par un ensemble de règles utilisées pour interpréter la signification de chaque composant de la structure. Par la suite, nous allons présenter le langage de modélisation le plus populaire dans le domaine du génie logiciel, l'UML [18].

IV.II.4.2 Unified modelling language (UML)

Le langage de modélisation unifié est un langage de modélisation à usage général. L'objectif principal d'UML est de définir une manière standard de visualiser la structure globale d'un système.

UML n'est pas un langage de programmation, c'est plutôt un langage visuel qui utilise les diagrammes pour décrire le comportement d'un système et il est utilisé par l'ingénierie logicielle, les hommes d'affaires et les architectes système dans la modélisation, la conception et l'analyse. Object Management Group a adopté le langage de modélisation unifié comme standard en 1997. Depuis, il est géré par OMG.

L'Organisation internationale de normalisation (ISO) a publié UML en tant que norme approuvée en 2005 [18].

UML utilise des éléments et des formulaires liés entre eux à partir de diagrammes qui peuvent être classés en deux types :

IV.II.4.2.1 Diagrammes structurels

Les diagrammes structurels UML décrivent les éléments d'un système qui sont indépendants du temps et qui transmettent les concepts d'un système et comment ils se rapportent les uns aux autres. Les éléments de ces diagrammes ressemblent aux noms dans un langage naturel, et les relations qui les relient sont des relations structurelles ou sémantiques. Capturent la structure statique du système.

Comme : Diagrammes de composants, diagrammes d'objets, diagrammes de classes et diagrammes de déploiement.

IV.II.4.2.2 Diagrammes de comportement

Les diagrammes de comportement UML visualisent, spécifient, construisent et documentent les aspects dynamiques d'un système ils sont classés comme suit : diagrammes de cas d'utilisation, diagrammes d'interaction, diagrammes d'état-transition et diagrammes d'activités.

La figure IV.II.8 illustre la hiérarchie des diagrammes en UML2.0

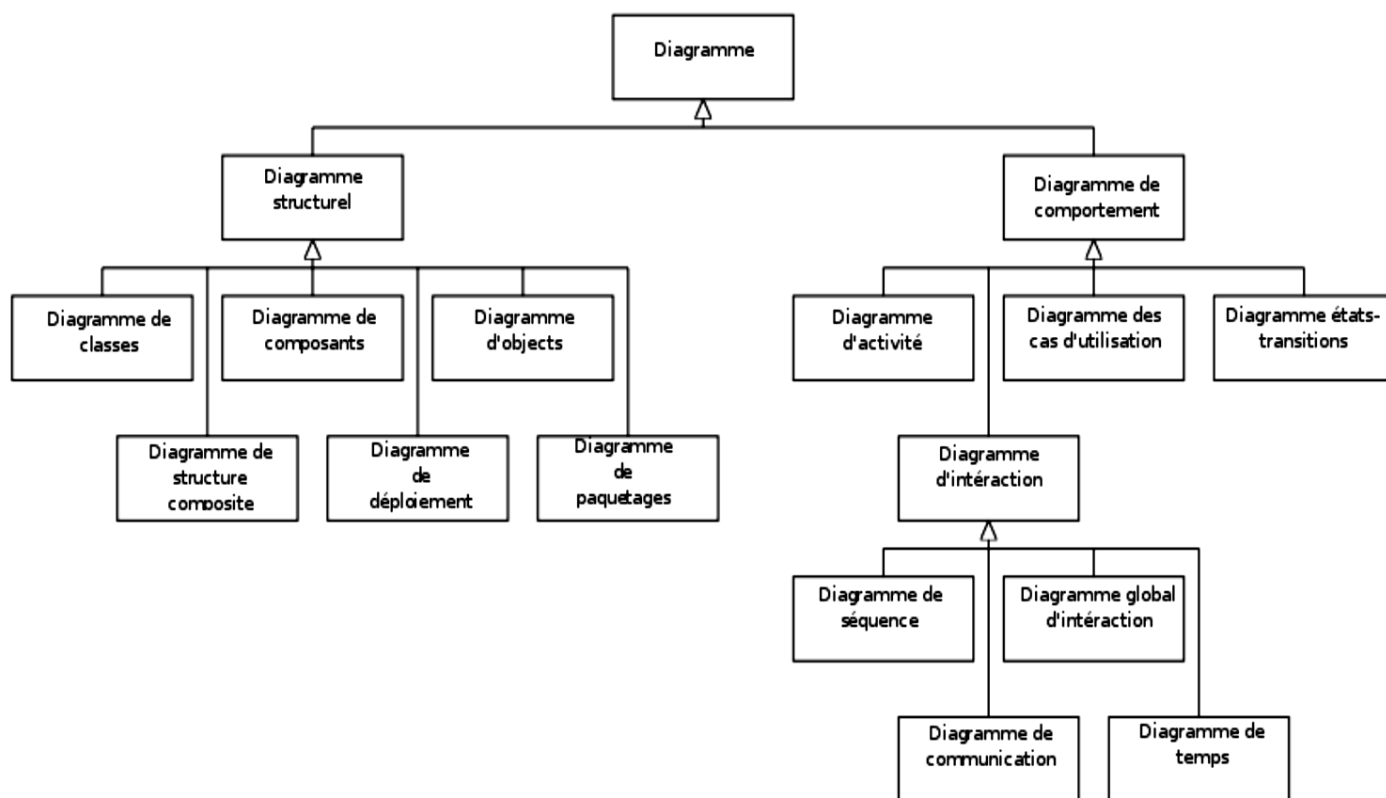


Fig.IV.II.8 La hiérarchie des diagrammes en UML2.0.

IV.II.4.2.3 Diagrammes d'activité

Les modèles UML représentent essentiellement trois types de diagrammes : les diagrammes de structure, les diagrammes d'interaction et les diagrammes de comportement. Un diagramme d'activité est un diagramme de comportement, c'est-à-dire qu'il décrit le comportement d'un système [18-19].




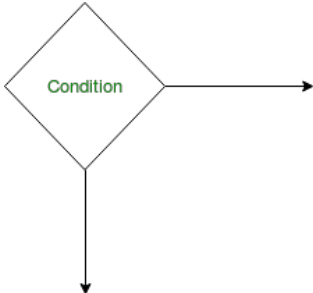
On utilise des diagrammes d'activités pour illustrer le flux de contrôle dans un système et on modélise aussi des activités séquentielles et simultanées à l'aide de ce diagramme. Ainsi, on décrit essentiellement les workflows visuellement. Un diagramme d'activité se concentre sur la condition du flux et la séquence dans laquelle il se produit.

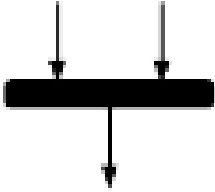
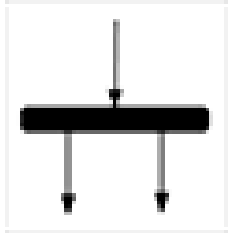




Un diagramme d'activité décrit le flux de contrôle d'un point de départ à un point d'arrivée montrant les différents chemins de décision qui existent pendant l'exécution de l'activité. Nous pouvons représenter à la fois le traitement séquentiel et le traitement simultané des activités à l'aide de ce diagramme. Ils sont utilisés aussi

dans la modélisation des opérations et des processus où leur utilisation principale est de décrire les aspects dynamiques d'un système.

IV.II.4.2.3.a Notations du diagramme d'activité

Dans le tableau suivant on regroupe les formes et les symboles de diagrammes d'activités UML les plus utilisés. [18,19]

Symbole	Nom	Description
	Symbole de début	Représente le début d'un processus ou d'un flux de travail dans un diagramme d'activités. Il peut être utilisé seul ou avec un symbole de note qui explique le point de départ.
	Symbole d'activité ou d'action	Une activité représente l'exécution d'une action sur des objets ou par des objets. Nous représentons une activité en utilisant un rectangle aux coins arrondis. Toute action ou tout événement est représenté avec une activité.
	Symbole de raccord	Les flux d'action ou les flux de contrôle sont également appelés chemins et bords. Ils sont utilisés pour montrer la transition d'un état d'activité à un autre.
	Symbole de décision	Lorsque nous devons prendre une décision avant de décider du flux de contrôle, nous utilisons le nœud de décision.

	Symbole de raccord/barre de synchronisation	Fusionne deux activités simultanées et en les réintroduise, une seule activité se produit à la fois. Elle est indiquée par une ligne verticale ou horizontale épaisse.
	Symbole d'embranchement	Divisez le flux d'activités en deux activités simultanées. Il est représenté par plusieurs flèches partant d'un raccord.
	Symbole de signal d'émission	Indique qu'un signal est envoyé à une activité réceptrice.
	Symbole du signal de réception	Représente l'acceptation d'un événement. Une fois l'événement reçu, le flux qui vient de cette action est terminé.
	Time Event	Nous pouvons avoir un scénario où un événement prend un certain temps. Nous utilisons un sablier pour représenter un événement temporel.
	Symbole de fin	L'état auquel le système atteint lorsqu'un processus ou une activité particulière prend fin est appelé état final ou état final. Nous utilisons un cercle rempli dans une notation de cercle pour représenter l'état final dans un diagramme de machine d'état. Un système ou un processus peut avoir plusieurs états finaux.

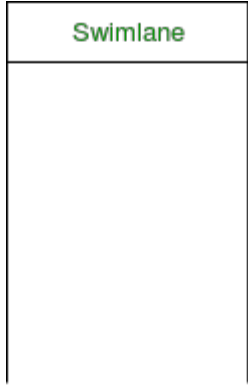
	Swimlanes	<p>Nous utilisons les couloirs pour regrouper les activités connexes dans une colonne. Les swimlanes peuvent être verticaux et horizontaux. Ils donnent plus de clarté au diagramme d'activité.</p>
---	-----------	---

Tableau IV.II.5 Notations du diagramme d'activité.

IV.II.5 Modélisation

Pour déplacer le pont roulant, le pontier laisse le couvercle de la boîte radiocommande fermé, un message sera affiché pour l'inviter à entrer le code d'atelier, si son code est valide un message sera envoyé par radiofréquence au deuxième microcontrôleur, ce dernier va traiter le signal reçu et exécute la commande adéquate. À chaque instant, le microcontrôleur compare la distance reçue des capteurs avec les coordonnées de l'atelier voulue. Une fois le pont arrive au but il s'arrête et envoie un message par radiofréquence au premier microcontrôleur et le LCD affiche un message de fin.

Pour que le pontier manipule le bras robotique ; il suffit d'ouvrir le couvercle et appuie sur une des boutons de commande des articulations.

Il a aussi la possibilité de sauvegarder ces mouvements par le bouton Save, exécuter ce qu'il a déjà sauvegarder par le bouton « Run », faire une pause ou effacer les anciennes étapes sauvegarder par le Reset.

Dans le cas d'un mal fonctionnement dans la radiocommande le pontier peut utiliser une commande manuelle qui assure la transition, le déplacement du chariot. Dans la commande manuelle on peut trouver aussi un bouton d'urgence qui permet d'arrêter la totalité du pont en cas d'urgence.

La figure suivante représente la modélisation du pont roulant avec le diagramme d'activité UML2.0.

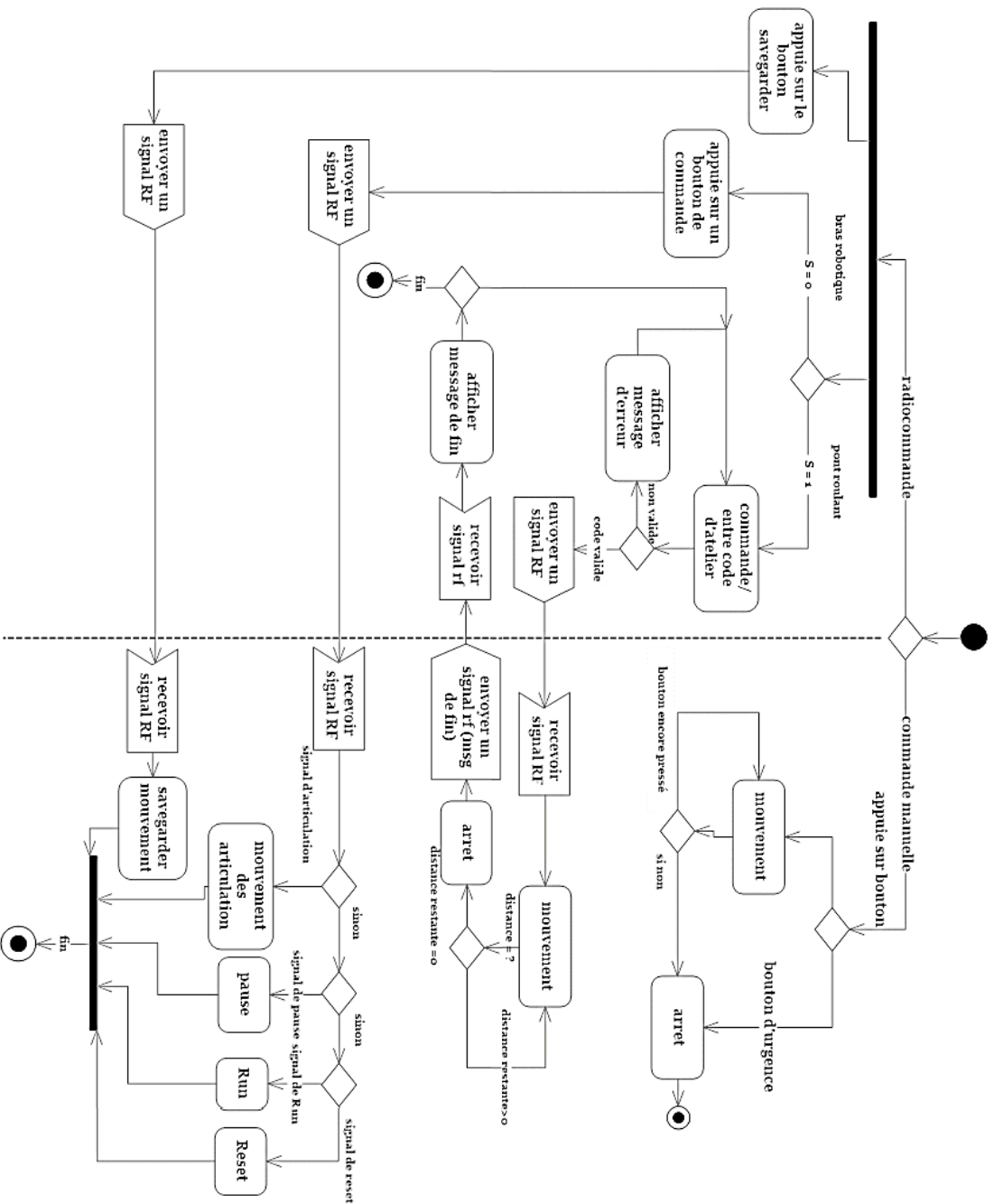


Fig. IV.II.9 Diagramme d'activité du pont.

IV.II.6 Résultats et discussion

La manipulation du pont roulant se fait quand le switch est en position 1.

Pour vérifier le bon fonctionnement de notre simulation on fait quelques tests sur l'environnement de simulation ISIS Proteus. La figure IV.34 montre l'affichage du message « Welcome » et demande ensuite d'entrer le code d'atelier (FigIV.35).

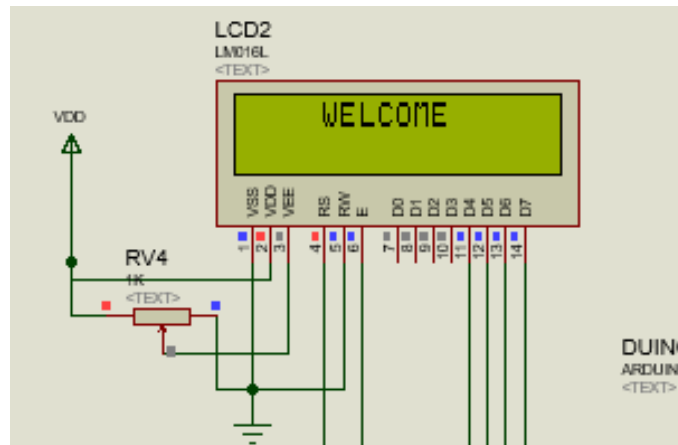


Fig.IV.II.10 Affichage du message « Welcome ».

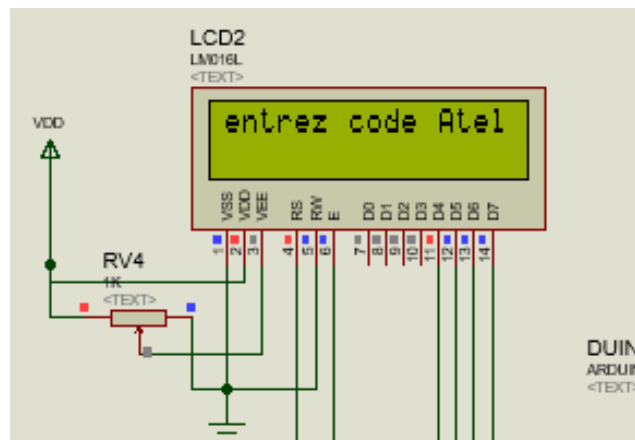


Fig.IV.II.11 Message de demande d'entrer le code d'atelier.

Après l'envoi du code d'atelier, le microcontrôleur traite le signal reçu et commence l'exécution de la requête.

La figure (Fig IV.II.12) montre le fonctionnement des moteurs X et Y pour déplacer le pont, les capteurs de distances indiquent la position d'arriver à l'atelier ciblé.

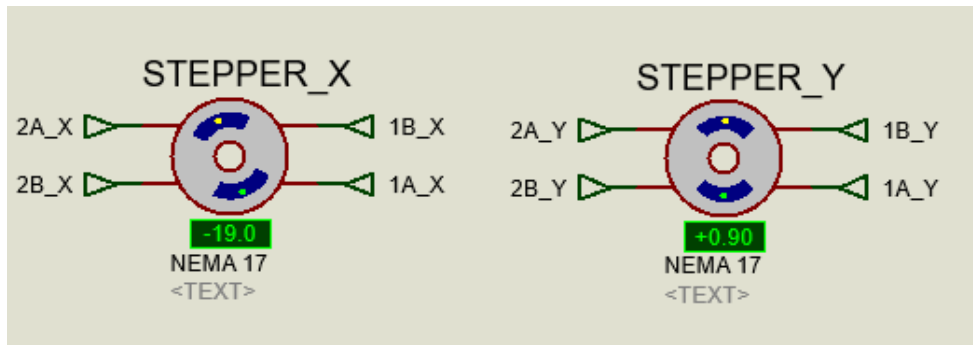


Fig.IV.II.12 Fonctionnement des moteurs X et Y.

Quand le chariot arrive à l’atelier ciblé un message sera envoyé à l’utilisateur pour indiquer son arrivage figure (Fig.IV.II.13).

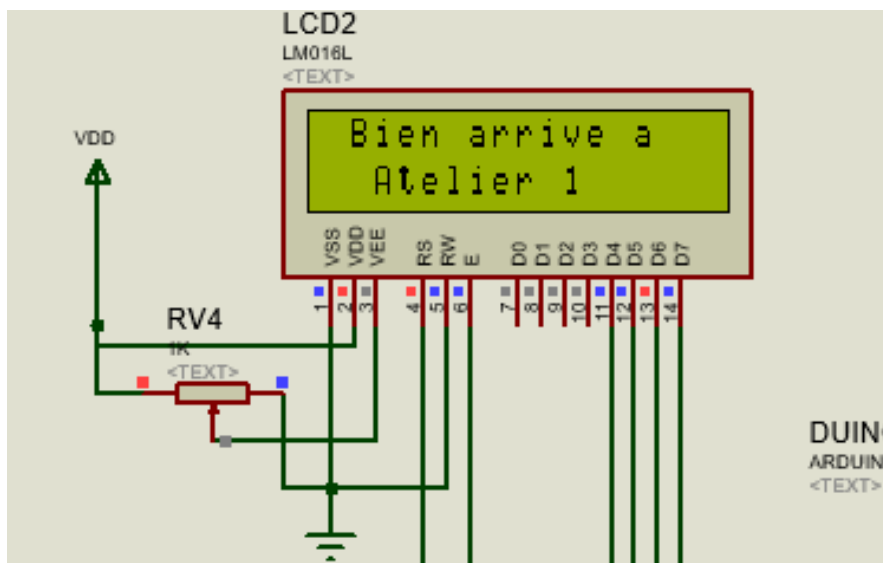


Fig.IV.II.13 Message d'arrivage.

La manipulation du bras robotique se fait quand le switch est en position 0, à l’état initial chaque servo moteur a un degré initial (Fig.IV.II.14).

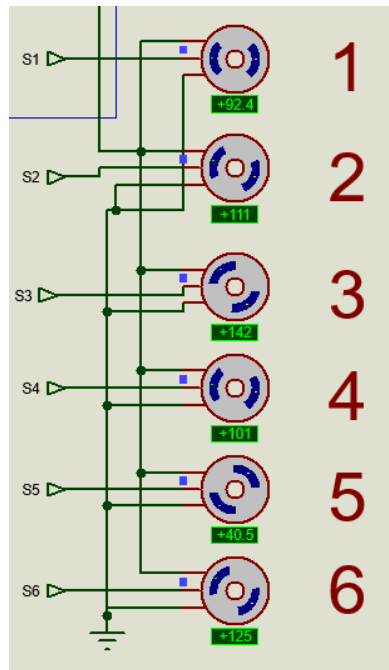


Fig.IV.II.14 Etat initial de chaque servo.

Chaque servo moteur contrôle une articulation (degré de liberté), l'utilisateur peut tourner le servo à gauche et à droite à partir de la boîte de commande radiofréquence. On prend en exemple le servo de la pince (FigIV.II.15).

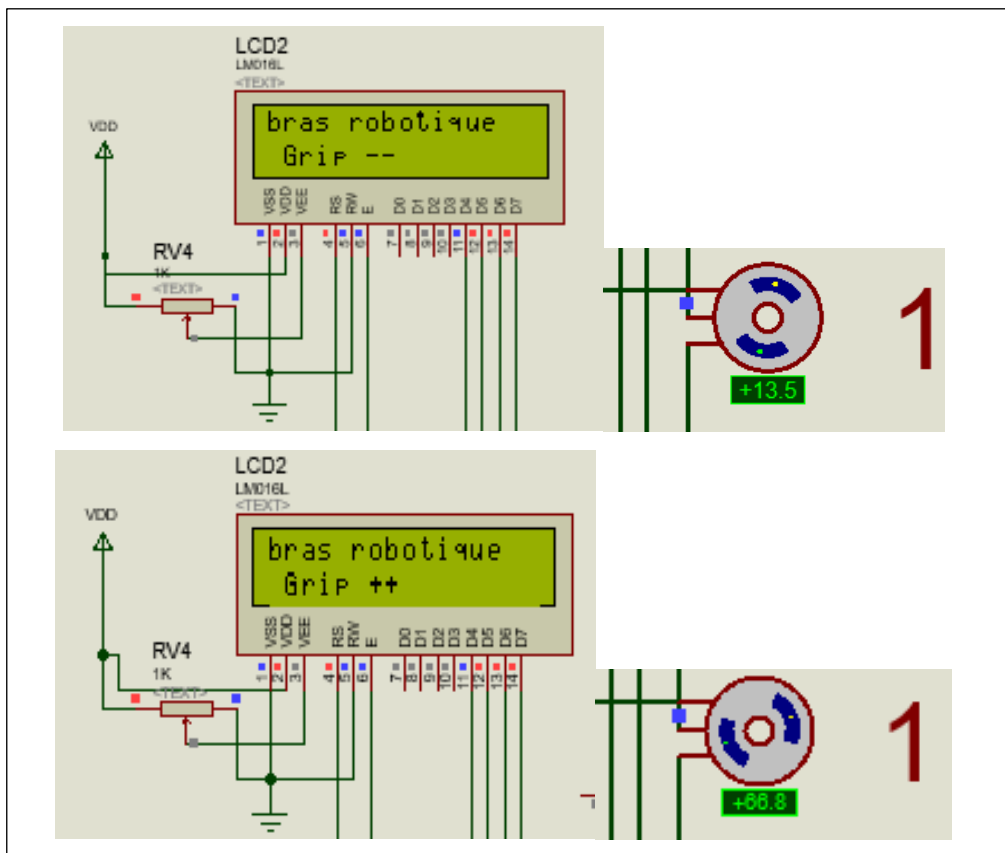


Fig.IV.II. 15 Exemple de manipulation.

Conclusion

On a conclu de cette partie que le choix des bibliothèques d'Arduino joue un rôle très important dans l'amélioration des performances du matériel, ainsi que la modélisation avec le diagramme d'activité permet de mieux visualiser les schémas de procédés et d'identifier les points à améliorer.

Références bibliographiques

- [1] Site web : « www.labcenter.com », une description générale de logiciel « Proteuse », consulté le 12/7/2020.
- [2] Le datasheet de : « *ArduinoMega* ».
- [3] Site web : « <http://www.elektronique.fr/logiciels/proteus.php> », consulté le 08/06/2020.
- [4] Le datasheet de microcontrôleur : « *Atmega2560* ».
- [5] Site web : « <https://www.pololu.com/product/2128> , » consulté le 27/06/2020.
- [6] Le datasheet de driver : « *A4988* ».
- [7] Le datasheet de : « *Nema 17 Stepper Motor* ».
- [8] Le datasheet de : « *capteur ultrason HCSR04* ».
- [9] Site web : « <https://www.carnetdumaker.net/articles/communiquer-sans-fil-en-433mhz-avec-la-bibliotheque-virtualwire-et-une-carte-arduino-genuino/> », consulté le 08/08/2020.
- [10] Le datasheet de : « *433Mhz_RF-TX&RX* ».
- [11] Site web : « <https://lastminuteengineers.com/433mhz-rf-wireless-arduino-tutorial/> », consulté le 08/08/2020.
- [12] Le datasheet de : « *LCD display Hitachi ADM1602K* ».
- [13] Le datasheet de : « *keypad- smallclac (4*4)* ».
- [14] Site web : « <http://www.solidworks.in> », site officiel de « Solidworks », consulté le 08/07/2020.
- [15] Site web : « <https://www.arduino.cc> », site officiel de « Arduino », consulté le 08/08/2020.
- [16] Site web : « <https://www.arduino.cc/en/reference/SPI> », site officiel de « Arduino », consulté le 27/06/2020.
- [17] Site web : « <https://www.arduino.cc/en/Reference/LiquidCrystal> », site officiel de « Arduino », article sur « LiquidCrystal », consulté le 18/08/2020.
- [18] Site web : « <https://www.geeksforgeeks.org/unified-modeling-language-uml-activity-diagrams/> », consulté le 15/08/2020.

[19] Site web : « <https://www.lucidchart.com/pages/fr/diagramme-dactivite/> », consulté le 28/07/2020.

[20] Site web : « circuitdigest.com/article/servo-motor-basics », consulté le 22/08/2020.

[21] Site web : « [https://engineering.eckovation.com/servo-motor-types-working-principle-explained/.](https://engineering.eckovation.com/servo-motor-types-working-principle-explained/) », consulté le 03/07/2020.

[22] Le datasheet de : « *MG996R Servo* ».

[23] Le datasheet de : « *SG90 Micro servo* ».

[24] M. Germain, les réseaux de téléphonie mobile. Un livre blanc de Forum ATENA, France (2014).

Conclusion générale

*L'*amélioration du fonctionnement du pont roulant a été une de nos préoccupations en ramenant de nouvelles technologies. Celles-ci donneront un rendement performant et épargnera les risques pouvant se produire au cours des opérations. Le résultat obtenu par l'automatisation, fournira un gain du temps.

Une étude sur les structures des ponts s'avère importante en les reliant avec une technologie récente qui consistera à l'ajout d'un bras manipulateur, ainsi nous obtiendrons une rapidité et une sécurité dans le travail. Parlant de sécurité, l'apparition de la radiofréquence reste l'une des solutions les plus sûres. Ce qui permettra de contrôler à distance en toute efficacité le fonctionnement du pont.

La stabilité et la rapidité sont assurées par l'étude qui consiste à convertir le modèle réel du pont et du bras manipulateur en un modèle mathématique. La vitesse du pont sera constante lors de l'opération pendant que le bras manipulateur sera commandé à distance en toute flexibilité avec un enregistrement des mouvements. Cette phase permettra de faciliter les tâches du pont roulant.

Nous avons établi un schéma général de montage, puis étudié les propriétés des éléments électroniques nécessaires pour la réalisation du prototype qui serviront pour la simulation du mouvement du système (pont et bras) en utilisant le logiciel ISIS. Nous avons pu visionner les schémas grâce au diagramme d'activité

Notre apport de cette nouvelle technologie semble prometteur car il offre des perspectives plus rassurantes en rendement et sécurité. Ce qui donnera plus de performances aux industries concernées.