

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE  
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE

UNIVERSITE MENTOURI CONSTANTINE 1  
FACULTE DES SCIENCES DE L'INGENIEUR  
DEPARTEMENT DE GENIE MECANIQUE

N° d'ordre : .....  
N° de série : .....

## **THÈSE**

Présentée pour l'obtention du diplôme de

### **DOCTORAT EN SCIENCES**

**SPECIALITE MECANIQUE**

**PAR**

**Khadraoui Amar**

**ALGORITHME PARALLELE ET SERIE POUR LA SIMULATION  
TEMPS REEL ET LA COMMANDE DES ROBOTS  
MANIPULATEURS.**

**APPLICATION A LA COOPERATION MULTI-ROBOTS**

Soutenu le 17/11/2013

Devant le jury :

Président	Mr. Ahmed. BELLAOUAR	Prof. Université Constantine 1
Rapporteur :	Mr. Abdelouahab. ZAATRI	Prof. Université Constantine 1
Examineurs:	Mr. Belkacem. BARKAT	Prof. Université Hadj Lakhdar - Batna
	Mr. Smail. BENNISSAD	Prof. Université Constantine 1
	Mr. Rabah. MANAA	MCA. Université Hadj Lakhdar - Batna
Invité	Mr. Chawki. MAHFOUDI	MCA. Université Oum El Bouaghi
Invité	Mr. Abdelhakim. CHERFIA	MCA. Université Constantine 1

Algorithme parallèle et série pour la simulation  
temps réel et la commande des robots  
manipulateurs.

Application à la coopération multi-robots

## *Remerciements*

Je tiens tout d'abord à remercier **ALLAH** le tout puissant et miséricordieux, qui m'a donné la force et la patience d'accomplir ce modeste travail.

Je tiens à saluer ici les personnes qui, de près ou de loin, ont contribué à la concrétisation de ce travail de thèse de doctorat. Ces remerciements sont rédigés dans un moment de doux relâchement intellectuel, sans véritable rigueur ni souci taxinomique. J'ai laissé au hasard de ma mémoire, plus impressionnée par les événements récents, répétés, ou chargés d'émotions, le soin de retrouver ses personnes. Dans un autre état d'esprit, ces remerciements auraient certainement été tout autres, et j'aurais peut-être oublier un des noms qui suivent. Mais j'ai choisi ce moment précis pour les écrire.

Mes premiers et particuliers remerciements vont, et c'est bien normal, à mon directeur de thèse le professeur **Abdelouaheb ZAATRI** pour sa patience, ses encouragements ainsi que pour sa rigueur intransigeante qui m'a beaucoup apprise. Je remercie en lui l'inspiration, l'aide et le temps qu'il a bien voulu me consacrer et sans qui cette thèse n'aurait jamais vu le jour. Merci beaucoup **Abdelouahab**

Mes remerciements particuliers iront tout naturellement à mon co-encadreur le professeur **Karim DJOUANI** de l'Université Paris XII de Val de Marne au LISSI de m'avoir proposé ce thème et m'a accueilli au sein du laboratoire LISSI de Paris-est. Je tiens à le remercier pour son orientation, sa confiance, sa patience, qui ont constitué un apport considérable sans lequel ce travail n'aurait pas pu être mené bon port. Merci pour tout **Karim**

Il m'est un agréable devoir d'exprimer ma reconnaissance à Monsieur **Chawki MAHFOUDI**, Maître de conférences à l'université Laarbi Ben M'hidi d'Oum El Bouaghi pour les réflexions, les questions et les conseils qu'il m'a prodigué d'une manière permanente tout au long de ce parcours difficile. Je salue en lui son attitude et sa contribution proprement scientifique vis à vis d'un thème aussi complexe dans le domaine de la robotique. Grand merci **Chawki**

Je remercie chaleureusement Monsieur Bilal Bouchemal, qui a notablement contribué à ma compréhension de certains problèmes liés à l'utilisation des outils et du matériel informatique concernant certains aspects scientifiques et techniques du thème. Merci infiniment **Bilal**.

Mes vifs remerciements vont également aux membres du jury pour l'intérêt qu'ils ont porté à notre recherche en acceptant d'examiner notre travail et de l'enrichir par leurs propositions et critiques.

Je tiens à exprimer ma profonde gratitude au professeur **Ahmed BELLAOUAR** de l'université Constantine 1, qui m'a fait l'honneur de présider le jury de thèse de doctorat.

Je tiens à remercier très vivement Monsieur ***Smail BENYASAAD*** Maître de Conférences à l'université Constantine 1 d'avoir accepté de participer dans le jury de ma thèse.

Je remercie vivement Monsieur ***Rabah MANAA*** maître de Conférences à l'université de Batna qui m'a fait l'honneur d'avoir participé au jury de cette thèse.

Toute ma reconnaissance va à Monsieur ***Belkacem BARKAT*** professeur à l'université Batna, à qui j'exprime mes sincères remerciements pour m'avoir fait l'honneur de participer au jury de ma soutenance.

Mes remerciements vont aussi à Monsieur ***Abdelhakim CHERFIA***, Maître de Conférences à l'université Constantine 1, et à Monsieur ***Chawki MAHFOUDI*** Maître de conférences à l'université Laarbi Ben M'hidi d'Oum El Bouaghi qui m'ont honoré de leur présence dans ce jury de soutenance de doctorat.

Je clos enfin ces remerciements en dédiant cette thèse de doctorat à la mémoire de mes parents, à ma famille et aux quelques amis que j'ai eus la chance d'avoir à mes côtés, qui m'ont soutenu tout au long de ces années de travail.

# Résumé

Le travail définit pour cette thèse concerne l'étude et l'analyse des robots manipulateurs sériels coopératifs pour mener à bien des tâches qui ne peuvent être effectuées par un seul robot manipulateur. Le travail concerne aussi l'étude, le choix et l'application de certains algorithmes sériels et parallèles permettant d'accélérer les calculs pour la simulation et surtout pour le contrôle temps réel des robots coopératifs et leur coordination.

Comme on ne dispose pas de modèles généraux de coopération, il est nécessaire en premier lieu de contribuer en établissant et en adaptant des modèles géométriques, cinématiques, et dynamiques appropriés pour le cas de la coopération en tenant compte des contraintes de l'interaction des robots avec l'objet à manipuler dans son environnement. Une contrainte majeure apparaît ici à cause du temps de calcul qui devient prohibitif et ne permet pas un contrôle temps réel du groupe de robots en coopération. C'est la raison pour laquelle, en second lieu, il est nécessaire de contribuer en explorant et en implémentant les approches de calcul sériel mais surtout parallèle en vue d'accélérer les temps de calculs pour la simulation et les applications nécessitant la commande et la coordination en temps réel. Cela peut s'effectuer en implémentant des algorithmes de calcul parallèle destinés à des réseaux de processeurs tel que permis actuellement par la technologie des transputers et des DSP (Digital Signal Processing), des clusters et des super-computers.

Restreinte aux cas des robots manipulateurs sériels en coopération, une analyse de l'état de l'art ainsi que les résultats de nos recherches ont permis de mettre en évidence les éléments suivants :

**a)-** Au niveau de la modélisation dynamique, bien que les modèles de Newton-Lagrange et Newton-Euler ont été développés pour les systèmes sériels en chaîne ouverte, ils restent adaptable et généralisable aux systèmes de robots manipulateurs coopérants. Cependant, dès que le nombre de robots dépasse 3 robots à 6 ddl chacun, le formalisme mathématique devient difficile à organiser et à exploiter par suite de la taille élevée des expressions symboliques concernant les équations du modèle en question.

**b)-** Pour le contrôle du système robotique en temps réel, le calcul parallèle est nécessaire pour réduire au minimum les temps de calcul en utilisant des algorithmes dédiés au calcul parallèle et implémentés sur des réseaux de multiprocesseurs. Dans ce contexte, deux approches principales ont été mises en évidence et testées. Une première approche d'utilisation du calcul parallèle consiste à décomposer les tâches (approche de Zomaya) du modèle dynamique qui utilise l'algorithme récursif de NE, et à affecter les tâches à un nombre de processeurs appartenant à une architecture multiprocesseur. Une seconde approche qui cherche à minimiser le temps de calcul en proposant des algorithmes basés sur des stratégies de décomposition et de manipulation des matrices du modèle dynamique en vue d'obtenir une décomposition ultime permettant la résolution du système d'équations du modèle dynamique (approche Fijani). Dans cette seconde approche, la taille du problème détermine le nombre de processeurs ; si le système robotique comporte  $N$  ddl, l'architecture du réseau comportera  $N$  processeurs. Elle permet une complexité de calcul d'ordre  $O(\log N)$ . La thèse comporte un nombre appréciable d'investigations et de contributions. On notera les quelques éléments suivants :

1. La résolution du problème de distribution des efforts de contact sur les robots manipulateurs sous contraintes de saisie et de non glissement de l'objet manipulé. Les contraintes sur l'objet ont été intégrées au problème de la distribution optimale d'effort entre les deux bras en minimisant un critère de coût quadratique .

2. La formulation de la dynamique et la commande de coordination globale dans l'espace opérationnelle d'un système multirobots manipulant un objet dans l'espace . Dans ce sens un simulateur sous Matlab a été réalisé ce qui nous a permis de tester le modèle et enfin des simulations ont permis d'illustrer les résultats de l'approche proposée.
3. Exploitation de l'approche de calcul parallèle par décomposition et allocation des tâches a un système multiprocesseurs ou le nombre de processeur est choisi par l'utilisateur (approche de Zomaya). Les simulations ont été effectuées sous Matlab et l'émulation et la visualisation de l'implémentation de l'algorithme de décomposition ont été développées sous Labview. Enfin une estimation du temps de calcul multiprocesseurs / monoprocesseurs a été dégagée.
4. Exploitation de l'approche de calcul parallèle par décomposition matricielle et allocation des tâches a un système multiprocesseurs ou le nombre de processeur est imposé par la taille du système (approche de Fijani). Un exemple d'application est présenté.

**Mots-clés** : Modelisation Dynamique et Commande ; Copération Multi-robots ; Distribution Optimale d'effort ; Programmation Quadratique ; Ordonnancement ; Traitement Parallèle ; Transputers ; multi-traitement.

# Abstract

The work defined for this thesis concerns the study and analysis of serial cooperative robot manipulators to carry out tasks that cannot be performed by a single robot manipulator . The work also relates to the study , the selection and application of certain serial and parallel algorithms to speed up the calculations for simulation purposes and especially for real-time cooperative robot control and coordination. As there is no general model of cooperation, it is necessary first to contribute by developing and adapting geometric, kinematic and dynamic models appropriate for the case of cooperation, taking into account the constraints of the interaction of robots with the object to manipulate in its environment . A major constraint appears here because of the computation time becomes prohibitive and does not allow for real-time control of a cooperative group of robots. This is why , in the second place , it is necessary to contribute in exploring and implementing serial and parallel computing approaches. But also to accelerate the computation time for simulation and applications requiring control and real time coordination. This can be done by implementing parallel computing algorithms dedicated to network of processors as currently permitted by the transputer technology and DSP (Digital Signal Processing), clusters and supercomputers. Restricted to cases of serial robot manipulators in cooperation , an analysis of the state of the art as well as the results of our research has highlighted the following elements :

**a)** - In terms of dynamic modeling , although the models of Newton - Lagrange and Newton-Euler have been developed for serial systems in open-chains , they are adaptable and generalized for cooperating robot manipulators systems . However, if the number of robots exceeds 3 robots of 6 DOF each, the mathematical formalism is difficult to organize and exploit because of the big size of symbolic expressions for the equations of the model.

**b)** - For the control of the robotic system in real-time , parallel processing is necessary to minimize the computation time using algorithms dedicated to parallel computing and implemented on multiprocessor network systems. In this context , two main approaches have been highlighted and tested. A first approach uses parallel computing ( approach Zomaya ) by breaking down the complete task into subtasks resulting from the dynamic model using the recursive algorithm NE. The subtasks are assigned to a number of processors belonging to a multiprocessor architecture. A second approach that seeks to minimize the computation time by proposing algorithms based on strategies of matrix decomposition and manipulation of the dynamic model in order to obtain a final decomposition for solving the system of equations the dynamic model ( Fijani approach). In this second approach , the size of the problem determines the number of processors , if the robotic system has N dof , the network architecture requires N processors . It allows a computational complexity of  $O(\log N)$ .

The thesis includes a significant investigations and many contributions . Note the following few elements :

1. Solving the problem of distribution of contact forces on robot manipulators under grasping constraints and without slip of the manipulated object . Constraints on the object have been included in the problem of optimal distribution of effort between the two arms by minimizing a quadratic cost criterion.
2. The formulations of the dynamics and control of overall coordination in the operational space of a system multirobots manipulating an object in space. In this sense a simulator in Matlab was developed which allowed us to test the model and finally simulations were used to illustrate the results of the proposed approach.

3. Exploitation of parallel computing approach by decomposition and allocation of tasks to a multiprocessor system where the number of processors is chosen by the user ( Zomaya approach ) . The simulations were performed using Matlab, emulation and visualization of the implementation of the decomposition algorithm were developed in Labview . Finally an estimate of the computing time comparing multiprocessor to monoprocessor has been derived.
4. Exploitation of parallel computing approach for matrix decomposition and allocation of tasks to a multiprocessor system where the number of processors is imposed by the size of the system ( approach Fijani). An illustrative example is given.

**Keywords :** Dynamic Modeling and Control ; Multi-robots Cooperation ; Optimal Force Distribution ; Quadratic Programming ; Parallel-Processing ; Transputers ; Multiprocessing ; Scheduling.

# Table des matières

<b>1</b>	<b>Introduction</b>	<b>18</b>
1.1	Motivation et Contexte de la Thèse . . . . .	18
1.2	Objectifs de la thèse . . . . .	19
1.3	Contributions . . . . .	20
1.4	Organisation du manuscrit . . . . .	21
<b>2</b>	<b>Etat de l'art sur les robots manipulateurs coopératifs</b>	<b>23</b>
2.1	Introduction . . . . .	23
2.2	Approches de modélisation des robots coopératifs . . . . .	24
2.3	Approches du calcul parallèle pour les robots coopératifs . . . . .	30
2.4	Conclusion . . . . .	35
<b>3</b>	<b>Modélisation géométrique et cinématique</b>	<b>37</b>
3.1	Introduction . . . . .	37
3.1.1	Structure ouverte simple . . . . .	38
3.1.2	Structure arborescente et fermée (chaînes complexes) . . . . .	38
3.1.3	Définition . . . . .	38
3.2	Modèle géométrique direct des robots à structure arborescente . . . . .	41
3.2.1	Application au système multi-robot coopératif . . . . .	41
3.3	Modèle géométrique direct du robot : . . . . .	43
3.3.1	Définition : . . . . .	43
3.4	Modèle géométrique inverse du robot (M.G.I) . . . . .	45
3.4.1	Introduction . . . . .	45
3.4.2	Calcul du modèle géométrique inverse par la méthode de Paul . . . . .	46

3.4.3	Application dans le cas de la coopération multi-robots : . . . . .	47
3.5	Etude cinématique . . . . .	53
3.5.1	Le modèle cinématique direct (MCD) . . . . .	53
3.5.2	Le MCD du système coopératif . . . . .	56
3.6	Calcul du modèle cinématique inverse (MCI) . . . . .	57
3.6.1	Analyse des singularités du système multi-robot . . . . .	58
3.6.2	Application au système multi-robot : . . . . .	59
3.7	Conclusion . . . . .	64
<b>4</b>	<b>Calcul et optimisation de la distribution d'efforts sur l'objet</b>	<b>65</b>
4.1	Distribution optimale des forces . . . . .	65
4.1.1	Introduction . . . . .	65
4.1.2	Formulation du problème . . . . .	66
4.2	Réduction et solution du Problème . . . . .	68
4.2.1	Etude de la continuité de la solution du problème . . . . .	70
4.2.2	Solution optimale du problème . . . . .	71
4.2.3	Simulations . . . . .	74
4.2.4	Résultats de la simulation . . . . .	75
4.3	Conclusion . . . . .	79
<b>5</b>	<b>Dynamique et commande des robots coopératifs</b>	<b>80</b>
5.1	Introduction . . . . .	80
5.2	Modèle dynamique inverse du système multi-robots . . . . .	81
5.2.1	Formalisme de Newton-Euler . . . . .	81
5.2.2	Implémentation de la méthode de Newton-Euler . . . . .	85
5.2.3	Formalisme de Lagrange . . . . .	86
5.3	Dynamique du système multi-robots coopératif . . . . .	89
5.4	Equation de la dynamique de l'objet manipulé . . . . .	89
5.5	Dynamique combiné (robots-objet) . . . . .	90
5.6	Commande des systèmes poly-articulés . . . . .	92
5.6.1	Introduction . . . . .	92
5.7	Commande par découplage non linéaire dans l'espace articulaire . . . . .	93

5.8	Commande par découplage non linéaire dans l'espace opérationnel . . . . .	95
5.9	Application au cas du système coopératif . . . . .	97
<b>6</b>	<b>Calcul parallèle par décomposition des tâches</b>	<b>102</b>
6.1	Introduction . . . . .	102
6.2	L'architecture parallèle . . . . .	103
6.3	Méthode de Zomaya . . . . .	103
6.3.1	Principe de la méthode : . . . . .	104
6.4	Formalisme de Newton Euler pour la dynamique . . . . .	104
6.5	Réorganisation des équations de Newton-Euler en fonction des tâches affectées : . . . . .	108
6.6	LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench) . . . . .	114
6.6.1	L'exécution des programmes sous LabVIEW . . . . .	114
6.6.2	L'implémentation de l'algorithme Newton-Euler sous LabVIEW . . . . .	114
6.6.3	La programmation Multi-cœur avec LabVIEW : . . . . .	116
6.7	Conclusion . . . . .	124
<b>7</b>	<b>Dynamique des robots coopératifs et factorisation matricielle</b>	<b>125</b>
7.0.1	Paramètres liés aux segments et aux articulations . . . . .	125
7.0.2	Quantités spatiales . . . . .	126
7.0.3	Quantités globales (définies pour $j=N$ à 1) . . . . .	127
7.1	Notations générales . . . . .	127
7.2	Expression factorisée de la matrice Jacobienne . . . . .	128
7.3	Formulation dynamique dans l'espace opérationnel . . . . .	129
7.3.1	Chaînes ouvertes . . . . .	129
7.3.2	Chaînes fermées . . . . .	131
7.3.3	Charge commune . . . . .	131
7.4	Calcul des forces de contact . . . . .	131
7.5	Factorisation des matrices $\mathcal{M}_i$ et $\mathcal{M}_i^{-1}$ . . . . .	133
7.6	Stratégie de décomposition des forces d'interaction entre les corps du segment $i$ .	134
7.7	Détermination et décomposition de la matrice d'inertie d'un système multi-robot coopératif . . . . .	142

**8 Conclusion générale**

**145**

# Table des figures

2.1	Système multi-robots coopératifs . . . . .	24
2.2	Manipulation d'un objet avec double contact unilatéral en A et B [53]	29
2.3	Travail ponctuel avec contact unilatéral de $R_B$ en B sur un objet présenté par le manipulateur $R_A$ [54] . . . . .	30
2.4	Robots manipulateurs 6R, coopérants pour le transfert d'une charge [35]. . . . .	35
3.1	Robot à structure ouverte simple . . . . .	38
3.2	Schéma d'une structure arborescente . . . . .	39
3.3	Articulation comportant une ramification . . . . .	40
3.4	Paramètres géométriques du système multi-robot . . . . .	41
3.5	Exemple d'un système multi-robot manipulateurs . . . . .	42
3.6	Modèle géométrique . . . . .	42
3.7	Configuration du robot et choix des repères . . . . .	43
3.8	Translation de l'objet suivant l'axe X . . . . .	49
3.9	Visualisation du mouvement dans le plan $XOY$ . . . . .	49
3.10	Translation de l'objet suivant l'axe Z . . . . .	50
3.11	Visualisation du mouvement dans le plan $YOZ$ . . . . .	50
3.12	Translation et rotation respectivement autour de Z et X . . . . .	51
3.13	Visualisation du mouvement dans le plan $XOZ$ . . . . .	51
3.14	Translation et rotation combinée autour de Z . . . . .	52
3.15	Visualisation du mouvement dans le plan $XOY$ . . . . .	52
3.16	Influence du type de l'articulation sur le repère terminal . . . . .	54
3.17	Position et orientation calculées du robot1 . . . . .	59
3.18	Position et orientation désirées du robot2 . . . . .	60

3.19	Position et orientation désirées de l'objet . . . . .	60
3.20	Coordonnées articulaires du robot1 . . . . .	61
3.21	Coordonnées articulaires du robot2 . . . . .	61
3.22	Profil des vitesses articulaires du robot1 . . . . .	62
3.23	Profil des vitesses articulaires du robot2 . . . . .	62
3.24	Vitesses cartésiennes et angulaires du point de contact du robot1 . . . . .	63
3.25	Vitesses cartésiennes et angulaires du point de contact du robot2 . . . . .	63
4.1	Vue des deux robots et choix des repères . . . . .	67
4.2	Pyramide de friction inscrite dans le cône . . . . .	69
4.3	Paramètres géométriques du système multi-robots coopératifs . . . . .	74
4.4	Exemple d'un système multi-robots coopératifs . . . . .	74
4.5	Vue de l'objet saisi par les trois robots . . . . .	76
4.6	Forces sur le premier robot . . . . .	76
4.7	Forces sur le deuxième robot . . . . .	77
4.8	Forces sur le troisième robot . . . . .	77
4.9	Contraintes $fx_j^2 + fy_j^2 \leq \mu^2 fz_j^2$ vérifiée . . . . .	78
5.1	Bilan des efforts au centre de gravité sur chaque corps . . . . .	82
5.2	Algorithmes cinématique et dynamique . . . . .	86
5.3	Schéma de la commande PID . . . . .	92
5.4	Architecture de la commande . . . . .	94
5.5	Torque control of the robot "j" . . . . .	95
5.6	Vue de l'objet supporté par les deux robots . . . . .	98
5.7	Position et orientation calculées du robot1 . . . . .	99
5.8	Position et orientation calculées du robot2 . . . . .	99
5.9	Position et orientation calculées de l'objet manipulé . . . . .	100
5.10	Coordonnées articulaires du robot1 . . . . .	100
5.11	Coordonnées articulaires du robot2 . . . . .	101
5.12	Forces sur les deux robots . . . . .	101
6.1	Affectation des tâches en première phase . . . . .	107
6.2	Affectation des tâches en deuxième phase . . . . .	108
6.3	Diagramme global des tâches réparties sur les trois processeurs . . . . .	110

6.4	Exemple d'un Robot 6ddl présentant une liaison prismatique . . . . .	111
6.5	Données géométriques du robot . . . . .	112
6.6	Le flux de données dans LabVIEW . . . . .	115
6.7	L'implémentation de la tâche T51 avec le script MATLAB sous LabVIEW . . . . .	115
6.8	L'implémentation de l'algorithme NE sous LabVIEW . . . . .	116
6.9	Les chemins de code indépendants peuvent s'exécuter simultanément .	118
6.10	La tâche T00 (l'initialisation) . . . . .	119
6.11	L'implémentation en séquentiel de la phase 1 . . . . .	120
6.12	L'implémentation en séquentiel de la phase 2 . . . . .	121
6.13	L'implémentation de la phase 1 . . . . .	122
6.14	L'implémentation de la phase 2 . . . . .	123
7.1	Deux robots coopératifs dans le plan . . . . .	143

# Liste des tableaux

3.1	Paramètres géométriques du système . . . . .	43
3.2	Paramètres géométriques du robot $j$ . . . . .	44
5.1	Paramètres physique du robot . . . . .	97

# Index

$S_i$	: Vecteur position entre le point $O_j^i$ et le centre de gravité du corps $j$ .
$\theta_j^i, \dot{\theta}_j^i, \ddot{\theta}_j^i$	: resp :Position, vitesse et accélération de l'articulation $j$ du segment $i$ ,
$h_j^i, \dot{v}_j^i \in \mathbb{R}^3$	: resp :Vitesse et accélération linéaire du corps $j$ appartenant au segment $i$ .
$k_j^i, \dot{v}_j^i$	: Premier et deuxieme moment d'inertie
$\Gamma_j^i$	: Force appliquée à l'articulation $j$ du segment $i$ .
$f_j^i, n_j^i \in \mathbb{R}^3$	: Force et moment d'interaction entre les corps $j - 1$ et $j$ du segment $i$ .
$m_i$	: Masse du segment $i$ .
$\mathcal{J}_i$	: Second moment de la masse du segment $i$ autour de son centre de masse.
$c(i, i + 1)$	: Matrice décrivant l'orientation du repère $i + 1$ par rapport au repère $i$ .
$G_i(Q_i)$	:forces gravitationnelles du robot.
$C_i(\dot{Q}_i, Q_i)$	: forces de Coriolis du robot
$[P_{xi}, P_{yi}, P_{zi}]^T$	:la position de l'effecteur terminal du robot i
$[\theta_i, \phi_i, \psi_i]^T$	:l'orientation l'effecteur terminal du robot i
$\mathbf{A}$	: matrice de rotation (chapitre 2); matrice d'inertie du robot (chapitre 5).
$\mathbf{a}_k$	: vecteur unitaire porté par l'axe de l'articulation "k".
$\mathbf{A}(\theta)$	: matrice d'inertie du robot.
$\hat{\mathbf{A}}(\theta)$	: estimation de $\mathbf{A}(\theta)$ .
$\mathbf{a}_i = \dot{\omega}_i$	: accélération de rotation instantanées du corps $C_i$ .
$\mathbf{A}, \bar{\mathbf{A}}$ et $\hat{\mathbf{a}}, \hat{\omega}$	: matrices préproduit.
$b, r, \gamma, d, \theta, \alpha$	: paramètres définissant la position de la plate-forme.
$\mathbf{B}_j$	: matrice antisymétrique formée par $P_{x,j}, P_{y,j}$ et $P_{z,j}$ .
$C_i$	: corps connecté à l'articulation $i$ .
$\mathbf{C}(\theta, \dot{\theta})$	: vecteur de dimension $(n \times 1)$ représentant les couples et forces de coriolis et les forces centrifuges.
$E_j$	: énergie cinétique du corps $C_j$ .

---

$e(t)$	: erreur du vecteur des coordonnées articulaires.
$\dot{e}(t)$	: erreur du vecteur des vitesses articulaires.
$\ddot{e}(t)$	: erreur du vecteur des accélérations articulaires.
$F$ et $M$	: respectivement le vecteur force et le vecteur moment agissant sur le corps du robot.
$G$	: vecteur force agissant sur le robot.
$H(\theta, \dot{\theta})$	: vecteur de dimension $(n \times 1)$ représentant les couples et forces de coriolis, les forces centrifuges ainsi que les forces gravité.
$\hat{H}(\theta, \dot{\theta})$	: estimation de $H(\theta, \dot{\theta})$ .
$I_3$	: matrice identité.
$J_j$	: jacobienne de la robot "j".
$J(q)$	: matrice Jacobienne de dimension $m \times n$ .
$k_p$ ; $k_v$	: matrices des gains, diagonales définies positives de dimension $(n \times n)$ .
$k_{xz} = \frac{F_x}{F_z}$	: rapport global des forces.
$L_{k,n}$	: vecteur d'origine $O_k$ et d'extrémité $O_n$
$m_0$	: la masse de l'objet manipulé.
$OP_j$	: vecteur position liant le point de contact du robot "j" et le centre de gravité de l'objet manipulé par rapport au repère $R$ .
$Q$	: vecteur des couples et forces de gravité.
$R$	: référentiel absolu.
$R_0$	: repère de la plate-forme.
$T$	: matrice de passage homogène.
$U_j$	: énergie potentielle du corps $C_j$ .
$v_i$	: vitesse du centre de gravité $G_i$ .
$W$ , $\overline{W}$ et $\widehat{W}$	: vecteur force/moment résultant sur le corps.
$w(t)$	: vecteur de commande.
$(x_0, y_0, z_0)$	: repère fixé au centre de gravité de l'objet.
$X, Y, Z$	: coordonnées du centre de gravité du robot (origine de $R_0$ ) dans le repère $R$ .
$\ddot{\theta}$	: vecteur des accélérations articulaires.
$\dot{\theta}$	: vecteur des vitesses articulaires.
$\theta$	: vecteur des coordonnées articulaires.
$\omega_i$	: vitesse de rotation instantanée du corps $C_i$ .

---

$\gamma_i = \dot{\mathbf{v}}_i$	: accélération du point $G_i$ .
$\omega_0$	: vitesse angulaire de l'objet.
$\gamma_{0,0}$	: accélération linéaire désirée de l'objet exprimée dans le repère $(x_0, y_0, z_0)$ .
$\omega_{0,0}$	: accélération angulaire désirée de l'objet manipulé.
$\varphi^0$	: matrice d'inertie de l'objet manipulé.
$\Gamma$	: forces et couples généralisés.
$\tau_{jmax}$	: couple articulaire maximum.
$\hat{\tau}_j$	: couple articulaire dû au poids et l'inertie du robot "j".
$\mu$	: coefficient de frottement entre robot-objet.
$P_{i,j}$	: Vecteur position de $O_j$ à $O_i$ .
$k$	nombre de segments composant le robot.
$N$	: Nombre de degré de liberté de chaque segment.
$M$	: Nombre de liaisons de chaque segment.

# Chapitre 1

## Introduction

### 1.1 Motivation et Contexte de la Thèse

Les progrès de la technologie et de la mécatronique ont orienté la robotique moderne vers la conception de systèmes de plus en plus complexes pour la prise en charge de tâches pouvant être extrêmement compliquées. Cette complexité peut se manifester au niveau d'un système robotique individuel qui sera alors composé de nombreux systèmes élémentaires comme c'est le cas des robots humanoïdes. Elle peut aussi se manifester par un système robotique formé par un groupe de robots homogènes ou hétérogènes distribués dans l'espace et destinés à coopérer ensemble. Cette complexité qui s'exprime par le nombre élevé de corps en liaison peut correspondre à une structure ouverte (robot sériel) ou fermée (robot parallèle). Elle peut aussi correspondre à une structure dont la topologie est dynamiquement variable pouvant être ouverte un certain temps puis se refermer et/ou vice-versa comme dans l'exemple de robots coopératifs manipulant en groupe un objet. Cette complexité pose nécessairement de nouveaux problèmes en matière de recherche scientifique et technologique surtout pour des systèmes qui sont destinés à opérer en temps réel tout en interagissant avec leur environnement.

Si on considère les robots manipulateurs sériels qui ont été introduit dans l'industrie depuis plusieurs décennies, il se trouve que la structure robotique la plus utilisée est un système polycaténaire en chaîne ouverte, généralement à six degrés de liberté (ddl). Dans ce contexte ; le choix d'un certain nombre de structures a été sélectionné (Kuka, Puma...). Les modèles géométriques, cinématiques et dynamiques de ces robots sériels sont bien établis. Divers modes de commande et de contrôle ont été étudiés et testés. Les limites et les performances sont bien identifiées.

Une extension naturelle des robots manipulateurs consiste à les utiliser dans un mode coopératif pour mener à bien des tâches qui ne peuvent être effectuées par un seul robot manipulateur comme dans certains cas de la prise, le transport, et la manipulation d'objets. Une telle situation a été envisagée depuis les deux dernières décennies. La topologie de l'ensemble est généralement variable, elle passe de certains états en chaînes ouvertes à des états en chaînes fermées. Pour cette raison, on ne dispose pas de modèles directement adaptés à ces cas complexes.

Plus spécifiquement, il est nécessaire en premier lieu d'établir des modèles géométriques, cinématiques, dynamiques appropriés pour le cas de la coopération en tenant compte des contraintes de l'interaction des robots avec l'objet à manipuler dans son environnement. Ces modèles sont logiquement beaucoup plus complexes que ceux d'un seul robot à cause de la complexité structurelle et de la variabilité topologique surtout pour la gestion temps réel.

Une contrainte majeure apparaît ici à cause du temps de calcul qui devient prohibitif et ne permet pas un contrôle temps réel du groupe de robots en coopération. Afin de vaincre cette contrainte, il est nécessaire en deuxième lieu d'explorer et d'implémenter les approches de calcul parallèle en vue d'accélérer les temps de calculs pour les applications temps réel. Cela peut s'effectuer en implémentant des algorithmes de calcul parallèle en utilisant des réseaux de processeurs tel que permis actuellement par la technologie des transputers et des DSP (Digital Signal Processing). D'où la motivation et le contexte du choix du sujet de cette thèse.

## 1.2 Objectifs de la thèse

Ce travail de recherche concernera plusieurs aspects fondamentaux, à savoir :

- Etudier l'état de l'art concernant la modélisation géométrique, cinématique, et dynamique, dans le cas de la coopération des robots manipulateurs.
- Explorer les techniques de calcul sériel et parallèle pour accélérer les calculs dans le cas de cette coopération.
- Etudier et analyser les systèmes de robots coopératifs manipulant des objets en commun. Cela concerne la modélisation géométrique, cinématique et dynamique prenant en compte l'ensemble de la structure, la variabilité dynamique de sa structure (topologie) et les contraintes de contact avec l'objet et l'environnement.
- L'extension, aux robots manipulateurs présentant des chaînes cinématiques fermées (coopération multi-robots).

- La résolution du problème de distribution temps réel des efforts de contact par résolution du problème d'optimisation quadratique.
- La commande de coordination globale dans l'espace opérationnelle d'un système multi-robots manipulant un objet dans l'espace .
- L'exploitation de l'approche du calcul parallèle par décomposition des tâches dans le cas du modèle dynamique de Newton-Euler.
- L'exploitation de l'approche de calcul parallèle par décomposition des matrices dans le cas du modèle dynamique de Newton-Euler. Le choix de la méthode proposée par Fijani a été exploitée à cause du  $O(\log N)$  par décomposition de Shur.

### 1.3 Contributions

Les contributions apportées par ce travail peuvent se résumer dans les points suivants :

1. Etude, et simulation des systèmes de robots coopératifs à savoir la modélisation géométrique, cinématique et dynamique approprié à ce genre de système , avec prise en compte de la saisie manipulé .
2. La résolution du problème de distribution des efforts de contact sur les robots manipulateurs sous contraintes de saisie et de non glissement de l'objet manipulé. Les contraintes dégagées ont été intégrés au problème de la distribution optimale d'effort entre les deux bras en minimisant un critère quadratique .
3. La formulations de la dynamique et la commande de coordination globale dans l'espace opérationnelle d'un système multirobots manipulant un objet dans l'espace . Dans ce sens un simulateur sous Matlab a été réalisé ce qui nous a permis de tester le modèle et enfin des simulations ont été illustrées pour vérifier l'approche proposée.
4. Exploitation de l'approche de calcul parallèle par décomposition et allocation des tâches à un système multiprocesseurs ou le nombre de processeur est choisi par l'utilisateur. Les simulations ont été réalisées sous Matlab et la visualisation de l'implémentation de l'algorithme de décomposition par Labview. Enfin une estimation du temps de calcul multiprocesseurs / monoprocesseurs a été dégagée.
5. Exploitation de l'approche de calcul parallèle par décomposition matricielle et allocation des tâches à un système multiprocesseurs ou le nombre de processeur est imposé par la taille du système. Enfin des exemples sont traités.

## 1.4 Organisation du manuscrit

Le but de ce travail est d'étudier la commande dans l'espace opérationnel de deux robots coopératifs présentant six degré de liberté chacun, saisissant un objet commun. Après une introduction générale cette these est organisée comme suit :

- **Dans le deuxième chapitre** on a exploré dans une première partie les travaux de recherche en matière de modélisation géométrique, cinématique et dynamique des robots coopératifs, suivi d'une deuxième partie de l'état de l'art concernant les approches algorithmiques pour le calcul sériel (monoprocasseur) et parallèle (multiprocasseur) applicable aux systèmes robotiques manipulateurs en structure ouverte et fermée.

**Le troisième chapitre** est consacré à :

- La modélisation géométrique directe (MGD) et inverse (MGI) des robots à structures ouvertes simples ou ramifiées, dans cette optique on utilise la méthode de Khalil-Kleinfinger pour calculer le modèle géométrique direct. La résolution par la suite du modèle géométrique inverse du système coopératif, nous permettra de transformer la trajectoire opérationnelle suivie par le centre de gravité de l'objet manipulé à la trajectoire articulaire correspondante.
- Détermination de la Jacobienne cinématique du robot, passant par le calcul du modèle cinématique direct (MCD) et inverse (MCI) qui expriment la vitesse de l'organe terminal en fonction des vitesses articulaires et inversement, et qui servira ultérieurement pour le calcul des vitesses opérationnelles résultant du mouvement du système multi-robots. Afin de résoudre le problème de distribution d'efforts sur l'objet manipulé par les robots.
- **Le quatrième chapitre** est dédié à l'optimisation des efforts de contact ou nous avons posé les contraintes associées à la prise de l'objet sans glissement. Les contraintes dégagées ont été intégrées au problème de la distribution optimale d'effort entre les deux bras en minimisant un critère d'énergie. Le problème d'optimisation ainsi formulé a été résolu en programmation mathématique non-linéaire sous contraintes.

**Le cinquième chapitre** traite la dynamique et la commande du système multi-robots coopératif. Cette étude a permis de dégager dans un premier temps les couples nécessaires qui doivent être fournis par les actionneurs en utilisant différents formalismes de la dynamique. Ces résultats sont utilisés pour le contrôle global de l'objet manipulé en utilisant la commande par découplage non linéaire de l'objet.

- **Le sixième chapitre** de cette thèse est consacré à l'analyse de la complexité algorithmique en vue d'améliorer l'algorithme de Newton Euler [2] et l'exploitation du parallélisme pour réduire le temps de calcul d'un  $O(N)$  dans le cas séquentiel à un  $O(\log(N))$  dans le cas parallèle avec  $O(N)$  processeurs. Dans ce sens la stratégie qui sera suivie est l'exploitation de l'approche du calcul parallèle par décomposition et allocation des tâches à un système multiprocesseurs.
- **Le septième chapitre** sera consacré à l'exploitation de l'approche de calcul parallèle par décomposition matricielle et allocation des tâches à un système multiprocesseurs et des exemples seront traités.

Cette thèse se termine par une conclusion générale où l'accent sera mis sur l'établissement des critères permettant de générer automatiquement des phases transitoires lors d'une modification éventuelle de l'environnement, la détermination d'autres méthodes de contrôle de la stabilité de l'objet. À ce sujet, l'utilisation de la logique floue est une approche à envisager. La préhension sur objet mous ou fortement déformables. D'autant plus que l'efficacité algorithmique de cette méthode augmente avec le nombre de degrés de liberté, d'où l'intérêt de l'application et l'extension du calcul parallèle pour les systèmes plus complexes comme le cas des systèmes présentant des structures arborescentes ou des architectures à chaînes fermées.

# Chapitre 2

## Etat de l'art sur les robots manipulateurs coopératifs

### 2.1 Introduction

Le problème de la coopération multirobots est un sujet très important en robotique moderne et couvre un grand nombre de domaines et d'approches. Un groupe de robots peut coopérer pour exécuter une tâche commune en étant localisé dans un même espace mobile ou pas. Un robot humanoïde, une main artificielle à plusieurs doigts, un robot mobile d'exploration transportant un bras manipulateur, un groupe de robots manipulateurs transportant ou manufacturant une pièce constituent des exemples de robots coopératifs. Un autre domaine assez spécifique mais non moins important est celui de la coopération de groupes formées d'un nombre relativement élevé de robots homogènes est celui de la robotique collective qui s'intéresse à imiter les mécanismes de coopération et de coordination entre agents comme pour les colonies animales (essaims d'oiseaux, bancs de poissons, ..). Les applications se trouvent à la fois dans la fabrication industrielle, le transport, les systèmes militaires autonomes, etc.

Les premiers exemples de systèmes de coopération sont modélisés au cas par cas car il n'existe pas de formalisme général vu les particularités et les conditions de travail. Les essaims de robots sont souvent modélisés dans le cadre de l'approche multi-agents.

Dans le contexte de notre travail, on s'intéressera particulièrement à la coopération des robots manipulateurs homogènes de type industriel manipulant et/ou opérant sur un objet ensemble. Ce problème peut invoquer de nombreux aspects comme la modélisation, la coordination, le

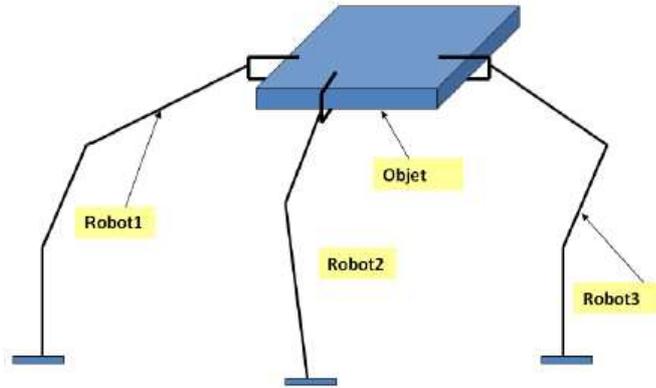


FIG. 2.1 – Système multi-robots coopératifs

contrôle, la prise d'objets, et l'implémentation de la commande en temps réel.

De manière générale Le terme système coopératif signifie la coordination simultanée de plusieurs participants engagés dans l'exécution d'une tâche bien spécifiée. En robotique, par exemple, le terme système coopératif est entendu comme un ensemble de robots manipulateurs saisissant un objet dans l'espace Fig (2.1). Le but principal d'un système coopératif en robotique est de manipuler un objet. La manipulation est réalisée dans le but de [1] :

- Changer la position dans l'espace d'un objet (le transférer d'un endroit à un autre).
- Suivre une trajectoire donnée de l'objet pour une orientation donnée le long de la trajectoire.
- Effectuer des travaux sur un objet fixe ou mobile.

Dans ce chapitre, on explorera et on synthétisera dans une première partie les travaux de recherche en matière de modélisation et en commande des robots manipulateurs coopératifs. Elle sera suivie dans la deuxième partie de l'état de l'art concernant les approches algorithmiques permettant d'accélérer les calculs pour les calculateurs sériels (monoprocésseurs) et parallèles (multiprocésseurs) applicables aux systèmes robotiques manipulant des objets en commun en structure ouverte et fermée.

## 2.2 Approches de modélisation des robots coopératifs

Le travail coopératif intègre le travail commun des différents robots, la coordination de leurs actions lors de l'exécution des tâches, le contact avec l'objet et l'environnement. Cette

coordination implique que chaque robot coopérant doit être capable de travailler simultanément tout en tenant compte de l'évolution des états des autres robots.

Le contrôle d'un système robotique coopératif en temps réel nécessite des lors [1] :

1. Une modélisation mathématique a la fois géométrique, cinématique et dynamique incluant également les forces de contact des robots avec l'objet tout en assurant la force closure voir [2].
2. Une stratégie de coopération des robots doit être élaborée en vue de permettre un mouvement coordonné et simultané de l'ensemble des robots et de l'objet.
3. On effectue une synthèse des lois de commande appropriées.
4. De plus, il faut assurer et sélectionner une architecture d'un calculateur qui élaborera les commandes vers les actionneurs afin d'exécuter la tâche définie.

Le problème de la coopération multi-robots reste un problème d'actualité et dans sa grande partie un problème ouvert [3]-[4]-[5]-[6]. L'intérêt d'utiliser plusieurs robots pour manipuler un même objet est très crucial pour l'industrie d'aujourd'hui où la précision de positionnement est le contrôle d'effort sont important. Dans ce contexte, plusieurs approches ont été développées. La plupart des travaux en modélisation et commande en coopération des robots manipulateurs dans l'état de l'art concerne la manipulation à deux bras. Ceci est particulièrement intéressant à la fois du point de vue pratique et théorique [7]. Du point de vue côté pratique, des tâches bi-manuelles seraient les plus courantes en rapport avec l'analogie des robots humanoïdes utilisant deux bras pour manipuler des objets. D'un point de vue théorique, le cas de deux robots manipulateurs est le cas particulier le plus simple de coopération a partir duquel on peut essayer de généraliser les résultats.

Dans la suite de cette section, on présentera les principales approches proposées pour résoudre le problème de manipulation à bras multiples.

En termes de modélisation, comme constaté par Jorge Gudino et al [8] , une analyse dynamique d'un système de plusieurs manipulateurs est présentée dans [9], où le formalisme de Newton-Euler pour les mécanismes à chaîne ouverte est étendu aux systèmes de chaînes fermées.

Une autre approche couramment utilisée est la méthode d'Euler-Lagrange [7]. Pour cette deuxième méthode, les équations de mouvement pour chaque bras manipulateur sont développées dans l'espace cartésien et la chaîne fermée est considérée lorsque l'objet tenu est en contact avec un milieu rigide comme le sol. Une autre approche générale pour obtenir le modèle dynamique d'un système de plusieurs robots en coopération est basée sur l'estimation de la

matrice de préhension (grasping Matrix) [10],[11], [12],[13] et [14] . Ici, la matrice de préhension est utilisée pour relier la dynamique des manipulateurs avec celle de l'objet, tout en utilisant la formulation de Newton-Euler.

L'analyse dynamique de coopération de robots manipulateurs avec des joints flexibles tenant un objet rigide est présentée par Jankowski et al dans [15].

Un modèle dynamique pour un système de robots coopératifs tenant un objet rigide sans friction est présenté dans [8]. Il a été expérimenté sur deux robots industriels. Le modèle dynamique des manipulateurs sont obtenus indépendamment l'un de l'autre à l'aide de la méthode de Lagrange. Dès que les robots prennent l'objet, leurs variables articulaires sont alors cinématiquement et dynamiquement couplées. En considérant que le couplage du système est décrit par des contraintes holonomes, les équations décrivant les modèles dynamiques des manipulateurs et de l'objet sont combinées pour obtenir le modèle dynamique global du système, qui peut être utilisé pour la simulation du comportement du système.

En termes de modélisation et de contrôle, une brève synthèse de ces recherches est rapportée par [16] et [17], qui notent qu'au cours des trente dernières années, de nombreuses recherches ont été menées sur plusieurs systèmes de manipulateurs. [16] et [17] ont noté que les multiples manipulateurs provoquent des contraintes internes dans l'objet manipulé, et typiquement un système de commande avec contrôle de force doit être utilisé en vue de minimiser ces forces.

[18] a présenté un système de contrôle à l'aide  $N$  robots avec  $m$  ddl chacun et rigidement relié à un objet manipulé en commun. Il a appelé la résultante du système qui est, l'effecteur terminal plus l'objet à manipuler, l'objet augmenté (the augmented object), étant donné que la description de l'ensemble du système prend en compte les caractéristiques d'inertie des effecteurs et de l'objet.

Cet objet augmenté est soumis à une force  $f_0$  dans le plan opérationnel qui s'exerce en un point opérationnel  $x_0$  de l'objet. Compte tenu des contraintes du modèle de l'objet augmenté, Khatib a écrit les équations du système de mouvement analogue à sa formulation dans l'espace opérationnel [19], fournissant une généralisation élégante dans l'espace opérationnel. Le contrôle est alors effectué dans l'espace opérationnel. Pour tenir compte des forces et des moments internes intervenants dans la manipulation d'un objet manipulé par de multiples robots coopératifs, D. Williams et O. Khatib [20] ont présenté "the virtual linkage model". Ce modèle concerne les robots manipulant l'objet et donc en situation de chaîne fermée. Ce modèle relie virtuellement les points de contact des robots à l'objet. Il a été appliqué à un système formé de deux robots

PUMA 560.

une autre approche appelée 'Schéma de contrôle symétrique' a été proposée (Uchiyama et Dauchez [21]) pour assurer la coordination de deux robots manipulant un objet en commun. La définition des vecteurs de forces, vitesses et positions sont considérées comme des fonctions symétriques des vecteurs correspondants aux forces, vitesses et positions dans l'espace articulaire est essentielle pour implémenter ce schéma symétrique. En utilisant une analyse statique de la chaîne cinématique fermée et en supposant que la déformation de l'objet était très petite, on dérive les vecteurs de l'espace de travail et puis on calcule les vitesses au moyen de la dualité force-vitesse. Les positions sont ensuite obtenues par intégration des vitesses. Le résultat comprend quatre variables significatives : les positions relatives et absolues ainsi que les orientations relatives et absolues entre les bras manipulateurs. Yamano et al [22] ont proposé d'autres extensions du schéma de contrôle symétrique pour tenir compte des bras manipulateurs flexibles. Une autre approche appelée "cooperative task-space", considère le sens inverse de la méthode du schéma de contrôle symétrique en définissant un espace composé des quatre variables précitées [23] et [24]. L'avantage de cette technique tient au fait que l'on peut relaxer les hypothèses d'un objet fermement tenu ; permettant l'application du modèle de prise d'objets flexibles. Mais à cause des forces internes et externes, un contrôle robuste du système nécessite l'usage du contrôle de forces. La théorie des quaternions et des dual quaternions a été utilisée dans l'analyse des trajectoires, Connolly et Pfeiffer [25], Dooley et McCarthy [26], Bruno Vilhena Adorno [17] Adorno Bruno Vilhena Philippe et al [27].

De nombreux autres travaux sont cités relevant de la modélisation ; du contrôle et des simulations de trajectoires.

Cao et al [28] ont présenté une approche sous optimale en temps minimum par inspection et planification de tâches. Dans cette approche les auteurs ont proposé un algorithme qui permet d'obtenir une séquence d'actions en temps minimum par l'utilisation d'un profil d'accélération articulaires. Le minimum obtenu n'est pas global.

Ahmadabadi et al [29] ont introduit le concept de mouvement contraint pour la résolution du problème de coopération.

Une nouvelle approche pour le calcul de la fermeture des forces de saisie d'objet dans le cas Bidimensionnel et tridimensionnel, a été proposée dans [2],[30],[31] où de nouvelles conditions nécessaires et suffisantes ont été implémentées afin d'assurer la fermeture des forces de saisies. D'autres approches ont été proposées pour la résolution de ce problème en utilisant la program-

mation mathématique linéaire [32],[33] et [34]. Nahon et Angeles [32] ont proposé une approche utilisant la programmation quadratique par minimisation des efforts internes aux mécanismes. Par ailleurs, Kwon et Lee [35] ont proposé une approche basée sur une distribution optimale d'effort sur les robots manipulateurs par utilisation de la dualité en programmation mathématique non-linéaire. Ce type d'approche a été aussi utilisé par Nakamura et al [36] pour la formulation et résolution du problème de coopération. Polen et Park [37] ont proposé une formulation de la coopération dynamique par l'utilisation de théorie des groupes de Lie. Le problème de la coopération de deux robots manipulateurs ou problème DUAL-ARM a été traité dans la littérature par plusieurs auteurs. L'inconvénient majeur des approches proposées reste leur spécificité aux types de robots utilisés et la difficulté de généraliser l'approche à plus de deux robots manipulateurs. Ainsi Gu et al [38] ont présenté une méthode basée sur la compensation dynamique. Hu et Goldenberg [39] ont proposé une commande optimale par minimisation des couples articulaires mis en oeuvre, ainsi qu'une approche adaptative pour la commande hybride effort/position [40]. Hogan [41] a étudié le problème de contact alors que la commande en impédance a été proposée par Schneider et al [34]. D'autres approches basées sur la théorie de Lyapounov ont été présentées [42]. Dans [43], Bonitz et Hsia ont étudié le problème du calcul du minimum d'effort interne nécessaire au maintien de l'objet à partir des contraintes de friction imposées et des efforts mesurés. L'algorithme d'optimisation en boucle fermée proposé est très indiqué pour une implémentation en temps réel. Le contrôleur utilise une mesure des moments au points de contact pour maintenir l'orientation désirée de l'objet manipulé. La dynamique non-linéaire de chaque robot manipulateur est compensée par un contrôleur auxiliaire robuste. Liu et AbdelMalek [4] ont proposé une approche basée sur l'utilisation d'un contrôleur robuste. Les contraintes de contact et de frottement ont été utilisées et un algorithme d'optimisation non-linéaire par minimisation de l'énergie a été proposé.

Dans [44] une analyse détaillée de la saisie d'objets déformables par une main à trois doigts a été réalisée, et il a été prouvé que les forces internes requises pour saisir des objets déformables varient en fonction de la taille de l'objet et l'angle de contact du doigt. Xydas et al [45] ont élaboré un software basé sur la méthode des éléments finis pour interpréter le contact mécanique des doigts. Dans [46], Hirai et al ont proposé une loi de commande robuste pour la préhension et la manipulation d'objets déformables. Ils ont développé une loi de commande pour saisir et manipuler un objet déformable à l'aide d'un système de vision en temps réel. Cutkosky [47] a montré que la stabilité de la prise est une fonction des modèles de contact. Mason et

Salisbury [48] ont donné des conditions à la retenue complète d'un objet en termes de forces internes. Kerr et Roth [49] ont proposé une méthode pour optimiser les forces internes basées sur l'approximation des contraintes de frottements entre l'objet saisi et les différents doigts. Meer et al [50], Patton et al [51] ont conçu un système qui met l'accent sur le contrôle de la force en fonction de l'ajustement de la déformation. Dans ces systèmes, le robot manipulateur est appelé à contrôler la déformation de l'objet. Dans [52] Howard et Bekey ont développé un algorithme généralisé d'apprentissage pour la manipulation des objets 3-D déformables dont la connaissance préalable des attributs de l'objet n'est pas nécessaire. Ils ont utilisé des réseaux de neurones admettant la masse, la constante d'amortissement et la raideur du ressort comme entrées et la force nécessaire pour saisir l'objet en tant que sortie du réseau.

La figure (2.2) montre un exemple de problème traité dans [53]. Le transfert dans le plan d'une charge maintenue par simple contact est optimisé compte tenu des contraintes d'unilatéralité et de non-glissement aux points de contact. Hirano et al.[54] considèrent la planification de

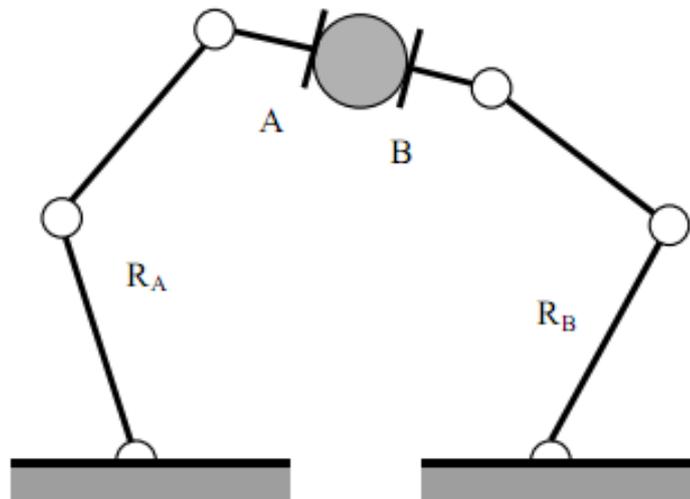


FIG. 2.2 – Manipulation d'un objet avec double contact unilatéral en A et B [53]

trajectoires de deux manipulateurs dont l'un tient un objet tandis que l'autre effectue un travail sur cet objet figure (2.3). Seule la trajectoire de l'outil sur l'objet est spécifiée. Les deux manipulateurs ajustent alors leurs mouvements pour minimiser les efforts actionneurs tout en respectant la trajectoire de l'outil sur l'objet ainsi que des contraintes de contact. La figure (2.4) représente le schéma de principe de deux robots manipulateurs universels 6R, coopérant

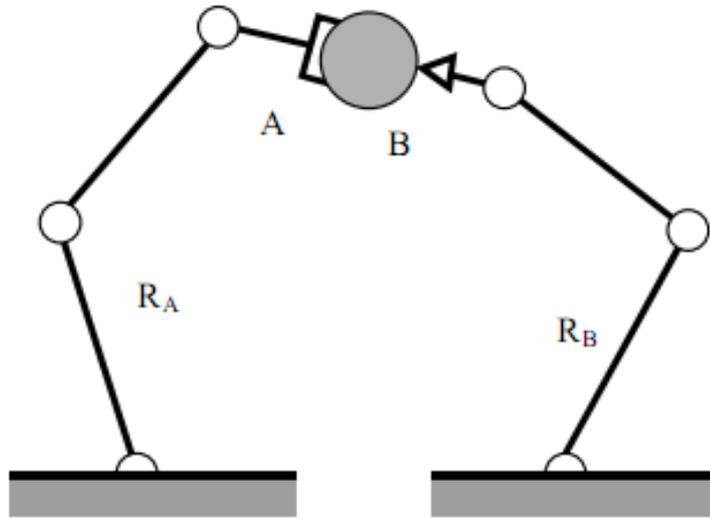


FIG. 2.3 – Travail ponctuel avec contact unilatéral de  $R_B$  en B sur un objet présenté par le manipulateur  $R_A$  [54]

pour le déplacement d'une charge [35]. La trajectoire de l'objet transporté est spécifiée. Le problème traité consiste à optimiser les forces de serrage de l'objet et les vitesses articulaires des manipulateurs durant le mouvement. Le système fonctionne ainsi de manière analogue à une pince de grande dimension. Les deux manipulateurs sont rigidement liés. Le système bouclé comporte douze articulations actionnées, mais ne dispose que des six degrés de liberté de la charge déplacée. Il est donc fortement sur-actionné.

## 2.3 Approches du calcul parallèle pour les robots coopératifs

La complexité des équations différentielles et algébriques modélisant les systèmes robotiques manipulateurs en coopération manipulant un objet en commun augmente avec le nombre de corps en liaison. En particulier, cette complexité pose des problèmes de délais lors de la résolution de ces équations pour effectuer des simulations mais surtout lors de la coordination et de la commande temps réel de ces robots.

Le besoin d'accélérer les calculs en vue de réduire le temps nécessaire pour la génération des commandes aux différents actionneurs afin de pouvoir réaliser un fonctionnement en temps

r el de ces robots s'est fait sentir des le d epart. Un effort consid erable a  et e continuellement soutenu par les chercheurs et ce depuis de nombreuses d ecennies en vue d'adapter les mod eles robotiques aux exigences du calcul rapide permettant d'effectuer des simulations offline dans des d elais les plus courts et permettant surtout la commande effective en temps r el. Ces efforts ont du s'adapter pour b en eficier de l' evolution technologique en mati ere a la fois du software et du hardware.

L'aspect fondamental de cette exigence ne peut  etre lev e que par le d evveloppement et par l'adaptation des mod eles de robotique pour les traduire en algorithmes destin es a  etre impl ement e pour s'ex ecuter sur des calculateurs disponibles dans des d elais minimums. Ces mod eles sont les mod eles g eom etriques, cin ematiques, dynamiques avec les contraintes de coop eration, et de contact avec les objets et l'environnement. Des le d ebut de la robotique industrielle, un remarquable et intensif effort a  et e d eploy e en vue de minimiser les temps de calcul dans le cas ou on dispose de calculateurs s equentiels de type monoprocesseurs. L'adaptation des algorithmes bas es sur Newton-Euler et Lagrange a permis d'optimiser les calculs pour r eduire la complexit e de l'ordre  $O(N^4)$  a l'ordre  $O(N)$ . Cette limitation de la complexit e  $O(N)$  semble  etre ind epassable a ce jour. Elle ne peut donc  etre essentiellement am elior ee que par l'accroissement des vitesses du processeur lui m eme qui d epend actuellement de la technologie des semi-conducteurs et de l'organisation architecturale.

Lorsque la complexit e du syst eme robotique est telle que l'usage de ces algorithmes ne permet pas d'obtenir des d elais permettant le calcul temps r el, les chercheurs ont port e leurs efforts en vue d'exploiter les architectures multiprocesseurs disponibles technologiquement sur le plan du hardware a savoir les transputers, les DSP, les clusters et les supercalculateurs avec les syst emes d'exploitation et les langages de programmation appropri es. Ces moyens plus ou moins couteux permettent effectivement d'am eliorer de fa con importante les d elais d'ex ecution des calculs exig es pour un fonctionnement temps r el.

Un autre facteur qui milite en faveur du calcul parall ele dans les syst emes robotiques en coop eration et qui n'est pas souvent not e dans les analyses de ce sujet tient au fait que les diff erents robots doivent agir en synchronisme et simultan ement. Ceci requi ere donc naturellement le d evveloppement d'architectures multiprocesseurs distribu es.

La suite de ce texte pr esentera bri evement un  etat de l'art sur ce sujet. Elle est donc d ediee aux approches algorithmiques pour le calcul s eriel (monoprocesseur) et parall ele (multiprocesseur) applicable aux syst emes robotiques. Dans ce sens, une synth ese de l' etat de l'art dans

ce domaine particulier montre qu'il ya deux approches globales concernant le développement d'algorithmes pour les architectures de calculateurs parallèles multiprocesseurs.

1. Une première approche s'intéresse aux développements d'algorithmes permettant de construire des modèles dans lesquels on peut effectuer une décomposition de l'algorithme en tâches élémentaires. Les modèles en question sont précisément les modèles dynamiques qui peuvent être obtenus par divers formalismes physiques (Lagrange, NE, etc). Les tâches élémentaires sont déterminées par le concepteur qui choisi le nombre de processeurs adéquat. Il affecte également par programmation les différentes tâches élémentaires aux différents processeurs tout en assurant leur coordination. Le nombre de processeurs est choisi selon un compromis tenant compte de la complexité de l'architecture/prix/temps de calcul final désiré. Cette approche est très pratique et effective pour le cas d'un nombre limité de robots en coopération (1, 2 et 3 robots) et s'adapte bien a une architecture de transputers. Elle permet donc de réduire le temps de calcul qui est de l'ordre  $O(N)$  dans le cas du calcul sériel.
2. Une deuxième approche s'intéresse aux développements d'algorithmes qui exploitent essentiellement la décomposition matricielle des modèles cinématiques (matrice Jacobienne directe et inverse) et dynamiques (matrice d'inertie directe et inverse). Cette approche est basée sur le principe de diagonalisation matricielle permettant en dernier lieu l'obtention de systèmes d'équations découplées pouvant être résolues par un réseau de processeurs. Pour cette approche, le nombre de processeurs est généralement du même ordre que le nombre de ddl du système robotique. Pour le modèle dynamique, certains types de décomposition comme la méthode de Shur, ont permis une amélioration de l'ordre  $O(\log N)$  en utilisant un réseau de  $N$  processeurs.

On présentera dans ce qui suit, l'évolution de l'état de l'art concernant les algorithmes se rapportant aux modèles dynamiques des robots.

Avec les calculateurs sériels mono-processeurs, le premier algorithme synthétisant la solution de la dynamique directe et inverse utilise la technique de Lagrange-Euler. Il a une complexité de calcul élevée de l'ordre de  $O(N^4)$ , ou  $n$  est le nombre de liaisons du bras du robot, mais il s'agit d'une formulation bien structurée et systématique. Le cout de calcul prohibitif de cet algorithme rend son implémentation en temps réel impossible, même pour un robot avec un petit nombre de degrés de liberté. Des algorithmes de la dynamique directe dont la complexité est de l'ordre de  $O(N^3)$  ont été développés, soit  $O(N^3)$  opérations de calcul [55],[56]. Pour

résoudre les accélérations articulaires, ces méthodes utilisent la stratégie suivante [57] :

- a) D'abord, les éléments de la matrice d'inertie sont évalués ;
- b) Ensuite, la décomposition de la matrice d'inertie est calculée numériquement par la décomposition de Cholesky ;
- c) Enfin, les accélérations articulaires sont résolus par des substitutions amont et en aval [58], respectivement. Comme la méthode de Cholesky est  $O(N^3)$ , l'algorithme de la dynamique est de l'ordre de  $O(N^3)$ .

Cependant, les approches alternatives qui résolvent le problème de la dynamique directe par des algorithmes itératifs de NE ont été également développées. Ils ont une complexité de calcul de l'ordre de  $N$ , soit  $O(N)$  permettant qu'ils soient efficacement implémentable

Le premier algorithme synthétisant la solution de la dynamique directe et inverse est de complexité  $O(N^4)$  ou  $n$  étant la taille du problème, correspondant au nombre de degrés de liberté (DDL) du robot manipulateur. Le coût de calcul prohibitif de cet algorithme rend son implémentation en temps réel impossible, même pour un robot avec un petit nombre de degrés de liberté. Motivé par ce besoin et au cours des dernières années les chercheurs ont développé des algorithmes et architectures parallèles performants pour atteindre la puissance de calcul nécessaire. Les algorithmes parallèles pour chaque problème spécifique avec les exigences architecturales associées sont bien maîtrisées aujourd'hui. Nous avons maintenant des algorithmes parallèles qui permettent d'atteindre le temps limite inférieure à  $O(\text{Log}N) + O(l)$  afin de résoudre des problèmes de calcul robotiques avec un nombre minimal de processeurs asymptotique.

Dans [59], une approche basée sur la méthode des systèmes multi-corps dans le cas dynamique des robots en vue de paralléliser les calculs. La stratégie utilisée dans ce contexte était de partitionner les équations symboliques du mouvement en modules indépendants pour l'évaluation simultanée. Les stratégies mises en oeuvre pour la parallélisation comprennent l'évaluation parallèle des équations de sous-systèmes, le calcul parallèle de la matrice d'inertie ainsi que sa factorisation, les vecteurs de force et les entrées de commande. La mise en place des structures parallèles est discutée en ce qui concerne l'utilisation optimale des ressources transputer. L'approche présentée donne une forte réduction du temps de calcul et prend en charge les simulations en temps réel de la commande des robots. Dans cet article la méthode est appliquée sur le robot STANFORD.

Dans [60] Dominik HENRICH et al, ont présenté différentes approches du calcul parallèle

concernant la cinématique directe et inverse ainsi que différentes techniques de parallélisation de la Jacobienne du robot. De ce fait ils ont montré que les approches du parallélisme utilisées dans ce contexte permettent d'avoir un temps d'exécution très court avoisinant 50 ms, permettant alors des applications en temps réel dans le cas de la commande des robots manipulateurs.

La formulation des modèles dynamiques a été un domaine de recherche actif pour les deux dernières décennies et plusieurs méthodes ont été développées par les chercheurs [61]. Parmi ces méthodes, les plus couramment utilisés sont la méthode de Lagrange-Euler (LE) et la méthode de Newton-Euler (NE). La technique (LE) a une complexité de calcul élevé de l'ordre de  $O(N^4)$ , où  $N$  est le nombre de liaisons du bras du robot, mais il s'agit d'une formulation bien structurée et systématique. D'autre part, la formulation (NE) est efficacement implémentable, de l'ordre de  $O(n)$ , mais avec un ensemble d'équations extrêmement récursifs. [62] ou le nombre de processeurs est déterminé par l'utilisateur.

En outre, Barhen [63] a réparti la dynamique d'un bras de robot présentant 6 liaisons en 66 tâches de calcul. Ces tâches ont été intégrées dans un ensemble de calcul parallèle appelé (Hypercube). Deux modes de fonctionnement ont été présentés. Dans le mode séquentiel tous les calculs ont lieu sur le processeur Intel 80286/80287 processeurs résidant sur le sous-système périphérique Ncube, tandis que dans le mode simultané des équations (tâches) sont résolus sur les noeuds hypercube.[64].

Featherstone [65] a proposé un algorithme de complexité  $O(\log(N))$ . Cette méthode consiste à construire l'équation de la dynamique par assemblage successif de corps simples. L'approche est appliquée aux robots présentant des structures fermées et arborescentes. Fijany [66] présente une méthode globale de parallélisation, dans le cadre de la dynamique des systèmes multi-corps. L'algorithme parallèle CF (Constraint Force) développé est également d'une complexité en  $O(\log(N))$ .

Des algorithmes parallèles  $O(\log(N))$  sont basés sur la décomposition des forces d'interactions résultant une nouvelle factorisation de la matrice de masse  $\mathcal{M}_i$  du système et de son inverse sous forme de complément de Schur :  $\mathcal{M}_i^{-1} = \mathcal{C}_i - \mathcal{B}_i^t \mathcal{A}_i^{-1} \mathcal{B}_i$  où  $\mathcal{A}_i$ ,  $\mathcal{B}_i$  et  $\mathcal{C}_i$  sont des matrices de blocs tridiagonale. Cette forme de décomposition matricielle est très adaptable au parallélisme des calculs

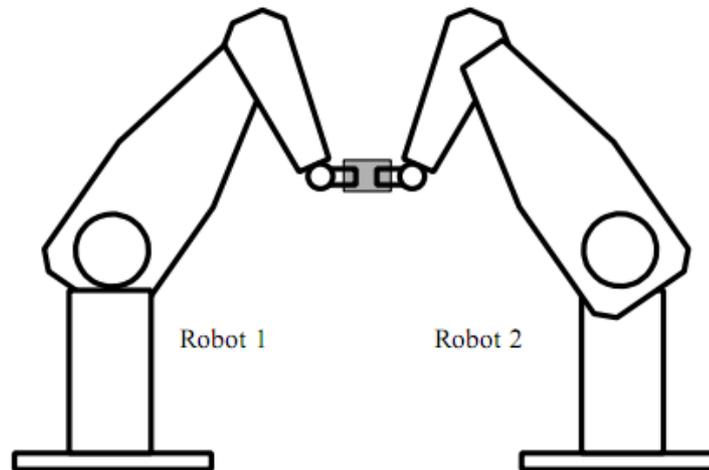


FIG. 2.4 – Robots manipulateurs 6R, coopérants pour le transfert d'une charge [35].

Cette partie est dédiée aux approches algorithmiques pour le calcul sériel (monoprocasseur) et parallèle (multiprocasseur) applicable aux systèmes robotiques. La formulation des modèles dynamiques a été un domaine de recherche actif pour les deux dernières décennies et plusieurs méthodes ont été développées par les chercheurs [61]. Plusieurs algorithmes parallèles ont été proposées par des chercheurs en vue de calculer la dynamique d'un système mécanique [67]. L'idée de base est de diviser l'équation de la dynamique régissant le mouvement du système par exemple les équations de Newton-Euler pour le cas d'un robot manipulateur en des tâches simples. Ensuite, ces tâches doivent être réparties de façon optimale sur un nombre donné de processeurs dans le but d'avoir un temps d'exécution le plus rapide que possible. voir [68] et [64].

## 2.4 Conclusion

Pour les systèmes manipulateurs coopératifs, la réduction des temps de calcul ne pourra pas se passer d'algorithmes parallèles nécessaire au calcul intensif. Par suite de l'analyse de l'état de l'art nous allons opter pour la méthode de Zomaya [62], et aussi de la méthode de décomposition matricielle telle que présentée par Fijani.

on va utiliser en premier lieu la méthode de Zomaya [62] qui est une méthode basée sur la décomposition des tâches sur les processeurs en vue de paralléliser les équations de Newton-

Euler pour la dynamique d'un robot manipulateur 6ddl présentant une liaison prismatique. L'avantage de cette methode permet a l'utilisateur (concepteur) de déterminer lui même le nombre de processeur. La methodologie sera egalement discutee pour le cas de son adaptation aux cas de plusieurs robots cooperatifs.

On utilisera en second lieu, la methode de prallelisation des calculs par la methode de Fijani qui donne une meilleure acceleration des calculs mais qui necessite un reseau de N processeurs pour un systeme a N ddl .

# Chapitre 3

## Modélisation géométrique et cinématique

### 3.1 Introduction

La description géométrique d'un robot est une étape importante dans toutes les étapes de l'élaboration des modèles, qu'ils soient géométriques, cinématiques ou dynamiques. C'est pourquoi nous abordons dans ce chapitre la paramétrisation géométrique des robots dans le cas général des structures ouvertes simples et des structures arborescentes, puis nous définissons les différents modèles géométriques et cinématiques qui seront utilisés dans les chapitres suivants pour la modélisation dynamique des robots coopératifs. Pour effectuer cette modélisation le robot est considéré comme constitué d'un ensemble de liaisons interconnectés entre elles aux différentes articulations. La topologie du système polyarticulé définit le type de manipulateur, soit un manipulateur série ou parallèle. Pour chaque type, plusieurs approches permettant d'obtenir les modèles géométriques et cinématiques ont été proposées. Dans la littérature plusieurs méthodes et notations ont été proposées pour la description des paramètres géométriques des structures en série, arborescentes ou comportant des boucles fermées, mais aucune méthode systématique n'a été proposée pour la description des robots coopératifs (ou parallèles). A la fin de ce chapitre nous présentons la modélisation géométrique du système multi-robot coopératif ainsi qu'une application numérique argumentée par des figures illustratives simulant le système en mouvement.

### 3.1.1 Structure ouverte simple

La méthodologie à suivre pour décrire les robots à structure ouverte simples consiste à considérer un manipulateur série composée de  $n + 1$  corps notés  $C_0, C_1, \dots, C_n$  et de  $n$  articulations. Le corps  $C_0$  désigne la base du robot et le corps  $C_n$  porte l'organe terminal. L'articulation  $j$  connecte le corps  $C_j$  au corps  $C_{j-1}$  [69], figure (3.1) .

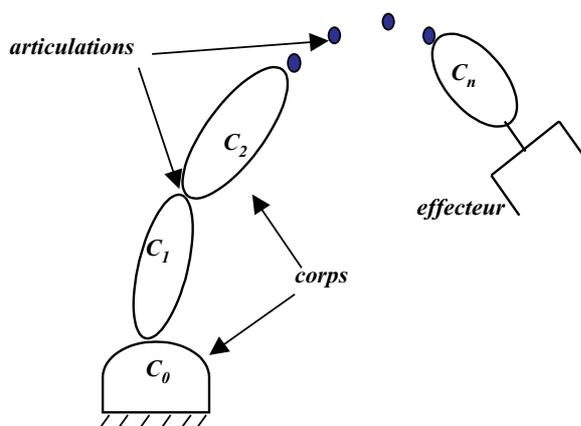


FIG. 3.1 – Robot à structure ouverte simple

### 3.1.2 Structure arborescente et fermée (chaînes complexes)

#### 3.1.3 Définition

Une structure arborescente fermée se distingue par l'existence d'un ou plusieurs corps pouvant porter plus d'une liaison avec la partie en aval de la chaîne, ce qui génère l'apparition de plusieurs organes terminaux. Une boucle est caractérisée par le retour de la structure vers la base  $C_0$  ; Une chaîne peut avoir plusieurs boucles.

Par convention, les corps et les articulations sont numérotés de la manière suivante, figure (3.2) :

- la base constitue le corps  $C_0$
- Les numéros des corps et des articulations sont croissants sur chaque branche en partant de la base vers un organe terminal ;
- Le corps  $C_j$  est articulé par l'articulation  $j$ , autrement dit, l'articulation  $j$  connecte le

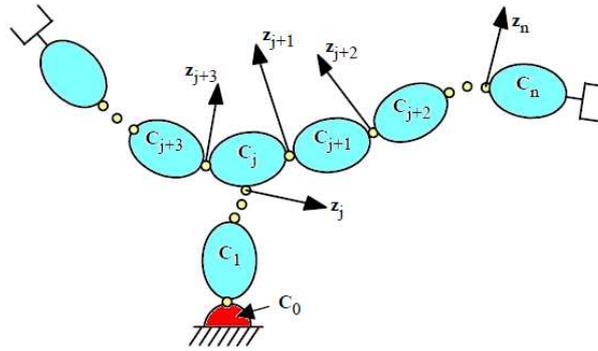


FIG. 3.2 – Schéma d'une structure arborescente

corps  $C_j$  au corps  $C_{a(j)}$ . Le corps  $C_{a(j)}$  est le corps antécédent du corps  $C_j$  lorsque l'on parcourt la chaîne de la base.

### Paramètres géométriques

Pour déterminer les paramètres géométriques nécessaires à la détermination des transformations entre les différents repères liés aux corps, on place les repères de la manière suivante, figure(3.3) :

- $R_i$  est fixé par rapport au corps  $C_i$
- $Z_i$  est porté par l'axe de l'articulation "i"
- $X_i$  est porté par la perpendiculaire commune à  $Z_i$  et à l'axe  $Z$  de l'un des corps en aval porté par le corps "i" Comme la figure(2.3) le montre, deux cas se présentent :

**1<sup>er</sup> cas** : Si  $X_i$  est la perpendiculaire commune à  $Z_i$  et  $Z_j$  alors le passage de  $R_i$  à  $R_j$  s'écrit comme dans le cas des chaînes simples, à l'aide des quatre paramètres géométriques ( $\alpha_j, d_j, \theta_j$  et  $r_j$ ).

${}^i T_j = Rot(X, \alpha_j)trans(X, d_j)Rot(Z, \theta_j)trans(Z, r_j)$  c'est le cas du corps "k" par rapport au corps "i".

**2<sup>eme</sup> cas** : Si  $X_i$  est la perpendiculaire commune à  $Z_i$  et à un autre axe  $Z_k$ , on construit la perpendiculaire commune  $X_{i'}$  aux axes  $Z_i$  et  $Z_j$ , le passage de  $R_{i'}$  (défini par  $X_{i'}$  et  $Z_i$ ) s'effectue à l'aide des quatre paramètres usuels ( $\alpha_j, d_j, \theta_j$  et  $r_j$ ). Pour le passage de  $R_i$  à  $R_{i'}$  deux paramètres supplémentaires doivent être introduits :

$\gamma_j$  : angle entre  $X_i$  et  $X_{i'}$  autour de  $Z_i$  ;

$b_j$  : distance entre  $X_i$  et  $X_{i'}$  le long de  $Z_i$ .

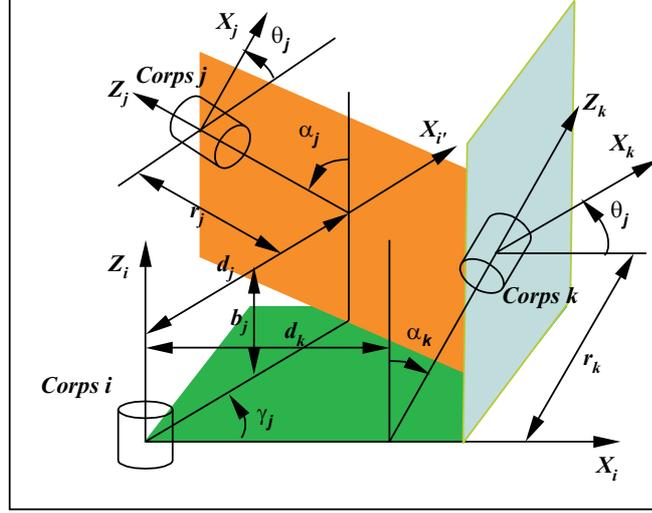


FIG. 3.3 – Articulation comportant une ramification

Ces six paramètres permettent de construire la matrice de transformation suivante :

$${}^i \mathbf{T}_j = \text{Rot}(Z, \gamma_j) \text{trans}(Z, b_j) \text{Rot}(X, \alpha_j) \text{trans}(X, d_j) \text{Rot}(Z, \theta_j) \text{trans}(Z, r_j) \quad (3.1)$$

$${}^{j-1} \mathbf{T}_j = \begin{pmatrix} C\gamma_j C\theta_j - S\gamma_j C\alpha_j S\theta_j & -C\gamma_j S\theta_j - S\gamma_j C\alpha_j C\theta_j & S\gamma_j C\alpha_j & d_j C\gamma_j + r_j S\gamma_j S\alpha_j \\ S\gamma_j C\theta_j + C\gamma_j C\alpha_j S\theta_j & -S\gamma_j S\theta_j + C\gamma_j C\alpha_j C\theta_j & -C\gamma_j S\alpha_j & d_j S\gamma_j - r_j C\gamma_j S\alpha_j \\ S\gamma_j S\theta_j & S\gamma_j C\theta_j & C\alpha_j & r_j C\alpha_j + b_j \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.2)$$

La transformation inverse  ${}^j \mathbf{T}_i$  s'écrit :

$${}^j \mathbf{T}_i = \begin{pmatrix} & -b_j S\alpha_j S\theta_j - d_j C\theta_j & & \\ & -b_j S\alpha_j C\theta_j + d_j S\theta_j & & \\ & -b_j S\alpha_j - r_j & & \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (3.3)$$

**Remarque :** La relation (3.1) représente la forme générale de la matrice homogène de transformation. Si  $b_j$  et  $\gamma_j$  sont nuls cela correspond au cas des chaînes ouvertes simples.

## 3.2 Modèle géométrique direct des robots à structure arborescente

Dans le cas d'une chaîne arborescente, il y a autant de chaînes que de corps terminaux. la matrice de passage pour une branche de l'arborescence portant le corps terminal "k" est la composition des matrices de transformation exprimant le repère  $R_k$  de ce corps dans le repère de base  $R_0$ .

$${}^0T_k = {}^0T_1 {}^1T_2 \dots {}^{a(k)}T_{a(k)} {}^{a(k)}T_k \quad (3.4)$$

### 3.2.1 Application au système multi-robot coopératif

Avant de donner dans ce paragraphe le modèle géométrique direct (MGD) et inverse (MGI), considérons l'architecture du système multi-robot manipulateur manipulant simultanément un même objet. Comme les robots sont similaires (symétriques), seule la configuration d'un robot suffira pour décrire le système coopératif donné. Les  $n$  robots manipulateurs ainsi que l'objet à saisir sont représentés en figures (3.4) et (3.5). Chaque robot est localisé à une distance  $a_j$  mesurée par rapport au système de référence global  $(X, Y, Z)$ . L'angle  $\phi_j$  représente l'orientation du système de coordonnées  $(x_{1,j}, y_{1,j}, z_{1,j})$  lié à la base du robot et le système de référence  $(X, Y, Z)$ . Le système multi-robot manipulateurs est assimilé à une structure arboréscence comportant des chaînes fermées, et la méthode de Khalil et Kleinfinger [70], dédiée à ce genre de structures [69] est utilisée.

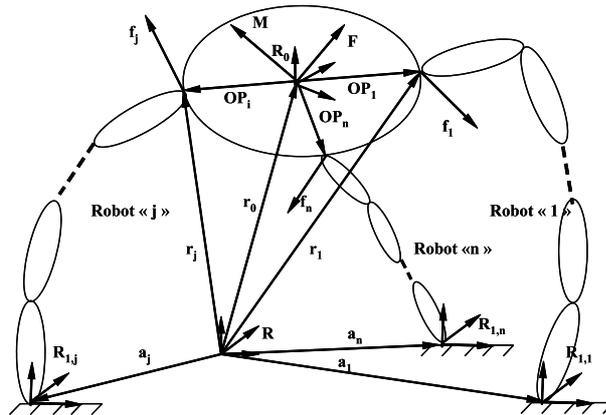


FIG. 3.4 – Paramètres géométriques du système multi-robot

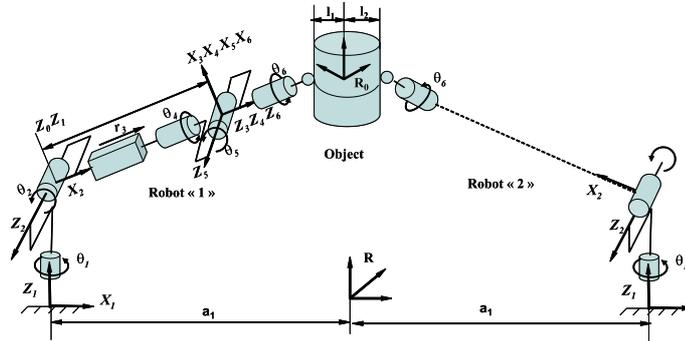


FIG. 3.5 – Exemple d'un système multi-robot manipulateurs

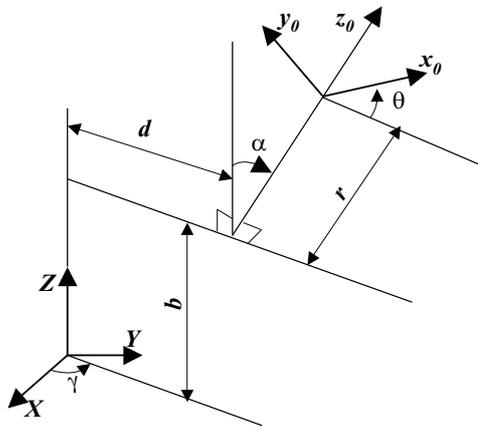


FIG. 3.6 – Modél géométrique

Le robot isolé et sa configuration est représenté. La transformation matricielle permettant le passage du repère  $R_i$  au repère  $R_{i-1}$  est donnée par :

$${}^{i-1}T_i = R(Z, \gamma)T(Z, b)R(X, \alpha)T(X, d)R(Z, \theta)T(Z, r) \quad (3.5)$$

La table(3.1) décrit la transformation du repère global  $(X, Y, Z)$  au repère local pris au point de contact "6" de chaque robot "j",  $j = 1, \dots, n$  avec l'objet manipulé .

La transformation matricielle donnant la position du point "6" correspondant à l'organe terminal de chaque robot est donnée par :

$${}^R T_6 = {}^R T_0 T_1^1 T_2^2 T_3^3 T_4^4 T_5^5 T_6 \quad (3.6)$$

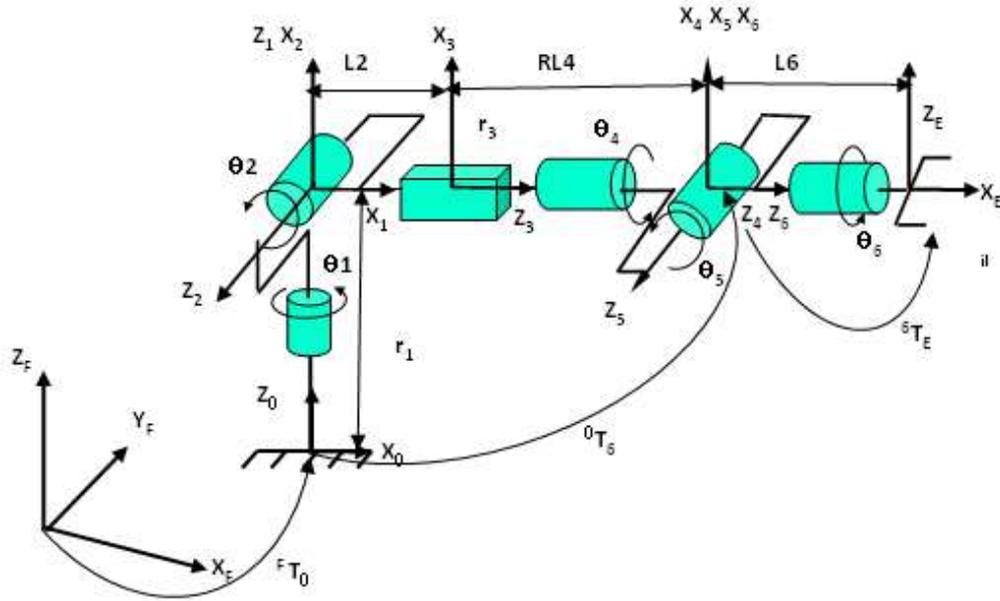


FIG. 3.7 – Configuration du robot et choix des repères

frame	$\alpha$	d	$\theta$	r	b	$\gamma$
object	$\alpha$	d	$\theta$	r	h	$\beta$
robot(1)	0	$OP_1$	$\theta_{1,1}$	0	0	0
robot"j"	0	$OP_j$	$\theta_{1,j}$	0	0	0
robot"n"	0	$OP_n$	$\theta_{1,n}$	0	0	0

TAB. 3.1 – Paramètres géométriques du système

### 3.3 Modél géométrique direct du robot :

#### 3.3.1 Definition :

Le modèle géométrique direct d'un robot manipulateur permet d'exprimer la situation de l'organe terminal du robot manipulateur en fonction de sa configuration définie par  $n$  coordonnées généralisées  $[\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6]^T$ .

Alors le modél géométrique direct(M.G.D) du robot manipulateur est décrit comme suit :

$$X = \mathcal{F}(\theta) \tag{3.7}$$

frame	$\alpha$	d	$\theta$	r	b	$\gamma$
joint"1"	0	$a_j$	$\theta_1$	0	0	$\phi_j$
joint"2"	0	0	$\theta_2 + \pi/2$	0	0	0
joint"3"	$\pi/2$	0	0	$r_3$	0	0
joint"4"	0	0	$\theta_4$	$r_4$	0	0
joint"5"	$-\pi/2$	0	$\theta_5$	0	0	0
joint"6"	$\pi/2$	0	$\theta_6$	0	0	0

TAB. 3.2 – Paramètres géométriques du robot  $j$

ou  $:X$  désigne le vecteur des coordonnées opérationnelles.

Pour notre étude, le système multi-robots manipulateurs coopératifs proposé, est composé de deux jusqu'à  $n$  robots seriels saisissant un objet en commun, la chaîne cinématique et la configuration de ce type de robots est représentée en fig(3.7). Le robot comporte 6 articulations dont une est prismatique. Les coordonnées articulaire sont représentées par le vecteur  $q = [\theta_1, \theta_2, r_3, \theta_4, \theta_5, \theta_6]^T$ . Le calcul sous Maple nous a permis de caculer les matrices de passage suivantes :

$$\begin{aligned}
 {}^F T_0 &= \begin{pmatrix} C\phi_0 & -S\phi_0 & 0 & a_j \\ S\phi_0 & C\phi_0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad {}^6 T_E = \begin{pmatrix} 0 & 0 & 1 & 0 \\ 0 & -1 & 0 & 0 \\ 1 & 0 & 0 & L6 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
 {}^0 T_1 &= \begin{pmatrix} C_1 & -S_1 & 1 & 0 \\ S_1 & C_1 & 0 & 0 \\ 0 & 0 & 1 & r_1 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad {}^1 T_2 = \begin{pmatrix} -S_2 & -C_2 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ C_2 & -S_2 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
 {}^2 T_3 &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & -1 & -L_2 - r_3 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, \quad {}^3 T_4 = \begin{pmatrix} C_4 & -S_4 & 0 & 0 \\ S_4 & C_4 & 0 & 0 \\ 0 & 0 & 1 & RL_4 \\ 0 & 0 & 0 & 1 \end{pmatrix}
 \end{aligned}$$

$${}^4T_5 = \begin{pmatrix} C_5 & -S_5 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -S_5 & -C_5 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}, {}^5T_6 = \begin{pmatrix} C_6 & -S_6 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ S_6 & C_6 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

Avec :

$$C_j = \cos(\theta_j) \text{ et } S_j = \sin(\theta_j)$$

Par extraction du vecteur  $X$ , représentant les coordonnées opérationnelles de la matrice  ${}^F T_E$  on obtient :

$$\begin{cases} p_x = dS_1\phi_0 + eC_1\phi_0 + d1 \\ p_y = -dC_1\phi_0 + eS_1\phi_0 \\ p_z = bL_6 + S_2(L + r_3) + r_1 \end{cases} \quad (3.8)$$

$$\begin{cases} \varphi = \text{atan} \left[ \frac{aS_1\varphi_0 - S_4S_5C_1\varphi_0}{aC_1\varphi_0 + S_4S_5S_1\varphi_0} \right] \\ \theta = \text{atan} \left[ \frac{-b}{\sqrt{1-b^2}} \right] \\ \psi = \text{atan} \left[ \frac{cS_6 + C_2S_4C_6}{cC_6 + C_2S_4S_6} \right] \end{cases} \quad (3.9)$$

ou :

$p_x, p_y$  et  $p_z$  représentent la position en coordonnées cartésiennes.

$\varphi, \theta$  et  $\psi$  désignent l'orientation en fonction des angles de cardans.

Avec :

$$\begin{cases} L = RL_4 + L_2 & a = C_2C_5 - S_2C_4S_5 \\ C_1\phi_0 = \cos(\theta_1 + \phi_0) & b = C_2C_4S_5 + S_2C_5 \\ S_1\phi_0 = \sin(\theta_1 + \phi_0) & c = C_2C_4C_5 - S_2S_5 \\ e = aL_6 + C_2(L + r_3) & d = L_6S_4S_5 \end{cases} \quad (3.10)$$

**Remarque :**  $L_6$  est la longueur de l'effecteur terminal le long de l'axe  $Z_{6,j}$

## 3.4 Modèle géométrique inverse du robot (M.G.I)

### 3.4.1 Introduction

Nous avons vu au paragraphe précédent que le modèle géométrique direct(M.G.D) d'un robot permet de calculer les coordonnées opérationnelles, donnant la situation de l'organe terminal en fonction des coordonnées articulaires. Le problème inverse consiste à calculer les coordonnées

articulaires correspondant à une situation désirée de l'organe terminal du robot, cette forme représente le modèle géométrique inverse (MGI).

La méthode utilisée est celle de Paul [71] qui traite séparément chaque cas particulier et convient pour la plupart des robots. Lorsqu'il n'est pas possible de trouver une forme explicite du modèle géométrique inverse, on peut calculer une solution particulière par des procédures numériques. La solution obtenue est locale en ce sens qu'elle dépend des conditions initiales.

### 3.4.2 Calcul du modèle géométrique inverse par la méthode de Paul

La méthode est valable pour chaque robot individuellement. En réalité, le MGI du système multi-robots doit fournir une solution lorsque le centre de gravité de l'objet manipulé est contraint à suivre une trajectoire spécifiée. Soit un robot "j" du système coopératif présenté dans notre étude, alors la matrice de transformation homogène donnée par le MGD est comme suit :

$${}^R \mathbf{T}_6 = {}^R \mathbf{T}_0 \mathbf{T}_1 \mathbf{T}_2 \mathbf{T}_3 \mathbf{T}_4 \mathbf{T}_5 \mathbf{T}_6 \quad (3.11)$$

Lorsque la position et l'orientation  $U_0$  du repère lié à l'effecteur terminal de chaque robot "j" sont connues avec :

$$U_0 = \begin{bmatrix} sx & nx & ax & Px \\ sy & ny & ay & Py \\ sz & nz & az & Pz \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.12)$$

Pour trouver les solutions de l'équation (3.11), Paul [71] a proposé une méthode qui consiste à prémultiplier successivement les deux membres de l'équation (3.11) par les matrices  ${}^{i-1} \mathbf{T}_i^{-1}$ . Ces opérations permettent d'isoler et d'identifier les variables articulaires l'une après l'autre. Le calcul nous donne les expressions des coordonnées articulaires  $\theta_{i,j}$  ( $i = 1..6$ ) ( $j = 1..n$ ) :

$$\left\{ \begin{array}{l} \theta_{1,j} = \arctan(Py/(-Px + d_1)) \\ \theta_{2,j} = \arctan(Pz - r_1)/ \\ \quad (C1(Px - d_1) + S1Py)) \\ r_{3,j} = C2(C1(Px - d_1) + S1Py) + \\ \quad S2(Pz - r_1) - R4 - l_2 \\ \theta_{4,j} = \arctan((S1ax - C1ay)/ \\ \quad (S2(C1ax + S1ay) - C2az)) \\ \theta_{5,j} = \arctan((C4S2(C1ax - S1ay) - \\ \quad C4C2az + S4(C1ay - C1sy))/ \\ \quad (C2(C1ax + S1ay) + S2az)) \\ \theta_{6,j} = \arctan((-S4(-S2(C1ax - S1ay) + \\ \quad C2sz + C4S1sx - C1sy))/ \\ \quad (C5(-C4S2C1sx - S2S1sy + C2sz) \\ \quad + S4(S1sx - C1sy) - \\ \quad S5(C1C2sx + S1C2sy + S2sz))) \end{array} \right.$$

**Remarque :**  $S^* = \sin(*)$ ;  $C^* = \cos(*)$ ;  $(P_x = P_{x,j}, P_y = P_{y,j}, P_z = P_{z,j})$ , représentent les coordonnées articulaires du point "6" du robot  $j$  exprimées dans le repère  $(X, Y, Z)$ .

### 3.4.3 Application dans le cas de la coopération multi-robots :

Pour pouvoir valider le générateur de trajectoire imposé au centre de gravité de l'objet manipulé, Nous allons nous intéresser plus spécialement à l'évolution de l'objet dans l'espace lorsque ce dernier est maintenue par trois robots. Le cadre de simulation concerne le suivi d'une trajectoire imposé en faisant varier individuellement ou simultanément les paramètres  $\alpha$ ,  $d$ ,  $\theta$  et  $r$ , qui représentent respectivement la rotation autour de  $ox$ , translation suivant l'axe  $ox$ , la rotation suivant l'axe  $oz$  et la translation suivant l'axe  $oz$ .

#### Paramètres et résultats de la simulation :

Les résultats de la simulation montrent quelques mouvements de l'objet manipulé, qui sont visualisés en figures (3.8), (3.10), (3.12) et (3.14).

- La figure (3.8) montre l'évolution en mouvement de translation suivant l'axe  $X$  selon une trajectoire  $x = d_x = d + 500t$ .

- La figure (3.10) illustre le mouvement de translation suivant l'axe  $Z$  selon l'équation  $r_z = r + 500t$ .
- La figure (3.12) visualise le mouvement de translation et de rotation respectivement suivant l'axe  $Z$  et  $X$  avec  $r = r_Z = R + 500t$  et  $\alpha_x = \alpha + 300t$
- La figure (3.14) montre l'évolution de l'objet dans l'espace faisant un mouvement combiné de translation et de rotation combiné autour de l'axe  $Z$  respectant les équations  $\theta_z = \theta + 200t$  et  $r_z = r + 500t$ .

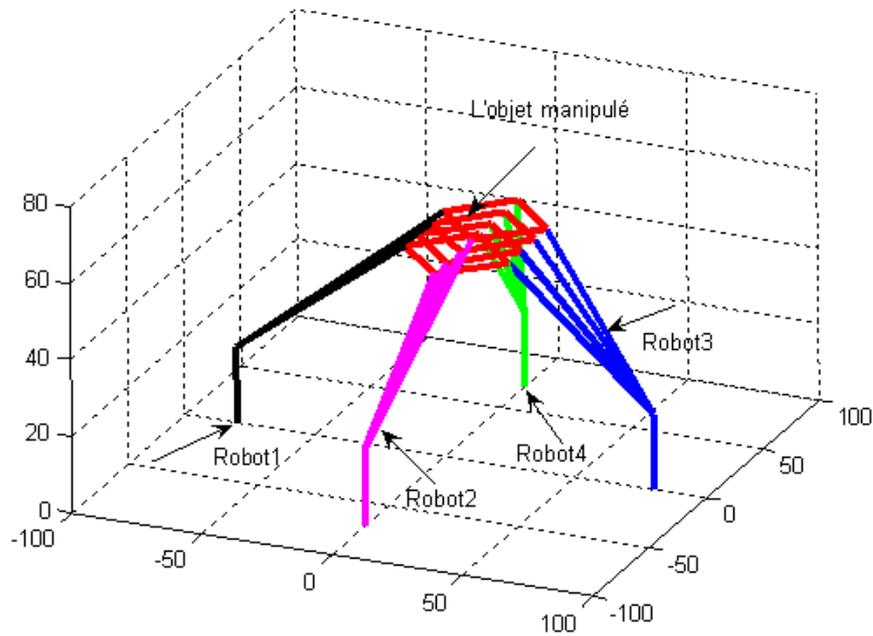


FIG. 3.8 – Translation de l'objet suivant l'axe X

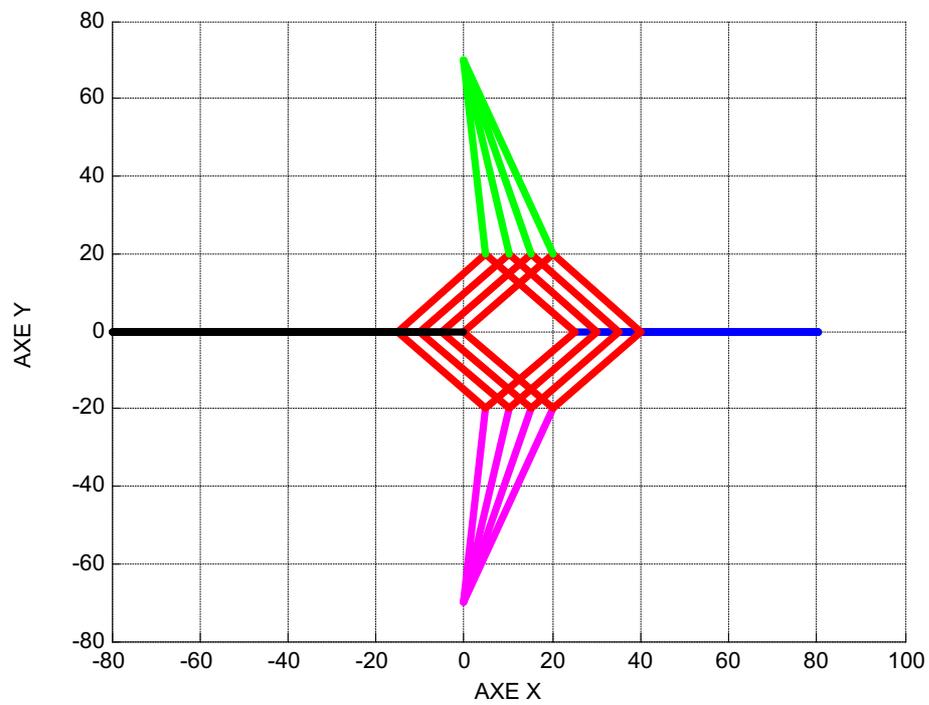


FIG. 3.9 – Visualisation du mouvement dans le plan XOY

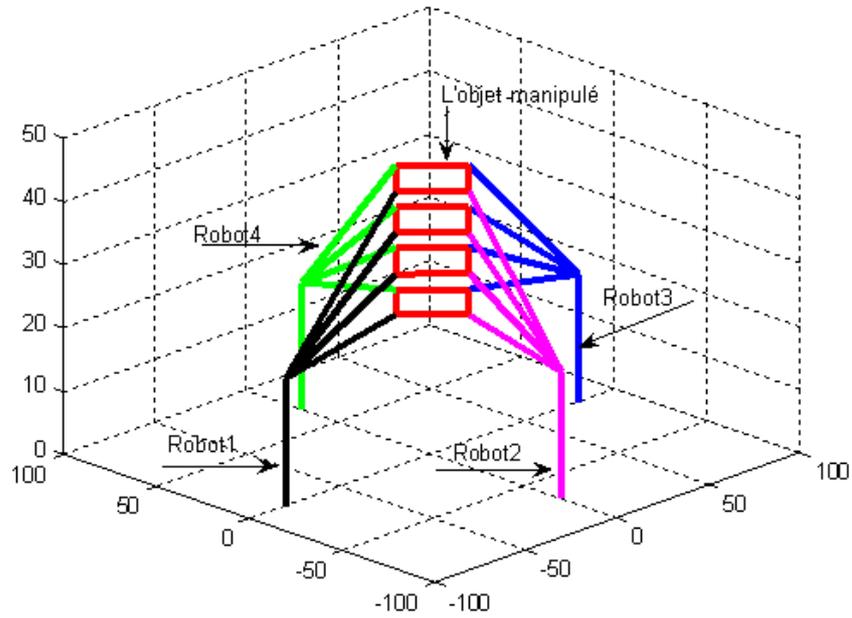


FIG. 3.10 – Translation de l'objet suivant l'axe Z

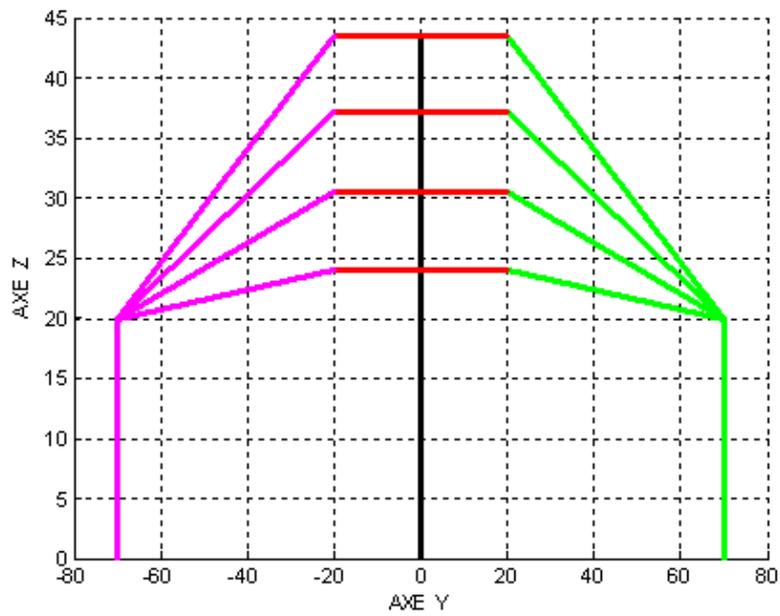


FIG. 3.11 – Visualisation du mouvement dans le plan YOZ

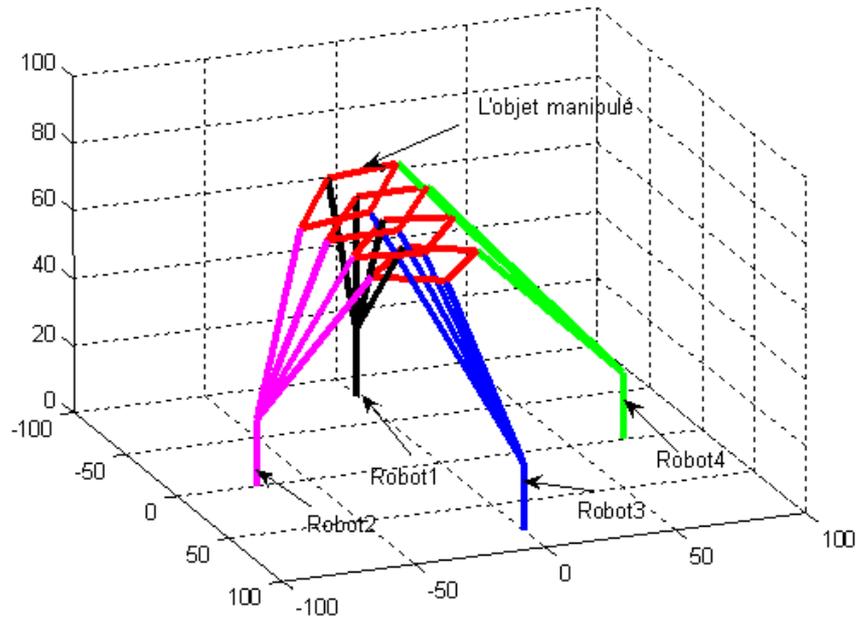


FIG. 3.12 – Translation et rotation respectivement autour de Z et X

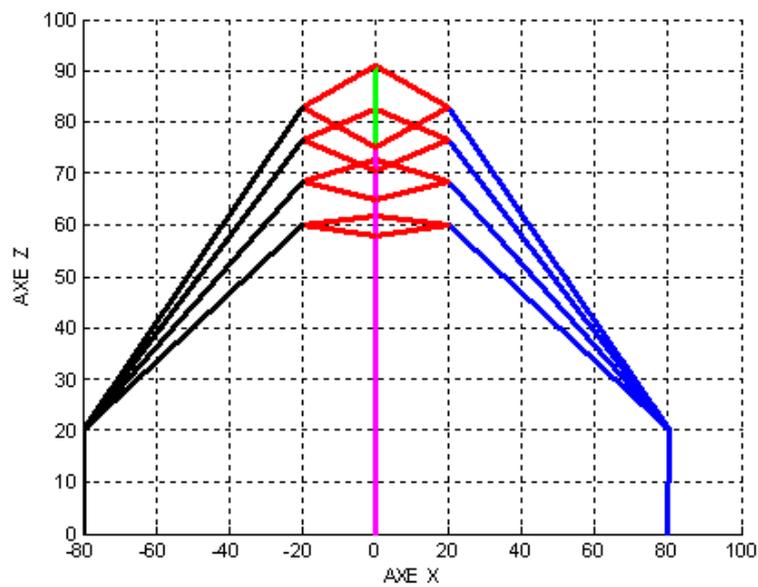


FIG. 3.13 – Visualisation du mouvement dans le plan XOZ

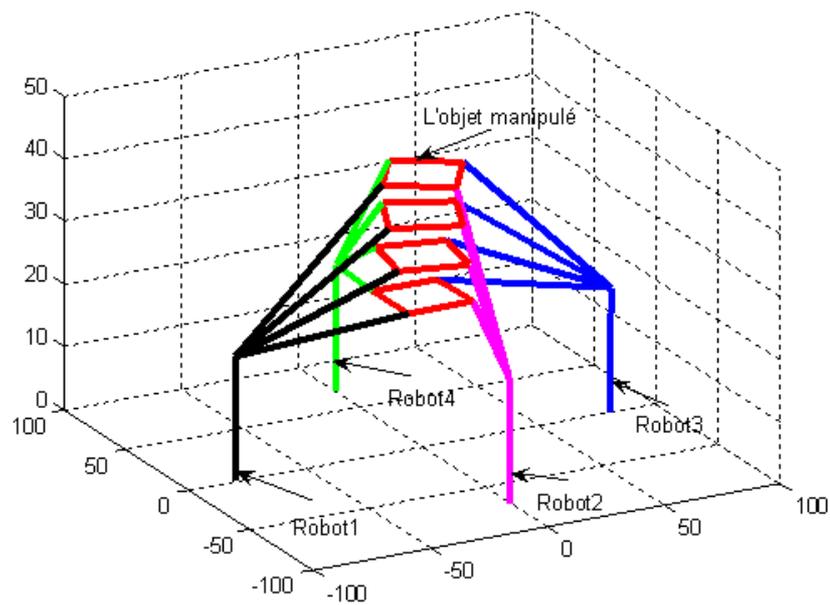


FIG. 3.14 – Translation et rotation combinée autour de Z

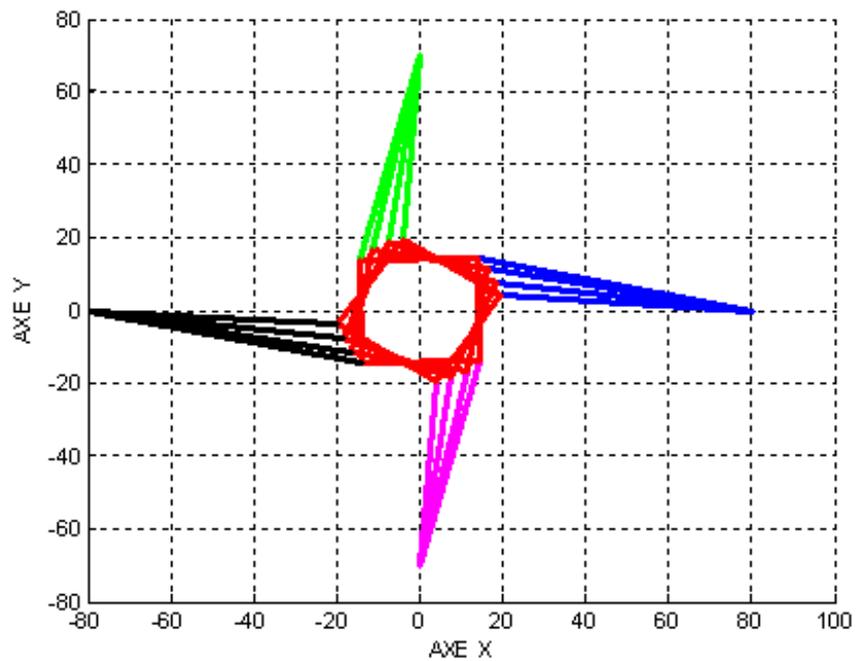


FIG. 3.15 – Visualisation du mouvement dans le plan XOY

## 3.5 Etude cinématique

Le modèle cinématique exprime les relations entre les vitesses articulaires de chaque liaison et les vitesses cartésiennes d'un corps de la chaîne cinématique, généralement l'organe terminal. Ce modèle est donc un modèle par accroissements infinitésimaux : chaque variation élémentaire de la valeur d'une articulation implique une variation de position de l'organe terminal, et inversement. Lorsque ces variations infinitésimales sont exprimées par rapport au temps, on peut les considérer comme des vitesses.

Le modèle cinématique permet donc non seulement de compléter éventuellement le modèle géométrique en tenant compte des vitesses, mais aussi de remplacer le modèle géométrique : en agissant par accroissements successifs, on peut se déplacer d'un point donné à un autre.

### 3.5.1 Le modèle cinématique direct (MCD)

Le MCD d'un robot, décrit les vitesses des coordonnées opérationnelles en fonction des vitesses articulaires, il est noté :

$$\dot{\mathbf{X}} = \mathbf{J}(\mathbf{q})\dot{\mathbf{q}} \quad (3.13)$$

où  $\mathbf{J}(\mathbf{q})$  désigne la matrice Jacobienne de dimension  $m \times n$  du mécanisme qui est fonction de la variable articulaire  $\mathbf{q}$ . Cette matrice est à la base du calcul du modèle cinématique, elle facilite aussi le calcul des singularités et la dimension de l'espace opérationnel accessible du robot.

#### Calcul de la matrice Jacobienne de base :

Si on écarte les méthodes de calcul symbolique permettant de dériver les équations du modèle géométrique direct, on peut utiliser une méthode très répandue pour le calcul cinématique, qui permet d'obtenir la matrice jacobienne par un calcul direct fondé, d'une part, sur la relation entre les vecteurs des vitesses de translation et de rotation  $\mathbf{V}_n$  et  $\omega_n$  du repère  $\mathbf{R}_n$  et les vitesses articulaires  $\dot{\mathbf{q}}$ , qui est donnée par :

$$\underline{\mathbf{V}}_n = \begin{bmatrix} \mathbf{V}_n \\ \omega_n \end{bmatrix} = \mathbf{J}_n \dot{\mathbf{q}}, \quad (3.14)$$

et d'autre part sur l'influence que produit chaque articulation d'ordre  $k$  de la chaîne sur le repère terminal  $\mathbf{R}_n$ . L'influence de la  $k^{ieme}$  articulation de vitesse  $\dot{\mathbf{q}}_k$  est qu'elle induit sur le repère terminal  $\mathbf{R}_n$  les vitesses  $\mathbf{V}_{k,n}, \omega_{k,n}$  figure (3.16) :

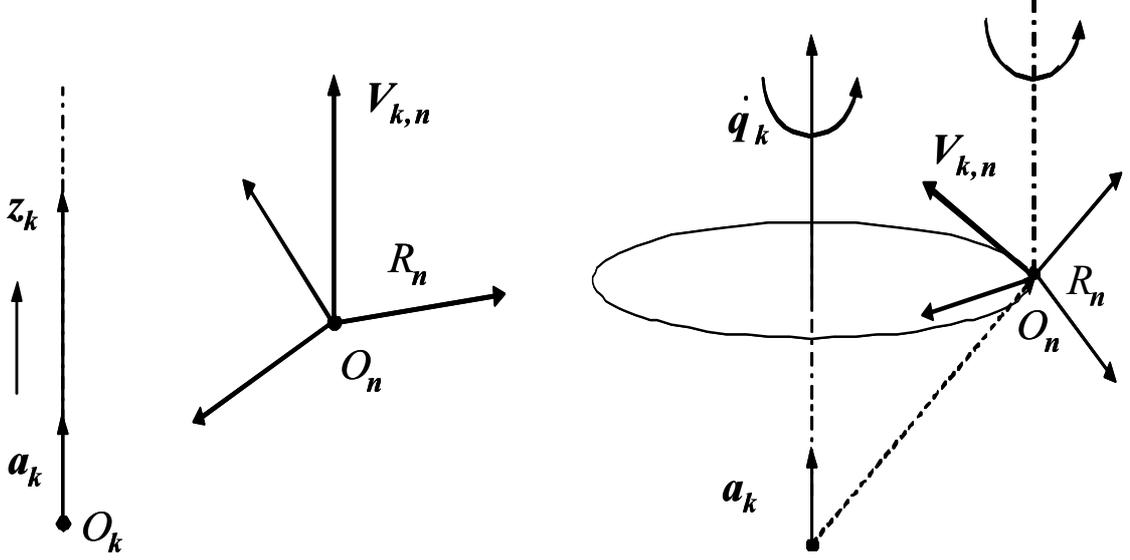


FIG. 3.16 – Influence du type de l'articulation sur le repère terminal

- dans le cas d'une articulation prismatique  $\sigma_k = 1$  :

$$\begin{cases} \mathbf{V}_{k,n} = \mathbf{a}_k \dot{q}_k \\ \omega_{k,n} = 0 \end{cases} \quad (3.15)$$

où :

$\mathbf{a}_k$  : est le vecteur unitaire porté par l'axe  $\mathbf{a}_k$  de l'articulation  $k$ ;

- dans le cas d'une articulation rotoïde  $\sigma_k = 0$  :

$$\begin{cases} \mathbf{V}_{k,n} = \mathbf{a}_k \dot{q}_k \times \mathbf{L}_{k,n} = (\mathbf{a}_k \times \mathbf{L}_{k,n}) \dot{q}_k \\ \omega_{k,n} = \mathbf{a}_k \dot{q}_k \end{cases} \quad (3.16)$$

où :

$\mathbf{L}_{k,n}$  : désigne le vecteur d'origine  $O_k$  et d'extrémité  $O_n$

En introduisant le coefficient binaire  $\sigma_k$ , les vecteurs  $\mathbf{V}_{k,n}$  et  $\omega_{k,n}$  s'écrivent :

$$\begin{cases} \mathbf{V}_{k,n} = [\sigma_k \mathbf{a}_k + \bar{\sigma}_k (\mathbf{a}_k \times \mathbf{L}_{k,n})] \dot{q}_k \\ \omega_{k,n} = \bar{\sigma}_k \mathbf{a}_k \dot{q}_k \end{cases} \quad (3.17)$$

En appliquant le théorème de composition des vitesses, on peut sommer toutes les contributions élémentaires de chaque articulation afin d'obtenir les vecteurs finaux des vitesses de translation et de rotation  $\mathbf{V}_n$  et  $\omega_n$  du repère terminal par l'expression :

$$\begin{cases} \mathbf{V}_n = \sum_{k=1}^n \mathbf{V}_{k,n} = \sum_{k=1}^n [\sigma_k \mathbf{a}_k + \bar{\sigma}_k (\mathbf{a}_k \times \mathbf{L}_{k,n})] \dot{q}_k \\ \omega_n = \sum_{k=1}^n \omega_{k,n} = \sum_{k=1}^n \bar{\sigma}_k \mathbf{a}_k \dot{q}_k \end{cases} \quad (3.18)$$

Par identification avec la relation (3.15), la matrice Jacobienne exprimée dans le repère  $\mathbf{R}_n$ , notée  $\mathbf{J}_n$ , s'écrit :

$$\mathbf{J}_n = \begin{bmatrix} \sigma_1 \mathbf{a}_1 + \bar{\sigma}_1 (\mathbf{a}_1 \times \mathbf{L}_{1,n}) & \dots & \sigma_n \mathbf{a}_n + \bar{\sigma}_n (\mathbf{a}_n \times \mathbf{L}_{n,n}) \\ \sigma_1 \mathbf{a}_1 & \dots & \sigma_n \mathbf{a}_n \end{bmatrix} \quad (3.19)$$

### Calcul de la matrice ${}^i \mathbf{J}_n$

La matrice jacobienne peut être exprimée aussi bien dans le repère de base  $\mathbf{R}_0$  que dans le repère de l'organe terminal  $\mathbf{R}_n$ . Elle peut aussi être projetée dans un repère  $\mathbf{R}_i$  intermédiaire. La matrice jacobienne notée  ${}^i \mathbf{J}_n$  s'écrit :

$${}^i \mathbf{J}_n = \begin{bmatrix} \sigma_1 {}^i \mathbf{a}_1 + \bar{\sigma}_1 ({}^i \mathbf{a}_1 \times {}^i \mathbf{L}_{1,n}) & \dots & \sigma_n {}^i \mathbf{a}_n + \bar{\sigma}_n ({}^i \mathbf{a}_n \times {}^i \mathbf{L}_{n,n}) \\ \sigma_1 {}^i \mathbf{a}_1 & \dots & \sigma_n {}^i \mathbf{a}_n \end{bmatrix} \quad (3.20)$$

En remarquant que :  ${}^i \mathbf{a}_k \times {}^i \mathbf{L}_{k,n} = {}^i \mathbf{A}_k {}^k \hat{\mathbf{a}}_k {}^k \mathbf{L}_{k,n}$  avec :

- ${}^i \mathbf{A}_k$  : matrice d'orientation de dimension  $(3 \times 3)$  du repère  $\mathbf{R}_k$  dans le repère  $\mathbf{R}_i$ ,
- ${}^k \mathbf{a}_k = [0 \ 0 \ 1]^T$  et  ${}^k \mathbf{L}_{k,n} = {}^k \mathbf{P}_k = [{}^k P_{nx} \ {}^k P_{ny} \ {}^k P_{nz}]$

On calcule alors la  $k^{ieme}$  colonne de la matrice Jacobienne, notée  ${}^i \mathbf{J}_{n,k}$ , projetée dans le repère  $\mathbf{R}_i$  par la formule :

$${}^i \mathbf{J}_{n,k} = \begin{bmatrix} \sigma_1 {}^i \mathbf{a}_k + \bar{\sigma}_1 (-{}^k P_{ny} {}^i \mathbf{s}_k + {}^k P_{nx} {}^i \mathbf{n}_k) \\ \bar{\sigma}_1 {}^i \mathbf{a}_k \end{bmatrix} \quad i = 0, \dots, n ; k = 1, \dots, n \quad (3.21)$$

où :

- ${}^i \mathbf{s}_k, {}^i \mathbf{n}_k$  et  ${}^i \mathbf{a}_k$  : sont respectivement les vecteurs colonne de la matrice  ${}^i \mathbf{A}_k$ ,
- ${}^k P_{nx}$  et  ${}^k P_{ny}$  : sont respectivement la 1<sup>ere</sup> et 2<sup>ieme</sup> composantes du vecteur  ${}^k \mathbf{P}_n$  qui représente la quatrième colonne de  ${}^k \mathbf{T}_n$  calculée précédemment par le modèle géométrique direct.

### 3.5.2 Le MCD du système coopératif

Compte tenu de l'équation (3.21) et des résultats du modèle géométrique direct du système multi-robots coopératifs, le logiciel Maple nous a permis de déterminer les jacobiniennes  ${}^F\mathbf{J}_6$ ,  ${}^3\mathbf{J}_6$ ,  ${}^6\mathbf{J}_6$  dans les différents repères.

$${}^F\mathbf{J}_6 = \begin{bmatrix} -(L+r_3)C_2S_1\varphi_0 & -(L+r_3)S_2C_1\varphi_0 & C_2C_1\varphi_0 & 0 \\ -(L+r_3)C_2C_1\varphi_0 & (L+r_3)S_2S_1\varphi_0 & C_2S_1\varphi_0 & 0 \\ 0 & S_1\varphi_0 & C_2C_1\varphi_0 & S_2S_4C_1\varphi_0 + C_4S_1\varphi_0 \\ 0 & S_1\varphi_0 & 0 & C_2C_1\varphi_0 \\ 0 & -C_1\varphi_0 & 0 & C_2S_1\varphi_0 \\ 1 & 0 & 0 & S_2 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ aC_1\varphi_0 + bS_1\varphi_0 & 0 & 0 & 0 \\ S_2S_4C_1\varphi_0 + C_4S_1\varphi_0 & aC_1\varphi_0 + bS_1\varphi_0 & 0 & 0 \\ S_2S_4S_1\varphi_0 + C_4C_1\varphi_0 & aS_1\varphi_0 + bC_1\varphi_0 & 0 & 0 \\ -C_2S_4 & C_2C_2C_4S_5 + S_2C_5 & 0 & 0 \end{bmatrix}. \quad (3.22)$$

$${}^3\mathbf{J}_6 = \begin{bmatrix} 0 & (L+r_3) & 0 & 0 & 0 & 0 \\ -(L+r_3)C_2 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ C_2 & 0 & 0 & 0 & -S_4 & C_4S_5 \\ 0 & 1 & 0 & 0 & C_4 & S_4S_5 \\ S_2 & 0 & 0 & 1 & 0 & C_5 \end{bmatrix}. \quad (3.23)$$

$${}^6\mathbf{J}_6 = \begin{bmatrix} -(L+r_3)C_2(S_4C_5C_6 + C_4S_6) & (L+r_3)(C_4C_5C_6 - S_4S_6) & -S_5C_6 & 0 & 0 & 0 \\ (L+r_3)C_2(S_4C_5C_6 + C_4S_6) & (L+r_3)(C_4C_5C_6 + S_4C_6) & -S_5S_6 & 0 & 0 & 0 \\ -(L+r_3)C_2S_4S_5 & -(L+r_3)C_4S_5 & -C_5 & 0 & 0 & 0 \\ C_2(C_4C_5C_6 - S_4S_6) - S_2S_5C_6 & S_4C_5C_6 + C_4S_6 & 0 & {}^6J_6[1,3] & S_6 & 0 \\ -C_2(C_4C_5S_6 + S_4C_6) + S_2S_5S_6 & -S_4C_5S_6 + C_4C_6 & 0 & {}^6J_6[2,3] & C_6 & 0 \\ C_2C_4S_5 + S_2C_5 & S_4S_5 & 0 & {}^6J_6[3,3] & 0 & 1 \end{bmatrix}. \quad (3.24)$$

Avec :  $L = RL_4 + l_2$ ,  $a = -S_2C_4S_5 + C_2C_5$ , et  $b = S_4S_5$ .

**Remarque :** Lors de la modélisation du système multi-robots, on a considérer que les coordonnées opérationnelles sont les coordonnées de la position de l'outil dans le repère  $R_F$ ; alors il revient à calculer les vitesses  ${}^F V_E$  et  ${}^F \omega_E$ .

En appliquant la relation (3.13), on trouve que :

$${}^F J_E \dot{q} = \begin{bmatrix} {}^F V_E \\ {}^F \omega_E \end{bmatrix} \quad (3.25)$$

Avec :

$$\begin{bmatrix} {}^F V_E \\ {}^F \omega_E \end{bmatrix} = \begin{bmatrix} {}^F A_6 {}^6 V_E \\ {}^F A_6 {}^6 \omega_E \end{bmatrix} \quad (3.26)$$

ou :

$$\begin{bmatrix} {}^6 V_E \\ {}^6 \omega_E \end{bmatrix} = \begin{bmatrix} {}^6 V_6 + {}^6 \hat{\omega}_E {}^6 P_E \\ {}^6 \omega_6 \end{bmatrix} \quad (3.27)$$

et :

$${}^6 J_6 \dot{q} = \begin{bmatrix} {}^6 V_6 \\ {}^6 \omega_6 \end{bmatrix} \quad (3.28)$$

$$\begin{bmatrix} {}^F V_E \\ {}^F \omega_E \end{bmatrix} = \begin{bmatrix} {}^F A_6 & -{}^F A_6 {}^6 \hat{P}_E \\ O_3 & {}^F A_6 \end{bmatrix} \begin{bmatrix} {}^6 V_6 \\ {}^6 \omega_6 \end{bmatrix} \quad (3.29)$$

On peut conclure :

$${}^F J_E = \begin{bmatrix} {}^F A_6 & -{}^F A_6 {}^6 \hat{P}_E \\ O_3 & {}^F A_6 \end{bmatrix} {}^6 J_6 \quad (3.30)$$

### 3.6 Calcul du modèle cinématique inverse (MCI)

Le MCI d'un robot permet de calculer à partir d'une configuration  $q$  donnée les vitesses articulaires  $\dot{q}$  qui assurent au repère terminal une vitesse opérationnelle  $\dot{X}$  imposée, il est calculé par :

$$\dot{q} = J(q)^{-1} \dot{X} \quad (3.31)$$

Le calcul du modèle cinématique inverse revient à l'inversion de la matrice jacobienne du robot. Dans le cas régulier où la matrice Jacobienne est carré d'ordre  $n$  et son déterminant est non nul, son inversion est simple.

### 3.6.1 Analyse des singularités du système multi-robot

Les configurations singulières peuvent être déterminées par une analyse du rang de la matrice  $\mathbf{J}^{-1}$ . En général, pour une chaîne cinématique complexe, deux types de singularités [72][73] peuvent être distingués :

1. *Singularité du premier type*

Cette singularité se produit lorsque :

$$\det(\mathbf{J}) = 0 \quad \implies \quad \det(\mathbf{J}^{-1}) \longrightarrow \infty \quad (3.32)$$

Elle correspond à une limite du domaine atteignable. Si le noyau de  $\mathbf{J}$  est non vide, l'équation (3.32) implique qu'il existe un ensemble de  $\dot{\mathbf{q}}_i$  non tous nuls pour lequel  $\dot{\mathbf{X}}$  est nul.

2. *Singularité du second type*

Ce type de singularité se produit quand nous avons :

$$\det(\mathbf{J}^{-1}) = 0 \quad \implies \quad \det(\mathbf{J}) \longrightarrow \infty \quad (3.33)$$

Ceci correspond à un mouvement incommandable de l'objet manipulé lorsque toutes les liaisons actives sont bloquées ( $\dot{\mathbf{q}}_i = 0$ ).

### 3.6.2 Application au système multi-robot :

Afin de démontrer l'efficacité du générateur de trajectoire imposé, on considère ici la simulation pour la validation du cas de la coopération de deux robots saisissant le même objet dans l'espace opérationnel. Les simulations sont réalisées sous MATLAB-SIMULINK.

Nous considérons que la trajectoire désirée de l'objet dans l'espace opérationnelle est donnée comme suit :

- $\alpha_{obj} = \pi/2 + 0.5\sin(2t)$  : rotation du centre de l'objet autour de l'axe  $x_{obj}$  de l'objet.
- $\theta_{obj} = \pi/2 + 0.5\sin(2t)$  : rotation du centre de l'objet autour de l'axe  $z_{obj}$  de l'objet.
- $r_{obj} = r_1 + 5 * t/4$  : translation de l'objet sur l'axe  $z_{obj}$  de l'objet

Les résultats de la simulation sont représentés en figures ci-dessous.

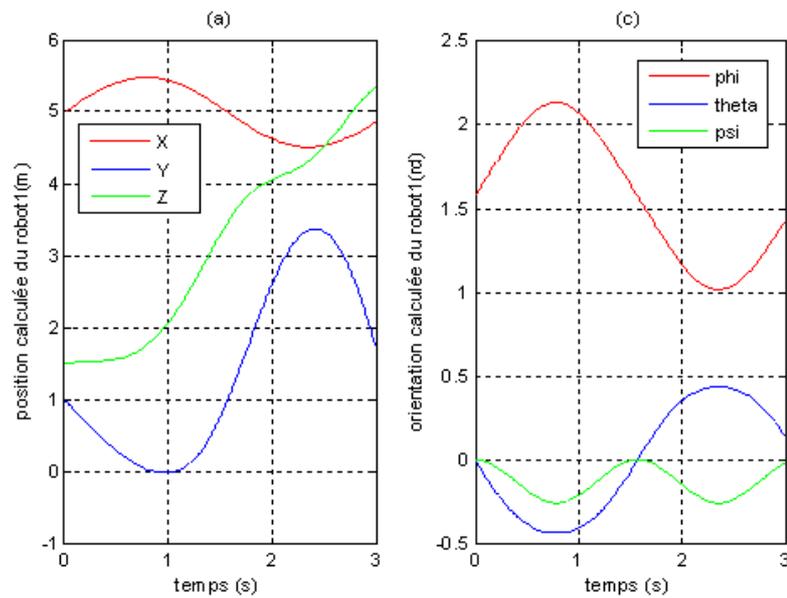


FIG. 3.17 – Position et orientation calculées du robot1

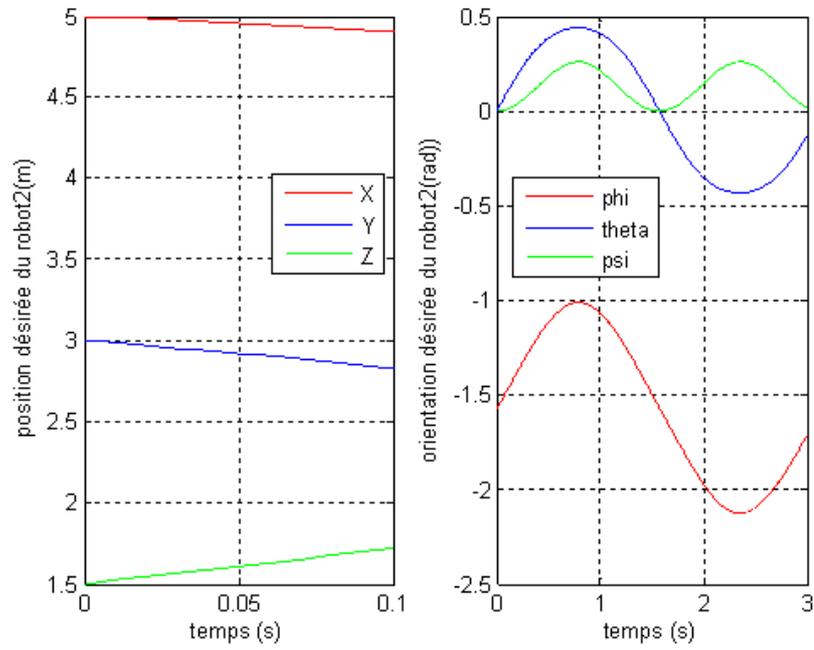


FIG. 3.18 – Position et orientation désirées du robot2

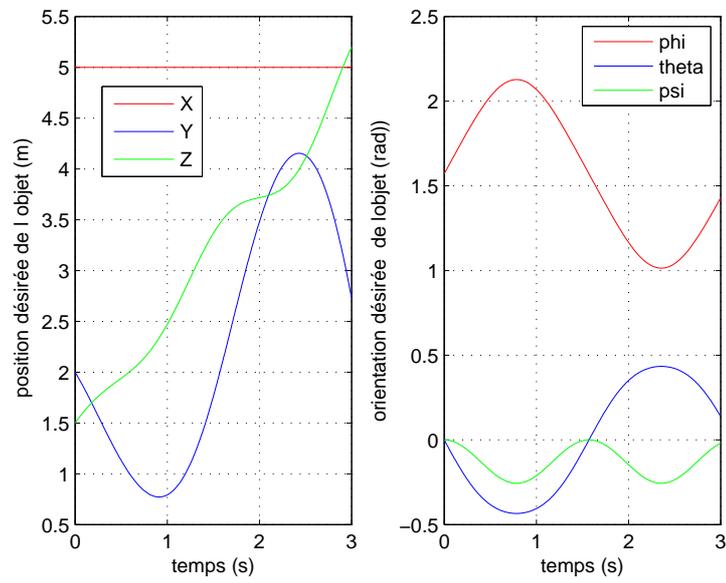


FIG. 3.19 – Position et orientation désirées de l'objet

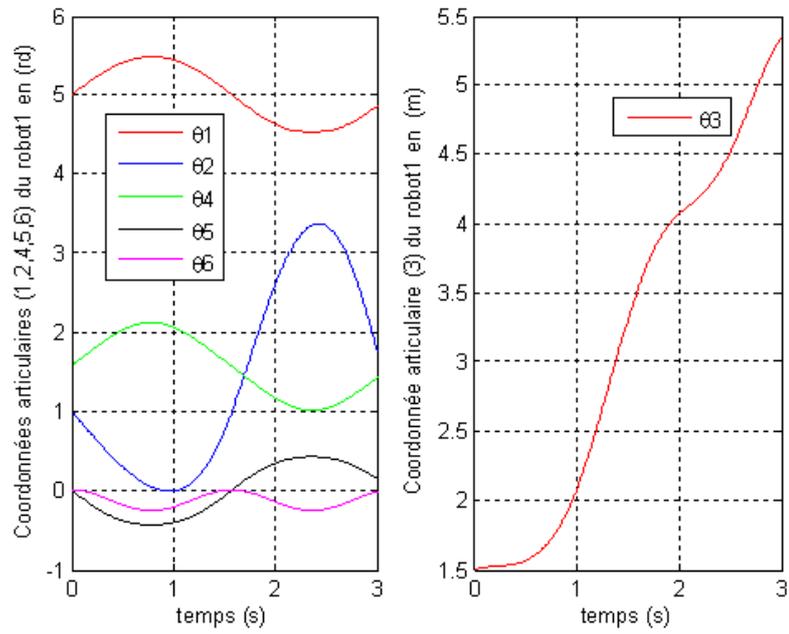


FIG. 3.20 – Coordonnées articulaires du robot1

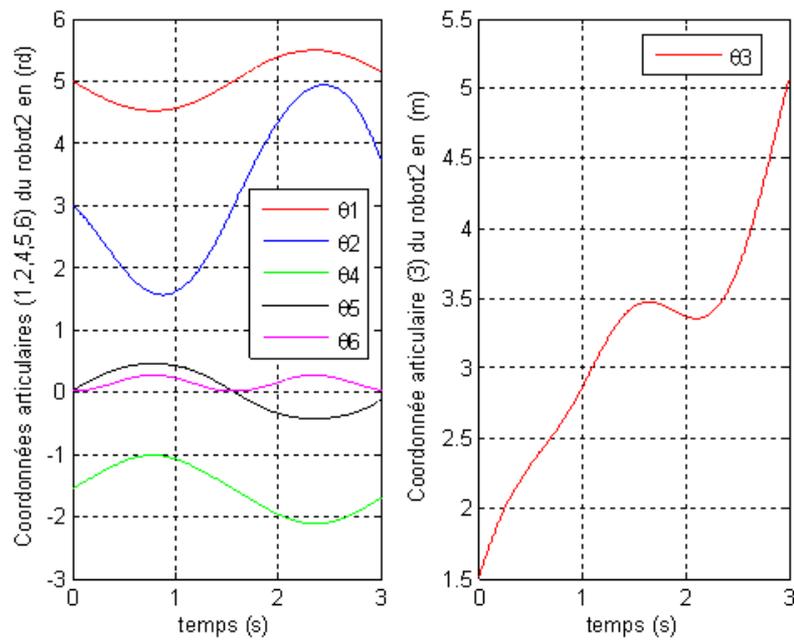


FIG. 3.21 – Coordonnées articulaires du robot2

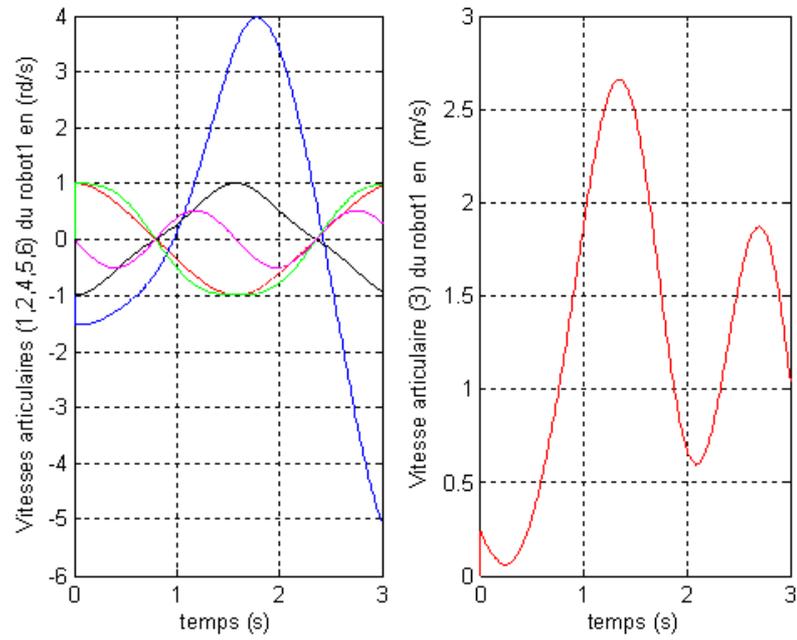


FIG. 3.22 – Profil des vitesses articulaires du robot1

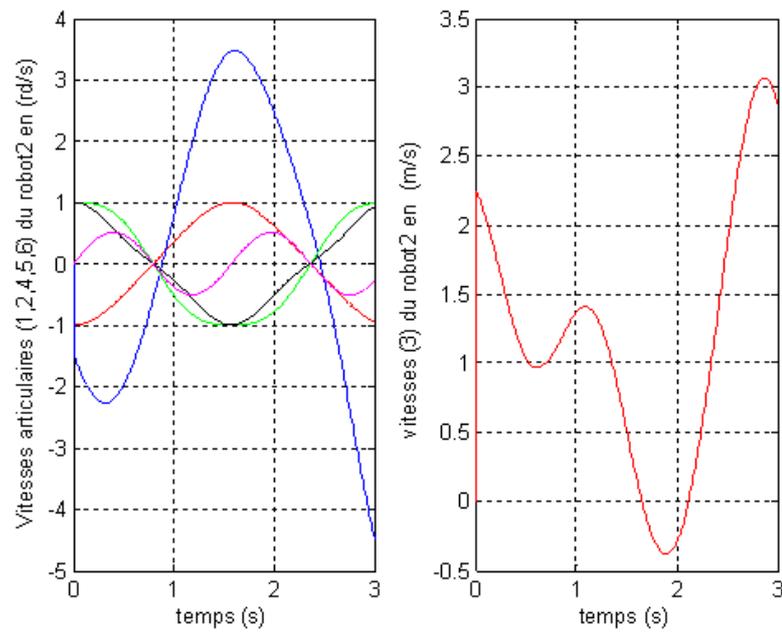


FIG. 3.23 – Profil des vitesses articulaires du robot2

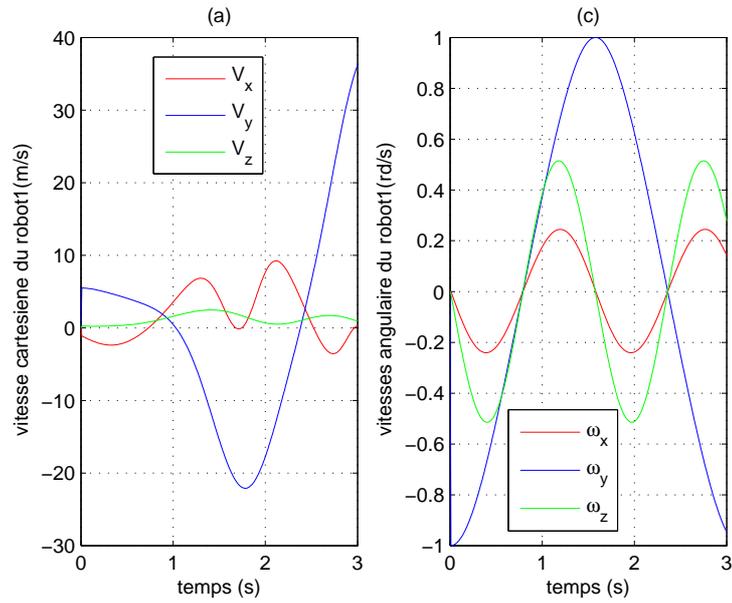


FIG. 3.24 – Vitesses cartésiennes et angulaires du point de contact du robot1

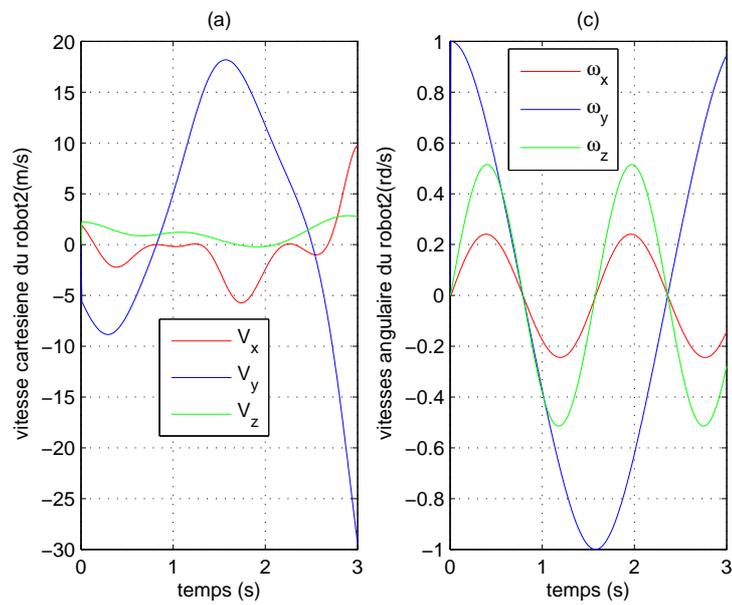


FIG. 3.25 – Vitesses cartésiennes et angulaires du point de contact du robot2

## 3.7 Conclusion

On a exposé dans ce chapitre les méthodes de calcul du modèle géométrique direct des robots à structure ouverte simple et arborescente.

Ces méthodes reposant sur l'utilisation des matrices de transformation homogène, leur application exige le respect des règles et conventions de repérage. Par ailleurs le modèle géométrique direct du robot "j" à l'aide de la méthode de Khalil-Kleinfinger [70] a été établi, ce qui a permis de trouver le modèle géométrique inverse (MGI), en utilisant la méthode de Paul.

Par la suite on a établi le modèle cinématique direct d'un robot en calculant sa matrice jacobienne de base ou celle projetée dans un repère intermédiaire quelconque. La détermination de l'inverse de la jacobienne permet le calcul du modèle cinématique inverse.

Par contre l'objectif principal de ce chapitre était de mettre en oeuvre un générateur de trajectoire assez général. l'objet manipulé par l'ensemble des robots peut se déplacer suivant une loi horaire quelconque, mais sous la contrainte du domaine atteignable des robots porteurs. Le générateur de trajectoire ainsi élaboré permet de récupérer à tout instant les valeurs de coordonnées, des vitesses, des accélérations articulaires, toute la cinématique du centre de gravité de l'objet ainsi que les coordonnées des points de contacts avec l'objet.

Les données fournies par le générateur, serviront comme base pour le calcul dynamique du système (robots-objet), à savoir :

1. Détermination des valeurs des couples articulaires.
2. L'équilibre dynamique de l'objet manipulé.
3. La distribution optimale d'effort pour les points de contacts (robots-objet)

# Chapitre 4

## Calcul et optimisation de la distribution d'efforts sur l'objet

### 4.1 Distribution optimale des forces

#### 4.1.1 Introduction

La stratégie de commande est une stratégie hybride qui prend en compte la distribution des efforts sur les deux robots, cela rend le calcul de cette distribution très important. Le système robot-objet est un système redondant. Cette caractéristique vient du fait qu'il a six articulations par robot, ce qui présente plus d'articulations à contrôler que le nombre de degrés de liberté de l'objet. Donc, mathématiquement, il existe moins d'équations d'équilibre que de variables inconnues, alors la solution de ces équations n'est pas unique. En plus, le robot a des contraintes physiques qui ne peuvent être représentées mathématiquement que par des inéquations modélisant le contact robot-objet, à savoir : la force normale doit être perpendiculaire à la surface de contact, l'objet ne doit pas glisser et le couple développé par le moteur de chaque articulation ne dépasse pas la valeur maximale admise. Evidemment, le calcul de la distribution des forces sur les deux robots nécessite, premièrement l'étude du modèle géométrique directe et inverse (voir chapitre 3), deuxièmement une optimisation des forces agissant sur les robots, considérant les équations d'équilibre et les inéquations des contraintes physiques. De nombreux travaux de recherche ont étudié ce problème et ils ont développé différents algorithmes pour l'optimisation de la solution. Plusieurs approches ont été proposées dans la littérature pour résoudre ce type de problème, selon la classe du problème en programmation mathématique, comme la linéarité

ou non du problème considéré. Parmi les approches on peut citer :

1. Les approches basées sur la programmation linéaire primales ou duales [9], [74],[75].
2. L'approche basée sur la programmation quadratique [32], [76]

La formulation et la résolution du problème de distribution des efforts de contact en programmation mathématique nécessite la prise en compte des contraintes suivantes :

- Eviter le glissement de l'objet manipulé lorsque il est tenu par le système multi-robots coopératif;
- Assurer un contact permanent lors de la saisie de l'objet par les effecteurs terminaux des différents robots.
- Faire varier graduellement les forces de contact lorsque les robots entre en contact avec l'objet manipulé.

L'approche adoptée ici pour la résolution du problème de distribution des efforts de contact est basée sur l'approche proposée dans [77],[78] et [79] qui est elle même une extention des méthodes basées sur la programmation quadratique, par réduction de la complexité algorithmique.

### 4.1.2 Formulation du problème

Considerons un objet de masse  $m_0$  susceptible d'être manipulé par les deux robots coopératifs. Le système de forces agissant sur l'objet à saisir est représenté en figure(4.1). Pour des raisons de simplicité on représente seulement les composantes de la force au niveau du point de contact.

Dans le cas général les moments aux points de contact sont négligés. Soient  $(x_0, y_0, z_0)$  le repère fixé au centre de gravité de l'objet manipulé, et  $(x_{6,j}, y_{6,j}, z_{6,j})$  le système de coordonnées fixé au point de contact du robot  $j$ .  $(x_{6,j}, z_{6,j})$  reste parallèle à  $(x_0, y_0, z_0)$  et l'axe  $z$  est normal à la surface de contact supposée parallèle au plan  $(x_0, y_0)$ .  $\mathbf{F} = [F_X F_Y F_Z]^T$  et  $\mathbf{M} = [M_X M_Y M_Z]^T$  sont respectivement le vecteur force et le vecteur moment agissant sur le corps du robot, résultant de la gravité et des forces extérieures agissant sur le corps. Soient  $f_{x,j}$ ,  $f_{y,j}$ , et  $f_{z,j}$  les forces qui agissent sur le robot "j" selon les directions de  $x_0$ ,  $y_0$  et  $z_0$  respectivement. Le nombre de robots coopératifs  $n$ , peut varier entre 2 et 3 dans cette étude.

Les équations d'équilibre, forces et moments, du robot s'écrivent ainsi :

$$\begin{cases} \sum_{j=1}^n \mathbf{f}_j = \mathbf{F} \\ \sum_{j=1}^n \mathbf{OP}_j \wedge \mathbf{f}_j = \mathbf{M} \end{cases} \quad (4.1)$$

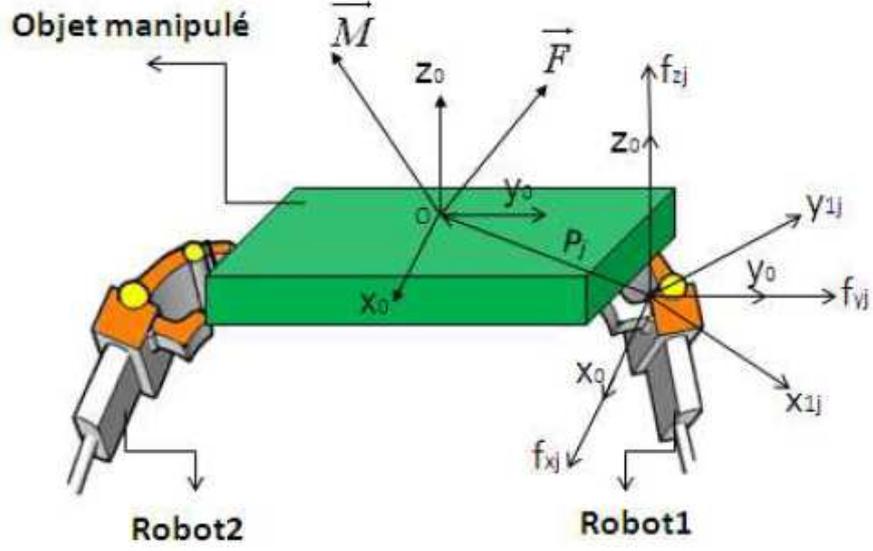


FIG. 4.1 – Vue des deux robots et choix des repères

avec  $OP_j$  le vecteur représentant la position liant le point de contact du robot "j" et le centre de gravité de l'objet manipulé. La forme générale de ce système d'équations s'écrit comme suit :

$$A G = W \quad (4.2)$$

avec :

$$\begin{cases} \mathbf{G} = [f_1^T f_2^T \dots f_n^T]^T & \in \mathbb{R}^{3n} \\ \mathbf{f}_j^T = [f_{x,j} f_{y,j} f_{z,j}]^T & \in \mathbb{R}^3 \\ \mathbf{W} = [\mathbf{F}^T \mathbf{M}^T]^T & \in \mathbb{R}^6 \end{cases} \quad \mathbf{A} = \begin{pmatrix} \mathbf{I}_3 & \dots & \dots & \mathbf{I}_3 \\ \mathbf{B}_1 & \dots & \dots & \mathbf{B}_n \end{pmatrix} \in \mathbb{R}^{6 \times 3n}$$

$$\mathbf{B}_j \equiv \widehat{OP}_j \equiv \begin{pmatrix} 0 & -P_{z,j} & P_{y,j} \\ P_{z,j} & 0 & -P_{x,j} \\ -P_{y,j} & P_{x,j} & 0 \end{pmatrix} \in \mathbb{R}^{3 \times 3}$$

Où  $\mathbf{I}_3$  est la matrice identité et  $\mathbf{G}$  est le vecteur force agissant sur le robot, correspondant à trois ( $\mathbf{G} \in \mathbb{R}^9$ ).  $\mathbf{A}$  est la matrice coefficient qui s'écrit en fonction des différents robots, et  $\mathbf{B}_j$  est la matrice antisymétrique formée par  $(P_{x,j}, P_{y,j}, P_{z,j})$ , représentant les coordonnées des positions du robot "j" dans  $(x_0, y_0, z_0)$ .  $\mathbf{W}$  est le vecteur force/moment résultant sur le corps. Il est clair que l'équation (4.2) est un système indéterminé et sa solution n'est pas unique. Il existe ainsi une infinité de solutions vérifiant le système d'équations. De ce fait, les forces sur

les robots doivent vérifier les contraintes physiques suivantes, sinon le système ne sera pas valide.

1. L'objet manipulé ne doit en aucun cas glisser lorsque il est manipulé par l'ensemble des robots, d'où les contraintes de non glissement suivantes :

$$\sqrt{f_{x,j}^2 + f_{y,j}^2} \leq \mu f_{z,j} \quad (4.3)$$

avec  $\mu$  est le coefficient de frottement sur la surface de contact

2. Du fait que les forces appliquées sur les robots coopératifs sont dues aux couples articulaires correspondants, qui sont bornés , Il en résulte :

$$-\tau_{jmax} \leq \mathbf{J}_j^{Tj} \mathbf{A}_{0j} \begin{pmatrix} f_{x,j} \\ f_{y,j} \\ f_{z,j} \end{pmatrix} \leq \tau_{jmax} \quad (4.4)$$

pour ( $j = 1 \dots n$ ), où  $\mathbf{J}_j \in \mathfrak{R}^{3 \times 6}$  est la Jacobienne du robot "j",  $\tau_{jmax} \in \mathfrak{R}^{6 \times 1}$  est le vecteur des couples articulaires maximaux du robot "j", et  $\mathbf{A}_{0j} \in \mathfrak{R}^{3 \times 3}$  est la matrice d'orientation de  $(x_{1,j}, y_{1,j}, z_{1,j})$  par rapport à  $(x_0, y_0, z_0)$ .

3. Pour que l'objet soit bien tenu par les organes terminaux des robots coopératif et que le contact robots-objet soit permanent sur la surface de contact, il faut s'assurer que la composante  $f_{z,j}$  vérifie la condition suivante :

$$f_{z,j} \geq 0 \quad (4.5)$$

Le problème d'optimisation non linéaire, résultant de la formulation du problème de calcul de la distribution des efforts de contact est un problème difficile, qui est dans certaines situation NP-complet. Aussi et pour des besoins algorithmiques, complexité espace-temps, une forme linéarisée du problème dans le cas de la coopération multi-robots est proposée.

## 4.2 Réduction et solution du Problème

L'équation(4.3) est la formulation du cône de friction voir figure(4.2). Afin d'éviter la non linéarité de cette équation, la majorité des recherches remplace le cône de friction par une

pyramide qui lui est inscrite [74] [80] [32] . Alors, la non linéarité de la contrainte de frottement est approximativement exprimée par les inéquations linéaires suivantes :

$$\begin{cases} f_{x,j} \leq \mu' f_{z,j} \\ f_{y,j} \leq \mu' f_{z,j} \\ j = 1, \dots, n \end{cases} \quad (4.6)$$

Où  $\mu' = \frac{\sqrt{2}\mu}{2}$  pour la pyramide de friction. Alors, le problème initial de contrainte non linéaire se

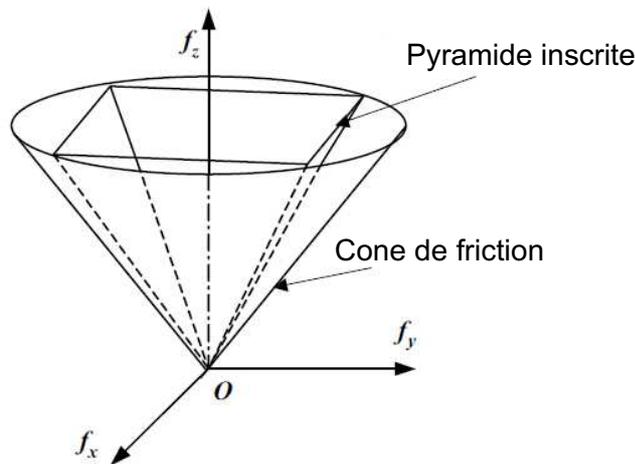


FIG. 4.2 – Pyramide de friction inscrite dans le cône

transforme en un problème linéaire en remplaçant l'équation (4.3) par l'équation linéaire (4.6) [74],[75], et [32]. Le phénomène de glissement sur la surface de contact lorsque l'objet est tenu par les effecteurs terminaux des robots coopératifs, peut être minimisé, en optimisant le rapport entre les composantes tangentielle et normales des forces agissant sur les robots. Liu et Wen [81] ont trouvé une relation qui lie les forces sur les robots et qui permet la transformation d'une partie du problème de friction du cas non linéaire au linéaire, en définissant le rapport global comme étant le rapport entre la force normale et tangentielle agissant sur l'objet manipulé. L'avantage de cette méthode est qu'une partie des forces va satisfaire le même rapport global des forces ; le reste des composantes doivent satisfaire les inéquations de contraintes, données par (4.6) . Par exemple, si on définit  $f_{x,j}$  comme étant les forces du premier type et  $f_{y,j}$  ( $j = 1, ..n$ ), celles du deuxième type, alors nous obtenons :

$$f_{x,j} = k_{xz} f_{z,j} \quad , (i = 1, \dots, n) \quad (4.7)$$

$$f_{y,j} \leq \mu^* f_{z,j} \quad , (i = 1, \dots, n) \quad (4.8)$$

ave  $k_{xz} = \frac{F_X}{F_Z}$  le rapport global des forces appliquées sur l'objet manipulé selon la direction  $x_0$  et  $z_0$ .  $\mu^*$  est le coefficient de frottement de cette nouvelle contrainte. D'après l'équation (4.3), on a  $\mu^* = \sqrt{\mu^2 - k_{xz}^2}$ . Finalement, le problème de distribution de forces devient linéaire par le remplacement de l'équation (4.3) par les équations (4.7) et (4.8).

### 4.2.1 Etude de la continuité de la solution du problème

Lors de la modélisation du système multi-robots coopératifs, on considère trois robots manipulant simultanément un objet rigide, alors  $\mathbf{G}$  et  $\mathbf{A}$  deviennent respectivement un vecteur de dimension  $9 \times 1$  et une matrice de  $6 \times 9$ . Le système d'équations (4.2) contient 9 variables inconnues avec six équations. En ajoutant les équations (4.7) aux équations (4.2), on obtient un système de 9 équations.

$$\overline{\mathbf{A}} \mathbf{G} = \overline{\mathbf{W}} \quad (4.9)$$

avec :

$$\overline{\mathbf{A}} = \begin{pmatrix} \mathbf{I}_3 & \mathbf{I}_3 & \mathbf{I}_3 \\ \mathbf{B}_1 & \mathbf{B}_2 & \mathbf{B}_3 \\ 1 & 0 & -k_{xz} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -k_{xz} & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -k_{xz} \end{pmatrix} \quad \mathbf{G} = \begin{pmatrix} f_1 \\ f_2 \\ f_3 \end{pmatrix} \quad \overline{\mathbf{W}} = \begin{pmatrix} \mathbf{F} \\ \mathbf{M} \\ 0 \\ 0 \\ 0 \end{pmatrix}$$

En combinant les différentes lignes de la matrice  $\overline{\mathbf{A}}$ , l'équation(4.9) s'écrit :

$$\widehat{\mathbf{A}} \mathbf{G} = \widehat{\mathbf{W}} \quad (4.10)$$

avec :

$$\widehat{\mathbf{A}} = \begin{pmatrix} 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 0 & 1 & 0 \\ 1 & 0 & -k_{xz} & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & -k_{xz} & 0 & 0 & 0 \\ -P_{y,1} & P_{x,1} & 0 & -P_{y,2} & P_{x,2} & 0 & -P_{y,3} & P_{x,3} & 0 \\ P_{z,1} & 0 & -P_{x,1} & P_{z,2} & 0 & -P_{x,2} & P_{z,3} & 0 & -P_{x,3} \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & -k_{xz} \\ 0 & -P_{z,1} & P_{y,1} & 0 & -P_{z,2} & P_{y,2} & 0 & -P_{z,3} & P_{y,3} \end{pmatrix} \quad \widehat{\mathbf{W}} = \begin{pmatrix} F_X \\ F_Y \\ 0 \\ 0 \\ M_Z \\ M_Y \\ 0 \\ M_X \end{pmatrix}$$

Où  $\widehat{\mathbf{A}} \in \mathfrak{R}^{8 \times 9}$  est la matrice résultante de  $\overline{\mathbf{A}}$  après combinaison.  $\mathbf{G} \in \mathfrak{R}^8$  est le vecteur force appliqué au robot.  $\widehat{\mathbf{W}} \in \mathfrak{R}^8$  est le vecteur résultant de  $\mathbf{W}$  après combinaison. Le problème de distribution de force est soumis aux inéquations de contraintes exprimées par (4.4), (4.5) (4.9).

$\mathbf{G}$  est un vecteur de dimension 18 , et  $\mathbf{A}$  est une matrice de dimension  $6 \times 18$ .  $\in \mathfrak{R}^{1 \times 18}$  sera réduit à 9 inconnues.

## 4.2.2 Solution optimale du problème

La solution des équations dynamiques du robot n'est pas unique, mais elle peut être choisie d'une façon optimale, en introduisant une fonction objectif. L'approche utilisée ici est de minimiser la somme pondérée des couples articulaires du robot, correspondant à la fonction objectif suivante [76],[35] :

$$f_{\mathbf{G}} = \mathbf{p}^T \mathbf{G} + \frac{\mathbf{G}^T \mathbf{Q} \mathbf{G}}{2} \quad (4.11)$$

avec :

$$\mathbf{p}^T = [\widehat{\tau}_1^T \mathbf{J}_1^T \dots, \widehat{\tau}_n^T \mathbf{J}_n^T] \in \mathfrak{R}^{3n}$$

$$\mathbf{Q} = \begin{pmatrix} \mathbf{J}_1 \mathbf{q}_1 \mathbf{J}_1^T & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \mathbf{J}_n \mathbf{q}_n \mathbf{J}_n^T \end{pmatrix} \in \mathfrak{R}^{3n \times 3n}$$

avec  $\widehat{\tau}_j$  le couple articulaire dû au poids et l'inertie du robot "j",  $\mathbf{J}_j$  la jacobienne du robot "j", et  $\mathbf{q}_j$  une matrice de pondération définie diagonale du robot "j". Cette fonction objectif est strictement convexe. Le temps d'obtention de la solution ne dépendent pas des conditions

initiales, la méthode "QP" est meilleure que la méthode "LP" en vitesse et qualité de la solution obtenue [76]. Le problème général de la "LQP" pour la distribution des forces sur le robot s'écrit comme suit :

$$f_G = p^T G + \frac{G^T Q G}{2} \quad (4.12)$$

$$\widehat{A} G = \widehat{W} \quad (4.13)$$

$$B G \leq C \quad (4.14)$$

Où  $G \in \mathfrak{R}^9$  désigne le vecteur forces inconnues. On rappelle que l'équation (4.13) représente l'équation(4.9), et l'équation(4.14) est la résultante des inéquations de contrainte par la combinaison de l'équation(4.4), l'équation (4.5) et (4.9), et avec :

$$B = [B_1^T B_2^T B_3^T B_4^T]^T \in \mathfrak{R}^{36 \times 9}$$

$$C = [\tau_{1max}, \tau_{2max}, \tau_{3max} \dots \tau_{6max} - \tau_{1max} - \tau_{2max} - \tau_{3max} \mathbf{000000}]^T \in \mathfrak{R}^{36}$$

avec :

$$B_1 = \begin{bmatrix} J_1^T R_1 & 0 & 0 \\ 0 & J_2^T R_2 & 0 \\ 0 & 0 & J_3^T R_3 \end{bmatrix} \in \mathfrak{R}^{18 \times 9}$$

$$B_2 = \begin{bmatrix} -J_1^T R_1 & 0 & 0 \\ 0 & -J_2^T R_2 & 0 \\ 0 & 0 & -J_3^T R_3 \end{bmatrix} \in \mathfrak{R}^{18 \times 9}$$

$$B_3 = \begin{bmatrix} 0 & 0 & -1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \end{bmatrix} \in \mathfrak{R}^{3 \times 9}$$

$$B_4 = \begin{bmatrix} 0 & 1 & -\mu^* & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & -\mu^* & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & -\mu^* \end{bmatrix} \in \mathfrak{R}^{3 \times 9}$$

Comme on a huit équations linéairement indépendantes, l'équation (4.14) sera transformée de façon à avoir une matrice unitaire dans les 8 premières colonnes  $I_8 \in \mathfrak{R}^{8 \times 8}$ [80], le résultat s'écrit alors :

$$[I_8 \quad \widehat{A}_r] \begin{bmatrix} \widehat{G}_b \\ \widehat{G}_r \end{bmatrix} = \widehat{W}_r \quad (4.15)$$

Où  $\widehat{\mathbf{A}}_r \in \mathfrak{R}^8$  est la nouvelle colonne après transformation de la matrice  $\widehat{\mathbf{A}}$  après transformation.  $\widehat{\mathbf{G}}_b \in \mathfrak{R}^8$  est le vecteur partiel de  $\mathbf{G}$ .  $\widehat{\mathbf{G}}_r \in \mathfrak{R}$  correspondant à la partie inconnu de  $\widehat{\mathbf{G}}$ .  $\widehat{\mathbf{W}}_r \in \mathfrak{R}^8$  est le vecteur résultant de  $\widehat{\mathbf{W}}$  après transformation. L'équation (4.15) sera écrite sous la forme suivante :

$$\mathbf{I}_8 \widehat{\mathbf{G}}_b + \widehat{\mathbf{A}}_r \widehat{\mathbf{G}}_r - \widehat{\mathbf{W}}_r = 0 \quad (4.16)$$

avec

$$\widehat{\mathbf{G}}_b = \widehat{\mathbf{W}}_r - \widehat{\mathbf{A}}_r \widehat{\mathbf{G}}_r. \quad (4.17)$$

Finalement, il en résulte

$$\mathbf{G} = \begin{bmatrix} \widehat{\mathbf{G}}_b \\ \widehat{\mathbf{G}}_r \end{bmatrix} = \begin{bmatrix} \widehat{\mathbf{W}}_r \\ 0 \end{bmatrix} + \begin{bmatrix} -\widehat{\mathbf{A}}_r \\ 1 \end{bmatrix} \widehat{\mathbf{G}}_r \quad (4.18)$$

soit  $\widehat{\mathbf{G}}_0 = [-\widehat{\mathbf{W}}_r^T \ 0]^T \in \mathfrak{R}^8$  et  $\mathbf{N} = [-\widehat{\mathbf{A}}_r^T \ 1]^T \in \mathfrak{R}^9$ , alors l'Eq(4.18) sera

$$\mathbf{G} = \widehat{\mathbf{G}}_0 + \mathbf{N} \widehat{\mathbf{G}}_r. \quad (4.19)$$

Substituant l'Eq(4.19) dans l'Eq (4.12) et (4.14), le "LQP" s'exprime ainsi :

$$\text{minimiser } f(\widehat{\mathbf{G}}_r), \quad (4.20)$$

$$\text{sous la contrainte } \mathbf{B} \mathbf{N} \widehat{\mathbf{G}}_r \leq \mathbf{C} - \mathbf{B} \widehat{\mathbf{G}}_0. \quad (4.21)$$

avec :

$$\begin{aligned} f(\widehat{\mathbf{G}}_r) &= \mathbf{p}^T \widehat{\mathbf{G}}_0 + \frac{1}{2} \widehat{\mathbf{G}}_0^T \mathbf{Q} \widehat{\mathbf{G}}_0 + \mathbf{p}^T \widehat{\mathbf{G}}_0 \\ &+ \frac{1}{2} \widehat{\mathbf{G}}_0^T \mathbf{Q} \mathbf{N} \widehat{\mathbf{G}}_r + \frac{1}{2} \widehat{\mathbf{G}}_r^T \mathbf{N}^T \mathbf{Q} \widehat{\mathbf{G}}_0 \\ &+ \frac{1}{2} \widehat{\mathbf{G}}_r^T \mathbf{N}^T \mathbf{Q} \mathbf{N} \widehat{\mathbf{G}}_r \end{aligned}$$

comme  $\widehat{\mathbf{G}}_r$  est une seule variable qu'on note  $x$ , la distribution optimale des forces peut être écrite comme suit :

$$\text{minimiser } a_0 x^2 + a_1 x + a_2 \quad \text{sous la contrainte } x \in [b_1 \ b_2] \quad (4.22)$$

avec

$$\begin{aligned} a_0 &= \mathbf{N}^T \mathbf{Q} \mathbf{N} \\ a_1 &= \mathbf{p}^T \mathbf{N} + \frac{1}{2} \widehat{\mathbf{G}}_0^T \mathbf{Q} \mathbf{N} + \frac{1}{2} \mathbf{p}^T \mathbf{Q} \widehat{\mathbf{G}}_0 \\ a_2 &= \mathbf{p}^T \widehat{\mathbf{G}}_0 + \frac{1}{2} \widehat{\mathbf{G}}_0^T \mathbf{Q} \widehat{\mathbf{G}}_0 \end{aligned}$$

où  $[b_1 \ b_2]$  sont les bornes résultant de l'équation(4.21). D'ou il est claire que  $a_0 \succ 0$  car la matrice  $\mathbf{Q}$  est définie positive, alors la solution obtenue est bien optimale.

### 4.2.3 Simulations

Le système multi-robots coopératifs à simuler ainsi que sa configuration géométrique et son paramétrage sont représentés en figure (4.3) et (4.4), Le système présente six articulations

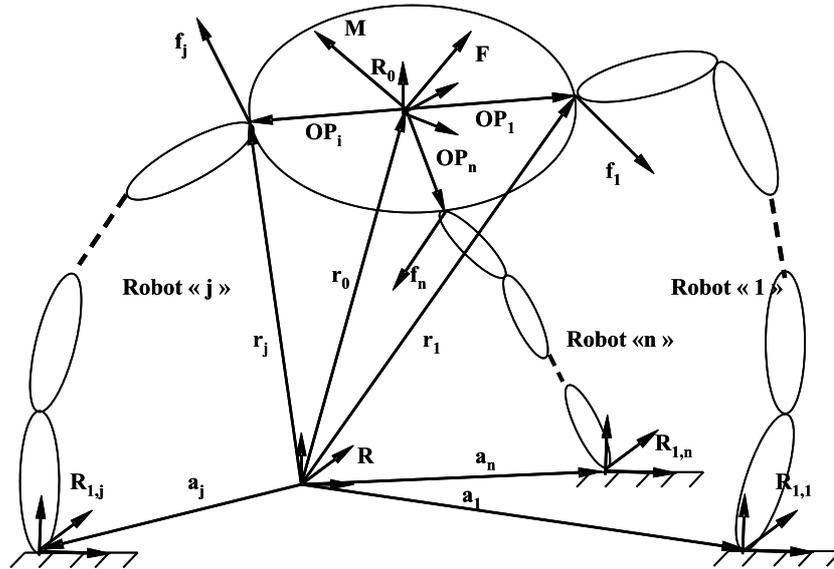


FIG. 4.3 – Paramètres géométriques du système multi-robots coopératifs

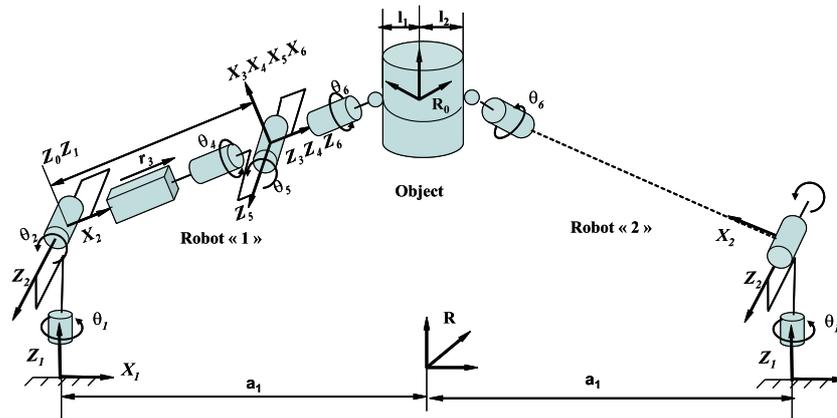


FIG. 4.4 – Exemple d'un système multi-robots coopératifs

actionnées :

$(\theta_{1,j}, \theta_{2,j}, \dots, \theta_{6,j})$  sur le robot "j", dont les couples articulaires sont notés  $(\tau_{1,j}, \tau_{2,j}, \dots, \tau_{6,j})$ , pour  $(j = 1, 2, 3)$ , respectivement. La Jacobienne du robot "j" peut être exprimée par l'équation

suivante :

$${}^0J_6^T = \begin{bmatrix} (L+r_3)C_{2,j}S_{1,j} & -(L+r_3)C_{2,j}C_{1,j} & 0 \\ -(L+r_3)S_{2,j}C_{1,j} & (L+r_3)S_{2,j}S_{1,j} & S_{1,j} \\ C_{2,j}C_{1,j} & C_{2,j}S_{1,j} & C_{2,j}C_{1,j} \\ 0 & 0 & S_{2,j}S_{4,j}C_{1,j} + C_{4,j}S_{1,j} \\ 0 & 0 & aC_{1,j} + bS_{1,j} \\ 0 & 0 & 0 \end{bmatrix} \quad (4.23)$$

la matrice d'orientaion de  $(x_{6,j}, y_{6,j}, z_{6,j})$  par rapport au repère  $(x_0, y_0, z_0)$  peut s'ecrire comme suit :

$$\mathbf{A}_{0j} = \begin{pmatrix} \cos \phi_{1,j} & \sin \phi_j & 0 \\ -\sin \phi_j & \cos \phi_j & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (4.24)$$

#### 4.2.4 Résultats de la simulation

Afin de montrer l'efficacité de l'approche proposée, des simulations ont été réalisées sous Matlab. Pour cela on considère que l'objet manipulé fait un mouvement de rotation et de translation combiné autour de l'axe  $Z_0$ . En outre, le tenseur de force agissant au centre du corps est donnée par :

$$F_x = -5, F_y = 10, F_z = -250 \text{ [N]}, M_x = 3, M_y = 1, M_z = 2 \text{ [Nm]}, \tau_j = [10 \ 10 \ 10 \ 10 \ 10 \ 10]^T$$

designe le vecteur couple ariculaire maximum et  $\mu = 0.05$  le coefficient de frottement statique

Pour la fonction objectif équation (4.11), la matrice de pondération est choisies comme suit :

$$\mathbf{p} = \mathbf{0} \text{ et } \mathbf{Q} = \mathbf{I} \text{ (la matrice identité).}$$

La figure (4.5) montre la trajectoire de l'objet manipulé dans le système d'axe X-Y-Z supporté par les trois robots.

Les figures (4.6), (4.7) et (4.8) illustrent les forces appliquées respectivement sur le robot1, le robot2 et le robot3. On peut voir que cette distribution des efforts valide bien les equations d'équilibre :

$$\sum f_{x_j} = F_x, \quad \sum f_{y_j} = F_y, \quad \sum f_{z_j} = F_z$$

Par ailleurs , la composante de la force  $f_{z_j}$  sur l'axe  $Z_0$  n'est jamais négative. Ce qui confirme que la condition de contact permanent de l'objet manipulé est bien respectée. On peut aussi voir en figure (4.9) que la contrainte donnée par la relation (4.3) est satisfaite. Ainsi pour tout

le robot(j),  $j = 1, 2, 3$  ou la courbe  $fx_j^2 + fy_j^2$  est toujours localisée en dessous de la courbe  $\mu^2 fz_j^2$ .

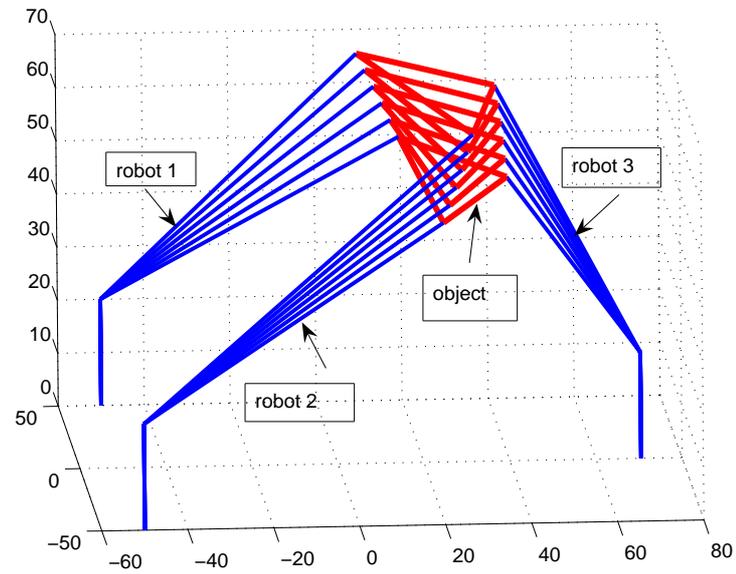


FIG. 4.5 – Vue de l'objet saisi par les trois robots

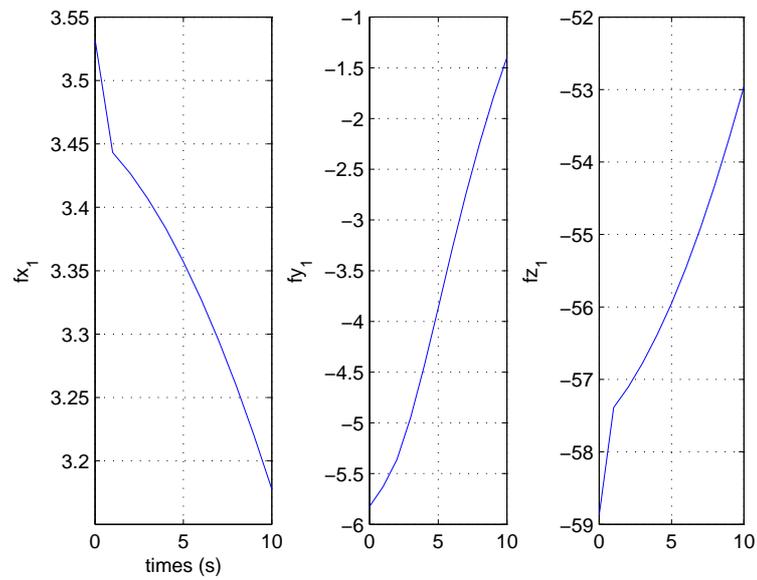


FIG. 4.6 – Forces sur le premier robot

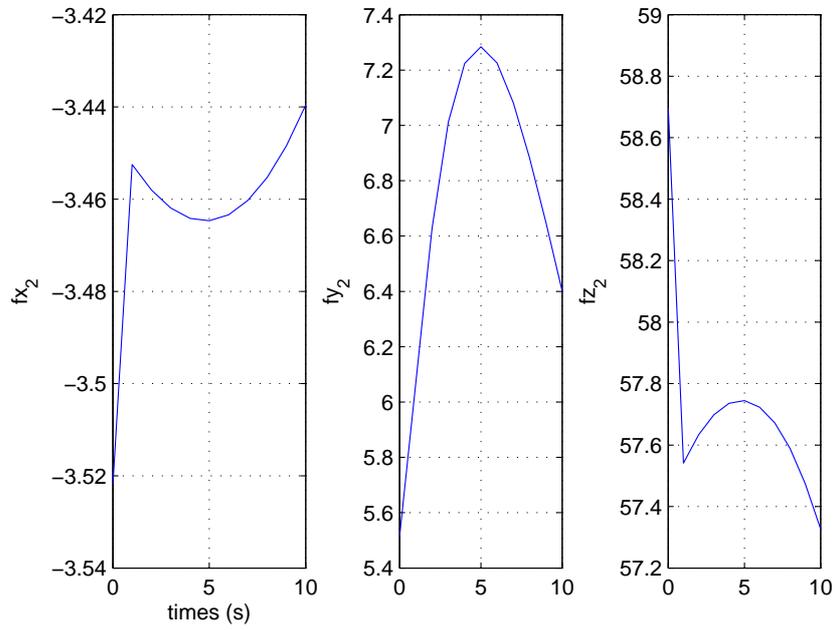


FIG. 4.7 – Forces sur le deuxième robot

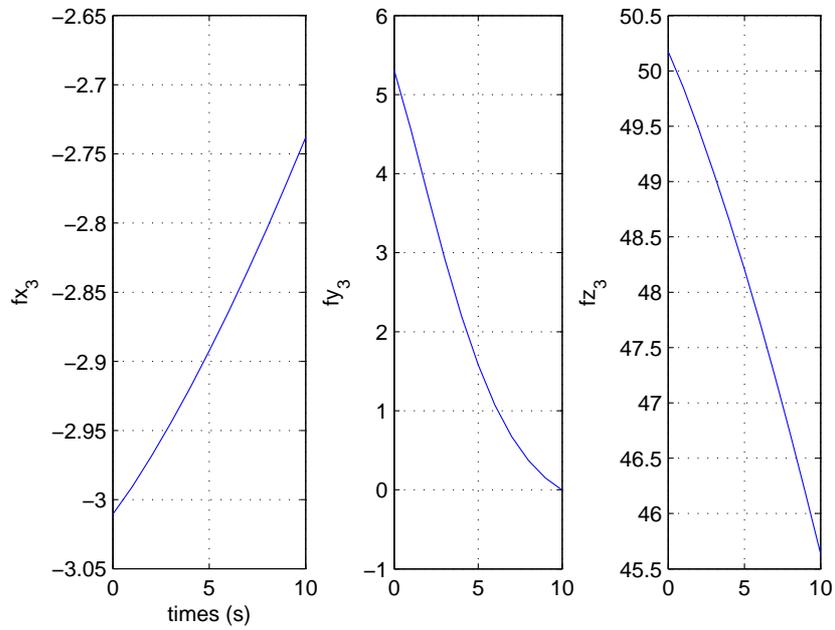


FIG. 4.8 – Forces sur le troisième robot

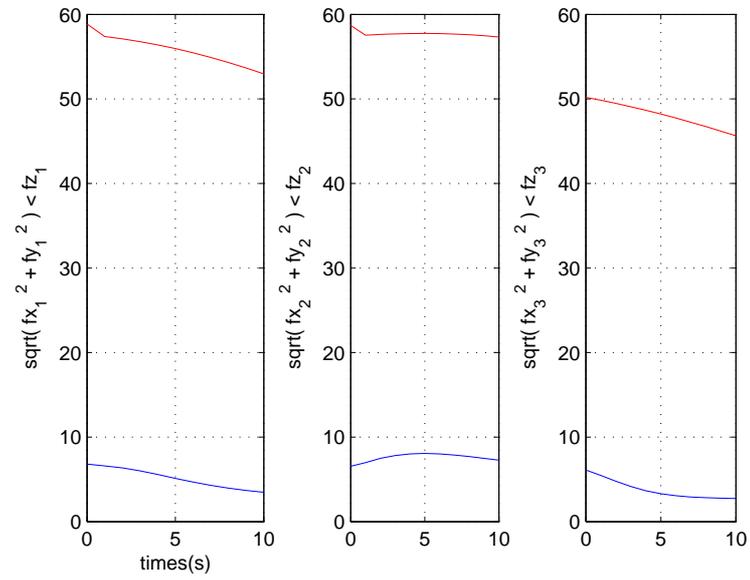


FIG. 4.9 – Contraintes  $f_{x_j}^2 + f_{y_j}^2 \leq \mu^2 f_{z_j}^2$  vérifiée

## 4.3 Conclusion

Dans ce chapitre on a décrit un algorithme efficace pour la résolution du problème d'optimisation de distribution des efforts sur l'objet manipulé par les différents robots sous plusieurs contraintes physiques. Les contraintes de friction qui sont des inéquations non linéaires sont transformées en des équations et inéquations linéaires, satisfaisant la condition de non glissement de l'objet entre les effecteurs terminaux des robots. Par ailleurs, par combinaison des lignes du système d'équations d'équilibre avec les équations de friction on réduit d'une unité le nombre d'inconnues. Finalement on introduit une fonction objectif quadratique sous des conditions physiques, formulées par des inéquations linéaires. La solution du système est donnée par la minimisation de cette fonction objectif.

# Chapitre 5

## Dynamique et commande des robots coopératifs

### 5.1 Introduction

La commande de robots manipulateurs, sous la contrainte de mouvements rapides et/ou précis, implique une modélisation mathématique assez fine, renonçant à un maximum d'hypothèses utopiques simplificatrices, comme la négligence des effets dus à l'inertie de la structure ou des interactions avec l'environnement lors de l'établissement des modèles géométriques et cinématiques. Il est alors indispensable d'établir un modèle dynamique prenant en compte les forces de couplages, dans le but de réduire l'écart entre le comportement recherché et celui réellement réalisé, et d'améliorer la vitesse du système. Le modèle dynamique d'un mécanisme articulé peut être défini comme étant une combinaison d'équations différentielles généralement non linéaires, liant les variables d'états du système aux forces extérieures dissipatives ou non, agissant sur chaque corps du mécanisme. Son obtention, dans le cas de structures sérielles, peut être déduite de plusieurs types de représentations

- fonctions de Gibbs,
- travaux virtuels d'Alembert,
- bond - graphes,
- Equations de Newton-Euler,
- Equations de Lagrange

L'établissement du modèle dynamique des structures présentant des chaînes cinématiques complexes fermées est plus délicat à résoudre. Certaines méthodes ont toutefois été développées utilisant notamment le principe de d'Alembert ou le formalisme d'Euler.

Une autre approche utilisant le calcul des multiplicateurs de Lagrange, afin de tenir compte de la dépendance entre les coordonnées articulaires due à la fermeture des boucles cinématiques, a été approfondies. Toutefois, la complexité des calculs rendent ces différentes modélisations inexploitable en vue d'une simulation ou d'une commande dynamique réalisée sous la contrainte temps réelle. La réduction de la complexité du modèle, d'un point de vue temps de calcul, passe par la détermination d'une expression factorisée du modèle cinématique et dynamique. L'utilisation du calcul parallèle sur du matériel informatique approprié, tels que les architectures multi-processeur, devrait alors permettre de contrôler le système poly-articulé en ligne. Dans cette optique, après avoir étudié les principales méthodologies du calcul dynamique des robots manipulateurs parallèles, l'approche, basée sur la factorisation de la matrice Jacobienne et des matrices inertielles dans l'espace articulaire et opérationnel, proposée pour les robots sériel et parallèles sera étendue aux robots coopérants. Cette approche dérive d'un formalisme global qui allège les équations régissant la dynamique des robots, permettant de ce fait une manipulation et une interprétation physique des équations plus aisée. Le nombre d'opérations arithmétiques nécessaires pouvait encore être réduit si le modèle était tout d'abord calculé analytiquement (symboliquement de manière automatique) dans une première phase puis numériquement dans une seconde phase [82] [83].

## 5.2 Modèle dynamique inverse du système multi-robots

Nous présentons dans ce qui suit un modèle dynamique capable de prendre en compte la présence de chaînes fermées. La littérature présente plusieurs modèles pour les robots à structure complexe [84] et [85].

### 5.2.1 Formalisme de Newton-Euler

#### Application du formalisme

En se référant à [69] et [86], l'algorithme de Newton-Euler s'articule autour de deux étapes ; une première étape directe pour le calcul des vitesses et des accélérations et une deuxième étape

pour le calcul des efforts et des moments.

### 1<sup>ère</sup> Etape : calcul des vitesses et accélérations

Soit  $C_i$  un élément quelconque d'un robot "j", Soit  $v_i$  la vitesse du centre de gravité  $G_i$  et  $\omega_i$  la vitesse de rotation instantanée du corps  $C_i$ . Soient  $\gamma_i = \dot{v}_i$  l'accélération du point  $G_i$  et  $a_i = \dot{\omega}_i$  l'accélération de rotation instantanées du corps  $C_i$ . Etant donné que :

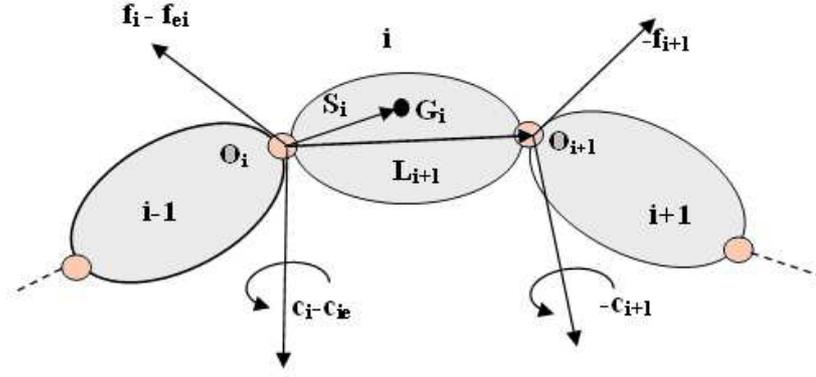


FIG. 5.1 – Bilan des efforts au centre de gravité sur chaque corps

$$P_{oi} = P_{o,i-1} + P_{i-1,i} \quad (5.1)$$

$$d_{oi} = P_{oi} + d_{i,i} \quad (5.2)$$

On peut écrire, en dérivant :

$$\dot{P}_{oi} = \dot{P}_{o,i-1} + \omega_{i-1} \wedge P_{i-1,i} + \sigma_i \dot{\theta}_i Z_i \quad (5.3)$$

avec :  $\sigma_i = 0$  si l'articulation est rotoïdes.

$$v_i = \dot{d}_{oi} = \dot{P}_{oi} + \omega_{i-1} \wedge d_{i,i}$$

Soit, en dérivant une seconde fois :

$$\ddot{P}_{oi} = \ddot{P}_{o,i-1} + a_{i-1} \wedge P_{i-1,i} + \omega_{i-1} \wedge (\omega_{i-1} \wedge P_{i-1,i}) + \sigma_i (2\dot{\theta}_i \omega_{i-1} \wedge Z_i + \ddot{\theta}_i Z_i) \quad (5.4)$$

et

$$\gamma_i = \dot{v}_i = \ddot{d}_{oi} = \ddot{P}_{oi} + a_{i-1} \wedge d_{i,i} + \omega_i \wedge (\omega_{i-1} \wedge d_{i,i}) \quad (5.5)$$

$\gamma_i$  : représente l'accélération du centre de gravité.

En tenant compte de l'accélération de la pesanteur, on obtient alors :

$$\alpha_i = \ddot{\mathbf{P}}_{oi} + g\mathbf{Z}_0 \quad (5.6)$$

Sachant que :

$$\mathbf{a}_i \wedge \star + \omega_i \wedge (\omega_i \wedge \star) = \underline{\mathbf{b}}_i \star \quad (5.7)$$

avec

$$\underline{\mathbf{b}}_i = \underline{\hat{\mathbf{a}}} + \underline{\hat{\omega}} \cdot \underline{\hat{\omega}} \quad (5.8)$$

$$\underline{\hat{\mathbf{a}}} = \begin{pmatrix} 0 & -a_z & a_y \\ a_z & 0 & -a_x \\ -a_y & a_x & 0 \end{pmatrix} ; \quad \underline{\hat{\omega}} = \begin{pmatrix} 0 & -\omega_z & \omega_y \\ \omega_z & 0 & -\omega_x \\ -\omega_y & \omega_x & 0 \end{pmatrix}$$

$\underline{\hat{\mathbf{a}}}$ ,  $\underline{\hat{\omega}}$  : représentent les matrices préproduit. D'une façon plus condensée on peut écrire :

$$\alpha_i = \sigma_i \mathbf{t}_i + r_{i-1} \quad (5.9)$$

avec :

$$\begin{cases} \mathbf{t}_i = 2\dot{\theta}_i \omega_{i-1} \wedge \mathbf{Z}_i + \ddot{\theta}_i \mathbf{Z}_i \\ \mathbf{r}_{i-1} = \alpha_{i-1} + \underline{\mathbf{b}}_{i-1} \mathbf{P}_{i-1,i} \end{cases}$$

remarque : pour notre cas  $\sigma_i \mathbf{t}_i = 0$  car  $\sigma_i = 0$ . De même :

$$\gamma_i + g\mathbf{Z}_0 = \alpha_i + \underline{\mathbf{b}}_i \mathbf{d}_{ii} \quad (5.10)$$

## 2<sup>eme</sup> Etape : Calcul des forces et moments

- Soit  $\mathbf{c}_i$  : le couple d'interaction exercé par le corps  $C_{i-1}$  sur le corps  $C_i$ .
- $\mathbf{f}_i$  : la force d'interaction exercée au point  $o_i$  par le corps  $C_{i-1}$  sur le corps  $C_i$ .
- $\mathbf{F}^i = \mathbf{f}_i + \sigma_i \Gamma^i \mathbf{Z}_i$  : la force exercée, au point  $o_i$  par le  $i^{eme}$  actionneur sur le corps  $C_i$ .
- $\mathbf{C}^i = \mathbf{c}_i + \bar{\sigma}_i \Gamma^i \mathbf{Z}_i$  : le couple exercé, par le corps  $C_{i-1}$  et le  $i^{eme}$  actionneur sur le corps  $C_i$ .

$\Gamma^i$  représente la valeur du couple fourni par le moteur de l'articulation "i" d'unité [N.m].

L'équilibre du corps  $C_i$  est donné par les deux équations suivantes :

$$\mathbf{F}_{res}^i = \mathbf{F}^i - \mathbf{F}^{i+1} - m_i g \mathbf{Z}_0 = m_i \gamma_i \quad (5.11)$$

$$\mathbf{C}_{res}^i = \mathbf{C}^i - \mathbf{C}^{i+1} - \mathbf{d}_{i,i} \wedge \mathbf{F}^i + \mathbf{d}_{i+1,i} \wedge \mathbf{F}^{i+1} \quad (5.12)$$

de même

$$\mathbf{C}_{res}^i = \varphi^i \mathbf{a}_i - \omega_i \wedge (\varphi^i \omega_i) \quad (5.13)$$

avec :

$$\varphi^i = \begin{pmatrix} \int (y^2 + z^2) dm & - \int xy dm & - \int xz dm \\ - \int xy dm & \int (x^2 + z^2) dm & - \int yz dm \\ - \int xz dm & - \int yz dm & \int (x^2 + y^2) dm \end{pmatrix}$$

avec :  $\varphi^i$  : tenseur d'inertie du corps  $C_i$

De l'équation (5.11) on trouve :

$$\mathbf{F}^i = \mathbf{F}^{i+1} + m_i(\alpha_i + \underline{\mathbf{b}}_i \mathbf{d}_{ii}) \quad (5.14)$$

de (5.12) et (5.13) on obtient :

$$\mathbf{C}^i = \mathbf{C}^{i+1} + \varphi^i \mathbf{a}_i + \omega_i \wedge (\varphi^i \omega_i) + \mathbf{d}_{ii} \wedge (\mathbf{F}^i - \mathbf{F}^{i+1}) + \mathbf{P}_{i,i+1} \wedge \mathbf{F}^{i+1} \quad (5.15)$$

Une fois le calcul terminé pour tout les robots, une équation de fermeture de la chaîne est nécessaire pour la l'objet manipulé. Soient :

- $\gamma_{0,0}$  : l'accélération linéaire désirée de l'objet manipulé exprimée dans le repère  $(x_0, y_0, z_0)$ .
- $\omega_{0,0}$  : l'accélération angulaire désirée de l'objet manipulé
- $\mathbf{F}_{1,j}^0$  : la force appliquée par le robot "j" au niveau de l'articulation "6" sur l'objet manipulé "0".
- $\mathbf{C}_{1,j}^0$  : le moment résultant appliqué par le robot "j" au niveau de l'articulation "6" sur l'objet manipulé "0".
- $\mathbf{P}_{01,j}^0$  : la distance entre l'articulation "6" du robot "j" et l'origine du repère  $(x_0, y_0, z_0)$  exprimée dans ce dernier.

L'application du principe fondamental de la dynamique, au centre de masse de l'objet manipulé permet d'obtenir l'équation matricielle suivante :

$$\begin{pmatrix} m_0 \mathbf{I}_3 & \mathbf{0} \\ \mathbf{0} & \underline{\varphi}^0 \end{pmatrix} \begin{pmatrix} \gamma_{0,0} \\ \omega_{0,0} \end{pmatrix} + \begin{pmatrix} -m_0 \mathbf{g}_0 \\ \omega_{0,0} \wedge (\underline{\varphi}^0 \omega_{0,0}) \end{pmatrix} = \begin{pmatrix} \mathbf{F} \\ \mathbf{M} \end{pmatrix} \quad (5.16)$$

où :

- $m_0$  : la masse de l'objet manipulé

- $\mathbf{I}_3$  : la matrice identité  $3 \times 3$ .
- $\underline{\varphi}^0$  : la matrice d'inertie de l'objet manipulé
- $\mathbf{g}_0$  : le vecteur gravité

$\mathbf{F}$  et  $\mathbf{M}$  sont définies comme suit :

$$\begin{cases} \mathbf{F} = \sum_{j=1}^6 \mathbf{F}_{1,j}^0 \\ \mathbf{M} = \sum_{j=1}^6 (\mathbf{C}_{1,j}^0 + \mathbf{P}_{01,j}^0 \wedge \mathbf{F}_{1,j}^0) \end{cases} \quad (5.17)$$

### 5.2.2 Implémentation de la méthode de Newton-Euler

La méthode de Newton-Euler peut exploiter un calcul extrinsèque, qui consiste à projeter les quantités qui interviennent dans le calcul alternativement dans les repères  $R_{i-1}$  et  $R_i$  ce qui peut être représenté par les organigrammes suivants ou :

$\mathbf{R}_{i,i-1}$  représente la matrice de rotation extraite de la matrice de transformation  $\mathbf{T}_{i,i-1}$ .

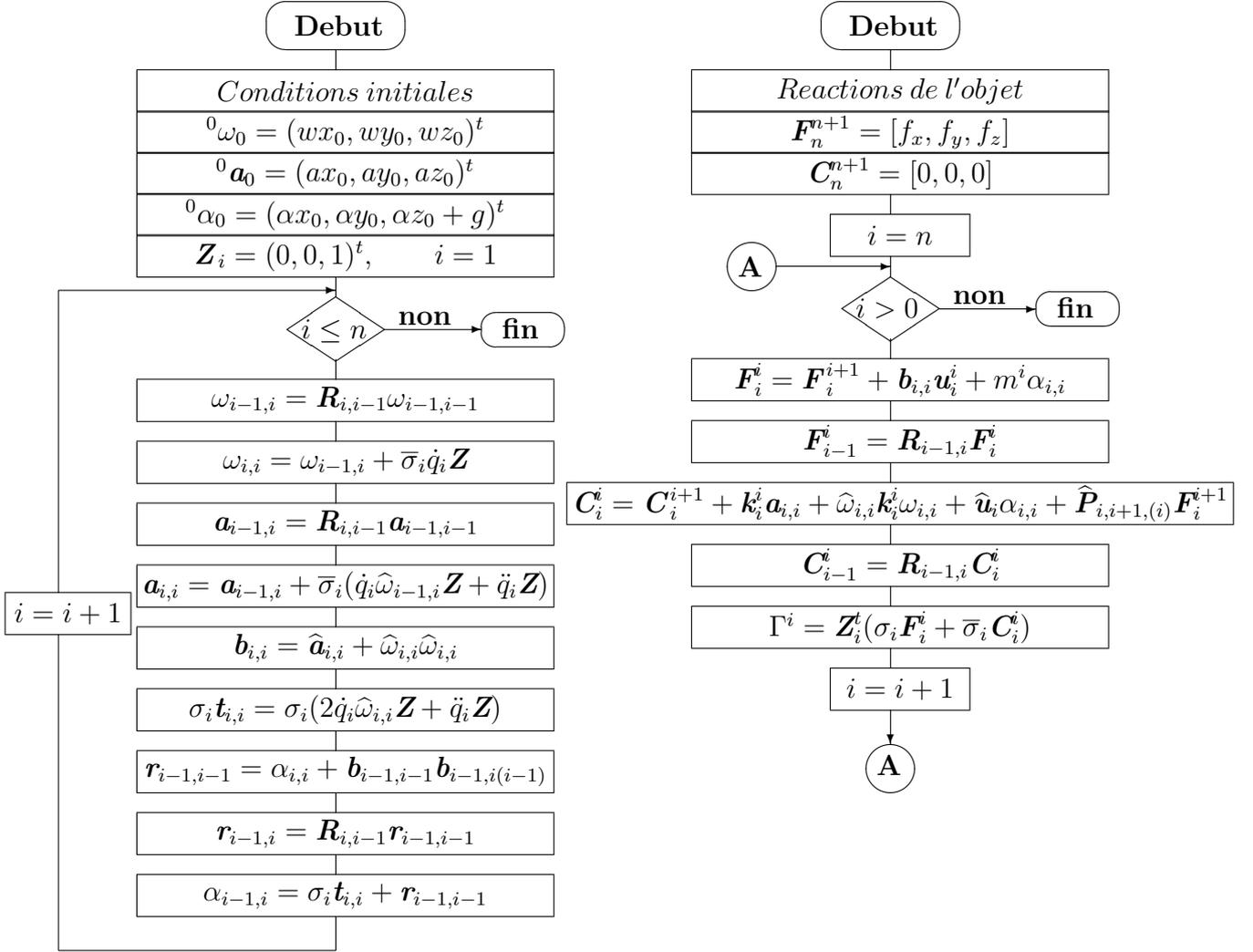


FIG. 5.2 – Algorithmes cinématique et dynamique

### 5.2.3 Formalisme de Lagrange

Le formalisme de Lagrange décrit les équations du mouvement en terme de travail et d'énergie du système, ce qui se traduit, lorsque l'effet extérieur sur l'organe terminal est nul par le système d'équations suivants :

$$\Gamma_i = \frac{d}{dt} \frac{\partial L}{\partial \dot{\theta}_i} - \frac{\partial L}{\partial \theta_i} \quad i = 1, \dots, n \quad (5.18)$$

avec :

- $L$  : le lagrangien du système égal à  $E - U$
- $E$  : l'énergie cinétique totale du système
- $U$  : l'énergie potentielle totale du système

## Forme générale des équations dynamiques

L'énergie cinétique du système est une fonction quadratique des vitesses articulaires

$$E = \frac{1}{2} \dot{\theta}_i^T \mathbf{A} \dot{\theta}_i \quad (5.19)$$

avec  $\mathbf{A}$  la matrice d'inertie du robot, d'éléments génériques  $\mathbf{A}_{ij}$ . C'est une matrice de dimension  $(n \times n)$ , symétrique et définie positive. Ses éléments sont fonction des variables articulaires  $\theta_i$ . L'énergie potentielle étant fonction des variables articulaires  $\theta_i$ , le couple  $\Gamma$  s'écrit à partir des équations (5.18) et (5.19) sous la forme suivante :

$$\Gamma = \mathbf{A}(\theta)\ddot{\theta} + \mathbf{C}(\theta, \dot{\theta})\dot{\theta} + \mathbf{Q}(\theta) \quad (5.20)$$

Avec :

- $\mathbf{C}(\theta, \dot{\theta})\dot{\theta}$  : le vecteur de dimension  $(n \times 1)$  représentant les forces de coriolis et les forces centrifuges, telque :

$$\mathbf{C}\dot{\theta} = \dot{\mathbf{A}}\dot{\theta} - \frac{\partial E}{\partial \theta}$$

- $\mathbf{Q} = [\mathbf{Q}_1, \dots, \mathbf{Q}_n]$  : vecteur des couples et forces de gravité. Les éléments du vecteur  $\mathbf{Q}$  se calculent en écrivant que :

$$\mathbf{Q}_i = \frac{\partial U}{\partial \theta_i}$$

## Calcul de l'énergie cinétique

L'énergie cinétique totale du système est donnée par la relation :

$$E = \sum_{j=1}^n E_j \quad (5.21)$$

Où  $E_j$  désigne l'énergie cinétique du corps  $C_j$ , donnée par l'équation suivante :

$$\mathbf{E}_j = \frac{1}{2} (\omega_j^T \varphi^j \omega_j + m_j \mathbf{V}_{G_j}^T \mathbf{V}_{G_j}) \quad (5.22)$$

Avec :

$\omega_j$  : la vitesse instantanée de rotation du corps "j" exprimée dans le repère "j".

$\mathbf{V}_{G_j}$  : la vitesse linéaire du centre de gravité du corps "j" exprimée dans le repère "j".

$$\mathbf{V}_{G_j} = {}^j \mathbf{V}_j + {}^j \omega_j \mathbf{d}_{jj} \quad (5.23)$$

sachant que :

$$\mathbf{k}_j = \varphi^j - m_j \widehat{\mathbf{d}}_{jj} \widehat{\mathbf{d}}_{jj} \quad (5.24)$$

l'équation (5.22) s'écrit alors :

$$\mathbf{E}_j = \frac{1}{2} (\omega_j^T {}^j \mathbf{k}_j {}^j \omega_j + m_j \mathbf{V}_j^T {}^j \mathbf{V}_j + 2m_j {}^j \mathbf{d}_{jj}^T ({}^j \mathbf{V}_j \wedge {}^j \omega_j)) \quad (5.25)$$

avec :

$${}^j \omega_j = {}^j \mathbf{A}_{j-1} {}^{j-1} \omega_{j-1} + \bar{\sigma}_j \dot{\theta}_j {}^j \mathbf{Z}_j = {}^j \omega_{j-1} + \bar{\sigma}_j \dot{\theta}_j {}^j \mathbf{Z}_j \quad (5.26)$$

$${}^j \mathbf{V}_j = {}^j \mathbf{A}_{j-1} ({}^{j-1} \mathbf{V}_{j-1} + {}^{j-1} \omega_{j-1} \wedge \mathbf{P}_{j-1,j}) + \sigma_j \dot{\theta}_j {}^j \mathbf{Z}_j \quad (5.27)$$

### Calcul de l'énergie potentielle

$$U = \sum_{j=1}^n U_j = \sum_{j=1}^n -m_j g^T (\mathbf{d}_{oj} + \mathbf{d}_{jj}) \quad (5.28)$$

$$U_j = -m_j g_0^T (\mathbf{d}_{oj} + {}^0 \mathbf{A}_j {}^j \mathbf{d}_{jj}) \quad (5.29)$$

Ainsi en considérant un robot de Nddl soumis à un tenseur d'effort  $\mathbf{f}$  au niveau de l'effecteur terminal, l'équation générale qui décrit la dynamique du robot s'écrit sous la forme suivante :

$$\Gamma - \mathbf{J}^T \mathbf{f} = \mathbf{A}(\theta) \ddot{\theta} + \mathbf{C}(\theta, \dot{\theta}) \dot{\theta} + \mathbf{Q}(\theta) \quad (5.30)$$

avec  $\mathbf{A} \in \mathbb{R}^{6 \times 6}$ ,  $\mathbf{C} \in \mathbb{R}^{6 \times 1}$ ,  $\mathbf{Q} \in \mathbb{R}^{6 \times 1}$  et  $(\mathbf{J}^T \mathbf{f}) \in \mathbb{R}^{6 \times 1}$ , on a 6 coordonnées articulaires et opérationnelles pour chaque robot.

Une deuxième écriture de l'équation précédente peut être donnée par :

$$\Gamma = \mathbf{A}(\theta) \ddot{\theta} + \mathbf{H}(\theta, \dot{\theta}) \quad (5.31)$$

avec :

$$\mathbf{H}(\theta, \dot{\theta}) = \mathbf{C}(\theta, \dot{\theta}) \dot{\theta} + \mathbf{Q}(\theta) + \mathbf{J}^T \mathbf{f}$$

Ce passage nous permet d'écrire :

$$\mathbf{H}(\theta, \dot{\theta}) = \Gamma \quad \text{si } \ddot{\theta} = 0$$

donc, en utilisant le formalisme de Newton-Euler avec les  $\ddot{\theta} = 0$ , on peut calculer les  $\mathbf{H}(\theta, \dot{\theta})$  qui correspondent à la troisième composante des couples, autrement dit les couples articulaires.

Ce passage a en réalité deux fonctions :

- il permet l'extraction du vecteur accélération  $\ddot{\theta}$ , en effet :

$$\ddot{\theta} = \mathbf{A}^{-1}(\theta)(\Gamma - \mathbf{H}(\theta, \dot{\theta})),$$

- cela va permettre d'éviter le calcul du vecteur  $\mathbf{C}(\theta, \dot{\theta})$  qui a un algorithme de calcul très redondant.

### 5.3 Dynamique du système multi-robots coopératif

L'équation de la dynamique dans l'espace articulaire du manipulateur  $i$  est donnée par :

$$\mathcal{M}_i(q_i)\ddot{q}_i + C_i(q_i, \dot{q}_i) + G_i(q_i) + J_i^T(q_i)F_{ei} = \tau_i \quad (5.32)$$

ou  $q_i$ ,  $\mathcal{M}_i$ ,  $\tau_i$ ,  $C_i(q_i, \dot{q}_i)$ ,  $G_i(q_i)$  sont définis pour le robot  $i$ ,  $F_{ei} \in \mathfrak{R}^n$  représente la force exercée par le robot  $i$  sur l'objet manipulé au point de contact, et  $J_i(q_i) \in \mathfrak{R}^{n \times n}$  est la matrice Jacobienne du robot  $i$ .

récrivons l'équation de la dynamique pour  $m$  manipulateurs d'une manière plus compacte comme suit :

$$\mathcal{M}(q)\ddot{q} + C(q, \dot{q}) + G(q) + J^T(q)F_e = \tau \quad (5.33)$$

ou :

$$q = [q_1^T, q_2^T, \dots, q_m^T]^T \in \mathfrak{R}^{mn}$$

$$\tau = [\tau_1^T, \tau_2^T, \dots, \tau_m^T]^T \in \mathfrak{R}^{mn}$$

$$\mathcal{M}(q) = \text{diag} [\mathcal{M}_1(q_1), \mathcal{M}_2(q_2), \dots, \mathcal{M}_m(q_m)] \in \mathfrak{R}^{mn \times mn}$$

$$C(q, \dot{q}) = \text{diag} [C_1(q_1, \dot{q}_1), C_2(q_2, \dot{q}_2), \dots, C_m(q_m, \dot{q}_m)] \in \mathfrak{R}^{mn \times mn}$$

$$G(q) = \text{diag} [G_1^T(q_1), G_2^T(q_2), \dots, G_m^T(q_m)] \in \mathfrak{R}^{mn}$$

$$F_e = [F_{e1}^T, F_{e2}^T, \dots, F_{em}^T]^T \in \mathfrak{R}^{mn}$$

$$J(q) = \text{diag} [J_1(q_1), J_2(q_2), \dots, J_m(q_m)] \in \mathfrak{R}^{mn \times mn}$$

### 5.4 Equation de la dynamique de l'objet manipulé

L'équation de la dynamique de l'objet manipulé par le système multi-robots manipulateur est donnée par :

$$\mathcal{M}_0(x_0)\ddot{x}_0 + C_0(x_0, \dot{x}_0) + G_0(x_0) = F_0 \quad (5.34)$$

ou  $\mathcal{M}_0(x_0) \in \mathfrak{R}^{n \times n}$  représente la matrice d'inertie de l'objet.

Ou  $x_0 \in \mathfrak{R}^n$  représente le vecteur de position et d'orientation de l'objet,  $F_0 \in \mathfrak{R}^n$  est la force appliquée au centre de gravité de l'objet,  $\mathcal{M}_0 \in \mathfrak{R}^{n \times n}$  est la matrice d'inertie symétrique définie positive de l'objet manipulé,  $C_0(x_0, \dot{x}_0) \in \mathfrak{R}^{n \times n}$  est la matrice représentant les termes de Coriolis et centrifuge et  $G_0(x_0) \in \mathfrak{R}^n$  est le vecteur force de gravité.

Définissons  $J_o(x_o) \in \mathfrak{R}^{mn \times n}$  comme suit :

$$J_o(x_o) = \left[ J_{o1}^T(x_o) \quad J_{o2}^T(x_o) \dots J_{om}^T(x_o) \right]^T \quad (5.35)$$

ou  $J_{oi} \in \mathfrak{R}^{n \times n}$  est la matrice Jacobienne calculée à partir du repère local du centre de gravité de l'objet jusqu'au repère lié à l'effecteur terminal du robot  $i$ . Alors la force  $F_o$  est donnée par [87].

$$F_o = J_o^T(x_o) F_e \quad (5.36)$$

ou  $F_e$  désigne la force au niveau de l'effecteur terminal qui peut être divisée en deux composantes orthogonales, l'une contribuant au mouvement de l'objet et l'autre générant une force interne ; alors  $F_e$  peut être déduite comme suit [87]-[88] :

$$F_e = (J_o^T(x_o))^+ F_o + F_I \quad (5.37)$$

ou  $J_o^T(x_o)^+ \in \mathfrak{R}^{mn \times n}$  est la pseudo-inverse de  $J_o^T(x_o)$  et  $F_I \in \mathfrak{R}^{mn}$  représente la force interne annulant l'espace de  $J_o^T(x_o)$  c'est à dire :

$$J_o^T(x_o) F_I = 0 \quad (5.38)$$

Alors on peut avoir :

$$F_e = (J_o^T(x_o))^+ [\mathcal{M}_0(x_0) \ddot{x}_0 + C_0(x_0, \dot{x}_0) + G_0(x_0)] + F_I \quad (5.39)$$

## 5.5 Dynamique combiné (robots-objet)

Soit  $x_i \in \mathfrak{R}^n$  le vecteur définissant la position et l'orientation de l'effecteur terminal du robot  $i$ , alors il existe une relation entre  $\dot{x}_i$  et  $\dot{q}_i$  liée à la Jacobienne  $J_i q_i$  comme suit :

$$\dot{x}_i = J_i(q_i) \dot{q}_i \quad (5.40)$$

et la relation entre  $\dot{x}_i$  et  $\dot{x}_o$  est donnée par :

$$\dot{x}_i = J_{oi}(x_o) \dot{x}_o \quad (5.41)$$

Alors et après combinaison des deux equations précédentes, on trouve la relation entre les vitesses articulaires du robot  $i$  et la vitesse de l'objet manipulé comme suit :

$$J_i(q_i)\dot{q}_i = J_{oi}(x_o)\dot{x}_o \quad (5.42)$$

on supposant que lorsque l'objet est tenue par l'ensemble des robots manipulateurs, l'espace de manipulation ne présente pas des régions singulières, par conséquent la matrice Jacobiënne  $J_i(q_i)$  est inversible alors on peut déduire que :

$$\dot{q} = J^{-1}(q)J_o(x_o)\dot{x}_o \quad (5.43)$$

Par différentiation de cette equation on obtient :

$$\ddot{q} = J^{-1}(q)J_o(x_o)\ddot{x}_o + \frac{d}{dt} \left( J^{-1}(q)J_o(x_o) \right) \dot{x}_o \quad (5.44)$$

En remplaçant ce résultat dans l'equation de la dynamique de l'ensemble des robots on obtient :

$$\begin{aligned} & \mathcal{M}(q)J^{-1}(q)J_o(x_o)\ddot{x}_o + \mathcal{M}(q)\frac{d}{dt} \left( J^{-1}(q)J_o(x_o) \right) \dot{x}_o + C(q, \dot{q})J^{-1}(q)J_o(x_o)\dot{x}_o \\ & + G(q) + J^T(q)(J_o^T(x_o))^+(\mathcal{M}_0(x_o)\ddot{x}_0 + C_0(x_o, \dot{x}_0) + G_0(x_o) + J^T(q)F_e = \tau \end{aligned}$$

Par multiplication des deux membres de l'equation précédente par  $J_o^T(x_o)J^{-T}(q)$  et sachant que  $J_o^T(x_o)F_I = 0$ , l'equation de la dynamique du système multi-robots coopératif couplé avec l'objet manipulé s'écrit :

$$\mathcal{M}_M(x_o)\ddot{x}_o + C_M(x_o, \dot{x}_o) + G_M(x_o) = \mu_M \quad (5.45)$$

Avec :

$$\begin{aligned} \mathcal{M}_M(x_o) &= J_o^T(x_o)J^{-T}(q)\mathcal{M}(q)J^{-1}(q)J_o(x_o) + \mathcal{M}_o(x_o) \\ C_M(x_o) &= J_o^T(x_o)J^{-T}(q)\mathcal{M}(q)\frac{d}{dt} \left( J^{-1}(q)J_o(x_o) \right) + J_o^T(x_o)J^{-T}(q)C(q, \dot{q})J^{-1}(q)J_o(x_o) + C_o(x_o, \dot{x}_o) \\ G_M(x_o) &= J_o^T(x_o)J^{-T}(q)G(q) + G_o(x_o) \\ \mu_M &= J_o^T(x_o)J^{-T}(q)\tau \end{aligned}$$

## 5.6 Commande des systèmes poly-articulés

### 5.6.1 Introduction

Dans le cadre de la coopération multi-robot , nous visons toujours à accroître les performances en termes de précision et de robustesse grâce à des techniques de commande avancée. Parmi les stratégies de commande existantes, l'approche prédictive est certainement celle (après le PID bien sûr) qui est la plus couramment employée dans le milieu industriel. Cette stratégie, apparue il y a une quarantaine d'années [Propoi 1963] [Lee et Makus 1967] [Rafal et Stevens 1968], consiste à optimiser, à partir des entrées/sorties d'un système (état, couples), le comportement futur prédit du système considéré. La prédiction est faite à partir d'un modèle interne du système sur un intervalle de temps fini appelé horizon de prédiction. La solution du problème d'optimisation est un vecteur de commande dont la première entrée de la séquence optimale est appliquée au système. Le problème est à nouveau résolu sur l'intervalle de temps suivant en utilisant les données du système mises à jour. Les commandes de type PID sont implémentées dans tous les contrôleurs de robots industriels actuels. Le système est considéré comme un système linéaire et chacune de ses articulations est asservie par une commande décentralisée de type PID à gains constants. Dans la pratique, une telle commande est implémentée selon le schéma de la suivante : La loi de commande dans ce cas est exprimé par :

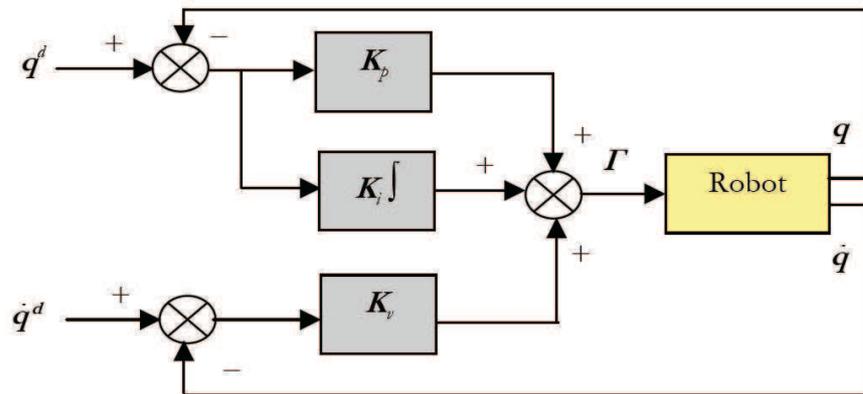


FIG. 5.3 – Schéma de la commande PID

$$\Gamma = k_p(q^d - q) + k_v(\dot{q}^d - \dot{q}) + k_i \int_t^{t_0} (q^d - q) d\tau \quad (5.46)$$

où :

$q$  et  $\dot{q}$  représentent respectivement les positions et vitesses articulaires,

$q^d$  et  $\dot{q}^d$  désignent les positions et vitesses désirées.

$k_p$ ,  $k_v$  et  $k_i$  sont des matrices diagonales définies positives, de dimension  $(n \times n)$ , représentant les gains proportionnels  $k_{pj}$ , dérivés  $k_{vj}$  et intégraux  $k_{ij}$  de chaque articulation  $j$ .

## 5.7 Commande par découplage non linéaire dans l'espace articulaire

Soit  $\hat{A}(\theta)$  et  $\hat{H}(\theta, \dot{\theta})$  les estimations respectives de  $A(\theta)$  et  $H(\theta, \dot{\theta})$ . On suppose que les positions et vitesses articulaires sont mesurables et que les mesures ne sont pas bruitées. Le modèle dynamique direct donne la loi de commande suivante :

$$\Gamma = A(\theta)\ddot{\theta} + \hat{H}(\theta, \dot{\theta}) \quad (5.47)$$

alors :

$$\Gamma = \hat{A}(\theta)\mathbf{w}(t) + \hat{H}(\theta, \dot{\theta}) \quad (5.48)$$

Donc, dans le cas idéal où le modèle est supposé parfait, le vecteur de commande s'écrit :

$$\ddot{\theta} = \mathbf{w}(t) \quad (5.49)$$

On se ramène donc à un problème de commande des systèmes linéaires, invariants, découplés et du second ordre. Si le mouvement désiré est complètement spécifié, alors  $\mathbf{w}(t)$  est donné par :

$$\mathbf{w}(t) = \ddot{\theta} + \mathbf{k}_v(\dot{\theta}_d - \dot{\theta}) + \mathbf{k}_p(\theta_d - \theta) \quad (5.50)$$

Où :

$\ddot{\theta}_d(t)$ ,  $\dot{\theta}_d(t)$  et  $\theta_d(t)$  désignent respectivement, l'accélération, la vitesse et la position désirées dans l'espace articulaire ;

$\mathbf{k}_p$  et  $\mathbf{k}_v$  sont des matrices diagonales définies positives de dimension  $(n \times n)$ ,

$$\mathbf{k}_v = \text{diag}[k_{vi}], \quad \mathbf{k}_p = \text{diag}[k_{pi}]$$

d'après l'équation (5.49), la réponse du système en boucle fermée sera donc décrite par l'équation linéaire découplée suivante :

$$\ddot{e} + \mathbf{k}_v\dot{e} + \mathbf{k}_pe = 0 \quad (5.51)$$

Où  $e(t) = \theta_d(t) - \theta(t)$  et  $\dot{e}(t) = \dot{\theta}_d(t) - \dot{\theta}(t)$ .

La solution de l'équation de l'erreur  $e(t)$  est globalement exponentiellement stable. Les gains  $k_{pi}$  et  $k_{vi}$  sont choisis pour imposer à l'erreur de l'axe  $i$  la dynamique désirée d'amortissement  $\xi_i$  et de pulsation  $\omega_i$ , quelque soit la configuration du robot :

$$k_{pi} = \omega_i^2 k_{vi} = 2\xi_i \omega_i \quad (5.52)$$

En général, on choisit un amortissement égal à 1 pour avoir une réponse sans dépassement. On spécifie le mouvement du robot dans l'espace articulaire et on applique sa commande dans cet espace.

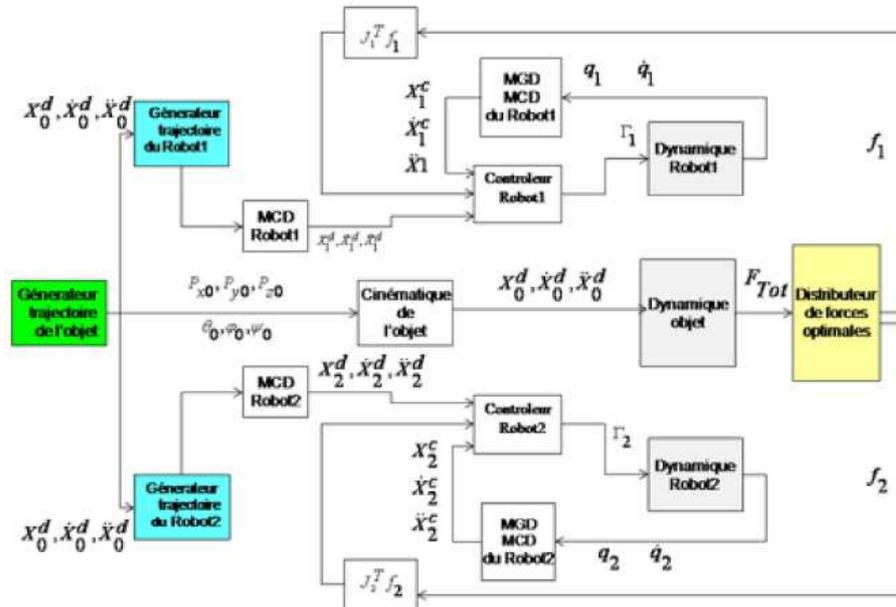


FIG. 5.4 – Architecture de la commande

Le signal de commande aux actionneurs comporte trois parties :

1. La première partie compense les couples de Coriolis, centrifuge, de gravité et de frottement ;
2. La deuxième partie est une correction de position et de vitesse à gains variables représentée respectivement par  $\hat{A}k_p$  et  $\hat{A}k_v$  ;
3. La troisième partie constitue une anticipation des forces d'accélération désirées  $\hat{A}\ddot{\theta}_d$  .

Lorsqu'il y a des erreurs de modélisation, l'équation de la boucle fermée est obtenue en utilisant

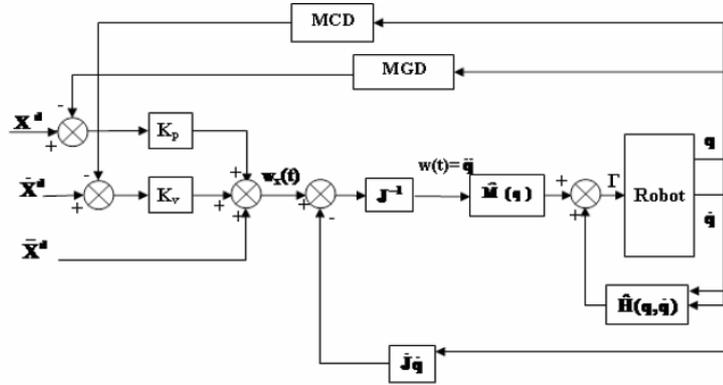


FIG. 5.5 – Torque control of the robot "j"

les relations (5.47) et (5.48) :

$$\hat{A}(\ddot{\theta}_d + k_v \dot{e} + k_p e) + \hat{H} = A\ddot{\theta} + H \quad (5.53)$$

et on déduit que :

$$\ddot{e} + k_v \dot{e} + k_p e = \hat{A}^{-1}[(A - \hat{A})\ddot{\theta} + H - \hat{H}] \quad (5.54)$$

On remarque avec la relation (5.54) que les erreurs de modélisation constituent une excitation pour l'équation de l'erreur. Il en résulte que lorsque ces erreurs sont importantes, il faut augmenter les gains de position et de vitesse, mais ces valeurs sont limitées pour des besoins de stabilité du système.

## 5.8 Commande par découplage non linéaire dans l'espace opérationnel

Les relations entre les variables de l'espace cartésien et celle de l'espace articulaire sont les suivantes :

1. Le modèle géométrique direct :

$$\mathbf{X} = f(\theta) \quad (5.55)$$

Il permet d'obtenir les coordonnées cartésiennes en fonction des coordonnées articulaires.

2. Le modèle géométrique inverse :

$$\theta = g(\mathbf{X}) \quad (5.56)$$

Il permet de calculer les coordonnées articulaires, en fonction des coordonnées cartésiennes.

3. Le modèle cinématique direct :

$$\dot{\mathbf{X}} = \mathbf{J}(\theta)\dot{\theta} \quad (5.57)$$

$\mathbf{J}$  est la matrice jacobienne.

4. Le modèle cinématique inverse :

$$\dot{\theta} = \mathbf{J}(\theta)^{-1}\dot{\mathbf{X}} \quad (5.58)$$

Par dérivation de l'équation (5.57) on obtient :

$$\ddot{\mathbf{X}} = \mathbf{J}(\theta)\ddot{\theta} + \dot{\mathbf{J}}(\theta, \dot{\theta})\dot{\theta} \quad (5.59)$$

avec :

$$\dot{\mathbf{J}}(\theta, \dot{\theta}) = \frac{d\mathbf{J}(\theta)}{dt}$$

A partir de l'équation (5.59) on écrit :

$$\ddot{\theta} = \mathbf{J}(\theta)^{-1}(\ddot{\mathbf{X}} - \dot{\mathbf{J}}(\theta, \dot{\theta})\dot{\theta}) \quad (5.60)$$

Le comportement dynamique du robot dans l'espace opérationnel est décrit par l'équation suivante, obtenue en substituant à  $\ddot{\theta}$  dans l'équation (6.65), son expression est donnée par l'équation (5.60).

$$\Gamma = \mathbf{A}(\theta)\mathbf{J}(\theta)^{-1}(\ddot{\mathbf{X}} - \dot{\mathbf{J}}(\theta, \dot{\theta})\dot{\theta}) + \mathbf{H}(\theta, \dot{\theta}) \quad (5.61)$$

Soit  $X_d(t)$  la trajectoire désirée du robot. Soit l'erreur de poursuite de la trajectoire produite par les coordonnées articulaires, [69] [84] .

$$\mathbf{e}_X(t) = X_d(t) - X(t) \quad (5.62)$$

Comme pour le cas de la commande dans l'espace articulaire, une loi de commande qui linéarise et découple les équations dans l'espace opérationnel est donnée par :

$$\Gamma = \hat{\mathbf{A}}(\theta)\mathbf{J}(\theta)^{-1}(\mathbf{w}(t) - \dot{\mathbf{J}}(\theta, \dot{\theta})\dot{\theta}) + \hat{\mathbf{H}}(\theta, \dot{\theta}) \quad (5.63)$$

avec :

$$\mathbf{w}(t) = \ddot{\mathbf{X}}_d + \mathbf{k}_v(\dot{\mathbf{X}}_d - \dot{\mathbf{X}}) + \mathbf{k}_p(\mathbf{X}_d - \mathbf{X}) \quad (5.64)$$

En utilisant cette loi de commande, le comportement du robot sera décrit par l'équation :

$$\ddot{\mathbf{e}}_X + \mathbf{k}_v\dot{\mathbf{e}}_X + \mathbf{k}_p\mathbf{e}_X = 0 \quad (5.65)$$

Le schéma de la commande (PID) est représenté en figure (5.3).

## 5.9 Application au cas du système coopératif

Afin de démontrer la faisabilité de l'approche proposée, on considère ici la simulation pour la validation du cas de la coopération de deux robots manipulant le même objet dans l'espace opérationnel. Les simulations sont réalisées sous MATLAB-SIMULINK. Soit un objet de masse  $m_0$  tenu de part et d'autre par deux robots identiques présentant six degré de liberté (6ddl), la saisie de l'objet est considérée dans l'espace opérationnelle. La base de chaque robot est localisée à une distance  $a_1$ . Soit  $X_i \in \mathfrak{R}^6$  ou :  $X_i = [P_{xi}, P_{yi}, P_{zi}, \theta_i, \phi_i, \psi_i]^T$ . la position de l'effecteur terminal du robot  $i$  est désignée par :  $[P_{xi}, P_{yi}, P_{zi}]^T$ , alors que l'orientation est représentée par  $[\theta_i, \phi_i, \psi_i]^T$ . on suppose que le vecteur force (forces généralisées) appliquées à l'objet est connu. La masse  $m(i)$  en Kg, le moment d'inertie  $Ix(i), Iy(i), Iz(i)$  en  $Kg.m^2$  et les longueurs en metre des segments de chaque robot utilisées pour la simulations sont connus est données dans la table(5.1) et  $\mu = 0.05$  représente le coefficient de frottement statique. Nous considérons

Link"i"	$Ix(i)$	$Iy(i)$	$Iz(i)$	m(i) Kg	l(i)
Link"1"	0.01	0.05	0.06	25	1
Link"2"	0.1	0.2	0.3	30	0.75
Link"3"	0.4	0.4	0.2	15	0.50
Link"4"	0.3	0.5	0.01	4	0.25
Link"5"	0.01	0.6	0.03	4	0.10
Link"6"	0.2	0.2	0.6	2	0.1

TAB. 5.1 – Paramètres physique du robot

que la trajectoire désirée de l'objet dans l'espace opérationnelle est donnée comme suit :  $x = 1 + 2.5.\sin(0.3)t$ ,

$y = -3 + 2.t$ ,  $z = r_{obj} = r_1 + \sin(2.t)$ ,  $\theta(t)_z = \pi/2 + 1.5.\sin(t)$  est la rotation autour de l'axe  $ox$  et  $\alpha_x(t) = \pi/2 + 0.5.\sin(0.3t)$  est la rotation autour de l'axe  $oy$ .  $\tau_j = [10\ 10\ 10\ 10\ 10\ 10]^T$  est la valeur maximale du couple articulaire, et pour ce qui est de la fonction objectif Eq(18), la matrice de pondération est choisie comme suit :  $\mathbf{p} = \mathbf{0}$  and  $\mathbf{Q} = \mathbf{I}$  (la matrice identité).

La figure (5.6) montre la trajectoire de l'objet supporté par les deux robots dans le repère X-Y-Z.

Les figures (5.7)et (5.8) montrent la Position  $P_x, P_y, P_z$  et l'orientation  $\theta, \phi, \psi$  calculées respectivement du robot1 et du robot2.

La figure (5.9) illustre montre la Position  $P_x$ ,  $P_y$ ,  $P_z$  et l'orientation  $\theta$ ,  $\phi$ ,  $\psi$  calculées de l'objet manipulé.

Les figures (5.10) et (5.11) montrent le profil des coordonnées articulaires ( $\theta_1, \theta_2, \theta_3, \theta_4, \theta_5, \theta_6$ ) respectivement du robot1 et du robot2.

La figure (5.12) montre l'évolution des forces sur les deux robots.

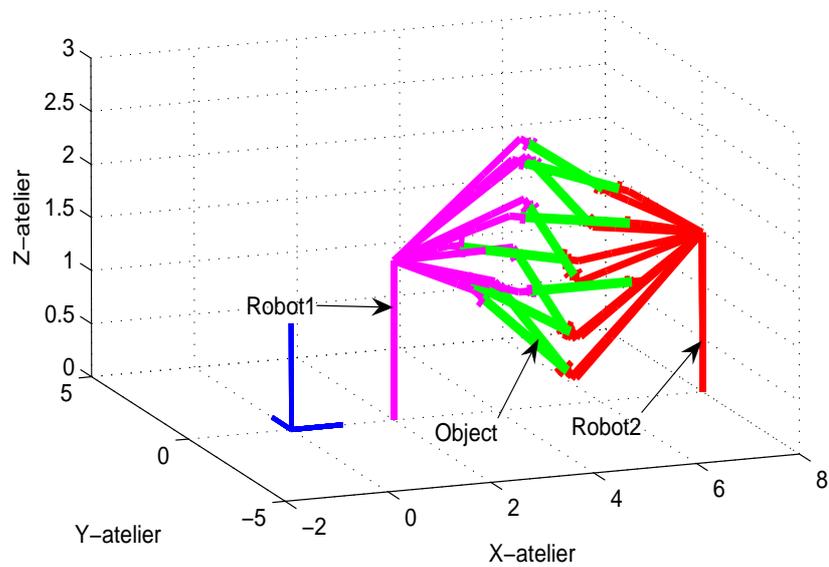


FIG. 5.6 – Vue de l'objet supporté par les deux robots

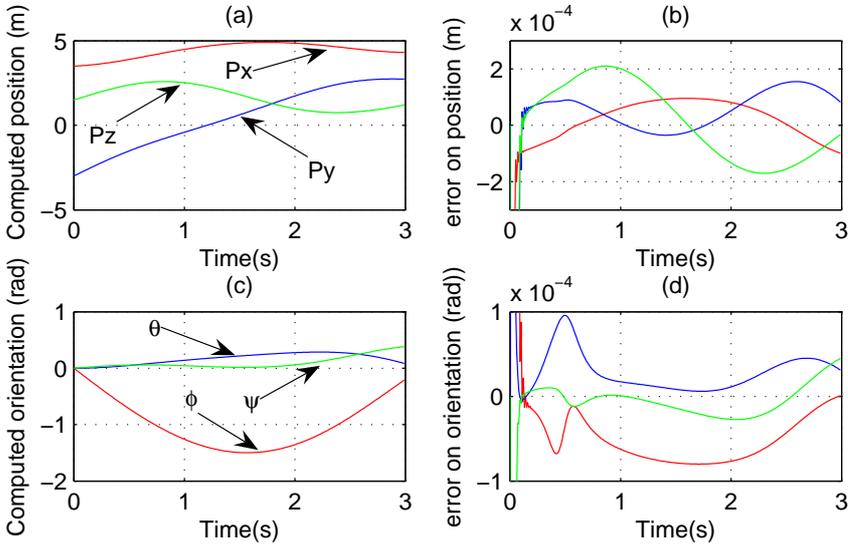


FIG. 5.7 – Position et orientation calculées du robot1

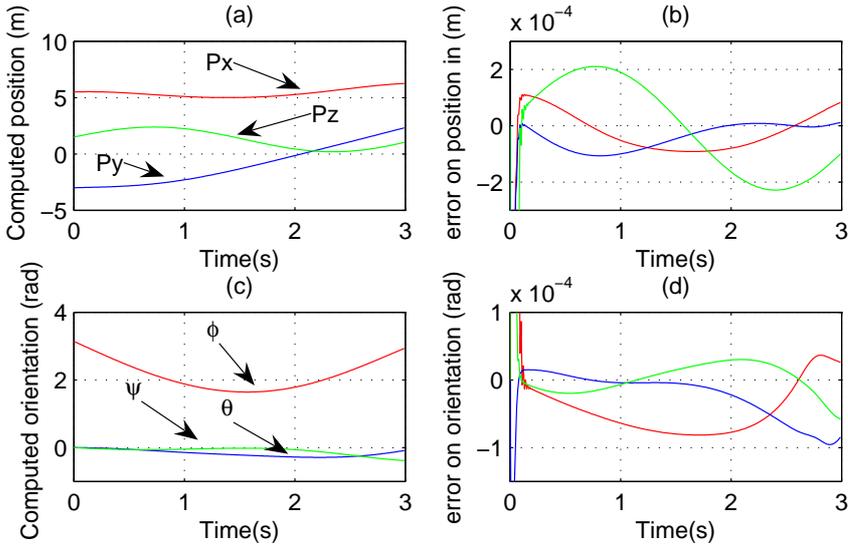


FIG. 5.8 – Position et orientation calculées du robot2

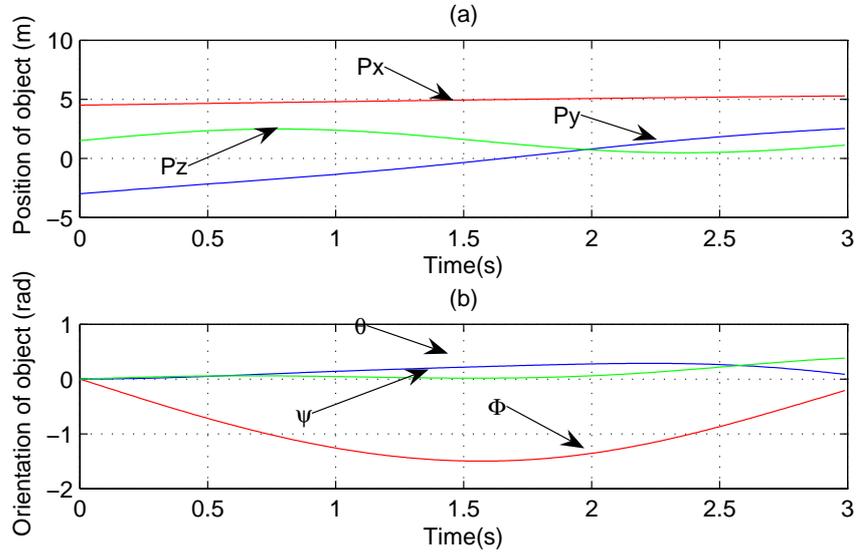


FIG. 5.9 – Position et orientation calculées de l'objet manipulé

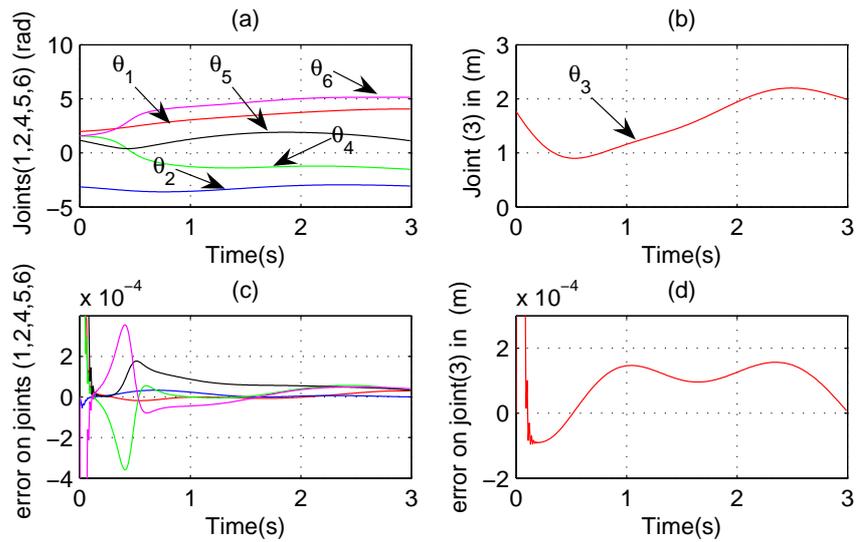


FIG. 5.10 – Coordonnées articulaires du robot1

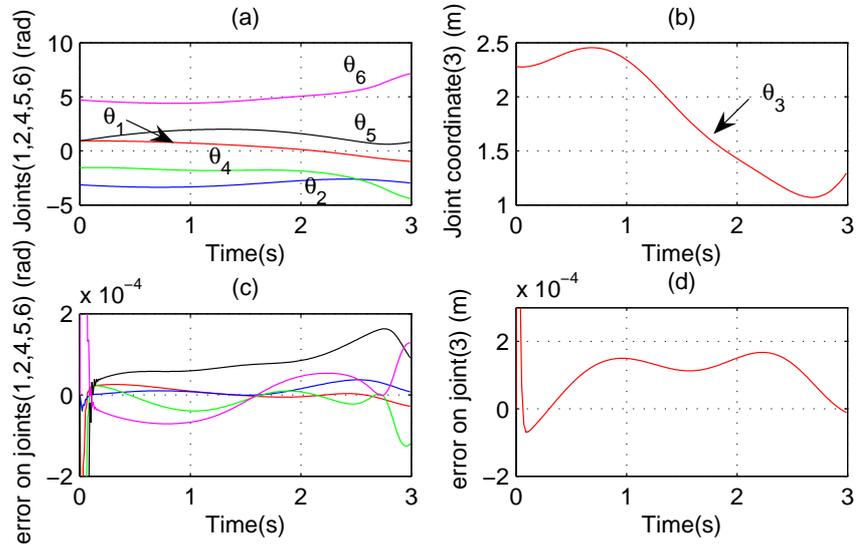


FIG. 5.11 – Coordonnées articulaires du robot2

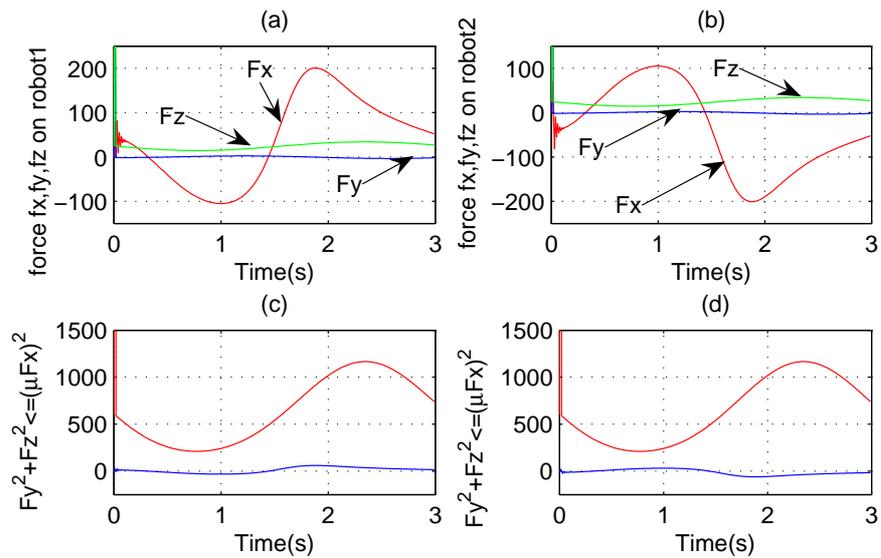


FIG. 5.12 – Forces sur les deux robots

# Chapitre 6

## Calcul parallèle par décomposition des tâches

### 6.1 Introduction

La simulation et l'implémentation temps réel d'algorithmes de commande et de contrôle des systèmes de robots manipulateurs manipulant des objets en commun est un problème crucial. La complexité des robots manipulateurs coopératifs induit une complexité croissante dans les calculs des configurations et des trajectoires et particulièrement dans le cas des robots contraints à coopérer en particulier dans les calculateurs monoprocesseurs. En effet, les calculateurs séquentiels restent limités en puissance de calcul même s'il y a beaucoup d'améliorations. Pour cette raison, les architectures des calculateurs multiprocesseurs parallèles sont nécessaires afin de dépasser cet inconvénient. Cependant, ces architectures parallèles doivent être conçues et organisées de façon cohérente et en adéquation avec le choix des algorithmes appropriés. En ajoutant judicieusement des processeurs et en partageant les tâches convenablement, on peut réduire le temps de calcul global comparativement au séquentiel. Cependant, la méthode de parallélisation est importante car un même problème peut être traité de plusieurs façons en calcul parallèle. L'augmentation du nombre de processeurs dans l'architecture ne conduit pas nécessairement à une réduction du temps de calcul à cause de la complexité de l'implémentation, et aux communications entre processeurs.

1. On traitera le cas du parallélisme selon la méthode de Zomaya, pour des systèmes ouverts et ensuite fermés en coopération.

2. On traitera le cas du parallélisme selon la méthode de Fijani, pour des systèmes ouverts et ensuite fermés en coopération

## 6.2 L'architecture parallèle

Le calcul parallèle multiprocesseurs en robotique concerne à la fois les modèles cinématiques directes et inverses ainsi que les modèles dynamiques directes et inverses. Pour la modélisation du système dynamique, les méthodes les plus utilisées en robotique sont le formalisme de Lagrange-Euler (LE) et celui de Newton-Euler (NE). Le formalisme de (LE), quoique bien structuré et systématique, il a une complexité mathématique de l'ordre de  $O(n^4)$ , où  $n$  est le nombre de ddl du robot. Par contre, le formalisme de Newton-Euler, constitue d'un ensemble d'équations récursives, et plus efficace du point de vue calcul car sa complexité est de l'ordre de  $O(n)$ . Dans le présent travail, on considérera le formalisme de Newton-Euler. On commencera par l'application aux robots sériels en chaîne ouverte, ensuite elle sera étendue aux robots sériels coopératifs donc en chaîne fermée. L'analyse dynamique des systèmes de robots manipulateurs ou le formalisme de Newton-Euler est étendu de celui des chaînes ouvertes à celui des chaînes fermées est présentée en référence [9]

## 6.3 Methode de zomaya

Dans le contexte du calcul parallèle par décomposition des tâches, de nombreuses approches sont proposées, on a sélectionné celle de Zomaya [62] pour sa simplicité. On examinera la méthode proposée par Zomaya pour proposer des architectures de réseau de multiprocesseurs pour le calcul parallèle. Logiquement, le problème fondamental est un problème de gestion de ressources. Il s'agit de distribuer un certain nombre de tâches entre un réseau de processeurs selon un planning à définir. Il s'agit de distribuer  $m$  tâches par rapport à  $p$  processeurs en temps optimal ; c-a-d de façon à minimiser le coût total d'exécution qui comprend les calculs effectués par les processeurs en incluant les délais des communications interprocesseurs. Ce problème est considéré comme très complexe du fait du nombre possibles de combinaisons à envisager. Ce problème est de type NP complet et une analyse systématique n'est pas envisageable.

### 6.3.1 Principe de la méthode :

Elle consiste à décomposer fonctionnellement les équations du modèle dynamique en un ensemble de tâches de calcul élémentaires ou simples. Ensuite, ces tâches sont affectées selon une organisation et un ordonnancement intéressant vers le réseau de calculateurs pour être traité en calcul parallèle. L'implémentation peut se faire dans des transputers. La distribution efficace des tâches sur un réseau de processeurs est assurée grâce à une planification appropriée. Le problème d'ordonnancement est résolu par la répartition optimale des différentes tâches sur  $p$  processeurs, tout en minimisant le coût de calcul impliqué pour l'exécution des différentes tâches affectées. Dans ce cas, le coût représente le temps d'exécution globale de l'algorithme impliquant le temps de communication entre les différents processeurs [89].

## 6.4 Formalisme de Newton Euler pour la dynamique

En utilisant la formulation de Newton-Euler, on peut déterminer le modèle dynamique inverse, connaissant les consignes d'entrée telles que les positions articulaires, les vitesses articulaires et les accélérations articulaires permettant de calculer le couple appliqué. Les équations de la dynamique d'un robot manipulateur dans l'espace articulaire est donné par :

$$\Gamma = \mathcal{M}(\theta) \ddot{\theta} + C(\theta, \dot{\theta}) + G(\theta) \quad (6.1)$$

où  $\mathcal{M}(\theta)$  représente la matrice d'inertie du robot manipulateur.

$G(\theta)$  et  $C(\theta, \dot{\theta})$  représentent respectivement les forces de gravité et les forces de Coriolis appliquées au manipulateur. L'algorithme de la dynamique du robot manipulateur peut être divisé en deux phases Récursives :

#### 1. Phase de récurrence avant :

Dans cette phase, l'itération vers l'avant propage les informations cinématique, tels que les vitesses linéaires, les accélérations angulaires et les accélérations linéaires du centre de masse de chaque liaison, à partir du repère local inertiel de chaque segment jusqu'au repère lié à l'effecteur terminal du robot.

#### 2. Phase de récurrence arrière :

Dans cette phase le processus itératif propage les calculs des forces et moments exercés sur chaque liaison commençant de l'effecteur terminal du robot jusqu'au repère de base.

**Récurrence avant :**

**Initialisation**

$$\omega_{0,0} = (w_{x0}, w_{y0}, w_{z0})^t$$

$$\mathbf{a}_{0,0} = (a_{x0}, a_{y0}, a_{z0})^t$$

$$\alpha_{0,0} = (\alpha_{x0}, \alpha_{y0}, \alpha_{z0} + g)^t$$

$$\mathbf{Z}_i = (0, 0, 1)^t$$

**Debut de la Phase1(Récurrence avant :)**

Pour  $i \in [1\dots n]$  faire

$$\omega_{i-1,i} = \mathbf{R}_{i,i-1} \omega_{i-1,i-1}$$

$$\omega_{i,i} = \omega_{i-1,i} + \bar{\sigma}_i \dot{q}_i \mathbf{Z}$$

$$\mathbf{a}_{i-1,i} = \mathbf{R}_{i,i-1} \mathbf{a}_{i-1,i-1}$$

$$\mathbf{a}_{i,i} = \mathbf{a}_{i-1,i} + \bar{\sigma}_i (\dot{q}_i \widehat{\omega}_{i-1,i} \mathbf{Z} + \ddot{q}_i \mathbf{Z})$$

$$\mathbf{b}_{i,i} = \widehat{\mathbf{a}}_{i,i} + \widehat{\omega}_{i,i} \widehat{\omega}_{i,i}$$

$$\sigma_i \mathbf{t}_{i,i} = \sigma_i (2\dot{q}_i \widehat{\omega}_{i,i} \mathbf{Z} + \ddot{q}_i \mathbf{Z})$$

$$\mathbf{r}_{i-1,i-1} = \alpha_{i,i} + \mathbf{b}_{i-1,i-1} \mathbf{b}_{i-1,i(i-1)}$$

$$\mathbf{r}_{i-1,i} = \mathbf{R}_{i,i-1} \mathbf{r}_{i-1,i-1}$$

$$\alpha_{i-1,i} = \sigma_i \mathbf{t}_{i,i} + \mathbf{r}_{i-1,i-1}$$

**Fin (Phase1)**

**Phase2 (Récurrence arrière) :**

Les équations composant la récurrence arrière sont obtenues à partir du bilan des efforts sur chaque corps, écrit à l'origine :

**Initialisation :**

$$\mathbf{F}_n^{n+1} = [f_x, f_y, f_z]$$

$$\mathbf{C}_n^{n+1} = [0, 0, 0]$$

Pour  $i \in [n\dots 1]$  faire

$$\mathbf{F}_i^i = \mathbf{F}_i^{i+1} + \mathbf{b}_{i,i} \mathbf{u}_i^i + m^i \alpha_{i,i}$$

$$\mathbf{F}_{i-1}^i = \mathbf{R}_{i-1,i} \mathbf{F}_i^i$$

$$\mathbf{C}_i^i = \mathbf{C}_i^{i+1} + \mathbf{k}_i^i \mathbf{a}_{i,i} + \widehat{\omega}_{i,i} \mathbf{k}_i^i \omega_{i,i} + \widehat{\mathbf{u}}_i \alpha_{i,i} + \widehat{\mathbf{P}}_{i,i+1,(i)} \mathbf{F}_i^{i+1}$$

$$\mathbf{C}_{i-1}^i = \mathbf{R}_{i-1,i} \mathbf{C}_i^i$$

$$\Gamma^i = \mathbf{Z}_i^t (\sigma_i \mathbf{F}_i^i + \bar{\sigma}_i \mathbf{C}_i^i)$$

**Fin (Phase2)**

Les deux tableaux présentés en figures (6.1) et (6.2) montrent l'affectation des différentes tâches conformément aux deux phases définies précédemment.

**Phase1 :**

Taches	Nbr d'op de multiplication	Nbr d'op d'addition
${}^1T_1 : R_{i,j-1}$	16n	4n
${}^1T_2 : \omega_{i-1,j} = R_{i,j-1}\omega_{i-1,j-1}$	9n	6n
${}^1T_3 : \omega_{i,j} = \omega_{i-1,j} + \bar{\sigma}_i \dot{q}_i Z$	3(n-1) pour la liaison prismatique	3
${}^1T_4 : a_{i-1,j} = R_{i,j-1}a_{i-1,j-1}$	9n	6n
${}^1T_5 : a_{i,j} = a_{i-1,j} + \bar{\sigma}_i (\bar{q}_i \hat{\omega}_{i-1,j} Z + \ddot{q}_i Z)$	30(n-1) pour la liaison prismatique	12(n-1)
${}^1T_6 : b_{i,j} = \hat{a}_{i,j} + \hat{\omega}_{i,j} \hat{\omega}_{i,j}$	27n	21n
${}^1T_7 : \sigma_i t_{i,j} = \sigma_i (2\dot{q}_i \hat{\omega}_{i,j} Z + \ddot{q}_i Z)$	30 seulement pour la liaison prismatique	9
${}^1T_8 : P_{i-1,j}$	7n	3n
${}^1T_9 : r_{i-1,j-1} = \alpha_{i,j-1} + b_{i-1,j-1} P_{i-1,j}$	9n	9n
${}^1T_{10} : r_{i-1,j} = R_{i,j-1} r_{i-1,j-1}$	9n	6n
${}^1T_{11} : \alpha_{i,j} = \sigma_i t_{i,j} + r_{i-1,j}$	3	0
Total des taches en phase2=11	119n=714	67n=402

FIG. 6.1 – Affectation des taches en première phase

**Phase2 :**

Taches	Nbr d'op de multiplication	Nbr d'op d'addition
${}^2T_1 : u_{i,j}$	3n	0
${}^2T_2 : F_i^i = F_i^{i+1} + b_{i,j}u_i^i + m_i\alpha_{i,j}$	12n	6n
${}^2T_3 : F_{i-1}^i = R_{i-1,j}F_i^i$	9n	6n
${}^2T_4 : k_i^i a_{i,j}$	9n	6n
${}^2T_5 : \hat{\alpha}_{i,j} k_i^i a_{i,j}$	(27.9).n=243	24n
${}^2T_6 : \hat{u}_i \alpha_{i,j}$	9n	6n
${}^2T_7 : \hat{F}_{i,j+1} F_i^{i+1}$	9n	6n
${}^2T_8 : C_i^i = C_i^{i+1} + k_i^i a_{i,j} + \hat{\alpha}_{i,j} k_i^i a_{i,j} + \hat{u}_i \alpha_{i,j} + \hat{F}_{i,j+1} F_i^{i+1}$	0	12n
${}^2T_9 : C_{i-1}^i = R_{i-1,j} C_i^i$	9n	6n
${}^2T_{10} : \Gamma^i = Z_i^t (\sigma_i F_i^i + \bar{\sigma}_i C_i^i)$	3n	2n
Total des taches en phase2=10	306n=	74n=444

FIG. 6.2 – Affectation des taches en deuxième phase

## 6.5 Réorganisation des équations de Newton-Euler en fonction des taches affectées :

Compte tenu des deux tableaux en figure (6.1) et (6.2) résumant la définition de chaque tache à répartir sur les différents processeurs à savoir le calcul de tous les vecteurs et matrices figurant dans les equations de Newton-euler, alors une étape aussi importe sera reservée à la réorganisation des equation de Newton-euler en fonction des taches su-citées. On obtient alors :

**Réorganisation des equation de Newton-Euler en phase1 :**

$${}^1T_1 = \mathbf{R}_{i-1,i}$$

$${}^1T_2 = \omega_{i-1,i} = {}^1T_1 \omega_{i-1,i-1}$$

$${}^1T_3 = \omega_{i,i} = {}^1T_2 + \bar{\sigma}_i \dot{q}_i \mathbf{Z}$$

$${}^1T_4 = \mathbf{a}_{i-1,i} = {}^1T_1 \mathbf{a}_{i-1,i-1}$$

$${}^1T_5 = \mathbf{a}_{i,i} = {}^1T_4 + \bar{\sigma}_i (\dot{q}_i {}^1T_2 \mathbf{Z} + \ddot{q}_i \mathbf{Z})$$

$${}^1T_6 = \mathbf{b}_{i,i} = {}^1T_5 = {}^1T_3 \times {}^1T_3$$

$${}^1T_7 = \sigma_i \mathbf{t}_{i,i} = \sigma_i (2\dot{q}_i {}^1T_3 \mathbf{Z} + \ddot{q}_i \mathbf{Z})$$

$${}^1T_8 = \mathbf{P}_{i,i-1}$$

$${}^1T_9 = \mathbf{r}_{i-1,i-1} = \alpha_{i-1,i-1} {}^1T_6 \times {}^1T_8$$

$${}^1T_{10} = \mathbf{r}_{i-1,i} = {}^1T_1 \times {}^1T_9$$

$${}^1T_{11} = \alpha_{i,i} = {}^1T_7 + {}^1T_{10}$$

**Réorganisation des equation de Newton-Euler en phase2 :**

$${}^2T_1 = \mathbf{u}_i^i$$

$${}^2T_2 = \mathbf{F}_i^i = \mathbf{F}_{i+1}^i + {}^1T_6 \times {}^2T_1 + m_i {}^1T_{11}$$

$${}^2T_3 = \mathbf{F}_i^{i-1} = {}^1T_1 \times {}^2T_2$$

$${}^2T_4 = \mathbf{k}_i^i \mathbf{a}_{i,i} = \mathbf{k}_i^{i1} T_5$$

$${}^2T_5 = \hat{\omega}_{i,i} \mathbf{k}_i^i \omega_{i,i} = {}^1T_3 \cdot \mathbf{k}_i^i \cdot {}^1T_3$$

$${}^2T_6 = \hat{\mathbf{u}}_i \cdot \alpha_{i,i} = {}^2T_1 \times {}^1T_{11}$$

$${}^2T_7 = {}^1T_8 \times {}^2T_3$$

$${}^2T_8 = \mathbf{C}_i^i = \mathbf{C}_{i+1}^i + {}^2T_4 + {}^2T_5 + {}^2T_6 + {}^2T_7$$

$${}^2T_9 = \mathbf{C}_i^{i-1} = {}^1T_1 \times {}^2T_8$$

$${}^2T_{10} = \Gamma^i = \mathbf{Z}_i^t (\sigma_i^2 T_2 + \bar{\sigma}_i^2 T_8)$$

Le schéma en figure (6.3) montre une représentation graphique appelée diagramme des tâches des équations de l'algorithme (NE). Le diagramme explique bien les différentes tâches et les différents niveaux de priorité qui existent entre les différentes tâches. En outre, le graphique illustre la dépendance de données entre les deux phases de l'algorithme (NE)

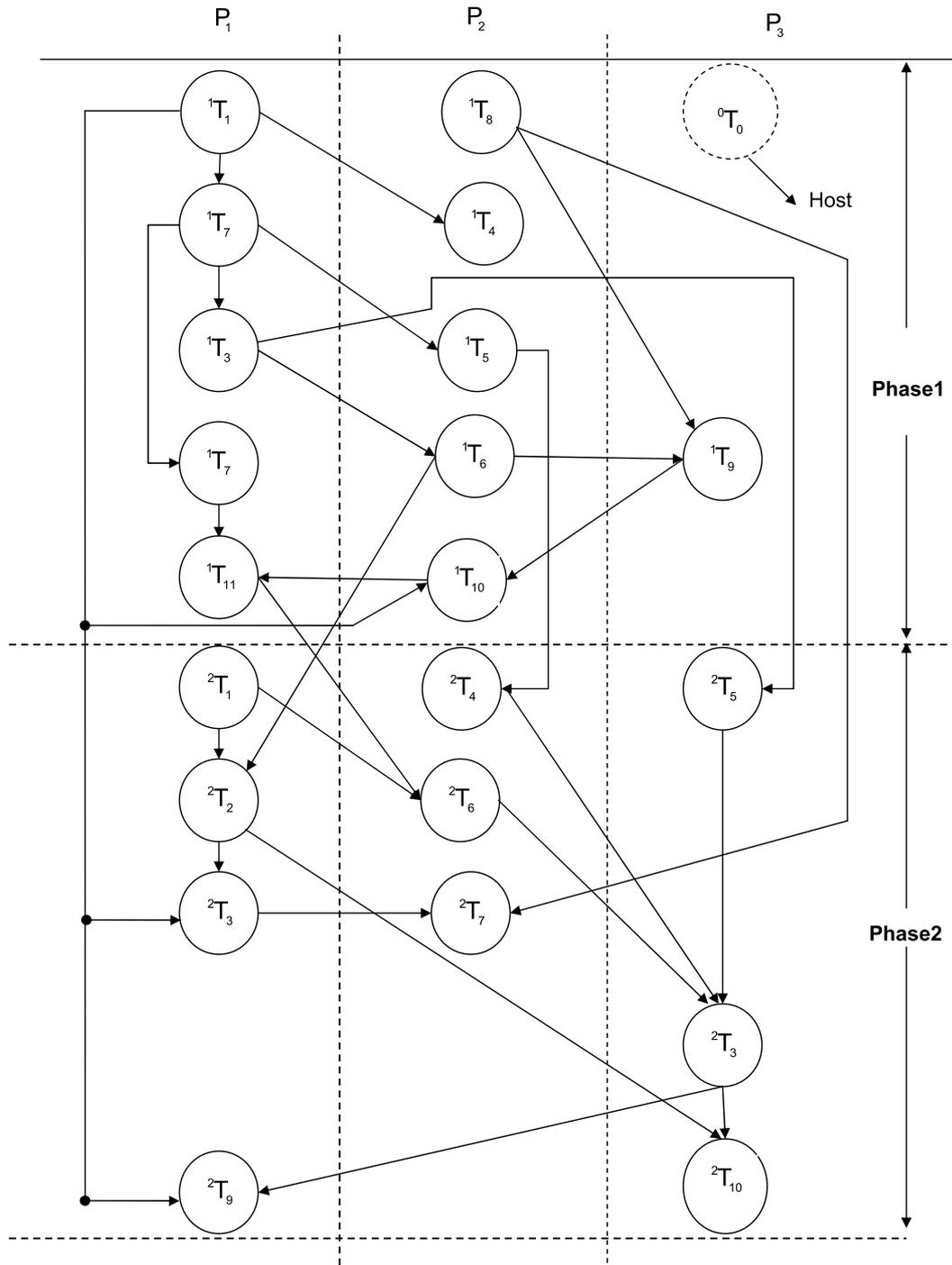


FIG. 6.3 – Diagramme global des tâches réparties sur les trois processeurs

**Application :**

Pour valider l'approche proposée, on prend l'exemple du robot 6ddl présentant une liaison prismatique qui a fait l'objet de notre étude dans les chapitres précédants. La configuration du robot ainsi que ces paramètres géométriques sont présentées en figures (6.4)et (6.5)

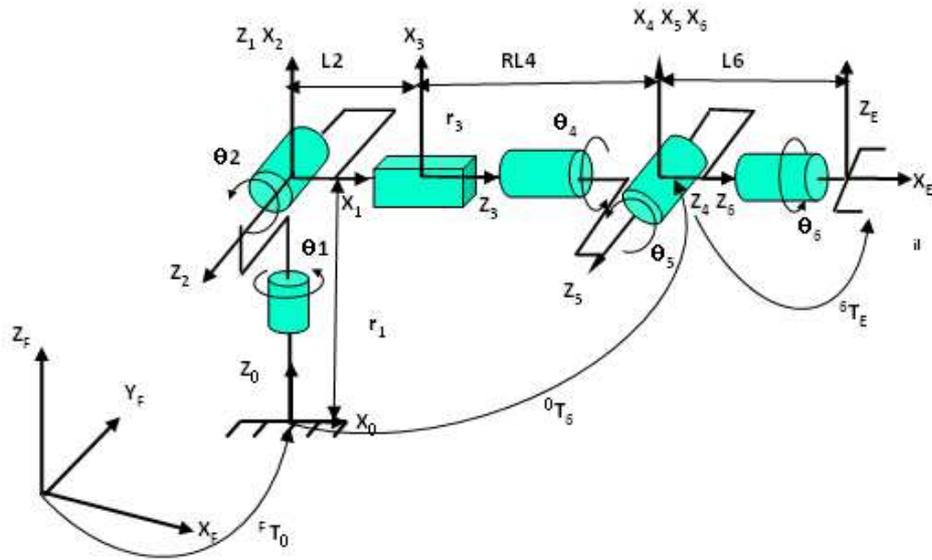


FIG. 6.4 – Exemple d'un Robot 6ddl présentant une liaison prismatique

$i$	$\sigma_i$	$d_i$	$\alpha_i$	$\theta_i$	$J_i$	$m_i$	$s_i$	$r_i$
1	0	$a$	$\alpha_1 = 0$	$\theta_1$	$J_1 = \begin{bmatrix} 0.9 & 0 & 0 \\ 0 & 0.9 & 0 \\ 0 & 0 & 0.6 \end{bmatrix} * 10^{-4}$	$m_1 = 0.7$	$s_1 = [0, 0, -11/2]$	$r_1$
2	0	0	$\alpha_2 = 0$	$\theta_2 + \pi/2$	$J_2 = \begin{bmatrix} 25 & 0 & 0 \\ 0 & 25 & 0 \\ 0 & 0 & 1.46 \end{bmatrix} * 10^{-4}$	$m_2 = 0.5$	$s_2 = [0, -12/2, 0]$	0
3	1	0	$\alpha_3 = \pi/2$	0	$J_3 = \begin{bmatrix} 22 & 0 & 0 \\ 0 & 22 & 0 \\ 0 & 0 & 0.15 \end{bmatrix} * 10^{-4}$	$m_3 = 0.3$	$s_3 = [0, 0, 13/2]$	$L_2 + r_3$
4	0	0	$\alpha_4 = 0$	$\theta_4$	$J_4 = \begin{bmatrix} 22 & 0 & 0 \\ 0 & 22 & 0 \\ 0 & 0 & 0.15 \end{bmatrix} * 10^{-6}$	$m_4 = 0.2$	$s_4 = [0, 0, -14/2]$	$RL_4$
5	0	0	$\alpha_5 = -\pi/2$	$\theta_5$	$J_5 = \begin{bmatrix} 22 & 0 & 0 \\ 0 & 22 & 0 \\ 0 & 0 & 0.15 \end{bmatrix} * 10^{-6}$	$m_5 = 0.1$	$s_5 = [0, -15/2, 0]$	0
6	0	0	$\alpha_6 = \pi/2$	$\theta_6$	$J_6 = \begin{bmatrix} 22 & 0 & 0 \\ 0 & 22 & 0 \\ 0 & 0 & 0.15 \end{bmatrix} * 10^{-6}$	$m_6 = 0.1$	$s_6 = [0, 0, 15 + 16/2]$	0

FIG. 6.5 – Données géométriques du robot

La transformation matricielle permettant le passage du repère  $\mathbf{R}_{i-1}$  au repère  $\mathbf{R}_i$  est donnée par :

$${}^{i-1}\mathbf{T}_i = \text{Rot}(Z, \gamma_i) \text{trans}(Z, b_i) \text{Rot}(X, \alpha_i) \text{trans}(X, d_i) \text{Rot}(Z, \theta_i) \text{trans}(Z, r_i) \quad (6.2)$$

$${}^{i-1}\mathbf{T}_i = \begin{pmatrix} C\gamma_i C\theta_i - S\gamma_i C\alpha_i S\theta_i & -C\gamma_i S\theta_i - S\gamma_i C\alpha_i C\theta_i & S\gamma_i C\alpha_i & d_i C\gamma_i + r_i S\gamma_i S\alpha_i \\ S\gamma_i C\theta_i + C\gamma_i C\alpha_i S\theta_i & -S\gamma_i S\theta_i + C\gamma_i C\alpha_i C\theta_i & -C\gamma_i S\alpha_i & d_i S\gamma_i - r_i C\gamma_i S\alpha_i \\ S\gamma_i S\theta_i & S\gamma_i C\theta_i & C\alpha_i & r_i C\alpha_i + b_i \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (6.3)$$

Ce qui permettra d'extraire les deux matrices  ${}^{i-1}\mathbf{R}_i$  et  ${}^{i-1}\mathbf{P}_i$ , respectivement d'orientation et de position suivantes nécessaire dans l'algorithme de Newton-Euler.

$${}^{i-1}\mathbf{R}_i = \begin{pmatrix} C\gamma_i C\theta_i - S\gamma_i C\alpha_i S\theta_i & -C\gamma_i S\theta_i - S\gamma_i C\alpha_i C\theta_i & S\gamma_i C\alpha_i \\ S\gamma_i C\theta_i + C\gamma_i C\alpha_i S\theta_i & -S\gamma_i S\theta_i + C\gamma_i C\alpha_i C\theta_i & -C\gamma_i S\alpha_i \\ S\gamma_i S\theta_i & S\gamma_i C\theta_i & C\alpha_i \end{pmatrix} \quad (6.4)$$

$${}^{i-1}\mathbf{P}_i = \begin{pmatrix} d_i C \gamma_i + r_i S \gamma_i S \alpha_i \\ d_i S \gamma_i - r_i C \gamma_i S \alpha_i \\ r_i C \alpha_i + b_i \end{pmatrix} \quad (6.5)$$

Le cadre de la simulation concerne le calcul du temps global du couple appliqué au robot manipulateur en utilisant les équations de la dynamique de Newton-Euler dans le cas parallèle.

Après avoir réparti les tâches sur les différents processeurs, et réorganiser les équations de Newton-Euler en fonction de ces tâches, le temps global de la simulation dans le cas parallèle se calcule comme suit :

### Calcul du temps global :

Le temps pour chaque tâche est normalement égal au temps d'exécution des tâches antérieures augmenté du temps d'exécution de la tâche elle-même, alors le temps global pour exécuter les différentes tâches représentées sur le digramme des tâches est donné par les relations suivantes :

$$t_T = \max [t_c({}^2T_9), t_c({}^2T_{10})] \quad (6.6)$$

Où par la suite  $t_c({}^jT_i)$  désigne le temps de chemin antérieur à la tâche .

${}^jT_i$  représente la tâche i dans la phase j.

$$t_c({}^2T_9) = \max [{}^1T_1, t_c({}^2T_8)] + {}^2T_9 \quad (6.7)$$

$$t_c({}^2T_8) = \max [t_c({}^2T_5), t_c({}^2T_4), t_c({}^2T_6)] + {}^2T_8 \quad (6.8)$$

$$t_c({}^2T_4) = {}^2T_4 + t_c({}^1T_5) \quad (6.9)$$

$$t_c({}^1T_5) = \max [({}^1T_2 + {}^1T_1), ({}^1T_4 + {}^1T_1)] + {}^1T_5 \quad (6.10)$$

$$t_c({}^2T_5) = {}^2T_5 + {}^1T_3 + {}^1T_2 + {}^1T_1 \quad (6.11)$$

$$t_c({}^2T_6) = {}^2T_6 + \max [({}^2T_1), t_c({}^1T_{11})] \quad (6.12)$$

$$t_c({}^1T_{11}) = {}^1T_{11} + \max [({}^1T_1 + {}^1T_2 + {}^1T_7), t_c({}^1T_{10})] \quad (6.13)$$

$$t_c({}^1T_{10}) = {}^1T_{10} + \max [{}^1T_1, t_c({}^1T_9)] \quad (6.14)$$

$$t_c({}^1T_6) = {}^1T_6 + \max [({}^1T_1 + {}^1T_2 + {}^1T_3), t_c({}^1T_5)] \quad (6.15)$$

$$t_c({}^1T_9) = {}^1T_9 + \max [{}^1T_8, t_c({}^1T_6)] \quad (6.16)$$

$$t_c({}^2T_{10}) = {}^2T_{10} + \max [t_c({}^2T_2), t_c({}^2T_8)] \quad (6.17)$$

$$t_c({}^2T_2) = {}^2T_2 + \max [{}^2T_1, t_c({}^1T_6)] \quad (6.18)$$

## 6.6 LabVIEW (Laboratory Virtual Instrumentation Engineering Workbench)

LabVIEW est un environnement de développement graphique appelé langage G qui permet de créer des applications modulaires (notion de sous *VI* 'virtual instrument') et extensibles, pour la conception d'applications de mesure, de contrôle et de test.

LabVIEW est un langage de programmation graphique qui utilise des icônes à la place de lignes de texte pour créer des applications. Contrairement aux langages de programmation textuels où les instructions déterminent l'exécution du programme, LabVIEW utilise une programmation par flux de données dans laquelle les données déterminent l'exécution. Pour le développeur, un programme en langage G se présente comme un schéma, le diagramme, réunissant différentes icônes reliées par des fils de couleur. Chaque fil symbolise le passage d'une donnée depuis une source dont elle sort (comme résultat), vers une cible où elle entre (comme paramètre).

### 6.6.1 L'exécution des programmes sous LabVIEW

LabVIEW utilise un modèle de flux de données pour exécuter les VIs. Un noeud de diagramme ne s'exécute que lorsque toutes ses entrées sont disponibles. Lorsque un noeud termine une exécution, il fournit les données à ses terminaux de sortie et transmet les données de sortie au noeud suivant dans le chemin de flux de données, voir figure (6.6).

### 6.6.2 L'implémentation de l'algorithme Newton-Euler sous LabVIEW

Les différentes tâches dans l'algorithme de Newton-Euler ont été implémentées en utilisant le block " MATLAB script ", ce block permet l'utilisation des équations programmées avec Matlab directement ; il suffit d'ajouter les entrées et les sorties nécessaires pour l'exécution de cette tâche sur le cadre de ce block et aussi pour savoir déterminer et gérer le flux de données. On a comme exemple la tâche T51 qui utilise les deux données "ai" (accélérations angulaires) et " wci " (vitesses angulaires) comme donnée d'entrée et calcule la donnée de sortie " as " (accélérations angulaires). Dans ce cas la tâche T51 ne peut pas être exécutée que si les deux entrées ai et wci sont présentes, figure (6.7).

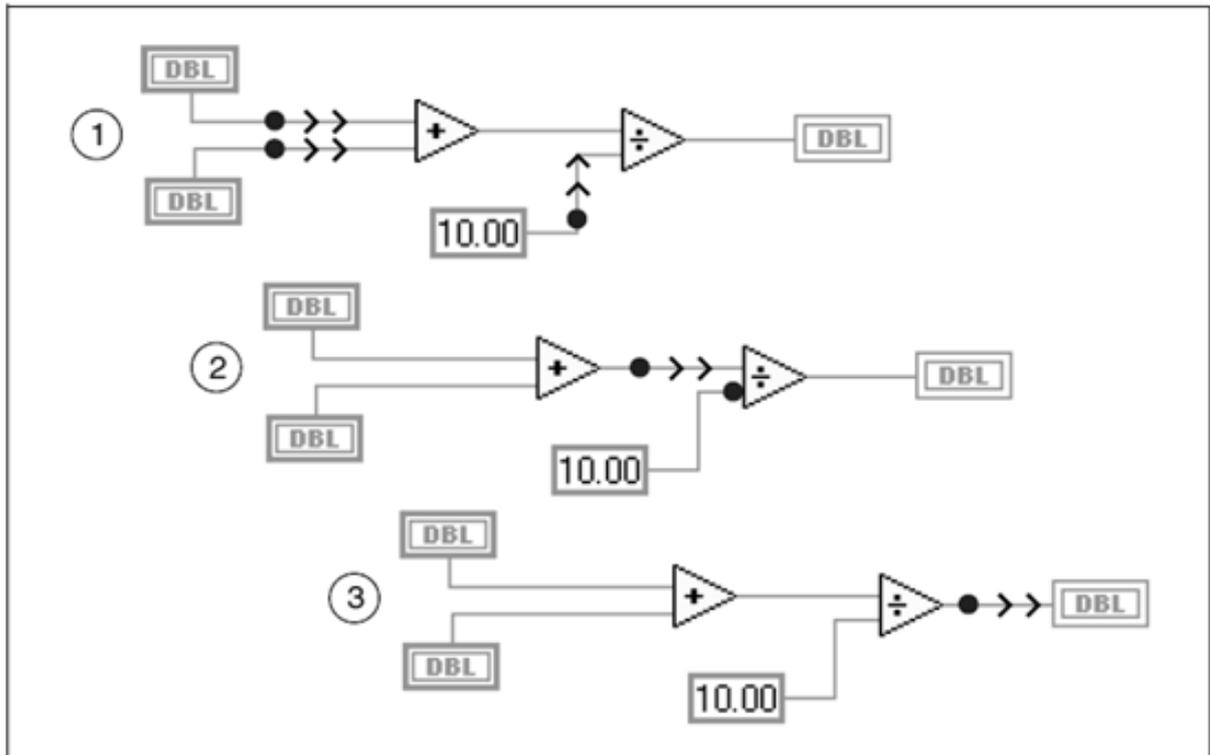


FIG. 6.6 – Le flux de données dans LabVIEW

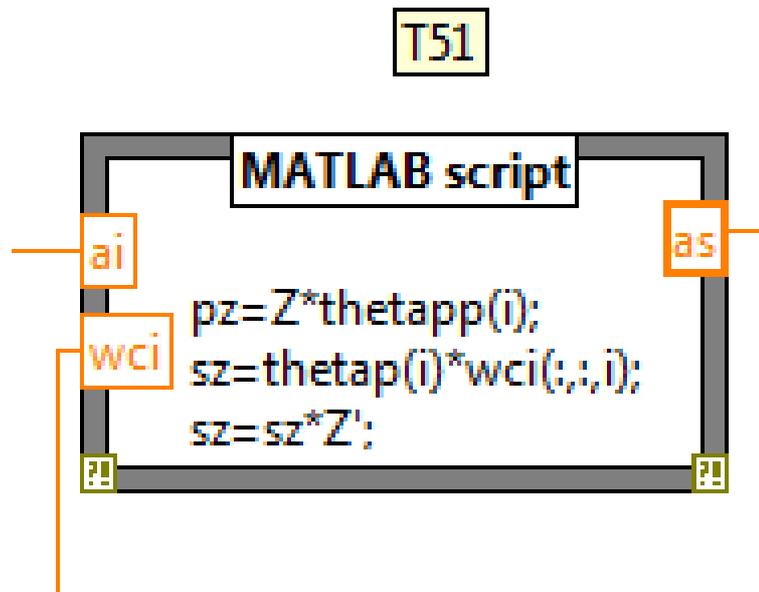


FIG. 6.7 – L'implémentation de la tâche T51 avec le script MATLAB sous LabVIEW

L'algorithme de Newton-Euler est divisé en 3 phases qui doivent s'exécuter séquentiellement, la 1ère phase permet la déclaration et l'initialisation des variables et constantes nécessaires, cette phase est implémentée comme une tâche d'initialisation T00 affectée au processeur maître ou " Host " qui gère les fonctions supplémentaires comme l'affichage et le partage des tâches sur les autres processeurs. Les phases 1 et 2 sont constituées de plusieurs tâches qui doivent s'exécuter n fois en parallèle pour chaque phase .

Sous LabVIEW, l'exécution séquentielle des trois phases est programmée graphiquement avec la structure " Flat Sequence ". Chaque rectangle est une phase séquentielle d'exécution, on peut ajouter des séquences avant ou après à l'aide de " Add Frame After " ou " Add Frame Before".

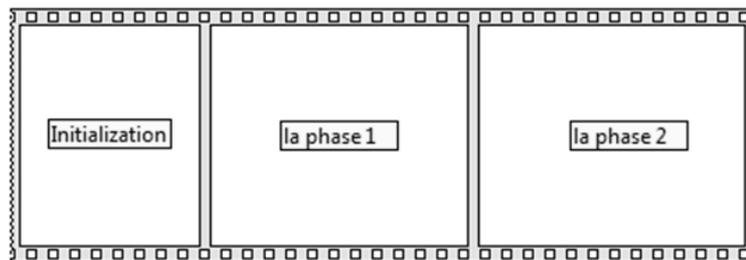


FIG. 6.8 – L'implémentation de l'algorithme NE sous LabVIEW

### 6.6.3 La programmation Multi-coeur avec LabVIEW :

L'approche de programmation graphique NI LabVIEW permet non seulement de gagner du temps, mais elle présente des avantages majeurs pour la programmation des processeurs multi-coeurs et autres matériels parallèles tels que des réseaux pré-diffusés programmables par l'utilisateur (FPGA). Un de ces avantages est la capacité à l'échelle des applications automatiquement au CPU à deux, quatre, ou plusieurs noyaux, souvent sans aucun effort de programmation supplémentaire. En outre, ( graphique code LabVIEW ) permet également aux ingénieurs et scientifiques de visualiser les opérations en parallèle, car il représente naturellement les modèles de programmation parallèle les plus populaires utilisés dans l'industrie et le milieu universitaire.

La programmation parallèle est en train de devenir une nécessité afin de profiter au maxi-

num des processeurs multi-coeurs. Pour de nombreuses raisons, y compris la consommation d'énergie et la bande passante de la mémoire, les processeurs modernes ont des fréquences d'horloge largement stagnantes ; les fabricants de CPU ont plutôt commencé à inclure plusieurs coeurs de processeurs sur une seule puce tout en maintenant, voire en réduisant la fréquence d'horloge. Ainsi, au lieu de s'appuyer sur des hausses de taux d'horloge pour accélérer automatiquement les applications, il serait souhaitable de concevoir des applications pouvant être exécutées sur des processeurs multi-coeurs.

La programmation dans LabVIEW consiste à créer du code graphique (G) qui ressemble à un organigramme, ce qui est nettement différent de celui de la programmation dans les langages traditionnels séquentiels. Ceci est connu comme la programmation par flux de données. Au lieu d'écrire une séquence de commandes qui exécutent un par un, les programmes LabVIEW contiennent des variables et des opérations qui relient une variable à l'autre - le compilateur LabVIEW détermine automatiquement l'ordre des commandes à exécuter pour produire des résultats corrects. Cela signifie également qu'avec LabVIEW, Lorsque deux sections parallèles de code sont indépendants les uns des autres, ils peuvent fonctionner en même temps sur différents coeurs d'un processeur multi-coeurs.

Pour visualiser cela, considérons le programme de la phase 1 pour les taches montrées en figure (6.9). LabVIEW peut reconnaître à la fois que la tache T81 et T11 peuvent s'exécuter en même temps, et comme se sont deux taches indépendantes l'une de l'autre, alors elles sont partagées sur 2 processeurs.

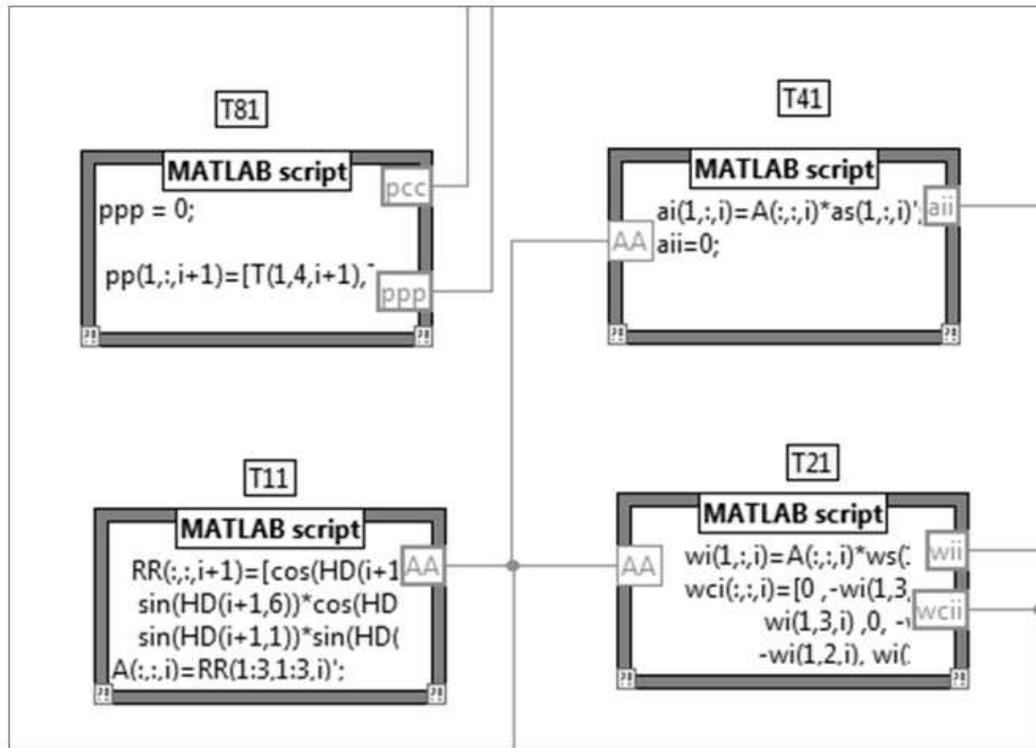


FIG. 6.9 – Les chemins de code indépendants peuvent s'exécuter simultanément

L'implémentation de l'algorithme est divisée en trois phases, la phase d'initialisation représentée par la tâche T00 en figure (6.10); et chacune des deux phases, la phase1 en figure (6.13) et la phase2 en figure (6.14) est implémentée selon la distribution des tâches.

T00

**MATLAB script**

```

global i HD A kk n g to wi wci ws wcs wcc p to acs as ai bb ri rs uc wckw uca kk
n=6;
g=10;
p(1,1:3,1:1:n) = 0;
wi(1,1:3,1:1:n) = 0;
ws(1,1:3,1:1:n+1) = 0;
wci(1:1:3,1:1:3,1:1:n) = 0;
A(1:1:3,1:1:3,1:1:n) = 0;
wcs(1:1:3,1:1:3,1:1:n+1) = 0;
bb(1:1:3,1:1:3,1:1:n) = 0;
to(1,1:3,1:1:n) = 0;
as(1,1:3,1:1:n) = 0;
ai(1,1:3,1:1:n) = 0;
ri(1,1:3,1:1:n) = 0;
rs(1,1:3,1:1:n) = 0;
uc(1:1:3,1:1:3,1:1:n) = 0;
acs(1:1:3,1:1:3,1:1:n) = 0;
wcc(1:1:3,1:1:3,1:1:n+1) = 0;
uca(1:1:3,1) = 0;
ka(1:1:3,1) = 0;
kw(1:1:3,1:1:3,1:1:n) = 0;
kk(1:1:3,1:1:3,1:1:n) = 0;
pF(1:1:3,1) = 0;
ucapf(1:1:3,1) = 0;
Ca(1:1:3,1) = 0;
Cs(1:1:3,1:1:3,1:1:n) = 0;
wckw(1:1:3,1) = 0;
C(1:1:3,1:1:3,1:1:n) = 0;
i = 1;

bx1=0.9*10^-4;ly1=0.9*10^-4;lz1=0.6*10^-4;
bx2=25*10^-4;ly2=25*10^-4;lz2=1.46*10^-4;
bx3=22*10^-4;ly3=22*10^-4;lz3=0.15*10^-4;
bx4=22*10^-4;ly4=22*10^-4;lz4=0.15*10^-4;

```

FIG. 6.10 – La tâche T00 (l'initialisation)

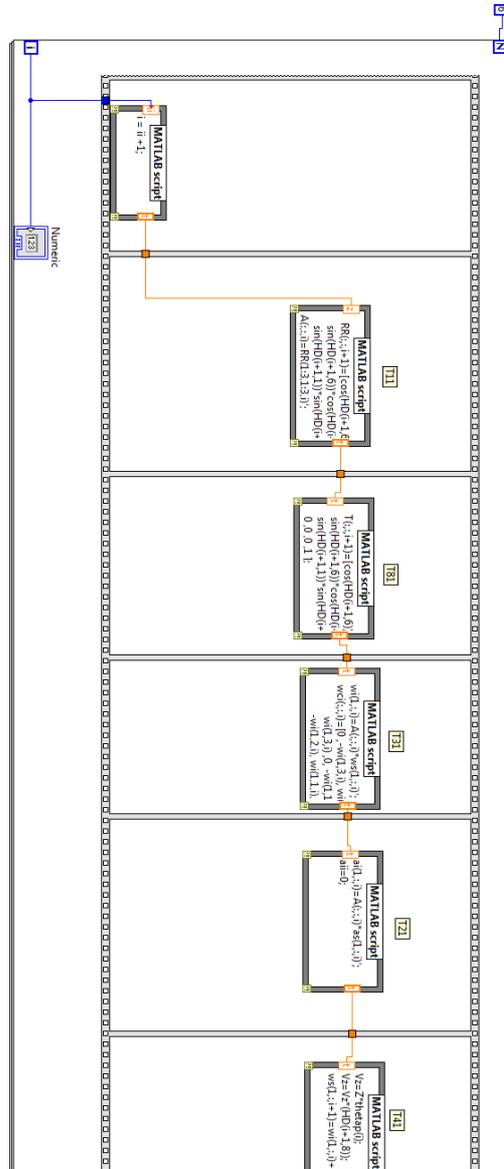


FIG. 6.11 – L'implémentation en séquentiel de la phase 1

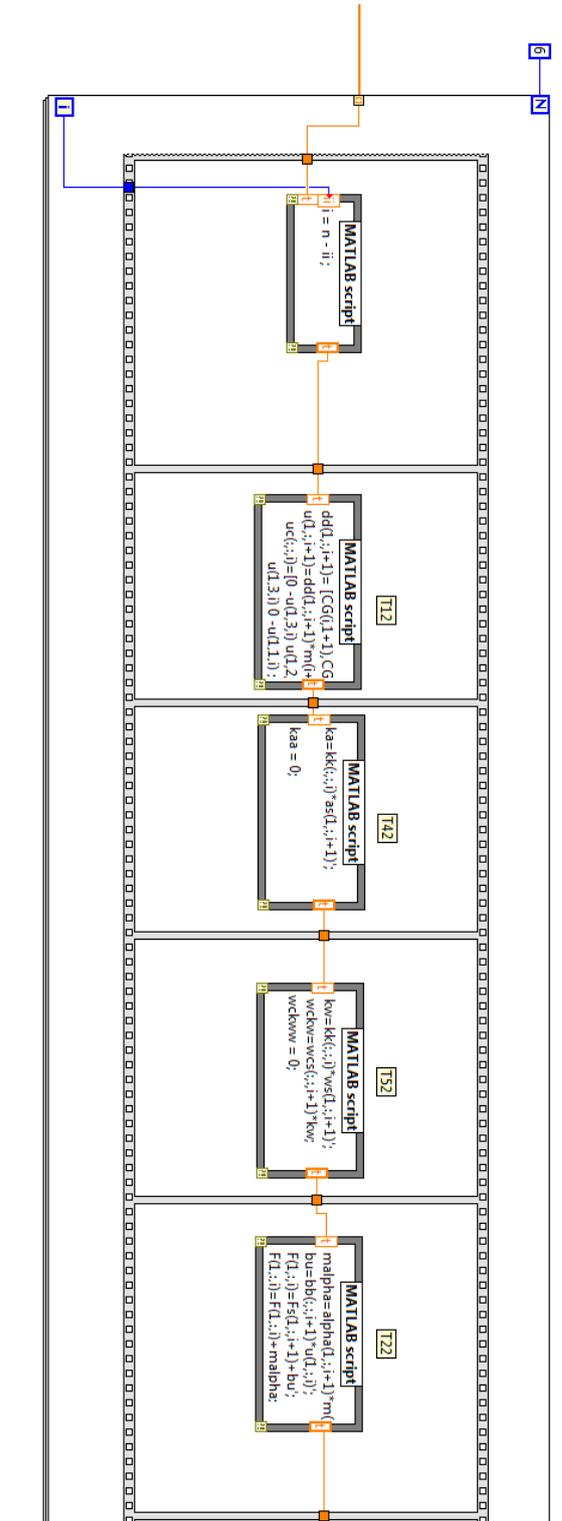


FIG. 6.12 – L'implémentation en séquentiel de la phase 2

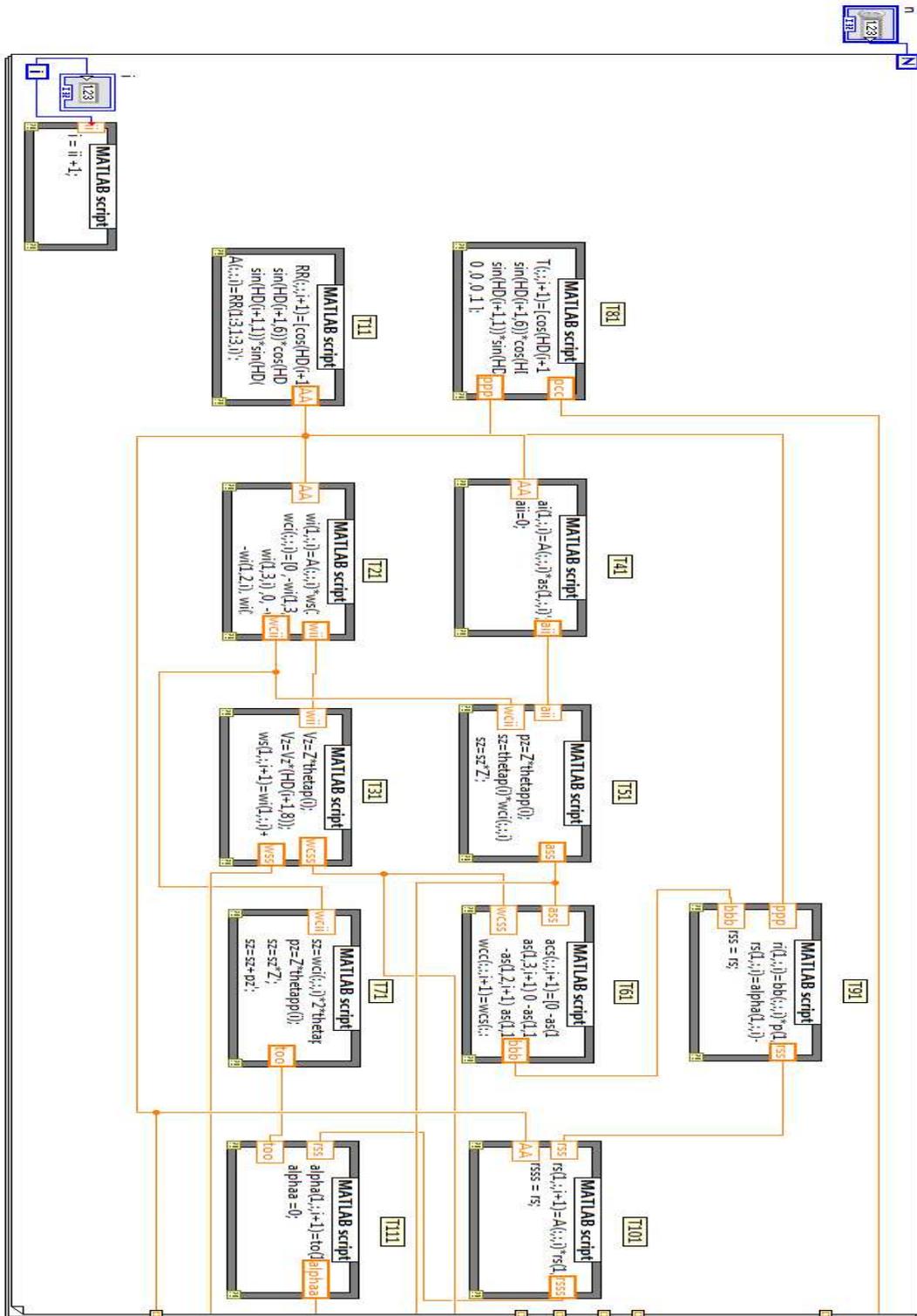


FIG. 6.13 – L'implémentation de la phase 1

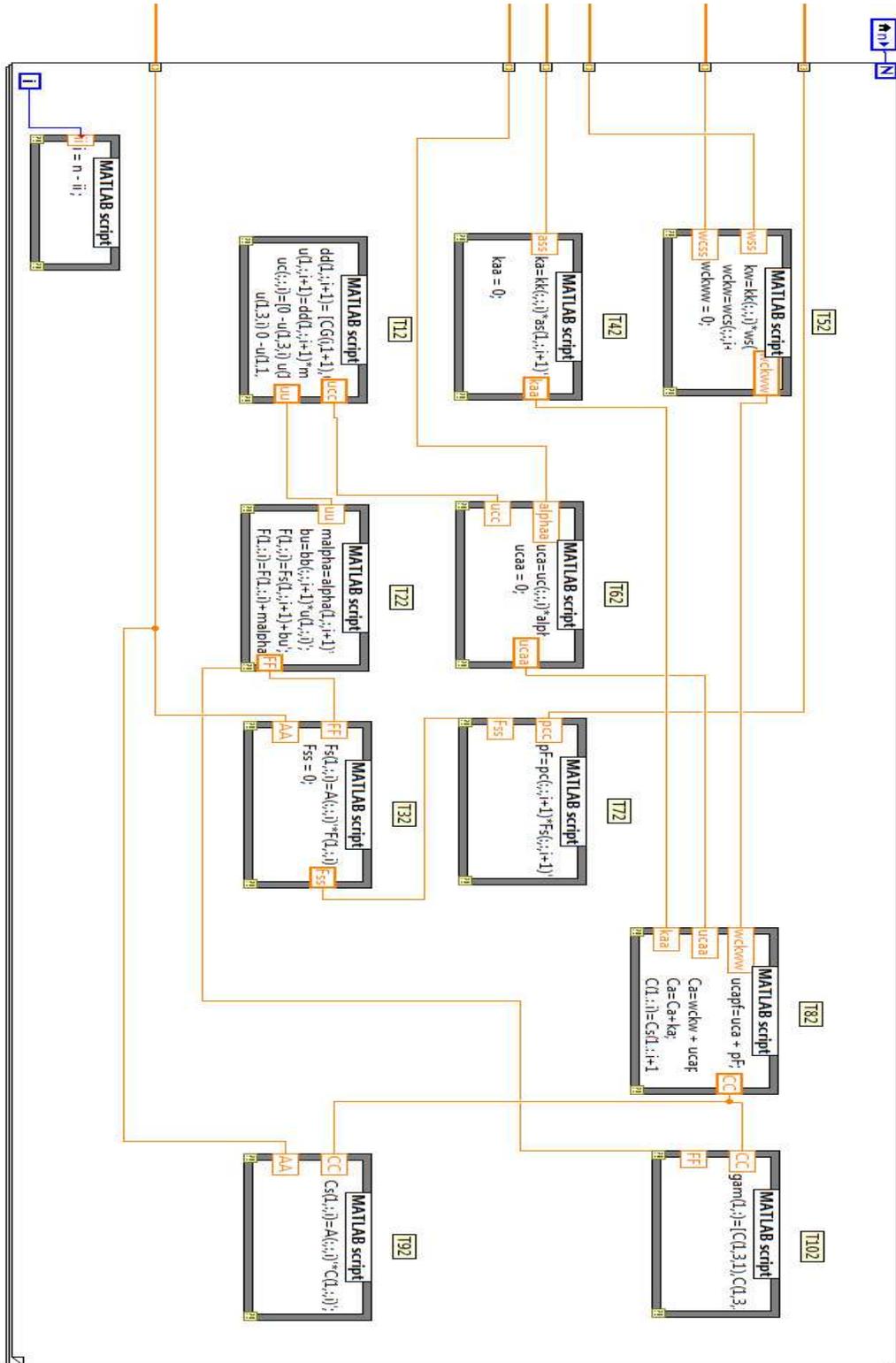


FIG. 6.14 – L'implémentation de la phase 2

## 6.7 Conclusion

L'exploration de l'état de l'art a permis de mettre en évidence une première approche de calcul parallèle en vue d'accélérer les calculs pour les applications temps réel pour les robots manipulateurs. Cette approche est basée essentiellement sur la décomposition explicite des tâches et leur affectation à un ordinateur à architecture multiprocesseurs. En effet, cette approche s'avère très pratique pour le cas de manipulateurs travaillant individuellement ou en coopération. Elle exploite le modèle dynamique bien connu qui est construit autour du formalisme récursif de Newton-Euler. Cette technique pour le calcul séquentiel est d'ordre  $O(N)$  où  $N$  est le nombre de ddl. Elle a également l'avantage que le choix du nombre de processeurs, la répartition des tâches aux différents processeurs ainsi que leur synchronisation et leur coordination sont effectuées et programmées par le concepteur. Du fait que toutes ces tâches sont effectuées par le concepteur, cette méthode est intéressante dans le cas d'un seul manipulateur ou d'un nombre limité de manipulateurs travaillant en coopération. Du point de vue technologique, elle est plus adaptée à une implémentation sur des transputers.

A l'exemple de la référence Zomaya, nous avons simulé cette technique pour un robot à 6 ddl (Robot caumo-polar présentant une liaison prismatique) en choisissant 3 processeurs d'une structure de transputer dont un constitue le host. Nous avons comparé cette simulation à celle d'une simulation séquentielle, un gain de temps de 50% a été obtenu à l'avantage du calcul multiprocesseurs. Cette simulation a été effectuée sous Labview et sous Matlab pour un PCi7 sous windows7 pour lequel la programmation explicite multiprocesseurs a été désactivée.

Par ailleurs, une simulation de l'implémentation multiprocesseurs a été illustrée sous Labview permettant de visualiser le fonctionnement et le parcours du flot de données.

Compte tenu de l'extension du formalisme récursif de Newton Euler pour le cas des manipulateurs en coopération comme présenté par la référence de Orin, il est possible d'étendre aussi la technique du parallélisme par décomposition et affectation des tâches à un système multiprocesseurs. (ceci reste à faire ou au moins à développer les équations du cas de deux robots en coopération).

# Chapitre 7

## Dynamique des robots coopératifs et factorisation matricielle

Après avoir exposé dans le chapitre précédant une première approche s'intéressant aux développements d'algorithmes permettant de construire des modèles dans lesquels on peut effectuer une décomposition de l'algorithme en tâches élémentaires ou le nombre de processeur est choisi par l'utilisateur, nous allons dans ce chapitre aborder plus spécialement la deuxième approche qui exploitent essentiellement la décomposition matricielle des modèles cinématiques (matrice Jacobienne directe et inverse) et dynamiques (matrice d'inertie directe et inverse) permettant l'obtention de systèmes d'équations découplées pouvant être résolues par un réseau de processeurs dont le nombre est imposé par la taille du système.

Considérons un système multi-robots manipulateurs composé de  $k$  robots sériels manipulant un même objet. Ce système constitue une chaîne cinématique fermée lors de la saisie de l'objet manipulé.

L'approche adoptée est basée sur l'approche développée par A.Fijany, K. Djouani et G.Fried [90]. Les différents paramètres et autres notations liés aux segments et aux articulations sont détaillées ci-après.

### 7.0.1 Paramètres liés aux segments et aux articulations

Les différents paramètres liés aux articulations ainsi que les quantités spatiales et globales sont présentés ci-dessous :

- $P_{i,j}$  Vecteur position de  $O_j$  à  $O_i$ .
- $-k$  nombre de segments composant le robot.
- $-N$  : Nombre de degré de liberté de chaque segment.
- $-M$  : Nombre de liaisons de chaque segment.
- $-S_i$  : Vecteur position entre le point  $O_j^i$  et le centre de gravité du corps  $j$ .
- $-\theta_j^i, \dot{\theta}_j^i, \ddot{\theta}_j^i$  : Position, vitesse et accélération de l'articulation  $j$  du segment  $i$ , exprimées dans le repère lié à l'articulation  $j$ .
- $-\omega_j^i, \dot{\omega}_j^i \in \mathfrak{R}^3$  : Vitesse et accélération angulaire du corps  $j$  appartenant au segment  $i$  exprimées dans le repère lié à l'articulation  $j$ .
- $-h_j^i, \dot{v}_j^i \in \mathfrak{R}^3$  : Vitesse et accélération linéaire du corps  $j$  appartenant au segment  $i$  exprimées dans le repère lié à l'articulation  $j$ .
- $-k_j^i, \dot{v}_j^i$  : Premier et deuxième moment d'inertie du corps  $i$  calculés au point  $O_j^i$ , appartenant au segment  $i$  ;
- $-\Gamma_j^i$  : Force appliquée à l'articulation  $j$  du segment  $i$ .
- $-f_j^i, n_j^i \in \mathfrak{R}^3$  : Force et moment d'interaction entre les corps  $j - 1$  et  $j$  du segment  $i$ .
- $-m_i$  : Masse du segment  $i$ .
- $-\mathcal{J}_i$  : Second moment de la masse du segment  $i$  autour de son centre de masse.
- $-c(i, i + 1)$  : Matrice décrivant l'orientation du repère  $i + 1$  par rapport au repère  $i$ .

## 7.0.2 Quantités spatiales

${}^iH_j$  : Axe spatial du corps  $j$  appartenant au segment  $i$ . avec :  ${}^iH_j = \begin{bmatrix} \mathbf{Z} & 0 \\ 0 & \mathbf{X} \end{bmatrix} \in \mathfrak{R}^{6 \times 2}$

pour une articulation à deux degrés de liberté, une rotation autour de l'axe  $Z$  et une translation suivant l'axe  $X$ .

$V_j^i = \begin{bmatrix} \omega_j^i \\ v_j^i \end{bmatrix} \in \mathfrak{R}^6, \dot{V}_j^i = \begin{bmatrix} \dot{\omega}_j^i \\ \dot{v}_j^i \end{bmatrix} \in \mathfrak{R}^6$  : Vitesse et accélération spatiales du corps  $j$  appartenant au segment  $i$  calculées par rapport au repère lié à l'articulation  $j$  du segment

$i$ .

$F_j^i = \begin{bmatrix} n_j^i \\ f_j^i \end{bmatrix} \in \mathfrak{R}^6$  : Forces d'interaction spatiales du corps  $j$  appartenant au segment  $i$ .

$V_{N+1} = \begin{bmatrix} \omega_{N+1} \\ v_{N+1} \end{bmatrix} \in \mathfrak{R}^6$  : Vitesse spatiale de l'organe terminal calculée par rapport au repère absolu.

$F_{N+1}$  : Force externe appliquée à l'organe terminal.

### 7.0.3 Quantités globales (définies pour $j=N$ à 1)

$\mathcal{H}_i = \text{Diag}(H_j^i) \in \mathfrak{R}^{6M \times N}$  : Matrice globale des axes spatiaux du segment  $i$ .

$\mathcal{I}_i = \text{Diag}(I_j^i) \in \mathfrak{R}^{6M \times N}$  : Matrice globale des inerties spatiales du segment  $i$ .

$\dot{k} = \text{Col}(\dot{\theta}_j^i) \in \mathfrak{R}^N$  : Vecteur global des vitesses articulaires du segment  $i$ .

$\ddot{k} = \text{Col}(\ddot{\theta}_j^i) \in \mathfrak{R}^N$  : Vecteur global des accélérations articulaires du segment  $i$ .

$\Gamma = \text{Col}(\Gamma_i) \in \mathfrak{R}^N$  : Vecteur global des forces appliquées aux articulations actives.

$V_i = \text{Col}(V_j^i) \in \mathfrak{R}^{6M}$  : Vecteur global des vitesses spatiales du segment  $i$ .

$\dot{V}_i = \text{Col}(\dot{V}_j^i) \in \mathfrak{R}^{6M}$  : Vecteur global des accélérations spatiales du segment  $i$ .

$F_i = \text{Col}(F_j^i) \in \mathfrak{R}^{6M}$  : Vecteur global des forces d'interaction du segment  $i$ .

$\mathcal{M}_i \in \mathfrak{R}^{N \times N}$  : Matrice d'inertie du segment  $i$ , définie dans l'espace articulaire.

$\mathcal{J} \in \mathfrak{R}^{6N}$  : Matrice Jacobienne cinématique du robot.

## 7.1 Notations générales

Pour chaque vecteur  $V$ , on associe une matrice antisymétrique  $\tilde{V}$  définie par :

$$\tilde{V}^t = -\tilde{V} \text{ et } \tilde{V}_1 V_2 = V_1 \wedge V_2$$

Soit  $\hat{V}$  la matrice associée au vecteur  $V$  par :

$$\hat{V} = \begin{bmatrix} U & \tilde{V} \\ 0 & U \end{bmatrix} \in \mathfrak{R}^{6 \times 6} \text{ et } \hat{V}^t = \begin{bmatrix} U & 0 \\ -\tilde{V} & U \end{bmatrix}$$

$U$  et  $0$  représentent respectivement les matrices identité et nulle de dimensions appropriées.

**Remarque :** Si on note  $\mathbf{P}_{A,B}$  le vecteur position de  $B$  vers  $A$ , la matrice  $\hat{P}_{A,B}$  aura les propriétés suivantes :

$$\hat{P}_{A,B} \hat{P}_{B,C} = \hat{P}_{A,C} \text{ et } \hat{P}_{A,B}^{-1} = \hat{P}_{B,A}$$

On appelle  $I_i$  et  $I_{i,c_i}$  les inerties spatiales du segment  $i$  calculées au point  $O_i$  et à son centre de gravité  $C_i$ . Ces inerties spatiales sont définies par :



alors :

$$\mathcal{V}_i = (\mathcal{P}_i^t)^{-1} \mathcal{H}_i \dot{Q}_i \quad (7.2)$$

Le vecteur vitesse spatial de l'effecteur terminal  $\dot{X}_E = V_{N+1}$ , est obtenu à partir de l'équation (7.2) pour  $i = N + 1$  :

$$V_{N+1}^i (\hat{P}_N^i)^t V_N^i = 0$$

Alors :

$$V_{N+1}^i = (\hat{P}_N^i)^t V_N^i \quad (7.3)$$

Soit :  $\beta_i = \begin{bmatrix} (\hat{P}_N^i)^t & 0 \end{bmatrix}$ , l'équation (7.3) s'écrit alors :

$$V_{N+1}^i = \beta_i \mathcal{V}_i$$

alors , par insertion de l'expression de  $V_i$  à partir de l'équation(7.2), on obtient :

$$V_{N+1}^i = \beta_i (\mathcal{P}_i^t)^{-1} \mathcal{H}_i \dot{Q}_i \quad (7.4)$$

La matrice Jacobienne est définie par la relation liant la vitesse spatiale de l'organe terminal aux vitesses articulaires :

$$V_{N+1}^i = \mathcal{J}_i \dot{Q}_i \quad (7.5)$$

En considérant le résultat obtenu en (7.3), nous pouvons assigner une matrice Jacobienne  $\mathcal{J}_i$  relative à chaque robot  $i$  :

$$\mathcal{J}_i = \beta_i (\mathcal{P}_i^t)^{-1} \mathcal{H}_i \quad (7.6)$$

## 7.3 Formulation dynamique dans l'espace opérationnel

Dans ce paragraphe, nous reprenons brièvement la formulation dynamique dans l'espace opérationnel, [19] [91].

### 7.3.1 Chaînes ouvertes

La dynamique d'un manipulateur  $i$  est décrite, dans l'espace articulaire par :

$$\mathcal{M}_i \ddot{Q}_i + C_i + G_i = \Gamma_i \quad (7.7)$$

où  $G_i(Q_i)$  et  $C_i(\dot{Q}_i, Q_i)$  représentent respectivement les forces de gravité et les forces de Coriolis appliquées au manipulateur  $i$ .

Dans l'espace opérationnel, la dynamique du robot est donnée par :

$$\Lambda_i \ddot{X}_{N+1}^i + C_i + G_i = F_{N+1}^i \quad (7.8)$$

où  $\Lambda_i \in \mathbb{R}^{6 \times 6}$  est la matrice d'inertie du robot  $i$  exprimée dans l'espace opérationnel. Les termes  $C_i$  et  $G_i$  étant les forces de Coriolis et les forces de gravité du robot  $i$ , définies au niveau de l'organe terminal.

Les forces, vitesses et accélérations spatiales sont liées aux forces, vitesses et accélérations articulaires par les relations suivantes :

$$\Gamma_i = \mathcal{J}_i^t F_{N+1}^i \quad (7.9)$$

$$V_{N+1}^i = \mathcal{J}_i \dot{Q}_i \quad (7.10)$$

$$\dot{V}_{N+1}^i = \mathcal{J}_i \ddot{Q}_i + \dot{\mathcal{J}}_i \dot{Q}_i \quad (7.11)$$

L'équation (7.7) peut être exprimée par :

$$\ddot{Q}_i + \mathcal{M}_i^{-1} (C_i + G_i) = \mathcal{M}_i^{-1} \Gamma \quad (7.12)$$

En multipliant l'équation (7.12) par  $\mathcal{J}_i$  (en considérant la matrice  $\mathcal{J}_i$  comme étant non singulière), on obtient finalement :

$$\mathcal{J}_i \ddot{Q}_i + \mathcal{J}_i \mathcal{M}_i^{-1} (C_i + G_i) = \mathcal{J}_i \mathcal{M}_i^{-1} \Gamma_i \quad (7.13)$$

En utilisant les équations ci-dessus et après quelques substitutions, on obtient :

$$\ddot{X}_{N+1} + \mathcal{J}_i \mathcal{M}_i^{-1} (C_i + G_i) - \dot{\mathcal{J}}_i \dot{Q}_i = \mathcal{J}_i \mathcal{M}_i^{-1} \mathcal{J}_i^t F_{N+1}^i$$

soit :

$$F_{N+1}^i = \left( \mathcal{J}_i \mathcal{M}_i^{-1} \mathcal{J}_i^t \right)^{-1} \ddot{X}_{N+1} + \left( \mathcal{J}_i \mathcal{M}_i^{-1} \mathcal{J}_i^t \right)^{-1} \left( \mathcal{J}_i \mathcal{M}_i^{-1} (C_i + G_i) - \dot{\mathcal{J}}_i \dot{Q}_i \right) \quad (7.14)$$

En comparant les équations (7.8) et (7.14), on déduit les relations suivantes :

$$\Lambda_i = \left( \mathcal{J}_i \mathcal{M}_i^{-1} \mathcal{J}_i^t \right)^{-1} \quad (7.15)$$

$$C_i = \Lambda_i \left( \mathcal{J}_i \mathcal{M}_i^{-1} C_i - \dot{\mathcal{J}}_i \dot{Q}_i \right) \quad (7.16)$$

$$G_i = \Lambda_i \mathcal{J}_i \mathcal{M}_i^{-1} G_i \quad (7.17)$$

### 7.3.2 Chaînes fermées

L'équation décrivant le comportement dynamique, dans l'espace articulaire, d'une chaîne cinématique dans un système à boucles fermées, est donnée par :

$$\Gamma_i = \mathcal{M}_i \ddot{Q}_i + C_i + G_i + \mathcal{J}_i^t \Phi_i \quad (7.18)$$

où  $\Phi_i \in \mathfrak{R}^6$  représente la force de contact exercée à l'extrémité du segment  $i$ , et  $\mathcal{M}_i$  la matrice d'inertie du segment  $i$  exprimée dans l'espace articulaire.

On en déduit :

$$\ddot{Q}_i = \mathcal{M}_i^{-1} (\Gamma_i - (C_i + G_i)) - \mathcal{M}_i^{-1} \mathcal{J}_i^t \Phi_i \quad (7.19)$$

### 7.3.3 Charge commune

Le comportement dynamique de la charge commune est donné par :

$$I_C \dot{V}_C = \sum_{i=1}^{i=N} \Phi_i + \mathcal{C}_C + \mathcal{G}_C \quad (7.20)$$

où  $I_C$ ,  $\dot{V}_C$ ,  $\mathcal{C}_C$  et  $\mathcal{G}_C$  représentent respectivement la matrice d'inertie, l'accélération spatiale, les forces de Coriolis et les forces de gravité, relatives à la charge commune.

## 7.4 Calcul des forces de contact

La relation liant la vitesse de la charge commune à la vitesse de l'organe terminal du robot  $i$  est donnée par l'équation suivante :

$$V_C^i = V_C = \left( \hat{P}_C^i \right)^t V_{N+1}^i \quad (7.21)$$

$P_C^i$  représente le vecteur déplacement entre  $O_{N+1}^i$  et C (centre de gravité de la charge commune).

On en déduit par simple dérivation, l'expression de l'accélération :

$$\dot{V}_C = \frac{d}{dt} \left( \hat{P}_C^i \right)^t V_{N+1}^i + \left( \hat{P}_C^i \right)^t \dot{V}_{N+1}^i \quad (7.22)$$

En considérant les équations (7.10) et (7.11), on en déduit une nouvelle expression de l'équation (7.22) :

$$\dot{V}_C = \left( \frac{d}{dt} (\hat{P}_C^i)^t \mathcal{J}_i + (\hat{P}_C^i)^t \dot{\mathcal{J}}_i \right) \dot{Q}_i + (\hat{P}_C^i)^t \mathcal{J}_i \mathcal{M}_i^{-1} (\Gamma_i - (C_i + G_i)) - (\hat{P}_C^i)^t \mathcal{J}_i \mathcal{M}_i^{-1} \mathcal{J}_i^t \Phi_i \quad (7.23)$$

Soit :

$$\dot{V}_C = \left( \frac{d}{dt} (\hat{P}_C^i)^t \mathcal{J}_i \right) \dot{Q}_i + (\hat{P}_C^i)^t \mathcal{J}_i \mathcal{M}_i^{-1} (\Gamma_i - (C_i + G_i)) - (\hat{P}_C^i)^t \mathcal{J}_i \mathcal{M}_i^{-1} \mathcal{J}_i^t \Phi_i \quad (7.24)$$

En considérant l'expression donnée en (7.6) , on peut écrire :

$$(\hat{P}_C^i)^t \mathcal{J}_i = (\hat{P}_C^i)^t \beta_i \mathcal{P}_i^{-t} \mathcal{H}_i \quad (7.25)$$

Soit :

$$\mathcal{J}_i^* = (\hat{P}_C^i)^t \mathcal{J}_i = \beta_i^* \mathcal{P}_i^{-t} \mathcal{H}_i \quad (7.26)$$

où  $\beta_i^* = \left[ \begin{array}{c|ccc} (\hat{P}_N^i \hat{P}_C^i)^t & 0 & \dots & 0 \end{array} \right] \in \mathfrak{R}^{6 \times 6N}$

La matrice  $\mathcal{J}_i^*$  représente en fait une matrice Jacobienne en considérant la charge commune comme étant un corps additionnel du robot  $i$ .

L'équation (7.24) s'écrit alors :

$$\dot{V}_C = \dot{\mathcal{J}}_i^* \dot{Q}_i + \mathcal{J}_i^* \mathcal{M}_i^{-1} (\Gamma_i - (C_i + G_i)) - \mathcal{J}_i^* \mathcal{M}_i^{-1} \mathcal{J}_i^t \Phi_i \quad (7.27)$$

En utilisant les équations (7.20) et (7.27), on obtient l'égalité suivante :

$$I_C^{-1} (G_C + C_C) + I_C^{-1} \sum_{i=1}^{i=N} \Phi_i = \dot{\mathcal{J}}_i^* \dot{Q}_i + \mathcal{J}_i^* \mathcal{M}_i^{-1} (\Gamma_i - (C_i + G_i)) - \mathcal{J}_i^* \mathcal{M}_i^{-1} \mathcal{J}_i^t \Phi_i \quad (7.28)$$

Soit :

$$B_i \oplus = \dot{\mathcal{J}}_i^* \dot{Q}_i + \mathcal{J}_i^* \mathcal{M}_i^{-1} (\Gamma_i - (C_i + G_i)) - I_C^{-1} (G_C + C_C) \quad (7.29)$$

avec  $\oplus = \text{Col}\{\Phi_i\} \in \mathfrak{R}^{6N}$

et  $B_i = \left[ \begin{array}{ccc|ccc} I_C^{-1} & \dots & I_C^{-1} + \mathcal{J}_i^* \mathcal{M}_i^{-1} \mathcal{J}_i^t & I_C^{-1} & \dots & I_C^{-1} \end{array} \right] \in \mathfrak{R}^{6 \times 6N}$

En écrivant l'équation (7.29) pour  $i$  allant de 1 à  $N$ , on obtient une équation globale donnée par :

$$B \oplus = A \quad (7.30)$$

avec  $\oplus = \text{Col}\{\Phi_i\} \in \mathfrak{R}^{6N}$  avec :

$$B = \begin{bmatrix} I_C^{-1} + \mathcal{J}_1^* \mathcal{M}_1^{-1} \mathcal{J}_1^t & I_C^{-1} & \dots & I_C^{-1} \\ I_C^{-1} & I_C^{-1} + \mathcal{J}_2^* \mathcal{M}_2^{-1} \mathcal{J}_2^t & \dots & I_C^{-1} \\ \vdots & & & \vdots \\ I_C^{-1} & \dots & I_C^{-1} + \mathcal{J}_{N-1}^* \mathcal{M}_{N-1}^{-1} \mathcal{J}_{N-1}^t & I_C^{-1} \\ I_C^{-1} & \dots & I_C^{-1} & I_C^{-1} + \mathcal{J}_N^* \mathcal{M}_N^{-1} \mathcal{J}_N^t \end{bmatrix} \in \mathfrak{R}^{6N} \quad (7.31)$$

et :

$$A = \begin{bmatrix} \dot{\mathcal{J}}_1^* \dot{Q}_1 + \mathcal{J}_1^* \mathcal{M}_1^{-1} (\Gamma_1 - (C_1 + G_1)) - I_C^{-1} (G_C + C_C) \\ \vdots \\ \dot{\mathcal{J}}_N^* \dot{Q}_N + \mathcal{J}_N^* \mathcal{M}_N^{-1} (\Gamma_N - (C_N + G_N)) - I_C^{-1} (G_C + C_C) \end{bmatrix} \in \mathfrak{R}^{6N} \quad (7.32)$$

## 7.5 Factorisation des matrices $\mathcal{M}_i$ et $\mathcal{M}_i^{-1}$

Le comportement dynamique d'un segment  $i$  (considéré comme une chaîne ouverte) est décrit dans l'espace articulaire par l'équation (7.18) alors on obtient :

$$\mathcal{M}_i \ddot{Q}_i = \Gamma_i - C_i - G_i \quad (7.33)$$

soit :

$$\ddot{Q}_i = \mathcal{M}_i^{-1} {}^i \mathcal{F}_T \quad (7.34)$$

ou  ${}^i \mathcal{F}_T = \text{Col} \{ {}^i F_{T_j} \} = \Gamma_i - C_i - G_i \in \mathfrak{R}^N$

$\{ {}^i F_{T_j} \}$  représente l'effort de contrôle au niveau de l'articulation  $j$  du segment  $i$ . Les propagations des accélérations et des forces le long d'une chaîne sériele (ici le segment  $i$ ) sont données par (les repères de projection étant parallèles pour  $j=N$  à 1) :

$${}^i \dot{V}_j = {}^i \hat{P}_j^t {}^i \dot{V}_{j-1} + {}^i H_j {}^i \ddot{\theta}_j \quad (7.35)$$

$${}^i F_j = {}^i I_j {}^i \dot{V}_j + {}^i \hat{P}_j {}^i F_{j+1} \quad (7.36)$$

Dans ces équations, on ne tient pas compte des termes non linéaires (ne dépendant pas directement de l'accélération) :

En utilisant la notation globale décrite précédemment, les équations (7.35) et (7.36) peuvent être réécrites par :

$$\mathcal{P}_i^t \dot{\mathcal{V}}_i = \mathcal{H}_i \ddot{Q}_i \quad (7.37)$$

$$\mathcal{P}_i \mathcal{F}_i = \mathcal{I}_i \dot{\mathcal{V}}_i \quad (7.38)$$

## 7.6 Stratégie de décomposition des forces d'interaction entre les corps du segment $i$

La factorisation de  $\mathcal{M}_i$  et de  $\mathcal{M}_i^{-1}$  sont basées sur une décomposition des forces d'interaction entre les différents corps  $j$  composant le segment  $i$ , de la forme :

$${}^iF_j = {}^iH_j {}^iF_{T_j} + {}^iW_j {}^iF_{s_j} \quad (7.39)$$

ou  ${}^iF_{s_j}$  représente la force contrainte.

${}^iW_j$  est une matrice de projection liée à l'articulation  $j$  du segment  $i$ .

Les matrices de projection  ${}^iH_j$  et  ${}^iW_j$  doivent satisfaire les conditions d'orthogonalité suivantes [92]-[93]

$${}^iW_j {}^t H_j = 0 \quad (7.40)$$

$${}^iH_j {}^iH_j {}^t + {}^iW_j {}^iW_j {}^t = U \quad (7.41)$$

$${}^iH_j {}^t {}^iH_j = {}^iW_j {}^t {}^iW_j = U \quad (7.42)$$

ou sous une forme globale :

$$W_i {}^t \mathcal{H}_i = 0 \quad (7.43)$$

$$\mathcal{H}_i \mathcal{H}_i {}^t + \mathcal{W}_i \mathcal{W}_i {}^t = U \quad (7.44)$$

$$\mathcal{H}_i {}^t \mathcal{H}_i = \mathcal{W}_i {}^t \mathcal{W}_i = U \quad (7.45)$$

avec  $\mathcal{W}_i = \text{Diag} \{ {}^iW_j \} \in \mathfrak{R}^{6N \times 5M}$

### Factorisation de $\mathcal{M}_i$

En multipliant l'équation (7.39) par  ${}^iH_j {}^t$ , et en considérant les relations décrites par (7.40) et (7.42), on déduit la relation suivante :

$${}^iF_{T_j} = {}^iH_j {}^t {}^iF_j \quad (7.46)$$

Soit sous une forme globale :

$${}^i\mathcal{F}_T = \mathcal{H}_i {}^t \mathcal{F} \quad (7.47)$$

En considérant les équations (7.47) (7.37) et (7.38) on obtient ainsi :

$${}^i\mathcal{F}_T = \mathcal{H}_i {}^t \mathcal{P}_i^{-1} \mathcal{I}_i \dot{\mathcal{V}}_i = \mathcal{H}_i {}^t \mathcal{P}_i^{-1} \mathcal{I}_i \mathcal{P}_i^{-t} \mathcal{H}_i \ddot{\mathcal{Q}}_i \quad (7.48)$$

Une expression factorisée de  $\mathcal{M}_i$  peut être déduite en comparant les équations (7.34) et (7.48) :

$$\mathcal{M}_i = \mathcal{H}_i^t \mathcal{P}_i^{-1} \mathcal{I}_i \mathcal{P}_i^{-t} \mathcal{H}_i \quad (7.49)$$

### Factorisation de $\mathcal{M}_i^{-1}$

En considérant les équations (7.37) et (7.38) on déduit les relations suivantes :

$$\dot{\mathcal{V}}_i = \mathcal{I}_i^{-1} \mathcal{P}_i \mathcal{F}_i \quad (7.50)$$

et

$$\mathcal{W}_i^t \mathcal{P}_i^t \dot{\mathcal{V}}_i = \mathcal{W}_i^t \mathcal{H}_i \ddot{\mathcal{Q}}_i = 0 \quad (7.51)$$

En insérant l'équation (7.50) dans (7.51) on déduit :

$$\mathcal{W}_i^t \mathcal{P}_i^t \mathcal{I}_i^{-1} \mathcal{P}_i \mathcal{F}_i = 0 \quad (7.52)$$

En substituant (7.39) dans (7.52), on obtient :

$$\mathcal{W}_i^t \mathcal{P}_i^t \mathcal{I}_i^{-1} \mathcal{P}_i \left( \mathcal{H}_i^i \mathcal{F}_T + \mathcal{W}_i^i \mathcal{F}_S \right) = 0 \quad (7.53)$$

Soit :

$$\mathcal{A}_i^i \mathcal{F}_S = -\mathcal{B}_i^i \mathcal{F}_T \quad (7.54)$$

ou  $\mathcal{A}_i = \mathcal{W}_i^t \mathcal{P}_i^t \mathcal{I}_i^{-1} \mathcal{P}_i \mathcal{W}_i \in \mathfrak{R}^{5M \times 5M}$  et  $\mathcal{B}_i = \mathcal{W}_i^t \mathcal{P}_i^t \mathcal{I}_i^{-1} \mathcal{P}_i \mathcal{H}_i \in \mathfrak{R}^{5M \times M}$  sont des matrices blocs triangulaires.

Le vecteur global  $\mathcal{F}_i$  peut être exprimé par :

$$\mathcal{F}_i = \left( \mathcal{H}_i - \mathcal{W}_i \left( \mathcal{W}_i^t \mathcal{P}_i^t \mathcal{I}_i^{-1} \mathcal{P}_i \mathcal{W}_i \right)^{-1} \mathcal{W}_i^t \mathcal{P}_i^t \mathcal{I}_i^{-1} \mathcal{P}_i \mathcal{H}_i \right)^i \mathcal{F}_T \quad (7.55)$$

En substituant l'équation (7.55) dans (7.50), on obtient :

$$\dot{\mathcal{V}}_i = \mathcal{I}_i^{-1} \mathcal{P}_i \left( \mathcal{H}_i - \mathcal{W}_i \left( \mathcal{A}_i \mathcal{W}_i^t \mathcal{P}_i^t \mathcal{I}_i^{-1} \mathcal{P}_i \mathcal{W}_i \right)^{-1} \mathcal{W}_i^t \mathcal{P}_i^t \mathcal{I}_i^{-1} \mathcal{P}_i \mathcal{H}_i \right)^i \mathcal{F}_T \quad (7.56)$$

En multipliant l'équation (7.37) par  $\mathcal{H}_i^t$ , alors  $\ddot{\mathcal{Q}}_i$  peut être calculé par :

$$\ddot{\mathcal{Q}}_i = \mathcal{H}_i^t \mathcal{P}_i^t \dot{\mathcal{V}}_i \quad (7.57)$$

Finalement, en insérant l'équation (7.56) dans (7.57), on trouve :

$$\ddot{\mathcal{Q}}_i = \left( \mathcal{H}_i^t \mathcal{P}_i^t \mathcal{I}_i^{-1} \mathcal{P}_i \mathcal{H}_i - \mathcal{H}_i^t \mathcal{P}_i^t \mathcal{I}_i^{-1} \mathcal{P}_i \mathcal{W}_i \left( \mathcal{W}_i^t \mathcal{P}_i^t \mathcal{I}_i^{-1} \mathcal{P}_i \mathcal{W}_i \right)^{-1} \mathcal{W}_i^t \mathcal{P}_i^t \mathcal{I}_i^{-1} \mathcal{P}_i \mathcal{H}_i \right)^i \mathcal{F}_T \quad (7.58)$$

La factorisation de  $\mathcal{M}_i^{-1}$  est obtenue en comparant les equations (7.58) et (7.34) :

$$\mathcal{M}_i^{-1} = \mathcal{C}_i - \mathcal{B}_i^t \mathcal{A}_i^{-1} \mathcal{B}_i \quad (7.59)$$

ou  $\mathcal{C}_i = \mathcal{H}_i^t \mathcal{P}_i^t \mathcal{I}_i^{-1} \mathcal{P}_i \mathcal{H}_i \in \mathfrak{R}^{M \times M}$  L'expression de  $\mathcal{M}_i^{-1}$  donnée en (7.59) présente une construction mathématique intéressante. En effet, si une matrice  $\mathcal{L}'_i$  est définie par :

$$\mathcal{L}'_i = \begin{bmatrix} \mathcal{A}_i & \mathcal{B}_i \\ \mathcal{B}_i^t & \mathcal{C}_i \end{bmatrix} \in \mathfrak{R}^{6M \times 6M} \quad (7.60)$$

alors,  $\mathcal{M}_i^{-1} = \mathcal{C}_i - \mathcal{B}_i^t \mathcal{A}_i^{-1} \mathcal{B}_i$  est le complément de Schur de  $\mathcal{A}_i$  dans  $\mathcal{L}'_i$  [94]. La structure de cette matrice permet non seulement une interprétation physique, mais aussi une approche relativement simple en vue d'une implémentation parallèle pour le calcul de  $\mathcal{M}_i^{-1}$  [93].

### Application de l'algorithme de factorisation au robot como-polar

#### 1. Algorithme de factorisation de la matrice d'inertie

L'approche proposée concerne la factorisation de la matrice de masse d'un robot sériel à N ddl. On a montré que la matrice masse d'un robot à N ddl s'écrit :

$$\mathcal{J} = \beta \times (\mathcal{P}^t)^{-1} \times \mathcal{H}$$

Nous détaillons ci-dessous l'algorithme proposé pour le calcul de la matrice masse et de la Jacobienne d'un robot 6ddl présentant une liaison prismatique, en utilisant les formes factorisées ci-dessus.

#### (a) Construction des matrices de projection ${}^{i-1}\mathbf{R}_i$

$$\mathbf{R}_i = \begin{pmatrix} C\gamma_j C\theta_j - S\gamma_j C\alpha_j S\theta_j & -C\gamma_j S\theta_j - S\gamma_j C\alpha_j C\theta_j & S\gamma_j C\alpha_j \\ S\gamma_j C\theta_j + C\gamma_j C\alpha_j S\theta_j & -S\gamma_j S\theta_j + C\gamma_j C\alpha_j C\theta_j & -C\gamma_j S\alpha_j \\ S\gamma_j S\theta_j & S\gamma_j C\theta_j & C\alpha_j \end{pmatrix} \quad (7.61)$$

#### (b) Construction des matrices $\hat{P}$

$$\hat{P}_i = \begin{bmatrix} U & \tilde{P}_i \\ 0 & U \end{bmatrix} \quad (7.62)$$

#### (c) Construction des matrices de projection du repère $i-1$ dans $i$ ${}^{i-1}\mathcal{P}_i$

$$C_i = \begin{bmatrix} R_i^t & 0 \\ 0 & R_i^t \end{bmatrix} \quad (7.63)$$

(d) **Construction de  $P_{global}$**

Tenant compte de la convention de projection dans les repères appropriés la matrice  $\mathcal{P}_{global}$  est donnée comme suit :

$$\mathcal{P}_{global} = \begin{bmatrix} U & 0 & 0 & 0 & 0 & 0 & 0 \\ -\hat{\mathcal{P}}_5 \times C_6^t & U & 0 & 0 & 0 & 0 & 0 \\ 0 & -\hat{\mathcal{P}}_4 \times C_5^t & U & 0 & 0 & 0 & 0 \\ 0 & 0 & \hat{\mathcal{P}}_3 \times C_4^t & U & 0 & 0 & 0 \\ 0 & 0 & 0 & \hat{\mathcal{P}}_2 \times C_3^t & U & 0 & 0 \\ 0 & 0 & 0 & 0 & \hat{\mathcal{P}}_1 \times C_2^t & U & 0 \end{bmatrix} \quad (7.64)$$

(e) **Construction de la matrice  $\beta$  projetée dans le repère absolu**

La matrice de projection pour passer de  $R_5$  à  $R_0$  est :

$$M = R_1 R_2 R_3 R_4 R_5$$

$$M_{60} = \begin{bmatrix} M & 0 \\ 0 & M \end{bmatrix} \quad (7.65)$$

Ainsi :

$$\beta = \begin{bmatrix} M_{60} \times P_6^t \cdot 0_{3 \times 3} & 0_{3 \times 3} \end{bmatrix} \quad (7.66)$$

(f) **Construction de  $\mathcal{H}_{global}$**

Comme le robot présente une liaison prismatique au niveau de la troisième liaison, la matrice  $\mathcal{H}_i$  se présente comme suit :

$$H_i = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{et} \quad 0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^t \quad (7.67)$$

Alors :

$$\mathcal{H}_{global} = \begin{bmatrix} \mathcal{H}_6 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathcal{H}_5 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathcal{H}_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathcal{H}_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathcal{H}_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathcal{H}_1 \end{bmatrix} \quad (7.68)$$

(g) La matrice Jacobienne factorisée

$$\mathcal{J} = \beta \times (\mathcal{P}_{global})^t)^{-1} \times \mathcal{H}_{global} \quad (7.69)$$

2. Algorithme de factorisation de la matrice de masse

$$\mathcal{M}^{-1} = \mathcal{C} - \mathcal{B}^t \mathcal{A}^{-1} \mathcal{B}.$$

Avec :

$$\mathcal{A} = \mathcal{W}^t \mathcal{P}^t \mathcal{I}^{-1} \mathcal{P} \mathcal{W}$$

$$\mathcal{A} = \mathcal{W}^t \mathcal{P}^t \mathcal{I}^{-1} \mathcal{P} \mathcal{H}$$

$$\mathcal{C} = \mathcal{H}^t \mathcal{P}^t \mathcal{I}^{-1} \mathcal{P} \mathcal{H}$$

$$\mathbf{R}_i = \begin{pmatrix} C\gamma_j C\theta_j - S\gamma_j C\alpha_j S\theta_j & -C\gamma_j S\theta_j - S\gamma_j C\alpha_j C\theta_j & S\gamma_j C\alpha_j \\ S\gamma_j C\theta_j + C\gamma_j C\alpha_j S\theta_j & -S\gamma_j S\theta_j + C\gamma_j C\alpha_j C\theta_j & -C\gamma_j S\alpha_j \\ S\gamma_j S\theta_j & S\gamma_j C\theta_j & C\alpha_j \end{pmatrix} \quad (7.70)$$

(a) Construction des matrices de projection  ${}^{i-1}\mathbf{P}_i$

$$\mathbf{P}_i = \begin{pmatrix} d_i C\gamma_i + r_i S\gamma_j S\alpha_i \\ d_i S\gamma_i - r_i C\gamma_i S\alpha_i \\ r_i C\alpha_i + b_i \end{pmatrix} \quad (7.71)$$

(b) Construction des matrices  $\tilde{\mathbf{P}}$

$$\tilde{\mathbf{P}}_i = \begin{bmatrix} 0 & -p_{i,3} & p_{i,2} \\ p_{i,3} & 0 & -p_{i,1} \\ -p_{i,2} & p_{i,1} & 0 \end{bmatrix} \quad (7.72)$$

(c) Construction des matrices  $\hat{\mathbf{P}}$

$$\hat{\mathbf{P}}_i = \begin{bmatrix} U & \tilde{\mathbf{P}}_i \\ 0 & U \end{bmatrix} \quad (7.73)$$

(d) Construction des matrices de projection du repère  $i - 1$  dans  $i$

$$C_i = \begin{bmatrix} R_i^t & 0 \\ 0 & R_i^t \end{bmatrix} \quad (7.74)$$

(e) Construction des matrices de projection  ${}^{i-1}P_i$

$$P_i = \begin{pmatrix} d_i C \gamma_i + r_i S \gamma_j S \alpha_i \\ d_i S \gamma_i - r_i C \gamma_i S \alpha_i \\ r_i C \alpha_i + b_i \end{pmatrix} \quad (7.75)$$

(f) Construction des matrices  $\tilde{P}$

$$\tilde{P}_i = \begin{bmatrix} 0 & -p_{i,3} & p_{i,2} \\ p_{i,3} & 0 & -p_{i,1} \\ -p_{i,2} & p_{i,1} & 0 \end{bmatrix} \quad (7.76)$$

(g) Construction des matrices  $\hat{P}$

$$\hat{P}_i = \begin{bmatrix} U & \tilde{P}_i \\ 0 & U \end{bmatrix} \quad (7.77)$$

(h) Construction des matrices de projection du repère  $i - 1$  dans  $i$

$$C_i = \begin{bmatrix} R_i^t & 0 \\ 0 & R_i^t \end{bmatrix} \quad (7.78)$$

(i) Construction de  $P_{global}$

Tenant compte de la convention de projection dans les repères appropriés la matrice  $\mathcal{P}_{global}$  est donnée comme suit :

$$\mathcal{P}_{global} = \begin{bmatrix} U & 0 & 0 & 0 & 0 & 0 \\ -\hat{\mathcal{P}}_5 \times C_6^t & U & 0 & 0 & 0 & 0 \\ 0 & -\hat{\mathcal{P}}_4 \times C_5^t & U & 0 & 0 & 0 \\ 0 & 0 & \hat{\mathcal{P}}_3 \times C_4^t & U & 0 & 0 \\ 0 & 0 & 0 & \hat{\mathcal{P}}_2 \times C_3^t & U & 0 \\ 0 & 0 & 0 & 0 & \hat{\mathcal{P}}_1 \times C_2^t & U \end{bmatrix} \quad (7.79)$$

(j) **Construction de  $\mathcal{H}_{global}$**

Comme le robot présente une liaison prismatique au niveau de la troisième liaison, la matrice  $\mathcal{H}_i$  se présente comme suit :

$$H_i = \begin{bmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \end{bmatrix} \quad \text{et} \quad 0 = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^t \quad (7.80)$$

Alors :

$$\mathcal{H}_{global} = \begin{bmatrix} \mathcal{H}_6 & 0 & 0 & 0 & 0 & 0 \\ 0 & \mathcal{H}_5 & 0 & 0 & 0 & 0 \\ 0 & 0 & \mathcal{H}_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & \mathcal{H}_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & \mathcal{H}_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & \mathcal{H}_1 \end{bmatrix} \quad (7.81)$$

(k) **Projection de  $s$  dans le repère  $i$**

$$s_i = s_i \times R_i$$

(l) **Construction de  $\tilde{s}$**

$$\tilde{s} = \begin{bmatrix} 0 & -s_{i,3} & s_{i,2} \\ s_{i,3} & 0 & -s_{i,1} \\ -s_{i,2} & s_{i,1} & 0 \end{bmatrix} \quad (7.82)$$

(m) **Projection de la matrice  $J_i$**

$$J_i = R_i^t J_i R_i$$

**12) Construction de  $\hat{s}$  et  $\hat{s}^t$**

$$\hat{s} = \begin{bmatrix} U & \tilde{s} \\ 0 & U \end{bmatrix} \quad \text{et} \quad \hat{s}^t = \begin{bmatrix} U & 0 \\ -\tilde{s} & U \end{bmatrix} \quad (7.83)$$

ou  $U$  est la matrice d'identité d'ordre 3

- (n) **Construction de la matrice d'inertie spatiale  $I_{i,C_i}$  autour du centre de masse**

$$I_{i,C_i} = \begin{bmatrix} J_i & 0 \\ 0 & m_i * U \end{bmatrix}$$

**Construction de la matrice d'inertie spatiale  $I$**

$$I = \hat{s}^t I_{i,C_i} \hat{s}^t$$

- (p) **Construction de  $I_{global}$**

$$\mathcal{P}_{global} = \begin{bmatrix} C_6^t \times I_6 C_6 & 0 & 0 & 0 & 0 & 0 \\ 0 & C_5^t \times I_5 C_5 & 0 & 0 & 0 & 0 \\ 0 & 0 & C_4^t \times I_4 C_4 & 0 & 0 & 0 \\ 0 & 0 & 0 & C_3^t \times I_3 C_3 & 0 & 0 \\ 0 & 0 & 0 & 0 & C_2^t \times I_2 C_2 & 0 \\ 0 & 0 & 0 & 0 & 0 & C_1^t \times I_1 C_1 \end{bmatrix} \quad (7.84)$$

- (q) **Construction de  $\mathcal{W}_{global}$**

$$w_i = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (7.85)$$

(Des rotations autour de x et y et des translations selon les axes x, y et z)

$$0_{wi} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix} \quad (7.86)$$

Alors :

$$\mathcal{W}_{global} = \begin{bmatrix} w_i & 0_{wi} & 0_{wi} & 0_{wi} & 0_{wi} & 0_{wi} \\ 0_{wi} & w_i & 0_{wi} & 0_{wi} & 0_{wi} & 0_{wi} \\ 0_{wi} & 0_{wi} & w_i & 0_{wi} & 0_{wi} & 0_{wi} \\ 0_{wi} & 0_{wi} & 0_{wi} & w_i & 0_{wi} & 0_{wi} \\ 0_{wi} & 0_{wi} & 0_{wi} & 0_{wi} & w_i & 0_{wi} \\ 0_{wi} & 0_{wi} & 0_{wi} & 0_{wi} & 0_{wi} & w_i \end{bmatrix} \quad (7.87)$$

(r) Construction des matrices  $\mathcal{A}$ ,  $\mathcal{B}$  et  $\mathcal{C}$

$$\mathcal{A} = \mathcal{W}^t \mathcal{P}^t \mathcal{I}^{-1} \mathcal{P} \mathcal{W}$$

$$\mathcal{B} = \mathcal{W}^t \mathcal{P}^t \mathcal{I}^{-1} \mathcal{P} \mathcal{H}$$

$$\mathcal{C} = \mathcal{H}^t \mathcal{P}^t \mathcal{I}^{-1} \mathcal{P} \mathcal{H}$$

(s) Construction de la matrice inverse  $\mathcal{M}^{-1}$

$$\mathcal{M}^{-1} = \mathcal{C} - \mathcal{B}^t \times \mathcal{A}^{-1} \times \mathcal{B}$$

## 7.7 Détermination et décomposition de la matrice d'inertie d'un système multi-robot coopératif

l'équation de la dynamique du système multi-robots coopératif couplé avec l'objet manipulé [95] (Voir chapitre5) est donnée par :

$$\mathcal{M}_M(x_o) \ddot{x}_o + C_M(x_o, \dot{x}_o) + G_M(x_o) = \mu_M \quad (7.88)$$

Avec :

$$\mathcal{M}_M(x_o) = J_o^T(x_o) J^{-T}(q) \mathcal{M}(q) J^{-1}(q) J_o(x_o) + \mathcal{M}_o(x_o)$$

$$C_M(x_o) = J_o^T(x_o) J^{-T}(q) \mathcal{M}(q) \frac{d}{dt} (J^{-1}(q) J_o(x_o)) + J_o^T(x_o) J^{-T}(q) C(q, \dot{q}) J^{-1}(q) J_o(x_o) + C_o(x_o, \dot{x}_o)$$

$$G_M(x_o) = J_o^T(x_o) J^{-T}(q) G(q) + G_o(x_o)$$

$$\mu_M = J_o^T(x_o) J^{-T}(q) \tau.$$

Dans cette partie on va s'intéresser beaucoup plus au calcul de la matrice d'inertie et à sa décomposition suivant le complément de Schur en vue d'une implémentation au calcul parallèle (Fijany). (Parallélisme par décomposition matricielle).

**Exemple d'application :**

Considérons un objet de masse  $m_o$  susceptible d'être saisi par deux robots manipulateurs comme le montre la figure (7.1).

La matrice globale du système des deux robots couplés avec l'objet est :

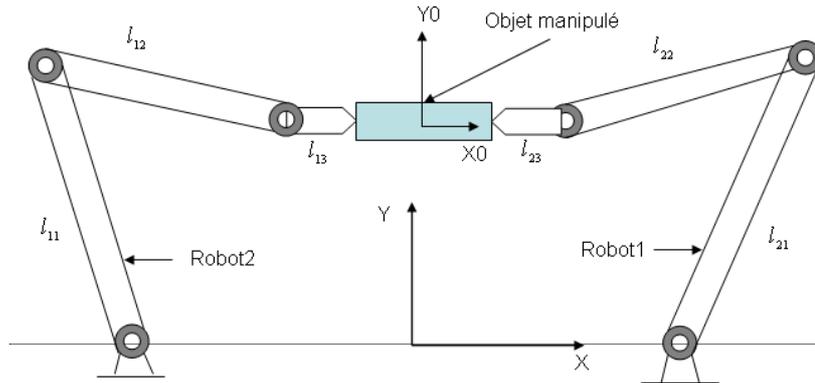


FIG. 7.1 – Deux robots coopératifs dans le plan

$$\mathcal{M}_M(x_o) = J_o^T(x_o)J^{-T}(q)\mathcal{M}(q)J^{-1}(q)J_o(x_o) + \mathcal{M}_o(x_o) \quad (7.89)$$

Les différentes matrices intervenant pour le calcul de la matrice d'inertie globale  $\mathcal{M}_M(x_o)$  dans le cas de deux robots sont :

$$J_o(x_o) = [J_{o1}^T(x_o) \quad J_{o2}^T(x_o)]^T, \quad J(q) = \text{diag}[J_1(q_1), J_2(q_2)], \quad \mathcal{M}(q) = \text{diag}[\mathcal{M}_1(q_1), \mathcal{M}_2(q_2)]$$

et  $\mathcal{M}_o(x_o)$ .

Calculons tous ces termes pour les deux robots :

$$J_{o1}(x_o) = \begin{bmatrix} 1 & 0 & l_{11}\sin(x_{o3}) \\ 0 & 1 & -l_{11}\cos(x_{o3}) \\ 0 & 0 & 1 \end{bmatrix} \quad \text{et} \quad J_{o2}(x_o) = \begin{bmatrix} 1 & 0 & l_{21}\sin(x_{o3}) \\ 0 & 1 & -l_{21}\cos(x_{o3}) \\ 0 & 0 & 1 \end{bmatrix}$$

De ces deux matrices on en déduit alors  $J_o(x_o) = [J_{o1}^T(x_o) \quad J_{o2}^T(x_o)]^T$

**Calcul de  $J_i(q_i)$**

**Pour le robot i on a :**

$$J_i(q_i) = \begin{bmatrix} j_{i11}(q_i) & j_{i12}(q_i) & j_{i13}(q_i) \\ j_{i21}(q_i) & j_{i22}(q_i) & j_{i23}(q_i) \\ j_{i31}(q_i) & j_{i32}(q_i) & j_{i33}(q_i) \end{bmatrix}$$

Les éléments de la matrice  $J_i(q_i)$  pour  $i = 1, 2$  sont respectivement données par :

$$\dot{j}_{i11}(q_i) = -l_{i1}\sin(q_{i1}) - l_{i2}\sin(q_{i1} + q_{i2}) - l_{i3}\sin(q_{i1} + q_{i2} + q_{i3})$$

$$\dot{j}_{i12}(q_i) = -l_{i2}\sin(q_{i1} + q_{i2}) - l_{i3}\sin(q_{i1} + q_{i2} + q_{i3})$$

$$\dot{j}_{i13}(q_i) = -l_{i3}\sin(q_{i1} + q_{i2} + q_{i3})$$

$$\dot{j}_{i21}(q_i) = -l_{i1}\cos(q_{i1}) + l_{i2}\cos(q_{i1} + q_{i2}) + l_{i3}\cos(q_{i1} + q_{i2} + q_{i3})$$

$$\dot{j}_{i22}(q_i) = l_{i2}\cos(q_{i1} + q_{i2}) + l_{i3}\cos(q_{i1} + q_{i2} + q_{i3})$$

$$\dot{j}_{i23}(q_i) = -l_{i3}\cos(q_{i1} + q_{i2} + q_{i3})$$

$$\dot{j}_{i31}(q_i) = \dot{j}_{i32}(q_i) = \dot{j}_{i33}(q_i) = 1$$

ou  $l_{i1}$ ,  $l_{i2}$ ,  $l_{i3}$  sont respectivement les longueurs des segments 1, 2 et 3 du robot  $i$

Une fois déterminée tous les éléments de la matrice Jacobienne  $J_i(q_i)$  on compose alors la matrice Jacobienne globale des deux robots comme suit :

$$J(q) = \text{diag}[J_1(q_1), J_2(q_2)],$$

La matrice d'inertie de l'objet saisi par les deux robots est :

$$\mathcal{M}_o(x_o) = \begin{bmatrix} m_o & 0 & 0 \\ 0 & m_o & 0 \\ 0 & 0 & I_0 \end{bmatrix}$$

ou  $m_o$  et  $I_0$  représente respectivement la masse et le moment d'inertie de l'objet manipulé. Finalement tous les matrices sont calculées on peut alors en déduire la matrice dinertie globale  $\mathcal{M}_M(x_o)$  du système multi-robots copératif comme suit :

$$\mathcal{M}_M(x_o) = J_o^T(x_o)J^{-T}(q)\mathcal{M}(q)J^{-1}(q)J_o(x_o) + \mathcal{M}_o(x_o)$$

La simulation symbolique sous Matlab nous a permis de calculer la matrice  $\mathcal{M}_M(x_o)$  et ensuite sa décomposition suivant le complément de schur ce qui nous a permis d'atteindre l'objectif visé dans cette partie qui est la recherche d'une forme matricielle diagonale souple pour une implémentation parallèle.

# Chapitre 8

## Conclusion générale

Face à la complexité des problèmes tant théoriques que pratiques soulevées par les systèmes de robots manipulateurs travaillant en coopération pour déplacer ou manipuler des objets en commun, l'objectif initial défini pour cette thèse était de chercher à contribuer à l'analyse des approches de modélisation de ces systèmes ainsi qu'à l'analyse de l'accélération du calcul sériel et parallèle pour la simulation et la commande temps réel. Une analyse de l'état de l'art, restreinte aux cas des robots manipulateurs sériels en coopération, a permis de mettre en évidence les deux éléments suivants :

**a)-** Au niveau de la modélisation dynamique, bien que les modèles de Newton-Lagrange et Newton-Euler qui ont été développés pour les systèmes sériels, ils restent adaptable et généralisable aux systèmes de robots manipulateurs coopérants. Cependant, dès que le nombre de robots dépasse 3 robots à 6 ddl, le formalisme mathématique devient difficile à organiser et à exploiter par suite de la taille des expressions symboliques concernant les équations du modèle en question.

**b)-** Pour le contrôle du système robotique en temps réel, il est nécessaire de réduire au minimum les calculs les temps en utilisant des algorithmes dédiés au calcul parallèle et implémentés sur des réseaux de multiprocesseurs (Transputer, DSP, clusters, supercomputers,..). Dans ce contexte, deux approches principales ont été mise en évidence. Une première approche d'utilisation du calcul parallèle consiste à décomposer les tâches du modèle dynamique qui utilise l'algorithme récursif de NE, et à affecter les tâches à un nombre de processeurs appartenant à un calculateur multiprocesseur. Une seconde approche qui cherche à minimiser le temps de calcul en proposant des algorithmes bases sur

des stratégies de décomposition et de manipulation des matrices du modèle dynamique en vue d'obtenir une décomposition ultime permettant la résolution du système d'équations du modèle dynamique. Dans cette seconde approche, la taille du problème détermine le nombre de processeurs ; si le système robotique comporte  $N$  ddl, le réseau de processeurs comportera  $N$  éléments.

Les contributions apportées par ce travail peuvent se résumer dans les points suivants :

1. Etude, et simulation des systèmes de robots coopératifs a savoir la modélisation géométrique, cinématique et dynamique appropriée a ce genre de système , avec prise en compte de la saisie de l'objet.
2. La résolution du problème de distribution des efforts de contact sur les robots manipulateurs sous contraintes de saisie et de non glissement de l'objet manipulé. Les contraintes dégagées ont été intégrées au problème de la distribution optimale d'effort entre les deux bras en minimisant un critère quadratique .
3. La formulations de la dynamique et la commande de coordination globale dans l'espace opérationnelle d'un système multirobots manipulant un objet dans l'espace . Dans ce sens un simulateur sous Matlab a été réalisé ce qui nous a permis de tester le modèle et enfin des simulations ont été illustrées pour verifier l'approche proposée.
4. Exploitation de l'approche du calcul parallèle par décomposition et allocation des taches a un système multiprocesseurs ou le nombre de processeur est choisi par l'utilisateur. Les simulations ont été réalisées sous Matlab et la visualisation de l'implémentation de l'algorithme de décomposition par Labview. Enfin une estimation du temps de calcul multiprocesseurs / monoprocesseurs a été dégagée.
5. Exploitation de l'approche de calcul parallèle par décomposition matricielle et allocation des taches a un système multiprocesseurs ou le nombre de processeur est impose par la taille du probleme.

**Les principales perspectives ouvertes par ce travail sont :**

- l'établissement des critères permettant de générer automatiquement des phases transitoires lors d'une modification éventuelle de l'environnement. En effet, dans l'étude que nous avons présentée, nous considérons que nous avons une connaissance parfaite de l'environnement extérieur et les allures des mouvements sont définies à l'avance.
- la détermination d'autres méthodes de contrôle de la stabilité de l'objet. A ce sujet, l'utilisation de la logique floue est une approche à envisager. En effet, la nature même

des corrections qu'elle autorise fait qu'il devient tout à fait possible d'élaborer des règles comportementales qui peuvent s'exprimer en termes de poussée plus ou moins forte des robots au contact avec l'objet.

- la préhension sur objet mous ou fortement déformables. Dans ce cas, il est nécessaire d'adapter des allures quasi-statiques périodiques ou libres et il serait très utile de disposer des jauges de contraintes sur les effecteurs terminaux.
- la mise au point des différentes tâches à réaliser par un superviseur incorporé à l'objet.
- Ces résultats peuvent être généralisés aux robots à structure complexe, dans l'espace articulaire et opérationnel. D'autant plus que l'efficacité algorithmique de cette méthode augmente avec le nombre de degrés de liberté. D'où l'intérêt de l'application et l'extension de cette méthode pour atteindre une complexité minimum pour les systèmes plus complexes comme le cas des systèmes présentant des structures arborescentes ou des architectures à chaînes fermées. Les robots hexapodes, les robots humanoïdes, les systèmes biomécaniques et les véhicules spatiaux sont quelques exemples d'applications pour l'avenir [96]-[97]-[98].

## Publications dans le cadre de cette thèse

1. **Analysis and Optimization of Grasping Force in the Case of Multi-Robots Cooperation**

A. Khadraoui, C. Mahfoudi, A. Zaatri, K. Djouani

International Review of Mechanical Engineering (I.R.E.M.E.), Vol 7, No1, January 2013

2. **Force distribution for multiple manipulator system grasping an object**

A. Khadraoui, A. Zaatri, C. Mahfoudi, K. Djouani

Revue Andru No8 VOL1-2011 ISSN/1112-7309

### Conférences Internationales

1. **Task Space Control of Grasped Object in the case of Multi-robots Cooperating System**

A. Khadraoui, C. Mahfoudi, A. Zaatri, K. Djouani

The 8th International Mediterranean Latin American Modelling Multiconference September 12-15, 2011 - Ergife Palace Hotel, Rome, Italy

2. **Optimal grasping force distribution for three robots holding a rigid object**

A. Khadraoui, A. Zaatri, C. Mahfoudi, and K. Djouani

7th International Mediterranean and Latin American Modelling Multiconference October 13rd-15th, 2010- Fes (Morocco)

3. **Contribution au Calcul des efforts de contacts dans le cas de la coopération Multi-robots**

A. Khadraoui, A. Chibani, K. Djouani, A. Zaatri Conception et Modélisation des Systèmes Mécanique, CMSM'2011, Sousse-Tunisie, 30 Mai - 01 Juin 2011.

4. **Réalisation et Etude de la dynamique d'un robot Monopode**

A. Khadraoui, C. Mahfoudi, K. Djouani

Conception et Modélisation des Systèmes Mécanique, CMSM'2005, Hammamet Tunisie, 23-25, Mars 2005.

## Conférences nationales

### 1. **Une nouvelle approche de modélisation et simulation des systèmes multi-robots coopératifs**

Khadraoui Ammar, Mahfoudi Chawki, Zaatri Abdelouaheb, Djouani Karim  
journées nationales de robotique, Université de batna, 12-14 Avril 2012

# Bibliographie

- [1] M.Zefran and V.Kumar. Optimal control of systems with unilateral constraints. *Procs. IEEE Int. Conf. on Robotics and Automation*, pages 2695–2700, 1995.
- [2] G.Hirano M.Yamamoto and A. Mohri. Cooperative motion planning for grasp-work type manipulators. *Procs. IEEE Int. Conf. on Robotics and Automation*, pages 940–945, 1998.
- [3] W. Kwon and B.C. Wen. A new optimal force distribution scheme of multiple cooperating robots using dual method. *J. of Intelligent and Robotic Systems*, 21 :301–326, 1998.
- [4] S. G. Tzafestas. Multi arm cooperating robots. *International Series on Microprocessor-Based and Intelligent Systems Engineering Springer*, 30, 2006.
- [5] B.Bounab D.Sidobre and A.Zaatri. On computing multi-finger force closure grasps of 2d objects. *ICINCO Int Conf Informatics in Control Automat Robot*, 2007.
- [6] LEE.S. Dual redundant arm configuration optimization with task-oriented dual arm manipulability. *IEEE Trans. Robotics and Automation*, 5 :78–97, 1989.
- [7] Karim AbdelMalek and Jin-Fan Liu. Robust control of planar dual-arm cooperative manipulators. *Robotics and Computer-Integrated Manufacturing.*, pages 109–120, (2000).
- [8] S. Arimoto F. Miyazaki and S. Kawamura. Cooperative motion control of multiple robot arms or fingers. *IEEE Trans. Robot. Autom.*, (1987).
- [9] DJ. Montana. Contact stability for two-fingered grasps. *IEEE Trans. Robotics and Automation*, 8, 1992.
- [10] Naniwa. T S. Arimoto and K. Wada. Learning and adaptive controls for coordination of multiple manipulators holding a geometrically constrained object. *IEEE International Conference on Intelligent Robots*, 1 :484–494, 1997.

- [11] Jorge Gudino-Lau and Marco A. Arteaga. Dynamic model control and simulation of cooperative robots : A case study mobile robotics moving intelligence. *Jonas Buchli (Ed.) ISBN : 3-86611-284-X*, 2006.
- [12] D. E. Orin and S. Y. Oh. Control of force distribution in robotic mechanisms containing closed kinematic chains. *Trans. of the ASME, J. of Dynamic Systems, Measurement and Control*, 102 :134–141, 1981.
- [13] Cole. A P. Hsu and S. Sastry. Dynamic control of sliding by robot hands for re-grasping. *IEEE International Conference on Robotics and Automation.*, 8 :42–52, 1992.
- [14] Kuc. Lee and B. Park. Learning control of cooperating robot manipulators handling an unknown object. *IEEE International Conference on System*, 3 :3354–3339, 1994.
- [15] Liu G. J. Li and Z. Li. Coordinated manipulation of objects by multifingered robotic hand in contact space and active joint space. *IEEE International Conference on Robotics and Automation.*, 4 :3743–3748, 2002.
- [16] Murray. R Z. Li and S. Sastry. A mathematical introduction to robotic manipulation. *Boca Raton Florida U. S. A CRC Press.*, 1994.
- [17] Yoshikawa. T and X. Zheng. Coordinated dynamic control for multiple robotic mechanisms handling an object. *IEEE/RSJ International Workshop on Intelligent Robots and Systems.*, 4 :315–320, 1991.
- [18] Jankowski K. H. ElMaraghy et W. ElMaraghy. Dynamic coordination of multiple robot arms with flexible joints. *International Journal of Robotics Research.*, 12-6 :505–528, 1993.
- [19] Caccavale F. et Uchiyama. Handbook of robotics chapter 29. *Springer*, pages 701–718, 2008.
- [20] Bruno Vilhena Adorno. Two-arm manipulation : From manipulators to enhanced human-robot collaboration. *Thèse de Doctorat de L'académie de Montpellier Université Montpellier II*, 2 octobre 2011.
- [21] khatib.o. Object manipulation in a multi-effector robot system. *In Proceedings of the 4th international symposium on Robotics Research. Cambridge MA USA MIT Press*, pages 137–144, 1988.

- [22] KHATIB.O. An unified approach for motion and force control of robot manipulator. *IEEE J Robotics and Automation*, 1987.
- [23] Williams.D and Khatib.O. The virtual linkage : a model for internal forces in multigrasp manipulation. *In Proc.IEEE International Conference on Robotics and Automation*, 18(7) :1025–1030, 1993.
- [24] Uchiyama.M and Dauchez.P. A symmetric hybrid position/force control scheme for the coordination of two robots. *In Proceedings of 1988 IEEE International Conference on Robotics and Automation IEEE Comput. Soc. Press.*, pages 350–356, 1988.
- [25] Yamano. M Kim J. Konno and Uchiyama . M. Cooperative control of a 3d dual-flexible-arm robot. *Journal of Intelligent and Robotic Systems*, 39 :1–15, 2004.
- [26] Caccavale. F. Chiacchio. P et Chiaverini. Task-space regulation of cooperative manipulators. *Automatica*, 36-6 :879–887, 2000.
- [27] Chiacchio P. Chiaverini S. et Siciliano. Direct and inverse kinematics for coordinated motion tasks of a two-manipulator system. *J. Dyn. Sys. Meas. Control*, 118 :691–697, 1996.
- [28] Connolly T. H and Pfeiffer. F. Cooperating manipulator control using dual quaternion coordinates. *In Proceedings of the 33rd IEEE Conference on Decision and Control Lake Buena Vista FL USA.*, 3 :2417–2418, 1994.
- [29] Dooley. J McCarthy. On the geometric analysis of optimum trajectories for cooperating robots using dual quaternion coordinates. *In Proceedings IEEE International Conference on Robotics and Automation IEEE Comput. Soc. Press.*, pages 1031–1036, 1993.
- [30] Adorno Bruno Vilhena Philippe Fraisse and Sebastien Druon. Dual position control strategies using the cooperative dual task-space framework. *IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2 :3955–3960, 2010.
- [31] Dodds.GI Cao.D and Irwin.GW. Practical approach to near time optimal inspection-task-sequence planning for two cooperative industrial robot arms. *Int J Robotics Res*, 3 :2179 –2184, 1998.
- [32] Ahmadabadi and al. Unified distributed cooperation strategy for multiple objects handling robots. *Proceeding of the 1998 IEEE Int. Conf on Robotics and Automation*, 10 :3625–30, 1998.

- [33] B.Bounab D.Sidobre and A.Zaatri. Central axis approach for computing n-finger force-closure grasp. *in Proc IEEE Int Conf Robot Automat*, pages 1169–1174, (2008).
- [34] N.Saito T.Satoh and H.Okano. Grasping force control in consideration of translational and rotational slippage of an object by a flexible contact sensor. *Electrical Engineering in Japan*, 165 :51–57, 2008.
- [35] Nahon.MA.and Angeles.J. Optimization of dynamic forces in mechanical hands. *J. Mech. Trans. Autom. Design*, pages 167–173, 1991.
- [36] Gu X. Feng C. Liu W. Huang Y. Jiang X. An G. and Liu J. Lu G. Dual arm coordination : model analysis and dynamic compensation control method. *Control Theory Appl*, 11 :168–176, 1994.
- [37] Schneider SA and Cannon RH. Object impedance control for cooperative manipulation : theory and experiment results. *IEEE 1989 Int. Conf on Robotics and Automat.*, 1989.
- [38] Nakamura and Ghoudoussi M. Dynamics computation of closed link robot mechanisms with nonredundant and redundant actuators. *IEEE Trans Robotics Automat*, 5(3) :292–302., 1989.
- [39] Polen.SR and Park.FC. Lee group formulation of the dynamics of cooperating robot systems. *Robotics and autonomous systems Amsterdam Netherlands Elsevier*, 21 :279, 1997.
- [40] Gu X. Feng C. Liu W. Huang Y. Jiang X. An G. and Liu J. Lu G. Dual arm coordination : model analysis and dynamic compensation control method. *Control Theory Appl.*, 11(2) :168–176, 1994.
- [41] Y.Hu and A.Goldenberg. Dynamic control of multiple coordinated redundant robots. *IEEE Trans. on System, Man, and Cybernetics*, 22 :568–574, 1992.
- [42] Hu YR.Goldenberg. A. adaptive approach to motion and force control of multiple coordinated robots. *ASEM J Dyn Systems MEAS Control*, 115 :6–9, 1993.
- [43] N.Hogan. On the stability of manipulators performing contact tasks. *IEEE Trans On Robotics and Automat*, 4 :677–686, 1988.
- [44] Yale Gary E. Agrawal and Brij .N. Lyapunov controller for cooperative space manipulators. *Journal of Guidance, Control, and Dynamics*, 21 :477–484, 1998.

- [45] Bonitz RG. and Hsia TC. Robust dual arm manipulation of rigid objects via palm grasping theory and experiments. *Proceeding of the 1996 IEEE 13th International conference on Robotics and Automat*, pages 3047–3053., 1996.
- [46] S. Mukherjee and K. J. Waldron. An exact optimization of interaction forces in three-fingered manipulation. *Trans. of the ASME, J. of Mechanical Design*, 114 :48–54, 1992.
- [47] Xydas. Study of soft finger contact mechanics using finite element analysis and experiments. *Proc. of the IEEE Intl. cong on robotics and Automation*, 3 :2179–2184, 2000.
- [48] Hirai.S Tsuboi.T and Wada .T. Robust grasping manipulation of deformable objects. *IEEE Symposium on Assembly and Task Planning.Japan*, pages 2179 –2184, 2001.
- [49] M.R. Cutkosky. On grasp choice, grasp models, and the design of hands for manufacturing tasks. *Trans Robot. Automat*, 5 :269–279, 1989.
- [50] Mason and J.K. Salisbury. Robot hands and the mechanics of manipulation. *Cambridge, MA : MIT Press*, 5, 1985.
- [51] J.Kerr and B. Roth. Analysis of multifingered hands. *International Journal of Robotics Research*, 4 :3–17, 1986.
- [52] D.W. Meer and S.M. Rock. Experiments in object impedance control for flexible objects. *Proceedings International Conference on Robotics and Automation*, 4 :1222–1227, 1994.
- [53] S. Tricamo R. Patton, F. Swern and A. van der Veen. Automated cloth handling using adaptive force feedback. *Journal of Dynamic Systems and Measurements and Control*, 114 :731–733, 1992.
- [54] A. M. Howard and G.A. Bekey. Intelligent learning for deformable object manipulation. *IEEE Transactions on Robotics and Automation*, 165 :15–20, 1999.
- [55] Sciavicco L. and Siciliano B. Modeling and control of robot manipulators. *McGraw-Hill New York*, 1996.
- [56] Angeles.J. Rational kinematics. *Springer-Verlag New York*, 4 :315–320, 1997.
- [57] Subir Kumar Saha. Simulation of industrial manipulators based on the  $udu^T$  decomposition of inertia matrix. *Multibody System Dynamics Kluwer Academic Publishers. Printed in the Netherlands.*, 9 :63–85, 2003.

- [58] Golub G.H. and Van Loan C.F. *Matrix Computations*. John Hopkins University Press Baltimore MD 1983.
- [59] P. Schafer and W. Schiehlen. Application of parallel computing to robot dynamics. *Proceeding of the IMAACS SICE linternational Symposium on Robotics Mechatronics and Manufacturing Sytems92 Kobe Japan Sep 416-20*, pages 599–606, 1992.
- [60] Dominik Henrich Joachim Karl und Heinz WÖRN. A review of parallel processing approaches to robot kinematics and jacobian. *Technical Report 10/97 University of Karlsruhe Computer Science Department*, pages 1432–7864, 1997.
- [61] R.J. Schilling. Fundamentals of robotics : Analysis and control. *Prentice-Hall Englewood Cliffs NJ*, pages 195–236, 1990.
- [62] Albert Y Zomaya. Practical aspects and experiences parallel processing for robot dynamics computations. *Elsevier Science B.V. Parallel Computing*, pages 649–668, 1995.
- [63] J.Barhen. Hypercube ensembles : An architecture for intelligent robots. in *Computer Architectures for Robotics and Automation*, ed. J.H. Graham Gordon and Breach Science Publishers New York, pages 195–236, 1987.
- [64] Albert Y Zomaya. Modelling and simulation of robot manipulators : A parallel processing approach. *World Scientific Publishing, New Jersey*, 1992.
- [65] Featherstone R. A divide-and-conquer articulated-body algorithm for parallel  $o(\log(n))$  calculation of rigid-body dynamics. part.1 : Basic algorithm. *Int. J. Robotics Res*, 18 No9, 1999.
- [66] Featherstone R. and A.Fijany. A technique for analyzing constrained rigid body systems, and its application to the constraint force algorithm. *IEEE Trans. on Robotics and Automation*, 15 No6, 1999.
- [67] Albert Y Zomaya. Efficient robot dynamics for high sampling rate motions : Case studies and benchmarks. *Int. J. Control*, pages 793–814, 1991.
- [68] A. Fijany and A. Bejczy. Parallel computation systems for robotics : Algorithms and architectures. eds *World Scientific Publishing Singapore*, 1992.
- [69] W. Khalil and E. Dombre. *Modélisation identification et commande des robots*. Hermes sciences, Paris, 1999.

- [70] Khalil and J.F. Kleinfinger. Une modelisation performante pour la commande dynamique des robots. *RAIRO-APII*, 19 No6, 1985.
- [71] R.Paul. *Robots manipulators mathematics programing and control*. MIT press, 1981.
- [72] F. Ouezdou. Outils d'aide à la conception de robots à locomotion articulée, thèse de doctorat. *Université de Piere et Marie Curie, France*, 1980.
- [73] G. Gosselin J. Angeeles. *Kinematics of parallel manipulators*. MCGill University, Monreal, Québec, Canada, Sep 1988.
- [74] C. A. Klein and S. Kittivatcharapong. Optimal force distribution for the legs of a walking machine with friction cone constraints. *IEEE Trans. on Robotics and Automation*, 6 :73–85, 1990.
- [75] F. T.Cheng and D. E. Orin. Efficient formulation of the force distribution equations for simple closed-chain. *IEEE Trans. on Systems, Man, and Cybernetics*, 21 :25–32, 1991.
- [76] M. A. Nahon and J. Angeles. Real-time force optimization in parallel kinematic chains under inequality constraints. *IEEE Trans. on Robotics and Automa.*, 12 :87–94, 1999.
- [77] X. Chen and K.Watanabe. Optimal force distribution for the legs of quadruped robot. *Machine inteligenca and robotique control*, 1 :87–94, 1999.
- [78] C. Mahfoudi K Djouani S. Rechak and M. Bouaziz. Optimal force distribution for the legs of an hexapod robot. In *CCA 2003. Proceedings of 2003 IEEE Conference on Control Applications*, volume 1, pages 657–663, Istanbul, Turkey, June 2003.
- [79] C. Mahfoudi K. Djouani S. Rechak and M.Bouaziz. Optimal force distribution for the legs of an hexapod robot. *IEEE Conference on Control Application CCA Istambul*, 1 :651–670, 2003.
- [80] F. T. Cheng and D. E. Orin. Efficient algorithm for optimal force distribution-the compact-dual lp method. *IEEE Trans. on Robotics and Automation*, 6 :178–187, 1990.
- [81] H. Liu and B. Wen. Force distribution for the legs of quadruped robot. *Journal of Robotique Systemes*, 14 :1–8, 1997.
- [82] W. Khalil and J.F. Kleinfinger. Une modelisation performante pour la commande dynamique des robots. *RAIRO-APII*, 19,No6, 1985.

- [83] M. Renaud. Iterative analytique computation of the dynamic model of a robot manipulator. *Rapport de recherche LAAS. No. 86014, Toulouse, France*, 1986.
- [84] F. Pfeiffer J. Eltze and H. Weidman. Six- legged technical walking considering biological principles. *Robotics and Autonomous Systems*, 14 :223–232, 1995.
- [85] M. J. Randall. *Stable adaptive neural control of systems with closed kinematic chains applied to biologically-inspired walking robots*. PhD thesis, University of the West of England, Bristol, 1999.
- [86] J. Boissonat and B. Faverjon. *Technique de la robotique, Architectures et commandes*. Hermes sciences, paris, 1988.
- [87] J-H Jean and L-C Fu. An adaptive control scheme for coordinated multimanipulators systems. *IEEE Trans. Robotics and Automation*, 9 :226–231, 1993.
- [88] G.Song and L.Cai. A smooth robust control approach to cooperation of multiple manipulators. *in Proc. American Control Conf Seattle WA*, 2 :1382–1386, 1995.
- [89] Hu T.C. Combinatorial algorithms. *Addison Wesley Reading MA*, 1982.
- [90] A.Fijany K.Djouani Y.Hamam and Pontnau. New serial and parallel algorithm for robot manipulator modeling and control. *ECPD Conference.Bremen*, 1997.
- [91] KHATIB.O. Inertia properties in robotic manipulation : an object level framework. *IEEE J. Robotics and Automation*, 1995.
- [92] A. Fijany Sharf I. and Eleuterio G. M. T. Parallel  $o(\log n)$  algorithms for computation of manipulator forward dynamics. *IEEE Transaction on Robotics and Automation*, 11 No3, 1995.
- [93] A.Fijany. Parallel  $o(\log n)$  algorithms for rigid multi-body dynamics. *JPL Engineering Memorandum, EM 343-92-1258*, Aout 1992.
- [94] R.W. Cottle. Manifestation of schur complement. *Lineair Algebra and its Application*, Vol 8, 1974.
- [95] S.S.GE TH.LEE and C.J HARRIS. *Adaptive Neural Network Control of Robotic Manipulators*. World Scientific. Singapore. New Jersey. London. Honkong.
- [96] Featherstone R. Efficient factorization of the joint space inertia matrix for branched kinematic trees. *Int. J. Robotics Research*, 24 no. 6, 2005.

- [97] Featherstone.R. An empirical study of the joint space inertia matrix. *Int. J. Robotics Research*, pages 859–871, 2004.
- [98] Featherstone.R. Rigid body dynamics algorithm for parallel computers. *ERCIM News*, 42 :859–871, 2000.