

République Algérienne Démocratique et Populaire
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
Université Mouloud MAMMERI, Tizi-Ouzou



Faculté de Génie Electrique et d'Informatique
Département d'Automatique

Mémoire de Fin d'Etudes

En vue de l'obtention du diplôme

D'Ingénieur d'Etat en Automatique

Thème

***Commande hybride en force position d'un bras
manipulateur par FPGA***

Proposé par : **Dr MELLAH RABAH**

Présenté par :

Dirigé par : **Dr MELLAH RABAH**

GUEZOUT Djamal
MAOUCHI Nafaa

Soutenu le : 11 / 07 /2010

Promotion 2011

Sommaire :

Introduction générale	1
-----------------------------	---

Chapitre I : Modélisation géométrique du bras

I.1) Introduction	4
I.1.1) Constituants d'un robot	4
I.1.2) Caractéristiques d'un robot	7
I.1.3) Les générations de robot	7
I.1.4) DEGRÉ DE LIBERTÉS – ARCHITECTURE	8
I.1.6) Exemple d'orientation d'un repère (utilisation des angles d'Euler).....	8
I.1.6.1) Les angles sont définis dans le cas présent, selon la convention (z, y, z), comme suit	9
I.1.6.2) Liaison	9
I.1.6.3) Mécanismes	9
I.1.7) Degré de liberté	10
I.1.7) Définition	10
I.1.7.2) Définition (redondant)	11
I.2) Modélisation géométrique du bras	12
I.2.1) Matrices de transformations homogènes	12
I.2.2) Généralités	13
I.2.2.1) Coordonnées homogènes d'un point... ..	13
I.2.2.2) Représentation d'une direction (vecteurs libre)	13
I.2.2.3) Représentation d'un plan	13
I.2.2.4) Transformation des repères	13
I.2.2.5) Transformation des vecteurs	14

I.2.2.6) Transformation des plans	14
I.2.2.7) Matrice de translation et de rotation	15
I.2.3) Propriétés de la matrice de transformation homogène.....	16
I.3) Représentation de l'orientation d'un solide dans l'espace	17
I.3.1) Les angles roulis, tangage et lacet	18
I.4) Modèle géométrique direct du robot (ARM AL5C)	18
I.4.1) Définition	18
I.4.2) Configuration et situation d'un bras manipulateur	19
I.4.3) Paramètres de Danavit-Hartenberg modifiés	20
I.4.5) Calcul du modèle géométrique direct.....	21
I.5) Modèle géométrique inverse du robot	24
I.5.1) Définition	24
I.5.2) Solubilité du MGI	24
I.5.3) La méthode de Paul	24
I.5.4) Calcul du modèle géométrique inverse	25

Chapitre II : La commande

II.1) Introduction	29
II.2) La commande du bras manipulateur.....	29
II.2.1) La commande par retour d'effort	29
II.2.2) La commande adaptative	29
II.2.3) Commande hybride dynamique	30
II.2.4) Commande hybride externe	31
II.2.5) Loi de commande d'impédance	32
II.3) Choix et application de la loi de commande à notre système	33
II.3.1) Loi de commande en Position LCP	33

II.3.1.1) Définition de la loi trapèze	33
II.3.1.2) Loi de synchronisation	34
II.3.1.3) Planification par une loi de commande en position « cas non redondant »	35
II.3.2) Loi de commande en Force LCF	35
II.3.3) Réalisation d'une Commande généralisée effort – compensation	36

Chapitre III : Description matérielle et logicielle

III.1) Description matérielle	40
III.1.1) La commande	40
III.1.1.1) Centre de commande	40
III.1.1.2) Organe de Commande	40
III.1.2) L'acquisition	40
III.1.2.1) Centre d'acquisition	40
III.1.2.2) Organes d'acquisition	42
III.1.2.2.1) Acquisition de la force	42
III.1.2.2.2) Acquisition de la Position	44
III.1.3) Mouvement du Bras manipulateur	45
III.1.3.1) Organe du mouvement	45
III.1.3 .1) Centre du mouvement	49
III.2) Description Logicielle	50
III.1.1) Conception de la commande	50
III.2.2) Traitement de l'acquisition	51

Chapitre IV : Réalisation

IV.1) Introduction	53
IV.2) Description du projet FPGA	53

IV.3) Description des périphériques externe.	54
IV.3.1) Interface Homme-Machine	54
IV.3.2) Entrée Sortie des Informations traitées	55
IV.3.3) Les mémoires externes du dispositif	55
IV.4) Description interne du système	56
IV.4.1) Les Périphériques internes.....	56
IV.4.1.1) Périphérique écran TFT	56
IV.4.1.2) Périphérique RS232	57
V.4.1.3) Périphériques Mémoires	57
V.4.1.4) Les composantes internes.....	57
V.4.2) Le projet embarqué	58
V.5) Un Dispositif d'acquisition et relai de commande	59
V.5) Résultats de la Simulation	60

DEDICACES

A qui puis je dédier cet humble travail si ce n'est à mes très chers parents, dont le sacrifice, la tendresses, l'amour, la patience, le soutien, l'aide et les encouragements sont l'essence de ma réussite. Sans eux je ne serais pas ce que je suis aujourd'hui.

Je dédie ce travail également,

A mes sœurs.

A tous mes amis.

A toute personne qui me porte de l'estime.

Djamal

Je dédie ce mémoire à toute ma famille, mon père, ma mère, ma grand-mère,
mes sœurs et mon frère.

Nafaa

Introduction Générale

Quand on parle de robotique, plusieurs idées viennent à l'esprit de chacun de nous. Historiquement, nous pourrions nous référer aux premiers concepts et automates de l'antiquité ou aux premiers robots comme à des personnages de la mythologie. Même le mot robot a sa propre histoire. Séparer la science de la science fiction n'est pas une chose aisée, surtout lorsque en robotique nous cherchons parfois à faire réalité la fiction. Un exemple de l'influence des fictions nous est donné par les lois de la robotique.

C'est au siècle dernier que le « boom » de la robotique industrielle a amorcé l'explosion des thèmes de recherche. A cette époque les robots étaient conçus en respectant les contraintes imposées par le milieu industriel, comme la répétabilité, la précision dans la réalisation des tâches, le respect des cadences de production, etc. Prenons l'exemple des bras manipulateurs industriels des années 80 très semblables aux bras industriels actuels, ils se déplacent à très grandes vitesses sur des trajectoires calculées hors ligne dans des environnements protégés et structurés, sans interaction avec les opérateurs humains.

C'est avec les développements scientifiques, spécifiquement de l'électronique et de l'informatique mais aussi automatique, mathématique, mécanique, matériaux, que la technologie robotique a progressé.

Les robots actuels sont dotés d'une « intelligence » qui leur donne une certaine autonomie qui va leur permettre de se diffuser dans de nouveaux domaines. Nous parlons maintenant de robots d'exploration, de robots d'intervention, de robots ludiques.

Les applications en contrôle en force/position de manipulateurs robotiques sont nombreuses comme, par exemple, toutes les tâches qui nécessitent une interaction entre un robot et son environnement comme les opérations d'assemblages, les opérations sur les lignes électriques, les robots bipèdes ou encore les manipulateurs spatiaux.

Il existe plusieurs algorithmes de commande en force. Deux approches se distinguent néanmoins dans la littérature : le contrôle hybride force/position abordé par Robert et Craig, étendu, entre autres, par DeSantis et le contrôle d'impédance présenté par Hogan et repris par de nombreux chercheurs depuis comme Hsia Seraji et Colbaugh OU Pelletier.

Le contrôle hybride force/position consiste à diviser l'espace opérationnel en deux sous-espaces. Ainsi, la force est contrôlée dans une direction et la position dans une direction perpendiculaire à celle de la force.

Le contrôle d'impédance régit une relation entre la position et la force exercée par le manipulateur sur l'environnement, donc le contrôle est effectué sur l'impédance mécanique du robot. Les applications présentées dans la littérature prennent pour hypothèse que la position, la rigidité de l'environnement sont mal connues ou que le type de surface change pendant le trajet. Aouzu, lui, suppose que l'environnement peut être en mouvement. Il n'est, en effet, pas irréaliste d'effectuer des

tâches dans des milieux soumis à des vibrations ou au vent. L'étude d'Aouzou a montré que la commande par contrôleur discontinu présentait le plus d'avantages. Ce contrôleur consiste à utiliser deux lois de contrôle : une dans l'espace libre et une autre dans l'espace de contrainte.

Nous nous proposons de réaliser une commande hybride force position d'un bras manipulateur et cela par une carte FPGA.

Dans le Chapitre I nous procéderons à la modélisation de notre bras manipulateur. On déterminera sont modèle géométrique direct et modèle géométrique indirect usant de différentes lois de modélisation robotique notamment la conversion de Denavit Hartenberg.

Dans le chapitre II nous évoquerons les différents types de commandes hybrides en force position et nous fixerons notre choix sur l'une de ces commandes. Le chapitre III se veut être un chapitre descriptif ou l'on fera une description sommaire des différentes composantes matérielles et logicielles de notre projet.

Enfin le Chapitre IV sera un rapide exposé sur la réalisation de notre projet. On citera notamment les différents procédés utilisé pour la conception matérielle et logicielle et les résultats obtenus après expérience.

Chapitre I : Modélisation géométrique du bras

I.1) Introduction :

Pour commander ou simuler le comportement d'un système mécanique articulé² (robot), on doit disposer d'un modèle .Plusieurs niveaux de modélisation sont possibles selon les objectifs, les contraintes de la tâche et les performances recherchées.

Les modèles mathématiques nécessaires sont :

- les modèles géométriques directs et inverses qui expriment la situation de l'organe terminal en fonction des variables articulaire et inversement.
- les modèles cinématiques direct et inverse qui expriment les vitesses de l'organe terminal en fonction des variables articulaires et inversement.

Difficultés : Complexité de la cinématique

Le nombre de degré de liberté

Type d'articulation (prismatique ou rotoïde)

Type de chaîne (ouverte, simple, arborescente ou fermée)

Pour obtenir un bon modèle il faut :

1/ Mettre en œuvre des procédures efficaces d'identification et de leurs paramètres constitutifs.

2/ Pour qu'une commande puisse être effectivement implantée sur un contrôleur de robot, les modèles doivent être calculés en ligne et donc le nombre d'opération à effectuer doit être minimum.

I.1.1) Constituants d'un robot :

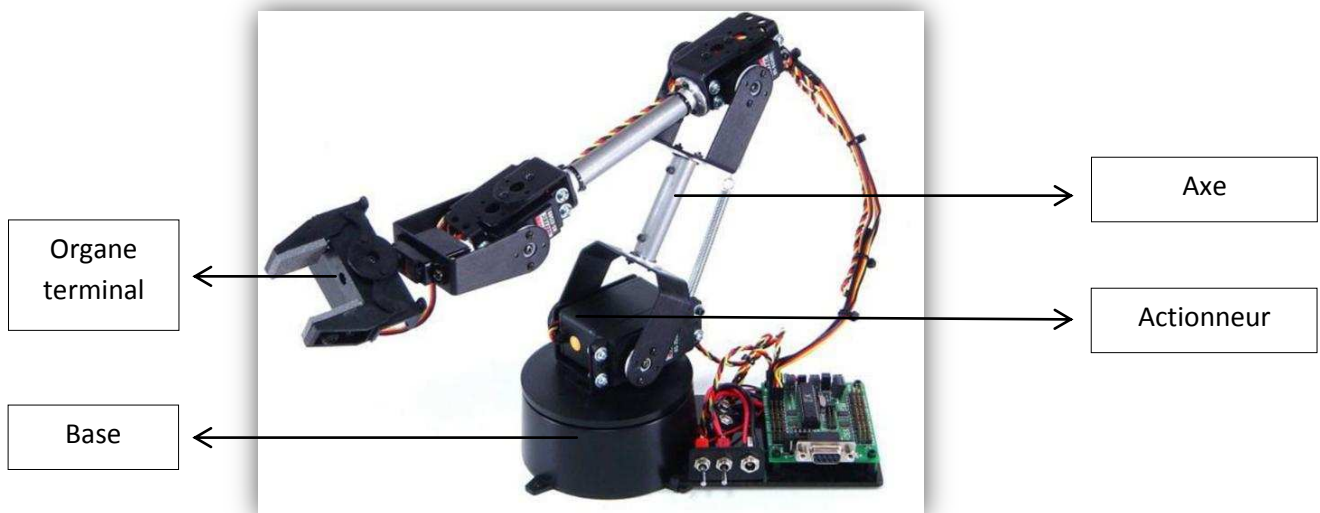


Fig I.1 Constituants d'un robot

On distingue classiquement 4 parties principales dans un robot manipulateur :

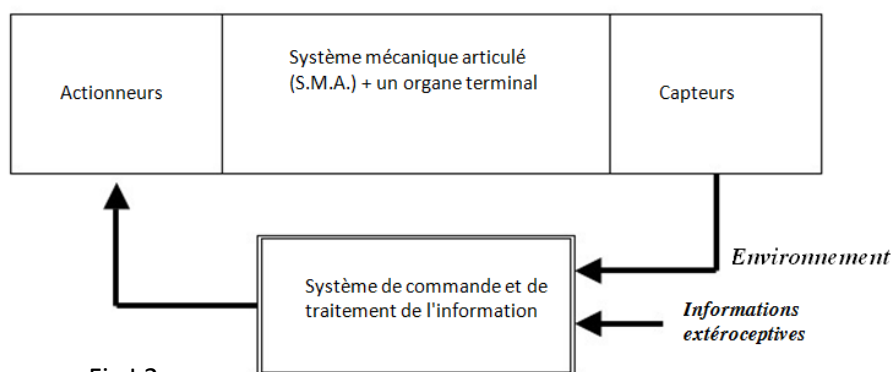


Fig I.2

- Sous le terme organe terminal, on regroupe tout dispositif destiné à manipuler des objets (dispositifs de serrage, dispositifs magnétiques, à dépression, ...), ou à les transformer (outils, torche de soudage, pistolet de peinture, ...). En d'autres termes, il s'agit d'une interface permettant au robot d'interagir avec son environnement. Un organe terminal peut être multifonctionnel, au sens où il peut être équipé de plusieurs dispositifs ayant des fonctionnalités différentes. Il peut aussi être monofonctionnel, mais interchangeable. Un robot, enfin, peut-être multi-bras, chacun des bras portant un organe terminal différent. On utilisera indifféremment le terme organe terminal, préhenseur, outil ou effecteur pour nommer le dispositif d'interaction fixé à l'extrémité mobile de la structure mécanique.

- Le système mécanique articulé (S.M.A.) est un mécanisme ayant une structure plus ou moins proche de celle du bras humain. Il permet de remplacer, ou de prolonger, son action (le terme "manipulateur" exclut implicitement les robots mobiles autonomes)³. Son rôle est d'amener l'organe terminal dans une situation (position et orientation) donnée, selon des caractéristiques de vitesse et d'accélération données. Son architecture est une chaîne cinématique de corps, généralement rigides (ou supposés comme tels), assemblés par des liaisons appelées articulations. Sa motorisation est réalisée par des actionneurs électriques, pneumatiques ou hydrauliques qui transmettent leurs mouvements aux articulations par des systèmes appropriés.

- Précisons la notion d'articulation : Une articulation lie deux corps successifs en limitant le nombre de degré de liberté de l'un par rapport à l'autre. Soit m le nombre de degré de liberté résultant, encore appelé mobilité de l'articulation. La mobilité d'une articulation est telle que :

$$0 \leq m \leq 6$$

Lorsque $m = 1$; ce qui est fréquemment le cas en robotique, l'articulation est dite simple : soit rotoïde, soit prismatique.

○ Articulation rotoïde : Il s'agit d'une articulation de type pivot, notée R, réduisant le mouvement entre deux corps à une rotation autour d'un axe qui leur est commun. La situation relative entre les deux corps est donnée par l'angle autour de cet axe (voir la figure suivante).



Fig I.3 Articulation rotoïde

○ Articulation prismatique : Il s'agit d'une articulation de type glissière, notée P, réduisant le mouvement entre deux corps à une translation le long d'un axe commun. La situation relative entre les deux corps est mesurée par la distance le long de cet axe (voir la figure suivante).



Fig I.4 : Symbole de l'articulation prismatique.

Remarque : Une articulation complexe, i.e., avec une mobilité supérieure à 1, peut toujours se ramener à une combinaison d'articulations prismatique ou rotoïde. Par exemple, une rotule est obtenue avec trois articulations rotoïdes dont les axes sont concourants.

Pour être animé, le S.M.A. comporte des moteurs le plus souvent avec des transmissions (courroies crantées), l'ensemble constitue les actionneurs. Les actionneurs utilisent fréquemment des moteurs électriques à aimant permanent, à courant continu, à commande par l'induit (la tension n'est continue qu'en moyenne car en général l'alimentation est un hacheur de tension à fréquence élevée ; bien souvent la vitesse de régime élevée du moteur fait qu'il est suivi d'un réducteur, ce qui permet d'amplifier le couple moteur). On trouve de plus en plus de moteurs à commutation électronique (sans balais), ou, pour de petits robots, des moteurs pas à pas.

Pour les robots devant manipuler de très lourdes charges (par exemple, une pelle mécanique), les actionneurs sont le plus souvent hydrauliques, agissant en translation (vérin hydraulique) ou en rotation (moteur hydraulique).

Les actionneurs pneumatiques sont d'un usage général pour les manipulateurs à cycles (robots tout ou rien). Un manipulateur à cycles est un S.M.A. avec un nombre limité de degrés de liberté permettant une succession de mouvements contrôlés uniquement par des capteurs de fin de course réglables manuellement à la course désirée (asservissement en position difficile dû à la compressibilité de l'air).

La perception permet de gérer les relations entre le robot et son environnement. Les organes de perception sont des capteurs dits proprioceptifs lorsqu'ils mesurent l'état interne du robot (positions et vitesses des articulations) et extéroceptifs lorsqu'ils recueillent des informations sur l'environnement (détection de présence, de contact, mesure de distance, vision artificielle).

La partie commande synthétise les consignes des asservissements pilotant les actionneurs, à partir de la fonction de perception et des ordres de l'utilisateur.

S'ajoutent à cela :

- L'interface homme-machine à travers laquelle l'utilisateur programme les tâches que le robot doit exécuter,
- Le poste de travail, ou l'environnement dans lequel évolue le robot.

La robotique est une science pluridisciplinaire qui requiert, notamment, des connaissances en mécanique, automatique, électronique, électrotechnique, traitement du signal, communications, informatique.

Classification des robots :

On retiendra pour notre part 3 types de robot :

- Les manipulateurs :
 - Les trajectoires sont non quelconques dans l'espace,
 - Les positions sont discrètes avec 2 ou 3 valeurs par axe,
 - La commande est séquentielle.
- Les télémanipulateurs : appareils de manipulation à distance (pelle mécanique, pont roulant), apparus vers 1945 aux USA :
 - Les trajectoires peuvent être quelconques dans l'espace,
 - Les trajectoires sont définies de manière instantanée par l'opérateur, généralement à partir d'un pupitre de commande (joystick).
- Les robots :
 - Les trajectoires peuvent être quelconques dans l'espace,
 - L'exécution est automatique,
 - Les informations extéroceptives peuvent modifier le comportement du robot.

Chapitre I MODELISATION GEOMETRIQUE DU BRAS

Pour cette dernière classe, on peut distinguer :

1. Les robots manipulateurs industriels, chargés de manipuler, soient :

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

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

2. Les robots didactiques, qui sont des versions au format réduit des précédents robots. La technologie est différente, de même que les constructeurs. Ils ont un rôle de formation et d'enseignement, ils peuvent aussi être utilisés pour effectuer des tests de faisabilité d'un poste robotisé.

3. Les robots mobiles autonomes : Les possibilités sont plus vastes, du fait de leur mobilité.

Notamment, ils peuvent être utilisés en zone dangereuse (nucléaire, incendie, sécurité civile, déminage), inaccessible (océanographie, spatial). De tels robots font appel à des capteurs et à des logiciels sophistiqués. On peut distinguer 2 types de locomotion : Les robots marcheurs qui imitent la démarche humaine, et les robots mobiles qui ressemblent plus à des véhicules.

Dans ce cas, on se restreint aux robots manipulateurs.

I.1.2) Caractéristiques d'un robot :

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

- La charge maximum transportable (de quelques kilos à quelques tonnes), à déterminer dans les conditions les plus défavorables (en élancement maximum).

- L'architecture du S.M.A., le choix est guidé par la tâche à réaliser (quelle est la rigidité de la structure ?).

- Le volume de travail, défini comme l'ensemble des points atteignables par l'organe terminal. Tous les mouvements ne sont pas possibles en tout point du volume de travail. L'espace de travail (reachable workspace), également appelé espace de travail maximal, est le volume de l'espace que le robot peut atteindre via au moins une orientation. L'espace de travail dextre (dextrous workspace) est le volume de l'espace que le robot peut atteindre avec toutes les orientations possibles de l'effecteur (organe terminal). Cet espace de travail est un sous-ensemble de l'espace de travail maximal.

- Le positionnement absolu, correspondant à l'erreur entre un point souhaité (réel) – défini par une position et une orientation dans l'espace cartésien – et le point atteint et calculé via le modèle géométrique inverse du robot. Cette erreur est due au modèle utilisé, à la quantification de la mesure de position, à la flexibilité du système mécanique. En général, l'erreur de positionnement absolu, également appelée précision, est de l'ordre de 0.1 mm.

- La répétabilité, ce paramètre caractérise la retourner vers un point (position, orientation) correspond à l'erreur maximum de point prédéfini dans le cas de trajectoires la répétabilité est de l'ordre de 0,1 mm. - La (vitesse maximum en élancement maximum), accélération.

- Mécanique,
- Micro-informatique,
- Energétique,
- Capteurs – actionneurs.

capacité que le robot a à donné. La répétabilité positionnement sur un répétitives. En général, vitesse de déplacement

- La masse du robot.
- Le coût du robot.
- La maintenance, ...

I.1.3) Les générations de robot :

Des progressions s'opèrent dans tous les domaines :

A l'heure actuelle, on peut distinguer 3 générations de robots :

1. Le robot est passif : Il est capable d'exécuter une tâche qui peut être complexe, mais de manière répétitive, il ne doit pas y avoir de modifications intempestives de l'environnement. L'auto-adaptativité est très faible. De nombreux robots sont encore de cette génération.
2. Le robot devient actif : Il devient capable d'avoir une image de son environnement, et donc de choisir le bon comportement (sachant que les différentes configurations ont été prévues). Le robot peut se calibrer tout seul.
3. Le robot devient « intelligent » : Le robot est capable d'établir des stratégies, ce qui fait appel à des capteurs sophistiqués, et souvent à l'intelligence artificielle.

I.1.4) DEGRÉ DE LIBERTÉS – ARCHITECTURE :

Positionnement d'un solide dans l'espace

La position d'un solide dans l'espace requiert 6 paramètres indépendants (cf. figure suivante) :

- 3 paramètres indépendants définissent la position d'un point, noté P, du solide (coordonnées cartésiennes, cylindriques, sphériques, ..., dans la base du repère fixe).
- 3 paramètres indépendants déterminent l'orientation du solide autour du point P (angles d'Euler, paramètres d'Euler, ...).

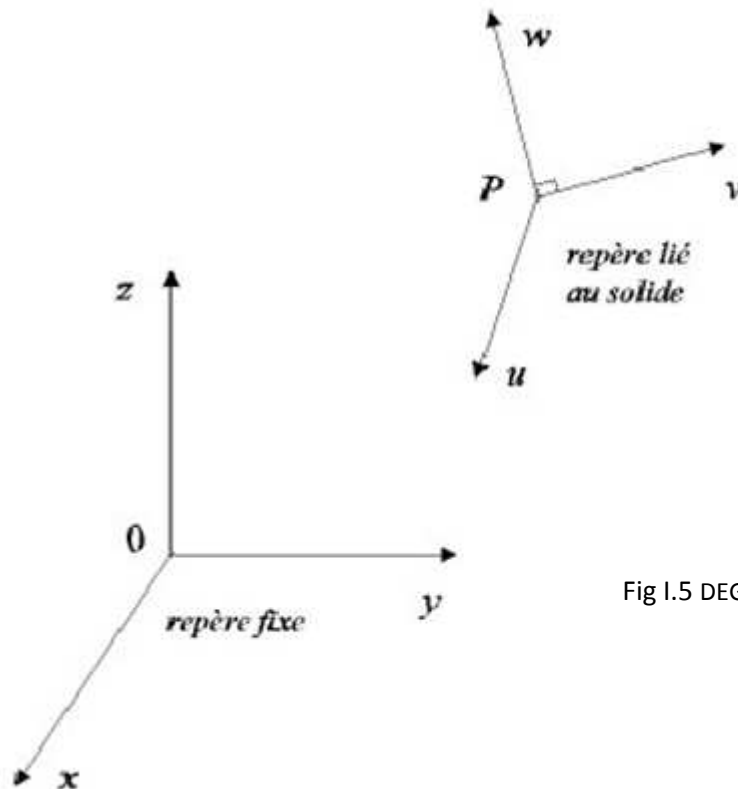


Fig I.5 DEGRÉ DE LIBERTÉS – ARCHITECTURE

I.1.6) Exemple d'orientation d'un repère (utilisation des angles d'Euler) :

L'orientation d'un repère orthonormé (lié au solide) par rapport au repère fixe est spécifiée par 3 angles ψ (psi), θ (thêta) et ϕ (phi), correspondant à une séquence de 3 rotations (cf. figure suivante). De tels angles sont très utilisés en mécanique. Ils permettent une définition minimum de l'orientation (3 paramètres), par contre, ils appréhendent difficilement le cas de composition de mouvement.

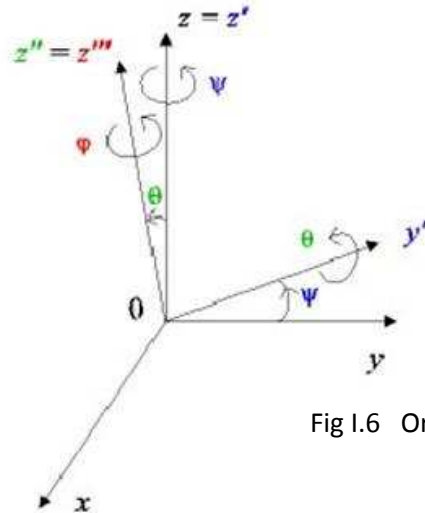


Fig I.6 Orientation d'un repère

I.1.6.1) Les angles sont définis dans le cas présent, selon la convention (z, y, z), comme suit :

- On fait une rotation d'un angle ϕ appelé lacet (yaw en anglais), autour de l'axe Oz ,
- On fait une rotation d'un angle θ appelé tangage (pitch en anglais), autour de l'axe Oy' ,
- On fait une rotation d'un angle ψ appelé roulis (roll en anglais), autour de l'axe Oz'' .

Un solide peut également être repéré par les coordonnées de 3 points (non situés sur une droite du solide).

Il en résulte 9 paramètres (3 coordonnées par point) non indépendants (3 équations expriment l'invariabilité de la distance entre les 3 points). Notons que de par le lien établi par ces équations, il reste 6 paramètres indépendants.

On dit qu'un solide situé dans l'espace possède 6 degrés de liberté (d.d.l.). Réciproquement, il faut 6 variables de commande indépendantes pour placer de manière quelconque un solide dans l'espace.

En pratique, les robots les plus courants sont dotés de 6 d.d.l., i.e., d'au moins 6 actionneurs, ce qui permet de spécifier de manière quelconque la situation (position et orientation) de leurs organes terminaux.

I.1.6.2) Liaison :

Une liaison entre 2 solides indéformables (en théorie) limite le d.d.l. d'un solide par rapport à l'autre. On appelle d.d.l. de la liaison le nombre de paramètres indépendants permettant de définir la localisation (position et orientation) d'un solide par rapport à l'autre dans tout déplacement (compatible avec la liaison).

I.1.6.3) Mécanismes :

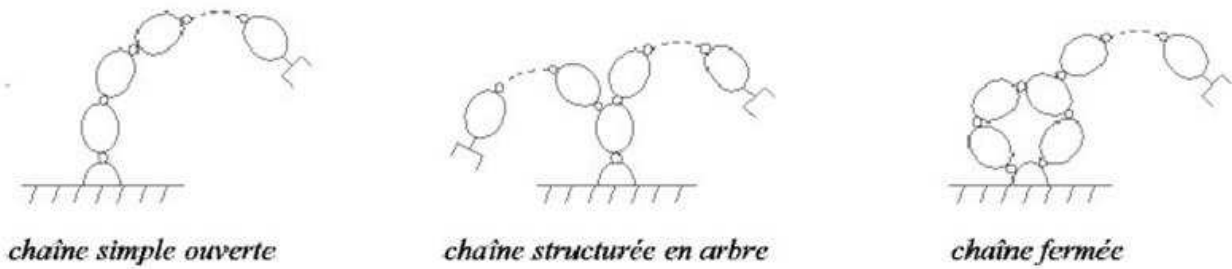
On appelle mécanisme un ensemble de solides reliés 2 à 2 par des liaisons. On distingue 2 types de mécanismes :

- Les mécanismes en chaîne simple ouverte (ou en série). Lorsque l'on parcourt le mécanisme, on ne repasse jamais 2 fois sur la même liaison, ou sur le même solide. Ce type de système est le plus répandu.
- Les mécanismes en chaîne complexe, i.e., tout ce qui n'est pas en série (au moins un solide avec plus de 2 liaisons). De tels systèmes se subdivisent en 2 groupes : les chaînes structurées en arbre, et les chaînes fermées (dont l'avantage est d'être a priori plus rigide, plus précis, capable de manipuler de lourdes charges). A titre d'exemple, le pantographe⁷ est un mécanisme en chaîne fermée.

Pour représenter un mécanisme, on dispose de 2 méthodes :

- **Le schéma cinématique** : On utilise la représentation normalisée des liaisons pour représenter le mécanisme, soit en perspective, soit en projection.

- **Le graphe** : non normalisé. A titre d'exemples, considérons quelques mécanismes :



II.1.7) Degré de liberté :

Fig I.7

II.1.7.1) Définition (degré de liberté, d.d.l.) : Le nombre de d.d.l. d'un mécanisme est le nombre de paramètres indépendants qui permettent de définir la position du mécanisme à un instant donné du mouvement.

Exemples :

- Chaînes simples ouvertes

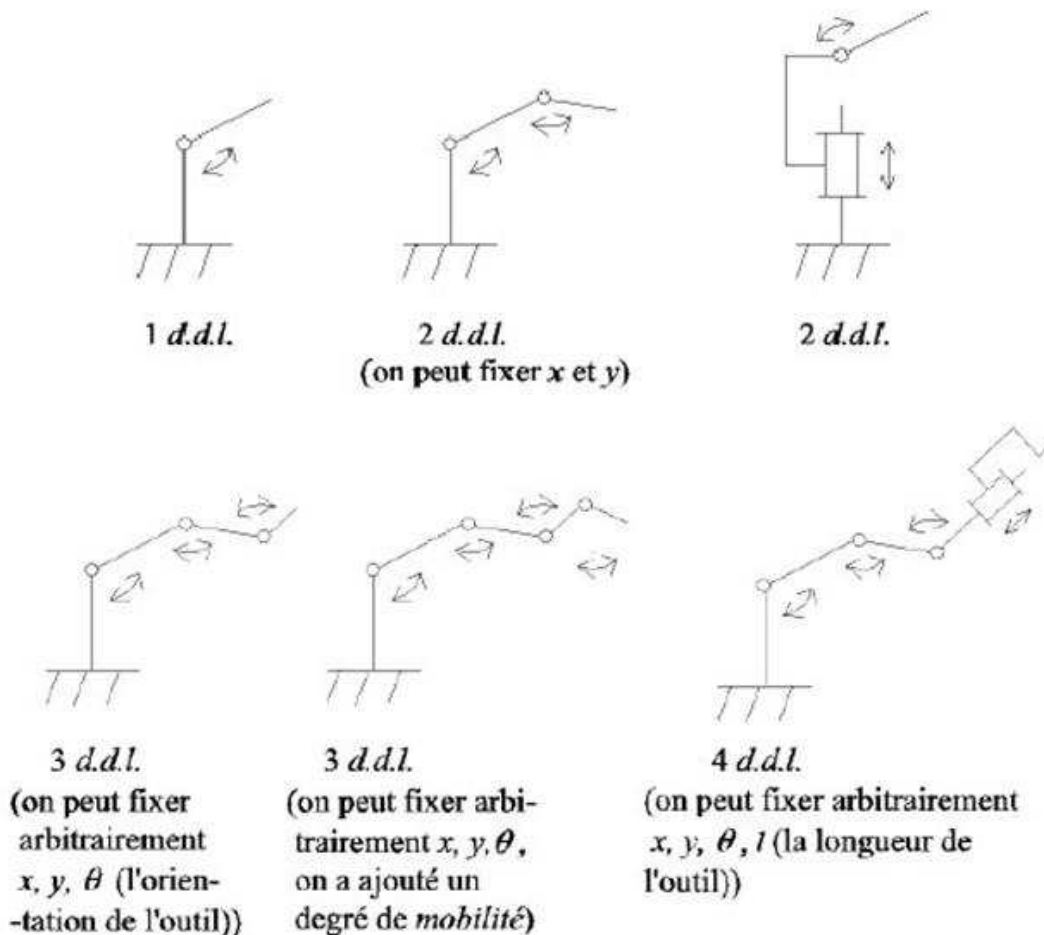


Fig I.8

- Chaînes complexes :

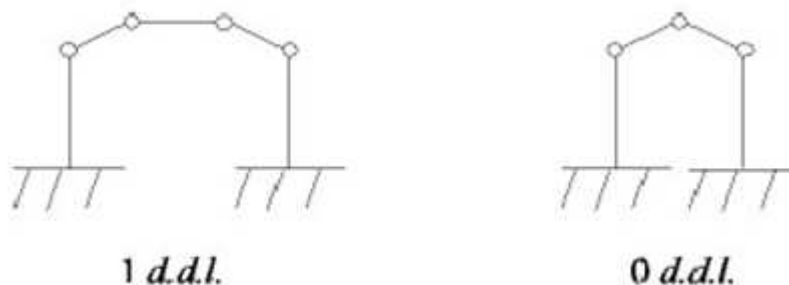


Fig I.9

I.1.7.2) Définition (redondant) : Un robot est redondant lorsque le nombre de d.d.l. du mécanisme est inférieur au nombre d'articulations indépendantes (motorisées). Cette propriété permet de préserver les capacités de déplacement de l'organe terminal en présence d'obstacles, le (ou les) d.d.l. supplémentaire(s) autorisant leur contournement.

I.1.7.3) Morphologie des robots manipulateurs :

Ce paragraphe est relatif aux chaînes ouvertes simples. Afin de dénombrer les différentes architectures possibles, on ne considère que 2 paramètres : le type d'articulation (rotoïde (R) ou prismatique (P)) et l'angle que font deux axes articulaires successifs (0° ou 90° ; sauf cas très particulier, les axes consécutifs d'un robot sont soit parallèles, soit perpendiculaires).

On convient d'appeler les 3 premiers d.d.l. le porteur du robot. Les d.d.l. résiduels forment le poignet, caractérisé par des dimensions beaucoup plus petites et une plus faible masse.

Sont schématisées dans la figure qui suit les 12 morphologies possibles de porteur (ces morphologies sont non redondantes (on élimine a priori les structures limitant les mouvements du porteur à des déplacements linéaires ou planaires : 3 liaisons prismatiques d'axes parallèles, par exemple, ou 3 liaisons rotoïdes d'axes parallèles)).

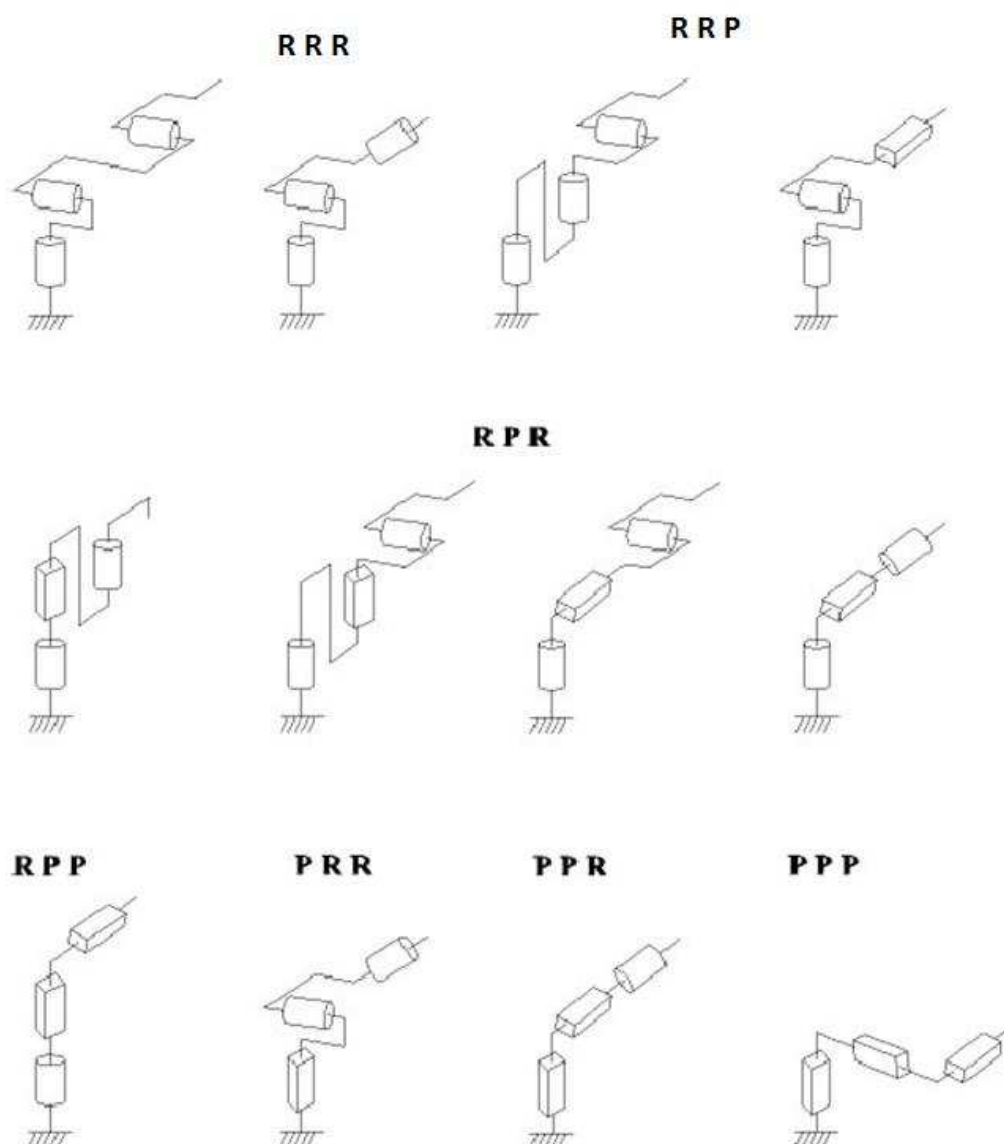


Fig I.10

Dans la pratique, on trouve les 5 structures suivantes :

- Les porteurs anthropomorphes (RRR), et plus précisément la première structure de cette classe (cf. figure précédente), comme par exemple les robots FANUC (LR, ARC), STÄUBLI RX, ACMA (V80 et SR400), UNIMATION (PUMA), SCEMI (6P-01), AID (V5), CINCINNATI (T3-7XX), AKR 3000, ASEA (IRB6 et 60), KUKA (IR600), AXEA (V08) ;
- Les porteurs sphériques (RRP) comme par exemple les robots STANFORD, UNIMATION (1000, 2000, 4000), PSA (BARNABE) ;
- Les porteurs toriques (RPR) et plus précisément la première structure de cette classe, comme par exemple les robots ACMA (H80), les robots de type SCARA (IBM, AXERA, ADEPT, ...) ;
- Les porteurs cylindriques (RPP) comme par exemple les robots ACMA (TH8), MANTEC (A, I et M), CINCINNATI (T3-363) ;
- Les porteurs cartésiens (PPP) comme par exemple les robots ACMA (P80), IBM (7565), SORMEL (CADRATIC), OLIVETTI (SIGMA).

La structure RRR dont les 3 axes sont concourants forme ainsi une rotule et s'utilise plus généralement comme un poignet. D'autres types de poignets de un à trois axes sont représentés sur la figure suivante :

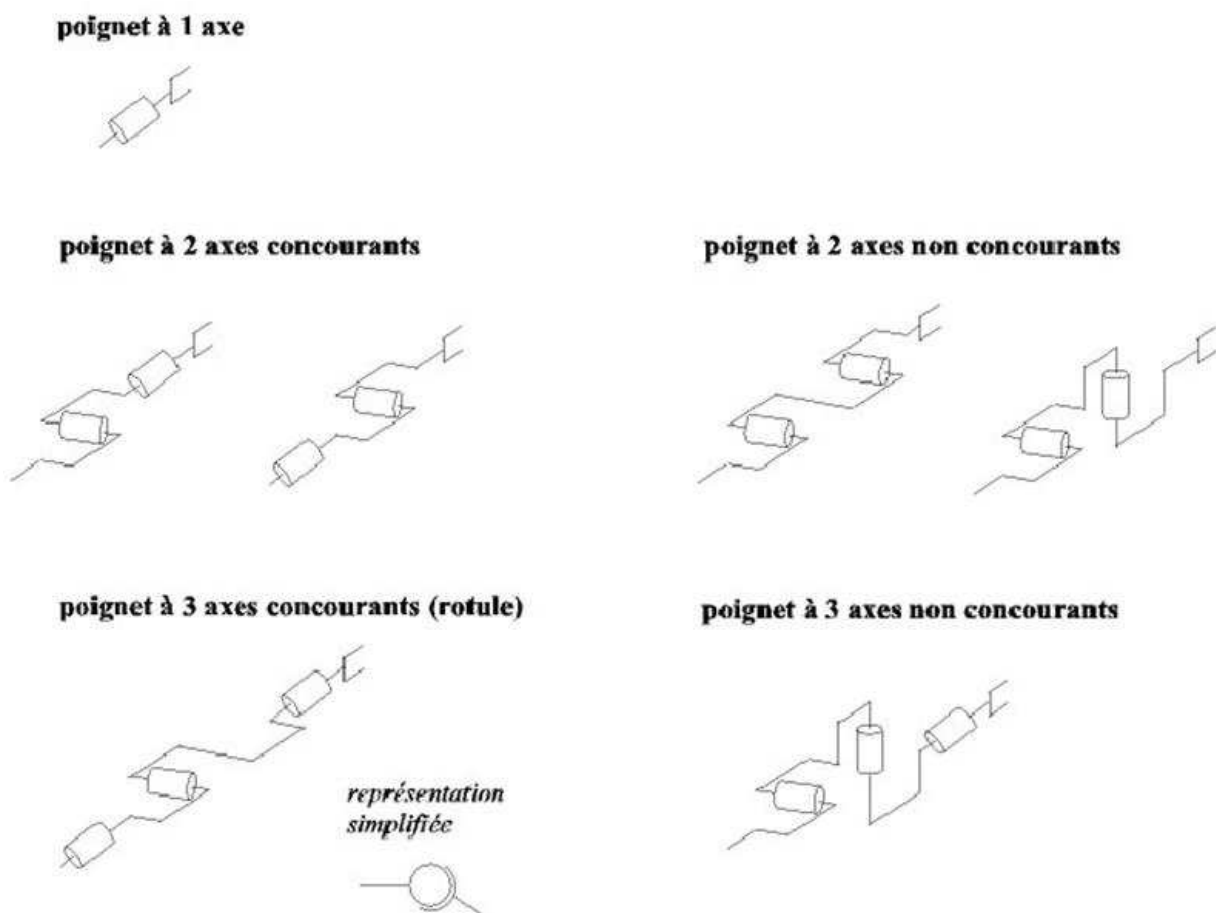


Fig I.11

I.2) Modélisation géométrique du bras :

I.2.1) Matrices de transformations homogènes :

Les transformations homogènes sont des outils mathématiques qui nous permettent de décrire d'une façon homogène les différents systèmes de coordonnées.

I.2.2) Généralités :

I.2.2.1) Coordonnées homogènes d'un point:

Un point est représenté par P_x, P_y, P_z coordonnées cartésiennes x, y, z :

$$P = \begin{bmatrix} w \cdot P_x \\ w \cdot P_y \\ w \cdot P_z \\ w \end{bmatrix}; w \text{ est un entier dit facteur d'échelle. En robotique, } w=1, \text{ donc :}$$

$$P = \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix}$$

I.2.2.2) Représentation d'une direction (vecteurs libre) :

$$V = \begin{bmatrix} V_x \\ V_y \\ V_z \\ 0 \end{bmatrix}$$

I.2.2.3) Représentation d'un plan :

Le plan $\alpha x + \beta y + \gamma z = \delta$ est représenté par un vecteur $Q = [\alpha, \beta, \gamma, \delta]$ pour tout point appartenant à Q , $Q \cdot P = 0$

$$Q \cdot P = [\alpha, \beta, \gamma, \delta] \cdot \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} = 0$$

I.2.2.4) Transformation des repères :

Pour transformer (translation ou rotation) un repère R_i et l'amener dans un autre repère R_j , on fait appel à une matrice dite matrice de transformation homogène ou bien matrice de passage homogène notée ${}^i T_j$

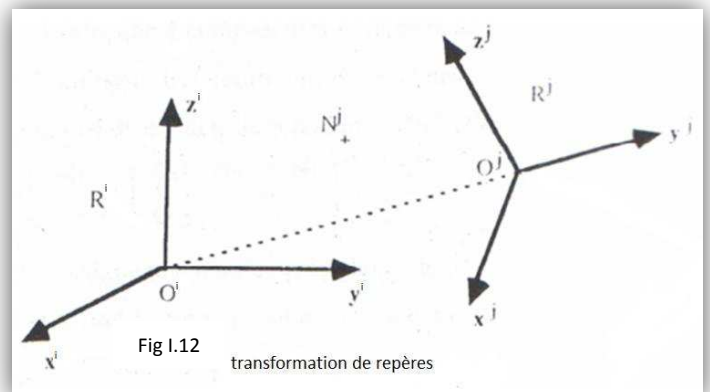
Soit le repère R_j défini par trois vecteurs unitaires a_j, b_j, c_j et son origine p_j .

Les coordonnées des vecteurs unitaires et de l'origine du repère R_j dans le repère de base R_i sont respectivement données comme suit :

$${}^i a_j = \begin{bmatrix} a_x \\ a_y \\ a_z \\ 0 \end{bmatrix}; \quad {}^i b_j = \begin{bmatrix} b_x \\ b_y \\ b_z \\ 0 \end{bmatrix}; \quad {}^i c_j = \begin{bmatrix} c_x \\ c_y \\ c_z \\ 0 \end{bmatrix};$$

$${}^i p_j = \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix}$$

La matrice de transformation homogène est donnée comme suit :



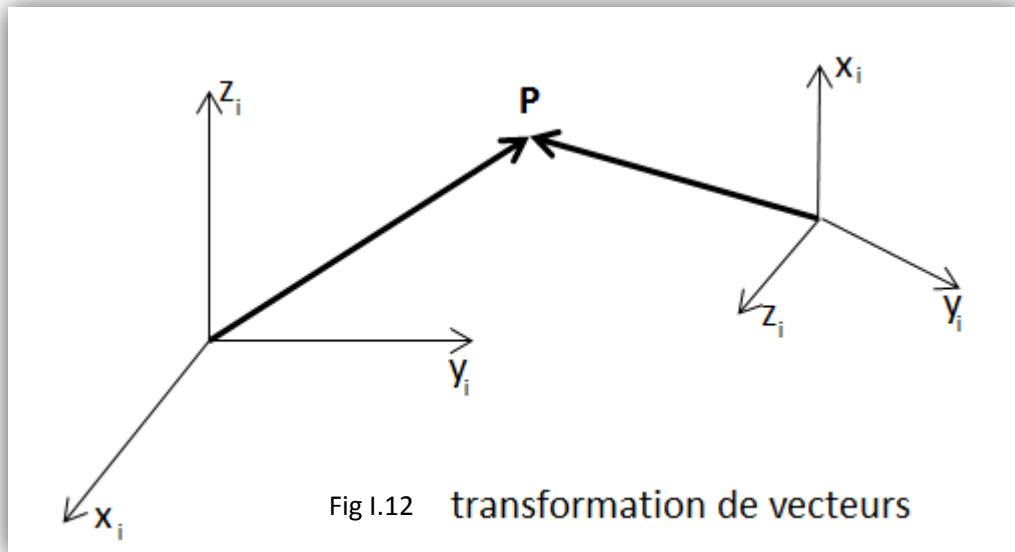
$${}^i T_j = [{}^i a_j, {}^i b_j, {}^i c_j, {}^i p_j] = \begin{bmatrix} a_x & b_x & c_x & p_x \\ a_y & b_y & c_y & p_y \\ a_z & b_z & c_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

I.2.2.5) Transformation des vecteurs : soit le vecteur V défini par son origine O et un point P de coordonnées homogènes dans le repère R_i : $P = [P_x \ P_y \ P_z \ 1]^t$

Exprimons le vecteur V défini dans R_j dans un autre repère R_i .

$${}^i V_j = {}^i a_j . P_x + {}^i b_j . P_y + {}^i c_j . P_z + {}^i O_j = {}^i T_j * P$$

La matrice ${}^i T_j$ nous permet d'exprimer dans le repère R_i les coordonnées de tout points appartenant au repère R_j .



I.2.2.6) Transformation des plans :

Soit un plan ϕ défini dans le repère R_i décrit par le vecteur ligne : ${}^i Q = [\alpha \ \beta \ \gamma \ \delta]$.

La matrice de transformation homogène qui nous permet d'exprimer un plan dans un autre repère est :

$${}^i Q = {}^i Q \ {}^i T_j$$

faisons subir au repère R_j une translation donnée par $l = [a \ b \ c]^t$

avec l coordonnée de l'origine du nouveau repère dans le repère R_i

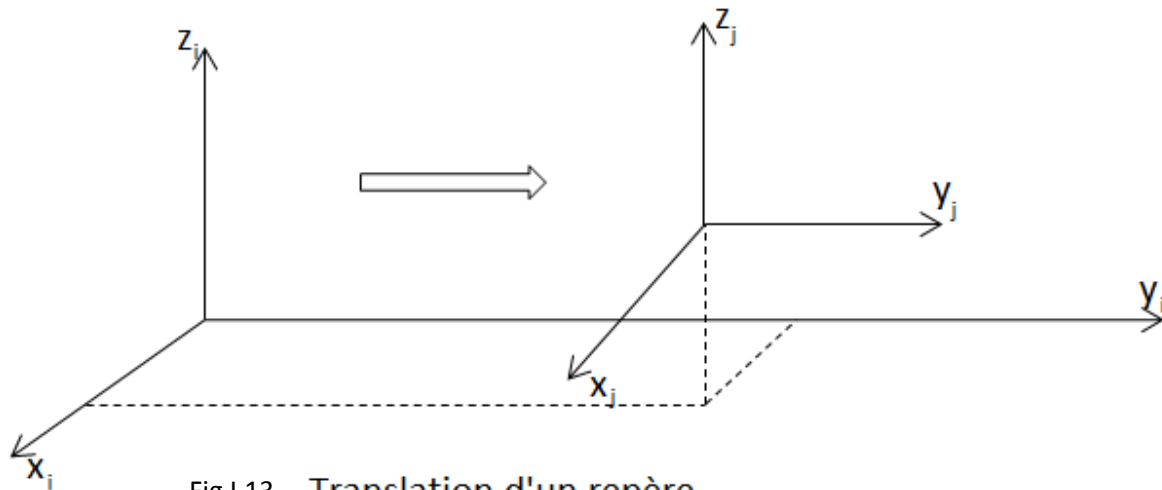


Fig I.13 Translation d'un repère

La translation notée **Trans(a,b,c)** est donnée par :

$$\mathbf{Trans(a,b,c)} = {}^i\mathbf{T}_j = \begin{bmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

1.2.2.7) Matrice de translation et de rotation :

a- Rotation autour de l'axe X :

Faisons subir au repère R_i une rotation θ autour de l'axe X.

$$\left\{ \begin{array}{l} \vec{a}_j = \vec{a}_i \\ \vec{b}_j = \cos(\theta) \cdot \vec{b}_i + \sin(\theta) \cdot \vec{c}_i \\ \vec{c}_j = \cos(\theta) \cdot \vec{c}_i - \sin(\theta) \cdot \vec{b}_i \end{array} \right.$$

la matrice de transformation de rotation autour l'axe X notée $\text{Rot}(x,)$ est donnée par :

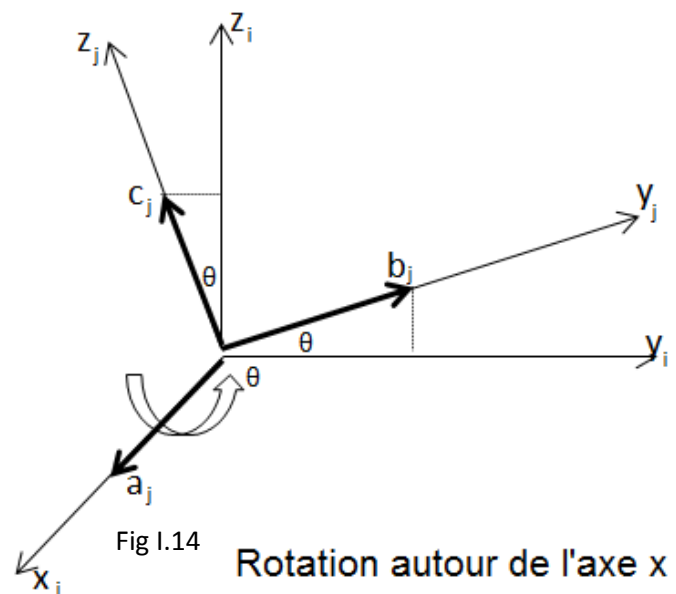


Fig I.14

Rotation autour de l'axe x

Pour simplifier les écritures matricielles, notons : $\cos(\theta) = c(\theta)$

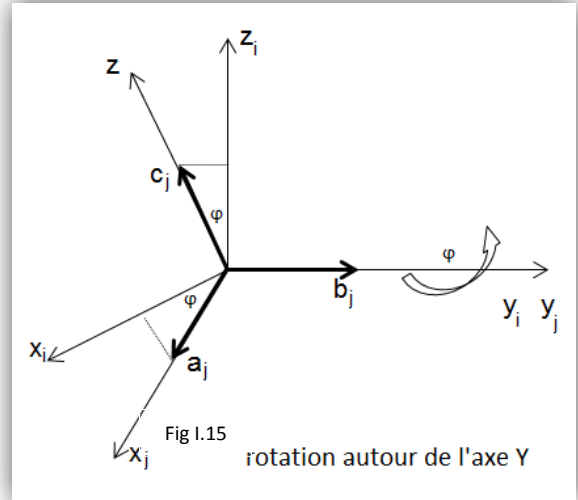
$\sin(\theta) = s(\theta)$

$$\text{Rot}(x, \theta) = {}^i T_j = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c(\theta) & -s(\theta) & 0 \\ 0 & s(\theta) & c(\theta) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

b- Rotation autour de l'axe Y :

$$\left\{ \begin{array}{l} \vec{a}_j \equiv \vec{a}_i \\ \vec{b}_j \equiv \cos(\theta) \cdot \vec{b}_i + \sin(\theta) \cdot \vec{c}_i \\ \vec{c}_j \equiv \cos(\theta) \cdot \vec{c}_i - \sin(\theta) \cdot \vec{b}_i \end{array} \right.$$

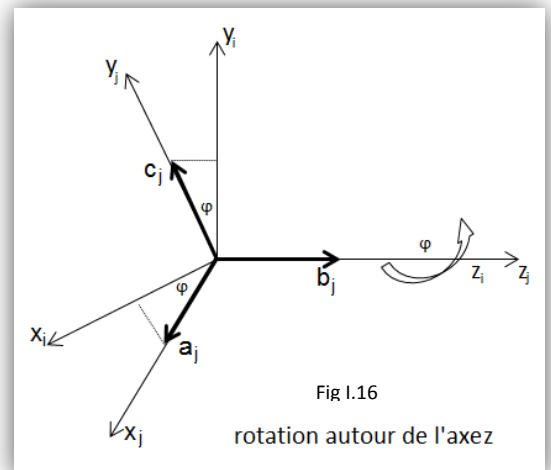
$$\text{Rot}(y, \phi) = {}^i T_j = \begin{bmatrix} c(\phi) & 0 & s(\phi) & 0 \\ 0 & 1 & 0 & 0 \\ -s(\phi) & 0 & c(\phi) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



c- Rotation autour de l'axe Z : faisons subir au repère R_i une rotation ψ autour de l'axe Z.

$$\left\{ \begin{array}{l} \vec{c}_j \equiv \vec{c}_i \\ \vec{b}_j \equiv -s(\psi) \cdot \vec{a}_i + c(\psi) \cdot \vec{b}_i \\ \vec{a}_j \equiv c(\psi) \cdot \vec{a}_i + s(\psi) \cdot \vec{b}_i \end{array} \right.$$

$$\text{Rot}(z, \psi) = {}^i T_j = \begin{bmatrix} c(\psi) & -s(\psi) & 0 & 0 \\ s(\psi) & c(\psi) & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



I.2.3) Propriétés de la matrice de transformation homogène : la matrice de transformation homogène peut s'écrire comme suit ;

$${}^i T_j = \begin{bmatrix} A & P & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad A : \text{la matrice de rotation}$$

- L'inverse de la matrice A est sa transposée ; $A^{-1} = A^T$
- L'inverse de la matrice ${}^i T_j$ est la matrice homogène ${}^j T_i$: ${}^i T_j^{-1} = {}^j T_i$
- rotations successives autour du même axe :

$$\text{Rot}(x, \theta_1) * \text{Rot}(x, \theta_2) = \text{rot}(x, \theta_1 + \theta_2)$$

○ **Multiplication à gauche:**

Soit la repère R_0 . Faisons lui subir K transformations pour l'amener au repère R_k . chaque transformation est définie par rapport au repère R_{k-1} .

La matrice de transformation homogène qui réalise cette opération est définie par :

$${}^0 T_k = {}^0 T_1 \cdot {}^1 T_2 \dots \dots \dots {}^{k-1} T_k$$

○ **Multiplication à droite :**

Soit les repères R_i et R_j . R_j est défini dans R_i par la transformation ${}^j T_i$.

Faisons subir au repère R_j une transformation donnée par T par rapport au repère R_i . Notons le repère résultant par R_j .

La matrice de transformation homogène qui réalise cette opération est défini comme suit : ${}^i T_j = T \cdot {}^i T_j$.

Remarque :

La multiplication à gauche est utilisée lorsque la transformation se fait par rapport au repère courant R_j .

La multiplication à droite est utilisée lorsque la transformation se fait par rapport au repère de référence R_i .

I.3) Représentation de l'orientation d'un solide dans l'espace :

Pour décrire l'orientation d'un solide, plusieurs méthodes sont disponibles. Généralement, on utilise des cosinus directeurs.

Les cosinus directeurs incomplets :

$$R = \begin{pmatrix} ax & bx & cx \\ ay & by & cy \\ az & bz & cz \end{pmatrix}$$

Les éléments de cette matrice sont dits cosinus. Les colonnes de R sont orthogonales entre elles, par conséquent, la connaissance de deux colonnes suffit. En robotique, on met la deuxième colonne. Les six éléments restants sont appelés cosinus directeurs incomplets, ils sont liés entre eux par trois relations :

✓ Les colonnes de R forment une base orthonormée, donc les deux colonnes restants sont orthogonales : $a_x c_x + a_y c_y + a_z c_z = 0$

✓ Les deux colonnes restant sont de norme unité :

$$a_x^2 + a_y^2 + a_z^2 = 1$$

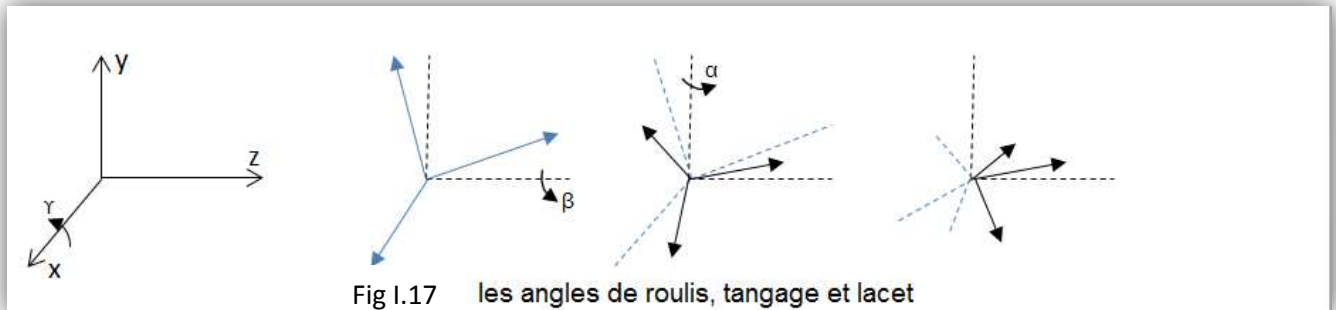
$$c_x^2 + c_y^2 + c_z^2 = 1$$

Ainsi pour décrire l'orientation de l'organe terminal, il faut donc conserver ces six paramètres ou utiliser un autre mode de représentation de l'orientation.

I.3.1) Les angles roulis, tangage et lacet :

Ces angles sont très utilisés en robotique industrielle, ils portent les noms de roll, pitch et yaw en anglais. Il s'agit en fait d'angles d'Euler non classiques, les rotations s'effectuent autour d'axes fixes.

Les angles α , β et γ sont respectivement désignées sous les noms d'angles de roulis, tangage et lacet.



Chaque nouvelle rotation effectuée par rapport à un axe du repère fixe R :

$$R = R(z, \alpha) \cdot R(x, \gamma) \cdot R(y, \beta)$$

Après développement, on aura :

$$R = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 & 0 \\ \sin\alpha & \cos\alpha & 0 & 0 \\ 0 & 0 & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & 0 & 0 \\ \cos\gamma & -\sin\gamma & 0 & 0 \\ \sin\gamma & \cos\gamma & 1 & 0 \end{pmatrix} \cdot \begin{pmatrix} \cos\beta & -\sin\beta & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \sin\beta & \cos\beta & 1 & 0 \end{pmatrix}$$

$$R = \begin{pmatrix} \cos\alpha \cos\beta & -\sin\alpha \cos\gamma + \sin\beta \cos\alpha \sin\gamma & \sin\alpha \sin\gamma + \cos\alpha \sin\beta \sin\gamma \\ \sin\alpha \cos\beta & \cos\alpha \cos\gamma + \sin\alpha \sin\beta \sin\gamma & -\cos\alpha \sin\gamma + \sin\beta \sin\alpha \cos\gamma \\ -\sin\beta & \sin\gamma \cos\beta & \cos\beta \cos\gamma \end{pmatrix}$$

La transformation inverse permet d'établir les angles de roulis, tangage et lacet à partir des cosinus directeurs :

- Si $\beta \neq \frac{\pi}{2}$:
 - $\alpha = \text{atan}^2(a_y, a_x)$
 - $\beta = \text{atan}^2(-a_z, \sqrt{a_x^2 + a_y^2})$
 - $\gamma = \text{atan}^2(b_z, c_z)$
- Si $\beta = \frac{\pi}{2}$:
 - $\alpha - \text{signe}(\beta) \gamma = \text{atan}^2(c_y, c_x)$

Donc γ et α sont indéterminés.

I.4) Modèle géométrique direct du robot (ARM AL5C) :

I.4.1) Définition :

Le calcul du modèle géométrique direct consiste à exprimer le vecteur des coordonnées opérationnelles **X** du robot (situation de l'organe terminal) en fonction du vecteur des coordonnées articulaires (généralisées) **q**.

$$X = f(q) \quad \text{tel que : } q \text{ est de dimension } n \quad X \text{ est de dimension } m$$

I.4.2) Configuration et situation d'un bras manipulateur :

De manière classique, la configuration d'un système mécanique est connue quand la position de tous ses points dans R_0 est connue. Pour un bras manipulateur, elle est définie par un vecteur q de n coordonnées indépendantes appelées coordonnées généralisées et elle est égale au nombre d'articulations motorisées.

La configuration est alors naturellement définie sur un espace N dont la dimension n est appelée indice de mobilité. N est appelé espace des configurations. Les coordonnées généralisées correspondent aux grandeurs caractéristiques des différentes articulations : angles de rotation pour les liaisons rotoïdes, translation pour les liaisons prismatiques.

On note $q = (q_1 \ q_2 \ \dots \ \dots \ q_n)$.

La situation X de l'organe terminal (OT) du bras manipulateur est alors définie par m coordonnées indépendantes dites coordonnées opérationnelles, qui donnent la position et l'orientation de OT dans R_0 . On définira généralement la situation en fonction de la tâche à accomplir : par exemple, on pourra ne considérer que la position de l'OT et non plus son orientation. Dans tous les cas, la situation de l'OT est définie sur un espace M , de dimension $m \leq 6$, appelé espace opérationnel.

On note $x = (x_1 \ x_2 \ \dots \ x_m)^T$

Le choix le plus commun pour les paramètres de position est celui des coordonnées cartésiennes. L'expression de l'orientation de l'OT dépend du choix des paramètres angulaires.

Dans le cas du robot ARM AL5C, le nombre d'articulations motorisées est $n=5$.

Le poignet est de type RR avec deux axes concourants, donc l'organe terminal ne peut pas faire des rotations qu'autour des axes z et x de son repère.

Par conséquent, l'orientation de l'organe terminal est décrite seulement par les angles α et γ .

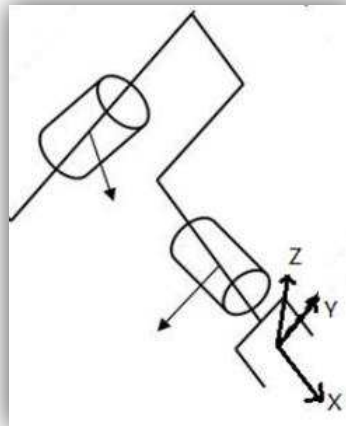


Fig I.18 l'orientation de l'organe terminal

D'où la dimension de l'espace opérationnel $m=5$.

I.4.3) Paramètres de Danavit-Hartenberg modifiés :

Le robot étudié dans ce projet est à structure ouverte simple, ayant cinq articulations rotides. Pour déterminer les paramètres géométriques du robot, utilisons la méthode de Khalil (méthode de Danavit-Hartenberg modifiée) qui impose les conventions suivantes :

❖ Le robot est composé de six corps noté : c_0 c_5 avec c_0 désigne la base du robot et c_5 désigne le corps porteur de l'organe terminal.

❖ L'articulation j connecte le corps c_{j-1} au corps c_j .

❖ Les corps sont rigides (les articulations du robot sont supposées idéales).

❖ La variable de l'articulation j est notée q_j .

❖ On associe au corps c_j le repère R_j qui est défini comme suit :

▪ l'axe z_j est porté par l'axe de l'articulation j

▪ l'axe x_j est porté par la perpendiculaire commune aux axes z_{j-1} et z_j

▪ si les axes z_{j-1} et z_j sont parallèles ou colinéaires le choix de l'axe x_j se fait de telle manière à simplifier notre système et rendre r_j nul.

❖ On prend le repère R_1 confondu avec le repère de référence R_0 correspondant à la base du robot.

❖ On prend de même l'axe x_5 du repère opérationnel R_5 (correspondant à l'organe terminal du robot) colinéaire à l'axe x_4 .

❖ On place les axes z_j portés par les axes des articulations puis les axes x_j .

❖ Le passage d'un repère R_{j-1} au repère R_j est défini par quatre paramètres :

▪ a_j : angle entre les axes z_{j-1} et z_j correspondant à une rotation autour de x_{j-1}

▪ l_j : distance entre z_{j-1} et z_j le long de x_{j-1}

▪ θ_j : angle entre x_{j-1} et x_j correspondant à une rotation autour de z_j

▪ r_j : distance entre x_{j-1} et x_j le long de z_j

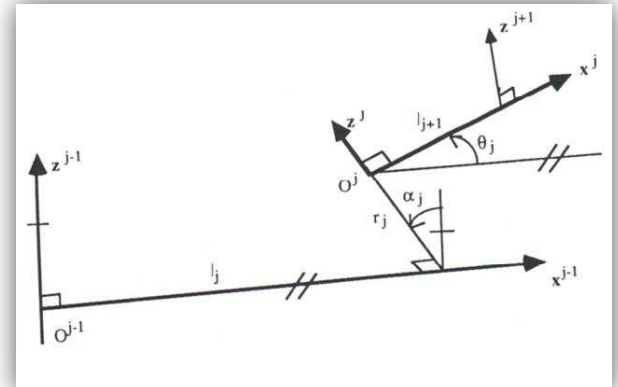


Fig I.18 Paramètres de Danavit-Hartenberg modifiés

Comme la variable articulaire q_j est soit θ_j soit r_j , on note :

$$q_j = \overline{\delta_j} \cdot \theta_j + \delta_j \cdot r_j$$

Avec $\delta = 0$ si l'articulation j est rotoïde ;

$\delta = 1$ si l'articulation j est prismatique

Dans notre cas toutes les articulations sont rotoïdes, donc, $q_j = \theta_j$ pour j allant de 1 à 5

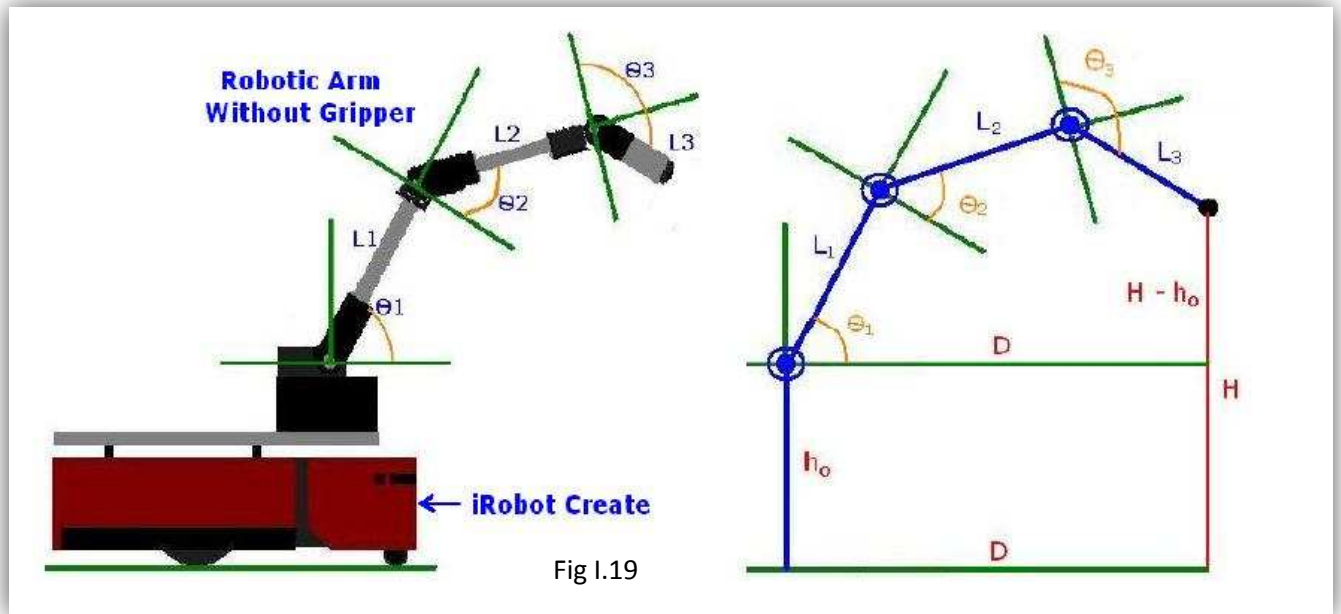


Fig I.19

Le tableau suivant représente tous les paramètres géométriques du robot :

J	δ_i	α_i	θ_i	l_i	r_i
1	0	0	θ_1	0	0
2	0	$\pi/2$	$\theta_2 + \pi/2$	0	0
3	0	0	$\theta_3 + \pi/2$	l_1	0
4	0	0	θ_4	l_2	0
5	0	$-\pi/2$	θ_5	0	0

I.4.5) Calcul du modèle géométrique direct :

Les coordonnées opérationnelles décrivent l'orientation et la position de l'organe terminal et elles sont notées :

$$X = [x_1 \ x_2 \ x_3 \ x_4 \ x_5]^T$$

Avec

- x_1, x_2 et x_3 sont les coordonnées cartésiennes de l'origine du repère opérationnel.
- x_4 et x_5 sont l'orientation du repère opérationnel donnée par les angles de roulis et lacet γ et α .

Utilisons la matrice de transformation homogène 0T_5 pour définir le vecteur des coordonnées opérationnelles X en fonction des variables articulaires θ_j :

$${}^0T_5 = {}^0T_1 \cdot {}^1T_2 \cdot {}^2T_3 \cdot {}^3T_4 \cdot {}^4T_5$$

Calcul des matrices de transformations élémentaires :

$$\text{Rot}(z, \theta_1) = {}^0T_1 = \begin{bmatrix} c(\theta_1) & -s(\theta_1) & 0 & 0 \\ s(\theta_1) & c(\theta_1) & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^1T_2 = \text{Rot}(x, \pi/2) \cdot \text{Rot}(z, \theta_2 + \pi/2)$$

Sachant que :

- $\cos(\theta + \pi/2) = -\sin(\theta)$
- $\sin(\theta + \pi/2) = \cos(\theta)$

$${}^1T_2 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c(\pi/2) & -s(\pi/2) & 0 \\ 0 & s(\pi/2) & c(\pi/2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} -s(\theta_2) & -c(\theta_2) & 0 & 0 \\ c(\theta_2) & -s(\theta_2) & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} -s(\theta_2) & -c(\theta_2) & 0 & 0 \\ 0 & 0 & -1 & 0 \\ c(\theta_2) & -s(\theta_2) & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^2T_3 = \text{trans}(x, l_1) \cdot \text{Rot}(z, \theta_3 - \pi/2)$$

Sachant que :

- $\cos(\theta - \pi/2) = \sin(\theta)$
- $\sin(\theta - \pi/2) = -\cos(\theta)$

$${}^2T_3 = \begin{bmatrix} 1 & 0 & 0 & l_1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} s(\theta_3) & c(\theta_3) & 0 & 0 \\ -c(\theta_3) & s(\theta_3) & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} s(\theta_3) & c(\theta_3) & 0 & l_1 \\ -c(\theta_3) & s(\theta_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^3T_4 = \text{trans}(x, l_2) \cdot \text{Rot}(z, \theta_4)$$

$${}^3T_4 = \begin{bmatrix} 1 & 0 & 0 & l_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} c(\theta_4) & -s(\theta_4) & 0 & 0 \\ s(\theta_4) & c(\theta_4) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c(\theta_4) & -s(\theta_4) & 0 & l_2 \\ s(\theta_4) & c(\theta_4) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$${}^4T_5 = \text{trans}(x, \pi/2) \cdot \text{Rot}(z, \theta_5)$$

$${}^4T_5 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & c(-\pi/2) & -s(-\pi/2) & 0 \\ 0 & s(-\pi/2) & c(-\pi/2) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} c(\theta_5) & -s(\theta_5) & 0 & 0 \\ s(\theta_5) & c(\theta_5) & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} c(\theta_5) & -c(\theta_5) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -s(\theta_5) & -c(\theta_5) & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Calcul de la matrice 0T_5 en partant de 4T_5 :

Notons la matrice ${}^{j-1}T_j$ par $\mathbf{u}_{j-1}$.

$$\mathbf{u}_4 = {}^4T_5$$

$$\mathbf{u}_3 = {}^3T_5 = {}^3T_4 \cdot {}^4T_5 = \begin{bmatrix} c(\theta_4)c(\theta_5) & -c(\theta_4)s(\theta_5) & -s(\theta_4) & l_2 \\ s(\theta_4)c(\theta_5) & -s(\theta_4)s(\theta_5) & c(\theta_4) & 0 \\ -s(\theta_5) & -c(\theta_5) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{u}_2 = {}^2T_5 = {}^2T_3 \cdot {}^3T_5 = \begin{bmatrix} s(\theta_3) & c(\theta_3) & 0 & l_1 \\ -c(\theta_3) & s(\theta_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} c(\theta_4)c(\theta_5) & -c(\theta_4)s(\theta_5) & -s(\theta_4) & l_2 \\ s(\theta_4)c(\theta_5) & -s(\theta_4)s(\theta_5) & c(\theta_4) & 0 \\ -s(\theta_5) & -c(\theta_5) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Sachant que :

- $\cos(\theta_1 + \theta_2) = \cos(\theta_1) \cdot \cos(\theta_2) - \sin(\theta_1) \cdot \sin(\theta_2)$
- $\sin(\theta_1 + \theta_2) = \sin(\theta_1) \cdot \cos(\theta_2) + \cos(\theta_1) \cdot \sin(\theta_2)$

$$\mathbf{u}_2 = \begin{bmatrix} s(\theta_4 + \theta_3)c(\theta_5) & -s(\theta_4 + \theta_3)s(\theta_5) & c(\theta_4 + \theta_3) & s(\theta_3)l_2 + l_1 \\ -c(\theta_4 + \theta_3) & c(\theta_4 + \theta_3)s(\theta_5) & s(\theta_4 + \theta_3) & -c(\theta_3)l_2 \\ -s(\theta_5) & -c(\theta_5) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{u}_1 = {}^1\mathbf{T}_5 = {}^1\mathbf{T}_2 \cdot {}^2\mathbf{T}_5$$

Les composantes de la matrice $\mathbf{u}_1$ sont :

$$a_x = c(\theta_2 + \theta_3 + \theta_4) c(\theta_5)$$

$$a_y = s(\theta_5)$$

$$a_z = s(\theta_2 + \theta_3 + \theta_4) c(\theta_5)$$

$$b_x = -c(\theta_2 + \theta_3 + \theta_4) s(\theta_5)$$

$$b_y = c(\theta_5)$$

$$b_z = -s(\theta_2 + \theta_3 + \theta_4) s(\theta_5)$$

$$c_x = -s(\theta_2 + \theta_3 + \theta_4)$$

$$c_y = 0$$

$$c_z = c(\theta_2 + \theta_3 + \theta_4)$$

$$p_x = c(\theta_2 + \theta_3) \cdot l_2 - s(\theta_2) \cdot l_1$$

$$p_y = 0$$

$$p_z = s(\theta_2 + \theta_3) \cdot l_2 + c(\theta_2) \cdot l_1$$

$$\mathbf{u}_0 = {}^0\mathbf{T}_5 = {}^0\mathbf{T}_1 \cdot {}^1\mathbf{T}_5$$

Les composantes de la matrice $\mathbf{u}_0$ sont :

$$a_x = c(\theta_1) \cdot c(\theta_2 + \theta_3 + \theta_4) c(\theta_5) - s(\theta_1) \cdot s(\theta_5)$$

$$a_y = s(\theta_1) \cdot c(\theta_2 + \theta_3 + \theta_4) c(\theta_5) - c(\theta_1) \cdot s(\theta_5)$$

$$a_z = s(\theta_2 + \theta_3 + \theta_4) c(\theta_5)$$

$$b_x = -c(\theta_1) \cdot c(\theta_2 + \theta_3 + \theta_4) c(\theta_5) - s(\theta_1) \cdot s(\theta_5)$$

$$b_y = -s(\theta_1) \cdot c(\theta_2 + \theta_3 + \theta_4) c(\theta_5) + c(\theta_1) \cdot s(\theta_5)$$

$$b_z = -s(\theta_2 + \theta_3 + \theta_4) s(\theta_5)$$

$$c_x = -c(\theta_1) \cdot s(\theta_2 + \theta_3 + \theta_4)$$

$$c_y = -s(\theta_1) \cdot s(\theta_2 + \theta_3 + \theta_4)$$

$$c_z = c(\theta_2 + \theta_3 + \theta_4)$$

$$P_x = c(\theta_1) \cdot [c(\theta_2 + \theta_3) \cdot l_2 - s(\theta_2) \cdot l_1]$$

$$P_y = s(\theta_1) \cdot [c(\theta_2 + \theta_3) \cdot l_2 - s(\theta_2) \cdot l_1]$$

$$P_z = s(\theta_2 + \theta_3) \cdot l_2 + c(\theta_2) \cdot l_1$$

La position du repère R_5 avec les cosinus directeurs incomplets est :

$$\left\{ \begin{array}{l} \mathbf{a}_x = \mathbf{x}_4 \\ \mathbf{a}_y = \mathbf{x}_5 \\ \mathbf{a}_z = \mathbf{x}_6 \\ \mathbf{c}_x = \mathbf{x}_7 \\ \mathbf{c}_y = \mathbf{x}_8 \\ \mathbf{c}_z = \mathbf{x}_9 \end{array} \right.$$

L'orientation du repère R_5 avec les angles de roulis et lacet:

$$\left\{ \begin{array}{l} \gamma = x_4 = \text{atan2}(b_z, c_z) \\ \alpha = x_5 = \text{atan2}(a_y, a_x) \end{array} \right.$$

Le vecteur des coordonnées opérationnelles $\mathbf{X}$ est donné par : $\mathbf{X} = [x_1 \ x_2 \ x_3 \ x_4 \ x_5]^T$

I.5) Modèle géométrique inverse du robot

I.5.1) Définition

Le problème consiste à calculer les coordonnées articulaires du robot à partir des coordonnées opérationnelles.

I.5.2) Solubilité du MGI :

La solubilité du MGI, n'est autre que l'existence d'un nombre fini de solutions trait fondamental en matière de conception. Supposons que la situation x d'un bras manipulateur à n liaisons est exprimée par un nombre m minimal de paramètres. Supposons que par ailleurs que x est une situation accessible avec le bras manipulateur, c'est-à-dire que la situation appartient à l'espace de travail. Alors (dans la majorité des cas) :

- si $n < m$, il n'existe pas de solution pour le MGI
- si $n = m$, il existe un nombre fini de solutions en dehors de certaines configurations, appelées configurations singulières.
- Si $n > m$, il existe une infinité de solutions

Pour calculer le MGI, on a le choix entre plusieurs méthodes, on cite, la méthode de Paul, la méthode de Pieper et la méthode générale de Raghavan et Roth. En ce qui nous concerne, nous avons choisi d'utiliser celle de Paul.

I.5.3) La méthode de Paul : la méthode de Paul consiste à calculer le modèle géométrique inverse (MGI) en résolvant le système d'équations suivant :

$$\mathbf{u}_0 = {}^0\mathbf{T}_1 \cdot {}^1\mathbf{T}_2 \cdot {}^2\mathbf{T}_3 \cdot {}^3\mathbf{T}_4 \cdot {}^4\mathbf{T}_5$$

La procédure est la suivante :

Pré multiplier successivement les deux membres de l'équation précédente par le terme ${}^j\mathbf{T}_{j-1}$ avec j variant de 1 à 4 afin d'isoler à chaque fois la variable articulaire q_j .

$$\text{sachant que : } {}^j\mathbf{T}_{j-1} = ({}^{j-1}\mathbf{T}_j)^{-1}$$

pour obtenir les q_j il suffit de résoudre les équations suivantes :

$$\mathbf{u}_0 = {}^0\mathbf{T}_1 \cdot {}^1\mathbf{T}_2 \cdot {}^2\mathbf{T}_3 \cdot {}^3\mathbf{T}_4 \cdot {}^4\mathbf{T}_5$$

$${}^1\mathbf{T}_0 \cdot \mathbf{u}_0 = \mathbf{u}_1 = {}^1\mathbf{T}_2 \cdot {}^2\mathbf{T}_3 \cdot {}^3\mathbf{T}_4 \cdot {}^4\mathbf{T}_5$$

$${}^2\mathbf{T}_1 \cdot \mathbf{u}_1 = \mathbf{u}_2 = {}^2\mathbf{T}_3 \cdot {}^3\mathbf{T}_4 \cdot {}^4\mathbf{T}_5$$

$${}^3\mathbf{T}_2 \cdot \mathbf{u}_2 = \mathbf{u}_3 = {}^3\mathbf{T}_4 \cdot {}^4\mathbf{T}_5$$

$${}^4\mathbf{T}_3 \cdot \mathbf{u}_3 = \mathbf{u}_4 = {}^4\mathbf{T}_5$$

L'application de cette méthode sur plusieurs robots a permis de résumer les types d'équations rencontrées avec la méthode de Paul dans le tableau suivant :

Type 1	$X \cdot r_i = Y$
Type 2	$X \cdot s\theta_i + Y \cdot c\theta_i = Z$
Type 3	$X_1 \cdot s\theta_i + Y_1 \cdot c\theta_i = Z_1$ $X_2 \cdot s\theta_i + Y_2 \cdot c\theta_i = Z_2$
Type 4	$X_1 \cdot r_i \cdot s\theta_i = Y_1$ $X_2 \cdot r_i \cdot s\theta_i = Y_2$
Type 5	$X_1 \cdot s\theta_i = Y_1 + Z_1 \cdot r_j$ $X_2 \cdot s\theta_i = Y_2 + Z_1 \cdot r_j$
Type 6	$w \cdot s\theta_j = X \cdot c\theta_i + Y \cdot s\theta_i + Z_1 \cdot r_j$ $w \cdot c\theta_j = X \cdot s\theta_i - Y \cdot c\theta_i + Z_2 \cdot r_j$
Type 7	$w_1 \cdot c\theta_j + w_2 \cdot s\theta_j = X \cdot c\theta_i + Y \cdot s\theta_i + Z_1$ $w_1 \cdot s\theta_j - w_2 \cdot c\theta_j = X \cdot c\theta_i + Y \cdot s\theta_i + Z_2$
Type 8	$X_1 \cdot s\theta_i + Y_1 \cdot c(\theta_i + \theta_j) = Z_1$ $X_2 \cdot s\theta_i + Y_2 \cdot c(\theta_i + \theta_j) = Z_2$

I.5.4) Calcul du modèle géométrique inverse :

Soit u_0 la situation désirée telle que :

$$u_0 = \begin{bmatrix} a_x & b_x & c_x & p_x \\ a_y & b_y & c_y & p_y \\ a_z & b_z & c_z & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix} = {}^0T_5 \quad \rightarrow \quad \text{On ne peut rien conclure de cette équation.}$$

Pré multiplions l'équation précédente par 1T_0 avec ${}^1T_0 = ({}^0T_1)^{-1}$

$${}^1T_0 = \begin{bmatrix} c(\theta_1) & s(\theta_1) & 0 & 0 \\ -s(\theta_1) & c(\theta_1) & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Les éléments de gauche sont :

$$u(1.1) = a_x \cdot c(\theta_1) + s(\theta_1) \cdot a_y$$

$$u(1.2) = b_x \cdot c(\theta_1) + s(\theta_1) \cdot b_y$$

$$u(1.3) = c_x \cdot c(\theta_1) + s(\theta_1) \cdot c_y$$

$$u(1.4) = p_x \cdot c(\theta_1) + s(\theta_1) \cdot p_y$$

$$u(2.1) = -a_x \cdot s(\theta_1) + c(\theta_1) \cdot a_y$$

$$u(2.2) = -b_x \cdot s(\theta_1) + c(\theta_1) \cdot b_y$$

$$u(2.3) = -c_x \cdot s(\theta_1) + c(\theta_1) \cdot c_y$$

$$u(2.4) = -p_x \cdot s(\theta_1) + c(\theta_1) \cdot p_y$$

$$u(3.1) = a_z$$

$$u(3.2) = b_z$$

$$u(3.3) = c_x$$

$$u(3.4) = p_z$$

Les éléments de droite sont ceux de la matrice 1T_5 calculée précédemment.

Par identification, on obtient :

$$-p_x \cdot s(\theta_1) + c(\theta_1) \cdot p_y = 0$$

Cette équation possède deux solutions:

$$\begin{cases} \theta_1 = \text{atan2}(p_y, p_x) \\ \theta_1' = \theta_1 + \pi \end{cases}$$

En identifiant les éléments u(1.4) et u(3.4), on aura le système suivant :

$$\begin{cases} P_x \cdot c(\theta_1) + s(\theta_1) \cdot p_y = c(\theta_2 + \theta_3) \cdot l_2 - s(\theta_2) \cdot l_1 \\ p_z = s(\theta_2 + \theta_3) \cdot l_2 - c(\theta_2) \cdot l_1 \end{cases} \dots\dots\dots (I)$$

pour simplifier les écritures, posons: $B = P_x \cdot c(\theta_1) + s(\theta_1) \cdot p_y$

$$(I) \Leftrightarrow \begin{cases} (B + s(\theta_2) \cdot l_2)^2 = (c(\theta_2 + \theta_3) \cdot l_2)^2 & \dots\dots\dots(1) \\ (p_z - c(\theta_2) \cdot l_2)^2 = (s(\theta_2 + \theta_3) \cdot l_2)^2 & \dots\dots\dots(2) \end{cases}$$

De (1) et (2), on aura :

$$B^2 + 2 \cdot B \cdot s(\theta_2) \cdot l_1 + l_1^2 + p_z^2 - 2 \cdot p_z \cdot c(\theta_2) \cdot l_1 = l_2^2$$

On obtient une équation en θ_2 de type 2 :

$$x \cdot s(\theta_2) + y \cdot c(\theta_2) = z$$

$$\text{avec } z = (l_2)^2 - (p_z)^2 - (l_1)^2 - (B)^2$$

$$y = -2 \cdot p_z \cdot l_1$$

$$x = 2 \cdot B \cdot l_1$$

La solution de cette équation est donnée comme suit :

$$C(\theta_2) = \frac{z \cdot y - \xi \cdot x \cdot \sqrt{(x^2 + y^2 - z^2)}}{(x^2 + y^2)}$$

$\text{avec } \xi \pm 1$

$$S(\theta_2) = \frac{z \cdot x + \xi \cdot y \cdot \sqrt{(x^2 + y^2 - z^2)}}{(x^2 + y^2)}$$

Donc deux valeurs de θ_2 sont possibles :

$$\theta_2 = \text{atan2}(s(\theta_2), c(\theta_2))$$

En connaissant θ_2 , on peut calculer θ_3 à partir de (I)

$$c(\theta_2 + \theta_3) = [B + s(\theta_2) \cdot l_1] / l_2$$

$$s(\theta_2 + \theta_3) = [p_z - c(\theta_2) \cdot l_1] / l_2$$

$$\theta_2 + \theta_3 = \arctan(s(\theta_2 + \theta_3), c(\theta_2 + \theta_3))$$

$$\theta_3 = \text{atan2}(s(\theta_2 + \theta_3), c(\theta_2 + \theta_3) - \theta_2)$$

A partir de u(1.3) et u(3.3), on obtient :

$$-s(\theta_2 + \theta_3 + \theta_4) = c_x \cdot c(\theta_1) + s(\theta_1) \cdot c_y$$

$$c(\theta_2 + \theta_3 + \theta_4) = c_z$$

c_x , c_y et c_z étant connus, donc en connaissant θ_1 , θ_2 et θ_3 , on peut déduire la valeur de θ_4

$$\theta_2 + \theta_3 + \theta_4 = \arctan(s(\theta_2 + \theta_3 + \theta_4), c(\theta_2 + \theta_3 + \theta_4))$$

$$\theta_4 = \text{atan2}(s(\theta_2 + \theta_3 + \theta_4), c(\theta_2 + \theta_3 + \theta_4) - \theta_2 - \theta_3)$$

enfin, en considérant les éléments u(1.2) et u(2.2) on obtient :

$$s(\theta_5) = -a_x \cdot s(\theta_1) + c(\theta_1) \cdot a_y$$

$$c(\theta_5) = -b_x \cdot s(\theta_1) + c(\theta_1) \cdot b_y$$

$$\theta_5 = \text{atan2}(s(\theta_5), c(\theta_5))$$

Chapitre II : La commande

II.1) Introduction:

Commander le mouvement d'un robot revient à lui spécifier un chemin qu'il doit suivre. Un chemin est une séquence de poses définies soit dans l'espace opérationnel (afin de situer l'organe terminal), soit dans l'espace articulaire du robot (afin d'indiquer les valeurs des paramètres de liaison).

Ces poses peuvent être planifiées soit par :

- Programmation par apprentissage
- Programmation explicite (à l'aide d'un langage de programmation)
- Programmation par CAO (conception assistée par ordinateur)

II.2) La commande des manipulateurs :

La plupart des tâches demandées à un manipulateur impliquent un contact avec l'environnement. où l'environnement réagit sur la structure du bras dès que celui-ci entre en mouvement. Ceci implique de réguler l'effort exercé par le robot. Les commandes en effort constituent donc l'essentiel des publications relatives à la commande des manipulateurs. Ceci est à prendre spécifiquement en compte lors du choix de la structure de commande pour piloter le manipulateur. Nous présentons ci-dessous des lois de commande en effort qui ont été réalisées pour différentes applications robotique.

II.2.1) La commande par retour d'effort :

Une loi de commande classique du type PID articulaire réalise l'asservissement du système :

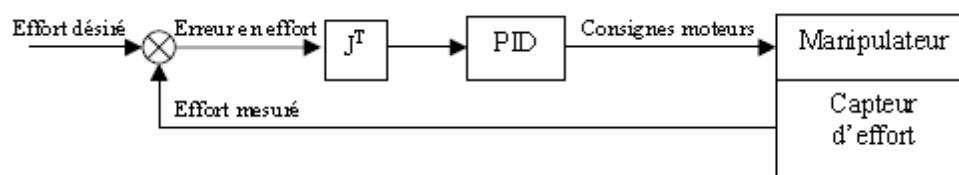


Fig II.1 Commande en effort cartésien, PID articulaire.

II.2.2) La commande adaptative :

D. Broome propose d'utiliser un contrôle adaptatif pour commander un bras manipulateur.

Il fait l'hypothèse que le système se comporte comme un système linéaire durant un intervalle de temps Δt . Il prend comme modèle de référence pour chaque articulation du bras, une fonction du 2^{ème} ordre décrite à l'équation 2.1.

$$(b_1 \cdot z^{-1} + b_2 \cdot z^{-2}) \cdot u_c(k) = (1 + a_1 \cdot z^{-1} + a_2 \cdot z^{-2}) \cdot y(k) + d(k) \quad (2.1)$$

$y(k)$: retour capteur position (angles des articulations),
 $u_c(k)$: signal de commande, consigne des moteurs,
 $d(k)$: perturbation.

Un algorithme d'adaptation par les moindres carrés étendus permet d'identifier les paramètres du modèle à chaque période d'échantillonnage (cf. équation 2.2).

$$\begin{aligned} \theta(k) &= \theta(k-1) + L(k) \cdot [y(k) - \theta^T(k-1) \cdot \phi(k)] \\ L(k) &= \frac{P(k-1) \cdot \phi(k)}{\lambda + \phi^T(k) \cdot P(k-1) \cdot \phi(k)} \\ P(k) &= \lambda^{-1} \cdot (I - L(k) \cdot \phi(k) \cdot \phi^T(k) \cdot P(k-1)) \end{aligned} \quad (2.2)$$

Avec :

$\theta_T(k) = (a_1, a_2, b_1, b_2)$: les paramètres du système,
 $\phi_T(k) = (-y(k-1), -y(k-2), u_c(k-1), u_c(k-2))$: vecteur des entrées/sorties précédentes,

Une fois que l'identification des paramètres est effectuée, les gains optimaux sont calculés par placement des zéros et pôles de la fonction de transfert, pour en assurer la stabilité. Le schéma de commande global est décrit à la figure ci-dessous :

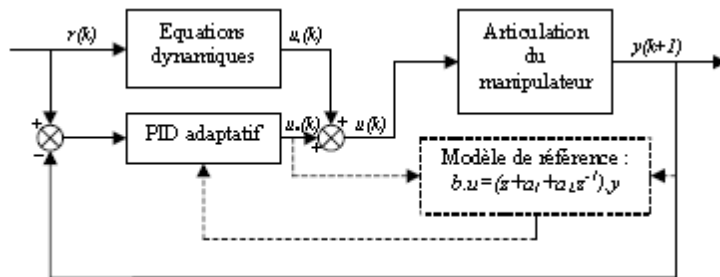


Fig II.2 Principe du PID adaptatif par identification d'un modèle simplifié.

$r(k)$: valeurs désirées,
 $e(k)$: erreur de position.

Les équations dynamiques du système utilisées dans le schéma décrit à la figure 2-IV comprennent uniquement les termes de gravité. Il n'est pas nécessaire de considérer le modèle global. L'équation 2.3 décrit, sous forme discrète, la loi de commande utilisée.

$$u_{ek} = -k_3 \cdot u_{ek-2} + k_0 \cdot (r_k - y_k) + k_1 \cdot (r_{k-1} - y_{k-1}) + k_2 \cdot (r_{k-2} - y_{k-2}) \quad (2.3)$$

r_k : valeurs désirées,
 y_k : données capteurs,
 k_0, k_1, k_2, k_3 : gains du PID.

La fonction de transfert en boucle fermée est écrite grâce aux équations 2.1 et 2.3. Soit le dénominateur de cette fonction qui s'écrit comme à l'équation 2.4.

$$T(z^{-1}) = 1 + t_1 \cdot z^{-1} + t_2 \cdot z^{-2} + t_3 \cdot z^{-3} + t_4 \cdot z^{-4} \quad (2.4)$$

Une relation explicite entre les paramètres du système et les gains de la fonction de correction peut alors être écrite (cf. équation 2.5). C'est en choisissant les paramètres t_1, t_2, t_3 et t_4 de manière à placer les pôles de la fonction de transfert à l'intérieur du cercle unité, que l'on acquiert la valeur adéquate des gains du PID, qui réalisent un asservissement stable.

$$k_0 = \frac{t_1 - a_1}{b_1}, \quad k_3 = \frac{b_1 \cdot b_2 \cdot t_3 - b_2^2 \cdot t_2 - b_1^2 \cdot t_4 + b_2^2 \cdot a_2 + b_2^2 \cdot k_0}{b_1 \cdot b_2 \cdot a_1 - b_2^2 - a_2 \cdot b_1^2}, \quad (2.5)$$

$$k_2 = \frac{t_4 - a_2 \cdot k_3}{b_2}, \quad k_1 = \frac{t_2 - a_2 - b_2 \cdot k_0 - s_3}{b_1}$$

II.2.3) Commande hybride dynamique

De nombreuses recherches ont été menées sur la commande en effort de manipulateurs industriels. La commande hybride dynamique implique une bonne connaissance du modèle dynamique de la

structure que l'on désire commander. Suivant le domaine d'évolution du robot, le modèle dynamique et les paramètres nécessaires à son établissement changent. L'équation 2.6 décrit une forme générique du modèle dynamique exprimé dans l'espace opérationnel d'un bras manipulateur.

$$F_x = M_x(\theta) \ddot{\theta} + V_x(\theta, \dot{\theta}) + G_x(\theta) \quad (II.6)$$

θ : Variable articulaire,

F_x : torseur des efforts et couples opérationnels (effecteurs).

M_x : matrice d'inertie du système, calculée dans l'espace opérationnel.

V_x : vecteur des forces de Coriolis et centrifuges, calculé dans l'espace opérationnel.

G_x : vecteur des forces de gravité, calculé dans l'espace opérationnel.

Le schéma de la commande hybride position/force dynamique est décrit à la figure ci-dessous :

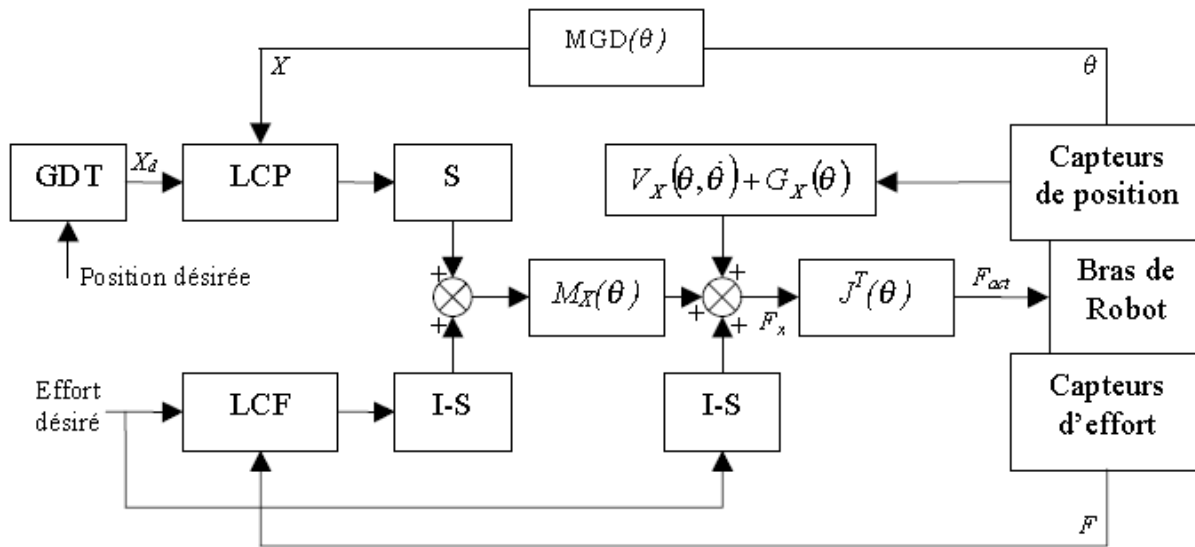


Fig II.3 Schéma de commande hybride dynamique.

LCP : Loi de Commande en Position,

LCF : Loi de Commande en Effort,

MGD : Modèle Géométrique Direct,

F_{act} : torseur des efforts et couples des actionneurs.

Cette loi de commande régule les consignes articulaires en fonction des erreurs de position et d'effort de l'effecteur, en effectuant un découplage non linéaire dans l'espace opérationnel. Chaque actionneur du robot reçoit simultanément des contributions à l'asservissement de position et à l'asservissement d'effort dans l'espace opérationnel, via la matrice jacobienne transposée. On utilise la matrice de sélection S pour désigner le type de commande (position ou effort) associée à chaque degré de liberté de l'effecteur.

II.2.4) Commande hybride externe :

La commande hybride externe propose elle aussi de piloter un bras manipulateur en effort et en position. Elle consiste en une hiérarchisation de la boucle de commande en effort sur la boucle de commande en position. Le principe de cette commande est de modéliser le contact avec l'environnement par un système ressort-amortisseur dont les paramètres (raideur et amortissement) sont implicitement inclus dans les gains de correction de la LCF (Loi de Commande en Force). Lorsque le robot est au contact, une variation de l'effort se traduit par un déplacement. Une erreur sur la consigne en effort peut donc être vue comme un déplacement à effectuer. De ce fait, cette information peut être utilisée pour provoquer un déplacement, en corrigeant la consigne initiale de position. Ceci implique un comportement du bras qui tend à asservir sa position à suivre une consigne qui rectifie automatiquement l'asservissement de l'effort, pour générer un contact dans les conditions désirées. Le schéma de cette structure de commande est présenté à la figure ci-dessous :

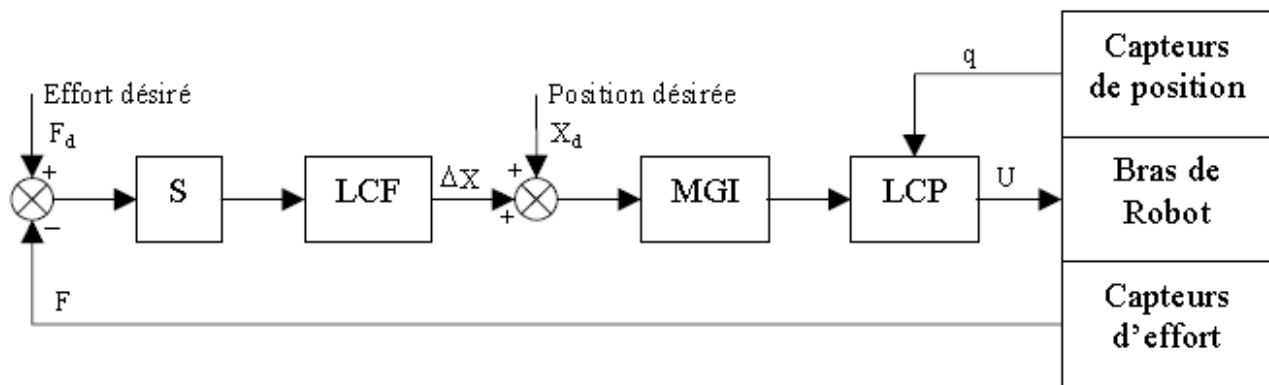


Fig II.4 Schéma de commande hybride externe, avec asservissement articulaire de la position.

LCP : Loi de Commande en Position.

LCF : Loi de Commande en Effort.

MGI : Modèle Géométrique Inverse.

U : consignes moteurs.

q : variables articulaires.

La consigne de déplacement ΔX vient s'ajouter à la consigne de position X_d . Il est à noter que la boucle interne de contrôle de la position est toujours active, même en cas de rupture de contact. Il existe cependant une condition pour que l'asservissement en effort ne soit pas affecté par la boucle de contrôle de la position. Si nous considérons une direction dans laquelle l'asservissement en effort est actif, c'est-à-dire suivant laquelle le déplacement est contraint, la consigne de position X_d peut être vue comme une perturbation extérieure constante. Or, pour une loi de commande linéaire de l'effort (ici LCF), le rejet d'une perturbation constante peut être effectué si l'asservissement contient au moins un intégrateur en amont de la perturbation. Si ce n'est pas le cas, une erreur statique apparaît.

II.2.5) Loi de commande d'impédance

Les structures de commande d'impédance ont pour principe non pas la commande directe de l'effort ou de la position, mais celle de la relation entre ces deux grandeurs :

L'impédance associée. La relation d'impédance choisie est généralement du second ordre. Elle est décrite à l'équation 2.7 :

$$F = K_a (\ddot{X}_d - \ddot{X}) + K_v (\dot{X}_d - \dot{X}) + K_p (X_d - X) \quad (2.7)$$

F : effort exercé à l'extrémité de l'effecteur,

X : position de l'effecteur.

X_d : position désirée de l'effecteur.

K_a : matrice d'inertie.

K_v : matrice d'amortissement.

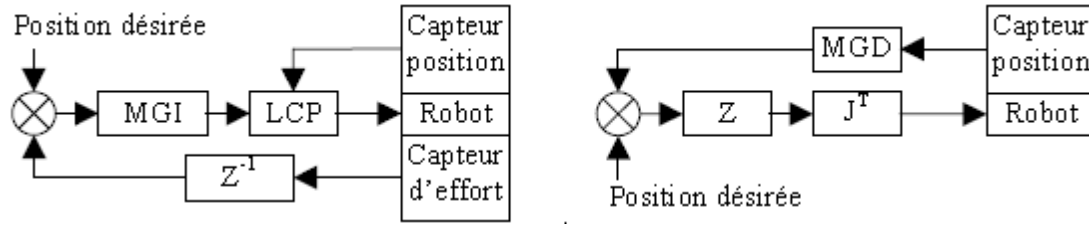
K_p : matrice de raideur.

Ou encore, dans le domaine de Laplace :

$$Z(p) = K_a \cdot p^2 + K_v \cdot p + K_p \quad (2.8)$$

Il existe deux manières de mettre en œuvre la commande d'impédance.

- La première consiste à mesurer l'effort d'interaction entre l'effecteur et l'environnement, et cette mesure est utilisée pour modifier la consigne de position, compte tenu de l'impédance souhaitée.
- La seconde utilise une mesure de la position et en déduit l'effort à exercer sur l'environnement, à partir duquel est calculé l'effort que doivent fournir les actionneurs



commande d'impédance, basée sur la position.

commande d'impédance, basée sur l'effort.

Fig 11.5

Les commandes par matrice de raideur ($K_a = K_v = 0$) ou d'amortissement ($K_a = K_p = 0$) sont des cas particuliers de la commande d'impédance.

II.3) Choix et application de la loi de commande à notre système :

II.3.1) Loi de commande en Position LCP :

Génération de mouvement dans l'espace articulaire :

Les mouvements (A,B, C, D, E et F) du bras sont tous en mode trapézoïdal, ainsi la loi de commande utilisée pour chaque axe de mouvement est de type trapèze.

II.3.1.1) Définition de la loi trapèze :

Cette loi a pour objectif d'assurer une continuité en vitesse avec un temps de parcours optimal, le mouvement de l'articulation j est représenté par les équations suivantes :

$$\left\{ \begin{array}{ll} q_j(t) = q_j^i + \frac{1}{2}t^2 \cdot k_{aj} \cdot \sin g(D_j) & \text{pour } 0 \leq t \leq \tau_j \\ q_j(t) = q_j^i + (t - \frac{\tau_j}{2})t^2 \cdot k_{vj} \cdot \sin g(D_j) & \text{pour } \tau_j \leq t \leq t_{fj} - \tau_j \\ q_j(t) = q_j^f - \frac{1}{2}(t_{fj} - t)^2 \cdot k_{aj} \cdot \sin g(D_j) & \text{pour } t_{fj} - \tau_j \leq t \leq t_{fj} \end{array} \right.$$

avec : q_j^i : position initiale

q_j^f : position finale

$$D_j = q_j^f - q_j^i$$

k_{vj} est l'accélération articulaire maximale de l'articulation j, calculée à partir du rapport de couple moteur maximal à l'inertie maximale vue par l'articulation j.

$$t_{fj} = \text{temps final} = 2 \cdot \sqrt{\frac{|D_j|}{K_{aj}}}$$

$$\tau_j = \frac{K_{vj}}{K_{aj}}$$

Les figures suivantes nous illustrent l'évolution de la position, la vitesse et de l'accélération en fonction du temps dans le cas de la loi trapèze :

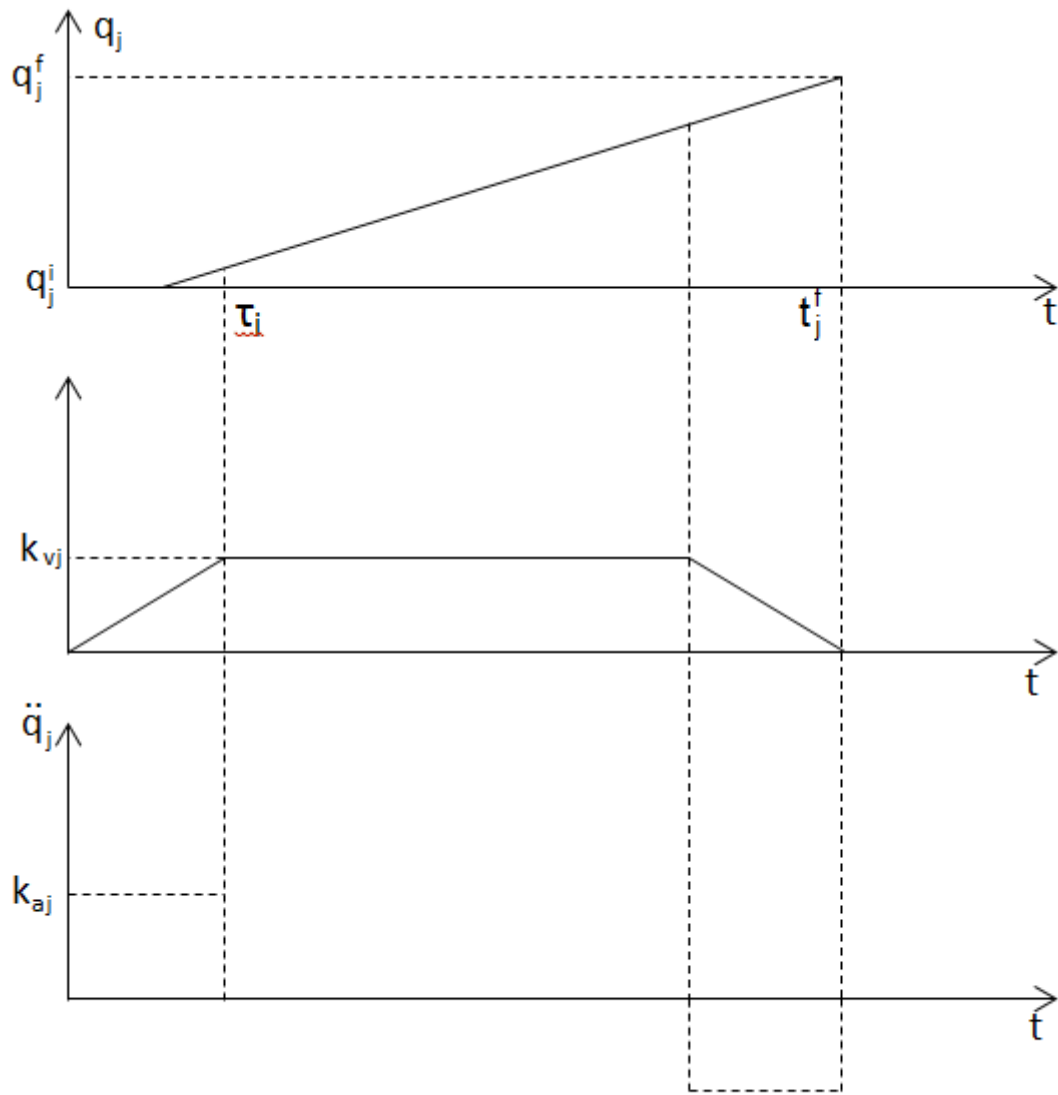


Fig II.6 Evolution de la position, de la vitesse et de l'accélération en fonction du temps avec la loi trapèze

II.3.1.3) Loi de synchronisation : Problème :

Pour effectuer une tâche coordonnée, le mouvement des différentes articulations doit être synchronisé.

Principe de la loi de synchronisation :

Soit L_i la durée d'évolution de la $i^{\text{ème}}$ articulation, soit L le plus grand des L_i .

Pour synchroniser les évolutions des cinq articulations, il suffit d'agir de telle façon que la durée d'évolution de chaque articulation soit L . les équations précédentes deviennent :

$$\left\{ \begin{array}{ll} q_j(t) = q_j^i + \frac{1}{2} (t \cdot L_i / L)^2 \cdot k_{aj} \cdot \sin g(D_j) & \text{pour } 0 \leq t \leq (\tau_j \cdot L / L_i) \\ q_j(t) = q_j^i + (t L_i / L - \frac{(\tau_j \cdot L_i / 2L)}{2}) t^2 \cdot k_{vj} \cdot \sin g(D_j) & \text{pour } (\tau_j \cdot L / L_i) \leq t \leq (t_{fj} - \tau_j) \cdot L / L_i \\ q_j(t) = q_j^f - \frac{1}{2} (t_{fj} - t)^2 \cdot k_{aj} \cdot \sin g(D_j) & \text{pour } (t_{fj} - \tau_j) \cdot L / L_i \leq t \leq t_{fj} \cdot L / L_i \end{array} \right.$$

II.3.1.3) Planification par une loi de commande en position « cas non redondant » :

Principe :

Cette méthode consiste à spécifier les coordonnées de l'organe terminal aux sommets de la tâche, puis exploiter le modèle géométrique élaboré précédemment afin d'effectuer un changement de coordonnées et de trouver les variables articulaires θ_i qui réalisent la tâche désirée.

Organigramme de la loi de commande en position :

La figure suivante présente l'organigramme du programme de changement de coordonnées pour les deux solutions du modèle géométrique.

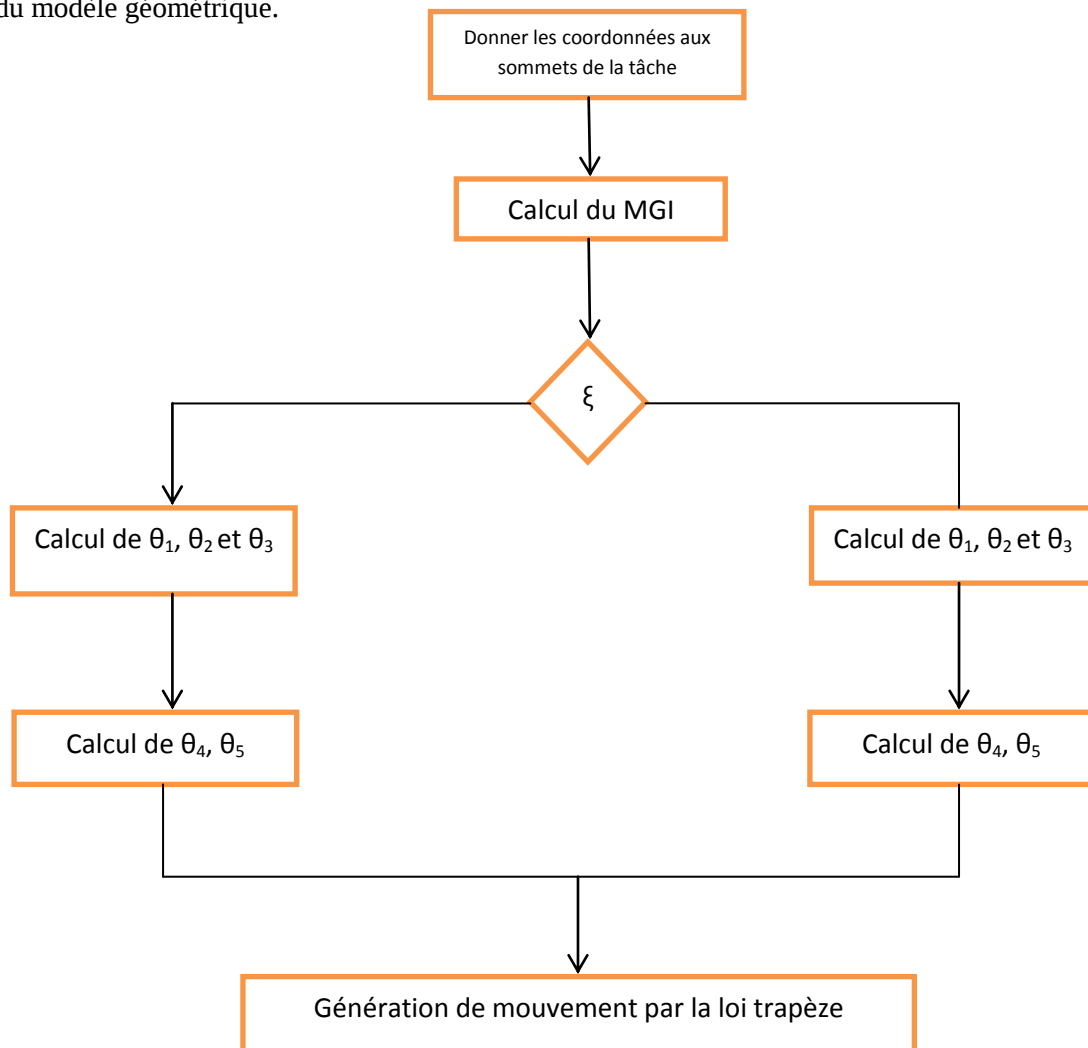


Fig II.7 l'organigramme du programme de changement de coordonnées pour les deux solutions du modèle géométrique

II.3.2) Loi de commande en Force LCF :

Considérée notre bras Manipulateur ayant une base fixe, Nous pouvons avoir une estimation de l'effort exercé par l'effecteur sur l'environnement. La lecture de l'effort de liaison permet d'asservir les actionneurs du bras de telle façon qu'il produise l'effort désiré sur l'environnement. Nous faisons l'hypothèse qu'il est au contact et en bonne position au début du test.

II.3.3) Réalisation d'une Commande généralisée effort – compensation :

Nous pouvons établir une structure de commande nous permettant d'assurer sur un même robot et avec une même loi de commande, la compensation des perturbations issues du bras lorsqu'il effectue un mouvement en espace libre, et le contrôle de la production d'un effort sur l'environnement.

Nous reprenons les schémas de commande précédemment exposés et les modifions en rajoutant une variable telle que :

- $\alpha = 1$ dans le cas de la commande en effort,
- $\alpha = 0$ dans le cas de la compensation.

Schéma de commande :

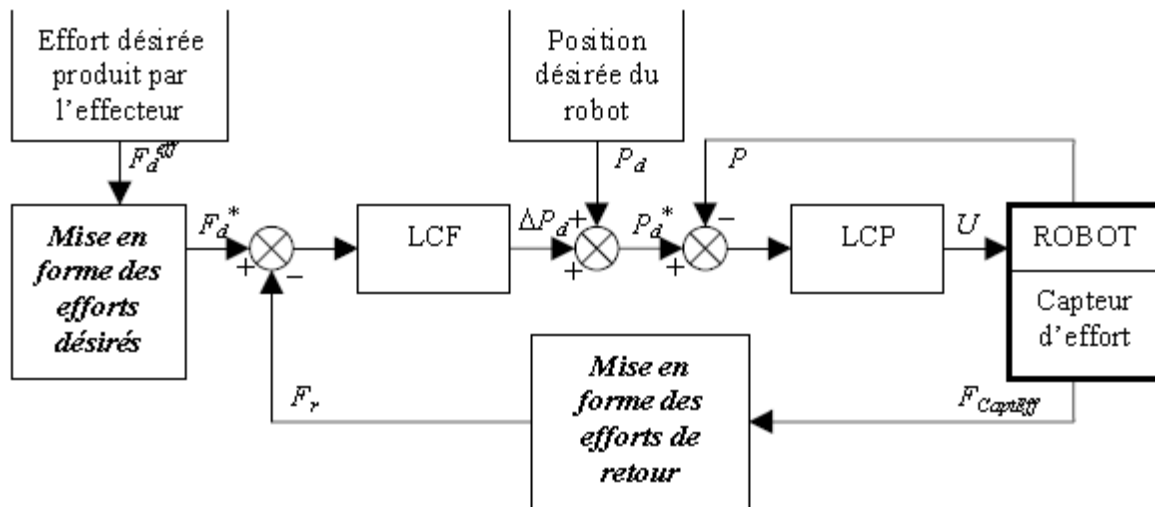


Fig II.8 Schéma de la loi de commande généralisée effort / compensation

Les blocs 'Mise en forme des efforts désirés' et 'Mise en forme des efforts de retour' sont détaillés à la figure ci-dessous :

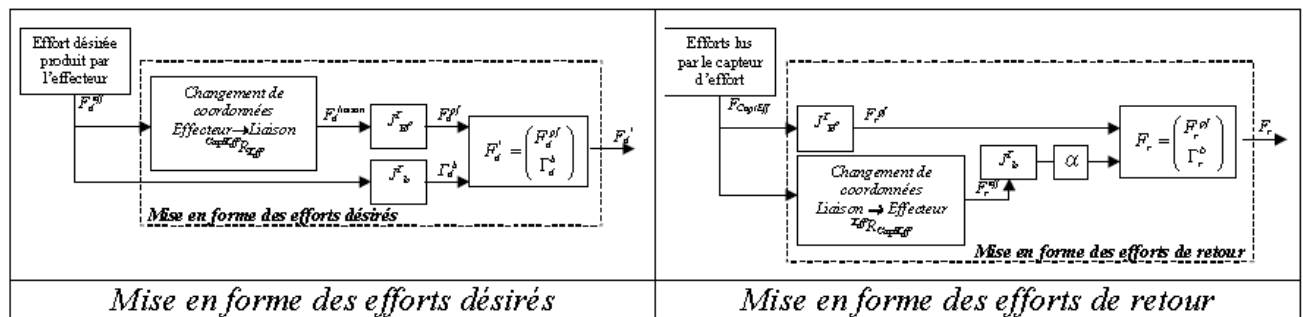


Fig II.9 Descriptif des blocs de mise en forme des efforts de retour et désirés.

Le changement de coordonnées effecteur $\rightarrow$ liaison est la matrice de rotation $R_{Capt.Eff}$ permettant d'exprimer les efforts désirés de l'effecteur dans le repère attaché à la liaison entre le bras et la plateforme.

Le changement de coordonnées liaison $\Rightarrow$ effecteur est la matrice de rotation $R_{Eff.Capt}$.

Soient P_d la position désirée de tous les organes du robot et P la position courante de ces éléments. P_d et P sont des vecteurs de dimension 5.

$$P_d = \begin{pmatrix} \eta_p^d & q_b^d \end{pmatrix}^T$$

$$P = \begin{pmatrix} \eta_p & q_b \end{pmatrix}^T$$

Soit F_{dEff} l'effort désiré produit par l'effecteur. F_{dEff} est un vecteur de dimension 2.

$$F_d^{Eff} = \begin{pmatrix} F_{dx}^{Eff} & F_{dz}^{Eff} \end{pmatrix}^T$$

Soit F_d^* l'effort désiré produit par les actionneurs du robot. Le vecteur F_d^* est de dimension 5 et se compose comme décrit à l'équation suivante :

$$F_d^* = \begin{pmatrix} F_d^{pf} \\ \Gamma_d^b \end{pmatrix} = \begin{pmatrix} J_{pf}^T \cdot \left({}^{Capt.Eff} R_{Eff} \cdot F_d^{Eff} \right) \\ J_b^T \cdot F_d^{Eff} \end{pmatrix}$$

JpfT: matrice jacobienne transposée de la plate-forme, de l'espace articulaire de la plate-forme à l'espace opérationnel de la plate-forme, le repère de la liaison entre le bras et la plate-forme. La dimension de JpfT est 2x3.

JbT: matrice jacobienne transposée du bras, de l'espace articulaire du bras à l'espace opérationnel du bras, le repère de l'effecteur. La dimension de JbT est 2x2.

Soit F_r l'effort de retour, issu du capteur d'effort. Le vecteur F_r est de dimension 5 et se compose comme décrit à l'équation suivante :

$$F_r = \begin{pmatrix} J_{pf}^T \cdot F_{Capt.Eff} & \alpha \cdot J_b^T \cdot \left({}^{Eff} R_{Capt.Eff} \cdot F_{Capt.Eff} \right) \end{pmatrix}^T$$

Le coefficient α permet de sélectionner la commande de la compensation ou la commande en effort. Définissons K_{peff} et K_{ieff} les matrices de gains proportionnels et intégraux des actionneurs de la boucle de commande en effort de la plate-forme et du bras, ainsi que K_{ppos} et K_{dpos} , les matrices des gains proportionnels et dérivés de la boucle de commande en position. L'erreur en effort s'écrit :

$$\tilde{F} = F_d - F_r$$

La loi de commande de l'effort s'écrit :

$$\Delta P_d = K_p^{eff} \cdot \tilde{F} + K_i^{eff} \cdot \int \tilde{F} \cdot dt$$

La nouvelle consigne de position s'écrit :

$$P_d^* = P_d + \Delta P_d$$

L'erreur de position :

$$\tilde{P} = P_d^* - P$$

La loi de commande de la position s'écrit :

$$U = \left(K_p^{pos} \cdot \tilde{P} + K_d^{pos} \cdot \dot{\tilde{P}} \right) \cdot G_{CNA}$$

G_{CNA} : matrice des gains des Convertisseurs Numériques Analogiques.

Dans le cas de la commande en effort ($\alpha = 1$), nous avons :

$$F_d = \begin{pmatrix} F_x^d & F_y^d & \tau_\theta^d & \tau_{\theta_1}^d & \tau_{\theta_2}^d \end{pmatrix}^T, \\ F_r = \begin{pmatrix} F_x & F_y & \tau_\theta & \tau_{\theta_1} & \tau_{\theta_2} \end{pmatrix}^T$$

Dans le cas de la compensation ($\alpha = 0$):

$$F_d = \begin{pmatrix} 0 & 0 & 0 & 0 & 0 \end{pmatrix}^T, \\ F_r = \begin{pmatrix} F_x & F_y & \tau_\theta & 0 & 0 \end{pmatrix}^T$$

On définit la commande généralisée effort / compensation, avec :

$$F_r = \begin{pmatrix} F_x & F_y & \tau_\theta & \alpha \cdot \tau_{\theta_1} & \alpha \cdot \tau_{\theta_2} \end{pmatrix}^T \\ F_r^{pf} = \begin{pmatrix} F_x & F_y & \tau_\theta \end{pmatrix}^T = J_{pf}^T \cdot F_{C_{aptEff}} \\ \Gamma_r^b = \begin{pmatrix} \tau_{\theta_1} & \tau_{\theta_2} \end{pmatrix}^T = J_b^T \cdot R_{C_{aptEff}} \cdot F_{C_{aptEff}}$$

Conclusion :

Nous avons vu dans ce chapitre différentes méthode de commande hybride et nous avons choisis la méthode du trapèze pour la loi de commande en force et la commande hybride par compensation pour la loi de commande en force cette dernière permet une compensation de l'effort par réajustement de la position.

Chapitre III : Description matérielle et logicielle

III.1) Description matérielle :

On procédera à la description des composants utilisés pour la réalisation de notre projet .On classe les constituantes physique du système par rapport à leurs fonctions respectives :

III.1.1) La commande :

III.1.1.1) Centre de commande :

La commande est assurée pas La nouvelle Desktop NanoBoard carte de développement FPGA d'Altium, elle permet le prototypage rapide de conceptions électroniques.



Fig III.1 Desktop NanoBoard

NanoBoard 3000 est un environnement de conception programmable qui est doté de tout le matériel, le logiciel et les IP dont tout concepteur a besoin pour concevoir les produits électroniques de prochaine génération .Pour plus d'information se reporter à la documentation de la Desktop NanoBoard.

III.1.1.2) Organe de Commande :

La commande est transmise via le Port série RS232. Sa transmission se fait à une vitesse de 115,2 K Bauds, la contrainte de temps réel est largement respectée.

III.1.2) L'acquisition :

III.1.2.1) Centre d'acquisition :

Le traitement des données d'acquisition est assuré pas la Xmega Ready-Board .La XMEGA-Ready est une solution complète pour le développement simple et rapide d'applications embarquées qui utilise le nouvel ATxmega128A1 d'Atmel dispositif connecté à un oscillateur 8MHz.

Principales caractéristiques:

- Le programme Bootloader chargé dans le microcontrôleur ATxmega128A1;
- communication USB-UART.
- Alimentation 7-23V AC ou 9-32V DC.

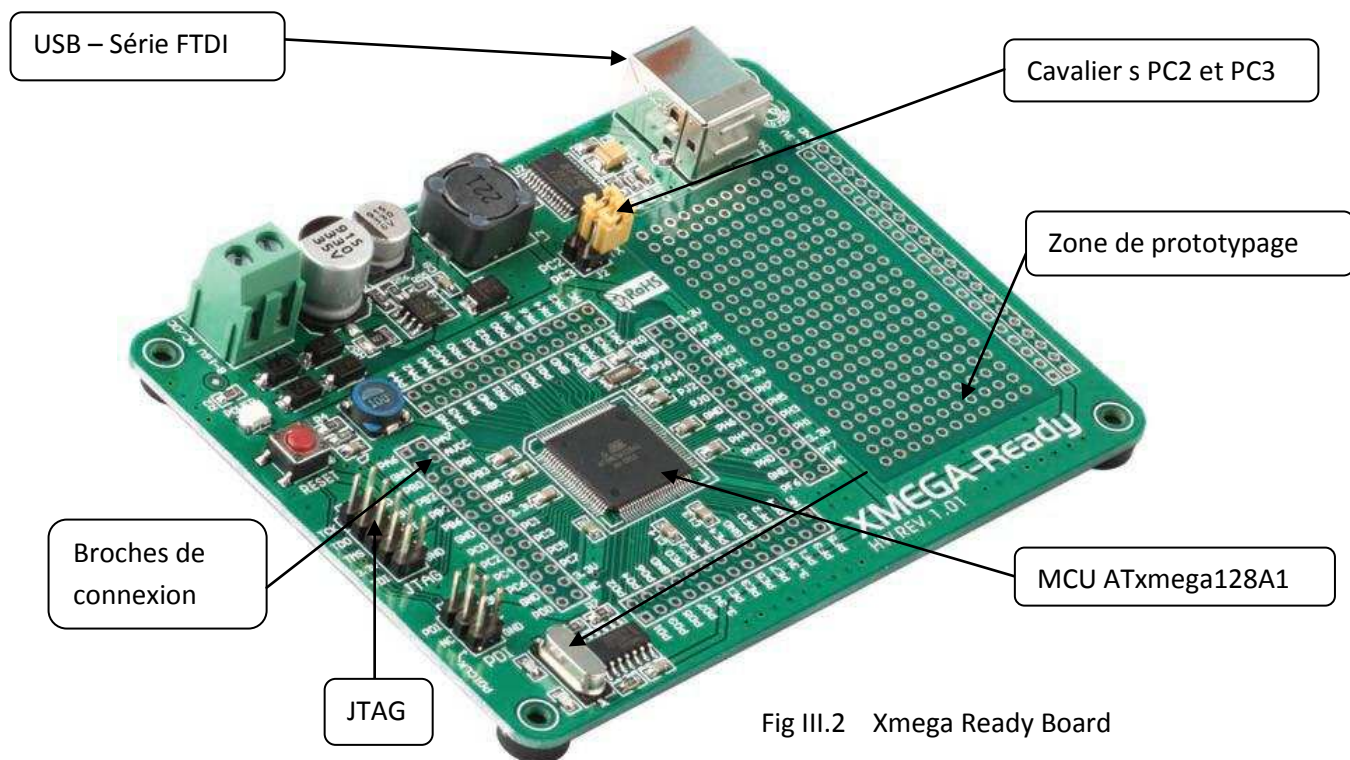


Fig III.2 Xmega Ready Board

Le XMEGA-Ready représente un système de développement en miniature qui peut être utilisé comme un appareil autonome. En raison à son programme bootloader pré chargés et du MCU 8-bits (32 MIPS)

Pour un accès facile aux broches du microcontrôleur fourni sur la XMEGA, nous pouvons utiliser les pads associés. Chaque pad est clairement marquée du nom de la broche à laquelle il est connecté. Nous pouvons également fournir à la XMEGA des composants supplémentaires en les plaçant sur la zone de prototypage.

Pour connecter la XMEGA à un PC, il est nécessaire de connecter le port USB sur le PC à un connecteur USBCN7 via un câble USB. Lorsque la connexion est établie, le PC communique avec la puce FTDI qui est relié aux broches PC2 et PC3 du MCU. Ces broches sont utilisées pour la communication UART série.

Le connecteur CN3 (PDI) est utilisé pour la programmation / débogage via l'interface PDI. Le CN6 connecteur (JTAG) est utilisé pour la programmation / débogage via l'interface JTAG

Schéma électronique de la XMEGA-Ready :

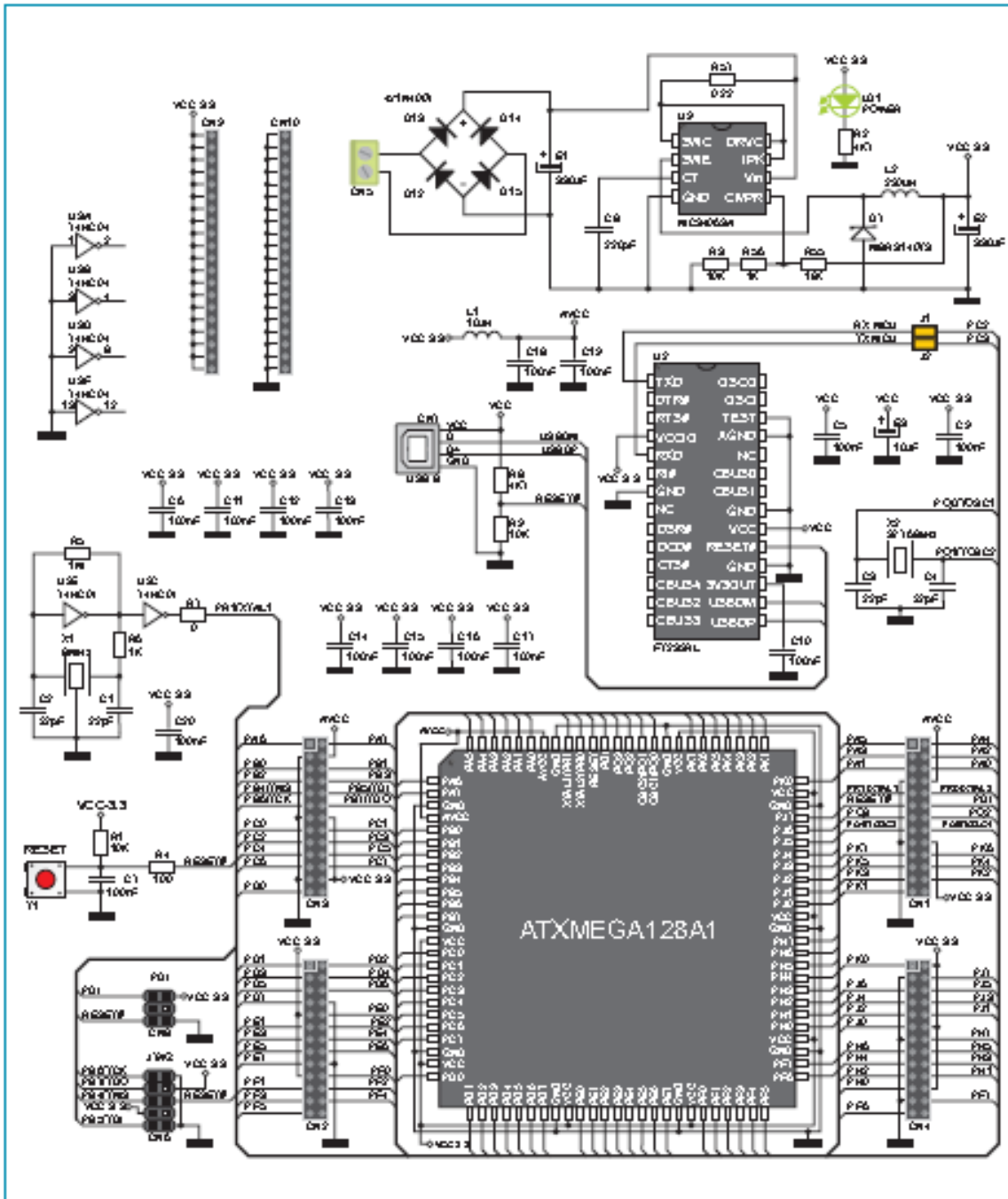


Fig III.3 Schéma électronique de la XMEGA-Ready

III.1.2.2) Organes d'acquisition :

III.1.2.2.1) Acquisition de la force :

L'acquisition de la force se fait grâce à des capteurs de force FSR placés sur différentes parties du bras manipulateur.

Le Capteurs de force (FSR) est un film polymère épais (FTC) dispositif qui présente une diminution de la résistance avec une augmentation de la force appliquée à tension active. Sa sensibilité est optimisée pour une utilisation en contact humain cela pour le contrôle d'appareils électroniques. Les FSR ne sont pas des cellules de charge ou des jauges de contrainte, même si elles ont des propriétés similaires.

Ces capteurs ont une plage de mesure de 10g à 10Kg selon le branchement et l'actionneur. La résistance varie entre 2MΩ et 1kΩ (10Kg). Leur durée de vie est d'environ 1, 000,000 d'actionnement.

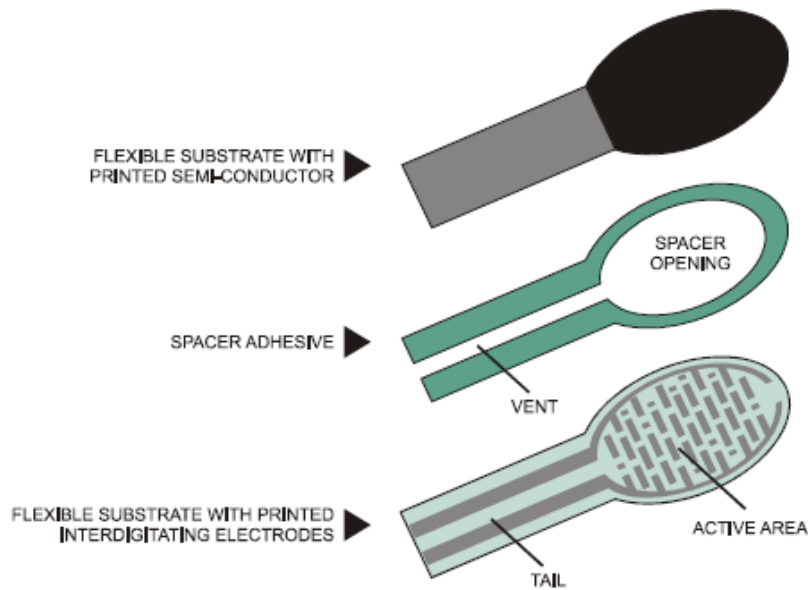


Fig III.4 Capteur FSR

Sur ces capteurs, la surface active est constituée par la zone striée (circulaire ou rectangulaire et de taille variable). La taille de cette zone et sa forme n'influence pas la variation de la résistance. Les différentes formes permettent d'adapter la surface active au montage mécanique.

Conditionnement du capteur :

Diviseur de tension FSR :

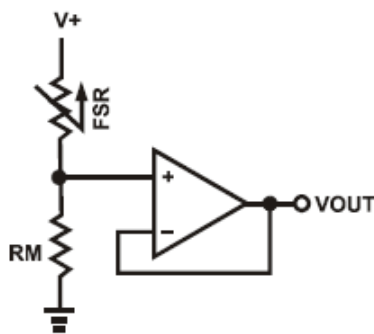


Fig III.5 Diviseur de tension FSR

Pour une simple conversion force tension, le dispositif FSR est reliée à une résistance de mesure dans une configuration diviseur de tension. La sortie est décrite par l'équation suivante:

$$V_{OUT} = (V+) / [1 + E_{SVAD} / R_M].$$

Dans la configuration représentée, la tension de sortie augmente avec l'augmentation de la force. Si on intervertit E_{SVAD} et R_M la sortie va diminuer avec une force croissante. Ces deux formes de sortie sont des images miroir sur la ligne $V_{OUT} = (V+) / 2$.

La résistance de mesure, R_M , est choisi pour maximiser la portée voulue de sensibilité de la force et pour limiter le courant.

Le courant doit être limité à travers le FSR à moins de 1 mA / cm carré de la force appliquée. Les amplificateurs suggérés pour une conception d'alimentation recto verso sont les LM358 LM324 et les Périphériques d'entrée FET LF355 et TL082 sont également bon. Les faibles courants de polarisation de ces ampli-op réduisent l'erreur due à l'impédance de source du diviseur de tension.

Etalonnage :

Le groupe de force en fonction des courbes V_{OUT} est illustré sur le graphique ci-dessous pour un FSR standard dans un diviseur de tension configuré avec différentes résistances R_M . A (V +) de +5 V a été utilisé pour ces exemples.

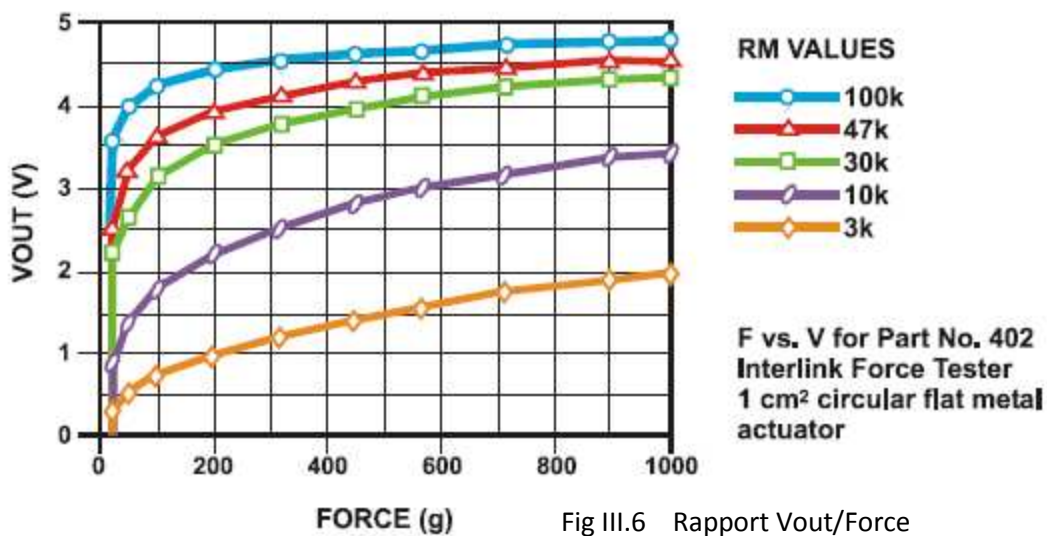


Fig III.6 Rapport Vout/Force

III.1.2.2.2) Acquisition de la Position :

Pour obtenir le retour de position nous avons dû modifier nos servomoteurs, cela en reliant un fil au potentiomètre interne de chaque servomoteur ce qui nous a permis de connaître la position de chaque articulation.



Le fil blanc est relié au potentiomètre du servomoteur.

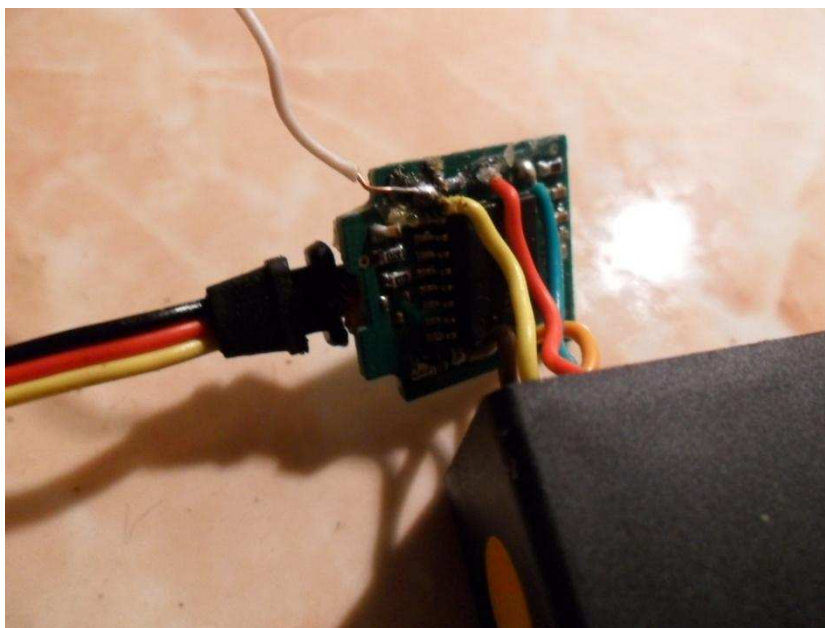


Fig III.7 Etape de modification des servomoteurs

Nous obtenons ainsi une tension comprise entre 0,5V et 1,76V, en étalonnant cette valeur nous avons pu déterminer la position exacte de chaque articulation du robot manipulateur.

III.1.3) Mouvement du Bras manipulateur :

III.1.3.1) Organe du mouvement :

Le Bras manipulateur est actionnée par 6 servomoteurs.

Qu'est ce qu'est un servomoteur ?

En modélisme, un servomoteur est souvent appelé servo. C'est un ensemble mécanique et électronique comprenant :

- un moteur à courant continu de très petite taille ;
- un réducteur en sortie de ce moteur diminuant la vitesse mais augmentant le couple ;
- un potentiomètre (faisant fonction de diviseur résistif) qui génère une tension variable proportionnelle à l'angle de l'axe de sortie.
- un dispositif électronique d'asservissement ;
- un axe dépassant hors du boîtier avec différents bras ou roues de fixation.

Les servos servent à actionner les parties mobiles du modèle : ailerons, volets et trains pour les avions, contrôle de direction dans le cas de modélisme automobile... À cette fin, les moteurs sont asservis et obéissant à une commande externe, généralement transmise par radio.

Fonctionnement :

Les servomoteurs sont commandés par l'intermédiaire d'un câble électrique à trois fils qui permettent d'alimenter le moteur et de lui transmettre des ordres de positions sous forme d'un signal codé en largeur d'impulsion. Cela signifie que c'est la durée des impulsions qui détermine l'angle absolu de l'axe de sortie et donc la position du bras de commande du servomoteur. Le signal est répété périodiquement, en général toutes les 20 ms, ce qui permet à l'électronique de contrôler et de corriger continuellement la position angulaire de l'axe de sortie, cette dernière étant mesurée par le potentiomètre.

Lorsque le moteur tourne, l'axe du servo change de position, ce qui modifie la résistance du potentiomètre. Le rôle de l'électronique est de commander le moteur pour que la position de l'axe de sortie soit conforme à la consigne reçue : c'est un asservissement.

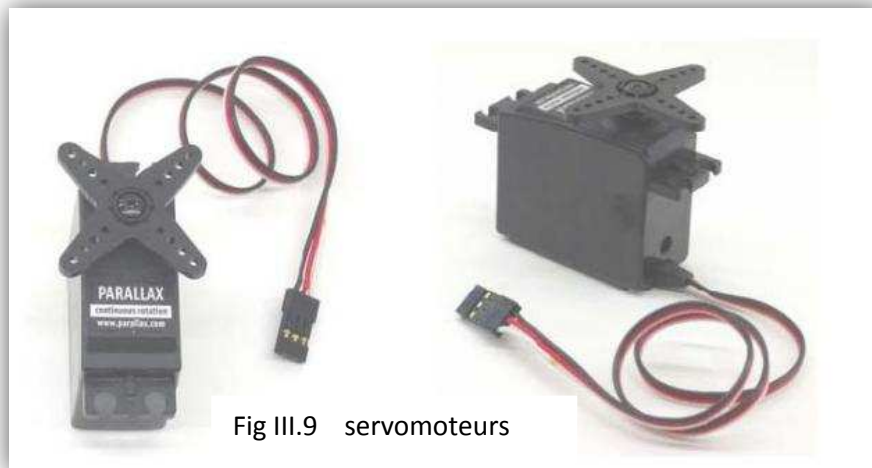
Présentation :

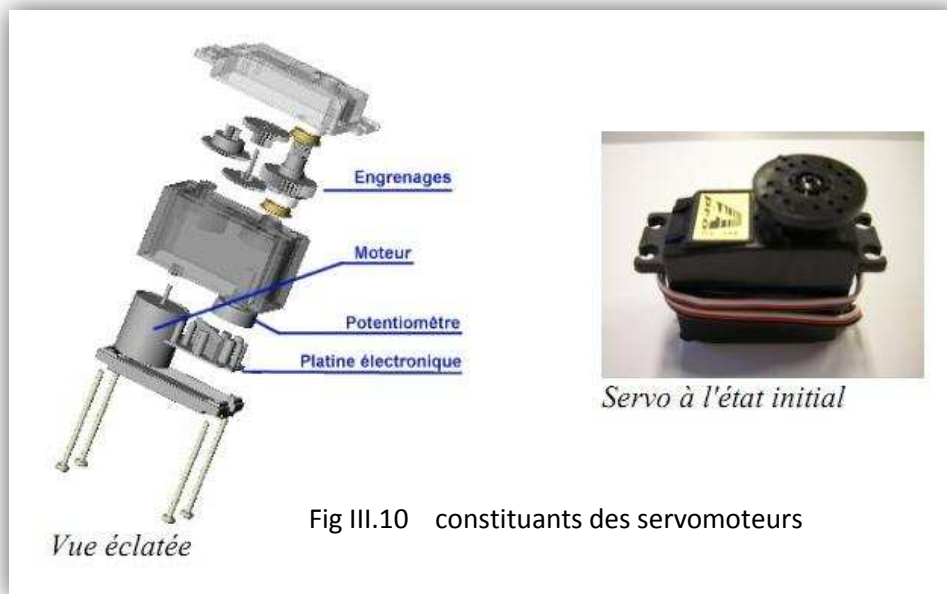
Fig III.9 servomoteurs

Les servomoteurs sont des blocs moteurs comprenant plusieurs organes :

- un moteur courant continu,
- un système de démultiplication à pignons,
- un capteur potentiométrique (résistance variable semi circulaire avec curseur solidaire de l'axe moteur) fournissant une tension proportionnelle à la position de l'axe,
 - des butées mécaniques (uniquement pour les servos asservis en position exemple option « gripper » Parallax)
- un circuit électronique de régulation.

Un servomoteur est commandé par un train d'impulsions périodiques (Fig.3-2): c'est à dire qu'il faut envoyer plusieurs impulsions successives pour qu'il puisse se mettre en mouvement. La largeur

L de l'impulsion détermine la position angulaire à atteindre pour les servos asservis en position, ou



la vitesse de rotation pour les servos à rotation continue (sur 360°)

Lorsque le signal de commande n'est plus appliqué, le servo s'arrête dans la dernière position atteinte et plus aucun couple moteur n'est généré.

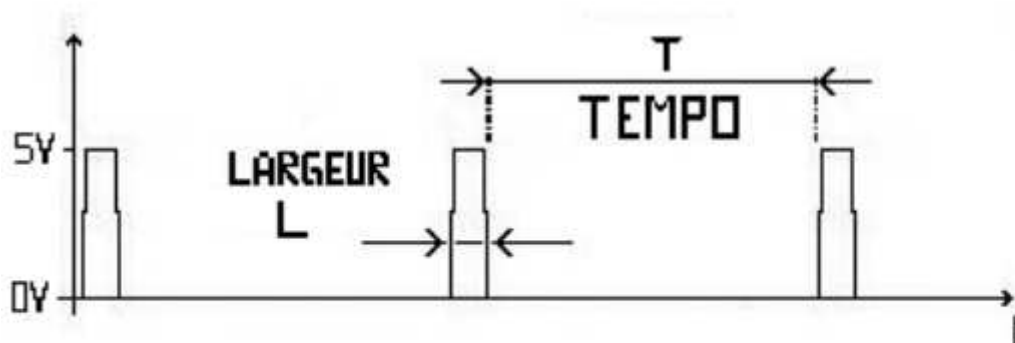
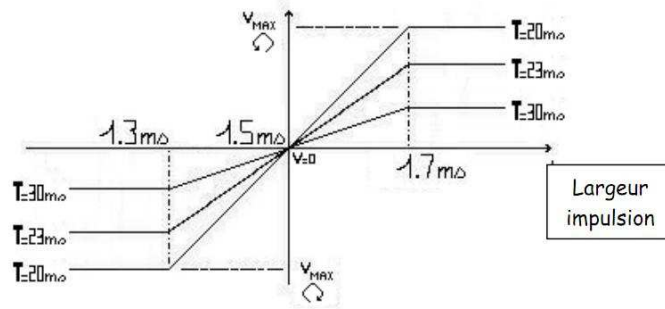


Fig III.11 Train d'impulsion envoyé au servomoteur



Pour les servomoteurs à rotation continue (locomotion du robot), le sens de rotation et la vitesse est défini par la largeur de l'impulsion(L) :

- $L < 1,5\text{ms}$, il tourne dans le sens horaire,
- $L = 1,5\text{ms}$, il se met en état stationnaire, (point neutre)
- $L > 1,5\text{ms}$, il tourne dans le sens antihoraire.

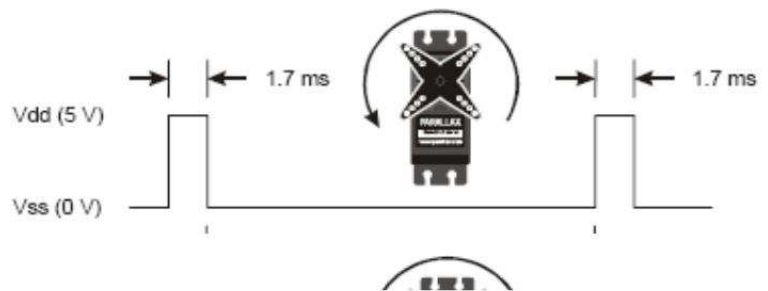


Fig III.12 sens de rotation des servomoteurs ms

Remarque : La largeur L d'impulsion doit être comprise entre $1,3\text{ms}$ et $1,7\text{ms}$ pour une commande correcte. Dans cet intervalle, la vitesse est « quasiment proportionnelle » à la largeur de l'impulsion. Au-delà de cette plage, la vitesse de rotation atteint des valeurs de saturation et peut même décroître.

L'architecture interne :

Un servo moteur contient un moteur à courant continu, un réducteur à roues dentées à axes parallèles et une électronique de commande. L'alimentation et la commande se font par un câble de trois fils, un commun, un fil d'alimentation et un autre de commande. Les couleurs sont conventionnelles pour le constructeur.

Inversion du sens de rotation du moteur :

La tension d'alimentation étant unipolaire, il faut utiliser une disposition classique modélisée ci-

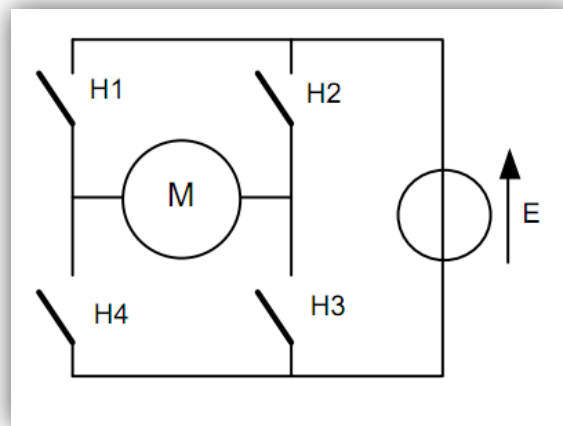
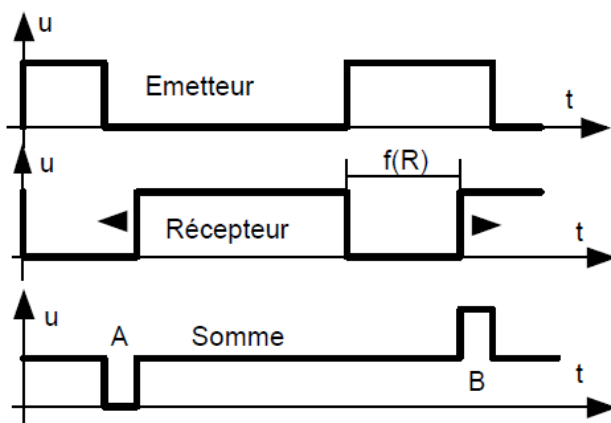


Fig III.13 schéma électrique de commande

dessous.

Le décodage de l'information position :



Le récepteur contient un générateur de signal comparable à celui de l'émetteur qui fournit une impulsion de largeur variable comme le montre le chronogramme ci-contre.

La largeur de l'impulsion est fonction de la position du curseur d'un potentiomètre. Ce dernier est solidaire de l'arbre de sortie. La position angulaire du servomoteur est donc connue et traduite, elle aussi, en une durée. L'électronique réalise la somme des signaux provenant de l'émetteur et du récepteur.

Nous voyons apparaître deux zones, A et B qui fournissent les renseignements souhaités, le sens de rotation et l'angle représentatif de l'erreur de position entre la consigne et la position réelle. Il suffit de commander les interrupteurs de la figure du § 3.1 pour faire tourner le moteur afin de réduire l'erreur. En effet, lorsque l'arbre moteur tourne, il entraîne avec lui l'axe du potentiomètre qui fixe la largeur de l'impulsion du récepteur. La rotation cesse quand l'erreur A ou B disparaît. Les flèches indiquent le sens de déplacement.

Le générateur contenu dans le récepteur doit être synchronisé au signal provenant de l'émetteur. Le front montant de ce dernier déclenche le fonctionnement du générateur local au récepteur.

III.1.3 .1) Centre du mouvement :

Les 6 servomoteurs sont actionnés par un contrôleur de servomoteurs en l'occurrence le carte SSC-32.

Le Contrôleur de servomoteurs SSC-32 Lynxmotion a une excellente précision (1us) pour des mouvements en douceur. La plage de contrôle se situe entre 0.50uS et 2.50uS pour 180 degrés.

Quelques caractéristiques de la SSC-32 :

- Contrôle jusqu'à 32 servomoteurs
- Fonction unique de mouvement de groupe permet de bouger plusieurs servomoteurs afin qu'ils débutent et terminent leur mouvements en même temps
- Interface série (TTL ou RS-232)

Une fonction unique de mouvement de groupe permet de bouger plusieurs moteurs afin qu'ils débutent et terminent leurs mouvements en même temps, même si les déplacements entre les servomoteurs sont différents. Cette fonction est très performante pour créer des robots marcheurs complexes.

Il est possible de recevoir des lectures de données venant des moteurs. Le SSC-32 a aussi 12 séquences intégrées pour robot Hexapode. N'importe quelles sorties peut être utilisées comme une sortie logique TTL et il y a aussi 4 entrées digitales ou analogiques disponibles.

III.2) Description Logicielle :

III.1.1) Conception de la commande :

La systémique embarquée sur la Desktop Nanoboard est réalisée grâce au logiciel Altium Designer.

Altium Designer est une application tout-en-un unique, qui intègre toutes les technologies et les fonctions nécessaires au développement de produits électroniques. Altium Designer intègre la conception de systèmes pour PCB (création des circuits imprimés en CAO électronique) avec FPGA embarqués, le développement de logiciels enfouis pour processeurs de type FPGA et processeurs discrets, ainsi que le routage, l'édition, la simulation des signaux mixtes et la fabrication de PCB, le tout au sein d'un environnement de conception collaboratif. Cela, allié à des fonctions modernes de gestion des données de conception intégrant un véritable PLM basé sur l'Enterprise Vault Server (Gestion avancée du cycle de vie des produits) et la notion de Cloud Computing à plusieurs niveaux, fait d'Altium Designer la solution complète de développement de produits électroniques .

Altium propose également en option un nouveau module de prototypage matériel reconfigurable, la NanoBoard, offrant des possibilités uniques qui en font une solution révolutionnaire. Avec Altium Designer et la NanoBoard, plate-forme matérielle reconfigurable, les frontières entre Logiciel et Matériel disparaissent pour laisser place à l'Innovation Station.

On pourra ainsi créer du matériel "Prêt-à-Industrialiser" à architecture variable, bien plus avantageux à maintenir/corriger/faire évoluer en fonction des besoins.

Prise en charge des conceptions grande vitesse :

L'accélération des vitesses d'horloge et le développement des interconnexions rapides en série impose au développement de cartes généralistes de se pencher sur la question de la conception haute vitesse. La propagation des signaux haute vitesse impose certaines contraintes particulières concernant la conception physique. Altium Designer prend en charge la conception haute vitesse avec des règles de conception ciblées, la prise en charge complète, à l'échelle du système, de la gestion des signaux différentiels et l'analyse intégrée de l'intégrité du signal.

Le système étendu de règles d'Altium Designer nous permet de créer des contraintes visant tout particulièrement la conception haute vitesse et la signalisation différentielle. Par exemple, on peut

définir la longueur maximale des équipotentiels, contrôler le nombre de traversées d'une équipotentielle et faire correspondre les longueurs des équipotentiels définies, en ajoutant automatiquement des segments en accordéon aux traces en fonction des besoins. Nous pouvons également spécifier la longueur maximale de traces parallèles pour éviter le crosstalk entre équipotentiels.

III.2.2) Traitement de l'acquisition :

On programme la Xmega Ready-Board avec le Logiciel AVRStudio5 qui est l'environnement de développement intégré (IDE) pour développer et déboguer les applications intégrées Atmel AVR. L'AVR Studio 5 IDE nous donne une facilité à utiliser l'environnement pour écrire, générer et déboguer nos codes C / C + + et assembleur.

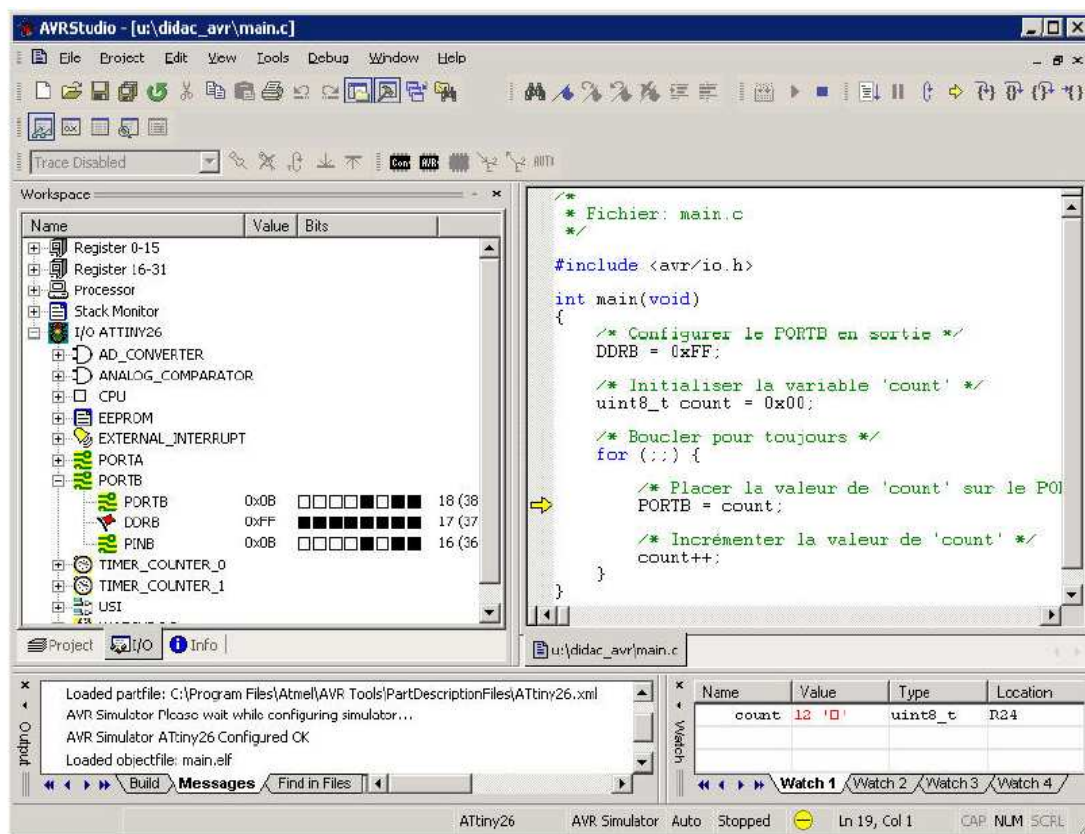


Fig III.14 AVR Studio 4

Chapitre IV : Réalisation

IV.1) Introduction :

Notre Projet se veut être totalement pratique, le choix des composants s'est fait au cours de la conception et de façon très spontanée.

Dans ce dernier chapitre nous allons aborder le coté réalisation de notre projet, pour cela nous aurons besoin de tous les dispositifs décrits dans les chapitres précédents

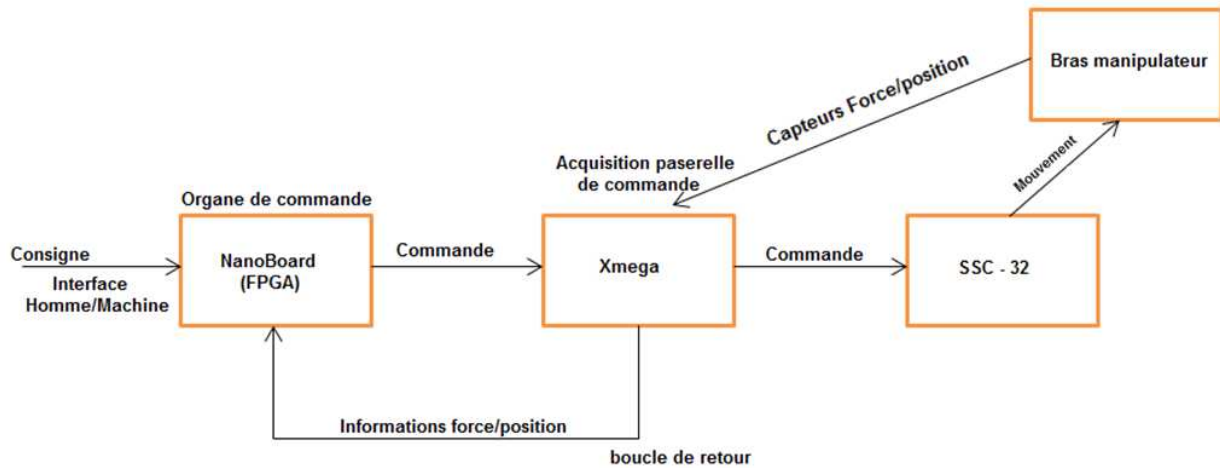


Fig IV.1 boucle de commande de notre projet

On vise à faire communiquer 3 dispositifs (cartes) en temps réel via un port RS232

La SSC-32 commandera le mouvement du bras en actionnant les Servomoteurs.

La Xmega-Ready assurera l'acquisition des données liée au robot c'est-à-dire d'une part sa position absolu et d'autre part les forces appliquées sur notre bras manipulateur, elle fera aussi office d'organe de liaison entre la Desktop Nanoboard dont elle recevra les instructions de mouvements du bras manipulateur et la SSC-32 qui assurera le mouvement du robot.

La Desktop Nanoboard sera le centre de traitement de l'information que lui renvoie la Xmega-Ready-board, son rôle est primordial car c'est elle qui assure la commande hybride en Force \ Position.

IV.2) Description du projet FPGA :

Pour assurer la commande du robot manipulateur nous avons configuré la Desktop NanoBoard en interface Homme-machine. L'interaction se fait via différents périphériques que nous allons décrire :

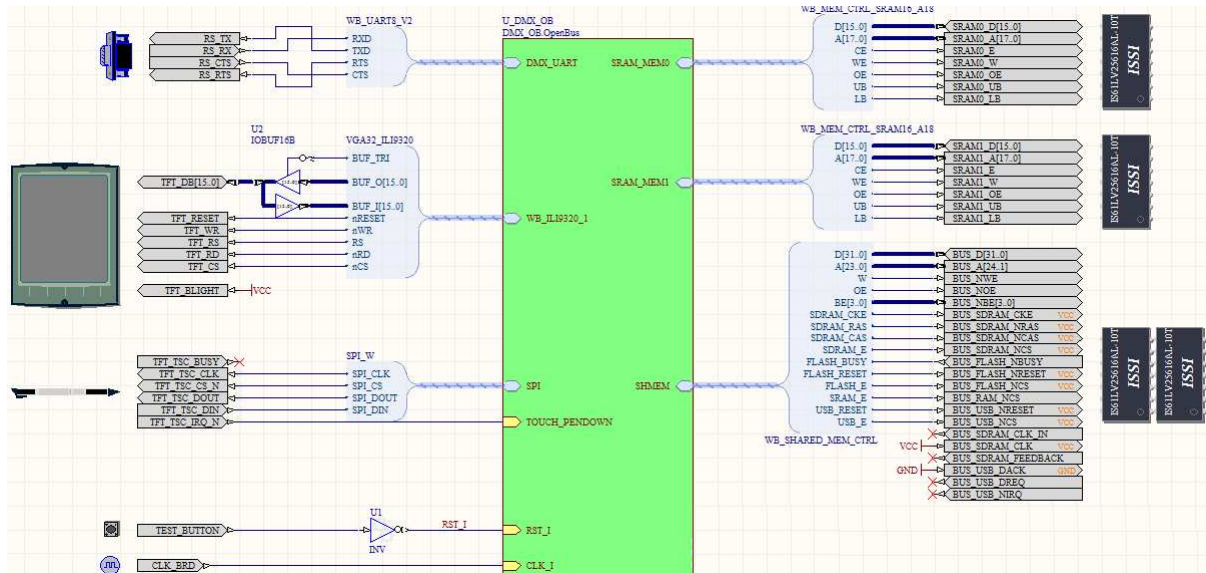


Fig IV.2 description du projet FPGA

La figure ci-dessus représente le schéma fonctionnel de notre dispositif. La partie systémique est contenu dans le block open_bus (technologie enfoui carré vert) n'apparaissent que les périphériques actifs de la Desktop Nanoboard qui sont connecté au système.

IV.3) Description des périphériques externe :

IV.3.1) Interface Homme-Machine :

- Les consignes de position et de force sont communiquées à la Desktop Nanoboard via un écran TFT tactile.

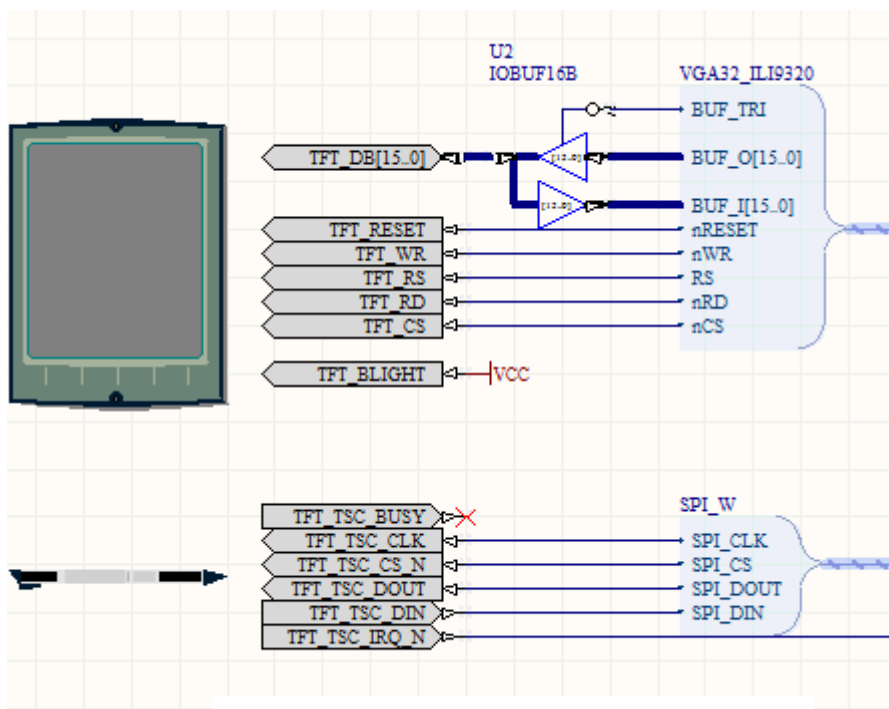


Fig IV.3 connexion de bus via un écran TFT

Sur cette partie du Schéma nous voyons les différents bus qu'utilise l'écran TFT pour assurer la communication avec notre système. Nous avons d'une part l'affichage de l'interface ergonomique qui

permet à l'utilisateur de piloter le bras manipulateur et d'autre part le coté tactile qui permet l'envoi d'informations vers la Desktop NanoBoard, cette envoi est assuré par un module SPI liée à l'écran TFT.

IV.3.2) Entrée Sortie des Informations traitées :

La Desktop Nanoboard communique avec les autre carte via le port RS232.

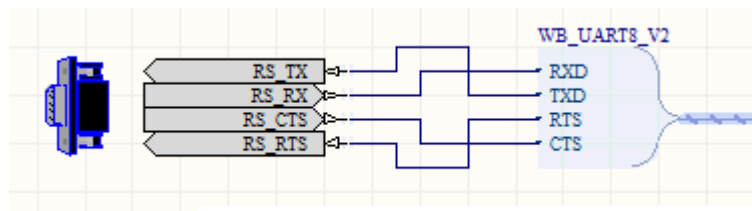


Fig IV.4 E/S des informations traitées

Sur cette partie du Schéma apparaissent les différents bus de communication du module RS232 de la Desktop Nanoboard. Il faut savoir que la communication via RS232 se fait en code ASCII à une vitesse de 115,2 k baud. La contrainte du temps réel est largement respectée.

IV.3.3) Les mémoires externes du dispositif :

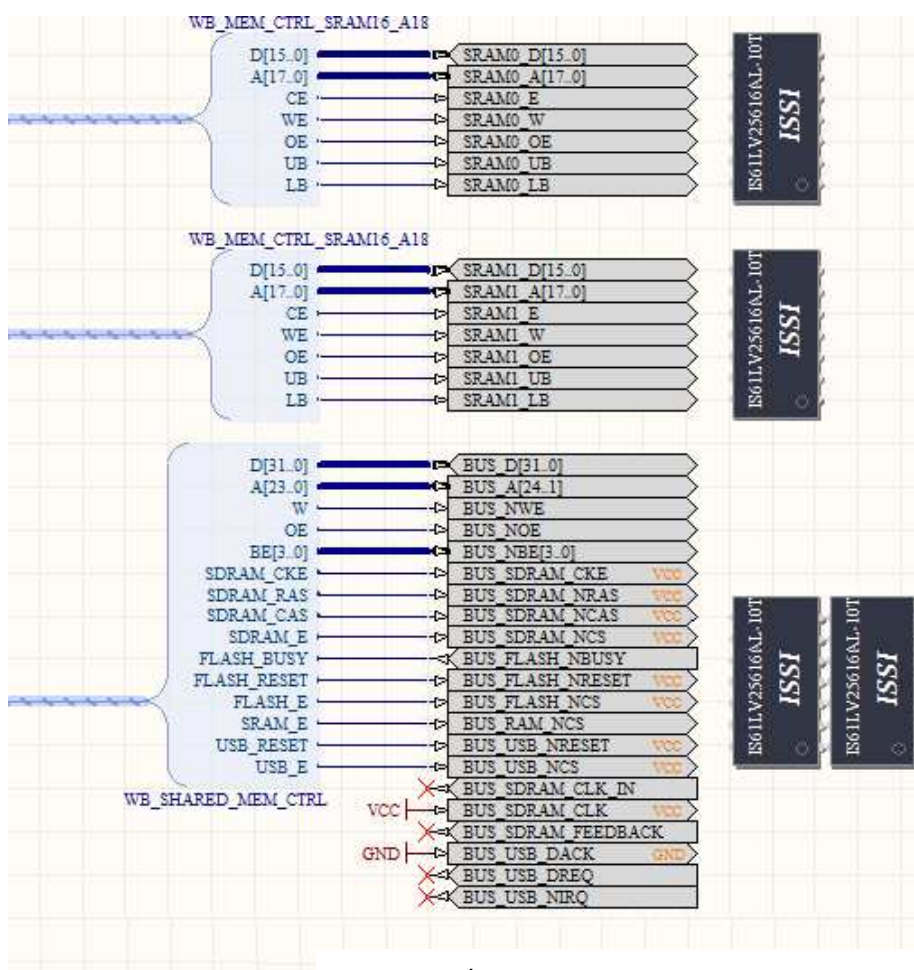


Fig IV.5 mémoires

Sur cette partie du schéma, on visualise les mémoires nécessaires au fonctionnement de notre système. Il y a deux blocks mémoire, un block mémoire asynchrone qui assure l’affichage des données graphique sur l’écran TFT tactile et un block SRAM qui contient les programmes C embarqué de notre système.

- Bouton de réinitialisation système et signal d’horloge:



Fig IV.6 Bouton de réinitialisation et signal d’horloge

IV.4) Description interne du système :

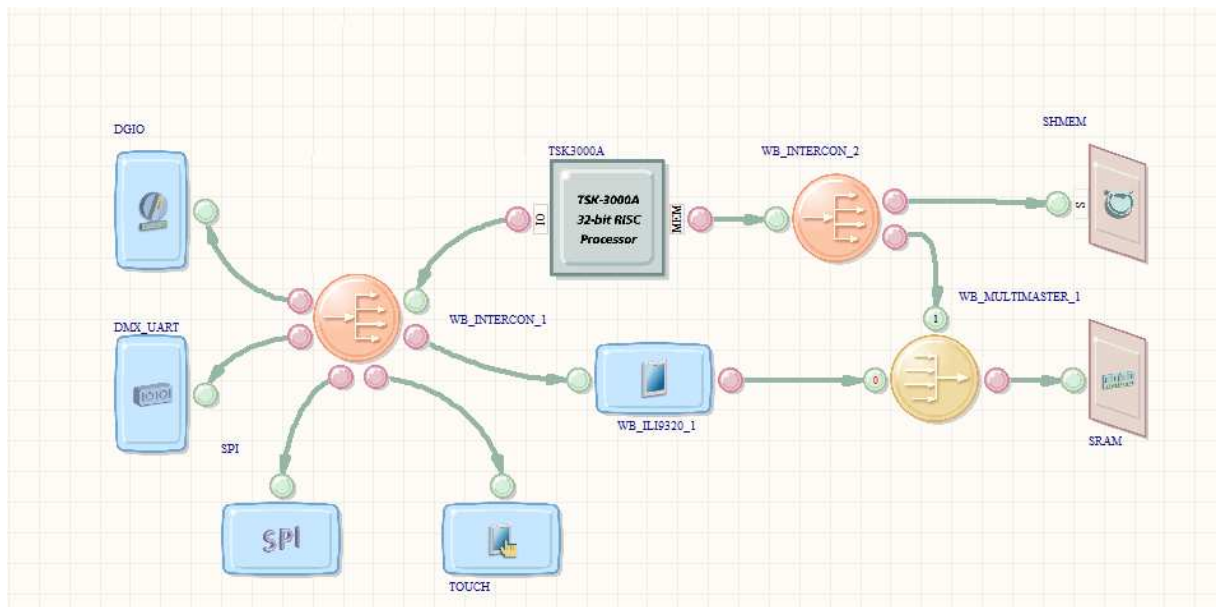


Fig IV.7 Le schéma Open_bus

Le Schéma Open_bus suivant décrit la structure interne de notre système. Procédons à la description des différents composants qui constituent notre dispositif :

IV.4.1) Les Périphériques internes :

On définit comme périphériques internes les composants du système interne qui sont reliée physiquement aux Broches FPGA et qui garantissent la communication avec l’extérieur.

IV.4.1.1) Périphérique écran TFT :

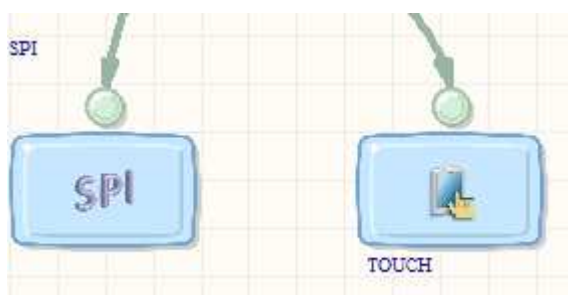


Fig IV.8 écran TFT

IV.4.1.2) Périphérique RS232 :



Fig IV.9 RS232

IV.4.1.3) Périphériques Mémoires :

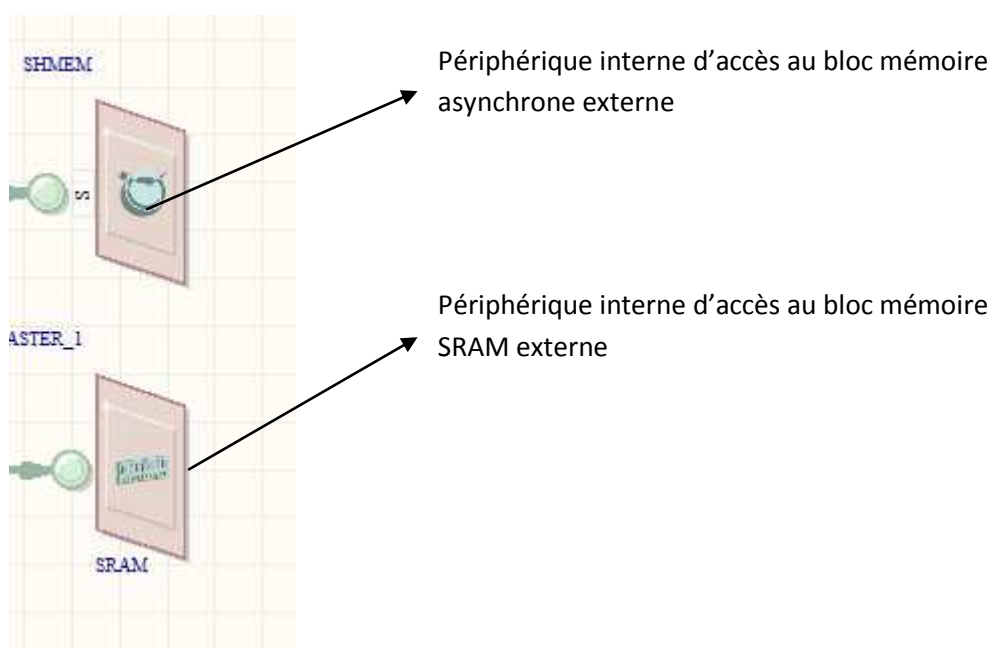


Fig IV.10 SRAM

IV.4.1.4) Les composantes internes :

Module de pilotage de l'écran TFT :

Le module VGA 32 bit ILI9320 permet la communication de l'écran TFT avec le processeur Softcore interne.

Le Cœur du système :

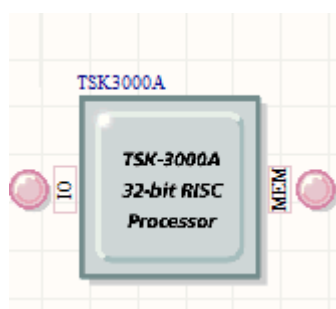


Fig IV.11 FPGA

Le traitement des informations et de la commande est faite par un processeur Softcore ici nous avons choisi le très performant TSK-3000 32-bits. Les différentes entrées \ sorties du processeur sont liées aux différents périphériques internes via des modules de liaison spécifique spécialement conçu pour acheminer les données. Ici nous avons utilisé un intercon qui permet au processeur de communiquer avec plusieurs périphérique et un Multi master qui fait communiquer un périphérique avec plusieurs composantes internes ici le block mémoire asynchrone d'une part et le TSK 3000 + le Module VGA 32 bit d'autre part.

Cette partie du projet FPGA représente le coté « HDL »de notre système.

IV.4.2) Le projet embarqué :

On définit par projet embarqué la partie code C de notre projet qui est chargé dans la mémoire externe SRAM et traitée par le processeur Softcore TSK-3000. Pour concevoir un projet embarqué il faut dans un premier temps inclure tous les drivers des périphériques externes de la Desktop NanoBoard. Pour cela Altium designer a mise à notre disposition une bibliothèque de driver C qu'on pourra inclure dans un fichier externe spécifique appelé Software Platform.

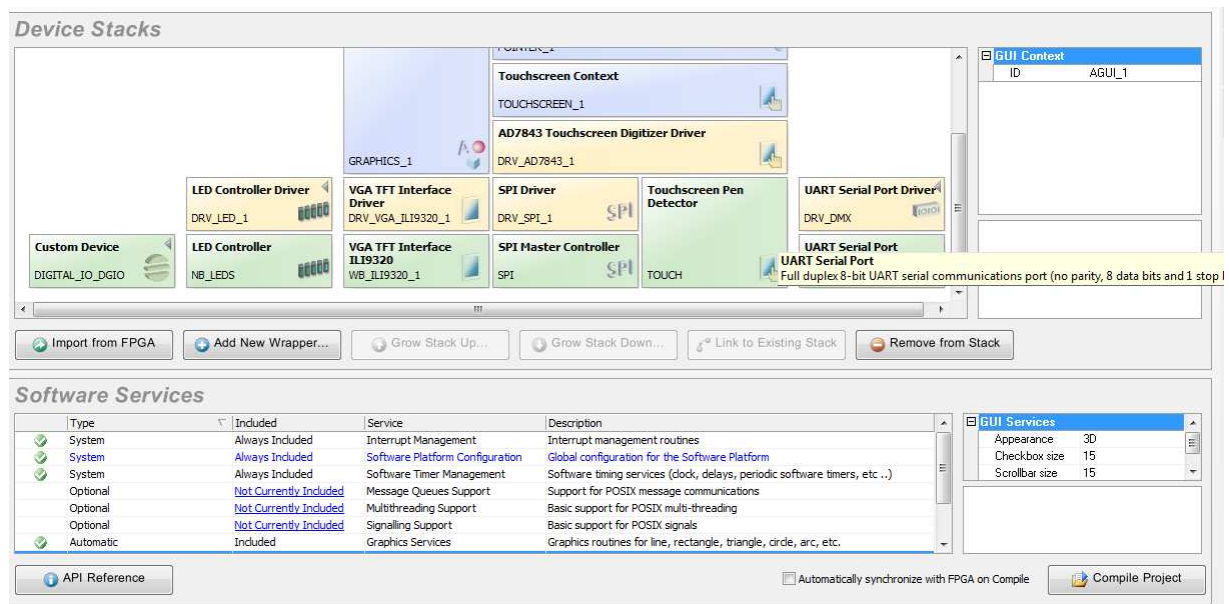
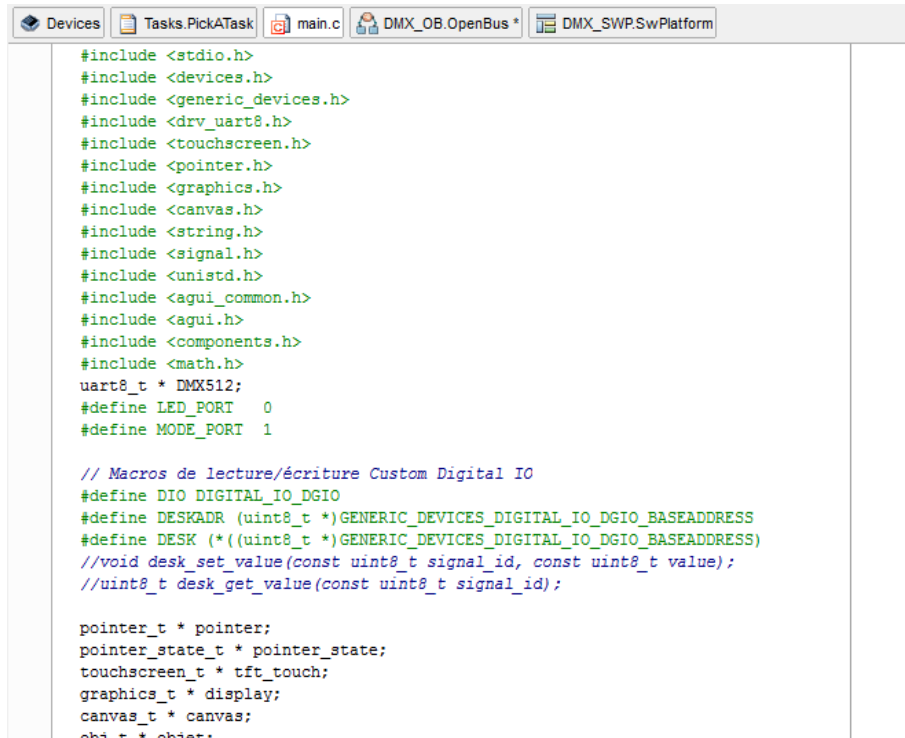


Fig IV.12 Le projet embarqué

La configuration d'un fichier Software Platform se fait automatiquement en important les composants présents sur la FPGA directement vers notre fichier. On pourra modifier différentes options liées à nos drivers dans le but d'obtenir les fonctionnalités voulues.

Nous pouvons maintenant créer notre fichier .C et commencer à implémenter notre Algorithme de commande et de traitement de l'information.



```
#include <stdio.h>
#include <devices.h>
#include <generic_devices.h>
#include <drv_uart8.h>
#include <touchscreen.h>
#include <pointer.h>
#include <graphics.h>
#include <canvas.h>
#include <string.h>
#include <signal.h>
#include <unistd.h>
#include <agui_common.h>
#include <agui.h>
#include <components.h>
#include <math.h>
uart8_t * DMX512;
#define LED_PORT 0
#define MODE_PORT 1

// Macros de lecture/écriture Custom Digital IO
#define DIO DIGITAL_IO_DGIO
#define DESKADR (uint8_t *)GENERIC_DEVICES_DIGITAL_IO_DGIO_BASEADDRESS
#define DESK (*(uint8_t *)GENERIC_DEVICES_DIGITAL_IO_DGIO_BASEADDRESS)
//void desk_set_value(const uint8_t signal_id, const uint8_t value);
//uint8_t desk_get_value(const uint8_t signal_id);

pointer_t * pointer;
pointer_state_t * pointer_state;
touchscreen_t * tft_touch;
graphics_t * display;
canvas_t * canvas;
ohi_t * ohier;
```

Fig IV.13 Céation de notre projet .c

Un fois notre Algorithme terminé nous pouvons compiler le projet emballé, on remarque que les fichiers .h liés aux drivers périphériques sont automatiquement générés via la Software platform.

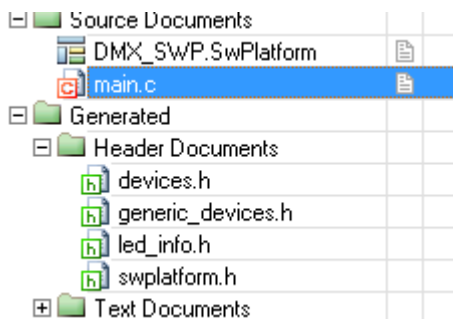


Fig IV.14 les fichiers headers

Après implémentation de notre algorithme de traitement qui sera chargé dans la SRAM

IV.5) Un Dispositif d'acquisition et relai de commande :

La Xmega –Readyboard qui grâce à ses 16 convertisseur analogique numérique, ses 8 port RS232 et sa fonctionnalité multi tâche permet une acquisition et une transmission rapide des différentes informations (force, position et commande) et cela en temps réel.

Elle se programme via un bootloader qui nous permet de charger le fichier .hex généré sous AVRstudio5 directement en mémoire de programme de l'ATxmega128A1.

On relit les 4 capteurs de force FSR qui sont placés sur le bras manipulateur et les sorties analogiques des 6 servomoteurs modifiées à notre carte d'acquisition.

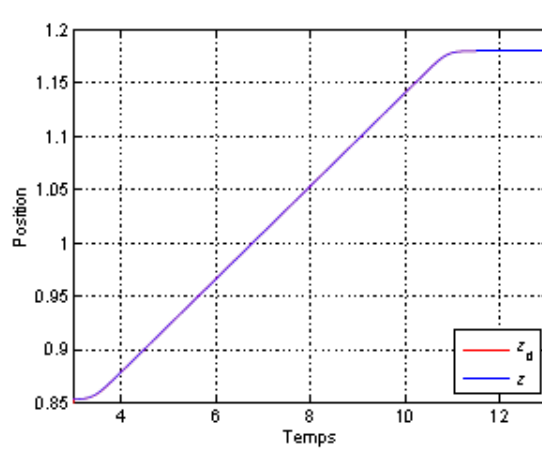
Le mouvement est assuré pas la carte SSC-32 qui reçoit la commande via le port RS232.

IV.5) Résultats de la Simulation :

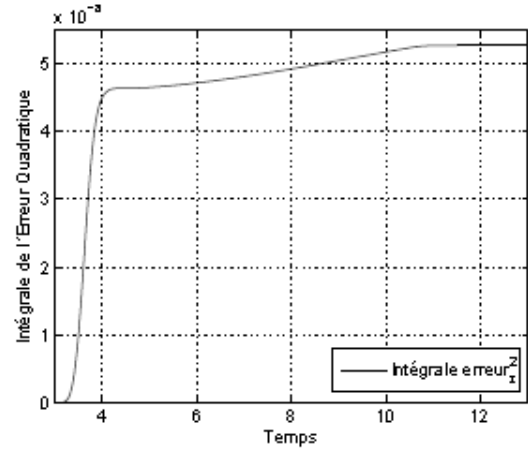
Nous souhaitons déplacer l'organe terminal du bras manipulateur d'une position d'origine PO

Vers une position finale PF

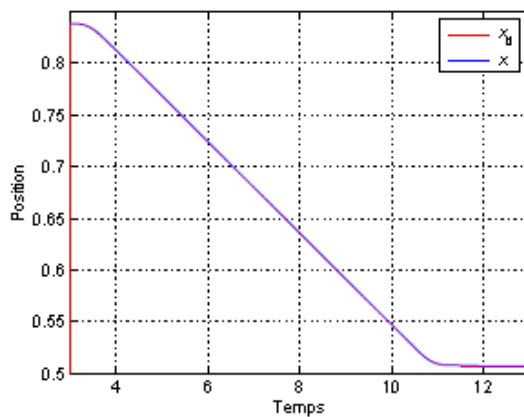
$$P_O = \begin{bmatrix} 0,840 \\ 0,000 \\ 0,860 \end{bmatrix} \quad P_F = \begin{bmatrix} 0,515 \\ -0,045 \\ 1,183 \end{bmatrix}$$



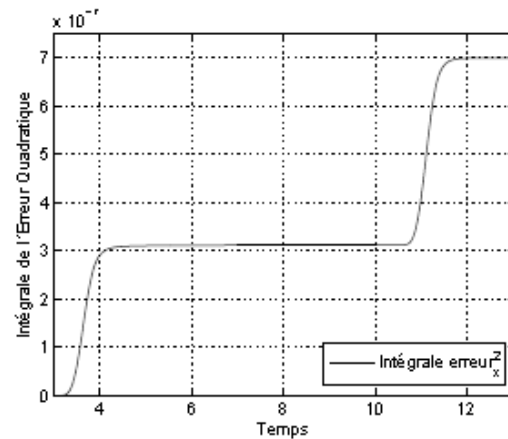
(e) z



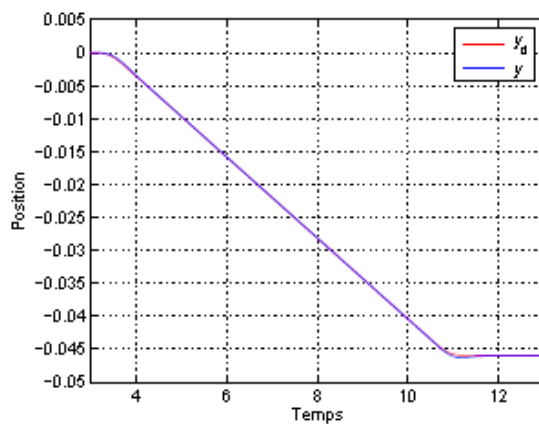
(f) Intégrale de l'erreur quadratique



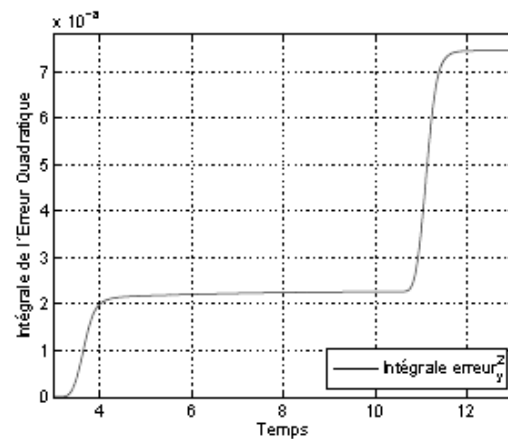
(a) x



(b) Intégrale de l'erreur quadratique



(c) y



(d) Intégrale de l'erreur quadratique

Fig IV.15 Évolution de la position et de l'intégrale de l'erreur quadratique en x, y et z.

Conclusion et perspectives

L'objectif de ce projet était d'une part de se familiariser avec différentes technologies de conception matériel et d'autre part la réalisation d'un bras manipulateur obéissant à une commande hybride force / position avec une vision de conception novatrice et assez personnelle.

En utilisant comme dispositif de commande une carte FPGA, nous avons voulu démontrer tout le potentiel que recèlent cette nouvelle technologie avancée et tous les avantages liés à son utilisation. La découverte de l'outil de développement Altium Designer a radicalement changé notre vision des dispositifs FPGA, nous avons démontré qu'il était facile et rapide de concevoir des systèmes embarqués et cela pour des procédés aussi complexes qu'un bras manipulateur.

Notre travail a aussi porté sur les moyens de faire de l'acquisition de données externes. Cela en utilisant le nouveau microcontrôleur ATxmega128A1 d'ATMEL sorti en 2010 et utilisée notamment par NASA pour réaliser le projet Mars Rover. Nous avons vu que cette nouvelle technologie Multi tâche et Multi accès mémoire était très performante et robuste pour ce qui est de faire du temps réel.

Les perspectives sont nombreuses surtout pour ce qui concerne la téléopération et la commande à distance de nos dispositifs. La nouvelle version d'Altium Designer l'AD10 propose des fonctionnalités Wifi étendues qui nous permettent dès à présent de mettre en réseau nos dispositifs de commande cela rend possible non seulement un contrôle à distance mais aussi un contrôle via internet en équipant nos dispositifs électroniques (Desktop Nanoboard) de mini serveur embarqués.

Bibliographie :

- 1- E.Dombre
«Analyse et modélisation des manipulateurs », Edition Hermès ,2001
- 2- P.Lopez, J.Numafoule
« Introduction à la robotique : notions de base » , Edition Test 1992
- 3- W.Khalil, Etienne Dombre
« Modélisation, identification et commande des robots » Edition Hermes, 1999
- 4- Alexandre Nketsa
« Circuits logiques programmables Mémoires PLD, CPLD et FPGA » Edition Ellipses 1998
- 5- L.Dutrieux, D.Demigny
« Logique programmable – Architecture des FPGAs et CPLDs –Méthode de conception – le langage VHDL » Edition Eyrolles 1997
- 6- R.Airiau , J.M.Berger
« VHDL langage, modélisation, synthèse » Edition CNET 2003
- 7- Mark Zwolinski
« Digital system design with VHDL » Edition Pearson 2004
- 8- Altium Designer Tutorials
www.ALtium.com
- 9- Tutoriels Mikroelektronika
www.microelektronika.com

www.lynxmotion.com

www.amel.com

Mots-clés :

Robotique, Robot, Bras manipulateur, FPGA, VHDL, modèle géométrique, modèle géométrique inverse, Commande hybride en force position, ATxmega, Altium Designer, schematics, openbus, Xilinx, Servomoteurs.