

Remerciements

*Je remercie **ALLAH** de nous avoir donné la
volonté et le courage qui nous ont permis de
réaliser ce travail.*

*J'aimerais spécialement remercier mon promoteur
Monsieur KAZED BOUALEM, pour sa
disponibilité et pour son grande aide tout au long
de ce travail.*

Ainsi qu'au président et les membres de jury

*Je tiens aussi à présenter mes remerciements
aussi à toutes les personnes qui ont participées de
près ou de loin à l'élaboration du présent mémoire.*

الجمهورية الجزائرية الديمقراطية الشعبية
République Algérienne démocratique et populaire

وزارة التعليم العالي والبحث العلمي
Ministère de l'enseignement supérieur et de la recherche scientifique

جامعة سعد دحلب البليدة
Université SAAD DAHLAB de BLIDA

كلية التكنولوجيا
Faculté de Technologie

قسم الإلكترونيك
Département d'Électronique



Mémoire de Projet de Fin d'Études

présenté par

KAOULA GHANIA

Pour l'obtention du diplôme de Master II en Électronique option SVR

Thème

Réalisation d'un robot Delta Parallèle à 3 degrés de liberté

Proposé par : **Mr. KAZED BOUALEM**

Année Universitaire 2014-2015

Résumé :

Le but de ce projet consiste à réaliser un robot Delta parallèle à 3 degrés de liberté. Pour ce faire nous allons d'abord passer par la réalisation de la structure de notre robot à l'aide de lego Mindstorms. En deuxième étape nous allons aborder la partie théorique et programmation, nous avons étudié les modèles géométriques direct et inverse pour ensuite les implémenter sous Matlab pour les tester et sous NXC pour manipuler notre robot.

Mots clés :

Robot Delta, robot parallèle, Mindstroms lego.

Abstract:

The aim of this project designs a Delta type robot arm. To achieve this we are going to study both the direct and invers mathematical models. This will be verified in simulation and implemented in a lego Mindstroms brick, which will control the 3 DC motors to drive the end effector to the desired position.

Key word:

Delta robot, parallel manipulator, lego Mindstroms.

المخلص:

الهدف من هذا المشروع ي تمثل في تصميم روبوت ذو تماوازية ذات 3 درجات حرية. و تحقق يقا لذلك مرزنا بمرحلة صناعة الهيكل بواسطة اسد تعمال مهندس تروالديغوم. في المرحلة الثانية ت طرق نال لجزء النظري الخاص بالبرمجة حيث اسد تعمالنا ال برنامج الالكتروني NXC و ماطلاب.

الكلمات الاساسية :

روبوت ذو تماوازية ، الديغومهندس تروم .

Sommaire

Introduction générale.....	1
Chapitre 1 : Etat d'art des robots	
1.1 Introduction.....	3
1.2 Definition d'un robot	3
1.3 Classification des robots.....	3
1.4 Type de robot industriel existants	4
1.5 Prototypes d'un robot industriel	23
1.5.1. degré de liberté d'un robot	23
1.5.2. Exemple d'un prototype à 2 DDL.....	23
1.5.3. Robot à 3 DDL.....	24
1.6. Avantages et inconvénients des Robots Delta	25
1.7 Conclusion	26
Chapitre 2 : Modélisation géométrique de la structure parallèle	
2.1 Introduction.....	27
2.2 Modélisation géométrique d'un robot parallèle.....	27
2.2.1 Modélisation géométrique directe et inverse d'un robot	27
2.3. Etude théorique	28
2.3.1 Etude géométrique inverse	28
2.3.2 Etude géométrique directe	33
2.4 Implémentation du modèle géométrique sous Matlab	38
2.5 Conclusion	42
Chapitre 3 : Description du robot Delta et étude pratique	
3.1 Introduction.....	43
3.2 Conception du robot Delta	43
3.3 Architecture du robot.....	44
3.4 Conclusion	63

Chapitre 4 : Tests et résultats pratiques

4.1 Introduction :	64
4.2 Réalisation de la structure du robot :	64
4.3 Programmation:.....	64
a) Tests et simulation sous Matlab	66
a-1) Test du mouvement du robot sous Matlab	72
4.4 Conclusion	74
Conclusion général.....	75

Le travail présenté dans ce mémoire a été consacré à la réalisation d'un robot parallèle Delta à trois degrés de liberté. En ce qui concerne la structure de notre robot nous avons utilisés des lego Mindstorms avec tout le kitte qui est fourni. En second lieu nous avons utilisés le modèle géométrique inverse et direct appliqués généralement pour des robots parallèles et ce pour la détermination des paramètres nécessaires pour la modélisation du mouvement de notre robot.

Pour ce qui est de but de notre projet nous pouvons conclure qu'il a été atteint d'une manière assez satisfaisante. Et la satisfaction est atteinte sur les deux plans partie théorique (la programmation) et partie pratique pour ce qui est de la réalisation du robot.

Pour la réalisation de la structure de notre robot et concernant la solution adaptée qui est le lego, ces modules fournis nous ont clairement facilité la tâche pour la réalisation mécanique du châssis de notre robot. Néanmoins, comme toutes technologie, cette dernière à des inconvénients, dans notre cas la stabilité de la structure n'était pas atteinte à 100% notamment au niveau des coudes et articulations ce qui a légèrement influencé sur le mouvement de notre robot.

Pour la partie programmation, nous avons en premier lieu établi les modèles géométriques inverse et directe afin d'établir les équations nécessaires pour l'obtention des paramètres dont nous avons besoin pour la réalisation du mouvement de notre robot. Nous avons implémentés en premier ces équations sous le langage Matlab pour effectuer le calcul des angles de rotations des articulations dans le cas de la géométrie inverse, et nous avons aussi implémenté la géométrie directe afin de vérifier les résultats obtenus. Les résultats obtenus suite à la simulation du mouvement de notre robot sous Matlab étaient plus que satisfaisantes. Ce qui nous a permis de vérifier la validité de notre étude et de pouvoir effectuer la programmation sous NXC directement pour la partie pratique. Cependant, il reste le volet mécanique de notre robot qui est à améliorer.

A cet effet et pour atteindre une plus haute satisfaction nous proposons comme extension de ce projet la, qui est la réalisation d'une structure plus rigide que la nôtre pour pouvoir réaliser des applications au projet plus variées et plus complexes et là satisfaction sera totale sur les deux volet, volet programmation et volet réalisation.

1.1. Introduction :

Ce chapitre permet de présenter les types de robots existants, de les classer par types d'architectures et par types d'applications.

Aussi de faire une comparaison entre les deux types de robots sériels et parallèles afin de comprendre les lacunes et les avantages de chaque type.

1.2. Définitions d'un robot :

Le terme « **robot** » vient du mot slave " **robota** " qui signifie : travail forcé ou corvée, cette dénomination a été utilisée par l'écrivain tchèque Karel Capek dans les années 20 pour désigner des machines automatiques appelées à remplacer l'homme dans des tâches les plus pénibles.

Le Petit Larousse définit un robot comme étant un appareil automatique capable de manipuler des objets, ou d'exécuter des opérations selon un programme fixe ou modifiable.

Pour mériter le nom de robot, un système doit posséder une certaine flexibilité, caractérisée par les propriétés suivantes :

- **La versatilité :** Un robot doit avoir la capacité de pouvoir exécuter une variété de tâches, ou la même tâche de différente manière.
- **L'auto - adaptabilité :** Un robot doit pouvoir s'adapter à un environnement changeant au cours de l'exécution de ses tâches.

1.3. Classification des robots :

Les robots peuvent être classés :

Selon le type de tâches, ou il existe 2 classifications dans ce cas:

- **Robots de substitution :** pour des robots qui ont des tâches répétitives et précisément définies (donc programmables).
- **Robots de coopération** ou « **Cobot** » : pour des robots qui ont des mouvements pilotés directement par l'homme, le robot apportant l'assistance à l'effort, à la précision du geste.

Et selon le type d'architecture :

- **Robots à structure sérielle :** SCARA, cylindriques, sphériques, cartésiens, anthropomorphes...
Exemples d'applications : Packaging/Conditionnement, Peinture/Pulvérisation, Soudage avec processus intégré, Vissage, Palettisation, Positionnement, Chargement, Polissage...

- **Robots à structure parallèle** : ils peuvent avoir jusqu'à 6 degrés de liberté : par exemple Robots Delta Exemples d'applications : Assemblage, Manutention, Encaissage, Prise et dépose à la volée « Pick and place »

Dans le cadre de ce projet nous allons s'intéresser au robot parallèle. Plus précisément la conception d'un robot parallèle à deux degré de liberté dédié à l'application de prise et de dépose d'objet et ce de type industriel.

1.4. Types de robots industriels existants :

a) Définition :

Un robot industriel est un manipulateur automatique asservi en position, polyvalent, reprogrammable, capable de positionner et d'orienter des matériaux, des pièces, des outils ou des dispositifs spécialisés au cours de mouvements variables et programmés pour l'exécution de tâches variées.

Un robot est composé de plusieurs segments articulés permettant de piloter un organe dit terminal ou effecteur. L'organe terminal ou effecteur est le dispositif avec lequel les objets peuvent être pris ou soumis à d'autres actions. La position et l'orientation de cet élément sont contrôlées et repérées par leurs coordonnées généralisées qui sont les coordonnées d'un point particulier de l'organe terminal exprimées dans le repère de base.

Le nombre de degrés de liberté (ddl) d'un mécanisme est le nombre de coordonnées généralisées indépendantes nécessaires pour définir la configuration d'un mécanisme. Il décrit aussi le nombre de mouvements indépendants que peut réaliser l'organe terminal d'un mécanisme. Ces mouvements sont décrits au maximum par trois translations et trois rotations selon des axes particuliers.

L'analyse de l'agencement des différents segments articulés d'un robot permet de dégager deux grandes familles : les robots sériels et les robots parallèles.

1.4.1. Robots sériels

La majorité des manipulateurs existants à l'heure actuelle présentent un caractère anthropomorphique marqué avec une forte ressemblance à un bras humain (voir Figure 1.1). Il s'agit d'une succession de solides, chacun étant relié à son prédécesseur et à son successeur par une articulation à un degré de liberté motorisée. Ce type d'architecture est qualifié de robot sériel.



Figure 1.1 – Robot Sériel

Ces robots ont pour avantage de disposer d'un grand volume de travail et d'être relativement simples à commander, par rapport aux robots parallèles.

Les principaux défauts d'un robot sériel classique sont résumés ci-dessous :

- 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 du à la disposition des segments en série;
- Agencement délicat des liaisons de puissance assurant l'alimentation des actionneurs (Câbles, tuyaux et 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.

1.4.2. Robots parallèles :

A. Définition :

Un robot parallèle contrôle le mouvement de son effecteur au moyen d'au moins deux chaînes cinématiques allant de l'effecteur à la base (voir Figure 1.2). 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.

C'est aussi un Mécanisme en chaîne cinématique fermée dont l'organe terminal est relié à la base par plusieurs chaînes cinématiques indépendantes.

En quelques sortes, l'organe terminal ou effecteur — partie qui agit sur l'environnement, est relié au bâti par plusieurs bras, chaque bras étant une « chaîne cinématique » (association de plusieurs pièces articulées entre elles). Sa mobilité est donc restreinte, puisqu'elle est limitée par les divers bras ; par contre, cela confère une plus grande résistance et précision, puisque les efforts sont répartis.



Figure 1.2 – Robot Parallèle

B. L'origine des robots parallèles :

Les travaux théoriques sur les mécanismes parallèles, et en particulier les hexapodes, remontent à plusieurs siècles quand les géomètres français et anglais étaient particulièrement intéressés par les

polyèdres. Cependant, un des premiers robots parallèles a été conçu par James E. Gwinnett, un agriculteur aux États-Unis, pour lequel il a déposé un brevet en 1928. Son invention était une plate-forme à mouvement sphérique à 3 ddl pour utilisation dans les salles de cinéma.

Une décennie plus tard, et seulement dix-sept ans après que le terme « robot » n'ait été inventé par Karel Čapek, un nouveau robot parallèle a été inventé par Willard L. V. Pollard. Cette invention ingénieuse était un robot parallèle à 5 ddl. Le robot de Pollard a été conçu pour la peinture au pistolet mais, malheureusement, n'a jamais été construit.

En 1934, Willard L. G. Pollard Jr. a déposé un brevet pour une machine de pulvérisation de peinture. L'invention comporte un système de commande et un manipulateur. Le système de commande se compose de films perforés, et le manipulateur est essentiellement un robot à cinq barres.

Pollard Jr. a accordé la licence de son invention à la compagnie DeVilbiss en 1937. En 1941, DeVilbiss, qui deviendra plus tard le premier fabricant de robots industriels, a réalisé un prototype d'un robot de peinture. Cependant, c'était un robot sériel qui n'utilisait que le système de contrôle proposé par Pollard Jr.

En 1947, de l'autre côté de l'Atlantique, un nouveau robot parallèle a été inventé, celui qui est devenu le robot parallèle le plus commun, l'hexapode à jambes télescopiques. Dr Eric Gough, qui a construit ce premier *hexapode octaédrique*, était un ingénieur chez Rubber Co. Dunlop en Angleterre.

La machine de Gough a été inventée pour résoudre les problèmes liés aux charges d'atterrissage des avions. Une machine universelle était nécessaire afin de déterminer les propriétés des pneus qui sont sous charges combinées. Mais l'hexapode octaédrique n'a pas été inventé à partir de rien. A cette époque, les systèmes à trois vérins verticaux et trois vérins horizontaux étaient déjà très communs. Ces hexapodes sont devenus très populaires parce que l'ajustement des vérins était simple et interprétable dans le cas de petites variations. Des systèmes de ce type sont connus sous le terme « table de simulation multiaxes », et sont encore fabriqués aujourd'hui.

La nouvelle fonctionnalité introduite dans la plate-forme de Gough était la disposition symétrique des six jambes. La machine a été construite dans les années 1950. Au tout début, les jambes extensibles étaient réglables manuellement.

Le travail de Gough est demeuré inconnu pendant plus d'une décennie. Pendant ce temps aux États-Unis, Menahem Suliteanu et William R. La Valley avait déposé une demande de brevet en 1962 pour un support d'antenne à 6 ddl composé de tripodes. Puis, en 1965, Everett R. Peterson a déposé une demande de brevet pour un hexapode octaédrique avec des doubles rotules. Sa demande a été cependant précédée de quelques mois par celle de Klaus L. Cappel de l'Institut Franklin qui allait devenir l'un des brevets les plus importants de l'histoire des robots parallèles. L'idée de simulateur de mouvement de Cappel résulte d'une demande faite par le siège social de la Sikorsky Aircraft Division de United Technologies pour la conception et la construction d'un simulateur de vol d'hélicoptère à 6 ddl. Ce fut le premier simulateur de vol basé sur un hexapode octaédrique.

La première licence a été accordée à Link, le principal fabricant de simulateurs de vol, à la fin des années 1960. La première infraction a été commise dans les années 1970 par CAE, le leader actuel de simulateurs de vol.

Pendant ce temps au Royaume-Uni, en 1965, un ingénieur mécanique, D. Stewart, qui ignorait le travail de Gough et de Cappel, a proposé un mécanisme à six pattes télescopiques pour utilisation comme simulateur de vol. Son mécanisme parallèle, cependant, est différent de l'hexapode octaédrique, qui est paradoxalement souvent désigné comme la « plate-forme de Stewart ». L'article de Stewart a suscité des discussions enflammées entre les chercheurs, dont l'un d'un était M. Gough, qui leur a rappelé l'existence de sa machine pour tester les pneus.

Bien que Stewart n'ait été ni l'inventeur de l'hexapode octaédrique ni le précurseur des simulateurs de vol à six pattes, il ne fait aucun doute que son travail a eu un impact majeur sur le développement ultérieur dans le domaine des robots parallèles. En fait, à cause de l'article de Stewart, Cappel a dû rééditer son brevet et en limiter le champ d'application.

Pendant près de deux décennies, les robots parallèles ont attiré peu d'attention, mais au début des années 1980, leur popularité a soudainement commencé à croître et n'a jamais cessé par la suite. En 1985 par exemple, Donald C. Fyler du laboratoire Charles Stark Draper (É-U), a eu l'idée d'utiliser un mécanisme à cinq barres comme robot. Il a affirmé que ce robot était la meilleure alternative au robot SCARA qui avait été inventé en 1979 par le professeur Hiroshi Makino.

Depuis le début des années 1980, des centaines de nouveaux modèles ont été proposés, et presque autant de demandes ont été déposées. Beaucoup de ces architectures parallèles sont véritablement novatrices, mais une seule a permis de développer le robot parallèle le plus réussi pour les applications industrielles : le *robot Delta*.

Au cours de l'été 1985, le professeur Reymond Clavel de l'EPFL en Suisse, a eu l'idée géniale d'utiliser des parallélogrammes pour construire un robot parallèle à 4 ddl pour les opérations de transfert rapide. Il a demandé une série de brevets, sur la base desquels plusieurs compagnies, dont ABB, ont éventuellement obtenu des licences. Pendant la durée de vie de ses principaux brevets, plus de 10 000 unités ont été fabriquées. Aujourd'hui, le design du robot Delta n'est plus protégé et des dizaines de sociétés proposent leurs propres versions, y compris FANUC, Motoman, et Kawasaki.

C. Type de robots parallèles :

Peu utilisés en robotique ludique, les robots parallèles gagnent à être connus car leur réalisation mécanique, certes complexe, est passionnante. Deux des robots parallèles les plus populaires sont l'**hexapode**, utilisé dans la plupart des simulateurs de mouvement, et le **robot Delta**, généralement utilisé pour le transfert rapide. Il y a moins de robots parallèles que de robots sériels qui sont utilisés, mais la variété de robots parallèles est plus grande.

Les nouveaux procédés de construction (Lego, CNC, imprimantes 3D) permettent de réaliser en amateur des robots parallèles jusque-là réservés à la recherche et à l'industrie.

Un **robot parallèle** contient plusieurs activateurs mécaniques agissant ensemble (en... parallèle) pour mettre en mouvement la même pièce. Ils sont donc tous reliés, généralement d'une base fixe à une pièce mobile.

1. L'hexapode :

Voici un exemple (voir Figure 1.3) de robot parallèle sous forme d'une plateforme à 6 degrés de liberté : grâce aux six vérins, l'hexagone supérieur peut se lever et se baisser (axe z), se déplacer d'avant en arrière (axe x) ou de gauche à droite (axe y), mais également pivoter sur lui-même (angle de lacet), se pencher en avant/arrière (angle de tangage) et se pencher sur les côtés (angle de roulis).



Figure 1.3 – Robot Parallèle Hexapode

Ce type de robot est utilisé dans les simulateurs de vol grâce à la grande liberté de mouvement.

Le robot a été construit en 1987, pèse 35 kg et est capable de mouvoir une charge de 600 kg avec une précision de 10 microns (millionnièmes de mètre ou mille fois plus petit qu'un millimètre).

Chacun des vérins a un déplacement linéaire, et est fixé à la base fixe et à la plateforme mobile par des rotules libres : il n'y a pas d'autres moteurs et c'est l'interaction synchronisée des 6 vérins qui

produit le mouvement souhaité : un système informatisé compose les valeurs des 6 mouvements souhaités pour calculer les vitesses de chaque vérin et leur donner la position nécessaire.

L'électronique de commande est transportable et l'ensemble se pilote avec un simple ordinateur.

2. Robots parallèles (Delta) :

Certains robots parallèles sont appelés "delta" à cause de la forme triangulaire de leurs longs membres.

Ces machines servent habituellement au "pick & place" : un dispositif asservi pour récupérer des objets et les déplacer à un endroit bien précis, comme pour ranger des chocolats dans une boîte ou pour positionner les composants à souder sur une carte électronique. (Voir Figure 1.4)



Figure 1.4 – Robot Parallèle (Pick & place)

Les robots delta commencent à séduire les modélistes par la haute technicité nécessaires. Certains ont même réussi à en construire en **Lego Mindstorms**.

Dans un *robot parallèle*, l'effecteur est relié à la base par l'intermédiaire de plusieurs chaînes de liens interconnectés. En d'autres termes, un robot parallèle a au moins deux « jambes » (ou « bras »). La

plupart de ses articulations ne sont pas actionnées, et beaucoup de ces *articulations passives* ont plusieurs *degrés de liberté (ddl)* (par exemple, les articulations sphériques, universelles, et planes). (Voir Figure 1.5)

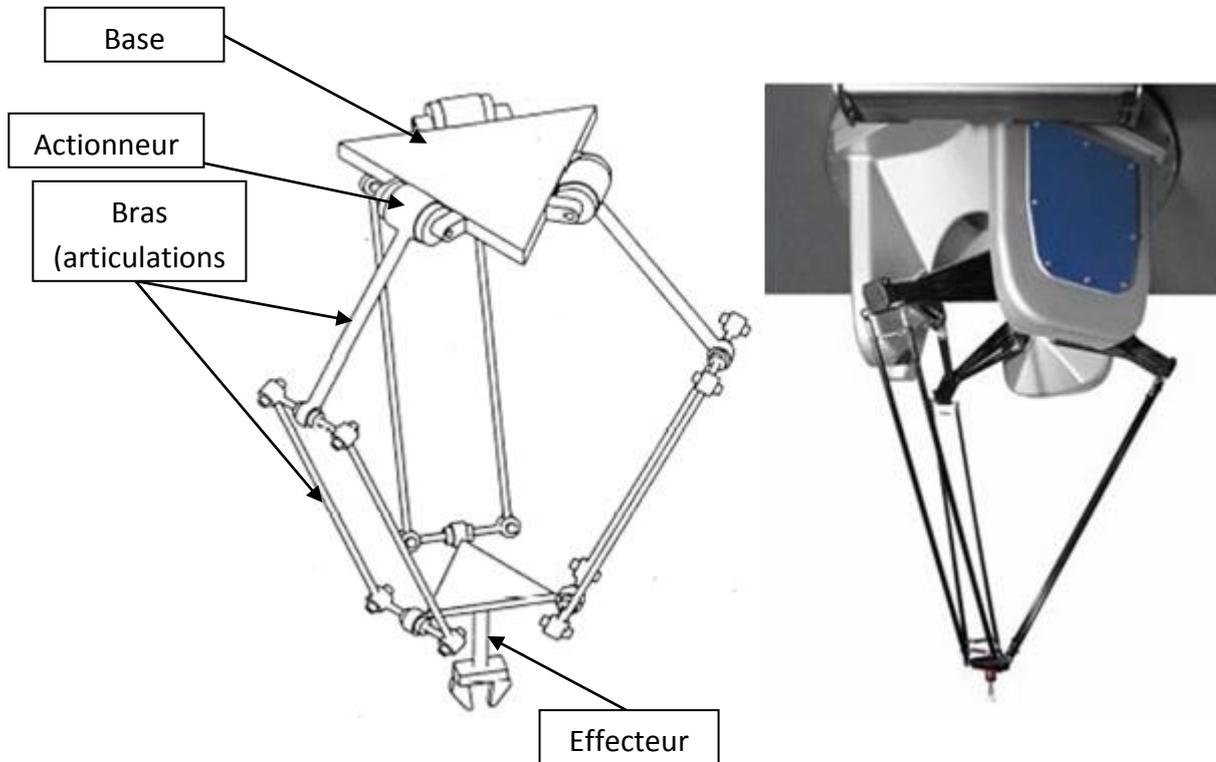


Figure 1.5 – Robot Parallèle

D. Est-ce que les robots parallèles sont difficiles à concevoir?

Ces jours-ci, concevoir un nouveau robot parallèle n'est pas facile. Des centaines de designs ont déjà été brevetés. Cependant, le développement d'un robot parallèle déjà connu, pour une application non- exigeante (par exemple pour l'impression 3D) est relativement simple.

En fait, pour concevoir un robot qui peut suivre des trajectoires prescrites avec son effecteur, tout ce qu'il faut faire est de résoudre sa cinématique inverse. Il s'agit de calculer les positions nécessaires des moteurs, pour la *pose* (position et orientation) désirée de l'effecteur. Pour un robot parallèle typique, c'est assez simple. Par exemple, la cinématique inverse de l'hexapode à pattes télescopiques se résume à trouver la distance entre les points d'attache de chaque patte.

L'*espace de travail* du robot doit alors être calculé. Il existe plusieurs méthodes complexes pour le faire, mais une simple méthode de discrétisation peut également être utilisée. Avec cette méthode, la cinématique inverse est résolue pour chaque pose potentielle de l'effecteur, et une vérification des contraintes est faite. Pour la plupart des robots, il existe plusieurs solutions à la cinématique inverse, aussi appelées *modes de fonctionnement*. Pour les robots parallèles complexes, il est en général impossible de représenter graphiquement l'espace de travail et il est donc commun de calculer l'*espace de travail à orientation constante*. (Voir Figure 1.6)



Figure 1.6 – L'espace de travail du robot IRB 360 (FlexPicker) de ABB

Un des problèmes majeurs des robots parallèles est la présence de *singularités de type 2*. Ce sont des configurations dangereuses où l'effecteur n'est plus totalement contraint. Dans certains robots parallèles ces configurations sont simples à décrire (par exemple dans le DexTAR, elles se produisent lorsque les liens distaux sont alignés), mais dans la plupart des robots elles sont très complexes à analyser. Les singularités de type 2 doivent être prises en compte lors du calcul de l'espace de travail, car elles sont difficiles à franchir et elles coupent l'espace de travail en sections.

Dans certaines applications, telles que la mesure de coordonnées ou de la manipulation rapide, la *cinématique directe* doit également être résolue. Il s'agit de calculer les poses possibles pour l'effecteur, compte tenu des positions des moteurs. Pour certains robots parallèles cette tâche est triviale pour d'autres, elle est relativement simple, mais pour la plupart, elle est très difficile. Par exemple, la cinématique directe du tripode d'ALIO Industries se résume à la résolution d'un polynôme mono-variable de degré 4.

Toutefois, ce polynôme est de degré 40 dans le cas de l'hexapode à jambes télescopiques, ce qui signifie que l'hexapode peut avoir 40 solutions différentes à sa cinématique directe, appelés *modes d'assemblage*. Pour cette raison, une méthode itérative numérique est le plus souvent utilisée.

Enfin, comme avec tout autre mécanisme, les robots parallèles doivent être optimisés. Puisque la performance de la plupart des robots parallèles varie considérablement d'une pose à l'autre, et puisque la plupart des critères de performance sont antagonistes, la conception optimale de la plupart des robots parallèles est un problème complexe. Malheureusement, il n'y a pas de procédure d'optimisation unique des robots parallèles.

E. Qu'y a-t-il de si spécial à propos des robots parallèles ?

On dit souvent que les robots parallèles sont plus rigides, plus rapides et plus précis que les robots sériels. La vérité est beaucoup plus complexe cependant, puisque les robots parallèles diffèrent énormément les uns des autres.

L'hexapode MB-EP-6DOF/60/14000KG de Moog, par exemple, peut déplacer jusqu'à 14 000 kg, tandis que l'immense robot industriel KR 1000 1300 TITAN PA de KUKA, ne peut soulever que 1 300 kg. La raison de cette différence est que les pattes d'un hexapode ne sont soumises à aucune flexion ou torsion, ce qui n'est pas le cas de nombreux robots parallèles, et il n'est pas non plus toujours vrai que la charge utile dans un robot parallèle est supportée par plusieurs pattes.

Il est vrai que les moteurs de nombreux robots parallèles sont fixés à la base. C'est pour cette raison que les robots Delta sont très rapides. Cependant, des progrès récents ont permis le développement de robots sériels extrêmement rapides.

Enfin, on peut affirmer que la plupart des appareils de positionnement de précision multi-axes sont basés sur des robots parallèles, pour la plupart des hexapodes et des tripodes. Cependant, cela s'explique moins par le fait que les erreurs dans un robot sériel s'accumulent, alors qu'elles sont réparties dans un robot parallèle, et serait plus dû au fait que les hexapodes et les tripodes sont rigides.

Le vrai mérite de robots parallèles est qu'il y a des centaines d'architectures possibles, chacune ayant des avantages et des inconvénients très spécifiques. Certains robots parallèles sont parfaits pour l'usinage ou pour la simulation de mouvement, parce qu'ils sont rigides. D'autres sont parfaits pour les opérations de transfert rapide parce que leur partie mobile est légère. Les robots à câbles, par exemple, sont idéals pour couvrir une très large zone de travail tel qu'un stade.

Il y a actuellement plus d'un million de robots industriels en service, et la grande majorité d'entre eux sont sériels. Cependant, la situation va sûrement changer.

F. Les robots parallèles et pour quelles applications ?

Des dizaines de compagnies proposent actuellement des robots parallèles, la plupart étant des hexapodes et des robots Delta. Dans le cas de la simulation de mouvement, l'une des trois applications les plus populaires pour robots parallèles, le leader de l'industrie est Moog, qui a vendu plus de 1 400 hexapodes.

La deuxième application la plus populaire pour les robots parallèles est l'opération de transfert rapide. C'est surtout les robots Delta qui sont utilisés à cette fin. Il y a cependant une nouvelle tendance, les robots à quatre bras identiques.

La troisième application la plus populaire pour les robots parallèles est le positionnement précis. La plupart de ces robots sont basés sur les hexapodes à pattes télescopiques, mais les tripodes et les hexapodes à pattes non extensibles sont aussi utilisés.

Les mécanismes parallèles sont également le premier choix pour la conception de dispositifs haptiques. La manette de jeu Falcon à 250\$ de Novint est un exemple de l'omniprésence des mécanismes parallèles.

Il y a aussi plusieurs machines-outils basées sur des mécanismes parallèles et qui ont des applications plus variées.

G. Robots parallèles existants :

1. Robots parallèles à deux degrés de liberté :

Les robots parallèles à deux degrés de liberté sont utilisés pour le pick-and-place (prise et pose d'un objet) et sont des mécanismes capables de réaliser deux translations (Voir Figure 1.7).



Figure 1.7 – Manipulateur à deux degrés de liberté

2. Robots parallèles à trois degrés de liberté :

a) Le robot Delta :

Les robots parallèles à trois degrés de liberté en translation présentent un intérêt important pour les opérations de transfert. En conséquence différents prototypes ont été proposés. Le plus utilisé a été développé initialement à l'Ecole Polytechnique de Lausanne et est commercialisé sous le nom " Delta " par la société Demarex (Voir Figure 1.8).

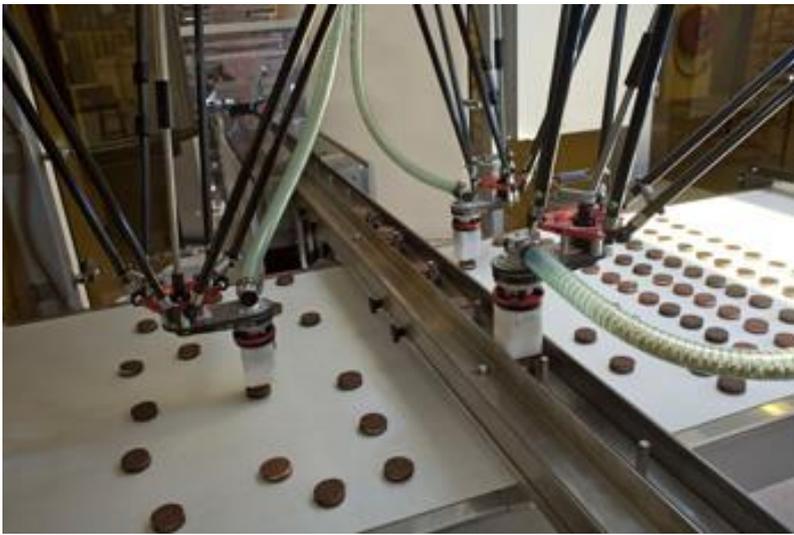


Figure 1.8 – Robot Delta à trois degrés de liberté

Ce robot était muni de trois moteurs rotatifs reliant une nacelle par l'intermédiaire de trois chaînes cinématiques identiques. Ces dernières sont composées d'un bras et d'un système à quatre barres en chaîne fermée. Ce système est communément nommé "parallélogramme spatial" car sa fonction est de garantir le parallélisme entre deux solides dans l'espace, à condition que les barres restent coplanaires deux à deux (ce qui est vérifié sur un Delta quand toutes les chaînes cinématiques sont assemblées). Ces robots parallèles présentent de nombreux avantages pour les tâches de pick-and-place.

b) Le robot Orthoglide :

L'Orthoglide est une machine de type parallèle possédant 3 articulations prismatiques orthogonales reliées au bâti (Voir Figure 1.9). La plate-forme mobile est connectée aux articulations prismatiques par 3 parallélogrammes articulés et bouge dans l'espace cartésien xy-z avec une orientation fixe.

L'Orthoglide est dédié à l'usinage à grande vitesse, car son architecture se rapproche des machines standards d'architecture série PPP (espace de travail Cartésien régulier et performances uniformes) et avec, en plus, les propriétés des structures parallèles (inerties moins importantes et meilleures performances dynamiques). Son espace de travail est proche d'un cube et ne possède aucune singularité.

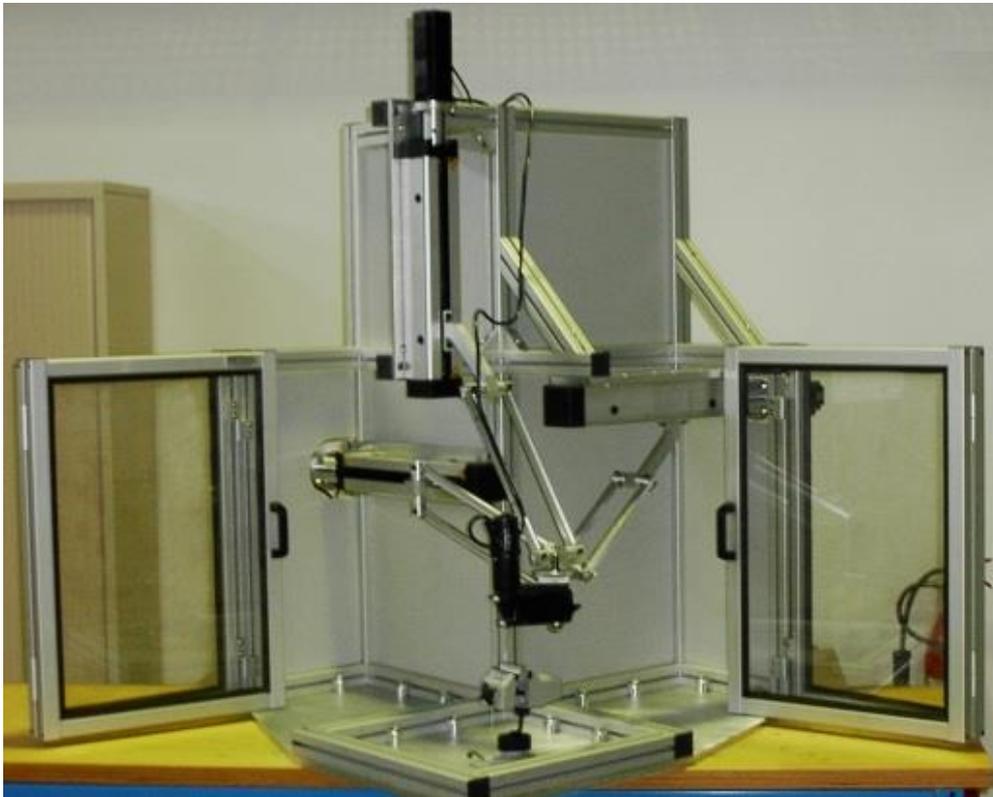


Figure 1.9 – Robot Orthoglide

3. Robots parallèles à quatre degrés de liberté :

Afin de répondre aux besoins des applications de manipulation d'objets, l'architecture Delta, originalement pourvue de trois degrés de liberté, fut modifiée afin d'y ajouter une quatrième mobilité. Ainsi, la rotation est obtenue en ajoutant une liaison rotoïde à la plateforme dont la rotation est

commandée à l'aide d'une chaîne cinématique de type RUPU (Voir Figure 1.10). Cette "patte télescopique" permet de réaliser une rotation illimitée, mais a l'inconvénient majeur de limiter les performances dynamiques du Delta ainsi modifié, et ce, d'autant plus pour des robots de grande taille. En effet, la fiabilité et la durée de vie de cette patte passive se trouvent réduites lorsque le robot réalise des mouvements dont les accélérations sont importantes.



Figure 1.10 – Robot Delta à quatre degrés de liberté

a) Le robot H4 symétrique :

Ce robot développé par Company [Com 99b] fut le premier mécanisme à introduire le concept de nacelle articulée. Cette notion peut être définie comme étant un dispositif composé d'au moins deux corps solides placés à l'extrémité des chaînes cinématiques du robot, et dont une mobilité interne est utilisée pour produire un degré de liberté au niveau de l'organe terminal.

Le H4 est réalisé à l'aide de quatre chaînes de type Delta actionnées à l'aide de moteurs rotatifs ou linéaires (Voir Figure 1.11). Sa nacelle comporte trois corps et deux liaisons; elle est ainsi composée de deux pièces liées par une barre transversale à l'aide de deux liaisons rotoïdes et dont la forme représente un "H". L'amplitude de la rotation ainsi produite est de $\pm 45^\circ$. Un système d'amplification peut y être ajouté afin de réaliser une amplitude de rotation suffisamment importante pour les applications de pick-and-place ($\pm 180^\circ$).

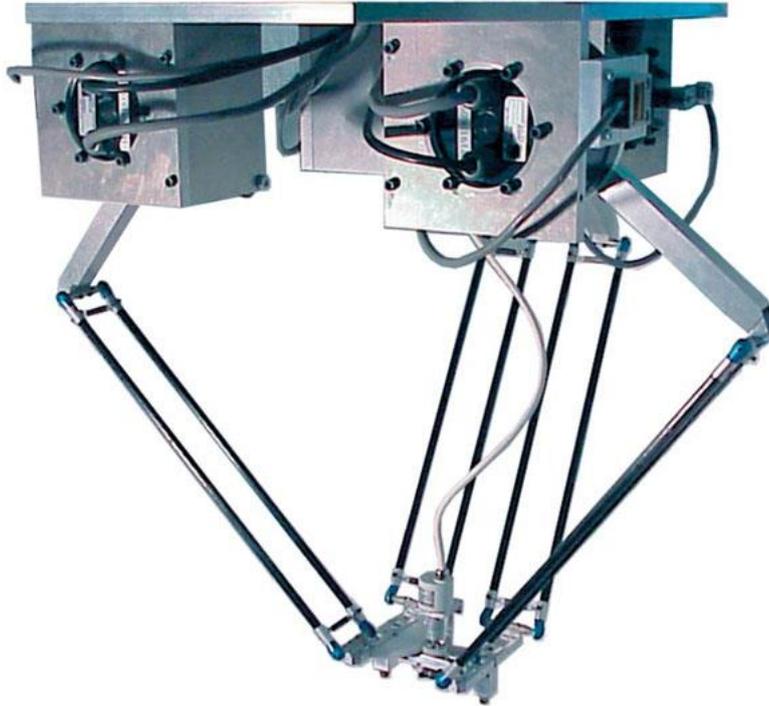


Figure 1.11 – Le robot H4 à quatre degrés de liberté

4. Robots parallèles à six degrés de liberté :

Plusieurs formes d'architectures parallèles à six degrés de liberté sont possibles et la plus répandue est la plate-forme de Gough-Stewart (Voir Figure 1.12). Il s'agit d'un manipulateur à six degrés de liberté, dont la plate-forme est déplacée par l'élongation de six actionneurs linéaires. Son application la plus connue est le simulateur de vol (Voir Figure 1.13) en raison, entre autres, de la masse élevée de la cabine (jusqu'à 15 000 kg) à laquelle on doit faire subir des mouvements. Les simulateurs de vol sont utilisés pour l'entraînement des pilotes au sol.



Figure 1.12 – Plate-forme de Gough-Stewart



Figure 1.13 – Simulateur de vol

Les hexapodes sont aussi utilisés afin de positionner et orienter précisément des objets de masse importante. Dans ce type d'applications, la dynamique de la structure n'est pas un critère de première importance, ce qui importe, c'est de donner une position précise à un objet pouvant être massique. Dans l'exemple de la (Voir Figure 1.14), l'hexapode présenté permet de positionner une charge de 200 kg suivant l'axe z avec une répétabilité de $\pm 1\mu\text{m}$ dans un volume de travail de $100 \times 100 \times 50 \text{ mm}^3$.

Figure 1.14 – Hexapode



Un autre exemple de robot parallèle léger est l'Hexa. Ce mécanisme à six degrés de liberté est composé de six chaînes cinématiques actionnées par des moteurs rotatifs. 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 (Voir Figure 1.15).



Figure 1.15 – Robot Hexa à six degrés de liberté

5. Robots hybrides

La définition exacte d'un mécanisme hybride est évasive. Dans cette classification nous considérerons comme hybride un mécanisme qui comporte plusieurs chaînes cinématiques reliant le bâti à l'organe terminal et dont une des chaînes au moins comporte plus d'un actionneur. Parmi les machines hybrides que l'on a cataloguées, on peut distinguer trois catégories :

a) les mécanismes à porteur parallèle et poignet sériel:

Le robot Tricept de Neos (Voir Figure 1.16) est une machine hybride : un mécanisme porteur d'architecture parallèle porte un poignet série. Une particularité de ce robot est la présence d'une patte passive (non-actionnée) qui contraint la cinématique du robot.

Ce robot a connu un réel succès commercial mais il semble que dans le domaine de la machine-outil les performances obtenues soient en deçà de celles espérées.



Figure 1.16 – Tricept 845 (Neos Robotics)

b) les mécanismes à porteur série et poignet parallèle :

La machine-outil Sprint Z3 de DS Technologies possède un mécanisme porteur x-y et un poignet d'architecture parallèle z-A-B. Destinée à l'industrie de l'aéronautique, cette machine comprend un portique d'architecture classique d'une course de 60 mètre, et une tête d'architecture parallèle (Voir Figure 1.17).



Figure 1.17 – La machine-outil Sprint Z3

c) les autres mécanismes hybrides :

Un exemple des robots série-parallèles, est le robot de l'INRIA constitué d'un robot parallèle à six degrés de liberté appelé le poignet actif monté sur un robot SCARA (Voir Figure 1.18). Un autre robot utilisant le même concept (Voir Figure 1.19), ce robot constitue un macro/mini manipulateur où le robot DELTA est utilisé comme poignet d'un robot SCARA, l'avantage de cette configuration est de dépasser la limitation de l'espace de travail. Ce concept de combinaison série-parallèle a aussi été utilisé dans le robot ARTISAN avec une structure parallèle type 3RPS.



Figure 1.18 – Poignet actif

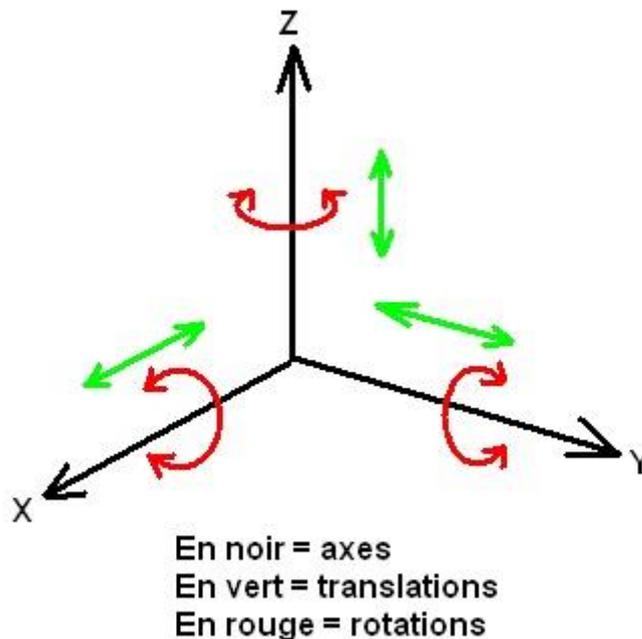


Figure 1.19 – Mini manipulateur

1.5. Prototypes d'un robot industriel :

1.5.1. Degré de liberté d'un robot :

Le nombre de d.d.l. (degré de liberté) d'un mécanisme est le nombre de paramètres indépendants qui permettent de définir la position du mécanisme à un instant donné du mouvement.



Ceci revient à dire que le degré de liberté d'un robot est égale au nombre de mouvement effectué par ce dernier.

1.5.2. Exemple d'un prototype à 2 DDL:

Les robots parallèles industriels générant uniquement 2 ddl en translation sont peu nombreux sur le marché. Cette section recense différentes architectures, qu'elles soient capables de positionner un point dans un plan, de contraindre l'orientation de l'effecteur ou qu'elles admettent une architecture spatiale.

Dans les cas très simples de lignes de production lentes ou intermittentes (arrêt du convoyeur lors de l'opération de prise-dépose), seuls deux ddl sont nécessaires : les translations suivant x et z. Effectivement, la trajectoire ADEPT définie plus haut peut être contenue dans un plan. C'est pourquoi les architectures robotisées générant des mouvements 2T sont bien adaptées. De manière à éviter l'arrêt de la ligne de production, les robots 2T peuvent être positionnés en complément d'autres axes de mouvements plus lents. Ils sont aussi appréciés afin de compléter une machine d'assemblage plus complexe de sorte à effectuer une tâche particulière.

Le robot Delta en Figure 1.20 marque le début de l'ère des applications de la robotique parallèle « légère ». Il a été développé par Clavel en 1989 [Clavel, 1989] à l'Institut de Microtechnique, EPFL de Lausanne.

Les déplacements de la plate-forme mobile résultent des mouvements de trois jambes articulées (2) reliant la plate-forme mobile (3) à la base (1). Les rotations de la plateforme sont éliminées grâce à trois parallélogrammes articulés (2). Le petit côté de chaque parallélogramme restant constamment parallèle à une direction fixée bloque alors, pour chaque jambe, une rotation de la plate-forme dans l'espace. Ce mécanisme possède une symétrie ternaire lorsque la plate-forme est en position centrée.

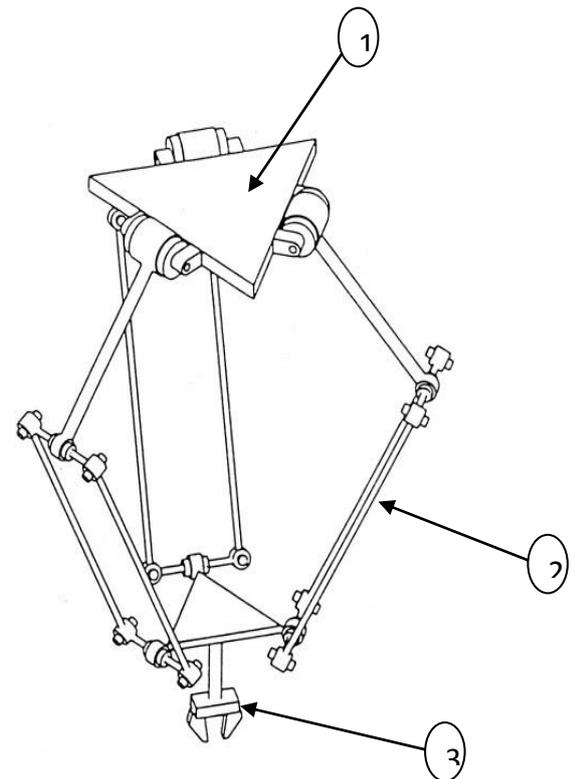


Figure 1.20 – Chaines cinématiques du Robot Delta

1.5.3. Robot Delta à 3 DDL:

Pour des applications rapides où un suivi du convoyeur par le robot est nécessaire, trois translations sont obligatoires. Le tracking permet de suivre et de prendre un objet en mouvement sur un convoyeur dans le plan $x y$ et de le poser sur un autre convoyeur en mouvement (déplacement selon z).

- Dans le cadre de notre mémoire le robot Delta est à 3 DDL.

1.6. Avantages et inconvénients des robots delta :

La mise en parallèle de plusieurs chaînes cinématiques, chacune ne comportant qu'un actionneur, conduit généralement aux avantages cités dans :

- La 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.
 - Facilité 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.

Par contre, les inconvénients des robots parallèles sont :

- 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.

Il est possible de distinguer deux types d'architectures parallèles :

- Les robots « légers » introduits par Clavel désignent les robots pour lesquels il est possible de positionner les moteurs sur la base (ils sont nommés mécanismes rapides par Merlet). Ces robots sont utiles pour les applications nécessitant des vitesses et accélérations élevées. Dans cette catégorie se trouvent les robots parallèles à câbles permettant aussi de déplacer de lourdes charges dans un grand espace de travail.

- Les robots « non légers », utiles pour les applications de déplacement de charges élevées. Les actionneurs ne sont pas forcément sur la base, les jambes doivent travailler en traction-compression, comme la plate-forme de Gough.
- ✚ Les robots parallèles offrent la possibilité d'atteindre de plus grandes vitesses et accélérations que les robots sériels. Une meilleure répétabilité 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.

Le vrai mérite de robots parallèles est qu'il y a des centaines architectures possibles, chacune ayant des avantages et des inconvénients très spécifiques. Certains robots parallèles sont parfaits pour l'usinage ou pour la simulation de mouvement, parce qu'ils sont rigides. D'autres sont parfaits pour les opérations de transfert rapide parce que leur partie mobile est légère. Les robots à câbles, par exemple, sont idéals pour couvrir une très large zone de travail tel qu'un stade.

1.7. Conclusion :

Dans ce chapitre nous avons présenté un bref historique des robots à structures parallèles et avec un ordre chronologique de leur développement depuis le premier hexapode octaédrique de Gough-Stewart en 1947. Ensuite on a donné quelques définitions utiles à l'état de l'art de la robotique parallèle, tout en citant quelques exemples de robots parallèles classés selon le nombre de degrés de liberté. Enfin on termine par citer les avantages et les inconvénients des robots parallèles.

2.1. Introduction :

La conception des robots nécessitent le calcul de certains modèles mathématique tels que les modèles de transformations 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 :

- Les modèles géométriques direct et inverse.
- Les modèles cinématiques direct et inverse.

La description d'un robot est une étape importante pour l'élaboration des modèles, qu'ils soient géométriques, cinématiques ou dynamiques. Et la modélisation d'un robot, considéré comme étant un système mécanique articulé, actionné et commandé, consiste à en établir un modèle mathématique.

Plusieurs méthodes et notations ont été proposées pour la description des paramètres géométriques des structures : série, arborescentes ou comportant des boucles fermées.

La méthode utilisée pour la description de notre robot parallèles est la modèle géométrique inverse est direct, ce dernier fait principalement appel aux méthodes de la géométrie analytique, et il emprunte aussi quelques notions à la théorie des graphes pour le traitement des chaînes cinématiques.

2.2. Modélisation géométrique d'un robot parallèle :

La modélisation géométrique est l'ensemble des outils mathématiques, numériques et informatiques qui combinés permettent de construire un modèle virtuel (ou modèle informatique) d'un objet réel. Cet objet peut être plus ou moins complexe, plus ou moins schématisé. Il peut être le fruit de l'imagination, d'une tendance ou plutôt une solution plus ou moins exacte d'un problème physique donné, voire un compromis entre les deux.

La modélisation géométrique sous-entend d'être en mesure de réaliser la construction et l'assemblage de formes élémentaires pour créer des objets de plus en plus complexes en respectant des contraintes topologiques.

La modélisation géométrique implique aussi de savoir reconstruire des objets à partir de numérisation d'objets existants sans modèle, mais aussi d'objets déjà modélisés et fabriqués pour contrôler l'écart entre le modèle virtuel et l'objet fabriqué.

2.1.1. Modélisation géométrique directe et inverse d'un robot :

Le modèle géométrique inverse permet de déterminer la configuration des liaisons, en fonction de la configuration (position et orientation) de l'effecteur d'un robot. Il existe 2 types de liaisons, les liaisons pivots et les liaisons glissières.

Chapitre 2 Modélisation géométrique de la structure parallèle

- La liaison pivot permet des mouvements de rotation et est caractérisée par un angle.
- La liaison glissière permet des mouvements de translation et est caractérisée par une distance.

Le modèle géométrique inverse est utilisé pour caractériser le fonctionnement d'un bras manipulateur. Concrètement le modèle géométrique inverse permet de calculer la position de chaque moteur (liaison) du robot en fonction de la position et de l'orientation de l'organe terminal (effecteur).

La fonction inverse qui consiste à déterminer la configuration de l'effecteur en fonctions de la configuration des liaisons est le modèle géométrique direct.

2.3. Etude théorique :

2.3.1. Etude géométrique inverse :

Les principes de la géométrie inverse sont les suivants :

- les membres sont des barres rigides, de longueur donnée, reliés entre eux par des articulations.
- les articulations supportent une certaine torsion maximale, qui varie selon la direction ou la vitesse.
- les mouvements sont supposés suffisamment réguliers - pour paraître naturels.

Mathématiquement, on obtient une équation dépendant de plusieurs variables. Dans l'exemple du bras humain, les paramètres seraient les différents angles formés par les articulations, et l'équation donnerait la distance entre l'index et l'objet à atteindre. On cherche à minimiser cette fonction, pour qu'il y ait contact entre l'index et l'objet. Dans l'extrême majorité des cas, des solutions analytiques n'existent pas, alors il faut utiliser des techniques d'optimisation non linéaire.

A) Calcule de l'angle θ_1 :

✚ Pour obtenir l'angle de rotation θ_1 nous procéderons comme suit :

On cherche en premier les longueurs ou distance B_y et B_z :

Chapitre 2 Modélisation géométrique de la structure parallèle

Dans notre nous allons chématiser le mécanisme comme suit:

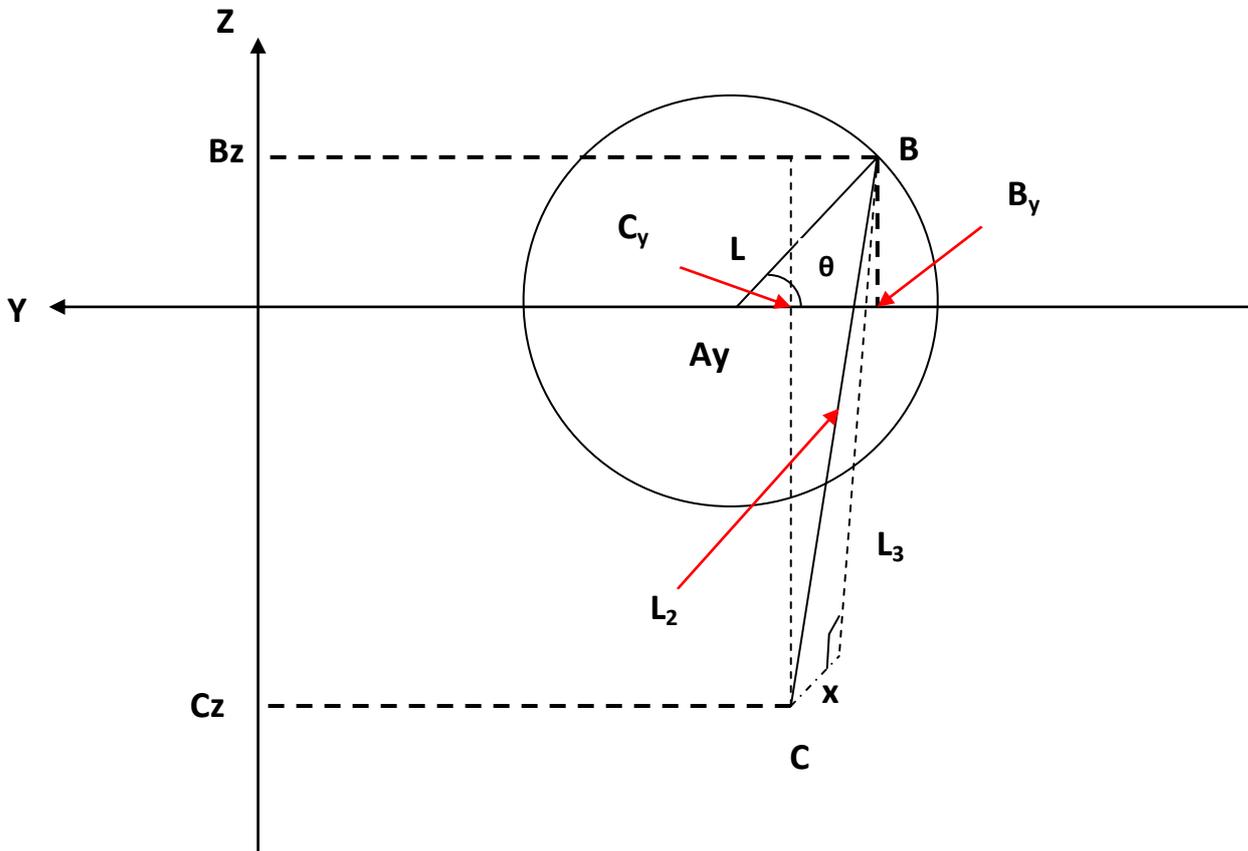


Figure a – Représentation des distances d'un bras du Robot

On considère que $Z = C_z$

Du théorème de Pythagore nous obtenons:

$$(B_y - A_y)^2 + B_z^2 = L_1^2 \quad (1)$$

$$(B_y - C_y)^2 + (B_z - C_z)^2 = L_3^2 = L_2^2 - C_x^2 \quad (2)$$

De l'équation (1) on obtient $\Leftrightarrow B_y^2 + A_y^2 - 2 B_y A_y + B_z^2 = L_1^2$.

De l'équation (2) on obtient $\Leftrightarrow B_y^2 + C_y^2 - 2 B_y C_y + B_z^2 + Z^2 - 2 B_z Z - L_3^2 = 0$

Chapitre 2 Modélisation géométrique de la structure parallèle

$$B_z^2 = L_1^2 - B_y^2 - Ay^2 + 2 B_y Ay$$

En procédant au remplacement dans l'équation N° (2) on trouve :

$$B_z = \frac{2(Ay - Cy)}{2Z} B_y + \frac{Cy^2 + L_1^2 - Ay^2 + Z^2 - L_3^2}{2Z}$$

d'où :

$$B_z = \alpha B_y + \beta$$

De l'équation (1) on obtient $\Leftrightarrow (B_y - Ay)^2 + (\alpha B_y + \beta) - L_1^2 = 0$

$$(B_y - Ay)^2 + \alpha^2 B_y^2 + 2 \alpha \beta B_y + \beta^2 - L_1^2 = 0$$

$$B_y^2 + Ay^2 - 2 B_y Ay + \alpha^2 B_y^2 + 2 \alpha \beta B_y + \beta^2 - L_1^2 = 0$$

On obtient une équation du second degré et avec deux solutions.

Dans le cas pratique nous considérerons la valeur la plus grande de B_y .

En remplaçant α et β par leur valeur respective on trouve :

$$B_{y1} = (-b - D)/(2*a)$$

$$B_{y2} = (-b + D)/(2*a)$$

Sachant que

$$a = 4*Z^2 + (2*Ay - 2*Cy)^2;$$

$$b = -8* Ay * Z^2 + (4*Ay - 4*Cy)*(L_1^2 - L_3^2 - Ay^2 + Cy^2 + Z^2);$$

$$c = 4*Ay^2 * Z^2 - (4*Z^2) * (L_1^2) + (L_1^2 - L_3^2 - Ay^2 + Cy^2 + Z^2)^2;$$

$$D = (b^2 - 4*a*c).$$

Chapitre 2 Modélisation géométrique de la structure parallèle

En dernier nous obtiendrons que :

$$L \times \cos \theta_1 = B_z$$

$$L \times \sin \theta_1 = B_y$$

A) Calcul des angles θ_2 et θ_3 :

Pour l'obtention des angles de rotation θ_2 et θ_3 nous procéderons comme suit :
On utilise une matrice de rotation :

➤ Matrice de rotation :

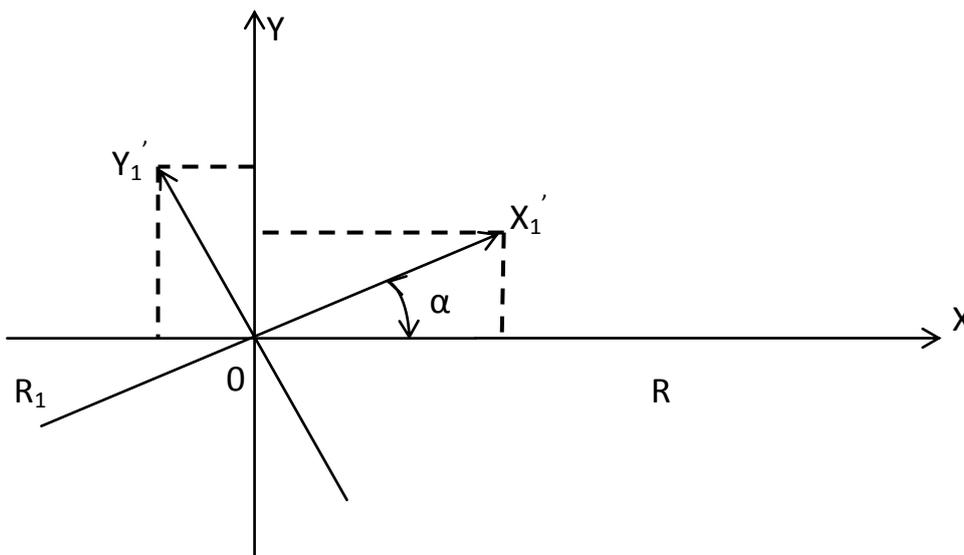


Figure b - Schéma matrice de rotation

Repère R_1 représente le Repère R après une rotation de $+\alpha$

$0 x_1'$ et $0 y_1'$ vecteurs unitaires

Les coordonnées de x_1 $\begin{bmatrix} \cos \alpha \\ \sin \alpha \end{bmatrix}$ et y_1 $\begin{bmatrix} -\sin \alpha \\ \cos \alpha \end{bmatrix}$

Chapitre 2 Modélisation géométrique de la structure parallèle

La matrice R =
$$\begin{bmatrix} \cos \alpha & -\sin \alpha \\ \sin \alpha & \cos \alpha \end{bmatrix}$$

R est appelée matrice de rotation, elle est notée R_1^0 pour montrer que nous voulons avoir la position de repère (1) dans le repère (0)

Si un point P est connu dans le repère (1) et nous cherchons ses coordonnées dans le repère (0) il suffit d'écrire $P^0 = R_1^0 P^1$

P^1 : coordonnées dans le repère (1)

P^0 : coordonnées dans le repère (0)

Pour notre robot nous allons calculer les coordonnées (x, y, z) dans un repère fixe (0, x, y, z).

Etant donné la symétrie du système le même travail est effectué comme application à notre cas, et ceci revient à faire tourner ce repère de $\pm 120^\circ$ autour de l'axe Z.

Pour désigner les coordonnées du point C dans un repère fixe (0, x, y, z) il faut établir les coordonnées (x, y) de ce point (Z n'est pas affecté) dans les deux autres repères, pour ensuite recalculer ces coordonnées dans ce repère fixe, cela revient à calculer θ_1 dans le repère (0, x, y, z) ou ($\alpha = 0$).

Ensuite prendre $\alpha = 120$ et faire le même calcul pour trouver θ_2 et $\alpha = -120$ pour calcul θ_3 .

nous utilisons la figure a ; nous pouvons considérer que cela revient à faire tourner le repère 0 vers le repère 1 d'un angle α ou bien de faire tourner le repère 1 vers le repère 0 d'un angle ($-\alpha$) donc la matrice de rotation peut se définir en remplaçant α par $-\alpha$

$$R' = \begin{bmatrix} \cos \alpha & \sin \alpha \\ -\sin \alpha & \cos \alpha \end{bmatrix}$$

Avec $\alpha = 120$ pour θ_2 et $\alpha = -120$ pour θ_3

2.3.2. Etude géométrique directe :

Le modèle géométrique direct d'un robot permet de calculer les coordonnées opérationnelles donnant la situation de l'organe terminal en fonction des coordonnées articulaires.

Il permet de déterminer la configuration (position, orientation) de l'effecteur d'un robot en fonction de la configuration de ses liaisons. Il existe 2 types de liaisons, les liaisons pivots et les liaisons glissières :

- La liaison pivot permet des mouvements de rotation, elle est caractérisée par un angle.
- La liaison glissière permet des mouvements de translation et est caractérisée par une distance.

Le modèle géométrique direct est utilisé pour caractériser le fonctionnement d'un bras manipulateur. Concrètement, le modèle géométrique direct permet de calculer la configuration de l'organe terminal (effecteur) du robot en fonction de la position de chaque moteur (liaison) du robot.

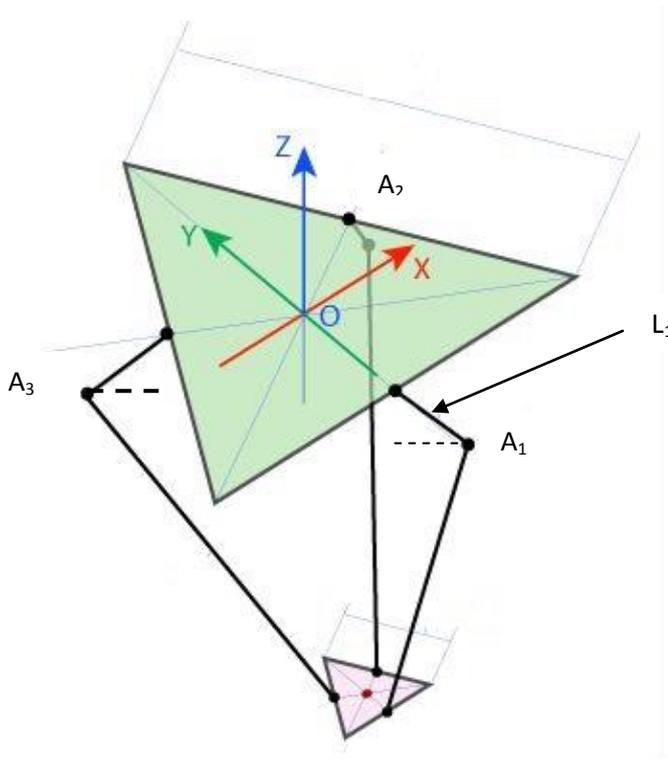


Figure 2.1 représentation géométrique du robot Delta

Chapitre 2 Modélisation géométrique de la structure parallèle

A_1 , A_2 et A_3 représente les 3 centres des sphères décrites dans la plateforme mobile si celle-ci était liée à l'un des 3 bras du.

Le point $P(x_0, y_0, z_0)$ correspond à l'intersection des 3 sphères, dans le schéma ci-dessus connaissons les angles θ_1 , θ_2 et θ_3 et cherchons à déterminer les coordonnées (x_0, y_0, z_0) du point P (position de l'effecteur final).

➤ Dans la plateforme mobile :

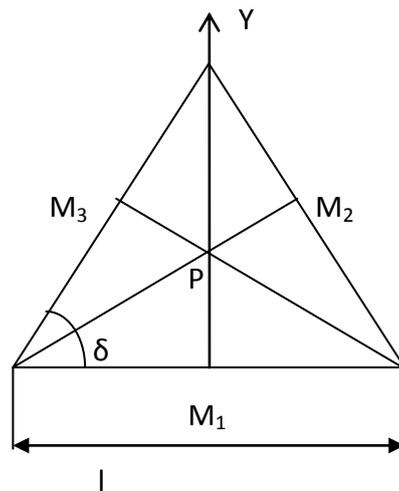


Figure c - Schéma Plateforme mobile

De la figure c nous obtiendrons :

$$P M_1 = P M_2 = P M_3 \text{ d'où } \delta = 30^\circ$$

En revenant à la figure 2.1 et en appliquant le même principe nous obtenons :

$$A_1 A_1' = A_2 A_2' = A_3 A_3' = P M_1$$

Ce qui implique :

$$A_1 A_1' = A_2 A_2' = A_3 A_3' = \frac{1}{2\sqrt{3}}$$

Chapitre 2 Modélisation géométrique de la structure parallèle

➤ Dans la plateforme fixe :

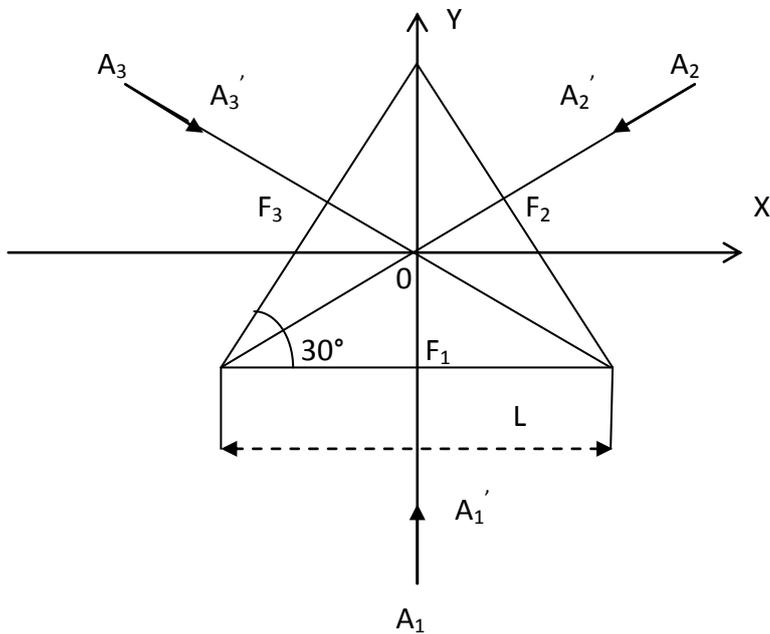


Figure d. Schéma Plateforme fixe

De la même manière de ce qui précède, nous obtenons que :

$$A_1 F_1 = L_1 \cos \theta_1$$

$$A_2 F_2 = L_1 \cos \theta_2$$

$$A_3 F_3 = L_1 \cos \theta_3$$

✓ Les coordonnées de A_1' sont $(0, -(0 F_1 + F_1 A_1 - A_1 A_1'), -L_1 \sin \theta_1)$

($Z_{A_1'}$ doit être négative).

De ce qui précède nous avons :

$$A_1' \left(0, -\frac{L-1}{2\sqrt{3}} - L_1 \cos \theta_1, -L_1 \sin \theta_1 \right)$$

Chapitre 2 Modélisation géométrique de la structure parallèle

✓ Les coordonnées de A_2' sont $(0, A_2' \cos 30, A_2' \sin 30, -L_1 \sin \theta_2)$

Ce qui nous donne :

$$A_2' \left(\left(\frac{L-1}{2\sqrt{3}} + L_1 \cos \theta_2 \right) \cos 30, \left(\frac{L-1}{2\sqrt{3}} + L_1 \cos \theta_2 \right) \sin 30, -L_1 \sin \theta_2 \right)$$

De même pour A_3' :

$$A_3' \left(\left(\frac{L-1}{2\sqrt{3}} + L_1 \cos \theta_3 \right) \cos 30, \left(\frac{L-1}{2\sqrt{3}} + L_1 \cos \theta_3 \right) \sin 30, -L_1 \sin \theta_3 \right)$$

L'équation d'une sphère est donnée par :

$$(x-x_1)^2 + (y-y_1)^2 + (z-z_1)^2 = R^2 \quad (R : \text{rayon de la sphère})$$

En l'appliquant à notre cas, $x_1 = 0$ (donc nous gardons que le plan $y-z$). Donc pour les 3 sphères :

$$\begin{cases} (x)^2 + (y-y_1)^2 + (z-z_1)^2 = L_2^2 \\ (x-x_2)^2 + (y-y_2)^2 + (z-z_2)^2 = L_2^2 \\ (x-x_3)^2 + (y-y_3)^2 + (z-z_3)^2 = L_2^2 \end{cases}$$

$$\Leftrightarrow \begin{cases} (x)^2 + y^2 + z^2 - 2y y_1 - 2z z_1 = L_2^2 - y_1^2 - z_1^2 & (1) \\ (x)^2 + y^2 + z^2 - 2x x_2 - 2y y_2 - 2z z_2 = L_2^2 - x_2^2 - y_2^2 - z_2^2 & (2) \\ (x)^2 + y^2 + z^2 - 2x x_3 - 2y y_3 - 2z z_3 = L_2^2 - x_3^2 - y_3^2 - z_3^2 & (3) \end{cases}$$

De ces équations nous aurons :

$$Y = \frac{1}{y_3 - y_1} \left(\frac{w_3 - w_1}{2} - x x_3 - (z_3 - z_1) z \right)$$

Sachant que : $w_i = x_i^2 + y_i^2 + z_i^2$

De même partant des équations ci-dessus nous obtiendrons x :

$$X = a_1 z + b_1$$

Chapitre 2 Modélisation géométrique de la structure parallèle

Sachant que :

$$a_1 = \frac{(y_2 - y_1)(z_3 - z_1)(y_3 - y_1)}{x_2(y_3 - y_1) - (y_2 - y_1)x_3}$$

$$b_1 = \frac{(w_2 - w_1)(y_3 - y_1) - (w_3 - w_1)(y_2 - y_1)}{2(x_2(y_3 - y_1) - (y_2 - y_1)x_3)}$$

$$\text{Et } d = x_2(y_3 - y_1) - (y_2 - y_1)x_3$$

Pour le calcul de la valeur de Z, nous obtiendrons l'équation suivante :

$$a z^2 + b z + c = 0$$

Sachant que :

$$a = a_1^2 + a_2^2 + 1$$

$$b = 2(a_1 b_1 + a_2(b_2 - y_1) - z_1)$$

$$c = b_1^2 + (b_2 - y_1) - L_2^2 + z_1^2$$

$$a_2 = \frac{(z_2 - z_1)x_3 + (z_3 - z_1)x_2}{d}$$

$$b_2 = \frac{(w_3 - w_1)x_2 + (w_2 - w_1)x_3}{2d}$$

Donc c'est une équation au second degré, ce qui implique que nous obtiendrons deux solutions. En pratique la solution qui devra être considérée est Z négatif.

2.4. Implémentation du modèle géométrique sous Matlab :

Après avoir effectués les calculs des modèles géométriques directes et inverses comme illustré dans le paragraphe précédent, et après obtention des équations désirées, nous les avons implémentés sous le langage Matlab.

Ce programme nous a permis d'effectuer le calcul les coordonnées des positions des articulations de notre robot autrement dit les angles de rotation θ_1 , θ_2 et θ_3 en fournissant la position de l'effecteur et ce dans le cas de la géométrie inverse ; inversement dans le cas de la géométrie directe nous fournirons au programme les angles de rotations pour obtenir la position finale de l'effecteur de notre robot.

Dans ce qui suit nous allons illustrer les résultats obtenus après exécution de notre programme sous Matlab :

Résultat pour les valeurs de $x = 10$, $y = 10$ et $z = -160$.



```
Command Window
File Edit Debug Desktop Window Help
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
>> x=10;
>> y=10;
>> z=-160;
>> [val1, theta1] = inverse (x, y, z, 0)

val1 =

    1

theta1 =

   -62.1161

>> [val2, theta2] = inverse (x, y, z, 120)

val2 =

    1

theta2 =

   -47.1968

>> [val3, theta3] = inverse (x, y, z, -120)

val3 =

    1

theta3 =

   -58.2896
```

```

Command Window
File Edit Debug Desktop Window Help
New to MATLAB? Watch this Video, see Demos, or read Getting Started.
1
theta2 =
-47.1968
>> [val3, theta3] = inverse (x, y, z, -120)
val3 =
1
theta3 =
-58.2896
>> [ x1, y1, z1 ] = direct( theta1, theta2, theta3 )
x1 =
-160.0000
y1 =
10.0000
z1 =
10.0000
fx >>
OVR

```

```
>> x=10;
```

```
>> y=10;
```

```
>> z=-160;
```

```
>> [val1, theta1] = inverse (x, y, z, 0)
```

```
val1 =
```

```
1
```

Chapitre 2 Modélisation géométrique de la structure parallèle

theta1 =

-62.1161

>> [val2, theta2] = inverse (x, y, z, 120)

val2 =

1

theta2 =

-47.1968

>> [val3, theta3] = inverse (x, y, z, -120)

val3 =

1

theta3 =

-58.2896

Chapitre 2 Modélisation géométrique de la structure parallèle

```
>> [ x1, y1, z1 ] = direct( theta1, theta2, theta3 )
```

```
x1 =
```

```
-160.0000
```

```
y1 =
```

```
10.0000
```

```
z1 =
```

```
10.0000
```

```
>>
```

Au finale, vue les résultats obtenu, nous vérifions que les deux modèles nous donnent les mêmes résultats.

En premier après avoir définis les valeurs de x , y , z nous obtenons les angles de rotations θ_1 , θ_2 et θ_3 et pour les même angles retrouvés réintroduites, et après calcul notre programme nous donne les même coordonnées de x , y et z fourni.

Donc, nous concluons que les équations obtenues dans le calcul théorique sont vérifiées et que les deux modèles obtenus sont correcte.

2.5. Conclusion :

Dans ce chapitre, on a commencé tout d'abord par la modélisation du modèle géométrique direct et inverse et la présentation de ces derniers. On a terminés par la détermination des paramètres nécessaires pour calculer les coordonnées opérationnelles donnant la situation de l'organe terminal en fonction des coordonnées articulaires (angles de rotations) et cela pour le modèle géométrique directe, et inversement pour le modèle inverse. Ou nous avons donnés les distance de articulation afin d'obtenir les angles de direction des actionneurs.

Au final, et après implémentation des deux modèles sous Matlab nous avons vérifiés la validité des équations obtenues lors du calcul théorique des modèles direct et inverse.

3.1. Introduction :

Un **robot parallèle** est un mécanisme dont l'architecture lui confère des propriétés remarquables.

Le **Robot Delta** est un robot à structure parallèle, ayant un bras de manipulation formé de 3 parallélogrammes, ce qui, vu sa légèreté, lui permet d'être rapide et de garder sa charge dans la même orientation. Il fait partie de la famille des robots parallèles parfois appelés hexapodes. L'architecture de ce robot a fait justement l'objet de plusieurs études afin de développer sa légèreté, sa précision et sa rapidité.

Dans ce chapitre, nous allons aborder l'architecture de notre robot delta à trois degrés de liberté ainsi que les paramètres de conception utilisés. Aussi le volet programmation du mouvement de ce robot.

3.2. Conception du robot Delta :

Dans notre cas et dans le cadre de ce mémoire, nous avons privilégiés la réalisation de la structure du robot à l'aide de Lego Mindstorms.

Cette technologie nous offre la possibilité de réaliser n'importe quelle structure désirée, vue la variété de sa gamme. Nous avons aussi la possibilité de jouer sur l'aspect mécanique en suivant bien sûr une logique de construction accompagnée du guide fourni spécialement par le fabricant.

Le choix s'est dirigé vers cette structure spéciale, vu que nous n'avions pas trop de choix avec d'autre structure. La structure d'un robot Delta exige beaucoup de précision et des calculs pointus pour atteindre un résultat satisfaisant. Malheureusement, avec ce qui existe sur le marché Algérien nous ne permettait pas d'obtenir la structure voulue.

Donc, à l'aide des pièces Lego assemblée entre elle suivant la logique de réalisation d'un robot Delta nous avons obtenus la structure montrée ci-dessous. Qui représente un robot parallèle Delta à trois degré de liberté.

Cette structure est munie de trois actionneurs, reliés à la brique NXT à l'aide de câbles. Ces actionneurs sont les provocateurs de mouvement de notre robot. Notre cerveau moteur n'est d'autre que la brique intelligente NXT. Le robot ce compose de deux plateforme une est fixe et l'autre est mobile, elles sont reliées entre elles à l'aide de tiges formons les articulations.

Le programme est réalisé sous NXC ou nous avons implantés les modèles de géométrie directe et inverse, ce programme est chargé directement sur le module NXT pour être finalement exécuté par les actionneurs.

Pour notre cas nous avons optés pour que notre robot effectue un mouvement circulaire. Le mouvement démarre de la position initiale du robot il effectue dix tours circulaires pour après s'arrêter et revenir à la position initiale.

Pour illustrer le mouvement du robot nous avons muni l'effecteur d'un stylo et ce pour pouvoir dessiner le mouvement exécuté.

3.3. Architecture du robot :

3.3.1. Description de l'architecture du robot :

Le robot réalisé est un robot d'architecture parallèle à trois degrés de liberté en translation et en rotation.

Son architecture est illustrée dans la figure 3.1.

Il est constitué d'une base fixe et de plate-forme mobile appelée aussi nacelle. Elles sont liées par trois chaînes cinématiques identiques constituées deux barres parallèles formant un parallélogramme. Chaque bras est entraîné par un moteur, solidaire à la base. La nacelle reste toujours parallèle à la base, les mouvements de translation de cette dernière résultent du mouvement combiné des trois actionneurs.

La structure de ce robot est réalisée à l'aide de **Lego Mindstorms NXT**, qui sont des briques intelligente programmable, utilisés pour concevoir des structure légère est modifiable et avec une flexibilité pour la réalisation de n'importe quelle architecture.

La commande du mouvement de cette structure ou de ce robot est faite à base du **NXC** (pas exactement du C) qui est un langage de programmation spécifique au robot NXT.

D'autre part, nous avons implantés les modèles de calcul directe et inverse sous Matlab et ce pour la réalisation des mouvements du robot virtuellement et de pouvoir tester et vérifier notre étude avant manipulation réelle.

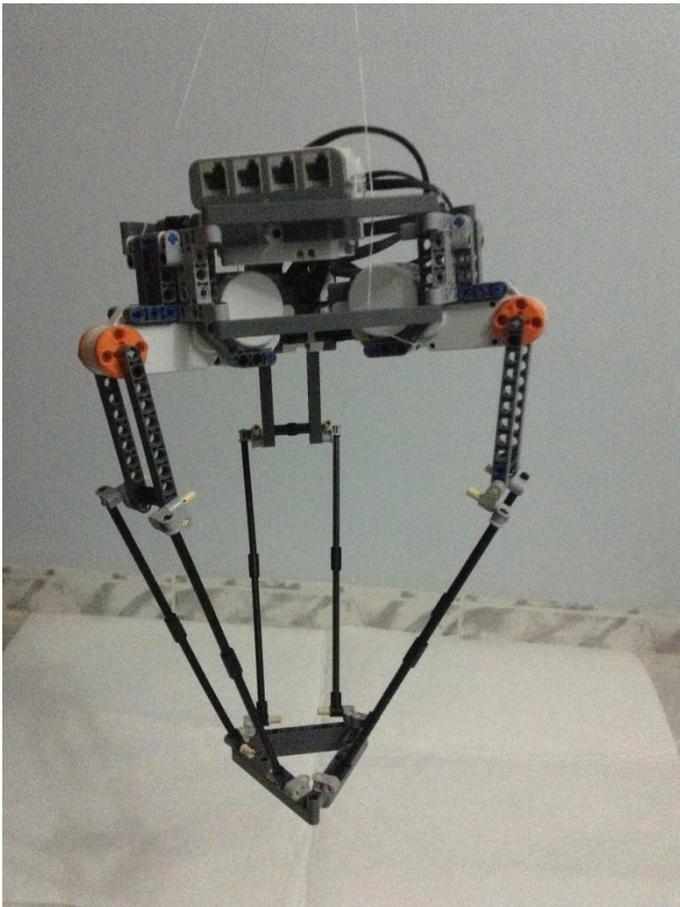


Figure 3.1 Structure du robot Delta Réalisé

3.3.2. Structure mécanique du robot :

a) Présentation des Lego Mindstorms NXT :

Les **Lego Mindstorms NXT** sont un jeu de construction et de robotique présenté par Lego en 2006. Il succède à la gamme Lego Mindstorms RCX. En 2010, Mindstorms NXT en est à sa version 2.0.

La gamme Mindstorms NXT est la dernière génération de Lego Mindstorms qui a été présentée le 4 janvier 2006 lors du Consumer Electronics Show à Las Vegas aux États-Unis. Les principales modifications sont l'usage d'un processeur 32 bit de type ARM ainsi que l'ajout de nouveaux capteurs sonores et ultra-soniques. De plus, il est désormais possible d'utiliser la technologie Bluetooth pour contrôler le robot (avec un téléphone portable par exemple).

Le robot Lego Mindstorms NXT est très utilisé pour l'enseignement et l'éducation des sciences et techniques. De nombreuses compétitions étudiantes sont basées sur le robot Lego comme la First Lego League ou la Coupe de France de robotique. Des compétitions internationales existent également sur la base de ce kit robotique.

Le kit Lego Mindstorms NXT est actuellement en version 2.

Ces lego sont composés de briques, plaques et des barres. Ce sont des composants fondamentaux qui sont utilisés pour construire les châssis, qui transmettent les efforts mécaniques et qui maintiennent fermement les détecteurs en place. Maîtrisez d'abord les fondamentaux de la construction LEGO et alors vous connaîtrez le succès. Ignorez-les, et alors vous passerez plus de temps à réparer votre robot que vous n'en avez passé à le construire.

b) Structure :

Les lego sont constitués de :

1. Briques :

Ceci est une brique de construction LEGO (voir Figure 3.2). Elle n'a subi que peu de changement depuis sa création en 1949. D'après la société LEGO, 320 milliards de briques ont été produites depuis lors. Les briques LEGO sont en plastique ABS. Elles sont fabriquées par injection dans des moules avec des tolérances précises (0.002mm)³. La face supérieure de la brique est couverte de protubérances cylindriques nommées tenons. Le dessous de la brique présente des trous cylindriques ou tubes. Quand vous emboîtez deux briques, les tubes se déforment légèrement autour des tenons, maintenant fermement les briques assemblées.

1.1 Dimensions :

L'habitude est de nommer les briques en donnant leurs dimensions : largeur, longueur et hauteur (bien que la hauteur soit souvent omise en ce qui concerne les briques de taille normale). Les dimensions sont alors données en tenons⁴. La pièce ci-dessous est une brique 2x4.

Les dimensions des briques LEGO sont basées sur le système métrique. Le tenon vaut alors 8mm.

2. Plaques

Pour l'essentiel, les plaques sont des briques aplaties. Elles sont trois fois moins épaisses : 3,2 mm ou 0,4 tenon. Les plaques sont nommées de la même manière que les briques. (Voir Figure 3.3)

Certaines plaques sont transpercées dans l'alignement des tubes. On les appelle des plaques Technique, ou encore, de manière moins obscure, des plaques trouées. La taille de ces trous est compatible avec l'usage des axes et des chevilles de connexions et rend ces plaques techniques très utiles.

3. Barre :

En 1977, la société LEGO créa la gamme Technic⁵, comprenant une série de modèles pour des enfants plus âgés. Les pièces maîtresses de cette gamme sont les barres qui sont des briques 1xn avec des trous latéraux. Les trous sont espacés d'un tenon et centrés entre les tenons de la brique. Les barres peuvent être empilées les unes sur les autres, tout comme des briques. Par ailleurs, des chevilles peuvent être placées dans les trous latéraux permettant ainsi l'assemblage des barres côte à côte. Le nombre de techniques d'assemblage avec ces nouvelles pièces est multiplié. (Voir Figure 3.4)

4. Axes et chevilles :

Le kit RIS6 contient une grande variété de chevilles et d'axes qui permettent d'attacher les barres entre elles. La plus utile est la cheville technique à friction noire ou plus simplement la « cheville à friction ». La cheville à friction est pourvue de petits bourrelets qui, dans les trous des barres, créent un lien serré procurant ainsi un assemblage solide entre barres. Une version longue de cette cheville permet d'assembler trois barres ensemble.

La cheville double est utile avec le bloc connecteur multiforme bleu transparent que l'on trouve dans le RIS 1.5 et 2.0. Elle possède également un trou d'axe qui peut s'avérer utile.

La cheville technique grise est légèrement moins courante. Semblable en apparence à la cheville à friction, elle ne présente pas de bourrelets et la connexion qu'elle procure est faible. C'est donc un choix judicieux pour des pivots ou des charnières. La partie courte de la cheville de longueur $\frac{3}{4}$ correspond parfaitement aux trous présents sur les côtés du RCX, c'est pourquoi on l'utilise couramment pour fixer des barres sur la brique programmable. La partie courte de la cheville de longueur $\frac{1}{2}$ à la même taille qu'un tenon et l'on s'en sert pour les imiter sur les côtés des barres. (Voir Figure 3.5)

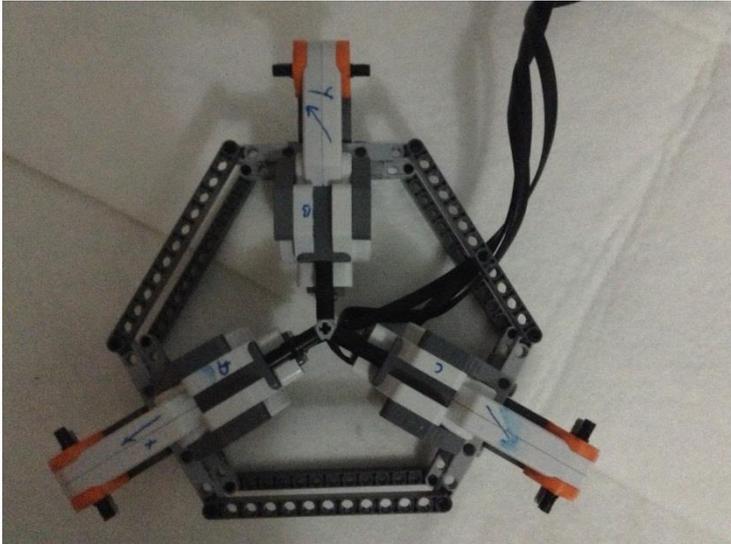
Les axes sont de longues tiges de section cruciforme. Ils coulissent facilement à travers les barres mais sont maintenus fermement dans les trous en forme de croix que l'on trouve sur les roues, les engrenages, les manchons et sur d'autres pièces de la gamme technique. La taille des axes varie de 2 à 12 pour les tailles paires (Voir Figure 3.6)

Il y a quelques pièces inclassables pour lesquelles aucune des catégories axe ou cheville ne convient. La première est le bien nommé axe-cheville qui est mi-axe mi-cheville. Il est le plus souvent utilisé pour fixer des roues dentées sur les barres. Une autre est l'axe 3L avec tenon, qui est un axe technique de taille 3 avec un tenon à l'extrémité. Si vous tentez de placer ce tenon dans le trou d'une barre technique, vous serez surpris de la solidité de la connexion. La cheville technique longue avec manchon est l'utilisée partout, comme pour une liaison par cheville démontable car le manchon offre une bonne prise. Le trou d'axe du manchon permet en outre d'y placer un axe pour créer des montages intéressants. (Voir Figure 3.7)

c) Structure réalisée :

Concernant la structure de notre robot et pratiquement pour la réaliser, nous avons procédé à l'assemblage de certaines pièces décrites dans le paragraphe précédent à savoir : les barres, les axes et les chevilles.

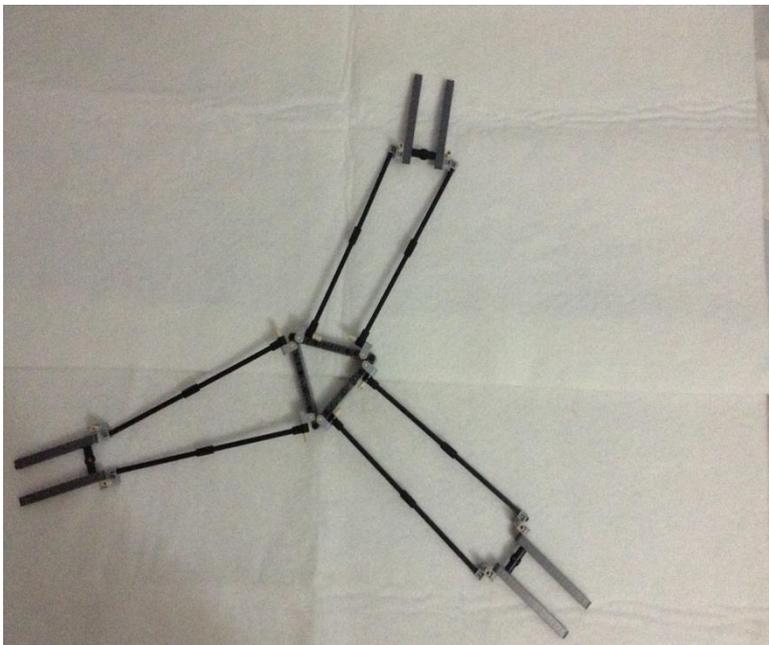
Dans les figures qui suivent nous allons voir les étapes d'assemblages de notre robot (voir Figure 3.8).



Montage des moteurs sur la plateforme fixe



montage des barres et chevilles



Montage des articulations sur la base mobile

Figure 3.8 Montage des modules du robot

d) L'électronique de Lego :

1) La brique intelligente NXT :

Le NXT est le cerveau du votre robot, un petit ordinateur en forme de brique LEGO (Voir Figure 3.9). Au cœur du RCX il y a un microprocesseur 32 bit ARM7 d'Atmel. Il est muni aussi d'une fonction Bluetooth (connexion à d'autres NXT ou à un PC et possibilité de contrôler le NXT avec un téléphone portable ou un autre appareil Bluetooth). Il est de dimensions 112 × 72 × 40 mm; Il se constitue aussi de :

- Un port USB 2.0 (12 Mb/s).
- Quatre ports d'entrée pour la connexion des capteurs nommés 1, 2, 3 et 4.
- trois ports de sortie pour les moteurs nommés A, B et C.
- écran à cristaux liquides 100 × 64 pixels.
- haut-parleur intégré (qualité sonore 8 kHz - 8 bit - échantillonnage 2-16 kHz).
- alimentation : six piles AA (1,5 V) ; une batterie 9 V est commercialisée par Lego.

Il est possible de connecter 3 briques intelligentes NXT en même temps à une autre par Bluetooth, mais la communication ne peut se faire qu'avec une seule à la fois.

Il est possible de connecter le NXT avec les capteurs et moteurs de la gamme RCX grâce à des câbles adaptateurs.

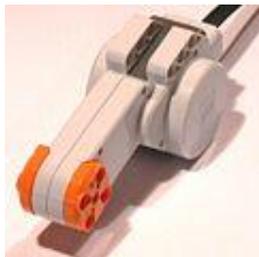


Figure 3.9 Brique intelligente NXT

2) Composants de la brique NXT :

Les principales constituantes de la nouvelle gamme Lego Mindstorms NXT sont (Voir Figure 3.10) :

- La brique intelligente programmable NXT est munie de 4 ports d'entrée et 3 ports de sortie de forme proche du standard RJ12 (seule la patte anti-retour est décalée).
- Des connexions USB et Bluetooth.
- De trois servomoteurs interactifs.
- De Plusieurs dizaines de modèles de capteurs (quatre capteurs par boîte) : ultrason, son, lumière, contact, couleur, autodirecteur infrarouge, gyroskopique, infrarouge, accéléromètre, boussole...
- Du NXT-G : le logiciel de programmation graphique, très intuitif basé sur Labview compatible avec Windows et Mac.
- De très nombreux accessoires supplémentaires (non fournis dans la boîte principale) comme une clé Bluetooth USB, une batterie rechargeable, des moyens de communication Wi-Fi, XBee, des multiplexeurs, des panneaux solaires, des pilotes de relais, des roues holonomiques...



moteur



Prise femelle



Prise mâle

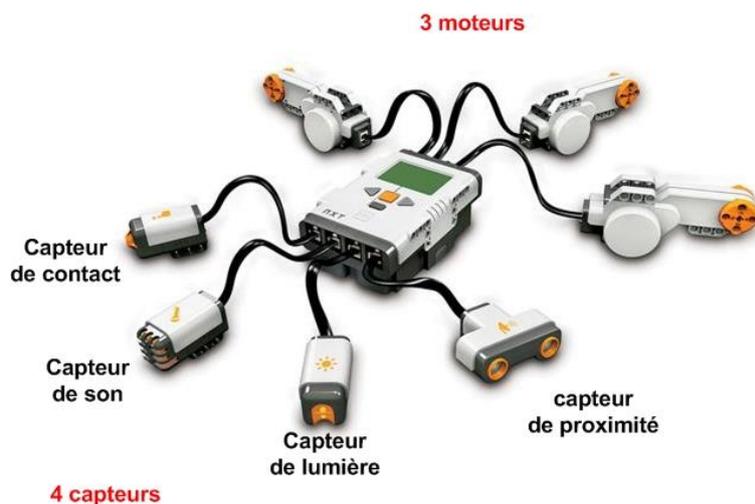


Figure 3.10 Composantes du module NXT

3) Les moteurs NXT :

Pour la réalisation de notre robot nous avons besoins de trois moteurs, tous identiques, ces moteurs sont commercialisés par le fabriquant de Lego Mindstorms et sont spécialement conçus pour la réalisation des robots XT.

Ces moteurs sont des moteurs à courant continu, ils permettent au robot de se déplacer aisément et avec précision.

Chaque moteur est équipé d'un capteur de rotation intégré qui permet au programmeur de contrôler précisément le nombre de rotation de la roue, à 2 degrés près.

Les moteurs pour Lego Mindstorms NXT sont pilotables par la brique intelligente Lego Mindstorms NXT et sont programmables. Le programme réalisé nous permet la maîtrise avec précision du nombre de tours ainsi que la vitesse. Cette fonctionnalité intégrée nous permet aussi d'estimer avec une grande précision les déplacements du robot, du bras articulé ou de toute autre architecture que nous concevrons.

L'image qui suite illustre un moteur NXT aussi nous pouvons voir un l'intérieur du moteur qui est décortiqué. (Voir Figure 3.11).

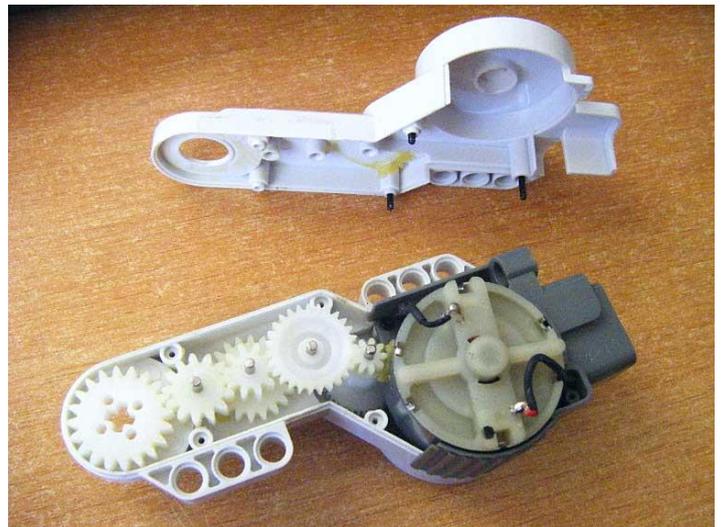


Figure 3.11 Moteur NXT et intérieur décortiqué

3.1) Le principe de fonctionnement des moteurs NXT :

Comme déjà développé dans le paragraphe précédent, les moteurs NXT sont des moteurs à courant continu, et dans ce qui suit nous allons exposer le principe de fonctionnement de ces derniers :

Le moteur à courant continu se compose :

- **de l'inducteur ou du stator** : partie fixe, constitué par un aimant ou un électroaimant appelé également inducteur qui crée un champ magnétique dirigé vers l'axe du rotor.
- **de l'induit ou du rotor** : partie mobile, appelé également induit, constitué par un cylindre d'acier doux à la périphérie duquel sont disposés des conducteurs reliés aux lames du collecteur sur lesquelles frottent deux charbons.
- **du collecteur et des balais** : assurent la liaison avec les bornes du moteur.

Lorsque le bobinage d'un inducteur de moteur est alimenté par un courant continu, sur le même principe qu'un moteur à aimant permanent (comme illustré dans la Figure 3.12), il crée un champ magnétique (flux d'excitation) de direction Nord-Sud.

Une spire capable de tourner sur un axe de rotation est placée dans le champ magnétique. De plus, les deux conducteurs formant la spire sont chacun raccordés électriquement à un demi collecteur et alimentés en courant continu via deux balais frotteurs.

D'après la loi de Laplace (tout conducteur parcouru par un courant et placé dans un champ magnétique est soumis à une force), les conducteurs de l'induit placés de part et d'autre de l'axe des balais (ligne neutre) sont soumis à des forces F égales mais de sens opposé en créant un couple moteur : l'induit se met à tourner !

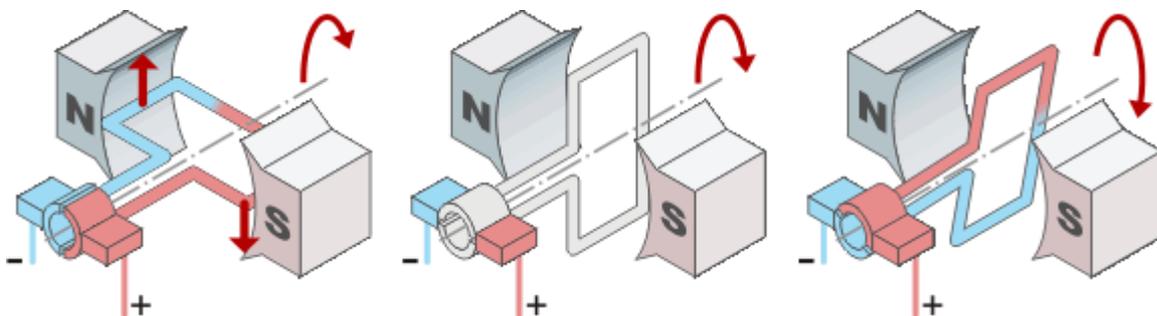


Figure 3.12 Fonctionnement d'un moteur CC

Si le système balais-collecteurs n'était pas présent (simple spire alimentée en courant continu), la spire s'arrêterait de tourner en position verticale sur un axe appelé communément "ligne neutre". Le système balais-collecteurs a pour rôle de faire commuter le sens du courant dans les deux conducteurs au passage de la ligne neutre. Le courant étant inversé, les forces motrices sur les conducteurs le sont aussi permettant ainsi de poursuivre la rotation de la spire.

Dans la pratique, la spire est remplacée par un induit (rotor) de conception très complexe sur lequel sont montés des enroulements (composés d'un grand nombre de spires) raccordés à un collecteur "calé" en bout d'arbre. Dans cette configuration, l'induit peut être considéré comme un seul et même enroulement semblable à une spire unique. (Voir Figure 3.13).

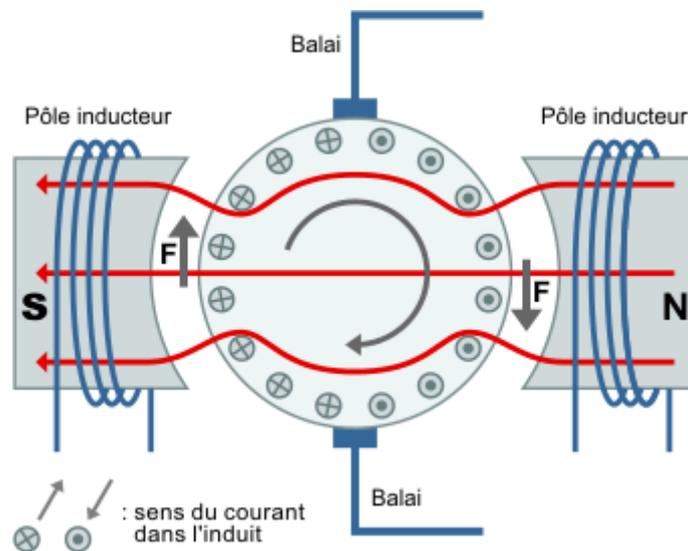


Figure 3.13 Schéma de l'intérieur d'un moteur CC

3.2) Les caractéristiques du moteur courant continu :

Le moteur courant continu (CC) est caractérisé par une constante de vitesse, et une pente vitesse/couple. Le courant est proportionnel à la charge ; et la vitesse est proportionnelle à la tension d'alimentation.

Le moteur courant continu sans fer ne présente pas de couple magnétique résiduel et les agitations électromagnétiques sont dérisoires. Son rendement, qui atteint 90%, surpasse celui des autres technologies de moteurs.

Son rotor en forme de cloche donne la possibilité d'accélérations très importantes et d'un couple de retenue inexistant.

Cela permet d'obtenir des positionnements précis et des vitesses faibles.

Le rotor est traditionnellement composé d'un stator à aimant permanent en terre rare de type Al-nico, ce qui explique les caractéristiques dynamiques très élevées.

3.3) Le fonctionnement des moteurs NXT :

Comme déjà vu dans le paragraphe précédent pour les moteurs à courant continu, le principe de fonctionnement de ces derniers s'applique sur les moteurs NXT excepté que dans les moteurs NXT le moteur continu est sans fer ce qui implique un mouvement à flux continu. Aussi les moteurs NXT sont équipés d'un capteur de rotation intégré qui permet de mesurer les rotations en degrés avec une précision de ± 1 degré. Ce capteur délivre une donnée numérique. Les moteurs NXT sont aussi équipés de réducteurs de vitesse ou d'un système à pignons qui représente la fonction PID du moteur, donc les moteurs NXT sont des moteurs asservis.

Dans la figure suivante (Figure 3.14) nous allons voir les composants d'un moteur NXT :

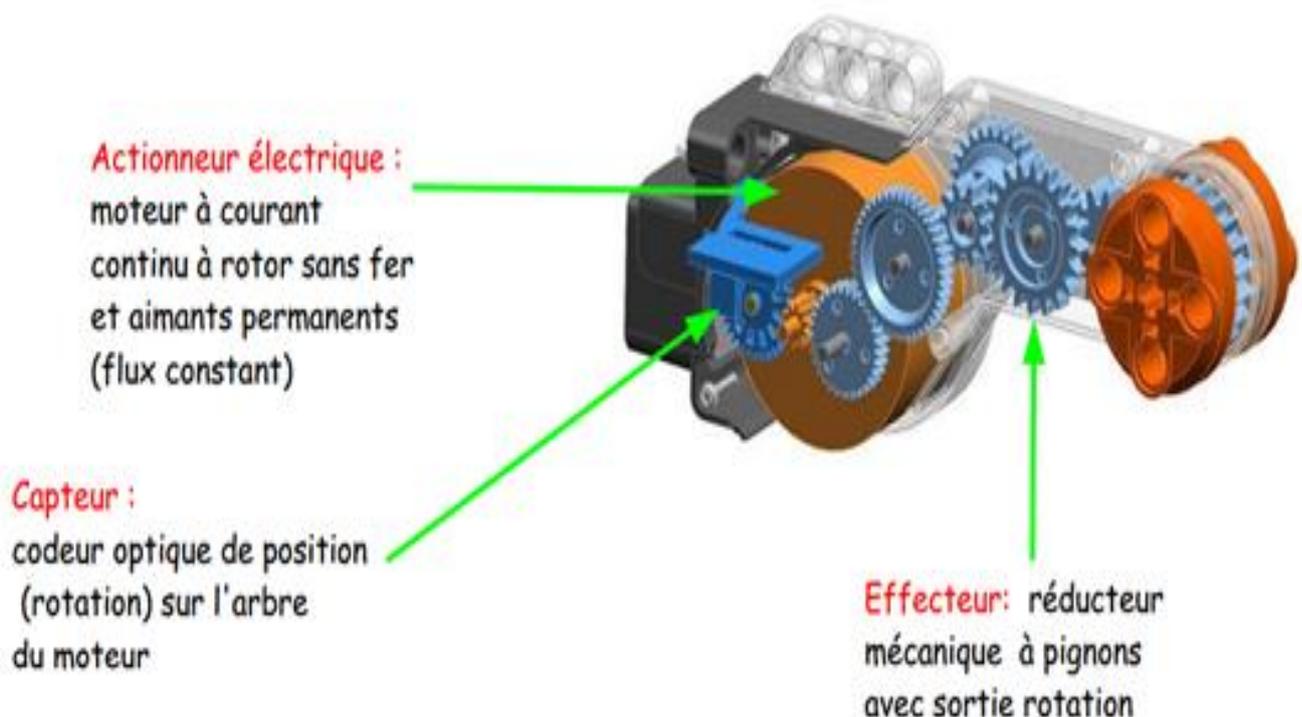


Figure 3.14 composants d'un moteur NXT

Aussi dans la figure suivante (Figure 3.15) nous verrons les capteurs d'un moteur NXT et les signaux engendrés:

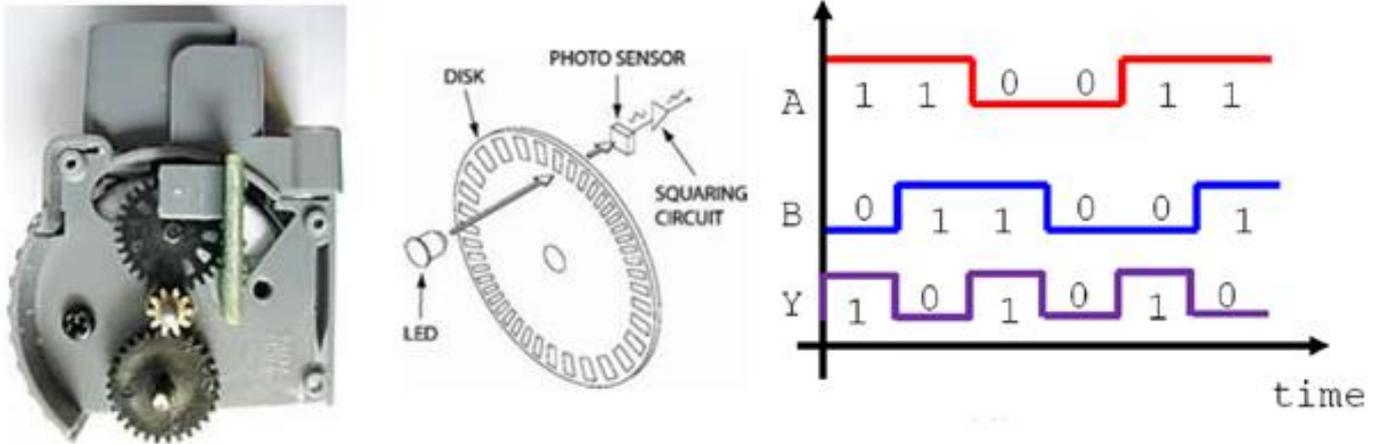


Figure 3.15 Capteur d'un moteur NXT et ses signaux

3.4) Montages des moteurs NXT :

Pratiquement pour la réalisation de notre robot nous avons utilisés 3 moteurs de types NXT, que nous avons monté sur la structure obtenue à l'aide de lego.

Dans les figures qui suivent nous allons voir l'emplacement et le montage de nos moteurs (voir Figure 3.16).



Figure 3.16 Montage des moteur NXT

e) La programmation :

Il existe de nombreuses possibilités pour programmer le Mindstorms NXT : Dans le kit grand public, un logiciel de programmation graphique est fourni. Il se nomme NXT-G. Il est basé sur Labview mais ne nécessite aucune compétence en LabView. Il est très facile de démarrer à l'aide de NXT-G.

Le logiciel NXT-G n'est pas fourni dans la version éducation du kit Lego.

Les types de programme existants sont les suivants :

- **Programmation textuelle :**
 - Langages .NET, tels C sharp ou Visual Basic .NET, grâce à Microsoft Robotics Studio.
 - Le NBC, un langage assembleur.
 - Le NXC, qui est un langage proche du C. Il est gratuit et open-source.
 - RobotC est un autre langage de programmation basé sur C, développé par l'université Carnegie Mellon aux États-Unis.
 - Lejos est une API open-source basée sur le langage Java.
 - Urbiscript : langage de la plate-forme logicielle Urbi (*Universal Real-Time Behavior Interface*).
 - En Ada (complet ou profil Ravenscar).
 - Une bibliothèque ROS existe également pour le robot Lego.
 - Une bibliothèque Matlab créée par l'université d'Aix-la-Chapelle¹ existe aussi
- **Programmation graphique :**
 - NXT-G, la version Lego de LabVIEW qui permet une prise en main aisée
 - Microsoft Robotics Studio.
 - DialogOS permet de commander les robots par la voix.
 - Robolab est une programmation possible du groupe National Instrument.

1) Le langage NXC :

Dans notre cas nous avons utilisés le **NXC** (pas exactement du C ou **Not eXactly C**) qui est un langage de programmation spécifique au robot NXT basé sur le NBC, c'est un langage assembleur.

La première version du NXC date de 2007, année de libération de la norme NXT sous licence publique Mozilla. Son développeur, John Hansen, avait déjà mis au point le NQC pour le RCX ; il développe également l'IDE BricxCC.

Jusqu'à la version 1.26 du micro-programme, l'utilisation de nombres à virgule flottante était impossible sur le NXT, car non utilisés pour la commande des moteurs et l'acquisition des mesures de capteurs. Depuis cette version, cela est aussi possible avec le NXC.

2) Exemples de programmes NXC :

Voici deux exemples de programmes réalisables avec le NXC, suivis de l'explication de leur effet.

Allumage d'un moteur :

```
task main()
{
  OnFwd(OUT_A, 75);
  Wait(1000);
}
```

Ici l'actionneur relié à la prise « A » du NXT démarre puis le programme s'arrête au bout d'une seconde.

Réaction à la lumière :

```
task main()
{
  SetSensorTouch(S1);
  SetSensorLight(S2);
  OnFwd(OUT_A);
  while(SENSOR_1 == FALSE)
  {
    if(SENSOR_2 > 50)
    {
      PlayTone(440, 10);
    }
  }
}
```

L'actionneur relié à la prise « A » du NXT est démarré et le type des capteurs 1 et 2 est défini ; si l'on met le capteur 2 (capteur de luminosité) devant une source de lumière, le NXT produit une tonalité (de fréquence 440 Hz) durant 10 millisecondes. L'appui sur le capteur 1 (capteur de contact) provoque l'arrêt du programme.

3) Environnement du langage NXC :

Dans la figure suivante nous illustrons l'espace de travail ou l'environnement du langage NXC utilisé. (voir Figure 3.17).

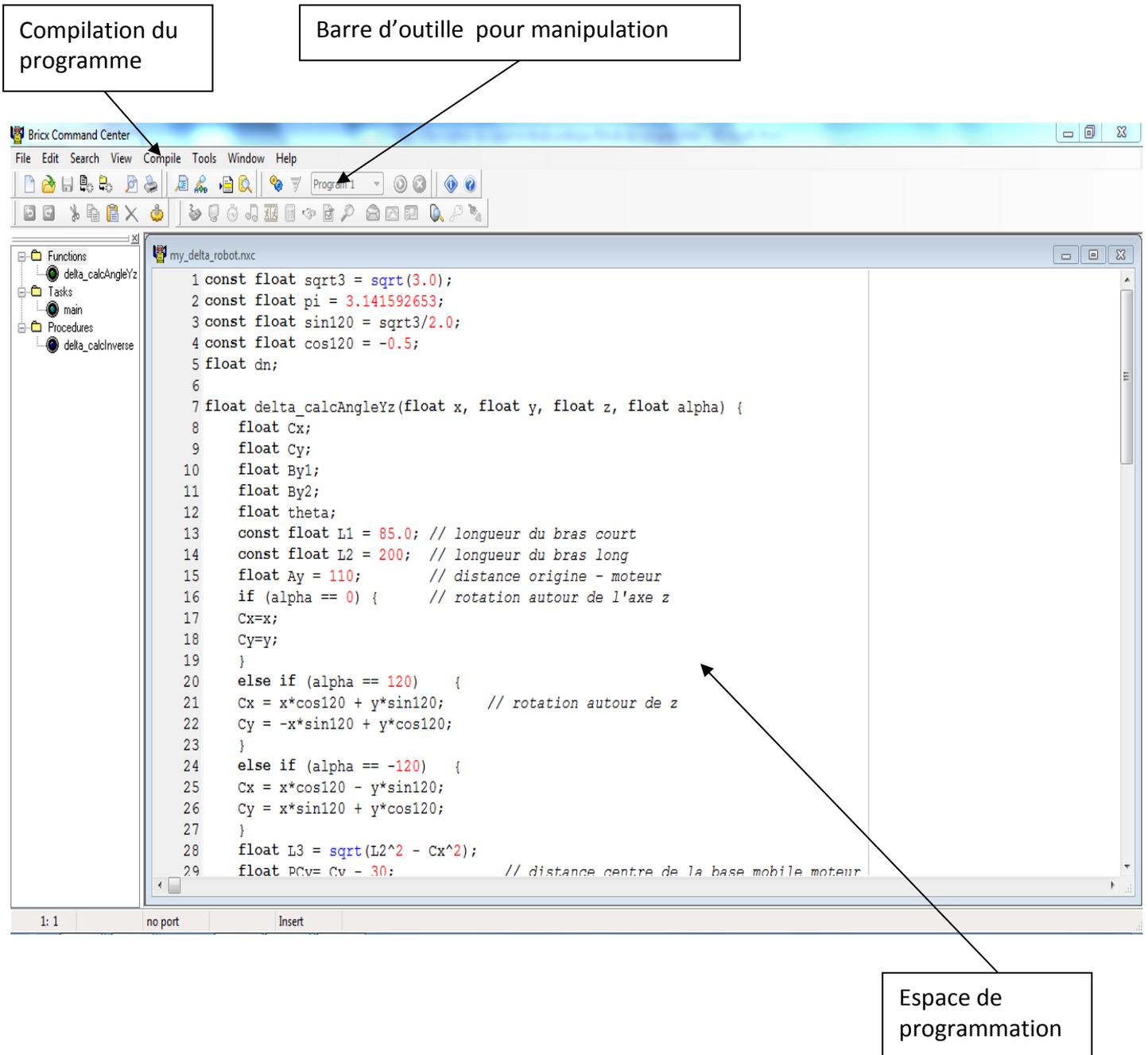


Figure 3.17 Environnement du Langage NXC

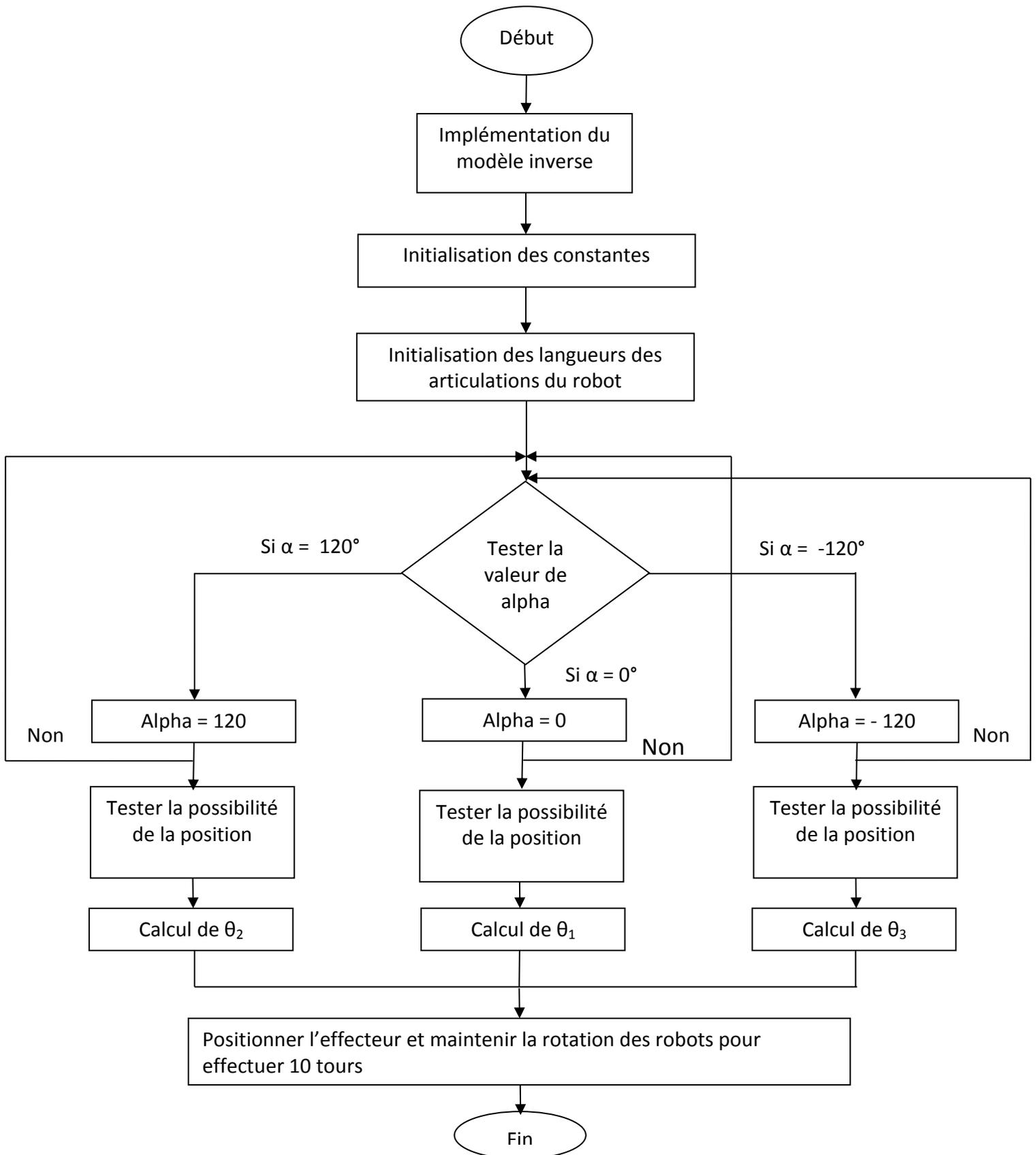


Figure 3.18 Organigramme du programme

Dans la figure précédente (Figure 3.18) nous avons illustrés l'organigramme de notre programme.

4) Fonctions utilisées sous le langage NXC :

Dans ce qui suit nous allons voir quelques exemples des fonctions et instructions utilisées dans notre programme sous NXC.

- ✓ **L'instruction Float** : qui nous permet d'initialiser les constantes et variables.

Exemples :

```
const float sqrt3 = sqrt(3.0);
```

```
const float pi = 3.141592653;
```

```
float Cx;
```

```
float Cy;
```

```
float By1;
```

- ✓ **L'instruction PosRegEnable** : qui nous permet de maintenir les actionneurs dans la position désiré tant que la condition donnée est vérifiée.

Exemples :

```
SetMotorRegulationTime(MS_10); // temps de reponse apres regulation PID
```

```
PosRegEnable(OUT_A);
```

```
PosRegEnable(OUT_B);
```

```
PosRegEnable(OUT_C);
```

3.4. Conclusion :

Dans ce chapitre nous avons abordés la structure mécanique de notre robot qui est constituée de module de lego assemblé entre elle. Nous avons aussi abordés l'aspect ou le volet programmation sous le langage NXC.

D'autre part, l'utilisation des lego nous offre des facilitations et l'opportunité d'avoir n'importe quelle structure qu'on désire à condition de suivre une logique de réalisation ce qui n'empêche pas des erreurs et des difficultés liés à l'assemblage et au choix des modules.

Pour ce qui est de la tache exécutée par le robot, nous avons une multitude de possibilités de choix de mouvement. Il suffit de munir l'effecteur d'une pièce spécifique à la tache désire (par exemples : une pince, une caméra, des pinces chirurgicaux, tête pour soudure, lampe...), adapter le programme à cette tâche et le faire exécuter par les actionneurs.

Pour notre cas, nous avons choisi de doter notre effecteur de stylo pour pouvoir dessiner un mouvement circulaire. C mouvement consiste à faire dix cercles.

Au finale, c'est vrais que la structure en lego nous donne des facilitations concernant la conception du châssis, mais qui n'exclut pas des problèmes lors de l'exécution du programme et la réalisation du mouvement. Ces problèmes se présentent dans la fixation de l'effecteur et la régularité du mouvement des articulations et qui sont liés à l'instabilité et la fragilité de la structure au niveau des coudes.

4.1. Introduction :

Après une étude des différentes composantes du système dans les chapitres précédents, nous allons consacrer ce chapitre aux tests et résultats que nous avons obtenu lors de nos expériences.

4.2. Réalisation de la structure du robot :

Comme déjà exposé dans le chapitre précédent, la structure de notre robot est réalisée à l'aide de module de lego Mindstorms, qui sont des pièces à assembler suivant le guide de construction en se basant sur une logique et les modèles de structure de robot parallèle Delta. Cette structure est constituée d'une base fixe (plateforme fixe) et d'une plateforme mobile reliée entre elles par des articulations (bras), à la base fixe nous avons fixés trois moteurs ou actionneurs qui ont pour taches d'exécuter le mouvement. Ces actionneurs sont liés à l'interface NXT ou nous chargeons notre programme téléchargé à partir d'un PC.

Le NXT est une brique LEGO intelligente, contrôlée par ordinateur, et qui constitue le cerveau du robot. Elle a une batterie autonome rechargeable, munie de connexions pour moteurs et capteurs. Le programme mis au point sous le langage NXC est téléchargé directement sur cet interface pour être exécuté finalement par les actionneurs.

4.3. Programmation :

La programmation est réalisée sous langage NXC qui est un langage spécifique à la programmation des robots lego. Ce langage fourni plusieurs fonctions pour les utiliser afin de réaliser notre objectif qui est le mouvement désiré.

C'est un langage assembleur (pas exactement C). Le robot NXT possède un interpréteur de code binaire qui est utilisé pour exécuté les programmes. Le compilateur NXC traduit un programme source écrit en NXC en un code binaire NXT et celui-ci peut être exécuté sur la brique.

D'autre part pour l'implémentation des modelés géométrique direct et inverse, nous avons utilisés le langage Matlab.

En premier, pour effectuer le calcul des paramètres désirés, soit les angles θ_i pour le modèle inverse, et la position pour le cas du modèle direct.

Nous présentons dans la figure suivante (Figure 4.1) l'organigramme sous Matlab des deux modèles direct et inverse :

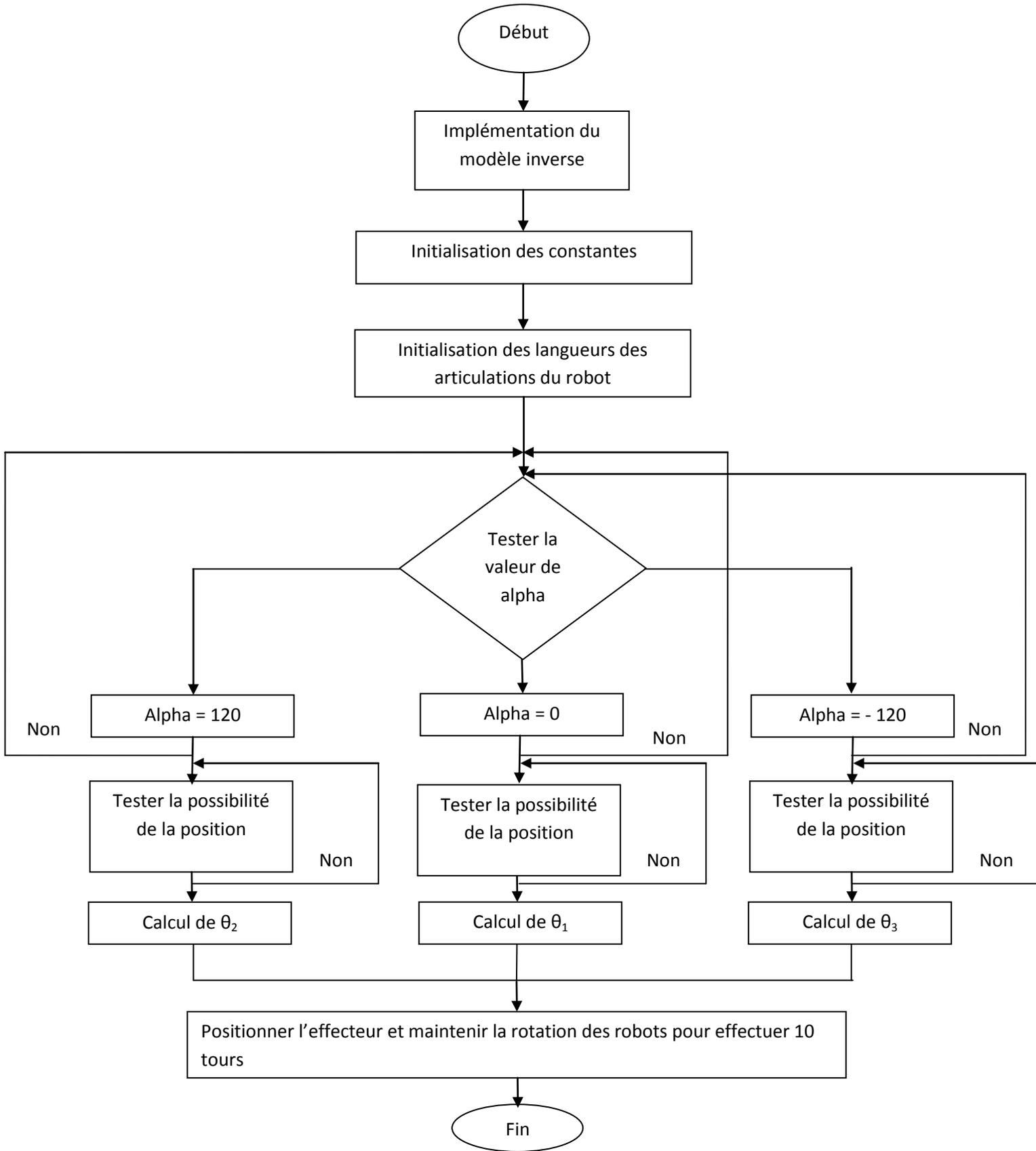


Figure 4.1 Organigramme du programme

a) Tests et simulation sous Matlab :

Dans ce qui suit nous allons voir quelques résultats de simulation sous Matlab du mouvement du robot aussi le mouvement réel de notre robot pour ces même valeur obtenues :

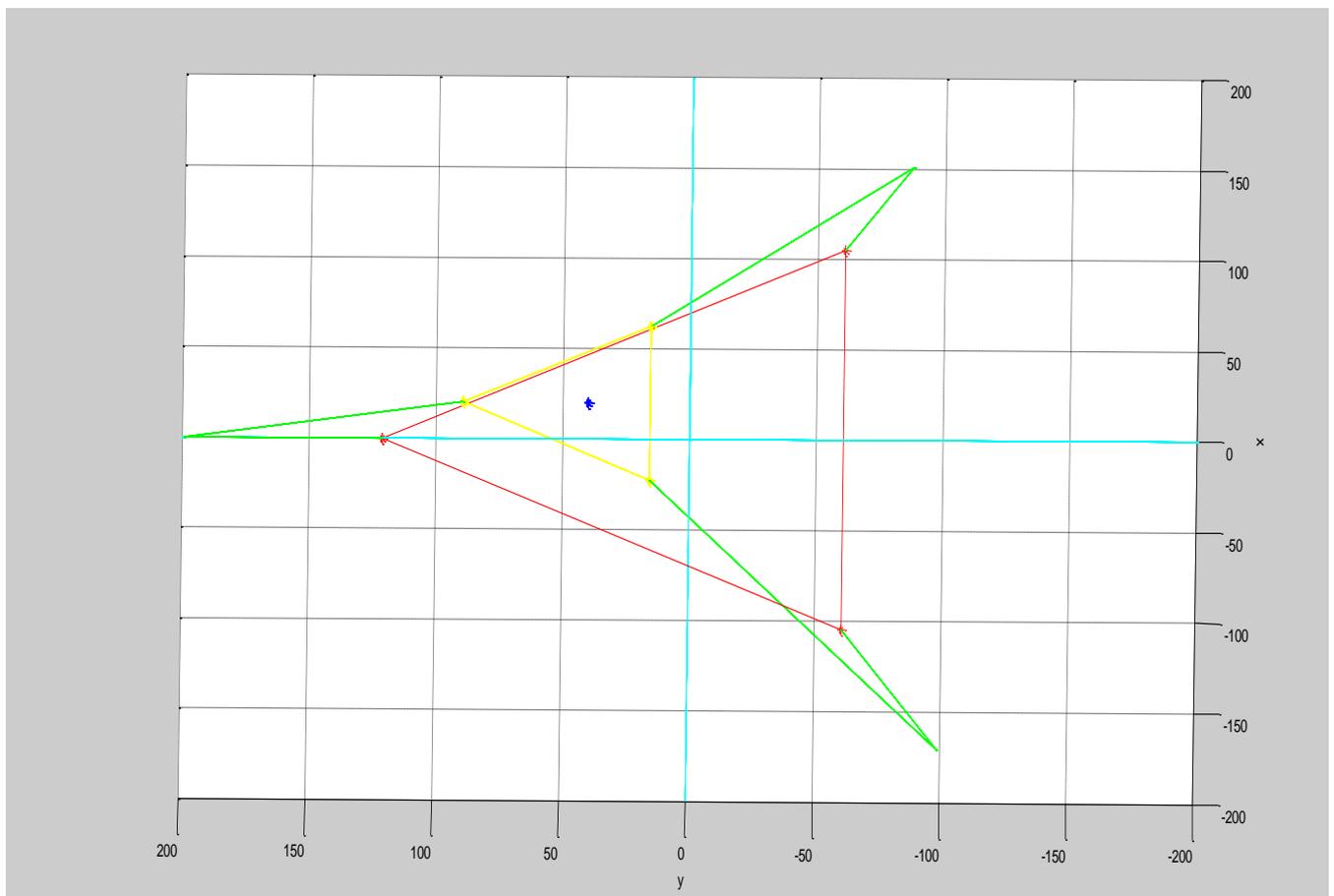
Mouvement pour $x=20$ $y=40$:1- **Résultat sous Matlab :** voir Figure 4.2

Figure 4.2 Résultat sou Matlab **Mouvement pour $x=20$ $y=40$:**

Dans cette figure le triangle rouge représente la plateforme fixe et le jaune représente la plateforme mobile pour les segments verts ils représentent les articulations de notre robot. Au finale le point bleu c'est le mouvement de notre effecteur.

2- Résultat réel : voir Figure 4.3



Figure 4.3 Résultat réel Mouvement pour $x=20$ $y=40$:

Mouvement pour $x = 30$ $y = 0$:

1- Résultat sous Matlab : voir Figure 4.4

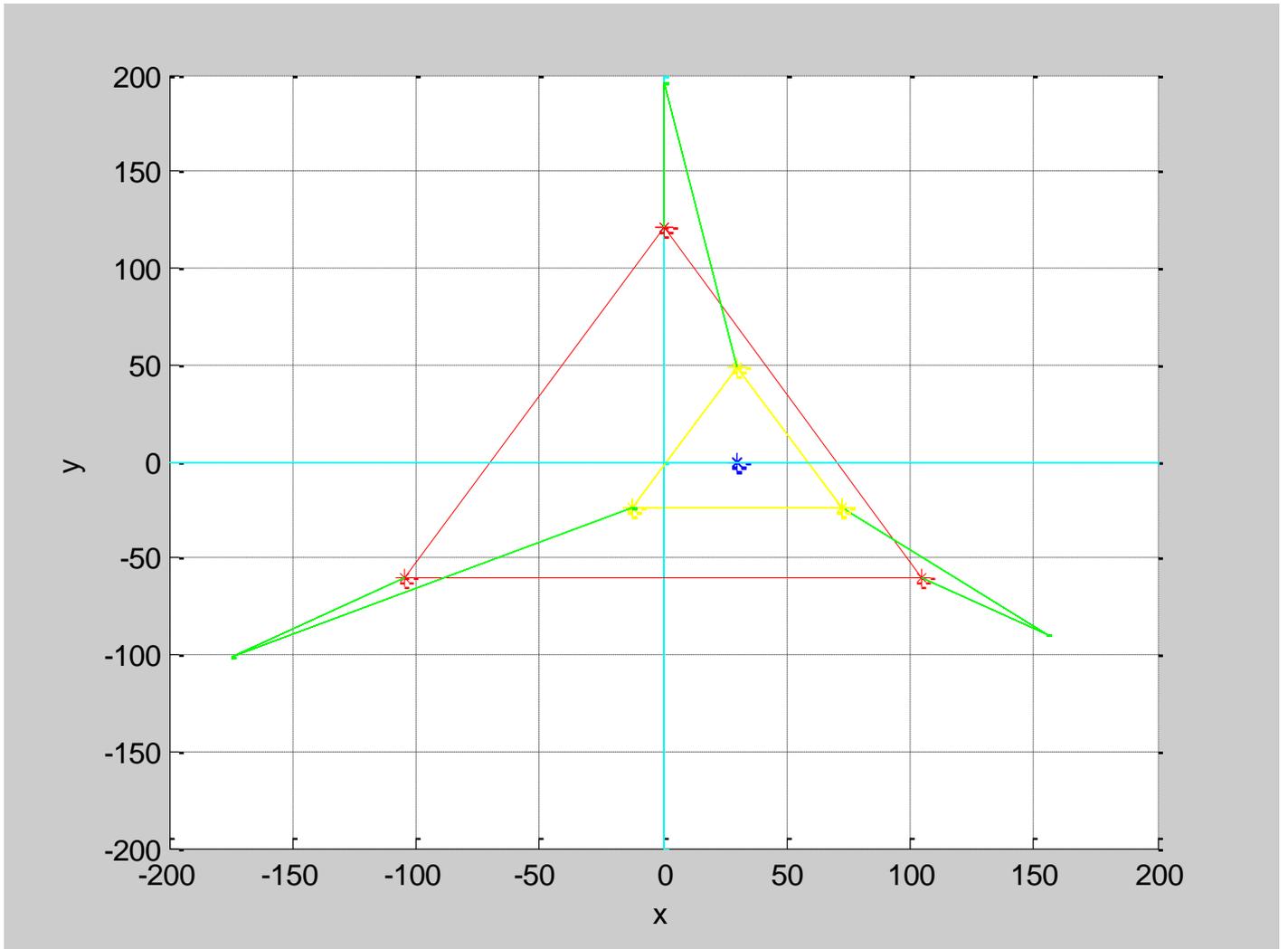


Figure 4.4 Résultat sou Matlab Mouvement pour $x = 30$ $y = 0$:

2- Résultat réel : voir Figure 4.5



Figure 4.5 Résultat réel Mouvement pour $x = 30$ $y = 0$:

2- Résultat réel : voir Figure 4.7

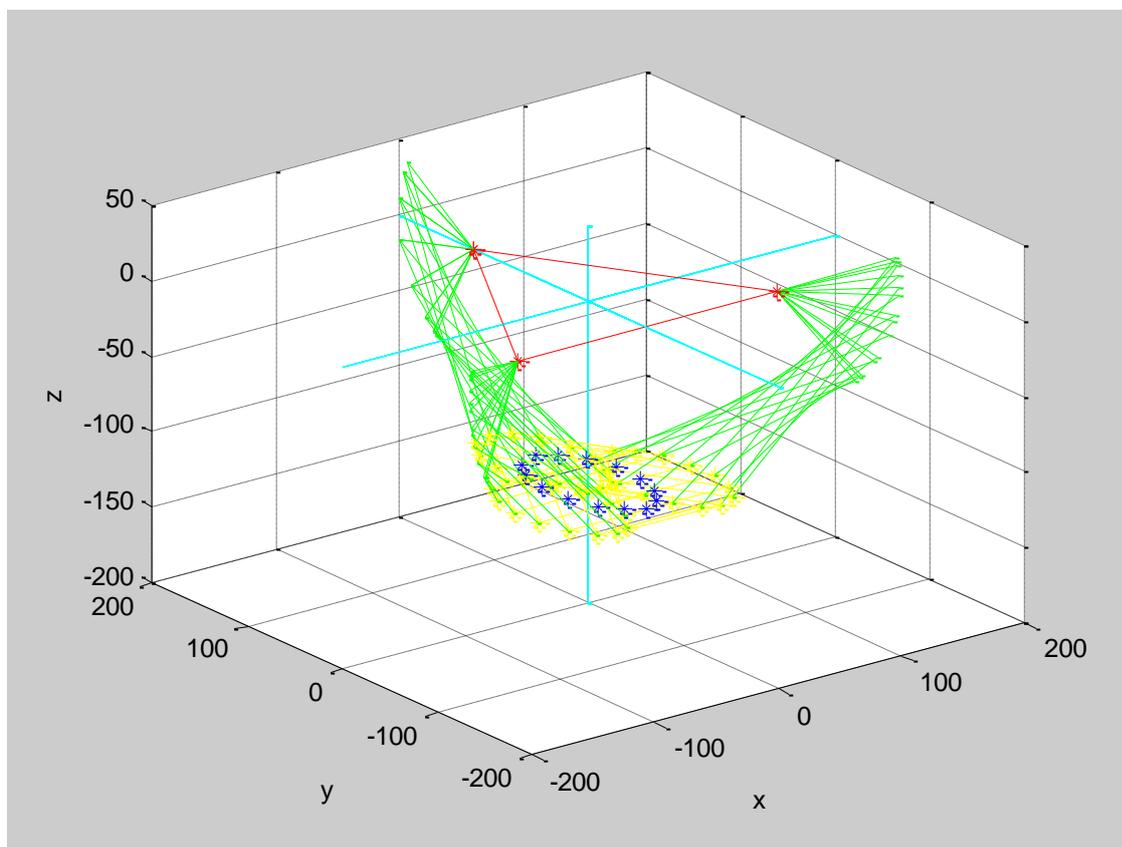


Figure 4.7 Résultat réel Mouvement pour $x = 0$ $y = 60$:

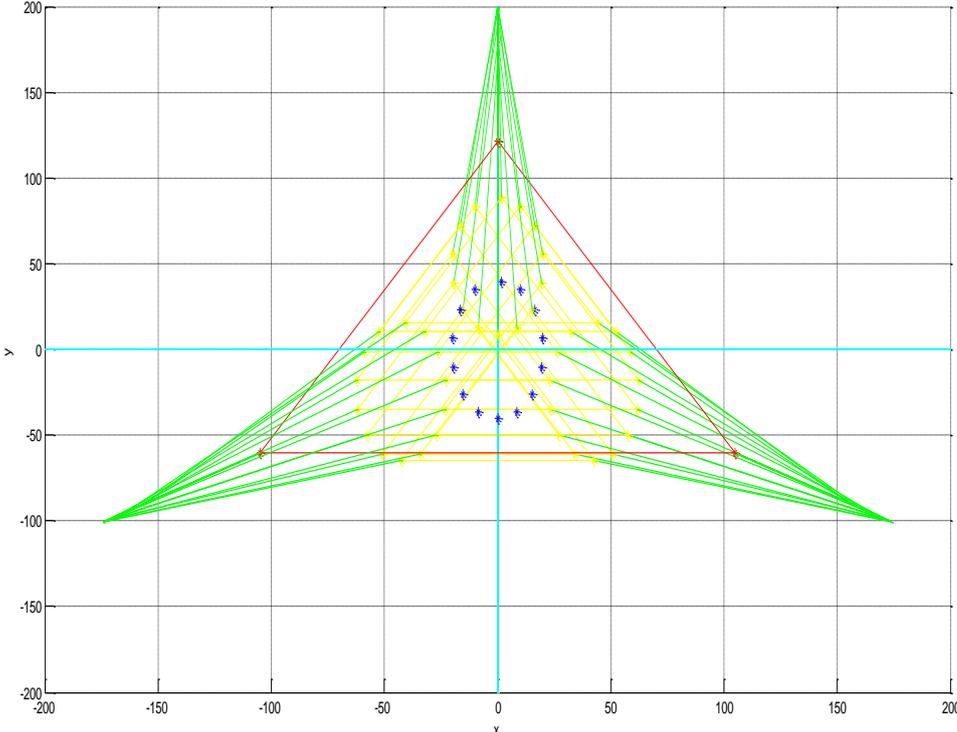
Des résultats précédents nous remarquons que les mouvements virtuels sous Matlab du robot sont identiques aux résultats obtenus réellement et ce après chargement du programme NXC dans l'interface NXT et son exécution.

a-1) Test du mouvement du robot sous Matlab :

Pour cette partie nous allons voir la simulation du mouvement de notre robot et ce sous Matlab, et dans la figure 4.8 nous pouvons voir quelques profil et vue sous différents angles de ce mouvement.



Profil A



Profil B

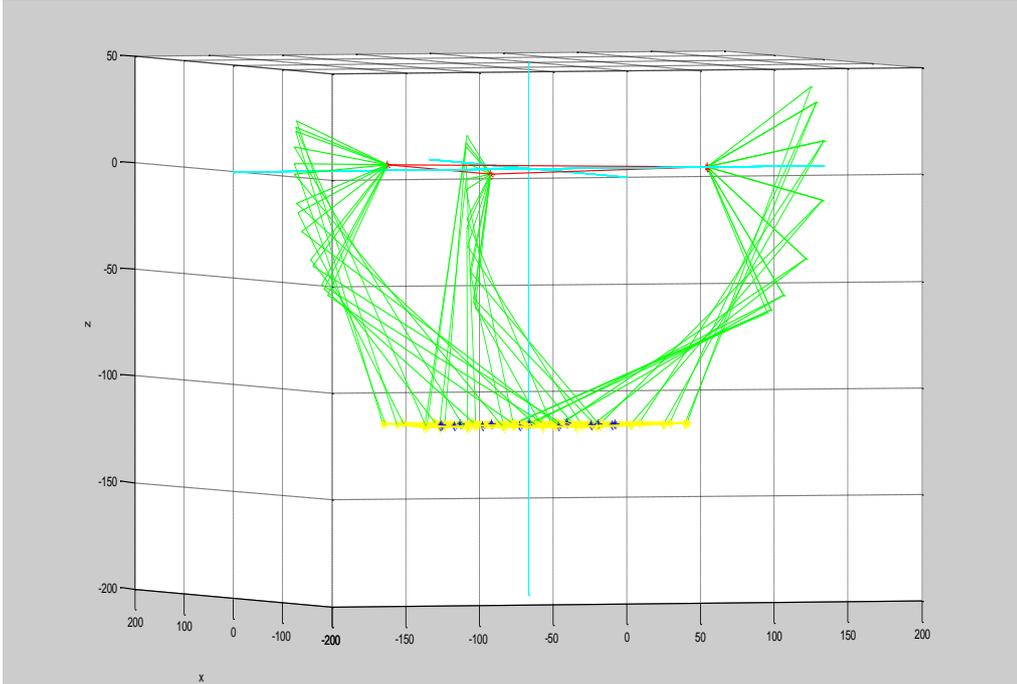


Figure 4.8 Simulation du mouvement du robot sous matlab

4.4. Conclusion :

Dans ce chapitre nous avons commencé par la présentation de la structure de notre robot. Au finale la structure obtenue est satisfaisante et a répondu au besoins désirés . néanmoins cette structure présente quelques points faibles, notamment au niveau des articulation et coudes ou nous avons constatés que ces derniers ne sont pas trop solide et ce qui engendre un léger decalage dans le mouvement de nos articulations.

Pour ce qui est de la partie programmation les résultats sont atisfaisants, en comparant les résultats de la simulation sous Matlab et les mouvements réels de notre robot, nous constatons l'exactitude de l'action.

Introduction générale

Le développement dans le domaine de la robotique a été poussé en premier lieu par une motivation d'imitation des capacités basiques de l'être humain, afin de lui faciliter les tâches difficiles et répétitives dans le domaine de l'industrie, telles que le soulèvement de pièces lourdes, serrage, découpage, assemblage, etc. Les premiers robots furent alors des robots à structure **série simple**. Ces robots ont pour avantage de disposer d'un grand volume de travail et d'être relativement simples sur le plan des calculs liés à leur commande. Par contre, ces robots présentent des inconvénients quand il s'agit d'effectuer des tâches demandant de la précision ou des tâches devant être exécutées avec rapidité.

Cette catégorie de robots a eu et a encore de nos jours du succès dans le domaine industriel, mais le progrès atteint par l'assistance de ces robots a fait que les ambitions des chercheurs se sont dirigées vers la création de structures capables d'effectuer des tâches de plus en plus complexes et variées demandant de meilleures performances, d'où vient l'idée des **robots parallèles**.

Les architectures parallèles sont apparues après la dernière guerre avec les premiers simulateurs de vol. Elles sont utilisées depuis dans d'autres applications nécessitant la manipulation d'objets lourds avec de grandes accélérations ou l'assemblage dans les applications qui requièrent une grande précision et la maîtrise des efforts de contact.

Cependant, pour leur utilisation pratique, les robots parallèles souffrent souvent de leur volume de travail réduit, comparé à celui des robots sériels. Une autre catégorie de robots dits hybrides a été proposée ; La structure hybride consiste à disposer en série des modules de structure parallèle. Les robots parallèles et hybrides sont des structures articulées complexes souvent difficiles à modéliser. On peut dire que le rôle des structures hybrides c'est de combiner les avantages des deux structures parallèles et sérielles.

Notre travail consiste à réaliser un robot parallèle delta à trois degrés de liberté en suivant des procédures et des techniques de calcul habituellement utilisées pour les robots parallèles, comportant des boucles fermées.

Ainsi que, les méthodes fondées sur la géométrie inverse et directe, nous permettons de déterminer les angles de rotations à fournir aux moteurs pour pouvoir positionner les articulations ainsi que l'extrémité du robot (effecteur). La structure de notre robot est réalisée à l'aide des **Lego Mindstorms**, qui sont des modules lego assemblées entre elles, ces modules sont spécialement conçus pour la réalisation des châssis de robots.

La programmation est réalisée à l'aide du langage **NXC (Not eXactly C)**, qui est un langage de programmation spécifique au robot **NXT**, basé sur le **NBC (Next Byte Codes** ou **NBC** est un langage assembleur ou compilateur).

Introduction générale

Outre l'introduction et la conclusion générales, ce mémoire est organisé en quatre chapitres répartis comme suit :

- **Le premier chapitre** est consacré à l'état d'art des robots parallèles, et quelques notions de bases, ainsi que quelques types des robots parallèles.
- **Le deuxième chapitre** traite l'étude théorique, ou l'étude géométrique de la structure parallèle.
- **Le troisième chapitre** est consacré à la partie de réalisation du robot et aux techniques liées à la programmation utilisée ainsi que l'étude pratique.
- **Le quatrième chapitre** aborde les résultats obtenus, aussi les vérifications et testes réalisées.

Bibliographie

- [1] **Guide de programmation NXC**, BELLINI Quentin, GNANAKULENTHIRAN Anitha, GOVINDEN Johana, MEZINE Ahcene, TIMZOUERT Chabane, 2012.
- [2] **AMOURI AMMAR**, << MODELISATION DYNAMIQUE D'UN ROBOT PARALLELE FORME DE PLUSIEURS MODULES EMPILES>>, UNIVERSITE L'ARBI BEN M'HIDI D'OUM EL BOUAGHI, Juin 2011.
- [3] NXC Programmer's Guide, Version 1.2.1 r5 , Generated by Doxygen 1.6.2, Oct 17 2011.
- [4] <http://www.mecademic.com/fr/Robot-parallele.html>
- [5] http://pages.towson.edu/davani/cosc465/NXT/welcome_to_nxt.html
- [6] <http://76.74.128.40/~sti2d568/flsi/files/article/313/MCC%20en%20asservissement.pdf>.
- [7] **Coralie Germain** ,<< Conception d'un robot parallèle à deux degrés de liberté pour des opérations de prise et de dépose>>, Ecole Centrale de Nantes, France, 2013.

Annexes

- Annexe Programmation NXC :

Pour les fonctions utilisées :

```
void PosRegSetAngle ( byte output
                    ,
                    long angle
                    ) [inline]
```

Change the current value for set angle.

Make the absolute position regulation going toward the new provided angle. Returns immediately, but keep regulating.

Warning:

This function requires the enhanced NBC/NXC firmware version 1.31+

Parameters:

output Desired output port. Can be a constant or a variable, see [Output port constants](#).

angle New set position, in degree. The 0 angle corresponds to the position of the motor when absolute position regulation was first enabled. Can be negative. Can be greater than 360 degree to make several turns.

Examples:

C'est un exemple des fonctions: [PosRegEnable](#), [PosRegSetAngle](#), [PosRegAddAngle](#), [PosRegSetMax](#), [SetMotorRegulationTime](#), [SetMotorRegulationOptions](#), [MotorRegulationTime](#), [MotorRegulationOptions](#), [MotorMaxSpeed](#), and [MotorMaxAcceleration](#) functions.

```
task main()
{
  byte rt = MotorRegulationTime ();
  SetMotorRegulationTime (MS 10);
  byte ro = MotorRegulationOptions ();
  SetMotorRegulationOptions (OUT_REGOPTION_NO_SATURATION);
  PosRegSetMax (OUT_A, 75, 15);
  byte ms = MotorMaxSpeed (OUT_A);
  byte ma = MotorMaxAcceleration (OUT_A);
```

```
PosRegEnable (OUT A) ;  
PosRegSetAngle (OUT A, 360) ;  
Wait (5000) ;  
PosRegAddAngle (OUT A, 360) ;  
Wait (5000) ;
```

Pour la fonction:

void SetMotorRegulationTime (byte *n*) [inline]

Set regulation time.

Set the motor regulation time in milliseconds. By default this is set to 100ms.

Parameters:

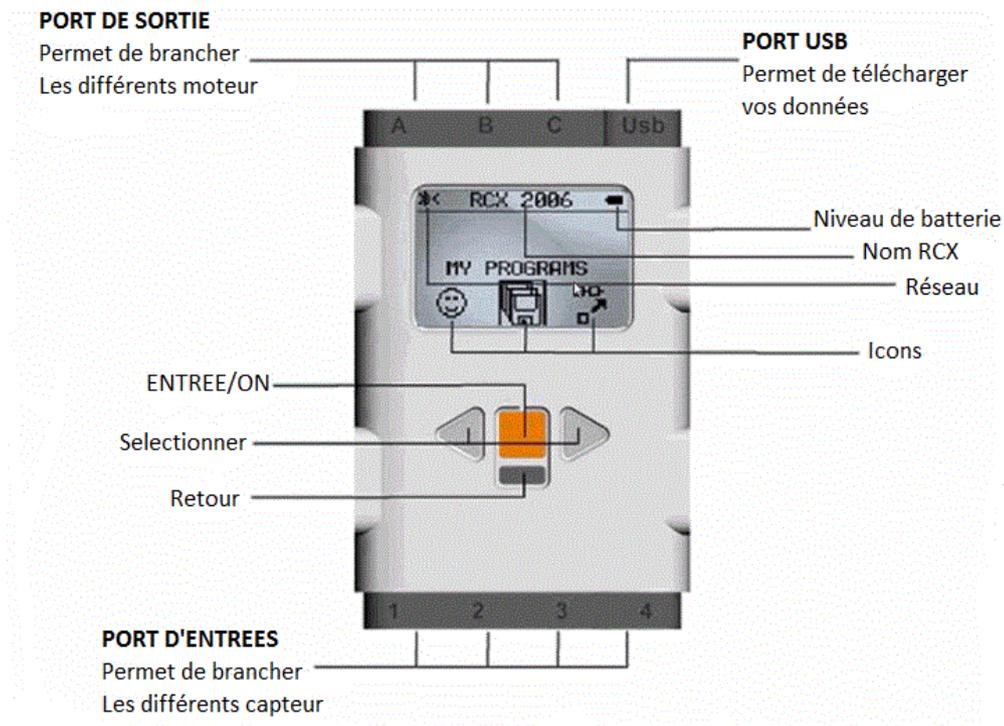
n The motor regulation time.

- Annexe brique NXT :

De quoi se compose un robot NXT ?

- D'une brique NXT qui stockera notre programme.
- De Servo-moteurs qui géreront les déplacement du robot.
- De différents capteurs (capteurs de couleur, sonars).
- De pièces Lego pour assembler le robot.

La brique NXT



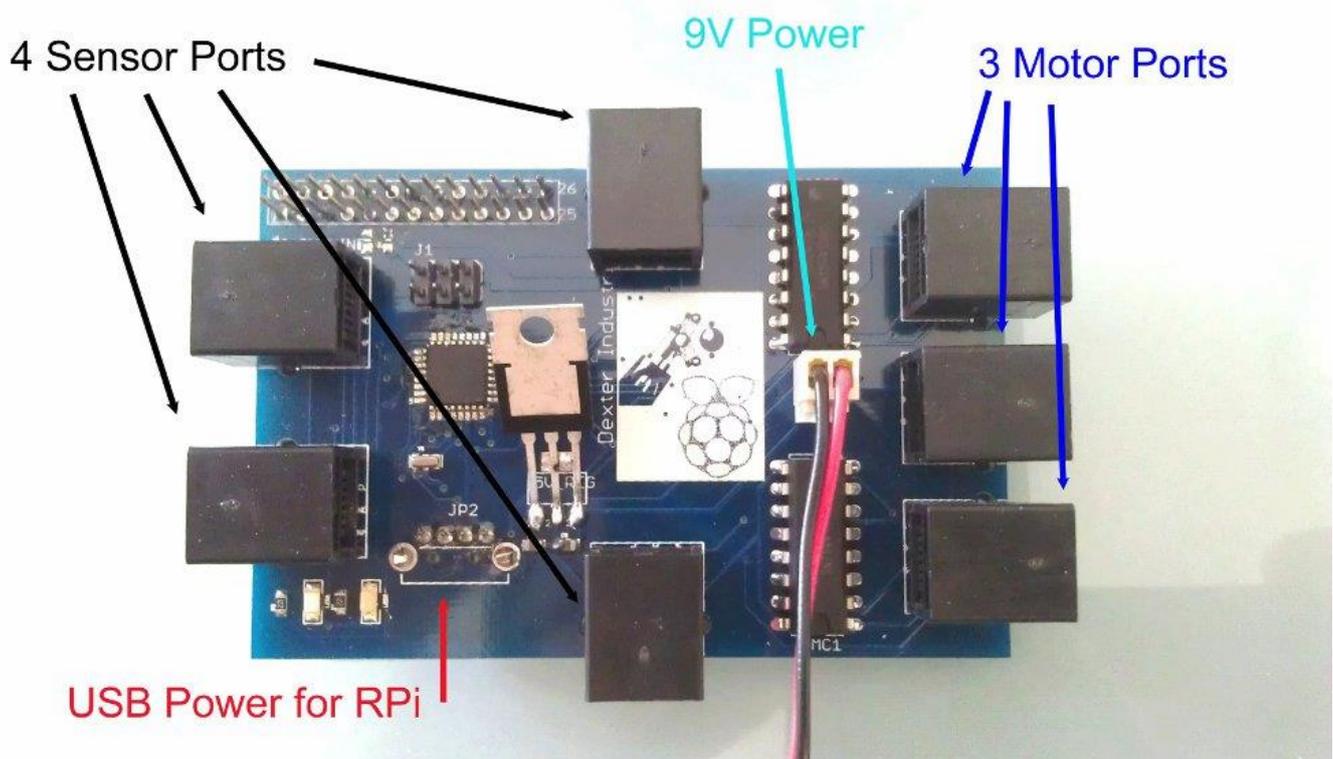
La brique NXT sera le cerveau de ton robot.

C'est dans ce centre nerveux que tu enregistreras le programme que le robot devra exécuter.

Elle comprend un microprocesseur d'une résolution de 32 bits, une mémoire Flash et prend en charge les technologies Bluetooth™ et USB 2.0!

Caractéristiques de la brique intelligente NXT :

- Microprocesseur 32 bits ARM7.
- Prise en charge de communications sans fil Bluetooth.
- 1 port USB 2.0.
- 4 ports d'entrée.
- 3 ports de sortie.
- Alimentation : 6 piles AA de 1,5 V (non incluses) N° 9798 ou N° 9693 Pile rechargeable au lithium (vendue séparément).
- Tu peux connecter jusqu'à trois briques NXT en même temps mais tu ne peux communiquer qu'avec une seule brique NXT à la fois



Carte de puissance brique NXT