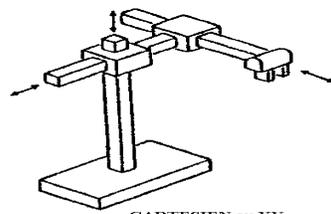


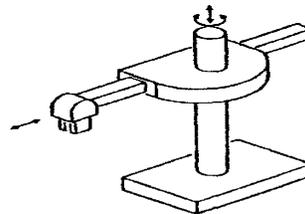
HAMDI HOCINE

INTRODUCTION A LA ROBOTIQUE

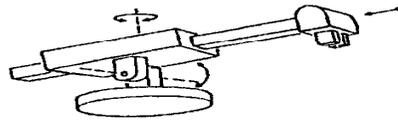
nouvelle édition revue augmentée et corrigée



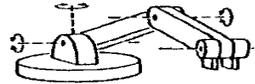
CARTESIEN ou XY



CYLINDRIQUE

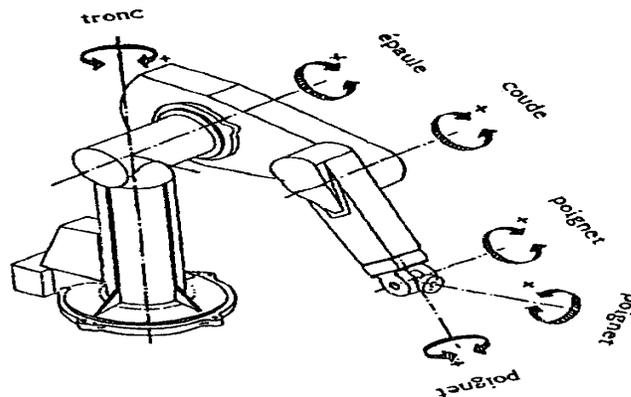


SPHERIQUE



ROTOIDE
ou de révolution

Structures des porteurs les plus utilisés



Robot PUMA à 6 degrés de liberté

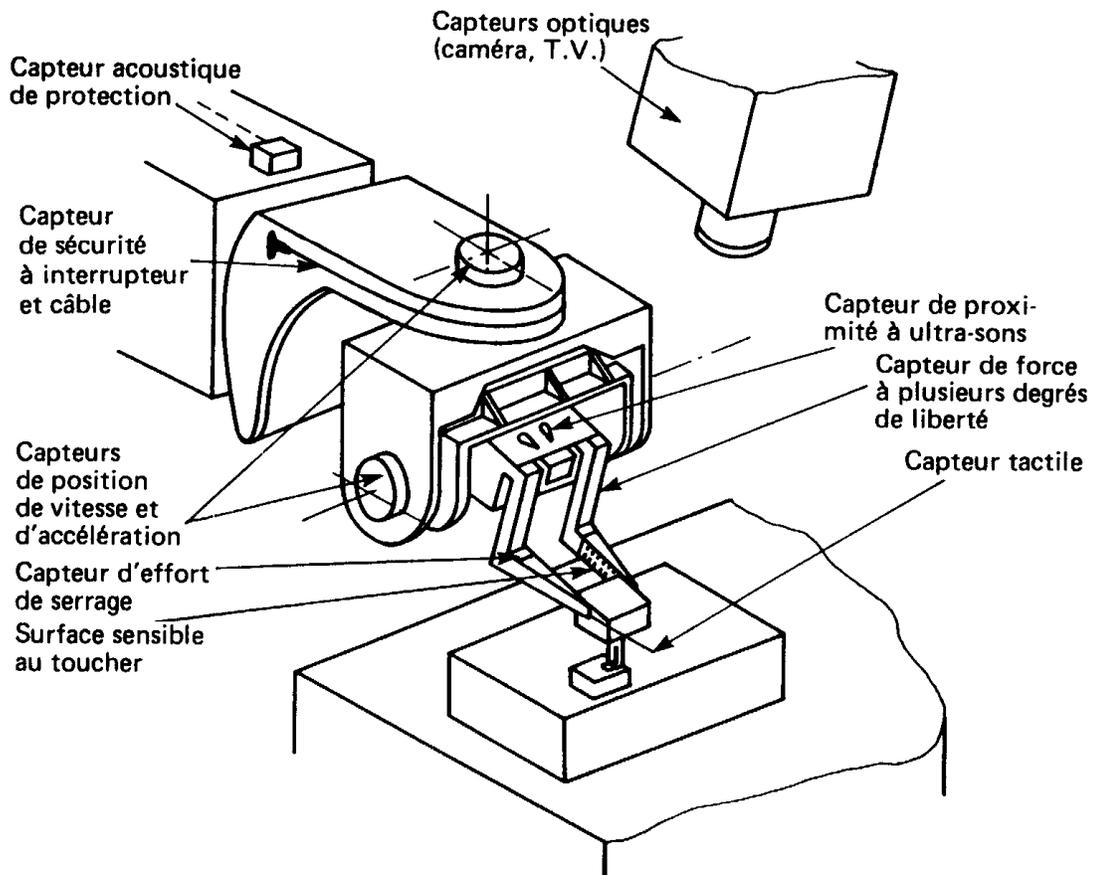
Les éditions de l'université Mentouri Constantine

2002-2003

HAMDI HOCINE

INTRODUCTION A LA ROBOTIQUE

nouvelle édition revue augmentée et corrigée



Exemple de mise en place d'un équipement sensoriel (doc. IFS Publications)

Les éditions de l'université Mentouri_Constantine

2002-2003

SOMMAIRE

CHAPITRE I : STRUCTURE GENERALE D'UN ROBOT	PAGE
-I- INTRODUCTION	1
-II- CLASSIFICATION	6
-III- ENSEMBLES CONSTITUANT UN ROBOT	8
-IV- MORPHOLOGIE ET DEGRES DE LIBERTE	11
CHAPITRE II : MODELISATION GEOMETRIQUE ET COMMANDE EN POSITION	
-I- RAPPELS MATHEMATIQUES	14
-II- DESCRIPTION ET MODELISATION GEOMETRIQUE D'UN SYSTEME MECANIQUE ARTICULE	17
-III- COMMANDE EN POSITION	25
-IV- AUTRES TYPES DE MMODELISATION	29
-V- EXEMPLES DE MODELISATION	
CHAPITRE III : PROGRAMMATION DES ROBOTS	
-I- GENERALITES ET OBJECTIFS DES SYSTEMES DE PROGRAMMATION	30
-II- METHODES DE PROGRAMMATION	31
-III- CARACTERISTIQUES DES DIFFERENTS LANGAGES DE PROGRAMMATION	36
-IV- EXEMPLE DE TACHE PROGRAMME E EN LANGAGE DE NIVEAU OBJET	40
CHAPITRE IV : INTERACTION ROBOT – ENVIRONNEMENT	
-I- LES CAPTEURS EXTEROCEPTIFS	47
-II- LES CAPTEURS INTEGRES OU SENSORIELS	52
-III- EXEMPLE DE TACHE METTANT EN ŒUVRE UN SYSTEME DE VISION	56
CHAPITRE V : CONCLUSION	61
ANNEXES :	
-ANN1- MODELISATION GEOMETRIQUE DU SMA ROBY1	63
-ANN2- MODELISATION GEOMETRIQUE DU SMA IR52C	100
-ANN3 –SUJETS D'EXERCICES DE MODELISATION	124
-ANN4-SOLUTIONS DES EXERCICES	131

- CHAPITRE I -
- STRUCTURE GENERALE D'UN ROBOT -

I-INTRODUCTION

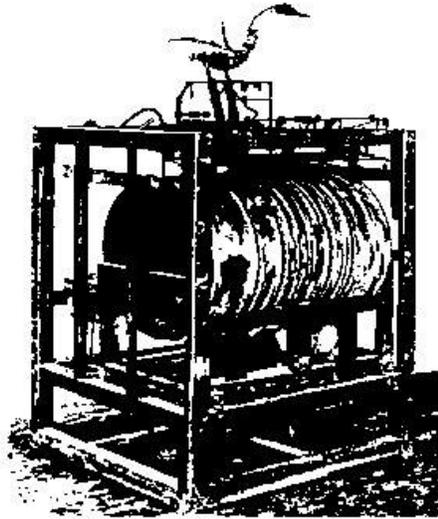
-1- Généralités

L'homme a depuis toujours été fasciné par la possibilité de pouvoir recréer la vie. Désir profond de jouer les créateurs, ou bien tentation de la mystification, ou encore simple jeu de copie de l'existant, les divers hommes d'airain, androïdes ou robots, créés dans la littérature, au cinéma ou dans la réalité, ont depuis la haute antiquité répondu à nos fantasmes.

Un passage des "Contes des mille et une nuits" nous entraîne chez la reine Abrizah où des statues parlaient et chantaient en gesticulant (49^e nuit). De nombreux contes chinois font aussi intervenir des androïdes: ainsi l'histoire de cet inventeur qui construisit pour un roi "un homme mécanique, d'une intelligence sans égale, capable de chanter, de danser et de se mouvoir comme un être humain". Héron qui vivait à Alexandrie vers la fin du deuxième siècle, décrit plus de cent appareils mécaniques parmi lesquels un groupe de statues dans une scène de sacrifices. Il est difficile dans ces textes de faire la part entre le merveilleux et ce qui a pu être réalité.

Ce n'est qu'avec le développement de l'horlogerie au XVI^e siècle que les automates ont pu acquérir une certaine complexité dans leurs mouvements. En effet tous les mécanismes précédents devaient fonctionner sur le principe des clepsydres, c'est à dire à partir d'un écoulement régulier d'eau (éventuellement remplacée par du mercure ou du sable). Les développements de la mécanique et de l'horlogerie entraînèrent la création de ces merveilleux automates qui font toujours l'admiration de tous, car quelques exemplaires ont pu survivre plusieurs siècles et arriver jusqu'à nous.

Il est cependant nécessaire de faire la distinction entre deux types d'automates: les automates cycliques qui exécutent un même mouvement de façon répétitive, et les automates à programme dont le mouvement est beaucoup plus riche. Dans le premier cas, le mouvement est obtenu par une mécanique figée qu'il est très difficile de modifier. Par contre, dans le deuxième cas les mouvements sont programmés à partir d'un dispositif central qui décrit leur séquence et éventuellement leur amplitude. Ces dispositifs débutèrent par des cylindres munis de picots ou de cames, pour évoluer vers les disques, rubans et cartes perforées.



Canard de Vaucanson et son tambour à came (Tambour à cames dirigeant tout le mécanisme)

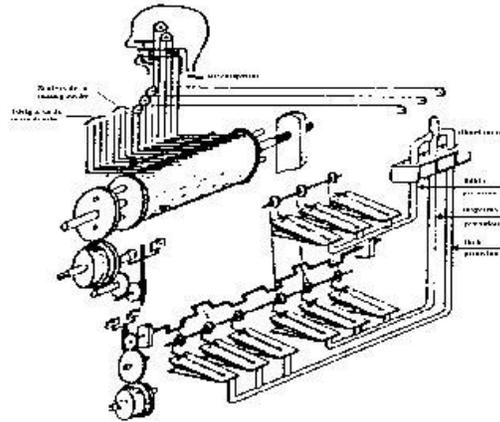
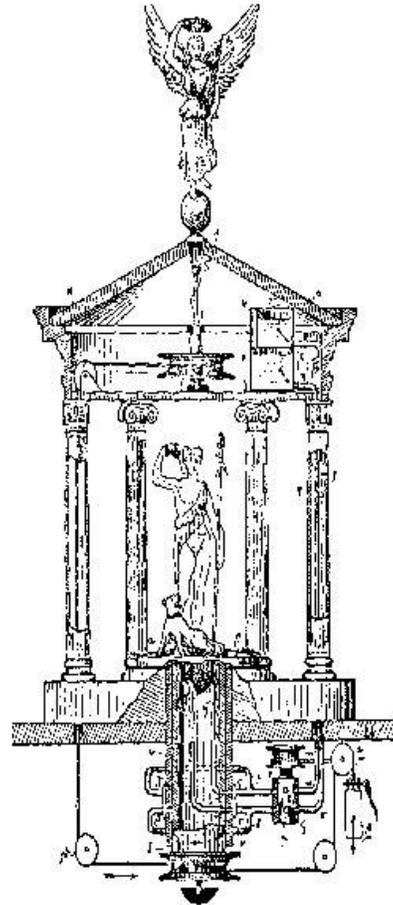


Schéma du Flûteur d'après la description de Vaucanson



Les trois automates de Vaucanson



Coupe partielle du Théâtre de Héron d'Alexandrie

Figure 1.1: Exemples d'automates « ancêtres » des robots

Il est difficile de dire à quand remonte ce concept de programme, et en particulier s'il existait sur les automates de l'antiquité. Les premières applications connues avec certitude datent du XVI^e siècle avec les horloges à personnages intervenant dans des scènes complexes et variables avec le temps: les Jacquemarts. Cette technique conduisit tout naturellement aux boîtes à musique (XVII^eS), et aux automates musiciens et dessinateurs (XVIII^eS), dont certains exemplaires sont encore conservés dans des musées. Les techniques introduites et leurs inventeurs ont largement participé à la mécanisation et l'automatisation de l'industrie, en particulier dans l'industrie textile on a vu apparaître des métiers à tisser entièrement automatiques.

Le XVIII^e siècle a ainsi vu démarrer le processus de mécanisation de l'industrie, qui passa d'un stade artisanal à un stade de forte concentration en moyens physiques et humains, l'amélioration de la productivité restant un souci constant.

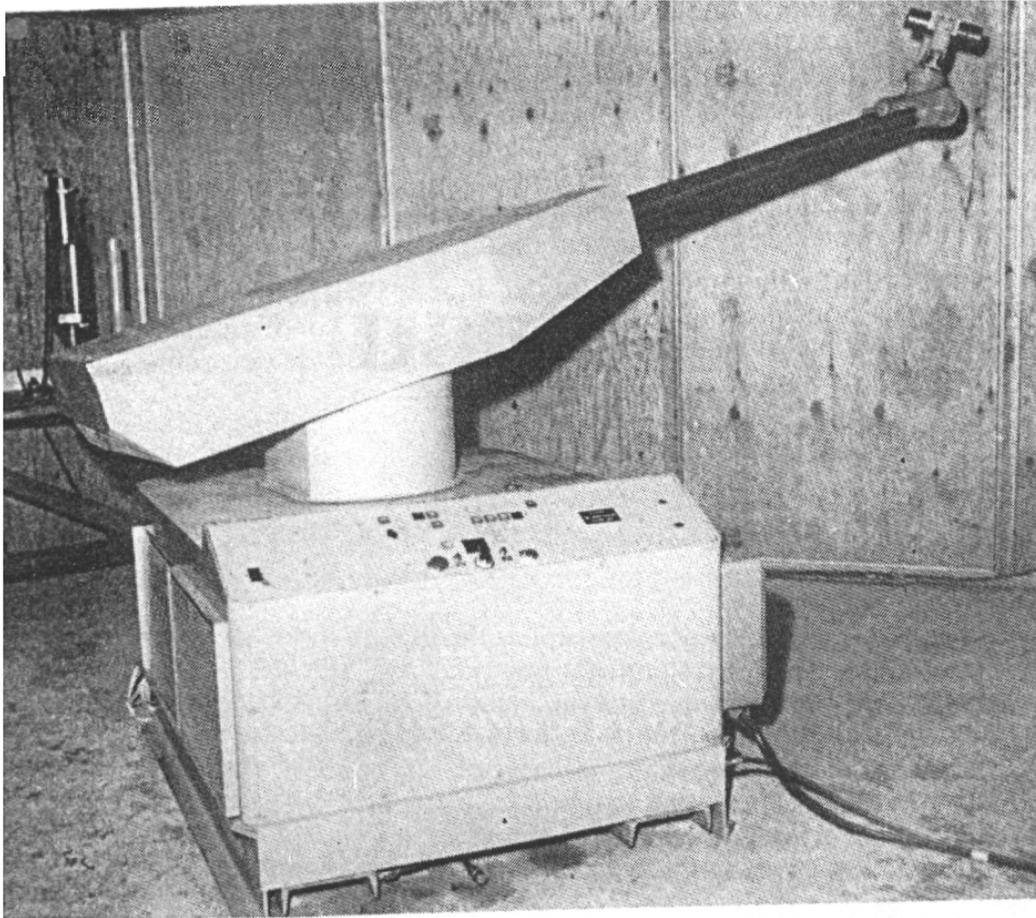
Malgré toutes les découvertes et inventions du XVIII^e siècle à nos jours, les deux grandes étapes du processus d'automatisation furent d'abord l'utilisation de la puissance mécanique (avec d'abord la machine à vapeur, puis plus tard les machines électriques), puis très récemment l'introduction de l'informatique dans les machines (avec toutes ses possibilités de stockage de l'information, et surtout de traitement de signaux variés).

C'est en effet grâce aux possibilités de stockage de l'information, sur des moyens électriques et électroniques, que les robots dits de "première génération " ont pu voir le jour avec le premier UNIMATE de G.DEVOL.

-2- Introduction

Le concept de robot date de plusieurs siècles, mais le terme robot fut inventé par le tchèque Karel Capek dans une pièce de théâtre écrite en 1920: "RUR ou les robots universels de Rossum". Ce terme est dérivé du verbe tchèque "robota" signifiant travail forcé ou corvée.

Il est certain que depuis fort longtemps, les réalisateurs d'automates ont cherché à pouvoir insuffler à leurs machines des comportements adaptés aux circonstances. Malheureusement, jusqu'au vingtième siècle, les techniques étaient trop primitives pour permettre de telles réalisations. Il fallut attendre 1959 pour que Georges Devol invente une machine originale, polyvalente et reprogrammable, ce qui a permis au robot d'acquérir une réalité industrielle. Mais en fait ce ne fut que vers la fin des années 1970 que les robots industriels de première génération ont vu le jour.



Prototype UNIMATE (1959), Robot et tambour programme

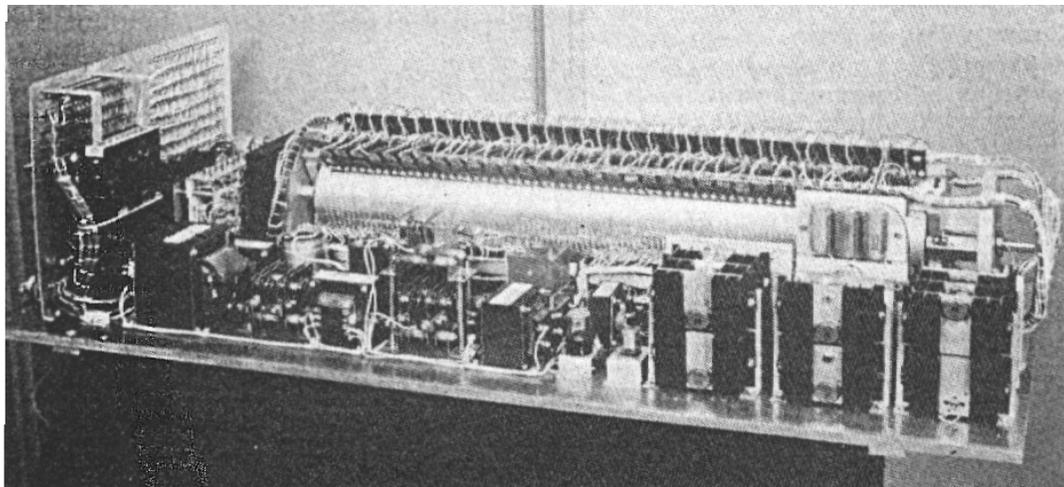


Figure 1-2: premier robot "industriel" de G.Devol

Depuis l'apparition des premiers robots industriels, leur essor a été considérable, et chaque année de nouveaux secteurs industriels s'ouvrent à la robotisation au fur et à mesure que les équipements se diversifient, deviennent plus adaptables et surtout meilleur marché. Les robots sont devenus tellement indispensables dans certains secteurs industriels (l'automobile par exemple), que leur emploi conditionne la survie économique des entreprises. Il devient donc important de maîtriser leur technologie.

Le terme robot peut désigner une large variété de réalisations technologiques, allant du simple dispositif mécanique exécutant des mouvements répétés, aux machines analogues morphologiquement aux bras humains et possédant une certaine intelligence.

A titre d'anecdote, la firme Unimation qui construisit le premier robot vers 1960, était pratiquement la seule sur le marché américain quinze ans après, et n'avait commercialisé que 1000 robots. Alors que le JIRA (association japonaise de robotique industrielle) annonçait en 1975 65000 robots?! Il s'est avéré par la suite que ce que les uns désignaient par simple automatisme, les autres l'appelaient robot!

Aujourd'hui il existe au Japon plus de 15000 robots industriels, soit environ près du tiers du parc mondial.

-II-CLASSIFICATION

L'AFRI (Association Française de Robotique Industrielle) distingue quatre classes de robots:

-1- Les Télémanipulateurs ou manipulateurs à commande manuelle

Ils sont commandés à distance et "en temps réel" par un opérateur humain. Cette télécommande se fait à plus ou moins longue distance par signaux mécaniques, hydrauliques, ou le plus souvent électriques. Ces manipulateurs sont employés en forge, fonderie, meulage-ébarbage, milieux "hostiles", etc..., mais nécessitent toujours la présence et l'intervention constante d'un opérateur.

-2- Les Manipulateurs automatiques à cycles prééglés

Leurs mouvements sont limités par des butées et cames réglables à la main. Ils sont commandés à l'aide de logiques à relais ou pneumatiques (séquences fixes), ou par automates programmables et cartes à microprocesseurs (séquences variables). Généralement modulaires, ces appareils sont conçus pour une application déterminée.

-3- Les robots programmables

Ils sont pilotés par des ordinateurs ou des armoires de commande numérique. Leurs mouvements continus dans l'espace sont alors programmés par apprentissage ou en langage symbolique par l'intermédiaire d'un clavier, ou encore sur l'écran d'un poste de CAO. Ils assurent des manipulations complexes, des opérations de soudage, usinage, découpe, peinture et pulvérisation, etc...

-4- Les robots dits "intelligents"

Equipés de capteurs (par exemple un système de vision artificielle ou de suivi de joint en soudage), ils peuvent analyser les modifications de leur environnement ou de leur trajectoire et réagir en conséquence. Ces machines appelées robots de "deuxième génération" commencent à être répandus dans l'industrie. La "troisième génération" disposant de capacités de raisonnement grâce à l'intelligence artificielle fait aujourd'hui l'objet de recherches approfondies.

-5- Définition du robot industriel

Parallèlement à cette classification, il existe une définition du robot industriel normalisée et publiée par l'AFNOR (institut français de normalisation): "Manipulateur commandé en position, reprogrammable, polyvalent, à plusieurs degrés de liberté capable de manipuler des matériaux, des pièces, des outils ou des dispositifs spécialisés au cours de mouvements variables, et programmé pour l'exécution d'une variété de tâches, il a souvent

l'apparence d'un ou plusieurs bras se terminant par un poignet. Son unité de commande utilise notamment un dispositif de mémoire et éventuellement de perception et d'adaptation à l'environnement et aux circonstances".

Il est à remarquer qu'aucune classification n'est satisfaisante, car aucune ne fait intervenir des critères techniques sans ambiguïté pour passer d'une classe à l'autre. Seules les deux dernières catégories de la classification de l'AFRI correspondent à la définition de l'AFNOR. C'est cette catégorie de robots qui nous intéresse dans la cadre de ce cours.

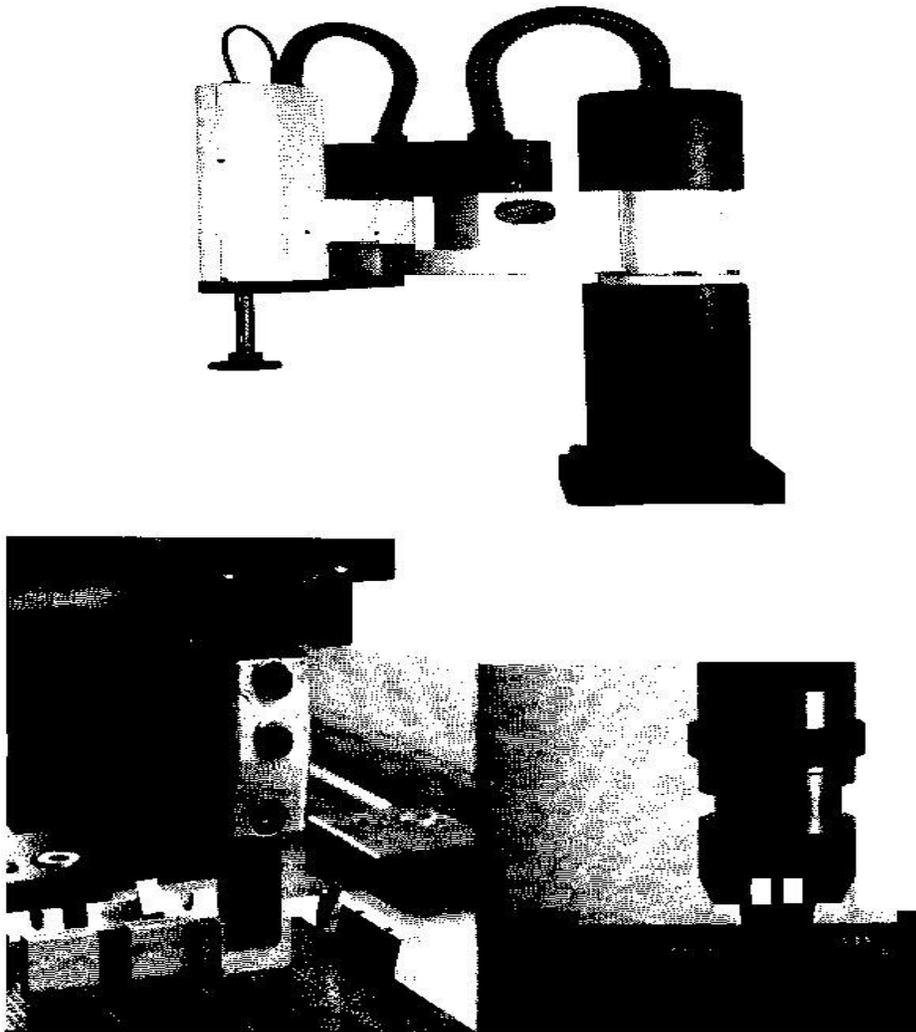


Figure 1.3: Exemple de robot industriel (IBM 7575)
utilisation dans une application d'insertion de composants électroniques

-III-ENSEMBLES CONSTITUANT UN ROBOT

"De même qu'un trou n'est défini que par ce qui l'entoure, un robot n'a de sens que dans un environnement qu'il modifie" (P.Coiffet). C'est pourquoi on peut distinguer quatre ensembles interactifs dans un robot en fonctionnement.

-1- Le système mécanique articulé (SMA)

Il s'agit d'une structure mécanique articulée dotée de plusieurs degrés de liberté (généralement six pour un robot destiné à la manutention d'objets), et équipée de moteurs qu'on appelle aussi actionneurs. Ces derniers peuvent être électriques ou pneumatiques pour les faibles charges (jusqu'à quelques décanewtons), ou hydrauliques pour les fortes charges (quelques centaines de newtons). Ils entraînent les articulations du manipulateur à travers des transmissions, elles aussi de natures diverses (câbles, courroies crantées, engrenages, bielles etc...).

Pour connaître à chaque instant la position des articulations, on fait appel à des capteurs (potentiomètres, codeurs optiques ou incrémentaux, "resolvers", etc...) appelés «proprioceptifs» par analogie avec le système humain. Le système mécanique articulé peut être considéré à la fois comme un générateur de déplacements de l'outil dans l'espace géométrique, et un générateur de forces.

Cette première entité correspond à ce qui est généralement montré dans une photographie de robot, et ce à quoi on pense dans la conversation courante quand on parle de robot. Pour notre part, nous emploierons indifféremment le terme de manipulateur ou de système mécanique articulé (SMA).

-2- L'environnement

C'est l'univers dans lequel est immergée la machine. Pour les robots à poste fixe, il se réduit à ce que l'on rencontre dans l'espace atteignable du manipulateur, défini comme l'ensemble des points atteints par l'effecteur lorsque le manipulateur passe par toutes les configurations géométriques possibles.

Dans cet environnement, le manipulateur va rencontrer des obstacles qu'il doit éviter et des objets d'intérêt, c'est à dire sur lesquels il doit agir. Par conséquent il existe une interaction entre le SMA et l'environnement.

Des informations sur l'état de l'environnement peuvent être obtenues grâce à des capteurs qu'on appelle "extéroceptifs", c'est à dire permettant de situer ce qui est à l'extérieur du manipulateur. On utilise des caméras, des détecteurs et capteurs de proximité, des capteurs tactiles etc...

-3- Les tâches à réaliser

On peut les définir de manière générale comme la différence entre deux états de l'environnement: l'état de départ d'exécution de la tâche, et l'état final lorsque cette dernière est terminée. La difficulté essentielle est de définir et de modéliser l'environnement.

Les tâches sont décrites au robot dans un langage approprié. Cette description peut revêtir diverses formes , et certains paramètres nécessaires à l'exécution peuvent être acquis en cours d'exécution.

Le support des langages peut être "gestuel" (on montre au robot ce qu'il faut faire par guidage), oral (encore du domaine de la recherche et les seules applications connues sont destinées aux handicapés moteurs), ou scriptural (on "écrit" dans un langage formalisé).

-4- Le contrôleur ou calculateur

C'est l'organe qui génère les commandes (signaux de puissance des actionneurs), qui vont induire les déplacements recherchés des différentes articulations du manipulateur, ou les forces appliquées sur les objets; et ce à partir d'informations à priori (connaissance de la tâche à exécuter) et à postériori (connaissance actuelle et passée du manipulateur et de l'environnement).

Le contrôle du robot inclut les fonctions qui lui permettent "d'apprendre" et d'être programmé pour une tâche spécifique, puis d'exécuter cette tâche. La séquence de mouvements, le type de mouvement entre deux points, et l'interaction avec les équipements externes sont toutes des parties de la fonction de contrôle.

Pour effectuer ses différentes fonctions, le contrôleur doit avoir dans ses mémoires:

- un modèle du manipulateur: c'est à dire les relations entre les signaux d'excitation des actionneurs et les déplacements du manipulateur qui en sont la conséquence;

- un modèle de l'environnement: qui est une description de ce qui se trouve dans l'espace atteignable;

- des programmes: comprenant les données relatives à la tâche à exécuter ainsi qu'un certain nombre d'algorithmes de commande.

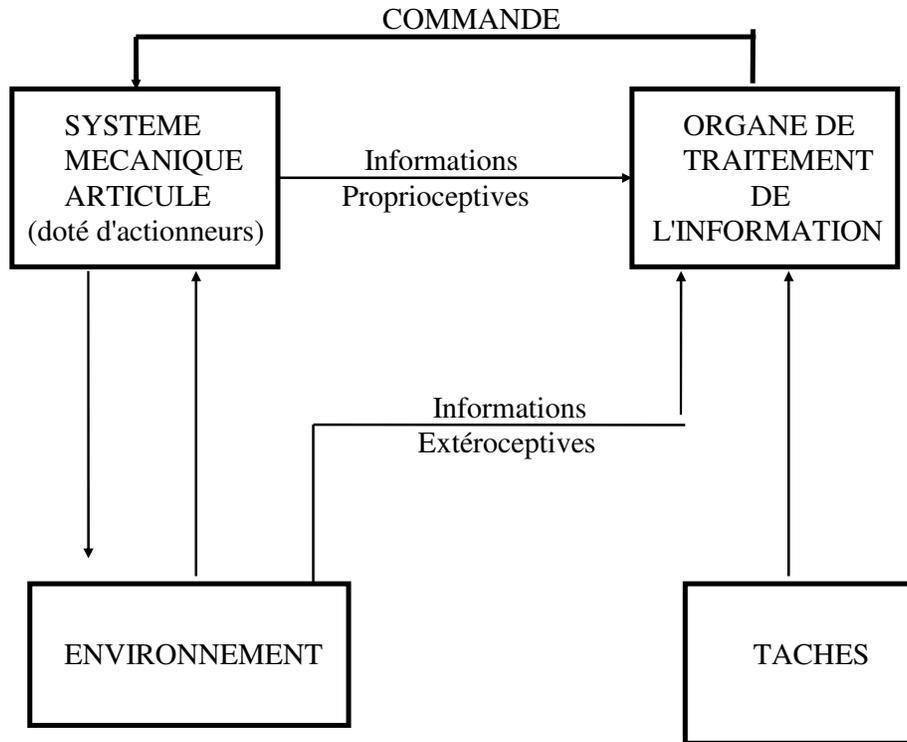


Figure 1-4: Ensembles intervenant dans un robot en fonctionnement

(source:P.coiffet;"les robots, tome 1», éditions Hermès)

-III-MORPHOLOGIE ET DEGRES DE LIBERTE

-1- Morphologie

Un manipulateur ayant pour vocation de modifier son univers par l'intermédiaire d'outils qu'il utilise correctement sur ordre, il est alors aisé de distinguer dans sa structure géométrique trois sous ensembles interconnectés et couplés mécaniquement.

1-a-Le véhicule

Pour qu'un manipulateur exécute des tâches dans un lieu donné, il faut d'abord l'amener sur place. C'est le rôle du véhicule: engin terrestre, sous-marin, satellite etc... (ayant entre deux et six degrés de liberté suivant le milieu dans lequel il se déplace). Pour les robots à poste fixe, il n'y a pas de véhicule et la base du manipulateur est fixée sur le lieu de travail. Ceci est le cas de la quasi totalité des robots industriels, hormis une catégorie de robots à portique où l'ensemble portique-roues peut être considéré comme un véhicule.

-1-b-Le porteur

Le porteur représente l'essentiel du système mécanique articulé; il a pour rôle d'amener l'organe terminal dans une situation donnée imposée par la tâche (la situation d'un corps peut être définie comme la position et l'orientation d'un repère attaché à ce corps par rapport à un repère de référence). Il est constitué de :

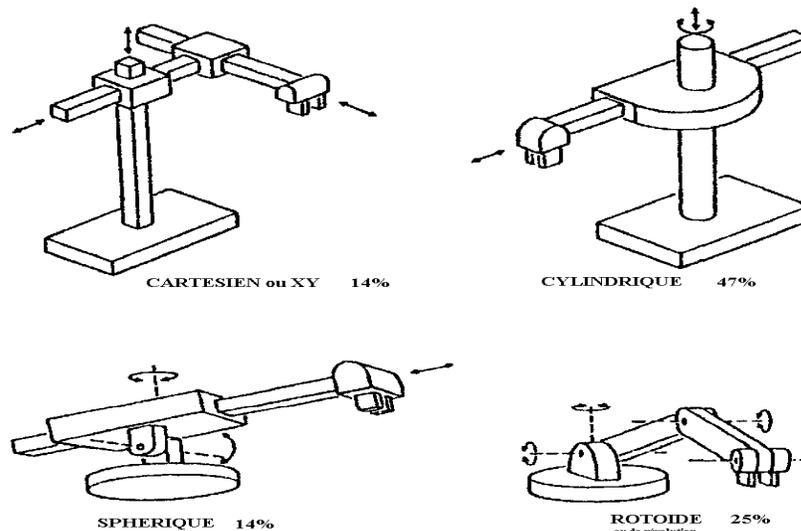
- segments: corps solides rigides susceptibles d'être en mouvement par rapport à la base du porteur, et les uns par rapport aux autres;
- articulations: système de liaisons mécaniques (rotoïdes ou prismatiques) limitant les possibilités de mouvements relatifs entre deux segments adjacents.

Selon l'arrangement de ces articulations, on peut obtenir différentes architectures possibles pour le porteur. Une étude de E.DOMBRE portant sur 115 robots a montré que si l'on considère les trois premières articulations (cf fig 5-a), seuls quatre arrangements principaux sont utilisés: robots cartésiens (3translations, 14%), cylindriques (1translation+1rotation+1translation, 47%), sphériques (2rotations+1translation, 13%), rotoïdes ou de révolution (3rotations, 25%).

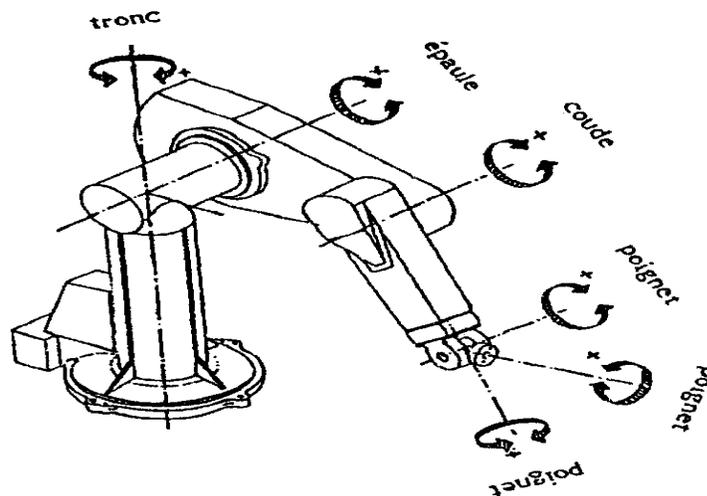
1-c-L'organe terminal ou effecteur

L'organe terminal désigne l'outil (une pince par exemple) qui est censé agir sur les objets. C'est lui qui impose au porteur ses exigences: positionnement et orientation vis à vis de l'environnement qu'il doit modifier ainsi que les forces à appliquer. C'est également une structure mécanique articulée avec ses propres degrés de liberté, qui ne sont pas comptabilisés dans le nombre de degrés de liberté du système.

A cette division en véhicule, porteur et organe terminal, on préfère le plus souvent une description "morphologique" se rapprochant de celle de l'être humain; à savoir une division en tronc, bras, avant bras, poignet et main (la main constituant l'outil et se réduisant le plus souvent à une pince à deux ou trois doigts). Des mains munies de capteurs (tactiles, de pression, ou de proximétrie) ont été développées, et on rencontre de plus en plus souvent des poignets munis de capteurs permettant de mesurer les forces et les couples exercés (cas de la commande en effort ou compliance).



5-a-Structures des porteurs les plus utilisés



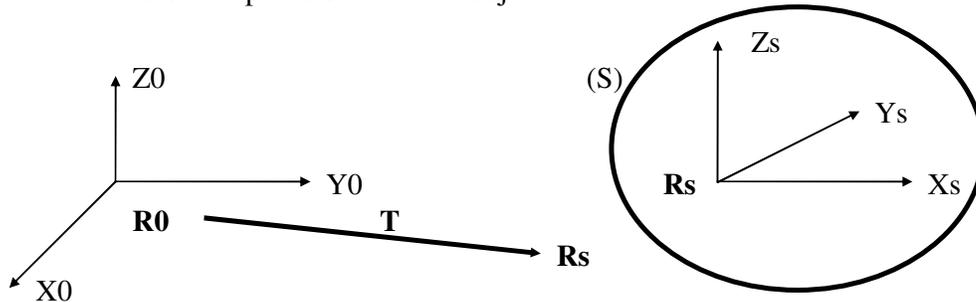
5-b-Robot PUMA à 6 degrés de liberté

Figure 1-5: Morphologie d'un manipulateur

-2- Degrés de liberté

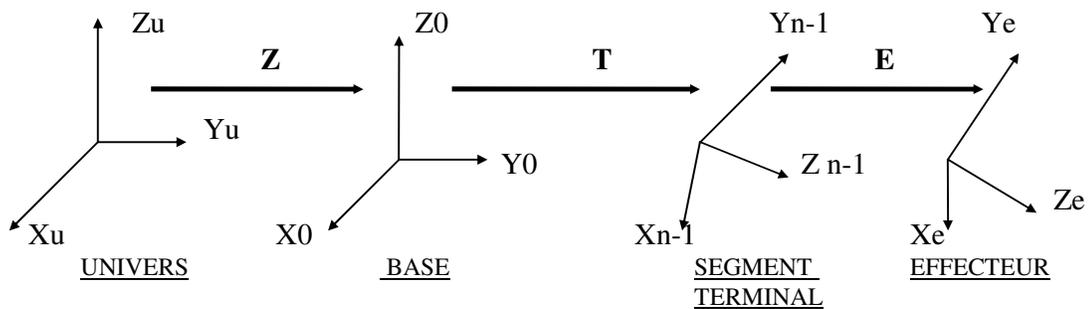
2-a-Degrés de liberté d'un corps dans l'espace

Soit R_s un repère attaché à un corps rigide (S). Le nombre de degrés de liberté maximum de ce corps est le nombre de paramètres indépendants nécessaires pour déterminer complètement à tout instant, la transformation géométrique T permettant le passage du repère de référence R_0 au repère R_s attaché à l'objet.



2-b-Degrés de liberté du porteur

Soit un manipulateur à n segments rigides reliés par des articulations. Le nombre de degrés de liberté du porteur est le nombre de paramètres indépendants nécessaires pour exprimer la transformation T faisant passer du repère de base (R_0) du manipulateur au repère du dernier segment (R_{n-1}) sur lequel est fixé, l'outil: $T: R_0 \rightarrow R_{n-1}$



Ce nombre de degrés de liberté est égal (ou inférieur) au nombre d'articulations indépendantes. On désigne par variable articulaire le paramètre (angulaire ou de translation) associé à chaque articulation.

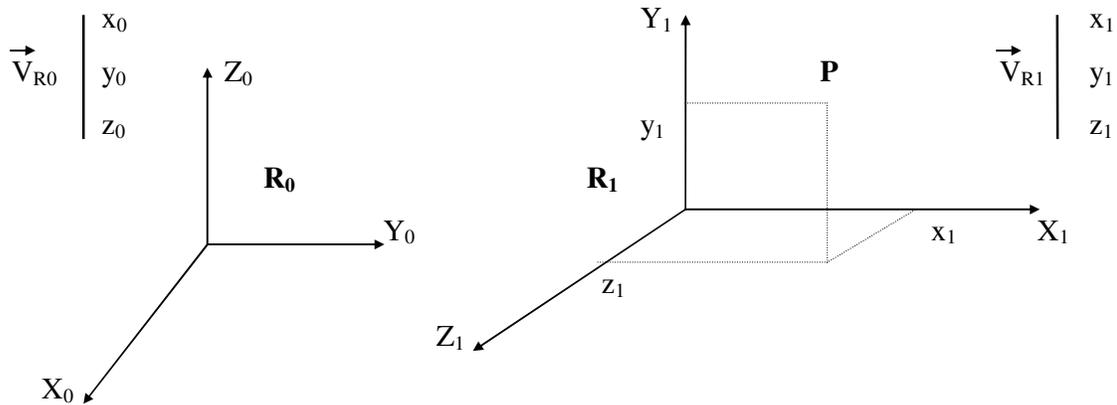
Pour spécifier complètement la situation d'un corps dans l'espace, six paramètres sont nécessaires: 3 coordonnées pour localiser son centre de gravité, et trois angles pour spécifier ses rotations par rapport à un axe passant par son centre de gravité. Une manipulation complètement générale de ce corps nécessite donc que le manipulateur ait six degrés de liberté.

- CHAPITRE II -
- MODELISATION GEOMETRIQUE ET COMMANDE EN POSITION -

-I- RAPPELS MATHÉMATIQUES

-1- Orientation d'un corps par rapport à un autre

-a-Changement de repère



- Figure 2-1 -

Désignons par $\vec{V}(R_0)$ et $\vec{V}(R_1)$ les coordonnées de \vec{V} dans les repères R_0 et R_1 .

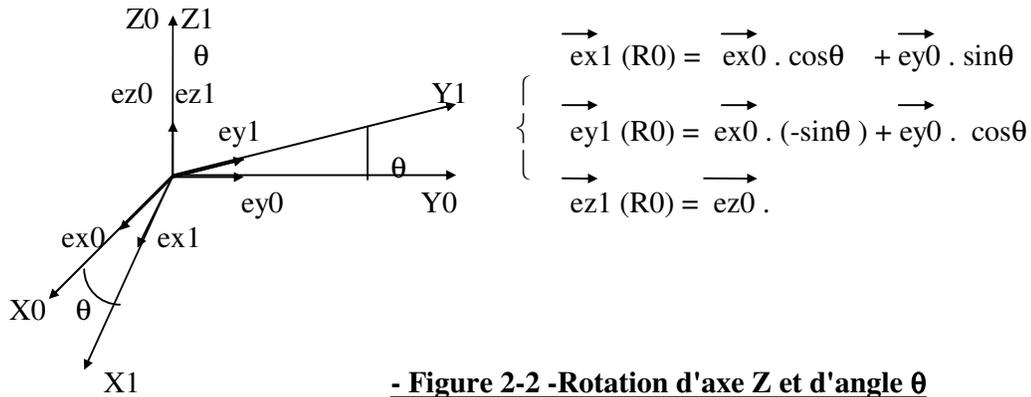
$$\left. \begin{aligned} \vec{V}(R_0) &= M_0^1 \times \vec{V}(R_1) \\ \vec{V}(R_1) &= M_1^0 \times \vec{V}(R_0) \end{aligned} \right\} \begin{aligned} &\text{avec } M_0^1 : \text{matrice de passage du repère } R_0 \text{ au repère } R_1. \\ &\text{Elle exprime la position du repère } R_1 \text{ par rapport au} \\ &\text{repère } R_0. \\ &M_1^0 = [M_0^1]^{-1} \end{aligned}$$

-b-Orientation et cosinus directeurs

Si on considère deux repères ayant des origines confondues, la matrice de passage M_0^1 se réduit à une matrice d'orientation, qui définit l'orientation d'un repère par rapport à un autre (dans notre cas R_1 par rapport à R_0). Elle correspond aux projections (cosinus directeurs) des trois vecteurs unitaires de base du repère considéré (R_1) dans le repère de référence (R_0).

Dans ce cas $M_1^0 = [M_0^1]^{-1} = [M_0^1]^t$.

1°Cas: R1 est obtenu à partir de R0 par rotation d'un angle θ autour de l'axe Z .



- Figure 2-2 -Rotation d'axe Z et d'angle θ

$${}^1 M_0 = \begin{matrix} \vec{ex1} & \vec{ey1} & \vec{ez1} \\ \vec{ex0} & \text{Pos11} & \text{Pos12} & \text{Pos13} \\ \vec{ey0} & \text{Pos21} & \text{Pos22} & \text{Pos23} \\ \vec{ez0} & \text{Pos31} & \text{Pos32} & \text{Pos33} \end{matrix} = \begin{matrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{matrix} \begin{matrix} \vec{ex0} & \vec{ey0} & \vec{ez0} \\ \vec{ex1} & \text{Pos11} & \text{Pos12} & \text{Pos13} \\ \vec{ey1} & \text{Pos21} & \text{Pos22} & \text{Pos23} \\ \vec{ez1} & \text{Pos31} & \text{Pos32} & \text{Pos33} \end{matrix} = \begin{matrix} \vec{ex1} & \vec{ey1} & \vec{ez1} \\ \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{matrix}$$

$$\begin{matrix} \vec{ex1} \\ \vec{ey1} \\ \vec{ez1} \end{matrix} = \begin{matrix} \vec{ex0} \\ \vec{ey0} \\ \vec{ez0} \end{matrix} \begin{matrix} t \\ x \\ M_0 \end{matrix} \Leftrightarrow \mathbf{R1} = \mathbf{R0} * \mathbf{Rot}(Z, \theta)$$

cette écriture exprime une transformation géométrique pour passer de R0 à R1.

$$[\vec{ex1} \ \vec{ey1} \ \vec{ez1}]^t = M_1 * [\vec{ex0} \ \vec{ey0} \ \vec{ez0}]^t \Leftrightarrow V(R1) = M_1 * V(R0)$$

$$[\vec{ex0} \ \vec{ey0} \ \vec{ez0}]^t = M_0 * [\vec{ex1} \ \vec{ey1} \ \vec{ez1}]^t \Leftrightarrow V(R0) = M_0 * V(R1)$$

2°Cas: R1 est obtenu à partir de R0 par une rotation d'angle θ autour de l'axe Y :

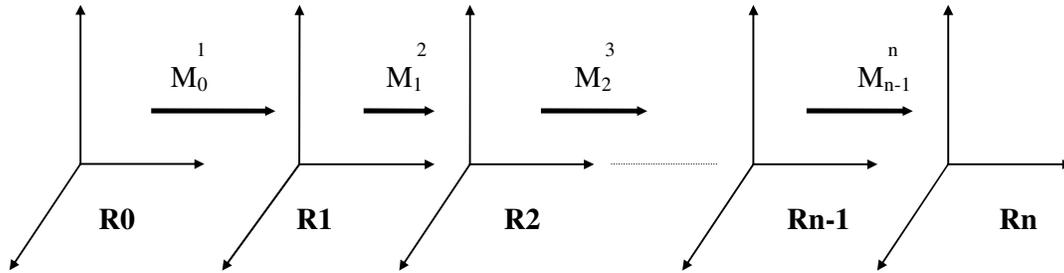
$$R1 = R0 * \text{Rot}(Y, \theta) \Leftrightarrow M_0 = \begin{matrix} \cos\theta & 0 & \sin\theta \\ 0 & 1 & 0 \\ -\sin\theta & 0 & \cos\theta \end{matrix}$$

3°Cas: R1 est obtenu à partir de R0 par une rotation d'angle θ autour de l'axe X :

$$R1 = R0 * \text{Rot}(X, \theta) \Leftrightarrow M_0 = \begin{matrix} 1 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta \\ 0 & \sin\theta & \cos\theta \end{matrix}$$

-2- Changements multiples de repères

Remarque préliminaire: $\vec{V}(R_0) = M_0^1 * \vec{V}(R_1)$. Il faut lire cette relation géométriquement de droite à gauche. C'est à dire que les coordonnées de \vec{V} dans le repère R_0 , sont données par les coordonnées de \vec{V} dans R_1 , multipliées par la position de R_1 par rapport à R_0 .



-Figure 2-3- Changements de repères successifs

$$\left. \begin{array}{l} \vec{V}(R_0) = M_{R_0}^{R_1} * \vec{V}(R_1) \\ \vec{V}(R_0) = M_{R_0}^{R_n} * \vec{V}(R_n) \end{array} \right\} \Rightarrow M_{R_0}^{R_n} = M_0^1 * M_1^2 * \dots * M_{n-1}^n$$

Exemple: n = 3

Soient R_0, R_1, R_2 et R_3 quatre repères successifs.

$$\begin{aligned} M_{R_0}^{R_3} &= M_{R_0}^{R_2} * M_{R_2}^{R_3} & M_{R_0}^{R_3} &= M_{R_0}^{R_1} * \begin{bmatrix} M_{R_1}^{R_2} & M_{R_1}^{R_3} \\ M_{R_2}^{R_3} \end{bmatrix} = M_{R_0}^{R_1} * M_{R_1}^{R_3} \\ \} &\Rightarrow \\ M_{R_0}^{R_2} &= M_{R_0}^{R_1} * M_{R_1}^{R_2} & M_{R_0}^{R_3} &= M_{R_0}^{R_2} * M_{R_2}^{R_3} \equiv M_{R_0}^{R_1} * M_{R_1}^{R_2} * M_{R_2}^{R_3} \end{aligned}$$

Attention!

Il faut disposer de gauche à droite toutes les matrices dans l'ordre des transformations géométriques ce qui correspond à la **prémultiplication des transformations géométriques**, puis faire le produit des matrices en les assemblant deux à deux (et en commençant par la droite de préférence) ce qui correspond à la **postmultiplication des matrices** (cf page 21).

En réalité on peut assembler les matrices comme on veut (car le produit matriciel est associatif) pourvu qu'on ne change pas l'ordre des matrices (produit matriciel non commutatif).

-II- DESCRIPTION ET MODELISATION GEOMETRIQUE D'UN S.M.A.

A l'aide d'un langage de description, qu'il soit graphique ou symbolique, le but poursuivi est de faire de la génération automatique de modèles par programmation.

-1- Description

La description d'un SMA est constituée: d'une part d'un outil de représentation, et d'autre part d'un ensemble de données ou paramètres caractéristiques du système : longueurs des segments, masses, inerties, degrés de liberté (ddl), agencement des ddl etc...

Parmi les nombreux outils de description, on citera ceux de :

- Roth-Pieper: permet l'élaboration d'un modèle géométrique ou d'un modèle cinématique, mais ne conduit pas à un modèle dynamique. Son principe de base repose sur le fait qu'à chaque articulation, nous n'avons qu'un seul mouvement possible: une rotation ou une translation. Elle ne s'applique donc pas aux articulations multiples. Cette méthode de description est surtout utilisée pour la comparaison des SMA .
- Wisama Khalil: conduit directement à la programmation et permet l'élaboration du modèle dynamique. Cette méthode est considérée comme une généralisation de la méthode de Denavit-Hartenberg (dont elle s'inspire) sur le plan de la programmation, dans la mesure où elle est valable quel que soit le type de SMA .
- Borrel: de type linguistique, cette description ne s'applique qu'aux SMA arborescents (c'est à dire sans boucle).

-2- Modélisation géométrique

-a-Généralités

Parmi les paramètres d'un SMA, on distingue:

- les paramètres de structure: donnant les caractéristiques des segments et leur ordonnancement dans la configuration. Ce sont des **constantes**.
- les paramètres de commande: ce sont les **variables articulaires** de type rotoïde (rotation) ou prismatique (translation), permettant de modifier la géométrie du SMA. Elles sont liées aux actionneurs et à leur commande.

Considérons un SMA constitué de $n+1$ segments (numérotés de S_0 à S_n) et associons un repère à chaque segment. La modélisation géométrique consiste à exprimer la position et l'orientation de l'organe terminal (effecteur) par rapport au repère R_0 de la base du SMA (considérée comme le segment S_0).

n

Il s'agit donc d'établir la matrice de transformation $T = M_0$ du manipulateur. N'importe quel point de l'espace atteint par l'extrémité de l'effecteur sera donné par rapport au repère de référence de la tâche Run (repère de l'univers) par l'expression : $POS = Z * T * E$, où :

Z: désigne la matrice donnant la position et l'orientation du repère de la base R_0 par rapport au repère de référence de la tâche Run .

T: désigne la matrice de transformation du manipulateur. Elle donne la position et l'orientation de l'origine du repère R_n (associé au dernier segment S_n) par rapport au repère de la base R_0 (associé au segment S_0).

E: désigne la matrice de l'effecteur. Elle donne la position et l'orientation du repère R_E attaché à l'effecteur (généralement à son extrémité) par rapport au repère R_n .

Faisons abstraction provisoirement des matrices Z et E (qui sont généralement constantes), et travaillons uniquement sur la base de l'effecteur, c'est à dire le segment S_n le supportant.

Sachant qu'il faut au maximum m paramètres ($m \leq 6$) pour configurer un corps en position et en orientation dans l'espace, la *situation*¹ de la base de l'organe terminal sera donnée (par rapport à R_0) par un vecteur $\vec{X}(R_0) = F(q)$, q étant le vecteur des paramètres de configuration, c'est à dire les variables de commande associées aux moteurs des articulations. Le vecteur \vec{X} peut se décomposer en deux parties² :

- le bloc **P** correspondant à la "translation". Ce sont les paramètres de position x, y et z . $\vec{P} = (x, y, z)^t$. Ce bloc est généralement appelé **vecteur position P**;

- le bloc **R** correspondant à l'orientation et généralement appelé **vecteur orientation R**. Différents choix de paramètres d'orientation (vecteur R) sont possibles.

Remarques:

¹: Situation: ce terme désigne la position et l'orientation. Il sera utilisé dans toute la suite de ce chapitre.

²: La matrice POS correspondant au même point sera également composée de deux blocs: un vecteur position du même type que P , et une matrice (3x3) d'orientation qui est une fonction de R .

-b-Paramètres d'orientation

* Cosinus directeurs: Ce sont les cosinus directeurs (projections des vecteurs unitaires) du repère lié à l'effecteur. Ils expriment l'orientation de ce repère par rapport au repère de base R0.

$$\vec{X} = \begin{vmatrix} x & y & z \\ \vec{e}_1 & \vec{e}_2 & \vec{e}_3 \end{vmatrix}^t$$

* Angles d'Euler: Ils correspondent à trois rotations:

ψ : précession = rot(Zi-1, ψ_i): rotation d'axe Zi-1 et d'angle ψ_i

θ : nutation = rot(Xi-1, θ_i): rotation d'axe Xi-1 et d'angle θ_i

ϕ : rotation = rot(Zi-1, ϕ_i): rotation d'axe Zi-1 et d'angle ϕ_i

$$\vec{X} = \begin{vmatrix} x & y & z \\ \psi & \theta & \phi \end{vmatrix}^t$$

La matrice ROT (3x3) d'orientation définissant l'orientation du repère Rn dans le repère R0 s'écrit comme le produit des matrices d'orientation intermédiaires:

$$ROT_{R_0}^{R_n} = \prod_{i=1}^n ROT_{R_{i-1}}^{R_i}$$

L'expression des angles d'Euler calculés à partir de la matrice d'orientation ROT (3x3) est la suivante:

$$\psi = \text{signe}(\text{Pos13}) \cdot \text{ArcCos} \left(-\text{Pos23} / \sqrt{1 - (\text{Pos33})^2} \right)^{-}$$

$$\theta = \text{ArcCos}(\text{Pos33})$$

$$\phi = \text{signe}(\text{Pos31}) \cdot \text{ArcCos} \left(-\text{Pos32} / \sqrt{1 - (\text{Pos33})^2} \right)^{-}$$

Cette représentation pose le problème de la singularité pour certaines configurations. Par exemple pour $\text{Pos33} = \pm 1$, soit $\theta = k\pi$, les valeurs de ψ et ϕ ne sont pas calculables.

* Angles LTR ou YPR (Lacet=Yaw, Tangage=Pitch, Roulis=Roll)

Inspirés des angles d'Euler, ces trois angles correspondent respectivement aux rotations par rapport aux trois axes d'un repère cartésien (X Y Z).

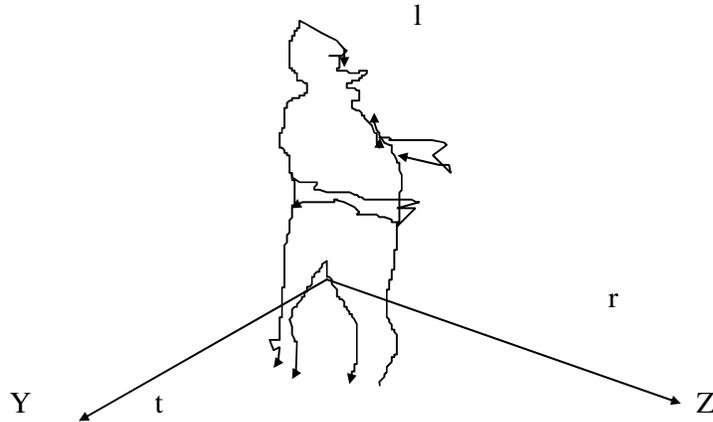
Souvent ces angles sont notés comme dans le formalisme d'Euler: Ψ , θ et ϕ , avec :

$\psi \equiv l = \text{rot}(X)$ appelé angle de lacet.

$\theta \equiv t = \text{rot}(Y)$ appelé angle de tangage ou pliage.

$\phi \equiv r = \text{rot}(Z)$ appelé angle de roulis ou pivot.

X



-Figure 2-4- Angles l, t, r : lacet, tangage, roulis

-c-Rappel sur les Transformations Homogènes

Toute matrice donnant la position et l'orientation d'un point P de l'espace par rapport à un repère R donné, peut être exprimée sous forme de matrice (4x4) comprenant : une matrice (3x3) d'orientation $ROT(\vec{n}, \vec{s}, \vec{a})$, un vecteur colonne de position L (Px,Py,Pz), un vecteur ligne de perspective f (1x3), et enfin un facteur d'échelle (1x1). Ces derniers paramètres (coefficient de perspective et facteur d'échelle) utilisés uniquement en graphique, sont des coefficients de correction pour tenir compte des déformations dues à la vision en perspective et à la distance focale. Ils sont pour nous toujours constants: perspective nulle et facteur d'échelle égal à un(cf figure 2-5).

Ainsi une rotation pure d'angle θ ou une translation pure de vecteur $\vec{V}(x,y,z)$, exprimées sous forme de transformations homogènes, seront respectivement données par les matrices 5-b et 5-c de la figure 2-5.

$$\begin{bmatrix} nx & sx & ax & Px \\ ny & sy & ay & Py \\ nz & sz & az & Pz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

-5-a-Matrice générale

$$\begin{bmatrix} nx & sx & ax & 0 \\ ny & sy & ay & 0 \\ nz & sz & az & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

-5-b-Rotation pure

$$\begin{bmatrix} 1 & 0 & 0 & x \\ 0 & 1 & 0 & y \\ 0 & 0 & 1 & z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

-5-c-Translation pure

Figure 2-5-Matrices de transformations homogènes

-d-Modélisation libre

Dans cette méthode, nous commençons par associer à chaque segment du porteur un repère dont l'origine et l'orientation seront choisies arbitrairement.

Ensuite pour passer d'un repère R_{i-1} à un repère R_i , nous procéderons par une succession de transformations géométriques élémentaires (rotation ou translation pure). A chacune de ces transformations nous associerons une matrice de transformation homogène.

Pour obtenir la matrice T_{i-1}^i de passage du repère R_{i-1} au repère R_i (notée A_i), on pose les matrices précédentes dans le même ordre que celui dans lequel ont été effectuées les transformations (**prémultiplication des transformations géométriques**). Puis on effectue le produit des matrices en les regroupant deux à deux et en commençant par la droite (**postmultiplication des matrices homogènes**).

La matrice T du porteur est enfin obtenue par multiplication des différentes matrices A_i , en respectant les mêmes règles que précédemment concernant la prémultiplication et la postmultiplication.

Pour la modélisation de l'effecteur³, on procède de la même manière que pour le porteur. On obtient une matrice E donnant la situation de l'effecteur (en général de son extrémité) par rapport au repère R_n du segment le supportant.

Remarque:

³: *Bien que dans le cas le plus simple l'effecteur soit constitué d'une simple pince à deux doigts, il n'est pas rare qu'il constitue à lui tout seul une structure mécanique articulée (avec ses propres degrés de liberté) supportant un outil.*

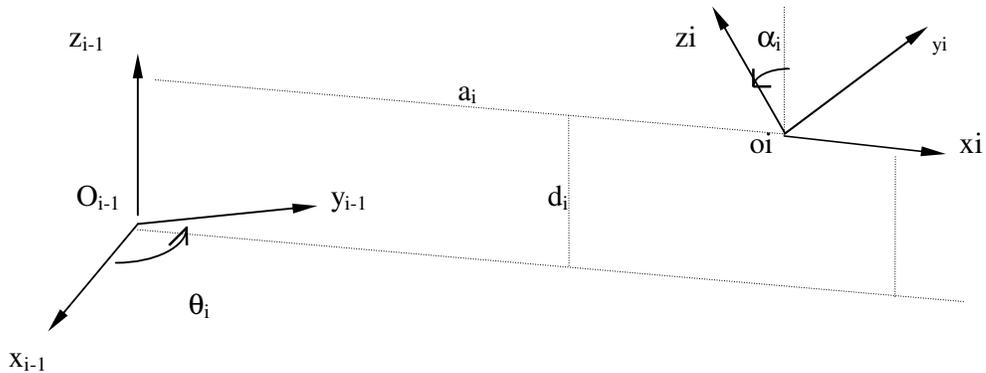
-e-Formalisme de Denavit-Hartenberg

*** Principe général**

L'articulation numéro i lie les segments S_{i-1} et S_i . Les segments sont numérotés de 0 à n , et les articulations de 1 à n . A chaque segment S_i sont associés l'articulation numéro i , les paramètres $(\theta_i, a_i, d_i, \alpha_i)$ ainsi que le repère $R_i (O_i, X_i, Y_i, Z_i)$.

Le passage du repère R_{i-1} au repère R_i se fait par le biais de quatre transformations géométriques successives (dans un ordre strict), et la matrice de passage est obtenue en effectuant le produit des quatre matrices de transformations homogènes correspondantes.

$T_{i-1}^i \equiv A_i \equiv \text{Rot}(Z_{i-1}, \theta_i) * \text{Trans}(Z_{i-1}, d_i) * \text{Trans}(X_i, a_i) * \text{Rot}(X_i, \alpha_i)$
aligne fait coïncider fait coïncider fait coïncider
X_{i-1} et X_i X_{i-1} et X_i O_{i-1} et O_i les repères



- Figure 2-6 - Changement de repère dans "Denavit-Hartenberg"

d_i : Distance entre O_{i-1} et le point de l'axe Z_{i-1} intersection des axes Z_{i-1} et X_i

a_i : Distance entre les deux axes de rotation numéro i et $i+1$ (c'est la normale commune aux axes Z_{i-1} et Z_i)

θ_i : $\theta_i = \angle(X_{i-1}, X_i)$

α_i : angle formé par les deux axes de rotation numéro i et $i+1$

$\alpha_i = \angle(Z_{i-1}, Z_i)$

La matrice de changement de repère a pour expression :

$\begin{bmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & \cos \theta_i \sin \alpha_i & a_i \sin \theta_i \\ 0 & 0 & \sin \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{bmatrix}$	mande -
---	---------

$$T_{i-1}^i \equiv A_i = \begin{array}{cccc} \sin \theta_i & \cos \theta_i \cos \alpha_i & -\sin \alpha_i \cos \theta_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{array}$$

Bien que cette méthode ait été définie à l'origine pour des articulations rotoïdes, elle peut s'appliquer pour des articulations prismatiques.

Si l'articulation est rotoïde, les paramètres a_i , d_i et α_i sont constants. La variable articulaire est θ_i ; par contre si l'articulation est prismatique, la variable articulaire est d_i et les paramètres a_i , θ_i et α_i sont constants. De plus $a_i = 0$ car la distance entre axes n'intervient plus.

Remarque:

Parfois le dernier segment (S_n) est lié au segment S_{n-1} par une articulation multiple (trois rotations pour un poignet). Mais **il n'y a pas d'articulations à l'extrémité du dernier segment S_n** . S'il en existe, elles ne font pas partie des variables articulaires du porteur. Elles font partie des paramètres propres à l'effecteur et n'interviennent que dans l'expression de sa matrice E.

* Règles de choix des emplacements des repères

Dans le cas d'articulations rotoïdes, des règles ont été définies pour le choix de l'emplacement du repère R_i et de son origine O_i (cf exemple en annexe). Il est recommandé de les suivre :

Origine O_i : placée à l'intersection de l'axe de rotation numéro $i + 1$, et de la normale commune aux axes de rotation numéro i et $i + 1$.

Lorsque les deux axes se rencontrent, O_i est placée à leur intersection. Dans le cas d'axes parallèles, l'emplacement de O_i est choisi de façon que la distance d_i soit nulle.

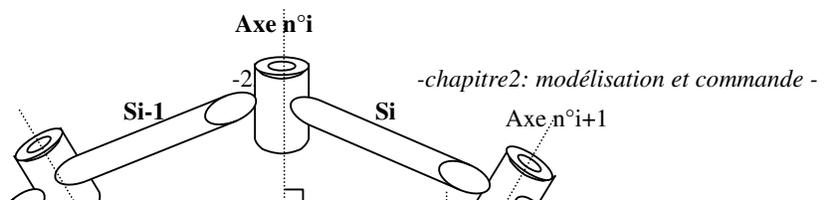
Axe Z_i : aligné avec l'axe de rotation $i + 1$.

Axe X_i : aligné avec la normale commune aux axes de rotation $n^o i$ et $i + 1$, son sens va de l'axe $n^o i$ vers $i + 1$ (dans le cas d'axes de rotation parallèles). Dans le cas d'axes qui se coupent, \bar{x}_i est choisi parallèle ou antiparallèle au vecteur $\bar{z}_{i-1} \wedge \bar{z}_i$.

Axe Y_i : $\bar{y}_i = +(\bar{z}_i \wedge \bar{x}_i)$.

Repère R_o : son axe z_o doit être aligné avec l'axe de rotation numéro 1. Son origine o_o peut être choisie n'importe où dans la base du porteur, de préférence confondue avec o_1 .

Repère R_n : son origine est de préférence confondue avec celle de R_{n-1} .



Axe n°i-1



-Figure 2-7- Choix des repères dans "Denavit-Hartenberg"

-f-Conclusion

Quelle que soit la méthode utilisée pour le choix des repères, dans la modélisation géométrique utilisant les transformations homogènes, la structure mécanique articulée sera modélisée par une matrice T et l'effecteur par une matrice E.

Soit S0 la base du SMA et Sn le dernier segment. T est le produit des matrices successives Ai, donnant chacune la position et l'orientation du repère Ri attaché à un segment Si par rapport au précédent.

$$T = \prod_{i=1}^{i=n} A_i$$

La matrice T donne la situation du dernier segment (support de l'effecteur) par rapport à R0.

La situation de l'extrémité de l'effecteur par rapport au repère de référence (Runivers) est donnée par la matrice de transformation homogène:

$$POS = Z * T * E$$

-III-COMMANDE EN POSITION

-1- Problème géométrique direct

Etant donné le vecteur de variables articulaires pour une posture donnée du SMA, (donc les valeurs des variables moteurs correspondant aux différentes articulations), le problème direct consiste à déterminer la situation (position et orientation) de l'extrémité P de

l'effecteur par rapport à un repère de référence. Sa résolution revient donc à déterminer la matrice de transformation homogène correspondant au produit Z^*T^*E , Z et E étant connues.

Connaissant d'une part les valeurs des variables articulaires θ_i , et connaissant d'autre part l'expression de T en fonction des θ_i , on en déduit donc la valeur de T. Il ne reste plus qu'à conclure la valeur du produit $POS = Z^*T^*E$. De cette matrice (4x4) on extrait directement le vecteur position $L(P_x, P_y, P_z)$. De la matrice d'orientation ROT (3x3) on extrait les paramètres d'orientation (l,t,r), si on a utilisé par exemple les paramètres lacet, tangage et roulis.

Ainsi on connaît donc le vecteur $Pos(x,y,z,l,t,r)$ donnant la situation du point atteint par l'extrémité de l'effecteur.

-2- Problème géométrique inverse

Connaissant les coordonnées du point à atteindre (x,y,z,l,t,r), et donc par voie de conséquence la matrice POS (4x4) indiquant la situation que l'on veut faire atteindre à l'effecteur, il s'agit de déterminer le vecteur de variables articulaires nécessaire pour obtenir cette configuration géométrique.

$$Z * T * E = POS \implies T = Z^{-1} * POS * E^{-1}$$

Les trois matrices Z, POS et E étant connues, il est aisé d'en déduire T. Connaissant la matrice T et les expressions des matrices de passage intermédiaires entre les repères attachés aux différents segments, on peut alors calculer analytiquement le vecteur de variables articulaires $\vec{\theta}(\theta_1, \theta_2, \dots, \theta_m)^4$.

Remarque:

⁴: Ce nombre de variables articulaires est en général supérieur au nombre de degrés de liberté, qui lui est égal au nombre de paramètres indépendants. En effet certaines articulations sont redondantes (par conséquent dépendantes) et correspondent à un même degré de liberté.

-3- Commande en position

Le premier problème que l'on se pose au niveau de la commande en position est celui du déplacement d'une position k à une position k+1. Pour cela il est plus aisé de raisonner dans l'espace des variables articulaires. Aussi peut-on formuler le problème ainsi: le système mécanique articulé se trouvant dans une configuration quelconque (ce qui signifie que les variables articulaires θ_i ont des valeurs quelconques mais mécaniquement possibles),

on désire amener l'organe terminal (c'est à dire le repère qui lui est associé) en un point de l'espace avec une certaine orientation vis à vis d'un repère fixe R_{un} lié à l'environnement. La situation par rapport à R_{un} d'un repère lié à ce point est donnée par la matrice POS_{k+1} .

La réponse à ce problème suppose que:

* chaque variable articulaire θ_i soit connue par rapport à une valeur origine θ_{i_0} . L'ensemble des θ_{i_0} va constituer la configuration de référence pour les calculs, laquelle sera représentée par le vecteur $\vec{\theta}_o(\theta_{1_0}, \theta_{2_0}, \dots, \theta_{n_0})$, n étant le nombre de degrés de liberté⁵ ;

* l'on soit à même de connaître les $\theta_{i_{k+1}}$ désirées à partir de la connaissance de la matrice POS_{k+1} liée à l'organe terminal: c'est l'objet de la résolution du problème géométrique inverse;

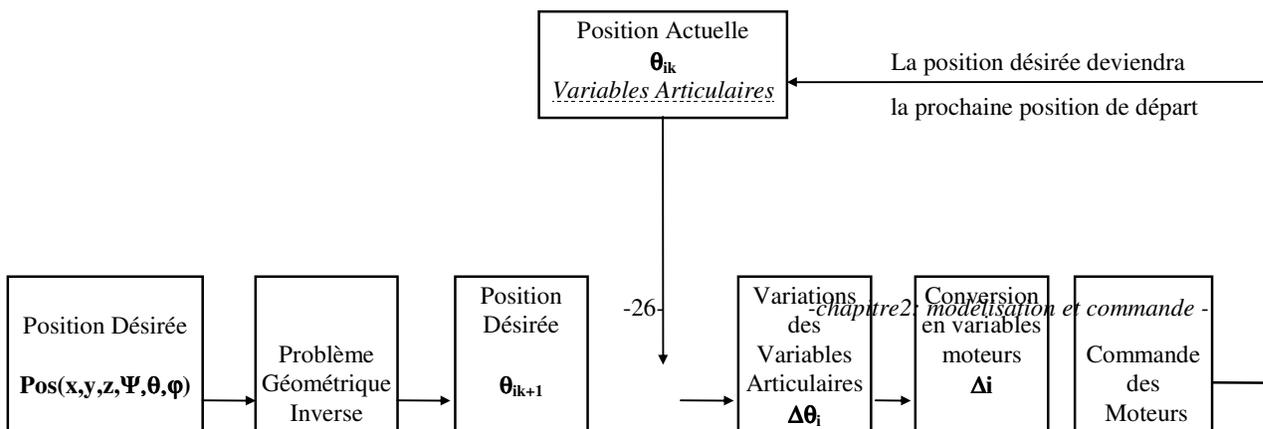
* l'on sache amener chaque θ_{i_k} à la valeur désirée $\theta_{i_{k+1}}$, ce qui met en jeu le système de commande. C'est à dire celui qui à partir de toutes les informations disponibles (entre autres les écarts $\Delta\theta_i$ et leur conversion en variables moteurs), va générer les signaux de puissance des actionneurs⁶, et faire atteindre ainsi aux θ_{i_k} les valeurs désirées $\theta_{i_{k+1}}$, et ce **selon un ordre prédéfini dans l'algorithme de commande** qui détermine la trajectoire de l'effecteur.

On peut schématiser globalement ce fonctionnement comme indiqué sur la figure 2-8 .

Remarques

⁵ : On supposera pour simplifier que le nombre de degrés de liberté m est égal au nombre de variables articulaires n .

⁶ : Nous passerons volontairement sous silence les couples ou forces qui agissent sur la structure mécanique, qui sortent du cadre de ce cours car ils interviennent uniquement dans le modèle dynamique.





-Figure 2-8- Organigramme d'une commande en position

-4- Résolvabilité

Dans la résolution du problème géométrique inverse, il s'agissait de déterminer analytiquement le vecteur de variables articulaires θ_i , nécessaire à la commande pour atteindre la position et l'orientation désirées de l'effecteur. Dans le cas où ce vecteur est unique on dit que le SMA est résolvable. Mais diverses contraintes peuvent apparaître dans le calcul des θ_i .

-a-Absence de solution

* pour raison géométrique: le point est non atteignable car il se trouve hors du domaine atteignable du SMA (point trop éloigné);

* pour raison mécanique: pour des raisons de butée articulaire, il se peut qu'une valeur de θ_i soit théoriquement possible mais mécaniquement impossible à atteindre, en raison de la limitation par butée lors de la construction du SMA. Par conséquent la valeur théorique trouvée pour θ_i ne fait pas partie de son domaine de définition.

* pour raison mathématique: la résolution analytique sur des équations non linéaires peut s'avérer impossible (déterminant d'une matrice (Jacobien) nul donc son inversion est impossible), ou bien on peut arriver à des situations telles que $\cos \theta_i > 1$?! Dans ces cas, on décale légèrement par rapport au point à atteindre et généralement l'impossibilité disparaît d'elle même.

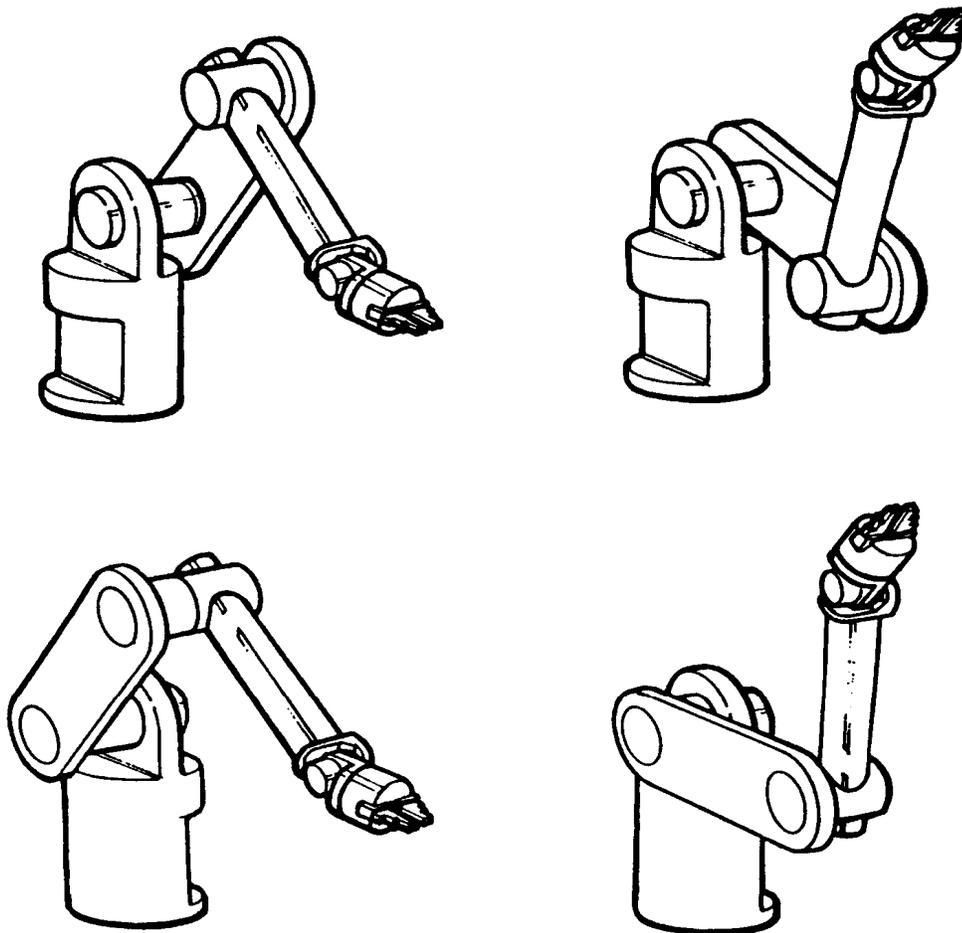
-b-Infinité de solutions

Si le nombre de contraintes k (en particulier les vecteurs position et orientation) est inférieur au nombre n de degrés de liberté, on aura donc $(n-k)$ variables articulaires arbitraires et compatibles avec la géométrie du SMA et du point imposé.

-c-Solutions en nombre fini mais supérieur à un

Dans ce cas particulier, il est généralement aisé d'imposer des restrictions ou contraintes supplémentaires (telles que coude vers le haut ou vers le bas , bras droitier ou gaucher) pour arriver à une solution unique et rendre ainsi le SMA résolvable.

Mis à part les cas a, b, c, dans tous les autres cas on dit que le manipulateur est résolvable.



Différentes configurations du robot AID V5(coude au-dessus ou en dessous du poignet, bras à gauche ou à droite)

Figure 2.9: exemples de structures permettant de réduire le nombre de solutions

-IV- AUTRES TYPES DE MODELISATION

Dans la modélisation géométrique, nous n'avons fait intervenir que les longueurs des segments et les variables articulaires. Ce type de modélisation convient à la commande en position ou point à point, où la vitesse de déplacement est fixée uniquement par les temps de réponse des dispositifs intervenant dans la structure et la commande du SMA.

Dans les commandes cinématique et dynamique où l'on veut contrôler la vitesse lors des déplacements, et même avoir des accélérations importantes, le modèle géométrique n'est plus valable pour la commande. On est obligé d'avoir recours aux modèles cinématique et dynamique, qui font intervenir les masses, les forces d'inertie etc... Ceci sort du cadre de ce cours qui n'est qu'une simple introduction à la robotique.

-V- EXEMPLES DE MODELISATION (cf annexes)

Le premier exemple donné en annexe 1 concerne le manipulateur "ROBY 1" dont toutes les articulations sont rotoïdes. En plus des articulations que l'on peut directement commander, ce manipulateur présente une articulation qui n'est pas directement contrôlée, mais qui est commandée par un couplage mécanique, permettant de maintenir la pince horizontale quelle que soit la position du manipulateur.

La modélisation est purement géométrique. Nous y avons utilisé délibérément la méthode de Denavit-Hartenberg, pour illustrer cette méthode qu'il est difficile d'expliquer. Il aurait été certes plus facile d'utiliser la méthode dite libre, le principe étant que l'on arrive dans tous les cas à une matrice de transformation homogène.

Cet exemple illustre bien le cas d'un manipulateur que l'on rend résoluble par imposition de contraintes supplémentaires. La technique utilisée dans notre cas consiste à lever l'ambiguïté sur les valeurs doubles des angles obtenus par la fonction arctangente, en remplaçant la fonction standard par une fonction arctangente à deux arguments. Si cela est insuffisant, on impose alors une contrainte en modifiant le domaine de définition des variables (plage de variation de l'angle).

L'exemple donné en annexe 2 concerne le robot IR52C, dont toutes les articulations sont également rotoïdes. On a utilisé la technique de modélisation « libre ». La résolution des équations pour déterminer les angles se fait analytiquement. Le SMA est rendu résoluble en utilisant les notions de coude au dessus ou au dessous du poignet.

- CHAPITRE III -
- PROGRAMMATION DES ROBOTS -

-I-GENERALITES ET OBJECTIFS DES SYSTEMES DE PROGRAMMATION

Un manipulateur asservi constitue la partie opérative d'un automatisme particulier. C'est par l'intermédiaire d'une interface que l'utilisateur peut avoir accès aux ressources mécaniques et aux algorithmes de contrôle (générateur de trajectoires, asservissements) prévus dans la conception d'un robot. La nature même de cette interface est primordiale, car les manipulateurs asservis ne constituent qu'une petite partie d'un processus automatisé. Le terme de cellule de travail est utilisé pour désigner un ensemble d'équipements automatisés, incluant un ou plusieurs robots; la programmation d'un robot doit donc être considérée à l'intérieur d'un problème plus large de programmation d'un ensemble d'équipements interconnectés.

Dans ce sens, l'interface d'un système de programmation de robot doit posséder un mode de développement de programmes. Celui-ci doit permettre à l'utilisateur de décrire la tâche attribuée au manipulateur, ainsi que les conditions de synchronisation et de coordination avec d'autres automatismes susceptibles d'être présents dans la cellule de travail. Les aides au développement de programmes comportent essentiellement une fonction d'édition, ainsi qu'une fonction d'aide à la mise au point, autorisant par exemple l'exécution de spécifications partielles (parties de la tâche) par le manipulateur. Enfin l'interface doit permettre à l'utilisateur de commander l'exécution d'un ou plusieurs programmes mémorisés. Les objectifs liés à la conception d'un système de programmation sont de faciliter la mise en oeuvre d'un manipulateur asservi pour la réalisation d'une nouvelle tâche. On souhaite donc qu'un tel système soit simple à utiliser et qu'il permette de décrire les tâches les plus complexes, sans limiter les possibilités d'expression des utilisateurs. Ces objectifs sont cependant difficiles à satisfaire simultanément, c'est la raison pour laquelle les systèmes actuels sont encore très diversifiés, conçus généralement pour un manipulateur particulier et/ou une catégorie particulière de tâches.

-II-METHODES DE PROGRAMMATION

Il existe principalement deux méthodes de communication avec le robot pour lui spécifier une tâche, selon que l'on se sert de son système mécanique articulé pour lui apprendre la tâche (programmation par l'exemple), ou qu'on la lui décrit dans un langage formel (programmation textuelle).

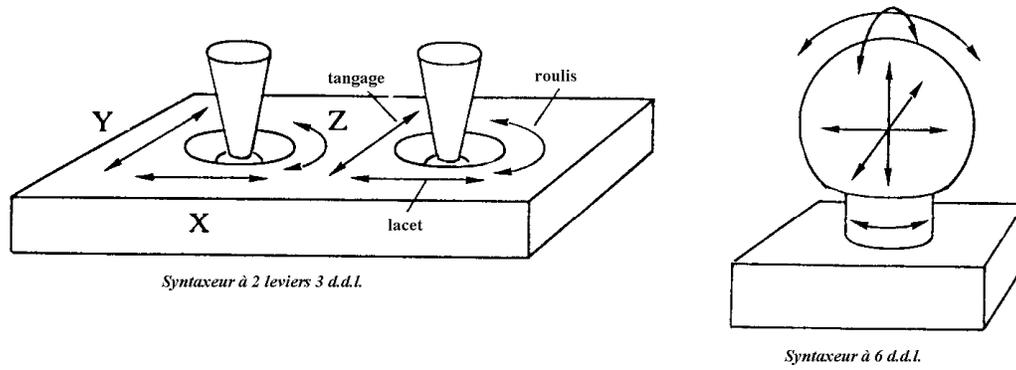
-1- Guidage ou programmation par l'exemple

En se servant d'un dispositif d'interaction tel qu'un "joystick" ou une "boîte à boutons" ou un "syntaxeur", le programmeur "télécommande" le manipulateur pour réaliser un échantillon d'exécution de la tâche. Les valeurs des variables articulaires sont enregistrées, et le manipulateur pourra ensuite exécuter la tâche de manière autonome.

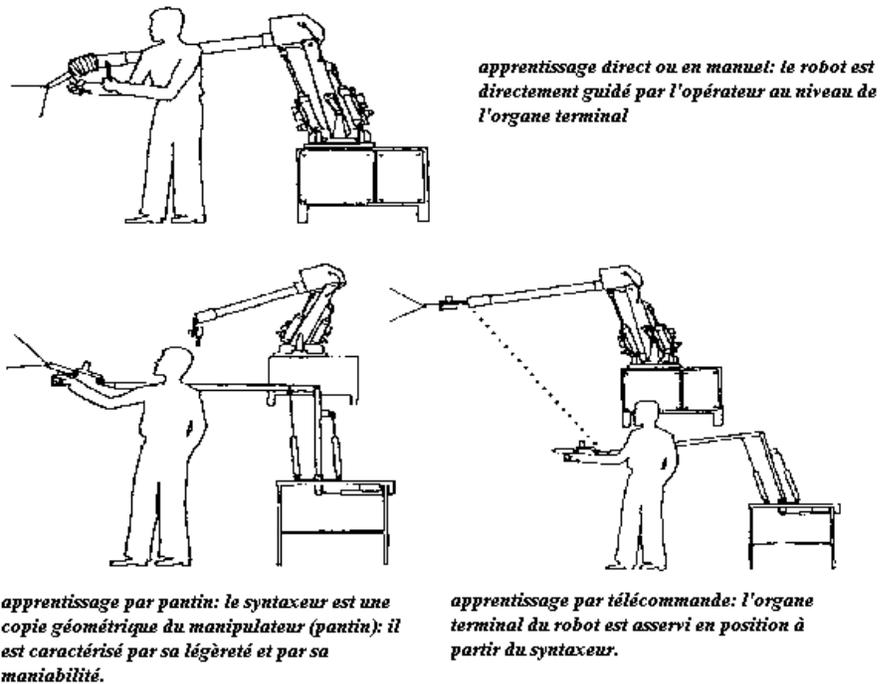
Si cette méthode présente l'avantage de la facilité de programmation (elle ne nécessite aucune connaissance spécialisée de l'utilisateur), elle présente aussi des inconvénients qui réduisent son application aux tâches simples et répétitives telles que la peinture par projection, le soudage par points ou la manutention d'objets simples. Parmi ses inconvénients bien connus, on peut citer :

- la nécessité d'avoir réalisé complètement le poste de travail, d'où l'impossibilité de faire de la programmation hors-ligne;
- la répétition des mêmes séquences de mouvements avec des changements mineurs à chaque itération (palettisation par exemple) est fastidieuse à programmer;
- les programmes enregistrés sont difficiles à modifier;
- il n'y a pas de signaux de capteurs pour traiter les incertitudes de position, ce qui nécessite la réalisation de robots et équipements périrobotiques de très bonne répétabilité;
- la coordination de plusieurs robots est impossible.

Les différences entre les divers systèmes de programmation de robots par guidage résident dans la façon dont les emplacements sont spécifiés et enregistrés (point à point, trajectoire continue ... etc.), ainsi que dans le répertoire des mouvements possibles entre les emplacements au moment de la restitution des trajectoires (mouvement indépendant de chaque articulation, mouvement coordonné des articulations pour réaliser une ligne droite par rapport à différents repères, ...etc.).



Syntaxeurs pour le guidage manuel



Modes de programmation en continu (doc AKR)

Figure 3-1: exemples de systèmes pour la programmation par guidage

Si le guidage fut longtemps la seule méthode de programmation des robots et continue aujourd'hui d'être largement utilisé, on le retrouve sous une forme ou sous une autre dans la majorité des systèmes de programmation des robots actuels, mais avec une autre finalité. Dans ces systèmes le guidage ne représente plus une méthode de programmation, mais un outil d'aide à la programmation, car on s'en sert uniquement dans la phase de modélisation et de déclaration de l'univers, pour enregistrer des paramètres correspondant à différents emplacements. La trajectoire du manipulateur est ensuite calculée par interpolation entre ces différents points, en fonction de la trajectoire voulue par le programmeur et explicitement exprimée dans le langage utilisé.

-2- Programmation symbolique ou textuelle

Dans ce type de programmation, l'utilisateur se sert d'un langage symbolique pour décrire la tâche au robot, cette description pouvant être explicite ou implicite. Dans la programmation explicite, l'utilisateur exprime dans un programme toutes les séquences de mouvement du manipulateur, en termes de positions et vitesses nécessaires pour réaliser la tâche. Dans la programmation implicite, il décrit sa tâche sous forme de primitives orientées-tâche sans exprimer les différentes opérations de mouvement. On distingue donc trois modes de programmation textuelle selon que la description est implicite, mixte ou explicite. Ces trois modes correspondent respectivement à trois niveaux d'abstraction distincts pour les concepts manipulés par le programmeur: le niveau objectif, le niveau objet et le niveau effecteur. L'exemple simple de la figure 3-2 nous permettra de préciser ces différents niveaux, et de présenter très simplement les mécanismes devant exister à chaque niveau.

Auparavant, il est utile de préciser que quel que soit le niveau de description utilisé, la programmation d'une tâche comprendra toujours une partie déclaration du modèle de l'univers (qui peut se faire soit de façon textuelle, soit par guidage, ou enfin à l'aide d'un dispositif graphique ou une station de CAO), et une partie spécification de la tâche proprement dite, qui devra engendrer des primitives de mouvement de niveau actionneur pour un manipulateur particulier. L'exemple ne traite que la partie spécification et d'une manière très schématique.

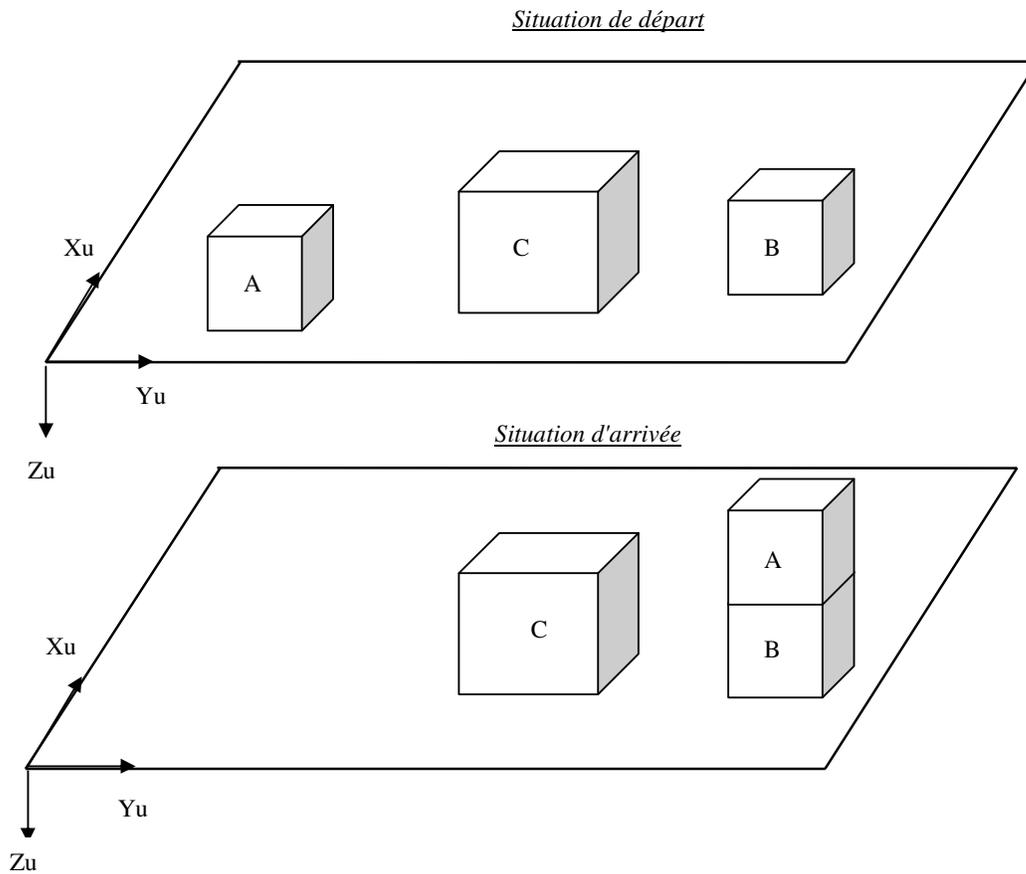


Figure 3.2: exemple de tâche de manipulation

(Il s'agit de déposer le cube A sur le cube B, avec face grise de A tournée vers l'axe Y du repère de référence)

2-a-Niveau effecteur

Dans une description de niveau effecteur, **l'utilisateur raisonne en termes de positions et orientations de l'organe terminal** du manipulateur. Il sera obligé de décomposer complètement sa tâche et d'en spécifier toutes les étapes au manipulateur, sous forme d'instructions de déplacement et d'action de l'effecteur. Il précisera donc au manipulateur d'amener son effecteur au dessus de A, d'ouvrir sa pince, de descendre sur A, de fermer sa pince, d'aller à une position intermédiaire (pour éviter C), d'amener sa pince à une position au dessus de B, d'orienter sa pince dans une direction donnée, de descendre sa pince (A sur B), d'ouvrir sa pince et enfin de se retirer. **Le programmeur a donc en charge la planification complète de la tâche.** L'unité de contrôle du robot n'aura qu'à traduire les instructions pour interpoler la trajectoire entre deux points, l'échantillonner puis générer les tensions de commande nécessaires aux asservissements pour la réalisation des déplacements.

2-b-Niveau objet

Même si on continue à spécifier certains mouvements du manipulateur, dans une description de niveau objet (qui est intermédiaire entre les descriptions explicite et implicite) on s'élève un peu dans le niveau d'abstraction, et **le programmeur raisonnera en termes d'objets** et non plus en termes d'emplacements de l'effecteur. **L'utilisateur n'est responsable qu'en partie de la planification de la tâche**, qui sera **décrite en termes de situations et de relations spatiales à réaliser sur les objets** réels. Il ordonnera par exemple au manipulateur de prendre l'objet A, d'aller à une position intermédiaire (pour éviter C), d'orienter la face 1 de A vers l'axe Y, et enfin de déposer A sur B. C'est le système de contrôle qui est chargé d'interpréter ces spécifications, d'extraire du modèle préenregistré de l'univers les positions des objets ainsi que la position de prise de A, d'envoyer ensuite les commandes de déplacement au manipulateur.

2-c-Niveau objectif

Dans un système de niveau objectif, **le programmeur est totalement déchargé de la planification de la tâche**. Il ne **raisonne plus en termes d'objets** mais **d'objectifs**, et il n'aura qu'à spécifier le but final de la tâche. Il écrira par exemple dans son programme : DEPOSER CUBE A SUR CUBE B AVEC FACE 1 DE CUBE A ORIENTEE AXE Y DE UNIVERS, sans se préoccuper des différentes phases de planification qui sont entièrement à la charge du système. Ce dernier devra alors être en possession d'une certaine "intelligence" pour traduire ces spécifications de niveau tâche en un ensemble de commandes pour les actionneurs.

Remarque : *Il existe en réalité un quatrième niveau qui est le niveau "actionneur". Le programmeur raisonne alors en termes de valeurs à faire atteindre aux variables articulaires, associées à chaque actionneur, pour faire atteindre au manipulateur une position donnée.*

-III-CARACTERISTIQUES DES DIFFERENTS LANGAGES

DE PROGRAMMATION

La programmation textuelle a été développée au cours des années 1970, par les laboratoires de recherche en intelligence artificielle. Les micro-ordinateurs actuels, puissants et peu coûteux, permettent de réaliser des contrôleurs de robots dont les tâches sont décrites à l'aide de programmes écrits dans des langages formels. Ces langages possèdent par rapport aux langages classiques comme Fortran, Basic, Pascal, des types de données et des opérations ou fonctions standards propres à la robotique. C'est pourquoi ils sont appelés langages de programmation des robots.

-1- Niveau actionneur ou articulaire

Dans ce niveau, l'utilisateur ne peut programmer que des déplacements des actionneurs de façon coordonnée ou non. Les seules variables disponibles sont le plus souvent les variables articulaires ou les efforts à fournir pour les actionneurs. Il n'y a pas à proprement parler de langage.

Comme le programmeur commande son manipulateur en donnant des "consignes" correspondant aux valeurs désirées sur les actionneurs, on ne peut concevoir la programmation d'une tâche industrielle à ce niveau sans faire appel au guidage. Ce niveau possède donc les avantages et les inconvénients déjà énoncés pour la programmation par guidage.

-2- Niveau effecteur

Dans ce type de programmation, on fait une description de la tâche en la considérant comme une suite de déplacements et d'actions de l'effecteur. La tâche est divisée en un certain nombre d'étapes, chacune étant caractérisée par une configuration de départ et une configuration d'arrivée de l'effecteur. On spécifiera donc au robot une suite de positions et d'orientations de l'effecteur.

Selon le degré de structuration du langage et de complexité des structures de données, on distingue les langages de type "primitives de mouvement" et les langages "structurés".

Dans les langages de type primitives de mouvement, on distingue généralement deux ou trois primitives de base (MOVE, DEPARTS, APPRO), une primitive de changement de vitesse (SPEED), l'appel de sous programme sans passage de paramètres (GOSUB), et comme données uniquement les données d'emplacement, qu'elles soient exprimées sous forme de point (vecteur de variables articulaires), ou sous forme de transformation géométrique (vecteur donnant les

paramètres de position et d'orientation). Ces langages sont peu structurés car dérivés souvent de langages classiques tels Basic, Fortran, Pl/1.

Les langages structurés sont fortement inspirés des langages procéduraux classiques tels Pascal, Algol ou C. Dans cette catégorie, la spécification d'une tâche comporte une importante phase de déclaration dans laquelle le programmeur doit préciser, par types, les variables qui seront utilisées, il peut également créer des modèles des objets de l'environnement. Par rapport aux types de données conventionnels et aux langages précédents, ces langages introduisent le type repère, le type vecteur et le type transformation, ainsi que des fonctions standards et des opérations sur ces types de données (composition de transformations, produit scalaire ou vectoriel, ...etc.). La représentation d'un objet solide est créée par le programmeur à l'aide de variables de type repère et de type transformation.

Le niveau effecteur intègre donc certains aspects qui le rendent plus général que le niveau précédent:

- instructions de branchements et sous programmes avec parfois passage de paramètres (langages structurés);
- beaucoup de possibilités d'interaction avec des capteurs, même si le plus souvent c'est par le biais de lignes « tout ou rien »;
- spécification des destinations en mode cartésien (par opposition à mode articulaire), et parfois mouvements relatifs ou en ligne droite;
- parfois primitives d'exécution en parallèle;
- l'utilisation de transformations de coordonnées et de repères rend possible théoriquement la programmation hors-ligne, car les différences entre le modèle déclaré de l'univers et son état réel peut être exprimée en termes de transformations.

Même si le niveau effecteur résout certains problèmes du niveau précédent, le raisonnement sur les mouvements du manipulateur plutôt que sur les objets à manipuler (ce qui nécessite une bonne maîtrise du manipulateur et de son fonctionnement), rend ce type de programmation presque impossible pour un non "initié". Même si les structures de contrôle assez riches des langages structurés les rendent aptes à activer des actions sur occurrences d'événements, l'inexistence de "logiciel de secours" (arrêt d'urgence ou aiguillage en cas d'erreur), ainsi que le peu de possibilités d'extensions à la programmation hors-ligne sont autant de désavantages de ce type de programmation.

-3- Niveau objet ou géométrique

Un langage de programmation de niveau objet est une extension d'un langage structuré de programmation de niveau effecteur. On peut le considérer comme une interface entre l'utilisateur, un langage explicite de niveau effecteur, et une base de modèles géométriques. L'intérêt d'une telle approche réside dans le fait qu'elle facilite le raisonnement géométrique dans l'espace tridimensionnel. Les systèmes de programmation correspondants utilisent un mode d'expression qui peut être textuel, graphique ou mélangé.

Le programmeur raisonne en termes de "situations géométriques" décrites symboliquement par des relations géométriques du type "face A de l'objet 1 CONTRE face B de l'objet 2". Le programmeur fournit une description (modèles) des objets, sous forme de repères et d'entités géométriques (telles que plan, face, droite, cercle, trou ...etc.), puis il demande au manipulateur de réaliser des liaisons (relations géométriques telles que CONTRE, COPLANNAIRE, ALIGNÉ) entre ces objets. L'interpréteur de ces langages engendre généralement un programme de niveau effecteur, ou parfois directement un programme de niveau actionneur, dans lequel l'accessibilité des situations successives de l'effecteur (déduites des situations géométriques déclarées et du modèle géométrique du manipulateur) peut être testée. Du seul point de vue géométrique, la programmation hors-ligne est donc possible.

Ce type de programmation suppose l'existence de structures de contrôle et de données complexes qui ne sont possibles que grâce à l'utilisation de langages structurés (en majorité à base de Pascal, Algol et C). Ceci a permis l'introduction de variables d'état prédéfinies, permettant des changements d'outils ou correspondant à des configurations particulières du manipulateur (bras droitier ou gaucher, coude vers le haut ou vers le bas), ainsi que l'amélioration des possibilités de traitement parallèle et de capteurs, notamment par l'introduction quasi-généralisée de capteurs dans le poignet et parfois dans les doigts, ainsi que des primitives pour traiter ces informations.

Si ce type de programmation est un pas vers la spécification de niveau tâche, deux de ses atouts deviennent des faiblesses quand on se place au niveau de l'utilisateur non initié : les notions de transformation et de programmation structurée sont difficiles à comprendre, à assimiler et à utiliser par un programmeur non "expert".

-4- Niveau objectif ou tâche

L'utilisateur spécifie la tâche directement en termes d'opérations à effectuer sur les objets, sous la forme par exemple "insère goujon dans trou". Le programmeur déclare des buts (relations physiques entre les objets), plutôt que les mouvements du manipulateur nécessaires pour arriver à ces buts. Il y a une différence fondamentale par rapport au niveau géométrique, où l'utilisateur gère les liens entre objets, par la déclaration des transformations les symbolisant, alors qu'ils sont déduits de manière implicite par le système de niveau objectif.

La programmation de niveau objectif suppose l'intégration dans le système de modules chargés de prendre en compte notamment :

- les modèles géométriques complets du (ou des) manipulateur(s) et de l'environnement;
- la génération automatique de la position de prise des objets;
- l'évitement d'obstacles et la modification de trajectoire par l'utilisation d'informations en provenance de capteurs.

Tous ces problèmes sont loin d'être résolus globalement (d'où l'inexistence de système opérationnel) et nécessitent une large utilisation de l'intelligence artificielle.

-5- Conclusion

Cette classification ambitieuse et fortement hiérarchisée (la sortie d'un programme de niveau k est un programme de niveau k-1 inférieur) a constitué une hypothèse d'école vers la fin des années 1970. On pensait probablement que la robotique avancerait beaucoup plus vite qu'elle ne l'a fait effectivement, et que les infrastructures mécaniques, électroniques et informatiques indispensables pour la mise en oeuvre de systèmes robotisés complets, seraient disponibles à tous les niveaux.

La réalité des manipulateurs existants, l'absence d'une méthodologie robuste dans des problèmes clés (couplage perception-commande par exemple), l'inexistence de fait d'une démarche de programmation acceptée par tous pour programmer des applications, ont conduit à l'heure actuelle à considérer la programmation des robots à deux niveaux: le niveau manipulation qui inclut les deux premiers niveaux, et le niveau tâche les deux derniers.

Même cette dernière classification n'est pas complètement satisfaisante car, si conceptuellement certains langages sont de niveau manipulation, ils n'en possèdent pas moins certaines caractéristiques qui permettent des représentations de niveau tâche.

-IV- EXEMPLE DE TACHE PROGRAMMEE EN LANGAGE DE NIVEAU OBJET

Le programme de niveau effecteur engendré est en langage VAL, destiné à la commande du robot PUMA 560.

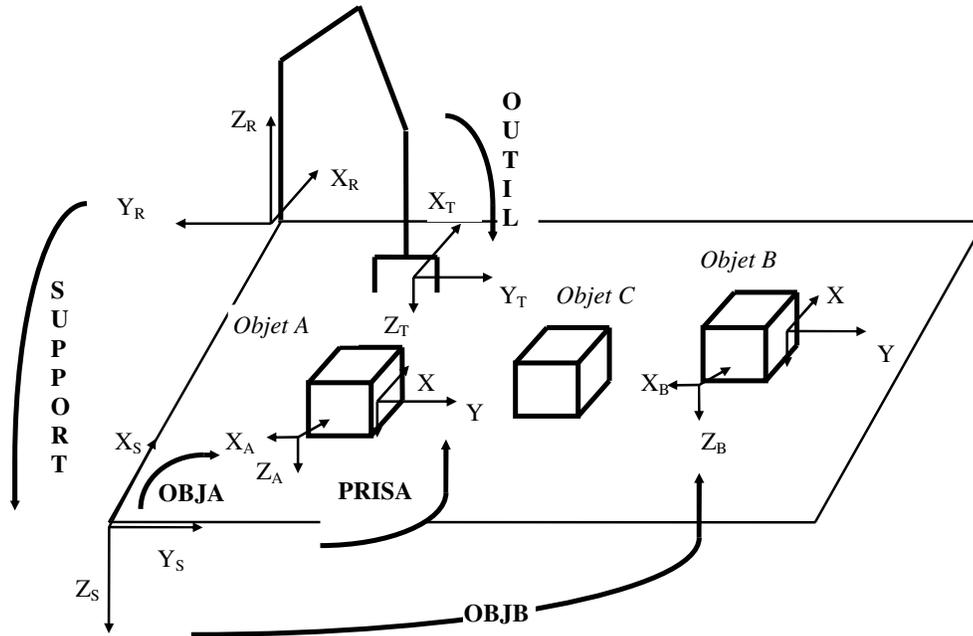


Figure 3-3-a : description de l'environnement avant l'exécution de la tâche

IV 1- Description de la tâche

La tâche sera décrite par un ensemble de quatre situations clés appelées situations géométriques. Chacune d'elles correspond à un état donné du manipulateur et de son environnement. Cela se traduit pour la commande du manipulateur par quatre états particuliers de sa pince, correspondant à quatre vecteurs (X,Y,Z,O,A,T). En effet dans les armoires de commande programmées en VAL et destinées à la commande des robots PUMA, chaque point de l'espace est décrit par une transformation qui est un vecteur à 6 composantes: les trois coordonnées cartésiennes (X,Y,Z) et trois angles (O,A,T).

a- Situation de départ

Elle correspond à celle décrite par le schéma précédent. Il est à noter trois transformations particulières que l'utilisateur déclare au départ et dont il ne se préoccupe plus après : SUPPORT qui donne l'emplacement du support par rapport au manipulateur , OUTIL qui décrit la pince par rapport à la bride de montage , et enfin PRISA qui est un repère fonctionnel propre à l'objet A et qui indique la position de prise de l'objet A par rapport à son propre repère.

b- Situation de prise de l'objet A

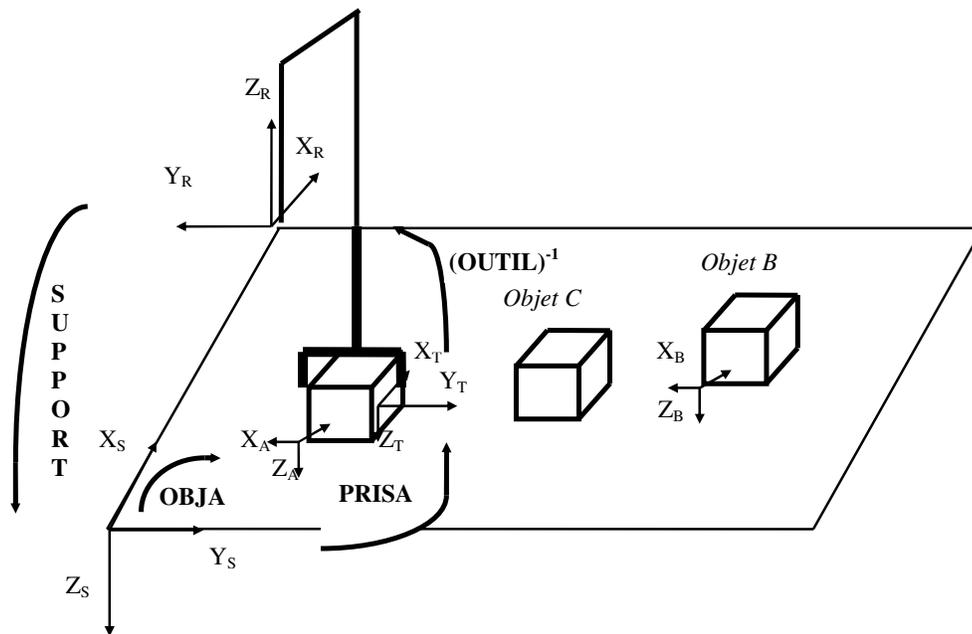


Figure 3-3-b: l'effecteur (pince) tient l'objet A

Dans la position de prise, le repère attaché à la pince devra coïncider avec le repère défini par la position de prise de l'objet A. Pour le manipulateur cela se traduit par le produit de transformations:

$$\text{SUPPORT: OBJA : PRISA : (OUTIL)}^{-1} .$$

En effet, la résolution du problème géométrique inverse qui consiste à déterminer le vecteur de variables articulaires nécessite que dans la position de prise , l'équation suivante soit vérifiée:

$$\text{T: OUTIL} = \text{SUPPORT : OBJA : PRISA} ,$$

où T désigne la transformation du manipulateur , c'est à dire celle qui donne la position et l'orientation de l'extrémité du dernier segment (base de l'outil) par rapport au repère de base du manipulateur (situé pour le PUMA 560 au niveau de son épaule).

c- Situation intermédiaire

Cette situation est introduite explicitement dans la planification de la tâche pour éviter la collision entre les objet A et B pendant le déplacement .

Dans un système réel de niveau tâche , cette situation intermédiaire n'est pas déclarée et c'est au "planner" d'essayer de trouver le bon chemin pour éviter cette collision .

Cette situation se caractérise par la transformation POSAB qui donne la position et l'orientation, du repère attaché à l'objet A par rapport à celui attaché à l'objet B.

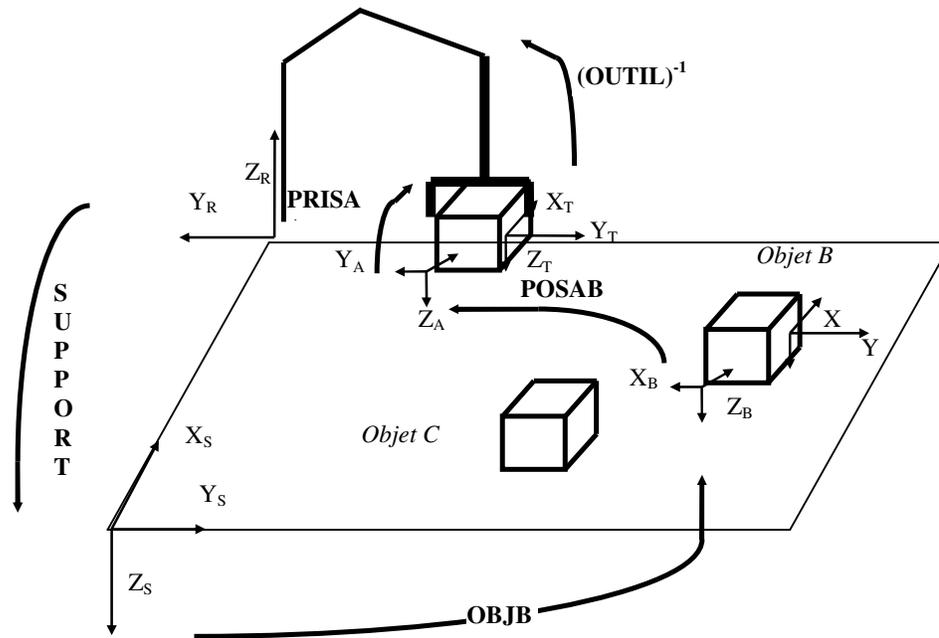


Figure 3-3-c : position d'évitement de collision (la pince tient toujours l'objet)

Au lieu de spécifier une transformation donnant directement la position de A par rapport au repère de référence, nous préférons passer par l'objet B pour mettre en relief le fait que cette transformation dépend de l'objet B.

Pour la commande de l'effecteur, cela se traduit par:

$$\text{SUPPORT: OBJB: POSAB: PRISA: (OUTIL)}^{-1}$$

d- Situation de dépôt de A sur B

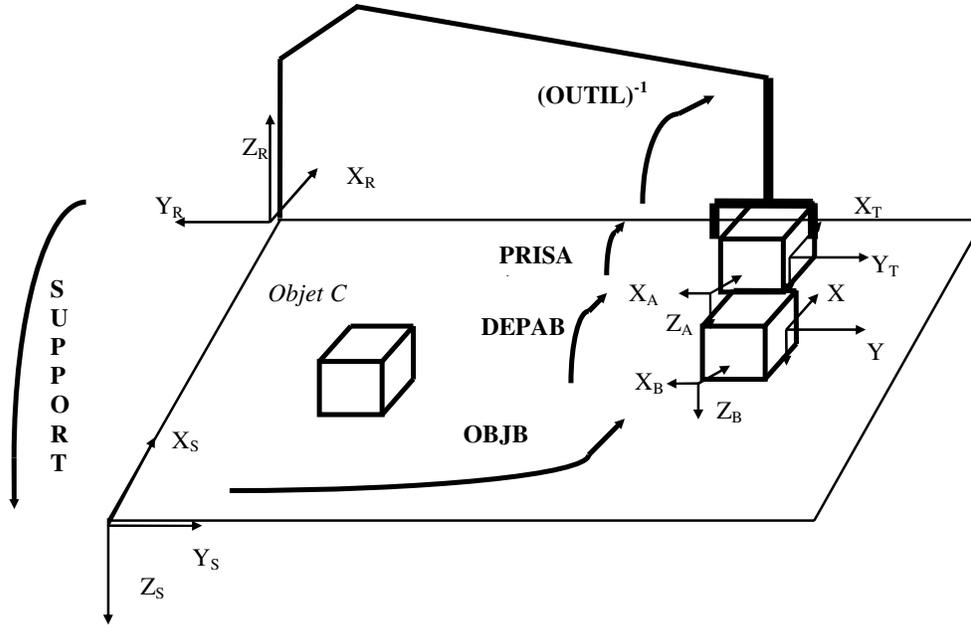


Figure 3-3-d : la pince dépose l'objet A sur l'objet B

-IV - 2 Exemple de programme en langage de niveau objet
et programme de niveau effecteur engendré (en langage VAL)

a- Programme entré par l'utilisateur

Il est utile de préciser au préalable que dans ce programme, nous considérons que le repère attaché à chaque objet est placé en son centre de gravité.

Partie « DECLARATIONS »

ORIG = TRANSL (- 250,660, -450)	
TR 1 = TRANSL (100)	
TR 2 = TRANSL (0, - 100)	Déclaration
AXEX = TRANSF (ORIG: TR 1)	du
AXEY =TRANSF (ORIG: TR 2)	repère
SUPPORT=FRAME(ORIG, AXEX, AXEY)	du support
OUTIL = TRANSL (0,0,90)	Déclaration de l'outil
TR 3 = TRANSL (100,100, - 14,5)	
RTI = ROTZ (-90)	Déclaration de l'objet A et
OBJET OBJA = TRANSF(TR 2: ROT 3)	de sa situation
position prise : PRISA = ROTZ (90)	et de sa position de prise
hauteur OBJA : 29	et de sa hauteur
TR 4 = TRANSL (200,200,-16.5)	
OBJET OBJB	Déclaration de l'objet B
position OBJB = TRANSF(TR 4: ROTI)	et de sa situation
position prise:	
hauteur OBJB :33	et de sa hauteur
POSAB = TRANSL (0,0,- 150)	Position intermédiaire
POSI = TRANSF (OBJB : POSAB)	

Remarques : Aucune position de prise n'a été déclarée pour l'objet B. Le système considère alors que c'est la transformation identité.

Les parties en gras sont des réponses du système.

Partie « INSTRUCTIONS »

PREND OBJA ,50,30	Vitesse d'approche = 50, Vitesse de prise =30
DEPLACE A POSI, 50	Vitesse de déplacement=50
DEPOSE SUR OBJB,,10	Vitesse d'approche = 50, Vitesse de dépose=10

b- Programme VAL généré

```
SPEED 50
APPRO SUPPORT : OBJA: PRISA :(OUTIL)-1 , 50
OPENI
SPEED 30
MOVE SUPPORT : OBJA : PRISA :(OUTIL)-1
CLOSEI
DELAY 0.5

DEPARTS 50
SPEED 50
LEFTY
MOVE SUPPORT:POS 1 : PRISA:(OUTIL)-1
SET OBJA =POSI

DEPARTS 50
FLIP
APPRO SUPPORT:OBJB :DEPAB : PRISA : ( OUTIL)-1 , 30
SPEED 10
MOVE SUPPORT : OBJB : DEPAB : PRISA : ( OUTIL )-1
OPENI
DELAY 0,5
SET OBJA = OBJB : DEPAB
```

c- Commentaires

La première remarque qui s'impose est que le programme écrit dans le "macro- langage " comporte très peu d'instructions et beaucoup de déclarations . Ceci est le cas de tous les langages de haut niveau où on essaie de spécifier une tâche par un minimum d'opérations sur les objets.

La comparaison des tailles des programmes source et cible pourrait laisser penser que le programme VAL est plus court que celui entré par l'utilisateur. Il n'en est rien car ce qu'il faut comparer ce sont la partie INSTRUCTIONS et le programme VAL engendré. En effet , dans ce dernier (qui est le programme tel qu'aurait à l'écrire directement sur le contrôleur un familier de VAL) n'apparaît pas la phase de déclarations des transformations et de leurs valeurs.

Sous le système VAL, un utilisateur doit déclarer chaque transformation sous forme de vecteur à six composantes (X,Y, Z, O, A,T). Or même si on maîtrise le langage VAL, il n'existe pas de mécanisme permettant d'exprimer directement la transformation géométrique de rotation sous forme de vecteur (X ,Y, Z,O,A,T). De plus la notion d'angles (O,A,T) est très difficile a percevoir et impossible à utiliser dans la déclaration d'une tâche sans guidage .

Pour cela, nous avons préféré décomposer la transformation (au sens VAL) en translations et rotations qui sont des notions beaucoup plus compréhensibles. La primitive TRANSF permet cependant également de spécifier des transformations comme dans VAL sous forme de vecteur à six composantes. Ainsi les quatre premières lignes de la partie DECLARATIONS qui permettent de créer la transformation SUPPORT, peuvent être remplacées par une seule déclaration : SUPPORT= TRANSF (-250,660,-450,O,A,T) si l'on connaît la position réelle du support par rapport au repère de base du robot, ainsi que les valeurs des angles O,A,T.

Par ailleurs, la façon de présenter les primitives permet à l'utilisateur d'englober sous le même nom les notions d'objet, repère attaché et transformation correspondante. Il peut alors mélanger les types sans avoir à faire un effort d'attention supplémentaire .

Il est à noter également que l'utilisateur ne gère pas l'évolution de la position des objets qui est modifiée automatiquement quand on les manipule . Il en est de même pour la relation entre les objet A et B (position dans la situation de dépose) qui est déduite de la géométrie des objets .

Enfin le changement de configuration (instructions du langage VAL «FLIP » et « LEFTY») est effectué automatiquement sans que l'utilisateur ait à s'en soucier , alors que si la programmation avait été faite directement sous le contrôleur VAL, ce changement n'aurait pas été effectué et l'exécution se serait arrêtée à ce niveau avec un message d'erreur .

- CHAPITRE IV -
- INTERACTION ROBOT-ENVIRONNEMENT -

Un capteur d'environnement donne au robot des informations sur l'univers dans lequel il opère: c'est grâce aux capteurs que le rebouclage sur la tâche en cours peut être réalisé.

Parmi les capteurs d'environnement, on distingue les capteurs extéroceptifs qui mesurent des grandeurs physiques, et les capteurs sensoriels qui simulent un sens humain comme la vision ou le toucher.

I - LES CAPTEURS EXTEROCEPTIFS

I-1- Les proximètres ou capteurs de proximité

Ils mesurent des distances très faibles (jusqu'à quelques centimètres) et fonctionnent en général en « tout-ou-rien » pour détecter la présence ou l'absence d'objets. Ils peuvent être optiques, pneumatiques ou électriques.

a- Optique

Il fonctionne par réflexion : un faisceau lumineux éclaire l'objet et la lumière réfléchi donne une information sur la présence de la pièce. Après quantification de la lumière réfléchi, on obtient une information de distance.

Ces capteurs ont une résolution très limitée et nécessitent un étalonnage pour chaque type de pièce. Ils sont utilisés en ambiance très perturbée et déflagrante, la lumière pouvant être véhiculée par des fibres optiques et l'électronique de commande déportée.

Ce type de capteur est peu utilisé en robotique.

b- Pneumatique

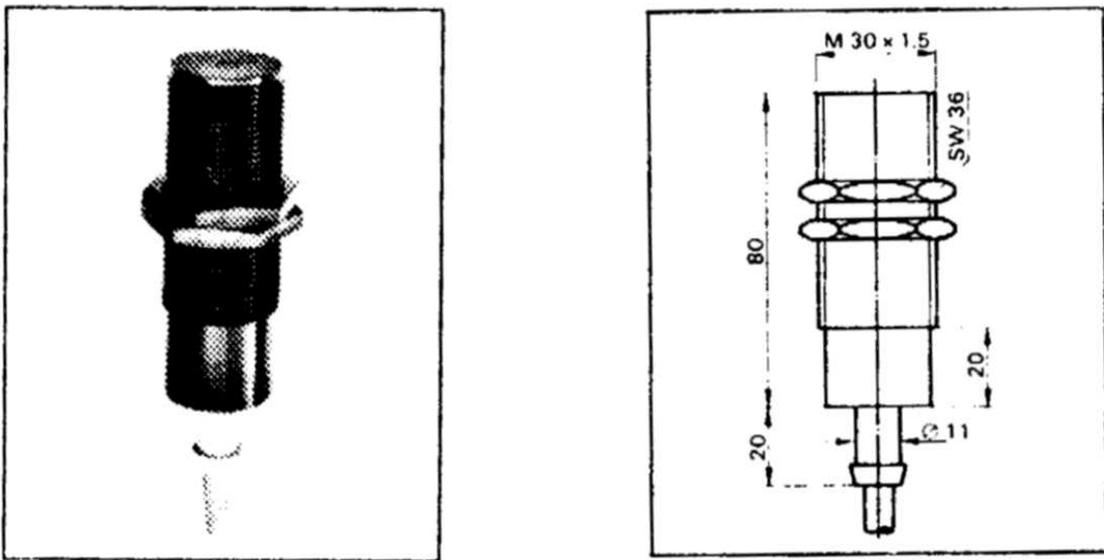
Le principe consiste à créer une différence de pression entre 2 chambres grâce à deux fuites contrôlées: l'une par réglage et l'autre par la variation de distance entre la pièce et la buse de fuite. La différence de pression est ensuite mesurée grâce à un capteur différentiel électrique.

Le capteur de proximité pneumatique est très peu utilisé car il est relativement complexe et nécessite deux sources d'énergie : air comprimé et électricité.

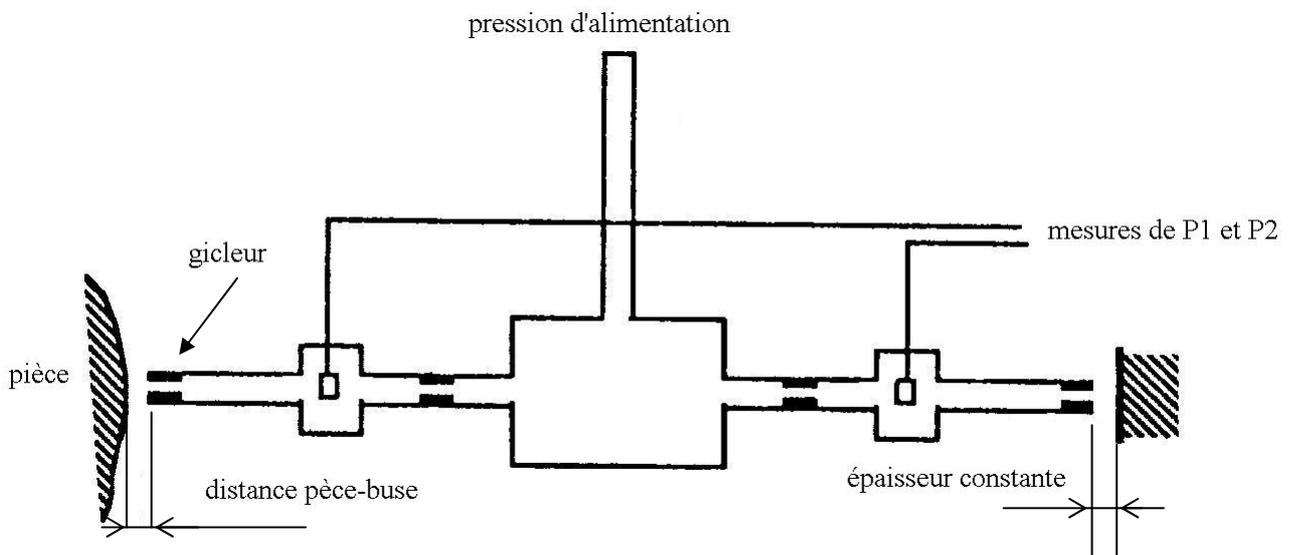
c- Electrique

Capacitifs (pour la détection de matériaux diélectriques) ou inductifs (pour la détection de métaux), ces capteurs sont en général constitués d'un pont de mesure, dont la variation de l'élément actif (épaisseur du diélectrique de la capacité ou variation de permittivité dans le champs de la bobine) engendre une tension qui est mesurée et donne une information de proximité.

Concrètement ces capteurs se présentent sous la forme d'une tige filetée de laquelle sort un câble. Les gammes de mesure vont de 0,1mm à quelques centaines de millimètres. La résolution des capteurs capacitifs est meilleure que celle des inductifs : 10^{-9} m contre 10^{-6} m. L'interfaçage avec le robot se fait en général par une entrée analogique-numérique.



-a- Un capteur de proximité inductif (doc Baumerelectric)



-Figures 4-1 : exemples de proximètres

I-2- Les télémètres

Ils permettent d'effectuer des mesures de distance sans contact. Ils complètent les proximètres par une gamme de mesure différente : quelques dizaines de centimètres à quelques dizaines de mètres.

a- Principe

Un télémètre classique est composé : d'un émetteur, d'un récepteur et d'un système de mesure du temps de parcours aller-retour de l'onde émise. Il existe deux méthodes pour mesurer ce temps de parcours :

- par mesure du temps de vol : déclenchement d'un compteur ou d'un timer à l'émission de l'onde, et arrêt à la réception de l'onde réfléchi.

- par triangulation : mesure du déphasage entre l'onde l'émission et l'onde reçue, la mesure de phase donnant une indication du temps de vol; l'onde émise doit être modulée.

b- Méthodes

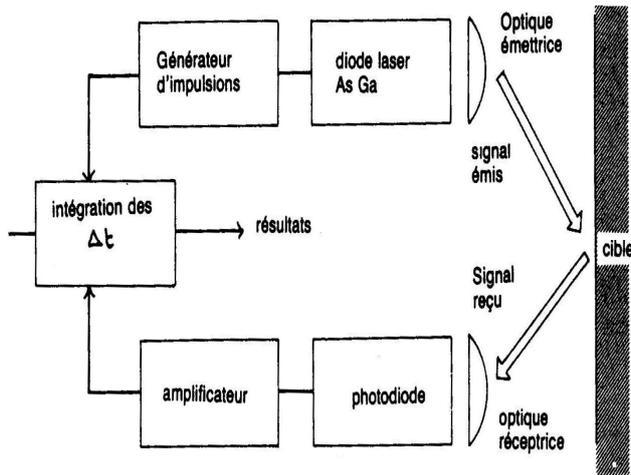
Deux méthodes sont utilisées : le télémètre ultra-son et le télémètre laser.

- * Le télémètre à ultra-son est surtout utilisé en robotique mobile pour la détection d'obstacles et l'aide à la navigation. Sa gamme de mesure s'étend de 0,6m à 6m avec une résolution voisine de 1cm pour une distance de 6m.

- * Le télémètre laser est utilisé pour des distances de 9mm à 1m. Sa précision est de 1mm pour une distance de 2m.

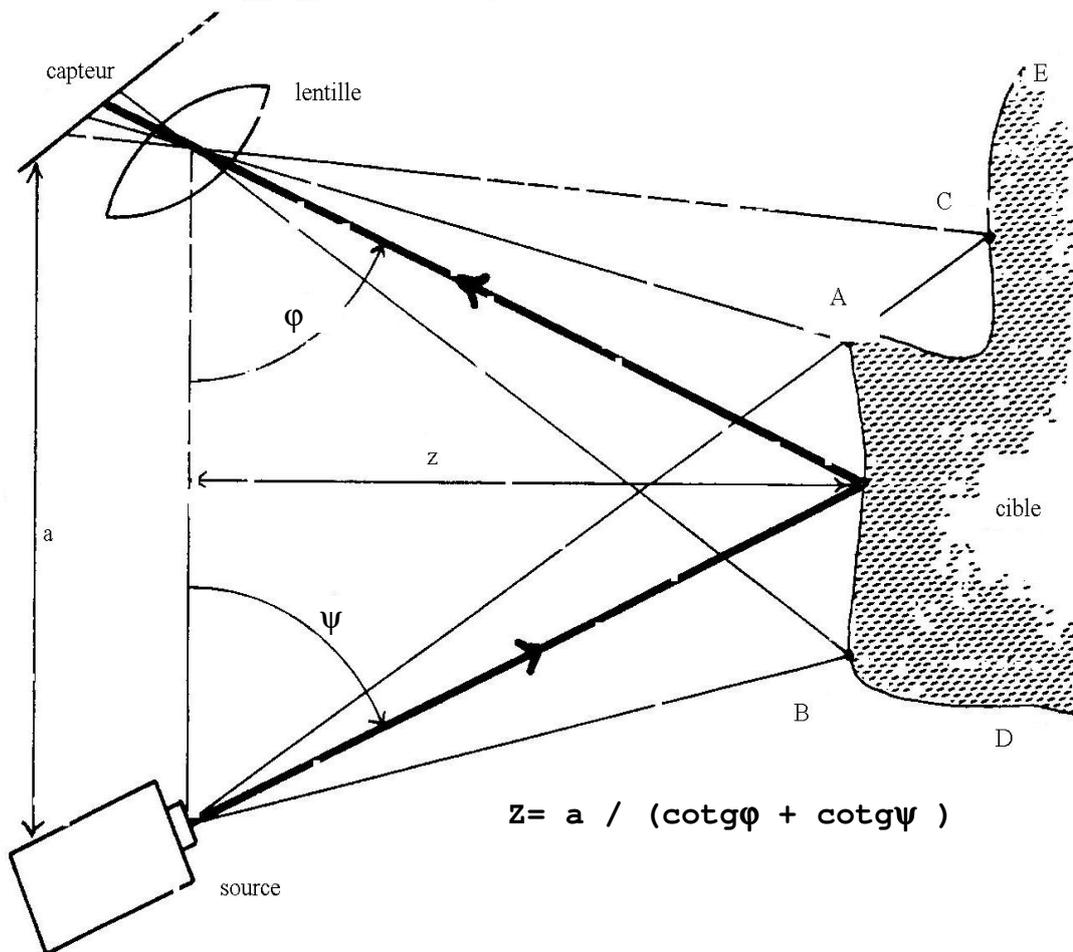
Le télémètre laser est surtout utilisé dans un système de vision active. Pour le suivi de joint en soudure par exemple : on projette une raie laser sur le joint, et une caméra filme la scène au travers d'un filtre passe bande. Il est donc utilisé en vision active essentiellement pour la reconstitution d'images 2 D et 3D. On utilise la projection de rayons lumineux de formes diverses (faisceau, raie, plan) et on utilise comme receptrer des barrettes de photodiodes ou des matrices CCD. L'image est en général prise au travers d'un filtre qui ne laisse passer que la longueur d'onde de la lumière d'illumination.

- * Il existe également des télémètres à balayage : c'est un télémètre laser qui passe dans un système de déviation optique permettant d'effectuer un balayage en deux dimensions de la scène. L'avantage c'est d'obtenir directement une image 3D (puisqu'on connaît les distances et les angles de déviation, donc on obtient l'information de profondeur). Le seul inconvénient est la lenteur de traitement (une image toutes les 2 secondes).



-a- Télémétrie par mesure du temps de vol

-b- Télémétrie laser à balayage



c- Télémétrie par triangulation

- Figure 4-2 : schémas de principe des télémètres

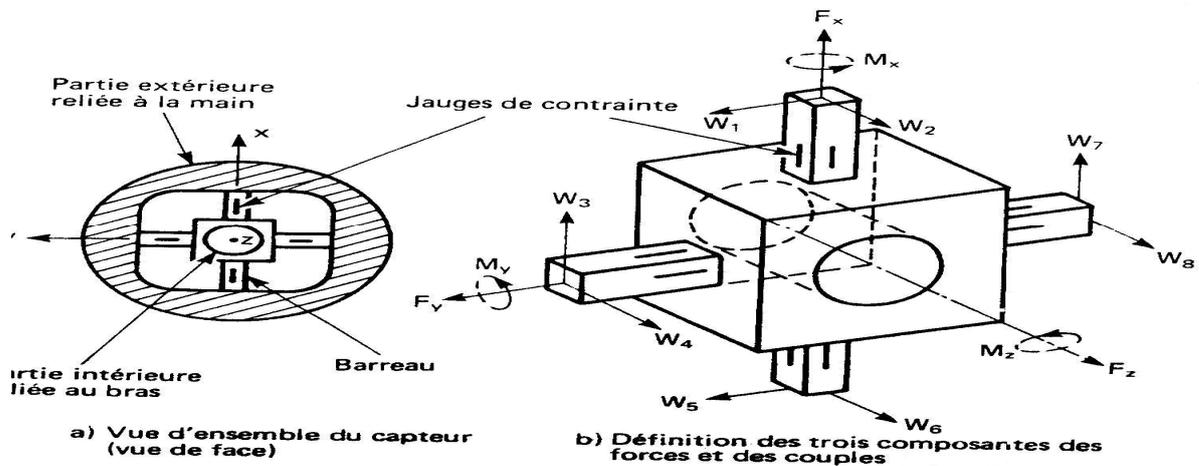
I-3- Capteurs d'efforts

On ramène la mesure des forces à la mesure d'une microdéformation en utilisant des jauges de contrainte, des capteurs inductifs ou capacitifs.

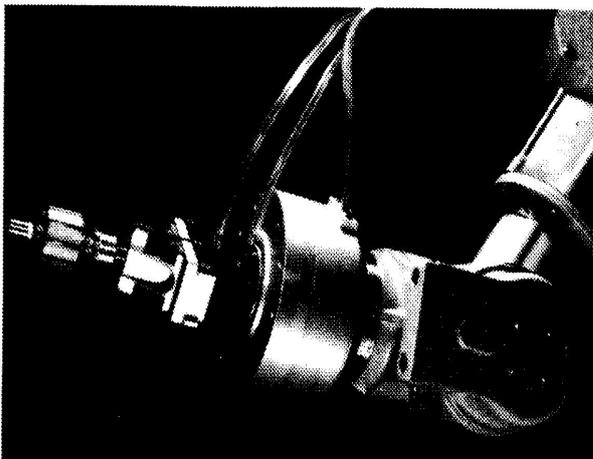
Ces microdéformations sont mesurées sur des poignets spéciaux intercalés entre le dernier segment du SMA et l'outil du robot.

Certains poignets possèdent des actionneurs permettant des microdéplacements. On parle alors de poignets actifs. Pour leur utilisation le SMA se positionne d'abord dans la zone de travail, ensuite le poignet actif compense les erreurs de positionnement grâce à ses microdéplacements, en prenant en compte les informations issues des capteurs d'efforts.

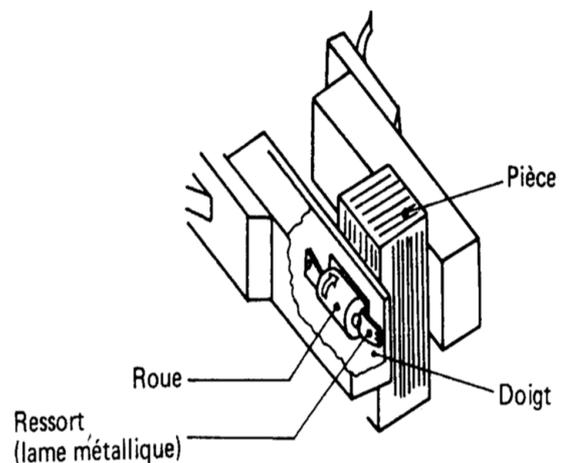
On les utilise principalement dans le domaine de l'assemblage.



-a- Capteur d'effort tridimensionnel



-b- Montage d'un capteur d'effort sur l'outil du robot (doc. Barry Wright Corp)



-c- Exemple de capteur de glissement

-Figure 4-3 : exemples de capteurs de forces

II LES CAPTEURS INTEGRES OU SENSORIELS

Ce sont des capteurs extéroceptifs qui simulent un sens humain. Ils permettent d'acquérir simultanément plusieurs informations.

II-1- Machine de vision ou système de vision artificielle

La vision passive est basée sur l'analyse d'images naturelles numérisées. L'image obtenue à l'aide d'une caméra perpendiculaire à la scène est en deux dimensions (2D) car c'est une projection sur un plan. On extrait de l'image des paramètres standards (contour, longueur, centre de gravité etc...) à l'aide d'algorithmes plus ou moins élaborés.

Si on veut obtenir la troisième dimension (celle de hauteur) on fait de la stéréovision : on utilise soit deux caméras qui donnent donc deux images de la même scène, soit on déplace la caméra pour obtenir une autre image de la même scène sous un angle différent. On reconstitue ainsi une image 3D.

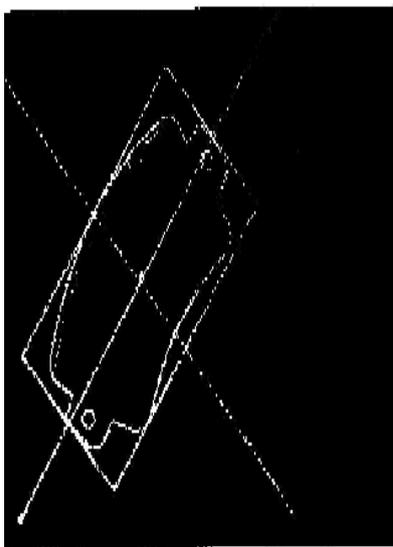
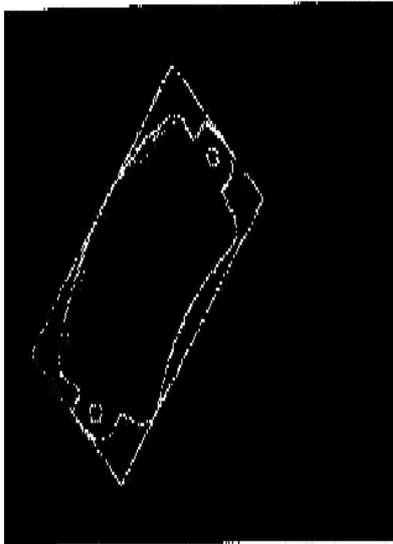
On peut également obtenir la troisième dimension en utilisant une machine de vision active: il s'agit de combiner une caméra avec un système de projection de lumière "active" tel que celui décrit dans la télémétrie laser.

Les systèmes de vision passive sont utilisés surtout dans les domaines du contrôle de qualité, de l'assemblage, de l'analyse et de la microscopie.

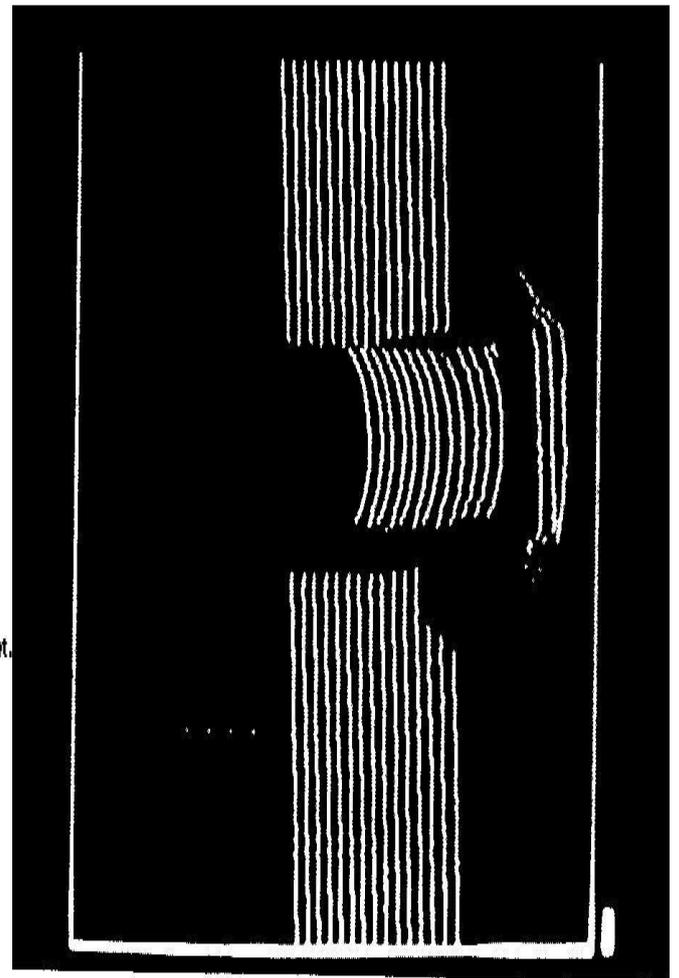
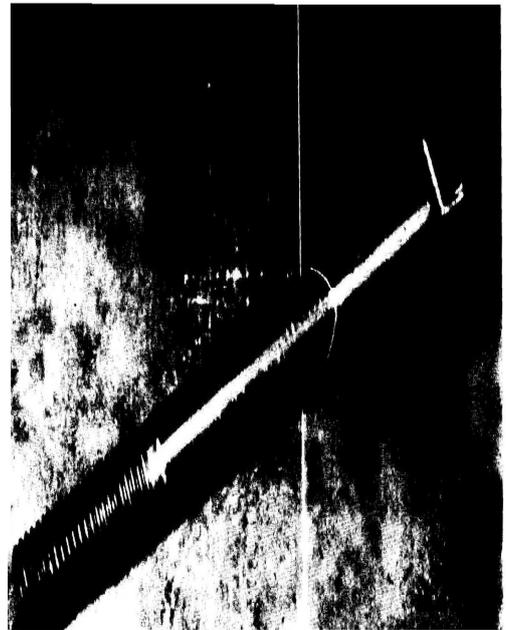
Les problèmes d'interfaçage sont en général limités, les informations échangées étant du type présence d'une pièce, coordonnées du centre de gravité d'une pièce, nombre de pièces d'un certain type etc...

La transmission d'informations plus complexes, comme la localisation des obstacles ou la modélisation d'environnement, nécessitent l'emploi d'un langage de primitives propres au système de vision.

Il est difficilement envisageable de reboucler un système de vision sur l'asservissement ou même sur la génération de trajectoire du contrôleur du robot, compte tenu de la lenteur du traitement des images. Il existe quand même des applications opérationnelles, et l'intégration matérielle poussée (câblage hardware des algorithmes) va rendre la vision passive plus attractive.



localisation en position et en orientation d'un objet. l'objet (ici une plaquette de frein) est parfaitement connu en nature mais il faut connaître la face sur laquelle il repose et le localiser en position et en orientation. Après extraction du contour on détermine le centre de gravité et les moments principaux d'inertie. (doc. ITMI)



-a- Système de vision passive

-b- Système de vision active

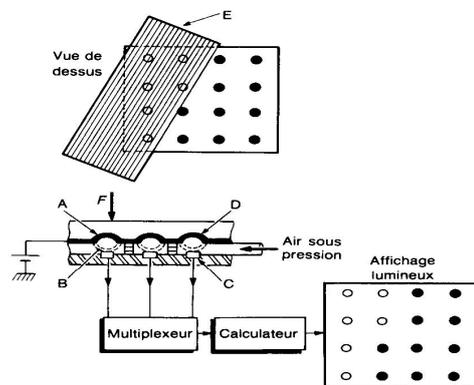
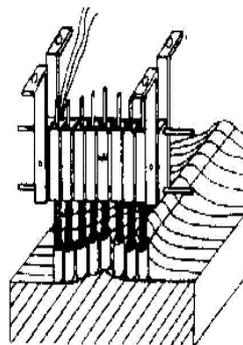
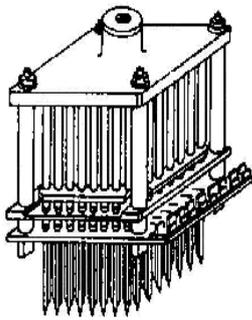
-Figure 4-4 : exemples d'utilisation de la vision dans l'industrie

II-2- Capteurs tactiles

Ils permettent la reproduction du toucher humain. L'information qu'ils délivrent est une répartition de pression sur une surface en contact avec l'objet.

On distingue les peaux artificielles en élastomère dans lequel sont noyées des électrodes qui mesurent la variation d'épaisseur (donc la pression de contact), et les lits de clous à mesure optique. Ces deux capteurs ne diffèrent que dans le mode de mesure des déformations générées par la pression de contact : mesure capacitive, résistive ou optique.

Ces capteurs permettent de mesurer l'effort de pression et de glissement de l'objet, et



- A surface flexible isolante
- B patin sphérique
- C électrode
- D position d'équilibre sans force externe
- E objet saisi par le manipulateur

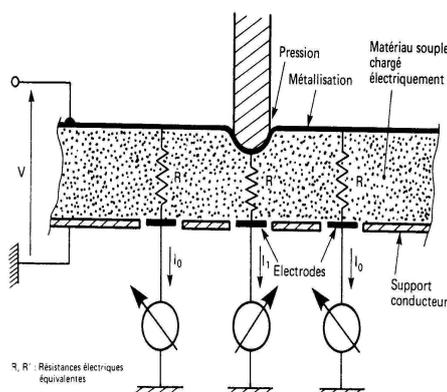
Le capteur est constitué d'une centaine de patins sphériques déformables.

D'après documents Toshiba et Institut für Produktions Technik Automatisierung (IPA)

permettent également de contribuer à la reconnaissance tactile des formes.

-a- Schéma de principe d'un capteur tactile magnétique (matrice d'aiguilles)

-b- Schéma de principe d'un capteur tactile pneumatique



-c- Détecteur d'efforts en peau artificielle (mesure et localisation)

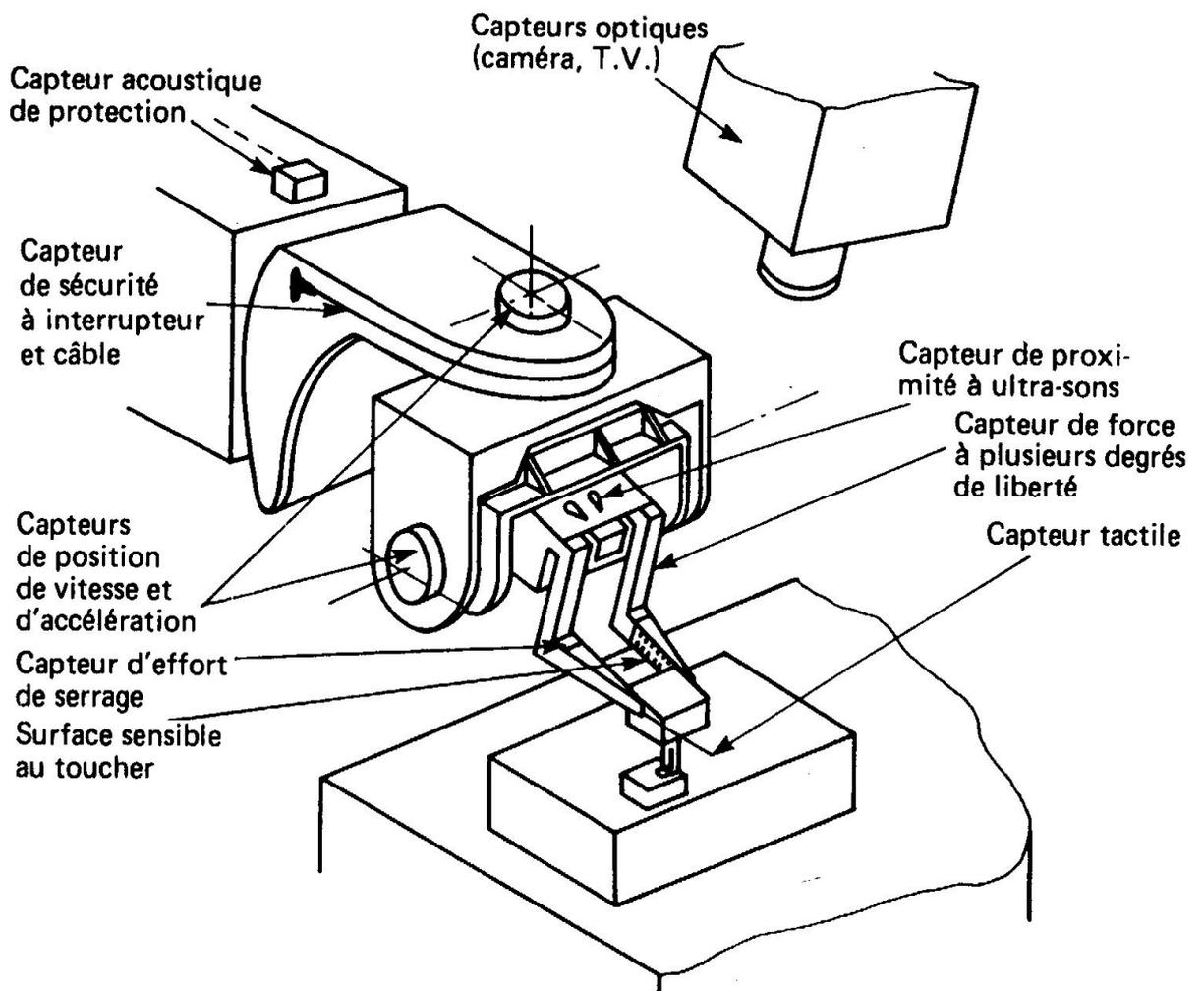
-d- Disposition de n électrodes de mesure constituant un échantillon de peau artificielle

Figure 4-5 : schémas de capteurs tactiles

II-3-Conclusion

Les informations issues des capteurs extéroceptifs et sensoriels sont souvent complémentaires et utilisées pour bâtir un modèle de l'environnement à partir duquel les stratégies de haut niveau sont élaborées. Une autre utilisation des capteurs d'environnement est de réaliser des actions réflexes, qui n'interviennent que dans des phases d'activités précises. Par exemple la saisie automatique d'objets à partir d'informations télémétriques, l'insertion d'un goujon dans un trou en se servant de capteurs d'efforts en assemblage, le suivi de joint en soudure avec un télémètre laser, ou le recalage par un système de vision.

Les capteurs extéroceptifs et sensoriels ne sont en général pas intégrés à la commande du robot: leur degré de complexité étant encore très élevé, ce sont des machines spécialisées et souvent autonomes, qui sont reliées au contrôleur du robot par des liaisons informatiques normalisées.



- Figure 4-6: exemple de mise en place d'un équipement sensoriel

-III- EXEMPLE DE TACHE METTANT EN OEUVRE UN SYSTEME DE VISION

Reprenons l'exemple du chapitre III et supposons que les emplacements des objets A et B seront donnés par la vision, et que nous utilisons le même langage de niveau objet.

L'utilisation de la vision nécessite une calibration (ou étalonnage) entre les systèmes robot et vision . Cette dernière est effectuée en dehors de la déclaration de la tâche .

III-1- Calibration

* Calibration caméra -support: il s'agit d'abord de faire un étalonnage, c'est à dire calculer le coefficient "m" qui permet d'établir une correspondance entre une grandeur mesurée dans le repère du plan image R_i en pixels (cf figure 4-7), et la même grandeur mesurée dans le repère du support R_s en millimètres. Ensuite il faut établir la relation qui permet de passer des coordonnées du point P_i (image d'un point P) exprimées dans le repère R_i : $P_i(x_i, y_i, z_i)$, aux coordonnées du point P exprimées dans le repère R_s : $P(x_s, y_s, z_s)$.

En termes de transformations homogènes cela revient à déterminer la matrice de passage TSI (cf figure 4-8) qui donne la position et l'orientation du repère R_i par rapport au repère R_s . Quant au point P par rapport à R_s il sera donné par sa matrice OBJ, produit de deux transformations: $OBJ = TSI * POS_I$ (POS_I donne les coordonnées de P dans le repère image).

Remarque:

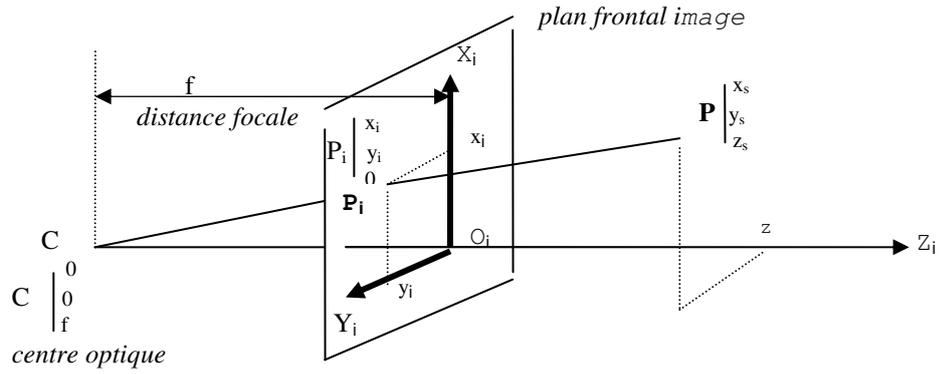
Le coefficient m est implicitement intégré dans les valeurs de P_x , P_y et P_z de la matrice TSI, donnée en termes de transformations géométriques par l'expression:

$$TSI = \text{Transl}(X_s, P_x) * \text{Transl}(Y_s, P_y) * \text{Transl}(Z_s, P_z) * \text{Rot}(Z_s, \theta).$$

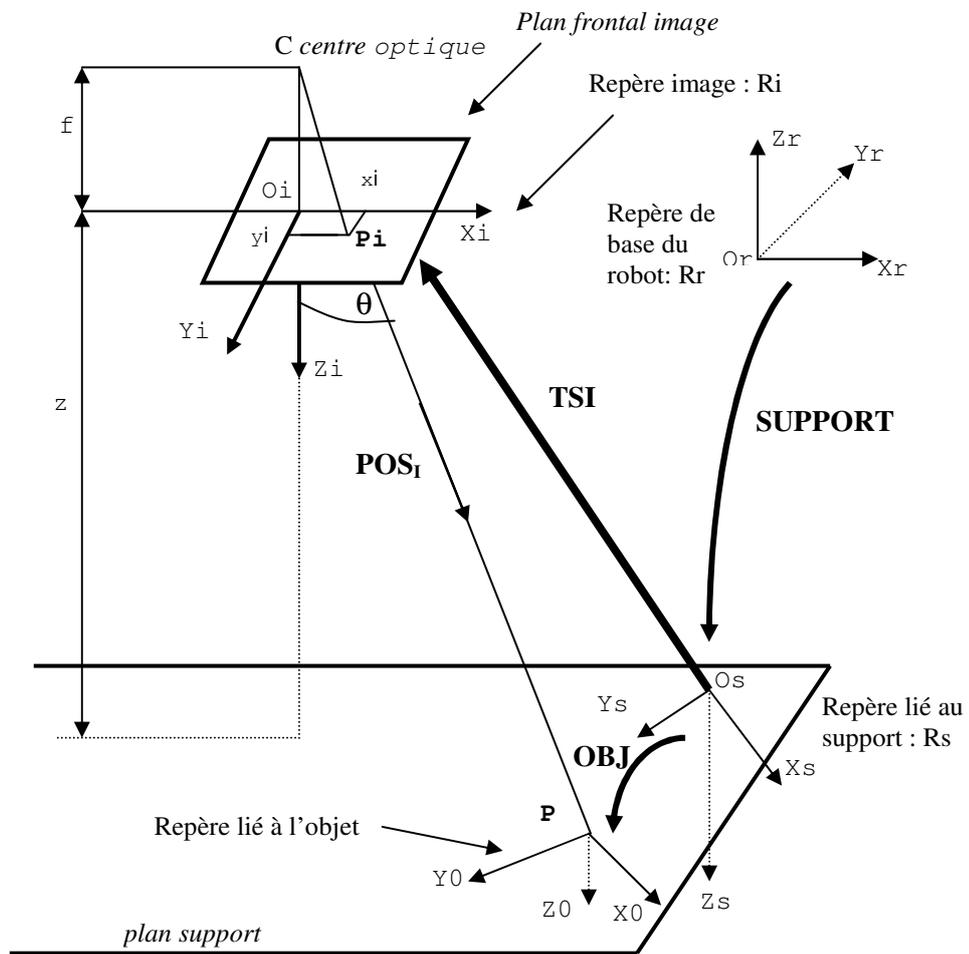
* Calibration support-robot : il s'agit de passer des coordonnées d'un point exprimées dans le repère R_s aux coordonnées de ce même point exprimées dans le repère R_r du robot. Il faut donc déterminer la matrice de transformation homogène SUPPORT qui donne l'emplacement du repère R_s par rapport au repère R_r .

La calibration étant faite, pour saisir un objet avec la pince du manipulateur il faut d'abord extraire de l'image les coordonnées du centre de gravité de l'homologue de l'objet dans le plan image, puis en appliquant la relation trouvée lors du calibrage de la caméra, calculer les coordonnées de l'objet dans le repère R_s . Il suffit ensuite de se servir de la transformation SUPPORT pour exprimer les coordonnées de l'objet dans le repère du robot:

$$POS = \text{SUPPORT} * \text{OBJ} = \text{SUPPORT} * \text{TSI} * POS_I .$$



-Figure 4-7 : modèle géométrique de la formation des images



$$TSI = \text{Transl}(P_x, P_y, P_z) * \text{Rot}(Z_s, \theta).$$

$$m = (f + z) / f$$

$$X_s = m(X_i \cos\theta - Y_i \sin\theta) + P_x.$$

$$Y_s = m(X_i \sin\theta + Y_i \cos\theta) + P_y.$$

$$Z_s = f(m-1) + P_z = 0.$$

Dans l'image l'angle θ sera donné par: $\theta = (O_i Z_i, O_i P_i)$.

-Figure 4-8 : modélisation de l'ensemble vision-robot

III-2- Programme source

Partie «DECLARATIONS»

VISION (SUPPORT)

OUTIL = TRANSL (0,0,90)

OBJET OBJA

position OBJA : VISION

position prise : PRISA = ROTZ (90)

hauteur OBJA : 29

VISION (OBJB)

position prise :

hauteur OBJB : 33

POSAB = TRANSL (20,30,-100)

Partie «INSTRUCTIONS »

VISION (OBJA)

PREND OBJA , 50,30

VISION (OBJB)

POSI = TRANSF (OBJB : POSAB)

DEPLACE A POSI, 50

DEPOSE SUR OBJB,,10

Remarque : les parties en relief sont des réponses des système.

III-3-Programme cible

VISION (OBJA)

PAUSE

SPEED 50

APPRO SUPPORT : OBJA : PRISA : (OUTIL)⁻¹ ,50

OPENI

SPEED 30

MOVE SUPPORT : OBJA : PRISA : (OUTIL)⁻¹

CLOSEI

DELAY 0,5

VISION(OBJB)

```
PAUSE
SET POSI = OBJB:POSAB
DEPARTS,50
SPEED 50
MOVE SUPPORT:POSI:PRISA:(OUTIL)-1
SET OBJA = POSI
DEPARTS,50
APPRO SUPPORT:OBJB:DEPAB:PRISA:(OUTIL)-1 ,30
SPEED 10
MOVE SUPPORT:OBJB:DEPAB:PRISA:(OUTIL)-1
OPENI
SET OBJA = OBJB:DEPAB
```

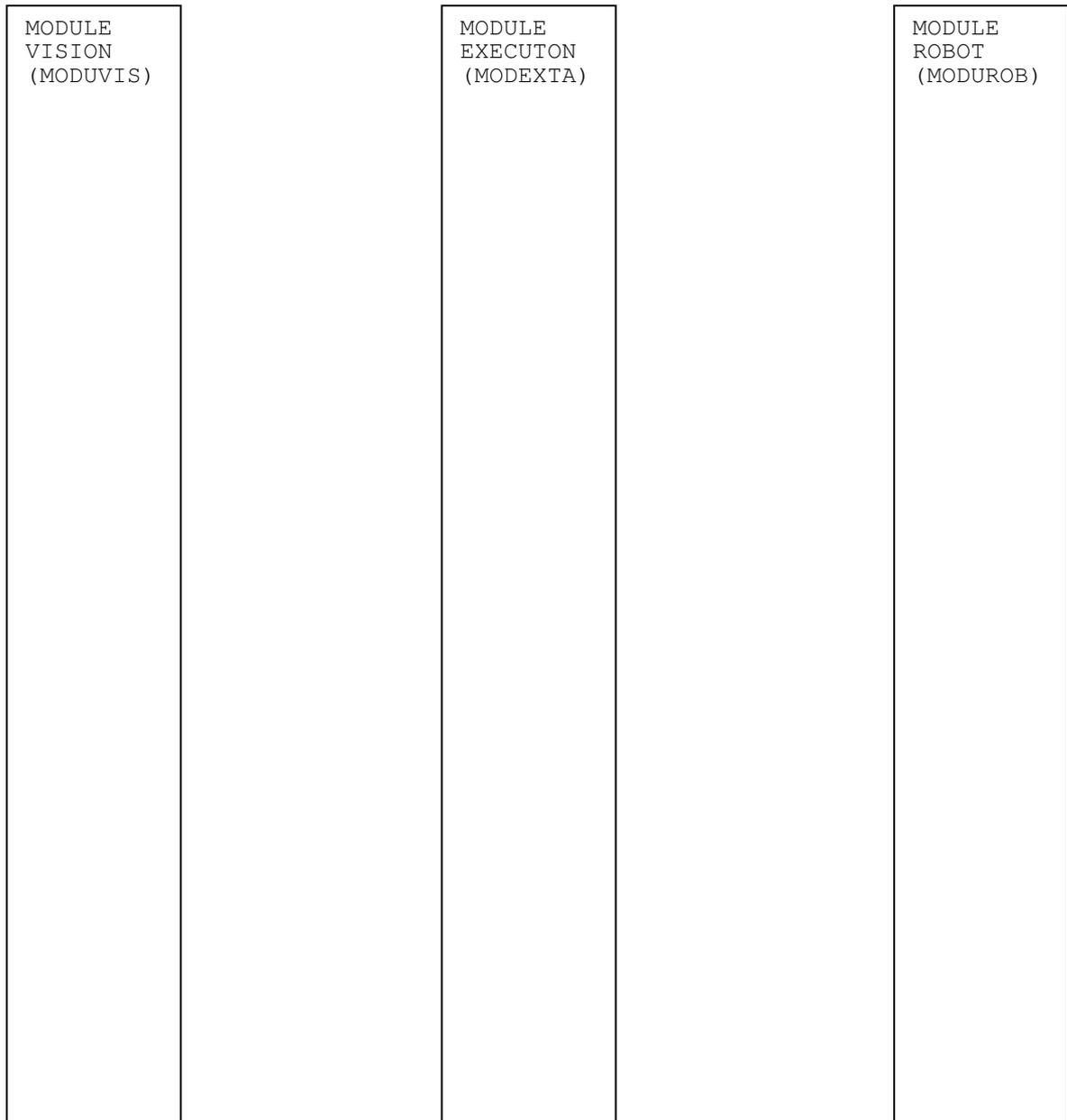
III-4- Commentaires

On remarque dans la partie DECLARATIONS les deux façons de déclarer les objets, ainsi que la spécification de la hauteur. Il ne faut pas perdre de vue que l'on fait de la vision 2D, donc on n'a aucune information de hauteur ; c'est à l'utilisateur de la donner dans le modèle de l'objet. Il est à noter au passage qu'elle est indispensable pour le calcul de la transformation DEPAB qui elle, n'est pas déclarée du tout. Le fait de ne pas spécifier une position de prise pour l'objet B signifie qu'elle est confondue avec la position de référence de l'objet qui est le centre de gravité.

Dans la partie INSTRUCTIONS, la prise d'image de l'objet B doit avoir lieu avant la spécification de la position intermédiaire, car la transformation POSI dépend de la position de l'objet B qui est donnée par la vision.

Comme le programme engendré par la spécification d'une tâche contient du code "vision" et du code "VAL" qui seront exécutés par deux modules différents (MODUVIS et MODUROB), le module maître (MODEXTA) qui supervise l'exécution de la tâche doit assurer la communication et la synchronisation entre les deux modules. Ainsi on remarquera dans le "code cible " les ordres " PAUSE " qui suivent les instruction de vision et qui suspendent momentanément l'exécution d'un programme VAL, permettant ainsi au système d'envoyer les coordonnées des objets pendant la phase d'exécution.

Le protocole de transmission pour l'exécution de la tâche est schématisé par la figure 4-9. On remarquera le fonctionnement en mode maître-esclaves. La transmission MODEXTA-MODUVIS est toujours synchrone alors que la transmission MODEXTA-MODUROB s'effectue aussi bien en synchrone qu'en asynchrone synchronisé : MODEXTA peut envoyer un message à MODUROB sans attendre la réponse de ce dernier . Mais pour communiquer à nouveau avec lui, il est obligé de se resynchroniser. Cet asynchronisme a pour avantage de permettre aux deux modules (vision et robot) de travailler en parallèle.



- Figure 4-9 : Représentation séquentielle de l'exécution de la tâche et du protocole de communication

- CHAPITRE V -
- CONCLUSION -

Le monde industriel a vu naître plusieurs générations de robots, en fonction de l'évolution de la technologie et donc de l'intégration de ces nouvelles techniques dans les robots.

Les robots de première génération étaient destinés à des tâches simples, répétitives, souvent dans un environnement hostile (peinture au pistolet, manutention en milieu nucléaire, forge etc...). Ce ne sont que des machines-outils à commande numérique très évoluées, qui **fonctionnent généralement en boucle ouverte**.

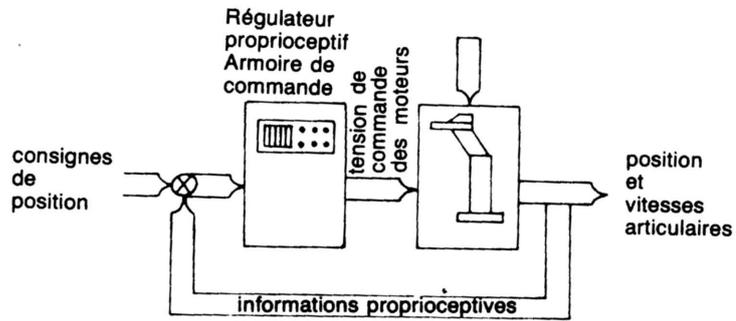
Les robots de deuxième génération sont capables de reconstituer et d'exécuter des trajectoires et des tâches complexes par rapport aux précédentes (usinage, assemblage). Ils nécessitent donc l'utilisation de capteurs proprioceptifs. Par conséquent ils possèdent un **fonctionnement en boucle fermée au niveau de la structure interne mais en boucle ouverte au niveau de la tâche**. Ce sont de véritables automates industriels.

Les robots de troisième génération ne sont pas fondamentalement différents des précédents, mais on enregistre le franchissement d'un pas supplémentaire dans le sens de l'amélioration de l'automatisme et la versatilité. Le concept de versatilité englobe deux notions : celle d'adaptativité et celle de polyvalence.

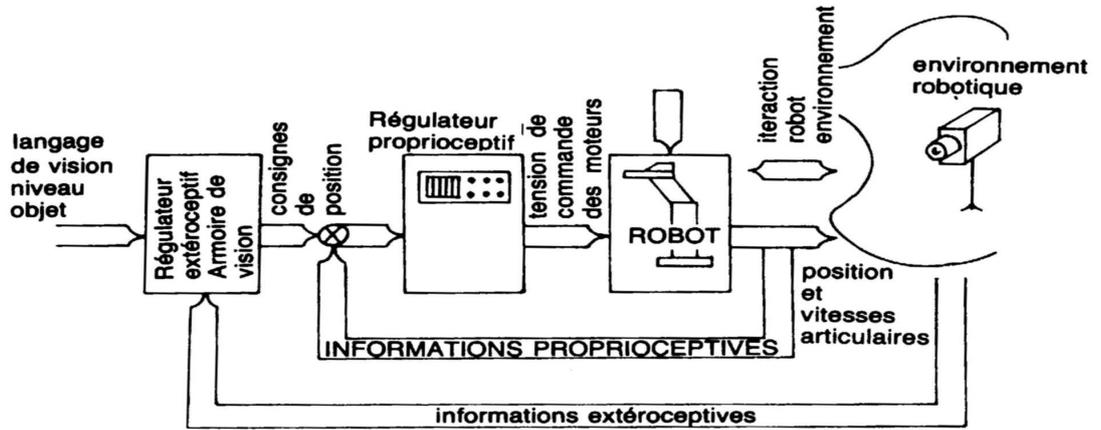
Dans l'adaptativité, il y a la notion de temps réel c'est à dire une capacité dynamique dans l'exécution de la tâche. Par conséquent on a un **fonctionnement en boucle fermée**.

Dans la polyvalence il y a la notion de temps à moyen terme : c'est la capacité matérielle et logicielle qui permet au robot de **changer de tâche sans changer de structure**, mais avec l'intervention de l'opérateur (pour le changement d'outil par exemple).

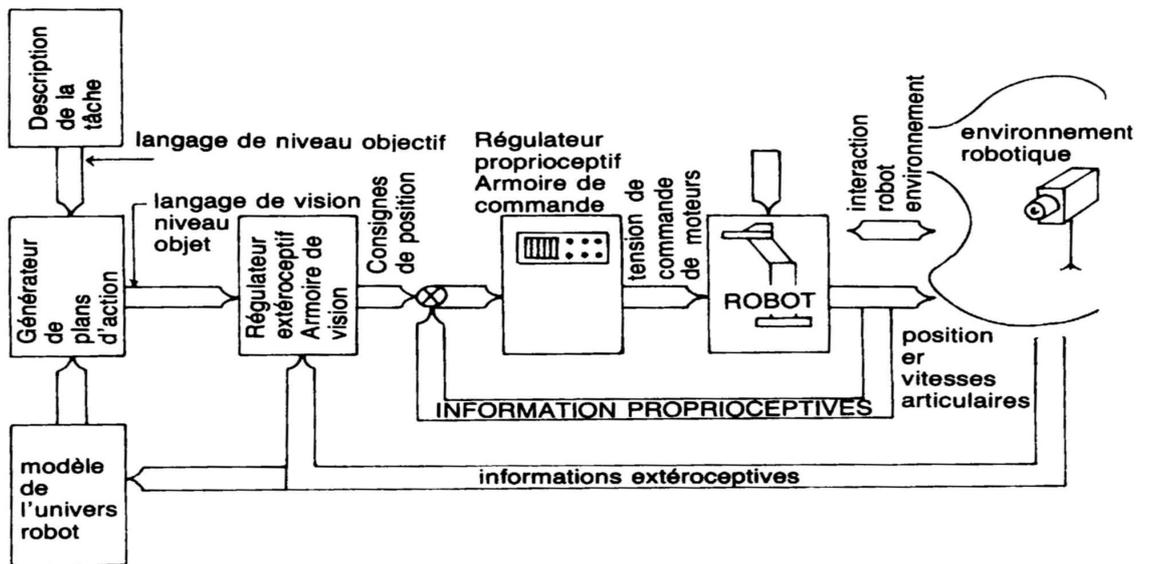
L'objectif de la recherche est d'intégrer la notion de polyvalence dans celle d'adaptativité en vue de **l'accroissement de la puissance décisionnelle** (changement d'outil par le robot lui-même par exemple).



Architecture d'un système robotique de 1ère génération.



Architecture d'un système robotique de la seconde génération



Architecture d'un système robotique de troisième génération

BIBLIOGRAPHIE

Chapitre 1:

- [1] Coiffet P. « Les Robots, tome 1 », édit. Hermes, 1984
- [2] Hamdi H. « Conception d'une station de programmation de robots prenant en compte un système de vision », thèse de docteur ingénieur, UTC-Compiègne, décembre 1986
- [3] « Les Robots, tome 5: programmation », édit. Hermes, 1984

Chapitre 2:

- [4] Hamdi H. « Modélisation et commande en point à point du manipulateur Armdroids1500 », mémoire de DEA, UTC-Compiègne, septembre 1984

Chapitre 3:

- [5] C.Melin & Hamdi H. « Les langages de programmation des robots », encyclopédie Techniques de l'Ingénieur, volume Robotique R7575, janvier 1987

Chapitre 4:

- [6] Lurgeau C. & Parent M. « Les machines de vision en productique », édit. E.T.A., 1985
- [7] Foulc J.N. & Lopez P. « Introduction à la robotique, t 1 & 2 », édit.édiTESTS, 1990

Chapitre 5:

- [8] Dupourque V. « Les controleurs de robots », édit. Inria, collection novotique, 1986

Annexes:

- [9] Boulouika T. & Saadna Y. « Modélisation et commande du robot Roby1 en point à point », mémoire d'ingénieur, IEUC-Constantine, septembre 1996
- [10] Bareche M.E.A. & Touat A. « Modélisation du robot IR52C et génération de trajectoire », mémoire d'ingénieur, IEUC-Constantine, juin 2000
- [11] Talbi L. & Ali S. « langage de programmation de niveau effecteur pour roby1 », mémoire d'ingénieur, IEUC-Constantine, septembre 1999

[4] Hamdi H. « Modélisation et commande en point à point du manipulateur Armdroids1500 », mémoire de DEA, UTC-Compiègne, septembre 1984

[PAU83] Paul R.P. & Renaud M. & Stevenson C.N.

« A systematic approach for obtaining the kinematics of recursive manipulators based on homogenous transformations », 1^ost I.S.R.R, Bretton woods 1983

[PIE 84] Pieper D.L.

« The kinematics of manipulators under computer control », P.H.D. Stanford University 1969

[LEE 84] Lee C.S.G. & Ziegler M.

« Geometric approach in solving inverse kinematics of PUMA robots », IEEE on aerospace and electronic systems , vol AES 520, n°6 nov 19984

[GOR 84] Gorla B. & Renaud M.

« modèle des robots manipulateurs: application à leur commande », Edit CEPADUES, 1984

- ANNEXE 1-

DESCRIPTION ET MODELISATION DU ROBOT ROBY1

- SOMMAIRE -

-ANN1-I- DESCRIPTION DU MANIPULATEUR ROBY 1	PAGE
-I-1-ENTRAINEMENT DES AXES	64
-I-2- CONTROLE	64
-I-3- DIMENSIONS	64
-I-4- SYSTEME MECANIQUE ARTICULE (SMA)	64
-ANN1-II- SYSTEME DE COMMANDE	
-II-1- LE TERMINAL	66
-II-2- L' AUTOMATE MIDICOM	66
-II-3- CARTE D'INTERFACE DU ROBOT (LM886A)	69
-ANN1-III- MODELISATION GEOMETRIQUE DE ROBY1	
-III-1- ELABORATION DU MODELE GEOMETRIQUE	70
-III-2-RESOLUTION DU PROBLEME GEOMETRIQUE INVERSE	85
-III-3-ETUDE DES CONTRAINTES ETAMBIGUITES SUR LES ANGLES	88
-ANN1-IV- DESCRIPTION DU LOGICIEL ROBY3	
-IV-1-LES DIFFERENTS MODES DU MENU DE ROBY3	90
-IV-2-INSTRUCTIONS DE ROBY3	92
-V-3-CODES ET MESSAGES D'ERREUR	95
-IV-EXEMPLES DE PROGRAMMES EN LANGAGE ROBY3	99

ANN1-I- DESCRIPTION DU MANIPULATEUR

Il s'agit d'un manipulateur pédagogique commercialisé par la firme allemande LUCAS NULLE (référence LM8862) et dénommé « **ROBY 1** »

-I-1-ENTRAINEMENT DES AXES

Les 3 axes de Roby 1 et sa pince (cf. FigureI-1) sont tous entraînés par des moteurs à courant continu avec transmission par vis sans fin. Le contrôle des axes se fait par modulation d'impulsion en largeur, alors que la commande de la pince est simplement contrôlée par une temporisation.

Le générateur d'impulsion est du type codeur incrémental, formé d'un barrage photoélectrique avec initiateur à fentes. La résolution est d'environ 0,5 cm en fonction de l'axe. Les détecteurs de fin de course sont des micro-interrupteurs à contact mécanique.

-I-2- CONTROLE

Epaule ou embase : fin de course / générateur d'impulsions

Bras et avant bras : fin de course / générateur d'impulsions.

Pince : fin de course / temporisation.

-I-3- DIMENSIONS

Hauteur max. : 35,25 cm.

Portée max. : 36,61 cm.

Ouverture max. de la pince : 5 cm.

-I-4- SYSTEME MECANIQUE ARTICULE (SMA)

Le manipulateur est représenté par des segments rigides successifs, connectés les uns aux autres par des articulations auxquelles est associé un système à couplage mécanique.

-a- Degrés de liberté

Comme l'indique la figure I-1, ils sont au nombre de trois (plus l'ouverture et la fermeture de la pince). La limitation des mouvements de chaque segment est effectuée par une butée mécanique articulaire. La détection de la butée se fait par un capteur de fin de course.

Les limites des courses par rapport à la position de repos (position de référence) sont les suivantes:

$0^{\circ} \leq \theta_1 \leq 240^{\circ}$; $30^{\circ} \leq \theta_2 \leq 90^{\circ}$; $-120^{\circ} \leq \theta_3 \leq -60^{\circ}$; ouverture pince: 5 cm.

-b- Couplage mécanique

En fait, il s'agit d'un couplage mécanique double reliant l'articulation n°3 à deux autres articulations: d'une part à l'épaule (articulation n°2) et d'autre part à la base de la pince (articulation n°4). Son rôle est de maintenir la pince à la position horizontale par rapport au plan de la base, quel que soit le mouvement effectué.

Par conséquent l'articulation de la pince (art4 d'angle φ) n'est pas contrôlée, elle est automatiquement commandée par ce couplage.

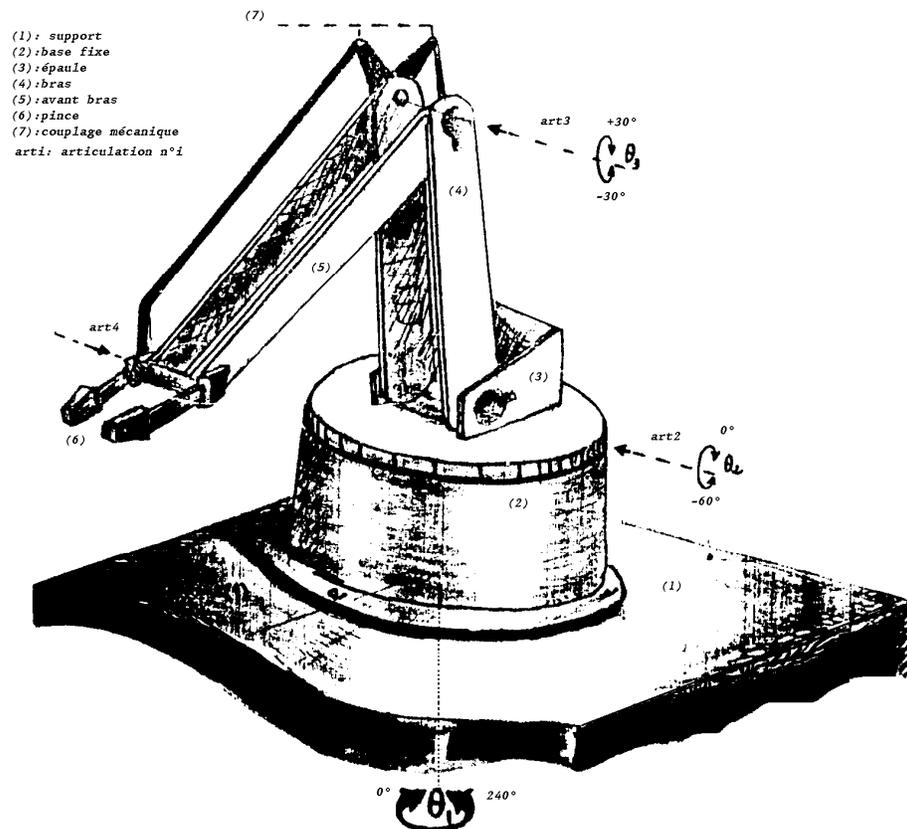


Figure I-1- Architecture du manipulateur

(dans la position de référence : 0, 90, -120)

ANN1-II- SYSTEME DE COMMANDE

-II-1- LE TERMINAL

C'est un compatible IBM PC. Deux logiciels sont utilisés:

- l'un développé par l'institut d'électronique (**robys3**) est un véritable environnement de travail en ligne et hors ligne. Il permet la commande aussi bien en mode guidage et apprentissage (boîte à boutons soft), qu'en mode programme à l'aide d'un langage de niveau effecteur. Une description en est donnée dans l'annexe au paragraphe ann1-IV page 90;
- l'autre livré avec le robot (*LN tools*) est un ensemble d'utilitaires qui gèrent aussi bien l'interface utilisateur que les échanges avec l'automate du robot. Le choix des menus et options disponibles nous offre la possibilité d'écrire des textes sources assembleur, les assembler, les transférer à l'automate et enfin les exécuter. Avec LN-tools dès que l'on tape sur une touche du clavier, le « programme terminal » se charge de transférer son code ASCII à l'automate (MIDICOM) via l'interface série (RS232-C), et de visualiser sur écran le message renvoyé par le MIDICOM.

-II-2- L' AUTOMATE MIDICOM

Le MIDICOM est un automate programmable à base de microprocesseur 8085 de la firme Intel. C'est l'organe qui génère les commandes induisant les déplacements recherchés des différentes articulations du manipulateur.

a. Connecteurs de la carte cpu

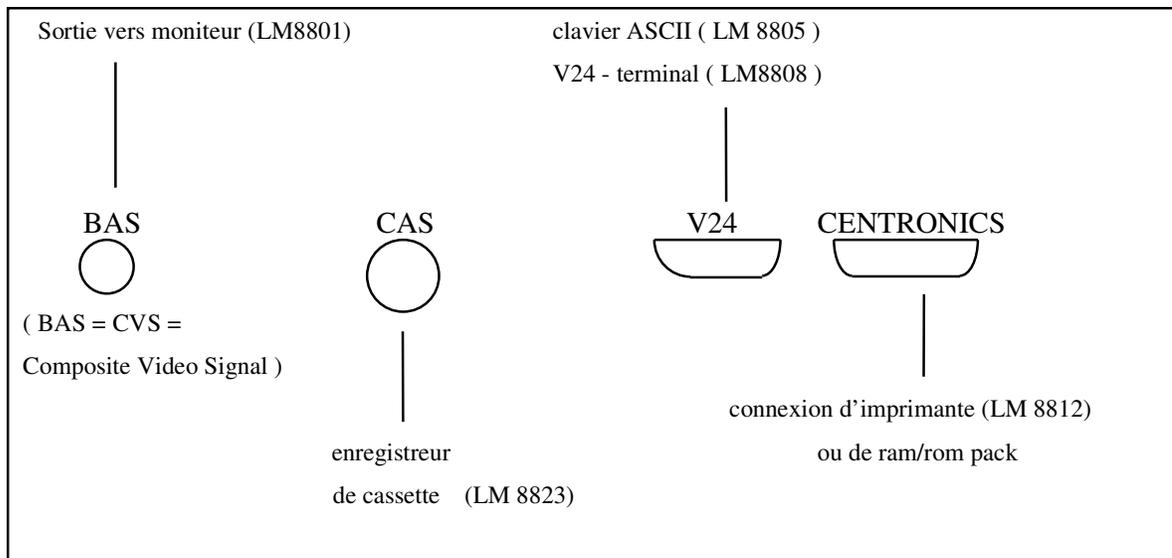


Figure -II-1-Connecteurs de périphériques

La figure II-1 illustre les différents connecteurs reliant la périphérie au MIDICOM :

- Une interface série RS 232-C (V24) pour connecter un PC ou un terminal (clavier et écran).
- Une interface vidéo (BAS signal) pour relier un écran.
- Une interface parallèle (centronics) utilisée pour l'imprimante et aussi pour connecter les RAM- / ROM-PACK, comprenant les programmes à exécuter.
- Une interface série (CAS) pour connecter un lecteur de cassette.

b. Carte cpu du midicom

Cette carte se compose du microprocesseur 8085 avec ses circuits de codage, d'EPROM, de RAM statique et de circuits d'entrée/sortie.

La RAM statique destinée au programme utilisateur (programme ROBOT) est de 40KO, alors que le moniteur stocké en EPROM est de 24 KO.

Le mapping mémoire est donné à la figure II-2.

Parmi les entrées/sorties disponibles, nous nous sommes intéressés uniquement à :

- Un bus d'interface (X1) parallèle à base de circuit 8255 (PIO1) à trois ports de 8 bits utilisés pour la commande du robot;
- Une interface série (RS 232-C) à base de circuit intégré 8251 (UART- 0) que nous utiliserons pour connecter notre terminal.

Adresse	Type	Application
0000	 EPROM 27256 (8K0)	Programme moniteur 24 K0
1000		
2000		
3000		
4000		
5000		
6000	RAM-1 4464 RAM-2 4464 RAM-3 4464 RAM-4 4464 RAM-5 4464 8K0x8	6000 RAM-Moniteur
7000		6800 Pile -Utilisateur
8000		7000 Programme utilisateur
9000		40 K0 - RAM
A000		
B000		
C000		
D000		
E000		
F000		
FFFF		

Figure II-2- Mapping mémoire

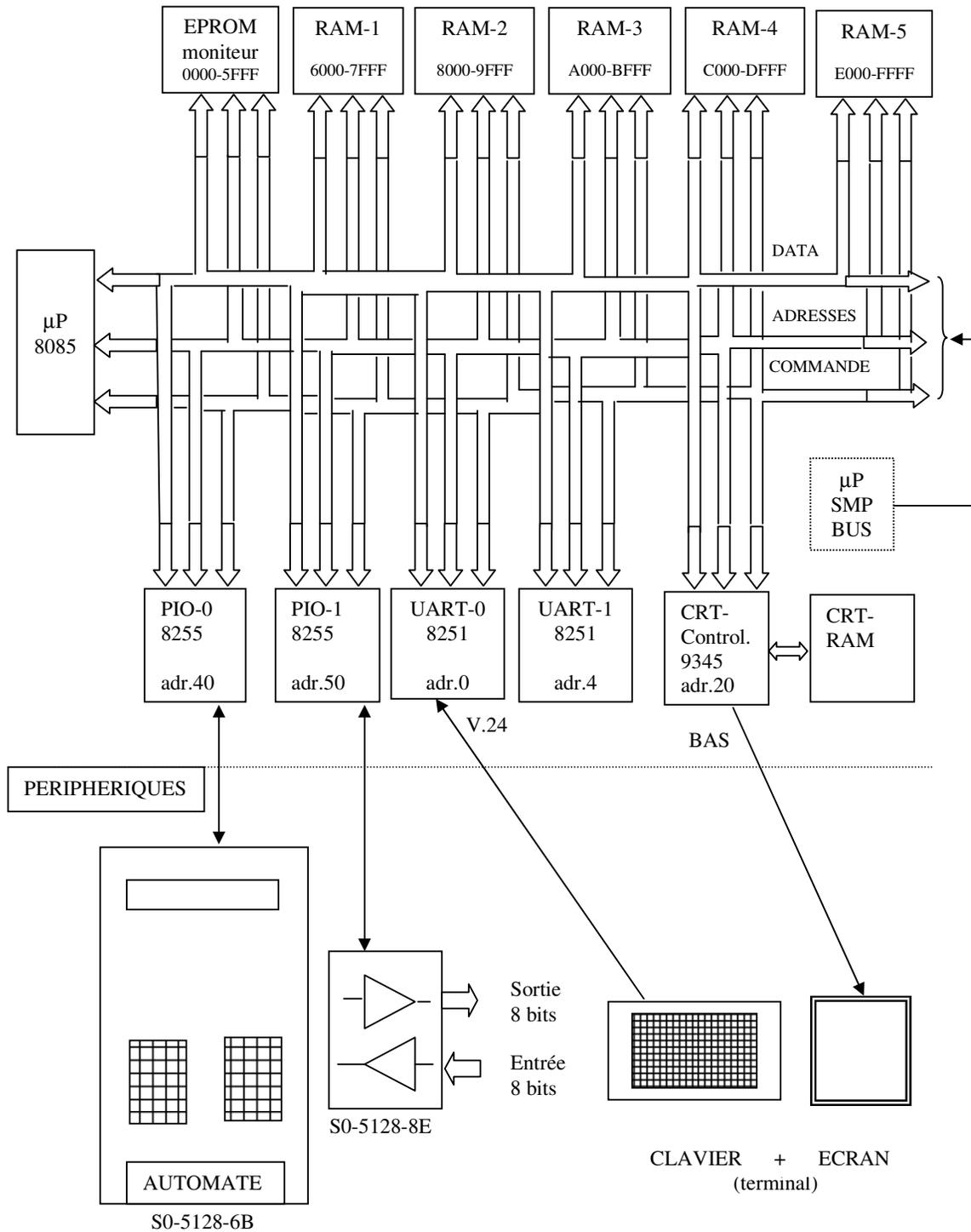


Figure II-3-Schéma de la carte CPU

II-3- CARTE D'INTERFACE DU ROBOT (LM886A)

Roby1 est équipé d'une carte d'interface fixée dans la cavité de l'automate. Son schéma synoptique est donné à la figure II-4.

Cette carte reçoit des données provenant du microprocesseur par le biais du PIO, ces données seront véhiculées à travers des bascules « D ». Les sorties de ces dernières seront amplifiées, puis envoyées via le bus (X2) afin de commander les moteurs du robot.

De même, cette carte reçoit des données provenant du robot à travers le bus (X2) qui seront inversées par des triggers de shmitt et amplifiées par deux quadruples amplificateurs de lignes (3 états), et enfin envoyées au microprocesseur par le biais du PIO (port B).

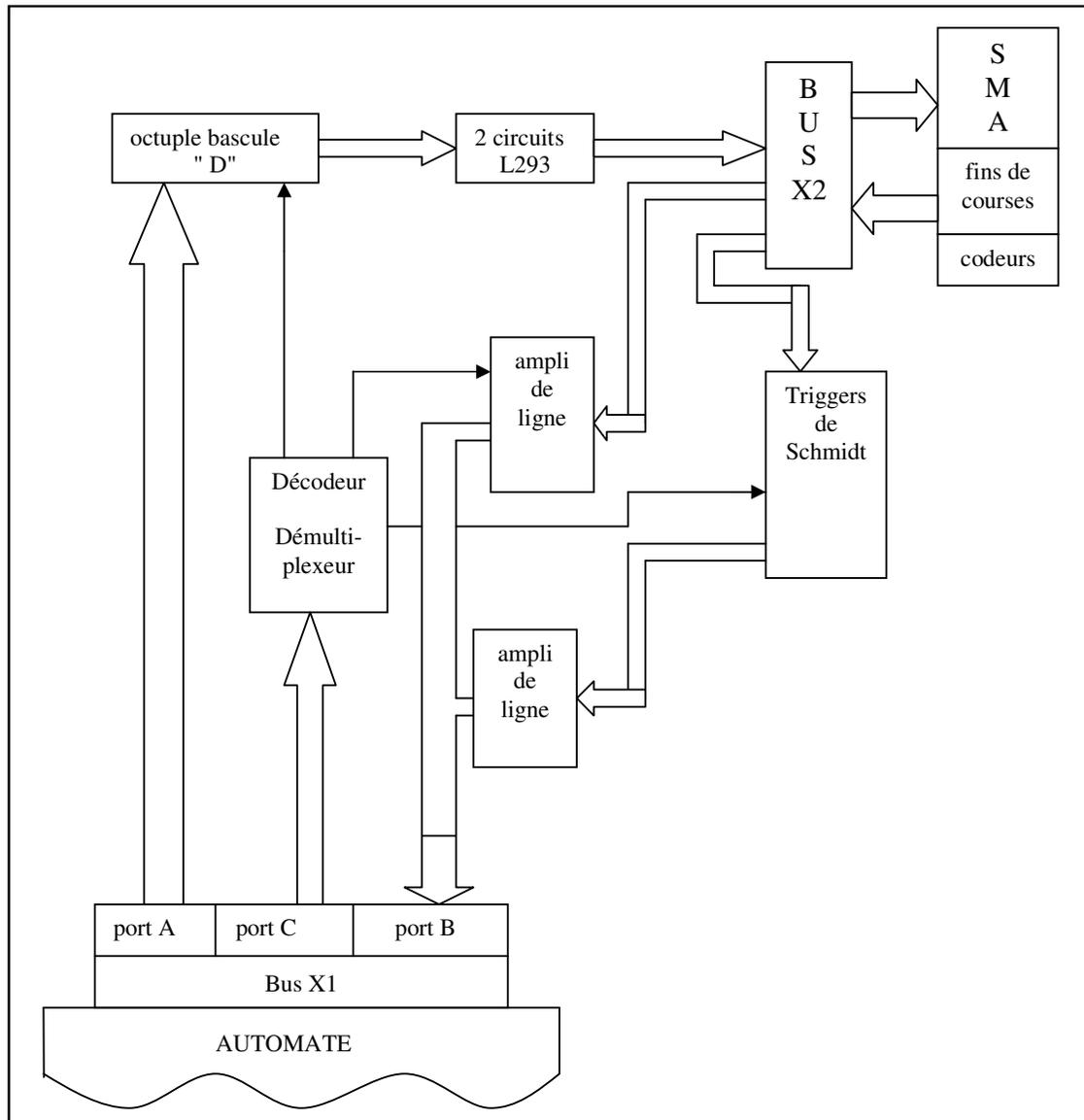


Figure II-4-Schéma synoptique de la carte d'interface du robot

ANN1-III- MODELISATION GEOMETRIQUE DE ROBY1

Dans les commandes cinématique et dynamique où l'on veut contrôler la vitesse lors des déplacements, et même avoir des accélérations importantes, on est obligé d'avoir recours aux modèles cinématique et dynamique qui font intervenir les masses, les forces d'inertie etc...

Dans la modélisation géométrique, nous n'avons fait intervenir que les longueurs des segments et les variables articulaires. Ce type de modélisation convient à la commande en position ou point à point, où la vitesse de déplacement est fixée uniquement par le temps de réponse des dispositifs intervenant dans la structure et la commande du système mécanique articulé (SMA).

-III-1- ELABORATION DU MODELE GEOMETRIQUE

Le manipulateur est représenté par des segments rigides successifs, connectés les uns aux autres par des articulations. En partant de la base pour arriver à l'organe terminal, on affecte à chaque segment rigide un repère. En utilisant les transformations homogènes, on calcule pour chaque repère R_i la matrice de transformation A_i , permettant le passage du repère R_{i-1} au repère R_i . On utilisera la méthode de Denavit-Hartenberg.

Dans la modélisation géométrique utilisant les transformations homogènes, la structure mécanique articulée sera modélisée par une matrice T et l'effecteur par une matrice E.

Soient S_0 la base du SMA et S_n le dernier segment. T est le produit des matrices successives A_i , donnant chacune la position et l'orientation du repère R_i attaché à un segment S_i par rapport au précédent.

$$T = \prod_{i=1}^{i=n} A_i$$

La matrice T donne la situation du dernier segment (support de l'effecteur) par rapport à R_0 .

La situation de l'extrémité de l'effecteur par rapport au repère de référence ($R_{UNIVERS}$) est donnée par la matrice de transformation homogène :

$$Pos = Z * T * E$$

où Z désigne la matrice donnant la position et l'orientation de R_0 par rapport à $R_{univers}$.

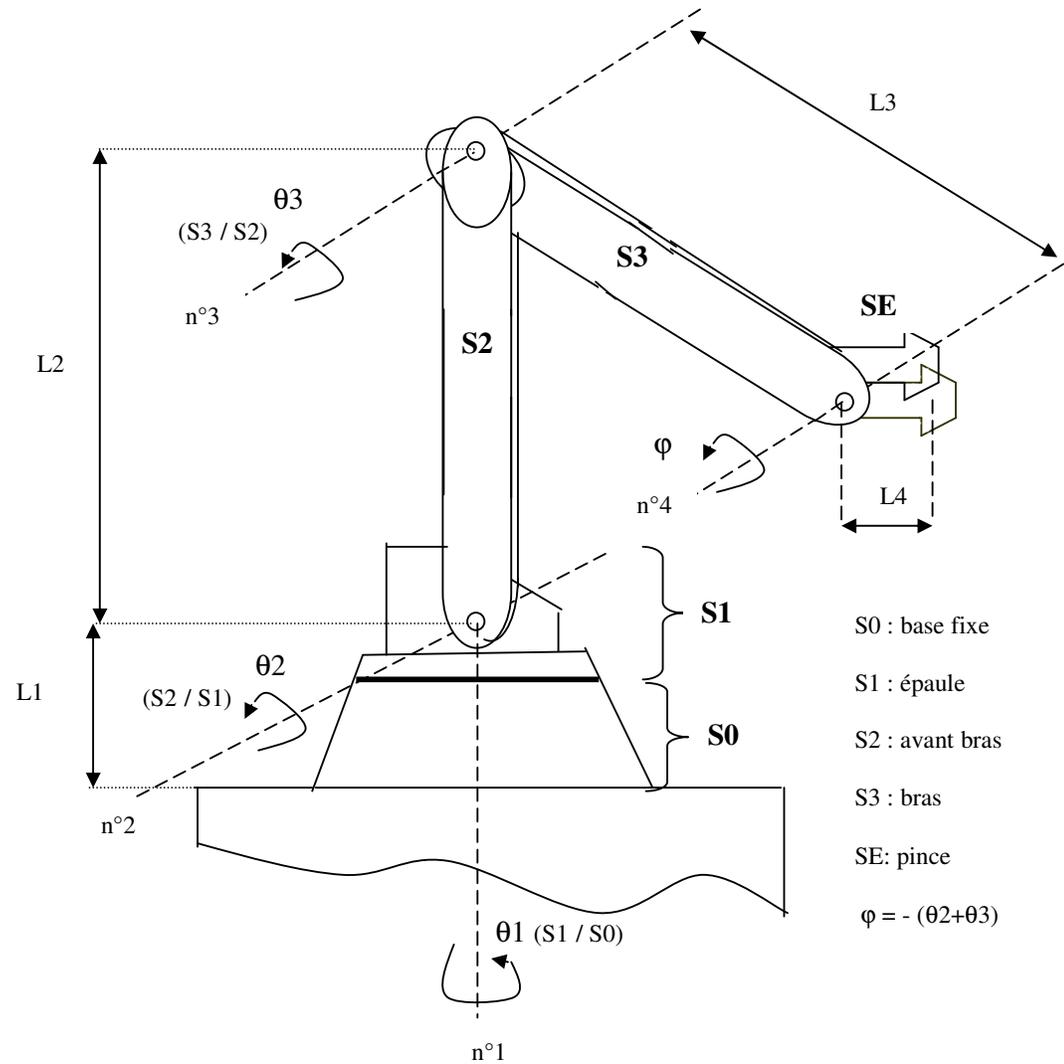


Figure -III-1-Squelette et paramètres structuraux du manipulateur ROBY1

Configuration correspondant à la position du point "référence" :

Articulaires : $\theta_1 = 0^\circ$, $\theta_2 = 90^\circ$, $\theta_3 = -120^\circ$ \longleftrightarrow Cartésiennes ($P_x = 21.7$ $P_y = 0$ $P_z = 16.4$)

Limites des butées articulaires :

$$\left. \begin{array}{l} 0 \leq \theta_1 \leq 240 \\ 30 \leq \theta_2 \leq 90 \\ -120 \leq \theta_3 \leq -60 \end{array} \right\} \longleftrightarrow \left\{ \begin{array}{l} -35.5 \leq P_x \leq 35.5 \\ -30.6 \leq P_y \leq 35.5 \\ 1 \leq P_z \leq 33.5 \end{array} \right.$$

Paramètres structuraux :

$L_{\text{Pince}} \equiv L_4 = 7.5$ mm ;
 $L_{\text{Segment2}} \equiv L_3 = 17.4$;
 $L_{\text{Segment1}} \equiv L_2 = 16.0$;
 $H_{\text{Base}} \equiv L_1 = 10.5$;

1-A- SEGMENT S₁

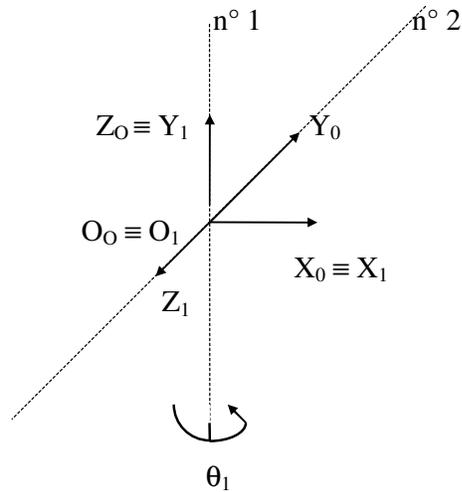


Figure III-2

1-a-1- REPERES

Repère R₁ : O₁ : intersection des axes de rotation n° 1 et 2.

Z₁ : Aligné avec l'axe de rotation n° 2.

X₁ : Comme les 2 axes de rotation n° 1 et 2 se coupent, donc X₁ et aligné avec Z₀ ∧ Z₁ , prenons X₁ = + Z₀ ∧ Z₁ .

Y₁ : Y₁ = Z₁ ∧ X₁ .

Repère R₀ : O₀ : Confondu avec O₁ .

Z₀ : Aligné avec l'axe de rotation n° 1.

X₀ : Comme θ₁ = 0 quand les 2 axes X₀ et X₁ sont alignés et ont même Direction, choisissons : X₀ ≡ X₁ .

Y₀ : Y₀ = Z₀ ∧ X₀ .

1-a-2-. PARAMETRES DE STRUCTURE

a₁ : c'est la distance des axes Z₁ et Z₀. Comme ils se coupent, donc a₁ = 0.

α₁ : c'est l'angle (Z₁, Z₀). α₁ = π / 2.

d₁ : C'est la distance entre O₀ et le pont de Z₀ intersection de Z₀ et X₁

$$\Rightarrow d_1 = 0$$

θ₁ : C'est l'angle (X₀, X₁) ; sur le schéma on a θ₁ = 0 .

I-a-3- TRANSFORMATION HOMOGENE ASSOCIEE AU REPERE R_1 (AU SEGMENT S_1)

Dans le cas où θ_1 est quelconque,

$$\begin{aligned} A_1 = T^1_0 &= R (Z_0 , \theta_1) * T (Z_0 , d_1) * T (X_1 , a_1) * R (X_1 , \alpha_1) \\ &= \text{Rot} (Z_0 , \theta_1) * \text{Rot} (X_1 , \pi / 2) \end{aligned}$$

$$A_1 = \begin{bmatrix} C_1 & -S_1 & 0 & 0 \\ S_1 & C_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_1 = \begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

REMARQUE :

On utilisera dans toute la suite, le même formalisme que précédemment à savoir :

Rot : pour rotation.

Trans : Pour translation.

C_i : Pour $\cos (\theta_i)$.

S_i : Pour $\sin (\theta_i)$.

C_{ij} : Pour $\cos (\theta_i + \theta_j)$.

S_{ij} : Pour $\sin (\theta_i + \theta_j)$.

1-B-. SEGMENT S2

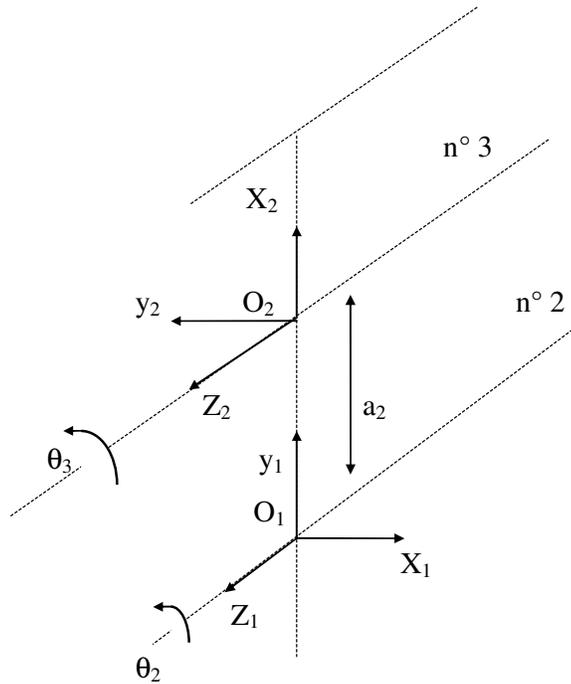


Figure III-3

1-b-1- REPERE : R2

θ_2 : placé à la normale commune aux axes de rotation n° 2 et n° 3, et sur l'axe n° 3 (θ_2 est choisi de façon que d_2 soit nulle).

Z_2 : aligné avec l'axe de rotation n° 3.

X_2 : aligné avec la normale commune aux axes de rotation n° 2 et n° 3, sa direction va de l'axe n° 2 vers l'axe n° 3.

$$Y_2 : Y_2 = + (Z_2 \wedge X_2).$$

1-b-2- PARAMETRES DE STRUCTURE

a_2 : distance des axes Z_1 et Z_2 (donc des axes de rotation n° 2 et n° 3),

$$a_2 = L_2 .$$

α_2 : angle entre les axes de rotation n° 2 et n° 3 , $\alpha_2 = (Z_1 , Z_2) = 0$.

d_2 : distance entre O_1 et le point de Z_1 intersection de Z_1 et X_2 , $d_2 = 0$.

θ_2 : $\theta_2 = (X_1 , X_2)$; sur les schéma , $\theta_2 = \pi / 2$.

1-b-3-TRANSFORMATION HOMOGENE ASSOCIEE AU REPERE R₂ (dans le cas θ₂ quelconque)

$$A_2 = T^2_1 = \text{Rot} (Z_1 , \theta_2) * \text{Trans} (Z_1 , d_2) * \text{Trans} (X_2 , a_2) * \text{Rot} (X_2 , \alpha_2) .$$

$$A_2 = \text{Rot} (Z_1 , \theta_2) * \text{Trans} (X_2 , a_2) .$$

$$A_2 = \begin{bmatrix} C_2 & -S_2 & 0 & 0 \\ S_2 & C_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 & L_2 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} C_2 & -S_2 & 0 & C_2 L_2 \\ S_2 & C_2 & 0 & S_2 L_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = A_2 = T^2_1$$

$$A_2 = T^2_1 = \begin{bmatrix} C_2 & -S_2 & 0 & C_2 L_2 \\ S_2 & C_2 & 0 & S_2 L_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

1-C-SEGMENT S₃

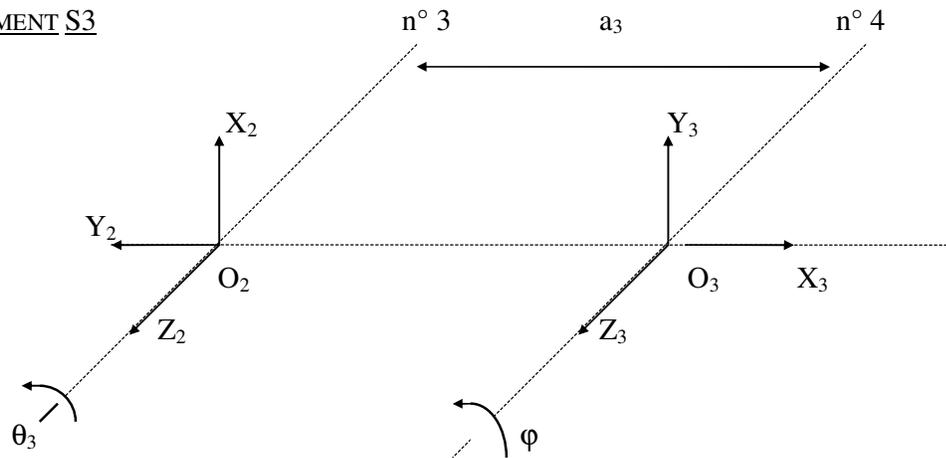


Figure III-4

1-c-1-REPERE : R_3

θ_3 : placé à la normale commune aux axes de rotation n° 3 et n° 4 .

Z_3 : aligné avec l'axe de rotation n° 4

X_3 : aligné avec la normale commune aux axes de rotation n°3 et n°4, sa direction va de l'axe n°3 vers l'axe n° 4 .

Y_3 : $Y_3 = + (Z_3 \wedge X_3)$

1-c-2-PARAMETRES DE STRUCTURE

a_3 : distance des axes Z_2 et Z_3 , $a_3 = l_3$.

α_3 : $\alpha_3 = (Z_2 , Z_3) = 0$.

d_3 : distance entre o_2 et le point de Z_2 , intersection de Z_2 et X_3 , $d_3 = 0$.

θ_3 : $\theta_3 (X_2 , X_3)$; sur le schéma on a $\theta_3 = - \pi / 2$.

1-c-3-TRANSFORMATION HOMOGENE ASSOCIEE AU REPERE R_3 (dans le cas θ_3 quelconque)

$$A_3 = \text{Rot} (Z_2 , \theta_3) * \text{Trans} (Z_2 , d_3) * \text{Trans} (X_3 , a_3) * \text{Rot} (X_3 , \alpha_3) = T^3_2$$

$$A_3 = T^3_2 = \text{Rot} (Z_2 , \theta_3) * \text{Trans} (X_3 , a_3) .$$

$$\Rightarrow A_3 = \begin{bmatrix} C_3 & -S_3 & 0 & C_3 L_3 \\ S_3 & C_3 & 0 & S_3 L_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

1-D-POSITIONNEMENT DE L'EXTREMITE DE LA PINCE : (MATRICE E)

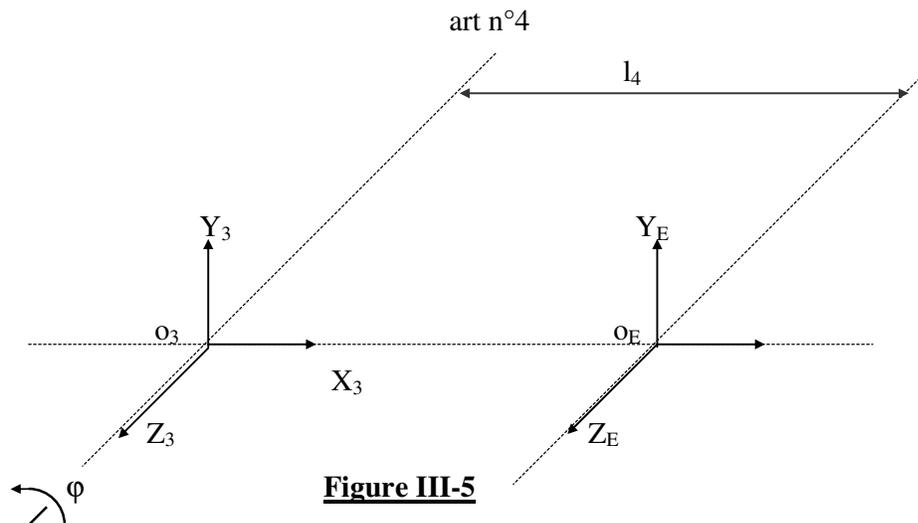


Figure III-5

La transformation E donne la position de l'extrémité de la pince par rapport au dernier segment du manipulateur, on appellera ce point, point objet $P_{ob} \equiv O_E$.

On passe de O_3 à P_{ob} par une transformation E telle que :

$$E = \text{Rot}(Z_3, \varphi) * \text{Trans}(X_3, l_4)$$

$$\Rightarrow E = \begin{bmatrix} C\varphi & -S\varphi & 0 & l_4 C\varphi \\ S\varphi & C\varphi & 0 & l_4 S\varphi \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

1-E-DETERMINATION DE L'ANGLE φ

Le couplage mécanique qui relie les deux segments S_3 et S_2 à l'effecteur, a pour effet de maintenir la position horizontale de la pince (effecteur) par rapport au plan (X_{UN}, Y_{UN}) de l'univers, cela engendre un angle de compensation entre le repère R_3 du dernier segment et le repère R_E de l'effecteur $\varphi = (X_3, X_E)$. On peut tout de suite voir d'après le schéma de la figure I-1, que θ_1 n'a aucun effet sur l'angle φ , alors que φ compense $\theta_2 + \theta_3$. Par conséquent $\varphi = -(\theta_2 + \theta_3)$. La démonstration se fera dans ce qui suit.

*Cas a) θ_2 et θ_3 au repos, θ_1 quelconque: figure III-6.a : Position de repos

$$\begin{aligned} \theta_3 &= (X_2, X_3) = -\pi/2 \\ \theta_2 &= (X_1, X_2) = \pi/2 \\ \varphi &= (X_3, X_E) = 0 \\ \theta_2 + \theta_3 &= \pi/2 - \pi/2 = 0 \\ \varphi &= 0, \text{ donc } \varphi = -(\theta_2 + \theta_3) \end{aligned}$$

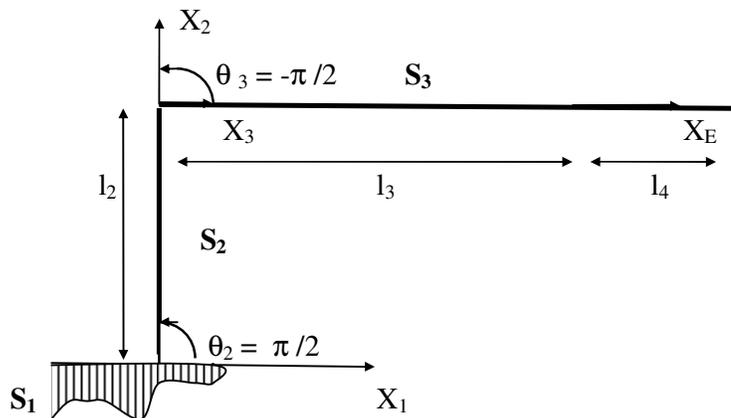


Figure III-6.a

*Cas b) θ_2 au repos, θ_1 et θ_3 quelconques : figure III-6.b

$$\theta_3 = (X_2, X_3) = -\pi/2 - \lambda$$

$$\theta_2 = (X_1, X_2) = \pi/2$$

$$\varphi = (X_3, X_E) = \lambda \text{ (angles ayant un côté commun, les 2 autres étant parallèles)}$$

$$\theta_2 + \theta_3 = \pi/2 - \pi/2 - \lambda = -\lambda$$

$$\Rightarrow \varphi = -(\theta_2 + \theta_3)$$

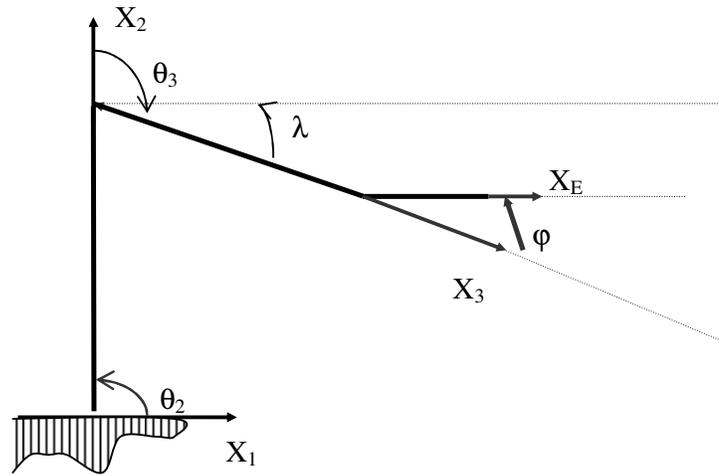


Figure III-6.b

Cas c) θ_3 au repos, θ_1 et θ_2 quelconques : figure III-6.c

$$\theta_3 = (X_2, X_3) = -\pi/2$$

$$\theta_2 = (X_1, X_2) = \pi/2 - \lambda$$

$$\varphi = (X_3, X_E) = \lambda$$

$$\theta_2 + \theta_3 = -\pi/2 + \pi/2 - \lambda = -\lambda$$

$$\Rightarrow \varphi = -(\theta_2 + \theta_3)$$

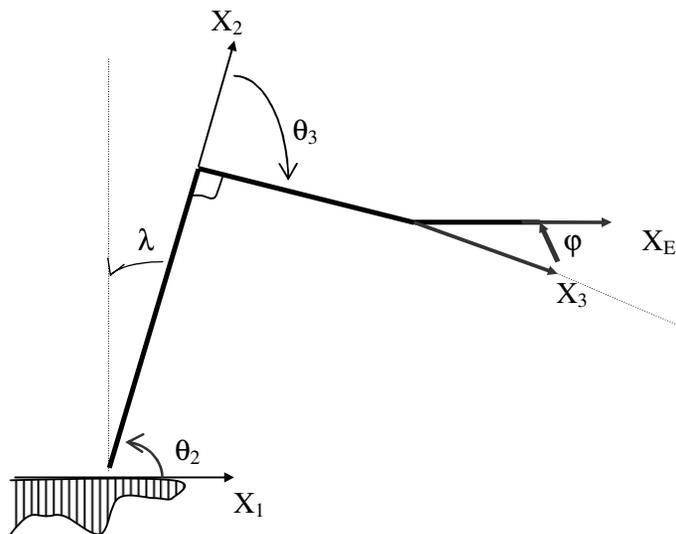


Figure III-6.c

Cas d) θ_1 et θ_2 et θ_3 quelconques : figure III-6.d

$$\theta_3 = (X_2, X_3) = -\pi/2 - \alpha$$

$$\theta_2 = (X_1, X_2) = \pi/2 - \lambda$$

$\varphi = (X_3, X_E) = \varphi_1 + \varphi_2 = \lambda + \alpha$ ($\varphi_1 = \alpha$: angles ayant un côté commun, les 2 autres étant parallèles; $\varphi_2 = \lambda$: angles ayant leurs côtés perpendiculaires)

$$\theta_2 + \theta_3 = -\lambda - \alpha = -\varphi$$

$$\Rightarrow \varphi = -(\theta_2 + \theta_3)$$

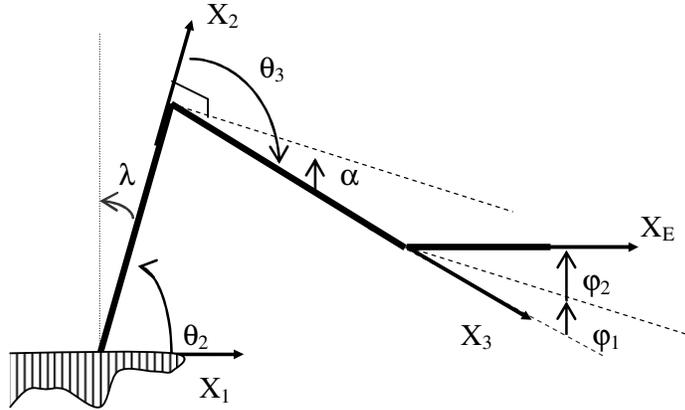


Figure III-6.d

On a démontré donc que $\varphi = -(\theta_2 + \theta_3)$

$$\Rightarrow C_\varphi = C_{23} = \cos(\theta_2 + \theta_3)$$

$$S_\varphi = -S_{23} = -\sin(\theta_2 + \theta_3)$$

$$E = \begin{bmatrix} C_{23} & S_{23} & 0 & l_4 C_{23} \\ -S_{23} & C_{23} & 0 & -l_4 S_{23} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

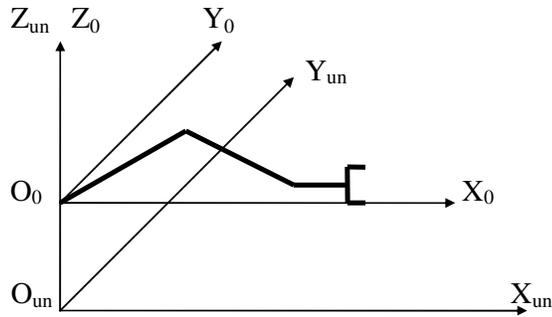


Figure III-7

Si on considère le repère de définition du poste de travail (que l'on appellera dans la suite repère de l'univers R_{UN}), comme étant fixé à la base du manipulateur, alors la transformation Z qui définit la position et l'orientation du repère de référence du manipulateur R_0 , est donnée par une translation d'axe Z_{UN} de longueur l_1 :

$$Z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

1-G-MATRICE DE TRANSFORMATION DU MANIPULATEUR : T

1G-1- Définition et graphe

Cette transformation permet le passage du repère R_0 au repère R_3 . Elle est donnée

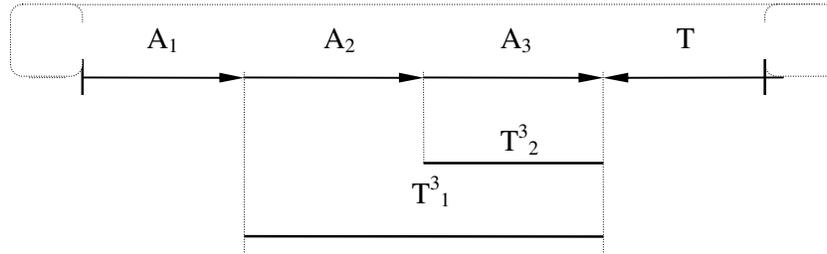
$$\text{par : } T = T_0^3 = T_0^1 \cdot T_1^2 \cdot T_2^3 = \prod_{i=1}^3 A_i$$

$$T = n_y \begin{bmatrix} n_x & s_x & a_x & H_x \\ s_y & a_y & H_y & \\ n_z & s_z & a_z & H_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

La dernière colonne de cette matrice donne les coordonnées de l'origine du dernier repère (R_3) que l'on appellera H dans le repère de référence.

Quant aux trois premières colonnes, elles donnent les directions des trois vecteurs unitaires (U_x, U_y, U_z) du repère R_3 dans le repère R_0 .

Le graphe associé à l'équation de transformation $T = \prod_{i=1}^3 A_i$ est le suivant :



Connaissant les différentes matrices A_i , calculons à présent les différentes matrices de transformation dérivées T^j_3 avec $j = 1$ à 2 (pour $j = 0$ on a $T^0_3 = T$).

$$A_2 \cdot A_3 = T^3_1 = A^{-1}_1 \cdot T = \begin{bmatrix} C_2 & -S_2 & 0 & C_2 l_2 \\ S_2 & C_2 & 0 & S_2 l_2 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} C_3 & -S_3 & 0 & l_3 C_3 \\ S_3 & C_3 & 0 & l_3 S_3 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_2 \cdot A_3 = \begin{bmatrix} C_{23} & -S_{23} & 0 & l_2 C_2 + l_3 C_{23} \\ S_{23} & +C_{23} & 0 & l_2 S_2 + l_3 S_{23} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$A_1 \cdot A_2 = T \cdot A^{-1}_3 = \begin{bmatrix} C_1 C_2 & -C_1 S_2 & S_1 & C_1 C_2 l_2 \\ S_1 C_2 & -S_1 S_2 & -C_1 & S_1 S_2 l_2 \\ S_2 & C_2 & 0 & S_2 l_2 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\begin{bmatrix} C_1 & 0 & S_1 & 0 \\ S_1 & 0 & -C_1 & 0 \\ & & & -81- \end{bmatrix} \begin{bmatrix} C_{23} & -S_{23} & 0 & l_2 C_2 + l_3 C_{23} \\ 0 & S_{23} & C_{23} & 0 \\ & & & l_2 S_2 \end{bmatrix} +$$

$l_3 S_{23}$

$$T = T_0^3 = A_1 \cdot (A_2 \cdot A_3) = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\Rightarrow T = \begin{bmatrix} C_1 C_{23} & -C_1 S_{23} & S_1 & C_1 (l_3 C_{23} + l_2 C_2) \\ S_1 C_{23} & -S_1 S_{23} & -C_1 & S_1 (l_3 C_{23} + l_2 C_2) \\ S_{23} & C_{23} & 0 & l_2 S_2 + l_3 S_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T.E = \begin{bmatrix} C_1 & 0 & S_1 & C_1 (l_4 + l_3 C_{23} + l_2 C_2) \\ S_1 & 0 & -C_1 & S_1 (l_4 + l_3 C_{23} + l_2 C_2) \\ 0 & 1 & 0 & l_2 S_2 + l_3 S_{23} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\text{Pos} = Z.T.E = \begin{bmatrix} C_1 & 0 & S_1 & C_1 (l_4 + l_3 C_{23} + l_2 C_2) \\ S_1 & 0 & -C_1 & S_1 (l_4 + l_3 C_{23} + l_2 C_2) \\ 0 & 1 & 0 & l_2 S_2 + l_3 S_{23} + l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Calculons également les inverses des matrices A_i dont on aura besoin dans la suite.

$$[A_1]^{-1} = \begin{bmatrix} C_1 & S_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ S_1 & -C_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad [A_2]^{-1} = \begin{bmatrix} C_2 & S_2 & 0 & -l_2 \\ -S_2 & C_2 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$[A_3]^{-1} = \begin{bmatrix} C_3 & S_3 & 0 & -l_3 \\ -S_3 & C_3 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

1-G-2- Calcul de la matrice T à partir des données

Le point à atteindre P étant défini par sa transformation Pos, on a alors l'équation :

$$Z * T * E = Pos$$

$$\Rightarrow T = Z^{-1} * Pos * E^{-1}$$

$$Z = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & l_1 \\ 0 & 0 & 0 & 0 \end{bmatrix} ; \quad E = \begin{bmatrix} C_{23} & S_{23} & 0 & l_4 C_{23} \\ -S_{23} & C_{23} & 0 & l_4 S_{23} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Pos = \begin{bmatrix} Pos_{11} & Pos_{12} & Pos_{13} & Px \\ Pos_{21} & Pos_{22} & Pos_{23} & Py \\ Pos_{31} & Pos_{32} & Pos_{33} & Pz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$Z^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -l_1 \\ 0 & 0 & 0 & 0 \end{bmatrix} ; \quad E^{-1} = \begin{bmatrix} C_{23} & -S_{23} & 0 & -l_4 \\ S_{23} & C_{23} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$T = Z^{-1} \cdot Pos \cdot E^{-1} = \begin{bmatrix} Pos_{11} C_{23} + Pos_{12} S_{23} & -Pos_{11} S_{23} + Pos_{12} C_{23} & Pos_{13} & Px - Pos_{11} l_4 \\ Pos_{21} C_{23} + Pos_{22} S_{23} & -Pos_{21} S_{23} + Pos_{22} C_{23} & Pos_{23} & Py - Pos_{21} l_4 \\ Pos_{31} C_{23} + Pos_{32} S_{23} & -Pos_{31} S_{23} + Pos_{32} C_{23} & Pos_{33} & Pz - Pos_{31} l_4 - l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

d'autre part :

$$T = \begin{bmatrix} nx & sx & ax & Hx \\ ny & sy & ay & Hy \\ nz & sz & az & Hz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Par identification, on aura :



$$nx = \text{Pos}_{11} C_{23} + \text{Pos}_{12} S_{23}$$

$$ny = \text{Pos}_{21} C_{23} + \text{Pos}_{22} S_{23}$$

$$nz = \text{Pos}_{31} C_{23} + \text{Pos}_{32} S_{23}$$

$$sx = \text{Pos}_{12} C_{23} - \text{Pos}_{11} S_{23}$$

$$sy = \text{Pos}_{22} C_{23} - \text{Pos}_{21} S_{23}$$

$$sz = \text{Pos}_{32} C_{23} - \text{Pos}_{31} S_{23}$$

$$ax = \text{Pos}_{13}$$

$$ay = \text{Pos}_{23}$$

$$az = \text{Pos}_{33}$$

$$Hx = Px - \text{Pos}_{11} l_4$$

$$Hy = Py - \text{Pos}_{21} l_4$$

$$Hz = Pz - \text{Pos}_{31} l_4 - l_1$$

Et on a calculé aussi :

$$\text{Pos} = Z \cdot T \cdot E = \begin{bmatrix} C_1 & 0 & S_1 & C_1 (l_4 + l_3 C_{23} + l_2 C_2) \\ S_1 & 0 & -C_1 & S_1 (l_4 + l_3 C_{23} + l_2 C_2) \\ 0 & 1 & 0 & l_2 S_2 + l_3 S_{23} + l_1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\Rightarrow \begin{array}{lll} \text{Pos}_{11} = C_1 & \text{Pos}_{21} = S_1 & \text{Pos}_{31} = 0 \\ \text{Pos}_{12} = 0 & \text{Pos}_{22} = 0 & \text{Pos}_{32} = 1 \\ \text{Pos}_{13} = S_1 & \text{Pos}_{23} = -C_1 & \text{Pos}_{33} = 0 \\ Hx = Px - C_1 l_4 & & \\ Hy = Py - S_1 l_4 & & \\ Hz = Pz - l_1 & & \end{array}$$

III-2-RESOLUTION DU PROBLEME GEOMETRIQUE INVERSE

Pour calculer les différentes variables θ_i ($i = 1 \text{ à } 3$), on va se servir des différentes équations de transformations dérivées de l'équation : $T = A_1 * A_2 * A_3$.

2-a . Calcul de θ_1

Nous avons calculé

$$T^3_1 = A_2 . A_3 = A^{-1}_1 . T = \begin{bmatrix} C_{23} & -S_{23} & 0 & l_2 C_2 + l_3 C_{23} \\ S_{23} & C_{23} & 0 & l_2 S_2 + l_3 S_{23} \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

D'autre part :

$$A_2 . A_3 = A^{-1}_1 . T = \begin{bmatrix} C_1 & S_1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ S_1 & -C_1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} nx & sx & ax & Hx \\ ny & sy & ay & Hy \\ nz & sz & az & Hz \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} C_1 nx + S_1 ny & C_1 sx + S_1 sy & C_1 ax + S_1 ay & C_1 Hx + S_1 Hy \\ nz & sz & az & Hz \\ S_1 nx - C_1 ny & S_1 sx - C_1 sy & S_1 ax - C_1 ay & S_1 Hx - C_1 Hy \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Par identification, on aura :

$$C_{23} = C_1 nx + S_1 ny \quad (1) \qquad C_1 sx + S_1 sy = - S_{23} \quad (7)$$

$$S_{23} = nz \quad (2) \qquad sz = C_{23} \quad (8)$$

$$S_1 nx - C_1 ny = 0 \quad (3) \qquad S_1 sx - C_1 sy = 0 \quad (9)$$

$$C_1 ax + S_1 ay = 0 \quad (4) \qquad C_1 Hx + S_1 Hy = l_3 C_{23} + C_2 l_2 \quad (10)$$

$$az = 0 \quad (5) \qquad Hz = l_3 S_{23} + l_2 S_2 \quad (11)$$

$$S_1 ax - C_1 ay = 1 \quad (6) \qquad S_1 Hx - C_1 Hy = 0 \quad (12)$$

(12) ==> $\text{tg } \theta_1 = Hy / Hx$

On a aussi :

$$Hx = Px - C_1 l_4 \quad (13)$$

$$Hy = Py - S_1 l_4 \quad (14)$$

$$Hz = Pz - l_1 \quad (15)$$

En remplaçant dans (12) ==> $S_1 (Px - C_1 l_4) - C_1 (Py - S_1 l_4) = 0$

$$\Leftrightarrow S_1 P_x - C_1 P_y = 0$$

$$\Leftrightarrow \boxed{\text{tg } \theta_1 = P_y / P_x}$$

(3) et (4) et (9) \Rightarrow

$$\boxed{\text{tg } \theta_1 = -a_x / a_y = n_y / n_x = s_y / s_x}$$

Ce qui donne deux valeurs possibles pour θ_1 : θ_1 et $\theta_1 + \pi$

2-b- Calcul de θ_3

Réécrivons les équations (10) et (11) :

$$C_1 H_x + S_1 H_y = l_3 C_{23} + C_2 l_2$$

$$H_z = l_3 S_{23} + l_2 S_2$$

D'autre part :

$$H_x = P_x - C_1 l_4 \quad (13)$$

$$H_y = P_y - S_1 l_4 \quad (14)$$

$$H_z = P_z - l_1 \quad (15)$$

Posons :

$$H'_x = C_1 H_x + S_1 H_y$$

$$H'^2_x + H'^2_z = l_2^2 C_2^2 + l_2^2 S_2^2 + l_3^2 C_{23}^2 + l_3^2 S_{23}^2 + 2 l_2 l_3 (C_{23} C_2 + S_{23} S_2)$$

$$\Rightarrow H'^2_x + H'^2_z = l_2^2 + l_3^2 + 2 l_2 l_3 C_3$$

D'où :

$$\boxed{C_3 = \frac{H'^2_x + H'^2_z - (l_2^2 + l_3^2)}{2 l_2 l_3} = \cos \theta_3}$$

Bien que ce calcul ne fournisse aucune information supplémentaire, au lieu de calculer θ_3 à partir de la fonction arccosinus, il est préférable d'utiliser la même fonction arctangente (y,x) pour tous les angles. On calculera donc :

$$\boxed{S_3 = \sin \theta_3 = \pm \sqrt{1 - (\cos \theta_3)^2}}$$

$$\boxed{\theta_3 = \text{Arctg} (\sin \theta_3 , \cos \theta_3)}$$

On remarque que l'on a deux valeurs possibles pour θ_3 : θ_3 et $\theta_3 + k\pi$.

On a $\theta_3 = (X_2, X_3)$; on remarque sur les figures III-4, III-7, ainsi que I-1, que par construction le sens de (X_2, X_3) est opposé au sens conventionnel (trigonométrique) choisi pour cette étude, d'où $\theta_3 < 0$ quelle que soit la position du manipulateur.

Donc les valeurs possibles pour θ_3 sont θ_3 et $\theta_3 - \pi$.

2-c- Calcul de θ_2

Reprenons les expressions de $H'x$ et H_z utilisées pour le calcul de θ_3 .

$$H'x = l_2 C_2 + l_3 C_{23}$$

$$H'x = C_2 (l_2 + l_3 C_3) - l_3 S_3 S_2$$

\Leftrightarrow

$$H_z = l_2 S_2 + l_3 S_{23}$$

$$H_z = S_2 (l_2 + l_3 C_3) + l_3 S_3 C_2$$

$$\Leftrightarrow (l_2 + l_3 C_3) C_2 - l_3 S_3 S_2 = H'x$$

$$l_3 S_3 C_2 + (l_2 + l_3 C_3) S_2 = H_z$$

Nous avons donc affaire à un système de deux équations à deux inconnues : S_2 et C_2 . Sa résolution donne :

$$S_2 = \frac{H_z (l_2 + l_3 C_3) - H'x l_3 S_3}{(l_2 + l_3 C_3)^2 + l_3^2 S_3^2}$$

$$C_2 = \frac{H'x (l_2 + l_3 C_3) + H_z l_3 S_3}{(l_2 + l_3 C_3)^2 + l_3^2 S_3^2}$$

$$\theta_2 = \text{Arctg} \frac{H_z (l_2 + l_3 C_3) - H'x l_3 S_3}{(l_2 + l_3 C_3) H'x + l_3 S_3 H_z}$$

Comme pour θ_3 les numérateurs et dénominateurs de la fonction arctangente donnent respectivement les signes de $\sin \theta_2$ et $\cos \theta_2$.

-III-3-ETUDE DES CONTRAINTES ET AMBIGUITES SUR LES ANGLES

La fonction arctangente donnant l'angle à π près, pour éviter l'indétermination, on utilisera une fonction à deux arguments : $\theta = \text{Arctg} [y , x]$. C'est-à-dire que l'on donnera les valeurs du sinus et du cosinus (en plus de leur rapport), car ce sont leurs signes qui permettront de lever l'indétermination .

a . Angle θ_1 : $0^\circ \leq \theta_1 \leq 240^\circ$

$$\theta_1 = \text{arctg} [P_y , P_x]$$

Les signes de P_y et P_x correspondent à ceux de $\sin \theta$, et $\cos \theta$.

* $\text{tg } \theta_1 \geq 0$

Si $P_x > 0$ et $P_y \geq 0 \implies \text{tg } \theta_1 \geq 0$ et θ_1 appartient au premier quadrant : $0^\circ \leq \theta_1 < 90^\circ$.

Si $P_x < 0$ et $P_y \leq 0 \implies \text{tg } \theta_1 \geq 0$ et θ_1 appartient au quatrième quadrant :

$180^\circ \leq \theta_1 < 270^\circ$, soit $180^\circ \leq \theta_1 \leq 240^\circ$ (valeur dans le domaine atteignable) .

Donc si $\text{tg } \theta_1 \geq 0$, suivant les signes de P_y et P_x , on aura, soit θ_1 (cos $P_x > 0$ et $P_y \geq 0$), soit $\theta_1 + \pi$ (cos $P_x < 0$ et $P_y \leq 0$).

* $\text{tg } \theta_1 < 0$

Si $P_x > 0$ et $P_y < 0 \implies \text{tg } \theta_1 < 0$ et θ_1 appartient au quatrième quadrant :

$270^\circ < \theta_1 < 360^\circ$, ce qui est impossible à atteindre car : $0^\circ \leq \theta_1 < 240^\circ$.

Si $P_x < 0$ et $P_y > 0 \implies \text{tg } \theta_1 < 0$ et θ_1 appartient au deuxième quadrant :

$90^\circ < \theta_1 \leq 240^\circ$, valeur qui est dans le domaine atteignable.

Donc si $\text{tg } \theta_1 < 0$, suivant les signes de P_y et P_x , on aura, soit θ_1 ($P_x < 0$ et $P_y > 0$) , soit on est dans le domaine non atteignable.

Par conséquent, il ne peut y avoir d'ambiguïté pour la valeur de θ_1 . Il suffit de préciser pour chaque calcul de θ_1 , les signes de P_y et P_x .

b . Angle θ_2 : $30^\circ \leq \theta_2 \leq 90^\circ$

$$\theta_2 = \text{arctg} [H_z (l_2 + l_3 C_3) - H'x l_3 S_3 , H'x (l_2 + l_3 C_3) + l_3 S_3 H_z]$$

Etant donné que les deux arguments de $\text{tg } \theta_2$ donnent les signes de $\sin \theta_2$ et $\cos \theta_2$, il ne peut y avoir d'ambiguïté sur l'angle θ_2 , le raisonnement est le même que pour θ_1 .

c . Angle θ_3 : $-120^\circ \leq \theta_3 \leq -60^\circ$

Comme par construction, θ_3 est négatif, on écrira donc pour $\sin \theta_3$:

$\forall \theta_3 \in [- 120^\circ ; - 60^\circ] :$

$$\sin \theta_3 = - \left| \sqrt{1 - \cos^2 \theta_3} \right| < 0 .$$

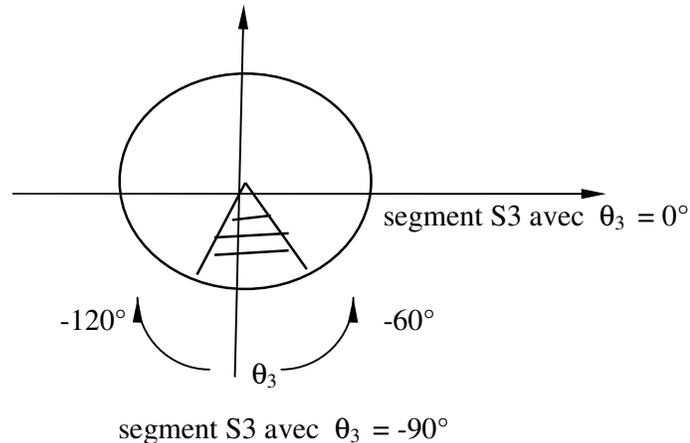
Si $\cos \theta_3 > 0$, donc la tangente sera négative et elle donnera un angle dans le deuxième quadrant. Il faudra donc prendre $\theta_3 = \arctg [\sin \theta_3 ; \cos \theta_3] - \pi$, pour avoir l'angle négatif;

Sinon si $\cos \theta_3 < 0$, la tangente sera positive et elle donnera un angle dans le premier quadrant. Il faudra donc prendre $\theta_3 = \arctg [\sin \theta_3 ; \cos \theta_3] - \pi$, pour avoir l'angle négatif.

Sinon si $\cos \theta_3 = 0$, donc $\sin \theta_3 = -1$, donc $\theta_3 = -\pi$, valeur qui n'appartient pas au domaine atteignable.

Donc $\sin \theta_3$ étant négatif, quel que soit le signe de $\cos \theta_3 \neq 0$, on aura :

$$\theta_3 = \arctg [\sin \theta_3 / \cos \theta_3] - \pi .$$



Remarque :

Pour le calcul d'un angle, parfois le sinus ou le cosinus est supérieur à 1, il faut alors considérer que le point correspondant n'est pas atteignable.

ANN1-IV-DESCRIPTION DU LOGICIEL ROBY3

-IV-1- LES DIFFERENTS MODES DU MENU DE ROBY3

1-1-Généralités

Pour démarrer le logiciel il faut obligatoirement avoir ces 4 fichiers: roby3.exe, roby3.ovr, tache.txt, aide.txt. Pour travailler dans le mode programme il faut d'abord charger une tâche (la fonction chargement se trouve dans l'option outils). Cette tâche doit exister dans le tableau des tâches (on peut utiliser l'option lister_nom_tache pour avoir la liste des tâches qui existent). Une fois que la tâche est chargée, l'utilisateur peut travailler dans le contexte de cette tâche. Il peut simuler, exécuter, ou éditer un programme.

1-2-Edition

L'option édition permet à l'utilisateur d'éditer une tâche ou un programme.

Dans l'édition d'une tâche l'utilisateur peut créer, modifier, ou supprimer une tâche.

La création d'une tâche revient à créer ses 4 fichiers sur disque:

- 1)nom_tache1.txt:contient les points de la tâche
- 2)nom_tache2.trj:contient les trajectoires de la tâche
- 3)nom_tache3.txt:contient les noms des programmes de la tâche
- 4)nom_tache4.txt:contient les noms des programmes déjà simulés

Dans la modification d'une tâche l'utilisateur peut soit supprimer un programme de cette tâche, soit ajouter (associer) un programme à cette tâche. Dans les deux cas l'utilisateur doit donner le nom de la tâche à modifier.

Dans la suppression d'une tâche la tâche indiquée sera complètement écrasée, c'est à dire tout ce qui a rapport avec la tâche est supprimé: les programmes sources avec leurs résultats de simulation ainsi que le code exécutable, soit les 4 fichiers qui définissent la tâche.

La fonction [édition_prg_source] permet l'édition d'un programme source en faisant appel à l'éditeur du DOS. Après l'édition du programme l'utilisateur peut simuler et/ou exécuter le programme édité.

1-3-Simulation

Dans ce mode l'utilisateur peut travailler soit en mode pas_à_pas, soit dans le mode global. Dans le mode simulation l'utilisateur peut lister le programme source sur lequel il travaille (programme en cours), comme il peut lister ses résultats de simulation. L'utilisateur peut aussi charger un autre programme pour le simuler. Ce programme devient alors le programme en cours.

1-4-Exécution

L'utilisateur a le choix soit de travailler dans le contexte de la tâche, soit hors du contexte de la tâche. Dans ce deuxième cas, l'utilisateur n'a pas besoin de charger de tâche.

Dans le cas où l'utilisateur a choisi de travailler dans le contexte d'une tâche, il faut qu'une tâche soit en cours.

1-5-Outils

Dans l'option outils l'utilisateur peut:

1-5-a-SAUVEGARDER

Dans cette option il peut sauver soit le programme source, soit le contexte de la tâche en cours. La sauvegarde du contexte de la tâche revient à sauvegarder sur disque les tableaux suivants: tableau des points, tableau des trajectoires, tableau des noms des programmes, tableau des programmes déjà simulés; ainsi que tout ce qui concerne le programme en cours (résultats de simulation + tableau des incréments à transmettre pour la commande des déplacements).

1-5-b-CHARGER

Dans cette option il peut soit charger une tâche (celle ci devient la tâche en cours), soit charger un programme source (ce programme devient le programme en cours).

Charger une tâche c'est charger les tableaux suivants:

- 1) le tableau des points: tab_point à partir du fichier nom_tache1.txt
- 2) le tableau des trajectoire [tab_trajectoire] à partir du fichier nom_tache2.trj
- 3) le tableau des nom de programme associés à la tâche [tab_nom_prg_text] à partir du fichier nom_tache3.txt
- 4) le tableau des programmes déjà simulés [tab_prg_deja_sim] à partir du fichier nom_tache4.txt.

1-5-c-LISTER

Dans cette option l'utilisateur peut lister un programme, les noms des tâches qui existent, le contenu de la tâche en cours.

- 1) Dans le listing du programme source l'utilisateur peut lister n'importe quel programme source du disque (la condition sur l'existence d'une tâche en cours n'est pas prise en considération);
- 2) Dans le listing du contenu de la tâche en cours l'utilisateur peut lister les points de la tâche en cours, les noms des trajectoires de cette tâche, les noms des programmes sources associés à cette tâche, les noms des programmes déjà simulés.

1-5-d-INIT LIASON MONITEUR

Cette fonction permet de travailler soit dans le mode réel, soit dans le mode local, ou bien de basculer d'un mode à un autre.

-IV-2-INSTRUCTIONS de ROBY3

pointea

L'exécution de cette instruction provoque le déplacement de l'extrémité de l'effecteur vers une nouvelle position, en fonction des valeurs des paramètres passés à l'instruction. Pour cette instruction les paramètres représentent les coordonnées articulaires du point qu'on veut atteindre.

Syntaxe de l'instruction: `pointea(t1,t2,t3)`

exemple: `pointea(0,0,-10)`:

Quel que soit l'état précédent du robot, après l'exécution de cette instruction, l'extrémité de l'effecteur sera à

la position $t1=0$ $t2=0$ $t3=-10$

pointec

L'exécution de cette instruction provoque le déplacement de l'extrémité de l'effecteur vers une nouvelle position, en fonction des valeurs des paramètres passés à l'instruction. Pour cette instruction les paramètres représentent les coordonnées cartésiennes du point qu'on veut atteindre.

Syntaxe de l'instruction: `pointec(x,y,z)`

exemples: `pointec(30,10, pointec(30,10,25)`

pointen

Cette instruction opère sur un nom de point. Son exécution déplace l'extrémité de l'effecteur vers les coordonnées de ce point, si ce dernier existe déjà dans le tableau {tab_point}.

Syntaxe de l'instruction: `pointen(nom_point)`

où 'nom_point' est une chaîne de caractères (max 10) débutant obligatoirement par une lettre.

exemple: `pointen(cam)` ;

'cam' est un nom de point prédéfini;

pointex

Instruction entraînant un déplacement axial de l'extrémité de l'effecteur, c_a_d le mouvement de l'effecteur se fait suivant une ligne parallèle à l'axe 'X'.

Syntaxe de l'instruction: `pointex(val)` où val est une valeur réelle

pointey

Instruction entraînant un déplacement axial de l'extrémité de l'effecteur, c'est à dire le mouvement de l'effecteur se fait suivant une ligne parallèle à l'axe 'Y'.

Syntaxe de l'instruction: *pointey*(val) où val est une valeur réelle

pointez

Instruction entraînant un déplacement axial de l'extrémité de l'effecteur, c_a_d le mouvement de l'effecteur se fait suivant une ligne parallèle à l'axe 'Z'.

Syntaxe de l'instruction: *pointezx*(val) où val est une valeur réelle;

reference

Cette instruction provoque le déplacement de l'extrémité de l'effecteur vers: (t1=0,t2=0,t3=-35.5), correspondant à (x=21.7,y=0,z=16.4).

defpointa

Cette instruction permet de définir un point par le biais de ses coordonnées articulaires.

Syntaxe de l'instruction: *defpointa*(nom_point,t1,t2,t3)

où 'nom_point' est une chaîne de caractères, et t1,t2,t3 sont des réels.

defpointc

Cette instruction permet de définir un point par le biais de ses coordonnées cartésiennes.

Syntaxe de l'instruction: : *defpointc*(nom_point,x,y,z)

où nom_point est une chaîne de caractères, et x,y,z sont des réels.

deftraject

Cette instruction permet la définition d'une nouvelle trajectoire

Syntaxe de l'instruction: *deftraject*(nom_traject,pt1,pt2,pt3...ptn)

où: pt1,pt2....ptn sont des noms de points existants dans le tableau des points,

n est le nombre de points (le max autorisé pour une trajectoire est 20)

exetraject

Cette instruction permet d'exécuter une trajectoire qui est prédéfinie.

Syntaxe de l'instruction: : *exetraject*(nom_traject)

ouvrir

Cette instruction permet de contrôler l'ouverture de la pince; on dispose de 3 positions : ouvrir(1), ouvrir(2), ouvrir(3)

exemple: ouvrir(1) correspond à une ouverture totale de la pince

fermer

Cette instruction permet de contrôler la fermeture de la pince

exemple:

stop

C'est une instruction de pause , c'est à dire lorsque le logiciel rencontre cette instruction au cours de l'exécution du programme, il suspend temporairement l'exécution jusqu'à appui sur une touche du clavier.

Syntaxe de l'instruction: :stop

fin

C'est une instruction qui indique la fin de programme. Sa présence est obligatoire dans un programme, sinon il ne peut être ni exécuté ni simulé.

Syntaxe de l'instruction: : fin

tempo

C'est une instruction de temporisation. Par l'exécution de cette instruction on peut suspendre l'exécution d'un programme pendant une durée déterminée. 3 bases de temps sont disponibles [1:1000ms][2:100ms][3:10ms].

exemple: tempo(1,20) 1est la base de temps, 20est la durée, donc c'est une temporisation de $1000ms * 20 = 20000ms = 20secondes$.

;

Chaque fois que ce caractère est présent au début d'une ligne, cette ligne est considérée comme commentaire et ignorée au moment de la simulation ou de l'exécution.

tache

C'est une instruction obligatoire qui marque le début du programme. Après sortie de l'édition d'un programme source, elle est décalée de la 1ère à la 5ème ligne, en vue des traitements ultérieurs. Son absence à la ligne 5 indiquera au cours des traitements, qu'il s'agit d'un fichier texte standard (extension txt) et non d'un fichier de ROBY3.

boucle

C'est un bloc d'instructions répété n fois. Il est compris entre les caractères [b] et [*n] , où n désigne le nombre de répétitions.

IV-3-CODES ET MESSAGES D'ERREUR

1

La trajectoire que vous voulez exécuter n'existe pas.

2

Instruction erronée manque une parenthèse "(".

3

Le paramètre de l'instruction "ouvrir" est incorrect.

4

La base de temps et la durée ne sont pas séparées par des virgules.

5

Durée plus grande que la durée autorisée.

6

La base de temps est incorrecte.

7

La valeur pour la commande axiale doit être réelle.

8

Au moins un des arguments de l'instruction n'est pas réel.

9

Le nom est incorrect, il faut qu'il commence par une lettre.

10

Un point au moins de la trajectoire n'existe pas.

11

Le nom de la trajectoire à définir existe déjà, voulez vous le remplacer?

12

Un paramètre au moins de l'instruction "defpoint" n'existe pas.

13

Un nom au moins de l'instruction "deftraject" est incorrect.

14

Le nom de l'instruction "defpoint" est incorrect, il doit commencer par une lettre.

15

Le point à définir n'appartient pas au DomAtt cartésien.

16

Le point à définir n'appartient pas au DomAtt articulaire.

17

Le point à définir existe déjà, voulez vous le remplacer?(o/n)

18

Le nom de l'instruction "pointen" est incorrect, il doit commencer par une lettre.

19

Instruction commence par "p" mais elle est inconnue ou incorrecte.

20

Instruction commence par "t" mais elle est inconnue ou incorrecte.

21

Instruction commence par "o" mais elle est inconnue ou incorrecte.

22

Instruction commence par "f" mais elle est inconnue ou incorrecte.

23

Instruction commence par "r" mais elle est inconnue ou incorrecte.

24

Instruction commence par "s" mais elle est inconnue ou incorrecte.

25

Instruction commence par "d" mais elle est inconnue ou incorrecte.

26

Instruction commence par "e" mais elle est inconnue ou incorrecte.

27

Il n'y a pas d'instruction qui commence par ce caractère.

28

Point hors DomAtt cartésien pour l'instruction "pointea".

29

Point hors DomAtt articulaire pour l'instruction "pointea".

30

Point hors DomAtt articulaire pour l'instruction "pointec".

31

Point hors DomAtt cartésien pour l'instruction "pointec".

32

Le nom du point spécifié par l'instruction "pointen" n'existe pas.

33

La durée est incorrecte.

34

Vous avez défini une trajectoire sans donner ses points.

35

Ce nom de programme n'appartient pas à la tâche en cours.

36

L'instruction "fin" n'existe pas dans le programme.

37

Le programme contient une ligne erronée.

38

La tâche que vous voulez créer existe déjà.

39

la tâche que vous voulez modifier n'existe pas.

40

Le nom du programme que vous voulez supprimer n'existe pas sur disque.

41

Le nom du programme que vous voulez rajouter existe déjà.

42

Le nom de la tâche que vous voulez supprimer n'existe pas.

43

Il n'y a pas de tâche en cours.

44

Le fichier qui contient le nombre d'incrémentations n'existe pas.

46

Votre programme ne contient pas le mot "tache" dans la 5eme ligne.

47

Le logiciel n'autorise pas les boucles imbriquées.

48

Fin de boucle détectée sans début "b".

49

Vous ne pouvez pas charger un programme s'il n'y a pas une tâche en cours.

52

Le nom de la tâche à charger n'existe pas dans la liste des tâches.

53

Point hors DomAtt pour l'instruction "pointex".

54

Point hors DomAtt pour l'instruction "pointey".

55

Point hors DomAtt pour l'instruction "pointez".

56

Les résultats de simulation sur disque sont perdus, on resimule le programme.

57

Erreur de communication, la liaison (pc_midicom) n'est pas établie.

58

Erreur dans le transfert du fichier Roby3.hex.

59

Vous êtes en mode local, pour exécuter ce programme repassez en mode réel.

60

Le nombre de répétitions d'un bloc d'instructions est compris entre 1 et 8.

63

Déclaration d'une boucle sans la fin de boucle "*n".

-IV-4-EXEMPLES DE PROGRAMMES EN LANGAGE DE ROBY3

PROGRAMME PRG2

```
tache (marius)
;début du programme
defpointa(P1,50, 0, -35)
depointa(P3,90, 0, -35)
defpointc(P5,-20, 18, 16)
deftraject(T1,P3, P1)
deftraject(T2,P1,P3,P5)
;fin des définitions et début des
actions
pointec(21.7,0,16.4)
boucle
;début de la boucle
b
exetraject(T1)
tempo(1,2)
exetraject(T2)
stop
*3
;fin de la boucle répétée 3 fois
pointea(0,0,-35)
fin
```

PROGRAMME PRG1

```
tache (toto)
defpointa(P2,100, 0, -35)
defpointa(P4,125, 0, -35)
defpointc(P6,-25, 0, 16.5)
deftraject(T3,P2,P4,P6)
pointen(P6)
reference
exetraject(T3)
fin
```

- ANNEXE 2-

DESCRIPTION ET MODELISATION DU ROBOT IR52C

- SOMMAIRE -

	PAGE
-ANN2-I- DESCRIPTION DU MANIPULATEUR IR52C	10
-I-1-PARAMETRES STRUCTRAUX	10
-I-2- VOLUME ATTEIGNABLE	10
-ANN2-II- ELABORATION DU MODELE GEOMETRIQUE	
-II-1- TRANSFORMATIONS HOMOGENES A_i ASSOCIEES AUX REPERES R_i	10
-II-2- LA MATRICE « T » DE TRANSFORMATION DU MANIPULATEUR	11
-II-3-DETERMINATION DE LA MATRICE DE POSITION POS ET DU VECTEUR DE COORDONNEES GENERALISEES (X,Y,Z,L,T,R)	11
-ANN2-III- RESOLUTION DU PROBLEME GEOMETRIQUE INVERSE	
-III-1- DETERMINATION DE LA MATRICE T	11
-III-2- CALCUL DES VARIABLES ARTICULAIRES « θ_i »	11
-III-3-ETUDE DE LA RESOLVABILITE (PAR FIXATION DU COUDE AU DESSUS OU AU DESSOUS DU POIGNET)	12

Ann2-I-DESCRIPTION DU MANIPULATEUR

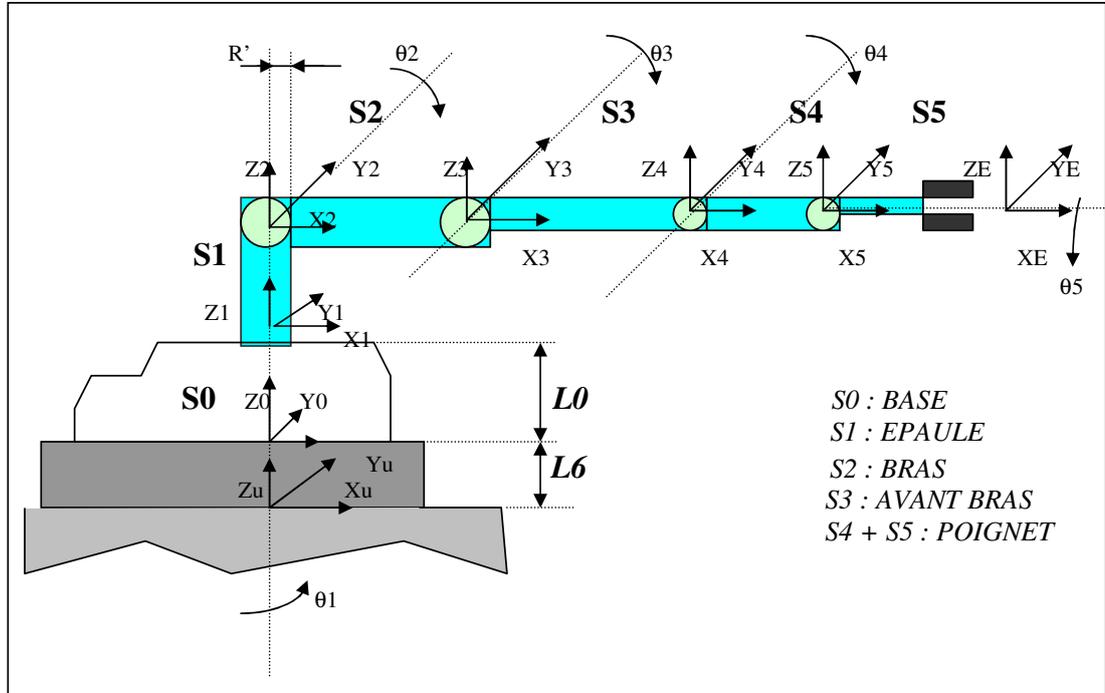


Figure 2.1 : Représentation du manipulateur et de ses repères en position de référence

I-1-Paramètres structuraux

La structure mécanique du robot IR-52-C est constituée de cinq articulations et six segments, et elle peut accepter un effecteur électrique ou pneumatique. Chacun des cinq joints indépendants est mis en mouvement par un moteur à courant continu dont la position est déterminée grâce à un codeur incrémental optique. Les moteurs sont très précis et possèdent une résolution moyenne de 415000 pas par axe.

Pour des raisons de sécurité le mouvement de chaque articulation est limité par des butées mécaniques. La carte contrôleur du SMA (logée dans sa base) utilise un processeur V25 de NEC et peut être pilotée grâce à l'interface série RS232-C d'un PC.

Les différents segments représentés sur la figure 1-2 sont:

segment S0: base du manipulateur, fixe et de longueur $L0 = 180$ mm;

segment S1: épaule de longueur $L1$ (105 mm), mobile par rapport au segment $S0$ par une rotation d'axe Z et d'angle $\theta1$.

segment S2: bras de longueur L2 (200 mm), mobile par rapport au segment S1 par une rotation d'axe Y et d'angle θ_2

segment S3: avant bras de longueur L3 (200 mm), mobile par rapport au segment S2 par une rotation d'axe Y et d'angle θ_3 .

segment S4: poignet de longueur L4 (85 mm), mobile par rapport au segment S3 par une rotation d'axe Y et d'angle θ_4

segment S5: suite du poignet de longueur L5 (96 mm) sur lequel est fixée la pince. Le segment est mobile par rapport au segment S4 par une rotation d'axe X (axe de la pince) d'angle θ_5 .

Support : support métallique rectangulaire de longueur L6 (660 mm) et de largeur 480 mm sur lequel est posé le segment S0. Ce support est lui-même fixé sur une table.

Les domaines de variation des variables articulaires (limitées par des buts mécaniques) sont fixés par construction. Par rapport à la position de référence indiquée par la figure 2.1, ces valeurs correspondent à :

$$\theta_1 \in (-150^\circ, 150^\circ) ; \quad \theta_2 \in (-195^\circ, 15^\circ).$$

$$\theta_3 \in (-110^\circ, 110^\circ) ; \quad \theta_4 \in (-115^\circ, 115^\circ).$$

$$\theta_5 \in (-180^\circ, 180^\circ) ; \quad \theta_{234} \in (-420^\circ, 240^\circ).$$

I-2-Volume atteignable (cf figure 2-2)

Selon la géométrie spécifique du S.M.A (longueurs des segments, plages de variation des moteurs, butées mécaniques sur les articulations) et son environnement (obstacles, objets d'intérêt), l'extrémité de l'effecteur ne peut se déplacer que dans un volume délimité de l'espace appelé *volume atteignable*. Dans ce qui suit on va déterminer les conditions qui le fixent.

-2-1-Condition 1: relative aux longueurs des segments

Rappelons que le module d'un vecteur dans l'espace cartésien est donné par :

$$|\vec{OP}| = \sqrt{P_x^2 + P_y^2 + P_z^2} \quad (1)$$

où : O désigne l'origine du repère, P_x , P_y et P_z les coordonnées du point P de l'extrémité du vecteur.

Soit P un point atteint par l'extrémité de l'effecteur, alors la distance entre ce point l'origine O2 du repère R2 est donnée par la même expression (1), avec P_x, P_y, P_z , désigne cette fois ci les coordonnées de ce point P dans (R2).

D'après la structure géométrique de notre S.M.A, l'effecteur peut atteindre une position extrême lorsque les segments S2, S3, S4 et la pince sont tendus conformément à figure 4. Dans ce cas on écrit $|\vec{op}| = |\vec{op}|_{max}$, tel que, $|\vec{op}|_{max} = L2+L3+L4+L5$.

Ainsi on dit qu'un point p est atteignable si $|\vec{op}| <= |\vec{op}|_{max}$, soit

$$P_x^2+P_y^2+P_z^2 <= (L2+L3+L4+L5)^2 \quad (2)$$

D'après les calculs effectués pour le modèle géométrique (cf paragraphe 2), le rep (R2) se déduit du repère (R0) par une translation de longueur (L0+L1) sur l'axe des Z. De l'expression (2) exprimée par rapport à (R0) devient :

$$P_x^2+P_y^2+ (P_z-(L0+L1))^2 < (L2+L3+L4+L5) \quad (I)$$

où P_x, P_y, P_z désignent les coordonnées du point P, extrémité de l'effecteur par rapport à R

Il est à remarquer que lorsqu'on fait varier θ_1 et θ_2 , l'extrémité de l'effecteur décrit une sphère centrée à l'origine du repère R2 (cf figure 2.2).

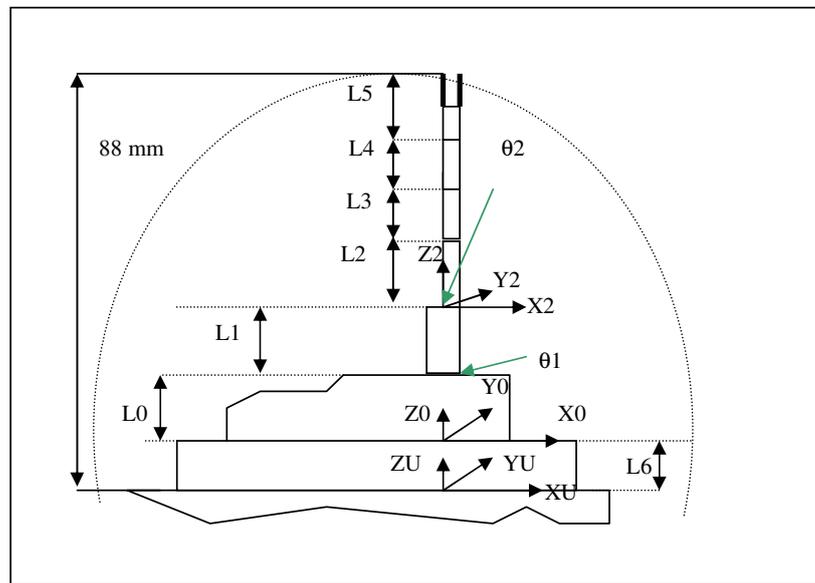


Figure 2.2: Position limite des segments dans la position "initiale"

-2-2- Condition 2: relative au segment S0 (figure2.3)

Pour éviter que l'effecteur touche la base du SMA (segment S0), on doit l'enlever du volume atteignable. D'après la position du repère R0 par rapport au segment S0 (cf figure : on dit qu'un point p n'est pas atteignable si : $-Q < P_x < M$ ET $-P < P_y < N$ ET $P_z < L_0$.

Donc, le point p est dans le volume atteignable si :

$$\left\{ P_x < -Q \text{ OU } P_x > M \right\} \text{ ET } \left\{ P_y < -P \text{ OU } P_y > N \right\} \text{ ET } \left\{ P_z < L_0 \right\}$$

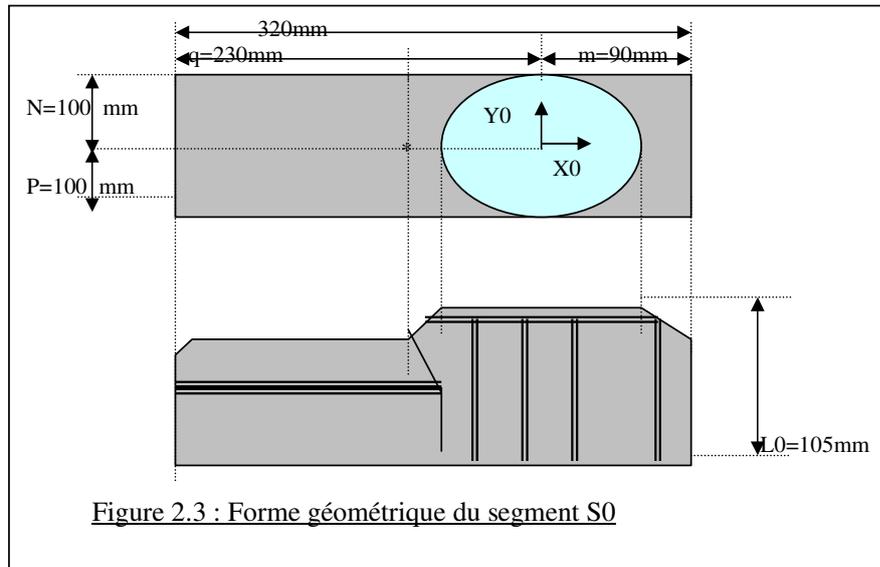
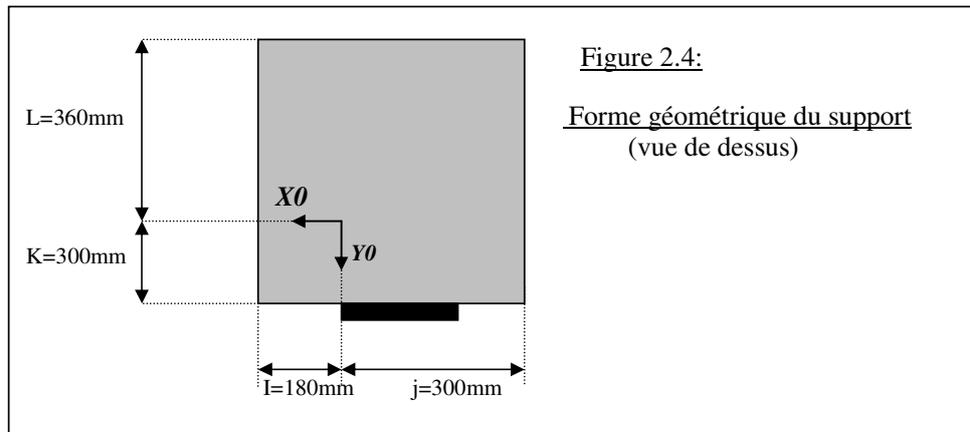


Figure 2.3 : Forme géométrique du segment S0

2-3-Condition 3: relative au support du SMA(cf figure 2.4)

Le robot repose sur un support de dimensions (665mm x 480mm x 15mm). Alors si point cible se trouve à l'intérieur de la surface du support (665mmX480mm), sa coordonnée Pz ne doit pas descendre sous 0mm (par rapport à R0), sinon l'effecteur touchera le support. Compte tenu de la forme géométrique du support et de la position de R0 sur ce dernier (figure 4), on pose la condition 3 de la manière suivante:

le point P est dit atteignable si : $(-j < P_x < i)$ et $(-l < P_y < k)$ et $(P_z > 0\text{mm})$.



-2-4-Condition 4: relative à la table de travail

Le robot et son support reposent sur une table qui est l'espace de travail du S.M. Pour éviter que l'effecteur ne touche la table, tous les points en dessous de la table ne font pas partie du domaine atteignable. Par conséquent cette condition s'exprime par rapport à R0 par :

$$P_z > -L_6 \quad (\text{cf figure 2.1}).$$

-2-5-Condition 5: relative au segment S1

En observant la structure mécanique du robot, on remarque que l'effecteur peut toucher le segment S1, en revanche et grâce aux butées mécaniques, les autres segments seront jamais atteints. Pour cela on intègre le segment S1 dans un cylindre que l'on enlève du volume atteignable. Ce cylindre est défini par le cercle de rayon R' et de hauteur L1 (figure 2.2).

On dit que le point de coordonnées Px, Py, Pz (par rapport à R0) est non atteignable si

$$P_x^2 + P_y^2 < R'^2 \quad \text{et} \quad L_0 < P_z < L_0 + L_1. \quad (V).$$

On dit qu'un point de coordonnées Px, Py, Pz (par rapport à R0) est atteignable si

$$P_x^2 + P_y^2 > R'^2$$

Commentaire [FS1] :

Après avoir délimité le volume atteignable à l'aide de conditions sur les coordonnées cartésiennes, le robot peut se déplacer sans problèmes. Dans ce qui suit on va reprendre les conditions précitées avec les valeurs numériques correspondant aux valeurs physiques.

- 1/ $P_x^2 + P_y^2 + (P_z - 285)^2 < (581)^2$.
- 2/ a/ $(P_x > 90\text{mm})$ et $(P_y < -100\text{mm})$ et $P_z < L_0$.
b/ $(P_x > 90\text{mm})$ et $(P_y > 100\text{mm})$ et $P_z < L_0$.
c/ $(P_x < -230\text{mm})$ et $(P_y > 100\text{mm})$ et $P_z < L_0$.
d/ $(P_x < -230\text{mm})$ et $(P_y < -100\text{mm})$ et $P_z < L_0$.
- 3/ $(-300\text{mm} < P_x < 180\text{mm})$ et $(-360\text{mm} < P_y < 300\text{mm})$ et $P_z > 0$.
- 4/ $P_z > -15\text{mm}$.
- 5/ $P_x^2 + P_y^2 > (60)^2$ et $180 < P_z < 285$.

Remarque

Conformément au paragraphe (1) de la page 108, le repère (R0) se déduit de (Run) une translation d'une distance L_6 sur l'axe des (Z). Par conséquent les conditions sur volume atteignable peuvent être réécrites par rapport à (Run) de la manière suivante:

- 1/ $P_x^2 + P_y^2 + (P_z - (L_6 + L_0 + L_1))^2 < (L_2 + L_3 + L_4 + L_5 +)^2$.
- 2/ a/ $(P_x > 90\text{mm})$ et $(P_y < -100\text{mm})$ et $P_z < (L_0 + L_6)$.
b/ $(P_x > 90\text{mm})$ et $(P_y > 100\text{mm})$ et $P_z < (L_0 + L_6)$.
c/ $(P_x < -230\text{mm})$ et $(P_y > 100\text{mm})$ et $P_z < (L_0 + L_6)$.
d/ $(P_x < -230\text{mm})$ et $(P_y < -100\text{mm})$ et $P_z < (L_0 + L_6)$.
- 3/ $(-300\text{mm} < P_x < 180\text{mm})$ et $(-360\text{mm} < P_y < 300\text{mm})$ et $P_z > L_6$.
- 4/ $P_z > 0\text{mm}$.
- 5/ $P_x^2 + P_y^2 > (60)^2$ et $(L_6 + L_1) < P_z < (L_0 + L_1 + L_6)$.

Ann2-II-ELABORATION DU MODELE GEOMETRIQUE

La modélisation géométrique consiste à exprimer la situation de l'organe termi «effecteur » par rapport au repère (R0) de la base du SMA. Considérons un SMA constitué (n+1) segments (S0 à Sn) où à chaque segment est associé un repère.

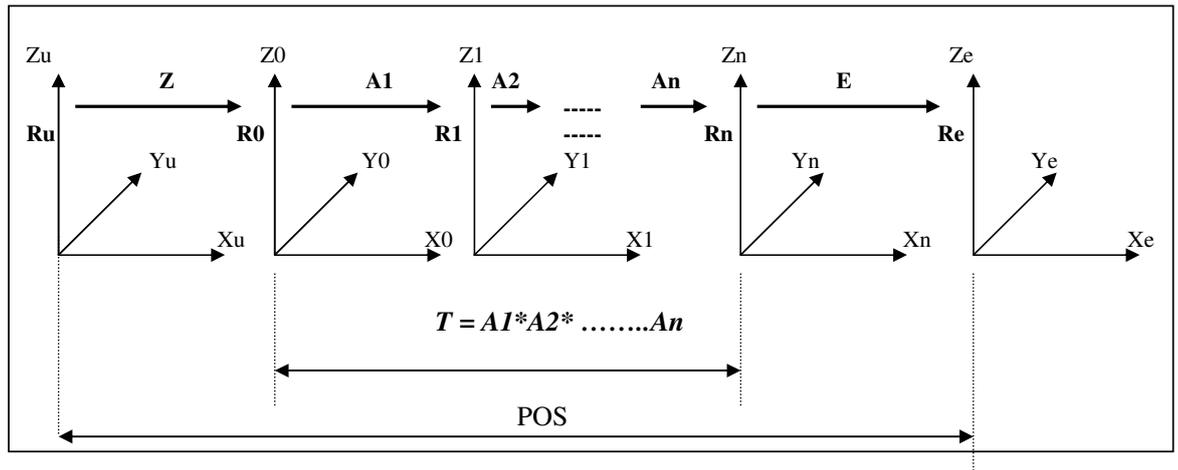


Figure 2.5 : Transposition de coordonnées

On est censé établir la transformation « T » du manipulateur. L'orientation et position de tout point de l'espace atteint par l'extrémité de l'effecteur seront données j rapport au repère de référence de la tâche Ru (repère univers) par l'expression $POS=Z*T^j$ où :

- Z : matrice donnant la position et l'orientation de R0 par rapport à Run.

T : matrice donnant la situation de Rn par rapport à R0.

E : matrice effecteur, elle donne la situation du repère Re attaché à l'effectu (généralement à son extrémité) par rapport au repère du dernier segment (Rn).

Dans le cas de notre SMA nous disposons de 6 segments numérotés (S0 à S5) et d articulations rotoïdes θ_i (i = 1 à 5).

Pour l'association des repères aux différents segments on a utilisé la modélisati géométrique dite "libre" qui est inspirée de la méthode de Denavit-Hartenberg. Elle consist associer à chaque segment du porteur un repère dont l'origine et l'orientation seront chois arbitrairement (dans notre cas nous avons choisi de positionner les repères à la base c segments auxquels ils sont associés).

Pour passer d'un repère à un autre, nous procéderons par une succession transformations géométrique élémentaires (translation ou rotation pure). A chacune de transformations nous associerons une matrice de transformation homogène.

II-1-Transformations homogènes Ai associées aux repères Ri

-1-1 Segment S1

$$A1 = T_0^1 = \text{TRANS}(Z0, L0) * \text{ROT}(Z0, \theta1)$$

$$= \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L0 \\ 0 & 0 & 0 & 1 \end{vmatrix} * \begin{vmatrix} C1 & -S1 & 0 & 0 \\ S1 & C1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$A1 = \begin{vmatrix} C1 & -S1 & 0 & 0 \\ S1 & C1 & 0 & 0 \\ 0 & 0 & 1 & L0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Remarque : Dans tout ce qui va suivre on réutilisera la notation utilisée précédemment :

ROT : Rotation ; TRANS : Translation

Si : Sin (θ_i); Ci : COS (θ_i); Cij : COS ($\theta_i + \theta_j$)

-1-2 - Segment S2

$$A2 = T1 = \text{TRANS}(Z1, L1) * \text{ROT}(V1, \theta2)$$

$$A2 = \begin{vmatrix} C2 & 0 & S2 & 0 \\ 0 & 1 & 0 & 0 \\ -S2 & 0 & C2 & L1 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

-1-3- Segment S3 :

$$A3 = T_2^3 = \text{TRANS}(X2, L2) * \text{ROT}(Y2, \theta3)$$

$$A3 = \begin{vmatrix} C3 & 0 & S3 & L2 \\ 0 & 1 & 0 & 0 \\ -S3 & 0 & C3 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

-1-4-Segment S4

$$A4 = T_3^4 = \text{TRANS}(X3, L3) * \text{ROT}(Y3, \theta4)$$

$$A4 = \begin{vmatrix} C4 & 0 & S4 & L3 \\ 0 & 1 & 0 & 0 \\ -S4 & 0 & C4 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

-1-5-Segment S5

$$A5 = T_4^5 = \text{TRANS}(X4, L4) * \text{ROT}(X4, \theta5)$$

$$A5 = \begin{vmatrix} 1 & 0 & 0 & L4 \\ 0 & C5 & -S5 & 0 \\ 0 & S5 & C5 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

-1-6 – positionnement de l'extrémité de l'effecteur : matrice E

$$E = \text{TRANS}(X5, L5)$$

$$E = \begin{vmatrix} 1 & 0 & 0 & L5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

1-7- Matrice Z :

Le repère univers est choisi de tel manière qu'il y ait le minimum de transformation géométriques pour passer de Ru au repère R0 de la base du SMA. Dans notre cas (Ro) est obtenu par une simple translation de l'axe Zu du repère (Run) d'une longueur (L6).

$$Z = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & L6 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

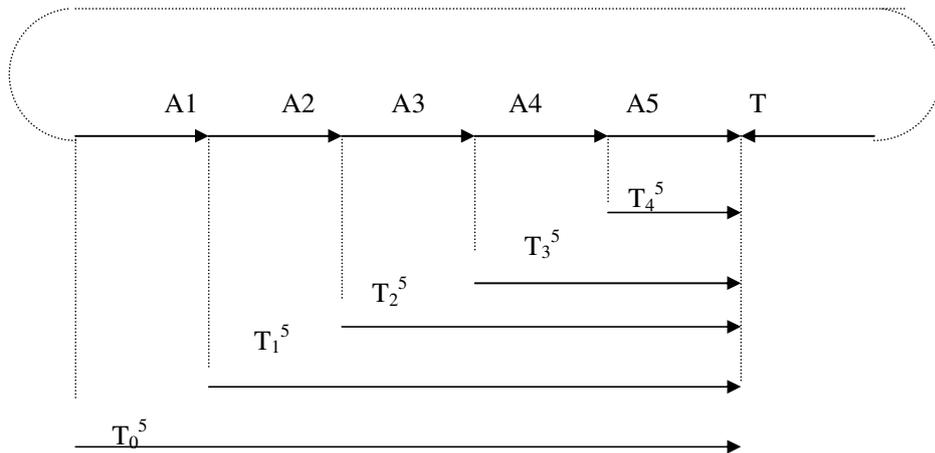
II-2-La matrice « T » de transformation du manipulateur

Elle permet le passage du repère Ro au repère R5, elle est donnée par :

$$T = T_0^5 = T_0^1 * T_1^2 * T_2^3 * T_3^4 * T_4^5 = A1 * A2 * A3 * A4 * A5$$

$$T = \begin{pmatrix} nx & sx & ax & Hx \\ ny & sy & ay & Hy \\ nz & sz & az & Hz \\ 0 & 0 & 0 & 1 \end{pmatrix} \dots(I)$$

Le graphe associé à l'équation de transformation $T = \prod_{i=1}^n A_i$ est le suivant:



Connaissant les différentes matrices Ai, on peut donc procéder au calcul de la matrice T :

$$T = \prod_{i=1}^5 A_i = \dots(II)$$

$$\begin{pmatrix} C1C234 & -S1C5 + C1S5S234 & S1S5+C1C5 S234 & C1[L4 C234 + L3C23 + L2C2] \\ S1C234 & C1C5 + S1S5S234 & -C1S5 + S1C5 S234 & S1[L4C234 + L3C23 + L2C2] \\ - S234 & S5C234 & C5 C234 & [(L1 + L0) - L4S234 - L3 S23- L2 S2] \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

II-3-Détermination de la matrice de position POS

et du vecteur de coordonnées généralisées (x,y,z,l,t,r)

POS est la matrice qui donne la position de l'extrémité de l'effecteur par rapport repère univers. Elle est égale au produit $Z * T * E$.

$$POS = \begin{bmatrix} C1 C234 & -S1C5 + C1 S5 S234 & S1 S5 + C1 C5 S234 & Px \\ S1 C234 & C1C5 + S1 S5 S234 & C1 S5 + S1 C5 S234 & Py \\ -S234 & S5 C234 & C5 C234 & Pz \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (A)$$

$$\text{avec : } \begin{cases} Px = C1 * [(L5 + L4)* C234 + L3* C23 + L2* C2] \\ Py = S1(L5 + L4)* C234 + L3 *C23 + L2 *C2 \\ Pz = (L1 + L6 + L0) - (L5 + L4)* S234 - L3* S23 - L2* S2 \end{cases}$$

Connaissant la matrice POS il reste à déterminer le vecteur de coordonnées généralisées A (Px, Py, Pz, l, t, r), donnant les coordonnées (x, y, z) et l'orientation (l, t, r) point à atteindre.

Commençons par exprimer la matrice POS en fonction des coordonnées généralisé On peut écrire la matrice POS comme une succession de transformations géométriques, donc sous la forme d'un produit de matrices homogènes:

POS = trans (X,Px) * trans (Y, Py) * trans (Z, Pz) * rot (X, l) * rot (Y, t) * rot (Z, r), alors

$$POS = \begin{bmatrix} 1 & 0 & 0 & Px & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & Ct & 0 & St & 0 & Cr & -Sr & 0 \\ 0 & 1 & 0 & 0 & 0 & 1 & 0 & Py & 0 & 1 & 0 & 0 & 0 & Cl & -Sl & 0 & 0 & 1 & 0 & 0 & Sr & Cr & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & Pz & 0 & Sl & Cl & 0 & -St & 0 & Ct & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \end{bmatrix}$$

$$POS = \begin{bmatrix} Ct .Cr & -Ct .Sr & St & px \\ Sl.St.Cr + Cl.Sr & -Sl.St.Sr + Cl .Cr & -Sl.Ct & py \\ -Cl.St.Cr + Sl.Sr & Cl.St.Sr + Sl .Cr & Cl.Ct & pz \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (B)$$

En faisant l'identification entre les deux expressions (A et B) de la matrices POS, éléments de la dernière colonne donnent directement les paramètres de position (x, y, z) reste à déterminer l'orientation du point P, connaissant le vecteur θ_i .

$$-S1 * Ct = -C1 * S5 + S1 * C5 * S234 \quad (a)$$

$$C1 * Ct = C5 * C234 \quad (b)$$

$$(a)/(b) \implies \boxed{l = \text{Atan} \left(\frac{C1 * S5 - S1 * S234}{C5 * C234} \right)} \quad (c)$$

$$St = S1 * S5 + C1 * C5 * S234 \quad (d)$$

$$P33 = C1 * Ct \rightarrow C5 * C234 = C1 * Ct \quad (d')$$

$$(d) \text{ et } (d') \implies \boxed{t = \text{Atan2}(S1 * S5 + C1 * C5 * S234 ; C5 * C234 / C1)} \quad (e)$$

$$Ct * Cr = C1 * C234 \quad (f)$$

$$-Ct * Sr = C1 * S5 * S234 - S1 * C5 \quad (g)$$

$$\text{de } (f) \text{ et } (g) \implies \boxed{r = \text{Atan2} \left(\frac{-C1 * S5 * S234 + S1 * C5}{Ct} ; \frac{C1 * C234}{Ct} \right)} \quad (h)$$

Ann2-III- RESOLUTION DU PROBLEME GEOMETRIQUE INVERSE

Connaissant les coordonnées (Px, Py, Pz) et l'orientation (l, t, r) du point à atteindre exprimées en coordonnées généralisées par le vecteur \vec{A} (Px, Py, Pz, l, t, r), il s'agit déterminer le vecteur de coordonnées articulaires θ_i (i = 1...5) permettant d'atteindre ce poi

-III-1- Détermination de la Matrice T

-1-a-Calcul de POS = f (Px, Py, Pz, l, t, r)

Connaissant le vecteur \vec{A} (Px, Py, Pz, l, t, r), on connaît donc numériquement matrice homogène correspondante POS, qui est une fonction des paramètres de ce vecteur.

En effet on peut écrire la matrice POS comme une succession de transformati géométriques, et donc sous la forme d'un produit de matrices homogènes (cf paragrap précédent):

POS = trans (X,Px) * trans (Y, Py) * trans (Z, Pz) * rot (X, l) * rot (Y, t) * rot (Z, r), alors

$$\text{POS} = \begin{vmatrix} \text{Ct} \cdot \text{Cr} & -\text{Ct} \cdot \text{Sr} & \text{St} & \text{px} \\ \text{Sl} \cdot \text{St} \cdot \text{Cr} + \text{Cl} \cdot \text{Sr} & -\text{Sl} \cdot \text{St} \cdot \text{Sr} + \text{Cl} \cdot \text{Cr} & -\text{Sl} \cdot \text{Ct} & \text{py} \\ -\text{Cl} \cdot \text{St} \cdot \text{Cr} + \text{Sl} \cdot \text{Sr} & \text{Cl} \cdot \text{St} \cdot \text{Sr} + \text{Sl} \cdot \text{Cr} & \text{Cl} \cdot \text{Ct} & \text{pz} \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$\text{POS} = \begin{vmatrix} \text{P11} & \text{P12} & \text{P13} & \text{Px} \\ \text{P21} & \text{P22} & \text{P23} & \text{Py} \\ \text{P31} & \text{P32} & \text{P33} & \text{Pz} \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

De ces deux égalités et en procédant par identification, on a :

$$\begin{aligned} \text{P11} &= \text{Ct} \cdot \text{Cr} & \text{P12} &= -\text{Ct} \cdot \text{Sr} & \text{P13} &= \text{St} \\ \text{P21} &= \text{Sl} \cdot \text{St} \cdot \text{Cr} + \text{Cl} \cdot \text{Sr} & \text{P22} &= -\text{Sl} \cdot \text{St} \cdot \text{Sr} + \text{Cl} \cdot \text{Cr} & \text{P23} &= -\text{Sl} \cdot \text{Ct} \\ \text{P31} &= -\text{Cl} \cdot \text{St} \cdot \text{Cr} + \text{Sl} \cdot \text{Sr} & \text{P32} &= \text{Cl} \cdot \text{St} \cdot \text{Sr} + \text{Sl} \cdot \text{Cr} & \text{P33} &= \text{Cl} \cdot \text{Ct} \end{aligned}$$

-1-b-Calcul de la Matrice T

Comme Z et E sont fixes, les trois matrices Z, POS et E sont alors connues et on peut en déduire la matrice T: $T = Z^{-1} * \text{POS} * E^{-1}$.

En inversant les matrices Z et E on obtient :

$$Z^{-1} = \begin{vmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & -L6 \\ 0 & 0 & 0 & 1 \end{vmatrix} \quad E^{-1} = \begin{vmatrix} 1 & 0 & 0 & -L5 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Alors, par conséquent :

$$T = Z^{-1} * \text{POS} * E^{-1} = \begin{vmatrix} \text{P11} & \text{P12} & \text{P13} & \text{Px} & -L5 & \text{P11} \\ \text{P21} & \text{P22} & \text{P23} & \text{Py} & -L5 & \text{P21} \\ \text{P31} & \text{P32} & \text{P33} & \text{Pz} & -L5 & \text{P31} & -L6 \\ 0 & 0 & 0 & & & 1 \end{vmatrix} = \begin{vmatrix} \text{nx} & \text{sx} & \text{ax} & \text{Hx} \\ \text{ny} & \text{sy} & \text{ay} & \text{Hy} \\ \text{nz} & \text{sz} & \text{az} & \text{Hz} \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

De cette égalité, et par identification, on peut déterminer :

$$\begin{cases} nx = P11 \\ ny = P21 \\ nz = P31 \end{cases} \quad \begin{cases} sx = P12 \\ sy = P22 \\ sz = P32 \end{cases} \quad \begin{cases} ax = P13 \\ ay = P23 \\ az = P33 \end{cases}$$

$$\begin{cases} Hx = Px - L5*P11 \\ Hy = Py - L5*P21 \\ Hz = Pz - L5*P31 - L0 \end{cases}$$

Remarque:

On peut distinguer un cas particulier intéressant, c'est quand le repère du po d'arrivée a la même orientation que le repère de l'univers ($l = t = r = 0$). Dans ce cas paramètres de la matrice POS deviennent :

$$P_{i,j} = \begin{cases} 1 & \text{si } i = j \\ 0 & \text{si } i \neq j \end{cases}$$

Et les paramètres de la matrice T seront donc :

$$\begin{cases} ax = ay = az = 1 \\ ay = az = ax = az = ax = ay = 0 \end{cases} , \begin{cases} Hx = Px - L6 \\ Hy = Py \\ Hz = Pz - L0 \end{cases}$$

-III-2- Calcul des variables articulaires « θ_i »

-2-1-Généralités

C'est un problème difficile car il n'y a pas de méthode générale pour déduire vecteur des variables articulaires θ_i ($i = 1...5$) à partir de la connaissance du vecteur coordonnées généralisées \vec{A} (Px, Py, Pz, l, t, r). Différentes approches ont été utilisées par lesquelles l'intuition a une part importante. Parmi les plus connues on peut citer: la méthode de l'analyse géométrique [LEE 84], les méthodes numériques itératives (jacobien) basées sur la linéarisation de la matrice homogène T [PIE 69] ou l'utilisation du jacobien [GOR 8 l'algèbre des quaternions (introduite par Hamilton en 1843), la méthode analytique [PAU 8

Dans la méthode analytique il s'agit de déterminer le vecteur des variables articulaires θ_i ($i = 1...5$), à partir de la connaissance de la matrice T, des différentes matrices A_i et leurs inverses.

Dans le cas d'un manipulateur quelconque il n'est pas possible de faire le calcul analytiquement. Mais il a été prouvé qu'une condition suffisante est que 3 axes consécutifs (rotation ou de translation) soient concourants ou parallèles entre eux [PIE 68].

Dans ce cas la méthode consiste à prémultiplier la matrice T successivement par inverses des matrices Ai. On obtiendra donc cinq équations matricielles:

$$T = A1A2.A3.A4.A5 \quad A1^{-1}*T=A2.A3.A4.A5 \quad A2^{-1}* A1^{-1}*T=A3.A4.A5$$

$$A3^{-1}* A2^{-1}* A1^{-1}*T=A4.A5 \quad A4^{-1}*A3^{-1}* A2^{-1}* A1^{-1}*T=A5$$

Les éléments des matrices des termes de gauche sont fonction des (n-1) premières coordonnées articulaires, alors que les éléments des matrices de droite sont fonction des coordonnées 1 à 5. En égalant terme à terme on peut aboutir à des équations isolant les termes à calculer. Ainsi par exemple la deuxième équation permet d'isoler l'angle θ_1 et de l'exprimer en fonction des composantes de la matrices T. Il en est de même pour les équations 3,4,5 qui permettent de calculer respectivement les variables articulaires θ_2 , θ_3 et θ_4 . Quant à θ_5 il est extrait de la première équation.

Remarque: C'est cette manière de procéder qui suggère d'effectuer les produits de matrices A4A5, puis A3A4A5 etc...lors du calcul de l'expression de la matrice T pendant le calcul du problème géométrique direct; ce qui évite de refaire des calculs pour l'opération de transposition inverse des coordonnées.

-2-2- Calcul de l'angle θ_1 :

$$A1^{-1}*T = A2.A3.A4.A5$$

$$A1^{-1} * T = \begin{vmatrix} C1nx + S1ny & C1sx + S1sy & C1ax + S1at & C1hx + S1ht \\ - S1nx + C1ny & - S1sx + C1sy & - S1ax + C1ay & - S1hx + C1ht \\ nz & sz & az & hz - L0 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$A2*A3*A4*A5 = \begin{vmatrix} C234 & S5S234 & C5S234 & L4C234 + L3C23 + L2C2 \\ 0 & C5 & -S5 & 0 \\ - S234 & S5 C234 & C5C234 & L1 - L4S234 - L3S23 - L2S2 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

Par identification on a :

$$\left\{ \begin{array}{l} C1nx + S1ny = C234 \quad (1) \\ -S1nx + C1ny = 0 \quad (2) \\ nz = -S234 \quad (3) \end{array} \right. \left\{ \begin{array}{l} C1sx + S1sy = S5 S234 \quad (4) \\ -S1sx + C1sy = C5 \quad (5) \\ sz = S5 C234 \quad (6) \end{array} \right. \left\{ \begin{array}{l} C1ax + S1ay = C5S234 \\ -S1ax + C1ay = -S5 \\ az = C5 C234 \end{array} \right. \quad (7)$$

$$\left\{ \begin{array}{l} C1Hx + S1Hy = L4C234 + L3C23 + L2C2 \quad (10) \\ -S1Hx + C1Hy = 0 \quad (11) \end{array} \right.$$

$$Hz - L0 = L1 - L4*S234 - L3*S23 - L2*C2 \quad (12)$$

Des équations (2) et (11) on peut extraire θ_1 :

$$\left\{ \begin{array}{l} C1*ny - S1*nx = 0 \\ C1*Hy - S1*Hx = 0 \end{array} \right. \rightarrow S1/C1 = \text{tg}\theta_1 = ny/nx = HY/HX$$

$$\theta_1 = \text{ARTG } HY/HX = \text{ARTG } ny/nx$$

-2-3- Calcul de l'angle θ_{234} :

D'après les équations (1), (3), (4), (6), (7) et (9)

$$\left\{ \begin{array}{l} S234 = -nz = \frac{C1*sx + S1*sy}{S5} = \frac{C1*ax + S1*ay}{C5} \quad (1)' \\ C234 = \frac{az}{C5} = \frac{sz}{S5} = C1*nx + S1*ny \quad (2)' \end{array} \right.$$

Donc : (1)'/ (2)' donne :

$$\text{tg}234 = S234/C234 = -nz/C1*nx + S1*ny = C1*sx + S1*sy/sz = C1*ax + S1*ay/az$$

$$\Rightarrow \theta_{234} = \text{atan} \frac{-nz}{C1*nx + S1*ny} = \text{atan} \frac{C1*sx + S1*sy}{sz}$$

$$\theta_{234} = \text{atan} \frac{(C1*ax + S1*ay)}{az}$$

On peut choisir l'une des trois égalités pour déterminer θ_{234} :

$$\theta_{234} = \text{atan2} \frac{-nz}{(C1*nx + S1*ny)}$$

 avec $\left\{ \begin{array}{l} S234 = -nz \\ \text{et} \\ C234 = C1*nx + S1*ny \end{array} \right.$

-2-5 calcul de θ_2 :

On pose

$$A_1^{-1} * T = \begin{vmatrix} a1 & b1 & d1 & g1 \\ a2 & b2 & d2 & g2 \\ a3 & b3 & d3 & g3 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

On calcule $A_2^{-1} * A_1^{-1} * T = A_3 * A_4 * A_5$.

$$A_2^{-1} * A_1^{-1} * T = \begin{vmatrix} C2 & 0 & -S2 & L1 * S2 \\ 0 & 1 & 0 & 0 \\ S2 & 0 & C2 & -L1 * C2 \\ 0 & 0 & 0 & 1 \end{vmatrix} * \begin{vmatrix} a1 & b1 & d1 & g1 \\ a2 & b2 & d2 & g2 \\ a3 & b3 & d3 & g3 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$= \begin{vmatrix} C2 * a1 - S2 * a3 & C2 * b1 - S2 * b3 & C2 * d1 - S2 * d3 & C2 * g1 - (g3 - L1) * S2 \\ a2 & b2 & d2 & g2 \\ S2 * a1 + C2 * a3 & S2 * b1 + C2 * b3 & S2 * d1 + C2 * d3 & S2 * g1 + (g3 - L1) * C2 \\ 0 & 0 & 0 & 1 \end{vmatrix} \dots \text{(III)}$$

D'autre part :

$$A_3 * A_4 * A_5 = \begin{vmatrix} C34 & S5 * S34 & C5 * S34 & L4 * C34 + L3 * C3 + L2 \\ 0 & C5 & -S5 & 0 \\ -S34 & S5 * C34 & C5 * C34 & -L4 * S34 - L3 * S3 \\ 0 & 0 & 0 & 1 \end{vmatrix} \dots \text{(IV)}$$

De (III) et (IV) on a :

$$C34 = C2 * a1 - S2 * nz$$

$$-S34 = C2 * nz + S2 * a1.$$

$$C5 = b2 = C1 * sy - S1 * sx.$$

$$-S5 = d2 = C1 * ay - S1 * ax.$$

$$S5 * S34 = b1 * C2 - S2 * sz \dots \text{(i)}$$

$$S5 * C34 = S2 * b1 + C2 * sz \dots \text{(j)}$$

En substituant l'expression de C34 et S34 dans (i) et (j), on trouve :

$$C2 * (b1 + S5 * nz) = S2 * (sz - S5 * a1) \dots \text{(k)}$$

$$C2 * (sz - S5 * a1) = -S2 * (b1 + S5 * nz) \dots \text{(l)}$$

$$\text{(k) et (l) donnent : } \theta_2 = \text{Atan} \left[\frac{b1 + S5 * nz}{sz - S5 * a1} \right] = \text{Atan} \left[\frac{-(sz - S5 * a1)}{b1 + S5 * nz} \right]$$

Cette expression nous donne deux valeurs possibles pour l'angle θ_2 . Malheureusement elle ne tient pas compte de la contrainte physique sur le porteur : les axes des angles θ_2 et θ sont parallèles. Cela s'exprime par des expressions de $\sin\theta_2$ et $\cos\theta_2$ en fonction de θ_3 .

En effet selon les matrices (III) et (VI), on écrit :

$$\begin{cases} C34=C2*a1-S2*nz \dots(6) & \text{et} & \begin{cases} C2*g1-S2*(g3-L1)=L4*C34+L3*C3+L2\dots(8) \\ S2*g1+C2*(g3-L1)=-L4*S34-L3-S3\dots(9) \end{cases} \\ -S34=C2*nz+S2*a1\dots(7) \end{cases}$$

En remplaçant (6) et(7) dans (8) et (9), on trouve :

$$\begin{cases} C2*(L4*a1-g1)=S1*(L4/nz-g3+L1)-L3*C3-l1. \\ C2*(L4*nz-g3+L1)=-S2*(L4*a1-g1)+L3*S3. \end{cases}$$

$$\rightarrow \begin{cases} C2 = \frac{L3*S3*(L4*nz-g3+L1)-(L3*C3+L2)*(L4*a1-g1)}{(L4*nz-G3+L1)^2 + (L4*a1-G1)^2} \\ S2 = \frac{-(L3*C3+L2)*(L4*nz+g3+L1)+L3*S3*(L4*a1+g1)}{(L4*nz+g3+L1)^2+(L4*a1-g1)^2} \end{cases}$$

$$\theta_2 = \text{Artan2}(S2, C2)$$

Par conséquent il faut d'abord calculer θ_3 avant de calculer θ_2 .

-2-4- Calcul de θ_3

$$\text{On pose } = A_2^{-1} * A_1^{-1} * T = \begin{vmatrix} M1 & B1 & D1 & G1 \\ M2 & B2 & D2 & G2 \\ M3 & B3 & D3 & G3 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

On calcule $A_3^{-1} * A_2^{-1} * A_1^{-1} * T = A_4 * A_5$

$$A_3^{-1} * A_2^{-1} * A_1^{-1} * T = \begin{vmatrix} C3 & 0 & -S3 & -L2*C3 \\ 0 & 1 & 0 & 0 \\ -S3 & 0 & C3 & -L2*S3 \\ 0 & 0 & 0 & 1 \end{vmatrix} * \begin{vmatrix} M1 & B1 & D1 & G1 \\ M2 & B2 & D2 & G2 \\ M3 & B3 & D3 & G3 \\ 0 & 0 & 0 & 1 \end{vmatrix}$$

$$= \begin{vmatrix} C3*M1-S3*M3 & C3*B1-S3*B3 & C3*D1-S3*D1 & C3*G1-S3*G3-L2*C3 \\ M2 & B2 & D2 & G3 \\ S3*M1+C3*M3 & S3*B1+C3*B3 & S3*D1+C3*D1 & S3*G1+C3*G3-L2*S3 \\ 0 & 0 & 0 & 1 \end{vmatrix} \dots(V)$$

$$A_4 * A_5 = \begin{vmatrix} C4 & S4*S5 & S4*C5 & L4*C4+L3 \\ 0 & C5 & -S5 & 0 \\ -S4 & C4*S5 & C4*C5 & -L4*S4 \\ 0 & 0 & 0 & 1 \end{vmatrix} \dots(VI)$$

En faisant l'identification entre les éléments des deux matrices résultantes (V) et (VI) on peut calculer l'angle θ_3 de la manière suivante :

$$(1) \begin{cases} C_4 = C_3 * M_1 - S_3 * M_3 \\ -S_4 = S_3 * M_1 + C_3 * M_3 \end{cases}$$

$$(2) \begin{cases} L_4 * C_4 + L_3 = C_3 * G_1 - S_3 * G_3 - L_2 * C_3 \\ -L_4 * S_4 = C_3 * G_1 + C_3 * G_3 - L_2 * S_3 \end{cases}$$

En remplaçant M_1 et M_2 par leurs expressions, on trouve :

$$(3) \begin{cases} C_4 = C_{23} * a_1 - S_{23} * n_z \\ -S_4 = C_{23} * n_z + S_{23} * a_1 \end{cases}$$

En substituant (3), G_1 et G_3 par leurs expressions dans (2), on trouve :

$$\begin{cases} L_3 + L_2 * C_{23} = C_{23} * (g_1 - L_4 * a_1) + S_{23} * (L_4 * n_z - (g_3 - L_1)) & (4) \\ L_2 * S_3 = S_{23} * (g_1 - L_4 * a_1) - C_{23} * (L_4 * n_z - (g_3 - L_1)) & (5) \end{cases}$$

$$(4)^2 + (5)^2 \rightarrow L_2^2 + L_3^2 + 2 * L_2 * L_3 * C_3 = (g_1 - L_4 * a_1)^2 + (L_4 * n_z - (g_3 - L_1))^2$$

$$C_3 = \frac{(g_1 - L_4 * a_1)^2 + (L_4 * n_z - (g_3 - L_1))^2 - (L_2^2 + L_3^2)}{2 * L_2 * L_3}$$

Sachant que:

$$\left. \begin{aligned} a &= (g_1 - L_4 * a_1) = C_1 * H_x + S_1 * H_y - L_4 * (C_1 * n_x + S_1 * n_y) \\ b &= (L_4 * n_z - (g_3 - L_1)) = L_4 * n_z - (H_z - L_0) - L_1 \end{aligned} \right\} C_3 = \frac{a^2 + b^2 - (L_2^2 + L_3^2)}{2 * L_2 * L_3}$$

$$C_3 = \frac{[C_1 * H_x + S_1 * H_y - L_4 * (C_1 * n_x + S_1 * n_y)]^2 + [L_4 * n_z - (H_z - L_0) - L_1]^2 - (L_2^2 + L_3^2)}{2 * L_2 * L_3}$$

$$S_3 = \pm \sqrt{1 - C_3^2} \quad \text{et} \quad \theta_3 = \text{Atan2}(S_3, C_3).$$

On aura donc deux valeurs possibles pour l'angle θ_3 en fonction du signe choisi pour la valeur du sinus.

-2-6-calcul de l'Angle θ_4

$$\theta_{234} = \theta_2 + \theta_3 + \theta_4$$

$$\Rightarrow \theta_4 = \theta_{234} - \theta_3 - \theta_2$$

-2-7- calcul de l'Angle θ_5

En faisant l'identification entre les deux expressions de la matrice T (I) et (II), on trouve :

$$\begin{aligned} S_5 * C_{234} &= s_z \\ C_5 * C_{234} &= a_z \end{aligned} \Rightarrow \begin{cases} S_5 = s_z / (C_{234}) = s_z / (C_1 * n_x + S_1 * n_y) \\ C_5 = a_z / (C_{234}) = a_z / (C_1 * n_x + S_1 * n_y) \end{cases} \Rightarrow \theta_5 = \text{Atan2}(S_5 ; C_5).$$

-III-3- Etude de la résolvabilité

-3-1-Angles θ_1 , θ_{234} , θ_5

Les angles (θ_1 , θ_{234} et θ_5) ont été déterminés avec précision et sans aucune ambiguïté car on a utilisé la fonction arctangente (atan2) à deux arguments (y,x). Cette fonction utilise les signes de y et x , pour localiser précisément l'angle sur le cercle trigonométrique.

$$\theta_1 = \text{Atan2} (HY , HX)$$

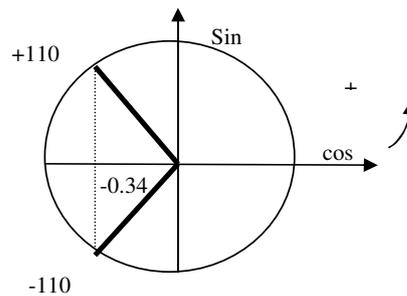
$$\theta_{234} = \text{Atan2} (S_{234}, C_{234}) = \text{Atan2} (-nz, C1*nx + S1*ny)$$

$$\theta_5 = \text{Atan2} (S_5, C_5) = \text{Atan2} (S1*ax - C1*ay , C1*sy - S1*sx)$$

Après avoir déterminé l'angle il reste à tester son appartenance à l'intervalle de travail

-3-2-Angle θ_3

Comme on vient de le voir au paragraphe précédent le $\cos \theta_3$ est connu avec précision. Quant au sinus il a été calculé à partir du sinus et peut prendre deux valeurs de signes opposés (\pm), il faut donc effectuer un choix entre ces deux valeurs.



Par construction $-110^\circ \leq \theta_3 \leq +110^\circ$.

- Si $\cos \theta_3 < -0,34$: il n'y a pas de problème puisque l'angle n'appartient pas à l'intervalle de fonctionnement.
- Par contre si $\cos \theta_3 \geq -0,34$ on a deux valeurs possibles pour θ_3 : l'une positive appartient à $[0, 110^\circ]$ et l'autre négative appartient à $[-110^\circ, 0]$.

On a donc le choix entre deux intervalles de fonctionnement, donnant deux signes différents pour le sinus. Pour résoudre cette ambiguïté sans limiter les mouvements du SM on a eu recours à la notion de "coude vers le haut" et "coude vers le bas".

Dans le SMA IR52-C le coude est représenté par l'angle θ_3 et le poignet par l'angle θ_4 . Le coude vers le haut correspond à $\theta_3 \in [0, 110^\circ]$, et le coude vers le bas à $\theta_3 \in [-110^\circ, 0]$.

Comme le coude vers le haut impose un angle positif, ceci implique:

$$\sin \theta_3 = +\sqrt{1 - C\theta_2^2} \quad \rightarrow \quad \theta_3 = \text{Atan2} [S\theta_3, C\theta_3].$$

Le coude vers le bas imposera un angle négatif et on écrit :

$$\sin \theta_3 = -\sqrt{1 - C\theta_2^2} \quad \rightarrow \quad \theta_3 = \text{Atan2} [S\theta_3, C\theta_3].$$

Le schéma suivant illustrera ces deux positions.

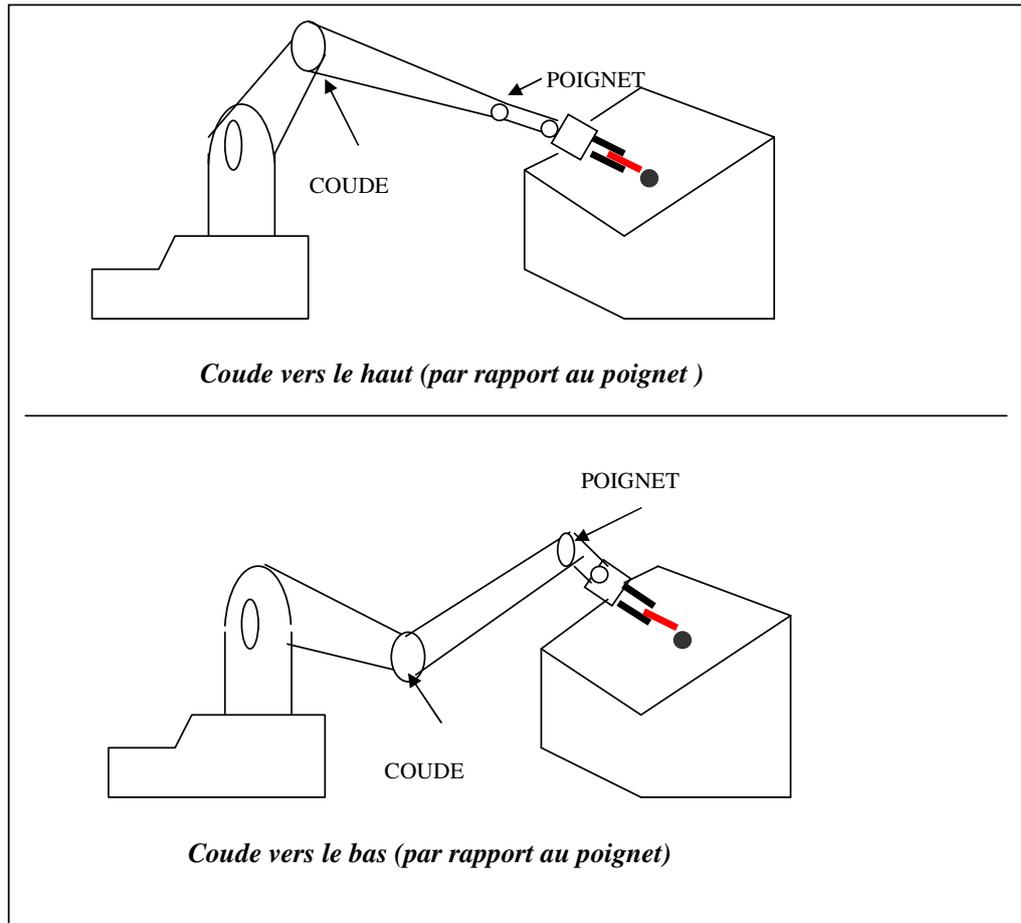


Figure 2.6: Positions du coude par rapport au poignet

Différentes contraintes peuvent orienter le choix de position du coude. Ces contraintes seront prises en compte automatiquement, lors de l'exécution du programme de commande pour fixer la valeur de l'angle θ_3 .

Une des contraintes a trait à l'environnement de la tâche. Le critère de choix entre coude en haut ou en bas par rapport au poignet, sera l'évitement des obstacles qui peuvent exister sur le trajet du SMA.

Une autre contrainte (qui est la plus utilisée) est celle du minimum d'énergie: on utilise le minimum d'énergie possible pour le déplacement de l'effecteur d'un point de départ à point d'arrivée. Par exemple si l'angle θ_{3d} de départ est positif, il est préférable que le bras se porte sur la position "coude en haut" qui donne un angle θ_{3a} d'arrivée positif. À l'inverse $\theta_{3d} < 0$, il est préférable que le coude soit dans la position "coude en bas" qui donne $\theta_{3a} < 0$. De manière plus précise, il serait souhaitable de raisonner sur la variation d'angle la plus petite possible ($\Delta\theta = \theta_{3a} - \theta_{3d}$). Cette contrainte peut être rajoutée comme critère de choix de l'option coude vers le haut ou vers le bas.

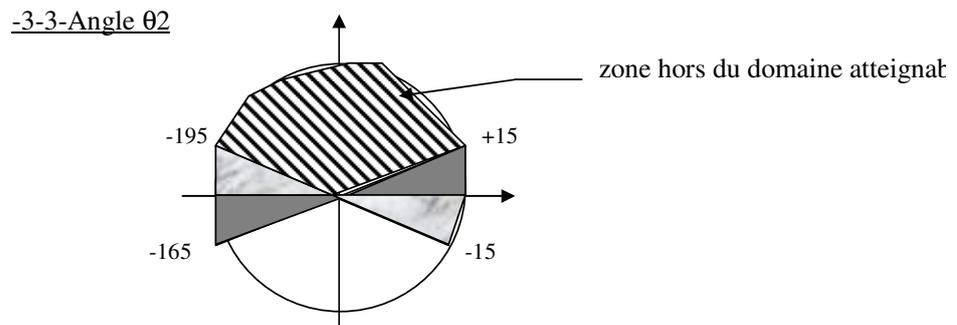


Figure 2.7: Domaine atteignable de l'angle θ_2

$$\theta_2 = \text{Atan} \frac{b_1 + S_5 * n_z}{sz - S_5 * a_1} = \text{Atan} \frac{-(sz - S_5 * a_1)}{b_1 + S_5 * n_z}$$

L'angle θ_2 ayant été calculé à partir d'une fonction arctangente, et compte tenu du domaine atteignable de θ_2 , si $-165 < \theta_2 < -15$, alors il n'y a aucune incertitude sur la valeur de l'angle.

Par contre les zones proches des butées mécaniques donnent des angles qui ont même tangente. Cela donne 4 zones couplées deux à deux dont les valeurs correspondent à $(-180 \leq \theta_2 \leq -165$ ou $0 \leq \theta_2 \leq 15$) et $(-195 \leq \theta_2 \leq -180$ ou $-15 \leq \theta_2 \leq 0)$.

Pour lever l'indétermination on peut procéder par la méthode du "choix de la valeur plus éloignée d'une butée articulaire". Si cette technique offre l'avantage de faire un choix manière automatique (sans intervention de l'opérateur), malheureusement cela peut donner un angle qui ne vérifie pas les contraintes sur les expressions de $\sin\theta_2$ et $\cos\theta_2$ en fonction de l'angle θ_3 :

$$\begin{cases} \sin\theta_2 = \frac{-(L_3 \cdot C_3 + L_2) \cdot (L_4 \cdot n_z + G_3 + L_1) + L_3 \cdot S_3 \cdot (L_4 \cdot a_1 + G_1)}{(L_4 \cdot n_z + G_3 + L_1)^2 + (L_4 \cdot a_1 - G_1)^2} \\ \cos\theta_2 = \frac{L_3 \cdot S_3 \cdot (L_4 \cdot n_z - G_3 + L_1) - (L_3 \cdot C_3 + L_2) \cdot (L_4 \cdot a_1 - G_1)}{(L_4 \cdot n_z - G_3 + L_1)^2 + (L_4 \cdot a_1 - G_1)^2} \end{cases}$$

En conclusion pour lever l'indétermination dans le choix de θ_2 , il faut faire appel à des expressions de θ_2 en fonction de θ_3 , car elles **expriment une contrainte physique de structure mécanique articulée** (axes de rotation du porteur consécutifs et parallèles). Il est par conséquent obligatoire de ne pas se contenter dans le calcul de θ_2 de l'expression de la tangente de θ_2 qui est indépendante de θ_3 .

Dès que θ_3 est fixé, donc son cosinus et son sinus sont fixés, l'ambiguïté sur la valeur de θ_2 se retrouve alors levée d'elle-même. Il suffit que la valeur trouvée appartienne au domaine atteignable.

-3-4- Angle θ_4

$$\theta_4 = \theta_{234} - \theta_3 - \theta_2.$$

Puisque l'indétermination est levée sur les angles θ_2 et θ_3 , et que θ_{234} est connu avec précision, par conséquent il n'y a aucune indétermination dans le calcul de θ_4 .

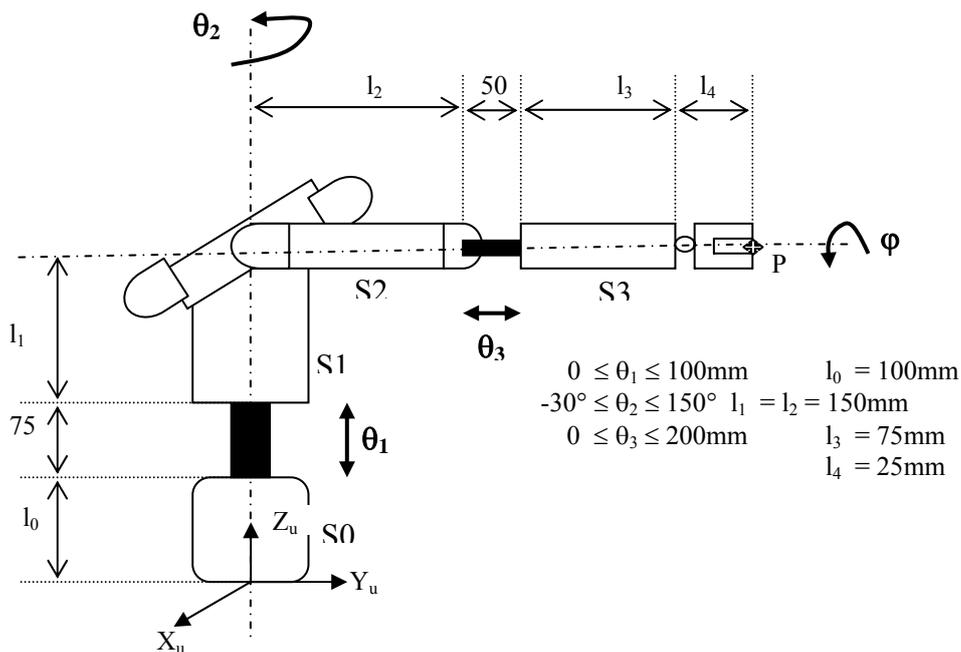
Annexes 3 : MODELISATION GEOMETRIQUE DE ROBOTS

(cf automatismes logiques vol2 partie 6)

Exercice 6.1 : Modélisation libre et problème géométrique inverse

On considère le SMA donné par la figure ci-après. Le porteur est constitué de de 4 segments dont 3 sont mobiles. Les articulations θ_1 et θ_3 sont prismatiques, et l'articulation θ_2 est rotoïde. La pince tourne d'un angle φ par rapport au segment S_3 . Sur la figure on a $\underline{\theta} (75, 0, 50)$ et $\varphi = 90^\circ$.

- 1° Faire une figure indiquant les différents repères associés aux segments du porteur .
- 2° Etablir les matrices de transformation homogène de changement de repère A_{i-1}^i , en détaillant chaque étape.
- 3° En déduire la matrice T du porteur.
- 4° Calculer les matrices E (effecteur par rapport à R_0) et Z (R_0 par rapport au repère de référence de l'univers R_u donné sur la figure).
- 5° Donner les expressions de X_P , Y_P , Z_P , en fonction des variables articulaires θ_i , des longueurs l_i et de φ . Que remarque-t-on? Que peut on en conclure?
- 6° Pour la configuration de la figure donner les valeurs de X_P , Y_P , Z_P .
- 7° Donner les expressions des variables θ_i en fonction de X_P , Y_P , Z_P et des longueurs l_i .
- 8° On veut atteindre un point de l'espace de coordonnées $X = Y = 2l_2$, $Z = 2l_0$. Le robot est -il résolvable? Combien y- a- t- il de configurations solutions? (on donnera dans chaque cas les valeurs des variables articulaires).
- 9° Même question que pour la question 8° avec $X = Y = l_1 \sqrt{2}$, $Z = 2 l_1$.



Exercice 6.2: modélisation libre avec articulations doubles, et problème géométrique direct

On considère le robot de type "scara" modifié, possédant 5 articulations dont deux sont doubles: rotation θ_1 , rotation θ_2 + translation T_1 (articulation double liant les segments de longueurs l_2 et l_1), translation T_2 + rotation φ de la tige de longueur l_3 par rapport à son axe de translation.

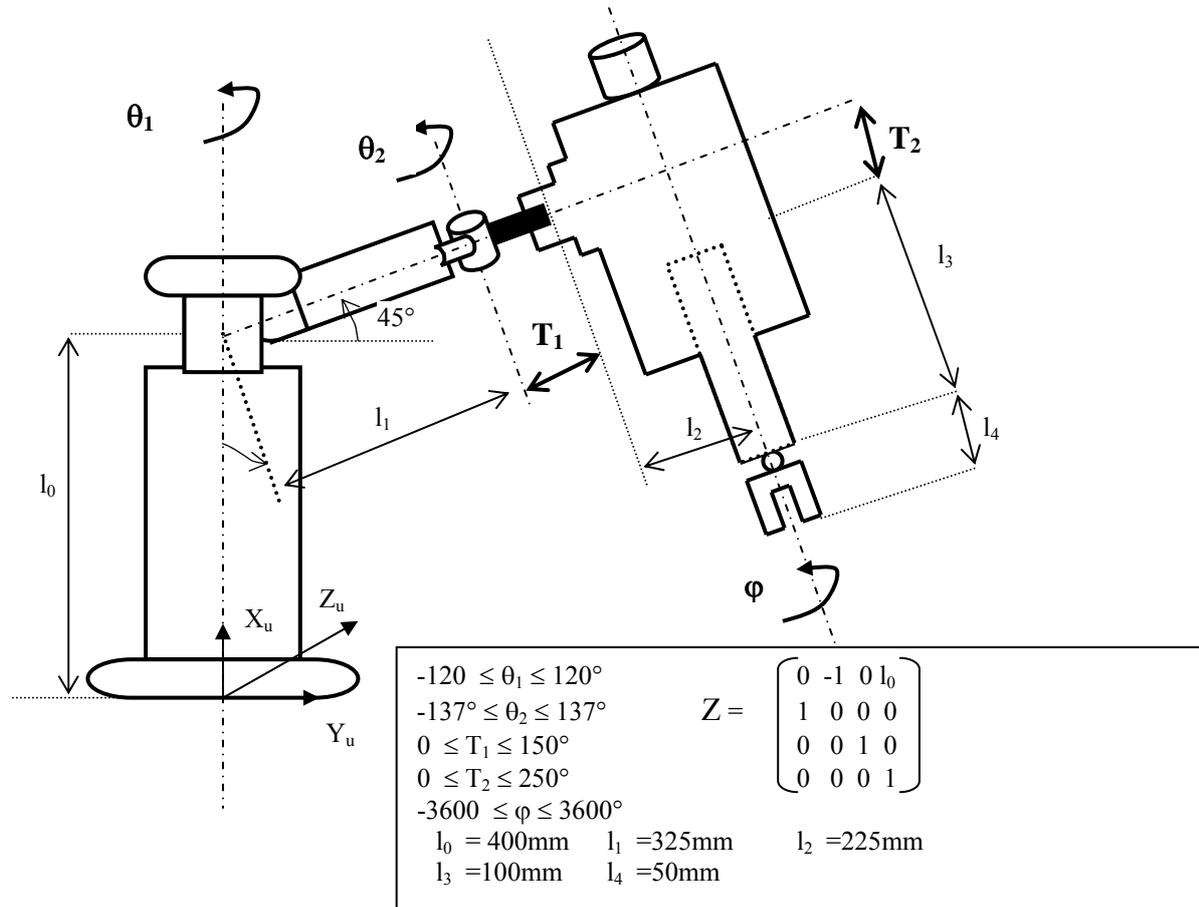
1° Faire une figure indiquant les différents repères associés aux segments du porteur et de l'effecteur.

2° Etablir les matrices de transformations homogènes de changements de repères (l'expression de Z est donnée sur la figure).

3° On veut faire prendre un objet par le manipulateur. Quand l'objet est tenu par la pince les capteurs proprioceptifs donnent pour vecteur de variables articulaires: $\theta_1 = 45^\circ$, $\theta_2 = \varphi = 0^\circ$, $T_1 = 50\text{mm}$, $T_2 = 75\text{mm}$. Calculez la matrice donnant la position et l'orientation de l'objet par rapport au repère de base R_0 du manipulateur. En déduire les transformations géométriques permettant de passer de R_0 au repère de l'objet. Tracer ces repères.

4° Déduire de la question 2 la matrice T du porteur.

5° Donner les expressions de X_P , Y_P , Z_P (coordonnées de l'extrémité de l'effecteur par rapport au repère de l'univers) en fonction des variables θ_1 , θ_2 , T_1 , T_2 , φ et des longueurs l_i .



Exercice 6.3: modélisation par la méthode de Denavit-Hartenberg

Considérons un SMA à 6 segments, dont les paramètres sont donnés dans la modélisation de Denavit-Hartenberg par le tableau suivant.

Un caméra est fixée sur le segment S4, et repérée dans le repère R4 par la matrice CAM. Tout objet sera décrit par rapport à Rc (caméra) par sa matrice OBJ, et par rapport à Re (effecteur) par une matrice X.

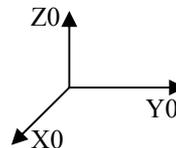
1° Exprimer X en fonction des Ai, CAM, Z, OBJ, et E. Pour simplifier les expressions on posera $T4 = Z \times A1 \times A2 \times A3 \times A4$.

2° Quand l'effecteur tient l'objet, quelle est la valeur de X ? Dans ce cas la caméra donne pour matrice OBJ celle indiquée plus bas, les capteurs proprioceptifs donnent $\underline{\theta} (0,0,0,0,0)$ et $d3=40$, et le système donne pour coordonnées de l'effecteur la matrice Y (coordonnées par rapport à R0).

- a°) Faire une figure sur laquelle seront indiqués tous les repères associés aux segments.
- b°) Donner à partir de la figure la valeur de la matrice T du porteur.
- c°) Calculer la valeur de la matrice E sans utiliser le résultat de T.
- d°) Calculer la matrice T du porteur. Conclure.
- e°) Si on intervertit α_1 et α_2 , calculer à nouveau T et E, puis TE. Conclure.

Seg men	θ_i	d_i	a_i	α_i
S1	θ_1	40	0	-90
S2	θ_2	30	0	90
S3	0	d_3	0	0
S4	θ_4	0	0	-90
S5	θ_5	10	0	90

On prend comme orientation pour le repère R0 le repère ci-contre.



CAM=

$$\begin{bmatrix} 0 & 0 & -1 & 5 \\ 0 & -1 & 0 & 0 \\ -1 & 0 & 0 & 10 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Y=

$$\begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 48 \\ 0 & 1 & 0 & 80 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

OBJ=

$$\begin{bmatrix} 0 & 0 & -1 & -8 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 5 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Exercice 6.4 : modélisation par la méthode de Denavit-Hartenberg et couplage robot-vision

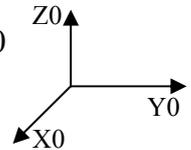
Considérons un SMA dont les paramètres sont donnés dans la modélisation de Denavit-Hartenberg par le tableau suivant. On rappelle que dans cette méthode le passage d'un repère à un autre se fait par une succession de 4 transformations géométriques ordonnées :

$$T_{i-1}^i \equiv A_i \equiv \text{Rot}(Z_{i-1}, \theta_i) * \text{Trans}(Z_{i-1}, d_i) * \text{Trans}(X_i, a_i) * \text{Rot}(X_i, \alpha_i)$$

aligne	fait coïncider	fait coïncider	fait coïncider
x_{i-1} et x_i	x_{i-1} et x_i	o_{i-1} et o_i	les repères

1° Donner les différentes matrices de changement de repère A_i

2° Positionner sur une figure les repères associés aux différents segments, ainsi que ceux de l'effecteur, avec comme valeurs des θ_i : (90, 0, 90, 0, 0, 0) et comme repère R_0



3° La structure mécanique articulée est déposée au coin d'une table.

Une caméra CAM1 est montée sur le segment Numéro 2 (segment S2) et sa situation est donnée par rapport à R_2 par une matrice CAM1. Une caméra fixe CAM2 observe la scène, et sa situation est donnée par rapport au repère de référence de la table par une matrice CAM2. Pour déterminer ces matrices (dont on a oublié les valeurs calculées lors de la calibration) on procède à une double expérience :

- On dispose un objet sur la table. On prend une image à l'aide de la caméra fixe CAM2. Elle donne les coordonnées de l'objet par rapport à CAM2 : CAM2OBJ.
- On déplace ensuite par guidage le manipulateur jusqu'à ce que la pince tienne l'objet. La caméra CAM1 prend alors une image et donne les coordonnées de l'objet par rapport à CAM1 : CAM1OBJ. Dans cette situation le calculateur du robot donne par le biais de ses capteurs proprioceptifs :

- les coordonnées de l'objet (par rapport à R_0) : matrice ROBOBJ .

- les valeurs des θ_i : $\theta_3 = 90^\circ$ $\theta_4 = 0^\circ$ $\theta_5 = 90^\circ$ $\theta_6 = 0^\circ$

a°) Donner la valeur de la matrice CAM2.

b°) Donner la valeur de la matrice CAM1.

c°) Donner la valeur de CAM2CAM1 qui donne la position et l'orientation de la caméra 1 par rapport à la caméra 2.

d°) Donner la valeur de CAM1CAM2 qui donne la position et l'orientation de la caméra 2 par rapport à la caméra 1.

4° Si l'on inverse les valeurs de alpha1 et alpha2, refaire les questions 1 et 2 (quels sont les changements? Donner les nouvelles valeurs des matrices, et tracer les nouveaux repères).

Remarque : On n'a pas besoin de calculer la matrice T du porteur.

Articulation	θ_i	d_i	a_i	α_i
1	θ_1	300	0	-90
2	θ_2	0	300	90
3	θ_3	0	0	0
4	θ_4	400	0	-90

5	θ_5	0	0	90
6	θ_6	130	0	0

$$A_i = \begin{pmatrix} \cos \theta_i & -\sin \theta_i \cos \alpha_i & \sin \theta_i \sin \alpha_i & a_i \cos \theta_i \\ \sin \theta_i & \cos \theta_i \cos \alpha_i & -\sin \alpha_i \cos \theta_i & a_i \sin \theta_i \\ 0 & \sin \alpha_i & \cos \alpha_i & d_i \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

} ← articulations propres à l'effecteur

$$\text{CAM1OBJ} \begin{pmatrix} 0 & 0 & -1 & -8 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 5 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\text{CAM2OBJ} \begin{pmatrix} 0 & 0 & -1 & -50 \\ 0 & 1 & 0 & 10 \\ 1 & 0 & 0 & 30 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

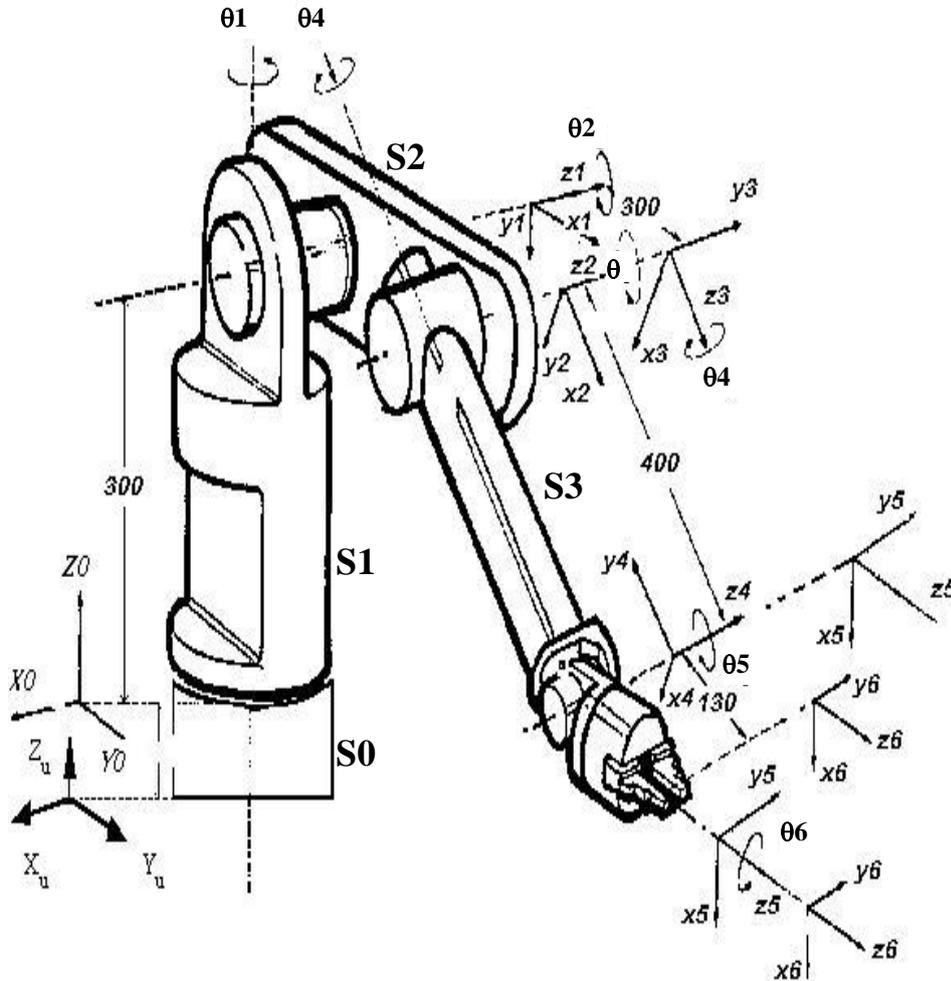
$$\text{ROBOBJ} \begin{pmatrix} 0 & -1 & 0 & 40 \\ 1 & 0 & 0 & -50 \\ 0 & 0 & 1 & 10 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$\text{ROB} \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 50 \\ 0 & 0 & 0 & 1 \end{pmatrix} \equiv Z$$

Exercice 6.5 : modélisation par la méthode de Denavit-Hartenberg

Déduire des repères indiqués sur la figure, la matrice donnant les valeurs des différents paramètres de transformation homogène dans la méthode de Denavit-Hartenberg, et cela pour chaque segment.

Comparer les résultats obtenus avec ceux de la question 4 de l'exercice précédent.



Solution

Articulation	θ_i	d_i	a_i	α_i
1	90	300	0	-90
2	0	0	300	0
3	90	0	0	90
4	0	400	0	-90

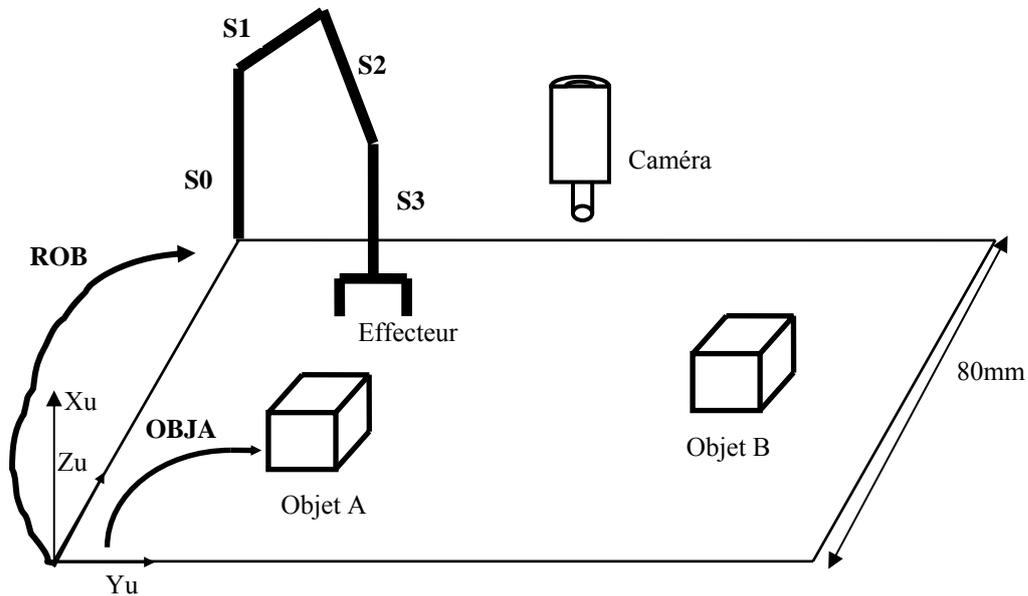
5	0	0	0	90
6	0	130	0	0

} ← articulations propres à l'effecteur

Exercice 6.6

Considérons une structure mécanique articulée à 4 segments (S0 à S3) disposée au coin supérieur gauche d'une table, le repère de l'univers étant lui au coin inférieur gauche. Deux *objets cubiques A et B sont disposés sur la table* et une caméra fixe observe la scène.

- 1°-
 - a°- Donner l'expression de la matrice T du porteur quand l'effecteur tient l'objet A.
 - b°- Calculer cette matrice.
 - c°- En déduire l'orientation du repère R3 puis celle de l'effecteur. Tracer ces repères ainsi que celui associé à S0.
- 2°-
 - a°- Donner l'expression de l'objet B par rapport à la base du manipulateur quand l'effecteur tient l'objet.
 - b°- Calculer cette matrice.
 - c°- En déduire l'orientation du repère associé à l'objet B. Tracer ce repère.
- 3°-
 - a°- Déduire des questions précédentes la hauteur des objets A et B, si on suppose que les repères des objets et de la caméra sont placés aux centres de gravité.
 - b°- Refaire le dessin de la table à l'échelle en disposant les repères (Ru, R0, RobjA, RobjB, Rcam).
 - c°- Que remarquez vous pour la disposition des objets A et B et la caméra?



ROB ≡ Z	OBJA	E	CAM	CAMOBJB
0 -1 0 0	1 0 0 10	0 -1 0 0	0 0 -1 80	0 0 -1 -50
1 0 0 0	0 1 0 80	1 0 0 0	0 -1 0 50	0 1 0 10
0 0 1 80	0 0 1 10	0 0 1 10	-1 0 0 40	1 0 0 30
0 0 0 1	0 0 0 1	0 0 0 1	0 0 0 1	0 0 0 1