UNIVERSITÉ DE MONTRÉAL

ROBOTIC JOINT-MOTION OPTIMIZATION OF FUNCTIONALLY-REDUNDANT TASKS FOR JOINT-LIMITS AND SINGULARITY AVOIDANCE

LIGUO HUO
DÉPARTEMENT DE GÉNIE MÉCANIQUE
ÉCOLE POLYTECHNIQUE DE MONTRÉAL

THÈSE PRÉSENTÉE EN VUE DE L'OBTENTION
DU DIPLÔME DE PHILOSOPHIÆ DOCTOR
(GÉNIE MÉCANIQUE)
AVRIL 2009

Your file Votre référence
ISBN: 978-0-494-49418-9
Our file Notre référence
ISBN: 978-0-494-49418-9

UNIVERSITÉ DE MONTRÉAL

ÉCOLE POLYTECHNIQUE DE MONTRÉAL

Cette thèse intitulée :

ROBOTIC JOINT-MOTION OPTIMIZATION OF FUNCTIONNALY-REDUNDANT TASKS FOR JOINT-LIMITS AND SINGULARITY AVOIDANCE

présenté par : HUO, Li Guo

en vue de l'obtention du diplôme de : Philosophiae Doctor

a été dûment accepté par le jury d'examen constitué de :

M. MASCLE Christian, Doct. ès sciences, président

M. BARON Luc, Ph.D., membre et directeur de recherche

M. GOURDEAU Richard, Ph.D., membre

M. CARRETARO Juan Antonio, Ph.D., membre

To my father...

ACKNOWLEDGEMENTS

I would like to thank my supervisor, Prof. Luc Baron. He directed me to the domain of robotics and provided the advice, encouragement, and enthusiasm that I needed. Thanks are also due to my industrial supervisor Chahe Bakmazjian, who shared with me his vast knowledge and experience on industrial robotic application.

The financial support from Jabez Technologies Inc. and the Natural Sciences and Engineering Research Council(NSERC) are gracefully acknowledgement. Their support really give me the chance to complete this work.

Finally, I am very grateful to my wife, Haiying Huang, for her love, support and understanding throughout my research. I know, I would not have been able to finish this thesis without her support.

RÉSUMÉ

L'objectif de recherche de cette thèse est de développer un algorithme de résolution de la redondance (RR) afin d'optimiser la trajectoire dans l'espace articulaire d'un robot industriel devant effectuer des tâches manufacturières.

La plupart des opérations d'usinage tel que le soudage, l'ébavurage ou le fraisage ont un axe de symétrie. Il est clair que la rotation de l'outil autour de l'axe de symétrie n'a pas d'effet sur la tache à accomplir. Si la tâche est effectuée avec un robot industriel possédant six axes de rotation, il y a un degré de liberté (DDL) redondant qui permet potentiellement d'optimiser. Cette sorte de redondance est appelée une redondance fonctionnelle, ce qui contraste avec la redondance intrinsèque bien connue par la plupart des chercheurs. Les tâches qui sont fonctionnellement redondantes sont très communes dans le domaine de la robotique industrielle mais demeurent ignorées par la plupart des chercheurs.

Concernant les exigences pour l'utilisation de robots industriels dans les opérations manufacturières, cette thèse propose une nouvelle approche dans la résolution de la redondance pour les tâches robotiques fonctionnellement redondantes. Cette approche est appelée méthode de décomposition du torseur de vitesse (Twist Decomposition Approach) (TWA). Au lieu de projeter un critère d'optimisation sur l'espace nul de la matrice Jacobienne tel que le font la majorité des schémas de résolution de la redondance, la méthode TWA décompose premièrement le torseur de vitesse cartésien de l'effecteur en deux sous-espaces ; l'un étant le sous-espace dans lequel la tâche principale s'effectue, tandis que l'autre est l'espace redondant. La tâche peut alors être optimisée dans le sous-espace redondant. Dans cette thèse, les limites des joints ainsi que l'évitement des singularités sont considérés comme étant les deux principaux objectifs d'optimisation.

Il a été démontré que le TWA est capable d'optimiser de manière efficace la trajectoire dans l'espace articulaire pour diverses tâches et divers robots industriels. Les applications possibles de TWA inclus le soudage, le fraisage, l'ébavurage, la peinture,

la coupe au laser et bien d'autres tâches qui requièrent moins de six degrés de liberté au repère de l'outil.

Afin de prendre pleinement avantage du potentiel de TWA, il y a une question cruciale qui doit être abordée. Ce sont les pondérations qui ont un rôle d'équilibre entre les sous-tâches et la contribution de chaque articulation dans l'optimisation. Étant donné que les pondérations ont une grande influence sur l'optimisation, et aussi sur le succès ou l'échec d'une tâche, la question de l'adaptation des pondérations mérite une étude plus approfondie. Dans cette thèse, deux méthodes d'adaptation des pondérations sont proposées, la méthode d'auto-adaptation et la méthode d'adaptation dynamique. Les deux méthodes identifient premièrement la sensibilité des pondérations. La méthode d'auto-adaptation utilise une méthode de recherche linéaire pour adapter les pondérations, tandis que la méthode d'adaptation dynamique développe quelques fonctions empiriques pour adapter dynamiquement les pondérations à chaque instant sur la trajectoire. Les deux méthodes fonctionnent dans des tâches variées. Par contre, la méthode d'adaptation dynamique a un éventail d'applications plus large et atteint de meilleurs résultats d'optimisation étant donné que les pondérations sont adaptées selon les besoins à chaque instant au lieu de les conserver fixés à certaines valeurs tel que dans la méthode d'auto-adaptation.

Cette thèse est composée de sept chapitres. Le chapitre 1 introduit le concept de tâches fonctionnellement redondantes et présente les tâches fonctionnellement redondantes comme étant commune dans le monde de la robotique industrielle.

Le chapitre 2 fait une revue des travaux de recherche sur la résolution de la redondance cinématique, incluant les approches d'optimisation locale et globale, et l'application de techniques de contrôle intelligentes. D'autres résolutions de la redondance fonctionnelle développée sont présentées.

Le chapitre 3 présente et applique le TWA aux problèmes d'évitement des limites articulaires, alors qu'au chapitre 4, l'évitement des singularités est ajouté en parallèle à l'évitement des limites des articulaires aux objectifs d'optimisation de la trajectoire du robot dans l'espace articulaire.

Les chapitres 5 et 6 proposent deux méthodes différentes pour l'adaptation des pondérations, l'auto-adaptation au chapitre 5 et l'adaptation dynamique au chapitre 6.

Finalement, la conclusion est présentée au chapitre 7.

ABSTRACT

The research objective of this thesis is to develop a redundancy-resolution (RR) algorithm to optimize the joint space trajectory of six-revolute industrial robot as performing manufacturing tasks.

Most of machining operations, such as welding, deburring or milling, have a symmetry axis. Clearly, the rotation of the tool around the symmetry axis is irrelevant to the view of the task to be accomplished. If the task is performed with a six-rotation-axis industrial robot, there is one *degrees of freedom* (DOF) of kinematic redundancy, which provides the potential of optimization. This kind of redundancy is called as *functional redundancy*, with contrast to intrinsic redundancy well known by most researchers. Functionally-redundant tasks have very common existence in the industrial robotic field, but still are ignored by most researchers.

Concerning the requirement for applying industrial robot in manufacturing, this thesis proposes a new redundant-resolution approach to solve functionally-redundant robotic tasks. This approach is called *Twist Decomposition Approach* (TWA). Instead of projecting an optimization criterion onto the null space of the Jacobian matrix as most of the redundancy-resolution schemes do, TWA firstly decomposes the Cartesian twist of the end-effector into two suitable subspaces; one being the subspace where the main task undergoes, while the other one being the redundant subspace. Then, the task can be optimized on the redundant subspace of the twist. In this thesis, joint-limits and singularity avoidance are considered as the two main optimization objectives.

TWA has been demonstrated to be able to optimize effectively the joint space trajectory for various tasks and various types of industrial robots. The possible application of TWA include welding, milling, deburing, painting, laser cutting and many other tasks requiring less than six-DOF in tool frame.

In order to take the full advantage of TWA's potential, there is a critical issue which need to be addressed. It is the weights that play the role of balancing among the

subtasks and contribution of each joint in the optimization. Since the weights have such a great influence on the optimization, and even the task success or failure, the weights adaptation issue deserves more study. In this thesis, two weights adaptation methods are proposed, namely the self-adaptation and dynamic-adaptation methods. Both methods identify sensitivity of weights component firstly. Self-adaptation method proposes the use of a linear space searching method to adapt weights, while dynamic-adaptation method develops some empirical functions to dynamically adapt weights at each instant of the trajectory. Both methods succeed in various tasks. However, dynamic-adaptation method has greater application range and reaches better optimization results, since weights are adapted according to the need of each instant (dynamic-adaptation method), instead of keeping them fixed at certain value (self-adaptation method).

This thesis is composed of seven chapters. Chapter 1 introduces the concept of functionally-redundant tasks, and presents the common existence of the functionally-redundant tasks in the industrial robotic field.

Chapter 2 reviews the research works on kinematic redundancy resolution, including the local and global optimization approaches, and the application of intelligent control techniques. Some other developed functional redundancy resolutions are also introduced.

In Chapter 3, TWA is presented and applied on avoiding joint limits problem, while in Chapter 4, the singularity avoidance is added into the optimization objective of the robot joint space trajectory besides the joint limit avoidance.

Chapters 5 and 6 propose two different methods on adapting weights, self-adaptation method in Chapter 5, and dynamic-adaptation method in Chapter 6.

Finally, the conclusions and future works are presented in Chapter 7.

CONDENSÉ EN FRANÇAIS

0.1 Introduction

Dans cette thèse, les sources de redondance cinématique d'un manipulateur sériel sont classées dans deux catégories : la redondance intrinsèque et la redondance fonctionnelle. Ces dernières sont définies au chapitre 1 (voir la définition 1.1, 1.2 et 1.3 en pages 6 et 9).

Reliées à la redondance intrinsèque, les méthodes de résolution de la redondance (RR) sont classifiées en méthodes optimales globales et locales. Le problème de RR a été également résolu au niveau des déplacements et des vitesses. La plupart des chercheurs ont travaillé au niveau de la vitesse et ont employé l'inverse généralisée de Moore-Penrose ou l'inverse généralisée pondérée de la matrice Jacobienne. Les méthodes employant l'inverse généralisée utilisent soit la solution de norme minimale (voir l'éq.(2.23)) soit la solution de norme non-minimale (voir l'éq.(2.25)). La solution de norme non-minimale ajoute une composante homogène à la solution de norme minimale. La composante homogène projette un vecteur arbitraire sur le noyau (ou espace nul) de la matrice Jacobienne. L'équation (2.25) est largement utilisée par les chercheurs pour résoudre des tâches redondantes.

Dans le cas de la redondance fonctionnelle, les méthodes de RR intrinsèques travaillant dans l'espace nul de la matrice Jacobienne ne peuvent être directement employées car la dimension de l'espace nul de la Jacobienne est égale à zéro. Afin d'obtenir un système sous-déterminé, il y a deux possibilités : augmenter la dimension du vecteur de vitesse articulaire, ou réduire la dimension du torseur de vitesse de l'effecteur. Correspondant aux deux possibilités, les méthodes de RR fonctionnelle peuvent être classifiées en deux groupes : l'approche augmentée et l'approche réduite. La méthode de l'articulation virtuelle selon l'approche augmentée et la méthode d'élimination selon l'approche réduite sont passées en revue au chapitre 2.

Dans cette thèse, on propose une nouvelle méthode de RR fonctionnelle qui est basée

sur la décomposition du torseur de vitesse. Cette méthode ne nécessite pas la projection sur l'espace nul de la matrice Jacobienne. La méthode de décomposition du torseur de vitesse a une applicabilité potentielle à toutes les tâches de moins de six DDL quelque soit le nombre de DDL du manipulateur.

0.2 La méthode de décomposition du torseur de vitesse (TWA) et évitement des limites articulaires

Tous les vecteurs de $\mathbb{R}^3$ peuvent être décomposés en deux parties orthogonales en utilisant le projeteur $\mathbf{M}$ et son complément orthogonal $\mathbf{M}^{\perp}$. Les projeteurs sont donnés pour les quatre dimensions possibles du sous-espace de $\mathbb{R}^3$: de la tâche de dimension zéro à la tâche de 3 dimensions (voir l'éq.(3.2)).

Selon la décomposition orthogonale des vecteurs, n'importe quelle rangée du torseur de l'espace $2 \times \mathbb{R}^3$ peut également être décomposée en deux parties orthogonales en utilisant le projeteur $\mathbf{T}$ et son complément orthogonal $\mathbf{T}^{\perp}$. Les projeteurs de torseurs de vitesse sont définis par une matrice diagonale des deux projeteurs des vecteurs de $\mathbb{R}^3$ (voir l'éq.(3.6)). Pour les manipulateurs séries fonctionnellement redondants, il est possible de décomposer le torseur de vitesse de l'effecteur en deux parties orthogonales dans le sous-espace tâche et le sous-espace redondant. Par conséquent, la décomposition orthogonale du torseur de vitesse peut se substituer à la solution de norme minimale de la cinématique inverse du manipulateur redondant (voir l'éq.(3.8)). La première partie du côté droit de l'éq.(3.8) atteint le déplacement articulaire exigé par la tâche, alors que la deuxième partie atteint le déplacement articulaire dans le sous-espace redondant. L'équation (3.8) est la contribution originale principale de cette thèse. Elle n'exige pas la projection sur l'espace nul de la matrice Jacobienne comme la majorité des algorithmes de résolution de redondance, mais exige plutôt une projection orthogonale basée sur la géométrie instantanée de la tâche à accomplir.

Puisque les deux projeteurs de vecteur dans la matrice de projeteur de torseur ont quatre dimensions possibles dans $\mathbb{R}^3$, le projeteur de torseur a 16 possibilités comme montré au Tableau 3.6. Selon la condition de la tâche, un projeteur différent du torseur sera choisi pour décomposer ce dernier. L'algorithme 3.1 montre la résolution de redondance fonctionnelle avec la méthode de décomposition du torseur de vitesse. Trois exemples sur différent manipulateurs ont montré les applications de TWA pour éviter les limites articulaires.

0.3 Évitement des limites articulaires et singularités

Afin d'éviter non seulement les limites articulaires, mais aussi les singularités même temps, un nouveau critère de performance relatives à ces dernières, nommé *paramètre de Singularity*, est proposé et analysé dans ce chapitre. Pour éviter les limites articulaires, le critère de performance peut être écrit comme l'éq.(4.1) pour maintenir le manipulateur le plus proche possible de la position milieu $\bar{\boldsymbol{\theta}}$ articulaire .

Il y a deux critères bien connus pour la détection de la configuration singulière, la manipulabilité ω_{mom} et le conditionnement ω_{cond}. Comme ω_{mom} ne représente que le volume de l'éllipsoïde de manipulabilité, alors que ω_{cond} représente seulement la forme, un indice de performance, qui représente à la fois le volume et la forme de l'éllipsoïde manipulabilité, est proposé et nommé paramètre de Singularity[68], définie comme l'éq.(4.6).

Afin de détecter la singularité, une valeur seuil de singularité ω_o est fixé. Lorsque ω_{ps} dépasse ω_o, la configuration correspondante à cet instant est enregistrée en tant que $\boldsymbol{\theta}_{Ts}$. Le critère de performance écrit comme l'éq.(4.7) est alors activé afin de maintenir le manipulateur aussi proche que possible de $\boldsymbol{\theta}_{Ts}$ devant toute l'étapes suivante, jusqu'à ce que le ω_{ps} soit inférieur à la valeur seuil ω_o.

Enfin, la fonction objectif pourrait être écrite comme l'éq.(4.8) avec l'évitement des limites articulaires et des singularités comme tâches secondaires, et l'éq.(3.8) pourrait être reformulé comme l'éq.(4.10).

Les tests numérique démontrent le bon fonctionnement de l'optimisation par l'évitement

des singularités, des limites articulaires, et montrent la grande influence de la pondération des critères sur l'optimization.

0.4 Auto-adaptation des poids en TWA

Les résultats obtenus par la méthode TWA sont aussi très sensible au choix des pondérations tel que la méthode classique GPM. Leur succès s'appuient sur le vecteur de pondération (l'importance relative de la tâche secondaire par rapport à la tâche principale), qui doit être ajusté pour que le processus d'optimisation réussisse. S'il est mal choisi, la tâche principale peut échouer. Jusqu'à présent, le vecteur est habituellement choisi par la méthode d'essai-erreur.
Un système d'auto-adaptation des pondérations est proposé. Ce système est composé de trois componsantes principales décrits ainsi :

1. Identification de la sensibilité des pondérations (IWS) ;
2. Algorithme d'évaluation des performances des trajectoires articulaires (JT-PEA) ;
3. Méthode de recherche linéaire (LSM) dans l'espace convexe 2D ;

L'ensemble de la procédure de réglage est illustré à l'algorithme 5.1. (en page 81).

0.4.1 Identification de la sensibilité des poids

La plupart des manipulateurs industriels série 6-axes sont découplés, et par conséquent, l'orientation d'effecteur est contrôlée par les trois dernières articulations, c'est-à-dire, de la 4ième à la 6ième articulations. Dans le cas de la rotation redondante d'effecteur, les pondérations des trois dernières articulations ont plus d'influence sur le movement redondant que les autres pondérations. Selon le rapport géométrique entre l'axe de symétrie de l'outil et l'axe de rotation de l'articulation, l'influence de chacune des articulations sur le movement redondant peut être identifiée. Enfin, le problème de recherche d'espace 6-D est simplifié à un problème d'espace 2-D convexe .

L'influence de ces trois poids sur la rotation autour de l'axe de l'articulation redondante $\mathbf{e}_t$ dépend de l'orientation relative entre $\mathbf{e}_6$ et $\mathbf{e}_t$, comme ci-après,

- Cas 1 : $\mathbf{e}_t \perp \mathbf{e}_6 \Rightarrow$ 6ième n'est pas sensible $\Rightarrow$ *Adapter 5ième et 4ième.*
- Cas 2 : $\mathbf{e}_t \parallel \mathbf{e}_6 \Rightarrow \mathbf{e}_t \perp \mathbf{e}_5 \Rightarrow$ 5ième n'est pas sensible $\Rightarrow$ *Adapter 6ième et 4ième.*
- Cas 3 : Ni $\mathbf{e}_t \perp \mathbf{e}_6$ ni $\mathbf{e}_t \parallel \mathbf{e}_6 \Rightarrow$ *Tous les trois sont pertinents.*

0.4.2 Algorithme d'évaluation des performances des trajectoires articulaires

Le processus d'évaluation est présente à la Fig. 5.1. Une fois les trajectoires articulaires obtenues par résolution de la cinématic inverse, la plus grande distance par rapport à la limite articulaire et les pires paramètre qui rendent le manipulateur singulier le long des trajectoires sont pris comme critères de performance de cette solution dans l'espace articulaire.

0.4.3 Méthode de recherche linéaire

Dans cette méthode, la dimension de l'espace de recherche est toujours deux, car à chaque fois seul un des éléments de $\mathbf{w}$ et de $\mathbf{k}$ sont réglés. La recherche de l'espace est supposé être 2D convexe selon nos expériences. Par conséquent, une simple méthode de recherche linéaire directe est effectuée pour rechercher la valeur minimale. La Fig. 5.2 montre la processus de recherche dans un espace 2-D convexe avec le point le plus bas inconnu.

0.4.4 Exemple numérique

Avec trois exemples d'applications, le système d'auto-adaptation est bien démontré et vérifié. Tous les exemples ont été mis en œuvre sur Robotmaster[71], un logiciel de programmation robotique.

- Exemple 1 : $\mathbf{e}_t \perp \mathbf{e}_6$ (en pages 88 et 89) ;
- Exemple 2 : $\mathbf{e}_t \parallel \mathbf{e}_6$ (en page 90) ;
- Exemple 3 : ni $\mathbf{e}_t \parallel \mathbf{e}_6$ ni $\mathbf{e}_t \perp \mathbf{e}_6$ (en pages 91 et 92).

0.5 Adaptation dynamique des pondération en TWA

Étant donné que les différents segments d'une tâche demandent généralement des optimisations différentes, la méthode d'auto-adaptation ne peut pas répondre à cette exigence, les pondérations sont fixés à la même valeur sur toute la trajectoire. Dans ce chapitre, un système d'adaptation dynamique des pondérations pour la redondance fonctionnelle est développé pour remplacer la méthode peu efficace qui fixe la valeur de l'adaptation. La système d'adaptation dynamique des pondérations est intégré dans le TWA développé, et est appliqué dans le cas de multiples tâches secondaires qui examinent non seulement l'optimisation des limites articulaires et de la singularité, mais aussi des vitesses articulaires.

L'évaluation de la fonction d'adaptation joue un rôle clé dans le système entier. Les fonctions sont développées dans la section 6.3. L'algorithme d'adaptation dynamique est présenté dans l'algorithme 6.1 (en page 108).

Afin d'atteindre ces objectifs, nous devons résoudre les trois problèmes suivants :

- Quelles sont les variables influant sur les pondérations de l'adaptation ?
- Quelle est la relation entre les pondérations et les variables d'entrée ?
- Comment exprimer la fonction d'adaptation sous forme mathématique ?

Il y a deux vecteurs et un scalaire variables influençant les pondérations d'adaptation. Ce sont des vecteurs en trois dimensions, $\boldsymbol{\beta}$, qui détecte l'influence des trois dernières articulations sur les déplacements redondants ; un vecteur $\boldsymbol{\delta}$ de six dimensions, qui mesure la distance de la position articulaire actuelle à ses limites ; et l'échelle ω_{mom}, qui mesure la distance de la configuration actuelle à la singularité.

À partir de la géométrie de l'outil symétrique entre l'axe $\mathbf{e}_t$ et l'axe de rotation $\mathbf{e}_i$, nous pouvons exprimer l'influence de $i$ième articulation par $\beta_i = \arccos(|\mathbf{e}_i^T \mathbf{e}_t|)$.

Comme l'angle β_i est en baisse à 0, la rotation autour de $\mathbf{e}_t$ devient sensible à la valeur des pondérations w_i.

Par expérimentation, nous constatons que les deux fonctions d'adaptation f_{joint} et f_{sing} sont des cas dépendants et ont de grandes différences selon les diverses applications. C'est-à-dire, il n'existe pas de fonction générale unique de f_{joint} et f_{sing} selon notre

expérience. Avec une série d'équations développées à partir de eq.(6.7) à (6.12), les pondérations **w** et **k** peuvent être adaptés dynamiquement en fonction des exigences instantanées.

Le long de la trajectoire essayé, l'instant où les indices de performance sont les pires est appellé l'instant critique de cette trajectoire. Afin d'affiner **w** et **k** pour répondre aux critères d'optimisation autour de cet instant critique, c_{joint} et c_{sing} des eqs.(6.7 6.8) sont adaptées sur la base des trois principes suivants :

- augmentation de c_{joint} afin de réduire au minimum les déplacements des l'articulations ;
- augmentation de c_{sing} pour mettre le manipulateur hors de la singularité ;
- réduction de c_{joint} ou c_{sing} pour diminuer la vitesse articulaire

Avec les exemples d'applications, le système d'adaptation dynamique des pondérations est en mesure d'atteindre des trajectoires de l'espace articularie qui sont plus optimisé que dans le système d'auto-adaptation.

0.6 Conclusion

La méthode de décomposition du torseur de vitesse peut atteindre une trajectoire plus lisse et plus précise que la méthode d'articulation virtuelle pour notre tâche. Bien que la méthode de décomposition du torseur de vitesse peut seulement résoudre des problèmes de redondance fonctionnelle, elle reste néanmoins significative et intéressante puisqu'un nombre élevé de tâches fonctionnelles redondantes existe dans le milieu industriel, telle que la soudure, la pulvérisation, le fraisage, le découpage par jet d'eau et le découpage au laser, etc.

Dans cette thèse, la méthode de décomposition du torseur de vitesse n'est pas seulement utilisée pour éviter le problème posé par les limites articulaires. Elle est utilisée pour éviter les singularités et les hautes vitesses des articulations.

Pendant le processus d'optimisation, nous notons que les choix de la posture initiale et de vecteur de pondération affectent considérablement la trajectoire optimisée. Un mauvais choix peut même causer l'échec total de l'optimisation. En fait, ces choix

sont fondés, la plupart du temps, sur l'expérience. Alors, deux méthodes d'adaptation des vecteurs pondération sont développées. Grâce aux développements, la méthode de décomposition du torseur de vitesse est plus efficace et est capable d'obtenir de meilleurs résultats.

CONTENTS

LIST OF FIGURES

LIST OF NOTATIONS AND SYMBOLS

DH : Denavit-Hartenberg ;

DKP : direct kinematics problem ;

DOF : degrees of freedom ;

EE : end-effector ;

GI : generalized inverse ;

GPM : gradient projection method ;

GS : graphics simulation ;

IKP : inverse kinematics problem ;

RR : redundancy resolution ;

FRR : functional redundancy resolution ;

TWA : twist decomposition approach ;

NN : neural network ;

GA : genetic algorithm ;

FL : fuzzy logic ;

FRMRA : fuzzy resolved motion rate algorithm ;

MRRA : motion-rate resolving algorithm ;

MOM : measure of manipulability ;

DKP : function solving DKP ;

IKP : function solving IKP ;

WGI : weighted generalized inverse ;

IWS : identification of weights sensitivity ;

JTPEA : joint trajectory performance evaluation algorithm ;

LSM : linear search method ;

$\mathcal{J}$: joint space ;

$\mathcal{T}$: task space ;

$\mathcal{O}$: operational space ;

n : dimension of the joint space ;

o : dimension of the operational space;

t : dimension of the task space;

r_I : degree of intrinsic redundancy;

r_F : degree of functional redundancy;

r_K : degree of kinematic redundancy;

$\mathcal{F}_i$: frame attached to body i;

$\mathcal{F}_A$: frame attached to the base;

$\mathcal{F}_B$: frame attached to the EE;

$\mathbf{p}$: position vector of the origin of $\mathcal{F}_B$ with respect to $\mathcal{F}_A$;

$\mathbf{Q}$: rotation matrix of $\mathcal{F}_B$ with respect to $\mathcal{F}_A$;

$\boldsymbol{\omega}$: angular velocity vector of the EE;

$\mathbf{t}$: twist array expressing the end-effector velocity as $\mathbf{t} \equiv \begin{bmatrix} \boldsymbol{\omega}^T & \dot{\mathbf{p}}^T \end{bmatrix}^T$;

${}^{\mathcal{F}_j}\mathbf{A}_{\mathcal{F}_i}$: homogeneous transformation expressing the pose of $\mathcal{F}_i$ with respect to frame $\mathcal{F}_j$;

vect : function transforming a 3×3 rotation matrix into axial vector;

$\mathbf{J}$: Jacobian matrix;

$\mathbf{A}$: the upper three rows of $\mathbf{J}$ pertaining to the angular velocity component;

$\mathbf{B}$: the lower three rows of $\mathbf{J}$ pertaining to the translational velocity component;

$\boldsymbol{\theta}$: joint position vector;

$\dot{\boldsymbol{\theta}}$: joint velocity vector;

$\mathbf{W}$: positive-definite weighting matrix;

$\mathbf{M}$: projector;

$\mathbf{P}$: plane projector;

$\mathbf{L}$: line projector;

$\mathbf{T}$: twist projector;

LIST OF TABLES

CHAPITRE 1

INTRODUCTION

A *manipulator* is a device that helps human beings to perform manipulating tasks. A *robotic manipulator* is to be distinguished from the previous for its ability to lead itself through computer control. Once programmed, it can implement the same task repeatedly. In general, robotic manipulators can be studied using the concept of *kinematic chain.* A kinematic chain is a set of rigid bodies, also called links, coupled by *kinematic pairs.* A kinematic pairs is the coupling of two rigid bodies so as to constrain their relative motion. There are two basic types of kinematic pairs, namely, *upper* and *lower* kinematic pairs. Upper kinematic pairs occur through either line contact or point contact between two bodies, and thus, appear in cam-and-followers, gear trains, and roller bearings, for example. Lower kinematic pairs occur through a common surface between two bodies[1]. As shown in Fig. 1.1, there are six lower kinematic pairs, namely, (E) Planar, (S) Spherical, (C) Cylindrical, (R) Revolute, (P) Prismatic, and (H) Helicoidal.

As shown in Fig. 1.2 and classified in Table 1.1, kinematic chains are termed either *simple* or *complex*; and either *open* or *closed.* Simple chains are those having links with a *degree of connectivity*[1] of less than or equal to two, while complex chains are those having at least one link with a degree of connectivity greater than two. Open and closed simple kinematic chains occur in *serial manipulators* and *linkages*, respectively. Open complex kinematic chains are sets of open kinematic chains in a tree-type structure, and occur therefore in tree-type manipulators. Closed complex kinematic chains are termed either *parallel* or *hybrid* depending on whether the kinematic loops lie in parallel arrays or not. Closed complex kinematic chains with loops in parallel

[1]The *degree of connectivity* of a body is defined as the number of bodies directly connected to the said body through kinematic pairs.

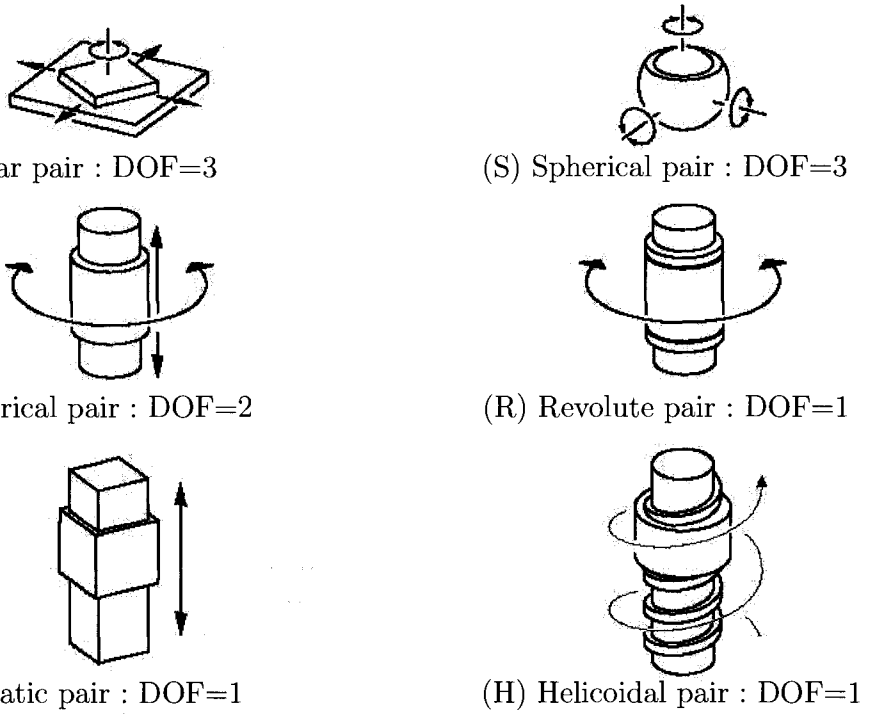


FIG. 1.1 The six lower kinematic pairs

arrays occur in *parallel manipulators*, while all the other kinematic chains are classified as *hybrid manipulators*, *i.e.*, those containing either more than two links with a degree of connectively greater than two, or those containing both closed and open kinematic chains. Figure 1.3 presents four examples of different manipulators, which include (a) a serial manipulator, (b) a parallel manipulator, (c) a robotic hand and (d) a walking machine. Clearly, both the robotic hand and the walking machine can be classified as tree-type manipulators.

In this thesis, we focus on serial manipulators, *i.e.*, simple open kinematic chains. In such manipulators, there are exactly two bodies with a degree of connectivity of one, called end-bodies, and all other bodies within the chain have a degree of connectivity equal to two. One end-body is arbitrary regarded as fixed and is called the *base*, while the other end-body is regarded as movable and is called the moving body, or

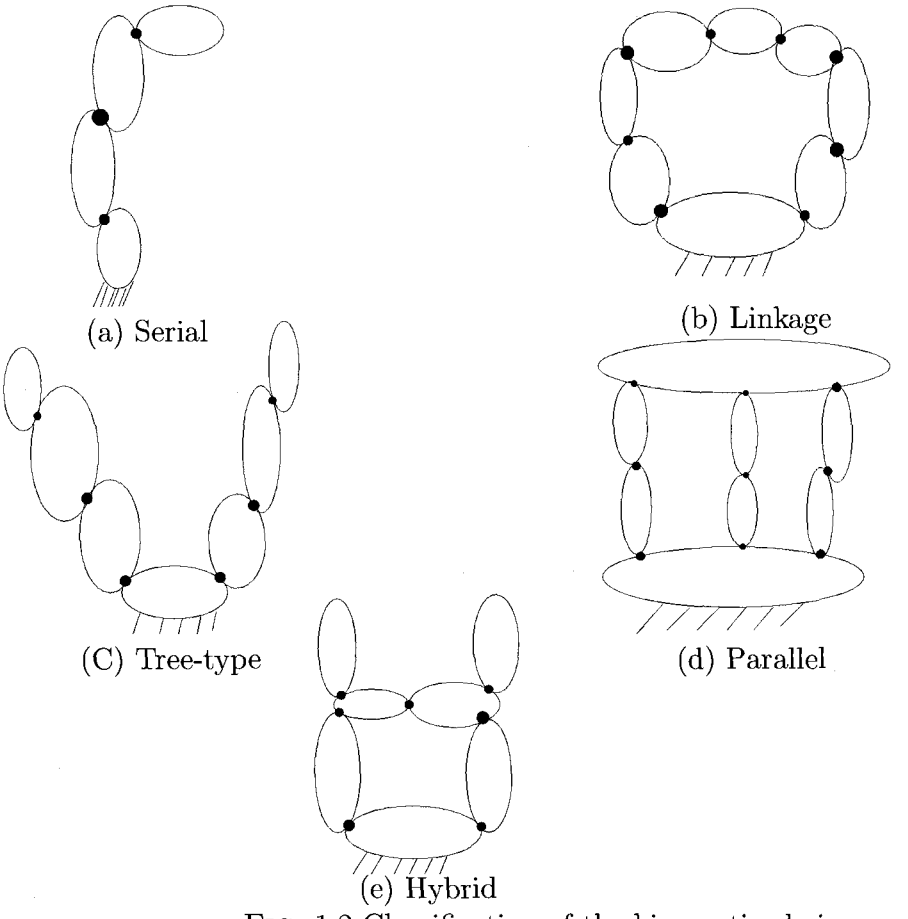


FIG. 1.2 Classification of the kinematic chains

the *end-effector* (EE) of the manipulator.

1.1 Background and Basic Terminology

1.1.1 Degrees of Freedom of a Mechanical System

In general, the minimum number of variables (also called coordinates) to completely specify the configuration of a mechanical system is called the degrees of freedom (DOF) for that system. For a serial manipulator, each independent variables is typically associated with a joint i, *e.g.*, θ_i. Since, the DOF related to a serial manipulator

	Open	Closed
Simple $C_{max} \leq 2$	Serial Manipulator $C_i = 1,\ i = 1,2$ $C_i = 2,\ i > 2$	Linkage $C_i = 2$
Complex $C_{max} > 2$	Tree-type manipulator $C_{max} > 2$ with no loop	Parallel manipulator $C_i > 2,\ i = 1,2$ $C_i = 2,\ i > 2$
		Hybrid manipulator all other cases

TAB. 1.1 Classification of the kinematic chains (C_i is the degree of connectivity of link i and C_{max} is the maximum degree of connectivity of all links).

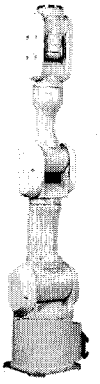
is the sum of DOF of each joint, the combination of the joint positions of a manipulator is referred to as a *posture*. We can combine the joint position into a vector, namely $\boldsymbol{\theta}$, given by

$$\boldsymbol{\theta} \equiv [\theta_1 \cdots \theta_n]^T \in \mathcal{J}^n, \tag{1.1}$$

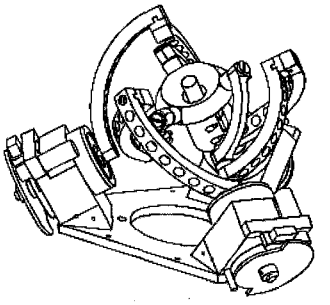
where $\mathcal{J}$ is the *Joint space* in which $\boldsymbol{\theta}$ is defined, and its dimension n is given by $n = dim(\mathcal{J})$, therefore n is the DOF of the manipulator.

1.1.2 Degrees of Freedom of the End-Effector

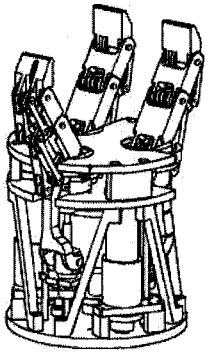
The combination of the position and the orientation of a rigid body is referred to as the *pose* of the corresponding body. For a rigid body freely moving in three-dimensional space, a minimum of six coordinates are required to completely define its mobility or DOF. At the displacement level, the pose of a rigid body can be defined by the position of a point of the body together with the orientation of the body around that point. The position of a point of the body can be defined by specifying its three Cartesian coordinates in some convenient coordinate frame, *e.g.*, p_x, p_y and p_z. Similarly, the orientation of the body around that point can be defined by three angles in the same coordinate frame, *i.e.*, θ_x, θ_y and θ_z. We can combine the six elements into one array,



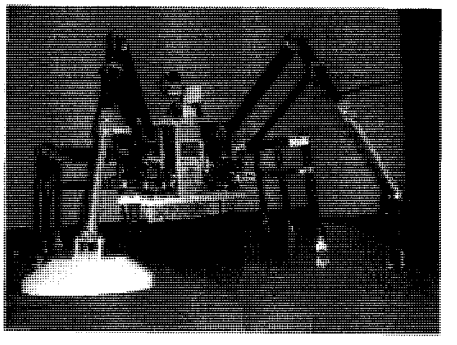
a. Serial manipulator



b. Parallel manipulator



c. Robotic hand



d. Walking machine

FIG. 1.3 Classification of the robotic manipulators : (a) PA10-6C from MITSUBISHI[2] ; (b) Agile Eye from Laval University[3] ; (c) SARAH Robotic Hand from Laval University[4] ; (d) COMET-II from Chiba University[5].

namely $\mathbf{x}$, given by

$$\mathbf{x} \equiv \begin{bmatrix} \theta_x & \theta_y & \theta_z & p_x & p_y & p_z \end{bmatrix}^T. \tag{1.2}$$

Consequently, the motion of the EE can be defined by the motion of $\mathbf{x}$. The space in which the EE undergoes its motion is usually called the *operational space*, denoted by $\mathcal{O}$, and its dimension o is given by $o = dim(\mathcal{O})$.

For a specific task, the motion of the EE may require the whole operational space $\mathcal{O}$ or only a subspace of $\mathcal{O}$. In both cases, we call the space in which the task is undergoing, the *task space* $\mathcal{T}$, its dimension t is given by $t = dim(\mathcal{T})$, where $t \leq o$,

since $dim(\mathcal{T}) \leq dim(\mathcal{O})$. In all cases, we must have $n \geq o \geq t$ with $\mathcal{T} \subseteq \mathcal{O}$, otherwise the task can not be performed by the manipulator.

1.1.3 Kinematic Redundancy

The main focus of this thesis belongs to kinematic redundancy.

Definition 1.1 : Kinematic redundancy

A pair made of a serial manipulator and a task is said to be kinematically redundant when the dimension of the joint space $\mathcal{J}$, denoted by $n = \dim(\mathcal{J})$, is greater than the dimension of the task space $\mathcal{T}$ of the EE, denoted by $t = \dim(\mathcal{T}) \leq 6$, while the task space being totally included into the resulting operation space of the manipulator, i.e., *$\mathcal{T} \subseteq \mathcal{O}$, and hence, $n > t$. The degree of kinematic redundancy of a pair of serial manipulator-task, namely r_K, is computed as*

$$r_K = n - t. \tag{1.3}$$

In a system with kinematic redundancy, it is possible to change the internal structure or configuration of the mechanism without changing the position and orientation of its EE[7], a so-called self-motion of the manipulator. In this thesis, the term "redundancy" refers to "kinematic redundancy".

A typical example of a redundant manipulator is the human arm, which has approximately seven DOF from the shoulder to the wrist. If the base and the hand position and orientation are both fixed, requiring six DOF; the elbow can still be moved, due to the additional mobility associate with the redundant DOF. Thus, it becomes possible to avoid obstacles in the workspace. Furthermore, if a joint of a redundant manipulator reaches its mechanical limit, there might be other joints that allow execution of the same prescribed EE motion.

1.2 Problem Formulation

Many redundant manipulators have been developed so far. Some were developed for research purposes, while others have already been used in real applications. However, the control of a redundant manipulator is more challenging, since there are infinitely many joint trajectories that exist for a given task. The operator must evaluate the best one according to a performance criterion. The solution strategy, which effectively exploits the potential advantages of kinematically redundant mechanisms, is called : *Redundancy Resolution* (RR) scheme. Since there is no general RR scheme solution, RR scheme has attracted the attention of many researchers for at least three decades.

Redundancy is a concept related to the definition of the task instead of being an intrinsic feature of the robot's structure. Even if a manipulator is kinematically redundant for a specific task, it may not be redundant for another task. Hence, according to the relation among joint space, operational space and task space, kinematic redundancy can be classified into two groups, *i.e.*, *functional redundancy* and *intrinsical redundancy*, for which we have purposed the following two definitions.

Definition 1.2 : Intrinsic redundancy

A serial manipulator is said to be intrinsically redundant when the dimension of the joint space $\mathcal{J}$, denoted by $n = \dim(\mathcal{J})$, is greater than the dimension of the resulting operational space $\mathcal{O}$ of the EE, denoted by $o = \dim(\mathcal{O}) \leq 6$, i.e., *$n > o$. The degree of intrinsic redundancy of a serial manipulator, namely r_I, is computed as*

$$r_I = n - o \tag{1.4}$$

Definition 1.3 : Functional redundancy

A pair made of a serial manipulator and a task is said to be functionally redundant when the dimension of the operational space $\mathcal{O}$ of the EE, denoted by $o = \dim(\mathcal{O}) \leq$

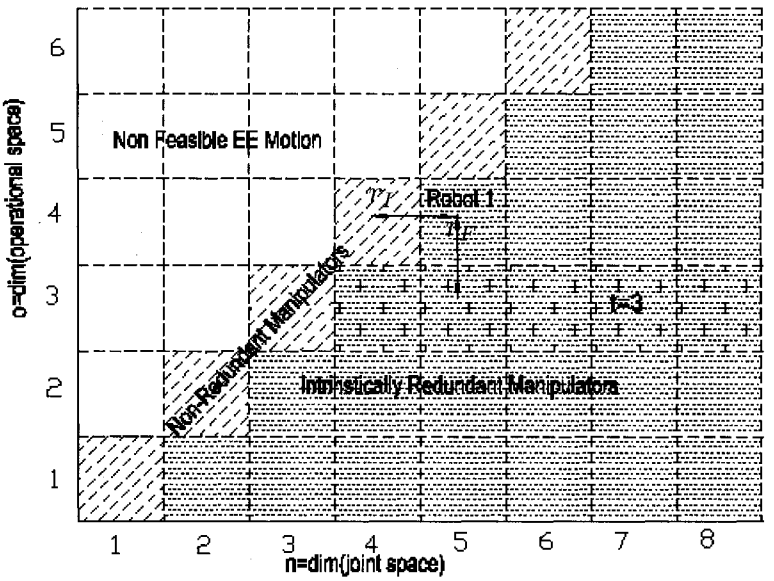


FIG. 1.4 Intrinsic and functional redundancies of serial robotic tasks

6, *is greater than the dimension of the task space* $\mathcal{T}$ *of the EE, denoted by* $t = \dim(\mathcal{T}) \leq 6$, *while the task space being totally included into the operation space of the manipulator,* i.e., $\mathcal{T} \subseteq \mathcal{O}$, *and hence,* $o > t$. *The degree of functional redundancy of a serial manipulator-task pair, namely* r_F, *is computed as*

$$r_F = o - t. \tag{1.5}$$

Evidently, from eqs.(1.4) and (1.5), the kinematic redundancy of eq.(1.3) can be rewritten as

$$r_K = r_I + r_F, \tag{1.6}$$

which makes clear that kinematic redundancy comes from two different sources : the functional redundancy and the intrinsic redundancy. The distinction among the different redundancies is shown in Fig. 1.4. When $n = o$, we have non-redundant manipulators. When $n > o$, we intrinsically redundant manipulators. Finally, when

$n < o$, the manipulator does not exist because it can not fulfill the motion in the operational space. For example, let us consider a $PRRRR$ serial manipulator as shown in Fig. 1.5. For this manipulator, $\mathcal{J}$ is of dimension 5, *i.e.*, $\dim(\mathcal{J}) = n = 5$, the resulting $\mathcal{O}$ is only of dimension 4 (positioning a point of the EE in 3D space and orienting the EE around a vertical axis), *i.e.*, $\dim(\mathcal{O}) = o = 4$, and hence, this manipulator has a degree of intrinsic redundancy of one, *i.e.*, $r_I = n - o = 5 - 4 = 1$. For the positioning task in 3D space without considering its orientation , $\mathcal{T}$ is only of dimension 3, *i.e.*, $\dim(\mathcal{T}) = t = 3$, and hence, the degree of functional redundancy of the pair of manipulator-task is one, *i.e.*, $r_F = o - t = 4 - 3 = 1$. Finally, the kinematic redundancy of this pair of manipulator-task is two because $r_K = r_I + r_F = 1 + 1 = 2$, and not one. In the literature, most of the research works on RR scheme of serial manipulators study only the case of $r_F = 0$, and thus, $r_K = r_I$. In this thesis, we rather study the opposite case, *i.e.*, $r_F \neq 0$ and thus, $r_K = r_F + r_I$.

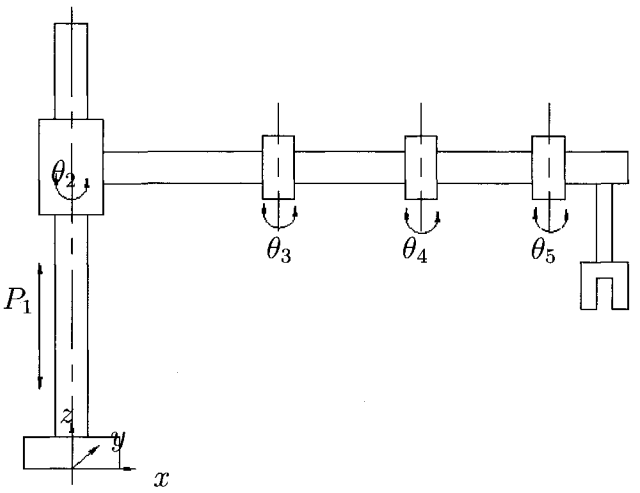


FIG. 1.5 Kinematic redundant manipulator : Robot 1

For example, the Canadarm2 used on the International space station is a seven-revolute serial manipulators that is intrinsically redundant. For intrinsically redun-

dant manipulators, the scheme of using the null space[2] of the Jacobian matrix can be directly used to select an optimized solution.

Six-revolute serial manipulators are the most well known and popular robots because they are multipurpose. These manipulators are also able to be functionally redundant when the tasks require less than the full six-DOF of the EE. As shown in Figs. 1.6 and 1.7, arc-welding, laser cutting and milling tasks require only 5-DOF ($t = 5$). It is because, for all these tasks, there exist an axis around which a rotation of the EE is irrelevant to the task. In these cases, the tasks and the manipulators are functionally-redundant. A similar situation occurs for pick and place operations of axis-symmetric objects ($t = 4$ or 5).

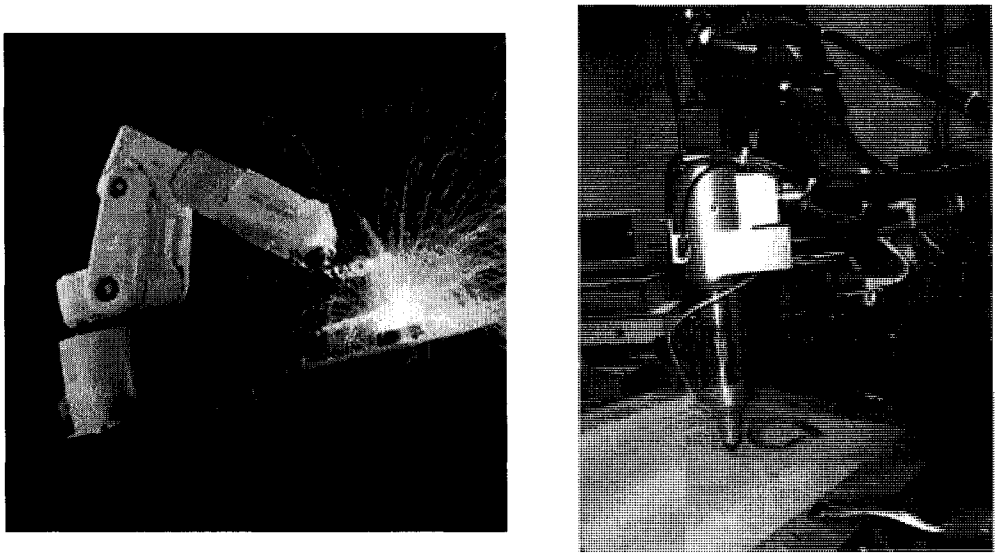


FIG. 1.6 Arc welding task (left) and laser cutting task (right)

Most of the RR schemes focus on the solution of intrinsically-redundant manipulators, and use the null-space of the Jacobian matrix. However, the RR schemes that use the null space of the Jacobian matrix can not directly be used to solve the functionally-redundant problem. In these cases, the Jacobian matrix is a full rank square matrix, and hence, the dimension of its null space is zero. Consequently, the well known RR

[2]Null space : If $\mathbf{T}$ is a linear transformation of $\mathbb{R}^n$, then the null space $\mathbf{Null(T)}$ is the set of all vectors $\mathbf{x}$ such that $\mathbf{T(x)} = \mathbf{0}$, *i.e.*, $\mathbf{Null(T)} \equiv \{\mathbf{x} : \mathbf{T(x)} = \mathbf{0}\}$.[8]

schemes that use the null space of the Jacobian matrix are not able to solve the functionally-redundant problem directly.

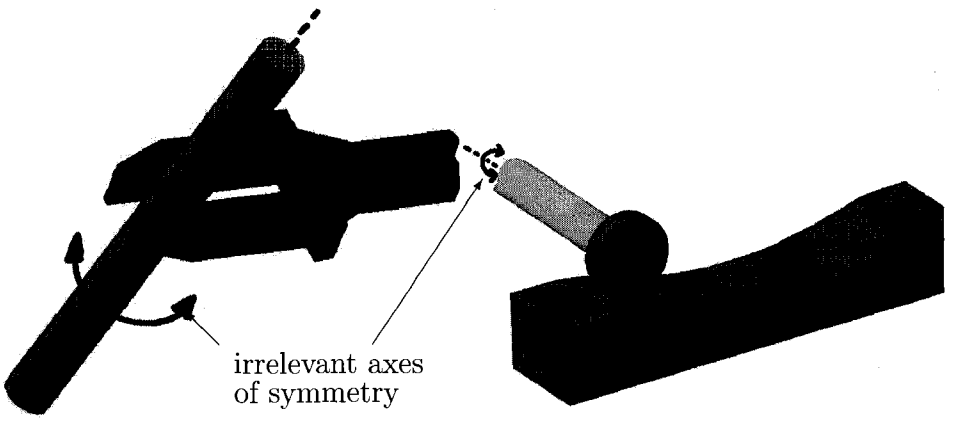


FIG. 1.7 Pick-up task (left) and milling task (right)

1.3 Research Objective

Many industrial tasks, such as arc-welding, milling, deburing, laser-cutting, and many others, require less than six-DOF, because of the presence of a symmetry axis of the EE. The EE rotation around the symmetry axis is irrelevant, and thus functional redundancy occur as these tasks are performed by the manipulators with DOF greater or equal to six, *i.e.*, $n \geq 6$. Hence, the main research objective of this thesis is to study how to take advantage of the functional redundancy to optimize the joint space solution for a given Cartesian path.

CHAPITRE 2

LITERATURE REVIEW

2.1 Level of Kinematic Analysis

The kinematic analysis of serial manipulators comprises the study for three aspects of such mechanical systems[1] :

1. the relations between *joint positions* and *Cartesian positions* of the EE, known as *displacement analysis*;
2. the relations between the time-rates of change of the joint positions, referred to as the *joint rates*, and the twist[1] of the EE, known as *velocity analysis*;
3. the relations between the second time-derivatives of the joint positions, referred to as the *joint accelerations*, with the time-rate of change of the twist of the EE, known as *acceleration analysis.*

2.1.1 Direct and Inverse Kinematic Problems

Figure 2.1 shows the mapping between joint space and operational space at the displacement level. The *direct kinematic problem* (DKP) is the mapping from joint space to operational space, *i.e.*, determining the pose of the EE for a given manipulator in a given posture. For serial manipulators, this problem is straightforward and admits a unique solution, which can be determined by simple matrix and vector multiplications. Alternatively, the *inverse kinematic problem* (IKP) is the inverse problem as expressed by its name, *i.e.*, it pertains to the mapping from operational space to joint space, *i.e.*, determining the posture of a given manipulator for a given pose of its EE.

[1]The twist refers to the combination of linear and angular velocity of the EE.

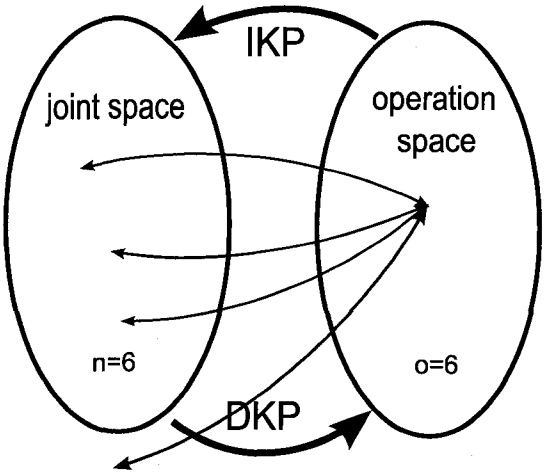


FIG. 2.1 Mapping between joint and operational spaces at the displacement level.

At the displacement level, a point in operational space may map onto a set of points in the joint space, while a point in joint space represents a unique pose of the EE. Accounting for the dependence of position and orientation of the EE with the joint positions, the DKP can be written as the following nonlinear algebraic system, *i.e.*,

$$\mathbf{x} = \mathbf{DKP}(\boldsymbol{\theta}), \tag{2.1}$$

where $\boldsymbol{\theta}$ is a point in $\mathcal{J}$ and $\mathbf{x}$ the corresponding point in $\mathcal{O}$. The function $\mathbf{DKP}(\cdot)$ allows the computation of the operational space variables $\mathbf{x}$ from the knowledge of the joint space variables $\boldsymbol{\theta}$ and the characteristics of the manipualtor.

Similarly, the IKP is also written as the following nonlinear algebraic system, *i.e.*,

$$\boldsymbol{\theta} = \mathbf{IKP}(\mathbf{x}). \tag{2.2}$$

In general, the IKP is much more complex and challenging than the DKP for the following reasons[9] :

– the equations to be solved are nonlinear, and thus, it is rarely possible to find a closed-form solution ;

- multiple solutions usually exist;
- an infinite number of solutions may exist, in the case of a kinematically redundant manipulator;
- there might be no admissible solutions, in view of the manipulator kinematic structure. One may ask for a pose of the EE outside of the reachable workspace of the manipulator.

At the displacement level, inverse kinematics requires the solution of a highly nonlinear algebraic system, for which no analytical closed-form solution exists for a general 6-R manipulator[10]. Pieper[11] showed that all 6-R manipulator with three succeeding revolute joint axes intersecting at a common point, termed as *decoupled manipulator*, always have closed-form solutions of eh IKP. Tsai and Morgan[12] found that although the number of real solutions changes from case to case, the total number of solutions (real and complex) is 16 for all the 6-R manipulators, except for the decoupled cases where there are only 8 solutions. The existence of joint limits and link obstruction may eventually reduce the number of reachable solutions.

2.2 Differential Kinematics and Redundancy

Differential kinematics of robot manipulators were introduced by Whitney[13] in 1969. He proposed to use differential relationships to solve the joint space motion from a given Cartesian space motion of the EE. The relationship between the EE velocity and the joint velocity can be represented by a linear algebraic equation. The coefficient of the linear equation is the *Jacobian matrix*, which is a nonlinear function of the joint angles and the manipulator's geometry. Whitney named this method the *resolved-motion rate control.*

Although the computation of closed-form nonlinear solution, if available, is generally less computationally expensive than that for resolved-motion rate control, the linearity of the equation at the velocity level allows the development of general dis-

cussion, whereas the computation of closed-form solution depends on the specific design of a robotic manipulator. Particularly, the differential kinematics are natural and unique in Cartesian trajectory control when the manipulator dynamics are considered because rigid-body dynamics and kinematics are naturally connected at the acceleration level. The relationship between the EE acceleration and the joint acceleration becomes a similar linear one, with the Jacobian matrix as a coefficient. This scheme is named *resolved acceleration control* as an extension of resolved motion rate control. It is noteworthy that differential kinematics are also related to EE force control[7].

2.2.1 Geometric and Analytical Jacobian Matrices

Suppose that the following relation holds between a k-dimensional vector $\boldsymbol{\xi} = [\xi_1, \xi_2, \ldots, \xi_k]^T$ and an l-dimensional vector $\boldsymbol{\eta} = [\eta_1, \eta_2, \ldots, \eta_l]^T$:

$$\eta_j = f_j(\xi_1, \xi_2, \ldots, \xi_k), \quad j = 1, 2, \ldots, l. \tag{2.3}$$

Then the $l \times k$ matrix

$$\mathbf{J}_\eta(\boldsymbol{\xi}) = \begin{bmatrix} \frac{\partial \eta_1}{\partial \xi_1} & \frac{\partial \eta_1}{\partial \xi_2} & \cdots & \frac{\partial \eta_1}{\partial \xi_k} \\ \frac{\partial \eta_2}{\partial \xi_1} & \frac{\partial \eta_2}{\partial \xi_2} & \cdots & \frac{\partial \eta_2}{\partial \xi_k} \\ \vdots & \vdots & & \vdots \\ \frac{\partial \eta_l}{\partial \xi_1} & \frac{\partial \eta_l}{\partial \xi_2} & \cdots & \frac{\partial \eta_l}{\partial \xi_k} \end{bmatrix} = \frac{\partial \boldsymbol{\eta}}{\partial \boldsymbol{\xi}} \tag{2.4}$$

is called the *Jacobian matrix*[14] of $\boldsymbol{\eta}$ with respect to $\boldsymbol{\xi}$. Furthermore, suppose that $\boldsymbol{\eta}$ and $\boldsymbol{\xi}$ are functions of time, then differentiating eq.(2.3) with respect to time and substituting eq.(2.4) yields

$$\dot{\boldsymbol{\eta}} = \mathbf{J}_\eta(\boldsymbol{\xi})\dot{\boldsymbol{\xi}}. \tag{2.5}$$

Using the Jacobian matrix, we can express the relation between the EE velocity and

the joint velocity of a manipulator in a compact form as follows :

$$\dot{\mathbf{x}} = \mathbf{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}. \tag{2.6}$$

If defining $\mathbf{t} \equiv \dot{\mathbf{x}}$, then eq.(2.6) is written as

$$\mathbf{t} = \mathbf{J}(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}. \tag{2.7}$$

The translational velocity of the EE, *i.e.*, $\dot{\mathbf{p}}$, can naturally be expressed by the time derivative of the position vector $\mathbf{p}$, however, there are many methods for expressing the rotation velocity of the EE as follows :

Method I : Select a vector ${}^{A}\boldsymbol{\phi}_{B}$ consisting of three variables for expressing the orientation of the EE (e.g., Euler angles, or roll, pitch and yaw angles), and use its time derivative ${}^{A}\dot{\boldsymbol{\phi}}_{B} = \frac{d^{A}\boldsymbol{\phi}_{B}}{dt}$ to express the rotation velocity of the EE[2].

Method II : The motion of $\mathcal{F}_B$ with respect to $\mathcal{F}_A$ at each instant of time is a rotation about an axis passing through the origin. This means that the rotation velocity of $\mathcal{F}_B$ can be described by the vector ${}^{A}\boldsymbol{\omega}_{B}$, which has the same direction as the instantaneous axis of rotation and a magnitude proportional to the rotational speed about this axis. The vector ${}^{A}\boldsymbol{\omega}_{B}$ is called the *angular velocity vector.*

From the viewpoint of the physical meaning of the vector expressing the velocity, ${}^{A}\boldsymbol{\omega}_{B}$ is superior to ${}^{A}\dot{\boldsymbol{\phi}}_{B}$. The three components of ${}^{A}\boldsymbol{\omega}_{B}$ represent the orthogonal angular velocity components about the X, Y, and Z axes of $\mathcal{F}_A$. In contrast, those of ${}^{A}\dot{\boldsymbol{\phi}}_{B}$ generally represent non-orthogonal components about the three axes of a skew coordinate system whose coordinate axes vary depending on the present value of ${}^{A}\boldsymbol{\phi}_{B}$. It has been shown that the mapping between ${}^{A}\dot{\boldsymbol{\phi}}_{B}$ and ${}^{A}\boldsymbol{\omega}_{B}$ introduces singularity into the linear algebraic system if ${}^{A}\dot{\boldsymbol{\phi}}_{B}$ is used, while no singularity is added if ${}^{A}\boldsymbol{\omega}_{B}$

[2]The superscript A and the subscript B of ${}^{A}\boldsymbol{\phi}_{B}$ indicate that $\boldsymbol{\phi}$ describes the orientation of $\mathcal{F}_B$ with respect to $\mathcal{F}_A$.

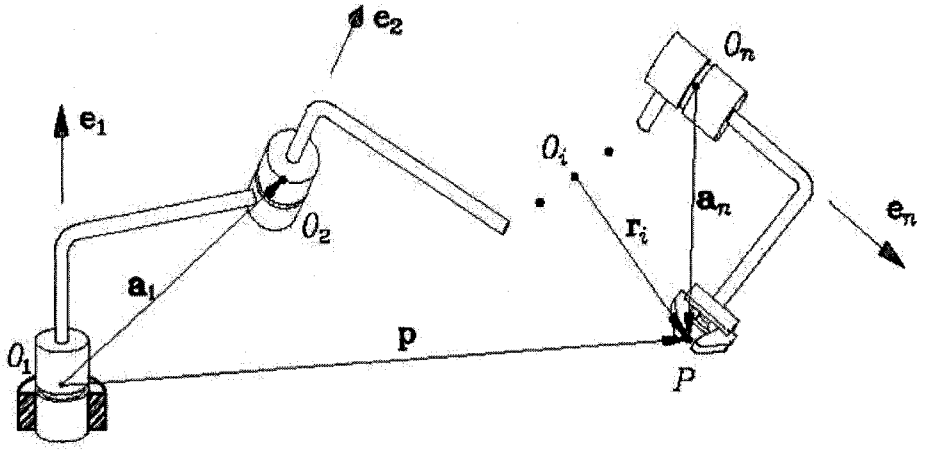


FIG. 2.2 General n axis manipulator[1]

is used.

Corresponding to the two methods of expressing the rotation velocity, there are two kinds of Jacobian matrices, termed as *geometric* and *analytical* Jacobian respectively. When method I is used for expressing the rotational velocity of the EE, the equation of reaching the analytic Jacobian $\mathbf{J}_a$ is as follows :

$$\mathbf{J}_a(\boldsymbol{\theta}) = \frac{\partial \boldsymbol{f}}{\partial \boldsymbol{\theta}} \in \mathbb{R}^{6\times n}. \tag{2.8}$$

Whitney[13]computed $\dot{\boldsymbol{\theta}}$ in this way with method I in 1969.

However, it would be computationally inefficient to try to evaluate the analytic Jacobian matrix. In 1972, Whitney[15] proposed another method of the Jacobian matrix, *i.e.*, *geometric Jacobian matrix* for method II.

From Fig. 2.2, the angular velocity vector of the EE, namely $\boldsymbol{\omega}$, is readily computed as

$$\boldsymbol{\omega} = \sum_{i=1}^{n} (\dot{\theta}_i \ \mathbf{e}_i), \tag{2.9}$$

where $\dot{\theta}_i$ and $\mathbf{e}_i$ are respectively defined as a joint rate and a unit vector associated with the direction of the revolute axis of joint i.

Moreover, the translational velocity of the origin of the frame $\mathcal{F}_n$ attached to the EE, namely $\dot{\mathbf{p}}$, is readily computed as

$$\dot{\mathbf{p}} = \sum_{i=1}^{n} (\dot{\theta}_i \ \mathbf{e}_i \times \ \mathbf{r}_i), \tag{2.10}$$

where vector $\mathbf{r}_i$ is defined as the position vector of the origin of $\mathcal{F}_n$ with respect to the origin of $\mathcal{F}_i$ and expressed in $\mathcal{F}_0$, *i.e.*,

$$\mathbf{r}_i \equiv \mathbf{a}_i + \mathbf{a}_{i+1} + ... + \mathbf{a}_n, \tag{2.11}$$

and $\mathbf{a}_i$ is defined as

$$\mathbf{a}_i \equiv \begin{bmatrix} l_i \cos(\theta_i) \\ l_i \sin(\theta_i) \\ r_i \end{bmatrix}. \tag{2.12}$$

Therefore, the velocity relationship involving the geometric Jacobian matrix as

$$\mathbf{t} = \mathbf{J}_g(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}, \tag{2.13}$$

where the geometric Jacobian matrix is defined as

$$\mathbf{J}_g = \begin{bmatrix} \mathbf{A} \\ \mathbf{B} \end{bmatrix}, \tag{2.14}$$

where

$$\mathbf{A} \equiv \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & ... & \mathbf{e}_n \end{bmatrix} \in \mathbb{R}^{3\times n}; \tag{2.15}$$

$$\mathbf{B} \equiv \begin{bmatrix} \mathbf{e}_1 \times \mathbf{r}_1 & \mathbf{e}_2 \times \mathbf{r}_2 & ... & \mathbf{e}_n \times \mathbf{r}_n \end{bmatrix} \in \mathbb{R}^{3\times n}. \tag{2.16}$$

and

$$\mathbf{t} \equiv \begin{bmatrix} \boldsymbol{\omega}^T & \dot{\mathbf{P}}^T \end{bmatrix}^T, \tag{2.17}$$

$$\dot{\boldsymbol{\theta}} \equiv \begin{bmatrix} \dot{\theta}_1 & \cdots & \dot{\theta}_n \end{bmatrix} \tag{2.18}$$

If the Euler angles are used as $\boldsymbol{\phi}$, *i.e.*, there is

$$\dot{\boldsymbol{\phi}} = \begin{bmatrix} \dot{\alpha} & \dot{\beta} & \dot{\gamma} \end{bmatrix}^T, \tag{2.19}$$

where α, β and γ are the rotation angles about the Z, Y and X axis, respectively.

$\boldsymbol{\omega}$ and $\dot{\boldsymbol{\phi}}$ are obtained from each other by the following mapping, *i.e.*,

$$\boldsymbol{\omega} = \mathbf{K}\dot{\boldsymbol{\phi}}, \tag{2.20}$$

where

$$\mathbf{K} \equiv \begin{bmatrix} 0 & -\sin\alpha & \cos\alpha\sin\beta \\ 0 & \cos\alpha & \sin\alpha\sin\beta \\ 1 & 0 & \cos\beta \end{bmatrix}. \tag{2.21}$$

Therefore, the analytical Jacobian $\mathbf{J}_a(\boldsymbol{\theta})$ and the geometric Jacobian $\mathbf{J}_g(\boldsymbol{\theta})$ are related to each other by

$$\mathbf{J}_g(\boldsymbol{\theta}) \equiv \begin{bmatrix} \mathbf{K} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} \end{bmatrix} \mathbf{J}_a(\boldsymbol{\theta}), \tag{2.22}$$

where $\mathbf{1} \in \mathbb{R}^{3\times 3}$ is an identity matrix. The coefficient matrix on the right-hand side of equation eq.(2.22) becomes singular when $\sin\beta = 0$. This means that although any rotational velocity can be described by ${}^A\boldsymbol{\omega}_B$, there are rotational velocities that can not be described by ${}^A\dot{\boldsymbol{\phi}}_B$ when $\sin\beta = 0$. Orientations of this kind are called *representation singularities* of ${}^A\dot{\boldsymbol{\phi}}_B$.

In general, it is anticipated that the geometric Jacobian will be adopted whenever it is necessary to refer to quantities of clear physical meaning, while the analytical Jacobian will be applied whenever it is necessary to refer to differential quantities of variables defined in the operational space.

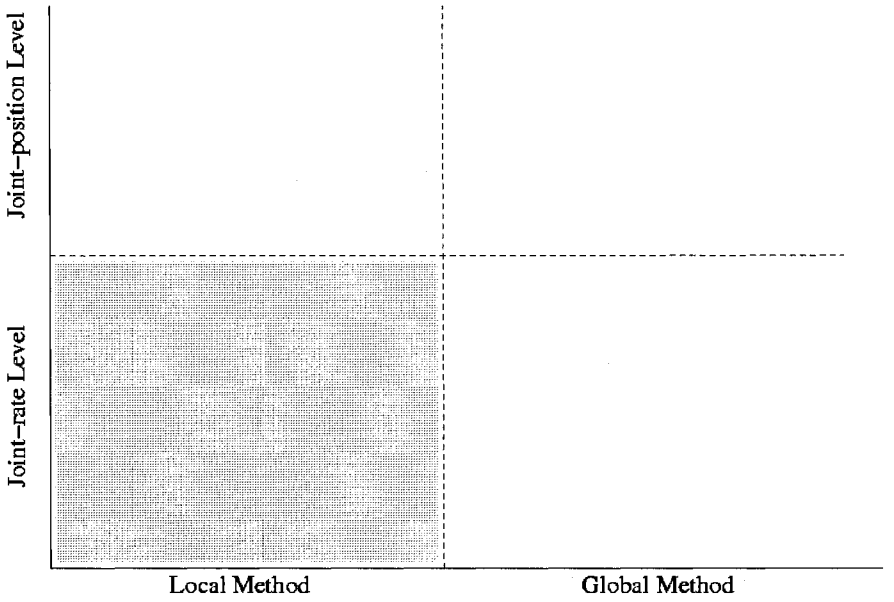


FIG. 2.3 Classification of RR schemes : the most popular ones are in the shaped region

2.2.2 Classification of Redundancy-Resolution Schemes

Redundant manipulators attracted the attention of many researchers in recent years, because of their extra DOF allowing more sophisticated motions than their non-redundant counterparts. Hence, the redundant manipulators have widely been used on avoidance tasks for obstacles, joint-limits and singularities. Many research works focused on the theoretical aspects of the RR schemes. For example, Baillieul[16][17] used redundancy for obstacle avoidance by using an extended Jacobian technique. Klein[18] solved redundancy for maximum dexterity of a manipulator. However, most of the reported works have been tested on simulations, while only a few implementations on real robots have been reported as Honegger and Codourey[19].

The RR schemes can be classified into local methods and global methods. The global optimal control methods need all the required data of the overall path before the movement is realized. The local optimal control methods give a solution for every instant and can also use the available sensory data. Thus, the locally optimal control

approach is suitable for real-time applications, such as sensor-based obstacle avoidance strategies. The globally optimal control approach is limited only to off-line trajectory planning for tasks requiring strict optimality, such as obstacle avoidance in a complicated but time invariant workspace and energy minimization[7]. Therefore, these two frameworks should be properly used, depending on the situation. Most of the RR schemes use local methods since their performance index depends on the instantaneous robot posture.

The RR problem has been solved at both joint-position level[20][21] and joint-rate level[17][22] using different methods. At the joint-position level, the RR problem is nonlinear, whereas it is linear at the joint-rate level. Figure 2.3 shows the classification of RR schemes discussed above. The shaded part is the most popular research domain, *i.e.*, RR scheme in local method at joint-rate level.

During the last two decades, more and more researchers have applied the artificial intelligent technique on the redundant manipulator control.

2.3 Local Optimization Algorithms

Nowadays, most of the RR researchers worked on the joint-rate level, and used the Moore-Penrose Generalized Inverse[23][3] (GI) or the Weighted Generalized Inverse

[3]Given an $m \times n$ matrix $\mathbf{B}$, the Moore-Penrose generalized matrix inverse is a unique $n \times m$ matrix pseudoinverse $\mathbf{B}^\dagger$. The Moore-Penrose inverse satisfies

$$\mathbf{B}\mathbf{B}^\dagger\mathbf{B} = \mathbf{B}$$

$$\mathbf{B}^\dagger\mathbf{B}\mathbf{B}^\dagger = \mathbf{B}^\dagger$$

$$(\mathbf{B}\mathbf{B}^\dagger)^T = \mathbf{B}\mathbf{B}^\dagger$$

$$(\mathbf{B}^\dagger\mathbf{B})^T = \mathbf{B}^\dagger\mathbf{B}.$$

It is also true that

$$\mathbf{z} = \mathbf{B}^\dagger\mathbf{c}$$

is the solution with shortest length to the problem

$$\mathbf{B}\mathbf{z} = \mathbf{c}.$$

(WGI) of the Jacobian matrix.

2.3.1 Schemes Using the Generalized Inverse

The Moore-Penrose pseudo-inverse can be used to find the joint-rate vector that has the smallest Euclidean norm, usually called the minimum-norm solution computed as

$$\mathbf{J}^{\dagger} = \mathbf{J}^{T}(\mathbf{J}\mathbf{J}^{T})^{-1}, \tag{2.23}$$

and hence eq.(2.7) becomes

$$\dot{\boldsymbol{\theta}} = \mathbf{J}^{\dagger}\mathbf{t}. \tag{2.24}$$

Klein and Huang (1983)[24] showed that such a solution may lead to noncyclic motions in joint space. At the cost of giving up the minimum-norm solution, an homogeneous component can be added to eq.(2.24) in order to optimize a secondary task into an additional criterion. Thus, this non-minimum-norm general inverse solution can be written as :

$$\dot{\boldsymbol{\theta}} = \underbrace{(\mathbf{J}^{\dagger})\mathbf{t}}_{\text{minimum-norm solution}} + \underbrace{(\mathbf{1} - \mathbf{J}^{\dagger}\mathbf{J})\mathbf{h}}_{\text{homogeneous solution}}, \tag{2.25}$$

where the first part of eq.(2.25) is the minimum-norm solution, known as the base solution, and the second part is an arbitrary vector in the null space of the Jacobian **J**. Equation (2.25) is used widely by many researchers such as Angeles *et al.* (1998)[25], and Siciliano (1992)[22] in order to solve redundant tasks. Vector **h** of eq.(2.25) comes from an optimized performance criterion. Different selection of **h** give different performances.

Liégeois (1977)[26] developed the *Gradient projection method* (GPM), which minimizes a position-dependent scalar performance index $p(\boldsymbol{\theta})$ and takes its gradient as vector

$\mathbf{h}$, *i.e.*,

$$\mathbf{h} = w\frac{\partial p}{\partial \boldsymbol{\theta}} = w\left[\begin{array}{cccc} \frac{\partial p}{\partial \theta_1} & \frac{\partial p}{\partial \theta_2} & \cdots & \frac{\partial p}{\partial \theta_n} \end{array}\right]^T, \tag{2.26}$$

where w is a positive scalar coefficient. Liégeois introduced a p that helps joint-limit avoidance, in the form

$$p = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{\theta_i - \theta_i^{mid}}{\theta_i^{mid} - \theta_i^{max}}\right)^2, \tag{2.27}$$

where $\theta_i^{mid} = (\theta_i^{min} + \theta_i^{max})/2$, θ_i^{min} and θ_i^{max} are lower and upper joint limits, respectively.

Other researchers developed different performance indeces related to various applications of RR schemes.

Yoshikawa (1984)[27] suggested a scaler value ω to be the *measure of manipulability* (MOM), namely,

$$\omega = \sqrt{det(\mathbf{J}\mathbf{J}^T)}, \tag{2.28}$$

and used ω as performance index p to avoid singularities. Yoshikawa also introduced a performance index for obstacle avoidance, namely,

$$p = \frac{(\boldsymbol{\theta} - \boldsymbol{\theta}_r)^T\mathbf{H}(\boldsymbol{\theta} - \boldsymbol{\theta}_r)}{2}, \tag{2.29}$$

where $\mathbf{H}$ is a diagonal matrix with constant entries greater than zero and $\boldsymbol{\theta}_r$ a given arm posture.

If using SVD for Jacobian, $\mathbf{J} = \boldsymbol{U}\Sigma V^T$, eq.(2.28) becomes

$$\omega = \sqrt{\mathrm{d}et(\boldsymbol{\Sigma}\Sigma^T)} = \sigma_1\sigma_2\cdots\sigma_m. \tag{2.30}$$

Thus, MOM is nothing but the product of the singular values of $\mathbf{J}$. When the Jacobian matrix degenerates, one or more singular values becomes zero, and so does MOM.

The condition number of the Jacobian matrix is

$$\kappa = \frac{\sigma_1}{\sigma_m}, \tag{2.31}$$

which is firstly used by Salisbury and Craig[28].

As described by Nakamura[7], the relationship between MOM and the condition number is clear when they are physically interpreted using the manipulability ellipsoid. MOM is proportional to the volume of the ellipsoid, whereas the condition number is the ratio of the lengths of the longest and shortest principal axes. In other words, MOM is related to the magnitude of the ellipsoid, whereas the condition number concerns the shape of the ellipsoid. Therefore, MOM prefers large ellipsoids, and the condition number prefers ellipsoids with spherical shape.

Yoshikawa's manipulability measure has been used extensively. However, the determinant cannot be a measure of how close a matrix is to singularity, as pointed out by Golub and Van Loan (1989)[29]. Therefore, Kosuge and Furuta (1985)[30] suggested to take the reciprocal (in order to have a number between 0 and 1) of the condition number of the Jacobian matrix as a *controllability measure.*

2.3.1.1 Inverse Kinematic Solutions Considering the Order of Priority

Nakamura and Hanafusa (1985)[33] found that there exist a lot of tasks which are composed of subtasks with different levels of significance, and call them tasks with the order of priority. For tasks with the order of priority, if it is impossible to perform all of the subtasks completely because of the degeneracy or the shortage of the DOF, they proposed to perform the most significant subtask preferentially and the less important subtask using the remaining DOF. In the case of a task with two subtasks. The subtask with the first priority is specified using the first manipulation variable, $\mathbf{r}_1 \in \mathbf{R}^{m_1}$, and the subtask with the second priority is specified using the second

manipulation variable, $\mathbf{r}_1 = 2 \in \mathbf{R}^{m_2}$. The kinematic relationships between the joint variables $\boldsymbol{\theta} \in \mathbf{R}^n$ and the manipulation variables are expressed as follows :

$$\mathbf{r}_i = \mathbf{f}_i(\boldsymbol{\theta}), \tag{2.32}$$

with $i = 1, 2$. Their differential relationships are expressed as follows :

$$\dot{\mathbf{r}}_i = \mathbf{J}_i(\boldsymbol{\theta})\dot{\boldsymbol{\theta}}, \tag{2.33}$$

where $\mathbf{J}_i(\boldsymbol{\theta}) \in \mathbf{R}^{m_i \times n}$ is the Jacobian matrix for the ith manipulation variable.

The general solution of eq.(2.33) for $i = 1$ is obtained using pseudoinverses as follows :

$$\dot{\boldsymbol{\theta}} = \mathbf{J}_1^{\dagger}(\boldsymbol{\theta})\dot{\mathbf{r}}_1 + (\mathbf{I} - \mathbf{J}_1^{\dagger}(\boldsymbol{\theta})\mathbf{J}_1(\boldsymbol{\theta}))\mathbf{y}. \tag{2.34}$$

Substituting eq.(2.34) into eq.(2.33) for $i = 2$, we get the following equation :

$$\mathbf{J}_2(\mathbf{I} - \mathbf{J}_1^{\dagger}(\boldsymbol{\theta})\mathbf{J}_1(\boldsymbol{\theta}))\mathbf{y} = \dot{\mathbf{r}}_2 - \mathbf{J}_2\mathbf{J}_1^{\dagger}\dot{\mathbf{r}}_1, \tag{2.35}$$

where $\mathbf{y} \in \mathbf{R}^n$ is an arbitrary vector. If the exact solution of $\mathbf{y}$ exists for eq.(2.35), it means that the second manipulation variable can be realized. However, the exact solutions does not generally exist. We get $\mathbf{y}$, which minimizes $\|\dot{\mathbf{r}}_2 - \mathbf{J}_2\dot{\boldsymbol{\theta}}\|$, in the same way as eq.(2.34), *i.e.*,

$$\mathbf{y} = \tilde{\mathbf{J}}_2^{\dagger}(\dot{\mathbf{r}}_2 - \mathbf{J}_2\mathbf{J}_1^{\dagger}\dot{\mathbf{r}}_1) + (\mathbf{I} - \tilde{\mathbf{J}}_2^{\dagger}\tilde{\mathbf{J}}_2)\mathbf{z}, \tag{2.36}$$

where $\tilde{\mathbf{J}}_2 = \mathbf{J}_2(\mathbf{I} - \mathbf{J}_1^{\dagger}\mathbf{J}_1)$, and $\mathbf{z} \in \mathbf{R}^n$ is an arbitrary vector.

The solution $\dot{\boldsymbol{\theta}}$ is obtained from eqs.(2.34) and (2.36) as follows :

$$\dot{\boldsymbol{\theta}} = \mathbf{J}_1^{\dagger}\dot{\mathbf{r}}_1 + (\mathbf{I} - \mathbf{J}_1^{\dagger}\mathbf{J}_1)\tilde{\mathbf{J}}_2^{\dagger}(\dot{\mathbf{r}}_2 - \mathbf{J}_2\mathbf{J}_1^{\dagger}\dot{\mathbf{r}}_1) + (\mathbf{I} - \mathbf{J}_1^{\dagger}\mathbf{J}_1)(\mathbf{I} - \tilde{\mathbf{J}}_2^{\dagger}\tilde{\mathbf{J}}_2)\mathbf{z}. \tag{2.37}$$

Equation (2.37) represents the inverse kinematic solution considering task priority.

2.3.1.2 Schemes Using the Weighted Generalized Inverse

The *Weighted Generalized Inverse* (WGI) solution to eq.(2.7) is

$$\dot{\boldsymbol{\theta}} = \mathbf{J}_w^{\dagger}\mathbf{t}, \tag{2.38}$$

where

$$\mathbf{J}_w^{\dagger} = \mathbf{W}^{-1}\mathbf{J}^T(\mathbf{J}\mathbf{W}^{-1}\mathbf{J}^T)^{-1}, \tag{2.39}$$

and $\mathbf{W}$ is a positive-definite weighting matrix. Whitney (1969)[13] applied priorities through the weighted matrix. Park, Chung and Youm (1996)[31] used the WGI and introduced the weight not only in the generalized inverse, but also in the Jacobian matrix and in the joint velocity, which they called *weighted Jacobian* and *weighted joint velocity*, respectively. Hence, they solved the equation

$$\mathbf{J}_w\dot{\boldsymbol{\theta}}_w = \mathbf{t}, \tag{2.40}$$

where $\mathbf{J}_w$ and $\dot{\boldsymbol{\theta}}_w$ are the weighted Jacobian and weighted joint velocity, respectively.

Chan and Dubey (1995)[32] compared WGI with GPM in the case of joint limits avoidance problem, and found WGI reached a joint trajectory requiring much less joint velocities than GPM.

2.3.2 Scheme Using Householder Reflection

Arenson, Angeles and Slutski (1998)[25] proposed to use *Householder reflection* in RR scheme. For the sake of brevity, we name it as AA householder reflection algorithm.

The algorithm is developed as follows. At first, equation (2.25) can be rewritten as :

$$\dot{\boldsymbol{\theta}} = \mathbf{k} + \mathbf{h}, \tag{2.41}$$

where $\mathbf{k}$ is shown as :

$$\mathbf{k} = \mathbf{J}^{\dagger}(\mathbf{t} - \mathbf{J}\mathbf{h}). \tag{2.42}$$

Equation (2.42) can be rewritten as :

$$\mathbf{J}\mathbf{k} = \mathbf{t} - \mathbf{J}\mathbf{h}. \tag{2.43}$$

In order to solve eq.(2.43), Householder reflections are used for the Jacobian matrix $\mathbf{J}$. When matrices $\mathbf{H}$ and $\mathbf{U}$ are found, they have a relation with $\mathbf{J}$ in the form

$$\mathbf{H}\mathbf{J}^{T} = \begin{bmatrix} \mathbf{U} \\ \mathbf{0} \end{bmatrix}, \tag{2.44}$$

where $\mathbf{U}$ is a $(o \times o)$ upper-triangular matrix, $\mathbf{H}$ is an orthogonal matrix $(n \times n)$, and $n > o$ for intrinsic redundancy. Hence, there is

$$\mathbf{H}^{T}\mathbf{H} = \mathbf{1}, \tag{2.45}$$

Rewriting eq.(2.43) to,

$$\mathbf{J}\mathbf{H}^{T}\mathbf{H}\mathbf{k} = \mathbf{t} - \mathbf{J}\mathbf{h}. \tag{2.46}$$

Because of :

$$\mathbf{J}\mathbf{H}^{T} = (\mathbf{H}\mathbf{J}^{T})^{T} = \begin{bmatrix} \mathbf{U}^{T} & \mathbf{0}^{T} \end{bmatrix}, \tag{2.47}$$

equation (2.46) is equal to

$$\begin{bmatrix} \mathbf{U}^{T} & \mathbf{0}^{T} \end{bmatrix} \mathbf{H}\ \mathbf{k} = \mathbf{t} - \mathbf{J}\mathbf{h}. \tag{2.48}$$

Then, $\mathbf{k}$ can be reached as follows :

$$\mathbf{k} = \mathbf{H}^T \begin{bmatrix} (\mathbf{U}^T)^{-1}(\mathbf{t} - \mathbf{Jh}) \\ \mathbf{0} \end{bmatrix}. \tag{2.49}$$

Substituting eq.(2.49) into eq.(2.41) and yields

$$\dot{\boldsymbol{\theta}} = \mathbf{H}^T \begin{bmatrix} (\mathbf{U}^T)^{-1}(\mathbf{t} - \mathbf{Jh}) \\ \mathbf{0} \end{bmatrix} + \mathbf{h}. \tag{2.50}$$

If we define :

$$\mathbf{y} \equiv \begin{bmatrix} (\mathbf{U}^T)^{-1}(\mathbf{t} - \mathbf{Jh}) \\ \mathbf{0} \end{bmatrix}, \tag{2.51}$$

Equation (2.50) can be rewritten as :

$$\dot{\boldsymbol{\theta}} = \mathbf{H}^T\mathbf{y} + \mathbf{h}. \tag{2.52}$$

The AA algorithm avoids the direct calculation of the generalized inverse of the Jacobian matrix, as with eq.(2.23) and even worst with eq.(2.25). Hence, the squaring of the condition number of $\mathbf{J}$ is avoided and the round-off error of the algorithm is not amplified. It is an interesting method for the calculation of joint-motion at ill-condition postures, because here the Jacobian matrix $\mathbf{J}$ may have a very high condition number.

2.4 Global Optimization Algorithms

Although the local optimal control approach requires a small amount of computation, it lacks a guarantee of global optimality. The global optimal control approach is better for off-line trajectory planning of tasks requiring strict optimality, such as obstacle avoidance in complicated working spaces and energy minimization.

There are less papers published on the topic of global optimal control than on local optimal control. Whitney[15] first suggested the integrated value of kinetic energy, which is approximated to square joint velocities, be as a global criterion for redundancy utilization

The global optimal control methods are based either on the variational calculus with constraints or on Pontryagin's maximum principle.

2.4.1 Schemes Using the Pontrygain's Maximum Principle

Nakamura and Hanafusa[33][34] have solved the global optimal control problem with Pontryagin's maximum principle.

The optimal control problem of redundancy is represented as follows :

$$\dot{\boldsymbol{\theta}} = \mathbf{J}^{\dagger}\dot{\mathbf{r}}_1(t) + (\mathbf{I} - \mathbf{J}^{\dagger}\mathbf{J})\mathbf{y} = \mathbf{g}(\boldsymbol{\theta}, t, \mathbf{y}). \tag{2.53}$$

$$r_2 = \int_{t_0}^{t_1} p(\boldsymbol{\theta}, t)dt. \tag{2.54}$$

where $\mathbf{r}_1 \in \mathbb{R}^{m_1}$ represents the constrained variable such as the position and the orientation of the end effector; $r_2 \in \mathbb{R}^{m_2}$, $m_2 = 1$, represents the performance index of the integral type, which evaluates the performance of redundancy utilization. Equation (2.53) can be regarded as a system equation of a time-variant, non-linear, dynamical system by considering $\boldsymbol{\theta}$ as a state vector and $\mathbf{y}$ as an input vector, thus eqs.(2.53) and (2.54) can be regarded as an ordinary optimal control with a variable first endpoint and a free last endpoint, then Pontryagin's maximum principle can be applied to the problem.

The Hamiltonian function for a fixed-time problem with a fixed first endpoint and a

free last endpoint is as follows :

$$\mathrm{H}(\boldsymbol{\psi}, \boldsymbol{\theta}, t, \mathbf{y}) = -p + \boldsymbol{\psi}^T \mathbf{g}, \tag{2.55}$$

where $\boldsymbol{\psi} \in \mathbb{R}^n$ is an adjoint vector and $\mathbf{y}(t)$ is the independent control variation and can be arbitrary. If a $\mathbf{y}^*(t)$ maximizing the Hamiltonian of eq.(2.55) at every moment t is chosen, there is

$$\left(\frac{\partial H}{\partial \mathbf{y}}\right)_* = 0. \tag{2.56}$$

The optimal joint trajectory $\boldsymbol{\theta}^*(t)$ is yielded by solving the following differential equations :

$$\dot{\boldsymbol{\theta}}^* = \left(\frac{\partial H}{\partial \boldsymbol{\psi}}\right)_*. \tag{2.57}$$

$$\dot{\boldsymbol{\psi}}^* = -\left(\frac{\partial H}{\partial \boldsymbol{\theta}}\right)_*. \tag{2.58}$$

where eq.(2.57) is equivalent to eq.(2.53).

One example of the representative performance index in the work of Nakamura and Hanafusa is given as :

$$r_2 = \int_{t_0}^{t_1} (k p_0(\boldsymbol{\theta}) + \dot{\boldsymbol{\theta}}^T \dot{\boldsymbol{\theta}}) dt \tag{2.59}$$

where k is a nonnegative scalar. If $k = 0$, there is the simplest case of eq.(2.59) :

$$r_2 = \int_{t_0}^{t_1} \dot{\boldsymbol{\theta}}^T \dot{\boldsymbol{\theta}} dt. \tag{2.60}$$

The integrand of the criterion is considered as a pseudo-kinetic energy, and eq.(2.60) is used as the performance index to minimizing the integral of pseudo-kinetic energy.

In order to plan the joint trajectory farthest from singular points, the measure of manipulability ω proposed by Yoshikawa[27] is taken into account, and ω is calculated as eq.(2.28). Then, the corresponding performance index to plan the joint trajectory

farthest from singular points is given as follows :

$$r_2 = \int_{t_0}^{t_1} \left(\frac{1}{\sqrt{det(\mathbf{J}\mathbf{J}^T)}} + \dot{\boldsymbol{\theta}}^T \dot{\boldsymbol{\theta}} \right) dt \tag{2.61}$$

If the trajectory approaches the singularity, the first term of the integrant of eq.(2.61) increases infinitely.

2.4.2 Schemes Using the Calculus of Variations

Hollerbach and Suh (1987)[35] offered a solution for the global torque optimization based on the calculus of variations. Kazerounian and Wang (1988)[36][37] also used the calculus of variations to develop global solutions for the least square joint rates and the least kinetic energy.

In the method of Kazerounian and Wang, a global optimization problem is firstly set up to minimize

$$\mathbf{I} = \int_{t_o}^{t_f} (\dot{\mathbf{q}}^T \dot{\mathbf{q}}) dt \tag{2.62}$$

subject to

$$\mathbf{G}_k(\mathbf{q}, t) = 0, \quad k = 1 \; to \; m \tag{2.63}$$

The augmented objective function $\mathbf{I}^*$ is defined as

$$\mathbf{I}^* = \int_{t_0}^{t_f} g(\mathbf{q}, \dot{\mathbf{q}}, t) dt \tag{2.64}$$

where

$$g = \dot{\mathbf{q}}^T \dot{\mathbf{q}} + \sum \mu(t) \mathbf{G}_k. \tag{2.65}$$

Then by using the Euler-Lagrange equation, the necessary condition for $\mathbf{q}$ to be an

optimal of the functional $\mathbf{I}^*$ given by eq.(2.64) is

$$\frac{\partial g}{\partial \mathbf{q}} - \frac{d(\frac{\partial g}{\partial \dot{\mathbf{q}}})}{dt} = \mathbf{0}. \tag{2.66}$$

Because

$$\frac{\partial g}{\partial \mathbf{q}} = \left(\frac{\partial \mathbf{G}}{\partial \mathbf{q}}\right)^T \boldsymbol{\mu}(t) = \mathbf{J}^T \boldsymbol{\mu}, \tag{2.67}$$

$$\frac{d\left(\frac{\partial g}{\partial \dot{\mathbf{q}}}\right)}{dt} = 2\ddot{\mathbf{q}}, \tag{2.68}$$

eq.(2.66) can be rearranged as

$$\mathbf{J}^T \boldsymbol{\mu} - 2\ddot{\mathbf{q}} = \mathbf{0}, \tag{2.69}$$

i.e., there is

$$\ddot{\mathbf{q}} = 0.5\mathbf{J}^T \boldsymbol{\mu}. \tag{2.70}$$

There is the acceleration relation between the EE and joints as

$$\mathbf{J}\ddot{\mathbf{q}} = \ddot{\mathbf{x}} - \dot{\mathbf{J}}\dot{\mathbf{q}}. \tag{2.71}$$

By substituting eq.(2.70) into eq.(2.71), $\boldsymbol{\mu}$ is reached as follows :

$$\boldsymbol{\mu} = 2(\mathbf{J}\mathbf{J}^T)^{-1}(\ddot{\mathbf{x}} - \dot{\mathbf{J}}\dot{\mathbf{q}}). \tag{2.72}$$

By substituting eq.(2.72) into eq.(2.70), the acceleration vector $\ddot{\mathbf{q}}$ is reached as follows :

$$\ddot{\mathbf{q}} = \mathbf{J}^T(\mathbf{J}\mathbf{J}^T)^{-1}(\ddot{\mathbf{x}} - \dot{\mathbf{J}}\dot{\mathbf{q}}). \tag{2.73}$$

Hence, eq.(2.73) is the local minimization of the joint acceleration and will result in the global minimization of joint velocities.

Although the global optimization provides a more meaningful and stable solution

than local optimization, the disadvantages of global optimization is the amount of computations involved in the global optimization process often makes the on-line scheme prohibitive.

2.5 Redundancy-Resolution in Intelligent Control

Intelligent control is a new research direction making control systems to be more intelligent and higher degree of autonomy. With proper development, intelligent control systems may have great potential for solving complex control problems. Owing to this motivation, a number of realistic intelligent control approaches have been proposed and justified for their feasible applications to robotic systems and other complex systems. The most common intelligent control methods applied to RR schemes include *neural networks* (NN), *fuzzy logics* (FL) and *genetic algorithms* (GAs). These techniques eliminate some or all of the modelling of the manipulator kinematics and/or dynamics that is usually needed to implement conventional control techniques. This characteristic of intelligent control algorithms is significant considering the complexity of the mathematical models of the robotic system.

2.5.1 Fuzzy-Based Redundancy-Resolution Approach

Recently, some researchers[38][39][40][41] solved the inverse kinematics problem using FL. However, most of them did not propose a systematic method for generating and adjusting membership functions of fuzzy sets.

Graca (1993)[42] proposed an intelligent control algorithm, namely, *Fuzzy Learning Control*, for non-redundant robotic manipulators to track specified trajectories in Cartesian space. This algorithm consisted of treating the robotic kinematic equation as a linear possibility system with fuzzy coefficients. This linear possibility system was then solved for the fuzzy coefficients using Fuzzy Regression[43] to obtain a fuzzy

version of the Jacobian inverse matrix, as opposed to symbolically determining the Jacobian. Graca[44] also extended this algorithm to redundant manipulator for optimizing a secondary task.

Kim and Lee[38][39] proposed *fuzzy resolved-motion rate algorithm* (FRMRA) for a 3-DOF planar redundant manipulator, in order to replace the pseudo-inverse of the Jacobian matrix by the fuzzy reasoning. They designed a new *motion-rate resolving algorithm* (MRRA) based on the *gradient method*, then converted MRRA to FRMRA by fuzzifying the differential relationship between $d\mathbf{x}$ and $d\boldsymbol{\theta}$ for reaching a more accurate solution.

2.5.2 Neural Networks-Based Redundancy-Resolution Approach

In recent years, new interests in *Neural Network* (NN) research have been generated due to the low complexity in implementation and the fast computational time of motion planning and control for manipulators ([45] — [53]). Various NN have been applied for kinematic/dynamic control and path planning of manipulators. Many NN for robot kinematic control are feed-forward networks or variants of it.

Recurrent NN [48][51] have also been applied for kinematic control. Unlike feedforward NN, most recurrent NN do not need offline supervised learning and thus are more suitable for real-time robot control in uncertain environments. In order to effectively handle the physical limit constraints, Wang *et al.*[52][53] proposed a dual NN for kinematic resolution problems. The simulation results of this network are discussed on control the 7-DOF PA10 robot arm and show that its applications are successful. However, the disadvantage of the dual network is that an inverse matrix operation is required. Moreover, as pointed out by Wang[52], the exponential convergence of the dual network can not be guaranteed when it is not full rank or physical limit constraints are considered.

2.5.3 Genetic Algorithms-Based Redundancy-Resolution Approach

Genetic Algorithms (GAs) are a particular class of evolutionary algorithms that use techniques inspired by evolutionary biology such as inheritance, mutation, selection, and crossover (also called recombination). The use of GAs for solving the inverse kinematics of redundant robots was introduced by Parker *et al.*[54]. The GAs were used to position the end-effector of a robot at a target location with minimum position error, while minimizing the maximum rotational displacement from the initial position. Davidor[55] proposed a technique to apply GAs to the problem of robot trajectory generation in environment free of obstacles. Nearchou[56] [57] used GAs to solve the inverse kinematics problem in environments with obstacles, and also presented five reasons that makes GAs well suited for use in RR as the following :

- GAs often find nearly global optima in complex spaces ;
- GAs do not require any form of smoothness ;
- GAs do not need the computation of the Jacobian matrix, and need only the forward kinematics equations ;
- GAs allow additional constraints to be easily specified ;
- GAs work with the joint variables represented as digital values that is more suitable for computer controlled robot systems.

In the GAs technique, the final position accuracy is not satisfactory, since it generates a large final positioning error. However, GAs combined with other heuristic methods seems to be able to reach the desired solutions of the inverse kinematics problem.

2.5.4 Pros and Cons of Intelligent Controls

There are many researches that want to replace the physical kinematic and dynamic modelling of related manipulators by a learning process of an intelligent control algorithm which then be used to control the manipulator. On the other hand, there are many researchers who believe that this approach is not a good use of intelligent

control algorithms and should keep as much physical modelling as possible, and let the intelligent control algorithm to handle the uncertainties and the unknown physical phenomenon of the mechanical system at hand.

2.6 Functional Redundancy-Resolution

Redundancy is a concept related to the definition of the task instead of the intrinsic feature of the robot's structure. Although this fact is still not well understood in practice, it has been recognized by several researchers. Samson *et al.*[58] clearly stated that redundancy depends on the task and may change with time. Sciavicco and Siciliano[9] said the manipulator can be functionally redundant when only a number of components of dimensional operational space are of concern for the specific task, even if the dimension of operational space and joint space are equal.

Most of the researchers use the generalize inverse together with the projection onto the null space of $\mathbf{J}$ to solve the IKP of redundant manipulators. However, in the case of functional redundancy, $\mathbf{J}$ is often a full rank square matrix, *i.e.*, its null space doesn't exist from a mathematical point of view.

Although Sciavicco and Siciliano proposed the concept of functional redundancy, they didn't developed the corresponding solution. Thus, a new algorithm corresponding to the cases of full rank $\mathbf{J}$ becomes necessary.

In order to change eq.(2.7) to an under-determined system, there are two possibilities. One is augmenting the dimension of joint-rate $\dot{\boldsymbol{\theta}}$, another one is reducing the dimension of twist $\mathbf{t}$. Corresponding to the two possibilities, the functional RR schemes can be classified into two groups, *i.e.*, *augmented approach* and *reduced approach*, as shown in Fig. 2.4. *Virtual joint method* pertains to the augmented approach, while *elimination method* and TWA pertains to reduced approach.

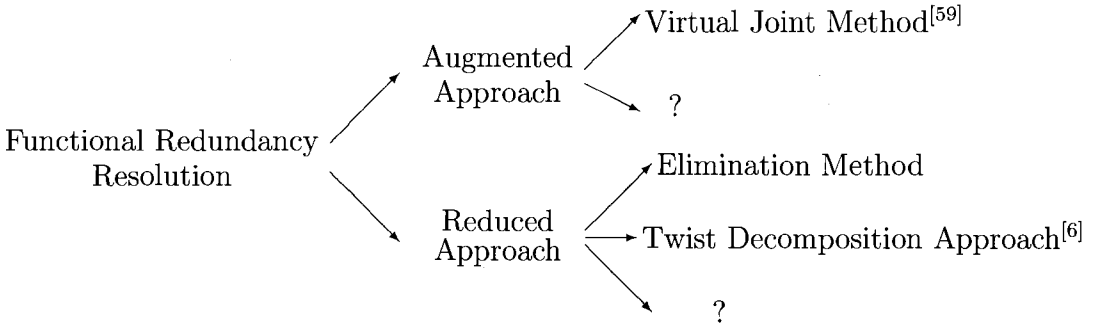


FIG. 2.4 Classification of functional redundancy-resolution schemes

The TWA is one of the main original contributions of this thesis, and was first proposed to solve the functional redundancy by Huo and Baron[6] in 2005. Instead of using the projection onto the null space of the Jacobian matrix, TWA is based on the orthogonal decomposition of the instantaneous twist into two orthogonal subspaces. TWA has great differences with the task-decomposition approach of Nakamura *et al.*[60]. Both of them consider the order of task priority, but the TWA projects the task from the robot base frame to the EE frame, and the motion of the secondary task is always constant in the EE frame, *e.g.*, the rotation around the symmetry axis of EE, while this secondary motion may or may not be constant in the base frame. Therefore, TWA classifies the order of task priority in the instantaneous EE frame instead of the robot base frame. Moreover, TWA is directly developed from the minimum-norm solution without considering the projection onto the null space of $\mathbf{J}$, while the development of task-decomposition approach is based on the general equation using the null space of $\mathbf{J}$. The TWA is fully developed in Chapter 3. Virtual joint method and elimination method are reviewed as the follow.

2.6.1 Elimination Method

In some specific cases, the redundant velocity in operational space can be identified and directly eliminated, and hence the functionally-redundant task is transformed to an intrinsically-redundant task. This method is called elimination method.

For example, Baron[61] proposed to use an elimination method to optimize the surfacing operation implemented by 5-axes CNC milling machine with a half-ball EE. In the case of end-point surface milling, the orientation of EE is irrelevant to the task, and the angular velocities can be eliminated from twist $\mathbf{t}$. Thus, eq.(2.7) can be rewritten as

$$\dot{\mathbf{p}} = \mathbf{B}\dot{\boldsymbol{\theta}}, \tag{2.74}$$

where $\mathbf{B}$ is lower part of $\mathbf{J}$ as defined in eq.(2.16), *i.e.*, a 3×5 matrix, while $\dot{\mathbf{p}}$ and $\dot{\boldsymbol{\theta}}$ are 3 and 5−dimensional vectors, respectively. Hence, the well known non-minimum-norm solutions to eq.(2.74) can be written as

$$\dot{\boldsymbol{\theta}} = (\mathbf{B}^{\dagger})\dot{\mathbf{p}} + (\mathbf{1} - \mathbf{B}^{\dagger}\mathbf{B})\mathbf{h}, \tag{2.75}$$

with $\mathbf{h}$ being a secondary task and $\mathbf{B}^{\dagger}$ the right generalized inverse of $\mathbf{B}$, *i.e.*, $\mathbf{B}^{\dagger} = \mathbf{B}^{T}(\mathbf{B}\mathbf{B}^{T})^{-1}$. Obviously the orientation of the milling tool is not totally irrelevant, and hence, the preferred orientation is specified by $\mathbf{h}$ as a secondary task onto which we accept some deviation to accommodate other criteria.

However, the redundant velocity of the EE in operational space is not always constant in the base frame, and hence, it is sometime impossible to identify the redundant velocity in the base frame. Thus, the elimination method can not always be applied to solve functional redundancy.

2.6.2 Virtual Joint Method

The functional redundant space is always constant in the EE frame, while may or may not be constant in the base frame. When the functional redundant space is constant in the base frame, elimination method using the null-space can be used. However, when it is not constant, we must use non-traditional technics as the virtual joint method or the TWA.

Virtual joint method changes a functionally-redundant robotic task into an intrinsically-redundant manipulator by adding a virtual joint. Although the virtual joint does not exist, it is added into the joint vector in order to obtain an under-determined linear algebraic system with at least one DOF of redundancy.

In 2000, Baron[59] proposed to add a virtual joint to the manipulator so that a column is added to $\mathbf{J}$, rendering it underdetermined. In the case of arc-welding, a virtual joint around the symmetric axis of the electrode is added. The Jacobian augmented by the virtual joint-rate $\dot{\theta}_{n+1}$, namely $\mathbf{J}_v$, maps the augmented joints-rate $\dot{\boldsymbol{\theta}}_v$ into the twist $\mathbf{t}$ of the EE as

$$\mathbf{t} = \mathbf{J}_v \dot{\boldsymbol{\theta}}_v, \tag{2.76}$$

where $\dot{\boldsymbol{\theta}}_v$ is defined as

$$\dot{\boldsymbol{\theta}}_v \equiv \begin{bmatrix} \dot{\boldsymbol{\theta}} \\ \dot{\theta}_{n+1} \end{bmatrix}. \tag{2.77}$$

The control of this virtual $(n+1)$-DOF robot along a fully constrained n-DOF task with $\mathbf{t} = \mathbf{J}_v \dot{\boldsymbol{\theta}}_v$ is equivalent to the control of the previous n-DOF manipulator along the $(n-1)$-DOF task with $\mathbf{t} = \mathbf{J}\dot{\boldsymbol{\theta}}$. In both cases, there is one DOF of redundancy.

Hence, based on eq.(2.25), the non-minimum-norm solutions to eq.(2.76) can be written as

$$\dot{\boldsymbol{\theta}}_v = \mathbf{J}_v^{\dagger}\mathbf{t} + (\ \mathbf{1} - \ \mathbf{J}_v^{\dagger}\mathbf{J}_v)\mathbf{h}, \tag{2.78}$$

where $\mathbf{1}$ denotes the $(n+1) \times (n+1)$ identity matrix, $\mathbf{h}$ the secondary task and $\mathbf{J}_v^{\dagger}$ the right generalized inverse of $\mathbf{J}_v$, *i.e.*,

$$\mathbf{J}_v^{\dagger} = \mathbf{J}_v^T(\mathbf{J}_v\mathbf{J}_v^T)^{-1} \tag{2.79}$$

This virtual joint method to solve redundant robotic tasks suffers from the potential ill-conditioning of the augmented $\mathbf{J}$, *i.e.*, $\mathbf{J}_v$, and the additional computation cost required to solve $\mathbf{J}_v$.

2.7 Conclusion

This chapter reviewed the main methods to solve the inverse kinematics of kinematically redundant robotic tasks, which included the global and local optimization methods, and the artificial intelligence methods.

Functional redundancy-resolution, a sub-domain on redundancy resolution and the main study issue of this thesis, is overviewed with more details in this chapters.

By taking advantage of the functional redundancy existing regularly in the robotic machining tasks, the objective of the optimization is to optimize the joint trajectories corresponding to the whole required path. A local optimization method named as TWA is developed in Chapter 3 and 4. For approaching the best joint trajectories , two methods of adapting weights are developed in Chapter 5 and 6, respectively.

CHAPITRE 3

TWIST DECOMPOSITION APPROACH AND JOINT-LIMITS AVOIDANCE

3.1 Introduction

This chapter presents the Twist Decomposition Approach (TWA), which solves redundant robotic tasks requiring less than six-degrees-of-freedom. Instead of projecting the secondary task onto the null space of the Jacobian matrix in order to take advantage of the redundancy, the TWA directly decomposes the task into two orthogonal subspaces where the main and secondary tasks lie, respectively. This approach has shown to be efficient, *i.e.*, having a low computation cost, and accurate, *i.e.*, having a low round-off error amplification. In this chapter, several numerical examples are shown for different manipulators and tasks.

The content of this chapter was published in the Transactions of the Canadian Society for Mechanical Engineering[6] in 2005.

3.2 Kinematic Inversion of Functionally-redundant Manipulators

3.2.1 Orthogonal-Decomposition of Three-Dimensional Vectors

Decomposing any vector $(\cdot)$ of $\mathbb{R}^3$ into two orthogonal parts, $[\cdot]_M$, the component lying on the subspace, $\mathcal{M}$, and $[\cdot]_{M^\perp}$, the component lying in the orthogonal subspace, $\mathcal{M}^\perp$, using the *projector* $\mathbf{M}$ and an *orthogonal complement* of $\mathbf{M}$, namely $\mathbf{M}^\perp$,

as follows :

$$(\,\cdot\,) = [\,\cdot\,]_M + [\,\cdot\,]_{M^\perp} = \mathbf{M}(\,\cdot\,) + \mathbf{M}^\perp(\,\cdot\,) = (\mathbf{M} + \mathbf{M}^\perp)(\,\cdot\,) \tag{3.1}$$

It is apparent from eq. (3.1), that $\mathbf{M}$ and $\mathbf{M}^\perp$ are related by $\mathbf{M} + \mathbf{M}^\perp = \mathbf{1}$ and $\mathbf{M}\mathbf{M}^\perp = \mathbf{O}$, where $\mathbf{1}$ and $\mathbf{O}$ are the 3×3 identity and zero matrices, respectively. The orthogonal complement of $\mathbf{M}$ thus defined, $\mathbf{M}^\perp$, is therefore unique, and hence, both $\mathbf{M}$ and $\mathbf{M}^\perp$ are projectors that verify the following properties :

- Symmetry : $\quad [\mathbf{M}]^T = \mathbf{M}, \quad [\mathbf{M}^\perp]^T = \mathbf{M}^\perp$
- Idempotency : $\quad [\mathbf{M}]^2 = \mathbf{M}, \quad [\mathbf{M}^\perp]^2 = \mathbf{M}^\perp$
- Rank-complementarity : $\quad \mathrm{rank}(\mathbf{M}) + \mathrm{rank}(\mathbf{M}^\perp) = 3$
- Subspace-complementarity : $\quad \mathcal{M} \oplus \mathcal{M}^\perp = \mathbb{R}^3$

The projector $\mathbf{M}$ projects vectors of $\mathbb{R}^3$ onto the subspace $\mathcal{M}$, while the orthogonal projector $\mathbf{M}^\perp$ projects those vectors onto the orthogonal subspace $\mathcal{M}^\perp$. These projectors are given for the four possible dimensions i of subspaces of $\mathbb{R}^3$ as :

$$\mathbf{M}_i = \begin{cases} \mathbf{1} \\ \mathbf{P} \\ \mathbf{L} \\ \mathbf{O} \end{cases}, \quad \mathbf{M}_i^\perp = \begin{cases} \mathbf{O} & i = 3 \quad \Rightarrow \text{3-D task} \\ \mathbf{L} & i = 2 \quad \Rightarrow \text{2-D task} \\ \mathbf{P} & i = 1 \quad \Rightarrow \text{1-D task} \\ \mathbf{1} & i = 0 \quad \Rightarrow \text{0-D task} \end{cases}, \tag{3.2}$$

where the plane and line projectors, $\mathbf{P}$ and $\mathbf{L}$, respectively, are defined as :

$$\mathbf{P} \equiv \mathbf{1} - \mathbf{L}, \quad \mathbf{L} \equiv \mathbf{e}\mathbf{e}^T, \tag{3.3}$$

in which $\mathbf{e}$ is a unit vector along the line $\mathcal{L}$ and normal to the plane $\mathcal{P}$. The null-projector $\mathbf{O}$ is the 3×3 zero matrix that projects any vector of $\mathbb{R}^3$ onto the null-subspace $\mathcal{O}$, while the identity-projector $\mathbf{1}$ is the 3×3 identity matrix that projects any vector of $\mathbb{R}^3$ onto itself.

3.2.2 Orthogonal-Decomposition of Twists

Any twist array $(\cdot)$ of $2\times\mathbb{R}^3$ can also be decomposed into two orthogonal parts, $[\cdot]_{\mathcal{T}}$, the component lying on the task subspace, $\mathcal{T}$, and $[\cdot]_{\mathcal{T}^\perp}$, the component lying on the orthogonal task subspace (also designated as the redundant subspace), $\mathcal{T}^\perp$, using the *twist projector* $\mathbf{T}$ and an *orthogonal complement* of $\mathbf{T}$, namely $\mathbf{T}^\perp$, as follows :

$$(\cdot) = [\cdot]_{\mathcal{T}} + [\cdot]_{\mathcal{T}^\perp} = \mathbf{T}(\cdot) + \mathbf{T}^\perp(\cdot) = (\mathbf{T} + \mathbf{T}^\perp)(\cdot) \tag{3.4}$$

It is apparent from eq. (3.4), that $\mathbf{T}$ and $\mathbf{T}^\perp$ are projectors of twists that must verify all the properties of projectors in section 3.2.1. However, twists are not vectors of $\mathbb{R}^6$, and hence, projectors of twists cannot be defined as in eqs. (3.2) and (3.3), *e.g.*,

$$\mathbf{T} \neq \mathbf{t}\mathbf{t}^T, \quad \mathbf{T}^\perp \neq \mathbf{1} - \mathbf{t}\mathbf{t}^T, \tag{3.5}$$

but must rather be defined as block diagonal matrices of projectors of $\mathbb{R}^3$, *i.e.*,

$$\mathbf{T} \equiv \begin{bmatrix} \mathbf{M}_\omega & \mathbf{O} \\ \mathbf{O} & \mathbf{M}_v \end{bmatrix}, \quad \mathbf{T}^\perp \equiv \mathbf{1} - \mathbf{T} = \begin{bmatrix} \mathbf{1} - \mathbf{M}_\omega & \mathbf{O} \\ \mathbf{O} & \mathbf{1} - \mathbf{M}_v \end{bmatrix}, \tag{3.6}$$

where $\mathbf{M}_\omega$ and $\mathbf{M}_v$ are projectors of $\mathbb{R}^3$ defined in eqs. (3.2) and (3.3) which allow the projection of the angular and translational velocity vectors, respectively. It is noteworthy that the matrices of eq. (3.5) do not verify the properties of projectors, and hence, cannot be used for orthogonal decomposition. Finally, eq. (3.4) becomes

$$\mathbf{t} = \mathbf{t}_{\mathcal{T}} + \mathbf{t}_{\mathcal{T}^\perp} = \mathbf{T}\mathbf{t} + (\mathbf{1} - \mathbf{T})\mathbf{t}. \tag{3.7}$$

3.2.3 Twist Decomposition Algorithm in Solving Functional Redundancy

For functionally-redundant serial manipulators, it is possible to decompose the twist of the EE into two orthogonal parts, one lying into task subspace and another one lying into the redundant subspace. Substituting eq. (3.7) into eq. (2.24) yields

$$\Delta\boldsymbol{\theta} = \underbrace{(\mathbf{J}^{\dagger}\mathbf{T})\mathbf{t}}_{\text{task displacement}} + \underbrace{\mathbf{J}^{\dagger}(\mathbf{1}-\mathbf{T})\mathbf{J}\mathbf{h}}_{\text{redundant displacement}}, \tag{3.8}$$

where $\mathbf{h}$ is an arbitrary vector of $\mathcal{J}$ allowing to satisfy a secondary task. Vector $\mathbf{h}$ is often chosen as the gradient of an objective function to be minimized (Baron [59]). For the avoidance of joint-limits, the objective function z can be written as to maintain the manipulator as close as possible to the mid-joint position $\bar{\boldsymbol{\theta}}$, *i.e.*,

$$z = \frac{1}{2}(\boldsymbol{\theta}-\bar{\boldsymbol{\theta}})^{T}\mathbf{W}^{T}\mathbf{W}(\boldsymbol{\theta}-\bar{\boldsymbol{\theta}}) \rightarrow \min_{\boldsymbol{\theta}}, \tag{3.9}$$

with $\bar{\boldsymbol{\theta}}$ and $\mathbf{W}$ being defined as

$$\bar{\boldsymbol{\theta}} \equiv \frac{1}{2}(\boldsymbol{\theta}_{max}+\boldsymbol{\theta}_{min}), \quad \mathbf{W} \equiv \mathbf{diag}\left(\frac{1}{\boldsymbol{\theta}_{max}-\boldsymbol{\theta}_{min}}\right). \tag{3.10}$$

Vector $\mathbf{h}$ is thus chosen as minus the gradient of z, *i.e.*,

$$\mathbf{h} = -\boldsymbol{\nabla} z. \tag{3.11}$$

The first part of the RHS of eq. (3.8) is the joint displacement required by the task, while the second part is the joint displacement in the redundant subspace (or irrelevant to the task). Clearly, eq. (3.8) does not require the projection onto the null-space of $\mathbf{J}$ as most of the redundancy-resolution algorithms do, but rather requires an orthogonal projection based on the instantaneous geometry of the task to be accomplished.

Algorithm 3.1 : Twist Decomposition Algorithm

1 $\boldsymbol{\theta} \longleftarrow$ initial joint position;

2 $\{\mathbf{p}_d, \mathbf{Q}_d\} \longleftarrow$ desired EE position and orientation;

3 $\{\mathbf{p}, \mathbf{Q}\} \Leftarrow \mathbf{DKP}(\boldsymbol{\theta})$

4 $\Delta\mathbf{Q} \Leftarrow \mathbf{Q}^T\mathbf{Q}_d$

5 $\Delta\mathbf{p} \Leftarrow \mathbf{p}_d - \mathbf{p}$

6 $\mathbf{t} \leftarrow \begin{bmatrix} \mathbf{Qvect}(\Delta\mathbf{Q}) \\ \Delta\mathbf{p} \end{bmatrix}$

7 $\mathbf{DKP}(\boldsymbol{\theta}) \Rightarrow \begin{cases} \mathbf{e} \Rightarrow \mathbf{M}_\omega \\ \mathbf{f} \Rightarrow \mathbf{M}_v \\ \mathbf{J} \end{cases}$,

8 $\mathbf{T} \Leftarrow \begin{bmatrix} \mathbf{M}_\omega & \mathbf{O} \\ \mathbf{O} & \mathbf{M}_v \end{bmatrix}$,

9 $\Delta\boldsymbol{\theta} \Leftarrow \mathbf{J}^\dagger\mathbf{T}\mathbf{t} + \mathbf{J}^\dagger(\mathbf{1} - \mathbf{T})\mathbf{J}\mathbf{h}$

10 if $\|\Delta\boldsymbol{\theta}\| < \epsilon$ then stop;
else

11 $\boldsymbol{\theta} \Leftarrow \boldsymbol{\theta} + \Delta\boldsymbol{\theta}$, and go to 3.

As shown in Algorithm 3.1, eq. (3.8) is used within a resolved-motion rate method. At lines 1-3, the joint position $\boldsymbol{\theta}$ and the desired EE pose $\{\mathbf{p}_d, \mathbf{Q}_d\}$ are first initialized, then the actual EE pose $\{\mathbf{p}, \mathbf{Q}\}$ is computed with the direct kinematic model $\mathbf{DKP}(\boldsymbol{\theta})$. At lines 4-6, an EE displacement $\mathbf{t}$ is computed from the difference between the desired and actual EE poses. The $\mathbf{vect}(\cdot)$ at line 6 is the function transforming a 3×3 rotation matrix into an axial vector as defined in [1] (page 34). At lines 7-8, the instantaneous orthogonal twist projector $\mathbf{T}$ is computed from $\mathbf{DKP}(\boldsymbol{\theta})$. At line 9, the orthogonal decomposition method is used to compute the corresponding joint displacement $\Delta\boldsymbol{\theta}$. Finally, the algorithm is stop whenever the norm of $\Delta\boldsymbol{\theta}$ is smaller than a certain threshold ϵ.

3.3 Example I : Puma 500

joint	θ_i	a_i	b_i	α_i
1	θ_1	0.0	0.0	$-\pi/2$
2	θ_2	0.4318	0.0	0.0
3	θ_3	-0.0203	0.1491	$\pi/2$
4	θ_4	0.0	0.4330	$-\pi/2$
5	θ_5	0.0	0.0	$\pi/2$
6	θ_6	0	0.055	0
unit	*rad.*	*m*	*m*	*rad.*

TAB. 3.1 DH parameters of PUMA 500.

When performing arc-welding operations, the electrode of the welding tool has an axis of symmetry around which the welding tool may be rotated without interfering with the task to be performed. This axis describes the geometry of the functional redundancy (or the redundant subspace of twists). The unit vector $\mathbf{e}$ denote the orientation of the symmetry axis along the electrode. The projection of $\boldsymbol{\omega}$ along $\mathbf{e}$ is the irrelevant component of $\boldsymbol{\omega}$, while its projection onto the plane normal to $\mathbf{e}$ is the relevant component of $\boldsymbol{\omega}$. For a general arc-welding task around the electrode axis $\mathbf{e}$, the twist projector is defined as

$$\mathbf{T}_{weld} \equiv \begin{bmatrix} (\mathbf{1} - \mathbf{e}\mathbf{e}^T) & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}, \quad \mathbf{T}_{weld}^{\perp} \equiv \begin{bmatrix} \mathbf{e}\mathbf{e}^T & \mathbf{0} \\ \mathbf{0} & 0 \end{bmatrix}, \tag{3.12}$$

Now, substituting eq. (3.12) into eq. (3.8) yields

$$\boldsymbol{\Delta\theta} = \mathbf{J}^{\dagger}\mathbf{T}_{weld}\mathbf{t} + \mathbf{J}^{\dagger} \begin{bmatrix} \mathbf{e}\mathbf{e}^T\mathbf{A}\mathbf{h} \\ \mathbf{0} \end{bmatrix}, \tag{3.13}$$

where $\mathbf{A}$ is the upper part of $\mathbf{J}$ as defined in eq. (2.15). Equation (3.13) can be used as line 9 of Algorithm 3.1 in order to solve the inverse kinematics of serial manipulators while performing a general arc-welding task.

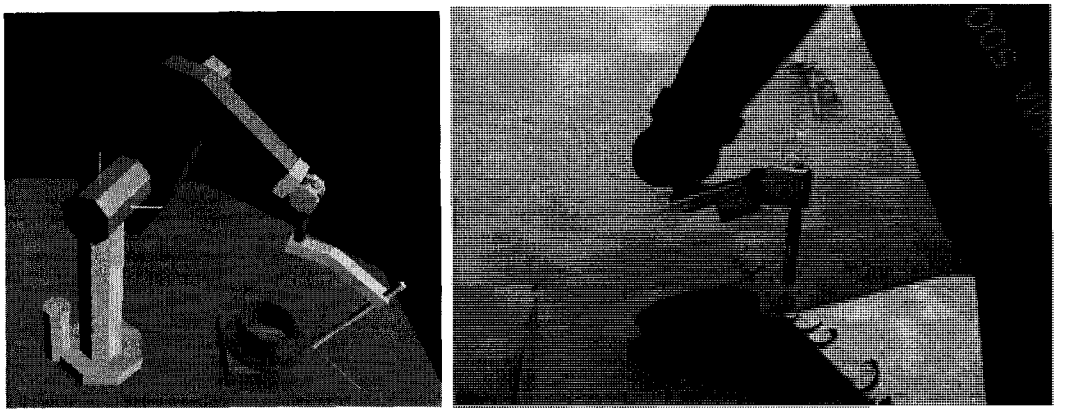


FIG. 3.1 Arc-welding task with the PUMA 500 manipulator

As shown in Fig. 3.1, a PUMA 500 serial manipulator is used to perform a pipe-to-bride welding task. Its DH parameters are described in Table 3.1. The welding tool has a transformation matrix $\mathbf{A}_{tool}$ as

$$\mathbf{A}_{tool} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\frac{\pi}{6}) & -\sin(\frac{\pi}{6}) & 0.1 \\ 0 & \sin(\frac{\pi}{6}) & \cos(\frac{\pi}{6}) & 0.501 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.14}$$

The EE must perform consecutively the trajectory Λ_1 ($T = 285$ sec.), *i.e.*,

$$\mathbf{p} = \begin{bmatrix} 0.1\cos(\omega t) \\ 0.6 + 0.1\sin(\omega t) \\ -0.59 \end{bmatrix}, \quad \mathbf{Q} = \begin{bmatrix} \cos\alpha & -\sin\alpha\cos\beta & \sin\alpha\sin\beta \\ \sin\alpha & \cos\alpha\cos\beta & \cos\alpha\cos\beta \\ 0 & \sin\beta & \cos\beta \end{bmatrix}, \tag{3.15}$$

with $\alpha = \frac{\pi}{2} + \omega t$, $\beta = \frac{-3\pi}{4}$, $\omega = \frac{2\pi}{T}$, $0 \leq t \leq T$, where distances and angles are expressed in meters and radians, respectively. The orientation of the electrode axis, namely $\mathbf{e}$, can be computed as

$$\mathbf{e} = \mathbf{Q}_1\mathbf{Q}_2\cdots\mathbf{Q}_6\mathbf{k}, \quad \mathbf{k} \equiv [0 \;\; 0 \;\; 1]^T. \tag{3.16}$$

The secondary task $\mathbf{h}$ is chosen to avoid the joint-limits such that :

$$\mathbf{h} = -\mathbf{W}(\boldsymbol{\theta} - \boldsymbol{\theta}_0), \tag{3.17}$$

where $\mathbf{W}$ is a positive-definite weighting matrix given as in eq. (3.10), and $\boldsymbol{\theta}_0$ the mean-joint position of the robotic manipulator, *i.e.*,

$$\boldsymbol{\theta}_0 \equiv \begin{bmatrix} \pi/2 & -\pi/3 & \pi & \pi/4 & \pi/3 & \pi \end{bmatrix}^T. \tag{3.18}$$

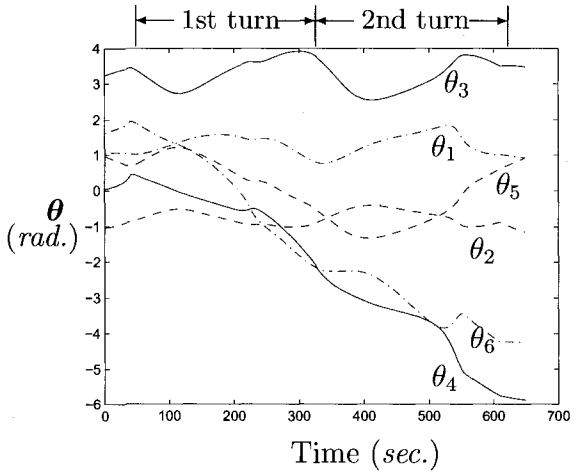


FIG. 3.2 Joint position with respect to time without using RR scheme

Figure 3.2 shows the joint positions to perform twice the trajectory Λ_1 as computed by the resolved-motion rate method without considering the functional redundancy, *i.e.*, using eq. (2.24) at line 9 of Algorithm 3.1. Apparently, without taking advantage of the axis of symmetry of the electrode, the manipulator is able to perform the first turn of the trajectory Λ_1 while the second consecutive turn is not possible without exceeding the joint limits. Figure 3.3 shows the joint positions to perform twice the trajectory Λ_1 as computed by the augmented approach, *i.e.*, using eq. (2.25) at line 9 of Algorithm 3.1. Apparently, the manipulator is able to perform multiple conse-

cutive turns without exceeding the joint limits. However, excessive joint velocities appear at every turn. Figure 3.4 shows the joint positions for two consecutive turns as computed by the TWA, *i.e.*, using eq. (3.8) at line 9 of algorithm 3.1. Apparently, the manipulator is able to perform multiple consecutive turns without exceeding the joint limits. Excessive joint velocities appear only at the first turn, because of the arbitrarily chosen initial conditions.

However, starting from the second turn, the joint space trajectories have no excessive joint velocities, and the same joint trajectories repeated in the following turns. The reason of causing this phenomenon is that the manipulator posture reaching the first path point is optimized after first turn.

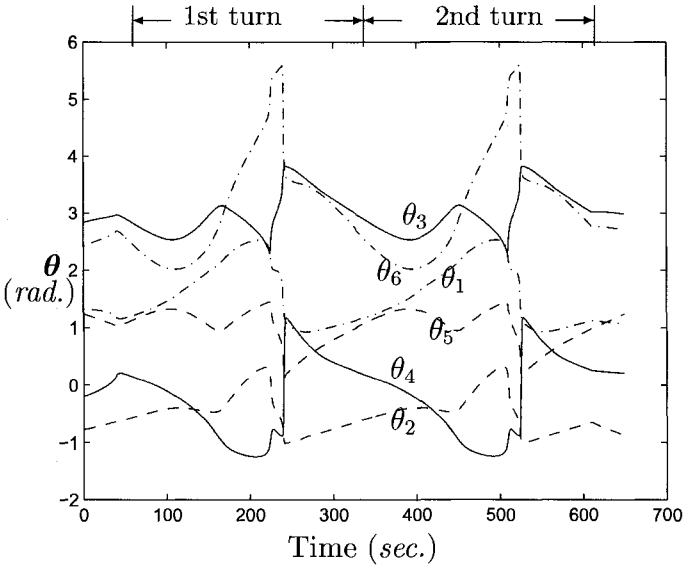


FIG. 3.3 Joint positions with respect to time for the virtual joint method

For the sake of comparing the performance of the augmented and projected approaches, the trajectory Λ_1 is discretized into n segments, for which the average reaching accuracy of their end-point is computed. For this performance evaluation, lines 3 to 10 of Algorithm 3.1 is performed at constant number of iterations rather than until a threshold is reached. The position error of segment j, namely e_{pj}, is computed as the norm of the difference between the corresponding reached position

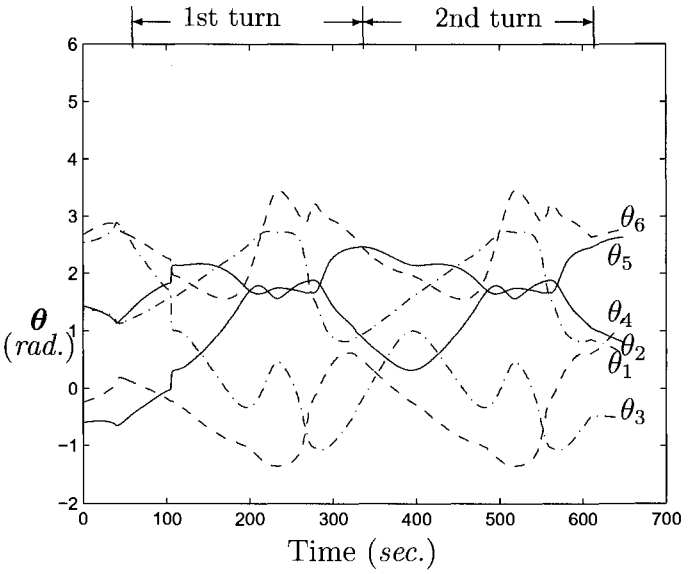


FIG. 3.4 Joint positions with respect to time for the twist decomposition method

$\mathbf{p}_{rj}$ and desired position $\mathbf{p}_{dj}$, *i.e.*,

$$e_{pj} = \|\mathbf{p}_{rj} - \mathbf{p}_{dj}\|, \tag{3.19}$$

and the average position error $\overline{e}_p$ along the whole trajectory Λ is given as

$$\overline{e}_p = \frac{1}{n}\sum_{j=1}^{n} e_{pj}. \tag{3.20}$$

Similarly, the orientation error of segment j, namely e_{ej}, is computed as the norm of difference between the corresponding reached orientation $\mathbf{Q}_{rj}$ and the desired orientation $\mathbf{Q}_{dj}$, *i.e.*,

$$e_{ej} = \|\mathbf{vect}(\mathbf{Q}_{rj}^T \mathbf{Q}_{dj})\|. \tag{3.21}$$

Since the rotation around the electrode axis does not affect the welding task, the orientation error around that axis is also irrelevant, and thus, only its projection onto the plane normal to that axis is meaningful. Let the unit vector $\mathbf{e}$ be aligned along the electrode axis in the base frame, then the projection of the orientation error of eq. (3.21) onto the plane normal to $\mathbf{e}$ is given as

$$e_{oj} = \|(\mathbf{1} - \mathbf{e}\mathbf{e}^T)\mathbf{Q}_{rj}\mathbf{vect}(\mathbf{Q}_{rj}^T\mathbf{Q}_{dj})\|, \tag{3.22}$$

and the average orientation error $\overline{e}_o$ along the whole trajectory Λ is given as

$$\overline{e}_o = \frac{1}{n}\sum_{j=1}^{n} e_{oj}. \tag{3.23}$$

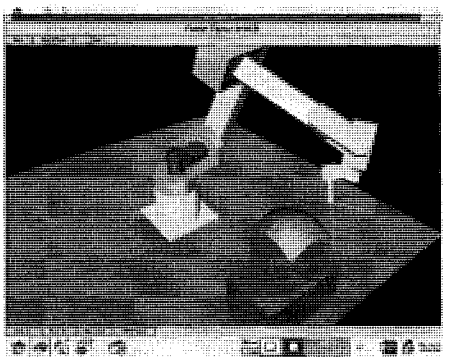
Table 3.2 shows the average position and orientation errors of the augmented and twist decomposition approaches computed with 20 iterations for each segment. It is apparent that the TWA has much lower position and orientation errors in the task space than the augmented approach. The TWA produces more accurate solutions than the augmented approach with the same number of iterations. In other words, the TWA is able to approach the desired posture faster than the augmented approach.

method	$\overline{e}_p$	$\overline{e}_o$
augmented	2.4742×10^{-6}	2.2511×10^{-5}
TWA	9.4712×10^{-8}	9.2936×10^{-6}
unit	meter	rad.

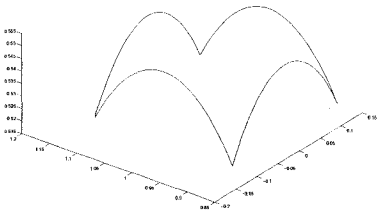
TAB. 3.2 Errors of the augmented and projected approaches.

3.4 Example II : Fanuc M16iB

In this example[62], the end-milling operation is performed by a Fanuc M16iB. The graphic simulator and translation path of the task are shown in Figure 3.5. When performing the end-milling operations, the milling tool has a symmetry axis, around which the milling tool may be rotated without interfering with the task to be performed. This axis describes the geometry of the functional redundancy (or the redundant subspace of twists). The unit vector $\mathbf{e}$ denote the orientation of the symmetry axis of the tool. The projection of $\boldsymbol{\omega}$ along $\mathbf{e}$ is the irrelevant component of $\boldsymbol{\omega}$, while its



(a)



(b)

FIG. 3.5 (a) Graphic simulator of the end-milling operation with a Fanuc M16iB (b) the end-point path in the base frame.

projection onto the plane normal to $\mathbf{e}$ is the relevant component of $\boldsymbol{\omega}$.

For a general end-milling task and the symmetry axis $\mathbf{e}$, the projectors $\mathbf{M}_\omega$ and $\mathbf{M}_v$ are defined as

$$\mathbf{M}_\omega \equiv [\mathbf{1} - \mathbf{e}\mathbf{e}^T], \quad \mathbf{M}_v \equiv [\mathbf{1}] \tag{3.24}$$

After substituting eq.(3.24) into eq. (3.6), eq.(3.8) becomes

$$\boldsymbol{\Delta\theta} = \mathbf{J}^\dagger \mathbf{T} \boldsymbol{\Delta} t + \mathbf{J}^\dagger \begin{bmatrix} \mathbf{e}\mathbf{e}^T \mathbf{A}\mathbf{h} \\ \mathbf{0} \end{bmatrix}, \tag{3.25}$$

where $\mathbf{A}$ is the upper part of $\mathbf{J}$ as defined in eq.(2.15). Equation (3.25) can be used as line 9 of Algorithm 3.1 in order to solve the inverse kinematics of serial manipulators while performing a general end-milling task. A Fanuc M16iB serial robot is used to

perform a sphere-edge milling task. Its DH parameters and joint limits are described in Table 3.3.

joint	a_i	b_i	α_i	Min.	Max.
1	0.150	0.525	$-\pi/2$	-2.9671	2.9671
2	0.770	0.0	0.0	-2.1817	2.1817
3	0.10	0.0	$\pi/2$	-4.0143	4.0143
4	0.0	0.740	$-\pi/2$	-3.4907	3.4907
5	0.0	0.0	$\pi/2$	-2.4435	2.4435
6	0	0.10	0	-7.8540	7.8540
unit	*m*	*m*	*rad.*	*rad.*	*rad.*

TAB. 3.3 DH parameters of the Fanuc M16iB.

meth.	*a.* Itera. No. = 1		*b.* $p_e < 0.001$; $o_e < 0.01$	
	Period A	Period B	Period A	Itera. No.
Aug.	7.2546	1.3748	8.0307	4809
TWA	5.6308	0.3950	6.0448	4787

TAB. 3.4 Computation time of the augmented (Aug.) and twist decomposition approaches (TWA) : (a) fixing the iteration number as one; (b) reaching the same accuracy. Period A — the computation time from line 3 to 11 in Algorithm 3.1; Period B — the computation time of line 9 in Algorithm 3.1. All the unit of time are seconds.

The milling tool has a transformation matrix $\mathbf{A}_{tool}$ as

$$\mathbf{A}_{tool} = \begin{bmatrix} \cos\beta & 0 & -\sin\beta & -0.0785 \\ 0 & 1 & 0 & 0 \\ \sin\beta & 0 & \cos\beta & 0.154 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \tag{3.26}$$

with $\beta = -0.4398$ *rad.* The EE must perform consecutively the trajectory Λ_2 ($T = 50$ sec.). The symmetry axis of the milling tool is always normal to the sphere surface along the trajectory Λ_2. The velocity of the tool is changing according the sine curve

(from $-\pi/2$ to $\pi/2$) along each edge. That is to say, EE velocity is zero at each corner of the path, and is accelerated smoothly to maximum at the middle point of the edge, then the EE velocity is decelerated smoothly till zero at the next corner. The path planning is done with the help of CATIA V5 machining module. The secondary task $\mathbf{h}$ is chosen to avoid the joint-limits such that :

$$\mathbf{h} = -\mathbf{W}(\boldsymbol{\theta} - \boldsymbol{\theta}_0), \tag{3.27}$$

where $\mathbf{W}$ is a positive-definite weighting matrix as in eq.(3.10) and $\boldsymbol{\theta}_0$ the mid-joint position of the robotic manipulator, *i.e.*,

$$\mathbf{W} \equiv \mathbf{Diag}(\ 0.10 \quad 0.12 \quad 0.10 \quad 0.10 \quad 0.10 \quad 0.10\ ), \tag{3.28}$$

$$\boldsymbol{\theta}_0 \equiv \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}^T. \tag{3.29}$$

Figure 3.6(a) shows the joint positions to perform twice the trajectory Λ as computed by the resolved-motion rate method without considering the functional redundancy, *i.e.*, using eq.(2.24) at line 9 of Algorithm 3.1. Apparently, without taking advantage of the tool symmetry axis, the manipulator is unable to perform the trajectory Λ. Figure 3.6(b) shows the joint positions to perform twice the trajectory Λ as computed by the augmented approach, *i.e.*, using eq.(2.25) at line 9 of Algorithm 3.1. Apparently, joint 4 is out of the joint limit starting from 30 second. On the other hand, excessive joint velocities appear at every turn. Figure 3.6(c) shows the joint positions for two consecutive turns as computed by TWA, *i.e.*, using eq.(3.8) at line 9 of Algorithm 3.1. Apparently, the manipulator is able to perform multiple consecutive turns without exceeding the joint limits shown in Table 3.3.

For the sake of comparing the efficiency, the computation time of the augmented and twist decomposition approaches are studied as shown in Table II. Apparently, the TWA is much faster than the augmented approach, and the computation of eq.(3.8) is almost three and half times faster than eq.(2.25). The main reason, which lags the

computation of augmented approach, is the computation of the pseudo inverse of a non-square Jacobian matrix rather than the inverse of a square jacobian matrix as in the TWA.

3.5 Example III : Fanuc 710c50

3.5.1 Task Description

The arc-welding operation is performed with a Fanuc M-710 iC/50. The DH parameters and joint limits of this robot are described in Table 3.5. The welding parts is made of two cylinder of radius $400mm$ and $250mm$, respectively. Figure 3.7 shows the robot and the welding parts. The welding path Λ_3 is the intersection curve of

Joint	θ_i	a_i	b_i	α_i	Max.	Min.
1	0	150	0	-90	180	-180
2	-90	870	0	180	75	-60
3	0	170	0	-90	230	-131.8
4	0	0	1016	90	360	-360
5	0	0	0	-90	125	-125
6	0	0	175	180	360	-360
unit	*degree*	*mm*	*mm*	*degree*	*degree*	*degree*

TAB. 3.5 DH parameters of Fanuc M-710ic/50.

the two cylinders, as shown in Fig. 3.8. The length of Λ_3 is $1829mm$. The EE must perform the welding operation consecutively along the path Λ_3, with the speed about $75mm/min$. The welding tool symmetry axis must always be directed toward point P, which locates at $[\ 0\ \ 0\ \ 0\ ]^T$ in the part frame, the intersection point of the axes of the two cylinders. The transformation matrix from the robot base frame to the

part frame is

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & 1150 \\ 0 & 1 & 0 & 200 \\ 0 & 0 & 1 & -200 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{3.30}$$

Figure 3.9 shows the joint positions to perform twice the trajectory Λ as computed by the resolved-motion rate method without considering the functional redundancy, *i.e.*, using eq.(2.24) at line 9 of Algorithm 3.1. Apparently, without taking advantage of the redundant axis, the sixth joint goes out of its joint limit, the manipulator is unable to perform this task. Thus, an optimization is needed to avoid the joint-limits.

3.5.2 Test I : Joint-limits Avoidance

The secondary task $\mathbf{h}$ is chosen to avoid the joint-limits such that

$$\mathbf{h} = -\boldsymbol{\nabla} z_{joint} = \mathbf{Diag}(\mathbf{w}_{ini})(\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}), \tag{3.31}$$

where $\bar{\boldsymbol{\theta}}$ is the mid-joint position of the robotic manipulator, and

$$\mathbf{w}_{ini} = \begin{bmatrix} 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 \end{bmatrix}. \tag{3.32}$$

Equation(3.31) is used as the secondary task to avoid the joint-limits. Figure 3.10 shows the joint positions for two consecutive turns as computed by the TWA, *i.e.*, using eq.(3.8) at line 9 of Algorithm 3.1. The trajectories of each joint are within its joint limits. The maximum rotation velocity of the fourth and sixth joints are $74.5 deg/sec$ and $56.9 deg/sec$, respectively.

Although the TWA, with only joint limits avoidance, can reach a solution within the joint limits. The robot configuration is close to singularity at the instants 61 and 194

seconds, where the corresponding singularity performance indices are

$$\omega_{cond} = 163, \quad \omega_{mom} = 0.0066, \quad \omega_{ps} = 157. \tag{3.33}$$

Clearly, these indices are very close to a singularity. Thus, a singularity avoidance strategy becomes necessary in addition to the joint-limits avoidance for TWA.

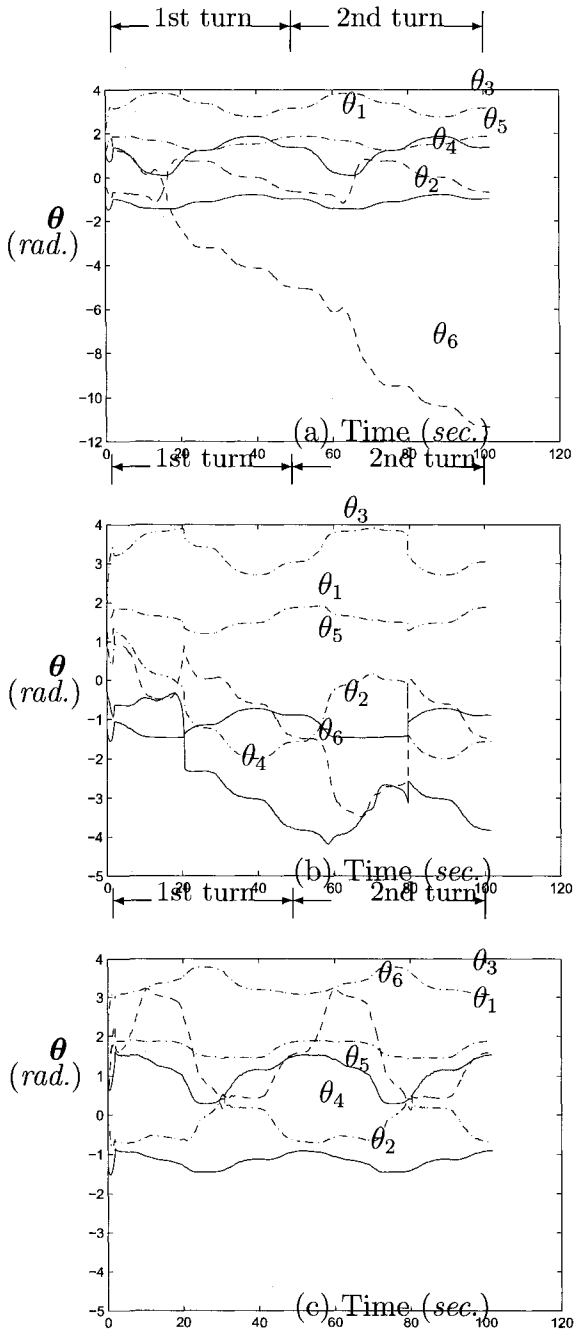


FIG. 3.6 Joint space trajectories of two consecutive turns of trajectory Λ as computed by the resolved-motion rate method with different approaches : a) No redundant-resolution approach; b) Augmented approach c) Twist decomposition approach

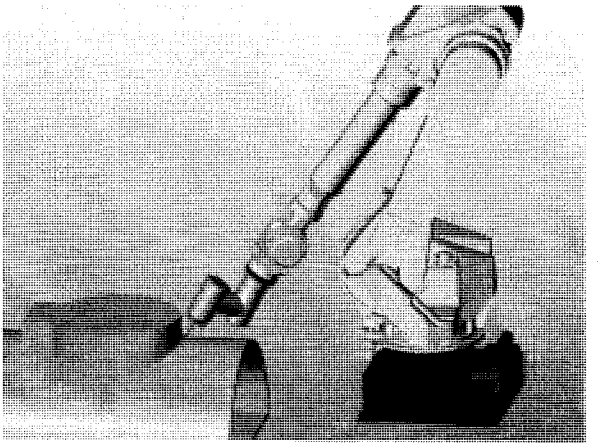


FIG. 3.7 Graphic simulator of arc-welding operation with a Fanuc M-710 iC/50 .

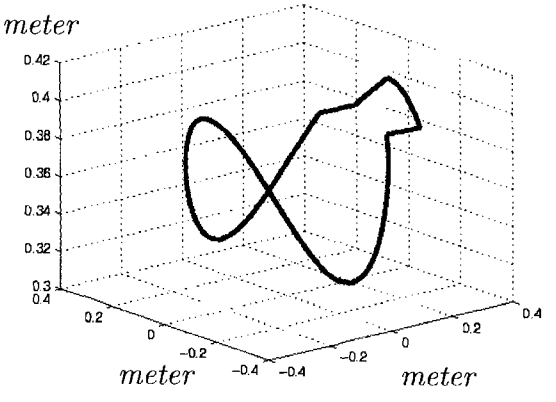


FIG. 3.8 The welding path in the part frame.

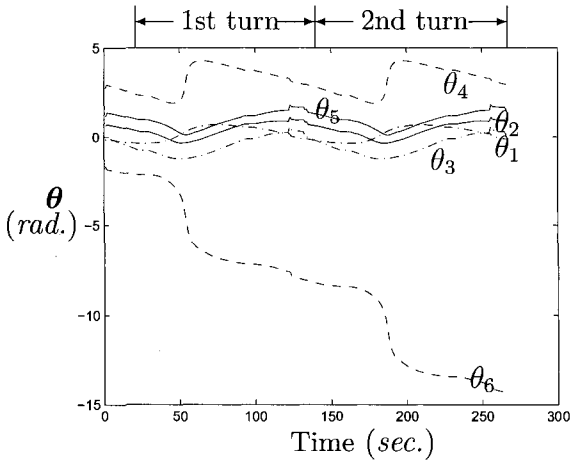


FIG. 3.9 Joint positions of two consecutive turns of trajectory Λ_3 as computed by the resolved-motion rate method (without considering the functional redundancy).

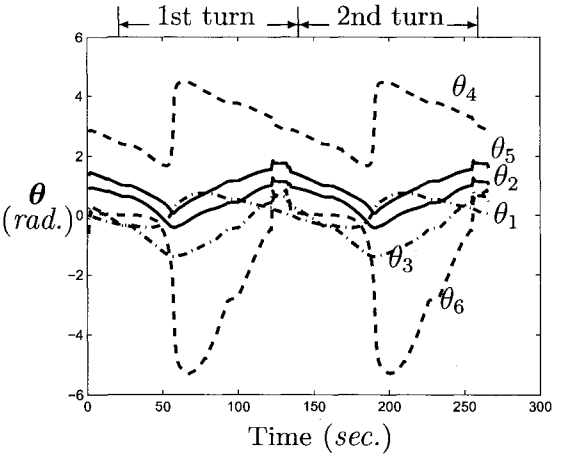


FIG. 3.10 Test I : Joint position with respect to time with the joint-limits avoidance strategy.

Twist projector : $\mathbf{T} \equiv \begin{bmatrix} \mathbf{M}_\omega & \mathbf{O} \\ \mathbf{O} & \mathbf{M}_v \end{bmatrix}$		Projector of angular velocity vector : $\mathbf{M}_\omega$			
		0-D task : $\mathbf{O}$	1-D task : $\mathbf{e}\mathbf{e}^T$	2-D task : $\mathbf{1} - \mathbf{e}\mathbf{e}^T$	3-D task : $\mathbf{1}$
Projector of translation velocity vector $\mathbf{M}_v$	0-D task $\mathbf{O}$	$\begin{bmatrix} \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} \end{bmatrix}$ All rotation and translation of EE are irrelevant motion.	$\begin{bmatrix} \mathbf{e}\mathbf{e}^T & \mathbf{O} \\ \mathbf{O} & \mathbf{O} \end{bmatrix}$	$\begin{bmatrix} (\mathbf{1} - \mathbf{e}\mathbf{e}^T) & \mathbf{O} \\ \mathbf{O} & \mathbf{O} \end{bmatrix}$	$\begin{bmatrix} \mathbf{1} & \mathbf{O} \\ \mathbf{O} & \mathbf{O} \end{bmatrix}$
	1-D task $\mathbf{f}\mathbf{f}^T$	$\begin{bmatrix} \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{f}\mathbf{f}^T \end{bmatrix}$	$\begin{bmatrix} \mathbf{e}\mathbf{e}^T & \mathbf{O} \\ \mathbf{O} & \mathbf{f}\mathbf{f}^T \end{bmatrix}$	$\begin{bmatrix} (\mathbf{1} - \mathbf{e}\mathbf{e}^T) & \mathbf{O} \\ \mathbf{O} & \mathbf{f}\mathbf{f}^T \end{bmatrix}$	$\begin{bmatrix} \mathbf{1} & \mathbf{O} \\ \mathbf{O} & \mathbf{f}\mathbf{f}^T \end{bmatrix}$
	2-D task $\mathbf{1} - \mathbf{f}\mathbf{f}^T$	$\begin{bmatrix} \mathbf{O} & \mathbf{O} \\ \mathbf{O} & (\mathbf{1} - \mathbf{f}\mathbf{f}^T) \end{bmatrix}$	$\begin{bmatrix} \mathbf{e}\mathbf{e}^T & \mathbf{O} \\ \mathbf{O} & (\mathbf{1} - \mathbf{f}\mathbf{f}^T) \end{bmatrix}$	$\begin{bmatrix} (\mathbf{1} - \mathbf{e}\mathbf{e}^T) & \mathbf{O} \\ \mathbf{O} & (\mathbf{1} - \mathbf{f}\mathbf{f}^T) \end{bmatrix}$	$\begin{bmatrix} \mathbf{1} & \mathbf{O} \\ \mathbf{O} & (\mathbf{1} - \mathbf{f}\mathbf{f}^T) \end{bmatrix}$
	3-D task $\mathbf{1}$	$\begin{bmatrix} \mathbf{O} & \mathbf{O} \\ \mathbf{O} & \mathbf{1} \end{bmatrix}$ Task requiring only 3-D positioning. Orienting are irrelevant Example : gluing	$\begin{bmatrix} \mathbf{e}\mathbf{e}^T & \mathbf{O} \\ \mathbf{O} & \mathbf{1} \end{bmatrix}$	$\begin{bmatrix} (\mathbf{1} - \mathbf{e}\mathbf{e}^T) & \mathbf{O} \\ \mathbf{O} & \mathbf{1} \end{bmatrix}$ Task requiring 3-D positioning and 2-D orienting. Example : arc-welding ; laser-cutting.	$\begin{bmatrix} \mathbf{1} & \mathbf{O} \\ \mathbf{O} & \mathbf{1} \end{bmatrix}$ All rotation and translation are required. No functional redundancy exist.

TAB. 3.6 Twist projector matrices.

3.6 General Task Projectors

There are four possible dimension for the projectors $\mathbf{M}_\omega$ and $\mathbf{M}_v$ respectively, in principle, the twist projector can be constructed with 16 possibilities as shown in Table 3.6, from fully free motion (both of $\mathbf{M}_\omega$ and $\mathbf{M}_v$ are equal to $\mathbf{0}$) to completely constrained motion (both of $\mathbf{M}_\omega$ and $\mathbf{M}_v$ are equal to $\mathbf{1}$).

Thus, if the tasks have more than one-DOF of functional redundancy, TWA can still be applied with corresponding $\mathbf{M}_\omega$ and $\mathbf{M}_v$. For example, if a half-sphere end-milling tool is used, then the irrelevant motion include the rotation not only around the tool symmetry axis Z, but also may around the X and Y axes under some limitations. In this case, we have $\mathbf{M}_\omega = \mathbf{0}$ and $\mathbf{M}_v = \mathbf{1}$.

Moreover, TWA is able to work on the tasks not only related to the tool geometry, but also related to part geometry. For example, as handling a cylindrical part, the orientation of the handler around the cylinder axis becomes irrelevant. In this case, the twist must be decomposed in the part frame instead of the tool frame. These more generalized tasks still need more analysis and are left for future research works.

3.7 Conclusion

In this chapter, the kinematic inversion of functionally-redundant serial manipulators is formulated using the orthogonal decomposition of the twist of the EE into a task subspace and a redundant subspace. The numerical simulation of three examples of serial manipulators has shown that the TWA is more effective relative to the augmented and non-redundant approaches.

Moreover, the simulation shows that the numerical stability greatly depends on the initial posture and the weights between the joints. Furthermore, the numerical conditioning is sometime very bad, the latter should not appear if taking the conditioning

into account in the secondary task as presented in the next chapter.

CHAPITRE 4

JOINT-LIMITS AND SINGULARITY AVOIDANCE IN TWA

4.1 Introduction

The aim of this chapter is to apply the TWA to optimize the joint-motion of six-revolute serial manipulators as avoiding joint-limits and singularities.

Some researchers[63] have realized that the success of the GPM relies on the evaluation of the performance criterion on the joints position. As the task requires not only the avoidance of the joint-limits but also keeping the robot configuration as far as possible from singularities, we need two different performance criteria relating to joint-limits and singularities, respectively. Here, the distance to the mid-joint position is used as the performance index relating to the joint-limits.

In order to avoid not only the joint limits but also the singularities at the same time, a new performance criterion relating to singularities, named *Parameter of Singularity*, is proposed and analyzed in this chapter. The numerical examples with application to arc-welding are also presented at the end of this chapter.

The content of this chapter has been published in Industrial Robot : An International Journal[64] in 2008.

4.2 Performance Criteria

4.2.1 Joint-limits Avoidance

In order to perform the joint-limits avoidance as the secondary task, we need to keep the manipulator as far as possible from its joint limits. Thus, in this case, the performance criterion can be written as to maintain the manipulator as close as possible to the mid-joint position $\bar{\boldsymbol{\theta}}$, *i.e.*,

$$z_{joint} = \frac{1}{2}(\boldsymbol{\theta} - \bar{\boldsymbol{\theta}})^T \mathbf{W}^T \mathbf{W}(\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}) \rightarrow \min_{\boldsymbol{\theta}}, \tag{4.1}$$

with $\bar{\boldsymbol{\theta}}$ and $\mathbf{W}$ being defined as

$$\bar{\boldsymbol{\theta}} \equiv \frac{1}{2}(\boldsymbol{\theta}_{max} + \boldsymbol{\theta}_{min}), \quad \mathbf{W} \equiv \mathbf{Diag}(\mathbf{w}). \tag{4.2}$$

The setting of the weighting vector $\mathbf{w}$ of eq.(4.2) is very important to the success of the optimization. If $\mathbf{w}$ is too small, the redundant displacement may not be sufficient to avoid the joint-limits; if $\mathbf{w}$ is too large, the redundant displacement may produce high joint velocities. Therefore, $\mathbf{w}$ is usually set based on trial and error. We will study this question in chapter 5, but here let us introduce the singularity avoidance.

4.2.2 Kinematic Singularity Avoidance

A singular posture is defined as the manipulator's configuration $\boldsymbol{\theta}^*$, where $\mathbf{J}(\boldsymbol{\theta}^*)$ is not full rank. The Singular Value Decomposition (SVD) of $\mathbf{J}$ is defined as

$$\mathbf{J} = \mathbf{U}\boldsymbol{\Sigma}\mathbf{V}^T, \quad \boldsymbol{\Sigma} \equiv \mathbf{Diag}(\sigma_1, \cdots, \sigma_m), \tag{4.3}$$

where $\mathbf{U}$ and $\mathbf{V}$ are, respectively, $m \times m$ and $n \times n$ orthogonal matrices, and $\mathbf{\Sigma}$ is an $m \times n$ diagonal matrix of the singular values of $\mathbf{J}$. The manipulator is in a singular state when at least one of its singular value equal to zero, *i.e.*, $\sigma_m = 0$. The corresponding column of $\mathbf{U}$ is referred to as the singular direction s. In this state, the motion along s is not possible[65].

Moreover, in the neighbourhood of singular points, even a small change in $\mathbf{t}$ requires an large change in $\boldsymbol{\Delta\theta}$, which is not-practically feasible and often dangerous on a real manipulator.

4.2.2.1 Manipulability Index

The first step in avoiding singularities is to detect them in the joint space. As presented by Yoshikawa[66], the *manipulability ellipsoid* represents an ability of manipulation, *i.e.*, the EE can move at higher speed in the direction of the major axis of the ellipsoid, while only with lower speed in the direction of the minor axes. If the ellipsoid is a sphere, the EE can move in all directions uniformly. Moreover, the larger the volume of the ellipsoid is, the faster the EE can move. One of the representative measures for the ability of manipulation derived from the manipulability ellipsoid is the volume of the ellipsoid. Hence, the measure of manipulability, namely ω_{mom}, could be written as

$$\omega_{mom} = \sqrt{det(\mathbf{J}\mathbf{J}^T)} = \sigma_1\sigma_2\cdots\sigma_m. \tag{4.4}$$

where $\sigma_1,\ \sigma_2,\ \cdots\sigma_m$ are the singular values of $\mathbf{J}$ ordered from maximum to minimum.

Thus, let $\mathbf{u}_i$ be the ith column vector of $\mathbf{U}$, the principle axes of the manipulability ellipsoid are $\sigma_1\mathbf{u}_1, \sigma_2\mathbf{u}_2, \ldots, \sigma_m\mathbf{u}_m$ (see Fig. 4.1). Marani et al.[67] have used the ω_{mom} as the distance criterion to avoid manipulator singularity. However, ω_{mom} can not represent the shape of the ellipsoid.

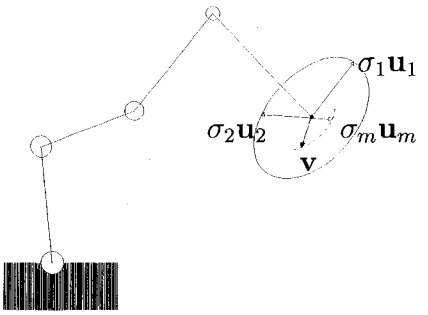


FIG. 4.1 Representation of the manipulability ellipsoid

4.2.2.2 Conditioning

Another index that might be induced from the manipulability ellipsoid is the *condition number*, namely ω_{cond}, of $\mathbf{J}$, which is defined as the ratio of the maximum and minimum radius of the ellipsoid, *i.e.*,

$$\omega_{cond} = \frac{\sigma_1}{\sigma_m} \geq 1, \tag{4.5}$$

and is independent of the ellipsoid's size. This ratio is an index of the directional uniformly of the ellipsoid. The closer to unity this index is, the more spherical the ellipsoid is. In a singular posture, the minor axis of the ellipsoid vanishes, *i.e.*, $\sigma_m = 0$, so the condition number becomes infinity. The condition number can be used as the distance criterion to avoid manipulator's singularities. But as versa to ω_{mom}, ω_{cond} only represents the shape of the ellipsoid.

4.2.2.3 Parameter of Singularity

Since ω_{mom} only represents the volume of the manipulability ellipsoid, while ω_{cond} only the shape. If keeping constant of the ellipsoid volume, *i.e.*, the product of the singular values of $\mathbf{J}$, but changing the ellipsoid shape, *i.e.*, the ration between σ_1 and

σ_m, ω_{mom} can not measure the difference but ω_{cond} can. Similarly, if scaling up the ellipsoid, there is no changing on ω_{cond} but ω_{mom} can measure the difference.

In order to better represent the manipulability ellipsoid, a performance index which represents both the volume and the shape of the manipulability ellipsoid is required. Here, we propose to combine the manipulability and conditioning together to generate a new performance index, named *Parameter of Singularity* (PS)[68], and is defined as

$$\omega_{ps} \equiv \sqrt{\frac{\omega_{cond}}{\omega_{mom}}} = \sqrt{\frac{1}{\sigma_2\sigma_3\cdots\sigma_m^2}}. \tag{4.6}$$

The index ω_{ps} represents both the volume and the shape of the manipulability ellipsoid at the same time. The smaller ω_{ps} is, the more spherical the ellipsoid is or the larger the ellipsoid's volume is, the faster the EE can move in the direction of the minor axis. Thus, the ω_{ps} can be used as a continuous measure that evaluate the kinematic quality of a robotic posture. The greater the ω_{ps} is, the worse the conditioning is or the lower the manipulability is.

In order to detect the singularity, a threshold singularity value ω_o is set. When ω_{ps} is passing over ω_o, the corresponding configuration at this instant is recorded as $\boldsymbol{\theta}_{Ts}$, and the performance criterion is activated to maintain the manipulator as close as possible to $\boldsymbol{\theta}_{Ts}$ in the following steps, until the ω_{ps} is lower than the preset threshold value ω_o. The performance criterion is written as

$$z_{sing} = \frac{\omega_{ps}^2}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_{Ts})^T\mathbf{K}^T\mathbf{K}(\boldsymbol{\theta} - \boldsymbol{\theta}_{Ts}) \rightarrow \min_{\boldsymbol{\theta}}, \tag{4.7}$$

where $\mathbf{K}$ is a weighting matrix obtained with the same method as $\mathbf{W}$, *i.e.*, heuristically.

4.2.3 Joint-Limits and Singularity Avoidance

The two secondary tasks described above, joint-limits and singularity avoidances, can be combined into an unique performance criterion, which is to maintain the manipulator as close as possible to the mid-joint position $\bar{\boldsymbol{\theta}}$ and as far as possible to the singularities at the same time. The objective function could be written as

$$z = z_{joint} + z_{sing}. \tag{4.8}$$

By tuning $\mathbf{W}$ and $\mathbf{K}$, the relative importance between the two sub-tasks is adjusted. Vector $\mathbf{h}$ of eqs.(2.25, 3.8) is thus chosen as minus the gradient of z, *i.e.*,

$$\mathbf{h} = -\boldsymbol{\nabla} z = \mathbf{W}(\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}) + \mathbf{K}\omega_{ps}(\boldsymbol{\theta}_{Ts} - \boldsymbol{\theta}), \tag{4.9}$$

where $\mathbf{K} = 0$ as $\omega_{ps} < \omega_o$.

Thus eq.(3.8) could be rewritten as

$$\boldsymbol{\Delta\theta} = \underbrace{(\mathbf{J}^{\dagger}\mathbf{T})\mathbf{t}}_{\text{task displacement}} + \underbrace{\mathbf{J}^{\dagger}(1-\mathbf{T})\mathbf{J}\mathbf{W}(\bar{\boldsymbol{\theta}} - \boldsymbol{\theta})}_{\text{redundant displacement 1}} + \underbrace{\mathbf{J}^{\dagger}(1-\mathbf{T})\mathbf{J}\mathbf{K}\omega_{ps}(\boldsymbol{\theta}_{Ts} - \boldsymbol{\theta})}_{\text{redundant displacement 2}}. \tag{4.10}$$

In eq.(4.10), the redundant displacement is decomposed to two components. The first one creates the displacement for avoiding the joint limits, and the second for avoiding the singularity.

4.3 Numerical Examples

We reuse the same manipulator (FanucM710iC/50) and task as in example III of section 3.5. Since we named the testing in section 3.5 as Test I, the tests in this section are named as Test II and III.

4.3.1 Test II : Joint-limits and Singularity Avoidances

In the end of Test I, we already realize a singularity avoidance strategy is necessary on this task besides of joint-limits avoidance. Thus here, we combine the joint-limits and kinematic singularity avoidances into one objective function as

$$z = z_{joint} + z_{sing} \rightarrow \min_{\boldsymbol{\theta}} , \tag{4.11}$$

and hence, the solution can be obtained as

$$\begin{aligned} \mathbf{h} &= -(\nabla z_{joint} + \nabla z_{sing}) \\ &= \mathbf{Diag}(\mathbf{w}_{ini})(\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}) \\ &\quad +\mathbf{Diag}(\mathbf{k}_{ini})\omega_{ps}(\boldsymbol{\theta}_{Ts} - \boldsymbol{\theta}) \quad , \end{aligned} \tag{4.12}$$

where

$$\mathbf{w}_{ini} = \mathbf{k}_{ini} = \begin{bmatrix} 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 \end{bmatrix}. \tag{4.13}$$

The preset threshold ω_{ps} is 5, *i.e.*, eq.(4.12) is applied only when the $\omega_{ps} \geq 5$,

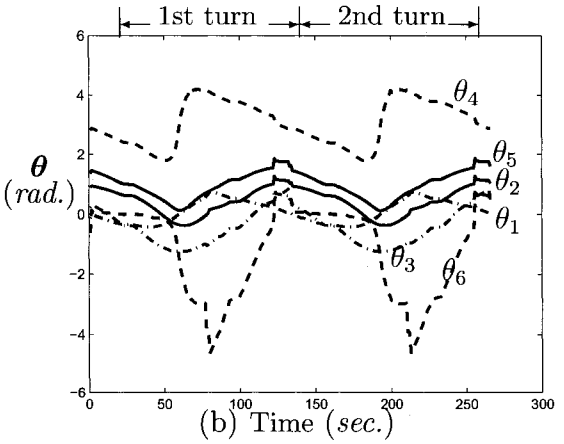


FIG. 4.2 Test II : Joint position with respect to time with the joint-limits and the singularity avoidance strategy.

otherwise, **h** is computed by eq.(3.31). The joint trajectory with TWA considering both joint-limits and singularity avoidance is shown in Fig. 4.2, where the joint motion range is smaller than the result reached in Test I. The closest configuration to singularity appears at instants 64 and 197 second, with the singularity indices of

$$\omega_{cond} = 56, \quad \omega_{mom} = 0.03, \quad \omega_{ps} = 43.2. \tag{4.14}$$

By comparing Test I and Test II, the maximum value of the condition number ω_{cond} is decreased from 163 to 56, *i.e.*, the shape of the manipulability ellipsoid becomes closer to a sphere. The minimum value of ω_{mom} increases from 0.0066 to 0.03, *i.e.*, the volume of the manipulability ellipsoid increase. The maximum value of ω_{ps} decreases from 156 to 43, *i.e.*, the EE move faster in the direction of the minor axis of the manipulability ellipsoid. As a result of the increasing of the distance from singularities, the maximum rotation velocity of the fourth joint is lower from $74.5 deg/sec$ to $22 deg/sec$, respectively.

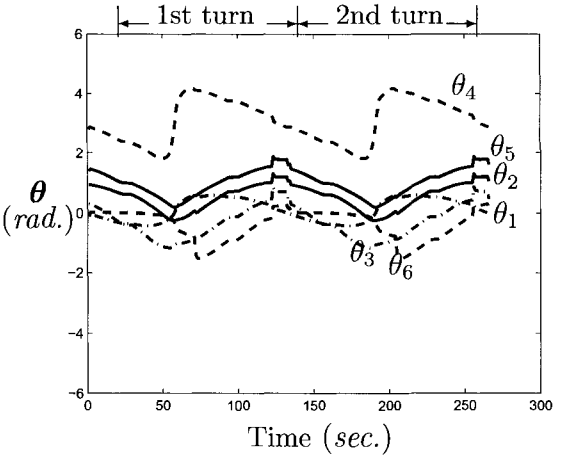


FIG. 4.3 Test III : Joint position with respect to time with joint-limits and singularity avoidant strategy and adapted weighting vectors.

4.3.2 Test III : Joint-limits and Singularity Avoidance with Adapted Weights

As mention before, the setting of the weighting vector has great influence on the result. Here, the weighting vectors $\mathbf{w}_{adp}$ and $\mathbf{k}_{adp}$ are adapted heuristically from their original setting, and then are used in eq.(4.12) as

$$\mathbf{w}_{adp} = \begin{bmatrix} 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.03 \end{bmatrix}, \tag{4.15}$$

$$\mathbf{k}_{adp} = \begin{bmatrix} 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.05 \end{bmatrix}. \tag{4.16}$$

The resulting joint trajectory with this adapted weighting vectors is shown in Fig. 4.3, where the motion range of the sixth joint is only 33% of Test II. The closest configuration to singularity appears at instant 61 and 194 second, with the singularity indices of

$$\omega_{cond} = 31, \quad \omega_{mom} = 0.076, \quad \omega_{ps} = 20.2. \tag{4.17}$$

These singularity indices are enhanced almost 100% as comparing to Test II. Among

Test	Motion range	Max. ω_{cond}	Min. ω_{mom}	Max. ω_{ps}	Max. J6 velocity	Max. J4 velocity	Error orien.	Error posi.
I	$-5.3 \sim 4.47$	163	0.0066	157	57	74	0.0013	0.0024
II	$-4.6 \sim 4.17$	56	0.03	43.2	56	22	0.0013	0.0024
III	$-1.52 \sim 4.17$	31	0.076	20.2	37	26	0.00077	0.0024
unit	*radian*				*deg./s*	*deg./s*	*radian*	*mm*

TAB. 4.1 Tests comparison where ω_{cond} is condition number of $\mathbf{J}$ and ω_{mom} is the manipulability measure.

all the three tests in sections 3.5.2, 4.3.1, and 4.3.2, the orientation and position errors between the reached and desired path in task space are studied in order to verify the solution accuracy. All these three tests can reach same level of position accuracy, *i.e.*, the maximum position error was $0.0024mm$. Relating to the orientation, Tests I and II can reach the same level accuracy, *i.e.*, the maximum norm of orientation error

is $0.00128 rad.$ Test III can reach higher orientation accuracy, its orientation error is $0.00077 rad.$

All these comparison are summarized in Table 4.1. Obviously, by applying the criterion in eq.(4.12) on Test II, the distance from singularities is increased while minimizing the the joint motion range; and with adapting the weighting vectors in Test III, a much better solution can be reached.

4.4 Conclusions

In order to keep the robot configuration as far as possible from singularities, the condition number ω_{cond} and manipulability ω_{mom} are not only used, but also combined to generate a new kinetostatic performance index ω_{ps} which evaluates both the shape and volume of the manipulability ellipsoid at the same time.

The numerical example compares the optimization results among three tests. The proposed twist decomposition approach, *i.e.*, TWA, with joint-limits and singularity avoidance as the secondary task is able to reach the best solution with this trajectory. Hence, the TWA has been proven to be able to optimize the joint space trajectory with the different performance criterions. However, we also see the selection of weights has very great influence on the success of the optimization, and deserve more study as we do in the following chapters.

CHAPITRE 5

SELF-ADAPTATION OF WEIGHTS IN TWA

5.1 Introduction

For a general 6-DOF task, many research works have reported algorithms allowing to choose an optimal solution when the manipulator has more joints than the corresponding DOF of its EE (e.g., [15, 22, 24, 25, 26, 59, 69, 70]). All of them use the *Gradient Projection Method* (GPM) to track the desired EE path, *i.e.*, the main task, and solve a trajectory optimization problem, *i.e.*, the secondary task. The latter is related to a performance criterion. The gradient of this function is projected onto the null space of the Jacobian matrix, namely **J**, in order to choose the best solution among the infinitely many that exist. Hence, the secondary task is performed under the constraint that the main task is realized.

The success of GPM relies greatly on the weighting parameters (the relative importance of the secondary task wrt. the main task), that have to be precisely tuned in order to ensure the fulfillment of both the main and secondary tasks. This problem on the setting of **W** has been noticed by Chaumette and Marchand[63]. They developed a tuning method for joint-limits avoidance, and tested it in a 9-DOF manipulator. This method avoids the joint limit by damping the joint motion after it passing over a preset activation threshold. It requires the manipulator has multiple redundant DOF, and is not suitable for the 1-DOF redundant tasks discussed earlier.

As closing to the singular state, there is much higher possibility on the occurrence of the joint "jerk" motion, and the solution in numerical method has lower accuracy. Moreover, it is also noticed that the joint-limits occurrence have high relevant with the configuration singular state. Thus, the joint-limits and singularity avoidance must

be considered together in order to reach a reachable trajectory.

Unfortunately, the result reached by TWA is also very sensitive to the setting of weights as the classical GPM, *i.e.*, both of their success rely on the weighting vector (the relative importance of the secondary task wrt. the main task), that has to be tuned in order to succeed with the optimization process. If badly chosen, the task may fail. Until now, the vector is usually set heuristically, *i.e.*, by trial-and-error method.

For a 6-axis serial manipulator, the weighting vector is composed of six weights, hence the tuning procedure becomes a searching problem into a six-dimensional space. Here, we try to reduce the dimension of the search space to two by identifying the two most important weights. Then a self-tuning algorithm is developed to search the best weighting vector.

This chapter presents the self-tuning method of weights, and shows three application examples. The main content of this chapter has been included in the journal paper[72] submitted to the journal of Robotics and Computer Integrated Manufacturing.

5.2 Joint-Limits and Singularity Avoidances

When the two secondary tasks of joint-limits and singularity avoidance are combined into a unique performance criterion, which is to maintain the manipulator as close as possible to the mid-joint position $\bar{\boldsymbol{\theta}}$ and as far as possible to singularities, both at the same time, the objective function z is given as in eq.(4.8).

Substituting eqs.(4.7) and (4.1) into (4.8), there is

$$z = \frac{1}{2}(\boldsymbol{\theta} - \bar{\boldsymbol{\theta}})^T \mathbf{W}^T \mathbf{W} (\boldsymbol{\theta} - \bar{\boldsymbol{\theta}}) + \frac{1}{2\omega_{mom}} (\boldsymbol{\theta} - \boldsymbol{\theta}_{Ts})^T \mathbf{K}^T \mathbf{K} (\boldsymbol{\theta} - \boldsymbol{\theta}_{Ts}). \qquad (5.1)$$

The self-motion related to the two sub-tasks is adjusted by tuning matrices $\mathbf{W}$ and $\mathbf{K}$. In this context, vector $\mathbf{h}$ used in eqs.(2.25) and (3.8) is chosen as minus the gradient

of z, *i.e.*,

$$\mathbf{h} = -\boldsymbol{\nabla} z = \mathbf{W}(\bar{\boldsymbol{\theta}} - \boldsymbol{\theta}) + \mathbf{K}\omega_{ps}(\boldsymbol{\theta}_{Ts} - \boldsymbol{\theta}), \tag{5.2}$$

where $\mathbf{W}$ and $\mathbf{K}$ are recalled to be defined as

$$\mathbf{W} \equiv \mathbf{Diag}(\mathbf{w}) \quad \mathbf{K} \equiv \mathbf{Diag}(\mathbf{k}). \tag{5.3}$$

The selection of weights in vectors $\mathbf{w}$ and $\mathbf{k}$ is very important for the success of the optimization. If the values of $\mathbf{w}$ and $\mathbf{k}$ are too small or too large, the redundant displacement may not be sufficient to avoid the joint-limits or singularities. Obviously, it is not efficient to manually tune these weights. Thus, a self-tuning method for weights becomes necessary.

5.3 Weights Self-Adaptation System

5.3.1 Overall Description

A weights self-adaptation system is proposed here. This system is composed by three main components described as follows :

1. Identification of weights sensitivity (IWS) ;
2. Joint trajectory performance evaluation algorithm (JTPEA) ;
3. Linear search method (LSM) in 2D convex space ;

Algorithm 4.1 : Weights Self-adaptation System

1 Set $\mathbf{w}_{ini}$ and $\mathbf{k}_{ini}$;

2 TWA $\rightarrow$ joint trajectory with $\mathbf{w}_{ini}$ and $\mathbf{k}_{ini}$;

3 IWS $\rightarrow$ most sensitive weights;

4 LSM $\rightarrow$ $\mathbf{w}_{adp}$ and $\mathbf{k}_{adp}$;

5 TWA $\rightarrow$ joint trajectory with $\mathbf{w}_{adp}$ and $\mathbf{k}_{adp}$;

6 JTPEA $\rightarrow$ better or worse than the previous

If (Better)

Go ahead along same searching direction

with same step length;

else (Worse)

Change searching direction or step length.

7 If (ΔPerform > Tol)

Go to step 4;

else if (ΔPerform < Tol for three times)

Terminate the adaptation.

The overall tuning procedure is shown as Algorithm 4.1. Firstly, from Steps 1 to 3, the initial set weights $\mathbf{w}_{ini}$ and $\mathbf{k}_{ini}$ are applied to the TWA, and the most sensitive weights are identified. In steps 4 and 5, based on LSM in 2D convex space, the weights are tuned to $\mathbf{w}_{adp}$ and $\mathbf{k}_{adp}$, TWA runs again by using $\mathbf{w}_{adp}$ and $\mathbf{k}_{adp}$ and reaches a new joint trajectory. In step 6, the new generated joint trajectory is evaluated by JTPEA in the aspect of the distance to joint limits and singularity. In step 7, the difference of the trajectory performance between the last two steps is calculated, if it is less than the preset tolerance for three times continually, then we say the tuning has been convergence and terminate the tuning procedure.

The three main components of the system are presented with more detail in the following.

5.3.2 Identification of Weights Sensitivity

Most of industrial 6-axes serial manipulators are decoupled, and hence, the EE orientation is controlled by the last three joints, *i.e.*, from the 4th to the 6th joints. In the cases of redundant EE rotation, the weights of the last three joints have more influence on the redundant motion than the other weights. In other words, the last three weights are the most sensitive ones, we only need to tune those three instead of all the six.

Let vector $\mathbf{e}_t$ represents the redundant joint axis, and vectors $\mathbf{e}_4$, $\mathbf{e}_5$ and $\mathbf{e}_6$ the corresponding 4th, 5th and 6th joint axes, the following relations hold from the orthogonal decoupling of the manipulator, *i.e.*,

$$\mathbf{e}_6 \perp \mathbf{e}_5, \quad \mathbf{e}_5 \perp \mathbf{e}_4 \tag{5.4}$$

The influence of the three weights on the redundant rotation around $\mathbf{e}_t$ depends on the relative orientation between $\mathbf{e}_6$ and $\mathbf{e}_t$, *i.e.*,

- Case 1 : $\mathbf{e}_t \perp \mathbf{e}_6 \Rightarrow$ 6th weight is not important $\Rightarrow$ *Tuning 5th and 4th weights.*
  In the case of $\mathbf{e}_6$ is perpendicular to $\mathbf{e}_t$, *i.e.*, $\mathbf{e}_t \perp \mathbf{e}_6$, the 6th element has not influence on the redundant motion. Hence, only the 5th and 4th weights need to be tuned .
- Case 2 : $\mathbf{e}_t \parallel \mathbf{e}_6 \Rightarrow \mathbf{e}_t \perp \mathbf{e}_5 \Rightarrow$ 5th weight is not important $\Rightarrow$ *Tuning 6th and 4th weights.*
  In the case where $\mathbf{e}_6$ is parallel to $\mathbf{e}_t$, *i.e.*, $\mathbf{e}_t \parallel \mathbf{e}_6$, we have $\mathbf{e}_6 \perp \mathbf{e}_5$, it is apparent that $\mathbf{e}_t \perp \mathbf{e}_5$, the 5th element has no influence on the redundant rotation. Thus, we only need to tune the 6th and 4th weights.
- Case 3 : Neither $\mathbf{e}_t \perp \mathbf{e}_6$ nor $\mathbf{e}_t \parallel \mathbf{e}_6 \Rightarrow$ *All three weight elements are relevant.*
  In the case where $\mathbf{e}_t$ is neither parallel nor perpendicular to $\mathbf{e}_6$, all the three joints contribute to the redundant rotation, we can detect their sensitivity by testing each

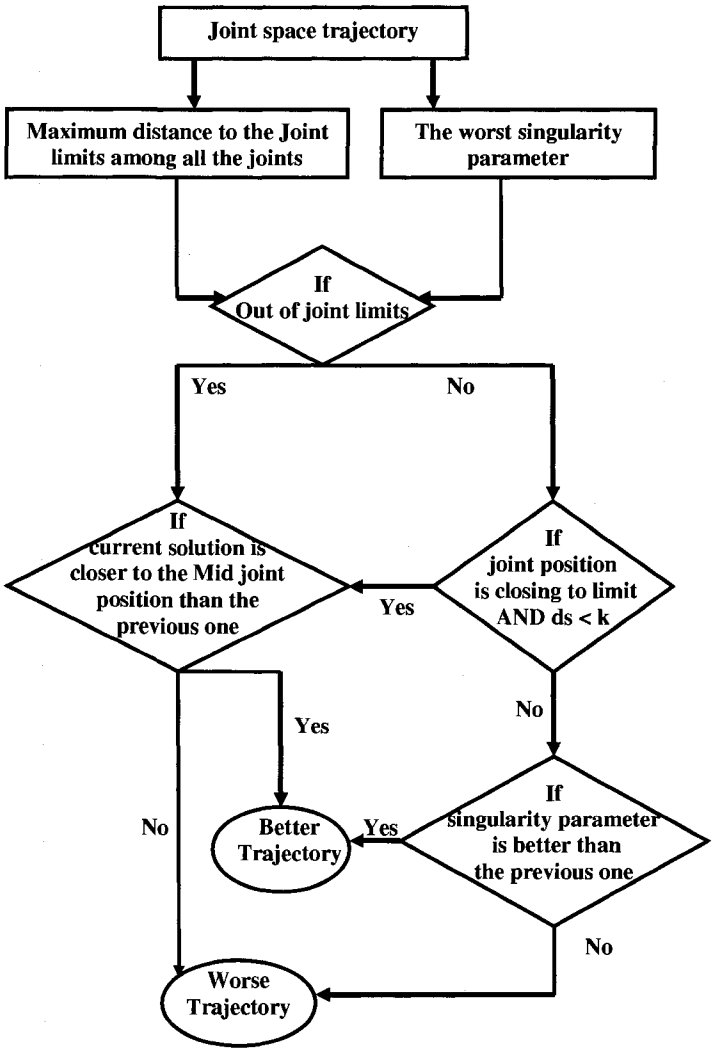
weight independently, and select only one or two weights with higher sensitivity for tuning.

5.3.3 Joint Trajectory Performance Evaluation

Both the joint limits and singularity avoidance are considered at the same time in our generated trajectory performance evaluation, so we define the greatest distance to the joint limit, and the lowest singularity parameter as our evaluation criteria. The distance is defined as positive if out-of-limit, and negative if within-limit.

The evaluation process is shown in Fig. 5.1. After the joint space trajectories of each joint are reached by solving IKP, the greatest distance to the joint limits among all the joints and the worst singularity parameter along the trajectories are picked out as the performance criterion value of this joint space solution.

The first step of evaluation is to check whether all the joint motion are within their joint limits or not. If any joint motion is out of the limit, the joint limits avoidance has higher priority to implement and only the distance to the joint limits is consider as the evaluation criterion. If no joint motion is out of the limit, but the motion is very close to the limit at some instants, and the difference of the singularity parameter is less than the set constant k, the distance to the joint limits is consider as the evaluation criterion. Otherwise, the parameter of singularity is consider.



ds --- the difference of the singularity parameter between the current trajectory with the previous one;
k --- constant, set as 0.5;

FIG. 5.1 Trajectory performance evaluation.

If all joint motions are far away with the joint limits, only the singularity parameter is consider as the evaluation criterion.

5.3.4 Linear Search Method

Because the optimization tasks require to avoid both the joint-limits and singularity, the two weighting vectors, **w** and **k**, need to be tuned at the same time. Both of **w** and **k** are 6 dimensional vectors, so there are twelve elements in total. It is difficult to search the best values in a twelve dimensional space.

With the help of the identification of the most sensitive weighting elements, only two elements of each weighting vectors need to be tuned. Thus, the dimension of the search space is reduced to four.

In our search algorithm, the two elements are tuned one by one. For example, in the case of tuning the 6th and 4th elements, firstly the 6th elements of **w** and **k** are tuned, secondly the 4th elements are tuned as applying the optimized value of the 6th elements. This adapting process could even be repeated several times to reach the solution with the best performance. In this method, the dimension of searching space is always two, since each time only one element of **w** and one of **k** are tuned. The searching space is assumed to be 2D convex. Hence, a simple linear direct search method is carried out to find the minimum value.

Figure 5.2 shows the searching process in a 2-dimension convex space with unknown lowest point `R`. The searching start point is at `o`, the weight value is tuned step by step along the searching direction `d1`, which is always and only applied as the first searching direction, until point `m+1`. Since the performance at point `m+1` is worse than at point `m`, `m` is treated as the best point along `d1`. Next, the search direction is changed to `d2` with `m` as new start point. Along `d2`, point `n` with the best performance is reached. Then, the searching direction is changed again to `d3`, along which the best

performance point is at point p. If p is still not good enough, next search will be along d2 from p. In briefly, the searching direction is in the order as : d1, d2, d3, d2, d3, ... The searching process terminates as the performance criterion convergent to a certain value. In the linear search method, the search direction and the step length

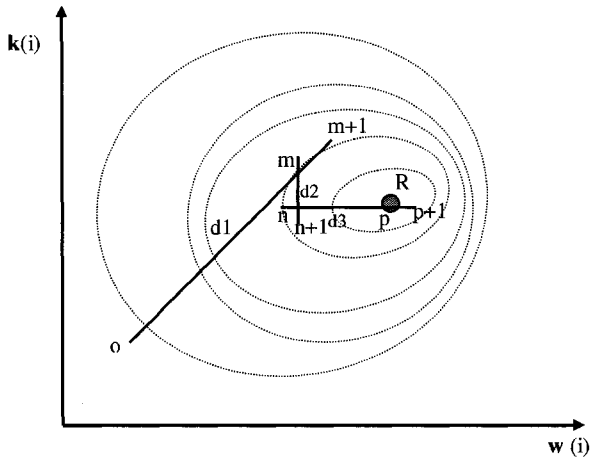


FIG. 5.2 Linear direct search method.

are the two most important issues. Here, three different searching directions are used. If searching along direction d1, the step length of $\mathbf{w}(i)$ and $\mathbf{k}(i)$ are the same ; if along direction d2, the step length of $\mathbf{w}(i)$ is keep as zero, only search along $\mathbf{k}(i)$. On the contrary, if along direction d3, the step length of $\mathbf{k}(i)$ is keep as zero.

In order to increase the tuning accuracy without loosing too much speed, the average value of current $\mathbf{w}(i)$ and $\mathbf{k}(i)$ is used as initial step length for a new search direction. After the best point using this step length is reached, the finer search around this point is carried out while step length is reduced 60%. This process may be repeated several times until the step length is shorter than a preset tolerance, or the tuning do not greatly influent the performance. In this method, the weight value along one search direction is finely tuned.

5.4 Numerical Examples

The self-adaptation system has been tested with many different robotic manipulators and tasks. Among them, we present below three examples corresponding to the three possible cases of orientation of $\mathbf{e}_t$ relative to $\mathbf{e}_6$ together with two different manipulators. All examples have been implemented on a robotic off-line programming software called Robotmaster[71]. Moreover, distance are expressed in meters, while angles are in radians.

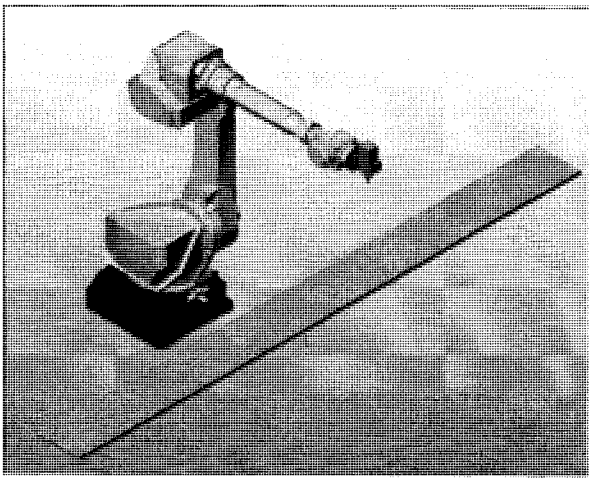


FIG. 5.3 Example 1 : Fanuc M710 with $\mathbf{e}_t \perp \mathbf{e}_6$ and the rectangular part.

5.4.1 Example 1 : $\mathbf{e}_t \perp \mathbf{e}_6$

As shown in Fig. 5.3, a Fanuc M710 robot with a tool axis $\mathbf{e}_t \perp \mathbf{e}_6$ is used to track a rectangular path. The weights are initially set as,

$$\mathbf{w}_{ini} = \mathbf{k}_{ini} = \begin{bmatrix} 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.01 \end{bmatrix}. \tag{5.5}$$

Without using the self-adaptation system, we obtain the joint trajectory shown in Fig. 5.4. Joint 4 goes from -1.047 to $+1.05$ *rad.*, which is out of the limits (*i.e.*, -0.78 to $+0.78$ in *rad.*).

Once we execute the self-adaptation system, it detects that the 6th weights are not sensitive since $\mathbf{e}_t \perp \mathbf{e}_6$, and hence, it turns the 4th and 5th weights as follow

$$\mathbf{w} = \begin{bmatrix} 0.01 & 0.01 & 0.01 & 0.0807 & 0.0167 & 0.01 \end{bmatrix}, \tag{5.6}$$

$$\mathbf{k} = \begin{bmatrix} 0.01 & 0.01 & 0.01 & 0.073 & 0.0128 & 0.01 \end{bmatrix}. \tag{5.7}$$

The corresponding joint trajectory is shown in Fig. 5.5. The motion range of 4th joint goes from -0.2437 to 0.2445, only 32% of the previous range and within the joint limits.

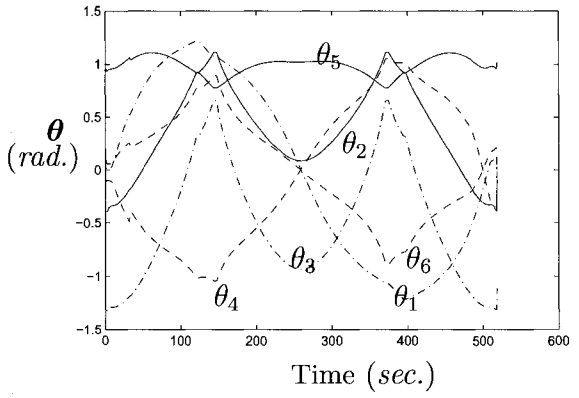


FIG. 5.4 Example 1 : Joint trajectory with the initial weights.

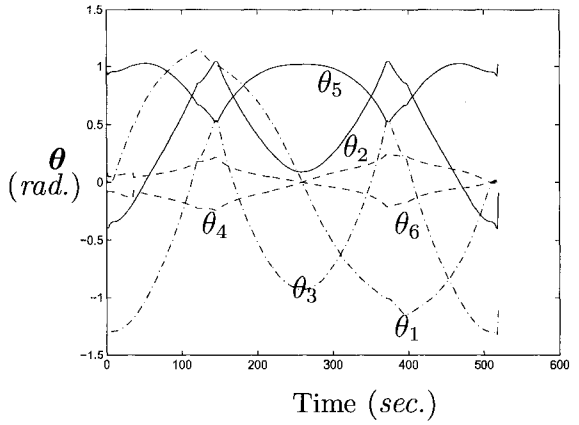


FIG. 5.5 Example 1 : Joint trajectory with the optimized weights.

5.4.2 Example 2 : $\mathbf{e}_t \parallel \mathbf{e}_6$ (Test IV)

As shown in Test I, II and III, the Fanuc M710 robot with a tool axis $\mathbf{e}_t \parallel \mathbf{e}_6$ is used to perform the path shown in Fig. 3.8. The weights are initially as in eq.(4.13). Without using the self-adaptation system, we obtain the joint trajectory shown in Fig. 4.2.

Once we execute the self-adaptation system, it detects that the 6th weight is sensitive, since $\mathbf{e}_t \parallel \mathbf{e}_6$, and hence, it tunes the weights as follow

$$\mathbf{w} = \begin{bmatrix} 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.0483 \end{bmatrix}, \tag{5.8}$$

$$\mathbf{k} = \begin{bmatrix} 0.01 & 0.01 & 0.01 & 0.01 & 0.01 & 0.0256 \end{bmatrix}. \tag{5.9}$$

The corresponding joint trajectory is shown in Fig. 5.6. Apparently, joint 6 goes from -0.9707 *rad.* to 0.1864 *rad.*, a range of only 21% of the one obtained without the self-adaptation system. Moreover, the minimum measure of manipulability along the trajectory is enhanced by 247%, *i.e.*, from 0.02954 to 0.07323 as shown in Fig. 5.7. The self-adaptation system has succeed in keeping the manipulator further from its joint-limits and singular configurations.

As we known, in Test III, the sixth elements of the two applied weights are $\mathbf{w}(6) = 0.03$ and $\mathbf{k}(6) = 0.05$, respectively, these weights are obtained heuristically. Now in Test IV, there are $\mathbf{w}(6) = 0.0483$ and $\mathbf{k}(6) = 0.0256$. The reached joint trajectories of Test III and IV is quite similar, but 6th joint motion range obtained in Test IV is smaller than the one in Test III, *i.e.*, the result reached in Test IV is better than Test III in the defined performance criterion.

Two consecutive turns of the path is implemented in this testing case, it is observed that the joint trajectories of each turn is cyclical, *i.e.*, the manipulator posture reaching one path point is identical between each successive turns.

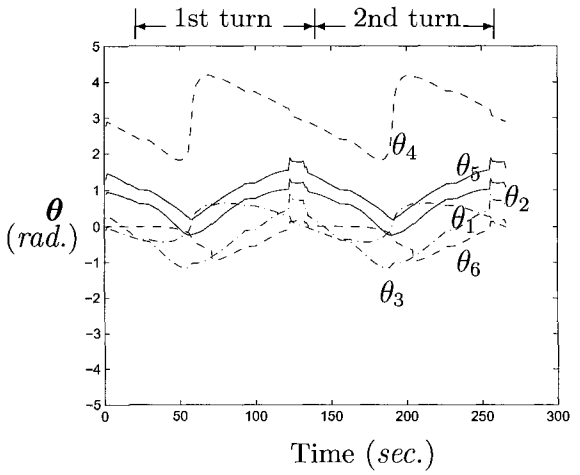


FIG. 5.6 Example 2 : Joint trajectory with the optimized weights.

5.4.3 Example 3 : neither $\mathbf{e}_t \parallel \mathbf{e}_6$ nor $\mathbf{e}_t \perp \mathbf{e}_6$

As shown in Fig. 5.8, the Fanuc M16 robot with a tool axis neither parallel nor perpendicular to $\mathbf{e}_6$ is used to perform the path shown in Fig. 5.9. The weights are initially set as in eq.(5.5). Without using the self-adaptation system, we obtain the joint trajectory and manipulability shown in Fig. 5.10. Once we execute the self-adaptation system, it is not clear which weights are the most sensitive, since $\mathbf{e}_t$ is neither parallel nor perpendicular to $\mathbf{e}_6$, and hence, three tests must be performed.
Test 1 : When we tune only the 5th and 4th weights, we obtain :

$$\mathbf{w} = \begin{bmatrix} 0.01 & 0.01 & 0.01 & 0.017 & 0.0279 & 0.01 \end{bmatrix}, \tag{5.10}$$

$$\mathbf{k} = \begin{bmatrix} 0.01 & 0.01 & 0.01 & 0.034 & 0.0987 & 0.01 \end{bmatrix}. \tag{5.11}$$

Test 2 : When we tune only the 5th and 6th weights, we obtain :

$$\mathbf{w} = \begin{bmatrix} 0.01 & 0.01 & 0.01 & 0.01 & 0.0354 & 0.0291 \end{bmatrix}, \tag{5.12}$$

$$\mathbf{k} = \begin{bmatrix} 0.01 & 0.01 & 0.01 & 0.01 & 0.1167 & 0.0213 \end{bmatrix}. \tag{5.13}$$

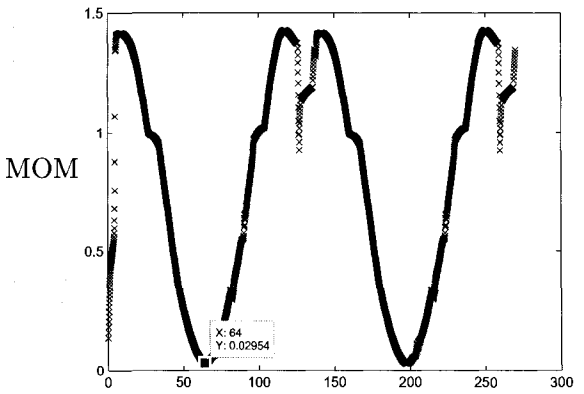


FIG. 5.7 Example 2 : The manipulability reached with (a) initial weighting vectors, and (b) optimize weighting vectors as eqs.(5.8, 5.9).

Test 3 : When we tune only the 6th and 4th weights, we don't reach a solution. The joint trajectories corresponding to Test 1 and 2 are shown in Fig. 5.11, while the manipulability are shown in Fig. 5.12. The results of the two tests are compared in Table 5.1. Apparently, both tests can reach better results than the initial weights. Test 2 can reach a result with higher manipulability and smaller joint motion range than Test 1.

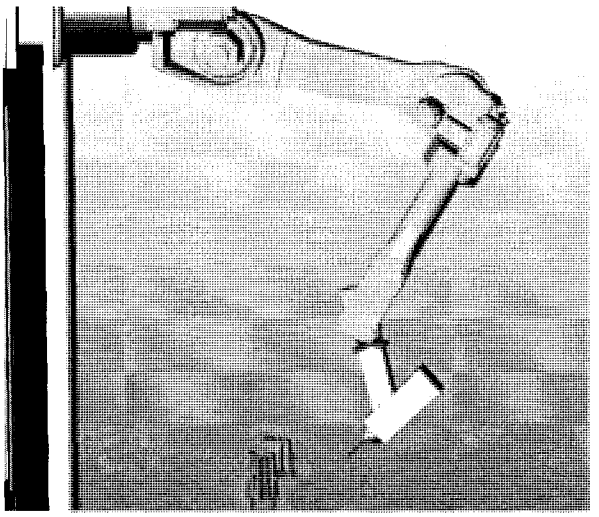


FIG. 5.8 Example 3 : Fanuc M16 robot and the working part

	Initial	Test 1	Test 2
min(MOM)	0.197	0.3098	0.4075
Joint out-of-limit	4th and 5th	none	none

TAB. 5.1 Performance results of Example 3.

5.5 Conclusion

This chapter studied the weighting vector tuning problem of the TWA, and proposed a weighting vector self-adaptation algorithm for the six-axis decoupled manipulators. The identification of the most sensitive weighting elements is classified into three cases relating to the geometry relation between the tool and the 6th joint. By the identification, the tuning problem is simplified from 6 dimensions to 2 dimensions.

With the help of the tuning algorithm, the TWA on avoiding the joint-limits and singularity becomes more robust, since their success highly relies on the setting of the weighting vectors. Corresponding to the three different geometric cases, three application tasks using the weighting vector self-adaptation algorithm are presented,

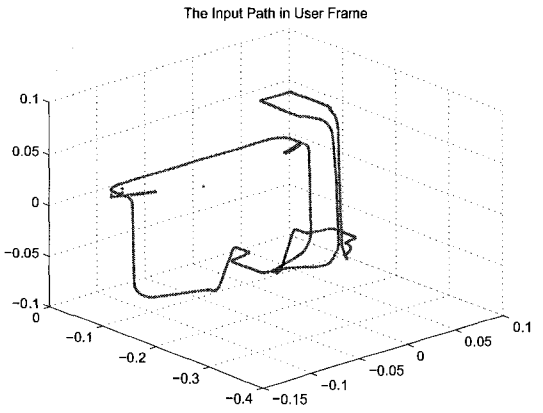


FIG. 5.9 Example 3 : The path in the part frame.

and the algorithm has been proved to improve the functional redundancy resolution results.

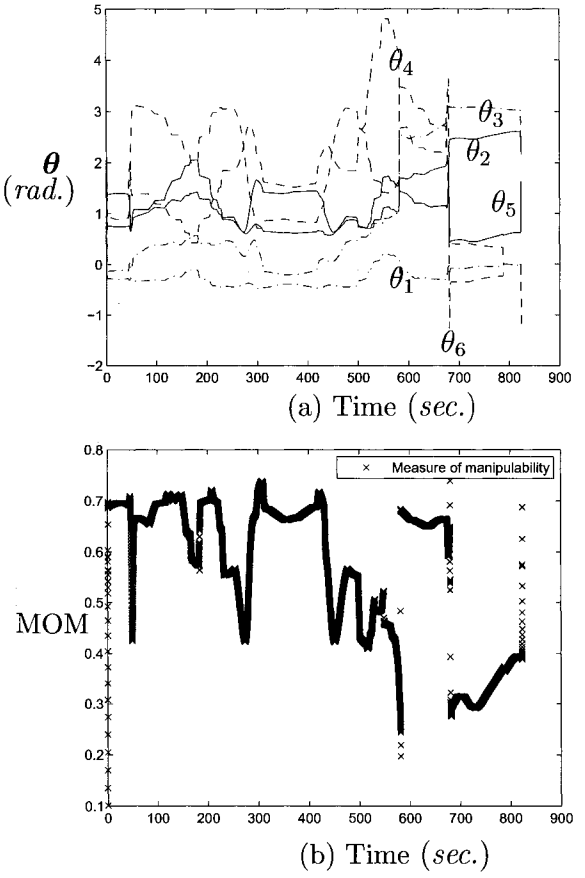


FIG. 5.10 Example 3 : Joint trajectory(a) and manipulability(b) with the initial weights.

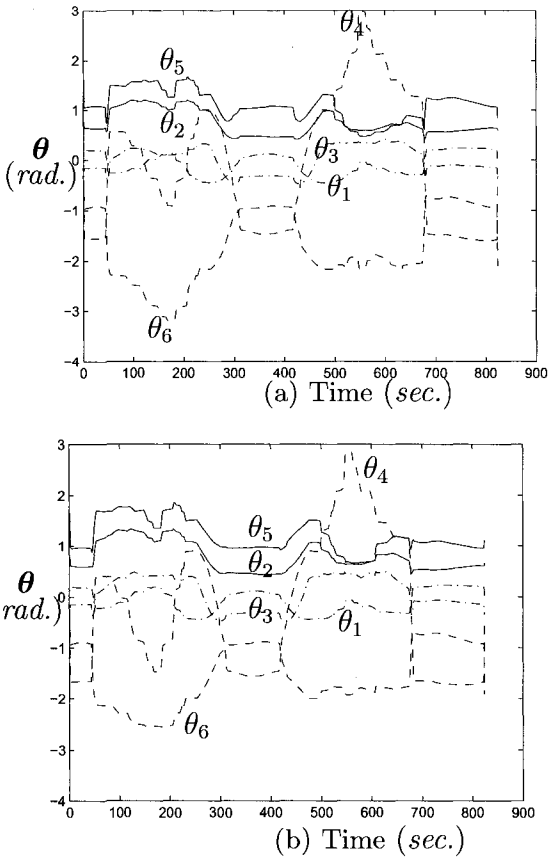


FIG. 5.11 Example 3 : Joint trajectory of Test 1 (a) and Test 2 (b) with the optimized weights.

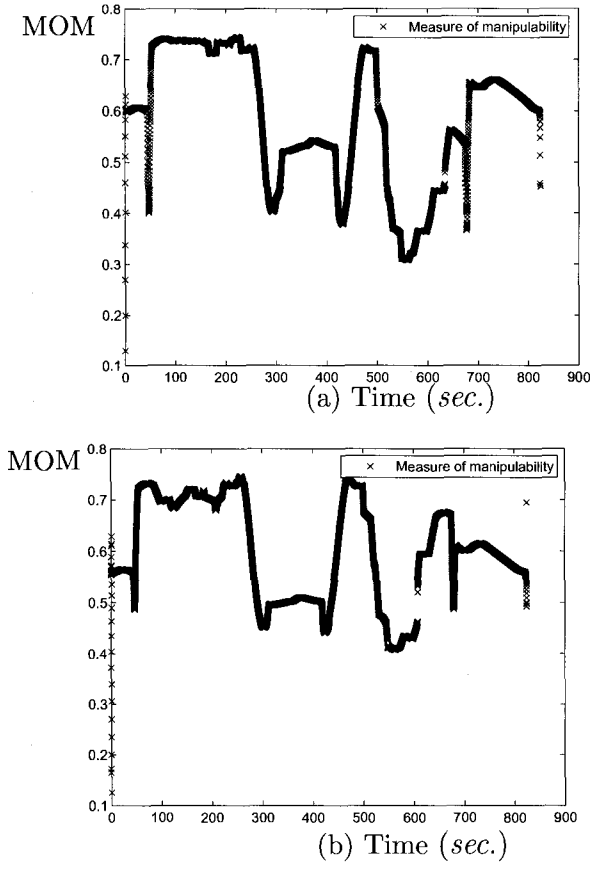


FIG. 5.12 Example 3 : The manipulability of Test 1 (a) and Test 2 (b) with the optimized weights.

CHAPITRE 6

DYNAMIC-ADAPTATION OF WEIGHTS IN TWA

6.1 Introduction

As presented in Chapter 5, a self-adaptation method of weights has been developed. This self-adaptation method searches a set of weighting parameters, and uses these values for the whole task. However, there are some tasks with complicate Cartesian path for industrial application. In these tasks, the different segment of a task usually have different optimal requirements. For example, segment A of a task requires to optimize the joint motion range, while segment B of the same task requires to optimize the singularity parameters. The weighting vector with fixed values for the whole path can only be a trade-off among different segments. Moreover, this trade-off value may even not exist in some industrial tasks. Apparently, these tasks require weights with different values at each instant or segment, instead of fixing them for the whole task. In this chapter, a dynamic-adaptation system of weights is developed for the tasks with a symmetric tool as implemented on a 6-revolute decoupled serial manipulators. This dynamic-adaptation system has been integrated into TWA, and is applied to cases with multiple secondary tasks, which consider not only the optimization of joint-limits and singularity, but also the joint velocities.

The main content of this chapter has been included in the paper[73] submitted to Industrial Robot : An International Journal.

6.2 Weights Dynamic-Adaptation System

The problem of adapting dynamically the weights includes the following subproblems :

1. When do the weights need to be adapted?
2. Which weight component need to be adapted?
3. How much do the weights need to be adapted?

Below, the three subproblems are analyzed and a series of solutions are proposed.

6.2.1 Subproblem I

In the case of joint-limit avoidance, when the ith joint is approaching to its limits, the weights for the joint-limit avoidance should be increased in order to bring the joint back to its mid-position. Thus, we define a parameter, namely $\boldsymbol{\delta}$, to measure the distance from the current joint position to its limits θ_i^{limit}, *i.e.*,

$$\boldsymbol{\delta} \equiv [\delta_1, \cdots, \delta_6]^T, \quad \text{where} \quad \delta_i = \frac{\theta_i - \theta_i^{limit}}{\theta_i^{max} - \theta_i^{min}}, \quad 1 \leq i \leq 6. \tag{6.1}$$

In the case of singularity avoidance, when the manipulator is approaching to singular configurations, the weights for singularity avoidance should be increased in order to bring the manipulator away from singularity. The measure of manipulability ω_{mom} is used to measure the distance from current configuration to the singularity.

Thus, the weights is adapted when the joint configuration becomes close to its motion-limits or singularities.

6.2.2 Subproblem II

In order to find which weight component that need to be adapted, we analyze the contribution of each joint on the rotation around $\mathbf{e}_t$. As presented in Section 5.3, the first three joints do not have contribution on this redundant rotation since the 6R serial manipulators are decoupled, the first three weights are kept constant at 0.01.

For the last three joints, the contribution of ith joint is greatly influenced by the angle β_i between the tool symmetry axis $\mathbf{e}_t$ and the joint rotation axis $\mathbf{e}_i$. Consequently, this angle β_i is a good parameter for detecting the influence of the ith joint on the redundant displacement. Thus, angle β_i is used to decide which weight component need to be adapted. Once, a threshold value β_{TH} is set, the weight component w_i is adapted when $\beta_i < \beta_{TH}$.

6.2.3 Subproblem III

Another question that the dynamic adaptation needs to answer is : *how much the weights need to be adapted ?* It requires to develop a function which is able to express the relationship between weights and some input variables. In order to reach this objectives, we have to solve the following three problems :

- What are these input variables influencing the weights adaptation ?
- What is the relationship between weights and these input variables ?
- How to express the adaptation function in a mathematic form ?

6.2.3.1 Input Variables Influencing Weights

In the subproblems I and II, there are two vectors and one scalar variables influencing the weights adaptation, *i.e.*, three-dimensional vector $\boldsymbol{\beta}$, which detects the influence of the last three joints on the redundant displacement ; six dimensional vector $\boldsymbol{\delta}$, which measures the distance from the current joint position to its limits ; and the scale ω_{mom}, which measures the distance from current configuration to singularities.

6.2.3.2 Relationship between Weights and Input Variables

From the geometry between the tool symmetry axis $\mathbf{e}_t$ and the joint rotation axis $\mathbf{e}_i$, we can express the influence of ith joint by $|\mathbf{e}_i^T \mathbf{e}_t|$. *i.e.*,

$$\beta_i = \arccos(|\mathbf{e}_i^T \mathbf{e}_t|), \tag{6.2}$$

where the angle β_i is $0 \leq \beta_i \leq \pi/2$.

As the angle β_i is decreasing to 0, the rotation around $\mathbf{e}_t$ becomes sensitive on the value of weight w_i.

6.2.3.3 Adapting Function Evaluation

In the case of joint limit avoidance, the last three elements of weights $\mathbf{w}$ could be expressed as a function f_{joint} of β_i and δ_i, *i.e.*,

$$w_i = f_{joint}(\beta_i, \delta_i), \quad 4 \leq i \leq 6. \tag{6.3}$$

If the joint limits problem occurring on the first three joints, the minimum δ_{min} among all elements of $\boldsymbol{\delta}$ is used in eq.(6.3) to replace δ_i. Then, eq.(6.3) becomes

$$w_i = f_{joint}(\beta_i, \delta_{min}), \quad 4 \leq i \leq 6, \tag{6.4}$$

where

$$\delta_{min} = \min\{\delta_i, 1 \leq i \leq 6\}. \tag{6.5}$$

In terms of singularities, the last three weights of $\mathbf{k}$ could be expressed as a function f_{sing} of β_i and ω_{mom}, *i.e.*,

$$k_i = f_{sing}(\beta_i, \omega_{mom}), \quad 4 \leq i \leq 6. \tag{6.6}$$

By the experiment of testing, f_{joint} and f_{sing} are shown to be case dependent functions, and there are so great differences among various application cases, *i.e.*, it does not exist a single general function of f_{joint} and f_{sing}. Here, a function evaluation method is proposed.

Firstly, $\mathbf{w}$ and $\mathbf{k}$ are decomposed into two parts, respectively, as

$$\mathbf{w} = \mathbf{a} + c_{joint}\mathbf{f}; \tag{6.7}$$

$$\mathbf{k} = \mathbf{b} + c_{sing}\mathbf{g}; \tag{6.8}$$

where c_{joint} and c_{sing} are damping scalars for adjusting the functions, and are set to zero by default. Both $\mathbf{a}$ and $\mathbf{b}$ are six-dimensional vectors, their first three elements are constant value as 0.01, whereas their last three elements are defined as

$$a_i = \frac{q(2 - (\frac{\beta_i}{\beta_{th}})^2)}{2\delta_i}, \quad \forall \quad 4 \leq i \leq 6; \tag{6.9}$$

$$b_i = \frac{m(1 - (\frac{\beta_i}{\beta_{th}})^2)}{\exp(2\omega_{mom})}, \quad \forall \quad 4 \leq i \leq 6. \tag{6.10}$$

The threshold angle for activating the adaptation is set to be β_{th}. So if there is $\beta_i \leq \beta_{th}$, a_i and b_i are calculated by eqs.(6.9) and (6.10), otherwise they just keep as their default value 0.01. The coefficient q in eq.(6.9) is set to $0.01rad$. The coefficient m in eq.(6.10) is set to 0.25 without unit, since measure of manipulability ω_{mom} is just a scalar value.

Moreover, $\mathbf{f}$ and $\mathbf{g}$ are also six dimensional vectors, with the first three elements as zero, and the last three are defined as

$$f_i = \frac{l(1 - (\frac{\beta_i}{\beta_{th}})^2)}{\exp(2\delta_i)}, \quad \forall \quad 4 \leq i \leq 6; \tag{6.11}$$

$$g_i = \frac{s(1 - (\frac{\beta_i}{\beta_{th}})^2)}{\exp(2\omega_{mom})}, \quad \forall \quad 4 \leq i \leq 6. \tag{6.12}$$

Here both the two coefficients l and s are set to 1.

All the four functions defined in eqs.(6.9) to (6.12) are plotted in 3-dimensional space as in Figs. 6.1 and 6.2, respectively.

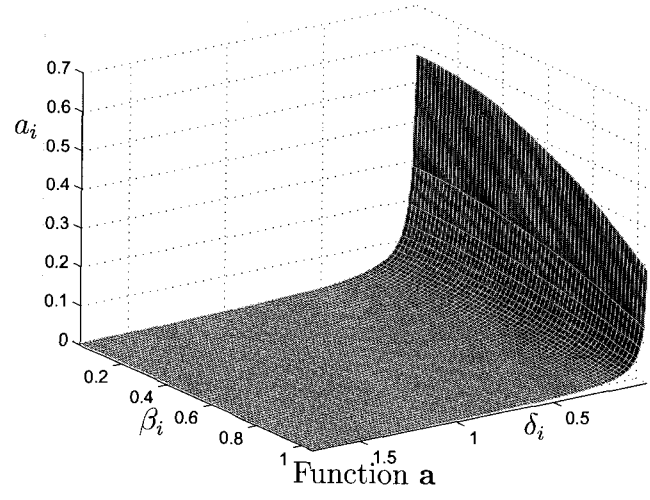
6.2.4 Dynamic-Adaptation Algorithm

The weights dynamic-adaptation algorithm has been integrated with TWA, the whole processing procedure is shown in Fig. 6.3.

Firstly, the c_{joint} and c_{sing} are set to the default value of 0, so we obtain the following equations from eqs.(6.7) and (6.8),

$$\mathbf{w} = \mathbf{a}, \tag{6.13}$$

$$\mathbf{k} = \mathbf{b}. \tag{6.14}$$



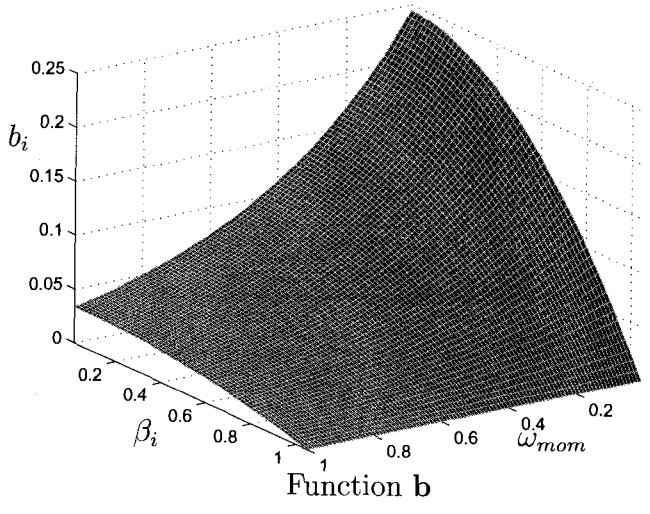


FIG. 6.1 Function **a** and **b**.

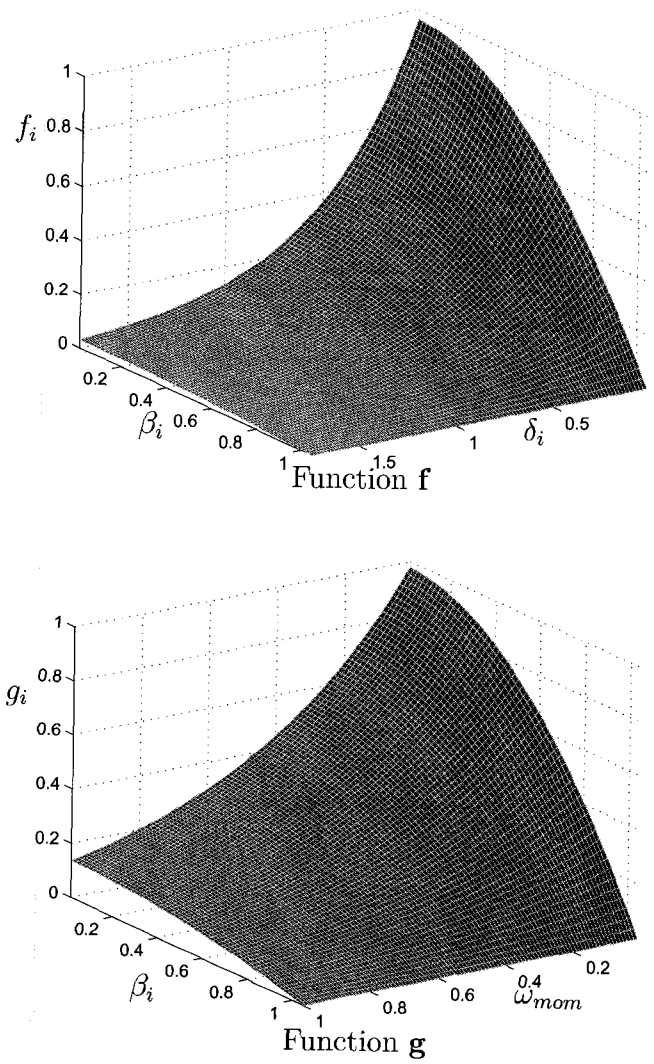


FIG. 6.2 Function f and g.

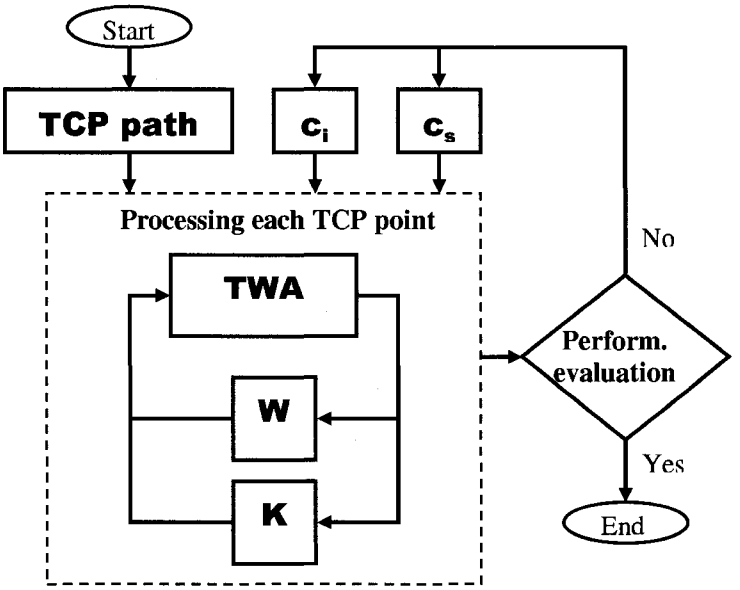


FIG. 6.3 Processing scheme of dynamic adaptation.

After finishing the processing on the whole input tool center point (TCP) path with the initial c_{joint} and c_{sing}, the computation is completed if the tentative trajectory with eqs.(6.13) and (6.14) satisfy the optimization criteria. Otherwise, the trajectory must be analyzed to detect the segments which are out of the joint limits, close to singularity or with high joint velocities. The instant along the tentative trajectory where the performance indices is the worst is called the critical instant of this trajectory. In order to refine **w** and **k** to meet the optimization criteria around this critical instant, c_{joint} and c_{sing} are adapted based on the following three principles :

- increase c_{joint} to minimize the joint motion range;
- increase c_{sing} to move away from singularity;
- reduce c_{joint} or c_{sing} to decrease joint velocities.

Apparently, the dynamic weights adaptation problem has been reduced to adapt two damping factors c_{joint} and c_{sing}, instead of two six-dimensional weights, and hence, the automatic adaption becomes much easier. The adaptation may be repeated for several times until reaching a satisfying result. More details of the processing are

shown in Algorithm 6.1, where t_1 and t_2 defined as the adaptation period of time around the critical instant, usually $10 \sim 20\%$ of the total time.

Algorithm 6.1 :

Weighting Vector Dynamic Adaptation Algorithm

1 $c_{joint} = c_{sing} = 0$;

2 TWA;

If the trajectories meet the task requirements;

Stop

3 Else, if the trajectories have problems at instant t

Δc_{joint}, Δc_{sing}, t_1, t_2, $d \Leftarrow$ adaptation principle;

adapting period $\Leftarrow t - t_1$ to $t + t_2$;

Go to step 2;

6.3 Numerical Examples

The weights dynamic adaptation method has been integrated into TWA, and successfully tested along different industrial tasks using Fanuc, Motorman and ABB 6-R decoupled robots, respectively, including the tasks examples shown in Section 5.4. Most of these tasks are succeed with $c_{joint} = c_{sing} = 0$, and it even does not need to implement step 3 of Algorithm 6.1. Here, one task requiring a careful adaptation on c_{joint} and c_{sing} is shown for demonstration purposes.

The same task of arc-welding path and Fanuc M-710 iC/50 robot described in Section 5.4.1 is used, but with a different welding tool. The angle between the tool symmetry axis $\mathbf{e}_t$ and the 6th joint axis $\mathbf{e}_6$ is 60 *deg*, instead of 0 *deg*. Figure 6.4 shows the Fanuc M-710 iC/50 with the new tool.

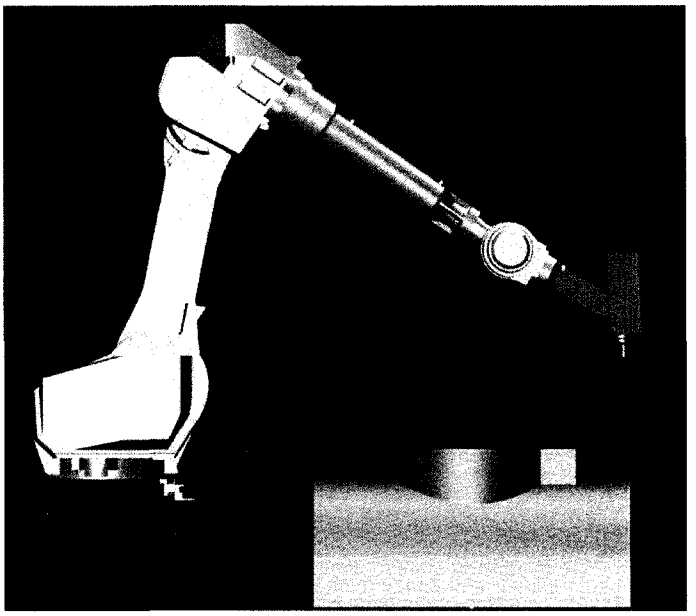


FIG. 6.4 Fanuc M710 iC/50 with new tool.

6.3.1 Test A : No Weights Adaptation

The joint space trajectory with the initial weights as in eq.(5.5) is shown in Fig. 6.5. The maximum angles of the 5th and 6th joints reach, respectively, 134 *deg* and 417 *deg*, while their corresponding limits being 120 *deg* and 360 *deg* (see Table 3.5). Obviously, this task can not be completed with these weights, and hence, they need to be adapted.

6.3.2 Test B : Adaptation with $c_{joint} = c_{sing} = 0$

Since the angle between $\mathbf{e}_t$ and $\mathbf{e}_6$ is 60 *deg*, all the last three joints are relevant to the redundant motion, and the corresponding three weights need to be adapted. After applying Algorithm 6.1 with the default setting of c_{joint} and c_{sing} at zero to adapt the weights, the joint motion range is greatly decreased. As shown in Fig. 6.6, the maximum angles of the 5th and 6th joints are reduced to only 97 *deg* and 130 *deg*.

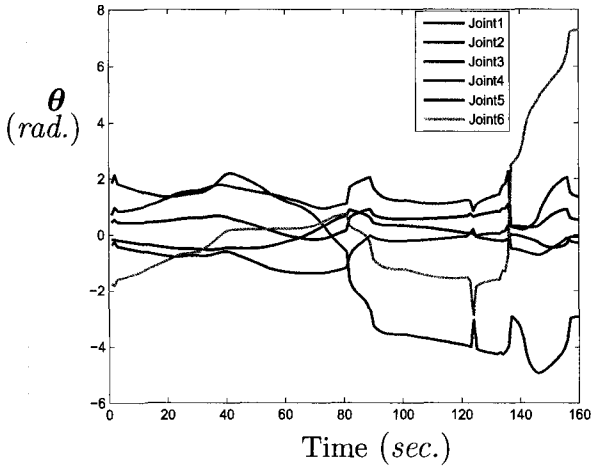


FIG. 6.5 Joint space trajectory reached with the initial weighting vectors.

(where the limits are recalled to be 120 *deg* and 360 *deg*). All joint position are now within the limits, but some high joint velocities appear around the instant 84, as shown in Fig. 6.7. The highest joint velocities reach 102 *deg/sec* and 74 *deg/sec* for 4th and 6th joints, respectively. Consequently, the task is still very difficult to be performed and deserved additional adaptation.

6.3.3 Test C : Adaptation with $c_{joint} = 0$ and $c_{sing} = -0.5$

In order to reducing joint velocities, step 3 of Algorithm 6.1 is activated. The instant 84 is identified as the critical instant, according to the adaptation principle, values of c_{joint} or c_{sing} are reduced in order to decreasing the joint velocity. After several time of adaption, c_{sing} is set to -0.5 during the period from instants 54 to 114. The optimized joint trajectory and velocities are shown in Figs. 6.8 and 6.9. The highest velocities of the 4th and 6th joints are decreased from 102 *deg/sec* to 27 *deg/sec*, and from 74 *deg/sec* to 13 *deg/sec*, respectively. The optimized trajectory shown in Fig. 6.8 is much smoother than the one in Fig. 6.6, so the trajectories becomes easier to implement on a real Fanuc M-710 iC/50. Moreover, the ω_{mom} is slightly

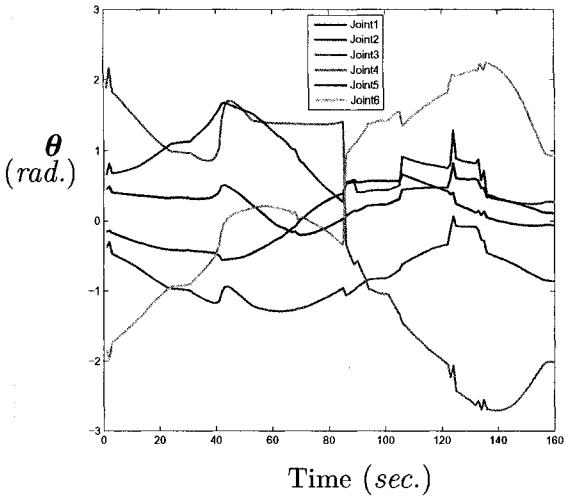


FIG. 6.6 Joint space trajectory reached with the default setting of $c_{joint} = c_{sing} = 0$.

decreased. Table 6.1 summarizes the results concerning the motion range, motion velocity and singularity of this task. Clearly, the results reached with the adaptation of $c_{sing} = -0.5$ are the best among the three tests.

Method	*Motion range*		*Motion velocity*		*Singularity*
	Max. 5*th*	*Max.* 6*th*	*Max.* 4*th*	*Max.* 6*th*	*min.* ω_{mom}
No adaptation	134	417	450	235	0.1276
$c_{joint} = c_{sing} = 0$	97	130	102	74	0.1431
$c_{joint} = 0$; $c_{sing} = -0.5$	96.6	130	27	13	0.12
unit	deg	deg	deg/sec	deg/sec	

TAB. 6.1 Testing results of weights dynamic adaptation.

As shown in eq.(4.10), TWA resolved the joint displacement in order to satisfy three sub-tasks, *i.e.*, the sub-displacement required to move the EE, redundant sub-displacement 1 and redundance sub-displacement 2. The norm of these three sub-displacements can be calculated. Figure 6.10 shows the sub-displacements corresponding to Fig. 6.8.

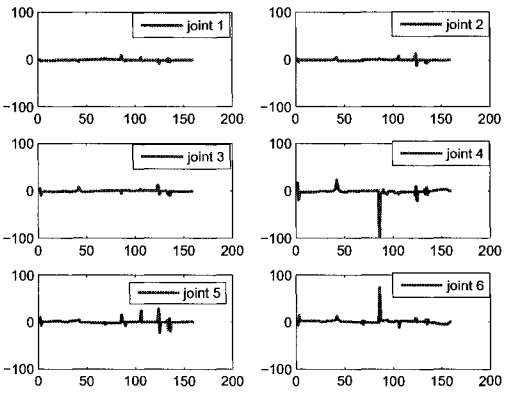


FIG. 6.7 Joint velocities reached with the default setting of $c_{joint} = c_{sing} = 0$.

6.4 Conclusion

This chapter studied the weights adaptation problem of the functional redundancy resolution, and proposes a weights dynamic adaptation for six-revolute decoupled serial manipulators. This algorithm makes the functional redundancy resolution not only robust on avoiding the joint-limits and singularity, but also expanding to the minimization of the joint velocity. Corresponding to different geometry relation between the tool and the 6th joint in several different application tasks, the weights dynamic adaptation algorithm is tested and reaches satisfying results. In the numerical example, the different secondary tasks are fulfilled with the help of weights adaptation, the empirical functions (eqs.(6.9) to (6.12)) are able to successfully adapting weights.

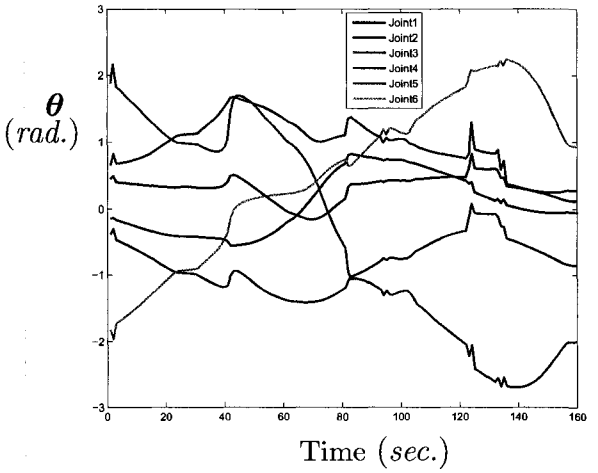


FIG. 6.8 Joint space trajectory reached with the setting of $c_{joint} = 0$, $c_{sing} = -0.5$ around the instant 84.

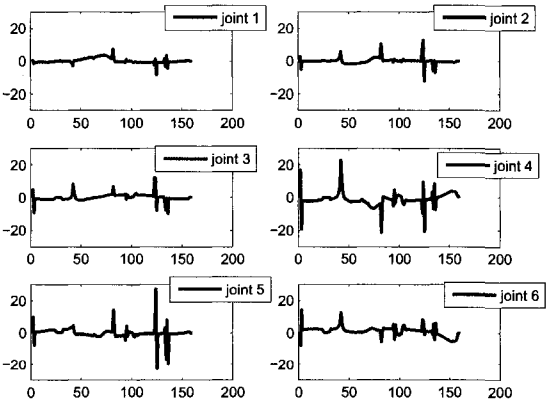


FIG. 6.9 Joint velocities reached with the default setting of $c_{joint} = 0$, $c_{sing} = -0.5$ around the critical instant 84.

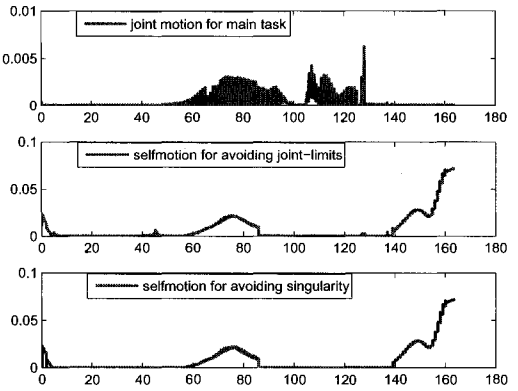


FIG. 6.10 Sub-displacements reached with the three components of eq.(4.10).

CHAPITRE 7

CONCLUSIONS

7.1 TWA Summary

In this thesis, TWA has been studied for the general functionally-redundant robotic tasks. The main applications of this thesis are on the domain of industrial robot programming. In order to implement these tasks by industrial robots, which are, in general 6-R decoupled manipulators, the robotic joint space trajectory is often required to be optimized to avoid joint limits, obstacles, singularities, etc. In industry, robot programmers have to optimize joint space trajectory based on their experience and skill, but it is very inefficient and has low success rate. With the help of TWA, these tasks can be optimized automatically.

Besides of the development of TWA, the multi-secondary tasks optimization and two weighting parameters adaptation methods are developed in this thesis. Hence, TWA is not only able to generate robot trajectory far away from its joint limits, but also from singularity with these weights adaptation methods. These features really make TWA usable for industrial application.

7.2 Original Contribution

The main contribution of this thesis are presented from Chapter 3 to Chapter 6. In Chapter 3, TWA is developed based on the orthogonal decomposition of vectors. The twist decomposition algorithm is presented in Algorithm 3.1. Three application examples with joint-limits avoidances are solved with TWA.

In Chapter 4, singularities-avoidance is additionally included into the optimization objective. Manipulability and conditioning are also analyzed in this chapter. One example requiring both of joint-limit and singularity avoidances is shown. The great influence of weights on the reaching solution is observed in this example.

In Chapter 5, a self-adaptation method of the weights in TWA is proposed. Firstly, by identifying the weights sensitivity, the dimension of search space is reduced from 6-dimension to 2-dimension. Then, a linear search algorithm is used to adapt the weights according to defined performance criteria. The adaptation procedure is presented in Algorithm 4.1. This self-adaptation method can reach the optimized weights automatically after multiple cycles of adaption. However, the self-adaptation method, although optimized, always uses a constant set of weights all along the trajectory. Consequently, it can not meet the requirement of some application tasks, where a variable set of weights is required.

In Chapter 6, a dynamic-adaptation method of weights is proposed. This method first identifies some input variables which have great influence on weights, then reaches the weights by a series of empirical functions representing the relationship between weights and these input variables. This method provides a variable set of weights along the trajectory, hence, it can meet the stricter optimization requirement than self-adaptation method. The processing scheme and algorithm of dynamic-adaptation are presented in Fig. 6.3 and Algorithm 6.1, respectively.

7.3 Future Works

In this thesis, the TWA has only been studied in the context of functional redundancy related to tool having a symmetry axis. However, functional redundancy also exists for symmetric parts in the case of pick and place operations as introduced in Section 1.2. It needs further development to integrate TWA with these pick and place tasks.

The TWA allows for taking advantages of functional redundancy. However, when functional and intrinsic redundancies are both presented, the TWA alone can not take advantage of both redundancies without generalizing the TWA as we have proposed. This issue deserves further study.

During the process of optimization, we notice that the selection of an initial posture starting a path greatly affect the optimized trajectory. A bad selection may even cause the failure of the optimization. In this thesis, it is observed that the posture may be optimized after one fully turn of the path in some test cases. But it still deserves further study in order to develop a more effective and rational way of selecting this initial posture.

Finally, we only apply TWA to kinematically decoupled manipulators, although over ninety percent of industrial robots belong to this kind of manipulators, the coupled manipulators still exist in the market for some special applications and because they do not have wrist singularity problem. Thus, it also needs further study on applying TWA to coupled manipulators.

RÉFÉRENCES

[1] Angeles, J., *Fundamentals of robotic mechanical systems : theory, methods and algorithms*, Springer, New York, 521 pages, 2003.

[2] Mitsubishi Heavy Industries, LTD., http ://www.sdia.or.jp/mhikobe_e/products/mechatronic/specfi/specfi_e.html, 2004.

[3] Laval University, http ://robot.gmc.ulaval.ca/research/theme103, 2009.

[4] Laval University, http ://robot.gmc.ulaval.ca/research/theme103, 2009.

[5] COMET-II, Chiba University, http ://mec2.tm.chiba-u.jp/%7Enonami, 2004.

[6] Huo, L., and Baron, L., *Kinematic inversion of functionally-redundant serial manipulators : application to arc-welding*, CSME Transactions, Vol. 29, No. 4, pp. 679-690, December 2005.

[7] Nakamura, Y., *Advanced robotics : redundancy and optimization*, Addison-Wesley Pub. Co., Massachusetts, 337 pages, 1991.

[8] Eric W. Weisstein. *Null Space.* MathWorld–A Wolfram Web Resource. http ://mathworld.wolfram.com/NullSpace.html.

[9] Sciavicco, L. and Siciliano, B., *Modelling and control of robot manipulators*, Springer, London, 377 pages, 2000.

[10] Mavroidis, C. and Roth, B., 1992 *Structural parameters which reduce the number of manipulator configurations.* Proc. ASME 22nd Biennial Mechanisms Conference, pp.359-366, Sept. 1992.

[11] Pieper, D.L., *The Kinematics of Manipulators under Computer Control,* Ph.D. thesis, Stanford University, 1968.

[12] Tsai, L.-W. and Mrogan, A. P., *Solving the kinematics of the most general six- and five-degree-of-freedom manipulators by continuation methods.* ASME J. Mechanisms, Transm., and Auto. in Design, pp. 189–200, 1985.

[13] Whitney, D.E., *Resolved motion rate control of manipulators and human prostheses.* IEEE Trans. Man-Machine Syst., vol. 10, no. 2, pp. 47–53. 1969.

[14] Itô, K., *Encyclopedic dictionary of mathematics*, MIT Press, 1986.

[15] Whitney, D.E., *The mathematics of coordinated control of prosthetic arms and manipulator,* ASME J. Dynamics Systems, Measurement and Control, Vol. 94, No. 4, pp. 303–309, 1972.

[16] Baillieul, J., *Kinematic programming alternative for redundant manipulator,* Proc. International Conference on Robotics and Automation, St-Louis, pp. 722–728, 1985.

[17] Baillieul, J., *Avoiding obstacles and resolving kinematic redundancy*, IEEE International Conference on Robotics and Automation, Washington, pp. 1698–1704, 1986.

[18] Klein, C. A., *Use of redundancy in the design of robotic systems,* 2nd, Int. Symp. of Robotics Research, MIT Press, pp. 207–214, August 1984.

[19] Honegger, M. and Codourey, A., *Redundancy resolution of a cartesian space operated heavy industrial manipulator*, IEEE International Conference on Robotics and Automation, Vol. 3, pp. 2094–2098, 1998.

[20] Yashi, O.S., and Ozgoren, K., *Minimal joint motion optimization of manipulators with extra degrees of freedom,* Mechanism and Machine Theory, Vol. 19, No. 3, pp. 325–330, 1984.

[21] Angeles, J., Anderson, K. and Gosselin, C., *An Orthogonal-Decomposition Algorithm for Constrained Least-Square Optimization,* ASME Robotics, Mechanisms and Machine Systems, Design Eng. Division, Vol. 2, pp. 215–220, 1987.

[22] Siciliano, B., *Solving manipulator redundancy with the augmented task space methode using the constraint Jacobian transpose,* IEEE Int. Conf. on Robotics and Automation, Tutorial M1, pp. 5.1–5.8, 1992.

[23] Weisstein, E.W., *Moore-Penrose Matrix Inverse.* MathWorld–A Wolfram Web Resource, http ://mathworld.wolfram.com/Moore-PenroseMatrixInverse.html.

[24] Klein, C. and Huang, C.H., *Review of pseudoinverse control for use with kinematically redundant manipulators,* IEEE Trans. on Systems, Man, and Cybernetics, Vol. SMC-13, No. 3, pp. 245–250, 1983.

[25] Arenson, N., Angeles, J. and Slutski, L., *Redundancy-resolution algorithms for isotropic robots,* Advances in Robot Kinematics : Analysis and Control, pp. 425–434, 1998.

[26] Liégeois, A., *Automatic Supervisory Control of the Configuration and Behavior of Multibody Mechanisms,* IEEE Trans. Syst., Man, Cybern., vol. SMC-7, pp. 245–250, Mar. 1977.

[27] Yoshikawa, T., *Analyais and control of robot manipulators with redundancy,* Robotics Research : The First International Symposium, pp. 735–747, 1984.

[28] Salisbury, J. K. and Craig, J. J. , *Articulated hands, force and kinematic issues*, The International Journal of Robotics Research, Vol. 1, No. 1, pp. 4–17, 1982.

[29] Golub, G.H. and Van Loan, C.F., *Matrix computations,* The Johns Hopkins University Press, Baltimore and London, 1989.

[30] Kosuge, K. and Furuta, K., *Kinematic and dynamic analysis of robot arm,* IEEE Int. Conf. on Robotics and Automation, pp. 1039–1044, 1985.

[31] Park, J., Chung, W. and Youm, Y., *Weighted decomposition of kinematics and dynamaics of kinematically redundant manipulators*, IEEE International Conference on Robotics and Automation, Vol. 1, pp. 480–486, 1996.

[32] Chang, T.-F. and Dubey, R.-V., *A weighted least-norm solution based scheme for avoiding joints limits for redundant manipulators,* IEEE Trans. Robot. Automat.,vol. 11, pp. 286–292, Apr. 1995.

[33] Nakamura, Y. and Hanafusa, H., *Task priority based redundancy control of robot manipulators*, MIT Press, pp. 155–162, 1985.

[34] Nakamura, Y. and Hanafusa, H., *Optimal redundancy control of robot manipulators*, International Journal of Robotics Research, Vol. 6, No. 1, Spring, pp. 32–42, 1987.

[35] Suh, K.C., Hollerbach, J.M., *Local versus global torque optimization of redundant manipulators*, IEEE International Conference on Robotics and Automation , pp. 619–24 vol.2, 1987.

[36] K. Kazerounian and Z. Wang, *Global versus local optimazation in redundancy resolution of robotic manipulators*, The International Journal of Robotics Research, VOL.7, pp. 3–12, Oct. 1988.

[37] Z. Wang and K. Kazerounian, *An efficient algorithm for global optimization in redundant manipulators*, Journal of Mech. Trans. and Auto. in Design, 1988.

[38] Kim, S.-W. and Lee, J.-J., *Resolved motion rate control of redundant robots using fuzzy logic*, IEEE International Conference on Fuzzy Systems, pp. 333–338, 1993.

[39] Kim, S.-W. *et al.*, *Inverse kinematics solution based on fuzzy logic for redundant manipulators*, Int. Conf. Intell. Rob. Syst., pp. 904–910, 1993.

[40] Xu, Y. and Nechyba, M. C., *Fuzzy inverse kinematic mapping : rule generation, efficiency, and implementation*, Int. Conf. Intell. Rob. Syst., pp. 911–918, 1993.

[41] Ramos, M.C. and Koivo, A.J., *Fuzzy logic-based optimization for redundant manipulators*, IEEE Transactions on Fuzzy Systems, vol. 10, no. 4, August, 2002, pp. 498–509.

[42] Graca, Randy A. and Gu, Y.-l., *A fuzzy learning algorithm for kinematic control of a robotic system*, Proceedings of the IEEE Conference on Decision and Control, Vol. 2, pp. 1274–1279, 1993.

[43] Teraco, T. *et al.*, *Fuzzy systems theory and its applications*, Boston ; Toronto : Academic Press, 269 pages, 1992.

[44] Graca, Randy A. and Gu, You-Liang, *Application of the fuzzy learning algorithm to kinematic control of a redundant manipulator with subtask optimization*, IEEE International Conference on Fuzzy Systems, Vol. 2, pp. 843–848, 1994.

[45] Kung, S. Y. and Hwang, J.N., *Neural network architectures for robotic application*, IEEE Trans. Robot. Automat., vol. 5, pp. 641–657, 1989.

[46] Guo, J., and Cherkassky, V., *A solution to the inverse kinematic problem in robotics using neural network processing*, Proc. IEEE Int. Joint Conf. Neural Networks, Washington, DC, 1989, vol. 2, pp. 299–304.

[47] Gardner, J. F. *et al.*, *Applicaitons of neural networks for coordinate transformations in robotics*, J. Intell. Robot. Syst., vol. 8, pp. 361–373, 1993.

[48] Lee, S. and Kil, R. M., *Redundant arm kinematic control with recurrent loop*, Neural Networks, vol. 7, pp. 643–659, 1993.

[49] Mao, Z. and Hsia, T. C., *Obstacle avoidance inverse kinematics solution of redundant robots by neural networks*, Robotica, vol. 15, pp. 3–10, 1997.

[50] Ding, H. and Tso, S. K., *Redundancy resolution of robotic manipulators with neural computation*, IEEE Trans. Ind. Electron., vol. 46, pp. 199–202, Feb. 1999.

[51] Ding, H. and Wang, J., *Recurrent neural networks for minimum infinity-norm kinematic control of redundant manipulators*, IEEE Trans. Syst., Man, Cybern. A, Syst. Humans, vol. 29, no. 3, pp. 269–276, May 1999.

[52] Xia, Y. S. and Wang, J., *A dual neural network for kinematic control of redundant robot manipulators*, IEEE Trans. Syst., Man, Cybern. B, Cybern., vol. 31, no. 1, pp. 147–154, Feb. 2001.

[53] Zhang, Y. *et al.*, *A dual neural network for redundancy resolution of kinematically redundant manipulators subject to joint limits and joint velocity limits*, IEEE Trans. Neural Netw., vol. 14, no. 3, pp. 658–667, May 2003.

[54] Parker, J. K. *et al.*, *Inverse kinematics of redundant robots using genetic algorithms*, IEEE Int. Conf. Rob. Autom, pp. 271–276, 1989.

[55] Davidor, Y., *Genetic algorithms and robotics - A heuristic strategy for optimization*, World Scientific Publishing Co. Pte. Ltd., pp. 164, 1991.

[56] Nearchou, A.C. and Aspragathos, N.A., *Application of genetic algorithms to point-to-point motion of redundant manipulators*, Mechanism and Machine Theory, v. 31, n 3, pp. 261–270, April, 1996.

[57] Nearchou, A.C., *Solving the inverse kinematics problem of redundant robots operating in complex environments via a modified genetic algorithm* Mechanism & Machine Theory, v. 33, n. 3, pp. 273–292, April 1998.

[58] Samson, C. *et al.*, *Robot control : the task function approach*, Oxford [Angleterre] : Clarendon Press, 364 pages, 1991.

[59] Baron, L. *A joint-limits avoidance strategy for arc-welding robots,* International Conference on Integrated Design and Manufacturing in Mechanical Engineering, Montreal, Canada, May 2000.

[60] Nakamura, Y. *et al.*, *Task-priority based redundancy control of robot manipulators*, Int. J. Robotics Research, Vol. 6, No. 2, 1987.

[61] Baron, L. *An Optimal Surfacing Post-Processor Module for 5-Axes CNC Milling Machines,* Third International Conf. on Industrial Automation, Montréal, Canada, June 1999.

[62] Baron, L. and Huo, L., *Inverse Kinematics of Functionlly-Redundant Serial Manipulators : A Comparative Study,* 12th World Congress on the Theory of Machines and Mechanisms, Besancon, France, 2007.

[63] Chaumette, F. and Marchand, É., *A redundancy-based iterative approach for avoiding joint limits : application to visual servoing*, IEEE Trans. on robotics and automation, vol. 17, no. 5, pp. 719–730, 2001.

[64] Huo, L. and Baron, L., *The joint-limts and singularity avoidance in robotic welding*, Industrial Robot : An International Journal, Vol. 35, No. 5, pp. 456–464, August 2008.

[65] Bedrossian, N. *Classification of singular configurations for redundant manipulators*, IEEE International Conference on Robotics and Automation, pp. 818–823, 1990.

[66] Yoshikawa, T., *Foundations of robotics : analysis and control*, The MIT press, 1990.

[67] Marani, G., *et al.* , *A real-time approach for singularity avoidance in resolved motion reate control of robotic manipualtors*, IEEE International Conference on Robotics and Automation, pp. 1973-1978, 2002.

[68] Huo, L. and Baron, L. (2007), "Inverse Kinematics of Functionally-Redundant Serial Manipulators under Joint Limits and Singularity Avoidance", *Conference on Systems and Control*, Marrakech, Morocco, May 16-18.

[69] Yoshikawa, T., *Basic optimization methods of redundant manipulator*, Lab. Robotics and Aut., vol. 8, No. 1, pp. 49–60, 1996.

[70] Angeles, J. *et al.*, *Efficient algorithms for the kinematic inversion of redundant robot manipulators*, Int. J. of Robotics and Automation, vol. 2, no. 3, pp. 106–115, 1987.

[71] http ://www.robotmaster.com/.

[72] Huo, L. and Baron, L., *The self-tuning of weights for joint-limits and singularity avoidances of functionally-redundant robotic tasks*, Robotics and Computer integrated Manufacturing, submitted, 2008.

[73] Huo, L. *et al. Weighting parameters dynamic-adaptation for the joint-limits and kinematic singularity avoidance in 6-axis robotic tasks*, Industrial Robot : An International Journal, submitted, 2009.