

**CONTRIBUTION A LA COMMANDE  
DES ROBOTS RAPIDES ET PRECIS  
APPLICATION AU ROBOT  
DELTA A ENTRAINEMENT DIRECT**

**THESE No 922 ( 1991 )**

**PRESENTEE AU DEPARTEMENT DE MICROMECHANIQUE**

**ECOLE POLYTECHNIQUE FEDERALE DE LAUSANNE**

**POUR L'OBTENTION DU GRADE DE DOCTEUR ES SCIENCES TECHNIQUES**

**PAR**

**ALAIN CODOUREY**

**Ingénieur en microtechnique diplômé EPFL  
originaire de Lovens et Cottens (FR)**

**acceptée sur proposition du jury :**

**Prof. C.W. Burckhardt, rapporteur  
Prof. R. Clavel, corapporteur  
Prof. A. Glattfelder, corapporteur  
Prof. R. Longchamp, corapporteur**

**Lausanne, EPFL  
1991**



A mes parents,  
à Patricia.



## Avant-propos

Ces quatre années passées au sein de l'institut de Microtechnique de L'Ecole Polytechnique Fédérale de Lausanne ont été pour moi une source d'enrichissement intellectuel et une ouverture d'esprit que je dois sans conteste à Monsieur le Professeur C.W. Burckhardt. Ses nombreux contacts industriels et son dynamisme ont contribué à la réussite de ce projet. Pour cela, je tiens à le remercier vivement.

Je désire exprimer toute ma reconnaissance à Monsieur le Professeur R. Clavel pour la confiance dont il m'a fait preuve en me laissant "l'éducation de son enfant". Ses conseils judicieux, sa disponibilité et son optimisme ont toujours su redonner les impulsions nécessaires lors des moments de doute. Sans lui, ce travail n'aurait jamais pu avoir lieu.

La maison SULZER Gebrüder AG à Winterthur mérite toute ma gratitude pour son soutien financier et pour l'aide technique apportée lors du développement du prototype du robot, notamment pour la motorisation. Je tiens plus particulièrement à remercier Monsieur L. Guzzella pour son intérêt particulier suscitant ma motivation et Monsieur le Professeur A. Glattfelder d'avoir accepté d'être membre du jury. Sa lecture attentive du rapport et ses conseils bien à propos ont contribué à la clarté du texte.

Je remercie aussi Monsieur le Professeur R. Longchamp, de l'institut d'Automatique, pour avoir encouragé les contacts avec les membres de son laboratoire et pour le soin apporté à l'examen de ma thèse.

Mes plus vifs remerciements vont à tous mes collègues de l'institut de Microtechnique et plus particulièrement à Messieurs Bays, Crausaz, Fournier, Maeder, Perrenoud, Rey et Thielen qui, à titres divers, ont contribué à la réalisation du prototype et de la commande, ainsi qu'à la correction du rapport. Plusieurs étudiants, par leurs projets, ont également été impliqués dans ce travail. Je tiens à signaler plus particulièrement les contributions de Baumann, Dayer, Fasnacht et Mansutti.

Je remercie également chaleureusement mes parents qui m'ont encouragé tout au long de mes études et sans qui je n'aurais jamais pu entreprendre ce travail. Enfin, ma chère Patricia mérite une immense gratitude car sans son soutien et ses encouragements ce travail n'aurait sans doute pas été terminé.

## Résumé

Ce travail de thèse se situe dans le domaine de la commande des robots rapides et précis. Ces deux dernières notions, de nature opposée, sont généralement difficile à concilier, et de ce fait, les robots actuellement sur le marché sont encore lents. Pour y parvenir, il est nécessaire de réunir trois conditions essentielles :

- avoir un robot dont la structure mobile est la plus légère possible
- utiliser un système d' entraînement rigide et sans jeu
- utiliser une commande performante tenant compte de la dynamique rapide du système.

Les robots à structure série habituellement rencontrés dans l'industrie ont le désavantage de devoir supporter le poids des moteurs distribués sur chaque articulation. Les robots parallèles permettent d'éliminer ce problème, mais ceci au dépend de leur volume de travail. Des récents développements à l'institut de Microtechnique de l'EPFL ont permis de réaliser une structure parallèle à 4 degrés de liberté appelée DELTA, ayant un volume de travail suffisant pour beaucoup d'applications industrielles. La légèreté de cette structure offre un potentiel de rapidité important.

Pour avoir un robot précis, il est indispensable d'utiliser un système d' entraînement sans jeu. Pour avoir un robot rapide, il est nécessaire de réduire au mieux son élasticité. Dans cette optique, l'élimination du réducteur, généralement source de bien des ennuis, est séduisante. C'est dans cet esprit qu'a été réalisé, dans ce travail, le robot DELTA à entraînement direct. Malheureusement, l'élimination de l'étage de réduction rend le moteur directement sensible aux variations d'inertie de la structure et aux couplages entre les axes. Le contrôle du robot devient plus délicat.

Ce rapport présente plusieurs aspects de la commande des robots rapides, avec application au robot DELTA à entraînement direct. Tout d'abord, la problématique de l' entraînement direct est présentée, montrant l'intérêt de l'incorporation du modèle dynamique du robot dans l'algorithme de commande. Un modèle dynamique simplifié

du robot DELTA est alors établi en tenant compte des contraintes temps réel. Son calcul sur Transputer T800 prend moins de 750 [ $\mu$ s].

Un régulateur utilisant ce modèle comme commande a priori est proposé, ce qui permet d'améliorer notablement les performances du robot. Pour effectuer des mouvements rapides, des trajectoires adéquates, compatibles avec la dynamique du système, ont été étudiées pour le cas particulier de la prise et dépose d'objets.

La juxtaposition de ces différents éléments a permis d'obtenir des cadences de travail supérieures à 3 [Hz] pour une précision de 0.5 [mm] et de près de 2 [Hz] pour une précision de 0.03 [mm]. Ainsi, à notre connaissance, le robot DELTA à entraînement direct est actuellement le robot le plus rapide au monde.

## Principales notations et définitions

La terminologie utilisée dans le texte est généralement issue de la littérature française dans le domaine de la robotique. Quelques définitions sont données ci-dessous afin de clarifier la lecture.

*Espace articulaire* : C'est l'espace des configurations articulaires d'un robot représentant la situation des différents corps entre eux. La solution la plus classique pour décrire cet état consiste à utiliser les variables de chacune des articulations.

*Espace opérationnel* : C'est l'espace dans lequel est représentée la situation de l'organe terminal. La solution la plus simple consiste à utiliser les coordonnées cartésiennes.

*Géométrie* : Etude de la position d'un corps solide sans considération de son évolution temporelle.

*Cinématique* : Etude du mouvement d'un corps solide sans considération de force.

*Dynamique* : Etude des forces généralisées agissant sur un corps solide en mouvement.

Les notations et symboles usuels de la littérature spécialisée dans le domaine de la commande des robots sont en général utilisés et sont définis la première fois qu'ils apparaissent dans le texte. Les vecteurs et matrices sont représentés en caractère gras. Les symboles principaux sont résumés ci-dessous :

- n : nombre de degrés de liberté du robot qui représente également la dimension de l'espace articulaire.
- m : dimension de l'espace opérationnel ( $m \leq 6$  et  $m \leq n$ ).
- $\mathbf{X}, \dot{\mathbf{X}}, \ddot{\mathbf{X}}$  : vecteurs (dimension m) des positions, vitesses et accélérations généralisées (rotation ou translation) dans l'espace opérationnel.

<b>q, <math>\dot{q}</math>, <math>\ddot{q}</math></b>	: vecteurs (n) des positions, vitesses et accélérations généralisées dans l'espace articulaire.
<b><math>\alpha, \dot{\alpha}, \ddot{\alpha}</math></b>	: cas particulier des coordonnées articulaires en rotation (cas du robot DELTA).
<b><math>\Gamma</math></b>	: vecteur (n) des forces généralisées (force ou couple) dans l'espace articulaire.
<b>M</b>	: cas particulier des couples articulaires (cas du robot DELTA).
<b>A</b>	: matrice (n x n) d'inertie du robot.
<b>H</b>	: vecteur (n) représentant l'effet des forces centrifuges, de Coriolis et de gravité.

## **Table des matières**

<b>Avant-propos .....</b>	<b>iii</b>
<b>Résumé.....</b>	<b>v</b>
<b>Principales notations et définitions.....</b>	<b>vii</b>
<b>Table des matières.....</b>	<b>ix</b>
<b>1. Introduction .....</b>	<b>1</b>
1.1. Généralités.....	1
1.2. Motivations et objectifs du travail.....	2
1.3. Contributions.....	3
1.4. Organisation du rapport.....	5
<b>2. L'entraînement direct et le robot DELTA.....</b>	<b>7</b>
2.1. Introduction .....	7
2.2. Etat de l'art de la robotique à entraînement direct .....	8
2.3. Motorisation.....	14
2.3.1. Introduction .....	14
2.3.2. Moteurs couples à courant continu .....	14
2.3.3. Moteurs couples sans balais.....	15
2.3.4. Moteurs couples à réluctance variable.....	16
2.4. Conséquences pour la conception mécanique du robot .....	16
2.4.1. Introduction .....	16
2.4.2. Problème de la gravité .....	17
2.4.3. Redistribution des moteurs.....	17
2.4.4. Capteurs de position angulaire.....	18
2.5. Conséquences pour la conception de la commande du robot.....	18
2.5.1. Introduction .....	18
2.5.2. Frottement.....	18
2.5.3. Rigidité de positionnement.....	19
2.5.4. Effet de l'inductance du moteur.....	19
2.5.5. Effet de la dynamique du bras .....	20

2.6. Le robot DELTA à entraînement direct.....	21
2.6.1. Le robot DELTA.....	22
2.6.2. DELTA à entraînement direct .....	23
2.6.3. Cahier des charges .....	25
2.7. Conclusion .....	26
<b>3. Modèle dynamique .....</b>	<b>27</b>
3.1. Introduction .....	27
3.2. Méthode de Newton-Euler.....	29
3.2.1. Introduction .....	29
3.2.2. Equations de Newton et d'Euler.....	29
3.2.3. Utilisation pour le calcul du modèle dynamique inverse d'un robot .....	31
3.3. Modèle dynamique temps réel pour le robot DELTA.....	33
3.3.1. Paramétrisation du système et référentiels.....	33
3.3.2. Modèle "complet" .....	35
3.3.3. Hypothèses simplificatrices.....	36
3.3.4. Modèle simplifié.....	36
3.3.5. Estimation du temps de calcul.....	40
3.4. Modèle dynamique "découplé" pour le robot DELTA.....	41
3.4.1. Etablissement du modèle découpé .....	41
3.4.2. Etude de la matrice d'inertie.....	43
3.4.3. Influence des termes centrifuges, de Coriolis et de gravité .....	45
3.5. Validation et limites du modèle dynamique.....	47
3.5.1. Comparaison avec le modèle de [Clavel 91] .....	47
3.5.2. Essai en boucle ouverte.....	48
3.5.3. Limites du modèle.....	49
3.6. Conclusion .....	50
<b>4. Structure de la commande de robot.....</b>	<b>51</b>
4.1. Introduction .....	51
4.2. Structure interne du contrôleur de robot .....	53
4.2.1. Génération des consignes dans l'espace articulaire.....	54
4.2.2. Génération des consignes dans l'espace opérationnel.....	55
4.3. Cas du robot DELTA.....	56
4.4. Conclusion .....	56

<b>5. Génération des consignes .....</b>	<b>57</b>
5.1. Introduction .....	57
5.2. Notions de trajectoire et de mouvement pour robots rapides .....	58
5.2.1. Paramétrisation de la trajectoire.....	58
5.2.2. Contraintes .....	59
5.2.3. Prise en compte des contraintes "dynamiques" .....	59
5.2.4. Recherche d'un optimum.....	60
5.3. Lois de mouvement.....	61
5.3.1. Préambule .....	61
5.3.2. Loi parabolique ou loi "bang-bang" .....	62
5.3.3. Loi cycloïde .....	64
5.3.4. Autres lois.....	65
5.3.5. Calcul du temps de parcours .....	66
5.4. Trajectoires .....	69
5.4.1. Trajectoire "naturelle" du robot DELTA .....	69
5.4.2. Ligne droite .....	71
5.4.3. Demi-ellipse.....	72
5.4.4. Clothoïde .....	74
5.4.5. Comparaison des différentes trajectoires .....	77
5.5. Conclusion .....	82
<b>6. Asservissement.....</b>	<b>85</b>
6.1. Introduction .....	85
6.2. Tour d'horizon des méthodes d'asservissement des robots .....	85
6.2.1. Commande classique.....	86
6.2.2. Méthodes basées sur le modèle dynamique du robot.....	87
6.2.3. Méthodes adaptatives .....	92
6.2.4. Commande par mode de glissement.....	94
6.2.5. Commande "neuronale" .....	95
6.3. Application aux robots rapides et précis .....	95
6.3.1. Introduction .....	95
6.3.2. Modélisation du moteur et de l'amplificateur.....	97
6.3.3. Régulateur de trajectoire .....	99
6.3.4. Régulateur de positionnement.....	100
6.3.5. Stratégie de commutation.....	102
6.3.6. Choix des gains du régulateur .....	102

6.3.7. Période d'échantillonnage .....	102
6.3.8. Prise en compte de la charge transportée .....	102
6.4. Conclusion .....	103
<b>7. Résultats expérimentaux avec le robot DELTA .....</b>	<b>105</b>
7.1. Introduction .....	105
7.2. Performances du régulateur .....	105
7.3. Effet de la trajectoire et de la loi de mouvement .....	111
7.3.1. Influence de la loi de mouvement.....	111
7.3.2. Influence de la trajectoire .....	112
7.4. Cadences.....	114
7.4.1. Trajectoire de prise et dépose .....	114
7.4.2. Trajectoire verticale.....	115
7.5. Conclusion .....	115
<b>8. Architecture matérielle.....</b>	<b>117</b>
8.1. Introduction .....	117
8.2. Architectures classiques.....	120
8.3. La solution à transputers.....	122
8.3.1. Implémentation matérielle de la commande.....	122
8.3.2. Implémentation logicielle .....	125
8.4. Evaluation des performances.....	126
8.5. Conclusion .....	127
<b>9. Conclusion.....</b>	<b>129</b>
9.1. Résumé et contributions .....	129
9.2. Perspectives d'avenir.....	131
<b>A. Transformation de coordonnées du robot DELTA pour le calcul en temps réel .....</b>	<b>135</b>
A.1. Introduction.....	135
A.2. Singularités mathématiques des modèles géométriques inverse du robot DELTA .....	136
A.2.1. Modèle de Sternheim .....	136
A.2.2. Modèle de Clavel.....	136
A.3. Nouveau modèle éliminant les singularités mathématiques du modèle inverse.....	138

A.4. Modèle géométrique direct .....	140
A.5. Evaluation des temps de calcul .....	142
A.6. Conclusion .....	142
<b>B. Equations du modèle dynamique .....</b>	<b>143</b>
B.1. Modèle dynamique simplifié du robot DELTA .....	143
B.2. Modèle dynamique "découplé" pour le robot DELTA .....	149
B.3. Représentation de la matrice d'inertie du robot DELTA .....	151
<b>C. Equations de la clohoïde .....</b>	<b>155</b>
C.1. Introduction.....	155
C.2. Clohoïde au sommet .....	157
C.3. Cas particulier de la clohoïde au sommet symétrique .....	159
<b>D. Paramètres du robot prototype.....</b>	<b>161</b>
D.1. Paramètres géométriques.....	161
D.2. Paramètres dynamiques .....	161
<b>E. Vue d'ensemble du prototype et de sa commande.....</b>	<b>163</b>
<b>Bibliographie .....</b>	<b>165</b>
<b>Curriculum vitae.....</b>	<b>171</b>



## 1. Introduction

### 1.1. Généralités

Si le concept du robot date de plusieurs siècles, le mot "robot" n'est apparu que très récemment. Il semble avoir été popularisé dès 1920 par la pièce de théâtre de Karel Capek intitulée R.U.R. (Rossum's Universal Robot). Depuis, de nombreux auteurs de science-fiction n'ont pas manqué d'imagination pour façonner le robot du futur. C'est d'ailleurs ainsi que le perçoit le grand public. Le robot est une machine anthropoïde dotée d'une mobilité et d'une autonomie supérieures à celles de l'homme, tout à fait consciente de son environnement et d'une intelligence logique hors du commun.

Même si les recherches entreprises dans le domaine de la robotique mobile ont beaucoup progressé ces dernières années, un robot doté de telles propriétés n'est actuellement pas envisageable, même à très long terme. Les principales difficultés rencontrées se situent dans les domaines de la perception de l'environnement et de la prise de décision (intelligence artificielle), deux domaines de recherche à part entière.

En milieu industriel, où l'environnement de travail est connu a priori et non-autoévolutif, ces deux problèmes sont écartés. Il n'y a donc rien d'étonnant si la robotique industrielle, augmentant la productivité et la flexibilité de l'entreprise, a eu un essor important jusqu'à aujourd'hui.

Le premier robot industriel connu est né en 1962 chez Unimation (U.S.A.). Il était composé de 5 axes hydrauliques et capable de transporter des charges de quelques kilos le long d'un trajet défini préalablement. Bien que peu performante, cette réalisation ouvrait alors un large domaine de recherche. Au fil des années, avec les progrès impressionnantes de l'électronique et de la microinformatique, les robots sont devenus plus précis et plus simples à utiliser. Ils font maintenant partie de l'outillage classique de la production industrielle.

## 1.2. Motivations et objectifs du travail

Malgré leurs nombreuses qualités, la plupart des robots actuels souffrent d'un problème majeur; ils sont lents et donc inadaptés au transfert de pièces légères à cadence élevée. En milieu industriel, les cadences typiques pour des tâches de palettisation sont actuellement inférieures à 1 [Hz] et la réalisation d'un robot rapide et précis apparaît comme un défi. Dans divers laboratoires au monde, des recherches sont en cours pour améliorer les cadences. Ce sont surtout les capacités d'accélération qui caractérisent les performances d'un robot rapide. Le robot joueur de ping-pong IfM [Faessler 88b] semble détenir le record avec des accélérations jusqu'à 240 [m/s<sup>2</sup>]. Un robot destiné à la découpe de filets de poisson par jet d'eau permet des accélérations de 160 [m/s<sup>2</sup>] [Dalum & Lund 90]. [Asada & al 84b] décrivent un robot caractérisé par un entraînement direct et un mécanisme de transmission parallèle qui atteint 52 [m/s<sup>2</sup>]. Un autre robot à entraînement direct, mais de type SCARA, développé par Mitsubishi au Japon [Takeshita 85] fournit des accélérations de 40 [m/s<sup>2</sup>].

Si l'on se réfère à la célèbre équation de Newton,

$$\mathbf{F} = m \mathbf{a}$$

on remarque que, à force égale, l'accélération augmente si la masse en mouvement diminue. Ainsi, plus la structure mécanique du robot sera légère, plus il pourra être rapide.

Les moteurs, généralement lourds, sont une part non négligeable de la masse totale d'un robot à structure série habituellement rencontré sur le marché. Trouver une structure mécanique telle que les moteurs restent immobiles, amène presque exclusivement au concept du robot parallèle. Généralement, on lui reproche d'avoir un volume de travail trop restreint, mais les travaux menés par R. Clavel à l'institut de Microtechnique (IMT) de l'EPFL ont permis de résoudre le problème grâce au robot parallèle DELTA [Clavel 88]. L'horizon ainsi ouvert pour la manutention de pièces légères à cadence élevée est important.

En robotique, on connaît depuis longtemps les problèmes engendrés par les réducteurs (jeu, frottement, élasticité, ...). La réalisation d'un robot précis implique nécessairement l'utilisation d'un système de transmission sans jeu. Dans ce but, l'idée

d'accoupler le bras du robot directement sur le rotor du moteur est séduisante. Des récents travaux dans le domaine des robots à entraînement direct [Asada & Youcef-Toumi 87] ont montré la faisabilité d'une telle approche. L'élimination de l'élasticité de l'entraînement nous laisse d'ailleurs espérer des performances dynamiques accrues.

Malheureusement, l'élimination du réducteur rend les phénomènes non-linéaires et les couplages de la structure mécanique beaucoup plus apparents au niveau du moteur. Ceci complique le réglage du robot qui doit alors tenir compte du modèle dynamique de la structure mobile. Pour obtenir une commande performante, il est nécessaire qu'il puisse être calculé en temps réel. De nombreux travaux ont permis d'optimiser ce calcul pour les robots de type série. Pour les robots parallèles, en raison de l'interdépendance des variables articulaires, ce calcul est plus compliqué et les modèles développés se limitent à quelques cas particuliers. Pour le robot DELTA, un modèle dynamique basé sur la méthode des travaux virtuels est présenté par [Clavel 91] et utilisé intensivement en simulation. Il ne satisfait cependant pas les contraintes temps réel et un nouveau modèle, adapté aux exigences de la commande, devra être développé.

Un système d'asservissement performant ne suffit pas pour obtenir un robot rapide. Il est encore nécessaire de générer des trajectoires appropriées. La structure du robot DELTA est particulièrement adaptée aux tâches de palettisation. Pour cette raison, ce travail se limitera à l'étude et à l'optimisation de mouvements de "prise et dépose", ce qui représente tout de même la plus grande partie des applications industrielles des robots.

L'application du concept "entraînement direct" au robot DELTA est intéressante dans l'optique de la réalisation "du robot le plus rapide au monde". Le but principal de ce travail est de maîtriser sa commande.

### 1.3. Contributions

Les développements effectués dans le cadre de ce travail ont permis la réalisation d'un robot très performant pour des tâches de palettisation de pièces légères.

Les contributions essentielles et les aspects originaux de cette recherche peuvent être résumés en cinq points :

*1) Mise en oeuvre d'un entraînement direct sur le robot parallèle DELTA.*

On obtient ainsi une transmission rigide et sans jeu, ce qui permet simultanément d'augmenter les cadences et d'avoir une meilleure précision.

*2) Etablissement d'un modèle dynamique simplifié du robot parallèle DELTA permettant son calcul en temps réel.*

Le modèle établi est basé sur la méthode de Newton-Euler. Des hypothèses simplificatrices judicieusement posées permettent de réduire grandement la complexité du modèle tout en conservant une bonne précision. Le temps de calcul obtenu est de 750 [ $\mu$ s]. Afin d'isoler la matrice d'inertie du robot, une méthode originale basée sur la dérivée numérique de la jacobienne est proposée.

*3) Implémentation d'un régulateur utilisant ce modèle dynamique comme commande a priori.*

Le calcul du modèle dynamique en temps réel permet son utilisation dans la boucle de réglage. Dans notre cas, le modèle est utilisé comme commande a priori, ce qui permet de diminuer les erreurs de poursuite de la trajectoire d'un facteur 6 environ par rapport à un régulateur PD classique.

Afin de garantir une erreur nulle au point d'arrivée, un régulateur de positionnement PID est utilisé dans cette deuxième phase. La commutation de l'un à l'autre des régulateurs est basée sur un heuristique. L'originalité de ce "régulateur en deux phases" réside dans la possibilité de choisir des régulateurs différents, adaptés à la tâche qu'ils doivent accomplir (poursuite d'une consigne ou rejet des perturbations).

*4) Etude et choix de trajectoires compatibles avec la dynamique du robot.*

Nous introduisons ici deux critères liés à la dynamique pour le choix des trajectoires : le couple maximal admissible et sa variation maximale liée au temps de montée du courant dans le moteur. Ce deuxième critère nous impose de choisir des trajectoires sans discontinuité d'accélération tangentielle (loi de mouvement) ni d'accélération transverse (courbure de la trajectoire). La clochoïde, courbe bien connue dans le domaine du génie civil, trouve ici une application originale en robotique.

### 5) Réalisation d'une architecture de commande parallèle multiprocesseurs sur Transputers.

Pour permettre le calcul des trajectoires, du modèle dynamique et de la régulation en temps réel, une architecture multiprocesseurs basée sur un réseau de trois transputers connectés en "pipe-line" a été développée. Ceci a permis de choisir une période d'échantillonnage de 1 [ms], nécessaire pour la commande d'un robot rapide.

Tout au long de ce projet, un accent particulier a été placé sur la réalisation pratique; ceci dans l'optique d'une implantation industrielle à court terme. L'originalité de ce travail réside surtout dans l'approche globale du problème, qui a permis la réalisation de l'un des robots les plus rapide du moment.

## 1.4. Organisation du rapport

Ce rapport de thèse présente les différents développements et les idées émises dans le but de réaliser un robot rapide et précis. Le chapitre 2 décrit, après un bref état de l'art de la robotique à entraînement direct, les avantages et les contraintes liés à ce nouveau concept. Le prototype du robot DELTA à entraînement direct y est ensuite présenté ainsi qu'un bref cahier des charges des performances visées. Le chapitre 3 présente le modèle dynamique simplifié du robot DELTA qui pourra, par la suite, être utilisé en temps réel dans l'algorithme de commande. Il se distingue des modèles traditionnels par une formulation dépendant non seulement des variables articulaires, mais aussi du mouvement de l'organe terminal. Cette formulation ne permet malheureusement pas d'isoler la matrice d'inertie du robot en vue de l'implantation du réglage par découplage non-linéaire [Dombre & Khalil 88]. Pour cette raison, le modèle développé est étendu en utilisant un calcul numérique de la matrice jacobienne du robot. Finalement, une expérimentation sur le système réel permet de valider le modèle.

Le chapitre 4, décrit très brièvement la structure interne d'un contrôleur de robot et les tâches qu'il doit réaliser. Le cas du robot DELTA y est étudié afin de choisir la structure la mieux adaptée. Le chapitre 5 traite des trajectoires et lois de mouvement pour robots rapides. Il se limite à l'étude de trajectoires de prise et dépose. Pour tenir compte de la constante de temps électrique élevée des moteurs pour l'entraînement direct, un critère basé sur la variation de couple est introduit pour la sélection des

trajectoires et des lois de mouvement. Ces dernières doivent alors être très douces et éviter les discontinuités d'accélération tangentielle et transverse.

Après un tour d'horizon des méthodes de réglage proposées dans la littérature, le chapitre 6 décrit le régulateur choisi pour le robot DELTA à entraînement direct. Il se décompose en un régulateur de trajectoire pour la phase de mouvement et en un régulateur de positionnement pour la phase finale d'arrivée sur le point. Le régulateur de trajectoire est basé sur l'utilisation du modèle dynamique comme commande a priori. Les résultats obtenus avec ce régulateur sont comparés à ceux d'un régulateur PD classique dans le chapitre 7. Les résultats obtenus suivant le type de trajectoire utilisé sont aussi présentés. Le chapitre 8 présente finalement l'architecture informatique réalisée pour la commande du robot DELTA à entraînement direct avant la conclusion générale de ce travail présentée au chapitre 9.

## 2. L'entraînement direct et le robot DELTA

### 2.1. Introduction

Depuis la mise sur le marché du premier robot Unimation en 1962, la robotique industrielle a beaucoup progressé surtout grâce à l'évolution technologique de la micro-informatique. Si les premiers robots utilisaient presque exclusivement des actionneurs hydrauliques, les moteurs électriques, bon marché, propres, de maintenance facile, se sont imposés pour la conception de la plupart des robots de faible et moyenne puissance.

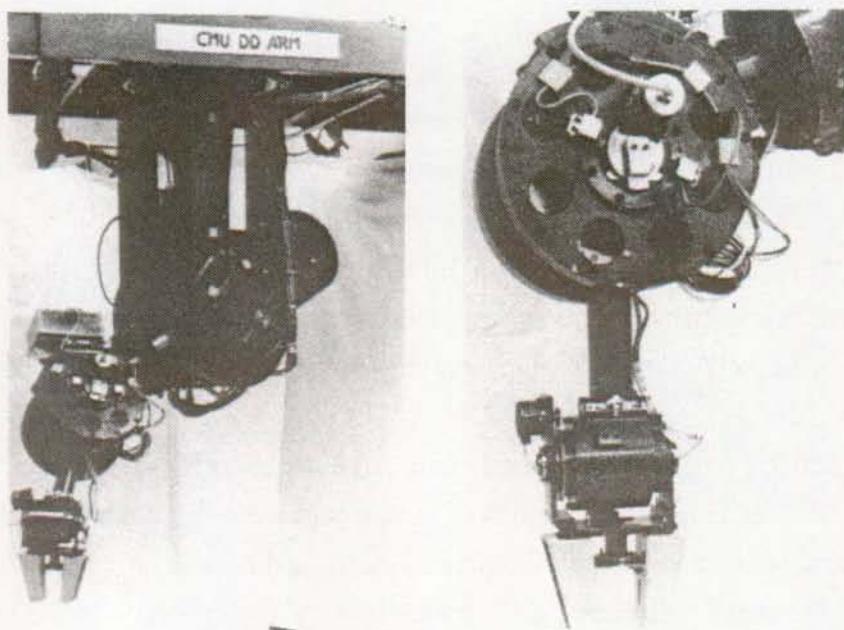
Afin de tirer les meilleures caractéristiques du moteur, on adapte généralement la charge au travers d'un réducteur [Burckhardt & al 89]. Pour obtenir un robot de grande précision, il est particulièrement important que le réducteur n'introduise pas de jeu. Même un jeu très faible au niveau de l'entraînement peut amener à de grandes erreurs en bout de bras. L'élimination du jeu dans les réducteurs se fait souvent en introduisant une précontrainte qui malheureusement augmente le frottement. Ce dernier est difficile à modéliser et détériore en général la précision de positionnement du robot.

D'autre part les réducteurs contribuent essentiellement aux compliances mécaniques. Une faible rigidité ne cause pas seulement des déviations du bras, mais limite également ses performances dynamiques. Ainsi, la diminution de la compliance mécanique du réducteur est une exigence cruciale pour la conception de robots rapides. A l'extrême, l'élasticité peut être éliminée complètement en fixant le bras du robot directement sur le rotor du moteur. C'est l'idée des robots à entraînement direct.

Dans ce chapitre, après un bref état de l'art de la robotique à entraînement direct, nous présenterons plus en détail les nouveaux problèmes posés par ce concept et qu'il faudra résoudre. Ceci amènera finalement à la présentation du robot DELTA à entraînement direct.

## 2.2. Etat de l'art de la robotique à entraînement direct

Le concept du robot à entraînement direct a été établi pour la première fois par H. Asada au début des années 80 [Asada & Kanade 81]. Un premier prototype, "CMU D-D Arm Model I" (photo 2.2.1), est développé à l'institut de robotique de l'université de Carnegie-Mellon (CMU). Il s'agit d'un robot série à 6 articulations rotoides. Les différents bras sont directement solidaires du rotor de chacun des moteurs. De cette manière, les problèmes généralement rencontrés en utilisant des réducteurs sont éliminés. Les actionneurs utilisés sont des moteurs à courant continu à aimant permanent. Le couple maximal qu'ils peuvent fournir est de l'ordre de 150 [Nm] pour un diamètre extérieur du stator de 60 [cm].



*Photo 2.2.1 : CMU Direct Drive Arm Model I  
(tiré de [Asada & Youcef-Toumi 87])*

Ce premier robot, bien que peu performant, démontrait le potentiel de l'approche "entraînement direct" et sa faisabilité. Il montrait surtout que les robots ainsi conçus nécessitent des moteurs de couple élevé, de taille et de poids réduits, sous peine d'une construction lourde et peu élégante.

Des travaux dans cette direction ont été poursuivis par H. Asada et K. Youcef-Toumi au "Massachusetts Institute of Technology" (MIT). Ainsi, en 1983, naissait le

"MIT D-D Arm Model I" (photo 2.2.2), un robot à 3DDL<sup>1</sup> [Asada & al 84a, Asada & Youcef-Toumi 87]. Les moteurs couple sans balais utilisés produisent un couple maximal de 660 [Nm] pour un stator ayant un diamètre de 35 [cm] seulement. La structure mécanique est conçue pour éliminer l'effet de la gravité sur les moteurs afin de ne pas les faire chauffer inutilement lorsque le robot est au repos.

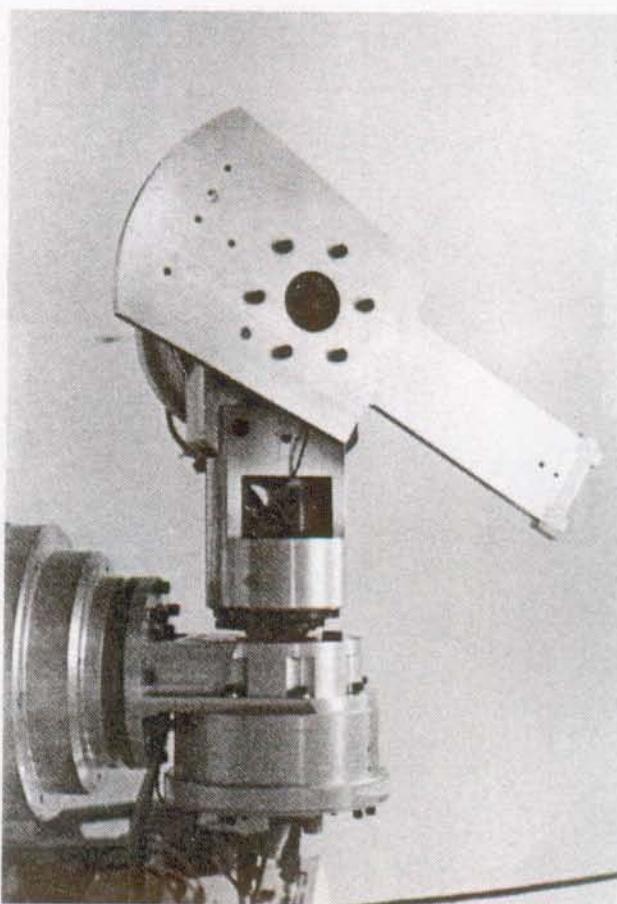


Photo 2.2.2 : MIT D-D Arm Model I (tiré de [Asada & Youcef-Toumi 87])

Une autre variante, "MIT D-D Arm Model II" [Asada & al 84b] est construite par le même groupe. Pour ce bras à 3DDL, ils utilisent une structure mécanique à chaîne cinématique fermée qu'ils nomment "five bar link". Les deux moteurs supérieurs sont situés sur la base et actionnent 2DDL à travers un parallélogramme. Ainsi, les moteurs, de poids élevé, ne bougent pas avec le bras, ce qui augmente sensiblement les performances dynamiques. Construit en aluminium, ce robot a malheureusement une fréquence propre très basse (14 [Hz]). Pour éliminer ce défaut, "MIT D-D Arm

<sup>1</sup> degré de liberté

"Model III" (photo 2.2.3) est construit en matériaux composites. La fréquence propre atteind alors 70 [Hz], ce qui permet des mouvements très rapides. La vitesse maximale est de 12 [m/s] et les accélérations sont de l'ordre de 50 [m/s<sup>2</sup>].

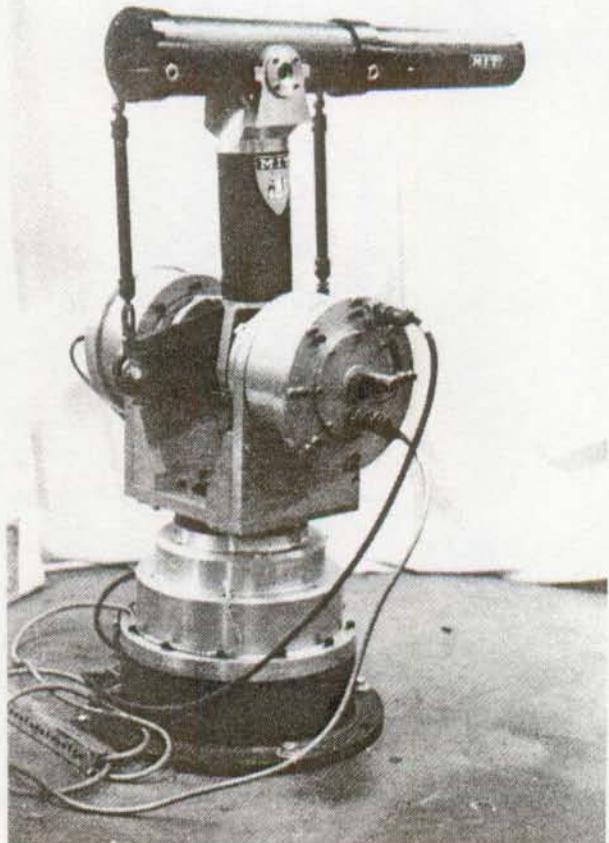
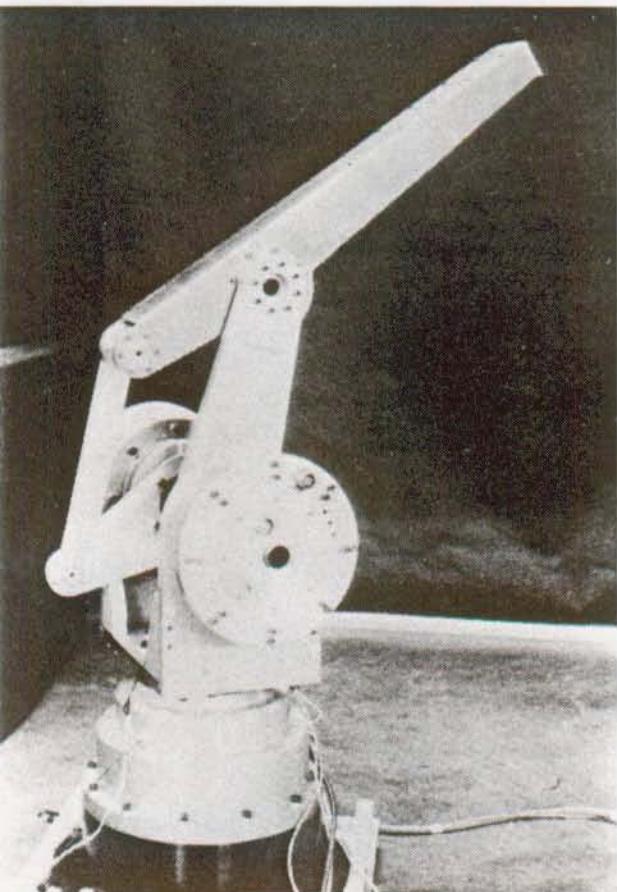


Photo 2.2.3 : MIT D-D Arm Model II & III (tiré de [Asada & Youcef-Toumi 87])

Parallèlement, D. Schmitz, T. Kanade et P. Khosla [Schmitz & al 85] développent le "CMU D-D Arm II" (photo 2.2.4). Afin de réduire au mieux l'effet de la gravité, une configuration de type SCARA est choisie. La simplicité de cette structure est appropriée pour l'entraînement direct. D'ailleurs, le premier robot à entraînement direct commercialisé, le robot AdeptOne est de type SCARA (photo 2.2.5). En fait, il s'agit plutôt d'un entraînement semi-direct dans la mesure où le deuxième moteur est situé sur la colonne principale du robot et transmet son mouvement par l'intermédiaire d'une bande d'acier rigide avec un rapport de réduction unité (1:1). Il est développé par ADEPT Technology Inc., à Sunnyvale, en Californie, en 1983. Les moteurs utilisés

ont été réalisés spécialement par Motornetics Inc, à Santa Rosa en Californie. Avec des vitesses jusqu'à 9 [m/s] et une répétabilité de  $\pm 0.0254$  [mm], ce robot est l'un des plus performants du moment, ce qui explique le succès commercial qu'il a rencontré.

Un autre robot à entraînement direct est commercialisé par Matsushita. Il possède aussi une structure SCARA, mais avec un système de parallélogramme horizontal permettant de conserver les moteurs sur la base (photo 2.2.6). La très bonne répétabilité ( $\pm 0.01$  [mm]) de ce robot est due essentiellement à des moteurs de très bonne linéarité du couple et à des capteurs de position à interférométrie laser de très haute résolution.

Plusieurs autres robots à entraînement direct ont été développés dans divers laboratoires au monde. A peu près tous rejoignent ceux cités précédemment en construisant des structures de type SCARA [Nihei 89, Iwakane 88] pour s'affranchir de la gravité, de type "five bar link" [Shimada & al 90] pour pouvoir ramener les moteurs vers la base du robot, ou encore de type SCARA avec parallélogramme horizontal [Takeshita 85, Kokkinis & al 87] afin de combiner ces deux derniers avantages. D'autres robots particuliers à entraînement direct ou semi-direct ont été développés selon des géométries propres à leurs applications [Takase 84, Suehiro & Takase 85, Arai & al 85, Hashimoto & al 87, Kazerooni & Kim 88].

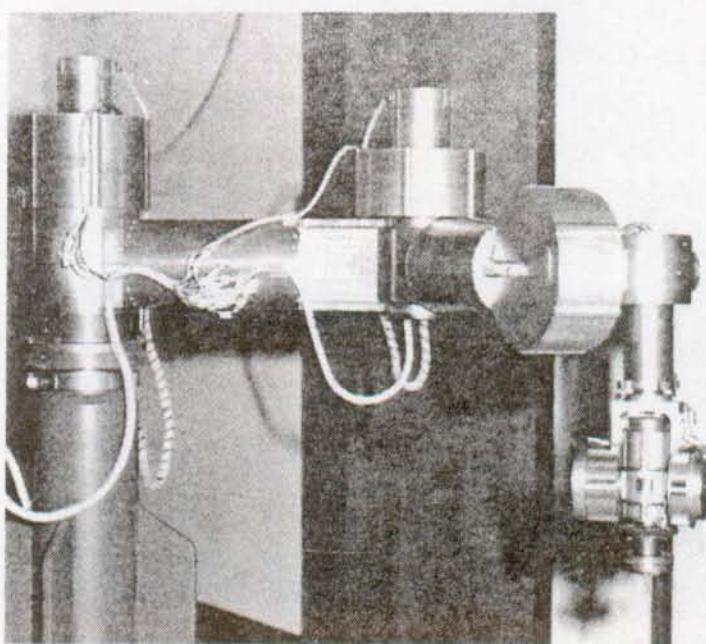
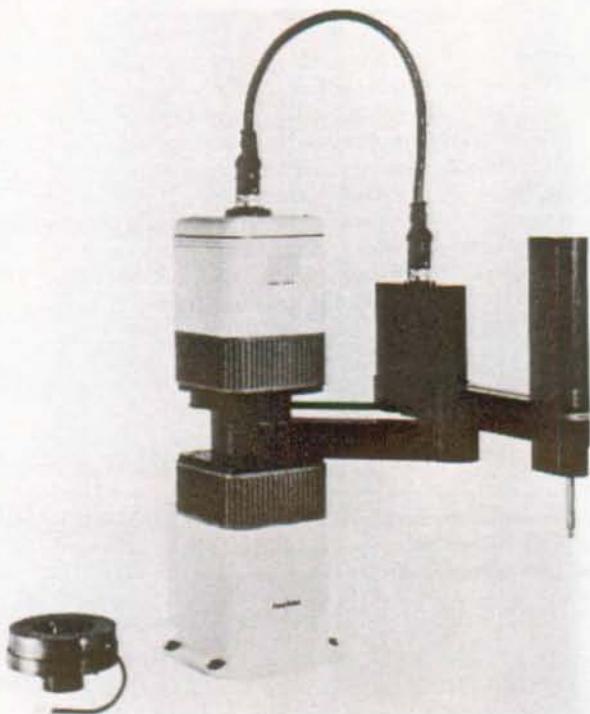


Photo 2.2.4 : CMU Direct Drive Arm II (tiré de [An & al. 88])



Photo 2.2.5 : Robot AdeptOne (tiré de [Adept 85])



*Photo 2.2.6 : Robot Matsushita (tiré de [Asada & Youcef-Toumi 87])*

## 2.3. Motorisation

### 2.3.1. Introduction

Le moteur et son amplificateur sont des éléments clés dans la conception des robots à entraînement direct. En effet, si le couple moteur peut être augmenté au travers d'un réducteur, l'élimination de ce dernier implique l'utilisation d'un moteur fournissant un couple beaucoup plus élevé. Pour ne pas pénaliser l'ensemble de la construction, le moteur doit être le plus léger et le plus petit possible pour le couple demandé.

D'autre part, les fluctuations du couple et de la vitesse doivent être minimisées si l'on désire obtenir de bonnes performances de positionnement. En se basant sur ces constatations, des moteurs spéciaux ont été réalisés pour la robotique à entraînement direct. On distingue trois technologies différentes :

- 1) moteurs couples à courant continu,
- 2) moteurs couples sans balais,
- 3) moteurs couples à réluctance variable.

Chacun de ces types est décrit brièvement dans les paragraphes qui suivent.

### 2.3.2. Moteurs couples à courant continu

Les moteurs à courant continu (c.c.), simples et performants, sont les actionneurs les plus utilisés en robotique. Leur utilisation pour l'entraînement direct nécessite une modification de leur conception afin d'obtenir des couples suffisamment élevés. Il n'y a cependant pas de différence essentielle entre un moteur c.c. et un moteur couple c.c., ce dernier étant conçu pour maximiser le couple de sortie plutôt que sa puissance.

On peut montrer [Asada & Youcef-Toumi 87] que le couple de sortie  $T$  d'un moteur c.c. est proportionnel à la longueur  $L_r$  du rotor et au carré de son diamètre  $D_r$ , soit<sup>1</sup> :

$$T^* = L_r^* D_r^{*2}.$$

---

<sup>1</sup>On utilise ici la notation des lois de similitude selon [Burckhardt 72]

Ainsi, une bonne méthode pour augmenter le couple d'un moteur c.c. est d'augmenter son diamètre plutôt que sa longueur. Ceci explique pourquoi les moteurs couples du marché sont généralement plus trapus que longs.

On sait aussi que le rapport couple/poids sera dépendant du type d'aimant que l'on utilise. Le choix d'aimants de grande "puissance massique" est donc primordial.

Pour un moteur c.c. idéal, le couple fourni est constant pour un courant donné quelle que soit sa position. En réalité, à l'approche des endroits de commutation, on observe généralement une baisse de couple. Aussi ce dernier varie lorsque le rotor tourne. Pour obtenir de bonnes performances de mouvement, surtout à basse vitesse, ces variations sont intolérables. Une disposition judicieuse des pôles permet de les minimiser. On obtient ainsi généralement de bonnes performances avec les moteurs couples c.c.. Le premier robot à entraînement direct, CMU DD Arm I [Asada & Kanade 81], utilisait de tels moteurs.

### 2.3.3. Moteurs couples sans balais

Les moteurs c.c., présentés précédemment, possèdent de nombreux avantages tels que : bon rendement, faibles variations de couples, caractéristique couple-vitesse linéaire, simplicité de construction, etc... . Cependant, leur grand désavantage est la commutation mécanique sur le collecteur qui crée des arcs électriques et donc une usure importante. D'autre part, le frottement des balais sur le collecteur n'est pas négligeable, ce qui perturbe les performances de positionnement.

Dans le cas des moteurs sans balais ("brushless"), la commutation mécanique est remplacée par un circuit électronique de commutation, éliminant les balais. Contrairement au moteur c.c., les aimants de ces moteurs sont situés sur le rotor, tandis que le bobinage se trouve sur le stator. Il s'agit là d'un double avantage, car de cette manière, on obtient d'une part un rotor de moindre inertie, et d'autre part une évacuation plus aisée de la chaleur. Ceci permet d'augmenter le courant dans le bobinage, et donc d'obtenir des performances accrues pour un même volume !

Comme déjà mentionné, les variations de couple suivant la position du moteur sont à éviter pour obtenir un bon moteur couple. Dans le cas des moteurs sans balais, ceci

est possible par une disposition judicieuse des pôles du moteur ainsi que par une stratégie de commutation adaptée.

Les avantages amenés par les moteurs sans balais (moins de frottement, moins d'usure, meilleurs rapports couple/inertie et couple/poids) justifient leur utilisation dans la plupart des robots à entraînement direct les plus récents.

#### **2.3.4. Moteurs couples à réluctance variable**

Un autre type de moteur à couple élevé et basse vitesse utilisé pour les robots à entraînement direct est le moteur à réluctance variable. Son principe de fonctionnement est similaire à celui d'un moteur pas à pas, mais au lieu de réaliser une commutation en boucle ouverte, celle-ci a lieu en fonction de la position du rotor. Avec une commande judicieuse, on peut éliminer en partie les variations de couple en fonction de la position, et avoir un nombre de points de positionnement quasi-infini, mais ceci reste très délicat.

Le moteur à réluctance variable a cependant un avantage sur le moteur sans balais. Il ne possède pas d'aimant, ce qui rend le couple réluctant inexistant. C'est ce genre de moteur qui a été choisi et développé pour le robot industriel AdeptOne.

### **2.4. Conséquences pour la conception mécanique du robot**

#### **2.4.1. Introduction**

Comme décrit dans le paragraphe précédent, les moteurs pour l'entraînement direct sont relativement lourds et volumineux. De ce fait, un robot de type série, avec un moteur placé sur chacune des articulations, amène à une structure lourde et peu élégante, les moteurs situés vers la base devant supporter et entraîner le poids des moteurs en amont ! On assiste alors à une croissance exponentielle des dimensions des moteurs de l'organe terminal à la base. Les performances dynamiques du robot s'en trouvent ainsi réduites. Le premier robot à entraînement direct construit, CMU DD Arm I, présentait ces défauts.

Pour éliminer ce problème, d'autres architectures mécaniques doivent être élaborées. Deux règles peuvent être formulées dans ce but :

- 1) Les moteurs ne doivent si possible pas supporter la gravité du bras.
- 2) Les moteurs doivent être situés autant que possible vers la base du robot, voir même immobiles.

#### **2.4.2. Problème de la gravité**

Lorsque le bras est immobilisé dans une position, son poids est une charge pour chacune des articulations. Si l'on utilise un réducteur, l'effet de la gravité est atténué par le facteur de réduction  $n_i$ . De plus, le réducteur introduit généralement du frottement qui agit dans la direction opposée au mouvement de l'articulation, diminuant encore l'effet de la gravité.

Si le réducteur est éliminé, c'est le moteur lui-même qui doit supporter toute la charge du bras. Il doit alors fournir des couples importants pendant une longue période et va chauffer. Il est nécessaire de trouver des solutions pour remédier à ce problème. Différentes possibilités ont été étudiées et réalisées en pratique. On peut citer par exemple :

- Utilisation d'un système d'équilibrage par ressorts, contre-poids ou vérin pneumatique
- Reprendre la force de gravité par les paliers en réalisant une construction judicieuse (ex. : robots SCARA)

#### **2.4.3. Redistribution des moteurs**

La distribution des moteurs sur chacune des articulations d'un robot série à entraînement direct amène immanquablement à une structure lourde et peu performante. Ainsi, pour la conception d'un tel robot, d'autres types de géométries doivent être réalisés. Si l'on veut placer les moteurs près de la base du robot pour les laisser immobiles, des mécanismes de transmission spéciaux doivent être utilisés. Les géométries développées pour résoudre ces problèmes ont généralement fait apparaître des structures parallèles ou à parallélogramme. Ceci contribue à diminuer la masse en

mouvement. Pour ce dernier point, l'utilisation de matériaux rigides et légers est aussi très importante. Plusieurs robots à entraînement direct ont vu leurs performances dynamiques augmenter par l'utilisation de matériaux composites à base de carbone, de kevlar ou autre [Asada & Youcef-Toumi 87, Kazerooni & Kim 88, Kokkinis & al 87].

#### **2.4.4. Capteurs de position angulaire**

Un autre point délicat dans la conception des robots à entraînement direct est l'utilisation de capteurs de position de très grande résolution. Les capteurs optiques incrémentaux permettent difficilement d'obtenir un nombre d'incrément suffisant tout en conservant une taille raisonnable. Pour avoir tout de même une bonne précision en bout de bras, des systèmes d'interpolation sont quelquefois utilisés. La plupart des robots à entraînement direct construits aujourd'hui utilisent plutôt des résolvers qui sont souvent intégrés au moteur lui-même [Welburn 84]. On atteint ainsi facilement 50'000 incrément par tour du moteur. De plus, l'utilisation de plusieurs résolvers déphasés entre eux permet des résolutions encore nettement plus grandes. Un autre capteur, développé pour les applications à entraînement direct est basé sur le principe de l'interférométrie laser et permet des résolutions élevées. [Kuwahara & al 86] proposent un capteur avec 1146880 incrément par tour. Ces capteurs sont malheureusement encore cher, et très peu sont disponibles sur le marché.

### **2.5. Conséquences pour la conception de la commande du robot**

#### **2.5.1. Introduction**

L'élimination du réducteur entraîne une augmentation de la taille et de la masse des moteurs. Dans le paragraphe précédent, il a été montré que ceci avait une influence non négligeable sur la conception mécanique du robot. Dans ce paragraphe, les effets du concept entraînement direct sur la commande du robot seront étudiés.

#### **2.5.2. Frottement**

Dans un robot, l'élimination de la réduction entraîne simultanément une diminution importante du frottement. Si cela est un avantage pour la modélisation du système (le

frottement est toujours très difficile à modéliser), il peut poser des problèmes pour la conception du régulateur. En effet, le facteur d'amortissement étant augmenté par le frottement, il est nécessaire de créer un effet analogue avec la boucle de régulation si l'on veut que le système se stabilise rapidement. L'utilisation d'une contre-réaction de vitesse (ou un terme déivateur dans la boucle de position) est un moyen simple et efficace de le réaliser. Cependant, le gain de la boucle de vitesse doit généralement être élevé pour obtenir de bonnes réponses. Une mesure précise de la vitesse, avec un faible niveau de bruit est alors nécessaire pour ne pas bruiter la commande. Malheureusement, avec un entraînement direct, la mesure des basses vitesses est plus délicate que pour les robots traditionnels, la vitesse n'étant pas multipliée par le facteur de réduction. Un capteur très performant approprié à la mesure des basses vitesses est alors nécessaire.

### **2.5.3. Rigidité de positionnement**

Un robot est soumis à diverses forces parasites (gravité, frottement, etc...). Lors d'un positionnement, il doit maintenir la position désirée malgré les charges extérieures. Sa capacité de réagir à ces forces statiques est définie par la rigidité de positionnement. Plus cette dernière est élevée, mieux le robot pourra les combattre.

La rigidité de positionnement est déterminée d'une part par la construction mécanique du robot, et d'autre part par les gains de contre-réaction des régulateurs de position. Pour un robot avec réducteur, les gains de la boucle, vus du bras, sont augmentés par le facteur de réduction  $n_i$  (comme le couple moteur). Avec un robot à entraînement direct, on ne peut pas profiter de cet avantage. Par conséquent, les gains du régulateur devront être beaucoup plus élevés. Ils seront cependant limités par les conditions de stabilité du système.

### **2.5.4. Effet de l'inductance du moteur**

Les moteurs pour l'entraînement direct sont conçus pour obtenir des couples élevés. Ceci amène à l'utilisation de bobinages de plus grandes dimensions et des moteurs plus volumineux.

La constante de temps électrique  $\tau_e$  est déterminée par le rapport de l'inductance moteur et de sa résistance, soit :

$$\tau_e = \frac{L}{R}$$

Les lois de similitudes [Burckhardt 72] nous disent que  $\tau_e$  augmente avec le carré des dimensions du moteur, soit :

$$\tau_e^* = l^{*2}$$

$l^*$  étant le rapport d'augmentation des dimensions.

L'augmentation de la constante de temps électrique peut alors causer des problèmes pour assurer la stabilité du système. Pour s'en affranchir, l'amplificateur d'un moteur à entraînement direct doit permettre des montées rapides du courant dans le moteur malgré la forte inductance. On réalise généralement ceci à l'aide d'une tension d'alimentation très élevée. D'autre part, le régulateur devra être dimensionné soigneusement pour avoir une marge de stabilité suffisamment grande.

### 2.5.5. Effet de la dynamique du bras

Généralement, le comportement dynamique d'un bras de robot est non-linéaire et couplé. En effet, d'une part son inertie dépend de sa configuration et d'autre part, l'accélération d'une articulation entraîne des forces parasites sur les autres articulations. Des forces non-linéaires dépendantes des vitesses de chacune des articulations agissent également sur la structure<sup>2</sup>.

Lorsque le bras est actionné à travers un réducteur  $n_i$ , les effets complexes de la dynamique du bras sont atténués lorsqu'ils apparaissent sur le moteur. En effet, vu du moteur on obtient alors l'équation suivante :

$$\Gamma_{mot} = \left( J_{rot} + \frac{J_{bras}}{n_i^2} \right) \cdot \ddot{\alpha}_i + \frac{\Gamma_{coup}}{n_i} + \frac{\Gamma_{n.l.}}{n_i}$$

avec :

$\Gamma_{mot}$  : couple exercé par le moteur

$\Gamma_{coup}$  : couplage entre les différents bras

$\Gamma_{n.l.}$  : couples non-linéaires du bras

---

<sup>2</sup>La modélisation de ces phénomènes sera traitée dans le chapitre 3.

$J_{\text{rot}}, J_{\text{bras}}$  : inerties du rotor et du bras respectivement

$\ddot{\alpha}$  : accélération du moteur

$n_i$  : rapport de réduction

Ainsi, plus le facteur de réduction est élevé, plus les phénomènes dynamiques non-linéaires et les couplages sont atténus. A la limite, ces termes peuvent être négligés lorsque  $n_i$  est suffisamment élevé. C'est d'ailleurs l'approximation qui est faite pour dimensionner les régulateurs de la plupart des robots sur le marché à l'heure actuelle.

Avec l'entraînement direct, où  $n_i = 1$ , cette approximation n'est plus valable et les effets non-linéaires de la dynamique sont directement ressentis sur le moteur. Ainsi, la régulation en est perturbée et des méthodes plus sophistiquées doivent être utilisées<sup>3</sup>. On notera ici que Asada et Youcef-Toumi [Asada & Youcef Toumi 87] proposent de réaliser une mécanique découplée pour éliminer ces effets non-linéaires et pour simplifier l'asservissement, mais les géométries développées sont alors trop restrictives et s'adaptent mal à une modification de la charge transportée.

## 2.6. Le robot DELTA à entraînement direct

Malgré les problèmes décrits dans les paragraphes précédents, l'entraînement direct est un concept très intéressant dans l'optique d'un robot rapide et précis. En effet, l'élimination de l'élasticité dans le réducteur permet des performances dynamiques accrues, et l'élimination du jeu augmente la précision du robot. D'autre part, pour obtenir un robot rapide, il est aussi nécessaire d'avoir la masse mobile la plus faible possible. Le poids des moteurs est une part non négligeable du poids total d'un robot série habituellement rencontré sur le marché. Rechercher une géométrie telle que les moteurs restent fixes amène presque exclusivement au concept de robot parallèle. Cependant, on leur reproche souvent d'avoir un volume de travail trop restreint. Ce problème a été résolu par R. Clavel [Clavel 88] grâce au robot parallèle DELTA.

---

<sup>3</sup>Différentes méthodes sont exposées au chapitre 6.

Ce nouveau concept est très intéressant, car il promet des performances dynamiques élevées. Lié à l'entraînement direct, c'est le mariage idéal pour obtenir l'un des robots les plus rapides au monde.

### 2.6.1. Le robot DELTA

Le concept DELTA a été développé à l'institut de Microtechnique de l'école polytechnique fédérale de Lausanne (IMT-EPFL) par R. Clavel [Clavel 88] dans le but de déplacer des objets légers (0 à 20 grammes) à cadence élevée. Son développement était surtout conditionné par l'industrie du chocolat pour la mise en boîte de pralinés. Ce robot parallèle à 4 degrés de liberté ( $x, y, z, \theta_z$ ) peut déjà atteindre une cadence de travail environ 2 fois supérieure à celle des robots les plus rapides actuellement sur le marché.

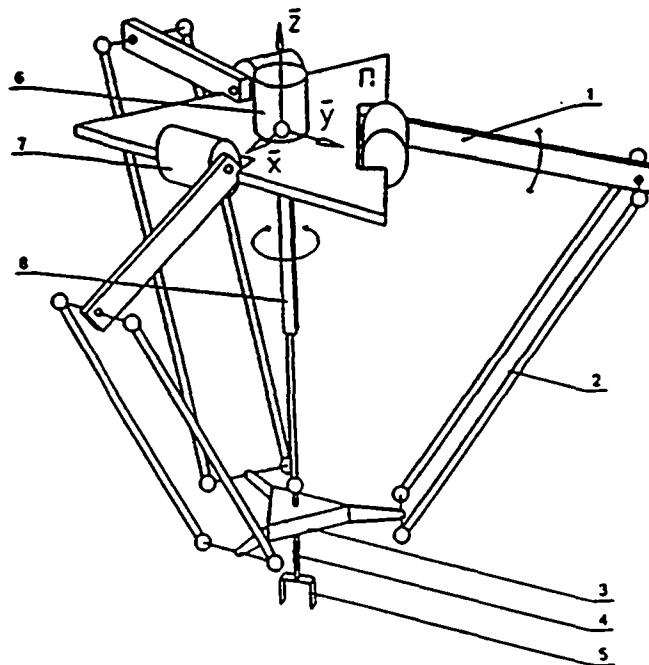


Figure 2.6.1 : Le robot DELTA

Un robot parallèle est caractérisé par le fait que la liaison entre la base fixe et l'extrémité mobile porteuse du préhenseur est assurée par plus d'une chaîne cinématique. Pour le robot DELTA (figure 2.6.1), la liaison de la nacelle (3) à la base est assurée par trois chaînes cinématiques identiques. Chacune d'entre elles est entraînée par un moteur (7) solidaire du bâti. Le déplacement de la nacelle dans

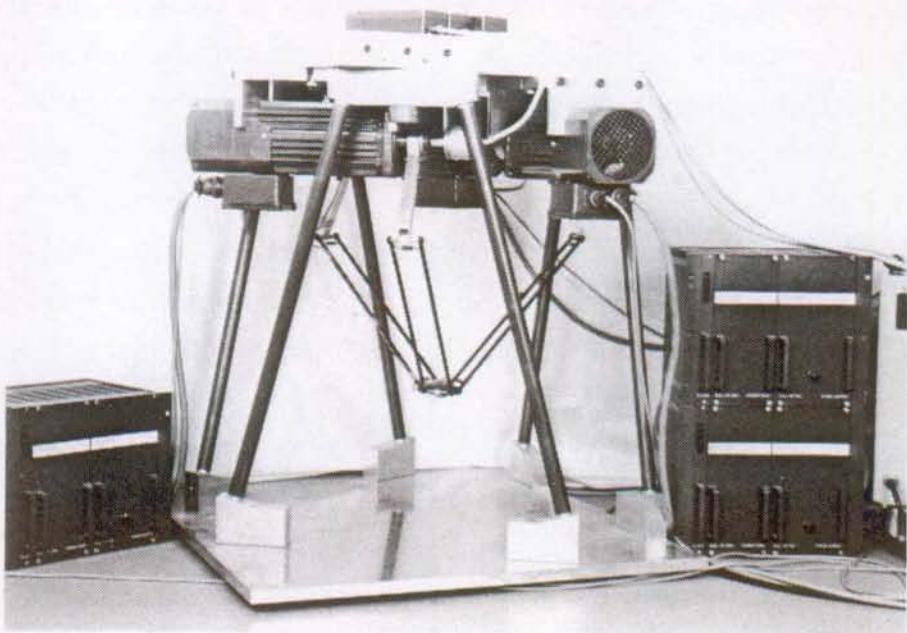
l'espace (x,y,z) est assuré par la combinaison des mouvements des bras (1) qui lui sont transmis par l'intermédiaire de trois paires de barres parallèles ou avant-bras (2) qui assurent également le maintien de son orientation. Ainsi, la nacelle reste constamment parallèle au plan  $\Pi$  des moteurs. Ce système est donc un parallélogramme de l'espace. Le préhenseur (5) est monté en pivotement perpendiculairement à la nacelle. Sa rotation est assurée par le moteur (6) au travers d'un arbre télescopique (8) muni d'un cardan à chacune de ses extrémités. Le moteur (6) est également solidaire du bâti. Par un choix approprié des matériaux de construction (aluminium et fibres de carbone), la structure mobile est très légère, ce qui permet des mouvements très rapides.

## 2.6.2. DELTA à entraînement direct

Par le fait que ses moteurs ne se déplacent pas lors du mouvement de la nacelle, le concept DELTA est très intéressant pour l'entraînement direct. Ainsi, le poids du moteur ne nuit pas aux performances dynamiques du robot. Le choix du moteur est alors relativement aisé, puisque les seules contraintes qui interviennent sont liées à ses performances d'accélération. La photo 2.6.2 montre le prototype réalisé, où le bras est accouplé directement sur l'axe du moteur. L'effet de la gravité par contre n'est pas complètement éliminé avec cette géométrie, mais les très faibles masses de la structure mécanique font que le couple à supporter par les moteurs est faible.

### *Caractéristiques géométriques et volume de travail*

Selon la formulation de [Clavel 91], le prototype réalisé est du type 580 (longueur du bras + longueur de l'avant-bras d'une chaîne cinématique, exprimées en [mm]). Son volume de travail, relativement complexe peut être ramené à un cylindre de 250 [mm] de rayon et de 140 [mm] de hauteur. Par soucis de simplification de la construction mécanique, le 4ème degré de liberté n'a pas été construit, ce qui ne va pas modifier les résultats essentiels de ce travail.



*Photo 2.6.2 : Prototype du robot DELTA à entraînement direct*

#### *Moteurs et amplificateurs*

Les moteurs (BLS-63-24-PSM-4-132) utilisés sont de type "brushless" fabriqués par la maison Landert et pilotés par un amplificateur de la maison Sulzer. Ils peuvent fournir un couple nominal de 7.5 [Nm] et jusqu'à 21.5 [Nm] en pointe. Il faut relever ici que ces moteurs ne sont pas des moteurs couples. La gamme des moteurs couples disponibles sur le marché à l'heure actuelle ont des vitesses de rotation maximales inférieures à 200 [t/min], ce qui est insuffisant pour les cadences visées. Le moteur choisi ici est trop puissant pour l'application, donc cher. En cas d'industrialisation du robot DELTA à entraînement direct, il serait alors souhaitable de développer un moteur spécial.

#### *Capteurs*

Avec les dimensions géométriques choisies, pour obtenir sur la nacelle une résolution d'environ 0.03 [mm], un capteur avec 40'000 incrément/tour est nécessaire. Pour ceci, un capteur incrémental de type Heidenhain ROD 456 à 2000 traits par tour est utilisé. Le signal de sortie du capteur est un "pseudo-sinus" et une

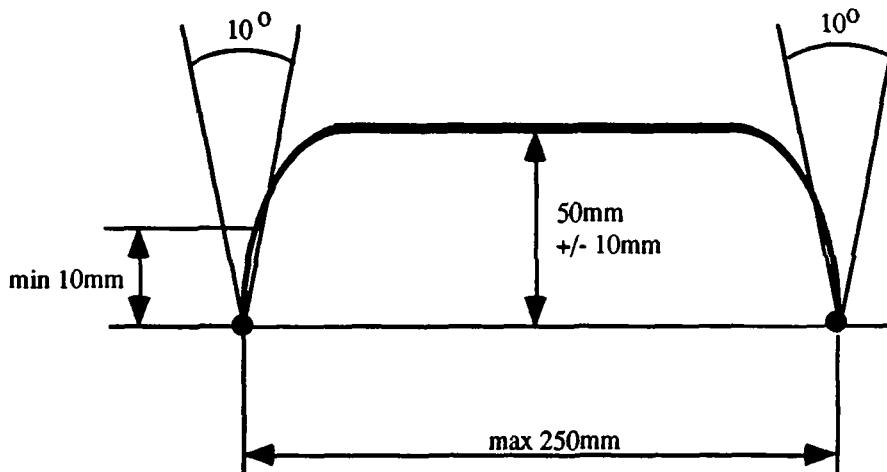
interpolation peut être effectuée pour augmenter la résolution. Pour cela, une carte Heidenhain EXE 602D est utilisée et permet d'obtenir la résolution désirée.

### 2.6.3. Cahier des charges

Avec ce nouveau prototype, des applications de palettisation de pièces légères sont envisagées. Un cahier des charges raisonnable répondant aux besoins industriels peut alors être établi comme suit :

- trajectoires : mouvements de prise et dépose d'objets conformément à la figure 2.6.3.
- charge à transporter inférieure à 20 [g]
- éloignement des points de prise et dépose : 250 [mm]
- hauteur de transit de 50 [mm]  $\pm 10$ [mm]
- accostage dans un cône d'ouverture  $10^\circ$ , avec hauteur minimale d'entrée de 10 [mm]
- dépassement maximal dans la direction de prise ou dépose :  $\leq 1$  [mm]
- cadences visées :  $\geq 3$  aller-retour par seconde

On peut préciser que les cadences maximales obtenues pour les robots classiques sont généralement inférieures à 1 [Hz] [Cronshaw 90]. Le robot DELTA à entraînement direct pourrait donc être 3 à 4 fois plus rapide.



*Figure 2.6.3 : Trajectoire de prise et dépose*

## 2.7. Conclusion

Ces dernières années, la recherche et le développement des robots à entraînement direct ont été menés parallèlement dans l'industrie et dans les milieux universitaires. La réussite de ces projets a surtout été marquée par le développement de moteurs, de capteurs et de structures mécaniques adaptés à l'entraînement direct ainsi que de systèmes de commande performants basés sur le modèle dynamique des robots.

Le robot parallèle DELTA, avec ses moteurs fixes sur le bâti, a une structure géométrique très favorable à la mise en oeuvre d'un entraînement direct. Cela en vaut la peine, car les avantages apportés par ce nouveau concept sont importants :

- structure mécanique simple
- élimination des jeux et de l'usure des réducteurs
- diminution des frottements
- élimination du bruit
- plus grande rigidité
- augmentation des performances dynamiques.

Le potentiel du robot DELTA à entraînement direct est grand. Il permet d'effectuer des travaux de palettisation 3 à 4 fois plus rapidement que les robots les plus rapides sur le marché actuellement, mais pour cela, il est nécessaire de développer une commande performante pour allier rapidité et précision. C'est ce que proposent les chapitres qui suivent.

## 3. Modèle dynamique

### 3.1. Introduction

Tout système physique en mouvement peut être décrit par un modèle dynamique. Dans le cas d'un robot, on en distingue généralement deux types : les modèles dynamiques direct et inverse.

Par le modèle dynamique direct, on cherche à connaître le mouvement de l'organe terminal du robot lorsqu'on applique un certain couple (ou force) sur les articulations. Ce modèle est très utile pour la simulation du comportement d'un robot à divers algorithmes de commande. Il est d'ailleurs utilisé intensivement pour le développement de nouveaux types de régulateurs pour robots.

Le modèle dynamique inverse, par opposition, exprime le couple (ou force)  $\Gamma$  que chaque actionneur doit fournir pour obtenir le déplacement désiré de l'organe terminal du robot. L'expression de ces couples est non-linéaire et dépend des positions  $q$ , vitesses  $\dot{q}$  et accélérations  $\ddot{q}$  de chacune des articulations du robot, soit :

$$\Gamma = f(q, \dot{q}, \ddot{q})$$

La connaissance du modèle dynamique inverse est très intéressante en pratique. En effet, comme ce dernier permet de déterminer les couples nécessaires pour un mouvement désiré, il permet d'une part de dimensionner les actionneurs au plus près de la réalité, mais surtout peut être utilisé par le système d'asservissement pour améliorer les performances lors du suivi d'une trajectoire<sup>1</sup>. Dans ce cas, une formulation permettant son évaluation en temps réel doit être développée.

Le modèle dynamique d'un robot est directement dépendant des lois de la mécanique classique. Plusieurs méthodes existent pour l'établir. Les plus connues sont :

- méthode de Newton-Euler
- méthode de Lagrange
- méthode de d'Alembert (travaux virtuels)

---

<sup>1</sup>Voir le chapitre 7.

La méthode de Newton-Euler est basée sur les lois de la mécanique de Newton. De nombreux travaux ont été réalisés ces dernières années en utilisant cette méthode [Luh & al 80, Khosla 86, Khalil & Kleinfinger 87], surtout parce qu'elle a l'avantage d'aboutir, pour les robots série, à un nombre minimal d'opérations et peut ainsi être utilisée en temps réel dans l'algorithme de commande. La méthode de Lagrange est basée sur l'expression des énergies dans le système. Sa formulation est plus simple et plus élégante que celle de Newton-Euler, mais elle est quasi-inutilisable en temps réel [Paul 81, Megahed 84, Dombre & Khalil 88]. La méthode de d'Alembert, quant à elle, est basée sur la notion de travail virtuel. Les travaux utilisant cette dernière sont peu nombreux [Lee & al 83]. Il s'agit en fait d'un compromis entre les deux précédentes et de ce fait ne présente actuellement pas d'intérêt direct pour la modélisation des robots série.

Les travaux cités ci-dessus concernent tous l'établissement de modèles dynamiques pour des robots à structure série. Pour les robots parallèles, le problème est beaucoup plus complexe, en raison de l'interdépendance des variables articulaires. Les travaux permettant de modéliser ces robots sont encore peu nombreux, et les résultats obtenus sont souvent spécifiques à la structure du robot modélisé. [Reboulet 88] se base sur la méthode de Newton-Euler pour la modélisation d'une plate-forme de Stewart. Une tentative de généralisation a été entreprise par [Kleinfinger 86] en utilisant le modèle dynamique (selon Newton-Euler) d'une structure arborescente équivalente et le calcul des multiplicateurs de Lagrange pour décrire les contraintes de fermeture des boucles. La méthode est intéressante, mais son utilisation est encore fastidieuse. [Clavel 91] présente un modèle numérique basé sur la méthode de d'Alembert pour le robot DELTA. Malgré une formulation simple, ce modèle n'est pas optimal pour une utilisation efficace en temps réel en raison de son temps de calcul élevé. Une nouvelle formulation est donc nécessaire pour son incorporation dans la commande.

Dans ce chapitre, un nouveau modèle dynamique du robot DELTA est étudié et développé pour son utilisation en temps réel. Il se distingue des modèles traditionnels par une formulation dépendant non seulement des variables articulaires, mais aussi de la cinématique de l'organe terminal. Ceci contribue à la simplification de sa formulation mathématique et donc à réduire le temps de calcul. Celui-ci est évalué en moins de 750 [ $\mu$ s] par un transputer T800. La méthode de Newton-Euler, utilisée pour son établissement, est préalablement exposée. Une

étude détaillée de chacun des termes du modèle et leur influence sur le comportement global du robot est aussi réalisée. Finalement, le modèle est validé par une expérimentation sur le système réel.

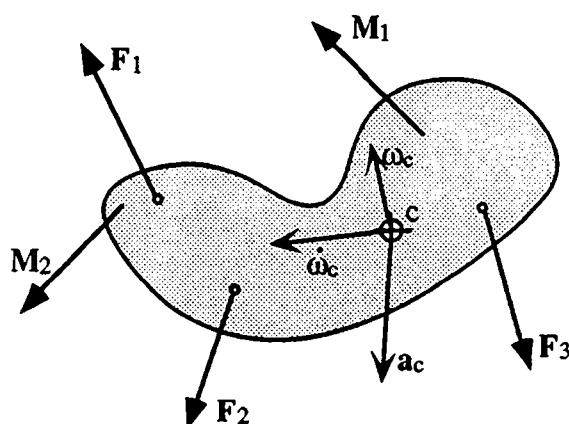
## 3.2. Méthode de Newton-Euler

### 3.2.1. Introduction

Chaque composant d'un robot peut être considéré comme un corps rigide. En connaissant son centre de masse et son tenseur d'inertie, toute sa distribution de masse est caractérisée. Pour déplacer ce corps dans l'espace, il est nécessaire de l'accélérer, puis de le décélérer. Son mouvement peut être décomposé en un mouvement de translation de son centre de masse ainsi qu'un mouvement de rotation autour de celui-ci. L'équation de Newton décrit alors les forces nécessaires pour le cas de la translation, alors que l'équation d'Euler décrit les couples pour le cas de la rotation.

### 3.2.2. Equations de Newton et d'Euler

Soit le corps rigide de la figure 3.2.1. Les différentes forces et couples agissant sur cette structure créent une accélération de translation du centre de masse ainsi qu'une accélération angulaire autour de ce dernier.



*Figure 3.2.1 : Corps rigide en mouvement*

L'équation de Newton nous donne alors :

$$\sum_i \mathbf{F}_i = m \cdot \mathbf{a}_c \quad [3.2.1]$$

avec :  $\mathbf{F}_i$  forces agissant sur la structure,  
 $m$  masse du corps,  
 $\mathbf{a}_c$  accélération de son centre de masse.

L'équation d'Euler est utilisée pour obtenir l'accélération angulaire, soit :

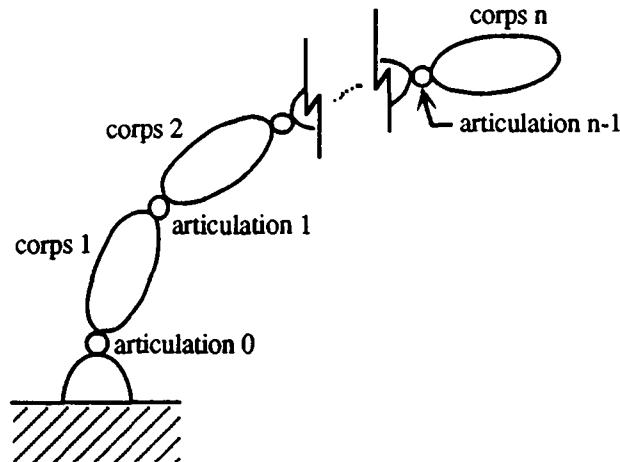
$$\sum_i \mathbf{M}_i = \mathbf{I}_c \dot{\omega}_c + \omega_c \wedge (\mathbf{I}_c \cdot \omega_c) \quad [3.2.2]$$

avec :  $\mathbf{M}_i$  couples agissant sur le centre de masse  
 $\mathbf{I}_c$  tenseur d'inertie du corps lié à un repère solidaire du centre de masse  
 $\omega_c$  vitesse angulaire du corps rigide  
 $\dot{\omega}_c$  accélération angulaire du corps rigide.

Le premier terme de l'équation correspond à l'effet de l'accélération angulaire alors que le deuxième résulte des variations de direction du vecteur vitesse. Ce dernier terme est généralement connu sous le nom de "couple gyroscopique".

### 3.2.3. Utilisation pour le calcul du modèle dynamique inverse d'un robot

Un robot peut être considéré comme une suite de corps rigides liés entre eux par des articulations (figure 3.2.2).



*Figure 3.2.2 : Un robot est une suite de corps rigides liés par des articulations*

Pour pouvoir effectuer un mouvement, chacune des articulations actives doit fournir un couple (ou force)  $\Gamma_i$ . Pour le calculer, on utilise alors les équations de Newton-Euler sur chacun des corps de la structure.

Pour isoler un corps, il est nécessaire de lui ajouter, sur chacune des articulations, les forces ou couples de liaison avec les corps voisins. Ceci est illustré par la figure 3.2.3, où l'on tient compte de la force de gravité  $m_i g$ , mais l'on néglige les forces de frottement et autres forces secondaires.

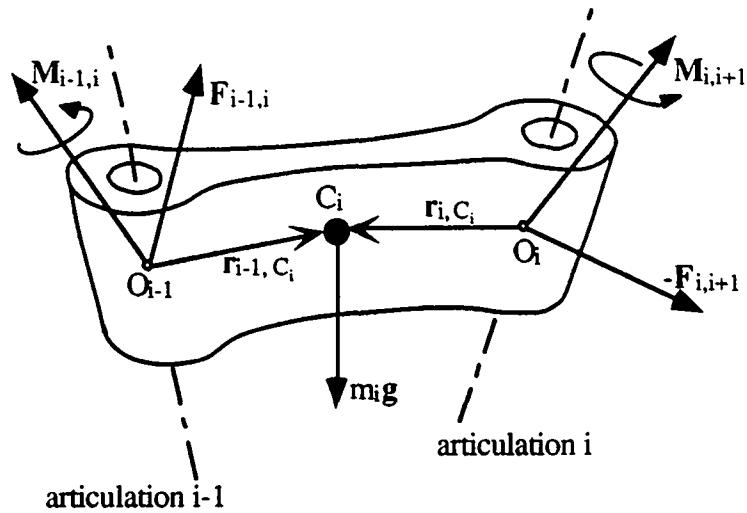


Figure 3.2.3 : isolation du corps i

Dans cette figure, les forces  $\mathbf{F}$  et les couples  $\mathbf{M}$  représentent les forces et couples de liaison avec les corps voisins.

L'application de l'équation de Newton sur le corps i donne alors :

$$\mathbf{F}_{i-1,i} - \mathbf{F}_{i,i+1} + m_i \cdot \mathbf{g} = m_i \cdot \mathbf{a}_{c_i} \quad [3.2.3]$$

Pour l'équation d'Euler, on obtient aussi :

$$\mathbf{M}_{i-1,i} - \mathbf{M}_{i,i+1} + \mathbf{r}_{i,c_i} \wedge \mathbf{F}_{i,i+1} - \mathbf{r}_{i-1,c_i} \wedge \mathbf{F}_{i-1,i} = \mathbf{I}_i \cdot \dot{\omega}_i + \omega_i \wedge (\mathbf{I}_i \cdot \omega_i) \quad [3.2.4]$$

Les équations (3.2.3) et (3.2.4) décrivent les caractéristiques dynamiques d'un corps isolé. Le modèle dynamique complet pour tout le robot est alors obtenu en évaluant ces 2 équations pour tous les corps du robot, soit pour  $i = 1, \dots, n$ .

En résolvant ce système d'équations, on aboutit à la forme générale de la dynamique inverse d'un robot, soit :

$$\Gamma = \mathbf{A}(\mathbf{q}) \cdot \dot{\mathbf{q}} + \mathbf{V}(\mathbf{q}, \dot{\mathbf{q}}) + \mathbf{G}(\mathbf{q}) \quad [3.2.5]$$

où  $\Gamma = (\Gamma_1, \Gamma_2, \dots, \Gamma_n)^T$  est le vecteur couple généralisé (couple ou force) que chacune des articulations doit fournir pour effectuer le mouvement

$\mathbf{q}(t) = (q_1(t), \dots, q_n(t))^T$ ,  $\mathbf{A}(\mathbf{q})$  est la matrice d'inertie du robot,  $\mathbf{V}(\mathbf{q}, \dot{\mathbf{q}})$  est un vecteur comprenant les effets centrifuges et Coriolis, et  $\mathbf{G}(\mathbf{q})$  est le vecteur contenant les termes de gravité. Chaque élément de  $\mathbf{A}(\mathbf{q})$  et  $\mathbf{G}(\mathbf{q})$  est une fonction non-linéaire qui dépend de la position de toutes les articulations du robot. Chaque élément de  $\mathbf{V}(\mathbf{q}, \dot{\mathbf{q}})$  est une fonction non-linéaire de la position, mais aussi de la vitesse de chacune des articulations.

### 3.3. Modèle dynamique temps réel pour le robot DELTA

Le robot DELTA, présenté dans le chapitre 2, a une structure parallèle qui rend son modèle dynamique difficile à obtenir en raison de l'interdépendance de ses variables articulaires. Le modèle dynamique présenté par [Clavel 91] a été intensivement utilisé en simulation afin de déterminer la configuration du robot la plus favorable à une minimisation des couples appliqués aux moteurs. La méthode de d'Alembert utilisée, a l'avantage d'être générale, mais amène à un temps de calcul trop important pour une utilisation raisonnable en temps réel. Un nouveau modèle a donc dû être développé et est présenté dans ce paragraphe.

#### 3.3.1. Paramétrisation du système et référentiels

##### Référentiels

La figure 3.3.1 montre le robot DELTA à 3 degrés de liberté qui sera traité ici. En raison de sa symétrie ternaire, il sera possible par la suite de traiter chaque chaîne cinématique  $i$  indépendamment en attribuant à chacune un référentiel  $R_i(O_i, x_i, y_i, z_i)$  (figure 3.3.1). Le référentiel de base  $R(O, x, y, z)$  est placé au centre de la base fixe du robot, son axe  $z$  est vertical et son axe  $x$  dans le plan de l'un des bras. Les référentiels  $R_i$  sont disposés de la même manière, pour chacun des bras  $i$ . Ils sont séparés du référentiel de base  $R$  d'un angle  $\theta_i$  autour de l'axe  $z$  et ont même origine ( $O_i = O$ ). La matrice de passage du référentiel de base  $R$  à chacun des référentiels  $R_i$  est alors donnée par :

$$P_i = \begin{pmatrix} \cos \theta_i & \sin \theta_i & 0 \\ -\sin \theta_i & \cos \theta_i & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad [3.3.1]$$

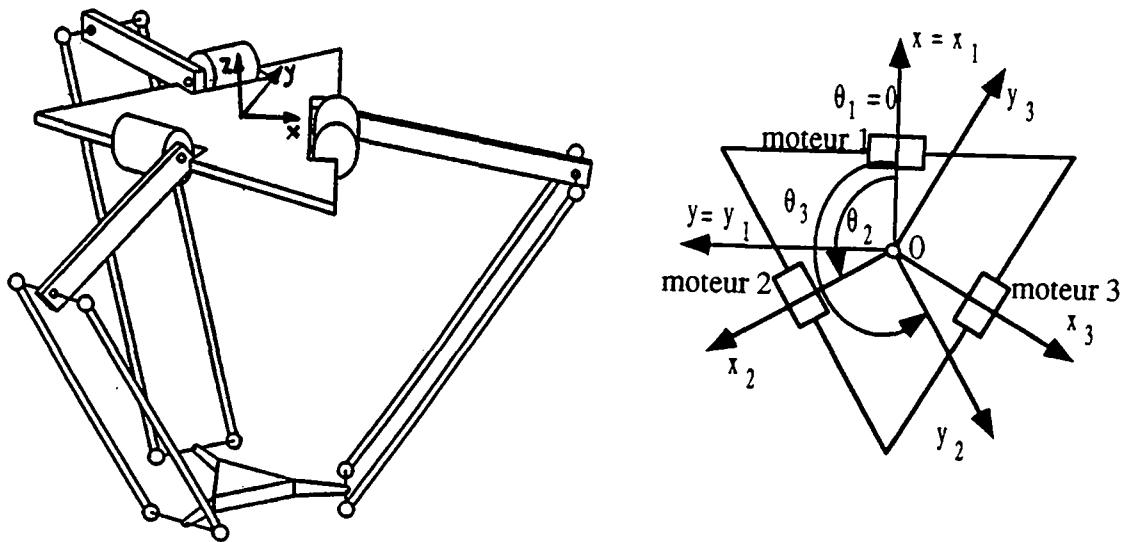


Figure 3.3.1 : Robot DELTA à 3 degrés de liberté et ses référentiels associés

#### Paramètres géométriques

Dans chacun des référentiels  $R_i$ , on peut alors représenter le bras  $i$  selon la figure 3.3.2. Les paramètres de la structure sont :

- $R_A$  : distance entre le centre de la base ( $O_i = O$ ) et l'axe de rotation du bras
- $R_B$  : distance entre le centre de la nacelle ( $P_i$ ) et l'axe de liaison inférieur du parallélogramme ( $B_i$ )
- $L_A$  : longueur du bras
- $L_B$  : longueur de l'avant-bras
- $R$  :  $= R_A - R_B$ . Ceci permettra, par la suite, de simplifier la représentation géométrique.
- $\alpha_i$  : angle entre le bras  $i$  et l'axe  $x_i$  du référentiel  $R_i$ . Par convention, il est défini positif lorsque le bras se situe du côté de la nacelle.
- $\beta_i$  : angle entre le plan du parallélogramme et le plan horizontal, mesuré dans un plan vertical  $\pi_i$  contenant le bras  $i$ .
- $\gamma_i$  : angle entre le plan vertical  $\pi_i$  et une tige du parallélogramme  $i$ .

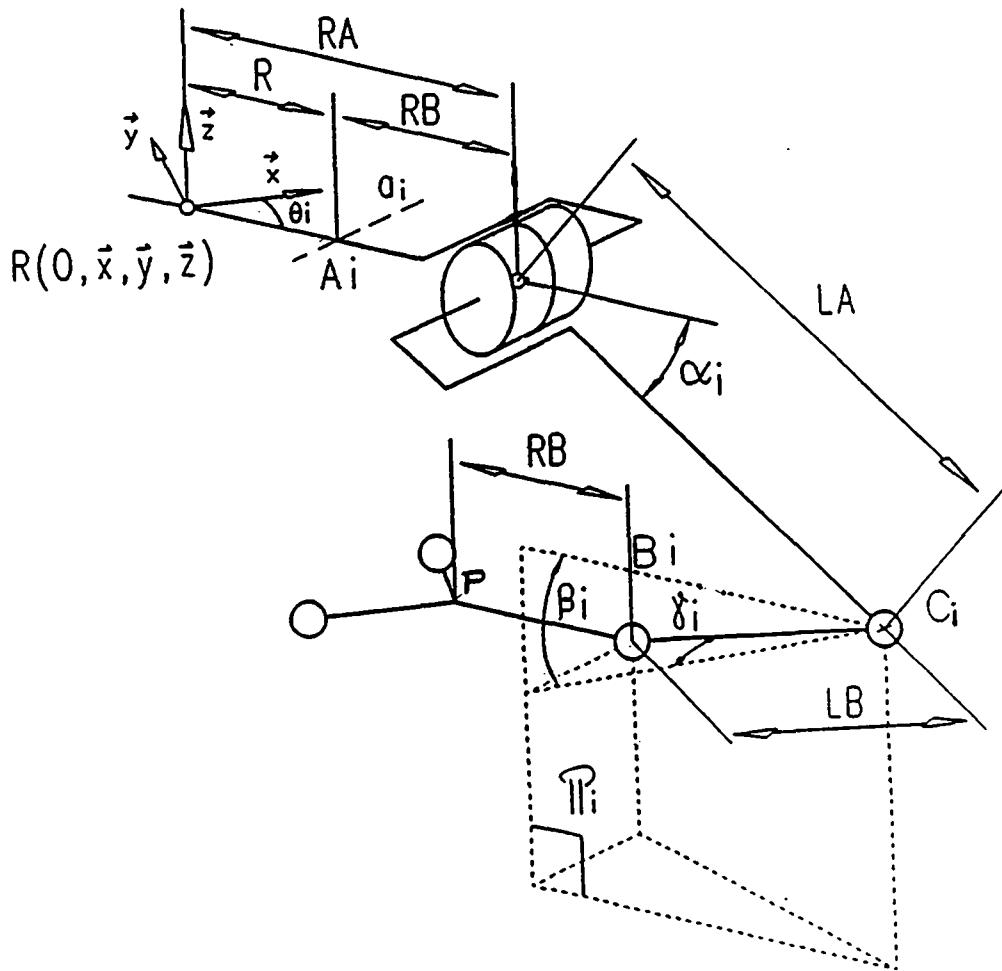


Figure 3.3.2 : Paramètres géométriques du robot DELTA

### 3.3.2. Modèle "complet"

L'application de la méthode de Newton-Euler sur tous les corps du robot DELTA nécessite aussi de connaître leurs accélérations. Si les accélérations des bras et de la nacelle sont bien connues, le mouvement réalisé par les avant-bras (barres parallèles) est beaucoup plus complexe. La valeur et la direction des forces de liaison permettant d'isoler les barres parallèles deviennent alors du même ordre de complexité que

l'expression de leur cinématique. Un modèle "complet"<sup>1</sup> tenant compte de l'effet des avant-bras a été établi par C. Dayer [Dayer 89]. La formulation est très dense, et la résolution du système relativement complexe. Les résultats obtenus sont comparables à ceux calculés par la méthode des travaux virtuels [Clavel 91], et le temps de calcul est quasiment le même.

### 3.3.3. Hypothèses simplificatrices

Tenir compte de tous les éléments du robot DELTA amène ainsi à un temps de calcul trop important. Une simplification judicieuse est alors nécessaire pour obtenir le modèle le plus précis tout en ayant un temps de calcul minimal.

La complexité du modèle complet vient essentiellement du mouvement des barres parallèles. Négliger leur inertie simplifie alors grandement le problème, puisque la force de liaison entre la nacelle et le bras est alors orientée dans la direction des barres parallèles. Cette hypothèse n'est d'ailleurs pas trop restrictive en raison de l'utilisation de fibres de carbone comme matériau de construction des barres parallèles. Ainsi, la masse négligée n'est pas très importante comparée à celle de la nacelle. En répartissant la masse des barres parallèles à leurs deux extrémités on limite encore les erreurs du modèle.

Les hypothèses simplificatrices pour l'établissement du modèle sont donc :

- la masse d'un avant-bras  $m_{ab}$  (barres parallèles) est répartie à ses deux extrémités selon la proportion suivante ( $0 < r < 1$ ) (voir § 3.5.1 pour la détermination de  $r$ ) :
  - $r \cdot m_{ab}$  sur sa partie supérieure (coude)
  - $(1-r) \cdot m_{ab}$  sur sa partie inférieure (nacelle)
- le moment d'inertie des avant-bras est négligé
- les frottements sont négligés

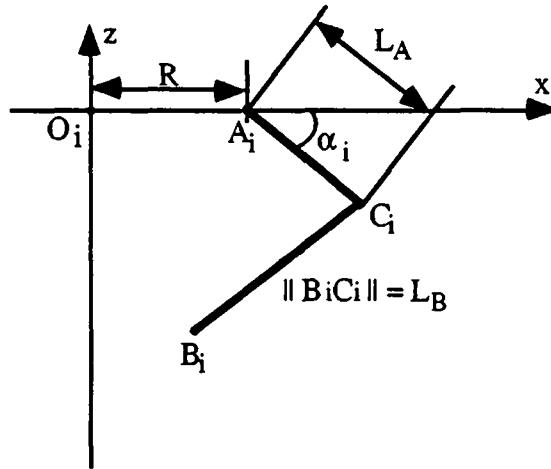
### 3.3.4. Modèle simplifié

En tenant compte des hypothèses simplificatrices citées ci-dessus, on peut alors établir le modèle dynamique du robot DELTA simplement. Chaque parallélogramme peut être remplacé par une tige unique sans masse comme représenté dans la figure

---

<sup>1</sup>Par modèle "complet", on entend un modèle qui tient compte de l'effet de tous les corps du robot, mais pas de l'effet du frottement.

3.3.3. Ici, seule la méthode de résolution, avec ses étapes principales est présentée. Le lecteur désirant connaître les détails du calcul est prié de se référer à l'annexe B.



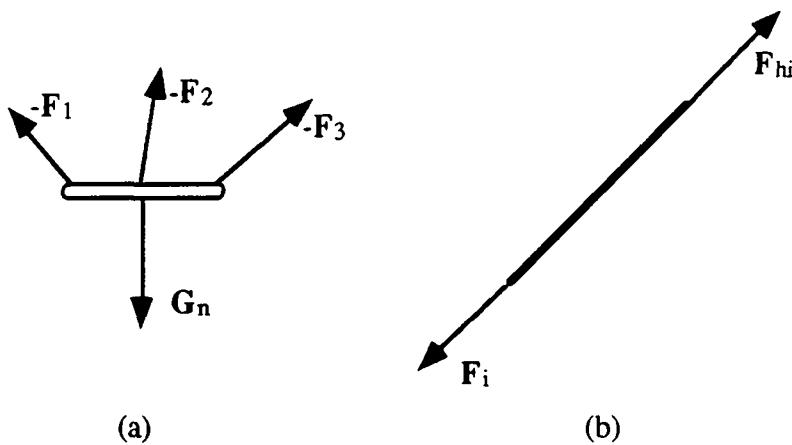
*Figure 3.3.3 : Chaîne cinématique dans le repère  $R_i$*

On isole tout d'abord la nacelle en lui ajoutant les forces de liaison  $\mathbf{F}_i$  avec chacun des trois avant-bras (figure 3.3.4). En y appliquant l'équation de Newton dans le référentiel  $R$ , on obtient alors :

$$-\sum_{i=1}^3 \mathbf{F}_i + \mathbf{G}_n = m_n \cdot \ddot{\mathbf{X}} \quad [3.3.2]$$

avec :

- $\mathbf{F}_i$  forces de liaison entre la nacelle et les avant-bras, exprimées dans le référentiel  $R$ .
- $\mathbf{G}_n$  force de gravité agissant sur la nacelle.
- $m_n$  masse totale de la nacelle, y compris la masse transportée et la masse des avant-bras rapportée sur la nacelle.
- $\ddot{\mathbf{X}}$  accélération de la nacelle exprimée dans le repère  $R$ .



*Figure 3.3.4 : isolation de la nacelle (a) et d'un avant-bras (b)*

Il est à noter ici que la construction mécanique empêche tout mouvement de rotation de la nacelle. Les efforts sont entièrement repris par les paliers et ne contribuent pas au couple moteur.

L'isolation de l'avant-bras dont on considère la masse et le moment d'inertie nuls nous donne alors (figure 3.3.4.b) :

$$\mathbf{F}_i + \mathbf{F}_{hi} = 0 \quad [3.3.3]$$

Les deux forces sont donc opposées et dans la direction  $\mathbf{B}_i\mathbf{C}_i$  de l'avant-bras. Pour éviter de passer par le calcul des angles intermédiaires  $\beta_i$  et  $\gamma_i$  (voir figure 3.3.2), on peut calculer ce vecteur de la manière suivante :

$$\mathbf{B}_i\mathbf{C}_i = \mathbf{B}_i\mathbf{O}_i + \mathbf{O}_i\mathbf{A}_i + \mathbf{A}_i\mathbf{C}_i \quad [3.3.4]$$

En connaissant la direction des forces  $\mathbf{F}_i$  il est alors possible de calculer leur norme à partir de l'équation [3.3.2]. En effet, en posant :

$$\mathbf{F}_i = F_i \frac{\mathbf{B}_i\mathbf{C}_i}{|\mathbf{B}_i\mathbf{C}_i|} \quad [3.3.5]$$

et en l'introduisant dans [3.3.2], on obtient un système de 3 équations dont les 3 inconnues sont  $F_1$ ,  $F_2$ ,  $F_3$ . Ainsi, les 3 forces de liaison  $\mathbf{F}_i$  sont entièrement définies (norme et direction).

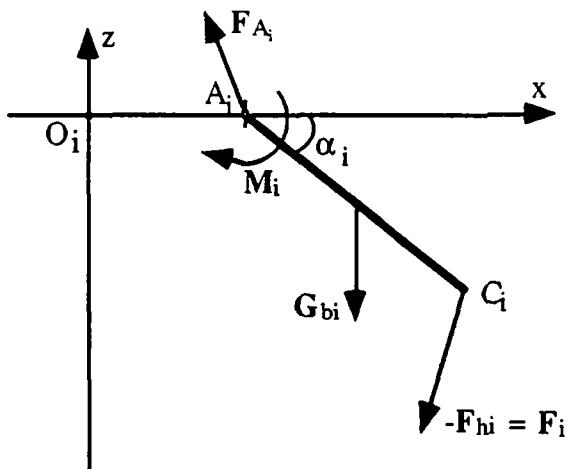


Figure 3.3.5 : isolation du bras i

En se plaçant maintenant dans le référentiel  $R_i$  lié à chacun des bras, on peut calculer le couple moteur en utilisant l'équation d'Euler (figure 3.3.5). On obtient :

$$A_i C_i \wedge F_i + r_{A_i, G_b} \wedge G_{bi} + M_i = I_{bi} \ddot{\alpha}_i \quad [3.3.6]$$

avec :

- $r_{A_i, G_b}$  vecteur exprimant la position du centre de gravité du bras  $i$ .
- $G_{bi}$  force de gravité du bras  $i$  exprimée dans le référentiel  $R_i$ .
- $M_i$  couple autour du point  $A_i$ .
- $I_{bi}$  moment d'inertie du bras autour du point  $A_i$ .
- $\ddot{\alpha}_i$  accélération angulaire du bras  $i$ .

Dans cette équation, le terme de couple gyroscopique est nul, car le vecteur vitesse de rotation du bras est toujours dans la même direction. Seul le couple autour de l'axe  $y$  nous intéresse, puisqu'il s'agit du couple moteur. Les autres couples sont repris par les paliers. De même, il n'est pas nécessaire de poser ici l'équation de Newton, car la force de liaison au point  $A_i$  est aussi reprise par les paliers. La résolution de l'équation [3.3.6] donne alors pour la composante  $y$  :

$$M_{yi} = I_{byi} \ddot{\alpha}_i - F_i \cdot \frac{L_A}{L_B} [z_i \cdot \cos \alpha_i - (R - x_i) \cdot \sin \alpha_i] - r_{G_b} \cdot m_b \cdot g \cdot \cos \alpha_i \quad [3.3.7]$$

où le couple  $M_{y_i}$  est défini positif selon la règle du tire-bouchon (voir figure 3.3.5), et avec :

$I_{by_i}$	moment d'inertie du bras i autour du point $A_i$
$x_i, z_i$	coordonnées de la nacelle exprimées dans le repère $R_i$
$r_{Gb}$	distance entre le point $A_i$ et le centre de gravité du bras i
$m_b$	masse totale du bras, y compris la masse des avant-bras rapportée
$g$	accélération terrestre

### 3.3.5. Estimation du temps de calcul

Afin d'utiliser le modèle dynamique développé ci-dessus dans l'algorithme de commande, il est bon de faire une estimation du temps nécessaire à son calcul. Le nombre d'opérations élémentaires ainsi que le temps de calcul sur un Transputer T800 à 20 MHz sont reportés dans le tableau 3.3.6. On remarque ainsi qu'une utilisation du modèle en temps réel est aisée.

Opérations	Nombre
+	27
-	28
*	77
/	1
Sin <sup>1</sup>	3
Cos	3
Temps de calcul (T800)	$\approx 750 \text{ [\mu s]}$

*Tableau 3.3.6 : Estimation du nombre d'opérations pour le calcul du modèle dynamique*

---

<sup>1</sup> Le transputer incorporant une unité arithmétique, le calcul des fonctions trigonométriques ne pose pas de problème particulier

### 3.4. Modèle dynamique "découplé" pour le robot DELTA

Le modèle dynamique du robot DELTA, développé dans le paragraphe précédent, est une fonction de l'accélération angulaire de chacun des bras, mais aussi de l'accélération de la nacelle. Avec une telle formulation, la méthode de réglage par découplage non-linéaire, exposée dans le chapitre 6 n'est pas applicable. En effet, la réalisation de cet algorithme implique un modèle dynamique calculable en temps réel et ayant la forme suivante :

$$\mathbf{M} = \mathbf{A}(\alpha) \cdot \ddot{\alpha} + \mathbf{C}(\alpha, \dot{\alpha}) \quad [3.4.1]$$

avec :	$\mathbf{M}$ :	couples sur chacun des moteurs
	$\mathbf{A}(\alpha)$ :	matrice d'inertie du robot
	$\ddot{\alpha}$ :	accélération des articulations
	$\mathbf{C}(\alpha, \dot{\alpha})$ :	effets centrifuges, de Coriolis et de gravité.

Pour obtenir un modèle dynamique sous cette forme, il est nécessaire d'établir la relation entre l'accélération  $\ddot{\alpha}$  des articulations et l'accélération  $\dot{\mathbf{X}}$  de la nacelle. Celle-ci peut s'obtenir en dérivant deux fois le modèle géométrique direct du robot. En raison de l'interdépendance des variables articulaires, une solution analytique à ce problème n'est pas triviale et amènerait à une grande complexité de calcul compromettant alors sa résolution en temps réel.

Dans ce paragraphe, une méthode numérique, basée sur le modèle simplifié établi précédemment, est présentée pour obtenir la forme adéquate de la dynamique du robot DELTA. L'algorithme développé tient compte principalement des contraintes temps réel nécessaires à son utilisation dans la boucle de réglage.

#### 3.4.1. Etablissement du modèle découplé

L'équation [3.3.7] exprimant le couple à appliquer à chacune des articulations pour obtenir un mouvement spécifique de la nacelle sert de base à l'établissement du modèle découplé. Le lecteur intéressé trouvera le détail du développement en annexe. Ici, seules les étapes principales de la méthode sont décrites.

En mettant l'équation [3.3.7] sous forme vectorielle, on obtient :

$$\mathbf{M} = \mathbf{I}_b \cdot \ddot{\boldsymbol{\alpha}} - \mathbf{K} \cdot \mathbf{F} - \mathbf{G}_b \quad [3.4.2]$$

On a aussi, tiré des équations [3.3.2] et [3.3.5] :

$$\mathbf{F} = \mathbf{K}_m \cdot \dot{\mathbf{X}} - \mathbf{G}_n \quad [3.4.3]$$

En remplaçant [3.4.3] dans [3.4.2], on obtient :

$$\mathbf{M} = \mathbf{I}_b \cdot \ddot{\boldsymbol{\alpha}} - \mathbf{K} \cdot \mathbf{K}_m \cdot \dot{\mathbf{X}} + \mathbf{K} \cdot \mathbf{G}_n - \mathbf{G}_b \quad [3.4.4]$$

La relation entre  $\dot{\mathbf{X}}$  et  $\ddot{\boldsymbol{\alpha}}$  peut être obtenue à partir du modèle géométrique direct. En effet, à partir de la relation  $\mathbf{X} = f(\boldsymbol{\alpha})$ , on obtient :

$$\dot{\mathbf{X}} = \begin{pmatrix} \frac{\partial f_x}{\partial \alpha_1} & \frac{\partial f_x}{\partial \alpha_2} & \frac{\partial f_x}{\partial \alpha_3} \\ \frac{\partial f_y}{\partial \alpha_1} & \frac{\partial f_y}{\partial \alpha_2} & \frac{\partial f_y}{\partial \alpha_3} \\ \frac{\partial f_z}{\partial \alpha_1} & \frac{\partial f_z}{\partial \alpha_2} & \frac{\partial f_z}{\partial \alpha_3} \end{pmatrix} \dot{\boldsymbol{\alpha}} = \mathbf{J}(\boldsymbol{\alpha}) \cdot \dot{\boldsymbol{\alpha}} \quad [3.4.5]$$

où  $\mathbf{J}$  désigne la matrice jacobienne du robot qui dépend de sa position  $\boldsymbol{\alpha}$ . Une dérivée supplémentaire de cette équation donne alors :

$$\ddot{\mathbf{X}} = \mathbf{J}(\boldsymbol{\alpha}) \cdot \ddot{\boldsymbol{\alpha}} + \mathbf{D}(\boldsymbol{\alpha}, \dot{\boldsymbol{\alpha}}) \quad [3.4.6]$$

où  $\mathbf{D}(\boldsymbol{\alpha}, \dot{\boldsymbol{\alpha}})$  contient tous les termes de dérivée partielle du deuxième ordre et est fonction des position et vitesse du robot.

En remplaçant [3.4.6] dans [3.4.4] on obtient alors :

$$\mathbf{M} = (\mathbf{I}_b - \mathbf{K} \cdot \mathbf{K}_m \cdot \mathbf{J}) \ddot{\boldsymbol{\alpha}} - \mathbf{K} \cdot \mathbf{K}_m \cdot \mathbf{D} + \mathbf{K} \cdot \mathbf{G}_n - \mathbf{G}_b \quad [3.4.7]$$

On obtient ainsi le modèle dynamique du robot DELTA sous la forme [3.4.1] désirée. On peut alors identifier la matrice d'inertie du robot :

$$\mathbf{A} = \mathbf{I}_b - \mathbf{K} \cdot \mathbf{K}_m \cdot \mathbf{J} \quad [3.4.8]$$

et la contribution des termes centrifuges, de Coriolis et de gravité :

$$\mathbf{C} = -\mathbf{K} \cdot \mathbf{K}_m \cdot \mathbf{D} + \mathbf{K} \cdot \mathbf{G}_n - \mathbf{G}_b \quad [3.4.9].$$

Ainsi, la matrice d'inertie  $\mathbf{A}$  du robot DELTA peut être calculée relativement simplement. La forme analytique de la matrice jacobienne  $\mathbf{J}$  est trop complexe pour être raisonnablement calculée en temps réel. Pour cette raison, un calcul numérique des dérivées partielles est effectué. Ceci nécessite alors de calculer 4 fois le modèle géométrique direct du robot (voir annexe B).

Si la matrice  $\mathbf{A}$  est relativement simple à calculer, il n'en va pas de même pour l'équation [3.4.9] dont le terme  $\mathbf{D}(\alpha, \dot{\alpha})$  est très complexe. Dans ce cas, il paraît plus intéressant de calculer  $\mathbf{C}$  en combinant [3.3.7], [3.4.1] et [3.4.8]. On obtient alors :

$$\mathbf{C} = \mathbf{M} - \mathbf{A} \cdot \ddot{\alpha} \quad [3.4.10].$$

Ainsi, à l'aide des équations [3.3.7], [3.4.8] et [3.4.10], il est possible de calculer un modèle dynamique temps réel découplé pour le robot DELTA.

### 3.4.2. Etude de la matrice d'inertie

L'étude de la matrice d'inertie d'un robot est intéressante à plus d'un titre. Les termes diagonaux permettent premièrement de se rendre compte du taux de variation de l'inertie propre vue par chacun des moteurs lorsque l'on se déplace dans le volume de travail. Les termes non-diagonaux permettent quant à eux de mesurer l'importance du couplage entre les différents axes. Dans ce paragraphe, ces deux paramètres seront analysés pour le robot DELTA en se basant sur l'équation [3.4.8] développée dans le paragraphe précédent.

Par soucis de simplification et pour obtenir une représentation agréable de cette matrice d'inertie, le volume de travail du robot DELTA considéré est un cylindre d'axe vertical, selon la définition de [Clavel 91]. L'étude des termes de la matrice est alors réalisée sur des tranches horizontales du cylindre situées à différentes hauteurs  $z$ . En considérant 3 plans situés respectivement au haut, au milieu et au bas du volume de travail, il est possible de se faire une idée raisonnable des variations maximales des paramètres inertIELS. Le lecteur intéressé trouvera une représentation graphique de cette matrice d'inertie dans l'annexe B. On peut y faire les constatations suivantes :

### 1. Inerties propres (termes diagonaux de la matrice)

- Les 3 termes diagonaux possèdent une symétrie ternaire ( $120^\circ$ ) reprenant ainsi la symétrie du robot DELTA.
- Les variations d'inertie vue par chacun des moteurs vont environ du simple au double (voir tableau 3.4.4).
- Les plus importantes variations ont lieu dans le plan situé au haut du volume de travail du robot, et plus le plan est bas dans ce volume, moins les variations sont importantes.

	valeur max. [kg cm <sup>2</sup> ]	valeur min. [kg cm <sup>2</sup> ]	variation [%]
$z = -230$ [mm]	95.0491	53.4542	78
$z = -300$ [mm]	94.4436	68.9104	37
$z = -350$ [mm]	91.4982	77.7602	17.7

*Tableau 3.4.4 : Inerties maximales et minimales du robot DELTA*

### 2. Inerties de couplage (termes non-diagonaux de la matrice)

- La matrice d'inertie est symétrique. Ceci confirme une règle bien connue de la physique des systèmes mécaniques (action = réaction).
- Le couplage est relativement important. Il représente jusqu'à 70 % de l'inertie minimale du robot.
- Les couplages les plus importants ont lieu dans la zone la plus basse du volume de travail et ils sont moins importants dans le haut de ce volume (voir tableau 3.4.5).
- Les couplages sont surtout importants sur les bords du volume de travail. Au centre, le terme de couplage est plus faible et passe même par zéro pour les points particuliers où les avant-bras sont orthogonaux deux à deux. A ce moment-là, la projection de la force créée par l'un des bras sur l'autre est nulle.

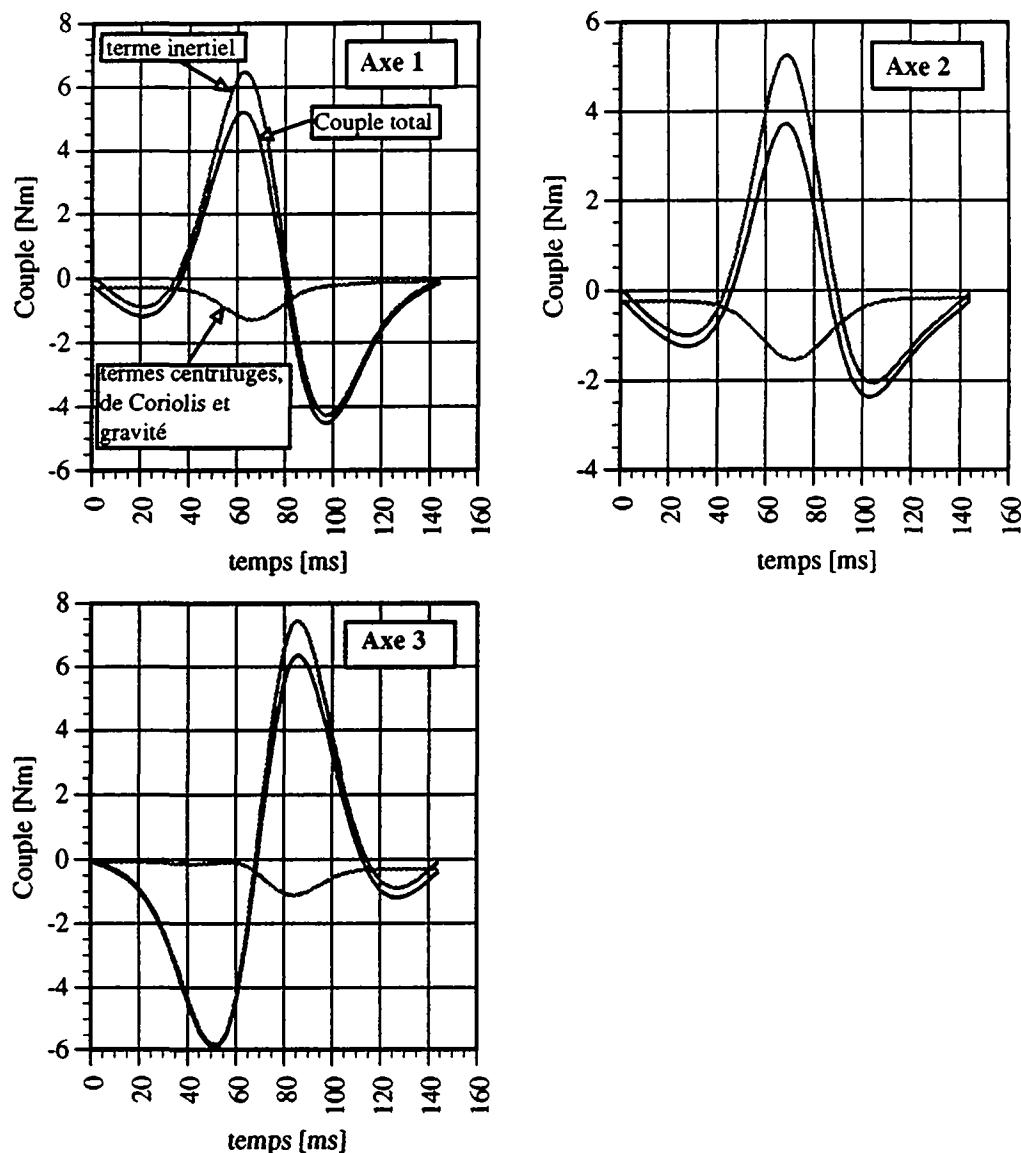
	valeur max [kg cm <sup>2</sup> ]	valeur min. [kg cm <sup>2</sup> ]
$z = -230$ [mm]	10.03	-30.39
$z = -300$ [mm]	14.99	-34.87
$z = -350$ [mm]	19.66	-37.41

*Tableau 3.4.5: Inerties maximales et minimales de couplage du robot DELTA*

### 3.4.3. Influence des termes centrifuges, de Coriolis et de gravité

Si le terme de gravité ne dépend que de la position du robot, les forces centrifuges et de Coriolis dépendent également des vitesses de chacun des bras. La plupart des commandes de robot sur le marché à l'heure actuelle ne tiennent pas compte de ces forces, ce qui est justifié pour des robots se déplaçant lentement. Cependant, lorsque les vitesses deviennent importantes, ces termes peuvent être aussi élevés que les termes inertIELS. Il convient alors d'en tenir compte. Dans ce paragraphe, l'influence de ces forces est montrée dans le cas d'une trajectoire typique du robot DELTA en se basant sur l'équation [3.4.10].

La figure 3.4.6 montre l'allure des différentes composantes de couple pour une trajectoire demi-ellipse dans un plan vertical à 45° par rapport au référentiel de base R du robot. Le terme inertIELL représente l'influence de l'inertie de chacun des bras. Il est le produit du vecteur accélération angulaire  $\ddot{\alpha}$  des bras et de la matrice d'inertie A. Le couple total est celui calculé par l'équation [3.3.7] et la différence de ces deux composantes (voir équation [3.4.10]) représente l'influence des termes dépendant des vitesses (centrifuge et Coriolis) et du terme de gravité.



*Figure 3.4.6: Influence des termes de gravité, centrifugés et de Coriolis*

Cette figure montre bien l'importance des termes de vitesse dans le cas du robot rapide DELTA, puisqu'ils interviennent pour plus de 30 % du couple total à fournir. Dans un cas (axe 2), on obtient même une contribution de près de 50 %. Il est intéressant de remarquer ici que les termes de vitesse ont tendance à diminuer le couple qui serait donné par les inerties propres et les couplages. Ceci se comprend aisément en considérant le mouvement de rotation des avant-bras, générateur de la force centrifuge au niveau de la nacelle.

### 3.5. Validation et limites du modèle dynamique

Pour pouvoir utiliser le modèle dynamique dans l'algorithme de réglage, il est tout d'abord nécessaire de le valider afin d'estimer dans quelle mesure il se rapproche du comportement réel du robot. Pour cela, deux tests ont été effectués. Premièrement, on compare le modèle simplifié développé ici avec le modèle "complet" selon [Clavel 91] pour estimer quelle est l'erreur engendrée par les hypothèses sur les avant-bras. Ensuite, un essai en boucle ouverte sur le robot réel est réalisé pour voir comment le robot suit la trajectoire désirée. Les limites du modèle sont ensuite décrites afin de se rendre compte dans quelles conditions il est acceptable de l'utiliser.

#### 3.5.1. Comparaison avec le modèle de [Clavel 91]

La comparaison a lieu ici pour la même trajectoire demi-ellipse que celle utilisée dans le paragraphe précédent. On a choisi plusieurs types de répartition des masses des avant-bras afin d'identifier le meilleur rapport  $r$ . Les résultats sont consignés dans le tableau 3.5.1.

A l'analyse de ce tableau, on remarque que les erreurs entre le modèle simplifié et le modèle "complet" sont inférieures à 5 % si l'on choisit  $r = 0.7$ . Ceci s'explique simplement si l'on songe que le moment d'inertie  $I$  d'une barre rigide (de longueur  $L$  et de masse  $m$ ) autour d'une de ses extrémités est donné par  $I = \frac{1}{3} \cdot mL^2$ . Cela revient à placer 1/3 de la masse à l'extrémité mobile de la barre (dans notre cas la nacelle) et à placer les 2/3 restant à l'endroit fixe (coude). Au vu de ces résultats, les hypothèses concernant les avant-bras (paragraphe 3.3.3) ne sont pas trop restrictives et ne détériorent pas le modèle.

		couple max. [Nm]	erreur [%]	couple min. [Nm]	erreur [%]
$r = 0$	axe 1	6.18026	20.99	-5.81593	27.37
	axe 2	4.09019	14.44	-2.75355	17.77
	axe 3	7.56381	21.08	-7.86734	29.5
$r = 0.3$	axe 1	5.76327	12.82	-5.25949	15.18
	axe 2	3.93075	9.98	-2.59316	10.91
	axe 3	7.04374	12.76	-7.03986	15.88
$r = 0.5$	axe 1	5.48538	7.38	-4.88853	7.06
	axe 2	3.82445	7.0	-2.48624	6.34
	axe 3	6.69703	7.21	-6.48821	6.8
$r = 0.7$	axe 1	5.21408	2.07	-4.51757	-1.06
	axe 2	3.71906	4.05	-2.37931	1.76
	axe 3	6.35032	1.66	-5.93655	-2.27
$r = 1.0$	axe 1	4.80712	-5.89	-3.96113	-13.24
	axe 2	3.57082	-0.09	-2.21892	-5.09
	axe 3	5.83026	-6.66	-5.10908	-15.9

Tableau 3.5.1: Comparaison du modèle simplifié et du modèle "complet" de Clavel

### 3.5.2. Essai en boucle ouverte

Un essai en boucle ouverte est très intéressant, car il permet d'estimer sur le système réel la conformité du modèle. Les figures 3.5.2 et 3.5.3 montrent respectivement le schéma-bloc de l'expérience et les résultats obtenus.

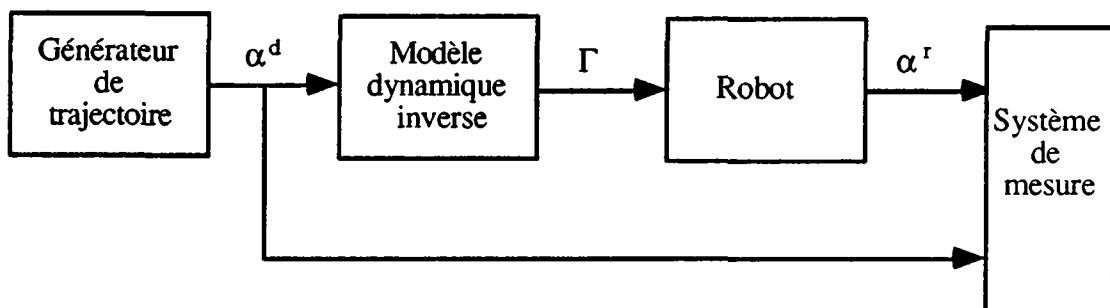


Figure 3.5.2 : Schéma-bloc du système de validation du modèle dynamique

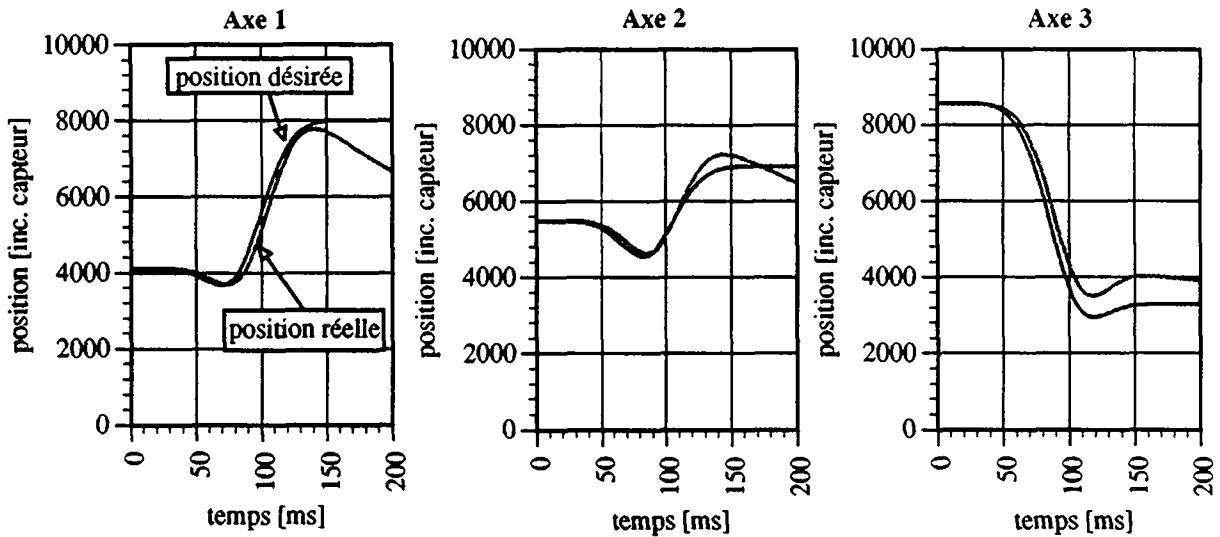


Figure 3.5.3 : Essai en boucle ouverte

On remarque que le modèle développé est relativement proche de la réalité et qu'il pourra ainsi réduire grandement la tâche du régulateur. On voit cependant que la position diverge à la fin de la trajectoire. Ceci est dû au couple réluctant du moteur qui a tendance à "tirer" chacun des axes dans une position stable différente de la position désirée. Ce phénomène n'étant pas modélisé ici, il ne peut être compensé.

### 3.5.3. Limites du modèle

Comme pour tout modèle, il est bon ici de discuter des limites d'utilisation du modèle développé. On distinguera deux types de limites; celles dépendantes des hypothèses de départ, et celles dues à la méthode de résolution adoptée.

#### Limites par les hypothèses de départ

On en citera principalement deux :

- Comme l'inertie des avant-bras a été négligée et que leurs masses ont été réparties à chacune de leurs extrémités, le modèle est quelque peu faussé. Il a été vu précédemment que cela n'introduisait pas des erreurs trop importantes, mais ceci à condition que ces masses soient petites vis-à-vis des autres masses de la structure. L'utilisation de fibres de carbone pour les avant-bras aide donc à l'obtention d'un meilleur modèle.

- Dans le modèle développé, le frottement est négligé. Ceci ne joue pas un rôle très important lors de l'utilisation d'un entraînement direct comme c'est le cas ici. Cependant il faudra se méfier de l'exactitude du modèle si un système de réduction est introduit entre le moteur et le bras, celui-ci amenant généralement un frottement assez important.

#### Limites dues à la méthode de résolution

Le modèle dynamique établi est basé sur une méthode numérique pour le calcul des dérivées (calcul des accélérations à partir des positions, calcul des dérivées partielles pour l'obtention de la matrice d'inertie, ...). Si cette manière de faire permet de gagner un temps de calcul précieux, elle peut cependant amener à des problèmes de bruit numérique dus aux limites de la représentation des nombres réels sur les ordinateurs. Il faudra donc veiller, avant toute utilisation du modèle que les paramètres choisis pour le calcul des dérivées n'introduisent pas de bruit numérique. Il faudra notamment agir avec prudence lorsque les accélérations sont très faibles et d'autre part choisir un pas de calcul des dérivées adapté.

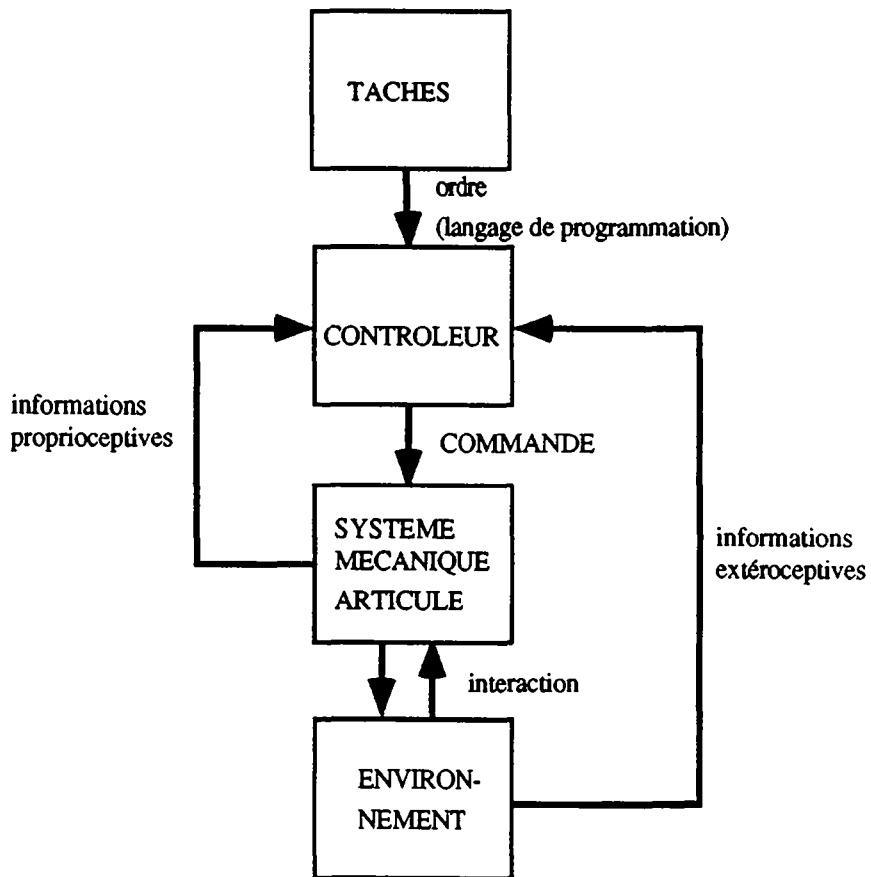
### 3.6. Conclusion

Dans ce chapitre, un nouveau modèle dynamique du robot DELTA a été présenté. Il est basé sur la méthode de Newton-Euler avec des hypothèses de départ permettant de réduire le nombre d'opérations sans introduire trop d'erreurs. Un calcul numérique de la jacobienne permet ensuite de transformer le modèle pour isoler la matrice d'inertie du robot. Les faibles temps de calcul de ces modèles permettent leur utilisation en temps réel dans la commande et seront exploités dans le chapitre 6.

## 4. Structure de la commande de robot

### 4.1. Introduction

Toute structure mécanique de robot, pour pouvoir effectuer des mouvements, est nécessairement liée à un système de commande. Pour éclaircir le rôle de celui-ci, il peut être intéressant de situer le contexte dans lequel il évolue. Comme le montre la figure 4.1.1, la structure générale d'un robot en fonctionnement se base essentiellement sur quatre ensembles interactifs :



*Figure 4.1.1 : Structure générale d'un robot*

### 1) Structure mécanique articulée

La structure mécanique est toujours la partie la plus apparente d'un robot. Elle est généralement constituée d'un ensemble de corps solides maintenus entre-eux par des liaisons à n degrés de liberté et est dotée d'actionneurs permettant son mouvement.

### 2) Environnement

C'est l'univers dans lequel est immergé la machine. L'environnement se réduit souvent à l'espace atteignable par le robot, c'est-à-dire à son volume de travail.

### 3) Les tâches

On peut définir une tâche comme étant la différence entre deux états de l'environnement. Les tâches désirées sont décrites dans un langage approprié, interprété par le contrôleur de robot.

### 4) Le contrôleur

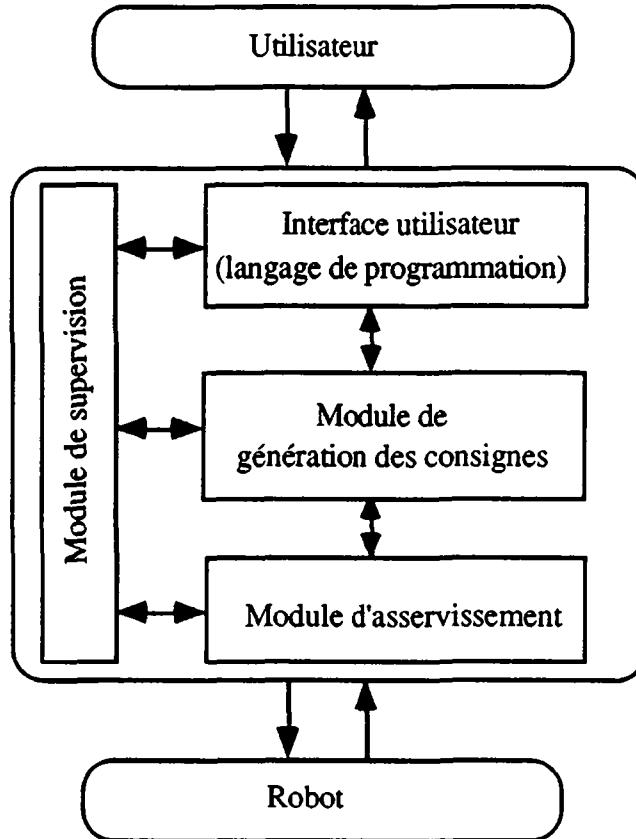
Le contrôleur est l'organe qui génère les commandes pour les actionneurs, afin que le comportement du robot soit aussi proche que possible de la tâche qui lui a été spécifiée. La commande est élaborée à partir d'informations *a priori* (la tâche) et *a posteriori* (état du robot et de l'environnement).

Les tâches demandées au robot sont généralement nombreuses et impliquent souvent des comportements antagonistes de la commande. Il est alors important qu'une optimisation soit entreprise en fonction des spécificités de l'application envisagée.

Dans ce chapitre, la structure interne du contrôleur de robot sera brièvement décrite avant d'analyser le cas du robot DELTA où une structure sera choisie en accord avec sa géométrie particulière et les tâches envisagées.

## 4.2. Structure interne du contrôleur de robot

La structure interne d'un contrôleur de robot peut être représentée sous la forme de la figure 4.2.1.



*Figure 4.2.1 : Structure interne d'un contrôleur de robot*

**Le module de supervision** affecte toute la commande. Il se charge de la gestion générale du système et de sa sécurité. Il doit détecter toute anomalie de fonctionnement et décider de la procédure à enclencher, selon son importance.

**L'interface utilisateur** s'occupe du dialogue entre la commande et le programmeur. Généralement, on trouve à ce niveau un langage de programmation qui comporte des ordres de mouvement du robot et de gestion des périphériques.

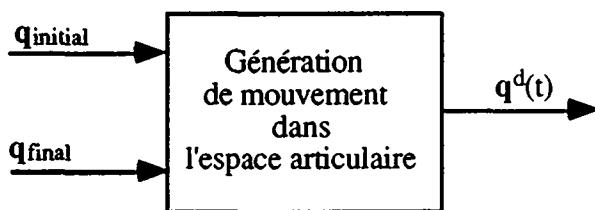
**Le module de génération des consignes** calcule la consigne à fournir au module d'asservissement pour réaliser la primitive désirée. Il fait appel à un interpolateur, et parfois au modèle géométrique inverse du robot. Dans la littérature,

[Dombre & Khalil 88] on distingue les générations de consignes dans l'espace articulaire et dans l'espace opérationnel. Avantages et inconvénients de ces deux méthodes seront cités aux paragraphes 4.2.1 et 4.2.2 respectivement.

**Le module d'asservissement** calcule la commande à envoyer aux actionneurs pour suivre la consigne désirée. Suivant les performances visées, ce module peut faire appel au modèle dynamique du robot. Différentes méthodes d'asservissement seront décrites au chapitre 6.

#### 4.2.1. Génération des consignes dans l'espace articulaire

Avec cette méthode, le calcul des consignes  $q^d(t)$  est réalisé pour chaque axe indépendamment selon une loi de mouvement (interpolation) définie entre les points de départ  $q_{initial}$  et d'arrivée  $q_{final}$ . La figure 4.2.2. illustre cette méthode.



*Figure 4.2.2 : génération des consignes dans l'espace articulaire*

Ses avantages sont :

- le mouvement est minimal sur chaque articulation,
- les contraintes de vitesses et couples maximaux sont connues avec précision puisqu'elles correspondent aux limites physiques des actionneurs,

cependant,

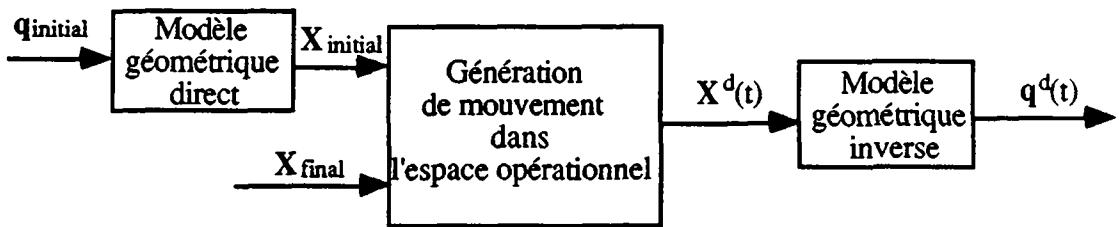
- la géométrie de la trajectoire dans l'espace opérationnel ne peut être imposée, et l'organe terminal se déplace de façon imprévisible, mais répétitive,
- il y a risque de collision dans un environnement encombré.

Ce type de mouvement est donc approprié pour réaliser des déplacements rapides dans un espace dégagé. On peut noter ici que les robots cartésiens ont l'avantage d'avoir une relation linéaire entre l'espace opérationnel et l'espace articulaire, ce qui

permet d'utiliser ce type de schéma en considérant directement les variables opérationnelles.

#### 4.2.2. Génération des consignes dans l'espace opérationnel

Si l'on veut garantir un mouvement bien défini de l'organe terminal du robot, il est nécessaire de spécifier sa trajectoire directement dans l'espace opérationnel à partir de la position actuelle  $q_{initial}$  du robot et de la position à atteindre  $X_{final}$ . La loi de commande engendrée  $X^d(t)$  doit ensuite être transformée en consignes articulaires  $q^d(t)$  par le modèle géométrique du robot. La figure 4.2.3 illustre cette méthode.



*Figure 4.2.3 : génération des consignes dans l'espace opérationnel*

L'avantage de cette méthode est qu'elle permet de contrôler la géométrie de la trajectoire. Par contre :

- elle implique le calcul du modèle géométrique inverse de chaque point de la trajectoire,
- elle est quelques fois irréalisable (passage hors du volume de travail, passage par une position singulière, ...),
- si l'on ne connaît pas le modèle dynamique du manipulateur, les limites en vitesse et couple des articulations ne peuvent pas facilement être interprétées dans l'espace opérationnel. On exprime alors généralement ces limites par des valeurs moyennes satisfaites quelle que soit la configuration du robot. De cette manière on impose au robot de travailler au-dessous de ses moyens réels, ce qui peut être un grand inconvénient.

L'utilisation de l'une ou l'autre de ces deux méthodes dépend beaucoup de l'application considérée, et les deux approches sont généralement présentes dans le contrôleur de robot.

### 4.3. Cas du robot DELTA

Comme mentionné au chapitre 2, le robot DELTA est principalement destiné à la manipulation de pièces légères à cadence élevée. Pour ce genre de travaux, le robot devra principalement réaliser des opérations de prise et dépose d'objet après déplacement, comme déjà indiqué par la figure 2.6.3.

Si l'on observe la structure géométrique du DELTA, on s'aperçoit que, si un seul des trois bras est en mouvement, la nacelle suit une trajectoire en arc de cercle dans un plan vertical (voir § 5.4.1). Cette constatation compromet sérieusement le choix d'une commande avec génération des consignes dans l'espace articulaire, car le risque de collision avec la table de travail est alors beaucoup trop élevé. Le choix de points de passage judicieux pourrait résoudre ce problème, mais alors la programmation du robot n'est plus aisée. En effet, si l'on donne une vitesse de passage nulle à ces points, il en résulte une perte de temps inutile. Par contre, il n'est pas simple de fixer une vitesse de passage non-nulle de façon à avoir le trajet optimal.

Au vu de ces constatations, une commande avec génération des consignes dans l'espace opérationnel s'impose pour le robot DELTA. La programmation du robot en est facilitée et le risque de collisions est éliminé. Des trajectoires de prise et dépose respectant le cahier des charges et proches de l'optimal doivent alors être recherchées.

### 4.4. Conclusion

Dans ce chapitre, la structure d'une commande de robot a été établie. Pour la génération des consignes, le choix entre une commande dans l'espace articulaire ou une commande dans l'espace opérationnel a été discuté. Bien que la génération de consigne dans l'espace articulaire amène aux mouvements minimaux, la génération de consigne dans l'espace opérationnel est préférée dans le cas du robot DELTA. En effet, en raison de la géométrie particulière du robot, une trajectoire libre de la nacelle est beaucoup trop dangereuse car le risque de collision est élevé. En choisissant des trajectoires de prise et dépose de façon judicieuse, les temps de parcours pourront certainement être proches de l'optimal. C'est ce qui va être étudié dans le prochain chapitre.

## 5. Génération des consignes

### 5.1. Introduction

La réalisation d'un robot rapide et précis impose un effort particulier à tous les étages de la commande. Dans ce cadre-là, le choix d'une consigne adaptée est primordiale. La recherche du mouvement à temps minimal a conduit les roboticiens à l'utilisation de lois de mouvement de type parabolique (bang-bang) dans l'espace articulaire. Cette méthode n'est cependant pas directement applicable au robot DELTA; la nécessité de la génération des consignes dans l'espace opérationnel ayant été montrée au chapitre 4. Il s'agira donc, dans le cadre de ce chapitre de trouver la trajectoire "optimale" dans cet espace pour le robot DELTA. La plupart des approches rencontrées dans la littérature définissent "l'optimalité" comme le temps minimal de parcours en fonction des contraintes d'accélération et de vitesse des actionneurs utilisés [Paul 78, Tondu 84]. Dans cet esprit, seule la cinématique de la trajectoire est prise en compte, l'aspect dynamique étant négligé. Dans le cas de robots effectuant des mouvements lents, cette approche est tout à fait réaliste, mais elle ne l'est plus lors de mouvements rapides. En effet, les contraintes dynamiques sont alors telles que le mouvement optimal au point de vue cinématique ne pourra pas forcément être suivi par le robot et son système d'asservissement.

La recherche d'une commande optimale peut être réalisée à l'aide de méthodes d'optimisation sous contraintes telle la programmation dynamique. La complexité d'une telle approche et les temps de calcul très importants qu'elle nécessite compromettent sérieusement la recherche de cet optimum. Par conséquent, une approche heuristique, basée sur des trajectoires particulières est préférée.

Ce chapitre propose la recherche de mouvements adaptés aux déplacements rapides du robot DELTA pour des tâches classiques de prise et dépose d'objets. Après un bref rappel des notions de trajectoire et de mouvement, les critères déterminants pour le choix des trajectoires seront exposés. Diverses lois de mouvement et trajectoires sont ensuite proposées pour des tâches rapides de prise et dépose.

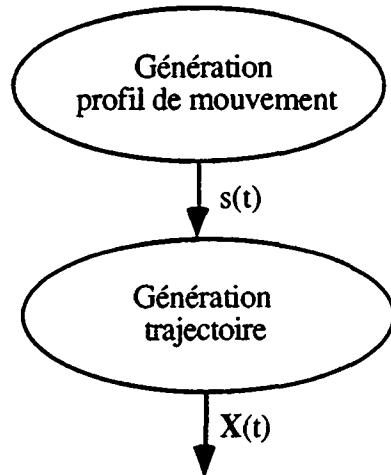
## 5.2. Notions de trajectoire et de mouvement pour robots rapides

### 5.2.1. Paramétrisation de la trajectoire

Le choix d'une trajectoire particulière dans l'espace opérationnel n'est pas suffisant pour effectuer un mouvement. Une loi horaire (ou loi de mouvement) sur cette dernière doit être donnée afin d'obtenir un déplacement. La génération du mouvement peut alors être décomposée en 2 étapes distinctes :

- 1) Choix de la géométrie de la trajectoire  $f(s)$  paramétrée en fonction de son abscisse curviligne  $s$
- 2) Imposition d'une loi de mouvement  $s(t)$  le long de cette trajectoire

Le schéma de la figure 5.2.1. illustre cette décomposition.



*Figure 5.2.1 : 2 étapes pour la génération de mouvement*

Finalement, le déplacement d'un mobile dans l'espace peut être décrit par une suite temporelle de positions. On notera alors  $X(t) = f(s(t))$  la position du mobile à l'instant  $t$ .

### 5.2.2. Contraintes

Le choix d'une trajectoire ne peut généralement pas être arbitraire et doit respecter certaines contraintes liées à l'environnement d'une part et aux limitations physiques du robot d'autre part.

#### *Contraintes dues à l'environnement*

L'environnement d'un robot est souvent encombré d'objets avec lesquels il faut éviter une collision. Ceci implique principalement des contraintes au niveau de la géométrie de la trajectoire. La prise en considération des obstacles est alors souvent réalisée par l'introduction de points de passage à vitesse nulle ou non-nulle entre le point de départ et d'arrivée.

#### *Contraintes dues au robot*

Généralement, les contraintes considérées dans la littérature sont de deux types :

- limitations géométriques du robot (volume de travail)
- cinématique : limitation en vitesse et accélération de chaque moteur

En fait, considérer la limite d'accélération d'un moteur est une approximation de la vraie limite qu'est le couple. Dans le cas de robots utilisant des moteurs à gros couple et faible inertie, cette approximation n'est plus valable et il devient nécessaire de reconsidérer les vraies limites.

### 5.2.3. Prise en compte des contraintes "dynamiques"

#### *Limitation du couple*

La variable de sortie d'un moteur commandé en courant est un couple (ou force) sur le rotor. Les limitations en accélération généralement décrites dans la littérature se déduisent de ce couple à condition de considérer la masse en mouvement comme constante. Il a été montré dans les précédents chapitres que ceci n'était pas le cas lorsque l'on considère un robot à entraînement direct. Dès lors, il semble plus judicieux de considérer directement le couple comme facteur limitatif plutôt que l'accélération. Pour un mouvement donné, la vérification du non-dépassement de cette limite est immédiat si le modèle dynamique du robot a été développé.

### *Limitation de la variation de couple*

Comme nous l'avons vu au paragraphe 2.5.4., l'utilisation de moteurs à entraînement direct augmente considérablement la constante de temps électrique du moteur en raison de l'augmentation de l'inductance  $L$  de son bobinage. La relation entre le courant et la tension dans une inductance est donnée par :

$$U = L \frac{di}{dt}$$

Ainsi, si la tension d'alimentation  $U$  est limitée, la variation du courant va diminuer lorsque  $L$  augmente. En considérant le couple proportionnel au courant, on obtient une limitation pour la variation de couple donnée par :

$$\delta\Gamma_{\max} = \left. \frac{\partial\Gamma}{\partial t} \right|_{\max} \approx \frac{\Gamma_p}{t_r} \quad [5.2.1]$$

avec :  $\delta\Gamma_{\max}$  : variation maximale de couple admissible  
 $\Gamma_p$  : couple de pointe du moteur  
 $t_r$  : temps de montée du courant dans le moteur

On peut vérifier aisément si la trajectoire choisie satisfait ce critère en vérifiant les maxima de la dérivée du couple obtenu par le modèle dynamique inverse du robot.

### **5.2.4. Recherche d'un optimum**

En considérant les critères dynamiques introduits ci-dessus, le choix d'une trajectoire optimale devient très complexe. Pour cette raison, la méthode utilisée dans ce chapitre considérera d'abord les limites cinématiques pour le choix des trajectoires et des lois de mouvement. Le calcul du modèle dynamique et de sa dérivée permettra ensuite de vérifier si les trajectoires choisies satisfont les contraintes dynamiques. Cette approche relativement simple permet tout de même par itérations successives, en jouant sur les différents paramètres définissant les trajectoires, de tendre vers une solution acceptable.

## 5.3. Lois de mouvement

Comme il a été montré précédemment, la génération du mouvement peut être décomposée en 2 étapes distinctes. Ce paragraphe décrit, tout d'abord, la génération des lois de mouvement qui seront ensuite "projétées" sur la trajectoire.

### 5.3.1. Préambule

Considérons un mobile devant parcourir une trajectoire quelconque de longueur totale  $d$ . A l'instant  $t=0$ , il se trouve au début de la trajectoire, soit  $s(0) = 0$ . A l'instant  $t = T_f$ , il doit avoir parcouru toute la trajectoire, et on a alors  $s(T_f) = d$ . La loi de mouvement a alors pour but de contrôler l'évolution temporelle de  $s$  satisfaisant les conditions de départ et d'arrivée, soit :

$$\left\{ \begin{array}{l} s(0) = 0 \\ s(T_f) = d \\ \dot{s}(0) = 0 \\ \dot{s}(T_f) = 0 \end{array} \right.$$

Toutes les formes imaginables sont possibles pour  $s(t)$ . Différentes approches sont décrites et comparées dans [Tondu 84] où il est démontré que la loi parabolique (bang-bang) est optimale en temps si l'on considère uniquement les contraintes cinématiques.

La prise en compte des limites de vitesse du moteur est généralement réalisée dans l'espace articulaire en insérant une phase à vitesse constante entre les phases d'accélération et de décélération. Cette approche est presque obligatoire lors de l'utilisation d'un réducteur entre le moteur et le bras du robot, car les limites de vitesse du moteur sont très vite atteintes. Dans le cas de l'entraînement direct, le moteur ne réalise jamais plus d'un tour et la faible distance de mouvement empêche d'atteindre la vitesse limite du moteur<sup>1</sup>. Insérer une phase à vitesse constante reviendrait alors à diminuer les cadences possibles. De ce fait, cette étude ne considérera que des lois de mouvement sans limites de vitesse.

D'autre part, la plupart des lois utilisées dans la pratique se basent sur des profils symétriques, c'est-à-dire où la valeur maximale de décélération est la même que celle de l'accélération. Si cette manière de faire simplifie quelque peu les calculs à effectuer,

---

<sup>1</sup>Il faut noter ici que ceci n'est pas vrai dans le cas des moteurs couples à réluctance variable. Toutefois, pour obtenir un robot rapide, il conviendra de ne pas choisir de tels moteurs sous peine de limiter les cadences.

elle peut cependant pénaliser la rapidité de mouvement du robot. En effet, si l'on considère que la phase d'arrivée sur un point doit être très "douce" et que la phase de départ peut être plus "violente", il devient alors intéressant de réaliser des lois de mouvement asymétriques. Afin de mesurer l'importance d'une telle asymétrie, les lois de mouvement seront développées en conséquence.

Les contraintes concernant la variation maximale de couple peuvent être assimilées, pour une première approximation au "jerk" (troisième dérivée de s) que l'on cherchera à limiter.

### 5.3.2. Loi parabolique ou loi "bang-bang"

La loi de mouvement parabolique est la plus utilisée en robotique, parce qu'elle est simple et de plus optimale en temps pour la génération de trajectoire dans l'espace articulaire. Elle est essentiellement constituée de phases d'accélération et de décélération constantes, comme le montre la figure 5.3.1.

Son équation a la forme suivante :

$$s(t) = \begin{cases} \frac{1}{2}a_a t^2 & \text{pour } 0 \leq t \leq T_a \\ s(T_a) + v_m(t-T_a) - \frac{1}{2}a_d(t-T_a)^2 & \text{pour } T_a < t \leq T_f \end{cases} \quad [5.3.1]$$

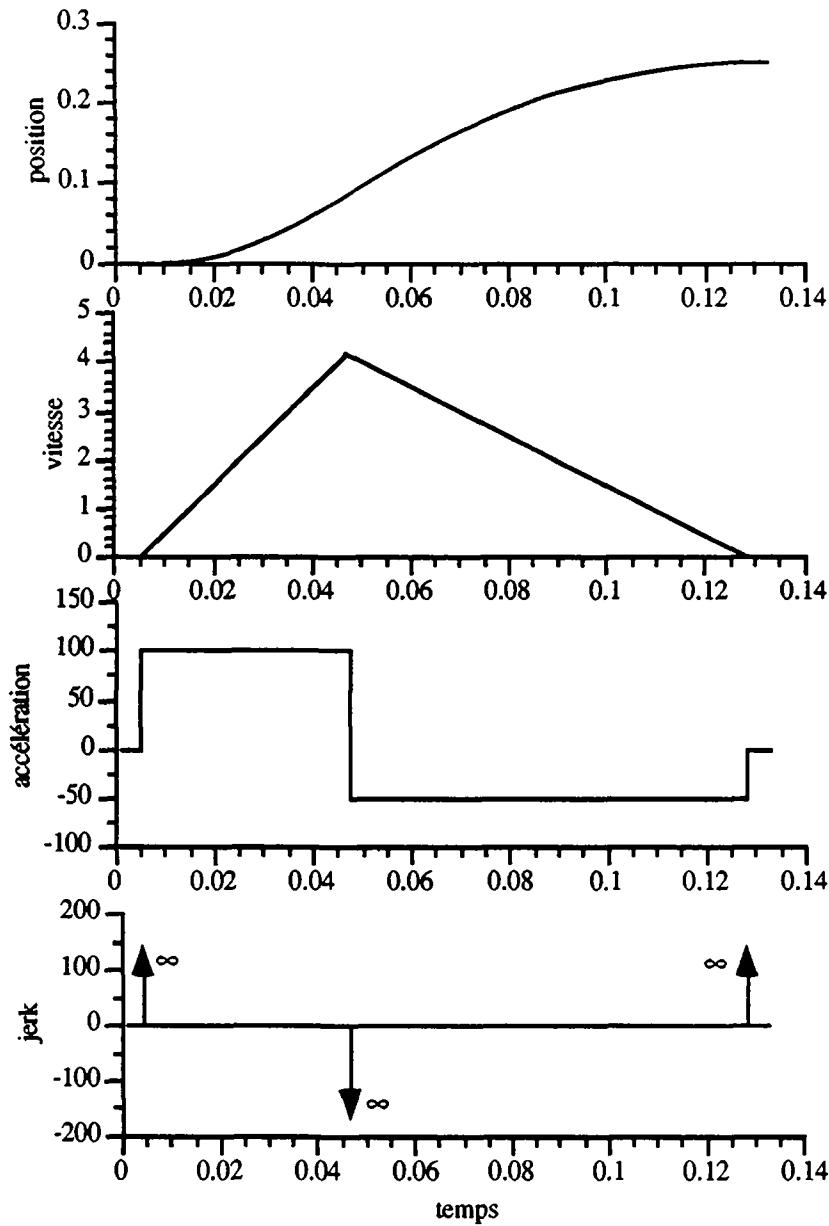
avec :  $a_a$ ,  $a_d$  : respectivement valeur maximale de l'accélération et de la décélération

$s(T_a) = \frac{1}{2} a_a T_a^2$ , distance parcourue à l'instant  $T_a$

$v_m = a_a T_a$ , vitesse maximale atteinte

$T_a$ ,  $T_f$ : respectivement temps d'accélération et temps total de parcours (leur calcul sera montré au paragraphe 5.3.5)

Si cette loi de mouvement a l'avantage d'obtenir un temps de parcours minimal, elle a cependant le désavantage d'un "jerk" infini à plusieurs endroits critiques (notamment à l'arrivée).



*Figure 5.3.1 : Loi de mouvement parabolique asymétrique*

### 5.3.3. Loi cycloïde

Pour chercher à limiter le jerk, il peut paraître intéressant de choisir une loi de mouvement plus "douce". Dans ce but, la loi de mouvement cycloïde a l'avantage de présenter une accélération continue de forme sinusoïdale. En raison de sa formulation, il serait cependant plus judicieux de parler de loi "sinus sur rampe", comme dans la dénomination anglo-saxonne (sine on ramp). Malgré ses avantages, elle est assez peu rencontrée dans la littérature, ceci certainement parce qu'elle nécessite le calcul d'un sinus. A l'heure actuelle, avec des unités arithmétiques à virgule flottante disponibles pour la plupart des microprocesseurs courants, ceci n'est plus un problème majeur. La figure 5.3.2. illustre ce mouvement.

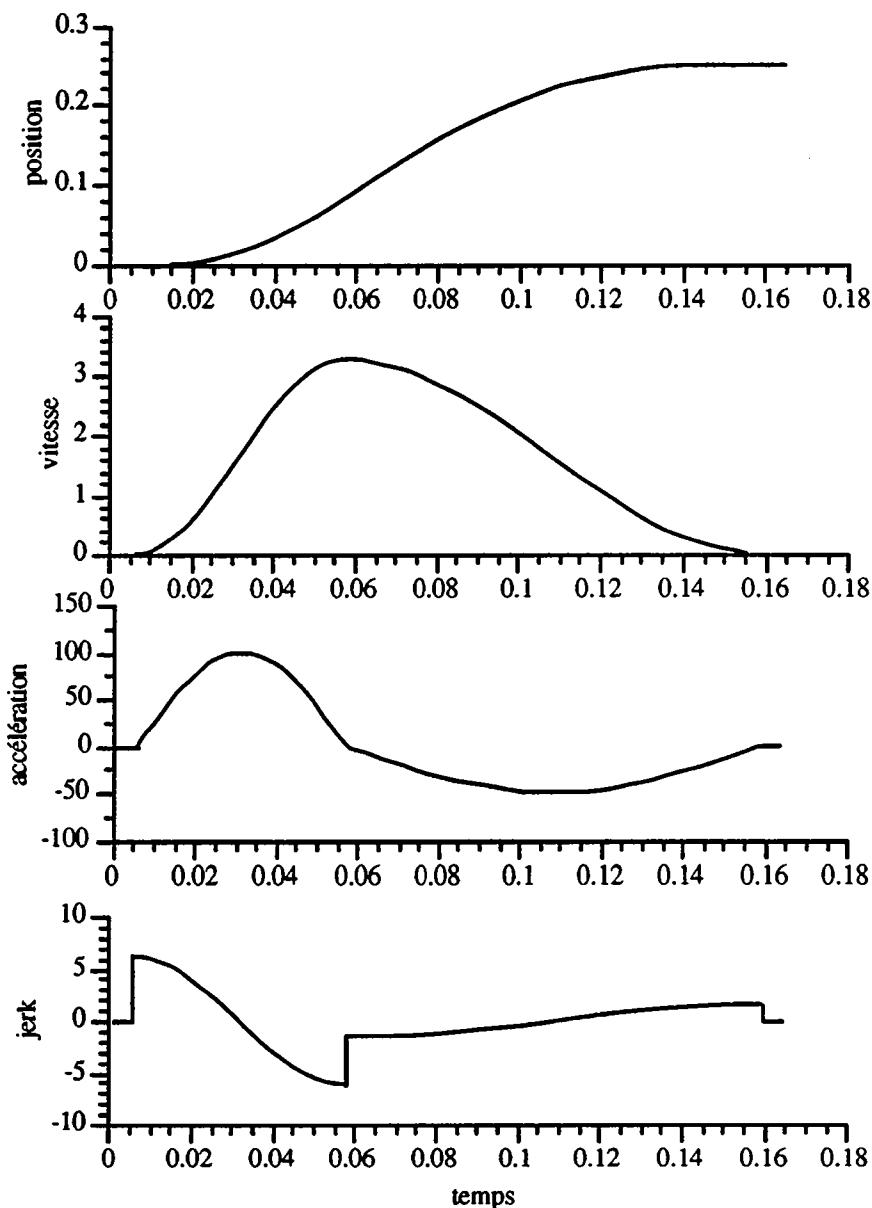


Figure 5.3.2 : Loi de mouvement cycloïde asymétrique

Son équation est la suivante :

$$s(t) = \begin{cases} a_a \frac{T_a}{\pi} t - a_a \frac{T_a^2}{\pi^2} \sin(\frac{\pi}{T_a} t) & \text{pour } 0 \leq t \leq T_a \\ a_d \frac{T_d}{\pi} (t - T_a) + \frac{a_a}{\pi} T_a^2 - a_d \frac{T_d^2}{\pi^2} \sin(\frac{\pi}{T_d} t + \varphi) & \text{pour } T_a < t \leq T_f \end{cases} \quad [5.3.2]$$

avec :  $T_d = T_f - T_a$  temps de freinage

$\varphi = \pi \cdot (1 - \frac{T_a}{T_d})$  paramètre pour le freinage

$T_a, T_f$  : respectivement temps d'accélération et temps total de parcours (cf. § 4.3.5)

#### 5.3.4. Autres lois

D'autres types de lois de mouvement sont utilisés dans la pratique. Parmi les plus connus, on citera :

- **Loi polynômiale** : basée sur un polynôme de degré 5 si l'on veut assurer la continuité en accélération [Yoshikawa 90, Dombre & Khalil 88].
- **Loi cosinus** : basée sur une accélération en forme de cosinus qui présente le désavantage d'une discontinuité d'accélération au départ et à l'arrivée [Tondu 84].
- **Loi "jerk sinusoïdal"** : qui comme son nom l'indique possède un jerk sans discontinuité de type sinus [Dayer 88].
- **Loi à jerk constant** : basée sur la limitation du jerk à des valeurs maximales et minimales constantes.

Ces différentes lois ne seront pas traitées dans ce travail, car elles n'apportent pas d'avantages notoires par rapport à la loi cycloïde.

### 5.3.5. Calcul du temps de parcours

Le temps de parcours est calculé en fonction de la distance à parcourir, des accélérations et décélérations désirées, ainsi que du type de profil choisi. Il est assez commode pour ce calcul de se ramener toujours à une loi parabolique symétrique en introduisant des coefficients tenant compte du profil. Trois conditions sont à considérer pour cela (voir figure 5.3.3) :

$$(1) \quad \int_0^{T_a} \ddot{s}(t) dt = \int_0^{\frac{T_f}{2}} a_{eq} dt \quad [5.3.3]$$

$$(2) \quad \int_{T_a}^{T_f} \ddot{s}(t) dt = \int_{\frac{T_f}{2}}^{T_f} -a_{eq} dt \quad [5.3.4]$$

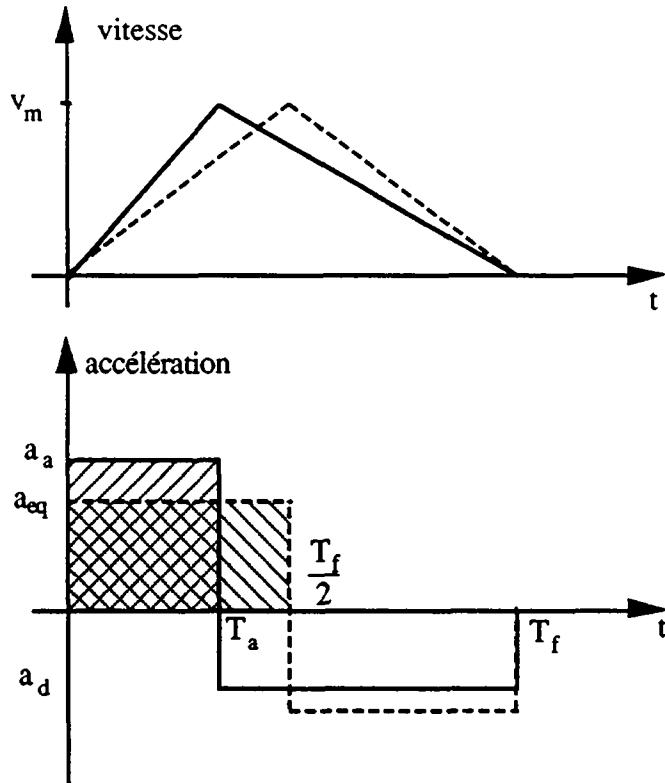
$$(3) \quad \int_0^{T_a} \ddot{s}(t) dt = - \int_{T_a}^{T_f} \ddot{s}(t) dt \quad [5.3.5]$$

où  $a_{eq}$  représente l'accélération d'un profil parabolique symétrique équivalent au profil considéré. Les deux premières équations garantissent que lorsque l'on écarte  $T_a$  du milieu ( $T_f/2$ ) les surfaces sous les courbes restent les mêmes, ou autrement dit que le profil équivalent symétrique et le profil réel aient la même vitesse maximale  $v_m$  (même énergie cinétique). La troisième condition garantit une vitesse d'arrivée nulle (pour une vitesse de départ nulle également).

En utilisant les équations [5.3.3] et [5.3.4] on peut calculer l'accélération équivalente pour les profils parabolique et cycloïde. On trouve :

$$\text{Profil parabolique : } a_{eq} = 2 \cdot \frac{a_a \cdot a_d}{a_a + a_d} \quad [5.3.6]$$

$$\text{Profil cycloïde : } a_{eq} = \frac{4}{\pi} \cdot \frac{a_a \cdot a_d}{a_a + a_d} \quad [5.3.7]$$



**Figure 5.3.3 : Profil parabolique symétrique équivalent à un profil parabolique asymétrique**

Le temps total de parcours peut être facilement calculé à l'aide de cette accélération équivalente et de la distance  $d$  à parcourir. On a :

$$T_f = 2 \sqrt{\frac{d}{a_{eq}}} \quad [5.3.8]$$

Le temps d'accélération est alors une proportion du temps final découlant de l'équation [5.3.5] :

$$T_a = \frac{a_d}{a_a + a_d} T_f \quad [5.3.9]$$

En remplaçant les valeurs des accélérations équivalentes dans la formule [5.3.8] on peut comparer les temps de parcours de différents profils symétriques ( $a = a_a = a_d$ ), comme le montre le tableau 5.3.4.

Type de mouvement	Expression de $T_f$	Ecart par rapport à la loi parabolique	Valeur maximale du jerk
parabolique	$2 \cdot \sqrt{\frac{d}{a}}$	-	$\infty$
cosinus	$\frac{\pi}{\sqrt{2}} \cdot \sqrt{\frac{d}{a}}$	+ 11 %	$\infty$
polynomiale (degré 5)	$\sqrt{\frac{10}{\sqrt{3}}} \cdot \sqrt{\frac{d}{a}}$	+ 20 %	$3\sqrt{3} \cdot \sqrt{\frac{\sqrt{3}}{10}} \cdot a \sqrt{\frac{a}{d}}$
cycloïde	$\sqrt{2\pi} \cdot \sqrt{\frac{d}{a}}$	+ 25 %	$\sqrt{2\pi} \cdot a \cdot \sqrt{\frac{a}{d}}$
jerk sinusoïdal	$2\sqrt{2} \cdot \sqrt{\frac{d}{a}}$	+ 41 %	$\frac{\pi}{2\sqrt{2}} a \cdot \sqrt{\frac{a}{d}}$

*Tableau 5.3.4 : Comparaison des temps de parcours et de la valeur maximale du jerk pour différents profils*

A l'analyse de ce tableau, on remarquera que la loi cycloïde est un bon compromis entre la douceur du mouvement et le temps de parcours.

## 5.4. Trajectoires

Après l'étude des lois de mouvement, il s'agit de définir des trajectoires compatibles avec la dynamique du robot. La recherche d'une trajectoire optimale en temps dans l'espace opérationnel du robot n'étant pas simple, l'approche envisagée ici se limitera à quelques cas particuliers de trajectoires de prise et dépose d'objets répondant au cahier des charges du paragraphe 2.6.2. Les lignes droites, demi-ellipses, et clothoïdes seront notamment décrites. Au préalable, la trajectoire "naturelle" du robot DELTA sera étudiée pour justifier la génération des trajectoires dans l'espace opérationnel.

### 5.4.1. Trajectoire "naturelle" du robot DELTA

Par trajectoire "naturelle", on entend la trajectoire réalisée par l'organe terminal du robot si l'on génère la consigne dans l'espace articulaire (loi parabolique sur chaque articulation). Du point de vue cinématique, il s'agit en fait du mouvement à temps minimal. La figure 5.4.1. montre l'allure de cette trajectoire dans le cas du robot DELTA pour des points de départ et d'arrivée situés à la même hauteur z.

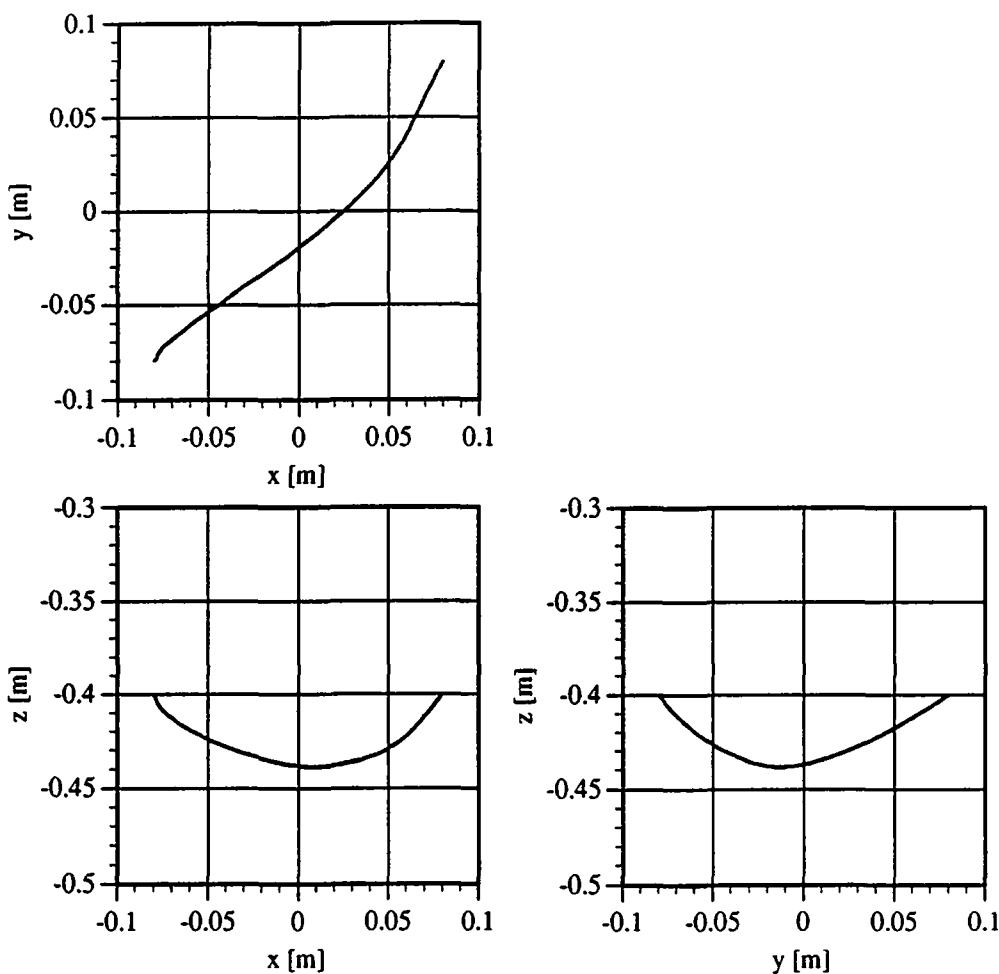


Figure 5.4.1 : Trajectoire naturelle du robot DELTA (projections sur les plans du repère de base)

Cette figure montre clairement que la trajectoire "naturelle" du robot DELTA est incurvée vers le bas, ce qui la rend difficilement utilisable en pratique en raison des risques de collision avec la table de travail. Ainsi, la génération de la trajectoire devra obligatoirement avoir lieu dans l'espace opérationnel.

### 5.4.2. Ligne droite

La trajectoire la plus simple est la ligne droite reliant deux points quelconques du volume de travail du robot. Elle n'est cependant pas toujours réalisable, notamment lorsque le volume de travail du robot est non convexe. Il conviendra donc de surveiller la trajectoire pour interdire de sortir du volume de travail. Dans le cas du robot DELTA, ce problème ne se pose pas, le volume de travail étant convexe.

La description mathématique de la droite en fonction de son abscisse curviligne est simple également. En posant  $X_d$  et  $X_f$  les points de départ et d'arrivée respectivement, on peut calculer la distance à parcourir :

$$d = |X_f - X_d| \quad [5.4.1]$$

Le vecteur directeur unitaire de la droite est donné par :

$$\mathbf{u} = \frac{\mathbf{X}_f - \mathbf{X}_d}{d} \quad [5.4.2]$$

La droite s'exprime alors, en fonction de son abscisse curviligne  $s$ , par :

$$\mathbf{f} = \mathbf{X}_d + s \cdot \mathbf{u} \quad [5.4.3]$$

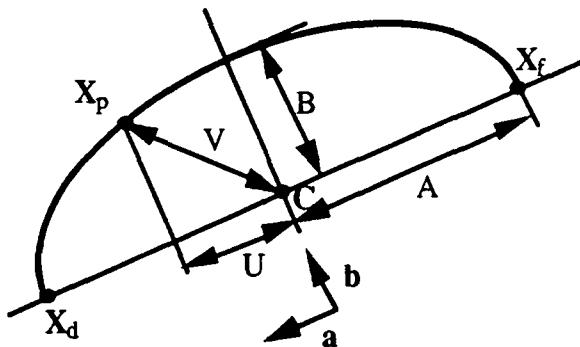
En y projetant une loi de mouvement  $s(t)$  on obtient finalement :

$$\mathbf{X}(t) = \mathbf{X}_d + s(t) \cdot \mathbf{u} \quad [5.4.4]$$

Si la ligne droite est intéressante pour sa simplicité, elle l'est aussi au point de vue dynamique, car elle n'introduit pas d'accélération centripète sur la nacelle, son rayon de courbure étant infini. Malheureusement, elle n'est pas utilisable pour les opérations de prise et dépose. Une combinaison de plusieurs segments de droite pourrait être envisagée, mais les contraintes dynamiques imposent des passages à vitesse nulle à chaque discontinuité de la trajectoire, ce qui ralentit naturellement le robot. Aussi, d'autres trajectoires doivent être envisagées.

### 5.4.3. Demi-ellipse

Une trajectoire particulièrement intéressante pour les opérations de palettisation est la demi-ellipse. Elle permet de moduler à volonté la distance entre les points de prise et la hauteur de transit. De plus, sa courbure ne présente pas de discontinuité, ce qui est favorable au point de vue de la dynamique.



*Figure 5.4.2 : Trajectoire demi-elliptique par 3 points*

On peut la calculer en fixant trois points. Soient  $X_d$  et  $X_f$  les points de départ et d'arrivée respectivement, et  $X_p$  un point de passage. On peut calculer le centre  $C$  de l'ellipse comme suit (figure 5.4.2) :

$$C = \frac{1}{2} (X_d + X_f) \quad [5.4.5]$$

La valeur du demi grand axe de l'ellipse vaut :

$$A = |X_d - C| \quad [5.4.6]$$

et sa direction est donnée par le vecteur normé :

$$\mathbf{a} = \frac{X_d - C}{A} \quad [5.4.7]$$

Soit  $U$  la valeur de la projection du vecteur  $CX_p$  sur  $\mathbf{a}$ ,

$$U = (X_p - C) \cdot \mathbf{a} \quad [5.4.8]$$

et  $V$  la norme de  $CX_p$

$$V = |X_p - C| \quad [5.4.9]$$

L'équation générale de l'ellipse permet alors de calculer la norme du demi petit axe.

On obtient :

$$B = \sqrt{\frac{(V^2 - U^2) \cdot A^2}{A^2 - U^2}} \quad [5.4.10]$$

Sa direction est donnée par le vecteur normé  $\mathbf{b}$  :

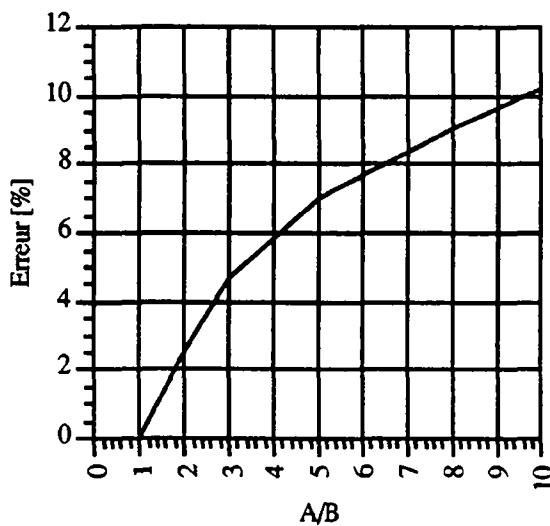
$$\mathbf{b} = \frac{\mathbf{X}_p - \mathbf{C} - \mathbf{U} \cdot \mathbf{a}}{\sqrt{V^2 - U^2}} \quad [5.4.11]$$

Le périmètre d'une ellipse n'étant pas défini analytiquement, il serait nécessaire d'avoir recours à une intégration numérique. Cependant, pour assurer un calcul en temps réel, il est préférable d'utiliser une formule approximative, ce qui aura pour effet de modifier quelque peu les accélérations effectives pour un temps de parcours donné (voir formule 5.3.8).

Selon [Schaum 79] le demi-périmètre de l'ellipse peut être approximé par :

$$d \approx \pi \cdot \sqrt{\frac{1}{2} (A^2 + B^2)} \quad [5.4.12]$$

Pour la raison évoquée ci-dessus, il n'est pas possible non plus de décrire simplement l'équation de l'ellipse en fonction de son abscisse curviligne. On peut cependant l'approximer en utilisant l'abscisse curviligne du cercle et en la projetant sur l'ellipse. Ceci modifie alors la loi de mouvement, ce qui est toutefois supportable si le grand axe et le petit axe de l'ellipse sont de grandeurs assez proches. La figure 5.4.3. illustre, en fonction du rapport entre A et B, l'erreur commise en utilisant l'approximation [5.4.12]. Comme on peut le constater, elle reste inférieure à 10 % dans tous les cas d'utilisation réalistes de l'ellipse ( $A/B < 10$ ).



**Figure 5.4.3 :** Erreur sur le périmètre de l'ellipse en fonction du rapport entre son grand axe et son petit axe

On définit alors :

$$\varphi = \frac{\pi \cdot s}{d} \quad \text{avec } s \in [0, d] \quad [5.4.13]$$

L'équation de l'ellipse devient :

$$f = C + A \cdot a \cdot \cos \varphi + B \cdot b \cdot \sin \varphi \quad [5.4.14]$$

En introduisant une loi de mouvement  $s(t)$ , on obtient :

$$X(t) = C + A \cdot a \cdot \cos\left(\frac{\pi \cdot s(t)}{d}\right) + B \cdot b \cdot \sin\left(\frac{\pi \cdot s(t)}{d}\right) \quad [5.4.15]$$

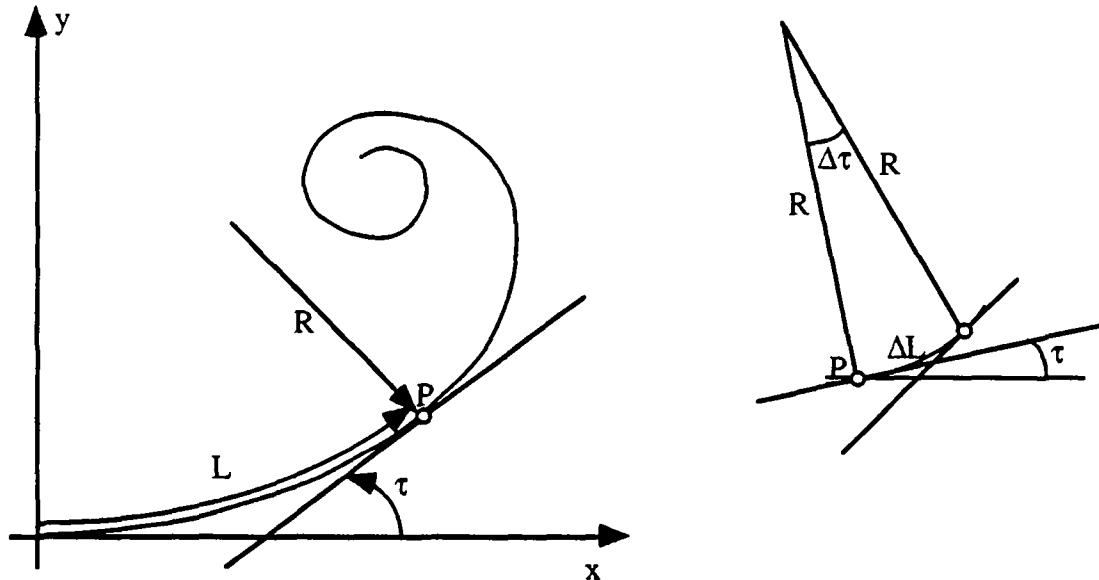
La trajectoire en demi-ellipse définie ci-dessus est intéressante au point de vue dynamique, mais elle ne permet malheureusement pas la prise et dépose d'objets situés à des hauteurs différentes en garantissant une arrivée et un départ à tangente verticale (nécessaire pour ne pas entrer en collision avec les posages). Une solution est d'ajouter des segments de droite de longueur variable à chaque extrémité de la demi-ellipse, mais ceci a le désavantage de créer des discontinuités de l'accélération centripète aux endroits de raccordement des deux courbes, ce qui va créer une discontinuité du couple. [Dayer 88] propose une solution avec des ellipses à tangentes verticales au départ et à l'arrivée, mais la trajectoire est strictement limitée aux applications de prise et dépose.

#### 5.4.4. Clothoïde

Pour éviter les discontinuités lors du raccord de deux droites de directions différentes, il peut être intéressant d'y ajouter une transition curviligne. [Paul 78] décrit une méthode utilisant des arcs de parabole pour relier les droites. Le désavantage de cette méthode est la discontinuité de l'accélération centripète. Pour éliminer cette discontinuité, il faudrait une trajectoire dont la courbure est une fonction continue de la distance parcourue. La clothoïde (figure 5.4.4) répond à ce critère, sa courbure étant proportionnelle à la distance parcourue. Il est à remarquer que cette courbe est utilisée abondamment pour le raccord des voies de chemin de fer. Son équation de base est la suivante :

$$L = A^2 \cdot K = A^2 \cdot \frac{1}{R} \quad [5.4.16]$$

- avec : L : distance parcourue
- K : courbure au point P
- R : rayon de courbure au point P
- A : facteur de proportionnalité



**Figure 5.4.4 : Paramètres de la clohoïde**

Sous forme différentielle, on obtient :

$$L = A^2 \cdot \frac{d\tau}{dL} \quad \text{ou} \quad L dL = A^2 d\tau \quad [5.4.17]$$

avec :  $\tau$  : angle entre la tangente au point P et la tangente au départ de la clohoïde

Après intégration, on obtient les 2 équations (équivalentes) :

$$L = \sqrt{2 A^2 \tau} \quad [5.4.18]$$

$$\tau = \frac{A^2}{2 R^2} \quad [5.4.19]$$

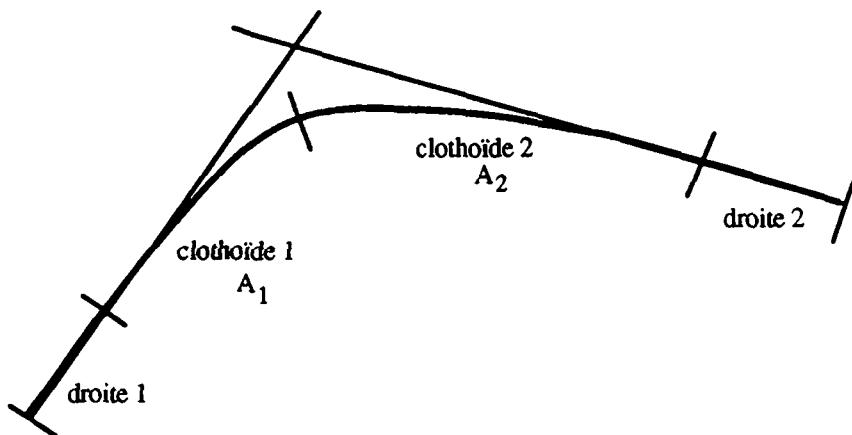
On peut alors trouver x et y de la clohoïde par :

$$x(s) = \frac{A}{\sqrt{2}} \int_0^s \frac{\cos \tau}{\sqrt{\tau}} d\tau \quad [5.4.20]$$

$$y(s) = \frac{A}{\sqrt{2}} \int_0^s \frac{\sin \tau}{\sqrt{\tau}} d\tau \quad [5.4.21]$$

On aboutit sur les intégrales de Fresnel qui n'ont pas de solution analytique, mais dont on peut utiliser la décomposition en série de Taylor (voir annexe C).

Pour le cas du raccordement de deux droites, on utilise la "clohoïde au sommet" qui relie les deux droites au moyen de 2 clohoïdes consécutives ayant le même rayon de courbure à l'endroit de raccordement. Ceci est illustré par la figure 5.4.5. On peut considérer deux cas de figure distincts suivant que les paramètres  $A_1$  et  $A_2$  des clohoïdes sont égaux ou non. On parlera alors respectivement de clohoïde au sommet symétrique et asymétrique. La description mathématique de cette courbe figure dans l'annexe C. Le lecteur intéressé trouvera plus d'informations sur la clohoïde dans [Baumann 91] ou [Ganguillet 82].



*Figure 5.4.5 : Cloïde au sommet*

Si la formulation de la cloïde est relativement complexe, les équations finales utilisées sont simples et permettent son calcul en temps réel. L'utilisation de lignes droites comme support des cloïdes permet à l'utilisateur une programmation aisée des trajectoires, puisqu'il a besoin d'introduire uniquement les points d'intersection des différentes droites considérées.

### 5.4.5. Comparaison des différentes trajectoires

Nous allons, dans ce paragraphe, comparer les performances dynamiques de chacune des trajectoires décrites. Les trajectoires utilisées doivent être conformes au cahier des charges du paragraphe 2.6.2.

Pour les simulations qui vont suivre, les points de passage choisis pour la trajectoire seront les suivants (valeurs en [mm]) :

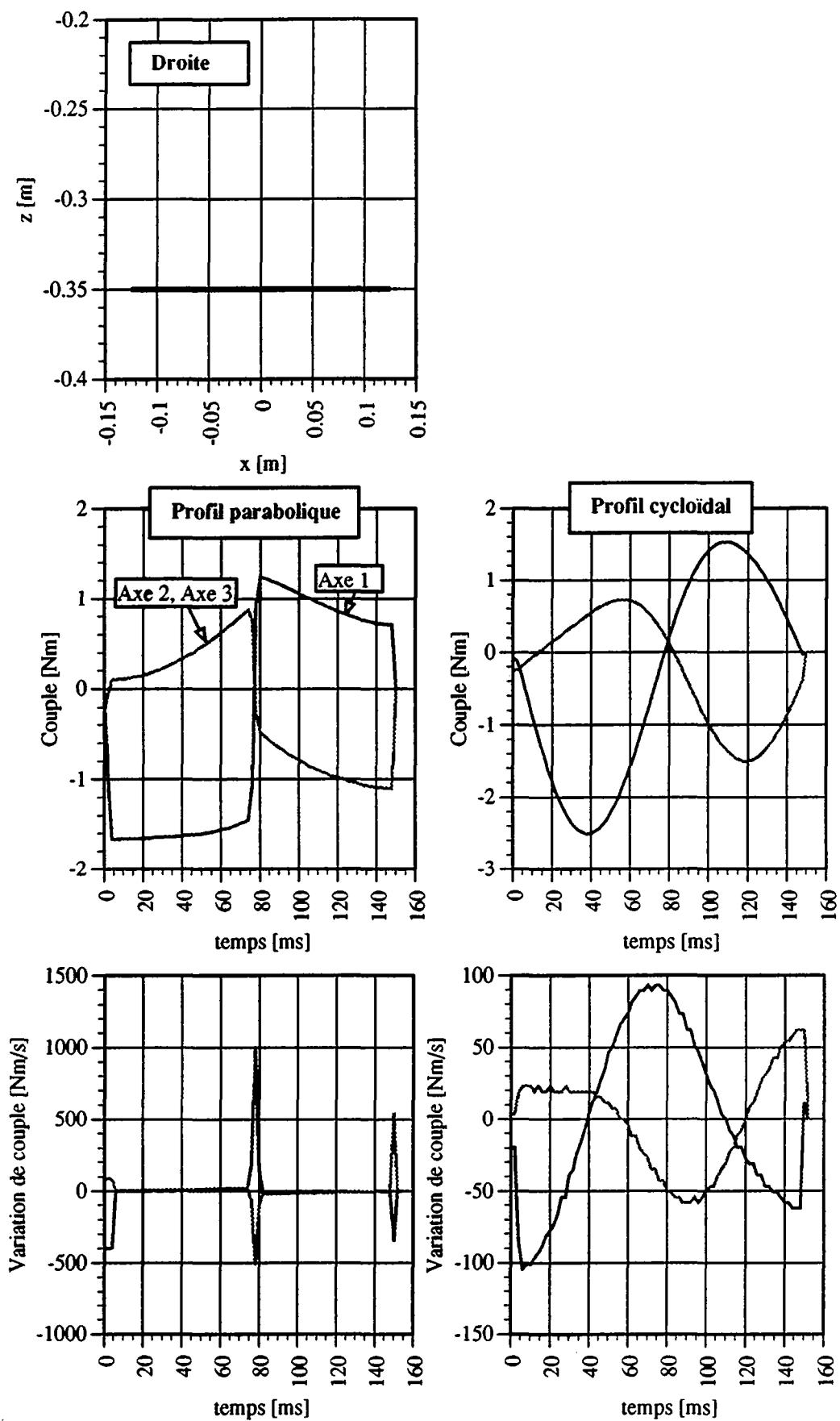
$$X_d = (125.0; 0.0; -350.0) \quad \text{point de départ}$$

$$X_f = (-125.0; 0.0; -350.0) \quad \text{point d'arrivée}$$

$$X_p = (0.0; 0.0; -310.0) \quad \text{point de passage (sauf pour la droite)}$$

Afin de permettre une comparaison aisée, on fixe le même temps de parcours pour chacune des trajectoires, soit  $T_f = 150$  [ms]. La comparaison des couples et des variations de couple de chacune des trajectoires nous renseigne alors sur leur "douceur" et donc sur les chances qu'elles auront de bien se comporter à travers la boucle de réglage.

Les figures 5.4.6 à 5.4.9 montrent l'allure du couple et de la variation de couple pour les différentes trajectoires et profils décrits dans ce chapitre. Les couples sont calculés selon l'équation [3.3.7] décrivant le modèle dynamique du robot DELTA. Les variations de couple sont obtenues par dérivée numérique du couple: Le tableau 5.4.10 résume les résultats obtenus.



*Figure 5.4.6 : Couple et variation de couple pour la ligne droite*

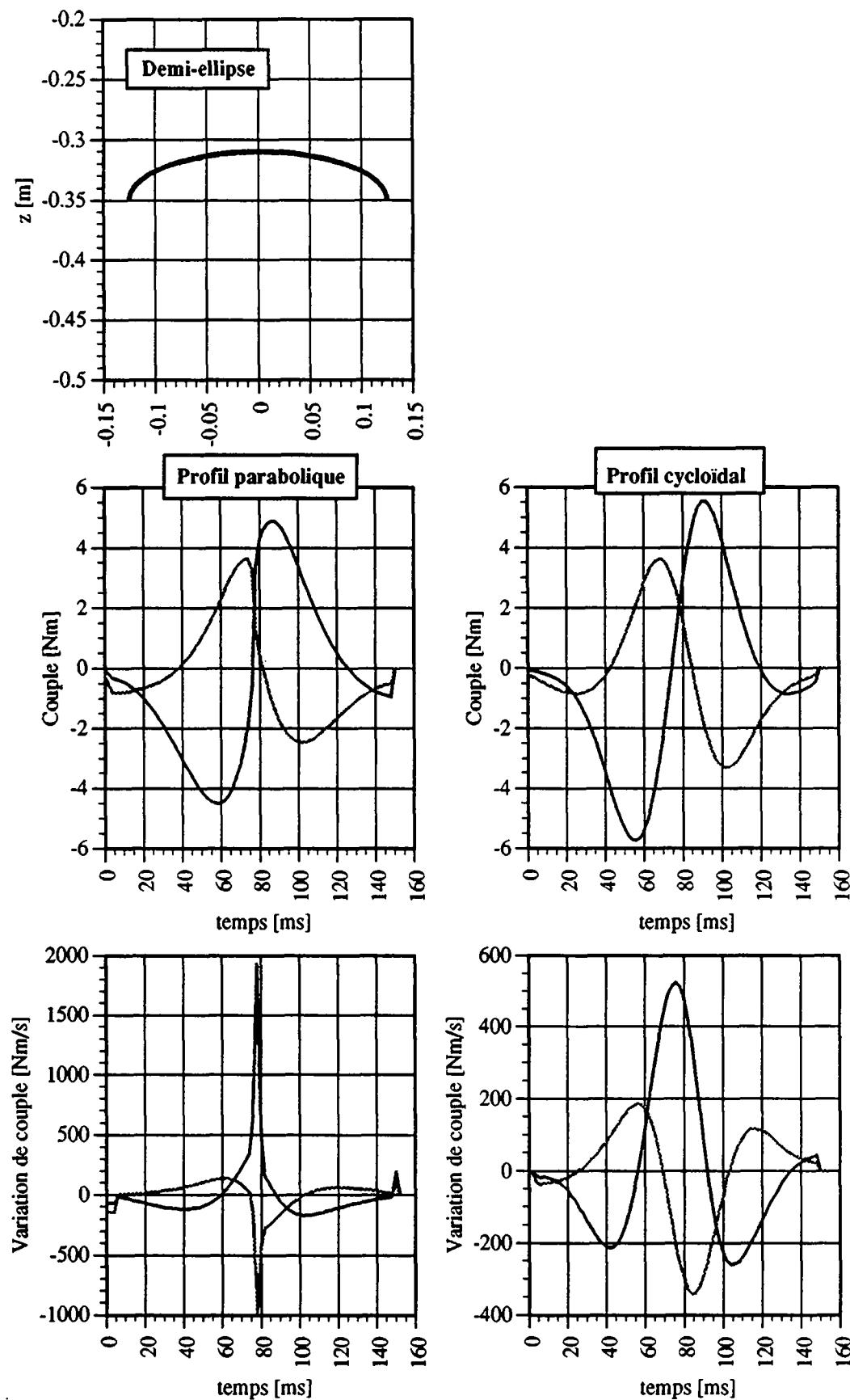
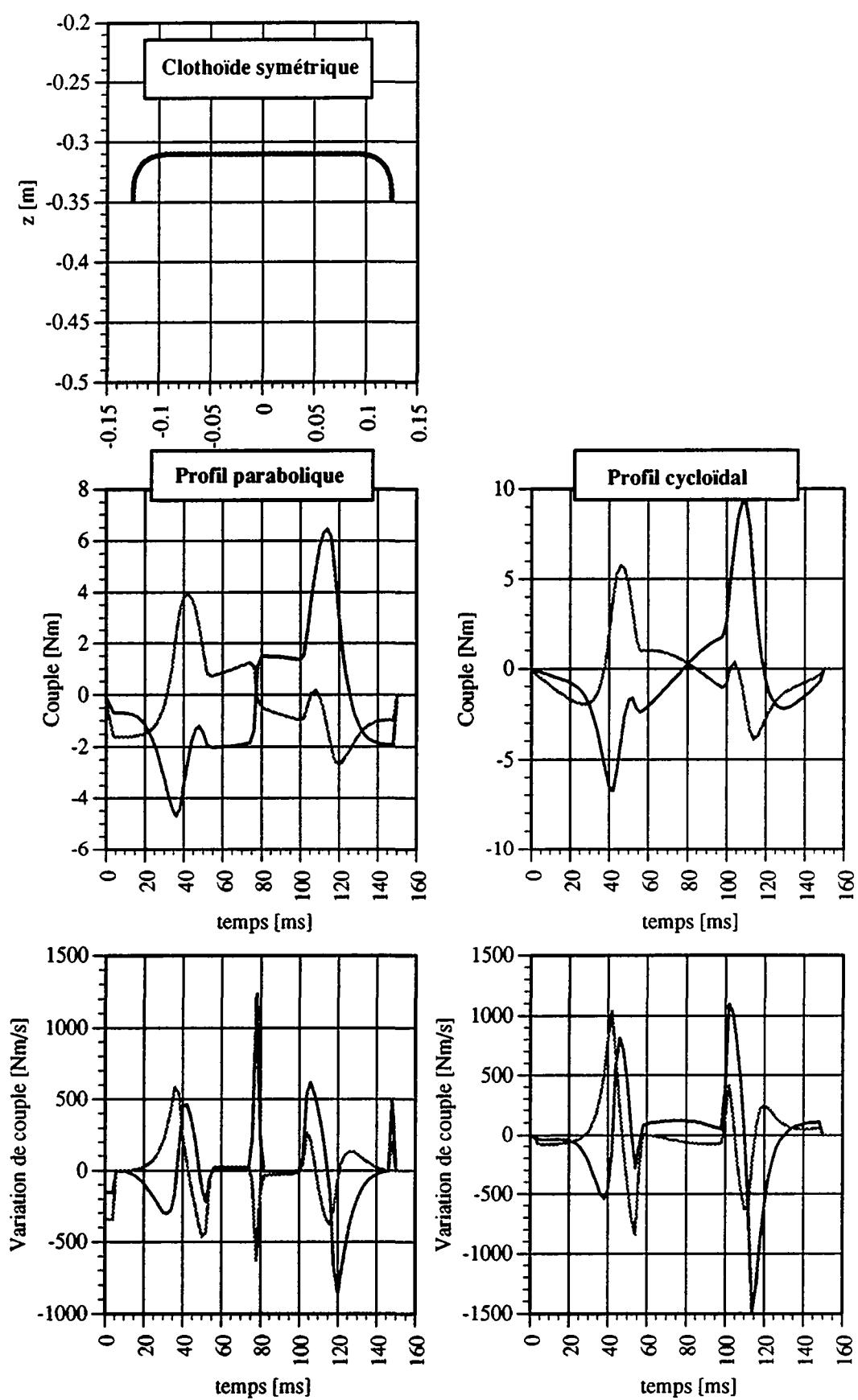


Figure 5.4.7 : Couple et variation de couple pour l'ellipse



*Figure 5.4.8 : Couple et variation de couple pour la clothoïde symétrique*

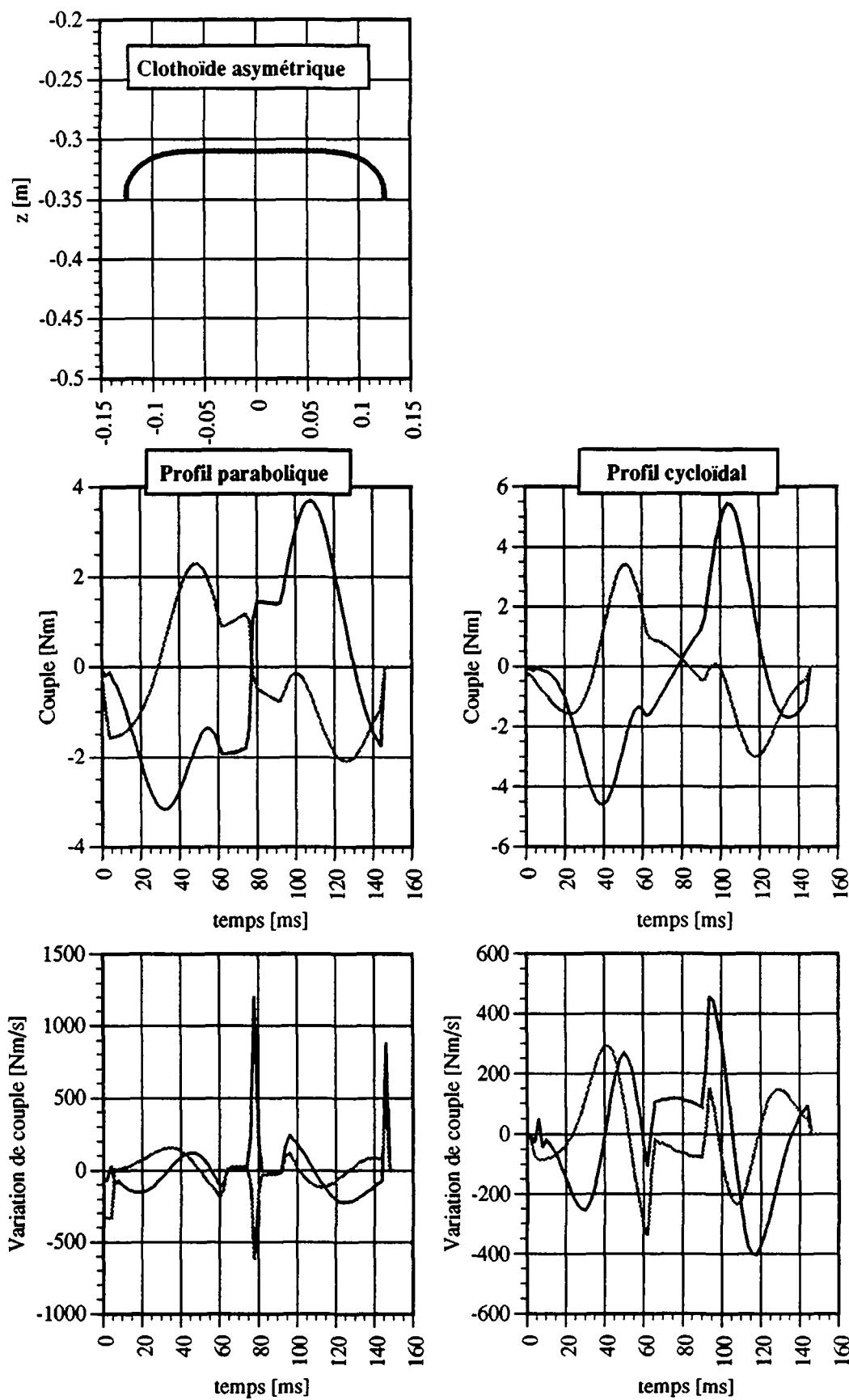


Figure 5.4.9 : Couple et variation de couple pour la clochôide asymétrique

Trajectoire	Profil	Couple maximal [Nm]	Variation maximale de Couple [Nm/s]
droite	parabolique	1.66	(1006)
	cycloïdal	2.51	94
demi-ellipse	parabolique	4.9	(1939)
	cycloïdal	5.74	525
Clothoïde sym.	parabolique	6.45	(1243)
	cycloïdal	9.25	1484
Clothoïde asym.	parabolique	3.7	(1208)
	cycloïdal	5.44	579

*Tableau 5.4.10 : Tableau récapitulatif du couple maximal et de la variation maximale de couple suivant le type de trajectoire et de profil*

Il faut remarquer ici que les valeurs maximales de variation de couple indiquées dans le tableau sont faussées pour le profil parabolique en raison de la dérivée numérique effectuée pour son calcul. Elles sont données ici à titre indicatif, mais sont bien plus élevées en réalité. De manière générale, on remarque assez clairement qu'un profil de mouvement sans discontinuité d'accélération est plus favorable qu'un profil parabolique en ce qui concerne la variation de couple. Le couple maximal à fournir est légèrement plus élevé, mais reste raisonnable, surtout avec les moteurs sans balais, qui permettent des couples de pointe bien supérieurs au couple nominal. Au niveau des trajectoires, la demi-ellipse et la clothoïde asymétrique montrent des résultats similaires. La clothoïde symétrique demande des couples de pointe plus élevés, principalement en raison de sa courbure assez importante et de sa longueur.

## 5.5. Conclusion

Le problème du choix d'une trajectoire optimale en temps dans l'espace opérationnel n'est pas trivial. Pour des raisons évidentes de simplification, la plupart des méthodes rencontrées dans la littérature tiennent compte uniquement de l'aspect cinématique de la trajectoire pour ce calcul. Dans le cas des robots rapides, cette approche n'est plus satisfaisante et il est nécessaire de tenir compte des contraintes dynamiques pour le choix de la trajectoire. Principalement deux types de contraintes sont à considérer :

- 1) limite de couple
- 2) limite de la variation du couple

Pour simplifier le problème, les trajectoires décrites ici ont été élaborées en considérant tout d'abord leur cinématique. La conformité aux contraintes dynamiques est ensuite observée. Par itérations successives, en corrigeant les paramètres cinématiques, il est alors possible de se rapprocher des limites dynamiques. Il ressort principalement de cette étude que toute discontinuité de l'accélération tangentielle (loi de mouvement) ou transverse (courbure de la trajectoire) doit être éliminée. L'approche heuristique adoptée ici a déjà permis de mettre en évidence quelques solutions à ce problème, mais une étude beaucoup plus détaillée serait encore nécessaire, notamment en considérant directement les contraintes dynamiques pour la recherche de trajectoires "optimales".



## 6. Asservissement

### 6.1. Introduction

Comme il a été montré au chapitre 4, le module d'asservissement se situe au niveau inférieur du contrôleur de robot. Il doit calculer la commande à envoyer aux actionneurs pour suivre au mieux la consigne désirée. Suivant le type d'actionneur utilisé (moteur c.c., moteur pas à pas, vérin pneumatique, vérin hydraulique, ...) ce module peut être complètement différent dans sa forme. Dans ce chapitre, seul les actionneurs dont on peut commander directement le couple seront traités, ce qui correspond à l'usage le plus fréquent en pratique. Afin que soient réduits les effets des perturbations extérieures et que la commande soit robuste vis-à-vis des erreurs de modélisation, celle-ci doit contenir une contre-réaction de vitesse et de position. La difficulté de la synthèse de la commande en boucle fermée provient du fait que le robot est un système multidimensionnel, non-linéaire, fortement couplé, et de plus variant dans le temps (modification de la charge transportée). Bien que l'asservissement des robots aie déjà fait l'objet de nombreux travaux, la difficulté du problème est de taille et beaucoup reste à faire dans ce domaine.

Afin d'évaluer quelle est la solution la plus simple et la plus efficace permettant d'atteindre l'objectif fixé, ce chapitre propose tout d'abord un tour d'horizon des méthodes rencontrées dans la littérature. Leur possibilité d'application à la commande des robots rapides et précis est ensuite discutée. Une méthode basée sur le modèle dynamique du robot est choisie et détaillée pour piloter le robot DELTA à entraînement direct. Diverses améliorations sont aussi proposées afin d'utiliser au mieux le potentiel du robot. Afin d'obtenir une plus grande rigidité et une meilleure précision à l'arrêt, le régulateur est divisé en un régulateur de positionnement pour la phase d'approche et en un régulateur de trajectoire pour la phase de mouvement. Une anticipation de la commande a priori permet aussi de réduire les erreurs dues aux retards introduits par l'amplificateur et le moteur.

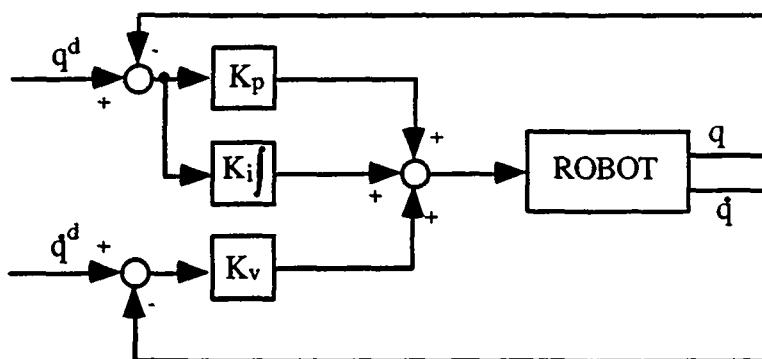
### 6.2. Tour d'horizon des méthodes d'asservissement des robots

Il n'est pas possible, dans le cadre de ce travail, de traiter en détail l'ensemble des approches existant dans la littérature. Seules les plus importantes seront considérées et

un poids plus important sera donné sur celles susceptibles d'être utilisées pour la commande du robot DELTA. Le lecteur intéressé pourra trouver d'autres informations dans [Dombre & Khalil 88], [Lind 88], [Craig 88], [Khosla 86], [Craig 86] ou encore [Asada & Slotine 86].

### 6.2.1. Commande classique

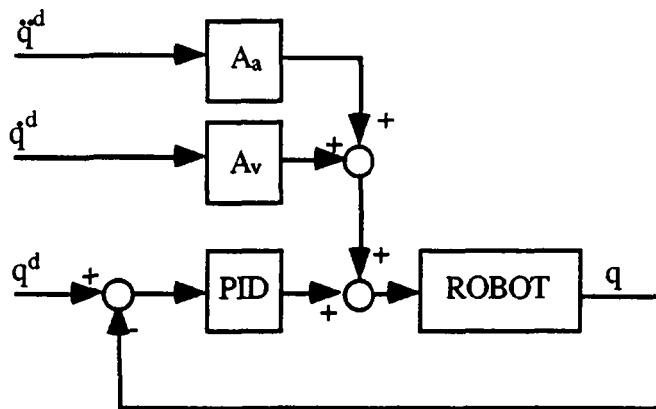
Le modèle dynamique décrit un système d'équations différentielles du second ordre, non linéaires et couplées. Pourtant, dans une commande classique, qui est celle de presque tous les robots industriels actuels, le mécanisme est considéré comme un système linéaire et chacune de ses articulations est asservie par une commande locale de type PID à gains constants. Ses avantages sont la facilité d'implantation et le faible coût en calcul. En contrepartie, la réponse temporelle du robot variant selon sa configuration, on constate des dépassements de consigne et une mauvaise précision de suivi dans les mouvements rapides. Dans bon nombre d'applications, ces inconvénients ne sont pas gênants. Ceci est d'ailleurs souvent grandement atténué par l'utilisation d'un réducteur entre le moteur et le bras (voir paragraphe 2.5.5). Le schéma de la figure 6.2.1 illustre cette approche, où chaque axe est réglé indépendamment des autres.



*Figure 6.2.1 : Schéma d'une commande classique PID*

On peut montrer, avec un tel schéma, que le terme intégral est indispensable pour éliminer l'erreur statique en position due aux forces de gravité et aux frottements secs. Cependant, pour une consigne de type rampe (vitesse constante) une erreur de traînée constante subsiste. Pour une consigne de type parabolique (accélération constante), l'erreur de traînée devient même infinie ! Pour s'affranchir de ces problèmes, certains ont ajouté une commande a priori de vitesse et/ou d'accélération locale sur chacun des

axes. De plus, si la mesure de vitesse n'est pas directement possible, un terme dérivateur sur la position aura le même rôle. Le schéma de cette commande ressemble alors à celui de la figure 6.2.2.



*Figure 6.2.2 : Commande classique PID avec commande a priori*

Bien que déjà plus complet que le précédent, ce schéma ne tient pas compte des non-linéarités et des couplages contenus dans un robot. Ceci a conduit les chercheurs à développer d'autres méthodes d'asservissement.

### 6.2.2. Méthodes basées sur le modèle dynamique du robot

Lorsque l'application exige des évolutions rapides du robot, et une grande précision lors du mouvement, il est nécessaire de concevoir un système de commande plus sophistiqué qui prenne en compte tout ou partie des forces d'interaction dynamiques.

Un premier schéma utilise le modèle dynamique complet du robot comme commande a priori et est connu sous le nom de "feedforward compensation" dans la littérature anglo-saxonne [Khosla 86]. Le modèle dynamique inverse permet de calculer les couples nécessaires à chaque articulation pour suivre la trajectoire désirée. Ce calcul peut s'effectuer soit avant le mouvement (hors-ligne) ou pendant le mouvement (en ligne ou en temps réel). Dans ce dernier cas, un modèle pas trop coûteux en temps de calcul doit être développé. En appliquant le couple calculé à chaque articulation, la trajectoire va être suivie sans erreur si le modèle du robot est exact et s'il n'y a pas de perturbation extérieure. En pratique, un modèle n'est jamais parfait et une boucle de contre-réaction doit être introduite (figure 6.2.3).

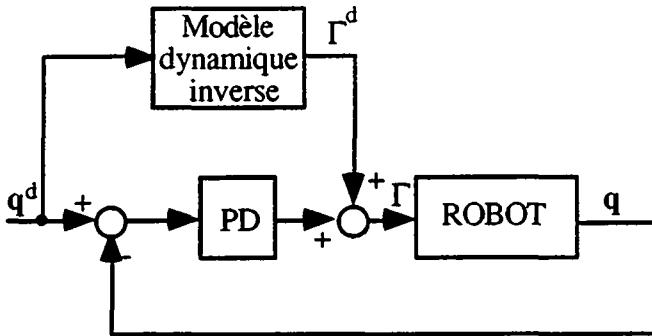


Figure 6.2.3 : Commande a priori selon le modèle dynamique

Le modèle dynamique inverse est calculé ici en fonction du mouvement désiré et prend la forme générale suivante, déjà rencontrée dans le chapitre 3 :

$$\Gamma^d = \tilde{A}(q^d) \cdot \dot{q}^d + \tilde{H}(q^d, \dot{q}^d) \quad [6.2.1]$$

La boucle de contre-réaction est constituée de n régulateurs PD indépendants, n étant le nombre d'axes du robot. Les couples appliqués aux différents axes du robot deviennent alors :

$$\Gamma = \Gamma^d + K_p \cdot (q^d - q) + K_v \cdot (\dot{q}^d - \dot{q}) \quad [6.2.2]$$

où  $K_p$  et  $K_v$  sont les matrices diagonales des gains de contre-réaction.

Une analyse de cette approche montre que le terme de commande a priori  $\Gamma^d$  a pour rôle de déplacer le point de fonctionnement du régulateur, de façon non-linéaire, et de considérer localement le système comme linéaire et découpé.

Toutefois, si l'on veut découpler globalement le robot, un autre schéma est nécessaire. L'utilisation de la commande par découplage non-linéaire constitue alors une bonne approche [Dombre & Khalil 88]. Elle est connue sous le nom de "computed-torque controller" dans la littérature anglo-saxonne [An & al 88]. Théoriquement, cette méthode assure le découplage et la linéarisation des équations du modèle se traduisant par une réponse uniforme quelle que soit la configuration du robot.

Il a été montré au chapitre 3, que le modèle dynamique d'un robot pouvait se mettre sous la forme suivante :

$$\Gamma = A(q) \cdot \dot{q} + H(q, \dot{q}) \quad [6.2.3]$$

où  $\Gamma$ ,  $q$  et  $H$  sont des vecteurs de dimension  $n$ , et  $A$  est la matrice d'inertie du robot, de dimension  $n \times n$  ( $n$  représente le nombre d'axes du robot).

A partir de cette équation, il est possible de calculer la relation inverse, appelée aussi modèle dynamique direct. On obtient :

$$\ddot{q} = A^{-1}(q) \cdot \{\Gamma - H(q, \dot{q})\} \quad [6.2.4]$$

Soient  $\tilde{A}$  et  $\tilde{H}$  les estimations respectives de  $A$  et  $H$ , obtenues par modélisation mathématique du robot. Choisissons alors une commande telle que :

$$\Gamma = \tilde{A}(q) \cdot w + \tilde{H}(q, \dot{q}) \quad [6.2.5]$$

où  $w$  est un nouveau vecteur de commande de dimension  $n$ .

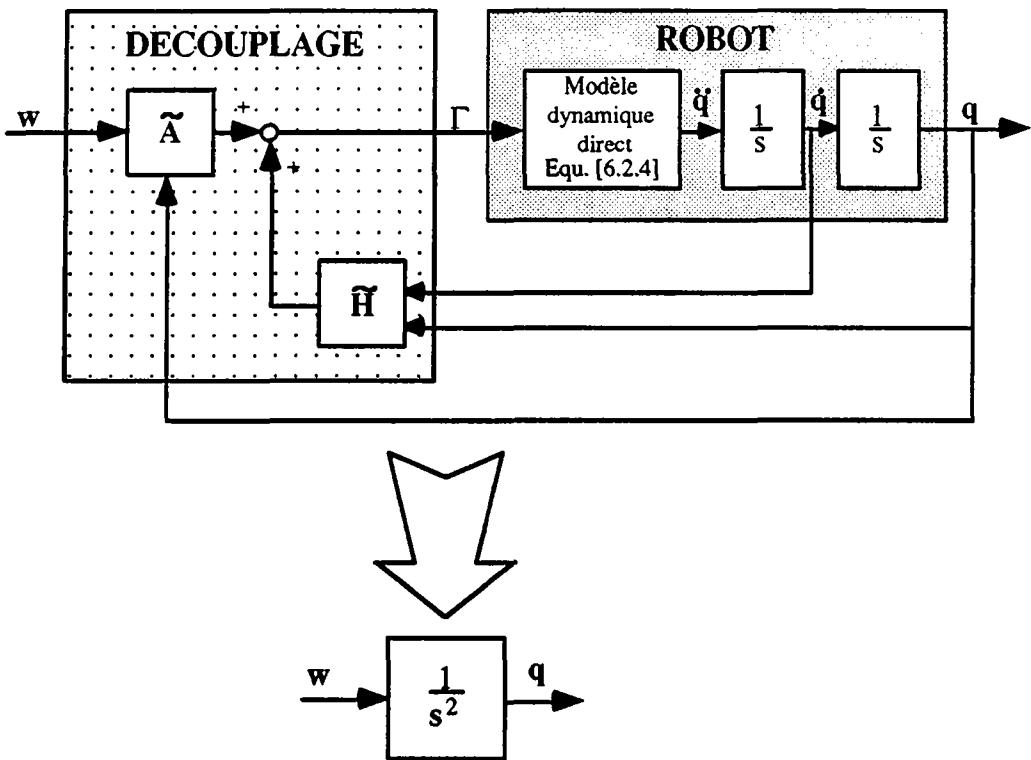
En remplaçant [6.2.5] dans [6.2.4], on obtient :

$$\ddot{q} = A^{-1} \cdot \{\tilde{A} \cdot w + \tilde{H} - H\} \quad [6.2.6]$$

Ainsi, dans le cas où le modèle est supposé parfait ( $\tilde{H} = H$  et  $\tilde{A} = A$ ), le système est régit par l'équation :

$$\ddot{q} = I \cdot w \quad [6.2.7]$$

où  $I$  représente la matrice identité de dimension  $n \times n$ . Ainsi, en utilisant la relation non-linéaire décrivant le modèle dynamique inverse du robot selon l'équation [6.2.5], il est possible de linéariser et découpler le système. On se ramène alors à un problème de commande de  $n$  systèmes linéaires, invariants, découplés et du second ordre (double intégrateur), comme illustré par la figure 6.2.4.



*Figure 6.2.4 : Principe de la commande par découplage non-linéaire*

Le vecteur de commande  $w$  s'identifie alors à l'accélération désirée, et il est généralement calculé sur la base d'une combinaison linéaire d'un signal de référence et d'un signal de contre-réaction. Différentes possibilités sont envisageables pour calculer cette commande. La plus fréquemment rencontrée est celle utilisant un régulateur PD pour le signal de contre-réaction, et une commande a priori basée sur l'accélération désirée  $\ddot{q}^d$  pour le signal de référence; ce dernier permettant de limiter dans une très large mesure les erreurs de traînée. A la limite, si le modèle est parfait et qu'il n'y a pas de perturbations extérieures, ce seul signal suffit, la contre-réaction n'étant alors plus nécessaire. La figure 6.2.5. illustre ce schéma de découplage non-linéaire.

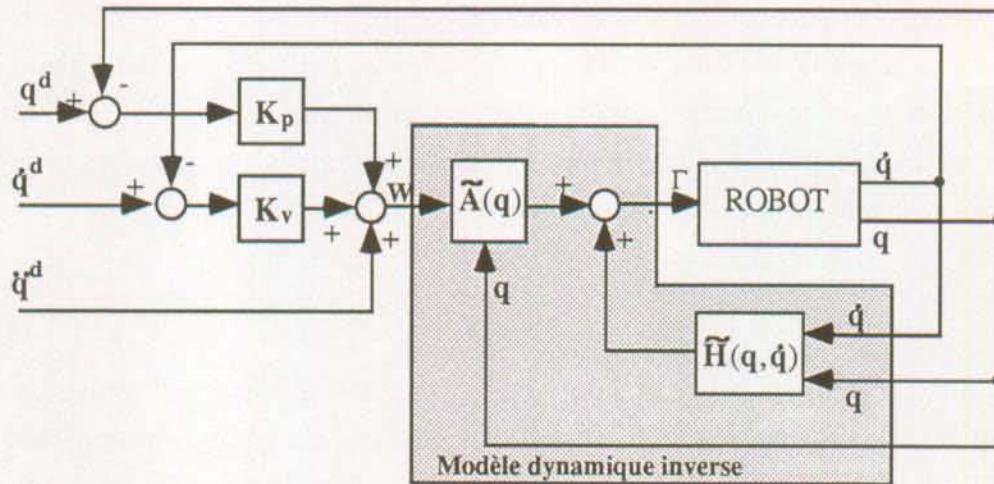


Figure 6.2.5 : Commande par découplage non-linéaire

Evidemment, l'exploitation de la commande par découplage non-linéaire suppose le calcul du modèle dynamique en temps réel, ce qui n'est pas toujours aisé. Ainsi, [Khalil 78] propose un autre schéma qu'il nomme "commande prédictive" dans lequel le calcul de  $\tilde{A}$  et  $\tilde{H}$  se fait, non pas en fonction des valeurs courantes de  $q$  et de  $\dot{q}$ , mais plutôt en fonction du mouvement désiré  $q^d$  et  $\dot{q}^d$ . Le schéma a alors l'allure de la figure 6.2.6. S'il n'y a pas d'erreur de modélisation et aucune perturbation, cette commande, pour des erreurs initiales nulles, linéarise et découple les équations du système comme dans le cas précédent.

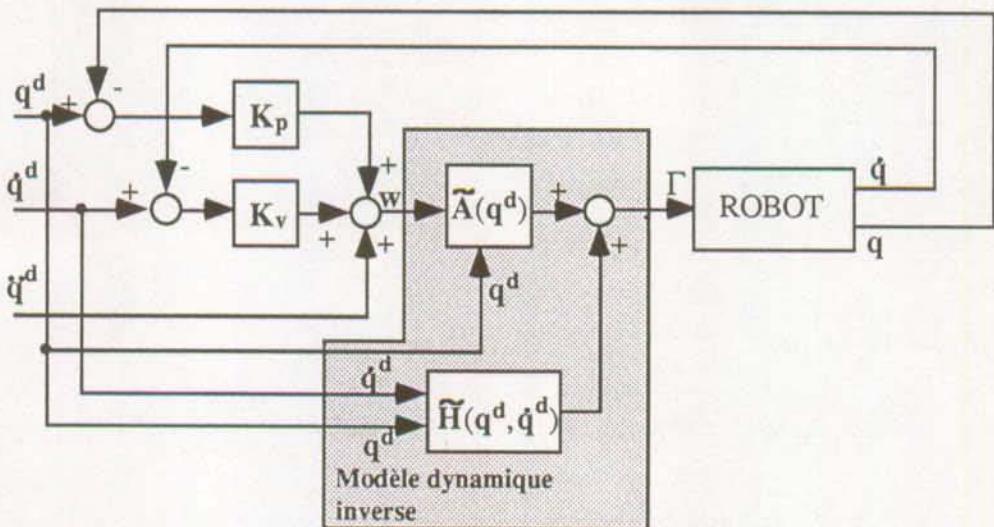


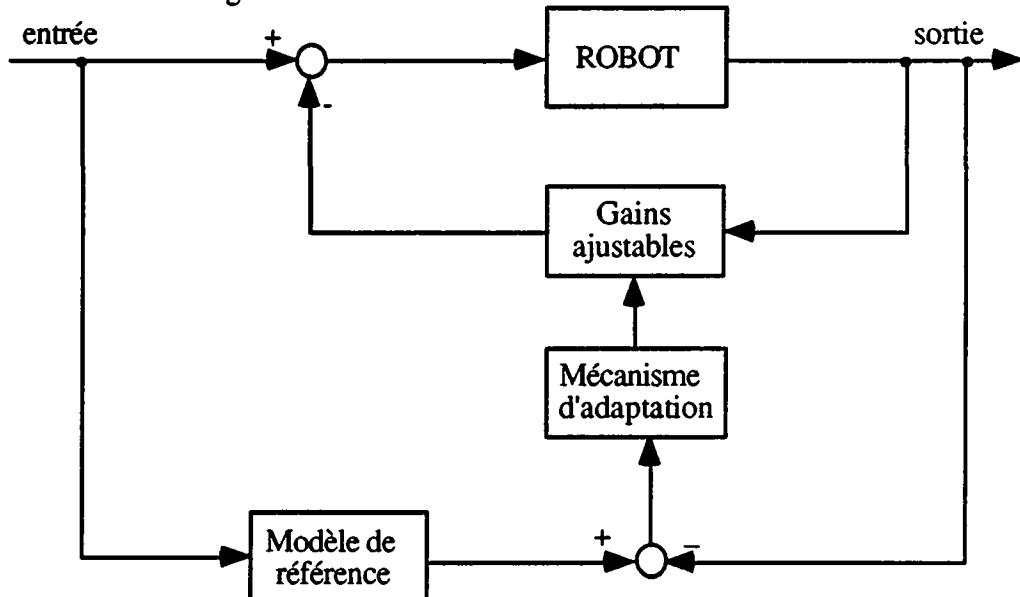
Figure 6.2.6 : Commande prédictive

Lorsque la trajectoire est spécifiée directement dans l'espace opérationnel, une méthode permettant de réaliser le découplage non-linéaire directement dans ce même espace est proposée par Luh, Walker et Paul [Luh & al 80], mais sa description sort du cadre de ce travail. On peut relever cependant qu'elle nécessite le calcul de l'inverse de la jacobienne ( $J^{-1}$ ) et de sa dérivée ( $\dot{J}$ ). Ceci compromet son utilisation en temps réel dans la plupart des cas.

### 6.2.3. Méthodes adaptatives

L'utilisation du modèle dynamique du robot dans la boucle de réglage paraît séduisante. Cependant, le temps à investir pour établir le modèle et la méconnaissance de ses paramètres ont poussé les chercheurs à développer d'autres méthodes d'asservissement. En particulier, beaucoup d'efforts ont été placés ces dernières années sur la "commande adaptive". L'idée de ces méthodes est d'adapter la loi de commande en se basant sur un critère d'erreur spécifique.

La commande adaptive avec modèle de référence ou MRAC (Model Reference Adaptive Control) est un schéma qui a été souvent rencontré dans la littérature. Il est représenté dans la figure 6.2.7.



*Figure 6.2.7 : Commande adaptive avec modèle de référence*

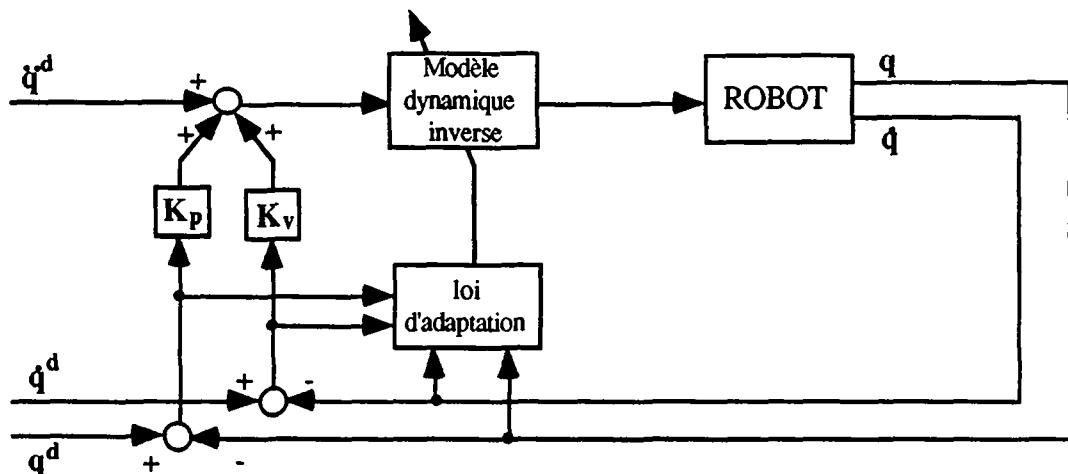
L'idée principale est de forcer le robot à suivre un modèle de référence le mieux possible. Ceci est réalisé en ajustant les gains de contre-réaction de la commande en fonction de l'erreur entre la réponse du modèle et celle du système réel. Souvent, le modèle de référence choisi est une équation différentielle linéaire du deuxième ordre

pour chaque degré de liberté [Dubowsky & DesForges 79]. Un tel modèle ne tient évidemment pas compte des forces de gravité et néglige les effets non-linéaires. De plus, la stabilité globale de ce schéma n'est pas toujours garantie, ce qui le rend inutilisable en pratique; surtout si, comme pour les robots rapides, les paramètres varient aussi rapidement que le phénomène d'adaptation. D'autres schémas du type MRAC utilisant différents concepts, ont été proposés, et le lecteur intéressé peut se référer à [Hsia 86] ou [Craig 88] qui font une bonne synthèse de ces méthodes.

D'autres algorithmes, basés sur la théorie des régulateurs auto-adaptatifs figurent aussi dans la littérature [Koivo & Guo 81], [van Griethuysen 87], mais leur désavantage est que le robot doit être modélisé comme un système linéaire et découplé, ce qui n'est pas conforme à la réalité.

Ainsi, la majorité des travaux dans le domaine de la commande adaptative ont simplement voulu appliquer aux robots les théories développées pour les systèmes linéaires, et ainsi, malgré les nombreuses publications dans ce sens, les quelques résultats expérimentaux réalisés sont peu prometteurs.

Pour cette raison, on a vu ces dernières années la réapparition du modèle dynamique dans les méthodes adaptatives et [Craig 88] propose une méthode adaptative basée sur le schéma de la commande par découplage non-linéaire (figure 6.2.8).

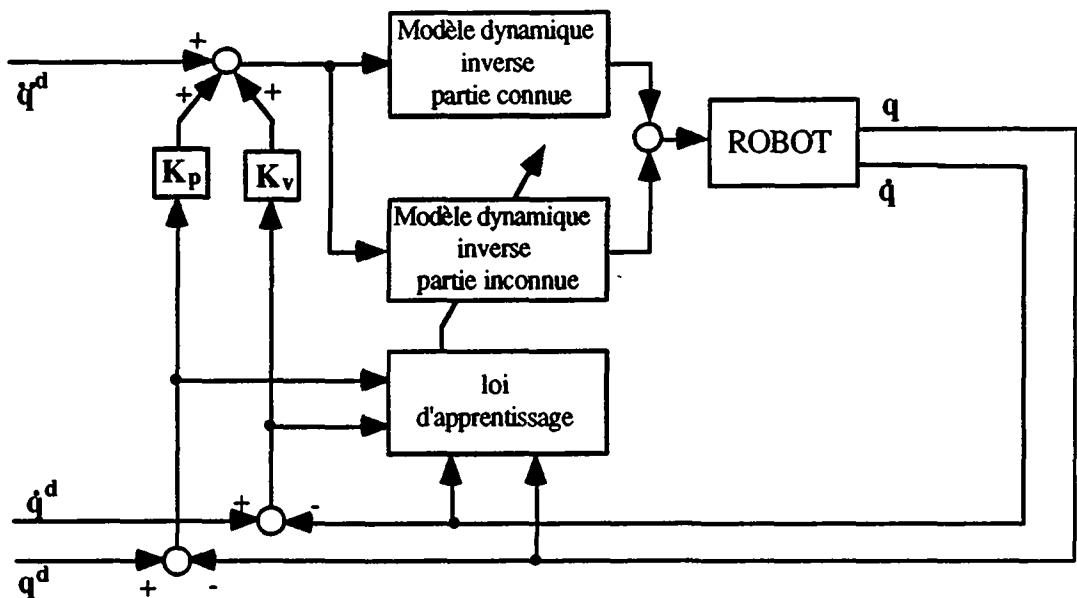


*Figure 6.2.8 : Commande adaptative avec identification des paramètres du modèle*

L'idée ici est d'identifier les paramètres du modèle non-linéaire complet plutôt que ceux d'un modèle linéaire qui ne reflète de toute manière pas la réalité. Ce schéma est

aussi adopté par [Faessler 88a], mais étendu avec un modèle de référence linéaire du deuxième ordre sur le système découplé.

Lorsque la trajectoire est répétitive, ce qui est souvent le cas dans les applications robotiques, une commande par apprentissage peut être utilisée pour corriger les phénomènes non-modélisés et ainsi, au fur et à mesure de son utilisation, le robot corrigera ses erreurs de suivi de trajectoire. Un schéma allant dans ce sens est proposé par [Craig 88] et est montré à la figure 6.2.9.



*Figure 6.2.9 : Commande par apprentissage*

#### 6.2.4. Commande par mode de glissement

Par opposition aux méthodes adaptatives qui corrigent les gains de rétroaction pour obtenir toujours la meilleure réponse garantissant la stabilité, les méthodes par mode de glissement utilisent une commande suffisamment grande permettant d'imposer au système une réponse désirée. La commande se fait généralement en tout ou rien, ce qui rend le régulateur robuste face à une large variation des paramètres du système [Guzzella 86]. Cependant, le signal haché de la commande peut exciter les modes négligés de haute fréquence et causer un bruit désagréable ou même détériorer la mécanique du système.

Pour résoudre ces problèmes, différents auteurs ont proposés des méthodes où une portion linéaire est introduite dans la loi de commande. D'autres introduisent le modèle

du robot afin d'ajuster les valeurs de commande à sa dynamique. Le lecteur intéressé par plus de détails sur le sujet trouvera des informations dans [Asada & Slotine 86] ou [Bühler 86].

Ces méthodes font actuellement encore partie du domaine de la recherche et très peu d'applications pratiques ont vu le jour à l'heure actuelle.

### 6.2.5. Commande "neuronale"

Ces dernières années, avec les progrès de la science cognitive, quelques méthodes ont été développées pour la commande de robot en se basant sur les réseaux de neurones. Les résultats obtenus à l'heure actuelle ne sont pas encore satisfaisants, et la phase d'apprentissage nécessaire pour organiser le réseau est encore lente. Cependant, les premiers résultats obtenus semblent déjà prometteurs et la méthode reste à affiner. Elle peut notamment devenir très intéressante pour l'apprentissage automatique du modèle dynamique du robot, comme le suggèrent [Kawato & al 88] et [Guglielmetti 90].

## 6.3. Application aux robots rapides et précis

### 6.3.1. Introduction

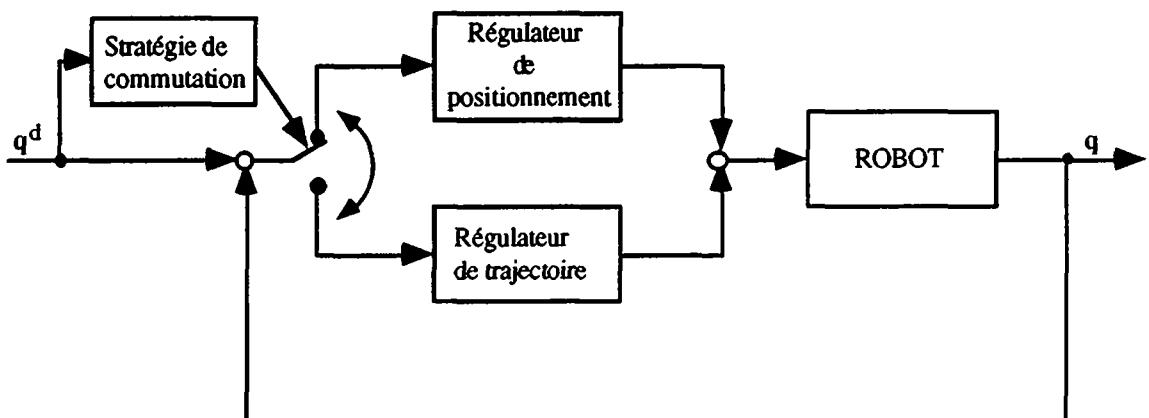
Le tour d'horizon effectué au paragraphe précédent montre l'utilité d'incorporer le modèle dynamique du robot dans l'algorithme de réglage. Ceci est d'autant plus important dans le cas de la commande des robots rapides, puisqu'à ce moment-là, les effets non-linéaires deviennent prédominants. La seule utilisation d'un régulateur classique n'est alors plus possible et le recours à d'autres méthodes est nécessaire. Parmi celles exposées précédemment, seules celles basées sur un modèle seront utilisées ici, les méthodes adaptatives ne permettant pas encore des déplacements rapides du robot et les autres n'étant pas encore assez évoluées au point de vue théorique.

Si l'on analyse le problème de la palettisation d'objets sous le regard du réglage, on peut considérer deux phases distinctes. La première est une phase de mouvement sur une trajectoire reliant le point de départ et le point d'arrivée dans un temps que l'on veut minimal. La deuxième est une phase de stabilisation sur le point d'arrivée. Celle-

ci aussi doit être la plus courte possible. On remarque ici le caractère antagoniste de ces deux phénomènes. En effet, dans la pratique, par l'utilisation de régulateurs classiques, il a été remarqué que plus le mouvement est rapide, plus le dépassement à l'arrivée est grand et donc plus le temps de stabilisation est important. Un compromis entre les deux n'est pas simple à trouver, et comme déjà mentionné dans le chapitre 5, le type de trajectoire et de loi de mouvement utilisé a son importance.

Les dépassements à l'arrivée proviennent généralement d'un mauvais suivi de la trajectoire et il est donc légitime de penser que si la trajectoire est parfaitement suivie, le temps de stabilisation tendra vers zéro. L'utilisation du modèle dynamique du robot pour améliorer le suivi de trajectoire lors de la phase de mouvement peut alors contribuer à une meilleure stabilisation à l'arrivée.

D'autre part, si l'on analyse ces deux phases séparément, on remarque que leur but n'est pas le même. En effet, lors du mouvement, on cherchera à suivre le mieux possible une consigne (asservissement) alors que dans le cas du positionnement, on cherchera plutôt à garder la même consigne face aux perturbations extérieures (régulation). La distinction entre un régulateur de positionnement et un régulateur de trajectoire apparaît alors comme une évidence, et une stratégie pour la commutation entre les deux régulateurs est à développer. La figure 6.3.1. montre le schéma de principe de ce régulateur. L'originalité de cette méthode réside dans la possibilité de choisir des régulateurs différents, et surtout adaptés à la fonction réelle qu'ils ont à remplir.

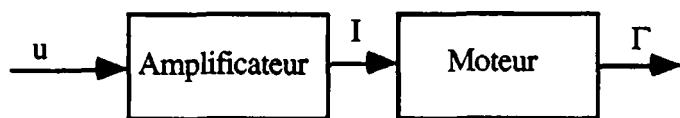


*Figure 6.3.1 : Régulateurs de positionnement et de trajectoire*

Ces deux régulateurs seront détaillés dans la suite de ce chapitre. Au préalable, il est cependant nécessaire de modéliser le moteur et son amplificateur, éléments de base pour le contrôle du robot.

### 6.3.2. Modélisation du moteur et de l'amplificateur

Afin de pouvoir appliquer de façon judicieuse la commande sur le moteur, il est nécessaire d'en établir un modèle. La figure 6.3.2. illustre sous forme de schéma bloc le système à modéliser.

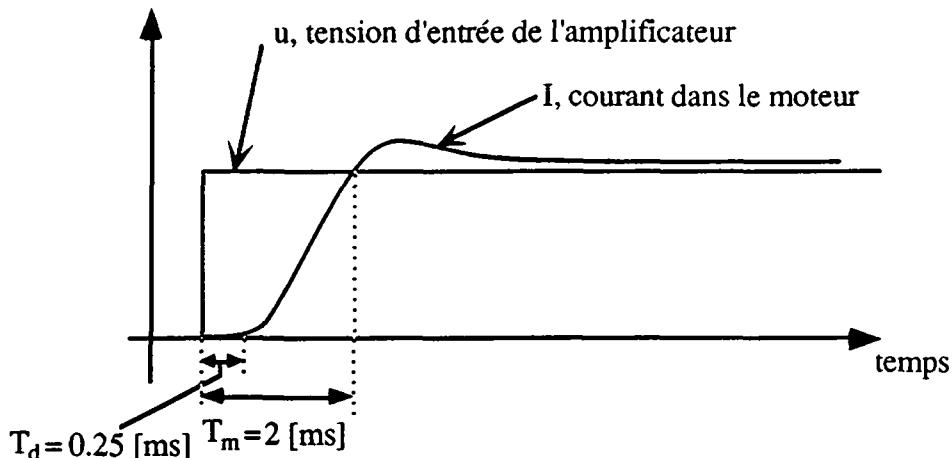


*Figure 6.3.2 : Schéma bloc amplificateur et moteur*

La variable d'entrée est une tension de commande et la variable de sortie le couple appliqué par le moteur. Pour les moteurs sans balais, la relation entre le courant dans le moteur et le couple de sortie s'exprime par un simple facteur de proportionnalité [Asada & Youcef-Toumi 87], soit :

$$\Gamma = K_T \cdot I \quad [6.3.1]$$

La relation entre la commande  $u$  et le courant  $I$  est légèrement plus complexe. Une mesure effectuée sur les amplificateurs et moteurs du DELTA à entraînement direct a permis de mettre en évidence la réponse du courant à un saut indiciel de la commande. Pour cette mesure, la valeur de la tension d'entrée est relativement faible (environ 10 % du couple maximal) afin d'éviter les effets non-linéaires provoqués par les limites physiques du moteur. Ceci est en accord avec l'utilisation future du moteur dans la mesure où il est piloté à partir d'une loi de mouvement limitant les sauts importants de la consigne. Cette mesure est représentée par la figure 6.3.3.



**Figure 6.3.3 : Réponse du courant à un saut unité (moteur bloqué)**

Comme on peut le constater, la réponse est composée d'un retard pur  $T_d = 250 \text{ } [\mu\text{s}]$  environ puis d'une montée du courant dans le moteur. La valeur de consigne du courant est atteinte après environ  $T_m = 2 \text{ [ms]}$ . Si le retard pur provient du temps de calcul nécessaire à l'amplificateur numérique, la montée du courant est liée à la constante de temps électrique du moteur et donc à l'inductance de son bobinage. Le fait d'utiliser un moteur de taille importante augmente immanquablement ce temps de montée. La réponse pourrait être modélisée par un système du deuxième ordre, mais l'utilisation que l'on désire en faire nous permet d'opter pour un modèle beaucoup plus simple considérant la réponse comme un saut unité retardé de  $T_m$ . Pour une période d'échantillonnage  $T_h = 1 \text{ [ms]}$ , ceci correspond à un retard de 2 périodes. Le modèle est le suivant :

$$I(t) = K_I \cdot u(t - T_m) \quad [6.3.2]$$

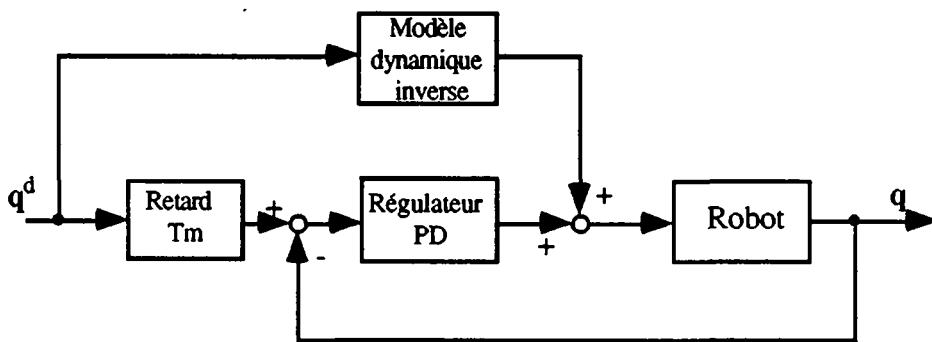
Le modèle total du système devient alors :

$$\Gamma(t) = K_T \cdot K_I \cdot u(t - T_m) \quad [6.3.3]$$

où  $T_m = 2 \text{ [ms]}$  dans notre cas.

### 6.3.3. Régulateur de trajectoire

Le but du régulateur de trajectoire est de suivre le mieux possible la consigne provenant du générateur de trajectoire. Les régulateurs classiques utilisés sur la plupart des robots du marché sont basés sur une structure conforme à celle de la figure 6.2.2. Ils ne tiennent compte ni de l'aspect non-linéaire du robot, ni du couplage entre les axes. Lorsque les mouvements sont rapides, une telle approche n'est plus satisfaisante et il est nécessaire de tenir compte du modèle dynamique du robot. [Khosla 86] a montré l'équivalence des performances entre la méthode basée sur une commande a priori de la dynamique et celle basée sur le découplage non-linéaire si le modèle dynamique du robot est proche de la réalité. Pour le robot DELTA, la formulation du modèle dynamique étant plus favorable à une utilisation comme commande a priori, c'est sous une forme similaire au schéma 6.2.3. que celui-ci a été implanté. Pour tenir compte du temps de retard  $T_m$  causé par la montée du courant dans le moteur, un retard du même ordre est introduit sur la consigne, afin qu'elle agisse sur le moteur en même temps que la commande a priori. Le modèle dynamique inverse est calculé à partir de l'équation [3.3.7] et utilisé en temps réel. La figure 6.3.4 illustre le principe de ce régulateur de trajectoire.



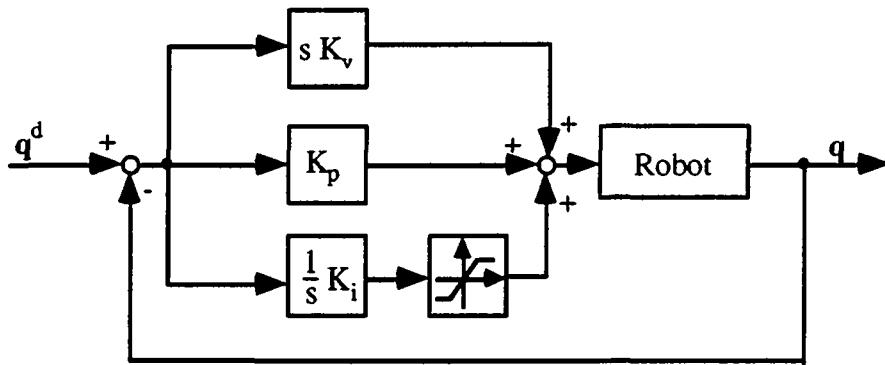
*Figure 6.3.4 : Régulateur de trajectoire*

Comme nous le verrons au chapitre 7, la prise en compte du retard  $T_m$  améliore encore les performances de suivi de trajectoire. De ce fait, elle devient même plus intéressante que la méthode par découplage non-linéaire, qui, basée uniquement sur des valeurs de contre-réaction ne permettrait pas simplement d'en tenir compte. Il est évident que ceci est possible uniquement si l'on connaît la trajectoire à l'avance sur la durée  $T_m$  au minimum; ce qui est le cas pour la plupart des applications industrielles. La boucle de contre-réaction est basée uniquement sur un régulateur PD suffisamment

robuste pour être insensible aux variations d'inertie du robot et aux autres perturbations extérieures.

#### 6.3.4. Régulateur de positionnement

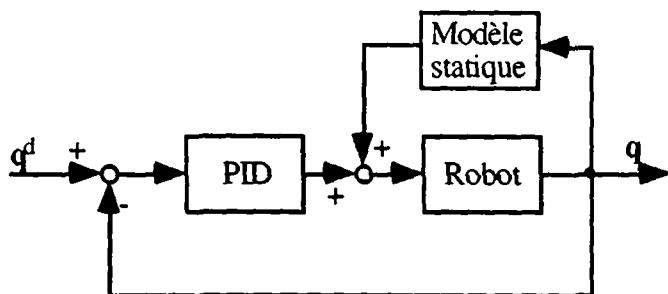
Une fois la trajectoire terminée, il s'agit de positionner le plus rapidement possible le robot à sa valeur de consigne puis de le maintenir à cette position malgré les perturbations extérieures (frottement, couple réluctant, forces parasites, ...). C'est le rôle du régulateur de positionnement. Il doit permettre d'éliminer les erreurs de statisme et compenser les perturbations extérieures. Pour cette raison, un régulateur de type PID a été choisi. Une surveillance de la sortie de l'intégrateur a été introduite, par mesure de sécurité, pour limiter la correction lors de perturbations extérieures trop importantes. La figure 6.3.5. en donne un schéma bloc. Il faut noter ici, que si l'on modélise le robot par un double intégrateur, il n'est théoriquement pas possible d'utiliser un terme intégrateur sous peine de déstabiliser le système. En pratique, le frottement et autres phénomènes négligés rendent toutefois son utilisation possible, mais la marge de stabilité est fortement réduite. Il convient alors de l'utiliser avec précaution.



*Figure 6.3.5 : Régulateur de positionnement*

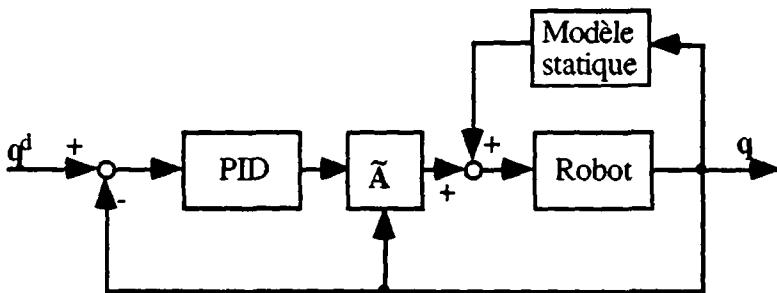
Sous cette forme, le régulateur permet d'éliminer les erreurs de statisme, mais il converge lentement. La vitesse de convergence vers la valeur finale est en effet limitée par la constante de temps de l'intégrateur, elle-même limitée par la marge de stabilité désirée. Pour remédier à ce problème, on pourrait par exemple incorporer un modèle "statique" du robot dans la contre-réaction (figure 6.3.6). Un modèle de gravité a été utilisé dans notre cas, mais dans l'application pratique celui-ci s'est avéré insuffisant, car négligeable vis-à-vis des frottements et surtout du couple réluctant des moteurs.

Un modèle tenant compte de ces phénomènes permettrait certainement d'obtenir des résultats intéressants.



*Figure 6.3.6 : Régulateur avec modèle statique*

Pour augmenter la robustesse du régulateur, il paraît intéressant d'introduire la matrice d'inertie du robot dans la boucle de réglage comme le montre la figure 6.3.7. Ce schéma aurait d'autre part l'avantage de découpler les axes entre eux et d'éviter ainsi que la correction sur l'un perturbe les autres et pourrait aussi être utilisé dans la phase de mouvement. Ceci sortant du cadre de ce travail n'a pas été testé.



*Figure 6.3.7 : Régulateur avec découplage par la matrice d'inertie*

Il est utile de noter ici que la prise en compte d'un modèle (statique ou dynamique) et l'introduction de la matrice d'inertie du robot dans la boucle de réglage ont pour but de linéariser le système et de le faire tendre vers un double-intégrateur. Dans cet esprit, l'utilisation du terme intégrateur du régulateur aura tendance à diminuer la marge de stabilité, et à la limite, si le modèle est parfait, à rendre le système complètement instable. Ainsi, d'autres solutions devraient être recherchées pour le régulateur de positionnement, comme par exemple, les régulateurs de type PID<sup>2</sup>, RST avec compensateur de perturbation ou régulateurs d'état.

### 6.3.5. Stratégie de commutation

La commutation entre le régulateur de trajectoire et le régulateur de positionnement a son importance pour la rapidité de convergence vers la valeur de consigne. On peut jouer aussi bien sur l'instant de commutation que sur les valeurs initiales des paramètres du régulateur de positionnement. La stratégie choisie dans notre cas est relativement simple. En effet, seul l'instant de commutation peut être modifié. Sa valeur est choisie par expérimentation sur le système. La valeur de l'intégrateur est initialisée à 0, bien qu'une autre valeur pourrait permettre de tendre plus rapidement vers la position finale. D'autres travaux devraient être entrepris ici pour améliorer la méthode.

### 6.3.6. Choix des gains du régulateur

Les régulateurs utilisés sont de type PD ou PID. Le choix de leurs gains a été réalisé par expérimentation, en considérant le cas le plus défavorable (inertie minimale). La méthode par pompage de Ziegler-Nichols est utilisée dans une première phase pour obtenir une estimation des gains. Ces derniers sont ensuite ajustés manuellement, par itérations, en observant la réponse du système.

### 6.3.7. Période d'échantillonnage

Le choix d'une valeur "optimale" de la période d'échantillonnage est un problème fort discuté en automatique. En fait celle-ci doit être compatible avec la dynamique de la boucle fermée. Dans le cas des robots rapides, il est donc nécessaire de la choisir suffisamment petite, mais dans la limite des possibilités de calcul de l'électronique de commande. La mesure de la vitesse par dérivée de la position lue sur un capteur incrémental amène un bruit de quantification qui augmente lorsque la période d'échantillonnage diminue. Avec un capteur dont le nombre d'incrément est limité, un compromis est à trouver. Dans notre cas, une période d'échantillonnage de 1 [ms] s'est avéré être un choix satisfaisant.

### 6.3.8. Prise en compte de la charge transportée

Un robot destiné à des opérations de palettisation devra généralement saisir différents objets de nature et surtout de masse différentes. Ceci se traduit sur la

commande par une variation des paramètres dynamiques de la structure mobile. Si le régulateur est suffisamment robuste, cette différence pourra être corrigée par la contre-réaction. Cependant, pour des masses relativement élevées par rapport au robot, ceci ne suffit plus, et d'autres méthodes doivent être envisagées. Pour la plupart des applications industrielles, la masse de l'objet à transporter est connue a priori. Il suffit alors de l'introduire dans le calcul du modèle dynamique inverse du robot pour en tenir compte. C'est cette méthode qui est envisagée dans notre approche. Toutefois, si la masse n'est pas connue a priori, l'utilisation de méthodes d'identification est envisageable, comme proposé dans la littérature par [An & al 88].

## 6.4. Conclusion

Parmi les nombreuses méthodes modernes de réglage des robots, seules celles qui utilisent un modèle dynamique de la structure mobile peuvent être raisonnablement appliquées à des robots rapides. Malgré tout, très peu de robots disponibles sur le marché proposent un tel régulateur, le temps de calcul du modèle dynamique étant souvent beaucoup trop élevé pour une utilisation raisonnable. Le modèle développé pour le robot DELTA est suffisamment performant pour être évalué à chaque période d'échantillonnage. Son utilisation comme commande a priori est alors intéressante pour améliorer les performances lors de la phase de mouvement du robot. Son anticipation permet même de s'affranchir du temps d'établissement du courant dans le moteur, un peu à l'image des procédés d'avance à l'allumage dans les moteurs à explosion où l'avance à la commutation utilisée pour une meilleure commande des moteurs synchrones ou des moteurs pas à pas. Pour la phase de positionnement, un régulateur simple, de type PID, est utilisé afin d'éliminer les erreurs de statisme. Il a cependant le désavantage d'être relativement lent et, d'avoir une marge de stabilité relativement faible. D'autres types de régulateurs (PID<sup>2</sup>, RST, régulateur d'état) pourraient être envisagés à l'avenir.

De manière générale, toute connaissance du système doit être utilisée dans le cas du réglage des systèmes à dynamique rapide, les performances en seront d'autant améliorées. Pour s'en convaincre, il ne reste plus qu'à effectuer quelques mesures permettant de comparer les différents régulateurs. Ceci fera l'objet du prochain chapitre.



## 7. Résultats expérimentaux avec le robot DELTA

### 7.1. Introduction

Dans ce chapitre, nous présentons les résultats expérimentaux obtenus avec le robot DELTA à entraînement direct. Les mesures réalisées doivent permettre de vérifier les hypothèses émises dans les chapitres précédents et de confirmer les choix effectués, soit en particulier :

- démontrer l'effet de l'incorporation du modèle dynamique du robot dans la boucle de réglage
- montrer l'importance de la commutation sur le régulateur de positionnement pour éliminer les erreurs de statisme à l'arrivée
- illustrer la nécessité de trajectoires et de lois de mouvement compatibles avec la dynamique du robot.
- vérifier le bien-fondé du choix d'un entraînement direct

### 7.2. Performances du régulateur

Le régulateur de trajectoire proposé dans le paragraphe 6.3.3. utilise le modèle dynamique inverse du robot comme commande a priori avec anticipation. Pour évaluer ses performances, il est intéressant de pouvoir le comparer avec des régulateurs classiques comme ceux que l'on trouve sur les robots du marché. Quatre types de régulateurs sont traités ici :

- 1) **Régulateur PD indépendant sur chaque axe**, dont l'équation du vecteur de commande est :

$$\mathbf{u} = \mathbf{K}_p(\mathbf{q}^d - \mathbf{q}) + \mathbf{K}_v(\dot{\mathbf{q}}^d - \dot{\mathbf{q}}) \quad [7.2.1]$$

où  $\mathbf{K}_p$  et  $\mathbf{K}_v$  sont les matrices diagonales des gains considérant le robot comme linéaire et découplé.

2) *Régulateur PD avec commande a priori d'accélération*, dont l'équation est :

$$\mathbf{u} = \mathbf{K}_p(\mathbf{q}^d - \mathbf{q}) + \mathbf{K}_v(\dot{\mathbf{q}}^d - \dot{\mathbf{q}}) + \mathbf{I}_m \ddot{\mathbf{q}}^d \quad [7.2.2]$$

où  $\mathbf{I}_m$  est une matrice diagonale représentant l'inertie moyenne typique de chaque axe du robot.

3) *Régulateur PD avec commande a priori du couple selon modèle dynamique*, dont le schéma est équivalent à celui de la figure 6.3.4 où  $T_m = 0$ . Son équation est la suivante :

$$\mathbf{u} = \mathbf{K}_p(\mathbf{q}^d - \mathbf{q}) + \mathbf{K}_v(\dot{\mathbf{q}}^d - \dot{\mathbf{q}}) + \mathbf{M} \quad [7.2.3]$$

où  $\mathbf{M}$  est le vecteur couple a priori calculé selon la formule 3.3.7.

4) *Régulateur PD avec commande a priori anticipée du couple*. Ce schéma est équivalent à celui de la figure 6.3.4. avec, dans notre cas,  $T_m = 2$  [ms] (voir § 6.3.2). L'équation de commande devient :

$$\mathbf{u}(t) = \mathbf{K}_p(\mathbf{q}^d(t-T_m) - \mathbf{q}(t)) + \mathbf{K}_v(\dot{\mathbf{q}}^d(t-T_m) - \dot{\mathbf{q}}(t)) + \mathbf{M}(t) \quad [7.2.4]$$

Pour permettre une comparaison aisée, la même trajectoire est choisie pour tous les essais. Il s'agit d'une demi-ellipse de points de passage conformes à ceux du paragraphe 5.4.5 et illustrée par la figure 7.2.1. La loi de mouvement est de type cycloïdal avec des accélérations maximales de 80 [m/s<sup>2</sup>], ce qui donne un temps de parcours de 153 [ms] pour la consigne.

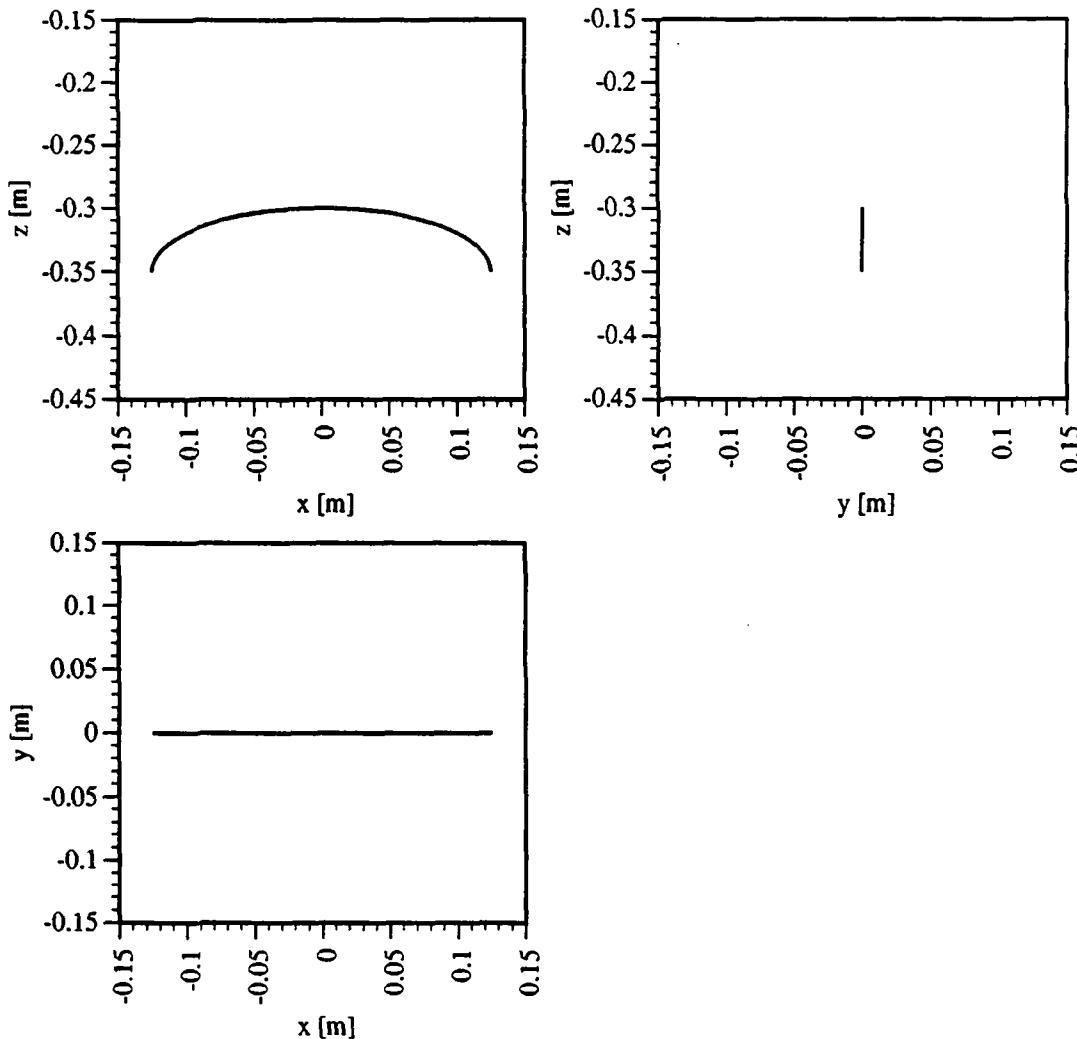
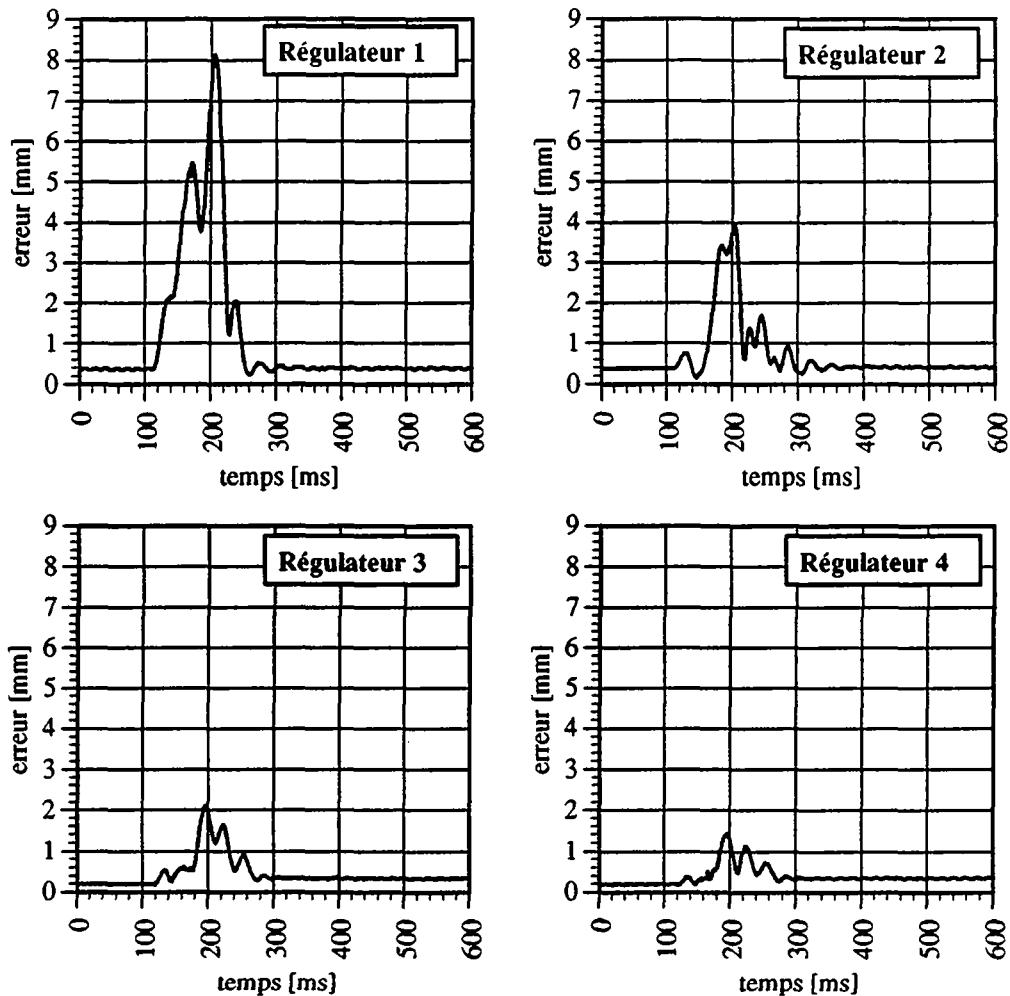


Figure 7.2.1 : Trajectoire demi-ellipse de test (projection dans les 3 plans du repère principal)

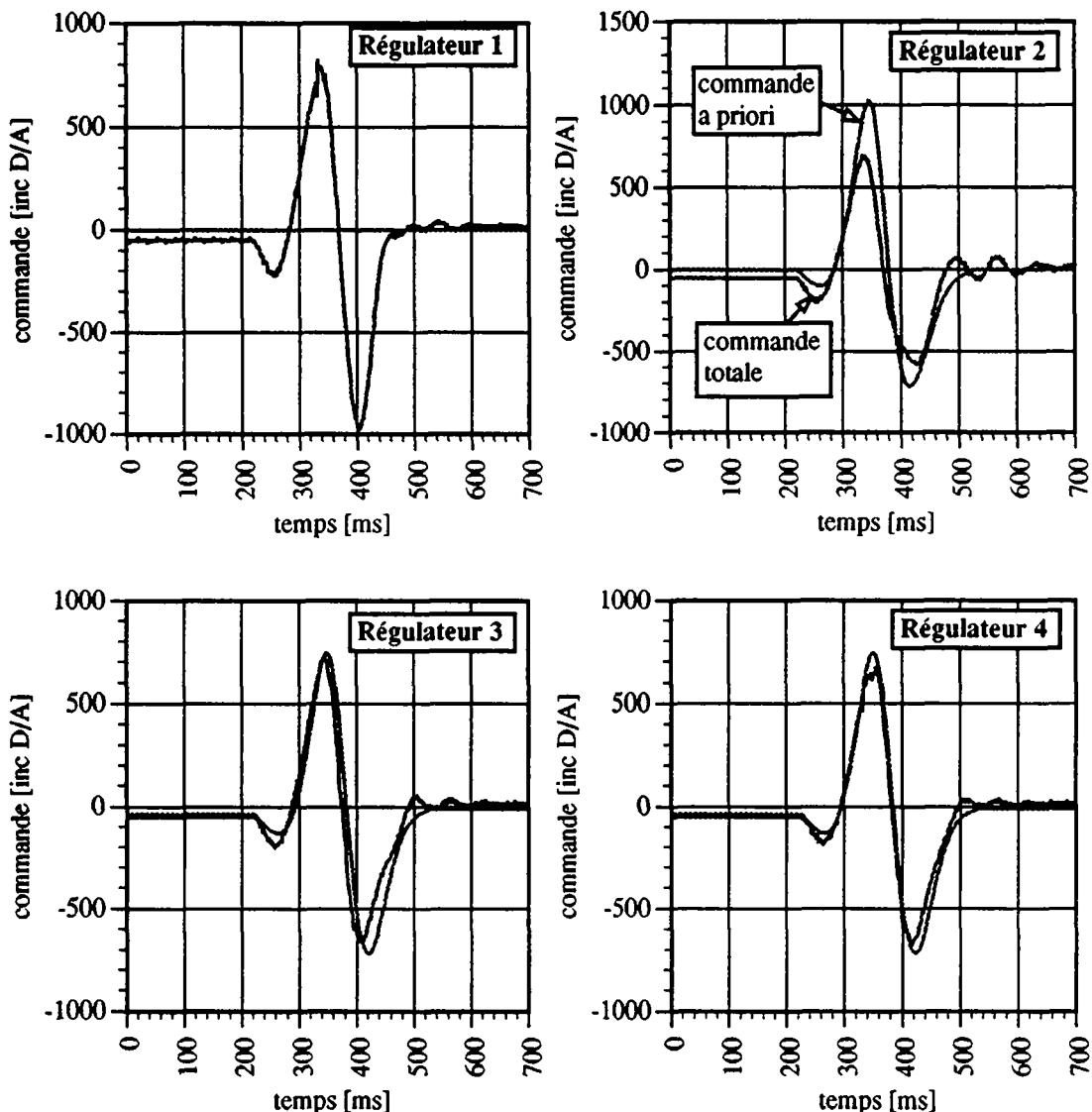
La figure 7.2.2 montre les erreurs de suivi de trajectoire obtenues pour chacun des régulateurs testés, la valeur des gains restant la même d'un régulateur à l'autre. Le mouvement commence au temps  $t = 120$  [ms], et atteind sa valeur finale au temps  $t = 273$  [ms].



*Figure 7.2.2 : Erreurs de suivi de trajectoire pour les 4 régulateurs testés*

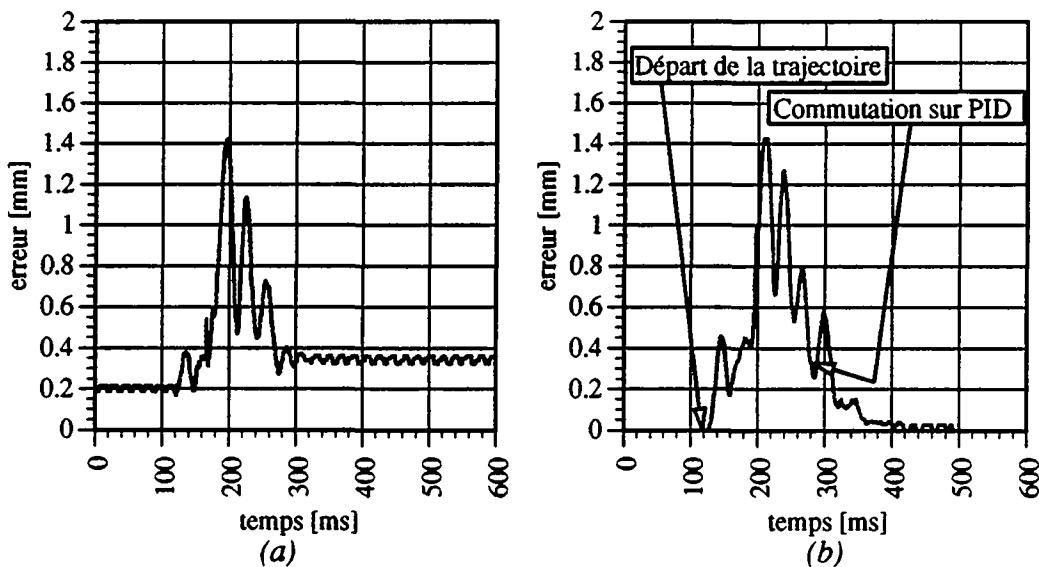
A l'analyse de ces graphes, on remarque que plus le modèle utilisé pour le calcul de la commande a priori est complet, meilleur est le suivi de trajectoire. L'amélioration globale en passant d'un simple régulateur PD (1) à un régulateur avec commande a priori anticipée du couple (4) est importante. L'erreur maximale peut ainsi être diminuée d'un facteur 5.7 environ! Ceci démontre la nécessité d'incorporer le modèle dynamique le plus complet possible dans l'algorithme de commande des robots rapides. Afin d'obtenir une certaine rigidité de la part du régulateur, les gains choisis sont assez élevés, ce qui produit une oscillation faiblement amortie d'une période d'environ 30 [ms].

La figure 7.2.3 montre l'allure des commandes pour chacun des régulateurs. Dans le cas 1, toute la commande est générée par le régulateur PD. Pour chacun des régulateurs suivants, la commande totale est composée d'une partie a priori et d'une partie venant de la correction par le régulateur PD. On remarquera que dans le cas du régulateur 4, la commande calculée a priori est très proche de la commande envoyée. Le travail du régulateur PD est donc très limité, ce qui explique les bons résultats obtenus et confirme la validité du modèle développé.



*Figure 7.2.3 : Commande totale et commande a priori pour les 4 régulateurs*

On remarque cependant, que l'erreur ne s'annule jamais avec de tels régulateurs, même lorsque la consigne finale est atteinte et stationnaire. Pour pouvoir y arriver, il est nécessaire d'ajouter un terme intégrateur pour éliminer les erreurs de statisme provenant du couple réluctant des moteurs et du frottement. C'est le rôle du régulateur de positionnement qui est enclenché dans la phase finale du mouvement. La figure 7.2.4 montre l'annulation de l'erreur (b) à la fin du mouvement, et peut être comparée à celle sans commutation sur le régulateur de positionnement (a).



*Figure 7.2.4 : Erreurs à l'arrivée : (a) sans, (b) avec commutation sur un régulateur PID*

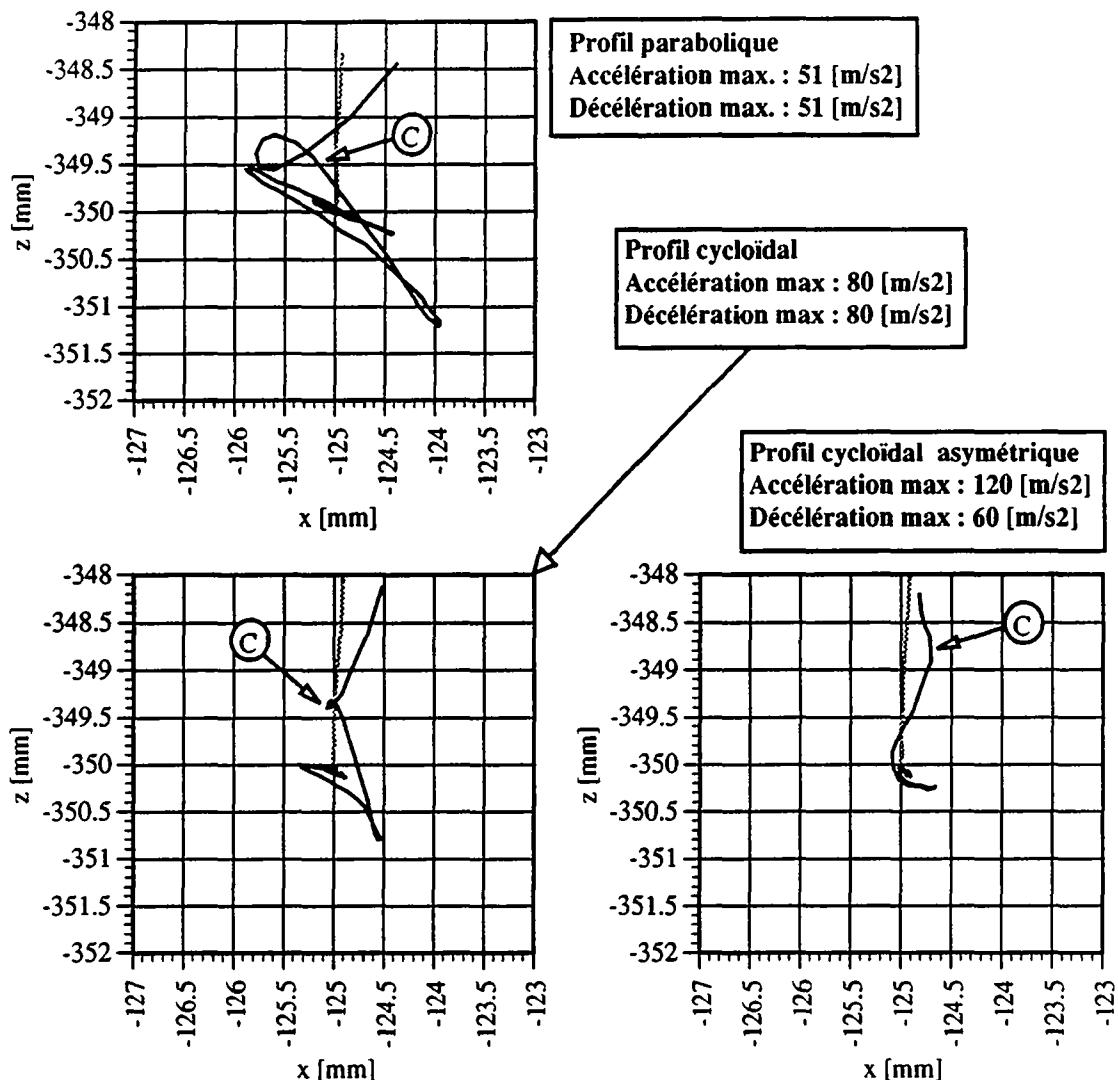
### 7.3. Effet de la trajectoire et de la loi de mouvement

Afin de montrer l'influence de la trajectoire et de la loi de mouvement sur la rapidité et la précision du robot, ce paragraphe se basera pour toutes les mesures sur le meilleur des régulateurs proposés précédemment, soit un régulateur PD avec commande a priori anticipée du couple pour la phase de mouvement et commutation sur un régulateur PID pour la phase de positionnement. L'influence de la loi de mouvement est tout d'abord montrée, puis celle de la trajectoire.

#### 7.3.1. Influence de la loi de mouvement

Pour mesurer cette influence, la trajectoire choisie est la même demi-ellipse que celle utilisée dans le paragraphe 7.2. La figure 7.3.1 montre le comportement du robot à l'arrivée pour des lois de mouvement parabolique symétrique, cycloïde symétrique et cycloïde asymétrique, qui donnent toutes le même temps de parcours  $t = 153$  [ms].

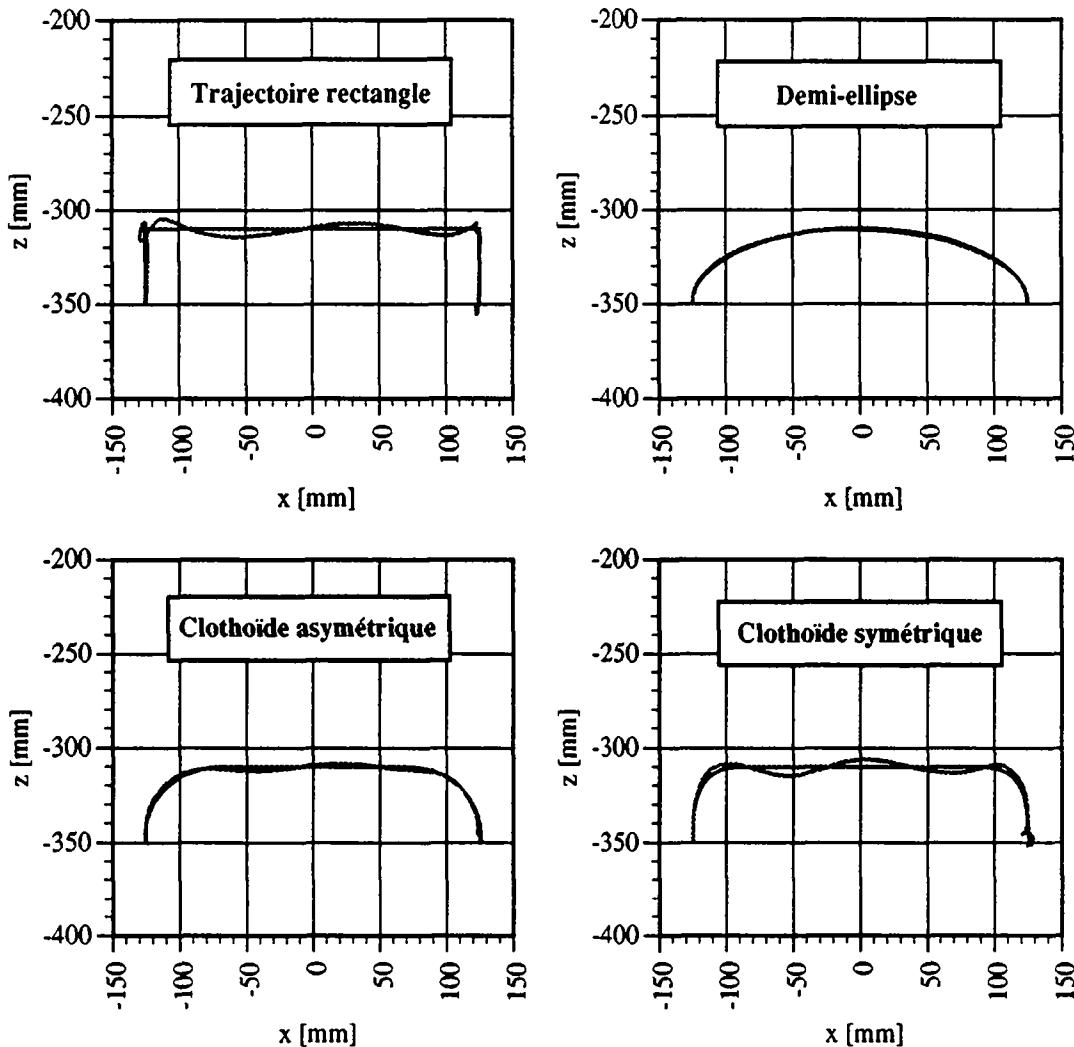
Ces courbes montrent que l'utilisation d'une loi de mouvement douce permet de diminuer les dépassemens à l'arrivée de façon significative. Le profil de mouvement parabolique est à éviter dans la mesure du possible, et une loi asymétrique montre un intérêt certain, le dépassement à l'arrivée devenant inférieur à 0.3 [mm]. Ceci illustre l'importance du profil lors d'un mouvement rapide et une étude plus approfondie devrait être réalisée en incorporant par exemple un freinage en plusieurs phases de façon à arriver en position sans dépassement. Toutefois, cette valeur satisfait déjà largement le cahier des charges fixé dans le paragraphe 2.6.3.



*Figure 7.3.1 : Arrivée en position pour différentes lois de mouvement sur trajectoire demi-ellipse (vue dans le plan de la trajectoire). © indique l'instant de commutation sur le régulateur PID.*

### 7.3.2. Influence de la trajectoire

Diverses trajectoires ont été proposées dans le chapitre 5 pour la prise et dépose d'objets. Dans ce paragraphe, les résultats obtenus sur le robot DELTA avec chacune d'entre elles sont exposés dans la figure 7.3.2. Toutes ces trajectoires ont un même temps de parcours  $t = 150$  [ms] et utilisent un profil de mouvement de type cycloïdal.



*Figure 7.3.2 : Trajectoires et leur suivi (projection dans le plan de la trajectoire)*

Les meilleurs résultats sont obtenus avec la demi-ellipse. Ceci se comprend aisément, car il s'agit de la trajectoire la plus courte en distance respectant le cahier des charges du paragraphe 2.6.3, ce qui donne les couples les plus faibles. Malheureusement, cette trajectoire ne permet pas la prise et dépose d'objets situés à différentes hauteurs. La clothoïde, développée dans ce but donne alors des résultats tout à fait satisfaisants à condition toutefois de ne pas avoir un rayon de courbure trop faible. Les couples à fournir deviennent à ce moment là vite importants. La différence de suivi de trajectoire entre la clothoïde symétrique et asymétrique est là pour le prouver. L'utilisation d'une trajectoire rectangulaire, même avec passage à vitesse nulle aux endroits de discontinuité (comme c'est le cas pour cette mesure), est à proscrire.

Finalement, si la demi-ellipse est intéressante pour ses performances, l'utilisateur obtient un confort de programmation beaucoup plus grand par l'utilisation des clothoïdes. Celles-ci lui offrent une liberté de mouvement qui a été un atout indéniable lors de la mise au point de programmes de démonstration.

## 7.4. Cadences

Afin de comparer le robot DELTA à entraînement direct avec les autres robots du marché, il est intéressant de déterminer les cadences possibles sur une trajectoire typique de palettisation conforme au cahier des charges du paragraphe 2.6.3. La cadence est définie par le nombre de trajets aller-retour possibles par seconde et est exprimée en [Hz]. Une trajectoire verticale est aussi étudiée car elle permet les cadences maximales du robot.

### 7.4.1. Trajectoire de prise et dépose

La trajectoire en demi-ellipse avec profil cycloïdal asymétrique, donnant les meilleurs résultats, est prise ici comme référence pour la détermination des cadences. Le régulateur utilisé est le régulateur (4) du paragraphe 7.2 avec commutation sur le régulateur PID de positionnement. Le tableau 7.4.1. regroupe les résultats obtenus pour différentes précisions désirées, en considérant un temps de prise ou dépose de 20 [ms].

Précision [mm]	Cadence [Hz]
0.5	3.05
0.1	2.32
0.03	1.89

*Tableau 7.4.1 : Cadences de palettisation pour le robot DELTA à entraînement direct*

On peut noter ici que pour obtenir une précision de 0.5 [mm], l'enclenchement du régulateur de positionnement n'est même pas nécessaire. Pour obtenir une précision de 0.03 [mm] (incrément du capteur), le temps consacré au mouvement est de 152 [ms]. Vient ensuite une phase de stabilisation d'environ 100 [ms]. Comme on peut le constater, la phase de stabilisation représente ici environ 40% du temps total. Ceci laisse encore un potentiel sérieux pour l'augmentation des cadences. Un meilleur

régulateur de positionnement et des lois de mouvement plus judicieuses pourraient être utilisés à cet effet.

#### 7.4.2. Trajectoire verticale

Les trajectoires verticales présentent l'avantage de générer des mouvements minimaux sur chacun des moteurs. Ceci permet d'exploiter pleinement le moteur sur tout le mouvement. Dans ces conditions, une trajectoire en ligne droite de 250 [mm] de longueur peut être parcourue avec des accélérations maximales de la nacelle jusqu'à 400 [m/s<sup>2</sup>] avec un profil cycloïdal. En réalisant des mouvements d'aller et retour en continu sur cette trajectoire, la cadence est de 5.7 [Hz]. Les erreurs de suivi de trajectoire sont inférieures à 3 [mm] et le dépassement à l'arrivée est d'environ 1 [mm].

### 7.5. Conclusion

Les résultats présentés dans ce chapitre ont permis de confirmer les hypothèses émises dans les différents chapitres de ce travail. Notamment, l'incorporation d'un modèle dynamique relativement précis dans l'algorithme de commande a permis de diminuer les erreurs de poursuite d'un facteur 5.7 environ. L'importance du choix de lois de mouvement douces comme la cycloïde a aussi été montrée, de même que celle de la trajectoire.

Finalement, un mouvement typique de prise et dépose entre deux points espacés de 250 [mm] peut être réalisé à des cadences proches de 3 [Hz] pour une précision de 0.5 [mm]. Comparé aux autres robots du marché, le robot DELTA à entraînement direct est donc 3 à 4 fois plus rapide. S'il est comparé avec une version industrielle du même robot DELTA [Clavel 89], mais avec un réducteur, les cadences sont deux fois supérieures. Ceci vérifie le bien-fondé de l'entraînement direct.

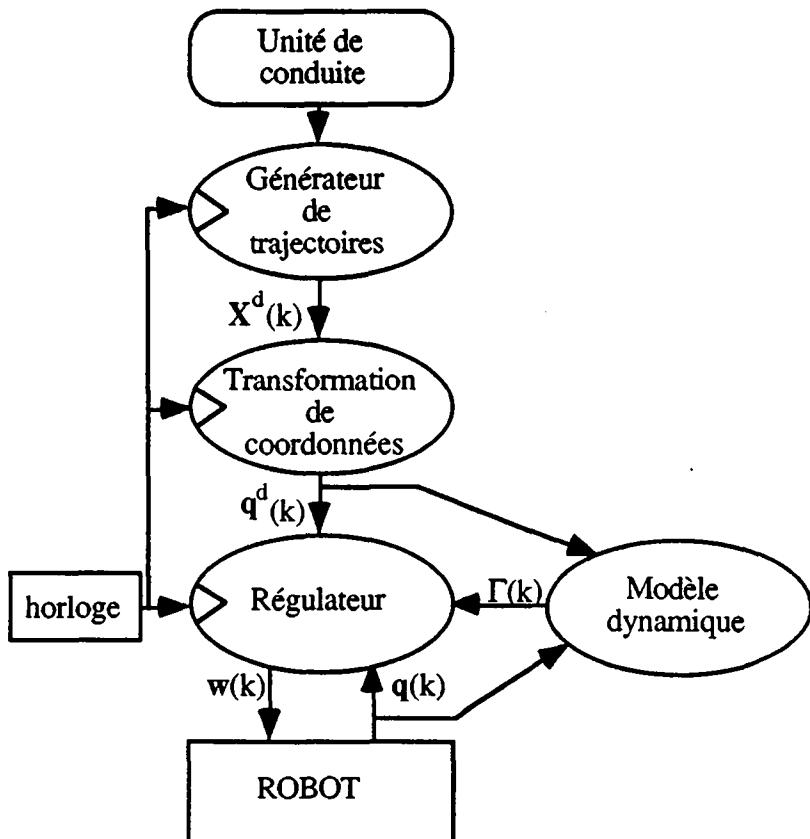
Pour une trajectoire verticale, particulièrement favorable au point de vue dynamique, le robot atteint même des cadences supérieures à 5.5 [Hz] lorsque l'on fixe des phases d'accélération et de décélération dont les pointes atteignent 400 [m/s<sup>2</sup>].



## 8. Architecture matérielle

### 8.1. Introduction

Le contrôleur de robot (cf. chapitre 4) doit réaliser un ensemble complexe d'opérations de façon rapide et souvent simultanée. La figure 8.1.1 rappelle brièvement les tâches principales qui apparaissent dans un contrôleur moderne tel qu'il est envisagé pour la commande du robot DELTA.



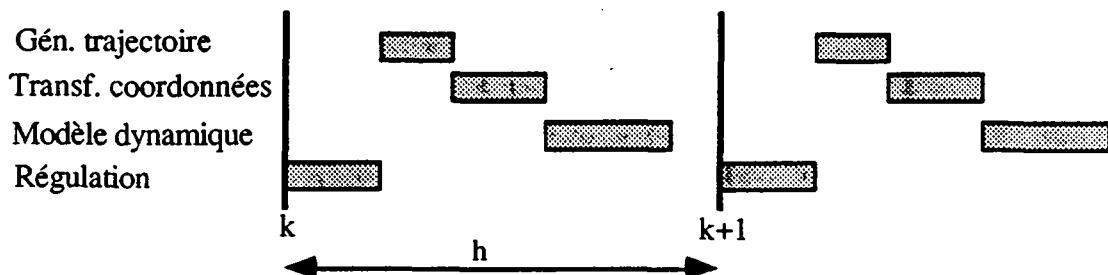
*Figure 8.1.1 : Tâches d'un contrôleur de robot moderne*

Le générateur de trajectoires fournit, selon les ordres de l'unité de conduite, les valeurs de consigne  $X^d(k)$  dans l'espace opérationnel. Celles-ci sont traduites en coordonnées  $q^d(k)$  de l'espace articulaire par le module de transformation de coordonnées. Le modèle dynamique, en fonction des valeurs articulaires réelles  $q(k)$

ou désirées  $q^d(k)$ , calcule les paramètres dynamiques  $\Gamma(k)$  et les fournit au régulateur. Ce dernier calcule alors la commande  $w(k)$  pour les actionneurs. Toute la commande est synchronisée par une horloge unique qui définit la période d'échantillonnage du système et toutes les tâches doivent être calculées pendant ce laps de temps. Il existe plusieurs possibilités de les planifier.

### 1) exécution séquentielle

Toutes les tâches sont effectuées séquentiellement dans l'ordre chronologique donné par la figure 8.1.1. Cependant, par principe on aime bien que la régulation s'effectue à intervalles réguliers. Pour ne pas dépendre de la durée de calcul variable des autres tâches, la régulation est la première opération effectuée au coup d'horloge. En procédant ainsi, la consigne calculée pendant la période  $[k, k+1[$  servira à l'instant  $k+1$ . La figure 8.1.2 illustre cette approche.



*Figure 8.1.2 : Exécution séquentielle*

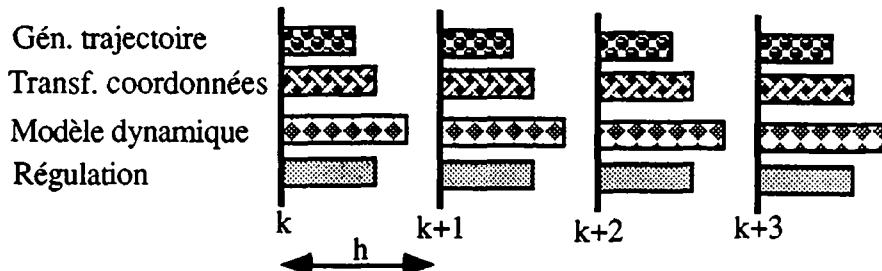
Dans ce cas, si l'on note  $T_i$  le temps d'exécution du processus  $p_i$ ,  $i=1..n$  et  $h$  la période d'échantillonnage, on aura obligatoirement :

$$h > \sum_i T_i$$

Par contre, un seul processeur est utilisé dans ce cas ( $N = 1$ ).

### 2) exécution parallèle

Dans ce cas, tous les processus (tâches) sont exécutés en parallèle sur plusieurs processeurs, comme le montre la figure 8.1.3.



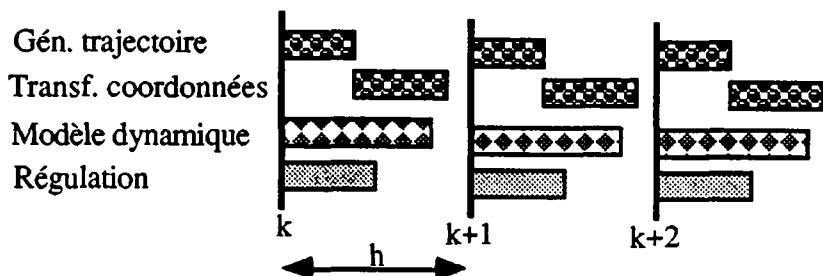
*Figure 8.1.3 : Exécution parallèle*

On notera que dans ce cas, le point de la trajectoire calculé en  $k$  servira au temps  $k+3$  seulement. Concernant le temps de calcul, et le nombre de processeurs utilisés, on obtient alors :

$$h > \max(T_i) \quad \text{et} \quad N = n$$

### 3) exécution mixte séquentielle/parallèle

Ici, toutes les combinaisons sont possibles, mais il est surtout intéressant de regrouper les tâches de façon à obtenir des temps de calcul comparables sur chaque processeur (figure 8.1.4). La connaissance du temps d'exécution de chaque processus est alors nécessaire.



*Figure 8.1.4 : Exécution mixte*

Il est évident que cette dernière solution optimise l'emploi du temps des processeurs et sera préférée à la seconde.

Le choix entre une solution ou l'autre dépendra de 3 facteurs :

- la période d'échantillonnage désirée
- la complexité des processus
- la puissance de calcul du (des) processeur(s) à disposition

Actuellement, avec des algorithmes de réglage de plus en plus sophistiqués, la somme impressionnante d'opérations à effectuer implique presque obligatoirement une architecture multiprocesseur permettant de répartir les différentes tâches.

Après une description des architectures classiques rencontrées le plus souvent dans les commandes du marché, ce chapitre sera consacré à la description d'une nouvelle architecture destinée au robot DELTA, basée sur le fameux transputer d'INMOS et permettant une implantation aisée de toutes les tâches de commande. Les avantages apportés par ce concept sont énormes: encombrement réduit, facilité d'extension, confort de programmation, ... . Ses performances seront finalement comparées avec la première commande du robot DELTA construite selon une architecture classique à base de microprocesseurs 680xx.

## 8.2. Architectures classiques

Etant donné le grand nombre d'opérations à effectuer dans une commande de robot, la solution mono-processeur a été très vite abandonnée, et des versions multiprocesseurs ont été développées. Les algorithmes de réglage simples utilisés considérant les axes comme indépendants entre eux, ont justifié la réalisation d'architectures de type "maître-esclave". Ce concept est d'ailleurs présent sur la majorité des contrôleurs de robot du marché. Généralement, le maître s'occupe de la gestion globale de la commande, alors que chacun des esclaves est indépendant et s'occupe de la conduite et de la régulation d'un seul axe. C'est dans cet esprit qu'a été développé à l'IMT la première commande du robot DELTA. La figure 8.2.1 montre sa composition.

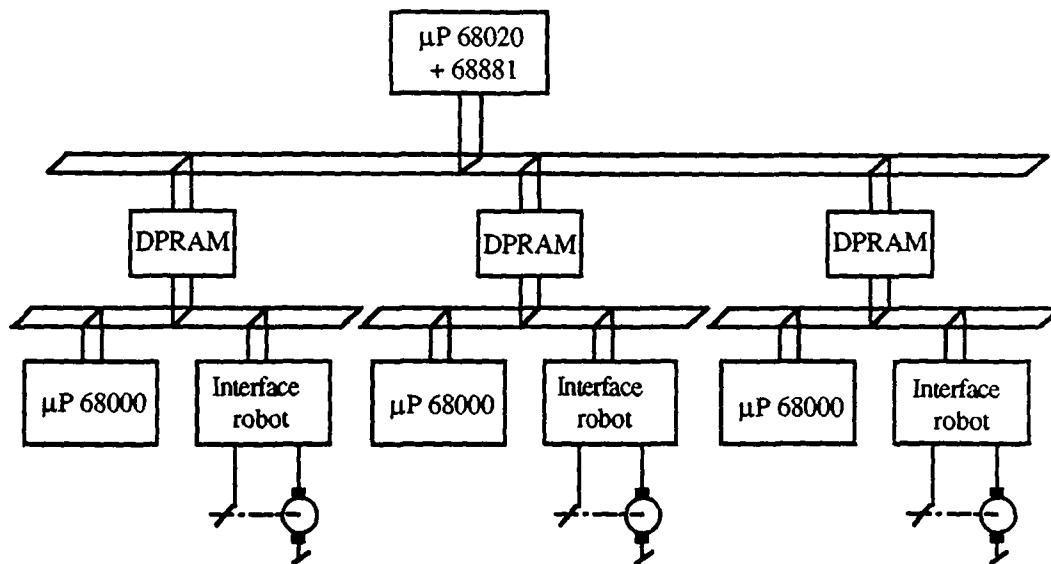


Figure 8.2.1 : Commande "maître-esclave" du robot DELTA basée sur microprocesseurs 68xxx

Le processeur maître (MC68020 + 68881) gère toute la commande et calcule la trajectoire, la transformation de coordonnée et le modèle dynamique du robot. Il communique ensuite les données nécessaires pour la régulation aux processeurs esclaves (MC68000) par l'intermédiaire de RAM à double accès (DPRAM). Chaque esclave effectue alors la régulation d'un axe. La synchronisation de tous les esclaves est assurée par une horloge unique provenant du maître et provoquant des interruptions à travers la DPRAM. La programmation de tout le système est réalisée en MODULA2. Bien qu'elle ait permis rapidement de réaliser les premiers essais, cette commande s'est vue rapidement dépassé par la somme des calculs à réaliser en temps réel (voir paragraphe 8.4). Son extension, pour ajouter un nouveau processeur, n'était pas simple et elle a été abandonnée au profit d'une architecture à base de "transputer" offrant une plus grande souplesse.

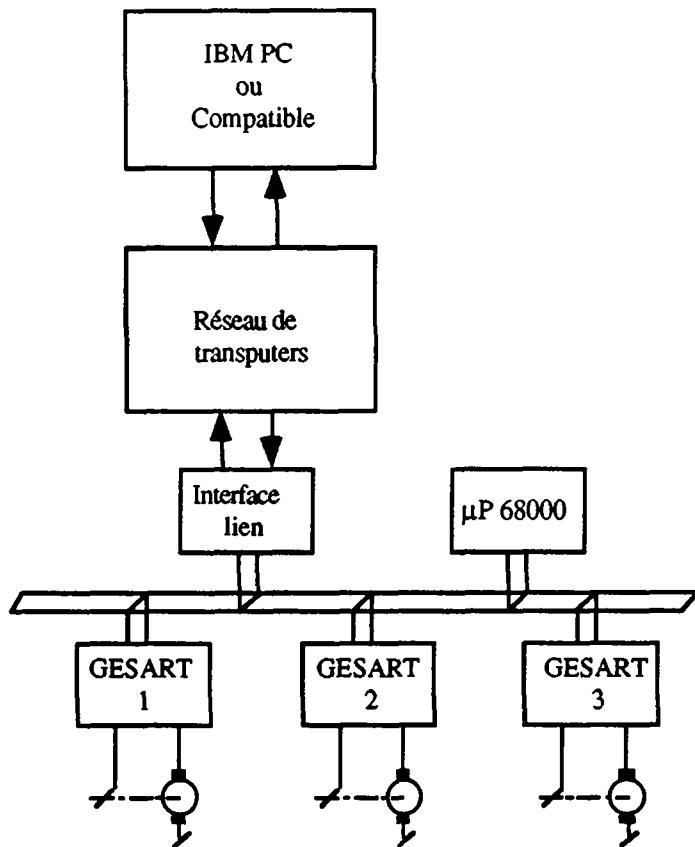
### 8.3. La solution à transputers

Le concept transputer développé en 1986 est assez particulier et a été pensé spécialement pour des applications nécessitant le parallélisme. L'originalité du transputer réside surtout dans la possibilité de pouvoir communiquer à des membres de la même famille par des lignes sérielles à très haut débit (10 à 20 Mbits/s). Ceci simplifie considérablement la conception de systèmes parallèles. Des réseaux de taille et de topologie arbitraires peuvent être construits selon l'application envisagée. La plupart des langages de haut niveau sont disponibles pour la programmation des transputers, mais jusqu'à présent, ils n'exploitaient pas complètement les possibilités de parallélisme inhérents au processeur. Le langage OCCAM a été conçu en même temps que le transputer et il reflète exactement son architecture matérielle. La programmation dans ce langage amènera alors à l'implémentation la plus efficace possible. C'est pour cette raison qu'il a été choisi pour le développement de la nouvelle commande. La description complète des possibilités des transputers et du langage OCCAM sortirait du cadre de ce travail. Le lecteur trouvera tous les détails qui l'intéressent dans la documentation [INMOS 89] ou dans [Thielen 90] qui en fait un bon résumé.

Dans les paragraphes qui suivent, nous allons plutôt montrer l'implémentation réalisée, puis la comparer avec une architecture classique.

#### 8.3.1. Implémentation matérielle de la commande

L'utilisation d'algorithmes de réglage modernes tenant compte du couplage entre les différents axes ne justifie plus une architecture de type "maître-esclave" et il est même préférable que l'information de chacun des axes puisse circuler librement à différents étages de la commande. Dans cette optique, le placement de tous les axes sur un même bus relié au réseau de transputer est intéressant. La figure 8.3.1 représente schématiquement l'architecture matérielle de la commande développée.



*Figure 8.3.1 : Architecture matérielle de la commande*

On y distingue essentiellement 3 niveaux :

*1) Un ordinateur PC-AT compatible IBM*

L'ordinateur sert principalement de console (écran + clavier) au système de développement pour transputer TDS et de serveur pour l'accès aux périphériques tels que disque dur, floppy, imprimante, etc...

*2) Un réseau de transputer*

Celui-ci est constitué d'un nombre variable de transputer en fonction de la complexité du problème et des performances recherchées. C'est cet étage qui calculera toute la commande du robot et nous verrons plus tard la topologie retenue pour la commande du robot DELTA.

### 3) Une interface avec les actionneurs et capteurs

Cette solution est basée sur le bus industriel G96 de Gespac, où l'on trouve :

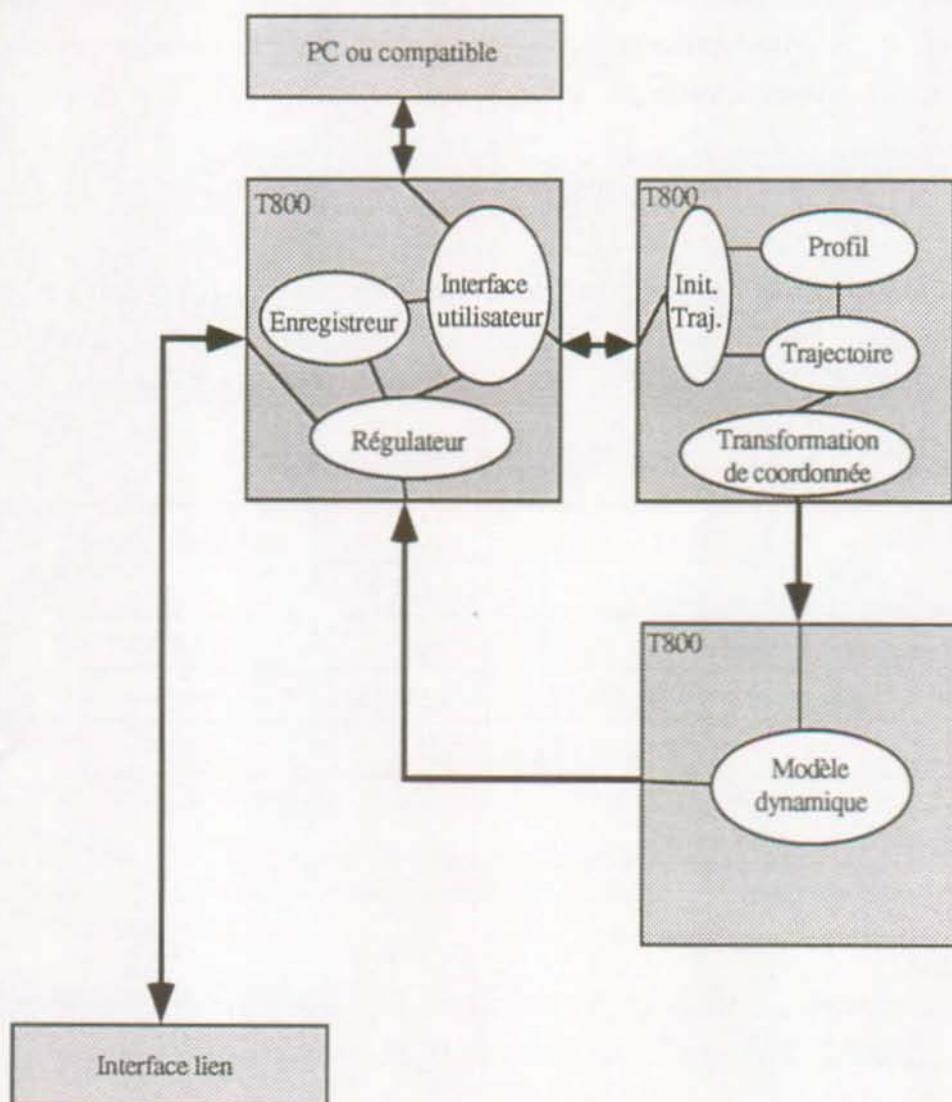
- 3 cartes GESART (développées à l'IMT) comportant principalement un compteur pour capteur incrémental et un convertisseur digital/analogique pour la commande d'un amplificateur.
- une interface lien (développée au LSP<sup>1</sup>) pour la connexion du bus G96 au réseau de transputer. Elle effectue principalement la conversion des données sérielles du lien en données parallèles pour le bus G96 et vice-versa.
- une unité processeur MC68000 GESMPU-4B (Gespac). Dans cette configuration, elle sert uniquement à la communication entre l'interface lien et les cartes GESART. Cette solution a été adoptée en fonction du matériel disponible. Une version future prévoit son remplacement par un transputer pouvant accéder directement au bus.

---

<sup>1</sup>Laboratoire de Systèmes Périphériques de l'EPFL

### 8.3.2. Implémentation logicielle

L'ensemble du logiciel de commande a été écrit en OCCAM2 avec le système de développement TDS. La figure 8.3.2 montre l'implémentation qui a été réalisée.



*Figure 8.3.2 : Implémentation logicielle*

Chaque bulle du schéma correspond à un processus OCCAM s'exécutant en parallèle avec les autres. Chaque ligne reliant les processus représente un canal de communication OCCAM. Grâce à la communication synchrone, tous les processus se synchronisent automatiquement sur le régulateur qui est exécuté à intervalles réguliers. L'évaluation des temps de calcul de chacun des processus (voir § 8.4) a permis leur implantation sur trois transputers T800 à 20 MHz en conservant une période d'échantillonnage de 1 [ms].

La plupart des processus représentés dans la figure 8.3.2 sont connus et ne seront pas décrits une nouvelle fois ici. Signalons toutefois qu'un "enregistreur" a été incorporé en parallèle avec le régulateur pour surveiller son activité et pouvoir mémoriser l'évolution temporelle de différentes valeurs (consigne, position réelle, erreur, etc...). L'interface utilisateur permet ensuite de les stocker sur disque pour une représentation graphique à l'aide de logiciels adaptés (MathCAD ou autre...).

#### 8.4. Evaluation des performances

Pour avoir une idée des performances de la commande développée, il a paru intéressant de la comparer avec la commande classique à base de 68xxx développée précédemment. Les temps de calcul des différents processus sont reportés dans le tableau 8.4.1.

Calcul	durée [ $\mu$ s]		gain
	T800	68020+68881	
Trajectoire <sup>2</sup>	149	486	3.26
Transf. coord.	244	859	3.52
Dynamique	704	2324	3.30

*Tableau 8.4.1 : Evaluation des temps de calcul*

De manière globale, l'utilisation des transputers T800 à 20 MHz a permis de tripler les performances de la commande. Ce résultat s'explique par 3 raisons :

- 1) l'horloge du transputer est légèrement supérieure (20 [MHz] contre 16 [MHz])
- 2) le T800 à lui-seul est plus puissant que la paire 68020 + 68881
- 3) OCCAM est plus proche du transputer que MODULA2 ne l'est du 68020/68881

<sup>2</sup>Demi-ellipse avec profil cycloïdal

Bien que le temps de calcul soit diminué sur le système transputer, son atout principal n'est pas dans ses performances de calcul (il existe des processeurs nettement plus puissants), mais plutôt dans l'intégration du processus de communication et du parallélisme au sein du processeur. Ceci donne une vision claire du flot des informations et de leur traitement, rendant le programme très transparent. Le parallélisme et les horloges temps réel faisant partie du langage OCCAM, on s'épargne des détours par des routines d'interruption et autres artifices.

Parmi les avantages du transputer, on peut encore citer :

- un encombrement réduit, grâce à l'incorporation de l'unité virgule flottante à l'intérieur du processeur
- l'élimination du bus souvent gourmand en composants sur les cartes processeurs classiques.
- des possibilités d'extension "à l'infini", grâce à la notion de lien qui permet de connecter d'autres transputers aisément.

## 8.5. Conclusion

Il a été montré dans ce chapitre, que le concept transputer était apte à résoudre les problèmes liés à la commande de robot. Par des concepts simples, on évite une bonne partie des tracas inhérents à la programmation parallèle en temps réel. L'aspect le plus important dans la solution transputer est son caractère modulaire. On peut imaginer sans peine que la commande réalisée ici soit associée à une unité de conduite "intelligente" ou un système de vision, développés sur transputers par des tiers. Grâce aux liens standards, l'interconnexion de ces différents modules ne poserait pas de problème majeur. Les commandes de robot du futur ne devraient pas laisser passer cette chance, le bénéfice qu'elles peuvent en tirer est important.



## 9. Conclusion

### 9.1. Résumé et contributions

Les robots actuellement sur le marché sont souvent critiqués pour leur lenteur. La réalisation d'un robot rapide et précis apparaît alors comme un défi. Concilier rapidité et précision n'est cependant pas simple dans la mesure où ces deux notions sont antagonistes. Pour y parvenir, il semblait important de réunir trois conditions :

- 1) Avoir un robot dont la structure en mouvement est la plus légère possible.
- 2) Utiliser un système d'entraînement rigide et sans jeu.
- 3) Réaliser une commande très performante tenant compte des effets dynamiques de la structure mobile.

La réalisation d'un entraînement direct pour le robot parallèle DELTA a permis de résoudre les deux premiers problèmes simultanément. La structure de ce robot, avec ses moteurs fixes sur le bâti est d'ailleurs très favorable à l'entraînement direct, car l'augmentation de la taille des moteurs ne modifie pas la masse en mouvement.

Ce travail a principalement traité l'étude et la réalisation d'une commande performante permettant de concilier rapidité et précision. Dans ce but, quatre points ont été développés plus particulièrement et forment les contributions essentielles de cette recherche :

- Etablissement d'un modèle dynamique simplifié du robot DELTA pour le calcul en temps réel.
- Implantation d'un régulateur utilisant ce modèle dynamique comme commande a priori.
- Etude et choix de trajectoires compatibles avec la dynamique du système; introduction de la notion de "variation maximale de couple".

- Réalisation d'une architecture de commande parallèle multiprocesseurs à base de transputers.

En raison de l'interdépendance des variables articulaires, l'établissement du modèle dynamique d'un robot parallèle n'est pas simple et amène souvent à des équations complexes incompatibles avec les exigences du temps réel. Par des hypothèses judicieusement posées, le modèle dynamique du robot DELTA devient simple tout en restant très proche de la réalité. Son calcul est possible en 750 [ $\mu$ s] sur transputer T800. Ceci permet son utilisation dans la boucle de réglage, comme commande a priori, à chaque période d'échantillonnage. Ainsi, l'erreur de suivi de trajectoire est diminuée d'un facteur 6 environ par rapport à un régulateur PD classique utilisé sur la plupart des commandes de robots du marché. Pour éliminer les erreurs de statisme à l'arrivée, un régulateur PID est enclenché dans une deuxième phase.

Pour effectuer des mouvements rapides, un bon régulateur ne suffit pas. Il est nécessaire aussi, de générer des trajectoires dont le temps de parcours est minimal, mais qui respectent les contraintes dynamiques du système. Deux critères liés au système d'entraînement, le couple maximal, et la variation maximale de couple ont été choisis pour sélectionner les trajectoires et les lois de mouvement. Il en résulte, selon le deuxième critère, que toute discontinuité de l'accélération doit être éliminée, autant pour l'accélération tangentielle (loi de mouvement) que pour l'accélération transverse (courbure de la trajectoire). La clohoïde avec loi de mouvement cycloïdale, répondant à ces 2 contraintes, est utilisée à cet effet. Entre deux points espacés de 250 [mm], une trajectoire "demi-ellipse" avec une loi de mouvement de type cycloïde permet des cadences de prise et dépose supérieures à 3 [Hz] pour une précision de 0.5 [mm]. Pour un mouvement vertical, particulièrement favorable à la dynamique du robot, des accélérations de pointe jusqu'à 400 [m/s<sup>2</sup>] ont été atteintes. Dans cette configuration, il est alors possible d'obtenir des cadences supérieures à 5.5 [Hz]. A notre connaissance, le robot DELTA à entraînement direct est actuellement le robot le plus rapide au monde !

L'électronique de commande développée est organisée sur un réseau de trois transputers connectés entre eux selon une architecture parallèle de type "pipe-line". Un premier processeur calcule la trajectoire et la transformation de coordonnée du robot. Il communique ensuite ses résultats à un deuxième processeur qui calcule le modèle dynamique du robot. Le troisième s'occupe principalement de la tâche de régulation.

## 9.2. Perspectives d'avenir

Par ses performances et son coût relativement faible, le robot DELTA à entraînement direct possède de sérieux atouts. Pour exploiter pleinement son potentiel, notamment dans le domaine de l'assemblage, des recherches que l'on peut résumer en 6 points doivent encore être menées :

- 1) Mise au point de moteurs adaptés à la robotique à entraînement direct (couple élevé à faible vitesse (200 tours/min.), meilleure linéarité).

Ce projet a mis en évidence l'inexistence sur le marché actuel de moteurs adaptés au DELTA à entraînement direct. Le prototype développé utilise des moteurs sous-utilisés en puissance. Le développement de nouveaux moteurs adaptés à l'entraînement direct sera importante lors de la phase d'industrialisation du robot.

- 2) Maîtrise de la précision absolue du robot (en vue de la génération automatique de programmes à partir des données topographiques du montage).

La mesure de la position du robot DELTA est réalisée par des capteurs situés directement sur le rotor du moteur. La position absolue du préhenseur est ensuite calculée par la transformation de coordonnée directe du robot. Celle-ci ne tient généralement pas compte des tolérances de fabrication des différents éléments du robot, ce qui introduit une erreur entre la position calculée et la position réelle du préhenseur par rapport à un repère absolu. Pour générer automatiquement des programmes d'assemblage, il est absolument nécessaire de connaître avec précision la position du préhenseur. Différentes solutions sont proposées dans la littérature pour corriger ces erreurs [Payannet 85]. La solution la plus courante consiste en un calibrage du robot par rapport à des positions absolues judicieusement choisies. Dans le cas de robots parallèles, comme le DELTA, le problème est plus difficile en raison de l'interdépendance des variables articulaires, et à notre connaissance, aucune méthode n'a été développée dans ce cas. Vu l'importance de ce point dans le cadre de l'assemblage flexible, il nous paraît intéressant de nous en préoccuper.

- 3) Développement de préhenseurs légers (< 50 [g]) et rapides (constante de temps inférieure à 20 [ms]).

Dès le moment où l'on désire augmenter les cadences de travail d'un robot, le temps consacré à la prise et à la dépose d'un objet doit être le plus petit possible pour ne pas pénaliser les efforts entrepris dans la minimisation du temps de trajet. Pour des cadences de 3[Hz], le temps à disposition pour cette manœuvre est inférieur à 20 [ms].

4) Génération de trajectoires "optimales" en temps et compatibles avec les exigences de la dynamique du robot.

Les trajectoires et lois de mouvement proposées dans ce travail sont une première étude du problème. Une approche prenant en compte l'interdépendance trajectoire-dynamique serait cependant encore nécessaire pour affiner les choix.

5) Etude de la modélisation dynamique des robots parallèles.

Le modèle dynamique du robot DELTA développé dans le cadre de ce travail est déjà très proche de la réalité. Cependant, vu l'importance de celui-ci pour l'amélioration des performances de la régulation, un modèle plus complet et plus précis pourrait être utile. Etant donnée la difficulté d'établissement d'un modèle précis par les méthodes traditionnelles, une méthode se basant sur une phase d'apprentissage sur le robot réel pourrait être envisagée.

6) Développement de régulateurs à dynamique rapide capables de s'adapter aux variations des paramètres du système.

Des techniques pour améliorer la vitesse de convergence sur le point d'arrivée doivent encore être développées en utilisant par exemple un régulateur d'état ou de type RST avec estimateur de perturbations. D'autre part, une adaptation automatique du modèle à la masse transportée serait souhaitable.

La maîtrise de tous ces points permettrait certainement l'extension des applications du robot DELTA, comme par exemple :

- assemblages microtechniques à cadences élevées
- dépôt de colle à haute vitesse
- découpe de tissus ou tôles minces par jet d'eau où par laser
- et à plus long terme, réalisation d'une machine à mesurer basée sur la structure DELTA.

Ce travail représente un premier pas dans le développement de robots rapides et précis. Espérons qu'il puisse motiver d'autres recherches dans ce domaine afin que l'on voie un jour dans l'industrie des robots pouvant concurrencer les chaînes de montage rigides !

## **Annexes**



## A. Transformation de coordonnées du robot DELTA pour le calcul en temps réel

### A.1. Introduction

L'établissement du modèle géométrique inverse d'un robot est indispensable lorsque l'on désire contrôler sa trajectoire dans l'espace opérationnel. Pour le robot DELTA, (à cause de sa structure géométrique particulière) le contrôle de la trajectoire est même obligatoire si l'on veut éviter des collisions avec des objets situés dans son volume de travail. Deux modèles différents ont déjà été développés pour le robot DELTA. Le premier [Sternheim 87], utilisé pour la représentation graphique dans un logiciel de simulation, calcule tous les angles intermédiaires, ce qui ralentit son temps de calcul. Le deuxième [Clavel 89], permet un calcul direct, sans passer par les angles intermédiaires.

Le modèle géométrique est généralement calculé en temps réel, par un microprocesseur associé ou non à un coprocesseur mathématique. Les calculs sont effectués la plupart du temps en virgule flottante, sur 32 ou 64 bits. Ainsi, la précision de calcul est limitée et des erreurs importantes peuvent apparaître lorsque les valeurs tendent vers l'infini ou vers zéro. La manipulation de ces nombres doit se faire avec une grande prudence et les équations mathématiques pouvant amener à ces erreurs doivent être contrôlées ou éliminées. Les divisions par des nombres proche de zéro ou les fonctions tendant vers l'infini ( $\text{tg}, \dots$ ) sont les plus grandes sources d'erreurs ! Les deux modèles développés pour le robot DELTA sont sujet à de telles singularités mathématiques, notamment en raison d'une indétermination de type "zéro sur zéro" pour le modèle inverse.

Cette annexe décrit le modèle géométrique du robot DELTA pour son calcul en temps réel et propose une correction du modèle inverse permettant d'éliminer les singularités mathématiques des autres modèles. Le modèle direct est tiré de [Clavel 89].

## A.2. Singularités mathématiques des modèles géométriques inverse du robot DELTA

### A.2.1. Modèle de Sternheim

Le premier modèle géométrique du robot DELTA a été établi par [Sternheim 87]. Pour les besoins de la simulation, ce modèle calcule les angles intermédiaires au niveau du coude.

Le modèle géométrique inverse présente une singularité mathématique de type "zéro sur zéro" dans le volume de travail du robot. En effet, on a

$$B_{1x} = X_p - R$$

et donc

$$B_{1x} = 0 \quad \text{si} \quad X_p = R$$

Dans [Sternheim 87], l'équation (11)  $\Theta = \arctg\left(\frac{B_{1x}}{-B_{1z}}\right)$  passe par zéro pour  $B_{1x} = 0$  ou pour  $X_p = R$  !

L'équation (13) comporte une expression de type  $\frac{B_{1x}}{\sin\Theta}$  qui passe donc par zéro au numérateur et dénominateur en même temps !

Ainsi, ce modèle comporte une infinité de singularités mathématiques situées dans 3 plans verticaux  $X_p = R$  perpendiculaires aux axes  $x_i$  des référentiels  $R_i$  liés aux 3 moteurs<sup>1</sup>.

### A.2.2. Modèle de Clavel

Le modèle de [Clavel 89], basé sur l'intersection d'une sphère et d'un cercle a l'avantage de se calculer plus rapidement que celui de Sternheim, ce qui n'est pas négligeable pour le calcul en temps réel.

Lors de l'établissement du modèle inverse, on tombe sur une équation du type :

$$a \cos\alpha_i + b \sin\alpha_i = c \quad [A.2.1]$$

---

<sup>1</sup>Pour la définition des référentiels et des paramètres géométriques se rapporter au § 3.3.1.

avec       $a = x - R$

$b = -z$

$$c = \frac{1}{2L_A} (-L_B^2 + L_A^2 + (x - R)^2 + y^2 + z^2)$$

où  $x, y$  et  $z$  sont les coordonnées de la nacelle dans chacun des repères  $R_i$ .

La résolution de l'équation [A.2.1] se fait en passant par la tangente demi, ce qui amène à

$$\alpha_i = 2 \cdot \text{arctg} \left( \frac{-b \pm \sqrt{b^2 + a^2 - c^2}}{a + c} \right) \quad [\text{A.2.2}]$$

On peut calculer à quel moment  $a+c$  (dénominateur) passe par zéro. En posant  $a+c = 0$ , on a :

$$(x - R) + \frac{1}{2L_A} (-L_B^2 + L_A^2 + (x - R)^2 + y^2 + z^2) = 0 \quad [\text{A.2.3}]$$

En développant, on aboutit sur :

$$[x - (R - L_A)]^2 + y^2 + z^2 = L_B^2 \quad [\text{A.2.4}]$$

On voit donc que le dénominateur de [A.2.2] passe par zéro pour tous les points situés sur une sphère de centre  $\begin{pmatrix} R - L_A \\ 0 \\ 0 \end{pmatrix}$  et de rayon  $L_B$ , ceci dans chacun des référentiels  $R_i$ . Il y a donc 3 sphères au total, une pour chacun des  $\alpha_i$ .

De plus, on peut montrer que le numérateur de l'équation [A.2.2] passe par zéro au même moment. On a donc une singularité mathématique de type "zéro sur zéro".

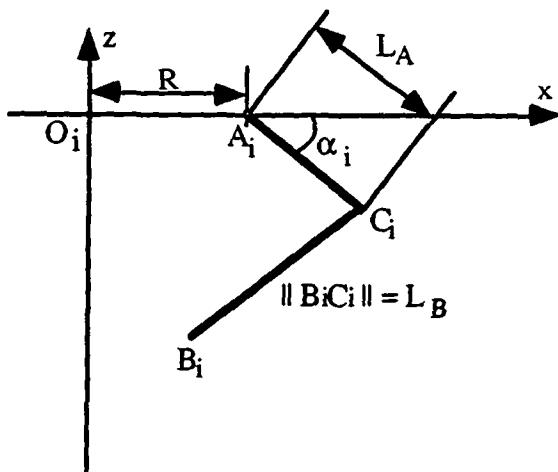
### A.3. Nouveau modèle éliminant les singularités mathématiques du modèle inverse

Ce modèle est basé sur les mêmes prémisses que celui de [Clavel 89].

Le but du modèle géométrique inverse est de calculer les angles  $(\alpha_1, \alpha_2, \alpha_3)$  en connaissant la position  $(x_0, y_0, z_0)$  de la nacelle. En travaillant dans les repères  $R_i$ , on peut calculer chacun des angles  $\alpha$  indépendamment. Dans le repère  $R_i$ , la position de la nacelle est donnée par la relation :

$$\begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix}_i = \begin{pmatrix} \cos\Theta & \sin\Theta & 0 \\ -\sin\Theta & \cos\Theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_0 \\ y_0 \\ z_0 \end{pmatrix}$$

Les avant-bras décrivent depuis la position  $B_i = (x_0, y_0, z_0)$  de la nacelle dans le repère  $R_i$  une sphère de rayon  $L_B$ , ce qui nous donne l'équation [A.3.1]. Le bras, centré en  $(R, 0, 0)$  décrit un cercle de rayon  $L_A$  (équation [A.3.2]). Le point d'intersection entre la sphère [A.3.1] et le cercle [A.3.2] se trouve dans le plan  $y = 0$  (équation [A.3.3]).



*Figure A.1 : bras et avant-bras dans le repère  $R_i$*

On obtient ainsi un système de 3 équations à 3 inconnues permettant de calculer les coordonnées du point  $C_i$  (figure A.1) :

$$(x - x_0)^2 + (y - y_0)^2 + (z - z_0)^2 = L_B^2 \quad [A.3.1]$$

$$(x - R)^2 + z^2 = L_A^2 \quad [A.3.2]$$

$$y = 0 \quad [A.3.3]$$

On peut résoudre ce système d'équation en remplaçant [A.3.3] dans [A.3.1], puis en effectuant la soustraction de cette nouvelle équation avec [A.3.2]. Ceci nous donne :

$$x(2x_0 - 2R) + 2z_0 z = L_A^2 - L_B^2 - R^2 + x_0^2 + y_0^2 + z_0^2 \quad [A.3.4]$$

En posant :  $A = L_A^2 - L_B^2 - R^2 + x_0^2 + y_0^2 + z_0^2$

et  $B = 2x_0 - 2R$ ,

on trouve :

$$z = \frac{A - Bx}{2z_0} \quad [A.3.5]$$

En remplaçant dans [A.3.2], on trouve, après quelques opérations :

$$x = \frac{b \pm \sqrt{b^2 - ac}}{a} \quad [A.3.6]$$

avec :  $a = 4z_0^2 + B^2$

$$b = 4Rz_0^2 + AB$$

$$c = A^2 + 4R^2z_0^2 - 4z_0^2L_A^2$$

On vérifie que  $a$  ne passe jamais par zéro. En effet, si l'on veut rester dans le volume de travail,  $z_0$  n'est jamais égal à zéro, et ainsi,  $a$  est toujours positif !

Les équations [A.3.6] et [A.3.5] nous donnent le point d'intersection de la sphère avec le cercle, c'est-à-dire la position du coude  $C_i$  dans le repère  $R_i$ . Dans l'équation [A.3.6] seule la solution avec le signe positif est retenue, car elle seule donne un résultat compris dans le volume de travail du robot.

Pour éviter les singularités, on calculera l'angle  $\alpha$  avec  $\arcsin$  plutôt qu'avec  $\arctg$ . Ainsi, on trouve pour  $\alpha$  (défini positif vers le bas) :

$$\alpha_i = -\arcsin\left(\frac{z}{L_A}\right) \quad [A.3.7]$$

Un problème apparaît cependant avec cette formulation. En effet, si l'angle  $\alpha_i$  plus grand que  $90^\circ$ , l'équation [A.3.7] va nous donner une solution fausse. Il est donc nécessaire de tester encore  $x$  (équation [A.3.6]).

Si  $x-R \geq 0$  on garde le résultat de l'équation [A.3.7],  
et si  $x-R < 0$  on corrige  $\alpha_i$  comme suit :  $\alpha_i := \pi - \alpha_i$ .

#### A.4. Modèle géométrique direct

Le modèle géométrique direct implémenté dans la commande du robot DELTA est similaire à celui développé dans [Clavel 89] et est rappelé ici à titre indicatif.

En posant :

$$D_i = -L_B^2 + L_A^2 + R^2 + 2RL_A \cos \alpha_i$$

$$E_i = 2(R + L_A \cos \alpha_i) \cos \theta_i$$

$$F_i = 2(R + L_A \cos \alpha_i) \sin \theta_i$$

$$G_i = 2L_A \sin \alpha_i$$

$$H_1 = E_1 G_2 - E_1 G_3 - E_2 G_1 + E_2 G_3 + E_3 G_1 - E_3 G_2$$

$$H_2 = -E_1 F_2 + E_1 F_3 + E_2 F_1 - E_2 F_3 - E_3 F_1 + E_3 F_2$$

$$H_3 = -E_1 D_2 + E_1 D_3 + E_2 D_1 - E_2 D_3 - E_3 D_1 + E_3 D_2$$

$$H_4 = F_1 D_2 - F_1 D_3 - F_2 D_1 + F_2 D_3 + F_3 D_1 - F_3 D_2$$

$$H_5 = -F_1 G_2 + F_1 G_3 + F_2 G_1 - F_2 G_3 - F_3 G_1 + F_3 G_2$$

alors :

$$x = z \frac{H_5}{H_2} + \frac{H_4}{H_2} \quad [A.4.1]$$

$$y = z \frac{H_1}{H_2} + \frac{H_3}{H_2} \quad [A.4.2]$$

et :

$$z = \frac{-M \pm \sqrt{M^2 - 4LQ}}{2L} \quad [A.4.3]$$

avec :

$$L = \frac{H_5^2 + H_1^2}{H_2^2} + 1$$

$$M = 2 \frac{H_5 H_4 + H_1 H_3}{H_2^2} \cdot \frac{H_5 E_1 + H_1 F_1}{H_2} \cdot G_1$$

$$Q = \frac{H_4^2 + H_3^2}{H_2^2} \cdot \frac{H_4 E_1 + H_3 F_1}{H_2} \cdot G_1$$

où l'on choisit uniquement la solution avec le signe moins dans l'équation [A.4.3].

Une optimisation du nombre des opérations a cependant été entreprise en éliminant les multiplications par 0 ou 1, en éliminant les multiplications par 1, en regroupant les termes communs et en favorisant les additions par rapport aux multiplications lorsque cela était possible. Ainsi, tous les termes  $F_1$  s'annulent car  $\Theta_1 = 0$ , et la multiplication par  $\cos \Theta_1$  dans  $E_1$  peut être éliminée pour la même raison.

### A.5. Evaluation des temps de calcul

On va donner, dans ce paragraphe, les temps de calcul des modèles géométriques présentés ici.

Le nombre des opérations pour les deux modèles ainsi que le temps de calcul sur transputer T800 sont résumés dans le tableau [A.2.]

Opération	Modèle inverse	Modèle direct
+	21	25
-	11	19
x	43	54
÷	6	2
arcsin	3	-
sqrt	3	1
sin	-	3
cos	-	3
temps de calcul [ $\mu$ s]	244	253

*Tableau A.2. : opérations et temps de calcul des modèles géométriques du DELTA*

On remarque ici que les temps de calcul sont suffisamment faibles pour une utilisation en temps réel.

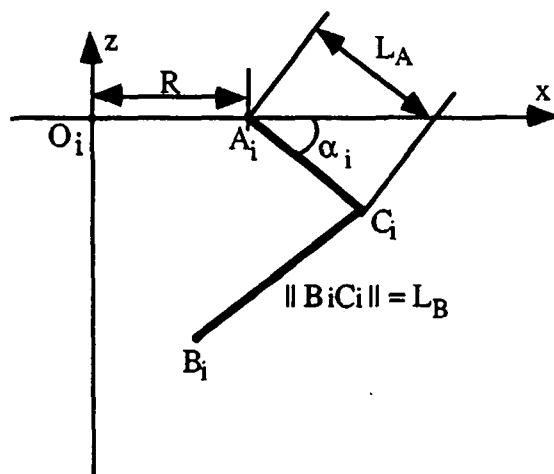
### A.6. Conclusion

Cette annexe a présenté les différents modèles géométriques implémentés sur la commande du robot DELTA. Ils sont inspirés de [Clavel 89], mais une nouvelle méthode est développée pour le modèle inverse afin d'éliminer ses singularités mathématiques. Le nombre des opérations à effectuer pour les deux modèles est minimisé, ce qui permet de les calculer raisonnablement en temps réel.

## B. Equations du modèle dynamique

### B.1. Modèle dynamique simplifié du robot DELTA

En tenant compte des hypothèses simplificatrices du paragraphe 3.3.2, on peut établir le modèle dynamique du robot DELTA d'une façon simple. Chaque parallélogramme peut être remplacé par une tige unique sans masse comme représenté dans la figure B.1.



*Figure B.1.: Chaîne cinématique dans le repère  $R_i$*

#### Vecteurs directeurs dans le repère $R_i$

- Vecteur directeur du bras

$$\mathbf{A}_i \mathbf{C}_i = \begin{pmatrix} L_A \cdot \cos \alpha_i \\ 0 \\ -L_A \cdot \sin \alpha_i \end{pmatrix} \quad [B.1.1]$$

- Vecteur directeur de l'avant-bras

$$\mathbf{B}_i \mathbf{C}_i = \mathbf{B}_i \mathbf{O}_i + \mathbf{O}_i \mathbf{A}_i + \mathbf{A}_i \mathbf{C}_i$$

$$\mathbf{B}_i \mathbf{C}_i = - \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} + \begin{pmatrix} R \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} L_A \cdot \cos \alpha_i \\ 0 \\ -L_A \cdot \sin \alpha_i \end{pmatrix} \quad [B.1.2]$$

où  $(x_i \ y_i \ z_i)^T$  est le vecteur exprimant les coordonnées de la nacelle dans le repère  $R_i$  soit :

$$\begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix} = \begin{pmatrix} \cos \theta_i & \sin \theta_i & 0 \\ -\sin \theta_i & \cos \theta_i & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} \quad [B.1.3]$$

On obtient donc pour le vecteur  $\mathbf{B}_i \mathbf{C}_i$  dans le repère  $R_i$  :

$${}^R \mathbf{B}_i \mathbf{C}_i = \begin{pmatrix} -x \cdot \cos \theta_i - y \cdot \sin \theta_i + R + L_A \cdot \cos \alpha_i \\ x \cdot \sin \theta_i - y \cdot \cos \theta_i \\ -z - L_A \cdot \sin \alpha_i \end{pmatrix} \quad [B.1.4]$$

### Vecteur directeur des avant-bras dans le repère R

Dans le repère R, le vecteur directeur de chacun des avant-bras est donné par :

$${}^R \mathbf{B}_i \mathbf{C}_i = \begin{pmatrix} (L_A \cdot \cos \alpha_i + R) \cdot \cos \theta_i - x \\ (L_A \cdot \cos \alpha_i + R) \cdot \sin \theta_i - y \\ -L_A \cdot \sin \alpha_i - z \end{pmatrix} \quad [B.1.5]$$

### Application de la loi de Newton sur la nacelle et les avant-bras

On isole tout d'abord la nacelle en lui ajoutant les forces de liaison  $\mathbf{F}_i$  avec chacun des trois avant-bras (figure B.2). En y applicant l'équation de Newton dans le référentiel R, on obtient alors :

$$-\sum_{i=1}^3 \mathbf{F}_i + \mathbf{G}_n = m_n \cdot \dot{\mathbf{X}} \quad [B.1.6]$$

avec :

- $F_i$  forces de liaison entre la nacelle et les avant-bras, exprimées dans le référentiel R.
- $G_n$  force de gravité agissant sur la nacelle.
- $m_n$  masse totale de la nacelle, y compris la masse transportée et la masse des avant-bras rapportée sur la nacelle.
- $\ddot{X}$  accélération de la nacelle exprimée dans le repère R.

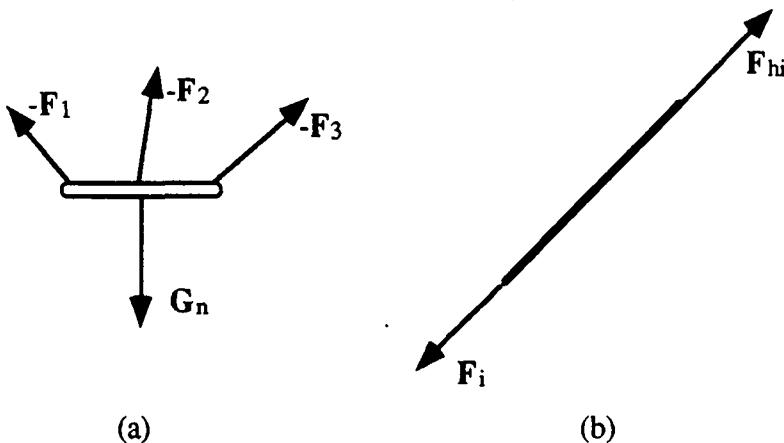


Figure B.2 : isolation de la nacelle (a) et d'un avant-bras (b)

L'isolation de l'avant-bras dont on considère la masse et l'inertie nulle nous donne alors (figure B.2.b) :

$$\vec{F}_i + \vec{F}_{hi} = \vec{0} \quad [B.1.7]$$

Les deux forces sont donc opposées et dans la direction  $B_iC_i$  de l'avant-bras.

En connaissant la direction des forces  $F_i$  il est alors possible de calculer leur norme à partir de l'équation [B.1.6]. En effet, en posant :

$$F_i = F_i \cdot \frac{B_i C_i}{\|B_i C_i\|} \quad \text{ou} \quad F_i = F_i \cdot \frac{B_i C_i}{L_B} \quad [B.1.8]$$

on obtient un système de 3 équations dont les 3 inconnues sont  $F_1$ ,  $F_2$  et  $F_3$ .

$$\begin{pmatrix} v_{x1} & v_{x2} & v_{x3} \\ v_{y1} & v_{y2} & v_{y3} \\ v_{z1} & v_{z2} & v_{z3} \end{pmatrix} \cdot \begin{pmatrix} F_1 \\ F_2 \\ F_3 \end{pmatrix} = \begin{pmatrix} A_x \\ A_y \\ A_z \end{pmatrix} \quad [B.1.9]$$

avec :  $\begin{cases} v_{xi} = \frac{1}{L_B} \cdot [(L_A \cdot \cos \alpha_i + R) \cdot \cos \theta_i - x] \\ v_{yi} = \frac{1}{L_B} \cdot [(L_A \cdot \cos \alpha_i + R) \cdot \sin \theta_i - y] \\ v_{zi} = \frac{1}{L_B} \cdot [- L_A \cdot \sin \alpha_i - z] \end{cases} \quad i = 1, 2, 3$

et :  $\begin{cases} A_x = - m_n \cdot \ddot{x} \\ A_y = - m_n \cdot \ddot{y} \\ A_z = m_n \cdot [-g - \ddot{z}] \end{cases}$

En posant :

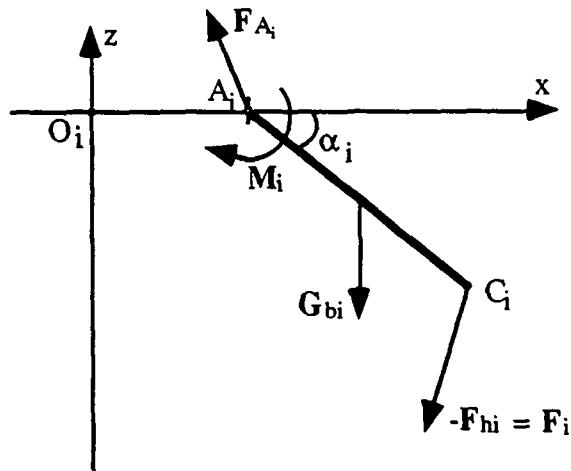
$$Q = \frac{1}{-v_{x3}v_{y2}v_{z1} + v_{x2}v_{y3}v_{z1} + v_{x3}v_{y1}v_{z2} - v_{x1}v_{y3}v_{z2} - v_{x2}v_{y1}v_{z3} + v_{x1}v_{y2}v_{z3}}$$

On trouve :

$$\begin{aligned} F_1 &= Q \cdot [A_z \cdot (v_{x2}v_{y3} - v_{x3}v_{y2}) + A_y \cdot (-v_{x2}v_{z3} + v_{x3}v_{z2}) + A_x \cdot (v_{y2}v_{z3} - v_{y3}v_{z2})] \\ F_2 &= Q \cdot [A_z \cdot (-v_{x1}v_{y3} + v_{x3}v_{y1}) + A_y \cdot (v_{x1}v_{z3} - v_{x3}v_{z1}) + A_x \cdot (-v_{y1}v_{z3} + v_{y3}v_{z1})] \\ F_3 &= Q \cdot [A_z \cdot (v_{x1}v_{y2} - v_{x2}v_{y1}) + A_y \cdot (-v_{x1}v_{z2} + v_{x2}v_{z1}) + A_x \cdot (v_{y1}v_{z2} - v_{y2}v_{z1})] \end{aligned}$$

[B.1.10]

Ainsi, les 3 forces de liaison  $F_i$  sont entièrement définies (norme et direction).

Application de la loi d'Euler sur chacun des bras*Figure B.3.: isolation du bras i*

En se plaçant maintenant dans le référentiel  $R_i$  lié à chacun des bras, on peut calculer le couple moteur en utilisant l'équation d'Euler (figure A.5.3). On obtient :

$$\mathbf{A}_i \mathbf{C}_i \wedge \mathbf{F}_i + \mathbf{r}_{\mathbf{A}_i, \mathbf{G}_{bi}} \wedge \mathbf{G}_{bi} + \mathbf{M}_i = I_{bi} \ddot{\alpha}_i \quad [B.1.11]$$

avec :

- $\mathbf{r}_{\mathbf{A}_i, \mathbf{G}_{bi}}$  vecteur exprimant la position du centre de gravité du bras  $i$ .
- $\mathbf{G}_{bi}$  forces de gravité du bras  $i$  exprimée dans le référentiel  $R_i$ .
- $\mathbf{M}_i$  couple autour du point  $A_i$ .
- $I_{bi}$  inertie du bras autour du point  $A_i$ .
- $\ddot{\alpha}_i$  accélération angulaire du bras  $i$ .

Dans cette équation, le terme de couple gyroscopique est nul, car le vecteur vitesse de rotation du bras est toujours dans la même direction.

### Calcul des produits vectoriels

$$\bullet \quad A_i C_i \wedge F_i = \begin{pmatrix} -\frac{L_A}{L_B} \sin \alpha_i \cdot F_i \cdot y_i \\ \frac{L_A}{L_B} F_i \cdot [z_i \cdot \cos \alpha_i - (R-x_i) \cdot \sin \alpha_i] \\ -\frac{L_A}{L_B} \cos \alpha_i \cdot F_i \cdot y_i \end{pmatrix}$$

$$\bullet \quad r_{A_i, G_b} \wedge G_{bi} = \begin{pmatrix} 0 \\ r_{Gb} \cdot m_b \cdot g \cdot \cos \alpha_i \\ 0 \end{pmatrix}$$

De cette équation, seul le couple autour de l'axe y nous intéresse, puisqu'il s'agit du couple moteur. Les autres couples sont repris par les paliers. De même, il n'est pas nécessaire de poser ici l'équation de Newton, car la force de liaison au point  $A_i$  est aussi reprise par les paliers. L'équation [B.1.11] donne alors pour la composante y :

$$M_{yi} = I_{byi} \ddot{\alpha}_i - F_i \frac{L_A}{L_B} [z_i \cdot \cos \alpha_i - (R-x_i) \cdot \sin \alpha_i] - r_{Gb} \cdot m_b \cdot g \cdot \cos \alpha_i \quad [B.1.12]$$

où le couple  $M_{yi}$  est défini positif selon la règle du tire-bouchon (voir figure B.3), et avec :

$I_{byi}$	inertie du bras i autour du point $A_i$
$x_i, z_i$	coordonnées de la nacelle exprimées dans le repère $R_i$
$r_{Gb}$	distance entre le point $A_i$ et le centre de gravité du bras i
$m_b$	masse totale du bras, y compris la masse des avant-bras rapportée
$g$	accélération terrestre

Il faut noter ici que, dans les équations [B.1.9] et [B.1.12] les accélérations  $\dot{X}$  et  $\ddot{\alpha}$  sont calculées par double dérivation numérique de  $X$  et  $\alpha$  respectivement. La relation entre ces deux derniers vecteurs est donnée par la transformation de coordonnée du robot (voir annexe A).

## B.2. Modèle dynamique "découplé" pour le robot DELTA

Le modèle développé dans le paragraphe précédent ne permet pas d'isoler la matrice d'inertie du robot, ce qui est nécessaire pour implémenter la méthode de réglage par découplage non-linéaire.

En mettant l'équation [3.3.7] sous forme vectorielle, on obtient :

$$\mathbf{M} = \mathbf{A}(\alpha) \cdot \ddot{\alpha} + \mathbf{C}(\alpha, \dot{\alpha}) \quad [\text{B.2.1}]$$

où :

$$\mathbf{M} = \begin{pmatrix} M_{y1} \\ M_{y2} \\ M_{y3} \end{pmatrix}, \quad \ddot{\alpha} = \begin{pmatrix} \ddot{\alpha}_1 \\ \ddot{\alpha}_2 \\ \ddot{\alpha}_3 \end{pmatrix}, \quad \mathbf{F} = \begin{pmatrix} F_1 \\ F_2 \\ F_3 \end{pmatrix}$$

$$\mathbf{I}_b = \begin{pmatrix} I_{b1} & 0 & 0 \\ 0 & I_{b2} & 0 \\ 0 & 0 & I_{b3} \end{pmatrix}$$

$$\mathbf{K} = \frac{L_A}{L_B} \cdot \begin{pmatrix} K_1 & 0 & 0 \\ 0 & K_2 & 0 \\ 0 & 0 & K_3 \end{pmatrix}$$

$$\text{avec } K_i = z_i \cdot \cos \alpha_i - (R - x_i) \cdot \sin \alpha_i, \quad i = 1, 2, 3$$

$$\mathbf{G}_b = \begin{pmatrix} r_{Gb} m_b g \cos \alpha_1 \\ r_{Gb} m_b g \cos \alpha_2 \\ r_{Gb} m_b g \cos \alpha_3 \end{pmatrix}$$

La résolution de l'équation [B.1.9] donne :

$$\mathbf{F} = \mathbf{Q} \cdot \mathbf{M} \cdot \begin{pmatrix} A_x \\ A_y \\ A_z \end{pmatrix} \quad [\text{B.2.2}]$$

avec

$$Q = \frac{1}{-v_{x3}v_{y2}v_{z1} + v_{x2}v_{y3}v_{z1} + v_{x3}v_{y1}v_{z2} - v_{x1}v_{y3}v_{z2} - v_{x2}v_{y1}v_{z3} + v_{x1}v_{y2}v_{z3}}$$

$$\mathbf{M} = \begin{pmatrix} V_{y2}V_{z3}-V_{y3}V_{z2} & -V_{x2}V_{z3}+V_{x3}V_{z2} & V_{x2}V_{y3}-V_{x3}V_{y2} \\ -V_{y1}V_{z3}+V_{y3}V_{z1} & V_{x1}V_{z3}-V_{x3}V_{z1} & -V_{x1}V_{y3}+V_{x3}V_{y1} \\ V_{y1}V_{z2}-V_{y2}V_{z1} & -V_{x1}V_{z2}+V_{x2}V_{z1} & V_{x1}V_{y2}-V_{x2}V_{y1} \end{pmatrix}$$

où :

$$\begin{cases} v_{xi} = \frac{1}{L_B} \cdot [(L_A \cdot \cos \alpha_i + R) \cdot \cos \theta_i - x] \\ v_{yi} = \frac{1}{L_B} \cdot [(L_A \cdot \cos \alpha_i + R) \cdot \sin \theta_i - y] \\ v_{zi} = \frac{1}{L_B} \cdot [-L_A \cdot \sin \alpha_i - z] \end{cases} \quad i = 1, 2, 3$$

et :

$$\begin{cases} A_x = -m_n \cdot \ddot{x} \\ A_y = -m_n \cdot \ddot{y} \\ A_z = m_n \cdot [-g - \ddot{z}] \end{cases}$$

En remplaçant  $A_x$ ,  $A_y$ ,  $A_z$  par leurs valeurs dans l'équation [B.2.2], on obtient :

$$\mathbf{F} = \mathbf{K}_m \cdot \dot{\mathbf{X}} - \mathbf{G}_n \quad [B.2.3]$$

avec :  $\mathbf{K}_m = -m_n \cdot Q \cdot \mathbf{M}$ ,  
 et  $\mathbf{G}_n = m_n \cdot Q \cdot \mathbf{M} \cdot \begin{pmatrix} 0 \\ 0 \\ g \end{pmatrix}$

La relation entre  $\dot{\mathbf{X}}$  et  $\ddot{\alpha}$  peut être obtenue à partir du modèle géométrique direct. En effet, à partir de la relation  $\mathbf{X} = f(\alpha)$ , on obtient :

$$\dot{\mathbf{X}} = \begin{pmatrix} \frac{\partial f_x}{\partial \alpha_1} & \frac{\partial f_x}{\partial \alpha_2} & \frac{\partial f_x}{\partial \alpha_3} \\ \frac{\partial f_y}{\partial \alpha_1} & \frac{\partial f_y}{\partial \alpha_2} & \frac{\partial f_y}{\partial \alpha_3} \\ \frac{\partial f_z}{\partial \alpha_1} & \frac{\partial f_z}{\partial \alpha_2} & \frac{\partial f_z}{\partial \alpha_3} \end{pmatrix} \cdot \dot{\alpha} = \mathbf{J}(\alpha) \cdot \dot{\alpha}$$

où  $\mathbf{J}$  désigne la matrice jacobienne du robot qui dépend de sa position  $\alpha$ . Une dérivée supplémentaire de cette équation donne alors :

$$\ddot{\mathbf{X}} = \mathbf{J}(\alpha) \cdot \ddot{\alpha} + \mathbf{D}(\alpha, \dot{\alpha}) \quad [B.2.4]$$

où  $D(\alpha, \dot{\alpha})$  contient tous les termes de dérivée partielle du deuxième ordre et est fonction des position et vitesse du robot.

On notera ici que les termes de la matrice jacobienne sont calculés numériquement de la manière suivante (exemple pour le premier terme) :

$$\frac{\partial f_x}{\partial \alpha_1} = \frac{f_x(\alpha_1 + \Delta, \alpha_2, \alpha_3) - f_x(\alpha_1, \alpha_2, \alpha_3)}{\Delta}$$

En combinant [B.2.4], [B.2.3] et [B.2.1] on obtient alors :

$$M = (I_b - K \cdot K_m \cdot J) \ddot{\alpha} - K \cdot K_m \cdot D + K \cdot G_n - G_b \quad [B.2.5]$$

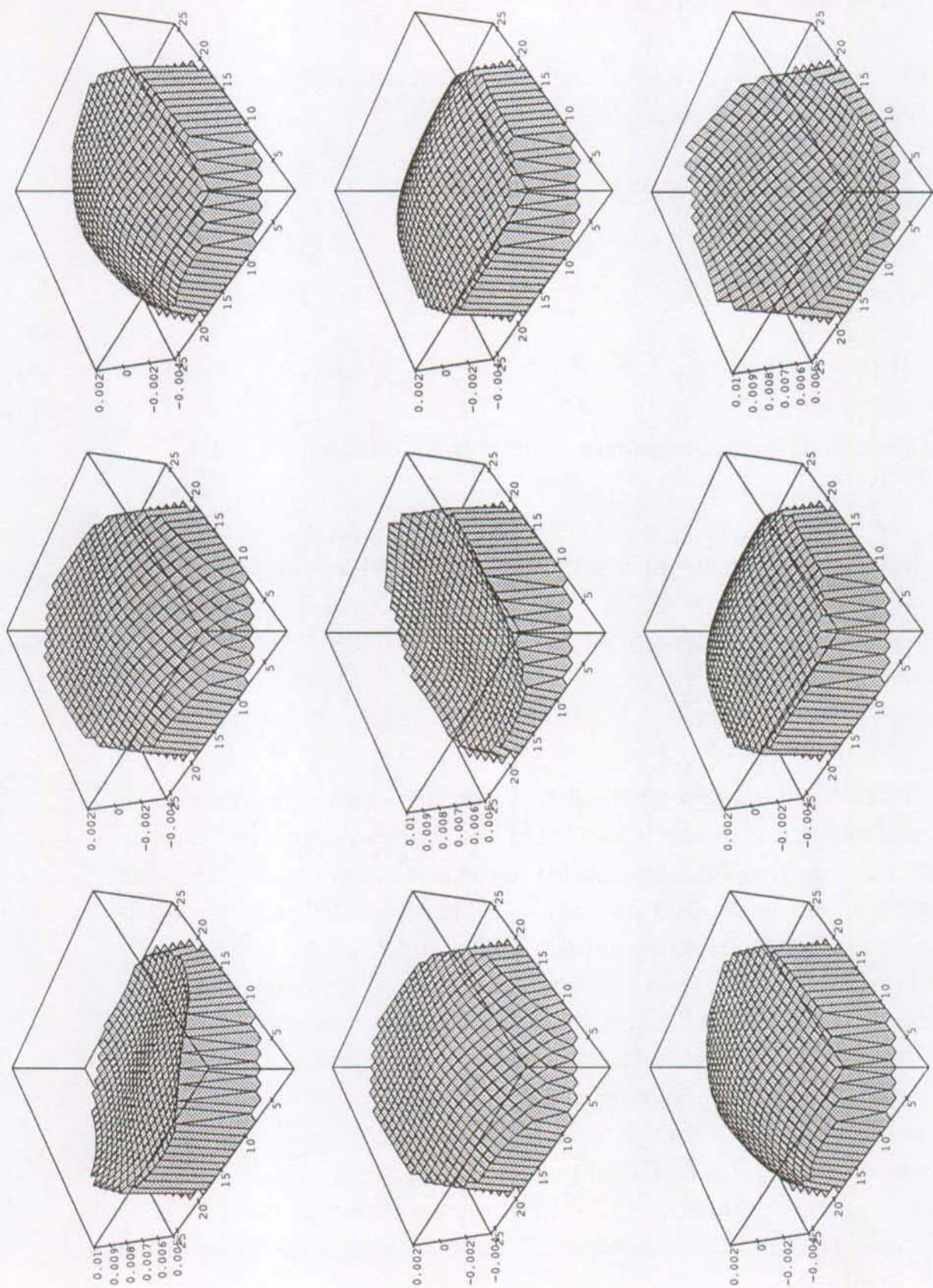
On obtient ainsi le modèle dynamique du robot DELTA sous la forme désirée.

### B.3. Représentation de la matrice d'inertie du robot DELTA

Dans l'équation [B.2.5] ci-dessus, le premier terme peut être identifié à la matrice d'inertie du robot, soit :

$$A = I_b - K \cdot K_m \cdot J \quad [B.3.1]$$

En considérant le volume de travail du robot DELTA comme un cylindre d'axe vertical, comme proposé dans [Clavel 91], on peut représenter graphiquement cette matrice d'inertie pour différentes tranches horizontales du cylindre. Les 3 tranches considérées, sont situées à des hauteurs  $z = -230$  [mm],  $z = -300$  [mm] et  $z = -350$  [mm] et représentées respectivement dans les figures B.3.1, B.3.2, et B.3.3. Les 9 graphiques représentent chacun des termes de la matrice A de dimension  $3 \times 3$ . Les termes diagonaux sont les termes d'inertie propre de chacun des axes et les termes non-diagonaux sont l'image de l'inertie de couplage. L'axe vertical des graphiques fournit la valeur de l'inertie en  $[kgm^2]$  et les axes horizontaux représentent la position de la nacelle du robot dans le plan xy, x et y étant orientés selon le sens trigonométrique. Ce plan est échantilloné entre les valeurs extrêmes du volume de travail, soit entre  $-250$  [mm] et  $+250$  [mm], par pas de  $20$  [mm]. Ainsi, sur le graphique, la valeur 0 correspond à  $-250$  [mm] et la valeur 25 correspond à  $+250$  [mm].



*Figure B.3.1 : Matrice d'inertie du robot DELTA pour un plan situé à  $z = -230$  [mm]*

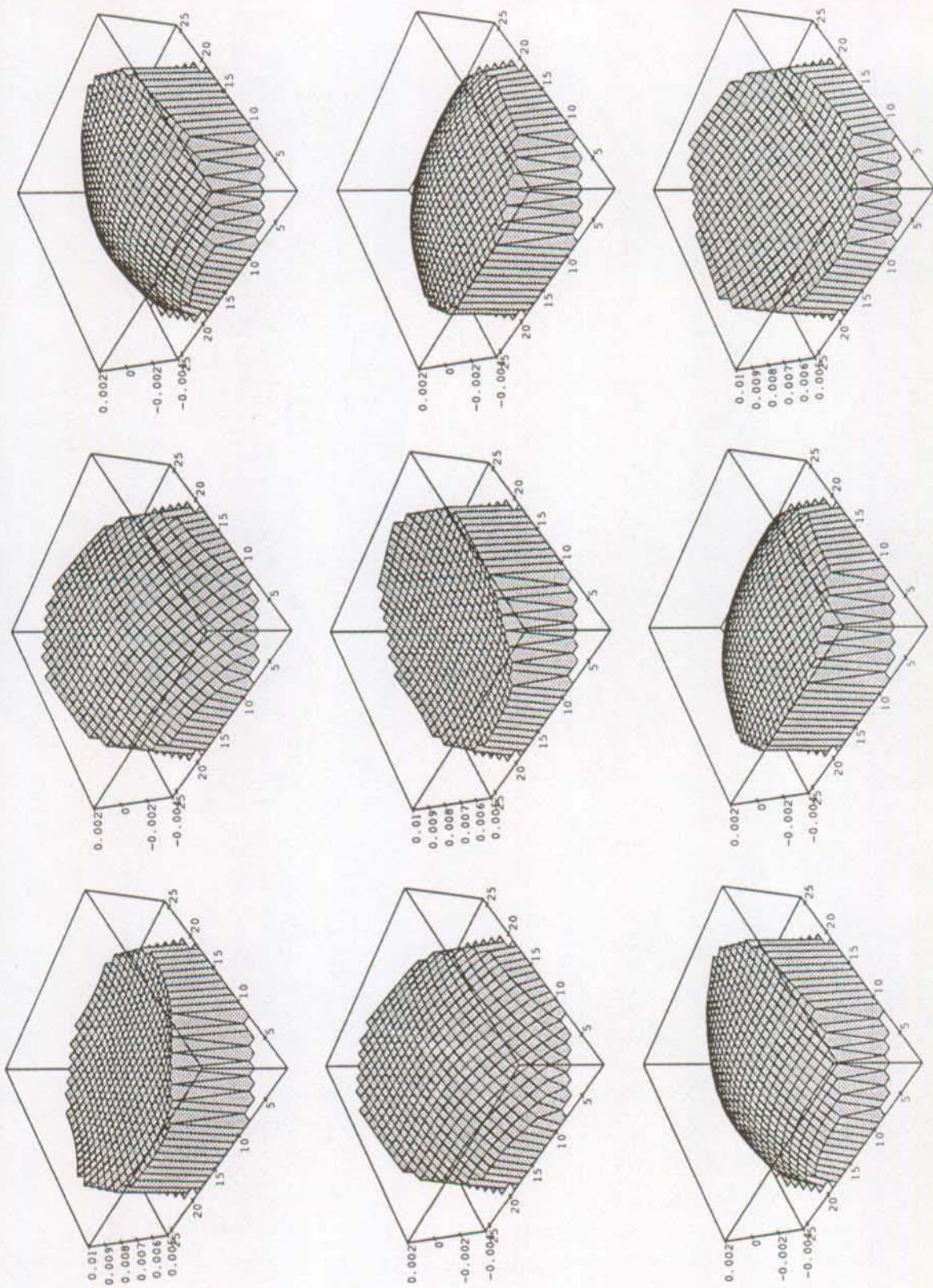
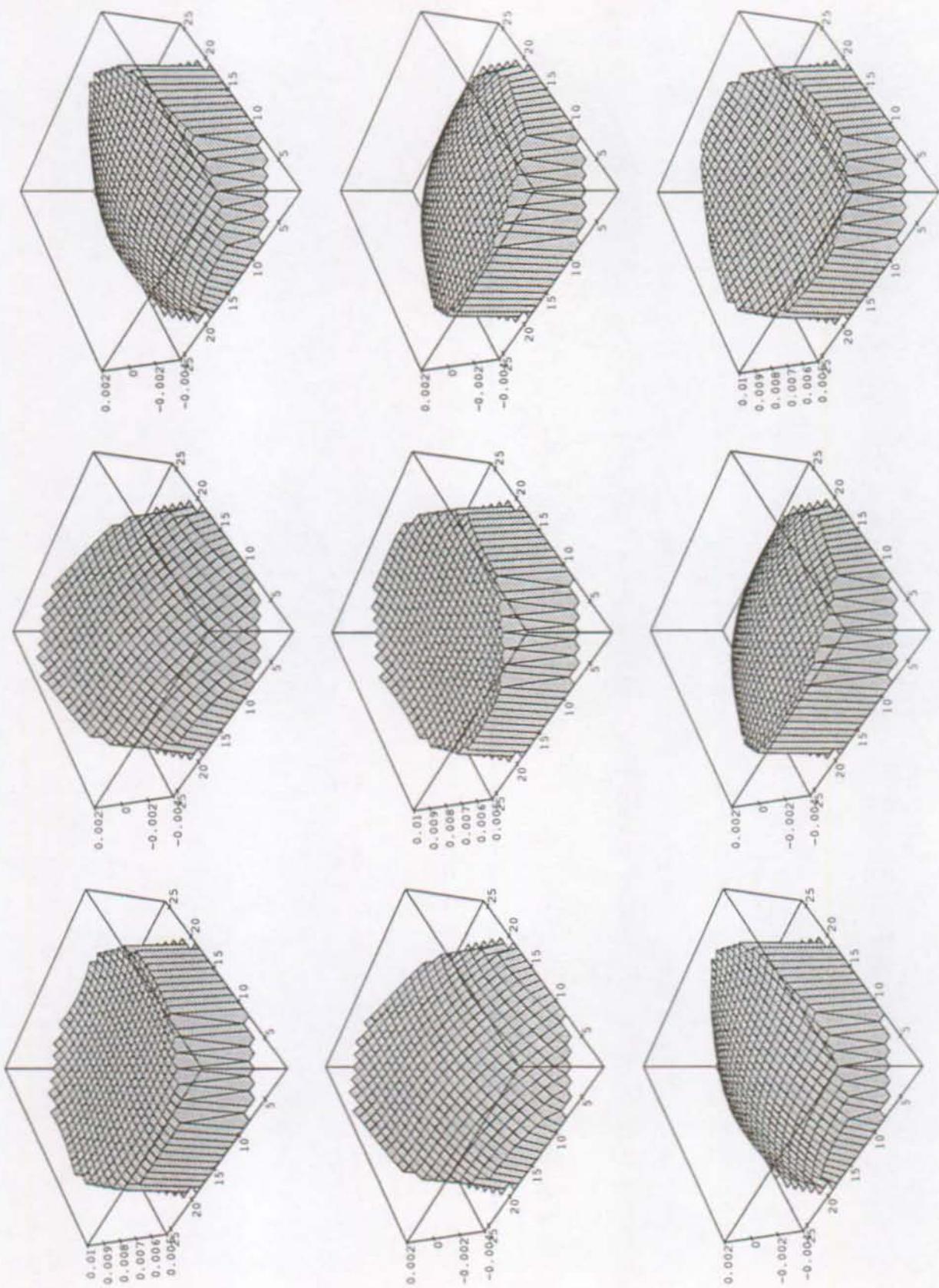


Figure B.3.2 : Matrice d'inertie du robot DELTA pour un plan sinusoïdal à  $z = -300 \text{ [mm]}$

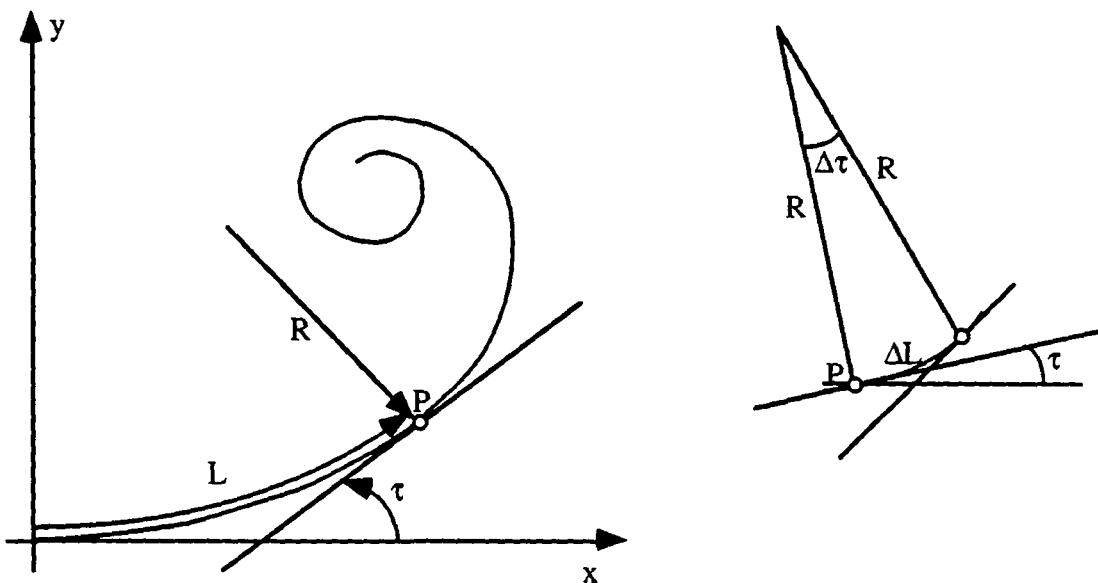


*Figure B.3.3 : Matrice d'inertie du robot DELTA pour un plan situé à  $z = -350 [mm]$*

## C. Equations de la clothoïde

### C.1. Introduction

La clothoïde (figure C.1.) est une courbe dont la courbure est proportionnelle à la distance parcourue. Du point de vue dynamique, c'est une courbe intéressante car, lorsqu'elle est parcourue à vitesse constante, la variation de l'accélération transverse est constante. Ceci permet d'obtenir une trajectoire sans discontinuité d'accélération pour passer d'une droite à une autre droite de direction différente.



*Figure C.1. : Paramètres de la clothoïde*

Son équation de base est la suivante :

$$L = A^2 \cdot K = A^2 \cdot \frac{1}{R} \quad [C.1.1]$$

- avec : L : distance parcourue  
 K : courbure au point P  
 R : rayon de courbure au point P  
 A : facteur de proportionnalité

Sous forme différentielle, on a :

$$L = A^2 \cdot \frac{d\tau}{dL} \quad \text{ou} \quad L dL = A^2 d\tau \quad [C.1.2]$$

avec :  $\tau$ : angle de la tangente au point P

Après intégration, on obtient les 2 équations (équivalentes) :

$$L = \sqrt{2 A^2 \tau} \quad [C.1.3]$$

$$\tau = \frac{A^2}{2 R^2} \quad [C.1.4]$$

On obtient alors pour x et y :

$$x(s) = \frac{A}{\sqrt{2}} \int_0^s \frac{\cos \tau}{\sqrt{\tau}} d\tau \quad [C.1.5]$$

$$y(s) = \frac{A}{\sqrt{2}} \int_0^s \frac{\sin \tau}{\sqrt{\tau}} d\tau \quad [C.1.6]$$

qui sont les intégrales de Fresnel. Celles-ci n'ont pas de solution analytique et il est nécessaire d'utiliser leur décomposition en série de Taylor. On obtient alors en fonction du rayon de courbure R et du facteur de proportionnalité A :

$$x = \sum_{n=1}^{\infty} (-1)^{n+1} \cdot \frac{A^{4n-2}}{2^{2n-2} (4n-3) (2n-2)! R^{4n-3}} \quad [C.1.7]$$

$$y = \sum_{n=1}^{\infty} (-1)^{n+1} \cdot \frac{A^{4n}}{2^{2n-1} (4n-1) (2n-1)! R^{4n-1}} \quad [C.1.8]$$

Pour le calcul numérique, il est intéressant d'exprimer le n<sup>ème</sup> terme de la série en fonction du n-1<sup>ème</sup> terme. Ainsi :

$$\bullet \quad t_x(n) = -t_x(n-1) \cdot C_x(n) \quad [C.1.9]$$

$$\text{avec : } t_x(1) = \frac{A^2}{R}$$

$$C_x(n) = \frac{A^4}{R^4} \cdot \frac{(4n-7)}{4 (4n-3) (2n-2) (2n-3)}, \quad n = 2, 3, \dots$$

- $t_y(n) = -t_y(n-1) \cdot C_y(n)$  [C.1.10]

avec :  $t_y(1) = \frac{A^4}{6R^3}$

$$C_y(n) = \frac{A^4}{R^4} \frac{(4n-5)}{4(4n-1)(2n-2)(2n-1)}, \quad n = 2, 3, \dots$$

## C.2. Clohoïde au sommet

La clohoïde au sommet relie de manière continue deux droites concourantes au moyen de deux clohoïdes consécutives (figure C.2.).

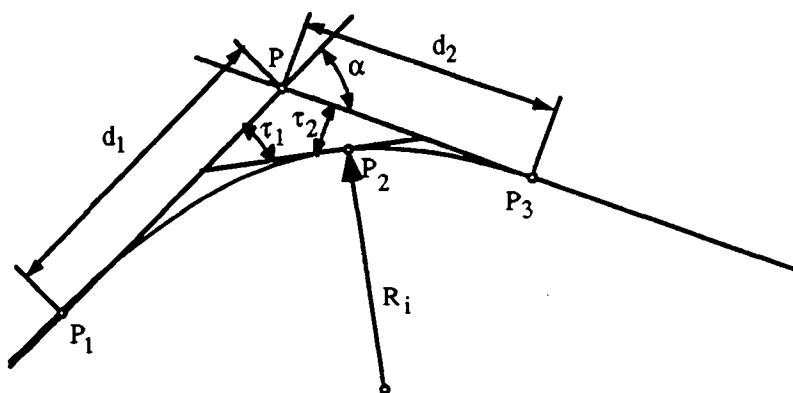


Figure C.2. : Clohoïde au sommet

Soit :

- α : angle entre les deux droites
- d<sub>1</sub> : distance du point P<sub>1</sub> de départ de la clohoïde sur la droite 1 au point P d'intersection des deux droites
- d<sub>2</sub> : distance entre le point P et le point P<sub>3</sub> sur la droite 2
- R<sub>i</sub> : rayon de courbure au point de raccordement P<sub>2</sub> des 2 clohoïdes
- A<sub>1</sub>, A<sub>2</sub> : constante de proportionnalité de la clohoïde 1, respectivement 2
- τ<sub>1</sub>, τ<sub>2</sub> : angle de la tangente au point P<sub>2</sub> avec la droite 1, respectivement 2

Posons :

$$\beta = \frac{\pi}{2} - \alpha$$

En projetant le point  $P_2$  sur chacune des deux droites on obtient les deux équations de la clohoïde au sommet suivantes :

$$y_1 + x_2 \cos \beta - y_2 \sin \beta = d_2 \cos \beta \quad [C.2.1]$$

$$y_2 + x_1 \cos \beta - y_1 \sin \beta = d_1 \cos \beta \quad [C.2.2]$$

où  $x_1$  et  $y_1$  sont les coordonnées du point  $P_2$  dans un référentiel lié à la droite 1 et dont l'origine est le point  $P_1$ , et  $x_2, y_2$  les coordonnées du point  $P_2$  dans un référentiel lié à la droite 2 centré en  $P_3$ .  $x_1, y_1, x_2$  et  $y_2$  sont fonction de  $A_1, A_2$  et  $R_i$  et sont calculés par les équations [C.1.7] et [C.1.8].

Au niveau des angles, on obtient également :

$$\tau_1 + \tau_2 = \alpha \quad \text{ou}$$

$$\frac{A_1^2}{2R_i^2} + \frac{A_2^2}{2R_i^2} = \alpha \quad [C.2.3]$$

En choisissant  $d_1$  et  $d_2$ , les équations [C.2.1],[C.2.2] et [C.2.3] permettent de calculer les paramètres  $A_1$  et  $A_2$  des 2 clohoïdes et le rayon de courbure  $R_i$  à leur point de jonction. La complexité de ces équations nécessite l'utilisation d'un algorithme de résolution numérique (Newton-Raphson) pour trouver les solutions.

[Baumann 91] montre que  $d_1$  et  $d_2$  ne peuvent pas être choisis arbitrairement, et que suivant la valeur de l'angle  $\alpha$  entre les deux droites le rapport maximal admissible entre ces deux distances varie. Dans le meilleur des cas (lorsque  $\alpha$  tend vers 0) le rapport entre  $d_1$  et  $d_2$  peut être au maximum de 2. Ceci limite considérablement l'intérêt d'utiliser une clohoïde asymétrique ( $d_1 \neq d_2$ ) et motive le développement du cas symétrique ( $d_1 = d_2$ ).

### C.3. Cas particulier de la cloïdoïde au sommet symétrique

Le cas de la cloïdoïde au sommet symétrique est beaucoup plus simple que le cas général. Celui-ci est représenté par la figure C.3.

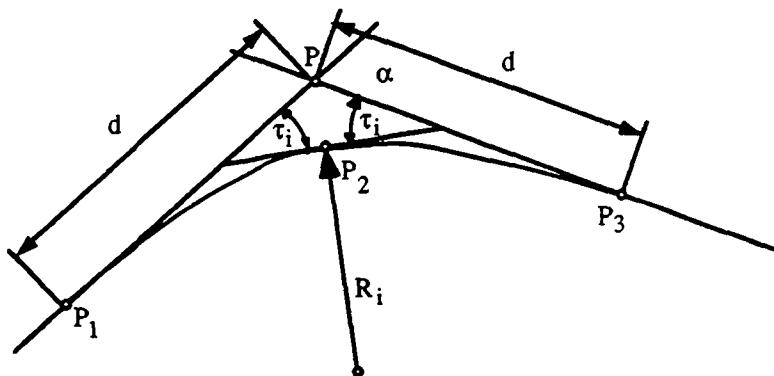


Figure C.3. : Cloïdoïde au sommet symétrique

Les notations deviennent :

- $\alpha$  : angle entre les deux droites
- $d$  : distance du point de départ de la cloïdoïde au point  $P$  d'intersection des deux droites
- $R_i$  : rayon de courbure au point de raccordement  $P_2$  des 2 cloïdoïdes
- $A$  : constante de proportionnalité des cloïdoïdes
- $\tau_i$  : angle de la tangente au point  $P_2$  avec chacune des 2 droites.

Avec cette configuration symétrique, le calcul de  $\tau_i$  est immédiat :

$$\tau_i = \frac{\alpha}{2} \quad [C.3.1]$$

En exprimant  $R_i$  avec [C.1.4] et en remplaçant sa valeur dans [C.1.7] et [C.1.8] on trouve pour  $x$  et  $y$  au point  $P_2$  :

$$x = A \cdot \sum_{n=1}^{\infty} (-1)^{n+1} \cdot \frac{(2\tau_i)^{2n-1.5}}{2^{2n-2} (4n-3) (2n-2)!} = A \cdot W_x(\tau_i) \quad [C.3.2]$$

$$y = A \cdot \sum_{n=1}^{\infty} (-1)^{n+1} \cdot \frac{(2\tau_i)^{2n-0.5}}{2^{2n-1} (4n-1) (2n-1)!} = A \cdot W_y(\tau_i) \quad [C.3.3]$$

où  $\tau_i$  est une fonction de  $\alpha$  définie par l'équation [C.3.1].

On remarque ainsi que  $x$  et  $y$  de la cloxoïde s'expriment linéairement en fonction de  $A$  au point  $P_2$ . La projection de  $P_2$  sur la droite permet alors de trouver la valeur de  $A$  en fonction de  $d$  et  $\alpha$ .

$$A = \frac{d}{W_x(\alpha) + W_y(\alpha) \cdot \tan \frac{\alpha}{2}} \quad [C.3.4]$$

En remplaçant  $\tau_i$  par  $\tau$  dans les équations [C.3.2] et [C.3.3] et en calculant  $\tau$  à partir de l'équation [C.1.3] on peut trouver chaque point de la cloxoïde en fonction de son abscisse curviligne. Selon [Baumann 91] seuls 6 termes de la série sont nécessaires pour obtenir une précision du pourmille.

## D. Paramètres du robot prototype

### D.1. Paramètres géométriques

Pour le robot prototype (montré sur la photo 2.6.2), les valeurs numériques des différents éléments constituants (définis dans le paragraphe 3.3.1) sont les suivantes :

$$R_A = 190 \text{ [mm]}$$

$$R_B = 25 \text{ [mm]}$$

$$L_A = 200 \text{ [mm]}$$

$$L_B = 380 \text{ [mm]}$$

### D.2. Paramètres dynamiques

Les valeurs numériques des paramètres dynamiques définis dans le paragraphe 3.3.4 sont les suivantes :

Accélération terrestre :  $g = 9.81 \text{ [m/s}^2\text{]}$

Masse des avant-bras :  $m_{ab} = 0.044 \text{ [kg]}$

Masse totale de la nacelle :  $m_n = m_{na} + m_{tr} + 3m_m$

avec :  $m_{na} = 0.0675 \text{ [kg]}$  masse de la nacelle

$m_{tr} = 0$  masse transportée

$m_m = (1-r) \cdot m_{ab}$  masse des avant-bras rapportée sur la nacelle

Masse totale du bras :  $m_b = m_{br} + m_c + m_{rc}$

avec :  $m_{br} = 0.08 \text{ [kg]}$  masse du bras

$m_c = 0.015 \text{ [kg]}$  masse du coude

$m_{rc} = r \cdot m_{ab}$  masse des avant-bras rapportée au coude

Rapport des masses rapportées :  $r = 0.7$

Inertie du moteur :  $I_m = 13 \cdot 10^{-4} \text{ [kgm}^2\text{]}$

Inertie du bras :  $I_{br} = L_A^2 (m_{br}/3 + m_c + m_{rc})$

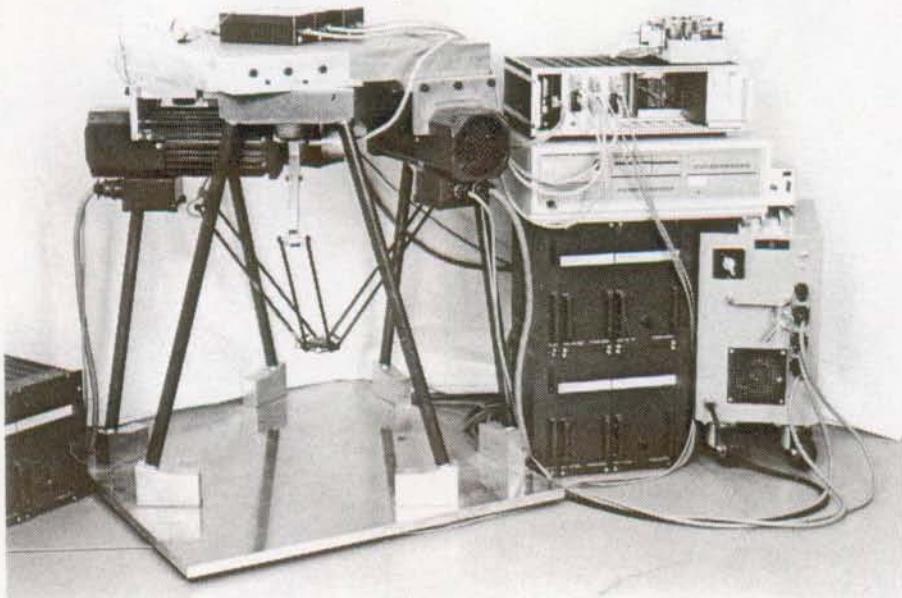
Inertie totale du bras :  $I_{bi} = I_{br} + I_m$

Distance moteur - centre de gravité du bras :  $r_{Gb} = L_A \cdot (0.5 \cdot m_{br} + m_c + m_{rc})/m_b$



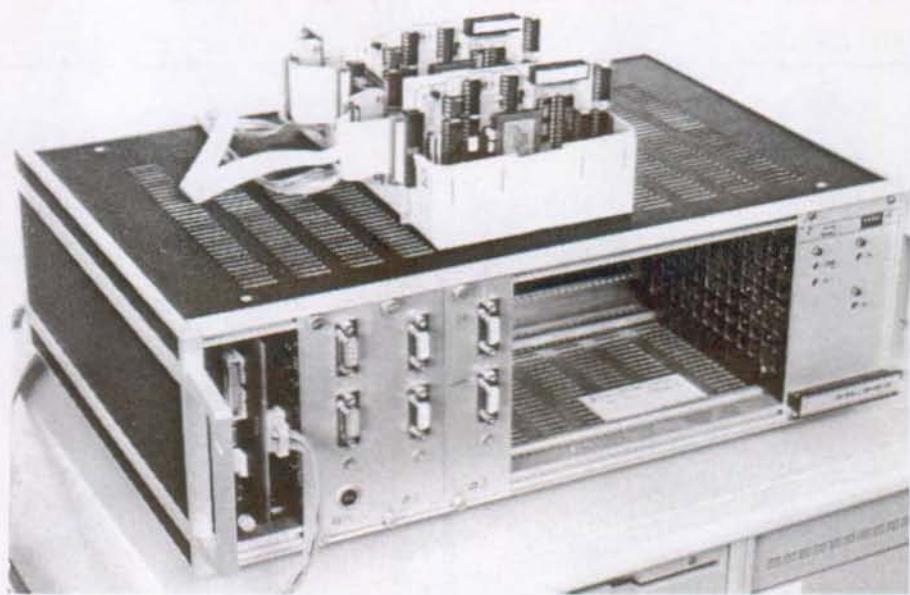
## E. Vue d'ensemble du prototype et de sa commande

La photo E.1. montre une vue d'ensemble du système, où l'on peut distinguer le robot utilisé pour les tests, son électronique de puissance (amplificateurs), et sa commande. On remarquera la taille réduite de cette dernière par rapport au reste du prototype.



*Photo E.1. : Vue d'ensemble du prototype et de son électronique de commande*

La commande est basée sur 3 transputers, dont l'un est situé dans un ordinateur type PC. Les 2 autres transputers, installés sur des cartes développées au Laboratoire de Systèmes Périphériques (LSP) de l'EPFL, sont logés ici dans une boîte en carton en guise de rack, comme le montre la photo E.2. Le rack situé en dessous contient essentiellement 3 cartes GESART, développées à l'IMT pour la commande d'axe robot, une carte interface-lien développée au LSP pour la liaison d'un transputer avec le bus GESPAC et une carte microprocesseur 68000 de GESPAC utilisée ici uniquement pour le dialogue entre les cartes GESART et l'interface-lien.



*Photo E.2. : Vue d'ensemble du rack d'interface robot et des cartes transputer*

## Bibliographie

[Adept 85] Adept Technology, inc., *The AdeptOne™ Manipulator*, Technical documentation, 1985.

[An & al. 88] C. An, C. Atkeson, J. Hollerbach, *Model-Based Control of a Robot Manipulator*, The MIT Press, 1988.

[Arai & al 85] T. Arai, E. Nakado, t. Yano, R. Hashimoto, I. Takeyama, *Development of Direct-Drive Human-Like Manipulator*, Proc. of the 15th ISIR, Tokyo, 1985.

[Asada & al 84a] H. Asada, K. Youcef-Toumi, R. Ramirez, *M.I.T. Direct-Drive Arm Project*, Robots 8, Conference Proceedings. Vol. 2 Future Considerations, 1984.

[Asada & al 84b] H. Asada, K. Youcef-Toumi, R. Ramirez, *Design of MIT Direct-Drive Arm*, International Symposium on Design and Synthesis, 1984.

[Asada & Kanade 81] H. Asada, T. Kanade, *Design Concept of Direct-Drive Manipulators Using Rare-Earth DC Torque Motors*, Proceedings of the 7th International Joint Conference on Artificial Intelligence, 1981.

[Asada & Slotine 86] H. Asada, J.-J. Slotine, *Robot Analysis and Control*, John Wiley and Sons, 1986.

[Asada & Youcef-Toumi 87] H. Asada, K. Youcef-Toumi, *Direct-Drive Robots, Theory and Practice*, The MIT Press, 1987.

[Baumann 91] T. Baumann, *Trajectoires pour le robot DELTA*, Projet de diplôme, Institut de Microtechnique EPFL, 1991.

[Burckhardt 72] C.W. Burckhardt, *Les lois de similitudes en microtechnique*, Journées de Microtechnique, EPFL, 1972.

[Burckhardt & al 90] C.W. Burckhardt, *Introduction aux robots industriels*, Cours EPFL, Institut de Microtechnique, 1990.

[Clavel 88] R. Clavel, *DELTA, a fast robot with parallel geometry*, Proc. Int. Symposium on Industrial Robots (ISIR), p. 91-100, 1988.

[Clavel 89] R. Clavel, *Une nouvelle structure de manipulateur parallèle pour la robotique légère*, APII (Automatique - Productique - Informatique Industrielle) No 23 p. 501-519, 1989.

[Clavel 91] R. Clavel, *Conception d'un robot parallèle rapide à 4 degrés de liberté*, Thèse EPFL No 925, Lausanne 1991.

[Craig 86] J.J. Craig, *Introduction to Robotics, Mechanics & Control*, Addison-Wesley Publishing Company, 1986.

[Craig 88] J.J. Craig, *Adaptive Control of Mechanical Manipulators*, Addison-Wesley Publishing Company, 1988.

[Cronshaw 90] T. Cronshaw, *The "Intelligent" Chocolate Packing Robot*, Proc. Int. Symposium on Industrial Robots (ISIR), p. 151-158, 1990.

[Dalum & Lund 90] L. Dalum, T. Lund, *Trimming Robot for Fish Fillets*, Proc. Int. Symposium on Industrial Robots (ISIR), p. 145-149, 1990.

[Dayer 88] C. Dayer, *Etude dynamique du robot DELTA3*, Projet de semestre, EPFL - Institut de Microtechnique, 1988.

[Dayer 89] C. Dayer, *Optimisation du DELTA4*, Projet de diplôme, EPFL, Institut de Microtechnique, 1989.

[Dombre & Khalil 88] E. Dombre, W. Khalil, *Modélisation et commande des robots*, Ed. Hermès, Paris, 1988.

[Dubowsky & DesForges 79] S. Dubowsky, D.T. DesForges, *On the development of high performance adaptive control algorithms for robotic manipulators*, Trans. of ASME, Journal of Dynamic Systems, Measurement, and Control, Vol. 101, p. 193-200, 1979.

**[Faessler 88a]** H. Faessler, *Concurrent Motion and Force Control for Robot Manipulators*, Thèse ETHZ No. 8717, Zürich, 1988.

**[Faessler 88b]** H. Faessler, K. Buffinton, E. Nielsen, *Design for a high speed robot. Skilled in the play of ping-pong*, Proc. Int. Symposium on Industrial Robots (ISIR) 221-232, april 1988.

**[Ganguillet 82]** C. Ganguillet, *Etude d'un système intégré pour la définition géométrique du tracé d'une voie de circulation utilisant en particulier le diagramme de courbure de l'axe polynomial*, Thèse EPFL No 458, Lausanne, 1982.

**[Guglielmetti 90]** P. Guglielmetti, R. Longchamp, *Real-Time Identification of a Robot Manipulator Using a "Neural Network" Approach*, IEEE Int. Workshop on Intelligent Motion Control, Istanbul, Turkey, 1990.

**[Guzzella 86]** L. Guzzella, *Robustheitseigenschaften von Reglern mit variabler Struktur*, Thèse ETHZ Nr. 8163, Zürich, 1986.

**[Hashimoto & al 87]** H. Hashimoto, F. Ozaki, K. Asano, K. Osuka, *Development of a pingpong robot system using 7 degrees of freedom direct drive arm*, Proceedings - IECON '87 : International Conference on Industrial Electronics, Control, and Instrumentation p. 608-615, 1987.

**[Hsia 86]** T.C. Hsia, *Adaptive Control of robot manipulators; a review*, Proc. IEEE Conference on Robotics and Automation, San Francisco, p. 183-189, 1986.

**[INMOS 89]** INMOS, *The Transputer Databook*, Inmos Limited, Second Edition, 1989.

**[Iwakane 88]** T. Iwakane, J. Hirai, Y. Miyamoto, *Mechanism and characteristics of direct drive motors*, Advanced Robotics, Vol. 2, No. 3, pp. 287-298, 1988.

**[Jufer 79]** M. Jufer, *Transducteurs électromécaniques*, Ed. Georgi, 1979.

**[Kamiya & al 84]** Y. Kamiya, S. Okabe, Y. Yokoyama, *Design of robot for high speed motion*, International Symposium on Design and Synthesis, Tokyo, 1984.

**[Kawato & al 88]** M. Kawato, Y. Uno, M. Isobe, R. Suzuki, *Hierarchical Neural Network Model for Voluntary Movement with Application to Robotics*, IEEE Control Systems Magazine, April 1988.

**[Kazerooni & Kim 88]** H. Kazerooni, S. Kim, *A New Architecture for Direct Drive Robots*, Proceedings - IEEE International Conference on Robotics and Automation, 1988.

**[Khalil 78]** W. Khalil, *Contribution à la commande automatique des manipulateurs avec l'aide d'un modèle mathématique des mécanismes*, Thèse d'Etat, USTL, Montpellier, 1978.

**[Khalil & Kleinfinger 87]** W. Khalil, J.-F. Kleinfinger, *Minimum operations and minimum parameters of the dynamic model of tree structure robots*, IEEE Journal of Robotics and Automation, Vol. RA-3(6), p. 517-526, 1987.

**[Khosla 86]** P. Khosla, *Real-Time Control and Identification of Direct-Drive Manipulators*, Ph. D. Thesis, Carnegie Mellon University, Pittsburgh, 1986.

**[Kleinfinger 86]** J.-F. Kleinfinger, *Modélisation dynamique des robots à chaîne cinématique simple, arborescente ou fermée, en vue de leur commande*, Thèse de Doctorat, ENSM, Nantes, 1986.

**[Koivo & Guo 81]** A.J. Koivo, T.H. Guo, *Control of Robotic Manipulator with Adaptive Control*, Proc. IEEE Conference on Decision and Control, p. 271-276, 1981.

**[Kokkinis & al 87]** T. Kokkinis, J. Wilson, L. Leka, *Direct-Drive Robot for Clean Room Assembly*, Third IEEE/CHMT International Electronic Manufacturing Technology Symposium, Anaheim, 1987.

**[Lee & al 83]** C.S.G. Lee, B.H. Lee, R. Nigam, *Development of the generalized d'Alembert equations of motion for mechanical manipulators*, Proc. of the 22nd Conference on Decision and Control, December 14-16, 1983.

**[Lind 88]** H. Lind, *Industrial Robotics - Accuracy and Speed*, Licentiate Thesis, Department of Machine Elements, The Royal Institute of Technology, Stockholm, 1988.

**[Luh & al 80]** J.Y.S. Luh, M.W. Walker, R.P.C. Paul, *Resolved-Acceleration Control of Mechanical Manipulators*, IEEE transactions on automatic control, Vol. AC-25, No.3, 1980.

**[Megahed 84]** Megaheb S., *Contribution à la modélisation géométrique et dynamique des robots manipulateurs ayant une structure de chaîne cinématique simple ou complexe; application à leur commande*, Thèse d'Etat, Université P. Sabatier, Toulouse, 1984.

**[Nihei 89]** R. Nihei, *Direct Drive Robot and its Control by Digital Servo*, ISIR 20th, Tokyo, 1989.

**[Paul 78]** R. Paul, *Advanced industrial robot systems*, Purdue University, Report TR-EE 78-25, 1978.

**[Paul 81]** R.P. Paul, *Robot Manipulators : Mathematics, Programming, and Control*, MIT Press, 1981.

**[Reboulet 88]** C. Reboulet, *Modélisation des robots parallèles*, dans "Techniques de la robotique; architectures et commandes", Ed. Hermès, Paris, 1988.

**[Schaum 79]** M. R. Spiegel, *Formules et Tables de Mathématiques*, Série Schaum, McGraw-Hill Inc., 1979.

**[Schmitz & al 85]** D. Schmitz, P. Khosla, T. Kanade, *Development of CMU Direct-Drive Arm II*, Proc. of the 15th ISIR, Tokyo, 1985.

**[Shimada & al 90]** A. Shimada, T. Umeda, N. Yokoshima, N. Kawawada, H. Watanabe, T. Shiota, M. Nagai, *Motion Control of a Vertically Articulated Direct-Drive Robot Manipulator*, Journal of Robotics and Mechatronics, Vol.2 No.1, 1990.

**[Sternheim 87]** F. Sternheim, *Computation of the direct and inverse geometric models of the DELTA4 parallel robot*, Robotersysteme, Vol. 3, p. 199-203, 1987.

**[Suehiro & Takase 85]** T. Suehiro, T. Takase, *Development of a direct-drive manipulator : ETA-3 and enhancement of servo stiffness by a second-order digital filter*, Proc. of the 15th ISIR, Tokyo, 1985.

- [Takase & al 84] K. Takase, T. Hasegawa, T. Suehiro, *Design and control of a direct drive manipulator*, International Symposium on Design and Synthesis, 1984.
- [Takeshita 85] M. Takeshita, H. Sekiguchi, T. Iwasaki, T. Hamira, *Development of direct drive robot*, Proc. of the 15th ISIR, Tokyo, 1985.
- [Thielen 90] P. Thielen, *Commande de robot basée sur transputers*, Travail de diplôme, Institut de Microtechnique, EPFL, 1990.
- [Tondu 84] B. Tondu, *Génération de mouvements point-à-point optimaux en robotique. Application à l'évaluation des performances et à la C.A.O. des robots*, Thèse Université des Sciences et Techniques du Languedoc, Montpellier, 1984.
- [van Griethuysen 87] J.-P. van Griethuysen, *Commande ajustable à microprocesseur pour robots industriels*, Thèse EPFL No 698, 1987.
- [Welburn 84] R. Welburn, *Ultra High Torque Motor System for Direct Drive Robotics*, Robots 8 Conference, 1984.
- [Yoshikawa 90] T. Yoshikawa, *Foundations of robotics; Analysis and Control*, MIT Press, 1990.

## Curriculum vitae

Nom : Codourey

Prénom : Alain

Lieu et date de  
naissance : Fribourg, le 29 octobre 1962

Lieu d'origine : Lovens et Cottens (Fribourg)

### Formation

Juillet 1978 Certificat d'études secondaires  
Ecole Secondaire des Garçons (ESG), Fribourg

Juillet 1982 Certificat de maturité, type C  
Collège Saint-Michel, Fribourg

Janvier 1987 Diplôme d'ingénieur en Microtechnique EPFL  
Ecole Polytechnique Fédérale de Lausanne

### Activités professionnelles

Dès Février 1987 Assistant à l'institut de Microtechnique  
Ecole Polytechnique Fédérale de Lausanne

*Février 1987 - Août 1987*  
Projet de robotique mobile dans le cadre d'un  
projet européen EUREKA

*Dès Septembre 1987*  
Travail de thèse

