

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE

Ministère de l'Enseignement Supérieur et de la Recherche

Scientifique

Université A. MIRA – BEJAIA

Faculté de Technologie

Département de Génie électrique



MEMOIRE DE FIN D'ETUDE

En vue de l'obtention du diplôme de Master en Electrotechnique

SPECIALITE

Automatismes industriels

Thème

Réalisation d'un environnement de modélisation, commande et simulation virtuel des robots manipulateurs

Présenté par :

Mr. AOUCHENNI Mourad

Mr. BENMOUSSA Larbi

Encadré par :

Mr. MELAHI Ahmed

Promotion 2019/2020

Année universitaire 2019-2020

Remerciements

*Notre remerciement s'adresse en premier lieu à Allah le tout
Puissant pour la volonté, la santé et la patience qu'il nous a donnée
Durant toutes ces longues années.*

*Ainsi, nous tenons également à exprimer nos vifs remerciements à
notre promoteur Mr A.MELAHI qui nous a apporté une aide
précieuse nous lui exprimons notre gratitude pour sa grande
disponibilité ainsi pour sa compréhension et les encouragements
qu'il nous a apportés...*

*Nous exprimons aussi notre sincère reconnaissance au personnel
des deux sociétés DPR AXXAM, SARL IBRAHIM et fils plus
particulièrement à Mr A.AMGHAR et Mr H.BOUCHEMALE pour
leurs aides, leurs disponibilités et leurs orientations tout au long de
nos stages.*

*Enfin, nous tenons à exprimés notre reconnaissances à tous nos
amis et collègues pour le soutien moral et matériels...*

Dédicaces

*Je tiens à dédié ce modeste travail
A mes cher Parents qui n'ont pas cessé de prier pour
moi et qui m'ont aidés durant tout la durée de mes
études, que dieu les garde pour nous.*

*A mon frère et ma sœur qui m'ont fourni tous leurs
effort et moyen pour que je termine mes études.*

A mes très chers amis.

*A mon binôme BENMOUSSA Larbi avec qui j'ai
partagé les plus beaux moments ainsi que toute sa
famille.*

*A ceux que j'ai la chance de connaitre, dans les
meilleures et les pires moments de ma vie, à mes amis
les plus fidèles.*

MOURAD.

Dédicaces

Je tiens à dédié ce modeste travail

A mes cher Parents qui n'ont pas cessé de prier pour moi et qui m'ont aidés durant tout la durée de mes études, que dieu les garde pour nous.

A mes frères et ma sœur qui m'ont fourni tous leurs effort et moyen pour que je termine mes études.

A mes très chers amis.

A mon binôme AOUCHENNI Mourad avec qui j'ai partagé les plus beaux moments ainsi que toute sa famille.

A ceux que j'ai la chance de connaitre, dans les meilleures et les pires moments de ma vie, à mes amis les plus fidèles.

LARBI.

Sommaire

Introduction générale

Chapitre 1 : Langages de programmation des robots.

1.1 Introduction	1
1.2 Définitions et historique.....	1
1.2.1 Définition d'un robot.....	1
1.2.2 Définition d'un algorithme.....	1
1.2.3 Définition d'un langage de programmation	1
1.2.3 Historique	1
1.3 Différents types des robots	2
1.3.1 Les robots cylindriques	2
1.3.2 Les robots sphériques	3
1.3.3 Les robots Cartésiens	3
1.3.4 Les robots parallèles.....	4
1.3.5 Les robots anthropomorphes	4
1.4 Critères de choix d'un robot	5
1.5 Structure mécanique d'un robot.....	6
1.5.1 Structure d'un robot industriel	6
1.6 Programmation d'un robot industriel.....	6
1.6.1 Langages de programmation des robots.....	7
1.6.2 Types de langage de robots	7
1.7 Le langage VAL.....	9
1.8 Conclusion.....	10

Chapitre II : Modélisation et commande des robots manipulateurs.

2.1 Introduction	12
2.2 Définition d'un robot manipulateur	12
2.3 Composition d'un robot manipulateur	12

2.4 Les Degrés de liberté d'un robot manipulateur	14
2.6 Les types d'analyse de la modélisation des robots	15
2.7 Introduction à la commande des systèmes robotisés.....	16
2.8 La conception des lois de commande.....	16
2.9 Commande des bras manipulateurs	17
2.10 Commande adaptative des robots manipulateurs.....	17
2.11 Commande jacobienne.....	19
2.12 Commande prédictive	19
2.13 Commande robuste	19
2.14 Commande optimale	20
2.15 Conclusion.....	20

Chapitre III : Mini langage de programmation et application sur un système robotisé

3.1 Introduction	22
3.2 Historique.....	22
3.3 Définition de la chaîne de production automatisée	22
3.4 Les Systèmes de production industrielle	23
3.4.1 Le fonctionnement des systèmes automatisés	23
3.4.2 Objectifs de l'automatisation	23
3.5 Décomposition d'une chaîne de production en chaînes fonctionnelles.....	24
3.5.1 La Partie opérative :	24
3.5.2 La partie commande :	24
3.5.3 La partie dialogue ou relation :	25
3.5.4 Notion de chaîne.....	25
3.5.4.1 Chaîne fonctionnelle :	25
3.5.4.2 Chaîne d'action :	25

3.5.4.3 Chaîne d'acquisition.....	26
3.6 Les éléments de la PC/PO d'une chaîne de production	26
3.6.1 Effecteurs :	26
3.6.2 Les convoyeurs :.....	26
3.6.3 Les actionneurs :.....	26
3.6.3.1 Les moteurs :	26
3.6.3.2 Les vérins :.....	26
3.6.4 Les Pré actionneurs :	26
3.6.4.1 Les Capteurs :	26
3.6.4.2 Contacteur :.....	27
3.6.4.3 Distributeur :.....	27
3.7 Virtuel Realm Builder.....	27
3.7.1 Présentation du logiciel	27
3.7.2 Installation du logiciel.....	27
3.8 Notre maquette	28
3.8.1 La plateforme	28
3.8.2 Les convoyeurs.....	31
3.8.3 Le robot	33
3.8.4 La maquette final.....	34
3.9 Simulink 3D toolbox.....	34
3.10 Modèle Simulink.....	35
3.11 Fonctions	37
3.12 Les instructions.....	38
3.13 Matlab GUIDE	39
3.13.1 Ouverture du GUIDE	39
3.14 Pupitre	41
3.15 Conclusion.....	42

Conclusion générale

Références bibliographique

Liste des Figures

Chapitre I : Langages de programmation des robots.

Figure 1.1 Le robot cylindrique.....	2
Figure 1.2 Robot Seiko.....	2
Figure 1.3 Le robot sphérique.	3
Figure 1.4 Robot Fanuc (sphérique).....	3
Figure 1.5 Robot Cartésien.....	3
Figure 1.6 Le robot Toshiba.	4
Figure 1.7 Robot Festo.	4
Figure 1.8 Robot anthropomorphe.	4
Figure 1.9 Robot Fanuc (anthropomorphe).....	5
Figure 1.10 Robot ABB.	5

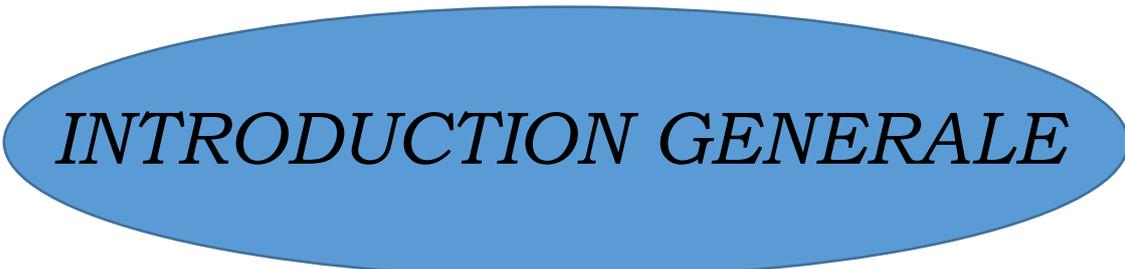
Chapitre II : Modélisation et commande des robots manipulateurs.

Figure 2.1 Nomenclature pour l'analyse d'un robot manipulateur.	11
Figure 2.2 Joints à un degré de liberté.....	12
Figure 2.3 Prototype de robot manipulateur avec trois degrés de liberté.....	13
Figure 2.4 Manipulateurs planaires avec des joints rotatifs.....	13

Figure 2.5 Quatre grands domaines de base de l'analyse de robots manipulateurs.....	14
Figure 2.6 une boucle fermée d'un système robot commandé.....	15
Figure 2.7 Schéma général de la commande adaptatif.....	17

Chapitre III : Mini langage de programmation et application sur un système robotisé

Figure 3.1 Structure d'un système automatisé.....	24
Figure 3.2 Chaîne d'action.....	25
Figure 3.3 Chaîne d'acquisition.....	26
Figure 3.4 interface principale de V-Realm.....	27
Figure 3.5 nouveau monde virtuel.....	28
Figure 3.6 Création d'objet.....	29
Figure 3.7 configurer la taille	30
Figure 3.8 choisir une texture	30
Figure 3.9 ajuster la luminosité.....	31
Figure 3.10 Le convoyeur	31
Figure 3.11 composition du convoyeur	32
Figure 3.12 Le robot	33
Figure 3.13 robot PUMA 560.....	33
Figure 3.14 la maquette finale	34
Figure 3.15 contenue de la bibliothèque 3D Animation.....	34
Figure 3.16 Modèle Simulink	35
Figure 3.17 sous-système grab	36
Figure 3.18 les fonctions.....	37
Figure 3.19 les instructions... ..	38
Figure 3.20 GUIDE Matlab... ..	39
Figure 3.21 guide par défaut.. ..	40
Figure 3.22 le pupitre.....	41



INTRODUCTION GENERALE

Introduction générale

L'être humain dans sa nature, a toujours recherché le moyen d'optimiser ses efforts. Il n'a jamais cessé de mettre son intelligence et son imagination au service de ce but et ceci afin de créer un partenaire qui fera le travail à sa place. De nos jours, les robots sont devenus des outils nécessaires dans l'environnement de fabrication, permettant des tâches à automatiser s'étendant du plus répétitif au plus complexe. Pendant que la technologie avance, la robotique s'impose de plus en plus dans l'industrie moderne. Actuellement, les chercheurs parviennent à concevoir des robots légers et rapides.

Le but de notre travail consiste à la réalisation d'un langage de programmation permettant de programmer une maquette virtuelle 3D d'une chaîne de production robotisée qui se compose de trois stations (perçage, déplacement et lissage).

Dans le premier chapitre nous avons présenté de manière générale la robotique, les différents types et l'utilisation des robots, ainsi les différents langages de programmation en expliquant quelques instructions essentiels du langage Victor's Assembly Language (VAL).

Le deuxième chapitre est dédié à la présentation des différentes méthodes de commande des robots manipulateurs et les modèles utilisés pour décrire le mouvement des articulations d'un bras manipulateur, pour les robots, la commande et la modélisation ont un rôle très important dans la conception, elles permettent au robot d'atteindre un état désiré ou d'exécuter une tâche bien spécifique avec une grande précision dans un environnement réel.

L'étape finale sera la réalisation de chaînes de production automatisée. La maquette de cette chaîne est réalisée avec le logiciel V-Realm Builder et la simulation avec Matlab sous Simulink 3D Animation. Ensuite, la création d'un mini langage de programmation pour commander les trois stations, et pour finir nous allons réaliser un pupitre de commande avec le GUIDE de Matlab.

Chapitre 1:
Langages de programmation des robots.

1.1 Introduction

Depuis la naissance de l'industrie robotisée, les robots industriels ont été conçus afin de remplacer les humains dans des tâches répétitives, fatigantes et qui impliquent souvent un travail manuel dangereux. Actuellement, grâce à l'augmentation de la production et à la diminution des prix, les robots industriels sont utilisés dans une grande variété d'applications [1].

Dans ce premier chapitre, nous présentons quelques généralités sur le domaine de la robotique et son histoire dans le domaine industriel. En effet, nous présenterons les différents langages de programmation d'un robot ainsi que les différents types des robots et on va terminer ce chapitre par la présentation du langage VAL.

1.2 Définitions et historique

1.2.1 Définition d'un robot

Un robot est un dispositif mécanique poly-articulé mu par des actionneurs et commandé par un contrôleur accomplissant automatiquement une grande variété de tâches qui sont généralement considérées comme dangereuses, pénibles, répétitives et impossibles pour les humains [1].

1.2.2 Définition d'un algorithme

Lorsqu'on fait de la programmation de robots, on met notamment au point des algorithmes. Un algorithme est une suite d'opérations simples permettant ainsi de résoudre un problème. Nous suivons une suite d'instructions simples dans un ordre précis. En cela, l'algorithme peut être décrit comme une méthode qui permet de parvenir à un résultat en réalisant une suite d'actions dans un ordre bien précis.

1.2.3 Définition d'un langage de programmation

Pour que les instructions données soient comprises par un robot programmable, il faut utiliser un langage compris par ce robot. C'est le langage de programmation de ce robot.

Il existe plusieurs langages de programmation de robots et tous suivent la même logique dans laquelle interviendront des variables, des conditions ou encore des fonctions.

1.2.3 Historique

Dans ce qui suit, on présente un bref historique sur l'utilisation des robots industriels [2] :

- 1947 : Premier manipulateur électrique télé-opéré.

- 1954 : Premier robot programmable.
- 1961 : Utilisation d'un robot industriel, commercialisé par la société UNIMATION (USA), sur une chaîne de montage de General Motors.
- 1961 : Premier robot avec contrôle en effort.
- 1963 : Utilisation de la vision pour commander un robot.
- 1973 : Premier robot mobile à roues.
- 1980/1990: robots mobiles (chenilles, roues, pattes).
- 1990: télémanipulation avec bras à retour d'effort.
- 1990/2000: robots nettoyeurs, sous-marins d'intervention, pilotes de dirigeables (Alpha).

1.3 Différents types des robots

Plusieurs types de robots sont proposés selon l'objectif de l'utilisation. Dans ce qui suit, on va vous présenter quelques types les plus utilisés.

1.3.1 Les robots cylindriques

Les robots cylindriques possèdent un espace de travail cylindrique. Ils sont très rapides. La figure suivante présente un schéma où on peut voir les différents types d'articulation (rotative à la base et linéaire ensuite) [1].

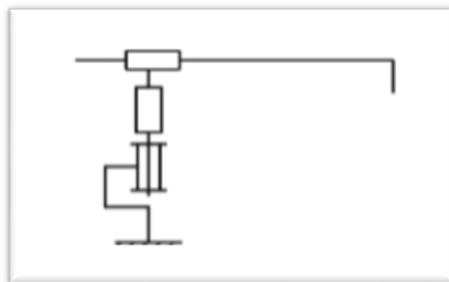


Figure 1.1 Le robot cylindrique.



Figure 1.2 Robot Seiko.

1.3.2 Les robots sphériques

Ce sont des robots spécifiés par leur espace de travail sphérique en plus d'une grande charge utile [1].

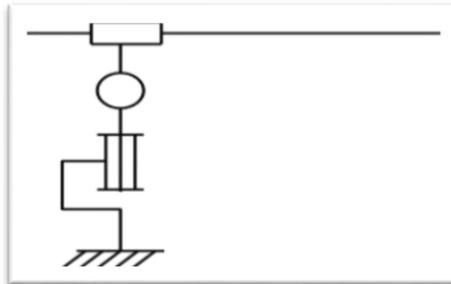


Figure 1.3 Le robot sphérique.



Figure 1.4 Robot Fanuc (sphérique).

1.3.3 Les robots Cartésiens

Les robots cartésiens ont une très bonne précision mais ils sont lents [1].

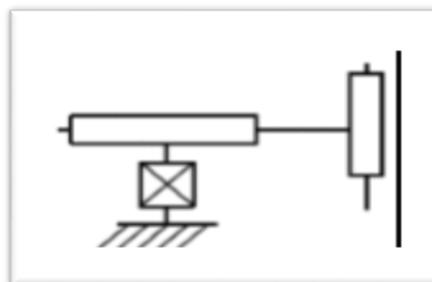


Figure 1.5 Robot Cartésien.



Figure 1.6 Le robot Toshiba.

1.3.4 Les robots parallèles

Les robots parallèles ce sont des robots caractérisés par un espace de travail réduit mais ils sont précis (grande rigidité de la structure) et rapides [1].

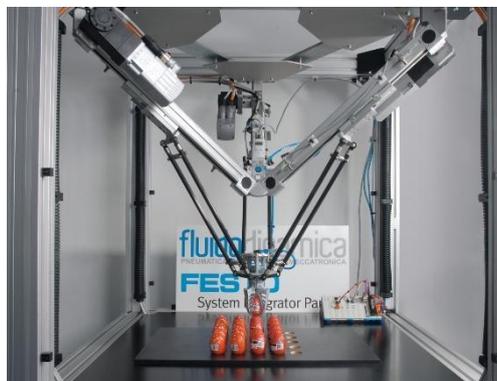


Figure 1.7 Robot Festo.

1.3.5 Les robots anthropomorphes

Ces robots reproduisent la structure d'un bras humain en ayant 6 axes en série (6 articulations rotatives). Leurs architecture est standard [1].

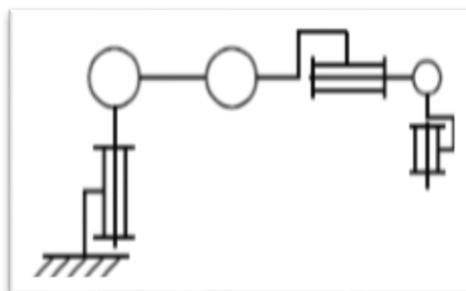


Figure 1.8 Robot anthropomorphe.

Comme l'exemple de robot anthropomorphe, on cite le robot Fanuc et le robot ABB [1].



Figure 1.9 Robot Fanuc (anthropomorphe).



Figure 1.10 Robot ABB.

1.4 Critères de choix d'un robot

Un robot doit être choisi en fonction de l'application qu'on lui réserve. Voici quelques paramètres à prendre [3].

- La charge maximum transportable (de quelques kilos à quelques tonnes), à déterminer dans les conditions les plus défavorables (en élongation maximum).
- L'architecture de la structure mécanique articulée, le choix est guidé par la tâche à réaliser.
- Le volume de travail, est accessible par l'outil du robot qui peut balayer l'outil. Le volume dépend de la géométrie du robot, de la longueur des segments et du débattement des articulations.
- Le positionnement absolu, correspondant à l'erreur entre un point souhaité défini par une position et une orientation dans l'espace cartésien – et le point atteint et calculé via le modèle géométrique inverse du robot. Cette erreur est due au modèle utilisé, à la quantification de la mesure de position, à la flexibilité du système mécanique. En

général, l'erreur de positionnement absolu, également appelée précision, est de l'ordre de 1 mm.

- La répétabilité d'un robot est l'erreur maximale de positionnement répétée de l'outil en tout point de son espace de travail. En général, la répétabilité < 0.1 mm.
- La vitesse de déplacement (vitesse maximum en élongation maximum), et l'accélération.
- La masse du robot.
- Le coût du robot.
- La maintenance.

1.5 Structure mécanique d'un robot

Le robot industriel est essentiellement un système mécanique articulé qui peut positionner et orienter un outil porté par son extrémité, de manière quelconque et désirée, dans un certain volume qu'on nomme espace de travail du robot. Cet espace est celui qui est balayé par les articulations [7].

Les articulations peuvent être disposées en série (robot série, qui peut évoquer la structure d'un bras humain) ou en parallèle (robot parallèle). Le plus souvent, ce squelette possède six degrés de liberté : les trois premiers permettent le positionnement et constituent le « porteur » du robot ; les trois derniers forment l'organe terminal du robot et donnent la faculté d'orientation [7].

Le porteur peut présenter plusieurs géométries en fonction du choix de la nature des articulations (translation ou rotation) [7].

L'organe terminal porte l'outil de travail qui dépend des tâches que l'on désire réaliser. Il s'agit souvent de pinces à deux ou trois doigts que l'on nomme préhenseurs. La plupart du temps, ces préhenseurs sont amovibles et peuvent être remplacés à la demande par d'autres outils (perceuse, ponceuse...) [7].

1.5.1 Structure d'un robot industriel

Ce sont des robots programmables sur trois axes ou plus, qui peuvent être fixés sur place ou mobiles, destinés à être utilisés dans des applications d'automatisation industrielle.

Le robot industriel inclut le manipulateur, actionneurs et le système de commande.

1.6 Programmation d'un robot industriel

Un programme peut être défini comme un chemin de l'espace à suivre par les robots, combiné avec des actions périphériques qui soutiennent le cycle de travail.

Un robot peut effectuer des tâches complexes sous contrôle des programmes stockés, qui peuvent être modifiés à volonté.

Un robot est programmé en entrant des commandes dans sa mémoire. La programmation peut être en ligne, hors ligne.

1.6.1 Langages de programmation des robots

Tous les robots sont programmés avec des langages de programmation. Ces langages de programmation sont utilisés pour commander le robot en utilisant les signaux d'e/s.

Le langage de programmation est ce qui donne à un robot sa flexibilité et permet à un robot de réagir à son environnement (par l'utilisation de capteurs), et de prendre des décisions.

Les langages de programmation des robots se répartissent en trois catégories de base :

- Langage spécialisé pour robot, VAL, AL, HELP...
- Bibliothèque de robots pour un nouveau langage polyvalent, KAREL...
- Bibliothèque des robots pour un langage d'informatique existant.

1.6.2 Types de langage de robots

Certains des langages informatiques de haut niveau actuellement sont utilisés pour programmer les robots. Parmi ces langages, on trouve :

Wave : C'est le premier langage évolué créé pour robot, développé par Stanford Artificial Laboratory en 1973.

AL (Arm Language) : Il est développé par le Centre de Recherche en Robotique de l'Université de Stanford.

AML : C'est le langage utilisé pour contrôler les robots IBM.

APT (Automatically Programmed Tools) : C'est un langage machine développé par Electronic Systems Laboratory en 1965.

ARCL (A robot Control Language) : Il se base sur une syntaxe proche du PASCAL.

ZDRL (Zhe Da Robot Language) : C'était un langage orienté mouvement. C'était un système interprétatif composé de 32 commandes système et de 37 instructions.

HELP : C'est un langage haut niveau développé pour les robots d'assemblage ALLEGRO de General Electric.

Karel : Karel, Karel 2, Karel 3 sont des langages utilisés pour quelques robots de FANUC.

ACL (Advanced Command Language) : C'est un langage multitâches avancé développé par Eshed Robotec (1982) Ltd. ACL est programmé dans une série de EPROM à l'intérieur d'un Controller-B, et peut être accéder par n'importe quelle terminal standard ou PC via l'utilisation d'un câble de communication RS232.

CAP1 (Conversational Auto Programming 1) : Il est le langage utilisé pour les robots FANUC 32-18-T.

MML : C'est un langage de haut niveau pour la programmation hors ligne, développé par California University.

RIPL (Robot Independent Programming Language) : Il se base sur un environnement de programmation orienté objet **RIPE (Robot Independent Programming Environment)**.

MCL (Manufacturing Control Language) : Il est développé pour le projet ICAM de l'Air Force Américaine par McDonnell Douglas.

RAIL : C'est un langage de programmation haut niveau développé par Automatrix.

RPL : C'est un langage de programmation de haut niveau, développé par SRI et utilisé pour configurer les systèmes industriels automatisés.

ARMBASIC : C'est une extension du langage BASIC utilisé pour le Microbot Mini-Mover 5.

Androtext : C'est un langage de haut niveau développé par Robotronic Corporation.

VAL (Victor's Assembly Language) : C'est un langage de programmation de haut niveau développé pour les robots PUMA. Il possède un ensemble complet de mots pour écrire et étudier des programmes de robots.

IBL (Instruction Based Learning) : C'est une méthode d'entraînement de robot en utilisant des instructions en langage naturel. Un robot est équipé d'une liste d'instructions basiques tel que tourner à gauche ou avancer. Les instructions verbales de l'utilisateur sont converties en une nouvelle procédure qui sera ajoutée aux connaissances du robot ce qui le rendra capable d'exécuter des tâches de plus en plus complexes.

1.7 Le langage VAL

VAL est le langage commercial le plus avancé et développé pour les robots d'animation, Il s'agit essentiellement d'un langage hors ligne dont lequel le programme définissant la séquence de mouvements peut être développé sans l'implication du robot, ceci donne au programmeur une plus grande flexibilité pour accomplir des tâches plus complexe en un temps plus réduit [8].

Comme exemple de tâches d'utilisation du langage VAL, on peut citer [8] ;

1. Avancer à un endroit au-dessus de la pièce dans la goulotte ;
2. Avancer vers la pièce ;
3. Fermer les mâchoires de la pince ;
4. Retirer la pièce de la goulotte ;
5. Transporter la pièce à un emplacement au-dessus de la boîte ;
6. Mettre la pièce dans la boîte ;
7. Ouvrir les mâchoires de la pince ;
8. Sortir de la boîte ;

Le programme VAL correspondant est le suivant [8] :

1. ? APPRIO PART, 50 @
2. ? MOVES PART @
3. ? CLOSE I @
4. ? DEPARTS 150@
5. ? APPROX BOX, 200@
6. ? MOVE BOX@
7. ? OPENI @
8. ? DEPART@

La signification exacte de chaque ligne est : [8]

1. Se placer à un emplacement situé à 50 mm au-dessus de la pièce dans la goulotte.
2. Se déplacer le long d'une ligne droite jusqu'à la pièce.
3. Fermer les mâchoires de la pince.
4. Retirer la pièce à 150 mm de la goulotte le long d'une trajectoire rectiligne.
5. Se déplacer en ligne droite jusqu'à un emplacement situé à 200 mm au-dessus de la boîte.
6. Mettre la pièce dans la boîte.
7. Ouvrir les mâchoires de la pince.
8. Se retirer de 75 mm de la boîte.

1.8 Conclusion

Le présent chapitre est une description du domaine de robotique en mettant l'accent sur les différents types des robots et les critères de choix d'un robot industriel. On a aussi cité les différents langages de programmation d'un robot tout en détaillant sur le langage VAL.

Chapitre 2 :
Modélisation et commande des robots manipulateurs.

2.1 Introduction

Les robots manipulateurs sont de nos jours une partie essentielle dans l'industrie, ils permettent d'automatiser des systèmes allant du plus répétitif au plus complexe.

Ce deuxième chapitre est dédié à l'analyse et la modélisation des robots manipulateurs, ainsi qu'à la commande des bras manipulateurs par quelques techniques de commandes conventionnelles fréquemment employées dans le but de présenter un état de l'art dans ce domaine et de discuter les différentes approches proposées dans la littérature.

2.2 Définition d'un robot manipulateur

L'Association Française de Normalisation (A.F.N.O.R.) définit un robot comme étant un système mécanique de type manipulateur commandé en position, reprogrammable, polyvalent (i.e., à usages multiples), à plusieurs degrés de liberté, capable de manipuler des matériaux, des pièces et des outils, au cours des mouvements variables et programmés pour l'exécution d'une variété de tâches. Il a souvent l'apparence d'un, ou plusieurs, bras se terminant par un poignet. Son unité de commande utilise, notamment, un dispositif de mémoire et éventuellement de perception et d'adaptation à l'environnement et aux circonstances. Ces machines polyvalentes sont généralement étudiées pour effectuer la même fonction de façon cyclique et peuvent être adaptées à d'autres fonctions sans modification permanente du matériel [4].

2.3 Composition d'un robot manipulateur

Les robots manipulateurs correspondent à des bras artificiels, généralement plusieurs liens rigides connectés par des articulations. La grande majorité des robots industriels sont des robots manipulateurs qui ont comme tâche de positionner des outils pour la soudure, la peinture, etc. La figure 2.1 présente les composants principaux d'un robot manipulateur. La figure 2.3 illustre des exemples concrets de mécanismes et composants pour un robot manipulateur prototype [5].

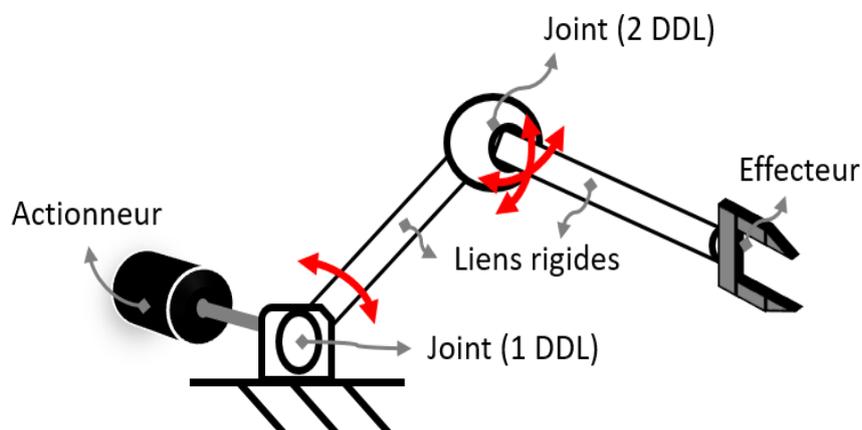


Figure 2.1 Nomenclature pour l'analyse d'un robot manipulateur.

- **Axes :** Un lien rigide d'un robot, c'est un ensemble de pièces fixes les unes par rapport aux autres qui forment un corps rigide [5].

- **Joints (articulations) :** Les joints sont les articulations d'un robot. Un joint prismatique (articulation linéaire), permet la translation selon un axe entre deux pièces, et contraint les rotations relatives. Un articulateur rotatif, permet à deux pièces de pivoter relativement selon un axe. La variable q_i utilisée pour décrire la configuration d'un joint i est une distance pour un joint prismatique et un angle pour un joint rotatif [5].

Joint prismatique : $q_i = x_i$ [m]

Joint rotatif : $q_i = \theta_i$ [rad]

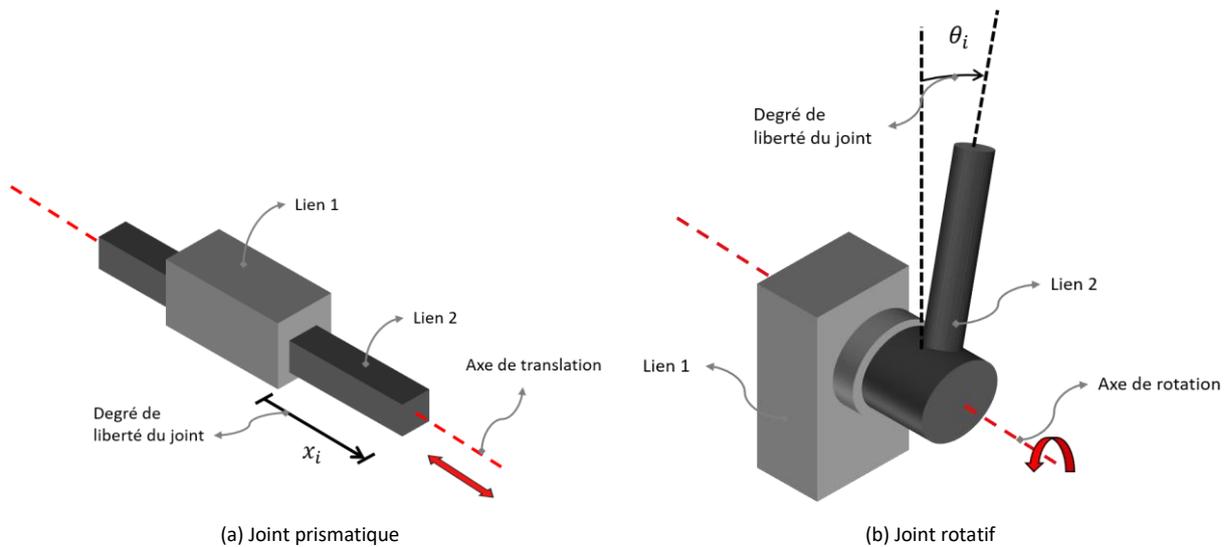


Figure 2.2 : Axes à un degré de liberté.

- **Effecteur :** L'effecteur est un terme générique qui se réfère à la position de l'outil d'un robot manipulateur (ex. : une pince, une tête de soudure, une caméra, etc.) [5].
- **Actionneurs :** Les actionneurs sont les muscles des bras robotiques. Un actionneur est fondamentalement un dispositif qui transforme de l'énergie en travail mécanique. La plupart des robots industriels utilisent des moteurs électriques pour actionner leurs joints, toutefois les actionneurs pourraient aussi être des vérins pneumatiques ou hydrauliques [5].
- **Capteurs :** Les capteurs sont les dispositifs qui collectent de l'information sur l'état interne d'un robot et/ou l'environnement. Pour la cinématique, les capteurs très régulièrement utilisés sont les encodeurs, qui mesurent directement la position des joints (variables q_i), et les systèmes de visions, qui mesurent la position de points dans le repère de la caméra [5].

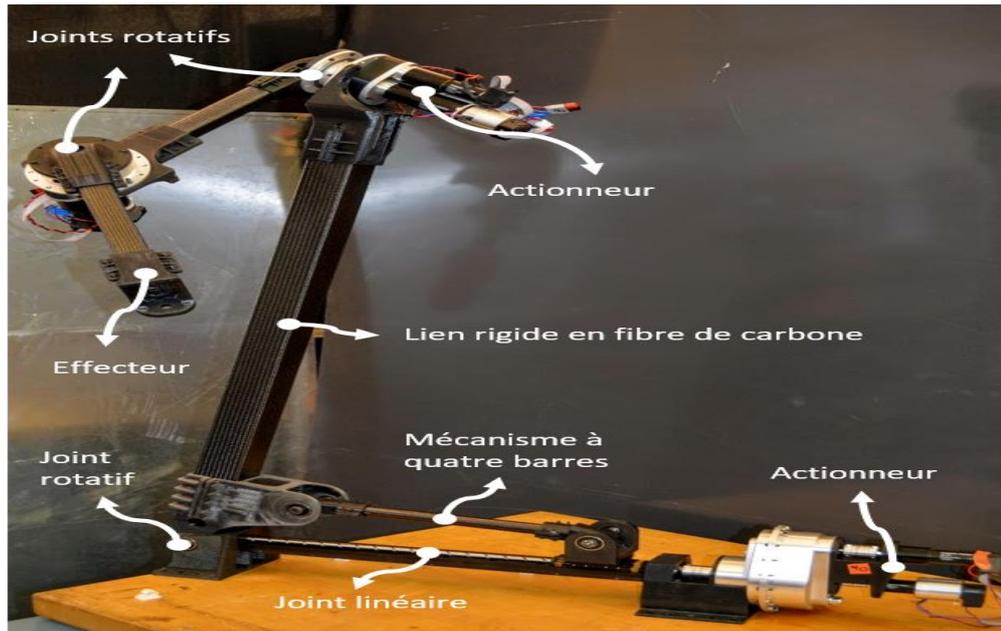


Figure 2.3 Prototype de robot manipulateur avec trois degrés de liberté [5].

2.4 Les Degrés de liberté d'un robot manipulateur

Le nombre de degrés de liberté (DDL), c'est le nombre de variables nécessaires pour décrire la configuration d'un robot. La plupart des robots industriels ont six DDL. Pour des robots qui utilisent seulement des axes rotatifs comme illustré à la figure 2.4, leurs nombre de DDL est égale au nombre de joints. Si le nombre de DDL est supérieur à six, le robot est dit redondant.

- Un robot redondant a plusieurs options de configuration des joints pour atteindre une position et une orientation désirées de l'effecteur [5].

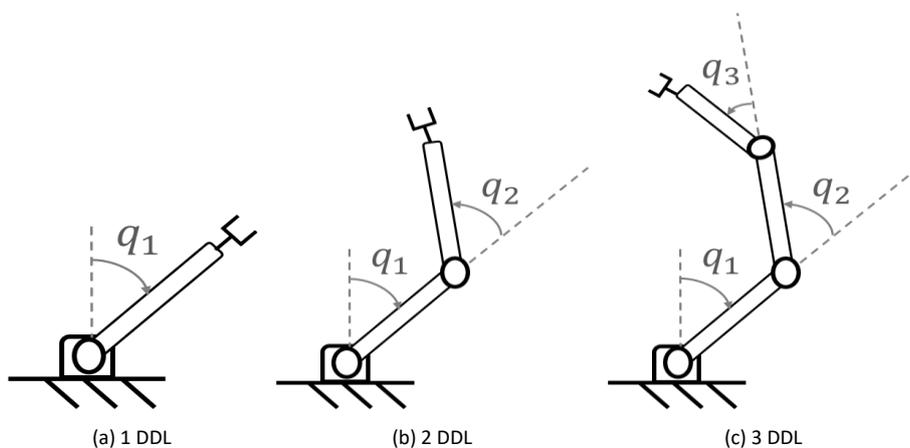


Figure 2.4 : Manipulateurs planaires avec des joints rotatifs.

2.5 Pourquoi modéliser et analyser des robots ?

Modéliser un système robotisé est essentiel lors de la phase de conception, par exemple pour valider qu'une géométrie de bras permet d'atteindre plusieurs positions désirées avec son effecteur (i.e. l'outil au bout du bras), et aussi lors de la programmation pour coordonner les différents moteurs et articulations d'un robot.

2.6 Les types d'analyse de la modélisation des robots

Comme illustré à la figure 2.1, la modélisation des robots peut être séparée en quatre modèles [5].

- Le modèle géométrique direct vise à calculer la position finale de l'effecteur d'un robot en fonction du positionnement des articulations. Lorsqu'on parle de cinématique inverse, l'objectif est de déterminer le positionnement des articulations qui mène à une position désirée de l'effecteur du robot.
- Le modèle géométrique différentielle vise à calculer la vitesse de l'effecteur du robot en fonction de la vitesse des moteurs qui actionnent les articulations.
- les analyses statiques visent généralement à déterminer les forces/couples nécessaires aux moteurs/actionneurs d'un robot pour le maintenir en place en fonction des forces externes.
- Le modèle dynamique est le domaine qui vise à déterminer les équations différentielles qui représentent la relation entre l'accélération des joints d'un robot et les forces appliquées [5].

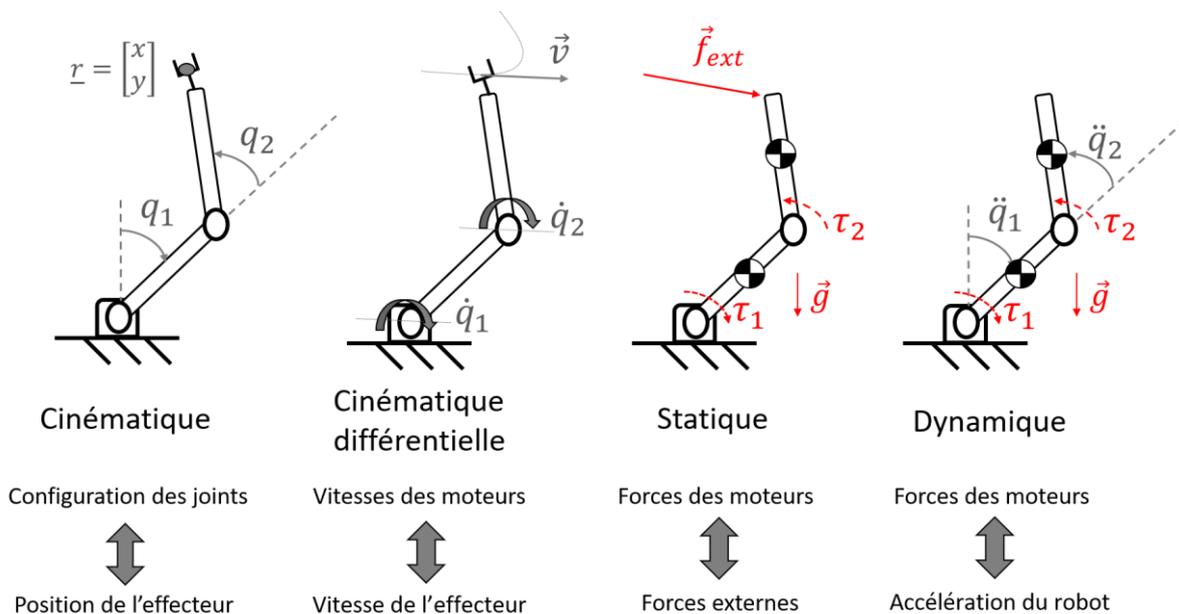


Figure 2.5 Quatre grands domaines de base de l'analyse de robots manipulateurs.

2.7 Introduction à la commande des systèmes robotisés

La commande traite du développement d'algorithmes qui déterminent les actions qu'un système robotisé doit prendre en fonction de la situation pour atteindre un objectif (par exemple une position cible). Ces algorithmes sont généralement implémentés sur un microcontrôleur, qui choisit les actions des actionneurs (par exemple la force des moteurs électriques) [4].

La partie commande synthétise les consignes des asservissements pilotant les actionneurs, à partir de la fonction de perception et des ordres de l'utilisateur, s'ajoutent à cela [4] :

- L'interface homme-machine à travers laquelle l'utilisateur programme les tâches que le robot doit exécuter,
- Le poste de travail, ou l'environnement dans lequel évolue le robot.
- On distingue 4 parties principales dans un robot manipulateur :

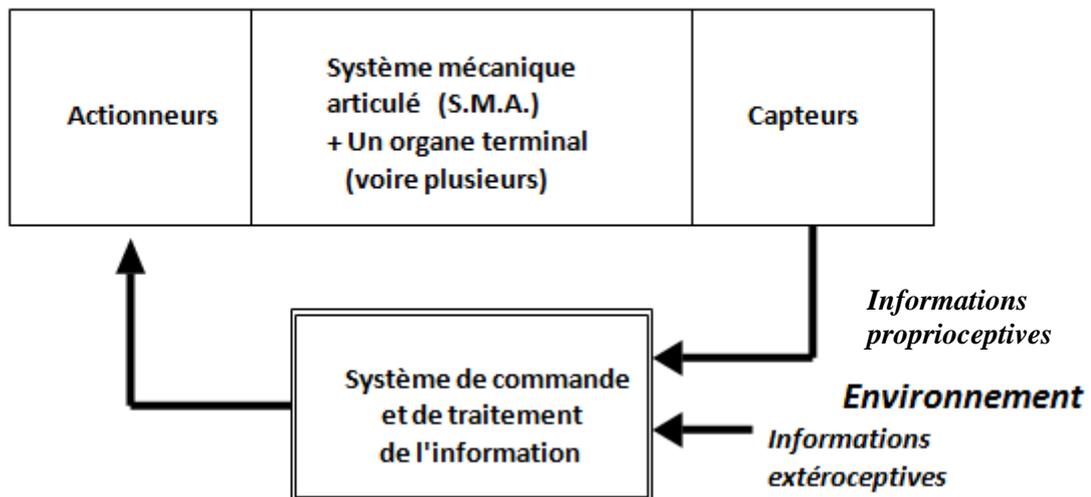


Figure 2.6 Une boucle fermée d'un robot commandé.

2.8 La conception des lois de commande

Lors de la conception des lois de commandes, plusieurs aspects peuvent être analysés selon le contexte [6] :

- La stabilité : Lorsqu'une loi de commande est proposée, un critère essentiel dans la plupart des situations est la stabilité du système. La stabilité caractérise si les états du système convergent vers les états désirés ou bien s'ils divergent [6].
- La faisabilité : Dans certains contextes, principalement ceux où les actions possibles sont limitées (ex. couple maximum des moteurs), on s'intéresse tout simplement à savoir s'il

existe une séquence d'actions possibles qui peuvent permettre au système d'atteindre l'objectif [6].

- Le temps de réponse, la stabilité et la faisabilité détermine si le système pourra atteindre l'objectif. Ensuite, on cherche généralement à atteindre l'objectif le plus rapidement possible ce qui est caractérisé par le temps de réponse [6].
- L'optimalité : Il y a généralement un compromis à faire entre la vitesse (le temps de réponse) et l'utilisation agressive des actions possibles du système qui ont généralement un coût énergétique. Un objectif commun lors de la conception d'un asservissement est de maximiser la performance. La maximisation de la performance dans le domaine de la commande est généralement formulée comme la minimisation d'un coût qui combine les dépenses énergétiques et le temps passé loin de l'objectif [6].
- La robustesse : Finalement, dans plusieurs situations il faut considérer l'incertitude dans le système lors de la conception des lois de commande. L'incertitude peut être due à des perturbations externes inconnues (ex. : rafales de vent pour un avion) ou bien à des variations des paramètres du système robotique (ex. : poids de la charge transportée pour un bras manipulateur). L'analyse de robustesse consiste à vérifier si les lois de commandes vont fonctionner dans divers scénarios possibles. Des méthodes existent pour garantir le fonctionnement d'un contrôleur malgré un certain niveau d'incertitude (commande robuste ou commande adaptative) [6].

2.9 Commande des bras manipulateurs

Le problème de la commande d'un robot manipulateur peut être formulé comme la détermination de l'évolution des forces généralisées (forces ou couples) que les actionneurs doivent exercer pour garantir l'exécution de la tâche tout en satisfaisant certains critères de performance. Différentes techniques sont utilisées pour la commande des bras manipulateurs. La structure mécanique du bras manipulateur a une influence sur le choix du schéma de commande. Un robot manipulateur est une structure mécanique complexe dont les inerties par rapport aux axes des articulations varient non seulement en fonction de la charge mais aussi en fonction de la configuration, des vitesses et des accélérations. [6]

Deux types de mouvements apparaissent quand on parle de commande du bras manipulateur. Un premier type considère que les mouvements nécessaires pour la réalisation de la tâche sont exécutés dans l'espace libre. Le deuxième type considère des mouvements spécifiques avec des forces de contact pour l'organe terminal qui se déplace dans un espace contraint. Toute tâche de robotique est réalisée par une combinaison de ces deux types de mouvement. Pour simplifier la commande, les deux types de mouvement sont abordés séparément [6].

2.10 Commande adaptative des robots manipulateurs

Le modèle dynamique d'un robot ne pouvant être parfait, il est nécessaire que les ingénieurs appliquent les théories de la commande adaptative pour estimer ou ajuster en ligne les valeurs de paramètres dynamiques utilisés dans la commande. La commande adaptative est

un système adaptatif mesure un index de performance (IP) en utilisant les entrées et les sorties d'un système ajustable. Cet index est comparé à un ensemble de valeurs fixées. A partir de cette comparaison, le mécanisme d'adaptation modifier les paramètres du système et engendre une entrée auxiliaire afin de maintenir la valeur de l'index proche des valeurs de référence, on désigne cinq stratégies, significatives des progrès réalisés dans ce domaine [6] :

1. Simplification du modèle dynamique.
2. Application des techniques adaptatives développées pour les systèmes linéaires au cas des robots.
3. Proposition d'une commande adaptative non linéaire asymptotiquement stable.
4. Utilisation des propriétés de la passivité pour développer une loi de commande adaptative non linéaire asymptotiquement stable.
5. Utilisation pour l'adaptation d'un modèle qui n'est pas fonction des accélérations articulaires tel que le modèle dynamique filtré ou le modèle énergétique.

Le schéma général d'un système de commande adaptatif peut être le suivant :

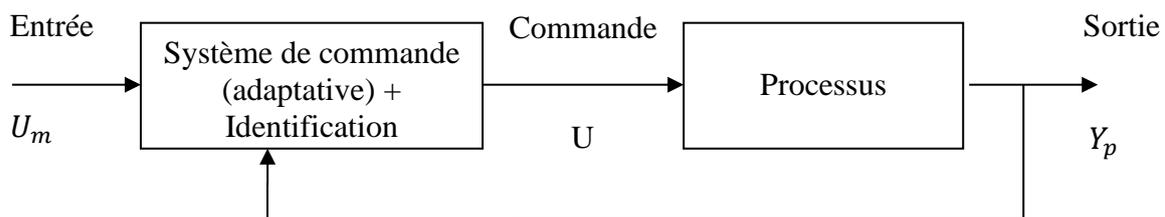


Figure 2.7 Schéma général de la commande adaptative [6].

Ce type de techniques vise à corriger les déficiences de la commande par découplage non linéaire, comme la connaissance approximative des paramètres du modèle du robot ou pour s'adapter aux différentes conditions opératoires. Ce type de schémas cherche à estimer ou ajuster en ligne les valeurs des paramètres utilisées dans le calcul de la loi de commande. Un travail plus intéressant sur ce sujet est la commande proposé par la commande de Slotine-Li ou commande adaptative passive [6].

Les avantages de ce type de techniques sont évidents, malheureusement la puissance de calcul demandée au système constitue un inconvénient important [6].

2.10.1 Inconvénient de la commande adaptative

Le problème de la conception de lois de commande adaptative pour des robots manipulateurs est d'assurer le suivi de trajectoire désirée. Le développement de contrôleurs adaptatifs efficaces représente une étape importante vers les applications robotiques à grande vitesse et précision, même dans une installation industrielle bien structurée, les robots peuvent faire face à une incertitude concernant les paramètres décrivant les propriétés dynamiques de

la charge de portée. Puisque ces paramètres sont difficiles de calculer ou mesurer, ils limitent le potentiel pour des robots pour manipuler avec précision des objets de taille et de poids considérable. Pour compenser cette incertitude paramétrique, beaucoup de chercheurs ont proposé des stratégies de commande adaptatives. Un avantage de l'approche adaptative est que l'exactitude des charges inconnues de transport d'un manipulateur s'améliore avec le temps [6].

L'utilisation des techniques adaptatives est motivée par l'ajustement automatique des paramètres du système quand ils sont inconnus et ont des variations imprévisibles dans le temps [6].

Les contrôleurs adaptatifs sont formulés en séparant les paramètres constants inconnus des fonctions connues dans l'équation dynamique de robot [6].

2.11 Commande jacobienne

Cette technique est utilisée depuis 1969 et elle est appelée de cette façon lorsqu'elle utilise la matrice jacobienne inverse du bras manipulateur pour calculer les vitesses de consigne aux articulations. Elle est aussi connue sous le nom de commande à mouvement résolu. Les approches les plus courantes sont de type [6] :

- Mouvement à vitesse résolu
- Mouvement à accélération résolu
- Mouvement à force résolu

La technique de mouvement résolu commande la position de l'organe terminal du manipulateur dans l'espace cartésien, par combinaison des mouvements de plusieurs articulations ou la vitesse calculée aux articulations sont utilisées directement comme des consignes pour les boucles de commande à chaque articulation [6].

2.12 Commande prédictive

Ce type de commande, en utilisant le modèle du système et les consignes, est capable de prédire son évolution, de telle manière qu'il est capable d'agir en fonction de l'erreur de prédiction. Un grand avantage de ce type d'approche est lié au fait que l'erreur de prédiction n'est pas contaminé par les bruits de mesure mais la dépendance au modèle reste forte [6].

2.13 Commande robuste

Dans le cas de paramètres fixes, il est connu que la technique de découplage non linéaire peut devenir instable en présence d'incertitudes. Si les paramètres du modèle ne sont pas connus de façon précise et si l'incertitude sur les paramètres admet des bornes connus, alors les techniques de commande robuste peuvent être utilisées. Par exemple, les travaux de Slotine [Slotine, 1985] considèrent la technique de mode glissant appelé aussi commande de structure variable. Cette technique utilise une surface de glissement où la stabilité du système est garantie [6].

2.14 Commande optimale

Pour réaliser une tâche, il peut exister un grand nombre de solutions. Dans ce cas, il peut être souhaitable de choisir une solution qui satisfasse un certain critère. La littérature présente différents types de critères pour la commande optimale : la commande en temps minimal du domaine des neurosciences. Maximiser la souplesse du mouvement et la minimisation du couple, entre autres. La complexité du problème de commande optimale a motivé les chercheurs pour diviser la tâche en deux étapes : la première étape est la planification de trajectoire suivie d'une étape d'asservissement de la trajectoire [6].

2.15 Conclusion

Dans ce deuxième chapitre nous avons présenté en première partie les différents composants d'un robot manipulateurs dans le but de les analyser et donner les étapes à suivre pour les modéliser, en deuxième partie on a expliqué la commande adaptative en donnant ses avantages et ses inconvénients, ainsi on a présenté quelques autres méthodes de commande.

Chapitre 3:

Mini langage de programmation et application sur un système robotisé

3.1 Introduction

Dans ce chapitre on va présenter la chaîne de production réalisée et le mini langage permettant sa programmation ainsi que le pupitre de commande.

Pour la simulation de la maquette virtuelle sous Matlab on va utiliser un modèle Simulink. Simulink 3D Animation permet d'établir une liaison entre les modèles Simulink et les algorithmes Matlab avec des objets graphique 3D. Simulink 3D Animation inclus des éditeurs et des visionneuses pour interagir avec les objets 3D.

3.2 Historique

La première chaîne de production industrielle fût inventée par Ransome Eli Olds en 1900.

Cette chaîne servait à produire l'automobile de marque Oldsmobile à Détroit au Michigan, à cette époque, il fallait plusieurs heures pour construire une automobile, quelques années plus tard (1913-14), Henry Ford apporta une grande amélioration à la chaîne de production en y introduisant le convoyeur de cette façon, l'automobile se déplaçait de station de travail en station automatique, ou les travailleurs assemblaient les pièces. Le temps de production fût ainsi réduit à quatre-vingt-treize minutes pour le modèle de Ford. Aujourd'hui, la production d'automobiles est encore plus rapide grâce à l'utilisation de robots dans les chaînes de production. [10]

3.3 Définition de la chaîne de production automatisée

Une chaîne de production est l'ensemble d'opérations de fabrication nécessaires à la réalisation d'un produit, qui passe consécutivement d'une opération à l'autre et étapes par étapes suivant un trajet bien défini grâce à un déroulement logique de processus jusqu'à ce que le produit soit fini. [9]

3.4 Les Systèmes de production industrielle

Un système de production industriel “SPI” est un ensemble d'équipements qui permet, à partir d'énergie et des produits bruts (matériaux, pièces initiales, ...) d'élaborer des objets de valeur supérieure qui peuvent être soit directement commercialisés ou des produits intermédiaires servant à la réalisation par la suite des produits finis. Le SPI est alimenté en énergies (électrique, pneumatique, ...) et approvisionné en consommables auxiliaires : lubrifiants, eau de refroidissement, etc. Il génère aussi un flux de déchets : eaux sales, fumées polluantes, chutes de coupes, ... D'autre part, l'exploitation et le fonctionnement du système de production industriel nécessitent l'intervention du personnel. [11]

3.4.1 Le fonctionnement des systèmes automatisés

Les systèmes automatisés ont des fonctions précises : Ils exécutent des tâches afin d'obtenir un résultat prévu à l'avance.

- Les tâches : Un système automatisé réalise un certain nombre d'actions appelées « Tâches ».
- Les effecteurs : L'élément du système automatisé qui produit un effet est appelé « l'effecteur ».
- Les cycles : Un système automatisé accomplit une suite d'opérations (appelé cycle) depuis un état initial jusqu'à un état final.

3.4.2 Objectifs de l'automatisation

Un système automatisé doit dans une certaine mesure reconstituer tous les mouvements dont un homme a besoin pour mener un travail bien déterminé. On devra, dès sa conception, tenir compte des contraintes de maintenance. Dans l'industrie, la production de grandes séries sont partiellement ou entièrement automatisées. Le produit est donc fabriqué sur une succession de poste de travail robotisés et ne demande qu'une intervention humaine limitée. Parmi ces objectifs on a : [9]

- L'amélioration de la compétitivité du produit face à la concurrence.
- La réduction des coûts de production.
- L'augmentation de la qualité du produit liée à la précision des actionneurs.
- Economies de matière ou d'énergie.

- Alléger les tâches physiques non valorisantes pour l'homme.

3.5 Décomposition d'une chaîne de production en chaînes fonctionnelles

D'une façon tout à fait générale, un Système Automatisé "SA" est composé de trois parties:

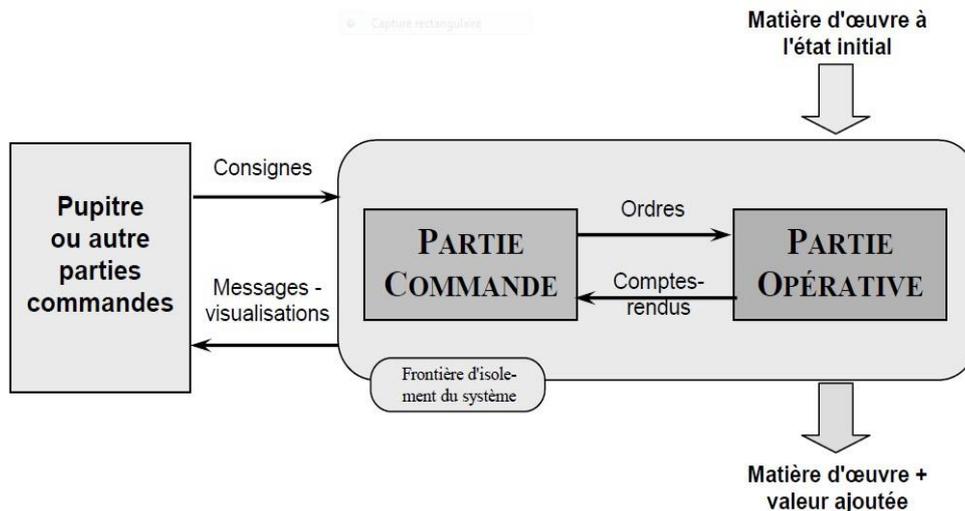


Figure 3.1 Structure d'un système automatisé.

3.5.1 La Partie opérative :

La Partie Opérative "PO" est le processus physique à automatiser. Elle opère sur la matière et les produits entrants en transformant, par exemple, des pièces brutes en pièces usinées ou en effectuant des mouvements de translation d'une cage d'ascenseur de l'étage de départ à celui d'arrivée. Les opérations sont obtenues lorsque les ordres donnés par la Partie Commande sont exécutés dans de bonnes conditions. La "PO" comporte en général, d'une part, les outillages et moyens divers mettant en œuvre le processus d'élaboration et d'autre part, les différents actionneurs destinés à mouvoir ou à mettre en œuvre ces moyens. [9]

La partie opérative se compose de trois ensembles :

- Les capteurs
- Les actionneurs
- Les prés actionneurs

3.5.2 La partie commande :

La partie commande est un automatisme qui élabore en sortie des ordres destinées aux processus et des signaux de visualisation en fonction des comptes rendus venant de processus et des consignes qu'il reçoit en entrée. En d'autre terme, c'est l'ensemble des moyens de traitement de l'information qui assurent le pilotage et la coordination des tâches des processus

souhaite. Elle traite les informations du dialogue avec le milieu extérieur et d'autres parties commandent. Elle coordonne les actions de la partie opérative. [9]

La partie commande se décompose en trois ensembles :

- Les interfaces d'entrées : Les interfaces d'entrées transforment les informations issues des capteurs placés sur la partie opérative ou dans la partie dialogue en information de nature et d'amplitudes avec les caractéristiques technologiques de l'automate.
- Les interfaces de sorties : L'interface de sorties transforment les informations élaborées par l'unité de traitement en information de nature et d'amplitude compatible avec les caractéristique technologiques des pré actionneurs d'une part et des visualisations d'avertisseurs d'autre part.
- L'unité de traitement : L'unité de traitement élabore les ordres destinés aux actionneurs en fonction des informations reçues des différents capteurs et des fonctionnements à réaliser. La partie commande peut-être représenter dans un système automatisé soit par une automate programmable industrielle ou par une carte spécialisée.

3.5.3 La partie dialogue ou relation :

Le pupitre ou la partie relation est l'organe servant d'interface Homme Machine, l'opérateur envoie des consignes à l'unité de traitement et reçoit en retour des informations émanant de la partie commande (bouton, voyant, clavier, écran...).

3.5.4 Notion de chaîne

3.5.4.1 Chaîne fonctionnelle : Une chaîne fonctionnelle est un ensemble de constituants (capteurs, actionneurs et des effecteurs) organisés en vue de l'obtention d'une même tâche.

3.5.4.2 Chaîne d'action : La chaîne d'action est constituée du pré actionneurs qui distribuent la puissance aux actionneurs.

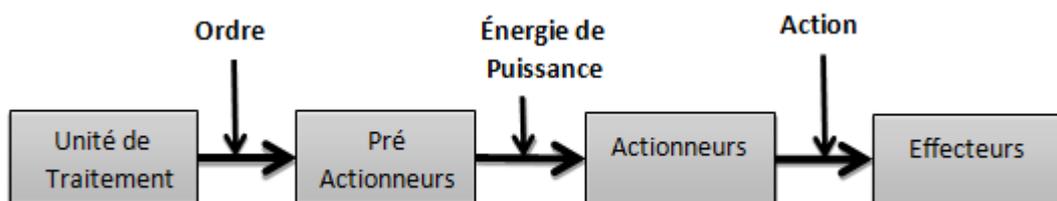


Figure 3.2 Chaîne d'action.

3.5.4.3 Chaîne d'acquisition : La chaîne d'acquisition est constituée des capteurs qui envoient des comptes rendus sur l'état de la P.O. à l'unité de traitement.

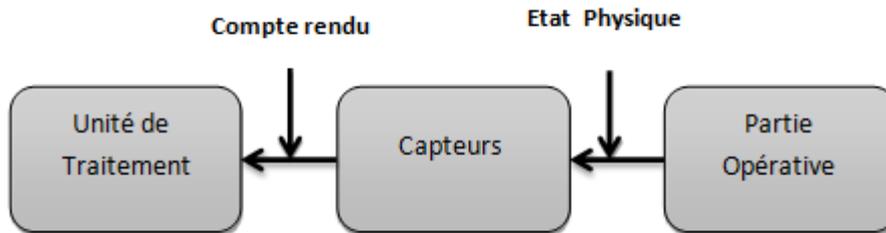


Figure 3.3 Chaîne d'acquisition.

3.6 Les éléments de la PC/PO d'une chaîne de production

3.6.1 Effecteurs : Les effecteurs sont multiples et variées et sont souvent conçus spécialement pour s'adapter à l'opération qu'ils ont à réaliser sur la matière d'œuvre, ils reçoivent leur énergie des actionneurs.

3.6.2 Les convoyeurs : Un convoyeur est un mécanisme qui sert à déplacer des objets le long d'une chaîne de production. Il est généralement muni d'un tapis roulant qui se déplace sur des roulements. Le tapis roulant est raccordé à un moteur électrique par l'entremise d'un train d'engrenage.

3.6.3 Les actionneurs : Convertissent l'énergie qu'ils reçoivent des pré actionneurs en une autre énergie utilisée par les effecteurs. Ils peuvent être pneumatiques, hydraulique ou électriques. Par exemple on a :

3.6.3.1 Les moteurs : L'entraînement en rotation d'éléments de machines est fréquemment réalisé par des moteurs électriques (moteur asynchrone, motoréducteur...), sa fonction est de convertir une puissance électrique fournie en puissance mécanique caractérisée par un couple et une vitesse de rotation sur son arbre.

3.6.3.2 Les vérins : Le vérin est un actionneur qui réalise un mouvement de translation, ou de rotation à partir d'une source d'énergie pneumatique ou hydraulique, et à chaque vérin on lui associe un distributeur qui assure la commutation de circuit.

3.6.4 Les Pré actionneurs : Les ordres émis par la partie commande sont généralement sous forme d'un signal électrique de faible puissance d'où la nécessité d'avoir recours à des organes chargés de traduire et d'amplifier les ordres émis par la partie commande, appelés pré actionneurs, le rôle général d'un pré actionneur est de distribuer l'énergie.

3.6.4.1 Les Capteurs : Un capteur est un dispositif électronique qui permet de détecter un état ou mesure une valeur et transmettre cette information à un ordinateur (microprocesseur), il existe plusieurs types de capteurs : capteur de contact et capteur de lumière renseignent la PC

sur l'état de la PO, ils peuvent détecter des positions, des pressions, des températures, des débits ..., peuvent être électrique ou pneumatique.

3.6.4.2 Contacteur : Est un pré-actionneur destiné à ouvrir ou fermer un circuit électrique par l'intermédiaire d'un circuit de commande. Il alimente le moteur électrique en énergie de puissance en fonction d'une consigne opérative issue de la partie commande.

3.6.4.3 Distributeur : Est chargé d'alimenter les vérins en énergie pneumatique ou hydraulique

3.7 Virtuel Realm Builder

3.7.1 Présentation du logiciel

V-Realm Builder est un logiciel utilisé pour concevoir des mondes virtuels et dessiner / importer des objets virtuels 3D. Une fois que l'utilisateur a construit son monde virtuel, il peut manipuler les objets et la scène virtuelle via les commandes MATLAB® et les modèles Simulink® pour animer la scène.

Les propriétés des objets 3D virtuels, telles que la translation, la rotation, l'échelle et la couleur peuvent être modifiée via les commandes MATLAB® et les modèles Simulink® [12].

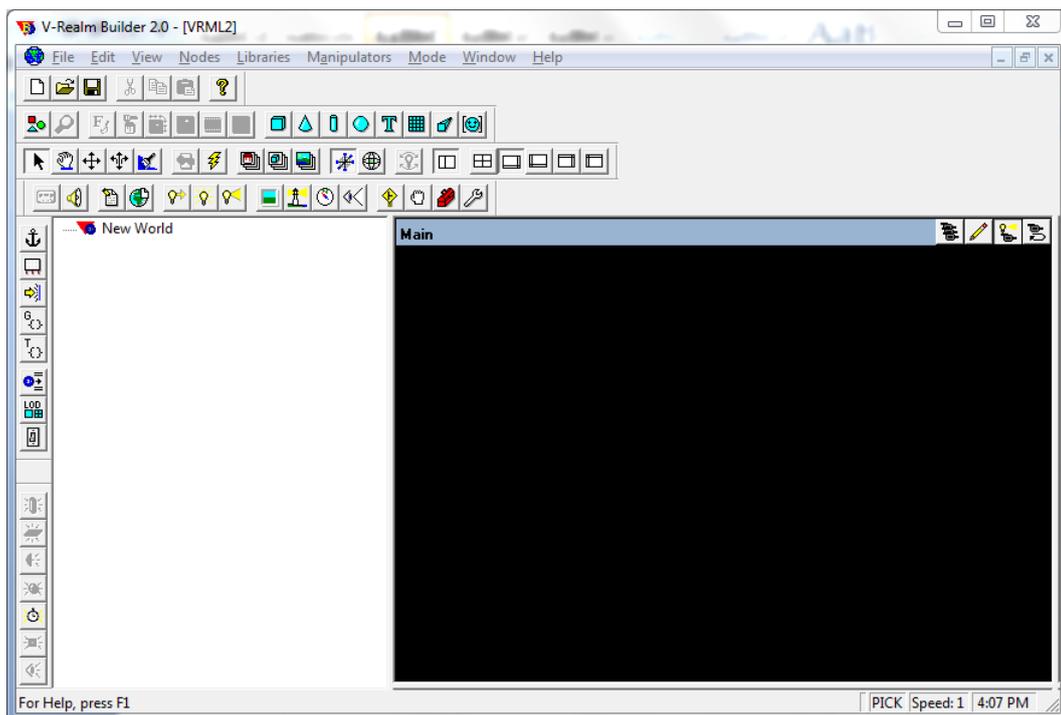


Figure 3.4 interface principale de V-Realm

3.7.2 Installation du logiciel

Pour installer le logiciel V-Realm Builder suivez ces étapes :

1. Sur votre bureau, cliquez avec le bouton droit sur l'icône MATLAB et sélectionnez Exécuter en tant qu'administrateur.
2. Dans la fenêtre de commande MATLAB, tapez

vrinstall –install

Ou tapez **vrinstall** ('- install')

La fenêtre de commande MATLAB affiche les messages suivants:

Starting editor installation... Done.

3. Tapez **vrinstall** Si l'installation de l'éditeur a réussi, la fenêtre de commande MATLAB affiche ce message: **Virtual World editor: installed**

4. Quittez MATLAB et redémarrez MATLAB.

5. Définissez l'éditeur par défaut sur V-Realm Builder. Dans MATLAB, entrez:

```
vrsetpref('Editor','*VREALM');
```

6. Pour ouvrir un fichier dans l'éditeur V-Realm, dans MATLAB, accédez à un fichier de monde virtuel, cliquez avec le bouton droit de la souris et sélectionnez Modifier.

Pour démarrer V-Realm Builder, l'utilisateur doit se rendre à l'emplacement où MATLAB® a été installé et accéder à l'emplacement de sous-répertoire suivant:

```
{Matlab installation folder}\toolbox\vr\vrealm\program [13]
```

3.8 Notre maquette

Dans cette partie nous allons vous expliquer comment crée des objets avec VRLM, tout en présentant notre maquette qui est une chaine de production robotisé.

3.8.1 La plateforme

Pour crée la plateforme suivez ces étapes :

1. Ouvrez le logiciel V-Realm Builder et cliquez sur **File>New** pour crée un nouveau monde virtuel.

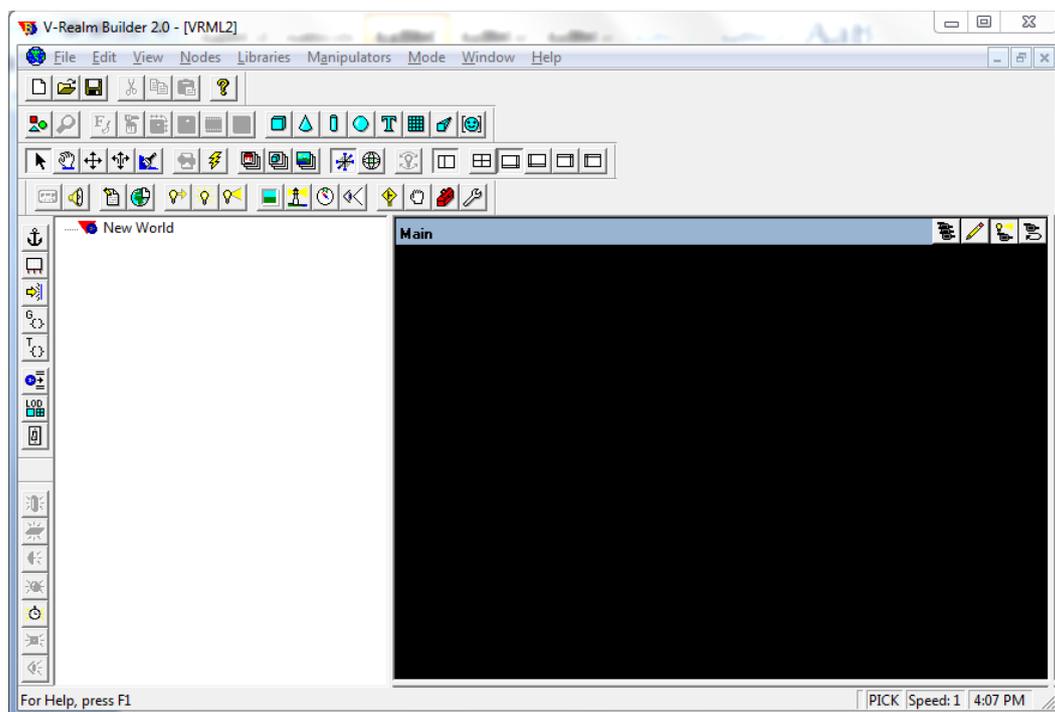


Figure 3.5 nouveau monde virtuel

2. Cliquez sur le bouton **Insert Background**, puis cliquez sur **Insert Box** pour ajouter un cube.

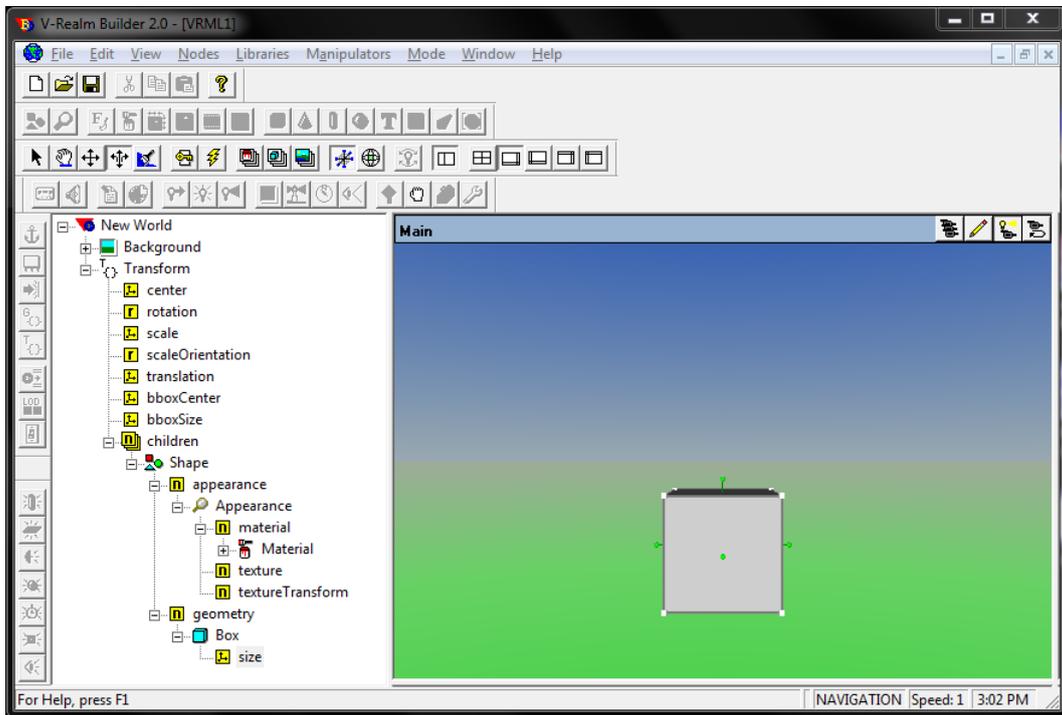


Figure 3.6 Création d'objet

3. Pour modifier la taille du cube double cliquez sur **size**,
X axis représente la largeur
Y axis représente la hauteur
Z axis représente la longueur

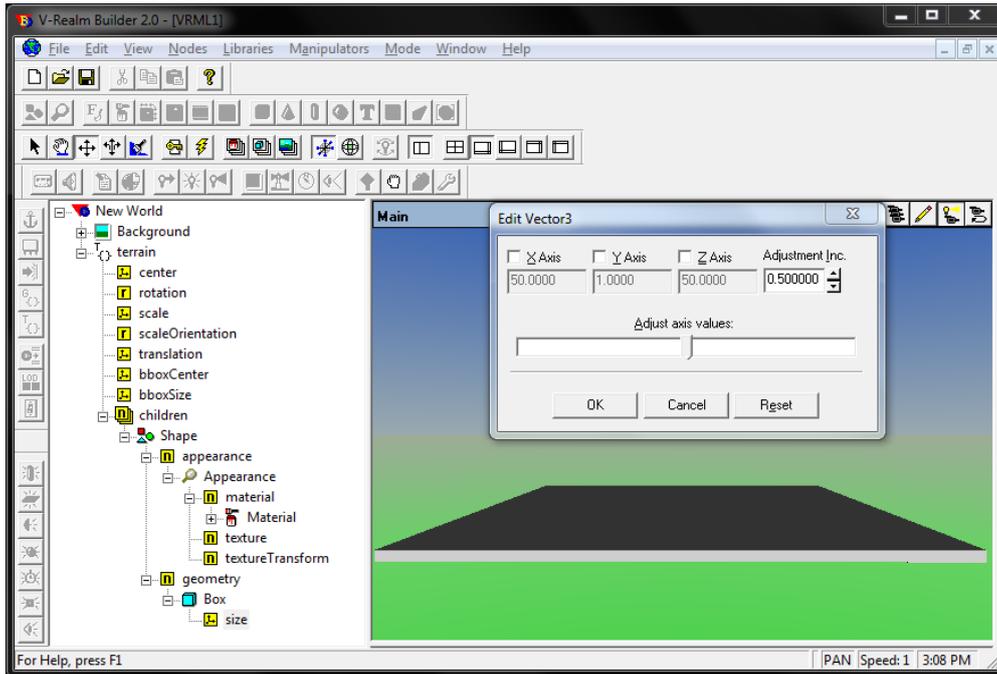


Figure 1.7 configurer la taille

4. Pour ajouter une texture cliquez sur **texture** ou utilisez le bouton **Select Texture**.

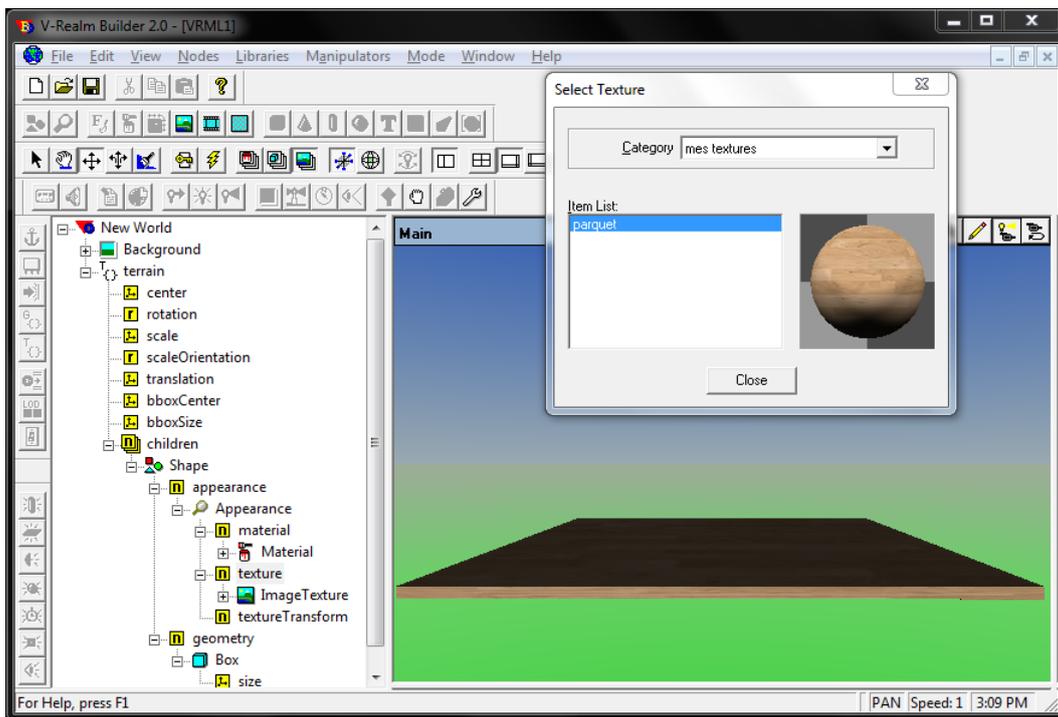


Figure 3.8 choisir une texture

5. Finalement on ajoute de la luminosité a notre maquette en cliquant sur **Add Light**
Puis on configure l'intensité et la position de chaque lumière.

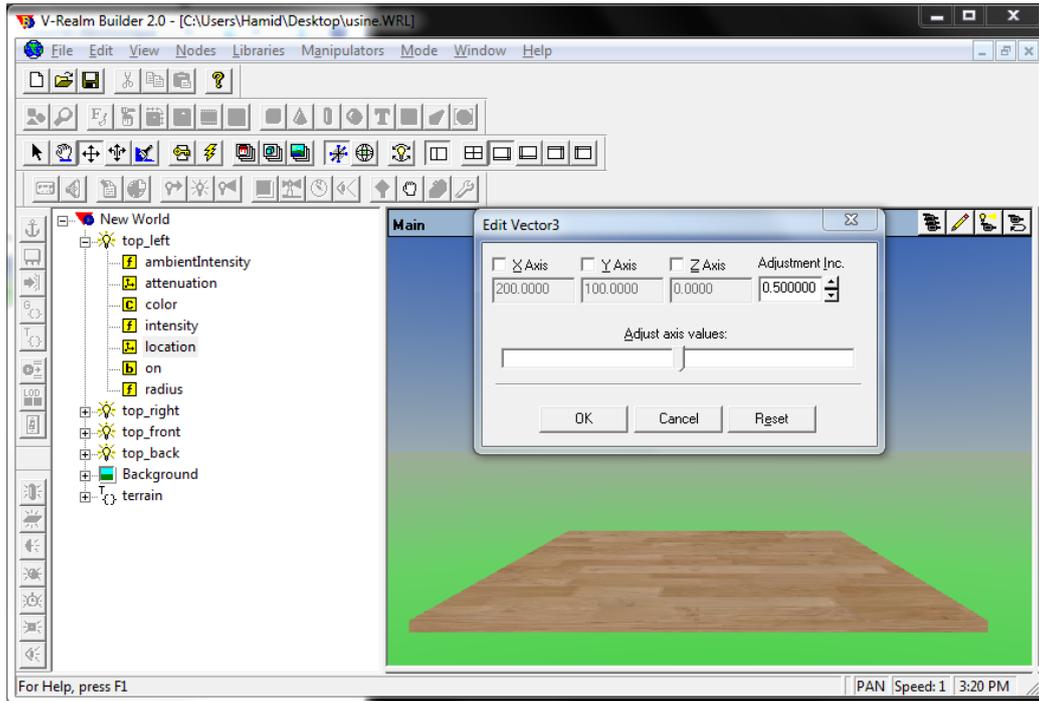


Figure 3.9 ajuster la luminosité

3.8.2 Les convoyeurs

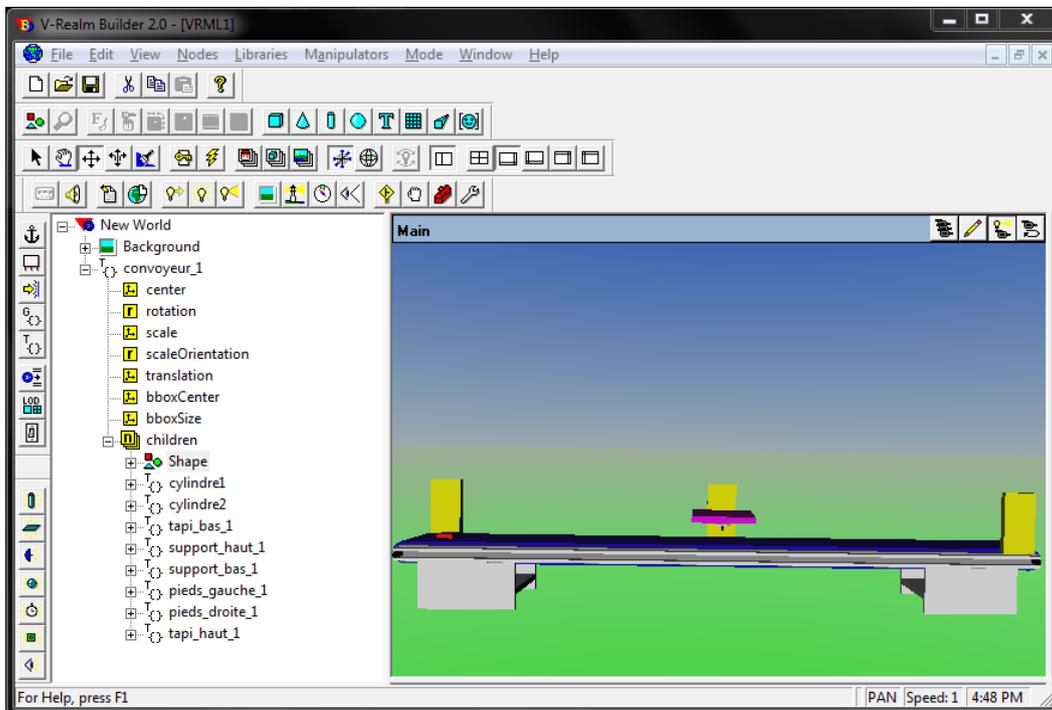


Figure 3.10 Le convoyeur

Conception du convoyeur :

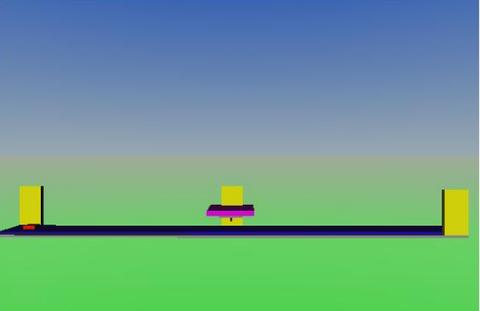
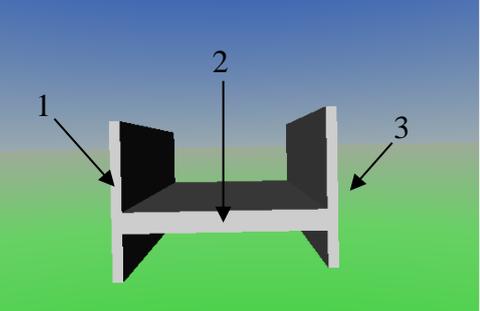
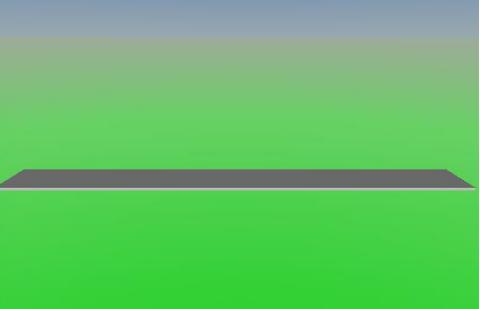
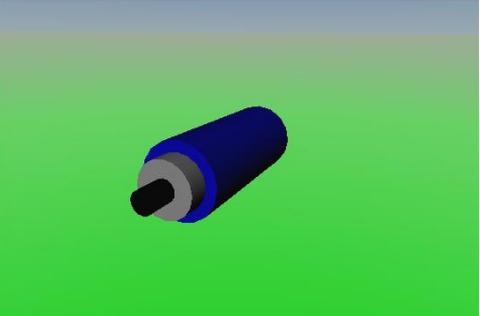
Informations	Objets
<p>_ La base : la base du convoyeur, tous les autres objets sont ses enfants.</p> <p>_ Taille : X=20 Y=0.5 Z=2</p> <p>Coordonnée : X=20 Y=0.5 Z=2</p>	
<p>_ Tapis_haut_1 : possède 4 enfants.</p> <p>_ Taille : X=20 Y=0.1 Z=2</p>	
<p>_ Les pieds :</p> <p>1 : X=3 Y=1.5 Z=0.1</p> <p>2 : X=3 Y=0.2 Z=2</p> <p>3 : X=3 Y=1.5 Z=0.1</p>	
<p>_ Les supports : X=20.2 Y=0.1 Z=2.4</p>	
<p>_ Les cylindres :</p> <p>Noir : Hauteur=2.4 Rayon=0.1</p> <p>Gris : Hauteur=1.9 Rayon=0.25</p> <p>Bleu : Hauteur=1.6 Rayon=0.35</p>	

Figure 3.11 composition du convoyeur

3.8.3 Le robot

Le robot est une succession père - fils d'objets en commençant par la base qui est un cylindre.

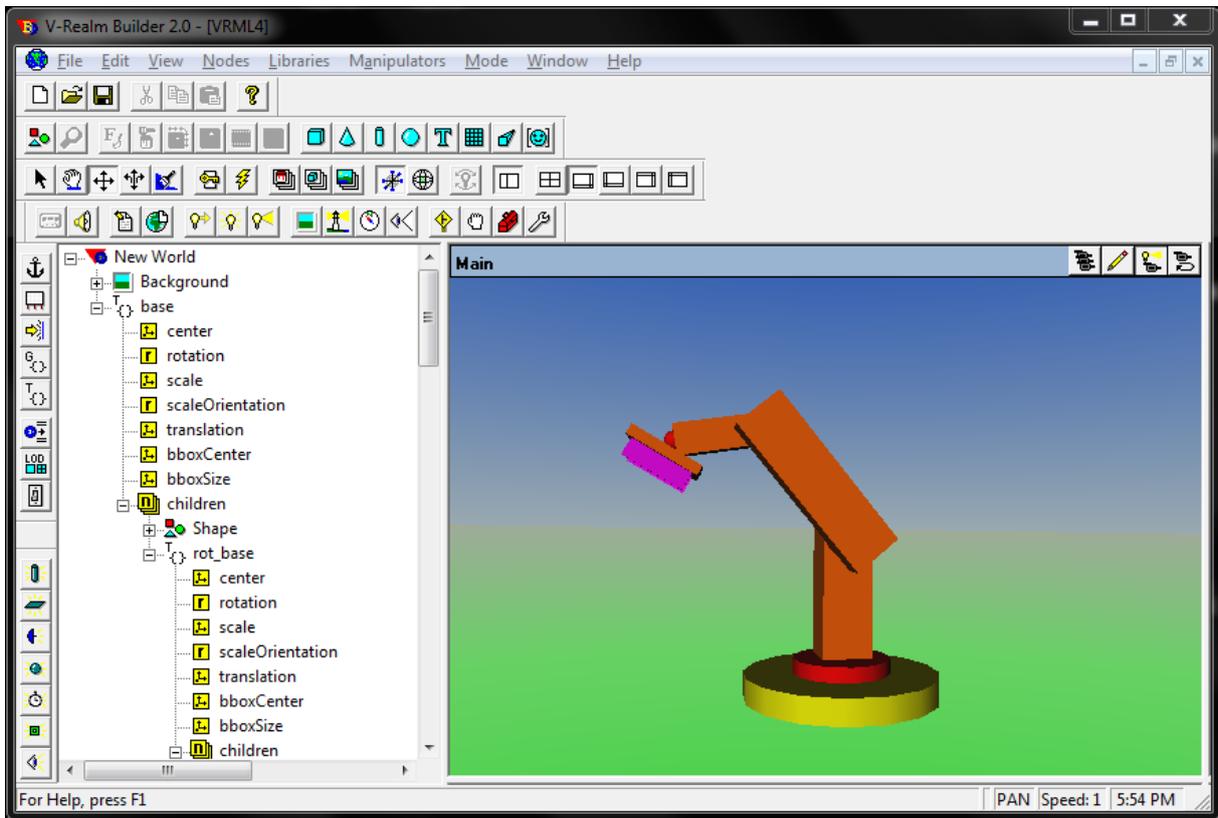


Figure 3.12 Le robot

Nous avons réalisé ce robot en suivant le robot PUMA 560.

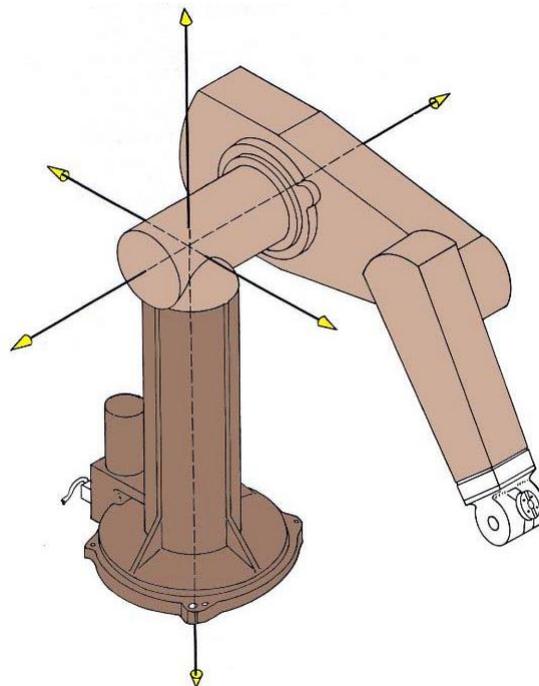


Figure 3.13 robot PUMA 560

3.8.4 La maquette final

La maquette étant réalisé il nous reste à configurer les vues, c'est-à-dire la maquette vue de différents angles, pour ce faire il suffit de cliquer sur le bouton **Access/Edit Viewpoint**, puis configurer sa position.

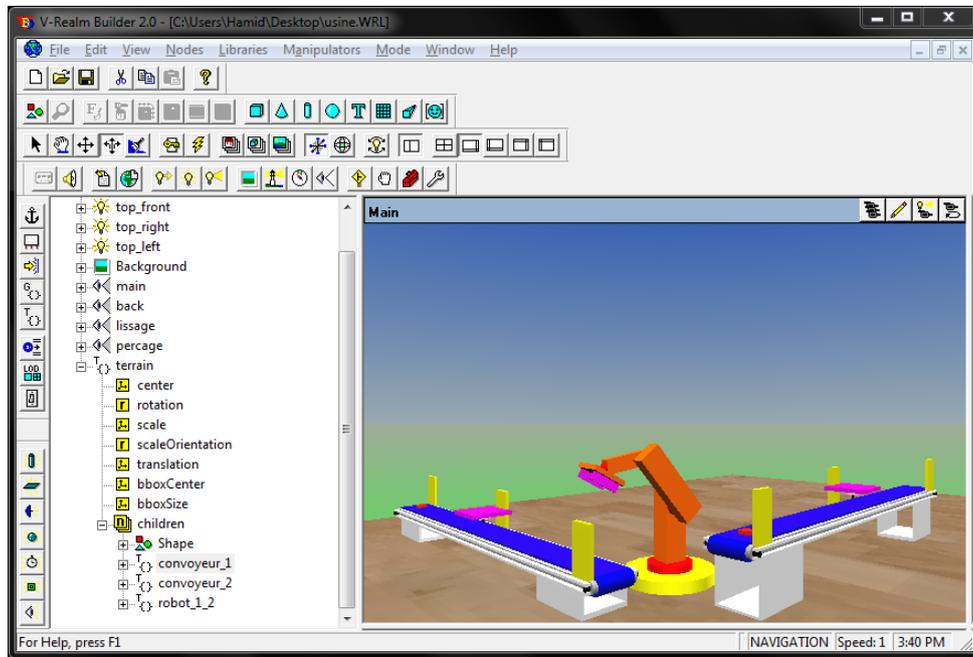


Figure 3.14 la maquette finale

3.9 Simulink 3D toolbox

Simulink 3D Animation relie les modèles Simulink et les algorithmes MATLAB® à des objets graphiques 3D dans des scènes de réalité virtuelle. Vous pouvez animer un monde virtuel en modifiant la position, la rotation, l'échelle et d'autres propriétés d'objet pendant la simulation de bureau ou en temps réel. Vous pouvez également détecter les collisions et autres événements dans le monde virtuel et les réinjecter dans vos algorithmes MATLAB et Simulink [2].

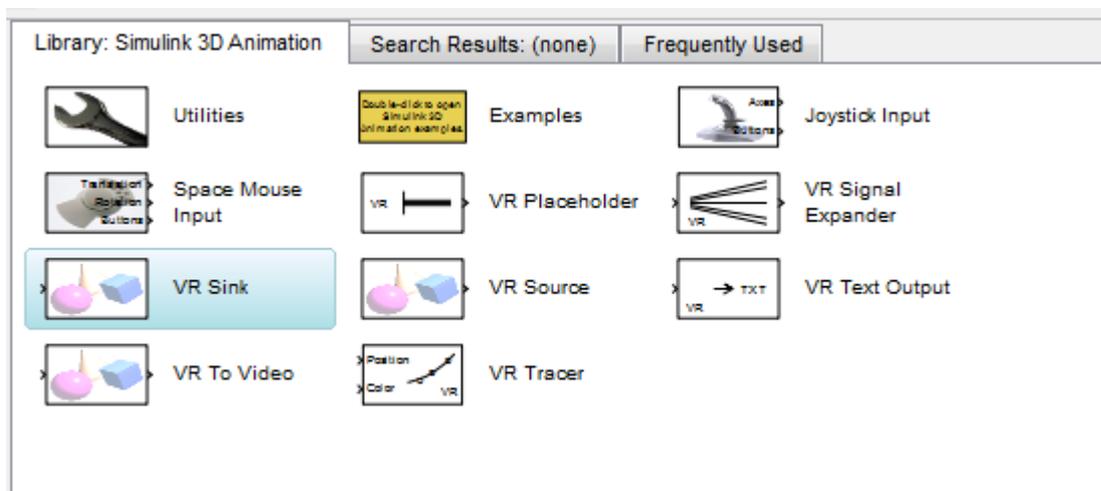


Figure 3.15 contenue de la bibliothèque 3D Animation

3.10 Modèle Simulink

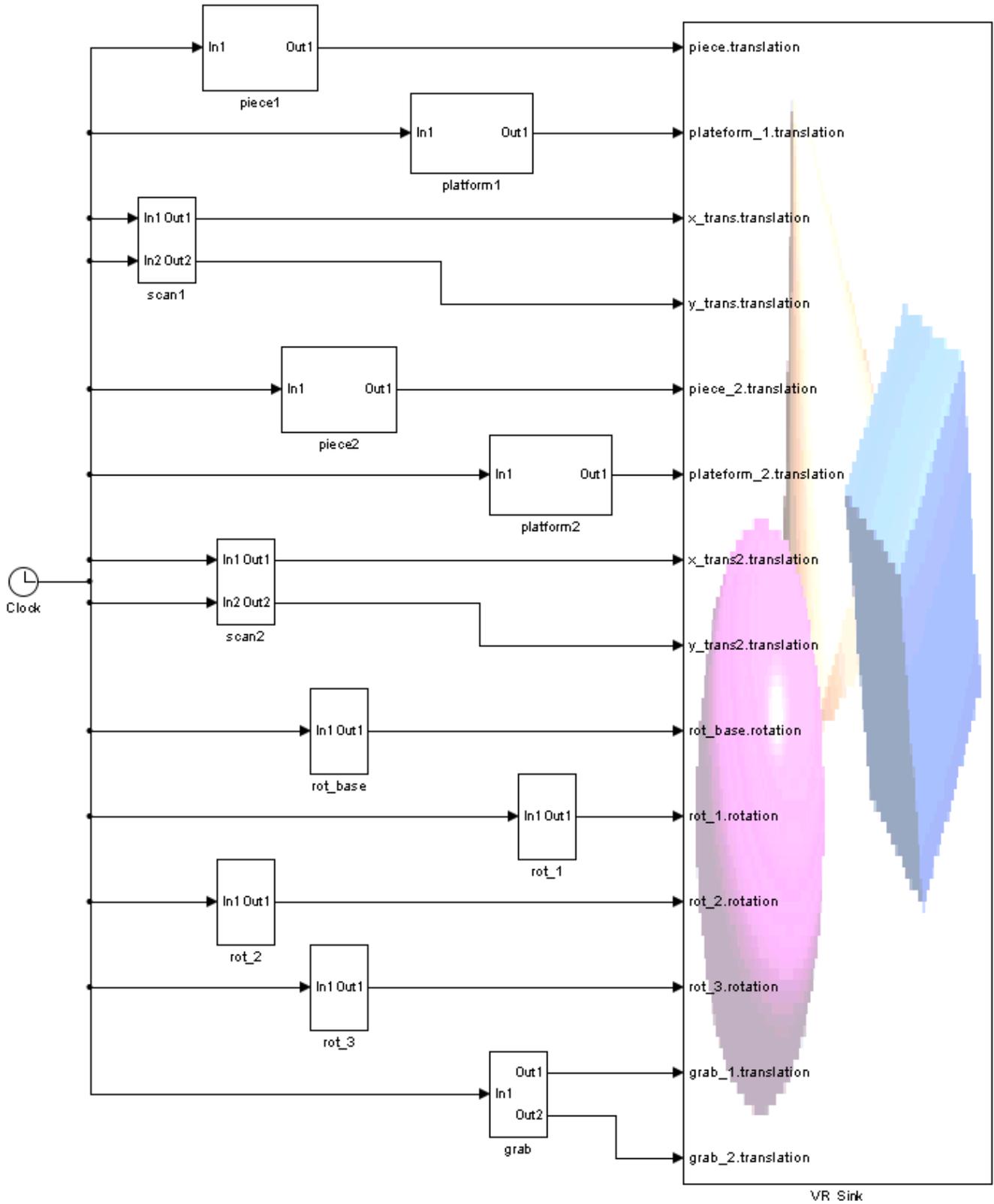


Figure 3.16 Modèle Simulink

Contenu du sous-système « grab ».

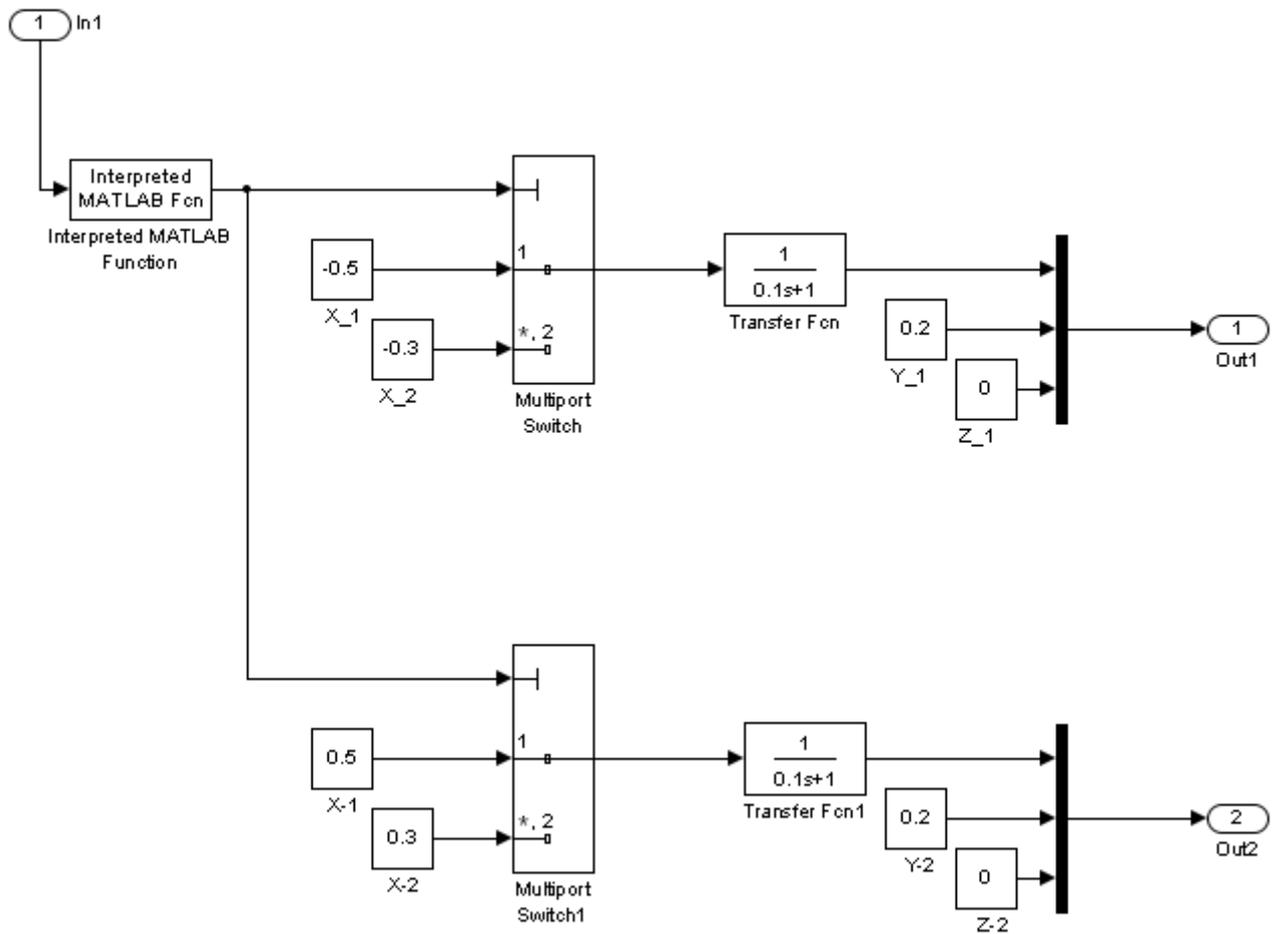


Figure 3.17 sous-système grab

_On a supposé que les moteurs et les différents actionneurs sont déjà régulé d'où nous avons utilisé des fonctions de transfert du premier ordre.

_Le bloc **Interpreted MATLAB Function** contient la fonction **f_grab**.

3.11 Fonctions

FONCTIONS	DESCRIPTION
f_piece1	Déplacer la pièce sur le convoyeur de perçage
f_piece2	Déplacer la pièce sur le convoyeur de lissage
f_platform1	Controler la position du robot de perçage (haute, basse)
f_platform2	Controler la position du robot de lissage (haute, basse)
f_scan1	Controler la position de l'outil de perçage
f_scan2	Controler la position de l'outil de lissage
f_rotbase	la rotation du robot selon l'axe Y
f_rot1	la rotation du bras du robot selon l'axe X
f_rot2	la rotation du coude du robot selon l'axe X
f_rot3	la rotation du poignet bras du robot selon l'axe X
f_grab	ouverture ou fermeture de la pince de l'outil du robot
delay()	Temporisation

Figure 3.18 les fonctions

3.12 Les instructions

SCRIPT	DESCRIPTIONS
initialiser	initialiser le système
robot1	prendre la pièce sur le convoyeur de perçage et la poser sur le convoyeur de lissage
cycle	effectuer un cycle complet
charger1	charger la pièce sur le convoyeur de perçage
charger2	charger la pièce sur le convoyeur de lissage
descendre1	descendre la Plateforme de perçage
descendre2	descendre la plateforme de lissage
monter1	monter la plateforme de perçage
monter2	monter la plateforme de lissage
pos_lissage	déplacer la pièce sous la plateforme de lissage
pos_percage	déplacer la pièce sous la plateforme de perçage
evacuer1	déplacer la pièce pour l'évacuer au convoyeur de lissage
evacuer2	déplacer la pièce pour l'évacuer
perçage	démarrer le perçage
lissage	démarrer le lissage

Figure 3.19 les instructions

3.13 Matlab GUIDE

Une interface utilisateur MATLAB est une fenêtre de figure à laquelle on peut ajouter des composants gérés par l'utilisateur. On peut sélectionner, dimensionner et positionner ces composants comme on le souhaite. À l'aide des callbacks, On peut générer des événements quand l'utilisateur clique sur ou manipule les objets créés.

On peut créer des interfaces utilisateur MATLAB de deux manières, en utilisant le GUIDE ou par programmation, nous avons choisis d'utiliser le GUIDE.

Cette approche commence par une figure qu'on remplit avec des composants à partir d'un éditeur de mise en page graphique. GUIDE crée un fichier de code associé contenant des rappels (callbacks) pour l'interface utilisateur et ses composants. GUIDE enregistre à la fois la figure (sous forme de fichier FIG) et le code fichier. On peut lancer l'application à partir de l'un ou de l'autre [14].

3.13.1 Ouverture du GUIDE

1. Démarrez GUIDE en tapant guide dans l'invite Matlab.

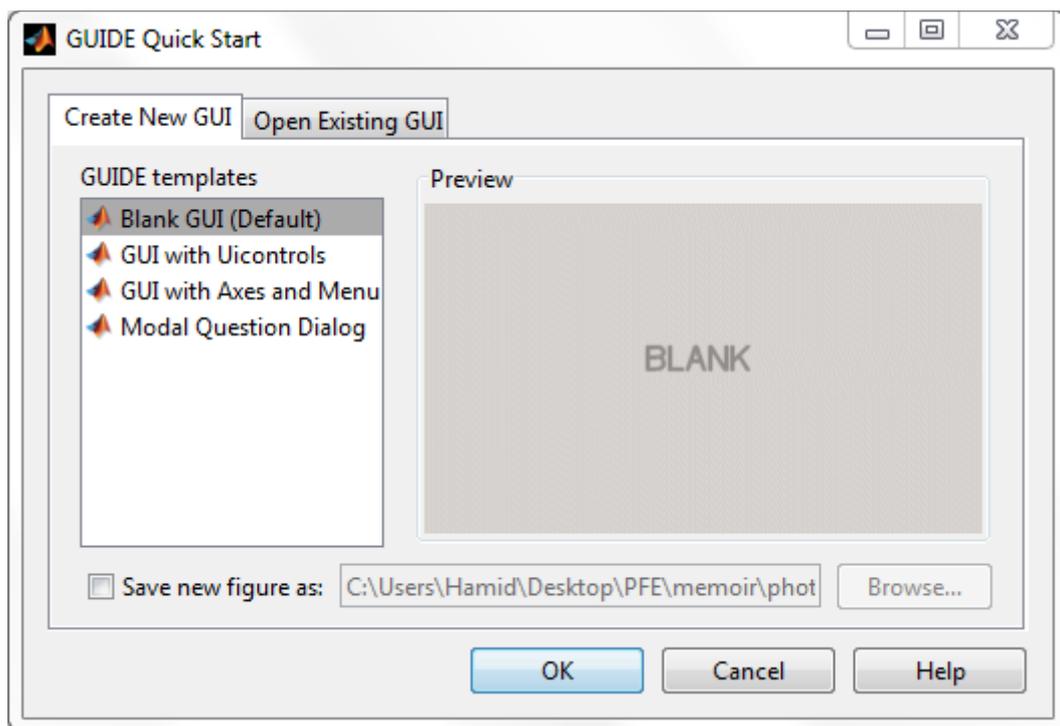


Figure 3.20 GUIDE Matlab

2. Dans la boîte de dialogue **GUIDE Quick Start**, sélectionnez le modèle **Blank GUI(Default)** et puis cliquez sur **OK**.

3. Affichez les noms des composants de l'interface utilisateur dans la palette d'outils:
 - a. Sélectionnez **File > Preferences > GUIDE**.
 - b. Sélectionnez **Show names in component palette**.
 - c. Cliquez **OK**.

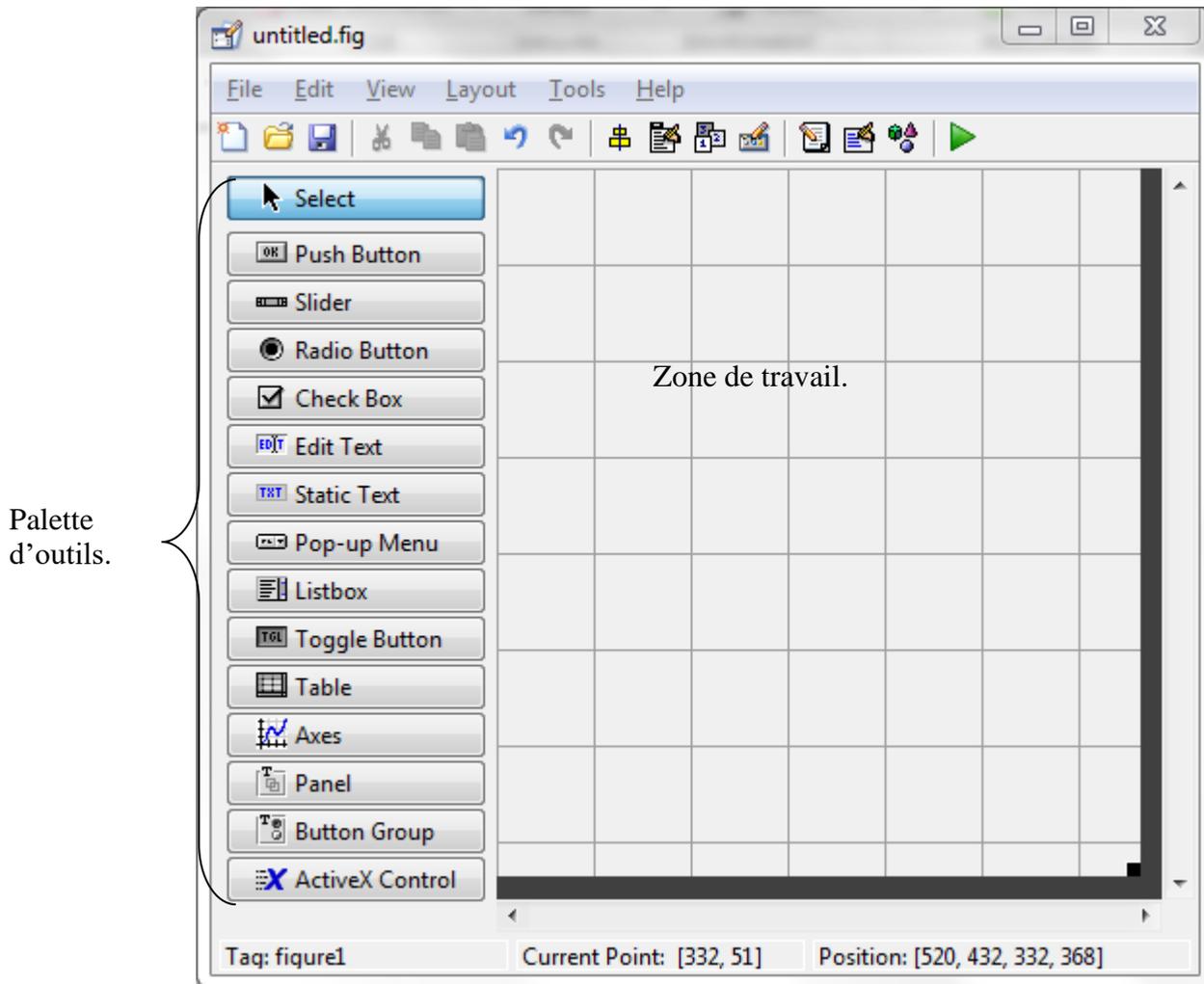


Figure 3.22 guide par défaut

3.14 Pupitre

Ci-dessous le pupitre que nous avons créé, divisé en deux partie :

main controls : le pupitre principal qui possède deux bouton, **initialiser** pour initialiser le système au condition de départ.

Manual : pour contrôler chaque station manuellement, les deux convoyeurs ainsi que le robot.

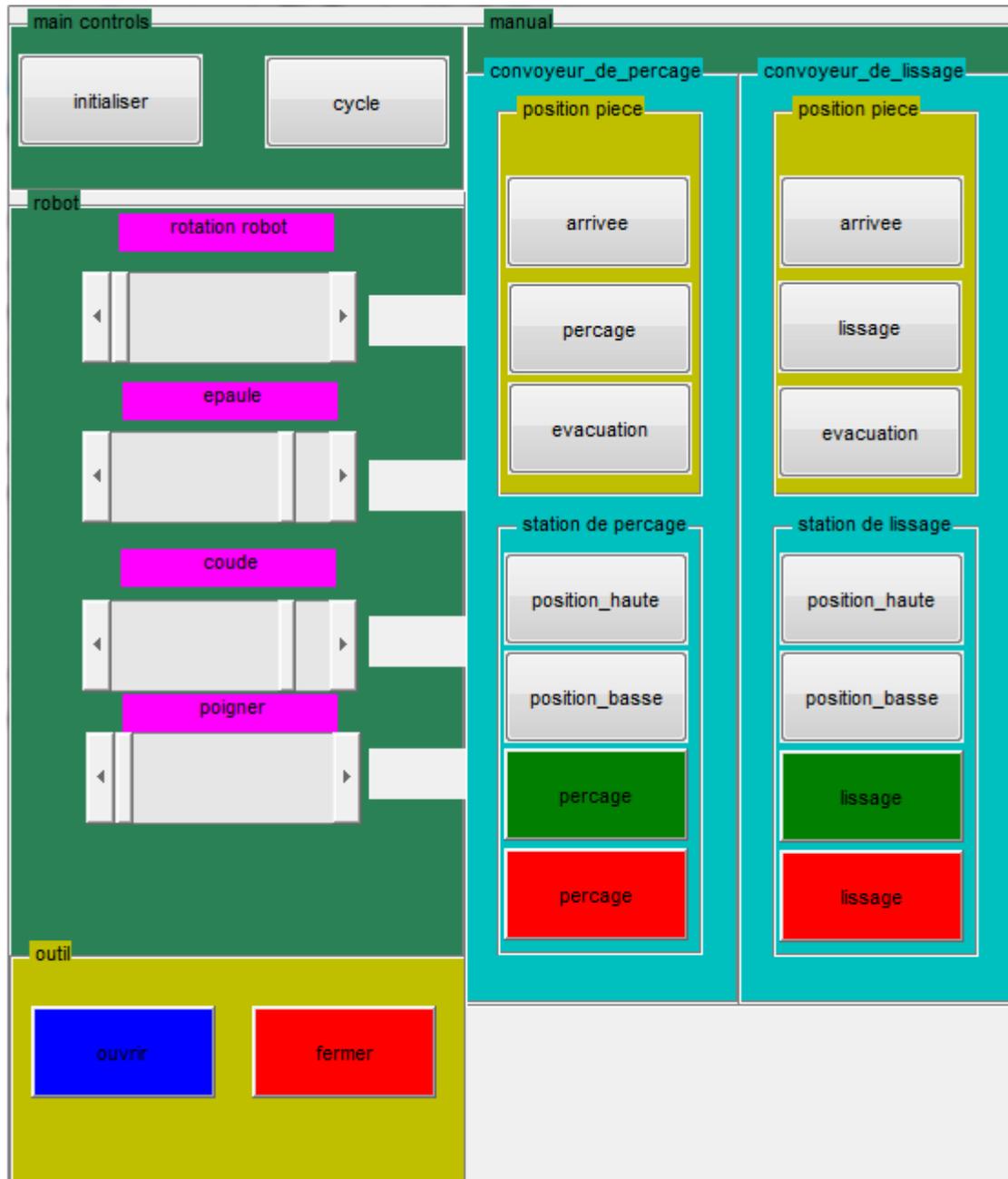
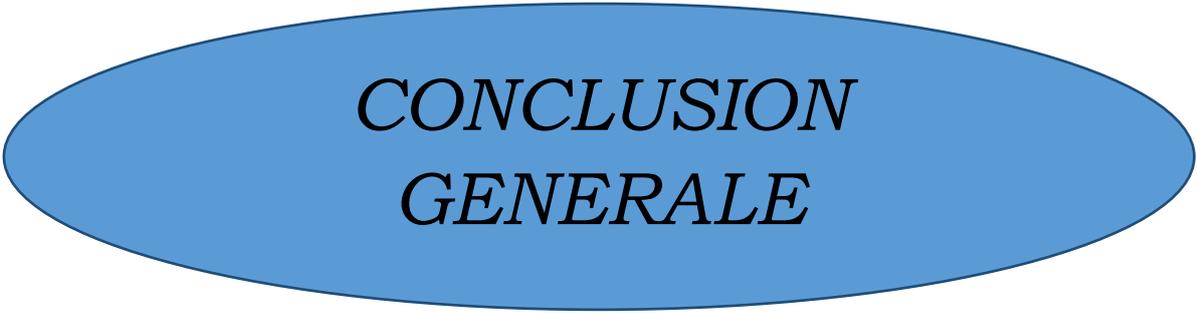


Figure 3.22 le pupitre

3.15 Conclusion

Dans ce chapitre on a simulé et visualiser trois maquettes virtuelles de trois stations de production différentes sous Matlab. Dont chaque station est suivie des différentes étapes de construction, de modélisation, de simulation et d'animation. Après avoir fait la simulation de chaque station seule, on a assemblé toutes les stations (trois stations) pour former un système complet (chaîne de production automatisée) et on a gardé le model dynamique de chaque station.

La simulation de toute la chaîne a validé notre approche successive. Cette approche consiste à utiliser la bibliothèque des composants pour former des stations, de module en même système complet sous forme virtuelle et d'élaborer la commande nécessaire et valider le résultat par simulation.



*CONCLUSION
GENERALE*

CONCLUSION GENERALE

Au cours de ce travail nous avons rappelé les différentes notions et définitions d'un système automatisé qui nous aide à la réalisation de la maquette virtuelle 3D d'une chaîne de production automatisée et à sa conception.

Le projet réaliser des composants peut être utilisé pour la réalisation de chaînes de production automatisées réelle et pratique. Pour cela, nous avons proposé trois stations de production. Chaque station est constituée d'un ensemble de composants.

Après avoir réalisé les trois stations sous forme de maquettes virtuelles, nous avons développé les modèles dynamiques pour trois stations afin d'avoir leurs fonctionnements dans un environnement 3D et les combinées ensuite sous forme d'une chaîne de production automatisée, ensuite nous avons créé un ensemble d'instructions qui forme un mini langage de programmation pour ce système.

Ce travail nous a permis de mettre en valeur nos connaissances théorique et pratique que nous avons cumulées pendant toute la durée de nos études et de nos stages pratiques, il nous a permis aussi d'acquérir une nouvelle expérience dans la programmation avec Matlab/Simulink 3D Animation sous le logiciel Virtuel Realm Builder.

Références bibliographiques

[1] RAZIBAOUENE Ihab, BENYOUCEF Samir., Automatisation d'un processus d'emballage à l'aide d'un robot série à quatre degrés de liberté. Mémoire de Master 2 : Génie Electrique option Automatique. Université M'hamed Bougara.BOUMERDES. Juin 2017.P.65

[2] SAMUEL Tardieu, S.D. Robotique et Systèmes Embarqués. Ecole Nationale Supérieure des Télécommunications. PARIS.

[3] JACQUES Gangloff., S.D. Cours de Robotique. ENSPS 3A MASTER ISTI.

[4] Cours Robotique ISTIA, Université Angers Jean-Louis Boimand.ANGERS.P.93.

[5] ALEXANDER Girard., Cours de Modélisation, Analyse et Commande des Systèmes Robotisés. Université de Sherbrooke.QUEBEC.P.163.

[6] MAKHLOUFI Fateh., Modélisation et Commande des Robots Manipulateurs Par les Outils de L'intelligence Artificielle. Thèse de Doctorat : Génie Mécanique. Université Badji Mokhtar. ANNABA.2015.P.153.

[7]<https://www.universalis.fr/encyclopedie/robots/2-le-robot-industriel/>

[8] (traduit de l'anglais) LIVRE: INDUSTRIAL AUTOMATION AND ROBOTICS. Ch16 P.511/512.

[9] AIDLI Abdelghani, AITLHADI Bilal., Utilisation de V-Realm Builder pour la réalisation des maquettes virtuelles des chaînes de production automatisées, Mémoire Master recherche en Electrotechnique option Automatismes Industriels, Université Abedrahman Mira de Bejaïa, 2012/2013, P.77.

[10] The Editors of Encyclopaedia Britannica, Ransom Eli Olds, <https://www.britannica.com/biography/Ransom-Eli-Olds>, consulté le 18 octobre 2020.

Et History.com Editors, Ford's assembly line starts rolling, <https://www.history.com/this-day-in-history/fords-assembly-line-starts-rolling>, consulté le 18 octobre 2020.

[11] TAHIRI, « Les systèmes de production automatisés », Support de Cours sur les Automatismes et l'Automatisation, ENIM (Ecole Nationale de l'Industrie Minérale) - Rabat(Maroc), 2010. [12] Khaled, N (2012). Virtual Reality and Animation for MATLAB® and Simulink® Users : Visualization of Dynamic Models and Control Simulations. Springer.

[13] The Math Works, Inc. Simulink® 3D Animation™ User's Guide (R2020a)

https://www.mathworks.com/help/pdf_doc/sl3d/sl3d_toolbox.pdf

[14] The Math Works, Inc. Creating Graphical User Interfaces (R2015b)

<http://www.apmath.spbu.ru/ru/staff/smirnovmn/files/buildgui.pdf>