

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE
UNIVERSITE MOHAMED BOUDIAF - M'SILA

FACULTE : Science & Technologie

DEPARTEMENT: Génie Mécanique

N° :.....



DOMAINE : Science & Technologie

FILIERE : Génie Mécanique

OPTION : Construction Mécanique

Mémoire présenté pour l'obtention
Du diplôme de Master Académique

Par: KHIRANI Hemza

Intitulé

Développement D'un Simulateur pour Simuler
Les Mouvements D'un robot à 6 axes

Soutenu devant le jury composé de:

<i>DR.MENASRI .N</i>	Université Mohamed BOUDIAF - M'Sila	<i>Président</i>
<i>DR.SLAMANI.M</i>	Université Mohamed BOUDIAF - M'Sila	<i>Rapporteur</i>
<i>DR.AMROUNE.S</i>	Université Mohamed BOUDIAF - M'Sila	<i>Examineur</i>
<i>DR.SAIB.CH</i>	Université Mohamed BOUDIAF - M'Sila	Examineur

Année universitaire : 2017 /2018

Remerciements

Je tiens à exprimer toute ma reconnaissance à mon Directeur de mémoire DR SLAMANI.M. Je la remercie de m'avoir encadré, orienté, aidé et conseillé. J'adresse mes sincères remerciements à tous les professeurs, intervenants et toutes les personnes qui par leurs paroles, leurs écrits, leurs conseils et leurs critiques ont guidé mes réflexions et ont accepté à me rencontrer et répondre à mes questions durant mes recherches.

Je remercie mes très chers parents, qui ont toujours été là pour moi, « Vous avez tout sacrifié pour vos enfants n'épargnant ni santé ni efforts. Vous m'avez donné un magnifique modèle de labeur et de persévérance. Je suis redevable d'une éducation dont je suis fier ».

Je remercie mon frère, et mes sœurs pour leur encouragement. Je remercie très spécialement mon encadreur dans entreprise KARAKE.O.

Enfin, je remercie tous mes Ami(e)s que j'aime tant, Pour leur sincère amitié et confiance, et à qui je dois ma reconnaissance et mon attachement. À tous ces intervenants, je présente mes remerciements, mon respect et ma gratitude.

Résumé

Dans ce travail, un environnement de simulation pour les robots industriels à six axes est proposé. Ce simulateur permet de commander et de contrôler les mouvements des robots industriels. Il peut être utilisée pour comprendre le comportement des robots industriels dans différents conditions (charge, vitesse, environnement, ...), l'évaluation de la disposition, les études de faisabilité, les présentations avec animation et la programmation hors ligne.

Cet environnement de simulation comprend quatre modules : a) analyse de mouvement du robot, b) modélisation géométrique directe c) modélisation géométrique inverse et d) analyse de la simulation.

Pour cela, un modèle CAO d'un robot industriel à six axes type ABB IRB 2600 est développé en détail (axe par axe) puis assemblé sur le logiciel Solid-Works. Une interface graphique de simulation est développée sur le logiciel Matlab en utilisant l'outil GUI. Par la suite, les deux composantes (interface graphique et modèle CAO) ont été intégrer pour développer notre simulateur. Les diagrammes blocs de contrôle de chaque axe sont obtenus à travers l'outil Simulink, SimMechanics et SimscapeMultibody.

Cette approche pourrait être facilement généralisée à tout type de robot en série (Fanuc, Kuka, Scara, Puma, etc.).

Ce logiciel de simulation 3D est destiné aussi aux étudiants de Master qui suivent un cours de robotique universitaire. Ce logiciel pédagogique permet d'illustrer de façon visuelle plusieurs concepts en robotique.

Mots clé: Simulation, Modélisation géométrique directe, Modélisation géométrique inverse, SolidWorks, interface graphique, SimMechanics,

Liste des figures

Liste des Figures

Liste des Figures

Figure 1. 1 Robot SCARA (Adept Technology) [6]	2
Figure 1. 2 Robot cartésien TTT [6]	3
Figure 1. 3 Robot cylindrique (RTT) [6].....	4
Figure 1. 4 Robot sphérique (RRT) [6]	4
Figure 1. 5 Configuration SCARA (RRT) [6]	5
Figure 1. 6 Robot angulaire poly articulé (RRR) [7].	6
Figure 1. 7 Robot parallèle à deux degrés de liberté (Mitsubishi) [7].	7
Figure 1. 8 Plateforme de Gough [7].....	8
Figure 1. 9 Robot Hexa [15]	9
Figure 1. 10 Modélisation des robots à 2 degrés de liberté utilisés pour la comparaison [16],	10
Figure 1. 11 Comparaison des volumes de travail [16],	10
Figure 1. 12 Comparaison de l'évolution du couple du moteur M1 [16],	11
Figure 1. 13 Modes de fonctionnement, singularités sérielles (noir), singularités parallèles (pointillés) [17].....	14
Figure 1. 14 Espace de travail du robot IRB 6660 [ABB] [15]	15
Figure 1. 15 Amplitude et fréquence des erreurs d'usinage et ses causes [19]	18
Figure 2. 1 placement des repères et notation pour le robot [21].	21
Figure 2. 2 Les quatre transformations consécutives de base pour obtenir F_i à partir de F_{i-1} [21]	23
Figure 2. 3 le robot IRB 2600	25
Figure 2. 4 les dimensions du robot ABB IRB 2600	26
Figure 2. 5 algorithme de Matlab pour calculer H_{06}	28
Figure 3. 1 La programmation en ligne	31
Figure 3. 2 Logiciel de programmation hors-ligne	33
Figure 3. 3 Méthodologie de programmation hors-ligne.....	34
Figure 3. 4 Structure arborescente composée d'objets d'interface .[23]	36
Figure 3. 5 Exemple d'interface	36
Figure 3. 6 mis en place les composants essentiels.....	40
Figure 3. 7 Programmation de l'interface graphique.....	41
Figure 3. 8 Modèle de robot IRB 2600 sur solide Works.....	41
Figure 3. 9 modèle robot IRB 2600 sur simulink.....	42
Figure 3. 10 Modèle Robot IRB 2600 avec les Slider	43
Figure 3. 11 algorithme de Matlab pour les Slider	44
Figure 4. 1 Format initial de notre simulateur.....	45
Figure 4. 2 Résultat de la cinématique direct.....	46
Figure 4. 3 Résultat de la cinématique inverse	47
Figure 4. 4 Notre modèle dans simscape (simulink).....	48
Figure 4. 5 Résultat de cinématique direct avec les slider est la déclaration de la position actuelle ...	48
Figure 4. 6 Notre Modèle Robot IRB 2600 dans simulinkk avec les slider	49



Liste des tableaux

Liste des Tableau

Liste des Tableau

Tableau 1. 1 Tolérances commerciales des procédés d'usinage.....	17
Tableau 1. 2 Types de causes des erreurs.	19
Tableau 2. 1 Paramètres géométriques du robot	23
Tableau 2. 2 Les axes du manipulateur	26
Tableau 2. 3 paramètres DH DE robot ABB IRB 2600	27
Tableau 2. 4 solutions de la cinématique inverse	30

Sommaire

1/Introduction générale	1
2. Introduction.....	2
2-1/Robots sériels.....	3
2-1-1/Géométries.	4
2-1-2/Précision de positionnement.....	7
2-2/Robots parallèles.....	7
2-3/ Comparaison des robots parallèles et sériels	11
2-3-1/ Architectures	11
2-3-2/ Volume de travail	11
2-3-3/ Dynamique	12
2-3-4/ Les défauts d'un robot sériel	13
2-3-5/ Les avantages d'un robot parallèle	13
2-3-6/ Les inconvénients des robots parallèles	14
2-4/ Espace de travail :.....	14
2-4-1/ Robot parallèle:.....	14
2-4-2/ Robot serials:.....	15
2-5/ Répétabilité de pose	16
2-5-1/ La théorie des ellipsoïdes stochastiques:.....	16
2-6/Précision absolue de pose.....	17
3/Introduction:	21
4/Modélisation Géométrique.....	22
5/Cinématiques directe:	23
5-1/La méthode Denavit-Hartenberg:	23
5-2/ Équation de la cinématique directe:.....	25
5-3/Etude cinématique direct du robot ABB IRB 2600	25
6/ Cinématiques inverse du robot ABB IRB 2600	29
7/Programmation:	31
7-1/Programmation du robot:.....	31
7-2/Programmation en ligne:	31
7-3/Programmation hors-ligne.....	33
7-4/Développement de l'interface graphique:.....	36
7-4-1/Création d'une interface graphique	37

Sommaire

7-4-2/Création du fichier : m correspondant	41
7-5/Création notre modèle de robot IRB 2600 sur solide Works:.....	42
8/Réalisation et validation du simulateur	46
8-1/Exécution de la cinématique directe	46
8-2/ Exécution de la cinématique inverse.....	47
8-3/.Commande axe par axe	48
8-4/.Visualisation graphique	50
Conclusion	51
Référence	52



Introduction Générale

1/Introduction générale

La robotique est une science de l'ingénierie spéciale qui traite de la conception, de la modélisation, du contrôle et de l'utilisation des robots. De nos jours, les robots accompagnent les gens dans la vie quotidienne et ont repris certaines de leurs procédures quotidiennes. La gamme d'utilisation des robots est très large, allant des jouets aux robots de bureau et industriels, en passant par les robots très sophistiqués nécessaires à l'exploration spatiale.

Avec les progrès de la robotique, la technologie des robots industriels s'améliore et leur coût diminue. Le secteur automobile est certainement celui qui a le plus participé au développement des robots. Les besoins sont toujours très importants dans ce domaine et cela devrait aller en s'amplifiant avec les nouveaux constructeurs issus des pays émergents. Cependant, depuis que les robots ont investi le secteur industriel, les solutions pour les optimiser n'ont cessé de progresser.

L'application de robots a augmentée pour les tâches qui exigent des vitesses élevées, des relations efficaces entre la charge et la puissance, une plus grande précision et une répartition efficace du poids pour un objet donné [1] [2]. C'est là que le développement des phases de contrôle et de simulation est essentiel pour bien maîtriser le comportement des robots industriels.

La simulation des systèmes robotisés devient très populaire, en particulier avec la baisse du coût des ordinateurs. La robotique s'appuie sur des outils informatiques aussi bien dans les phases de conception, de programmation, de modélisation (géométrique, cinématique, dynamique, contraintes, etc.) que de simulation. Ces logiciels qui sont souvent indispensables, aident à la maîtrise globale d'un système robotique, sont à ce jour presque exclusivement réservés à des spécialistes. En particulier, les outils permettant d'aborder les aspects de simulation ou de programmation hors-ligne de robots restent d'utilisations complexes.

L'objectif de cette étude est de donner une contribution dans le domaine de la robotique par le développement d'un environnement de simulation permettant de simuler le mouvement d'un robot industriel à six axes. Cela peut être achevé par la création d'un environnement de simulation virtuelle dans un système CAO (SolidWorks) à travers une interface graphique réalisée sur un logiciel de programmation comme Matlab.

Chapitre I

Généralités sur La Robotique

2. Introduction

La robotique est un domaine de la technologie moderne qui traverse les frontières de l'ingénierie traditionnelle, Comprendre la complexité des robots et de leurs applications nécessite des connaissances dans différents domaines de l'ingénieur ,tel que la mécanique, l'électrique, systèmes et génie industriel [3].

Le terme «robot» a été introduit pour la première fois dans le vocabulaire par le dramaturge tchèque Karel Capek dans sa pièce de théâtre Rossum's Universal Robots en 1920, le mot «robot» étant le mot tchèque pour le travail. Depuis lors, le terme a été appliqué à une grande variété de dispositifs mécaniques, tels que les téléopérateurs, les véhicules sous-marins, les véhicules terrestres autonomes, etc. Pratiquement tout ce qui fonctionne avec un certain degré d'autonomie, généralement sous contrôle informatique, a été appelé un robot [4].

Selon la norme internationale ISO 8373 [5], un robot industriel est un (manipulateur multi-application reprogrammable commandé automatiquement, programmable sur trois axes ou plus, qui peut être fixe sur place ou mobile, destiné à être utilisé dans des applications d'automatisation industrielle). Selon cette même norme, un manipulateur est une "machine dont le mécanisme est généralement composé d'une série de segments, articulés ou coulissants l'un par rapport à l'autre, ayant pour but de saisir et/ou de déplacer des objets (pièces ou outils) généralement suivant plusieurs degrés de liberté". La partie extrême du manipulateur qui porte l'outil (préhenseur, pince de soudage, etc.) s'appelle l'effecteur du robot On peut classer les robots fixes selon 2 groupes principaux:

- les robots **sériels**, caractérisés par le fait qu'une seule chaîne cinématique relie la base fixe à l'organe terminal mobile
- les robots **parallèles** caractérisés par le fait que plus d'une chaîne cinématique assure la liaison entre la base fixe et l'organe terminal mobile; on parle également de cinématique bouclée.

2-1/Robots sériels

Un robot en série est constitué de plusieurs liens reliés en série par différents types d'articulations, typiquement tournants et prismatiques. Une extrémité du robot est fixée au sol et l'autre extrémité est libre de se déplacer dans l'espace. Le lien fixe est appelé base, et l'extrémité libre où un préhenseur ou une main mécanique est attachée, est appelé effecteur terminal.

Pour qu'un robot exécute une tâche spécifique, l'emplacement de l'effecteur terminal par rapport à la base doit être établi en premier. C'est ce qu'on appelle un problème d'analyse de position [6].

Le robot sériel le plus répandu dans le domaine du pick-and-place est le robot SCARA (Selective Compliance Assembly Robot Arm) [7]. Selon la norme ISO/TR 8373 [8], ce mécanisme est un "robot rotoïde dont les axes de rotation des articulations simples du bras sont verticaux, avec compliance horizontale". Ce robot possède quatre *ddl*, c'est-à-dire les trois translations x, y, z et la rotation autour de l'axe z (Figure 1.1).

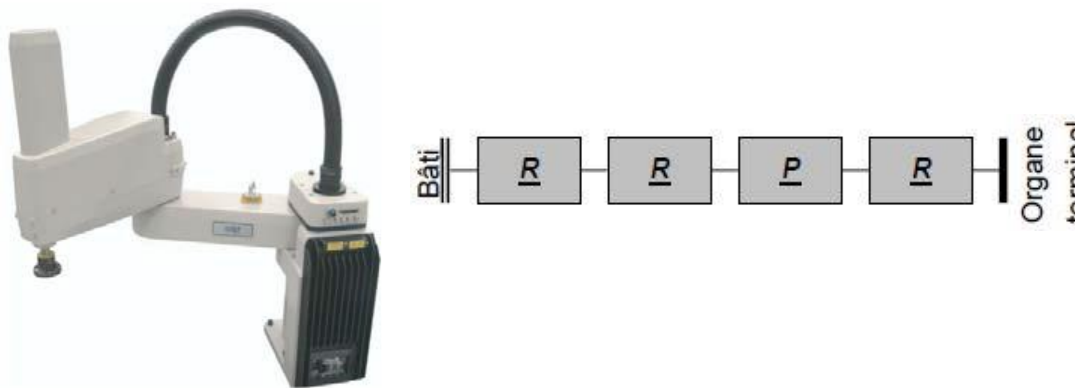


Figure1. 1 Robot SCARA (Adept Technology) [6]

Il existe deux types de problèmes d'analyse de position: la position directe ou la cinématique directe et la position inverse ou les problèmes de cinématique inverse.

Pour la cinématique directe, les variables communes sont données et le problème est de trouver l'emplacement de l'effecteur terminal. Pour la cinématique inverse, la localisation de l'effecteur terminal est donnée et le problème est de trouver les variables de joint nécessaires pour amener l'effecteur terminal à la position désirée. Pour un robot série, la cinématique directe est assez simple, alors que la cinématique inverse devient très difficile.

Pour un robot déficient, effectuer de fin ne peut pas être positionné librement dans l'espace, et pour le robot redondant il peut y avoir plusieurs infinités de solutions cinématiques inverses correspondant à un emplacement d'effecteur donné, en fonction des degrés de redondance. En résolvant le problème de la cinématique inverse, nous cherchons souvent à obtenir une solution de forme fermée, c'est-à-dire à réduire le problème à une équation algébrique reliant la localisation de l'effecteur à une seule variable conjointe. De cette manière, toutes les solutions possibles et les positions du manipulateur peuvent être prises en compte.

2-1-1/Géométries.

Rappelons que pour les robots sériels, les trois premiers degrés de liberté définissent ce que l'on appelle le *porteur*, alors que les trois suivants constituent le *poignet*. En ce qui concerne le porteur, les axes (appelés axes principaux) peuvent être en translation ou en rotation. Si l'on ne prend pas en compte l'ordre, on dénombre 4 configurations principales: TTT, RTT, RRT, RRR. La configuration TTT (3 translations) est bien adaptée à notre représentation cartésienne de l'espace, elle correspond au monde du concepteur. La position des axes les uns par rapport aux autres ne variant pas au cours d'un mouvement (on parle de géométrie différentielle constante dans l'espace), le calcul de trajectoires et le contrôle (pas de variation d'inertie) s'en trouvent simplifiés. Cette variante est particulièrement intéressante pour l'assemblage de précision ou pour les grandes manipulations; sa mécanique peut être assez délicate.

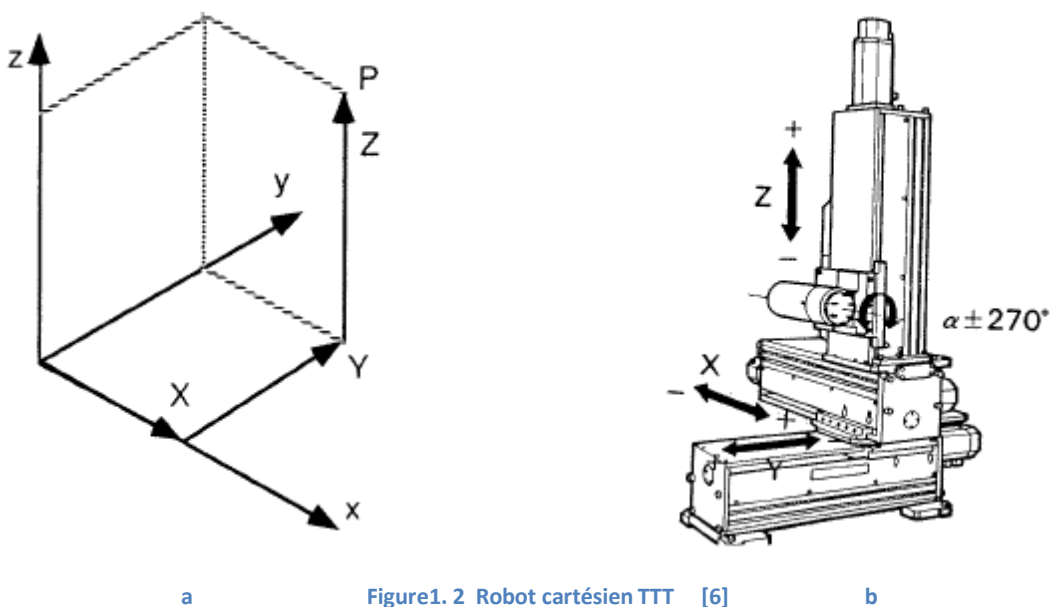
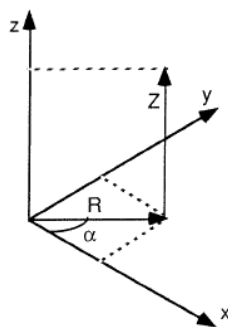
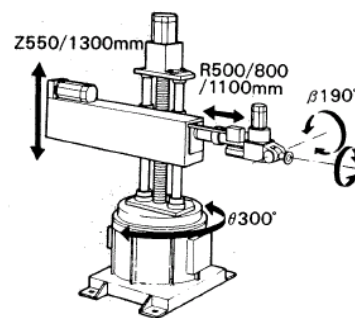


Figure1. 2 Robot cartésien TTT [6]

La configuration RTT (à coordonnées cylindriques) est la combinaison d'une rotation et de deux translations (Figure 1.3 a) (Figure 1.3 b). Elle conduit à une variante, le SCARA, qui a eu du succès en assemblage. Les complications de construction (guidages prismatiques) et la résolution variable selon le rayon d'extension réduisent de plus en plus son utilisation; un robot SCARA le remplace avantageusement pour les applications de moyenne précision et un robot cartésien pour les hautes précisions.



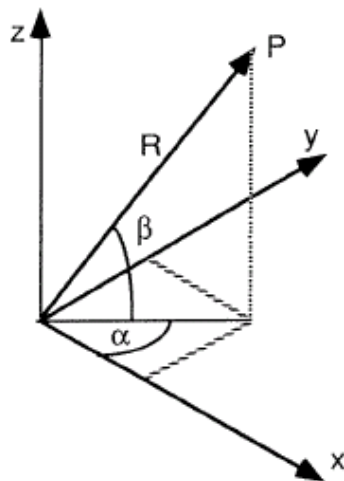
a



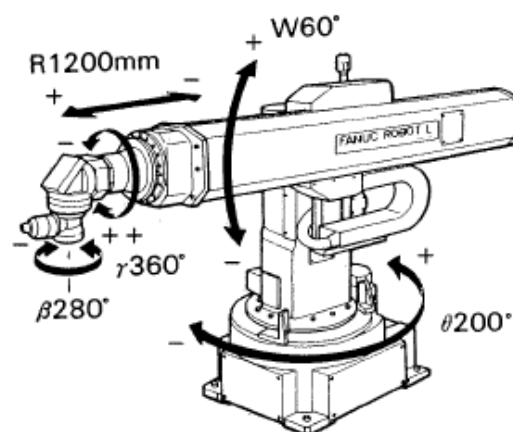
b

Figure1. 3 Robot cylindrique (RTT) [6]

Dans la configuration RRT (Figure 1.4 a), on trouve une grande variété de combinaisons. La plus ancienne d'entre elles est le robot à coordonnées sphériques (Figure1.4 b).



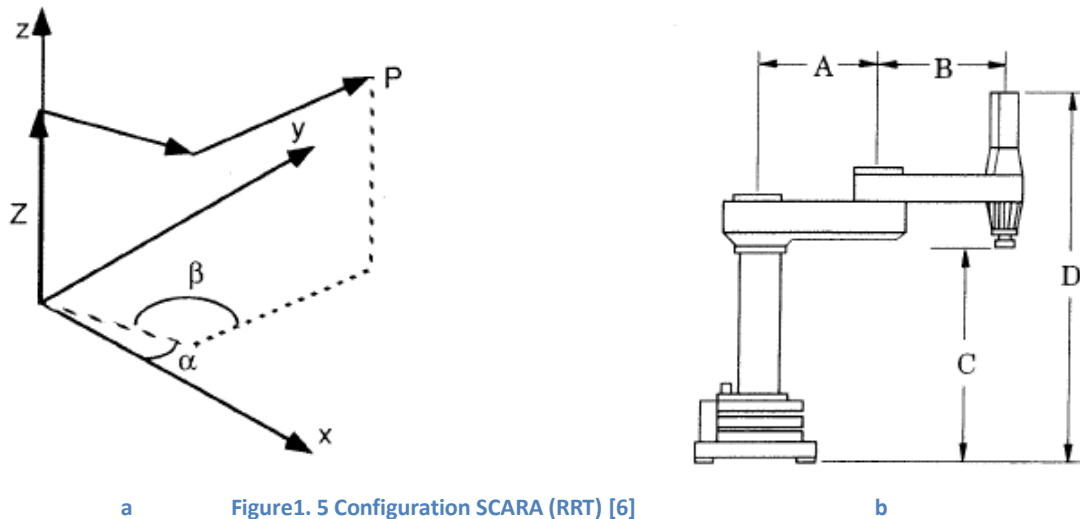
a



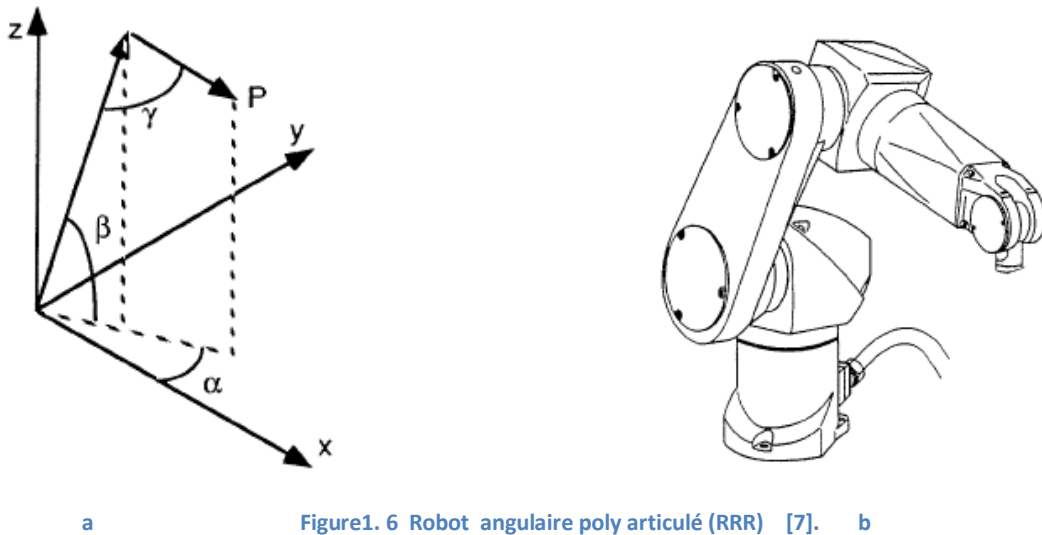
b

Figure1. 4 Robot sphérique (RRT) [6]

D'autres robots industriels disponibles sur le marché présentent la translation sur un axe vertical (Figure 1.5). Cette configuration RRT, dénommée *SCARA* (*Selective Compliance Assembly Robot Arm*), a été développée au Japon dans les années 1980. Elle est la configuration la plus répandue dans le domaine de l'assemblage, alors que le type « tourelle » (Figure 1.5 b) voit son utilisation fortement réduite.



Le système RRR à coordonnées angulaires également appelé robot poly articulé (Figure 1.6 a et b) est certainement le plus simple du point de vue mécanique (formé uniquement par des axes rotatifs), mais il nécessite une commande sophistiquée, par exemple pour effectuer des déplacements en ligne droite [7]. Il faut également apporter une grande attention à la conception des axes rotatifs qui doivent être très rigide pour assurer une fréquence propre élevée, nécessaire à une bonne répétitivité. Certains fabricants (Stäubli par exemple) ont fait en sorte que la totalité du câblage passe à l'intérieur du robot, au travers des articulations qui, de ce fait, deviennent plus compliquées [7].



a

Figure1. 6 Robot angulaire poly articulé (RRR) [7]. b

Les trois axes du porteur ne sont pas forcément indispensables (Figure1.10), si l'on travaille selon un plan, par exemple. Mais l'économie en axes se fait le plus souvent au niveau du poignet. En considérant certaines symétries (un corps cylindrique ne nécessite que 5 coordonnées pour la description de sa position et une sphère n'en nécessite que 3) ou en se limitant à une famille d'opérations simples, on peut réduire le nombre d'axes du robot industriel à 5, 4 ou même 3 [7].

2-1-2/Précision de positionnement

"Il est communément admis que, dans la plupart des cas, la précision d'un robot série est médiocre" [8].

2-2/Robots parallèles

Un robot parallèle est un mécanisme dont l'architecture lui confère des propriétés remarquables c'est un Mécanisme en chaîne cinématique fermée dont l'organe terminal est relié à la base par plusieurs chaînes cinématiques indépendantes.

Les robots parallèles sont une catégorie de robots présentant des chaînes cinématiques fermées. Le champ des architectures cinématiques possibles est très vaste, ainsi que le nombre de domaines d'applications qui s'étendent de la micromanipulation à la manipulation de charges élevées, en passant par des applications industrielles plus classiques telles que le pick-and-place à haute cadence (jusqu'à 3 ou 4 produits déplacés par seconde). Les performances des robots parallèles sont complémentaires et généralement à l'opposé de celles des robots série. Ils ouvrent donc de nouvelles pistes pour la robotisation de certaines tâches. Ils représentent aujourd'hui une faible part de marché pour les robots industriels commercialisés, mais ils s'imposent naturellement pour certaines applications. Pour d'autres, ils sont en

concurrence avec leurs homologues série. Une fois leur architecture cinématique choisie, leur étude demande de la méthodologie et une modélisation fine afin d'évaluer leurs performances. Cet aspect ne doit pas être négligé car les performances sont extrêmement sensibles à leur géométrie, en particulier à cause de la présence de nouveaux types de singularités par rapport aux robots série.

Selon Merlet [9], "un manipulateur parallèle généralisé est un mécanisme en chaîne cinématique fermée dont l'organe terminal est relié à la base par plusieurs chaînes cinématiques indépendantes". De plus, "un manipulateur pleinement parallèle est un manipulateur parallèle dont le nombre de chaînes est strictement égal au nombre de degrés de liberté de l'organe terminal."

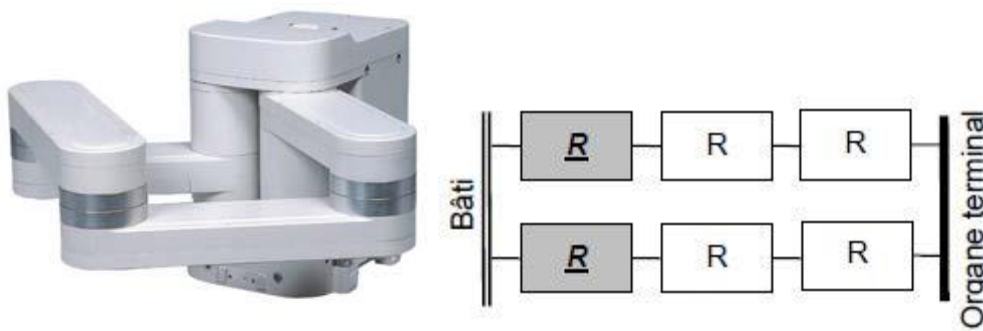


Figure1. 7 Robot parallèle à deux degrés de liberté (Mitsubishi) [7].

Les robots parallèles ont la réputation d'être précis et sont capables de manipuler des charges élevées tout en gardant une excellente rigidité (Figure1.11).

Selon les mécanismes, les actionneurs de ces architectures peuvent être soit fixés sur le bâti, soit en mouvement (dans ce cas, le moteur reste malgré tout proche du bâti). Quoiqu'il en soit, les masses en mouvement sont très faibles en comparaison avec les architectures série ce qui leur confère de très bonnes performances dynamiques. Cependant, leur défaut majeur est leur faible volume de travail par rapport à leur empreinte au sol.

L'évolution récente de ce type d'architecture a connu deux étapes majeures : la création du premier hexapode dans les années 50 et la mise au point du premier "robot parallèle léger" dans les années 80 [10].

- **L'ère des hexapodes**

Le premier représentant de cette famille est la plateforme de Gough [11]. Ce mécanisme à six *ddl* était destiné aux tests du comportement des pneumatiques. Il est composé de six chaînes cinématiques reliant la nacelle au bâti. Chacune de ces chaînes, actionnée par un vérin hydraulique, est reliée d'une part au bâti par une liaison cardan et d'autre part à la nacelle par des liaisons sphériques (Figure 1.12)

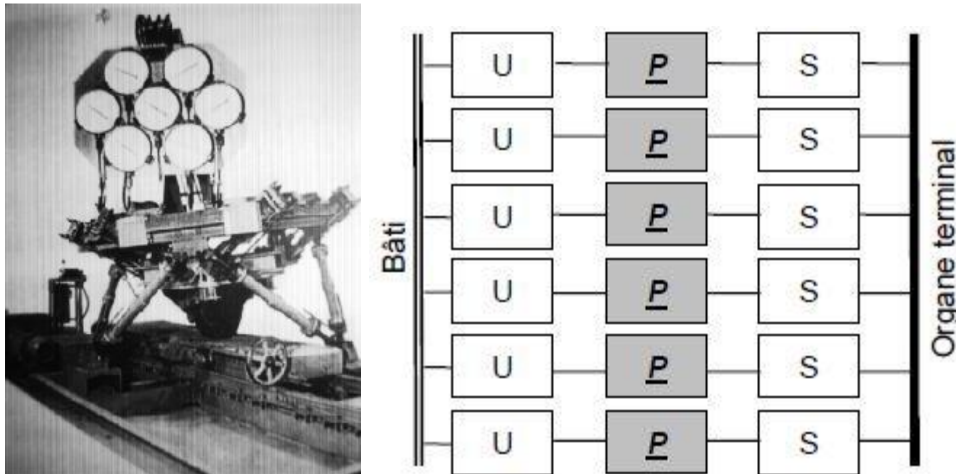


Figure1. 8 Plateforme de Gough [7].

Le principe proposé par Gough a été repris par Stewart [12] dont le but fut de créer un simulateur de vol en utilisant une architecture voisine.

Même si les actionneurs de ces deux architectures sont en mouvement, leur performances dynamiques, bien que limitées, sont déjà meilleures que celles des robots de type série. De plus, Gough et Stewart ont réalisé leurs mécanismes à l'aide d'architectures parallèles afin de répartir les charges transportées sur les six chaînes cinématiques. Ces robots sont donc capables de mouvoir des masses importantes, tout en garantissant une bonne précision.

- **L'ère des robots parallèles légers**

Cette génération de robot est beaucoup plus récente que les hexapodes. Son premier représentant fut le Delta développé par Clavel dans les années 80 [13] à l'Ecole Polytechnique Fédérale de Lausanne (EPFL). Ce fut le premier robot parallèle ayant moins de mobilités que les six *ddl* habituels.

La principale caractéristique de ces robots est non pas la capacité de déplacer de fortes charges, mais celle d'atteindre des dynamiques très élevées. En effet, les masses en mouvement de ces robots sont minimales : les moteurs sont tous fixés sur le châssis et les pièces utilisées sont de masses très réduites.

Les robots parallèles légers commercialisés sont actuellement capables d'atteindre des vitesses de l'ordre de 4 à 5 m/s et des accélérations de 10 g dans toutes les directions du volume de travail. Dans une version utilisant des moteurs à entraînement direct, le prototype de l'EPFL a même été capable d'atteindre des accélérations de 50 g [14]. Cette performance fut réalisée avec une version du robot à trois *ddl* ors de déplacements simples

Un autre exemple de robot parallèle léger est l'Hexa (Figure1.13) [15]. Ce mécanisme à six *ddl* est composé de six chaînes cinématiques actionnées par des moteurs rotatifs.

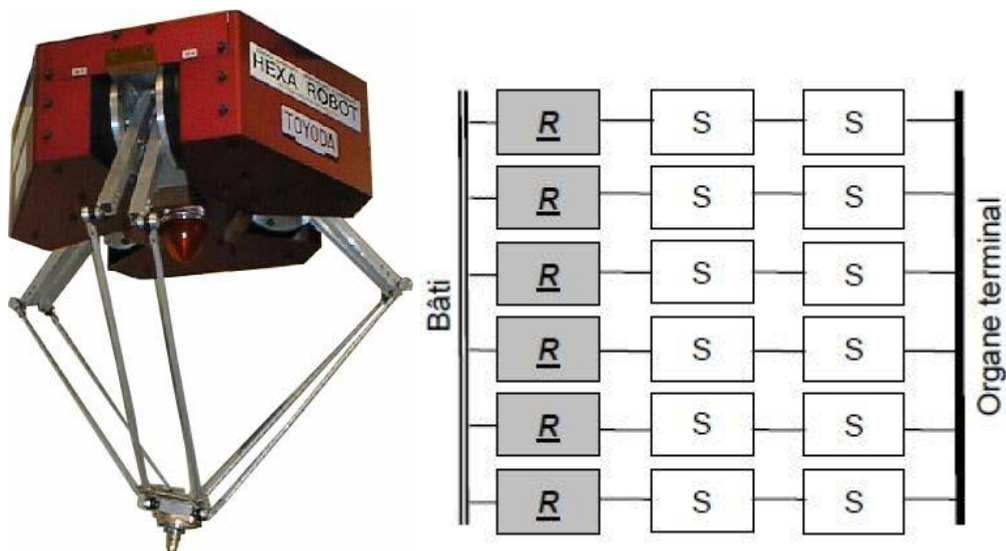


Figure1. 9 Robot Hexa [15]

Grâce à leurs capacités dynamiques très élevées, ces robots sont particulièrement bien adaptés aux applications de manipulation rapide, et en particulier aux applications de pick-and-place.

Les robots industriels les plus rapides actuellement proposés sur le marché du pick and place sont tous des robots Delta. Leurs accélérations maximales étant de l'ordre de 10 g, ils sont adaptés jusqu'à un certain point aux cadences imposées par les lignes de production. Cependant, lorsque les cadences deviennent trop importantes, l'utilisation de deux robots devient indispensable, ce qui a pour effet d'engendrer des surcoûts conséquents.

2-3/ Comparaison des robots parallèles et sériels

2-3-1/ Architectures

Il est communément admis que les robots parallèles ont une meilleure dynamique que les robots série, bien que ces derniers aient un volume de travail plus important.

Nous proposons de comparer à l'aide de données concrètes ces deux familles de robots afin d'évaluer ces différences en termes de dynamique et de volume de travail. Pour cela, nous étudions deux mécanismes à deux *ddl*, l'un sériel et l'autre parallèle. Il est à noter que ces deux robots sont simulés avec des composants exactement identiques (longueurs l , masses m , inerties i_z suivant z au centre de gravité) [16], comme présenté à la (Figure 1.14)

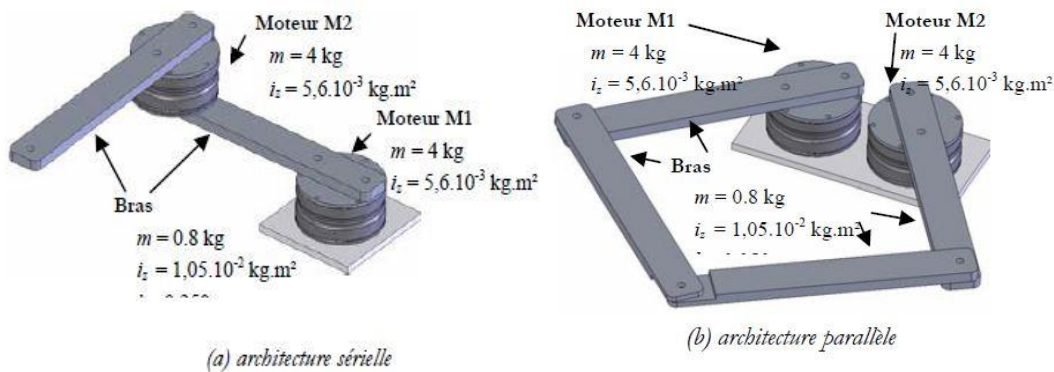


Figure1. 10 Modélisation des robots à 2 degrés de liberté utilisés pour la comparaison [16],

2-3-2/ Volume de travail

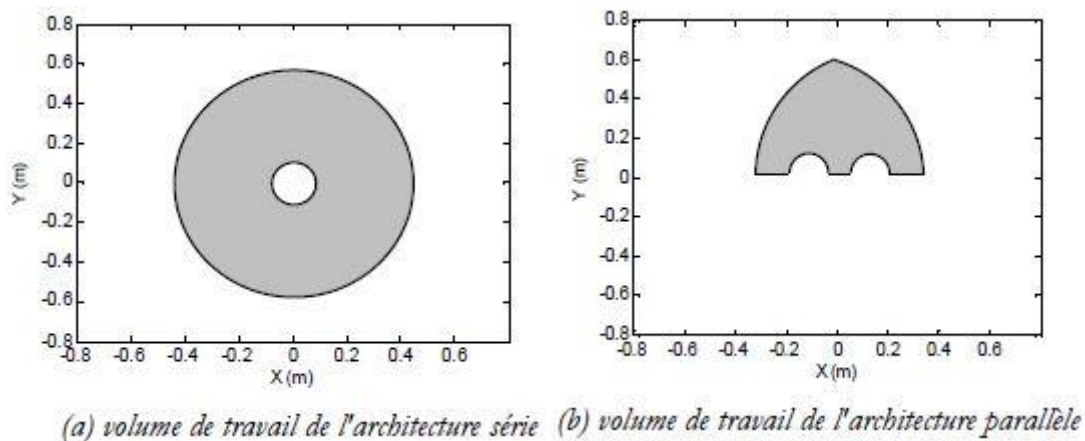


Figure1. 3 Comparaison des volumes de travail [16],

La Figure 1.15 montre clairement l'intérêt majeur des robots de type sériel vis-à-vis de leur volume de travail. En effet, ces architectures permettent de couvrir un espace bien plus important que les robots parallèles. Il est toutefois possible, dans ce cas particulièrement

simple, d'améliorer grandement le volume de travail du robot parallèle en rapprochant les axes des deux moteurs, voire en les plaçant de manière colinéaire.

2-3-3/ Dynamique

La capacité d'un robot à atteindre des dynamiques élevées est directement liée aux couples moteurs mis en jeu lors d'un déplacement. Ainsi, plus les couples engendrés seront faibles, plus le mécanisme aura la capacité d'atteindre des accélérations importantes pour un couple moteur disponible donné.

Dans le cas de Cet exemple, la simulation d'un déplacement identique pour les deux architectures dans des configurations articulaires similaires. Ce déplacement impose à l'organe terminal des robots un mouvement linéaire dont l'accélération est égale à 10 g et la vitesse est de 6 m/s. Les couples des moteurs M1 (Figure 1.16)

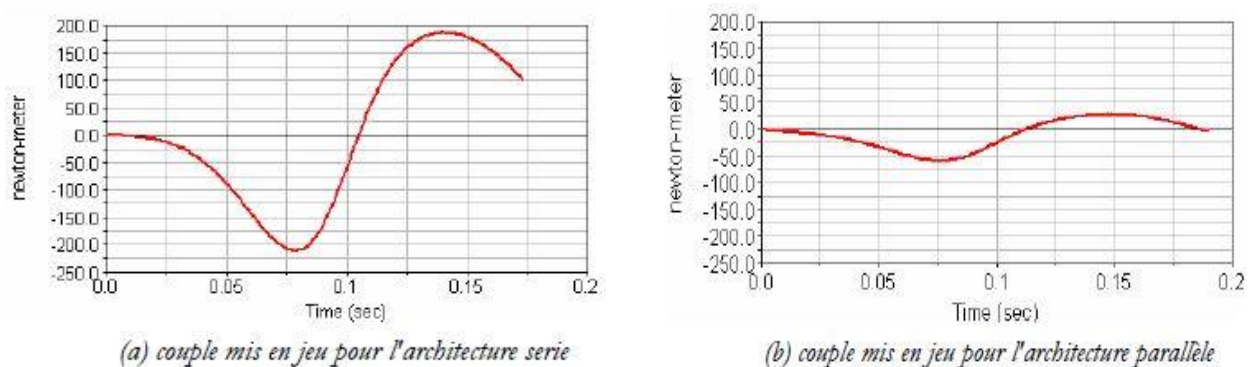


Figure1. 4 Comparaison de l'évolution du couple du moteur M1 [16],

Cette comparaison montre clairement les différences importantes de couples mis en jeu pour un même déplacement pour les deux types d'architectures ; dans le cadre de cet exemple, un robot série devra produire un couple quatre fois supérieur au mécanisme parallèle. Ainsi, dans le cas des applications de pick-and-place à haute cadence, les robots parallèles sont bien plus appropriés, malgré leur faible volume de travail.

Les robots sériels ont des avantages de disposer d'un grand volume de travail et d'être relativement simples à commander, par rapport aux robots parallèles.

Les robots parallèles offrent la possibilité d'atteindre de plus grandes vitesses et accélérations que les robots sériels. Une meilleure respectabilité statique et dynamique ainsi qu'une amélioration de la raideur intrinsèque seraient possibles par ce type d'architecture. Cependant, ces dernières affirmations font encore l'objet de multiples discussions : les performances en termes de précision dépendent des mécanismes comparés, des performances recherchées, de la manière dont sont calculées les performances et des types d'applications envisagées

2-3-4/ Les défauts d'un robot sériel

- manque de raideur par la mise en série d'éléments élastiques
- inertie élevée due aux masses réparties sur toute la chaîne cinématique (actionneurs, organes de transmission)
- faible rapport charge utile/masse du robot. Il est généralement inférieur à 0,15
- cumul des erreurs de positionnement dû à la disposition des segments en série
- agencement délicat des liaisons de puissance assurant l'alimentation des actionneurs (câbles, tuyaux flexibles). Ces éléments déformables longent les segments du robot et subissent une fatigue due aux mouvements du robot
- fatigue et usure des liaisons assurant la circulation des informations entre les capteurs et la commande. Sur le plan de la sécurité, ce point est essentiel puisqu'une erreur de transmission peut avoir des conséquences désastreuses sur les mouvements du robot

2-3-5/ Les avantages d'un robot parallèle

- Possibilité de positionner les actionneurs directement sur la base fixe ou très proche de celle-ci ; cette particularité a les conséquences positives suivantes :
 - grand choix de moteurs et de réducteurs puisque leur masse n'a pas d'influence sur l'inertie du robot en mouvement;
 - simplification importante des problèmes de liaisons entre les moteurs, les capteurs et le contrôleur (câblage plus simple et plus fiable)
 - facilité de refroidissement des actionneurs, donc diminution des problèmes de précision dus aux dilatations, et puissance potentielle élevée
 - facilite d'isolation des moteurs de l'espace de travail pour des activités en atmosphère propre, avec risque de déflagration, ou encore pour des applications en milieux humides
- Faible masse mobile
- Possibilité de mouvements à haute dynamique (accélérations élevées)
- De manière intrinsèque, raideur mécanique élevée (puisque la charge est reprise par l'ensemble des jambes et plus d'éléments ne sont soumis qu'à des sollicitations de traction/compression)
- Capacité de charge élevée
- Facilité d'intégration de capteurs
- Construction mécanique modulaire, simplicité de fabrication et utilisation d'éléments standards par la présence de plusieurs composants identiques sur un robot.

2-3-6/ Les inconvénients des robots parallèles

- Les modèles associés à leur commande sont complexes et fortement couplés à cause de la redondance des chaînes cinématiques
- Ils possèdent un petit espace de travail par rapport à leur empreinte au sol
- De nombreuses singularités peuvent être présentes dans leur espace de travail
- L'utilisation d'un grand nombre de liaisons passives induit des jeux, rendant le comportement du robot difficilement modélisable. Il faut pourtant les maîtriser pour définir des zones dangereuses à proximité des singularités et pour améliorer la précision du robot.

2-4/ Espace de travail :

2-4-1/ Robot parallèle:

L'espace de travail d'un robot parallèle correspond au volume total parcouru par l'effecteur, ou la plateforme mobile, lorsque le manipulateur parcourt l'ensemble des positions articulaires possibles. L'espace de travail est déterminé en fonction des paramètres géométriques du manipulateur. Pour une position des actionneurs, il peut exister plusieurs positions possibles de la plate-forme mobile. Ces différentes positions correspondent à différentes configurations des jambes, appelées modes d'actionnement. Les différents modes d'actionnement sont séparés les uns des autres par des singularités de Type 2, et un aspect de l'espace de travail est associé à chaque mode d'assemblage [17]. De même, pour une position fixée de la plate-forme mobile, il peut exister plusieurs positions des actionneurs, appelées modes de fonctionnement. Les différents modes d'assemblage sont généralement séparés par une singularité de Type 2.

Il est aujourd'hui encore très complexe de déterminer le lieu des singularités de Type 2 pour les robots à plus de trois degrés de liberté. De plus, il est généralement considéré qu'il est impossible pour un mécanisme de traverser ces singularités. Les robots parallèles sont alors limités à une partie de leur espace de travail total, appelée espace de travail opérationnel.

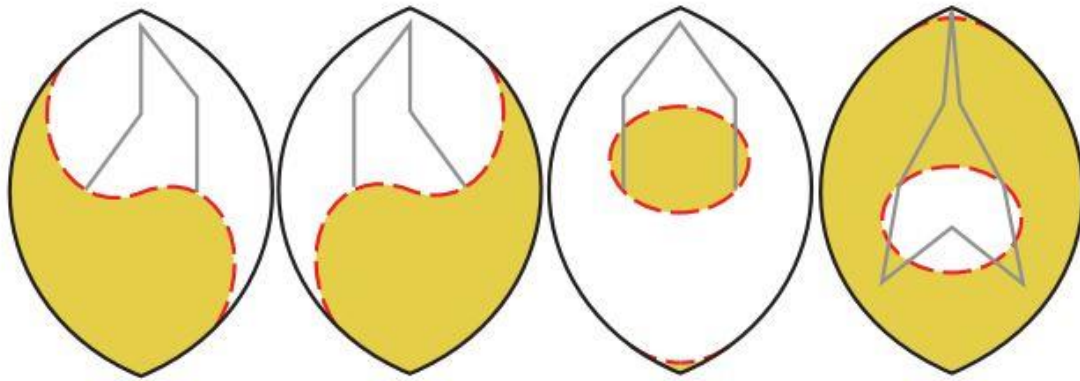


Figure1. 5 Modes de fonctionnement, singularités sérielles (noir), singularités parallèles (pointillés) [17]

Afin d'illustrer l'impact des singularités de Type 2 sur l'espace de travail des mécanismes parallèles, on s'intéresse à un mécanisme plan 5R (mécanisme à cinq barres). La figure 15 représente les différents aspects de l'espace de travail d'un mécanisme 5R sans prendre en compte les éventuelles collisions mécaniques. Pour chacun des quatre modes de fonctionnement du mécanisme, on remarque qu'il existe deux modes d'assemblage. Le premier, représenté en blanc sur la figure 15, est l'espace accessible sans rencontrer de singularité. Le deuxième, représenté en jaune, est séparé du premier par une singularité de Type 2 (ou singularité parallèle), en rouge sur la figure 15. Enfin les singularités sérielles (en noir) délimitent l'espace de travail total. Finalement, on remarque que quel que soit le mode de fonctionnement, l'espace de travail opérationnel est largement inférieur à l'espace de travail total du mécanisme.

2-4-2/ Robot serials:

L'espace de travail d'un robot sériel est défini comme l'espace atteignable par l'effecteur final. La taille de l'espace de travail est fonction de plusieurs facteurs : le nombre de degrés de liberté, la longueur des éléments constituant le robot, la disposition des articulations, les limites des articulations (comme les butées mécaniques) et les interférences mécaniques entre ses éléments [15]. La Figure 2- 5 présente l'espace de travail ou zone atteignable par le centre du poignet du robot IRB 6660 de ABB dans un plan donné.

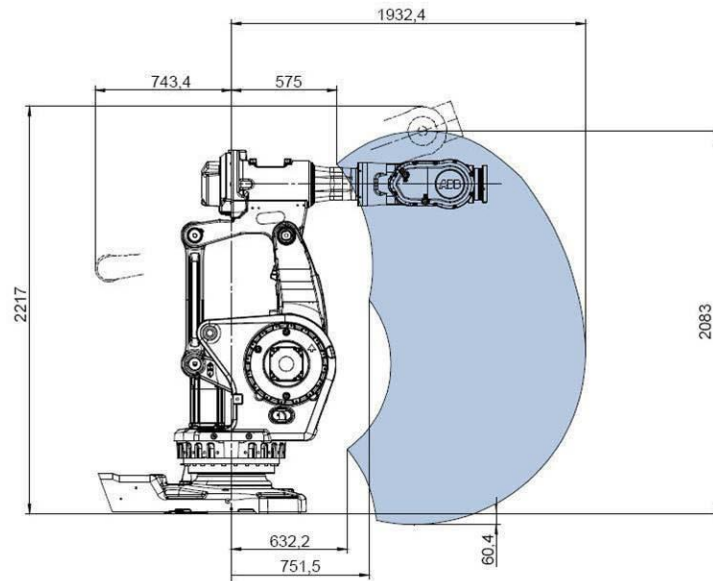


Figure1. 6 Espace de travail du robot IRB 6660 [ABB] [15]

Cet espace de travail permet de vérifier que l'ensemble de la tâche est accessible.

2-5/ Répétabilité de pose

Pour comparer les deux types d'architecture, il faut suivre la procédure suivante:

1. Choisir une paire de robots série/parallèle avec des actionneurs prismatiques ou rotoïdes identiques. On considère alors que les erreurs sur ces actionneurs sont distribuées statistiquement de la même manière
2. Utiliser la théorie des ellipsoïdes stochastiques pour modéliser le phénomène de répétabilité.
3. Définir des critères de comparaison classiques et plus innovants issus de la théorie des ellipsoïdes stochastiques. Ces critères doivent pouvoir aussi ultérieurement être mesurés et estimés physiquement.

2-5-1/ La théorie des ellipsoïdes stochastiques:

Depuis l'introduction des ellipsoïdes de manipulabilité par Yoshikawa, de nombreux travaux ont été publiés autour de ce concept, notamment les ellipsoïdes de vitesse et d'effort obtenus en considérant la transformation mécanique du système vitesse/couple articulaire dans l'espace opérationnel. Cette approche cinématique est intéressante pour repérer des points singuliers ou pour choisir les axes les plus adaptés pour commander le robot en vitesse. Cependant elle souffre de limitations. Tout d'abord, elle ne tient pas suffisamment compte des caractéristiques de performance intrinsèques à chaque articulation. Puis, elle n'est pas

applicable à un robot ayant plusieurs types d'actionneurs rotoïde/prismatique en raison de l'hétérogénéité des grandeurs physiques qui interviennent.

La répétabilité peut être modélisée par la théorie des ellipsoïdes stochastiques qui s'applique aussi bien aux robots série qu'aux robots parallèles.

Dans le cas d'un robot série, la commande du robot nécessite la connaissance de son modèle géométrique direct, liant coordonnées articulaires X et articulaires θ : $X=F(\theta)$

2-6/Précision absolue de pose

Lors de la réalisation de la tâche, le robot doit garantir une précision compatible avec le processus à réaliser.

Les robots sont généralement conçus en fonction de critères tels que l'accessibilité, la taille de l'espace de travail et la sollicitation de la tâche. Ces critères permettent de définir le nombre d'axes, la charge que doit supporter le robot et le niveau de répétitivité à atteindre. Toutefois, la précision absolue (*absolute ou volumetric accuracy*, en anglais) est rarement spécifiée.

Cette précision est à comparer avec les exigences des différents procédés d'usinage qui peuvent être réalisés sur un robot. Le Tableau 1.1 donne une idée de l'ordre de grandeur de différents procédés d'usinage.

Tableau 1. 1 Tolérances commerciales des procédés d'usinage

Processus	Tolérances
commerciales ($\pm mm$)	Tournage 0,025 – 0,13
Alésage, Forage 0,025	Perçage 0,075
Fraisage 0,13 – 0,25	Façonnage 0,05 – 0,13
Brochage 0,025 – 0,15	Tronçonnage 0,8
Processus	Tolérances
commerciales ($\pm mm$)	Tournage 0,025 – 0,13
Alésage, Forage 0,025	Perçage 0,075

La précision de pose de l'outil doit être inférieure à la qualité d'usinage attendue afin de maîtriser la précision d'usinage. En effet, si nous considérons un assemblage à réaliser entre deux pièces de quelques microns, il est nécessaire d'avoir une exactitude et une répétitivité du moyen de production de l'ordre du micron afin d'assurer un positionnement relatif entre les deux pièces compatibles avec la qualité attendue.

De plus, le comportement in situ du robot évolue par rapport à celui spécifié par le constructeur. En effet, l'étalonnage réalisé par le fabricant dans un environnement maîtrisé à l'aide d'équipements dédiés n'est pas forcément optimum lorsque le robot est en situation de

production. Il est alors nécessaire de proposer des méthodes d'étalonnage applicable dans un environnement industriel.

Les fabricants de robot développent ainsi des procédures particulières permettant d'améliorer l'exactitude de pose de l'outil sur une ligne de production. La compagnie ABB a développé une option nommée « Absolute Accuracy » pour assurer une précision de positionnement du *TCP* (Tool Center Point) d'au moins 0,1 mm dans tout l'espace de travail. Cette méthode est basée sur la mesure de la pose de l'effecteur de l'outil à 100 emplacements différents dans tout l'espace de travail. Les positions réelles de l'effecteur sont déterminées à l'aide de mesure de position par un système laser. Un ensemble de paramètres de compensation est identifié. Ces paramètres ne sont pas détaillés par ABB et semblent prendre en compte les imperfections mécaniques et les déformations causées par le poids des éléments du robot et les charges embarquées .

Dans le cas de Fanuc robotics, plusieurs logiciels pour l'étalonnage des robots comme *iRVision* et *iR Calibration* ont été développés afin d'obtenir une précision dans l'ordre de 0,1 mm [18].

D'autres produits sont commercialisés pour l'étalonnage de robots. Ainsi l'outil *Isios Robotics* semble permettre d'atteindre une précision de l'ordre de 0,1 mm et jusqu'à 0,06 mm [19] Cette méthode est basée sur la mesure de la position d'un capteur laser placé directement sur l'effecteur final du robot, lorsque l'extrémité du robot est déplacée le long du faisceau laser, les plus petits écarts par rapport à la ligne droite sont détectés aisément. Les mesures de ces écarts donnent un modèle du comportement du robot. *Nikon Metrologya* le produit « *K-Series robots MMC system* » qui offre la possibilité d'atteindre une précision de 0,06 mm dans un espace de travail de 17 m³ grâce à la mesure de LED infrarouges en utilisant une technologie par vision à l'aide d'une caméra.

Ainsi, nous pouvons constater qu'il existe une multitude de procédures industrielles permettant d'étalonner un robot. Toutefois, ces méthodes ne sont pas formalisées et peuvent rester perfectibles car nécessitant des précautions particulières non maîtrisées par l'utilisateur final.

Les causes des erreurs de précision de pose des robots ont été largement étudiées par des chercheurs qui les ont classées, de façon générale, suivant plusieurs facteurs.

Nous pouvons constater que les causes des erreurs de précision de pose des robots proviennent de deux sources : les causes relatives à la géométrie intrinsèque du robot et les causes liés aux facteurs d'utilisation du robot.

Pour synthétiser, le Tableau 1.2 résume les différents types de causes des erreurs de précision pour les robots sériels et les phénomènes associés.

Tableau 1. 2 Types de causes des erreurs.

<i>Types de causes des errors Phénomene</i>	Phénomène
Géométrique	- Longueurs des éléments du robot et orientation des liaisons.
Non-géométrique	- Le frottement. - La raideur
Environnementaux	- La température.

Dans le cas de l'usinage avec des robots à structure parallèle, Pritschow propose la classification de différents comportements d'une cellule d'usinage suivant l'origine des erreurs de pose de l'effecteur de façon cohérente avec les travaux précédents Figure 1.19

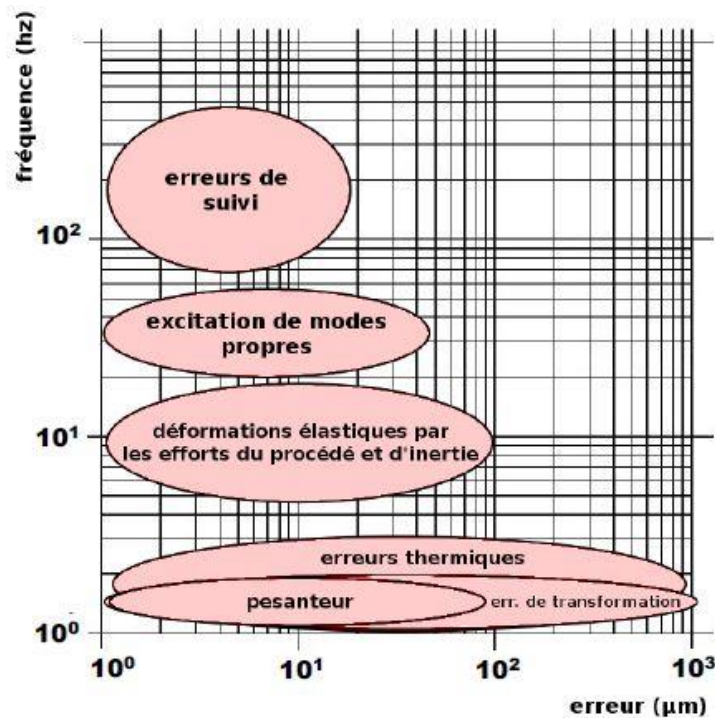


Figure1. 7 Amplitude et fréquence des erreurs d'usinage et ses causes [19]

La précision des robots sériels, à l'heure actuelle, ne répond pas aux exigences en termes de précisions pour des opérations d'usinage. En effet, les processus répandus de fabrication ont des tolérances entre 25 et 50 µm. Les facteurs affectant la précision sont de diverses natures et l'influence de chacun de ces facteurs n'est pas totalement identifiable. Bien que ces facteurs soient multiples, les facteurs associés à la géométrie du robot sont généralement invariants dans le temps, ce qui facilite la mise en place d'une stratégie ou méthode de compensation

De plus, les erreurs associées aux paramètres géométriques ont été signalés comme les erreurs les plus influentes dans la précision de pose de l'outil.

Chapitre II

Modélisation géométrique

3/Introduction:

La modélisation d'un robot, considéré comme étant un système mécanique articulé, actionné et commandé, consiste à en établir un modèle mathématique. Outre une fonction générale d'aide à la conception, elle a de multiples utilisations pour la prédiction des mouvements, l'adaptation des actionneurs, la planification des tâches, l'établissement des lois de commande, l'incorporation du robot dans des simulations informatiques, etc. Dans le langage courant, la modélisation précède la simulation sans que l'on distingue une séparation nette entre ces deux activités.

La conception et la commande des robots nécessitent le calcul de certains modèles mathématiques, tels que :

- les modèles de transformation entre l'espace opérationnel (dans lequel est définie la situation de l'organe terminal), et l'espace articulaire (dans lequel est définie la configuration du robot), on distingue parmi ces modèles :

1 -Les modèle géométrique direct et inverse, qui expriment la situation de l'organe terminal en fonction de variable articulaire du mécanisme et inversement.

2 -Les modèle cinématique directe et inverse exprime la vitesse de l'organe terminal en fonction des vitesses articulaire et inversement.

3 -les modèle dynamique définissent l'équation du mouvement du robot, qui permettent d'établir la relation entre le couple du force exercée par les actionneurs et les positions, vitesses et accélérations des articulations.

Le formalisme mathématique fait appel aux matrices de transformation homogènes de dimension (4x4). La matrice homogène ${}^i T_j$ représente la transformation permettant de passer du repère R_i au repère R_j

$${}^i T_j = \begin{bmatrix} {}^i A_j & & & \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} {}^i s_j & {}^i n_j & {}^i a_j & {}^i p_j \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.1)$$

Ou ${}^i s_j$, ${}^i n_j$, et ${}^i a_j$ désignent respectivement les vecteur unitaires suivent les axes x_j , y_j et z_j du repère R_j exprimés dans le repère R_i , et ou ${}^i p_j$ est le vecteur exprimant l'origine du repère R_j , les vecteur ${}^i s_j$, ${}^i n_j$, ${}^i a_j$ de la matrice d'orientation ${}^i A_j$ sont les cosinus directeur. [20].

4/Modélisation Géométrique

Le bras manipulateur à étudier est un robot articulé à 6 d. d.

La cinématique du porteur est de type RRR et le poignet comporte trois rotations.

On détermine ensuite les paramètres géométriques du robot. Le placement des repères est indiqué sur la Figure 2.1 et les paramètres géométriques sont donnés dans le tableau 2.1 [20].

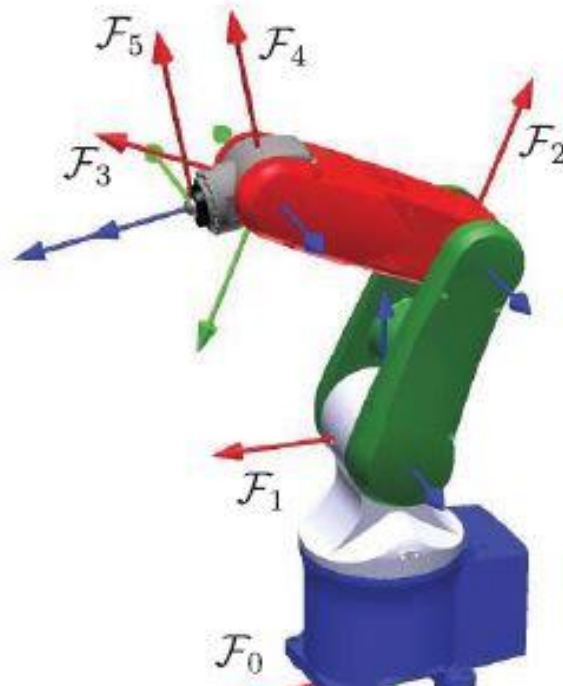


Figure 2. 1 placement des repères et notation pour le robot [21].

Tableau 2. 1 Paramètres géométriques du robot

Joint	θ_i (°)	d_i (mm)	a_i (mm)	t_i (°)
1	θ_1	0	75	-90
2	$\theta_2 - 90$	0	300	180
3	$\theta_3 + 180 + \theta_2$	0	-75	90
4	$\theta_4 + 180$	-320	0	90
5	θ_5	0	0	-90
6	θ_6	-80	0	180

5/Cinématiques directe:

La cinématique direct se réfère à l'utilisation des équations cinématiques d'un robot pour calculer la position de l'effecteur à partir des valeurs spécifiées pour les paramètres de l'articulation[20].

Les équations cinématiques du robot sont utilisées dans la robotique, les jeux informatiques et l'animation. Le processus inverse qui calcule les paramètres de joint qui atteignent une position spécifiée de l'effecteur est connu sous le nom de cinématique inverse[20]

Dans ce chapitre, on va parlez sur une méthode systématique pour développer le modèle cinématique d'un robot sériel à n articulations. Cette méthode consiste en deux étapes principales : placer un référentiel sur chaque lien du robot et ensuite, pour chaque paire de référentiels consécutifs, trouver quatre paramètres géométriques qui décrivent la pose de l'un par rapport à l'autre. Il s'agit de la méthode Denavit-Hartenberg qui à été proposée en 1955 et reste toujours la méthode de loin la plus utilisée en robotique sérielle.[21]

5-1/La méthode Denavit-Hartenberg:

Pour résoudre la cinématique directe d'un robot sériel spatial à n articulations (rotoïdes ou prismatiques), il est presque inévitable de placer un référentiel sur chaque lien du robot. Ainsi, dans le cas d'un robot à n articulations, donc à n+1 liens, le modèle de la cinématique directe du robot sera exprimé par l'équation suivante:

$$H_n^0 = H_1^0 H_2^1 H_3^2 \dots \dots \dots H_n^{n-1} (2.2)$$

ou H_n^0 représente la pose du référentiel F_n (celui de la bride du robot, qu'ABB appelle tool0) par rapport au référentiel F_0 (celui de la base). Le problème principal avec un choix aléatoire de référentiels est que l'obtention des matrices H_i^{i-1} serait assez laborieuse. [21]

Un autre problème est bien sur le fait qu'une telle approche ne serait pas systématique.

Dans les années 1950s, les messieurs Jacques Denavit and Richard Hartenberg ont eu l'excellente idée de proposer une méthode simple et systématique pour placer des référentiels sur chaque lien d'un mécanisme sériel qui facilite énormément le calcul des matrices H_i^{i-1} .

Leur idée était de placer les référentiels de telle manière que l'axe z_i (ou $i = 0; 1; 2; ; ; n-1$) est le long de l'axe de l'articulation $i + 1$, dans le cas d'une articulation rotoïdes, ou parallèle à la direction de l'articulation $i + 1$, dans le cas d'une articulation prismatique, et de contraindre l'axe x_i d'intersecté l'axe z_{i-1} à angle droit. Grace à cette double contrainte (non seulement intersecté mais aussi à angle droit), nous avons besoin de seulement quatre paramètres géométriques, plutôt que six, pour décrire la pose du référentiel F_i par rapport au référentiel F_{i-1}

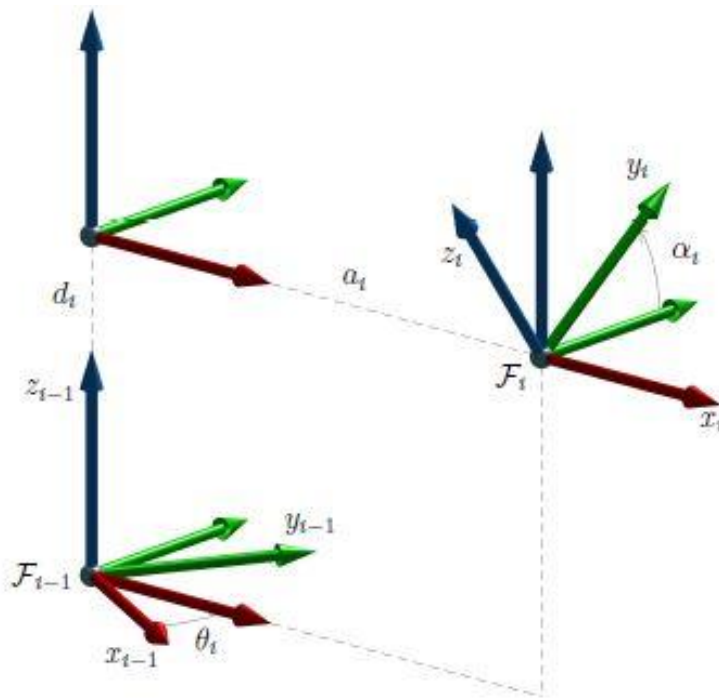


Figure 2. 2 Les quatre transformations consécutives de base pour obtenir F_i à partir de F_{i-1} [21]

La Figure 2.2 démontre deux référentiels arbitraires, F_i et F_{i-1} , qui respectent cette contrainte cette figure illustre clairement que nous pouvons obtenir le référentiel F_i à partir du référentiel F_{i-1} , l'aide de seulement quatre transformations consécutives simples (translation le long d'un axes x , y ou z , ou rotation autour d'un axe x , y ou z). Ces quatre transformations peuvent être faites dans l'ordre suivant:

1. rotation de θ_i autour de l'axe z_{i-1}
2. translation de d_i le long de l'axe z du nouveau référentiel
3. translation de a_i le long de l'axe x du nouveau référentiel
4. rotation de α_i autour de l'axe x du nouveau référentiel

Nous allons appeler θ_i , d_i , a_i et α_i , les quatre paramètres DH. Ainsi, la matrice H_i^{i-1} qui représente la pose du référentiel F_i par rapport au référentiel F_{i-1} peut être obtenue en fonction des quatre paramètres DH par l'équation suivante:

$$\begin{aligned}
 H_i^{i-1} &= H_{rot,z}(\theta_1)H_{trans}(0,0,d_1)H_{trans}(a_1,0,0)H_{rot,x}(\alpha_1) \\
 &= \begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (2.3)
 \end{aligned}$$

5-2/ Équation de la cinématique directe:

Une fois les paramètres DH obtenus, il ne reste que de les substituer dans l'équation (2.3) pour obtenir les matrices $H_1^0, H_2^1, \dots, H_n^{n-1}$ et les substituer dans l'équation (2.2). Il est important de comprendre que même si les référentiels F_0 et F_n sont déjà choisis (par exemple, par le fabricant du robot), il existe tout de même un nombre infini de possibilités pour les autres référentiels DH et, par conséquent, pour les matrices H_i^{i-1} . Cependant, le résultat final (la matrice H_n^0) devrait être le même.

Enfin, si vous avez un référentiel outil, et un référentiel atelier, la pose du premier par rapport au dernier est définie par l'équation suivante:

$$H_{outil}^{atelier} = H_0^{atelier} H_n^0 H_{outil}^n \quad (2.4)$$

5-3/ Etude cinématique direct du robot ABB IRB 2600

L'IRB 2600 permettra d'enrichir les fonctions, de distribution, d'usinage, de mesure, de montage et de soudage à l'arc.

Le modèle 3D du manipulateur utilisé dans cette étude est montré à la Figure 2.3. Il s'agit d'un manipulateur ABB IRB 260 à six articulations rotatives actionnées par des moteurs AC sans brosse. La lecture de l'angle de rotation de chaque articulation est fournie par un résolveur directement monté sur l'arbre de chacun des moteurs.

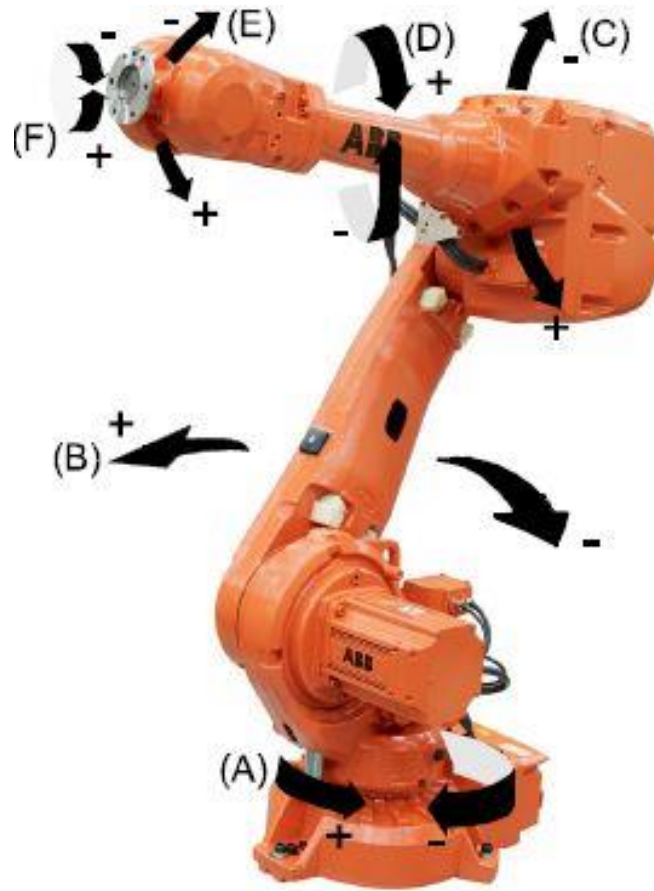


Figure 2. 3le robot IRB 2600

Tableau 2. 2 Les axes du manipulateur

Rep	Description	Rep	Description
A	Axe 1	B	Axe 2
C	Axe3	D	Axe 4
E	Axe5	F	Axe 6

Une fois les référentiels DH placés, la prochaine étape est de trouver les paramètres DH en se référant aux Figures 2.3 et Figure 2.4. Encore une fois, il est nécessaire de s'imaginer le robot avec ses référentiels DH bougé.

Tableau 2. 3 paramètres DH DE robot ABB IRB 2600

Joint	α	a	θ_i	d
1	0	0	θ_1	450
2	$-\pi/2$	150	$\theta_2 - \pi/2$	0
3	0	900	θ_3	0
4	$-\pi/2$	115	θ_4	795
5	$\pi/2$	0	θ_5	0
6	$-\pi/2$	0	$\theta_6 + \pi$	0

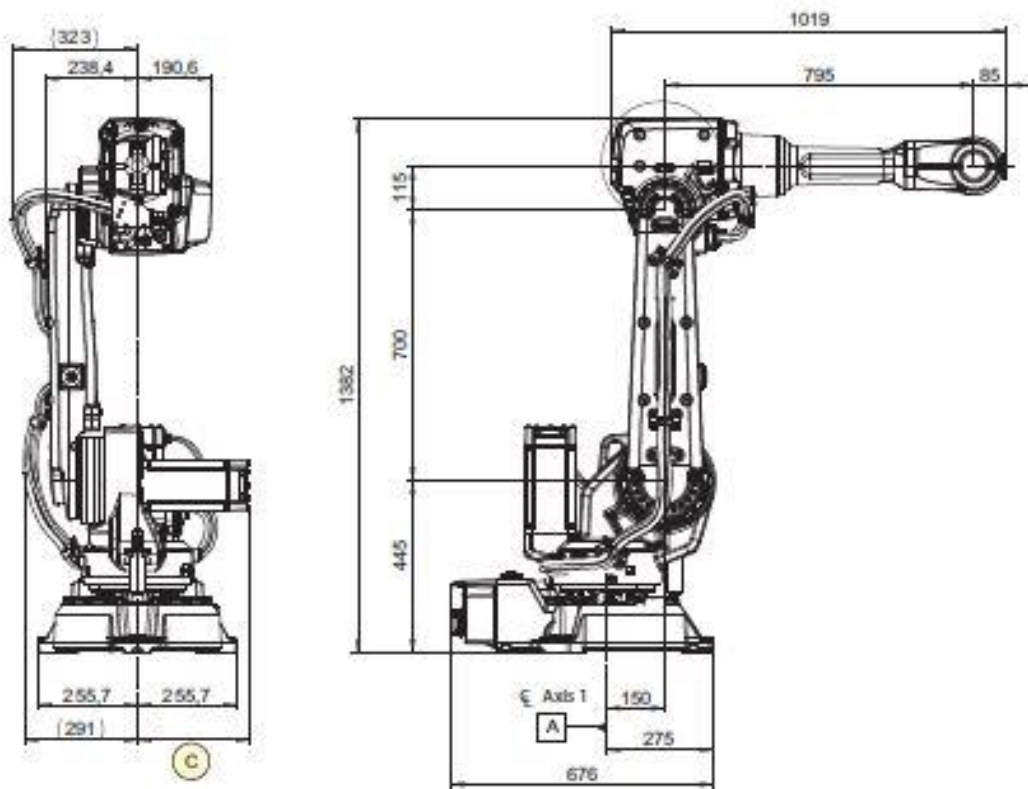


Figure 2. 4 les dimensions du robot ABB IRB 2600

Les paramètres DH sont données au tableau 2.3.

La prochaine étape est de substituer les paramètres DH du tableau 2.2 dans l'équation (2.3)

$$H_1^0 = \begin{pmatrix} \cos \theta_1 & -\sin \theta_1 & 0 & 0 \\ \sin \theta_1 & \cos \theta_1 & 0 & 0 \\ 0 & 0 & 1 & 450 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.5)$$

$$H_2^1 = \begin{pmatrix} \sin \theta_2 & 0 & \cos \theta_2 & 150 \sin \theta_2 \\ -\cos \theta_2 & 0 & \sin \theta_2 & -150 \cos \theta_2 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.6)$$

$$H_3^2 = \begin{pmatrix} \cos \theta_3 & -\sin \theta_3 & 0 & 900 \cos \theta_3 \\ \sin \theta_3 & \cos \theta_3 & 0 & 900 \sin \theta_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.7)$$

$$H_4^3 = \begin{pmatrix} \cos \theta_4 & 0 & -\sin \theta_4 & 115 \cos \theta_4 \\ \sin \theta_4 & 0 & \cos \theta_4 & 115 \sin \theta_4 \\ 0 & -1 & 0 & 795 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.8)$$

$$H_5^4 = \begin{pmatrix} \cos \theta_5 & 0 & \sin \theta_5 & 0 \\ \sin \theta_5 & 0 & -\cos \theta_5 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.9)$$

$$H_6^5 = \begin{pmatrix} -\cos \theta_6 & 0 & -\sin \theta_6 & 0 \\ -\sin \theta_6 & 0 & \cos \theta_6 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.10)$$

Enfin, il ne reste que de substituer les matrices ci-dessus dans l'équation (2.2) et faire quelques manipulations trigonométriques ont utilisant le logiciel Matlab pour calcule cette matrice Figure 2.5:

```

1 - clear all
2 - clc
3 - syms t1 t2 t3 t4 t5 t6
4
5 - a1=[cos(t1) -sin(t1) 0 0;sin(t1) cos(t1) 0 0;0 0 1 450;0 0 0 1];
6 - a2=[sin(t2) 0 cos(t2) 150*sin(t2); -cos(t2) 0 sin(t2) -150*cos(t2) ;0 -1 0 0; 0 0 0 1];
7 - a3=[cos(t3) -sin(t3) 0 900*cos(t3); sin(t3) cos(t3) 0 900*sin(t3) ;0 0 1 0; 0 0 0 1];
8 - a4=[cos(t4) 0 -sin(t4) 115*cos(t4);sin(t4) 0 cos(t4) 115*sin(t4); 0 -1 0 795; 0 0 0 1];
9 - a5=[cos(t5) 0 sin(t5) 0; sin(t5) 0 -cos(t5) 0 ;0 1 0 0; 0 0 0 1];
10 - a6=[-cos(t6) 0 -sin(t6) 0 ; -sin(t6) 0 cos(t6) 0; 0 -1 0 0; 0 0 0 1];
11
12 - a=a1*a2*a3*a4*a5*a6
    
```

Figure 2. 5algorithme de Matlab pour calculer H⁰₆

$$H_6^0 = H_1^0 H_2^1 H_3^2 H_4^3 H_5^4 H_6^5 =$$

$c6(c12s5-c34s12c5)+s12s34s6,$ $c6(s12s5+c12c34c5)-c12s34s6,$ $c34s6+s34c5c6,$ $0,$	$-c12c5-c34s12s5,$ $c12c34s5-s12c5,$ $s34s5,$ $0,$	$s6(c12s5-c34s12c5)-s12s34c6,$ $s6(s12s5+c12c34c5)+c12s34c6,$ $s34c5s6-c34c6,$ $0,$	$795c12+150s12+115c34s12+900s12c3$ $795s12-150c12-115c12c34-900c12c3$ $450-900s3-115s34$ 1
--	---	--	---

6/ Cinématiques inverse du robot ABB IRB 2600

La cinématique inverse est la partie la plus difficile durant la modélisation. Contrairement à la cinématique directe, il n'existe pas de méthodes systématiques pour résoudre la cinématique inverse d'un robot sériel sous forme analytique. D'ailleurs, il n'est

même pas possible de résoudre la cinématique inverse, sous forme analytique, de certains robots sériels. Par contre la solution de la cinématique inverse d'un robot sériel implique uniquement la solution des équations trigonométriques relativement simples. Souvent, ce qui est difficile n'est pas de résoudre la cinématique inverse sous forme analytique, mais de le faire de manière élégante, avec une compréhension géométrique des résultats, et sans oublier aucun cas particulier.

Pour résoudre la cinématique inverse d'un robot à n articulation, il faut généralement résoudre sa cinématique directe et poser l'équation suivante :

$$H_{outil}^{atelier} = H_0^{atelier} H_n^0 H_{outil}^n$$

ou les $H_0^{atelier} H_{outil}^n$ sont connues, ainsi que la matrice $H_{outil}^{atelier}$ alors que la matrice H_n^0 est connue sous forme d'une expression en fonction des n variables articulaires qui sont maintenant considérées comme inconnues.

Cette équation peut être réécrite sous la forme suivante :

$$H_n^0 = (H_0^{atelier})^{-1} H_{outil}^{atelier} (H_{outil}^n)^{-1} = \begin{pmatrix} n_x & o_x & a_x & p_x \\ n_y & o_y & a_y & p_y \\ n_z & o_z & a_z & p_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (2.11)$$

$$H_6^0 = (H_0^{atelier})^{-1} H_{outil}^{atelier} (H_{outil}^6)^{-1} = \begin{pmatrix} 90 & 0 & -90 & 50 \\ 90 & 45 & 0 & 30 \\ 45 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (6)$$

L'équation (6) a été calculée en utilisant le logiciel Matlab.

Pour conclure la cinématique inverse, nous allons présenter les variables articulaires pour les quatre solutions qui correspondent au tableau 2.4

Tableau 2. 4 solutions de la cinématique inverse

θ_1	θ_2	θ_3	θ_4	θ_5	θ_6
30.5	60.95	40	90	20	40.256

Chapitre III

Développement de l'Interfaces graphique

7/Programmation:

Dans le domaine de l'informatique, la programmation est l'ensemble des activités qui permettent l'écriture des programmes informatiques. C'est une étape importante du développement de logiciels.

Pour écrire un programme, on utilise un langage de programmation. Un logiciel est un ensemble de programmes (qui peuvent être écrits dans des langages de programmation différents) dédié à la réalisation de certaines tâches par un (ou plusieurs) utilisateurs du logiciel.

La programmation représente donc ici la rédaction du (ou des) code source d'un logiciel. On utilise plutôt le terme développement pour dénoter l'ensemble des activités liées à la création d'un logiciel et des programmes qui le composent (cela inclut la spécification du logiciel, sa conception, puis son implémentation proprement dite au sens de l'écriture des programmes dans un langage de programmation bien défini et aussi la vérification de sa correction).

7-1/Programmation du robot:

La programmation du robot a très vite progressé depuis ces dernières années. Dans cette étude, la programmation du robot se limitera à la configuration appelée « six axes ».

Les méthodes de programmation du robot sont généralement classées en deux familles :

1. La programmation en ligne (Programmation par apprentissage)
2. La programmation hors-ligne

Les questions liées à la programmation du robot ainsi que les avantages de chaque mode de programmation seront présentés dans les paragraphes suivants.

7-2/Programmation en ligne:

La programmation en ligne est toujours la méthode de programmation la plus utilisée dans les applications industrielles. Il s'agit de la programmation par apprentissage. Son principe est de montrer manuellement au robot ce qu'il doit faire par l'opérateur. Dans des applications de revêtement et projection thermique on demande généralement l'opérateur de manipuler le robot à l'aide d'un pupitre d'apprentissage et de bouger le CDO (centre d'outil)

du robot jusqu'à la position désirée. Cette opération continue jusqu'à ce que la trajectoire couvre toute la surface à revêtir. Une série de points est ensuite enregistrée selon l'ordre d'exécution dans la mémoire du contrôleur de robot. La trajectoire indiquée par l'opérateur sera répétée automatiquement par le robot afin d'accomplir tous le travail.

Cette technique est très pratique car elle permet de réaliser la programmation rapidement et facilement. En plus, elle peut être maîtrisée par un personnel non qualifié(Figure 3.1).

On estime à 80% le nombre de robots qui sont programmés de cette façon dans les applications industrielles.



Figure 3. 1 La programmation en ligne

Par contre il existe des limitations de cette méthode de programmation:

- L'acquisition de la trajectoire est réalisée avec le déplacement manuel du robot dans la cellule robotisée , c'est-à-dire qu'il faut arrêter la tâche en cours au moment de la programmation.
- En cas de trajectoires définies sur des pièces relativement complexes, il est difficile d'acquérir précisément les points des trajectoires manuellement dans un espace 3D, surtout si l'on veut travailler avec des distances ou des angles de projection constants.

La programmation en ligne est une méthode de programmation classique et pratique dans les cas où la pièce à revêtir ou à usiner est très simple de géométrie. Mais dès que la complexité s'accroît (courbe, angle droit, etc.) il est presque impossible de paramétrer manuellement la trajectoire de façon précise. Et c'est là qu'intervient la programmation hors-ligne pour mettre en œuvre la stratégie afin de générer des trajectoires automatiquement et pour assurer la précision de la programmation.

7-3/Programmation hors-ligne

La programmation hors-ligne correspond à la génération d'un programme robot à exécuter sur site à partir des données CAO (Conception Assistée par Ordinateur) générées avec un logiciel sur ordinateur notamment le logiciel virtuel RobotStudio™ équipé d'une bibliothèque du robot. Ce type de programmation est en train de devenir une tendance pour des nombreuses applications robotiques.

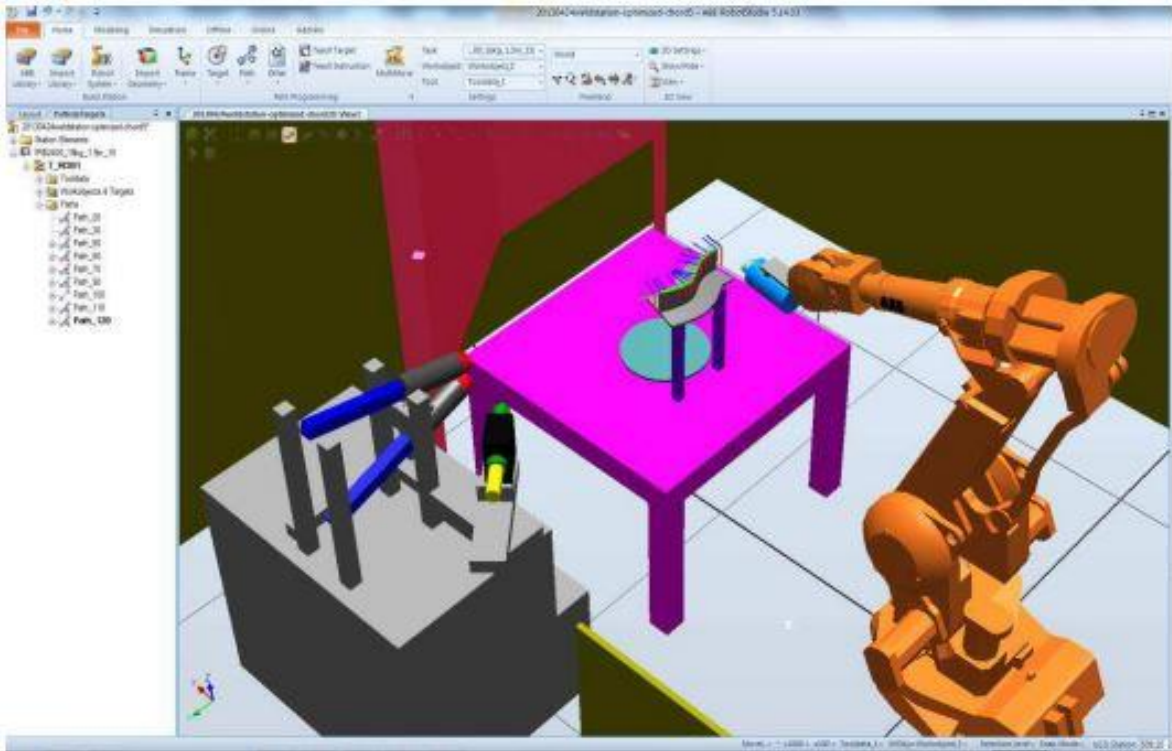


Figure 3. 2 Logiciel de programmation hors-ligne

Le principe de la programmation hors-ligne est de générer un modèle de la pièce en 3D avec un logiciel CAO ou par l'ingénierie inverse. Ce modèle est ensuite importé dans le logiciel de programmation hors-ligne. Grâce à l'environnement graphique, le logiciel permet de spécifier le type de robot et de modéliser l'environnement de la cellule de travail, l'outil de travail, etc. La trajectoire du robot est ensuite générée à partir du modèle 3D et enregistrée en format du code exécutable pour qu'il puisse être exécuté par le robot. La Figure 3.2 représente l'environnement virtuel modélisé dans un logiciel de programmation hors-ligne d'ABB :

La méthodologie de la programmation hors-ligne peut être décrite par la Figure 3.3.



Figure 3. 3 Méthodologie de programmation hors-ligne

Les caractéristiques de la programmation hors-ligne doivent permettre :

- D'assurer la sécurité humaine : l'environnement virtuel est une réplique exacte de la cellule du robot, donc le programme du robot peut être conçu dans le bureau de l'opérateur, ce qui permet de l'éloigner de lieux où peut subsister un quelconque danger.
- De diminuer le temps de cycle: le programme du robot peut être préparé à l'avance, ou être conçu malgré l'indisponibilité du robot (en production par exemple), ce qui réduit le temps de préparation.
- D'augmenter la précision du programme : l'environnement numérique fournit la possibilité d'intégration d'outil extensionnel pour analyser la surface à revêtir, ce qui augmente la précision de la programmation. L'analyse numérique présente la possibilité d'optimiser la trajectoire du robot pour l'adapter au besoin spécifique.
- De diminuer le risque de collision entre l'outil de travail, le robot et la pièce dans la cellule : le programme du robot peut être simulé dans le logiciel de programmation hors-ligne pour visualiser la détection de la collision avant l'exécution dans la cellule sur site.

- De faciliter la programmation du robot : grâce à la duplication de la cellule de robot, le programme préparé par une interface graphique du logiciel facilite le processus de programmation, et une même tâche peut être affectée à plusieurs robots.

Cependant, la programmation hors-ligne a des inconvénients :

- Les comportements trop complexes dans la cellule du robot ne sont pas modélisés (ex. câble électrique trop court de la torche plasma), ce qui conduit à des erreurs entre l'environnement réel et l'environnement virtuel.
- La taille et la position de la pièce sont mesurées de façon approximative, ce qui augmente les erreurs de programmation.
- La stratégie de génération de la trajectoire n'est pas fournie, un logiciel extensionnel doit être développé pour pouvoir intervenir sur les caractéristiques essentielles.

7-4/Développement de l'interface graphique:

En informatique, une interface graphique (en anglais *GUI* pour *graphical user interface*) ou un environnement graphique est un dispositif de dialogue homme-machine, dans lequel les objets à manipuler sont dessinés sous forme de pictogrammes à l'écran, de sorte que l'utilisateur peut utiliser en imitant la manipulation physique de ces objets avec un dispositif de pointage, le plus souvent une souris.

Ce type d'interface a été créé par les ingénieurs du Xerox PARC à la fin des années 1970, pour remplacer les interfaces en ligne de commande, mis sur le marché avec le Star de Xerox et le Lisa d'Apple, puis popularisé par cette dernière firme avec l'ordinateur Macintosh, commercialisé en 1978 [22].

Les interfaces graphiques sont mises en œuvre par un ensemble de logiciels souvent inclus dans les systèmes d'exploitation (Windows) ou fournis avec eux par les distributions (Linux). Ils sont devenus vers le milieu des années 1990 le standard des appareils informatiques, notamment ordinateurs, tablettes, téléphones, récepteurs GPS et guichets automatiques de billetterie (dont bancaires).

La version 6 et 7 de MATLAB possède un outil nommé GUIDE (Graphical User Interface Développement Environnement) facilitant la génération des scripts MATLAB nécessaires à la création d'une interface utilisateur .[23]

Un GUI se présente comme une structure arborescente (ci-dessous) composée d'objets d'interface Figure 3.4

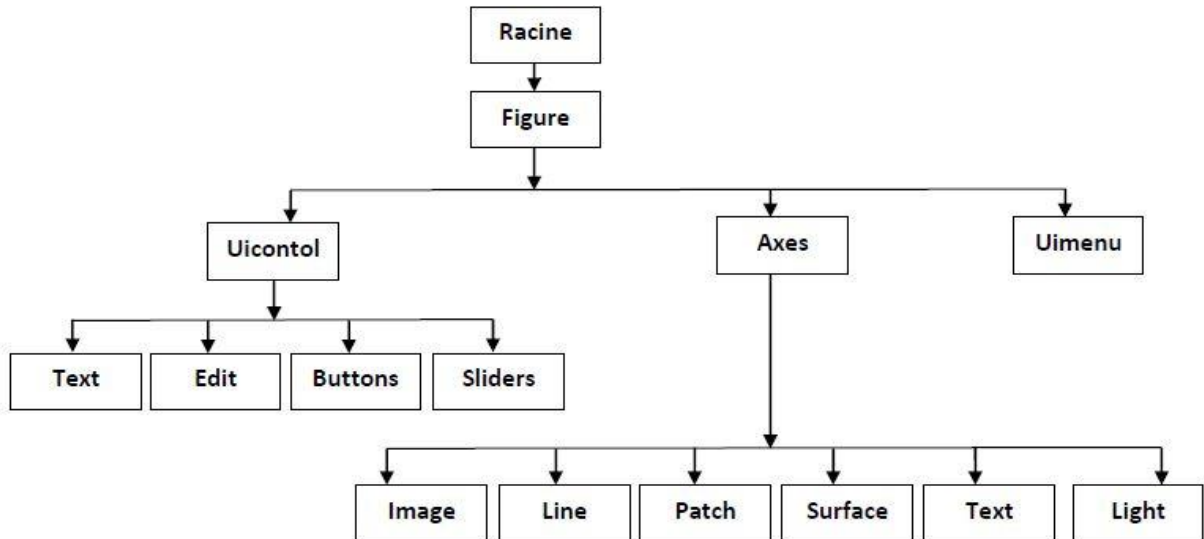


Figure 3. 4 structure arborescente composée d'objets d'interface .[23]

7-4-1/Création d'une interface graphique

Une interface graphique permet de contrôler une application interactivement avec la souris, plutôt que par lancement des commandes au clavier.

Une interface graphique comprend des menus, des boutons, des "ascenseurs", des cases à cocher, des listes de choix, des zones de texte. Exemple d'interface figure 3.5

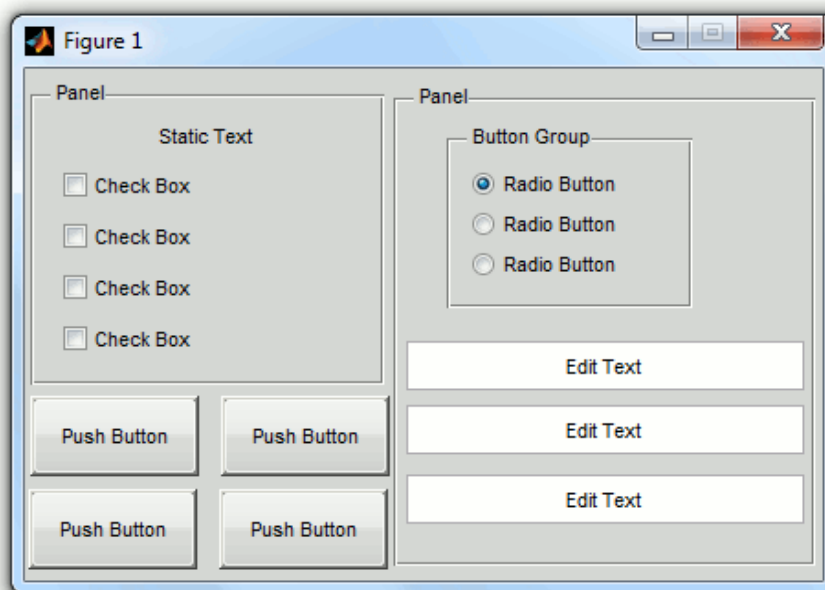


Figure 3. 5Exemple d'interface

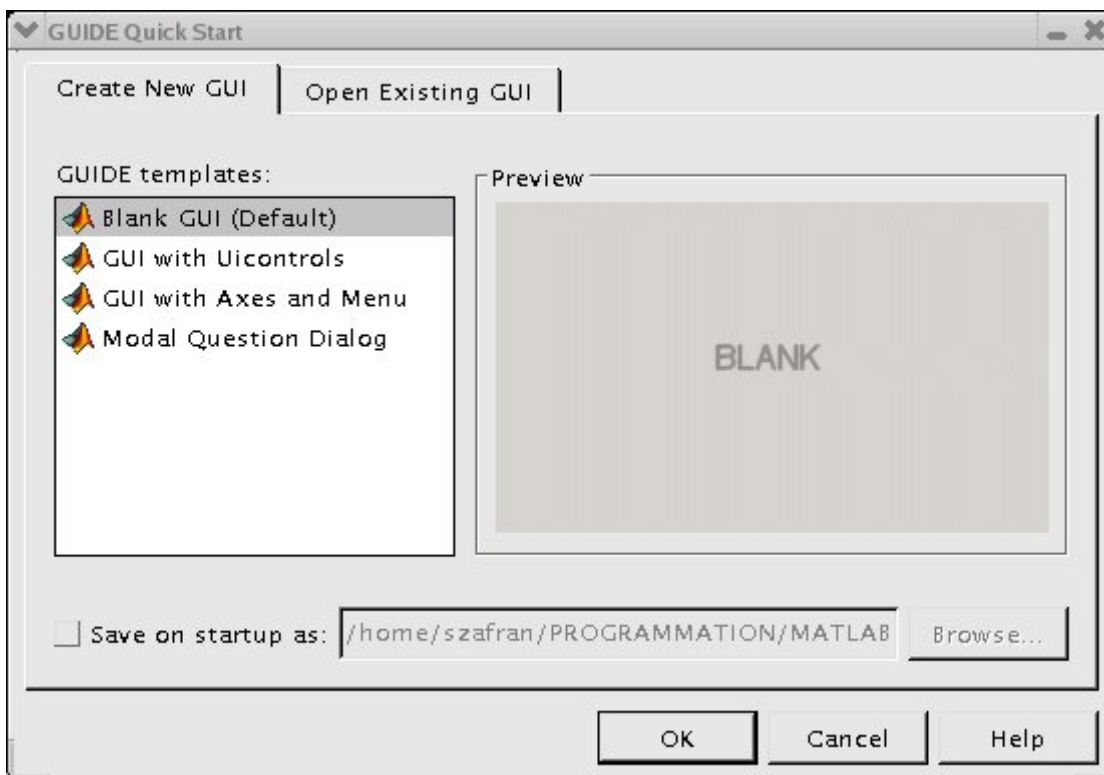
Matlab permet d'écrire assez simplement une interface graphique pour faire une application interactive utilisable par des utilisateurs non formés à Matlab.

Les notions principales d'une interface graphique sont :

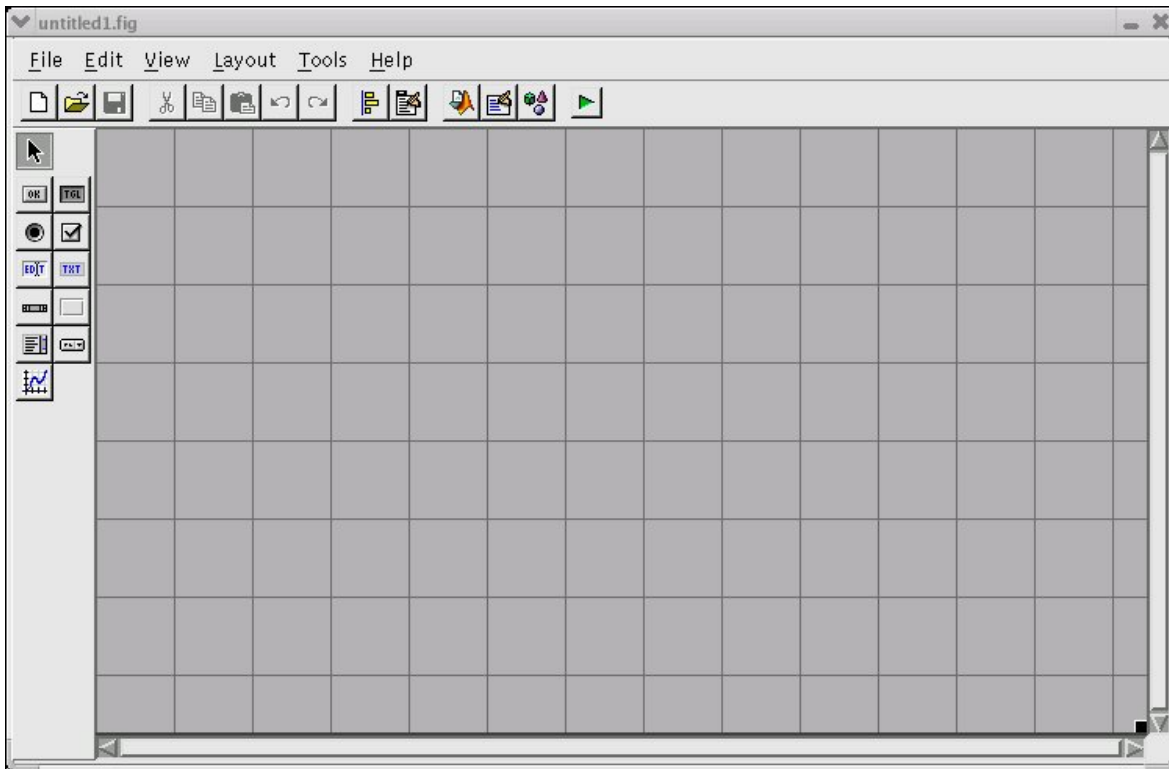
- les divers objets graphiques, auxquels sont attribués des noms symboliques; ces "poignées De main" ou "*handles*" permettent de les repérer dans l'interface; pour envisager par exemple une modification dynamique (grisé d'un bouton provisoirement non utilisable, changement du texte d'un bouton, modification d'une liste de choix...)
- les *propriétés* des objets (couleur, disposition, taille, variable associée)
- les fonctions exécutées par le clic souris sur les éléments ou "*callbacks*" (décrites en ligne de commande Matlab).

Les versions actuelles de Matlab permettent de construire ces interfaces directement avec la souris grâce au GUIDE (Graphical User Interface Développement Environnement). Cet outil est capable de construire des applications de très haut niveau. Cependant, son approche est très délicate pour un utilisateur peu expérimenté. L'approche classique par description textuelle de l'interface, moins performante mais compréhensible, permet de se familiariser avec les principales notions.[24]

pour crée interface graphique Tapez la commande guide, la fenêtre suivante s'ouvre :

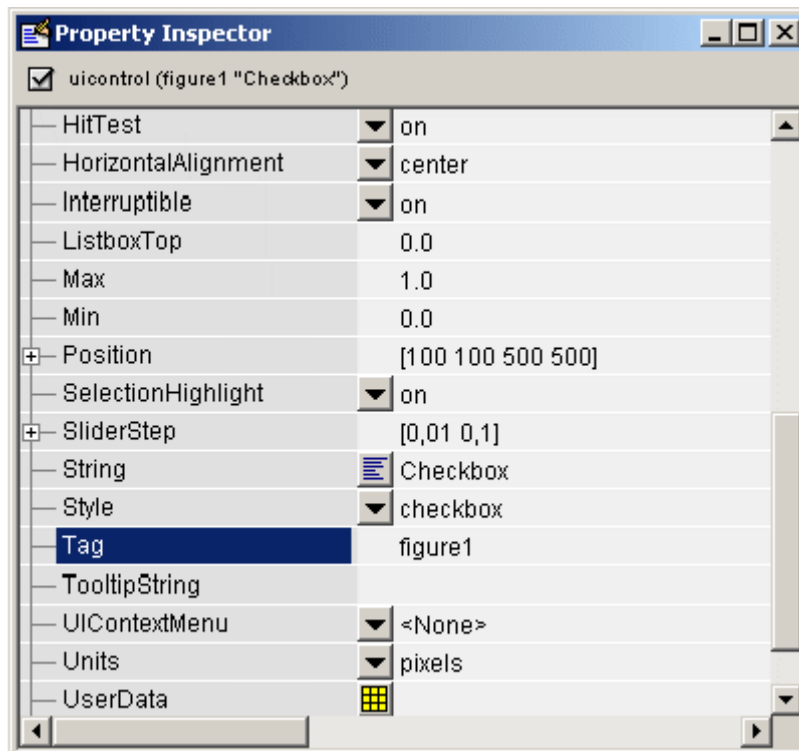


Cliquez sur OK pour créer une nouvelle interface. La fenêtre d'édition d'interface s'ouvre :



Cette fenêtre permet l'ajout/édition/suppression des différents éléments de l'interface. Au début, seul l'objet correspondant à la *figure* est présente.

Pour éditer les propriétés de la figure, double-cliquez dessus : la fenêtre d'édition des propriétés s'ouvre :



Par exemple, mettez la valeur pixels pour la propriété Unit puis les valeurs 100 100 500 500 pour les champs de la propriété Position, et figure1 pour la propriété Tag Ajoutez ensuite les autres éléments pour former l'interface équivalente à celle de l'exemple précédent :



:pour l'axe, il faut préciser les propriétés Units (pixels), Position, et Tag (axes1) .



:pour le menu popup, il faut préciser les propriétés Units (pixels), Position, String (les noms des différents items), et Tag (popup1) .



:pour le bouton on-off, il faut préciser les propriétés Unit (pixels),Position,Min(0),Max(1),Value(0),String(Pointille) ,et Tag(slider1).



: pour l'ascenseur, il faut préciser les propriétés Units (pixels), Position, Min(1) ,Max(5), Value(5), et Tag slider1)



: pour le label, il faut préciser les propriétés Units (pixels), Position, String ([-5.0, 5.0]), et Tag (text1) .



:: pour le bouton, il faut préciser les propriétés Units (pixels), Position, String (Quitter), et Tag(pushbutton1).

Sauvegardez l'interface ainsi créée (File save) en le nommant par exemple fichier.fig.

Le fichier fichier.fig est un fichier MATLAB contenant l'ensemble de l'interface graphique.

A ce fichier.fig est associé un fichier .m contenant le code MATLAB correspondant qu'il va falloir compléter pour ajouter les différentes actions associées à l'interface. On Mis en place tout les composants essentiels dans l'interface (slider, axes, edittext...etc.) puis choisir les

différentes propriétés de chaque composant (taille, couleur, position...etc.)

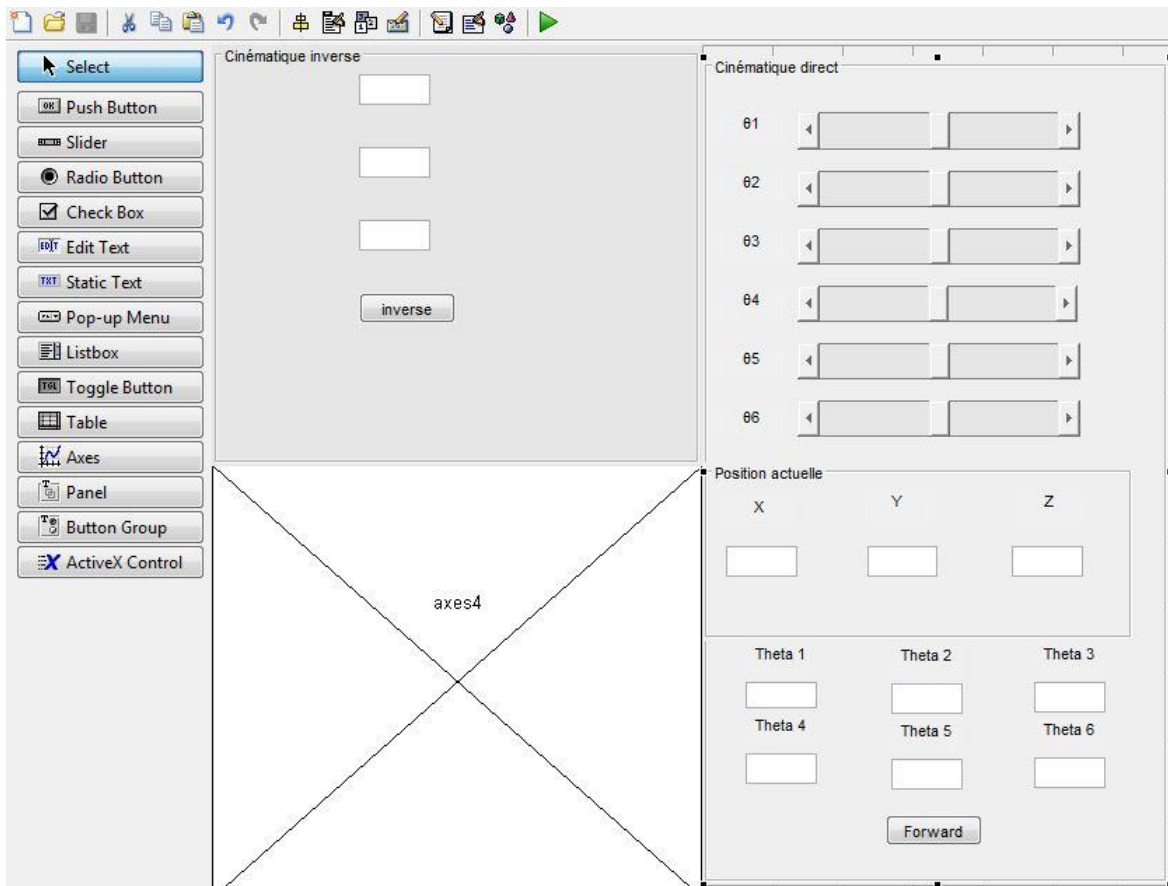


Figure 3. 6 mis en place les composants essentiels

7-4-2/Création du fichier : m correspondant

Une fois les différents composants de l'interface graphique mis en place, il faut les faire interagir, c'est-à-dire, programmer l'interface graphique figure 3.7. GUIDE permet la génération automatique d'un fichier :m correspondant à un fichier :fig. Celui-ci contient notamment une fonction s'exécutant au lancement du programme et les appels aux différents composants.

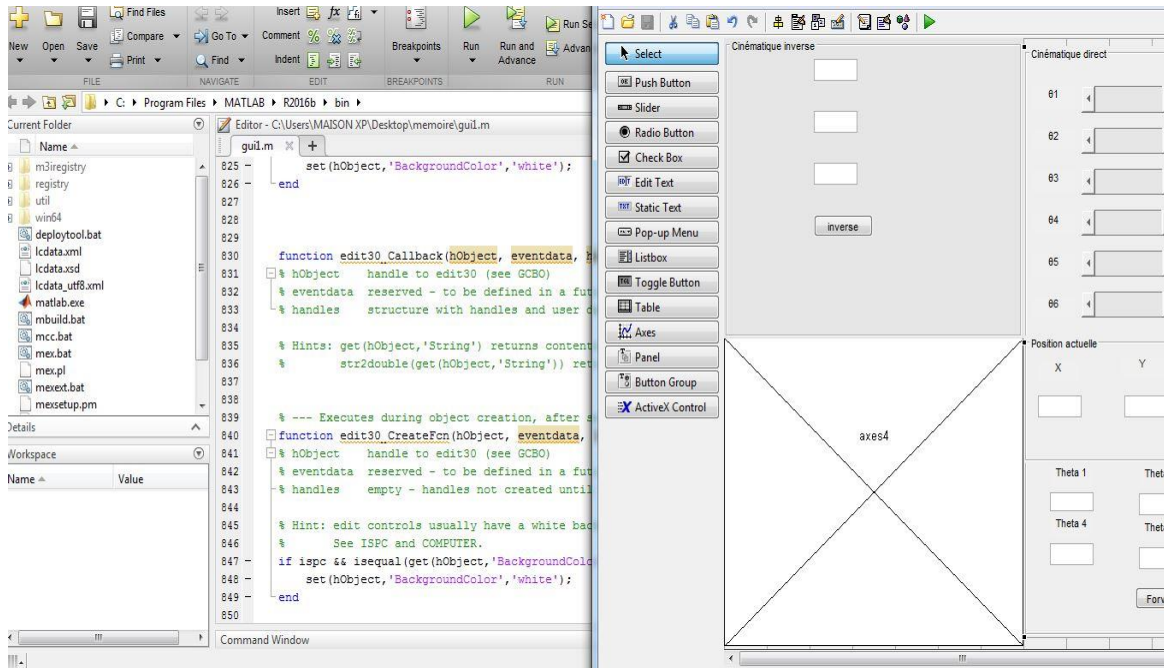


Figure 3. 7 Programmation de l'interface graphique

7-5/Création notre modèle de robot IRB 2600 sur solide Works:

La première étape de modélisation consiste la création d'un model CAO de notre robot sur le logiciel Solid-Works. Chaque axe du robot est modélisé séparément puis assemblé avec les autres axes pour avoir le model final (Figure 3.8).

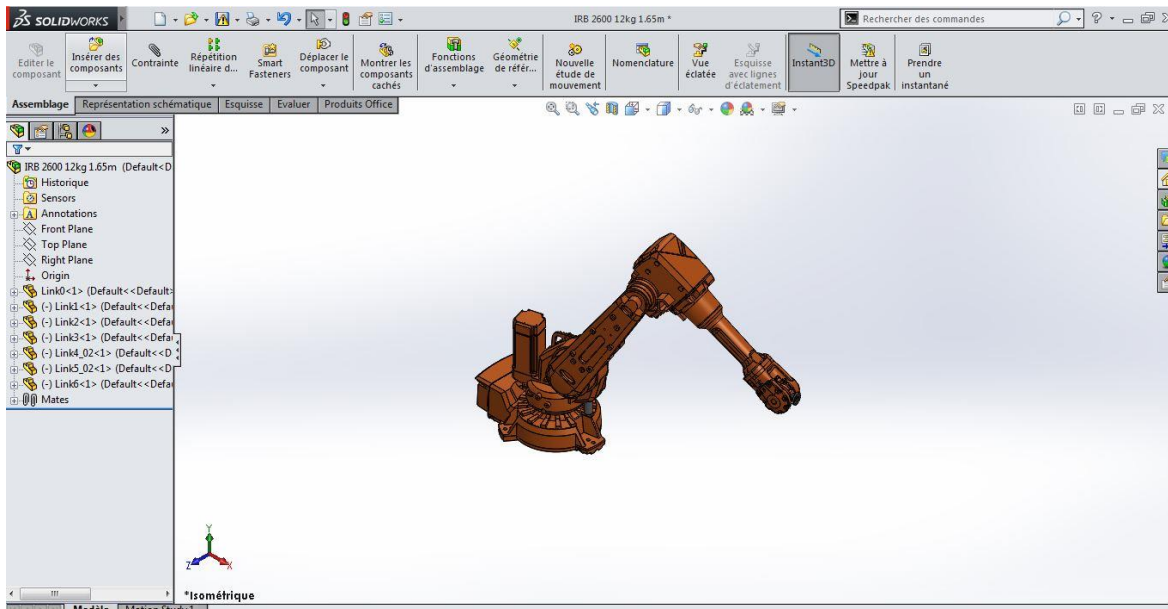


Figure 3. 8 Modèle de robot IRB 2600 sur solide Works

Puis on utilise SimscapeMultibody Link pour exporter ce modèle sous forme première génération pour l'intégrer sur matlab.

Simscape, un langage multi-disciplinaire, est une extension de Matlab/Simulink (Simulink est un environnement graphique de simulation et de modélisation de systèmes dynamiques. Il est intégré à Matlab, fournissant ainsi un accès immédiat à ses nombreux outils de développement algorithmique, de visualisation et d'analyse de données) pour modéliser les systèmes électroniques, mécaniques, hydrauliques et thermiques.

Pour notre cas, on importe notre modèle sur matlab (simulink) avec le code (`mech_import('IRB 2600 12kg 1.65m.xml')`) Figure 3.9.

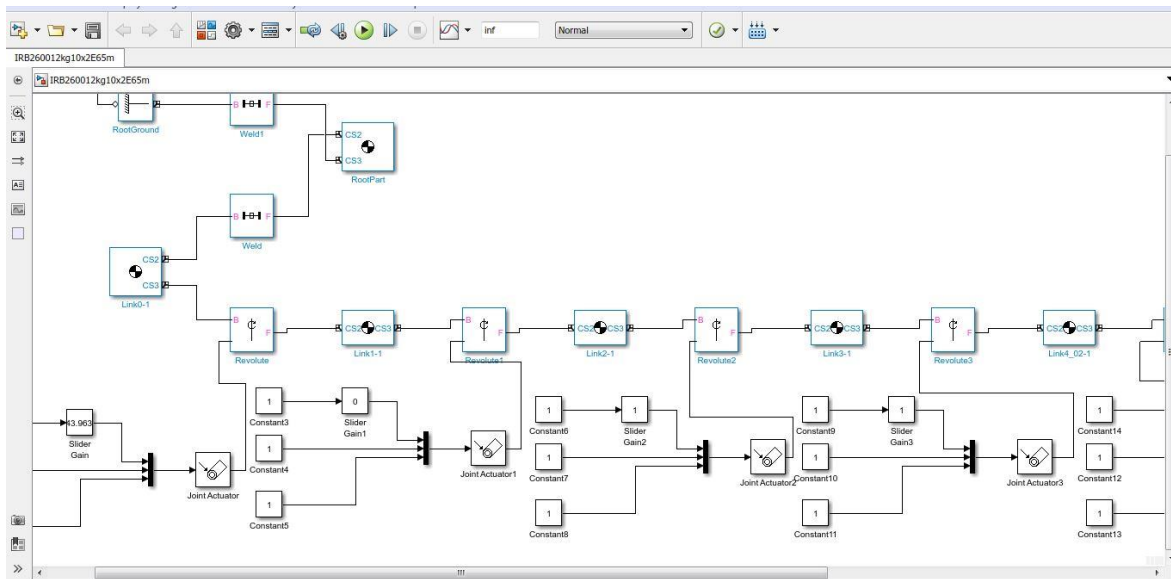
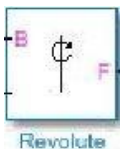


Figure 3. 9 modèle robot IRB 2600 sur simulink



Ce bloc représente une articulation avec un degré de liberté de rotation. Une primitive révolutionnaire fournit le degré de liberté de rotation. Les origines des bases et des suiveurs restent coïncidentes pendant la simulation.



Le bloc Slider Gain effectue un gain scalaire que vous pouvez modifier pendant la simulation. Modifiez le gain en utilisant le paramètre du curseur.



Un joint entre deux corps représente des degrés de liberté relatifs (DoF) entre les corps. Le bloc Joint Actuator actionne un bloc Joint connecté entre deux corps avec l'un de ces signaux:

1. Une force généralisée:
 - Force pour le mouvement de translation le long d'une primitive conjointe prismatique

- Couple pour le mouvement de rotation autour d'une primitive conjointe rotative

2. Une motion

- Mouvement translationnel pour une primitive conjointe prismatique, en termes de position linéaire, de vitesse et d'accélération.
- Mouvement de rotation pour une primitive conjointe rotative, en termes de position angulaire, de vitesse et d'accélération



Le bloc Constant génère une valeur constante réelle ou complexe. Le bloc génère une sortie scalaire, vectorielle ou matricielle



Le bloc Weld représente un joint sans degrés de liberté (DoF). Les deux corps connectés de chaque côté du bloc de soudure sont étroitement liés l'un à l'autre, sans mouvement relatif possible

Après la création du model Simulink, on crée des slider pour chaque articulation, pour contrôler les mouvements du robot Figure 3.10.

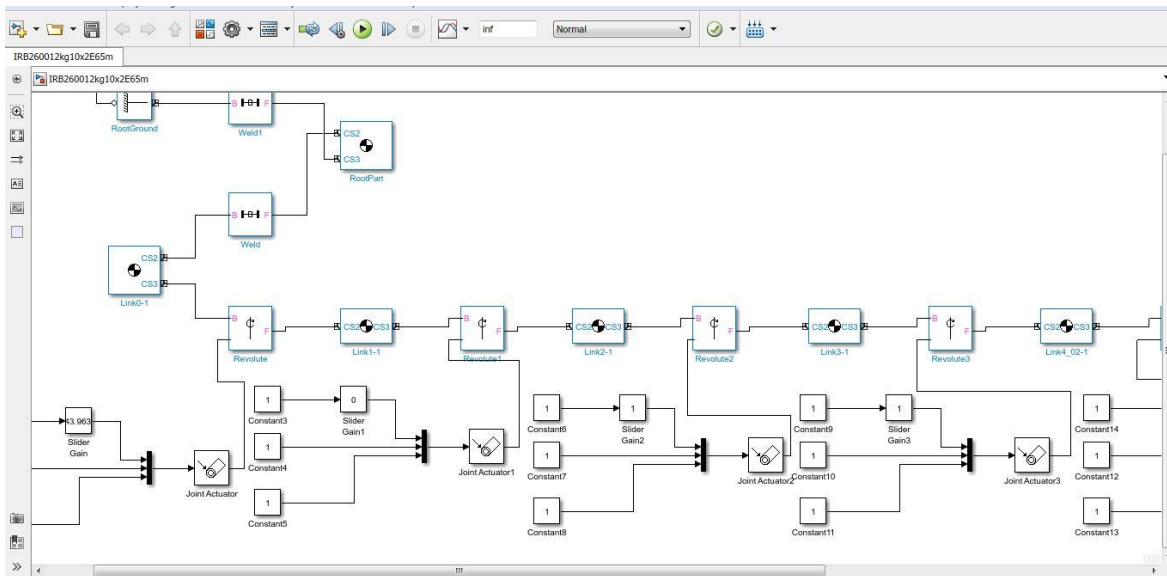


Figure 3. 10 Modèle Robot IRB 2600 avec les slider

Pour faire le lien entre l'interface graphique présentée à la Figure 2.6 et le model Simulink montré à la Figure 3.10, des algorithmes de commande ont été créé on utilisant la fonction callback du Matlab (Figure. 3.11).

```

1 if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
2 set(hObject,'BackgroundColor',[.9 .9 .9]);
3 modelName = 'IRB 2600 12kg 1.65m';

4 t1=get(handles.Theta1,'value');
5 set(handles,edit1,'string',num2str(t1));

6 t2=get(handles.Theta2,'value');
7 set(handles,edit2,'string',num2str(t2));

8 t3=get(handles.Theta3,'value');
9 set(handles,edit3,'string',num2str(t3));

10 t4=get(handles.Theta4,'value');
11 set(handles,edit4,'string',num2str(t4));

12 t5=get(handles.Theta5,'value');
13 set(handles,edit5,'string',num2str(t5));

14 t6=get(handles.Theta6,'value');
15 set(handles,edit6,'string',num2str(t6));

16 set_param([modelName '/Slider gain1'],'Gain',num2str(t1))
17 set_param([modelName '/Slider gain1'],'Gain',num2str(t2))
18 set_param([modelName '/Slider gain2'],'Gain',num2str(t3))
19 set_param([modelName '/Slider gain3'],'Gain',num2str(t4))

20 set_param([modelName '/Slider gain4'],'Gain',num2str(t5))
21 set_param([modelName '/Slider gain5'],'Gain',num2str(t6))

22 T1=[cos(t1) -sin(t1) 0 0;sin(t1) cos(t1) 0 0;0 0 1 450;0 0 0 1];
23 T2=[sin(t2) 0 cos(t2) 150*sin(t2); -cos(t2) 0 sin(t2) -150*cos(t2) ;0 -1 0 0; 0 0 0 1];
24 T3=[cos(t3) -sin(t3) 0 900*cos(t3); sin(t3) cos(t3) 0 900*sin(t3) ;0 0 1 0; 0 0 0 1];
25 T4=[cos(t4) 0 -sin(t4) 115*cos(t4);sin(t4) 0 cos(t4) 115*sin(t4); 0 -1 0 795; 0 0 0 1];
26 T5=[cos(t5) 0 sin(t5) 0; sin(t5) 0 -cos(t5) 0 ;0 1 0 0; 0 0 0 1];
27 T6=[-cos(t6) 0 -sin(t6) 0 ; -sin(t6) 0 cos(t6) 0; 0 -1 0 0; 0 0 0 1];
28 T=T1*T2*T3*T4*T5*T6;
29 Px=T(1,4);
30 Py=T(2,4);
31 Pz=T(3,4);
32 set(handles.edit7,'string',num2str(Px));
33 set(handles.edit8,'string',num2str(Py));
34 set(handles.edit9,'string',num2str(Pz));
35 end

36 % --- Executes on slider movement.
37 function Theta_2_Callback(hObject, eventdata, handles)
38 % hObject handle to Theta_2 (see GCBO)
39 % eventdata reserved - to be defined in a future version of MATLAB
40 % handles structure with handles and user data (see GUIDATA)

41 % Hints: get(hObject,'Value') returns position of slider

```

Figure 3. 11 algorithme de matlab pour les slider

Chapitre IV

Réalisation et validation du simulateur

8/Réalisation et validation du simulateur

Comme présenté à la Figure 4.1, notre interface graphique comporte quatre fonctionnalités,

- Exécution de la cinématique directe
- Exécution de la cinématique inverse
- Commande axe par axe
- Visualisation graphique

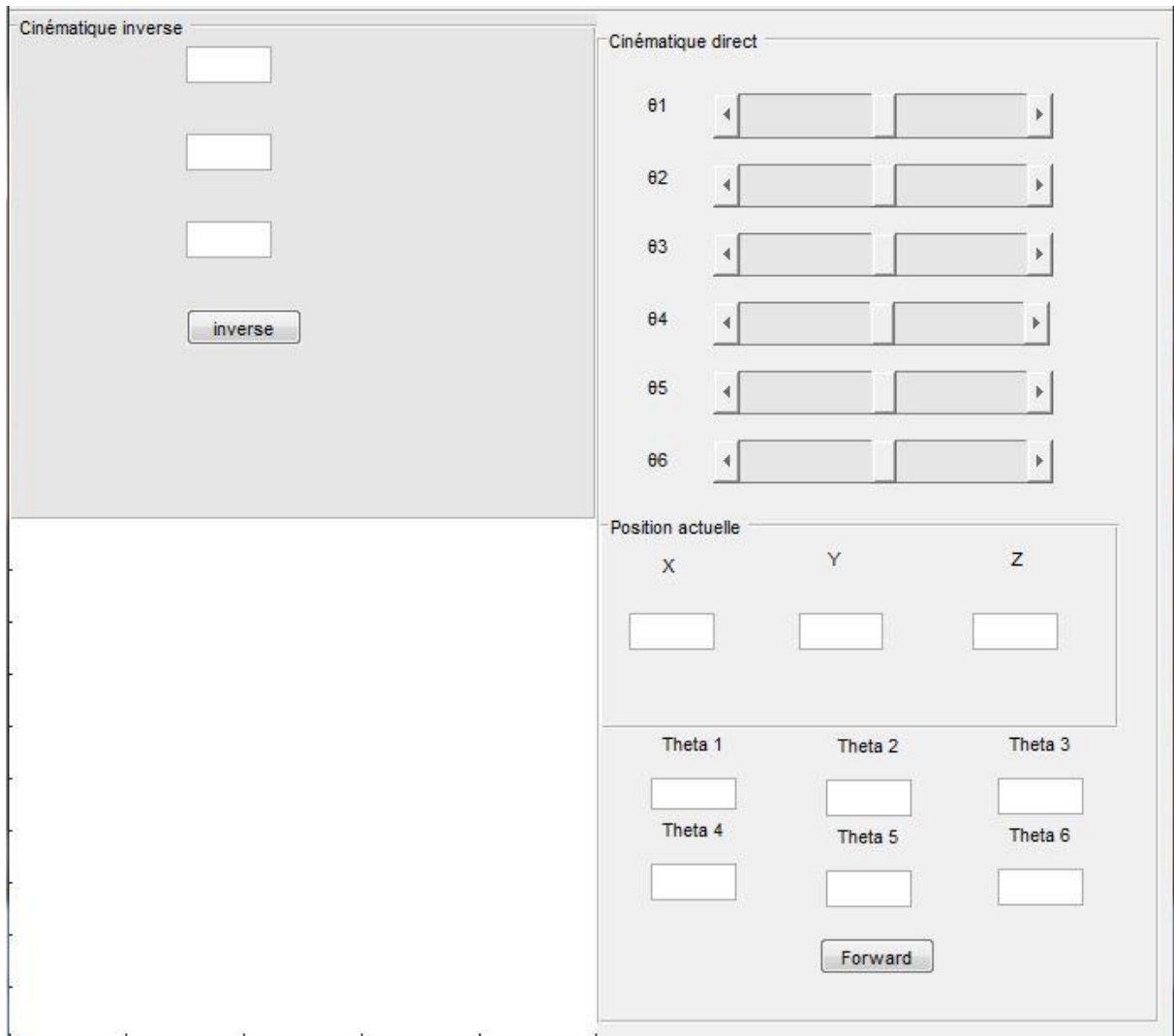


Figure 4. 1 Format initial de notre simulateur

8-1/Exécution de la cinématique directe

L'exécution de la cinématique direct ce fait par l'introduction des valeurs des angles θ_1 , $\theta_2 \dots \theta_6$ dans notre interface puis cliquer sur le bouton forward. En cliquant sur forward, Matlab fait appeler et exécuter notre programme de la cinématique direct déjà inclus précédemment dans la librairie de simulateur. Après l'exécution du programme, les valeurs

des positions actuelles de l'organe terminal (x , y et z) fournis par la fonction cinématique directe vont être affichées sur l'interface graphique.

Comme exemple, faisons entrer les valeurs $\theta_1 = 90$, $\theta_2 = 0$, $\theta_3 = 45$, $\theta_4 = 35$, $\theta_5 = 70$ et $\theta_6 = 90$ dans l'interface graphique (Figure 4.1). Lorsque l'on clique sur l'icône forward les valeurs de position actuelles $X = 204$, $Y = 197$ et $Z = 87$ sont affichées sur l'interface graphique.

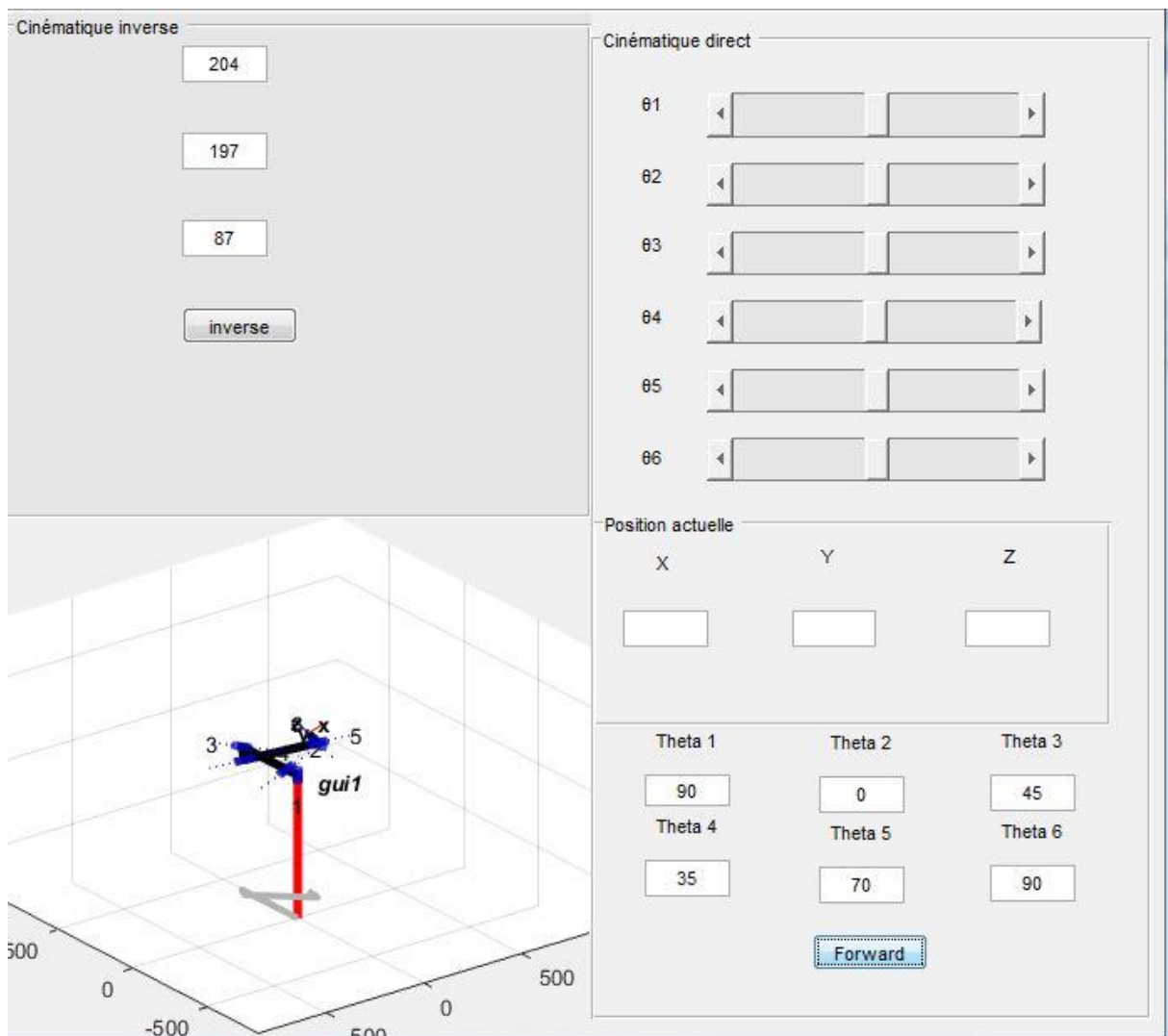


Figure 4. 2 Résultat de la cinématique direct

8-2/ Exécution de la cinématique inverse

Donnant la position à atteindre (X , Y , Z), dans les icônes désirés de notre interface puis cliquant sur le bouton inverse. La clique sur ce bouton fait appeler à la fonction callback. Généralement un callback est une fonction qui est appelée lorsqu'un événement particulier se produit dans une interface graphique. La propriété *CallBack*, permet de définir la fonction, qui sera appelée lorsqu'on clique sur un bouton. Dans cet exemple, la fonction callback est

rattachée au programme de la fonction cinématique inverse. L'exécution de ce programme donne les valeurs des angles $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5$ et θ_6 .

Par exemple si on donne $X=129, Y=194$ et $Z=166$, la clique sur le bouton inverse conduit à l'exécution du programme de la cinématique inverse. Les valeurs calculées des différents angles sont envoyées, reçues et affichées dans l'interface graphique comme montré à la Figure 4.3. Par conséquence le robot déplace à la position désirée.

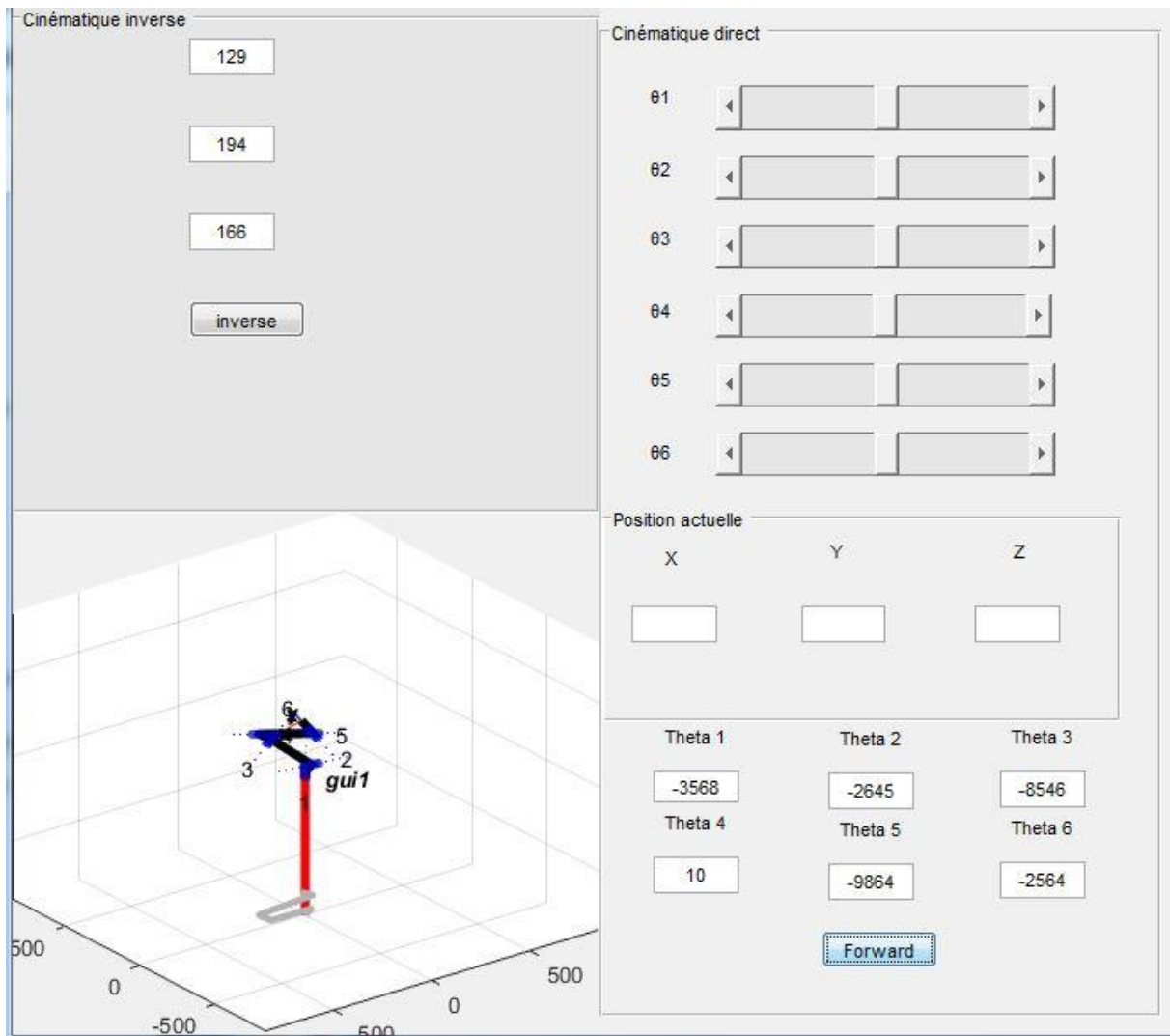


Figure 4. 3 Résultat de la cinématique inverse

8-3/.Commande axe par axe

La commande axe par axe est assuré par des sliders via l'outil Simulink. Chaque slider est associé à un axe du robot. Pour un robot 6 axes, il faut avoir 6 sliders. Ces sliders ne peuvent avoir une valeur en dehors du min et du max.

Pour assurer cette commande, le modèle CAO du robot assemblé dans Solidworks est exporté vers Matlab via une branche(plug-in) qui interconnecte les deux plates-formes et qui, à son

tour, est dirigé vers l'outil Simulink. La conversion peut être faite en utilisant SimMechanics et Simscape Multibody.

Lors de l'exécution de la conversion, un diagramme fonctionnel est obtenu (Figure. 4.6) et l'axe de la simulation est préalablement configuré. En outre, une entrée de commande avec une valeur numérique est conçue pour répartir le déplacement d'axe et de l'actionneur dans la simulation.

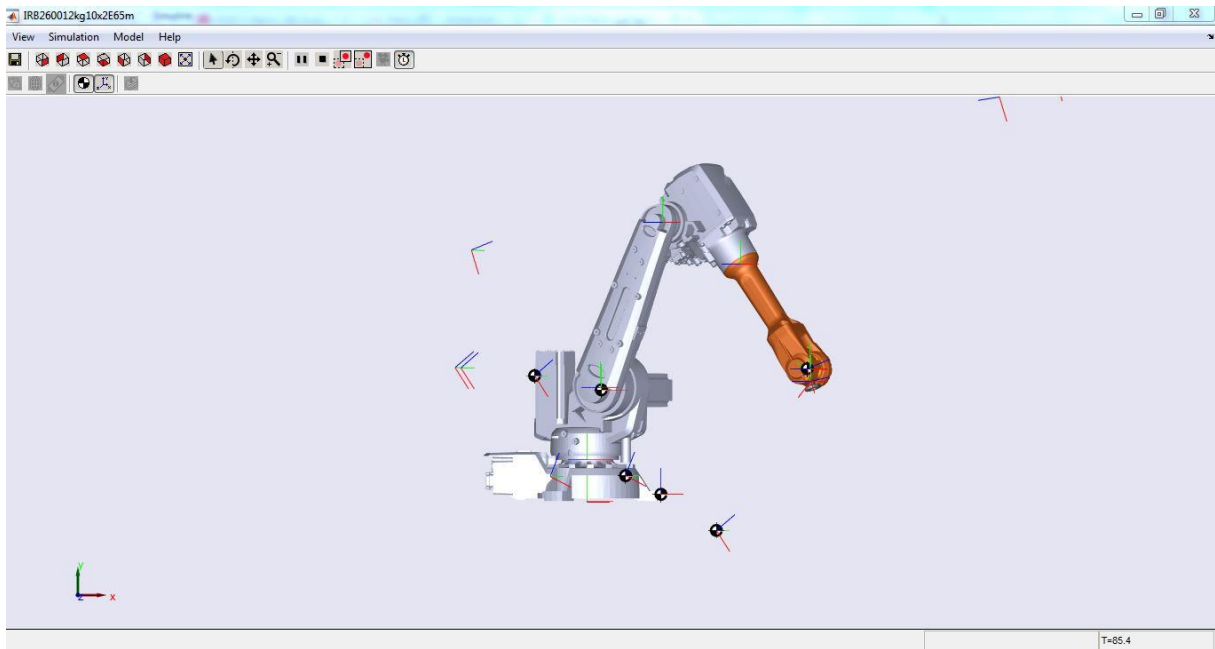


Figure 4. 4 Notre modèle dans simscape (simulink)

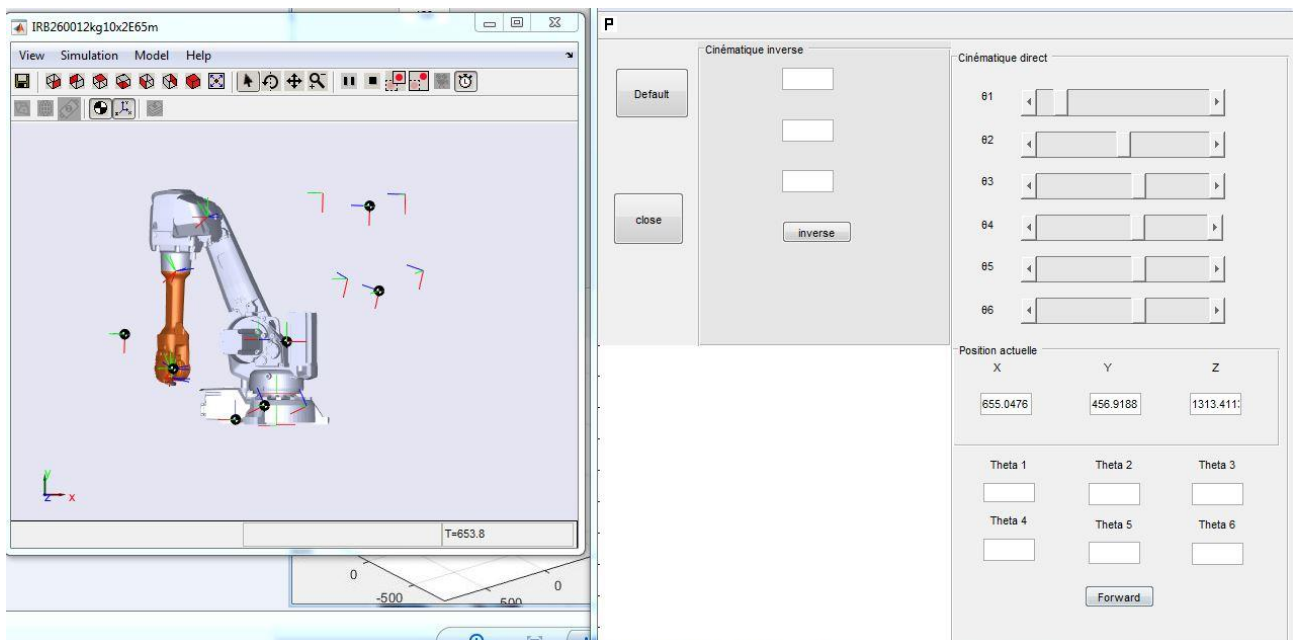


Figure 4. 5 Résultat de cinématique direct avec les slider est la déclaration de la position actuelle

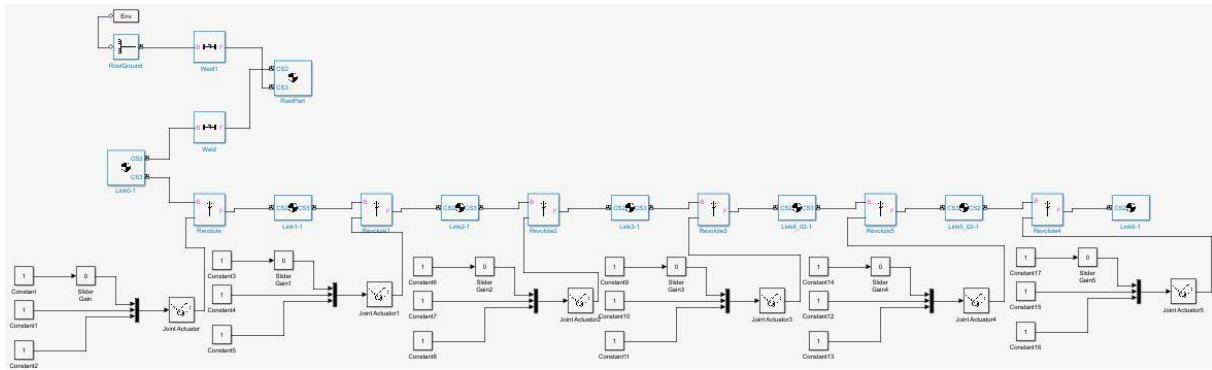


Figure 4. 6 Notre Modèle Robot IRB 2600 dans simulinkk avec les slider

8-4. Visualisation graphique

Comme présenté précédemment, le logiciel produit comprendra une partie visualisation graphique importante (visualisation des exécutions) et une partie de manipulation par slider. Sur cette interface (Figure. 4.5), le bouton « Default » permet de remettre à zéro les valeurs de la cinématique directe et inverse. Le bouton « close » permet de rafraichir la représentation graphique du robot.

CONCLUSION

Conclusion

Pour contrer les désavantages de la programmation par enseignement, il existe une alternative appelée méthode de programmation hors ligne. Celle-ci consiste à programmer le robot et simuler ses mouvements en utilisant des logiciels de simulation dédiés. Le robot n'est pas physiquement utilisé lors de la programmation, ne causant aucun arrêt de production. En résumé, pour profiter des avantages de la programmation hors ligne, il est nécessaire que le robot soit doté d'une bonne précision absolue.

Dans ce travail, les logiciels SolidWorks et MATLAB / Simulink ont été utilisés pour développer un simulateur 3D et simuler le mouvement d'un robot à six axes.

Les résultats obtenus comportent la mise en œuvre d'un environnement virtuel d'un robot sériel à six axes à travers un assemblage tridimensionnel complet de la structure divisée en sous-ensembles, un modèle CAO pour chaque articulation, un bloc diagramme avec les articulations et les pièces correspondantes et une interface de contrôle graphique qui permet la simulation du comportement des robots et le temps de réponse, afin de fournir plus de facilité à l'utilisateur dans la manipulation du modèle virtuel.



Références bibliographique

Référence

- [1] J.P. Merlet, Paralell Robots, Editions Hermes, INRIASophia-Antipolis, Francia, 1997.
- [2] R. Aracil, R. Saltaren, J.M. Sabater, O. Reinoso. Robots paralelos: Máquinas con un pasado para unarobótica del futuro”, RevistaIberoamericana de Automática e InformáticaIndustrial, vol. 3, pp. 16-20, Enero, 2006
- [3] J. Olaya, N. Pintor, “Diseño de un Ambiente Virtual para unaPlataformaRobóticaParalela”, Trabajotesis final de pregrado, Dep. Ingeniería y CienciasBásicas, ProgramaIngenieríaElectrónica, Universidad Central de Colombia, Bogotá, 2016
- [4] COMPARISON OF THE CHARACTERISTICS BETWEEN SERIAL AND PARALLEL ROBOTS, Zoran PANDILOV, 2. Vladimir DUKOVSKI
- [5] ISO, Robots et composants robotiques { Vocabulaire, ISO 8373, 2012
- [6] TSAI L. W., 1999, Robot Analysis: The Mechanics of Serial and ParallelManipulators, New York: John Wiley& Sons, Inc.,1999
- [7] Robots sériels : géométries, applications R.Clavel, 2009
- [8] ISO/TR 8373 : 1988, Robots manipulateurs industriels
- [9] Les robots parallèles, Merlet 1997
- [10] Clavel, R. (1988). Delta, a fast robot withparallelgeometry. Proc. of the 18th International Symposium on Industrial Robots, pages 91–100, Lausanne, Switzerland.
- [11] Gough, V. E. (1956-1957). Contribution to discussion of papers on research in automotivestability, control and tyre performance. Proc. Auto Div., Institute of mechanical engineering
- [12] D. Stewart, “A platformwith six degrees of freedom”, Proceedings of the IMechE, Vol. 180, Pt. 1, No. 15, pages 371–385, 1965.
- [13] Clavel, R. (1989). Une nouvelle structure de manipulateur parall`ele pour la robotique l`eg`ere. Automatique Productique Informatique Industrielle, 23: 501–519.
- [14] Clavel, R. (1994). Robots parall`eles. Techniques de l’Ing`enieur, S(7710).
- [15] Pierrot F., Dauchez P., Fournier A. "Hexa: a fast six-doffullyparallel robot", in proc. of IEEE ICAR: International Conferenceon Advanced Robotics, Pise, Italy, pp. 1159-1163, June 19-22, 1991

[16] INITIATION A LA MODELISATION ET AUX PERFORMANCES DES ROBOTS,
H.DEMOUVEAU

[17] Wenger and Chablat, 1997

[18] W. Khalil et E. Dombre. Modélisation, identification et commande des robots. Hermès, 1999..

[19] Etude du comportement mécanique des machines outils à structure parallèle en Usinage Grande Vitesse, Thomas Bonnemain, 1982

[20] DJOKHRAB, Ala Eddine (2015) Planification et Optimisation de Trajectoire d'un Robot Manipulateur à 6 D. D. L. par des Techniques Neuro-Floues. Masters thesis, Université Mohamed Khider - Biskra

[21] GPA546, ilian Bonev et Yanick Noieux

[22] Ronald M. Baecker, Readings in Human-Computer Interaction : Toward the Year 2000, Morgan Kaufmann, 1995

[23] <http://hmf.enseeiht.fr/travaux/CD0405/optmfn/mci/reports/2004/b/vpwm/Interfgraphique/interface.htm>

[24] Introduction Matlab - Compléments GUI Création d'une interface graphique

