

# ROBOTIQUE

ISTIA, Université Angers

Jean-Louis Boimond

La **robotique** peut être définie comme l'ensemble des techniques et études tendant à concevoir des systèmes mécaniques, informatiques ou mixtes, capables de se substituer à l'homme dans ses fonctions motrices, sensorielles et intellectuelles.

## 1 GÉNÉRALITÉS

- 1.1 Définitions
- 1.2 Constituants d'un robot
- 1.3 Classification des robots
- 1.4 Caractéristiques d'un robot
- 1.5 Les générations de robot
- 1.6 Programmation des robots

## 2 DEGRÉ DE LIBERTÉS - ARCHITECTURE

- 2.1 Positionnement d'un solide dans l'espace
- 2.2 Liaison
- 2.3 Mécanismes
- 2.4 Morphologie des robots manipulateurs

## 3 MODÈLE GÉOMÉTRIQUE D'UN ROBOT EN CHAÎNE SIMPLE

- 3.1 Nécessité d'un modèle
- 3.2 Coordonnées opérationnelles
- 3.3 Translation et rotation
- 3.4 Matrices de transformation homogène
- 3.5 Obtention du modèle géométrique
- 3.6 Paramètres de *Denavit-Hartenberg* modifié
- 3.7 Exemple
- 3.8 Exercice
- 3.9 Inversion du modèle géométrique - Méthode de Paul
- 3.10 Solutions multiples – Espace de travail – Aspects

## 4 GÉNÉRATION DE MOUVEMENT

## 5 DESCRIPTION PRATIQUE DU ROBOT STÄUBLI RX 90

- 5.1 Description générale
- 5.2 Mise en route du système
- 5.3 Mise en position initiale du bras du robot
- 5.4 Arrêt du système

## 6 MISE EN MOUVEMENT DU BRAS DU ROBOT STÄUBLI RX 90

- 6.1 Les modes de déplacement
- 6.2 Contrôle du robot à partir du pendant
- 6.3 Contrôle du robot à partir d'un programme

## **7 DESCRIPTION PRATIQUE DES ROBOTS FANUC LR MATE 100 IB ET ARC MATE 100 IB**

- 7.1 Description générale
- 7.2 Mise en route du système
- 7.3 Arrêt du système

## **8 MISE EN MOUVEMENT DU BRAS DES ROBOTS FANUC LR MATE 100 IB ET ARC MATE 100 IB**

- 8.1 Les modes de déplacement
- 8.2 Mise en mouvement en mode manuel
- 8.3 Mise en mouvement en mode programme
- 8.4 Quelques instructions de mouvement
- 8.5 Entrées/Sorties
- 8.6 Structures algorithmiques de base
- 8.7 Sous-programme
- 8.8 Exemple de programme

### **Bibliographies :**

- 1) *Modeling, Identification & Control of Robots*, W. Khalil, E. Dombre, Hermes Penton Science 2002, 480 pages
- 2) *Robotique. Aspects fondamentaux*, J.-P. Lallemand, S. Zeghloul, Masson 1994, 312 pages
- 3) *Modélisation et commande des robots*, W. Khalil, G. Lebre, Cours EI3 Automatique de l'ECN 94/95
- 4) *Cours de robotique*, J.-L. Ferrier, DESS ASC
- 5) *Robots. Principes et contrôle*, C. Vibet, Ellipses 1987, 207 pages
- 6) *Cours de robotique*, J. Gangloff, ENSPS, <http://hp2gra.u-strasbg.fr/library/teaching/robotics/>
- 7) *Introduction to Robotics Mechanics and Control*, 2<sup>th</sup> edition, John J. Craig, Addison-Wesley Publishing Company, 1989, 450 pages.

## **1 GÉNÉRALITÉS**

Pour *concevoir, simuler* ou *commander* un robot, il est nécessaire, entre autres, de disposer de *modèles* du mécanisme. Plusieurs niveaux de modélisation sont possibles. Ils dépendent des spécifications du cahier des charges de l'application envisagée : il en découle des modèles *géométriques*, *cinématiques*<sup>1</sup> et *dynamiques* à partir desquels peuvent être engendrés les mouvements du robot, ou bien des modèles *statiques* qui décrivent les interactions du mécanisme avec son environnement.

L'obtention de ces différents modèles n'est pas aisée, la difficulté variant selon la complexité de la cinématique de la chaîne articulée. Entrent en ligne de compte le nombre de degrés de liberté, le type des articulations mais aussi le fait que la chaîne peut être ouverte simple, arborescente ou fermée.

### **1.1 Définitions**

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

En fait, l'image que chacun s'en fait est généralement vague, souvent un robot est défini comme un *manipulateur automatique à cycles programmables*.

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

---

<sup>1</sup> *cinématique* : partie de la mécanique qui étudie les mouvements des corps, abstraction faite des forces qui les produisent.

- La *versatilité*<sup>2</sup> : Un robot doit avoir la capacité de pouvoir exécuter une variété de tâches, ou la même tâche de différente manière ;
- L'*auto-adaptativité* : Un robot doit pouvoir s'adapter à un environnement changeant au cours de l'exécution de ses tâches.

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

Historique :

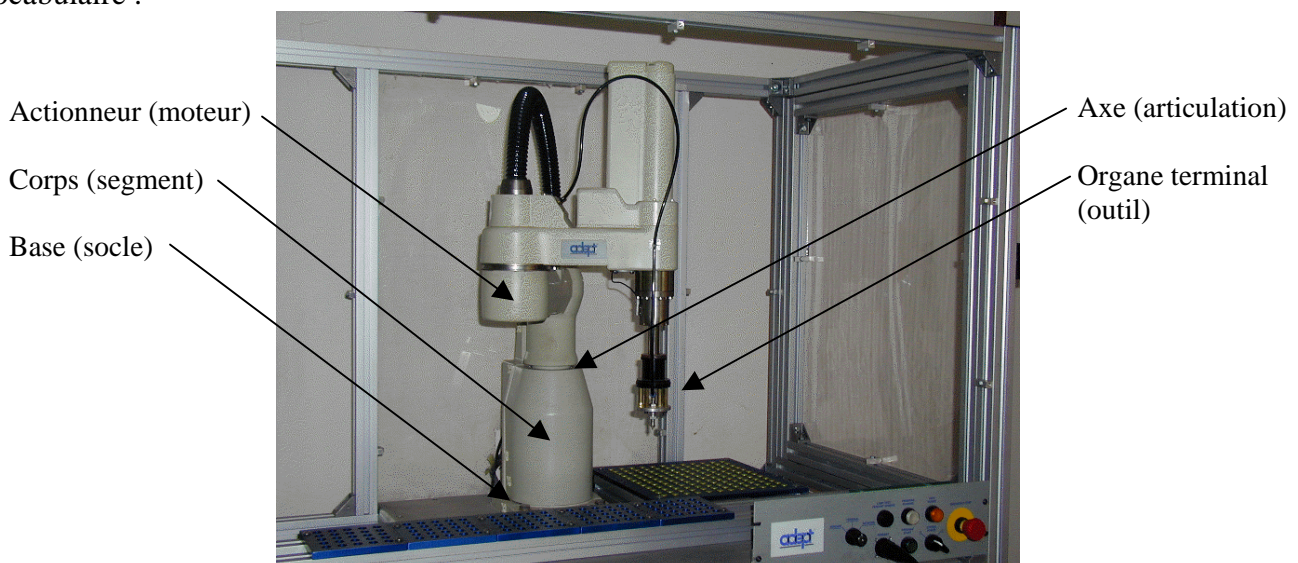
- 1947 : Premier manipulateur électrique téléopéré.
- 1954 : Premier robot programmable.
- 1961 : Utilisation d'un robot industriel, commercialisé par la société UNIMATION (USA), sur une chaîne de montage de General Motors.
- 1961 : Premier robot avec contrôle en effort.
- 1963 : Utilisation de la vision pour commander un robot.

Pour illustrer ce cours, nous utiliserons deux robots industriels : un robot Stäubli de la gamme RX, un robot FANUC ARC (6 axes) ou LR (5 axes).

- Une synthèse sur le thème de la *robotique* est faite dans un article écrit par B. Espiau (INRIA Rhône-Alpes) et intitulé *La robotique : Histoire et perspectives*. Ce document est disponible sur le WEB à l'adresse <http://www.inrialpes.fr/bip/Bip-2000/article-long.html>.

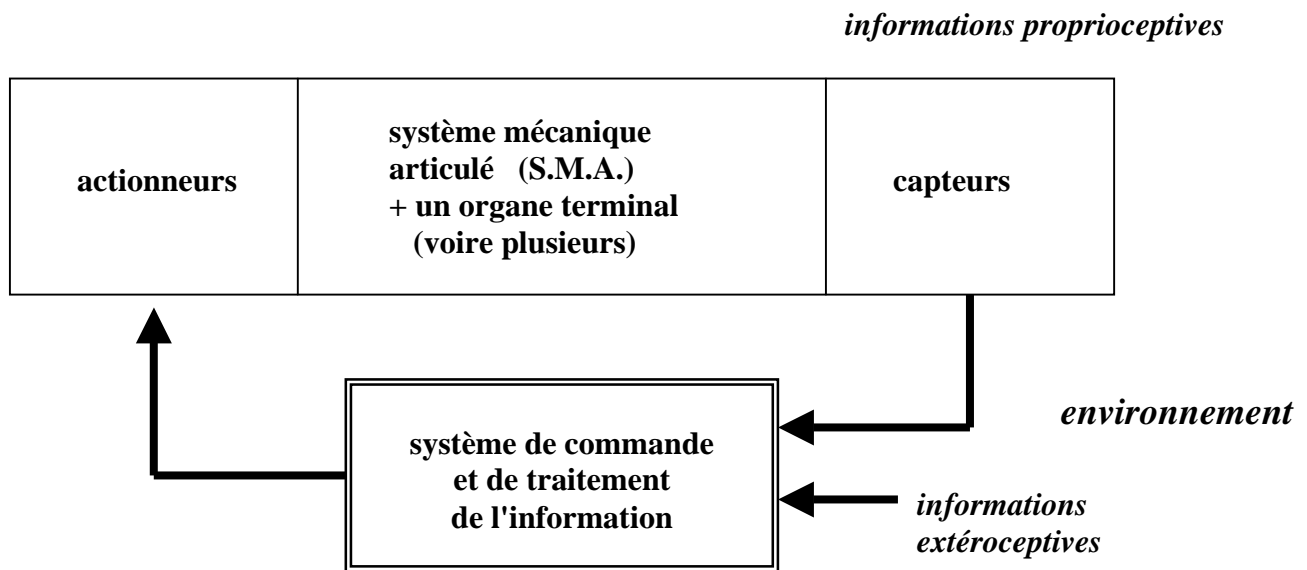
## 1.2 Constituants d'un robot

Vocabulaire :



<sup>2</sup> **versatilité** : caractère versatile (changeant).

On distingue classiquement 4 parties principales dans un robot manipulateur :



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

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

Précisons la notion d'**articulation** : Une *articulation* lie deux corps successifs en limitant le nombre de degré de liberté (notion précisée au §2.2) de l'un par rapport à l'autre. Soit  $m$  le nombre de degré de liberté résultant, encore appelé *mobilité de l'articulation*. La mobilité d'une articulation est telle que :

$$0 \leq m \leq 6.$$

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

<sup>3</sup> Les caractéristiques propres aux robots mobiles ne sont pas prises en compte dans ce cours. La mobilité d'un robot permet notamment d'augmenter son espace de travail.

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



Figure : Symbole de l'articulation *rotoïde*.

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



Figure : Symbole de l'articulation *prismatique*.

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

- Pour être animé, le S.M.A. comporte des moteurs le plus souvent avec des transmissions (courroies crantées), l'ensemble constitue les **actionneurs**. Les actionneurs utilisent fréquemment des moteurs *électriques* à aimant permanent, à courant continu, à commande par l'induit (la tension n'est continue qu'en moyenne car en général l'alimentation est un hacheur de tension à fréquence élevée ; bien souvent la vitesse de régime élevée du moteur fait qu'il est suivi d'un réducteur, ce qui permet d'amplifier le couple moteur). On trouve de plus en plus de moteurs à commutation électronique (sans balais), ou, pour de petits robots, des moteurs pas à pas.  
Pour les robots devant manipuler de très lourdes charges (par exemple, une pelle mécanique), les actionneurs sont le plus souvent *hydrauliques*, agissant en translation (vérin hydraulique) ou en rotation (moteur hydraulique).  
Les actionneurs *pneumatiques* sont d'un usage général pour les *manipulateurs à cycles* (*robots tout ou rien*). Un manipulateur à cycles est un S.M.A. avec un nombre limité de degrés de liberté permettant une succession de mouvements contrôlés uniquement par des capteurs de fin de course réglables manuellement à la course désirée (asservissement en position difficile dû à la compressibilité de l'air).
- La perception permet de gérer les relations entre le robot et son environnement. Les organes de perception sont des **capteurs** dits *proprioceptifs*<sup>4</sup> lorsqu'ils mesurent l'état interne du robot

<sup>4</sup> **proprioception** : sensibilité propre aux os, aux muscles, aux tendons et aux articulations et qui renseigne sur la statique, l'équilibration, le déplacement du corps dans l'espace, *etc.*

(positions et vitesses des articulations) et *extéroceptifs*<sup>5</sup> lorsqu'ils recueillent des informations sur l'environnement (détection de présence, de contact, mesure de distance, vision artificielle).

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

S'ajoutent à cela :

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

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

### 1.3 Classification des robots

On retiendra pour notre part 3 types de robot :

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

Pour cette dernière classe, on peut distinguer :

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

- |              |   |
|--------------|---|
| Des pièces : | Stockage – déstockage,<br>Palettisation – dépalettisation,<br>Chargement – déchargement de machine outil,<br>Manipulation d'éprouvettes,<br>Assemblage de pièces, ... |
| Des outils : | Soudure en continu ou par points,<br>Peinture,<br>Collage,<br>Ebavurage, ...  |

---

<sup>5</sup> **information extéroceptive** : information issue de récepteurs sensoriels situés à la surface du corps et stimulés par des agents extérieurs à l'organisme (chaleur, piquûre).

2. Les **robots didactiques**, qui sont des versions au format réduit des précédents robots. La technologie est différente, de même que les constructeurs. Ils ont un rôle de formation et d'enseignement, ils peuvent aussi être utilisés pour effectuer des tests de faisabilité d'un poste robotisé.
3. Les **robots mobiles autonomes** : Les possibilités sont plus vastes, du fait de leur mobilité. Notamment, ils peuvent être utilisés en zone dangereuse (nucléaire, incendie, sécurité civile, déminage), inaccessible (océanographie, spatial). De tels robots font appel à des capteurs et à des logiciels sophistiqués. On peut distinguer 2 types de locomotion : Les robots *marcheurs* qui imitent la démarche humaine, et les robots *mobiles* qui ressemblent plus à des véhicules.

Dans ce cours, on se restreint aux *robots manipulateurs*.

## 1.4 Caractéristiques d'un robot

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

- La charge maximum transportable (de quelques kilos à quelques tonnes), à déterminer dans les conditions les plus défavorables (en élongation maximum).
- L'architecture du S.M.A., le choix est guidé par la tâche à réaliser (quelle est la rigidité de la structure ?).
- Le volume de travail, défini comme l'ensemble des points atteignables par l'organe terminal. Tous les mouvements ne sont pas possibles en tout point du volume de travail. L'espace de travail (*reachable workspace*), également appelé espace de travail maximal, est le volume de l'espace que le robot peut atteindre *via* au moins une orientation. L'espace de travail dextre (*dextrous<sup>6</sup> workspace*) est le volume de l'espace que le robot peut atteindre avec toutes les orientations possibles de l'effecteur (organe terminal). Cet espace de travail est un sous-ensemble de l'espace de travail maximal.
- Le positionnement absolu, correspondant à l'erreur entre un point souhaité (réel) – défini par une position et une orientation dans l'espace cartésien – et le point atteint et calculé *via* le modèle géométrique inverse du robot. Cette erreur est due au modèle utilisé, à la quantification de la mesure de position, à la flexibilité du système mécanique. En général, l'erreur de positionnement absolu, également appelée précision, est de l'ordre de 1 mm.
- La répétabilité, ce paramètre caractérise la capacité que le robot a à retourner vers un point (position, orientation) donné. La répétabilité correspond à l'erreur maximum de positionnement sur un point prédéfini dans le cas de trajectoires répétitives. En général, la répétabilité est de l'ordre de 0,1 mm.
- La vitesse de déplacement (vitesse maximum en élongation maximum), accélération.
- La masse du robot.
- Le coût du robot.
- La maintenance, ...

## 1.5 Les générations de robot

Des progressions s'opèrent dans tous les domaines :

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

---

<sup>6</sup> *dextrous* signifie adroit, habile.

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

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

## 1.6 Programmation des robots

Classiquement, 2 étapes sont utilisées pour faire en sorte qu'un robot connaisse la tâche à exécuter.

### 1. L'apprentissage :

- Enregistrement dans une mémoire de la trajectoire à exécuter, sous contrôle d'un opérateur humain,
- *Pantin* : Structure mécanique identique à celle du robot, qui est déplacée et qui mémorise les points "pertinents",
- *Syntaxeur* : Un manche de pilotage (*joystick*) commande les déplacements de l'organe terminal,
- *Boîte à boutons* : Un interrupteur par actionneur.

2. La **génération de trajectoires** et les **opérations à réaliser** le long de ces trajectoires, ce qui permet de définir la tâche à réaliser : On fait appel à un logiciel qui, à partir du modèle du robot, et des trajectoires à réaliser, élabore la succession des commandes des actionneurs. Les langages de programmation les plus courants sont : WAVE, VAL (Unimate), LM (Hitachi). Nous utiliserons pour notre part les langages associés au robot Stäubli RX 90 (langage V+) et au robot FANUC ARC ou LR.

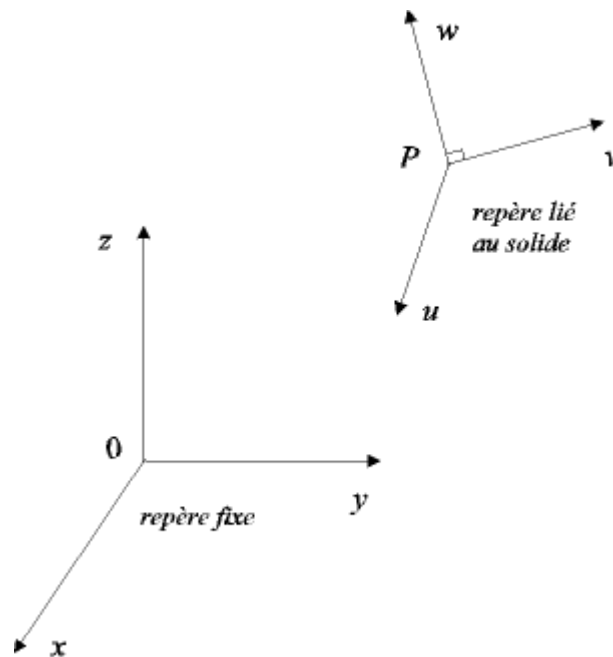
## 2 DEGRÉ DE LIBERTÉS - ARCHITECTURE

### 2.1 Positionnement d'un solide dans l'espace

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

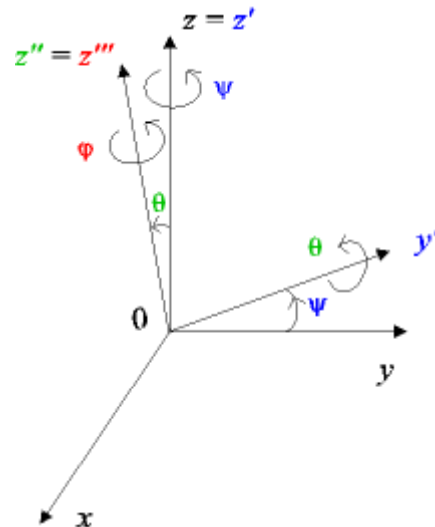
- 3 paramètres indépendants définissent la position d'un point, noté *P*, du solide (coordonnées cartésiennes, cylindriques, sphériques, ..., dans la base du repère fixe),
- 3 paramètres indépendants déterminent l'orientation du solide autour du point *P* (angles d'*Euler*, paramètres d'*Euler*, ...).





**Exemple d'orientation d'un repère** (utilisation des angles d'Euler) :

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



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

- On fait une rotation d'un angle  $\psi$ , appelé *lacet* (*yaw* en anglais), autour de l'axe  $Oz$ ,
- On fait une rotation d'un angle  $\theta$ , appelé *tangage* (*pitch* en anglais), autour de l'axe  $Oy'$ ,
- On fait une rotation d'un angle  $\varphi$ , appelé *roulis* (*roll* en anglais), autour de l'axe  $Oz''$ .

On verra en TP que l'on dispose *via* la console du robot de la situation (position et orientation) du repère terminal (outil), à travers les coordonnées  $X, Y, Z$  du point d'origine du repère terminal (*i.e.*, la flasque) et des angles d'Euler ( $\psi, \theta, \varphi$ ) selon la convention (z, y, z) pour le robot Stäubli RX 90 (*cf.* Manuel Formation V+1 (CS7), p.3-19, ..., 3-22), selon la convention (x, y, z) pour le robot FANUC ARC ou LR (*cf.* Manuel de mise en service du FANUC, p. 3-50).

Un solide peut également être repéré par les coordonnées de 3 points (non situés sur une droite du solide). Il en résulte 9 paramètres (3 coordonnées par point) non indépendants (3 équations expriment l'invariabilité de la distance entre les 3 points). Notons que de par le lien établi par ces équations, il reste 6 paramètres indépendants.

On dit qu'un solide situé dans l'espace possède 6 *degrés de liberté* (*d.d.l.*). Réciproquement, il faut 6 variables de commande indépendantes pour placer de manière quelconque un solide dans l'espace. En pratique, les robots les plus courants sont dotés de 6 *d.d.l.*, *i.e.*, d'au moins 6 actionneurs, ce qui permet de spécifier de manière quelconque la situation (position et orientation) de leurs organes terminaux.

## 2.2 Liaison

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

### Exemples :

- Un cube sur un plan a 3 *d.d.l.* : 2 pour fixer les coordonnées d'un point dans le plan, 1 pour déterminer son orientation dans le plan.
- Une sphère sur un plan a 5 *d.d.l.* : 2 pour fixer les coordonnées d'un point dans le plan, 3 pour déterminer son orientation dans le plan.
- Une porte par rapport au mur a 1 *d.d.l.*

## 2.3 Mécanismes

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

- Les *mécanismes en chaîne simple ouverte* (ou *en série*). Lorsque l'on parcourt le mécanisme, on ne repasse jamais 2 fois sur la même liaison, ou sur le même solide. Ce type de système est le plus répandu.
- Les *mécanismes en chaîne complexe*, *i.e.*, tout ce qui n'est pas *en série* (au moins un solide avec plus de 2 liaisons). De tels systèmes se subdivisent en 2 groupes : les *chaînes structurées en arbre*, et les *chaînes fermées* (dont l'avantage est d'être *a priori* plus rigide, plus précis, capable de manipuler de lourdes charges). A titre d'exemple, le *pantographe*<sup>7</sup> est un *mécanisme en chaîne fermée*.

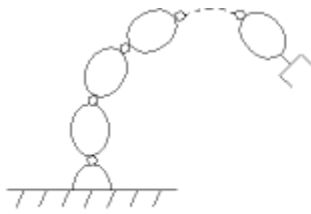
Pour représenter un mécanisme, on dispose de 2 méthodes :

- Le schéma cinématique<sup>8</sup> : On utilise la représentation normalisée des liaisons pour représenter le mécanisme, soit en perspective, soit en projection.
- Le graphe, non normalisé. A titre d'exemples, considérons quelques mécanismes :

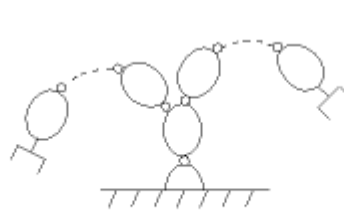
---

<sup>7</sup> Un *pantographe* est un instrument formé de 4 tiges articulées, servant à reproduire mécaniquement un dessin, le cas échéant à une échelle différente.

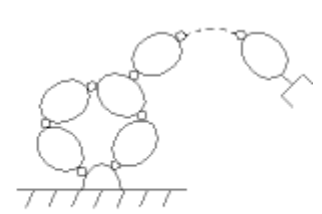
<sup>8</sup> Relatif au mouvement.



*chaîne simple ouverte*



*chaîne structurée en arbre*

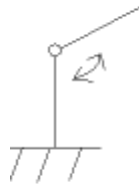


*chaîne fermée*

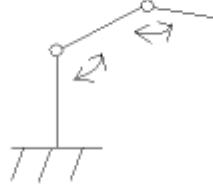
**Définition** (degré de liberté, *d.d.l.*) : Le nombre de *d.d.l.* d'un mécanisme est le nombre de paramètres indépendants qui permettent de définir la position du *mécanisme* à un instant donné du mouvement.

**Exemples :**

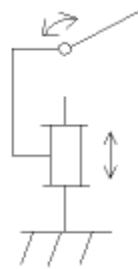
➤ *Chaînes simples ouvertes*



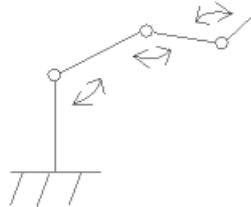
**1 d.d.l.**



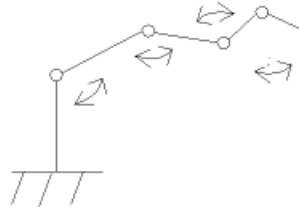
**2 d.d.l.**  
(on peut fixer  $x$  et  $y$ )



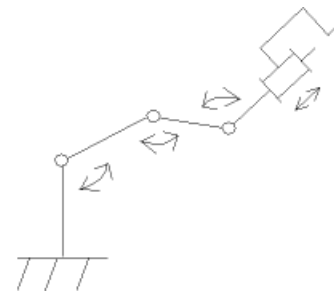
**2 d.d.l.**



**3 d.d.l.**  
(on peut fixer arbitrairement  $x, y, \theta$  (l'orientation de l'outil))

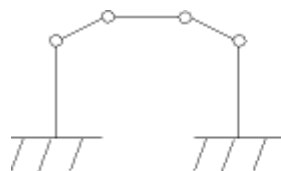


**3 d.d.l.**  
(on peut fixer arbitrairement  $x, y, \theta$ , on a ajouté un degré de *mobilité*)

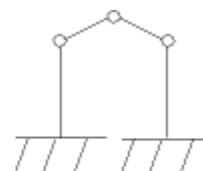


**4 d.d.l.**  
(on peut fixer arbitrairement  $x, y, \theta, l$  (la longueur de l'outil))

➤ *Chaînes complexes*



**1 d.d.l.**



**0 d.d.l.**

**Définition** (redondant) : Un robot est *redondant* lorsque le nombre de *d.d.l.* du mécanisme est inférieur au nombre d'articulations indépendantes (motorisées). Cette propriété permet de préserver les capacités

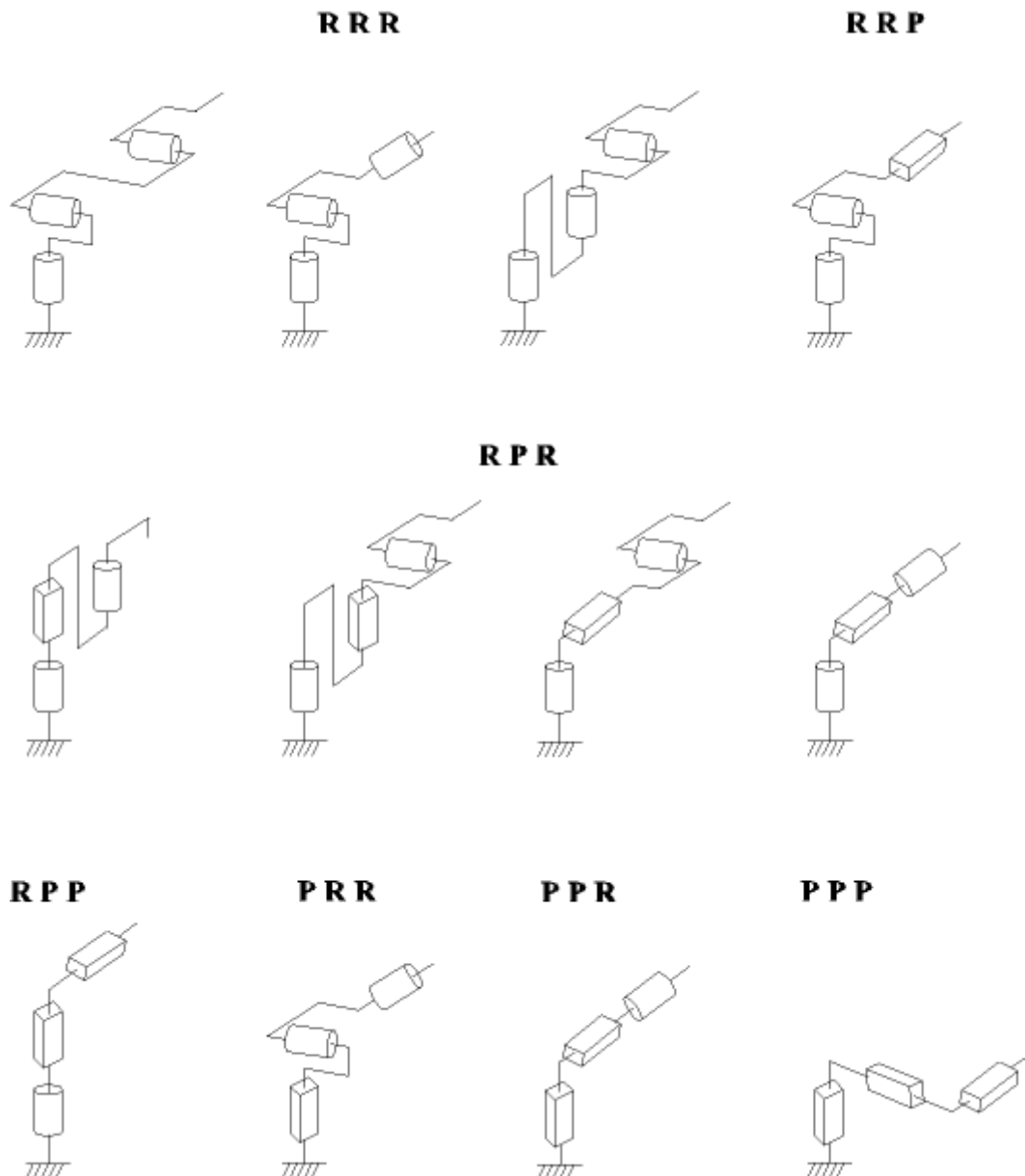
de déplacement de l'organe terminal en présence d'obstacles, le (ou les) *d.d.l.* supplémentaire(s) autorisant leur contournement.

## 2.4 Morphologie des robots manipulateurs

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

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

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

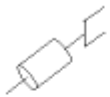


Dans la pratique, on trouve les 5 structures suivantes :

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

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

**poignet à 1 axe**



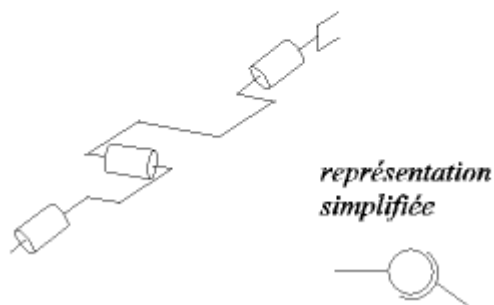
**poignet à 2 axes concourants**



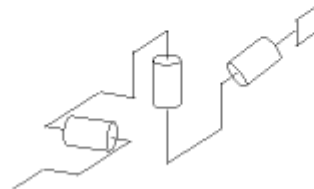
**poignet à 2 axes non concourants**



**poignet à 3 axes concourants (rotule)**



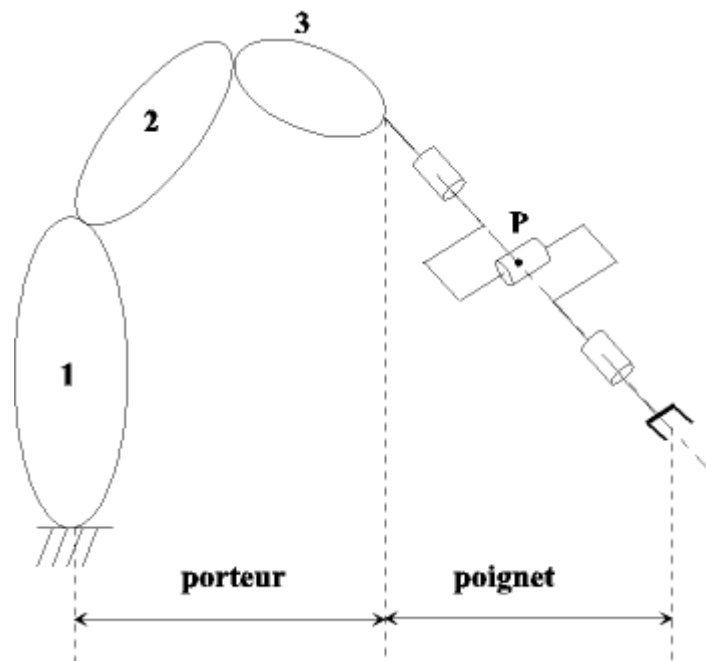
**poignet à 3 axes non concourants**



Dans la pratique, le *poignet* de type rotule est très répandu. Le robot, obtenu en lui associant un porteur à 3 *d.d.l.*, est la structure la plus classique à 6 *d.d.l.*. Elle permet d'assurer un découplage entre la position et l'orientation de l'organe terminal :

- Le porteur a pour rôle de fixer la position du point d'intersection, noté  $P$ , des axes des 3 dernières articulations (centre du poignet) ; cette position ( $P$ ) ne dépend que de la configuration des solides (corps) 1, 2 et 3 (*i.e.*, du porteur),
- Le poignet est destiné à l'orientation de l'organe terminal (pince, outil).

Voir la figure suivante.



Cette structure de robot à poignet de type rotule correspond à celle des robots Stäubli RX-90 et FANUC ARC ou LR que nous considérerons en TP.

### 3 MODÈLE GÉOMÉTRIQUE D'UN ROBOT EN CHAÎNE SIMPLE

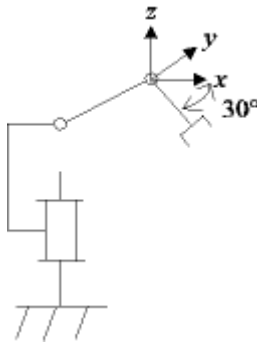
#### 3.1 Nécessité d'un modèle

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

- Les modèles de transformation entre l'*espace opérationnel* (dans lequel est définie la situation de l'organe terminal) et l'*espace articulaire* (dans lequel est définie la configuration du robot). Parmi ces modèles, on distingue :
  - Les modèles *géométriques direct* et *inverse* qui expriment la situation de l'organe terminal en fonction de la configuration du mécanisme et inversement,
  - Les modèles *cinématiques direct* et *inverse* qui expriment la vitesse de l'organe terminal en fonction de la vitesse articulaire et inversement,
- Les modèles *dynamiques* définissant les équations du mouvement du robot, qui permettent d'établir les relations entre les couples ou forces exercés par les actionneurs et les positions, vitesses et accélérations des articulations.

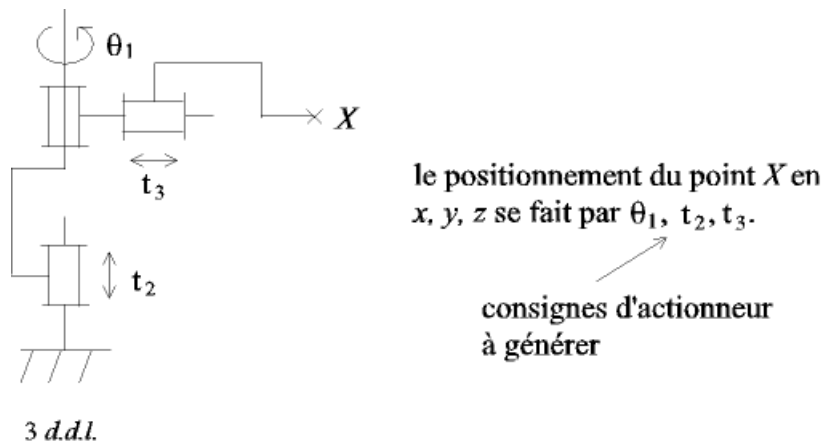
Définir les différentes tâches d'un robot réclame de pouvoir positionner l'organe terminal par rapport à un repère de référence. En effet :

- Les informations proprioceptives (issues du S.M.A.) sont généralement définies dans des repères liés aux différents solides du robot.



- La position à atteindre est souvent définie dans un repère lié au socle du robot,
- L'objet à saisir peut être défini dans un repère mobile indépendant du robot (par exemple, des pièces à prendre sur un tapis roulant),
- Les informations extéroceptives (issues de l'environnement) sont définies dans divers repères.

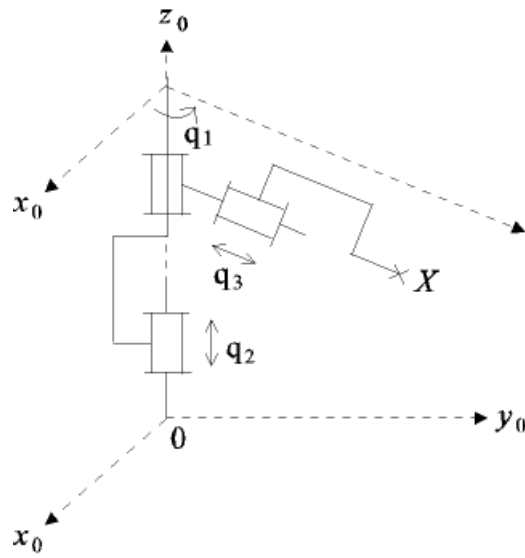
Aussi, il faut un référentiel commun afin de "ramener" les diverses informations dans un même référentiel, notamment pour concevoir les consignes des actionneurs.



On a une relation matricielle du type :  $\underline{X} = f(\underline{q})$  avec  $\underline{X} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ ,  $\underline{q} = \begin{pmatrix} q_1 \\ q_2 \\ q_3 \end{pmatrix}$ ,

où  $f$  est une fonction vectorielle statique (la variable temporelle n'intervient pas). Dans le cas de l'exemple précédent, on a :  $q_1 = \theta_1, q_2 = t_2, q_3 = t_3$ .

**Exercice :**



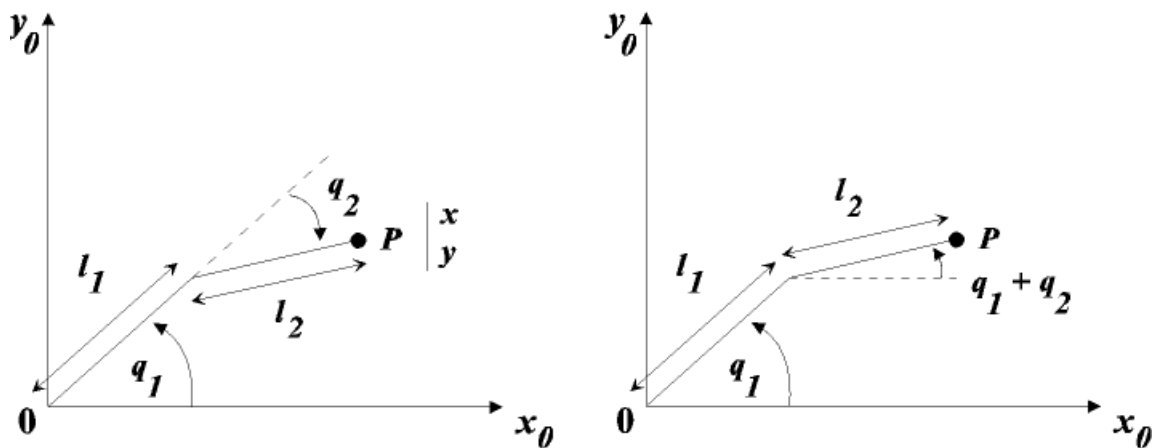
**Question :** Ecrire le **modèle géométrique direct** du manipulateur, soit la relation :  $\underline{X} = f(\underline{q})$ .

Ecriture du **modèle géométrique inverse** : Connaissant la position du point  $X$  (coordonnée opérationnelle), on veut connaître les variables articulaires  $\underline{q}$  (actionneurs, coordonnée articulaire). Pour connaître le vecteur  $\underline{q}$  conduisant à une position donnée du point  $X$ , il faut disposer d'une relation du type :

$$\underline{q} = g(\underline{X}),$$

correspondant au **modèle géométrique inverse** (on a  $g = f^{-1}$ ).

**Exercice :** Soit le manipulateur RR plan décrit dans la figure suivante.



- 1) Etablir le modèle géométrique direct.
- 2) Réaliser un programme (MatLab, Scilab) permettant de représenter l'espace de travail maximal (espace atteignable à travers au moins une orientation) du robot sachant que  $l_1 = l_2 = 10 \text{ cm}$  et  $0 \leq q_1 \leq 90^\circ$ ,  $-100^\circ \leq q_2 \leq 90^\circ$ .

### 3.2 Coordonnées opérationnelles



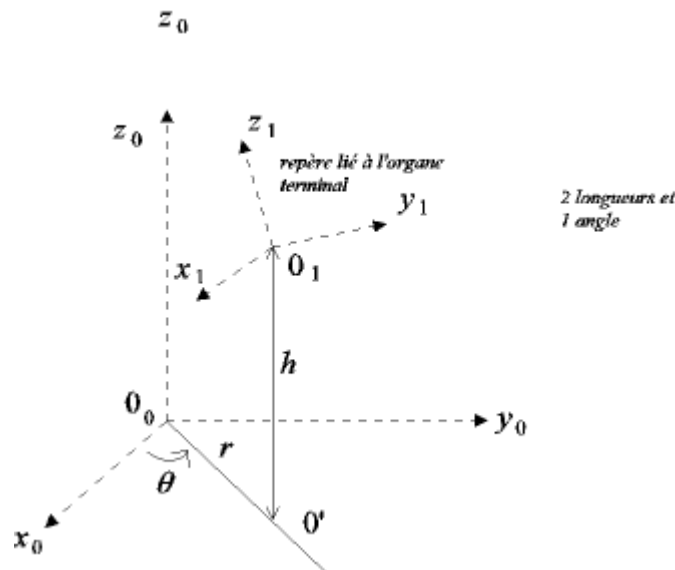
Plusieurs possibilités existent pour la définition du vecteur  $\underline{X}$  selon les méthodes utilisées pour spécifier la *position* et l'*orientation*. Par exemple, si l'orientation est spécifiée à l'aide des *cosinus directeurs* (décrits ci-dessous), on aura :

$$\underline{X} = \left[ \begin{array}{ccc|cccccccccc} M_x & M_y & M_z & a_{11} & a_{21} & a_{31} & a_{12} & a_{22} & a_{32} & a_{13} & a_{23} & a_{33} \end{array} \right].$$

voir a)
voir b)

a) En général, on définit la *position* d'un point du repère lié à l'organe terminal *via* des coordonnées cartésiennes (3 longueurs), cylindriques (2 longueurs + 1 angle) ou sphériques (1 longueur + 2 angles).

**Exemple des coordonnées cylindriques :**



b) Il existe différentes possibilités pour définir l'*orientation* de l'organe terminal :

- Les angles d'Euler,
- D'autres types d'angles (Bryant, ...),
- Les *cosinus directeurs*. Leur utilisation revient à considérer l'ensemble des projections des vecteurs unités portés par les axes du repère  $R_1$ , sur les axes du repère  $R_0$  (repères orthonormés). Il en résulte  $3 \times 3$  paramètres, en effet :
  - 6 relations sont nécessaires pour indiquer que la base est orthonormée (3 pour indiquer des normes unitaires + 3 pour indiquer l'orthogonalité de la base),
  - et 3 paramètres pour décrire l'orientation du repère.

**Construction des *cosinus directeurs* :**

Repère initial  $\begin{matrix} \vec{x}_0 \\ \vec{y}_0 \\ \vec{z}_0 \end{matrix} \begin{pmatrix} \vec{x}_1 & \vec{y}_1 & \vec{z}_1 \\ a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$ . Cette matrice de rotation vérifie 6 relations parmi les 9 paramètres (à savoir  $\|\vec{x}_1\| = \|\vec{y}_1\| = \|\vec{z}_1\| = 1$ ,  $\vec{x}_1 \cdot \vec{y}_1 = \vec{x}_1 \cdot \vec{z}_1 = \vec{y}_1 \cdot \vec{z}_1 = 0$ <sup>9</sup>).

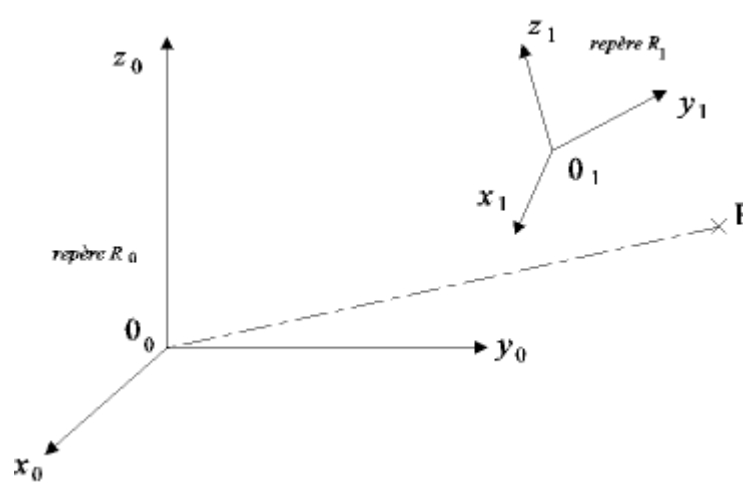
Le vecteur de base  $\vec{x}_1$  de la base  $R_1$  s'exprime dans la base  $R_0$  par la relation :

$\vec{x}_1 = a_{11} \vec{x}_0 + a_{21} \vec{y}_0 + a_{31} \vec{z}_0$ , le vecteur  $\begin{pmatrix} a_{11} \\ a_{21} \\ a_{31} \end{pmatrix}$  représente le vecteur unitaire  $\vec{x}_1$  (du repère  $R_1$ )

suivant les axes  $\vec{x}_0, \vec{y}_0, \vec{z}_0$  (du repère  $R_0$ ).

### 3.3 Translation et rotation

On peut montrer que la position du repère de base  $R_1$  se déduit de celle du repère de base  $R_0$  par une translation et une rotation.



Le but est d'exprimer dans le repère de base  $R_0$  les coordonnées du point  $P$  ayant pour coordonnées  $(X_1, Y_1, Z_1)$  dans le repère de base  $R_1$  (c-à-d :  $\overrightarrow{O_1 P}_{/1} = (X_1 \ Y_1 \ Z_1)^t$ ), sachant que l'origine du repère  $R_1$ , c'est-à-dire  $O_1$ , a pour coordonnées  $(a, b, c)$  dans le repère  $R_0$ .

On a :  $\overrightarrow{O_1 P}_{/1} = \begin{pmatrix} X_1 \\ Y_1 \\ Z_1 \end{pmatrix}$ , c'est-à-dire  $\overrightarrow{O_1 P} = X_1 \vec{x}_1 + Y_1 \vec{y}_1 + Z_1 \vec{z}_1$ .

<sup>9</sup>  $\|\vec{x}_1\|^2 = a_{11}^2 + a_{21}^2 + a_{31}^2$  et  $\vec{x}_1 \cdot \vec{x}_2 = a_{11} a_{12} + a_{21} a_{22} + a_{31} a_{32}$ .

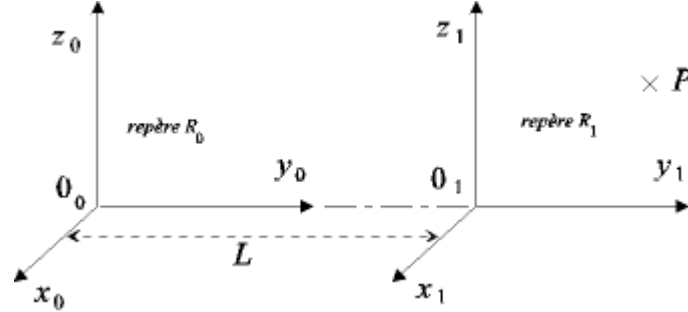
$$\begin{aligned}
\text{On a : } \overrightarrow{O_0 P_{/0}} &= \overrightarrow{O_0 O_{1/0}} + \overrightarrow{O_1 P_{/0}} \\
&= \overrightarrow{O_0 O_{1/0}} + R_{01} \times \overrightarrow{O_1 P_{/1}} \\
&= \begin{pmatrix} a \\ b \\ c \end{pmatrix} + \begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} \times \begin{pmatrix} X_1 \\ Y_1 \\ Z_1 \end{pmatrix}.
\end{aligned}$$

La matrice de rotation, notée  $R_{01}$ ,  $\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix}$  contient les vecteurs de base de  $R_1$ , c'est-à-dire  $\vec{x}_1, \vec{y}_1, \vec{z}_1$ , exprimés dans la base  $R_0$ , c'est-à-dire en fonction des vecteurs  $\vec{x}_0, \vec{y}_0, \vec{z}_0$ .

Par exemple, la position du repère de base  $R_1$  pourra se déduire de celle du repère de base  $R_0$  à travers :

- une translation  $\overrightarrow{O_0 O_1}$ ,
- des rotations autour des 3 axes  $(\vec{x}_0, \vec{y}_0, \vec{z}_0)$  du repère  $R_0$ , auxquelles on peut associer respectivement les matrices élémentaires de rotations suivantes :  $R_{01}(x_0, \theta_{x_0})$ ,  $R_{01}(y_0, \theta_{y_0})$  et  $R_{01}(z_0, \theta_{z_0})$ , définies plus loin dans cette section. On a alors :  
 $R_{01} = R_{01}(x_0, \theta_{x_0}) \times R_{01}(y_0, \theta_{y_0}) \times R_{01}(z_0, \theta_{z_0})$ .

➤ *Cas d'une simple translation*



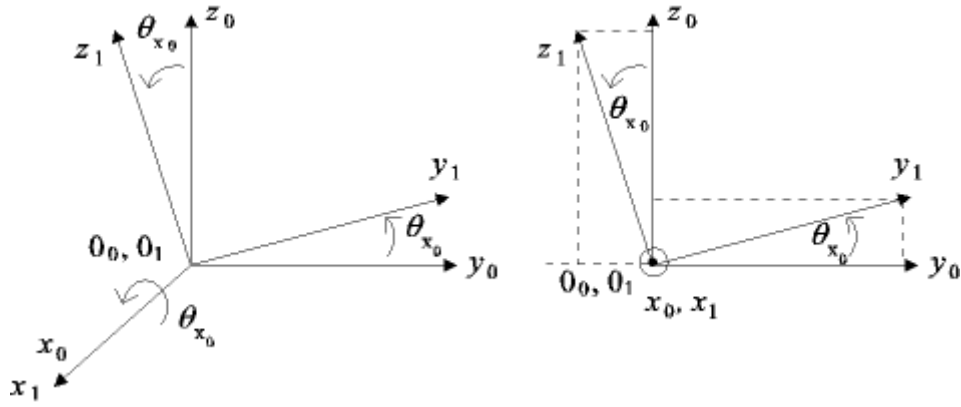
$$\begin{aligned}
\text{On a : } \overrightarrow{O_0 P_{/0}} &= \overrightarrow{O_0 O_{1/0}} + R_{01} \times \overrightarrow{O_1 P_{/1}} \\
&= \begin{pmatrix} 0 \\ L \\ 0 \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} X_1 \\ Y_1 \\ Z_1 \end{pmatrix} = \begin{pmatrix} X_1 \\ L + Y_1 \\ Z_1 \end{pmatrix}.
\end{aligned}$$

La matrice (vecteur) de translation opère selon l'axe  $\vec{y}_0$ .

La matrice de rotation (d'angle nul) est telle que :  $\vec{x}_1 = \vec{x}_0, \vec{y}_1 = \vec{y}_0, \vec{z}_1 = \vec{z}_0$ .

➤ *Matrices de rotation : Cas particulier de rotation autour d'un axe*

Par convention, on considère que l'articulation rotoïde d'un robot se fait autour d'un des 3 axes d'un repère orthonormé. Soit, par exemple, une rotation d'angle  $\theta_{x_0}$  autour de l'axe  $\vec{x}_0$ .



On a :  $R_{01}(x_0, \theta_{x_0}) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_{x_0}) & -\sin(\theta_{x_0}) \\ 0 & \sin(\theta_{x_0}) & \cos(\theta_{x_0}) \end{pmatrix},$

soient  $\vec{x}_1 = \vec{x}_0$ ,  $\vec{y}_1 = \cos(\theta_{x_0}) \vec{y}_0 + \sin(\theta_{x_0}) \vec{z}_0$ ,  $\vec{z}_1 = -\sin(\theta_{x_0}) \vec{y}_0 + \cos(\theta_{x_0}) \vec{z}_0$ .

**Remarque :** On a  $\|\vec{x}_1\| = \|\vec{y}_1\| = \|\vec{z}_1\| = 1$ ,  $\vec{x}_1 \cdot \vec{y}_1 = \vec{x}_1 \cdot \vec{z}_1 = \vec{y}_1 \cdot \vec{z}_1 = 0$ .

➤ *Exemple d'une translation et d'une rotation autour de l'axe  $\vec{x}_0$*

Exprimons dans le repère  $R_0$  les coordonnées  $(X_1, Y_1, Z_1)$  du point  $P$  dans le repère  $R_1$ , sachant que l'origine  $O_1$  du repère  $R_1$  a pour coordonnées  $(a, b, c)$  dans le repère  $R_0$ .

On a :  $\vec{O_0 P_{/0}} = \vec{O_0 O_{1/0}} + \vec{O_1 P_{/0}}$   
 $= \vec{O_0 O_{1/0}} + R_{01} \times \vec{O_1 P_{/1}}$   
 $= \begin{pmatrix} a \\ b \\ c \end{pmatrix} + \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos(\theta_{x_0}) & -\sin(\theta_{x_0}) \\ 0 & \sin(\theta_{x_0}) & \cos(\theta_{x_0}) \end{pmatrix} \times \begin{pmatrix} X_1 \\ Y_1 \\ Z_1 \end{pmatrix}$   
 $= \begin{pmatrix} a + X_1 \\ b + \cos(\theta_{x_0}) Y_1 - \sin(\theta_{x_0}) Z_1 \\ c + \sin(\theta_{x_0}) Y_1 + \cos(\theta_{x_0}) Z_1 \end{pmatrix}.$

**Exercice :** Exprimer les matrices de rotations  $R_{01}(y_0, \theta_{y_0})$  et  $R_{01}(z_0, \theta_{z_0})$ .

**Remarques :**

- Le produit de matrice de rotation n'est pas toujours commutatif :  $R_{12} \times R_{23} \neq R_{23} \times R_{12}$ .
- L'inverse de  $R_{12}$  est donné par  $R_{12}^t$  :  $R_{12}^{-1} = R_{21} = R_{12}^t$  (i.e.,  $R_{12}$  est une matrice orthogonale<sup>10</sup>).
- La combinaison de translation est commutative :  
 $(Translation(x, a) Translation(y, b) = Translation(y, b) Translation(x, a))$ , l'inversion est obtenue par simple changement de signe :  $(Translation(x, a))^{-1} = Translation(x, -a)$ .

<sup>10</sup> Une matrice  $A$  est **orthogonale** si la matrice inverse est égale à la matrice transposée :  $A \times A^t = A^t \times A = Id$ . Une matrice  $A$  est orthogonale si, et seulement si, les vecteurs lignes sont orthonormaux.

### 3.4 Matrice de transformation homogène

La présence conjointe de produits et de sommes dans l'équation vectorielle  $\overrightarrow{O_0 P_{/0}} = \overrightarrow{O_0 O_{1/0}} + R_{01} \times \overrightarrow{O_1 P_{/1}}$  est peu commode pour effectuer des calculs systématiques, dus par exemple à des changements successifs de repères. On lui préfère une représentation matricielle de dimension 4, basée sur les coordonnées homogènes.

La représentation en coordonnées homogènes consiste à doter toute notation vectorielle d'un facteur d'échelle en introduisant une coordonnée supplémentaire. Soit par exemple, un point  $M$  de l'espace, rapporté à trois axes rectangulaires, donné par la relation :

$$\overrightarrow{OM} = \begin{pmatrix} a \\ b \\ c \end{pmatrix},$$

alors la représentation du point  $M$  à l'aide de coordonnées homogènes est faite avec un quaternion, *i.e.*,

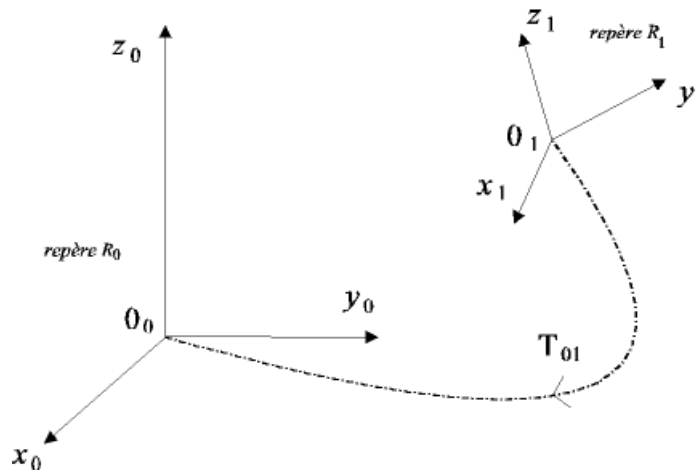
$$\overrightarrow{OM} = \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix}, \text{ avec } a = \frac{x}{w}, b = \frac{y}{w}, c = \frac{z}{w}.$$

Soit la matrice de transformation homogène  $T_{01}$  correspondant à la matrice partitionnée suivante :

$$T_{01} = \begin{pmatrix} R_{01(3,3)} & t_{01(3,1)} \\ 0 & 1 \end{pmatrix}. \quad \begin{array}{l} \text{rotation} \\ \text{translation} \end{array}$$

Remarquons que le facteur d'échelle est unitaire ( $w = 1$ ).

La matrice  $T_{01}$  représente la transformation permettant de passer du repère  $R_1$  au repère  $R_0$ .



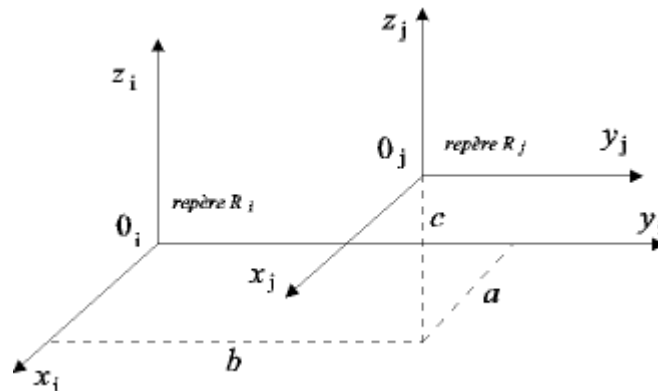
En effet, elle permet d'exprimer dans le repère  $R_0$  les coordonnées d'un vecteur exprimées dans le repère  $R_1$ . Autrement dit, on a :

$$\begin{pmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{pmatrix} = T_{01} \begin{pmatrix} X_1 \\ Y_1 \\ Z_1 \\ 1 \end{pmatrix}.$$

➤ Cas de transformations homogènes

- D'une simple translation

On note  $Trans(x, a)$  la matrice de transformation homogène correspondant à une translation de  $a$  selon l'axe  $\vec{x}$ .



$$\text{On a : } T_{ij} = \begin{pmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

$$\text{On a : } T_{ij} = Trans(x_i, a) \times Trans(y_i, b) \times Trans(z_i, c)$$

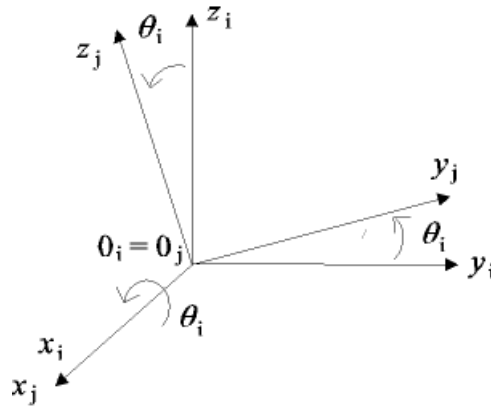
$$= \begin{pmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

Soit  $\begin{pmatrix} x \\ y \\ z \end{pmatrix}$  les coordonnées d'un point  $M$  dans le repère  $R_j$  (c-à-d :  $\overrightarrow{O_j M_{/j}} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ ), alors les

$$\text{coordonnées du point } M \text{ dans le repère } R_i \text{ sont : } T_{ij} \times M_{/j} = \begin{pmatrix} 1 & 0 & 0 & a \\ 0 & 1 & 0 & b \\ 0 & 0 & 1 & c \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x+a \\ y+b \\ z+c \\ 1 \end{pmatrix}.$$

- D'une simple rotation

On note  $Rot(x, \theta)$  la matrice de rotation d'un angle  $\theta$  autour de l'axe  $\vec{x}$ .



On a :  $T_{ij} = Rot(x_i, \theta_i) = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta_i) & -\sin(\theta_i) & 0 \\ 0 & \sin(\theta_i) & \cos(\theta_i) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} R_{ij}(x_i, \theta_i) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}.$

Soit  $\begin{pmatrix} x \\ y \\ z \end{pmatrix}$  les coordonnées d'un point  $M$  dans le repère  $R_j$ , alors les coordonnées du point  $M$  dans

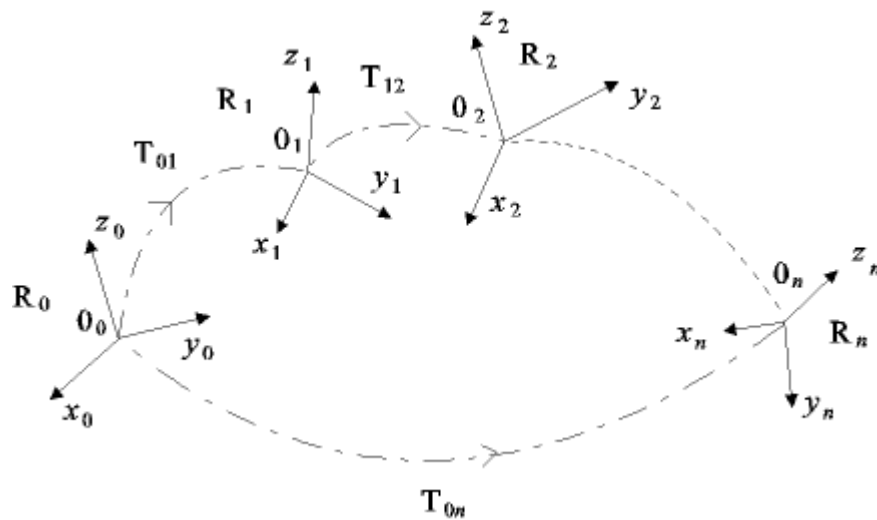
le repère  $R_i$  sont :  $T_{ij} \times M_{/j} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\theta_i) & -\sin(\theta_i) & 0 \\ 0 & \sin(\theta_i) & \cos(\theta_i) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ \cos(\theta_i) y - \sin(\theta_i) z \\ \sin(\theta_i) y + \cos(\theta_i) z \\ 1 \end{pmatrix}.$

Une matrice de transformation peut se décomposer en 2 matrices de transformation :

$$T = \begin{pmatrix} A_{(3,3)} & t_{(3,1)} \\ 0 & 1 \end{pmatrix} = \begin{pmatrix} I & t_{(3,1)} \\ 0 & 1 \end{pmatrix} \times \begin{pmatrix} 0 \\ A_{(3,3)} \\ 0 \\ 1 \end{pmatrix}.$$

*translation      rotation*

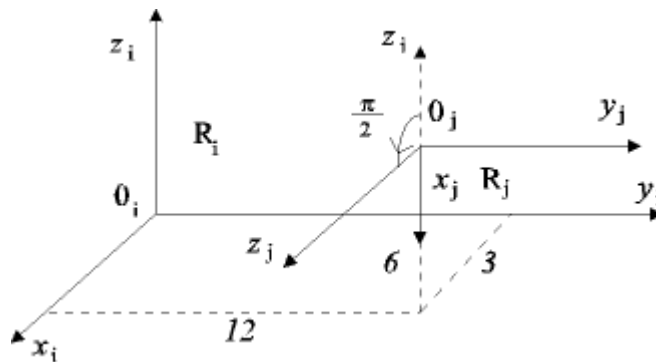
**Remarque** : Le produit de matrices de transformation homogène n'est pas commutatif, du fait de la non commutativité de la rotation.



On a :  $T_{0n} = T_{01} \times T_{12} \times \dots \times T_{n-1, n}$ .

Soit  $\overrightarrow{O_n M}_{/n} = \begin{pmatrix} x \\ y \\ z \end{pmatrix}$ , alors  $T_{0n} \times \begin{pmatrix} x \\ y \\ z \end{pmatrix}$  exprime les coordonnées du point  $M$  dans le repère  $R_0$  (c-à-d :  $\overrightarrow{O_0 M}_{/0} = T_{0n} \times \overrightarrow{O_n M}_{/n}$ ).

**Exercice :** Calculer la matrice de transformation homogène  $T_{ij}$  correspondant au changement de repères suivant :



### 3.5 Obtention du modèle géométrique direct

On s'intéresse à la relation liant l'*espace articulaire* associé aux différentes liaisons du robot avec l'*espace opérationnel* dans lequel est définie la situation de l'organe terminal. Cette relation s'exprime à l'aide de l'équation suivante :

$$\underline{X} = f(\underline{q}). \quad (\text{cf. §3.1})$$

La méthode proposée utilise les matrices de transformation homogène. On associe un repère à chaque solide du robot, en commençant par le socle. Si une articulation a plusieurs degrés de libertés (*d.d.l.*), on introduit des solides fictifs (de masse et de longueur nulles). La situation de l'organe terminal par rapport au socle correspond au produit des matrices de transformation homogène des différents repères associés au solide du robot. Notons que l'écriture des matrices de transformation homogène n'est pas unique (il existe une infinité de façon de lier une repère à un solide).



### 3.6 Paramètres de Denavit-Hartenberg modifié

Les paramètres de Denavit-Hartenberg modifié<sup>11</sup> permettent de disposer d'un paramétrage des liaisons tel que les matrices de passage aient toutes la même forme littérale, ce qui facilite les calculs.

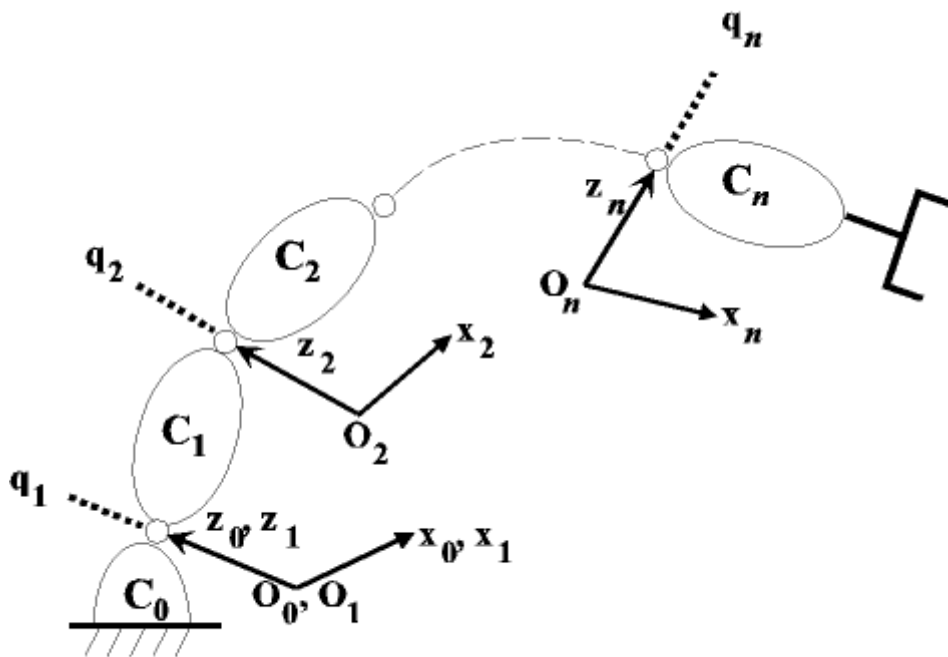
La méthode qui suit s'applique lorsque le robot correspond à une chaîne simple ouverte et que ses articulations sont rotoïdes, ou prismatiques (ce qui est le cas en général). Les corps constituant le robot sont supposés parfaitement rigides et connectés par des articulations idéales (pas de jeu mécanique, pas d'élasticité).

#### ➤ Notations :

On numérote les solides par ordre croissant en partant du socle. Ainsi le robot est composé de  $n + 1$  corps, notés  $C_0, \dots, C_n$ , et de  $n$  articulations ( $n \geq 1$ ). Le corps  $C_0$  désigne le socle (la base) du robot, le corps  $C_n$  le corps portant l'organe terminal.

Le repère  $R_j$  est lié au corps  $C_j$  du robot.

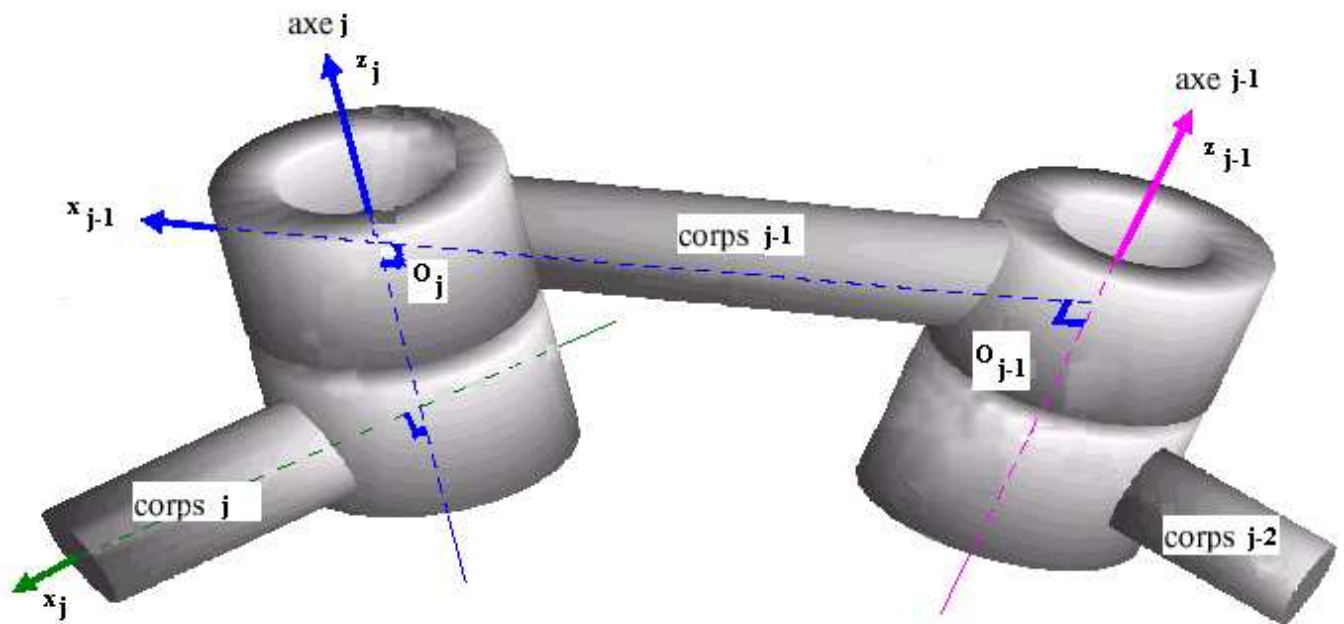
La variable de l'articulation  $j$ , qui lie le corps  $C_j$  au corps  $C_{j+1}$ , est notée  $q_j$ .



#### ➤ Détermination du repère $R_j$ (lié au corps $C_j$ ) :

- L'axe  $\overrightarrow{o_j z_j}$  est porté par l'axe de rotation, ou de translation, de l'articulation  $j$ .
- L'axe  $\overrightarrow{o_j x_j}$  est porté par la perpendiculaire commune aux axes  $\overrightarrow{o_j z_j}$  et  $\overrightarrow{o_{j+1} z_{j+1}}$ . Si les axes  $\overrightarrow{o_j z_j}$  et  $\overrightarrow{o_{j+1} z_{j+1}}$  sont parallèles, le choix de  $\overrightarrow{o_j x_j}$  n'est pas unique, il est alors dicté par des considérations de symétrie ou de simplicité.

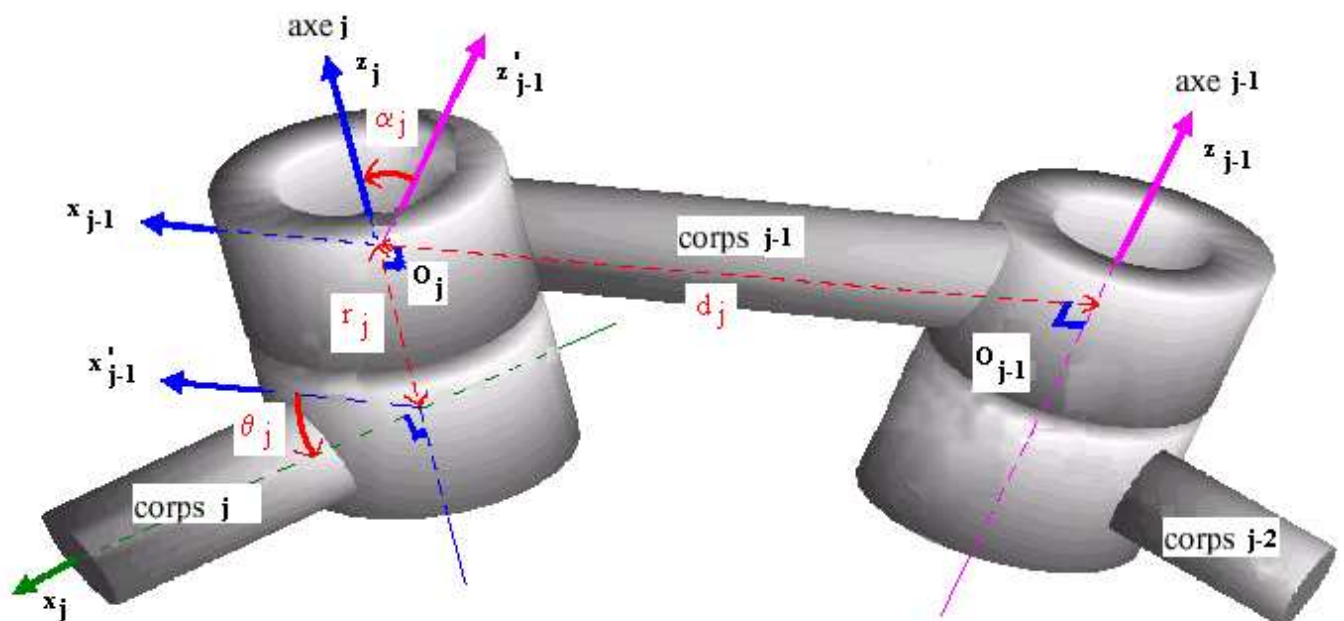
<sup>11</sup> La convention de Denavit-Hartenberg (non modifié) est telle que l'axe  $z_j$  du repère  $R_j$  est confondu avec l'axe n°  $j + 1$  (et non  $j$ ) du robot.



➤ **Passage du repère  $R_{j-1}$  au repère  $R_j$ , détermination des paramètres de Denavit-Hartenberg modifié :**

Le passage du repère  $R_{j-1}$  au repère  $R_j$  s'exprime en fonction des 4 paramètres suivants :

- On note  $\alpha_j$  l'angle de rotation entre les axes  $\vec{z}_{j-1}$  et  $\vec{z}_j$  autour de l'axe  $\vec{x}_{j-1}$ ,
- On note  $d_j$  la distance entre  $\vec{z}_{j-1}$  et  $\vec{z}_j$  mesurée le long de l'axe  $\vec{x}_{j-1}$ ,
- On note  $\theta_j$  l'angle de rotation entre les axes  $\vec{x}_{j-1}$  et  $\vec{x}_j$  autour de l'axe  $\vec{z}_j$ ,
- On note  $r_j$  la distance entre  $\vec{x}_{j-1}$  et  $\vec{x}_j$  mesurée le long de l'axe  $\vec{z}_j$ .



**Remarque :** La variable articulaire  $q_j$ , associée à l'articulation  $j$ , est soit  $\theta_j$ , soit  $r_j$ , selon que cette articulation est respectivement de type rotoïde ou prismatique, ce qui se traduit par la relation :

$$q_j = (1 - \sigma_j) \theta_j + \sigma_j r_j$$

avec  $\sigma_j = 0$  si l'articulation  $j$  est rotoïde et  $\sigma_j = 1$  si elle est prismatique.

Autrement dit, si l'articulation est une *rotation* alors  $\begin{cases} q_j = \theta_j \text{ est variable} \\ \alpha_j, d_j, r_j \text{ sont constants} \end{cases}$ .

Si l'articulation est une *translation* alors  $\begin{cases} q_j = r_j \text{ est variable} \\ \alpha_j, d_j, \theta_j \text{ sont constants} \end{cases}$ .

Bien souvent, la rotation d'angle  $\alpha_j$  est multiple de  $\frac{\pi}{2}$ .

Le passage du repère  $R_{j-1}$  au repère  $R_j$  se fait à travers 4 étapes :

- Une rotation d'angle  $\alpha_j$  autour de l'axe  $\overrightarrow{x_{j-1}}$ ,
- Une translation de distance  $d_j$  mesurée le long de l'axe  $\overrightarrow{x_{j-1}}$ ,
- Une rotation d'angle  $\theta_j$  autour de l'axe  $\overrightarrow{z_j}$ ,
- Une translation de distance  $r_j$  mesurée le long de l'axe  $\overrightarrow{z_j}$ .

Il en résulte la matrice de transformation homogène  $T_{j-1,j}$  suivante :

$$\begin{aligned} T_{j-1,j} &= Rot(x_{j-1}, \alpha_j) \times Trans(x_{j-1}, d_j) \times Rot(z_j, \theta_j) \times Trans(z_j, r_j) \\ &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\alpha_j) & -\sin(\alpha_j) & 0 \\ 0 & \sin(\alpha_j) & \cos(\alpha_j) & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 & d_j \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} \cos(\theta_j) & -\sin(\theta_j) & 0 & 0 \\ \sin(\theta_j) & \cos(\theta_j) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \times \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & r_j \\ 0 & 0 & 0 & 1 \end{pmatrix} \\ &= \begin{pmatrix} \cos(\theta_j) & -\sin(\theta_j) & 0 & d_j \\ \cos(\alpha_j) \sin(\theta_j) & \cos(\alpha_j) \cos(\theta_j) & -\sin(\alpha_j) & -r_j \sin(\alpha_j) \\ \sin(\alpha_j) \sin(\theta_j) & \sin(\alpha_j) \cos(\theta_j) & \cos(\alpha_j) & r_j \cos(\alpha_j) \\ 0 & 0 & 0 & 1 \end{pmatrix}. \end{aligned}$$

**Remarques :**

- Pour la définition du repère de référence  $R_0$ , le choix le plus simple consiste à prendre  $R_0$  confondu avec le repère  $R_1$  quand  $q_1 = 0$  (voir §3.6, 1<sup>er</sup> figure). Ceci signifie que  $z_0$  est confondu avec  $z_1$ , de plus  $O_0 = O_1$  lorsque l'articulation 1 est rotoïde, et  $x_0$  est parallèle à  $x_1$  lorsque l'articulation 1 est prismatique.
- Pour une articulation  $j$  prismatique, l'axe  $z_j$  est parallèle à l'axe de l'articulation mais la position de cet axe dans l'espace peut être quelconque.
- Bien souvent, les deux axes de liaisons consécutifs d'un robot sont orthogonaux ou parallèles, il en résulte un angle  $\alpha$  égale à  $0^\circ, \pm 90^\circ, 180^\circ$ .
- La transformation inverse est obtenue par la formule suivante :

$$T_{j,j-1} = Trans(z_j, -r_j) \times Rot(z_j, -\theta_j) \times Trans(x_{j-1}, -d_j) \times Rot(x_{j-1}, -\alpha_j),$$

$$\text{soit } T_{j,j-1} = \begin{pmatrix} & & -d_j \cos(\theta_j) \\ & A_{j-1,j}^t & d_j \sin(\theta_j) \\ & & -r_j \\ 0 & 0 & 0 & 1 \end{pmatrix}.$$

On a :  $A_{j,j-1} = A_{j-1,j}^{-1} = A_{j-1,j}^t$ , cf. Remarques situées à la fin de §3.3.

### Obtention des modèles géométriques direct et inverse

Le modèle géométrique direct est l'ensemble des relations qui permettent d'exprimer la situation de l'organe terminal, les coordonnées opérationnelles, du robot en fonction de ses coordonnées articulaires (cf. §3.5). Dans la cas d'une chaîne simple ouverte, il peut être représenté par la matrice  $T_{0,n}$  qui se calcule par :

$$T_{0,n} = T_{0,1}(q_1) \times T_{1,2}(q_2) \times \dots \times T_{n-1,n}(q_n).$$

Le modèle géométrique direct du robot peut se représenter aussi par la relation :

$$\underline{X} = f(\underline{q}),$$

où  $\underline{X}$  est le vecteur des coordonnées opérationnelles exprimées dans le repère de référence  $R_0$ , et  $\underline{q}$  les variables articulaires.

La matrice  $T_{0,n}$  représente la position et l'orientation, exprimées dans le repère de référence  $R_0$ , de l'organe terminal du robot.

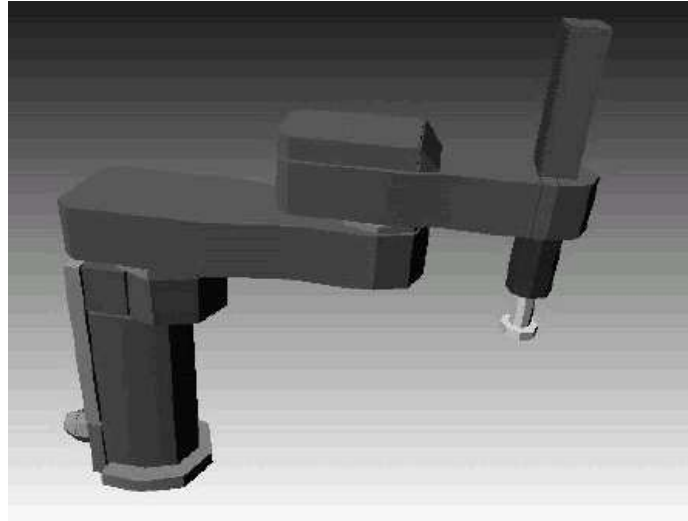
Le modèle géométrique inverse est le problème inverse qui permet de connaître les variables articulaires en fonction de la situation de l'organe terminal, ce qui peut se représenter par la relation :

$$\underline{q} = g(\underline{X}).$$

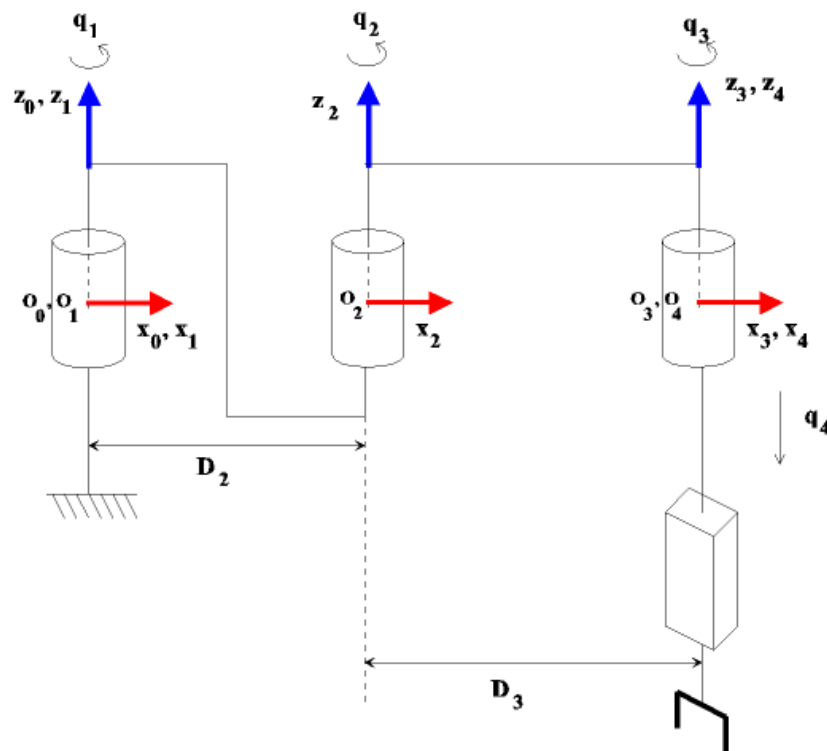
### 3.7 Exemple

On se propose d'établir le modèle géométrique direct du robot SCARA<sup>12</sup> à 4 degrés de libertés représenté, dans sa configuration initiale, dans la figure suivante :

<sup>12</sup> SCARA : Selective Compliance Articulated Robot for Assembly (compliance = conforme).<sup>13</sup> Paul R.C.P., *Robot Manipulators: Mathematics, Programming and Control*, MIT Press, Cambridge, USA, 1981.



et de façon schématique dans la figure suivante. Afin de se faciliter la tâche, les différents repères permettant d'établir les paramètres de Denavit-Hartenberg modifié sont représentés.



De façon pratique, il s'agit de placer :

- Les axes  $z_j$  sur les axes des différentes articulations,
- Puis, les axes  $x_j$  selon les conventions décrites précédemment (perpendiculaire commune aux axes  $z_j, z_{j+1}$ ).

Une fois les différents repères  $R_0, \dots, R_n$  définis, on détermine les paramètres géométriques liés à chacun des repères  $R_j$  par rapport à  $R_{j-1}$ .

1) Retrouver les paramètres de Denavit-Hartenberg modifié du robot listés ci-dessous.

$j$	$\sigma_j$	$\alpha_j$	$d_j$	$\theta_j$	$r_j$
-----	------------	------------	-------	------------	-------

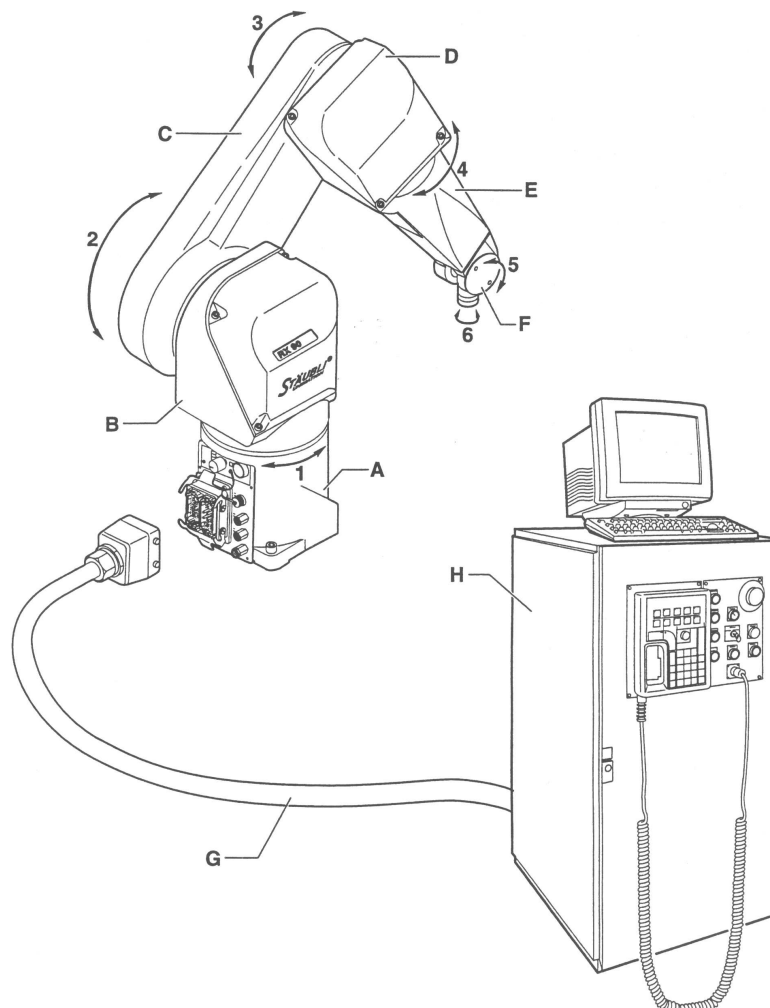
1	0	0	0	$q_1$	0
2	0	0	$D_2$	$q_2$	0
3	0	0	$D_3$	$q_3$	0
4	1	0	0	0	$q_4$

*paramètres de Denavit-Hartenberg*

- 2) Calculer la matrice  $T_{0,4}$  dans le cas où le robot est dans sa configuration initiale, c'est-à-dire, lorsque  $q_1 = q_2 = q_3 = q_4 = 0$  (voir figure précédente).
- 3) Retrouver, à l'aide de la matrice  $T_{0,4}$ , la situation de l'organe terminal par rapport au repère  $R_0$  lorsque le robot est dans sa configuration initiale.
- 4) Dédire l'expression de la matrice  $T_{0,4}$  à partir de la posture du robot correspondant à  $q_1 = 0, q_2 = -\frac{\pi}{2}, q_3 = \frac{\pi}{2}, q_4 = 0$ .

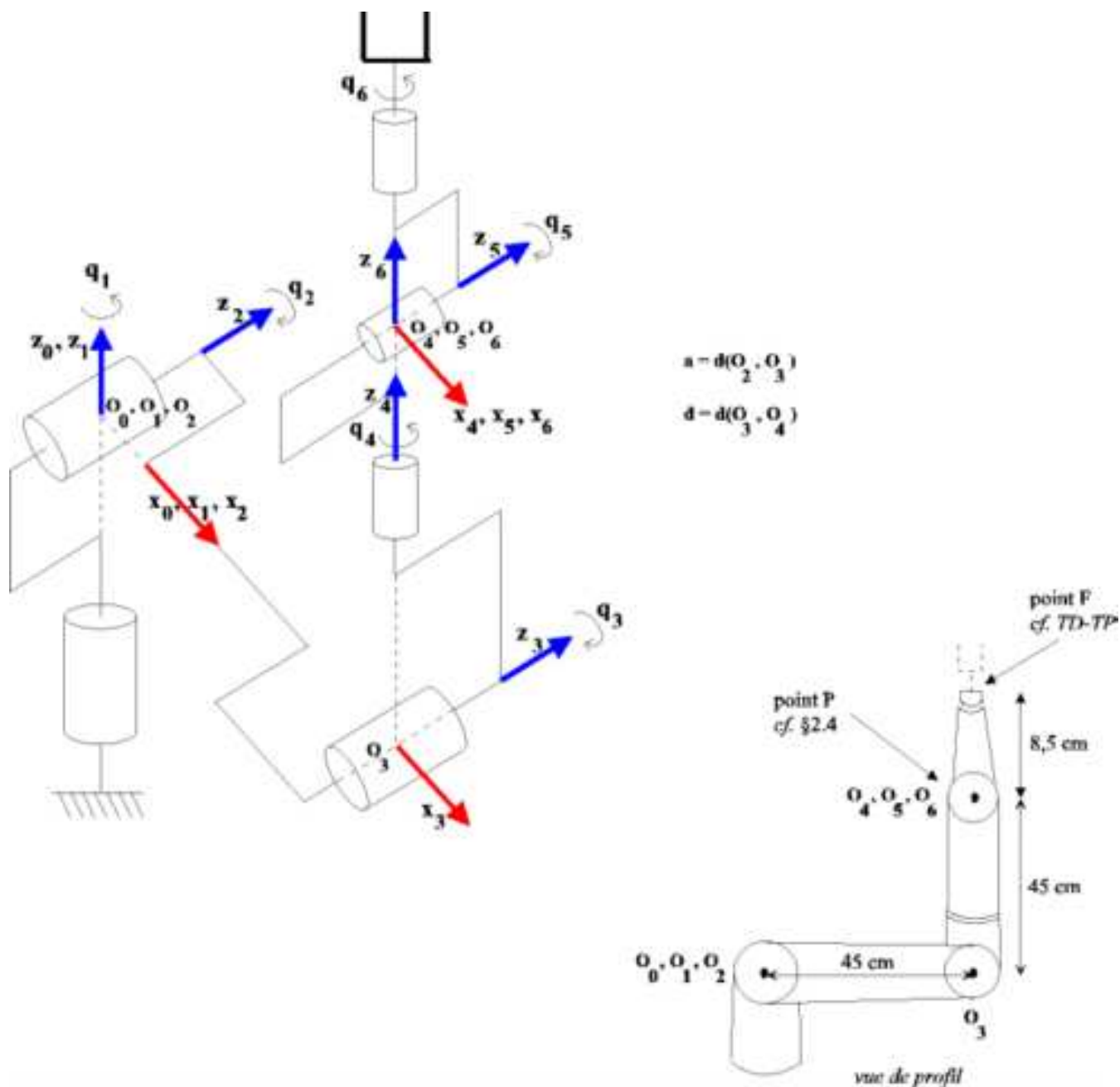
### 3.8 Exercice

Il s'agit d'établir le modèle géométrique direct du robot RX-90. Ce robot est un bras manipulateur industriel Stäubli (voir la figure suivante).

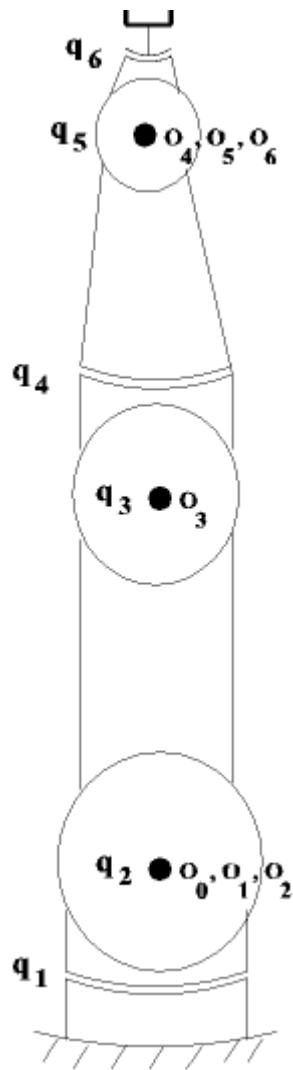


L'épaule du robot (articulations 1, 2 et 3) est du type RRR, le poignet est du type rotule, c'est-à-dire, est tel que les articulations 4, 5 et 6 sont de type rotoïde dont les axes se coupent en un même point.

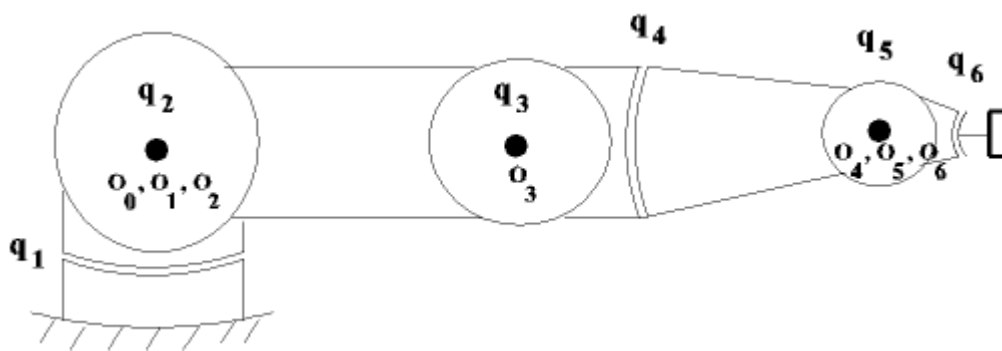
La configuration initiale du robot est donnée par la figure suivante. Afin de vous faciliter la tâche, les différents repères permettant d'établir les paramètres de Denavit-Hartenberg modifié sont représentés.



- 1) Etablir les paramètres de Denavit-Hartenberg modifié du robot.
- 2) Proposer un programme (MatLab ou Scilab) qui calcule la matrice  $T_{0,6}$  étant donné les paramètres  $q_1, \dots, q_6$ . Calculer la matrice  $T_{0,6}$  dans le cas où :
  - Le robot est dans sa configuration initiale, c'est-à-dire, lorsque  $q_1 = q_2 = \dots = q_6 = 0$  (voir figure précédente).
  - Le bras du robot est tendu à la verticale (vue de profil du robot dans la figure suivante) :

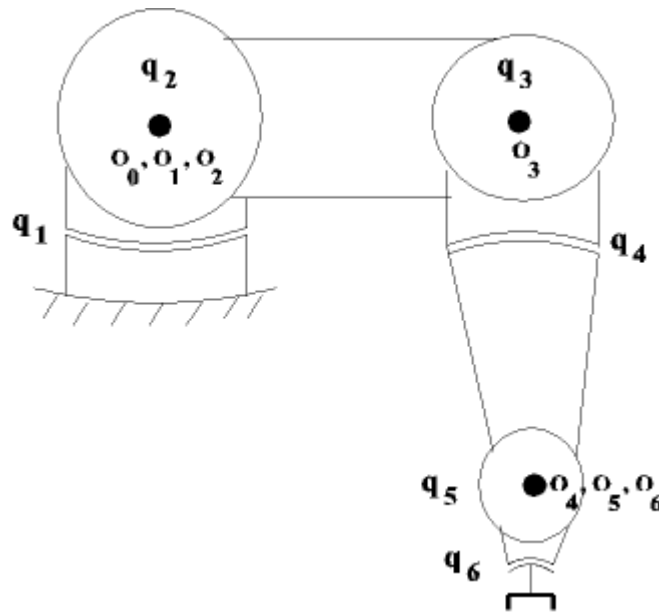


- Le bras du robot est tendu à l'horizontale selon l'axe  $\vec{x}_0$  (vue de profil du robot dans la figure suivante) :



- Le bras du robot est plié avec la pince en bas, comme représenté dans la figure suivante :





3) Pour chacun des cas, retrouver, à l'aide de la matrice  $T_{0,6}$ , la position et l'orientation de l'organe terminal par rapport au repère  $R_0$ .

### 3.9 Inversion du modèle géométrique - Méthode de Paul

Il s'agit de déterminer les coordonnées articulaires  $q$  permettant d'obtenir une situation désirée pour l'organe terminal et spécifiée par les coordonnées opérationnelles  $\underline{X}$  (cf. fin du §3.6).

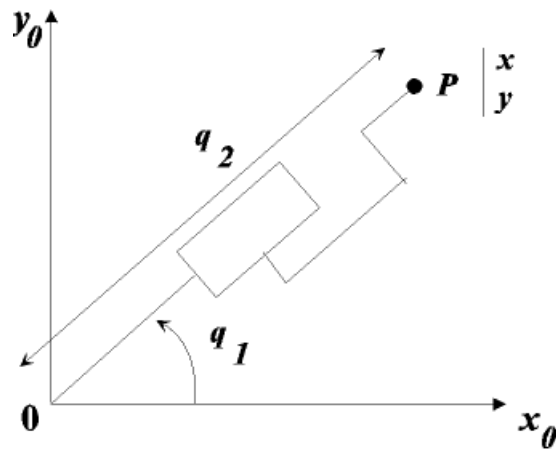
Il n'existe pas de méthode systématique d'inversion du modèle géométrique. Lorsqu'elle existe, la forme explicite, issue d'une inversion mathématique, qui donne toutes les solutions possibles au problème inverse (il y a rarement unicité de la solution) constitue le **modèle géométrique inverse**. Il existe un certain nombre de méthodes pour calculer le modèle géométrique inverse, notamment la méthode de Paul<sup>13</sup> qui traite séparément chaque cas particulier et qui convient pour la plupart des robots industriels.

Lorsque le modèle géométrique inverse n'existe pas, c'est-à-dire qu'il n'existe pas une forme explicite, on peut calculer une solution particulière du problème inverse par des procédures numériques, qui est une solution locale au sens où elle dépend des conditions initiales. Notons que de telles méthodes sont pénalisantes du point de vue du temps de calcul.

#### Exemples (simples)

##### ➤ 1<sup>er</sup> exemple

Soit le manipulateur évoluant dans un plan et décrit dans la figure suivante.



On a le modèle géométrique direct suivant :  $\begin{cases} x = q_2 \cos(q_1) \\ y = q_2 \sin(q_1) \end{cases}$ .

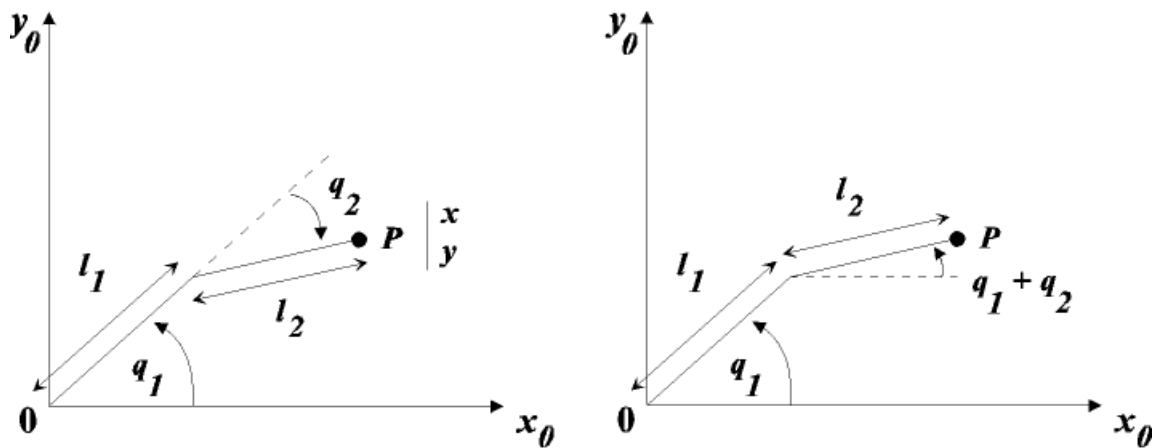
Une démarche analytique simple permet de déterminer le modèle géométrique inverse. On a :

$$\operatorname{tg}(q_1) = \frac{y}{x} \Rightarrow q_1 = \operatorname{Arctg}\left(\frac{y}{x}\right),$$

$$\text{et } x^2 + y^2 = q_2^2 \Rightarrow q_2 = \sqrt{x^2 + y^2}.$$

## ➤ 2<sup>e</sup> exemple

Soit le manipulateur RR plan décrit dans la figure suivante.

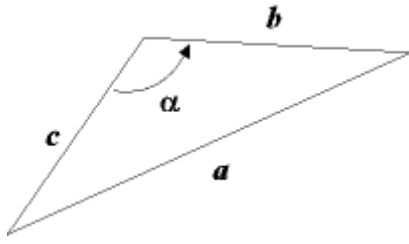


On a le modèle géométrique direct suivant :  $\begin{cases} x = l_1 \cos(q_1) + l_2 \cos(q_1 + q_2) \\ y = l_1 \sin(q_1) + l_2 \sin(q_1 + q_2) \end{cases}$ ,

soit un système de 2 équations à 2 inconnues.

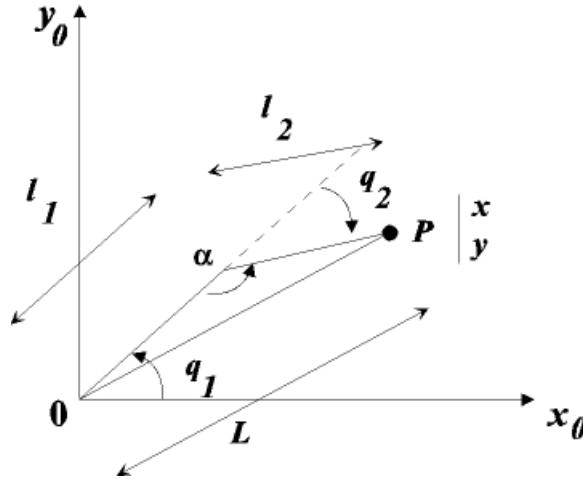
Une démarche analytique, procédant par substitution, permet de déterminer le modèle géométrique inverse.

*Rappel :* (Théorème de Pythagore généralisé)



on a :  $a^2 = b^2 + c^2 - 2 b c \cos(\alpha)$ .

Complétons le schéma du modèle géométrique direct :



On a les relations suivantes :

$$L^2 = x^2 + y^2,$$

et  $L^2 = l_1^2 + l_2^2 - 2 l_1 l_2 \cos(\alpha)$  avec  $\alpha = \pi + q_2$  (d'après le théorème de Pythagore généralisé),  
d'où

$$x^2 + y^2 = l_1^2 + l_2^2 + 2 l_1 l_2 \cos(q_2) \quad (\text{on a : } \cos(\pi + a) = -\cos(a)), \quad (1)$$

soit

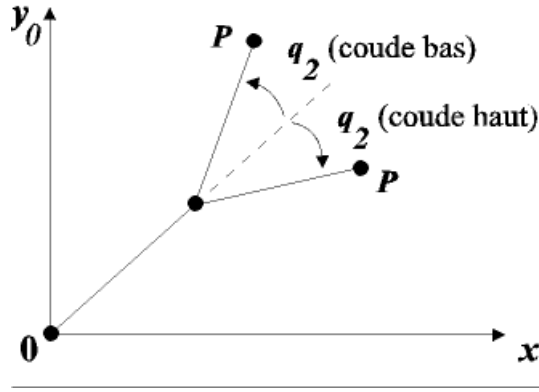
$$\cos(q_2) = \frac{x^2 + y^2 - (l_1^2 + l_2^2)}{2 l_1 l_2},$$

sachant que  $\cos q = a$  avec  $a \in [-1, 1] \Rightarrow q = \pm \text{Arcos}(a)$ , il en résulte que

$$q_2 = \pm \text{Arcos}\left(\frac{x^2 + y^2 - (l_1^2 + l_2^2)}{2 l_1 l_2}\right), \text{ sous réserve que } -1 \leq \frac{x^2 + y^2 - (l_1^2 + l_2^2)}{2 l_1 l_2} \leq 1.$$

Remarques :

- La condition  $-1 \leq \frac{x^2 + y^2 - (l_1^2 + l_2^2)}{2 l_1 l_2} \leq 1$  indique que la position du point  $P$  doit être atteignable.
- Lorsque  $q_2$  est positif (resp., négatif), le robot a une posture *coude bas* (resp., *coude haut*), voir figure ci-dessous.



D'autre part, en développant les expressions  $\cos(q_1 + q_2)$  et  $\sin(q_1 + q_2)$ <sup>14</sup> dans le système correspondant au modèle géométrique direct, on obtient :

$$\begin{cases} (l_1 + l_2 \cos(q_2)) \cos(q_1) - l_2 \sin(q_2) \sin(q_1) = x \\ l_2 \sin(q_2) \cos(q_1) + (l_1 + l_2 \cos(q_2)) \sin(q_1) = y \end{cases}$$

Calculons le déterminant de ce système d'équations linéaires par rapport aux 2 inconnues  $\cos(q_1)$  et  $\sin(q_1)$ , on a :

$$\begin{vmatrix} l_1 + l_2 \cos(q_2) & -l_2 \sin(q_2) \\ l_2 \sin(q_2) & l_1 + l_2 \cos(q_2) \end{vmatrix} = l_1^2 + l_2^2 + 2 l_1 l_2 \cos(q_2) = x^2 + y^2 \text{ (cf. Eq. (1))},$$

soient, *via* la méthode de Cramer,

$$\cos(q_1) = \frac{\begin{vmatrix} x & -l_2 \sin(q_2) \\ y & l_1 + l_2 \cos(q_2) \end{vmatrix}}{x^2 + y^2} \text{ et } \sin(q_1) = \frac{\begin{vmatrix} l_1 + l_2 \cos(q_2) & x \\ l_2 \sin(q_2) & y \end{vmatrix}}{x^2 + y^2},$$

soient

$$\begin{aligned} \cos(q_1) &= \frac{1}{x^2 + y^2} (x (l_1 + l_2 \cos(q_2)) + y l_2 \sin(q_2)), \\ \sin(q_1) &= \frac{1}{x^2 + y^2} (y (l_1 + l_2 \cos(q_2)) - x l_2 \sin(q_2)). \end{aligned}$$

Il en résulte :

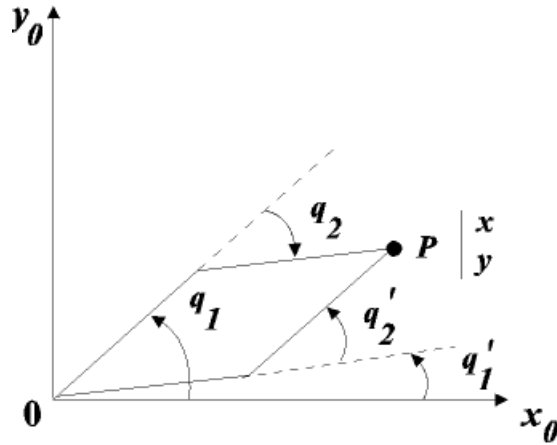
$$q_1 = \text{Arctg} \left( \frac{y (l_1 + l_2 \cos(q_2)) - x l_2 \sin(q_2)}{x (l_1 + l_2 \cos(q_2)) + y l_2 \sin(q_2)} \right).$$

Ainsi, on aboutit au modèle géométrique inverse suivant :

$$\begin{aligned} q_1 &= \text{Arctg} \left( \frac{y (l_1 + l_2 \cos(q_2)) - x l_2 \sin(q_2)}{x (l_1 + l_2 \cos(q_2)) + y l_2 \sin(q_2)} \right), \\ q_2 &= \pm \text{Arcos} \left( \frac{x^2 + y^2 - (l_1^2 + l_2^2)}{2 l_1 l_2} \right). \end{aligned}$$

On remarque qu'il y a 2 solutions, correspondant à 2 postures différentes du bras (sous réserve qu'il n'y ait pas de butées sur les articulations) : L'une est dite "coude haut", l'autre "coude bas" (cf. figure suivante).

<sup>14</sup>  $\cos(a + b) = \cos(a)\cos(b) - \sin(a)\sin(b)$ ,  $\sin(a + b) = \sin(a)\cos(b) + \cos(a)\sin(b)$



## Méthode de Paul

Dans le cas de robots à géométrie simple (pour lesquels la plupart des distances  $d_j$  et  $r_j$  sont nulles et les angles  $\theta_j$  et  $\alpha_j$  sont égaux à  $0, \pm \pi/2$ ), le modèle géométrique inverse (M.G.I.) peut être obtenu analytiquement *via* la méthode de Paul.

### ➤ Présentation

Considérons le robot décrit par la matrice de transformation suivante :

$$T_{0,n} = T_{0,1}(q_1) \times T_{1,2}(q_2) \times \cdots \times T_{n-1,n}(q_n).$$

Soit  $U_0$  la situation du repère  $R_n$  (lié à l'organe terminal) décrit par :

$$U_0 = \begin{bmatrix} S_x & N_x & A_x & P_x \\ S_y & N_y & A_y & P_y \\ S_z & N_z & A_z & P_z \\ 0 & 0 & 0 & 1 \end{bmatrix}.$$

*Rappel (cf. §2) :* On a :

- $\vec{x}_n = S_x \vec{x}_0 + S_y \vec{y}_0 + S_z \vec{z}_0$ ,  $\vec{y}_n = N_x \vec{x}_0 + N_y \vec{y}_0 + N_z \vec{z}_0$ ,  $\vec{z}_n = A_x \vec{x}_0 + A_y \vec{y}_0 + A_z \vec{z}_0$ , (3 paramètres (indépendants) pour définir l'orientation du repère  $R_n$  par rapport au repère de base).
- $\vec{O_0 O_n} = P_x \vec{x}_0 + P_y \vec{y}_0 + P_z \vec{z}_0$  (3 paramètres (indépendants) pour définir la position du point  $P$ ).

Le M.G.I. est obtenu en résolvant l'équation matricielle suivante :

$$U_0 = T_{0,1}(q_1) \times T_{1,2}(q_2) \times \cdots \times T_{n-1,n}(q_n). \quad (2)$$

La méthode de Paul permet la détermination de  $q_1$ , puis  $q_2$  et ainsi de suite jusqu'à  $q_n$ . Il s'agit de déplacer l'une après l'autre chacune des variables articulaires ( $q_1, \dots, q_n$ ) dans le membre de gauche de

l'équation. Pour cela, on multiplie par  $T_{j,j-1}$  (en prenant successivement  $j = 1, \dots, n$ ) de part et d'autre dans l'équation.

Considérons un robot à 6 *d.d.l.* ( $n = 6$ ). Appliquons la méthode :

- Pré-multiplier l'équation précédente par  $T_{1,0}$ , soit :

$$T_{1,0}(q_1) \times U_0 = T_{1,2}(q_2) \times \dots \times T_{5,6}(q_6). \quad (3)$$

Les éléments situés dans le membre de gauche sont soit indépendants, soit fonctions de  $q_1$ .

Les éléments situés dans le membre de droite sont soit constants, soit fonctions de  $q_2, \dots, q_6$ .

- Déduire  $q_1$  de l'équation précédente.
- Pré-multiplier l'équation précédente par  $T_{2,1}$ , soit :

$$T_{2,1}(q_2) \times T_{1,0}(q_1) \times U_0 = T_{2,3}(q_3) \times \dots \times T_{5,6}(q_6).$$

- En déduire  $q_2$ .
- Continuer cette procédure pour en déduire  $q_3, \dots, q_6$ .

En résumé, on utilise les équations suivantes pour calculer les variables articulaires :

$$\begin{aligned} U_0 &= T_{0,1}(q_1) \times T_{1,2}(q_2) \times T_{2,3}(q_3) \times T_{3,4}(q_4) \times T_{4,5}(q_5) \times T_{5,6}(q_6) \\ T_{1,0}(q_1) \times U_0 &= T_{1,2}(q_2) \times T_{2,3}(q_3) \times T_{3,4}(q_4) \times T_{4,5}(q_5) \times T_{5,6}(q_6) \\ T_{2,1}(q_2) \times U_1 &= T_{2,3}(q_3) \times T_{3,4}(q_4) \times T_{4,5}(q_5) \times T_{5,6}(q_6) \\ T_{3,2}(q_3) \times U_2 &= T_{3,4}(q_4) \times T_{4,5}(q_5) \times T_{5,6}(q_6) \\ T_{4,3}(q_4) \times U_3 &= T_{4,5}(q_5) \times T_{5,6}(q_6) \\ T_{5,4}(q_5) \times U_4 &= T_{5,6}(q_6) \end{aligned}$$

avec  $U_j = T_{j,6} = T_{j,j-1} \times U_{j-1}$  pour  $j = 1, 2, 3, 4$ .

La résolution de ces équations est intuitive, mais fait intervenir (en principe) quelques types d'équations dont la solution analytique est connue ; par exemple du type :

- $X r_i = Y$ ,
- $X \sin(\theta_i) + Y \cos(\theta_i) = Z$ ,
- $\begin{cases} X1 \sin(\theta_i) + Y1 \cos(\theta_i) = Z1 \\ X2 \sin(\theta_i) + Y2 \cos(\theta_i) = Z2 \end{cases}$ ,
- ...

### ➤ Cas d'un robot à 6 *d.d.l.* ayant un poignet

La méthode de Paul décrite par la suite peut s'appliquer dans le cas de robots industriels à 6 *d.d.l.* ayant un poignet. A titre d'exemple, elle sera appliquée au robot Stäubli RX 90. On appelle  $P$  le point d'intersection des axes concourants des 3 dernières articulations (voir figure suivante).



$$\begin{bmatrix} S & N & A \end{bmatrix} = A_{06}(\underline{q}),$$

soit encore,

$$A_{30}(q_1, q_2, q_3) \times \begin{bmatrix} S & N & A \end{bmatrix} = A_{36}(q_4, q_5, q_6).$$

Afin de simplifier l'écriture de cette équation, on pose :

$$\begin{bmatrix} F & G & H \end{bmatrix} = A_{36}(q_4, q_5, q_6).$$

La matrice  $\begin{bmatrix} F & G & H \end{bmatrix}$  est connue sachant que les paramètres  $q_1, q_2, q_3$  ont été préalablement déterminés. Aussi, on obtient successivement les paramètres  $q_4, q_5, q_6$  en pré-multipliant successivement l'équation précédente par  $A_{43}$ , puis par  $A_{54}$ .

## Application de la méthode de Paul au cas du robot Stäubli

### Equation de position

On a :

$$\begin{aligned} \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} &= T_{01} \times T_{12} \times T_{23} \times T_{34} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}, \\ &= \begin{bmatrix} \cos(\theta_1) [d \sin(\theta_2 + \theta_3) + a \cos(\theta_2)] \\ \sin(\theta_1) [d \sin(\theta_2 + \theta_3) + a \cos(\theta_2)] \\ d \cos(\theta_2 + \theta_3) - a \sin(\theta_2) \\ 1 \end{bmatrix}. \end{aligned}$$

- Pré-multiplication par la matrice  $T_{10}$ , on obtient :

$$T_{10} \times \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} = T_{12} \times T_{23} \times T_{34} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

*Rappel* : On a :

$$T_{01} = \begin{bmatrix} \cos(\theta_1) & -\sin(\theta_1) & 0 & 0 \\ \sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{ d'où } T_{10} = \begin{bmatrix} \cos(\theta_1) & \sin(\theta_1) & 0 & 0 \\ -\sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \text{ (cf. §3.6),}$$

et



$$\begin{aligned}
& T_{12} \times T_{23} \times T_{34} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\sin(\theta_2) & -\cos(\theta_2) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & a \\ \sin(\theta_3) & \cos(\theta_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \cos(\theta_4) & -\sin(\theta_4) & 0 & 0 \\ 0 & 0 & -1 & -d \\ \sin(\theta_4) & \cos(\theta_4) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} \cos(\theta_2 + \theta_3) & -\sin(\theta_2 + \theta_3) & 0 & a \cos(\theta_2) \\ 0 & 0 & 1 & 0 \\ -\sin(\theta_2 + \theta_3) & -\cos(\theta_2 + \theta_3) & 0 & -a \sin(\theta_2) \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ -d \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} d \sin(\theta_2 + \theta_3) + a \cos(\theta_2) \\ 0 \\ d \cos(\theta_2 + \theta_3) - a \sin(\theta_2) \\ 1 \end{bmatrix}.
\end{aligned}$$

Il en résulte les équations suivantes :

$$\begin{cases} \cos(\theta_1) P_x + \sin(\theta_1) P_y = d \sin(\theta_2 + \theta_3) + a \cos(\theta_2) \\ -\sin(\theta_1) P_x + \cos(\theta_1) P_y = 0 \\ P_z = d \cos(\theta_2 + \theta_3) - a \sin(\theta_2) \end{cases}.$$

La deuxième équation permet de déduire la valeur  $\theta_1$ , en effet, on a :

$$\frac{\sin(\theta_1)}{\cos(\theta_1)} = \frac{P_y}{P_x} \rightarrow \begin{cases} \theta_1 = \text{Arctg}\left(\frac{P_y}{P_x}\right) \\ \theta_1' = \theta_1 + \pi \end{cases}.$$

- Pré-multiplication par la matrice  $T_{21}$ , on obtient :

$$T_{21} \times T_{10} \times \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} = T_{23} \times T_{34} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix}.$$

*Rappel* : On a :

$$T_{12} = \begin{bmatrix} \cos(\theta_2) & -\sin(\theta_2) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -\sin(\theta_2) & -\cos(\theta_2) & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \text{ d'où } T_{21} = \begin{bmatrix} \cos(\theta_2) & 0 & -\sin(\theta_2) & 0 \\ -\sin(\theta_2) & 0 & -\cos(\theta_2) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix},$$

on obtient alors :

$$\begin{aligned}
T_{21} \times T_{10} \times \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} &= \begin{bmatrix} \cos(\theta_2) & 0 & -\sin(\theta_2) & 0 \\ -\sin(\theta_2) & 0 & -\cos(\theta_2) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \cos(\theta_1) & \sin(\theta_1) & 0 & 0 \\ -\sin(\theta_1) & \cos(\theta_1) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} P_x \\ P_y \\ P_z \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} \cos(\theta_2) & 0 & -\sin(\theta_2) & 0 \\ -\sin(\theta_2) & 0 & -\cos(\theta_2) & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} \cos(\theta_1) P_x + \sin(\theta_1) P_y \\ -\sin(\theta_1) P_x + \cos(\theta_1) P_y \\ P_z \\ 1 \end{bmatrix} \\
&= \begin{bmatrix} \cos(\theta_2) (\cos(\theta_1) P_x + \sin(\theta_1) P_y) - \sin(\theta_2) P_z \\ -\sin(\theta_2) (\cos(\theta_1) P_x + \sin(\theta_1) P_y) - \cos(\theta_2) P_z \\ -\sin(\theta_1) P_x + \cos(\theta_1) P_y \\ 1 \end{bmatrix}.
\end{aligned}$$

$$\text{Par ailleurs, on a : } T_{23} \times T_{34} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta_3) & -\sin(\theta_3) & 0 & a \\ \sin(\theta_3) & \cos(\theta_3) & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ -d \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} d \sin(\theta_3) + a \\ -d \cos(\theta_3) \\ 0 \\ 1 \end{bmatrix}.$$

Aussi, on a :

$$\begin{bmatrix} \cos(\theta_2) (\cos(\theta_1) P_x + \sin(\theta_1) P_y) - \sin(\theta_2) P_z \\ -\sin(\theta_2) (\cos(\theta_1) P_x + \sin(\theta_1) P_y) - \cos(\theta_2) P_z \\ -\sin(\theta_1) P_x + \cos(\theta_1) P_y \\ 1 \end{bmatrix} = \begin{bmatrix} d \sin(\theta_3) + a \\ -d \cos(\theta_3) \\ 0 \\ 1 \end{bmatrix},$$

soit  $b_1 = \cos(\theta_1) P_x + \sin(\theta_1) P_y$ , il en résulte les équations suivantes :

$$\begin{cases} d \sin(\theta_3) = \cos(\theta_2) b_1 - \sin(\theta_2) P_z - a \\ d \cos(\theta_3) = \sin(\theta_2) b_1 + \cos(\theta_2) P_z \end{cases}. \quad (4)$$

En sommant ces 2 équations, préalablement mises au carré, on obtient l'équation suivante :

$$d^2 = b_1^2 + P_z^2 + a^2 - 2ab_1 \cos(\theta_2) + 2aP_z \sin(\theta_2),$$

soit l'expression :

$$X \sin(\theta_2) + Y \cos(\theta_2) = Z \quad \text{avec} \quad X = 2aP_z; Y = -2ab_1; Z = d^2 - b_1^2 - P_z^2 - a^2.$$

On en déduit la valeur suivante du paramètre  $\theta_2$  :

$$\begin{cases} \sin(\theta_2) = \frac{XZ + \varepsilon Y \sqrt{X^2 + Y^2 - Z^2}}{X^2 + Y^2} \\ \cos(\theta_2) = \frac{YZ - \varepsilon X \sqrt{X^2 + Y^2 - Z^2}}{X^2 + Y^2} \end{cases} \quad \text{avec} \quad \varepsilon = \pm 1,$$

sachant que

$$\theta_2 = \text{Arctg}\left(\frac{\sin(\theta_2)}{\cos(\theta_2)}\right).$$

A partir du système d'équations (4) et connaissant la valeur du paramètre  $\theta_2$ , on déduit la valeur du paramètre  $\theta_3$ , soit :

$$\theta_3 = \text{Arctg}\left(\frac{b_1 \cos(\theta_2) - P_z \sin(\theta_2) - a}{b_1 \sin(\theta_2) + P_z \cos(\theta_2)}\right).$$

### Equation d'orientation

Une fois obtenus les paramètres  $q_1, q_2, q_3$ , le calcul des paramètres  $q_4, q_5, q_6$  se fait à partir de la relation :

$$A_{36}(q_4, q_5, q_6) = \begin{bmatrix} F & G & H \end{bmatrix} \quad (5)$$

avec

$$\begin{bmatrix} F & G & H \end{bmatrix} = A_{30}(q_1, q_2, q_3) \times \begin{bmatrix} S & N & A \end{bmatrix}.$$

Calcul préalable de  $A_{30}$  :

$$\begin{aligned} A_{30} &= A_{32} \times A_{21} \times A_{10} = \begin{pmatrix} \cos(\theta_3) & \sin(\theta_3) & 0 \\ -\sin(\theta_3) & \cos(\theta_3) & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos(\theta_2) & 0 & -\sin(\theta_2) \\ -\sin(\theta_2) & 0 & -\cos(\theta_2) \\ 0 & 1 & 0 \end{pmatrix} \begin{pmatrix} \cos(\theta_1) & \sin(\theta_1) & 0 \\ -\sin(\theta_1) & \cos(\theta_1) & 0 \\ 0 & 0 & 1 \end{pmatrix}, \\ &= \begin{pmatrix} \cos(\theta_3)\cos(\theta_2)\cos(\theta_1) - \sin(\theta_3)\sin(\theta_2)\cos(\theta_1) & \cos(\theta_3)\cos(\theta_2)\sin(\theta_1) - \sin(\theta_3)\sin(\theta_2)\sin(\theta_1) & -\cos(\theta_3)\sin(\theta_2) - \sin(\theta_3)\cos(\theta_2) \\ -\sin(\theta_3)\cos(\theta_2)\cos(\theta_1) - \cos(\theta_3)\sin(\theta_2)\cos(\theta_1) & -\sin(\theta_3)\cos(\theta_2)\sin(\theta_1) - \cos(\theta_3)\sin(\theta_2)\sin(\theta_1) & \sin(\theta_3)\sin(\theta_2) - \cos(\theta_3)\cos(\theta_2) \\ -\sin(\theta_1) & \cos(\theta_1) & 0 \end{pmatrix} \end{aligned}$$

Calcul de la matrice  $\begin{bmatrix} F & G & H \end{bmatrix} = A_{30} \times \begin{bmatrix} S & N & A \end{bmatrix}$  :

On a :

$$F = \begin{bmatrix} F_x \\ F_y \\ F_z \end{bmatrix} = A_{30} \times \begin{bmatrix} S_x \\ S_y \\ S_z \end{bmatrix} = \begin{pmatrix} \cos(\theta_2 + \theta_3)(\cos(\theta_1)S_x + \sin(\theta_1)S_y) - \sin(\theta_2 + \theta_3)S_z \\ -\sin(\theta_2 + \theta_3)(\cos(\theta_1)S_x + \sin(\theta_1)S_y) - \cos(\theta_2 + \theta_3)S_z \\ -\sin(\theta_1)S_x + \cos(\theta_1)S_y \end{pmatrix},$$

de même, on a

$$G = \begin{bmatrix} G_x \\ G_y \\ G_z \end{bmatrix} = A_{30} \times \begin{bmatrix} N_x \\ N_y \\ N_z \end{bmatrix} = \begin{pmatrix} \cos(\theta_2 + \theta_3)(\cos(\theta_1)N_x + \sin(\theta_1)N_y) - \sin(\theta_2 + \theta_3)N_z \\ -\sin(\theta_2 + \theta_3)(\cos(\theta_1)N_x + \sin(\theta_1)N_y) - \cos(\theta_2 + \theta_3)N_z \\ -\sin(\theta_1)N_x + \cos(\theta_1)N_y \end{pmatrix},$$

et

$$H = \begin{bmatrix} H_x \\ H_y \\ H_z \end{bmatrix} = A_{30} \times \begin{bmatrix} A_x \\ A_y \\ A_z \end{bmatrix} = \begin{pmatrix} \cos(\theta_2 + \theta_3)(\cos(\theta_1)A_x + \sin(\theta_1)A_y) - \sin(\theta_2 + \theta_3)A_z \\ -\sin(\theta_2 + \theta_3)(\cos(\theta_1)A_x + \sin(\theta_1)A_y) - \cos(\theta_2 + \theta_3)A_z \\ -\sin(\theta_1)A_x + \cos(\theta_1)A_y \end{pmatrix}.$$

- Pré-multiplication de l'équation (5) par la matrice  $A_{43}$ , on obtient :

$$A_{43} A_{36}(q_4, q_5, q_6) = A_{43} \begin{bmatrix} F & G & H \end{bmatrix},$$

avec

$$A_{43} = \begin{pmatrix} \cos(\theta_4) & 0 & \sin(\theta_4) \\ -\sin(\theta_4) & 0 & \cos(\theta_4) \\ 0 & -1 & 0 \end{pmatrix}.$$

Il en résulte le système suivant :

$$\begin{pmatrix} \cos(\theta_4)F_x + \sin(\theta_4)F_z & \cos(\theta_4)G_x + \sin(\theta_4)G_z & \cos(\theta_4)H_x + \sin(\theta_4)H_z \\ -\sin(\theta_4)F_x + \cos(\theta_4)F_z & -\sin(\theta_4)G_x + \cos(\theta_4)G_z & -\sin(\theta_4)H_x + \cos(\theta_4)H_z \\ -F_y & -G_y & -H_y \end{pmatrix} = \begin{pmatrix} \cos(\theta_5)\cos(\theta_6) & -\cos(\theta_5)\sin(\theta_6) & \sin(\theta_5) \\ \sin(\theta_6) & \cos(\theta_6) & 0 \\ -\sin(\theta_5)\cos(\theta_6) & \sin(\theta_5)\sin(\theta_6) & \cos(\theta_5) \end{pmatrix}.$$

L'élément (2, 3), à savoir  $(-\sin(\theta_4)H_x + \cos(\theta_4)H_z = 0)$ , permet de déduire le paramètre  $\theta_4$ . En effet, on a :

$$\frac{\sin(\theta_4)}{\cos(\theta_4)} = \frac{H_z}{H_x} \rightarrow \begin{cases} \theta_4 = \text{Arctg}\left(\frac{H_z}{H_x}\right) \\ \theta_4' = \theta_4 + \pi \end{cases}.$$

Les éléments (1,3) et (3,3) permettent de déduire le paramètre  $\theta_5$ . En effet, on a :

$$\begin{cases} \cos(\theta_4)H_x + \sin(\theta_4)H_z = \sin(\theta_5) \\ -H_y = \cos(\theta_5) \end{cases},$$

soit

$$\theta_5 = \text{Arctg}\left(\frac{\cos(\theta_4)H_x + \sin(\theta_4)H_z}{-H_y}\right).$$

Enfin, les éléments (2,1) et (2,2) permettent de déduire le paramètre  $\theta_6$ . En effet, on a :

$$\begin{cases} -\sin(\theta_4)F_x + \cos(\theta_4)F_z = \sin(\theta_6) \\ -\sin(\theta_4)G_x + \cos(\theta_4)G_z = \cos(\theta_6) \end{cases},$$

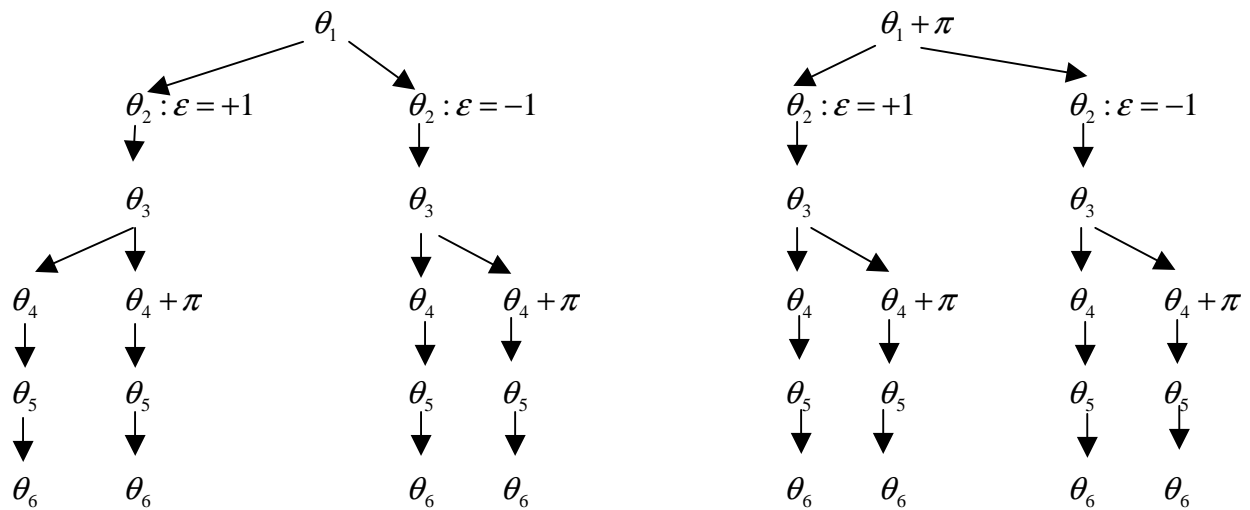
soit

$$\theta_6 = \text{Arctg}\left(\frac{-\sin(\theta_4)F_x + \cos(\theta_4)F_z}{-\sin(\theta_4)G_x + \cos(\theta_4)G_z}\right).$$

### Nombre de solutions du M.G.I. :

Il existe 8 solutions, dans le cas général, correspondant au M.G.I. du robot Stäubli RX-90. Notons que certaines positions dites *singulières* du robot occasionnent un nombre infini de solutions. C'est par exemple, le cas de la configuration initiale sans offset ( $\theta_1 = \dots = \theta_6 = 0$ ) du robot (cf. §3.8) où les arguments de la fonction *Arctg* utilisés pour déterminer le paramètre  $\theta_4$  sont nuls ( $H_x = H_z = 0$ ), ce qui rend ce paramètre indéterminé. Le choix de la valeur de ce paramètre étant libre, on assigne fréquemment la valeur courante  $\theta_4$  du robot. Notons que ce choix fixe alors la valeur du paramètre  $\theta_6$ .

A titre d'exercice, proposer un programme (MatLab ou Scilab) qui calcule ces solutions étant donné les matrices d'orientation ( $S, N, A$ ) et de position ( $P$ ).



Notons que certaines solutions peuvent correspondre à des configurations non accessibles du fait de contraintes (limitations angulaires) au niveau des variables articulaires.

### Exercice :

- 1) Réaliser un programme - MatLab ou Scilab - permettant de calculer les 4 jeux de paramètres  $\theta_1, \theta_2, \theta_3$  en fonction d'une position souhaitée du point  $P$  (centre du poignet). A titre de vérification, considérer la configuration initiale du robot (voir §3.8) et retrouver, parmi les 4 solutions possibles, le jeu de paramètres  $\theta_1 = \theta_2 = \theta_3 = 0$  (rappel :  $a = d = 0,45 \text{ m}$ ).
- 2) La baie de commande du robot Staübli RX 90 fournit différentes informations, notamment :
  - (en mode World) la situation de l'organe terminal, c'est-à-dire, de la flasque (point  $F$  situé à  $8,5 \text{ cm}$  du point  $P (= O_6)$  selon l'axe  $\vec{z}_6$ ), à travers 6 informations :
    - la position de la flasque : les coordonnées  $X, Y, Z$  en  $\text{mm}$  ;
    - l'orientation de la flasque : les angles  $y, p, r$  (*lacet* (yaw en anglais), *tangage* (pitch) et *roulis* (roll)), en degré, correspondant respectivement aux 3 angles d'Euler  $\psi, \theta, \varphi$  selon la convention  $(z, y, z)$ , décrits au §2.1).
  - (en mode Joint) les valeurs angulaires des 6 articulations, notées  $\theta_1, \dots, \theta_6$ .

Soient, par exemple :

$X (\text{mm})$	$Y$	$Z$	$y (^\circ)$	$p$	$r$
598,629	-372,697	518,632	-23,395	93,034	47,881
$\theta_1 (^\circ)$	$\theta_2$	$\theta_3$	$\theta_4$	$\theta_5$	$\theta_6$
-33,064	-65,607	141,025	29,283	20,053	19,586

Il s'agit de retrouver les valeurs des paramètres  $\theta_1, \theta_2, \theta_3$  à partir des informations donnant la situation de la flasque  $F$ , c'est-à-dire, à partir de  $X, Y, Z, y, p, r$ . Pour cela, compléter le programme précédent en permettant de déduire les coordonnées du point  $P$ , c'est-à-dire,  $P_x, P_y, P_z$ , à partir de celles du point  $F$ , c'est-à-dire,  $X, Y, Z$  (issues de la baie de commande).

### 3.10 Solutions multiples – Espace de travail – Aspects

On a vu précédemment que le M.G.D. d'un robot peut se représenter à l'aide de l'application :

$$\underline{X} = f(\underline{q}),$$

cette application étant définie de l'espace articulaire ( $Q$ ) vers l'espace opérationnel ( $X$ ), tous deux de dimension  $n \leq 6$ .

Cette application n'est pas *biunivoque*, i.e., à un élément de  $Q$  correspond une seule image dans  $X$ , par contre un élément de  $X$  pourra être l'image de plusieurs éléments de  $Q$ . Dans le cas du robot Staübli RX 90, il existe (au maximum) 8 solutions possibles pour atteindre un point de l'espace opérationnel.

Dans la pratique, cette non propriété peut poser des problèmes, par exemple, dans le cas où des obstacles sont à proximité du robot. En effet, la réalisation d'une trajectoire rectiligne peut s'avérer irréalisable physiquement, ou le bras du robot peut être amené à se reconfigurer, ce qui peut induire des comportements "intempestifs" du bras du robot.

Une alternative consiste à subdiviser l'espace articulaire  $Q$  en un ensemble de  $m$  domaines disjoints, notés  $(A_i)$ ,  $i = 1$  à  $m$ , appelés *aspects*<sup>15</sup>, dans lesquels l'application  $f$  sera *biunivoque*. Ainsi dans chacun de ces domaines, il est assuré que chaque point de l'espace opérationnel correspondant (soit  $f(A_i)$ ) soit atteignable à travers une seule et même configuration (posture) possible. Décrivons la méthode permettant le calcul de ces domaines.

Soit le M.G.D. décrit sous la forme :

$$x_i = f_i(q_1, q_2, \dots, q_n) \quad i = 1 \text{ à } n.$$

On définit le modèle cinématique, relatif aux vitesses, du type  $\dot{\underline{X}} = g(\dot{\underline{q}})$ , décrit par l'équation matricielle  $n \times n$  suivante :

$$\begin{pmatrix} \dot{x}_1 \\ \vdots \\ \dot{x}_n \end{pmatrix} = \begin{pmatrix} \frac{\partial f_1}{\partial q_1} & \dots & \frac{\partial f_1}{\partial q_n} \\ \vdots & \ddots & \vdots \\ \frac{\partial f_n}{\partial q_1} & \dots & \frac{\partial f_n}{\partial q_n} \end{pmatrix} \begin{pmatrix} \dot{q}_1 \\ \vdots \\ \dot{q}_n \end{pmatrix},$$

soit encore,

$$\dot{\underline{X}} = J \dot{\underline{q}},$$

où  $J$  est la matrice *Jacobienne* du M.G.D..

Il s'avère que l'application  $f$  est *biunivoque* dans un domaine restreint de  $Q$ , si dans ce domaine le déterminant de la matrice  $J$  garde un signe constant, ce qui permet une description exhaustive de l'espace opérationnel de travail du manipulateur<sup>16</sup>.

## Espace de travail – Aspects

Le domaine articulaire  $Q$  admissible est en général un hyperparallélépipède défini par :

$$q_{i_{\min}} \leq q_i \leq q_{i_{\max}} \quad i = 1 \text{ à } n,$$

<sup>15</sup> Borrel P., *Contribution à la modélisation géométrique des robots manipulateurs ; application à la conception assistée par ordinateur*, Thèse d'Etat, USTL Montpellier, juillet 1986.

<sup>16</sup> Ce résultat s'applique pour la plupart des robots industriels, plus précisément pour les robots dits non "*cuspidals*" (voir *Modeling, Identification & Control of Robots*, W. Khalil, E. Dombre, Hermes Penton Science 2002 pour plus de détails).

où  $q_{i_{\min}}$  et  $q_{i_{\max}}$  représentent les butées présentes au niveau de l'articulation  $i$ .

L'équation  $\det J(\underline{q}) = 0$  définit une partition de l'espace articulaire  $Q$  en  $m$  domaines disjoints  $(A_i)$ ,  $i = 1$  à  $m$ , appelés *aspects*. Ainsi un *aspect* est défini par la relation :

$$\forall \underline{q} \in A_i, \text{ Signe de } ([\det J(\underline{q})]) = \text{constante}$$

avec les propriétés suivantes :

$$(A_i) \cap (A_j) = \emptyset \quad \forall i, j = 1, \dots, m$$

$$(Q) = \bigcup_{i=1}^m (A_i).$$

*Remarque* : Les configurations pour lesquelles  $\det J(\underline{q}) = 0$  sont appelées *configurations singulières*, ou *singularités*, du mécanisme.

Définissons le sous-domaine, noté  $(X_i)$ , du domaine opérationnel  $(X)$  des points  $q$  de  $(A_i)$  images par l'application  $f$ , soit :

$$(X_i) = \text{Im} (A_i) .$$

En général, les domaines  $(X_i)$  ne sont pas disjoints (*i.e.*,  $\exists i, j (i \neq j) \mid (X_i) \cap (X_j) \neq \emptyset$ ), contrairement aux domaines  $(A_i)$ .

L'espace opérationnel  $(X)$  correspond alors à l'union des  $m$  images d'*aspects*, soit :

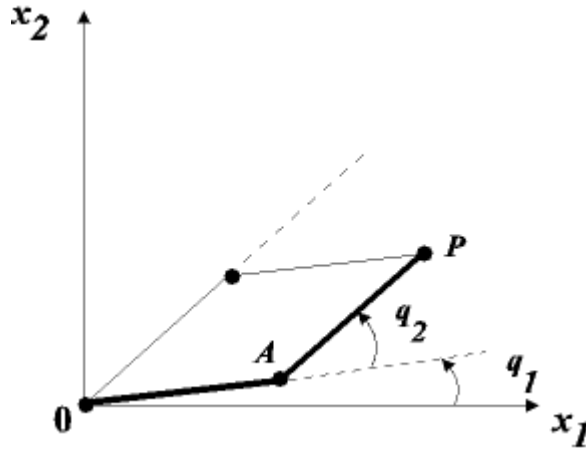
$$(X) = \bigcup_{i=1}^m \text{Im} (A_i) .$$

Pour un point  $q \in (A_i)$ , le manipulateur est dit dans la *configuration*, ou la *posture*, numéro  $i$ . Par exemple, soit le point  $x \in (X_i) \cap (X_j) \cap (X_k)$ , alors ce point est l'image d'un point de  $(A_i)$ , d'un point de  $(A_j)$  et d'un point de  $(A_k)$ , ce qui signifie que ce point peut être atteint dans les postures  $i, j$  et  $k$ .

Cette description permet de résoudre le problème du transfert dans une posture donnée du manipulateur d'un point  $x^1$  vers un point  $x^2$ . Notamment, on peut se poser la question de savoir si les deux points peuvent être raliés par une trajectoire quelconque, par exemple rectiligne.

### Exemple

Reprenons le manipulateur RR considéré dans l'exemple 2 du §3.9, avec  $l_1 = l_2 = L$ .



### ▪ Définition des aspects

Soit le M.G.D. suivant :

$$\begin{cases} x_1 = L \cos(q_1) + L \cos(q_1 + q_2) = f_1(q_1, q_2) \\ x_2 = L \sin(q_1) + L \sin(q_1 + q_2) = f_2(q_1, q_2) \end{cases},$$

avec pour contraintes :

$$0 \leq q_1 \leq 90^\circ, -100^\circ \leq q_2 \leq 90^\circ.$$

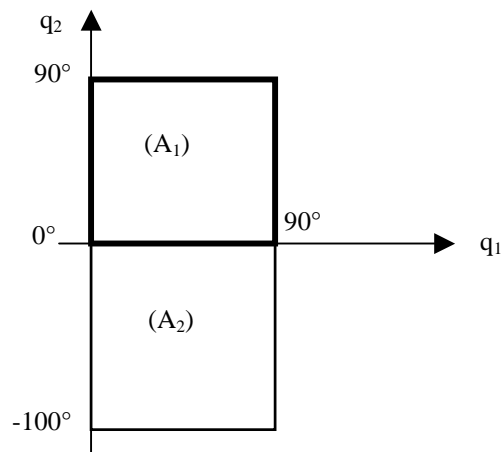
On a la matrice *Jacobienne* suivante :

$$J = \frac{\partial f}{\partial q} = \begin{pmatrix} -L \sin(q_1) - L \sin(q_1 + q_2) & -L \sin(q_1 + q_2) \\ L \cos(q_1) + L \cos(q_1 + q_2) & L \cos(q_1 + q_2) \end{pmatrix},$$

il en résulte que :

$$\det J = L^2 \sin(q_2).$$

Sachant que  $\det J = 0$  entraîne  $q_2 = 0$ , il s'ensuit la séparation du domaine articulaire ( $Q$ ) en deux aspects ( $A_1$ ) et ( $A_2$ ) selon la figure suivante :



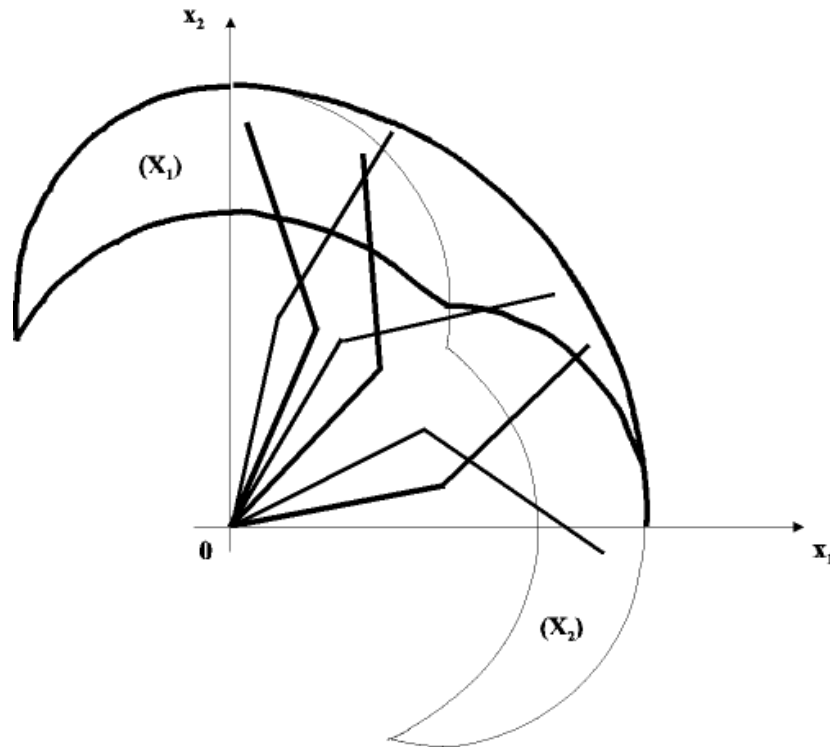
L'aspect ( $A_1$ ), domaine représenté en trait gras, correspond à  $\det J > 0$ , soit  $q_2 > 0$ , et à la posture *coude bas* (représenté en trait gras) du manipulateur. L'aspect ( $A_2$ ), domaine représenté en trait fin, correspond à  $\det J < 0$ , soit  $q_2 < 0$ , et à la posture *coude haut* (représenté en trait fin) du manipulateur. Dans



l'espace cartésien, la posture en trait gras est l'image de  $Q_1$ , tandis que la posture en trait fin est l'image de  $Q_2$ .

### ■ Génération de l'espace de travail

Pour construire les domaines  $(X_1)$  et  $(X_2)$ , images respectivement de  $(A_1)$  et  $(A_2)$ , il suffit de se déplacer sur la frontière des *aspects* et de tracer géométriquement l'évolution de l'extrémité  $P$  du manipulateur. Le domaine atteint par l'extrémité du manipulateur étant le domaine  $(X) = (X_1) \cup (X_2)$ , voir la figure suivante.



Trois cas sont à considérer pour les solutions du M.G.I. :

- $P \in (X) - (X_2)$  : le M.G.I. a une solution dans  $(A_1)$ , le manipulateur est en posture *coude bas*.
- $P \in (X) - (X_1)$  : le M.G.I. a une solution dans  $(A_2)$ , le manipulateur est en posture *coude haut*.
- $P \in (X_1) \cap (X_2)$  : le M.G.I. a 2 solutions, l'une dans  $(A_1)$ , l'autre dans  $(A_2)$ , le manipulateur peut prendre les 2 postures.

### ■ Réalisation de trajectoires entre deux points $I$ et $F$

Soient les points :  $I_1 \in (X) - (X_1)$ ,  $F_1 \in (X) - (X_2)$ ,  $I_2$  et  $F_2 \in (X_1) \cap (X_2)$ .

- Considérons la trajectoire  $I_1 F_1$ , il est impossible de rallier  $I_1$  et  $F_1$  sans reconfiguration. La trajectoire passe nécessairement par un point de la frontière commune, il est en particulier impossible de réaliser la trajectoire rectiligne entre les points  $I_1$  et  $F_1$ .
- Considérons la trajectoire  $I_2 F_2$ , la trajectoire rectiligne entre les points  $I_2$  et  $F_2$  est réalisable, aussi bien en posture *coude haut* qu'en posture *coude bas*.
- Considérons la trajectoire  $I_2 F_1$ , la trajectoire rectiligne entre les points  $I_2$  et  $F_1$  est réalisable si au point  $I_2$  le manipulateur est en posture *coude bas*. Dans le cas contraire il y aura reconfiguration.

## 4 GÉNÉRATION DE MOUVEMENT

La tâche de déplacement d'un robot est spécifiée en définissant un chemin que le robot doit suivre. Un *chemin* est une séquence de *points* définis soit dans l'espace des tâches (opérationnel) (afin de situer l'organe terminal), soit dans l'espace des configurations (articulaire) du robot (afin d'indiquer les valeurs des paramètres de liaison).

Ces *points* peuvent être :

- programmés par apprentissage,
- issus d'une base de données d'un système de CAO, ...

Le problème de la génération de mouvement est de calculer les séquences souhaitées (consigne) de variables articulaires ou de variables liées à l'organe terminal qui assurent le passage du robot par le *chemin* désiré.

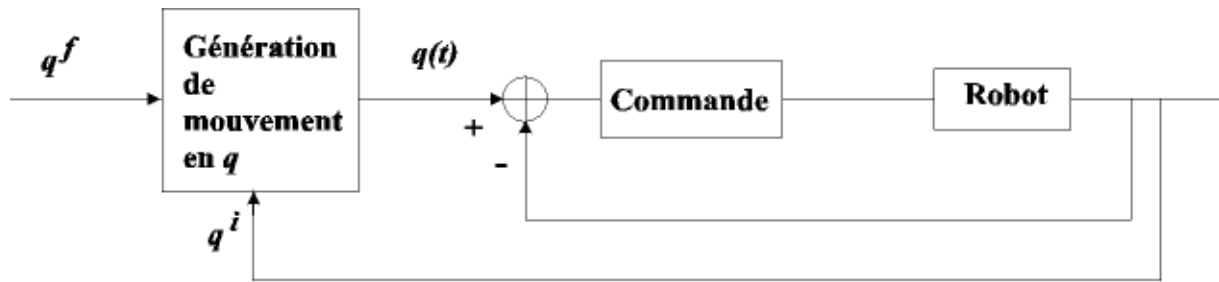
Les trajectoires d'un robot peut être classifiées comme suit :

- les mouvements entre 2 points avec des mouvements libres entre les points,
- les mouvements entre 2 points *via* une séquence de points intermédiaires désirés, spécifiés notamment pour éviter les obstacles ; la trajectoire est libre entre les points intermédiaires,
- les mouvements entre 2 points, la trajectoire étant contrainte entre les points (trajectoire rectiligne par exemple),
- les mouvements entre 2 points *via* des points intermédiaires, la trajectoire étant contrainte entre les points intermédiaires.

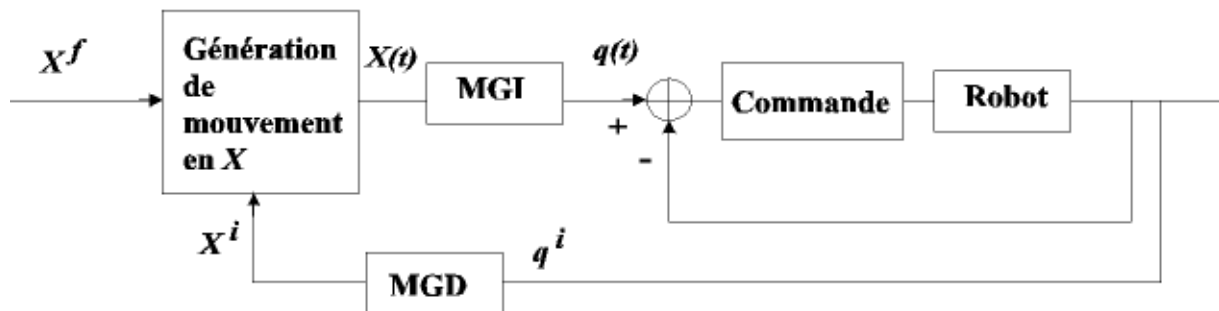
Dans les deux premiers cas, la génération de mouvement peut se faire directement dans l'espace des configurations : elle se traduit par une séquence de positions articulaires constituant les consignes des asservissements.

Dans les deux derniers cas, la trajectoire étant fixée à tout instant dans l'espace opérationnel, il est préférable de raisonner dans cet espace. La loi de commande engendrée doit ensuite être transformée en consignes articulaires par le changeur de coordonnées.

Ces 2 approches – génération de mouvement dans l'espace articulaire et génération de mouvement dans l'espace opérationnel – sont schématisées sur les figures suivantes.



***Génération de mouvement dans l'espace articulaire***



***Génération de mouvement dans l'espace opérationnel***

- La génération de mouvement dans l'*espace articulaire* présente plusieurs avantages :
  - le mouvement est minimal sur chaque articulation,
  - elle nécessite moins de calcul en ligne (au sens où il n'y a pas de changeur de coordonnées),
  - le mouvement n'est pas affecté par le passage sur les configurations singulières,
  - les contraintes de vitesse et de couples maximaux sont connues avec précision puisqu'elles correspondent aux limites physiques des actionneurs.

En contrepartie, la géométrie de la trajectoire dans l'espace opérationnel ne peut être imposée. Entre 2 points donnés, l'organe terminal se déplace de façon imprévisible mais répétitive (ce qui peut occasionner des collisions lorsque le robot évolue dans un environnement encombré). Ce type de mouvement est par conséquent approprié pour réaliser des déplacements rapides dans un espace dégagé.

- La génération de mouvement dans l'*espace opérationnel* permet de contrôler la géométrie de la trajectoire (mouvement rectiligne par exemple). Par contre :
  - elle implique la transformation en coordonnées articulaires de chaque point de la trajectoire,
  - elle peut être mise en échec lorsque la trajectoire calculée passe par une position singulière,
  - elle peut être mise en échec chaque fois que les points de la trajectoire engendrée ne sont pas dans le volume accessible du robot ou chaque fois que la trajectoire impose une reconfiguration du mécanisme (changement d'aspect en cours de trajectoire).

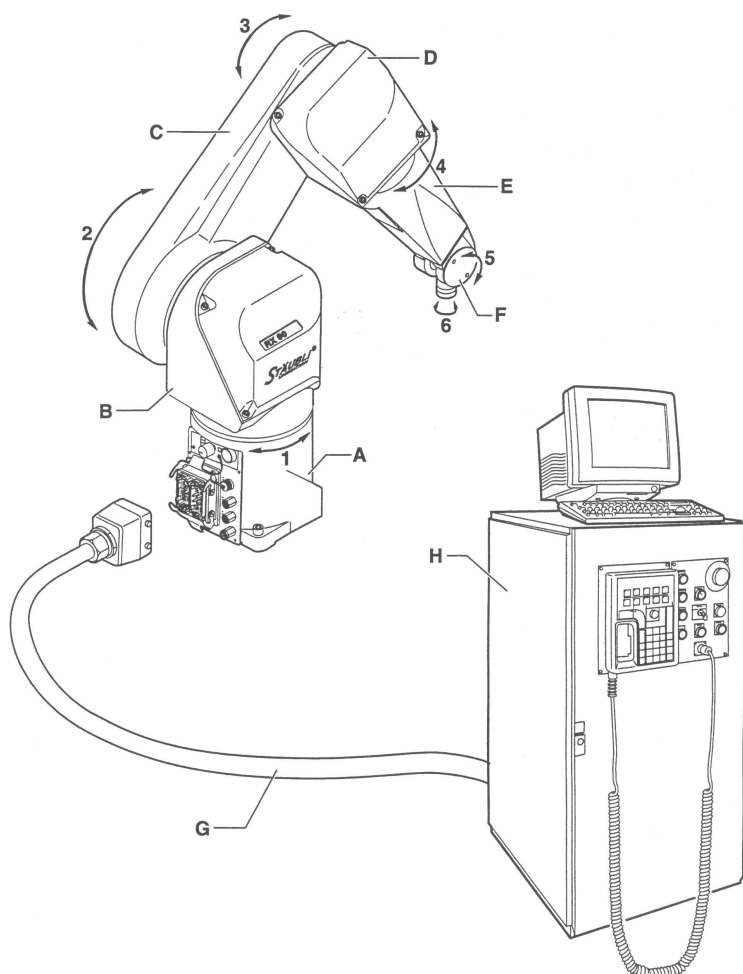
## 5 DESCRIPTION PRATIQUE DU ROBOT STÄUBLI RX 90



### 5.1 Description générale

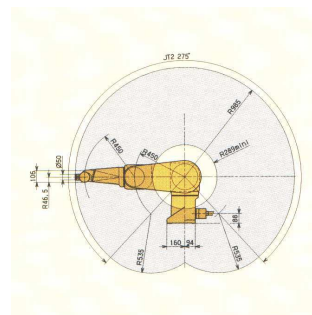
Les robots de la famille RX série 90 sont du type polyarticulé à 6 degrés de liberté. Ils se composent, voir figure ci-dessous, d'un organe mécanique bras (A à F) et d'une baie de commande (H) (voir documentation « Caractéristiques Baie de Commande CS7 » pour plus détails), le tout étant relié par un câble de liaison (G).

Le bras est constitué de segments reliés entre eux par des articulations. Chaque articulation constitue un axe autour duquel deux segments pivotent. Les mouvements des articulations du robot sont générés par des servomoteurs (moteurs asservis) sans balais, couplés à des résolveurs (capteurs de précision). Les différents éléments du bras du robot sont le pied (A), l'épaule (B), le bras (C), le coude (D), l'avant-bras (E) et le poignet (F) (voir figure ci-dessous). L'ensemble bras du robot contient la motorisation, les freins, les mécanismes de transmission du mouvement, le faisceau de câbles, les circuits pneumatique et électrique pour l'utilisateur et le système d'équilibrage (effectué par un système intégré de ressorts) du bras.



### Quelques caractéristiques du robot RX 90 :

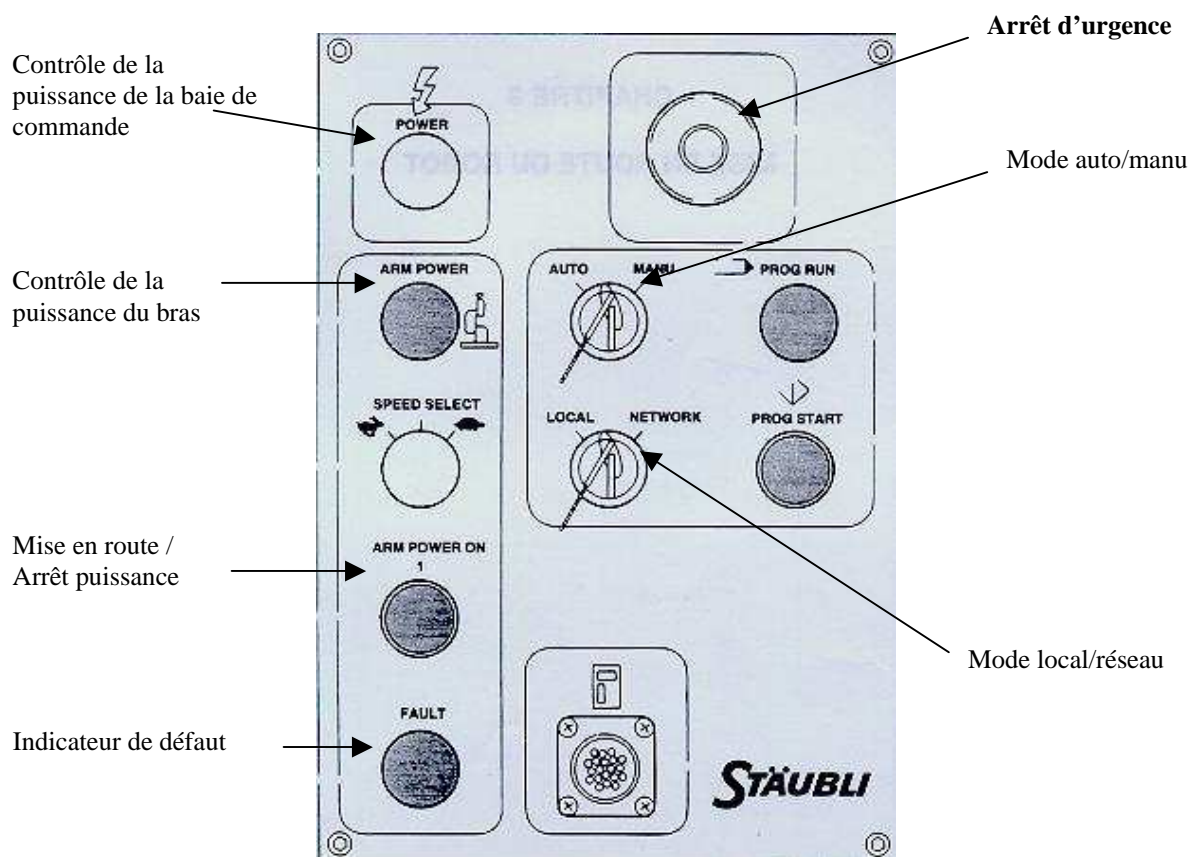
Nombre de degrés de liberté	4, 5 ou 6
Charge transportable nominale	6 Kg
Charge transportable maximale	12 Kg
Rayon d'action	985 mm
Répétabilité	+/- 0,02 mm
Langage de programmation	V+



Axe	1	2	3	4	5	6
Amplitude (°)	320 (+/- 160)	275 (+/- 137,5)	285 (+/- 142,5)	540 (+/- 270)	225 (+120/-105)	540 (+/- 270)
Vitesse nominale (°/s)	236	200	286	401	320	580
Vitesse maximale (°/s)	356	356	296	409	480	1125

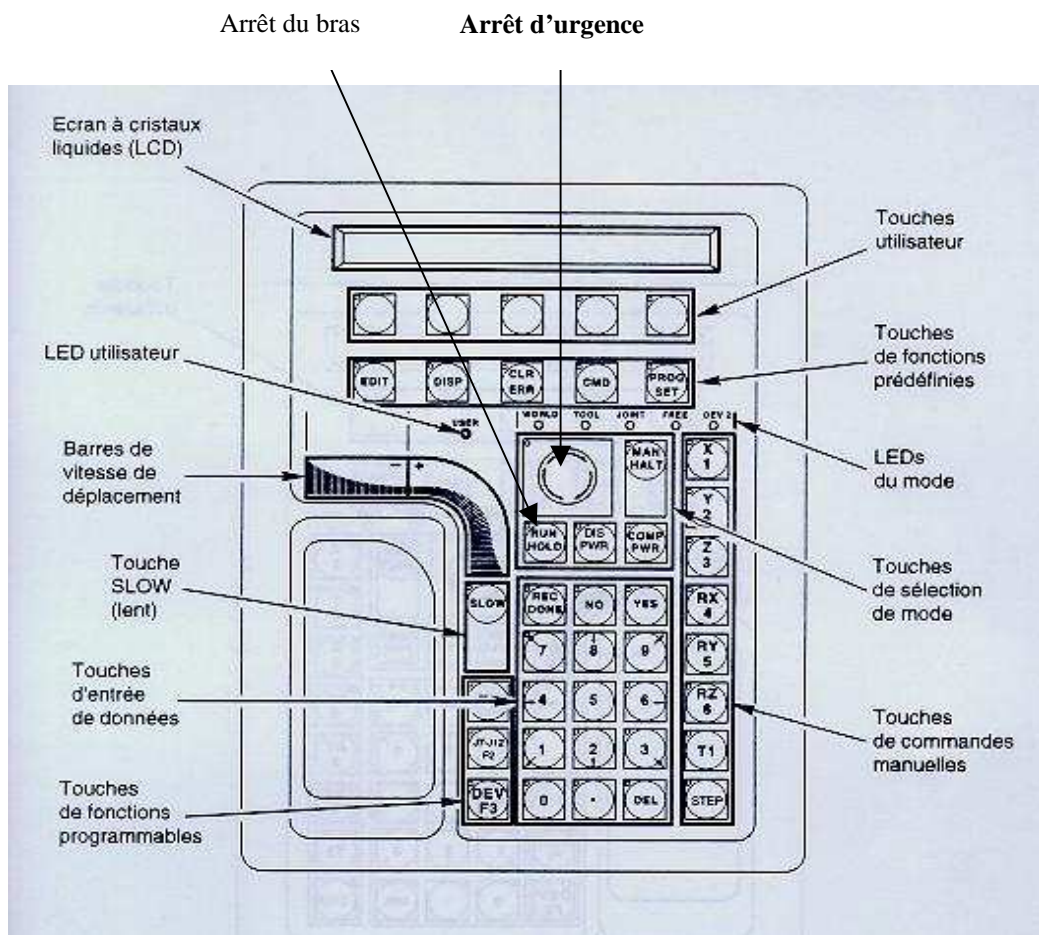
N.B. : Les amplitudes (°) indiquées ci-dessus sont exprimées relativement à la configuration initiale du robot (issue d'une commande READY).

**La face avant de la baie de commande CS7 se présente comme suit :**



Elle permet la mise en marche, l'arrêt, la signalisation, ...

Le pendant est le boîtier permettant de commander manuellement le robot, il se compose comme suit :



La commande manuelle du robot se fait à partir du pendant, la commande automatique se faisant au clavier *via* des commandes spécifiques.

Le pendant va permettre de définir des points de passage (appelés variables *points*) du bras du robot (cf. §6.3.3.b).

### Règles de sécurité :

- Personne ne doit être dans l'aire de travail du bras du robot ;
- La phase de test du robot s'effectue toujours en vitesse faible (SPEED 10) ;
- Durant la phase de test du robot, être prêt à appuyer sur le bouton « Run/Hold » du pendant pour arrêter le bras.

## 5.2 Mise en route du système

- Mettre l'interrupteur général (de couleur rouge), situé au dos de la baie de commande, sur **1**. Vérifier que le voyant **Power**, situé sur la face avant, est allumé.
- Mettre l'écran du terminal en fonctionnement.
- Notons que le commutateur **Pendant/Terminal/Network**, situé sur la face avant de la baie, est positionné sur **Terminal**, ce qui rend le terminal actif.
- Attendre la fin du démarrage du système (c-à-d, l'apparition du prompt « . » à l'écran du terminal).

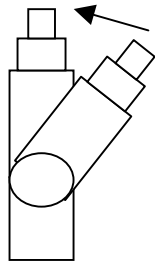
- Afin de demander la mise sous puissance du bras du robot, au choix :
  - Taper au clavier l'instruction **ENABLE POWER (enter)**,
  - Ou, appuyer sur le bouton **Comp/Pwr** du pendant.
- *Dans les 15 secondes qui suivent*, mettre la puissance sur le bras, en appuyant sur le bouton **Arm Power On** (qui clignote) de la face avant de la baie. La puissance est signalée par la présence sur la face avant de la baie du voyant **Arm Power**. Dans le cas contraire et si le voyant **Fault** est allumé, vérifiez que les boutons d'arrêt d'urgence du pendant et de la face avant de la baie ne sont pas enclenchés et que le déblocage des freins (situé sur le pied du robot) est sur 0.
- Le système est prêt à accepter vos commandes. Si le terminal est actif, mettre une vitesse faible, en tapant au clavier l'instruction **SPEED 10 (enter)**.

### 5.3 Mise en position initiale du bras du robot

La configuration initiale du robot correspond à une position verticale du bras du robot (voir figure

suivante), on a :  $T_{0,6} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0,9 \\ 0 & 0 & 0 & 1 \end{pmatrix}$ . Les valeurs articulaires correspondantes sont :

$$\theta_1 = 0, \theta_2 = -\frac{\pi}{2}, \theta_3 = \frac{\pi}{2}, \theta_4 = \theta_5 = \theta_6 = 0.$$



A l'aide du terminal, taper au clavier (en mode *Commande*, signalé par le prompt « . ») les instructions :

- **SPEED 10 (enter)**
- **DO READY (enter)**

**Remarque** : le préfixe **DO** permet une exécution de la commande, en l'occurrence **READY**, en mode commande.

### 5.4 Arrêt du système

- *L'arrêt immédiat du bras (en mouvement)* est réalisé en appuyant sur le bouton **Run/Hold** du pendant (*l'arrêt après l'instruction en cours d'exécution* se fait en tapant au clavier l'instruction **ABORT (enter)**).
- *La coupure de la puissance du bras* est réalisée, au choix :
  - En appuyant sur le bouton d'arrêt d'urgence de la face avant de la baie ou celui du pendant.
  - En tapant l'instruction **DISABLE POWER (enter)** au clavier.

**Attention** : Ne jamais couper la puissance de la baie de commande (*via l'interrupteur général situé au dos de la baie de commande*) quand le bras est sous puissance.

- Couper l'alimentation de l'écran du terminal.



- Mettre l'interrupteur général, situé au dos de la baie de commande, sur 0.

## 6 MISE EN MOUVEMENT DU BRAS DU ROBOT STÄUBLI RX 90

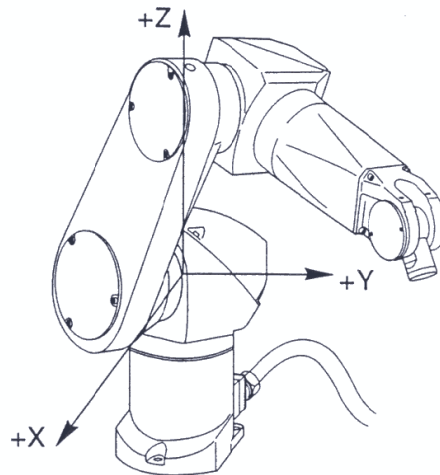
Une fois la mise en route du système effectuée (cf. §5.2), on dispose de 2 moyens pour mettre le bras en mouvement : Soit *via* le pendant, soit *via* l'exécution d'un programme (à l'aide du terminal). Avant cela, décrivons les différents modes de déplacements possibles du bras.

### 6.1 Les modes de déplacement

La situation (position et orientation) de l'organe terminal d'un robot est déterminée à partir de 6 paramètres.

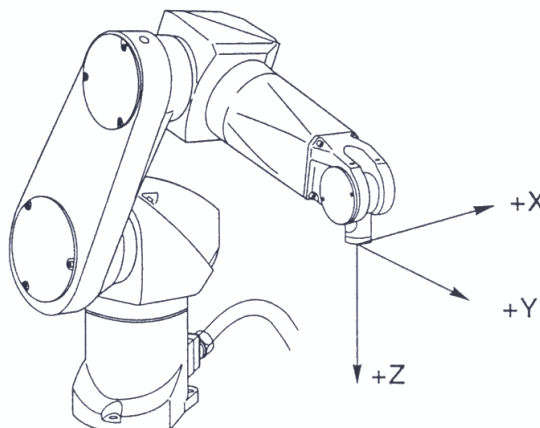
#### ➤ Le mode *World*

Tout déplacement est rapporté aux coordonnées universelles (*World*, concrètement le socle). Les rotations *RX*, *RY*, *RZ* se font par rapport à ces coordonnées (voir figure suivante).



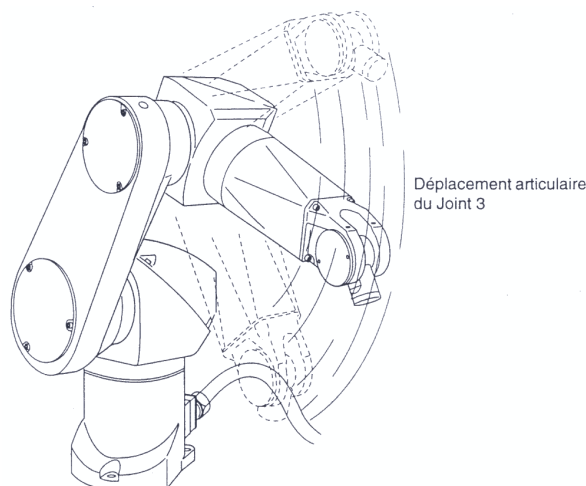
#### ➤ Le mode *Tool*

Tout déplacement est rapporté aux coordonnées d'outil (*Tool*). L'axe *X* est aligné avec la rainure rotative située dans la flasque. Les rotations *RX*, *RY*, *RZ* se font par rapport à ces coordonnées (voir figure suivante).



## ➤ Le mode *Joint*

Le déplacement articulaire s'effectue autour des 6 différents axes (voir figure suivante).



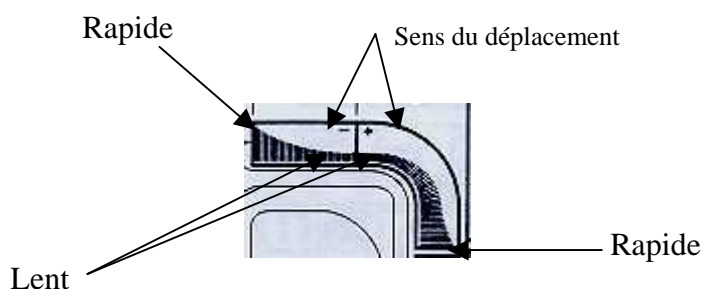
### 6.2 Contrôle du robot à partir du pendant

Les modes de déplacement (cf. §6.1) sont sélectionnés à l'aide de la touche **Man/Halt** du pendant. Après la mise sous tension du système, le système est en mode **World**. L'appui successif sur cette touche permet le passage d'un mode à l'autre (**World** ⇒ **Tool** ⇒ **Joint** ⇒ **World**).

Une fois le mode de déplacement sélectionné, on peut choisir l'axe sur lequel on veut se déplacer à l'aide des touches **X/1**, **Y/2**, **Z/3**, **RX/4**, **RY/5**, **RZ/6**. Les lettres **X**, **Y** et **Z** représentent les 3 axes ; **RX**, **RY**, **RZ** représentent les rotations autour de ces axes. Les chiffres 1 à 6 représentent, dans le contexte du mode **Joint**, les 6 axes du robot.

✓ En ce qui nous concerne, on sélectionnera le mode **Joint**.

La barre de vitesse permet de choisir la vitesse et le sens (+/-) du déplacement manuel :



La touche **Slow** du pendant permet de choisir entre deux plages de vitesses différentes des barres de vitesses : On peut choisir une vitesse de déplacement normale ou lente.

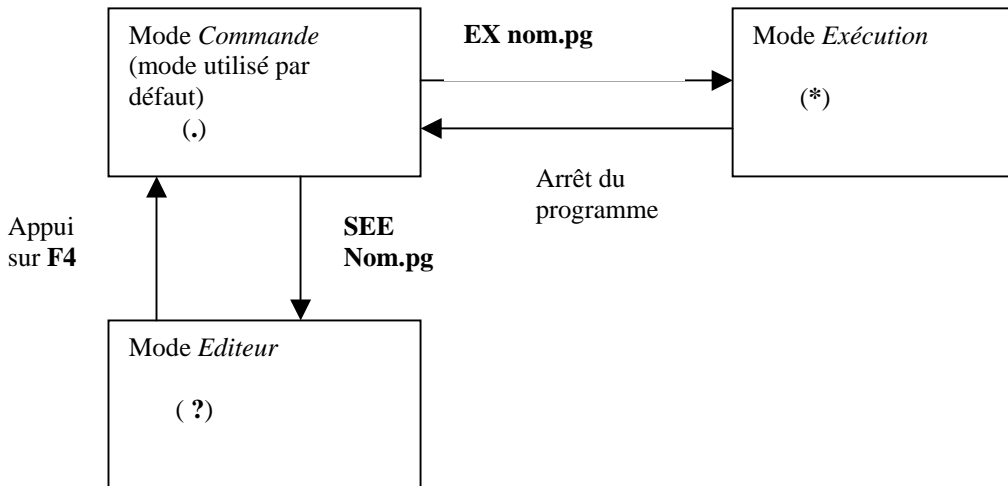
Nous verrons au §6.3.3.b la manière de définir des points de passage (appelés variables *points*) du bras du robot.

### 6.3 Contrôle du robot à partir d'un programme

Il existe 3 prompts à l'écran :

- Le prompt « . » pour le mode *Commande*. On fournit au robot des instructions du type **DIR**, **FDIR**, **FORMAT**, etc...
- Le prompt « \* » pour le mode *Exécution*. On fournit des instructions du type **ABORT**, **EXECUTE**, **PANIC**, **RETRY**, **XSTEP**, etc...
- Le prompt « ? » pour le mode *Editeur*. On fournit des instructions *programme*, ou des commandes *éditeur*.

On peut passer d'un mode à l'autre de la manière suivante :



**Remarque :** Le préfixe **DO com** permet l'exécution de l'instruction **com** en mode *Commande* (par exemple **DO READY**).

### 6.3.1 Le programme du robot existe déjà

Pour exécuter le programme **nom.pg** (situé dans la mémoire vive de la machine), utiliser l'instruction **EXECUTE nom.pg (enter)**. Le nom du programme doit commencer par une lettre et comporte au maximum 15 caractères.

Le fait de taper, avant d'utiliser l'instruction **EXECUTE** ou en cours d'exécution du programme, l'instruction **ENABLE TRACE (enter)** fait apparaître à l'écran les différentes instructions au fur et à mesure de leurs exécutions. L'instruction **DISABLE TRACE (enter)** permet un retour en mode normal.

L'instruction **XSTEP nom.pg (enter)** permet l'exécution du programme en mode pas à pas. Le fait de taper **X (enter)** permet d'exécuter le pas suivant.

Notons que la touche **Run/Hold** du pendant provoque un arrêt *immédiat* du mouvement du bras, ainsi que du programme, alors que l'instruction **ABORT (enter)** tapée au clavier provoque l'arrêt qu'à la fin de l'instruction en cours.

La reprise du cycle se fait *via* l'instruction **RETRY (enter)**.

### 6.3.2 L'éditeur SEE

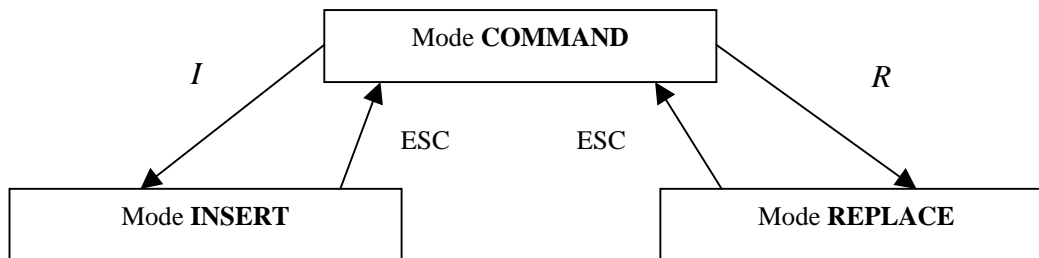
L'instruction **SEE nom.pg (enter)** provoque le passage du mode *Commande* au mode *Editeur* ; **SEE** est un éditeur pleine page qui permet, en insérant, ou modifiant des instructions, de créer un programme. Appuyer sur la touche **F4** pour quitter l'éditeur.

Cet éditeur a 3 modes d'édition : **COMMAND**, **INSERT**, **REPLACE**.

- Le mode **COMMAND** est celui par défaut. Dans ce mode, le code programme n'est pas entré, seules les commandes spéciales éditeur peuvent aboutir.
- Le mode **INSERT** permet à partir de l'emplacement du curseur l'insertion de nouveaux caractères dans le programme.
- Le mode **REPLACE** permet à partir de l'emplacement du curseur le remplacement du texte existant par un nouveau texte.

En mode **COMMAND**, on peut lire les données (il est possible de supprimer un ou des caractères en appuyant sur la touche **D**).

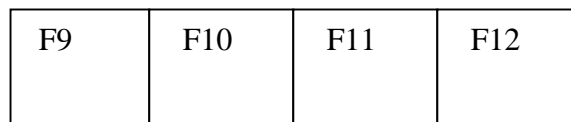
Pour écrire ou remplacer, il suffit de taper sur la touche **I** (mode **INSERT**) ou **R** (mode **REPLACE**). Schématiquement, on a :



**Remarque :** Lors de l'écriture d'un programme, on peut effectuer des *copier/coller* :

<shift-F9> : Coupe la ligne sur laquelle se trouve le curseur

<shift-F10> : Colle le contenu du buffer à l'emplacement du curseur



<F9> : Copie la ligne sur laquelle se trouve le curseur

<F10> : Colle la dernière ligne entrée dans le buffer à l'emplacement du curseur

**Remarque :** En fait, les programmes générés sont sous forme de fichiers texte, ce qui permet l'utilisation d'un traitement de texte tel que le Bloc-notes (*NotePad*) de Windows.

### 6.3.3 Quelques commandes

La programmation du robot se fait *via* le langage V+, qui est un langage de programmation interprété, structuré de haut niveau, multitâche. Seuls quelques éléments de base sont donnés.

Le signe « ; » permet d'insérer un commentaire.

#### a) Éléments standard de programmation

Listons tout d'abord les éléments standard à un langage :

1) Les *identificateurs* commencent obligatoirement par un caractère alphabétique, suivi d'un nombre quelconque de lettres, chiffres ou points (qui jouent le rôle du « \_ » en Pascal, C). Ils n'ont pas à être déclarés, le type étant donné par le contexte (3 types sont possibles (voir ci-dessous)).

2) Il existe différentes classes d'*opérateurs*, notamment des opérateurs :

- mathématiques : +, -, \*, /, MOD,
- relationnels : <, <=, =, >, >=, >,
- logiques : AND, OR, NOT.

3) *Structures algorithmiques* (les plus classiques)

L'instruction **GOTO** permet une exécution non séquentielle des instructions. Elle permet, par exemple, la création de boucle *via* l'utilisation de *label* et de l'instruction **GOTO label**.

Structure **CASE ... OF**

```
CASE 2*i + 1 OF
VALUE 1, 2 :
:
VALUE 3 :
:
ANY;    (sans mettre “:”)
:
END
```

Structure **IF ... THEN ... ELSE ... END**

```
IF (i MOD 2 == 0) OR (i > j) THEN
:
ELSE
:
END
```

Structure **FOR ... END**

```
FOR i = max TO min STEP -1
T[i]=i
END
```

Structure **WHILE ... DO ... END**

```
WHILE (i < max) AND (t[i] <> 0) DO
:
END
```

Structure **DO ... UNTIL**

```
DO
:
UNTIL t[i] == 0
```

L'instruction **WAIT condition** interrompt l'exécution du programme jusqu'à ce que l'expression *condition* (par exemple, **SIG 1002**) soit valide.

## b) Déclarations de variables

Il existe **3 types de variables** :

- Une variable de type **point** permet de mémoriser un *point* du bras du robot, défini par 6 paramètres indépendants : 3 pour la position, 3 pour l'orientation. Une variable de type **point** peut être considérée comme un repère donné par son origine et son orientation, ou comme une transformation entre deux repères.

Une manière simple pour mémoriser dans une variable **A** (de type **point**) *le point de passage courant* du bras du robot (atteint par exemple *via* le pendant) consiste à taper au clavier :

**HERE A (enter)**

X/JT1 Y/JT2 Z/JT3 y/JT4 p/JT5 r/JT6  
120 12 -80 85 -40 0

**CHANGE ? (enter)**

L'instruction **SET B = A** permet d'affecter à la variable point **B** la valeur **A** (qui est sous-entendu un point).

L'instruction **SET C = HERE** permet d'affecter à la variable point **C** *le point de passage courant* du bras du robot.

L'instruction **SET loc1 = TRANS(550, 450, 750, 0, 180, 45)** permet d'affecter à la variable point **loc1** le point correspondant aux *coordonnées opérationnelles*  $X = 550\text{ mm}$ ,  $Y = 450\text{ mm}$ ,  $Z = 750\text{ mm}$ ,  $y(aw) = 0^\circ$ ,  $p(itch) = 180^\circ$ ,  $r(oll) = 45^\circ$ .

L'instruction **SET #loc2=#PPOINT(80, -20, 120, 0, 76, -182)** permet d'affecter à la variable point **#loc2** le point correspondant aux *coordonnées articulaires*  $JT1 = 80^\circ$ ,  $JT2 = -20^\circ$ , ...,  $JT6 = -182^\circ$  (attention, les # sont obligatoires et il n'y a pas d'espace de part et d'autre du signe =).

L'instruction **SET arrivee = DEST** permet d'affecter à la variable point **arrivee** *le point de destination courante* du bras du robot.

- Les variables **réelles** ou **entières** (entre -16 777 216 et 16 777 215) :

Exemples : **A = 12**

**B = 2.356**

**C = B**

- Les variables **chaînes** définies par un \$ en première position :

Exemple : **\$réponse = « Répondre par Oui ou Non »**

### c) Commandes associées à la mémoire vive

Les programmes, ainsi que les variables, doivent être placés dans la mémoire vive de la machine afin d'être exécutés.

**Pour visualiser les titres** des programmes situés dans le répertoire courant, taper **DIR (enter)**.

**Pour visualiser les contenus** (instructions disponibles uniquement en mode *Commande*) :

- du programme **nom.pg**, taper : **LISTP nom.pg (enter)**,
- des coordonnées des points **A** et **B**, taper : **LISTL A,B (enter)**,
- des variables réelles **X** et **Y**, taper : **LISTR X, Y (enter)**,
- des variables chaînes **\$réponse** et **\$nom**, taper : **LISTS \$réponse, \$nom (enter)**.

**Pour copier un programme**, taper **COPY nom du nouveau prog = nom de l'ancien prog (enter)**

Pour renommer un programme, taper **RENAME nom du nouveau prog = nom de l'ancien prog**  
(enter)

Pour supprimer :

- le programme **Nom.pg**, taper : **DELETEP nom.pg (enter)**,
- les coordonnées du point **nom\_point**, taper : **DELETEL nom\_point (enter)**,
- la variable réelle **var\_réelle**, taper : **DELETER var\_réelle (enter)**,
- la variable chaîne **\$var\_chaine**, taper : **DELETES \$var\_chaine (enter)**.

Pour vérifier l'existence du programme **toto.pg**, taper : **TESTP toto.pg (enter)**.

L'instruction **ZERO** permet de vider le contenu total de la mémoire vive.

#### d) Opérations de sauvegarde sur une disquette ou un disque dur

Les programmes, ainsi que les variables, peuvent être sauvegardés sur une disquette et/ou un disque dur.

Pour sauvegarder, sur une disquette et/ou un disque, un programme, appelé **nom**, et ses variables associées éventuelles, on utilise l'instruction **STORE nom**. Il en résulte un fichier **nom.V2**.

On utilise les instructions **STOREP**, **STOREL**, **STORER**, **STORES** pour sauvegarder uniquement respectivement des programmes, des points, des réels, des chaînes.

Pour charger un fichier (situé sur une disquette et/ou un disque) en mémoire vive, on utilise l'instruction **LOAD**. Par exemple :

**LOAD nom.pg** charge le fichier **nom.pg** situé dans le répertoire courant en mémoire vive.

L'instruction **FDIRECTORY** permet :

- De visualiser les fichiers contenus dans le répertoire courant.
- De créer un répertoire à partir du répertoire courant :  
*Exemple* : Créer le répertoire **c:\DESS\TD**, sachant que le répertoire courant est **c:\**, taper :  
**FDIR/C c:\DESS\**

puis  
**FDIR/C c:\DESS\TD\**

- De supprimer un répertoire à partir du répertoire courant:  
*Exemple* : Supprimer le répertoire **TD**, situé dans le répertoire **DESS**, taper :  
**FDIR/D c:\DESS\TD\**

L'instruction **CD** permet de changer de répertoire du disque dur (c :) ou de la disquette (a :), par exemple :

**CD=c:\Repertoire1\Repertoire2\**

**Remarque** : **CD ..** permet de remonter d'un répertoire.

L'instruction **FLIST** permet de visualiser le contenu d'un fichier (situé sur une disquette et/ou un disque), par exemple :

**FLIST c:\nom.pg**

L'instruction **FDELETE** permet de supprimer un fichier (situé sur une disquette et/ou un disque), par exemple :

**FDELETE nom.pg**

L'instruction **FCOPY** permet de **copier un fichier** (situé sur une disquette et/ou un disque), par exemple :

**FCOPY fichier cible = fichier source**

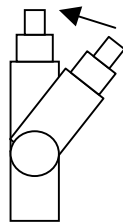
L'instruction **FRENAME** permet de **changer le nom d'un fichier** (situé sur une disquette et/ou un disque), par exemple :

**FRENAME nouveau nom fichier = ancien nom fichier**

#### e) Quelques instructions *programme*

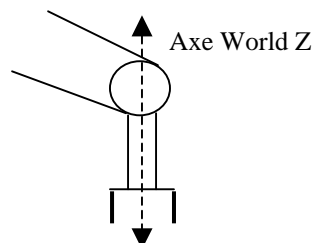
#### Les mouvements \_\_\_\_\_

- ✓ **(DO) READY (DO** permet une exécution en mode *Commande*) remet le bras du robot dans sa configuration initiale.



- ✓ L'instruction **WHERE** permet de connaître la configuration du robot et l'état du préhenseur au moment où cette instruction est exécutée. Par exemple, si la variable **A** de type point est présente, on peut afficher ses valeurs *via* l'instruction **WHERE A**.

- ✓ L'instruction **ALIGN** permet de placer l'axe Z du repère outil (terminal) parallèlement à l'axe le plus proche du référentiel de base (World) (à savoir X, Y ou Z). Cette instruction évite de faire de nombreuses manipulations *via* le pendant afin de rendre l'axe Z du repère outil colinéaire avec, par exemple, l'axe Z du repère de base (World) :

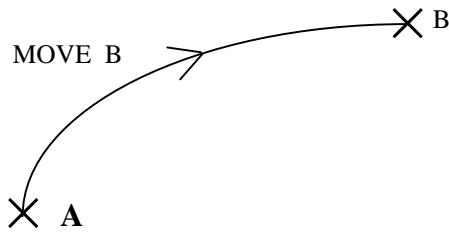


- On spécifie toujours la destination du repère terminal, et éventuellement des contraintes sur la façon d'atteindre la destination (par exemple en ligne droite). La destination est toujours un point (ou repère), ainsi le mouvement impose d'atteindre une position donnée, avec une orientation donnée. Il n'existe pas de moyen de demander un mouvement en laissant libre, par exemple, l'un des angles d'orientation.

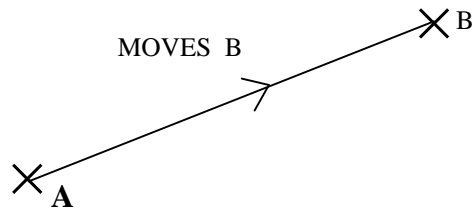
- ✓ Cinq types de primitives de mouvement sont listés ci-dessous : **MOVE(S)**, **APPRO(S)**, **DEPART(S)**, **DRIVE**, **DELAY**.

- 1) L'instruction **MOVE point** (précédée de **DO** si exécution en mode commande) ordonne un déplacement du repère outil du robot vers le point spécifié selon une trajectoire non imposée par l'utilisateur (dépendante de la morphologie du manipulateur) :

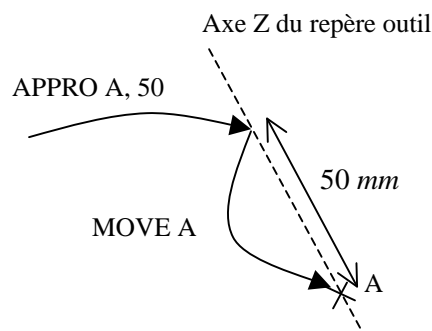




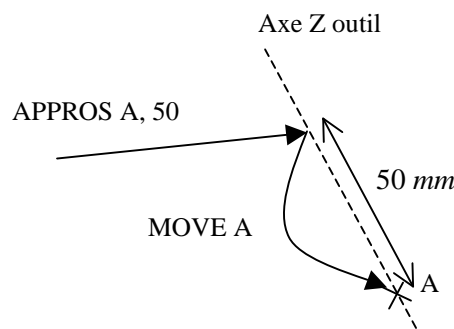
L'instruction **MOVES point** (précédée de **DO** si exécution en mode commande) est similaire à l'instruction précédente, mais est telle que la trajectoire de l'origine du repère terminal est *une droite* :



- 2) L'instruction **APPRO point, distance**, généralement suivie de l'instruction **MOVE**, permet au robot d'approcher le point spécifié, en restant en retrait de la distance spécifiée (en *mm*), selon l'axe Z du repère outil (voir lien avec l'instruction **ALIGN**).

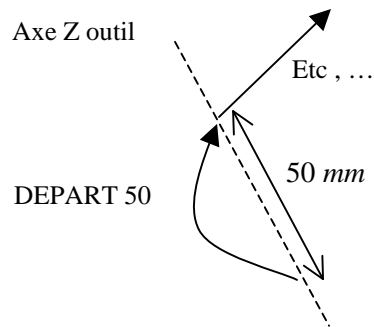


L'instruction **APPROS** est similaire à l'instruction précédente, mais est telle que la trajectoire de l'origine du repère terminal est *une droite* :



- 3) L'instruction **DEPART distance** réalise la fonction inverse de l'instruction **APPRO** ; elle permet de se dégager de la distance indiquée (en *mm*) selon l'axe Z du repère outil, par rapport à la position courante (*i.e.*, la position juste avant l'exécution de l'instruction **DEPART**).

Soit **B** la position juste avant l'exécution de l'instruction « **DEPART 50** ». Soit **A** la position du repère outil résultant du mouvement, alors l'instruction « **APPRO A, 50** » permet de retourner en **B**.



L'instruction **DEPARTS** est similaire à l'instruction précédente, mais est telle que la trajectoire de l'origine du repère terminal est *une droite*.

- 4) L'instruction **DRIVE articulation, angle, pourcentage de vitesse** (précédée de **DO** si exécution en mode commande) permet d'assurer un déplacement articulaire suivant l'articulation désignée. Par exemple :

**DRIVE 1, 30, 50**

crée un déplacement suivant l'articulation 1 de 30 degrés à 50% de la vitesse moniteur.

- 5) **DELAY temps** permet au robot de s'arrêter pendant le temps stipulé dans l'instruction. Ce temps doit être exprimé en secondes et ne peut être inférieur à 16 ms.

Soit par exemple, l'instruction : **DELAY 30**

- L'instruction **DELAY** est considérée comme une instruction de mouvement sans déplacement (soit encore un mouvement vers le point courant).

- ✓ Instructions de mouvements de la pince :

<b>OPENI</b>	; permet d'ouvrir la pince,
<b>CLOSEI</b>	; permet de fermer la pince.

---

**Important :** Par défaut les mouvement du bras du robot sont *asynchrones*, autrement dit une *instruction de mouvement se termine dès que le mouvement est entamé*. Aussi, le programme se poursuit avec l'évaluation de l'instruction suivante, sauf si celle-ci est elle-même une instruction de mouvement (rappelons que c'est le cas de l'instruction **DELAY**). Cette notion - qui existe dans tous langages de programmation de haut niveau de robots - constitue une différence essentielle entre la programmation d'un robot et la programmation d'une application informatique classique.

---

- ✓ Pour que le programme ne se poursuive qu'après achèvement du mouvement, il faut le demander explicitement à l'aide de l'instruction **BREAK**. En effet, cette instruction suspend l'exécution du programme jusqu'à ce que le mouvement ait atteint sa destination. Cette instruction est utile pour

éviter l'exécution d'instructions de calculs, de gestions entrées/sorties, ... (autres que des instructions de mouvements), situées entre deux mouvements.

Comparons par exemple les 2 portions de code suivantes :

**MOVE A**

**TYPE "le mouvement commence"**

**MOVE A**

**TYPE "le mouvement commence"**

**BREAK**

**TYPE "le mouvement est terminé"**

- ✓ On peut également donner des indications de vitesse, soit en pourcentage de la vitesse maximale pour un mouvement en articulaire, soit en millimètres par seconde pour un mouvement en cartésien.

Il est possible d'introduire, au sein du programme, des instructions de vitesse. Ces vitesses sont systématiquement multipliées par la vitesse dite « moniteur », introduite au clavier avant l'exécution du programme.

Ainsi, si la vitesse indiquée dans le programme est de 40 % alors que la « vitesse moniteur » est de 80 %, alors la vitesse résultante du manipulateur sera de 32 % de la vitesse nominale du robot. Il en va de même pour une vitesse exprimée en *mm/sec* (notez que celle-ci sera respectée que pour une « vitesse moniteur » égale à 100 %).

Par exemple :

- **SPEED 50 ALWAYS**  $\Rightarrow$  vitesse de 50 % de la « vitesse moniteur » pour un mouvement en articulaire. Si **ALWAYS** n'est pas indiqué, seul le prochain mouvement en articulaire est concerné. Sinon, l'indication est valable jusqu'à la prochaine indication de vitesse en articulaire.
- **SPEED 500 MMPS ALWAYS**  $\Rightarrow$  vitesse de 500 *mm/sec* pour un mouvement en cartésien. Si **ALWAYS** n'est pas indiqué, seul le prochain mouvement en cartésien est concerné. Sinon, l'indication est valable jusqu'à la prochaine indication de vitesse en cartésien.

## Les sous-programmes \_\_\_\_\_

Seule la notion de sous-programmes sans paramètres existe. Un sous-programme est défini par un identificateur et se termine par **RETURN**. L'appel du sous-programme s'effectue par l'instruction **CALL <nom\_programme>**. Toutes les variables sont globales.

Par exemple, l'instruction :

**CALL prg.pg ( )**

permet d'appeler le programme **prg.pg**.

## Entrées/sortie écran/clavier \_\_\_\_\_

La commande **PROMPT** permet la lecture de données introduites au clavier.

Par exemple, l'instruction :

**PROMPT « Donner la position : », x, y, z**

est telle que le message « **Donner la position :** » s'affiche à l'écran et le programme attend que trois valeurs soient introduites avant de poursuivre.

La commande **TYPE** permet d'écrire à l'écran un message.

Par exemple, l'instruction :

**TYPE « La vitesse est : », v, « et l'accélération est : », gamma**  
permet l'écriture à l'écran du message entre « », accompagné des valeurs des variables **v** et **gamma**.

## Entrées/sorties digitales (tout ou rien) \_\_\_\_\_

En général, un robot interagit avec d'autres systèmes. C'est le cas du robot situé en salle 215, au sens où le robot communique avec le système de transfert.

### ✓ **Programmation des sorties :**

Les adresses des sorties sont 1 pour la sortie n°1, 2 pour la sortie n°2, ..., 8 pour la sortie n°8 (sachant que les sorties 7 et 8 sont généralement réservées et que le nombre de sorties peut être étendu).

L'instruction **SIG** permet d'écrire (d'actionner le contact) sur une sortie digitale. Par exemple, l'instruction :

**SIG 2, -1, 4**  
permet de mettre à l'état 1 les sorties n° 2 et 4, et à l'état 0 la sortie 1.

L'instruction **RESET** remet à zéro toutes les sorties.

### ✓ **Les entrées :**

Les adresses des entrées sont 1001 pour l'entrée n°1, 1002 pour l'entrée n°2, ..., 10012 pour l'entrée n°12 (leur nombre peut être étendu).

L'instruction **SIG** permet de lire l'état des entrées digitales.

Par exemple, l'instruction :

**SIG 1002, 1004**  
permet la lecture des entrées 2 et 4.

### ✓ **Scrutation des entrées :**

<b>WAIT SIG(1001)</b>	; attendre que <i>E1</i> soit à l'état 1
<b>WAIT SIG(1001) AND SIG(-1002)</b>	; attendre que <i>E1</i> soit à l'état 1 et <i>E2</i> à l'état 0
<b>WAIT SIG(1001) OR SIG(1002)</b>	; attendre que <i>E1</i> ou <i>E2</i> soit à l'état 1
<b>IF SIG(1001) THEN ...</b>	; si <i>E1</i> est à l'état 1 alors ...

### ✓ **Réaction aux entrées :**

L'instruction

**REACT 1001, SPRG1**

permet l'exécution du sous-programme **SPRG1** si *E1* est à l'état 1. Le mouvement en cours ne sera pas interrompu. Si le sous-programme contient des instructions de mouvement, elles ne seront exécutées qu'à la fin du mouvement en cours. Le retour au programme interrompu s'effectue sur l'instruction qui suit l'instruction après laquelle le déroutement a eu lieu.

L'instruction

### REACTI 1002, SPRG2

permet d'aller **immédiatement** exécuter le sous-programme **SPRG2** si *E2* est à l'état 1. Le mouvement en cours est interrompu. Après exécution de **SPRG2**, le programme interrompu est poursuivi à l'instruction suivante, aussi le mouvement interrompu n'est pas terminé. Pour terminer le mouvement, le programmeur peut, par exemple, mémoriser la destination au moment de l'interruption et demander l'exécution du mouvement à la fin du sous-programme **SPRG2**.

## 7 DESCRIPTION PRATIQUE DES ROBOTS FANUC LR MATE 100 IB ET ARC MATE 100 IB

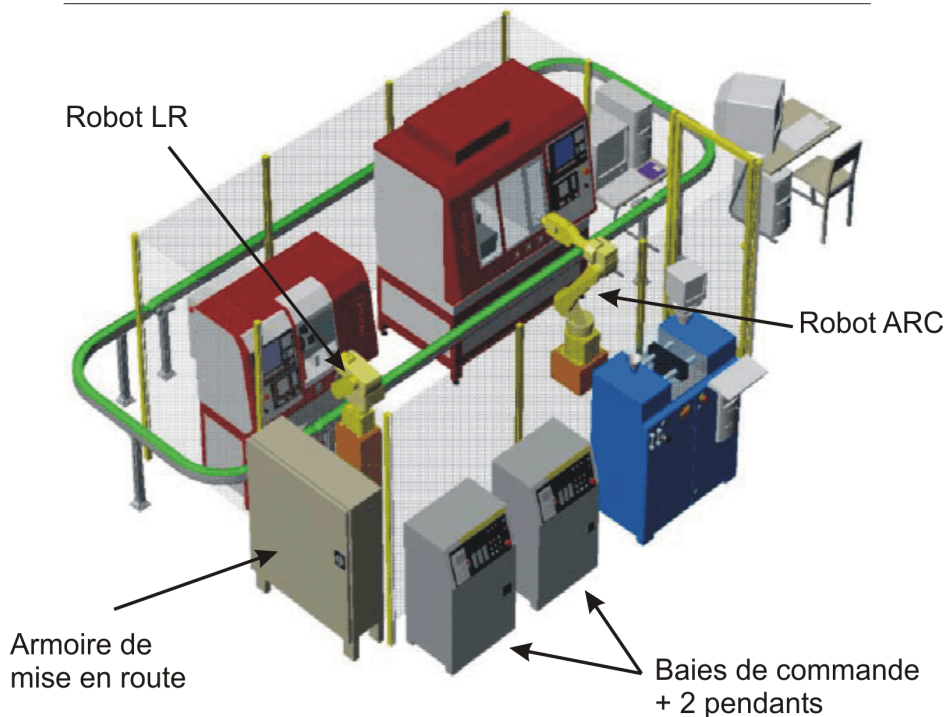
Robot 5 axes LR Mate 100 IB



Robot 6 axes ARC Mate 100 IB



Ces deux robots se situent au sein de la chaîne de production du hall de technologie :



## 7.1 Description générale

<p>Le robot FANUC <b>LR Mate 100 IB</b> est un robot électrique 5 axes destiné au chargement déchargement de machine, à la soudure et à la manutention. Sa petite taille et ses capacités sont adaptées pour les petites applications industrielles et les utilisations en laboratoire. Le robot LR Mate 100 IB est conçu pour être fiable dans les environnements sévères.</p>	<p>Le Robot FANUC <b>ARC Mate 100 IB</b> est un robot électrique 6 axes destiné aux applications, précises et rapides, de soudure et découpe. Basé sur une construction simple et fiable, le robot ARC Mate 100 IB dernière génération de robot soudure ARC, a un design compact avec des vitesses et une plage de mouvement accrues. La conception flexible et compacte simplifie l'intégration, augmente les possibilités d'accès dans des espaces confinés et permet une installation dense de robots et de leurs périphériques.</p>
---	---

### Quelques caractéristiques :

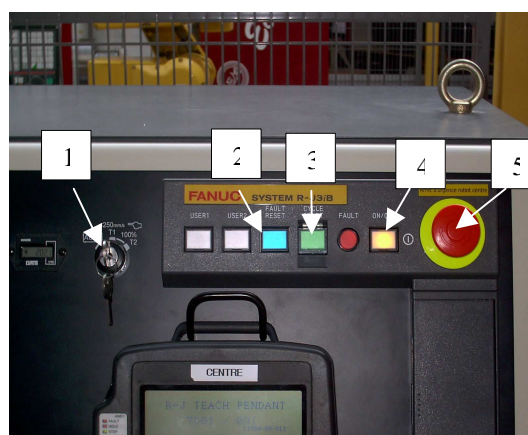
<b><i>Robot LR Mate 100 IB</i></b>	<b><i>Robot ARC Mate 100 IB</i></b>
Charge maximale : 3-5 kg Charge à l'axe J3 : 12 Kg Rayon d'action (mm) : 620 Répétabilité : +/- 0.04 mm	Charge maximale : 6 kg Charge à l'axe J3 : 12 Kg Rayon d'action (mm) : 1373 Répétabilité : +/- 0.08 mm
Plage de mouvement (°)    Vitesse nominale (°/s) J1 : 320                      240 J2 : 185                      270 J3 : 365                      240 J4 : 240                      330 J5 : 400                      480	Plage de mouvement (°)    Vitesse nominale (°/s) J1 : 340                      150 J2 : 250                      160 J3 : 315                      170 J4 : 380                      400 J5 : 280                      400 J6 : 720                      520

### Face avant de la baie de commande



1. Sélecteur des 3 modes (AUTO, T1, T2)
2. Bouton d'arrêt d'urgence
3. Sectionneur robot tour (interrupteur de sécurité)

### Face avant de la baie de commande



1. Sélecteur des 3 modes (AUTO, T1, T2)
2. Relâchement de l'alarme
3. Bouton de départ
4. Bouton de mise sous tension
5. Bouton d'arrêt d'urgence

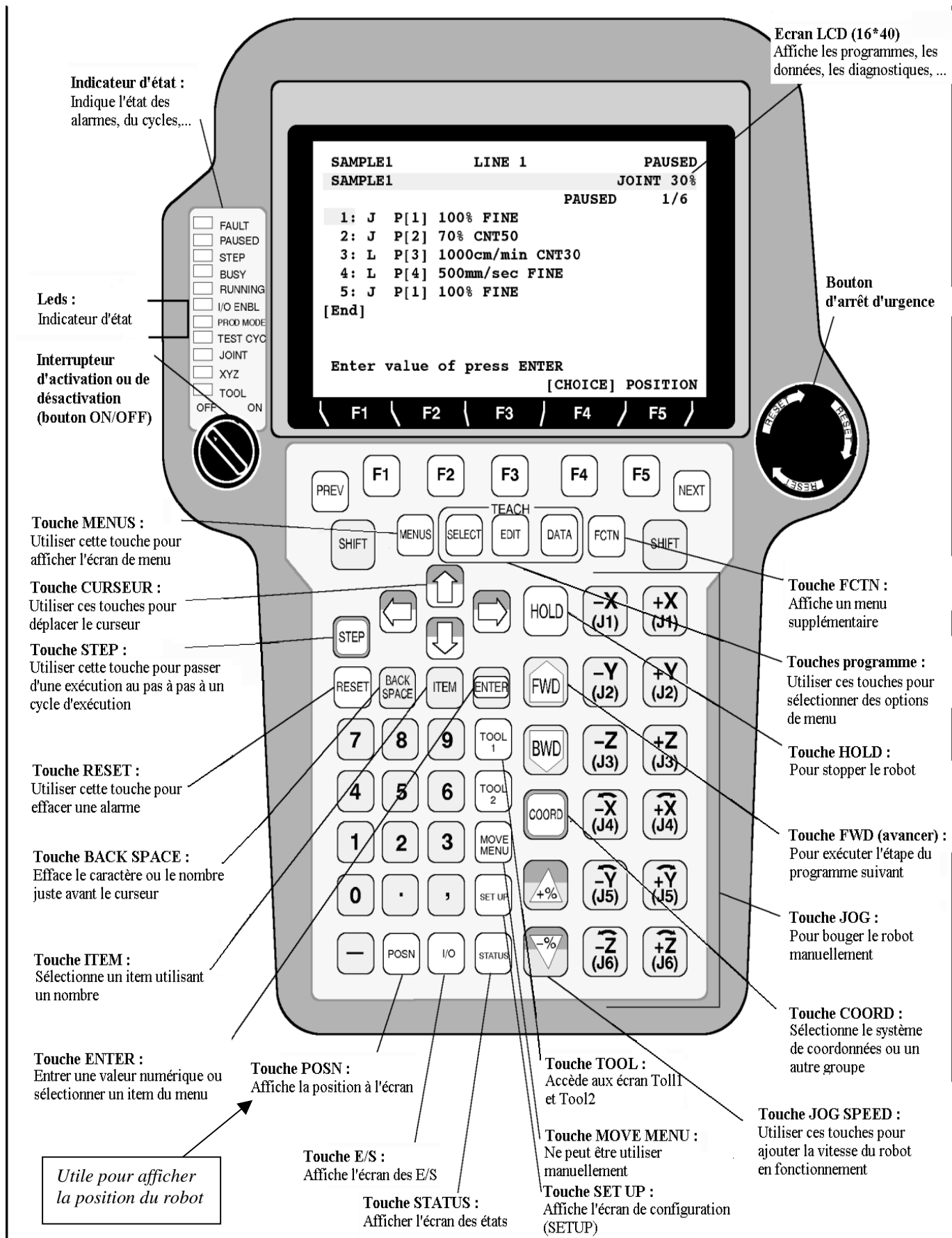
### Description du pendant

Le *pendant* est connecté à la carte PC, située dans le contrôleur, par un câble et constitue l'interface entre le logiciel *Handling tool* et l'opérateur.

Les opérations suivantes peuvent être réalisées avec le pendant :

- Bouger le robot manuellement (cf. 8.2.)
- Créer un programme (cf. 8.3.3.)
- Exécuter un test (cf. 8.3.1.)

NB : Les codes d'erreurs et remèdes sont accessibles dans les annexes A1-A2-A3 du « Manuel de mise en service du FANUC ».





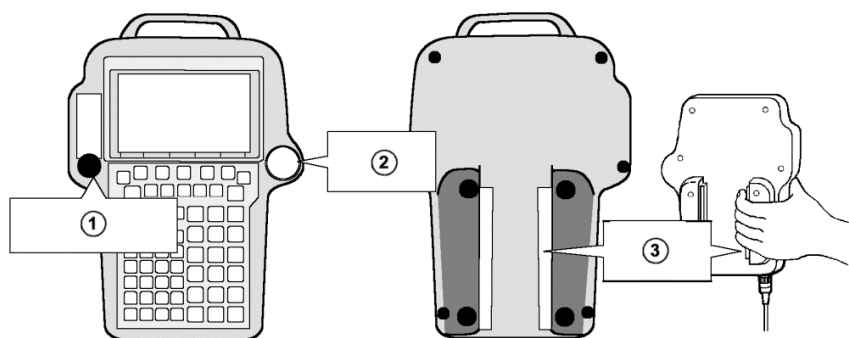
## 7.2 Mise en route du système

### Règles de sécurité :

- Personne ne doit être dans l'espace de travail du bras du robot.
- La phase de test du robot s'effectue toujours en vitesse réduite : **Se mettre en mode T1 (vitesse < 250 mm/s).**
- Être prêt à appuyer sur le **bouton arrêt d'urgence** durant la phase de test du robot.

### Le corps mort :

L'interrupteur du *corps mort* (voir ③ figure ci-dessous) est utilisé comme un moyen d'activer les mouvements. Lorsque le pendant est activé, il faut appuyer sur l'interrupteur du *corps mort* pour autoriser les mouvements. En le relâchant, le robot stoppe immédiatement, c'est également le cas lors d'un appui sur le bouton d'arrêt d'urgence ②. Noter que le robot stoppe également lorsque l'interrupteur du *corps mort* est trop comprimé par l'opérateur.



### Pour entrer dans l'enceinte :



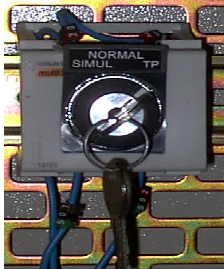

Pour accéder aux robots, maintenir appuyer le bouton **“demande d'accès”** situé à côté de la porte et tirer la manette de manière à ouvrir la porte. L'ouverture de la porte empêche le mouvement du bras des robots (une lumière rouge apparaît au dessus de l'armoire de mise en route).

Pour permettre la mise en mouvement du bras des robots : fermer la porte, puis ré-armer l'enceinte à l'aide du bouton vert situé à quelques pas et à droite de la porte (la lumière rouge s'éteint).

**Attention :** En mode **T1** (voir sélecteur situé sur la baie de commande), le bras des robots peut être mis en mouvement à vitesse réduite lorsque la porte est ouverte !! Ceci permet d'être proche du bras du robot et ainsi de programmer des mouvements précis.

### Mise en route générale :

#### 1. ARMOIRE DE MISE EN ROUTE (GAUCHE)

Sur le côté de l'armoire, mettre « l' <b>Intersectionneur transitique</b> » sur la position « <b>ON</b> ».	
Sur la face avant de l'armoire, appuyer sur le bouton vert « <b>Mise en énergie</b> ».	
Ouvrir l'armoire du bas et positionner le commutateur sur « <b>TP</b> » à l'aide de la clé.	
Sur le côté de l'armoire, ouvrir la vanne de l'air comprimé ( <b>celle du bas !!</b> ).	

<b>2. BAIE DE COMMANDE</b>	
Positionner la clé (la même pour les 2 baies de commande) sur « <b>T1</b> ».	
Positionner le « <b>Sectionneur robot</b> » sur « <b>ON</b> ».	
Uniquement pour le robot ARC Mate 100 IB (robot de droite) : Appuyer sur le bouton de mise sous tension.	

### 3. PENDANT

Positionner l'interrupteur d'activation et de désactivation (cf. §7.2, 1<sup>er</sup> figure - ①) sur « **ON** ».

## 7.3 Arrêt du système

Afin d'arrêter le système, il faut :

- Exécuter le programme *TP* afin de remettre le robot en configuration initiale.

Au niveau du **pendant** :

- Positionner l'interrupteur d'activation et de désactivation sur « **OFF** ».

Au niveau de la **baie de commande** :

- Uniquement pour la baie de commande du robot ARC Mate 100 IB (6 axes) : Appuyer sur le bouton de mise hors tension.
- Mettre le “Sectionneur robot” sur la position « **OFF** ».
- Positionner la clé (la même pour les 2 baies de commande) sur « **AUTO** ».

Au niveau de l'**armoire de mise en route** :

- Sur le côté de l'armoire, fermer la vanne de l'air comprimé.
- Ouvrir l'armoire de mise en route et positionner le commutateur sur « **SIMUL** » à l'aide de la clé.
- Sur le côté de l'armoire, mettre l'intersectionneur transitique sur la position « **OFF** ».

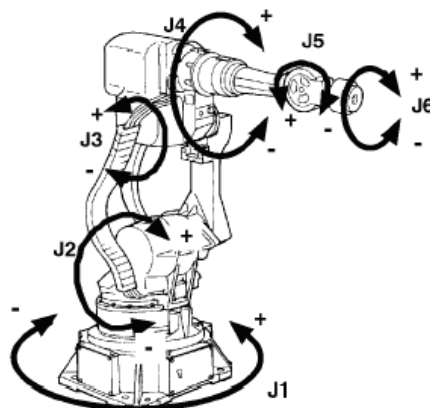
## 8 MISE EN MOUVEMENT DU BRAS DES ROBOTS FANUC LR MATE 100 IB ET ARC MATE 100 IB

Une fois la mise en route générale effectuée, on dispose de 2 moyens pour mettre le bras en mouvement : soit manuellement *via* le pendant, soit automatiquement *via* l'exécution d'un programme (à l'aide du terminal). Décrivons les différents modes de déplacements possibles du bras.

### 8.1 Les modes de déplacement

#### ➤ *Le mode JOINT*

Dans un déplacement *JOINT*, on peut déplacer chaque axe du robot indépendamment.

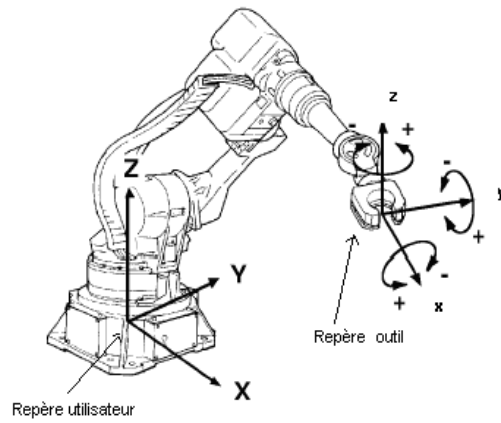


Robot ARC Mate 100 IB

**Remarque :** Le déplacement de J2 entraîne un déplacement de J3.

#### ➤ *Le mode XYZ - USER*

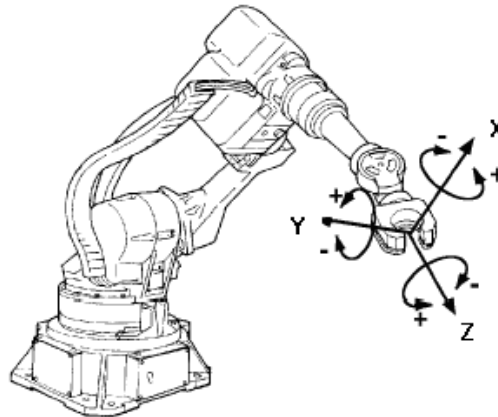
Dans un mouvement cartésien, le centre outil du robot effectue des translations selon les axes X, Y et Z du repère utilisateur. Le robot effectue des rotations d'outil autour des axes x, y et z du repère outil.



Robot ARC Mate 100 IB

### ➤ *Le mode TOOL*

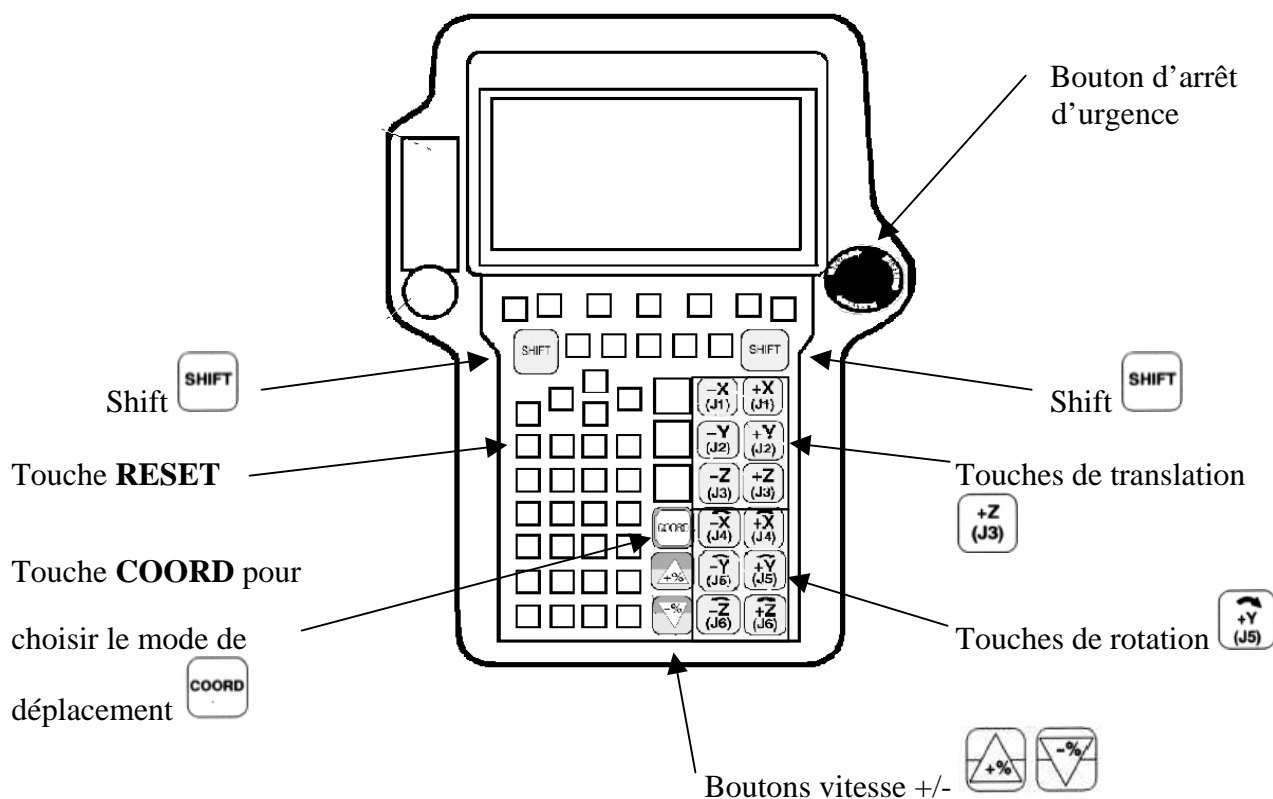
Lors du mouvement de l'outil, le centre outil (TCP) se déplace selon les axes  $X$ ,  $Y$  et  $Z$  du système de coordonnées outil défini pour le poignet du robot. Le robot effectue des rotations d'outil autour des axes  $x$ ,  $y$  et  $z$  du repère outil. Le repère est fixé à l'outil, les déplacements se font par rapport à l'outil.



Robot ARC Mate 100 IB

## 8.2 Mise en mouvement en mode manuel

## Les touches du mode *manuel* :



## Fonctionnement :

1. Prendre le pendent en appuyant sur l'interrupteur du corps mort.
2. Les modes de déplacement sont sélectionnés à l'aide de la touche **COORD** du pendent, l'appui successif permettant le passage d'un mode à l'autre (*JOINT*, *USER*, *TOOL*). Le système de coordonnées sélectionné est affiché dans le coin supérieur gauche du pendent.
3. Appuyer sur la touche **RESET** pour enlever le mode **FAULT** (indicateur situé en haut et à gauche du pendent) dû au déclenchement d'une alarme.
4. On peut choisir l'axe sur lequel on veut se déplacer **en maintenant** la touche **SHIFT** et les touches de rotation, ou de translation, selon le mode de déplacement choisi.

Touches de translation :

$-Z$ (J3)	$-Y$ (J2)	$-X$ (J1)	$+Z$ (J3)	$+Y$ (J2)	$+X$ (J1)
--------------	--------------	--------------	--------------	--------------	--------------







Touches de rotation :

$-Z$ (J6)	$-Y$ (J5)	$-X$ (J4)	$+Z$ (J6)	$+Y$ (J5)	$+X$ (J4)
--------------	--------------	--------------	--------------	--------------	--------------

## Remarques :

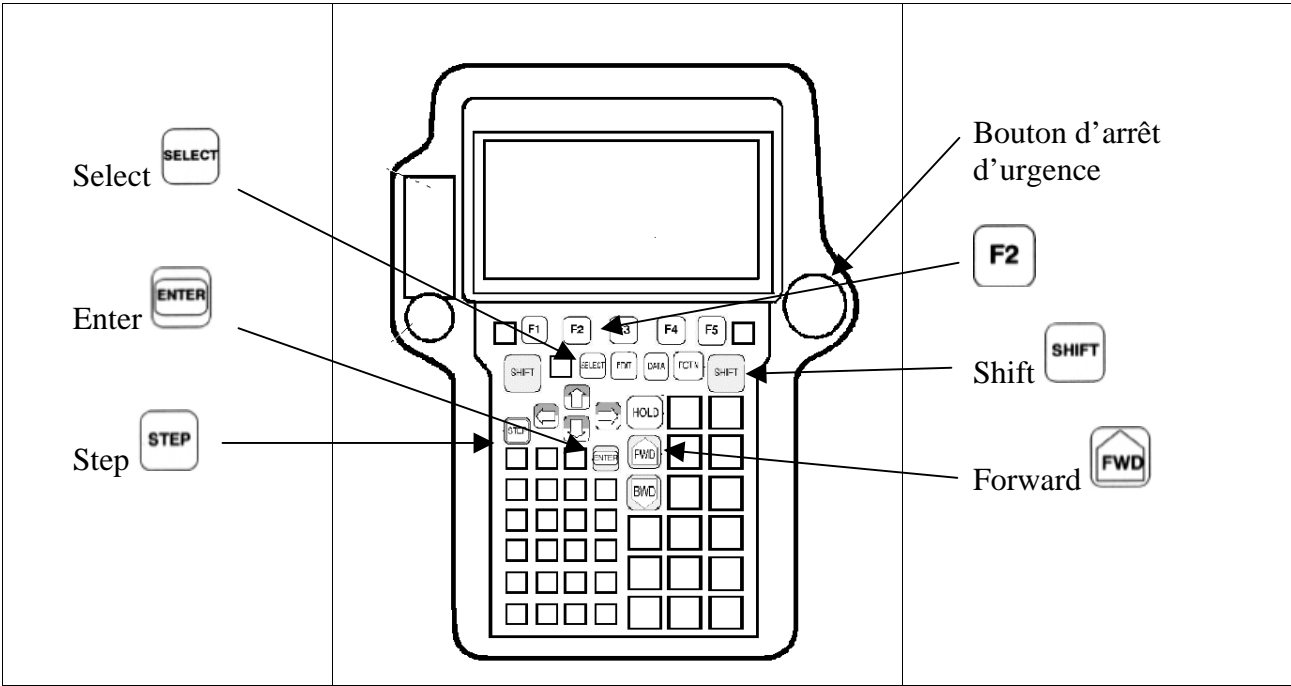
- Les touches  $+%$  et  $-%$  ajustent la vitesse du mouvement sélectionné.
- Appuyer sur la touche **POSN** pour connaître la position du robot.

## Exemples :







<i>JOINT</i> (articulaire)	<i>USER</i> (cartésien)	<i>TOOL</i> (outil)
<b>Rotation de la 3ème articulation :</b> Appuyer sur  + 	<b>Translation selon l'axe Ox du repère fixe :</b> Appuyer sur  + 	<b>Rotation selon l'axe Oy du repère outil :</b> Appuyer sur  + 

### 8.3 Mise en mouvement en mode programme

Les touches du mode *programme* :







#### 8.3.1 Exécution d'un programme

- Appuyer sur la touche  pour accéder à la liste des programmes.
- Se positionner sur le programme souhaité à l'aide des touches   et presser la touche  pour visualiser les lignes du programme sélectionné.
- Appuyer simultanément sur les touches  et  pour exécuter le programme.



Exemples d'exécution de programmes :

Robot LR Mate 100 IB	Robot ARC Mate 100 IB
Exécuter le programme <i>TP</i>	Exécuter le programme <i>Prgtest1</i>



Remarque : Il est possible d'exécuter un programme **en mode pas à pas** en sélectionnant le mode **step**, pour cela :

1. Appuyer sur la touche . Le voyant **STEP** s'allume en haut à gauche du pendant.
2. Appuyer simultanément sur les touches  et  pour exécuter une ligne du programme.
3. Appuyer sur la touche  pour quitter le **mode pas à pas**.

### 8.3.2 Arrêt d'un programme


Le relâchement de la touche  suspend l'exécution du programme. On sort du programme en appuyant sur le bouton , il s'ensuit un accès à la liste des programmes disponibles.

### 8.3.3 Création d'un programme


Pour accéder au menu des programmes, appuyer sur la touche . Appuyer sur la touche  - **CREATE** pour créer un nouveau programme.


Pour entrer le nom du programme, utiliser les flèches   de manière à avoir accès aux options :

- *words*, ce qui permet la sélection d'un mot prédéfini,
- *upper case*, ce qui permet la sélection de lettres majuscules,
- *lower case*, ce qui permet la sélection de lettres minuscules,
- *option*, ce qui permet d'insérer, d'effacer, ...

Valider le nom du programme par un appui sur la touche .

Appuyer sur la touche  - **EDIT** pour éditer un programme.

Appuyer sur la touche  - **POINT** pour afficher le menu d'instruction standard de mouvements.

Sélectionner l'instruction de mouvement souhaitée en appuyant sur la touche . Il reste à définir les différents paramètres de l'instruction de mouvement.

Voir p. 5.21 du « Manuel de mise en service du FANUC » pour insérer, détruire une ligne de commande.

**Rappel :** *W* pour *yaW* signifie lacet, *P* pour *Pitch* signifie tangage, *R* pour *Roll* signifie roulis.

## 8.4 Quelques instructions de mouvement

Une instruction de mouvement déplace le bras du robot à une position spécifiée, dans les limites de la zone de travail, avec une rapidité et un mode de mouvement prédéfinis. Dans une instruction de mouvement, les items énumérés ci-dessous doivent être spécifiés. La structure d'une instruction de mouvement est indiquée dans la figure qui suit.

1. Structure de mouvement : Spécifie la façon de contrôler le chemin de déplacement jusqu'à une position donnée.
2. Structure des données de position : Registre qui permet d'enregistrer les données de position.
3. Données de position : Spécifie une position vers laquelle le robot doit se déplacer.
4. Vitesse : Spécifie la vitesse du robot.
5. Trajectoire de positionnement : Définit la méthode avec laquelle le robot termine son mouvement.

<div style="text-align: center;"> </div>			
	* Un numéro de position peut être aussi grand que la capacité de mémoire le permet.		
1	Structure du mouvement J (Joint) L (Linéaire) C (Circulaire)	4	Vitesse : 1 à 100 % 1 à 2000 mm/sec 1 à 12000 cm/min 0.1 à 4724.0 inch/min 1 à 272 deg/sec 1 à 3200 deg/sec 1 à 32000 m/sec
2	Structure des données de position : P 1 à 1500* PR 1 à 10	5	Trajectoire de positionnement FINE CNT 0 à 100
3	Données de position : UF : 0 UT : 1 X : 1500.374 W : 10.000 Y : -342.992 P : 20.000 Z : 956.895 R : 40.000 CONF : N, R, D, F, 0, 0, 0		

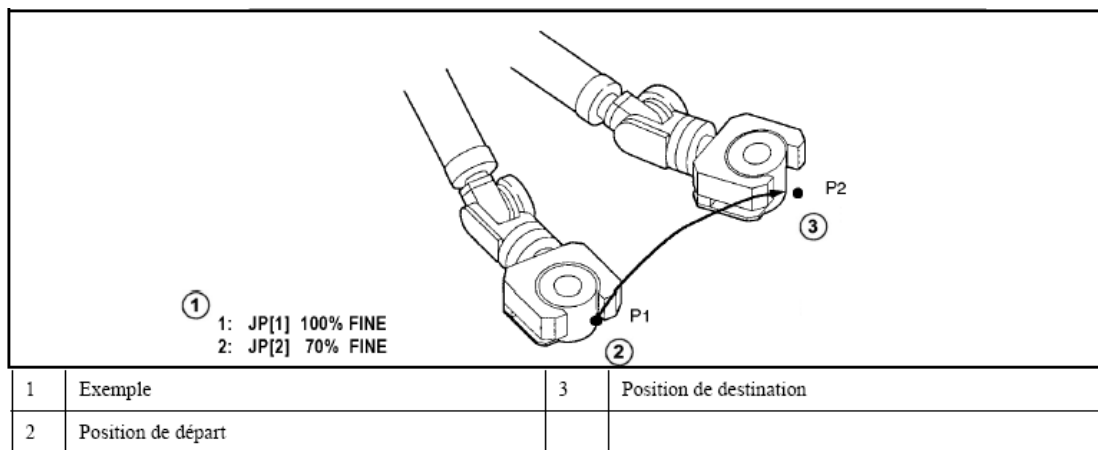
#### 8.4.1 Structure de mouvement ①

Le chemin d'un mouvement jusqu'à une position définie est spécifiée par la structure de mouvement. Trois types de mouvement sont possibles :

##### ➤ Mouvement Joint (J)

Le mode de mouvement *JOINT* est le mode de base pour déplacer le robot dans une position spécifique. Le robot accélère le long, ou autour, de tous les axes, bouge à une vitesse spécifiée, décélère, et s'arrête. La trajectoire de mouvement est, habituellement, non linéaire. Un pourcentage de vitesse maximum est défini comme vitesse de mouvement *JOINT*.



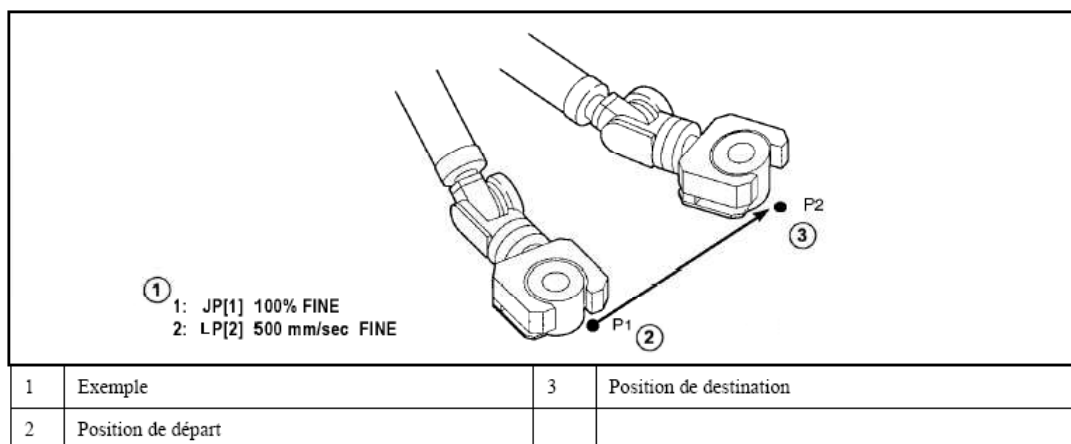


Lorsque le type de mouvement est *JOINT*, la vitesse est spécifiée ainsi :

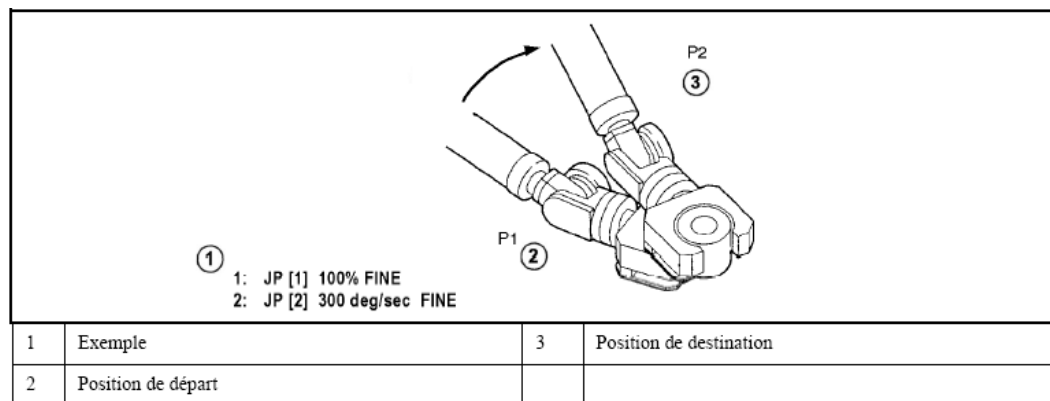
- Un **pourcentage** de 1 à 100 % de la vitesse maximum doit être défini.
- Lorsque l'unité de mesure est la seconde, spécifier une valeur de 0,1 à 3200 *sec*, représentant le temps pris pour un mouvement.
- Lorsque l'unité de mesure est la milliseconde, spécifier une valeur de 1 à 32000 *msec*, représentant le temps pris pour un mouvement.

#### ➤ **Mouvement linéaire (L)**

Le mode de mouvement *Linéaire* contrôle la trajectoire du centre outil par une droite allant d'un point de départ à un point final. La structure du mouvement est spécifiée de façon à apprendre une position finale. Pour la spécification de vitesse linéaire, l'option désirée doit être choisie en **mm/sec**, **cm/min** ou **inch/min**.

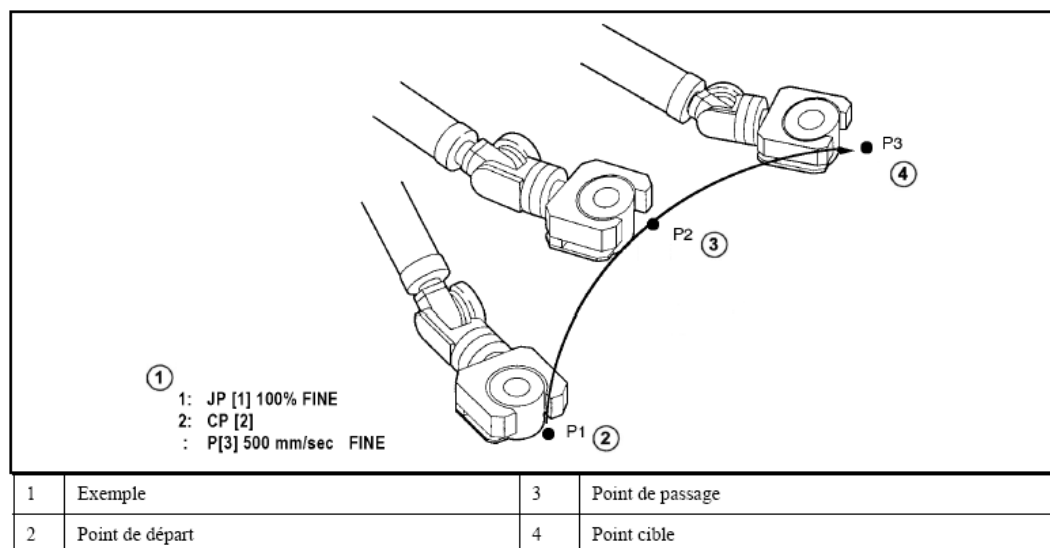


L'opération de rotation est un procédé de mouvement par lequel l'outil est tourné autour de la pointe du centre outil, depuis la position de départ à la position cible, en utilisant une opération linéaire. La vitesse est spécifiée en **deg/sec**. Le point est commandé **linéairement**.



### ➤ Mouvement circulaire (C)

Le mode de mouvement *Circulaire* contrôle la trajectoire du centre outil depuis le point de départ jusqu'au dernier point en passant par un point de passage intermédiaire. Le point de passage et le point cible sont indiqués dans une instruction. Pour la spécification de la vitesse circulaire, l'option désirée doit être choisie en **mm/sec**, **cm/min** ou **inch/min**.



### 8.4.2 Structure des données ② et ③ de position

L'information de position décrit l'emplacement du centre outil lorsqu'une instruction de mouvement est ajoutée à un programme. L'information sur la position est effectuée pour sept composants, comme indiqué ci-dessous.

$$P[n] = (X, Y, Z, W, P, R, \text{config})$$

Les composants d'emplacement ou de localisation (**X**, **Y**, **Z**) décrivent l'emplacement tridimensionnel de la position. Les composants d'orientation (**W**, **P**, **R**) décrivent la rotation autour de X, Y et de Z. Le composant de configuration (**config**) décrit l'état des axes lorsque le robot arrive à la position de destination. L'orientation des axes du poignet à la position de destination reste la même mais l'orientation des autres axes peut changer.

Il existe deux types d'enregistrement de position :

Commande de position P[n]	Registre de position PR[x]
---------------------------	----------------------------

**n** est le numéro de position. Une commande de position stocke l'information sur la position actuelle du bras du robot.

**x** est le numéro du registre de position. Un registre de position stocke l'information sur la position dans un emplacement de stockage séparé du programme.

### 8.4.3 Trajectoire de positionnement ⑤

La trajectoire de positionnement définit la méthode avec laquelle le robot termine son mouvement dans une instruction. Deux sortes de positionnement de trajectoire sont possibles : *FINE* et *CNT*.

**Positionnement de trajectoire FINE**      **JP[i] 50% FINE** : Lorsque le positionnement de trajectoire FINE est spécifié, le robot s'arrête à un point de destination avant de repartir jusqu'au point suivant.

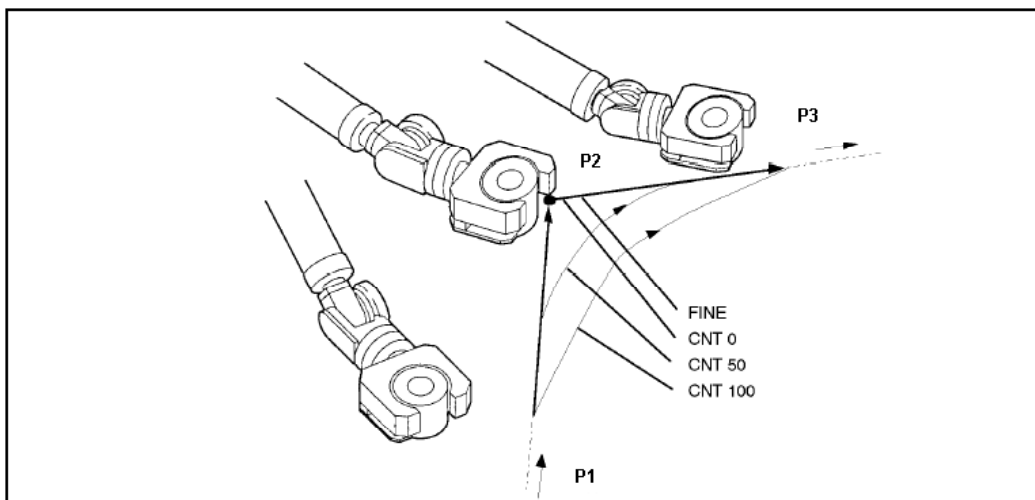
**Positionnement de trajectoire CNT**      **JP[i] 50% CNT50** : Lorsque le positionnement de trajectoire CNT est spécifié, le robot s'approche du point de destination mais ne s'arrête pas et continue jusqu'au point suivant.

La proximité à laquelle le robot doit s'approcher du point de destination peut être définie en spécifiant une valeur de 0 à 100.

Sur CNT0, le robot est au plus près avec une décélération maximum.

Sur CNT100, le robot est au plus loin avec une décélération minimum.

Lorsqu'une instruction comme une **instruction d'attente** est apprise, le robot s'arrête **au point cible** pour exécuter cette instruction.





1	Point de départ P1	3	Point suivant P3
2	Point cible P2		



➤ **Exemple** : Déplacement du robot ARC Mate sur une position souhaitée



Position : X = 210.574      Y = 712.501      Z = 357.396  
W = 103.041      P = -60.693      R = 124.391  
J1 = 87.957      J2 = -23.175      J3 = -24.453  
J4 = -72.955      J5 = 44.234      J6 = 89.797


NB : *yaW* correspond à l'angle de lacet, *Pitch* à l'angle de tangage et *Roll* à l'angle de roulis.

Appuyer sur la touche  - **POINT**.

Se déplacer dans le menu avec les touches  , appuyer sur la touche  pour sélectionner la commande voulue. Pour réaliser un mouvement simple, on choisit : **JP [..] 100% FINE**





Pour modifier les différentes caractéristiques, se déplacer sur la ligne de commande à l'aide des touches  .


Sélectionner **P [..]** et appuyer sur la touche  - **POSITION** pour entrer les coordonnées de la position désirée (appuyer sur la touche  - **REPRE** pour sélectionner le repère souhaité, soit le repère **USER** (X, Y, Z, W, P, R), soit le repère **JOINT** (J1, ..., J6)).

Appuyer sur la touche  - **DONE** pour enregistrer les coordonnées entrées.





NB : Voir le manuel (section 5-18) en ce qui concerne les modifications d'un programme (insérer, effacer, ...).

## 8.5 Entrées/Sorties





Appuyer sur la touche  pour accéder au menu des entrées/sorties et sélectionner **I/O** à l'aide des touches  , valider votre choix en appuyant sur la touche .

Appuyer sur la touche  - **TYPE** pour accéder aux différentes entrées/sorties.

**Exemple :** Accéder aux E/S du robot

Appuyer sur la touche  - **TYPE**, choisir **ROBOT** à l'aide des touches   et . Sur cette page, sont listées toutes les E/S automates du robot dans des registres de type **RO [..] = ...**, notamment l'ouverture et la fermeture de la pince.

**Pour insérer une E/S dans un programme :**

Lors de l'édition d'un programme (pour cela appuyer sur la touche **EDIT**), appuyer sur la touche  - **INST** (appuyer, si nécessaire, sur la touche **NEXT** pour accéder à **INST**), choisir **I/O** à l'aide des touches   et presser .

**Exemple :** Ouverture et fermeture de la pince

Choisir les registres **RO [..]=...**, entrer le numéro du registre (RO [1]) pour la pince.

Ouverture de la pince → RO [1 : FERMETURE PINCE] = OFF

Fermeture de la pince → RO [1 : FERMETURE PINCE] = ON

## 8.6 Structures algorithmiques de base







Instruction de branchement inconditionnel : **JMP LBL[x]**

Instruction de branchement conditionnel : **IF RO [1 : FERMETURE PINCE] = OFF ...**

Instructions WAIT : **WAIT ... (sec)**

**WAIT RO [1 : FERMETURE PINCE] = OFF**

## 8.7 Sous-programme

Pour faire appel au sous-programme **program**, insérer l'instruction **CALL program** en appuyant sur la touche  - **INST**, ensuite à l'aide des flèches  , choisir **CALL**. Sélectionner le sous-programme dans la liste donnée à l'aide des flèches   puis appuyer sur .

## 8.8 Exemple de programme

Déplacement de tubes de stylo

```
RO [1 : FERMETURE PINCE] = OFF
JPR [6 : depart] 100% FINE
JPR [7 : POS STYLO] 100% FINE
RO [1 : FERMETURE PINCE] = ON
JPR [8 : SOULEVE] 100% FINE
JPR [9 : TOURME] 100% FINE
JPR [10 : DEPLACE 1] 100% FINE
JPR [11 : DPELACE 2] 100% FINE
RO [1 : FERMETURE PINCE] = OFF
JPR [12] 100% FINE
JPR [13] 100% FINE
JPR [14] 100% FINE
RO [1 : FERMETURE PINCE] = ON
JPR [15] 100% FINE
JPR [16] 100% FINE
JPR [17] 100% FINE
JPR [18] 100% FINE
RO [1 : FERMETURE PINCE] = OFF
JPR [19] 100% FINE
CALL TP
END
```