Thèse de Doctorat

Philip LONG

Mémoire présenté en vue de l'obtention du
grade de Docteur de l'École centrale de Nantes
sous le label de l'Université de Nantes Angers Le Mans

École doctorale : Sciences et technologies de l'information, et mathématiques

Discipline : Automatique, Productique
Unité de recherche : Institut de Recherche de Commuinications et Cybernétique de Nantes

Soutenue le 15 October 2014

Contributions to the Modeling and Control of Cooperative Manipulators

JURY

Présidente :	**Mme Véronique PERDEREAU**, Professeur, Université Pierre et Marie CURIE, Paris, ISIR
Rapporteurs :	**M. Grigore GOGU**, Professeur, Insitut Pascal, Clermont-Ferrand, IFMA
	M. Philippe POIGNET, Professeur, Université Montpellier 2, LIRMM
Directeur de thèse :	**M. Wisama KHALIL**, Professeur, École Centrale de Nantes, IRCCyN
Co-directeur de thèse :	**M. Philippe MARTINET**, Professeur, École Centrale de Nantes, IRCCyN

Abstract

Cooperative manipulation strategies are essential as the domain of robot applications is increased. By using two or more robots, a greater range of tasks can be accomplished. This thesis treats several cases of cooperative manipulation, from the manipulation of rigid objects to the separation of deformable materials. The contributions of this thesis are threefold.

Firstly, a study of a lower mobility cooperative system grasping a rigid object is undertaken. A kinematic and dynamic analysis is carried out to obtain the mobility, singular configurations and optimum actuation scheme of the system.

Secondly, a general dynamic model of a closed chain robot with flexibility is derived. The analysis focuses on cooperative robots with flexible objects. The object is modeled using distributed flexibility and a closed form relation is derived for the dynamic model. This method is applied to the Gough Stewart manipulator with flexible platform and the dynamic model is obtained.

Finally, the separation of deformable bodies using multiple robots is investigated. A simulator is created where a multi-arm meat cutting system is modeled. Force/Vision control schemes are proposed that allow the system to adapt to on-line deformations of the target object. An experimental validation is carried out that shows the how the resistive cutting force can be used by the controller to avoid globally deforming the object.

Key words: Cooperative Manipulators, Flexible Object Manipulation, Multi-arm system, Force/Vision Control, Robotic Cutting, Flexible Robots.

Résumé

L'utilisation de robots coopératifs deviendra essentielle dans différents d'applications. En employant deux robots, une plus large gamme de tâches peut être réalisée. Cette thèse focalise sur la manipulation coopérative. Elle contient trois contributions principales.

La première concerne l'étude analytique d'un système coopératif de basse mobilité qui tient un objet rigide. Les études cinématiques et dynamiques permettent d'obtenir la mobilité, les singularités et le meilleur choix d'ensemble d'actionneurs.

La seconde porte sur la modèle dynamique d'un manipulateur coopératif souple. L'analyse focalise sur les robots coopératifs avec des objets flexible. L'objet est modélisé par les fonctions de formes et une solution de forme fermée est dérivée. On exploite cette méthode pour obtenir le modèle dynamique d'un robot parallèle, le Gough Stewart robot.

La dernière concerne la séparation d'objets mous par plusieurs robots. La construction d'un simulateur d'un système multi-bras pour la découpe de viande est décrite. Une commande par vision/effort est développée qui permet le système de s'adapter d'en fonction de l'état de l'objet. Des expérimentations sont effectuées et montrent comment, pendent la découpe, l'effort qui est généré par la résistance de l'objet peut servir pour éviter les déformations globales d'objets.

Mot clés: Robots Coopératifs, Manipulation d'objets mous, Commande par vision et effort, Découpe Robotique, Robots souples.

Contents

List of Figures

List of Tables

Acknowledgements

The research described in this thesis was carried out from October 2011 to October 2014 at the Institut de Recherche en Communications et Cybernétique de Nantes (IRCCyN). The thesis was financially supported by the French National Research Agency (**ANR**) through the ARMS project (ANR-10-SEGI-0008).

First of all I would like to thank my supervisors Professor Philippe Martinet and Professor Wisama Khalil for the tremendous help they gave me throughout this work. It was a privilege to carry out this thesis under your supervision and I am truly grateful for all the time we spent together discussing my research. I cannot overstate the positive influence you have had on me and in particular I would like to thank you both for all your advice, guidance and encouragement during the three years of my thesis.

I would like to thank the members of the jury for having accepted to assess my thesis. Firstly, the reporters, Professor Grigore Gogu and Professor Philippe Poignet, for the time spent writing the report on the thesis and all the interesting comments that followed. Secondly, Professor Véronique Perdereau for accepting to be a member of the jury and for her interest in my work.

I would like to thank all my fellow PhD students of IRCCyN in Nantes for their friendship and for their company during all those lunch breaks together. I would like to thank the technical personal who were always available to help me with software and hardware problems and those who manufactured the tools required for my experiments. In particular, in IRCCyN, I would like to thank Dr. Amine Abou Moughlbay for all the useful technical discussions we've had of course for the friendship we shared. I am also grateful to Dr. Stéphane Caro, who not only aided me enormously with my thesis but whose kindness and friendship since I arrived in Nantes is very much appreciated. In addition to this I am grateful to those in Institut Pascal in Clermont Ferrand for welcoming me and helping me during the course of the ARMS project, especially Erol Ozgur with whom I had several interesting discussions.

I am lucky to have close friends not only in Ireland but all over the world and I want you all to know that I am very grateful for all the visits, emails and phone calls during the last few years. Your friendship has been a pleasure and I hope it will continue long into the future.

I feel very extremely fortunate to have a family and fiancée that are always so supportive. I would like to thank my brother David for all the advice and encouragement during the years. You were the first person to introduce me to robots though they have advanced quite a bit since then! I would like to thank my sister Emer for making me laugh whenever we speak and taking my mind off all my worries. I would like to thank my fiancée, Meadhbh, for all the love and understanding given to me, not only during my thesis, but throughout our time together that has been exceptional. You really made this possible and I am lucky to have you! Your presence during the last stages of this work was crucial to me and I am looking forward to starting the next stage of our life together.

Finally, I would like to thank my parents. Words really cannot express how grateful I am for your unconditional love and support. I have now been aboard for several years and missed many important events, which has been very difficult for all of us. However, I want you to know that all the telephone conversations, visits and emails have been a source of constant pleasure to me and kept me going through all the hard times. I could not have achieved this without you. Thus, I dedicate this work to you.

Philip Long

Nomenclature

General Notation

$\mathbf{q}_i$ Joint positions of leg i

$\dot{\mathbf{q}}_i$ Joint velocities of leg i

$\ddot{\mathbf{q}}_i$ Joint accelerations of leg i

$\mathbf{\Gamma}_i$ Actuated joint torque of leg i

$\mathbf{r}_i$ Position Vector from origin to point i

$\mathcal{R}_o$ Represents the frame o

$\mathbf{h}_i$ Wrench at point i

$\mathbf{f}_i$ Force at point i

$\mathbf{n}_i$ Moment at point i

$\mathbf{V}_i$ Kinematic twist at point i

$\mathbf{v}_i$ Linear Velocity at point i

$\boldsymbol{\omega}_i$ Angular Velocity at point i

${}^j\mathbf{S}_i$ Screw transformation matrix from frame i to frame j

${}^j\mathbf{T}_i$ Transformation matrix of frame i with respect to frame j

${}^j\mathbf{x}_i$ Vector representation of transformation matrix from frame i to frame j

$\mathbf{A}_i$ Inertia matrix of chain i in joint space

$\mathbf{B}$ Vector of Coriolis, Centrifugal and joint torques matrix in Cartesian Space

$\mathbf{M}$ Inertia matrix of robot in Cartesian space

$\mathbf{c}_i$ Vector of Coriolis, Centrifugal and joint torques matrix of chain i

${}^j\mathbf{J}_i$ Kinematic Jacobian matrix of chain i in frame j

$\mathbf{I}_n$	Identity matrix of dimension n
$\mathbf{0}_n$	Zero matrix of dimension n

Chapter 2

$\mathbf{W}$	Grasp Matrix
$\$_\lambda$	Screw of pitch λ
ν_λ	A twist in screw notation of pitch λ
ζ_λ	A wrench in screw notation of pitch λ
$\boldsymbol{\tau}_{cl}$	Closed Loop Motor torques
$\mathbf{q}_a$	Vector of actuated joint positions
$\mathbf{q}_p$	Vector of passive joint positions
$\mathbf{q}_c$	Vector of cut joint positions
$\dot{\mathbf{q}}_a$	Vector of actuated joint velocities
$\dot{\mathbf{q}}_p$	Vector of passive joint velocities
$\dot{\mathbf{q}}_c$	Vector of cut joint velocities
$\ddot{\mathbf{q}}_a$	Vector of actuated joint accelerations
$\ddot{\mathbf{q}}_p$	Vector of passive joint accelerations
$\ddot{\mathbf{q}}_c$	Vector of cut joint accelerations

Chapter 3

$\mathbf{q}_e$	Generalized elastic position variables
$\dot{\mathbf{q}}_e$	Generalized elastic velocity variables
$\ddot{\mathbf{q}}_e$	Generalized elastic acceleration variables
$\mathbf{Q}_p$	Elastic generalized force
$\mathbf{\Phi}_{dk}(i)$	Displacement shape function of mode k at point i
$\mathbf{\Phi}_{rk}(i)$	Rotation shape function of mode k at point i

Chapter 4

$\mathbf{s}$	The vector of image features
$\mathbf{L}$	The Interaction matrix
$\mathbf{C}$	Matrix of collineation or camera parameters
ζ	The slicing/pressing ratio

Résumé Étendu

Introduction

Il y a, en ce moment, un désir général d'appliquer des robots aux tâches non industrielles. Par conséquent, les robots doivent évoluer pour mieux agir dans un environnement qui n'est pas conçu spécifiquement pour eux. Grâce á cela il y avait, récemment, plusieurs innovations robotiques dans les champs de conception et commande. Dans cette thèse, on focalise sur une de ces innovations, les robots coopératifs.

Cette thèse est effectuée dans la cadre de le projet ARMS. Le projet ARMS est un projet de recherche composé de deux partenaires académiques (IRCCyN et INSTITUT PASCAL), un partenaire industriel (CLEMESSY) et un Centre technique (l'ADIV). L'objectif du projet ARMS est de proposer un système robotique qui peut séparer de muscles de bouef dans un environnement dynamique. Le projet utilise un système multi-bras pour simultanément repérer la trajectoire de découpe, suivre et couper autour cette trajectoire, fixer le muscles et applique un effort pour aider la séparation. Il y a cinq modules dans ce projet:

1. Module de Faisabilité (ADIV): Cette tâche est définie comme une étude de faisabilité de différents scénarios de séparation de la viande. En outre, les outils de validation pour la découpe qui sont développées dans ce module.
2. Module de commande (IRCCyN): Cette tâche est définie comme la modélisation et le contrôle des systèmes multi-bras pour qu'une stratégie de contrôle cohérent puisse être mis en œuvre qui prend en compte les capteurs de système. La commande est fortement liée au module de perception. En plus, un modèle avancé d'objet est utilisé pour obtenir des entrées pour la commande par exemple l'effort de traction. Le modèle d'objet est construit par l'INSTITUT PASCAL.
3. Module de la Perception (INSTITUT PASCAL): Cette tâche nécessite la construction de un système de vision active qui peut extraire une trajectoire de coupe. Le module de perception est également nécessaire d'initier et de ré-initialiser le modèle d'objet avancé.
4. Module de Conception (INSTITUT PASCAL): Les tâches de conception consistent à identifier et ensuite spécifier le système de multi-bras nécessaire. Ce mod-

ule prend en compte la conception de l'outil de coupe et le système qui retenir la viande.

5. Module d'intégration (CLEMESSY): Cette tâche est définie comme l'intégration de tous les modules dans la cellule robotisée. L'évaluation expérimentale de la stratégie globale fait partie de ce module.

Cette thèse est focalisée sur le module de commande. Cependant, á partir de cette motivation centrale plusieurs autres avenues de recherche sont examinées. Donc, il y a trois contributions de cette thèse concernant différentes configurations de manipulation coopérative.

La première concerne l'étude analytique d'un système coopératif de basse mobilité qui tient un objet rigide. Le système comprend deux bras, chacun de 5 DDL (degré de liberté) . En totale le système a dix actionneurs, donc quand un objet est saisi la boucle fermée est sur actionnée. Des études cinématiques permettent d'obtenir la mobilité qui est quatre, donc ça veut dire qu'il ne faut que quatre moteurs indépendants pour commander l'objet dans l'espace. Pour choisir le meilleur ensemble d'actionneurs indépendants on utilise des analyses cinématiques et dynamiques pour trouver les singularités et la performance dynamique respectivement.

La seconde porte sur la modèle dynamique d'un système coopératif qui tient un objet flexible. On propose une nouvelle façon de modéliser ce système qui consiste de séparer les parties rigides, les bras de le robot, et les parties flexibles. L'objet est représenté par les fonctions de formes tandis que les bras sont modélisés par les méthodes classiques. Les deux parties sont liées en calculant les efforts á le repère de chaque organe terminale. En suivant cette méthode une solution de forme fermée est dérivée. La modélisation est validée par la comparant avec un simulateur dynamique commerciale. Pour montre la pertinence de ce système, on montre comment cet algorithme pourrait être appliqué aux robots parallèles, notamment au robot Gough Stewart.

La dernière concerne la séparation d'objets mous par plusieurs robots. La construction d'un simulateur d'un système multi-bras pour la découpe de viande est décrite. Une commande par vision/effort est développée qui permet le système de s'adapter d'en fonction de l'état de l'objet. Des expérimentations sont effectuées et montrent comment, pendent la découpe, l'effort qui est généré par la résistance de l'objet peut servir pour éviter les déformations globales d'objets. Cette stratégie de découpe est utilisée avec un système de vision pour que le robot puisse suivre une trajectoire déformable. On propose deux schémas différents. Le premier schéma utilise PBVS pour couper autour une trajectoire planaire avec une estimation hors-ligne. Le deuxième schéma utilise IBVS pour couper autour une trajectoire 3D sans estimation.

La thèse est organisée dans trois chapitres principaux: **Coopération Manipulation des objets rigides**, **Coopération Manipulation des objets flexibles** et **La découpe d'objets mous par commande en vision/effort**. Ce résume décrit en détail la contribution de chaque chapitre.

Chapitre 2: Coopération Manipulation des objets rigides

Contribution

Dans cette chapitre, on focalise sur la manipulation coopérative des objets rigides. La première partie de cette chapitre décrit l'état de l'art dans ce thématique. Elle comprend les méthodes principales de modéliser, commander et analyser ces systèmes. On décrit la différence entre des systèmes avec une mobilité complète et celles avec mobilité réduite. Dans la deuxième partie de cette chapitre on présente notre contribution, celle d'une analyse cinématique et dynamique d'un système coopératif à mobilité réduite. On analyse la boucle fermée qui consiste de deux bras de robot NAO et un objet rigide. Dans cette section, on résume la contribution de cette chapitre.

Description du système

Chaque bras du robot comprend cinq articulations pivots. Les deux premières articulations sont coïncidentes mais orthogonales et donc forme un joint de Cardan. Les trois dernières articulations forment une rotule. Par conséquent, en tenant en compte l'objet rigide, on peut décrire le système comme 2-U-S robot parallèle. On modélise le système par deux façons différentes.

D'abord, en utilisant le paramètres de M-DH [KK86], on obtient les équations de contraintes cinématiques. Pour faire cela, on modélise le système comme robot arborescent en coupant la chaîne à une certaine articulation. Les équations de contraintes garantissent que la boucle reste fermée.

Deuxièmement, on utilise la théorie des vis [Hun78, KG07]. La théorie des vis est un outil géométrique souvent employé dans les études sur les robots parallèles. L'avantage de théorie des vis est la facilité de trouver des configurations intéressantes. Une vis peut représenter soit un vecteur de vitesse, nommé une vis cinématique, soit un vecteur d'effort, nommé une vis d'effort. Pour un système de vis de dimension n, il existe un système réciproque de dimension $6 - n$.

Analyse de mobilité

La vis cinématique des deux bras, est obtenue d'une étude de la position des liaisons. Pour chaque bras, on trouve la vis d'effort réciproque á la vis cinématique de ce bras. On les appelle ζ^c_{0r} et ζ^c_{0l} où l'union est $\mathcal{W}^c$. La vis cinématique de l'objet, qui est tenu par les deux bas, est celle qui est réciproque au $\mathcal{W}^c$. On peut trouver le DDL de l'objet par le rang de la vis cinématique.

Le résultat de cet analyse montre, que dehors les configurations spéciales, l'objet possède quatre DDL malgré le fait que chaque bras possède cinq DDL. Par ailleurs, cette analyse peut montrer la nature de ces motions. À partir de cette étude on trouve trois types de motions différentes, qui dépend sur le rapport entre ζ_{0r}^c et ζ_{0l}^c et donc sur la configuration des bras. Si ζ_{0r}^c et ζ_{0l}^c sont parallèles, la motion possible est 2T2R (deux translations et deux rotations) , si ζ_{0r}^c et ζ_{0l}^c sont sécantes, la motion possible est 1T3R (une translation et trois rotations), et finalement si ζ_{0r}^c et ζ_{0l}^c ne sont pas parallèles ni sécantes, la motion est aussi 1T3R (une translation et trois rotations).

Schéma d'actionnement

Le robot possède de dix moteurs tandis que l'objet ne possède que quatre DDL. Par conséquent, le système est sur-actionné donc il faut choisir quatre actionneurs qui peuvent commander l'objet. Les autres actionneurs restent passifs et prennent des valeurs pour respecter le systèmes de contraint. En totale, il y a 210 schémas d'actionnement, cependant en éliminant tous les schémas symétrique, et tous qui sont inadmissible, il reste 71 schémas distincts.

Un schéma inadmissible contient un ensemble des actionneurs qui ne peuvent pas fixer l'objet malgré la configuration du robot. Pour découvrir si un schéma est inadmissible il faut analyser la vis d'effort qui est appliqué sur l'objet par chaque actionneur.

Donc, pour un schéma d'actionnement, on utilise la théorie des vis pour trouver la vis d'effort qui est associée avec chaque actionneur. Celle-ci est réciproque aux tous les *autres* actionneurs de la chaîne cinématique. On répète ce processus pour les quatre actionneurs de schéma pour trouver le torseur appliquer sur l'objet par ce schéma, qui est nommé $\mathcal{W}^a$. Le système de contraintes de l'objet comprend $\mathcal{W}^a$ et $\mathcal{W}^c$. Si ce système n'a pas un rang égal à 6, il existe au moins une vis cinématique de l'objet que le schéma d'actionnement ne peut pas commander. En réalité, ça veut dire que l'objet peut bouger librement dans une direction malgré les actions de moteurs.

On trouve 39 schémas d'actionnement qui sont inadmissible. On montre pourquoi ils sont inadmissibles et la vis cinématique objet qui ne peut pas être commandée par les moteurs est décrite.

Analyse de singularité

Dans cette section, on discute les singularités du système. Il y a deux genres de singularités: la singularité de bras et la singularité parallèle [ACWK11].

Les singularités de bras

Les singularités du bras se produit où le système de la vis cinématique d'un bras, normalement de rang 5, perd rang. On peut trouver les singularités de bras facilement et pour un robot de structure U-S, elles sont bien connues. En revanche, dans cette section on montre l'effet du schéma d'actionnement sur le comportement du système lorsque qu'il est dans une position singulière. En bref, l'objet perd un DDL, mais avec une mauvaise sélection de schéma d'actionnement le système souffre aussi d'une perte de rigidité dans un de ces liens. Cette sorte de singularité est souvent appelée une singularité interne.

Les singularités parallèles

On définit deux types des singularités parallèles, les singularités à cause de la structure robotique et les singularités à cause du choix de schéma d'actionnement, qui sont désigné comme des singularités de contraintes et des singularités de d'actionnement, respectivement.

Les singularités de contraint sont indépendantes de choix de schéma d'actionnement et elles se passent où le système de contraints, $\mathcal{W}^c$, perd rang i.e. la configuration où ζ_{0r}^c et ζ_{0l}^c sont dépendantes. En trouvant cette configuration, on constate que le système n'est pas seulement dans une singularité de contrainte mais aussi que chaque bras est dans une singularité de bras. Par conséquent, pour découvrir le comportement d'objet il faut effectuer une étude de haut détail de tout le torseur de système. Cette étude montre qu'en lieu de perdre la rigidité, comme attendu, le système perd un DDL

Les singularités de d'actionnement sont dépendantes sur les actionneurs choisi. Elles se produisent où le système global de contraints, qui comprend $\mathcal{W}^c$ et $\mathcal{W}^a$ perd rang mais le rang de $\mathcal{W}^c$ est maximal. On donne, dans cette section, deux exemples de singularités de d'actionnement.

Comparaison entre la théorie des vis et les moyens classiques

Dans cette section, on valide l'analyse effectuée par la théorie de vis en utilisant les moyens classiques, notamment par une analyse de la matrice Jacobienne. Dans la première section de ce chapitre, on modélise le système avec le M-DH paramètres. Cette modélisation permet le calcul de trois matrices qui peuvent être utilisée pour analyser profondément le comportement du système. Les trois matrices sont, $\mathbf{G}_p$, $\mathbf{J}_p$ et $\mathbf{J}_{act}$, qui sont obtenues à partir de (2.59), (2.64), et (2.66), respectivement.

Pour plusieurs configurations aléatoires, les valeurs numériques de ces matrices sont étudiées. On montre la correspondance entre la théorie des vis et l'étude numérique dans Tableau 2.4.

Performance dynamique de schéma d'actionnement

Afin de classer les schémas d'actionnement, il faut considérer la performance dynamique des schémas admissible. On utilise plusieurs trajectoires pour tester la performance dynamique. À cet effet, on doit obtenir le modelé dynamique de la boucle fermée qui est faite par deux étapes. D'abord, on calcule le modèle dynamique inverse du robot arborescente équivalente. Puis, on trouve le modèle dynamique inverse de la chaîne fermée en utilisant les équations de contraintes cinématiques.

La trajectoire est définie par une polynômial de degré 5 dans l'espace articulaire. Elle consiste de bouger le robot d'une configuration initiale à une configuration finale dans un temps fixe. Le schéma est montré dans Fig.2.20.

De cette manière, on peut tester la performance de tous les schémas d'actionnement pour la même trajectoire. En totale 300 trajectoires sont utilisées. Pour chaque trajectoire on mesure la performance de 64 schémas d'actionnement. Il y a deux critères pour juger la performance d'un schéma d'actionnement. Le premier critère indique la perte de puissance pendant une trajectoire, définie par [CA01]:

$$\eta = \int_0^t \boldsymbol{\tau}^T \boldsymbol{\tau} \, dt$$

Le deuxième critère est la violation de la valeur maximum de couple pendant la trajectoire. On présente deux figures qui montent les valeurs de ces deux critères pour tous les essais.

Sélection de Schéma Actionnement

La section finale de ce chapitre discute le choix de schéma actionnement. Dans cette section, on compare la performance dynamique des schémas en ce qui concerne les essais dans la section précédent. En plus, on élimine les schémas qui ont les singularités d'actionnement qui réduiraient l'espace de travail. Finalement, on favorise les schémas qui distribuent les moteurs également aux deux bras. Par conséquent, on pourra sélectionner le meilleur schéma parmi les 71 candidats admissibles.

Chapitre 3: Coopération Manipulation des objets flexible

Contribution

Dans cette chapitre, on focalise sur la manipulation coopérative des objets flexible. La première partie de cette chapitre décrit l'état de l'art dans ce thématique. Elle comprend les définitions d'objets flexibles utilisé en robotique et aussi les méthodes principales de modéliser ces objets. On présente un formalisme généralisé souvent utilisé pour les robots simples et les robots parallèles avec flexibilité qui comprend les relations cinématiques et dynamiques. Finalement, on présente les stratégies coopératives qui existent pour commander et changer les formes des objets flexibles. Dans la deuxième partie de cette chapitre on présente notre contribution, la modélisation dynamique de robots coopératifs avec objets souples. Dans cette section, on résume la contribution de cette chapitre.

Description du système

On présente une nouvelle méthode pour la modélisation dynamique de plusieurs robots qui tiennent un objet commun souple. Cette méthode est générale et donc pourrait être utilisé pour n'importe quel système. La solution est basée sur la décomposition de robot en deux systèmes, un rigide et l'autre flexible. Le système rigide comprend les bras et la base et le système flexible comprend l'objet. Les deux systèmes sont liés en utilisant les efforts appliqués par les organes terminaux. La modélisation généralisée concerne un système qui a:

- n bras tenant un objet flexible
- Chaque bras consiste de a actionneurs et m liens
- L'articulation au point d'attachement peut transmettre un torseur de dimension c
- La flexibilité de l'objet est d'ordre N

Á partir ces conditions, on dérive le modèle directe et inverse d'un série de robots coopératifs qui manipule un objet flexible.

La modélisation de système rigide

Le but de cette section est d'obtenir les relations entre les efforts et couple au point d'attachement et les efforts appliqués par les actionneurs. On utilise les relations classiques pour trouver la vélocité et accélération de l'organe terminal i:

$$\dot{\mathbf{q}}_i = \mathbf{J}_i^{-1}\mathbf{V}_i$$
$$\ddot{\mathbf{q}}_i = \mathbf{J}_i^{-1}\left(\dot{\mathbf{V}}_i - \dot{\mathbf{J}}_i\dot{\mathbf{q}}_i\right)$$

Finalement, on calcule l'effort à ce point comme une fonction de les dynamiques de la jambe et les efforts appliqués par les actionneurs.

$$\mathbf{F}_i = \mathbf{J}_{ai}\tau_{ai} - \mathbf{J}_i^{-T}\left(\mathbf{A}_i\ddot{\mathbf{q}}_i + \mathbf{c}_i\right)$$

La modélisation d'objet flexible

Comme la dernière section, le but est de trouver les relations entre l'origine de l'objet et les variables au point d'attachement, où l'origine se situe dans le centre de l'objet. La flexibilité d'objet est représentée par des fonctions de formes [BK98, RCM10]. Ces fonctions nous permettent de trouver la position d'un point de la plateforme par la somme de la position rigide et la déformation. En plus, la même méthodologie pourrait être utilisée pour trouver la vélocité et l'accélération. Les fonctions de frontière pour l'objet sont définies par les conditions au point d'attachement. La position d'un point d'attachement du robot i pourrait être trouvé par la position de l'origine par:

$$\mathbf{p}_i = \mathbf{p}_p + \mathbf{r}_i$$

La vélocité est obtenue:

$$\mathbf{v}_i = \mathbf{v}_p + \boldsymbol{\omega}_p \times \mathbf{r}_i + \boldsymbol{\Phi}_d(i)\dot{\mathbf{q}}_e$$

L'accélération est obtenue:

$$\dot{\mathbf{v}}_i = \dot{\mathbf{v}}_p + \dot{\boldsymbol{\omega}}_p \times \mathbf{r}_i + \boldsymbol{\omega}_p \times (\boldsymbol{\omega}_p \times \mathbf{r}_i + \boldsymbol{\Phi}_d(i)\dot{\mathbf{q}}_e) + \boldsymbol{\Phi}_d(i)\ddot{\mathbf{q}}_e + \boldsymbol{\omega}_p \times \boldsymbol{\Phi}_d(i)\dot{\mathbf{q}}_e$$

Finalement, les efforts au point d'attachement sont obtenus à partir du torseur á l'origine d'objet:

$$\begin{bmatrix} \mathbf{h}_p \\ \mathbf{Q}_p \end{bmatrix} = \mathbf{W} \begin{bmatrix} \mathbf{F}_1 \\ \vdots \\ \mathbf{F}_6 \end{bmatrix}$$

Le torseur á l'origine d'objet égale aussi l'effet de dynamiques d'objet flexible:

$$\begin{bmatrix} \mathbf{f}_p \\ \mathbf{n}_p \\ \mathbf{Q}_p \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{rr} & \mathbf{A}_{re} \\ \mathbf{A}_{re}^T & \mathbf{m}_{ee} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}}_p \\ \dot{\boldsymbol{\omega}}_p \\ \ddot{\mathbf{q}}_e \end{bmatrix} + \begin{bmatrix} \mathbf{c}_r \\ \mathbf{c}_e \end{bmatrix}$$

Modèle Dynamique Générale

On trouve un modèle générale en remplaçant des efforts au point d'attachement avec

- Les dynamiques de l'objet y compris les effets de flexibilité
- Les dynamiques des jambes et les efforts appliqués par les actionneurs

Par cette méthode on peut écrire un modèle dynamique générale d'un système de n bras qui tient un objet flexible. La résolution des équations de modèle dynamique dépend sur la forme d'objet qui est saisi par le système. On identifie trois cas:

Les objets rigides Il n'y a pas de flexibilité dans l'objet donc les variables élastiques égalent zéro. En plus, si le système peut appliquer plus que six efforts sur l'objet des variables supplémentaire apparaît dans les équations. Ces variables sont des efforts internes qui ne contribuent pas à la motion de l'objet.

Les objets flexibles Il y a de flexibilité dans le système et nous ne pouvons pas la commander directement les variables qui le représentent. Pour résoudre ces équations, il faut deux étapes. D'abord il faut trouver une représentation des accélérations de variables élastiques en termes de les accélérations Cartésiennes et les couples des actionneurs. Ensuite, il faut utiliser cette expression pour éliminer des accélérations de variables élastiques dans le modèle dynamique.

Les objets articulés Il y a de flexibilité dans le système cependant la flexibilité est limité. En effet, le degré de flexibilité égale exactement le nombre des efforts *redondants* appliqués par les actionneurs sur l'objet.

Dans cette section nous résolvons ces équations pour chaque objet. Les solutions sont validées par une comparaison avec un simulateur commercial. Le système comprend deux robots planaires avec un objet flexible. La comparaison montre un bon accord entre le simulateur et notre méthode.

Modèle Dynamique de Gough Stewart Robot

Dans cette section, on applique la méthode générale au Gough Stewart robot. Ce robot est souvent utilisé dans pour applications industrielles et pour la recherche. Grâce à hautes accélérations qui pourraient être atteint par ce robot il est important de prendre en compte la flexibilité.

On décrit le système rigide y compris le modèle géométrique, cinématique et dynamique. L'idée est de trouver des relations entre les variables au point d'attachement et les variables de jambes, notamment celles entre les efforts transmis par la rotule et les couples de l'actionneur prismatique. Il y a un actionneur prismatique dans toutes les jambes. Un cardan connecte la jambe et la base.

$$\mathbf{f}_i = \mathbf{a}_{3i}\Gamma_{3i} - \mathbf{J}_i^{-T}\left(\mathbf{A}_i\ddot{\mathbf{q}}_i + \mathbf{c}_i\right)$$

Dans cette section, on illustre les étapes nécessaires pour trouver le modèle dynamique du système entier. La modèle dynamique de ce robot crée un rapport entre les accélérations Cartésiennes et les couples de jambes. Le modèle direct et le modèle inverse sont calculés dans la même manière. Pour trouver ces modèles on suit les étapes suivantes. D'abord, on réécrit le modèle dynamique de la jambe pour remplacer les variables articulaires par les variables au point d'attachement. Deuxièmement, on remplace les variables de point d'attachement avec les variables d'origine de la plateforme. Les équations pourraient être manipulées pour obtenir le modèle dynamique:

$$\mathbf{A}\dot{\mathbf{V}}_p + \mathbf{c} = \mathbf{J}_{sys}^{-T}\boldsymbol{\Gamma}$$

La matrice $\mathbf{A}$ représente l'inertie totale de le système:

$$\begin{aligned}\mathbf{A} = \mathbf{A}_{rr} &+ \mathbf{W}_p\mathbf{A}_x\mathbf{W}_p^T \\ &- \mathbf{A}_{re}\mathbf{A}_{ee}^{-1}\left(\mathbf{A}_{re}^T + \mathbf{W}_e\mathbf{A}_x\mathbf{W}_p^T\right) \\ &- \mathbf{W}_p\mathbf{A}_x\mathbf{W}_e^T\mathbf{A}_{ee}^{-1}\left(\mathbf{A}_{re}^T + \mathbf{W}_e\mathbf{A}_x\mathbf{W}_p^T\right)\end{aligned}$$

La matrice $\mathbf{J}_{sys}$ est le Jacobian de le système:

$$\mathbf{J}_{sys}^{-T} = \mathbf{J}_p^{-T} - \left(\mathbf{A}_{re} + \mathbf{W}_p\mathbf{A}_x\mathbf{W}_e^T\right)\mathbf{A}_{ee}^{-1}\mathbf{J}_e^{-T}$$

Finalement, le vecteur **c** représente les efforts Coriolis, centrifuge et gravité:

$$\begin{aligned}\mathbf{c} = \mathbf{c}_r + \mathbf{W}_p\mathbf{A}_x\mathbf{h} + \mathbf{W}_p\mathbf{c}_x - \\ \left(\mathbf{A}_{re} + \mathbf{W}_p\mathbf{A}_x\mathbf{W}_e^T\right)\mathbf{A}_{ee}^{-1}\left(\mathbf{W}_e\mathbf{A}_x\mathbf{h} + \mathbf{W}_e\mathbf{c}_x + \mathbf{c}_e\right)\end{aligned}$$

Nous pouvons constater que ce système comprend un objet flexible plutôt qu'un objet articulé ou un objet avec flexibilité réduite. Á cause de ça, on utilise la solution de deux étapes comme décrit dans la section précédente. Le système est validé encore par l'utilisation du simulateur. Dans ce cas-là on valide le comportement du modèle inverse et aussi le modèle directe. Les résultats montrent un très bon accord entre les sorties de simulateur commercial et les variables calculés par notre modèle.

Validation de Modèle Dynamique

Dans cette section, la méthode pour valider les modèles dynamiques est décrite. La plateforme est modélisée par les éléments finis, notamment par le programme MSC NASTRAN. Ce logiciel nous permet d'obtenir les fonctions de forme, la raideur et la masse généralisée de cette plateforme. Ensuite, la plateforme est implantée dans MSC ADMAS et branchée aux jambes par les rotules. Une trajectoire est définie dans l'espace Cartésien pour le robot et pendant cette trajectoire on sauvegarde toutes les données nécessaires pour la validation du modèle inverse et du modèle directe.

Modèle Dynamique Inverse

Le modèle dynamique inverse est souvent utilisé dans la commande dynamique de robots. La modèle doit calculer le couple pour une accélération désirée et un état actuel de robot. Pour valider le modèle dynamique inverse, on prend l'accélération, vélocité et position Cartésienne et aussi la position et vélocité de les variables élastiques sortant du système MSC ADAMS. Á partir de ces variables on calcule le couple. Le résultat est comparé avec le couple d'ADAMS et on montre que la modèle peut calculer précisément ce variable. La deuxième étape s'agit du calcul de l'accélération des variables élastiques. On montre que le modèle peut aussi calculer ces variables.

Modèle Dynamique Direct

Le modèle dynamique direct est souvent utilisé dans la simulation dynamique de robots. La modèle doit calculer l'accélération résultant de couple. Pour valider le modèle dynamique inverse, on prend couple, vélocité et position Cartésienne sortant du système MSC ADAMS et on calcule le couple. Á partir de ces variables on calcule l'accélération désirée. Le résultat est comparé avec l'accélération d'ADAMS et on montre que la modèle peut calculer précisément ce variable.

Chapitre 4: La découpe d'objets mous par commande vision/effort

Contribution

Dans cette chapitre, on propose une loi de commande en vision/effort qui pourrait séparer des objets mous. La première partie de cette chapitre décrit l'état de l'art dans ce thématique. D'abord, on décrit l'utilisation d'asservissement visuel y compris la type de primitive, la location de la caméra et la matrice d'interaction. Ensuite, on montre comment on peut intègre ces schémas avec une consigne d'effort. Finalement, on discute la séparation d'objets mous par des robots. Cette partie comprend la simulation des objets déformable et la loi de commande de robots qui font la séparation. Dans la deuxième partie de cette chapitre on présente notre contribution la découpe d'objets mous par commande vision/effort. Il y a deux objectives principales. D'abord, la construction d'un simulateur pour tester la loi de commande pour la cellule robotique de le projet ARMS. Deuxièmement, la validation expérimentale d'un loi de commande qui pourrait séparer des objets mous en utilisant le commande par vision/effort. Dans les deux cas, on utilise le Kuka robot, un robot de 7-DOF qui pourrait être commandé en couple.

La modélisation et commande cellule robotique pour la découpe de viande

Dans cette section on décrit la construction et fonctionnement d'un simulateur robotique. Ce simulateur pourrait servir pour optimiser la position des robots, pour tester le commande en vision/effort et finalement pour tester le schéma de résolution de redondance.

Construction du simulateur

Les robots doivent travailler ensemble pour séparer un objet déformable qui représente les muscles. Le simulateur consiste de trois modules et trois robots.

1. **Commande** Ce module est écrit en Matlab et Simulink. Il consiste de la loi de commande qui génère soit un couple motorisé soit une vélocité articulaire. La consigne comprend l'état des robots, l'état de l'objet déformable et les sorties visuelles.

2. **Dynamique Simulateur** Ce module est écrit en MSC ADAMS. Il consiste de trois robots qui peut être commandés en couple ou en vélocité articulaire, et aussi l'objet déformable. Le robot est construit par une série des corps rigides liée par des articulations de 1-DDL.

3. Objet déformable L'objet déformable représente les muscles qui doivent être séparés. Les deux muscles sont modélisés par les méthodes des éléments finis. Cette représentation est efficace et le comportement pourrait être calculé dans les temps réels. Pourtant, pour que le couteau puisse séparer l'objet, un troisième objet déformable est défini qui lie les deux muscles. En réalité, le troisième objet déformable représente les aponévroses. Le comportement des aponévroses élastique est modélisée par une série des ressort et amortisseurs.

Loi de commande et résultats

La loi de commande, les entrées/sorties et comportement désirées de chaque robot est décrit dans cette section. La cellule comporte trois robots.

On appelle le premier robot le robot qui coupe. Ce robot doit suivre la trajectoire et séparer les deux muscles. Le couteau doit trancher les aponévroses sans entrant dans les deux muscles. Il faut que le robot suive une trajectoire qui change en X Y et Z. Pour le faire, on crée une trajectoire qui varie en X et Y mais pas en Z. On appelle cette trajectoire un passage. Pour chaque passage une estimation de la surface est utilisée pour générer une trajectoire en position, vélocité et accélération. Pendant la découpe, il faut faire une mise á jour de cette trajectoire á cause des déformations d'objets pour assurer que le couteau suivre la nouvelle définition de la surface. Cette mise á jour locale change la position et l'orientation de le couteau. Pour commander ce robot, on utilise un commande par *computed torque*.

On appelle le deuxième robot le robot qui tire. Ce robot doit fixer l'objet déformable. En plus le robot tire sur l'objet. Pendant la découpe, l'effort élastique engendré par les aponévroses diminue. Donc l'effort appliqué par le robot ouvre la vallée de découpe de plus en plus. Il y a deux objectifs pour ce comportement. D'abord, le couteau peut entrer plus profond dans la vallée grâce á cet élargissement. Deuxièmement, dans l'application réelle, pour aider le système de la vision il faudrait ouvrir, autant que possible, la vallée de découpe. Pour commander ce robot, on utilise un commande par *impedance* classique ou l'effort désiré est appris hors-ligne.

On appelle le troisième robot le robot de vision. Ce robot est responsable pour l'extraction de la trajectoire On commande ce robot dans l'espace d'image. Ça veut dire qu'un image qui représente la région de découpe est commandé directement. Il est désirable que cet image reste dans le champ de vision. Ce robot est commandé par sa vélocité articulaire.

L'efficacité de la loi de commande est montré dans cette section. Le système est capable de séparer l'objet déformable en faisant plusieurs passages. Les résultats montrent comment la mise á jour compense pour la déformation de l'objet. En plus, le robot qui tire, montre un comportement désiré i.e un élargissement graduel grâce á l'effort constant.

La découpe d'objets mous par commande vision/effort

Dans cette section, la validation expérimentale de la loi de commande est décrite. La limitation principale de le simulateur est le manque d'effort de la découpe. Cet effort se produit pendant la découpe et dans cette section on montre comment la loi de commande peut en servir pour mieux séparer l'objet. En plus, cette section montre comment on peut utiliser un système de vision avec cet effort de suivre une trajectoire flexible.

Commande en force pour découper des corps mous

L'effort nécessaire de séparer un objet mou peut être formalisé par l'énergie du système qui est la somme de l'energie de découpe, l'énergie perdu á cause du frottement et l'énergie perdu pendant la déformation de l'objet.

$$W_r = W_c + W_f + U$$

Dans cette section on propose une loi de commande qui réduit l'effort nécessaire pendant la découpe. Il y a deux motivations de réduire l'effort de découpe. D'abord, pour certains outils délicats, un gros effort peut les endommager. Deuxièmement, si l'effort est trop haut la qualité de la découpe est réduite notamment avec de grosses déformations permanents du matériel.

La loi de commande utilise l'effort capturé pendant la découpe pour changer la trajectoire du couteau. Si l'effort est trop haut une contrainte de cisaillement est engendrée par la loi de commande. Ce phénomène réduit l'effort comme illustré dans [AXJ04, RTLMM12]. Á la même temps, le deuxième robot tire l'objet pour réduire l'énergie perdu á cause du frottement. La loi de commande est testée pour plusieurs valeurs de paramètres et aussi plusieurs valeurs de l'angle de découpe.

Les résultats valident la loi de commande comme une méthode de réduire l'effort de séparation. Les résultats des courbes qui montrent la réduction d'effort et aussi des images qui illustrent la qualité de la découpe. Dans les sections suivantes on utilise cette loi de commande avec un système de vision (une caméra embarquée sur l'outil) pour séparer un objet déformable.

Commande par PBVS

Dans cette section, on propose le premier de deux lois de commande qui combine la stratégie de découpe avec un système de vision. Dans ce cas, on utilise un système PBVS. L'objet est planaire est la hauteur de l'objet est connue. La trajectoire est définie par une courbe fixée au objet. Une estimation de cette courbe est disponible. En utilisant cette estimation, on définit une vélocité constante pour couper l'objet.

Un étalonnage est effectué pour trouver les rapports entre la caméra et le couteau, et les paramètres internes de la caméra. Á chaque instant, le système de vision prendre une vue locale de la région de découpe. Á partir de cet image la position 3D de trois points de la trajectoire est reconstruite. Ensuite une déviation Cartésien est généré qui corrige l'erreur entre l'estimation hors-ligne et la position réelle de la trajectoire. Par cette stratégie, le système peut compenser pour la déformation de l'objet. Á la même temps la loi de commande d'effort assure que des déformations globales et les dégradations de qualité sont évitées.

Le deuxième robot tire sur l'objet pour réduire la quantité d'énergie perdu par les effets de frottement. Le robot est commandé en utilisant une loi de commande *impedance*:

$$\boldsymbol{\tau} = {}^0\mathbf{J}_t^T \left(\mathbf{k}_p \, {}^0d\mathbf{X}_t \, + \mathbf{f}_p\right) + \mathbf{H}$$
$${}^0d\mathbf{X}_t = \left[\begin{array}{cc} {}^0d\mathbf{p}_t^T & {}^0\delta_t^T \end{array}\right]^T$$

Les résultats de cette section montrent l'efficacité de la loi de commande. On peut voir clairement la réduction d'effort grâce á la stratégie de coupe. En plus, la précision de suivi de trajectoire est évident malgré la déformation á cause de le deuxièmement robot. Finalement, les courbes montrent peu d'effort quand le couteau passe par les régions déjà coupées qui montrent l'avantage de le deuxième robot.

Commande par IBVS

Dans cette section, on propose le deuxième de deux lois de commande qui combine la stratégie de découpe avec un système de vision. Dans ce cas, on utilise un système IBVS. On propose ce système puisque il y a quelques limitations avec la stratégie de PBVS et par utilisant un système IBVS, avec un image beaucoup plus riche, il est possible de surmonter ces limitations.

Hauteur de l'objet: La hauteur de l'objet est connue pour le système PBVS, sinon on ne pourrait pas garantir une découpe á une profondeur constante. Avec IBVS, même si la hauteur de l'objet change on pourrait fixer une différence constante entre l'objet et la caméra.

Orientation de l'objet: Avec le système PBVS, l'objet est dans un plan. Si l'orientation de l'objet change, il y aurait de problèmes avec la reconstruction 3D. Avec l'IBVS, le système peut réagir aux changements d'orientation pour assurer que la caméra reste parallèle á la surface de l'objet.

Étalonnage de cellule: Avec le système PBVS, puisque le système est commandé dans l'espace Cartésien il est très sensible au étalonnage, en fait, la précision est définie par l'étalonnage qui est très difficile et longue d'effectuer. Par contre, pour le système IBVS, la commande est effectuée dans l'espace d'image. Donc, si la tâche est bien définie, les erreurs d'étalonnage pourraient être éliminé par le boucle ferme présente dans la loi de commande.

Pour commander dans l'espace image il faut choisir une primitive convenable. Donc ce cas-là on choisit *Image Moments* [Cha04, TC05]. Cette primitive nous permet de commander toutes les motions de la caméra. Par exemple, le centre de gravité de l'image est utilisé de commander directement les motions X,Y de caméra. Il faut noter que si l'objet est parallèle á la caméra, l'aire de l'image peut commander le Z direction de la caméra.

La trajectoire est composée d'une série des marqueurs identiques. Á chaque instant pendant la trajectoire, la caméra extrait les moments du prochain marqueur. Le but est de faire converger les moments actuels avec les moments désirés. Pour trouver les moments désirés, le robot se place manuellement dans une bonne position. Dans cette position, la caméra est parallèle á la surface de l'objet est le couteau est prêt á couper. Cette méthode est connue comme *teaching by showing*. L'erreur entre les moments désirés et les moments actuels est transformé d'abord á la vélocité Cartésienne et puis á la vélocité articulaire par une matrice d'interaction:

$$\dot{\mathbf{q}} = \mathbf{J}_t^t \mathbf{S}_c \left(\mathbf{L}_{\mathbf{s}=\mathbf{s}^*}^{-1} \lambda \left(\mathbf{s} - \mathbf{s}^* \right) \right)$$

Finalement, pour utiliser notre stratégie de découpe, il faut introduire d'effort capturé dans la loi de commande. Puisque, la caméra reste toujours parallèle avec la surface de l'objet, la structure de la matrice d'interaction nous permet d'introduire la contrainte de cisaillement directement dans l'espace image. L'effort de découpe est lié directement avec l'aire de l'image qui grâce á cette matrice d'interaction change la vélocité Z de la caméra.

Les résultats dans cette section montrent comment le système peut couper une trajectoire qui varie en 3D. Le système peut maintenir une profondeur constante malgré ce changement dans la Z direction. En plus, le robot peut changer l'orientation de la caméra sans connaissance de structure de l'objet. Finalement, on montre comment la stratégie de découpe, proposé dans la section précédente peut être impliquée dans cette loi de commande.

Conclusion Générale

Cette thèse focalise sur la manipulation coopérative par plusieurs robots. Trois tâches différentes sont considérées. D'abord, la manipulation des objets rigides par de robot coopératifs de basse mobilité. Deuxièmement, la manipulation d'objets flexibles par de robots coopératifs. Finalement, la séparation d'objets mous par plusieurs robots en utilisant la commande vision/effort. Le travail est validé par les simulations et ou possible par les expérimentations. La contribution de ce travail est dans la cadre de modélisation et commande de robots coopératifs.

Coopération Manipulation des objets rigides

Une perspective intéressante serait de formuler une loi de commande qui change le choix d'actionner en utilisant la connaissance de singularité pour optimiser la performance. Autrement, il serait très intéressant avec beaucoup d'applications potentielles de voir si les vis d'efforts pourraient utiliser pour changer la forme d'un objet déformable. Finalement, on devrait analyser, avec la même méthodologie, la possibilité d'utiliser un schéma d'actionnement redondant.

Coopération Manipulation des objets flexibles

Il y a plusieurs façons de continuer ce travail. D'abord, le plus important serait de valider expérimentalement cette modélisation avec deux robots coopératifs et un objet flexible. Deuxièmement, un travail important serait d'inclure la flexibilité de jambes ou bras dans la modélisation et par conséquent avoir une représentation complète d'un robot flexible.

la découpe d'objets mous par commande vision/effort

Pour le simulateur, puisque il y a plusieurs robots dans un espace commun il est très important de prendre en compte des tâches secondaires comme la possibilité de collision ou les butées articulaires. Donc un système de résolution de redondance est nécessaire. Le but est de montre que une loi de commande centralisée pourrait, avec résolution de redondance, permettre le système de faire la tâche principale et respecter les contraintes secondaires.

Le travail expérimental doit être validé sur la vraie cellule avec la vraie viande. Ce serait possible aussitôt que la cellule serait prête. Finalement, la combinaison d'effort et IBVS est très nouveau et pourrait avoir beaucoup des avantages par rapport l'approche classique. Donc, il serait très intéressant d'utiliser cette méthode pour autres tâches et pas seulement la séparation des objets mous.

1

General Introduction

The idea of an autonomous machine or robot capable of replacing human labor, for tedious and time consuming tasks, has being a goal of engineers and inventors from ancient civilizations up to the present day. However, until very recently, the scope of robot applications was limited to highly repeatable industrial scenarios. Typically, these scenarios were monotonous for the human laborer and often the work environment was dangerous or at the very least uncomfortable. The industrial robots allowed an increase in production volume without a corresponding increase in cost, meaning that robots soon became an indispensable tool for manufacturing tasks.

The widespread success in these areas soon lead to a desire to apply robots to a variety of assignments in the service, education, medical and military domains. In attempting to accomplish this, it was observed that the serial industrial robot, though extremely efficient in the industrial environment, did not have a sufficient level of sophistication to function in the real-world environment. Furthermore, it was obvious that there was not an optimum solution, with respect to the robot's architecture, that was superior for all tasks. This lead to many innovations in robotic design, control and sensory capabilities. It is clear that robotics has become an interdisciplinary field and has attracted interest from a wide range of research institutions. However, in spite of their differences, the construction of every robot has certain commonalities, for example expertise in the subjects of mechanical and electronic engineering and computer science is required.

Today, as a result of the diversification of robot design, there exists a multitude of different types of robot architectures, for example mobile robots, parallel robots, unmanned aerial vehicles, self driving cars, bio-inspired robots and swimming robots. These robots are specially designed to function in their own surroundings and perform their respective tasks with a high degree of efficiency. Many of these robots therefore suffer from the same drawbacks as the original industrial robots i.e. they are optimally designed for one task and hence are unable to execute different tasks or function in disparate environments.

An alternative view of robot design, is to create the system with the same capabilities as a human being, and thus logically the robot will be able to function at a high level in our environment. In recent years in response to this idea, many humanoid robots have been built which has led to increased interest and innovation in several domains, for example vision based control and walking biped robots. Included in this alternative view has been a renewed focus on cognitive systems which endow the robots with intelligence, memory and learning capabilities. The ability to reason allows a robots to adapt and achieve complex goals in a real world environment.

This thesis focuses on another such innovation in recent years, namely the ability of multiple robots to cooperate on a common task. The use of multiple or cooperative manipulators has become imperative as robots move from the optimized industrial cells to a real world environment. To work together on a common task, different modeling and control strategies must be created. For example in a closed chain configuration, when a common object is transported by two or more manipulators, the serial robots are no longer free but are constrained by the object behavior. Although this creates more complex control schemes, it also allows the systems to complete a multitude of tasks and manipulate a diverse set of objects, in particular flexible objects. The resulting control strategies are useful not only for multi-robot systems but also essential for interaction and cooperation between humans and robots. There are many potential applications for such cooperation and the field of human robot cooperation is expected to greatly increase in the near future.

In this thesis, the cooperative tasks in question vary from manipulation of rigid and flexible objects to the controlled separation of deformable bodies. The tasks outlined in this thesis are similar to many problems present in the industrial and in the service domains. For instance, one such task involves the separation of meat by a multi-arm robotic system. This project is based on an industrial requirement to robotize a process that is facing severe labor shortages. Though the application is specific, the modeling and solutions obtained can be extended to several other operations.

The robot formalisms described in this thesis require advanced modeling strategies and the improved sensor capabilities. Modeling strategies allow a measure of prediction of task performance and generally improve the operating efficiency of the robot. Sensor based strategies allow the robots to interact with unknown objects and to react in a dynamic environment. In order to achieve any complex task autonomously, robot control schemes must exploit a judicious combination of the two. In the future, these tasks will become more prevalent as the role of robots is increased in everyday life.

This thesis is carried out in the framework of the ARMS project. The ARMS research project [1] *A multi-arms Robotic system for Muscle Separation*, funded by the ANR (Agence Nationale de la Recherche), reference ANR-10-SEGI-000, aims to contribute to the separation of beef rounds (hindquarters) by an autonomous robotic cell. The ARMS project consortium is composed of two academic partners, (IRCCyN and INSTITUT PASCAL), one industrial partner (CLEMESSY) and one technical center that focuses on the robotization of the meat processing industry (ADIV).

In France, meat processing accounts for over 25% of the food industry's total employees and includes over 2,000 companies. The robotization of meat cutting tasks is of increasing importance for several reasons. The unsocial working hours along with the strenuous, uncomfortable working conditions have created a shortage of skilled labor at a time when competition from low cost labor regions, notably from the MERCOSUR countries, is growing. Furthermore the physical tasks involved in the work lead to a high rate of musculoskeletal injuries [INS09].

The robotization of the meat processing industry has been the focus of several works worldwide. A general overview of the role of robots in the meat processing industry is outlined in [BVD08, CZF+13]. The Danish pig slaughter industry is an example of a successful robotization of a manual process. The automation process has improved both hygiene and accuracy in the manufacturing environment [Hin10]. In [GSGL10b, GSGL10a], a specific robotic meat cutting cell is analyzed from the point of view of the cutting parameters, while using bones as a positional guide. In Japan, robots have been widely introduced in poultry cutting operations [Kus10]. The previous works deal with highly repeatable scenarios in controlled environments, often aiming to optimize a well known existing process.

In contrast, the objective of the ARMS project is to enable the robotic system to autonomously separate highly variable beef rounds. A multi-arm system is proposed in order to deal with key challenges such as the irregularity of the target object and its deformable nature. This cell comprises three serial robots, two 6-DOF ADEPT Viper robots and a 7-DOF Kuka lwr. In addition to this a holding system for the meat muscles is devised with 2 DOF in order to allow greater access to the meat.

The robotic system must complete the same tasks as the human worker i.e. the first arm carries a knife and executes the cutting task while the second arm grasps the object and by applying force attempts to open up the cutting valley, finally the third robot carries the perception system that is used to obtain the cutting trajectory and update this trajectory as the object deforms. An advanced object model is created to predict object deformation and generate control signals for the multi-arm system. Therefore the project spans research domains such as cooperative robot motion, robot cell design, mechanical modeling of soft materials, force/vision control and visual tracking of deformable bodies.

The scientific challenges are grouped into five modules:

1. arms.irccyn.ec-nantes.fr

1. Feasibility/Exploration Module (ADIV): This task is defined as a feasibility study of various meat separation scenarios. In addition to this, the validation tools for the resulting cut are developed in this module.
2. Control Module (IRCCyN): This task is defined as the modeling and control of a multi-arm system such that a coherent control strategy can be implemented that takes into account the disparate sensors of the systems. The control is strongly linked to the perception module which is led by INSTITUT PASCAL. An advanced object model is used to obtain the inputs for the control such as pulling force. The object model is constructed by INSTITUT PASCAL.
3. Perception Module (INSTITUT PASCAL): This task requires the construction of an active vision system that can extract a cutting trajectory for the system by the on-line tracking of a deformable object. The perception module is also required to initiate and reset the advanced object model.
4. Design Module (Insitut Pascal): The design tasks consists of the identification and specification of the required multi-arm system. This module takes into account the design the cutting tool, the retaining system for the meat and the grippers.
5. Integration Module (CLEMESSY): This tasks is defined as the integration of the all modules into the robotic cell. The experimental evaluation of the proposed strategy is part of this module.

The realization of the objectives of the ARMS project is the primary motivation behind this thesis. The principal topics are multi-robot cooperation and the control of deformable/flexible components. In particular, the objective within this project is the modeling and control of cooperative manipulators in order to separate deformable objects. The central problem is thus the coordination of a multi-arm robot, however this scenario motivates research in several related domains:

- Cooperative manipulation
- Force/Vision control of cooperative robots
- Closed chain robots
- Separation of soft materials
- Deformable object modeling
- Robots with flexible components

In order to develop the separation strategy for the meat cutting task, studies on cooperative manipulators in closed chain configurations are carried out. Chapter 2 focuses on the cooperative manipulation of rigid objects by a dual arm system. Chapter 3 focuses on the cooperative manipulation of flexible objects by a dual arm system, deriving a dynamic model of a general closed chain system handling a deformable object. However, due to the variability of the meat muscles in the industrial environment, the applicability of the model-based approach to control is questionable. Hence, Chapter 4, proposes a series of sensor-based control schemes, in this case force and vision sensors, to adjust the robot's behavior as a function of the object's flexibility.

Therefore it can be seen that although different aspects of cooperative manipulation are outlined in each chapter of the thesis, there are many topics and strategies which are shared. The relationship between these chapters, the above research domains and the central motivation of the ARMS project is illustrated graphically in Fig.1.1. From this image, the areas that overlap are clearly defined, for example deformable object modeling, which is the focus of Chapters 3 and Chapter 4. Therefore the overall motivation for this work is to demonstrate the use of cooperative modeling and control formulations to execute complex tasks and to represent a range of robotic systems. Excluding the introduction, general conclusions and future work, this thesis is composed of three chapters. In the following, an overview of each chapter is given.

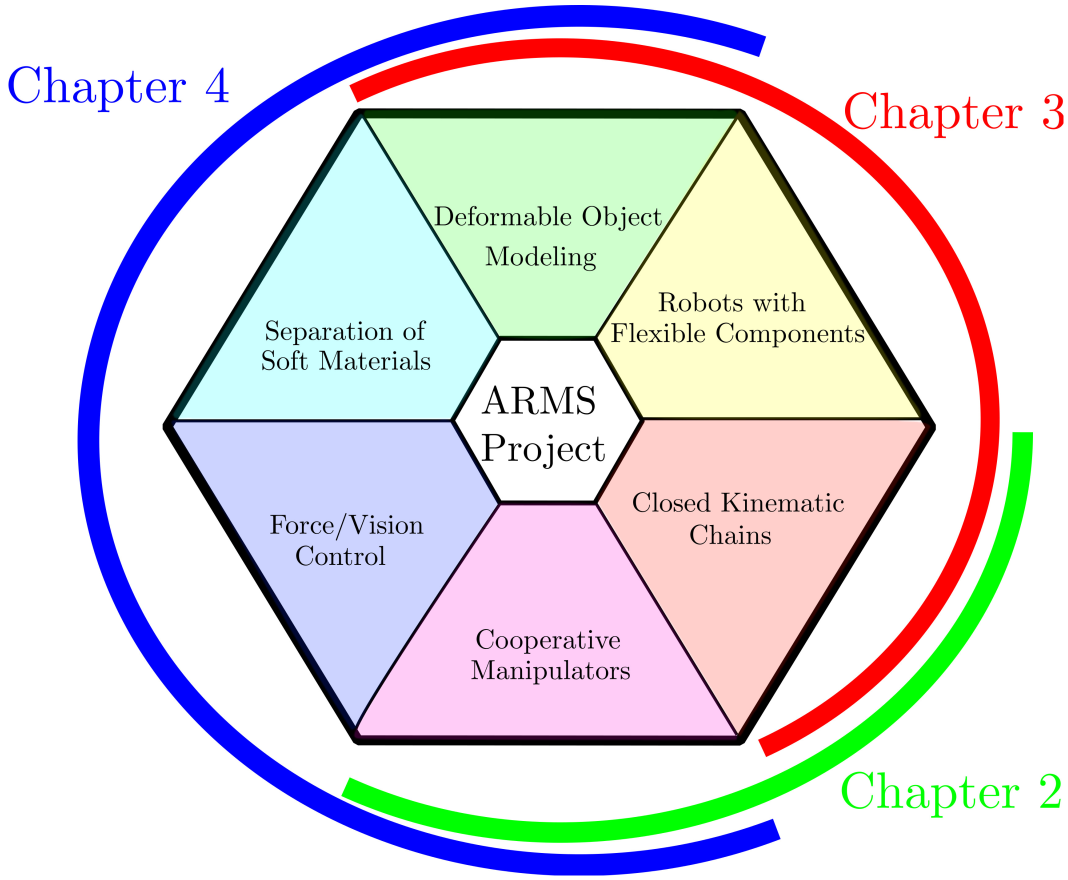

Figure 1.1: Areas of research treated in the thesis

Chapter 2

Chapter 2, *Cooperative Manipulation of Rigid Objects*, deals with the cooperative manipulation of a rigid object by a dual arm system. In this case the object is firmly grasped by both manipulators. Therefore the system forms a kinematic closed chain, consisting of the manipulator arms, the object and the ground. Chapter 2 is divided into two principal sections.

In Section 2.2, the state of the art of cooperative manipulation is given. This section gives an overview of task based formulations for object and internal force control. Furthermore the different control schemes associated with these formulations are detailed. Finally, lower mobility cooperative manipulators are examined. These systems consist of serial robotic arms of less than 6 degrees of freedom, meaning that the object behavior is complex and the controllable directions are unknown without complex studies. There are numerous ways to analyze such systems.

In Section 2.3, the kinematic and dynamic analysis of a lower mobility cooperative manipulator is outlined. The system in question comprises two arms of 5-DOF grasping a rigid object. The system is redundantly actuated, however in order to simplify the control strategies, we propose the use of a minimum number of actuators to spatially transport the object. All possible minimum actuation schemes are enumerated. We show how screw theory can be used to investigate and understand complex singular configurations. Furthermore the effects of the chosen actuation scheme on the system's kinematic and dynamic performance is demonstrated.

Chapter 3

Chapter 3, *Cooperative Manipulation of Flexible Objects*, deals with the flexibility in closed chain robot manipulators. This flexibility may be present in the robotic structure for example in the limbs or, in the case of cooperative manipulators, the grasped object may be deformable. There are two principal approaches to dealing with flexibility in robotics. Firstly, if the effects flexibility can be accurately modeled, rigid body techniques can be extended to take them into account in the modeling and control strategies. Secondly, if the flexibility is a priori unknown or too complicated to model, sensor based control strategies must be employed to allow the system to cope with hitherto unknown deformations. Chapter 3 is divided into three principal sections.

In Section 3.2, a literature review of robots whose parts can undergo deformations is described. This chapter gives an outline of the modeling strategies of deformable objects and how these strategies can be implanted into existing robotic formulations. Two different scenarios are presented. Firstly, where the deformation or flexibility is due to use of lightweight materials or induced during trajectories with particularly high acceleration. In this case the aim is generally to follow a rigid body trajectory while aiming to bound the deformation or vibration to an acceptable level. Secondly, where the deformation itself is part of the task definition, typically when the manipulator is grasping a deformable object. In this case, referred to as shape control, the robotic system must change the form or shape of the object such that it reaches a desired shape.

In Section 3.3, a general dynamic model of a set of cooperative manipulators handling a flexible object is derived. The system consists of n robots grasing an object of N degree of flexibility. The grasped object is modeled using distributed flexibilities. By combining the Generalized Newton Euler formulation for flexible bodies with the classical dynamic model of closed chain manipulators, a closed form equation for the dynamic model is obtained. Three case studies are carried out that show the resolution of the dynamic equation for different types of objects. The solutions are validated by a comparison with a commercially available dynamic simulator.

In Section 3.4, the general dynamic model derived in Section 3.3 is applied to the well-known Gough-Stewart manipulator. The platform is modeled as flexible and the legs and base as rigid. A closed form solution is obtained for both the inverse and direct dynamic model. The dynamic models are validated using the commercial simulator.

Chapter 4

Chapter 4, *Force/Vision Control of a Meat Cutting Robotic Cell*, deals with cooperative manipulation of unknown deformable bodies. However in this case the cooperative manipulators do not form an closed chain system. Rather the task, the separation of soft bodies, requires more than one robot in order to be completed. Thus the cooperative behavior is linked to the deformation of the object. Chapter 4 is divided into three principal sections.

In Section 4.2, a literature review of the primary methods required for robotic separation of deformable objects is given. This includes the simulation of cutting deformable objects, force vision control of manipulators and cutting formulations.

In Section 4.3, the modeling and control of a robotic meat cutting cell is outlined. This cell represents the multi-arm system defined in the ARMS project. The modeling strategy is outlined and a new force vision controller is proposed to cope with on-line object deformations. We show how this scheme can change the robot's trajectory despite the lack of the object model.

In Section 4.4, the experimental validation of the simulated cell is carried out. The experimental setup also allows us to propose a new force/vision controller which takes into account the resistive forces at the tool frame. The force controller ensures that global deformations and material *bunching* is avoided. Two separate vision controllers are proposed. We show that the use of an image based controller allows the robot to adapt to 3D surface profiles.

2

Cooperative Manipulation of Rigid Objects

2.1 Introduction

This chapter describes the manipulation of a rigid object using cooperative robots. The robots rigidly grasp the object and thus a closed chain is formed comprising the robot's arms, the object and the ground. The properties of the system depend on the architecture of the serial robots. For example, if an object is grasped by 2 serial arms, each of 6-DOF, it is immediately apparent that the object has 6-DOF whereas the two arms have a total of 12-DOF. The redundant actuators can be used to apply forces to the object or simply satisfy the closed loop constraint equations. A review of the kineostatics that demonstrate these relations and an outline of the controllers generally used, is given in this chapter.

The most frequent cases of cooperative robots involve either two 6-DOF robots spatially manipulating an object, or two 3-DOF robots working in a plane. In both cases, the complexities due to working in a subspace are avoided. These robots are known as full mobility systems [Tan08]. On the other hand, if the manipulators do not have 6-DOF and do not belong to a subgroup of the displacement group of rigid body motions, the situation becomes more complicated. Such cases are referred to as lower mobility formulations [Gog10] and, in terms of analytical techniques, have much common with parallel manipulators and closed chain robots. A review of such analytical techniques is given.

This chapter, excluding the introduction, is divided into three sections, the state of the art given in Section 2.2, the contributions of this thesis to this field given in Section 2.3 and finally, in Section 2.4 a conclusion to the chapter is given. The outline of main sections is given in the following.

State of the Art

Firstly, in Section 2.2, the principal methods of modeling, control and analysis of cooperative systems are outlined. In Section 2.2.1 the modeling strategies for cooperative manipulation are outlined. The kineostatics associated with an object grasped at several points is demonstrated. Kinematic and dynamic formulations naturally follow this derivation for full mobility systems. An alternative way of viewing the system, as a redundantly actuated closed chain mechanism, is outlined in Section 2.2.2. In this case the constraints due to common object manipulation are considered in the configuration space. In Section 2.2.3, the two primary classes of control schemes, hybrid position/force and impedance control, for cooperative manipulation are described. In Section 2.2.4, the formulations for lower mobility cooperative manipulators are examined. Furthermore, this section includes a review of the techniques used to analyze lower mobility parallel manipulators.

Contribution

Secondly, in Section 2.3 the contribution to this field, the modeling, kinematic and dynamic analysis of a system with two cooperative manipulators working together on a common task is outlined. The kinematic analysis is a crucial tool to understand the object behavior when grasped by multiple serial arms. Potential applications are workspace analysis, singularity avoidance and the selection of grasping locations of a common object. The dynamic analysis has applications in design and advanced controllers.

In Section 2.3.1 the dual-arm system of the humanoid robot Nao, where the serial architecture of each arm has five degrees of freedom, is described. The stiffness of some motors can be reduced until they behave as passive joints. Certain joints are then chosen as actuated (independent) and the others as passive (dependent). The passive or dependent actuators adapt values to fulfill the constraint equations ensuring the closure of the loop. The advantage of using minimum actuators is twofold. Firstly they lead to a simpler control scheme, since there are less variables to control. Secondly by using passive joints, antagonistic forces on the object due to poor trajectory tracking can be avoided [LKC12a]. The system is modeled as a closed chain mechanism. The mobility of the closed-loop system is analyzed in Section 2.3.2. Screw theory is used to analyze the system's mobility, singularities and motion type [LKC12b]. The benefit of this approach is that special configurations such as the loss of stiffness, loss of DOF etc., can be determined without the complex derivation of the Jacobian matrices (or their inverses). Furthermore the analysis of the constraint screws give a greater insight into the reasons for singular configurations and allows us to discover the link between the actuation scheme and the system performance. In Section 2.3.3, a list of all possible minimum actuation schemes for the robot is given. The inadmissible actuation schemes, the reasons behind and the effects of this inadmissibility, are illustrated and explained in geometrical terms. The nature of the possible motions are explored in Section 2.3.4. The serial and parallel singularities associated with the minimally actuated system are outlined in Section 2.3.5 and compared with the classical methods in Section 2.3.6. The robot configuration, object DOF and task specification drive the choice of independent actuators. However it is possible to select an actuation scheme that has a better overall performance. In order to do so, the kinematic analysis and dynamic performance are considered. In Section 2.3.7 the dynamic performance over a large number of trajectories is tested for each actuation scheme. A criterion related to the total power of the motors is used as a comparison tool. Each trajectory that violates motor capacities are noted and used as a secondary selection criterion. In Section 2.3.8, a suitable actuation scheme is chosen from the aforementioned kinematic and dynamic analysis [LKC14].

2.2 State of the Art: Cooperative Manipulation

2.2.1 Cooperative Robot Formulations

Object Kineostatics

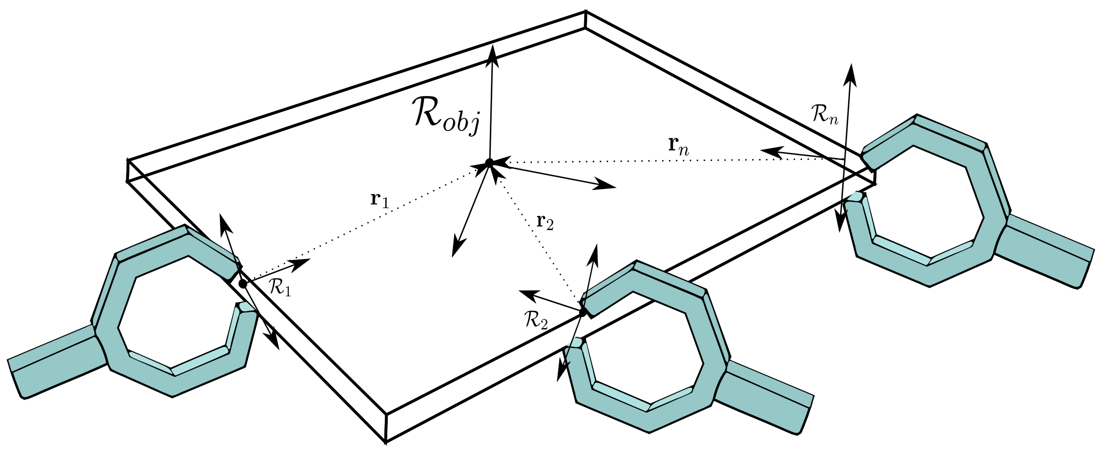

Figure 2.1: Object grasped by n manipulators

When transporting an object in space a cooperative robotic system grasps the object at several different locations. At each location the arm applies a wrench to the object. In order to analyze the behavior of the object a supplementary frame $\mathcal{R}_{obj}$ is introduced and fixed to the grasped object, as shown in Fig. 2.1. The position vector from the ith end effector to $\mathcal{R}_{obj}$ is denoted as $\mathbf{r}_i$. An end effector i applies a wrench $\mathbf{h}_i$ on the object. By assuming a rigid object, the *motion-causing* wrench at the object frame due to the ith end effector can be obtained using the 6×6 screw transformation matrix, ${}^{i}\mathbf{S}_{obj}^{T}$:

$$\mathbf{h}_{obj} = {}^{i}\mathbf{S}_{obj}^{T}\mathbf{h}_i \tag{2.1}$$

A wrench $\mathbf{h}$ is a 6×1 vector consisting of a pure force $\mathbf{f}$ and a pure moment $\mathbf{n}$, thus (2.1) becomes:

$$\begin{bmatrix} \mathbf{f}_{obj} \\ \mathbf{n}_{obj} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 \\ -\hat{\mathbf{r}}_i & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \mathbf{f}_i \\ \mathbf{n}_i \end{bmatrix} \tag{2.2}$$

If the object is gripped simultaneously by n end effectors (2.2) is rewritten as:

$$\begin{bmatrix} \mathbf{f}_{obj} \\ \mathbf{n}_{obj} \end{bmatrix} = \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 & \mathbf{I}_3 & \mathbf{0}_3 & \dots & \mathbf{I}_3 & \mathbf{0}_3 \\ -\hat{\mathbf{r}}_1 & \mathbf{I}_3 & -\hat{\mathbf{r}}_2 & \mathbf{I}_3 & \dots & -\hat{\mathbf{r}}_n & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{n}_1 \\ \mathbf{f}_2 \\ \mathbf{n}_2 \\ \vdots \\ \mathbf{f}_n \\ \mathbf{n}_n \end{bmatrix} = \mathbf{W} \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{n}_1 \\ \mathbf{f}_2 \\ \mathbf{n}_2 \\ \vdots \\ \mathbf{f}_n \\ \mathbf{n}_n \end{bmatrix} \tag{2.3}$$

For a rigid object, $\mathbf{W}$ is a $6 \times 6n$ non-square matrix of maximum rank 6 and is referred to as the grasp matrix [CU08]. There exists a null space of this matrix whose dimension is determined by the number of manipulators grasping the object. In order to obtain the manipulator forces from the object forces, (2.3) must be inverted using a generalized inverse of $\mathbf{W}$. Due to the existence of the null space, a new term $\mathbf{h}_{int} = \begin{bmatrix} \mathbf{f}_{int} \\ \mathbf{n}_{int} \end{bmatrix}$ appears in (2.4):

$$\begin{bmatrix} \mathbf{f}_1 \\ \mathbf{n}_1 \\ \mathbf{f}_2 \\ \mathbf{n}_2 \\ \vdots \\ \mathbf{f}_n \\ \mathbf{n}_n \end{bmatrix} = \mathbf{W}^{(-1)} \begin{bmatrix} \mathbf{f}_{obj} \\ \mathbf{n}_{obj} \end{bmatrix} + \mathbf{\Lambda} \begin{bmatrix} \mathbf{f}_{int} \\ \mathbf{n}_{int} \end{bmatrix} \tag{2.4}$$

Equation (2.4) demonstrates how the system can be partitioned into a motion causing wrench, denoted as $\mathbf{h}_{obj}$, and a so-called internal wrench denoted as $\mathbf{h}_{int}$. The internal wrench generates stresses in the object which may be used for grasping purposes or for other tasks. The internal wrench is projected into the null space of the motion causing wrench and thus has no effect on the motion of the object.

However the inverse of this statement, that the motion causing wrench has no effect on the internal loading of the object, is not true in some cases. In [WFM89], it is demonstrated that an arbitrary choice of $\mathbf{W}^{(-1)}$ may not be sufficient to completely partition the system. Referring to (2.3), it can be seen that due to the shape of $\mathbf{W}$, there are infinite solutions to its inversion. A poor choice of $\mathbf{W}^{(-1)}$ means that $\mathbf{h}_{obj}$ will generate internal object loading. Instead, the choice of $\mathbf{W}^{(-1)}$, given in (2.5), decouples the system and ensures the internal forces can be completely determined by $\mathbf{\Lambda}$, the null space term.

$$\mathbf{W}^{(-1)} = \frac{1}{n} \cdot \begin{bmatrix} \mathbf{I}_3 & \mathbf{0}_3 \\ \hat{\mathbf{r}}_1 & \mathbf{I}_3 \\ \mathbf{I}_3 & \mathbf{0}_3 \\ \hat{\mathbf{r}}_2 & \mathbf{I}_3 \\ \vdots & \vdots \\ \mathbf{I}_3 & \mathbf{0}_3 \\ \hat{\mathbf{r}}_n & \mathbf{I}_3 \end{bmatrix} \tag{2.5}$$

On the other hand, the choice of $\mathbf{\Lambda}$ is not unique. Indeed, $\mathbf{\Lambda}$ can be chosen as *any* matrix whose columns span the null space of $\mathbf{W}$ i.e. $\mathbf{W\Lambda} = 0$. One particular example of $\mathbf{\Lambda}$, which shows the properties of the projection, is given in (2.6). It can be demonstrated [BH96] that $\mathbf{\Lambda}$ is independent of the object frame. This is intuitive, since the arbitrary nature of the location of the object frame should not affect the internal loading. Instead $\mathbf{\Lambda}$ is dependent on the relative location between the end effectors that grasp the object.

$$\mathbf{\Lambda} = \begin{bmatrix} -\mathbf{I}_3 & \mathbf{0}_3 \\ \hat{\mathbf{r}}_2 - \hat{\mathbf{r}}_1 & \mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 \\ \hat{\mathbf{r}}_3 - \hat{\mathbf{r}}_2 & -\mathbf{I}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 \\ \hat{\mathbf{r}}_4 - \hat{\mathbf{r}}_3 & \mathbf{I}_3 \\ \vdots & \vdots \\ \hat{\mathbf{r}}_n - \hat{\mathbf{r}}_{n-1} & -\mathbf{I}_3 \\ \mathbf{I}_3 & \mathbf{0}_3 \\ \mathbf{0}_3 & \mathbf{0}_3 \end{bmatrix} \tag{2.6}$$

Finally, by rewriting (2.4), for any set of end effector forces and moments, the motion causing forces at an arbitrarily chosen object frame, and the internal force experienced by the object can be obtained from:

$$\begin{bmatrix} \mathbf{f}_{obj} \\ \mathbf{n}_{obj} \\ \mathbf{f}_{int} \\ \mathbf{n}_{int} \end{bmatrix} = \begin{bmatrix} \mathbf{W} \\ \mathbf{\Lambda}^{+} \end{bmatrix} \begin{bmatrix} \mathbf{f}_1 \\ \mathbf{n}_1 \\ \mathbf{f}_2 \\ \mathbf{n}_2 \\ \vdots \\ \mathbf{f}_n \\ \mathbf{n}_n \end{bmatrix} \tag{2.7}$$

If the cooperative system consists of two robots each of 6-DOF, the grasp matrix has rank 6 and a null space has dimension 6 (similarly in a planar case rank($\mathbf{W}$)=rank($\mathbf{\Lambda}$)=3). This particular system is fully actuated, but not redundantly actuated, meaning there are exactly the same amount of variables that need to be controlled as actuators in the system (12 or 6 for spatial and planar cases respectively). This feature, labeled here full mobility cooperative manipulation is a very attractive feature and thus in cooperative manipulation much research has been focused on such systems.

Kinematic Formulations

Using the principle of virtual work, velocity equations analogous to (2.4) can be defined. In doing so, a set of kinematic equations, that fully characterizes the system, is derived. The formulation is written in terms of the object velocity $\mathbf{V}_{obj}$ and the internal velocity $\mathbf{V}_{int}$.

It is obvious that the internal velocity can not be used to actively control the object's internal loading. Instead this variable is akin to a set of constraint equations that ensure a safe object grasp. In some cases however, the object may undergo infinitesimal deformations meaning that the internal velocity $\mathbf{V}_{int}$ is related to the internal forces via a stiffness relationship that is dependent of the material properties of the object. Otherwise if the objects are flexible the internal velocity variables can be used to control this flexibility.

In order to model the relationship between the joint variables and the task velocities, the Cartesian velocities of the end effectors must be introduced. For a dual manipulator system with $i = r, l$ denoting the right and the left arms respectively:

$$\mathbf{V}_r = \mathbf{J}_r(\mathbf{q}_r)\dot{\mathbf{q}}_r \qquad \mathbf{V}_l = \mathbf{J}_l(\mathbf{q}_l)\dot{\mathbf{q}}_l \tag{2.8}$$

where $\mathbf{q}_r$, $\mathbf{q}_l$, $\dot{\mathbf{q}}_r$ and $\dot{\mathbf{q}}_l$ denotes the vector of joint positions and velocities of the right and left manipulator respectively. $\mathbf{V}_r$ and $\mathbf{V}_r$ denotes the tool velocities of the right and left manipulators. $\mathbf{J}_r$ and $\mathbf{J}_l$ are the kinematic Jacobian matrices associated with the right and left arm respectively.

In [UD88], a formulation is derived using the concept of virtual sticks. These sticks are modeled as rigid vectors extending from the end effectors to a predefined object frame, for example the center of mass of the object. The end effector velocities are transformed to this frame by the screw transformation matrix:

$$\mathbf{V}_{rs} = {}^{obj}\,\mathbf{S}_r \mathbf{V}_r \qquad\qquad \mathbf{V}_{ls} = {}^{obj}\,\mathbf{S}_l \mathbf{V}_l \tag{2.9}$$

The total velocity at the object frame is then calculated as:

$$\mathbf{V}_{obj} = \frac{1}{2}\left(\mathbf{V}_{rs} + \mathbf{V}_{ls}\right) \tag{2.10}$$

The relative velocity in this case is defined as

$$\mathbf{V}_{int} = \mathbf{V}_{rs} - \mathbf{V}_{ls} \tag{2.11}$$

This permits the partition of the object space into external (motion causing) and internal (force generating) variables for a defined object frame. The system can be rewritten using (2.8), (2.10) and (2.11) as:

$$\begin{bmatrix} \mathbf{V}_{obj} \\ \mathbf{V}_{int} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\,{}^{obj}\mathbf{S}_r\mathbf{J}_r & \frac{1}{2}\,{}^{obj}\mathbf{S}_l\mathbf{J}_l \\ {}^{obj}\mathbf{S}_r\mathbf{J}_r & -{}^{obj}\mathbf{S}_l\mathbf{J}_l \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_r \\ \dot{\mathbf{q}}_l \end{bmatrix} \tag{2.12}$$

The disadvantage of (2.12) is the assumption of a constant vector from the end effectors to the frame of interest. In order to improve upon this formulation, in [CCS96], the concept of the cooperative task space is introduced. The idea is that instead of modeling the system about a fixed frame, the manipulators' grasp position defines the controlled object frame. Although similar to virtual stick formulation given in (2.12), the cooperative task space eliminates the requirement of a known frame on the object leading to a more robust method of modeling dual arm systems. Furthermore the object in question may be articulated or deformable, where the extra DOF can be controlled by $\mathbf{V}_{int}$. Equation (2.12) is rewritten as:

$$\begin{bmatrix} \mathbf{V}_{obj} \\ \mathbf{V}_{int} \end{bmatrix} = \begin{bmatrix} \frac{1}{2}\mathbf{J}_r & \frac{1}{2}\mathbf{J}_l \\ \mathbf{J}_r & -\mathbf{J}_l \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_r \\ \dot{\mathbf{q}}_l \end{bmatrix} \tag{2.13}$$

There are several different representations of the task space variables for cooperative manipulators. For example using quaternions to represent orientation [CCC00] eliminates representational singularities, alternatively using dual quaternions to represent both position and orientation reduces the number of required equations [AFD10]. By using (2.13), an inverse kinematics controller can be implemented where joint velocities are obtained for desired task variables.

Dynamic Formulation

In this section, cooperative manipulation considering the system's dynamics is described. Generally the mass properties of the grasped object are not well known. However by measuring the force sensed by the manipulators at the grasp location, the effects of the object's dynamics can be taken into account. Furthermore internal forces and moments can be applied to the object. The two primary schemes using force control are extensions of those for single manipulator systems, namely hybrid position/force control [RC80], and impedance control [Hog85]. These schemes are described in the following sections. Firstly, the dynamic model of the cooperative system is described. The inverse dynamic model for the i^{th} robot is given as:

$$\boldsymbol{\tau}_i = \mathbf{A}_i \ddot{\mathbf{q}}_i + \mathbf{c}_i + \mathbf{J}_i^T \mathbf{h}_i \tag{2.14}$$

$\boldsymbol{\tau}_i$ is the vector of joint torques or forces. $\mathbf{A}_i$ is the symmetric positive definite inertia matrix. $\ddot{\mathbf{q}}_i$ denotes the vector of joint accelerations. $\mathbf{c}_i$ is the vector of Coriolis, centrifugal and gravity torques. Equation (2.14) can also be extended to include all manipulators:

$$\mathbf{A} \begin{bmatrix} \ddot{\mathbf{q}}_1 \\ \vdots \\ \ddot{\mathbf{q}}_n \end{bmatrix} + \begin{bmatrix} \mathbf{c}_1 \\ \vdots \\ \mathbf{c}_n \end{bmatrix} + \mathbf{J}^T \begin{bmatrix} \mathbf{h}_1 \\ \vdots \\ \mathbf{h}_n \end{bmatrix} = \begin{bmatrix} \boldsymbol{\tau}_1 \\ \vdots \\ \boldsymbol{\tau}_n \end{bmatrix} \tag{2.15}$$

where

$$\mathbf{A} = \begin{bmatrix} \mathbf{A}_1 & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \mathbf{A}_n \end{bmatrix}, \quad \mathbf{J}^T = \begin{bmatrix} \mathbf{J}_1^T & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \mathbf{J}_n^T \end{bmatrix} \tag{2.16}$$

Alternatively the dynamic equation can transformed into task space as follows:

$$\ddot{\mathbf{q}}_i = \mathbf{J}_i^{-1} \cdot (\dot{\mathbf{V}}_i - \dot{\mathbf{J}}_i \cdot \dot{\mathbf{q}}_i) \tag{2.17}$$

$$\Rightarrow \boldsymbol{\tau}_i = \mathbf{A}_i \left(\mathbf{J}_i^{-1} \cdot (\dot{\mathbf{V}}_i - \dot{\mathbf{J}}_i \cdot \dot{\mathbf{q}}_i)\right) + \mathbf{c}_i + \mathbf{J}^T \mathbf{h}_i \tag{2.18}$$

The object dynamics also contribute to the calculation of the motor torques. The force at the object frame is obtained from the dynamics of the object. This force can be calculated using the Newton-Euler equations:

$$\mathbf{h}_{obj} = \mathbf{M}_{obj} \dot{\mathbf{V}}_{obj} + \mathbf{B}_{obj} \left(\mathbf{x}_{obj}, \mathbf{V}_{obj}\right) + \mathbf{h}_e \tag{2.19}$$

From (2.1), it has been shown how the object forces are related to the applied end effector forces:

$$\mathbf{W} \begin{bmatrix} \mathbf{h}_1 \\ \vdots \\ \mathbf{h}_n \end{bmatrix} = \mathbf{M}_{obj} \dot{\mathbf{V}}_{obj} + \mathbf{B}_{obj} \left(\mathbf{x}_{obj}, \mathbf{V}_{obj}\right) + \mathbf{h}_e \tag{2.20}$$

$\mathbf{M}_{obj}$ is the inertia matrix of the object and $\mathbf{B}_{obj}$ represents the Coriolis, centrifugal and gravity torques. $\mathbf{h}_e$ denotes the wrench due to interaction with the environment. Finally from (2.4), (2.15) and (2.20), a complete dynamic description of the system is given as:

$$\begin{bmatrix} \boldsymbol{\tau}_1 \\ \vdots \\ \boldsymbol{\tau}_n \end{bmatrix} = \mathbf{J}^T[\mathbf{W}^+ (\mathbf{h}_{obj}) + \boldsymbol{\Lambda} (\mathbf{h}_{int})] + \mathbf{A} \begin{bmatrix} \ddot{\mathbf{q}}_1 \\ \vdots \\ \ddot{\mathbf{q}}_n \end{bmatrix} + \begin{bmatrix} \mathbf{c}_1 \\ \vdots \\ \mathbf{c}_n \end{bmatrix} \tag{2.21}$$

2.2.2 Closed Chain Representation

Since the manipulators are rigidly constrained the system can be viewed as a closed chain system. In this case the system becomes a redundantly actuated parallel manipulator, and as such the same techniques can be used for modeling and analysis. The redundant actuators can be exploited to optimize secondary criteria as shown in 2.2.2. Alternatively the redundant actuators can behave as passive joints and ensure the mechanism respects the closed chain constraints. The advantage of the passive joint approach is a simplification of the control scheme since there are less variables to be controlled. Furthermore the formulation can be used for different types of closed chain mechanisms, where the passive joints represent task conditions or grasping constraints [YSYO99, LXB99, CYL03, ÖÖ01]. A general outline for closed chain approaches is given in the following sections.

Closed Chain Kinematics

The closed chain is cut at a convenient point leading to an equivalent tree-structure robot [KK86]. We suppose that in this simple example the robot is a single closed loop mechanism as shown in Fig.2.2. Then the equivalent tree-structure robot comprises two serial robots that share a base frame and whose terminal frames are $\mathcal{R}_r$ and $\mathcal{R}_l$ respectively. $\mathcal{R}_r$ and $\mathcal{R}_l$ are coincident however each frame is fixed on a different branch of the tree-structure robot. The location of each frame is a function of the joint variables of the serial chain, the location of $\mathcal{R}_r$ is a function of $\mathbf{q}_r$ and the location of $\mathcal{R}_l$ is a function of $\mathbf{q}_l$. The geometric constraints of the robot are given as:

$$^0\mathbf{T}_1 \ldots {}^{l-2}\mathbf{T}_{l-1} \, {}^{l-1}\mathbf{T}_l = ^0 \mathbf{T}_{l+1} \ldots {}^{r-2}\mathbf{T}_{r-1} \, {}^{r-1}\mathbf{T}_r \tag{2.22}$$

which is reduced to:

$$^0\mathbf{T}_l = ^0 \mathbf{T}_r \tag{2.23}$$

The velocity of the terminal frames must also be equal (assuming the object is rigid). Hence the kinematic constraints are written as:

$$^0\mathbf{V}_l = ^0 \mathbf{V}_r \tag{2.24}$$

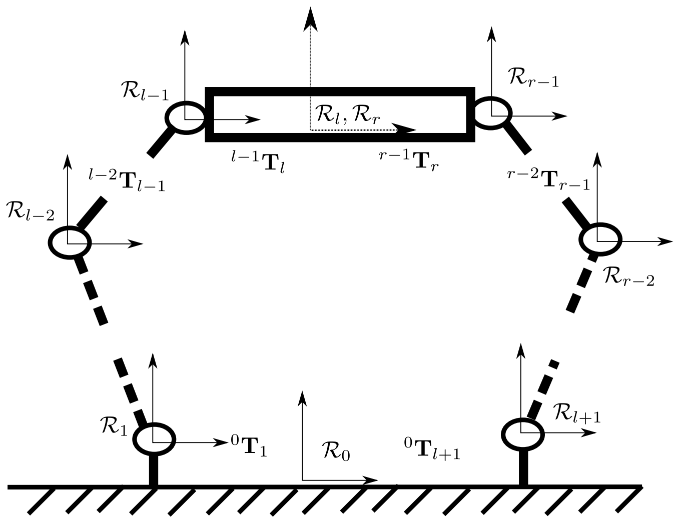

Figure 2.2: Closed Loop Representation

These constraints can be transformed in terms of the joint variables:

$$ {}^0\mathbf{J}_l\dot{\mathbf{q}}_l = {}^0\mathbf{J}_r\dot{\mathbf{q}}_r \tag{2.25}$$

Finally this matrix is rearranged to obtain a relationship between the passive joint velocities and the actuated joint velocities:

$$\mathbf{J}_s \begin{bmatrix} \dot{\mathbf{q}}_a \\ \dot{\mathbf{q}}_p \\ \dot{\mathbf{q}}_c \end{bmatrix} = \mathbf{0} \tag{2.26}$$

where $\mathbf{q}_a$, $\mathbf{q}_p$ and $\mathbf{q}_c$ denote the vectors containing the actuated, passive and cut joints respectively and $\mathbf{J}_s$ is the constraint Jacobian matrix. This matrix allows the calculation of the passive joint velocities that satisfy the closed loop kinematic constraint equations. Furthermore $\mathbf{J}_s$ is widely used to analyze the performance of the system at a given configuration as discussed in Section 2.2.4.

Closed Chain Dynamics

The inverse dynamic model of closed chain robots can be derived in several ways for example in [NG89], in the following the derivation of [KD04] is briefly outlined. In order to find the Closed Loop Inverse Dynamic Model (CLIDM), first the IDM for the tree structure is found and then converted to CLIDM by using the following relation:

$$\begin{bmatrix} \boldsymbol{\tau}_{cl} \\ 0_p \end{bmatrix} = \begin{bmatrix} \boldsymbol{\Gamma}_a \\ \boldsymbol{\Gamma}_p \end{bmatrix} + \left(\frac{\partial\,[\mathbf{q}_{tr}]}{\partial \mathbf{q}_a}\right)^T \boldsymbol{\lambda} \tag{2.27}$$

$\mathbf{\Gamma}_a$, $\mathbf{\Gamma}_p$ are equivalent tree structure joint torques for the actuated and passive joints respectively. $\mathbf{q}_{tr}$ is the vector of joint positions of the tree structured robot comprising $\mathbf{q}_a$ the actuated joint positions and $\mathbf{q}_p$ the passive joint positions. $\boldsymbol{\lambda}$ is vector of constraint forces at the cut joint required for loop closure. Since the passive joints are unable to apply torque, the value for the constraint forces can be calculated from (2.27). The constraint forces are transformed into the configuration space using the Jacobian matrix $\left(\frac{\partial\,[\mathbf{q}_{tr}]}{\partial \mathbf{q}_a}\right)^T$. By solving this expression and substituting back into (2.27), the CLIDM is written as:

$$\boldsymbol{\tau}_{cl} = \begin{bmatrix} \left(\frac{\partial \mathbf{q}_a}{\partial \mathbf{q}_a}\right)^T & \left(\frac{\partial \mathbf{q}_p}{\partial \mathbf{q}_a}\right)^T \end{bmatrix} \begin{bmatrix} \mathbf{\Gamma}_a \\ \mathbf{\Gamma}_p \end{bmatrix} = \begin{bmatrix} \mathbf{I}_a & \left(\frac{\partial \mathbf{q}_p}{\partial \mathbf{q}_a}\right)^T \end{bmatrix} \begin{bmatrix} \mathbf{\Gamma}_a \\ \mathbf{\Gamma}_p \end{bmatrix} \tag{2.28}$$

It should be noted that in this case the mass properties of the object must be known since the object is simply treated as another link in the closed chain system. This means that a change in grasp conditions, for example a re-positioning of an end effector with respect to the object frame, would require a new dynamic model.

Actuator Redundancy in Closed Chain Systems

Actuator redundancy occurs in parallel manipulators when the closed chain is actuated by more joints that the mobility of the structure. Similarly when the cooperative manipulators firmly grasp an object an over-actuated closed chain system is formed. Actuator redundancy is also present when an end effector grasps an object with several fingers.

Redundant actuation in parallel manipulators has several advantageous over conventional actuation schemes [CLY+01], it is obvious that cooperative manipulators share the same benefits.

For instance, the supplementary actuators can be used to optimize a secondary criteria while ensuring the closed chain constraints are respected. Redundant actuation has been used in the minimization of the driving torques for improved efficiency or in order to respect actuator constraints [WWWL09]. By using the redundant actuators the system can distribute the load according to manipulator capabilities [ZL88], [NA89] and optimize the resultant force [SRPD05, WFM89]. Other possible uses of redundant actuators are the modification of the end effector stiffness [Mul06], creation of frequency modulations [YOS99] that could be useful in assembly operations or eliminating the effects of backlash[Mul05].

To take in account the redundancy, (2.28) must be changed into:

$$\tau_r = \begin{bmatrix} \left(\frac{\partial \mathbf{q}_r}{\partial \mathbf{q}_a}\right)^T \end{bmatrix}^+ \begin{bmatrix} \mathbf{I}_a & \left(\frac{\partial \mathbf{q}_p}{\partial \mathbf{q}_a}\right)^T \end{bmatrix} \begin{bmatrix} \mathbf{\Gamma}_a \\ \mathbf{\Gamma}_p \end{bmatrix} \tag{2.29}$$

where τ_r and $\mathbf{q}_r$ denote the torque and position vectors of *all* actuated joints respectively.

2.2.3 Cooperative Control Schemes

Hybrid Position/Force

Due to the redundant nature of cooperative manipulators, hybrid position force schemes have been widely utilized in several scenarios. The most common application is in full mobility systems [Hay86] where a set of 6-DOF (or 3 DOF) serial robots must manipulate a common object. However hybrid position force schemes have also been applied to lower mobility systems [YSYO99] and to systems that contain passive joints [TTI06].

The idea of cooperative hybrid position/force control, taken from the single manipulator equivalent [RC80], is to divide the control space into two independent subspaces, such that the directions that are constrained in position are controlled by force and those constrained in force are controlled in position. The control torque input to the robot is the sum of two feedback loops. One loop controls the force variables and the other loop controls the position or velocity variables. Thus distinct position and force subspaces must be maintained.

For example, suppose that the system consists of two 6-DOF manipulators grasping the object. The external variables, that is the object's motion in space, can be controlled in position while the internal variables, representing the internal loading of the object, are controlled in force. In this scenario the cooperative manipulators move the object in 6 dimensional space, at the same time applying a 6 dimensional wrench on the object.

In the special case of dual-arm systems where one manipulator is solely controlled in force (compliantly) while the second manipulator is controlled in position the scheme becomes a master-slave controller [AMK87]. The advantage of this scheme is that only one manipulator needs to be equipped with a costly force sensor.

The hybrid position/force control scheme for dual arm manipulation of an object is shown in Fig.2.3 [UD88]. $\mathbf{x}_d$ and $\mathbf{h}_d$ are the desired variables of position and force respectively. Both vectors are composed of variables describing the interaction with the environment ($\mathbf{x}_{obj}$, $\mathbf{h}_{obj}$) and variables describing internal object state ($\mathbf{x}_{int}$, $\mathbf{h}_{int}$). Therefore in the case of two 6-DOF manipulators grasping an object, both $\mathbf{x}_d$ and $\mathbf{h}_d$ contain twelve elements. $\mathbf{C}$ is a diagonal matrix known as the *compliance selection matrix*. $\mathbf{C} = diag(c_1, c_2, \dots c_{12})$, if $c_i = 1$ then the i^{th} degree of freedom is position controlled whereas if $c_i = 0$ then the i^{th} degree of freedom is force controlled. The position controller is composed of n-DOF while the force controller is composed of $(12 - n)$ DOF.

The advantages and drawbacks of serial variation of hybrid position force schemes are outlined in [PD96]. Hybrid position/force control assumes that position and force

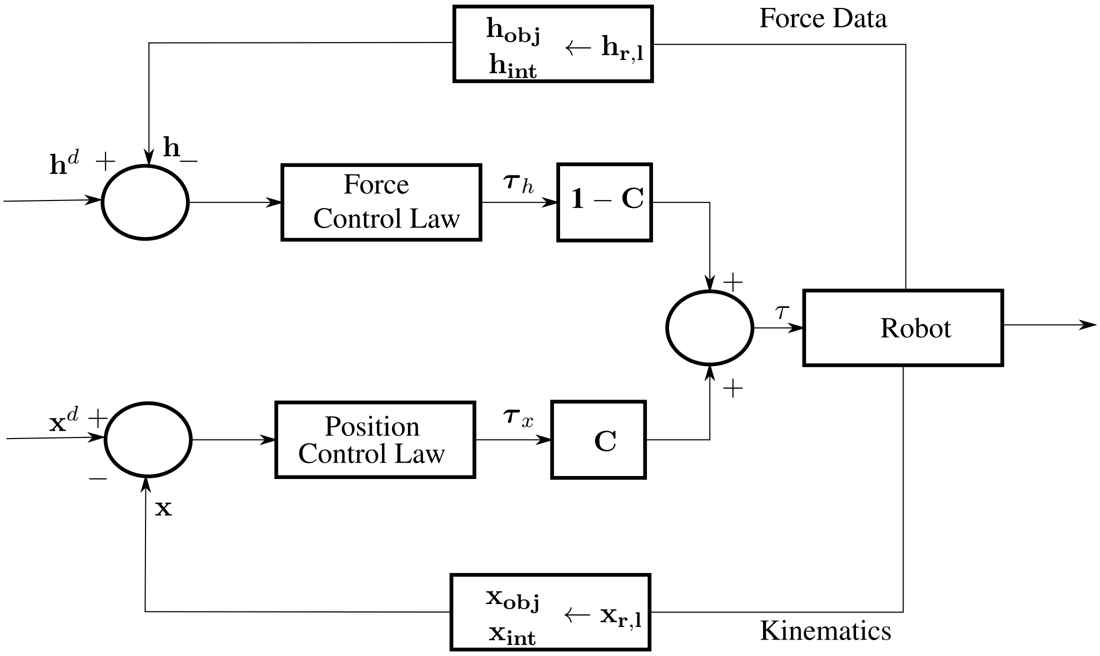

Figure 2.3: Hybrid Position/Force Scheme for dual arm cooperative manipulator

spaces are orthogonal and this property must be maintained throughout the trajectory. This means that the task must be precisely defined within the two spaces. A change in the task would necessitate a change in the switching matrix that could lead to control instability. Furthermore the environment needs to be perfectly known, otherwise there is a chance force errors occur in position controlled directions and vice versa. The errors induced in this case could not be corrected by the control scheme.

Impedance Control

The second fundamental method of force control is impedance control [Hog85]. An impedance controller does not have to maintain separate subspaces for position and force. Rather, a relationship is enforced between them, avoiding to use of a complex switching controller. This relationship, known as the programmable impedance is represented by a mass spring damper system defined by the parameters for inertia, damping and stiffness $\mathbf{M}$, $\mathbf{B}$ and $\mathbf{K}$ respectively, as shown in (2.30). Certain knowledge of the environment is required in order to successfully tune the gains. Moreover some schemes have been known to suffer from local minima when the force term cancels out the motion term.

$$\Delta\mathbf{h} = \mathbf{M}(\dot{\mathbf{V}}_d - \dot{\mathbf{V}}) + \mathbf{B}(\mathbf{V}_d - \mathbf{V}) + \mathbf{K}(\mathbf{x}_d - \mathbf{x}) \tag{2.30}$$

Internal impedance control schemes have been proposed for multi-arm systems to control the internal force of the object by enforcing a relationship between velocity of the end effector and internal force. These schemes have been extended to dual impedance controllers that enforce not only an impedance relation between the end effectors and the internal force but also the object motion and any environmental forces. In the following sections the internal impedance controllers [BH96] and dual impedance schemes [CCMV08, MP10] are described.

Internal Impedance Controllers

When the manipulator imposes a wrench on the grasped object, internal and external forces are created. These can be decomposed according to (2.4). The impedance relation from equation (2.30) is rewritten to enforce a relationship between the internal force error at each manipulator and the velocity of the manipulator.

$$\Delta \mathbf{h}_{int\ i} = \mathbf{M}(\dot{\mathbf{V}}_{di} - \dot{\mathbf{V}}_i) + \mathbf{B}(\mathbf{V}_{di} - \mathbf{V}_i) + \mathbf{K}(\mathbf{V}_{di} - \mathbf{V}_i) \tag{2.31}$$

This equation is solved to find an expression for $\ddot{\mathbf{x}}_i$

$$\dot{\mathbf{V}} = \mathbf{M}_i^{-1} \cdot \left[\mathbf{M}_i \cdot \dot{\mathbf{V}}_{di} + \mathbf{B}_i\left(\mathbf{V}_{di} - \mathbf{V}_i\right) + \mathbf{K}_i\left(\mathbf{x}_{di} - \mathbf{x}_i\right) - \Delta \mathbf{h}_{int\ i}\right] \tag{2.32}$$

Finally by substituting (2.32) into equation (2.18):

$$\tau = \mathbf{A}_i \left\{\mathbf{J}_i^{-1}(\mathbf{M}_i^{-1}\left[\mathbf{M}_i \dot{\mathbf{V}}_{id} + \mathbf{B}_i \Delta \mathbf{V}_i + \mathbf{K}_i \Delta \mathbf{x}_i - \Delta \mathbf{h}_{int\ i}\right] - \dot{\mathbf{J}}_i \cdot \dot{\mathbf{q}}_i)\right\} + \mathbf{c}_i + \mathbf{J}_i^T \mathbf{h}_i \tag{2.33}$$

For a desired position $\mathbf{x}_{di}$ and internal force $\mathbf{h}_{intd}$ the required joint torque can be obtained. Therefore an impedance controller is implemented for each manipulator grasping the object. The closed loop constraints are generated by the grasp geometry, that relates the desired object location to a desired manipulator location.

Dual Impedance Control

The dual impedance scheme shown in Fig.2.5 is proposed in [CV00]. The object must follow a desired trajectory defined in position, velocity and acceleration as $\mathbf{x}_{obj}^{d}$, $\mathbf{V}_{obj}^{d}$, $\dot{\mathbf{V}}_{obj}^{d}$. If an environmental force acts on the object, this trajectory is modified using an impedance relationship, generating a new trajectory $\mathbf{x}_{obj}^{d*}$, $\mathbf{V}_{obj}^{d*}$, $\dot{\mathbf{V}}_{obj}^{d*}$. These variables are used as an input to a second internal impedance controller which is implemented in the same manner as Fig. 2.4.

2.2.4 Lower Mobility Cooperative Systems

A lower mobility cooperative manipulator consists of two arms cooperating on a common task and where the mobility of the arms does not belong to a kinematic displacement subgroup [Her99]. Table 2.1 recalls the 12 displacement Lie subgroups.

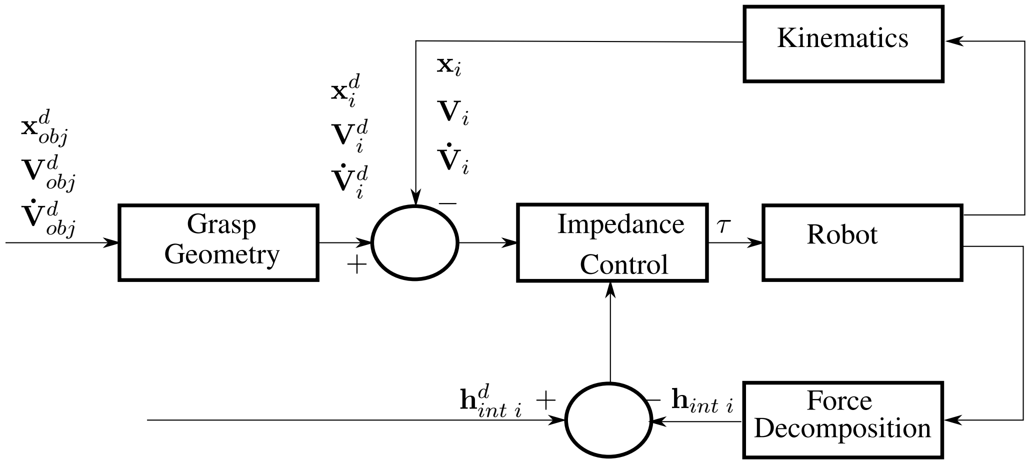

Figure 2.4: Decentralized Internal Impedance

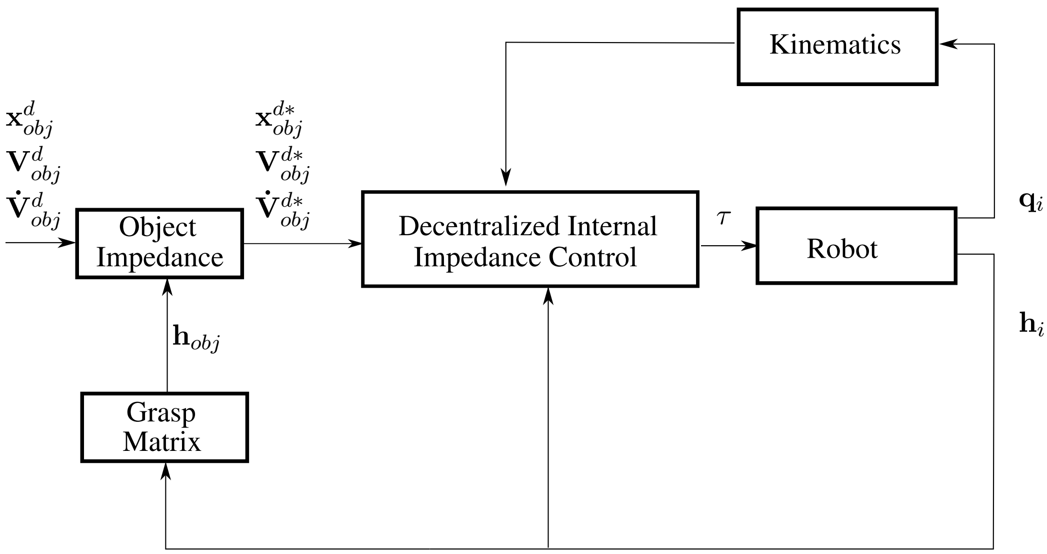

Figure 2.5: Outline of Dual Impedance Control

Table 2.1: List of Displacement Subgroups

No.	DOF	Motion	Possible Joint Configuration
1	0	Zero Motion	
2	1	Translation along axis	P
3		Rotation about axis	R
4		Screw motion about axis	H
5	2	Translation in plane	P-P
6		Cylindrical motion	P-R
7	3	Translation in space	P-P-P
8		Planar Motion	R-R-R
9		Spherical Motion	R-R-R
10		Translation in plane with Screw motion normal to plane	P-P-H
11	4	Schoenflies Motion i.e planar motion with translation normal to plane	R-R-R-P
12	6	Spatial Motion	R-R-R-R-R-R

where R indicates a revolute joint, P indicates a prismatic joint and H indicates a screw or helical joint i.e. a coupled rotation and translation about/along an axis.

If the system consists of an arm whose motion is outside those given in Table 2.1. The task variables can no longer be neatly partitioned into external and internal variables as in (2.13), instead a complex analysis of the effect of each serial robot arm on the grasped object must be carried out.

Therefore to work with lower mobility cooperative manipulators issues such as parallel singularities, closed chain mobility should be considered. Lower mobility systems suffer from three types of singularities, limb (serial) singularities, actuation and constraint (parallel) singularities.

In order to analyze such systems, techniques normally applied to lower mobility parallel manipulators can be used. A review of the the main approaches and formulas used to analyze mechanisms is given in [Gog05].

However cooperative manipulators, which can function as serial robotic arms, are fully actuated, thus a selection of independent joint variables must be made.

Analysis of Lower Mobility Systems by Jacobian Methods

One of the first works to address this problem, is known as Moroskine's method. The mobility of the mechanism can be calculated by using closed loop kinematic constraint equations from (2.26).

$$N = l - rank\left(\ \mathbf{J}_s\ \right) \tag{2.34}$$

where N denotes the mobility of the system and l is the number of joints. The matrix $\mathbf{J}_s$ is denoted as the constraint Jacobian matrix and is usually obtained numerically for a number of random configurations.

In [OW99] singular conditions are defined with respect to the system Jacobian matrices. These matrices are defined similarly to (2.13), except the system is partitioned in actuated and passive joints rather than joints from the left or right manipulator. The velocity of the object is written as:

$$\mathbf{V}_{obj} = \frac{1}{2}\mathbf{J}_t \begin{bmatrix} \dot{\mathbf{q}}_a \\ \dot{\mathbf{q}}_p \end{bmatrix} \tag{2.35}$$

The constraint equations are given:

$$\mathbf{0} = \mathbf{J}_c \begin{bmatrix} \dot{\mathbf{q}}_a \\ \dot{\mathbf{q}}_p \end{bmatrix} \tag{2.36}$$

The Jacobian matrices can be rewritten as:

$$\mathbf{J}_t = \begin{bmatrix} \mathbf{J}_{ta} & \mathbf{J}_{tp} \end{bmatrix} \tag{2.37}$$

$$\mathbf{J}_c = \begin{bmatrix} \mathbf{J}_{ca} & \mathbf{J}_{cp} \end{bmatrix} \tag{2.38}$$

By rearranging the above equations $\mathbf{V}_{obj}$ can be obtained solely in terms of the actuated joint velocity:

$$\mathbf{V}_{obj} = \mathbf{J}_{act}\dot{\mathbf{q}}_a \tag{2.39}$$

$$\mathbf{J}_{act} = \frac{1}{2}\left(\mathbf{J}_{ta} - \mathbf{J}_{tp}\mathbf{J}_{cp}^{+}\mathbf{J}_{ca}\right) \tag{2.40}$$

Thus an analysis of the system can be carried out from (2.37), (2.38), (2.36) (2.39) and (2.40). Firstly, from (2.39) it is obvious that a degeneracy of $\mathbf{J}_{act}$ implies that there exists a value of $\mathbf{V}_{obj}$ which cannot be generated by $\dot{\mathbf{q}}_a$. Therefore, when $\mathbf{J}_{act}$ degenerates the grasped object loses at least 1-DOF. Secondly, from (2.36), it can be seen that:

$$\mathbf{J}_{ca}\dot{\mathbf{q}}_a = \mathbf{J}_{cp}\dot{\mathbf{q}}_p \tag{2.41}$$

If $\mathbf{J}_{cp}$ is not full rank, then the null space of this matrix is not the empty set i.e. $\mathcal{N}\left(\mathbf{J}_{cp}\right) \neq \{\}$. Therefore there exists a value of $\dot{\mathbf{q}}_p$ such that:

$$\mathbf{J}_{cp}\dot{\mathbf{q}}_p = 0 \tag{2.42}$$

Equations (2.41) and (2.42) show that there is a non-zero value of $\dot{\mathbf{q}}_p$, that has no affect on the value of $\dot{\mathbf{q}}_a$. This causes a loss of stiffness in the mechanism. By using (2.35) and (2.37) and assuming that $\dot{\mathbf{q}}_p = \mathcal{N}\left(\mathbf{J}_{cp}\right)$, we obtain the following expression:

$$\mathbf{V}_{obj} = \mathbf{J}_{ta}\dot{\mathbf{q}}_a + \mathbf{J}_{tp}\mathcal{N}\left(\mathbf{J}_{cp}\right) \tag{2.43}$$

The effect of the loss of stiffness on the mechanism can be explained using (2.43). If $\mathbf{J}_{tp} \cdot \mathcal{N}(\mathbf{J}_{cp}) \neq 0$, from (2.42), it can be concluded that there exists a non-zero value of $\dot{\mathbf{q}}_p$ that does not effect the value of $\dot{\mathbf{q}}_a$, yet generates a motion of the object. This corresponds to a loss of stiffness for the grasped object. On the other hand if $\mathbf{J}_{tp} \cdot \mathcal{N}(\mathbf{J}_{cp}) = 0$, there exists a non-zero value of $\dot{\mathbf{q}}_p$ that neither affects the value of $\dot{\mathbf{q}}_a$ nor the motion of the object. This corresponds to a loss of stiffness for a limb in the cooperative robot. In summary, this analysis permits us to calculate and define three singular configurations:

1. Loss of rank of $\mathbf{J}_{act}$ corresponds to a loss of DOF at the object frame
2. $\mathbf{J}_{tp} \cdot \mathcal{N}(\mathbf{J}_{cp}) \neq 0$ corresponds to a loss of stiffness at the object frame
3. $\mathbf{J}_{tp} \cdot \mathcal{N}(\mathbf{J}_{cp}) = 0$ corresponds to a loss of stiffness at one of the limbs

The latter two cases are defined as parallel singularities. In this thesis, parallel singularities are defined as any configuration where the mechanism suffers from a lack of stiffness.

In [BMB95] a thorough examination of lower mobility systems is carried out obtaining the system kinematics in the form of (2.44). By examining the matrix $\begin{bmatrix} \mathbf{C}_1 & \mathbf{C}_2 \end{bmatrix}$, the authors define the following quantities:

Total Mobility: The minimum number of parameters required to fix the location of the object and fix the location of every link

Connectivity: The minimum number of parameters required to fix the location of the object with respect to the base frame

Indeterminacy number: The minimum number of parameters required to fix the location of the object when all the joints are locked

Redundancy number: The minimum number of parameters required to fix the location of every link

Furthermore this analysis allows the formulation of a manipulability ellipsoid as a performance index for closed chain systems.

$$\begin{bmatrix} \mathbf{C}_1 & \mathbf{C}_2 \end{bmatrix} \begin{bmatrix} \mathbf{V}_{obj} \\ \dot{\mathbf{q}} \end{bmatrix} = 0 \tag{2.44}$$

where $\dot{\mathbf{q}}$ is the velocity of all joints.

In [LXB99], the presence of parallel singularities in a cooperative system with passive joints is explored. The Jacobian matrix $\mathbf{J}_{cp}$ from (2.38) is derived for dual arm under-actuated dual arm system. The system is considered singular when this matrix losses rank, however in contrast to [OW99] no distinction is made between a loss of stiffness of the object and a loss of stiffness in one of the manipulator's links.

The issue of a valid selection of actuators, that is a set of actuators that is capable of controlling the grasped object outside special singular configurations, is addressed for two 6-DOF robots in [ÖÖ01]. This analysis is carried out with respect to Jacobian matrices of the system.

In summary, many works have focused on an in depth analysis of the serial Jacobian matrices of the robotic arm in order to examine mobility and special singular configurations. This method is attractive since the serial structure of each manipulator is usually given. On the other hand the results are difficult to generalize since the matrix is dependent both on the joint configuration and the shape of the object that has been grasped.

Analysis of Lower Mobility Systems by non-Jacobian Methods

There exists methods to analyze the mobility of lower mobility mechanisms without using the Jacobian matrices, instead the physical structure of the system is examined. The advantages of these methods are the inherent simplicity and ease of use however they can lead to erroneous inferences and do not provide any supplementary information.

The first method is the known as the Chebychev-Grübler-Kutzbach (CGK) method. The CGK formula is used to determine the mechanism DOF from the number of joints, loops and constraint type. CGK is very easy to use but gives incorrect results for a number of mechanisms. l is the number of joints, c_j is the independent kinematic constraint equations for loop j. b is the number of independent loops j:

$$n = l - \sum_{j=1}^{b} c_j \tag{2.45}$$

To overcome the limitations of the CGK, Gogu [Gog07] proposed a scheme that determines the correct mobility yet does not require the costly calculation of the kinematic constraint equations. Instead the dimension of the operational space of each serial manipulator i, denoted as $dim(\mathbf{E}_i)$, is used. The mobility is given by:

$$l - (dim(\mathbf{E}_1) + dim(\mathbf{E}_2)) + dim(\mathbf{E}_1 \bigcap \mathbf{E}_2) \tag{2.46}$$

In order to find the dimension of the common space (intersection space) of $\mathbf{E}_1$ and $\mathbf{E}_2$, denoted by $dim(\mathbf{E}_1 \bigcap \mathbf{E}_2)$, the operational spaces that can be generated by each serial arm, which *minimize* the intersection are examined. In order to illustrate this idea, we take a simple example. Suppose there are two cooperative manipulators each consisting of two revolute joints $(l = 4)$ situated in the $x{-}y$ plane. Each manipulator can generate a 2-DOF planar motion, $dim(\mathbf{E}_i = 2)$, that is composed of a mixture of the three possible directions i.e. $\mathbf{E}_i \subset \begin{bmatrix} v_x & v_y & \omega_z \end{bmatrix}$. Therefore a minimum intersection is achieved if $\mathbf{E}_1 = \begin{bmatrix} v_x & v_y \end{bmatrix}$ and $\mathbf{E}_2 = \begin{bmatrix} v_x & \omega_z \end{bmatrix}$ meaning $dim(\mathbf{E}_1 \bigcap \mathbf{E}_2) = 1$. Using (2.46), the mobility of the system is given as 1.

Screw Theory

Screw theory is a geometric tool that can be used to analyze the instantaneous motions of complex mechanisms [Hun78, KG07]. In contrast to the classical methods, screw theory can be used to locate and understand parallel singularities in closed chain mechanisms using simple geometric relations [ZBG02]. The advantage is that special configurations can be easily obtained without analyzing complex expressions in the robot's Jacobian matrix. A screw of pitch λ is defined as:

$$\$_\lambda = \begin{bmatrix} \mathbf{s} \times \mathbf{r} + \lambda \mathbf{s} \\ \mathbf{s} \end{bmatrix} \tag{2.47}$$

s a unit vector along the axis of the screw, $\mathbf{r}$ is a vector directed from any point on the axis of the screw to the origin $\mathcal{R}_o$. A zero-pitch screw and an infinite-pitch screw are expressed respectively as follows:

$$\$_0 = \begin{bmatrix} \mathbf{s} \times \mathbf{r} \\ \mathbf{s} \end{bmatrix} \tag{2.48}$$

$$\$_\infty = \begin{bmatrix} \mathbf{s} \\ \mathbf{0}_{3\times 1} \end{bmatrix} \tag{2.49}$$

For every screw system, consisting of *n* linearly independent screws, there exists a reciprocal screw system of dimension $6 - n$. Two screws $\$_1$ and $\$_2$ are reciprocal if their instantaneous power is zero, namely,

$$\left(\begin{bmatrix} \mathbf{0}_{3\times 3} & \mathbf{I}_3 \\ \mathbf{I}_3 & \mathbf{0}_{3\times 3} \end{bmatrix} \$_1 \right)^T \$_2 = 0 \tag{2.50}$$

The following reciprocity conditions can be defined from [KG07]:

1. $\$_0$ is reciprocal to $\$_\infty$ if and only if their axes are orthogonal;
2. $\$_\infty$ is always reciprocal to another $\$_\infty$;
3. Two $\$_0$ are reciprocal if and only if their axes are coplanar (two coplanar axes are either intersecting or parallel);

A zero-pitch twist ν_0 corresponds to a pure rotation about its axis. An ∞-pitch twist ν_∞ corresponds to a pure translation along its direction. A zero-pitch wrench ζ_0 corresponds to a pure force along its axis. An ∞-pitch wrench ζ_∞ corresponds to a pure moment about its direction.

2.3 Kinematic & Dynamic Analysis of a Lower Mobility Cooperative System

In this section, an kinematic and dynamic analysis of a lower mobility cooperative system is carried out. Motion type, singularities and dynamic performance are all evaluated and the results allow us to select an optimum set of minimum actuators.

2.3.1 System Description

The dual-arm system analyzed in this work is shown in Fig. 2.6, while the kinematic architecture of the arms are given in Fig. 2.8. Each arm of the robot has five independent revolute joints. The right arm consists of joints 1-5 and the left arm consists of joints 6-10.

Figure 2.6: Nao T14, (Courtesy of Aldebaran Robotics)

The robotic system is described by the Modified Denavit-Hartenberg (MDH) notation as proposed by Khalil and Kleinfinger [KK86], given in Table 2.2 and modeled using [KC97]. A frame $\mathcal{R}_i$ is fixed on link i such that $\mathbf{z}_i$ is along the joint axis i. $\mathbf{x}_i$ is the common perpendicular between $\mathbf{z}_i$ and one of the succeeding joint axes which are fixed on i. Fig. 2.7 shows a case where frames k and j are attached to link i. A frame $a(j)$ denotes the antecedent of a frame j. Therefore referring to Fig. 2.7, it can be seen that $a(j) = a(k) = i$.

Since $\mathbf{x}_i$ is along the common normal between $\mathbf{z}_i$ and the proceeding frame $\mathbf{z}_j$, four geometric parameters are required to define the transformation matrix ${}^i\mathbf{T}_j$:

$$ {}^i\mathbf{T}_j = rot_x(\alpha_i) \cdot trans_x(d_i) \cdot rot_{zj}(\theta_i) \cdot trans_z(r_i) \tag{2.51} $$

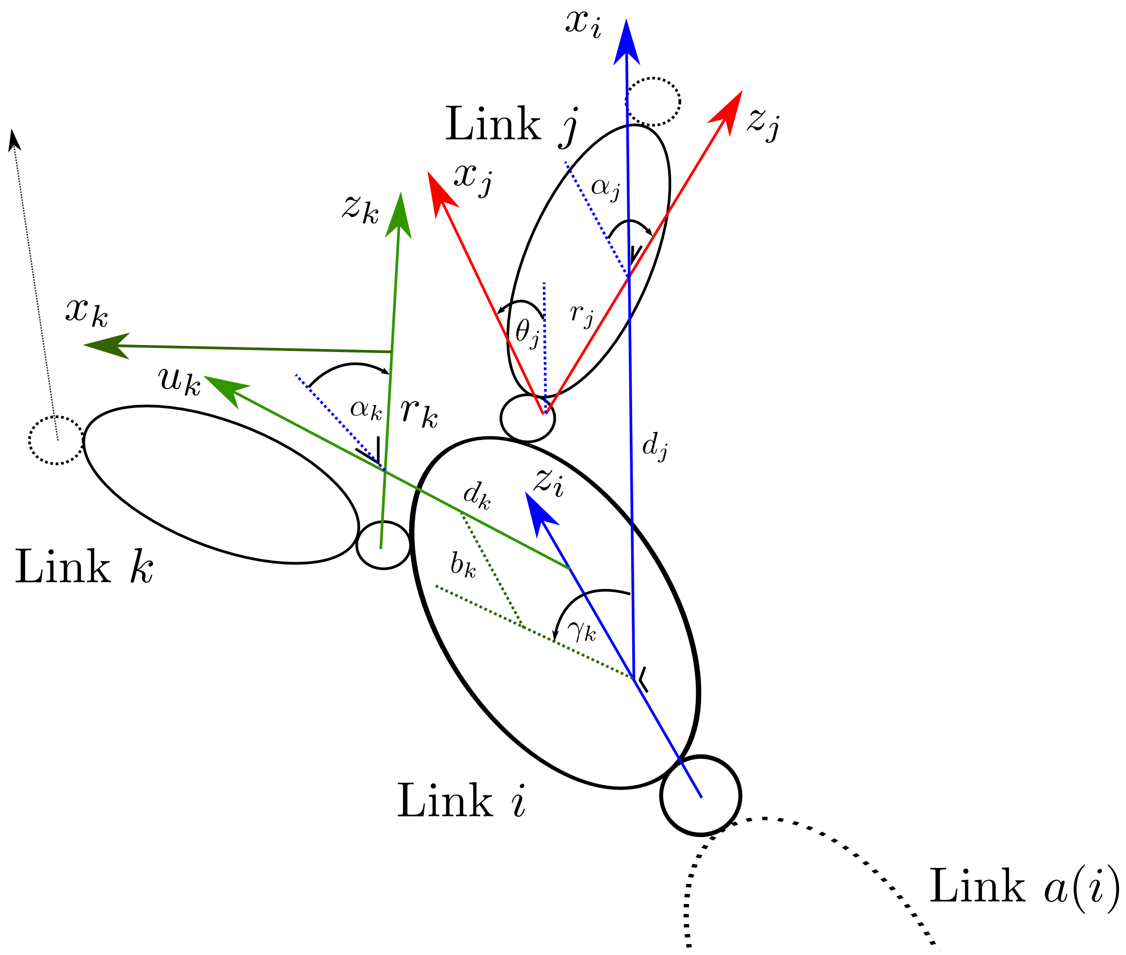

Figure 2.7: Modified Denavit-Hartenberg (MDH) notation [KK86]

where d_j is the distance between $\mathbf{z}_i$ and $\mathbf{z}_j$ along $\mathbf{x}_i$. α_j is the angle between $\mathbf{z}_i$ and $\mathbf{z}_j$ about $\mathbf{x}_i$. θ_j is the angle between $\mathbf{x}_i$ and $\mathbf{x}_j$ about $\mathbf{z}_j$ and r_j is the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ along $\mathbf{z}_j$.

In contrast $\mathbf{x}_i$ is not along the common normal between $\mathbf{z}_i$ and the proceeding frame $\mathbf{z}_k$. Therefore six geometric parameters are required to define the transformation matrix ${}^{i}\mathbf{T}_k$, such that:

$$ {}^{i}\mathbf{T}_k = rot_z(\gamma_k) \cdot trans_z(b_k) \cdot rot_x(\alpha_k) \cdot trans_x(d_k) \cdot rot_{zj}(\theta_k) \cdot trans_z(r_k) \quad (2.52) $$

A common normal $\mathbf{u}_k$ is created between $\mathbf{z}_i$ and the succeeding frame $\mathbf{z}_k$. $\mathbf{u}_k$ is defined by γ_k the angle between $\mathbf{x}_i$ and $\mathbf{u}_k$ about $\mathbf{z}_i$ and b_k the distance between $\mathbf{x}_i$ and $\mathbf{u}_k$ along $\mathbf{z}_i$. Finally it should be noted q_i denotes the joint variable i and is equal to θ_i in the case of a revolute joint or r_i in the case of a prismatic joint.

Once the object is grasped, a closed-loop is formed, as shown in Fig. 2.8. The system

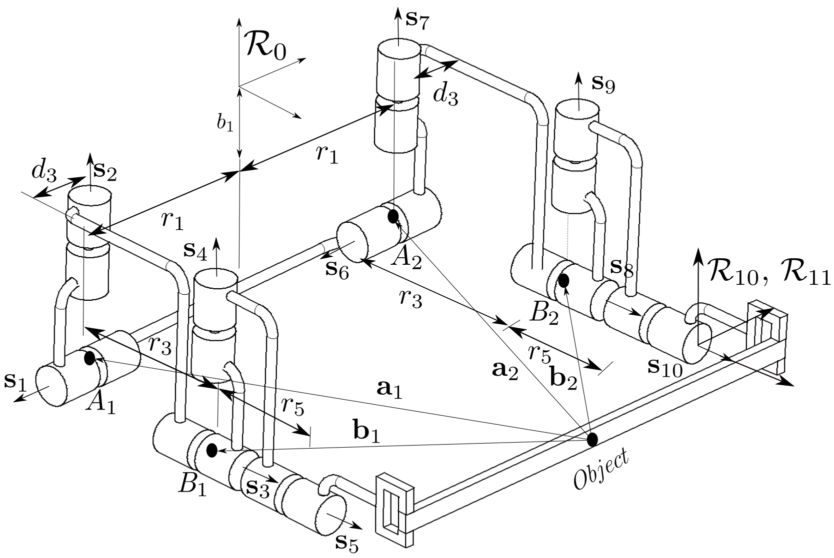

Figure 2.8: Closed-Loop Formulation

has, in this case, only nine bodies. Joint 10 is chosen as the cut joint therefore, to define the equivalent tree structure, link 5 of the closed chain now contains the object, link 5 and link 10. Frame 10 becomes fixed on link 5. We introduce frame 11, which is aligned to frame 10, but its antecedent is frame 5. The parameters of frame 11 are defined once the robot has grasped the object. The locations of frame 10 and frame 11 are equivalent when calculated via either chain. This ensures a constant object grasp throughout the trajectory. The geometric constraint equations are given by:

$$ {}^{0}\mathbf{T}_{1}{}^{1}\mathbf{T}_{2}{}^{2}\mathbf{T}_{3}{}^{3}\mathbf{T}_{4}{}^{4}\mathbf{T}_{5}{}^{5}\mathbf{T}_{11} = {}^{0}\mathbf{T}_{6}{}^{6}\mathbf{T}_{7}{}^{7}\mathbf{T}_{8}{}^{8}\mathbf{T}_{9}{}^{9}\mathbf{T}_{10} \tag{2.53} $$

It should be noted that since the axes of joints 1 and 2 intersect at point A_1, while the axes of joints 3, 4 and 5 intersect at point B_1, the arm of the robot can be represented as a U-joint and a spherical joint serially connected. Therefore in the closed-loop configuration the robot is viewed as a 2-US parallel architecture.

The kinematic constraints are given by:

$$ \begin{bmatrix} {}^{0}\mathbf{v}_{11} \\ {}^{0}\omega_{11} \end{bmatrix} = \begin{bmatrix} {}^{0}\mathbf{v}_{10} \\ {}^{0}\omega_{10} \end{bmatrix} = \begin{bmatrix} {}^{0}\mathbf{v}_{obj} \\ {}^{0}\omega_{obj} \end{bmatrix} \tag{2.54} $$

Table 2.2: MDH Parameters of the closed-loop chain

j	$a(j)$	γ	b	d	α	θ	r
1	0	0	b_1	0	$-\pi/2$	θ_1	-0.098
2	1	0	0	0	$\pi/2$	θ_2	0
3	2	0	0	-0.015	$\pi/2$	θ_3	0.105
4	3	0	0	0	$-\pi/2$	θ_4	0
5	4	0	0	0	$\pi/2$	θ_5	0.05595
6	0	0	b_1	0	$-\pi/2$	θ_6	0.098
7	6	0	0	0	$\pi/2$	θ_7	0
8	7	0	0	0.015	$\pi/2$	θ_8	0.105
9	8	0	0	0	$-\pi/2$	θ_9	0
10	9	0	0	0	$\pi/2$	θ_{10}	0.05595
11	5	γ_{11}	b_{11}	d_{11}	α_{11}	θ_{11}	r_{11}

$$\begin{bmatrix} {}^0\mathbf{v}_{obj} \\ {}^0\omega_{obj} \end{bmatrix} = {}^0\mathbf{J}_{11} \begin{bmatrix} \dot{q}_1 \\ \dot{q}_2 \\ \dot{q}_3 \\ \dot{q}_4 \\ \dot{q}_5 \end{bmatrix} = {}^0\mathbf{J}_{11}\, \dot{\mathbf{q}}_r \tag{2.55}$$

$$\begin{bmatrix} {}^0\mathbf{v}_{obj} \\ {}^0\omega_{obj} \end{bmatrix} = {}^0\mathbf{J}_{10} \begin{bmatrix} \dot{q}_6 \\ \dot{q}_7 \\ \dot{q}_8 \\ \dot{q}_9 \\ \dot{q}_{10} \end{bmatrix} = {}^0\mathbf{J}_{10}\, \dot{\mathbf{q}}_l \tag{2.56}$$

The superscript ^{0} indicates that the variable is represented in the fixed world frame $\mathcal{F}_0$. As frames 10 and 11 are the same, from (2.56) and (2.55):

$${}^0\mathbf{J}_{11}\, \dot{\mathbf{q}}_r = {}^0\mathbf{J}_{10}\, \dot{\mathbf{q}}_l \tag{2.57}$$

or rewritten as

$$\mathbf{J}_s \begin{bmatrix} \dot{\mathbf{q}}_r \\ \dot{\mathbf{q}}_l \end{bmatrix} = 0 \quad \text{where} \quad \mathbf{J}_s = \begin{bmatrix} {}^0\mathbf{J}_{11} & -{}^0\mathbf{J}_{10} \end{bmatrix} \tag{2.58}$$

where $\dot{\mathbf{q}}_r$ and $\dot{\mathbf{q}}_l$ contain the joint velocities of the right arm and the left arm, respectively. ${}^0\mathbf{v}_j$ is the linear velocity and ${}^0\omega_j$ the angular velocity of frame j with respect to frame 0, ${}^0\mathbf{J}_j$ is the 6×5 kinematic Jacobian matrix of frame j with respect to frame 0. By rearranging the rows and columns of (2.57), a relationship is obtained between the passive joint velocities and the actuated joint velocities:

$$\begin{bmatrix} \mathbf{G}_a & \mathbf{G}_p & \mathbf{0} \\ \mathbf{G}_{ac} & \mathbf{G}_{pc} & \mathbf{G}_c \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_a \\ \dot{\mathbf{q}}_p \\ \dot{\mathbf{q}}_c \end{bmatrix} = \mathbf{0} \tag{2.59}$$

$\mathbf{q}_a$, $\mathbf{q}_p$ and $\mathbf{q}_c$ denote the vectors containing the actuated, passive and cut joints respectively. Upon differentiation of (2.59) with respect to time the acceleration constraints equation is expressed as:

$$\begin{bmatrix} \mathbf{G}_a & \mathbf{G}_p & \mathbf{0} \\ \mathbf{G}_{ac} & \mathbf{G}_{pc} & \mathbf{G}_c \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{q}}_a \\ \ddot{\mathbf{q}}_p \\ \ddot{\mathbf{q}}_c \end{bmatrix} + \dot{\mathbf{J}}_s \, \dot{\mathbf{q}} = 0 \tag{2.60}$$

From (2.59), we obtain:

$$\dot{\mathbf{q}}_p = -\mathbf{G}_p^{-1} \, \mathbf{G}_a \, \dot{\mathbf{q}}_a \tag{2.61}$$

$$\dot{\mathbf{q}}_c = -\mathbf{G}_c^{-1} \left(\mathbf{G}_{ac} - \mathbf{G}_{pc} \, \mathbf{G}_p^{-1} \, \mathbf{G}_a \right) \dot{\mathbf{q}}_a \tag{2.62}$$

Furthermore from (2.54), (2.55) and (2.56):

$$\begin{bmatrix} \frac{1}{2}{}^0\mathbf{J}_{10} & \frac{1}{2}{}^0\mathbf{J}_{11} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_r \\ \dot{\mathbf{q}}_l \end{bmatrix} = \begin{bmatrix} {}^0\mathbf{v}_{obj} \\ {}^0\omega_{obj} \end{bmatrix} \tag{2.63}$$

By rearranging the rows and columns of (2.63), a relationship is obtained between the passive joint velocities, the actuated joint velocities and the object twist can be found:

$$\begin{bmatrix} \mathbf{J}_a & \mathbf{J}_p & \mathbf{J}_c \end{bmatrix} \begin{bmatrix} \dot{\mathbf{q}}_a \\ \dot{\mathbf{q}}_p \\ \dot{\mathbf{q}}_c \end{bmatrix} = \begin{bmatrix} {}^0\mathbf{v}_{obj} \\ {}^0\omega_{obj} \end{bmatrix} \tag{2.64}$$

Finally an actuated Jacobian matrix, $\mathbf{J}_{act}$ is derived that defines a relationship between the actuated joint velocities and the object velocity. Using (2.64) and (2.59):

$$\mathbf{J}_{act} \, \dot{\mathbf{q}}_a = \begin{bmatrix} {}^0\mathbf{v}_{obj} \\ {}^0\omega_{obj} \end{bmatrix} \tag{2.65}$$

$$\begin{aligned} \mathbf{J}_{act} = {} & \mathbf{J}_a + \mathbf{J}_p \left(-\mathbf{G}_p^{-1} \, \mathbf{G}_a \right) \\ & + \mathbf{J}_c \left(-\mathbf{G}_c^{-1} \left(\mathbf{G}_{ac} - \mathbf{G}_{pc} \mathbf{G}_p^{-1} \mathbf{G}_a \right) \right) \end{aligned} \tag{2.66}$$

The mobility of the system, given in section 2.3.2, is equal to 4. Hence the dimension of $\mathbf{G}_a$ is 5×4 , $\mathbf{G}_p$ is 5×5 ,$\mathbf{G}_c$ is a scalar that is, due to the modeling procedure, always equal to one, where 4, 5 and 1 are the numbers of active, passive and cut joints, respectively. $\mathbf{G}_p$ degenerates at configurations where the constraints become linearly dependent, as shown in Section 2.3.6.

2.3.2 Mobility Analysis

The degree of freedom (DOF) of the system is equal to the number of independent coordinates required to control it. The DOF can be obtained by several methods for example Chebychev-Grübler-Kutzbach, or Gogu's Method [Gog07]. In order to elucidate the motion type, *screw theory* is used. In summary, each serial arm has 5-DOF however once the object is firmly grasped by the two arms a closed chain is formed and the object DOF becomes four. In the following sections a screw theory analysis, as described in Section 2.2.4, is applied to the closed-loop system. Firstly the twists associated with each arm are defined. Then the wrenches applied on the object are obtained. Finally, using this information the DOF of the object can be analyzed.

The screw theory analysis is carried out with respect to an intermediate frame $\hat{\mathcal{R}}_{obj}$ positioned on the object. More precisely, this frame is described as the frame whose origin coincides with the origin of the object frame $\mathcal{R}_{obj}$, but whose orientation is always equal to that of the world frame $\mathcal{R}_0$. An illustration is given in Fig. 2.9.

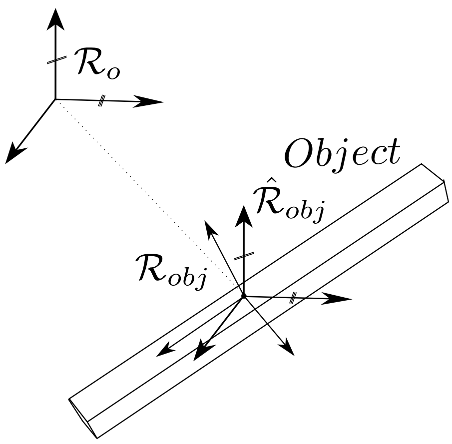

Figure 2.9: Representation of origin frame

Twist System of Nao Robot

The twist system associated with one arm of the Nao robot is spanned by five zero-pitch twists. The twist system, $\mathtt{T}^{\mathtt{r}}$, of the right arm is spanned by $\nu_{01},\ \nu_{02},\ \nu_{03},\ \nu_{04},\ \nu_{05},$

$$\nu_{01} = \begin{bmatrix} \mathbf{a_1} \times \mathbf{s}_1 \\ \mathbf{s}_1 \end{bmatrix} \quad (2.67a) \qquad \nu_{02} = \begin{bmatrix} \mathbf{a_1} \times \mathbf{s}_2 \\ \mathbf{s}_2 \end{bmatrix} \quad (2.67b)$$

with

$$\nu_{03} = \begin{bmatrix} \mathbf{b_1} \times \mathbf{s}_3 \\ \mathbf{s}_3 \end{bmatrix} \quad (2.67c) \qquad \nu_{04} = \begin{bmatrix} \mathbf{b_1} \times \mathbf{s}_4 \\ \mathbf{s}_4 \end{bmatrix} \quad (2.67d)$$

$$\nu_{05} = \begin{bmatrix} \mathbf{b_1} \times \mathbf{s}_5 \\ \mathbf{s}_5 \end{bmatrix} \quad (2.67e)$$

Likewise, the twist system of the left arm, $\mathtt{T}^1$, is spanned by $\nu_{06},\ \nu_{07},\ \nu_{08},\ \nu_{09},\ \nu_{010},$

$$\nu_{06} = \begin{bmatrix} \mathbf{a_2} \times \mathbf{s}_6 \\ \mathbf{s}_6 \end{bmatrix} \quad (2.68a) \qquad \nu_{07} = \begin{bmatrix} \mathbf{a_2} \times \mathbf{s}_7 \\ \mathbf{s}_7 \end{bmatrix} \quad (2.68b)$$

$$\nu_{08} = \begin{bmatrix} \mathbf{b_2} \times \mathbf{s}_8 \\ \mathbf{s}_8 \end{bmatrix} \quad (2.68c) \qquad \nu_{09} = \begin{bmatrix} \mathbf{b_2} \times \mathbf{s}_9 \\ \mathbf{s}_9 \end{bmatrix} \quad (2.68d)$$

$$\nu_{010} = \begin{bmatrix} \mathbf{b_2} \times \mathbf{s}_{10} \\ \mathbf{s}_{10} \end{bmatrix} \quad (2.68e)$$

$\mathbf{s}_1$, $\mathbf{s}_2$ are the unit vectors of the first and second revolute joint axes of the U-joint of the right arm, while $\mathbf{s}_6$, $\mathbf{s}_7$ are the equivalent unit vectors of the left arm. $\mathbf{s}_3$, $\mathbf{s}_4$ and $\mathbf{s}_5$ are the unit vectors of the revolute joints associated with the S-joint of the right arm, while $\mathbf{s}_8$, $\mathbf{s}_9$ and $\mathbf{s}_{10}$ are the equivalent unit vectors of the left arm. Let $\mathbf{a}_1$, $\mathbf{a}_2$, $\mathbf{b}_1$ and $\mathbf{b}_2$ represent the Cartesian vectors from A_1, A_2, B_1 and B_2 to the origin $\hat{\mathcal{F}}_{obj}$ respectively, as shown in Fig. 2.8. The twist system $\mathtt{T}$ of the Nao robot is the intersection of $\mathtt{T}^r$ and $\mathtt{T}^l$.

Constraint Wrench System of Nao Robot

From Section 2.3.2, the twist systems $\mathtt{T}^r$ and $\mathtt{T}^l$ associated with the right and left arms of Nao robot are 5-systems when the arms do not reach any singular configuration. Therefore, the constraint wrench system $\mathtt{W}_c^r$ of the right arm and the constraint wrench system $\mathtt{W}_c^l$ of the left arm are $(6-5)$-systems, i.e., 1-systems, when the arms do not reach any singular configuration. $\mathtt{W}_c^r$ and $\mathtt{W}_c^l$ are reciprocal to $\mathtt{T}^r$ and $\mathtt{T}^l$ respectively and are expressed as follows:

$$\mathtt{W}_c^r = \mathrm{span}\left(\zeta_{0r}^c\right) \quad (2.69)$$
$$\mathtt{W}_c^l = \mathrm{span}\left(\zeta_{0l}^c\right) \quad (2.70)$$

From reciprocity condition (3):

$$\zeta^c_{0r} = \begin{bmatrix} \mathbf{b_1} \times \mathbf{u}_r \\ \mathbf{u}_r \end{bmatrix} \tag{2.71}$$

$$\zeta^c_{0l} = \begin{bmatrix} \mathbf{b_2} \times \mathbf{u}_l \\ \mathbf{u}_l \end{bmatrix} \tag{2.72}$$

$\mathbf{u}_r$ is the unit vector intersecting points A_1 and B_1, while $\mathbf{u}_l$ is the unit vector intersecting points A_2 and B_2. The constraint wrench system $\mathtt{W}^c$ of the rigid object firmly grasped by the hands of the robot is the linear combination of the wrench systems $\mathtt{W}^r_c$ and $\mathtt{W}^l_c$, namely,

$$\mathtt{W}^c = \mathtt{W}^r_c + \mathtt{W}^l_c = \text{span}\,(\zeta^c_{0r},\, \zeta^c_{0l}) \tag{2.73}$$

As a result, the constraint wrench system $\mathtt{W}^c$ of Nao robot is a 2-system spanned by two pure forces (zero pitch wrenches) as long as the robot does not reach a constraint singularity. These forces intersect both the U-joint and the S-joint of the arms, i.e., the axis of the twists of each arm.

2.3.3 Actuation Schemes

In this section a selection criterion is given for suitable actuated joints using actuator wrenches. By focusing on the minimum number of actuators required to fully control the 4-DOF of the object, it is possible to find $\left(\dfrac{10!}{4!(10-4)!} = \right)$ 210 possible actuation schemes. $\mathbf{q}_a$ denotes the vector of actuation joints. $\mathbf{q}_a = [q_i \;\; q_j \;\; q_k \;\; q_l]$ means that the i^{th}, j^{th}, k^{th} and l^{th} joints of the closed loop kinematic chain are actuated where $i,\ j,\ k,\ l = 1 \ldots 10,\ i \neq j \neq k \neq l$.

Due to the symmetry of the two arms, the number of kinematically distinct schemes is significantly less than 210. For example we treat the actuation scheme no. 1 $\mathbf{q}_a = [q_1 \;\; q_2 \;\; q_3 \;\; q_4]$ and its mirror image $\mathbf{q}_a = [q_6 \;\; q_7 \;\; q_8 \;\; q_9]$ as equal. On the other hand scheme no.23 $\mathbf{q}_a = [q_1 \;\; q_2 \;\; q_6 \;\; q_7]$ has no symmetric equivalent. By excluding symmetrical actuation schemes, 110 actuation schemes remain as given in Table 2.3. Each scheme has its actuation scheme number written on the left. Schemes that have no symmetric equivalent are marked with the superscript * for example scheme no. 43.

The schemes can be subdivided into either inadmissible actuation schemes or admissible actuation schemes. The reason for the inadmissibility, a degeneracy in the global wrench system, is demonstrated in this section. The inadmissible schemes, 39 in total, are written with a strike-through notation. The closed-loop scheme contains ten revolute joints, each passive joint is denoted as 0 whereas each actuated joint is denoted as 1. For example scheme no. 4, $\mathbf{q}_a = [q_1 \;\; q_2 \;\; q_3 \;\; q_8]$ is represented as 11100—00100.

In summary, from the 210 schemes, by excluding all the schemes which are either inadmissible or have symmetric equivalent, the total number of admissible kinematically distinct actuation schemes is found to be 71. Furthermore, schemes 1, 7, 13, 34 and 77, though technically feasible, create systems where one arm contains only passive joints. Therefore if excluded the number of valid cooperative schemes would reduce to 67 (since scheme no. 77 is inadmissible, it has already been excluded).

Actuation Wrench System of Nao Robot arms

In order to analyze the actuation schemes, the wrenches exerted by each joint when chosen as an actuated joint are examined. The wrench exerted by an actuator on the object is defined as the wrench reciprocal to all the twists of the specified arm except the twist corresponding to the selected actuator itself. The actuator wrench for joint j is denoted as $\zeta^a_{\lambda j}$. The zero pitch actuation wrenches associated with the right arm are deduced from the reciprocity conditions as given in Section 2.2.4. For clarity, Fig. 2.10 shows the unit vectors associated with the wrench system of the right arm.

$$\zeta^a_{01} = \begin{bmatrix} \mathbf{b_1} \times \mathbf{u}_1 \\ \mathbf{u}_1 \end{bmatrix} \quad (2.74a) \qquad \zeta^a_{02} = \begin{bmatrix} \mathbf{b_1} \times \mathbf{u}_2 \\ \mathbf{u}_2 \end{bmatrix} \quad (2.74b)$$

$$\zeta^a_{03} = \begin{bmatrix} \mathbf{a_1} \times \mathbf{u}_3 \\ \mathbf{u}_3 \end{bmatrix} \quad (2.74c) \qquad \zeta^a_{04} = \begin{bmatrix} \mathbf{a_1} \times \mathbf{u}_4 \\ \mathbf{u}_4 \end{bmatrix} \quad (2.74d)$$

$$\zeta^a_{05} = \begin{bmatrix} \mathbf{a_1} \times \mathbf{u}_5 \\ \mathbf{u}_5 \end{bmatrix} \quad (2.74e)$$

The terms in (2.74e) are defined in the following and by referring to Fig.2.10. It should be recalled that A_1 is located at the center of the U-joint B_1 is located at the center of the S-joint of the right arm. The unit vectors are described as follows:

1. $\mathbf{u}_1$ is the unit vector passing through point B_1 (thus reciprocal to twists ν_{03}, ν_{04} and ν_{05}) and parallel to $\mathbf{s}_2$ (thus reciprocal to twist ν_{02}).
2. $\mathbf{u}_2$ is the unit vector passing through point B_1 and parallel to $\mathbf{s}_1$ (thus reciprocal to twist ν_{01}).
3. $\mathbf{u}_3$ is the unit vector of the intersection line $\mathcal{L}_3$ of planes $\mathcal{P}_{45}$ and $\mathcal{P}_{12}$.
4. $\mathbf{u}_4$ is the unit vector of the intersection line $\mathcal{L}_4$ of planes $\mathcal{P}_{35}$ and $\mathcal{P}_{12}$.
5. $\mathbf{u}_5$ is the unit vector of the intersection line $\mathcal{L}_5$ of planes $\mathcal{P}_{34}$ and $\mathcal{P}_{12}$, where $\mathcal{P}_{12}$ is the plane spanned by vectors $\mathbf{s}_1$ and $\mathbf{s}_2$ passing through point A_1.

where

1. $\mathcal{P}_{34}$ is the plane spanned by vectors $\mathbf{s}_3$ and $\mathbf{s}_4$ passing through point B_1.
2. $\mathcal{P}_{35}$ is the plane spanned by vectors $\mathbf{s}_3$ and $\mathbf{s}_5$ passing through point B_1.

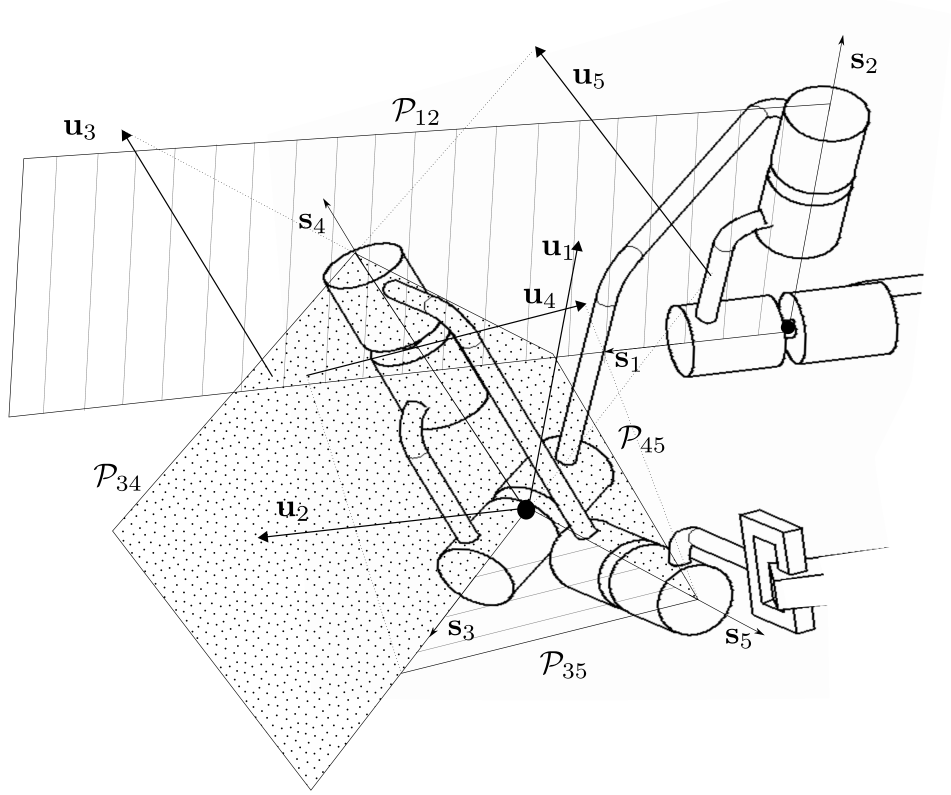

Figure 2.10: Unit vectors of wrench system of right arm

3. $\mathcal{P}_{45}$ is the plane spanned by vectors $\mathbf{s}_4$ and $\mathbf{s}_5$ passing through point B_1.

Similarly, the zero pitch actuation wrenches associated with the left arm are defined as:

$$\zeta^a_{06} = \begin{bmatrix} \mathbf{b_2} \times \mathbf{u}_6 \\ \mathbf{u}_6 \end{bmatrix} \quad (2.75a) \qquad \zeta^a_{07} = \begin{bmatrix} \mathbf{b_2} \times \mathbf{u}_7 \\ \mathbf{u}_7 \end{bmatrix} \quad (2.75b)$$

$$\zeta^a_{08} = \begin{bmatrix} \mathbf{a_2} \times \mathbf{u}_8 \\ \mathbf{u}_8 \end{bmatrix} \quad (2.75c) \qquad \zeta^a_{09} = \begin{bmatrix} \mathbf{a_2} \times \mathbf{u}_9 \\ \mathbf{u}_9 \end{bmatrix} \quad (2.75d)$$

$$\zeta^a_{010} = \begin{bmatrix} \mathbf{a_2} \times \mathbf{u}_{10} \\ \mathbf{u}_{10} \end{bmatrix} \quad (2.75e)$$

where

1. $\mathbf{u}_6$ is the unit vector passing through point B_2 (thus reciprocal to twists ν_{08}, ν_{09} and ν_{010}) and parallel to $\mathbf{s}_7$ (thus reciprocal to joints ν_{07}).
2. $\mathbf{u}_7$ is the unit vector passing through point B_2 and parallel to $\mathbf{s}_6$(thus reciprocal to twist ν_{06}).
3. $\mathbf{u}_8$ is the unit vector of the intersection line $\mathcal{L}_8$ of planes $\mathcal{P}_{910}$ and $\mathcal{P}_{67}$.
4. $\mathbf{u}_9$ is the unit vector of the intersection line $\mathcal{L}_9$ of planes $\mathcal{P}_{810}$ and $\mathcal{P}_{67}$.
5. $\mathbf{u}_{10}$ is the unit vector of the intersection line $\mathcal{L}_{10}$ of planes $\mathcal{P}_{89}$ and $\mathcal{P}_{67}$, where $\mathcal{P}_{67}$ is the plane spanned by vectors $\mathbf{s}_6$ and $\mathbf{s}_7$ passing through point A_2.

where

1. $\mathcal{P}_{89}$ is the plane spanned by vectors $\mathbf{s}_8$ and $\mathbf{s}_9$ passing through point B_2.
2. $\mathcal{P}_{810}$ is the plane spanned by vectors $\mathbf{s}_8$ and $\mathbf{s}_{10}$ passing through point B_2.
3. $\mathcal{P}_{910}$ is the plane spanned by vectors $\mathbf{s}_9$ and $\mathbf{s}_{10}$ passing through point B_2.

It should be recalled that A_2 is located at the center of the U-joint B_2 is located at the center of the S-joint of the left arm.

For any choice of actuators, the actuator wrench system is spanned by the chosen actuator wrenches. Taking for example, scheme no.1 $\mathbf{q}_a = [q_1 \ q_2 \ q_3 \ q_4]$ (11110—00000). The actuation wrench system, $\mathtt{W}^a$, is spanned by the following four zero pitch wrenches:

$$\mathtt{W}^a = \text{span}\,(\zeta^a_{01}, \ \zeta^a_{02}, \ \zeta^a_{03}, \ \zeta^a_{04}) \tag{2.76}$$

The global wrench system $\mathtt{W}$ is the wrench system spanned by the constraint wrench system $\mathtt{W}^c$ and the actuation wrench system $\mathtt{W}^a$, namely:

$$\mathtt{W} = \text{span}\,(\mathtt{W}^c, \, \mathtt{W}^a) \tag{2.77}$$

For this example scheme:

$$\mathtt{W} = \text{span}\,(\zeta^c_{0r}, \ \zeta^c_{0l}, \ \zeta^a_{01}, \ \zeta^a_{02}, \ \zeta^a_{03}, \ \zeta^a_{04}) \tag{2.78}$$

Inadmissible Actuation Scheme

An inadmissible actuation scheme signifies a choice of actuated joints that render the object uncontrollable. This occurs when for any configuration of the robot arms the global wrench system from (2.77) degenerates, i.e. $\text{rank}(\mathtt{W}) < 6, \ \text{rank}(\mathtt{W}^c) = 2, \ \text{rank}(\mathtt{W}^a) \leq 4$. In order to determine these schemes using screw theory, descriptions of the actuation and constraint wrenches that hold in any configuration are examined using (2.73), (2.74) and (2.75). For the 71 inadmissible schemes, the reason for the inadmissibility can be divided into two cases.

Table 2.3: Minimum actuation schemes of Cooperative System, * indicates that there is no symmetric equivalent

1	11110—00000	29	10110—00001	57	10011—00001	85	~~01101~~ — ~~00010~~
2	11100—00001	30	10110—00010	58	10011—00010	86	~~01101~~ — ~~00100~~
3	11100—00010	31	10110—00100	59	10011—00100	87	~~01101~~ — ~~01000~~
4	11100—00100	32	10110—01000	60	10011—01000	88	~~01010~~ — ~~00011~~
5	11100—01000	33	10110—10000	61	10011—10000	89	~~01010~~ — ~~00101~~
6	11100—10000	34	10111—00000	62	10000—00111	90	~~01010~~ — ~~00110~~
7	11101—00000	35	10100—00011	63	10000—01011	91	~~01010~~ — ~~01001~~
8	11010—00001	36	10100—00101	64	10000—01101	92	~~01010~~ — ~~01010~~*
9	11010—00010	37	10100—00110	65	10000—01110	93	~~01011~~ — ~~00001~~
10	11010—00100	38	10100—01001	66	10001—00011	94	~~01011~~ — ~~00010~~
11	11010—01000	39	10100—01010	67	10001—00101	95	~~01011~~ — ~~00100~~
12	11010—10000	40	10100—01100	68	10001—00110	96	~~01011~~ — ~~01000~~
13	11011—00000	41	10100—10001	69	10001—01001	97	~~01000~~ — ~~00111~~
14	11000—00011	42	10100—10010	70	10001—01010	98	~~01001~~ — ~~00011~~
15	11000—00101	43	10100—10100*	71	10001—01100	99	~~01001~~ — ~~00101~~
16	11000—00110	44	10101—00001	72	10001—10001*	100	~~01001~~ — ~~00110~~
17	11000—01001	45	10101—00010	73	~~01110~~ — ~~00001~~	101	~~01001~~ — ~~01001~~*
18	11000—01010	46	10101—00100	74	~~01110~~ — ~~00010~~	102	~~00110~~ — ~~00011~~
19	11000—01100	47	10101—01000	75	~~01110~~ — ~~00100~~	103	~~00110~~ — ~~00101~~
20	11000—10001	48	10101—10000	76	~~01110~~ — ~~01000~~	104	~~00110~~ — ~~00110~~*
21	11000—10010	49	10010—00011	77	~~01111~~ — ~~00000~~	105	~~00111~~ — ~~00001~~
22	11000—10100	50	10010—00101	78	~~01100~~ — ~~00011~~	106	~~00111~~ — ~~00010~~
23	~~11000~~ — ~~11000~~*	51	10010—00110	79	~~01100~~ — ~~00101~~	107	~~00111~~ — ~~00100~~
24	11001—00001	52	10010—01001	80	~~01100~~ — ~~00110~~	108	~~00101~~ — ~~00011~~
25	11001—00010	53	10010—01010	81	~~01100~~ — ~~01001~~	109	~~00101~~ — ~~00101~~*
26	11001—00100	54	10010—01100	82	~~01100~~ — ~~01010~~	110	~~00011~~ — ~~00011~~*
27	11001—01000	55	10010—10001	83	~~01100~~ — ~~01100~~*		
28	11001—10000	56	10010—10010*	84	~~01101~~ — ~~00001~~		

Case 1: The closed chain can rotate freely about axis $(B_1 B_2)$. This inadmissibility is present in 1 of the 110 schemes given in Table 2.3 (scheme number 23 denoted as 11000—11000)

Case 2: The closed chain can rotate freely about axis $(A_1 A_2)$. This inadmissibility is present in 38 of the 110 schemes given in Table 2.3

In the following these two cases are analyzed in detail:

Case 1: Scheme no. 23 11000—11000 i.e. $\mathbf{q}_a = [q_1 \;\; q_2 \;\; q_6 \;\; q_7]$.

Since it is generally preferable to actuate joints close to the base, the case where the base U-joints are actuated is examined. The global wrench system as illustrated in Fig. 2.11, is spanned by six pure forces, namely, $\mathtt{W} = span(\zeta_{0r}^c \; \zeta_{0l}^c \; \zeta_{01}^a \; \zeta_{02}^a \; \zeta_{06}^a \; \zeta_{07}^a)$. In this case the six pure forces, $\zeta_{0r}^c \; \zeta_{0l}^c \; \zeta_{01}^a \; \zeta_{02}^a \; \zeta_{06}^a \; \zeta_{07}^a$, all intersect line $(B_1 B_2)$. This implies that regardless of the configuration, there exists a zero pitch twist, whose axis is the axis $(B_1 B_2)$, that is reciprocal to all forces in the global wrench system.

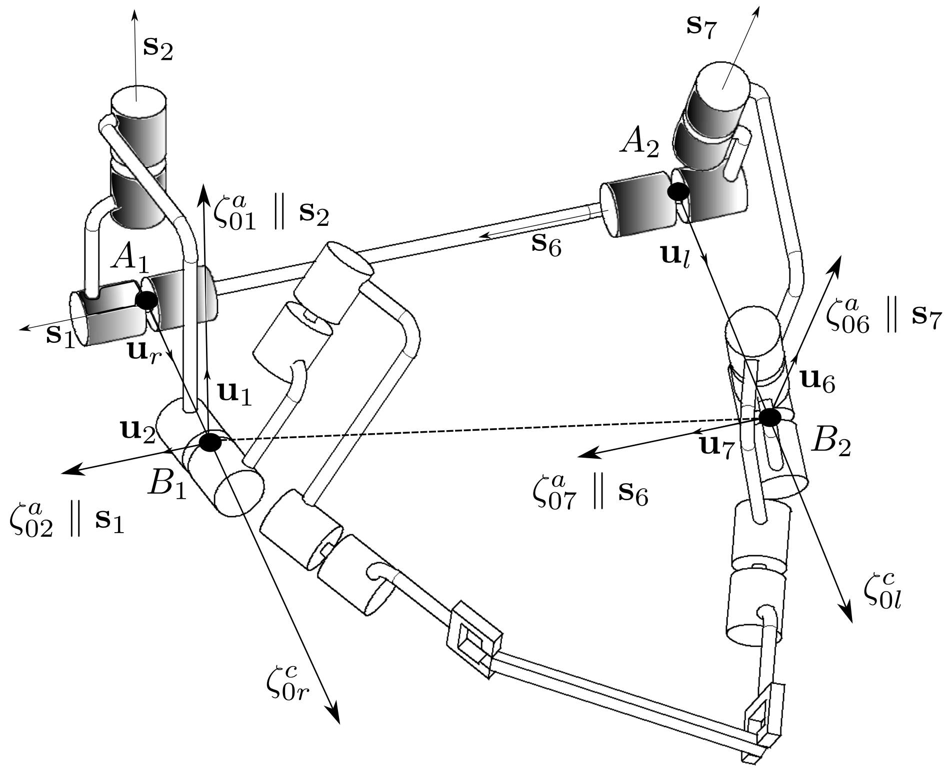

Figure 2.11: Non-admissible actuation scheme, scheme no. 23

Case 2: Neither joint 1 nor joint 6 is actuated, e.g. Scheme no. 83 01100-01100 i.e. $\mathbf{q}_a = [q_2 \;\; q_3 \;\; q_7 \;\; q_8]$.

The global wrench system as illustrated in Fig. 2.12, is given by

$$\mathtt{W} = span(\zeta_{0l}^{c}\ \zeta_{0r}^{c}\ \zeta_{02}^{a}\ \zeta_{03}^{a}\ \zeta_{07}^{a}\ \zeta_{08}^{a}) \tag{2.79}$$

In this case the six pure forces, $\zeta_{0l}^{c}\ \zeta_{0r}^{c}\ \zeta_{02}^{a}\ \zeta_{03}^{a}\ \zeta_{07}^{a}\ \zeta_{08}^{a}$, all intersect line (A_1A_2). This implies that regardless of the configuration, there exists a zero pitch twist, whose axis is the line (A_1A_2) and passing through point A_1, that is reciprocal to all forces of the global wrench system.

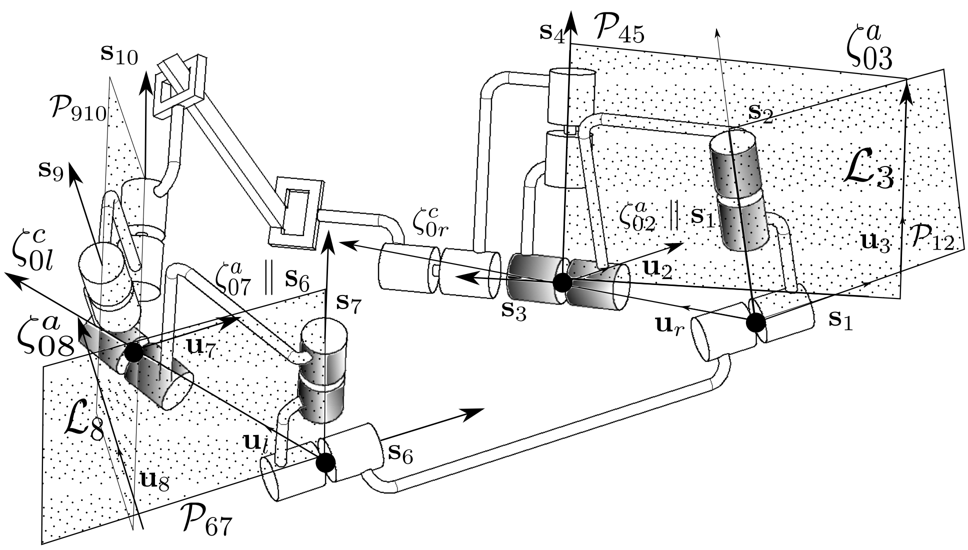

Figure 2.12: Non-admissible actuation scheme, case 2, scheme no. 83

2.3.4 Local Motion Analysis Based on Screw Theory

In this section the local motion of mechanism is obtained by examining the constraint forces in different configurations. The possible motions of the object frame are reciprocal to these constraint forces.

When the constraint forces are parallel, i.e., $\zeta_{0r}^{c} \parallel \zeta_{0l}^{c}$ [1], there are two independent ∞-pitch twists, $\epsilon_{\infty 1}$ and $\epsilon_{\infty 2}$, reciprocal (reciprocity condition 1. Section 2.2.4) to ζ_{0r}^{c} and ζ_{0l}^{c}. There are also two independent zero-pitch twists, ϵ_{01} and ϵ_{02}, reciprocal (coplanar, reciprocity condition 3. Section 2.2.4) to ζ_{0r}^{c} and ζ_{0l}^{c} as shown in Fig. 2.13. Therefore,

1. $\parallel$ implies the axes of the two screws are parallel, whereas $\nparallel$ implies they are not parallel

locally the motion can be decomposed into 2 translations normal to the constraint forces and any two linearly independent rotations in the plane formed by the constraint forces. Then the infinitesimal motion type is 2T2R [2].

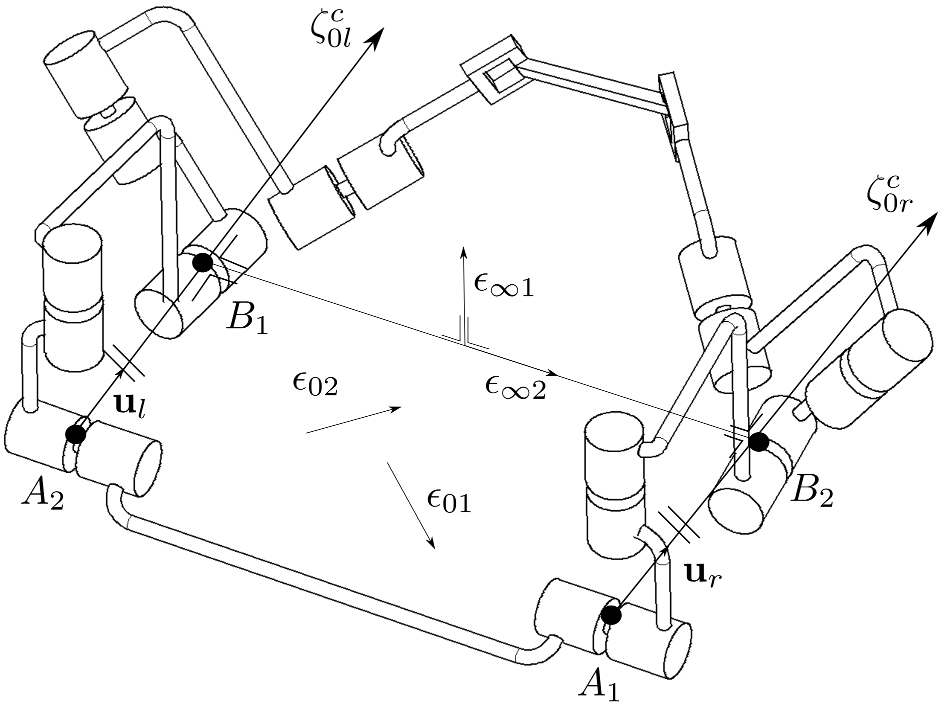

Figure 2.13: Reciprocal twists to parallel constraint forces (2T2R infinitesimal motion type)

When $\zeta^c_{0r} \nparallel \zeta^c_{0l}$ but the axes intersect, there is one ∞-pitch twist $\epsilon_{\infty 1}$ reciprocal (normal) to both ζ^c_{0r} and ζ^c_{0l} and there are three independent zero-pitch twists, ϵ_{01}, ϵ_{02} and ϵ_{03}, reciprocal (coplanar) to both ζ^c_{0r} and ζ^c_{0l} as shown in Fig. 2.14. In this case, the object can perform three infinitesimal rotations about the intersection point of the constraint forces and one infinitesimal translation along the normal to both constraint forces. Therefore, the infinitesimal motion type is 1T3R.

A final more general case is where the constraint forces are neither parallel nor intersecting as shown in Fig. 2.15. In this case the infinitesimal motion type is still 1T3R however the three rotational axes do not intersect. It is noteworthy that the set of all lines intersecting two given skew lines generates a linear line variety of dimension 4 called a hyperbolic congruence [ACWK11]. As a consequence the object can perform three infinitesimal rotations about three axes that do not intersect and one infinitesimal translation along the direction normal to the two constraint forces.

2. T and R stand for Translation and Rotation, respectively.

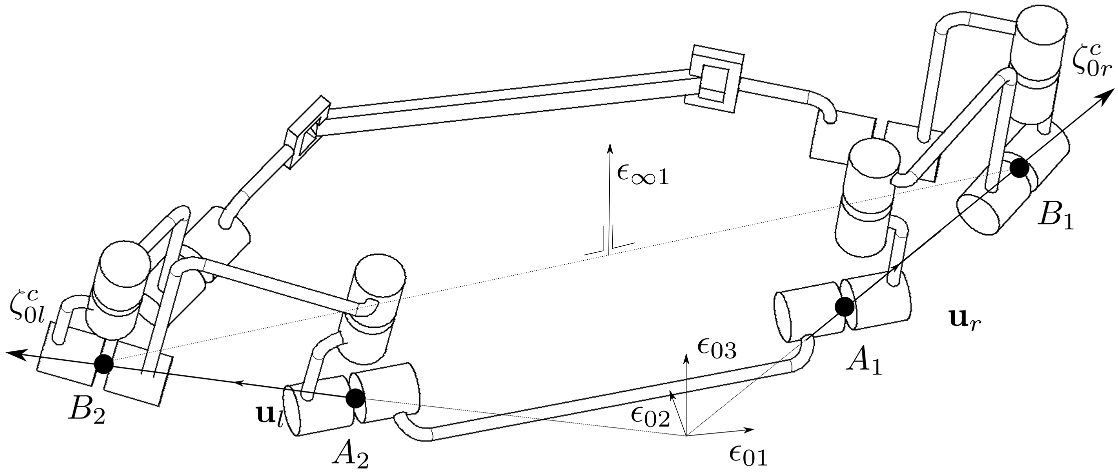

Figure 2.14: Reciprocal twists to intersecting constraint forces (1T3R infinitesimal motion mode)

Finally in the above cases, it should be noted for control purposes that the motion does not belong to a kinematic displacement subgroup as described in Section 2.2, Table 2.1. Therefore three of any combination of the six spatial motions can be simultaneously controlled. For instance, a motion type 3T can be generated for the object as long as the fourth variable is left uncontrolled.

2.3.5 Singularity Analysis

This section deals with the singularity analysis of the NAO T14 cooperating arms when it firmly grasps an object. The singularities are defined according to the convention established in [ACWK11], i.e.

Arm singularities: Arm singularities are characterized by a loss of DOF of the arm

Parallel singularities: Parallel singularities are characterized by a gain of DOF or a lack of stiffness of the manipulator. Parallel singularities are further classified as:

- *Constraint Singularities:* the lack of stiffness in the mechanism is independent of the choice of actuation scheme.
- *Actuation Singularities:* the lack of stiffness in the mechanism is dependent on the choice of actuation scheme.
- *Inner Singularities:* the lack of stiffness occurs at a link but not at the end effector of the mechanism.

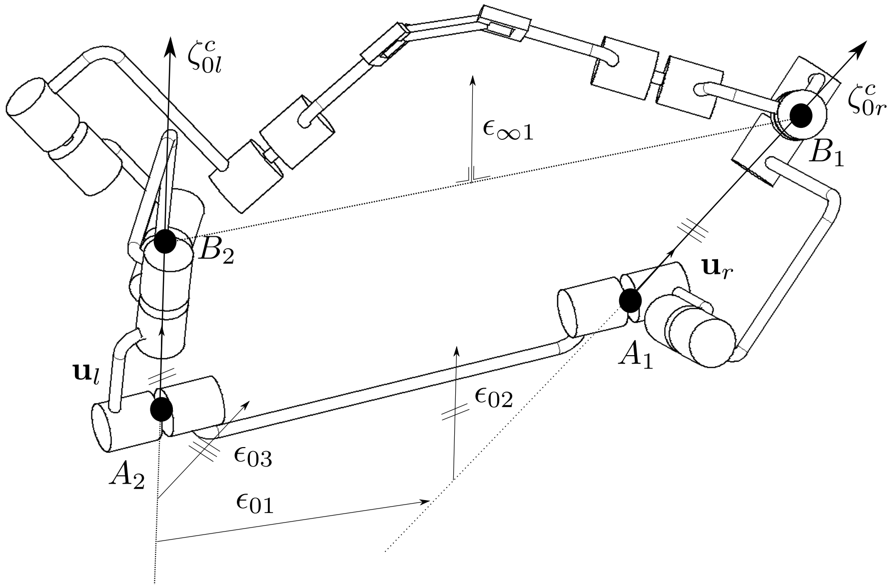

Figure 2.15: Reciprocal twists to non-intersecting and non parallel constraint forces (1T3R infinitesimal motion mode)

Arm Singularities

An arm singularity is similar to the singularity of a serial manipulator. It occurs for the dual-arm system when the arm kinematic screw system (twist system) degenerates. Consequently, the grasped object loses one or more DOF in such a configuration. From (2.67) the kinematic Jacobian matrix of the right arm can be written as:

$$\mathbf{J}_r = \begin{bmatrix} \mathbf{a}_1 \times \mathbf{s}_1 & \mathbf{a}_1 \times \mathbf{s}_2 & \mathbf{b}_1 \times \mathbf{s}_3 & \mathbf{b}_1 \times \mathbf{s}_4 & \mathbf{b}_1 \times \mathbf{s}_5 \\ \mathbf{s}_1 & \mathbf{s}_2 & \mathbf{s}_3 & \mathbf{s}_4 & \mathbf{s}_5 \end{bmatrix} \tag{2.80}$$

To simplify the analysis, the origin of frame $\hat{\mathcal{R}}_{obj}$ is transformed to the S-joint center. Equation (2.80) becomes:

$$\mathbf{J}_r = \begin{bmatrix} \mathbf{a}_1 \times \mathbf{s}_1 & \mathbf{a}_1 \times \mathbf{s}_2 & \mathbf{0}_3 & \mathbf{0}_3 & \mathbf{0}_3 \\ \mathbf{s}_1 & \mathbf{s}_2 & \mathbf{s}_3 & \mathbf{s}_4 & \mathbf{s}_5 \end{bmatrix} \tag{2.81}$$

The right arm reaches a limb singularity when $\mathbf{J}_r$ is rank deficient. There are two possible singular configurations leading to rank deficiency. Firstly when $\mathbf{s}_3$ is parallel to $\mathbf{s}_5$ obtained at $\theta_4 = 0 \pm \pi$. Secondly when the S-joint center lies on the line passing through point A_1 and parallel to $\mathbf{s}_1$, meaning $\mathbf{s}_1 \times \mathbf{a}_1 = \mathbf{0}_3$. This configuration occurs at $\theta_2 = \text{atan}\left(\dfrac{-r_3}{d_3}\right)$.

The effect of arm singularities in closed-loop The serial singularity in closed-loop means that the object loses 1-DOF regardless of the actuation scheme. This is due to the fact that there are now three independent constraint wrenches applied on the object ζ^c_{0r}, ζ^c_{0l} and the wrench due to the singularity. However the actuation scheme affects whether or not there is internal motion in the mechanism, i.e., if a link can move locally without affecting the pose of the object. This type of local motion is known as an inner singularity.

In summary, for the serial singularity condition $\theta_4 = 0$, when neither joint 3 nor joint 5 are actuated, the global wrench system degenerates and $Link\ 4$ can move freely. Equally in the second serial singularity case $\theta_2 = \text{atan}\left(\frac{-r_3}{d_3}\right)$, the linearly dependent joints are joints 1, 3, 4 and 5. Again, if none of these joints are actuated during the singular configuration, an inner singularity will occur.

Example: Scheme no. 12 11010—10000 i.e. $\mathbf{q}_a = [q_1\ \ q_2\ \ q_4\ \ q_6]$. When the arm is in a serial configuration $\theta_4 = 0$, we propose to analyze the mechanism by *breaking* the chain at $Link\ 4$, separating the two linearly dependent joints as shown in Fig. 2.16. By breaking the chain in this way, $Link\ 4$ becomes analogous to the object and the procedure outlined in Section 2.3.3 can be reapplied. If the loop is broken so the two linearly independent joints remain on the same serial chain, this procedure breaks down. Hence, in this case the twist system associated with the right and the left chains are:

$$\mathtt{T}^{\mathtt{r}} = \text{span}\left(\nu_{01},\ \ \nu_{02},\ \ \nu_{03},\ \ \nu_{04}\right) \tag{2.82}$$

$$\mathtt{T}^{\mathtt{l}} = \text{span}\left(\nu_{06},\ \ \nu_{07},\ \ \nu_{08},\ \ \nu_{09},\ \ \nu_{010},\ \ \nu_{05}\right) \tag{2.83}$$

The left arm is now composed of six joints while the right arm is composed of four joints. Thus there are no constraint forces associated with the $\mathtt{T}^{\mathtt{l}}$, meanwhile there are now two constraint wrenches associated with the right arm: ζ^c_{0r1} the constraint wrench expressed in (2.71), and ζ^c_{0r2} equivalent to ζ^a_{05} from (2.74e) since it should be reciprocal to $\nu_{01},\ \ \nu_{02},\ \ \nu_{03},\ \ \nu_{04}$, hence

$$\zeta^c_{0r1} = \begin{bmatrix} \mathbf{b_1} \times \mathbf{u}_r \\ \mathbf{u}_r \end{bmatrix} \tag{2.84a}$$

$$\zeta^c_{0r2} = \begin{bmatrix} \mathbf{a_1} \times \mathbf{u}_5 \\ \mathbf{u}_5 \end{bmatrix} \tag{2.84b}$$

The first, second and fourth joints are actuated, therefore the actuation wrench system is spanned by the following pure forces:

$$\zeta^a_{01} = \begin{bmatrix} \mathbf{b_1} \times \mathbf{u}_1 \\ \mathbf{u}_1 \end{bmatrix} \tag{2.85a}$$

$$\zeta^a_{04} = \begin{bmatrix} \mathbf{b_1} \times \mathbf{u}_{4s} \\ \mathbf{u}_{4s} \end{bmatrix} \tag{2.85b}$$

$$\zeta^a_{02} = \begin{bmatrix} \mathbf{b_2} \times \mathbf{u}_2 \\ \mathbf{u}_2 \end{bmatrix} \tag{2.85c}$$

$\mathbf{u}_{4s}$ is the unit vector of the line passing through point A_1 and parallel to $\mathbf{s}_3$. Joint 6, of the left arm is actuated, therefore the actuation wrench system is spanned by the following finite pitch wrench, namely:

$$\zeta_{06}^{a} = \begin{bmatrix} \mathbf{r}_{6s} \times \mathbf{u}_{6s} + \lambda_{6s}\mathbf{u}_{6s} \\ \mathbf{u}_{6s} \end{bmatrix} \tag{2.86}$$

$\mathbf{u}_{6s}$ is the unit vector of the screw reciprocal to $\nu_{07},\ \nu_{08},\ \nu_{09},\ \nu_{010},\ \nu_{05}$, while $\mathbf{r}_{6s}$ is a vector pointing from any point on this axis to the origin.

As a result, the global wrench system applied on *Link 4* is spanned by ζ_{0r1}^{c}, ζ_{0r2}^{c}, ζ_{01}^{a}, ζ_{02}^{a}, ζ_{04}^{a}, and $\zeta_{\lambda 6}^{a}$, namely:

$$\mathtt{W} = \text{span}\left(\zeta_{0r1}^{c},\ \zeta_{0r2}^{c},\ \zeta_{01}^{a},\ \zeta_{02}^{a},\ \zeta_{04}^{a},\ \zeta_{\lambda 6}^{a}\right) \tag{2.87}$$

The singularity condition means that ν_{03} and ν_{05} are linearly dependent, therefore logically any screw that is reciprocal ν_{03} is reciprocal to ν_{05} and vice versa. It follows that with this choice of actuators, ν_{03} and ν_{05} are reciprocal to all the wrenches of the global wrench system.

Using (2.50), (2.84), (2.85), and (2.86):

$$\begin{bmatrix} \mathbf{u}_r & \mathbf{b_1} \times \mathbf{u}_r \\ \mathbf{u}_5 & \mathbf{a_1} \times \mathbf{u}_5 \\ \mathbf{u}_1 & \mathbf{b_1} \times \mathbf{u}_1 \\ \mathbf{u}_2 & \mathbf{b_2} \times \mathbf{u}_2 \\ \mathbf{u}_{4s} & \mathbf{b_1} \times \mathbf{u}_{4s} \\ \mathbf{u}_{7s} & \mathbf{r_7} \times \mathbf{u}_{7s} + \lambda \mathbf{u}_{7s} \end{bmatrix} \begin{bmatrix} \mathbf{b_1} \times \mathbf{s}_3 \\ \mathbf{s}_3 \end{bmatrix} = 0 \tag{2.88}$$

Hence the null space is spanned by ν_{03} and ν_{05}. Since the null space exists, the global wrench system from (2.87) must be rank deficient, and unable to fully constrain $Link\ 4$.

Parallel Singularities

Constraint Singularities A constraint singularity occurs when the constraint wrench system (2.73) degenerates, i.e., when ζ_{0r}^{c} and ζ_{0l}^{c} are linearly dependent[Gog08, ZBG02]. This configuration is illustrated in Fig.2.17. The closed-loop system reaches such a configuration when the two S-joint centers lie on $\mathbf{s}_1$ and $\mathbf{s}_6$, namely:

$$\theta_2 = \text{atan}\left(\frac{-r_3}{d_3}\right) \quad \text{and} \quad \theta_7 = \text{atan}\left(\frac{r_3}{d_3}\right) \tag{2.89}$$

It is noted that when the closed-loop system reaches a constraint singularity, both arms are in arm singularity configuration as described in Section 2.3.5. From Section 2.3.5,

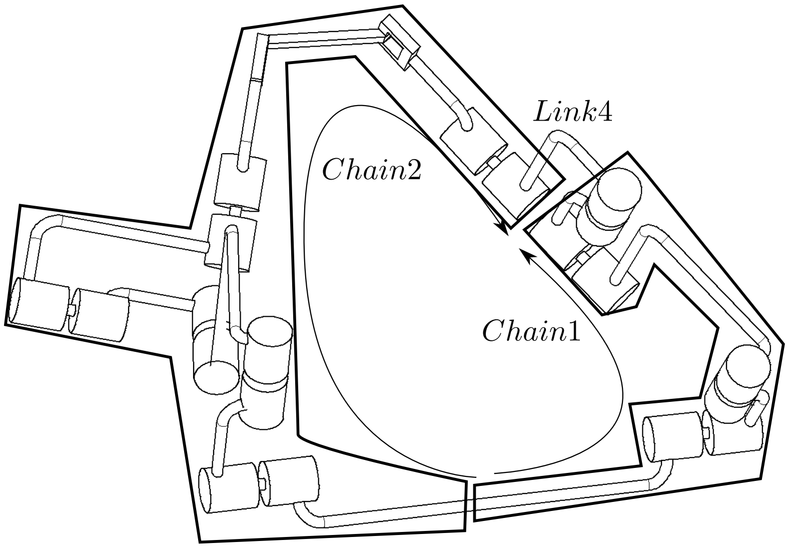

Figure 2.16: Serial chains for inner singularity analysis

it can be seen that an arm singularity in a serial mechanism increases the degree of the constraint wrench system. Let the constraint wrench due to the serial singularities of the right and left arm be denoted as ζ_{0s1} and ζ_{0s2} respectively, which are obtained as:

$$\zeta_{0sr}^{c} = \begin{bmatrix} \mathbf{b_1} \times \mathbf{u_1} \\ \mathbf{u_1} \end{bmatrix} \tag{2.90}$$

$$\zeta_{0sl}^{c} = \begin{bmatrix} \mathbf{b_2} \times \mathbf{u_6} \\ \mathbf{u}_6 \end{bmatrix} \tag{2.91}$$

Hence at the studied configuration, four wrenches forming a 3-system as described in Fig. 2.17 are applied on the object: the constraint wrenches ζ_{0r}^{c} and ζ_{0l}^{c} and the wrenches due to the serial singularity of each arm ζ_{0sr}^{c} and ζ_{0sl}^{c}. Consequently, the object has 3-DOF in this configuration.

The infinitesimal motion type varies depending on the relationship between ζ_{0sr}^{c} and ζ_{0sl}^{c}. If they are parallel, there is one ∞-pitch twists, ϵ_∞ reciprocal (normal) to ζ_{0r}^{c}, ζ_{0l}^{c}, ζ_{0sr}^{c} and ζ_{0sl}^{c} while there are two independent zero-pitch twists, ϵ_{01} and ϵ_{02}, reciprocal (coplanar) to ζ_{0r}^{c}, ζ_{0l}^{c}, ζ_{0sr}^{c} and ζ_{0sl}^{c}. Therefore, locally the motion can be decomposed into 1 translation normal to the constraint forces and any two linearly independent rotations in the plane formed by the constraint forces. Then the infinitesimal motion type is 1T2R.

Contrarily if ζ^c_{0sr} and ζ^c_{0sl} are not parallel, there are three independent zero-pitch twists, ϵ_{01}, ϵ_{02} and ϵ_{03} reciprocal (coplanar) to ζ^c_{0r}, ζ^c_{0l}, ζ^c_{0sr} and ζ^c_{0sl}. In this configuration the infinitesimal motion type is 0T3R.

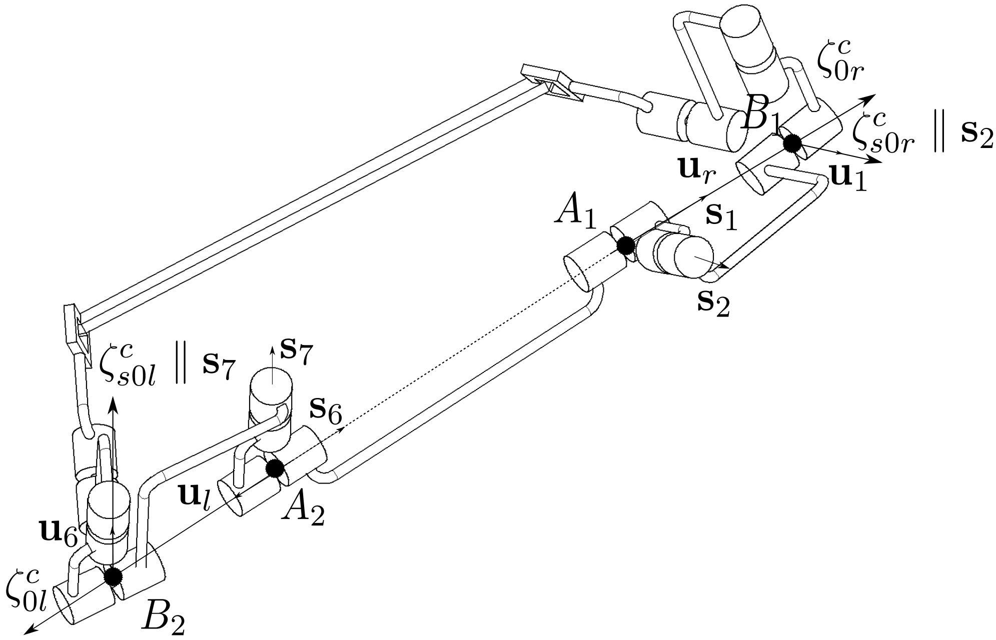

Figure 2.17: Constraint singularity of the dual-arm system

Actuation Singularities: The choice of actuators means that the object can be controlled but may encounter actuation singularities at certain configurations. An actuation singularity occurs when the global wrench system (2.77) degenerates while the constraint wrench system (2.73) does not. Due to the large number of viable actuation schemes, each presumably containing several actuation singularities, a more general illustration is presented below.

Example Actuation Singularity 1: Considering the configuration $\theta_1 = \theta_6$ for any actuation scheme where $[q_2\ q_7] \in \mathbf{q}_a$

The two actuation wrenches associated with joints 2 and 7, ζ^a_{02} and ζ^a_{07}, and the two constraint wrenches associated with the systems geometry ζ^c_{0l} and ζ^c_{0r}, normally constitute a 4-system. When $\theta_1 = \theta_6$, the unit vectors $\mathbf{u_r},\ \mathbf{u_l},\ \mathbf{u_2},\ \mathbf{u_7}$ all lie in $\mathcal{P}$, where $\mathcal{P}$ is the plane containing points $A_1,\ B_1,\ A_2,\ B_2$. The global wrench system degenerates due to the linear dependence of the wrenches ζ^a_{02}, ζ^a_{07}, ζ^c_{0l} and ζ^c_{0r}. In this case there are three twists reciprocal to all the wrenches. For any choice of the remaining two actuators, the global wrench system will degenerate. One such example of this degeneracy is found in scheme 5, $\mathbf{q}_a = [q_1,\ q_2,\ q_3,\ q_7]$ 11100—01000. The actuation singularity is shown in Fig. 2.18.

Example Actuation Singularity 2 Considering the configuration $\theta_1 = \theta_6$ and $\theta_3 = \pi/2$ for any actuation scheme where either joint 2 or 7, and joint 5 are actuated.

In this configuration the actuation wrench associated with joint 5 lies in the plane $\mathcal{P}$, where $\mathcal{P}$ is the plane containing points $A_1,\ B_1,\ A_2,\ B_2$. The global wrench system degenerates due to the linear dependence of the wrenches ζ^a_{02} (or ζ^a_{07}) , ζ^a_{05}, ζ^c_{0l} and ζ^c_{0r}. One such example of this degeneracy is found in scheme 7, $\mathbf{q}_a = [q_1,\ q_2,\ q_3,\ q_5]$ 11101—00000. The actuation singularity is shown in Fig. 2.19.

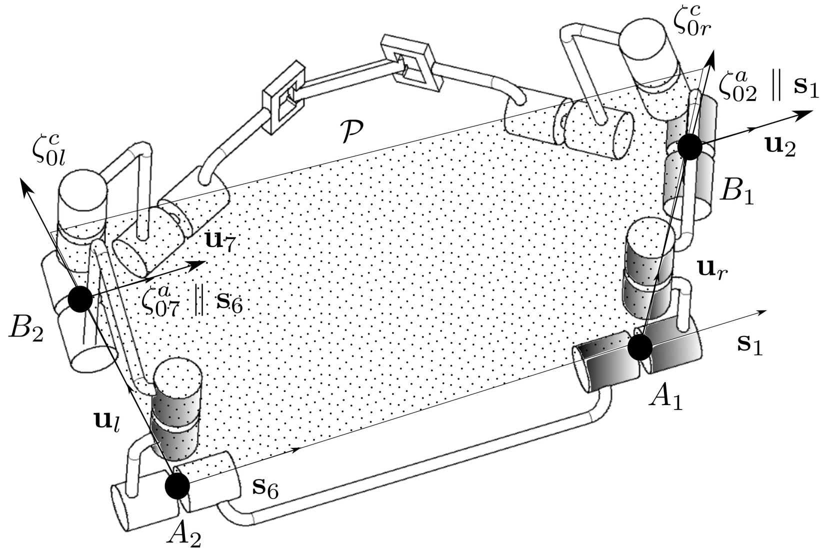

Figure 2.18: Actuation singularity for scheme no. 5

2.3.6 Comparison of Screw theory and Jacobian Methods

In this section a comparison is made between existing numerical methods outlined in Section 2.2.4, classically used to kinematically analyze the system, and the screw theory methods demonstrated in the previous sections. In order to carry out this comparison the relations derived in Section 2.3.1 are used.

Mobility Analysis

The mobility of a parallel mechanism can be calculated exactly by using closed loop kinematic constraint equations. It is defined as the number of independent joints

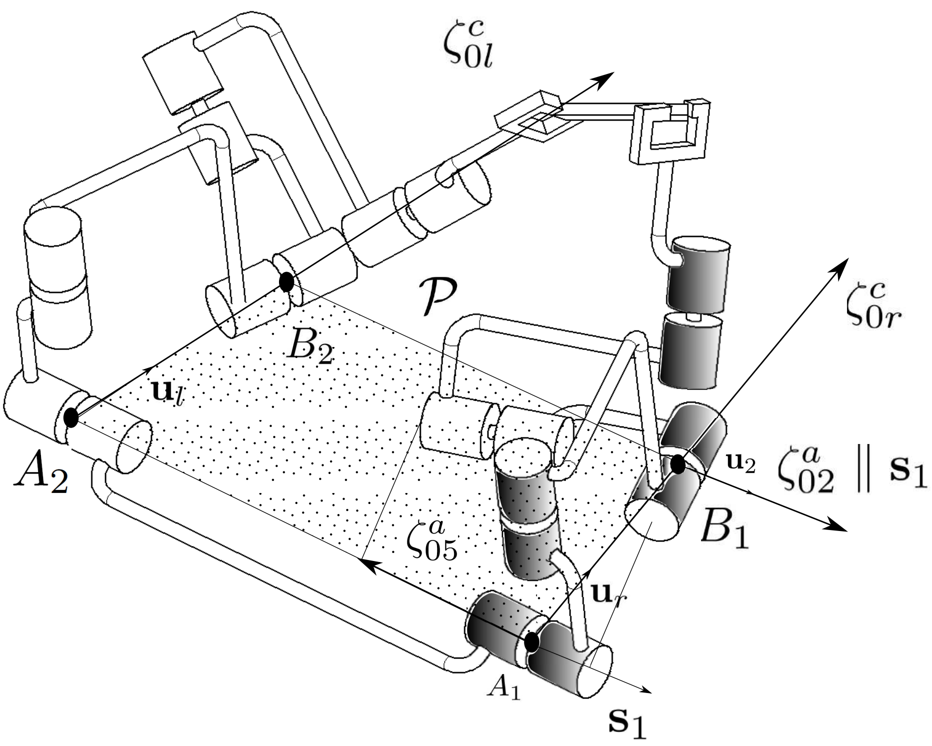

Figure 2.19: Actuation singularity for scheme no. 7

before loop closure minus those that lose their independence after the loop is closed. Using (2.34):

$$N = l - c = l - rank(\mathbf{J}_s) \tag{2.92}$$

$$rank(\mathbf{J}_s) = 6 \tag{2.93}$$

$$N = 10 - 6 = 4 \tag{2.94}$$

l is the total DOF of the mechanism's joints while c is the number of independent constraint equations. For the cooperative system $\mathbf{J}_s$ is defined in (2.58), the closed chain kinematic constraints. The primary drawback is that rank$(\mathbf{J}_s)$ is difficult to obtain symbolically, therefore it is obtained numerically for random configurations. The numerical calculation gives rank$(\mathbf{J}_s) = 6$.

Singularity analysis

Table 2.4 analyzes the singular configurations obtained in Section 2.3.5, with respect to the constraint equations. In [OW99] singular conditions are defined with respect to the Jacobian matrices defined in (2.37), (2.38) and (2.40). Recall, that a degeneration of $\mathbf{J}_{act}$ from (2.66) is an unmanipulable singularity. By obtaining the null space of $\mathbf{G}_p$, denoted as $\mathcal{N}(\mathbf{G}_p)$, parallel singularities can be investigated. $\mathbf{J}_p \cdot \mathcal{N}(\mathbf{G}_p) \neq 0$ corresponds to an unstable singularity (a loss of stiffness in the platform). $\mathbf{J}_p \cdot \mathcal{N}(\mathbf{G}_p) = 0$ corresponds to a self motion or inner singularity. To find the inadmissible actuation schemes the matrix $\mathbf{G}_p$ is examined. If this matrix is rank deficient for every value of $\mathbf{q}_a$, $\mathbf{q}_p$, and $\mathbf{q}_c$ the actuation scheme is inadmissible.

Table 2.4 validates the screw theory analysis on the system notably with respect to serial singularities. For example it shows that by changing the actuation scheme, the lack of stiffness in the system when $\mathbf{J}_p \cdot \mathcal{N}(\mathbf{G}_p) = 0$ can be avoided.

Table 2.4: Numerical Status at Singularity

Name	Conditions	No.	Actuation Scheme	rank($\mathbf{J}_{act}$)	rank($\mathbf{G}_p$)	$\mathbf{J}_p \cdot \mathcal{N}(\mathbf{G}_p)$
Benchmark		4	11100—00100	4	5	$\{\}$
Inadmissible Scheme 1		23	11000—11000	4	4	$\neq 0$
Inadmissible Scheme 2		83	01100—01100	4	4	$\neq 0$
Serial Singularity 1	$\theta_4 = 0$	12	11010—10000	4	4	$= 0$
Serial Singularity 1	$\theta_4 = 0$	13	11011—00000	3	5	$\{\}$
Serial Singularity 2	$\theta_7 = \text{atan}(r_3/d_3)$	11	11010—01000	4	4	$= 0$
Serial Singularity 2	$\theta_7 = \text{atan}(r_3/d_3)$	12	11010—10000	3	5	$\{\}$
Actuation Singularity 1	$\theta_1 = \theta_6$	5	11100—01000	4	4	$\neq 0$
Actuation Singularity 2	$\theta_1 = \theta_6$, $\theta_3 = \pm\frac{\pi}{2}$	7	11101—00000	4	5	$\neq 0$
Constraint Singularity	$\theta_2 = \text{atan}(r_3/d_3)$ $\theta_7 = \text{atan}(r_3/d_3)$	5	11100—01000	3	4	$= 0$
Constraint Singularity	$\theta_2 = \text{atan}(-r_3/d_3)$ $\theta_7 = \text{atan}(r_3/d_3)$	4	11100—00100	3	4	$\neq 0$

2.3.7 Dynamic Performance of Actuation Schemes

In this section, the dynamic performance of the actuation schemes is analyzed with respect to a large number of trajectories. Firstly the dynamic model of closed loop robots is recalled. Then the dynamic performance of admissible actuation schemes is compared when the closed chain system is transporting an object in space.

Calculation of the Inverse Dynamic Model

The inverse dynamic model (IDM) calculates the active motor torques $\boldsymbol{\tau}$ in terms of $\mathbf{q}$, $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ of the joints. The modeling procedure mentioned in Section 2.2.2 is expanded upon in the following.

Let $\mathbf{q}_{tr} = \left[\mathbf{q}_a^T \;\; \mathbf{q}_p^T\right]^T$ denote the joint variables of the tree structure. In order to find the Closed Loop Inverse Dynamic Model (CLIDM), first the IDM for the tree structure is found and then converted to CLIDM by using the following relation [KD04]:

$$\boldsymbol{\tau} = \left(\frac{\partial \mathbf{q}_{tr}}{\partial \mathbf{q}_a}\right)^T \boldsymbol{\Gamma}_{tr} = \boldsymbol{\Gamma}_a + \left(\frac{\partial \mathbf{q}_p}{\partial \mathbf{q}_a}\right)^T \boldsymbol{\Gamma}_p \tag{2.95}$$

where $\boldsymbol{\Gamma}_{tr}$ denotes the joint torques of the tree structure. It can be expressed as:

$$\boldsymbol{\Gamma}_{tr} = \begin{bmatrix} \boldsymbol{\Gamma}_a \\ \boldsymbol{\Gamma}_p \end{bmatrix} = \mathbf{A}_{tr}\left(\mathbf{q}_{tr}\right) \begin{bmatrix} \ddot{\mathbf{q}}_a \\ \ddot{\mathbf{q}}_p \end{bmatrix} + \mathbf{c}_{tr}\left(\mathbf{q}_{tr}, \dot{\mathbf{q}}_{tr}\right) \tag{2.96}$$

$\boldsymbol{\Gamma}_a$ and $\boldsymbol{\Gamma}_p$ represent the torque on the actuated and passive joints of the tree structure, respectively. $\mathbf{A}_{tr}$ and $\mathbf{H}_{tr}$ are tree structure inertia matrix and the tree structure matrix of Coriolis, Centrifugal and Gravity forces respectively. The inertial parameters of the closed chain system are given in the appendix. Using (2.61) and (2.95), we obtain:

$$\boldsymbol{\tau} = \begin{bmatrix} \mathbf{I}_N & (-\mathbf{G}_p^{-1}\,\mathbf{G}_a)^T \end{bmatrix} \begin{bmatrix} \boldsymbol{\Gamma}_a \\ \boldsymbol{\Gamma}_p \end{bmatrix} \tag{2.97}$$

$\mathbf{I}_N$ is the identity matrix of dimension N, where N is equal to the DOF of the system. Substituting the general expression for the tree dynamic model given by (2.96) into (2.95), the closed loop dynamic model is obtained as:

$$\boldsymbol{\tau} = \begin{bmatrix} \mathbf{I}_N & (-\mathbf{G}_p^{-1}\,\mathbf{G}_a)^T \end{bmatrix} \mathbf{A}_{tr} \begin{bmatrix} \ddot{\mathbf{q}}_a \\ \ddot{\mathbf{q}}_p \end{bmatrix} + \begin{bmatrix} \mathbf{I}_N & (-\mathbf{G}_p^{-1}\,\mathbf{G}_a)^T \end{bmatrix} \mathbf{c}_{tr} \tag{2.98}$$

The direct dynamic model (DDM) calculates the independent joint accelerations $\ddot{\mathbf{q}}_a$ from the motor torques $\boldsymbol{\tau}$. It can be obtained after substituting $\ddot{\mathbf{q}}_p$ in terms of $\ddot{\mathbf{q}}_a$ using (2.60) in (2.98) and solving the obtained expression to obtain $\ddot{\mathbf{q}}_\mathbf{a}$.

Dynamic Parameters of closed loop robot

The dynamic parameters of the Nao robot are taken directly from the documentation. The inertia tensor of link j is given with respect to frame j as follows:

$${}^j\mathbf{I}_j = \begin{bmatrix} XX_j & XY_j & XZ_j \\ XY_j & YY_j & YZ_j \\ XZ_j & YZ_j & ZZ_j \end{bmatrix}$$

The first moments of link j are calculated using the mass, denoted as M_j and the vector of center-of-mass coordinates denoted as $\mathbf{S}_j$, as follows:

$$M\mathbf{S}_j = \begin{bmatrix} MX_j & MY_j & MZ_j \end{bmatrix}$$

The numerical values parameters of the robot are given in Table 2.5. In the case of the closed chain formulation, the inertial parameters of the extended link 5, is composed of link 5, the object and link 10.

Table 2.5: Inertial parameters of the closed chain system

	Link 1(6)	Link 2(7)	Link 3(8)	Link 4(9)	Link 5
XX (g mm^2)	71025.99	82285.13	5503.19	25194.83	122113.5
XY (g mm ^{2})	-2024.58	-39780.57	-22.43	-2162.93	-10072.8
XZ (g mm^2)	-17.22	7526.01	-15.34	718.46	77907.38
YY (g mm^2)	14057.99	290014.459	62254.05	88903.15	558947.02
YZ (g mm^2)	8.41	-1529.45	5.59	-108.09	-5861.52
ZZ (g mm^2)	73166	268423.31	63251.24	86868.67	559976.06
MX (g mm)	-1.78	18.85	-25.6	25.56	63.6
MY (g mm)	24.96	-5.77	0.01	-2.73	-1.66
MZ (g mm)	0.18	0.65	-0.19	0.96	9.54
M (g)	69.96	123.09	59.71	77.24	333.06

In order to compare the dynamic performance, the motor capacities must be taken into account. For the Nao T14 robot, the motor specifications are given in Table 2.6.

Table 2.6: Motors specifications

Joint Motor	No load speed	Stall Torque	Nominal torque	Reduction ratio
1	11 900 rpm	15.1 mNm	3.4 mNm	150.27
2	11 900 rpm	15.1 mNm	3.4 mNm	173.21
3	11 900 rpm	15.1 mNm	3.4 mNm	150.27
4	11 900 rpm	15.1 mNm	3.4 mNm	173.21
5	11 900 rpm	15.1 mNm	3.4 mNm	50.64

Trajectory Definition

The trajectory is defined in the configuration space, such that from an initial position $\mathbf{q}_a^i$, a final position $\mathbf{q}_a^f$ is reached in time t_{final}. The position, velocity and acceleration are calculated as:

$$\begin{aligned}\mathbf{q}_a(t) &= \mathbf{q}_a^i + r(t)\left(\mathbf{q}_a^f - \mathbf{q}_a^i\right)\\ \dot{\mathbf{q}}_a(t) &= \dot{r}(t)\left(\mathbf{q}_a^f - \mathbf{q}_a^i\right)\\ \ddot{\mathbf{q}}_a(t) &= \ddot{r}(t)\left(\mathbf{q}_a^f - \mathbf{q}_a^i\right)\end{aligned}$$

The trajectory is calculated from the initial and final conditions:

$$\begin{aligned}&\mathbf{q}_a(t=0) = \mathbf{q}_a^i, \quad \mathbf{q}_a(t_{final}) = \mathbf{q}_a^f, \quad \dot{\mathbf{q}}_a(t=0) = \dot{\mathbf{q}}_a(t_{final}) = 0\\ &\ddot{\mathbf{q}}_a(t=0) = \ddot{\mathbf{q}}_a(t_{final}) = 0\end{aligned} \tag{2.99}$$

$r(t)$ is the 5^{th} degree polynomial interpolation function calculated as:

$$r(t) = 10\left(\frac{t}{t_{final}}\right)^3 - 15\left(\frac{t}{t_{final}}\right)^4 + 6\left(\frac{t}{t_{final}}\right)^5 \tag{2.100}$$

For each simulation an arbitrarily shaped rigid object is grasped by the two arms; the object is transported along a spatial trajectory by the cooperative system in a fixed time t_{final}. The trajectory is defined between two points in the joint space using a fifth degree polynomial given in (2.100). It is continuous in both velocity and acceleration. The definition of the scheme is given in Fig.2.20.

It should be noted that the choice of actuation joints for the trajectory generation does not have to correspond to the actual actuation scheme, rather its purpose is to generate a feasible closed loop trajectory. In order to avoid biasing the results, the actuation scheme for the trajectory generation is not included in the results, however the performance can be judged from its symmetric equivalent which is scheme no.1 11110-—00000.

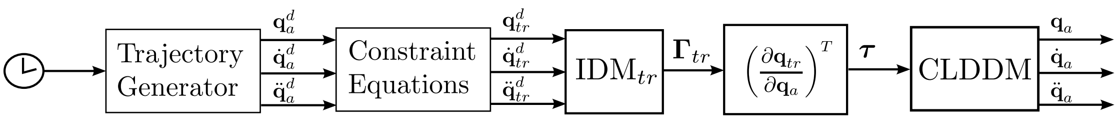

Figure 2.20: Scheme for dynamic Comparison

Dynamic Performance Results

The comparison criterion is taken as the integral of the sum of squares of the motorized torques, which indicates the motor's power loss [CA01]. The power loss of a motor is proportional to the square of the current flowing through it. Since the torque of the motor is directly proportional to the current, a power loss criterion given in (2.101), is defined from time= 0 to time= t, where η has units N^2m^2s. The motor torques can be calculated using (2.98).

$$\eta = \int_0^t \boldsymbol{\tau}^T \boldsymbol{\tau} \, dt \tag{2.101}$$

To facilitate the presentation, it is supposed that joint 10 is always modeled as the cut joint, therefore actuation schemes containing joint 10 are excluded in this configuration. [3] However in order to maximize the number of schemes that are tested, the symmetric equivalent of the these schemes are included. To illustrate this point, take scheme no 2, 11100—00001 . By cutting the chain at joint 10 this actuation scheme cannot be realized, instead the scheme 00001—11100 is used. The two schemes will have the same performance when tested over many random trajectories. In total 64 actuation schemes are tested, (there are 7 kinematically admissible schemes that remain untested).

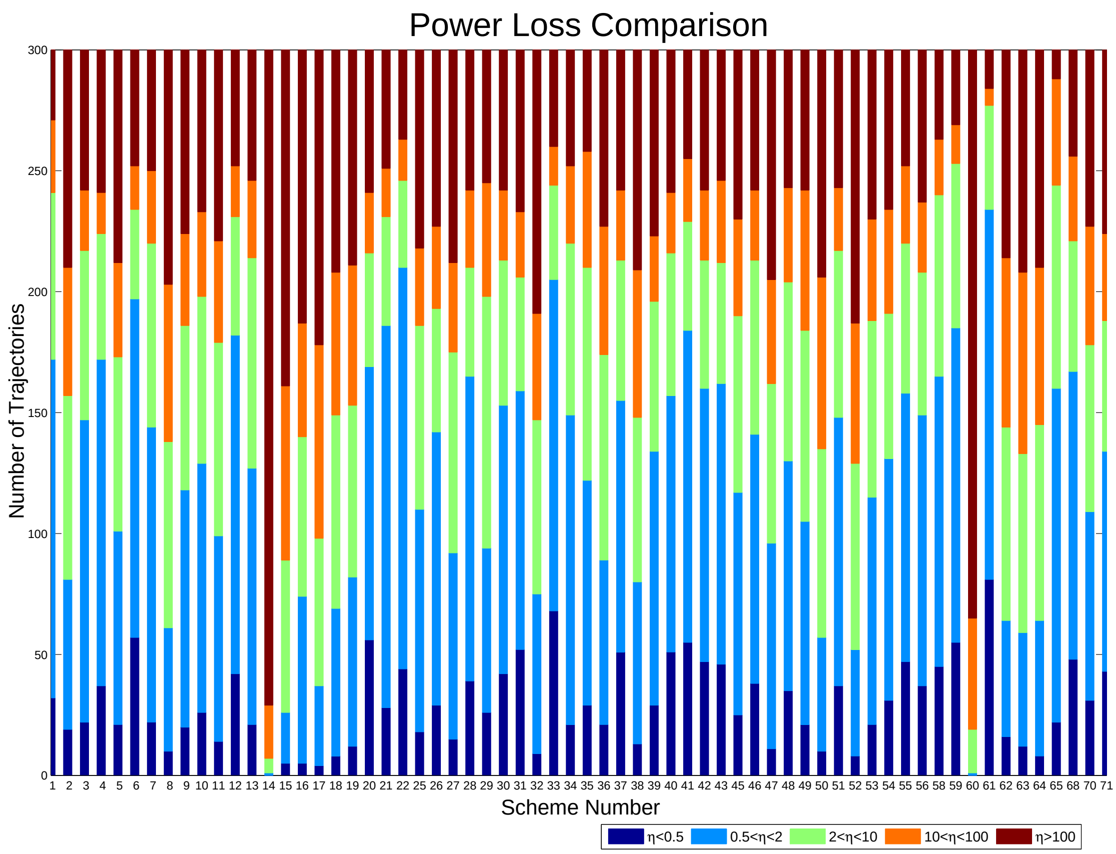

Figure 2.21: Power Loss for Random trajectories

3. To actuate q_{10} a new tree structure robot must be defined.

Figure 2.21 shows the results for three hundred trajectories. The x-axis gives the scheme number and the y-axis measures the number of trajectories. The simulation is repeated using the same object and the same trajectory, for each actuation scheme (64 times). Three hundred simulations are executed. Each simulation uses a new object and a new spatial path.

The chart shows how often η lies between the defined bands. For example, taking actuation scheme 1 11110—00000, from the three hundred trajectories, $\eta < 0.5$ in 32 cases, $0.5 < \eta < 2$ in 140 cases, $2 < \eta < 10$ in 69 cases, $10 < \eta < 100$ in 30 cases and $\eta > 100$ in 29 cases. Obviously the most attractive cases are those where $\eta < 0.5$.

The chart clearly shows a disparity of η between the schemes, notably no 60, 10011-—01000 and no 61, 10011—10000. In this case, the change of one actuator has a large effect on the resulting dynamic performance, increasing the number of trajectories where the $\eta > 100$ from 16 to 235. Furthermore a large decrease is seen in the trajectories that have a very good performance from 81 to 0. A similar phenomenon can be seen for schemes no 11, 11010—01000 and no 12, 11010—10000. Generally it can be seen that best performing schemes contain actuated joints 1 and 6, the base joints.

In Fig.2.22, the number of trajectories where the required actuated joint torque is greater than the motor's peak torque (as given in the appendix) are shown. For example, taking actuation scheme 1 11110—00000, from the three hundred trajectories, 43 require a torque greater than the peak torque.

In terms of best performing schemes, this chart correlates with the results seen in Fig.2.21. In contrast, however this chart, shows the effect of the lower torque limit for joint 5. For instance, from Fig.2.21, it is seen that scheme no 71 has an acceptable dynamic performance. However Fig.2.22 shows a high number of violations of nominal torque.

The torque required is proportional to the desired acceleration, therefore by increasing t_{final}, the time taken to complete the trajectory, the maximum torque can be reduced. Taking this into account, Fig.2.22 also indicates the actuation scheme that are capable of transporting the object in the least amount of time without violating the motor constraints.

2.3.8 Actuation Scheme Selection

The use of any inadmissible scheme can be immediately ruled out. As we have seen in Section 2.3.3, there are 39 such schemes in which the object has an uncontrollable DOF. Recall that to avoid this scenario, either joint 1 or joint 6 must be actuated. The joint configuration, $\theta_1 = \theta_6$ generates an actuation singularity when both q_2 and q_7 are actuated. This would restrict the range of the base joints leading to a large reduction in the workspace. By actuating either q_3 or q_5 on the right arm and q_8 or q_{10} on the left arm, the lack of stiffness linked to the inner singularity can be avoided. This lack of stiffness will occur when either of the arms pass through the arm singularity.

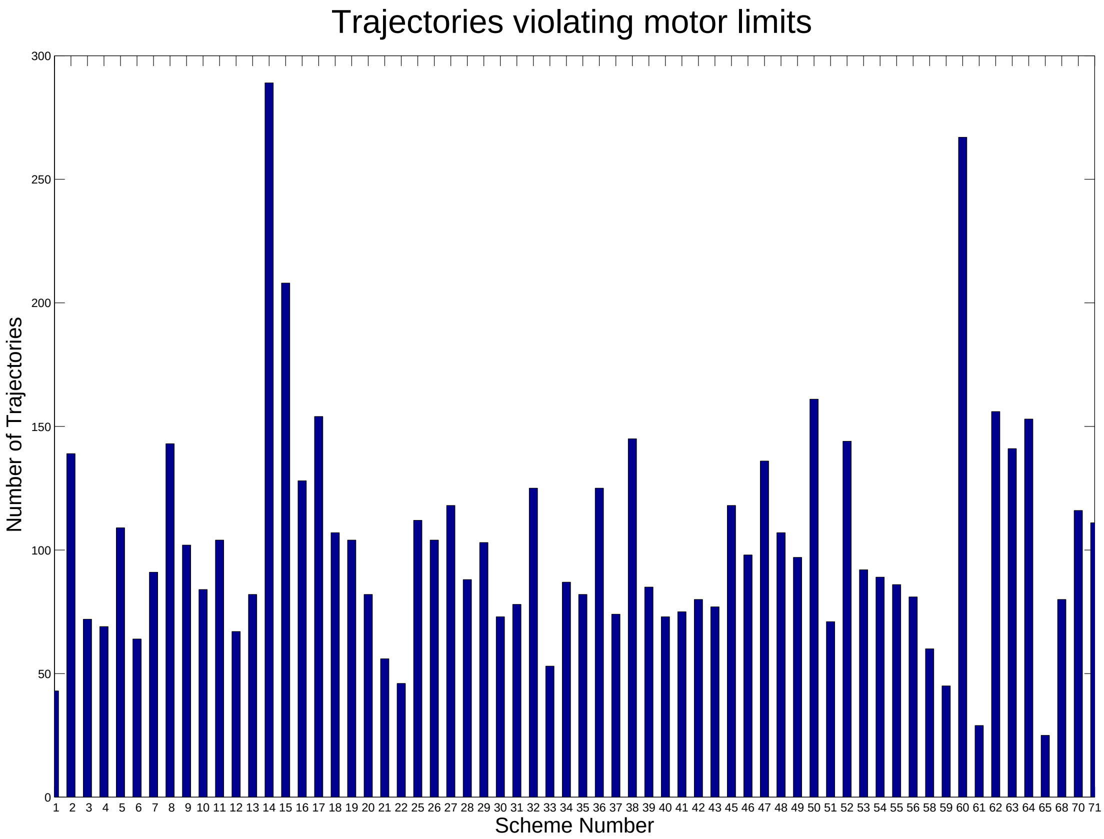

Figure 2.22: Trajectories that violate torque constraints

From studying the results of the power loss test, it is clear that the best performing schemes actuate both joint 1 and joint 6. Moreover there is a clear and logical tendency for schemes that contain actuators near the base to preform better, for example scheme number 6, 11010—10000 and scheme number 22, 11000—10100. Finally, Fig. 2.22, clearly shows that if the real motor parameters are taken into account, schemes that actuate either joint 5 or joint 10 should be avoided.

In order to propose an optimum actuation scheme, both the kinematic and dynamic considerations must be used. As shown in Fig.2.21, actuation scheme 41, 10100—10001 has a good dynamic performance. Despite the fact that joint ten is actuated, relative to the other schemes the percentage of peak torque violations is low. Furthermore the selected actuated joints, avoid the loss of stiffness associated with the singular configurations. Finally the scheme distributes the motors equally in the two arms.

2.4 Conclusion

This chapter focuses on the cooperative manipulation of rigid objects. A state of the art concerning cooperative manipulators is given. The cooperative manipulation schemes are divided into two classes, full mobility and lower mobility. For full mobility cooperative manipulators the object can be positioned arbitrarily in the operational space while fully regulating the internal forces. The redundant actuators of the system can also to used to optimize secondary system criteria or ensure that the object experiences no internal forces. Several force/position control schemes are described that are capable of controlling the object forces. For lower mobility cooperative manipulators, the effect of two cooperating robots on the object mobility is difficult to discern without a detailed analysis. The main analysis tools to deal with such systems are presented.

The contribution of this chapter is an analysis of a lower mobility cooperative manipulator grasping a rigid object. The DOF of the grasped object are explored using screw theory. The infinitesimal motion of the object, for different configurations are investigated. It is shown that the nature of this motion may change throughout the workspace due to the relationship between the constraint wrenches. The serial singularities of each arm are shown and an investigation into their possible effect on the closed chain system is undertaken. The screw theory analysis demonstrates that the effect of serial singularities in closed loop is dependent on the actuation scheme. This is validated by a numerical comparison. The constraint singularity due to the closed loop structure is illustrated and the nature of the resulting motion is described. The motion is more complex than a loss of stiffness since the arms are simultaneously in a serial singularity configuration. The results from a corresponding numerical analysis are difficult to interpret due to the complexity of this case. A detailed investigation into the choice of actuated joints is carried out. By considering the wrenches exerted by the actuators on the object, all admissible actuation schemes can be enumerated. In addition the schemes considered inadmissible and the causes of this inadmissibility, are illustrated in detail. The dynamic model of the closed chain system is developed. The model is used to assess the performance with respect to a power loss criterion and motor capacities over many trajectories. This analysis permits the selection of feasible actuation schemes with respect to their dynamic capabilities.

Although this chapter presents an analysis of a specific cooperative system, certain contributions can be generalized to a range of closed chain systems. The comparison between the numerical methods based on the serial kinematic Jacobian matrices of each arm and the screw theory, demonstrates that while numerical comparison can be used to obtain the location of singular configurations, screw theory is required to gain a deeper understanding of the causes and resulting behavior. In addition to this, the importance of the actuation scheme to the dynamic performance of the manipulator is shown. In particular, it is shown that a judicious choice of actuation scheme can result in a large difference in the power loss of trajectories.

The work presented in this chapter has led to the publication of two papers presented at international conferences [LKC12a] [LKC12b] and one journal article[LKC14].

3

Cooperative Manipulation of Flexible Objects

3.1 Introduction

This chapter concerns the cooperative manipulation of deformable objects. This includes robots with flexible links and robots whose payloads can undergo deformation. The deformation may be an unwanted result of the motion or, in contrast, the robot may have to deform a body in order to complete the desired task. In both cases, to achieve a high level of task precision the effects of the flexibility must be taken into account.

For the first class of systems, i.e. robots with flexible components, the flexibility has many advantages over classical rigid multi-body systems. For example, by manufacturing the system using lightweight material the overall mass of a manipulator can be greatly decreased. This allows the system to attain high accelerations and reduce energy consumption. Furthermore a robot with light, flexible links is inherently safer than its rigid counterpart and thus permits closer cooperation with human workers. The Kuka lwr robot is an example of a robot where the added flexibility is used for closer human-robot interaction.

On the other hand the flexibility in the links may be an unintended consequence of the task definition. For example, space robots operating in orbit must manipulate very large loads over a sizable area. To operate throughout the work space, the limbs must be of a great length leading to flexibility when transporting a load.

Likewise high speed machining operations can induce large vibrations in the robot that in turn can lead to imprecision in the task execution. The standard method to re-

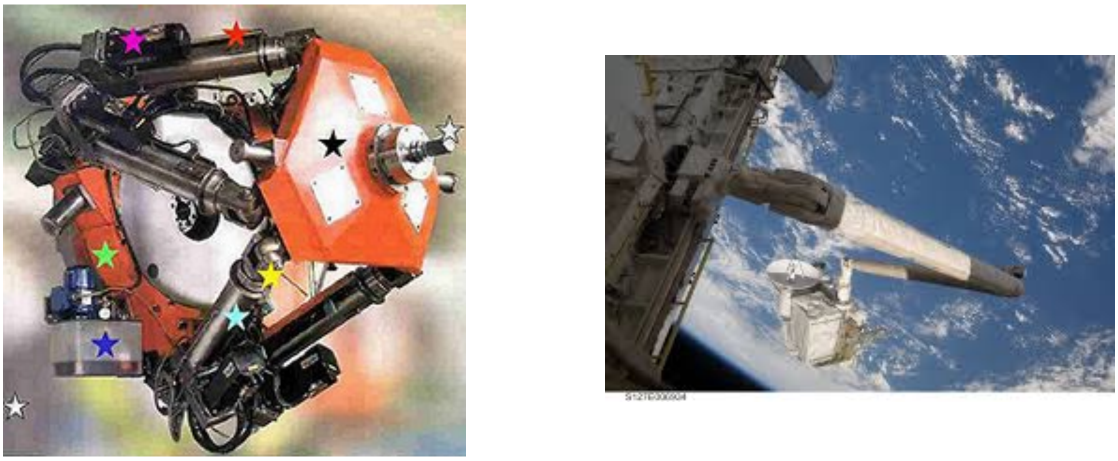

Figure 3.1: (a) Hexapode Robot, (courtesy of CMW) (b) remote manipulator system (RMS) robot arm of the Space Shuttle Endeavour

ducing this vibration is to increase the weight of the tool, leading to mechanisms with considerable mass. For example, the hexapode robot shown in Fig.3.1, a 5 DOF parallel manipulator principally used for milling applications, has a mass of over 900kg. Other standard tools can greatly exceed this mass. If the mass is reduced, the flexibility must be taken into account after which a model can be used to predict and damp the vibration of the system. If the flexibility is limited to small deformations, the system can be modeled using methods based on equivalent rigid body models and controllers can be implemented based on standard algorithms.

The second class of systems are robots whose payloads can undergo deformation and where the deformation is part of the task definition. In this case, in contrast to the rigid object manipulation, the robot must change the inner state of the payload. There are many potential applications of solutions in this scenario, for example in the textile industry, the food processing industry, and in medical robotics. Typically for these applications, the object deforms in a large non-linear fashion in response to external forces. Moreover, the variability of the target object means that manipulation solutions need to use sensor rather than model based control.

With respect to cooperative manipulators, as described in Chapter 2, the robots, the object and the ground form a closed chain system. The object can then be treated as a link which must satisfy closed chain constraints. However in contrast to the systems described in Chapter 2, the object can now undergo deformations in response to either system dynamics or violations of the closed chain constraints.

This chapter, excluding the introduction, is divided into three sections, the state of the art is given in Section 3.2, the contributions of this thesis in this field are given in Sections 3.3 and 3.4 and the conclusion is given in Section 3.5. The outline of each section is given in the following.

State of the Art

Firstly, in Section 3.2, a state of the art of cooperative manipulation of deformable objects is described. This section is divided into the following subsections. In Section 3.2.1, a broad introduction and definition of deformable objects is given. In Sections 3.2.2, 3.2.3,and 3.2.4 the primary means of modeling deformable objects and examples of their use in robotic formulations are outlined. In Section 3.2.5, the generalized Newton Euler model, a complete modeling strategy for robots whose links are flexible, is described. Section 3.2.6 focuses on the control of deformable objects by cooperative manipulators, including the control of the object's shape.

Contribution

This contribution of this chapter, given in Section 3.3, is the derivation of a method to calculate the direct and inverse dynamic models of a closed chain system where the object is flexible [LKM15]. A general algorithm is presented that decouples the robot into rigid and flexible sub-systems. The systems are linked by calculating the reaction wrench at the grasp position. In Section 3.3.1, the general system is presented. The dynamics of the rigid sub-system i.e. the serial arms of the cooperative system, are given in Section 3.3.2. The grasped object is considered as a flexible body and modeled by distributed flexibility using generalized Newton-Euler model in Section 3.3.3. Section 3.3.4 describes how the dynamic model of the flexible robot is derived. In Section 3.3.5, a numerical simulation validating the proposed model is given for a set of different objects. In this section the derived model is compared with a system constructed using commercial software. Finally, in Section 3.3.6, a classification method of objects used in robotic manipulation is given.

In order to demonstrate how this model can apply to a wide range of robotic systems, in Section 3.4 a step by step approach is illustrated for a Gough-Stewart 6-DOF parallel robot [LKM14a]. In Section 3.4.1, the overall procedure is outlined, as well as the prescribed solution. In Section 3.4.2 the geometric, kinematic and dynamic models of the Gough-Stewart robot's legs are presented. The platform of the parallel manipulator is considered as a flexible body and modeled in Section 3.4.3. Section 3.4.4 describes how the inverse dynamic model and direct dynamic model of the flexible robot are derived. The direct dynamic model gives the elastic and Cartesian accelerations in terms of the input torques and the current state of the system i.e. the position and velocities of both the rigid and elastic variables. The inverse dynamic model calculates the elastic accelerations and the actuator torques from the current state variables and the desired acceleration of the platform. In Section 3.4.5, a numerical simulation validating the proposed model is given. In this section the derived model is compared with an identical system constructed using commercial software.

3.2 State of the Art: Robotic Manipulation of Deformable Objects

3.2.1 Deformable Object Definition

Suppose there exists a number of fixed frames on an object. The object can be classified according to the relative behavior between these frames. Three possible types of object behavior can be defined:

Rigid: There is no relative motion between the frames, in spite of the object motion or external forces.

Deformable: There is relative motion between frames where the motion of one point can be obtained from the motion of the second point using information about the object's properties.

Articulated: The frames can move freely with respect to each other, where the velocity of a frame has no effect on the velocity of a different frame.

Deformation in objects can occur for two reasons. Firstly the application of external forces either directly or due to the acceleration of the body. Secondly due to a change in temperature of the body. This thesis is limited to the first case, i.e. objects whose deformation is due to force.

Fig.3.2 shows a comparison of a rigid body moving in space versus a deformable body moving in space. A point on the object is denoted as $\mathbf{p}$ before the motion and $\mathbf{p}'$ after the motion. The vector $\mathbf{r}_p$ denotes the position of this point with respect to the object frame, $\mathcal{R}_{obj}$. For the rigid object the vector $\mathbf{r}_p$ is constant for every point. Therefore the position of $\mathbf{p}$ in the world frame, $\mathcal{R}_0$, can be reconstructed as follows:

$$\mathbf{p}_0 =^0 \mathbf{T}_{obj} \begin{bmatrix} \mathbf{r}_p \\ 1 \end{bmatrix} \tag{3.1}$$

It is obvious that at any moment during the trajectory the rigid object is completely defined by frame $\mathcal{R}_{obj}$. In contrast to this, for the deformable object, the vector $\mathbf{r}_p$ is no longer constant but is dependent of the state of the object. The number of variables required to obtain the position of every point on the object is known as the object's internal DOF. Thus the general definition of a deformable body is a body for which the relative position between any two points is variable.

In the following sections, the main modeling strategies used to represent the deformable behavior are outlined. It should be noted that often the techniques described in the following sections are combined together to better represent the system.

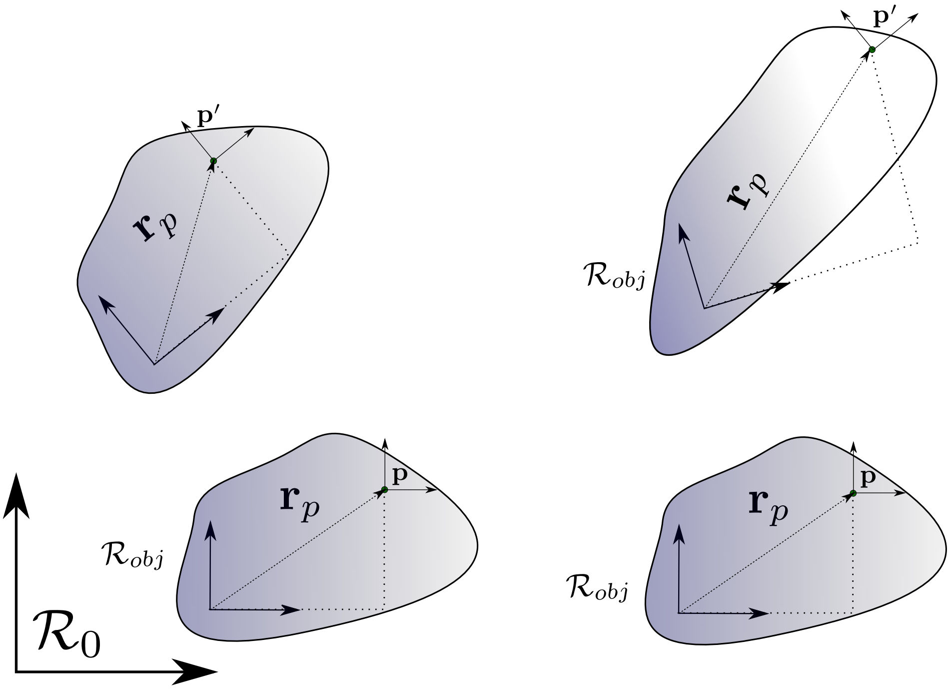

Figure 3.2: Rigid object versus deformable object motion

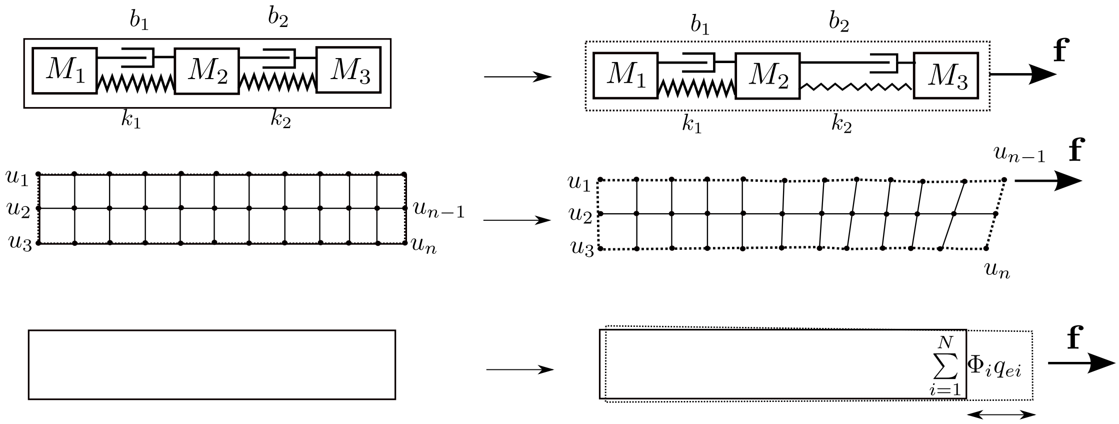

Figure 3.3: Deformable Object Models, from left to right (i) Mass spring damper system, (ii) Finite Element Analysis, (iii) Modal superposition

3.2.2 Modeling using Lumped Parameter Methods

Lumped parameter models, also known as mass spring damper models, describe a deformable object by a set of discrete point masses. The masses are linked by springs and dampers. Fig. 3.3 shows an object modeled by a mass spring damper system. Though this is an extremely simple system, the object in question is represented by three point masses, it can be used to illustrate the advantages and drawbacks of the mass spring damper model.

Suppose a force f is applied to the body at M_1 In order to solve the system, a free body diagram is constructed for each node of interest, normally at the point mass. For example the force at M_1 is defined as:

$$f = M_1\dot{v}_1 - b_1 v_1 - k_1(x_1 - x_2) \tag{3.2}$$

Similar balance equations can be written for the remaining masses of the system. By doing so, the complete state of the system can be described by a series of differential equations.

From the above example, the ease of modeling deformations using the mass spring damper approach is demonstrated. This technique can be used to represent a wide range of deformable behaviors. As discussed in Chapter 4, Section 4.2, the main drawback of this system is the difficulty in accurately modeling realistic object behavior. The values of M, k and b must be carefully chosen, leading to parameter identification models. Furthermore, the system does not preserve the volumetric properties of a object during deformation.

Lumped Parameter Methods in Robotic Systems

In spite of the limitations described in the previous section, for multi-body systems with limited elasticity, mass-spring damper systems have generally been used to represent joint flexibility [KG00, PGP01].

In [KG00], a complete description of robotic systems with lumped elasticity is given. The authors shown how geometric, kinematic and dynamic models of the system can be obtained that are similar in form to the equivalent rigid body models. This approach deals with systems of limited elasticity, in particular in cases where the elasticity is due to joint deformations, a subject that is also treated in [Spo87]. Due to the assumption of joint elasticity, the number of generalized coordinates is doubled leading to a dynamic model as follows:

$$\begin{bmatrix} \Gamma \\ \mathbf{0} \end{bmatrix} = \mathbf{A} \begin{bmatrix} \ddot{\mathbf{q}} \\ \ddot{\mathbf{q}}_e \end{bmatrix} + \begin{bmatrix} \mathbf{h} \\ \mathbf{h}_e \end{bmatrix} + \begin{bmatrix} 0 \\ \mathbf{k}_e \Delta \mathbf{q}_e \end{bmatrix} \tag{3.3}$$

where $\mathbf{k}_e$ is the joint stiffness matrix, $\mathbf{h}_e$ the Coriolis and Centrifugal forces due to the elastic variables and $\mathbf{q}_e$ the coordinates of elasticity. The main drawback of this method is the limited representation of flexibility in the system. By increasing the number of elastic elements in the system, the solution begins to resemble finite element models described in Section 3.2.4 leading to larger computational times and thus losing the main attraction of the strategy.

Mass-spring damper systems have also been used to describe an object that is manipulated by a cooperative system, for instance in [WHKK01] and [DDDBV93]. In contrast to the previous works, for cooperative systems the goal is to control the state of the object. Since the model of the manipulated object is not well known, the spring damper representation is sufficient when complemented with sensor or learning algorithms.

3.2.3 Modeling using Modal Analysis

Modal analysis is a technique used to discover the shape of the vibration of an object and the natural frequency of this vibration. The basic assumption is that the displacement of the body at any point is a sum of the rigid body displacement and the flexible body deformation. A mode shape is independent of other mode shapes and depends on the object's shape, material properties and boundary conditions.

Mode shapes and their natural frequencies can be obtained analytically, for simple objects such as beams and plates. This gives a continuous description of the object's deformation. Alternatively modal analysis can also be performed using Finite element techniques as shown in Section 3.2.4. In this case the mode shapes act as a transformation from a smaller space, the modal or generalized coordinates, to a much larger space, the nodal displacements. Finally, in order to obtain the mode shapes experimentally a shaker is used to apply a sinusoidal time varying force while an accelerometer is used to measure the response.

Modal Analysis in Robotic Systems

For multi-body dynamics, modal analysis is very widely used. The principal difficulty in using model analysis is obtaining the analytic solution to the continuous modal functions for a given body. Generally these solutions are only available for simple bodies. Therefore a common technique to combine modal analysis with robots is to treat the grasped object or manipulator link as a beam.

The behavior of the beam can be easily modeled using the Euler–Bernoulli beam theory. By using this theory an analytic description of the free vibration of the beam is given as:

$$EI\frac{\partial^4 w}{\partial x^4} + \rho a \frac{\partial^2 w}{\partial t^2} = 0 \tag{3.4}$$

where E, I, w, x, ρ, a are the values for the Elastic Modulus, second moment of area, deflection, distance along beam, density and cross-sectional area respectively. This equation can be solved using the Fourier decomposition to obtain characteristic equation of the beam. The assumed modes for the beam are described as:

$$\Phi(x) = a_1 \cosh(\beta x) + a_2 \cos(\beta x) + a_3 \sinh(\beta x) + a_4 \sin(\beta x) \tag{3.5}$$

where a_i, for $i = 1 \ldots 4$, is a constant that depends on the boundary conditions and β is known as the wave number which is a functions of the frequency. The following common boundary conditions at either end of the beam can be used to solve the characteristic equation:

- Clamped at 0, $\rightarrow \Phi(0) = 0$, $\dot{\Phi}(0) = 0$
- Free at 0, $\rightarrow \ddot{\Phi}(0) = 0$

By substituting the boundary condition values into (3.5), an infinite number of mode shapes can be obtained. Generally modes of very high frequency are ignored, the highest frequency, known as cut off frequency, is determined by the application.

Since, the structure of links in a serial manipulator closely resemble a beam, this modeling strategy has been widely used in research, for example in [Boo84]. Examples of the use of modal analysis for serial manipulators, parallel manipulators can be found in [BC96] and [BK13].

Due to the simplicity of the beam equations, many works have focused on cooperative manipulators handling grasping a beam like object. The typical case is where the beam is modeled using clamped-free boundary conditions [SL97, TEJ09]. However, cases where the ends of the beam are clamped clamped[AYAH07] or pinned-pinned [EBS13] have also been modeled.

It should be noted, that while in reality both ends of the beam are fixed rigidly to the cooperative manipulators, the choice of boundary conditions depends only on the modeling strategy of the system. The methods described in the above section can be applied to different types of object provided a description of the flexibility is available.

3.2.4 Finite Element Methods

The Finite Element method (FEM) refers is a numerical technique that uses a series of finite elements connected together at discrete points called nodes. For the analysis of an object using FEM, the first step is to discretize the body into multiple elements as shown in Fig. 3.3. The greater the number of elements the larger the resolution of the solution. The location where two elements meet is marked by a node; in this case there are n nodes. The second step is to create a function that approximates the displacement, denoted u_i of node i. A linear equations is created that solves for these displacements using the applied nodal forces $f_1 \dots f_n$, typically:

$$\mathbf{K}\begin{bmatrix} u_1 \\ u_2 \\ \vdots \\ u_n \end{bmatrix} = \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix} \tag{3.6}$$

The above expression represents only static considerations and gives a linear relationship between nodal forces and nodal displacements. If the effects of dynamics are included in the analysis, the behavior of the body is given as follows:

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{C}\dot{\mathbf{u}} + \mathbf{K}\mathbf{u} = \mathbf{f} \tag{3.7}$$

where $\mathbf{u}$, $\dot{\mathbf{u}}$ and $\ddot{\mathbf{u}}$ are nodal displacements, velocities and acceleration. $\mathbf{M}$, $\mathbf{C}$ and $\mathbf{K}$ contain the sum of the mass, damping and stiffness of all elements of the object respectively. For an arbitrarily shaped object the modal analysis is carried out as follows. Equation (3.7) is rewritten for a free vibration study i.e. $\mathbf{f} = 0$ and without the damping effects:

$$\mathbf{M}\ddot{\mathbf{u}} + \mathbf{K}\mathbf{u} = 0 \tag{3.8}$$

By assuming the vibration is sinusoidal, $u = a\cos(\omega t)$, the following relation is obtained:

$$\ddot{\mathbf{u}} = -\omega^2\mathbf{u} \tag{3.9}$$

Substituting (3.9) into (3.8):

$$\left[\mathbf{K} - \omega^2\mathbf{M}\right]\mathbf{u} = 0 \tag{3.10}$$

where ω is known as the natural frequency or eigenvalue of the system. ω^2 can be obtained by solving the characteristic equation $det\left[\mathbf{K} - \omega^2\mathbf{M}\right] = 0$. For each natural frequency i, there is a corresponding mode shape $\mathbf{\Phi}_i$. By obtaining this mode shapes (or eigenvectors), the system can be rewritten in terms of its generalized coordinates as follows:

$$\mathbf{u} = \sum_{i=1}^{N}\mathbf{\Phi}_i q_{ei} = \mathbf{\Phi}\mathbf{q}_e \qquad \dot{\mathbf{u}} = \sum_{i=1}^{N}\mathbf{\Phi}_i \dot{q}_{ei} = \mathbf{\Phi}\dot{\mathbf{q}}_e \qquad \ddot{\mathbf{u}} = \sum_{i=1}^{N}\mathbf{\Phi}_i \ddot{q}_{ei} = \mathbf{\Phi}\ddot{\mathbf{q}}_e \tag{3.11}$$

where q_{ei}, $\dot{q}_{ei}$ and $\ddot{q}_{ei}$ denote the position, velocity and acceleration of the generalized elastic coordinates of the object, $\mathbf{q}_e$, $\dot{\mathbf{q}}_e$ and $\ddot{\mathbf{q}}_e$ are the corresponding vectors and $\mathbf{\Phi} = [\mathbf{\Phi}_1 \dots \mathbf{\Phi}_N]$

The generalized mass and stiffness for mode i of the system are obtained as:

$$\mathbf{\Phi}_i^T \mathbf{M} \mathbf{\Phi}_i = m_{eei} \qquad\qquad \mathbf{\Phi}_i^T \mathbf{K} \mathbf{\Phi}_i = k_{eei} \tag{3.12}$$

where $\mathbf{m}_{eei}$ and $\mathbf{k}_{eei}$ are the generalized mass and stiffness associated with the i^{th} mode. The mode shapes are not unique but can be scaled in order to obtain a desired form. The final step in a modal analysis is the scaling of the mode shapes to obtain the mass ortho-normalized mode shapes. By doing so the following relationships are obtained for any mode i:

$$\mathbf{\Phi}_i^T \mathbf{M} \mathbf{\Phi}_i = 1 \qquad\qquad \mathbf{\Phi}_i^T \mathbf{K} \mathbf{\Phi}_i = k_{eei} = \omega_i^2 \tag{3.13}$$

Other quantities, for example the stress, can be obtained using the nodal displacement and the material properties. FEM is unrivaled it the ability to accurately represent complex geometries and with the increase of computation power the large number of nodes required to represent the system is no longer a critical limitation.

Finite element Methods in Robotic Systems

In multi-body system, finite element methods are generally used in conjunction with modal analysis to represent the flexibility of the system. The biggest drawback of the finite element method is the high order of the dynamic equations, however there exists methods to reduce this dimension, principally Guyan reduction [Guy65] and component mode synthesis [BC68]. A comparison between the two reduction methods is carried out [SSJ88] which shows that in general while both methods are equally viable the component mode synthesis is more efficient, thus is more widely used in the robotic community.

An example of Guyan reduction is given in [MI96], where a multi-body system used in the automotive industry carries out the assembly of flexible materials. PD, computed torque and hybrid position/force control schemes are validated on the system. Using the Guyan reduction (3.7) is partitioned into nodes that have a large effect on the stiffness and inertia, denoted as master nodes a represented with subscript m, and nodes that have a negligible effect, denoted as slave nodes and represented with a subscript s:

$$\begin{bmatrix} \mathbf{M}_{mm} & \mathbf{M}_{ms} \\ \mathbf{M}_{sm} & \mathbf{M}_{ss} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{u}}_m \\ \ddot{\mathbf{u}}_s \end{bmatrix} + \begin{bmatrix} \mathbf{C}_{mm} & \mathbf{C}_{ms} \\ \mathbf{C}_{sm} & \mathbf{C}_{ss} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{u}}_m \\ \dot{\mathbf{u}}_s \end{bmatrix} + \begin{bmatrix} \mathbf{K}_{mm} & \mathbf{K}_{ms} \\ \mathbf{K}_{sm} & \mathbf{K}_{ss} \end{bmatrix} \begin{bmatrix} \mathbf{u}_m \\ \mathbf{u}_s \end{bmatrix} = \begin{bmatrix} \mathbf{f}_m \\ \mathbf{f}_s \end{bmatrix} \tag{3.14}$$

Using the *Guyan* transformation matrix, $\mathbf{T}_{guy}$, the stiffness and inertial effects associated with $slave$ nodes are transferred to the master nodes. Therefore the dynamics of the system can be entirely written in terms of the masters nodes:

$$\hat{\mathbf{M}}\ddot{\mathbf{u}}_m + \hat{\mathbf{C}}\dot{\mathbf{u}}_m + \hat{\mathbf{K}}\mathbf{u}_m = \hat{\mathbf{f}} \tag{3.15}$$

where

$$\begin{aligned} \hat{\mathbf{M}} &= \mathbf{T}_{guy}^T \mathbf{M} \mathbf{T}_{guy} & \hat{\mathbf{C}} &= \mathbf{T}_{guy}^T \mathbf{C} \mathbf{T}_{guy} \\ \hat{\mathbf{K}} &= \mathbf{T}_{guy}^T \mathbf{K} \mathbf{T}_{guy} & \hat{\mathbf{f}} &= \mathbf{T}_{guy}^T \mathbf{f} \\ \mathbf{T}_{guy} &= \begin{bmatrix} \mathbf{1}_m \\ \mathbf{K}_{ss}^{-1} \mathbf{K}_{sm} \end{bmatrix} \end{aligned} \tag{3.16}$$

Therefore, the order of the equations is reduced by the dimension of $\mathbf{u}_s$.

Recently however, component mode synthesis has become the standard way of representing multi-body systems with flexible parts using finite elements. Examples of the use of component mode synthesis is given in [ZZG00], where the controllability of a robot with flexible payload is studied and [PM02] where an actuated gripper grasps a flexible object. For component mode synthesis, (3.7) is partitioned into variables that represent the internal DOF of the system, denoted as $\mathbf{u}_i$, and variables that represent the interface (juncture) coordinates, denoted as $\mathbf{u}_j$.

$$\begin{aligned} &\begin{bmatrix} \mathbf{M}_{ii} & \mathbf{M}_{ij} \\ \mathbf{M}_{ji} & \mathbf{M}_{jj} \end{bmatrix} \begin{bmatrix} \ddot{\mathbf{u}}_i \\ \ddot{\mathbf{u}}_j \end{bmatrix} + \begin{bmatrix} \mathbf{C}_{ii} & \mathbf{C}_{ij} \\ \mathbf{C}_{ji} & \mathbf{C}_{jj} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{u}}_i \\ \dot{\mathbf{u}}_j \end{bmatrix} \\ &\qquad + \begin{bmatrix} \mathbf{K}_{ii} & \mathbf{K}_{ij} \\ \mathbf{K}_{ji} & \mathbf{K}_{jj} \end{bmatrix} \begin{bmatrix} \mathbf{u}_i \\ \mathbf{u}_j \end{bmatrix} = \begin{bmatrix} \mathbf{f}_i \\ \mathbf{f}_j \end{bmatrix} \end{aligned} \tag{3.17}$$

The static behavior of the system is given as

$$\begin{bmatrix} \mathbf{K}_{ii} & \mathbf{K}_{ij} \\ \mathbf{K}_{ji} & \mathbf{K}_{jj} \end{bmatrix} \begin{bmatrix} \mathbf{u}_i \\ \mathbf{u}_j \end{bmatrix} = \begin{bmatrix} \mathbf{f}_i \\ \mathbf{f}_j \end{bmatrix} \tag{3.18}$$

The reduction is achieved by replacing the internal nodal coordinates with a small set of modal coordinates. The first step is to obtain the relationship between the internal variables and the variables at the juncture. The internal variables can be divided into those associated with the juncture variables, denoted as $\mathbf{u}_{ij}$, and those that can be represented by a small set of modal coordinates, denoted as $\mathbf{u}_{ii}$:

$$\mathbf{u}_i = \mathbf{u}_{ij} + \mathbf{u}_{ii} \tag{3.19}$$

$\mathbf{u}_{ij}$ are the displacements of the internal variables due to the displacement at the juncture when there are no internal forces on the body i.e. $\mathbf{f}_i = 0$. Therefore setting the internal forces to zero, (3.18) becomes:

$$\mathbf{u}_{ij} = -\mathbf{K}_{ii}^{-1} \mathbf{K}_{ij} \mathbf{u}_j \tag{3.20}$$

The next step is to solve the characteristic equation (3.8), when there is zero displacement at the interface, $\mathbf{u}_j = 0$:

$$\mathbf{M}_{ii} \ddot{\mathbf{u}}_{ii} + \mathbf{K}_{ii} \mathbf{u}_{ii} = 0 \tag{3.21}$$

This expression is solved in the same manner as (3.11) to obtain the reduced form:

$$\mathbf{u}_{ii} = \mathbf{\Phi}\mathbf{q}_e \tag{3.22}$$

Finally, using (3.19), (3.20) and (3.22), the linear transformation between modal coordinates and the component mode synthesis space is given as:

$$\begin{bmatrix} \mathbf{u}_i \\ \mathbf{u}_j \end{bmatrix} = \begin{bmatrix} \mathbf{\Phi} & -\mathbf{K}_{ii}^{-1}\mathbf{K}_{ij} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{q}_e \\ \mathbf{u}_j \end{bmatrix} = \mathbf{T}_{cms}\mathbf{q}_{cms} \tag{3.23}$$

Finally, the dynamic equation of the system is written as:

$$\hat{\mathbf{M}}\ddot{\mathbf{q}}_{cms} + \hat{\mathbf{C}}\dot{\mathbf{q}}_{cms} + \hat{\mathbf{K}}\mathbf{q}_{cms} = \hat{\mathbf{f}} \tag{3.24}$$

where

$$\begin{aligned} \hat{\mathbf{M}} &= \mathbf{T}_{cms}^T\mathbf{M}\mathbf{T}_{cms} & \hat{\mathbf{C}} &= \mathbf{T}_{cms}^T\mathbf{C}\mathbf{T}_{cms} \\ \hat{\mathbf{K}} &= \mathbf{T}_{cms}^T\mathbf{K}\mathbf{T}_{cms} & \hat{\mathbf{f}} &= \mathbf{T}_{cms}^T\mathbf{f} \end{aligned} \tag{3.25}$$

3.2.5 Generalized Newton-Euler Approach

Overview of Generalized Newton-Euler Approach

In Sections 3.2.2, 3.2.3, and 3.2.4, an overview is given of the primary means of modeling deformable objects. In this section, we focus on a complete formulation where the deformable object is integrated into a robotic system.

There are several methods of modeling flexible components that are in embedded in a multi-body system. A review of the main methods including implementations, applications, advantages and disadvantages is given in [WN03].

The first and most widely used approach is known as the *floating-frame* system [Sha13] which is typically used in industrial applications. A floating frame is used to approximate the movement of each rigid body in the chain. This frame is generally fixed to joint frame. In this case, the deformation of the body can be viewed as linear perturbations to the reference frame motion [BKBLV07] and thus modal reduction methods outlined in Sections 3.2.3,and 3.2.4 can be employed. The drawback of this approach is the inability to deal with large deformations. On the other hand, the approach is easily extended to existing multi-body dynamic formulations [Boo84].

The second method is known as the *inertial frame* approach. In contrast to the *floating-frame* system, the *inertial frame* approach is based on fixed reference frame. The advantage of this approach is the ability to handle large deformations in an accurate manner. The disadvantage however is the increase in computation time and the fact that the approach is incompatible with modal analysis. Furthermore this formalism is

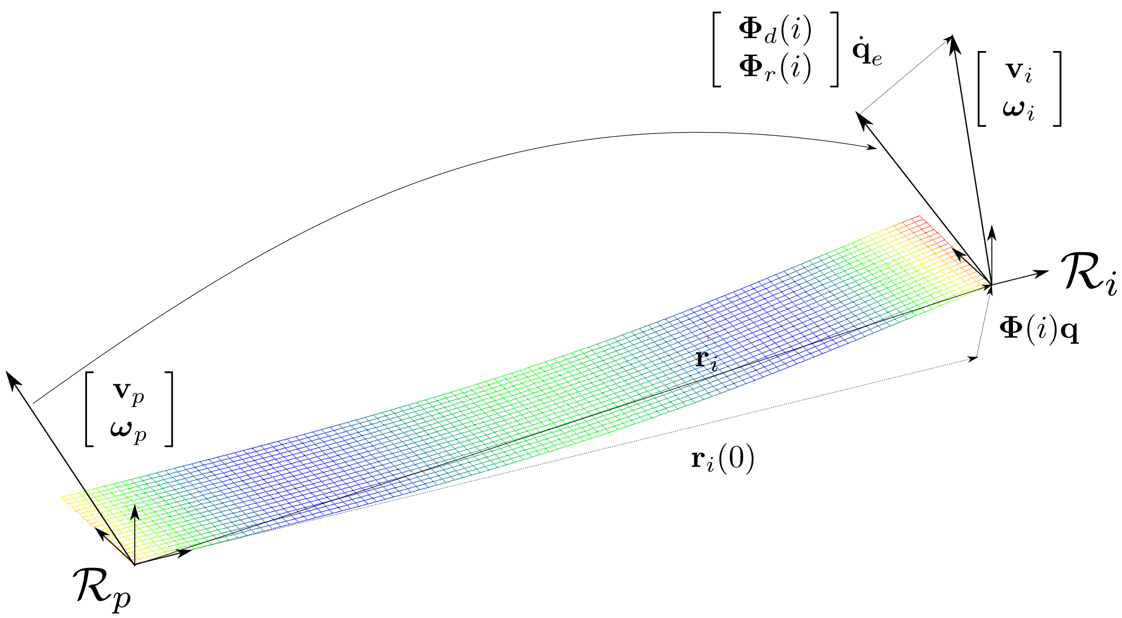

Figure 3.4: Deformable body represented by modal functions

difficult to integrate with existing multi-dynamic approach and can lead to a very high level of complexity.

In this section, an overview is given of the Generalized Newton-Euler Approach, also sometimes referred to as modal superposition method for robot manipulators. The advantage of this approach is two-fold. Firstly, the recursive nature of the calculations mean that the model can be quickly computed. Secondly, the formulation is very suitable for robot control algorithms.

The overall idea is to unite two different formalisms. The Lagrangian formalism deals with the deformation state and elastic coordinates of the body while the Euler variables deal with the rigid motions of the body. The elastic variables depend on the current configuration of the body and the reference or undeformed configuration. On the other hand, the rigid variables are defined by their velocity at an instant t.

This Generalized Newton-Euler Approach is a very adaptable means of modeling a system's flexibility and has been used for serial manipulators [BK98] and parallel manipulators [BK13]. In the following sections an outline of the Generalized Newton-Euler Approach is given. In contrast to the modal superposition relationships given in Section 3.2.1, the deformable object is now subject to global non-linear rigid body motion which must be taken into account when calculating its flexibility.

Kinematic Model of Deformable Body

Consider a free flexible body as shown in Fig.3.4. The position of any point of the flexible body is obtained as a sum of the position of the local body reference frame, denoted as $\mathcal{R}_p$, and the deformation of the body with respect to this frame at frame $\mathcal{R}_i$.

$$\mathbf{p}_i = \mathbf{p}_p + \mathbf{r}_i \tag{3.26}$$

$$\mathbf{r}_i = \mathbf{r}_i(0) + \sum_{k=1}^{N} \mathbf{\Phi}_{dk}(i) q_{ek} \tag{3.27}$$

It should be noted that the mode shapes functions have been divided into those that relate the modal coordinates to a position change, denoted as $\mathbf{\Phi}_{dk}$ and those that relate the modal coordinates to a change in orientation change denoted as $\mathbf{\Phi}_{rk}$. These terms are not independent instead they are related to each other by the curl operator [BKBLV07]:

$$\mathbf{\Phi}_{dk}(i) = \frac{1}{2}\nabla \times \mathbf{\Phi}_{rk}(i) \tag{3.28}$$

As long the deformation of the object respects the conditions of modal superposition, the above expressions can be used to obtain the position of every point of the deformable object. By obtaining the derivative of this equation:

$$\dot{\mathbf{r}}_i = \boldsymbol{\omega}_p \times \mathbf{r}_i + \sum_{k=1}^{N} \mathbf{\Phi}_{dk}(i) \dot{q}_{ek} \tag{3.29}$$

Equation (3.29) gives the change in the relative position of point i. It should be noted that the change in position is also a function of the rigid body velocity. The kinematic twist of the flexible body at point i is given as the the effect of the rigid body motion transformed to that point plus the effect of flexibility evaluated at that point.

$$\begin{bmatrix} \mathbf{v}_i \\ \boldsymbol{\omega}_i \end{bmatrix} = {}^{i}\mathbf{S}_p \begin{bmatrix} \mathbf{v}_p \\ \boldsymbol{\omega}_p \end{bmatrix} + \begin{bmatrix} \mathbf{\Phi}_d(i) \\ \mathbf{\Phi}_r(i) \end{bmatrix} \dot{\mathbf{q}}_e \tag{3.30}$$

$$\mathbf{\Phi}_d(i) = \begin{bmatrix} \Phi_{d1}(i) & \Phi_{d2}(i) & \dots & \Phi_{dN}(i) \end{bmatrix}$$

$$\mathbf{\Phi}_r(i) = \begin{bmatrix} \Phi_{r1}(i) & \Phi_{r2}(i) & \dots & \Phi_{rN}(i) \end{bmatrix} \tag{3.31}$$

Dynamic Model of Deformable Body

The dynamic model of the flexible body, developed by Boyer in [Boy94], is obtained using the principle of virtual powers. The dynamic model of the flexible body is obtained using the principle of virtual powers. The premise of the principle power, similar to that of the well known principle of virtual work, is that the virtual power of the acceleration of the system is equal to the virtual power of the internal loads plus the virtual power of any external loads acting on the system [BK98, BKBLV07].

In order to simply the formulations, nine inertia invariants can be calculated using the mode shapes, the nodal masses and the position of the center of gravity with respect to each node. The inertia invariants are given in Table 3.1. The table gives the invariant name, how it is obtained, a description if possible, and the size of the element. For example, $\mathcal{I}2$ is a 3×1 vector representing the first moment of inertia of the rigid part of the platform. On the other hand, $\mathcal{I}3$ is a tensor of N components, where the k^{th} denoted $\mathcal{I}3_k$, component is a 3×1 vector and $\mathcal{I}9_{jk}$ is a $N \times N$ tensor where each component is a 3×3 matrix.

Table 3.1: Inertia invariants for flexible body, where $j, k = 1 \ldots N$

Variable	Calculation	Description	Size
$\mathcal{I}1$	$\int_{\Sigma_{P0}} dm$	Mass of body	scalar
$\mathcal{I}2$	$\int_{\Sigma_{P0}} \mathbf{r}_p dm$	1^{st} moments of inertia	3×1
$\mathcal{I}3_k$	$\int_{\Sigma_{P0}} \mathbf{\Phi}_{dk} dm$	Elastic 1^{st} moments of inertia	3×1
$\mathcal{I}4_k$	$\int_{\Sigma_{P0}} \hat{\mathbf{r}}_p \mathbf{\Phi}_{dk} dm$		3×1
$\mathcal{I}5_{jk}$	$\int_{\Sigma_{P0}} (\mathbf{\Phi}_{dk} \mathbf{\Phi}_{dj})\, dm$		3×1
$\mathcal{I}6$	$\int_{\Sigma_{P0}} \left(\mathbf{\Phi}_d^T \mathbf{\Phi}_d\right) dm$	Elastic inertia matrix	$N \times N$
$\mathcal{I}7$	$\int_{\Sigma_{P0}} \hat{\mathbf{r}}_p^T \hat{\mathbf{r}}_p dm$	Rigid inertia matrix	3×3
$\mathcal{I}8_k$	$\int_{\Sigma_{P0}} \hat{\mathbf{r}}_p \hat{\mathbf{\Phi}}_{dk} dm$	Rigid Elastic inertia tensor	3×3
$\mathcal{I}9_{jk}$	$\int_{\Sigma_{P0}} \hat{\mathbf{\Phi}}_{dj} \hat{\mathbf{\Phi}}_{dk} dm$	Elastic Elastic inertia tensor	3×3

Σ_{P0} is defined as the initial, undeformed state of the flexible platform. The dynamic equation of the flexible platform is given as:

$$\begin{bmatrix} \mathbf{f}_p \\ \mathbf{n}_p \\ \mathbf{Q}_p \end{bmatrix} = \begin{bmatrix} m\mathbf{1}_3 & \mathbf{M}\hat{\mathbf{S}}_r^T & \mathbf{MS}_{de} \\ \mathbf{M}\hat{\mathbf{S}}_r & \mathbf{I}_{0p} & \mathbf{MS}_{re} \\ \mathbf{MS}_{de}^T & \mathbf{MS}_{re}^T & \mathbf{m}_{ee} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{v}}_p - \mathbf{g} \\ \dot{\boldsymbol{\omega}}_p \\ \ddot{\mathbf{q}}_e \end{bmatrix} + \begin{bmatrix} \mathbf{f}_c \\ \mathbf{n}_c \\ \mathbf{Q}_c \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \mathbf{K}_{ee}\mathbf{q}_e \end{bmatrix} + \begin{bmatrix} 0 \\ 0 \\ \mathbf{D}_{ee}\dot{\mathbf{q}}_e \end{bmatrix} \tag{3.32}$$

The components of the mass matrix of the body are obtained from the inertia invariants in the following. The mass and the first moments of inertia are defined as:

$$m = \mathcal{I}1 \qquad \mathbf{M}\hat{\mathbf{S}}_r^T = \mathcal{I}2 + \sum_{k=1}^{N} \hat{\mathcal{I}}3_k q_{ek} \tag{3.33}$$

The total inertia tensor of the body is defined as:

$$\mathbf{I}_{0p} = \mathcal{I}2 + \sum_{k=1}^{N} \left(\mathcal{I}8_k + \mathcal{I}8_k^T\right) q_{ek} + \sum_{k=1}^{N}\sum_{j=1}^{N} \mathcal{I}9_{jk} q_{ej} q_{ek} \tag{3.34}$$

The row matrix of vectors $\mathbf{MS}_{de}$ is defined as:

$$\mathbf{MS}_{de} = \left[\ \mathcal{I}3_1 \quad \mathcal{I}3_2 \quad \ldots \quad \mathcal{I}3_N\ \right] \tag{3.35}$$

The row matrix of vectors $\mathbf{MS}_{re}$ is defined as:

$$\mathbf{MS}_{re} = \left[\mathcal{I}4_1 + \sum_{k=1}^{N} \mathcal{I}5_{k1} q_{ek}, \quad \mathcal{I}4_2 + \sum_{k=1}^{N} \mathcal{I}5_{k2} q_{ek}, \quad \ldots \quad , \mathcal{I}4_N + \sum_{k=1}^{N} \mathcal{I}5_{kN} q_{eN}\right] \tag{3.36}$$

The vector of forces due to centrifugal and Coriolis effects is given as:

$$\mathbf{f}_c = 2\boldsymbol{\omega}_p \times \sum_{k=1}^{N} \mathcal{I}3_k \dot{q}_{ek} + \boldsymbol{\omega}_p \times \left(\boldsymbol{\omega}_p \times \mathbf{M}\hat{\mathbf{S}}_r\right) \tag{3.37}$$

The vector of moments due to centrifugal and Coriolis effects is given as:

$$\mathbf{n}_c = \boldsymbol{\omega}_p \times (\mathbf{I}_{0p}\boldsymbol{\omega}_p) + 2\left(\sum_{k=1}^{N} \mathcal{I}8_k \boldsymbol{\omega}_p \dot{q}_{ek}\right) + 2\left(\sum_{j=1}^{N}\sum_{k=1}^{N} \mathcal{I}9_{jk} \dot{q}_{ej} q_{ek}\right) \tag{3.38}$$

The vector of generalized elastic forces due to centrifugal and Coriolis effects is given as:

$$\mathbf{Q}_c = \begin{bmatrix} \boldsymbol{\omega}_p^T \mathcal{I}8_1^T \boldsymbol{\omega}_p^T - \boldsymbol{\omega}_p^T \sum\limits_{k=1}^{N} \mathcal{I}9_1^T \boldsymbol{\omega}_p q_{ek} + 2\boldsymbol{\omega}_p^T \sum\limits_{k=1}^{N} \mathcal{I}5_{1k} \dot{q}_{ek} \\ \boldsymbol{\omega}_p^T \mathcal{I}8_2^T \boldsymbol{\omega}_p^T - \boldsymbol{\omega}_p^T \sum\limits_{k=1}^{N} \mathcal{I}9_2^T \boldsymbol{\omega}_p q_{ek} + 2\boldsymbol{\omega}_p^T \sum\limits_{k=1}^{N} \mathcal{I}5_{2k} \dot{q}_{ek} \\ \vdots \\ \boldsymbol{\omega}_p^T \mathcal{I}8_N^T \boldsymbol{\omega}_p^T - \boldsymbol{\omega}_p^T \sum\limits_{k=1}^{N} \mathcal{I}9_N^T \boldsymbol{\omega}_p q_{ek} + 2\boldsymbol{\omega}_p^T \sum\limits_{k=1}^{N} \mathcal{I}5_{Nk} \dot{q}_{ek} \end{bmatrix} \tag{3.39}$$

Finally, $\mathbf{K}_{ee} = \mathrm{diag}_{i,j=1\ldots N} k_{ij}$ is the matrix of generalized stiffness. The matrix of generalized damping is defined as $\mathbf{D}_{ee} = \mathrm{diag}_{i,j=1\ldots N} 2d_{ij}\sqrt{k}_{ij}$, while $\mathbf{g}$ is the gravity vector. It should be noted that the above variables are all calculated in the origin of the deformable body. Therefore for a multi-body dynamic system, the origin of the deformable body is generally chosen to coincide with the joint frame.

Recursive Solution to Generalized Newton-Euler Approach

When the dynamic expression (3.32) is written for all bodies in the system, a recursive algorithm can be established to calculate the required joint torques in a multibody system that aims to track the rigid variables while stabilizing the elastic variables [BK98]. The solution consists of three recursive calculations as follows:

1. From $i = 1 \dots n$ calculate the angular velocity of each link, inertial terms and link transformations. This calculation is initialized by $\omega_0 = 0$.
2. From $i = n \dots 1$ calculate the elastic accelerations and link forces as functions of the rigid body acceleration of the link.
3. From $i = 1 \dots n$ calculate the value of the elastic accelerations and link forces and the joint torques. This calculation is initialized by $\dot{\mathbf{V}}_0 = 0$.

This algorithm efficiently calculates the dynamic model of a serial manipulator that comprises many flexible links. It has also been extended to parallel manipulators in [BK13].

3.2.6 Robotic Control of Deformable Objects

In this section, the primary works concerning cooperative manipulation of flexible objects are outlined. This scenario is shown in Fig.3.5. The input variables to the system are typically the acceleration of the end effectors $\ddot{\mathbf{x}}_1$ and $\ddot{\mathbf{x}}_2$ which are controlled by applying joint torques. The task frame variables $\mathbf{x}_{obj}$ can either be reconstructed from its effects on the end effectors or using an external sensor such as vision.

Task Definition

The control objectives of a robot with flexible components are similar to standard robotic control objectives i.e trajectory following or regulation in the Cartesian or joint space.

For robot's handling the deformable objects, the same tasks apply, however in addition the shape of the object may be changed:

– Moving the end effectors to modify an object's configuration

This task is referred to as shape-control [DS11], and it is achieved by deforming the object such that its form converges to a desired shape. For example in Fig.3.5, the task is defined as changing the object shape from its initial configuration defined as $\mathcal{O}_s$ to a desired configuration denoted as $\mathcal{O}_s^*$.

The aforementioned tasks are difficult due to the vibrations of the flexible components and also since the variables describing the deformation of the object are generally either unknown or estimated. A damping controller is usually applied to ensure the magnitude of the vibrations are below a critical threshold, however, in certain configurations the vibrations cannot be controlled by the robot [ZZG00].

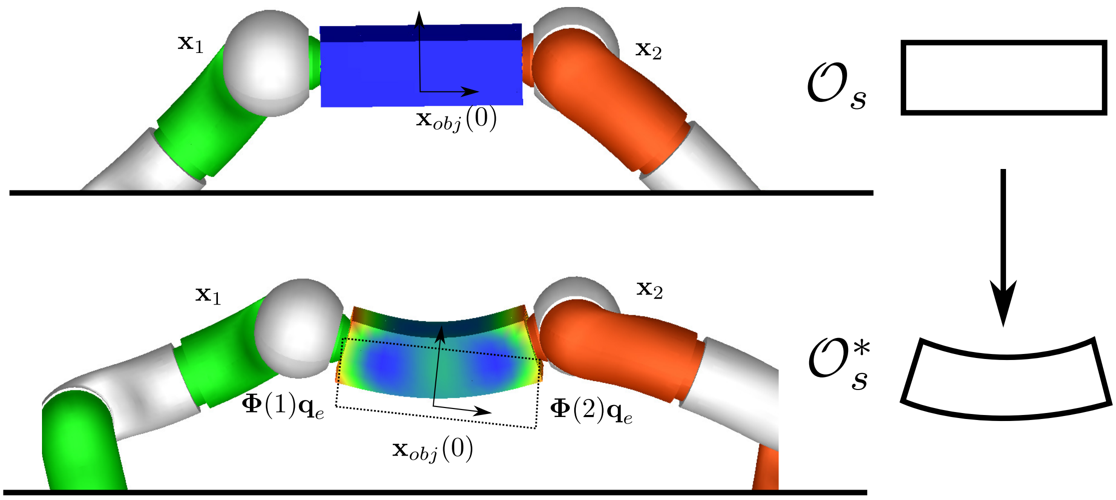

Figure 3.5: Cooperative system manipulating a flexible object, with free free boundary conditions executing shape control of object from initial state $\mathcal{O}_s$ to desired state $\mathcal{O}_s^*$

Trajectory Following

For the trajectory following problem, the research can be divided into two subgroups, those that make use of advanced object models to predict and then damp vibrations and those that simply control the rigid coordinates.

An impedance controller is proposed in [SL97], where due to the limited flexibility, an internal force can be applied to the object. Therefore the impedance system must control three different types of forces. Type one that neither causes rigid motion nor vibration but creates an internal force, type two that does not cause rigid motion but may deform the body and type three, a motion causing force.

The combination of position and force is also explored in [SL01] where a non-linear continuous controller is proposed for a cooperative system handling a beam. The forces experienced by the beam are also controlled using a hybrid position force controller. The objective of the controller is track a trajectory and ensure the vibrations of the beam remain bounded.

In [TEJ09], a time scale control called single value perturbation is used. The vibration of the beam is controlled by the fast control law and the slower control law controls the rigid body coordinates. Sliding mode controllers have been used in [AYAH07] to ensure the beam like object follows a trajectory with an acceptable level of vibration.

In all of the above works, the dynamic model is derived for the common object that is modeled using a beam formulation. The dynamics of the beam are written as:

$$\mathbf{A}_x \begin{bmatrix} \ddot{\mathbf{x}}_0 \\ \ddot{\mathbf{q}}_e \end{bmatrix} + \begin{bmatrix} \mathbf{c}_r \\ \mathbf{c}_e \end{bmatrix} = \mathbf{W} \begin{bmatrix} \mathbf{h}_1 \\ \mathbf{h}_2 \end{bmatrix} \tag{3.40}$$

The dynamics of the manipulator can be introduced by replacing the Cartesian wrench, $\mathbf{h}$ with expression given in (2.15), to obtain a expression of the system in terms of Cartesian variables and joint torques. The matrix $\mathbf{W}$ relates the Cartesian forces applied by the end effector to the forces at the object frame and is therefore the grasp matrix as given in (2.3). However, in this case the structure of the grasp matrix is decided by the boundary conditions of the flexible object. An example of the structure of $\mathbf{W}$ for clamped-free (cf), clamped clamped (cc) and free-free (ff) is as follows:

$$\mathbf{W}_{cf} = \begin{bmatrix} \mathbf{W}_1 & \mathbf{W}_{12} \\ \mathbf{0} & \mathbf{W}_{22} \end{bmatrix}, \qquad \mathbf{W}_{cc} = \begin{bmatrix} \mathbf{W}_1 & \mathbf{W}_{12} \\ \mathbf{0} & \mathbf{0} \end{bmatrix}, \qquad \mathbf{W}_{ff} = \begin{bmatrix} \mathbf{W}_1 & \mathbf{W}_{12} \\ \mathbf{W}_{21} & \mathbf{W}_{22} \end{bmatrix} \tag{3.41}$$

As in the rigid case, if the null space to this matrix exists, it means that object can undergo internal loading to due manipulator forces as shown in [SL97].

On the other hand, in [MI96], the stability of a rigid control laws (PID, Computed torque, Hybrid Position Force) for manipulators handling a non-rigid object in the automotive assembly industry is demonstrated.

Finally, an interesting a wave based technique is proposed in [O'C07]. This approach is very useful for the first and second type of task outlined in Section 3.2.6. The paper proposes to split the trajectory into two phases, The first phase moves the body and uses the wave feedback to damp the effects while storing (or recording) this feedback as the wave reaction. The second part of the trajectory uses the *recorded effects* as a feed forward canceling the wave reaction and arriving at the desired point without vibration. The body must behave like a wave in order for this non-model based technique to work.

Shape Control

Shape control i.e applying force to the object in order to modify its state as shown in Fig.3.5, is a difficult task requiring either high sensor capacity, advanced object models or on-line learning schemes. However there are many potential applications, for instance in assembly operations or the food industry.

Objects of limited DOF In [DDDBV93] a hybrid position force scheme, an extension of the rigid case, is used to manipulate a deformable object of a known structure specifically a 1-DOF spring. A 1-DOF object is also used in [aH00]. In this case the forces are divided into those causing motion, those causing internal forces and finally those that compress the springs. The combination of force and position is achieved using an impedance controller.

In [SU94], a flexible object is controlled using a hybrid position force control law. The force and the moments applied to the object by the manipulators as well as the end-point accelerations are considered to be the control inputs for the Cartesian level control. Vibration suppression is introduced to regulate the damping properties of the elastic system. The object elastic's model is known.

In [WR94], the classical object impedance control for cooperative manipulators is extended to a class of deformable objects. The control law is constructed for objects that have less than 6-DOF. If this criteria is met, then a lumped parameter model of the object is created where the lumped parameters are controlled in the impedance law. The impedance control law, given in Chapter 2, Section 2.2 can be directly applied in this case simply by changing the task variables.

In fact in all of the above works, due to the limited object DOF, the extended task vector $\mathbf{x} = \begin{bmatrix} x & y & \theta & s \end{bmatrix}^T$ is used, where s is the flexible variable that is directly controlled. Once again this behavior can be characterized by the form of the grasp matrix $\mathbf{W}$.

On the other hand in [KJM97] a hybrid position force scheme is developed in order to bend beams until the beams converge to a desired state. The task is decomposed into two types of bending, which gives references variables to track in position and force for the robots.

Objects of unknown DOF The above works focus on objects whose DOF are limited allowing the manipulators to fully control the object state. In general, this is not the case, and an alternative view of the deformable body must be taken.

In [WHKK01] a coarse planar object model is meshed then built using simple spring models. The position of the mesh points must be known. The mesh on the deformable object is defined into three different point types, *manipulation* points where the robots apply forces, *positioned* points that must be positioned, and the rest *non-target* points. The objective is to apply forces to the *manipulation* point so that positioning points converge to a desired configuration.

A similar work, which overcomes the requirement of having a estimation of the object model, is given in [NAhLRL13]. Visual servoing, as described in Chapter 4, Section 4.2 is used to track the object state during deformation. The objective is to deform the object such that the extracted features reach a desired position. The object is defined as an elastic model and several visual markers, defined by the vector $\mathbf{s}$, are located on its surface. A deformation Jacobian matrix links the end effector actions to the deformation of the points. The Jacobian matrix is unknown a priori and is a function of the material properties of the object, therefore it must be learned on-line. The authors propose the Broyden updating rule commonly used in visual servoing to learn an interaction matrix. The updating rule is defined as:

$$\hat{\mathbf{J}}(t) = \hat{\mathbf{J}}(t-1) + \frac{\delta\mathbf{s}(t) - \hat{\mathbf{J}}(t-1)\delta\mathbf{x}(t)}{\delta\mathbf{x}^T(t)\delta\mathbf{x}(t)}\delta\mathbf{x}^T(t) \tag{3.42}$$

with

$$\mathbf{J}(t) = \frac{\partial\mathbf{s}}{\partial\mathbf{x}} \tag{3.43}$$

The estimate of the Jacobian matrix is then used in a passivity based controller where, allowing for certain assumptions concerning the accuracy of the vision system, the stability is proved.

In [DS11], the shape of a mass spring damper type planar object is controlled by multiple manipulators. The shape of the object is described by a closed curve parametrically described, for example a circle would be described by $\mathbf{c} = \begin{bmatrix} r\cos(t) & r\sin(t) \end{bmatrix}$. A Jacobian is constructed that relates the joint variables to the curve parameters. The control scheme aims to change the shape of the object such that it converges to a desired curve. The authors use the simulation of the planar case to demonstrate the increased efficiency of the scheme when extra manipulators are employed. The difficulty in such an approach is the extension to higher dimensions.

3.3 Dynamic Modeling of Cooperative Manipulators with flexible objects

This section outlines the dynamic model of closed chain robots whose terminal link is flexible. This includes parallel robots with flexible platforms and cooperative manipulators handling flexible objects.

In order to proceed, the closed chain system is decomposed into two sub-systems, one is flexible and the other is rigid. The rigid sub-system comprises the robotic system handling the object i.e. the serial robots. The flexible sub-system consists of the grasped object. The object is described using distributed flexibility [BK98, RCM10] and modeled using Cartesian coordinates and the Newton-Euler formulation.

The decomposition is performed by opening (virtually) the connection points between the rigid system and the flexible system. The connection points are located at the point where the end effector grasp the object. The two sub-systems are connected by calculating the reaction wrench at the connection points between the object and the arms. By connecting the two subsystems, the system's dynamic model can be obtained. The inverse dynamic model obtains the joint torques and forces for a desired acceleration of the flexible system using the state variables of the robot (the positions and velocities). The direct dynamic model gives the elastic and rigid accelerations of the system's variables in terms of the input torques and the state of the system.

3.3.1 System Description

The procedure is outlined for a system of n non-redundant serial arms, composed of rigid links, grasping a common object. The object is defined by N flexible coordinates (the dimension of the modal representation of the flexibility). The i^{th} arm for $i = 1 \dots n$,

- contains a_i actuated joints and m_i movable links. The remaining $m_i - a_i$ joints are considered as passive
- transmits a wrench of dimension c_i to the object ($1 \leq c_i \leq 6$). For instance, if the end effector grasp constitutes a spherical joint, only linear forces can be transmitted from the serial robot to the end effector, thus $c_i = 3$. Furthermore if the i^{th} robot is operating in a subspace, for example a planar system, only a planar wrench can be trasmitted by the robot to the object, thus $c_i = 3$.

The following terms are defined, $a = \sum_{i=1}^{n} a_i$, $m = \sum_{i=1}^{n} m_i$ and $c = \sum_{i=1}^{n} c_i$.

3.3.2 Rigid Sub-System Modeling

The $(m_i \times 1)$ vectors of joint positions, velocities, accelerations and torques of arm i are denoted as $\mathbf{q}_i$, $\dot{\mathbf{q}}_i$, $\ddot{\mathbf{q}}_i$ and $\mathbf{\Gamma}_i$ respectively. The $(c_i \times 1)$ vectors of velocity and acceleration at grasping point i are defined as:

$$\mathbf{V}_i = \mathbf{J}_i \dot{\mathbf{q}}_i \qquad \dot{\mathbf{V}}_i = \mathbf{J}_i \ddot{\mathbf{q}}_i + \dot{\mathbf{J}}_i \dot{\mathbf{q}}_i \tag{3.44}$$

where $\mathbf{J}_i$ $(c_i \times m_i)$ is the kinematic Jacobian of arm i. Furthermore, the following quantities must be calculated: the inertia matrix $\mathbf{A}_i$ $(m_i \times m_i)$ and the $(m_i \times 1)$ vector of Coriolis, centrifugal and gravity torques $\mathbf{c}_i$.

$\mathbf{A}_{xi}$, a $(c_i \times c_i)$ matrix, and $\mathbf{c}_{xi}$, a $(c_i \times 1)$ vector, can be respectively seen as the robots inertia matrix and the vector of Coriolis, centrifugal and gravity torques transformed to the Cartesian space at the grasp point i [Kha87]:

$$\mathbf{A}_{xi} = \mathbf{J}_i^{-T} \mathbf{A}_i \mathbf{J}_i^{-1} \qquad \mathbf{c}_{xi} = \mathbf{J}_i^{-T} \mathbf{c}_i - \mathbf{A}_{xi} \dot{\mathbf{J}}_i \dot{\mathbf{q}}_i \tag{3.45}$$

The kinematic Jacobian matrix relates the vector of joint torques of robot i to the wrench transmitted by grasp point i, denoted as $\mathbf{h}_i$:

$$\mathbf{h}_i = \mathbf{J}_i^{-T} \left(\mathbf{\Gamma}_i - \mathbf{A}_i \ddot{\mathbf{q}}_i - \mathbf{c}_i \right) \tag{3.46}$$

If the serial arm contains passive joints, which have always zero torque, then the columns of $\mathbf{J}_i^{-T}$ that correspond to the actuated joint must be extracted resulting in a matrix $\mathbf{J}_{ai}^{-T}$ with dimension $(c_i \times a_i)$. Thus using (3.44), and (3.45), (3.46) becomes:

$$\mathbf{h}_i = \mathbf{J}_{ai}^{-T} \boldsymbol{\tau}_i - \mathbf{A}_{xi} \dot{\mathbf{V}}_i - \mathbf{c}_{xi} \tag{3.47}$$

where $\boldsymbol{\tau}_i$ is a vector of a_i components containing the actuated joint torques. Equation (3.47) gives a relationship between joint torques and the grasp wrench for the i^{th} manipulator. Finally, (3.47) can be rewritten for all arms grasping the object as follows:

$$\begin{bmatrix} \mathbf{h}_1 \\ \vdots \\ \mathbf{h}_n \end{bmatrix} = \begin{bmatrix} \mathbf{J}_{a1}^{-T} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \mathbf{J}_{an}^{-T} \end{bmatrix} \begin{bmatrix} \boldsymbol{\tau}_1 \\ \vdots \\ \boldsymbol{\tau}_n \end{bmatrix} - \mathbf{A}_x \begin{bmatrix} \dot{\mathbf{V}}_1 \\ \vdots \\ \dot{\mathbf{V}}_n \end{bmatrix} - \mathbf{c}_x \tag{3.48}$$

where $\mathbf{A}_x$ is a $(c \times n) \times (c \times n)$ block diagonal matrix where the i^{th} block is $\mathbf{A}_{xi}$ and $\mathbf{c}_x$ is an $c \times 1$ vector such that $\mathbf{c}_x = \begin{bmatrix} \mathbf{c}_{x1}^T & \dots & \mathbf{c}_{x6}^T \end{bmatrix}^T$. $\boldsymbol{\tau}$ is an $a \times 1$ vector composed of all the actuated joint forces/torques of the system such that $\boldsymbol{\tau} = \begin{bmatrix} \boldsymbol{\tau}_1^T & \dots & \boldsymbol{\tau}_n^T \end{bmatrix}^T$

3.3.3 Flexible Sub-System Modeling

Object Description

The flexibility is represented by a series of shape functions and is modeled using the generalized Newton-Euler model as described in Section 3.2.5. It is recalled that the main hypothesis in this formalism is that the object undergoes small deformations that can be described using modal shape functions. Therefore the motion of the flexible object can be approximated by the sum of the rigid body motion and the flexible body deformation.

The frame $\boldsymbol{\Sigma}_p$ is fixed with the object, and its origin is located at an operational point, for example the geometric center. The location of the $\boldsymbol{\Sigma}_p$ in the world frame is defined by ${}^0\mathbf{T}_p$, the 4×4 homogeneous transformation matrix. The object contains N flexible DOF, i.e. the total number of shape functions characterizing the flexible behavior.

The most common boundary conditions for representing the flexibility are clamped-free, free-free, pinned-pinned and clamped-clamped. The boundary conditions not only define the deformation characteristics of the object but also the reaction force transmitted by the end effector. In this case the object has n boundary conditions, located at the grasp points. We model the object using free boundary conditions, where the shape functions can be obtained using finite element software, for instance MSC Nastran ©.

In the following the Generalized Newton-Euler model is applied to the cooperative system, to obtain geometric, kinematic and dynamic relations which take into account the flexibility of the object.

Object Kinematics

The position of the i^{th} end effector, denoted as $\mathbf{p}_i$ for $i = 1 \ldots n$, can be calculated from the position of the object and the position vector from the origin to this connection point as:

$$\mathbf{p}_i = \mathbf{p}_p + \mathbf{r}_i \tag{3.49}$$

where $\mathbf{r}_i$ is defined as the position vector from the origin of the object frame, denoted as $\mathbf{p}_p$, to $\mathbf{p}_i$. The vector $\mathbf{r}_i$ is a function of the flexible parameters of the object. It is obtained by the summation of the rigid body position, $\mathbf{r}_i(0)$, and the deformation due to flexibility using (3.50):

$$\mathbf{r}_i = \mathbf{r}_i(0) + \sum_{k=1}^{N} \boldsymbol{\Phi}_{dk}(i) q_{ek} \tag{3.50}$$

$\mathbf{q}_e = (q_{e1} \ldots q_{ek} \ldots q_{eN})$, is the $N \times 1$ vector of generalized elastic coordinates. The derivative of (3.50), leads to

$$\dot{\mathbf{r}}_i = \boldsymbol{\omega}_p \times \mathbf{r}_i + \sum_{k=1}^{N} \boldsymbol{\Phi}_{dk}(i) \dot{q}_{ek} \tag{3.51}$$

$\boldsymbol{\Phi}_{dk}(i)$ and $\boldsymbol{\Phi}_{rk}(i)$ are the k^{th} displacement and rotation shape functions at point i. Since the shape functions are defined with respect to the object frame $\boldsymbol{\Sigma}_p$, all other variables in the following are represented in this frame unless otherwise stated. $\boldsymbol{\omega}_p$ is defined as the vector of angular velocity of the moving platform. $\dot{\mathbf{q}}_e$ and $\ddot{\mathbf{q}}_e$ are the vectors of velocity and acceleration of the generalized elastic coordinates.

The velocity screw at the object origin is defined as $\mathbf{V}_t$ which is composed of $\mathbf{v}_t$ and $\boldsymbol{\omega}_t$ the total (including the effects of flexibility) linear and angular velocity of the object respectively. The total velocity screw can be obtained as the sum of the rigid body velocity screw evaluated at that point and the velocity due to the effects of the flexibility.

$$\mathbf{V}_t = \mathbf{V}_p + \begin{bmatrix} \boldsymbol{\Phi}_d(p) \\ \boldsymbol{\Phi}_r(p) \end{bmatrix} \dot{\mathbf{q}}_e \tag{3.52}$$

with

$$\boldsymbol{\Phi}_d(p) = \begin{bmatrix} \Phi_{d1}(p) & \Phi_{d2}(p) & \dots & \Phi_{dN}(p) \end{bmatrix} \tag{3.53}$$

$$\boldsymbol{\Phi}_r(p) = \begin{bmatrix} \Phi_{r1}(p) & \Phi_{r2}(p) & \dots & \Phi_{rN}(p) \end{bmatrix} \tag{3.54}$$

$$\mathbf{V}_p = \begin{bmatrix} \mathbf{v}_p \\ \boldsymbol{\omega}_p \end{bmatrix} \qquad \mathbf{V}_t = \begin{bmatrix} \mathbf{v}_t \\ \boldsymbol{\omega}_t \end{bmatrix} \tag{3.55}$$

$\mathbf{v}_p$ is defined as the component of rigid velocity of the object, while $\dot{\mathbf{v}}_p$ and $\dot{\boldsymbol{\omega}}_p$ denote the linear and angular acceleration. The kinematic twist of the flexible body at end effector i is given as:

$$\begin{bmatrix} \mathbf{v}_i \\ \boldsymbol{\omega}_i \end{bmatrix} = \begin{bmatrix} \mathbf{I}_3 & -\hat{\mathbf{r}}_i \\ \mathbf{0}_3 & \mathbf{I}_3 \end{bmatrix} \begin{bmatrix} \mathbf{v}_p \\ \boldsymbol{\omega}_p \end{bmatrix} + \begin{bmatrix} \boldsymbol{\Phi}_d(i) \\ \boldsymbol{\Phi}_r(i) \end{bmatrix} \dot{\mathbf{q}}_e \tag{3.56}$$

where $\hat{\mathbf{x}}$ designates the 3×3 skew symmetric matrix associated with a vector $\mathbf{x}$, such that $\hat{\mathbf{x}}\mathbf{y} = \mathbf{x} \times \mathbf{y}$. Equation (3.56) becomes:

$$\mathbf{V}_i = {}^i\mathbf{S}_p\ \mathbf{V}_p \ + \ \boldsymbol{\Phi}(i)\ \dot{\mathbf{q}}_e \tag{3.57}$$

$$\boldsymbol{\Phi}(i) = \begin{bmatrix} \boldsymbol{\Phi}_d^T(i) & \boldsymbol{\Phi}_r^T(i) \end{bmatrix}^T \tag{3.58}$$

${}^i\mathbf{S}_p$ is the (6×6) kinematic transformation matrix from frame p to frame i. $\mathbf{V}_i$ denotes the kinematic screw evaluated at point i. The forces and moments, denoted as $\mathbf{f}_i$ and $\mathbf{n}_i$, applied to point i are transformed to the flexible object's origin by:

$$\begin{bmatrix} \mathbf{h}_p \\ \mathbf{Q}_p \end{bmatrix} = \begin{bmatrix} {}^i\mathbf{S}_p^T \\ \boldsymbol{\Phi}^T(i) \end{bmatrix} \begin{bmatrix} \mathbf{f}_i \\ \mathbf{n}_i \end{bmatrix} \tag{3.59}$$

where $\mathbf{Q}_p$ denotes the generalized elastic forces. $\mathbf{h}_p$ denotes the wrench at the object frame that contains the forces, $\mathbf{f}_p$, and the moments, $\mathbf{n}_p$.

However, depending on the grasping condition and the structure of the serial robot, the i^{th} end effector may not be able to transmit a 6×1 wrench to the object. Therefore the matrix $\mathbf{L}_i$ is used to transform the spatial object variables into the constraint space of the i^{th} end effector. $\mathbf{L}_i$ is a $(6 \times c_i)$ matrix where each column represents a constraint direction. For example, $\mathbf{L}_i^p$ and $\mathbf{L}_i^{ty}$ that represent a planar and 1-DOF prismatic grasp condition respectively, are given as:

$$\mathbf{L}_i^p = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{L}_i^{ty} = \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{3.60}$$

By using $\mathbf{L}_i$, for $i = 1 \dots n$, a block diagonal matrix selection matrix $\mathbf{L}$, of dimension $(6n \times c)$, is obtained that projects the object variables into the connection point space for all end effectors.

Therefore, the object variables are related to the variables at the grasping points by using $\mathbf{L}$ and a grasp matrix $\mathbf{W}$. The dimension of $\mathbf{W}$, $(6 + N \times 6n)$, depends on the number of manipulators and the flexible coordinates. $\mathbf{W}$ transforms forces applied by the cooperative arms to the object origin likewise its transpose relates the velocities of the connection points to the object velocity.

Rewriting (3.56) for all manipulators:

$$\begin{bmatrix} \mathbf{V}_1 \\ \vdots \\ \mathbf{V}_n \end{bmatrix} = \mathbf{L}^T \mathbf{W}^T \begin{bmatrix} \mathbf{V}_p \\ \dot{\mathbf{q}}_e \end{bmatrix} = \mathbf{L}^T \begin{bmatrix} \mathbf{W}_p^T & \mathbf{W}_e^T \end{bmatrix} \begin{bmatrix} \mathbf{V}_p \\ \dot{\mathbf{q}}_e \end{bmatrix} \tag{3.61}$$

Similarly, from (3.59) and (3.61) it can be seen that:

$$\begin{bmatrix} \mathbf{h}_p \\ \mathbf{Q}_p \end{bmatrix} = \mathbf{W}\mathbf{L} \begin{bmatrix} \mathbf{h}_1 \\ \vdots \\ \mathbf{h}_n \end{bmatrix} = \begin{bmatrix} \mathbf{W}_p \\ \mathbf{W}_e \end{bmatrix} \mathbf{L} \begin{bmatrix} \mathbf{h}_1 \\ \vdots \\ \mathbf{h}_n \end{bmatrix} \tag{3.62}$$

$\mathbf{W}$ is decomposed into $\mathbf{W}_p$, a $(6 \times 6n)$ matrix, and $\mathbf{W}_e$ a $(N \times 6n)$, matrix as follows:

$$\mathbf{W} = \left[\begin{array}{c|c|c} {}^1\mathbf{S}_p^T & \dots & {}^n\mathbf{S}_p^T \\ \mathbf{\Phi}^T(1) & \dots & \mathbf{\Phi}^T(n) \end{array} \right] = \left[\begin{array}{c|c|c} \mathbf{W}_{p1} & \dots & \mathbf{W}_{pn} \\ \hline \mathbf{W}_{e1} & \dots & \mathbf{W}_{en} \end{array} \right] \tag{3.63}$$

By differentiation of (3.61), a similar relationship can be obtained for the acceleration.

$$\begin{bmatrix} \dot{\mathbf{V}}_1 \\ \vdots \\ \dot{\mathbf{V}}_n \end{bmatrix} = \mathbf{L}^T \left(\mathbf{W}^T \begin{bmatrix} \dot{\mathbf{V}}_p \\ \ddot{\mathbf{q}}_e \end{bmatrix} + \mathbf{b} \right) \tag{3.64}$$

where $\mathbf{b} = \begin{bmatrix} \mathbf{b}_1^T & \dots & \mathbf{b}_N^T \end{bmatrix}^T$, and for $i = 1 \dots n$:

$$\mathbf{b}_i = \begin{bmatrix} \boldsymbol{\omega}_p \times \boldsymbol{\omega}_p \times \mathbf{r}_i + 2\boldsymbol{\omega}_p \times \boldsymbol{\Phi}_d(i)\dot{\mathbf{q}}_e \\ \boldsymbol{\omega}_p \times \boldsymbol{\Phi}_r(i)\dot{\mathbf{q}}_e \end{bmatrix} \tag{3.65}$$

Object Dynamics

The dynamic equation of a flexible body given by (3.32) in Section 3.2, is rewritten for convenience as:

$$\begin{bmatrix} \mathbf{h}_p \\ \mathbf{Q}_p \end{bmatrix} = \begin{bmatrix} \mathbf{A}_{rr} & \mathbf{A}_{re} \\ \mathbf{A}_{re}^T & \mathbf{m}_{ee} \end{bmatrix} \begin{bmatrix} \dot{\mathbf{V}}_p \\ \ddot{\mathbf{q}}_e \end{bmatrix} + \begin{bmatrix} \mathbf{c}_r \\ \mathbf{c}_e \end{bmatrix} \tag{3.66}$$

where:

$$\mathbf{A}_{rr} = \begin{bmatrix} m\mathbf{1}_3 & \mathbf{M}\hat{\mathbf{S}}_r^T \\ \mathbf{M}\hat{\mathbf{S}}_r & \mathbf{I}_{0p} \end{bmatrix} \qquad \mathbf{A}_{re} = \begin{bmatrix} \mathbf{MS}_{de} \; \mathbf{MS}_{re} \end{bmatrix} \tag{3.67}$$

$$\mathbf{c}_r = \begin{bmatrix} \mathbf{f}_c \\ \mathbf{n}_c \end{bmatrix} + \begin{bmatrix} \mathbf{f}_g \\ \mathbf{n}_g \end{bmatrix} \qquad \mathbf{c}_e = \mathbf{Q}_c + \mathbf{Q}_g + \mathbf{K}_{ee}\mathbf{q}_e + \mathbf{D}_{ee}\dot{\mathbf{q}}_e - \boldsymbol{\lambda}_f \tag{3.68}$$

The term $\boldsymbol{\lambda}_f$ must be introduced for the general algorithm. This represents a set of elastic forces generated due to possibility of a robot operating in a different kinematic subspace to the object. For example, if a planar manipulator is grasping a spatial object, any *out-of-plane* vibrations in the object will have no effect on the robot's joint variables. However, these vibrations will induce the generalized elastic force, $\boldsymbol{\lambda}_f$, at the object origin. To obtain the value of $\boldsymbol{\lambda}_f$, the *out-of-plane* constraint forces are transformed to the object origin using (3.62).

The terms $\mathbf{f}_g$, $\mathbf{n}_g$ and $\mathbf{Q}_g$ contain the effects of gravity on the platform, obtained from (3.32):

$$\begin{bmatrix} \mathbf{f}_g \\ \mathbf{n}_g \\ \mathbf{Q}_g \end{bmatrix} = \begin{bmatrix} m\mathbf{1}_3 & \mathbf{M}\hat{\mathbf{S}}_r^T & \mathbf{MS}_{de} \\ \mathbf{M}\hat{\mathbf{S}}_r & \mathbf{I}_{0p} & \mathbf{MS}_{re} \\ \mathbf{MS}_{de}^T & \mathbf{MS}_{re}^T & \mathbf{m}_{ee} \end{bmatrix} \begin{bmatrix} -\mathbf{g} \\ \mathbf{0}_3 \\ \mathbf{0}_N \end{bmatrix} \tag{3.69}$$

For the numerical simulation given in Section 3.3.5, to calculate the above variables, the object is discretized in a series of elements joined together at nodes. Therefore the calculation of the inertial invariants is no longer achieved by integrating across a continuous medium, instead the discrete sum is taken. The modal analysis gives the mass of each node i.e. dm_m for $m = 1 \dots M$, the generalized stiffness for each mode, the shape functions of each mode evaluated at every node i.e. $\boldsymbol{\Phi}_k(m)$ for $m = 1 \dots M$ for $k = 1 \dots N$ and the distance from each node to the object origin i.e $\mathbf{r}_m$. A full description of deformable body modeling using this method is given in Appendix A.

3.3.4 System Resolution & Identification

The rigid arm variables and the flexible object variables are combined to derive two system Jacobian matrices, denoted as $\mathbf{J}_p$ and $\mathbf{J}_e$. The matrix $\mathbf{J}_p^{-T}$ and $\mathbf{J}_e^{-T}$ relate the joint torques to the rigid and elastic object variables, respectively and are written as:

$$\begin{aligned}\mathbf{J}_p^{-T} &= \left[\begin{array}{ccc}\mathbf{W}_{p1}\mathbf{L}_1\mathbf{J}_{a1}^{-T} & \dots & \mathbf{W}_{pn}\mathbf{L}_n\mathbf{J}_{an}^{-T}\end{array}\right] \\ \mathbf{J}_e^{-T} &= \left[\begin{array}{ccc}\mathbf{W}_{e1}\mathbf{L}_1\mathbf{J}_{a1}^{-T} & \dots & \mathbf{W}_{en}\mathbf{L}_n\mathbf{J}_{an}^{-T}\end{array}\right]\end{aligned} \tag{3.70}$$

where $\mathbf{J}_p^{-T}$ is $(6 \times a)$ matrix and $\mathbf{J}_e^{-T}$ is $(N \times a)$ matrix. Finally, the complete system Jacobian matrix is defined as :

$$\mathbf{W}_s = \left[\begin{array}{c}\mathbf{J}_p^{-T} \\ \mathbf{J}_e^{-T}\end{array}\right] \tag{3.71}$$

This matrix relates the object variables, both rigid and elastic, to the actuated joint variables. It should be noted that $\mathbf{J}_p^{-1}$ and $\mathbf{J}_e^{-1}$ are defined directly and are not obtained from inverting any matrices.

The system is resolved by eliminating the variables at the grasping points. Firstly the acceleration of the end effector is replaced in (3.48) by using (3.64):

$$\left[\begin{array}{c}\mathbf{h}_1 \\ \vdots \\ \mathbf{h}_n\end{array}\right] = \left[\begin{array}{ccc}\mathbf{J}_{a1}^{-T} & \dots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \dots & \mathbf{J}_{an}^{-T}\end{array}\right]\left[\begin{array}{c}\boldsymbol{\tau}_1 \\ \vdots \\ \boldsymbol{\tau}_n\end{array}\right] - \mathbf{A}_x\mathbf{L}^T\left(\left[\begin{array}{cc}\mathbf{W}_p^T & \mathbf{W}_e^T\end{array}\right]\left[\begin{array}{c}\dot{\mathbf{V}}_p \\ \ddot{\mathbf{q}}_e\end{array}\right] + \mathbf{b}\right) - \mathbf{c}_x \tag{3.72}$$

Secondly, by transforming (3.72) to the object's origin using (3.62), and using the relationships given in (3.70), the complete dynamic model of the system is given as:

$$\mathbf{W}_p\mathbf{L}\left[\begin{array}{c}\mathbf{h}_1 \\ \vdots \\ \mathbf{h}_n\end{array}\right] = \mathbf{J}_p^{-T}\boldsymbol{\tau} - \mathbf{W}_p\mathbf{L}\left(\mathbf{A}_x\mathbf{L}^T\left(\mathbf{W}_p^T\dot{\mathbf{V}}_p + \mathbf{W}_e^T\ddot{\mathbf{q}}_e + \mathbf{b}\right) + \mathbf{c}_x\right) \tag{3.73}$$

$$\mathbf{W}_e\mathbf{L}\left[\begin{array}{c}\mathbf{h}_1 \\ \vdots \\ \mathbf{h}_n\end{array}\right] = \mathbf{J}_e^{-T}\boldsymbol{\tau} - \mathbf{W}_e\mathbf{L}\left(\mathbf{A}_x\mathbf{L}^T\left(\mathbf{W}_e^T\dot{\mathbf{V}}_p + \mathbf{W}_e^T\ddot{\mathbf{q}}_e + \mathbf{b}\right) + \mathbf{c}_x\right) \tag{3.74}$$

The next step is to introduce the platform dynamics into the above expressions. By using (3.62), the Newton-Euler equation of the flexible object given in (3.66) can be rewritten in terms of the forces at the connection points.

$$\left[\begin{array}{c}\mathbf{W}_p \\ \mathbf{W}_e\end{array}\right]\mathbf{L}\left[\begin{array}{c}\mathbf{h}_1 \\ \vdots \\ \mathbf{h}_n\end{array}\right] = \left[\begin{array}{cc}\mathbf{A}_r & \mathbf{A}_{re} \\ \mathbf{A}_{re}^T & \mathbf{m}_{ee}\end{array}\right]\left[\begin{array}{c}\dot{\mathbf{V}}_p \\ \ddot{\mathbf{q}}_e\end{array}\right] + \left[\begin{array}{c}\mathbf{c}_r \\ \mathbf{c}_e\end{array}\right] \tag{3.75}$$

In order to simplify the development, the matrix $\mathbf{L}$ is dropped, therefore:

$$\mathbf{W}_p = \mathbf{W}_p\mathbf{L}, \quad \mathbf{W}_p^T = \mathbf{L}^T\mathbf{W}_p^T, \quad \mathbf{W}_e = \mathbf{W}_e\mathbf{L}, \quad \mathbf{W}_e^T = \mathbf{L}^T\mathbf{W}_e^T, \quad \mathbf{b} = \mathbf{L}^T\mathbf{b} \tag{3.76}$$

Finally, by equating (3.73), (3.74) and (3.75), the system dynamics are obtained as:

$$\mathbf{W}_s\boldsymbol{\tau} = \begin{bmatrix} \mathbf{A}_r + \mathbf{W}_p\mathbf{A}_x\mathbf{W}_p^T & \mathbf{A}_{re} + \mathbf{W}_p\mathbf{A}_x\mathbf{W}_e^T \\ \mathbf{A}_{re}^T + \mathbf{W}_e\mathbf{A}_x\mathbf{W}_p^T & \mathbf{m}_{ee} + \mathbf{W}_e\mathbf{A}_x\mathbf{W}_e^T \end{bmatrix} \begin{bmatrix} \dot{\mathbf{V}}_p \\ \ddot{\mathbf{q}}_e \end{bmatrix} + \begin{bmatrix} \mathbf{c}_r + \mathbf{W}_p\left(\mathbf{A}_x\mathbf{b} + \mathbf{c}_x\right) \\ \mathbf{c}_e + \mathbf{W}_e\left(\mathbf{A}_x\mathbf{b} + \mathbf{c}_x\right) \end{bmatrix} \tag{3.77}$$

The matrix $\mathbf{W}_s$ has dimension $(6 + N \times a)$. In the following section, the closed chain system is analyzed by examining this matrix, which in turn leads to an object classification.

3.3.5 Case Studies

A solution for three classes of objects are given for the planar cooperative system shown in Fig.3.6. The system is constructed using the dynamic simulator MSC ADAMS. MSC ADAMS is a widely used dynamic multi-body simulation software. By using the ADAMS Flex module, finite element bodies can be integrated into the simulator environment. The flexibility is represented by a series of mode shapes obtained from a finite element analysis created using MSC Nastran. It should be noted that the flexible body modeling in Adams is a variation of the component mode synthesis described in Chapter 3, Section 3.2. This means that the arbitrary disabling and enabling of modes is no longer possible. Instead, the strain energy over a trajectory can be used as a criterion to disable modes. The finite element model of the flexible object is integrated into the rigid body model of the two arms.

The objective is to validate the proposed model using a commercial dynamic simulator as shown as illustrated in Fig.3.7. A trajectory for an equivalent rigid body system is used to generate a joint torque. This torque is sent to the MSC ADAMS model. During the trajectory, the following quantities, required for the validation of inverse and direct dynamic model, are recorded.

– The Cartesian position, velocity and acceleration
– The joint torques
– The generalized elastic position, velocity and acceleration variables

All other quantities can be calculated from the algorithm. The output of the dynamic model derived in (3.77) is compared with the output of the commercial simulator.

System Description

The robotic system consists of two planar manipulators (functioning in the $X - Y$ plane) handling a flexible object. The geometric parameters of the links, $j = 1 \ldots 4$, for

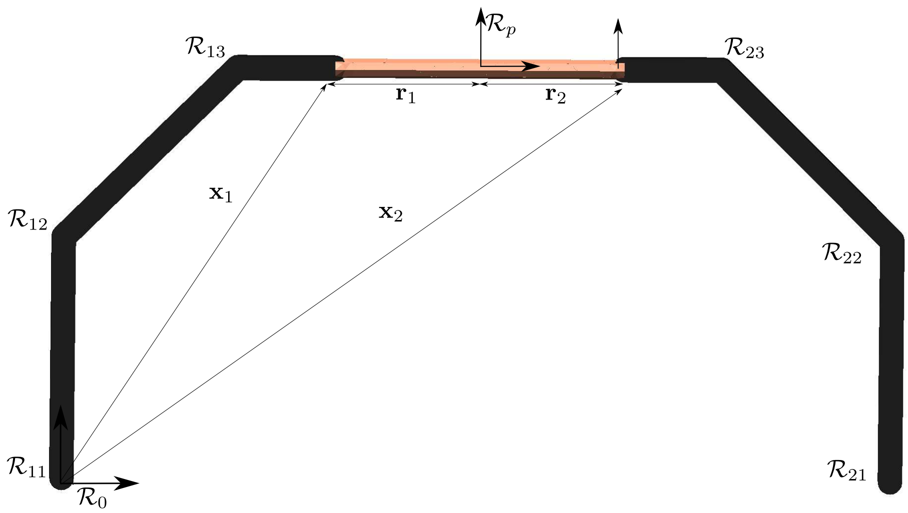

Figure 3.6: Dual arm robots with object

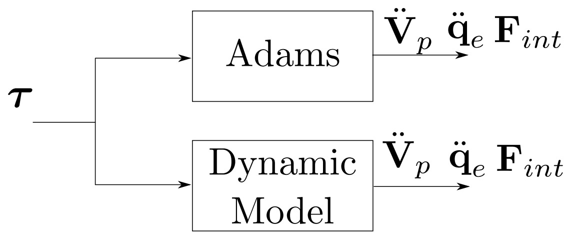

Figure 3.7: Validation Procedure of Dynamic Model

each leg i are given in Table 3.2, for $i = 1 \ldots 2$. $\sigma_{ji} = 1$ indicates the joint is prismatic and $\sigma_{ji} = 0$ indicates that the joint is revolute while $\sigma_{ji} = 2$ denotes a fixed joint. The parameters γ_{ji}, b_{ji}, α_{ji}, d_{ji}, θ_{ji} and r_{ji} define the location of frame j of leg i. Frame 4 represents the attachment point i.e. the i^{th} end effector. The inertial parameters are given in Table 3.3. The parameters that are zero have no effect on the model.

Both robots contain 3 revolute joints, thus $m_1 = m_2 = 3$. Each revolute joint may be actuated or passive depending on the current case study. There is an offset between the two robots along the global X axis of 1.7 meters. Gravity acts in the negative Y direction. The initial value of the joints is for robot 1 and robot 2 are given respectively as: $\mathbf{q}_1 = \begin{bmatrix} \frac{\pi}{2} & -\frac{\pi}{4} & -\frac{\pi}{4} \end{bmatrix}$ and $\mathbf{q}_2 = \begin{bmatrix} \frac{\pi}{2} & \frac{\pi}{4} & \frac{\pi}{4} \end{bmatrix}$.

The object is a cylinder modeled using finite element methods, with 382 nodal coordinates, as described in the Appendix. The object has a total mass of 5.03 kg and has 22 non-rigid modes that represent the flexibility. Table 3.4 contains the natural frequency, generalized stiffness and generalized damping of each mode. It should be noted that the modes are normalized such that the generalized mass, $\mathbf{m}_{ee}$, is the identity matrix.

Table 3.2: Geometric parameters for Arm i

j_i	σ_{ji}	γ_{ji}	b_{ji}	α_{ji}	dji	θ_{ji}	r_{ji}
1_i	0	0	0	0	0	q_{1i}	0
2_i	0	0	0	0	0.5	q_{2i}	0
3_i	0	0	0	0	0.5	q_{3i}	0
4_i	2	0	0	0	0.2	0	0

Table 3.3: Base Inertial parameters of Cooperative Arms

Link	1	2	3
ZZ (kg m^2)	5.0	5.0	4.0
XY (kg m^2)	0	0	0
XZ (kg m^2)	0	0	0
YZ (kg m^2)	0	0	0
MX (kg m)	0	0	0
MY (kg m)	0	0	0
MZ (kg m)	0	0	0
M (kg)	10.0	5.0	5.0

Since both robots act in a plane, only a planar wrench may be transmitted to the object. Therefore, for $c_1 = c_2 = 3$. The selection matrix, **L** is defined as follows:

$$\mathbf{L}_1 = \mathbf{L}_2 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 1 \end{bmatrix} \qquad \mathbf{L}_1 = \begin{bmatrix} \mathbf{L}_1 & \mathbf{0}_{6\times3} \\ \mathbf{0}_{6\times3} & \mathbf{L}_2 \end{bmatrix} \tag{3.78}$$

Three different cases are studied where the object flexibility is modified in each case as shown in Table 3.5. For cases 1,2 and 3 the resulting grasped object is referred to as rigid, flexible and articulated, respectively. It should be noted that the object definition is no longer simply dependent on its degree of flexibility but also on the definition of the cooperative system that grasps it.

In each case, the object is moved through a point to point trajectory in Cartesian space by the dual arm system. Finally, it should also be noted that the object is not fully fixed to the first robot, instead a prismatic joint is used that allows out of plane vibrations. This is required by the commercial simulator to solve the articulated object case.

Table 3.4: Modal properties of Flexible Object

Mode No.	Natural frequency (rad/s)	Generalized Stiffness	Damping ratio
7	1232.0	1517729.372	0.5
8	1233.0	1520171.4942	0.5
9	3275.5	10729141.534	0.5
10	3286.6	10801919.436	0.5
11	6222.4	38718880.06	0.5
12	6255.4	39129513.552	0.5
13	7375.2	54393868.1	0.5
14	9546.1	91128963.767	0.5
15	9937.7	98757174.305	0.5
16	9965.5	99311604.882	0.5
17	13124.0	172226682.76	0.5
18	16018.0	256574387.36	0.5
19	16179.0	261763969.65	0.5
20	17315.0	299817191.08	0.5
21	22069.0	487021427.75	0.5
22	22508.0	506620691.74	0.5
23	48255.0	2328512285.9	0.5
24	52224.0	2727319144.4	0.5
25	84302.0	7106905288.8	0.5
26	85060.0	7235228610.1	0.5
27	98925.0	9786239875.9	0.5
28	101570	10317478861	0.5

Table 3.5: Summary of the case studies for general dynamic model, showing the object flexibility and the number of independent actuators in the system

Case Study	Object Type	a	Actuated joints	N	Modes	dim($\mathbf{W}_s$)
1	Rigid	6	$\mathbf{q}_{11}$, $\mathbf{q}_{12}$, $\mathbf{q}_{13}$, $\mathbf{q}_{21}$, $\mathbf{q}_{22}$, $\mathbf{q}_{23}$	0	-	3×6
2	Flexible	3	$\mathbf{q}_{11}$, $\mathbf{q}_{12}$, $\mathbf{q}_{21}$	22	7-28	28×6
3	Articulated	3	$\mathbf{q}_{11}$, $\mathbf{q}_{12}$, $\mathbf{q}_{21}$	3	7,8,11	6×6

Case 1: Rigid Object

The system is simulated with 6 actuated joints. The object is rigid i.e. $\mathbf{q}_e = \dot{\mathbf{q}}_e = \ddot{\mathbf{q}}_e = 0$. There are no elastic forces in the dynamic model and the elastic part of (3.77) is ignored. Therefore $\mathbf{W}_s = \mathbf{J}_p^{-T}$ is a (3×6) matrix. In this case, the solution is identical to that of a parallel robot with rigid legs described in [KI07], but represented in the Cartesian space:

$$\mathbf{W}_s\boldsymbol{\tau} = (\mathbf{A}_r + \mathbf{W}_p\mathbf{A}_x\mathbf{W}_p^T)\dot{\mathbf{V}}_p + \mathbf{c}_r + \mathbf{W}_p\left(\mathbf{A}_x\mathbf{b} + \mathbf{c}_x\right) \tag{3.79}$$

From the structure of $\mathbf{W}_s$, it is clear that the system is redundantly actuated, thus the object may undergo internal loading. By definition, the internal loading has no effect on the motion causing forces. Therefore inverting (3.79) results in the classical formulation for cooperative manipulators handling a rigid object [BH96] and as described in Chapter 2.

$$\boldsymbol{\tau} = \begin{bmatrix} (\mathbf{W}_s)^{(+)} & \boldsymbol{\Lambda} \end{bmatrix} \begin{bmatrix} (\mathbf{A}_r + \mathbf{W}_p\mathbf{A}_x\mathbf{W}_p^T)\dot{\mathbf{V}}_p + \mathbf{c}_r + \mathbf{W}_p\left(\mathbf{A}_x\mathbf{b} + \mathbf{c}_x\right) \\ \mathbf{F}_{int} \end{bmatrix} \tag{3.80}$$

where $\boldsymbol{\Lambda}$ may be any matrix that spans the null space of $\mathbf{W}_s$ and $\mathbf{F}_{int}$ denotes the internal forces.

The internal loading and planar Cartesian acceleration, $\dot{\mathbf{V}}_p$, are calculated from the joint torque and robot's state as shown in Fig.3.8 and in Fig.3.9, respectively. From Fig.3.8, it is clear that the model can successfully predict internal loading in the object from the robot state and recorded joint torques. The object's calculated Cartesian acceleration is compared with the Adams equivalent in Fig.3.9. A good correlation is seen between the Adams model and the predicted values.

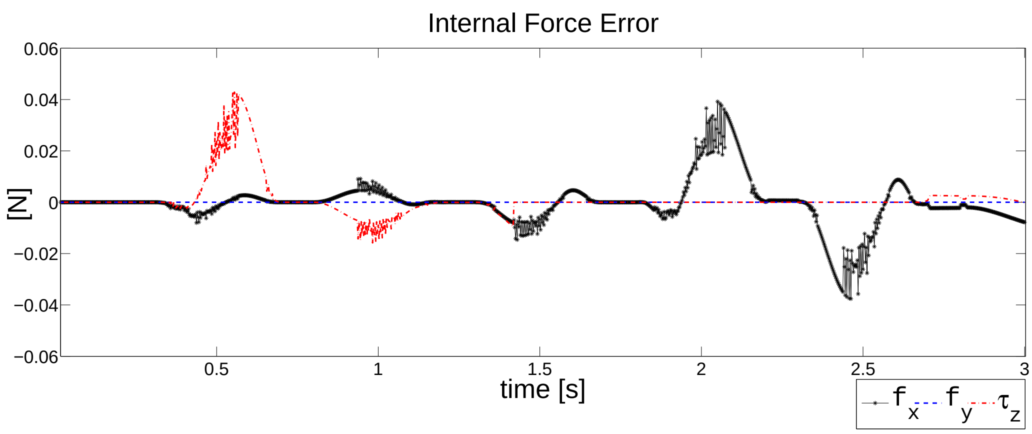

Figure 3.8: Rigid Object Forces: Difference between Adams and predicted model for desired internal force $\mathbf{F}_{int} = [0 \quad 100 \quad 0]$

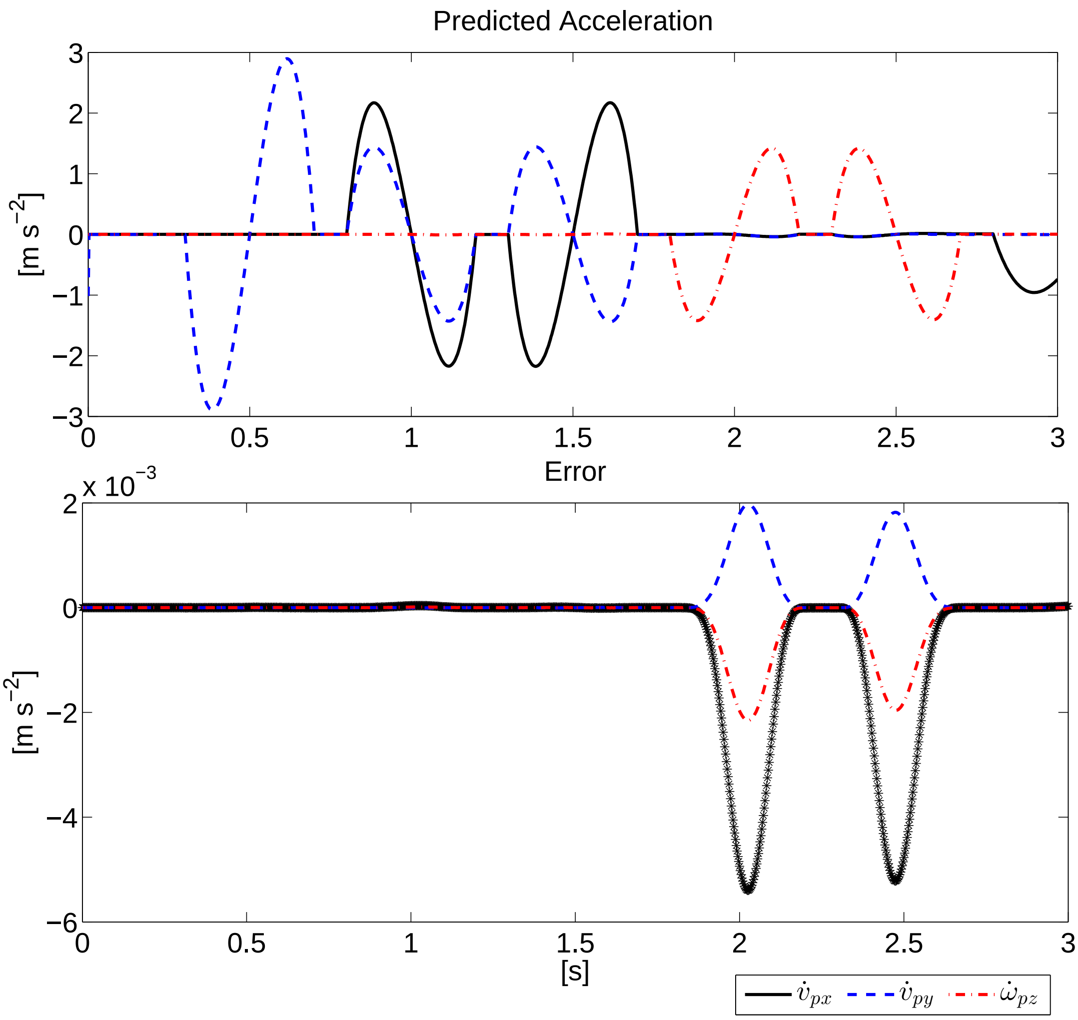

Figure 3.9: Rigid Object Acceleration: (Top) Predicted acceleration of object, (Bottom) Difference between Adams and predicted model

Case 2: Flexible Object

The object is flexible; the elastic variables cannot be controlled, but may be damped by a judicious choice of controller.

In this case, the system contains 3 actuated joints and the object has a degree of flexibility of dimension 22. This means that $\mathbf{W}_s$ is a rectangular matrix of dimension (28×3) and thus can not be inverted. To obtain the joint torques a 2-step solution is required. Firstly using (3.77) the elastic variables are rewritten as:

$$\ddot{\mathbf{q}}_e = \mathbf{A}_{ee}^{-1}\left(\mathbf{J}_e^{-T}\boldsymbol{\tau} - \left(\mathbf{A}_{re}^T + \mathbf{W}_e\mathbf{A}_x\mathbf{W}_p^T\right)\dot{\mathbf{V}}_p - \mathbf{W}_e\mathbf{A}_x\mathbf{b}\right) - \mathbf{A}_{ee}^{-1}\left(\mathbf{W}_e\mathbf{c}_x - \mathbf{c}_e\right) \quad (3.81)$$

where for convenience

$$\mathbf{A}_{ee}^{-1} = \left(\mathbf{m}_{ee} + \mathbf{W}_e\mathbf{A}_x\mathbf{W}_e^T\right)^{-1} \quad (3.82)$$

Equation (3.81) is then back-substituted into (3.77) to obtain the dynamic model of the manipulator:

$$\mathbf{A}\dot{\mathbf{V}}_p + \mathbf{c} = \mathbf{J}_{sys}^{-T}\boldsymbol{\tau} \quad (3.83)$$

This dynamic model can be used to solve both the direct and the inverse dynamic problems. The 6×6 matrix $\mathbf{A}$, the equivalent total inertia matrix of the legs and the flexible object, is written as:

$$\begin{aligned}\mathbf{A} = \mathbf{A}_{rr} + \mathbf{W}_p\mathbf{A}_x\mathbf{W}_p^T - \mathbf{A}_{re}\mathbf{A}_{ee}^{-1}\left(\mathbf{A}_{re}^T + \mathbf{W}_e\mathbf{A}_x\mathbf{W}_p^T\right) \\ - \mathbf{W}_p\mathbf{A}_x\mathbf{W}_e^T\mathbf{A}_{ee}^{-1}\left(\mathbf{A}_{re}^T + \mathbf{W}_e\mathbf{A}_x\mathbf{W}_p^T\right)\end{aligned} \quad (3.84)$$

The 6×6 Jacobian matrix is given by:

$$\mathbf{J}_{sys}^{-T} = \mathbf{J}_p^{-T} - \left(\mathbf{A}_{re} + \mathbf{W}_p\mathbf{A}_x\mathbf{W}_e^T\right)\mathbf{A}_{ee}^{-1}\mathbf{J}_e^{-T} \quad (3.85)$$

The 6×1 vector $\mathbf{c}$, the total Coriolis, centrifugal and gravity torques of the legs and the flexible object, is given as:

$$\begin{aligned}\mathbf{c} = \mathbf{c}_r + \mathbf{W}_p\mathbf{A}_x\mathbf{h} + \mathbf{W}_p\mathbf{c}_x - \\ \left(\mathbf{A}_{re} + \mathbf{W}_p\mathbf{A}_x\mathbf{W}_e^T\right)\mathbf{A}_{ee}^{-1}\left(\mathbf{W}_e\mathbf{A}_x\mathbf{h} + \mathbf{W}_e\mathbf{c}_x + \mathbf{c}_e\right)\end{aligned} \quad (3.86)$$

To validate the model the Cartesian acceleration is obtained by inverting the *positive-definite* total inertia matrix to solve (3.83) from the given joint torques. Once $\dot{\mathbf{V}}_p$ is obtained the generalized elastic variables are calculated from (3.81).

Fig.3.10 shows a small errors between the ADAMS output and the calculated Cartesian accelerations despite large vibrations indicating a good level of agreement between the two models. In addition to this, Fig.3.11 shows a comparison between the ADAMS output and the calculated of the generalized elastic variables' acceleration for three high magnitude modes, thus demonstrating how the model can accurately predict vibration in the system.

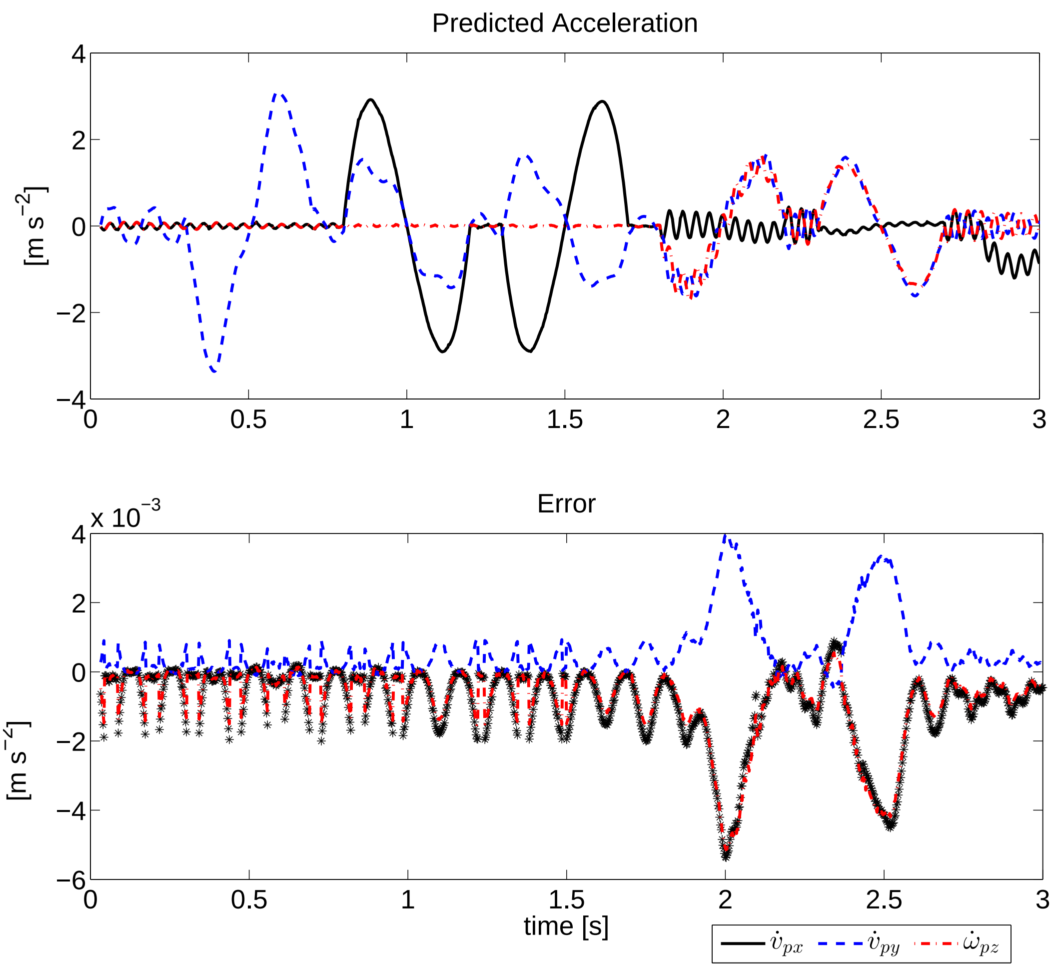

Figure 3.10: Flexible Object: (Top) Predicted Acceleration of flexible object, (Bottom) Difference between Adams and predicted model

Case 3: Articulated Object

In this case, there are sufficient actuators to fully control the object's state. The system contains 6 actuated joints and the object has a degree of flexibility of dimension 3. For this type of system, the vibration induced by the flexibility can be perfectly suppressed by the joint controller. However, there may be cases where the vibration is uncontrollable [ZZG00]. For example, in the above system, the vibration of certain modes may not be controlled if they are acting out of plane.

If $\mathbf{W}_s$ has full rank. From examination of (3.77), it is clear that by inverting $\mathbf{W}_s$, joint torques can be obtained that allow the system to perfectly achieve a desired rigid body and elastic generalized acceleration:

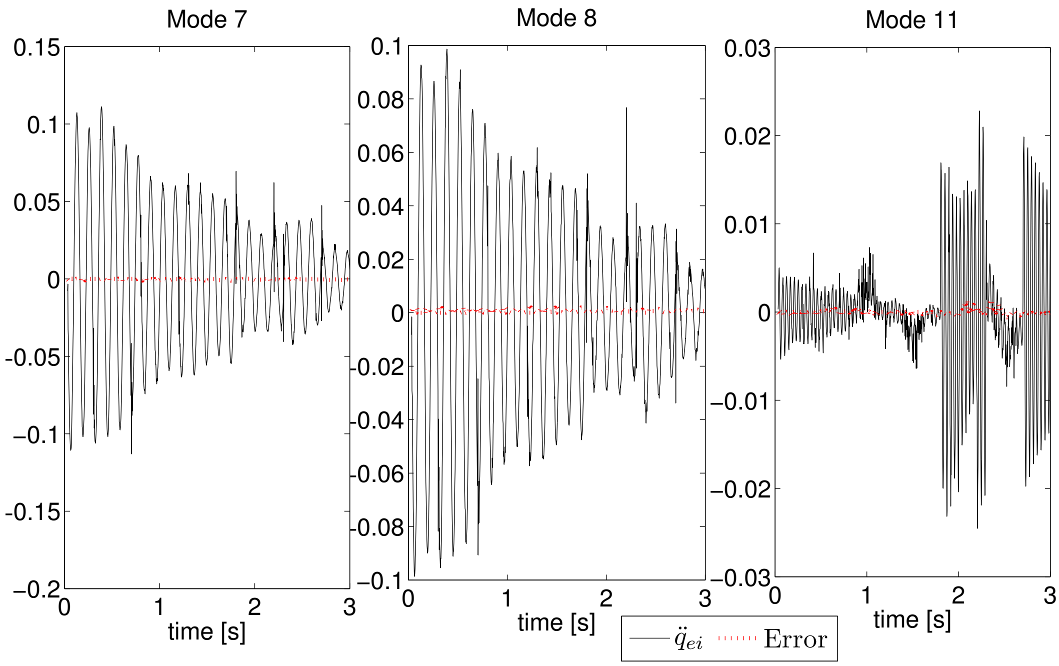

Figure 3.11: Flexible Object: Error between predicted & measured acceleration for high magnitude modes

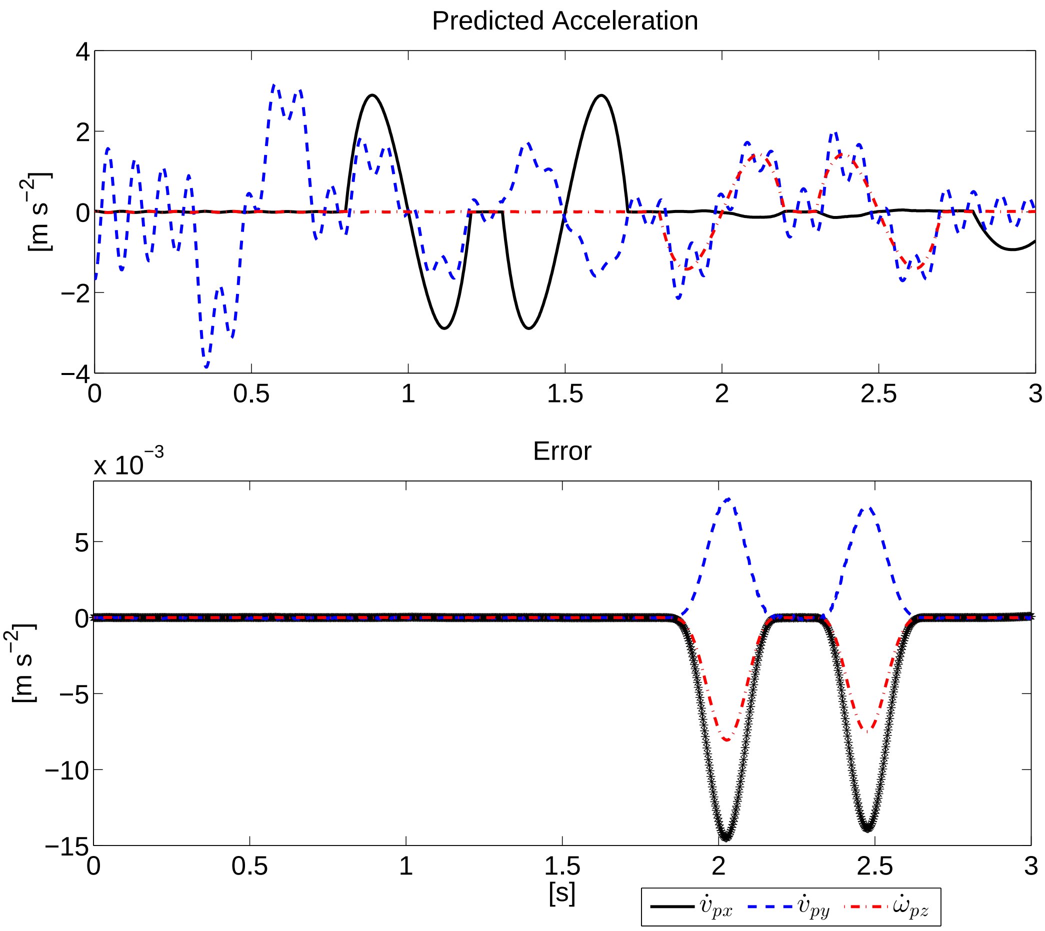

Figure 3.12: Articulated Object: (Top) Predicted Acceleration of flexible object, (Bottom) Difference between Adams and predicted model

$$\boldsymbol{\tau} = \mathbf{W}_s^{-1}\left(\begin{bmatrix} \mathbf{A}_r + \mathbf{W}_p\mathbf{A}_x\mathbf{W}_p^T & \mathbf{A}_{re} + \mathbf{W}_p\mathbf{A}_x\mathbf{W}_e^T \\ \mathbf{A}_{re}^T + \mathbf{W}_e\mathbf{A}_x\mathbf{W}_p^T & \mathbf{m}_{ee} + \mathbf{W}_e\mathbf{A}_x\mathbf{W}_e^T \end{bmatrix}\begin{bmatrix} \dot{\mathbf{V}}_p \\ \ddot{\mathbf{q}}_e \end{bmatrix} + \begin{bmatrix} \mathbf{c}_r + \mathbf{W}_p\left(\mathbf{A}_x\mathbf{b} + \mathbf{c}_x\right) \\ \mathbf{c}_e + \mathbf{W}_e\left(\mathbf{A}_x\mathbf{b} + \mathbf{c}_x\right) \end{bmatrix}\right) \tag{3.87}$$

To validate the model the Cartesian acceleration and the generalized elastic variables are calculated from the joint torques directly by solving (3.77). Fig.3.12 shows a comparison between the ADAMS output and the calculated Cartesian accelerations. Fig.3.13 shows a comparison between the ADAMS output and the calculated of the generalized elastic variables' acceleration. In both cases a small error is seen between the results obtained by the commercial simulator and the results of the dynamic model.

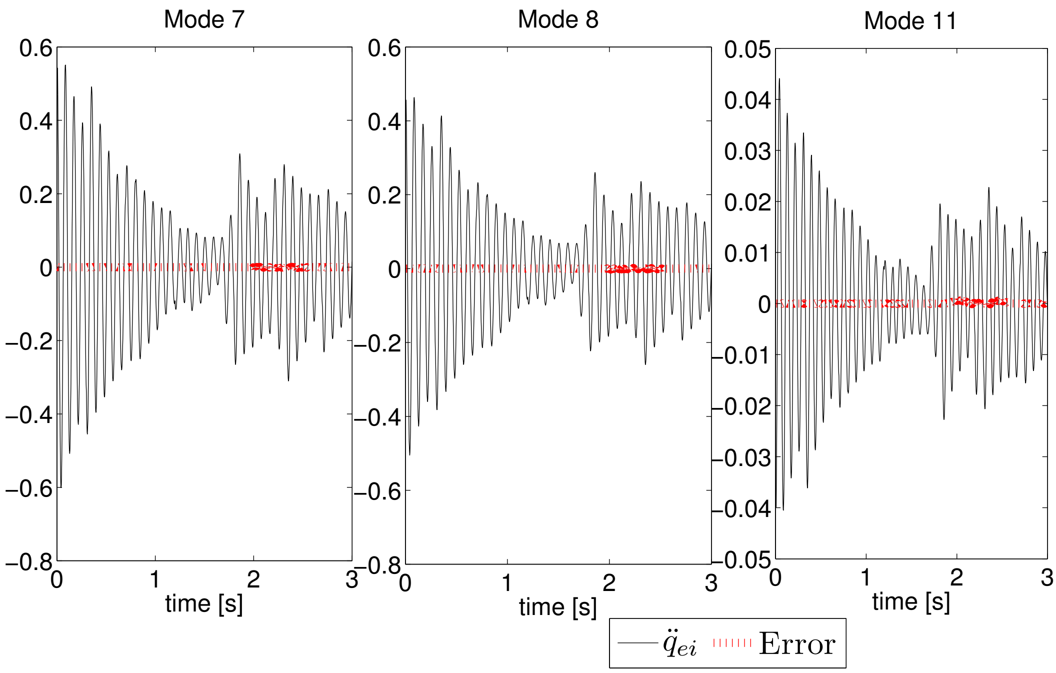

Figure 3.13: Articulated Object: Error between predicted & measured acceleration for active modes

3.3.6 Object Identification

From the dynamic formulation and case studies given in Section 3.3.5. An identification procedure for objects based on the form of $\mathbf{W}_s$ be carried out.

Rigid Object $\mathbf{W}_s$ has rank 6 where $N = 0$. In this case the object is rigid, and the dimension of internal forces that can be created is equal to $\mathcal{N}(\mathbf{W}_{sys})$.

Flexible Object: $\mathbf{W}_s$ is a rectangular matrix where $(6+N) > (a)$. The object is flexible; the elastic variables cannot be controlled but can be damped by a judicious choice of controller. $\mathbf{W}_s$ cannot be inverted, therefore a 2-step solution is used to first obtain the elastic variables in terms of the joint torques followed by substituting the result into (3.77).

Articulated Objects: $\mathbf{W}_s$ has full rank and is square i.e. the system has $6+N$ independent actuators. From examination of (3.87), it is clear that by inverting $\mathbf{W}_s$, joint torques can be obtained that allow the system to perfectly achieve a desired rigid body and elastic generalized acceleration. For this type of system, the vibration induced by the flexibility can be perfectly suppressed. Alternatively, a desired deformation can be achieved.

Object of reduced flexibility: $\mathbf{W}_s$ is a rectangular matrix, where $6 < (6+N) < (a)$ $(N \neq 0)$. In order to resolve (3.77), a generalized inverse of $\mathbf{W}_s$ can be taken. However, due to the existence of the null space of $\mathbf{W}_s$, i.e. $\mathcal{N}(\mathbf{W}_s)$, a term representing the internal forces on the object appears. This case is common for cooperative manipulators and the resolution is discussed in Section 2.2. Thus, the system is redundantly actuated and despite the flexibility of the object, the system is capable of creating an internal force of dimension equal to the $\mathcal{N}(\mathbf{W}_s)$.

3.4 Dynamic Modeling of Gough Stewart Robot with flexible platform

In order to illustrate the applicability of this method, a step by step approach is carried out for a Gough-Stewart manipulator with flexible platform. The rigid elements of the robot, which consist of the legs and the fixed base of the legs, are described as a tree structure robot using the Modified Denavit Hartenberg Parameters [KK86] and modeled using joint variables. The platform is considered flexible and modeled using the generalized Newton Euler method [BK98, BKBLV07]. The systems are linked using by the procedure outlined in Section 3.3.

3.4.1 Background

The dynamic modeling of Gough-Stewart robot with rigid elements has attracted many works with different algorithms. For instance, the Lagrange-Euler formalism has been used in the works of Lee and Shah [LS88], Geng et al. [GHLC92] and Lebret et al. [LLL93], Ait-Ahmed [AA93], Bhattacharya et al. [BHG97, BNU98] and Liu et al. [LLL00]. The principle of virtual work has been used by Tsai [Tsa00], Codourey [CB97] and Staicu [Sta11]. On the other hand, Newton-Euler equations have been used in the work of Sugimoto [Sug89], Reboulet et al. [RB91], Ji [Ji93], Gosselin [Gos93] and Dasgupta et al. [DM98, DC99]. However, recently, Carricato and Gosselin [CG09], Afroun et al [ADV12], Fu et al [FYW07] and Vakil et al [VPZ08], have pointed out common errors in many methods related to parameterization and instantaneous kinematic behavior of the legs. These errors may cause kinematic and dynamic miscalculations. The correct dynamic modeling of the rigid Gough-Stewart robot, which avoids these errors, has been demonstrated using different formalisms. For example using screw theory in Gallardo et al [GRF+03], the Newton-Euler approach in Khalil and Guegan [KG02], Khalil and Ouarda [KI07] and by Lagrange methods in Abdellatif and Heimann [AH09].

In the literature the main approaches to model flexibility in parallel robots are concerned with limb flexibility, this is because the limb's flexibility can be approximated using beam elements. For instance, for the Gough-Stewart robot the effects of leg flexibility are examined in [MDM07, MNK09]. The optimum choice of flexibility representation is investigated in [WW03]. In [KG00, Rod90] lumped spring mass approximations have been used.

The aim of this work is to extend the dynamic method first described in [KG02] and [KI07] to parallel robots with flexible platforms by using the modeling technique derived in Section 3.3. There are several applications for this work. For example, for robots with large platforms, where flexibility can no longer be neglected. The platform's flexibility can be taken into account in the design of the controller, thanks to this model.

Alternatively, for robots that carry out high speed machining tasks, during which large vibrations are induced. Generally to counteract this, the platform's mass is increased until the effects of vibration are negligible. This solution leads to manipulators with high mass and greater energy consumption.

To give an idea of the dimension involved, consider CMW's 6-DOF parallel robot the hexapode. The platform of this robot has a mass of over 200 kg with a diameter of $600mm$. The total mass of the system is 900kg. The maximum speed is just over 0.8m/s. If the flexibility is modeled, these manipulators can be designed with low weight platforms, thereby reducing the total mass and permitting the use of high acceleration trajectories.

3.4.2 Leg System Description and Modeling

Geometric Parameters

The studied system is a Gough-Stewart structure, as shown in Fig.3.14. The platform has 6-DOF and is connected to the fixed base by six legs. Each leg is connected to the base with a 2-DOF universal joint (U-joint) and to the platform with a 3-DOF spherical joint (S-joint). Each leg has a variable length by means of an actuated prismatic joint (P-joint).

The base frame and the platform frame are denoted by $\mathbf{\Sigma}_o$ and $\mathbf{\Sigma}_p$, respectively. The connection points between the base and the U-Joints are denoted as $\mathbf{b}_i$ and are arranged according to the convention established in [KG02]. The connection points between the platform origin and the legs are denoted as $\mathbf{p}_i$, for $i = 1 \dots 6$.

After opening virtually the spherical joints, each leg i is composed of three joints and three links. The geometric parameters of the links, $j = 1 \dots 3$, for each leg i are given in Table 3.6, for $i = 1 \dots 6$.

$\mu_{ji} = 1$ for an actuated joint or $\mu_{ji} = 0$ for a passive joint. $\sigma_{ji} = 1$ indicates the joint is prismatic and $\sigma_{ji} = 0$ indicates that the joint is revolute. The parameters γ_{ji}, b_{ji}, α_{ji}, d_{ji}, θ_{ji} and r_{ji} define the location of frame j of leg i, defined as $\mathbf{\Sigma}_{ji}$, with respect to its antecedent frame.

Table 3.6: Geometric parameters for leg i

ji	μ_{ji}	σ_{ji}	γ_{ji}	b_{ji}	α_{ji}	dji	θ_{ji}	r_{ji}
1_i	0	0	γ_{1i}	b_{1i}	$-\frac{\pi}{2}$	d_{1i}	q_{1i}	0
2_i	0	0	0	0	$\frac{\pi}{2}$	0	q_{2i}	0
3_i	1	1	0	0	$\frac{\pi}{2}$	0	0	q_{3i}

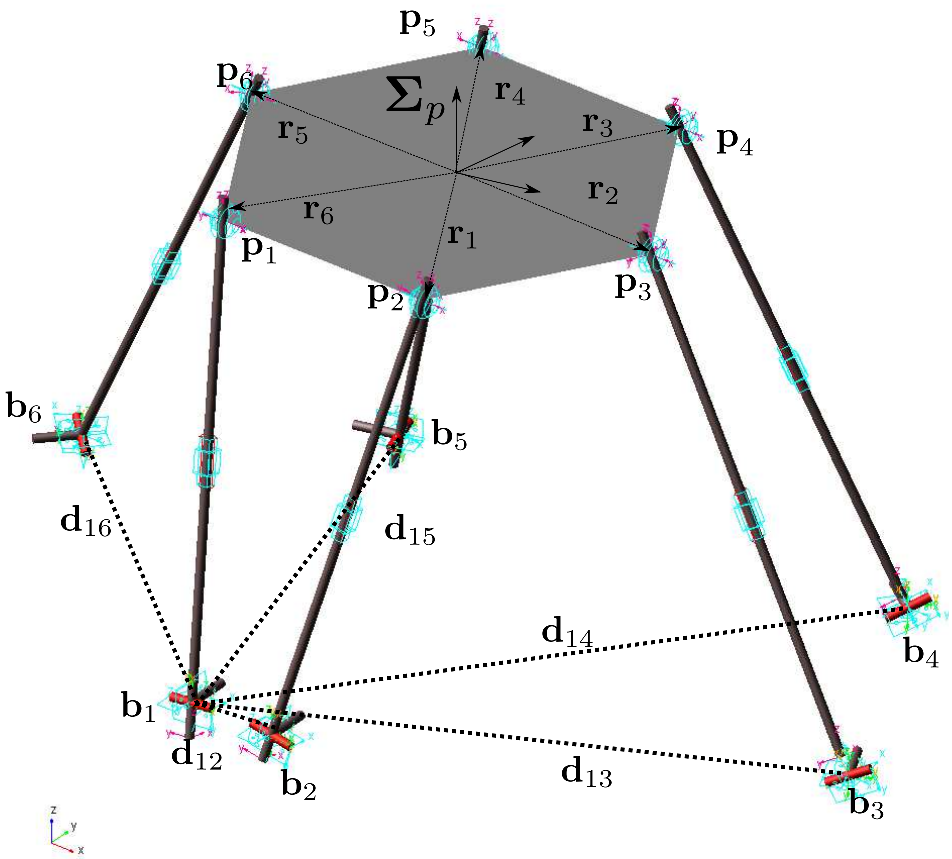

Figure 3.14: Gough-Stewart manipulator as modeled in MSC ADAMS

Kinematic Models of the Legs

The 3×1 vectors denoting joint position, velocity and acceleration for leg i are denoted as $\mathbf{q}_i$, $\dot{\mathbf{q}}_i$ and $\ddot{\mathbf{q}}_i$ respectively. The actuated joint of leg i is denoted as q_{3i} and the vector of all actuated joints of the system is given as $\mathbf{q}_a = \begin{bmatrix} q_{31} & q_{32} & q_{33} & q_{34} & q_{35} & q_{36} \end{bmatrix}^T$ where q_{3i} is the distance between $\mathbf{p}_i$ and $\mathbf{b}_i$. The velocity of the connection point, $\mathbf{p}_i$, is a linear velocity $\mathbf{v}_i$ which can be obtained from the joint velocity of the corresponding leg using the kinematic Jacobian matrix of the leg:

$$\mathbf{v}_i = \mathbf{J}_i \dot{\mathbf{q}}_i \tag{3.88}$$

The inverse kinematic model is written as:

$$\dot{\mathbf{q}}_i = \mathbf{J}_i^{-1} \mathbf{v}_i \tag{3.89}$$

The Jacobian matrix of leg i represented in the frame of $\mathbf{\Sigma}_{0i}$ is given as:

$$
{}^{0i}\mathbf{J}_i = \begin{bmatrix} -q_{3i}\sin(q_{1i})\sin(q_{2i}) & q_{3i}\cos(q_{1i})\cos(q_{2i}) & \cos(q_{1i})\sin(q_{2i}) \\ 0 & q_{3i}\sin(q_{2i}) & -\cos(q_{2i}) \\ -q_{3i}cos(q_{1i})\sin(q_{2i}) & -q_{3i}\cos(q_{2i})\sin(q_{1i}) & -\sin(q_{1i})\sin(q_{2i}) \end{bmatrix} \tag{3.90}
$$

The inverse Jacobian matrix of leg i represented in the frame of $\mathbf{\Sigma}_{0i}$ is given as:

$$
{}^{0i}\mathbf{J}_i^{-1} = \begin{bmatrix} -\dfrac{\sin(q_{1i})}{(q_{3i}\sin(q_{2i}))} & 0 & -\dfrac{\cos(q_{1i})}{(q_{3i}\sin(q_{2i}))} \\ \dfrac{(\cos(q_{1i})\cos(q_{2i}))}{q_{3i}} & \dfrac{\sin(q_{2i})}{q_{3i}} & -\dfrac{(\cos(q_{2i})\sin(q_{1i}))}{q_{3i}} \\ \cos(q_{1i})\sin(q_{2i}) & -\cos(q_{2i}) & -\sin(q_{1i})\sin(q_{2i}) \end{bmatrix} \tag{3.91}
$$

It should be noted that the third row of the Jacobian matrix corresponds to the unit vector along the axis of the prismatic joint of the serial leg i, $\mathbf{a}_{3i}^T$. Therefore

$$
\mathbf{J}_{ai}^{-T} = \mathbf{a}_i \tag{3.92}
$$

This leads to an expression of the actuated joint velocity of leg i in terms of $\mathbf{v}_i$ as:

$$
\dot{q}_{3i} = \mathbf{a}_{3i}^T\mathbf{v}_i \tag{3.93}
$$

The second order inverse kinematic model of the leg is given by:

$$
\ddot{\mathbf{q}}_i = \mathbf{J}_i^{-1}\left(\dot{\mathbf{v}}_i - \dot{\mathbf{J}}_i\dot{\mathbf{q}}_i\right) \tag{3.94}
$$

$\mathbf{v}_i$ and $\dot{\mathbf{v}}_i$ are obtained from $\mathbf{V}_p$ and $\dot{\mathbf{V}}_p$, the velocity and acceleration of the platform respectively, using the elastic equations given in Section 3.4.3.

Inverse Dynamic Model of the legs

The inverse dynamic model of leg i is obtained by considering the tree structure sub-system of the legs (after separating the platform). Let the dynamic model of the 3 DOF system be given as $\boldsymbol{\tau}_i = \mathbf{A}_i(\mathbf{q}_i)\ddot{\mathbf{q}}_i + \mathbf{c}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i)$ where $\boldsymbol{\tau}_i$ represents the joint torques if the leg is not connected to the platform. The positions, velocities and accelerations of the joints are obtained from the position, velocity and acceleration of point $\mathbf{p}_i$ using the inverse kinematic model. $\mathbf{A}_i$ is the inertia matrix of leg i whereas $\mathbf{c}_i$ is the vector of Coriolis, Centrifugal and gravity torques. $\mathbf{\Gamma}_i$ is the torque of the closed loop structure of leg i, composed of the dynamic of open loop $\mathbf{A}_i(\mathbf{q}_i)\ddot{\mathbf{q}}_i + \mathbf{c}_i(\mathbf{q}_i, \dot{\mathbf{q}}_i)$ and the effect of the forces generated by the moving platform on the legs. It can be written as:

$$
\mathbf{\Gamma}_i = \mathbf{A}_i\ddot{\mathbf{q}}_i + \mathbf{c}_i + \mathbf{J}_i^T\mathbf{f}_i \tag{3.95}
$$

with

$$\mathbf{\Gamma}_i = \begin{bmatrix} 0 & 0 & \Gamma_{3i} \end{bmatrix}^T \tag{3.96}$$

The first two components of $\mathbf{\Gamma}_i$ are zero as they correspond to the torques of the passive joints. This equation permits the calculation of the reaction forces of the leg on the platform in terms of the actuated joint torques of the manipulator:

$$\mathbf{f}_i = \mathbf{J}_i^{-T} \left(\mathbf{\Gamma}_i - \mathbf{A}_i \ddot{\mathbf{q}}_i - \mathbf{c}_i \right) \tag{3.97}$$

Equations (3.91) and (3.96), (3.97) can be manipulated to leave an expression in terms of the torque of the actuated joint:

$$\mathbf{f}_i = \mathbf{a}_{3i} \Gamma_{3i} - \mathbf{J}_i^{-T} \left(\mathbf{A}_i \ddot{\mathbf{q}}_i + \mathbf{c}_i \right) \tag{3.98}$$

3.4.3 Modeling of flexible Gough-Stewart platform

Fig.3.15 shows the flexible platform. To obtain the geometric, kinematic and dynamic models the procedure demonstrated in Section 3.3.3 is repeated. For the Gough-Stewart platform, the connection point is defined by a spherical joint therefore for $i = 1 \ldots 6$:

$$\mathbf{L}_i = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{3.99}$$

In order to find the grasp matrix for the platform, the matrix defined by (3.63) is modified using (3.76) and (3.99) as follows:

$$\mathbf{W} = \mathbf{WL} = \left[\begin{array}{ccc} \mathbf{1}_3 & \ldots & \mathbf{1}_3 \\ \hat{\mathbf{r}}_1 & \ldots & \hat{\mathbf{r}}_6 \\ \hdashline \mathbf{\Phi}_d^T(1) & \ldots & \mathbf{\Phi}_d^T(6) \end{array} \right] = \left[\begin{array}{c} \mathbf{W}_p \\ \hdashline \mathbf{W}_e \end{array} \right] \tag{3.100}$$

The linear velocity at the spherical joints can be obtained from the platform velocity using (3.61) and (3.100):

$$\begin{bmatrix} \mathbf{v}_1 \\ \vdots \\ \mathbf{v}_n \end{bmatrix} = \begin{bmatrix} \mathbf{W}_p^T & \mathbf{W}_e^T \end{bmatrix} \begin{bmatrix} \mathbf{V}_p \\ \dot{\mathbf{q}}_e \end{bmatrix} \tag{3.101}$$

Similarly, from (3.59) and (3.100) a relationship is obtained linking the platform wrench to the forces transmitted by the spherical joints:

$$\begin{bmatrix} \mathbf{h}_p \\ \mathbf{Q}_p \end{bmatrix} = \begin{bmatrix} \mathbf{W}_p \\ \mathbf{W}_e \end{bmatrix} \begin{bmatrix} \mathbf{f}_1 \\ \vdots \\ \mathbf{f}_6 \end{bmatrix} \tag{3.102}$$

From (3.70), (3.92) and (3.100), the system Jacobian matrices can be obtained for the Gough Stewart platform as:

$$\begin{aligned} \mathbf{J}_p^{-T} &= \begin{bmatrix} \mathbf{a}_{31} & \dots & \mathbf{a}_{36} \\ \hat{\mathbf{r}}_1\mathbf{a}_{31} & \dots & \hat{\mathbf{r}}_6\mathbf{a}_{36} \end{bmatrix} \\ \mathbf{J}_e^{-T} &= \begin{bmatrix} \mathbf{\Phi}_d^T(1)\mathbf{a}_{31} & \dots & \mathbf{\Phi}_d^T(6)\mathbf{a}_{36}^T \end{bmatrix} \end{aligned} \tag{3.103}$$

Finally, it can be seen that:

$$\mathbf{J}_{pi}^{-T} = \mathbf{W}_{pi}\mathbf{a}_i = \left(\frac{\partial \dot{q}_{3i}}{\partial \mathbf{V}_p}\right)^T \qquad \mathbf{J}_{ei}^{-T} = \mathbf{W}_{ei}\mathbf{a}_i = \left(\frac{\partial \dot{q}_{3i}}{\partial \dot{\mathbf{q}}_e}\right)^T \tag{3.104}$$

3.4.4 Dynamic Model of the Gough Stewart Robot

In the following, the steps taken to calculate the dynamic model of the system are outlined. From the dynamic relations, the inverse dynamic problem, outlined in Section 3.4.4, and the direct dynamic problem, outlined in Section 3.4.4, can be solved. From (3.71) and (3.103), it is clear that the dimension of $\mathbf{W}_s$ is of dimension $(6+N\times 6)$.

$$\mathbf{W}_s = \begin{bmatrix} \mathbf{a}_{31} & \dots & \mathbf{a}_{36} \\ \hat{\mathbf{r}}_1\mathbf{a}_{31} & \dots & \hat{\mathbf{r}}_6\mathbf{a}_{36} \\ \mathbf{\Phi}_d^T(1)\mathbf{a}_{31} & \dots & \mathbf{\Phi}_d^T(6)\mathbf{a}_{36}^T \end{bmatrix} \tag{3.105}$$

Therefore, for any non-rigid object the solution required is the two step solution outlined for flexible objects for **Case 2: Flexible Object** in Section 3.3.5. Firstly, the acceleration of the generalized elastic coordinates are expressed as (3.81). Secondly, this term is back substituted to obtain the dynamic model given in (3.83)

Inverse Dynamic Problem

The inverse dynamic model of a parallel robot gives the actuated joint torques as a function of the desired trajectory of the platform frame and the current state of the robot. The main objective of this model is in non-linear control strategies, for instance the computed torque algorithm.

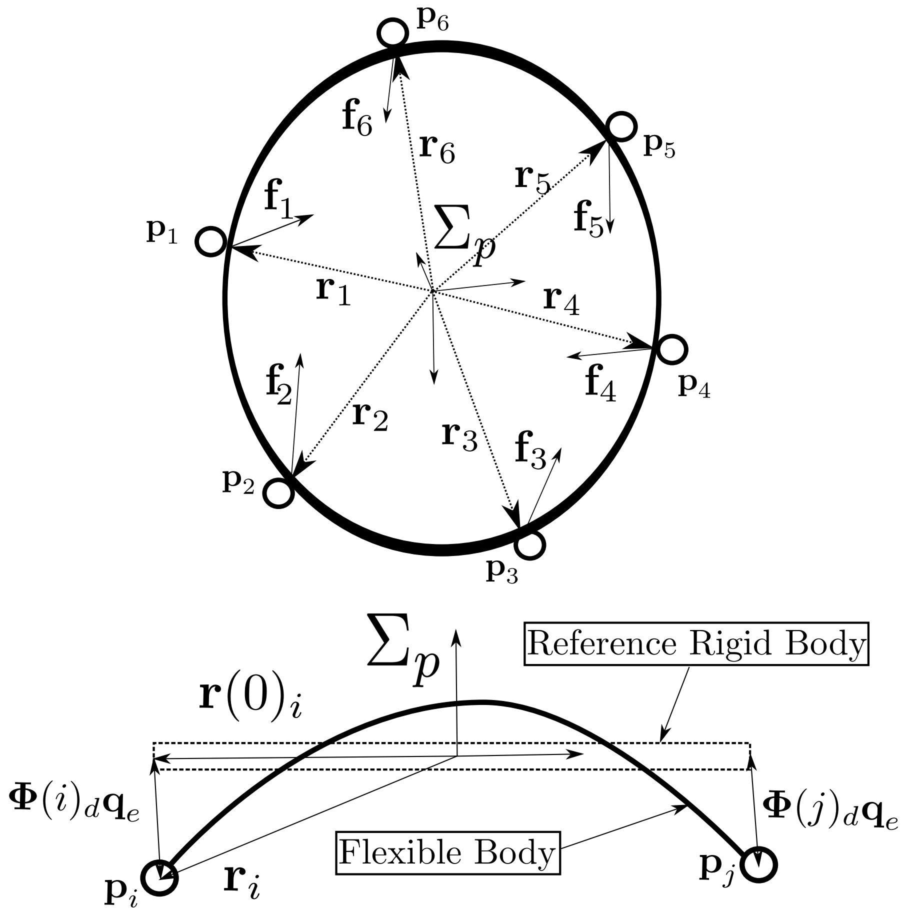

Figure 3.15: Flexible Platform, (Top) Flexible platform forces and attachment point vectors (Bottom) Representation of Free-Free Boundary conditions

Inputs:

$\dot{\mathbf{V}}_p$: The desired rigid body Cartesian velocity of the platform.

$({}^0\mathbf{T}_p, \mathbf{V}_p, \mathbf{q}_e, \dot{\mathbf{q}}_e)$: The state of the robot i.e. the position and velocity of the rigid and the generalized elastic variables respectively.

The joint positions and velocities are obtained from the platform variables using (3.49) and (3.101), followed by the inverse geometric and kinematic model of each leg, respectively. The generalized elastic variables are obtained by integration.

Outputs:

$\mathbf{\Gamma}$: The vector of motor torques is obtained by solving (3.83), by using the Jacobian matrix of the system, $\mathbf{J}_{sys}^T$, after first obtaining the total inertia matrix from (3.84) and the total Coriolis, centrifugal and gravity torques from (3.86).

$\ddot{\mathbf{q}}_e$: The generalized elastic accelerations of the platform, can be obtained using (3.81).

Direct Dynamic Problem

The direct dynamic model of the robot gives the platform accelerations and the accelerations of the generalized elastic coordinates as a function of the input torque of the motorized joints and the state of the robot, which consists of the positions and velocities of the rigid and elastic variables. The primary use of the direct dynamic model is the simulation of robotic systems.

Inputs:

$\mathbf{\Gamma}$: The vector of motor torques.

$({}^0\mathbf{T}_p, \mathbf{V}_p, \mathbf{q}_e, \dot{\mathbf{q}}_e)$: The state of the robot i.e. the position and velocity of the rigid and the generalized elastic variables respectively.

Outputs:

$\dot{\mathbf{V}}_p$: The rigid body Cartesian acceleration of the platform. It can be obtained by (3.83) after inverting the inertia matrix, $\mathbf{A}$, of the system. The inertia matrix, the vector containing the total Coriolis, centrifugal and gravity torques and the system Jacobian matrix are obtained in the same manner as in the inverse dynamic problem. Finally the rigid body velocity, $\mathbf{V}_p$, and the pose of the platform, ${}^0\mathbf{T}_p$, are obtained by integrating the rigid body Cartesian acceleration.

$\ddot{\mathbf{q}}_e$: The generalized elastic accelerations of the platform, can be obtained using (3.81).

3.4.5 Model Validation

In order to validate the algorithm a simulation is carried out with a Gough-Stewart manipulator, modeled using MSC ADAMS and MSC Nastran as shown in Fig.3.14. The finite element model of the flexible platform is integrated into the rigid body model of the six legs.

The proposed dynamic models are validated using the procedure shown in Fig.3.16. The MSC ADAMS model follows a spatial point to point trajectory. The measured values are compared with those calculated by dynamic models as described in Sections 3.4.4 and 3.4.4 and previously carried out in Section 3.3.5.

Legs Model

The dynamic parameters of each leg are identical and are given in Table 3.7. The inertial parameters that are set to zero, either have no effect on the model or do not exist due to the symmetry of the links.

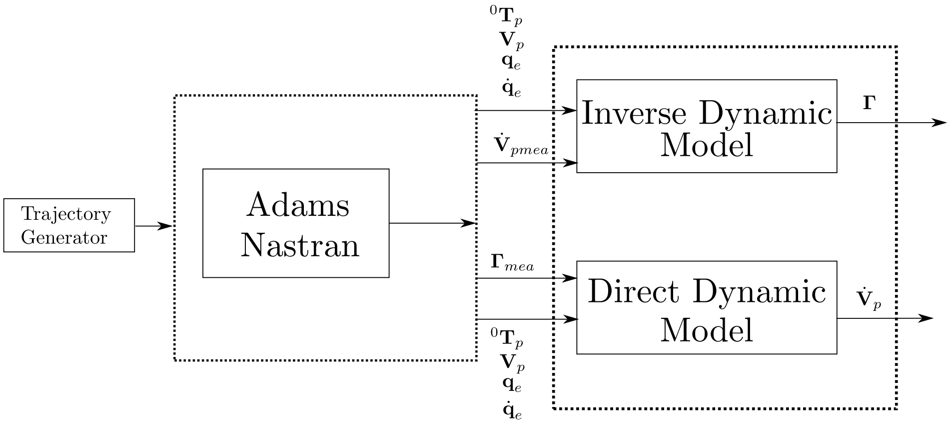

Figure 3.16: Simulation setup for Model validation

Finally, Table 3.8 gives the location of the base frame j, denoted as $\mathbf{b}_j$, for $j = 1 \dots 6$ in the world frame. The components along z, $\mathbf{b}_j(z) = 0$, for all legs. Furthermore, in this table, the initial joint values of the robot are given.

Table 3.7: Base Inertial parameters of Legs

Link	1	2	3
XX (kg m^2)	0	4.0	1.0
YY (kg m^2)	0	1.0	1.0
ZZ (kg m^2)	5.0	5.0	0.0234
XY (kg m^2)	0	0.0	0
XZ (kg m^2)	0	0	0
YZ (kg m^2)	0	0	0
MX (kg m)	0	0.0	0
MY (kg m)	0	0.0	0
MZ (kg m)	0	0	0.0
M (kg)	3.0	15.0	5.0

Platform Model

The flexible platform is a regular hexagon that is inscribed in a circle of radius 0.5m. The plate has a Young's Modulus of 2×10^{10}Nm, a uniform thickness 15×10^{-3}m, and a density of 7.5×10^3 kg/m^3. The total mass of the flexible platform is 73.069kg. A modal analysis is carried out on the hexagonal plate using MSC Nastran ©. The plate is represented by 1261 nodes.

Table 3.8: Coordinates of Legs in the world frame

leg	1	2	3	4	5	6
$\mathbf{b}_j(x)(m)$	0.1	0.3	1.3	1.0	-0.2	-0.7
$\mathbf{b}_j(y)(m)$	-0.2	-0.2	0.4	1.0	0.9	0.5
$\mathbf{q}_{1j}(rad)$	-1.671	-1.373	-2.016	-1.958	-1.656	-1.936
$\mathbf{q}_{2j}(rad)$	1.767	1.764	1.814	1.859	1.389	1.345
$\mathbf{q}_{3j}(m)$	1.025	1.039	1.142	1.126	1.0203	1.099

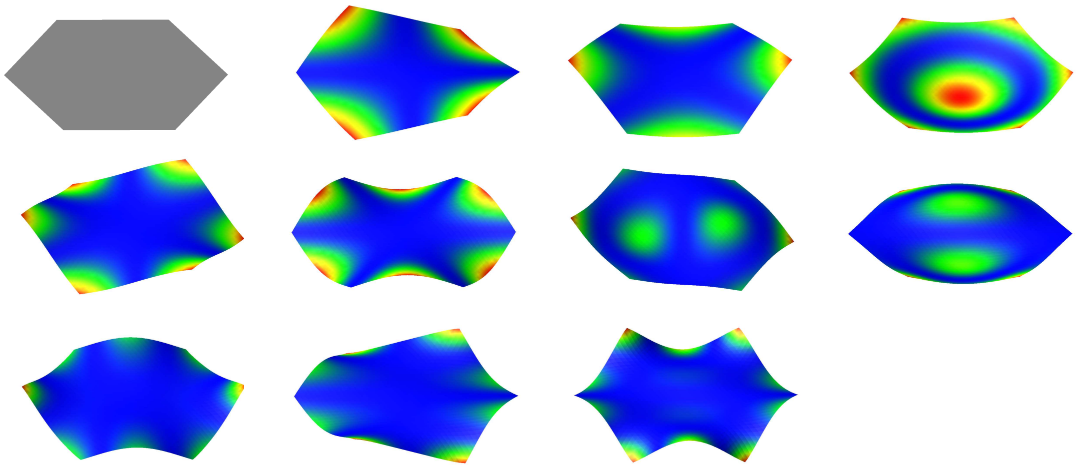

Figure 3.17: Mode shapes of Flexible Platform
From top to bottom left to right, undeformed platform followed by deformation associated with modes 7-16

The first forty non-rigid body modes are used to represent the flexibility of the platform. Table 3.9 contains the natural frequency, generalized stiffness and generalized damping of each mode. A visualization of the mode shapes $7-16$ is given Fig.3.17. The figures show the natural deformation of the mode shape alongside the undeformed platform (in gray).

Table 3.9: Modal properties of Flexible Plate

Mode No.	Natural frequency (rad/s)	Generalized Stiffness	Damping ratio
7	188.7473201	35625.55084	0.5
8	188.7475262	35625.62866	0.5
9	315.7168971	99677.1591	0.5
10	401.0187086	160816.0046	0.5
11	474.7467769	225384.5021	0.5
12	709.5586013	503473.4087	0.5
13	709.5813831	503505.7393	0.5
14	828.2868067	686059.0342	0.5
15	828.3061424	686091.0655	0.5
16	1310.580705	1717621.785	0.5
17	1310.592618	1717653.011	0.5
18	1373.566586	1886685.167	0.5
19	1859.199326	3456622.135	0.5
20	3467.281813	12022043.17	0.5
21	3467.39848	12022852.22	0.5
22	3900.053675	15210418.67	0.5
23	4644.289592	21569425.82	0.5
24	4644.467184	21571075.43	0.5
25	6702.730743	44926599.41	0.5
26	9119.086385	83157736.5	0.5
27	9119.182958	83159497.81	0.5
28	9650.782311	93137599.21	0.5
29	9651.05866	93142933.26	0.5
30	10180.45378	103641639.2	0.5
31	10181.08775	103654547.7	0.5
32	11442.96902	130941540	0.5
33	11443.36412	130950582.4	0.5
34	12557.52289	157691381	0.5
35	12772.5502	163138038.6	0.5
36	14623.90771	213858676.7	0.5
37	14688.31576	215746619.8	0.5
38	14688.46408	215750977.1	0.5
39	15403.11582	237255976.9	0.5
40	16393.83391	268757790.1	0.5

Results

The inverse dynamic model, denoted IDM, calculates the joint torques from the desired acceleration of the platform and the robot's Cartesian state variables. A comparison between the measured joint torque and the predicted joint torque is shown Fig.3.18. A strong correlation can be seen between the measured torques and the calculated torques with an error of less than five newtons.

The direct dynamic model, denoted DDM, calculates the rigid linear and angular acceleration of the platform from the input torque and the robot's state variables. A comparison between the measured acceleration and the predicted acceleration is shown Fig.3.19. This graph shows very little error between system modeled in MSC ADAMS and the direct dynamic model given in Section 3.4.4.

Finally, the generalized elastic variables are obtained by (3.81) using the joint torques calculated from the IDM. This vector has thirty-four components, however for simplicity, the results for the modes that contribute most to the object's strain energy, $i = 7, 8, 9, 14$ are shown. The graphs show a good correlation between the predicted values, denoted $\mathbf{q}_{ei}^{cal}$, and the measured values, denoted $\mathbf{q}_{ei}^{mea}$, in spite of the large discontinuities present in the system. The limits of precision can be also be seen in this graph where for small values of the $\mathbf{q}_{ei}^{mea}$ there is a large error for the predicted values. This error is a result of an accumulation of errors from the calculation of the joint torque as during the same period, there is also a significant error for the predicted joint torques as shown in Fig.3.18.

The sources of error in the graphs may originate for numerical reasons. In addition to this there may some discrepancy between the flexible dynamic model calculated by Nastran and that which has been calculated using the procedure outlined in Section 3.4.3. Finally, it should be noted that the large values seen at the beginning of the trajectory, which are instantaneous, are due to small errors in the initial position that also effects the overall accuracy of the model.

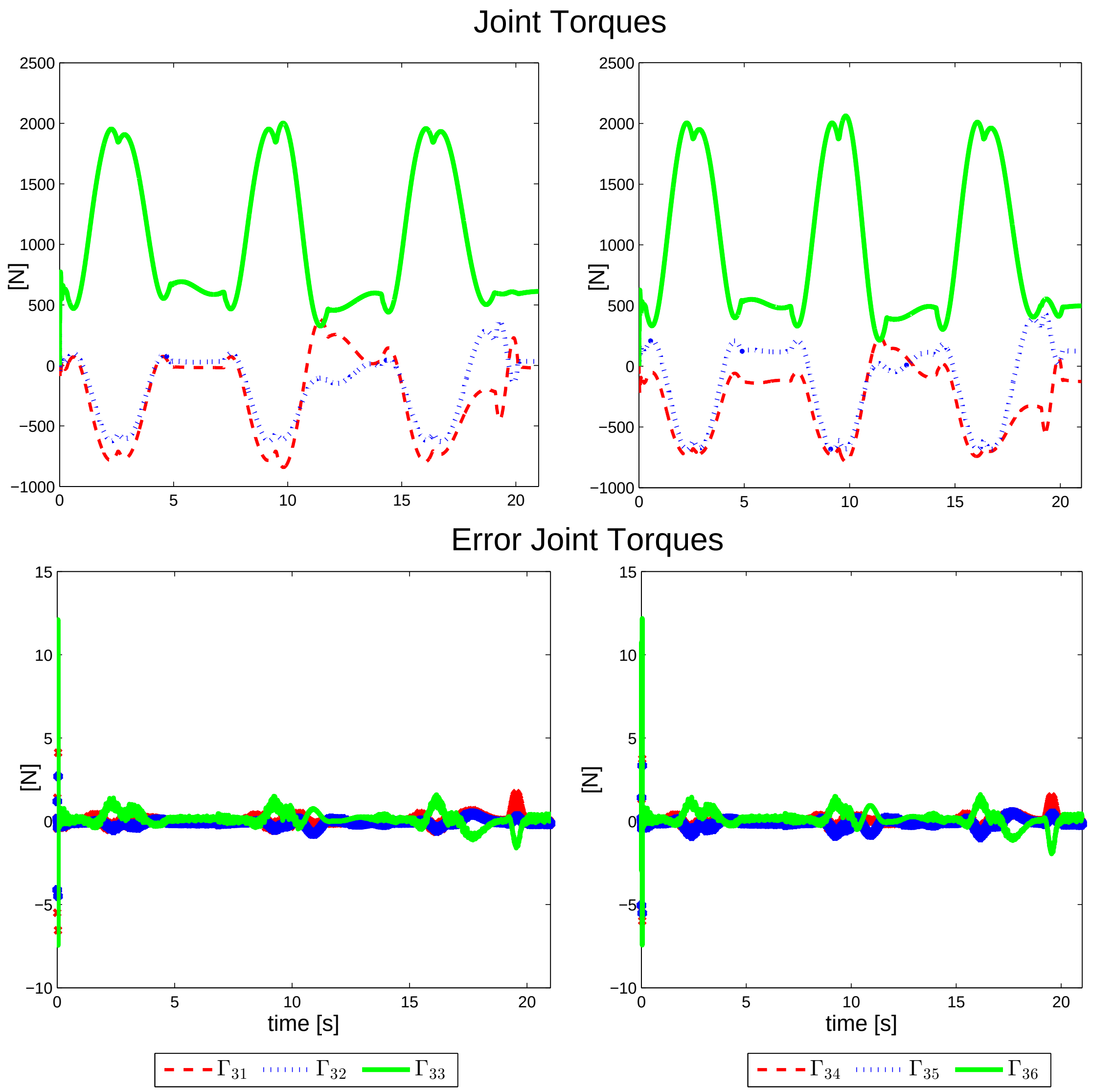

Figure 3.18: Actuated prismatic joint forces. (Top) Measured Torques from ADAMS simulation , (Bottom) The error between measured torques and torques calculated from the Inverse Dynamic Model

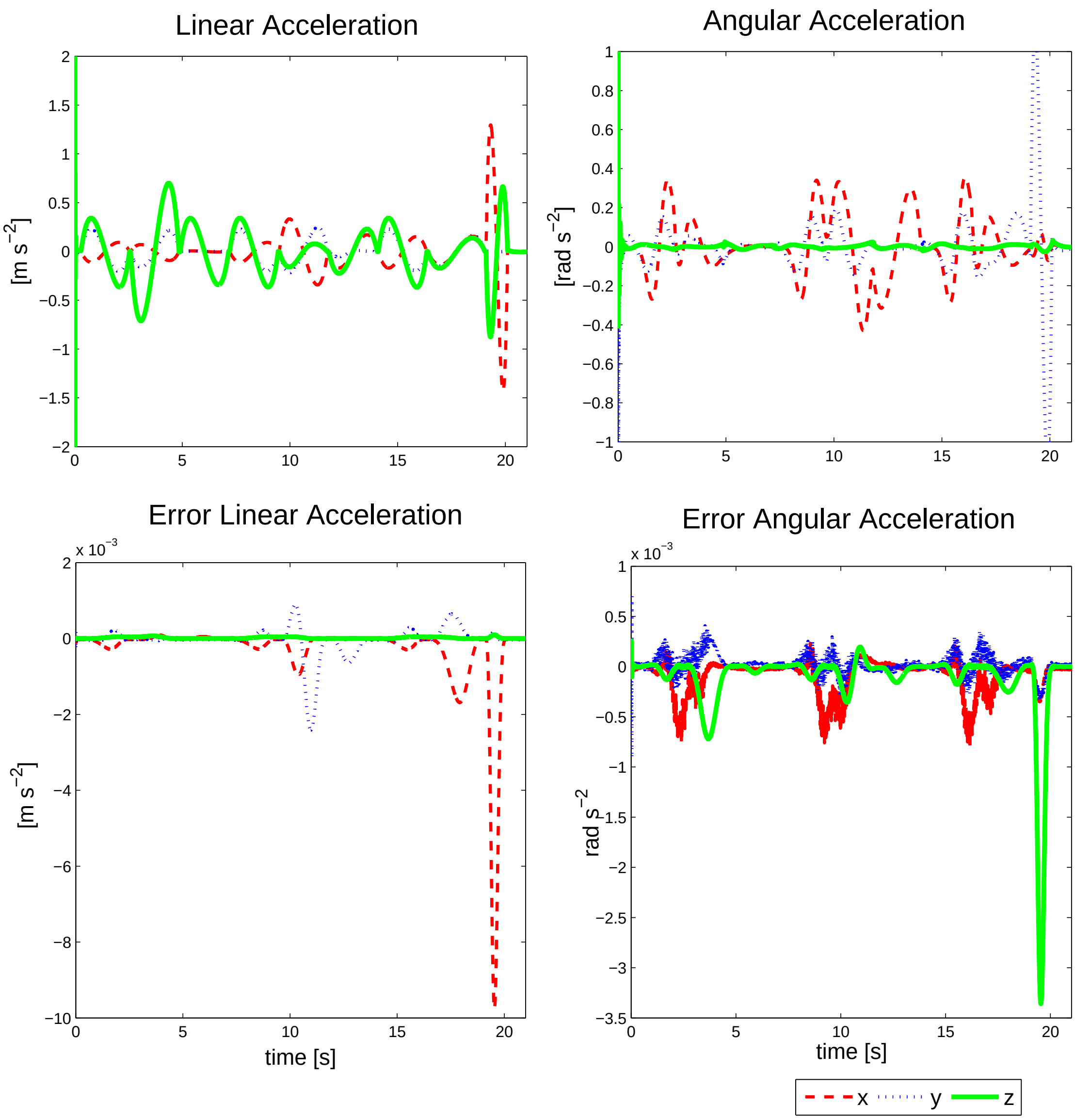

Figure 3.19: (Top) Measured acceleration of Platform Frame, difference between measured and calculated acceleration from DDM(bottom)

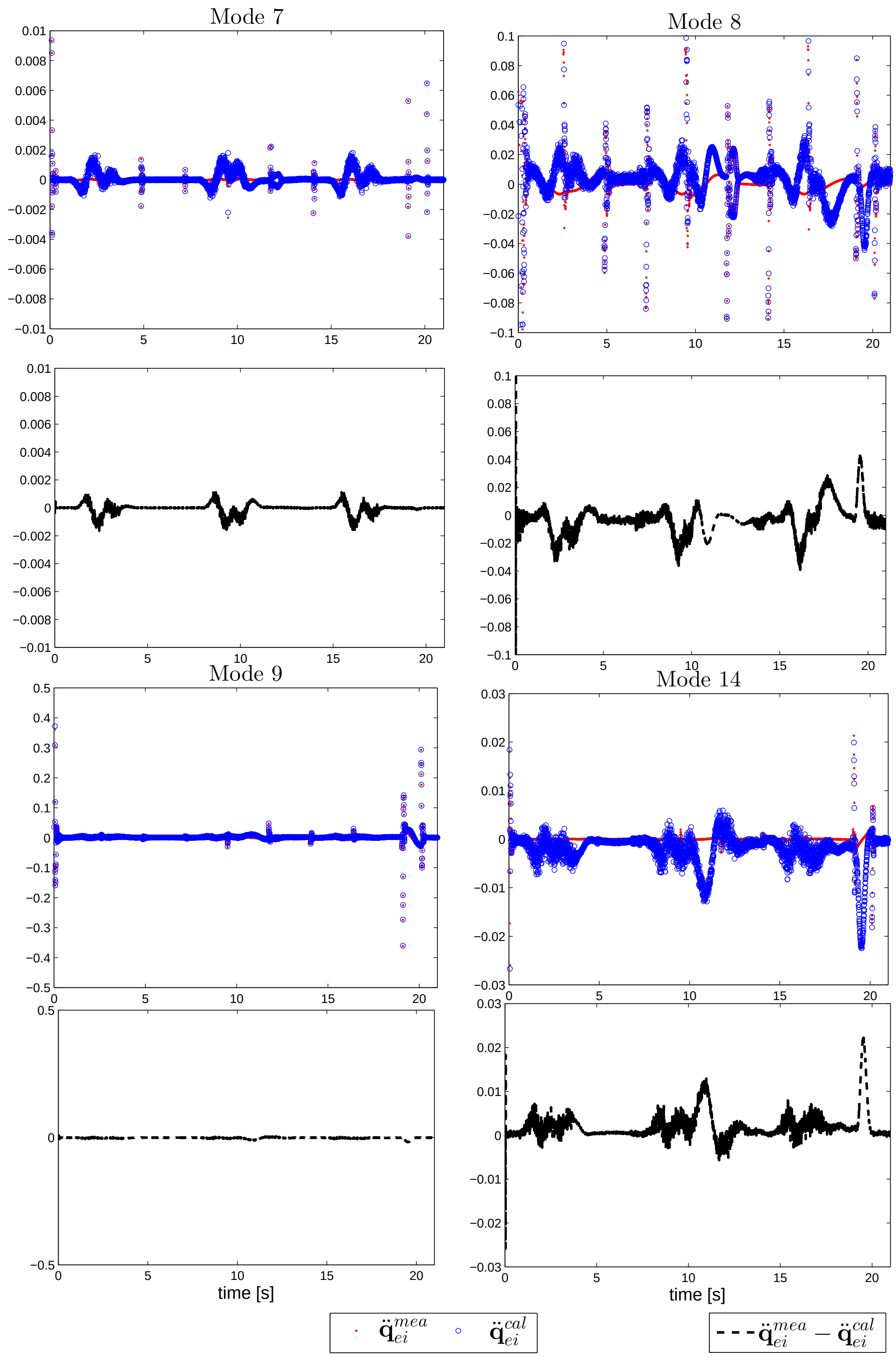

Figure 3.20: From each mode the top graph shows a comparison between the predicted values and the measured values. The bottom graph shows the error.

3.5 Conclusion

This chapter focuses on the cooperative manipulation of flexible objects. This is a broad subject that ranges from the dual arm manipulation of common objects to robots with flexible components. A general definition of a deformable object is given. This definition is based on the relative behavior between any two points of the object. The main methods of modeling deformable bodies i.e. lumped parameter modeling, modal analysis and finite element analysis are described. Furthermore examples of the use of these techniques in multi-body systems are given. The generalized Newton-Euler formulation, a complete modeling strategy of manipulators that contain flexible components is described. This formulation is used to obtain the dynamic model of robots with flexible links but can also be extended to robots with flexible payloads. Finally the robotic control of deformable objects is described. This consists of two or more manipulators grasping a common deformable object. The tasks associated with this system are divided into trajectory following tasks and shape control. Examples of trajectory following tasks, where the system must transport the deformable object in space, and shape control, where the system must apply forces to modify the form of the object, are outlined and explained.

This contribution of this chapter is a general strategy for modeling closed chain manipulators handling flexible objects. The robot is decomposed into two sub-systems, the first consisting of the rigid arms and base, the second of the flexible object. The effects of the flexible sub-system on the rigid sub-system and vice versa are obtained by calculating the reaction forces at the grasp locations. A dynamic modeling equation is derived in terms of the Cartesian accelerations of the object. A closed form solution, relating the actuated joint torques to the Cartesian accelerations, is given in terms of the elastic and inertial parameters of the robot. The resulting equations give an insight into the object type. A numerical comparison with a commercially available dynamic simulator, MSC ADAMS, is carried to validate the model. By varying the degree of flexibility, different objects can be tested. The solution and validation of three case studies is presented. Each case study focuses on a different type of object: rigid, flexible and articulated. The results validate the modeling strategy for the multiple object types. Finally, a method is defined that can classify the object. This method is based on the structure of the matrix $\mathbf{W}_s$ which relates the motor torques to the Cartesian variables.

The second contribution of this chapter is the application of the general dynamic modeling strategy to a well known parallel manipulator, the Gough-Stewart platform. By deriving $\mathbf{W}_s$, the object type and thus solution type is identified. A numerical simulation of the Gough-Stewart manipulator is created, where the flexibility of the platform is represented by 40 modes. The simulation compares the output of the inverse and direct dynamic model with a simulated Gough-Stewart manipulator. The results show a good correlation between the two systems for both models. This demonstrates the applicability of the general modeling strategy to a wide range of closed chain systems.

The work presented in this chapter has led to the publications of one journal paper [LKM14a] and one conference paper [LKM15].

4

Force/Vision Control of a Meat Cutting Robotic Cell

4.1 Introduction

Frequently, manufacturing tasks related to deformable soft materials require the cutting or separation and of the target objects. Robotic cutting of soft material is an area with many potential applications. Cutting differs from contour following tasks since during the cut, in order to separate the object the tool must necessarily pass through the contour. Furthermore, the controlled force is unknown a priori and is resistive to motion rather than orthogonal. Therefore, a more sophisticated control strategy must be undertaken.

In contrast to classical manipulation tasks that use modeling strategies, the separation not only requires a object model but also an update of this model as the cutting progresses. Therefore using an off-line computed model no longer suffices. An example of the use of deformable object models for separation tasks is found in surgical applications, where the model is used to construct a simulator for training purposes. The simulator can be used to predict required input forces that deform the object in the desired way. Conversely, the object model can calculate the deformation in response to the tool interaction. It can be seen that the accuracy is largely dependent on computation time. Hence using highly accurate models may be unfeasible for control applications.

Alternatively, instead of using a complex object model, the robot can be equipped with exteroceptive sensors allowing an on-line modification of a trajectory to compensate for object behavior. Force and vision sensors allow robots to interact with a dynamic environment while executing complex tasks.

This chapter focuses on the robotic cutting of materials, in particular the focus is on the separation of soft and deformable objects by multiple robots. This chapter, excluding the introduction, is divided into four sections, the state of the art given in Section 4.2, the contributions of this thesis in this field given in Section 4.3 and in Section 4.4, finally the conclusions are given in Section 4.5. The outline of each section is given in the following.

State of the Art

In Section 4.2, the principal methods of modeling object separation, generally regarding surgical applications, are described. The section is organized as follows. In Section 4.2.1, the state of the art regarding visual servoing is given. This include the camera configurations and a comparison between image and position based visual servoing. Section 4.2.2 shows how vision sensor can be combined with force sensors for on-line modification of a robot's trajectory. The control schemes regarding these sensors are shown. In Section 4.2.3, the principal means of modeling, simulating and controlling the separation of soft materials is described. The majority of these schemes are proposed for surgical applications. In Section 4.2.4, a review of robotic cutting systems is outlined where a description of the mathematical modeling of the cut is given. The cutting formulations allow the calculation of the cutting force required for a particular object's material properties.

Contributions

The contributions in this chapter, found in Section 4.3 and Section 4.4, concern the robotic separation of soft materials. Two cases are presented, the simulation of the multi-arm system proposed by the ARMS project to separate beef muscles and an experimental validation of a set of force vision controllers. In the following, the outline of each section is given.

In Section 4.3, the modeling, simulation and control of a meat cutting robotic cell is described. A multi-arm system is proposed to complete the separation of a deformable object. The cell comprises three robots referred to as the cutting, pulling and vision robot respectively. Sections 4.3.1, 4.3.2, 4.3.3 and 4.3.4 concern the construction of the simulator and the deformable object. The deformable object represents a beef shoulder muscle in a meat cutting robotic cell. The modeling of the beef shoulder muscle and its integration into the simulator environment is outlined. The modeling strategy takes into account the deformable nature of the object and its interaction with the cutting tool to establish a realistic cell behavior. Furthermore the construction of a set of visual primitives used to obtain the cutting trajectory is given. In Section 4.3.5 a control scheme, using external vision and force sensors, is proposed for the separation task. In order to complete the separation task each robot is assigned a particular function, allowing the system to localize the desired cutting trajectory, move the cutting tool along this trajectory and ensure a gradual, controlled opening of the cutting valley. The separation is performed by repeating a series of cuts, called passages, along an unknown trajectory. The proposed control scheme is validated using the simulator environment and the results of the control scheme are given in Section 4.3.6.

In Section 4.4, an experimental validation of a force/vision control strategy proposed to separate soft deformable materials using cooperative robots is carried out. The force feedback of the tool is used to modify the cutting trajectory to prevent global deformation of the material. In Section 4.4.1, the robotic cell, consisting of two robots, a force sensor, a cutting tool and a camera fixed to this cutting tool, used to complete the separation is described. In Section 4.4.2, a new force controller that addresses the problem of robotic cutting of deformable objects is proposed. The performance of the force controller is evaluated and compared with respect to a baseline position controller in Section 4.4.3. Two different visual servoing schemes are tested experimentally. A PBVS controller that can reconstruct the planar position of a trajectory is described in Section 4.4.4. In Section 4.4.5 the results and discussion for the PBVS control scheme are given. In Section 4.4.6 the global control scheme based on IBVS is illustrated. In Section 4.4.7 the results for the second set of experiments are given. In both cases, the visual feedback modifies the robot motion in order to follow the flexible cutting trajectory while the force feedback ensures a controlled crack propagation, leading to a precise cut.

4.2 State of the Art: Robotic Separation of Soft Materials using Force/Vision control

4.2.1 Visual Servoing

Visual servoing consists of using a visual feedback system in order to control the robot's end effector. It has been shown to provide a flexible, stable and reactive mode of control for robot manipulators [HHC96]. Visual servoing spans many different disciplines including computer vision, image processing, kinematic modeling and non-linear control theory. In this section the primary focus is on the kinematic and control aspects of the schemes.

Vision data is acquired by one or more cameras either mounted on the robot or mounted on an external frame. The data is then processed and features pertaining to the chosen control scheme are extracted. The controlled features are denoted as s. The robot task is defined in terms of these features. An error term is formulated between the current features and a desired set:

$$\mathbf{e} = \mathbf{s} - \mathbf{s}^* \tag{4.1}$$

The choice of $\mathbf{s}$ is crucial and a diffeomorphism must exist between the feature error $\mathbf{e}$ and the task coordinates $\mathbf{x}$. If such a relationship exists, the feature is a time varying function of the relative pose between the camera and the object $\mathbf{x}$. This variation is given by:

$$\dot{\mathbf{s}} = \frac{\partial \mathbf{s}}{\partial \mathbf{x}} \frac{\partial \mathbf{x}}{\partial t} + \frac{\partial \mathbf{s}}{\partial t} = \mathbf{L}_s \, {}^c\mathbf{V}_c + \frac{\partial \mathbf{s}}{\partial t} \tag{4.2}$$

In the case of a fixed feature, $\dfrac{\partial \mathbf{s}}{\partial t} = 0$, therefore:

$$\dot{\mathbf{s}} = \mathbf{L}_s^c \mathbf{V}_c \tag{4.3}$$

${}^c\mathbf{V}_c$ is kinematic screw of the camera frame composed of $\mathbf{v}_c$, the instantaneous linear velocity, and $\boldsymbol{\omega}_c$ the instantaneous angular velocity. Equation (4.3) relates the camera velocity to the change in the controlled features by $\mathbf{L}_s$ which is known as the interaction matrix. Depending on the choice of $\mathbf{s}$, $\mathbf{L}_s$ may not be obtained exactly at each instant meaning an estimation of this matrix, denoted as $\hat{\mathbf{L}}_s$, must be used. There are three different approaches to estimating the interaction matrix [CH08]:

Interaction matrix at each iteration At each iteration the interaction matrix is estimated i.e. $\hat{\mathbf{L}}^+ = \mathbf{L}_s^+$. A perfect estimation will create a straight line trajectory in the controlled space. However there is a chance that the interaction matrix becomes singular as it passes through certain configurations. Furthermore the image noise directly affects the calculation of the matrix.

Interaction matrix at equilibrium The interaction matrix is constant and takes the value at the desired location: $\hat{\mathbf{L}}^+ = \mathbf{L}^+_{\mathbf{s}=\mathbf{s}^*}$. This method ensures that the matrix is always full rank and undisturbed by noise in the image data. However the use of this matrix is only valid in a localized zone around the desired configuration.

Half some of interaction matrices By combining the above approaches [Mal04], such that

$$\hat{\mathbf{L}}^+ = \frac{1}{2} \left(\mathbf{L}^+_{\mathbf{s}=\mathbf{s}^*} + \mathbf{L}^+_{\mathbf{s}} \right)$$

a good practical performance is obtained and a smooth trajectory is found both in image in in Cartesian space.

Typically an exponential decrease of the error is desirable ($\dot{\mathbf{e}} = -\lambda \, \mathbf{e}$). By differentiating equation (4.1) and combining with equation (4.3):

$$\dot{\mathbf{e}} = \mathbf{L}_s \, {}^c\mathbf{V}_c \tag{4.4}$$

$${}^c\mathbf{V}_c = -\lambda \mathbf{L}_s^+ (\mathbf{s} - \mathbf{s}^*) \tag{4.5}$$

In order to create more viable control variables, first the camera velocity is related to the end effector velocity by the twist transformation matrix ${}^{e}\mathbf{S}_c$. Finally the end effector velocity can be related to the joint velocity by means of the robot kinematic matrix $\mathbf{J}$ (this matrix is often referred to as the kinematic Jacobian matrix in robotics) producing a control law in joint space as shown in equation (4.7):

$$ {}^{e}\mathbf{V}_e = {}^{e}\mathbf{S}_c\, {}^{c}\mathbf{V} \tag{4.6} $$

$$ \dot{\mathrm{q}} = \mathbf{J}^{-1}\, {}^{e}\mathbf{V}_e = -\lambda \mathbf{J}^{-1}\, {}^{e}\mathbf{S}_c\, \mathbf{L}_s^{+}(\mathbf{s} - \mathbf{s}^{*}) \tag{4.7} $$

Camera Configuration

In visual servoing typically two camera configurations are used [HHC96]. The first configuration is called eye-in-hand, illustrated in Fig. 4.1. In this configuration the camera is attached to the end effector of the robot. A geometric transformation matrix between the camera frame and the tool frame, denoted as ${}^{E}\mathbf{T}_c$, is defined. This matrix is constant throughout the trajectory and can be used to easily calculate the end effector velocity from the camera velocity. The perceived motion of the target object is due both to self motion and to the motion of the end effector.

The second configuration is called eye-to-hand illustrated in Fig. 4.2. In this scheme the camera is fixed somewhere in the workspace independent of the robot end effector motion. Thus the target object motion is independent of the end effector motion.

When a camera is introduced into the control scheme, steps must be taken to obtain its intrinsic and extrinsic parameters. The intrinsic parameters give the relationship between a position of a point in image space, denoted as (u_i, v_i), and its position in the normalized or perspective plane denoted as $({}^{c}x_{ni}, {}^{c}y_{ni})$. For a simple pinhole camera without distortion, the linear relationship is given in (4.8). The matrix $\mathbf{C}_c$ is the matrix of collineation parameters, where (u_0, v_0) are the coordinates of the principle point and f represents the focal length. The position of the point in the camera frame can be fully obtained by (4.9) if the distance to the object from the camera along the optical axis, denoted as ${}^{c}z_i$, is known.

$$ \begin{bmatrix} {}^{c}x_{ni} \\ {}^{c}y_{ni} \\ 1 \end{bmatrix} = \mathbf{C}_c \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} \tag{4.8} $$

$$ {}^{c}z_i \begin{bmatrix} {}^{c}x_{ni} \\ {}^{c}y_{ni} \\ 1 \end{bmatrix} = \begin{bmatrix} {}^{c}x_i \\ {}^{c}y_i \\ {}^{c}z_i \end{bmatrix} \tag{4.9} $$

where

$$ \mathbf{C}_c = \begin{bmatrix} f_u & 0 & u_0 \\ 0 & f_v & v_0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.10} $$

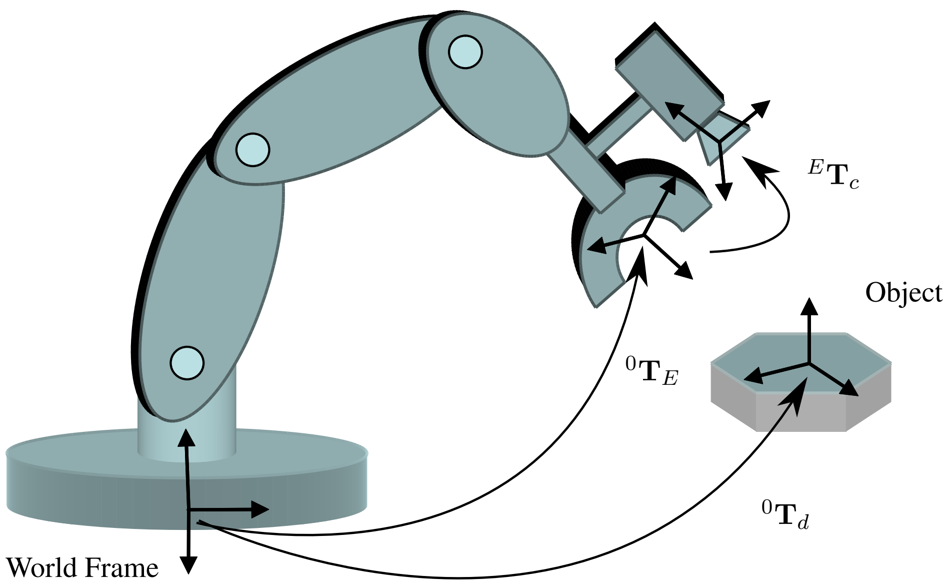

Figure 4.1: Eye in hand camera configuration

The extrinsic parameters give the location of the camera and can be obtained using visual markers [Tsa87]. This consist of finding a fixed transformation matrix between the camera frame and the frame of interest. Referring Fig.4.1 and Fig.4.2 it can be seen that for eye-in-hand systems the calibration procedure must obtain $^E\mathbf{T}_c$, whereas for eye-to-hand systems $^0\mathbf{T}_c$ must be found.

Image Primitives

The choice of image primitives is determined by various factors and has a critical effect on the behavior of the control system. The image primitive determine both the complexity of the control and the DOF that can be controlled.

There are three different ways the image primitives can be used in the control scheme. Firstly, by reconstructing the location of an object from the camera's image i.e. PBVS as outlined in Section 4.2.1. Secondly, for inducing camera motion such that the image converges to a desired image type i.e. IBVS as described in Section 4.2.1. Finally, by combining the above schemes HBVS schemes can be constructed.

A multitude of geometric features [HHC96, CRE93]have been successfully used in visual servoing including 3D points, 2D line segments, cylindrical coordinates of an image point and image moments [CH07].

Image moments provide a robust description of a diverse image objects and have been widely used in computer vision tasks [Hu62]. Recently image moments have been applied to visual servoing tasks. By careful combination of image moments, an

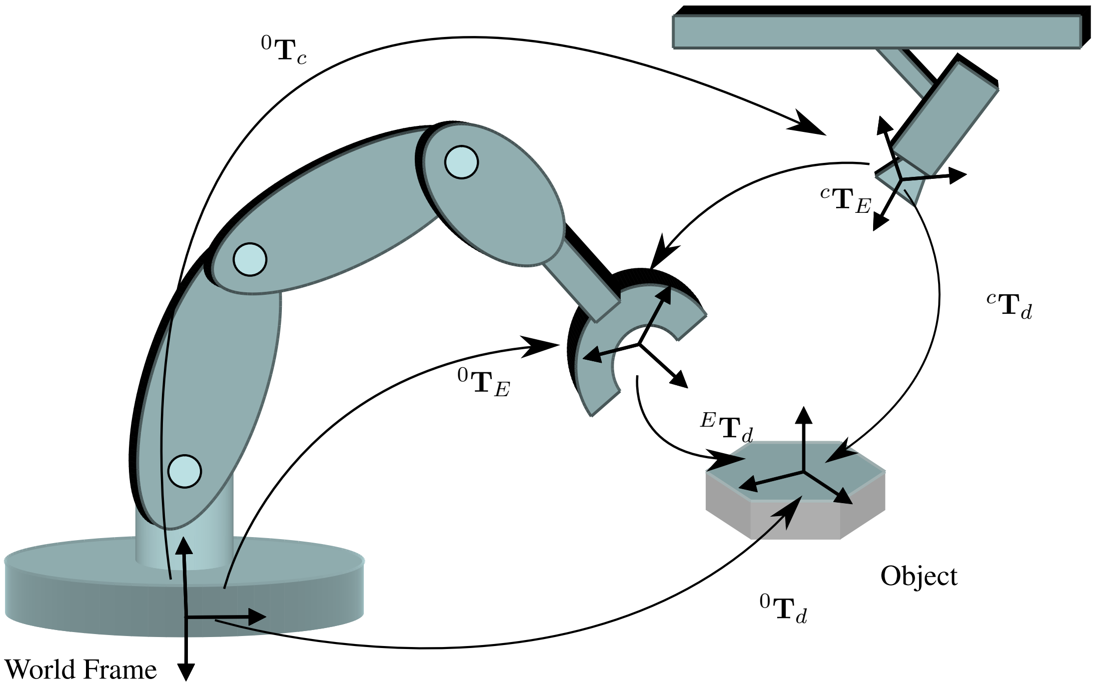

Figure 4.2: Eye to hand camera configuration

interaction matrix of maximal decoupled structure and low condition number can be built for a variety of complex images [Cha04, TC05].

Position Based Visual Servoing (PBVS)

In PBVS the visual data is used to reconstruct the 3D pose of an object of interest with respect to the camera. A typical PBVS scheme is shown in Fig.4.3. Examples of PBVS for robotic applications are given in [TMCG02, MG99].

For PBVS the state vector $\mathbf{s}$ is commonly defined as the parametrization of the camera pose with respect to a reconstructed 3D frame. In order to simplify matters, $\mathbf{s}^*$ can be chosen as zero.

For example, referring to Fig.4.1, by using some known information about the object, CAD data for instance, its 3D location with respect to the camera can be obtained i.e $^c\mathbf{T}_d$. The task can be defined so that in the desired configuration $^c\mathbf{T}_d = \mathbf{1}$. Six variables are required to fully parametrize a frame in space, thus this matrix can be represented in a 6×1 vector form as $\mathbf{s}$. One such choice, $\mathbf{s} = (^c\mathbf{p}_d, \theta\ ^c\mathbf{u}_d)$, is known to be asymptotically stable if the pose is perfectly estimated. This leads to a straight line trajectory for the camera *in the camera frame*, where $^c\mathbf{p}_d$ gives the position coordinates and $\theta\ ^c\mathbf{u}_d$ is the angle axis representation of rotation matrix $^c\mathbf{R}_d$.

Referring to (4.5), it can be seen that the interaction matrix must convert $\dot{\mathbf{s}}$ to the kinematic screw of the end effector. In this case, (4.5) is rewritten as follows:

$$ {}^e\mathbf{V}_e = {}^e\mathbf{S}_c \begin{bmatrix} -\mathbf{1} & {}^c\hat{\mathbf{p}}_d \\ 0 & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{1} & 0 \\ 0 & \mathbf{B}(\theta\, {}^d\mathbf{u}_c)) \end{bmatrix} \begin{bmatrix} {}^c\dot{\mathbf{p}}_d \\ \dot{\overbrace{\theta\, {}^c\mathbf{u}_d}} \end{bmatrix} \tag{4.11} $$

where

$$ \mathbf{L}_s^{+} = \begin{bmatrix} \mathbf{1} & -{}^c\hat{\mathbf{p}}_d \\ 0 & \mathbf{1} \end{bmatrix} \begin{bmatrix} \mathbf{1} & 0 \\ 0 & \mathbf{B}(\theta\, {}^d\mathbf{u}_c)) \end{bmatrix} \tag{4.12} $$

$\mathbf{B}(\theta\, {}^d\mathbf{u}_c))$ is required to convert $\dot{\overbrace{\theta\, {}^c\mathbf{u}_d}}$ to angular velocity. The parametrization has an effect on the resulting controller. Consider the same task but with a change in parametrization.

$$ {}^e\mathbf{V}_e = {}^e\mathbf{S}_c \begin{bmatrix} {}^c\mathbf{R}_d & 0 \\ 0 & {}^c\mathbf{R}_d\mathbf{B}(\theta\, {}^d\mathbf{u}_c)) \end{bmatrix} \begin{bmatrix} {}^d\dot{\mathbf{p}}_c \\ \dot{\overbrace{\theta\, {}^d\mathbf{u}_c}} \end{bmatrix} \tag{4.13} $$

where

$$ \mathbf{L}_s^{+} = \begin{bmatrix} {}^c\mathbf{R}_d & 0 \\ 0 & {}^c\mathbf{R}_d\mathbf{B}(\theta\, {}^d\mathbf{u}_c)) \end{bmatrix} \tag{4.14} $$

In this case, the interaction matrix is a block diagonal matrix and therefore exhibits desirable decoupling properties. From (4.11) and 4.13 it is clear that any errors due to poor calibration of the intrinsic parameters, or external parameters lead to an error in the calculated velocity. This sensitivity to calibration is the primary drawback of PBVS. Furthermore the 3D model of the object is required. Finally as the control is not carried out in the image space the image may leave the field of view during the task execution [CH01].

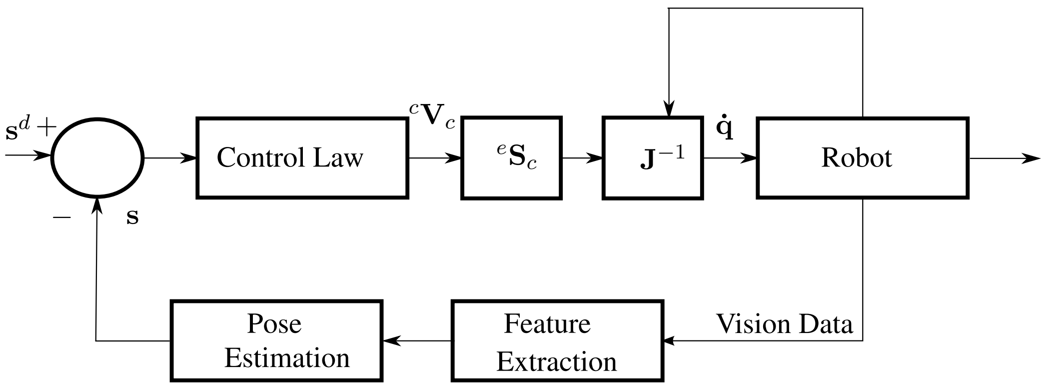

Figure 4.3: Position Based Visual Servoing Scheme

Image Based Visual Servoing (IBVS)

Image based visual servoing or 2D visual servoing extracts geometric features from the image and uses them directly in the control loop. The idea is to make s, the current state of the chosen image feature, converge to s* a desired pose that has been learned offline. Using IBVS, s will converge to s* along a straight line trajectory in image space. A typical IBVS scheme is shown in Fig. 4.4.

IBVS does not reconstruct the Cartesian pose at each iteration, this leads to an almost model free control loop thus eliminating a source of error due to the calibration of the system. Moreover, by controlling the system in the image space, the IBVS system can ensure the image always remains in the field of view [Cha98, CH01]. Furthermore, by using information about the robots structure joint and torques limits can also be taken into account [SPDC06]. However, some knowledge of the 3D environment must be known, usually the depth of the image feature, defined in as 4.15 as ${}^c z_i$.

Suppose, the image feature is taken as an image point of coordinates $\mathbf{s} = [x_p \; y_p \; 1]$. Assuming a pinhole camera model, from (4.9) the relationship with the normalized coordinates, denoted as $\mathbf{p}_n$ and the 3D point coordinates denoted as ${}^c\mathbf{p}_i$ is given as:

$$\mathbf{s} = \mathbf{C}_c \mathbf{p}_n = \mathbf{C}_c \frac{{}^c\mathbf{p}_i}{{}^c z_i} \tag{4.15}$$

By calculating the time derivative of (4.15) and assuming a constant collineation matrix, (4.3) can be rewritten as:

$$\dot{\mathbf{s}} = \mathbf{C}_c \frac{\partial \mathbf{p_n}}{\partial {}^c\mathbf{p}_i} \begin{bmatrix} -\mathbf{I}_3 & {}^c\hat{\mathbf{p}}_i \end{bmatrix} {}^c\mathbf{V}_c = \mathbf{L}_s^c \mathbf{V}_c \tag{4.16}$$

$\mathbf{L}_s$ is calculated as :

$$\mathbf{L}_s = \begin{bmatrix} -\dfrac{f_u}{{}^c z_i} & 0 & \dfrac{x_p}{{}^c z_i} & \dfrac{x_p \, y_p}{f_v} & -f_u - \dfrac{x_p^2}{f_u} & y_p \, \dfrac{f_u}{f_v} \\ 0 & -\dfrac{f_v}{{}^c z_i} & \dfrac{y_p}{{}^c z_i} & f_v + \dfrac{y_p^2}{f_v} & -\dfrac{x_p \, y_p}{f_u} & -x_p \, \dfrac{f_v}{f_u} \end{bmatrix} \tag{4.17}$$

The following general observations for IBVS can be made from simply examining (4.17). Firstly some 3D knowledge is required, in this case the z coordinate of the point in the camera frame. In order to overcome this drawback, estimation [DLOG08] of the z is possible using a dynamic observer. Alternatively the desired value can be used. Secondly, analogous to controlling a manipulator in joint space, when the end effector is controlled through image points its Cartesian trajectory can be unpredictable and the robot could violate Cartesian constraints. This is clear from the fact that a pure x motion of an image point does not generate a pure x motion of the camera. Finally a major drawback is the presence of local minima and singularities in the chosen interaction matrix[Cha98]. A examination of equation (4.3) shows that if the interaction matrix is not full rank there exists a camera velocity ${}^{c}\mathbf{V}_c$ such that ${}^{c}\mathbf{V}_c \in \mathcal{N}(\mathbf{L_s})$, i.e. a non-zero camera velocity which produces a zero change in the image feature.

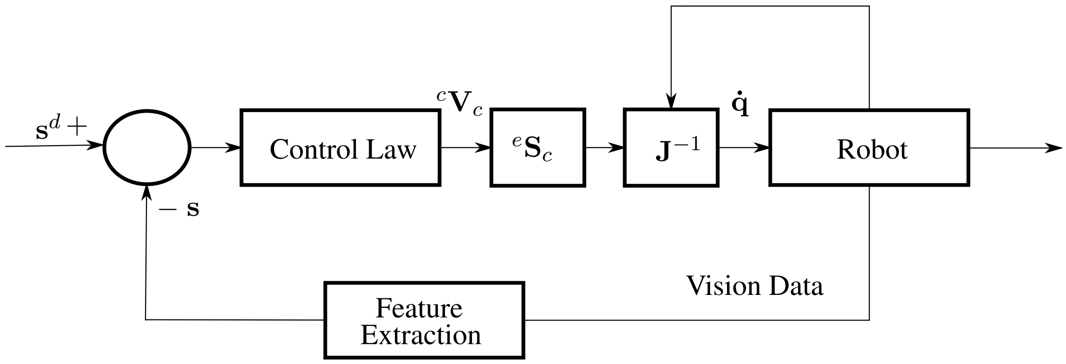

Figure 4.4: Image Based Visual Servoing Scheme

Hybrid Based Visual Servoing (HBVS)

The term hybrid visual servoing is used to encompass various advanced schemes that use a combination of PBVS and IBVS, in order exploit their respective advantages [AMAM05]. Generally HBVS uses a IBVS controller to determine certain velocities, normally the linear velocities, while using *epipolar geometry* to determine the angular velocities [MOP07].

For example, in [CH01] the control space is partitioned so that a separate controller is used for the translational, and rotational velocities along and around the Z axis. The aim is to avoid the problems associated with Z-axis rotations in IBVS. A representation is illustrated in Fig. 4.5. However since HBVS uses pose estimation, the scheme suffers from the same sensitivities as PBVS.

4.2.2 Force/Vision Control in Robotics

The use of exteroceptive sensors can reduce the need for a complex object model when performing tasks that require contact. Furthermore these sensors allows a greater degree of autonomy for the robot in a dynamic environment. In Chapter 2, Section 2.2 an outline of the fundamental schemes to manage force and position in terms of cooperative manipulators is given. Increasingly however, the position control is been executed by a vision controller. The vision controller can furnish more information about the environment and therefore cope with uncertainties. In this section, the application of these schemes in the context of visual servoing is illustrated. The types of force/vision control can been divided into three categories [NMK95]:

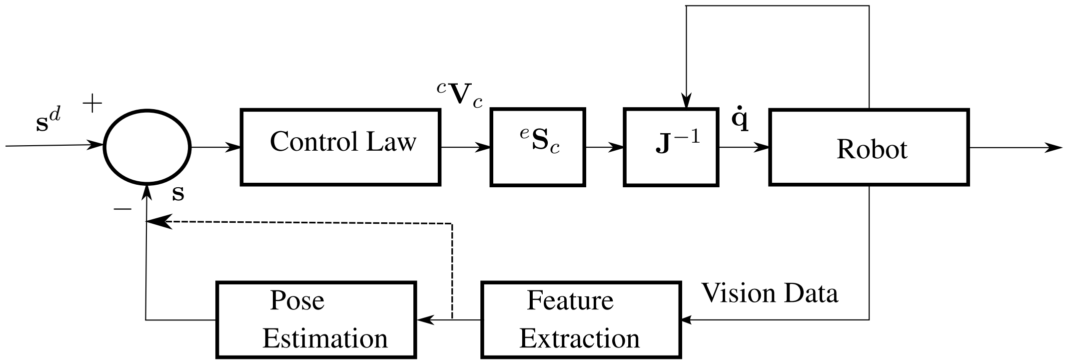

Figure 4.5: Hybrid 2D/3D Based Visual Servoing Scheme

Traded Control The same directions are controlled in both vision and force by simple alternation. This belongs to the class of *commutative controllers*. Generally when the manipulator is in open space it follows the chosen trajectory closely. As it approaches an object or obstacle it switches to a force controller in order to limit any contact forces.

Hybrid Control The control space is divided into two orthogonal spaces, vision and force. This belongs to the class of *partitioned controllers*.

Impedance/Shared Control Vision and force act on the same directions with some sense of priority. This belongs to the class of *hierarchical controllers*. This class of controllers is also known as admittance or external hybrid position force control.

From the above schemes, we focus on Hybrid control and Impedance control since these schemes allow a greater degree of autonomy than Traded control. Furthermore we divide schemes according to whether the control is conducted in the image space or in the Cartesian space.

Hybrid Force/Vision Control

Hybrid Force/Vision Control using PBVS The robot's possible motions are controlled either in force or position. The position controlled directions are defined and rebuilt from visual features. A typical Hybrid Force/Vision Controller is shown in Fig.4.6.

The most common task is defined as contour following. This consists in following a trajectory on a surface while the robot applies a force perpendicular to the contour. The definition of the surface can be estimated using force measurements[LLH06, XGXT00]. Firstly the tangent to the surface, denoted as $\mathbf{t}_s$, is estimated using the change in end effector motion $d\mathbf{x}$:

$$\mathbf{t}_s = d\mathbf{x}/\|d\mathbf{x}\| \tag{4.18}$$

Then the normal to the surface, $\mathbf{n}_s$, can be rebuilt using the force measurements and taking into account the friction which opposes the motion of the end effector

$$\mathbf{n}_s = \frac{\mathbf{f} - (\mathbf{f}\, d\mathbf{x}\mathbf{t}_s)}{\|\mathbf{f} - (\mathbf{f}\, d\mathbf{x}\mathbf{t}_s)\|} \tag{4.19}$$

The drawback of this method is the introduction of the inherent noise of the force measurements into the positioning task of the end effector. Alternatively, in order to overcome this disadvantage, a secondary external camera can be used to observe the angle between the tool and the surface [Cha06].

After the constraint surface has been reconstructed, the control is partitioned such that the position controller moves the robot tangential to the surface. Typically, the vision system locates the trajectory on the contour.

Likewise in [BDS99], a scheme is proposed that follows a contour using PBVS, while allowing a free rotation about the optical axis to position the camera. 1-DOF is force controlled the normal force, the rest of the directions are velocity controlled.

In [OJR04] a combination of hybrid vision force and impedance control is used. The manipulator reacts to the presence of a force by altering the trajectory in the *force* controlled directions. The vision controlled directions are independently controlled via a different feedback loop.

Hybrid Force/Vision Control using IBVS For this class of controllers the robot's possible motions are controlled either in force or image space. This means that the relationship between the tool motions and the image motion must be known.

In [NMK95] a perfect knowledge of the environment is assumed. This scheme uses static predefined orthogonal subspaces. In fact the space is partitioned into three subspaces: Image space, Force Space and End Effector velocity space.

In [HIA98], the partition of the space is completed on line to ensure no interference between the two control signals. If such interference occurs, force is automatically given priority by eliminating the force controlled direction from the image Jacobian matrix. This scheme is similar to those proposed in the [ZC04, PJ00]. The force measurements are used to obtain the normal to the contour using (4.19). This normal can then be used in conjunction with the image to partition the controlled space.

Impedance Force/Vision Control

Impedance Control using PBVS

A relationship is enforced between force and velocity along or about each axis of motion. The vision system is used to reconstruct the Cartesian pose of the object or a trajectory.

In [LSV06] a PBVS impedance control scheme is used to manage vision/force. In this scheme the visual features are used to rebuild the 3D Cartesian pose. The sensed

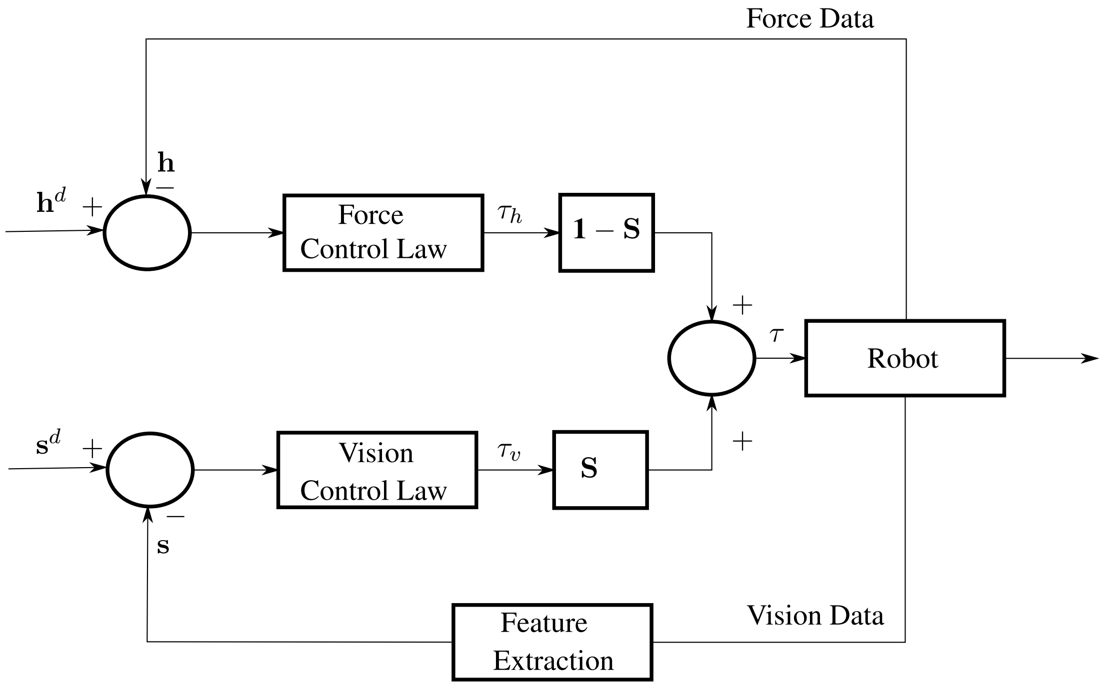

Figure 4.6: Classical Hybrid Vision Force Scheme

force $\mathbf{h}$ is then combined with the desired trajectory $\ddot{\mathbf{x}}_d, \dot{\mathbf{x}}_d, \mathbf{x}_d$, in order to create a new *compliant* reference trajectory $\ddot{\mathbf{x}}_c, \dot{\mathbf{x}}_c, \mathbf{x}_c$ via the classical impedance controller:

$$\mathbf{h} = \mathbf{M}(\ddot{\mathbf{x}}_d - \ddot{\mathbf{x}}_c) \ + \mathbf{B}(\dot{\mathbf{x}}_d - \dot{\mathbf{x}}_c) + \mathbf{K}(\mathbf{x}_d - \mathbf{x}_c) \tag{4.20}$$

Similarly, in [BGGCO+10], a trajectory is generated by vision. This trajectory is modified using the force measurements after it has been converted to position. The force is mainly used is ensure that any contact forces do not reach dangerous magnitudes.

Impedance Control using IBVS

Impedance control using IBVS, also known as External Hybrid Vision Force, implies that the impedance relationship is enforced directly between the image features and the sensed forces. The advantage of this scheme is the added robustness with respect to modeling errors. An illustration of this scheme is given in Fig.4.7 where $\mathbf{L}_x$ is defined such that $\dot{\mathbf{s}} = \mathbf{L}_x \, \delta\mathbf{x}$.

In [PMdPL07, MPM07] control schemes are proposed where the force control loop is closed around an internal vision control loop. The force changes the reference signal of the desired image feature by first converting the force to a Cartesian deviation and then using an interaction matrix to convert it to an image velocity. This is an extension of the position force scheme outlined in[PD93].

In [MM98] only the damping term is used for the force control law making it a *damping* or *accommodation* control scheme.

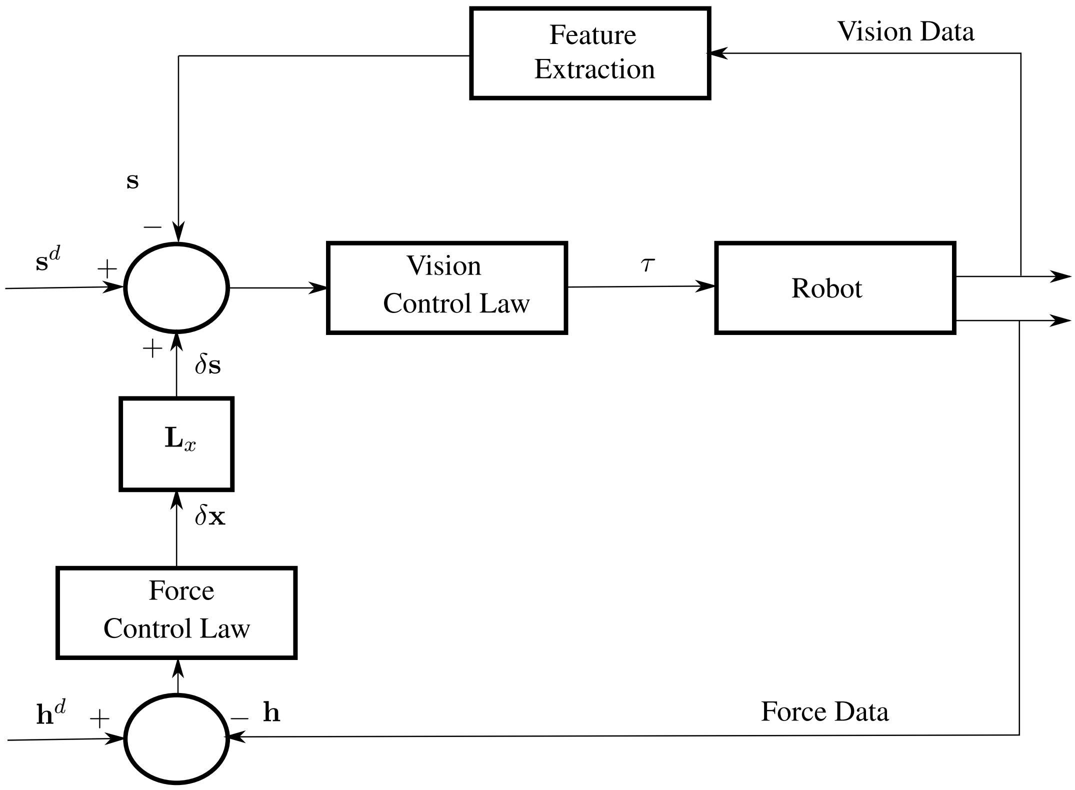

Figure 4.7: External Hybrid Vision Force Scheme

4.2.3 Simulating Deformable object separation

The simulation of cutting tasks has been investigated principally with respect to surgical applications [MRO08]. Generally, a virtual reality model is built for surgical training purposes. The objective of such a model is to replicate deformable body behavior, thus the cutting force, if considered, is used as a haptic output rather than controlled input that causes the rupture. In addition to this the training tool must be visually accurate to the surgeon which means the calculation of physical attributes, such as reaction force, is often sacrificed in order to maintain a realistic appearance. Nevertheless the extensive work on the simulation of surgical incision can exploited for a diversity of tasks. There are several different approaches to the modeling of soft tissue deformation, the choice of which is largely dependent on the final application and is a trade off between accuracy and calculation speed. The different approaches can be summarized in Table 4.1. It should be noted that vision based techniques can also be combined deformable object models to accurately track deformation in real time for example [RPL08].

Table 4.1: Modeling Methods for Separable Soft Object

	Numerical	Discrete models	Other Methods
Method	FEM BEM	Mass-spring model Lumped volume	Fracture Mechanics Regional Models
Advantages	Very accurate Complex geometry	Low complexity Cutting easily modeled Low computational load	Global deformation local cut modeled accurately
Disadvantages	Valid for small deformations Computation time Re-meshing issues	Surfacic models Difficulty modeling physical behavior	Specific cases Cutting region is a priori known

Discrete Methods

The fastest techniques tend to be based on discrete modeling methods such as mass spring damper systems [KCM00, LMZP11]. In this case the object is meshed and represented as a series of point masses that are linked by spring-damper systems. Cutting can be easily simulated by simply removing the links between the point masses. On the other hand, the values of the springs and dampers must be individually tuned to replicate the material properties leading to a poor a physical likeness [DCA99]. In order to overcome the drawbacks of the spring damper method an analogous volumetric technique is proposed in [FG99], this however, creates a higher computational cost. Mass spring models that introduce anisotropy to allow a greater degree of realism have developed in [EBG12]. Furthermore such models can be improved by using advanced techniques such as active observers [MLZP12].

Numerical Techniques

Numeric techniques such as Finite Element Methods(FEM) and Boundary Element Methods (BEM) can be used to greatly increase the precision of the model. These tools have been used in many different domains and are hence very advanced. FEM can approximate complex geometries and deformable material properties, the difficulty lies in the interaction between the cutting tool and the object. As the knife cuts through the material, the tetrahedral elements that the knife touches must either be re-meshed [ML03]

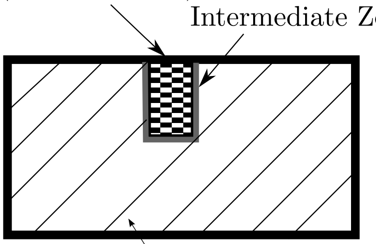

Figure 4.8: Cutting Region Approach

or removed [BN98]. Since both schemes are computationally heavy, alternatives such as constraining the knife's motion [NvdS00] along the element boundaries have been explored. By discretization of the boundary of the deformable object, BEM [MLM+05] reduces the dimension of the problem. However the drawback of such methods is the requirement that the object's interior behavior can be represented mathematically.

Cutting Region & Energy Based Methods

It is clear from the literature that introduction of the cutting requirement increases the required complexity of the model. A promising approach is to create two deformable models that are joined together to form a *cutable* object whose behavior can be calculated in real time [DCA99, VVW04]. The first model makes use of highly accurate numeric techniques to cover a large and complex geometry. The second model is created for a small region of interest. It is within this region where the cutting will take place, thus the tool must be constrained to remain in this zone. Finally an intermediate zone must be defined between the two deformable models which can transmit forces and maintain the integrity of the geometry. An illustration of this idea is given in Fig.4.8.

Finally, an interesting idea is to use energy based approach, which as shown in Section 4.2.4, can also be used to minimize the work done by the cutting tool. Fracture mechanics provides a continuous approach to the modeling of the cutting task and has been experimentally verified for local deformations in [MH01].

4.2.4 Robotic Cutting Formalisms

Historically robots have been widely used to complete cutting tasks of rigid materials, for instance in milling [MSYO99] or bone cutting and surgical applications [TS03]. In addition to this, cutting can be performed using a variety of methods such as laser cutting, ultrasonic cutting and plasma techniques. A recent overview of robotic cutting is given in [Bog08]. Generally these cases use a constant input of a predefined cutting force where this force can be learned or defined offline [ZH97].

Cutting with Shear

When a cut is been performed, it is intuitively clear that the required cutting force can be greatly reduced if a *pressing* and *slicing* approach is performed. This means that rather than just applying a pure force normal to the surface, a force parallel to the surface is also applied. Thus in order to cut the material, a perpendicular (pressing) motion should be combined with a transversal (slicing) motion. As shown in Fig.4.9, the shearing force can be induced in one of two ways, either a *slicing* velocity is added to the tool velocity or alternatively the tool is positioned at an angle to the material [ALZR09].

There are several ways to explain the reduction in cutting forces, the most straightforward is the energy based approach given in [AXJ04]. As shown in Fig.4.9, to cut the material the robot must move the knife along the x-axis of the tool. In order to move the tool a distance of Δx_t the robot must overcome a resistive force, denoted as ${}^t f_x$, therefore the work done by the cutting tool is written as:

$$ {}^t f_x \Delta x_t = K_c w \Delta x_t \tag{4.21} $$

where K_c is known as the material's fracture toughness while w is the width of the blade in contact with the material as shown. If a slicing motion is added, the work required to propagate the cut is now a product of the work done in both the *pressing* and *slicing* directions:

$$ {}^t f_x \Delta x_t + {}^t f_z \Delta z_t = K_c w \Delta x_t \tag{4.22} $$

The resultant force and instantaneous displacement of the tool are defined respectively as ${}^t f_r = \sqrt{({}^t f_x^2 +{}^t f_z^2)}$ and $\Delta p_t = \sqrt{(\Delta x_t^2 + \Delta z_t^2)}$. Therefore assuming the resultant forces are used purely to cut the material, the energy balance can also be written as:

$$ {}^t f_r \Delta p_t = K_c w \Delta x_t \tag{4.23} $$

$$ \xi = \frac{\Delta z_t}{\Delta x_t} \tag{4.24} $$

By introducing the *slice/push* ratio, given in (4.24), into (4.22) and (4.23), the authors derive the following relation:

$$ \frac{{}^t f_r}{K_c w} = \sqrt{\frac{1}{1+\xi^2}} \tag{4.25} $$

From (4.25), it can be seen that an increase in ξ reduces the resultant forces provided K_c is constant. In practice, the assumptions in the derivation mean this relationship is

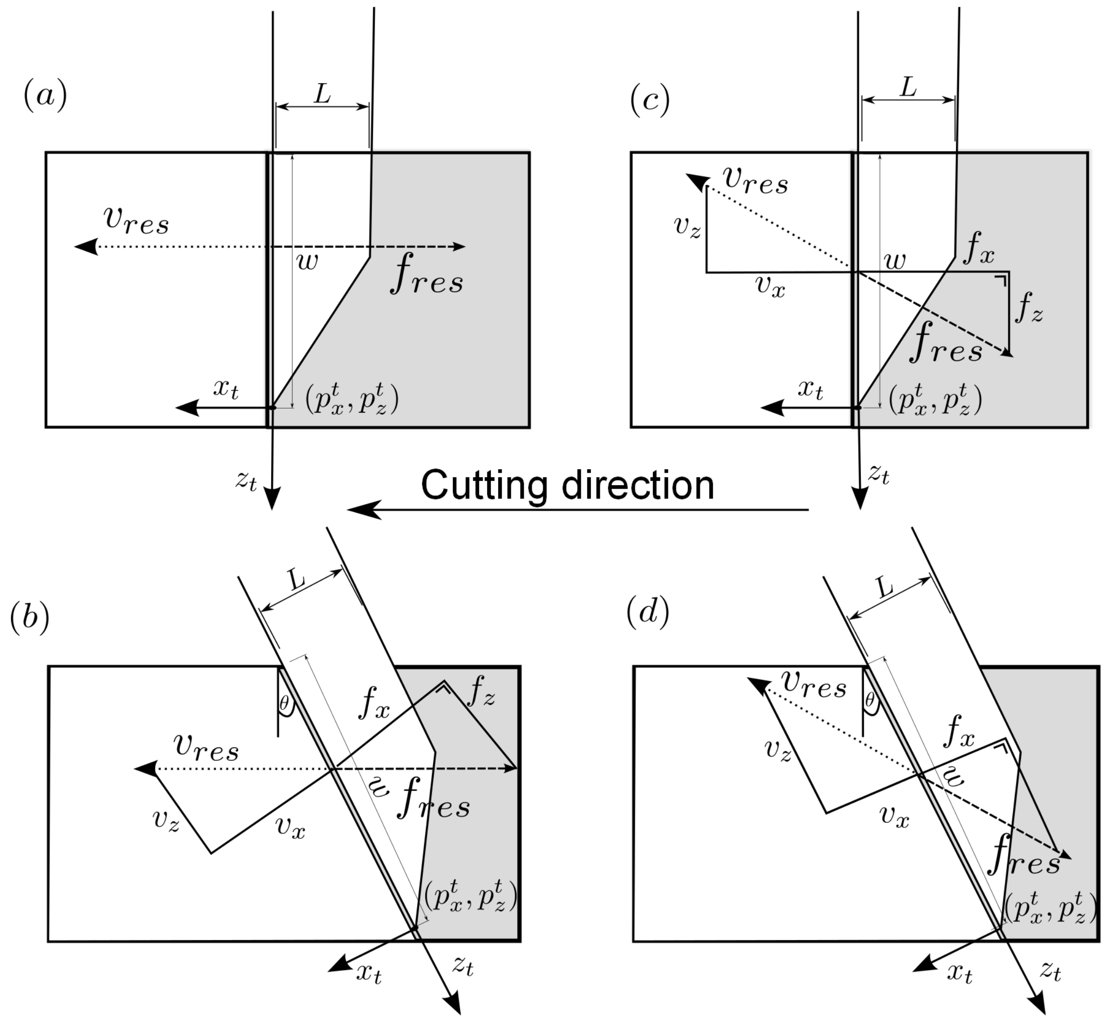

Figure 4.9: Cutting Cases: (a) Cutting angle zero pure pressing, (b) Cutting angle θ pure pressing, (c) Cutting angle zero, pressing and slicing, (d) Cutting angle θ, pressing and slicing

applicable in limited cases. In [ZCLM06, RTLMM12], the reduction of cutting forces, when a shearing motion is introduced is explained by an in depth analysis of the stress at the area of contact. If the cutting action is decomposed into a crack initialization and crack propagation problem [MH01], [RTLMM12] states that the *slice/push* is only effective in the initialization phase. During the propagation phase the effect of slicing on the resultant force is negligible.

4.3 Modeling & Control of a Robotic Meat Cutting Cell

In this section, the modeling, simulation and control of a meat cutting robotic cell is described. The modeling process that takes into account the robot model, the deformable object model, the visual primitives and cutting process is outlined. Force/Vision control schemes are proposed to enable the system to cut an deformable trajectory.

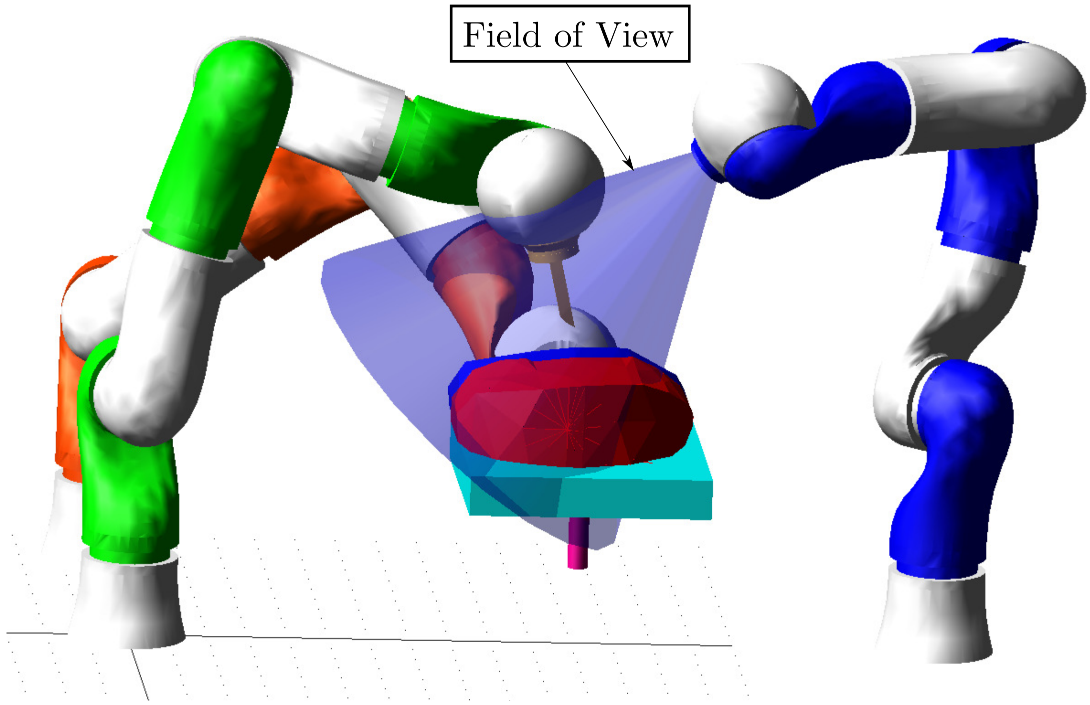

Figure 4.10: Robotic Cell

4.3.1 Robot Model

A global view of the simulation environment is given in Fig.4.10. The system is composed of three Kuka LWR robots, a *cutting* robot, a *pulling* robot and a *vision* robot which will be denoted using the subscripts c, p and v respectively. The robots are kinematically redundant with 7 revolute joints. In the following, all quantities are described in MKS. The robots are fixed with respect to the origin of the world frame.

The Modified Denavit-Hartenberg (MDH) notation [KK86] is used to describe the kinematics of the system. A frame $\mathcal{R}_j$ is fixed on link j such that $\mathbf{z}_j$ is along the joint axis j and $\mathbf{x}_j$ is the common perpendicular between $\mathbf{z}_j$ and $\mathbf{z}_{j+1}$. The parameters defining the frames are given in Table 4.2, where d_j is the distance between $\mathbf{z}_{j-1}$ and $\mathbf{z}_j$ along $\mathbf{x}_{j-1}$. α_j is the angle between $\mathbf{z}_{j-1}$ and $\mathbf{z}_j$ about $\mathbf{x}_{j-1}$. θ_j is the angle between $\mathbf{x}_{j-1}$

and $\mathbf{x}_j$ about $\mathbf{z}_{j-1}$ and r_j is the distance between $\mathbf{x}_{j-1}$ and $\mathbf{x}_j$ along $\mathbf{z}_{j-1}$. The origins of the cutting, pulling and vision robot in the world frame are given as $[0.0, 0.0, 0.0]$, $[0.0, 0.4, 0.0]$ and $[1.0, 0.0, 0.0]$ respectively. The physical location of the joint origins is shown in Fig.4.11.

Table 4.2: MDH Parameters of Kuka robot

j	d	α	θ	$r(m)$
1	0	0	θ_1	0.3105
2	0	$\frac{\pi}{2}$	θ_2	0
3	0	$-\frac{\pi}{2}$	θ_3	0.4
4	0	$-\frac{\pi}{2}$	θ_4	0
5	0	$\frac{\pi}{2}$	θ_5	0.39
6	0	$\frac{\pi}{2}$	θ_6	0
7	0	$-\frac{\pi}{2}$	θ_7	r_7

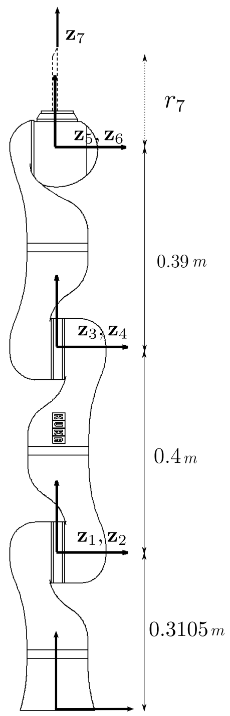

Figure 4.11: MDH Parameters of Kuka robot with physical location of frames

The transformation matrix ${}^{j-1}\mathbf{T}_j$, from frame $j-1$ to frame j is the 4×4 matrix given by:

$$
{}^{j-1}\mathbf{T}_j = \begin{bmatrix} \cos(\theta_j) & -\sin(\theta_j) & 0 & d_j \\ \cos(\alpha_j)\sin(\theta_j) & \cos(\alpha_j)\cos(\theta_j) & \sin(\alpha_j) & -r_j\sin(\alpha_j) \\ \sin(\alpha_j)\sin(\theta_j) & \sin(\alpha_j)\cos(\theta_j) & \cos(\alpha_j) & r_j\cos(\alpha_j) \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4.26}
$$

r_7 represents the tool offset of the robots, for $i = c$, $r_7 = 0.178$ whereas for $i = p, v$, $r_7 = 0.078$. From these parameters, for each robot $i = c, p, v$, the following models are obtained:

$$
{}^0\mathbf{T}_i = \begin{bmatrix} {}^0\mathbf{R}_i & {}^0\mathbf{p}_i \\ 0\,0\,0 & 1 \end{bmatrix} \tag{4.27}
$$

$\mathbf{R}_i$ represents the orientation and $\mathbf{p}_i$ the position of the task frame of robot i w.r.t its base frame. Using a minimal representation of orientation $\mathbf{u}\psi_i$, the Cartesian position, kinematic screw, acceleration of robot i in vector form is given as:

$$\mathbf{x}_i = \begin{bmatrix} \mathbf{p}_i \\ \mathbf{u}\psi_i \end{bmatrix} \tag{4.28}$$

$$\mathbf{V}_i = \mathbf{J}_i \dot{\mathbf{q}}_i \tag{4.29}$$

$$\dot{\mathbf{V}}_i = \mathbf{J}_i \ddot{\mathbf{q}}_i + \dot{\mathbf{J}}_i \dot{\mathbf{q}}_i \tag{4.30}$$

The dynamic model of each robot can be written as:

$$\boldsymbol{\tau}_i = \mathbf{A}_i \ddot{\mathbf{q}}_i + \mathbf{c}_i + \mathbf{J}_i^T \mathbf{h}_i \tag{4.31}$$

$\mathbf{J}_i$ is the kinematic Jacobian matrix and $\mathbf{q}_i$ the vector of joint coordinates while $\dot{\mathbf{q}}$ and $\ddot{\mathbf{q}}$ are the velocities and accelerations respectively. The inertial parameters are taken from the equivalent CAD model. The inertia matrix and the matrix of centrifugal, Coriolis and gravity torques are denoted as $\mathbf{A}_i$, and $\mathbf{c}_i$. The Cartesian wrench is denoted as $\mathbf{h}_i$ while $\boldsymbol{\tau}_i$ is the joint torque.

4.3.2 Deformable object model

The deformable object is at the center of the simulation strategy, it represents the two beef shoulder muscles that must be separated. The simulated object must react to pulling forces in a realistic manner. Furthermore the object must be *separable*, i.e. react coherently to the incisions of the robot controlled knife.

To create the deformable object model, the cutting region approach [DCA99][VVW04] is used. In this case, the knife can only interact with the object within a defined cutting region. As such, two distinct deformable models are generated, a computationally heavy model that allows tool interaction within the cutting region and a computationally efficient model that does not interact with the tool.

For the meat cutting application, this approach is convenient since the objective is to separate two beef muscles that are joined by a non-homogeneous region known as the aponeurosis. The aponeurosis represents the cutting region and is located between the muscles. Therefore this region can be modeled specifically for tool interaction. An example of this approach for meat separation is given in [EBG12].

In summary, three deformable models are used to simulate the for the separation of the beef muscles. Two computationally efficient models, representing the beef muscles, are created off-line using FEM techniques. One spring damper system is used to create the aponeurosis which links the beef muscles together. In the following sections, the steps taken to create the deformable models are outlined.

Beef Muscles In order to create the beef muscles, the following steps, are taken as shown in Fig.4.12. Firstly a visual scan of a generic beef round is obtained, using MRI technology, after separation and converted into a 3D-geometry. The two muscles are reconstructed within a CAD program and the exact cutting surface is extracted. Two simplified muscles are created using the exact cutting surface in order to reduce the computational cost during the simulation. The simplified models are discretized volumetrically using a meshing program. Attachment points are placed at certain nodes, notably on the cutting surface of each muscle. It should be noted that for each attachment point on the cutting surface of one muscle, there exists a corresponding attachment point on the other muscle, whose position is identical in the world frame. These attachment points allow forces and constraints to be applied to the object in the simulator environment. The attachment points on the cutting surface are used to knit the deformable models together allowing the three models to interact. A modal analysis is performed for each muscle and the resulting output is a .mnf file. This file contains the object geometry, the orthonormalization of the Craig-Bampton modes [BC68, Ott00], and the generalized mass and stiffness for the mode shapes. A full description of deformable body modeling using this method is given in Appendix A.

Aponeurosis The surface of separation of the beef shoulder is distinguished by a set of aponeurosis, that are similar to tendons, acting as links between the main beef muscles. The aponeurosis are modeled as the second deformable object located in an intermediate layer in the beef shoulder. The aponeurosis store elastic energy, then recoil when unloaded. This behavior is approximated as a series of spring damper systems fixed to the muscle at discrete points. Each spring damper links two corresponding attachment points on each beef muscle. Thus using the attachment points as nodes a series of spring-damper systems are spread across the cutting the surface. When the two objects are perfectly mated and at rest, the spring damper systems are at their equilibrium points and the net force is zero. Fig.4.13 gives a visualization of the aponeurosis, as the muscles are been separated.

4.3.3 Cutting Process Model

The object is cut by removing the aponeurosis that links the beef muscles together in response to the passage of the knife. In order to achieve this, at each iteration the position of the cutting tool in the world frame denoted as $\mathbf{p}_c = [x_c, y_c, z_c]$ is compared with the position of the aponeurosis located on the cutting surface. The aponeurosis are modeled as a series of spring-damper systems. A spring-damper system denoted as $\mathbf{k}$, acts between two nodes $\mathbf{k}_1$ and $\mathbf{k}_2$ located at $[x_1, y_1, z_1]$ and $[x_2, y_2, z_2]$ respectively. Therefore, throughout the simulation the position of the tool is compared to the position of the line segment $|\mathbf{k}_1\mathbf{k}_2|$. Since the object is fixed to the plane of the table, in order to simplify the calculation it is assumed that the cutting tool always approaches from the positive z direction. This means that for any point on the knife ${}^0z_k$, it is assumed that

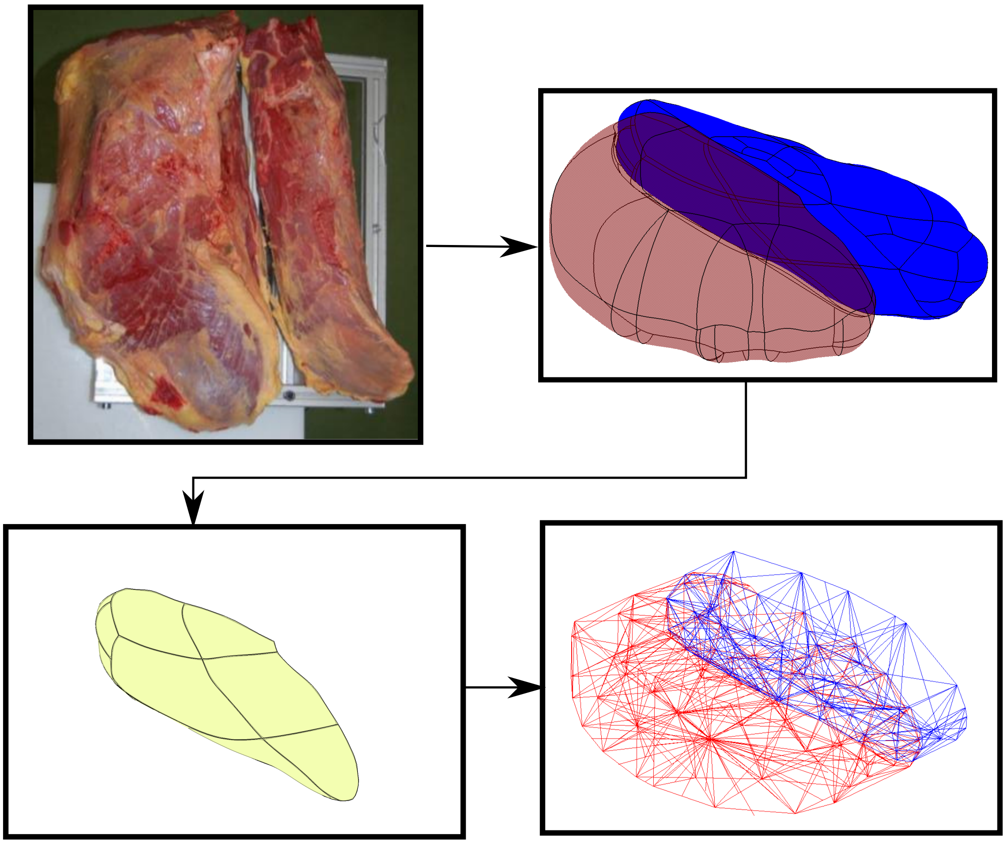

Figure 4.12: Process for meat model
From 3D scan to Finite element mesh, for clarity the different muscles are shown in blue and red

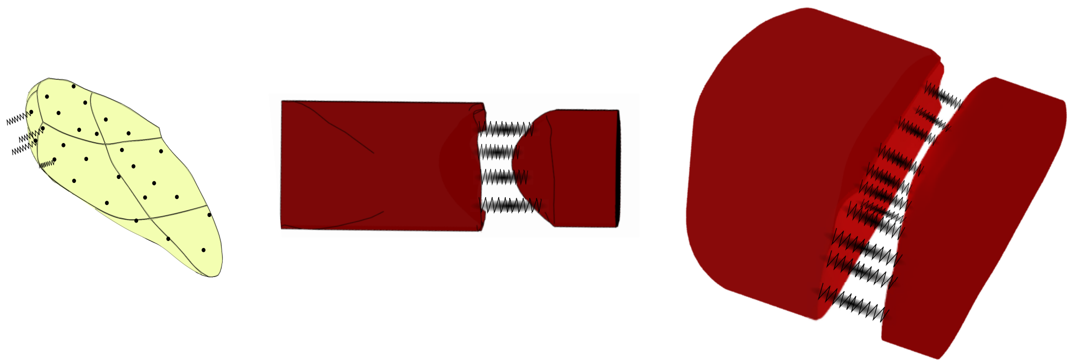

Figure 4.13: Aponeurosis of meat model
The Aponeurosis as a set of spring damper system, distributed over the surface on left, after integration between the beef muscles shown in the middle and right images

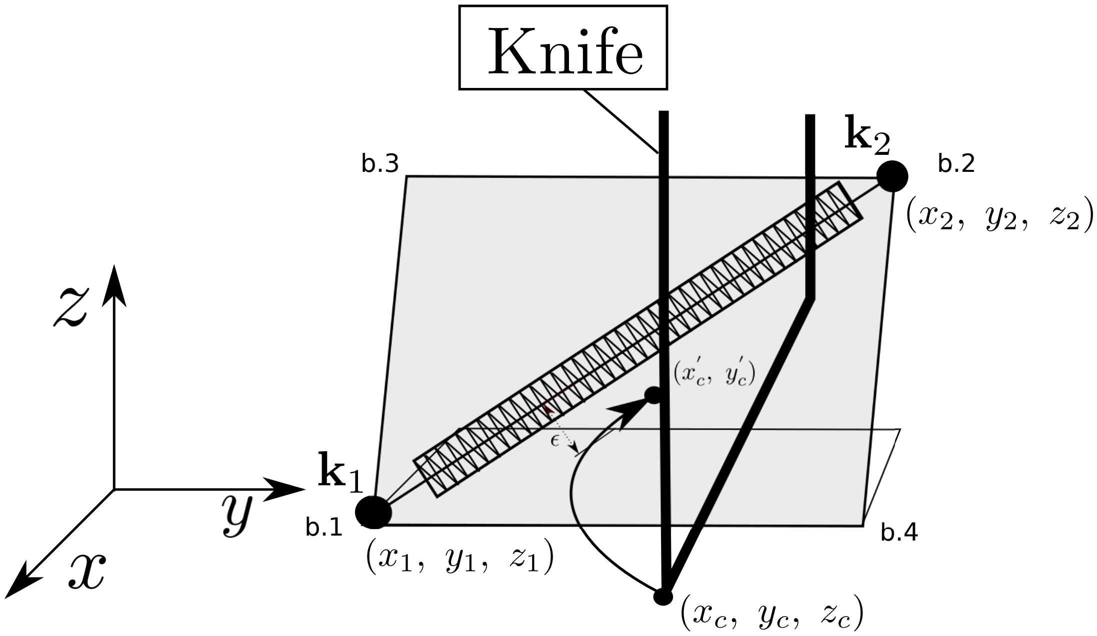

Figure 4.14: Cutting model

${}^0z_c \leq {}^0 z_k$. To check if a spring-damper $\mathbf{k}$ has been cut, all of the following conditions must be met as shown in Fig.4.14:

Condition 1.

$\mathbf{p}_c$ must be below the virtual spring-damper, this condition follows from the assumption that cutting tool always approaches from the positive z direction. The condition ensures knife has reached a sufficient cutting depth.

$$z_c \leq min\,(z_1, z_2) \tag{4.32}$$

Condition 2.

The projection of $\mathbf{p}_c$ into the xy plane must lie within the bounding box $(b1 \ldots b4)$ of the line segment $|\mathbf{k}_1\mathbf{k}_2|$. The bounding box $b1 \ldots b4$ is the rectangle of minimum area enclosing the points $\mathbf{k}_1$ and $\mathbf{k}_2$ whose sides are parallel to x and y axes. This condition ensures the knife is between nodes of the spring damper system, i.e within the cutting region.

$$min\,(x_1, x_2) \leq x_c \leq max\,(x_1, x_2) \tag{4.33}$$

$$min\,(y_1, y_2) \leq y_c \leq max\,(y_1, y_2) \tag{4.34}$$

Condition 3.

The projection of $\mathbf{p}_c$ into the xy plane must lie on the line segment $|\mathbf{k}_1\mathbf{k}_2|$ (ϵ is a

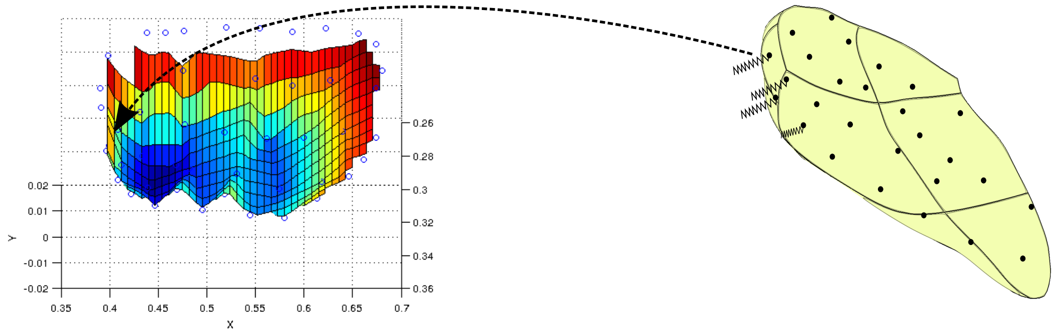

Figure 4.15: Reconstruction of Surface of Separation using visual primitives

tolerance). This condition ensures the knife is lying on the line segment.

$$(y_2 - y_1)(x_2 - x_c) - (y_2 - y_k)(x_2 - x_1) < \epsilon \tag{4.35}$$

If all of the above conditions are fulfilled, the cutting tool is crossing **k**. In response the spring-damper **k** is removed and the simulator is updated.

4.3.4 Generation of vision primitives

The third robot is equipped with an eye-in-hand camera. This camera provides the location of the guide line in space. By following this line the separation of the meat can be achieved. Thus the guideline is the visual primitive that must be taken into account in our environment. In Section 4.3.2, it is shown that the surface can be discretized in order to create an intermediate cutting layer. It is assumed that the vision system is capable of extracting the location of the attachment points. Using the location of these points, the surface can be reconstructed in the control environment using a surface interpolation procedure, the Matlab function **TriScatteredInterp** as shown in Fig.4.15. To create a trajectory for the cutting robot, the curve is extracted from the interpolated surface for the desired cutting depth.

4.3.5 Control Scheme

In this section, the problem of controlling the robotic cell is addressed. Each arm is controlled independently in their respective tasks while the coupling effects are felt through the interaction with the deformable body. A desired value of a parameter is represented as the same variable with the superscript ^{d}. The prefix Δ denotes the difference between a desired value and the current value of variable, for example $\Delta\mathbf{x}_i = \mathbf{x}_i^d - \mathbf{x}_i$.

A global overview of the control scheme is given in Fig. 4.16. From this figure, the three controllers, for the cutting, pulling and vision robot can be distinguished. Each

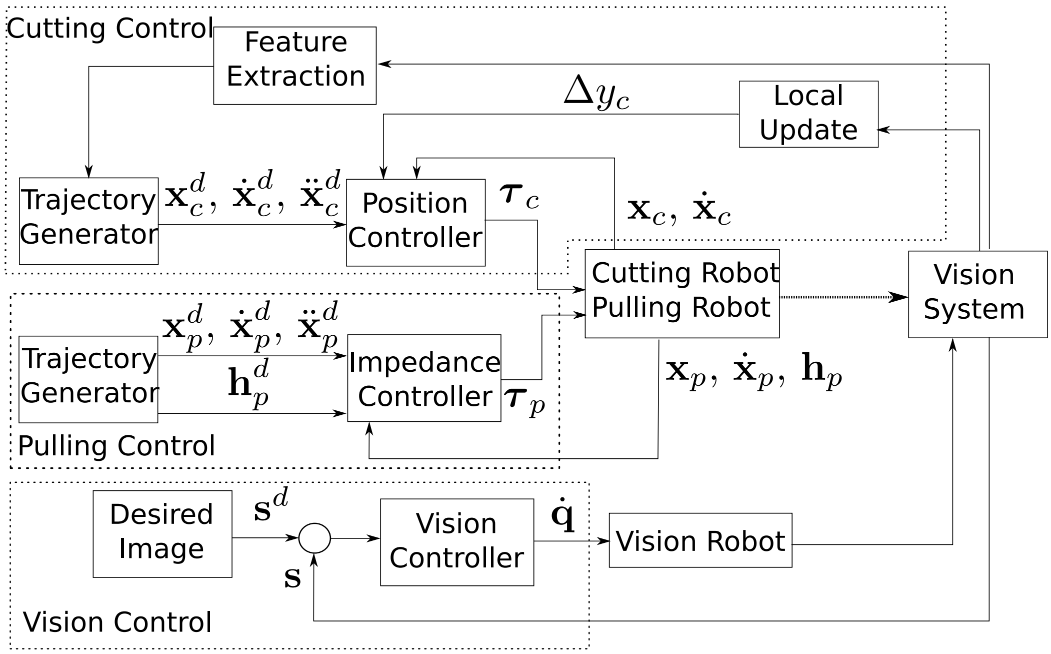

Figure 4.16: Global Control Scheme

robot fulfills a particular function for the separation task. The cutting robot follows a trajectory generated by feature extraction module. During the cut, the meat will deform, therefore a deviation is added to the desired trajectory via the local update module. The pulling robot both retains and applies a force to the object by using force controller to aid the separation and allow greater access for the vision system. Finally, the vision robot must position itself such that the embedded camera can extract the necessary visual features. To do so, the robot is controlled in image space. In the following, the controller for each robot is described in much greater detail.

Task Definition

Primary Task The primary task is the separation of the two meat muscles. To complete this task, the spring-damper links, representing the aponeurosis, must be removed by the passage of the knife. The cutting tool must only interact with the aponeurosis and avoid cutting into the meat muscles at either side. This necessitates a series of cuts, called passages, at increasing depths along the visible guide line. The pulling robot is responsible for creating an opening so that the knife can pass, unobstructed, along the guide line. After each passage the opening will increase allowing the knife to move deeper into the valley until the two objects have been completely separated. The vision robot must alter its pose so that the guide line is kept within the field of view as the meat deforms. In total in order to execute their primary tasks, the cutting robot is required to use 6-DOF, the pulling robot 6-DOF and the vision robot 4-DOF.

Secondary Task Since the robots have 7-DOF, the null space motion must be controlled. A secondary task $\mathbf{Z}$, is used to damp any motion in the null space. The secondary task is projected into the primary task using the classical orthogonal projector $\mathbf{P}_i = \mathbf{I} - \mathbf{J}_i^+ \mathbf{J}_i$, where $\mathbf{I}$ is the 7×7 identity matrix and $^+$ denotes the pseudo-inverse.

Cutting Robot

At the beginning of each passage, the visual primitives are used to reconstruct the guide line as outlined in section 4.3.4. A curve is then fitted to the guide line. This curve is represented by a polynomial expression. For a given cutting depth z, the desired trajectory is defined by:

$$y = a_2 x^2 + a_1 x + a_0 \tag{4.36}$$

The total curvilinear length, D of the polynomial curve is obtained by integrating (4.37), where a and b are the extremities of the surface.

$$D = \int_b^a \sqrt{1 + \frac{\partial y^2}{\partial x^2}}\, dx = |_b^a f(x) \tag{4.37}$$

$$D = f(a) - f(b) \tag{4.38}$$

A variable $\mathcal{T}(t)$ representing the curvilinear distance along the curve is defined using the temporal constraints (4.39), (4.40), (4.41).

$$\mathcal{T}(t=0) = 0 \qquad \mathcal{T}(t = t_{final}) = D \tag{4.39}$$

$$\dot{\mathcal{T}}(t=0) = 0 \qquad \dot{\mathcal{T}}(t = t_{final}) = 0 \tag{4.40}$$

$$\ddot{\mathcal{T}}(t=0) = 0 \qquad \ddot{\mathcal{T}}(t = t_{final}) = 0 \tag{4.41}$$

At any time t, $\mathcal{T}(t)$ is calculated as:

$$\mathcal{T}(t) = r(t)D \tag{4.42}$$

where $r(t)$ is the interpolation function, a 5-DOF polynomial continuous in acceleration given as:

$$r(t) = 10\left(\frac{t}{t_{final}}\right)^3 - 15\left(\frac{t}{t_{final}}\right)^4 + 6\left(\frac{t}{t_{final}}\right)^5 \tag{4.43}$$

$x(t)$ can be obtained by substituting $s(t)$ into (4.38), and then solving the following:

$$f(x) = \mathcal{T}(t) + f(b) \tag{4.44}$$

$y(t)$ is calculated from using $x(t)$ the polynomial expression in (4.36). To complete the cutting task definition, the orientation of the knife must be considered. Before each passage, the orientation of the knife is equal to the 3×3 rotation matrix $\mathbf{R}_{init}$ given in (4.45). The cutting side of the knife must be aligned to the cutting direction, the approach angle is defined by the angle θ, a rotation around the z axis. The desired rotation matrix during the passage, $\mathbf{R}^d(t)$ is calculated from (4.47).

$$\mathbf{R}_{init} = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix} \tag{4.45}$$

$$\theta = \frac{\partial y}{\partial x}(x(t)) \tag{4.46}$$

$$\mathbf{R}^d(t) = \mathbf{R}_{init} \begin{bmatrix} \cos(-\theta) & -\sin(-\theta) & 0 \\ \sin(-\theta) & \cos(-\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{4.47}$$

From the above a desired trajectory is generated in position and orientation, velocity and acceleration, i.e. $\mathbf{x}^d$, $\mathbf{V}^d$ and $\dot{\mathbf{V}}^d$. To track the desired variables, using Cartesian computed torque [Kha87] the desired Cartesian acceleration,$\mathbf{w}_c$, is defined as:

$$\mathbf{w}_c = \dot{\mathbf{V}}^d + \mathbf{K}_d\,(\Delta\mathbf{V}) + \mathbf{K}_p\,(\Delta\mathbf{x}) - \dot{\mathbf{J}}_c\dot{\mathbf{q}} \tag{4.48}$$

where $\mathbf{K}_d$ $\mathbf{K}_p$ are positive gains. $\mathbf{w}_c$ is then transformed to the joint space, and a new desired acceleration exploiting the redundancy of the system is defined:

$$\mathbf{w} = \mathbf{J}_c^+\,(\mathbf{w}_c + \mathbf{P}_c\,\mathbf{Z}) \tag{4.49}$$

Finally a joint torque realizing this acceleration is obtained:

$$\tau_c = \mathbf{A}_c\mathbf{w} + \mathbf{H} \tag{4.50}$$

However the object deforms during the passage of the knife changing the profile of the cutting surface. This is due both to the force applied by the pulling robot and to the effects of cutting the aponeurosis in the intermediate layer. In order to compensate for this motion, the desired position is updated on-line by using, y_g, the exact position of the guide line extracted from the visual primitive. y^d is updated as:

$$y^{*d}(t) = y^d(t) + \Delta y \tag{4.51}$$

$$\Delta y = y_g - y_c \tag{4.52}$$

Pulling Robot

To complete the desired task, the pulling robot must be force controlled. The desired behavior is a gradual opening of the cutting valley as the cutting depth increases. For each passage the number of links between the deformable objects is reduced leading to a smaller retaining force. An impedance controller [Hog85] is applied where $\Delta\mathbf{h} = \mathbf{h}_d - \mathbf{h}$ is the difference between the desired and current pulling force, and where λ is a gain matrix representing the inverse of the desired inertial behavior. Equation (4.48) is modified to include these terms:

$$\mathbf{w}_p = \dot{\mathbf{V}} + \lambda \left(\mathbf{K}_d \left(\Delta\mathbf{V}\right) + \mathbf{K}_p \left(\Delta\mathbf{x}\right) - \mathbf{K}_f(\Delta\mathbf{h})\right) - \dot{\mathbf{J}}_p\dot{\mathbf{q}} \tag{4.53}$$

An overview of the impedance controller can be found in Section 2.2.3.

Vision Robot

For the vision robot, the task is to maintain the cutting zone in the field of view. To do so the vision robot must be controlled in image space. A desired image, which is defined to optimize the field of view, is denoted as $\mathbf{s}_d$. The guide line is extracted by the camera and parametrized into a segment feature variable $\mathbf{s}$ as:

$$\mathbf{s} = \begin{bmatrix} x & y & \theta & l \end{bmatrix}^T \tag{4.54}$$

x, y are the coordinates of the segment's center, θ is the angle and l is the length. The desired velocity is generated using the interaction matrix denoted as $\mathbf{L}_s$:

$$\dot{\mathbf{q}}_v^d = -\mathbf{K}_p \left(\mathbf{L}_s\mathbf{J}_v\right)^+ \left(\mathbf{s}_d - \mathbf{s}_{im}\right) \tag{4.55}$$

4.3.6 Results

The simulator is executed in Msc Adams ©, a multi-body dynamic simulation software. The control system is managed by the Adams control plugin via a co-simulation. In this case the control system is written and executed in Matlab/Simulink ©. The deformable object is created externally by a Finite Element program and integrated using the Adams Flex plugin. Two different experiments are discussed in this section, differing with respect to the reference trajectory:

1. Using the interpolated guide line state at the beginning of each passage
2. Locally updating the guide line using predicted errors

In order to fully separate the muscles the knife must cut a distance of 80mm. The meat is separated by a repeatably cutting along the surface of separation with the knife. The

Table 4.3: Cutting depths per passage

Passage number	1	2	3	4	5
Cutting Depth (m)	0.35	0.33	0.31	0.29	0.27

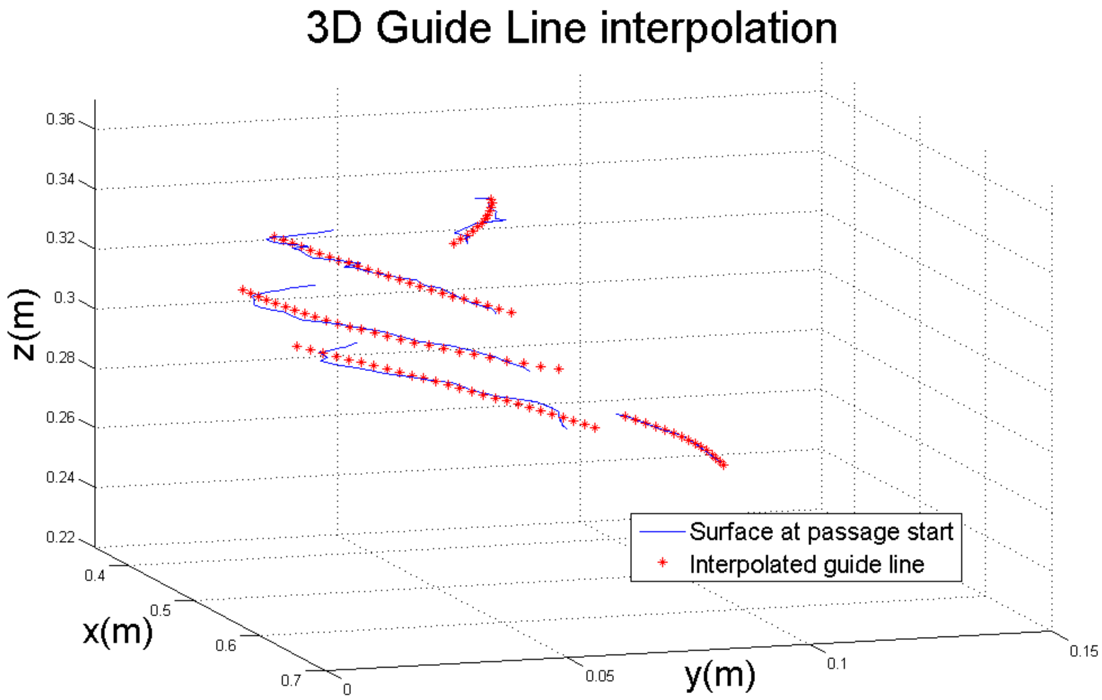

Figure 4.17: Guide Line Interpolation

meat is positioned on a table at a height of 270mm The cutting depth per passage in the world coordinates is shown in Table 4.3.

A 3D view of the cutting trajectories and the initial surface state is given in Fig.4.17. In Fig.4.18 and Fig.4.19, the motion in the $x - y$ plane for the first four passages of the cutting tool is shown. The graphs show: the initial guide before cutting has commenced, the interpolated trajectory for this line, the position of the guide line during the cutting trajectory and the position of the robot cutting tool. Fig.4.22 shows both the sensed forces at the pulling frame and also the y-position of the pulling frame in the world frame. Finally, in Fig.4.20, the difference between the performance of the off-line estimation scheme and the scheme that uses the local updates. It should be noted that during the trajectory, the surface interpolation function may not be able to reconstruct the exact surface position, for example when the cutting depth is higher that the meat muscle at certain x-y $positions$. In this case the controller reverts to the offline estimated trajectory.

Referring to Fig.4.17, it can be seen that the interpolated cutting trajectories give a good approximation of the surface state before the robot begins the cut. However both Fig.4.18 and Fig.4.19 show a large difference between the initial guide line before cutting has begun and the position of the guide line during the trajectory. This is due to the deformation of the object as the links are severed and the continuous application of the pulling force.

The pulling force is shown in Fig.4.22 to remain constant at 100N, while it can be seen that during the trajectory the position of the pulling frame changes in the y-position, corresponding to the cutting of the aponeurosis. As the links are cut the retaining elastic forces are unable to withstand the pulling forces leading to a gradual opening of the deformable object.

Without the local vision update, the robot is unable to compensate for the on-line deformation of the object and, as shown in Fig.4.18, follows the initial interpolated trajectory instead of the new deformed guide line. For large deformations, this trajectory may no longer be valid and risks moving the knife into contact with the beef muscles instead of the aponeurosis. Furthermore, large changes in the curvature of guide line would mean that the cutting edge of the knife is no longer tangent to the trajectory. By using the local vision system, we can see that the visual deviation compensates for these changes by changing the position and orientation of knife in response to the object deformation. In Fig.4.19, it can be seen that this compensation changes the robot motion such that the cutting tool is much closer to the current guide line position.

This difference is illustrated more clearly in Fig.4.20. This graph shows the error in the y direction for the two schemes. The error is defined as the difference between the actual position of the meat surface and the position of the cutting tool. Initially, during passage 1, the error is small. However, as the cutting progresses and the effect of the pulling robot is increased, the difference between the two schemes becomes clear.

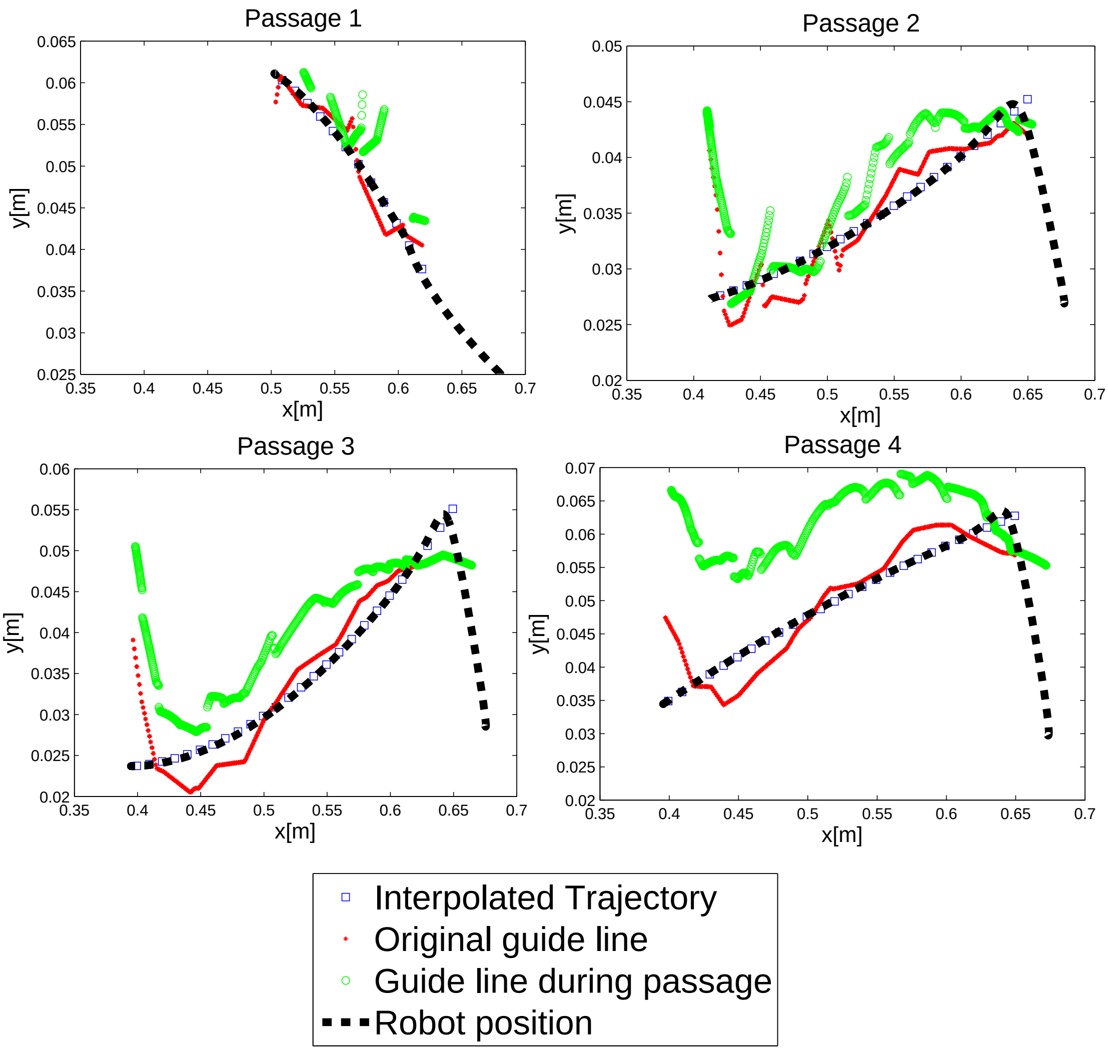

Figure 4.18: Robot Trajectory for each passage

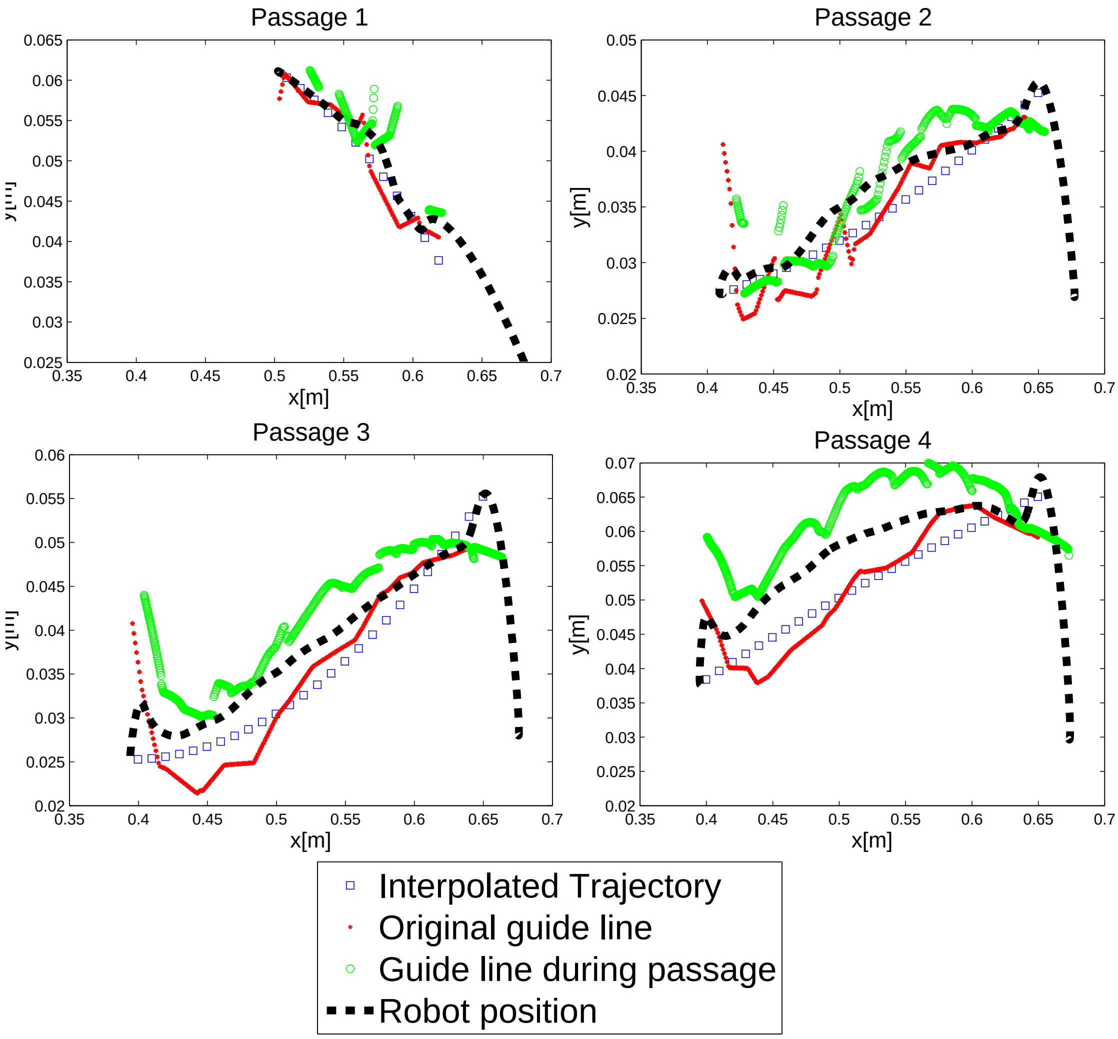

Figure 4.19: Robot Trajectory modified by local updates

Finally an overview of the resulting behavior can be seen in Fig.4.21. This image is split into six panes. Each pane gives two separate views of the simulator. By examining the image, the evolution of the system can be seen as the cutting progresses.

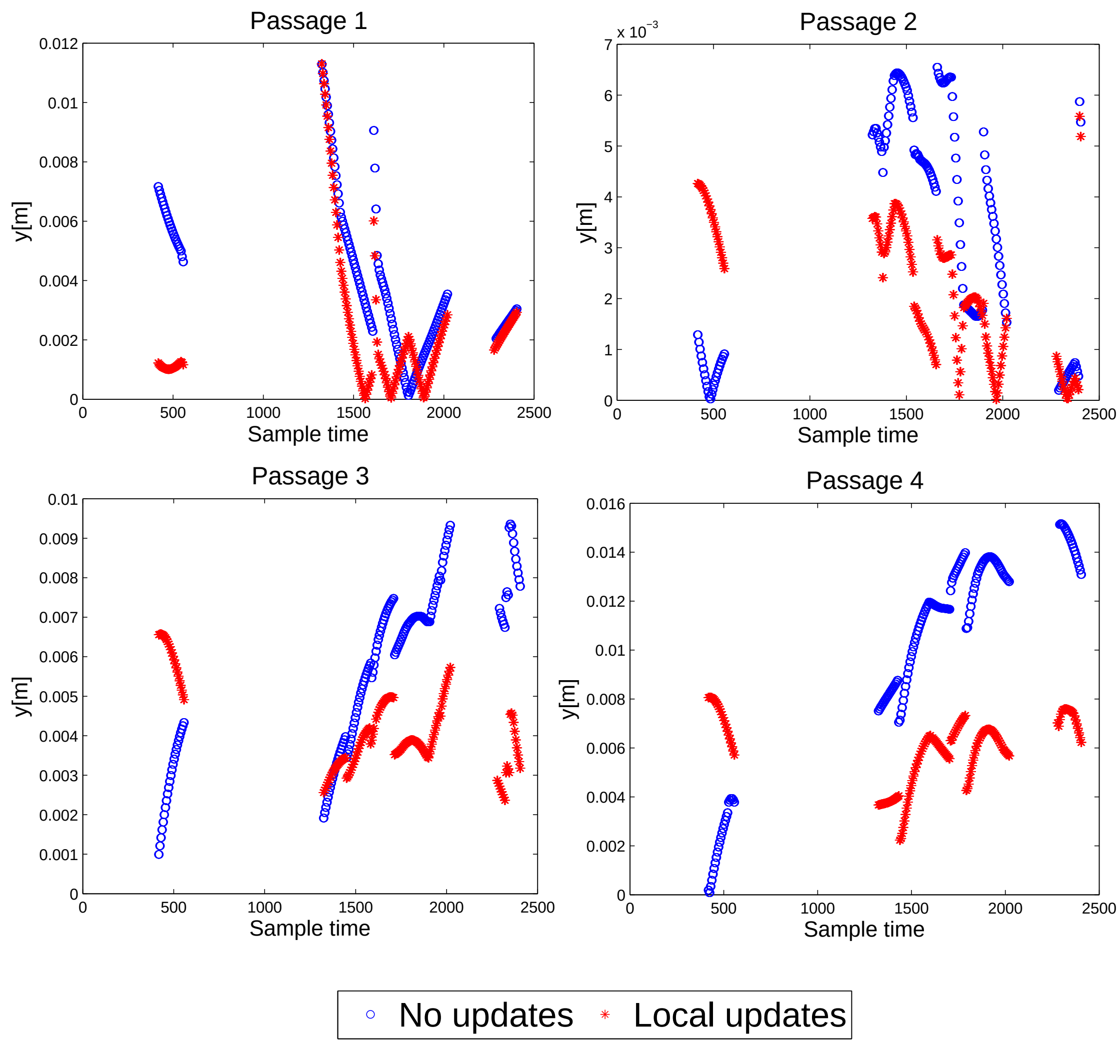

Figure 4.20: Absolute error in the y direction between the actual surface y position at time t and the robot y position at time t

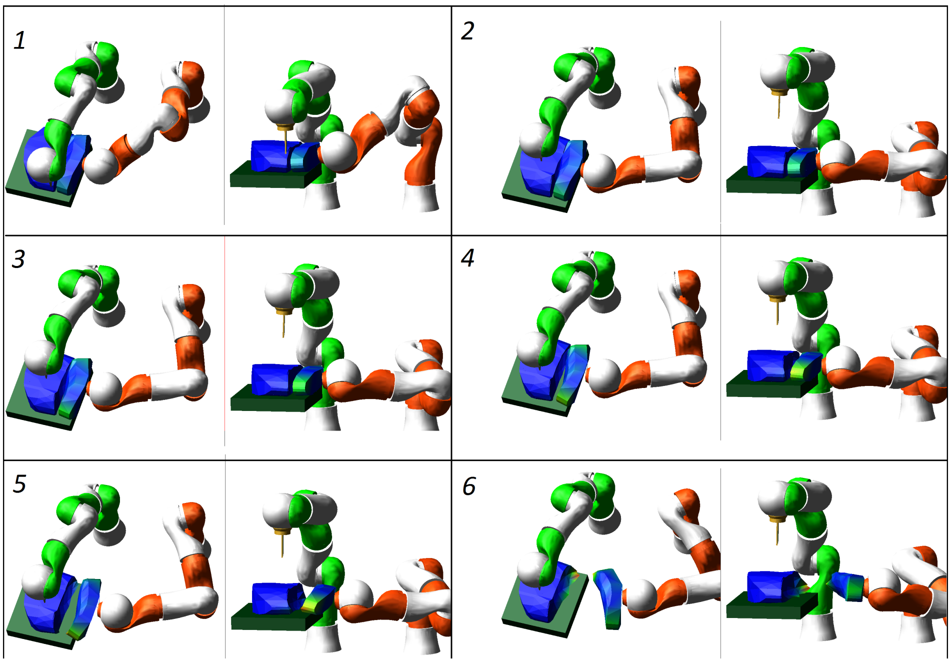

Figure 4.21: Snapshot of separation process

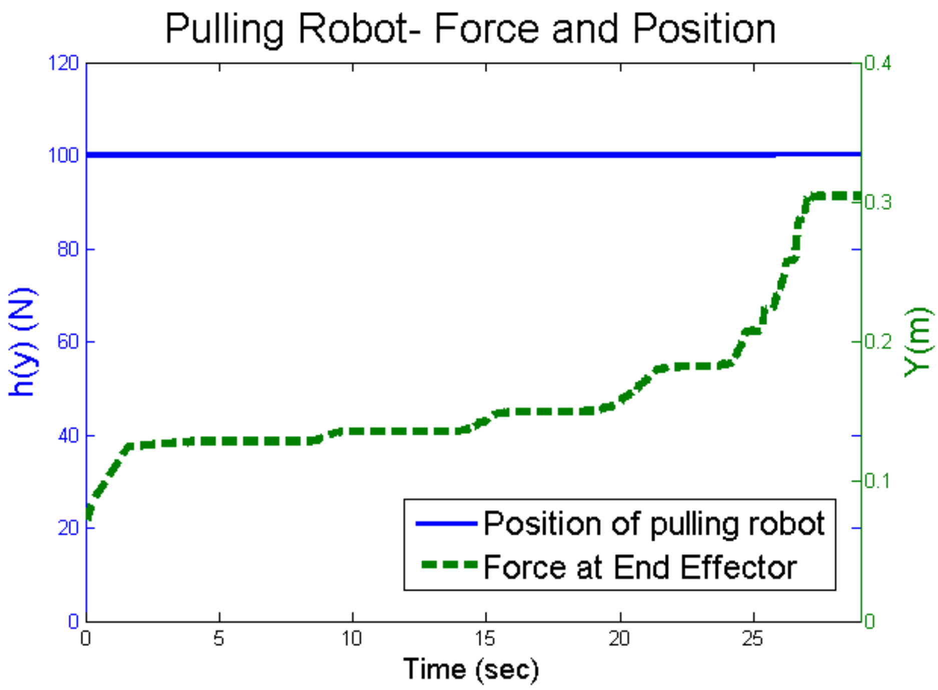

Figure 4.22: Force and Position in Y-direction

4.4 Force/Vision Control for robotic cutting of Deformable Materials

In this section, an experimental validation of a proposed force/vision controller is carried out. The objective is to experimentally verify the control scheme proposed in Section 4.3. In addition to this, the aspects that could not be treated in the simulator environment for example the cutting forces and imperfect visual extraction are examined. In particular, the force feedback of the tool is used to modify the cutting trajectory to prevent global deformation of the material.

4.4.1 Robotic Cell

Two 7-DOF Kuka LWR robots are used for this experiment, as shown in Fig.4.23. The cutting robot is fitted with a ATI gamma 6-axis force sensor, a marlin 1394 camera and a razor blade, whereas the pulling robot is equipped with a set of hooks to grasp the soft object. The parameters for the robot are given in Section 4.3, Table 4.2 and in Section 4.3, Fig.4.11. From these parameters, the following geometric and kinematic models are obtained:

$$ {}^0\mathbf{T}_t = \begin{bmatrix} {}^0\mathbf{R}_t & {}^0\mathbf{p}_t \\ 0 & 1 \end{bmatrix} \quad , \qquad \mathbf{V}_t = {}^0\,\mathbf{J}_t\dot{\mathbf{q}} \tag{4.56} $$

Furthermore, the following quantities are obtained through off-line calibration methods:

– ${}^t\mathbf{T}_f$ the force sensor with respect to the tool frame. This matrix is obtained from CAD data and confirmed by precision measurement of the force sensor dimensions.
– ${}^7\mathbf{T}_t$ the tool frame with respect to frame 7. This matrix is obtained by precision measurement of the tool dimensions.
– ${}^t\mathbf{T}_c$ the transformation matrix of the camera with respect to the tool frame. This transformation matrix is obtained using the calibration procedure outlined by [Tsa87]. This method consists of using an object of known 3D parameters such that the vision system can reconstruct ${}^c\mathbf{T}_{obj}$, the location of the object with respect to the camera. Several poses are recorded and for each pose the transformation matrix from ${}^0\mathbf{T}_t$ is also noted. Since ${}^0\mathbf{T}_{obj}$ is constant, a least square method can be used to resolve the over-determined set of equations.
– $\mathbf{C}$ is the matrix of intrinsic camera parameters. This matrix is obtained using a calibration sheet of known properties. The calibration is carried out using the ViSP program [MSC05].

$\mathcal{R}_t$, which denotes the tool frame, is fixed to the terminal point of the razor blade. It can be seen that the optical axis, denoted by the axis z_c is not aligned with the axis of the tool. Instead the camera is tilted towards the tool in order to allow a greater view of the cutting zone.

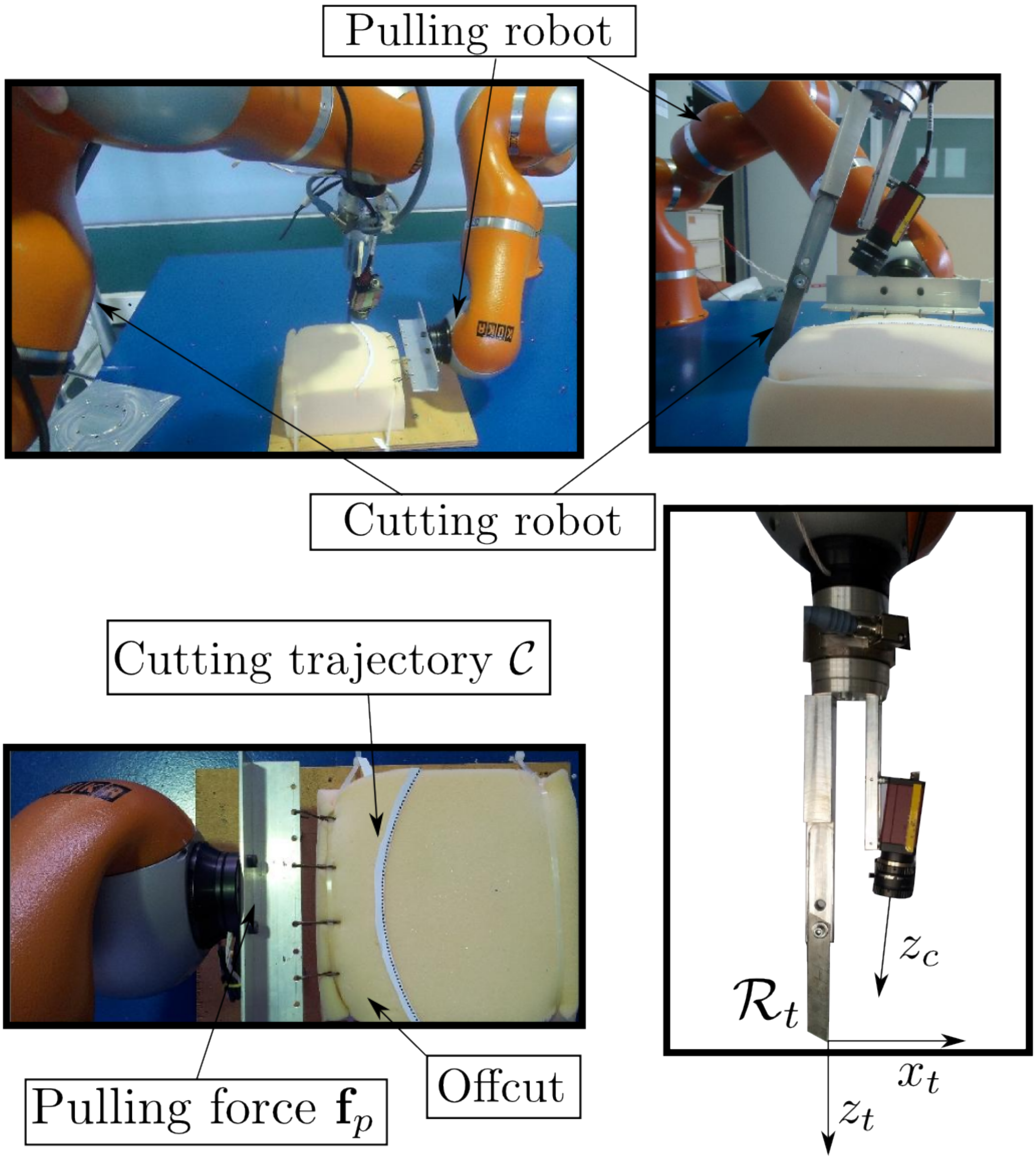

Figure 4.23: Experimental Platform

4.4.2 Proposed Cutting strategy

The objective of the cutting strategy is to minimize the required cutting force at the tool frame for two reasons. Firstly if the cutting force is too large, a soft material can undergo global deformation rather than rupture, leading to the clustering of material around the cutting tool. This global deformation is undesirable from the cutting point of view since the result reduces the product quality. Secondly a smaller cutting force reduces the energy input of the system whereas a larger cutting force may be outside the capabilities of the tool.

An energy balance equation (4.57) is used to describe the cutting process. W_r is the work done by the cutting tool. It is defined as the sum of W_c, the energy required to cut the material; W_f, the work done in overcoming the frictional effects on the blade and U, the strain energy due to global deformation of the soft material.

$$W_r = W_c + W_f + U \tag{4.57}$$

During a pure cutting motion, it is assumed that the global deformation caused by the cutting tool is negligible, $U = 0$, therefore $W_r = W_c + W_f$.

Neglecting for the moment the frictional effects, thereby assuming the work done by the robot is used purely to cut the material, (4.57) becomes $W_r = W_c$. It is shown in Section 4.2.4, that the energy required to initialize a crack can be reduced by adding a shear element to the cutting motion. It should be recalled that the ratio of cutting velocity with shearing velocity is defined in Section 4.2, equation (4.24) as $\xi = \frac{\Delta z_t}{\Delta x_t}$.

Therefore, by examining (4.57) we propose two strategies in order to minimize the required cutting force:

1. To modify the ratio ξ, in response to the presence of resistive forces
2. To decrease the force required to overcome the frictional effects.

It is shown in Section 4.2, Fig.4.9, that the ratio ξ can be increased by either changing the cutting angle of the blade or by increasing the velocity in the z direction i.e. the velocity parallel to the cutting surface. It is undesirable to increase the cutting angle during the trajectory due to both the practical difficulties and the reduction in material feed. Therefore the *slicing* velocity is linked to the resistive cutting force by an impedance controller as detailed in Section 4.4.4. In doing so, global deformations are avoided as the knife enters regions of varying material toughness. On the other hand in the absence of resistive forces, typically during the crack propagation phase, the trajectory of the cutting tool remains unchanged.

In order to reduce the effects of friction, a force denoted as $\mathbf{f}_p$ is applied by the pulling robot. The friction is due to shear as the soft material rubs against the sides of the blade. The work required to overcome the friction is defined as [AXJ04]:

$$W_f = 2Lw\tau_f\Delta_x \tag{4.58}$$

where τ_f is a shear stress acting over length L. The pulling force opens the cutting valley meaning that contact between the cutting tool and the material is reduced. However the drawback of this pulling force is the increase in the deformation of the object, notably the cutting trajectory.

Force Controller

The role of the force controller in the proposed cutting strategy is to ensure the material is cut cleanly i.e. no global deformation occurs. In this case an adaptive controller is defined at the tool frame to generate the force correction term ${}^{ob}d\mathbf{X}_t^f$. However in contrast to the standard force controllers, for the cutting task the adaptive controller is designed such that the resistive force creates a change in position in an orthogonal axis. ${}^t f_c$ is defined as the resistive force of the cut, therefore the change in position is generated as follows:

$$\Delta z_t = \min\left(0,\ \ k_z\,{}^t f_c\right) \tag{4.59}$$

where k_z is a positive gain. From (4.24) and (4.59), it is obvious that the controller will increase the *slice/press* ratio in response to a resistive force thereby reducing the resistive forces and allowing the cutting to continue without deforming the material. By using the min function the positive values of (4.59) are rejected. These values are due to noisy force sensor measurements and would cause the knife to enter deeper into the material.

4.4.3 Proof of Concept: Force Controller

In this section, a series of experiments are described that demonstrate the feasibility of this force controller in a simple cutting scenario. The behavior of the force controller is investigated with respect to changes in the cutting angle, θ as shown in Fig.4.9, and the gain k_z from (4.59). Furthermore, this section is to demonstrates the advantages of this control law with respect to a baseline position control.

Experimental Setup

For each experiment the robot followed a straight line cutting trajectory with a constant velocity. This trajectory is defined by a linear interpolation from point to point. The control law is given as:

$$ {}^t\mathbf{V}_t = {}^t\mathbf{S}_{ob}\left(\mathbf{k}_p d\mathbf{X} + \mathbf{k}_v \mathbf{V}^d\right) + \begin{bmatrix} 0 \\ 0 \\ k_z^t f_c \\ 0 \\ 0 \\ 0 \end{bmatrix} \tag{4.60}$$

$$\dot{\mathbf{q}} = {}^t\mathbf{J}^+\,{}^t\mathbf{V}_t \tag{4.61}$$

where $d\mathbf{X}$ is the position error in the object frame and $\mathbf{V}^d$ is the desired cutting velocity. When the knife exits the media, due to the slicing effect of the controller, the robot returns to the initial position to restart the passage.

Results

In total twelve experiments were carried out. The test matrix and the quality of the cut for each test is shown in Table 4.4. The quality of the cut, which depends on the level of global deformation and rupture in the object, was decided by visual inspection. An example of three cases is shown in Fig.4.27. These cases are described as:

- Good: No global deformation, an extremely clean cut
- Medium: Slight global deformation, in the cutting region
- Poor: Large global deformation and permanent damage to surrounding area

Table 4.4: Test Matrix for Force Controller

	$k_z = 0.0$	$k_z = 0.001$	$k_z = 0.005$	$k_z = 0.01$
$\theta = \dfrac{\pi}{12}$	Poor*	Medium	Good:	Good
$\theta = \dfrac{\pi}{6}$	Poor	Medium	Good	Good:
$\theta = \dfrac{\pi}{4}$	Poor	Good	Good	Good:

The table shows that as expected, the quality of the cut can be increased either by changing the cutting angle or by increasing the force gain. It should be noted that for the experiment $k_z = 0.0$, $\theta = \frac{\pi}{12}$, the knife deformed the object without any cutting. This resulted in a constant increase in force until the experiment was stopped, to prevent damage to the robot and the tool. The increase in force can been seen in Fig.4.24.

The graphical results for $\theta = \frac{\pi}{12}$, $\theta = \frac{\pi}{6}$ and $\theta = \frac{\pi}{4}$ are shown in Fig.4.24, Fig.4.25 and Fig.4.26, respectively. Each figure consists of six sub-figures arranged in two rows and three columns. The top row shows the cutting forces as the cutting distance is increased. The bottom row shows the corresponding cutting depth as the cutting distance is increased. Each of the three columns shows the results of a particular passages. Although in the case of $k_z = 0.0$, the robot completes is only one passage since there is no slicing action.

For each cutting angle, it can be seen that by increasing the value of k_z, the resulting resistive force is reduced. Furthermore for each value of $k_z > 0$, the results show a decrease in the cutting forces as the controller begins the slicing phase. This generates a n-shaped for the force response and thus shows the effectiveness of the proposed controller. In contrast, the position controller $k_z = 0.0$, results not only in a poor quality, as shown in Table 4.4, but also high forces on the cutting tool reaching up to 32 Newtons in Fig.4.24.

However, a drawback of increasing the force gain is the reduction in cutting distance. For example Fig.4.26, the control law with $k_z = 0.001$ has cut a distance of over $200mm$ at the end of the third passage whereas $k_z = 0.001$ has cut less than half this distance.

By increasing the cutting angle, the force on the blade is decreased for all tests. This is expected since the cutting angle also increases the *slice/press* ratio. For this set of experiments, the cutting depth was constant, however in practice by increasing the cutting angle, the possible cutting depth and therefore cutting feed is reduced.

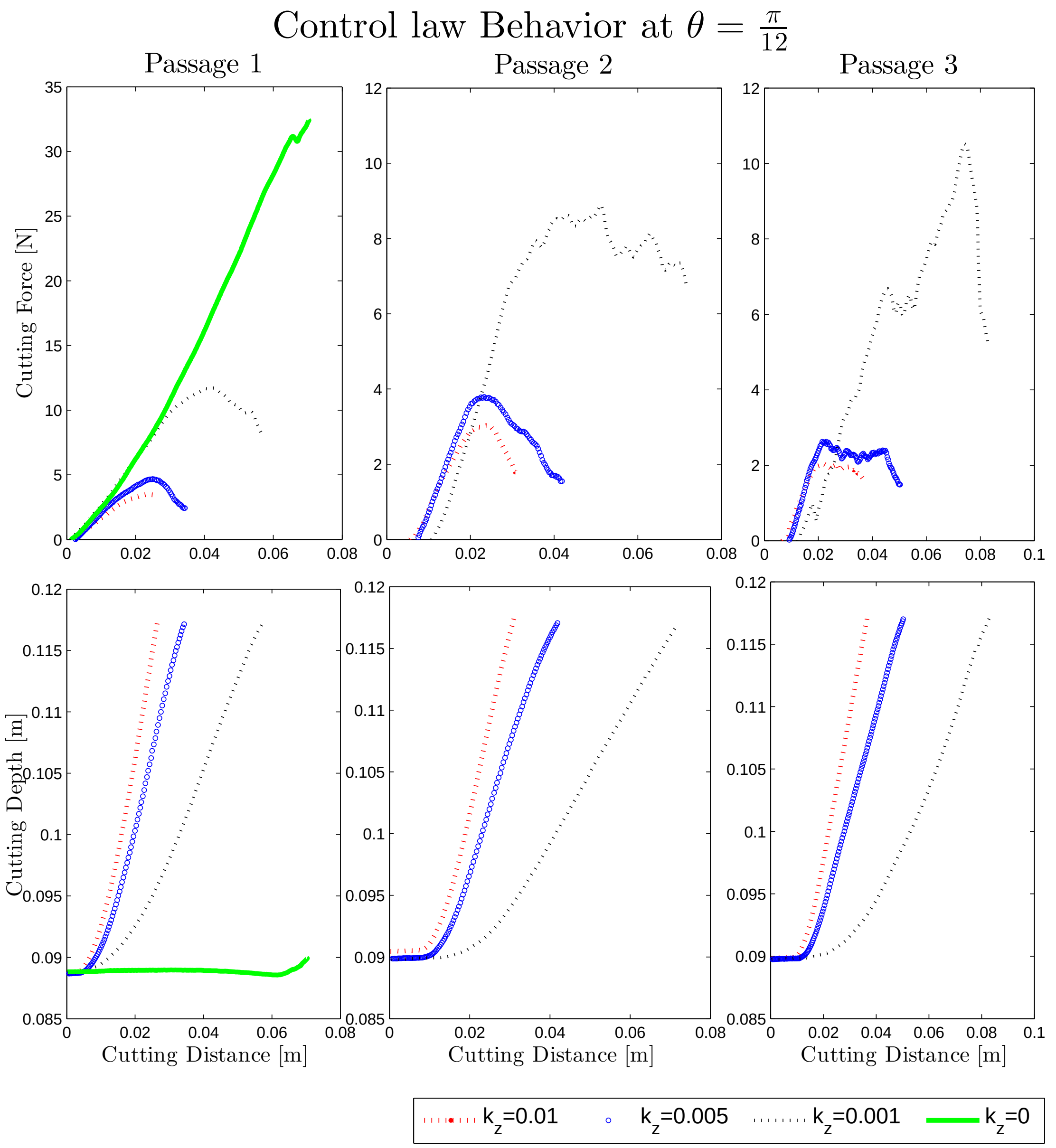

Figure 4.24: Cutting Force versus displacement with $\theta = \frac{\pi}{12}$. The cutting forces versus cutting distance are shown on the top row. The cutting depth versus cutting distance are shown on the bottom row. Each column represents a passage.

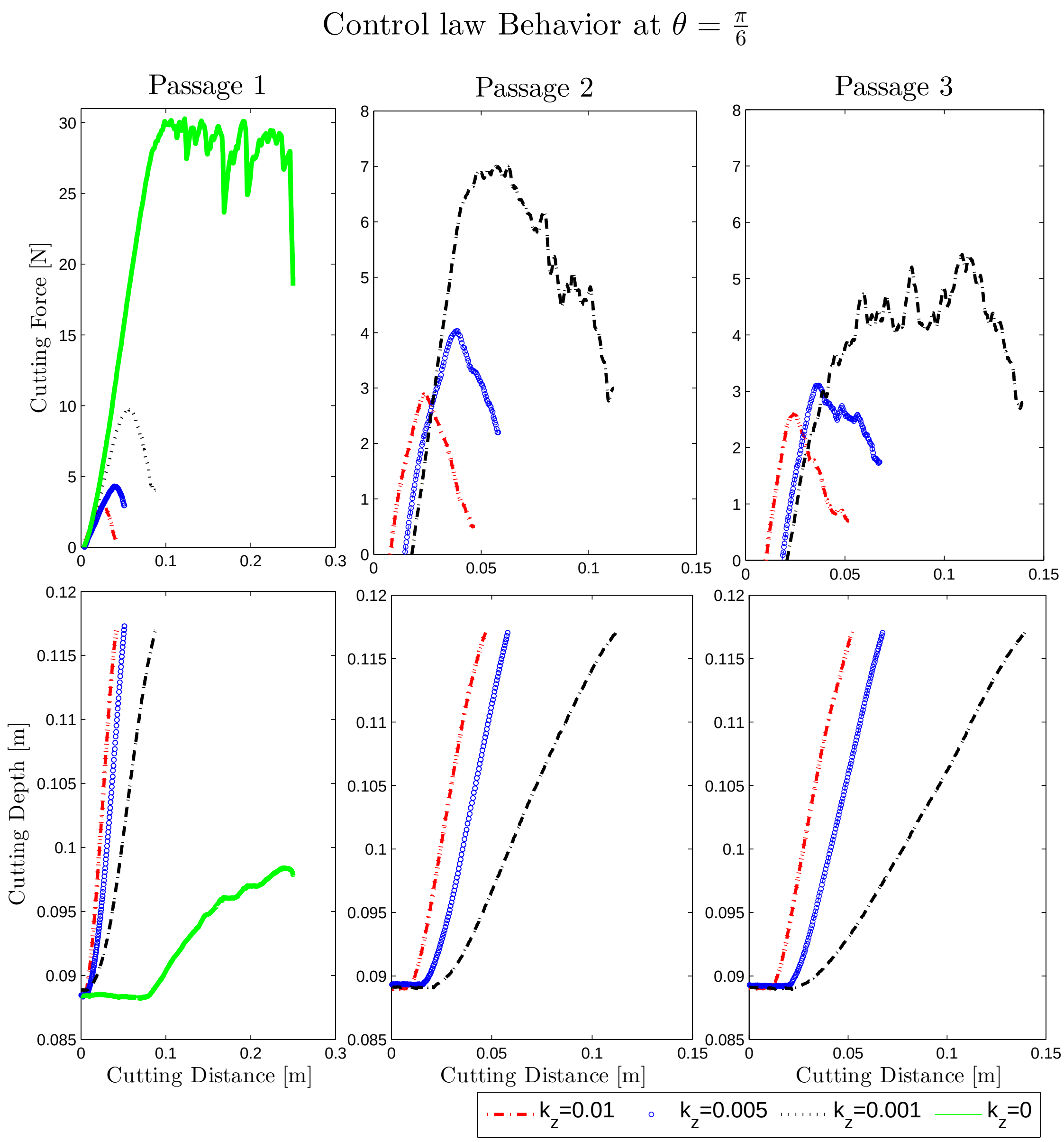

Figure 4.25: The cutting forces versus cutting distance are shown on the top row. The cutting depth versus cutting distance are shown on the bottom row. Each column represents a passage.

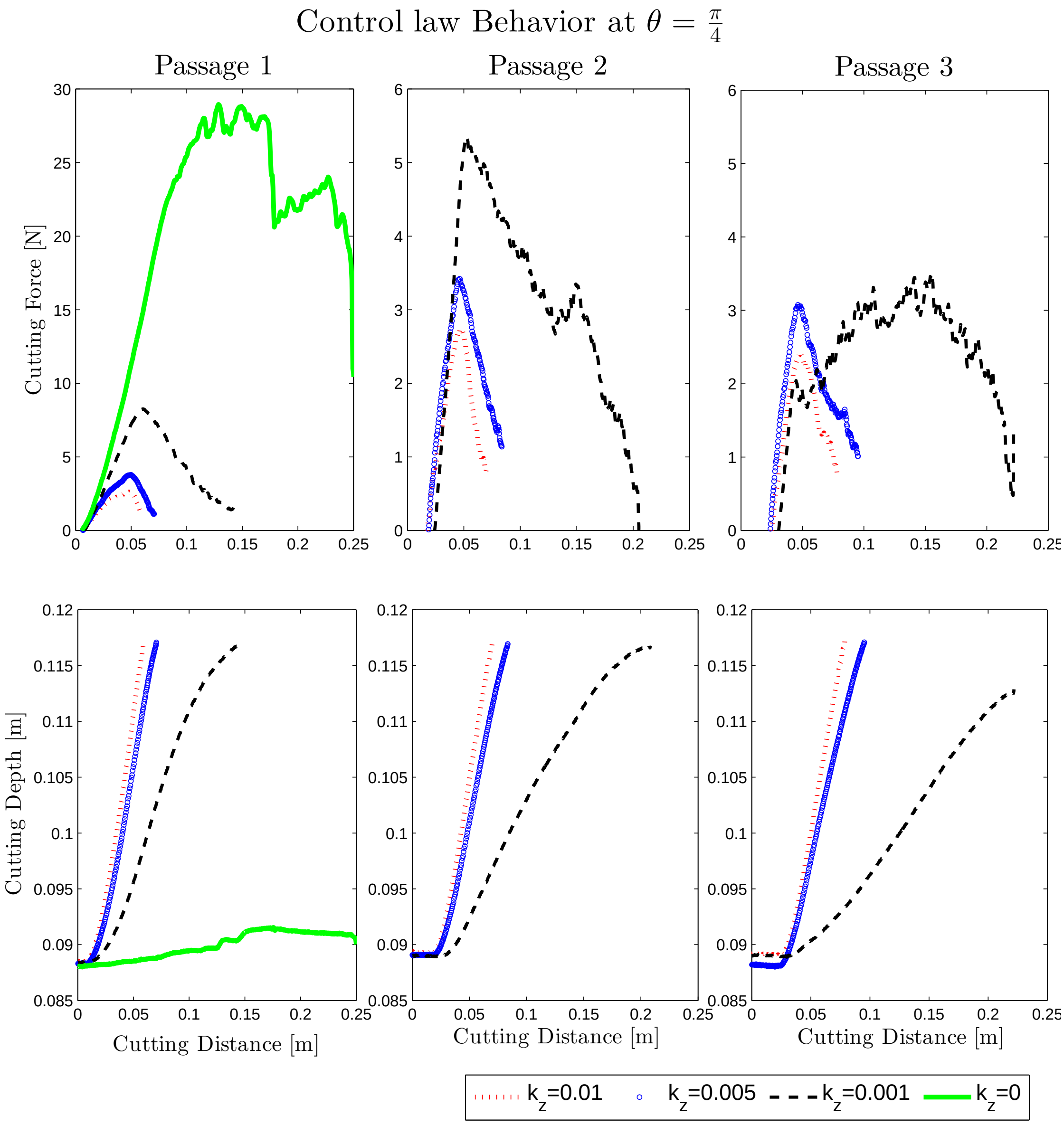

Figure 4.26: The cutting forces versus cutting distance are shown on the top row. The cutting depth versus cutting distance are shown on the bottom row. Each column represents a passage.

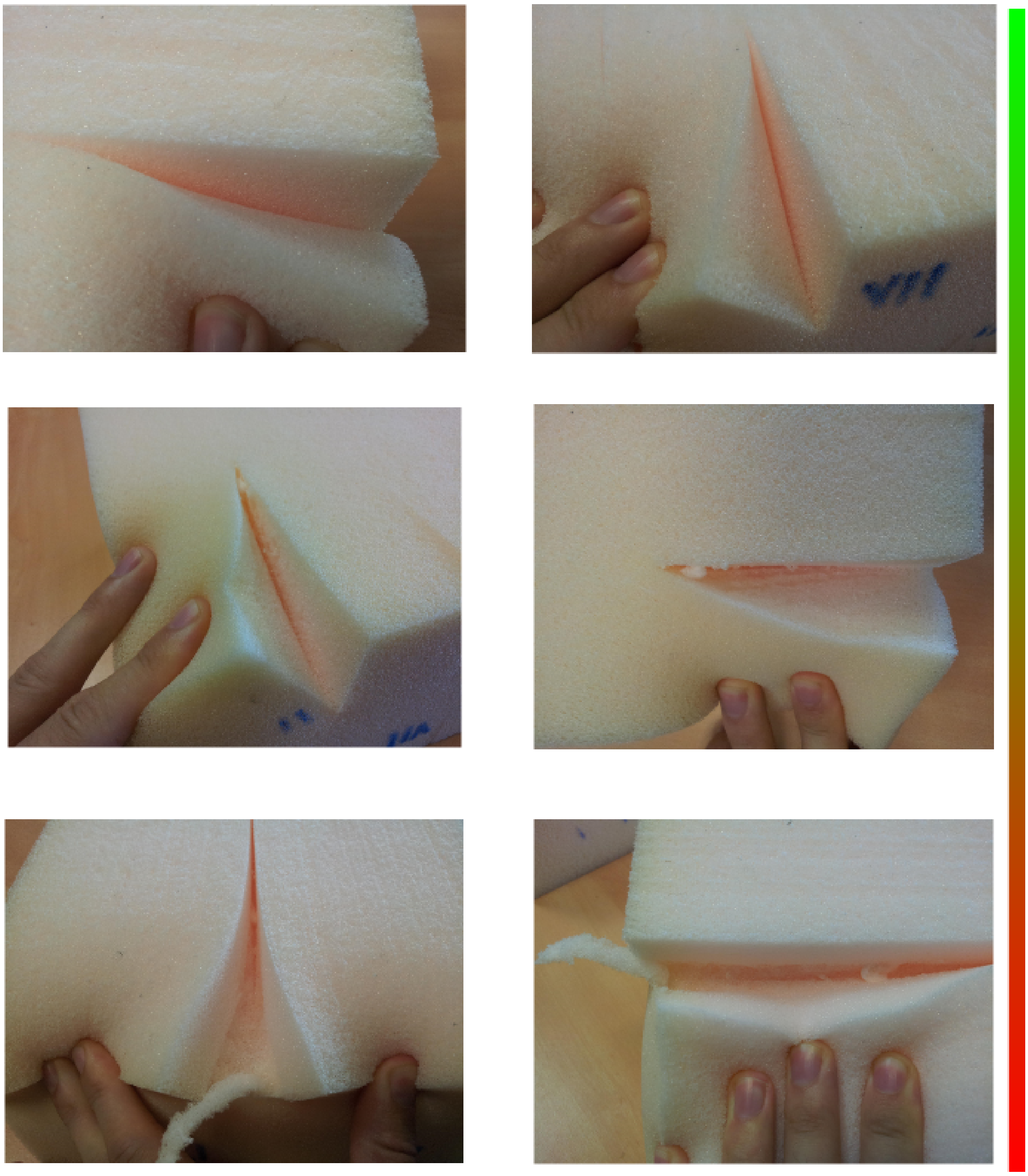

Figure 4.27: Comparison of Cut Quality for the proposed force controller, (Top) A good quality cut with no global deformations, where the force gain $k_z = 0.01$ and the cutting angle $\theta = \frac{\pi}{4}$, (Middle) A medium quality with some small deformations where the force gain $k_z = 0.001$ and the cutting angle $\theta = \frac{\pi}{12}$, (Bottom) Poor quality with large global deformations where the force gain $k_z = 0.0$ and the cutting angle $\theta = \frac{\pi}{4}$.

4.4.4 Force/Vision Controller using PBVS

Outline

In order to separate the object, the cutting robot must follow a deformable trajectory on the soft body. The desired pose is updated using vision and force. The controller generates a Cartesian velocity that is transformed into a joint velocity, by firstly representing the kinematic Jacobian matrix in the object frame and then obtaining the pseudoinverse of this matrix, denoted as ${}^{ob}\mathbf{J}^{+}$. The joint velocity is transformed into a joint torque using the Kuka's internal controller before being sent to the motors. The global control scheme is shown in Fig. 4.28.

Pulling Robot Controller

The pulling robot is used to both hold the object in place and to open up the cutting valley. The pulling force is applied to the object as shown in Fig.4.29. The reference value for the pulling force is learned from experimental trials. The pulling robot is controlled using a Cartesian stiffness strategy:

$$\boldsymbol{\tau} = {}^{0}\mathbf{J}_t^T \left(\mathbf{k}_p \, {}^{0}d\mathbf{X}_t + \mathbf{f}_p\right) + \mathbf{H} \tag{4.62}$$

$$ {}^{0}d\mathbf{X}_t = \begin{bmatrix} {}^{0}d\mathbf{p}_t^T & {}^{0}\delta_t^T \end{bmatrix}^T \tag{4.63}$$

where $\boldsymbol{\tau}$ is the commanded joint torque, ${}^{0}d\mathbf{X}_t$ contains the differential position and orientation vectors, representing the error in Cartesian space, $\mathbf{H}$ denotes the system dynamics and $\mathbf{k}_p$ is the stiffness matrix.

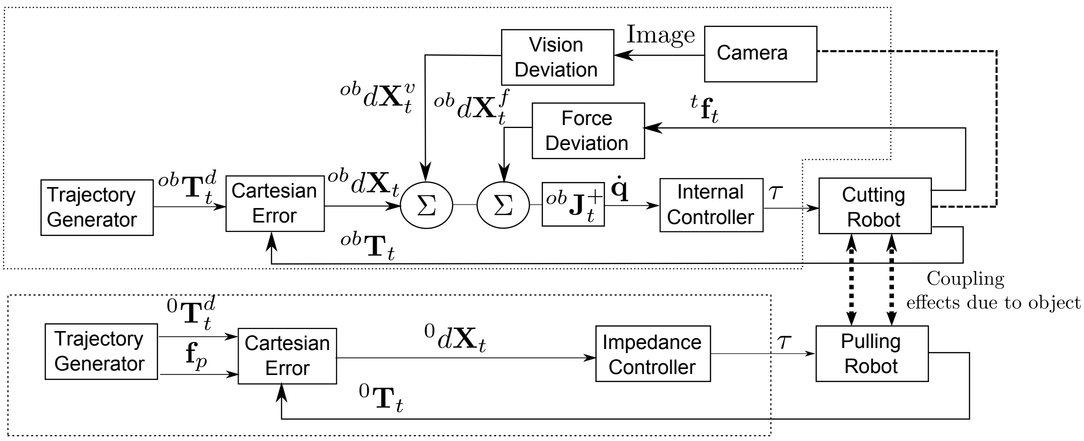

Figure 4.28: Global Control Scheme

Trajectory Generator

The trajectory generator for the experimental validation, follows that given in Section 4.3.5. However in this case, a more general approach is taken to obtain a desired cutting frame location at each instant.

The cutting robot follows a polynomial curve, $\mathcal{C}$ defined in the object frame $\mathcal{R}_{ob}$ in the form $y = a_n x^n + a_{n-1}x^{n-1} + \ldots a^0$. It is important that the cutting feed, i.e. the tangential velocity, remains constant. Thus the trajectory is defined in terms of the distance traveled along the curve defined as $\mathcal{T}(t)$ in (4.42).

p_x^d and p_y^d, the desired positions in the x direction and y direction respectively, are obtained from $\mathcal{T}(t)$ and $\mathcal{C}$. In order to complete the separation, the robot must cut the material along the curve at a desired depth denoted as p_z^d. This motion is known as a passage. At time t, as shown in Fig. 4.29, the knife's desired location in the object frame is given as ${}^{ob}\mathbf{T}_t^d = {}^{ob}\mathbf{T}_c\, {}^{c}\mathbf{T}_t$. In detail:

$$ {}^{ob}\mathbf{T}_t^d = \begin{bmatrix} \mathbf{t}^d & \mathbf{n}^d & \mathbf{a}^d & \mathbf{p}^d \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot {}^{c}\mathbf{T}_t(\theta) \tag{4.64} $$

where ${}^{c}\mathbf{T}_t(\theta.)$ is used to make the trajectory consistent with the cutting angle θ. The angle θ is fixed throughout the trajectory and should be chosen to ensure the camera has a good view of the trajectory close to the knife. The cutting angle is defined by a rotation around the y-axis of the tool frame. $\mathbf{p}^d$ the desired position is given as:

$$ \mathbf{p}^d = \begin{bmatrix} p_x^d & p_y^d & p_z^d \end{bmatrix}^T \tag{4.65} $$

The rotation matrix is calculated as:

$$ \mathbf{t}^d = \begin{bmatrix} \frac{1}{\sqrt{\left(1+\frac{\partial y}{\partial x}^2\right)}}, & \frac{\frac{\partial y}{\partial x}}{\sqrt{\left(1+\frac{\partial y}{\partial x}^2\right)}}, & 0 \end{bmatrix}^T \tag{4.66} $$

$$ \mathbf{n}^d = \begin{bmatrix} \frac{-\frac{\partial y}{\partial x}}{\sqrt{\left(1+\frac{\partial y}{\partial x}^2\right)}}, & \frac{1}{\sqrt{\left(1+\frac{\partial y}{\partial x}^2\right)}}, & 0 \end{bmatrix}^T \tag{4.67} $$

$$ \mathbf{a}^d = \begin{bmatrix} 0, & 0, & -1 \end{bmatrix}^T \tag{4.68} $$

As shown in Fig.4.29, $\mathbf{t}^d$ is the desired cutting direction, which is tangential to $\mathcal{C}$. $\mathbf{a}^d$ is the axis normal to the object's surface while $\mathbf{n}^d$ is the remaining orthogonal axis of the frame. $\frac{\partial y}{\partial x}$ is the value of $\frac{\partial y}{\partial x}$ evaluated at p_x^d.

Vision Controller

The vision controller updates the trajectory of the knife in response to on-line deformations by creating a deviation, denoted as ${}^{ob}d\mathbf{X}_t^v$ as follows:

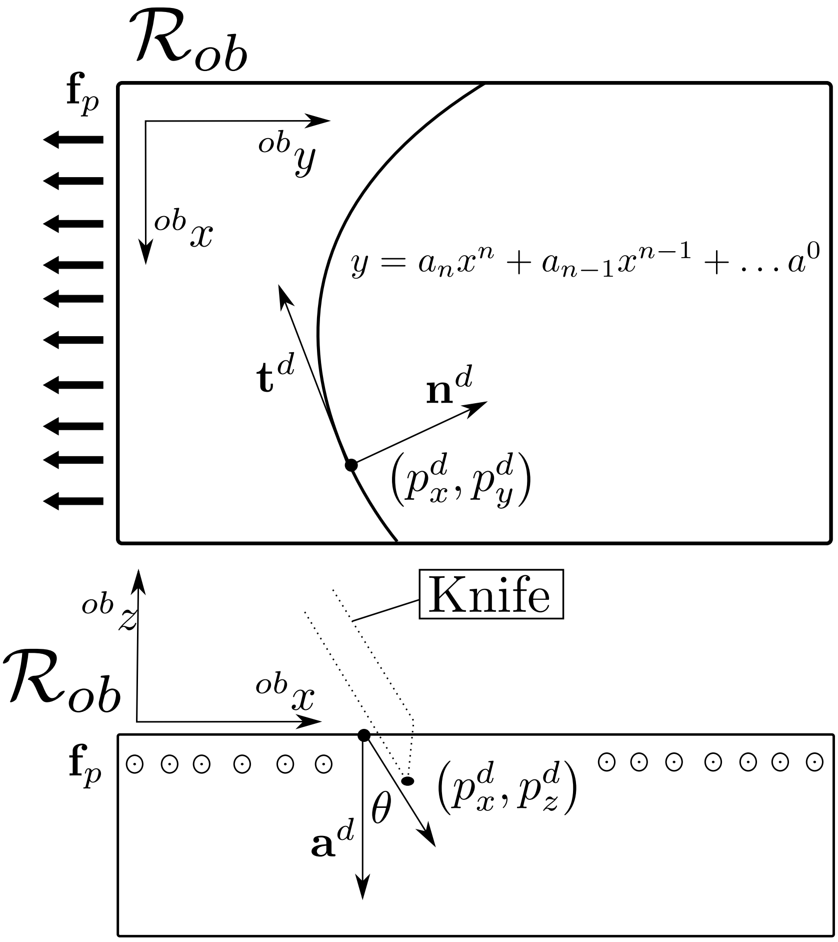

Figure 4.29: Definition of desired variables

1. The vision system extracts the image coordinates of (u_i, v_i), (u_j, v_j) and (u_k, v_k), a series of points ahead of the image projection of tool point
2. The normalized position of a point i is reconstructed using the intrinsic camera parameters, **C**, which relate the image coordinates to the coordinates in the perspective plane:

$$\begin{bmatrix} \frac{p_{xi}^v}{p_{zi}^v} \\ \frac{p_{yi}^v}{p_{zi}^v} \\ 1 \end{bmatrix} = \mathbf{C} \begin{bmatrix} u_i \\ v_i \\ 1 \end{bmatrix} \tag{4.69}$$

3. The depth of a point, p_{zi}^v, is estimated using the material height and the tool position, the depth estimation allows the reconstruction of the 3D position of the point

4. Since the camera gives a local view of the trajectory, the curvature within this window is quite small and can be approximated by a straight line. By fitting this line to the Cartesian position of points i, j and k the vectors $\mathbf{t}$ and then $\mathbf{n}$ are obtained
5. From step 3 and step 4 the desired matrix given by vision ${}^{ob}\mathbf{T}_i^v$ at a point i can be written

In order to generate an error vector, the curve $\mathcal{C}$ is evaluated at p_{xi} allowing a desired matrix ${}^{ob}\mathbf{T}_i^d$ to be obtained. This in turn is used to calculate the vision generated deviation which acts in one translational direction and three rotational directions:

$$\Delta p_{yi} = p_{yi}^d - p_{yi}^v \tag{4.70}$$

$$\Delta {}^{ob}\mathbf{R}_i = {}^{ob}\mathbf{R}_i^d \left({}^{ob}\mathbf{R}_i^v\right)^T \tag{4.71}$$

4.4.5 Validation of Force/Vision Controller using PBVS

In this section the cutting experiment using PBVS is described. The robot cuts the material until the resistive force grows in magnitude and a slicing phase is performed. If the cutting tool frame reaches a greater height than the object, the robot returns to the initial position to restart the passage.

Experimental Setup

The experimental setup is shown in Fig.4.23. A 200mm $\times$ 200mm $\times$ 100mm block of foam known as Bultex © is used. A series of dots, serving as the visual markers, are attached to the foam. The cutting trajectory is offset from these dots by a small distance to ensure the knife does not cut the visual marker.

Results

Each passage allows the robot to cut further and further along the trajectory. This gradual progression is seen in Fig.4.30, Fig.4.31 and Fig.4.32.

In Fig.4.30, the off-line estimation of the curve, the visually extracted curve and the robot position are shown. This graph shows that the desired trajectory is deformed due to the force applied by the pulling robot. The vision controller allows the robot to cut along the new trajectory. In Fig.4.30, Passage 20 shows that the cutting trajectory begins to resemble the initial estimation as the separation reaches its end. This is expected since $\mathbf{f}_p$ at this moment is applied to the offcut. Therefore as the cutting proceeds, the deformation effect due to $\mathbf{f}_p$ on the main part of the object is reduced, meaning the cutting trajectory returns to its original undeformed shape.

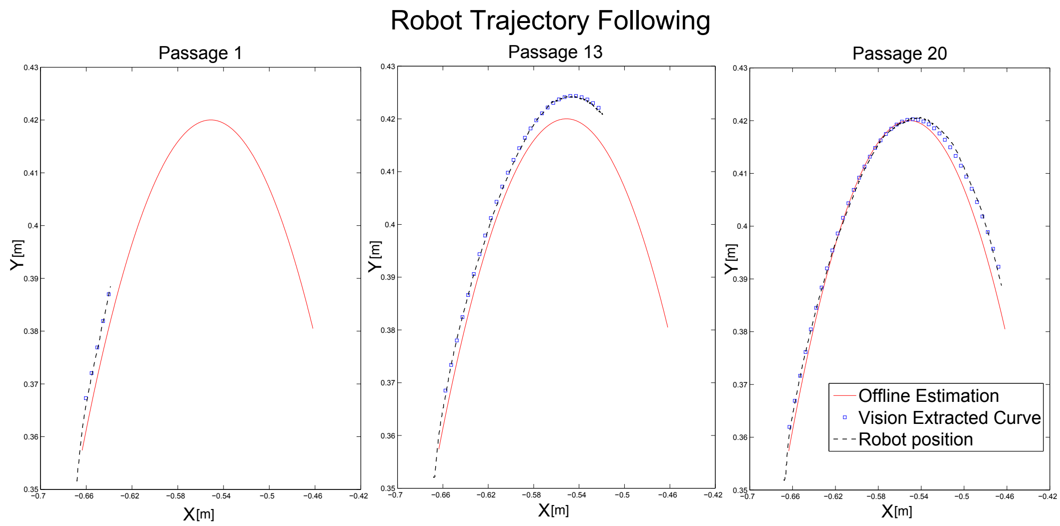

Figure 4.30: Displacement in the x-y plane of the object

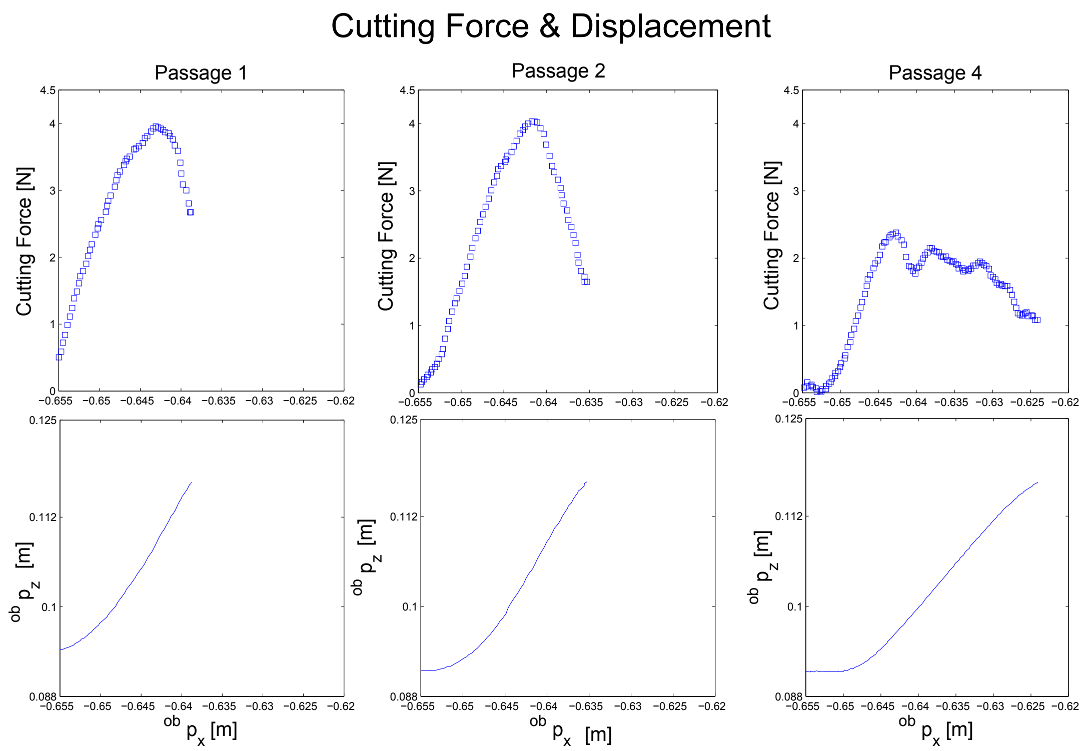

Figure 4.31: Cutting force and the displacement along the x and z axes during the initial cutting phase

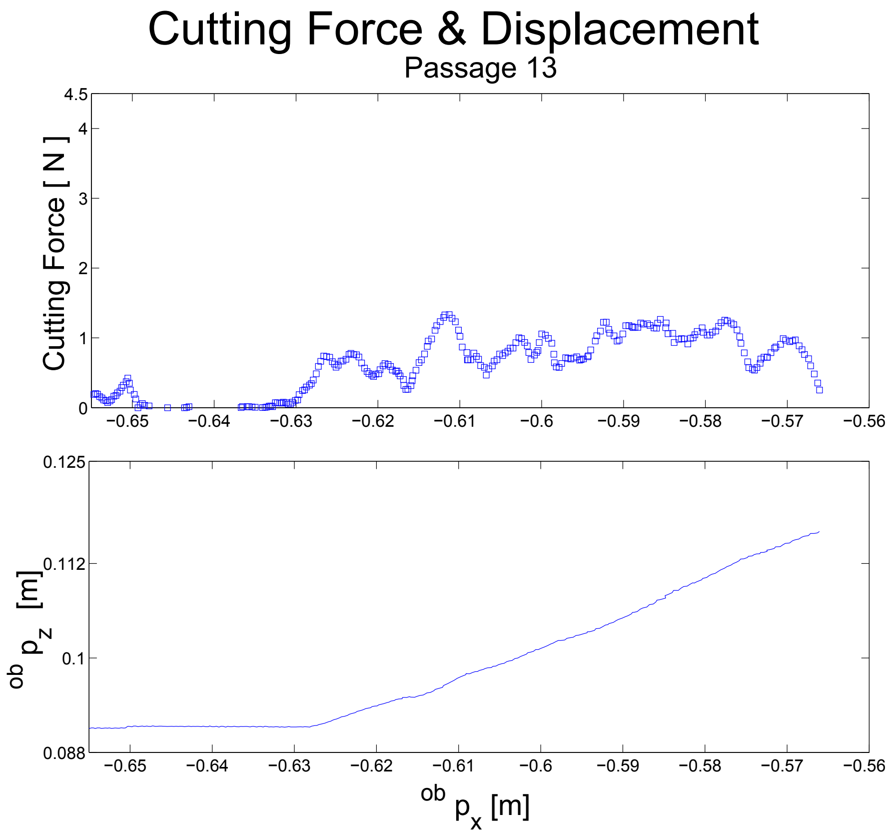

Figure 4.32: Cutting Forces and displacement along the x and z axes during the cutting propagation phase

Fig.4.31 and Fig.4.32 show the cutting force in the tool frame and the z position of the tool versus the progress along the curve during the crack initialization and propagation phase respectively. In Fig.4.31 the increase in the resistive cutting force causes the controller to create a slicing action which in turn results in a decrease of the force. During the crack propagation phase Fig.4.32 the magnitude of the resistive force has decreased. Finally in Fig.4.32, for the phases where cutting has taken place, the sensed force is close to zero demonstrating the absence of frictional forces due to the pulling robot.

4.4.6 Force/Vision Controller using IBVS

Outline

In Sections 4.4.4 and 4.4.5, it is shown how the vision system can be used to reconstruct the current trajectory during each passage. In this section, the control scheme is modified so that the tool is controlled entirely in the image space. The control of an image feature is known as IBVS as described in Section 4.2. In this case, the image feature can control all 6-DOF of the cutting tool. Furthermore the force signals, which are generated using the proposed cutting strategy outlined in Section 4.4.2, act directly on the desired image features. An overview of the image based robotic control strategy is given in Fig.4.33.

The merits of using IBVS, for instance robustness in the face of calibration errors, have been outlined in Section 4.2.1. The proposed force control using IBVS has also another advantage. It is shown in Section 4.2.2 that force/vision research has mainly focused on contour following tasks where a force is applied normal to the surface. Therefore if the surface is unknown, its pose with respect to the tool must be obtained. By studying the force sensors measurements and approximating the frictional effects of the motion, the normal to the surface can be constructed [HIA98, LLH06, PJ00], as shown in (4.19). The disadvantage of this approach is that the orientation of the tool relies on noisy force measurements and requires a filtering operation to obtain an accurate solution [XGXT00]. Moreover in cutting applications this approach is no longer valid, since in order to separate the object the tool must necessarily pass through the contour. In [Cha06], the normal to the surface is obtained using vision data by using an external camera observing the scene. However, these so called eye-to-hand systems are limited, notably due to problems regarding spatial resolution and limited field of view [PL03].

Tool Modification

The Kuka robot is used to perform the separation task, however the formulation of image based servoing task can be greatly simplified if the desired image is taken when the object is parallel to the image plane. Therefore the tool described in Section 4.4.1 must be modified. The new cutting tool is illustrated in Fig.4.34. It can be seen that the camera frame, denoted as $\mathcal{R}_c$, is parallel with the tool frame.

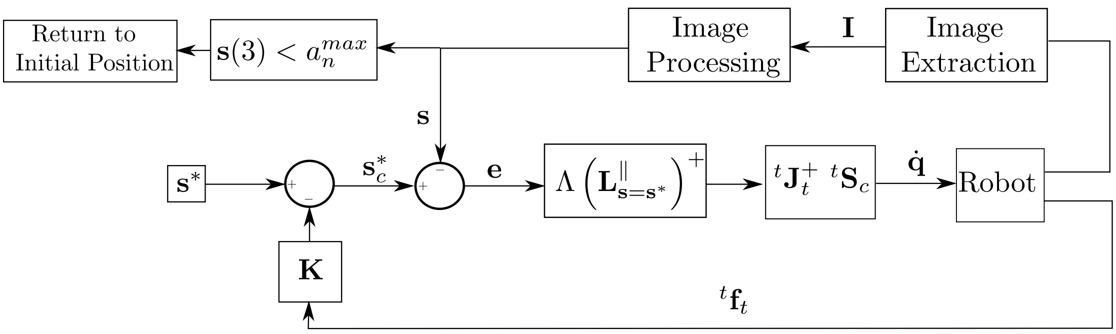

Figure 4.33: Global Control scheme

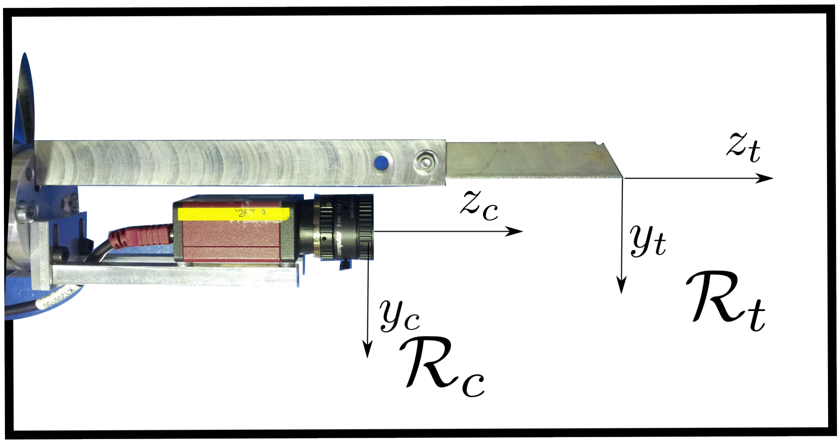

Figure 4.34: Cutting Tool for Image Based Cutting

Force Control by Image Features

As described in Section 4.4.3, the force controller is used to ensure that excessive resistive forces are avoided during to the cut. However, in contrast to the force controller proposed in Section 4.4.3, in this case it is proposed to link the resistive force directly to the desired image features as:

$$\delta a_n^* = \text{min}\left(0, \;\; k_z\, {}^t f_c\right) \tag{4.72}$$

The min function is used to prevent the knife entering deeper into the material. The image moment a_n^* is denoted as the normalized area of the image object. In the following sections, it will be shown why this image moment can be used to increase the *slice/press* ratio of the cutting action.

Image Based Visual Servoing

The cutting trajectory is composed of a series of dense objects located along an unknown curve that varies in three dimensions as shown in Fig.4.35. The visual controller is used to position all six degrees of freedom of the end effector by ensuring that the extracted image feature $\mathbf{s}$ converges to a desired image feature $\mathbf{s}^*$. The desired image specifies the reference cutting frame, including the desired depth and the cutting angle. The camera velocity is found from:

$$ {}^c\mathbf{V}_c = \boldsymbol{\Lambda}\mathbf{L}_s^+\left(\mathbf{s}^* - \mathbf{s}\right) \tag{4.73}$$

where $\mathbf{L}_s$ is the interaction matrix. ${}^c\mathbf{V}_c$ is the camera velocity composed of a 3×1 vector of linear velocity ${}^c\mathbf{v}_c = \left[\begin{array}{ccc} v_x & v_y & v_z \end{array}\right]^T$ and a 3×1 vector of angular velocity ${}^c\boldsymbol{\omega}_c = \left[\begin{array}{ccc} \omega_x & \omega_y & \omega_z \end{array}\right]^T$. $\boldsymbol{\Lambda}$ is an adaptive gain matrix.

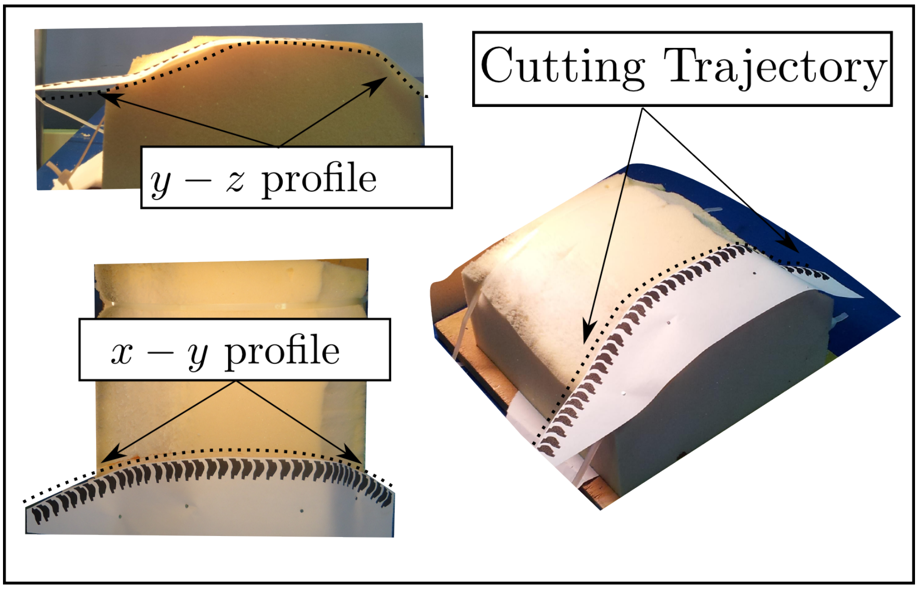

Figure 4.35: Target Object with Cutting Trajectory

The flow diagram in Fig.4.36 describes the steps taken to ensure the knife follows a trajectory rather than simply converging to a desired pose. The control scheme selects an object and binarizes the image such that the pixels within the objects boundaries are given an intensity value $I = 1$. The remaining part of the image, including other objects in the field of view, are given a value of $I = 0$. Once the object has converged to the desired position, the next object on the trajectory is selected.

Image Moments Image moments provide a robust description of a diverse set of image objects and have been widely used in computer vision tasks [Hu62]. Recently image moments have been applied to visual servoing schemes. By careful combination of image moments, an interaction matrix of maximal decoupled structure and low condition number can be built for a variety of complex images [Cha04, TC05, CBL11].

Image moments can describe a diverse shape of objects after segmentation or binarization. For a binary image, with pixel coordinates (x, y), where the object of interest has intensity $I(x, y) = 1$, the raw moments are defined by:

$$m_{ij} = \sum_{k=1}^{n} x_k^i y_k^j \tag{4.74}$$

The central moments are defined as :

$$\mu_{ij} = \sum_{k=1}^{n} (x_k - x_g)^i \, (y_k - y_g)^j \tag{4.75}$$

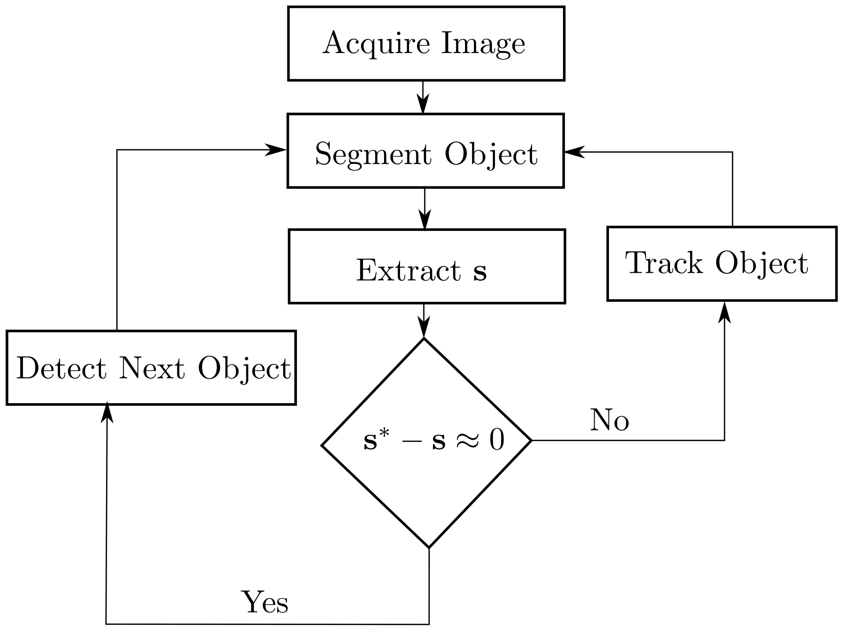

Figure 4.36: Image Segmentation Process

where x_g and y_g are defined as the center of gravity of the object:

$$x_g = \frac{m_{10}}{m_{00}}, \qquad y_g = \frac{m_{01}}{m_{00}} \tag{4.76}$$

The image moments are selected to ensure that both the linear and angular velocity of the camera can be controlled. Furthermore they are chosen such that the condition number of the resulting interaction matrix is low. In order to control the cutting tool the following image moments are selected:

$$\mathbf{s} = \begin{bmatrix} x_n & y_n & a_n & \tau_x & \tau_y & \alpha \end{bmatrix} \tag{4.77}$$

The first three components are known as the normalized coordinates of center of gravity and the normalized area [TC05]. These components are used to control the linear velocity of the camera, they are defined as:

$$x_n = a_n x_g \qquad y_n = a_n y_g \qquad a_n = z^* \sqrt{\frac{a^*}{a}} \tag{4.78}$$

where a^* and z^* represent the desired area and the distance between the object and the camera in the desired configuration respectively. The angular velocity, ω_z, about the optical axis of the camera is controlled by the component α. This feature is calculated using the central moments from (4.75) as:

$$\alpha = \frac{1}{2}\left(2\frac{\mu_{11}}{\mu_{20} - \mu_{02}}\right) \tag{4.79}$$

Finally, in order to control the velocities ω_x and ω_y, a careful selection of the image moments τ_x, τ_y must be carried out with respect to the task, the object and the desired object. Referring to Fig.4.34 and Fig.4.35, it can be seen that during the cutting trajectory, the camera must undergo large rotations around its x axis in order to keep the object parallel to the image plane the camera. Therefore, an off-line selection method proposed in [Tah04] is used to obtain a satisfactory set of image moments. For any set of image moments (c_i, c_j), the following error is computed:

$$e = (c_i - c_i^*)^2 + (c_j - c_j^*)^2 \tag{4.80}$$

where c_i, c_j are the values of the image moment at the current camera configuration and c_i^*, c_j^* are the values of the image moment at the desired configuration. By calculating (4.80) for camera rotations around the x_c and y_c axes, a set of image moments can be obtained that will allow error convergence in spite of these rotations. By using this procedure, the following pair of image moments are selected:

$$\tau_x = \frac{I_3}{I_4} \qquad \tau_y = \frac{I_{14}}{I_{15}} \tag{4.81}$$

I_3, I_4, I_{14} and I_{15} are four invariant image moments given in [Tah04] calculated using (4.74) and (4.75) and based on those originally obtained in [Hu62]:

$$I_3 = (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2 \tag{4.82}$$
$$I_4 = (\mu_{30} + \mu_{12})^2 + (\mu_{21} + \mu_{03})^2 \tag{4.83}$$
$$I_{14} = (\mu_{50} - 2\mu_{32} - 3\mu_{14})^2 + (\mu_{05} - 2\mu_{23} - 3\mu_{41})^2 \tag{4.84}$$
$$I_{15} = (\mu_{50} - 10\mu_{32} + 5\mu_{14})^2 + (\mu_{05} - 10\mu_{23} + 5\mu_{41})^2 \tag{4.85}$$

τ_x and τ_y are invariant [Cha04] to translational motions and to rotations around the optical axis.

Desired Image Feature

The desired image feature $\mathbf{s}^*$ is obtained off-line using a *teaching by showing* method. In order to achieve this, the knife is moved to a desirable cutting location, which is normal to the object and at the desired cutting depth. At this location, an image of the scene is taken and the value of the image moments are extracted. It is important to note that $\mathbf{s}^*$ is constant, therefore even though the object is flexible, the marker, which is adhered to the surface, does not deform.

Interaction Matrix

As shown in (4.73) an interaction matrix relates the image features' error to the camera's kinematic screw. The chosen interaction matrix is computed from the desired image, $\mathbf{s}^*$ where the desired object is parallel to the image plane. By ensuring the desired image is parallel to the image, the interaction matrix is greatly simplified.

The use of a constant intersection matrix, computed using the desired configuration, is justified since the trajectory consists of a series of objects very close to each other.

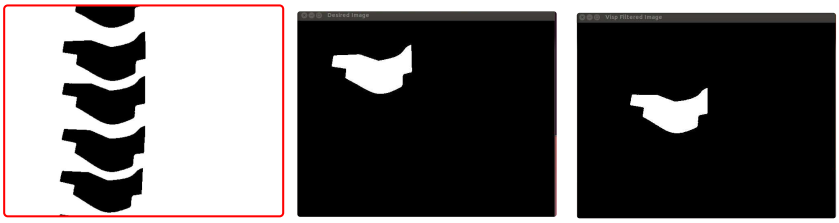

Figure 4.37: (a) Extracted scene at time t, (b) Desired Image, (c) Current Image

Thus the current object is unlikely to be outside the localized area of convergence. By making the above simplifications an interaction matrix is obtained with the following form:

$$\mathbf{L}^{\|}_{\mathbf{s}=\mathbf{s}^*} = \begin{bmatrix} -1 & 0 & 0 & x_{n\omega x} & x_{n\omega y} & y_n^* \\ 0 & -1 & 0 & y_{n\omega x} & y_{n\omega y} & -x_n^* \\ 0 & 0 & -1 & a_{n\omega x} & a_{n\omega y} & 0 \\ 0 & 0 & 0 & \tau_{x\omega x} & \tau_{x\omega y} & 0 \\ 0 & 0 & 0 & \tau_{y\omega x} & \tau_{y\omega y} & 0 \\ 0 & 0 & 0 & \alpha_{\omega x} & \alpha_{\omega y} & -1 \end{bmatrix} \tag{4.86}$$

The analytic expressions of the quantities expressed in (4.86) can be found in the following [Cha04, TC05], however the decoupled structure of the matrix is more important.

A particularly interesting feature is the direct link between the distance to the object i.e. the cutting depth and the area of the image object. This relationship is exploited by the force controller as described in (4.72). Any resistive force, changes the desired value of the normalized area, thus generating a slicing motion of the tool.

4.4.7 Validation of Force/Vision Controller using IBVS

In this section the cutting experiment using IBVS is described. The control scheme is designed to move the camera such that the next image in the trajectory sequence converges to the desired image, as illustrated in Fig.4.37. If the tool exits the material, the robot returns to the initial position. In order to overcome the discrete nature of the trajectory, an adaptive gain matrix $\boldsymbol{\Lambda}$ in (4.73) is used to increase the rate of convergence as the error decreases.

Experimental Setup

The experimental setup is shown in Fig.4.35. A $200mm \times 200mm \times 100mm$ block of foam known as Bultex © is used. The foam is pre-cut to create an irregular surface. A series of identical objects are adhered to the foam in a curvilinear configuration. The curve varies in the x, y and z directions. The desired image is shown in Fig.4.37. At the desired configuration, the object is parallel to the image plane and the distance is given as $z^d = 0.08m$. The numerical value of the interaction matrix is given as:

$$\mathbf{L}_{\mathbf{s}=\mathbf{s}^*}^{-1} = \begin{bmatrix} -1 & 0 & 0 & 0.016 & 0.071 & 0.007 \\ 0 & -1 & 0 & 0.030 & -0.297 & -0.008 \\ 0 & 0 & -1 & 0.006 & -0.028 & 0 \\ 0 & 0 & 0 & 0.385 & -3.687 & 0 \\ 0 & 0 & 0 & -0.194 & -0.867 & 0 \\ 0 & 0 & 0 & -0.044 & -0.056 & -1 \end{bmatrix} \quad (4.87)$$

while the value of the desired image feature is given as:

$$\mathbf{s}^* = \begin{bmatrix} -8.04 & -7.0 & 80 & 11067 & 1038.0 & 38.556 \end{bmatrix} \times 10^3 \quad (4.88)$$

Experimental Results

Fig.4.38 gives the evolution of the camera position in the $y - z$ plane, the surface profile and the orientation of the optical axis. Since the convergence of an image feature does not correlate with a straight line trajectory in Cartesian space, at times the camera performs motions that are contrary to cutting requirements. However, it can be seen that the orientation of camera's optical axis is adjusted throughout the trajectory in order to keep the object parallel to the image plane.

Fig.4.39 shows the resistive force due to the cutting action. As the force increases the controller induces a change in the desired normalized area as described by (4.72). This slicing action then results in a drop in the magnitude of the force. This is particularly clear for passage 10. The decoupling structure of the interaction matrix means that the slicing action does not perturb the convergence of the other image primitives.

Fig.4.40 shows the error of the image moments for two consecutive objects. It can be seen that all the image features converge to their desired values. The spikes in the graphs at iteration number 2300 and 3400 indicate the detection of the next object in the trajectory and thus an instantaneous change in the value of **s** as described in Fig.4.36. It is also clear from Fig.4.40 that the convergence of τ_x and τ_y is much slower than the other image features.

Finally 4.41 shows the commanded camera velocity i.e. the control signal. The close relationship between the image primitives and the camera velocity is clear when a comparison is made between Fig.4.40 and 4.41. However, the discontinuity still exists in the control signal.

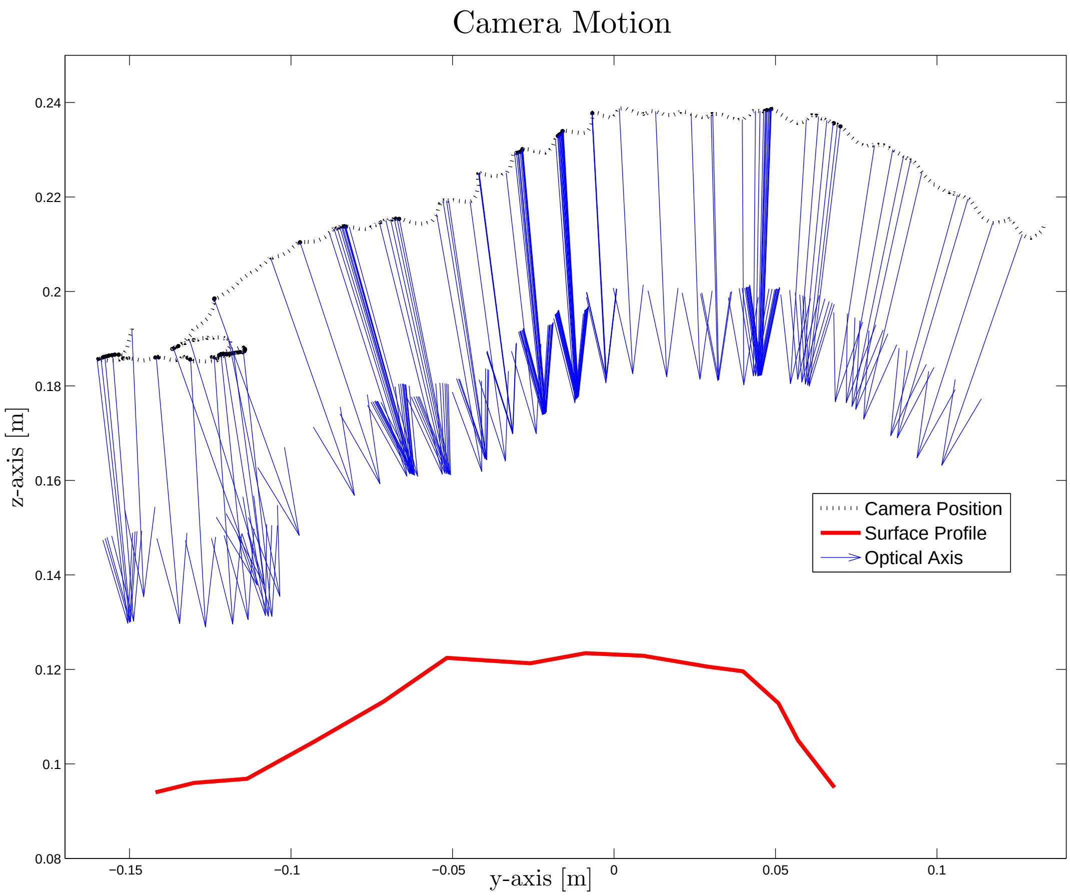

Figure 4.38: Camera Trajectory with optical axis in the $y - z$ plane in the fixed world frame. This graph shows how the controller can rotate in order to ensure the camera remains normal to the surface profile

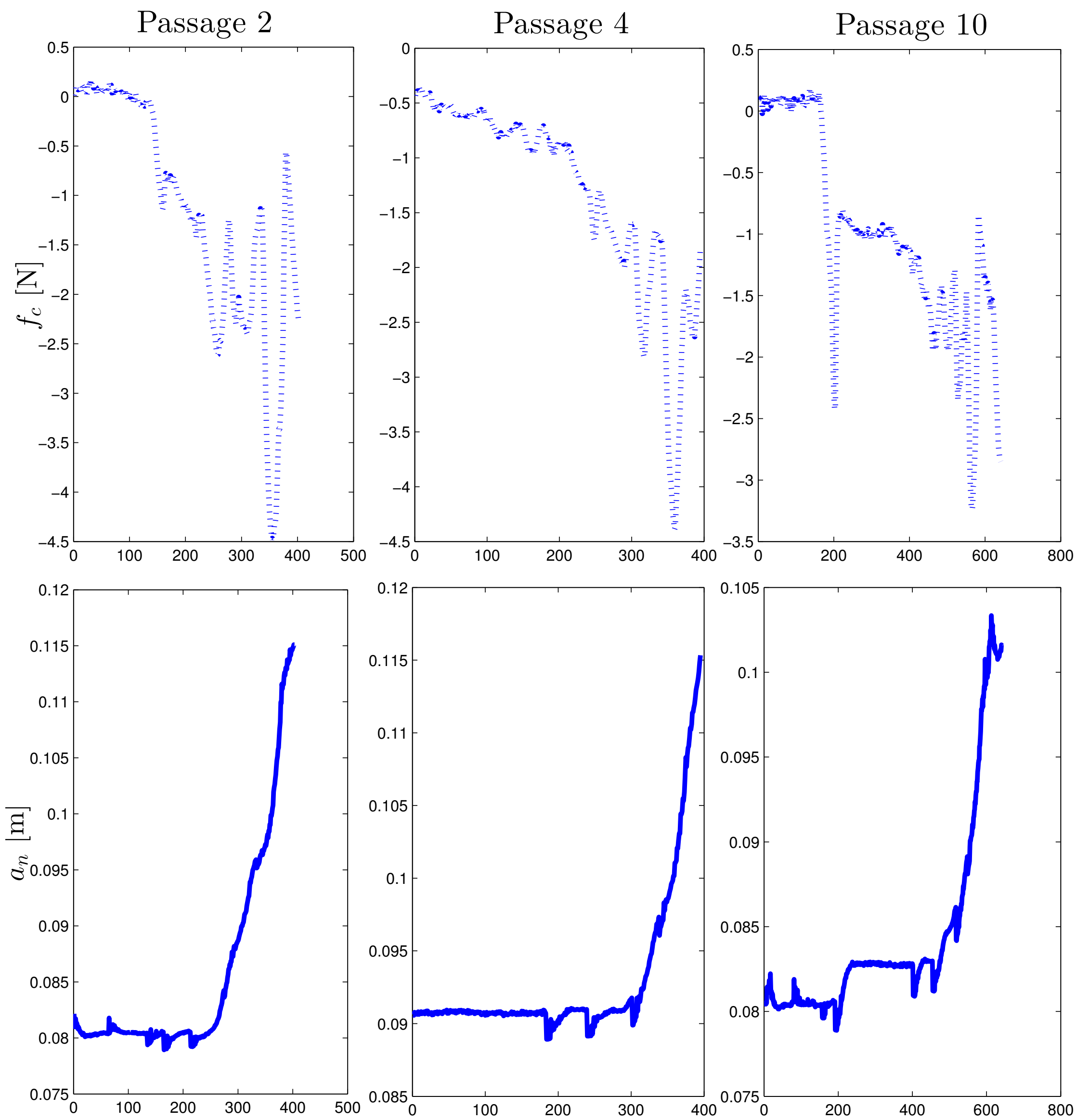

Figure 4.39: Deviation due to resistive cutting force. Due to the decoupling effect of the image moments on the interaction matrix, the slicing action can be directly applied to image space. The bottom panel shows the desired normalized image area. The top panel shows the resistive cutting forces that alter the desired value for the area.

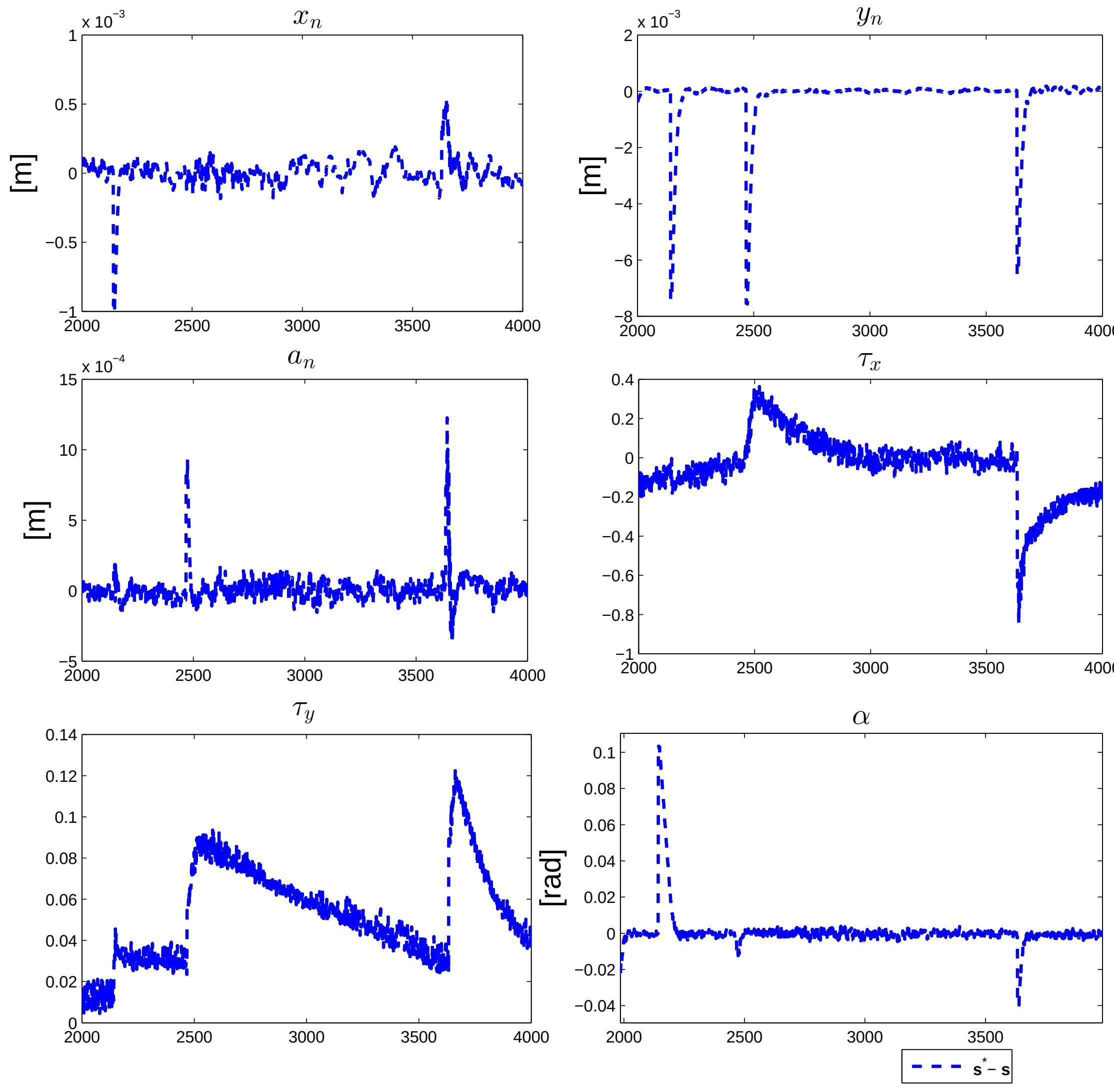

Figure 4.40: Error of image moments during point to point trajectory. This graph shows the convergence of two markers to the desired position.

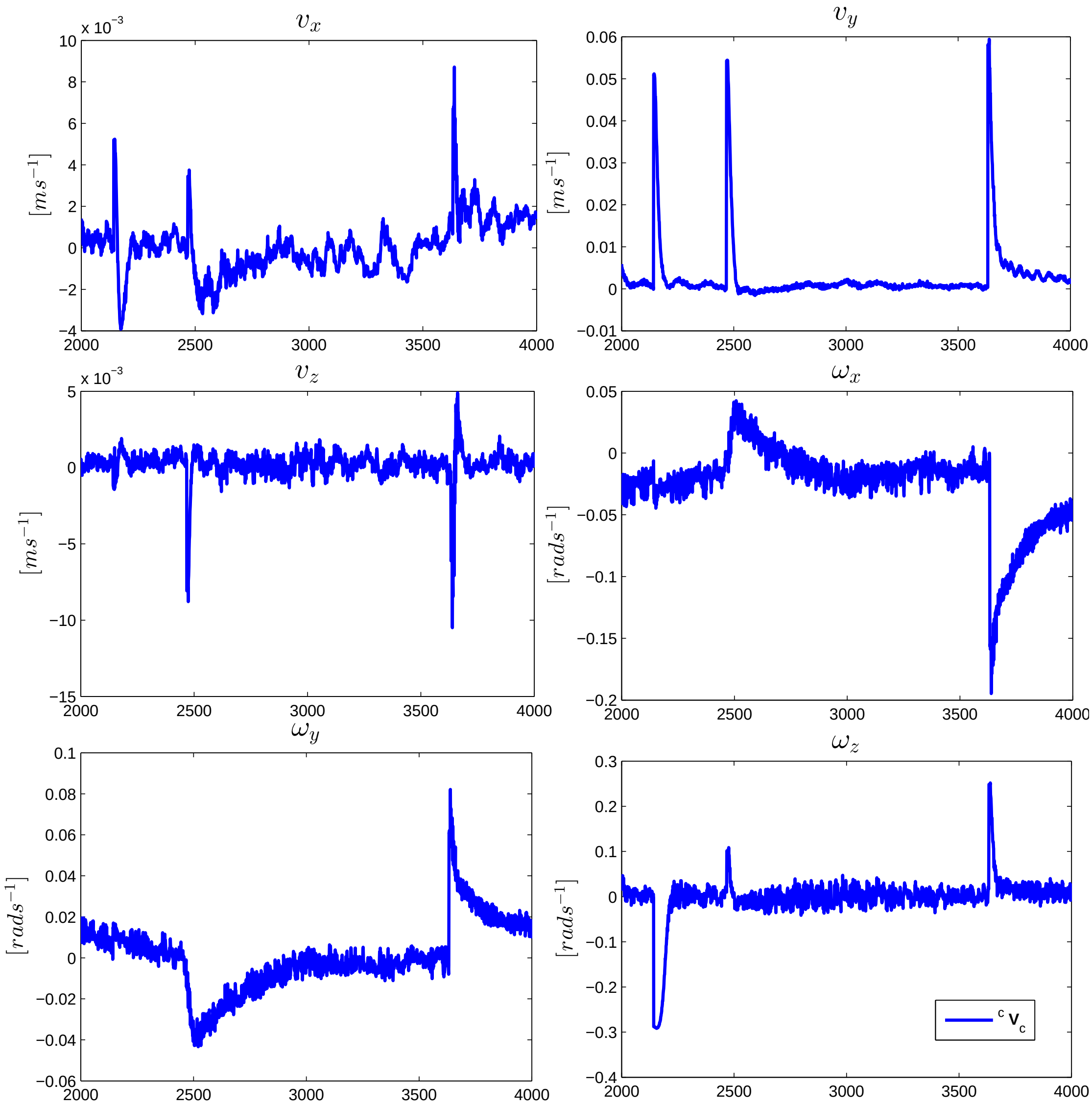

Figure 4.41: Commanded velocity of camera during point to point trajectory. The six graphs show the commanded velocity component of the camera kinematic screw represented in the camera frame

4.5 Conclusions

This chapter has focuses on robotic cutting tasks. A review of the state of the art of the techniques required for the separation of deformable objects is presented. Three areas of research are particularly pertinent to this task, the use of exteroceptive sensors to allow greater flexibility when dealing with unknown objects, the simulation of deformable objects and a mathematical model of cutting. Control formulations that exploit force and vision sensors are described. These schemes can be classified according to how the signals are combined i.e. partitioned or shared and the nature of the visual servoing scheme whether it is IBVS or PBVS. By using force and vision sensors the manipulator can interact safely with a dynamic environment. The primary methods of simulating deformable objects are given. By constructing an object model, robotic control schemes can be tested with realistic behavior. Furthermore force and visual feedback can be taken from the resulting representation. Finally, the cutting action is examined. The formulations that describe the resistive force at the cutting tool frame are outlined. It is shown that a shearing action at the cutting edge decreases the resulting resistive force.

The modeling, simulation and control of a robotic meat cutting cell is described. The contribution of this section is the construction of an advanced simulator and the validation of control schemes that use the local update methodology. The modeling process for the robot, the meat, visual primitives and the interaction between modules is outlined. A control scheme is proposed using visual and force data to cope with uncertainties about the object behavior. The results show how, both local and global visual primitives, can be used to compensate for object deformations. In particular, it is shown that large deformations require a fast local system in order to re-adjust the robot's motion on-line. Future work will concentrate on the construction of a centralized multi-point controller. This controller will treat the robots as one system that will allow an advanced redundancy resolution scheme to take into account secondary tasks such as collision avoidance and field of view optimization. This type of centralized controller is important when multiple robots are functioning in close proximity. Finally, it should be noted that the experimental validation of this multi-arm control scheme is pending the installation and integration of the ARMS project robotic cell. The validation of this strategy for the meat cutting cell is an important future work.

A force/vision controller is proposed and validated for the separation of deformable objects. The contribution of this section is the design of a new force controller for cutting applications. Furthermore this controller is combined with two visual servoing system, to allow the robotic system to cut along deformable trajectories. Finally, a novel force/image controller is proposed and validated that allows the system to cut along unknown 3D surface profiles. The proposed solution can also be applied to many generic contour following tasks. The estimation of a contour using this method is an important future work.

The adaptive force controller is proposed such that an increase in the resistive force generates a slicing motion thus avoiding any rupture or global deformation of the object. A series of experiments have been carried out to evaluate the performance of this controller. In summary, these test prove that the proposed force controller performs much better than a position controller. It would be possible, by changing the cutting angle and velocities, to construct an effective cutting controller without a force feedback as shown by the increase in performance of all control laws as θ increases. Indeed, previous researchers have focused on optimizing cutting parameters for specific materials. However by using the resistive force as an input to induce a slicing motion, it has been shown that this costly step can be avoided. Future work is focused on removing the arbitrary selection of the gain value in the force controller.

Two different visual servoing control laws are combined with this force controller, a PBVS and an IBVS. In the following, the possible avenues of future research are discussed for the two controllers.

By using this PBVS system a clean and efficient cut can be performed in spite of object deformations. Furthermore a desired cutting velocity can be set off-line allowing for a very accurate force control with a reasonable cutting speed.

However the PBVS control scheme requires an estimate of the trajectory. This estimate allows the calculation of the curve parameters and thus the formulation of a trajectory that results in a constant cutting speed. Furthermore information about the material such as the height and surface profile are required. The controller needs knowledge of the material height in order to reset the trajectory once the knife has left the material. In future work this requirement could be eliminated by using a more precise force controller. Finally, it should be noted that the scheme is very sensitive to calibration errors and thus requires a painstaking and precise calibration of intrinsic and extrinsic camera parameters.

On the other hand IBVS, allows the cutting tool to follow a deformable three dimensional surface while maintaining a constant cutting depth. By controlling the system using IBVS and by adding the force deviation directly to the image space, the sensitivity to calibration errors is reduced. Another advantage of the IBVS with respect to previous works, is the ability of the local vision system to control the angle between the surface and the cutting tool and hence eliminating the effect of noisy force measurements on the orientation task. In addition to this, by using a local camera instead of an external system, a clear unimpeded view of the desired trajectory is provided. For the separation of deformable objects, a local view is essential due to the deformable nature of object.

However the IBVS cutting scheme has several drawbacks. Firstly, unlike the PBVS scheme, discrete objects are used to represent a continuous trajectory, this means that the convergence of the image features may lead to an unsatisfactory cutting trajectory in the Cartesian space. This discontinuity can be seen both in the feature convergence shown in Fig.4.40 and in the control signal given in Fig.4.41. As the tool is embedded in the material, during the cutting task this could lead to unnecessary forces. In future work, two possible solutions to this problem will be tested. Firstly, by increasing the resolution of the camera so that the discrete objects better approximate a continuous trajectory. Secondly, the value of the future marker could be taken into account in the controller. For instance suppose the cutting trajectory is defined by a series markers defined as $1 \ldots i,\ i+1 \ldots n$ where the value of marker i is defined as $\mathbf{s}_i$. The IBVS cutting scheme defines the image errors as $\Delta\mathbf{s} = \mathbf{s}^* - \mathbf{s}_i$. To reduce the effect of the discontinuous change in marker it may be appropriate to take into account the value of the next marker therefore the image error would be defined as $\Delta\mathbf{s} = \mathbf{s}^* - (\delta\mathbf{s}_i + (1-\delta)\mathbf{s}_i)$. Alternatively this solution could be applied directly to the control signal such that $\mathbf{V} = \delta\mathbf{V}_i + (1-\delta)\mathbf{V}_i$. Another drawback of the IBVS is the variability of the cutting speed, since each image converges exponentially to a desired image. The effect of the exponential control law on the cutting feed is mitigated in the above work by using the adaptive gain matrix $\mathbf{\Lambda}$ from (4.73). Nevertheless there is still a variation of the control signal that can be seen in Fig.4.41. This can complicate the relationship between the force controller and the velocity controller. This is demonstrated by the fact that the effects of the force controller shown in Fig.4.39 are not as obvious as those of the PBVS case given in Fig.4.31 and Fig.4.32. Finally, the IBVS scheme assumed that the image marker remains constant. This is reasonable for the case of an external marker adhered to a surface. However, if the image moment is extracted directly from the object, the deformation must be known and taken into account in the definition of $\mathbf{s}^*$.

The work presented in this paper has led to the publication of three papers presented at international conferences [LKM13], [LKM14b], [LKM14c] and two international workshops [LMKM13] [LKM14d].

5

General Conclusions

This work has focused on cooperative manipulation in robotics. The tasks range from closed chain manipulation of a rigid object to cooperation between serial robots to achieve a common objective. The work has been validated using simulations and where possible using experimental techniques. The contributions of this thesis are in the area of cooperative modeling and control. In the following, a description of the contributions of each topic is given.

Chapter 2 focuses on the rigid object manipulation by cooperative manipulators. The work presented in this chapter has led to the publication of two papers presented at international conferences [LKC12a] [LKC12b] and one journal article [LKC14]. The main contribution of this chapter is the methodology of actuation scheme selection for a class of lower mobility cooperative manipulators. A method to analyze dual arm robots transporting rigid objects is proposed. This method demonstrates the advantages of screw theory when examining the object's behavior at singular configurations. A comparison between the numerical and the geometric methods gives an insight into their equivalence but also the difficulty in judging object behavior at singular configurations using only the serial Jacobian matrices. In addition to this, the importance of considering the system's dynamic when selecting an actuation scheme is shown. In particular, it is demonstrated that a judicious choice of actuation scheme can result in a large difference in the power loss during trajectory following.

Chapter 3 focuses on flexible object manipulation by closed chain systems. The work presented in this chapter has led to the publication of one journal paper [LKM14a] and one conference paper [LKM15]. In Section 3.3, a general algorithm is given that derives a closed form solution for the inverse and direct dynamic model of a closed chain robot with flexible end effector. The approach consists in dividing the system into a rigid and flexible sub-systems. Each sub-system is solved independently and then linked by the constraint wrenches at the grasp location. The general formalism is derived for multiple robots of different structure and grasping conditions. Furthermore, a method of object identification is proposed. This method analyzes the object type as a part of a closed robotic chain. The formalism is validated by a numerical comparison with a commercial simulation software for three different object types.

In Section 3.4, the dynamic modeling strategy is applied to a specific parallel manipulator, the Gough-Stewart manipulator with flexible platform, in order to derive the closed form dynamic model. The resulting model is validated by comparison with a commercially available simulation software. This algorithm has applications for mass reduction in machine tools and for cooperative manipulators transporting a common deformable object.

Chapter 4 focuses on the robotic separation of deformable objects. This sensor based approach is preferable to the model based approach derived in Chapter 3, due to the variability of the meat muscles in the industrial process. The work presented in this chapter has led to the publication of three papers presented at international conferences [LKM13], [LKM14b], [LKM14c] and two international workshops [LMKM13] [LKM14d]. The contributions of this chapter are divided into two sections.

Section 4.3, the modeling and control of a robotic meat cutting cell, consisting of three robots, is described. The main contribution of this section is the construction of a multi-robot dynamic simulator that can interact with deformable objects. The simulator is used to test a control scheme designed for the separation of meat muscles. We have shown how this controller allows the multi-arm system to compensate for the non-linear object motion despite the lack of the deformation model. In addition this the desired behavior, that of a gradual opening of the cutting valley to ensure a deeper cut, is exhibited. Finally, the benefits of a local vision based update are illustrated.

Section 4.4 focus on the experimental validation of a series of cutting controllers. In this section a new adaptive force controller is proposed to cut a soft material. This controller ensures that global deformation and material *bunching* can be avoided by linking the resistive force to an increase in the *slicing/pressing* ratio. This meant that the force along the cutting axis generates a deviation in position in an orthogonal axis. We have shown how this control law performs better with respect to purely position based cutting. Moreover, the change in performance with different control law parameters is shown. The force controller is combined with two different visual servoing systems so that the system can adapt to on-line deformations of the soft material. Both vision systems are validated experimentally on soft foam material. The combination of force and image control to follow 3D surface profiles has many applications for contour following tasks.

Future Work

There are several possible avenues for research stemming from this thesis concerning cooperative manipulation and control schemes. For the sake of clarity, these will be described for each part of the thesis.

Cooperative Manipulation of Rigid Objects

A possible future work in this domain would be the development of a switching mode control scheme. The kinematic and dynamic analysis has shown the configuration based performance of various actuation schemes. A control scheme could be constructed that effectuates a switch in independent actuators at different points during the trajectory. By doing so, singularities could be avoided and the optimum dynamic performance could be attained. Switching based controllers suffer from problems of discontinuity and thus this would also need to be taken into account for any controller.

Another potential for future work is to apply the same techniques to a deformable object. In this case the constraint wrench applied on the object by both arms may change the object's form. Therefore an interesting application would be to apply shape control techniques using these constraint wrenches. This means that by controlling the wrenches on the object a desired deformation form may be reached.

Finally, this current work could extended to take into account the case of redundant actuation. Although redundant actuation increases the complexity of the resulting control scheme, it allows a range of secondary criteria to be satisfied and therefore should be considered. For example, there may be scenarios where internal forces are desired. These tasks can be achieved by the redundant actuation.

Cooperative Manipulation of Flexible Objects

Future work will in this case focus on several principal objectives. Firstly, to derive the dynamic model for a completely flexible parallel manipulator. In this case, the flexibility is present not only in the platform but also in the legs. In order to solve this problem, the closed form solution presented in **Section** 3.4 must be combined with previous techniques that model leg flexibility, notably with [BK13].

This problem leads to the second possible avenue of future research to rewrite the Cartesian Dynamic Model to obtain a form that is used classically in parallel robots. For example, consider a comparison between the model derived in **Section** 3.4 and the general dynamic model for a parallel robot given [KI07]. For the model of the parallel manipulator with flexible platform all the terms are calculated in the Cartesian space, using the Cartesian space variables whereas, in the classical formulation for purely rigid robots the force on the joints are due to the platform forces and those due to the leg's dynamics. It would be interesting, in future work to see if the model of the Gough Stewart manipulator with flexible platform can be decomposed along the same lines.

Future work should focus on the experimental validation of this algorithm. This could be carried out for a set of cooperative manipulators transporting a flexible object. Since the cooperative manipulators holding a common object are typically redundantly actuated, the supplementary actuators could be used to reduce the vibration of the object by minimizing the generalized elastic variables. It would be extremely interesting and valuable if such a scheme could be experimentally verified, since there would be many applications in high speed machining tasks, where the induced vibrations require extra mass to be added to the platform.

From the algorithm, a method is defined that can classify an object grasped by multiple serial chains. This method is based on the structure of the matrix $\mathbf{W}_s$ which relates the motor torques to the Cartesian variables. An interesting future work would be the identification of this matrix on-line for objects of unknown composition.

Force/Vision Control of a Meat Cutting Robotic Cell

The simulator presented in this chapter uses a decentralized controller for the multi-robot system. In its current form the redundancy resolution scheme is used simply to damp unnecessary null space motion. Therefore in future work, the simulator will be used to test centralized redundancy resolution schemes, notably using the new projector proposed in [AM13]. In total the three robots contain 21-DOF (7 DOF per robot), yet the task is only of dimension 16 (6 for the cutting task, 6 for the pulling task and 4 for the vision task), therefore the system is *task* redundant. However, during the meat cutting tasks there are an array of sub-tasks which should be executed, for example collision avoidance, energy minimization, occlusion avoidance, singularity avoidance, joint limit avoidance, field of view optimization etc. Taking these supplementary constraints into account means that a redundancy resolution strategy is essential. The simulator is therefore an ideal platform to test redundancy resolution methods and task planning algorithms. The objective is to show the benefits of a coherent redundancy resolution scheme and of using a centralized control algorithm for a set of cooperative manipulators.

The experimental work for this thesis was carried out using the two Kuka robots in IRCCyN. In future work experimental validation will be carried out on the ARMS meat cutting cell, which consists of two Adept Viper robots and one Kuka lwr robot. This cell is pending completion, therefore once finished, future work will focus on the experimental validation of the separation strategy. In this case, instead of using a local vision system, an advanced perception system is fixed to a third robot that is used to locate the cutting trajectory. Furthermore an object model will be integrated into the control loop. The advantage of having an object model is threefold. Firstly, if the model updates significantly faster than the vision system, which is likely considering the complexity of visually tracking a deformable object, the system can use the object model as an indicator of trajectory location between vision samples. This is preferable to slowing down the overall speed of the controller. Secondly, the object can supplement the vision data in the case of occlusions. Finally, by using an object model, the value of the pulling force and other variables in the system can be obtained directly rather than through trial and error.

For the adaptive force controller, we have seen how the *pressing/slicing* strategy can prevent global deformations and reduce resistive forces yet there are possible areas of improvement. The main drawback of this method is the use of the gain terms that relates the cutting force to the slicing deviation. These gains are user inputs; thus some knowledge of the material properties is necessary. Future work will investigate if there is a method to learn or adapt these gains on-line. One promising idea is to exploit the derivative of the force term since an abrupt change of force may signal the beginning of a global deformation in the object (that leads to damage to the surrounding gain and to eventually material *bunching*).

The PBVS cutting controller is the sensitive to calibration, hence future work should focus on on-line adaptive methods in order to compensate for poorly known camera parameters (intrinsic and extrinsic). In addition to this, the PBVS scheme should be improved such that 3D surface trajectories can be followed.

Section 4.4 shows that by using image moments as a feature all 6-DOF of the tool can be positioned. However, the drawback of this method is the need to have a known image rich in information. It would be interesting to see in future work if, during the meat cutting operation, an image of the cutting region provides sufficient information. Another drawback in the current work is the use of discrete objects to represent a continuous cutting trajectory. This means the convergence of the image features may lead to an unsatisfactory cutting trajectory. In future work in order to improve the performance of the IBVS, the problem with discontinuities must be addressed either by increasing the resolution of the camera or by changing the control law.

It should be noted, that the combination of force measurements with image control is a topic that has not been researched in great detail and due to the robustness to calibration errors deserves more attention. Most research has been focused on contour following on a surface, where the force measurements reconstruct the normal to the surface and the vision information control motion tangential to the surface. It would also be a worthwhile study, to investigate whether contour following using the 6-DOF image control in conjunction with force measurements would provide an improvement compared to the classical case.

A

Implementation of Deformable Object Modeling

In this chapter, a step by step description is given of the modeling of deformable objects used in Chapter 3, Section 3.4 and Chapter 4, Section 4.3. In summary, this approach consists of a modal analysis carried out in a Finite element environment to generate a set of modes. The finite element program described in this section is Patran© which serves as a graphical interface for the MSC Nastran© program, however it is possible to carry out the same analysis with Abaqus© . The goal is to generate a .mnf file that contains the object flexibility information. It should be noted that this file can *only* be opened using MSC ADAMS flex and not with any standard text editor.

A.1 Finite Element Modeling of flexible bodies

The desired geometry can be created externally using a CAD program and imported into PATRAN as a .stp file. Otherwise a geometry can be created using the default Patran CAD program. The following steps should be taken into account during the modeling of the geometry:

Geometry:

1. Before importing a file into Patran, ensure that the default units match the .stp file.
2. It should be noted that all inertial variables are calculated with respect to the origin of the geometry in Patran

3. If the model is a solid, rather than a shell or beam, the geometry must be check using the Patran command $Geometry \rightarrow Verify$. In any the geometry must be closed.

Properties:

1. Material properties are set in this section the material may be $Isotropic$, $Orthotropic$ etc. and may exists as a $2Dshell$, $3Dsolid$

Meshing:

1. The object is meshed as desired, then a set of $ASET$ (interface) nodes are defined in $DOF\ LIST$. These points correspond to the location of the joints or the loads that will be applied in ADAMS. The addition of $ASET$ points increases the number of modes of the structure. Note, at least one interface node must be defined for the analysis to run.
2. To better model some interactions, the option $MPC \rightarrow RPE2$ can be chosen. This allows loads to be transferred from the interface nodes to the surrounding nodes. This function creates a rigid link between the independent node (the interface node) and surrounding nodes.

Analysis:

1. The analysis type is defined as a Normal modes analysis or $SOL103$
2. In $SolutionParameters \rightarrow Adams\ Preparation$ units must be changed before the solution is computed
3. Finally in this section ensure that the $ASET$ is selected.

Among, the resulting files from this analysis, there are two of particular interest. Firstly, the .mnf file that contains all the information about the flexible object. Secondly, the .f06 that contains useful information about the analysis.

A.1.1 Using flexible bodies in ADAMS

The .mnf file can be directly imported into Adams. This file also supplies the necessary information to calculate the dynamic parameters of the flexible object as given in Section A.2. Once the body is imported, joints and forces can be applied to the interface nodes.

Typically, during modal analysis, high frequency modes that have little effect on the object behavior are disabled to reduce the complexity without losing resolution of the solution. Due to the flexible body modeling in Adams, a variation of the component mode synthesis described in Chapter 3, Section 3.2, the arbitrary disabling and enabling of modes is no longer possible. Instead, in order to reduce the number of modes the following strategy must be carried out:

1. Run the simulation with the desired parameters with all modes enabled
2. Choose a strain energy threshold for example 0.01%
3. Disable all modes whose contribution to the strain energy over the entire simulation is under this threshold

By doing so, a minimum set of modes can describe the flexibility of the mode during the *particular* simulation.

A.2 Dynamic Modeling

To calculate the inertia parameters given in Table 3.1, the following information is required. It should be noted that the numerical values of the inertia parameters do not necessarily come from Patran program and for geometrically simple objects can be obtained quite easily.

Geometric: The position of every node with respect to the origin of the body must be known. The position for node m is defined as $\mathbf{r}_m$

Inertial: The nodal mass, defined in this thesis for node m as dm_m must be known It should be noted that this mass is not equal for every node but is dependent on the mass distribution.

Flexibility:

1. The shape function must be obtained for every node in the body. The shape function represent the displacement due to mode k at node m is given as $\mathbf{\Phi}_{dk}(m)$
2. The vector generalized stiffness as $\mathbf{k}_{ee}$ which has N components.

Using the above information, the discrete equivalents of Table 3.1 are given in Table A.1. The numerical values of the inertia parameters are available in the .mnf file, however the values of the Centrifugal and Coriolis forces are not available and should be calculated using the formulas given in Chapter 3, Section.3.2.

Table A.1: Discrete Inertia invariants for flexible body of M nodes, where $j, k = 1 \ldots N$

Variable	Calculation	Description	Size
$\mathcal{I}1$	$\sum_{m=1}^{M} dm_m$	Mass of body	scalar
$\mathcal{I}2$	$\sum_{m=1}^{M} \mathbf{r}_m dm_m$	1^{st} moments of inertia	3×1
$\mathcal{I}3_k$	$\sum_{m=1}^{M} \mathbf{\Phi}_{dk}(m) dm_m$	Elastic 1^{st} moments of inertia	3×1
$\mathcal{I}4_k$	$\sum_{m=1}^{M} \hat{\mathbf{r}}_m \mathbf{\Phi}_{dk}(m) dm_m$		3×1
$\mathcal{I}5_{jk}$	$\sum_{m=1}^{M} \left(\mathbf{\Phi}_{dk}(m)\mathbf{\Phi}_{dj}(m)\right) dm_m$		3×1
$\mathcal{I}6$	$\sum_{m=1}^{M} \left(\mathbf{\Phi}_{d}^{T}(m)\mathbf{\Phi}_{d}(m)\right) dm_m$	Elastic inertia matrix	$M \times N$
$\mathcal{I}7$	$\sum_{m=1}^{M} \hat{\mathbf{r}}_m^T \hat{\mathbf{r}}_m dm_m$	Rigid inertia matrix	3×3
$\mathcal{I}8_k$	$\sum_{i=1}^{M} \hat{\mathbf{r}}_m \hat{\mathbf{\Phi}}_{dk}(m) dm_m$	Rigid Elastic inertia tensor	3×3
$\mathcal{I}9_{jk}$	$\sum_{m=1}^{M} \hat{\mathbf{\Phi}}_{dj}(m) \hat{\mathbf{\Phi}}_{dk}(m) dm_m$	Elastic Elastic inertia tensor	3×3

Bibliography

[AA93] M. Ait-Ahmed. *Contribution à la modélisation géométrique et dynamique des robots parallèles*. PhD thesis, LAAS, Toulouse, 1993. 131

[ACWK11] S. Amine, S. Caro, P. Wenger, and D. Kanaan. Singularity analysis of the H4 robot using Grassmann–Cayley algebra. *Robotica*, 1(1):1–10, 2011. 18, 73, 74

[ADV12] M. Afroun, A. Dequidt, and L. Vermeiren. Revisiting the inverse dynamics of the gough–stewart platform manipulator with special emphasis on universal–prismatic–spherical leg and internal singularity. *Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 226(10):2422–2439, 2012. 131

[AFD10] B. V Adorno, P. Fraisse, and S. Druon. Dual position control strategies using the cooperative dual task-space framework. In *Proceedings of the 2010 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3955–3960, 2010. 45

[aH00] A.S. alYahmadi and T.C. Hsia. Internal force-based impedance control of dual-arm manipulation of flexible objects. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation, (ICRA)*, volume 4, pages 3296–3301 vol.4, 2000. 109

[AH09] H. Abdellatif and B. Heimann. Computational efficient inverse dynamics of 6-dof fully parallel manipulators by using the lagrangian formalism. *Mechanism and Machine Theory*, 44(1):192–207, 2009. 131

[ALZR09] G. Arnold, L. Leiteritz, S. Zahn, and H. Rohm. Ultrasonic cutting of cheese: Composition affects cutting work reduction and energy demand. *International Dairy Journal*, 19(5):314–320, 2009. 165

[AM13] A. Abou Moughlbay. *Contributions in task sequencing and redundancy resolution - Application to humanoid and multi-arm robots*. PhD thesis, Ecole Centrale de Nantes, IRCCyN Laboratory, 2013. 216

[AMAM05] H.H. Abdelkader, Y. Mezouar, N. Andreff, and P. Martinet. 2 1/2 d visual servoing with central catadioptric cameras. In *Proceedings of* *the 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3572 – 3577, aug. 2005. 158

[AMK87] S. Arimoto, F. Miyazaki, and S. Kawamura. Cooperative motion control of multiple robot arms or fingers. In *Proceedings of the 1987 IEEE International Conference on Robotics and Automation (ICRA)*, volume 4, pages 1407–1412. IEEE, 1987. 50

[AXJ04] A.G. Atkins, X. Xu, and G. Jeronimidis. Cutting, by 'pressing and slicing,' of thin floppy slices of materials illustrated by experiments on cheddar cheese and salami. *Journal of Materials Science*, 39(8):2761–2766, 2004. 27, 165, 186

[AYAH07] Amer S. Al-Yahmadi, Jamil A., and T.C. Hsia. Modeling and control of two manipulators handling a flexible object. *Journal of the Franklin Institute*, 344(5):349 – 361, 2007. Modeling, Simulation and Applied Optimization Part II. 98, 108

[BC68] M.C.C. Bampton and J.R.R. Craig. Coupling of substructures for dynamic analyses. *AIAA Journal*, 6(7):1313–1319, 1968. 100, 170

[BC96] F. Boyer and P. Coiffet. Generalization of newton-euler model for flexible manipulators. *Journal of Robotic Systems*, 13(1):11–24, 1996. 98

[BDS99] J. Baeten and J. De Schutter. Improving force controlled planar contour following using online eye-in-hand vision based feedforward. In *Proceedings of the 1999 IEEE/ASME International Conference on Advanced Intelligent Mechatronics*, pages 902–907, 1999. 160

[BGGCO$^+$10] I. Bonilla, E.J. González-Galván, C. Chávez-Olivares, M. Mendoza, A. Loredo-Flores, F. Reyes, and B. Zhang. A vision-based, impedance control strategy for industrial robot manipulators. In *Proceedings of the 2010 IEEE Conference on Automation Science and Engineering (CASE)*, pages 216–221. IEEE, 2010. 161

[BH96] RC Bonitz and T.C. Hsia. Internal force-based impedance control for cooperating manipulators. *IEEE Transactions on Robotics and Automation*, 12(1):78–89, 1996. 43, 52, 123

[BHG97] S. Bhattacharya, H. Hatwal, and A. Ghosh. An on-line parameter estimation scheme for generalized stewart platform type parallel manipulators. *Mechanism and machine theory*, 32(1):79–89, 1997. 131

[BK98] F. Boyer and W. Khalil. An efficient calculation of flexible manipulator inverse dynamics. *The International Journal of Robotics Research*, 17(3):282–293, 1998. 21, 103, 105, 107, 112, 131

[BK13] S. Briot and W. Khalil. Recursive and symbolic calculation of the elastodynamic model of flexible parallel robots. *The International Journal of Robotics Research*, 33:469–483, 2013. 98, 103, 107, 215

[BKBLV07] F. Boyer, W. Khalil, M. Benosman, and G. Le Vey. Modeling and control of flexible robots. In Etienne Dombre and Wisama Khalil, editors, *Robot Manipulators: Modeling, Performance Analysis and Control*, chapter 7, pages 337–394. ISTE Ltd, Cambridge, Massachusetts, 2007. 102, 104, 105, 131

[BMB95] A. Bicchi, C. Melchiorri, and D. Balluchi. On the mobility and manipulability of general multiple limb robots. *IEEE Transactions on Robotics and Automation*, 11(2):215–228, 1995. 56

[BN98] M. Bro-Nielsen. Finite element modeling in surgery simulation. *Proceedings of the IEEE*, 86(3):490–503, 1998. 164

[BNU98] S. Bhattacharya, D.N. Nenchev, and M. Uchiyama. A recursive formula for the inverse of the inertia matrix of a parallel manipulator. *Mechanism and machine theory*, 33(7):957–964, 1998. 131

[Bog08] R. Bogue. Cutting robots: a review of technologies and applications. *Industrial Robot: An International Journal*, 35(5):390–396, 2008. 165

[Boo84] WJ Book. Recursive lagrangian dynamics of flexible manipulator arms. *The International Journal of Robotics Research*, 3(3):87–101, 1984. 98, 102

[Boy94] F. Boyer. *Contribution à la modélisation et à la commande dynamique des robots flexibles*. PhD thesis, Université Paris, 1994. 104

[BVD08] J. Billingsley, A. Visala, and M. Dunn. Robotics in agriculture and forestry. In Bruno Siciliano and Oussama Khatib, editors, *Springer handbook of robotics*, pages 1065–1077. Springer-Verlag, Berlin, Germany, 2008. 33

[CA01] C. Chevallereau and Y. Aoustin. Optimal reference trajectories for walking and running of a biped robot. *Robotica*, 19(5):557–569, 2001. 20, 85

[CB97] A. Codourey and E. Burdet. A body-oriented method for finding a linear form of the dynamic equation of fully parallel robots. In *Proceedings of the 1997 IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1612–1618. IEEE, 1997. 131

[CBL11] C. Copot, A. Burlacu, and C. Lazar. Visual predictive control architecture based on image moments for manipulator robots. In *Proceedings of the 2011 IEEE International Symposium on Industrial Electronics (ISIE)*, pages 963–968. IEEE, 2011. 201

[CCC00] F. Caccavale, P. Chiacchio, and S. Chiaverini. Task-space regulation of cooperative manipulators. *Automatica*, 36(6):879–887, 2000. 45

[CCMV08] F. Caccavale, P. Chiacchio, A. Marino, and L. Villani. Six-DOF impedance control of dual-arm cooperative manipulators. *IEEE/ASME Transactions on Mechatronics*, 13(5):576–586, 2008. 52

[CCS96] P. Chiacchio, S. Chiaverini, and B. Siciliano. Direct and inverse kinematics for coordinated motion tasks of a two-manipulator system. *Journal of Dynamic Systems, Measurement, and Control*, 118:691, 1996. 45

[CG09] M. Carricato and C. Gosselin. On the modeling of leg constraints in the dynamic analysis of gough/stewart-type platforms. *Journal of computational and nonlinear dynamics*, 4(1), 2009. 131

[CH01] P.I. Corke and S.A. Hutchinson. A new partitioned approach to image-based visual servo control. *IEEE Transactions on Robotics and Automation,*, 17(4):507–515, 2001. 156, 157, 158

[CH07] F. Chaumette and S. Hutchinson. Visual servo control. II. Advanced approaches [Tutorial]. *IEEE Robotics & Automation Magazine*, 14(1):109–118, March 2007. 154

[CH08] F. Chaumette and S. Hutchinson. Visual servoing and visual tracking. *Handbook of Robotics*, pages 563–583, 2008. 152

[Cha98] F. Chaumette. Potential problems of stability and convergence in image-based and position-based visual servoing. *The confluence of vision and control*, pages 66–78, 1998. 157, 158

[Cha04] F. Chaumette. Image moments: a general and useful set of features for visual servoing. *IEEE Transactions on Robotics*, 20(4):713–723, 2004. 29, 155, 201, 203, 204

[Cha06] W.C. Chang. Hybrid force and vision-based contour following of planar robots. *Journal of Intelligent & Robotic Systems*, 47:215–237, 2006. 10.1007/s10846-006-9074-0. 160, 199

[CLY+01] H. Cheng, GF Liu, YK Yiu, ZH Xiong, and ZX Li. Advantages and dynamics of parallel manipulators with redundant actuation. In *Proceedings of the 2001 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pages 171–176, 2001. 49

[CRE93] F. Chaumette, P. Rives, and B. Espiau. Classification and realization of the different vision-based tasks. In K. Hashimoto, editor, *Visual Servoing*, pages 199–228. Singapore:World Scientific, 1993. 154

[CU08] Fabrizio Caccavale and Masaru Uchiyama. Cooperative manipulators. In B Siciliano and O Khatib, editors, *Springer Handbook of Robotics*, pages 701–718. Springer Berlin Heidelberg, 2008. 42

[CV00] F. Caccavale and L. Villani. Impedance control of cooperative manipulators. *Machine, Intelligence & Robotic Control*, 2:51–57, 2000. 52

[CYL03] H. Cheng, Y. Yiu, and Z. Li. Dynamics and control of redundantly actuated parallel manipulators. *IEEE/ASME Transactions on Mechatronics*, 8(4):483–491, December 2003. 47

[CZF+13] Sang C., G. Zhang, T. Fuhlbrigge, T. Watson, and R. Tallian. Applications and requirements of industrial robots in meat processing. In *Proceedings of the 2013 IEEE International Conference on Automation Science and Engineering (CASE)*, pages 1107–1112, Aug 2013. 33

[DC99] B. Dasgupta and P. Choudhury. A general strategy based on the newton–euler approach for the dynamic formulation of parallel manipulators. *Mechanism and Machine Theory*, 34(6):801–824, 1999. 131

[DCA99] H. Delingette, S. Cotin, and N. Ayache. A hybrid elastic model allowing real-time cutting, deformations and force-feedback for surgery training and simulation. In *Computer Animation, 1999*, pages 70–81, 1999. 163, 164, 169

[DDDBV93] X. Delebarre, E. Dégoulange, P. Dauchez, and Y. Bouffard-Vercelli. Force control of a two-arm robot manipulating a deformable object. In Raja Chatila and Gerd Hirzinger, editors, *Experimental Robotics II*, volume 190 of *Lecture Notes in Control and Information Sciences*, pages 255–269. Springer Berlin Heidelberg, 1993. 97, 109

[DLOG08] A. De Luca, G. Oriolo, and P.R. Giordano. Feature depth observation for image-based visual servoing: Theory and experiments. *The International Journal of Robotics Research*, 27(10):1093–1116, 2008. 157

[DM98] B. Dasgupta and T.S. Mruthyunjaya. A newton-euler formulation for the inverse dynamics of the stewart platform manipulator. *Mechanism and Machine Theory*, 33(8):1135–1152, 1998. 131

[DS11] J. Das and N. Sarkar. Autonomous shape control of a deformable object by multiple manipulators. *Journal of Intelligent & Robotic Systems*, 62(1):3–27, 2011. 107, 111

[EBG12] N. Essahbi, Belhassen C. Bouzgarrou, and G. Gogu. Soft material modeling for robotic manipulation. *Applied Mechanics and Materials*, 162:184–193, 2012. 163, 169

[EBS13] B. Esakki, R B Bhat, and C.Y. Su. Robust control of collaborative manipulators-flexible object system. *International Journal of Advanced Robotic Systems*, 10(257), 2013. 98

[FG99] S.F. Frisken-Gibson. Using linked volumes to model object collisions, deformation, cutting, carving, and joining. *IEEE Transactions on Visualization and Computer Graphics,*, 5(4):333–348, 1999. 163

[FYW07] S. Fu, Y. Yao, and Y. Wu. Comments on 'A Newton–Euler formulation for the inverse dynamics of the Stewart platform manipulator'. *Mechanism and Machine Theory*, 42(12):1668–1671, 2007. 131

[GHLC92] Z. Geng, L.S. Haynes, J.D. Lee, and R.L. Carroll. On the dynamic model and kinematic analysis of a class of stewart platforms. *Robotics and Autonomous Systems*, 9(4):237–254, 1992. 131

[Gog05] G. Gogu. Mobility of mechanisms: a critical review. *Mechanism and Machine Theory*, 40(9):1068 – 1097, 2005. 54

[Gog07] G. Gogu. *Structural Synthesis of Parallel Robots: Part 1: Methodology*. Springer Verlag, 2007. 57, 64

[Gog08] G. Gogu. Constraint singularities and the structural parameters of parallel robots. In J. Lenarcic and P. Wenger, editors, *Advances in Robot Kinematics: Analysis and Design*, pages 21–28. Springer, 2008. 77

[Gog10] G. Gogu. *Structural Synthesis of Parallel Robots: Part 3: Topologies with Planar Motion of the Moving Platform*, volume 173. Springer, 2010. 39

[Gos93] C.M. Gosselin. Parallel computational algorithms for the kinematics and dynamics of parallel manipulators. In *Proceedings of the 1993 IEEE International Conference on Robotics and Automation (ICRA)*, pages 883–888. IEEE, 1993. 131

[GRF+03] J. Gallardo, J.M. Rico, A. Frisoli, D. Checcacci, and M. Bergamasco. Dynamics of parallel manipulators by means of screw theory. *Mechanism and machine theory*, 38(11):1113–1131, 2003. 131

[GSGL10a] G. Guire, L. Sabourin, G. Gogu, and E. Lemoine. Robotic cell for beef carcass primal cutting and pork ham boning in meat industry. *Industrial Robot: An International Journal*, 37(6):532–541, 2010. 33

[GSGL10b] G. Guire, L. Sabourin, G. Gogu, and E. Lemoine. Robotic cell with redundant architecture and force control: application to cutting and boning. In *Proceedings of the 2010 IEEE 19th International Workshop on Robotics in Alpe-Adria-Danube Region (RAAD)*, pages 99–104. IEEE, 2010. 33

[Guy65] R.J. Guyan. Reduction of stiffness and mass matrices. *AIAA journal*, 3(2):380–380, 1965. 100

[Hay86] S. Hayati. Hybrid position/force control of multi-arm cooperating robots. In *Proceedings of the 1986 IEEE International Conference on Robotics and Automation (ICRA)*, volume 3, pages 82 – 89, apr 1986. 50

[Her99] JM Hervé. The lie group of rigid body displacements, a fundamental tool for mechanism design. *Mechanism and Machine theory*, 34(5):719–730, 1999. 52

[HHC96] S. Hutchinson, G.D. Hager, and P.I. Corke. A tutorial on visual servo control. *IEEE Transactions on Robotics and Automation*, 12(5):651 – 670, oct 1996. 151, 153, 154

[HIA98] K. Hosoda, K. Igarashi, and M. Asada. Adaptive hybrid control for visual and force servoing in an unknown environment. *Robotics & Automation Magazine, IEEE*, 5(4):39–43, 1998. 160, 199

[Hin10] L. Hinrichsen. Manufacturing technology in the danish pig slaughter industry. *Meat science*, 84(2):271–275, 2010. 33

[Hog85] N. Hogan. Impedance control: An approach to manipulation. *Journal of Dynamic Systems Measurements and Control*, 107(March):17–24, 1985. 46, 51, 177

[Hu62] M.K. Hu. Visual pattern recognition by moment invariants. *IRE Transactions on Information Theory*, 8(2):179–187, 1962. 154, 201, 203

[Hun78] K. Hunt. *Kinematic geometry of mechanisms*. Cambridge Univ Press, 1978. 17, 58

[INS09] INSEE. Industrie agroalimentaire. http://www.insee.fr/fr/themes/document.asp?ref_id=T10F181, 2009. Accessed: 2013-04-22. 33

[Ji93] Z. Ji. Study of the effect of leg inertia in stewart platforms. In *Proceedings of the 1993 IEEE International Conference on Robotics and Automation (ICRA)*, pages 121–126. IEEE, 1993. 131

[KC97] W. Khalil and D. Creusot. Symoro+: a system for the symbolic modelling of robots. *Robotica*, 15:153–161, 1997. 59

[KCM00] U. Kühnapfel, H.K. Cakmak, and H. Maaß. Endoscopic surgery training using virtual reality and deformable tissue simulation. *Computers & Graphics*, 24(5):671–682, 2000. 163

[KD04] W. Khalil and E. Dombre. *Modeling, identification and control of robots*. Butterworth-Heinemann, 2004. 48, 83

[KG00] W. Khalil and M. Gautier. Modeling of mechanical systems with lumped elasticity. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation (ICRA)*, volume 4, pages 3964–3969. IEEE, 2000. 96, 131

[KG02] W. Khalil and S. Guegan. A novel solution for the dynamic modeling of gough-stewart manipulators. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 817–822. IEEE, 2002. 131, 132

[KG07] X. Kong and C.M. Gosselin. *Type synthesis of parallel mechanisms*. Springer Publishing Company, Incorporated, 2007. 17, 58

[Kha87] O. Khatib. A unified approach for motion and force control of robot manipulators: The operational space formulation. *IEEE Journal of Robotics and Automation*, 3(1):43–53, 1987. 113, 176

[KI07] W. Khalil and O. Ibrahim. General solution for the dynamic modeling of parallel robots. *Journal of Intelligent and robotic systems*, 49(1):19–37, 2007. 123, 131, 215

[KJM97] W. Kraus Jr and B.J. McCarragher. Hybrid position/force coordination for dual-arm manipulation of flexible materials. In *Proceedings of the 1997 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pages 202–207. IEEE, 1997. 110

[KK86] W. Khalil and J. Kleinfinger. A new geometric notation for open and closed-loop robots. In *Proceedings of the 1986 IEEE International Conference on Robotics and Automation (ICRA)*, volume 3, pages 1174–1179. IEEE, 1986. 7, 17, 47, 59, 60, 131, 167

[Kus10] Y. Kusuda. The use of robots in the japanese food industry. *Industrial Robot: An International Journal*, 37(6):503–508, 2010. 33

[LKC12a] P. Long, W. Khalil, and S. Caro. Control of a lower mobility dual arm system. In *Proceedings of the 10th IFAC Symposium on Robot Control (SYROCO)*, volume 10, pages 307–312, 2012. 40, 90, 213

[LKC12b] P. Long, W. Khalil, and S. Caro. Kinematic analysis of lower mobility cooperative arms by screw theory. In *Proceedings of the 9th International Conference on Informatics in Control Automation and Robotics (ICINCO)*, pages 208–285, 2012. 40, 90, 213

[LKC14] P. Long, W. Khalil, and S. Caro. Kinematic and dynamic analysis of lower-mobility cooperative arms. *Robotica*, FirstView:1–22, 2014. 41, 90, 213

[LKM13] P. Long, W. Khalil, and P. Martinet. Modeling & control of a meat-cutting robotic cell (Finalist for Best Paper). In *Proceedings of the 2013 16th International Conference on Advanced Robotics (ICAR),*, pages 1–6. IEEE, 2013. 212, 214

[LKM14a] P. Long, W. Khalil, and P. Martinet. Dynamic modeling of parallel robots with flexible platforms. *Mechanism and Machine Theory*, 81:21–35, 2014. 93, 147, 213

[LKM14b] P. Long, W. Khalil, and P. Martinet. Force/vision control for robotic cutting of soft materials. In *Proceedings of the 2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2014. 212, 214

[LKM14c] P. Long, W. Khalil, and P. Martinet. Robotic cutting of soft materials using force control & image moments. In *Proceedings of the International Conference on Control, Automation, Robotics and Vision (ICARcv)*, 2014. 212, 214

[LKM14d] P. Long, W. Khalil, and P. Martinet. Robotic deformable object cutting: From simulation to experimental validation. In *European Workshop on Deformable Object Manipulation (EWDOM) Lyon, France, March 20th*, 2014. 212, 214

[LKM15] P. Long, W. Khalil, and P. Martinet. Dynamic modeling of cooperative manipulators with flexible objects (submitted). In *Proceedings of the 2015 IEEE International Conference on Robotics and, Automation (ICRA)*, 2015. 93, 147, 213

[LLH06] A.C. Leite, F. Lizarralde, and Liu H. Hybrid vision-force robot control for tasks on unknown smooth surfaces. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2244 –2249, may 2006. 159, 199

[LLL93] G. Lebret, K. Liu, and F.L. Lewis. Dynamic analysis and control of a stewart platform manipulator. *Journal of Robotic systems*, 10(5):629–655, 1993. 131

[LLL00] M.-J. Liu, C.X. Li, and C.N. Li. Dynamics analysis of the gough-stewart platform manipulator. *IEEE Transactions on Robotics and Automation*, 16(1):94–98, 2000. 131

[LMKM13] P. Long, A. Moughlbay, W. Khalil, and P. Martinet. Robotic meat cutting. In *ICT-PAMM Workshop, Kumamoto, Japan, November 9-10th*, 2013. 212, 214

[LMZP11] C. Liu, P. Moreira, N. Zemiti, and P. Poignet. 3d force control for robotic-assisted beating heart surgery based on viscoelastic tissue model. In *Proceedings of the Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 7054–7058. IEEE, 2011. 163

[LS88] K-M Lee and D.K. Shah. Dynamic analysis of a three-degrees-of-freedom in-parallel actuated manipulator. *IEEE Journal of Robotics and Automation*, 4(3):361–367, 1988. 131

[LSV06] V. Lippiello, B. Siciliano, and L. Villani. Robot interaction control using force and vision. In *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1470–1475. IEEE, 2006. 160

[LXB99] Y.H. Liu, Y. Xu, and M. Bergerman. Cooperation control of multiple manipulators with passive joints. *IEEE Transactions on Robotics and Automation*, 15(2):258–267, April 1999. 47, 56

[Mal04] E. Malis. Improving vision-based control using efficient second-order minimization techniques. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1843–1848. Ieee, 2004. 152

[MDM07] P. Mukherjee, B. Dasgupta, and A.K. Mallik. Dynamic stability index and vibration analysis of a flexible stewart platform. *Journal of sound and vibration*, 307(3):495–512, 2007. 131

[MG99] P. Martinet and J. Gallice. Position based visual servoing using a non-linear approach. In *Proceedings of the 1999 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pages 531–536. IEEE, 1999. 155

[MH01] M. Mahvash and V. Hayward. Haptic rendering of cutting: A fracture mechanics approach. *Haptics-e*, 2(3):1–12, 2001. 164, 166

[MI96] J.K. Mills and J.G.L. Ing. Dynamic modeling and control of a multi-robot system for assembly of flexible payloads with applications to automotive body assembly. *Journal of Robotic Systems*, 13(12):817–836, 1996. 100, 109

[ML03] C. Mendoza and C. Laugier. Simulating soft tissue cutting using finite element models. *Robotics and Automation, 2003*, 2003. 163

[MLM+05] U. Meier, O. Lopez, C. Monserrat, M.C. Juan, and M. Alcaniz. Real-time deformable models for surgery simulation: a survey. *Computer methods and programs in biomedicine*, 77(3):183–197, 2005. 164

[MLZP12] P. Moreira, Chao L., N. Zemiti, and P. Poignet. Soft tissue force control using active observers and viscoelastic interaction model. In *Proceedings of the 2012 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4660–4666, May 2012. 163

[MM98] G. Morel and E. Malis. Impedance based combination of visual and force control. *Robotics and Automation, 1998.*, (May):1743–1748, 1998. 161

[MNK09] M. Mahboubkhah, M.J. Nategh, and S.E. Khadem. A comprehensive study on the free vibration of machine tools' hexapod table. *The International Journal of Advanced Manufacturing Technology*, 40(11-12):1239–1251, 2009. 131

[MOP07] G. L. Mariottini, G. Oriolo, and D. Prattichizzo. Image-based visual servoing for nonholonomic mobile robots using epipolar geometry. *IEEE Transactions on Robotics*, 23(1):87 –100, feb. 2007. 158

[MP10] S. A.A Moosavian and E. Papadopoulos. Cooperative object manipulation with contact impact using multiple impedance control. *International Journal of Control, Automation and Systems*, 8(2):314–327, 2010. 52

[MPM07] Y. Mezouar, M. Prats, and P. Martinet. External hybrid vision/force control. In *Proceedings of the 2007 International Conference on Advanced Robotics (ICAR)*, 2007. 161

[MRO08] S. Misra, K.T. Ramesh, and A.M. Okamura. Modeling of tool-tissue interactions for computer-based surgical simulation: a literature review. *Presence: Teleoperators and Virtual Environments*, 17(5):1–41, 2008. 162

[MSC05] E. Marchand, F. Spindler, and F. Chaumette. Visp for visual servoing: a generic software platform with a wide class of robot control skills. *IEEE Robotics & Automation Magazine*, 12(4):40–52, 2005. 184

[MSYO99] S. Matsuoka, K. Shimizu, N. Yamazaki, and Y. Oki. High-speed end milling of an articulated robot and its characteristics. *Journal of materials processing technology*, 95(1):83–89, 1999. 164

[Mul05] A. Muller. Internal preload control of redundantly actuated parallel manipulators its application to backlash avoiding control. *IEEE Transactions on Robotics*, 21(4):668–677, 2005. 49

[Mul06] A. Muller. Stiffness control of redundantly actuated parallel manipulators. In *Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1153–1158, 2006. 49

[NA89] M.A. Nahon and J. Angeles. Force optimization in redundantly-actuated closed kinematic chains. In *Proceedings of the 1989 IEEE International Conference on Robotics and Automation (ICRA)*, pages 951–956, 1989. 49

[NAhLRL13] D. Navarro-Alarcon, Yun hui Liu, J.G. Romero, and Peng Li. Model-free visually servoed deformation control of elastic objects by robot manipulators. *IEEE Transactions on Robotics*, 29(6):1457–1468, Dec 2013. 110

[NG89] Y. Nakamura and M. Ghodoussi. Dynamics computation of closed-link robot mechanisms with nonredundant and redundant actuators. *IEEE Transactions on Robotics and Automation*, 5(3):294–302, Jun 1989. 48

[NMK95] B.J. Nelson, J.D. Morrow, and P.K. Khosla. Improved force control through visual servoing. In *Proceedings of the 1995 American Control Conference*, volume 1, pages 380–386. IEEE, 1995. 158, 160

[NvdS00] H.W. Nienhuys and A.F. van der Stappen. Combining finite element deformation with cutting for surgery simulations. In *EuroGraphics short presentations*, pages 43–52. Citeseer, 2000. 164

[O'C07] W.J. O'Connor. Control of flexible mechanical systems: wave-based techniques. In *American Control Conference, 2007. ACC '07*, pages 4192–4202, 2007. 109

[OJR04] T. Olsson, R. Johansson, and A. Robertsson. Flexible force-vision control for surface following using multiple cameras. In *Proceedings of the 2004 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pages 789–803. IEEE, 2004. 160

[ÖÖ01] B. Özkan and M.K. Özgören. Invalid joint arrangements and actuator related singular configurations of a system of two cooperating scara manipulators. *Mechatronics*, 11(4):491–507, 2001. 47, 57

[Ott00] G. Ottarsson. Modal flexibility method in ADAMS/Flex. Technical report, Mechanical Dynamics Inc., August 2000. 170

[OW99] J.F. O'Brien and J.T. Wen. Redundant actuation for improving kinematic manipulability. In *Proceedings of the 1999 IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1520–1525. IEEE, 1999. 55, 56, 81

[PD93] V. Perdereau and M. Drouin. A new scheme for hybrid force-position control. *Robotica*, 11(05):453–464, 1993. 161

[PD96] V. Perdereau and M. Drouin. Hybrid external control for two robot coordinated motion. *Robotica*, 14(02):141–153, 1996. 50

[PGP01] M.T. Pham, M. Gautier, and P. Poignet. Identification of joint stiffness with bandpass filtering. In *Proceedings of the 2001 IEEE International Conference on (ICRA) Robotics and Automation*, volume 3, pages 2867–2872. IEEE, 2001. 96

[PJ00] A. Pichler and M. Jagersand. Uncalibrated hybrid force-vision manipulation. In *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, volume 3, pages 1866–1871 vol.3, 2000. 160, 199

[PL03] J.A. Piepmeier and H. Lipkin. Uncalibrated eye-in-hand visual servoing. *The International Journal of Robotics Research*, 22(10-11):805–819, 2003. 199

[PM02] E.J. Park and J.K. Mills. Dynamic modeling of flexible payloads grasped by actuated grippers using component mode synthesis. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation (ICRA)*, volume 3, pages 2443–2448. IEEE, 2002. 101

[PMdPL07] M. Prats, P. Martinet, A.P. del Pobil, and S. Lee. Vision force control in task-oriented grasping and manipulation. *Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1320–1325, October 2007. 161

[RB91] C. Reboulet and T. Berthomieu. Dynamic models of a six degree of freedom parallel manipulators. In *Proceedings of the Fifth International Conference on Advanced Robotics, 'Robots in Unstructured Environments', (ICAR)*, pages 1153–1157. IEEE, 1991. 131

[RC80] MH Raibert and JJ Craig. Hybrid position/force control of manipulators. In *American Society of Mechanical Engineers, Winter Annual Meeting, Chicago, Ill*, page 1980, 1980. 46, 50

[RCM10] M. Rognant, E. Courteille, and P. Maurine. A systematic procedure for the elastodynamic modeling and identification of robot manipulators. *IEEE Transactions on Robotics*, 26(6):1085–1093, 2010. 21, 112

[Rod90] G. Rodriguez. Spatial operator approach to flexible manipulator inverse and forward dynamics. In *Proceedings of the 1990 IEEE International Conference on Robotics and Automation (ICRA)*, pages 845–850. IEEE, 1990. 131

[RPL08] R. Richa, P. Poignet, and C. Liu. Deformable motion tracking of the heart surface. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3997–4003. IEEE, 2008. 163

[RTLMM12] E. Reyssat, T. Tallinen, M. Le Merrer, and L. Mahadevan. Slicing softly with shear. *Physical review letters*, 109(24):244301, 2012. 27, 166

[Sha13] A. Shabana. *Dynamics of multibody systems*. Cambridge university press, 2013. 102

[SL97] Dong S. and Yunhui L. Modeling and impedance control of a two-manipulator system handling a flexible beam. In *Proceedings of 1997 IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1787–1792 vol.2, 1997. 98, 108, 109

[SL01] D. Sun and Y. Liu. Position and Force Tracking of a System Manipulating a Flexible Beam. *Journal of Robotic Systems*, 18:197–212, 2001. 108

[SPDC06] M. Sauvée, P. Poignet, E. Dombre, and E. Courtial. Image based visual servoing through nonlinear model predictive control. In *Proceedings of the 2006 45th IEEE Conference on Decision and Control*, pages 1776–1781. IEEE, 2006. 157

[Spo87] M.W. Spong. Modeling and control of elastic joint robots. *Journal of dynamic systems, measurement, and control*, 109(4):310–318, 1987. 96

[SRPD05] J.-P. Saut, C. Remond, V. Perdereau, and M. Drouin. Online computation of grasping force in multi-fingered hands. In *Proceedings of the 2005 2005 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1223–1228, Aug 2005. 49

[SSJ88] M. Subbiah, A.M. Sharan, and J. Jain. A study of the dynamic condensation techniques for the machine tools and robotic manipulators. *Mechanism and machine theory*, 23(1):63–69, 1988. 100

[Sta11] S. Staicu. Dynamics of the 6-6 stewart parallel manipulator. *Robotics and Computer-Integrated Manufacturing*, 27(1):212–220, 2011. 131

[SU94] M.M. Svinin and M. Uchiyama. Cartesian-level control strategy for a system of manipulators coupled through a flexible object. In *Proceedings of the 1994 IEEE/RSJ/GI International Conference on Intelligent Robots and Systems (IROS)*, volume 1, pages 687–694 vol.1, 1994. 109

[Sug89] K. Sugimoto. Computational scheme for dynamic analysis of parallel manipulators. *Transaction of the ASME, Journal of mechanics, Transmission and Automation in Design*, 111:29–33, 1989. 131

[Tah04] O. Tahri. *Application des moments à l'asservissement visuel et au calcul de pose*. PhD thesis, Université de Rennes, 2004. 203

[Tan08] Tanio K Tanev. Geometric algebra approach to singularity of parallel manipulators with limited mobility. In J. Lenarcic and P. Wenger, editors, *Advances in Robot Kinematics: Analysis and Design*, pages 39–48. Springer, 2008. 39

[TC05] O. Tahri and F. Chaumette. Point-based and region-based image moments for visual servoing of planar objects. *IEEE Transactions on Robotics*, 21(6):1116–1127, 2005. 29, 155, 201, 202, 204

[TEJ09] A. Tavasoli, M. Eghtesad, and H. Jafarian. Two-time scale control and observer design for trajectory tracking of two cooperating robot manipulators moving a flexible beam. *Robot. Auton. Syst.*, 57(2):212–221, February 2009. 98, 108

[TMCG02] B. Thuilot, P. Martinet, L. Cordesses, and J. Gallice. Position based visual servoing: keeping the object in the field of vision. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1624–1629. IEEE, 2002. 155

[TS03] R.H. Taylor and D. Stoianovici. Medical robotics in computer-integrated surgery. *IEEE Transactions on Robotics and Automation*, 19(5):765–781, 2003. 164

[Tsa87] R.Y. Tsai. A versatile camera calibration technique for high-accuracy 3d machine vision metrology using off-the-shelf tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, 1987. 154, 184

[Tsa00] L.W. Tsai. Solving the inverse dynamics of a stewart-gough manipulator by the principle of virtual work. *Journal of Mechanical design*, 122:3, 2000. 131

[TTI06] R. Tinos, M.H. Terra, and J.Y Ishihara. Motion and Force Control of Cooperative Robotic Manipulators With Passive Joints. *IEEE Transactions on Control Systems Technology*, 14(4):725–734, October 2006. 50

[UD88] M. Uchiyama and P. Dauchez. A symmetric hybrid position/force control scheme for the coordination of two robots. In *Proceedings of the 1988 IEEE International Conference on Robotics and Automation (ICRA)*, pages 350–356, 1988. 45, 50

[VPZ08] M. Vakil, H. Pendar, and H. Zohoor. Comments on 'Closed-form dynamic equations of the general Stewart platform through the Newton–Euler approach' and 'A Newton–Euler formulation for the inverse dynamics of the Stewart platform manipulator'. *Mechanism and Machine Theory*, 43(10):1349–1351, 2008. 131

[VVW04] L. Vigneron, J. Verly, and S. Warfield. On extended finite element method (xfem) for modelling of organ deformations associated with surgical cuts. *Medical Simulation*, pages 134–143, 2004. 164, 169

[WFM89] I.D. Walker, R.A. Freeman, and Steven I. Marcus. Internal object loading for multiple cooperating robot manipulators. In *Proceedings of the 1989 IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 606–611, May 1989. 43, 49

[WHKK01] T. Wada, S. Hirai, S. Kawamura, and N. Kamiji. Robust manipulation of deformable objects by a simple pid feedback. In *Proceedings of the 2001 IEEE International Conference on Robotics and Automation, (ICRA)*, volume 1, pages 85–90 vol.1, 2001. 97, 110

[WN03] T. M Wasfy and Ahmed K Noor. Computational strategies for flexible multibody systems. *Applied Mechanics Reviews*, 56(6):553–614, 2003. 102

[WR94] D.M. Weer and S.M. Rock. Experiments in object impedance control for flexible objects. In *Proceedings of the 1994 IEEE International Conference on Robotics and Automation, (ICRA)*, pages 1222–1227 vol.2, 1994. 110

[WW03] O. Wallrapp and S. Wiedemann. Comparison of results in flexible multibody dynamics using various approaches. *Nonlinear Dynamics*, 34(1-2):189–206, 2003. 131

[WWWL09] J. Wu, J. Wang, L. Wang, and T. Li. Dynamics and control of a planar 3-DOF parallel manipulator with actuation redundancy. *Mechanism and Machine Theory*, 44(4):835–849, April 2009. 49

[XGXT00] D. Xiao, B.K. Ghosh, N. Xi, and T.J. Tarn. Sensor-based hybrid position/force control of a robot manipulator in an uncalibrated environment. *IEEE Transactions on Control Systems Technology*, 8(4):635–645, 2000. 159, 199

[YOS99] B. Yi, S. Oh, and I. Suh. A five-bar finger mechanism involving redundant actuators: Analysis and its applications. *IEEE Transactions On Robotics And Automation*, 15(6):1001, 1999. 49

[YSYO99] H. Yeo, I. H. Suh, B. Yi, and S. Oh. A single closed-loop kinematic chain approach for a hybrid control of two cooperating arms with a passive joint: an application to sawing task. *IEEE Transactions on Robotics and Automation*, 15(1):141–151, 1999. 47, 50

[ZBG02] D. Zlatanov, I.A. Bonev, and C.M. Gosselin. Constraint singularities of parallel mechanisms. In *Proceedings of the 2002 IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 496–502. IEEE, 2002. 58, 77

[ZC04] Y. Zhao and C.C. Cheah. Hybrid vision-force control for robot with uncertainties. In *Proceedings of the 2004 IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 261–266 Vol.1, 2004. 160

[ZCLM06] D. Zhou, M.R. Claffee, K.M. Lee, and G.V. McMurray. Cutting,'by pressing and slicing', applied to the robotic cut of bio-materials. ii. force during slicing and pressing cuts. In *Proceedings of 2006 Proceedings of the 2000 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2256–2261. IEEE, 2006. 166

[ZH97] G. Zeng and A. Hemami. An adaptive control strategy for robotic cutting. In *Proceedings of 1997 Proceedings of the 2000 IEEE International Conference on Robotics and Automation (ICRA)*, volume 1, pages 22–27. IEEE, 1997. 165

[ZL88] Y.F. Zheng and JYS Luh. Optimal load distribution for two industrial robots handling a single object. In *Proceedings of the 1988 IEEE International Conference on Robotics and Automation (ICRA)*, pages 344–349, 1988. 49

[ZZG00] T. Zhou, J.W. Zu, and A.A. Goldenberg. Vibration controllability of flexible robot-payload systems. In *Proceedings of the 2000 IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1484–1489, 2000. 101, 107, 126

Publications

P. Long, W. Khalil, and S. Caro, Kinematic and dynamic analysis of lower-mobility cooperative arms, *Robotica*, FirstView:pages 1–22, 2014.

P. Long, W. Khalil, and P. Martinet, Dynamic modeling of parallel robots with flexible platforms, *Mechanism and Machine Theory*, 81:21–35, 2014.

P. Long, W. Khalil, and S. Caro, Kinematic analysis of lower mobility cooperative arms by screw theory, in *9th International Conference on Informatics in Control Automation and Robotics*, pages 208–285, 2012.

P. Long, W. Khalil, and S. Caro, Control of a lower mobility dual arm system, in *10th IFAC Symposium on Robot Control (SYROCO)*, vol. 10, pages 307–312, 2012.

P. Long, W. Khalil, and P. Martinet, Modeling & control of a meat-cutting robotic cell (Finalist for Best Paper), in *2013 16th International Conference on Advanced Robotics (ICAR)*, pages 1–6, IEEE, 2013.

P. Long, W. Khalil, and P. Martinet, Force/vision control for robotic cutting of soft materials , in *2014 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2014*, 2014.

P. Long, W. Khalil, and P. Martinet, Robotic Cutting of Soft Materials using Force Control & Image Moments, in *International Conference on Control, Automation, Robotics and Vision (ICARcv)*, 2014.

P. Long, W. Khalil, and P. Martinet, Dynamic Modeling of Cooperative Manipulators with Flexible Objects (Submitted), in *International Conference on Robotics and, Automation (ICRA)*, 2015.

P. Long, W. Khalil, and P. Martinet, Robotic deformable object cutting: From simulation to experimental validation, in *European Workshop on Deformable Object Manipulation (EWDOM) Lyon, France, March 20th*, 2014.

P. Long, A. Moughlbay, W. Khalil, and P. Martinet, Robotic meat cutting, in *ICT-PAMM Workshop, Kumamoto, Japan, November 9-10th*, 2013.

P. Martinet, A. Moughlbay, and P. Long, Control of redundancy in complex systems : from theoretical concepts to applications, in *International Workshop on Wireless Communications and User centered Services in Pervasive Environments Hanoi, Vietnam, 19-20 September*, 2013.

l'université nantes angers le mans
COMMUNAUTÉ D'UNIVERSITÉS ET ÉTABLISSEMENTS

Centrale Nantes

Thèse de Doctorat

Philip LONG

Contributions à la Modélisation et la Commande de Robots Coopératifs

Contributions to the Modeling and Control of Cooperative Manipulators

Résumé

L'utilisation de robots coopératifs deviendra essentielle dans différents d'applications. En employant deux robots, une plus large gamme de tâches peut être réalisée. Cette thèse focalise sur la manipulation coopérative. Elle contient trois contributions principales.
La première concerne l'étude analytique d'un système coopératif de basse mobilité qui tient un objet rigide. Les études cinématiques et dynamiques permettent d'obtenir la mobilité, les singularités et le meilleur choix d'ensemble d'actionneurs.
La seconde porte sur la modèle dynamique d'un manipulateur coopératif souple. L'analyse focalise sur les robots coopératifs avec des objets flexible. L'objet est modélisé par les fonctions de formes et une solution de forme fermée est dérivée. On exploite cette méthode pour obtenir le modèle dynamique d'un robot parallèle, le Gough Stewart robot.
La dernière concerne la séparation d'objets mous par plusieurs robots. La construction d'un simulateur d'un système multi-bras pour la découpe de viande est décrite. Une commande par vision/effort est développée qui permet le système de s'adapter d'en fonction de l'état de l'objet. Des expérimentations sont effectuées et montrent comment, pendent la découpe, l'effort qui est généré par la résistance de l'objet peut servir pour éviter les déformations globales d'objets.

Mots clés

Robots Coopératifs, Manipulation d'objets mous, Commande par vision/effort, Découpe Robotique, Robots souples.

Abstract

Cooperative manipulation strategies are essential as the domain of robot applications is increased. By using two or more robots, a greater range of tasks can be accomplished. This thesis treats several cases of cooperative manipulation, from the manipulation of rigid objects to the separation of deformable materials. The contributions of this thesis are threefold.
Firstly, a study of a lower mobility cooperative system grasping a rigid object is undertaken. A kinematic and dynamic analysis is carried out to obtain the mobility, singular configurations and optimum actuation scheme of the system.
Secondly, a general dynamic model of a closed chain robot with flexibility is derived. The analysis focuses on cooperative robots with flexible objects. The object is modeled using distributed flexibility and a closed form relation is derived for the dynamic model. This method is applied to the Gough Stewart manipulator with flexible platform and the dynamic model is obtained.
Finally, the separation of deformable bodies using multiple robots is investigated. A simulator is created where a multi-arm meat cutting system is modeled. Force/Vision control schemes are proposed that allow the system to adapt to on-line deformations of the target object. An experimental validation is carried out that shows the how the resistive cutting force can be used by the controller to avoid globally deforming the object.

Key Words

Cooperative Manipulators, Flexible Object Manipulation, Multi-arm system, Force/Vision Control, Robotic Cutting, Flexible Robots.